VIRTUAL CABLE

Administration Manual
UDS Enterprise 4.0

INDEX

1. ACCESS TO UDS ENTERPRISE ADMINISTRATION 4
2. SUMMARY 7
3. SERVICES 10
3.1 Service Providers 11
3.1.1 VDI with VMware vSphere 13
3.1.2 VDI with OpenNebula 36
3.1.3 VDI with Citrix XenServer / XCP-ng 45
3.1.4 VDI with Microsoft HyperV 60
3.1.5 VDI with Proxmox 74
3.1.6 VDI environment with OpenStack 87
3.1.7 VDI with Nutanix Acropolis 100
3.1.8 VDI with Microsoft Azure 112
3.1.9 VDI with Amazon AWS 125
3.1.10 VDI environment with Google Cloud 136
3.1.11 Direct connection to remote computers 148
3.1.12 RDS Application Provider 159
3.2 Servers 171
3.2.1 Managed 171
3.2.1 Unmanaged 176
4. AUTHENTICATION 184
4.1 Authenticators 184
4.1.1 Active Directory 187
4.1.2 Entra ID 192
4.1.3 IP 197
4.1.4 Internal Database 200
4.1.5 OAuth2 204
4.1.6 Radius Authenticator 209
4.1.7 Regex LDAP 213
4.1.8 SAML 218
4.1.9 eDirectory 224
4.2 Multifactor 228

4.2.1 Email Multi Factor 229
4.2.2 Radius OTP Challenge 232
4.2.3 SMS via HTTP 235
4.2.4 TOTP Based MFA 238
4.3 Users, groups, and metagroups 240
4.3.1 Creating Groups and Users "Internal Database"" 245
4.3.2 Creating groups and users "IP Authenticator"" 247
5. OS MANAGERS 249
5.1.1 Linux Active Directory 250
5.1.2 Linux 254
5.1.3 Linux Random Password 256
5.1.4 RDS 258
5.1.5 Windows Basic 259
5.1.6 Windows Domain 261
5.1.7 Windows Random Password 265
6. CONNECTIVITY 268
6.1 Transports 268
6.1.1 NICE DCV 221
6.1.2 NoMachine 225
6.1.3 RDP 229
6.1.4 RDS For vApps 236
6.1.5 SPICE 242
6.1.6 URL Launcher 245
6.1.7 X2Go 248
6.1.8 HTML5 RDP 252
6.1.9 HTML5 RDP For vApps 256
6.1.10 HTLM5 SSH 261
6.1.11 PCoIP Cloud Access 264
6.2 Networks 267
6.3 Tunnels 270
7. POOLS 273
7.1 Service pools 273

7.2 Meta Pools 283
7.3 Groups 288
7.4 Access calendars and scheduled tasks 289
7.4.1 Calendars 290
7.4.2 Scheduled Actions: 294
7.5 Accounting 296
8. TOOLS 298
8.1 Gallery 298
8.2 Reports 299
8.3 Notifiers 300
8.3.1 Telegram 300
8.3.2 Email 302
8.4 Configuration 305
8.4.1 UDS 307
8.4.2 Security 310
8.4.3 Admin 311
8.4.4 Custom 312
8.4.5 WYSE 313
8.5 Flush Cache 314
9. PERMISSIONS 315
10. ACCESSING SERVICES 317
11. ABOUT VIRTUAL CABLE 319
UDS ENTERPRISE, THE SOFTWARE VDI OF VIRTUAL CABLE 319
About UDS Enterprise 319
About Virtual Cable 319

1. ACCESS TO UDS ENTERPRISE ADMINISTRATION

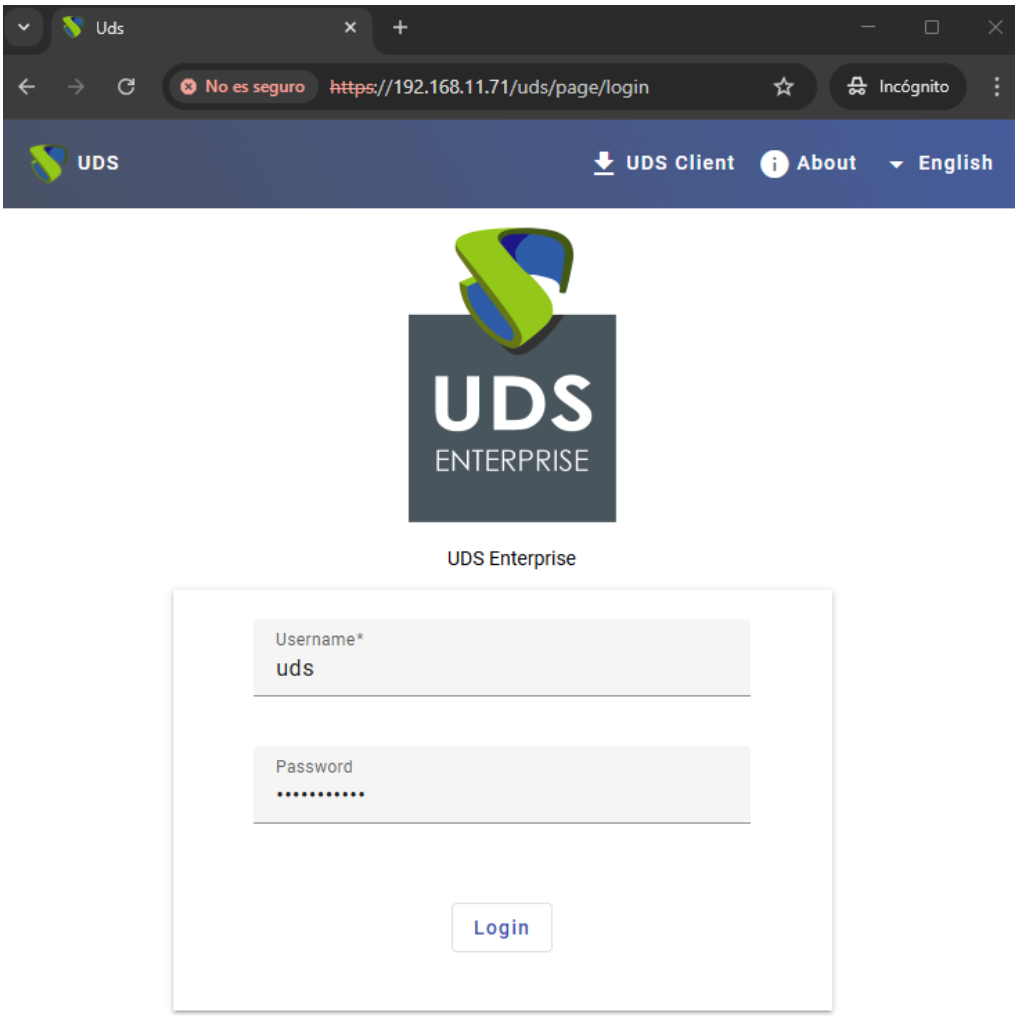
To access the UDS Enterprise administration we will need to have the "UDS Server" component installed and configured. Access will be through a web browser in a current version, with the main browsers being supported: Firefox, Chorme, Edge, Opera, Safari, etc...

We must have a user with administration permissions registered in an authenticator or use the super-user configured in the configuration wizard of the "UDS Server" component. This user will not be validated against any authenticator, being always available unless specifically disabled within the UDS advanced settings.

To access the administration, we will first access the login portal by indicating the name or IP address of the "UDS Server" component using the format:

https://*name/IP_UDSServer*

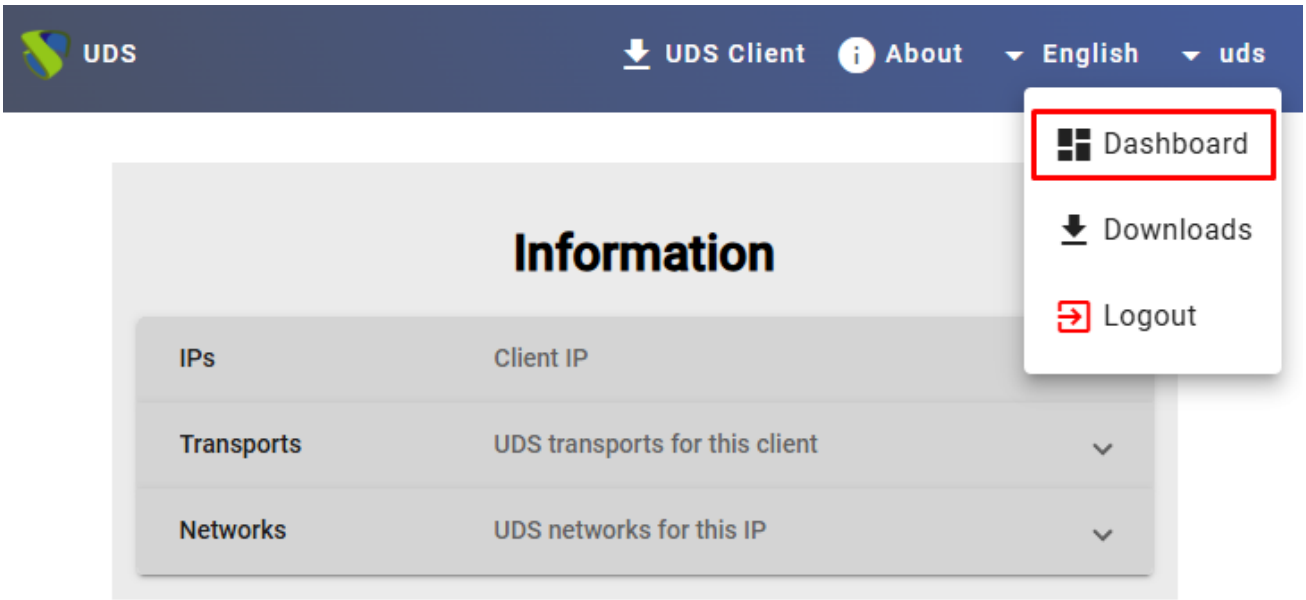
Once in the login portal, we will indicate the credentials of a user with administrator permissions:



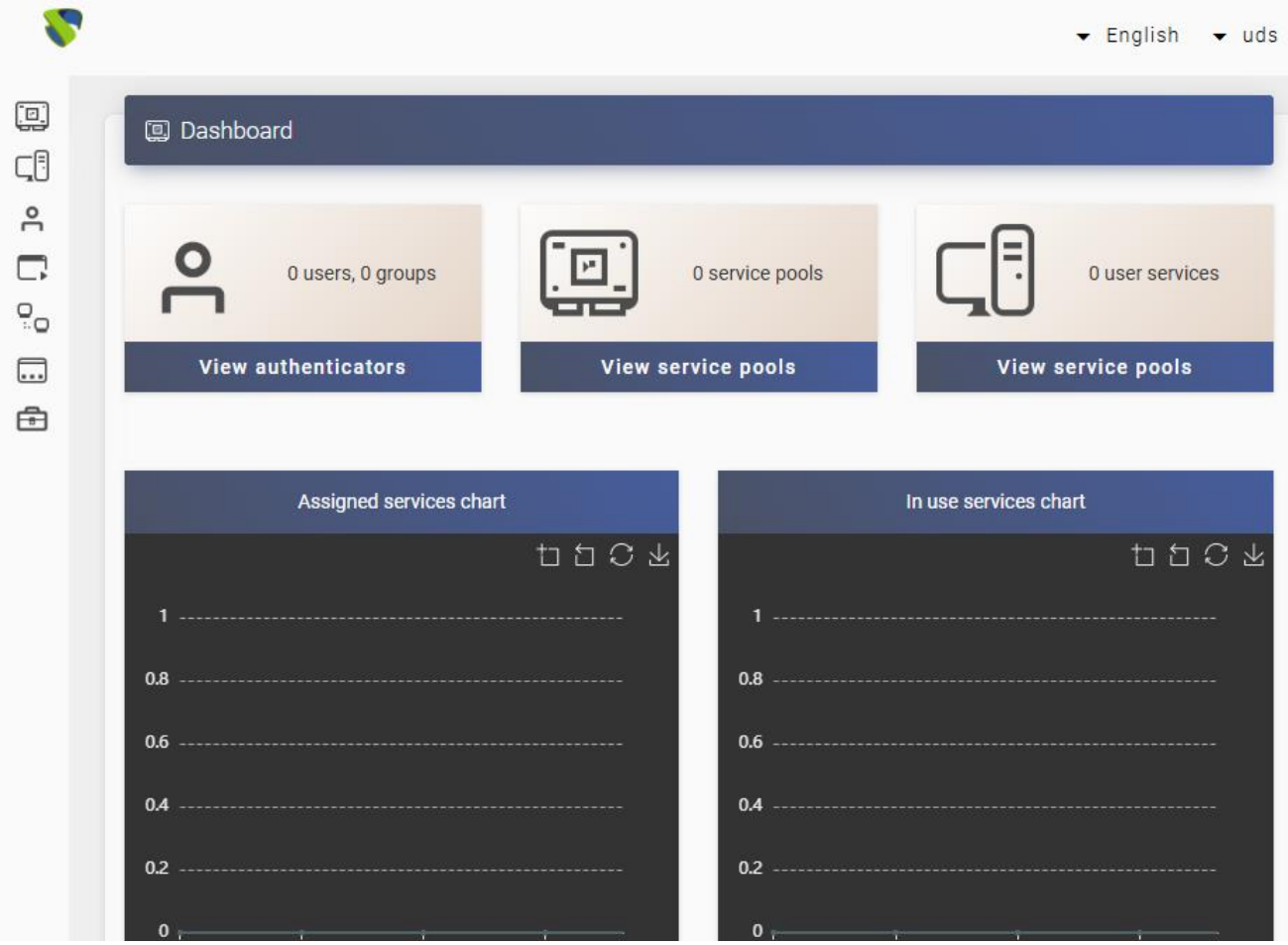
© Virtual Cable S.L.U.

NOTE: If we make the first access after the installation of the UDS server, it will always be necessary to use the credentials of the super-user (indicated in the configuration wizard of the "UDS Server" component), since we will not yet have any authentication system configured in the environment. Once you have configured an authentication system and created a user with administration permissions, it is possible to disable this super-user if necessary.

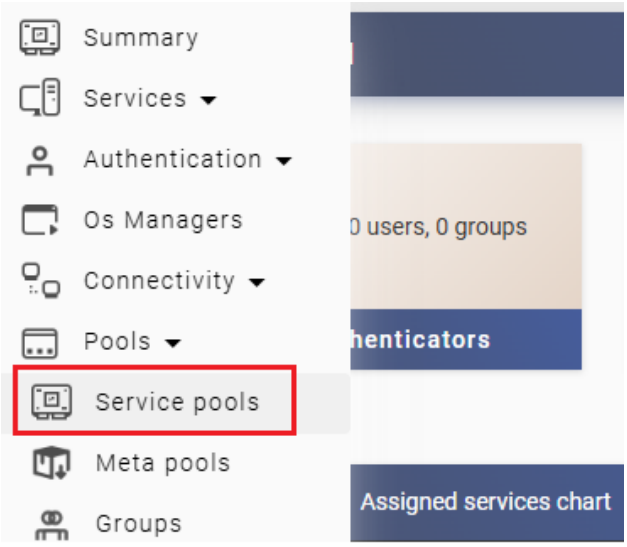
If we have indicated correct credentials, we will be allowed to access the user services portal (user mode) and also if we have administration permissions, in the user menu we will see access to the control panel to manage our UDS environment:



Once inside the UDS control panel, we will be able to configure the different elements that will make up our UDS Enterprise environment in order to publish virtual desktops, virtual application sessions, virtual desktop sessions, remote access to static computers, etc... and make them available to our users so that they can be used by them.

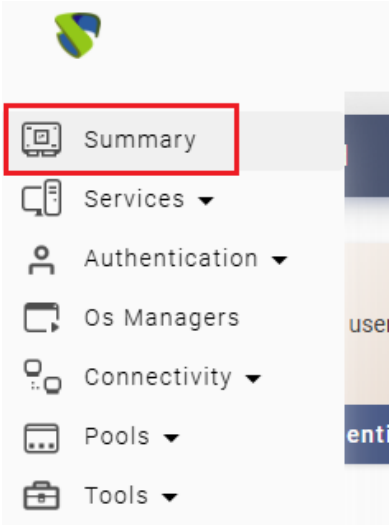


The publications of the different services will be carried out through the "Service Pool" component, which will require the existence and configuration of certain elements that will be detailed in this guide.



2. SUMMARY

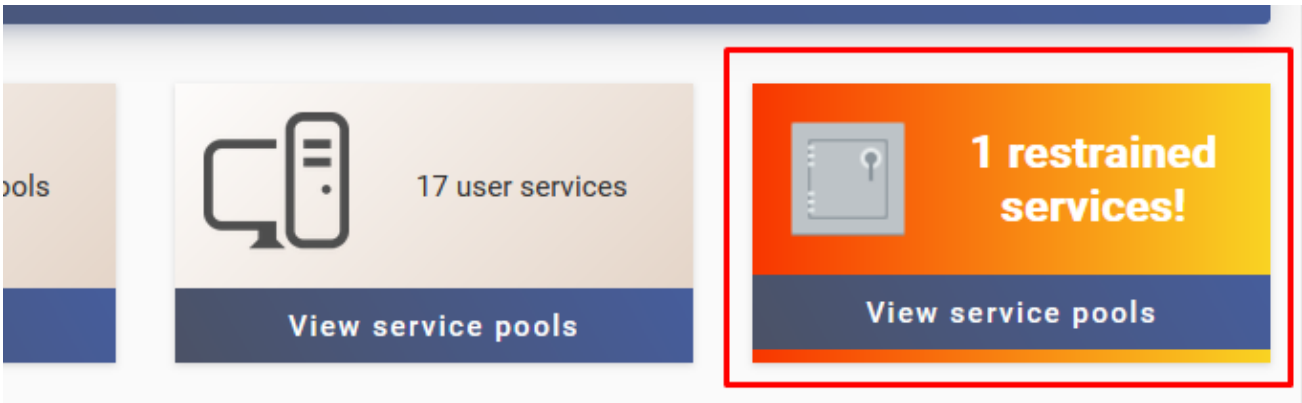
In the extendable menu located on the left of the control panel, the first item will be "Summary", which will allow us to quickly visualize the general status of our UDS environment.



En la parte superior tendremos inicialmente 3 apartados donde ver información básica de nuestro entorno:

- **Users, groups, and serviced users:** It will indicate the total sum of users, groups and users with services of all the authenticators registered in the system. By clicking on "View authenticators" we will go directly to the section: authenticators.
- **Service pools:** Total number of service pools (of any type: VDI, vAPP, static, etc...) existing in the environment. By clicking on "View service pools" we will go directly to the section: service pools.
- **User Services:** Total number of services (of any type: VDI, vAPP, static, etc...) and those assigned to users, regardless of whether they are in use "yes" or "no". By clicking on "View service pools" we will go directly to the section: service pools.

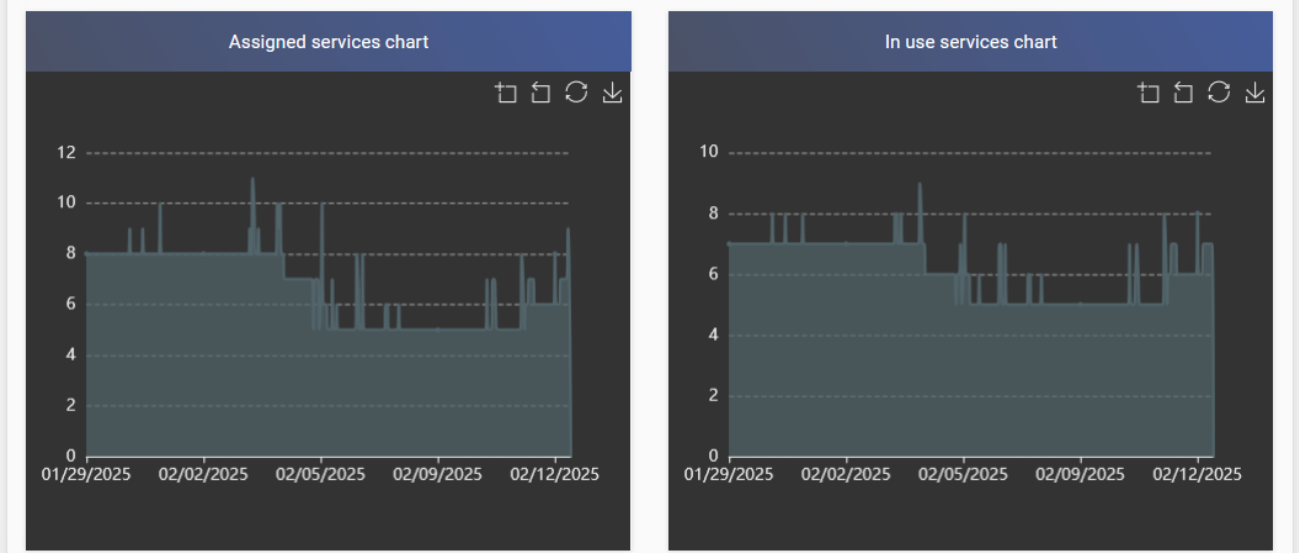
In addition to these 3 sections, we will be able to view a fourth when there are pools of restricted services in the environment. A restricted service pool will indicate that a certain number of failures have occurred during a predefined period of time.



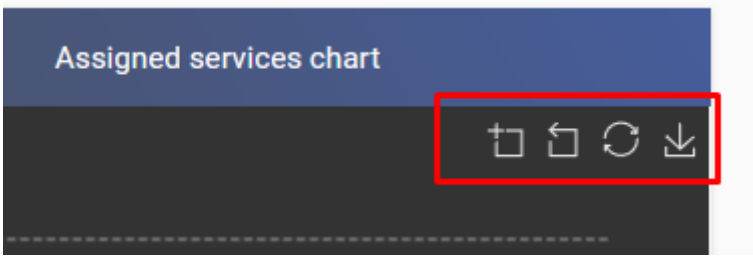
NOTE: When it is detected that there is a restricted service pool, it will be necessary to access it, open the logs of the failed services (to identify the problem) and try to solve the error. The number of errors and the time in which they have to occur for a service pool to enter restricted mode will be controlled by the variables "restrainTime" and "restrainCount" located in the advanced UDS parameters.

At the bottom of the control panel we will see two graphs, with a history of 15 days, which will show us:

- Total services assigned to users of any type (VDI, vAPPs, static, etc…).
- Total number of services assigned to users and that are in "yes" use of any type (VDI, vAPPs, static, etc...).



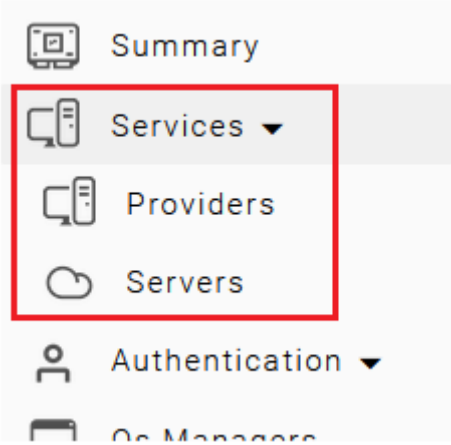
The graphs will allow us to enlarge dates for a better visualization or even download it, using the tools in the upper right corner.



NOTE: These graphs are indicative, for a more accurate data collection it is recommended to use the default reports available or other real-time graph generation tools such as: Grafana, Zabbix, etc…

3. SERVICES

The second element that we will have available in the administration of UDS will be "Services", which will allow us to register service providers and create groups of servers for some of them.

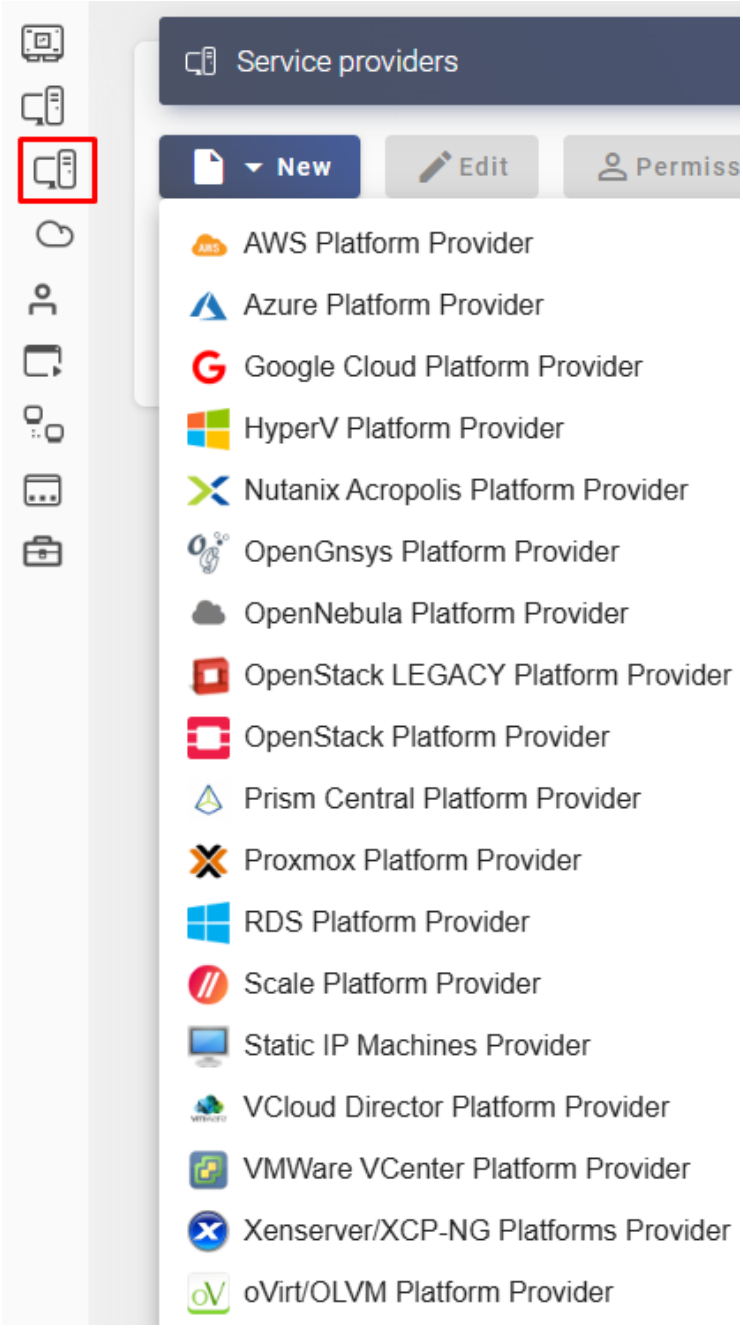


3.1 Service Providers

A "***Service Provider***" is the entity in charge of offering services such as: virtual desktops hosted on a virtualization platform, desktop sessions, on-demand virtual application sessions, and access to specific physical/virtual machines by assigning their IP address or name.

In order to build a "***Service Pool***" and be able to publish the different services supported by UDS, it is necessary to have created at least one "***Service Providers***", being supported the use of multiple "***Service Providers***" simultaneously.

Currently UDS supports the following "***Service Providers'***":



Safe Employment Procedure:

Download and install the Service Provider's certificate from the broker in order to enable SSL communications between them.

Many times a user with very special permissions is needed to interact with the service provider since they must be able to create, copy and delete services.

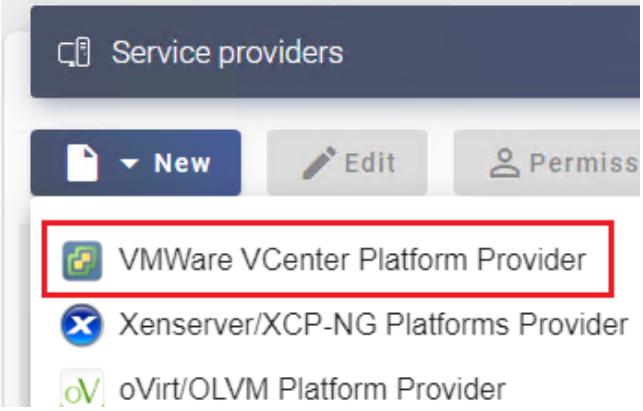
Do not use the "Administrator" user of the Service Provider within UDS. Create specific users (on the platforms that allow it, with the right administration permissions), so that it is easy to identify the tasks performed by it.

3.1.1 VDI with VMware vSphere

Deployment of self-generated virtual desktops by UDS or connection to existing machines, in a VMware vSphere virtualization infrastructure.

3.1.1.1 VMware vCenter Platform Provider Registration

Click on "***New***" and select "***VMware vCenter Platform Provider***":



In a "***VMware vCenter Platform Provider***" the parameters to be configured are:

- Main:

Name. Name to identify this provider

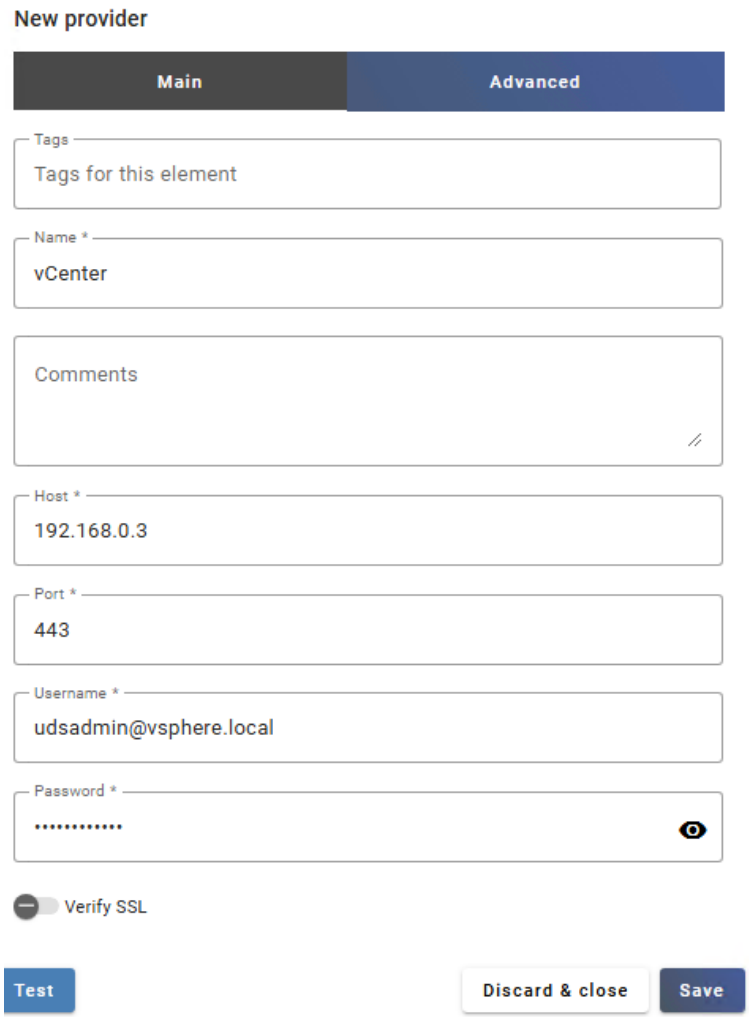
Host. Dirección IP o nombre del servidor vCenter

Port Communication Port with vCenter Server (default 443)

Username. Username with permissions for UDS integration with the provider. While it's a good idea to use a user with admin permissions (use a user created specifically for this purpose), you can also ask your support team for specific permissions to create a user with limited permissions.

Password. Password of the user used in the previous field

Verify SSL. If enabled, an SSL connection will be made to the vCenter server. It will be necessary to use the name of the server (in no case the IP address) and in addition the CA of the certificate used in the vCenter it will be necessary to import it to the UDS server so that it can be recognized by it.



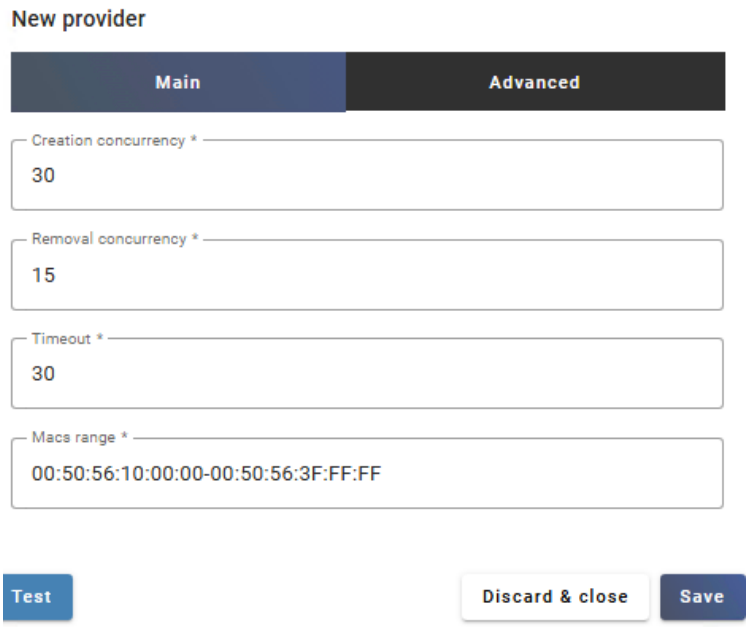
- Advanced:

Creation concurrency. Maximum number of concurrent desktop creation tasks, in the different service pools that use it, for this provider.

Removal concurrency. Maximum number of concurrent desktop deletion tasks, in the different service pools that use it, for this provider.

Timeout. The "***Timeout***" time on the connection to vCenter.

Macs range. MAC address range used by UDS to assign to automatically created virtual desktops. The modification of the range is always allowed to higher values, which will be applied immediately, but if it is reduced, it will be necessary that there is no record and it will also be necessary to wait a reasonable time of at least 24 hours for the records to be released and the change to be applied (to reduce the range, the registration of a new provider is recommended).



By clicking on the "*Test*" button, we will check that the connection is correct.

NOTE: To enable SSL verification with this provider, we will have to import the CA of the existing certificate in the vCenter, we can follow the following procedure:

We must download the vCenter certificate to our UDS Server, for example, with the wget command as follows:

```
wget --no-check-certificate https://nombre_servidor_vcenter/certs/download.zip
```

```
root@broker-400:/tmp# wget --no-check-certificate https://vcenter.vc.local/certs/download.zip
--2025-02-13 21:52:55--  https://vcenter.vc.local/certs/download.zip
Resolving vcenter.vc.local (vcenter.vc.local)... 192.168.0.3
Connecting to vcenter.vc.local (vcenter.vc.local)|192.168.0.3|:443... connected.
WARNING: The certificate of 'vcenter.vc.local' is not trusted.
WARNING: The certificate of 'vcenter.vc.local' doesn't have a known issuer.
HTTP request sent, awaiting response... 200 OK
Length: 10830 (11K) [zip]
Saving to: 'download.zip'

download.zip                        100%[=====================================================

2025-02-13 21:52:55 (255 MB/s) - 'download.zip' saved [10830/10830]

root@broker-400:/tmp#
```

Once downloaded, we extract its content:

```
root@broker-400:/tmp# unzip download.zip
Archive:  download.zip
  inflating: certs/lin/5ac25349.r0
  inflating: certs/mac/5ac25349.r0
  inflating: certs/win/5ac25349.r0.crl
  inflating: certs/lin/5ac25349.0
  inflating: certs/mac/5ac25349.0
  inflating: certs/win/5ac25349.0.crt
root@broker-400:/tmp#
```

The next step will be to copy the certificate for Linux computers (the one with the .0 extension) in the path: ***/usr/local/share/ca-certificates/*** and we will also change the extension to **.crt**

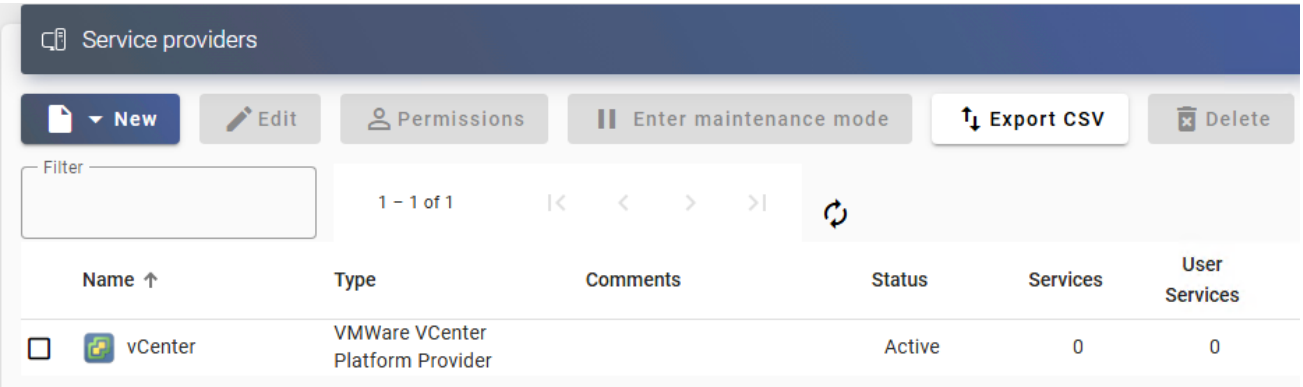
```
root@broker-400:/tmp# cp certs/lin/5ac25349.0 /usr/local/share/ca-certificates/vcenter.crt
root@broker-400:/tmp#
```

To finish the process, we will need to update the list of certificates and confirm that the new certificate has been added. We'll use the:
`update-ca-certificates`

```
root@broker-400:/tmp# update-ca-certificates
Updating certificates in /etc/ssl/certs...
rehash: warning: skipping duplicate certificate in TERENA_SSL_CA_3.pem
rehash: warning: skipping ca-certificates.crt,it does not contain exactl
1 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...
done.
root@broker-400:/tmp#
```

Once the process is finished, the certificate will be added and we will be able to activate the check for SSL validation (we must always use the DNS name of the vCenter server in the "host" section since the certificate is always associated with a name and never with an IP address)

Save the configuration and we will have a valid ***"Service Providers***" to start registering base services of the VMware vSphere platform.

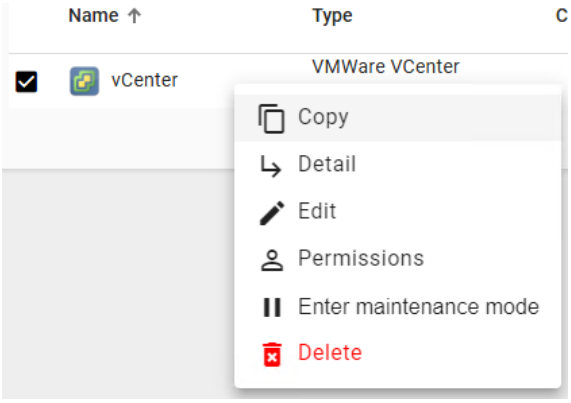


NOTE: If we have several VMware vSphere platforms managed by different vCenter servers, we will be able to register all the "Service Providers" of the "VMware vCenter Platform Provider" type that we need.

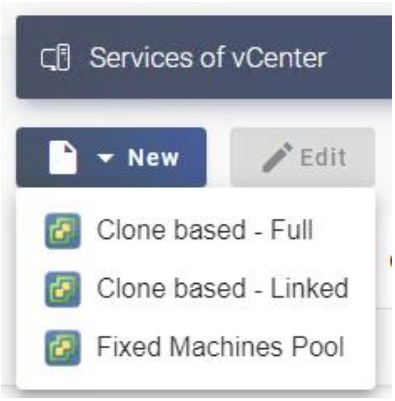
To modify a parameter in an existing ***"Service Providers***", select it and click on "***Edit***".

By clicking on the "***Enter Maintenance Mode***" button, you can pause all operations produced by the UDS server on a service provider. It is recommended to put a service provider in maintenance cases where communication with that service provider has been lost or a maintenance shutdown is planned.

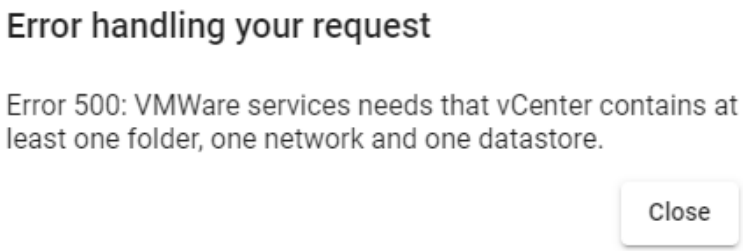
Once the VMware vSphere platform is integrated into UDS, different types of base services can be created. To do this, we double-click on the service provider created or in the provider's menu we select "***Detail***":



We will have the following types of basic services:



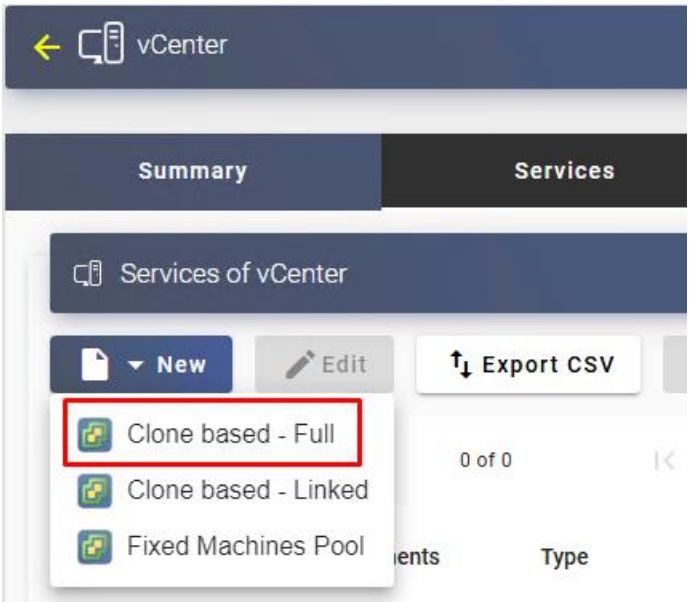
NOTE: To create the base services of type: "clone based – Full" and "clone based – Linked", we will need to have at least one network to connect virtual machines, a datastore where to store publications and desktops and a folder in the "VMs and Templates" view. If you do not have any of these elements, you will get the following error:



3.1.1.2 Configure "Clone based - Full service"”

This type of service will create, within the virtual infrastructure, complete virtual desktops independent of the active publication (its creation time and occupation on disk is much longer than those of the "Linked" type, its use is only recommended in very specific cases).

To create base services of type "***Clone based - Full***" we access the "***Service Providers***", select the "***Services***" tab, click on "***New***" and select "***Clone based - Full***”.



The minimum parameters to be configured in this type of service are:

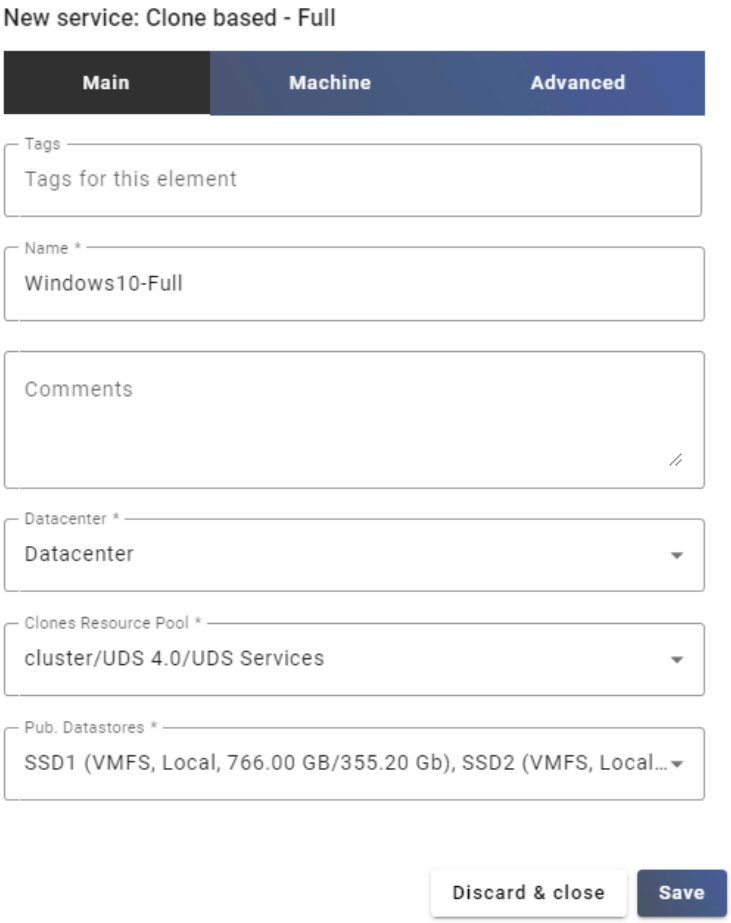
- Main:

Name: Name of the service.

Datacenter: Datacenter in the VMware vSphere environment where the service is hosted.

Resource Pool Clones: Pool of vCenter resources where publications and virtual desktops will be hosted (if there are no Pools in the VMware infrastructure, they will be created at the root).

Pub. Datastores: Location where the service publication will be stored.
You can select one, several or all of the available datastores. If you select several, the system will always locate the new publications in the datastore that has the freest space.



- Machine:

Resource Pool: Pool of vCenter resources where the template virtual machine (Gold Image) to be used by the service is located.

Base Machine: Template virtual machine (Gold Image) used by the UDS system for the deployment of virtual desktops.

Memory: Amount of memory in Mb that will be allocated to the generated full Clones virtual desktops.

Main Network: Communication network of the UDS actor with the UDS server. It will also define the network interface to be modified in the generated virtual desktops.

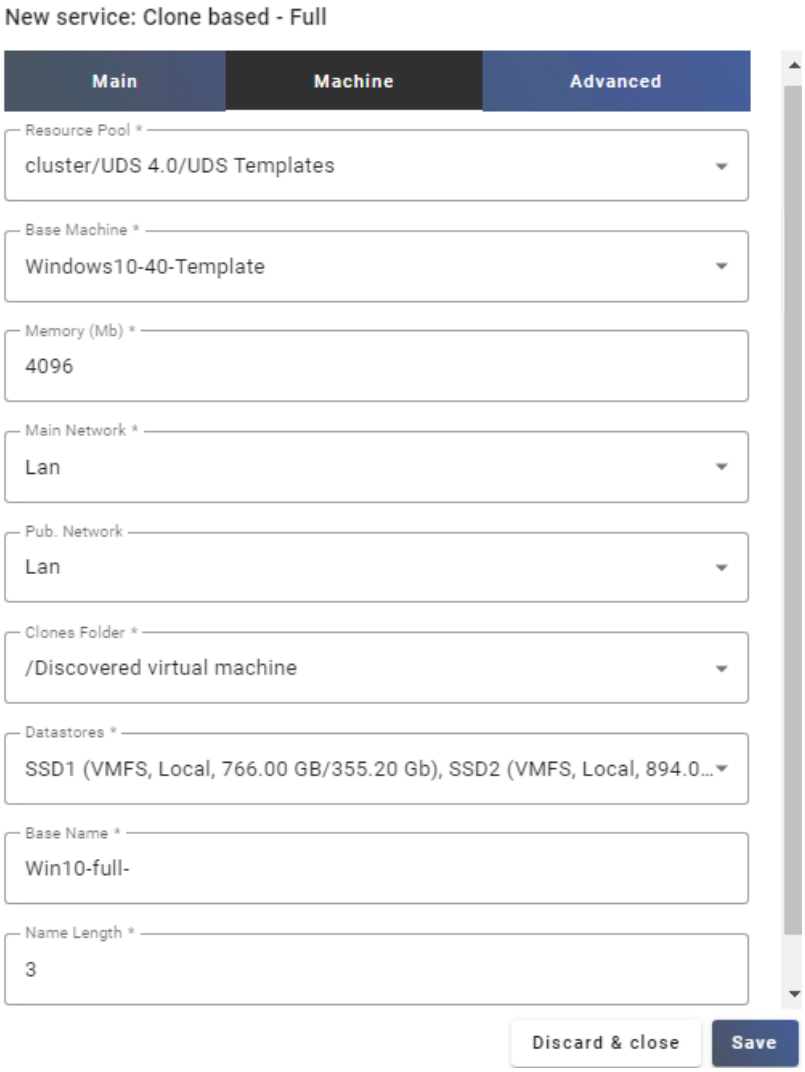
Pub. Network: Network where virtual desktops will be connected.

Clones Folder: Folder where new virtual desktops will be registered in the "VMs and Templates" view of vCenter.

Datastores: Location where the generated virtual desktops will be stored. We can select one, several or all datastores. If you select several, the system will always place the new virtual desktops in the datastore that has the freest space.

Base Name: Root of the name of all Full Clones virtual desktops to be deployed in this service. (e.g. Base Name= Win10-full-).

Name Length: Number of digits of the counter attached to the root of the desktop name (e.g.: Name Length= 3. The final name of the generated desktops would be: Win10-full-000**,** Win10-full-001... Win10-full-999).



- Advanced:

Reserved Space: When there is less free space in a datastore than indicated in this parameter (in GB), UDS will not use it to host publications or virtual desktops.

Remove found duplicates: If this option is enabled (default), when UDS finds a virtual desktop with the same name (generated by UDS, but not registered) as the desktop it is about to create, it will delete it (it is recommended to use it to avoid "lost" VMs in the vCenter inventory).

Keep on error: When activated, the system ignores all errors in accessing the service, thus preventing desktop deletion. Very useful and recommended for use when using persistent machines, evitando que bajo prácticamente ninguna circunstancia el escritorio sea eliminado por el sistema (excepto si un administrador lo elimina manualmente).

Try SOFT Shutdown first: When activated, the system will attempt to perform a "Soft" shutdown on the OS of the virtual desktop. It is necessary that the VMware tools are installed and running on the machine to be deleted, otherwise, after a period of time a forced shutdown will be performed.

Put back to cache: Parameter that defines whether machines will be returned to the cache, so that they can be used by another user, once they are freed:

- **No. Never put it back to cache:** When a user logs out of the desktop and the machine is non-persistent, it will be destroyed and will NOT be returned to the cache to be reused by more users.

- **Yes, try to put it back to cache:** When a user logs out of the desktop and the machine is non-persistent, it will be returned IF it will be returned to the cache to be reused by more users.

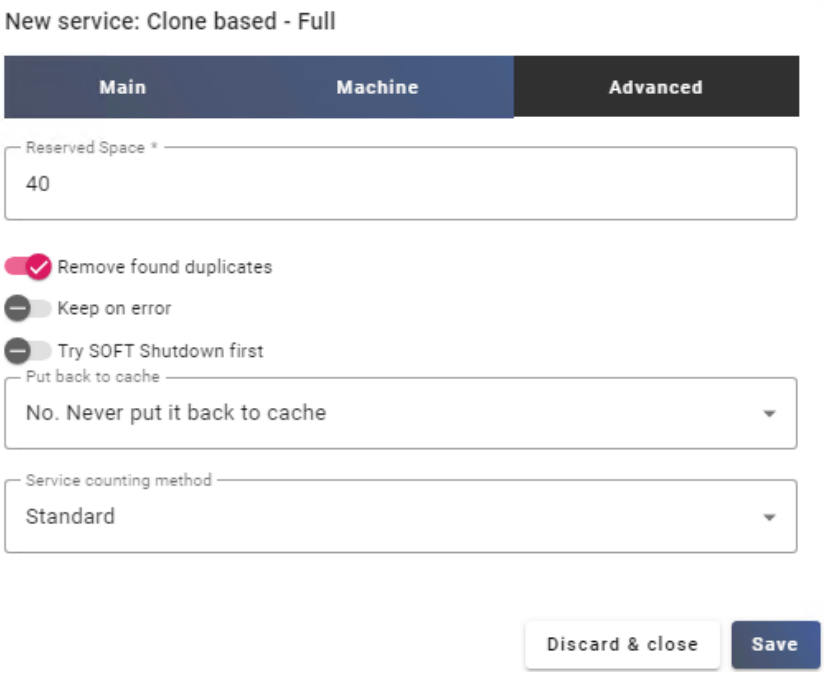
Service counting method: Method that the system will use to count machines to comply with the parameter "maximum number of services to be provided" of a Service Pool:

- **Standard:** This counting mode only takes into account the machines that are being created and during the process of creating and destroying machines, it is possible to exceed the value of the parameter "maximum number of services to be provided" of a Service Pool.

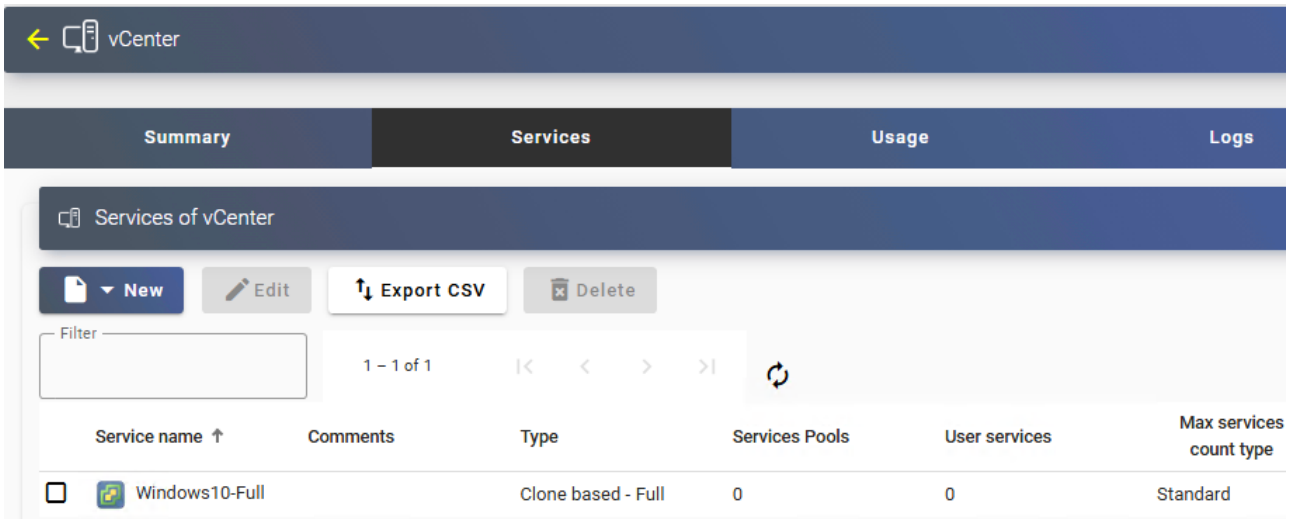
  For example, if a service pool, whose "maximum number of services to be provided" parameter is set to 5, we delete a service, there will be a time when that pool contains 6 machines: 4 configured, 1 being created and 1 being deleted.

- **Conservative:** This counting mode takes into account the machines that are being created and those that are being deleted, at no time will it be possible to exceed the value of the parameter "maximum number of services to be provided" of a Service Pool.

  For example, if a service pool, whose parameter of "maximum number of services to be provided" is set to 5, we delete a service, until that service is completely eliminated the system will not generate a new service, avoiding exceeding the value of 5 existing machines in the pool.



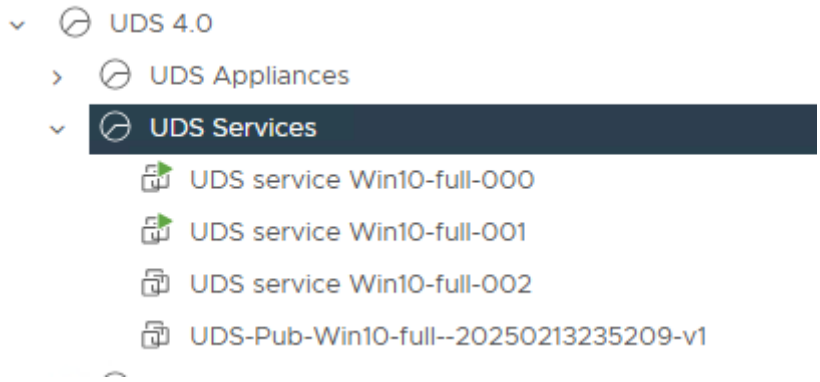
Save the configuration and we will have a valid "***Clone based - Full***" on the VMware vCenter platform. We will be able to register all the "***Clone based - Full***" services that we need.



Once we have the entire UDS environment configured (Services, Authenticators, OS Managers and Transports) and created the first "***Service Pool***", we will be able to observe on the vCenter server how the desktop deployments are carried out.

The first task that vCenter will perform will be to generate a publication of the template (this machine will be generated every time we publish a service). It will be a clone of the template chosen when registering for the service, with a disk size and characteristics equal to said template.

Once the process of creating the publication is finished (the system will name it as: "UDS-Pub-Base_Name-date-vnúmero_publicación") the virtual desktops are automatically created in vCenter, (the system names them as: "UDS service *Base_Name+Name_Length*") based on the cache parameters configured in the "***Service Pools'***".

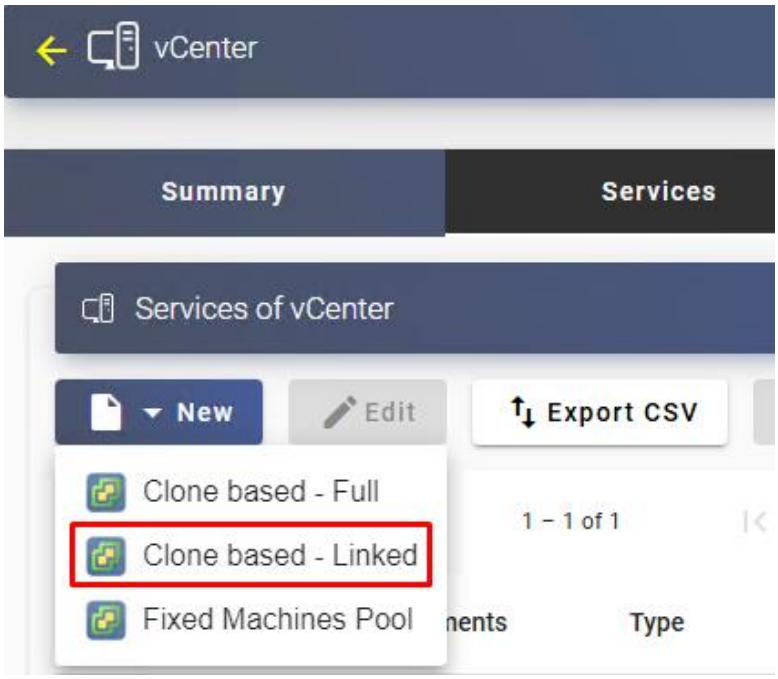


NOTE: The disk space that the virtual desktops ("**Full Clones**") will occupy will be the same as that of the template machine.

3.1.1.3 Configure service based on "Clone based - Linked"

This type of service will create, within the virtual infrastructure, virtual desktops in linked clone mode (differentials) dependent on active publishing (their creation time and occupation on disk is much less than those of the "Full" type, their use is recommended in most use cases).

To create base services of type "***Clone based - Linked***" we access the "***Service Providers***", select the "***Services***" tab, click on "***New***" and select "***Clone based - Linked***".



The minimum parameters to be configured in this type of service are:

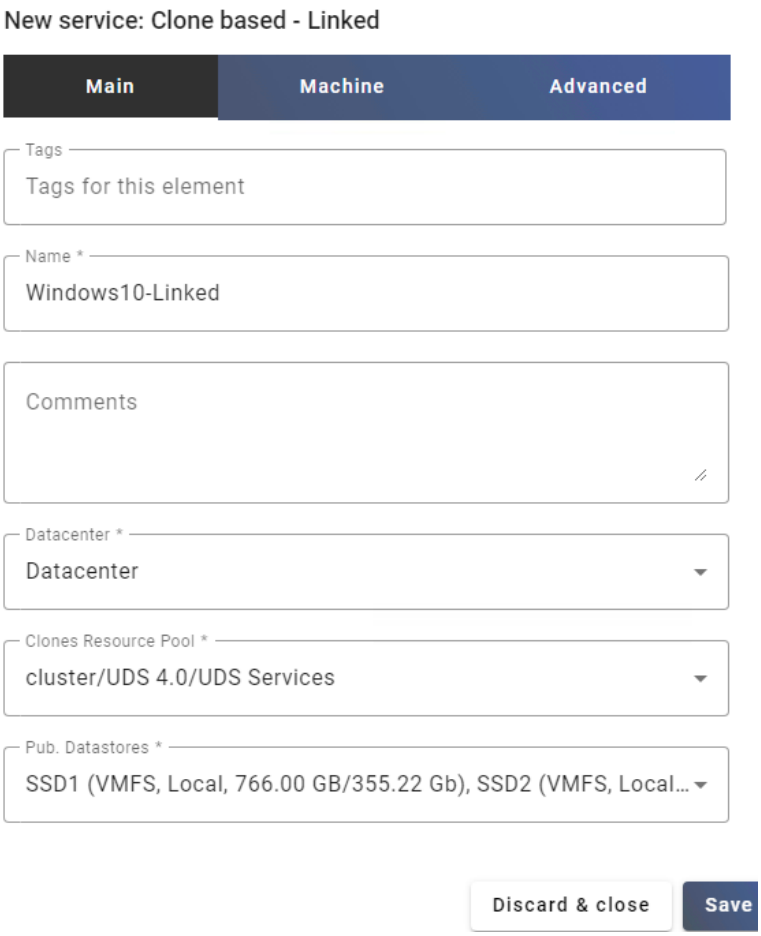
- Main:

Name: Name of the service.

Datacenter: Datacenter in the VMware vSphere environment where the service is hosted.

Resource Pool Clones: Pool of vCenter resources where publications and virtual desktops will be hosted (if there are no Pools in the VMware infrastructure, they will be created at the root).

Pub. Datastores: Location where the service publication will be stored.
We can select one, several or all of the available datastores. If you select several, the system will always locate the new publications in the datastore that has the most free space.



- Machine:

Resource Pool: Pool of vCenter resources where the template virtual machine (Gold Image) to be used by the service is located.

Base Machine: Template virtual machine (Gold Image) used by the UDS system for the deployment of virtual desktops.

Memory: Amount of memory in Mb that will be allocated to the generated Linked Clone virtual desktops.

Main Network: Communication network of the UDS actor with the UDS server. It will also define the network interface to be modified in the generated virtual desktops.

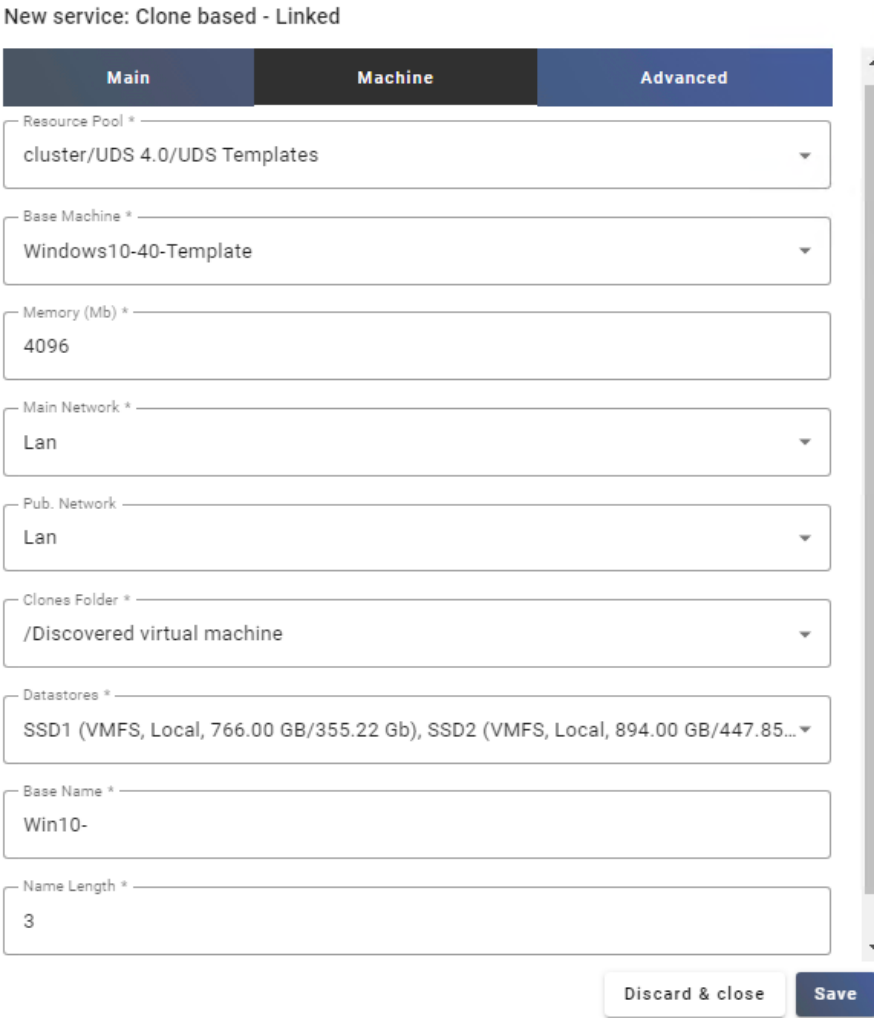
Pub. Network: Network where virtual desktops will be connected.

Clones Folder: Folder where new virtual desktops will be registered in the "VMs and Templates" view of vCenter.

Datastores: Location where the generated virtual desktops will be stored. We can select one, several or all datastores. If you select several, the system will always place the new virtual desktops in the datastore that has the most free space.

Base Name: Root of the name of all Linked Clone virtual desktops to be deployed in this service. (ex: Base Name= Win10-).

Name Length: Number of digits of the counter attached to the root of the desktop name (e.g.: Name Length= 3. The final name of the generated desktops would be: Win10-000**,** Win10-001... Win10-999).



- Advanced:

Reserved Space: When there is less free space in a datastore than indicated in this parameter (in GB), UDS will not use it to host publications or virtual desktops.

Remove found duplicates: If this option is enabled (default), when UDS finds a virtual desktop with the same name (generated by UDS, but not registered) as the desktop it is about to create, it will delete it (it is recommended to use it to avoid "lost" VMs in the vCenter inventory).

Keep on error: When activated, the system ignores all errors in accessing the service, thus preventing desktop deletion. Very useful and recommended for use when using persistent machines, preventing the desktop from being deleted by the system under virtually any circumstances (except if an administrator manually deletes it).

Try SOFT Shutdown first: When activated, the system will attempt to perform a "Soft" shutdown on the OS of the virtual desktop. It is necessary that the VMware tools are installed and running on the machine to be deleted, otherwise, after a period of time a forced shutdown will be performed.

Put back to cache: A parameter that defines whether machines will be returned to the cache, so that they can be used by another user, once they are released:

- **No. Never put it back to cache:** When a user logs out of the desktop and the machine is non-persistent, it will be destroyed and will NOT be returned to the cache to be reused by more users.

- **Yes, try to put it back to cache:** When a user logs out of the desktop and the machine is non-persistent, it will be returned IF it will be returned to the cache to be reused by more users.

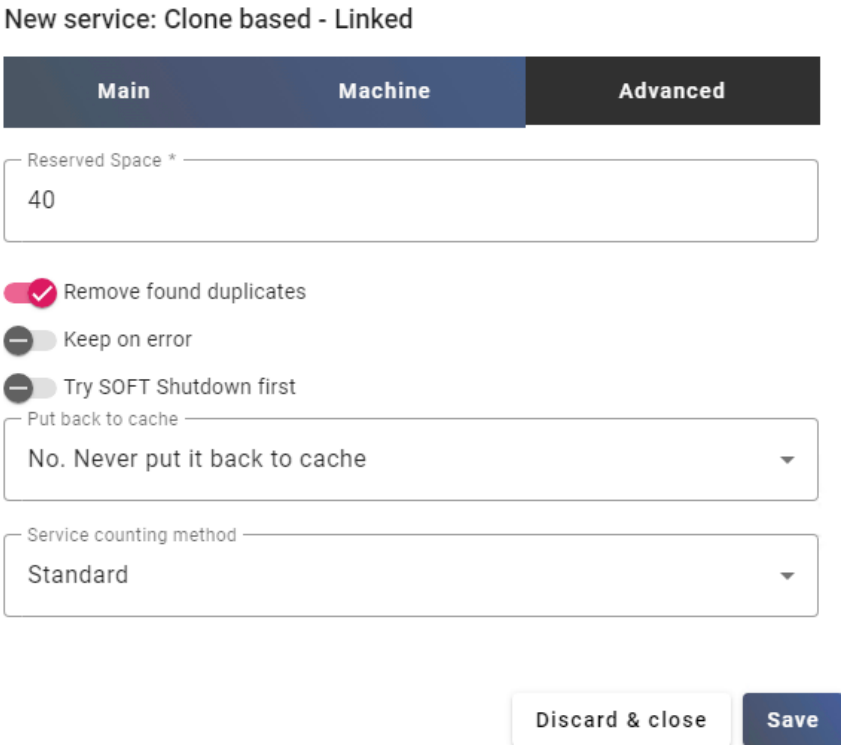
Service counting method: Method that the system will use to perform the machine count to comply with the parameter "maximum number of services to be provided" of a Service Pool:

- **Standard:** This counting mode only takes into account the machines that are being created and during the process of creating and destroying machines, it is possible to exceed the value of the parameter "maximum number of services to be provided" of a Service Pool.

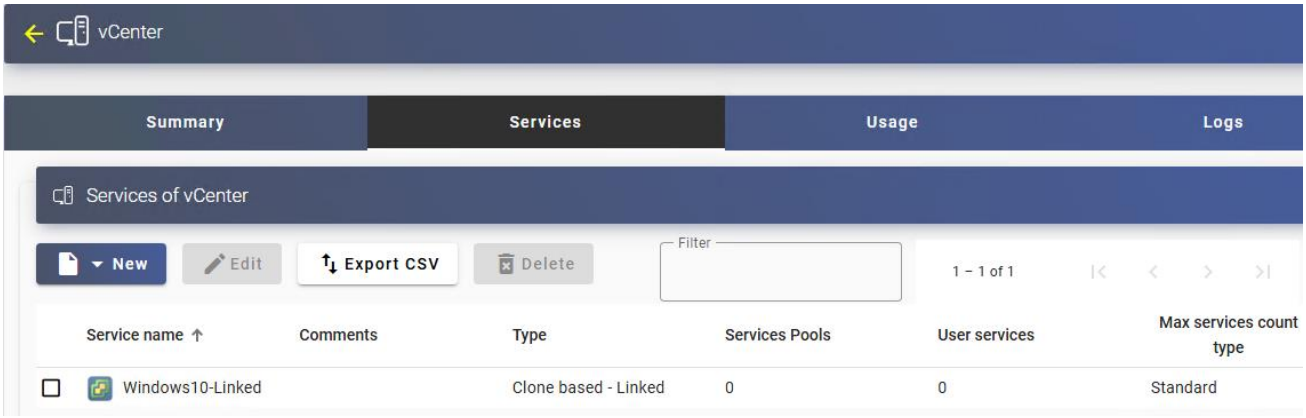
  For example, if a service pool, whose "maximum number of services to be provided" parameter is set to 5, we delete a service, there will be a time when that pool contains 6 machines: 4 configured, 1 being created and 1 being deleted.

- **Conservative:** This counting mode takes into account the machines that are being created and those that are being deleted, at no time will it be possible to exceed the value of the parameter "maximum number of services to be provided" of a Service Pool.

  For example, if a service pool, whose parameter of "maximum number of services to be provided" is set to 5, we delete a service, until that service is completely eliminated the system will not generate a new service, avoiding exceeding the value of 5 existing machines in the pool.



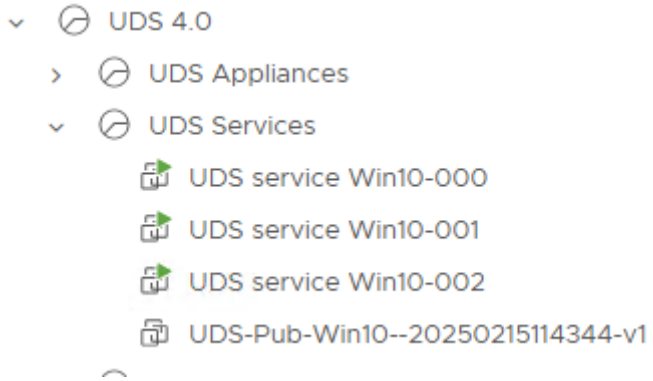
Save the configuration and we will have a valid "***Clone based - Linked***" on the VMware vCenter platform. We will be able to register all the "***Clone based - Linked***" services that we need.



Once we have the entire UDS environment configured (Services, Authenticators, OS Managers and Transports) and created the first "***Service Pool***", we will be able to observe on the vCenter server how the desktop deployments are carried out.

The first task that vCenter will perform will be to generate a publication of the template (this machine will be generated every time we publish a service). It will be a clone of the template chosen when registering for the service, with a disk size and characteristics equal to said template.

Once the process of creating the publication is finished (the system will name it as: "UDS-Pub-Base_Name-date-vnúmero_publicación") the virtual desktops are automatically created in vCenter (the system names them as: "UDS service *Base_Name*+*Name_Length*") based on the cache parameters configured in the "***Service Pools***".

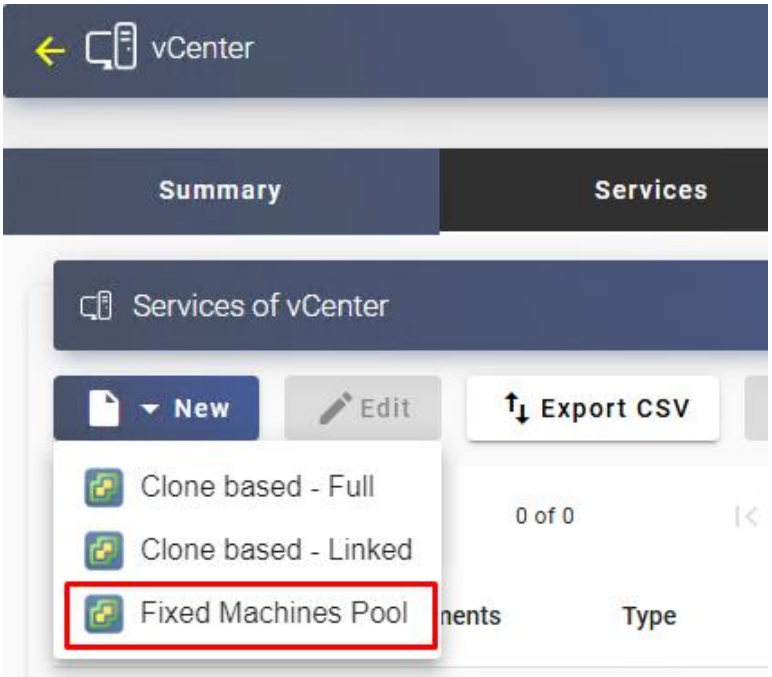


NOTE: The disk space that virtual desktops will occupy ("Linked Clones") will be exclusively that occupied by the modifications made by users on the (incremental) machines after their initial connection.

3.1.1.4 Configure service based on "Fixed Machines Pool""

This type of service will allow users to connect directly to existing virtual machines in a VMware vSphere environment pool.

To create base services of type "***Fixed Machines Pool***" we access the "***Service Providers***", select the "***Services***" tab, click on "***New***" and select "***Fixed Machines Pool***".



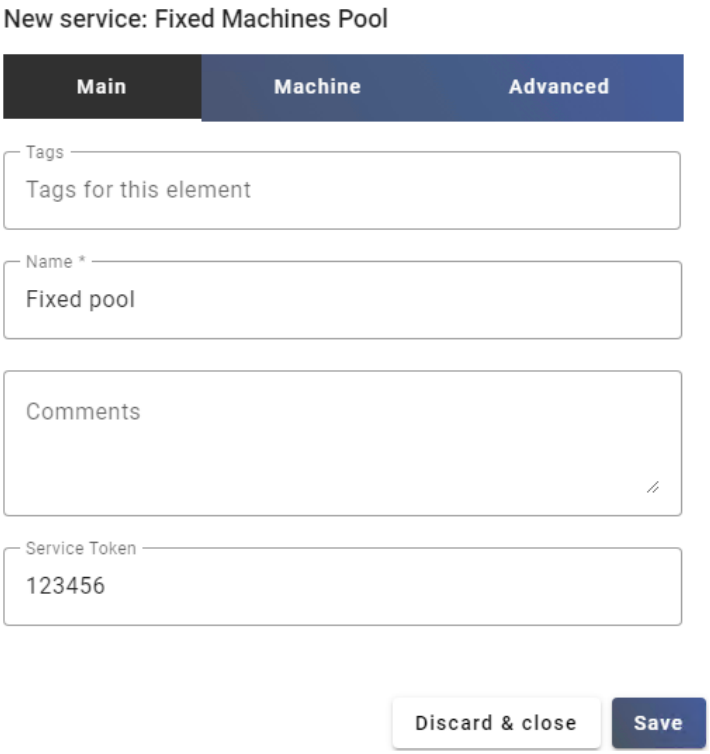
The minimum parameters to be configured in this type of service are:

- Main:

Name: Name of the service.

Service Token: If UDS is required to recognize the login and ***logout*** of a user accessing a service of this type (so that when a ***logout*** is made, UDS automatically releases the machine and makes it available to another user), it will be necessary to fill in this field with alphanumeric characters. If the field is left empty, UDS will permanently assign (until an administrator manually releases the service) the machine to a user, as it will have no way to control when a user enters or exits the machine.

NOTE: If the "**Service Token**" field is filled in, on the machines that are selected for the connection it will be necessary to have the UDS "**UDSActorUnmanagedSetup-x.x.x.exe**" actor installed and in the configuration of this, indicate those same characters in the "**Service Token" field**".



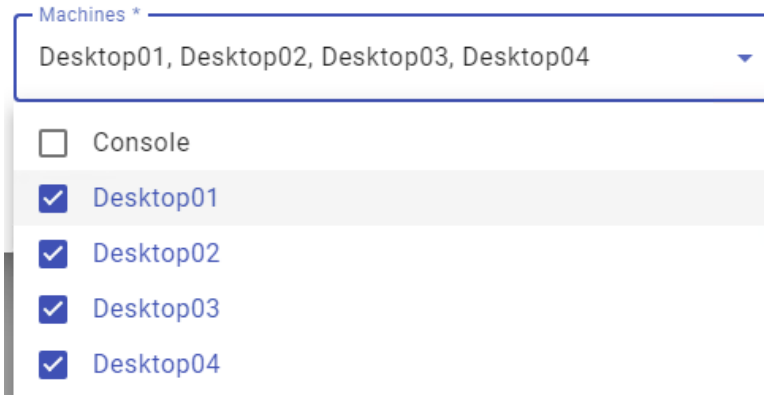
- Machine:

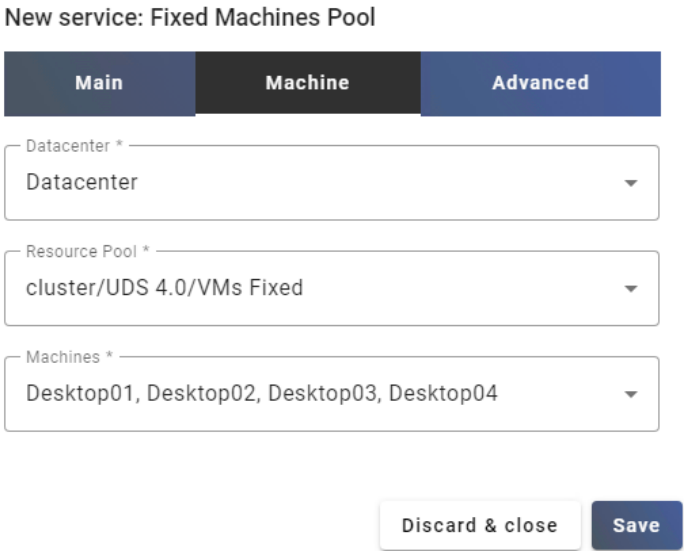
Datacenter: Datacenter of the vSphere environment where the virtual machines that we want to manage and assign to users are located.

Resource Pool: Pool of resources in the vSphere environment where the virtual machines that we want to manage and assign to users are located.

Machines: Virtual machines existing in the pool of resources indicated above that will be managed by UDS and to which access can be provided to the different users. We will have to indicate which machines will be accessible by users, the machines not selected, will be ignored and will not be assigned.

If the machines have the unamanaged actor installed and a "token" has been configured both in the actor (machine) and in the service definition (administration), it will be possible to control the access of users to these machines and they can be released so that other users can use them, in case they do NOT have the unmanaged actor installed, When a machine is assigned to a user, that assignment will be maintained until manually deleted by an administrator.





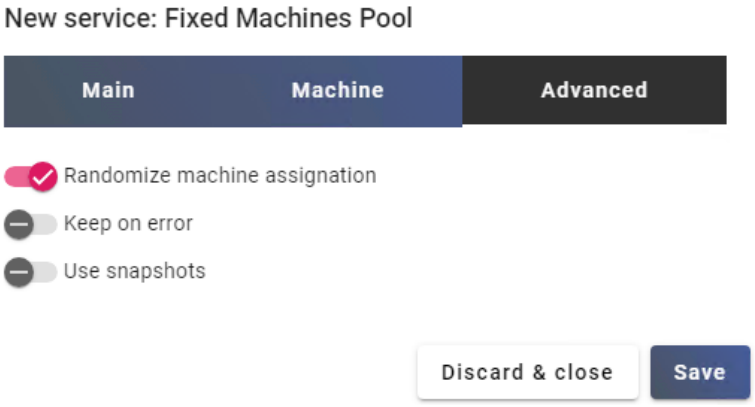
- Advanced

Randomize machine assignment: This parameter indicates how the order of assignments to the users will be, if it is active (by default) the order of assignment will be random, that is, when a user requests a machine, it will be unknown which one will be assigned. If the parameter is not active, the assignment order will be sequential, the first user will be given the first machine indicated in the list, the second the second machine and so on.

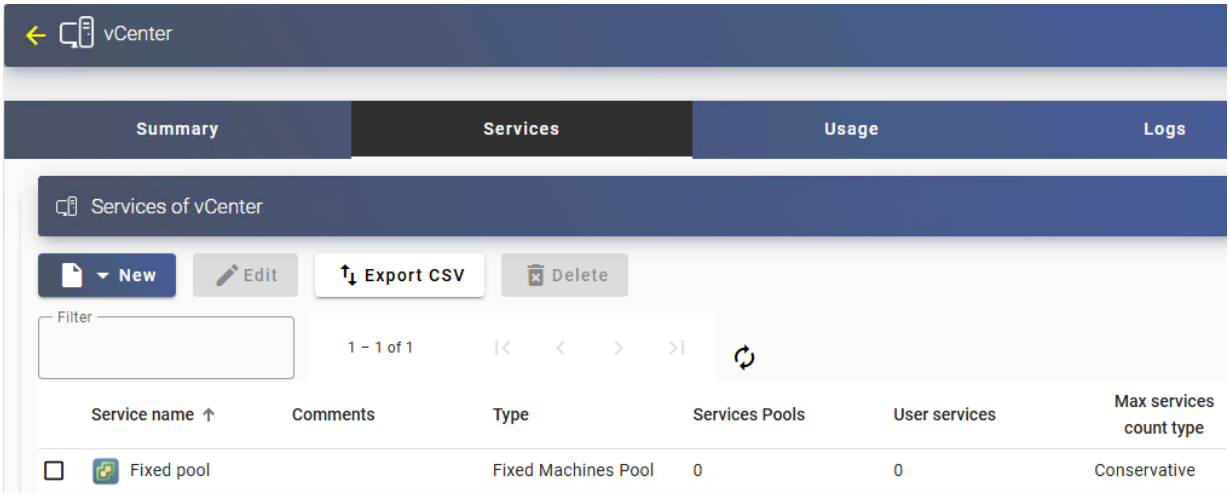
Keep on error: When activated, the system ignores all service access errors, allowing the machine to remain available for access (useful for occasions when machines may not be available for a short period of time but should be available at some point).

Use Snapshots: If this parameter is enabled, UDS will create a snapshot on the virtual machine before the user's connection and once its assignment is removed (through an automatic detection of the user's logout or manual deletion of an administrator) the snapshot will be recovered to leave the machine in the same state just before the user's connection. If this parameter is not activated, UDS will not recover any previous state of the machine when the assignment with a user is deleted.

NOTE: If the **"Use Snapshots" option is enabled** and the virtual machine already has a snapshot, UDS will not take any additional snapshots and will use and recover the existing snapshot.



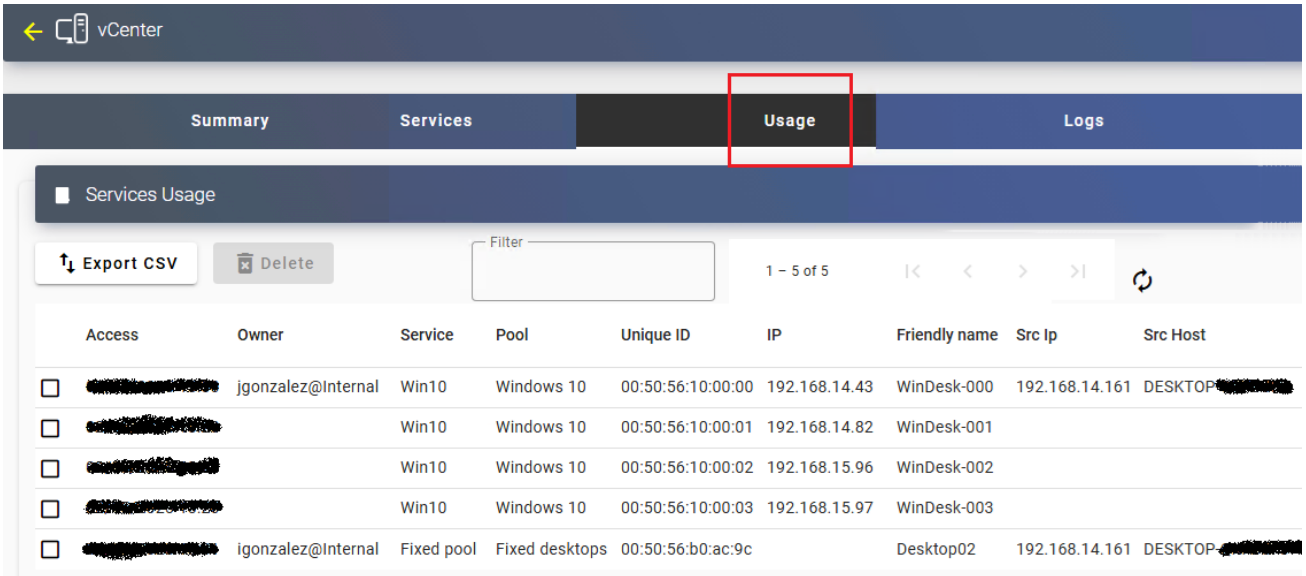
Save the configuration and we will have a valid "***Fixed Machines Pool***" on the VMware vCenter platform. We will be able to register all the "***Fixed Machines Pool***" services that we need.


vCenter
Summary
Services
Usage
Logs
Services of vCenter
New
Edit
Export CSV
Delete
Filter
1 – 1 of 1
Service name
Comments
Type
Services Pools
User services
Max services count type
Fixed pool
Fixed Machines Pool
0
0
Conservative

3.1.1.5 Service usage summary and error logs

Within a "***Service Provider***" we will have a tab called "***Usage***" that will allow us to have a quick view with detailed information and perform basic actions of the services deployed in this "***Service Provider***".

To access this information on the use of the services, we access the "***Service Providers***" and select the "***Usage***" tab:



We will be able to visualize:

Access: Date and time the service was accessed.

Owner: User and authenticator assigned to the service in usuario@autenticador format. If it appears empty, it means that the service has not been assigned to any user and is available in the cache.

Service: Name of the base service belonging to the "***Service Provider***" from which the service was generated.

Pool: Name of the "***Service Pools***" to which the service belongs.

Unique ID: MAC address of the service.

IP: IP address of the service.

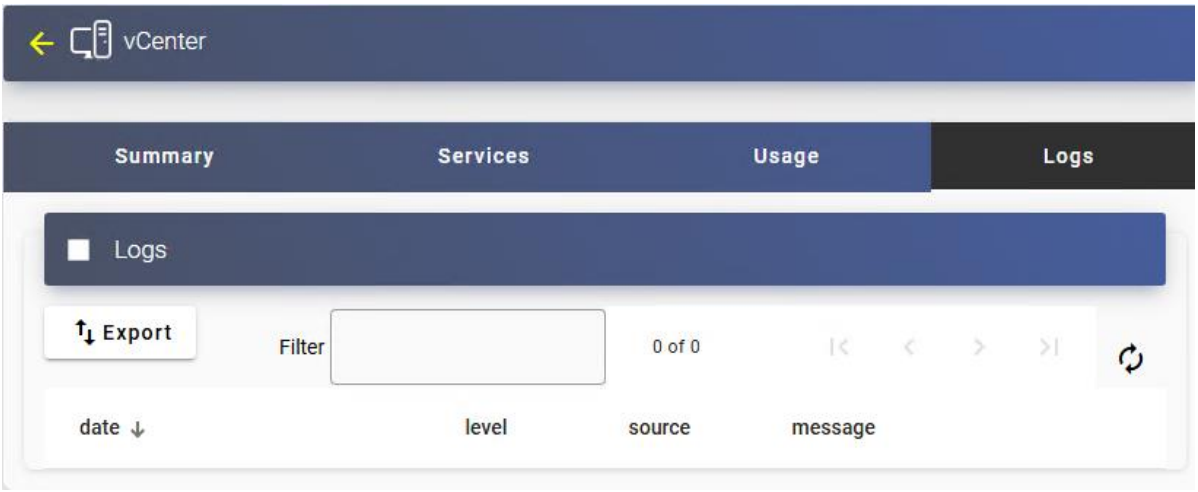
Friendly name: Name of the service. In the base services "***Clone based – Full***" and "***Clone base – Linked***" will also be the DNS name of the virtual machine autogenerated by UDS.

Src Ip: IP address of the connecting client accessing the service.

Src Host: Nombre de DNS del cliente de conexión que accede al servicio. En caso de no poder acceder a dicho nombre, se indicará su dirección IP.

It will also be possible to select one or more services to proceed with their deletion (in the base service of type "***Fixed Machine Pool***" we will only delete the assignment with the user).

The "***Logs***" tab will show us information about a possible problem that occurred in the "***Service Provider'***":

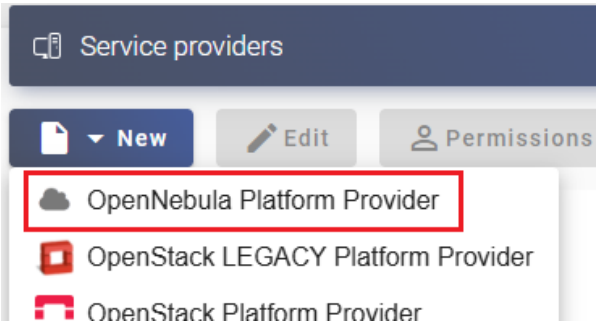


3.1.2 VDI with OpenNebula

Deployment of self-generated virtual desktops by UDS, in an OpenNebula virtualization infrastructure

3.1.2.1 Registration of Service Provider "OpenNebula Platform Provider"”

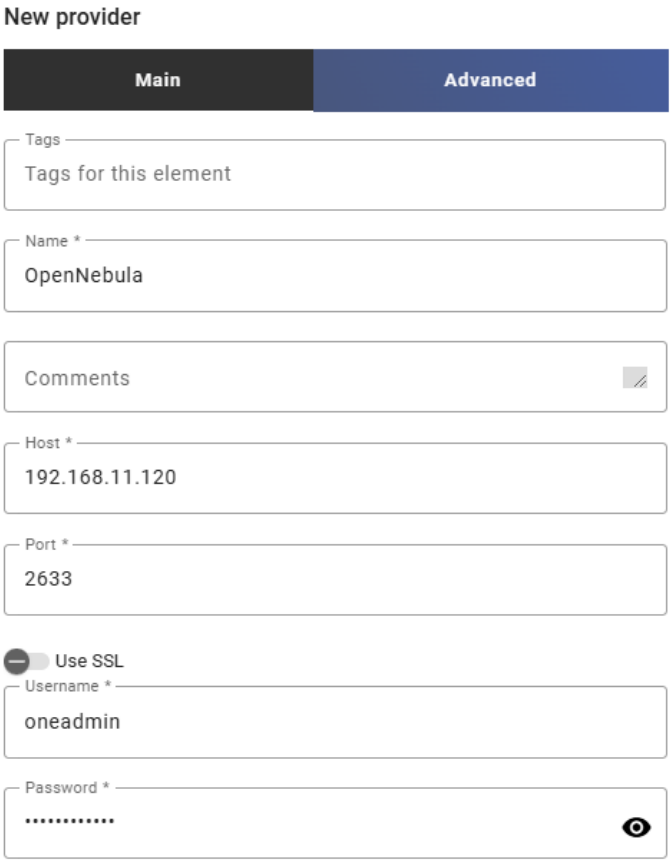
Click on "***New***" and select "***OpenNebula Platform Provider***”.



In an "***OpenNebula Platform Provider***" the parameters to be configured are:

- Main:

Service provider name, OpenNebula IP or server name, connection port, username and password with administration rights.

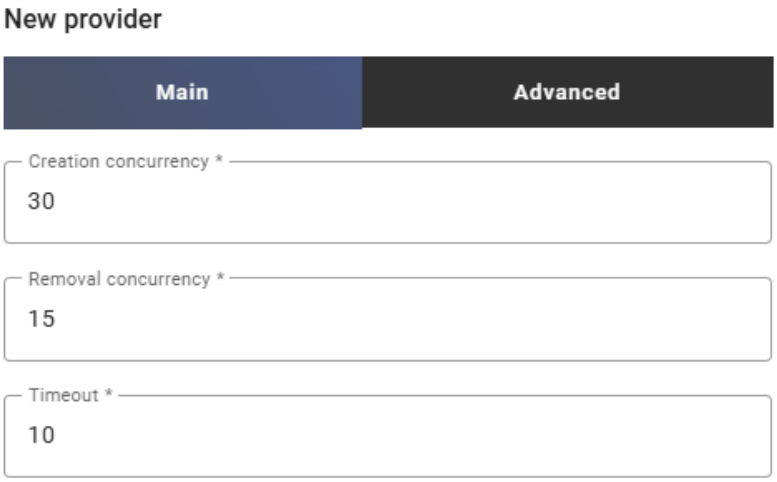


- Advanced:

Creation concurrency. Maximum number of concurrent desktop creation tasks, in the different service pools that use it, for this provider.

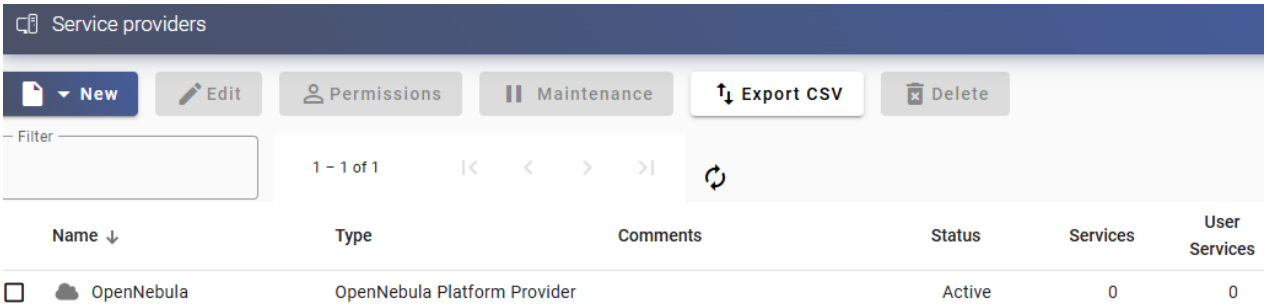
Removal concurrency. Maximum number of concurrent desktop deletion tasks, in the different service pools that use it, for this provider.

Timeout. The "***Timeout***" time on the connection



Using the "***Test***" button we will check that the connection is successful.

Save the configuration and we will have a valid "***Service Providers***" to start registering basic services on the OpenNebula platform.

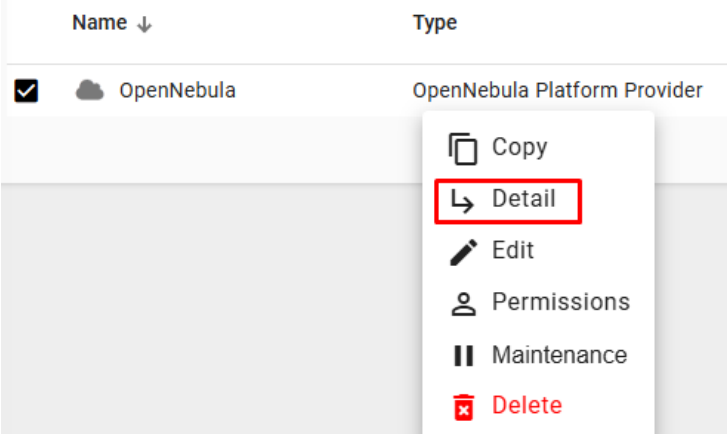


NOTE: If we have several OpenNebula platforms, we will be able to register all the "Service Providers" of the "OpenNebula Platform Provider" type that we need.

To modify a parameter in an existing ***"Service Providers***", select it and click on "***Edit***".

By clicking on the "***Enter Maintenance Mode***" button, you can pause all operations produced by the UDS server on a service provider. It is recommended to put a service provider in maintenance cases where communication with that service provider has been lost or a maintenance shutdown is planned.

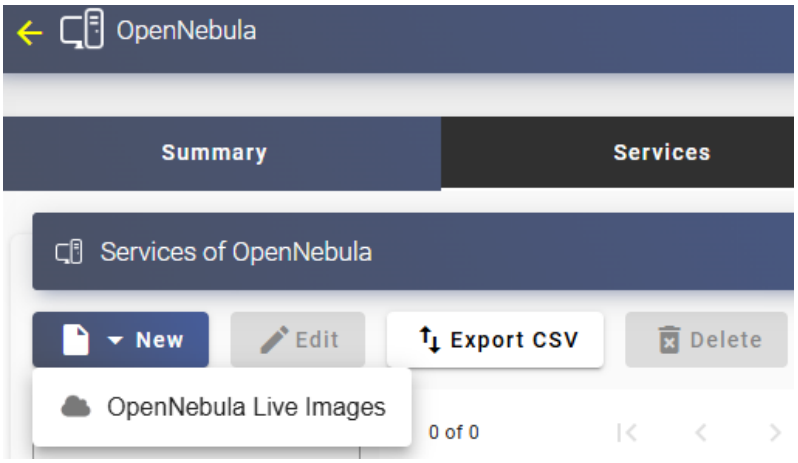
Once the OpenNebula platform is integrated into UDS, base services can be created. To do this, we double-click on the service provider created or in the provider's menu we select "***Detail***":


Name
Type
OpenNebula
OpenNebula Platform Provider
Copy
Detail
Edit
Permissions
Maintenance
Delete

3.1.2.2 Configure service based on "OpenNebula Live Images"”

This type of service will create, within the virtual infrastructure, virtual desktops dependent on active publishing.

To create base services of type "***OpenNebula Live Images***" we access the "***Service Providers***", select the "***Services***" tab, click on "***New***" and select "***OpenNebula Live Images***”.

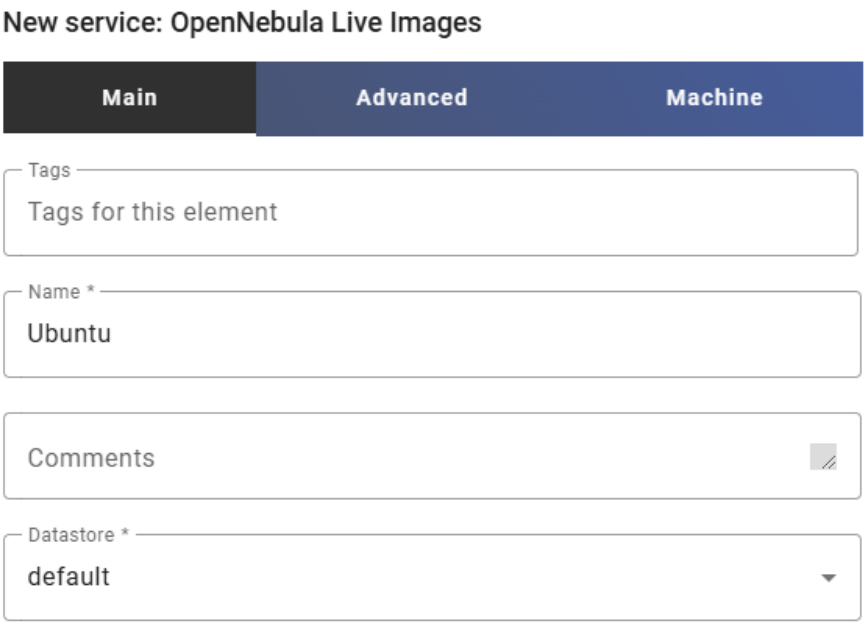


The minimum parameters to be configured in this type of service are:

- Main:

Name: Name of the service.

Datastore: Location where generated publications and virtual desktops will be stored.



- Advanced:

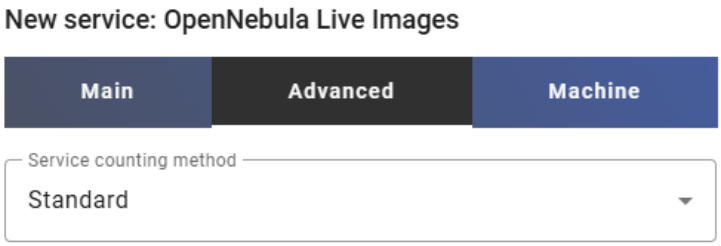
Service counting method: Method that the system will use to perform the machine count to comply with the parameter "maximum number of services to be provided" of a Service Pool:

- **Standard:** This counting mode only takes into account the machines that are being created and during the process of creating and destroying machines, it is possible to exceed the value of the parameter "maximum number of services to be provided" of a Service Pool.

  For example, if a service pool, whose "maximum number of services to be provided" parameter is set to 5, we delete a service, there will be a time when that pool contains 6 machines: 4 configured, 1 being created and 1 being deleted.

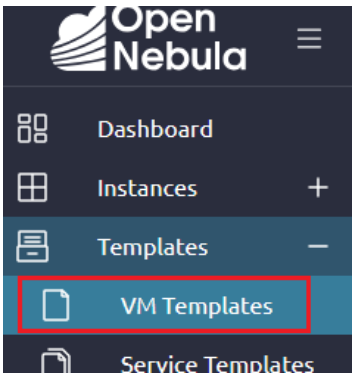
- **Conservative:** Este modo de conteo tiene en cuenta las máquinas que se están creando y las que se están eliminando, en ningún momento será posible superar el valor del parámetro “número máximo de servicios a proveer” de un Pool de servicios.

  Por ejemplo, si un pool de servicios, cuyo parámetro de “número máximo de servicios a proveer” está configurado a 5, borramos un servicio, hasta que dicho servicio no está eliminado completamente el sistema no generará un nuevo servicio, evitando superar el valor de 5 máquinas existentes en el pool.



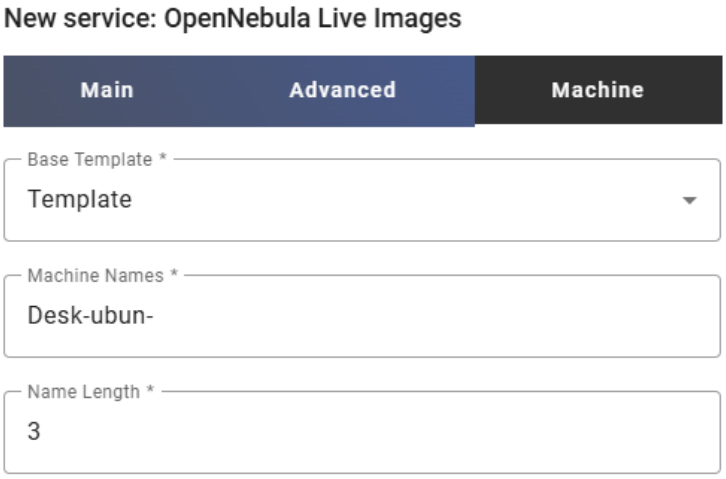
- Machine:

Base Template: Existing template in the OpenNebula environment used by the UDS system for the deployment of virtual desktops.

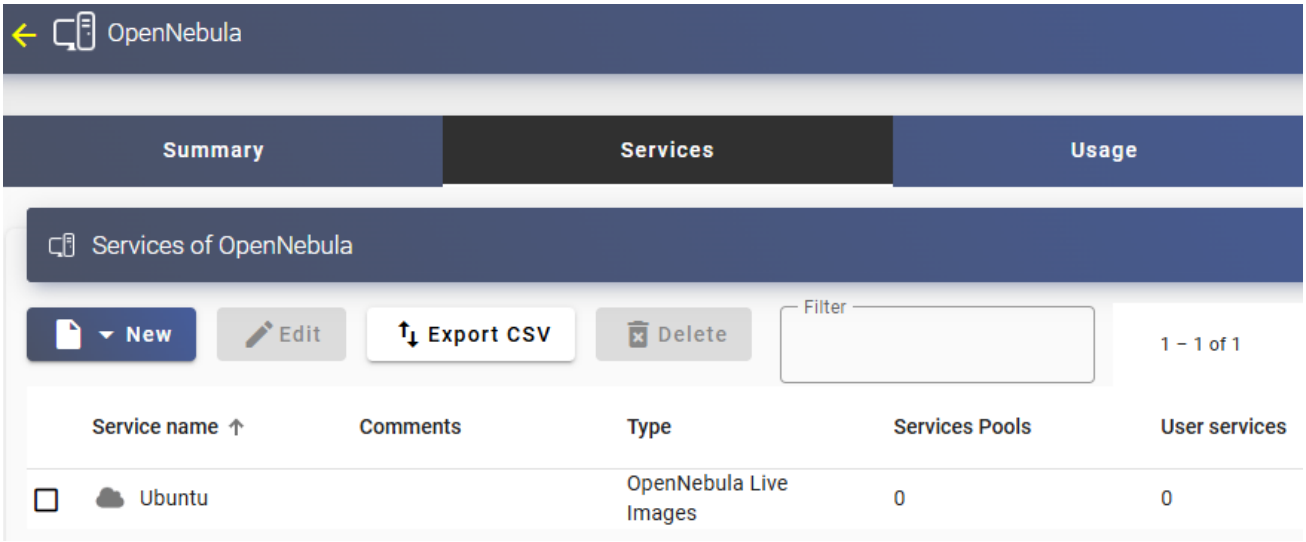


Machine Names: Root of the name of all virtual desktops to be deployed in this service. (e.g.: Machine Names= Desk-ubun-).

Name Length: Number of digits of the counter attached to the root of the desktop name (e.g.: Name Length= 3. The final name of the generated desktops would be: Desk-ubun-000**,** Desk-ubun-001... Desk-ubun-999).



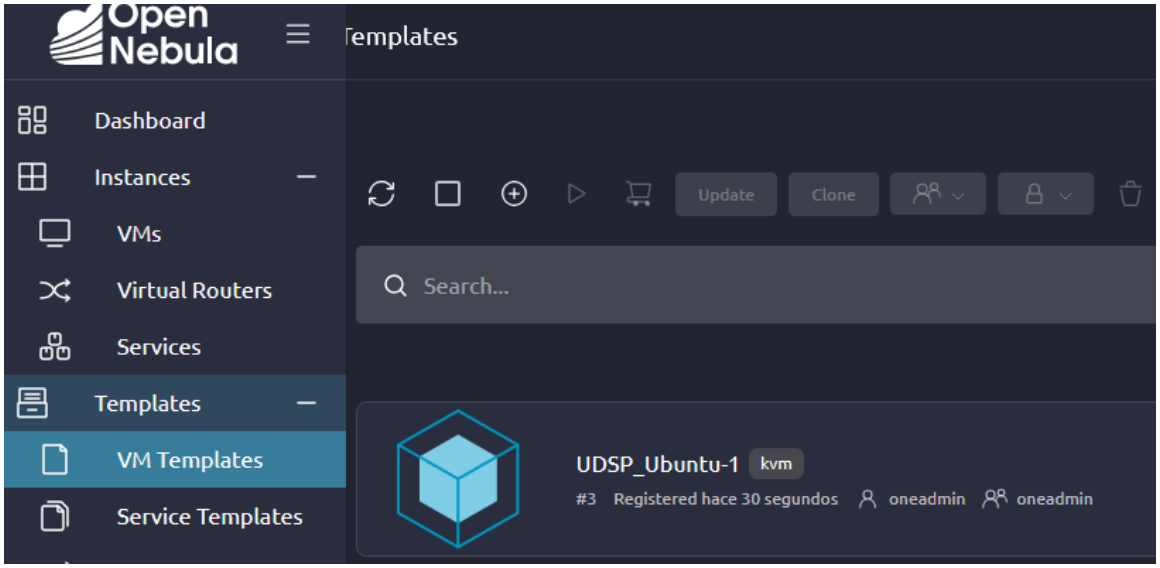
Save the configuration and we will have an "***OpenNebula Live Images***" valid on the OpenNebula platform. We will be able to register all the services of the type "***OpenNebula Live Images***" that we need.

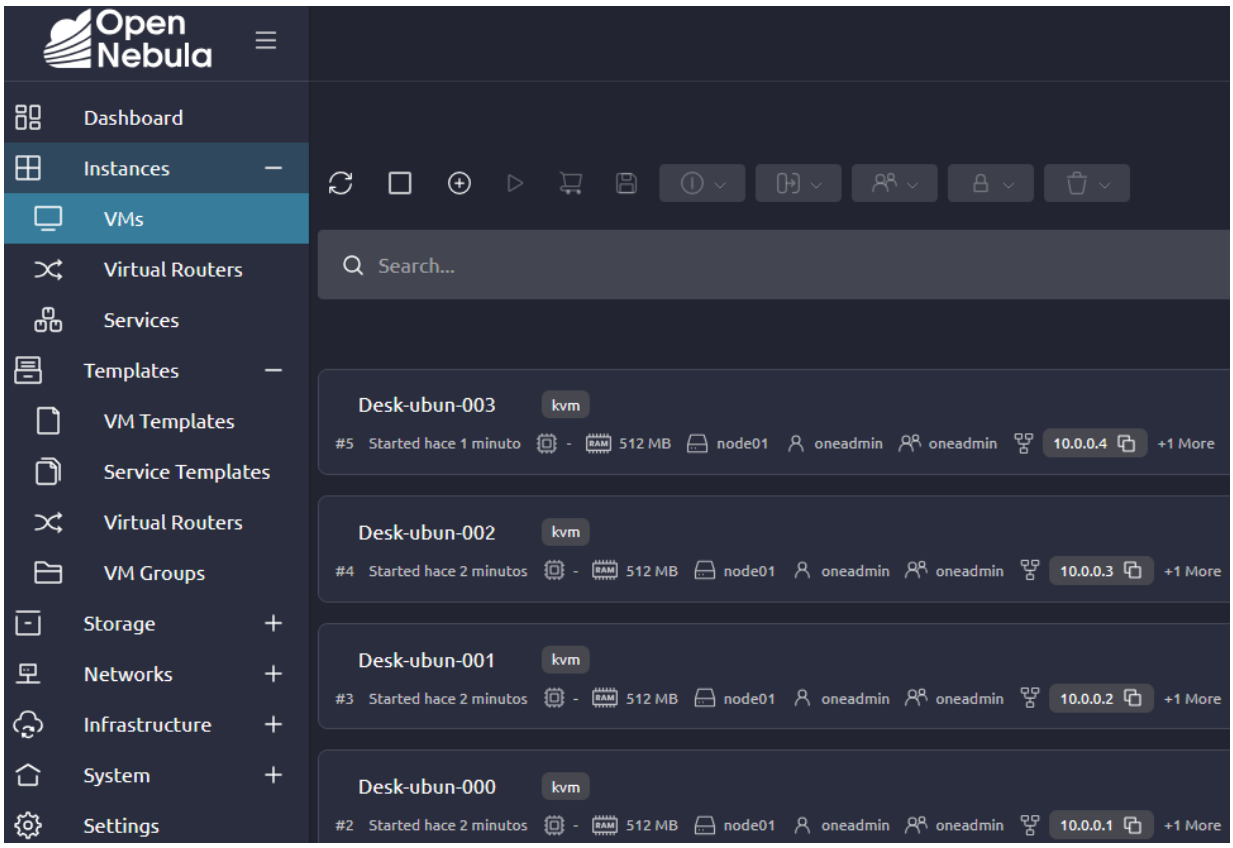


Once we have the entire UDS environment configured (Services, Authenticators, OS Managers and Transports) and created the first "***Service Pool***", we will be able to observe in OpenNebula how the desktop deployments are carried out.

The first task that will be performed will be to generate a publication of the template (this template will be generated every time we make a publication of a service). It will be a clone of the template chosen when registering for the service, with a disk size and characteristics equal to said template.

Once the process of creating the publication is finished (the system will name it as: "UDSP_*nombre_pool-número_publicación*") the desktops are automatically created (the system names them as: "*Machine_Name-Name_Length*") based on the cache parameters configured in the "***Service Pools***"".

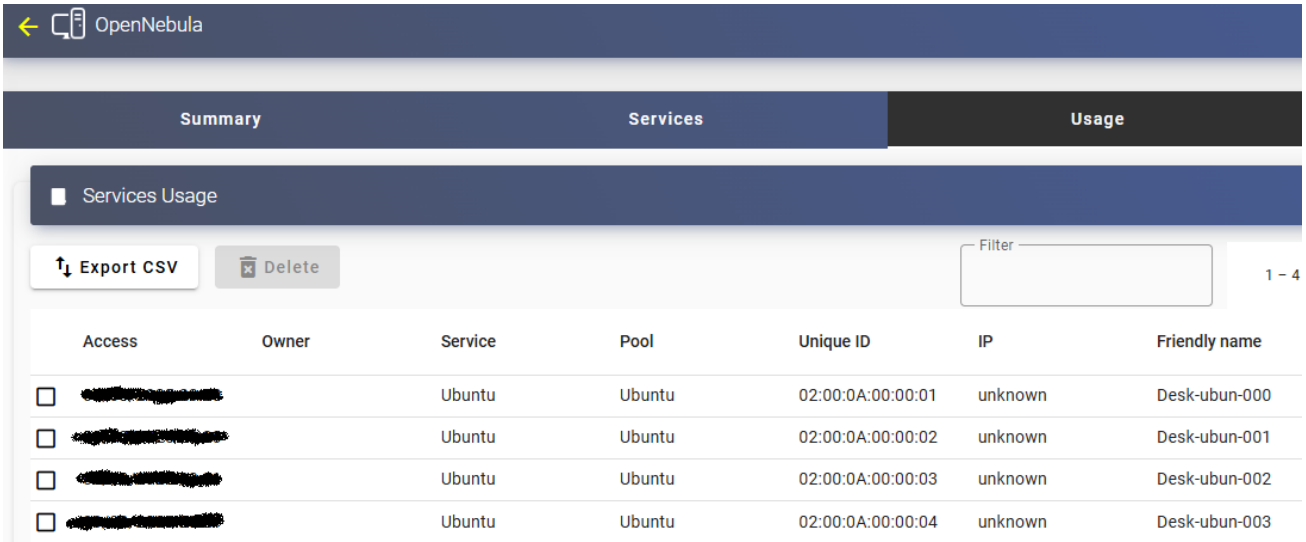




3.1.2.3 Service usage summary and error logs

Within a "***Service Provider***" we will have a tab called "***Usage***" that will allow us to have a quick view with detailed information and perform basic actions of the services deployed in this "***Service Provider***".

To access this information on the use of the services, we access the "***Service Providers***" and select the "***Usage" tab.***":



We will be able to visualize:

Access: Date and time the service was accessed.

Owner: User and authenticator assigned to the service in usuario@autenticador format. If it appears empty, it means that the service has not been assigned to any user and is available in the cache.

Service: Name of the base service belonging to the "***Service Provider***" from which the service was generated.

Pool: Name of the "***Service Pools***" to which the service belongs.

Unique ID: MAC address of the service.

IP: IP address of the service.

Friendly name: Name of the service. It will also be the DNS name of the UDS auto-generated virtual machine.

Src Ip: IP address of the connecting client accessing the service.

Src Host: DNS name of the connecting client accessing the service. If you cannot access this name, your IP address will be indicated.

It will also be possible to select one or more services to proceed with their removal.

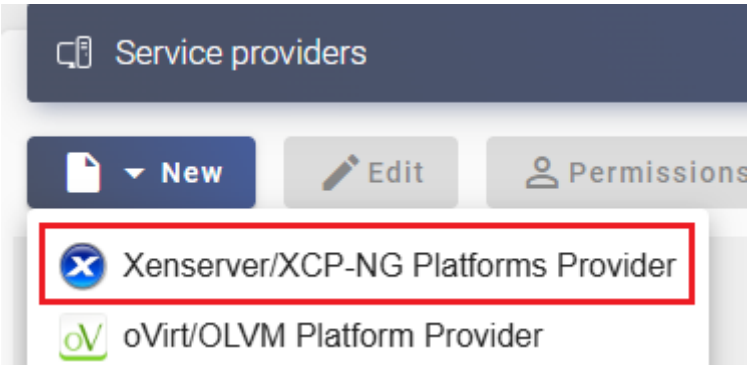
The "***Logs***" tab will show us information about a possible problem that occurred in the "***Service Provider'***":

3.1.3 VDI with Citrix XenServer / XCP-ng

Deployment of UDS-generated virtual desktops or connection to existing machines, on a Citrix XenServer or XCP-ng virtualization infrastructure.

3.1.3.1 Registration of Service provider “XenServer / XCP-NG”

Click on "***New***" and select "***Xenserver / XCP-NG Platform Provider***”.



En un "***Xenserver / XCP-NG Platform Provider***" los parámetros a configurar son:

- Main:

Name. Name to identify this provider

Host. IP address or XenServer or XCP-ng server name (In case of a server pool, the master must be indicated)

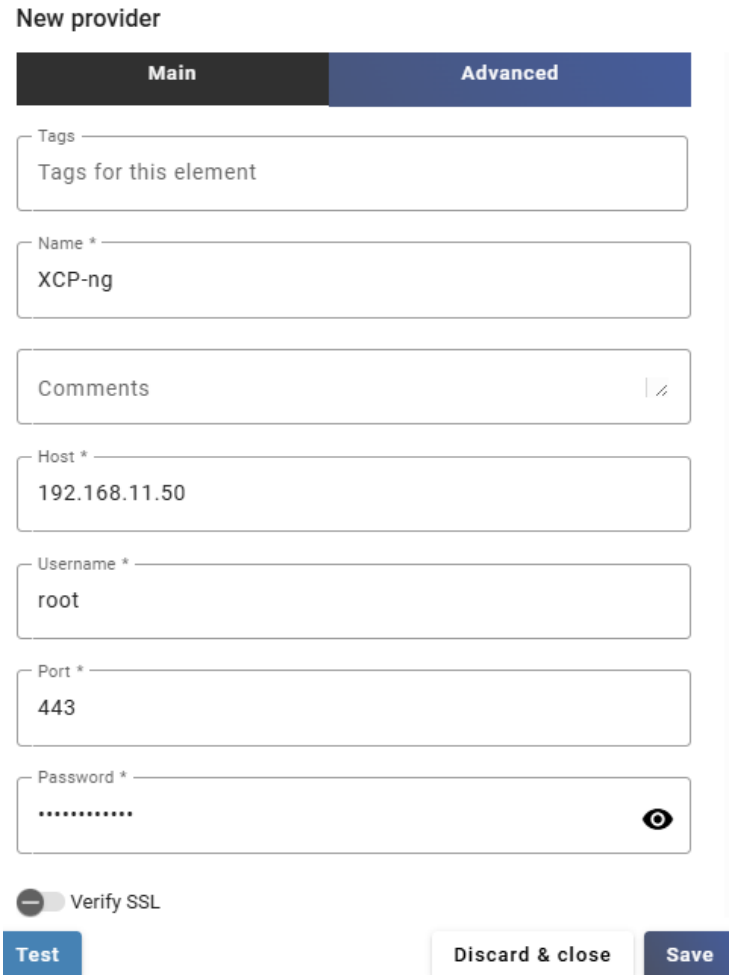
Port. Communication port with XenServer or XCP-ng server (default 443)

Username. Username with permissions for UDS integration with the provider. It is recommended that you use a user with admin permissions (use a user created specifically for this purpose).

Password. Password of the user used in the previous field

Verify SSL. If enabled, an SSL connection will be made to the server. It will be necessary to use the name of the server (in no case the IP address) and in addition the CA of the certificate used in the server it will be necessary to import it to the UDS server so that it can be recognized by it.

Safe Operating Procedure: Passwords must be of sufficient length and include upper and lower case, numbers and special characters.



- Advanced:

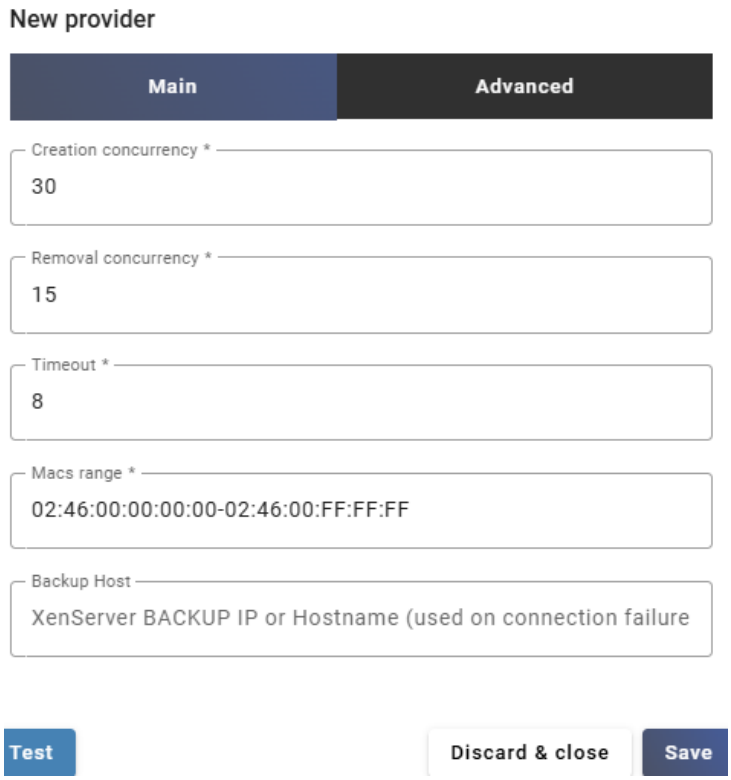
Creation concurrency. Maximum number of concurrent desktop creation tasks, in the different service pools that use it, for this provider.

Removal concurrency. Maximum number of concurrent desktop deletion tasks, in the different service pools that use it, for this provider.

Timeout. The time of "***Timeout***" on the connection to the server.

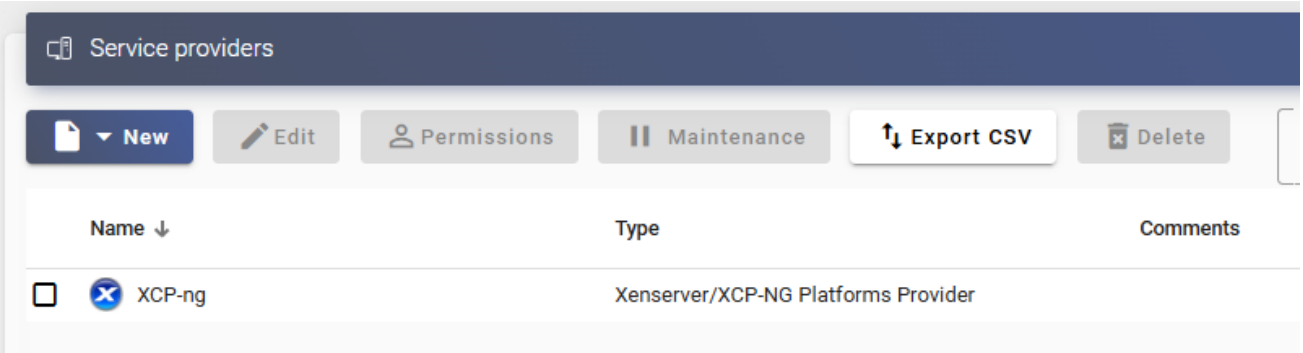
Macs range. MAC address range used by UDS to assign to automatically created virtual desktops. The modification of the range is always allowed to higher values, which will be applied immediately, but if it is reduced, it will be necessary that there is no record and it will also be necessary to wait a reasonable time of at least 24 hours for the records to be released and the change to be applied (to reduce the range, the registration of a new provider is recommended).

Backup Host: Backup server that will be used when there is no communication with the main (master) server indicated in the "**Host**" field of the "**Main tab**"".



Using the "***Test***" button we will check that the connection is successful.

Save the configuration and we will have a valid "***Service Providers***" to start registering base services on the Citrix XenServer or XCP-ng platform.

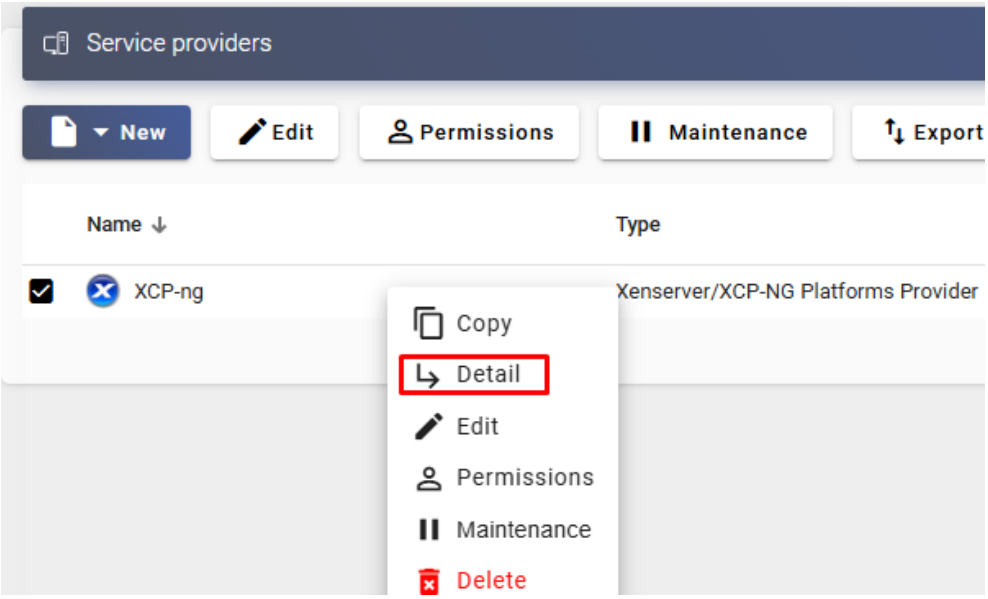


NOTE: If we have several Citrix XenServer or XCP-ng platforms, we will be able to register all the "Service Providers" of the type "Xenserver / XCP-NG Platform Provider" that we need.

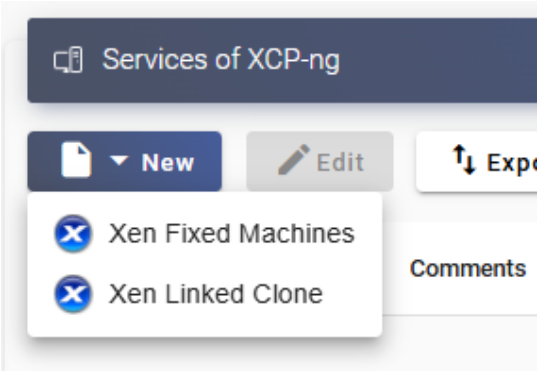
To modify a parameter in an existing ***"Service Providers***", select it and click on "***Edit***".

By clicking on the "***Enter Maintenance Mode***" button, you can pause all operations produced by the UDS server on a service provider. It is recommended to put a service provider in maintenance cases where communication with that service provider has been lost or a maintenance shutdown is planned.

Once the Citrix XenServer or XCP-ng platform is integrated into UDS, base services can be created. To do this, we double-click on the service provider created or in the provider's menu we select "***Detail***":



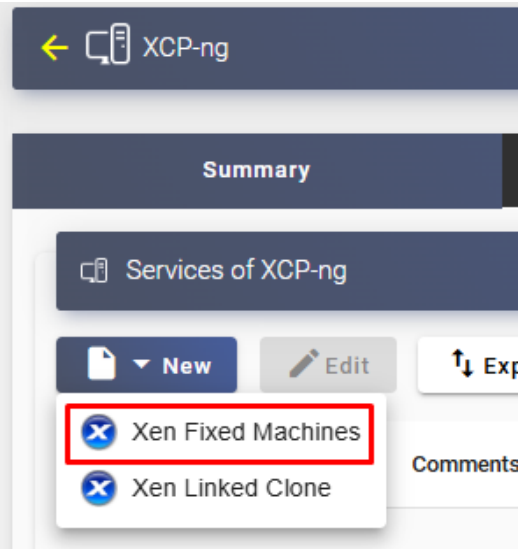
We will have the following types of basic services:



3.1.3.2 Configure service based on "Xen Fixed Machines""

This type of service will allow users to connect directly to existing virtual machines on a server or pool in the XenServer or XCP-ng environment.

To create base services of type "***Xen Fixed Machines***" we access the "***Service Providers***", select the "***Services***" tab, click on "***New***" and select "***Xen Fixed Machines***".



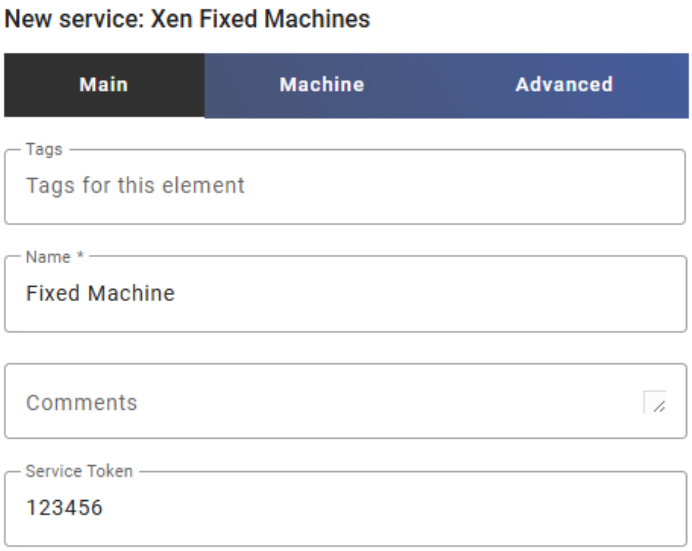
The minimum parameters to be configured in this type of service are:

- Main:

Name: Name of the service.

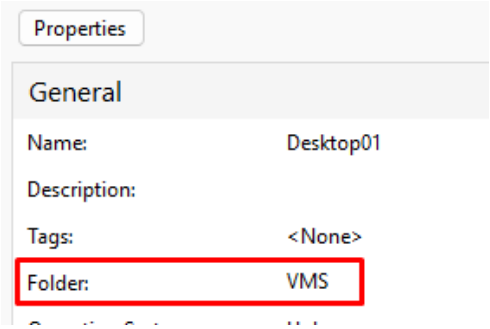
Service Token: If UDS is required to recognize the login and ***logout*** of a user accessing a service of this type (so that when a ***logout*** is made, UDS automatically releases the machine and makes it available to another user), it will be necessary to fill in this field with alphanumeric characters. If the field is left empty, UDS will permanently assign (until an administrator manually releases the service) the machine to a user, as it will have no way to control when a user enters or exits the machine.

NOTE: If the "**Service Token**" field is filled in, on the machines that are selected for the connection it will be necessary to have the UDS "**UDSActorUnmanagedSetup-x.x.x.exe**" actor installed and in the configuration of this, indicate those same characters in the "**Service Token" field**".



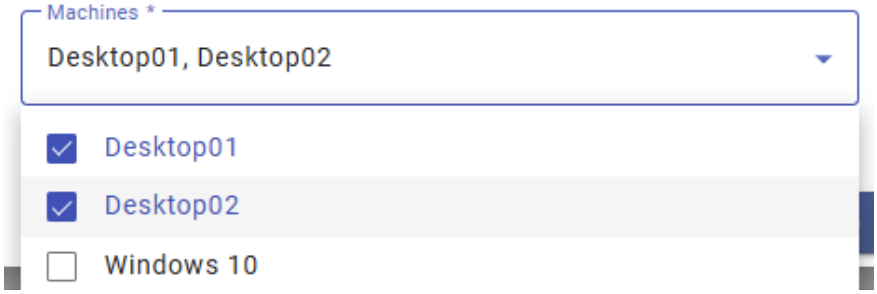
- Machine:

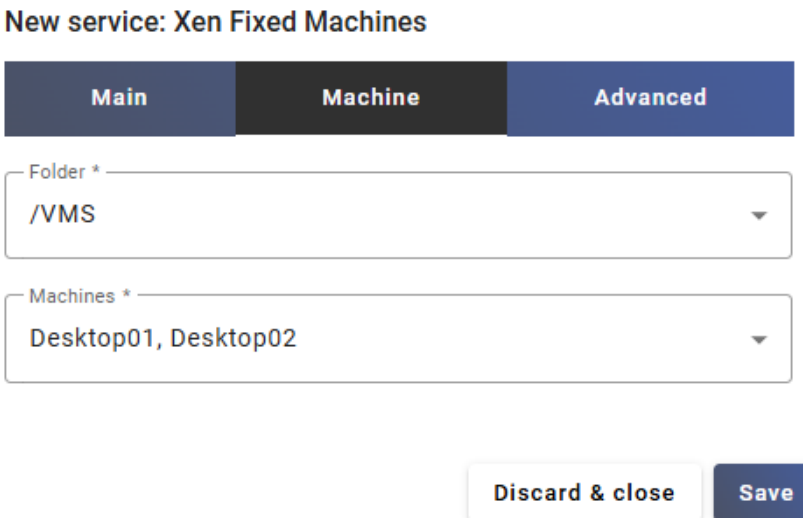
Folder: Folder in the XenServer or XCP-ng environment where the virtual machines that we want to manage and assign to users are located. Machines must be assigned to this folder to be visible



Machines: Virtual machines existing in the pool of resources indicated above that will be managed by UDS and to which access can be provided to the different users. We will have to indicate which machines will be accessible by users, the machines not selected, will be ignored and will not be assigned.

If the machines have the unamanaged actor installed and a "token" has been configured both in the actor (machine) and in the service definition (administration), it will be possible to control the access of users to these machines and they can be released so that other users can use them, in case they do NOT have the unmanaged actor installed, When a machine is assigned to a user, that assignment will be maintained until manually deleted by an administrator.





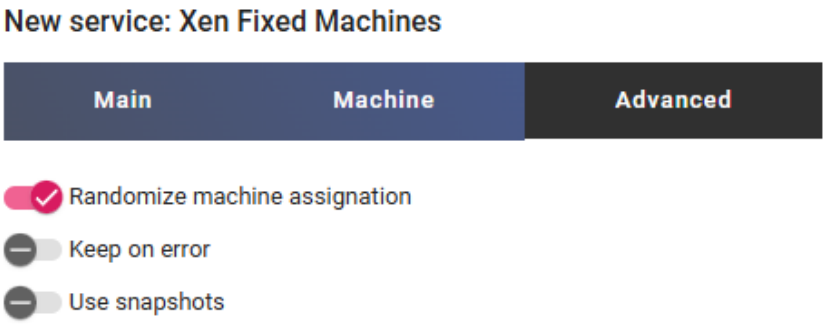
- Advanced

Randomize machine assignment: This parameter indicates how the order of assignment of the machines to the users will be, if it is active (by default) the order of assignment will be random, that is, when a user requests a machine, it will be unknown which one will be assigned. If the parameter is not active, the assignment order will be sequential, the first user will be given the first machine indicated in the list, the second the second machine and so on.

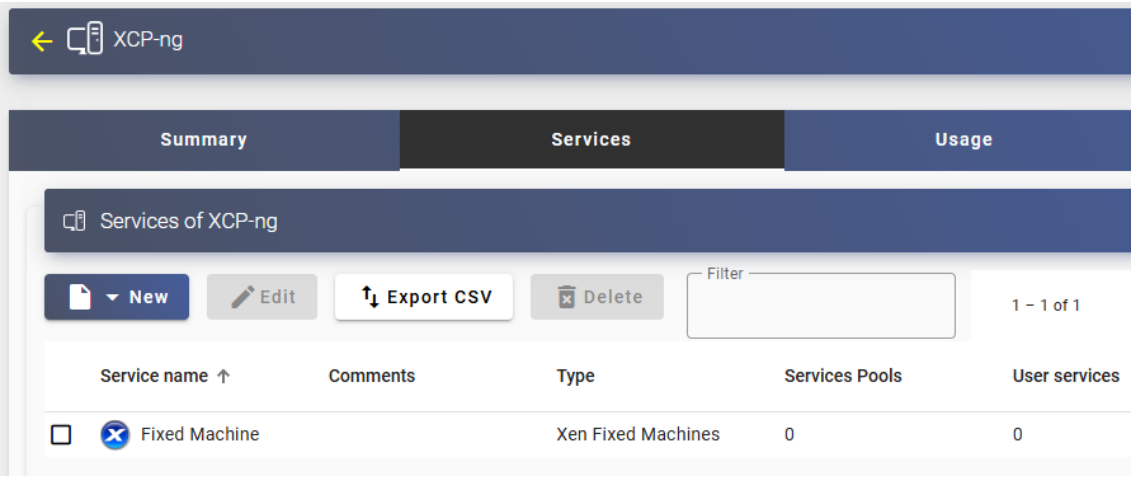
Keep on error: When activated, the system ignores all service access errors, allowing the machine to remain available for access (useful for occasions when machines may not be available for a short period of time but should be available at some point).

Use Snapshots: If this parameter is enabled, UDS will create a snapshot on the virtual machine before the user's connection and once its assignment is removed (through an automatic detection of the user's logout or manual deletion of an administrator) the snapshot will be recovered to leave the machine in the same state just before the user's connection. If this parameter is not activated, UDS will not recover any previous state of the machine when the assignment with a user is deleted.

NOTE: If the "**Use Snapshots" option is enabled** and the virtual machine already has a snapshot, UDS will not take any additional snapshots and will use and recover the existing snapshot.



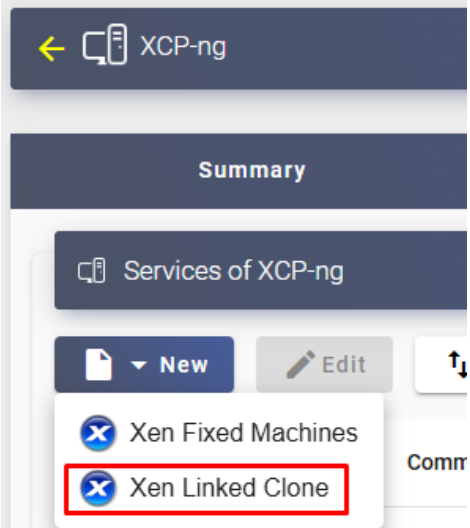
Save the configuration and we will have a valid "***Fixed Machines Pool***" on the XenServer or XCP-ng platform. We will be able to register all the "***Fixed Machines Pool***" services that we need.


XCP-ng
Summary
Services
Usage
Services of XCP-ng
New
Edit
Export CSV
Delete
Filter
1 – 1 of 1
Service name
Comments
Type
Services Pools
User services
Fixed Machine
Xen Fixed Machines
0
0

3.1.3.3 Configure service based on "Xen Linked Clone"

This type of service will create, within the virtual infrastructure, virtual desktops dependent on active publishing.

To create base services of type "***Xen Linked Clone***" we access the "***Service Providers***", select the "***Services***" tab, click on "***New***" and select "***Xen Linked Clone***".



The minimum parameters to be configured in this type of service are:

- Main:

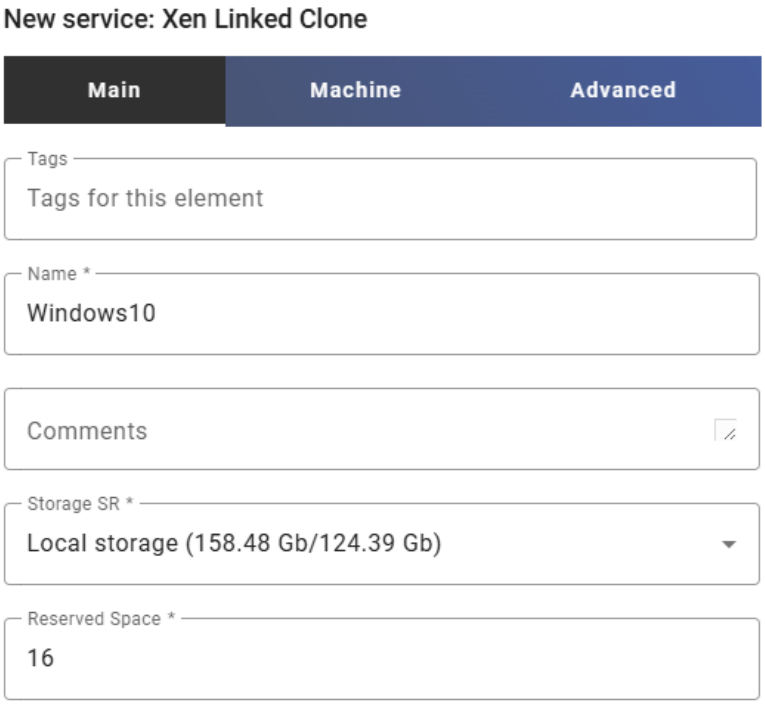
Name: Name of the service.

Storage SR: The location where generated publications and virtual desktops will be stored.

Only shared storage is supported, if we are connecting with a single Host and we want to use its local storage, we can manually modify it from the host console and convert it into SR storage in order to use it with UDS.

```
Server01 /]# xe sr-param-set uuid=27ba1986-b79d-6d5e-d995-7dcc2e4bd908 shared=true
```

Reserved Space: When there is less free space in a storage than indicated in this parameter (in GB), UDS will not use it to host publications or virtual desktops.



- Machine:

Base Names: Root of the name of all Linked Clones virtual desktops to be deployed in this service. (e.g.: Machine Names= Win10-).

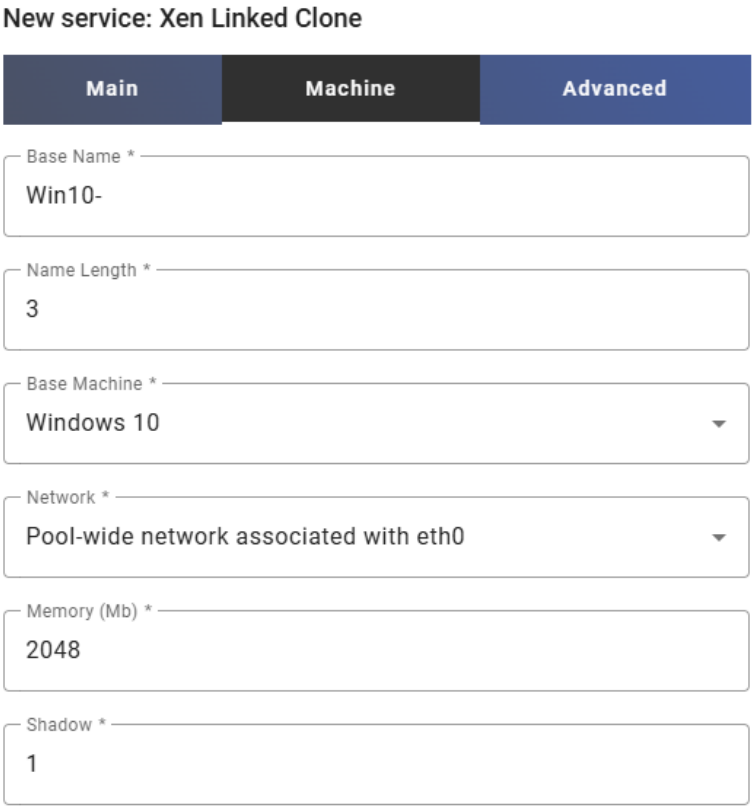
Name Length: Number of digits of the counter attached to the root of the desktop name (e.g.: Name Length= 3. The final name of the generated desktops would be: Win10-000**,** Win10-001... Win10-999).

Base Machine: Template virtual machine (Gold Image) used by the UDS system for the deployment of virtual desktops.

Network: Network where virtual desktops will be connected.

Memory: Amount of memory in Mb that will be allocated to the generated virtual desktops.

Shadow: Memory multiplier for the VM. The memory will be requested from the host and if available it will be applied to the VM using this value.



- Advanced:

Remove found duplicates: If this option is enabled (default), when UDS finds a virtual desktop with the same name (generated by UDS, but not registered) as the desktop you are about to create, it will delete it (it is recommended to use it to avoid "lost" VMs in the inventory).

Keep on error: When activated, the system ignores all errors in accessing the service, thus preventing desktop deletion. Very useful and recommended for use when using persistent machines, preventing the desktop from being deleted by the system under virtually any circumstances (except if an administrator manually deletes it).

Try SOFT Shutdown first: When activated, the system will attempt to perform a "Soft" shutdown on the OS of the virtual desktop. It is necessary that the tools are installed and running on the machine to be deleted, otherwise, after a period of time a forced shutdown will be performed.

Put back to cache: Parameter that defines whether machines will be returned to the cache, so that they can be used by another user, once they are freed:

- **No. Never put it back to cache:** When a user logs out of the desktop and the machine is non-persistent, it will be destroyed and will NOT be returned to the cache to be reused by more users.

- **Yes, try to put it back to cache:** When a user logs off the desktop and the machine is non-persistent, it will be returned IF it will be returned to the cache to be reused by more users.

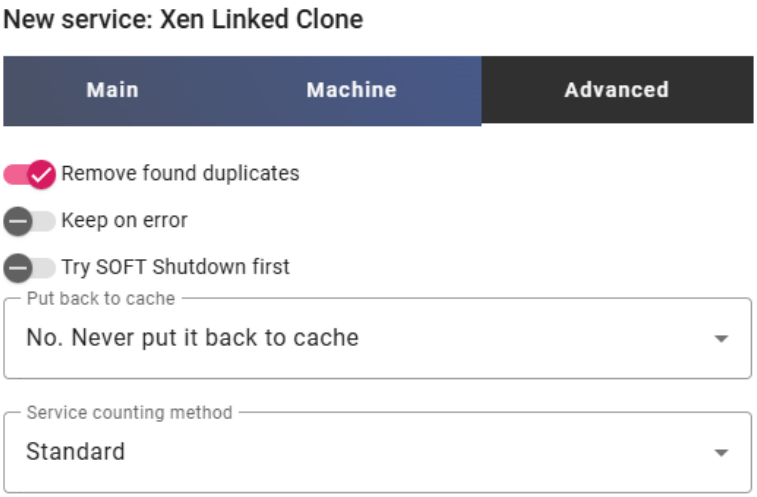
Service counting method: Method that the system will use to perform the machine count to comply with the parameter "maximum number of services to be provided" of a Service Pool:

- **Standard:** This counting mode only takes into account the machines that are being created and during the process of creating and destroying machines, it is possible to exceed the value of the parameter "maximum number of services to be provided" of a Service Pool.

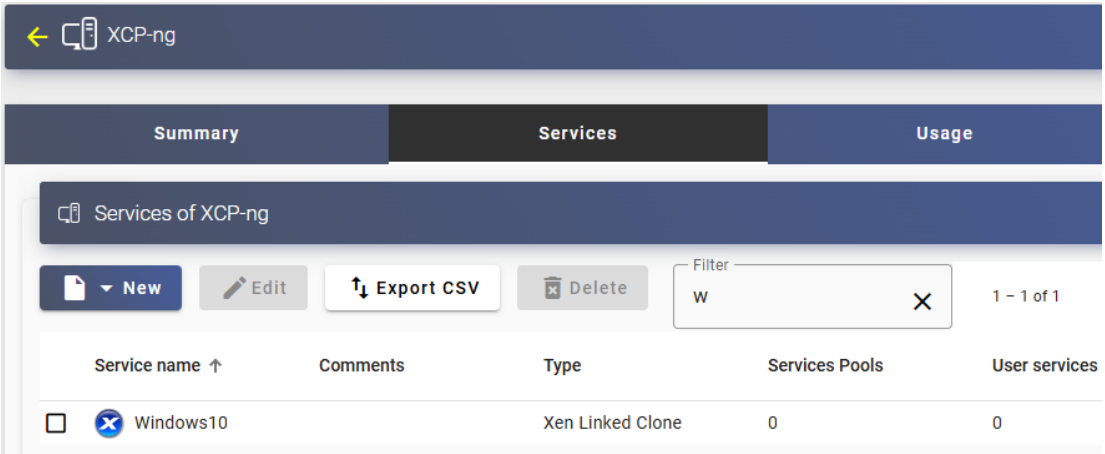
  For example, if a service pool, whose "maximum number of services to be provided" parameter is set to 5, we delete a service, there will be a time when that pool contains 6 machines: 4 configured, 1 being created and 1 being deleted.

- **Conservative:** This counting mode takes into account the machines that are being created and those that are being deleted, at no time will it be possible to exceed the value of the parameter "maximum number of services to be provided" of a Service Pool.

  For example, if a service pool, whose parameter of "maximum number of services to be provided" is set to 5, we delete a service, until that service is completely eliminated the system will not generate a new service, avoiding exceeding the value of 5 existing machines in the pool.



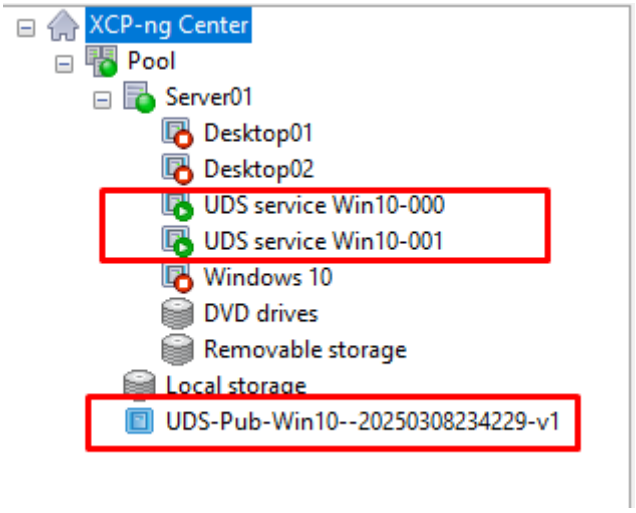
Save the configuration and we will have a valid " *Xen Linked Clone* " on the XenServer or XCP-ng platform. We will be able to register all the services of the " *Xen Linked Clone* " type that we need.



Once we have the entire UDS environment configured (Services, Authenticators, OS Managers and Transports) and created the first "***Service Pool***", we will be able to observe in XenCenter or XCP-ng Center how the desktop deployments are carried out.

The first task that will be performed will be to generate a publication of the template (this machine will be generated every time we publish a service), which will be a clone of the template chosen in the registration of the service, with a disk size and characteristics equal to said template.

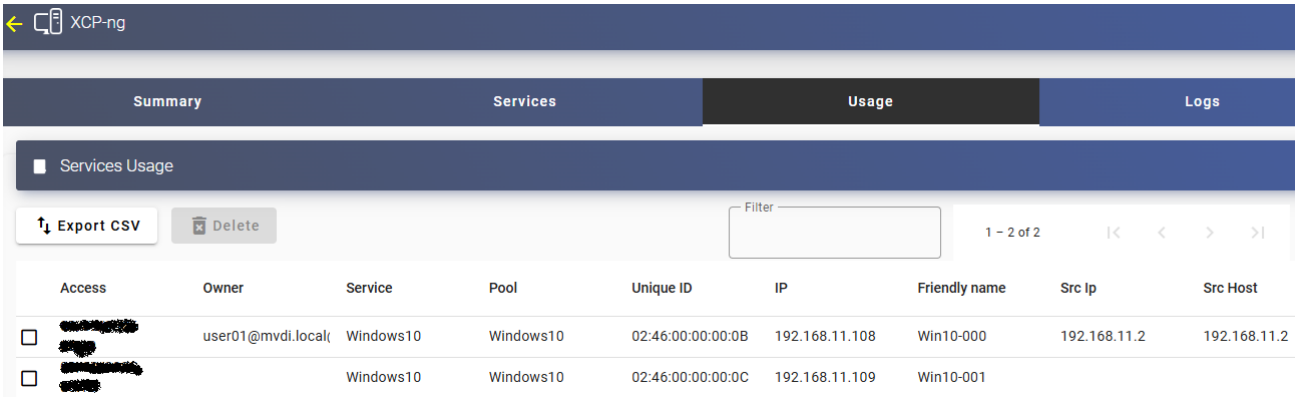
Once the process of creating the publication is finished (the system will name it as: "UDS-Pub-Base_Name-date-vnúmero_publicación") the desktops are automatically created (the system names them as: "UDS service *Machine_Name-Name_Length*") based on the cache parameters configured in the "***Service Pools***".



3.1.3.4 Service usage summary and error logs

Within a "***Service Provider***" we will have a tab called "***Usage***" that will allow us to have a quick view with detailed information and perform basic actions of the services deployed in this "***Service Provider***".

To access this information on the use of the services, we access the "***Service Providers***" and select the "***Usage" tab.***":



We will be able to visualize:

Access: Date and time the service was accessed.

Owner: User and authenticator assigned to the service in usuario@autenticador format. If it appears empty, it means that the service has not been assigned to any user and is available in the cache.

Service: Name of the base service belonging to the "***Service Provider***" from which the service was generated.

Pool: Name of the "***Service Pools***" to which the service belongs.

Unique ID: MAC address of the service.

IP: IP address of the service.

Friendly name: Name of the service. It will also be the DNS name of the virtual machine autogenerated by UDS.

Src Ip: IP address of the connecting client accessing the service.

Src Host: DNS name of the connecting client accessing the service. If you are unable to access this name, your IP address will be indicated.

It will also be possible to select one or more services to proceed with their deletion (in the base service of type "***Fixed*** " we will only delete the assignment with the user)..

La pestaña "***Logs***" nos mostrará información de algún posible problema ocurrido en el "***Service Provider***":

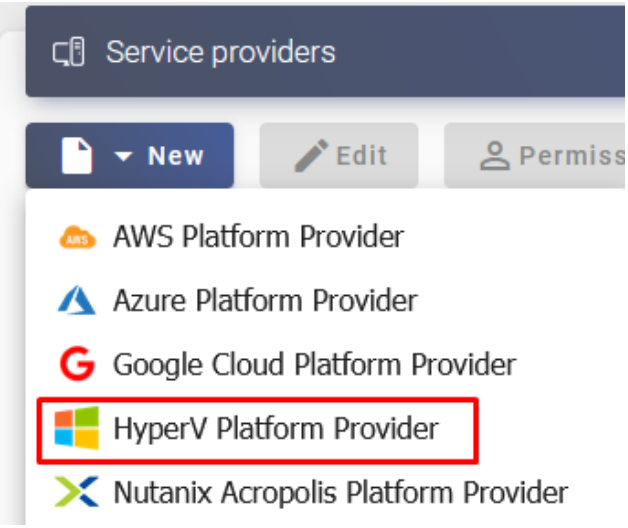
XCP-ng
Summary
Services
Usage
Logs
Logs
Export
Filter
0 of 0
date
level
source
message

3.1.4 VDI with Microsoft HyperV

Deployment of self-generated virtual desktops by UDS or connection to existing machines, in a Microsoft HyperV virtualization infrastructure.

3.1.4.1 Service Provider Registration "HyperV" Platform Provider"

Click on "***New***" and select "***HyperV Platform Provider***":



In a "***HyperV Platform Provider***" the minimum parameters to be configured are:

- Main:

Name: Name to identify this provider

Host: IP address or server name of the HyperV server, If we connect against a cluster, the IP or name of the cluster must be indicated.

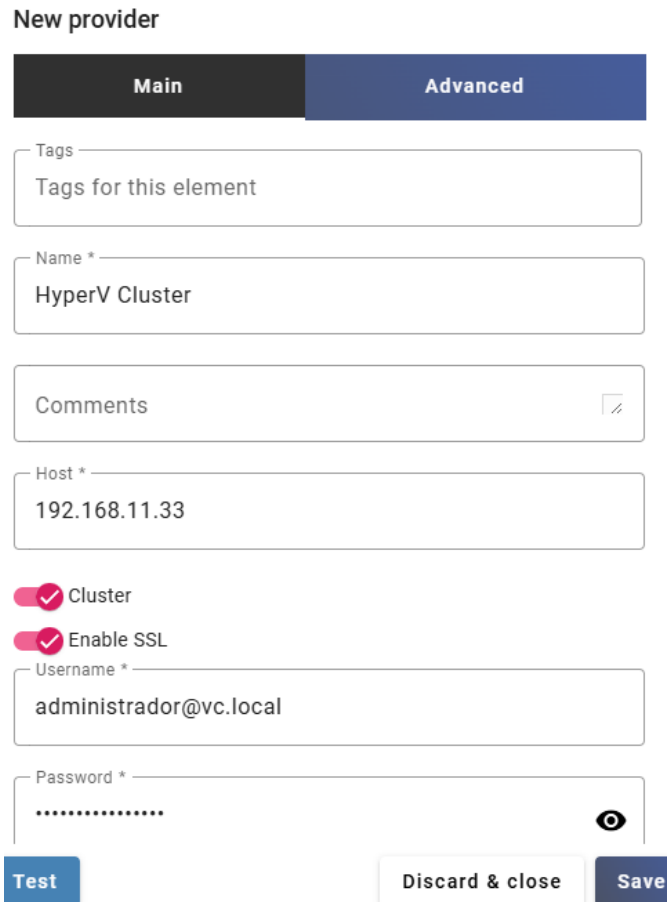
Cluster: If the HyperV environment to which we connect is composed of a cluster, it will be necessary to enable this option. If we are going to connect to a single HyperV Host in a "Standalone" node, it will not be necessary to enable it.

Enable SSL: Enables SSL communication with the HyperV server or cluster.

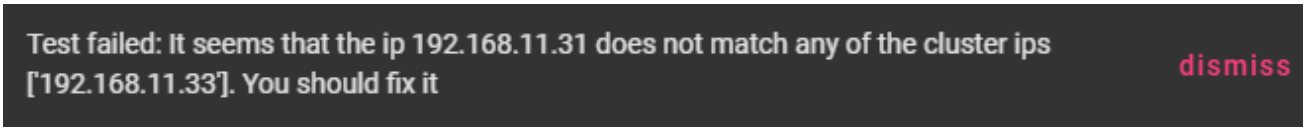
Username: Username with permissions for UDS integration with the provider. It is recommended that you use a user with admin permissions (use a user created specifically for this purpose).

Password: Password of the user used in the previous field.

Safe Operating Procedure: Passwords must be of sufficient length and include upper and lower case, numbers and special characters.



NOTE: If we try to directly connect a Hyper-V server that is part of a cluster, the system will warn us that we will have to connect against the cluster that manages it.



For the connection to succeed, the following script will need to be run on all Hyper-V nodes (even if they are part of a cluster) from PowerShell:

```
Invoke-Expression ((New-
ObjectSystem.Net.Webclient).DownloadString('https://images.udsenterprise.com/file
s/hyperv/EnableRemoting.ps1'))
```

- Advanced:

Creation concurrency: Maximum number of concurrent desktop creation tasks, in the different service pools that use it, for this provider.

Removal concurrency: Maximum number of concurrent desktop removal tasks, in the different service pools that use it, for this provider.

Port: Connection port with the HyperV server (via WSman), if not indicated the default (5985) will be used.

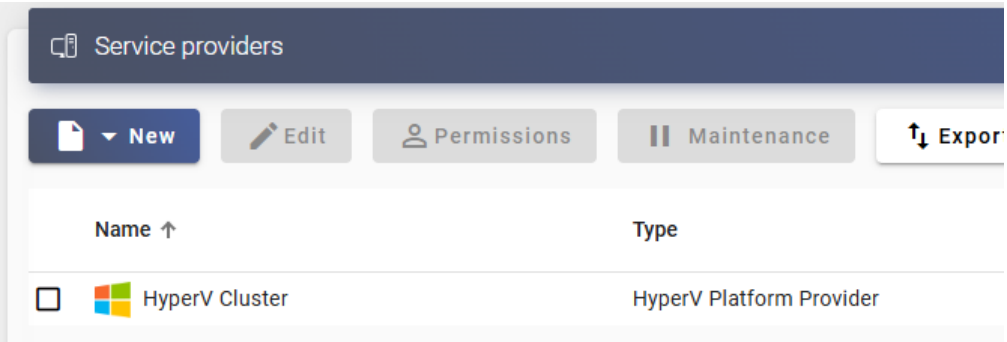
Force SSL certificate verification: If enabled, a certificate check will be performed. It will be necessary to use the name of the server (in no case the IP address) and in addition the CA of the certificate used in the server it will be necessary to import it to the UDS server so that it can be recognized by it.

Timeout: The time of "***Timeout***" in the connection to the server.

Macs range: Range of MAC addresses used by UDS to assign to automatically created virtual desktops. The modification of the range is always allowed to higher values, which will be applied immediately, but if it is reduced, it will be necessary that there is no record and it will also be necessary to wait a reasonable time of at least 24 hours for the records to be released and the change to be applied (to reduce the range, the registration of a new provider is recommended).

By clicking on the "***Test***" button, we will check that the connection is correct.

Save the configuration and we will have a valid "***Service Providers***" to start registering basic services of the Microsoft Hyper-V platform.

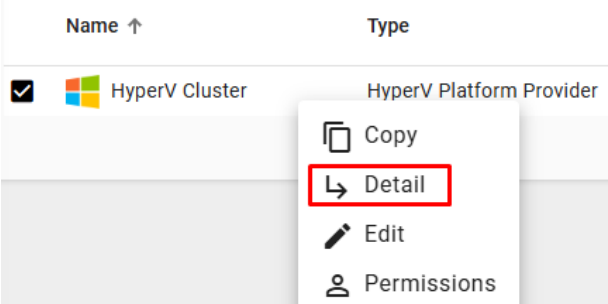


NOTA: If we have several Microsoft Hyper-V platforms (clustered or Standalone), we will be able to register all the "Service Providers" of the "HyperV Platform Provider" type that we need.

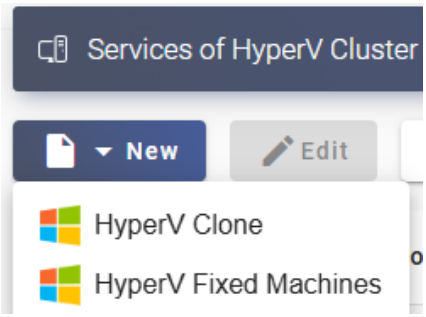
To modify a parameter in an existing ***"Service Providers***", select it and click on "***Edit***".

By clicking on the "***Enter Maintenance Mode***" button, you can pause all operations produced by the UDS server on a service provider. It is recommended to put a service provider in maintenance cases where communication with that service provider has been lost or a maintenance shutdown is planned.

Once the Microsoft Hyper-V platform is integrated into UDS, base services can be created. To do this, we double-click on the service provider created or in the provider's menu we select "***Detail***":



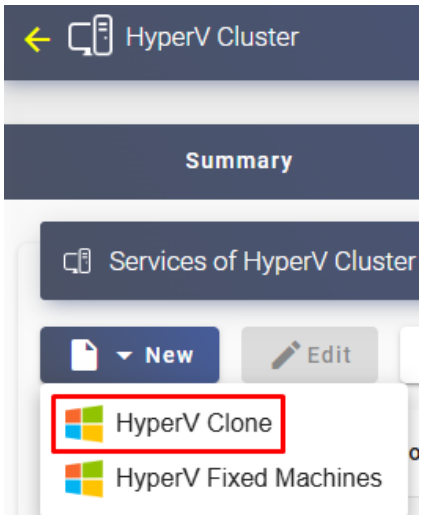
We will have the following types of basic services:



3.1.4.2 Configure service based on "HyperV Clone"

This type of service will create, within the virtual infrastructure, virtual desktops dependent on active publishing.

To create base services of type "***HyperV Clone***" we access the "***Service Providers***", select the "***Services***" tab, click on "***New***" and select "***HyperV Clone***".



The minimum parameters to be configured in this type of service are:

- Main:

Name: Name of the service.

Datastores Drives: Location where the generated publications and virtual desktops will be stored (In this storage a folder called "***UDSData***" will be created where all the disks of the generated machines will be saved).

- Machine:

Network: Network where virtual desktops will be connected.

Memory: Amount of memory in Mb that will be allocated to the generated virtual desktops.

Base Machine: Template virtual machine (Gold Image) used by the UDS system for the deployment of virtual desktops (Generation 1 machines must be used).

Machine Names: Root of the name of all virtual desktops to be deployed in this service. (e.g.: Machine Names= Ubuntu-).

Name Length: Number of digits of the counter attached to the root of the desktop name (e.g.: Name Length= 3. The final name of the generated desktops would be: **Ubuntu-000,** Ubuntu-001... **Ubuntu-999**).

- Advanced:

Reserved Space: When there is less free space in a storage than indicated in this parameter (in GB), UDS will not use it to host publications or virtual desktops.

Use Full Clones: The "Linked Clone" model will not be used and each desktop generated will occupy the total disk space and will be independent of the publishing machine.

Enable Nested Virtualization: Enables the "Nested Virtualization" feature on the generated virtual desktops".

Enable Compatibility for Magration: Enables the processor compatibility option on the generated virtual desktops to be able to migrate machines between physical hosts with a different processor version.

Remove found duplicates: If this option is enabled (default), when UDS finds a virtual desktop with the same name (generated by UDS, but not registered) as the desktop you are about to create, it will delete it (it is recommended to use it to avoid "lost" VMs in the inventory).

Keep on error: When activated, the system ignores all errors in accessing the service, thus preventing desktop deletion. Very useful and recommended for use when using persistent machines, preventing the desktop from being deleted by the system under virtually any circumstances (except if an administrator manually deletes it).

Try SOFT Shutdown first: When activated, the system will attempt to perform a "Soft" shutdown on the OS of the virtual desktop. It is necessary that the tools are installed and running on the machine to be deleted, otherwise, after a period of time a forced shutdown will be performed.

Put back to cache: Parameter that defines whether machines will be returned to the cache, so that they can be used by another user, once they are freed:

- **No. Never put it back to cache:** When a user logs out of the desktop and the machine is non-persistent, it will be destroyed and will NOT be returned to the cache to be reused by more users.

- **Yes, try to put it back to cache:** When a user logs out of the desktop and the machine is non-persistent, it will be returned IF it will be returned to the cache to be reused by more users.

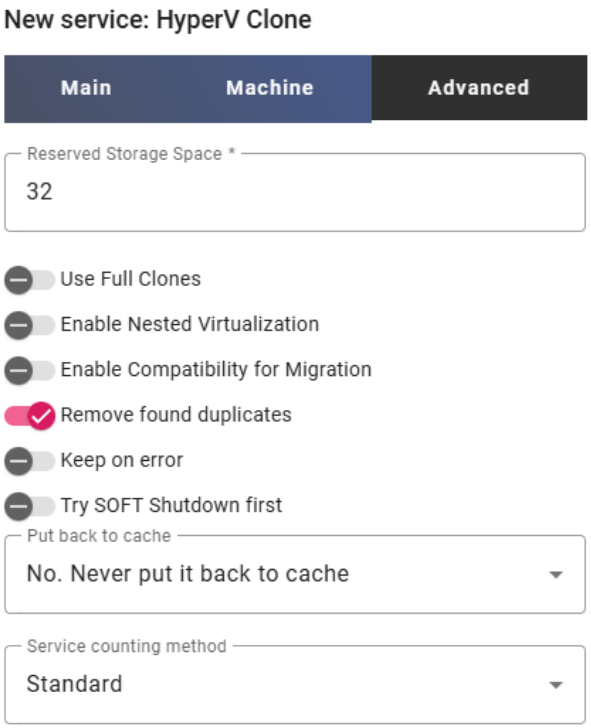
Service counting method: Method that the system will use to perform the machine count to comply with the parameter "maximum number of services to be provided" of a Service Pool:

- **Standard:** This counting mode only takes into account the machines that are being created and during the process of creating and destroying machines, it is possible to exceed the value of the parameter "maximum number of services to be provided" of a Service Pool.

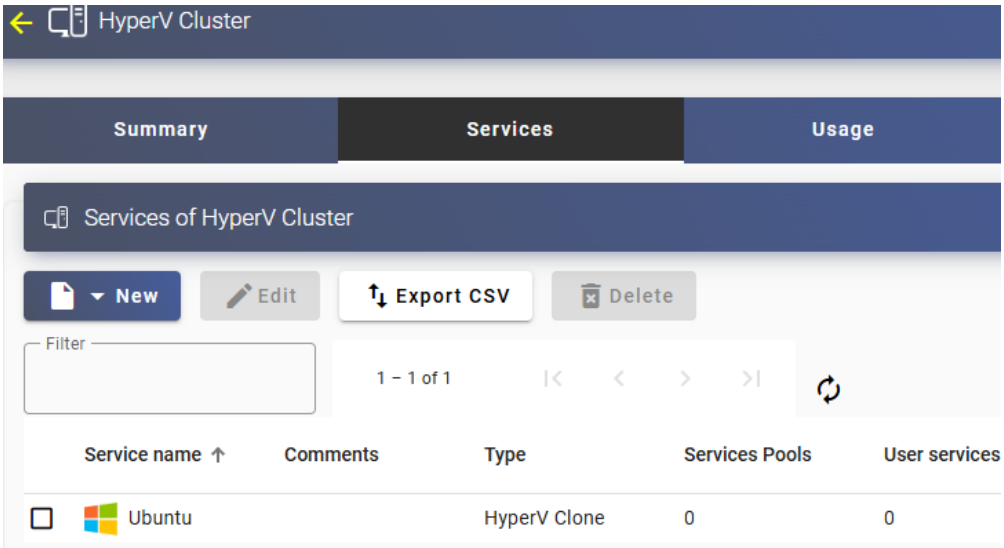
  For example, if a service pool, whose "maximum number of services to be provided" parameter is set to 5, we delete a service, there will be a time when that pool contains 6 machines: 4 configured, 1 being created and 1 being deleted.

- **Conservative:** This counting mode takes into account the machines that are being created and those that are being deleted, at no time will it be possible to exceed the value of the parameter "maximum number of services to be provided" of a Service Pool.

  For example, if a service pool, whose parameter of "maximum number of services to be provided" is set to 5, we delete a service, until that service is completely eliminated the system will not generate a new service, avoiding exceeding the value of 5 existing machines in the pool.



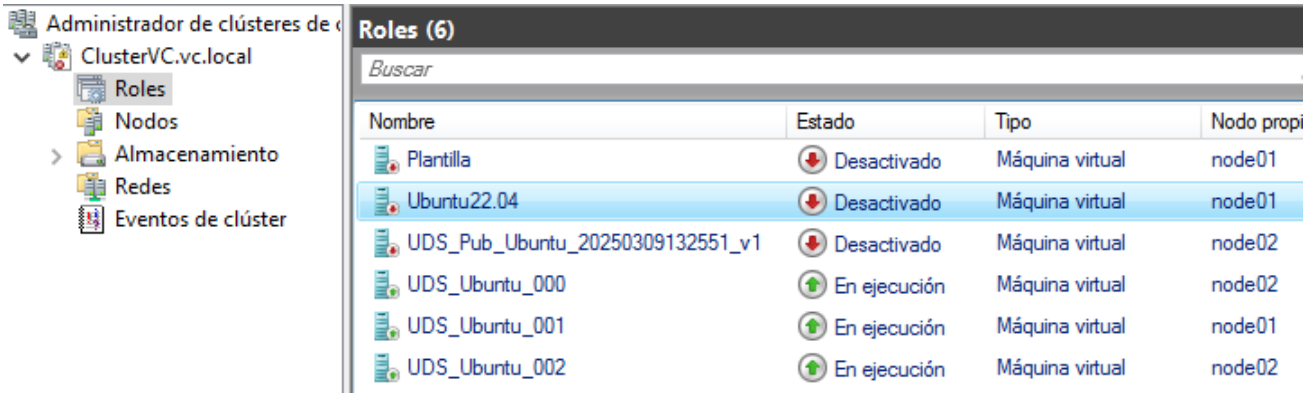
Save the configuration and we will have a valid "***HyperV Clone***" on the Hyper-V platform. We will be able to register all the "***HyperV Clone***" we need.



Once we have the entire UDS environment configured (Services, Authenticators, OS Managers and Transports) and created the first "***Service Pool***", we will be able to observe in the administration of the Hyper-V cluster or the Hyper-V Standalone server how the desktop deployments are carried out.

The first task that will be performed will be to generate a publication of the template (this machine will be generated every time we make a publication of a service). It will be a clone of the template chosen when registering for the service, with a disk size and characteristics equal to said template.

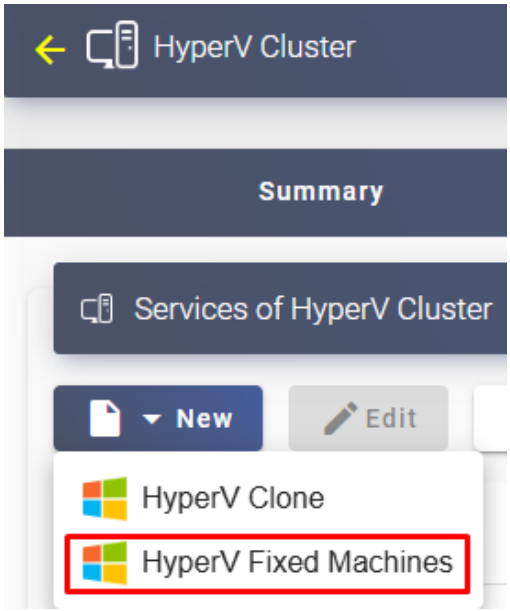
Once the process of creating the publication is finished (the system will name it as: "UDS-Pub-Base_Name-date-vnúmero_publicación") the virtual desktops are automatically created (the system names them as: "UDS_ *Machine_Name-Name_Length*") based on the cache parameters configured in the "***Service Pools***"".



3.1.4.3 Configure service based on "HyperV Fixed Machines""

This type of service will allow users to connect directly to existing virtual machines on a HyperV server or cluster.

To create base services of type "***HyperV Fixed Machines***" we access the "***Service Providers***", select the "***Services***" tab, click on "***New***" and select "***HyperV Fixed Machines***".



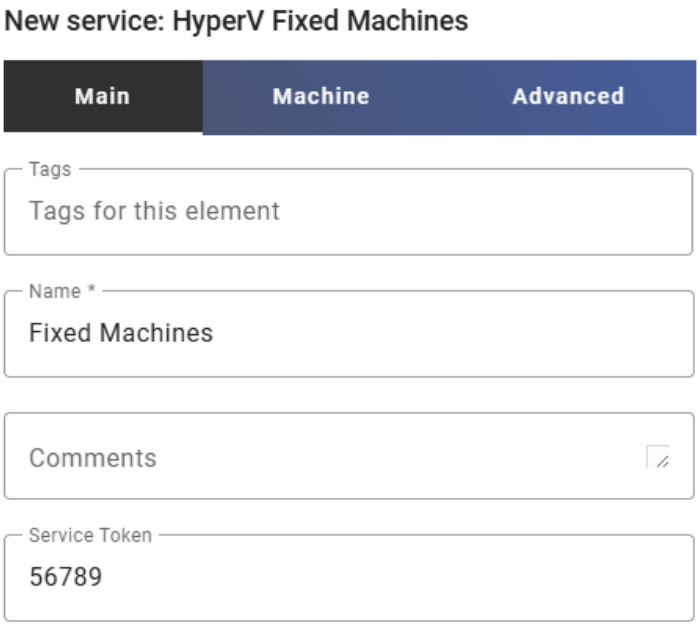
The minimum parameters to be configured in this type of service are:

- Main:

Name: Name of the service.

Service Token: If UDS is required to recognize the login and ***logout*** of a user accessing a service of this type (so that when a ***logout*** is made, UDS automatically releases the machine and makes it available to another user), it will be necessary to fill in this field with alphanumeric characters. If the field is left empty, UDS will permanently assign (until an administrator manually releases the service) the machine to a user, as it will have no way to control when a user enters or exits the machine.

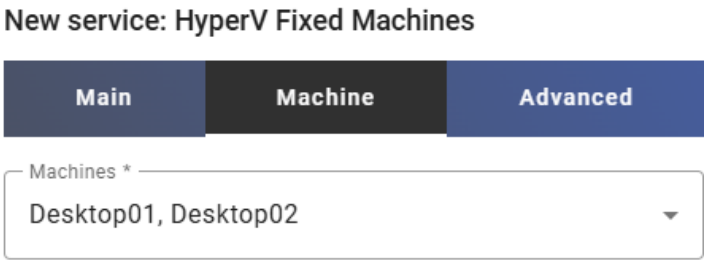
NOTE: If the "**Service Token**" field is filled in, on the machines that are selected for the connection it will be necessary to have the UDS "**UDSActorUnmanagedSetup-x.x.x.exe**" actor installed and in the configuration of this, indicate those same characters in the "**Service Token" field**".



- Machine:

Machines: Virtual machines existing in the virtualization environment that will be managed by UDS and to which access can be provided to the different users. We will have to indicate which machines will be accessible by users, the machines not selected, will be ignored and will not be assigned.

If the machines have the unamanaged actor installed and a "token" has been configured both in the actor (machine) and in the service definition (administration), it will be possible to control the access of users to these machines and they can be released so that other users can use them, in case they do NOT have the unmanaged actor installed, When a machine is assigned to a user, that assignment will be maintained until manually deleted by an administrator.



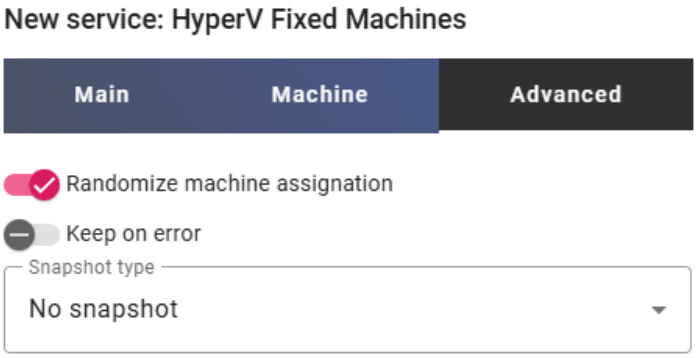
- Advanced

Randomize machine assignment: This parameter indicates how the order of assignment of the machines to the users will be, if it is active (by default) the order of assignment will be random, that is, when a user requests a machine, it will be unknown which one will be assigned. If the parameter is not active, the assignment order will be sequential, the first user will be given the first machine indicated in the list, the second the second machine and so on.

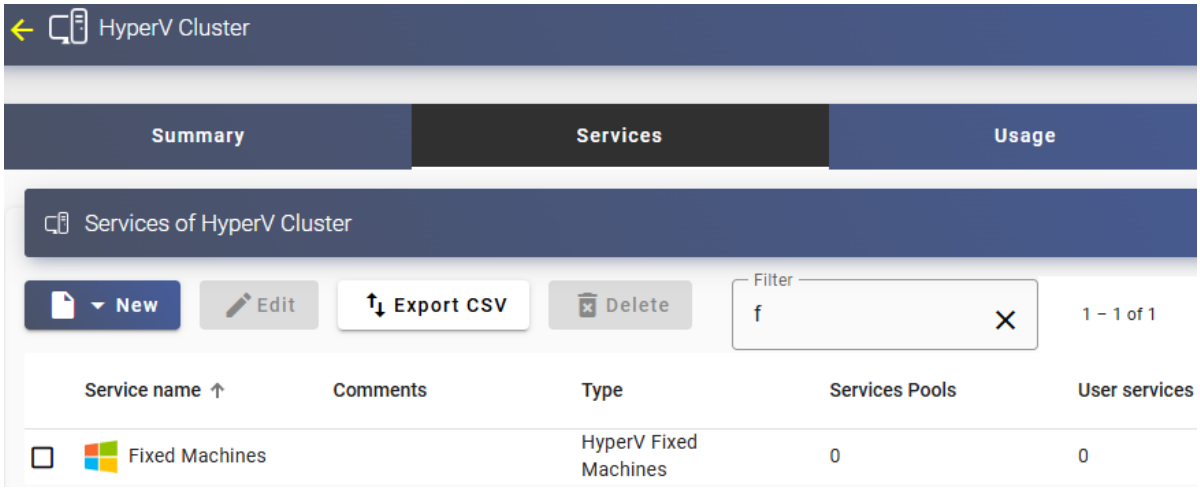
Keep on error: When activated, the system ignores all service access errors, allowing the machine to remain available for access (useful for occasions when machines may not be available for a short period of time but should be available at some point).).

Use Snapshots: If this parameter is enabled, UDS will create a snapshot on the virtual machine before the user's connection and once its assignment is deleted (through an automatic detection of the user's logout or manual deletion of an administrator) the snapshot will be recovered to leave the machine in the same state just before the user's connection. If this parameter is not activated, UDS will not recover any previous state of the machine when the assignment with a user is deleted.

NOTE: If the **"Use Snapshots" option is enabled** and the virtual machine already has a snapshot, UDS will not take any additional snapshots and will use and recover the existing snapshot.



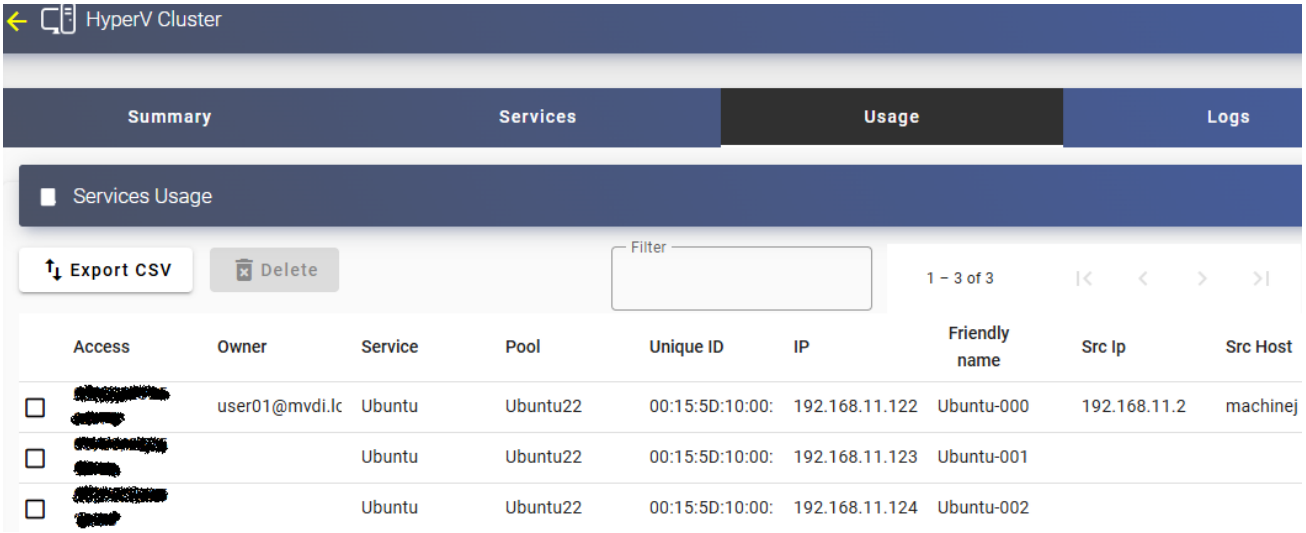
Save the configuration and we will have a valid " ***HyperV Fixed Machines*** " on the HyperV platform. We will be able to register all the " ***HyperV Fixed Machines*** " services that we need.



3.1.4.4 Service usage summary and error logs

Within a "***Service Provider***" we will have a tab called "***Usage***" that will allow us to have a quick view with detailed information and perform basic actions of the services deployed in this "***Service Provider***".

To access this information on the use of the services, we access the "***Service Providers***" and select the "***Usage***" tab:



We will be able to visualize:

Access: Date and time the service was accessed.

Owner: User and authenticator assigned to the service in usuario@autenticador format. If it appears empty, it means that the service has not been assigned to any user and is available in the cache.

Service: Name of the base service belonging to the "***Service Provider***" from which the service was generated.

Pool: Name of the "***Service Pools***" to which the service belongs.

Unique ID: MAC address of the service.

IP: IP address of the service.

Friendly name: Name of the service. It will also be the DNS name of the UDS auto-generated virtual machine.

Src Ip: IP address of the connecting client accessing the service.

Src Host: DNS name of the connecting client accessing the service. If you cannot access this name, your IP address will be indicated.

It will also be possible to select one or more services to proceed with their deletion (in the base service of type "***Fixed*** " we will only delete the assignment with the user).

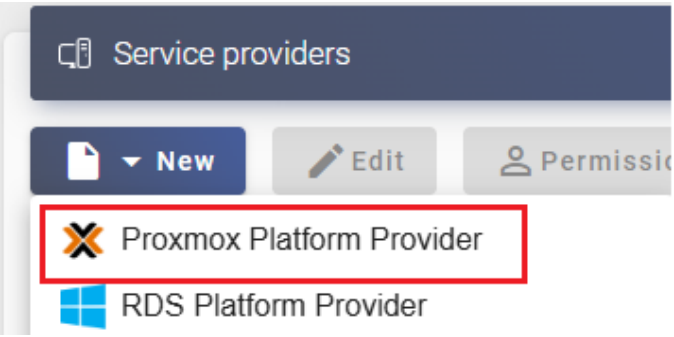
The "***Logs***" tab will show us information about a possible problem that occurred in the "***Service Provider'***":

3.1.5 VDI with Proxmox

Deployment of UDS-generated virtual desktops or connection to existing machines, in a Proxmox virtualization infrastructure.

3.1.5.1 Registration of a service provider "Proxmox Platform Provider""

Click on "***New***" and select "***Proxmox Platform Provider***".



In a "***Proxmox Platform Provider***" the minimum parameters to be configured are:

- Main:

Name. Name to identify this provider.

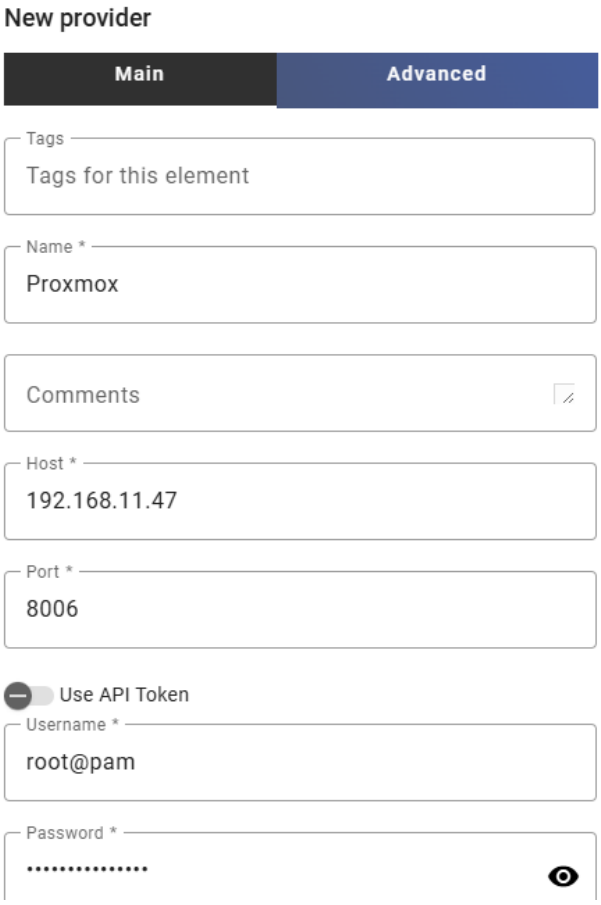
Host. IP address or Proxmox server name.

Port. Communication port with Proxmox server (default 8006)

Username. Username with permissions for UDS integration with the provider. It is recommended that you use a user with admin permissions (use a user created specifically for this purpose).

Password. Password of the user used in the previous field.

Safe use procedure: Passwords must be of sufficient length and include upper and lower case, numbers and special characters.



- Advanced:

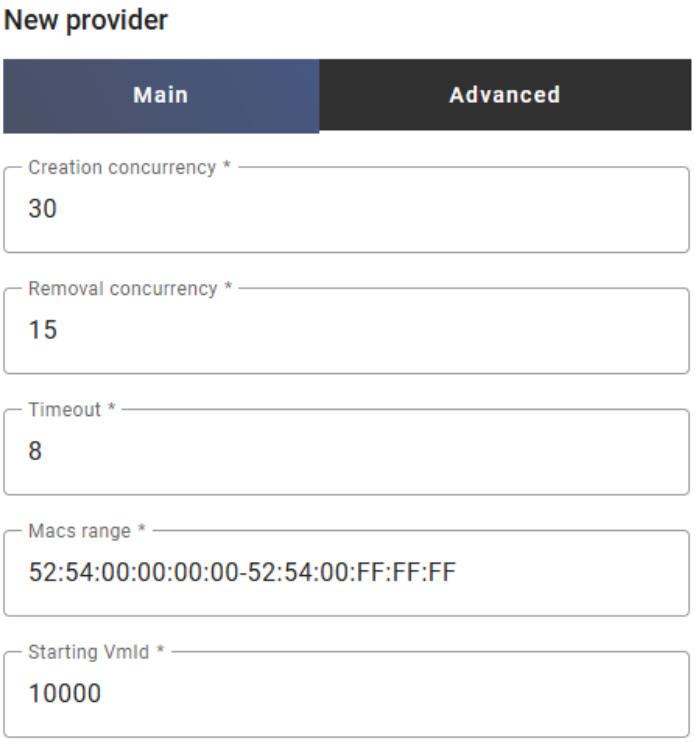
Creation concurrency. Maximum number of concurrent desktop creation tasks, in the different service pools that use it, for this provider.

Removal concurrency. Maximum number of concurrent desktop deletion tasks, in the different service pools that use it, for this provider.

Timeout. The time of "***Timeout***" on the connection to the server.

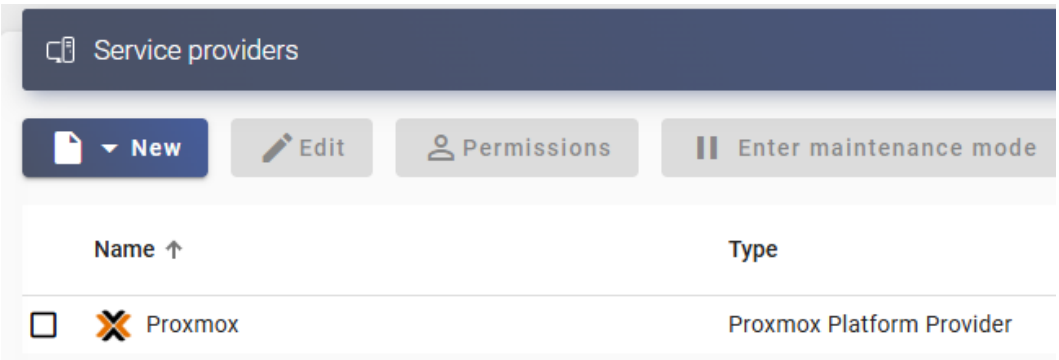
Macs range. MAC address range used by UDS to assign to automatically created virtual desktops. The modification of the range is always allowed to higher values, which will be applied immediately, but if it is reduced, it will be necessary that there is no record and it will also be necessary to wait a reasonable time of at least 24 hours for the records to be released and the change to be applied (to reduce the range, the registration of a new provider is recommended).

Starting Vmid: Machine ID by which UDS will start generating machines.



Using the "***Test***" button we will check that the connection is successful.

Save the configuration and we will have a valid ***"Service Providers***" to start registering base services on the Proxmox platform.

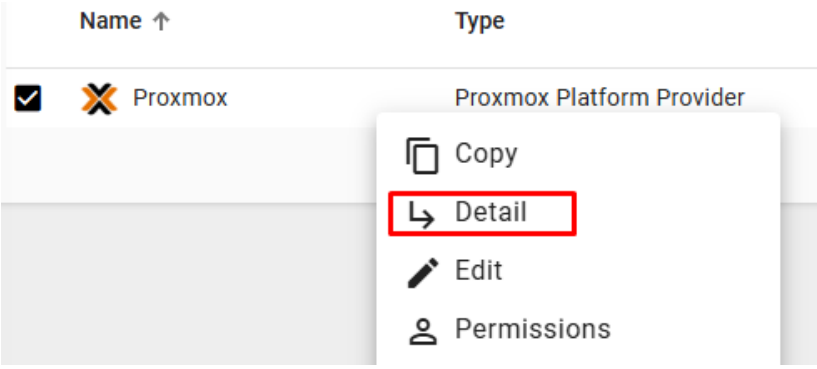


NOTE: If we have several Proxmox platforms (both independent servers and clusters), we will be able to register all the "Service Providers" of the "Proxmox Platform Provider" type that we need.

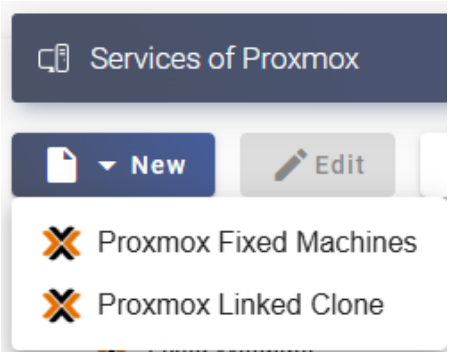
To modify a parameter in an existing ***"Service Providers***", select it and click on "***Edit***".

By clicking on the "***Enter Maintenance Mode***" button, you can pause all operations produced by the UDS server on a service provider. It is recommended to put a service provider in maintenance cases where communication with that service provider has been lost or a maintenance shutdown is planned.

Once the Proxmox platform is integrated into UDS, base services can be created. To do this, we double-click on the service provider created or on the provider's menu and select "***Detail***":



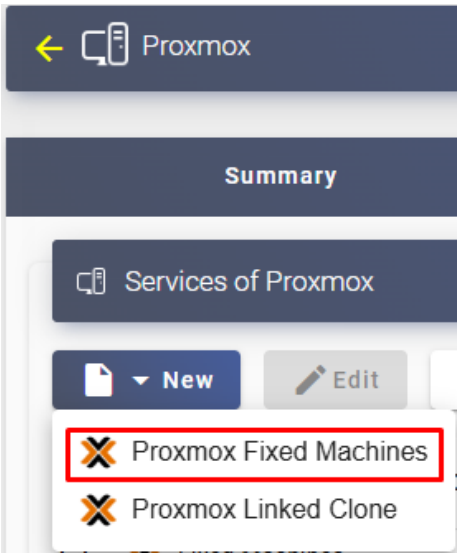
We will have the following types of basic services:



3.1.5.2 Configure service based on "Proxmox Fixed Machines"

This type of service will allow users to connect directly to existing virtual machines on a server or cluster in the Proxmox environment.

To create base services of type "***Proxmox Fixed Machines***" we go to the "***Service Providers***", select the "***Services***" tab, click on "***New***" and select "***Proxmox Fixed Machines***".



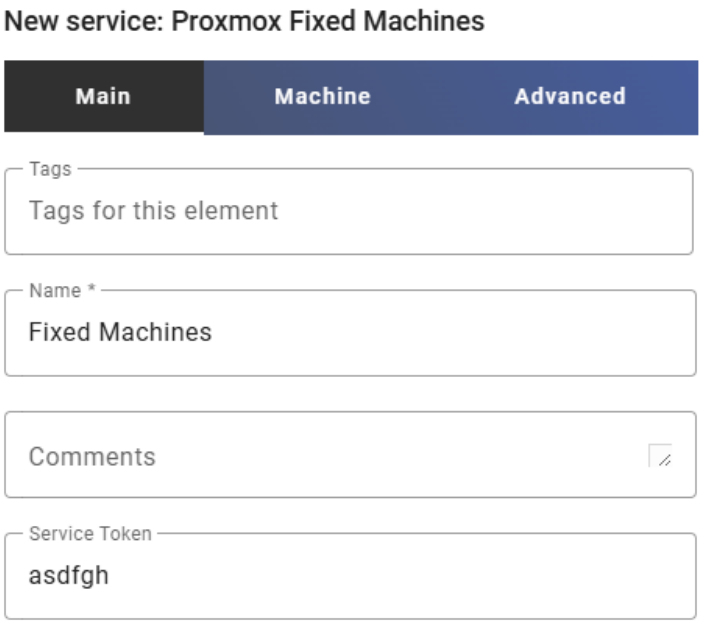
The minimum parameters to be configured in this type of service are:

- Main:

Name: Name of the service.

Service Token: If UDS is required to recognize the login and ***logout*** of a user accessing a service of this type (so that when a ***logout*** is made, UDS automatically releases the machine and makes it available to another user), it will be necessary to fill in this field with alphanumeric characters. If the field is left empty, UDS will permanently assign (until an administrator manually releases the service) the machine to a user, as it will have no way to control when a user enters or exits the machine.

NOTE: If the "**Service Token**" field is filled in, on the machines that are selected for the connection it will be necessary to have the UDS "**UDSActorUnmanagedSetup-x.x.x.exe**" actor installed and in the configuration of this, indicate those same characters in the "**Service Token" field**".

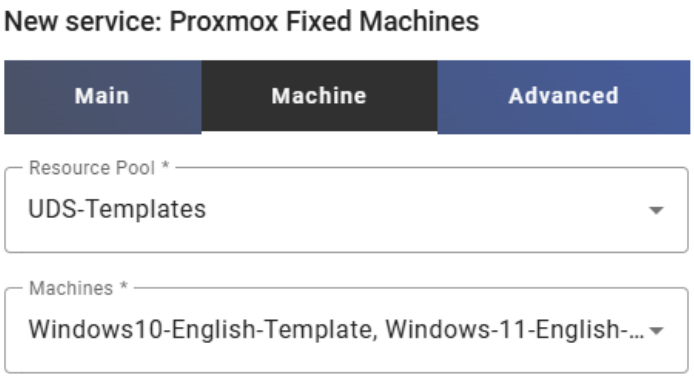


- Machine:

Resource Pool: Pool of resources in the Proxmox environment where the virtual machines that we want to manage and assign to users are located. Machines must be assigned to this folder to be visible

Machines: Virtual machines existing in the pool of resources indicated above that will be managed by UDS and to which access can be provided to the different users. We will have to indicate which machines will be accessible by users, the machines not selected, will be ignored and will not be assigned.

If the machines have the unamanaged actor installed and a "token" has been configured both in the actor (machine) and in the service definition (administration), it will be possible to control the access of users to these machines and they can be released so that other users can use them, in case they do NOT have the unmanaged actor installed, When a machine is assigned to a user, that assignment will be maintained until manually deleted by an administrator.



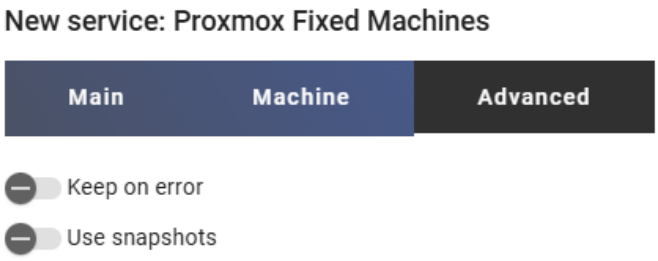
- Advanced

Randomize machine assignment: This parameter indicates how the order of assignments to the users will be, if it is active (by default) the order of assignment will be random, that is, when a user requests a machine, it will be unknown which one will be assigned. If the parameter is not active, the assignment order will be sequential, the first user will be given the first machine indicated in the list, the second the second machine and so on.

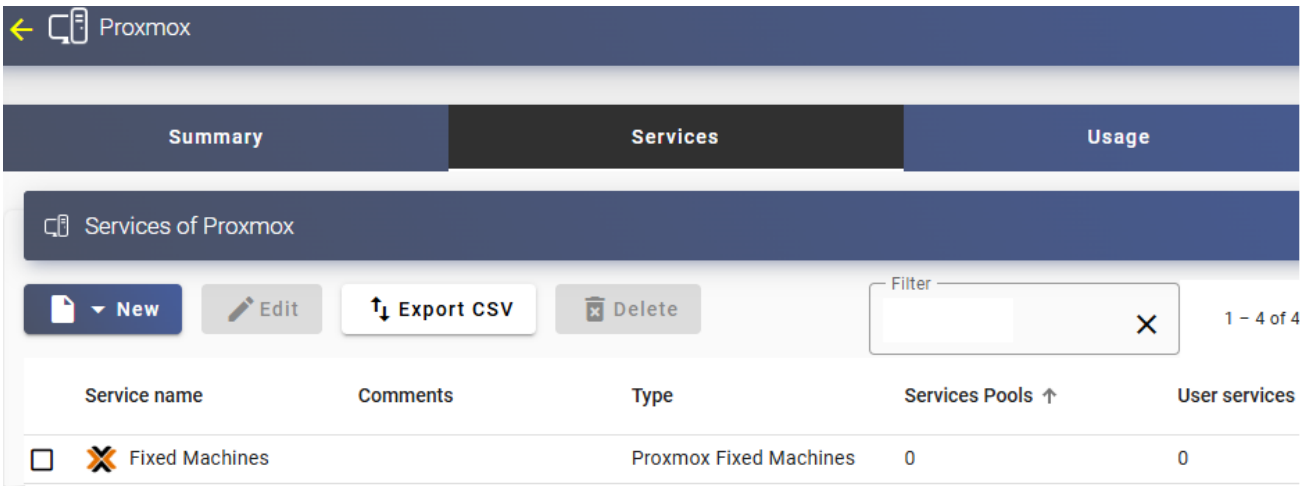
Keep on error: When activated, the system ignores all service access errors, allowing the machine to remain available for access (useful for occasions when machines may not be available for a short period of time but should be available at some point).

Use Snapshots: If this parameter is enabled, UDS will create a snapshot on the virtual machine before the user's connection and once its assignment is removed (through an automatic detection of the user's logout or manual deletion of an administrator) the snapshot will be recovered to leave the machine in the same state just before the user's connection. If this parameter is not activated, UDS will not recover any previous state of the machine when the assignment with a user is deleted.

NOTE: If the **"Use Snapshots" option is enabled** and the virtual machine already has a snapshot, UDS will not take any additional snapshots and will use and recover the existing snapshot.



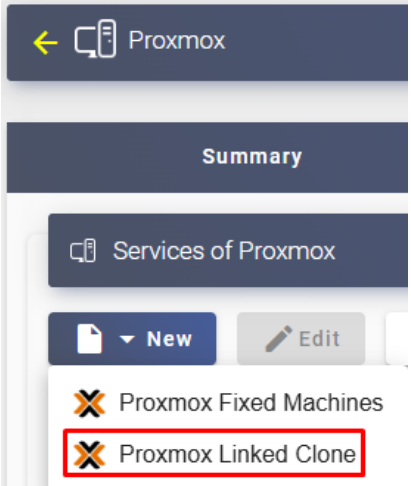
Save the configuration and we will have a valid "***Proxmox Fixed Machines***" on the Proxmox platform. We will be able to register all the "***Proxmox Fixed Machines***" type services that we need.



3.1.5.3 Configure service based on "Proxmox Linked Clone"

This type of service will create, within the virtual infrastructure, virtual desktops dependent on active publishing.

To create base services of type "***Proxmox Linked Clone***" we access the "***Service Providers***", select the "***Services***" tab, click on "***New***" and select "***Proxmox Linked Clone***".



The minimum parameters to be configured in this type of service are:

- Main:

Name: Name of the service.

Pool: Pool that will contain the machines created by UDS.

HA: Enables generated machines to use "***HA Group***".

- Machine:

Base Names: Root of the name of all virtual desktops to be deployed in this service. (e.g.: Machine Names= win11-).

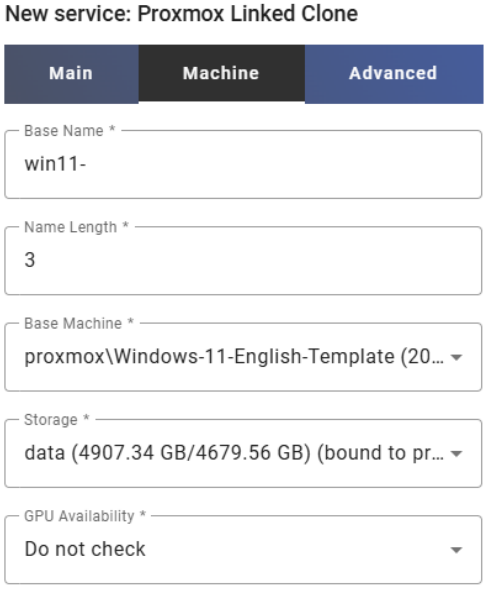
Name Length: Number of digits of the counter attached to the root of the desktop name (e.g.: Name Length= 3. The final name of the generated desktops would be: win11-000**,** win11-001... win11-999).

Base Machine: Template virtual machine (Gold Image) used by the UDS system for the deployment of virtual desktops.

Storage: The location where the generated publications and virtual desktops will be stored. **Storages that allow the creation of "Snapshots" are supported (and will be shown). For more information visit the following URL: https://pve.proxmox.com/wiki/Storage**.

GPU Availability: It will allow you to check if hosts have vGPUs to be able to host VMs:

- Do not check: Hosts will not be checked if they have vGPUs, machines will be created on any host.
- Only if available: The hosts will be checked if they have vGPUs. New VMs created will be hosted only on hosts **WITH** vGPUs
- Only if NOT available: The hosts will be checked if they have vGPUs. New VMs created will be hosted only on hosts **WITHOUT** vGPUs



- Advanced:

Remove found duplicates: If this option is enabled (default), when UDS finds a virtual desktop with the same name (generated by UDS, but not registered) as the desktop you are about to create, it will delete it (it is recommended to use it to avoid "lost" VMs in the inventory).

Keep on error: When activated, the system ignores all errors in accessing the service, thus preventing desktop deletion. Very useful and recommended for use when using persistent machines, preventing the desktop from being deleted by the system under virtually any circumstances (except if an administrator manually deletes it).

Try SOFT Shutdown first: When activated, the system will attempt to perform a "Soft" shutdown on the OS of the virtual desktop. It is necessary that the tools are installed and running on the machine to be deleted, otherwise, after a period of time a forced shutdown will be performed.

Put back to cache: Parameter that defines whether machines will be returned to the cache, so that they can be used by another user, once they are freed:

- **No. Never put it back to cache:** When a user logs out of the desktop and the machine is non-persistent, it will be destroyed and will NOT be returned to the cache to be reused by more users.

- **Yes, try to put it back to cache:** When a user logs out of the desktop and the machine is non-persistent, it will be returned IF it will be returned to the cache to be reused by more users.

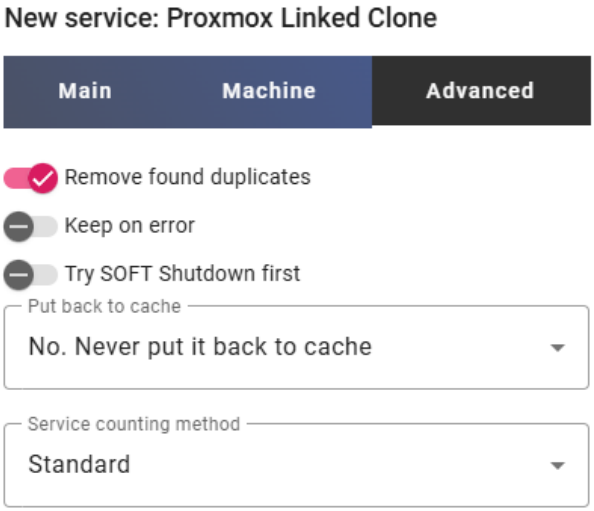
Service counting method: Method that the system will use to perform the machine count to comply with the parameter "maximum number of services to be provided" of a Service Pool:

- **Standard:** This counting mode only takes into account the machines that are being created and during the process of creating and destroying machines, it is possible to exceed the value of the parameter "maximum number of services to be provided" of a Service Pool.

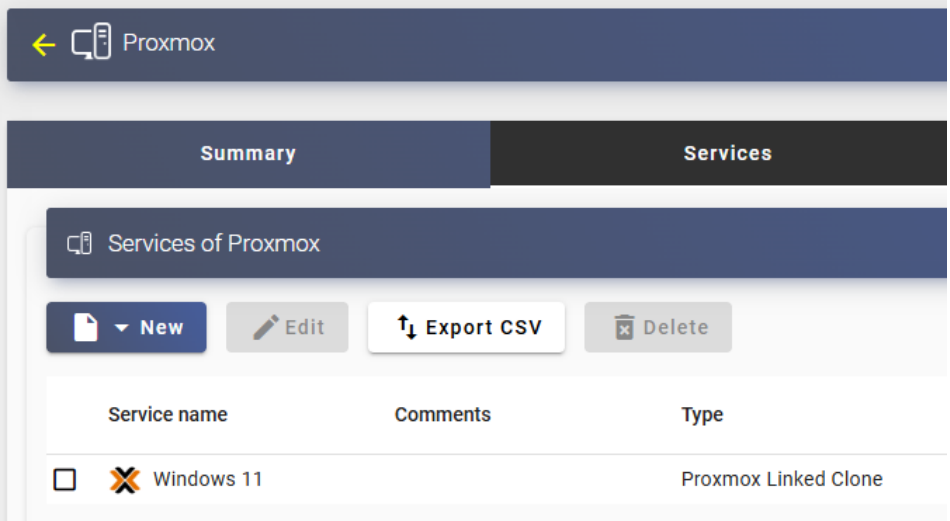
  For example, if a service pool, whose "maximum number of services to be provided" parameter is set to 5, we delete a service, there will be a time when that pool contains 6 machines: 4 configured, 1 being created and 1 being deleted.

- **Conservative:** This counting mode takes into account the machines that are being created and those that are being deleted, at no time will it be possible to exceed the value of the parameter "maximum number of services to be provided" of a Service Pool.

  For example, if a service pool, whose parameter of "maximum number of services to be provided" is set to 5, we delete a service, until that service is completely eliminated the system will not generate a new service, avoiding exceeding the value of 5 existing machines in the pool.



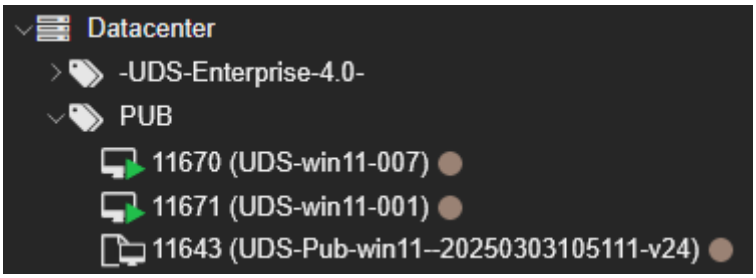
Save the configuration and we will have a valid "***Proxmox Linked Clone***" on the Proxmox platform. We will be able to register all the services of the "***Proxmox Linked Clone***" type that we need.



Once we have the entire UDS environment configured (Services, Authenticators, OS Managers and Transports) and created the first "***Service Pool***", we will be able to observe in the Proxmox environment how the desktop deployments are carried out.

The first task that will be performed will be to generate a publication of the template (this machine will be generated every time we publish a service). It will be a clone of the template chosen when registering for the service, with a disk size and characteristics equal to said template.

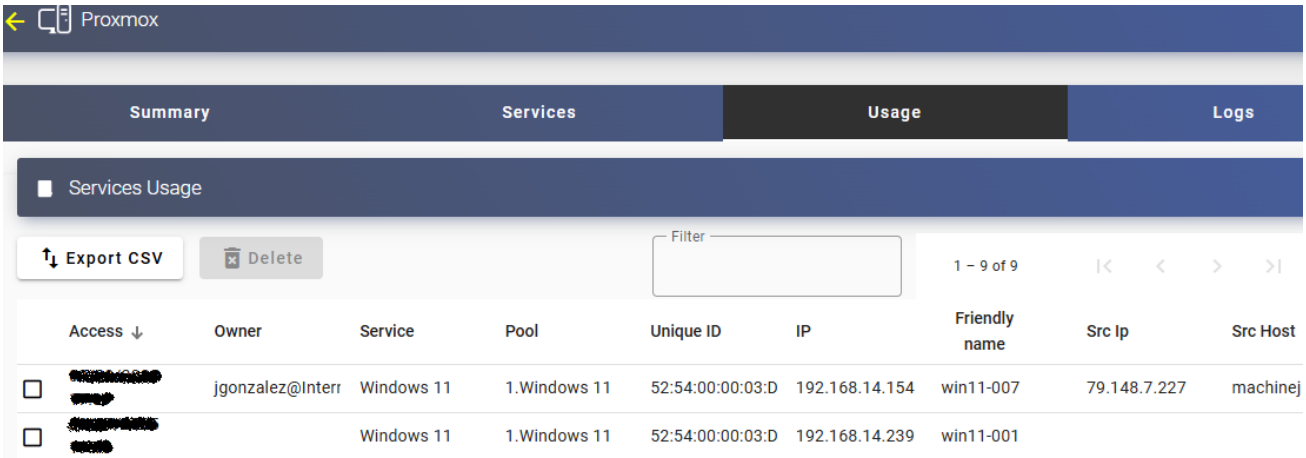
Once the process of creating the publication is finished (the system will name it as: "UDS-Pub-Base_Name-date-vnúmero_publicación") the desktops are automatically created (the system names them as: "UDS-Machine_Name-Name_Length") based on the cache parameters configured in the "***Service Pools***".



3.1.5.4 Service usage summary and error logs

Within a "***Service Provider***" we will have a tab called "***Usage***" that will allow us to have a quick view with detailed information and perform basic actions of the services deployed in this "***Service Provider***".

To access this information on the use of the services, we access the "***Service Providers***" and select the "***Usage" tab.***":



We will be able to visualize:

Access: Date and time the service was accessed.

Owner: User and authenticator assigned to the service in usuario@autenticador format. If it appears empty, it means that the service has not been assigned to any user and is available in the cache.

Service: Name of the base service belonging to the "***Service Provider***" from which the service was generated.

Pool: Name of the "***Service Pools***" to which the service belongs.

Unique ID: MAC address of the service.

IP: IP address of the service.

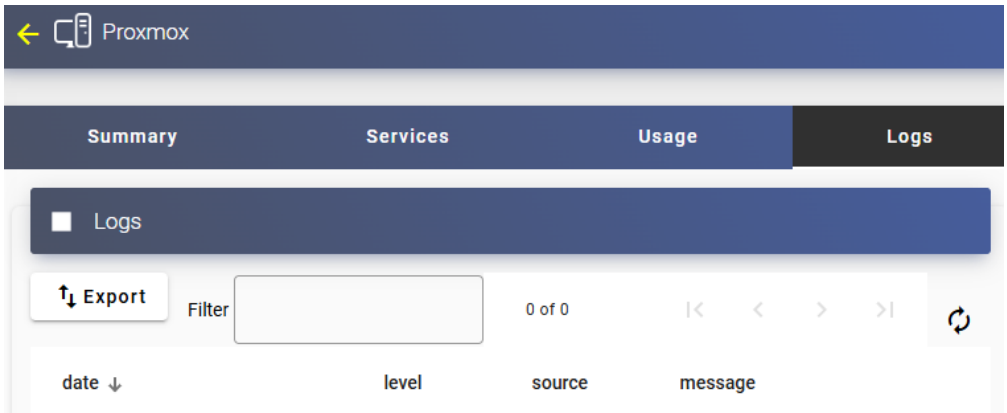
Friendly name: Name of the service. It will also be the DNS name of the UDS auto-generated virtual machine.

Src Ip: IP address of the connecting client accessing the service.

Src Host: DNS name of the connecting client accessing the service. If you are unable to access this name, your IP address will be indicated.

It will also be possible to select one or more services to proceed with their deletion (in the base service of type "***Fixed*** " we will only delete the assignment with the user).

The "***Logs***" tab will show us information about a possible problem that occurred in the "***Service Provider***":

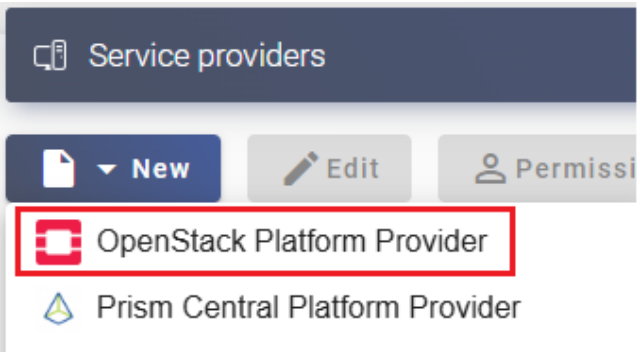


3.1.6 VDI environment with OpenStack

Deployment of self-generated virtual desktops by UDS or connection to existing machines, in an OpenStack virtualization infrastructure.

3.1.6.1 Registration of an "OpenStack Platform Provider" service provider"

Click on "***New***" and select "***OpenStack Platform Provider***".



In an "***OpenStack Platform Provider***" the minimum parameters to be configured are:

- Main:

Name: Name to identify this provider.

Identity endpoint: Identity service address

Authentication method: The type of authentication used.

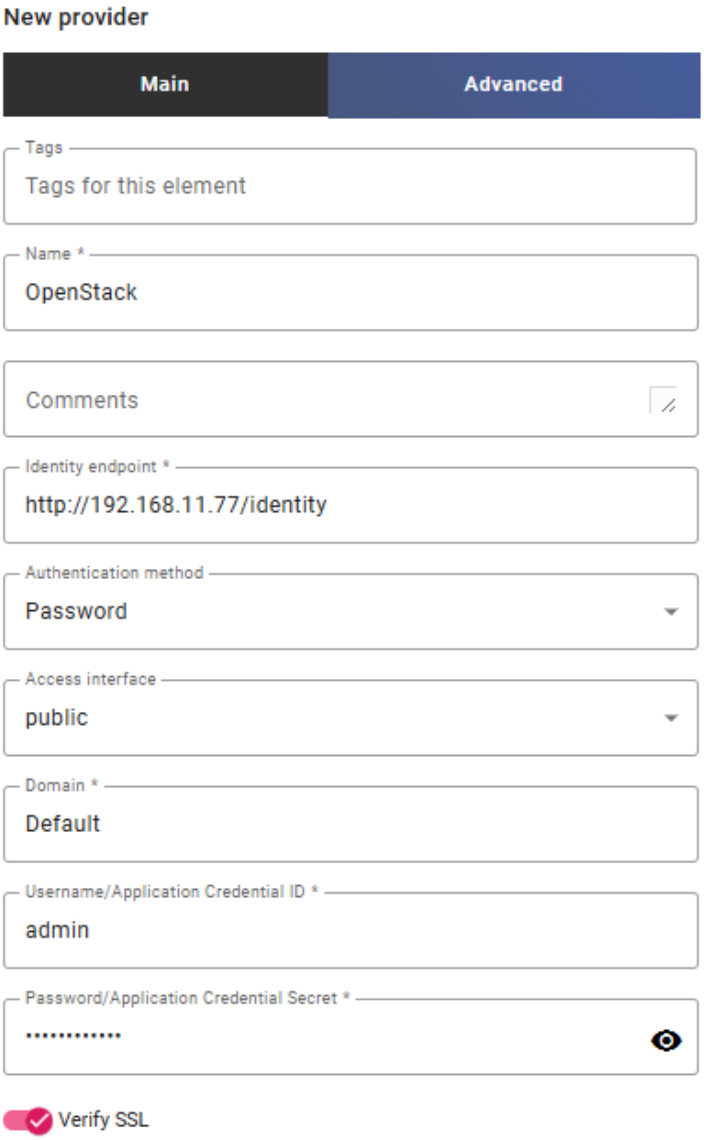
Access interface: Name of the access interface.

Domain: Domain used in the integration

Username: Username or ID with permissions for UDS integration with the provider.

Password: Password of the user used in the previous field.

Safe Operating Procedure: Passwords must be of sufficient length and include upper and lower case, numbers and special characters.



- Advanced:

Project ID: ID of the project that will be used in this provider.

Region: Region used in integration.

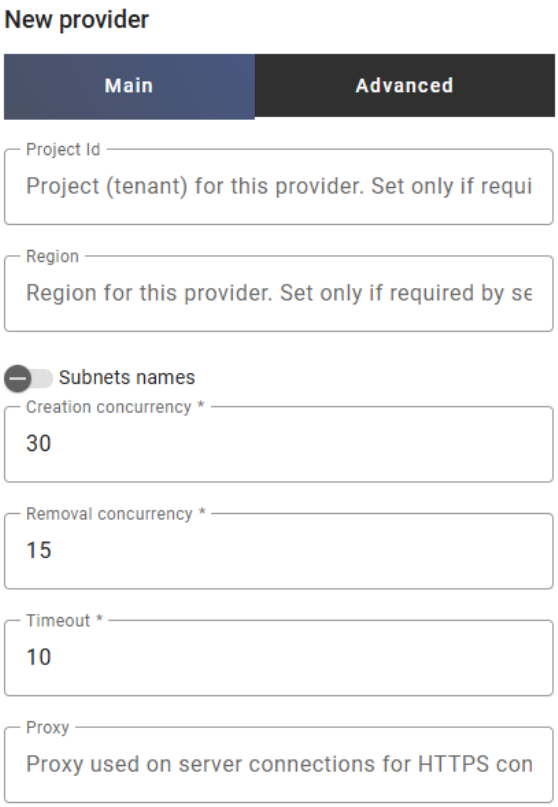
Subnet names: If enabled, an attempt will be made to retrieve the name of the networks instead of their ID.

Creation concurrency. Maximum number of concurrent desktop creation tasks, in the different service pools that use it, for this provider.

Removal concurrency. Maximum number of concurrent desktop deletion tasks, in the different service pools that use it, for this provider.

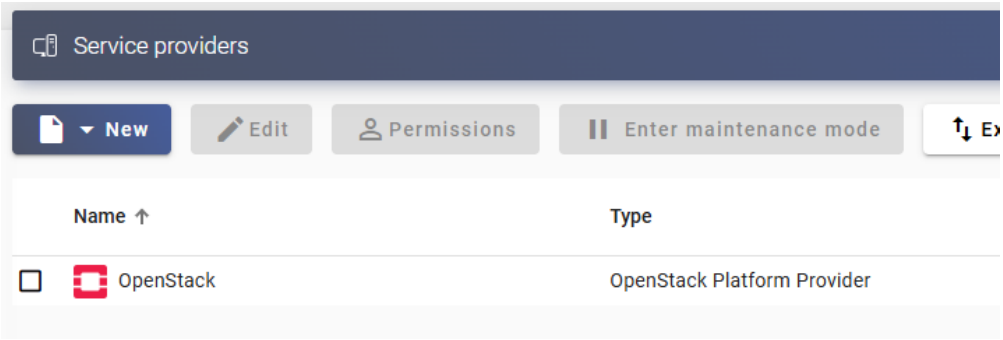
Timeout. The time of "*Timeout*" on the connection to the server.

Proxy: Proxy address used in the connection to the provider.



Using the "*Test*" button we will check that the connection is successful.

Save the configuration and we will have a valid "*Service Providers*" to start registering base services on the OpenStack platform.

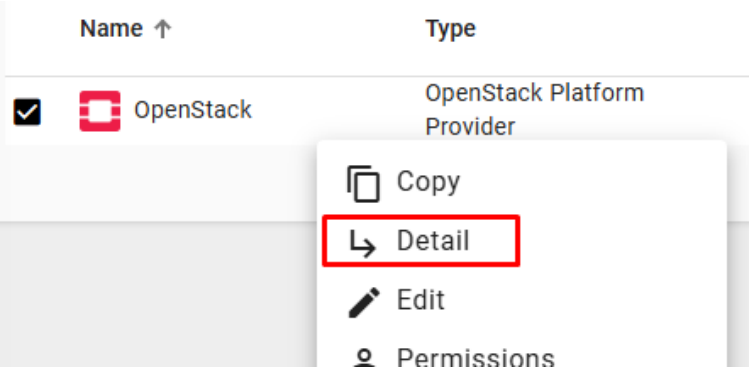


NOTE: If we have several OpenStack platforms, we will be able to register all the "Service Providers" of the "OpenStack Platform Provider" type that we need.

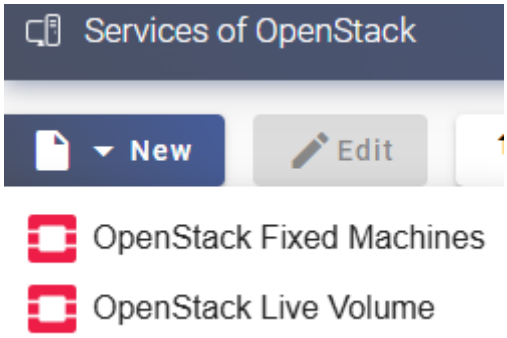
To modify a parameter in an existing *"Service Providers*", select it and click on "*Edit*".

By clicking on the "***Enter Maintenance Mode***" button, you can pause all operations produced by the UDS server on a service provider. It is recommended to put a service provider in maintenance cases where communication with that service provider has been lost or a maintenance shutdown is planned.

Once the OpenStack platform is integrated into UDS, base services can be created. To do this, we double-click on the service provider created or on the provider's menu and select "***Detail***":



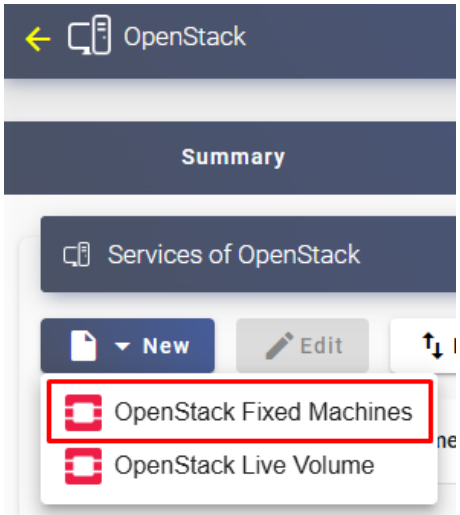
Dispondremos de los siguientes tipos de servicios base:



3.1.6.2 Configure OpenStack Fixed Machines-based service”

This type of service will allow users to connect directly to existing virtual machines on a server or cluster in the Openstack environment.

To create base services of type "***Openstack Fixed Machines***" we access the "***Service Providers***", select the "***Services***" tab, click on "***New***" and select "***Openstack Fixed Machines***”.



The minimum parameters to be configured in this type of service are:

- Main:

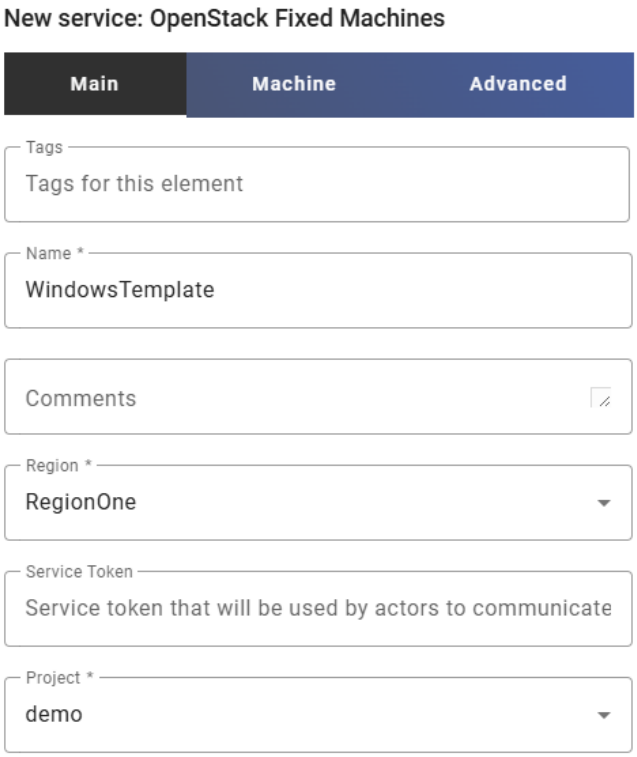
Name: Name of the service.

Region: Region used in integration.

Service Token: If UDS is required to recognize the ***login*** and ***logout*** of a user accessing a service of this type (so that when a ***logout*** is made, UDS automatically releases the machine and makes it available to another user), it will be necessary to fill in this field with alphanumeric characters. If the field is left empty, UDS will permanently assign (until an administrator manually releases the service) the machine to a user, as it will have no way to control when a user enters or exits the machine.

NOTE: If the "**Service Token**" field is filled in, on the machines that are selected for the connection it will be necessary to have the UDS "**UDSActorUnmanagedSetup-x.x.x.exe**" actor installed and in the configuration of this, indicate those same characters in the "**Service Token" field**”.

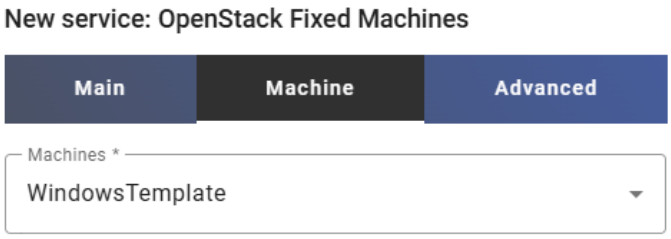
Project: Project used in the integration and from which the VMs to be used will be obtained.



- Machine:

Machines: Virtual machines existing in the project indicated above that will be managed by UDS and to which access can be provided to the different users. We will have to indicate which machines will be accessible by users, the machines not selected, will be ignored and will not be assigned.

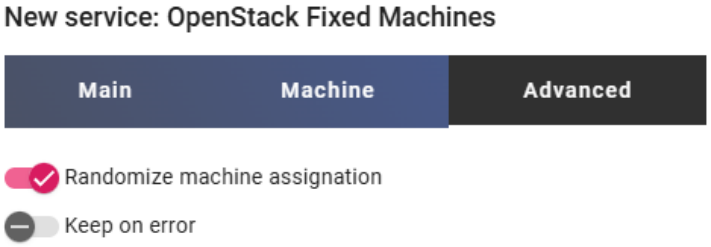
If the machines have the unamanaged actor installed and a "token" has been configured both in the actor (machine) and in the service definition (administration), it will be possible to control the access of users to these machines and they can be released so that other users can use them, in case they do NOT have the unmanaged actor installed, When a machine is assigned to a user, that assignment will be maintained until manually deleted by an administrator.



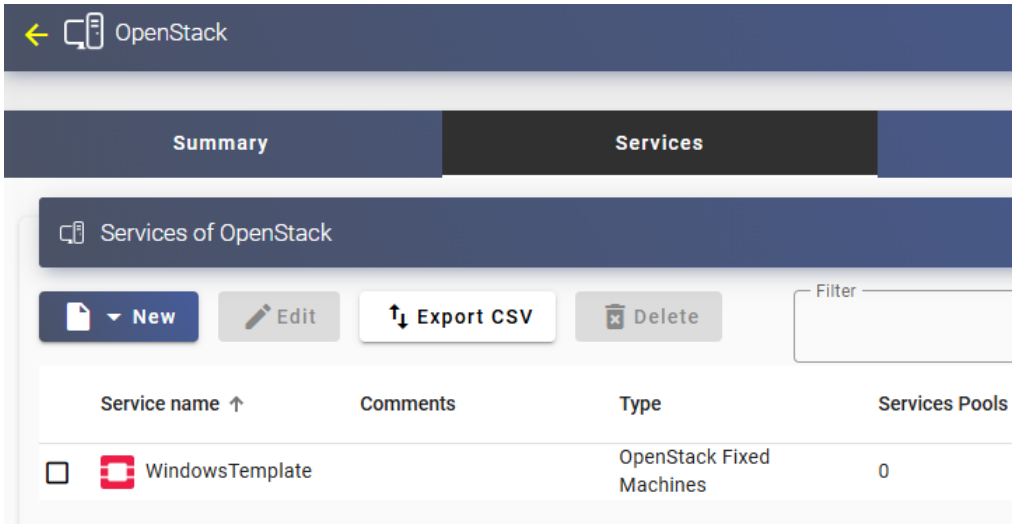
- Advanced

Randomize machine assignation: This parameter indicates how the order of assignment of the machines to the users will be, if it is active (by default) the order of assignment will be random, that is, when a user requests a machine, it will be unknown which one will be assigned. If the parameter is not active, the assignment order will be sequential, the first user will be given the first machine indicated in the list, the second the second machine and so on.

Keep on error: When activated, the system ignores all service access errors, allowing the machine to remain available for access (useful for occasions when machines may not be available for a short period of time but should be available at some point).).



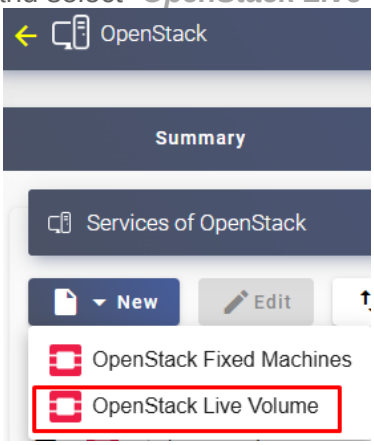
Save the configuration and we will have a valid "***OpenStack Fixed Machines***" on the OpenStack platform. We will be able to register all the " ***OpenStack Fixed Machines***" services that we need.



3.1.6.3 Configure service based on “OpenStack Live Volume”

This type of service will create, within the virtual infrastructure, virtual desktops dependent on active publishing.

To create base services of type "***OpenStack Live Volume***" we go to the "***Service Providers***", select the "***Services***" tab, click on "***New***" and select "***OpenStack Live Volume***”.



The minimum parameters to be configured in this type of service are:

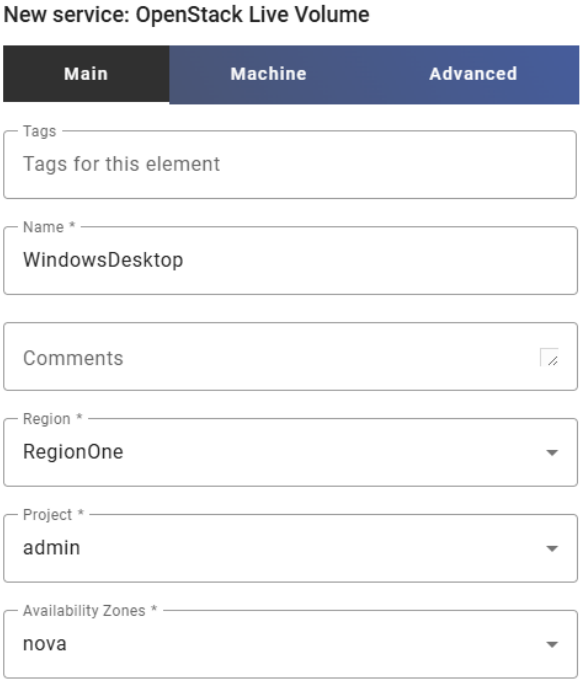
- Main:

Name: Name of the service.

Region: Region used in integration.

Project: Project used in the integration and from which the volumes to be used will be obtained.

Availability Zones: Zone in which desktops will be deployed.



- Machine:

Volume: The volume used to build the virtual desktop instances.

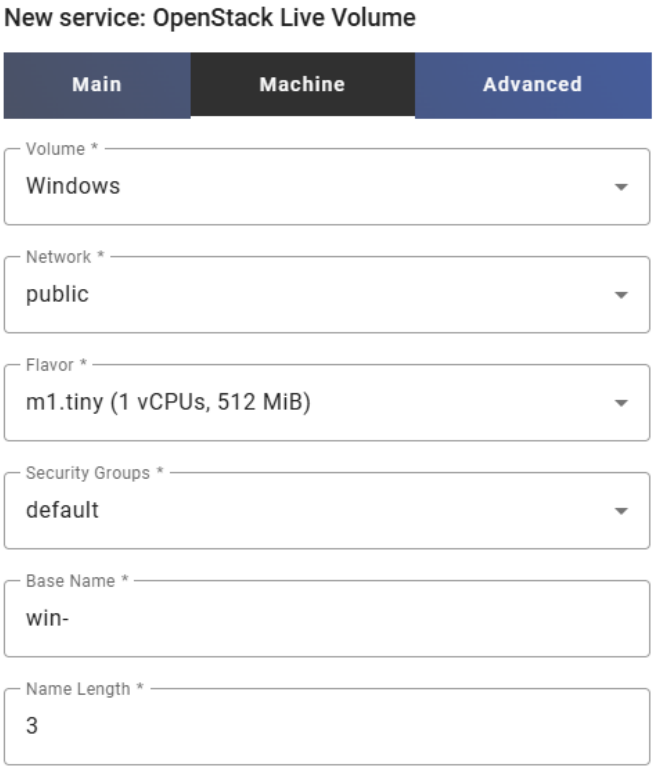
Network: Network to which instances connect.

Flavor: Machine configuration applied at the resource level.

Security Groups: Security groups that the instances will join.

Base Names: Root of the name of all instances to be deployed in this service. (e.g.: Machine Names= win-).

Name Length: Number of digits of the counter attached to the root of the desktop name (e.g.: Name Length= 3. The final name of the generated desktops would be: **win-000,** win-001... **win-999**).



- Advanced:

Remove found duplicates: If this option is enabled (default), when UDS finds a virtual desktop with the same name (generated by UDS, but not registered) as the desktop you are about to create, it will delete it (it is recommended to use it to avoid "lost" VMs in the inventory).

Keep on error: When activated, the system ignores all errors in accessing the service, thus preventing desktop deletion. Very useful and recommended for use when using persistent machines, preventing the desktop from being deleted by the system under virtually any circumstances (except if an administrator manually deletes it).

Try SOFT Shutdown first: Cuando se activa, el sistema intentará realizar un apagado “Soft” en el S.O. del escritorio virtual. Es necesario que las tools estén instaladas y en ejecución en la máquina a eliminar, en caso contrario, pasado un periodo de tiempo se realizará un apagado forzado.

Put back to cache: A parameter that defines whether machines will be returned to the cache, so that they can be used by another user, once they are released:

- **No. Never put it back to cache:** When a user logs out of the desktop and the machine is non-persistent, it will be destroyed and will NOT be returned to the cache to be reused by more users.

- **Yes, try to put it back to cache:** When a user logs out of the desktop and the machine is non-persistent, it will be returned. returned to the cache to be reused by more users.

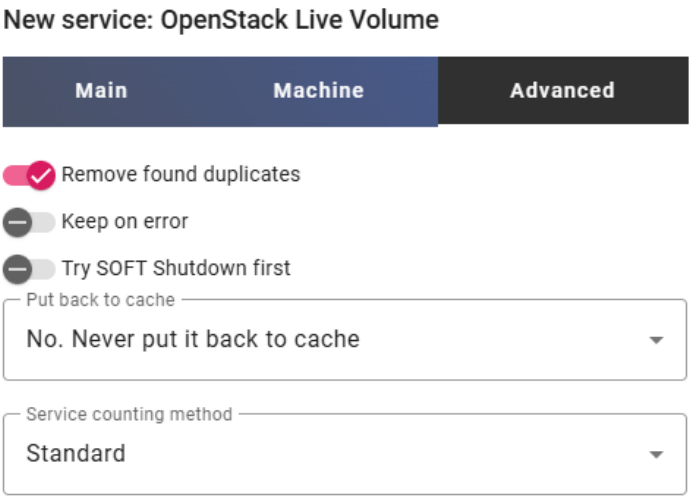
Service counting method: Method that the system will be used to perform count to comply with the parameter "maximum number of services to be provided" of a Service Pool:

- **Standard:** This counting mode only takes into account the machines that are being created and during the process of creating and destroying machines, it is possible to exceed the value of the parameter "maximum number of services to be provided" of a Service Pool.

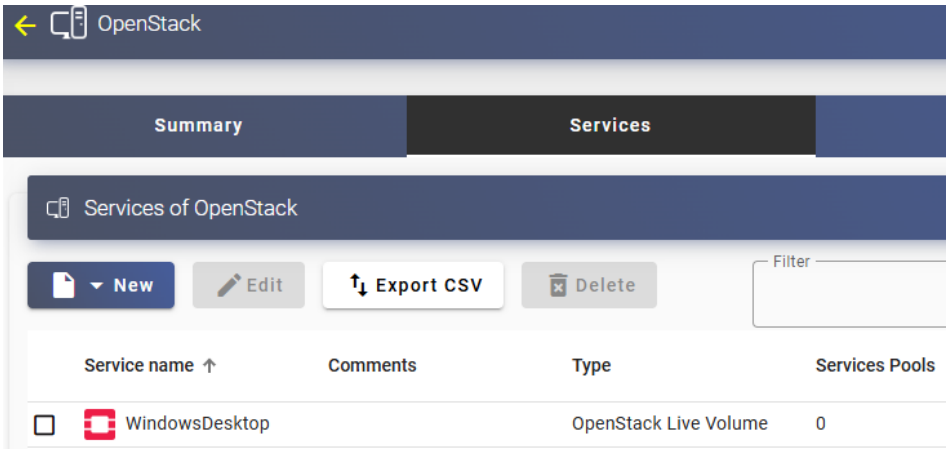
  For example, if a service pool, whose "maximum number of services to be provided" parameter is set to 5, we delete a service, there will be a time when that pool contains 6 machines: 4 configured, 1 being created and 1 being deleted.

- **Conservative:** This counting mode takes into account the machines that are being created and those that are being deleted, at no time will it be possible to exceed the value of the parameter "maximum number of services to be provided" of a Service Pool.

  For example, if a service pool, whose parameter of "maximum number of services to be provided" is set to 5, we delete a service, until that service is completely eliminated the system will not generate a new service, avoiding exceeding the value of 5 existing machines in the pool.



Save the configuration and we will have a valid " ***OpenStack Live Volume*** " on the OpenStack platform. We will be able to register all the services of the type " ***OpenStack Live Volume*** " that we need.



Once we have the entire UDS environment configured (Services, Authenticators, OS Managers and Transports) and created the first "***Service Pool***", we will be able to observe in the OpenStack environment how the desktop deployments are carried out.

The first task that will be performed will be to generate a publication of the template (this machine will be generated every time we publish a service). It will be a clone of the template chosen when registering for the service, with a disk size and characteristics equal to said template.

Once the process of creating the publication using a snapshot is finished (the system will name it as: "UDS-Pub-Base_Name-date-vnúmero_publicación") the desktops are automatically created (the system names them as: "UDS-Machine_Name-Name_Length") based on the cache parameters configured in the "***Service Pools'***".

Mostrando 3 articulos

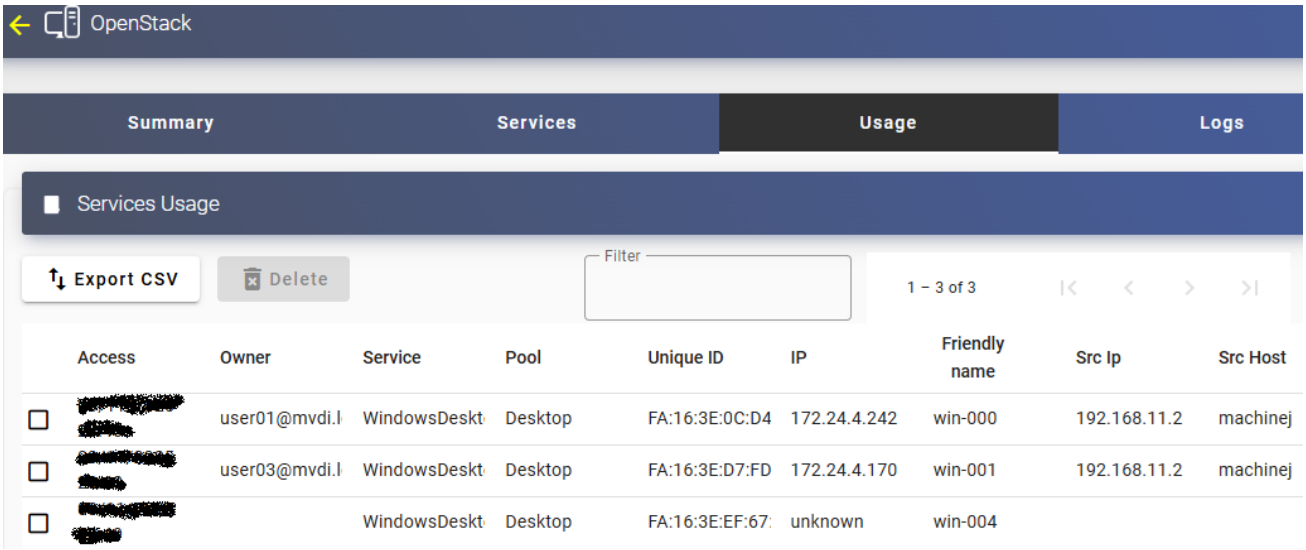
Instance Name	IP Address	Flavor	Key Pair	Status
UDS_win-002	172.24.4.163, 2001:db8::1fc	m1.tiny	-	Activa
UDS_win-001	172.24.4.170, 2001:db8::b	m1.tiny	-	Activa
UDS_win-000	172.24.4.242, 2001:db8::72	m1.tiny	-	Activa

Mostrando 3 articulos

3.1.6.4 Service usage summary and error logs

Within a "***Service Provider***" we will have a tab called "***Usage***" that will allow us to have a quick view with detailed information and perform basic actions of the services deployed in this "***Service Provider***".

To access this information on the use of the services, we access the "***Service Providers***" and select the "***Usage" tab.***":



We will be able to visualize:

Access: Date and time the service was accessed.

Owner: User and authenticator assigned to the service in usuario@autenticador format. If it appears empty, it means that the service has not been assigned to any user and is available in the cache.

Service: Name of the base service belonging to the "***Service Provider***" from which the service was generated.

Pool: Name of the "***Service Pools***" to which the service belongs.

Unique ID: MAC address of the service.

IP: IP address of the service.

Friendly name: Name of the service. It will also be the DNS name of the UDS auto-generated virtual machine.

Src Ip: IP address of the connecting client accessing the service.

Src Host: DNS name of the connecting client accessing the service. If you cannot access this name, your IP address will be indicated.

También será posible seleccionar uno o varios servicios para proceder a su eliminación (en el servicio base de tipo "***Fixed*** " únicamente eliminaremos la asignación con el usuario).

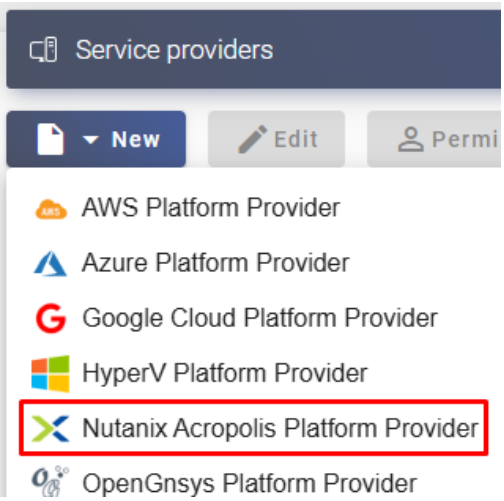
The "***Logs***" tab will show us information about a possible problem that occurred in the "***Service Provider'"***":

3.1.7 VDI with Nutanix Acropolis

Deploying and connecting to virtual desktops on a Nutanix AHV virtualization infrastructure.

3.1.7.1 Service provider registration “Nutanix Acropolis Platform Provider”

Click on "***New***" and select "***Nutanix Acropolis Platform Provider***”.



In a "Nutanix Acropolis Provider" the minimum parameters to be configured are:

- Main:

Name: Name to identify this provider.

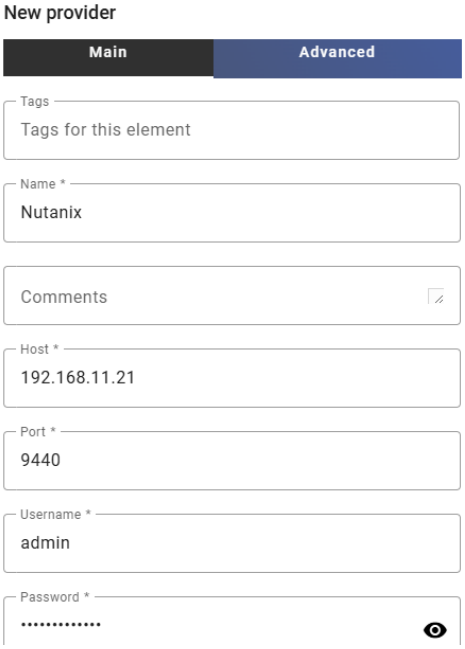
Host: IP address or name of the Nutanix Prims Element server.

Port: Communication port with Nutanix Prims server (default 9440)

Username: Username with permissions for UDS integration with the provider. It is recommended that you use a user with admin permissions (use a user created specifically for this purpose).

Password: Password of the user used in the previous field.

Safe use procedure: Passwords must be of sufficient length and include upper and lower case, numbers and special characters.



- Advanced:

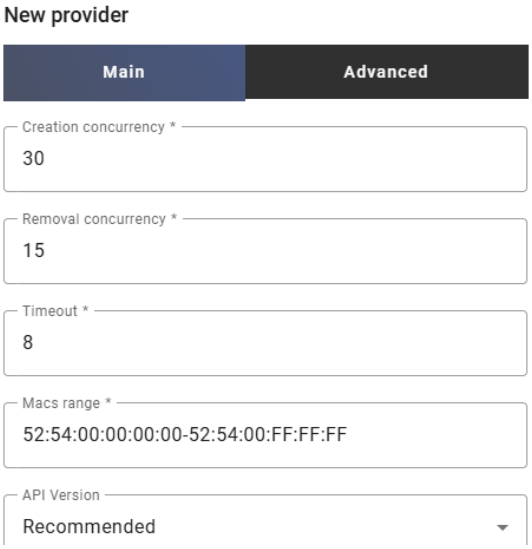
Creation concurrency. Maximum number of concurrent desktop creation tasks, in the different service pools that use it, for this provider.

Removal concurrency. Maximum number of concurrent desktop deletion tasks, in the different service pools that use it, for this provider.

Timeout. The time of "***Timeout***" on the connection to the server.

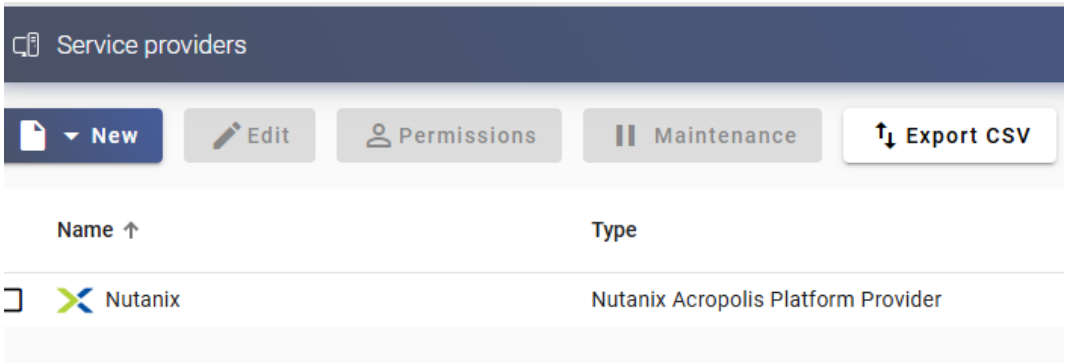
Macs range. MAC address range used by UDS to assign to automatically created virtual desktops. The modification of the range is always allowed to higher values, which will be applied immediately, but if it is reduced, it will be necessary that there is no record and it will also be necessary to wait a reasonable time of at least 24 hours for the records to be released and the change to be applied (to reduce the range, the registration of a new provider is recommended).

API Version: API to use.



Using the "*Test*" button we will check that the connection is successful.

Save the configuration and we will have a valid *"Service Providers*" to start registering base services on the Nutanix AHV platform.

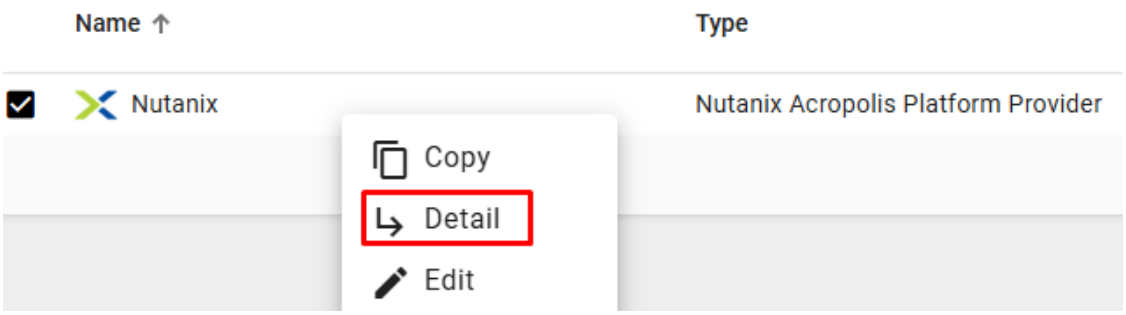


NOTE: If we have several Nutanix AHV platforms, we will be able to register all the "Service Providers" of the "Nutanix Acropolis Provider" type that we need.

To modify a parameter in an existing *"Service Providers*", select it and click on "*Edit*".

By clicking on the "*Enter Maintenance Mode*" button, you can pause all operations produced by the UDS server on a service provider. It is recommended to put a service provider in maintenance cases where communication with that service provider has been lost or a maintenance shutdown is planned.

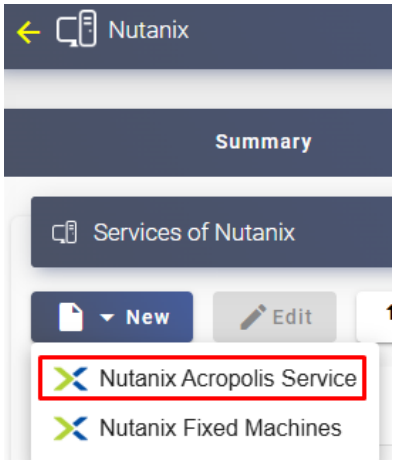
Once the Nutanix AHV platform is integrated into UDS, base services can be created. To do this, we double-click on the service provider created or in the provider's menu we select "*Detail*":



3.1.7.2 Configure service based on "Nutanix Acropolis Service"

This type of service will create, within the virtual infrastructure, virtual desktops dependent on active publishing.

To create base services of type "***Nutanix Acropolis Service***" we access the "***Service Providers***", select the "***Services***" tab, click on "***New***" and select "***Nutanix Acropolis Service***".



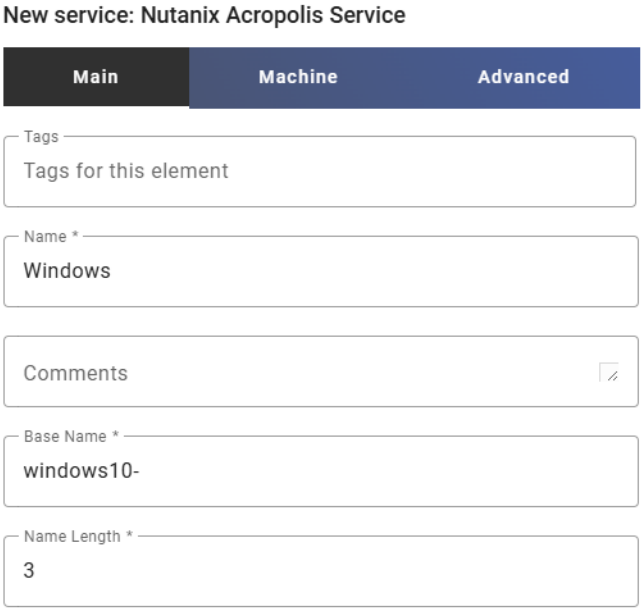
The minimum parameters to be configured in this type of service are:

- Main:

Name: Name of the service.

Machine Names: Root of the name of all virtual desktops to be deployed in this service. (e.g.: Machine Names= Windows10-).

Base Name: Number of digits of the counter attached to the root of the desktop name (e.g.: Name Length= 3. The final name of the generated desktops would be: Windows10-000**,** Windows10-001... Windows10-999).



- Machine:

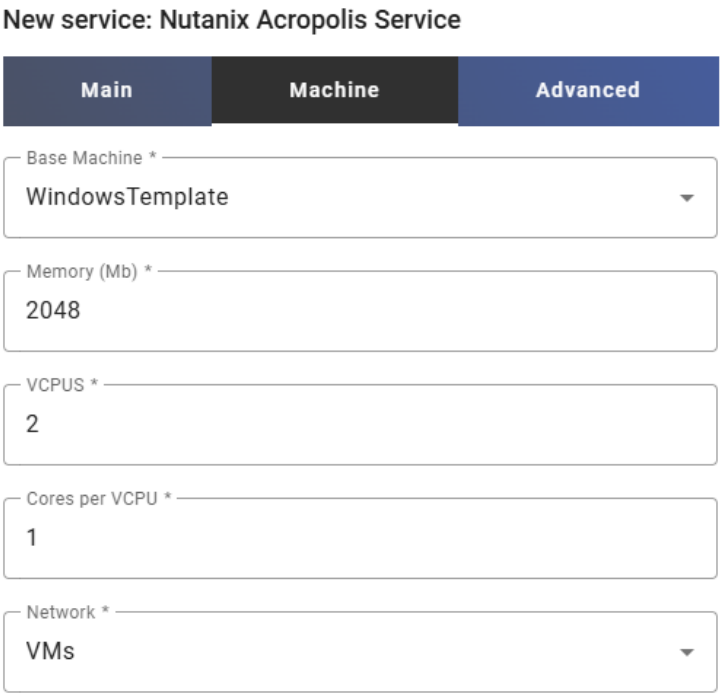
Base Machine: Template virtual machine (Gold Image) used by the UDS system for the deployment of virtual desktops.

Memory: Amount of memory in Mb that will be allocated to the generated virtual desktops.

VCPUS: Number of vCPUs that the generated desktops will have (the value -1 will use the template configuration)

Cores per VCPU: Number of cores per vCPUs that the generated desktops will have (the value -1 will use the template configuration)

Network: Network where virtual desktops will be connected.



- Advanced:

Remove found duplicates: If this option is enabled (default), when UDS finds a virtual desktop with the same name (and generated by UDS) as the desktop you are about to create, it will delete it (it is recommended to use it to avoid "lost" VMs).

Keep on error: When activated, the system ignores all errors in accessing the service, thus preventing desktop deletion. Very useful and recommended for use when using persistent machines, preventing the desktop from being deleted by the system under virtually any circumstances (except if an administrator manually deletes it).

Reset type: Type of desktop reboot that is applied when reboot is indicated from user mode.

Try SOFT Shutdown first: When activated, the system will attempt to perform a "Soft" shutdown on the OS of the virtual desktop. It is necessary that the tools are installed and running on the machine to be deleted, otherwise, after a period of time a forced shutdown will be performed.

Put back to cache: Parameter that defines whether machines will be returned to the cache, so that they can be used by another user, once they are freed:

- **No. Never put it back to cache:** When a user logs out of the desktop and the machine is non-persistent, it will be destroyed and will NOT be returned to the cache to be reused by more users.

- **Yes, try to put it back to cache:** When a user logs out of the desktop and the machine is non-persistent, it will be returned IF it will be returned to the cache to be reused by more users.

Service counting method: Method that the system will use to perform the machine count to comply with the parameter "maximum number of services to be provided" of a Service Pool:

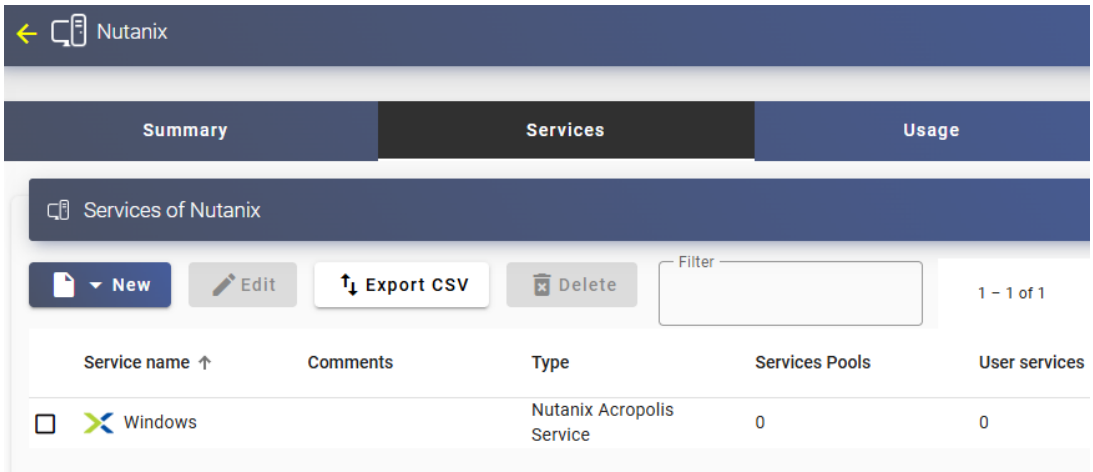
- **Standard:** This counting mode only takes into account the machines that are being created and during the process of creating and destroying machines, it is possible to exceed the value of the parameter "maximum number of services to be provided" of a Service Pool.

  For example, if a service pool, whose "maximum number of services to be provided" parameter is set to 5, we delete a service, there will be a time when that pool contains 6 machines: 4 configured, 1 being created and 1 being deleted.

- **Conservative:** This counting mode takes into account the machines that are being created and those that are being deleted, at no time will it be possible to exceed the value of the parameter "maximum number of services to be provided" of a Service Pool.

  For example, if a service pool, whose parameter of "maximum number of services to be provided" is set to 5, we delete a service, until that service is completely eliminated the system will not generate a new service, avoiding exceeding the value of 5 existing machines in the pool.

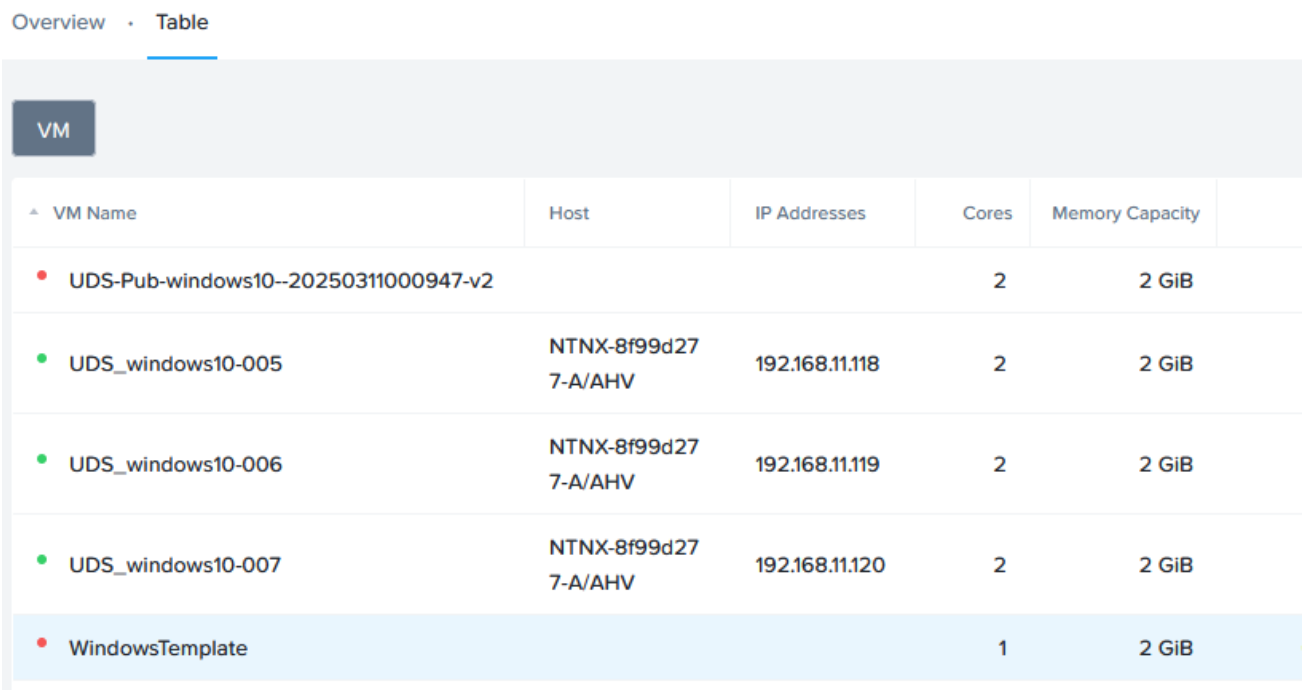
Salvamos la configuración y dispondremos de un "Nutanix Acropolis Service" válido en la plataforma Nutanix. Podremos dar de alta todos los "Nutanix Acropolis Service" que necesitemos.



Once we have the entire UDS environment configured (Services, Authenticators, OS Managers and Transports) and created the first "***Service Pool***", we will be able to observe in the Nutanix Prism web console how the desktop deployments are carried out.

The first task that will be performed will be to generate a publication of the template (this machine will be generated every time we publish a service). It will be a clone of the template chosen when registering for the service, with a disk size and characteristics equal to said template.

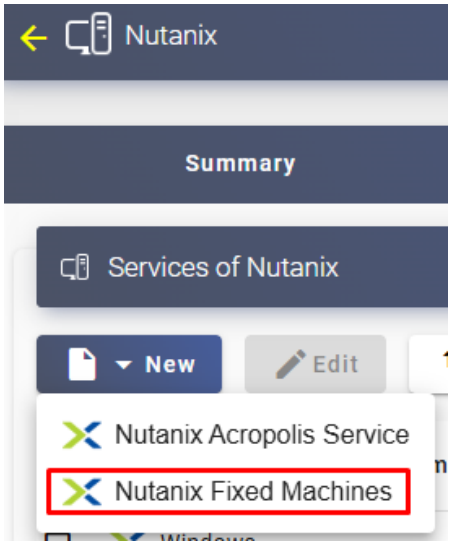
Once the process of creating the publication is finished (the system will name it as: "UDS-Pub-Base_Name-date-vnúmero_publicación") the desktops are automatically created (the system names them as: "UDSS-número_publicación Machine_Name+*Name_Length*") based on the cache parameters configured in the "***Service Pools***".



3.1.7.3 Configure service based on "Nutanix Fixed Machines"

This type of service will allow users to connect directly to existing virtual machines on a server or cluster in the Nutanix environment.

To create base services of type "***Nutanix Fixed Machines***" we access the "***Service Providers***", select the "***Services***" tab, click on "***New***" and select "***Nutanix Fixed Machines***".



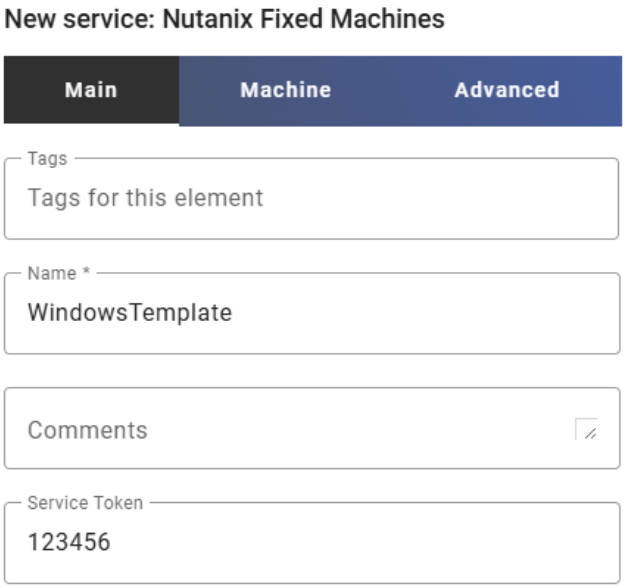
The minimum parameters to be configured in this type of service are:

- Main:

Name: Name of the service.

Service Token: If UDS is required to recognize the login and ***logout*** of a user accessing a service of this type (so that when a ***logout*** is made, UDS automatically releases the machine and makes it available to another user), it will be necessary to fill in this field with alphanumeric characters. If the field is left empty, UDS will permanently assign (until an administrator manually releases the service) the machine to a user, as it will have no way to control when a user enters or exits the machine.

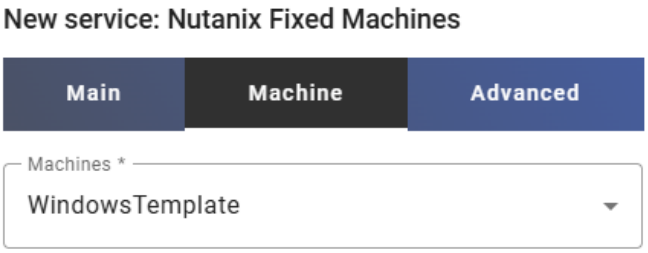
NOTE: If the "**Service Token**" field is filled in, on the machines that are selected for the connection it will be necessary to have the UDS "**UDSActorUnmanagedSetup-x.x.x.exe**" actor installed and in the configuration of this, indicate those same characters in the "**Service Token" field**".



- Machine:

Machines: Existing virtual machines that will be managed by UDS and to which access can be provided to the different users. We will have to indicate which machines will be accessible by users, the machines not selected, will be ignored and will not be assigned.

If the machines have the unamanaged actor installed and a "token" has been configured both in the actor (machine) and in the service definition (administration), it will be possible to control the access of users to these machines and they can be released so that other users can use them, in case they do NOT have the unmanaged actor installed, When a machine is assigned to a user, that assignment will be maintained until manually deleted by an administrator.



- Advanced

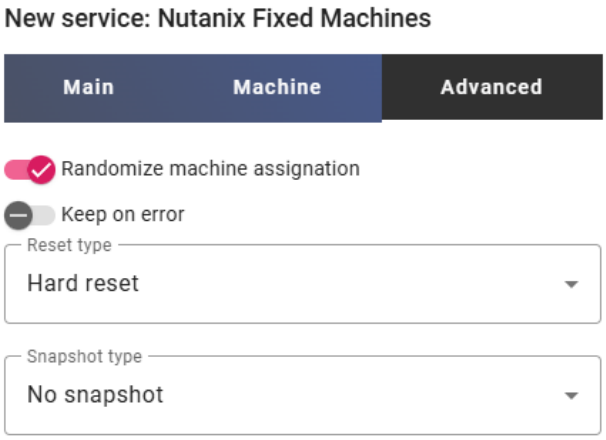
Randomize machine assignment: This parameter indicates how the order of assignment of the machines to the users will be, if it is active (by default) the order of assignment will be random, that is, when a user requests a machine, it will be unknown which one will be assigned. If the parameter is not active, the assignment order will be sequential, the first user will be given the first machine indicated in the list, the second the second machine and so on.

Keep on error: When activated, the system ignores all service access errors, allowing the machine to remain available for access (useful for occasions when machines may not be available for a short period of time but should be available at some point).

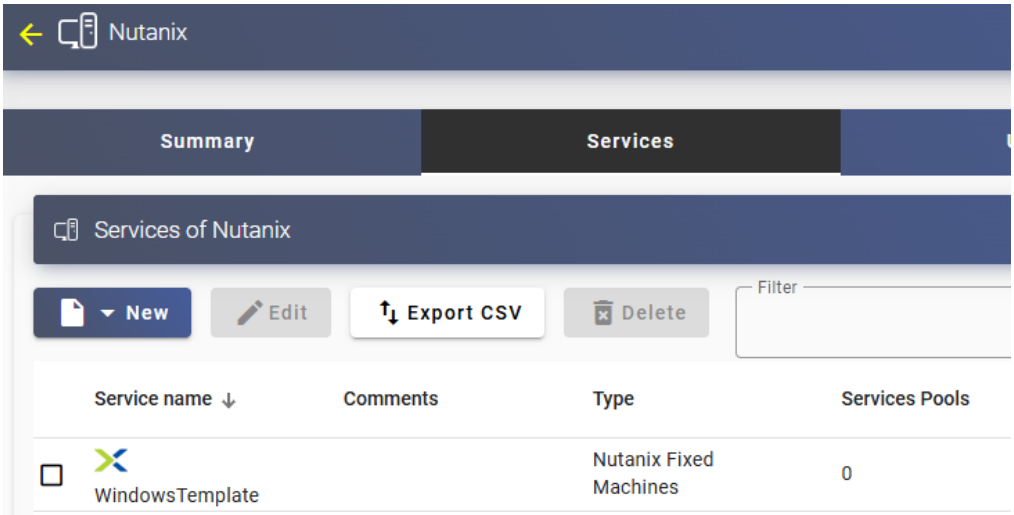
Reset type: Type of desktop reboot that is applied when reboot is indicated from user mode

Use Snapshots: If this parameter is enabled, UDS will create a snapshot on the virtual machine before the user's connection and once its assignment is removed (through an automatic detection of the user's logout or manual deletion of an administrator) the snapshot will be recovered to leave the machine in the same state just before the user's connection. If this parameter is not activated, UDS will not recover any previous state of the machine when the assignment with a user is deleted.

NOTE: If the "**Use Snapshots" option is enabled** and the virtual machine already has a snapshot, UDS will not take any additional snapshots and will use and recover the existing snapshot.



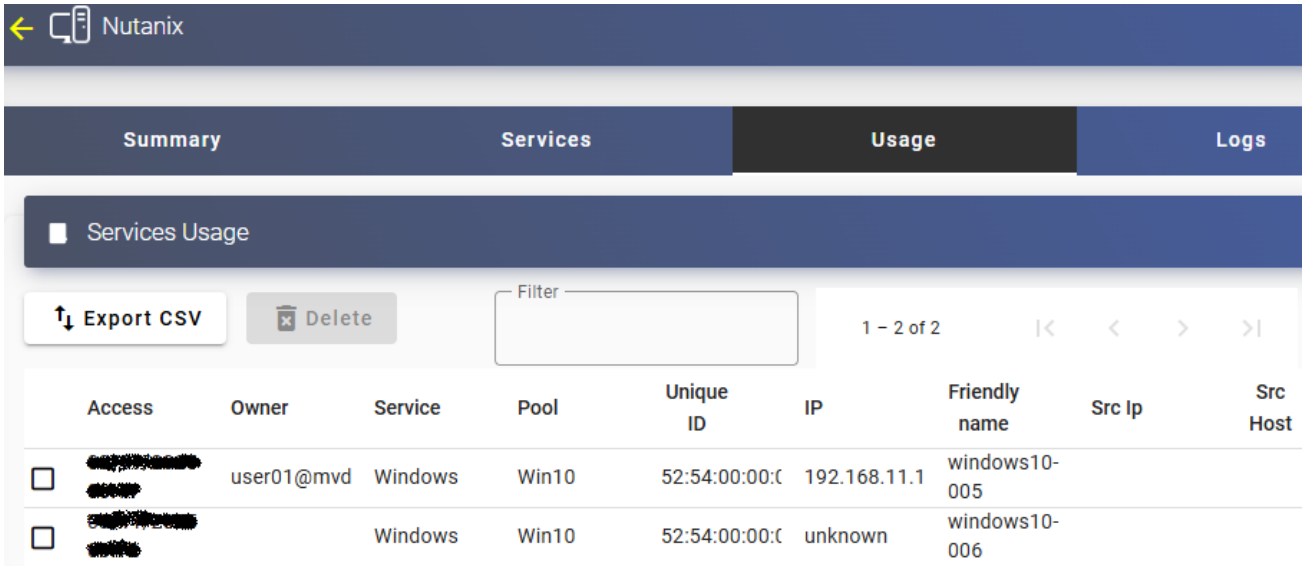
Save the configuration and we will have a valid " ***Nutanix Fixed Machines*** " on the Nutanix platform. We will be able to register all the "***Nutanix Fixed Machines***" services that we need.



3.1.7.4 Service usage summary and error logs

Within a "***Service Provider***" we will have a tab called "***Usage***" that will allow us to have a quick view with detailed information and perform basic actions of the services deployed in this "***Service Provider***".

To access this information on the use of the services, we access the "***Service Providers***" and select the "***Usage" tab.***":



We will be able to visualize:

Access: Date and time the service was accessed.

Owner: User and authenticator assigned to the service in usuario@autenticador format. If it appears empty, it means that the service has not been assigned to any user and is available in the cache.

Service: Name of the base service belonging to the "***Service Provider***" from which the service was generated.

Pool: Name of the "***Service Pools***" to which the service belongs.

Unique ID: MAC address of the service.

IP: IP address of the service.

Friendly name: Name of the service. It will also be the DNS name of the UDS auto-generated virtual machine.

Src Ip: IP address of the connecting client accessing the service.

Src Host: DNS name of the connecting client accessing the service. If you are unable to access this name, your IP address will be indicated.

It will also be possible to select one or more services to proceed with its deletion.

The "***Logs***" tab will show us information about a possible problem that occurred in the "***Service Provider'***":

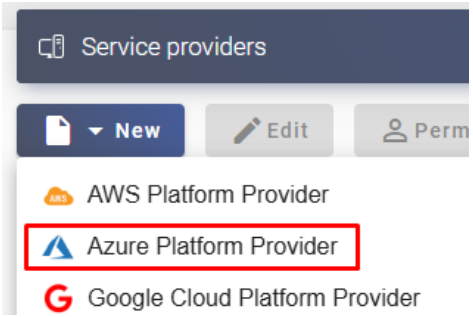
3.1.8 VDI with Microsoft Azure

Deployment of self-generated virtual desktops by UDS or connection to existing machines, in a virtualization infrastructure in the Microsoft Azure cloud.

3.1.8.1 Service provider registration “Azure Platform Provider”

The integration of UDS Enterprise with an Azure platform requires a series of previous tasks and requirements to be carried out in the Azure environment itself, so it is advisable to consult the manual "***VDI with UDS Enterprise and Microsoft Azure***" located in the documentation section of the UDS Enterprise website.

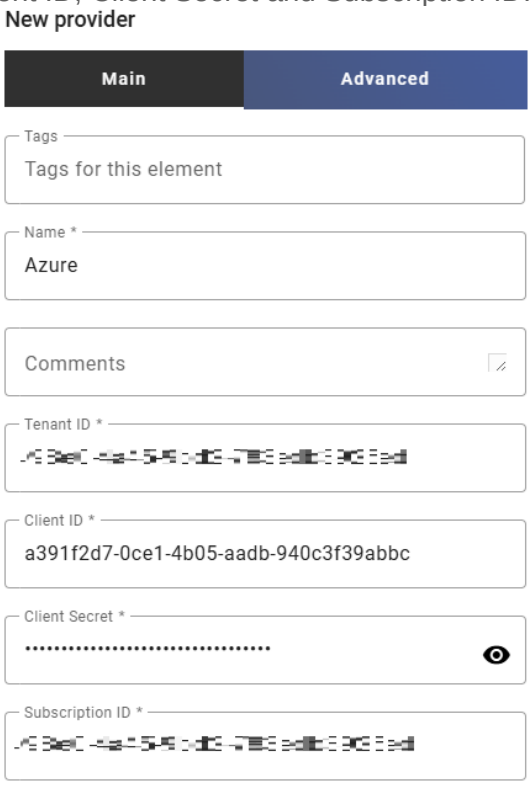
Click on "***New***" and select "***Azure Platform Provider***”.



In an "***Azure Platform Provider***" the minimum parameters to be configured are:

- Main:

Service Name, Tenant ID, Client ID, Client Secret and Subscription ID.



NOTE: To locate and generate all the necessary data requested, you can consult the manual "VDI with UDS Enterprise and Microsoft Azure" located in the documentation section of the UDS Enterprise website.

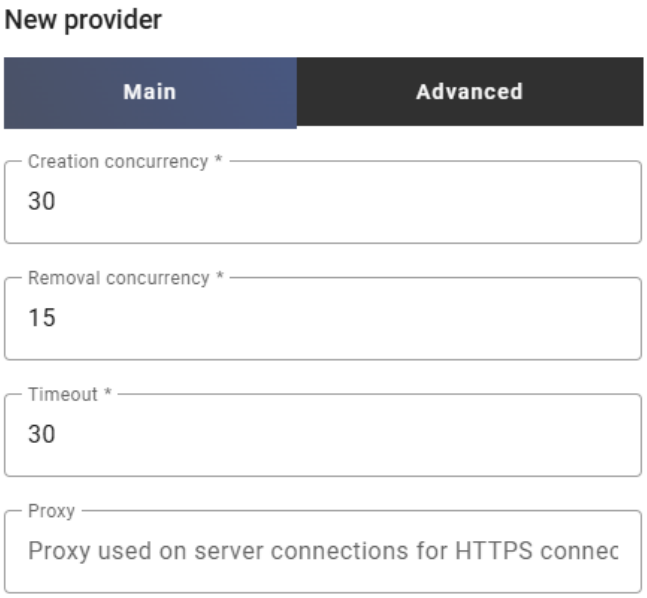
- Advanced:

Creation concurrency. Maximum number of concurrent desktop creation tasks, in the different service pools that use it, for this provider.

Removal concurrency. Maximum number of concurrent desktop deletion tasks, in the different service pools that use it, for this provider.

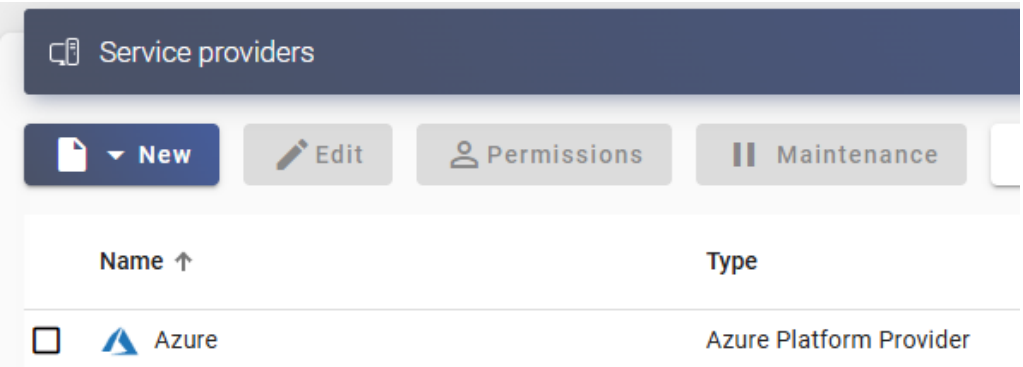
Timeout. The time of "***Timeout***" on the connection to the server.

Proxy: Proxy address used in the connection to the provider.



Using the "***Test***" button we will check that the connection is successful.

Save the configuration and we will have a valid "***Service Provider***" to start registering base services on the Microsoft Azure platform.

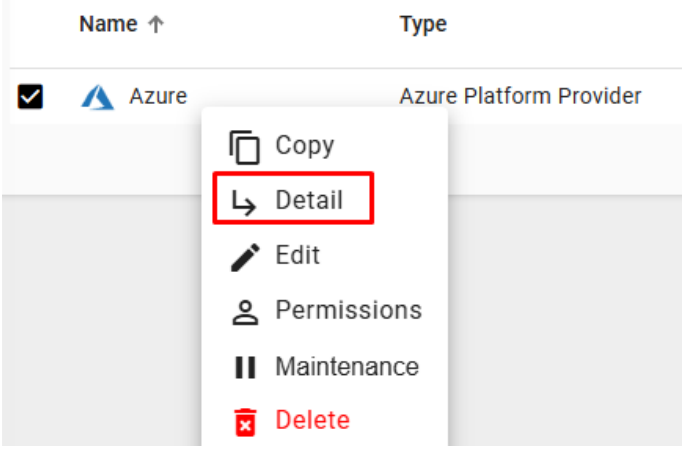


NOTE: If we have several subscriptions in the Microsoft Azure cloud, we will be able to register all the "Service Providers" of the "Azure Platform Provider" type that we need.

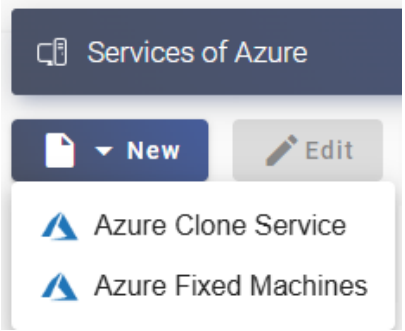
To modify a parameter in an existing ***"Service Provider***", select it and click on "***Edit***".

By clicking on the "***Enter Maintenance Mode***" button, you can pause all operations produced by the UDS server on a service provider. It is recommended to put a service provider in maintenance cases where communication with that service provider has been lost or a maintenance shutdown is planned.

Once the Azure platform is integrated into UDS, base services can be created. To do this, we double-click on the service provider created or in the provider's menu we select "***Detail***":



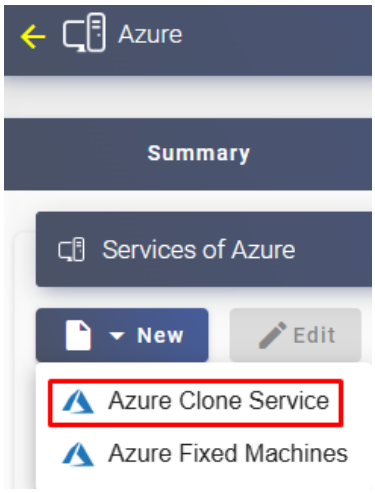
We will have the following types of basic services:



3.1.8.2 Configure service based on “Azure Clone Service”

This type of service will create, within the Azure subscription, virtual desktops dependent on active publishing.

To create base services of type "***Azure Clone Service***" we access the "***Service Provider***", select the "***Services***" tab, click on "***New***" and select "***Azure Clone Service***”.



The minimum parameters to be configured in this type of service are:

- Main:

Name: Name of the service.

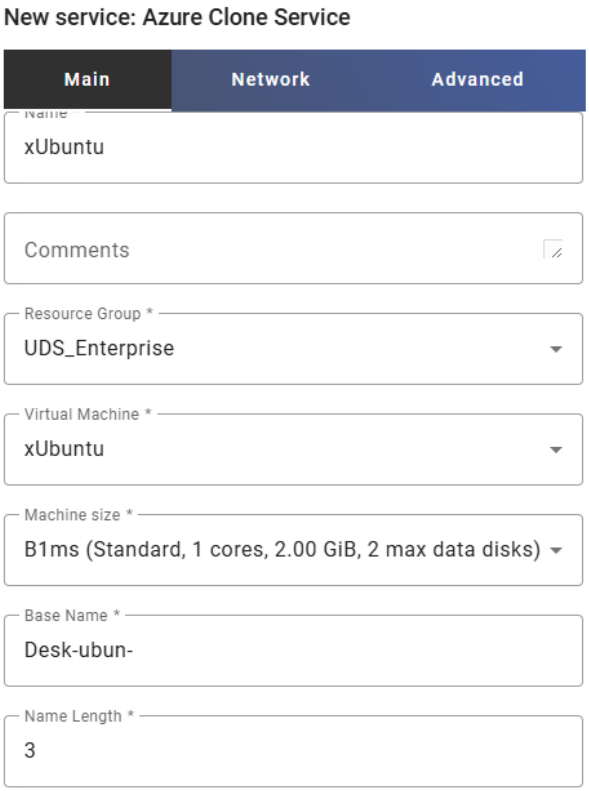
Resource Group: A group of existing resources on the Azure platform where the template machine must be hosted and on which the virtual desktops will be deployed.

Virtual Machine: Template virtual machine (Gold Image) used by the system for the deployment of virtual desktops.

Machine Size: Amount of resources that the virtual desktops will have automatically deployed by UDS (this list will show all the types of machines available in Azure. Therefore, we must make sure that the chosen type is supported by our Azure subscription).

Machine Names: Root of the name of all virtual desktops to be deployed in this service. (e.g.: Machine Names= Desk-ubun-).

Name Length: Number of digits of the counter attached to the root of the desktop name (e.g.: Name Length= 3. The final name of the generated desktops would be: Desk-ubun-000**,** Desk-ubun-001... Desk-ubun-999).

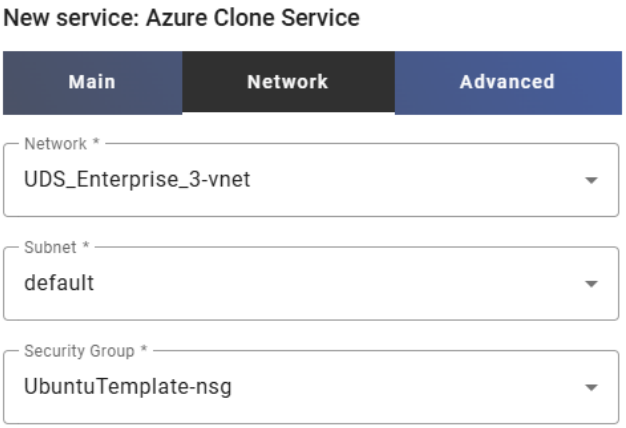


- Network:

Network: Existing virtual network of the Azure environment to which the virtual desktops will be connected (there must be connectivity with the UDS Server component).

Subnet: Existing subnet in the Azure environment to which the virtual desktops will connect.

Security Group: Security group (if necessary) to assign to the new virtual desktops.



- Advanced:

Caching policy: Configuring the Disk Cache.

Accelerated network: Enables the use of this technology (cannot be used with most types of machines, only with: D/DSv3, E/ESv3, Fsv2 and Ms/Mms and Linux OS).

Pricing tier: Level of redundancy applied.

Keep on error: When activated, the system ignores all errors in accessing the service, thus preventing desktop deletion. Very useful and recommended for use when using persistent machines, preventing the desktop from being deleted by the system under virtually any circumstances (except if an administrator manually deletes it).

Try SOFT Shutdown first: When activated, the system will attempt to perform a "Soft" shutdown on the OS of the virtual desktop. It is necessary that the tools are installed and running on the machine to be deleted, otherwise, after a period of time a forced shutdown will be performed.

Put back to cache: Parameter that defines whether machines will be returned to the cache, so that they can be used by another user, once they are freed:

- **No. Never put it back to cache:** When a user logs out of the desktop and the machine is non-persistent, it will be destroyed and will NOT be returned to the cache to be reused by more users.

- **Yes, try to put it back to cache:** When a user logs out of the desktop and the machine is non-persistent, it will be returned IF it will be returned to the cache to be reused by more users.

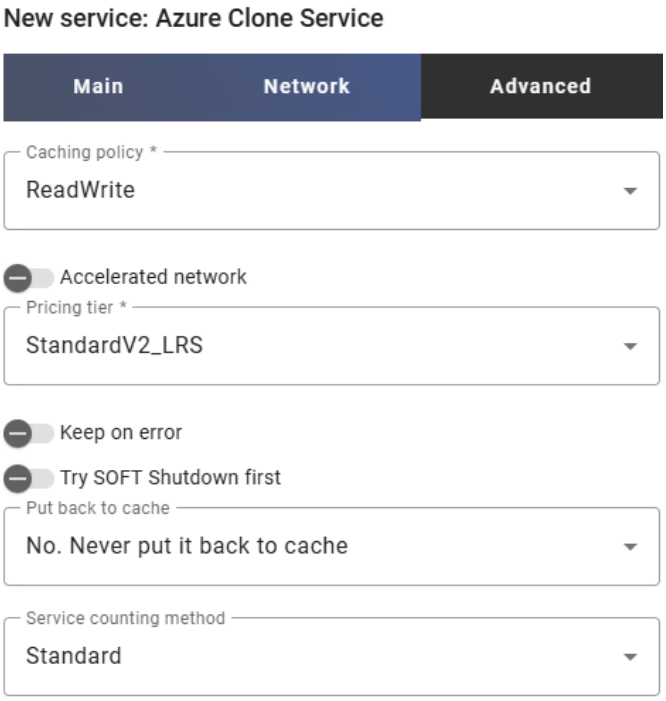
Service counting method: Method that the system will use to perform the machine count to comply with the parameter "maximum number of services to be provided" of a Service Pool:

- **Standard:** This counting mode only takes into account the machines that are being created and during the process of creating and destroying machines, it is possible to exceed the value of the parameter "maximum number of services to be provided" of a Service Pool.

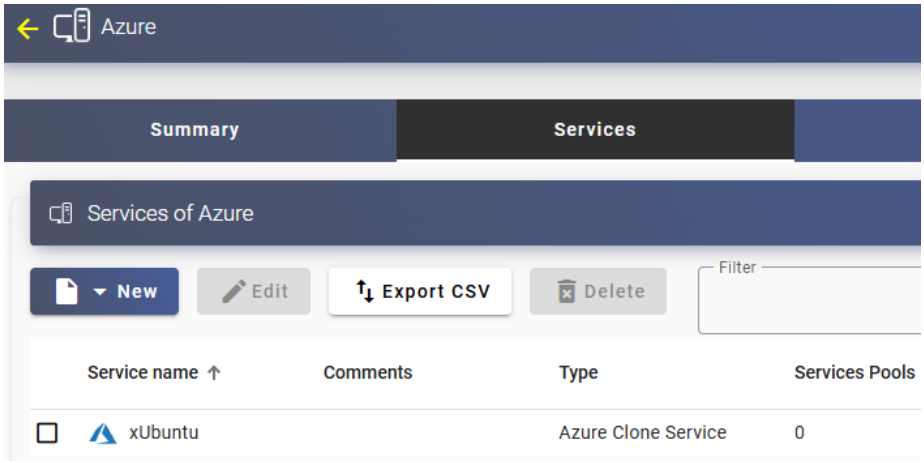
  For example, if a service pool, whose "maximum number of services to be provided" parameter is set to 5, we delete a service, there will be a time when that pool contains 6 machines: 4 configured, 1 being created and 1 being deleted.

- **Conservative:** This counting mode takes into account the machines that are being created and those that are being deleted, at no time will it be possible to exceed the value of the parameter "maximum number of services to be provided" of a Service Pool.

  For example, if a service pool, whose parameter of "maximum number of services to be provided" is set to 5, we delete a service, until that service is completely eliminated the system will not generate a new service, avoiding exceeding the value of 5 existing machines in the pool.



Save the configuration and we will have a valid "***Azure Clone Service***" on the Microsoft Azure platform. We will be able to register all the services of the "***Azure Clone Service***" type that we need.



Once we have the entire UDS environment configured (Services, Authenticators, OS Managers and Transports) and created the first "***Service Pool***", we will be able to observe in the Azure console ("***Virtual Machines***") how the desktop deployments are carried out.

The first task that will be performed will be to generate a publication of the template (this machine will be generated every time we publish a service). It will be a clone of the template chosen when registering for the service, with a disk size and characteristics equal to said template.

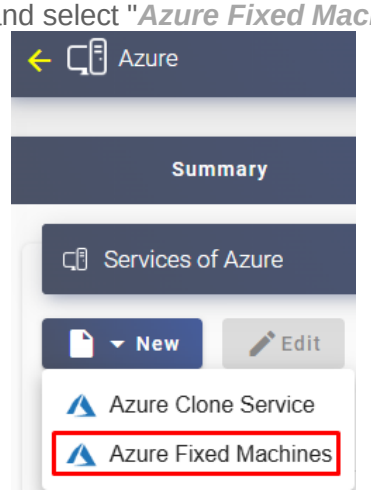
Once the publication creation process is finished, the desktops are automatically created (the system names them as: "UDS_*Machine_Name_Name_Length*") based on the cache parameters configured in the "***Service Pools***"".

UDS3-dbserver	Virtual machine	Running	UDS_Enterprise_3
UDS3-Server	Virtual machine	Running	UDS_Enterprise_3
UDS3-Tunnel	Virtual machine	Running	UDS_Enterprise_3
UDS_Desk_ubun_000_v1_389cfb78a96684_4a8360d57d3a_VM	Virtual machine	Running	UDS_Enterprise_3
UDS_Desk_ubun_001_v1_389cfb85d7a752_4a8360d57d3a_VM	Virtual machine	Running	UDS_Enterprise_3
UDS_Desk_ubun_002_v1_389cfb9305c582_4a8360d57d3a_VM	Virtual machine	Running	UDS_Enterprise_3
UDS_Desk_ubun_003_v1_389cfba034cfda_4a8360d57d3a_VM	Virtual machine	Running	UDS_Enterprise_3
xUbuntu18	Virtual machine	Stopped	UDS_Enterprise_3

3.1.8.3 Configure service based on “Azure Fixed Machines”

This type of service will allow users to connect directly to existing VMs in an Azure subscription

To create base services of type "***Azure Fixed Machines***" we access the "***Service Providers***", select the "***Services***" tab, click on "***New***" and select "***Azure Fixed Machines***”.



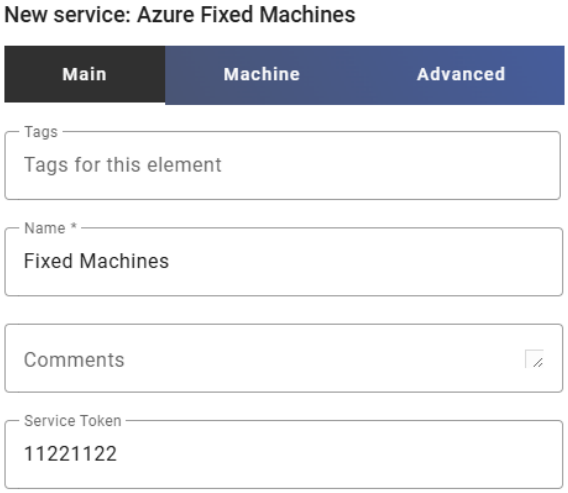
The minimum parameters to be configured in this type of service are:

- Main:

Name: Name of the service.

Service Token: If UDS is required to recognize the login and ***logout*** of a user accessing a service of this type (so that when a ***logout*** is made, UDS automatically releases the machine and makes it available to another user), it will be necessary to fill in this field with alphanumeric characters. If the field is left empty, UDS will permanently assign (until an administrator manually releases the service) the machine to a user, as it will have no way to control when a user enters or exits the machine.

NOTE: If the "**Service Token**" field is filled in, on the machines that are selected for the connection it will be necessary to have the UDS "**UDSActorUnmanagedSetup-x.x.x.exe**" actor installed and in the configuration of this, indicate those same characters in the "**Service Token" field**”.



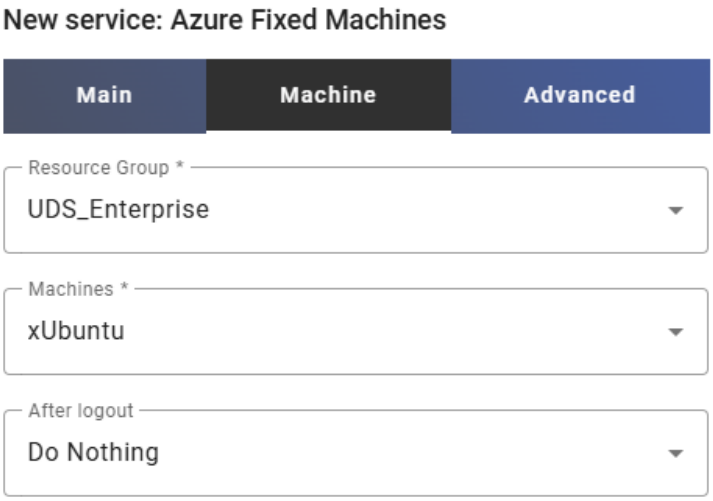
- Machine:

Resource Pool: Pool of resources in the Azure subscription where the virtual machines that we want to manage and assign to users are located. Machines must be assigned to this pool in order to be visible.

Machines: Virtual machines existing in the pool of resources indicated above that will be managed by UDS and to which access can be provided to the different users. We will have to indicate which machines will be accessible by users, the machines not selected, will be ignored and will not be assigned.

If the machines have the unamanaged actor installed and a "token" has been configured both in the actor (machine) and in the service definition (administration), it will be possible to control the access of users to these machines and they can be released so that other users can use them, in case they do NOT have the unmanaged actor installed, When a machine is assigned to a user, that assignment will be maintained until an administrator manually deletes it.

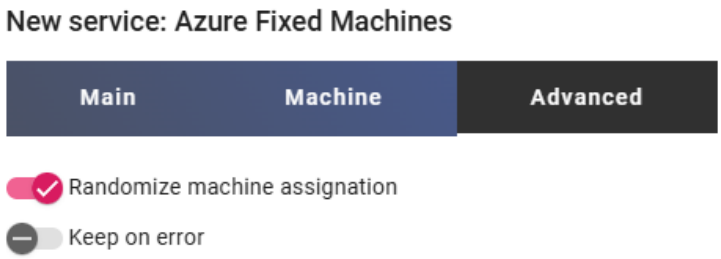
After Logout: Action that the system will perform when the user's logout is detected (it must have the actor installed). If we select "Stop Machine" the machine will shut down.



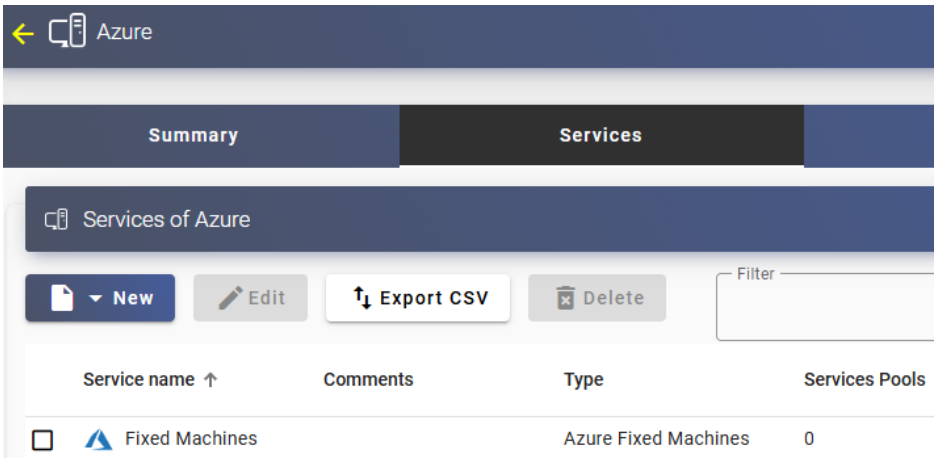
- Advanced

Randomize machine assignation: This parameter indicates how the order of assignment of the machines to the users will be, if it is active (by default) the order of assignment will be random, that is, when a user requests a machine, it will be unknown which one will be assigned. If the parameter is not active, the assignment order will be sequential, the first user will be given the first machine indicated in the list, the second the second machine and so on.

Keep on error: When activated, the system ignores all service access errors, allowing the machine to remain available for access (useful for occasions when machines may not be available for a short period of time but should be available at some point).).



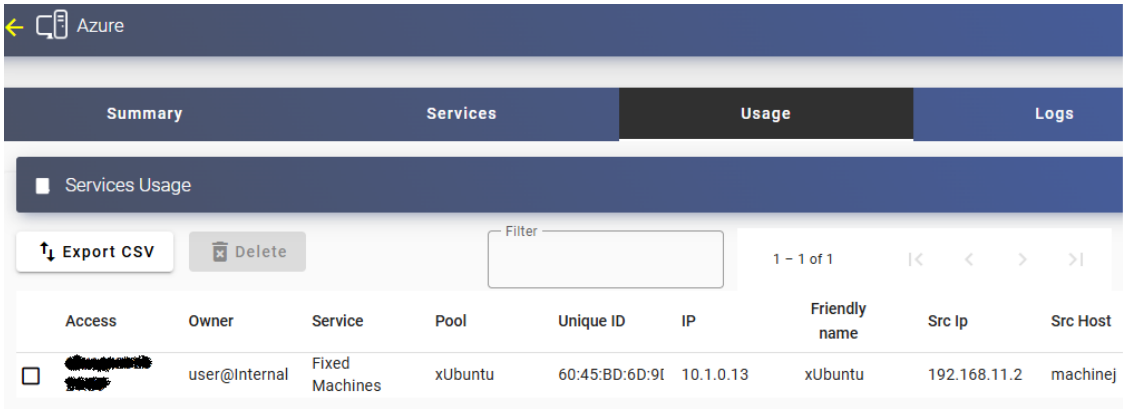
Save the configuration and we will have a valid " ***Azure Fixed Machines***" on the Azure platform. We will be able to register all the " ***Azure Fixed Machines***" services that we need.



3.1.8.4 Service usage summary and error logs

Within a "***Service Provider***" we will have a tab called "***Usage***" that will allow us to have a quick view with detailed information and perform basic actions of the services deployed in this "***Service Provider***".

To access this information on the use of the services, we access the "***Service Providers***" and select the "***Usage" tab.***":



We will be able to visualize:

Access: Date and time the service was accessed.

Owner: User and authenticator assigned to the service in usuario@auntenticador format. If it appears empty, it means that the service has not been assigned to any user and is available in the cache.

Service: Name of the base service belonging to the "***Service Provider***" from which the service was generated.

Pool: Name of the "***Service Pools***" to which the service belongs.

Unique ID: MAC address of the service.

IP: IP address of the service.

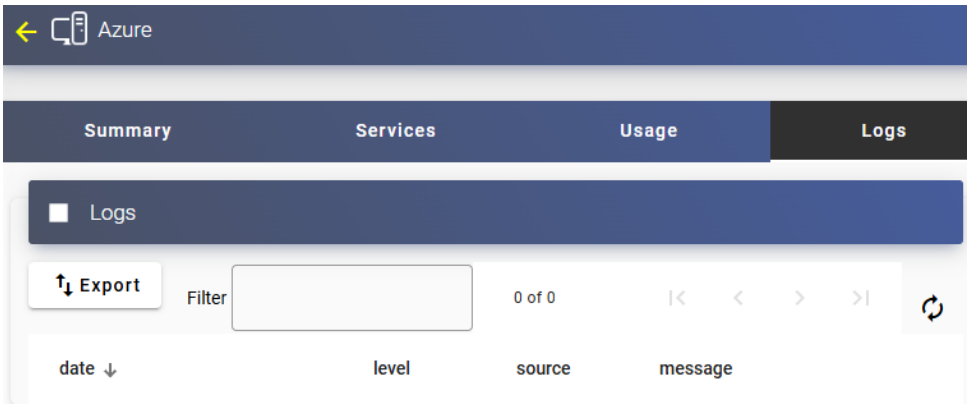
Friendly name: Name of the service. It will also be the DNS name of the UDS auto-generated virtual machine.

Src Ip: IP address of the connecting client accessing the service.

Src Host: DNS name of the connecting client accessing the service. If you are unable to access this name, your IP address will be indicated.

It will also be possible to select one or more services to proceed with their removal.

The "***Logs***" tab will show us information about a possible problem that occurred in the "***Service Provider***":



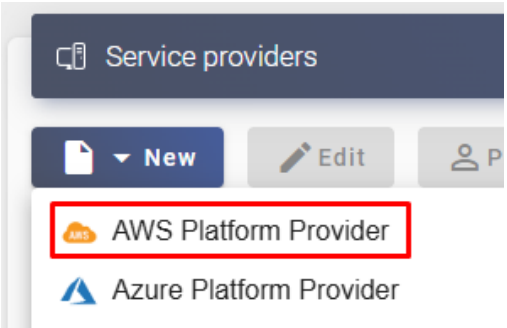
3.1.9 VDI with Amazon AWS

Deployment of self-generated virtual desktops by UDS or connection to existing machines, in an Amazon AWS cloud virtualization infrastructure.

3.1.9.1 Service provider registration “AWS Platform Provider”

The integration of UDS Enterprise with an AWS platform requires a series of previous tasks and requirements to be carried out in the AWS environment itself, so it is advisable to consult the manual "***VDI with UDS Enterprise and Amazon AWS"*** located in the documentation section of the UDS Enterprise website.

Click on "***New***" and select "***AWS Platform Provider***”.



In an AWS ***Platform Provider***, the minimum parameters to be configured are:

- Main:

Service Name, Access Key ID, and Secret Access Key.

NOTE: To locate and generate all the necessary data requested, you can consult the manual "VDI with UDS Enterprise and Amazon AWS" located in the documentation section of the UDS Enterprise website.

- Advanced:

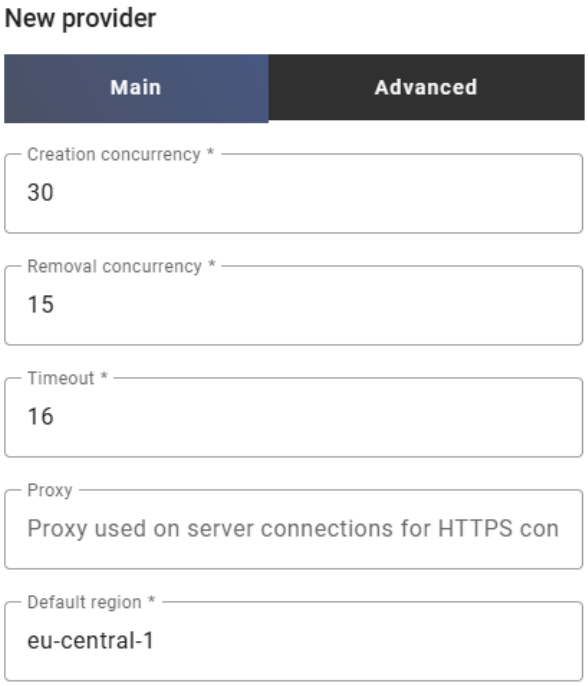
Creation concurrency. Maximum number of concurrent desktop creation tasks, in the different service pools that use it, for this provider.

Removal concurrency. Maximum number of concurrent desktop deletion tasks, in the different service pools that use it, for this provider.

Timeout: Timeout time in connecting to the AWS environment.

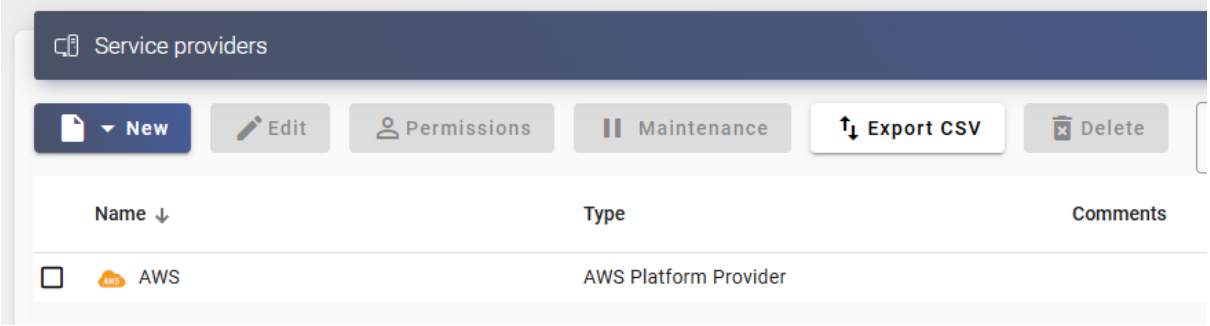
Proxy: Proxy server used in connection to the AWS environment.

Default region: Region to use in the AWS environment.



Using the "***Test***" button we will check that the connection is successful.

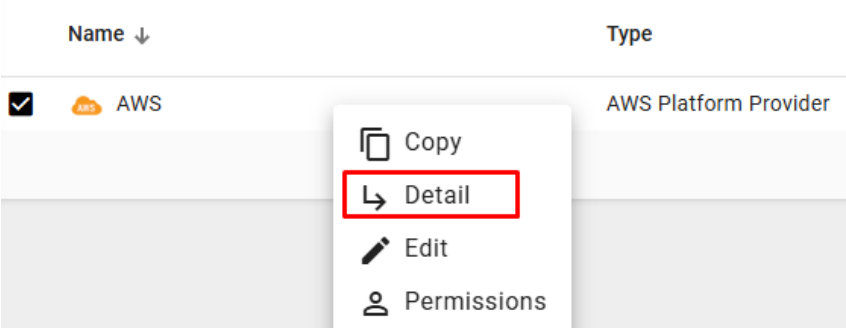
Save the configuration and we will have a valid "***Service Provider***" to start registering base services on the Amazon AWS platform.



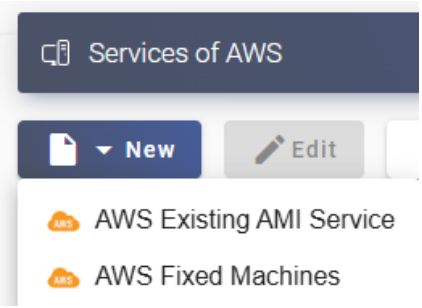
NOTE: If we have several Amazon AWS platforms, we will be able to register all the "Service Providers" of the "AWS Platform Provider" type that we need.

To modify a parameter in an existing *"Service Provider*", select it and click on "*Edit*".

By clicking on the "*Enter Maintenance Mode*" button, you can pause all operations produced by the UDS server on a service provider. It is recommended to put a service provider in maintenance cases where communication with that service provider has been lost or a maintenance shutdown is planned. Once the Azure platform is integrated into UDS, base services can be created. To do this, we double-click on the service provider created or in the provider's menu we select "*Detail*":



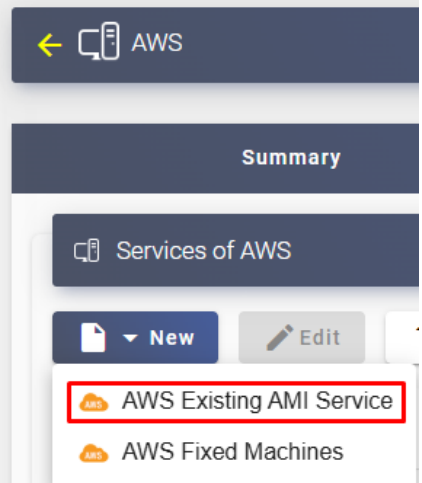
We will have the following types of basic services:



3.1.9.2 Configure service based on “AWS Existing AMI Service”

This type of service will create, within the virtual infrastructure, virtual desktops dependent on active publishing.

To create base services of type "***AWS Existing AMI Service***" we access the "***Service Provider***", select the "***Services***" tab, click on "***New***" and select "***AWS Existing AMI Service***”.



The minimum parameters to be configured in this type of service are:

- Main:

Name: Name of the service.

Region: The region of the AWS environment over which desktops will be unpacked. The AMIs to be used and the key pairs must also be hosted in this region.

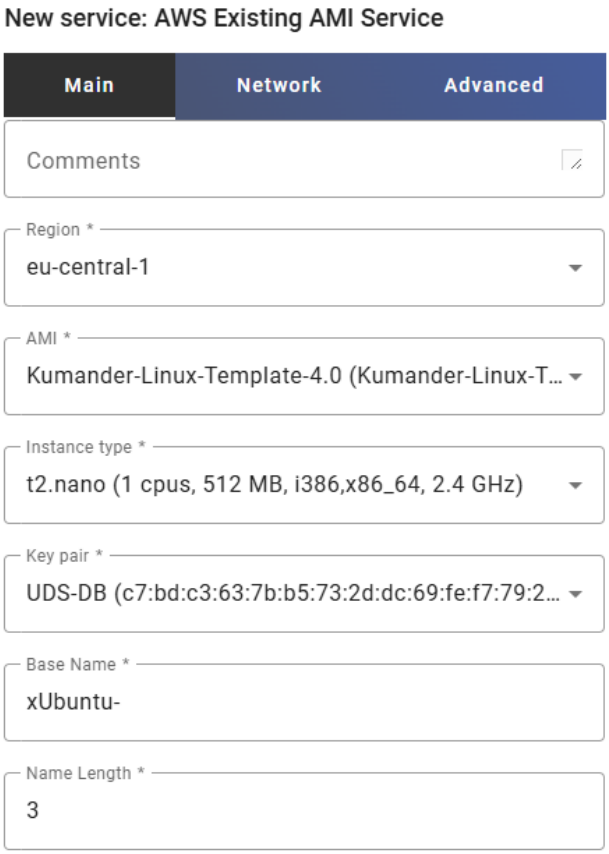
AMI: template (Gold Image) used by the system for the deployment of virtual desktops.

Instance type: Number of resources that the virtual desktops will have automatically deployed by UDS (this list will show all the types of machines available in AWS. Therefore, you need to make sure that the chosen type is supported by your AWS Region.)

Key pair: A pair of keys that will be assigned to the generated virtual desktops.

Base Name: Root of the name of all virtual desktops to be deployed in this service. (e.g.: Machine Names= xUbuntu-).

Name Length: Number of digits of the counter attached to the root of the desktop name (e.g.: Name Length= 3. The final name of the generated desktops would be: xUbuntu -**000,** xUbuntu -**001**... xUbuntu -**999**).

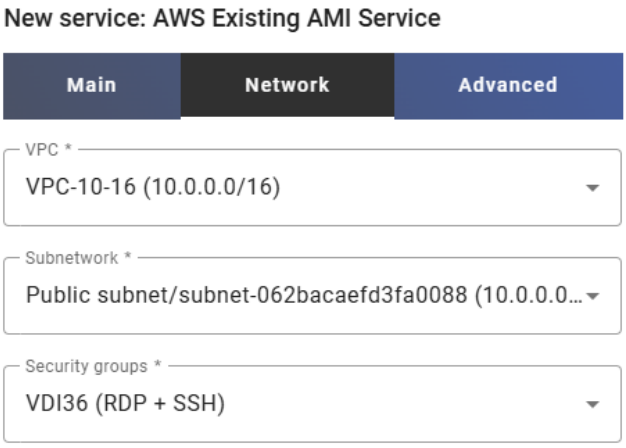


- Network:

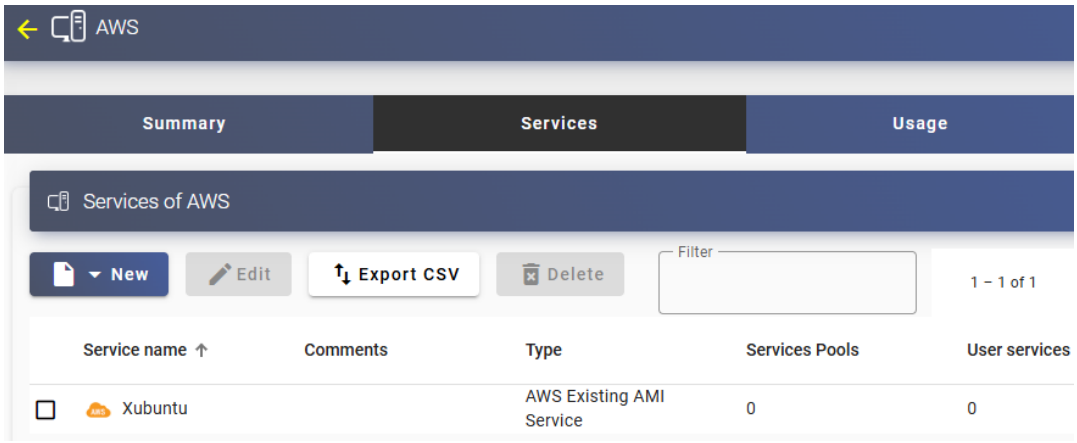
VPC: Existing VPC of the AWS environment to which the virtual desktops will be connected (there must be connectivity with the UDS Server and UDS Tunnel component).

Subnetwork: Existing subnet of the AWS environment to which the virtual desktops will be connected (there must be connectivity with the UDS Server and UDS Tunnel component).

Security Group: Security group to which to assign the new virtual desktops (this security group must have the necessary rules so that the generated desktops have communication with the different elements of the infrastructure).



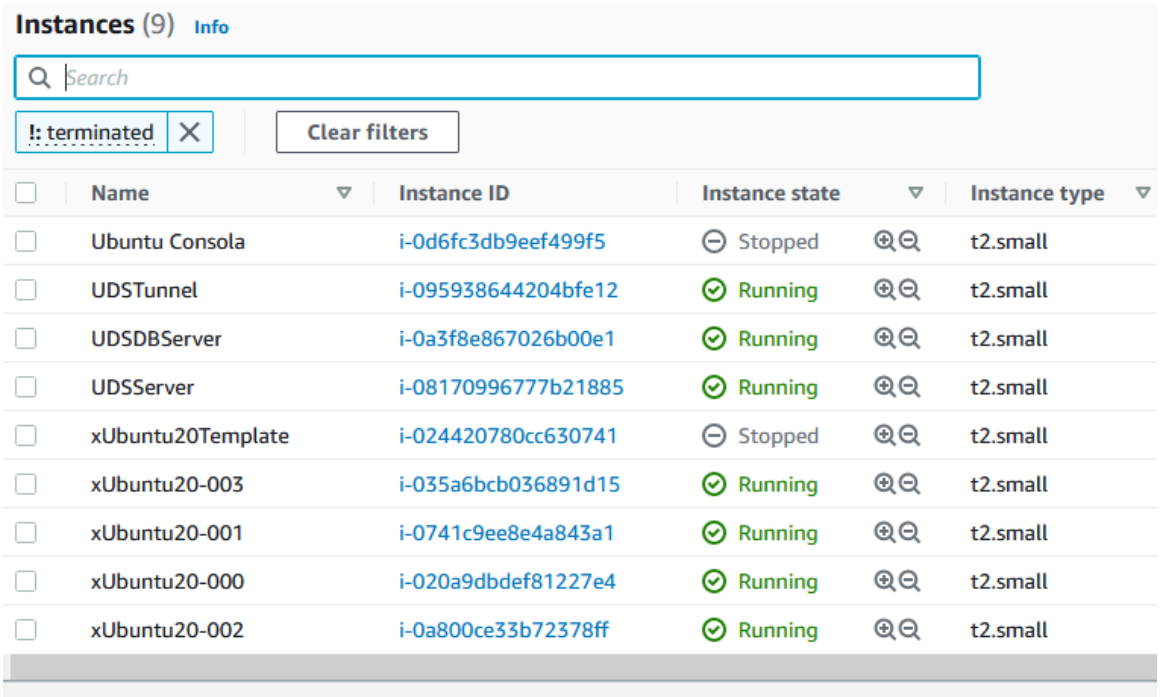
Save the configuration and we will have a valid " ***AWS Existing AMI Service*** " on the AWS platform. We will be able to register all the services of the " ***AWS Existing AMI Service*** " type that we need.



Once we have the entire UDS environment configured (Services, Authenticators, OS Managers and Transports) and created the first "***Service Pool***", we will be able to observe in the AWS EC2 console ("***Instances***") how the desktop deployments are carried out.

The first task that will be performed will be to generate a publication of the template (this machine will be generated every time we publish a service).

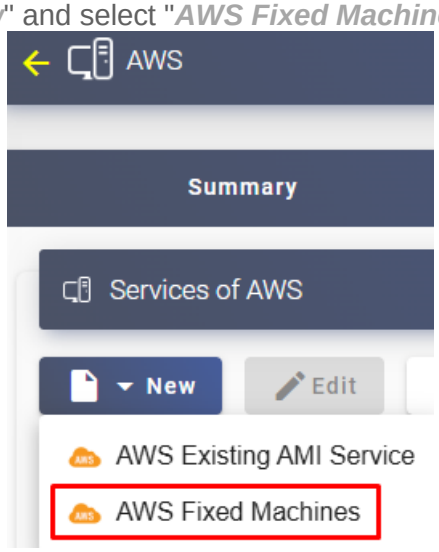
Once the publication creation process is finished, the desktops are automatically created (the system names them as: "UDS_*Machine_Name_Name_Length*") based on the cache parameters configured in the "***Service Pools***"".



3.1.9.3 Configure service based on “AWS Fixed Machines”

This type of service will allow users to connect directly to existing VMs in an AWS subscription

To create base services of type "***AWS Fixed Machines***" we access the "***Service Providers***", select the "***Services***" tab, click on "***New***" and select "***AWS Fixed Machines***”.



Los parámetros mínimos a configurar en este tipo de servicio son:

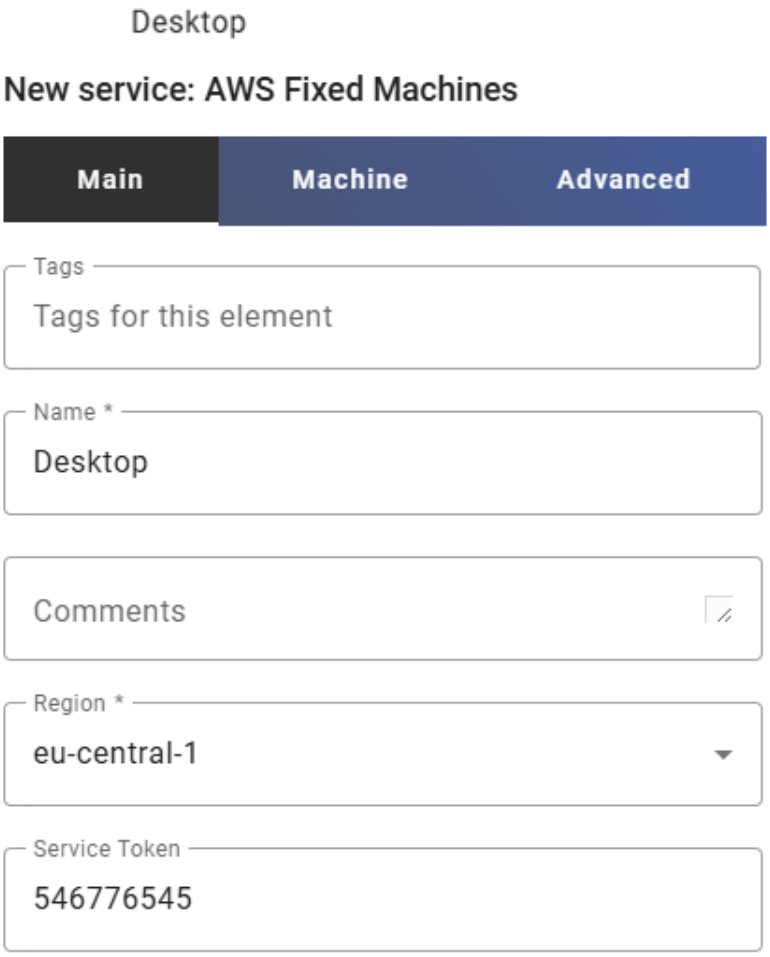
- Main:

Name: Name of the service.

Region: Region where the instances to which you want to give access will be located.

Service Token: If UDS is required to recognize the login and ***logout*** of a user accessing a service of this type (so that when a ***logout*** is made, UDS automatically releases the machine and makes it available to another user), it will be necessary to fill in this field with alphanumeric characters. If the field is left empty, UDS will permanently assign (until an administrator manually releases the service) the machine to a user, as it will have no way to control when a user enters or exits the machine.

NOTE: If the "**Service Token**" field is filled in, on the machines that are selected for the connection it will be necessary to have the UDS "**UDSActorUnmanagedSetup-x.x.x.exe**" actor installed and in the configuration of this, indicate those same characters in the "**Service Token" field**”.

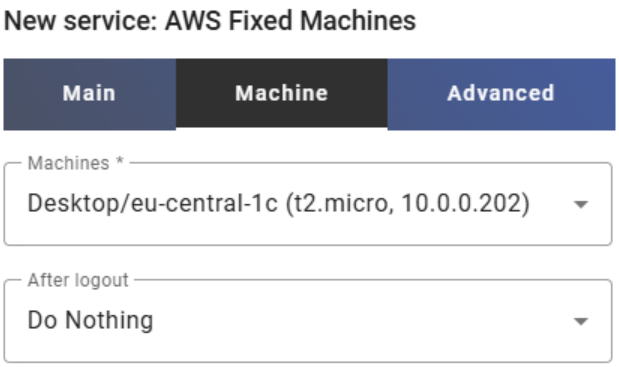


- Machine:

Machines: Virtual machines existing in the pool of resources indicated above that will be managed by UDS and to which access can be provided to the different users. We will have to indicate which machines will be accessible by users, the machines not selected, will be ignored and will not be assigned.

If the machines have the unamanaged actor installed and a "token" has been configured both in the actor (machine) and in the service definition (administration), it will be possible to control the access of users to these machines and they can be released so that other users can use them, in case they do NOT have the unmanaged actor installed, When a machine is assigned to a user, that assignment will be maintained until an administrator manually deletes it.

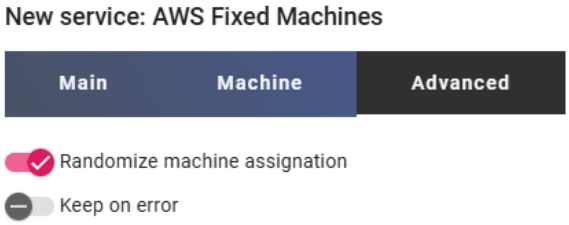
After Logout: Action that the system will perform when the user's logout is detected (it must have the actor installed). If we select "Stop Machine" the machine will shut down.



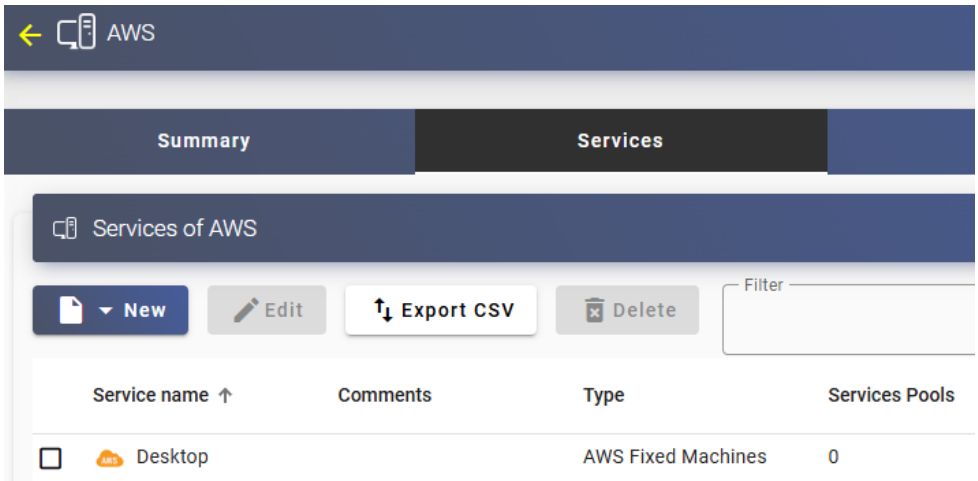
- Advanced

Randomize machine assignment: This parameter indicates how the order of assignment of the machines to the users will be, if it is active (by default) the order of assignment will be random, that is, when a user requests a machine, it will be unknown which one will be assigned. If the parameter is not active, the assignment order will be sequential, the first user will be given the first machine indicated in the list, the second the second machine and so on.

Keep on error: When activated, the system ignores all service access errors, allowing the machine to remain available for access (useful for occasions when machines may not be available for a short period of time but should be available at some point).).



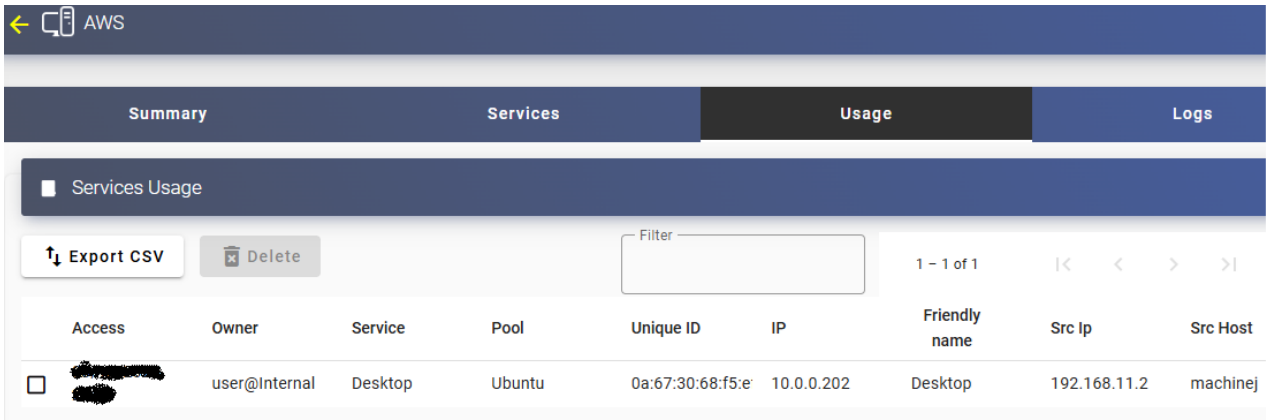
Save the configuration and we will have a valid "***AWS Fixed Machines***" on the Azure platform. We will be able to register all the " ***AWS Fixed Machines***" services that we need.



3.1.9.4 Service usage summary and error logs

Within a "***Service Provider***" we will have a tab called "***Usage***" that will allow us to have a quick view with detailed information and perform basic actions of the services deployed in this "***Service Provider***".

To access this information on the use of the services, we access the "***Service Providers***" and select the "***Usage" tab.***":



We will be able to visualize:

Access: Date and time the service was accessed.

Owner: User and authenticator assigned to the service in usuario@auntenticador format. If it appears empty, it means that the service has not been assigned to any user and is available in the cache.

Service: Name of the base service belonging to the "***Service Provider***" from which the service was generated.

Pool: Name of the "***Service Pools***" to which the service belongs.

Unique ID: MAC address of the service.

IP: IP address of the service.

Friendly name: Name of the service. It will also be the DNS name of the UDS auto-generated virtual machine.

Src Ip: IP address of the connecting client accessing the service.

Src Host: DNS name of the connecting client accessing the service. If you are unable to access this name, your IP address will be indicated.

It will also be possible to select one or more services to proceed with their removal.

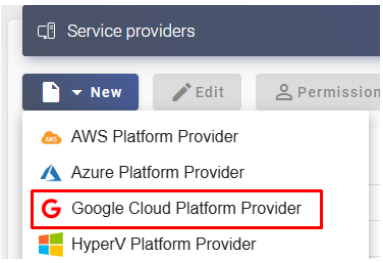
The "***Logs***" tab will show us information about a possible problem that occurred in the "***Service Provider'***":

3.1.10 VDI environment with Google Cloud

Deployment of virtual desktops auto-generated by UDS or connection to existing machines, in a Google Cloud cloud virtualization infrastructure.

3.1.10.1 Service provider registration “Google Cloud Platform Provider”

Click on "***New***" and select "***Google Cloud Platform Provider***”.



In a "***Google Cloud Platform Provider***" the minimum parameters to be configured are:

- Main:

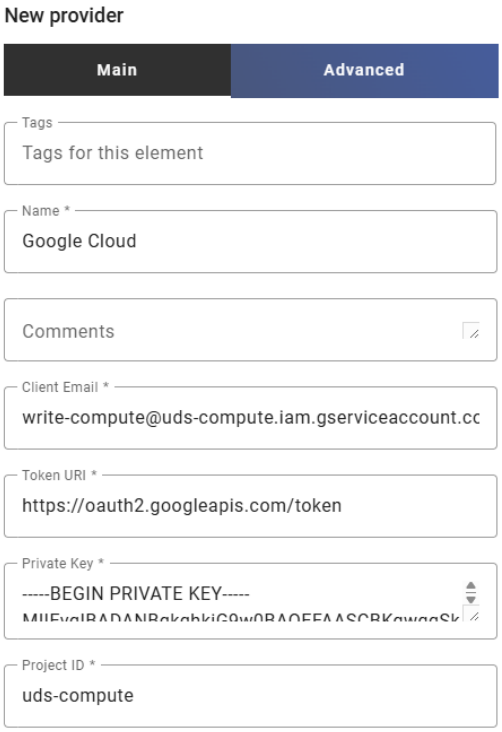
Name: Name of the service provider.

Client Email: Email registered in the service account.

URI Token: Platform token URL

Private Key: Service account key.

Project ID: Name of the project to be used.



- Advanced:

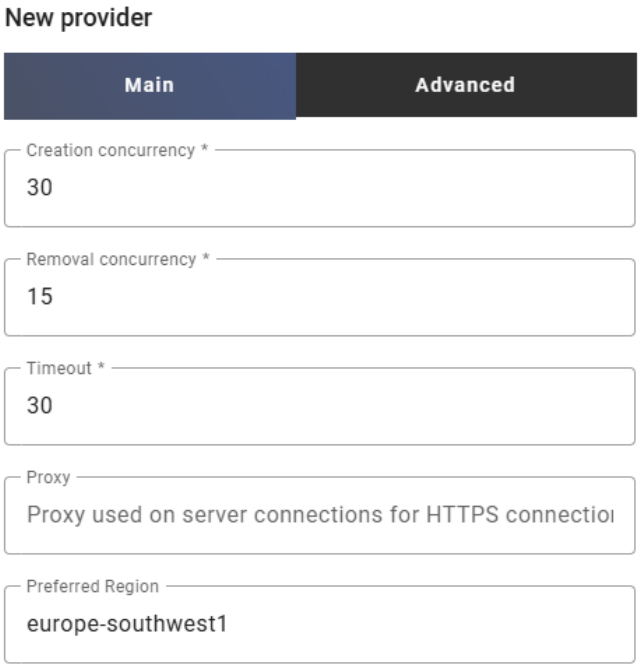
Creation concurrency: Maximum number of concurrent desktop creation tasks, in the different service pools that use it, for this provider.

Removal concurrency: Maximum number of concurrent desktop removal tasks, in the different service pools that use it, for this provider.

Timeout: The time of "***Timeout***" in the connection to the server.

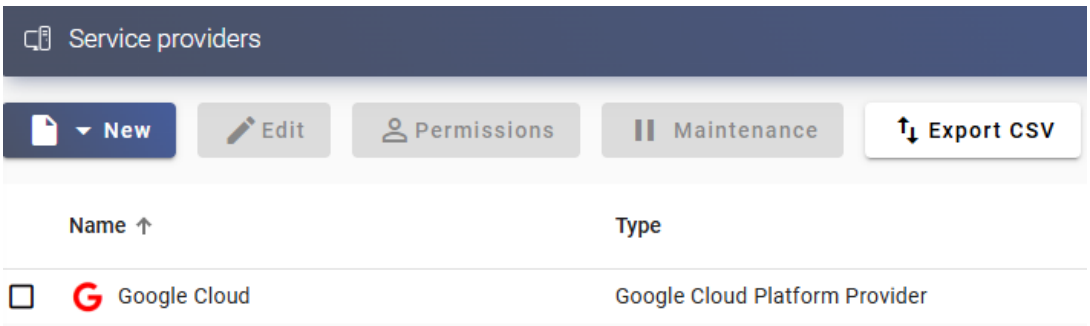
Proxy: Proxy address used in the connection to the provider.

Preferred Region: Region that will be used by default in deployed services.



Using the "***Test***" button we will check that the connection is successful.

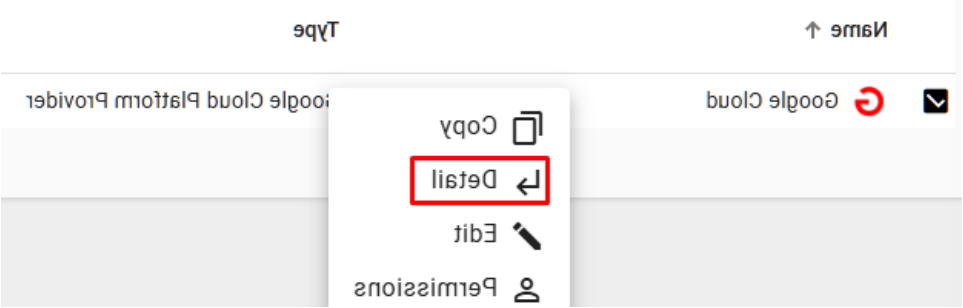
Save the configuration and we will have a valid "***Service Provider***" to start registering base services on the Google Cloud platform.



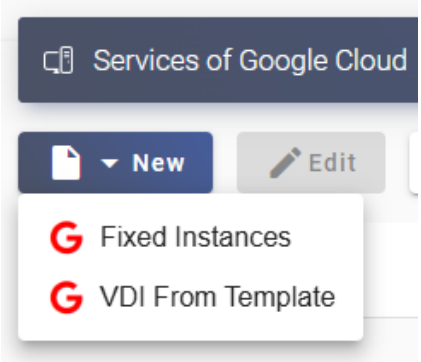
NOTE: If we have several subscriptions in the Google Cloud cloud, we will be able to register all the "Service Providers" of the "Google cloud Platform Provider" type that we need.
To modify a parameter in an existing ***"Service Provider***", select it and click on "***Edit***".

By clicking on the "***Enter Maintenance Mode***" button, you can pause all operations produced by the UDS server on a service provider. It is recommended to put a service provider in maintenance cases where communication with that service provider has been lost or a maintenance shutdown is planned.

Once the Google Cloud platform is integrated into UDS, base services can be created. To do this, we double-click on the service provider created or in the provider's menu we select "***Detail***":



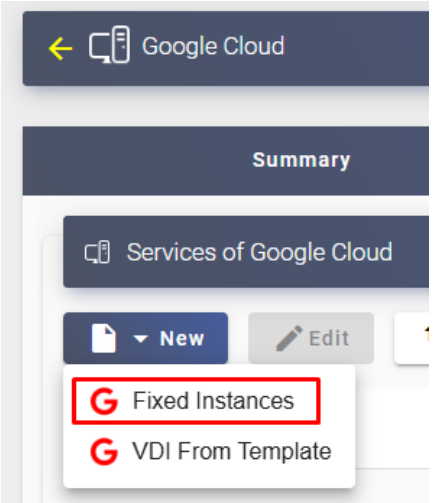
We will have the following types of basic services:



3.1.10.2 Configure service based on “Google Cloud Fixed Machines”

This type of service will allow users to connect directly to existing virtual machines in a Google Cloud subscription

To create base services of type "***Google Cloud Fixed Machines***" we access the "***Service Providers***", select the "***Services***" tab, click on "***New***" and select "***Google Cloud Fixed Machines***”.



The minimum parameters to be configured in this type of service are:

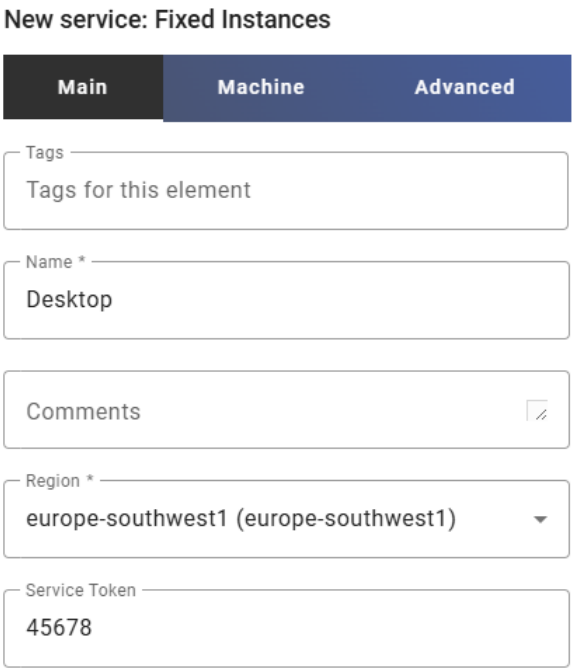
- Main:

Name: Name of the service.

Region: Region where the instances to be selected will be located.

Service Token: If UDS is required to recognize the login and ***logout*** of a user accessing a service of this type (so that when a ***logout*** is made, UDS automatically releases the machine and makes it available to another user), it will be necessary to fill in this field with alphanumeric characters. If the field is left empty, UDS will permanently assign (until an administrator manually releases the service) the machine to a user, as it will have no way to control when a user enters or exits the machine.

NOTE: If the "**Service Token**" field is filled in, on the machines that are selected for the connection it will be necessary to have the UDS "**UDSActorUnmanagedSetup-x.x.x.exe**" actor installed and in the configuration of this, indicate those same characters in the "**Service Token" field**”.

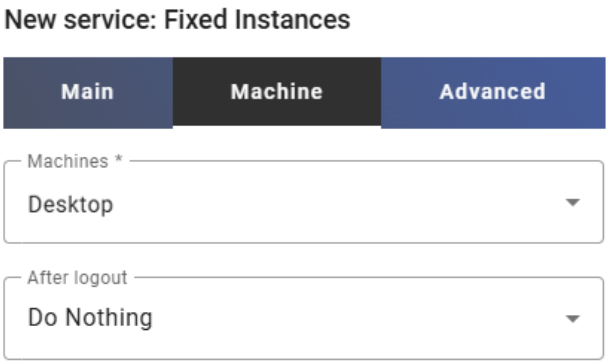


- Machine:

Machines: Existing virtual machines that will be managed by UDS and to which access can be provided to the different users. We will have to indicate which machines will be accessible by users, the machines not selected, will be ignored and will not be assigned.

If the machines have the unamanaged actor installed and a "token" has been configured both in the actor (machine) and in the service definition (administration), it will be possible to control the access of users to these machines and they can be released so that other users can use them, in case they do NOT have the unmanaged actor installed, When a machine is assigned to a user, that assignment will be maintained until an administrator manually deletes it.

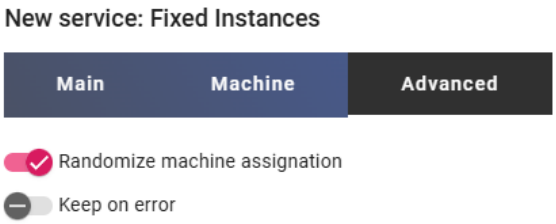
After Logout: Action that the system will perform when the user's logout is detected (it must have the actor installed). If we select "Stop Machine" the machine will shut down.



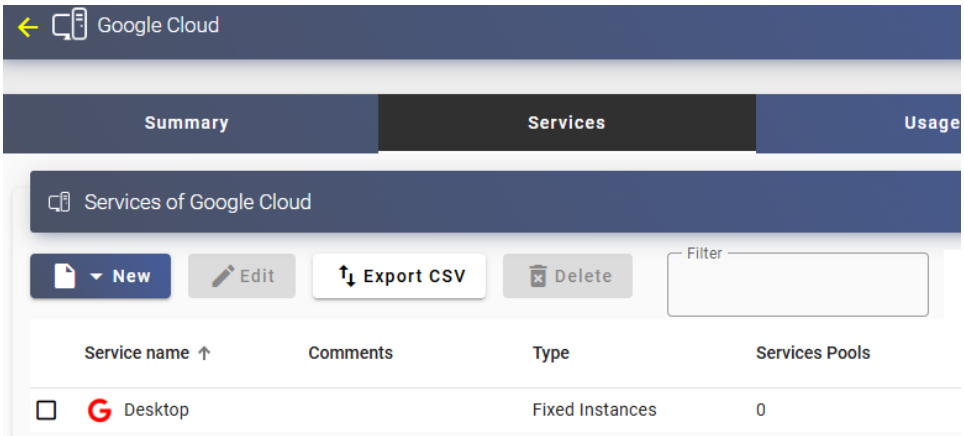
- Advanced

Randomize machine assignation: This parameter indicates how the order of assignment of the machines to the users will be, if it is active (by default) the order of assignment will be random, that is, when a user requests a machine, it will be unknown which one will be assigned. If the parameter is not active, the assignment order will be sequential, the first user will be given the first machine indicated in the list, the second the second machine and so on.

Keep on error: When activated, the system ignores all service access errors, allowing the machine to remain available for access (useful for occasions when machines may not be available for a short period of time but should be available at some point).).



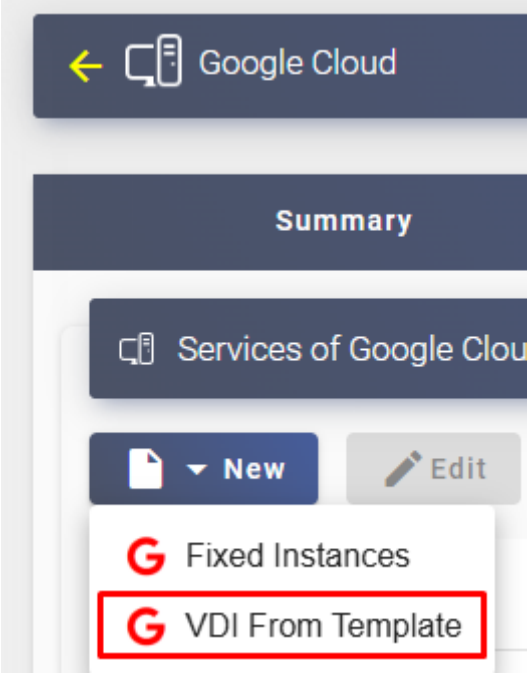
Save the configuration and we will have a "***Google Cloud Fixed Machines***" valid on the ***Google Cloud*** platform. We will be able to register all the "***Google Cloud Fixed Machines***" services that we need.



3.1.10.3 Configure service based on “VDI From Template”

This type of service will create, within the Google Cloud subscription, virtual desktops dependent on active publishing.

To create base services of type "***VDI From Template*** " we access the "***Service Provider***", select the "***Services***" tab, click on "***New***" and select "***VDI From Template***”.



The minimum parameters to be configured in this type of service are:

- Main:

Name: Name of the service.

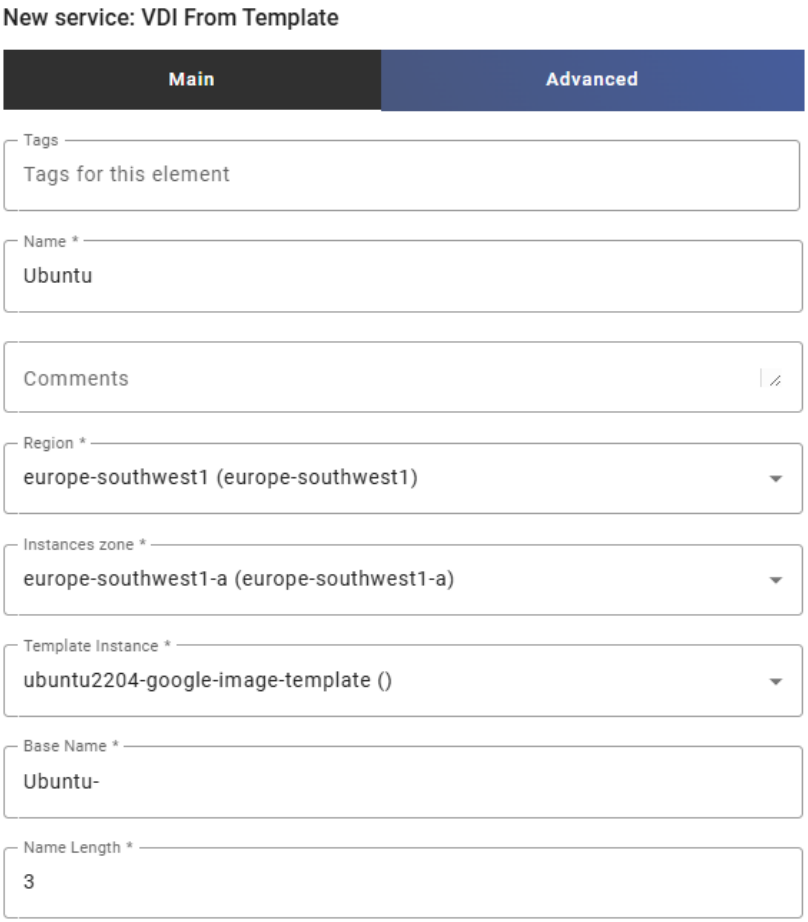
Region: Region used and where the template to be used is located.

Instances zone: Zone used and available in the selected region.

Template Instance: Template used to deploy desktops. Available instances will be displayed based on the selected region and zone.

Base Names: Root of the name of all virtual desktops to be deployed in this service. (e.g.: Machine Names= Desk-ubun-).

Name Length: Number of digits of the counter attached to the root of the desktop name (e.g.: Name Length= 3. The final name of the generated desktops would be: Desk-ubun-000**,** Desk-ubun-001... Desk-ubun-999).



- Advanced:

Keep on error: When activated, the system ignores all service access errors, thus preventing desktop deletion. Very useful and recommended for use when using persistent machines, preventing the desktop from being deleted by the system under virtually any circumstances (except if an administrator manually deletes it).

Try SOFT Shutdown first: When activated, the system will attempt to perform a "Soft" shutdown on the OS of the virtual desktop. It is necessary that the tools are installed and running on the machine to be deleted, otherwise, after a period of time a forced shutdown will be performed.

Put back to cache: Parameter that defines whether machines will be returned to the cache, so that they can be used by another user, once they are freed:

- **No. Never put it back to cache:** When a user logs out of the desktop and the machine is non-persistent, it will be destroyed and will NOT be returned to the cache to be reused by more users.

- **Yes, try to put it back to cache:** When a user logs off the desktop and the machine is non-persistent, it will be returned IF it will be returned to the cache to be reused by more userss.

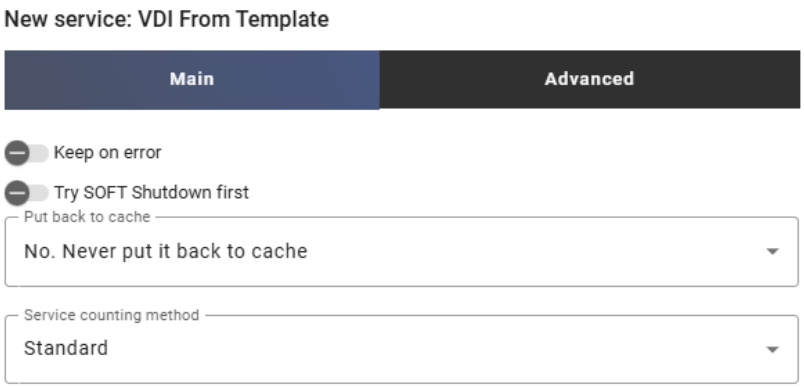
Service counting method: Method that the system will use to perform the machine count to comply with the parameter "maximum number of services to be provided" of a Service Pool:

- **Standard:** This counting mode only takes into account the machines that are being created and during the process of creating and destroying machines, it is possible to exceed the value of the parameter "maximum number of services to be provided" of a Service Pool.

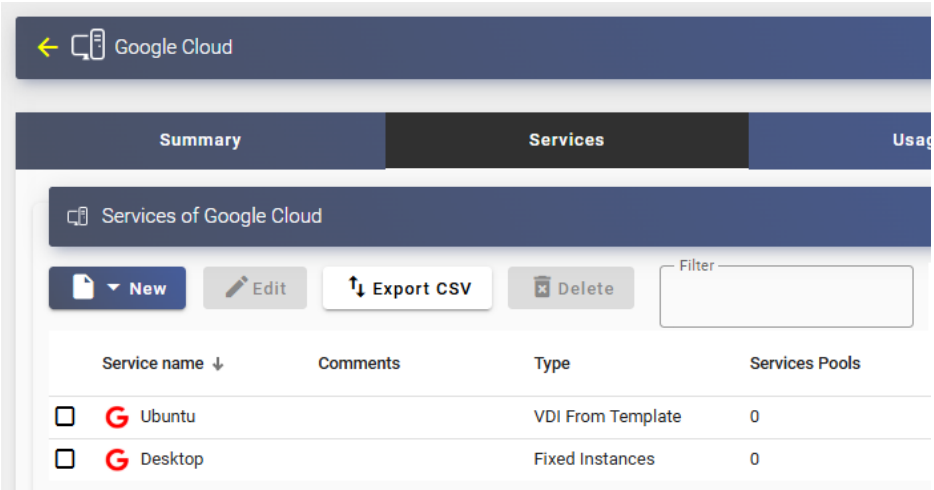
  For example, if a service pool, whose "maximum number of services to be provided" parameter is set to 5, we delete a service, there will be a time when that pool contains 6 machines: 4 configured, 1 being created and 1 being deleted.

- **Conservative:** This counting mode takes into account the machines that are being created and those that are being deleted, at no time will it be possible to exceed the value of the parameter "maximum number of services to be provided" of a Service Pool.

  For example, if a service pool, whose parameter of "maximum number of services to be provided" is set to 5, we delete a service, until that service is completely eliminated the system will not generate a new service, avoiding exceeding the value of 5 existing machines in the pool.



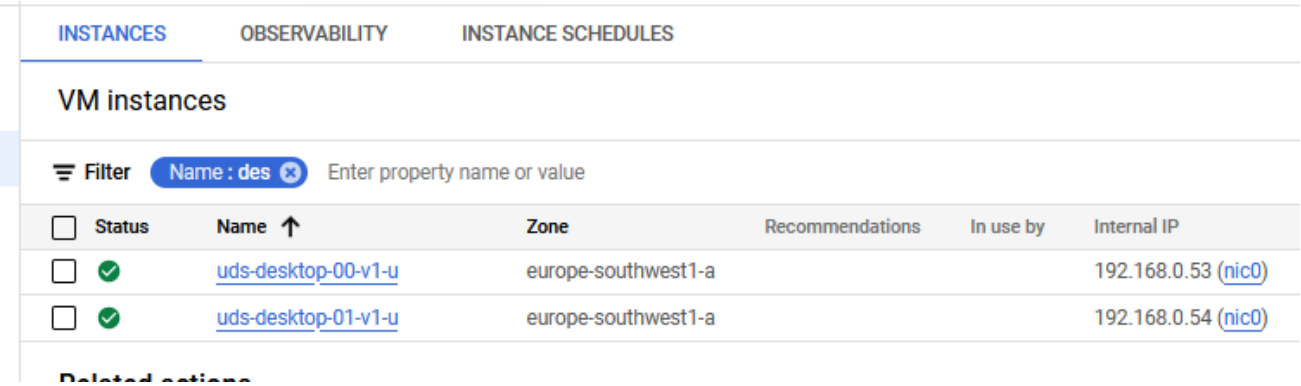
Save the configuration and we will have a valid "***VDI From Template***" on the Microsoft Google Cloud platform. We will be able to register all the services of the "***VDI From Template***" type that we need.



Once we have the entire UDS environment configured (Services, Authenticators, OS Managers and Transports) and created the first "***Service Pool***", we will be able to observe in the Google Cloud console ("***Virtual Machines***") how the desktop deployments are carried out.

The first task to be performed will be to generate a template post. It will be a clone of the template chosen when registering for the service, with a disk size and characteristics equal to said template.

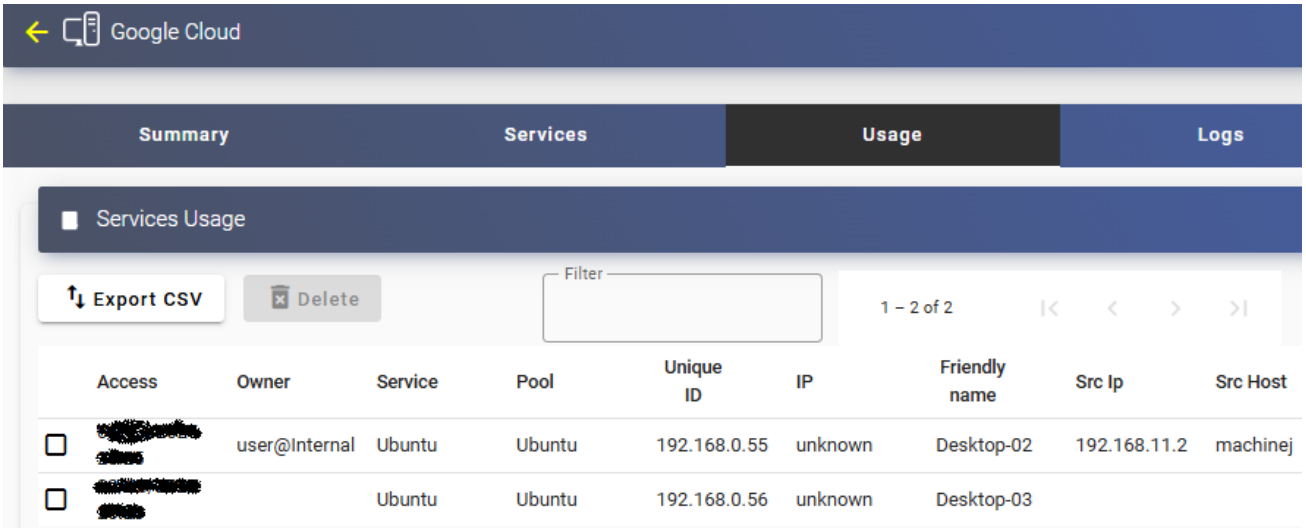
Once the publication creation process is finished, the desktops are automatically created (the system names them as: "UDS_*Machine_Name_Name_Length*") based on the cache parameters configured in the "***Service Pools***"".



3.1.10.4 Service usage summary and error logs

Within a "***Service Provider***" we will have a tab called "***Usage***" that will allow us to have a quick view with detailed information and perform basic actions of the services deployed in this "***Service Provider***".

To access this information on the use of the services, we access the "***Service Providers***" and select the "***Usage" tab.***":



We will be able to visualize:

Access: Date and time the service was accessed.

Owner: User and authenticator assigned to the service in usuario@auntenticador format. If it appears empty, it means that the service has not been assigned to any user and is available in the cache.

Service: Name of the base service belonging to the "***Service Provider***" from which the service was generated.

Pool: Name of the "***Service Pools***" to which the service belongs.

Unique ID: MAC address of the service.

IP: IP address of the service.

Friendly name: Name of the service. It will also be the DNS name of the UDS auto-generated virtual machine.

Src Ip: IP address of the connecting client accessing the service.

.**Src Host:** DNS name of the connecting client accessing the service. If you cannot access this name, your IP address will be indicated.

También será posible seleccionar uno o varios servicios para proceder a su eliminación.

The "***Logs***" tab will show us information about a possible problem that occurred in the "***Service Provider'"***":

3.1.11 Direct connection to remote computers

This type of service ***provider*** allows the assignment of users to persistent or static computers, whether they are physical machines or virtual machines.

We will have two types of services:

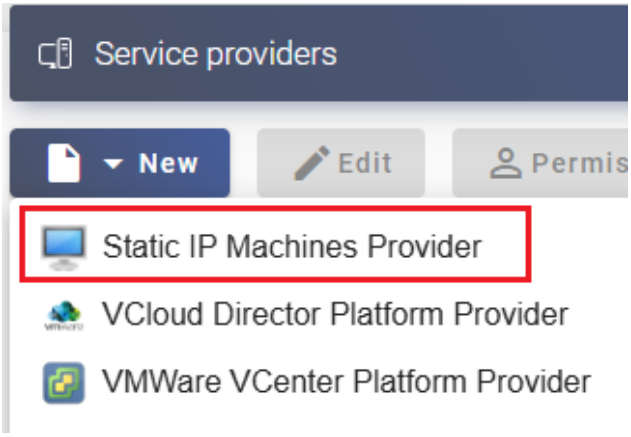
- "***Static Multiple IP***", which will allow us to create a base service to connect a single user with a computer (IP address). We can indicate one or more IP addresses so that the system can enable user access to these IPs. The assignment will be made in order of access, i.e. the first user who accesses this service will be assigned the first IP address in the list, although it is also possible to perform a selective assignment (an existing user in an authenticator to an IP address).

- "***Static Single IP***", which will allow us to create a base service to connect several users with a single computer. If your computer allows multiple sessions, each user who accesses it will start a new session on it.

NOTE: In order for users to be able to make a successful connection, it is necessary that the machines indicated by their IP address (either in the "Static Multiple IP" or "Static Single IP" type) are previously turned on and have the assigned connection protocol installed and enabled to perform the access.

3.1.11.1 Service provider registration “Static IP Machines Provider”

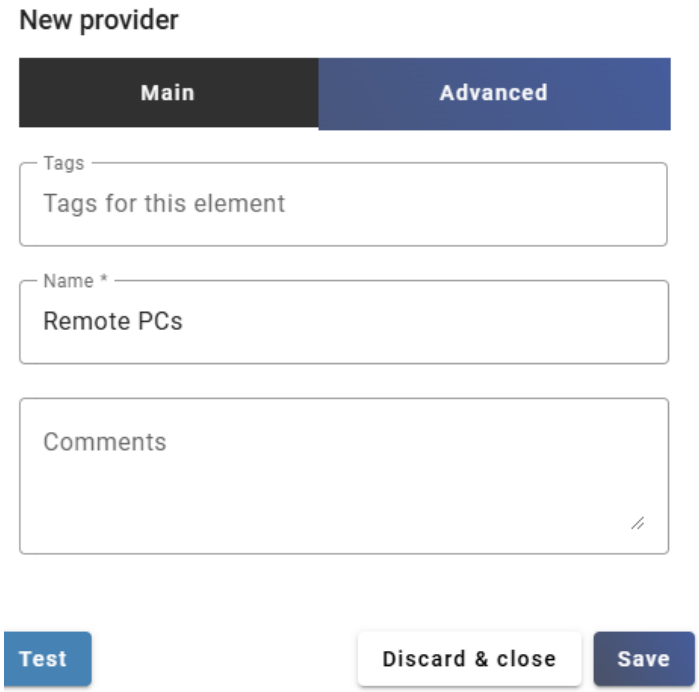
Click on "***New***" and select "***Static IP Machines Provider***”.



In a "***Static IP Machines Provider***" the minimum parameters to be configured are:

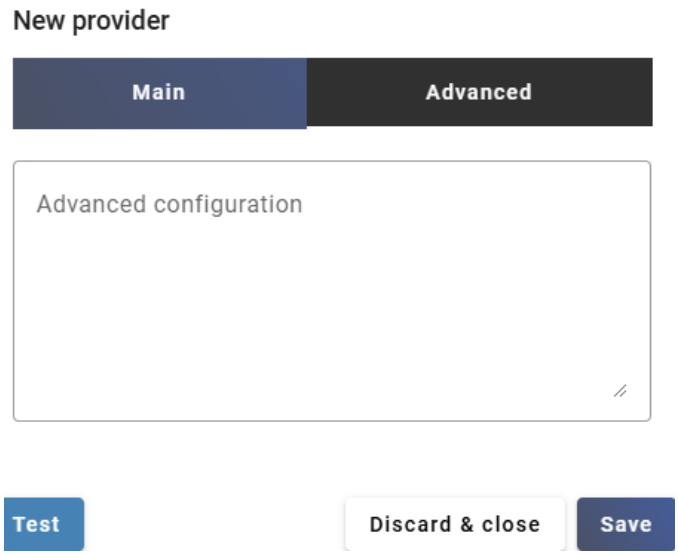
- Main:

Service Provider Name.



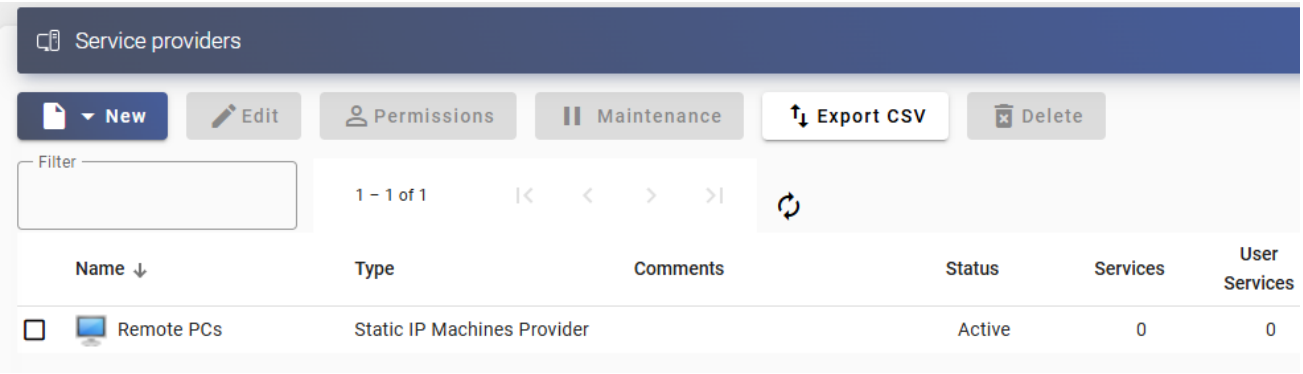
- Advanced:

Advanced configuración: We can indicate the [wol] option and thus enable "**Wake on LAN**" on physical computers.



NOTE: If you want to use the "Advanced configuration" section to enable WoL on physical computers, we recommend you consult the manual: "Wake on LAN of physical machines with UDS Enterprise" located in the documentation section of the UDSEnteprise website.

Save the configuration and we will have a valid "***Service Providers***" to start registering base services in a provider of type "***Static IP Machines Provider'***".

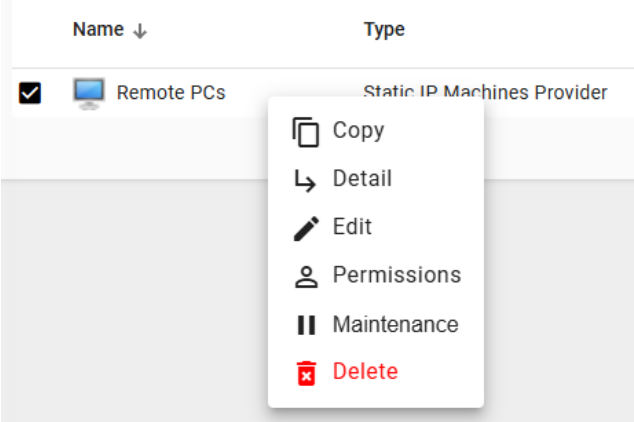


NOTA: We will be able to register all the "Service Providers" of the "Static IP Machines Provider" type that we need.

To modify a parameter in an existing ***"Service Providers***", select it and click on "***Edit***".

By means of the "***Enter Maintenance Mode***" button we will be able to pause all the operations produced by the UDS server on a service provider.

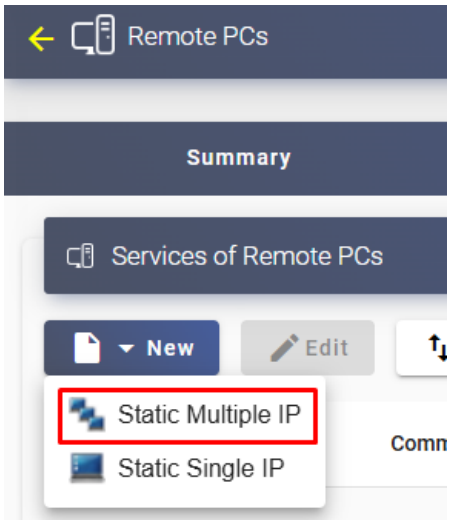
Once created, you can create base services. To do this, we double-click on the service provider created or in the provider's menu we select "***Detail***":



3.1.11.2 Configure service based on "Static Multiple IP"

This type of service will allow users to access different computers (physical or virtual). The connection will always be made one to one, that is, one user to a computer.

To create base services of type "***Static Multiple IP***" we access the "***Service Providers***", select the "***Services***" tab, click on "***New***" and select "***Static Multiple IP***".



The minimum parameters to be configured in this type of service are:

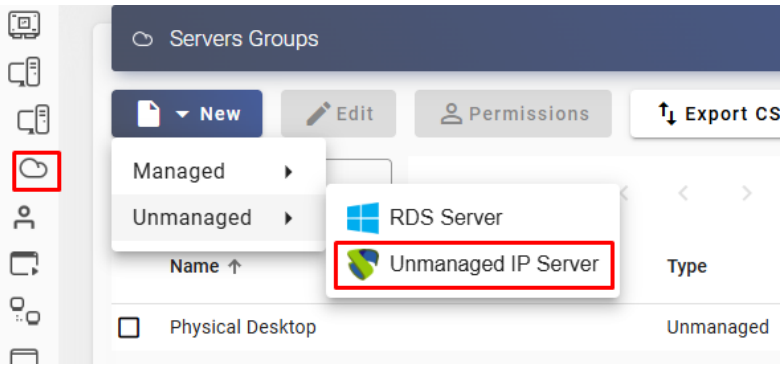
- Main:

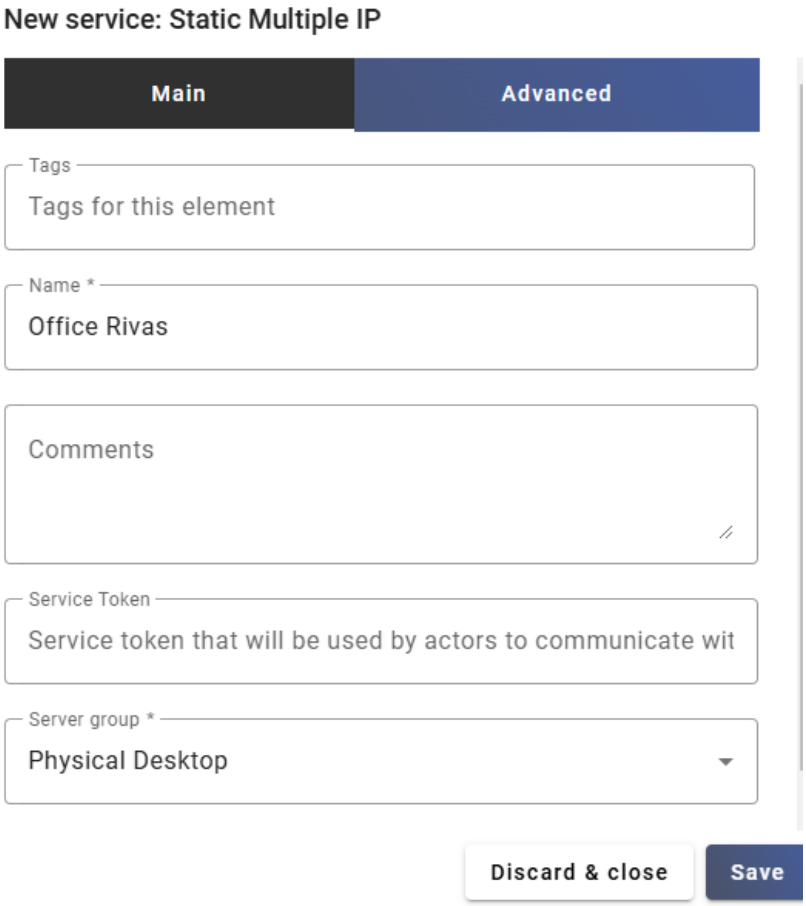
Name: Nombre del servicio.

Service Token: If there is no token indicated in this field (empty) the system will not control the user sessions on the computers and, therefore, when a computer is assigned to a user, that assignment will be maintained until manually deleted by an administrator. If a token exists, user sessions will be monitored. When they sign out of the computers, they will be released to become available again for other users to access.

If we indicate a token, it will be necessary for the computers belonging to the "***Server group***" to have the UDS Actor that manages static machines installed ("***UDSActorUnmanagedSetup-....***").

Server group: Group of machines that users will have access to. This element will be registered in "***Services***" - "***Servers***" and must be of type "***Unmanaged***" - "***Unmanaged IP Server***"





- Advanced:

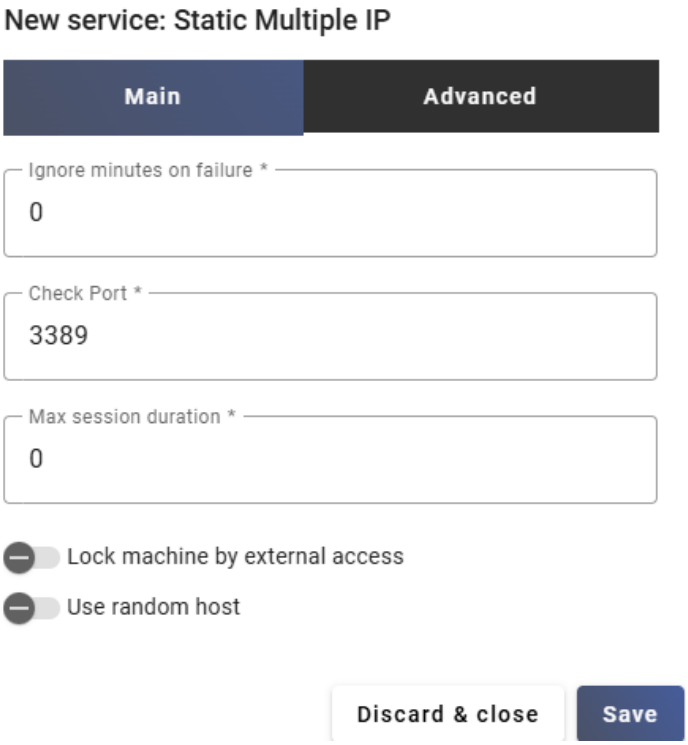
Ignore minutes on failure: If a port is indicated in the "***Check Port***" field and a machine is not available, we can indicate a time that will prevent the re-checking of that machine. If it indicates 0, the machines will always be checked. This parameter is indicated in minutes.

Check Port: Si hay un puerto indicado, antes de asignar un servicio a un usuario se comprueba que la máquina está accesible. En caso de que no esté disponible a través del puerto indicado, el sistema asigna la siguiente máquina disponible de la lista. Si no está indicado ningún puerto, no se verificará el acceso a las máquinas y se asignará sin importar el estado en el que se encuentre.

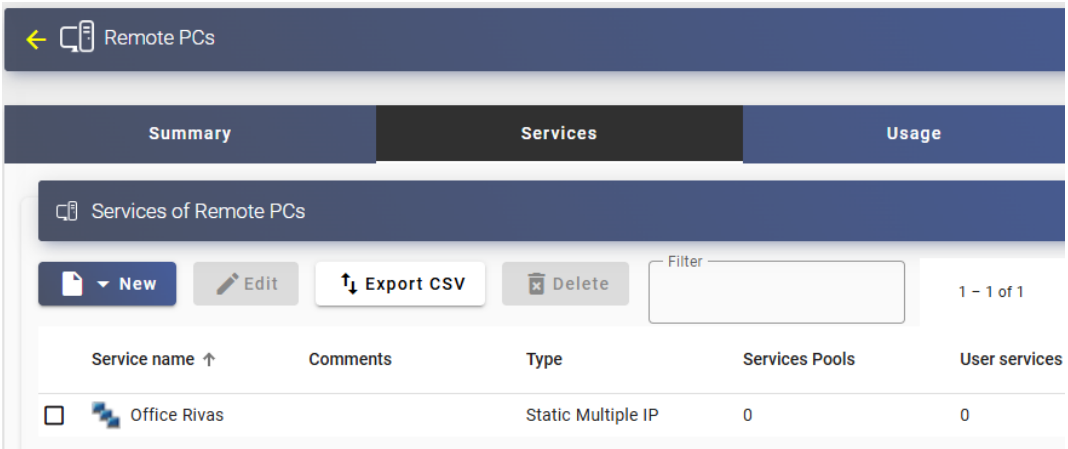
Max session duration: If the "**Lock machine by external Access**" parameter is enabled, a maximum time can be indicated for UDS to consider the equipment as reusable, even if access (from outside UDS) of a user has been detected.

Lock machine by external access: If enabled and the UDS "unmanage" actor is installed on the computer, UDS will prevent access to the service if there is already a logged-in user. For example, a physical machine that a user has accessed locally (not from UDS).

User random host: This parameter indicates how the order of assignment of the machines to the users will be, if it is active the order of assignment will be random, that is, when a user requests a machine, it will be unknown which one will be assigned. If the parameter is not active (by default), the assignment order will be sequential, the first user will be given the first machine indicated in the list, the second the second machine and so on.



Save the configuration and we will have a valid "***Static Multiple IP***". We will be able to register all the services of the "***Static Multiple IP***" type that we need.



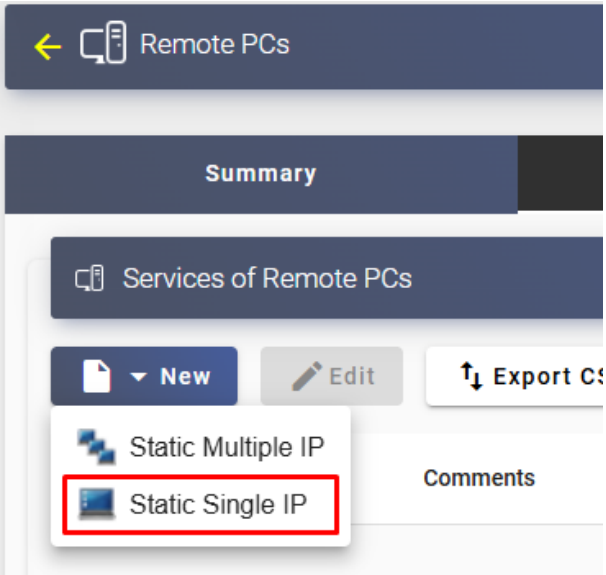
Once we have the entire UDS environment configured (Services, Authenticators and Transports) and created the first "***Service Pool***", users will access the IPs or names of the different computers registered in the service.

From a "***Service Pool***" it will also be possible to make a selective assignment, indicating which equipment is assigned to each user.

3.1.11.3 Configure service based on "Static Single IP""

This type of service will allow different users to access the same computer (physical or virtual). Each user will start a new session on the computer, as long as the computer is configured for this purpose.

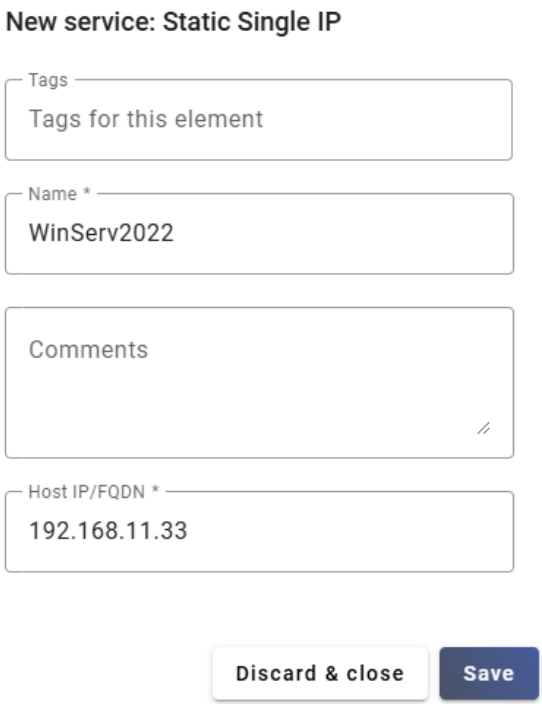
To create base services of type "***Static Single IP***" we access the "***Service Providers***", select the "***Services***" tab, click on "***New***" and select "***Static Single IP***".



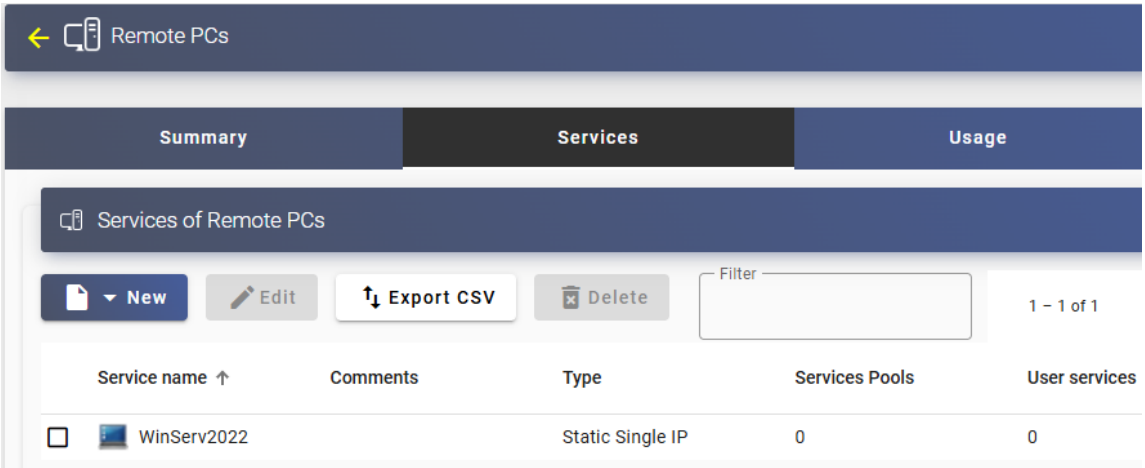
The minimum parameters to be configured in this type of service are:

Name: Name of the service.

IP Host/FQDN: IP address or name of the computer to which users will connect. The machine must allow access through the different user sessions.



Save the configuration and we will have a valid "***Static Single IP***". We will be able to register all the "***Static Single IP***" services that we need.

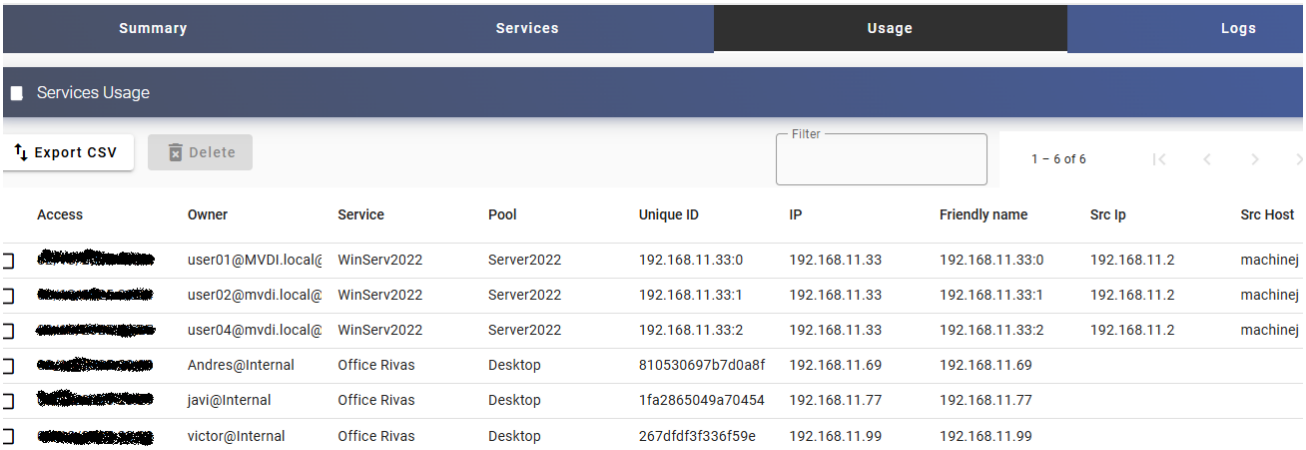


Once we have the entire UDS environment configured (Services, Authenticators and Transports) and created the first "***Service Pool***", users will access the IP of the indicated computer by starting a new session.

3.1.11.4 Service usage summary and error logs

Within a "***Service Provider***" we will have a tab called "***Usage***" that will allow us to have a quick view with detailed information and perform basic actions of the services deployed in this "***Service Provider***".

To access this information on the use of the services, we access the "***Service Providers***" and select the "***Usage" tab.***":



Access	Owner	Service	Pool	Unique ID	IP	Friendly name	Src Ip	Src Host
[redacted]	user01@MVDI.local(	WinServ2022	Server2022	192.168.11.33:0	192.168.11.33	192.168.11.33:0	192.168.11.2	machinej
[redacted]	user02@mvdi.local@	WinServ2022	Server2022	192.168.11.33:1	192.168.11.33	192.168.11.33:1	192.168.11.2	machinej
[redacted]	user04@mvdi.local@	WinServ2022	Server2022	192.168.11.33:2	192.168.11.33	192.168.11.33:2	192.168.11.2	machinej
[redacted]	Andres@Internal	Office Rivas	Desktop	810530697b7d0a8f	192.168.11.69	192.168.11.69		
[redacted]	javi@Internal	Office Rivas	Desktop	1fa2865049a70454	192.168.11.77	192.168.11.77		
[redacted]	victor@Internal	Office Rivas	Desktop	267dfdf3f336f59e	192.168.11.99	192.168.11.99		

We will be able to visualize:

Access: Date and time the service was accessed.

Owner: User and authenticator assigned to the service in format: usuario@autenticador.

Service: Name of the base service belonging to the "***Service Provider***" from which the service was generated.

Pool: Name of the "***Service Pools***" to which the service belongs.

Unique ID: The IP address of the service or identifier.

IP: IP address of the service.

Friendly name: IP address of the service.

Src Ip: IP address of the connecting client accessing the service.

Src Host: DNS name of the connection client. If you cannot access this name, your IP address will be indicated.

It will also be possible to select one or more services to proceed with its deletion.

The "***Logs***" tab will show us information about a possible problem that occurred in the "***Service Provider'***":

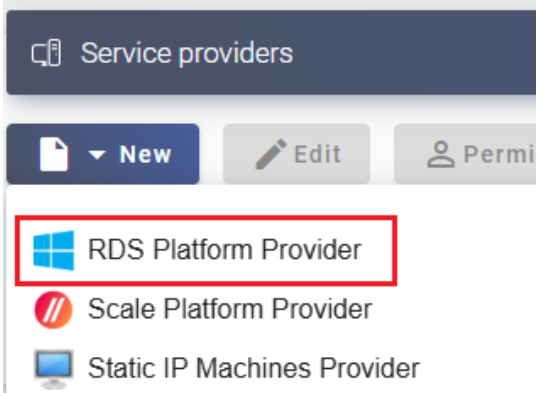
3.1.12 RDS Application Provider

This type of service ***provider*** allows deployment and connection to Windows virtual application sessions through Microsoft Remote Desktop Services (RDS).

With this provider, users who are validated from an authentication system other than "Active Directory" will also be able to access to start application sessions. To achieve this it will be necessary to use mapping users that UDS will automatically create on an existing AD server.

3.1.12.1 RDS Platform Provider Registration"

Click on "***New***" and select "***RDS Platform Provider***":



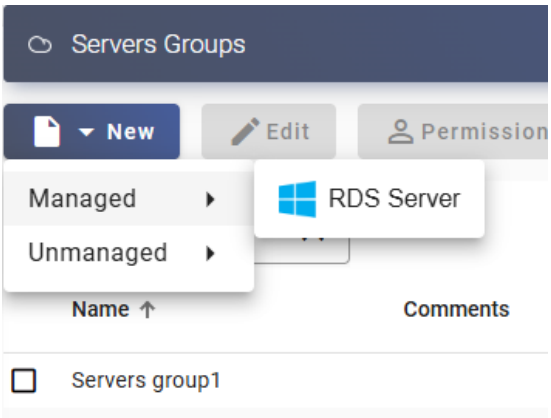
In an "RDS ***Platform Provider***" the minimum parameters to be configured are:

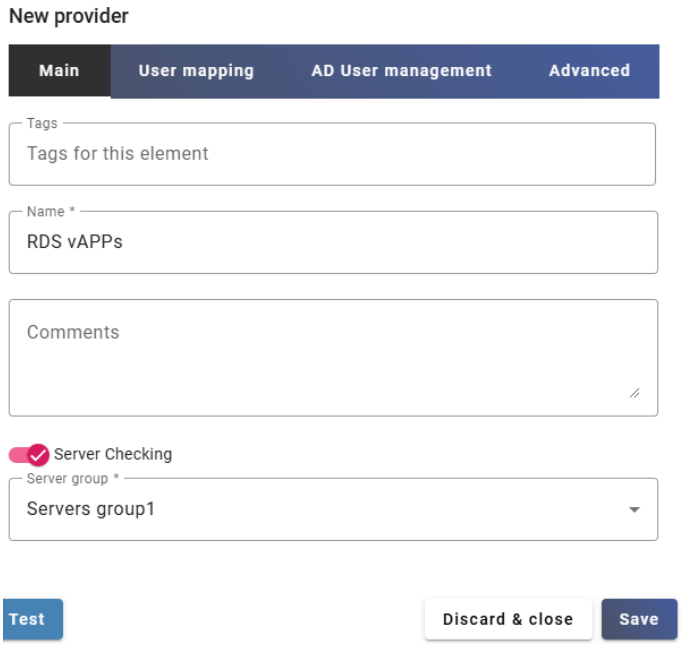
- Main:

Name: Name of the service.

Server checking: Checks if the RDS application server is available whenever there is a request to use it from a user. In case of not being able to contact the first server on the list, UDS will look for the next available server to offer the service.

Server group: A group of machines that users will have access to. This element will be registered in "***Services***" - "***Servers***" and must be of type "***Manage***" or "***Unmanaged***" - "***RDS Server***"





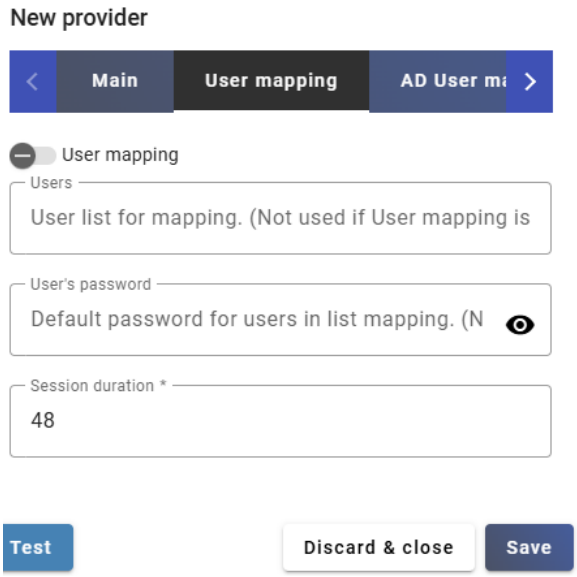
- User mapping:

NOTE: Obslete, only configurations from previous versions will be migrated, neither editing nor creation is allowed

User mapping: "*Yes*" indicates that specific users (indicated in the "Users" section) will be used to access the applications.*No*" will use the user of the UDS login portal to access the applications (in this case it has to be an AD user).

Users: Only applies if "*User mapping*" is set to "*Yes*". Active Directory "mapping" users who will be able to log in to RDS application servers and will be used by UDS only to run an application session.

User's password: Password of all the users indicated in the previous section. All mapping users need to have the same password.



- AD User management:

NOTE: Only enable "User auto creation on AD" in case you need to specify users from an AD domain. These users will be automatically created by UDS in an AD with LDAPS connectivity.

User auto creation on AD: "***Yes***" indicates that specific users self-created by UDS in an AD will be used to access applications.***No***" will use the user of the UDS login portal to access the applications (in this case it has to be an AD user).

AD Server: IP or name of the Active Directory server where new users will be created (the server must have LDAPS connection enabled).

Port: Port used in the connection.

AD server OU for created users: An organizational unit where new users will be created.

Username: Domain user with permissions to create and delete users. In format: *user@domain.xxx*

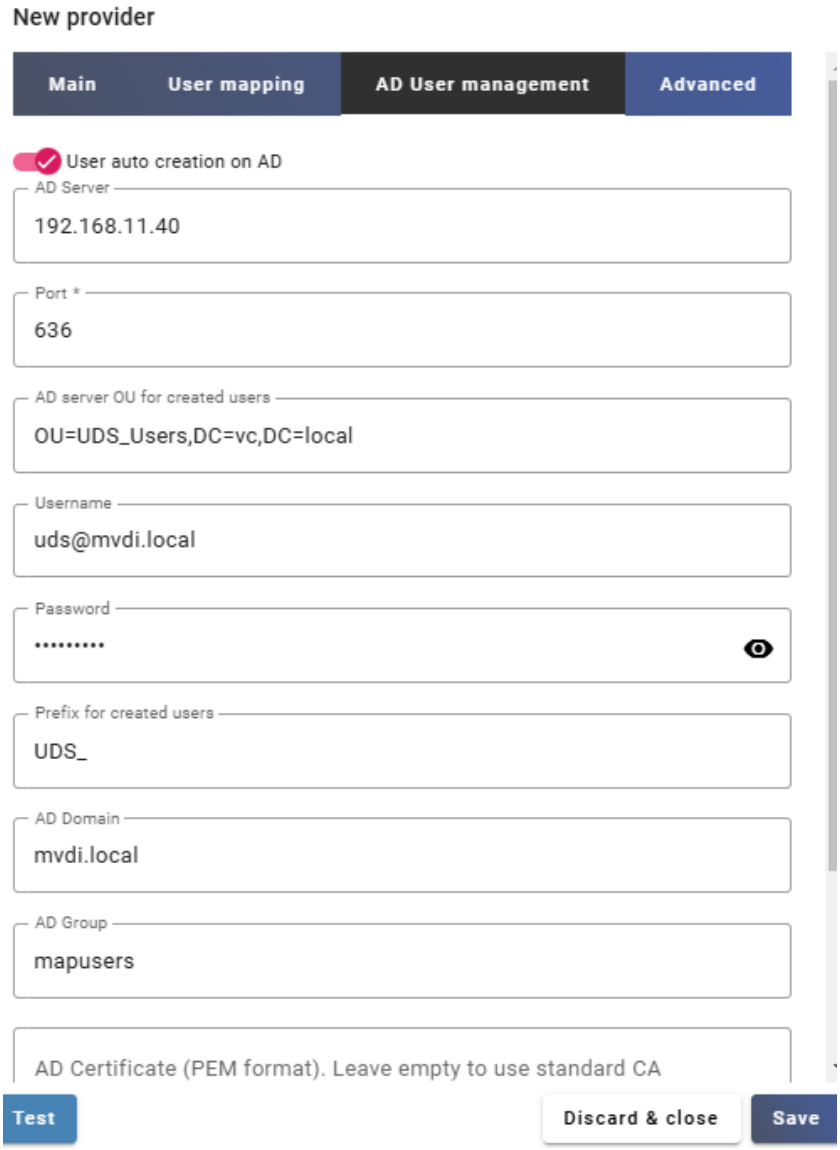
Password: Password of the indicated user

Safe use procedure: Passwords must be of sufficient length and include upper and lower case, numbers and special characters.

Prefix for created users: A prefix that will be added to the name of the user created in the AD. The final name of the created user will be: *prefix+nombre_usuario*.

AD Domain: Name of the domain where new users will register. If it is not indicated, the domain of the field: "Username" will be used.

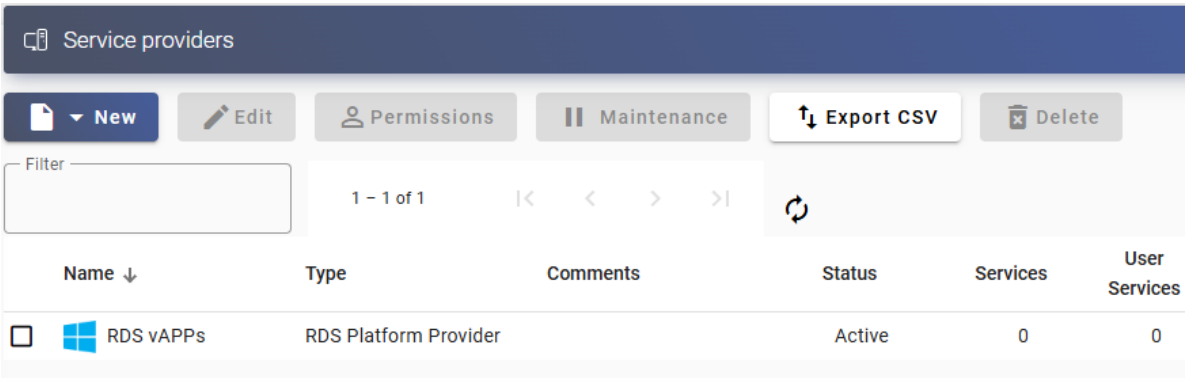
AD Group: Name of the group (must exist) to which UDS will add the newly created users.



NOTE: Only one of the two methods for user mapping can be activated: "User mapping" or "User auto creation on AD".

Using the "***Test***" button we will check that the connection is successful.

Save the configuration and we will have a valid "***Service Providers***" to start registering virtual application sessions.

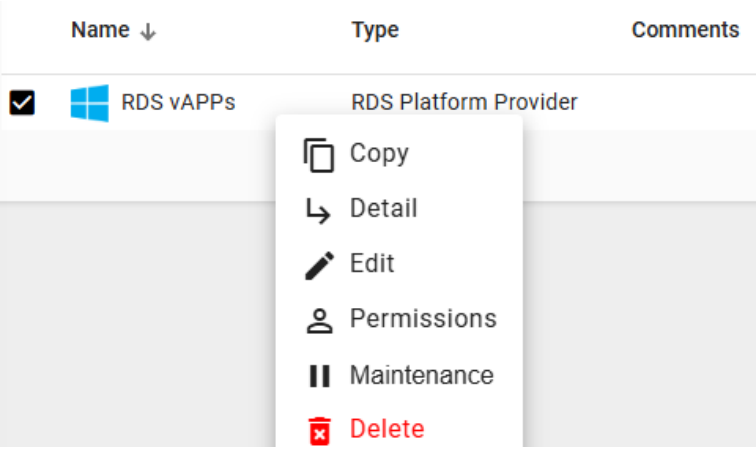


NOTE: If we need to create new application servers, we can register all the "Service Providers" of the "RDS Platform Provider" type that we need.

To modify a parameter in an existing "***Service Providers***", select it and click on "***Edit***".

By clicking on the "***Enter Maintenance Mode***" button, you can pause all operations produced by the UDS server on a service provider. It is recommended to put a service provider in maintenance cases where communication with that service provider has been lost or a maintenance shutdown is planned.

Once the application servers are integrated into UDS, base services can be created. To do this, we double-click on the service provider created or on the provider's menu and select "***Detail***":

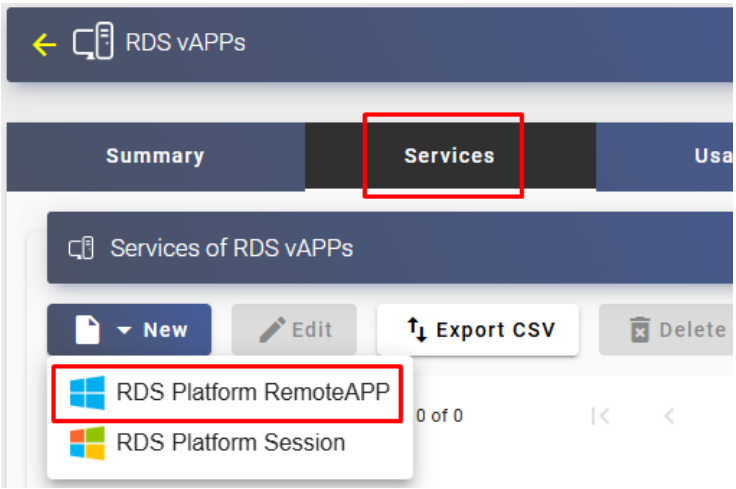


3.1.12.2 Configure service based on "RDS Platform RemoteAPP""

This base service will allow Windows virtual application sessions to be published to users.

Once the service provider has been created with a group of servers, composed of one or more application servers and the RDS server agent (***UDSRDSServerSetup-x.x.x.exe***) or appropriate UDS Actor installed in them, base services of type "***RDS Platform RemoteAPP" must be created***, where we will indicate the application to be virtualized.

To create base services of type "***RDS Platform RemoteAPP***" we access the "***Service Providers***", select the "***Services***" tab, click on "***New***" and select "***RDS Platform RemoteAPP***".



The minimum parameters to be configured in this type of service are:

- Main:

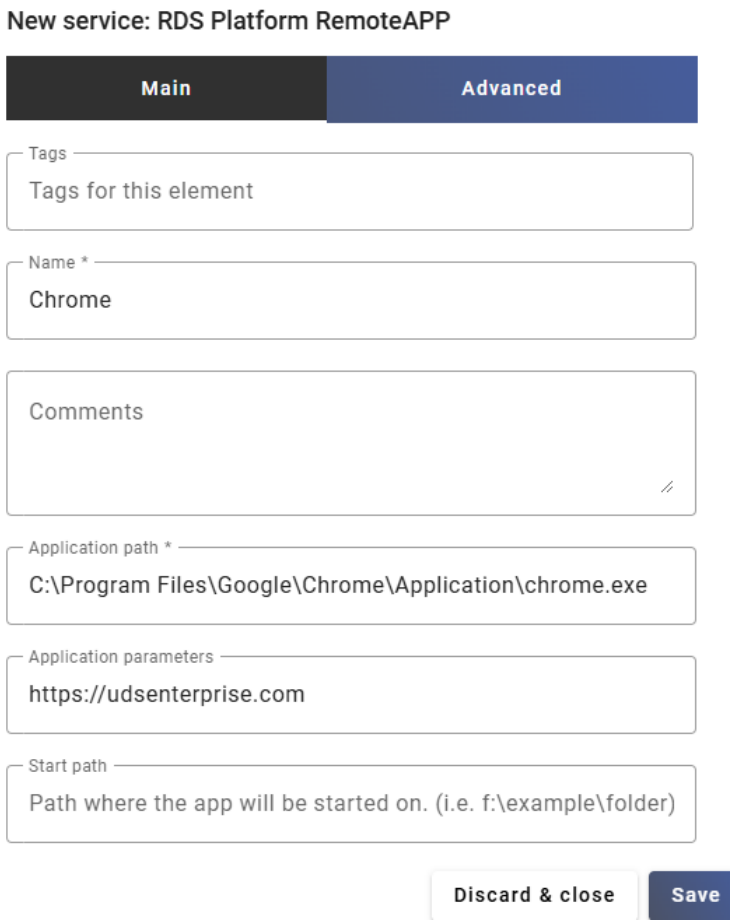
Name: Identifying name of the base service.

Application path: The execution path of the application, hosted on the RDS servers, that you want to virtualize. If there are multiple servers within the pool, the application must have the same path on all of them.

NOTE: In some cases, it is advisable to include the application call within a script. For example, a .bat script that calls the application and executes another command simultaneously, that script is .bat the one that we would indicate its path in this field.

Application Parameters: Parameters can be passed to any application in this field to customize the execution of the application. For example, if the application we virtualize is a browser, we can indicate the home page here.

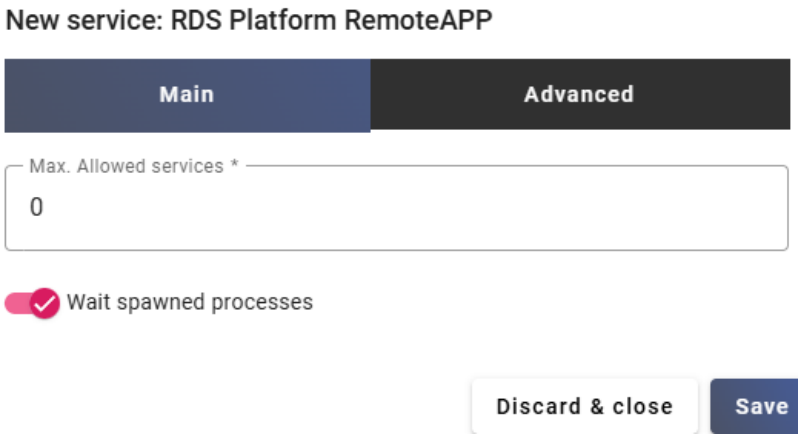
Start Path: Path where the application will be executed.



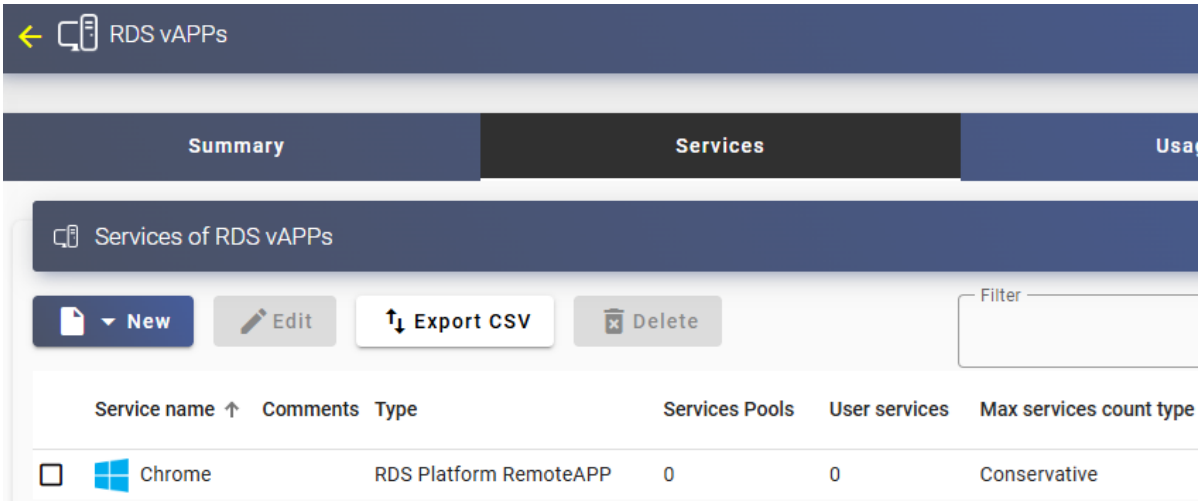
- Advanced:

Max. Allowed services: Maximum number of application sessions, which can run simultaneously (0 = unlimited).

Wait spawned processes: Wait until all processes derived from the application are complete before considering the application offline. Enabled in most applications.



Save the configuration and we will have a valid "***RDS Platform RemoteAPP***". We will be able to register all the "***RDS Platform RemoteAPPs***" that we need on the UDS platform.



Once we have the entire UDS environment configured (Services, Authenticators, OS Managers and Transports) and created the first "***Service Pool***", we will be able to give access to users to connect with the sessions of the virtual applications.

NOTE: UDS needs one or more external Windows 2016/2019/2022/2025 RDS servers to be able to connect and provide applications. It is necessary that the application server is properly configured and integrated with the RDS Agent/server or with the UDS Actor for proper functioning.

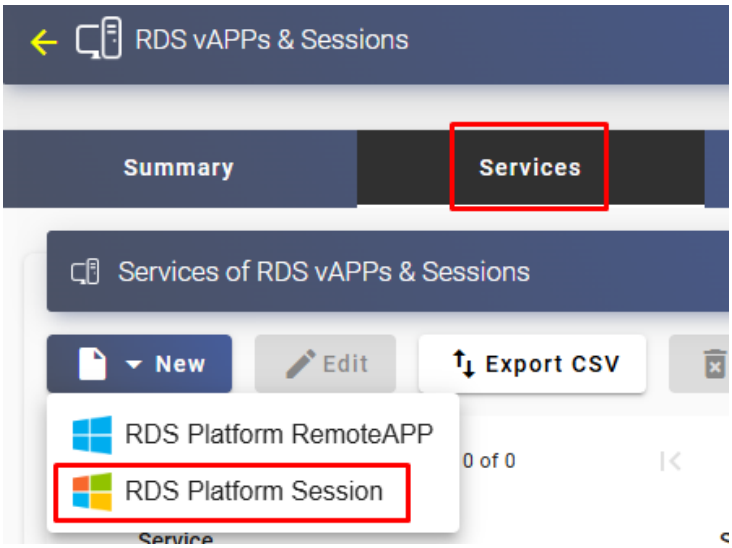
To configure and integrate RDS with UDS, you can consult this document: **Virtualization of Windows applications with UDS Enterprise** available in the documentation section of the website www.udsenterprise.com

3.1.12.3 Configure service based on "RDS Platform Session""

This base service will allow Windows desktop sessions to be published to users.

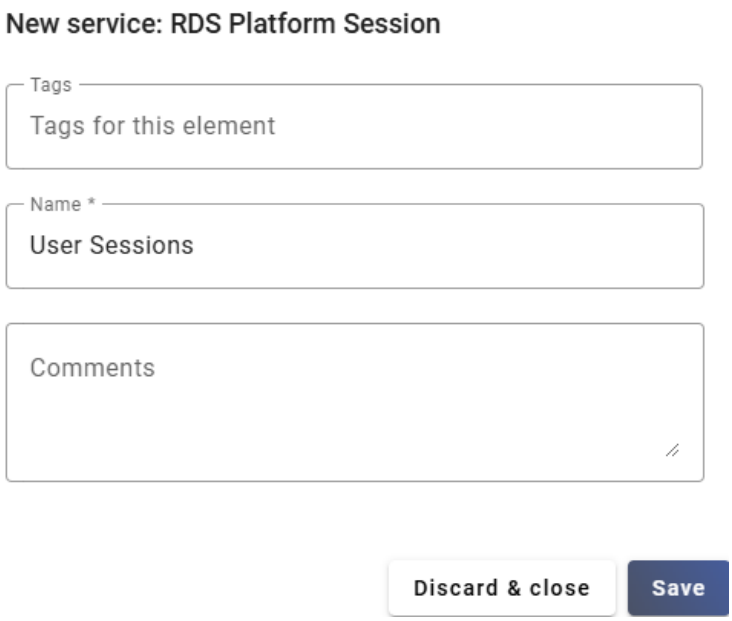
After you have created the service provider with a pool of servers, consisting of one or more session servers, and the RDS Server Agent (***UDSRDSServerSetup-x.x.x.exe***) is installed, you must create base services of type "***RDS Platform Session***".

To create base services of type "***RDS Platform Session***" we access the "***Service Providers***", select the "***Services***" tab, click on "***New***" and select "***RDS Platform Session***".

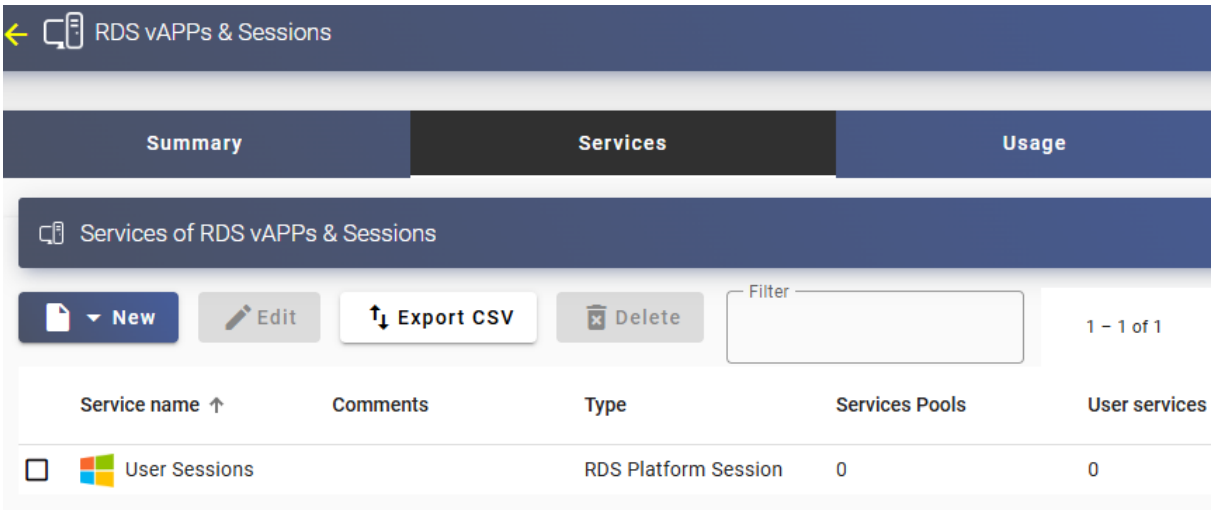


The minimum parameters to be configured in this type of service are:

Name: Identifying name of the base service.



Save the configuration and we will have a valid "***RDS Platform Session***". We will be able to register all the "***RDS Platform Sessions***" that we need on the UDS platform.



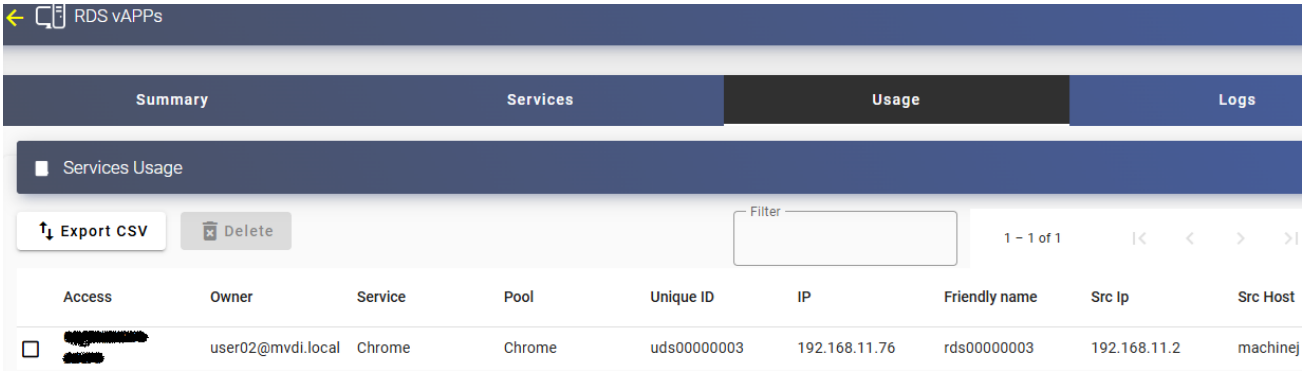
Once we have the entire UDS environment configured (Services, Authenticators and Transports) and created the first "***Service Pool***", we will be able to give access to users to connect with the desktop session sessions.

NOTE: UDS needs one or more external Windows 2016/2019/2022/2025 RDS servers to be able to connect and provide sessions. The session server must be properly configured and integrated with the RDS Agent/server for proper operation.

3.1.12.4 Service usage summary and error logs

Within a "***Service Provider***" we will have a tab called "***Usage***" that will allow us to have a quick view with detailed information and perform basic actions of the services deployed in this "***Service Provider***".

To access this information on the use of the services, we access the "***Service Providers***" and select the "***Usage" tab.***":



We will be able to visualize:

Access: Date and time the service was accessed.

Owner: User and authenticator assigned to the service in usuario@autenticador format.

Service: Name of the base service belonging to the "***Service Provider***" from which the service was generated.

Pool: Name of the "***Service Pools***" to which the service belongs.

Unique ID: Service ID.

IP: The IP address of the application server that provided the application.

Friendly name: Generic name of the service.

Src Ip: IP address of the connecting client accessing the service.

Src Host: DNS name of the connecting client accessing the service. If you are unable to access this name, your IP address will be indicated.

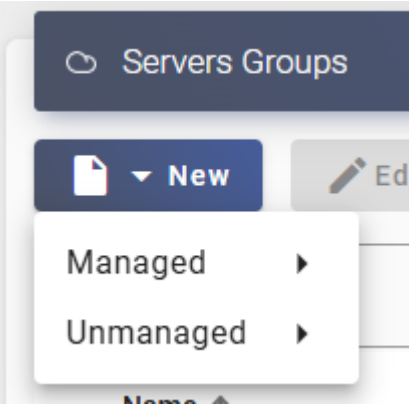
It will also be possible to select one or more services to proceed with their removal.

The "***Logs***" tab will show us information about a possible problem that occurred in the "***Service Provider'***":

3.2 Servers

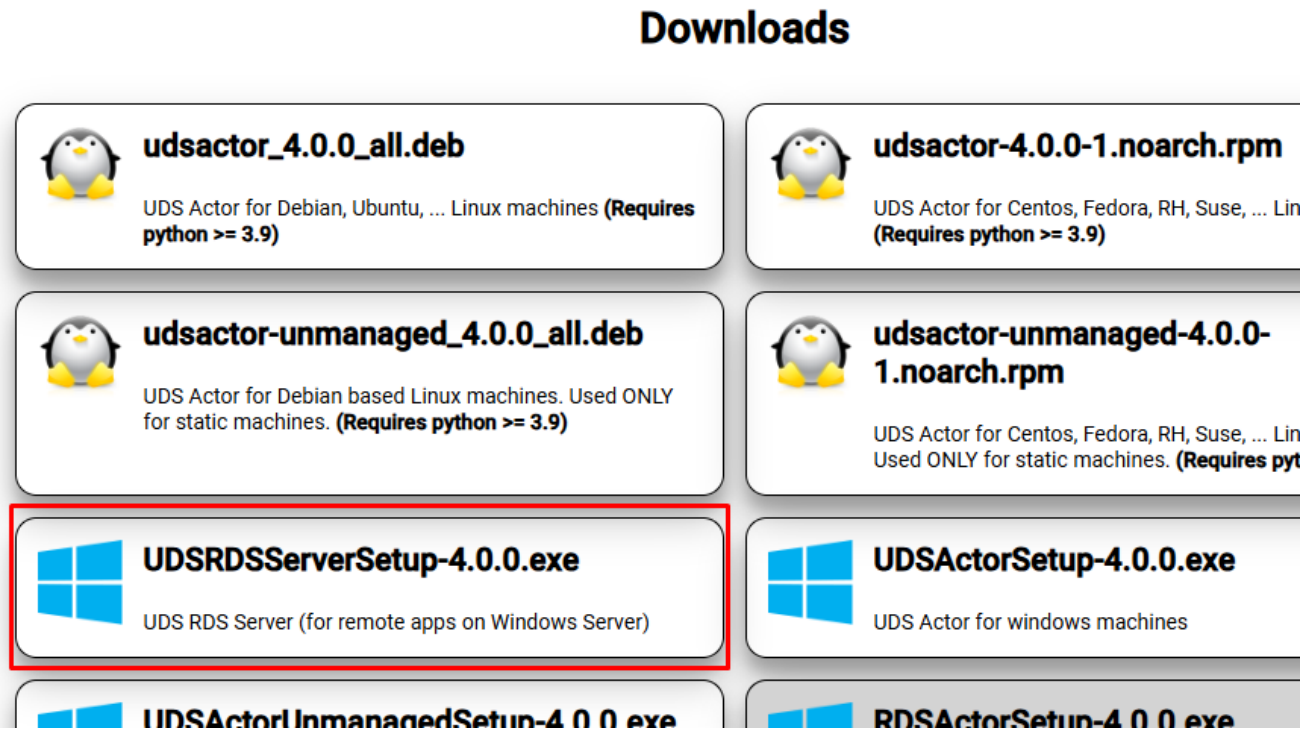
A "***Servers Groups***" will allow us to define groups of servers that can be used in service providers such as "***RDS Platform Provider***" and "***Static IP Machines Provider***". These groups of servers will be registered as independent objects and can be used by a provider together, allowing configurations in high availability, different groups of IP addresses or names for direct assignment, etc.

UDS currently supports two types of server groups, "***Managed***" and "***Unmanaged'***":



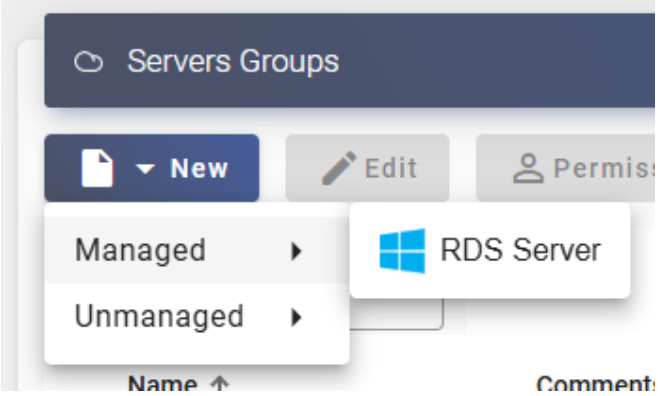
3.2.1 Managed

A group of servers of type "***Managed***", it is necessary that they have the server agent UDSRDSServerSetup-x.x.x.exe, available in the download area of UDS agents and actors:



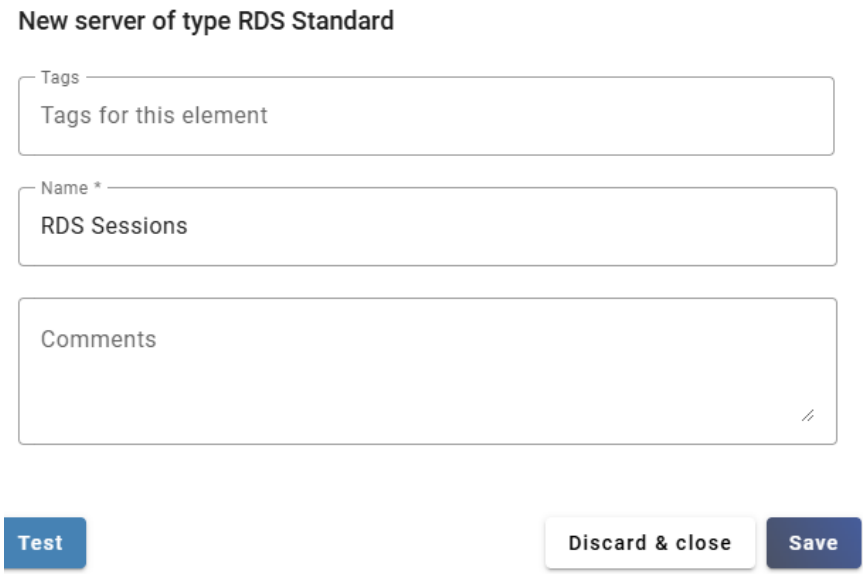
This type of group is used to manage servers used to provide virtual application sessions and desktop sessions to users and will be the recommended option from UDS version 4.0 onwards. Through this connector and using the indicated server agent, the system will be able to perform certain advanced actions of type: control of user balancing by load, session detection, etc...

The groups of servers supported by the "***Managed***" mode will be those destined for virtual application sessions and desktop sessions:

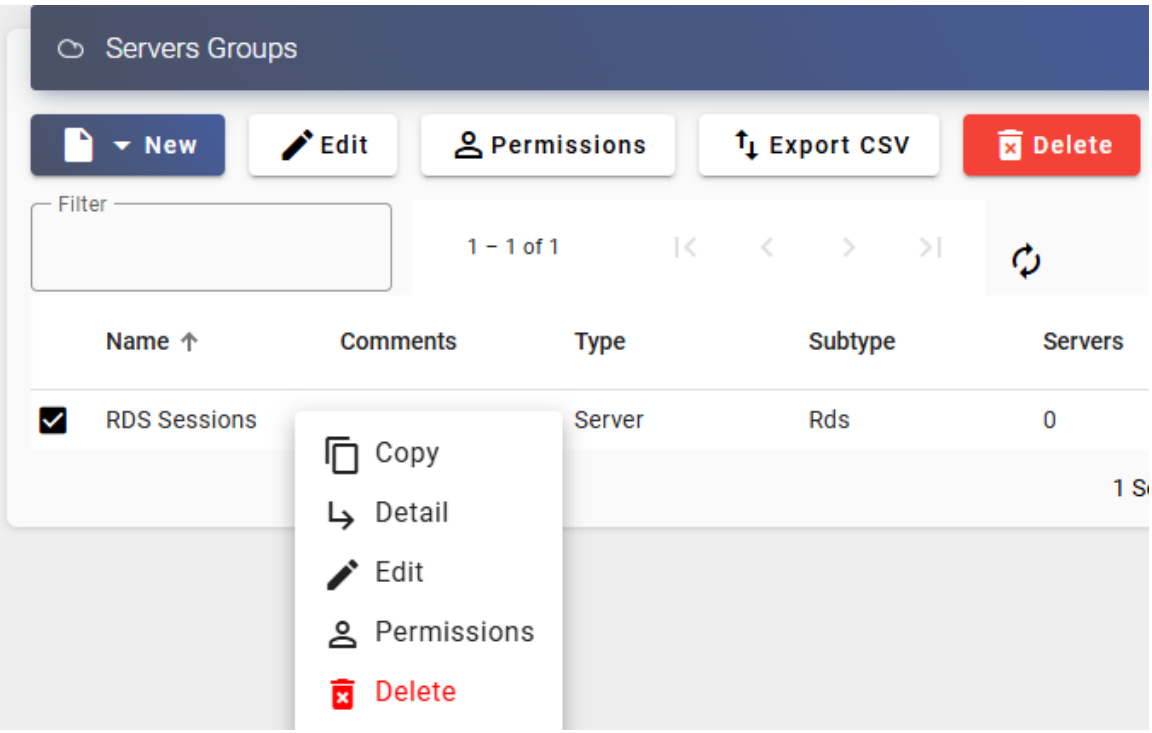


To register a group of servers of type "***Managed***" for application and session servers, click on "***New***", mark "***Managed***" and select "***RDS Server***".

We will indicate a name and a description if necessary and click on "Save":

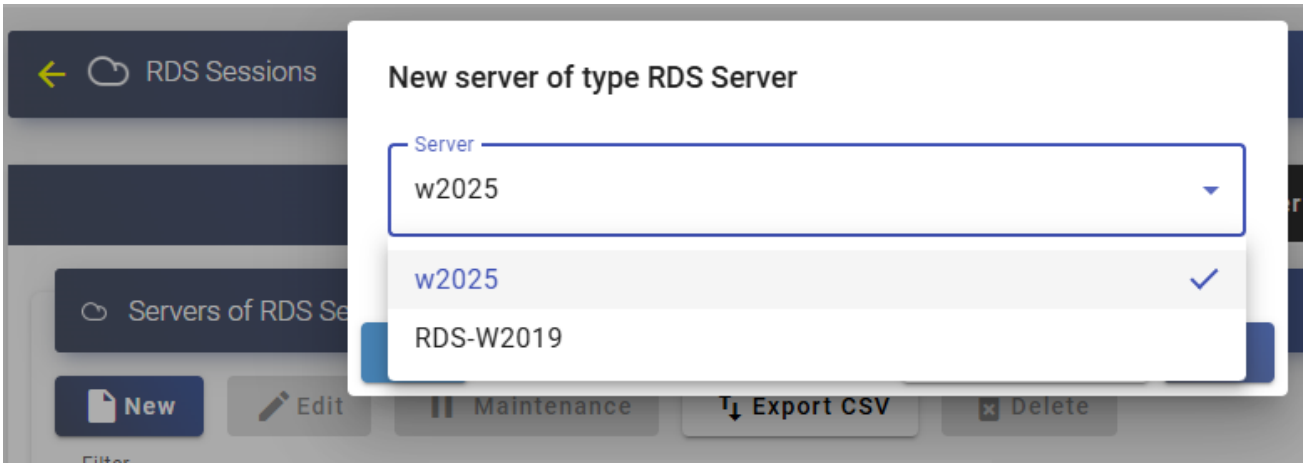


Once created, we double-click on the item or in the menu select "***Detail***":



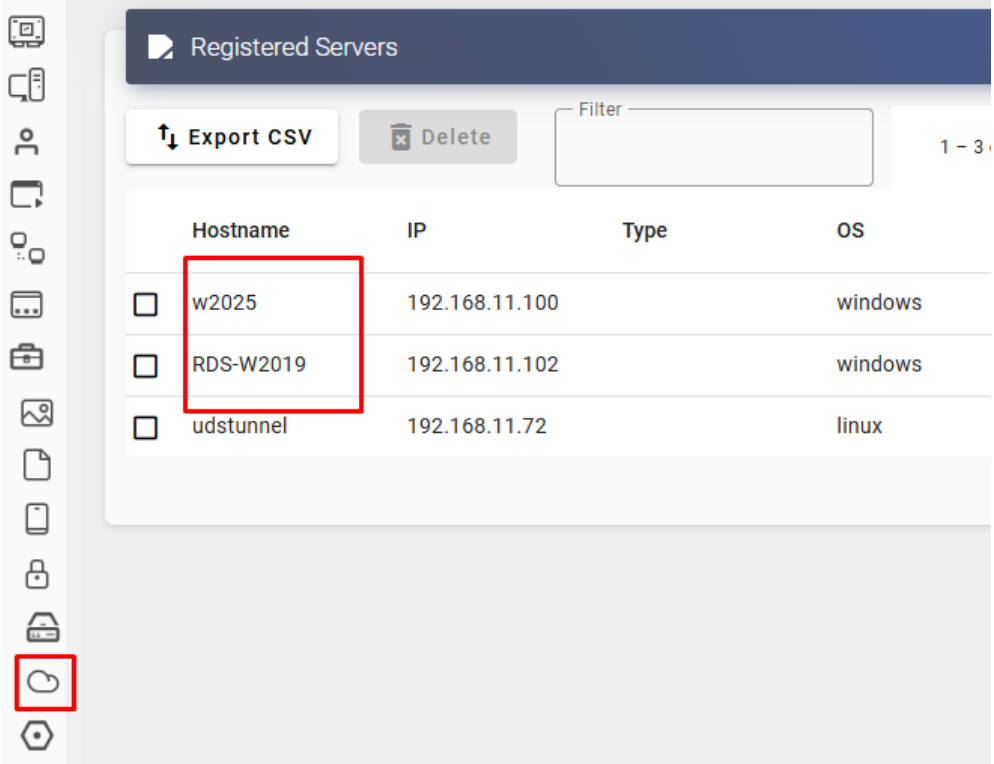
Now we will indicate all the servers that will make up the "***Server Group***", these servers must have the server agent UDSRDSServerSetup-x.x.x.exe (available in the download area of UDS agents and actors) installed and configured to appear in the list of available servers.

Click on "***New***", select the server (DNS name of the server) and click on "***Save***"

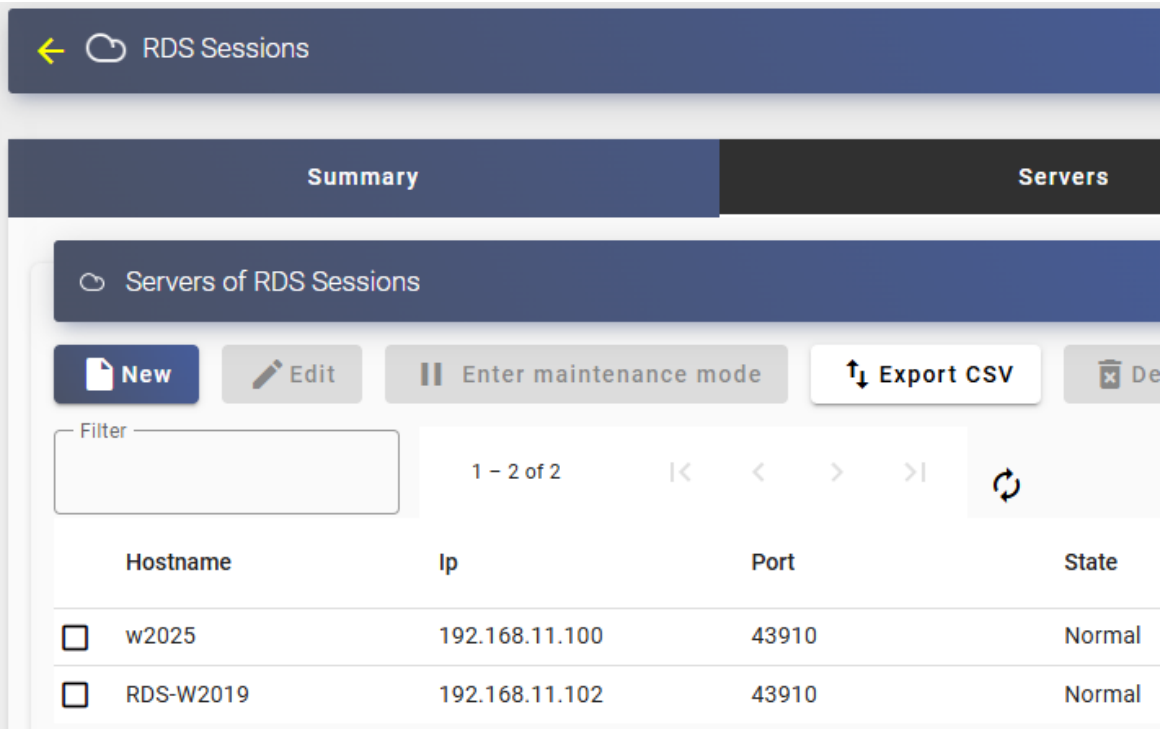


NOTE: Servers cannot be registered manually, they must be registered prior to the installation of the server agent in order to appear as selectable.

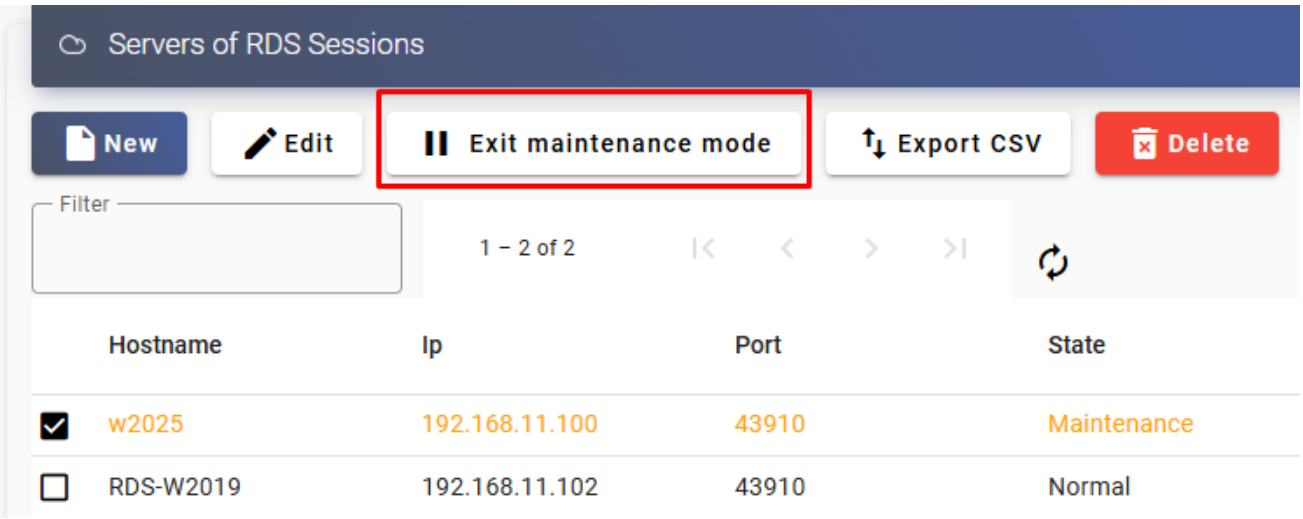
In the "Tokens" section within the "Tools" menu, if we access the "Registered Servers", the servers to be used must be registered in order to appear on the list.



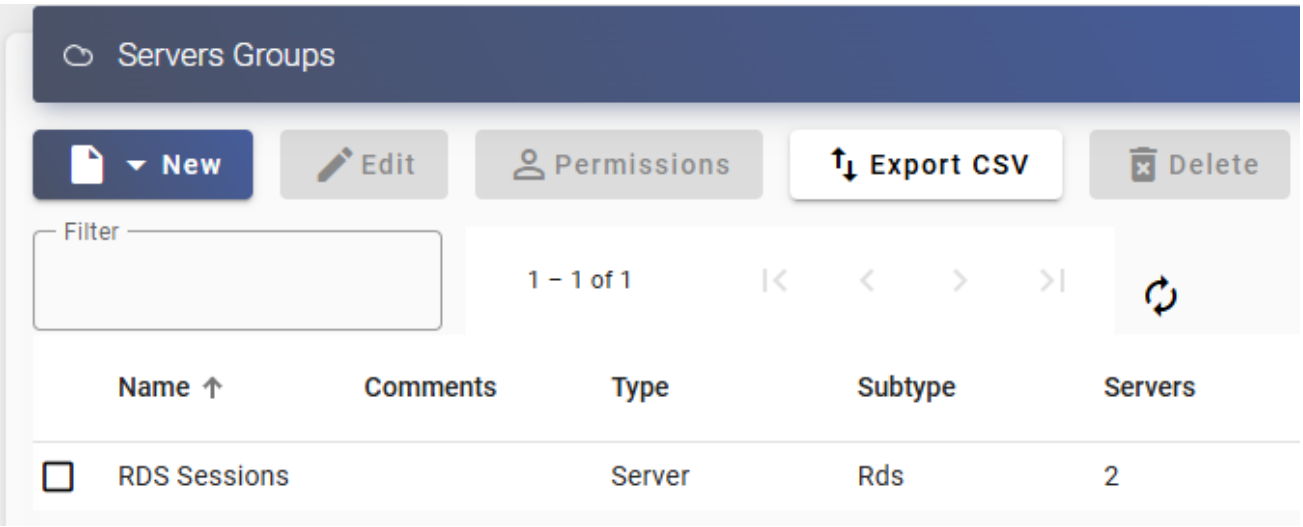
We can add all the servers that we want to form the "***Servers Groups***"".



If at any time it is necessary to perform some maintenance work on the server and it will not be available, we can delete the server or put it in mono maintenance so that it is not used.



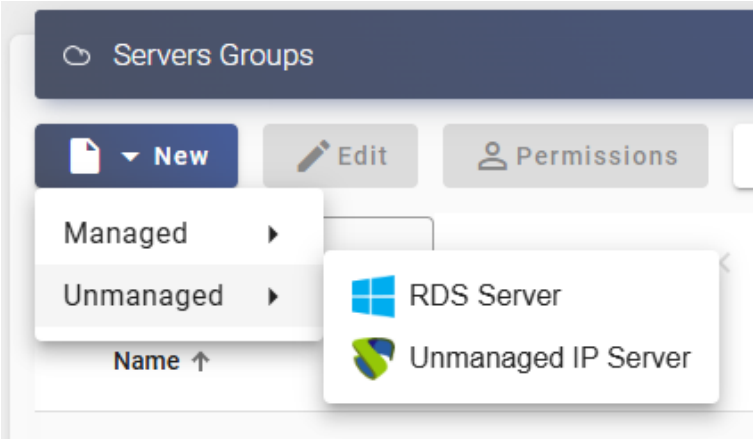
Once the configuration of the "*Server Group" is completed, it can be used by the service providers of the type "RDS Platform Provider*" and "*Static IP Machines Provider'*".



3.2.1 Unmanaged

A group of "***Unmanaged***" servers can be managed by the UDS actor for RDS servers that provide virtual application sessions (RDSActorSetup-x.x.x.exe) or they can have no actor installed if they are used for the "Static IP" provider providing remote access and remote access.

Depending on which service provider we want to assign the server group to, we will select one of the supported types:

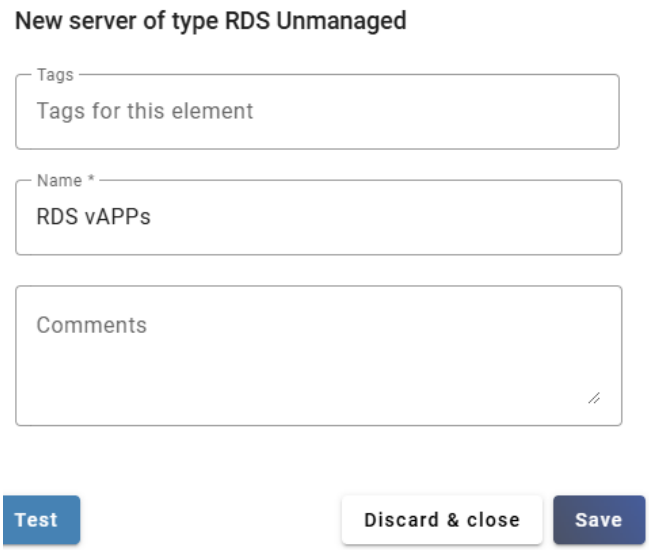


3.2.1.1 RDS Server

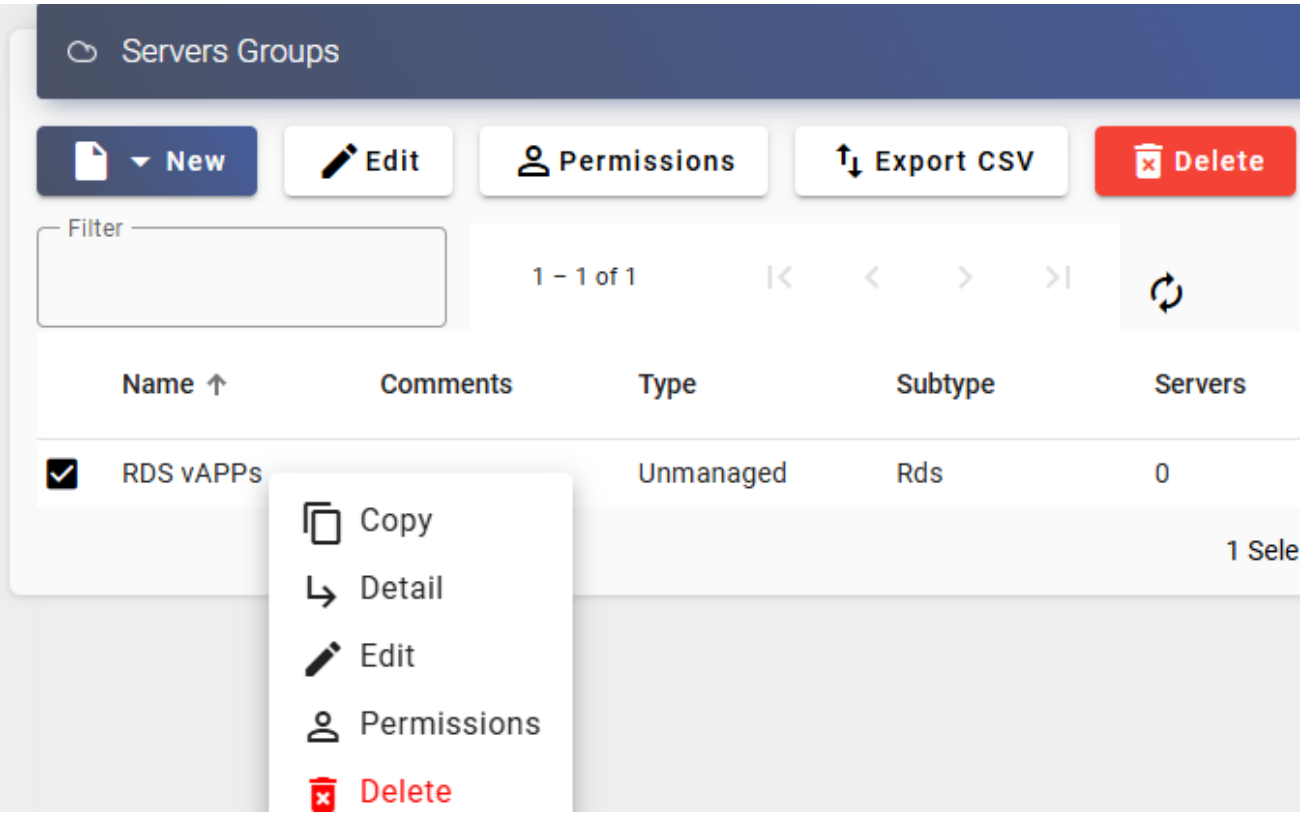
We will use this group of "***Unmanaged***" servers to configure application servers that we will use with the RDS ***Platform Provider*** and that have the "RDS Actor" installed and configured.

NOTE: It is recommended to use "RDS Server" server pools of the "Managed" type since the "Unmanage" type will be discontinued in future versions.

Once selected, we will indicate a descriptive name and click on "***Save***"

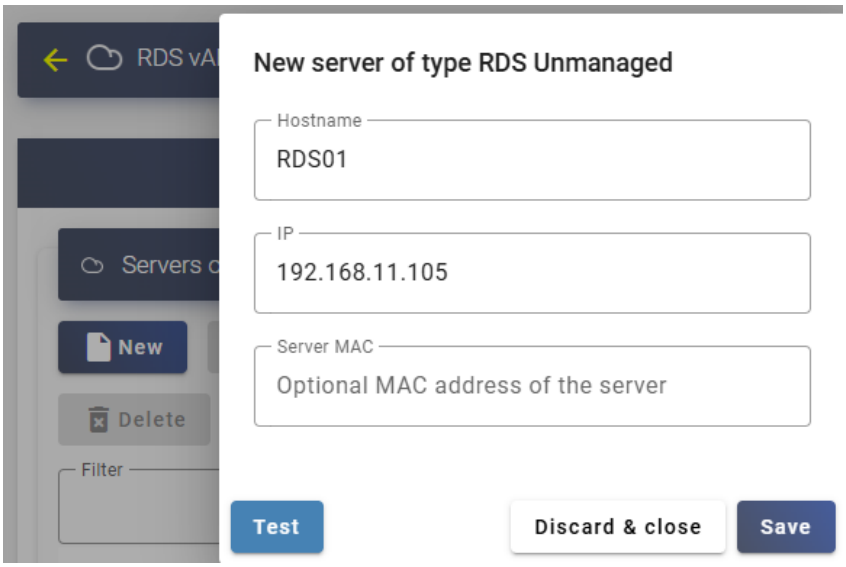


Once created, we double-click on the item or in the menu select "***Detail***":

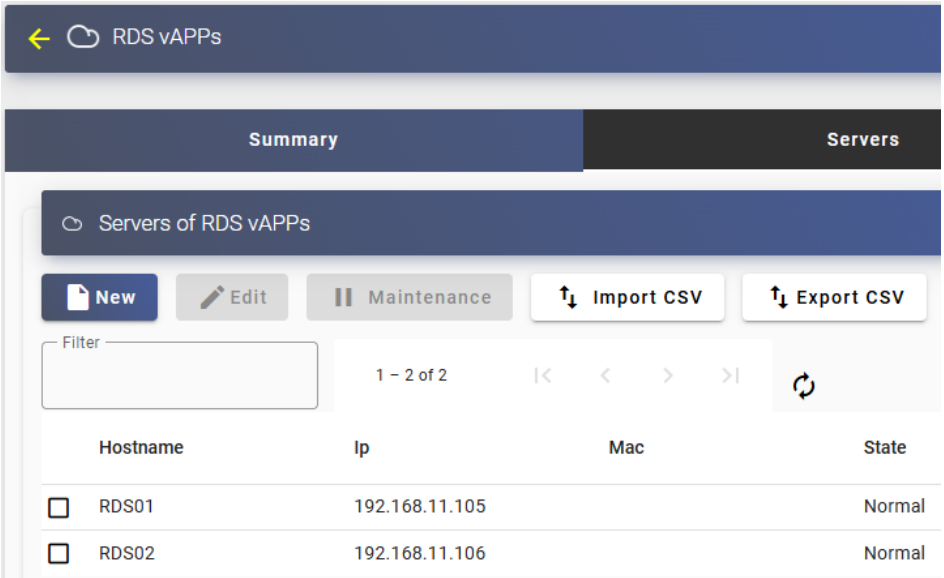


Now we will indicate all the servers that will form the "***Server Group***", these servers must have the RDSActorSetup-x.x.x.exe actor (available in the UDS actors download area) installed and configured.

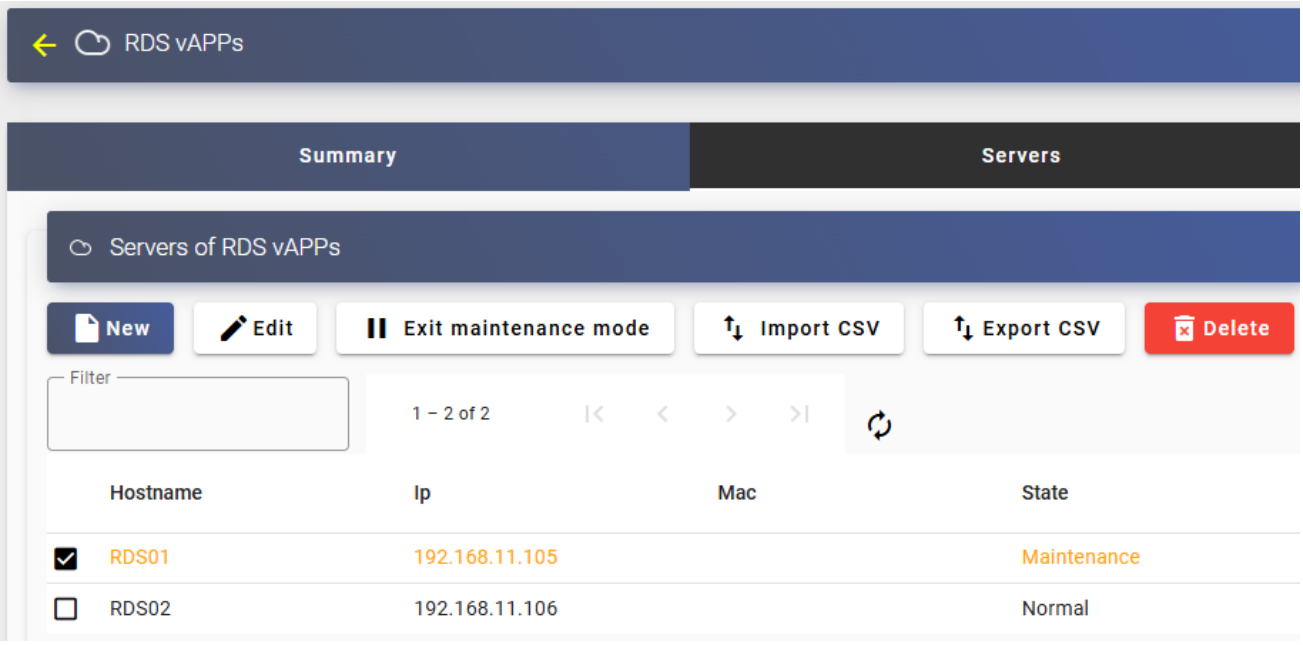
Click on "***New***", indicate a name for the element, enter the IP address or name of the RDS server and click on "***Save***"



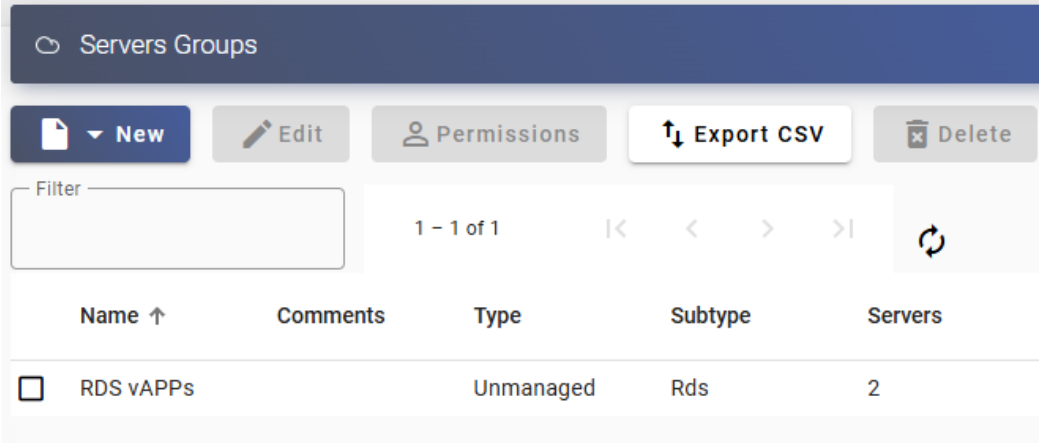
We can add all the servers that we want to form the "***Servers Groups***"".



If at any time it is necessary to perform some maintenance work on the server and it will not be available, we can delete the server or put it in mono maintenance so that it is not used.



Once the configuration of the "***Server Group" is completed, it can be used by*** the service provider of the "***RDS Platform Provider" type***".

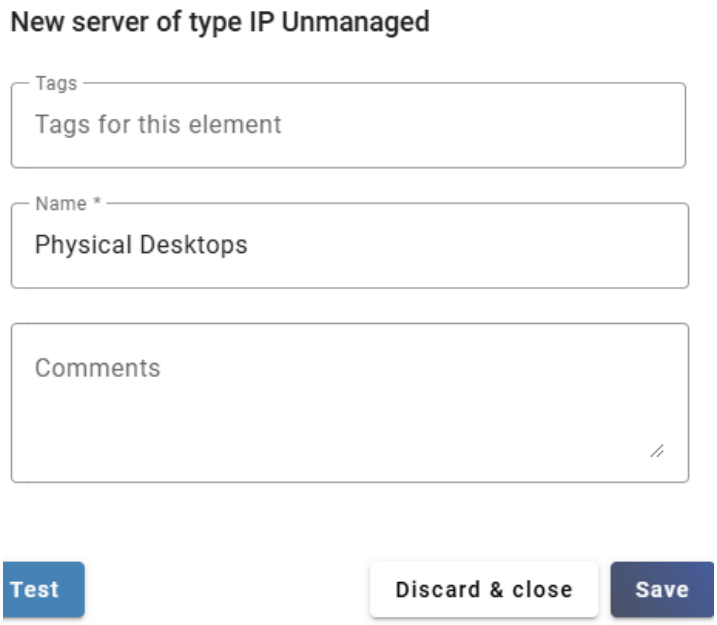

Servers Groups
New
Edit
Permissions
Export CSV
Delete
Filter
1 – 1 of 1
Name
Comments
Type
Subtype
Servers
RDS vAPPs
Unmanaged
Rds
2

3.2.1.2 Unmanaged IP Server

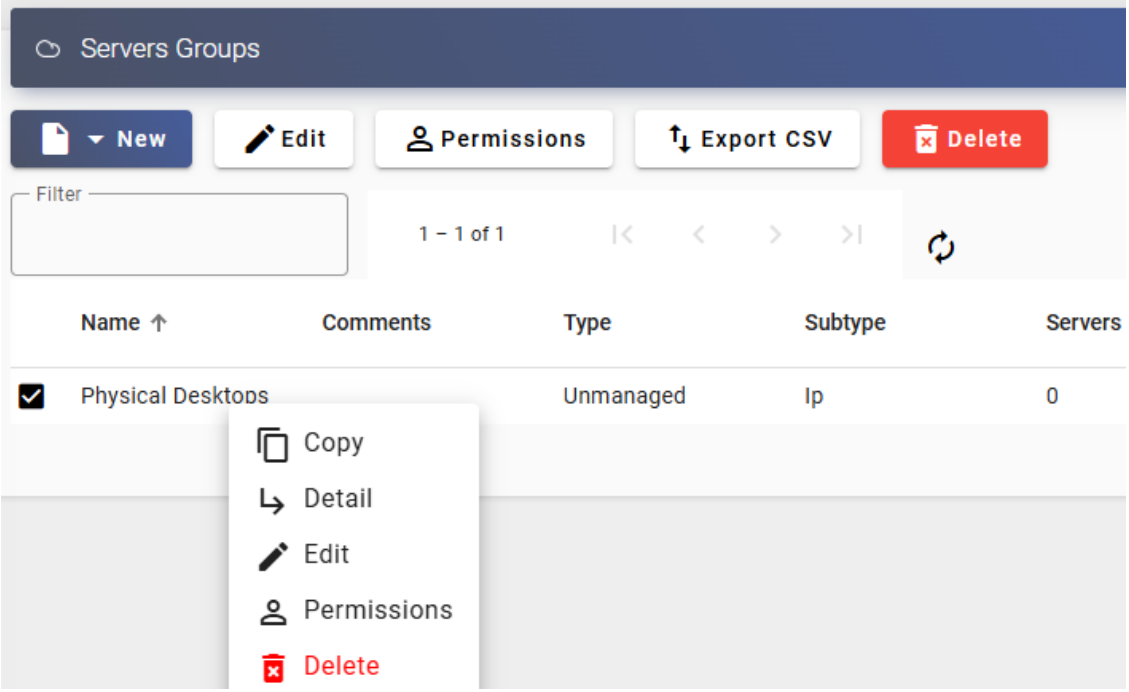
We will use this group of servers of type "***Unmanaged***" to access machines that use the service provider "***Static IP Machines Provider***" and that may have the "UDSActorUnmanaged" actor or not installed.

NOTE: If the UDS Unmanaged actor is installed, the machine assignment can be dynamic. If we do not install the actor, the system will not be able to detect the user's logout and the assignment will be static (one user same machine).

Once selected, we will indicate a descriptive name and click on "***Save***"

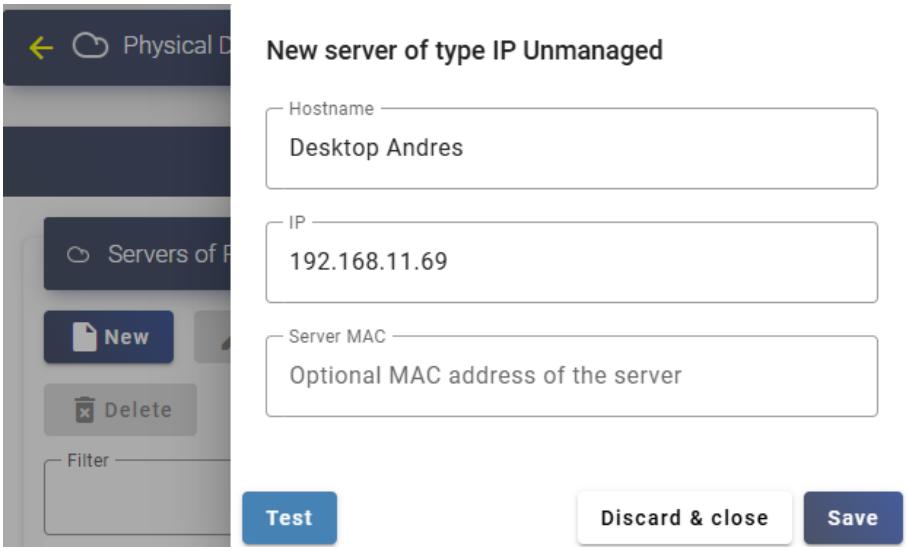


Once created, we double-click on the item or in the menu select "***Detail***":

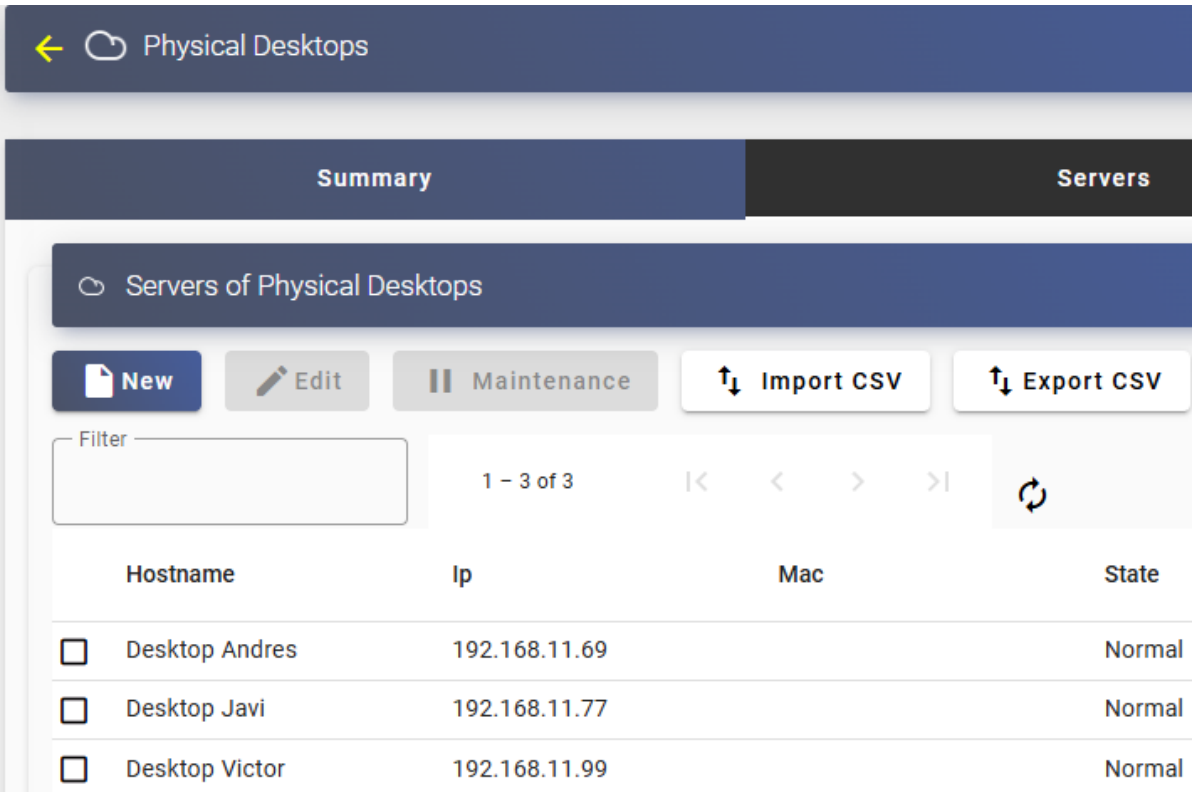


Now we will indicate all the servers that will form the "***Server Group***", these servers can have the UDS Unamnaged actor depending on their purpose.

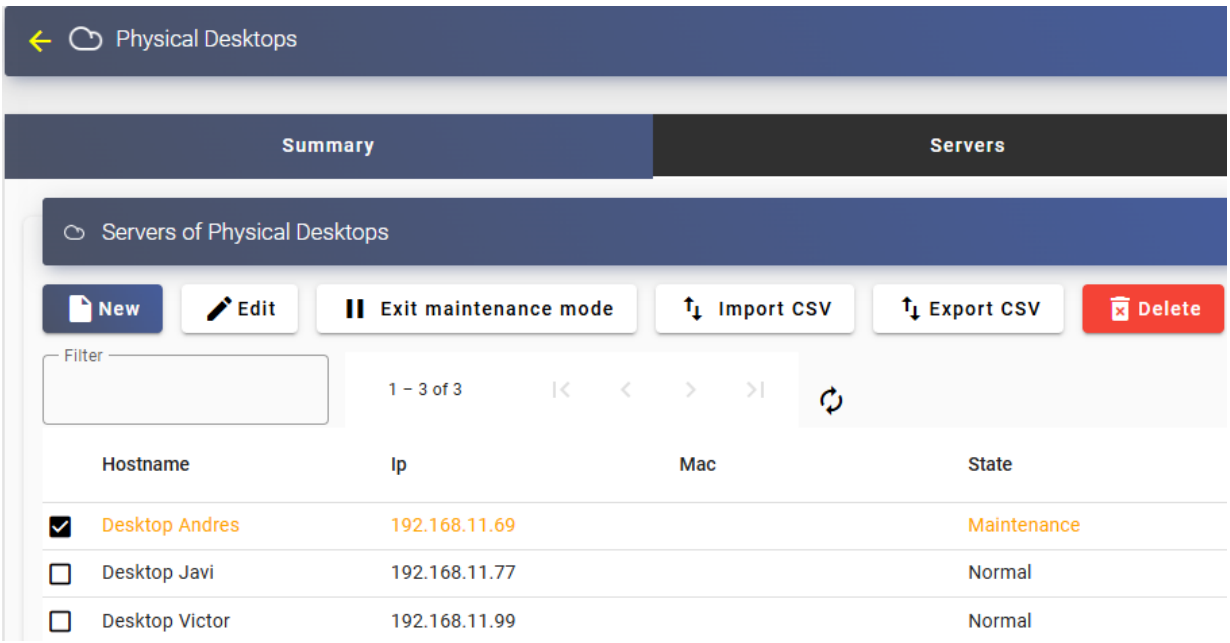
Click on "***New***", indicate a name for the element, enter the IP address or name of the machine and click on "***Save***"



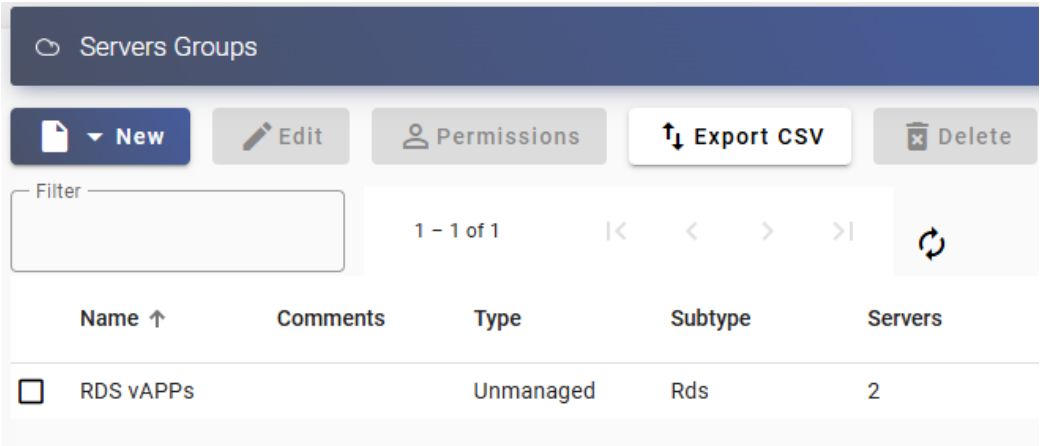
We can add all the servers that we want to form the "***Servers Groups'***".



If at any time it is necessary to carry out some maintenance work on the machine and it will not be available, we can remove it or put it in mono maintenance so that it is not used.

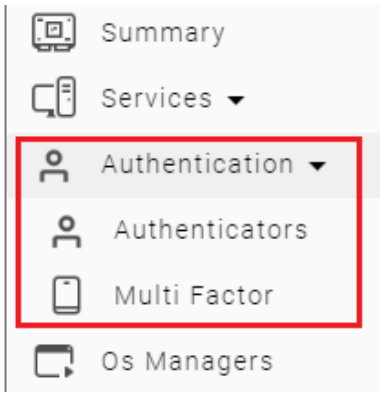


Once the configuration of the "***Server Group" is completed, it can be used by*** the service provider of the "***Static IP Machines Provider" type***".


Servers Groups
New
Edit
Permissions
Export CSV
Delete
Filter
1 – 1 of 1
Name
Comments
Type
Subtype
Servers
RDS vAPPs
Unmanaged
Rds
2

4. AUTHENTICATION

This element will manage any user directory that accesses the UDS environment, whether they are external/internal authentication systems and also the level of security applied, such as two-factor authentication systems.



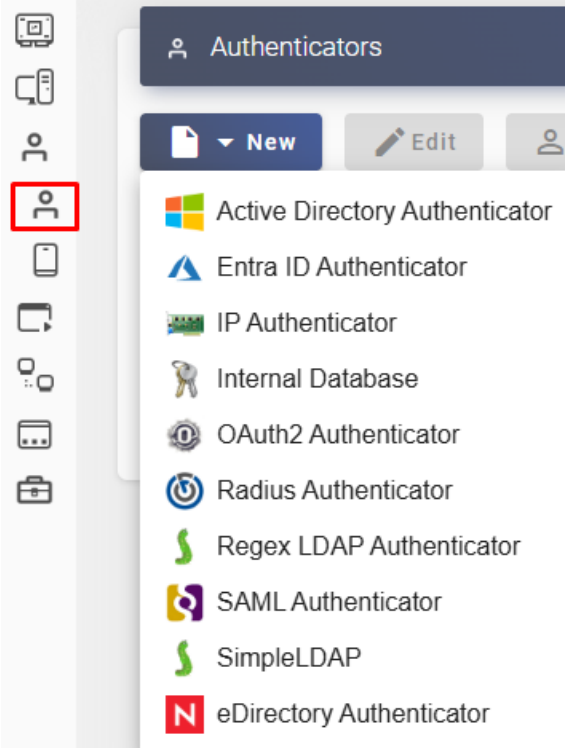
4.1 Authenticators

An "***Authenticator***" is a fundamental component within a UDS platform. It validates users in the login portal and provides users and user groups to which permissions can be assigned to connect to the different desktop services and virtual applications.

An ***"Authenticator***" is not a necessary component for the creation of a "***Service Pool***", but it is for its use. If at least one is not assigned, there will be no users who can make connections to the platform's services.

You can choose between different types of "***Authenticators,"*** whether they are external (Active Directory, eDirectory, OpenLDAP, etc...) or internal type (Internal Database and IP Authentication).

Currently UDS supports the following "***Authenticators'***":



NOTE: We will be able to register all the authenticators, of the same type or different, that we need.

Safe Employment Procedure:

Download and install the certificate on the UDS server in order to enable SSL communications between the UDS Server and the authenticator.

Do not use the "Administrator" user of the Authenticator, create one for this purpose.

Procedural recommendations:

It is recommended to have more than one authenticator created or at least one backup.

In the case of external authenticators of the type: federated (SAML, OAuth, Entra ID, ...) or IP, login is done automatically and if there is an error in them (and they have the lowest priority), it could enter a "loop", making it impossible to access UDS.

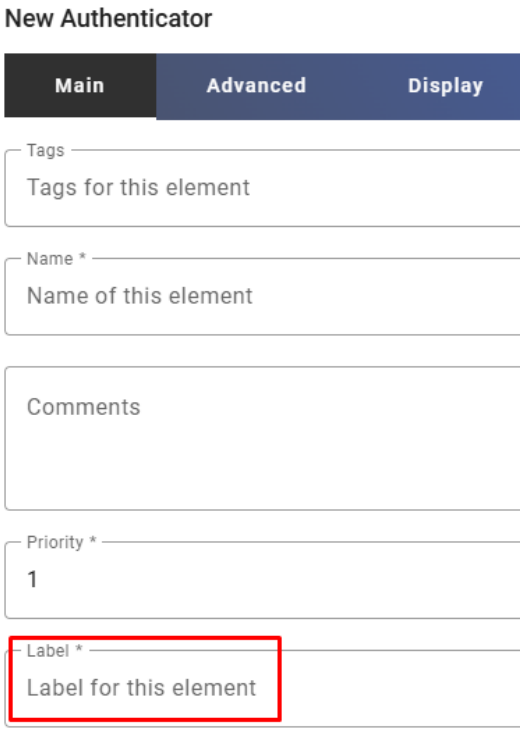
To solve this problem, we can disable these two types of authenticators (federated or IP) and access with another authenticator or directly with the super-user that was created in the configuration wizard of the UDS Server component. To do this, we must indicate in the access URL adding /disabled at the end, leaving the URL as this example:

- https://*Server _UDS*/uds/page/login/**disabled**

If we have more than one authenticator and they are in a visible state, a drop-down menu will be enabled in the login portal to select which of them the system should validate the user. We can also make use of the tags of these, since when creating an authenticator as a mandatory requirement is to create a.

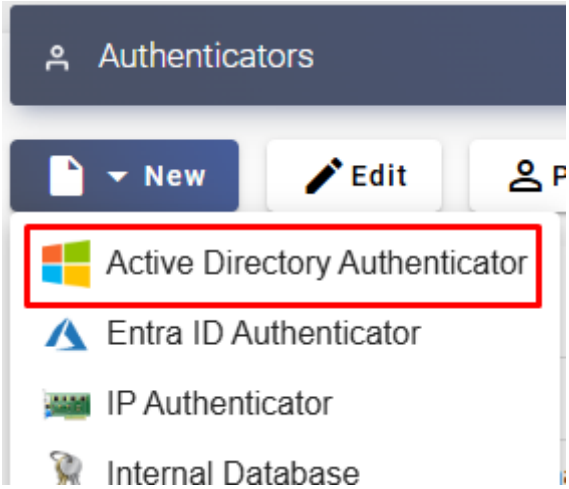
To access the different authenticators through their tag and avoid using the drop-down menu, we must launch a URL like the following:

- https:// *Server _UDS* /uds/page/login/**Label-authenticator**



4.1.1 Active Directory

This external authenticator allows access to the UDS environment to groups of users belonging to an Active Directory.



En un "***Active Directory Authenticator***" los parámetros mínimos a configurar son:

- Main:

Name: Name of the authenticator.

Priority: The priority that this authenticator will have. If there are several authenticators, the lower their priority, the higher they will appear in the list of authenticators available in the UDS login portal. This field supports negative values.

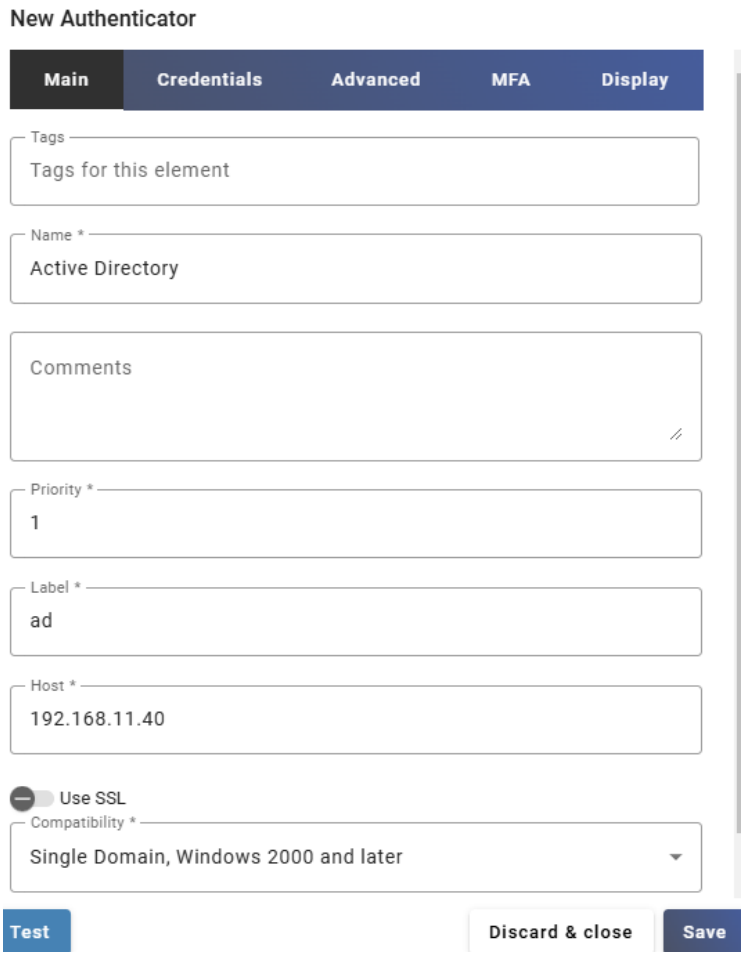
Label: Enables direct validation on this authenticator. It allows the user to validate themselves in the login portal with said authenticator, even if the UDS environment has more authenticators. To do this, we will have to access the UDS login portal with the following format: ***UDSServer/uds/page/login/label*** (For example: https://UDSServer/uds/page/login/ad).

Host: IP address or Active Directory server name (we can directly indicate the name of the AD domain, for example: mvdi.local).

Use SSL: If enabled, an SSL connection, via port 636, will be used against the authenticator. (Recommended to provide greater security to the platform).

NOTE: Windows Server 2025 by default only accepts connections via SSL

Compatibility: We will select the compatibility level of our Active Directory server.

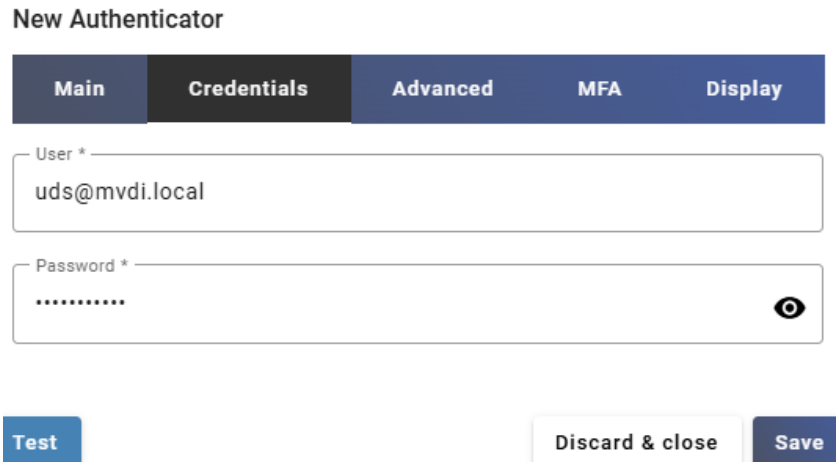


- Credentials:

User: User with read permissions on authenticator in format: ***user@domain***

Password: User Password

Safe Operating Procedure: Passwords must be of sufficient length and include upper and lower case, numbers and special characters.



- Advanced:

Override Base: This field allows you to use a search base different from the one used by default (the default search base is extracted from the user indicated in the "***User***" field in the "***Credentials***" section). We will only fill in this field if we need to indicate a specific user search base (for example: dc=mvdi2,dc=local).

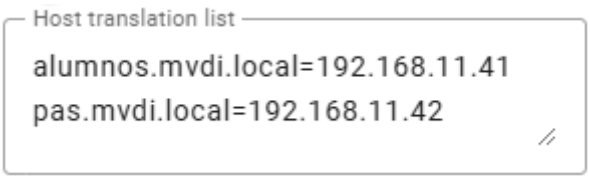
Default domain: Domain that will be added to the user's name if the user has not indicated it when logging in to the access portal.

Backup server: Authentication servers that will be used in case the main server indicated in the "***Host***" field of the "Main" section ***is not available***. If multiple are indicated, they must be separated by commas.

Timeout: Timeout time of the connection to the authenticator.

Verify SSL: If active, the authentication server certificate will be verified. The certificate used for verification must be included in the "Certificate" field.

Host translation list: Used to force a DNS resolution, indicating for example the name of a subdomain and its IP address:



Certificate: Certificate from the authentication server to check if the "***Verify SSL***" field is active.

Ignore domains: Servers that will be ignored in the search for users separated by a comma.

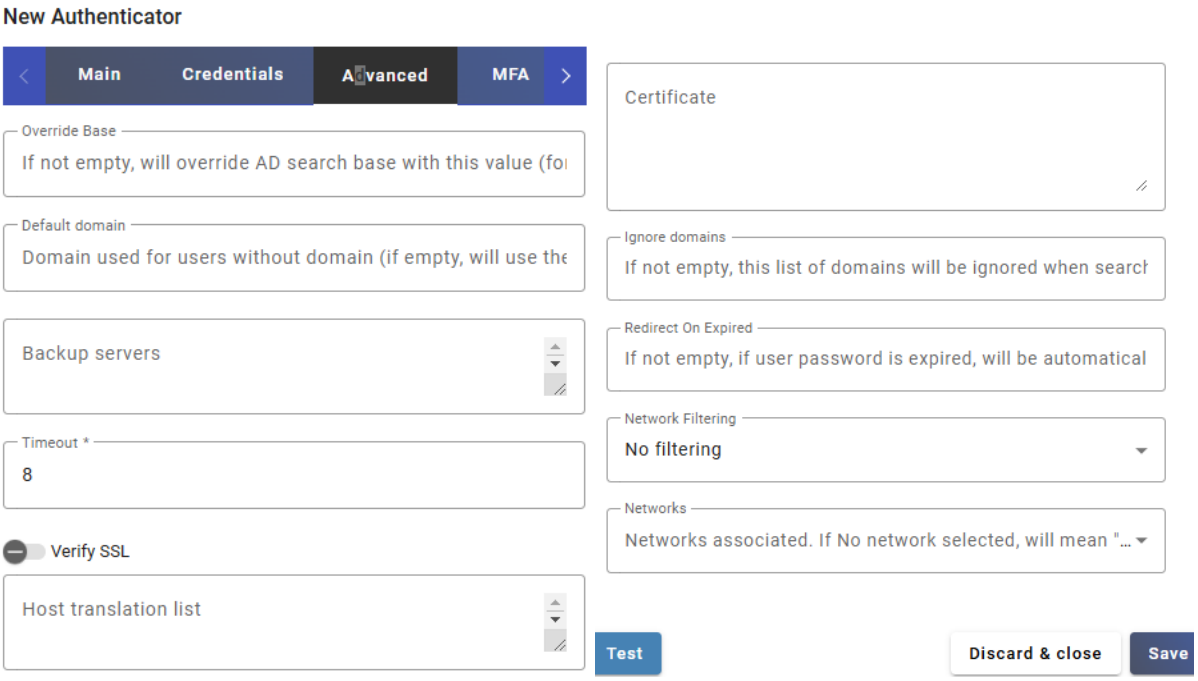
Redirect On Expired: If during authentication it is detected that a user has an expired password, they are automatically redirected to the URL indicated in this field. If the field is empty, the user with the expired password will receive an access error.

NOTE: The password must pre-exist and be reported as an expired password to allow automatic redirection to the URL provided.

Network Filtering: We will indicate the action to be carried out on the networks indicated in the "***Networks***" section. We may not perform filtering: "***No filtering***", allow only the indicated networks: "***Allow select networks***" or not allow only the indicated networks: "***Deny selected networks***".

Networks: Network ranges, subnets or IP addresses indicated in the "***Networks***" section of the "Connectivity" section. Used in conjunction with the "***Network Filtering***" field to allow or disable authentication of users based on their network location.

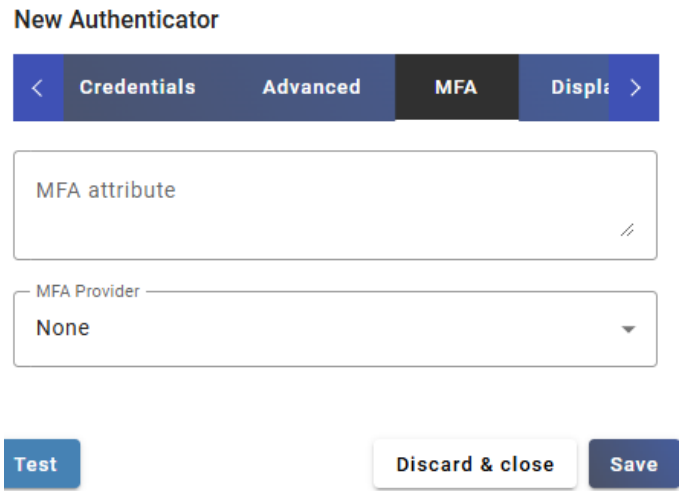
NOTE: If none is selected, the authenticator will be visible on any network.



- MFA:

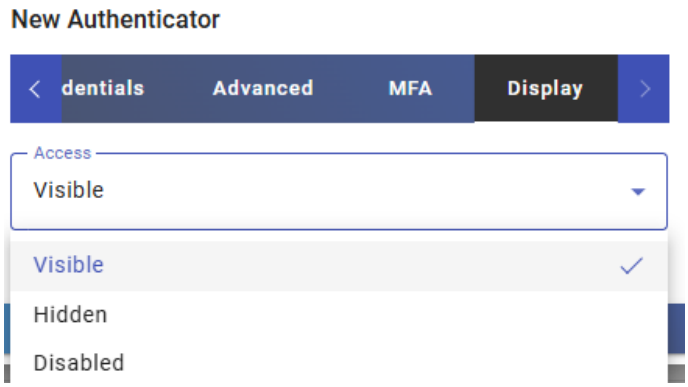
MFA attribute: We will use this field to indicate which user attribute will be used by the MFA system to notify you of your token, usually the attribute where the email or phone number is indicated, for the MFA systems for sending SMS and email. If you use the TOTP system, you will not need to fill in this field, but for all other supported systems, you will need to.

MFA Provider: Two-factor authentication (MFA) system applied to the authenticator. You must be previously registered in the "***Authentication***" section "***Multi Factor'''***".



- Display:

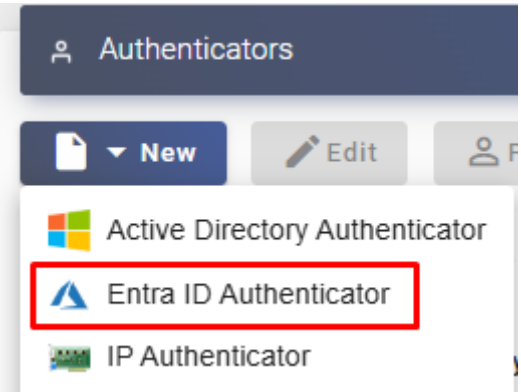
Access: Allows you to indicate whether the authenticator will be visible and active to users: "***Visible***", if it will be active, but not visible: "***Hidden***" or if it will be completely disabled so that it cannot be viewed or used: "***Disabled'''***".



By using the "***Test***" button we will check that the connection with the authenticator is made correctly.

4.1.2 Entra ID

This external authenticator allows you to give access to the UDS environment to groups of users belonging to an Entra ID (Azure Active Directory).



NOTE: It is important to note that as authentication is not carried out in the UDS login portal (this type of authenticator validates users directly in its portal and once the validation is done, if it is correct, it redirects the user to the UDS services portal), it is not possible to redirect credentials to the services, since the environment does not know the credential.

In an "***Entra ID Authenticator***" the minimum parameters to be configured are:

- Main:

Name: Name of the authenticator, cannot contain spaces.

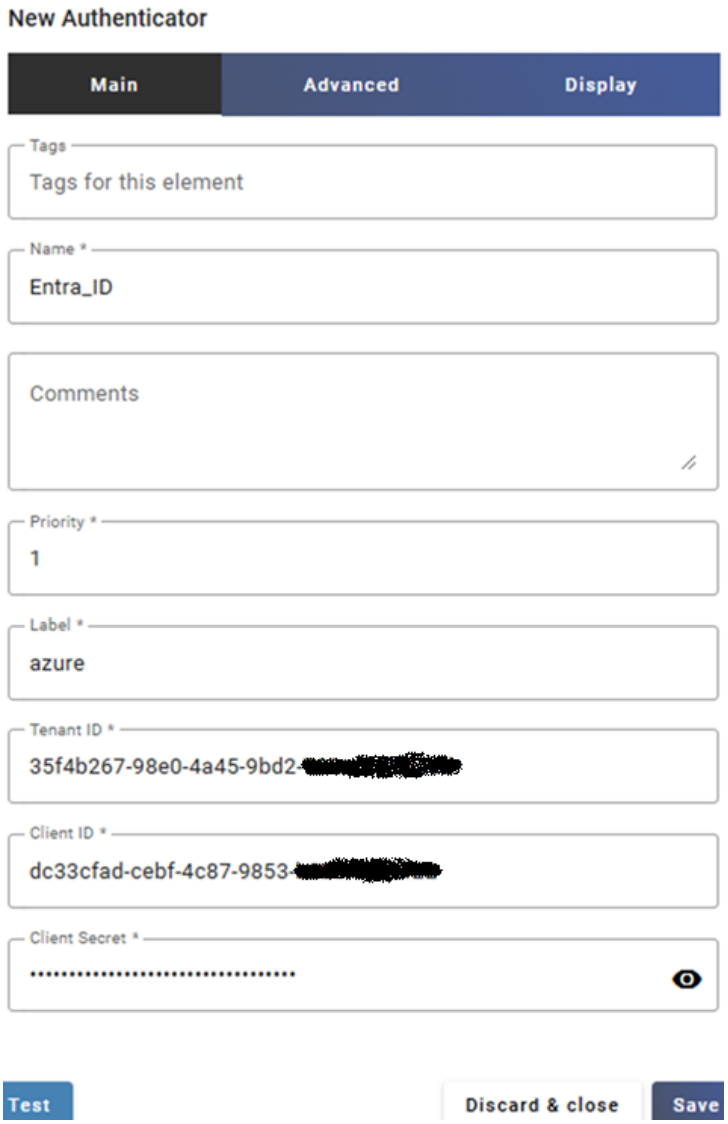
Priority: The priority that this authenticator will have. If there are several authenticators, the lower their priority, the higher they will appear in the list of authenticators available in the UDS login portal. This field supports negative values.

Label: Enables direct validation on this authenticator. It allows the user to be validated in the login portal with said authenticator even if the UDS environment has more authenticators. To do this, we will have to access the UDS login portal with the following format: ***UDSServer/uds/page/login/label*** (For example: https://UDSServer/uds/page/login/azure).

Tenant ID: This value can be obtained in our Azure account, from the "***Microsoft Entra ID***", "***Properties***", "***Tenant ID***" service.

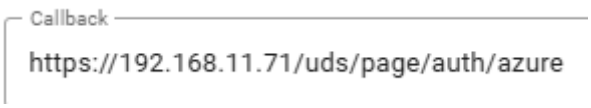
Client ID: To obtain this value it will be necessary to access the Azure account, section "***App registrations***", select the previously created application and copy the value of "***Application (client) ID***".

Client Secret: This value will be obtained from the previously registered application. Click on it (in the "***App registrations***" service) and access "***Certificates & secrets***". It will only be visible during the moment of creation.



- Advanced:

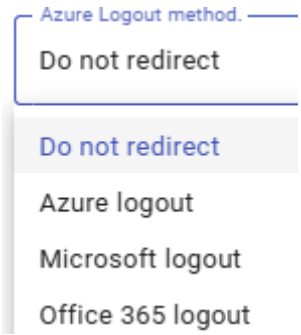
Callback: This field is automatically filled in by UDS once the authenticator is saved and its content must be copied to the "***App Registrations***" created in the Azure environment. Once saved, if we edit it we will see that it has been automatically filled in:



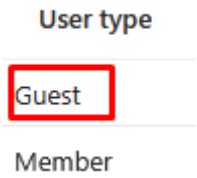
Proxy: Proxy server to be used in the connection to the authenticator

Enable School Data Sync Integration: If enabled, it will allow integration with the "***School Data Sync Integration***" tool which must be connected to the "***Azure AD"*** authenticator.

Azure logout method: Allows you to indicate the type of logout performed:



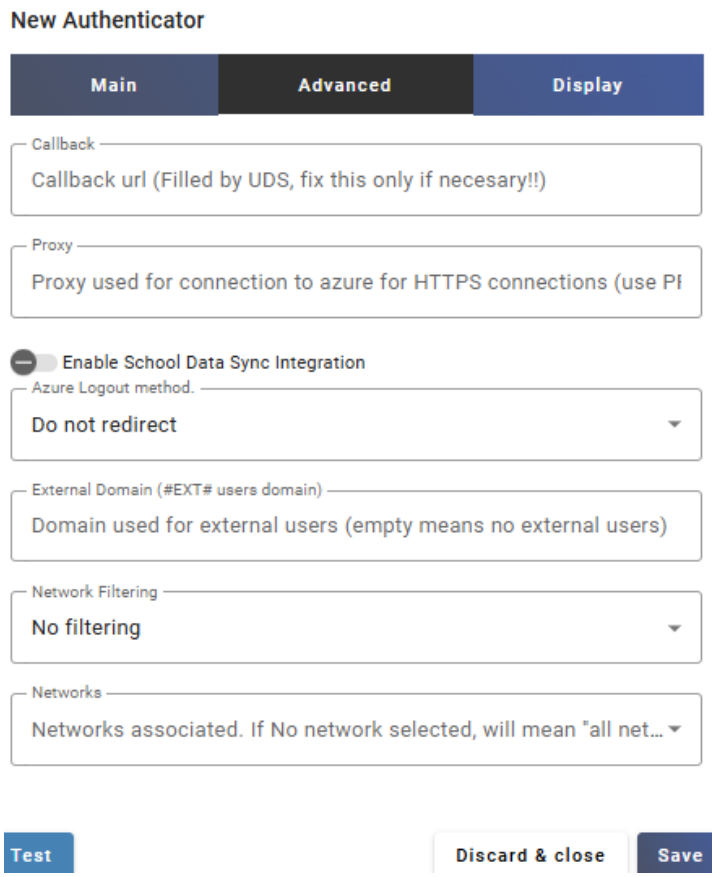
External Domain: Domain to be used for external users (users whose type is "Guest" within a directory.



Network Filtering: We will indicate the action to be carried out on the networks indicated in the "***Networks***" section. We may not perform filtering: "***No filtering***", allow only the indicated networks: "***Allow select networks***" or not allow only the indicated networks: "***Deny selected networks***".

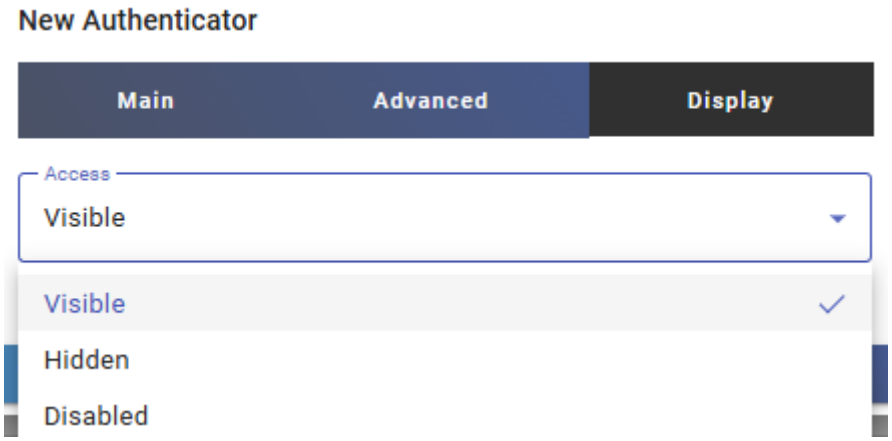
Networks: Network ranges, subnets or IP addresses indicated in the "***Networks***" section of the "Connectivity" section. Used in conjunction with the "***Network Filtering***" field to allow or disable authentication of users based on their network location.

NOTE: If none is selected, the authenticator will be visible on any network.

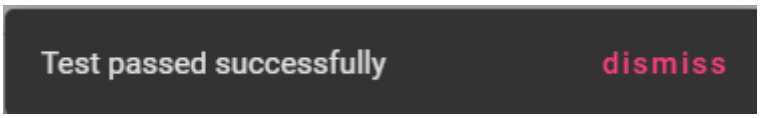


- Display:

Access: Allows you to indicate whether the authenticator will be visible and active to users: "***Visible***", if it will be active, but not visible: "***Hidden***" or if it will be completely disabled so that it cannot be viewed or used: "***Disabled'***".



NOTE: The integration of UDS Enterprise with an Entra ID authenticator requires a series of previous tasks and requirements to be carried out in the Azure environment itself, so it is advisable to consult the manual "VDI with UDS Enterprise and Microsoft Azure" located in the documentation section of the UDS Enterprise website. If these previous tasks are not performed, when we run the connection test we may have errors of the following type:

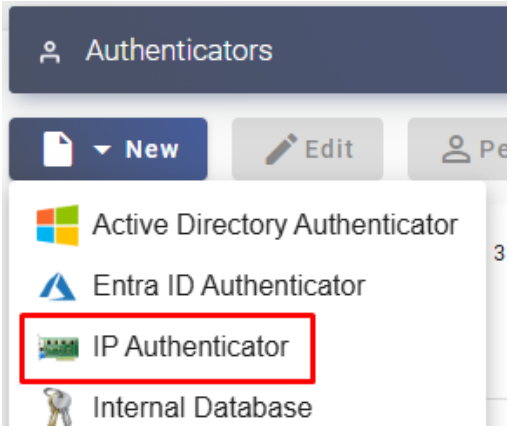


By using the "***Test***" button we will check that the connection with the authenticator is made correctly.

4.1.3 IP

This internal Authenticator allows you to give direct access to connection clients (*Single Sign-On*), through their IP address, to virtual desktops and applications.

IP addresses work like the users of other authenticators, allowing direct validation of connection clients in the UDS login portal. User groups in an ***IP Authenticator*** can range from specific network ranges to entire subnets or specific IPs.



En un "***IP Authenticator***" los parámetros mínimos a configurar son:

- Main:

Name: Nombre del autenticador.

Priority: Priority that this authenticator will have. If there are several authenticators, the lower their priority, the higher they will appear in the list of authenticators available in the UDS login portal. This field supports negative values.

Label: Enable direct validation on this authenticator. It allows the user to validate in the user login portal with said authenticator, even if the UDS environment has more authenticators. To do this, we will have to access the UDS login portal with the following format: ***UDSServer/uds/page/login/label*** (For example: https://UDSServer/uds/page/login/ip).

- Advanced:

Allowed only from this networks: Allows you to filter the networks from which the authenticator will be accessible. It is possible to indicate in this field IP addresses (separated by commas), complete subnets (e.x. 192.168.11.0/24) or ranges separated by a hyphen (e.x. 192.168.11.1-192.168.11.71).

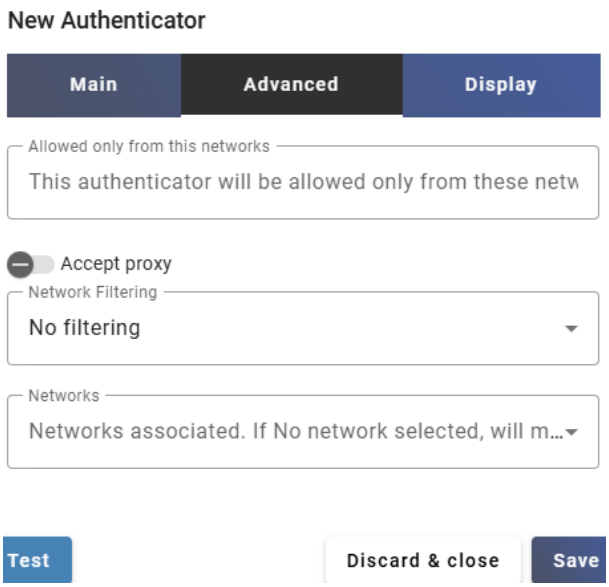
Accept proxy: This option is necessary to enable when there is a component prior to access the UDS server (login portal), it allows the detection of the IP address of the connection client to be correct (for example, when there is a UDS environment in HA and there are load balancers, if we do not activate it, the IP detected can be that of the balancer instead of that of the client).

NOTE: For the detection to be successful, the component prior to the UDS server (login portal) must have the X-Forwarded-For headers enabled.

Network Filtering: We will indicate the action to be carried out on the networks indicated in the "***Networks***" section. We may not perform filtering: "***No filtering***", allow only the indicated networks: "***Allow select networks***" or not allow only the indicated networks: "***Deny selected networks***".

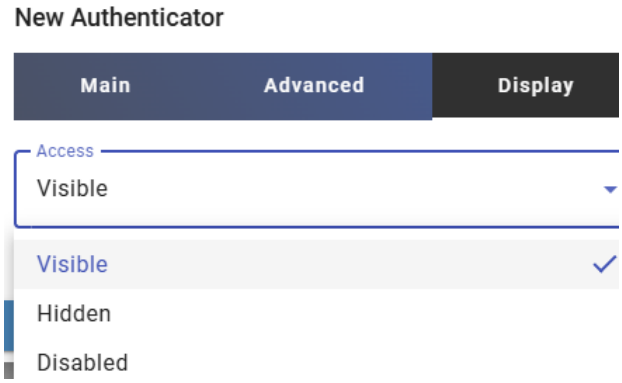
Networks: Network ranges, subnets or IP addresses indicated in the "***Networks***" section of the "Connectivity" section. It is used in conjunction with the "***Network Filtering***" field to allow or disable authentication of users based on their network location.

NOTE: If none is selected, the authenticator will be visible on any network.



- Display:

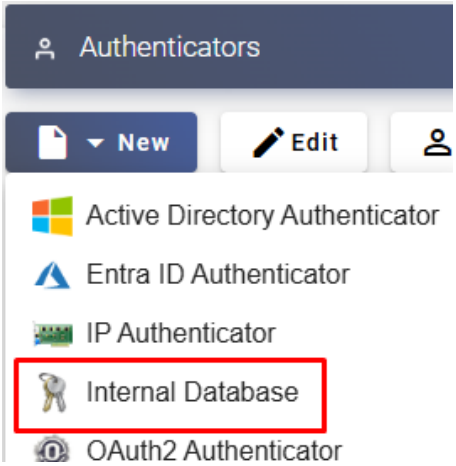
Access: Allows you to indicate whether the authenticator will be visible and active to users: "***Visible***", if it will be active, but not visible: "***Hidden***" or if it will be completely disabled so that it cannot be viewed or used: "***Disabled'***".



4.1.4 Internal Database

In environments where no external authenticator is available, it is possible to use the "***Internal Database***" authenticator. This authenticator allows the creation of users and groups manually for subsequent access to the different desktop services and virtual applications provided by the UDS platform.

All user and group data is stored in the database to which the UDS server connects.



In an "***Internal Database***" the minimum parameters to be configured are:

- Main:

Name: Name of the authenticator.

Priority: The priority that this authenticator will have. If there are several authenticators, the lower their priority, the higher they will appear in the list of authenticators available in the UDS login portal. This field supports negative values.

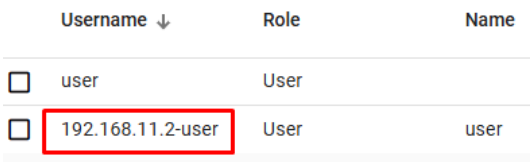
Label: Enables direct validation on this authenticator. It allows the user to validate themselves in the login portal with said authenticator, even if the UDS environment has more authenticators. To do this, we will have to access the UDS login portal with the following format: ***UDSServer/uds/page/login/label*** (For example: https://UDSServer/uds/page/login/int).

- Advanced:

Different user for each host: Allows authentications to be performed using a single (generic) connection user. To do this, add a root to the name of an existing user at the time of its validation in the UDS login portal. This root is the IP address of the connecting client

The newly created user has the following structurea:

IP_Client_conetion-user_generic



Reverse DNS: It behaves exactly like the "***Different user for each host***" option, but the root added to the user is the DNS name of the connection client. A correct DNS resolution is required. Otherwise, the IP address will be used.

The newly created user has the following structure:

Name_client_conextion-user_generic

Accept proxy: This option is necessary to enable when there is a component prior to UDS server access, such as a load balancer.

By default, UDS automatically detects the IP address of the connection client. In environments where load balancers or other similar elements are configured, this detection is not performed correctly since the detected IP address corresponds to those balancers. By enabling this option we will achieve correct IP detection.

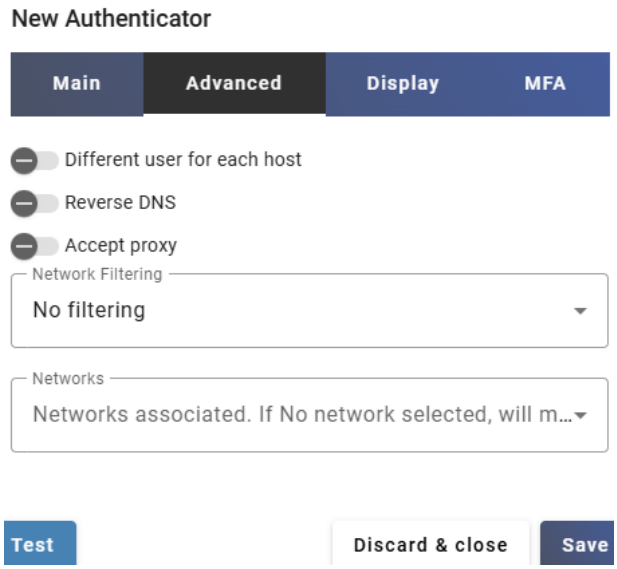
NOTE: For detection to be successful, the component prior to the UDS server (login portal) must have X-Forwarded-For headers enabled.

In environments where the "***Different user for each host***" option is used and load balancers exist, this option must be enabled.

Network Filtering: We will indicate the action to be carried out on the networks indicated in the "***Networks***" section. We may not perform filtering: "***No filtering***", allow only the indicated networks: ***"Allow select networks***" or not allow only the indicated networks: "***Deny selected networks***".

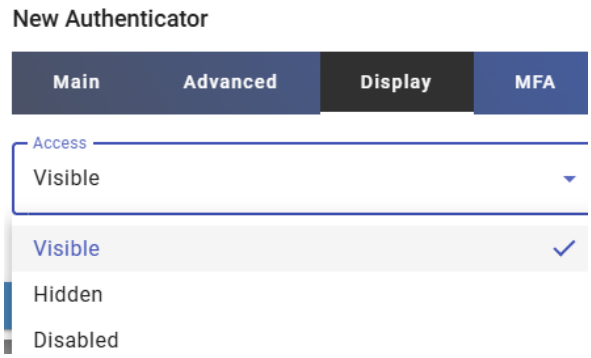
Networks: Network ranges, subnets or IP addresses indicated in the "***Networks***" section of the "Connectivity" section. Used in conjunction with the "***Network Filtering***" field to allow or disable authentication of users based on their network location.

NOTE: If none is selected, the authenticator will be visible on any network.



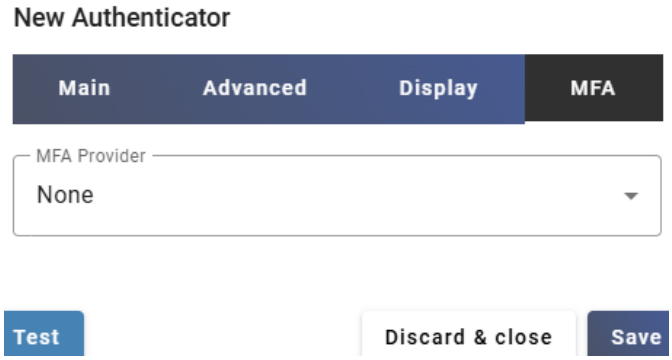
- Display:

Access: Allows you to indicate whether the authenticator will be visible and active to users: "Visible", if it will be active, but not visible: "Hidden" or if it will be completely disabled so that it cannot be viewed or used: "Disabled"



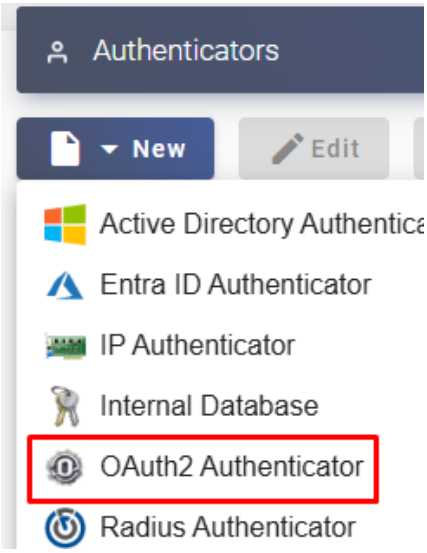
- MFA:

MFA Provider: Two-factor authentication (MFA) system applied to the authenticator. You must be previously registered in the "***Authentication***" section "***Multi Factor***".



4.1.5 OAuth2

This External Authenticator allows you to give access to the UDS environment to groups of users belonging to an identity provider with OAuth2 support.



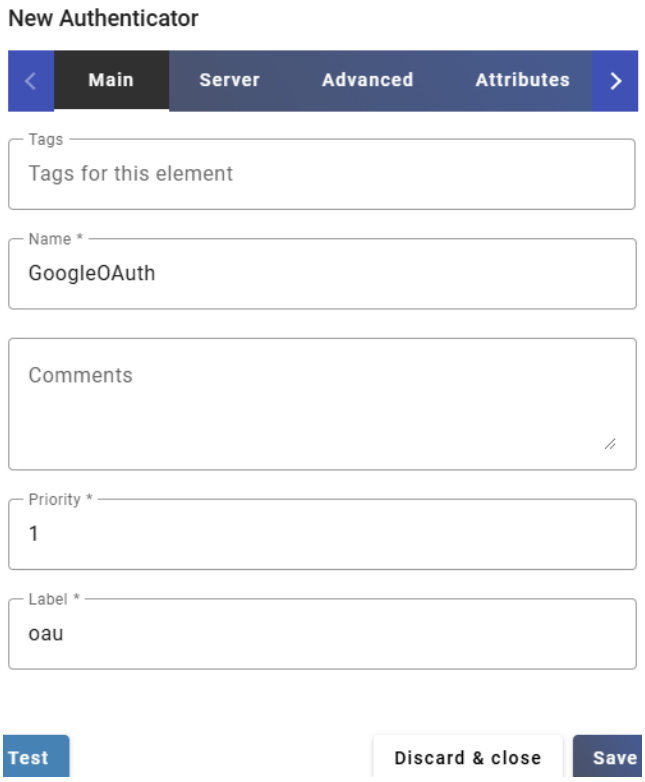
In an "***OAuth2 Authenticator***" we configure the following sections:

- Main:

Name: Name of the authenticator (cannot contain spaces).

Priority: The priority that this authenticator will have. If there are several authenticators, the lower their priority, the higher they will appear in the list of authenticators available in the UDS login portal. This field supports negative values.

Label: Enables direct validation on this authenticator. It allows the user login portal to be validated with this authenticator, even if the UDS environment has more authenticators. To do this, we will have to access the UDS login portal with the following format: ***UDSServer/uds/page/login/label*** (For example: https://UDSServer/uds/page/login/oau).



- Server:

Client ID: ID of the client created (in the authentication system used) for this authenticator.

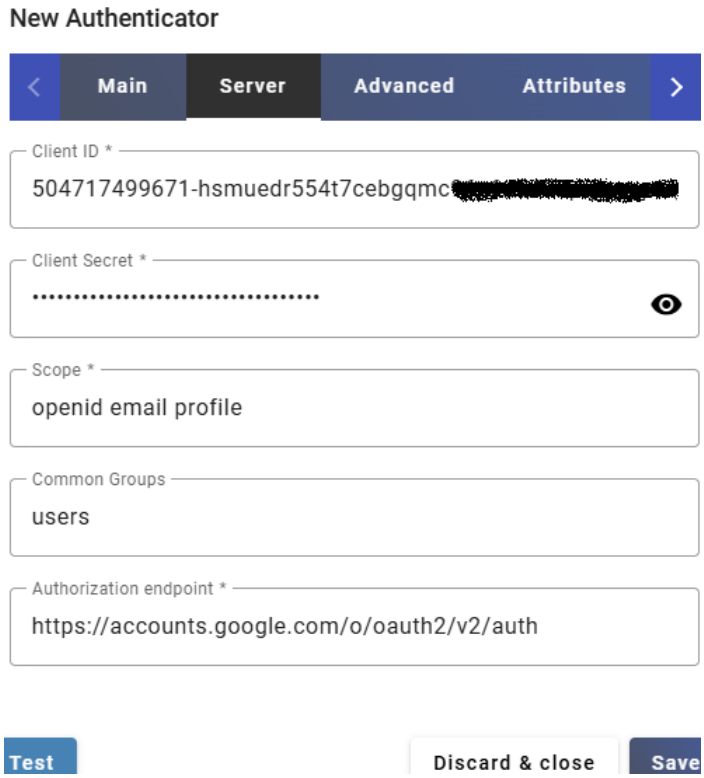
Client Secret: Password generated for the generated client ID

Scope: Scopes supported by the authentication system, for example:

```
"scopes_supported": [
 "openid",
 "email",
 "profile"
],
```

Common Groups: Users who are authenticated will be added to this common group. You can put several separated by commas.

Authorization endpoint: URL of the "authorization endpoint" of the authentication system used Users who are authenticated will be added to this common group. They can be put several separated by commasUsers who are authenticated will be added to this common group. They can be put several separated by commas



- Advanced:

Endpoint redirection: This field will be automatically populated when you save the item and will use the data from the current URL (if configured while connected to UDS via IP, the server name will not be used) to compose the redirection to the "endpoint".

Response type: Type of response used by the authentication server.

Endpoint token: URL "Token endpoint" used (supplied by the authentication system).

User information endpoint: URL "user information endpoint" used (supplied by the authentication system).

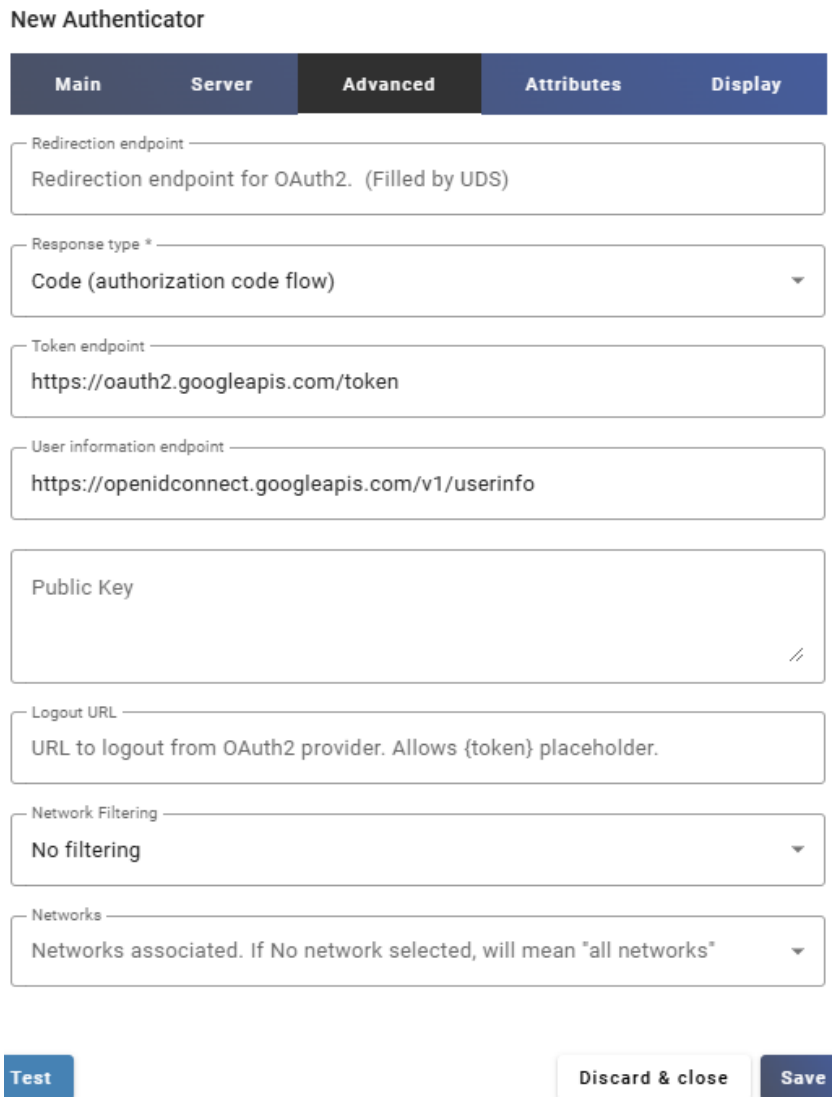
Public Key

Logout URL: URL for the users' logout.

Network Filtering: We will indicate the action to be carried out on the networks indicated in the "***Networks***" section. We may not perform filtering: "***No filtering***", allow only the indicated networks: ***"Allow select networks***" or not allow only the indicated networks: "***Deny selected networks***".

Networks: Network ranges, subnets or IP addresses indicated in the "***Networks***" section of the "Connectivity" section. Used in conjunction with the "***Network Filtering***" field to allow or disable authentication of users based on their network location.

NOTE: If none is selected, the authenticator will be visible on any network.

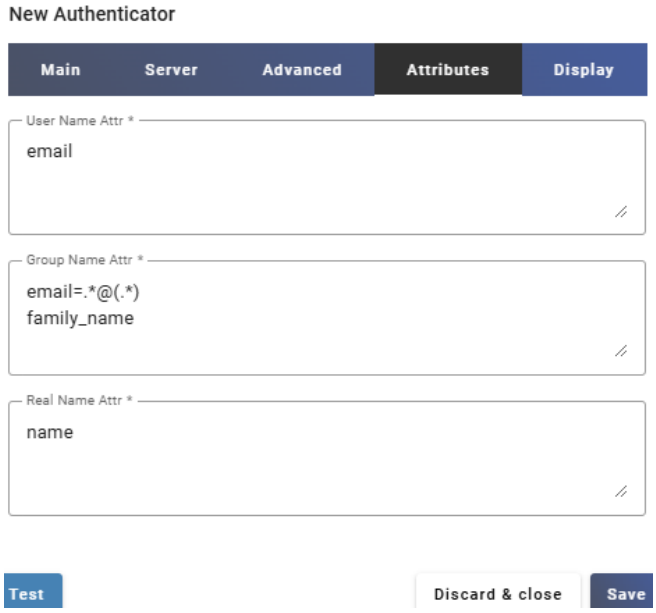


- Attributes:

User name attrs: Existing attribute in the authentication system, which will define the user's name to log in to the UDS access portal.

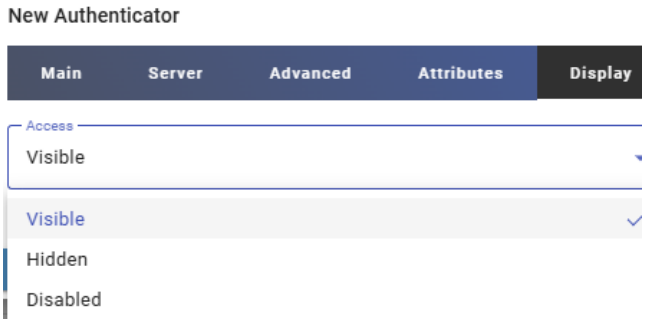
Group name attrs: An existing attribute in the authentication system, which will define a user's membership in a group. Different attributes can be specified to define different group attributes (one on each line) and also allows the use of regular expressions to extract or compose specific values.

Real name attrs: An existing attribute in the authentication system, which will define the user's name information. It does not affect the UDS login portal.



- Display:

Access: Allows you to indicate whether the authenticator will be visible and active to users: "***Visible***", if it will be active, but not visible: "***Hidden***" or if it will be completely disabled so that it cannot be viewed or used: "***Disabled'***".



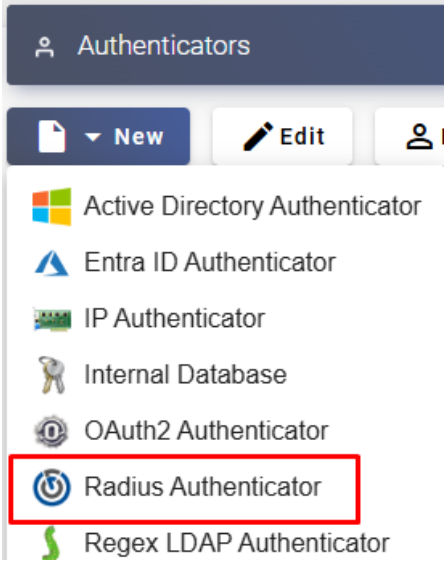
NOTE: The screenshots show an example of integration with the Google Workspace directory, almost all of the information needed for the integration is found at the following URL:

https://accounts.google.com/.well-known/openid-configuration

To integrate another authentication system, the corresponding information will be necessary.

4.1.6 Radius Authenticator

This external authenticator allows access to the UDS environment to groups of users belonging to a RADIUS system.



In a "***Radius Authenticator***" the minimum parameters to be configured are:

- Main:

Name: Name of the authenticator.

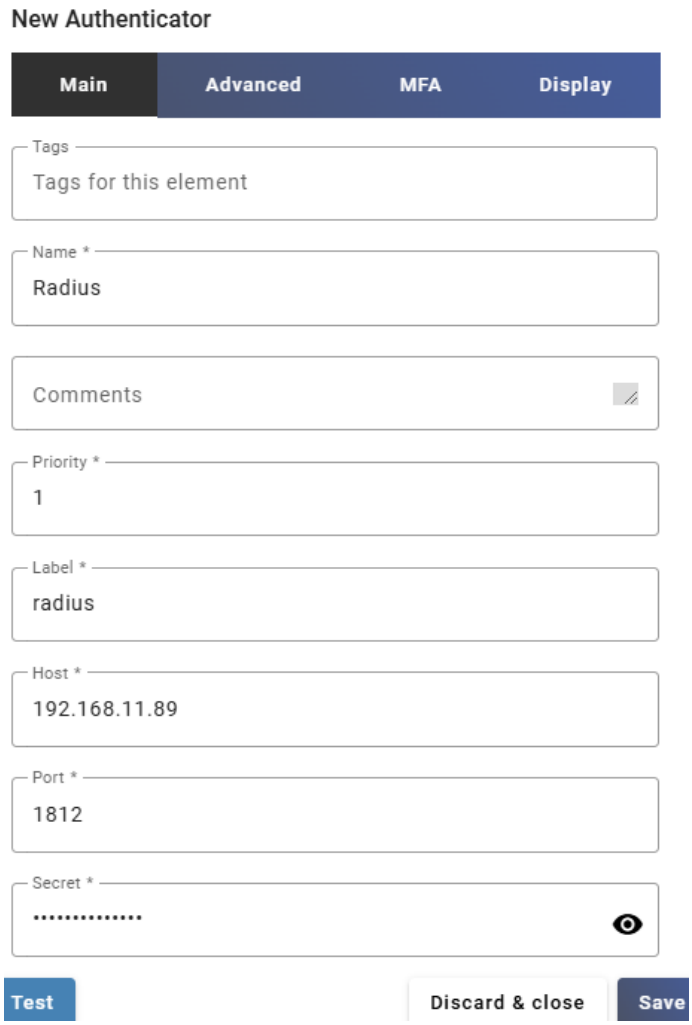
Priority: The priority that this authenticator will have. If there are several authenticators, the lower their priority, the higher they will appear in the list of authenticators available in the UDS login portal. This field supports negative values.

Label: Enables direct validation on this authenticator. It allows the user to validate themselves in the login portal with said authenticator, even if the UDS environment has more authenticators. To do this, we will have to access the UDS login portal with the following format: ***UDSServer/uds/page/login/label*** (For example: https://UDSServer/uds/page/login/radius).

Host: IP address or RADIUS server name.

Port: Communication port with the RADIUS server.

Secret: Validation string against the RADIUS server (defined on the RADIUS server itself).



- Advanced:

NAS Identified: Identifies UDS within the RADIUS server, allowing you to filter if necessary.

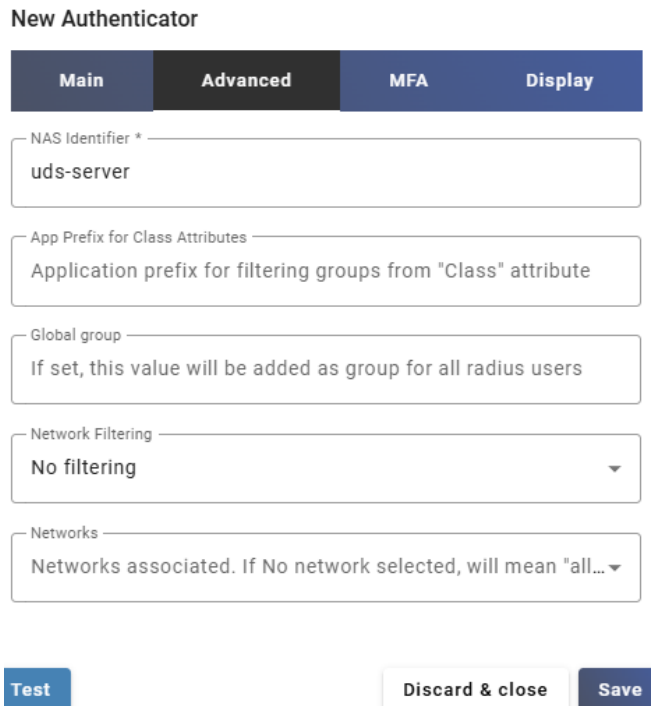
App Prefix for Class Attributes: Allows you to filter which groups you get from the "class" attribute of the RADIUS server.

Global group: Allows you to make all users belong, by force, to a group. This allows a RADIUS server (which is still a "simple" authenticator) that does not contain groups, to assign ALL users to a group (even if it also contains groups).

Network Filtering: We will indicate the action to be carried out on the networks indicated in the "***Networks***" section. We may not perform filtering: "***No filtering***", allow only the indicated networks: ***"Allow select networks***" or not allow only the indicated networks: "***Deny selected networks***".

Networks: Network ranges, subnets or IP addresses indicated in the "***Networks***" section of the "Connectivity" section. Used in conjunction with the "***Network Filtering***" field to allow or disable authentication of users based on their network location.

NOTE: If none is selected, the authenticator will be visible on any network.

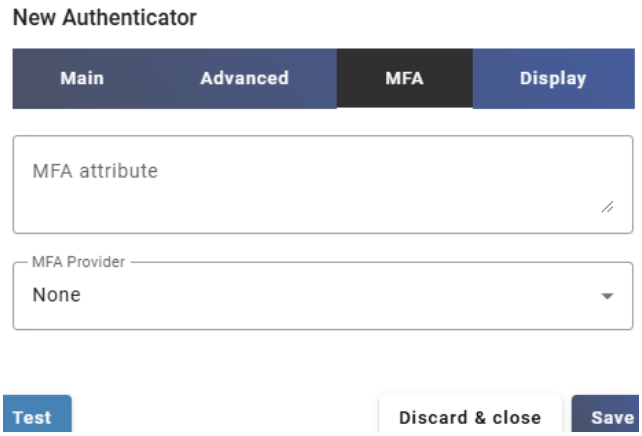


NOTE: By default, UDS will extract from the Radius "Class" attribute the elements that have the form "group=..."
If we define in "App Prefix for Class Attributes", for example, "lab", UDS will look only for "class" attributes that have the form "labgroup=..."

- MFA:

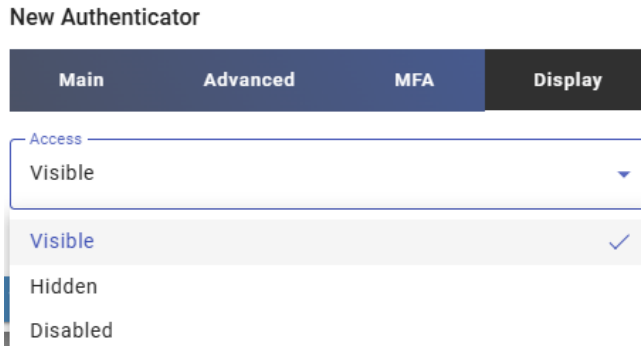
MFA attribute: We will use this field to indicate which user attribute will be used by the MFA system to notify you of your token, usually the attribute where the email or phone is indicated, for MFA systems for sending SMS and email. If you use the TOTP system, you will not need to fill in this field, but for all other supported systems, you will need to.

MFA Provider: Two-factor authentication (MFA) system applied to the authenticator. You must be previously registered in the "***Authentication***" section "***Multi Factor'''***".



- Display:

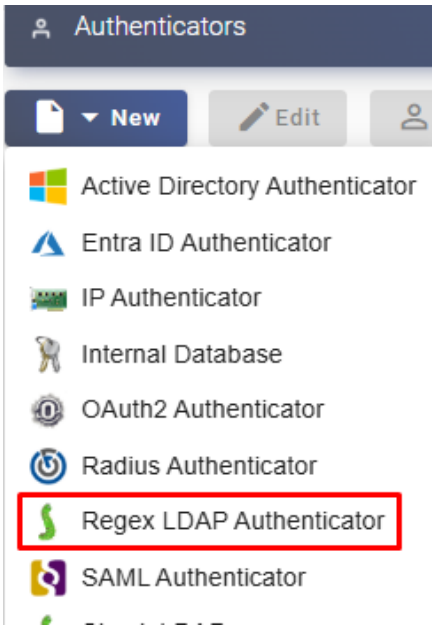
Access: Allows you to indicate whether the authenticator will be visible and active to users: "***Visible***", if it will be active, but not visible: "***Hidden***" or if it will be completely disabled so that it cannot be viewed or used: "***Disabled'***".



Using the "***Test***" button we will check that the connection with the authenticator is successful.

4.1.7 Regex LDAP

This external authenticator allows access to the UDS environment to groups of users belonging to almost any authenticator based on the LDAP access protocol.



In a ***"Regex LDAP Authenticator***" the minimum parameters to be configured are:

- Main:

Name: Name of the authenticator.

Priority: The priority that this authenticator will have. If there are several authenticators, the lower their priority, the higher they will appear in the list of authenticators available in the UDS login portal. This field supports negative values.

Label: Enables direct validation on this authenticator. It allows the user to validate themselves in the login portal with said authenticator, even if the UDS environment has more authenticators. To do this, we will have to access the UDS login portal with the following format: ***UDSServer/uds/page/login/label*** (For example: https://UDSServer/uds/page/login/ldap).

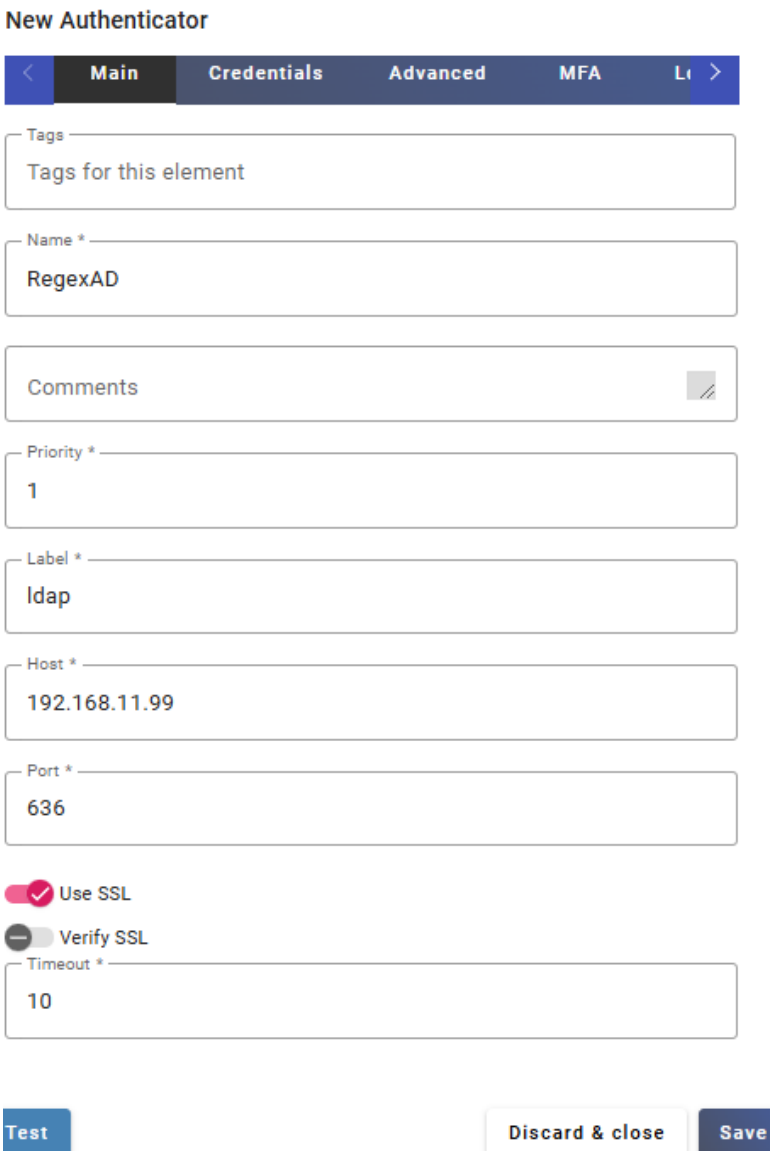
Host: IP address or LDAP server name.

Port: Communication port with the LDAP server (if SSL connectivity is enabled, the default port is 636)

Use SSL: If enabled, SSL connection will be used against the authenticator.

Verify SSL: If enabled, certificate verification will be performed.

Timeout: Timeout time of the connection to the authenticator.

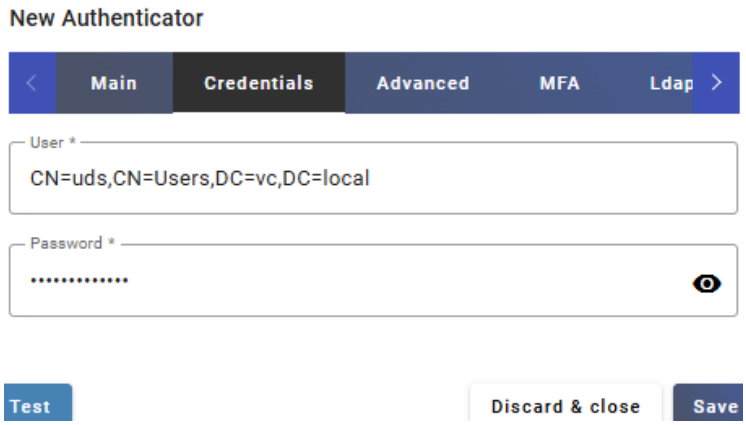


- Credentials:

User: User with read permissions on the authenticator in format: ***CN=...,uid=...,ou=...,dc=...,dc=...***

Password: User Password.

Safe Operating Procedure: Passwords must be of sufficient length and include upper and lower case, numbers and special characters.



- Advanced:

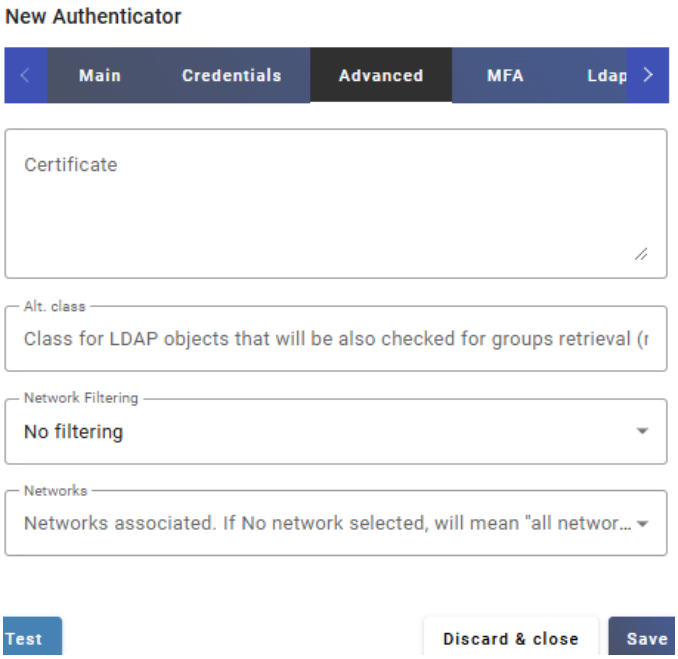
Certificate: Certificate used to perform the validation of the "***Verify SSL" field***)

Class Alt: We can specify more object classes to search for users and user groups.

Network Filtering: We will indicate the action to be carried out on the networks indicated in the "***Networks***" section. We may not perform filtering: "***No filtering***", allow only the indicated networks: ***"Allow select networks***" or not allow only the indicated networks: "***Deny selected networks***".

Networks: Network ranges, subnets or IP addresses indicated in the "***Networks***" section of the "Connectivity" section. Used in conjunction with the "***Network Filtering***" field to allow or disable authentication of users based on their network location.

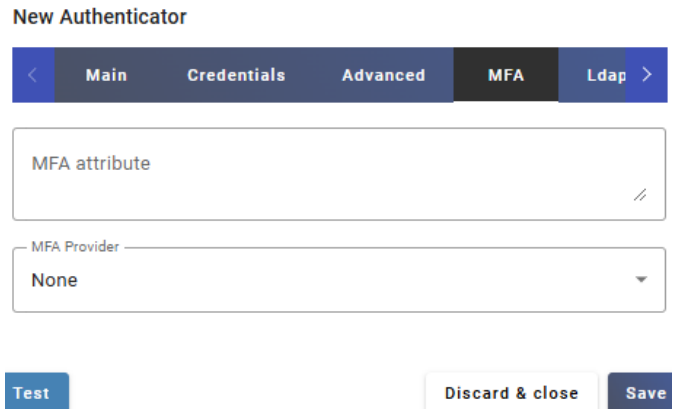
NOTE: If none is selected, the authenticator will be visible on any network.



- MFA:

MFA attribute: We will use this field to indicate which user attribute will be used by the MFA system to notify you of your token, usually the attribute where the email or phone number is indicated, for the MFA systems for sending SMS and email. If you use the TOTP system, you will not need to fill in this field, but for all other supported systems, you will need to.

MFA Provider: Two-factor authentication (MFA) system applied to the authenticator. You must be previously registered in the "***Authentication***" section "***Multi Factor'''***".



- Ldap info:

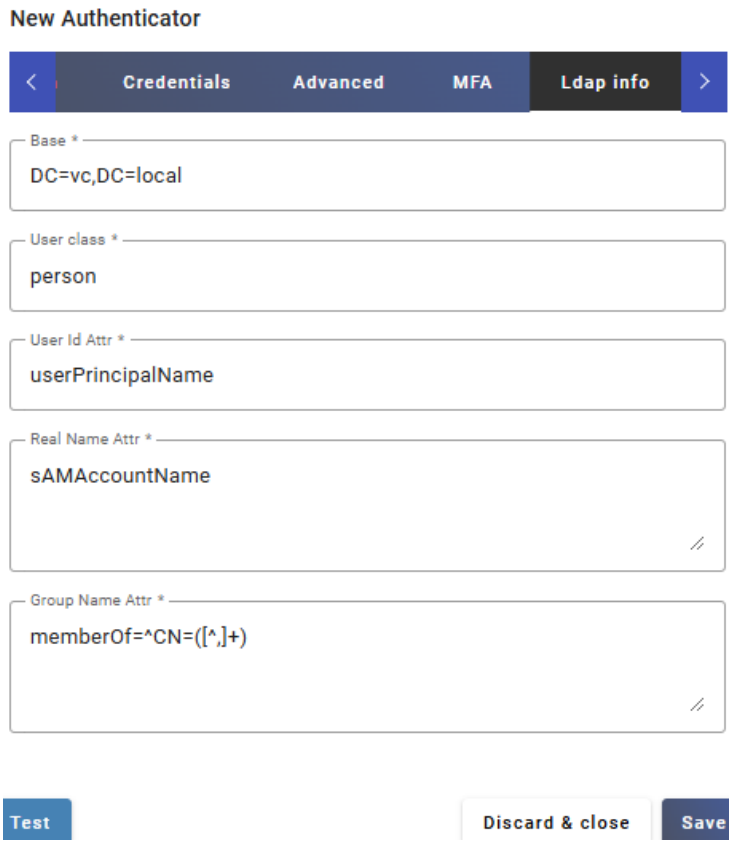
Base: Search directory where the system will locate groups and users to be validated.

User class: Common class that all users must have.

User Id Attr: LDAP attribute that will define the user's name to log in to the UDS access portal.

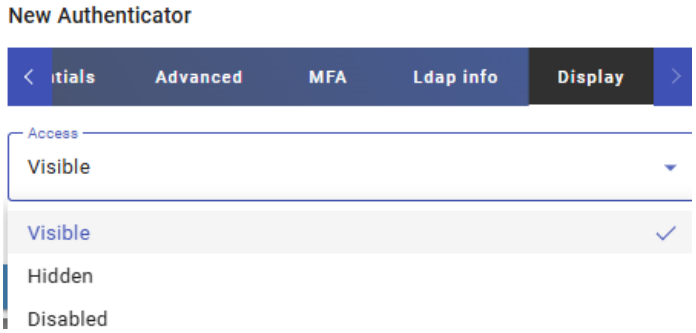
Real Name Attr: An LDAP attribute that will define the user's name information. It does not affect the UDS login portal. In addition, it allows the use of regular expressions to extract or compose specific values.

Group Name Attr: An LDAP attribute that will define a user's membership in a group. Different attributes can be specified to define different group attributes (one on each line). In addition, it allows the use of regular expressions to extract or compose specific values.



- Display:

Access: Allows you to indicate whether the authenticator will be visible and active to users: "***Visible***", if it will be active, but not visible: "***Hidden***" or if it will be completely disabled so that it cannot be viewed or used: "***Disabled'***".

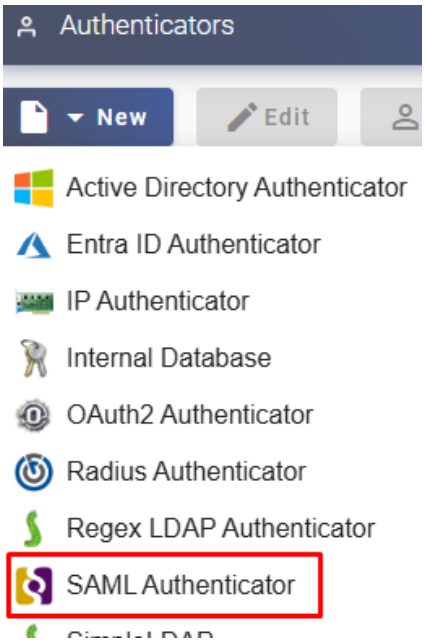


By using the "***Test***" button we will check that the connection with the authenticator is made correctly.

4.1.8 SAML

This External Authenticator allows access to the UDS environment to groups of users belonging to an identity provider (IDP) with SAML 2.0 protocol support.

SAML is used for the exchange of authentication and authorization data between security domains, i.e., between an identity provider (an assertion producer) and a service provider (an assertion consumer).



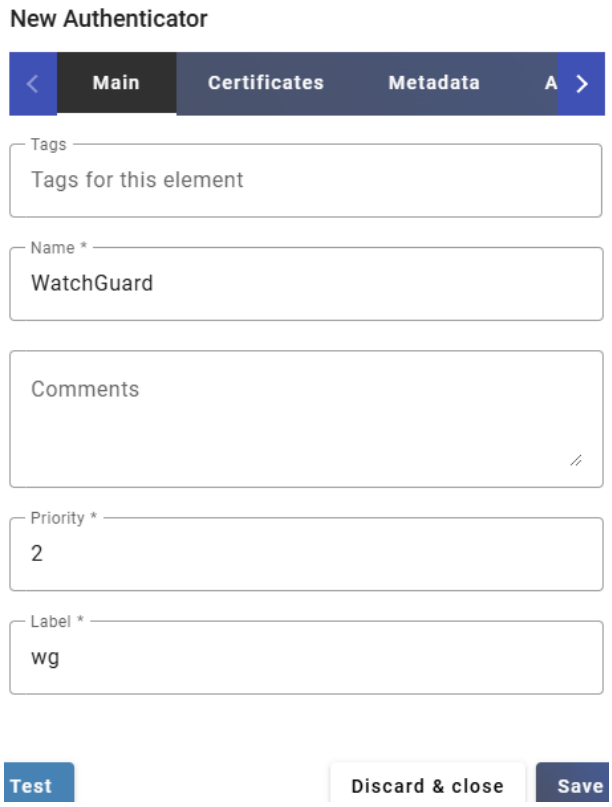
In a "***SAML Authenticator***" we configure the following sections:

- Main:

Name: Name of the authenticator (cannot contain spaces).

Priority: The priority that this authenticator will have. If there are several authenticators, the lower their priority, the higher they will appear in the list of authenticators available in the UDS login portal. This field supports negative values.

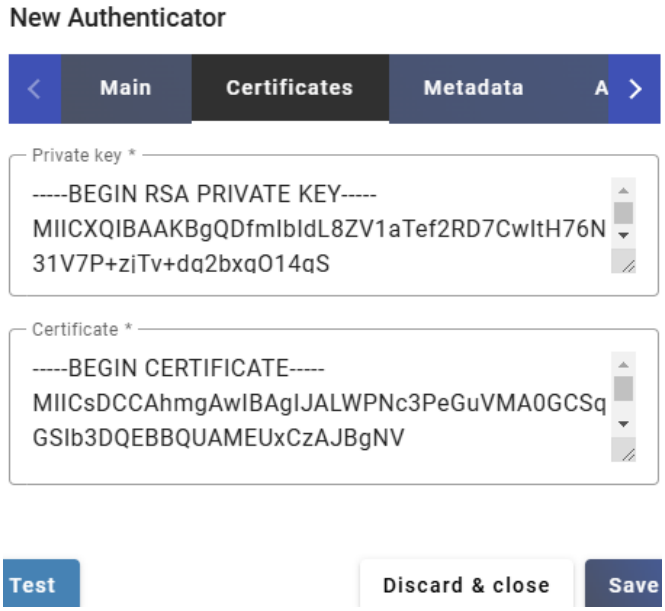
Label: Enables direct validation on this authenticator. It allows the user login portal to be validated with this authenticator, even if the UDS environment has more authenticators. To do this, we will have to access the UDS login portal with the following format: ***UDSServer/uds/page/login/label*** (For example: https://UDSServer/uds/page/login/saml).



- Certificates:

Private key: The certificate's private RSA key that is used for signatures and encryptions.

Certificate: Server certificate that is used for the connection.



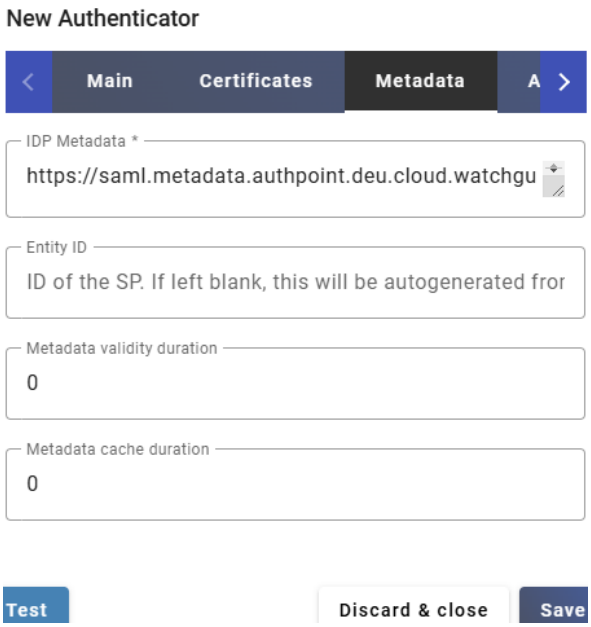
- Metadata:

IDP Metadata: In this field we will indicate the IDP metadata, being able to choose between the URL that contains it or directly the content in XML format.

Entity ID: This field must be initially left empty. When you save the authenticator, the URL with the SP metadata (UDS) will be automatically generated. The URL generated in this field or its metadata must be used to register the authenticator as an SP on the federated server (IDP).

Metadata validity duration : Duration, in days, of metadata

Metadata cache duration: Duration, in days, that the cached metadata will be.

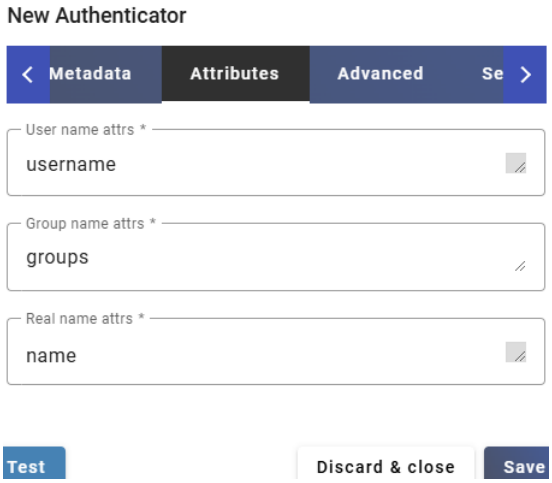


- Attributes:

User name attrs: Existing attribute sent by the IDP, which will define the user's name to log in to the UDS access portal.

Group name attrs: An existing attribute sent by the IDP, which will define a user's membership in a group. Different attributes can be specified to define different group attributes (one on each line) and also allows the use of regular expressions to extract or compose specific values.

Real name attrs: An existing attribute sent by the IDP, which will define the user's name information. It does not affect the UDS login portal.



- Advanced:

Global logout: If it is active, when we log out of UDS, we will try to perform a global "logout" (in any browser tab).

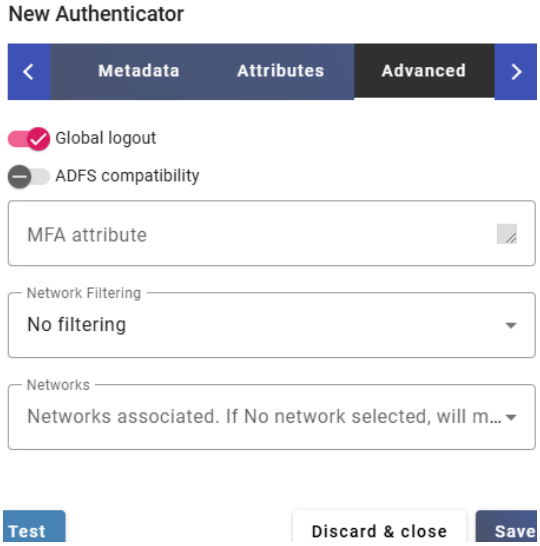
ADFS compatibility: Enables support for Active Directory Federation Services

MFA attribute: We will use this field to indicate which user attribute will be used by the MFA system to notify you of your token, usually the attribute where the email or phone number is indicated, for the MFA systems for sending SMS and email. If you use the TOTP system, you will not need to fill in this field, but for all other supported systems, you will need to.

Network Filtering: We will indicate the action to be carried out on the networks indicated in the "***Networks***" section. We may not perform filtering: "***No filtering***", allow only the indicated networks: ***"Allow select networks***" or not allow only the indicated networks: "***Deny selected networks***".

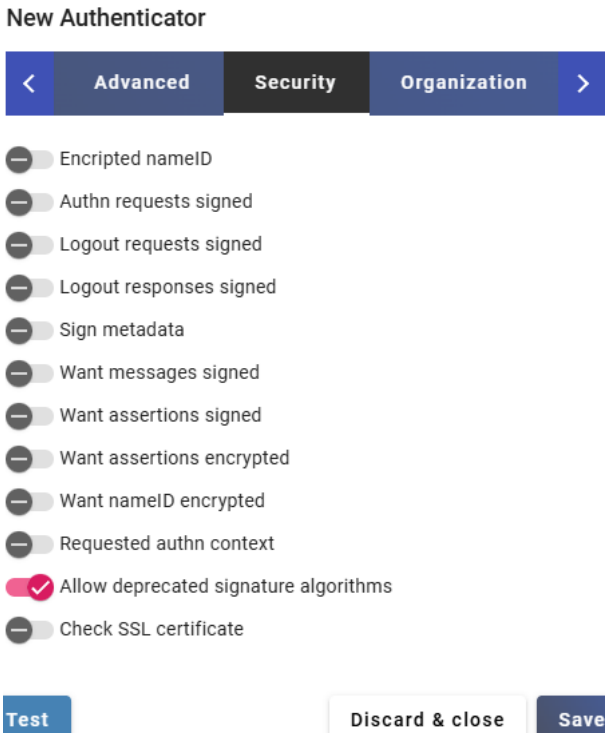
Networks: Network ranges, subnets or IP addresses indicated in the "***Networks***" section of the "Connectivity" section. Used in conjunction with the "***Network Filtering***" field to allow or disable authentication of users based on their network location.

NOTE: If none is selected, the authenticator will be visible on any network.



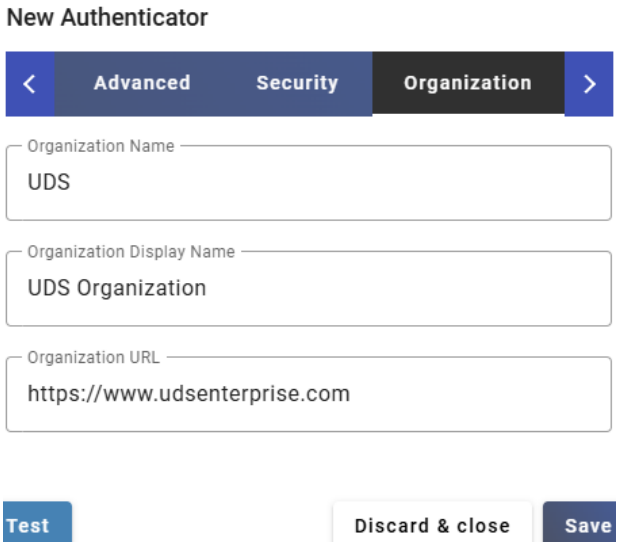
- Security:

In this section we will indicate everything related to the security of the connection. We will enable everything that is necessary from the IDP



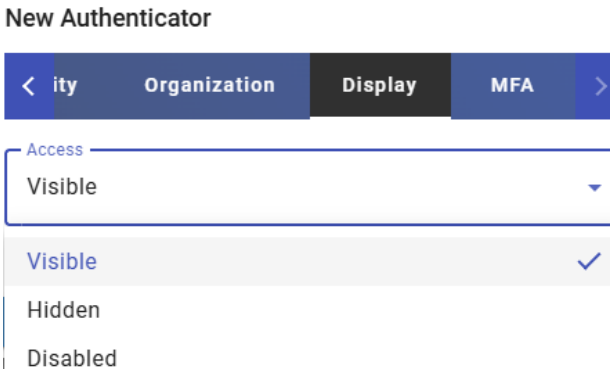
- Organization:

In this section we will indicate all the data referring to our organization, this data will be indicated in the metadata generated by the system for the connection.



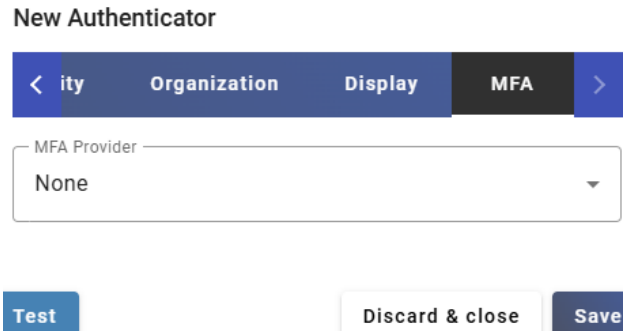
- Display:

Access: Allows you to indicate whether the authenticator will be visible and active to users: "***Visible***", if it will be active, but not visible: "***Hidden***" or if it will be completely disabled so that it cannot be viewed or used: "***Disabled'''***".



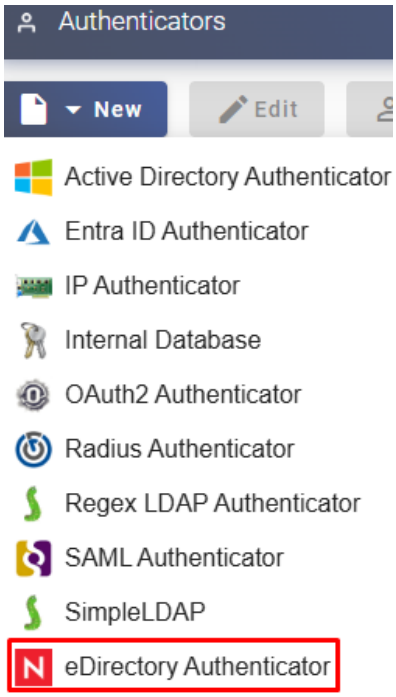
- MFA:

MFA Provider: Two-factor authentication (MFA) system applied to the authenticator. You must be previously registered in the "***Authentication***" section "***Multi Factor'''***".



4.1.9 eDirectory

This External Authenticator allows groups of users belonging to a Novell eDirectory-type authenticator to be given access to the UDS environment



In an "eDirectory Authenticator" the minimum parameters to be configured are:

- Main:

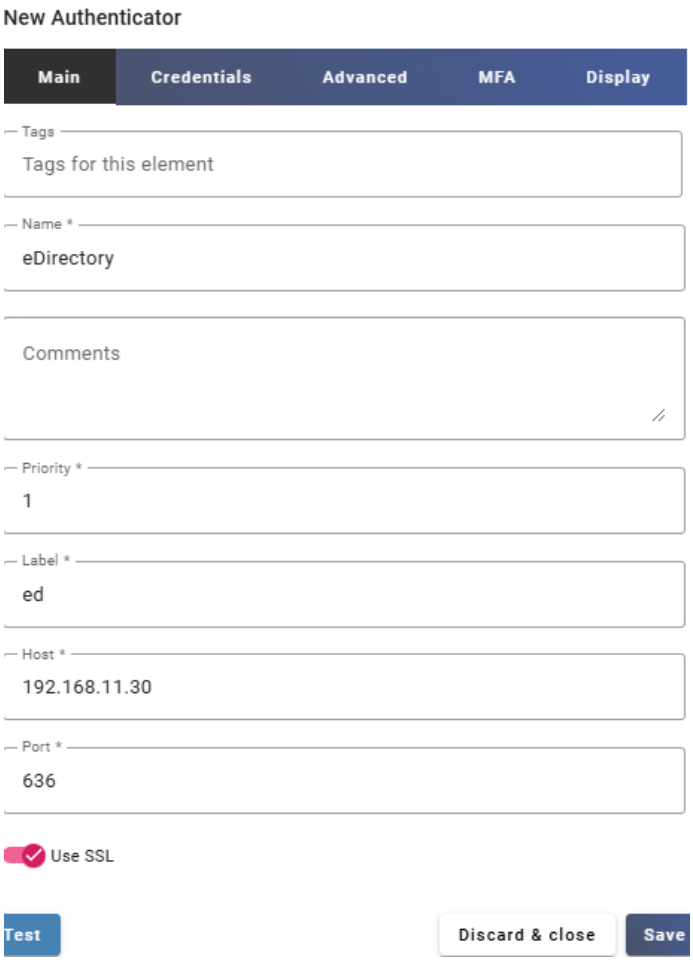
Name: Name of the authenticator.

Priority: The priority that this authenticator will have. If there are several authenticators, the lower their priority, the higher they will appear in the list of authenticators available in the UDS login portal. This field supports negative values.

Label: Enables direct validation on this authenticator. It allows the user to validate themselves in the login portal with said authenticator, even if the UDS environment has more authenticators. To do this, we will have to access the UDS login portal with the following format: ***UDSServer/uds/page/login/label*** (For example: https://UDSServer/uds/page/login/ed).

Host: IP address or eDirectory server name.

Use SSL: If enabled, SSL connection will be used against the authenticator.

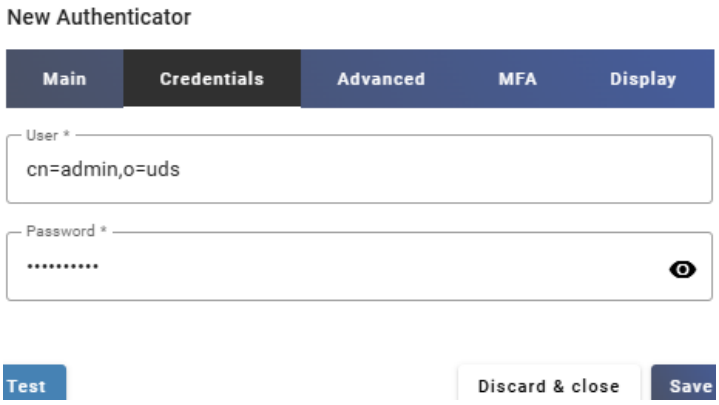


- Credentials:

User: User with read permissions on the authenticator in format: ***CN=user, O=domain.***

Password: User password.

Safe Operating Procedure: Passwords must be of sufficient length and include upper and lower case, numbers and special characters.



- Advanced:

Timeout: Timeout time of the connection to the authenticator.

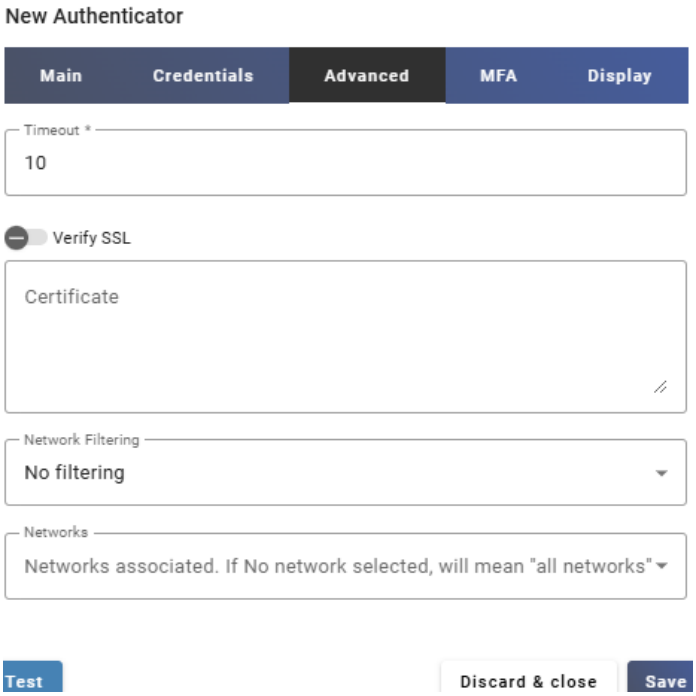
Verify SSL: If active, the authentication server certificate will be verified. The certificate used for verification must be included in the "Certificate" field.

Certificate: Certificate from the authentication server to check if the "***Verify SSL***" field is active.

Network Filtering: We will indicate the action to be carried out on the networks indicated in the "***Networks***" section. We may not perform filtering: "***No filtering***", allow only the indicated networks: ***"Allow select networks***" or not allow only the indicated networks: "***Deny selected networks***".

Networks: Network ranges, subnets or IP addresses indicated in the "***Networks***" section of the "Connectivity" section. It is used in conjunction with the "***Network Filtering***" field to allow or disable authentication of users based on their network location.

NOTE: If none is selected, the authenticator will be visible on any network.



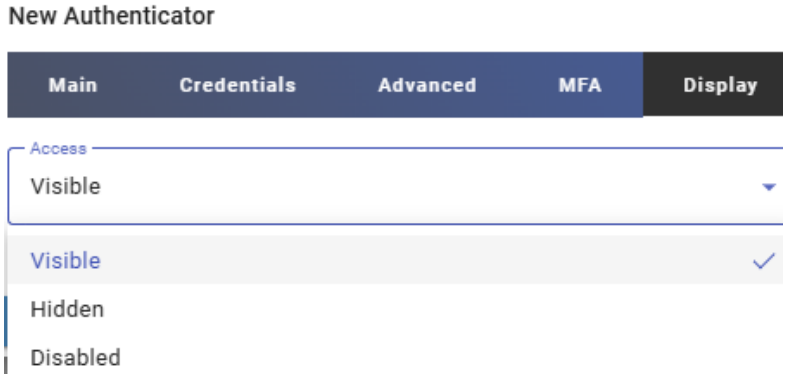
- MFA:

MFA attribute: We will use this field to indicate which user attribute will be used by the MFA system to notify you of your token, usually the attribute where the email or phone number is indicated, for the MFA systems for sending SMS and email. If you use the TOTP system, you will not need to fill in this field, but for all other supported systems, you will need to.

MFA Provider: Two-factor authentication (MFA) system applied to the authenticator. You must be previously registered in the "***Authentication***" section "***Multi Factor'''***".

- Display:

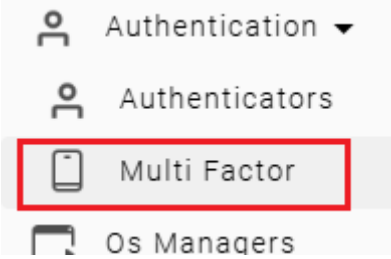
Access: Allows you to indicate whether the authenticator will be visible and active to users: "***Visible***", if it will be active, but not visible: "***Hidden***" or if it will be completely disabled so that it cannot be viewed or used: "***Disabled'***".



By using the "Test" button we will check that the connection with the authenticator is made correctly.

4.2 Multifactor

Multi-factor authentication (MFA) adds a layer of protection to the user sign-in process.



UDS natively supports several multi-factor authentication systems:

- Email Multi Factor
- Radius OTP Challenge
- SMS via HTTP
- TOTP Based MFA

It also supports other MFAs integrated into the authenticator itself (SAML, OAuth, Entra ID, etc...), so that when the authenticator is incorporated into UDS Enterprise, MFA is automatically incorporated.

Any method created from multi-factor will have to be assigned to an authenticator later in order to be used and applied in user validation.

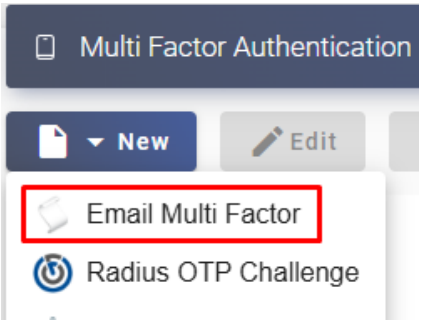
A "Multifactor" configuration is not a necessary component for the creation of a "Service Pool"

Safe Employment Procedure:

It is recommended to use any system, at least one, of multi-factor authentication.

4.2.1 Email Multi Factor

Through this method, the user will first authenticate with their username and password and then be redirected to a second authentication process, in which they will receive an email with the necessary code to finally be able to authenticate on their UDS Enterprise platform.



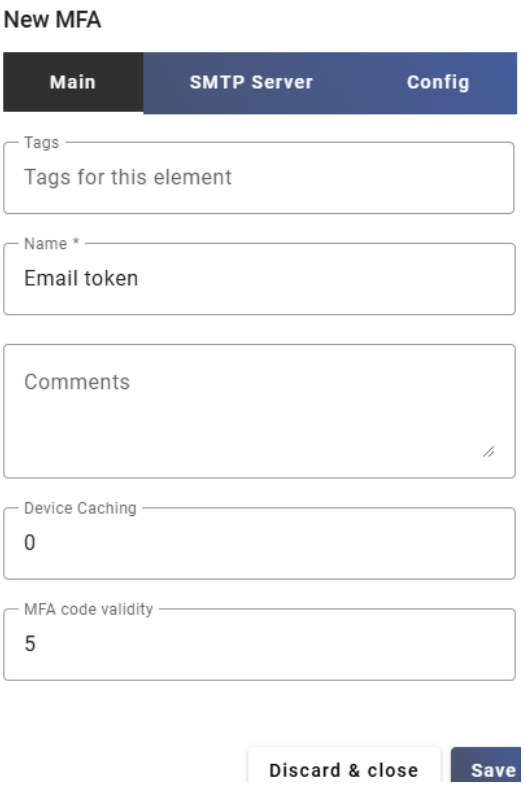
In a "***Multi-Factor Email***" the minimum parameters to be configured are:

- Main:

Name: Name that will identify the element.

Device Caching: Time in hours that the device will be cached so that the Code is not requested again.

MFA code validity: Time in minutes to allow the use of MFA code.



- SMTP Server:

SMTP Host: Name or IP address of the SMTP server that will send the emails with the codes.

Security: Security protocol to be used.

Username: You have permission to access the SMTP server.

Password: User password.

- Config:

Subject: Subject of the email that will be sent to users.

From Email: The email address that will be used as the sender.

Enable HTML: If enabled, it will use the text hosted in the "Mail HTML" section, otherwise, the text in the "Mail text" field will be used.

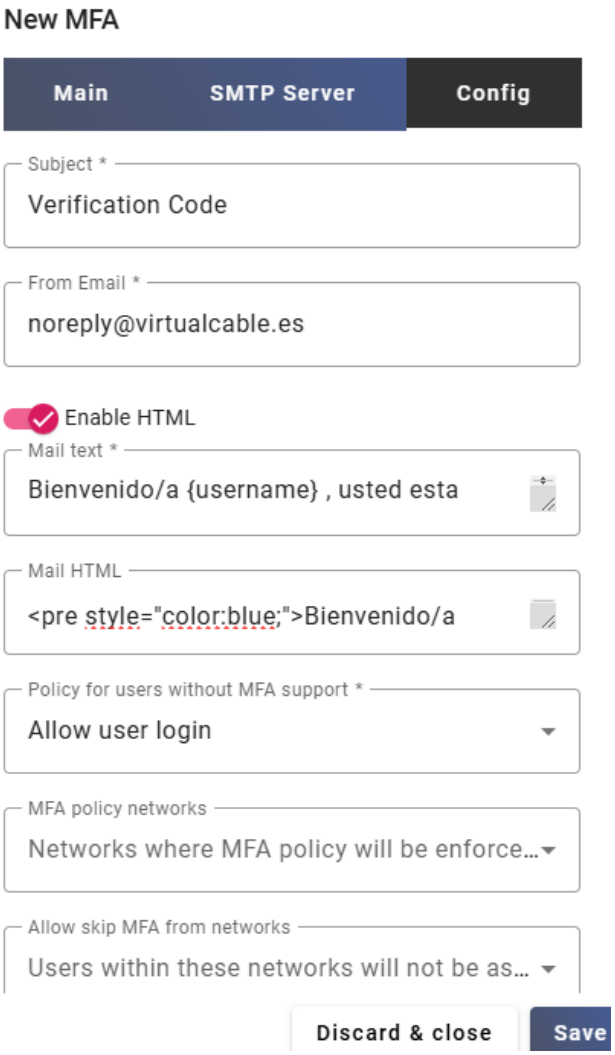
Mail Text: Text that will be displayed in the sent email.

HTML Mail: Text to be displayed in email sent with HTML support

Policy for users without MFA Support: Policy that will be used with those users who do not have an MFA attribute in their authenticated (for example, if the MFA attribute indicated in an authenticator is "email" and a user does not have it filled, the system will allow the user to authenticate or not, depending on this field).

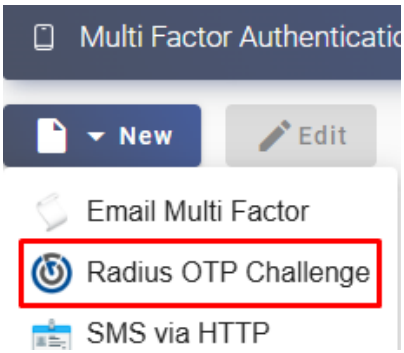
MFA policy networks: We will select which networks will apply to the policy in the "Policy for users without MFA Support" section**".**

Allow skip MFA from networks: We can indicate the networks on which the MFA system will not be applied (for example, when a user connects from the LAN, we can indicate that MFA validation is not required, but if they connect from the WAN, we will enable it).



4.2.2 Radius OTP Challenge

Through this method, the user will first authenticate with their username and password and then be redirected to a second authentication process where they must enter the corresponding code provided by their radius server in order to finally authenticate on their UDS Enterprise platform.



In a "***Radius OTP Challenge***" the minimum parameters to be configured are:

- Main:

Name: Name that will be indicated to the element.

Host: Name of the host or ip of the radius server.

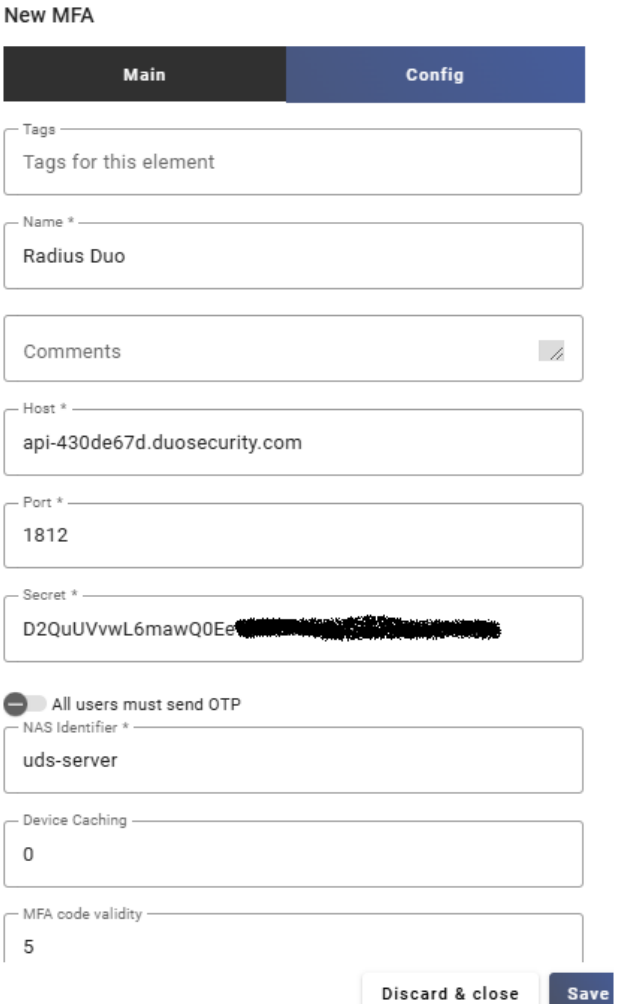
Port: Connection port to the radius server.

Secret: "Secret" of the Radius customer.

NAS Identifier: NAS identifier for radius Server.

Device Caching: Time in hours that the device will be cached so that the code is not requested again.

MFA code validity: Time in minutes to allow the use of MFA code.



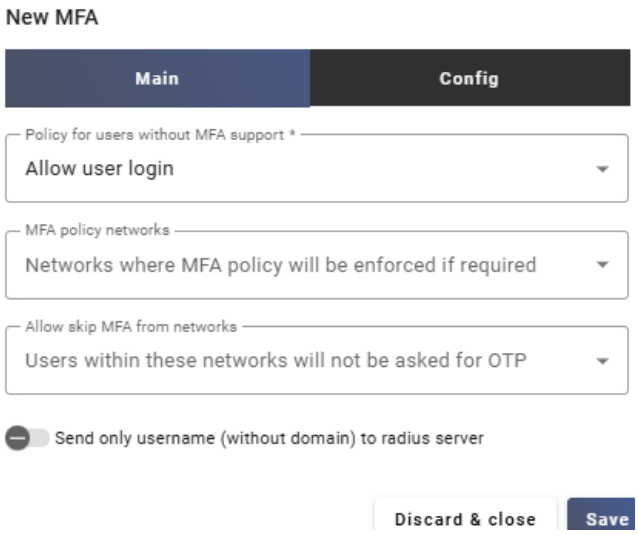
- Config:

Policy for users without MFA Support: Policy that will be used with those users who do not have an MFA attribute in their authenticated (for example, if the MFA attribute indicated in an authenticator is "email" and a user does not have it filled, the system will allow the user to authenticate or not, depending on this field).

MFA policy networks: We will select which networks will apply to the policy in the "Policy for users without MFA Support**" section.**

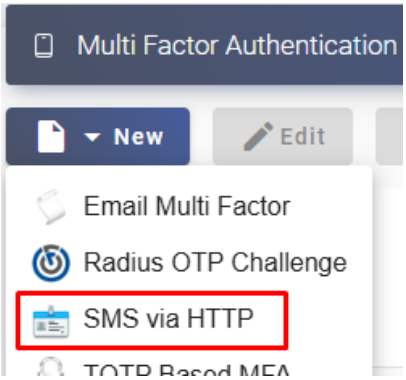
Allow skip MFA from networks: We can indicate the networks on which the MFA system will not be applied (for example, when a user connects from the LAN, we can indicate that MFA validation is not necessary, but if they connect from the WAN, we will enable it).

Send only username (without domain) to radius server: If enabled, the user without the domain will be sent to the radius server.


New MFA
Main
Config
Policy for users without MFA support *
Allow user login
MFA policy networks
Networks where MFA policy will be enforced if required
Allow skip MFA from networks
Users within these networks will not be asked for OTP
Send only username (without domain) to radius server
Discard & close
Save

4.2.3 SMS via HTTP

Through this method, the user will first authenticate with their username and password and then be redirected to a second authentication process where they will receive an SMS with the necessary code to finally be able to authenticate on their UDS Enterprise platform.



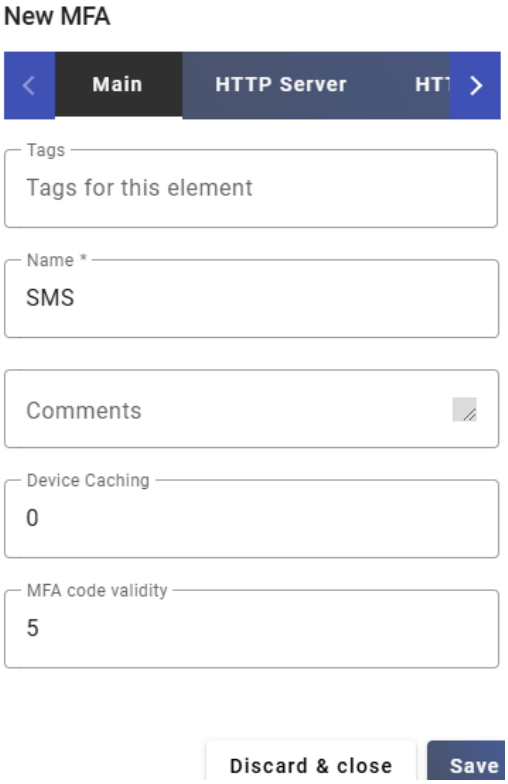
En un "***SMS Via HTTP*** " Los parámetros mínimos por configurar son

- Main:

Name: Name that the element will be indicated.

Device Caching: Time in hours that the device will be cached so that the Code is not requested again.

MFA code validity: Time in minutes to allow the use of MFA code.



- HTTP Server:

URL pattern for SMS Sending: URL of the server that will send SMS messages.

Ignore certificate errors: If active, it will ignore errors in the certificate verification.

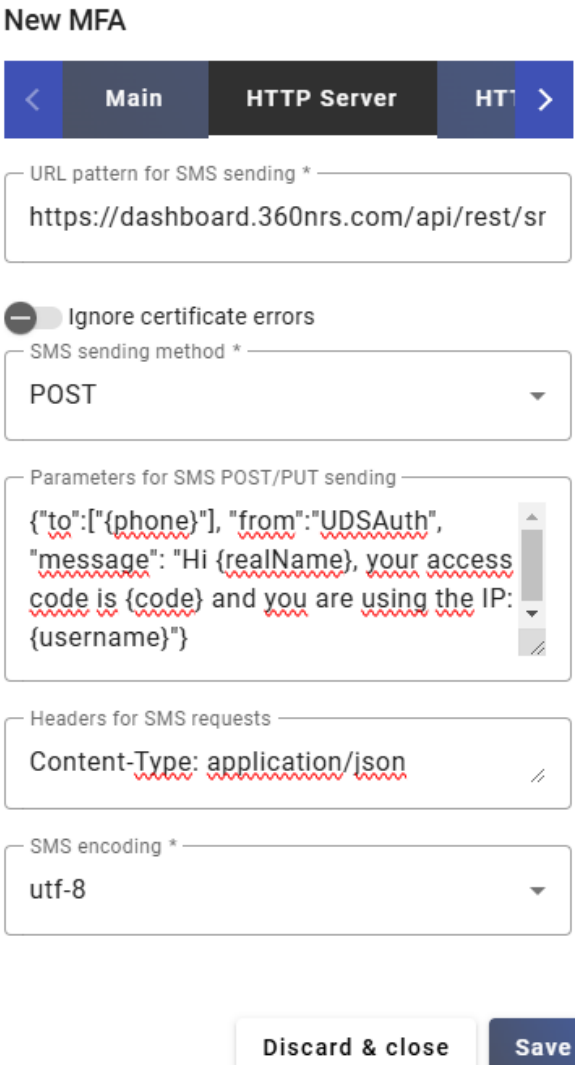
SMS Sending method: method that will be used to send SMS.

Parameters for SMS POST/PUT sending: Text and variables used. It can contain the following variables:

* {code} - The code to be sent
* {phone/+phone} - the phone number
* {username} - the username
* {justUsername} - the username without @.

Headers for SMS request: Header for SMS Request

SMS Enccoding: Encoding that will be used to send the SMS.

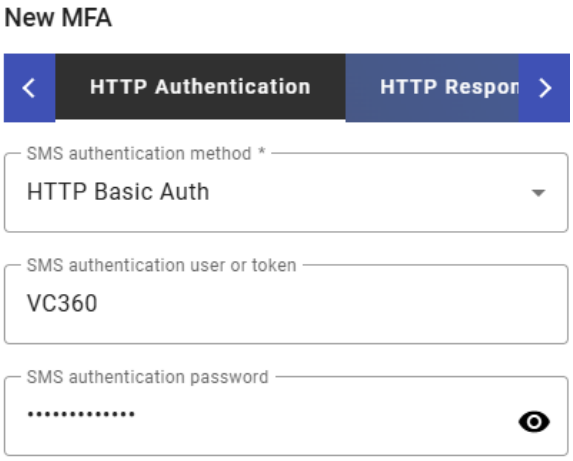


- HTTP Authentication:

SMS authentication method: SMS authentication method

SMS authentication user or Token: User or token for SMS authentication

SMS authentication password: Password for SMS authentication



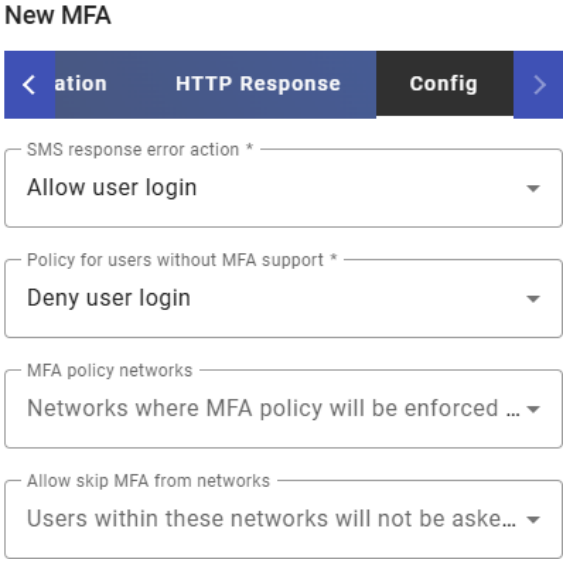
- Config:

SMS response error action

Policy for users without MFA Support: Policy that will be used with those users who do not have an MFA attribute in their authenticated (for example, if the MFA attribute indicated in an authenticator is "email" and a user does not have it filled, the system will allow the user to authenticate or not, depending on this field).

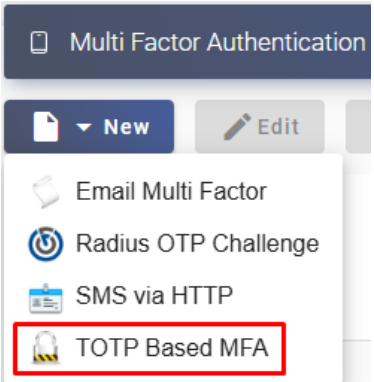
MFA policy networks: We will select which networks will apply to the policy in the "Policy for users without MFA Support**" section.**

Allow skip MFA from networks: We can indicate the networks on which the MFA system will not be applied (for example, when a user connects from the LAN, we can indicate that MFA validation is not necessary, but if they connect from the WAN, we will enable it).



4.2.4 TOTP Based MFA

Through this method, the user will first authenticate with their username and password and then be redirected to a second authentication process where they must enter the TOTP code generated from time to time in an application such as **Google authenticator**, **Microsoft authenticator**, etc... to finally be able to authenticate to your UDS Enterprise platform.



In a ***"TOTP Based MFA"*** the minimum parameters to be configured are:

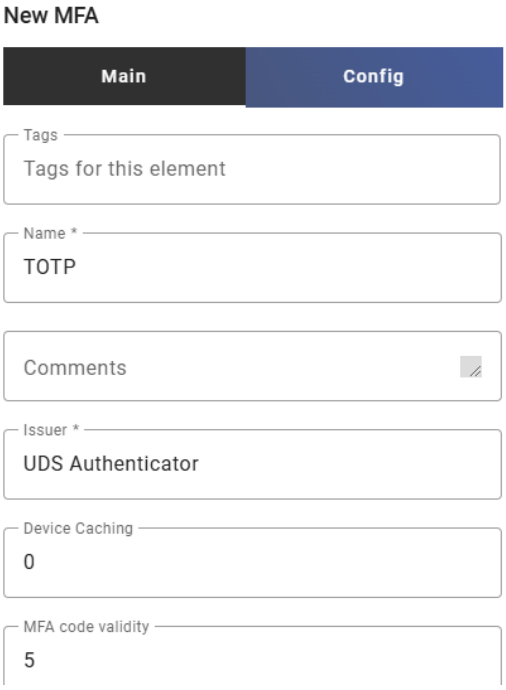
- Main:

Name: Name that the element will be indicated.

Issuer: OTP issuer, once created it cannot be changed.

Device Caching: Time in hours that the device will be cached so that the Code is not requested again.

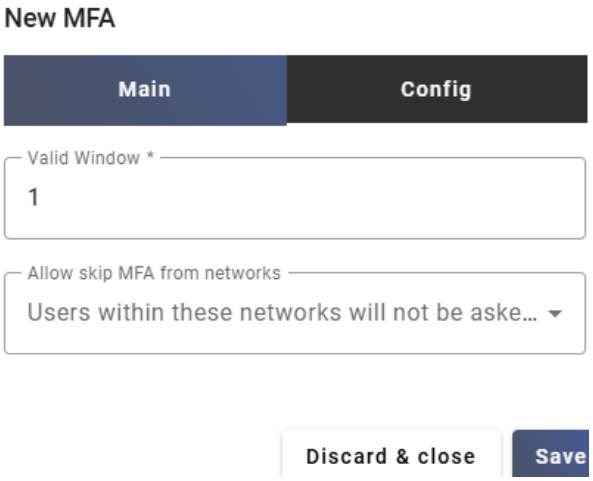
MFA code validity: Time in minutes to allow the use of MFA code.



- Config:

Valid Window: Number of codes that will be valid before and after the current one.

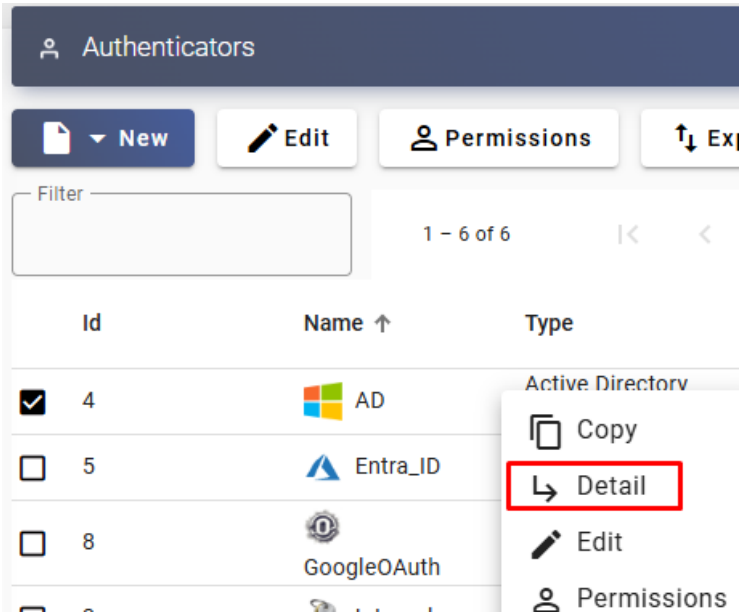
Allow skip MFA from networks: We can indicate the networks on which the MFA system will not be applied (for example, when a user connects from the LAN, we can indicate that MFA validation is not necessary, but if they connect from the WAN, we will enable it).



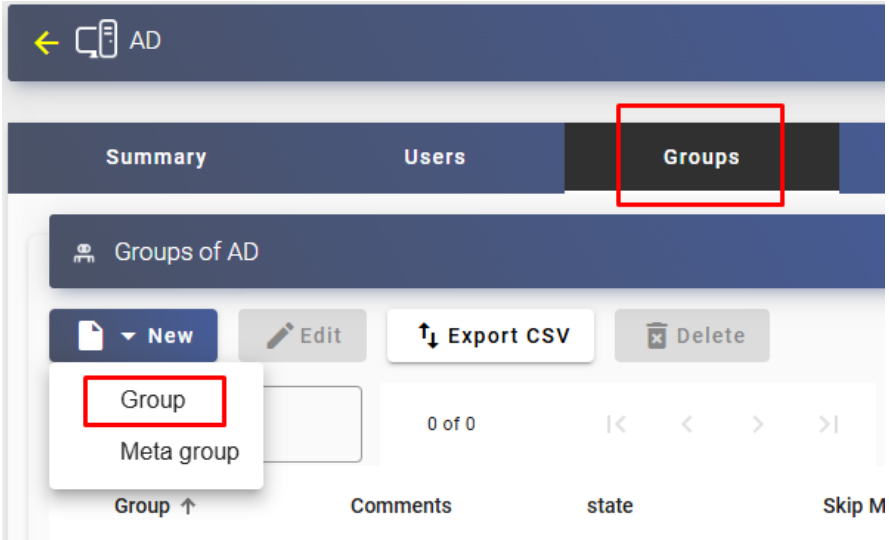
4.3 Users, groups, and metagroups

Once an authenticator has been configured in UDS, it is necessary to register in UDS Enterprise the existing user groups in the authenticator that contain the users who are going to be given access through the login portal to desktop services and virtual applications. It is also possible to create metagroups, which will be used to combine several groups of users.

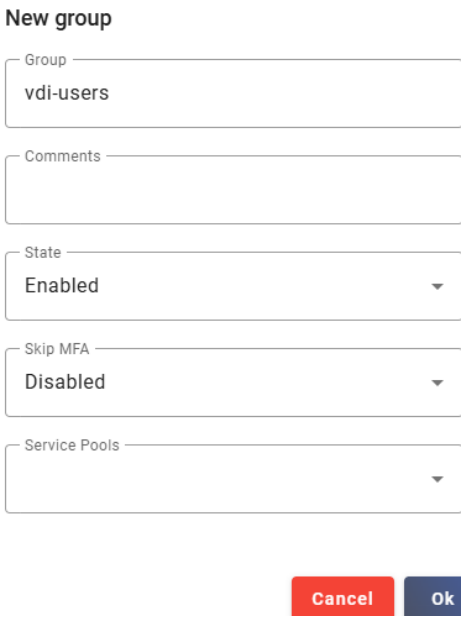
To add groups or metagroups to an authenticator, select it and double-click on it or in the provider menu select "***Detail***":



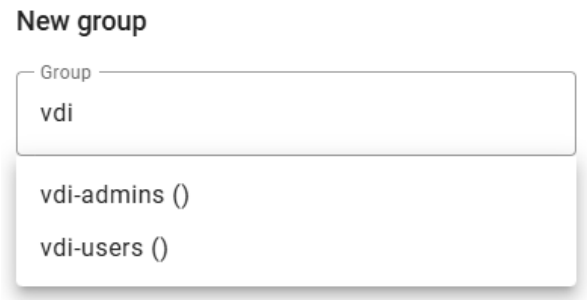
Once inside the authenticator, select the "***Groups***" tab and in "***New***" select "***Group***":



We will indicate the name of the group using the "***Group***" field, its status (enabled or disabled), if we are going to apply the MFA method and we can also assign it directly to one or more "***Service Pools'***".



On some authenticators such as "***Active Directory***", an automatic lookup will be performed:



In others such as "***Regex LDAP***", "***SAML***", "***eDirectory***"... it will be necessary to indicate it manually.

NOTE: In internal authenticators such as "Internal Database" and "IP Authenticator", it will be necessary to create groups manually, since they do not connect to any external authentication system.

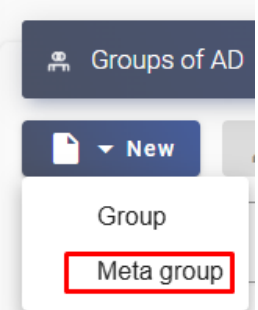
All users who belong to a group (of a specific authenticator), will be allowed access to the UDS login portal.

Safe Employment Procedure:

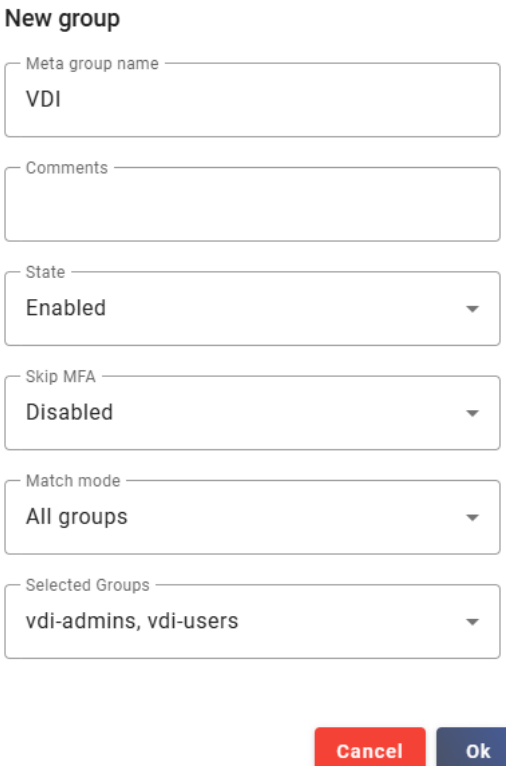
No autorizar a más usuarios/grupos, ni asignar más permisos de los estrictamente necesarios.

If you want to give access to a single user, you must create a group with that user that registers the user directly.

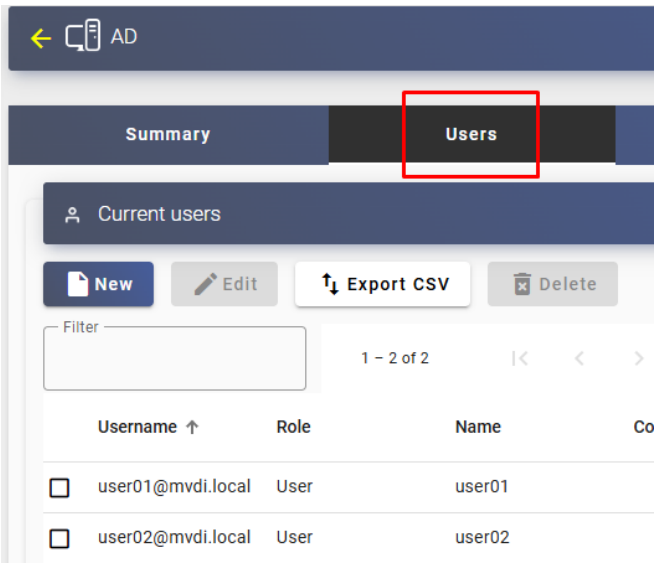
To create a metagroup (group made up of several groups) select the "***Groups***" tab and in "***New***" select "***Meta group***":



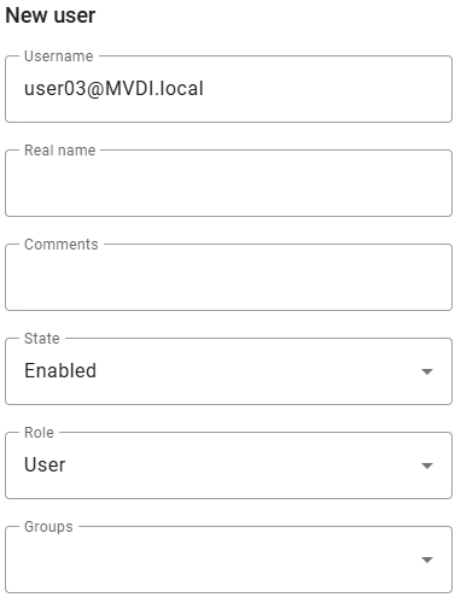
We will indicate the name of the metagroup through the "***Meta group name***" field, its status (enabled or disabled), if we are going to apply MFA method, the mode of operation through the "***Match mode***" field and select the groups that will be part of the metagroup.



In the "***Users***" tab of an authenticator, users will be added automatically when they are validated in the UDS login portal, provided that these users belong to existing groups in the authenticator:



If we need to register users manually, to assign special permissions before they connect for the first time, we will click on "***New***" and select the user, their status (enabled or disabled), the level of access ("***Role***" field). If you do not use the "***Internal Database" authenticator***, we will leave the "***Groups***" field empty, since the system must automatically add the user to your membership group (once the user has been saved, we will be able to edit it and check if it has been correctly assigned to its group).



Roles:

- **Admin**: It makes a user an administrator of the environment, allowing them to access the downloads section and the control panel with both read and edit permissions.

- **Staff Member**: It allows access to the downloads section and the UDS control panel in a limited way based on the permissions assigned to each element.

  To assign permissions to a user with the Staff Role in the different elements of the administration we will have to go to each of the elements and click on "Permissions"".

    - In the permissions window, click on "New permission..." for groups/users and select the authenticator and group/user on which the permission will be applied.
    - We will have to specify whether this user or group will have read access to the element ("Read Only") or full control ("Full Access")”).
    - Permissions will be assigned directly to each element and will also be applied to its sub-elements.
    - "Full Access" ("Manage") permissions can only be applied to elements that have a second level ("Services", "Calendars", "Service Pools", etc…).

- **User**: You will only be allowed to view the UDS services window (this is the role that will be assigned to a user by default).

NOTE: Users of an "Internal Database" authenticator always have to be created manually.

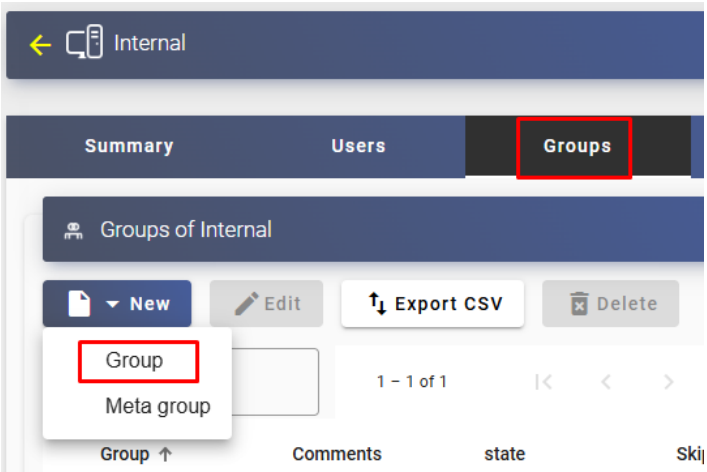
Once the user has been added, we can modify it by clicking on the "***Edit***" button.

To delete a user, a group or metagroup, select it and click on the "***Delete***" button. If we have users registered in the system who belong to a group and it is deleted, the users will be left without an assigned group and will not be able to be validated in the system.

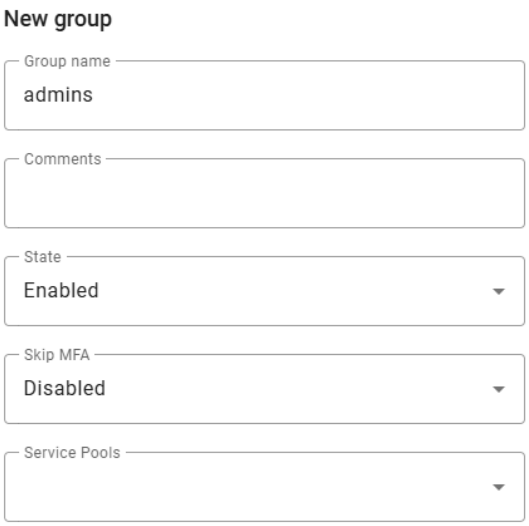
4.3.1 Creating Groups and Users "Internal Database""

In an authenticator of type "***Internal Database***" it will be necessary to manually create groups of users that we will assign to a "***Service Pool***"".

Access the previously created "***Internal Database***" authenticator and in the "***Groups***" tab click on "***New - Group***".



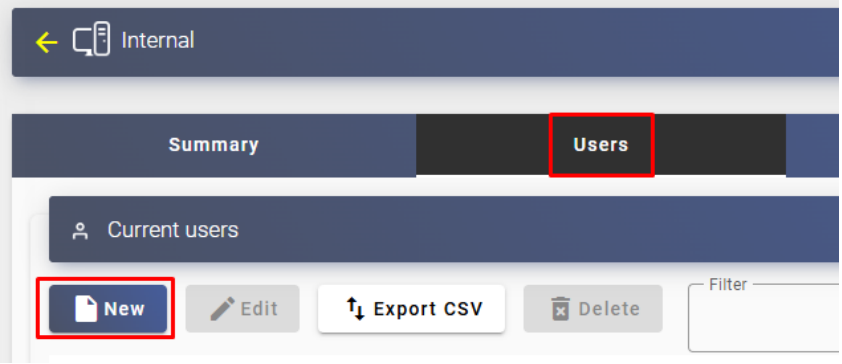
Indicate the name of the new group, its status (enabled or disabled) and we can also assign it directly to one or more "***Service Pools***".



Follow the same procedure if we need to create a metagroup.

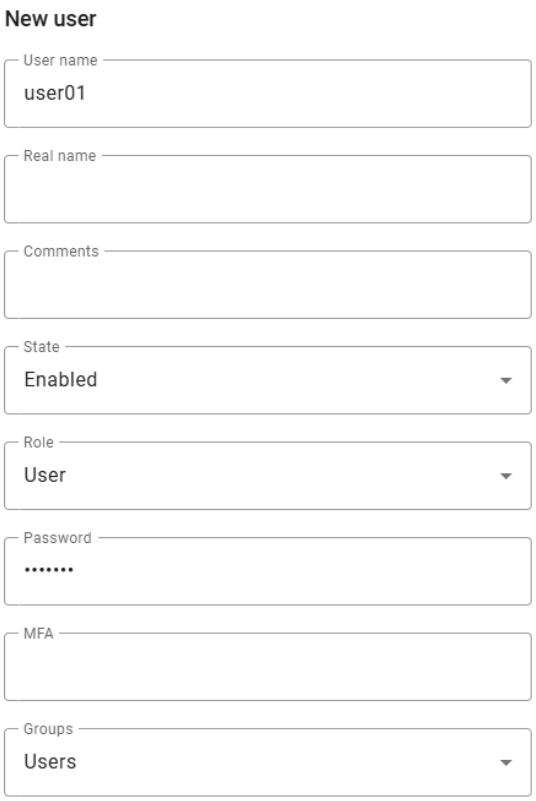
Once we have created the user group or groups, we register users and assign them to one or more groups.

Access the "***Internal Database***" authenticator previously created and in the "***Users***" tab click on "***New***".



We indicate the name of the new user, their status (enabled or disabled), the level of access ("***Role***" field), a password and the MFA attribute in case you have to enable it (for example, for a type of MFA SMS, we will indicate the telephone number or for the email type, we will indicate the email).

In the "***Groups***" field, we will indicate which groups it will belong to (from the existing groups we can select one or more).



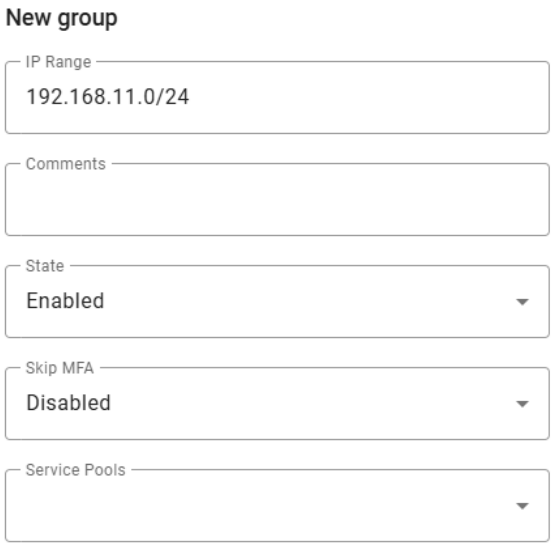
4.3.2 Creating groups and users "IP Authenticator""

In an authenticator of type "***IP Authenticator***" it will be necessary to manually create groups of users. In this case a group will be a range of IPs, an entire subnet, or comma-separated IP addresses. In each case we will use the following format:

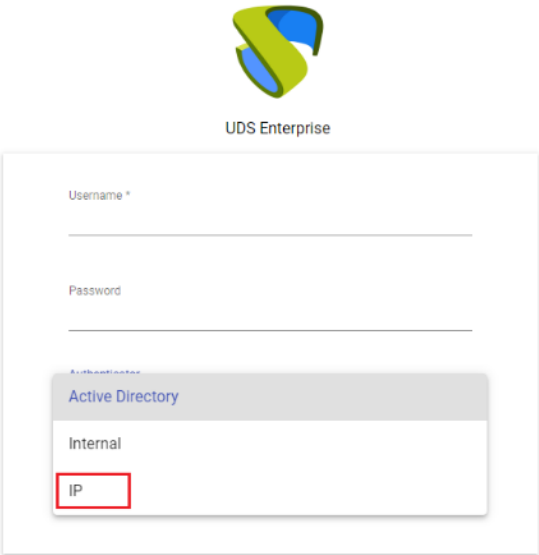
- IP Addresses: xxx.xxx.xxx.xxx,xxx.xxx.xxx.xxx (For example: 192.168.11.33,192.168.11.50)
- Full Subnet: xxx.xxx.xxx.xxx/x (For example: 192.168.11.0/24)
- IP address range: xxx.xxx.xxx.xxx-xxx.xxx.xxx.xxx (For example: 192.168.11.1-192.168.11.155)

Access the previously created "***IP Authenticator***" and in the "***Groups***" tab click on "***New-Group***".

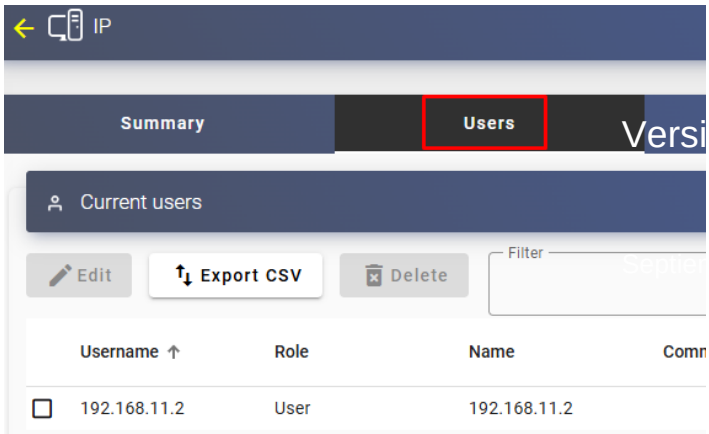
We indicate a range of IPs, an entire subnet or IP addresses separated by commas ("***IP Range***" field), if we are going to apply MFA method and its status (enabled or disabled). In addition, we can assign it directly to one or more "***Service Pools'***".



When a user selects this authenticator in the UDS login portal (i.e. the default authenticator), a check of the IP address of their connection client will be performed. If this address is within the range indicated in the group ("***IP Range***" field), it will automatically validate you.



Once the user is validated in the UDS login portal, their IP address will be automatically registered in the "Users tab"".



If we need to change its status (enabled or disabled) or access level ("***Role***" field), we will select it and click on "Edit".

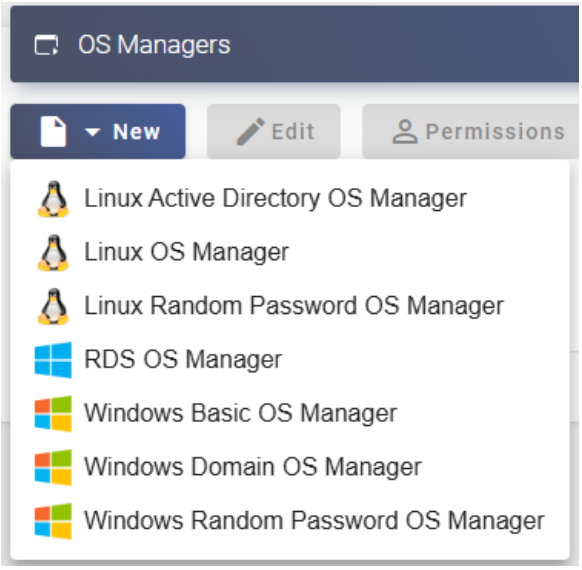
5. OS MANAGERS

An OS ***Manager*** is an OS manager that initiates a previously configured service type.

The UDS Actor, hosted on the virtual desktop or application server, is in charge of the interaction between the OS and the UDS server based on the configurations or type of the assigned ***"OS Manager***".

We will be able to register as many "***OS Managers***" as we need on the UDS Enterprise platform.

You can choose between different types based on the needs of the services to be deployed.



NOTE:

Any type of service deployed in UDS will require an **OS Manager** except when we use the Static **IP Machines Provider**".

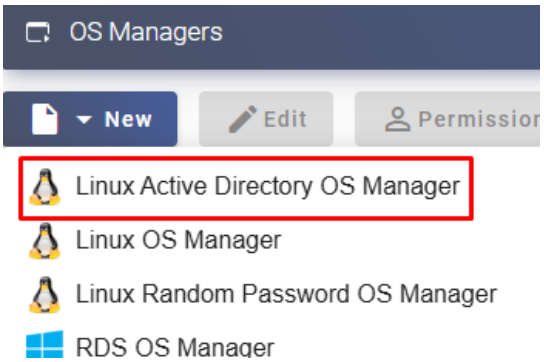
Safe Employment Procedure:

Be very conservative with the actions to be carried out by the OS Manager, it can allow you to create more services than necessary or leave in legacy mode services that are already in disuse.

For many of the actions, the OS Manager needs very special permissions, including administrator permissions, to execute its actions. Do not use the "Administrator" user of the Operating System in OS Managers, create specific users with the right administration permissions for this purpose, so that it is easy to revoke the permissions given to this user assigned to an OS Manager.

5.1.1 Linux Active Directory

A "***Linux Active Directory OS Manager***" is used for virtual desktops based on Linux systems that are part of an AD domain. Performs renaming, inclusion to an AD domain, and session control on virtual desktops.



In a "***Linux Active Directory OS Manager***" the minimum parameters to be configured are:

- Main:

Name: The descriptive name of the element.

Domain: The name of the AD domain to which the virtual desktops will be joined. It is required to use the FQDN format (For example: vc.local), NetBIOS is not supported.

Account: A username with permissions to join machines to the domain.

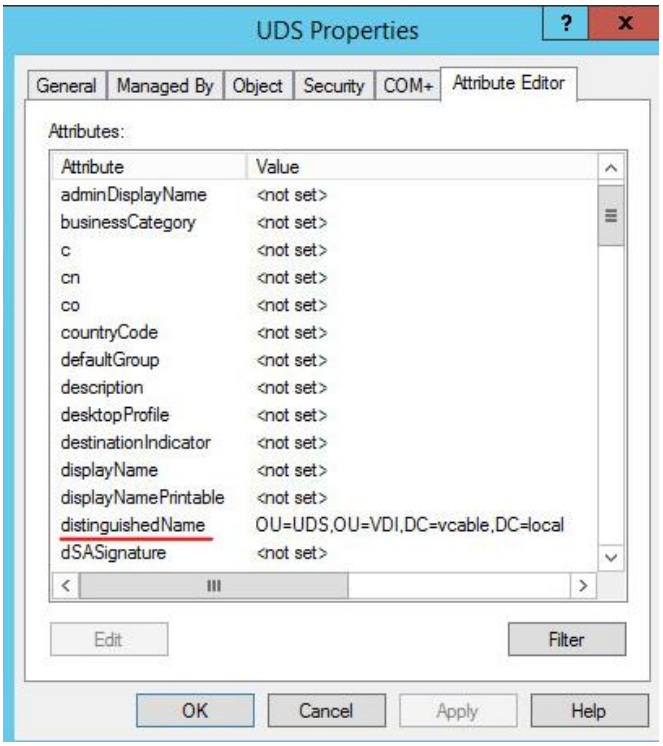
Password: User password in the "Account" field".

Safe use procedure: Passwords must be of sufficient length and include upper and lower case, numbers and special characters.

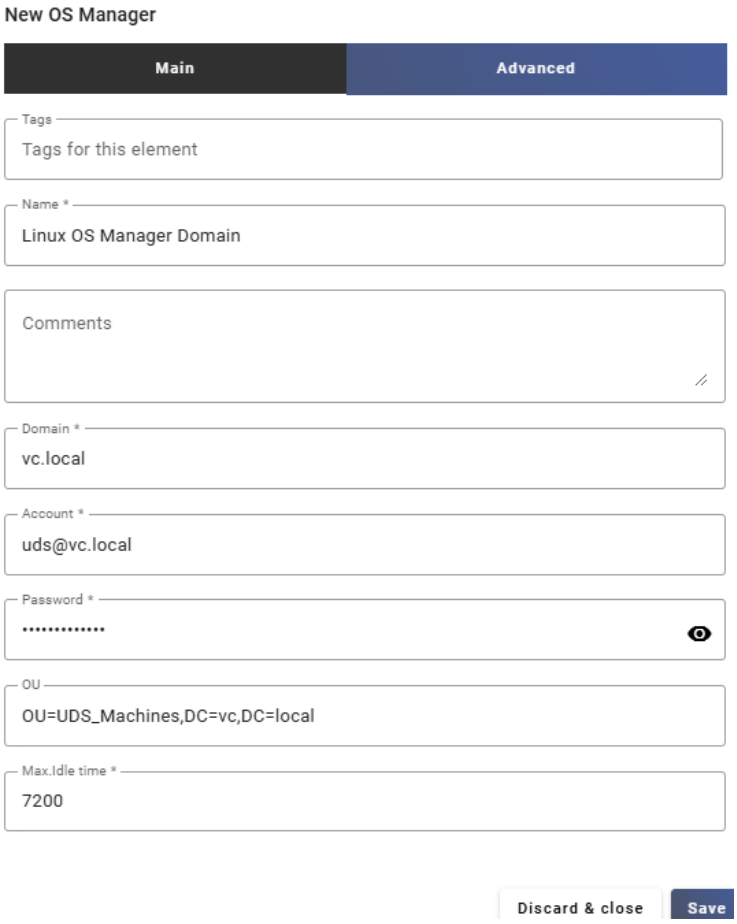
OU: The organizational unit where virtual desktops will be registered (if none is indicated, desktops will be registered in the default OU "***Computers***"). The supported OU format is:

OU=name_OU_Last_level,OU=name_OU_first_level,DC=name_domain,DC=extension_domain

To avoid errors in formatting entry, it is recommended to query the "***distinguishedName***" attribute in the OU properties.



Max. Idle time: Maximum time (indicated in seconds) of inactivity on the virtual desktop. After this time, the UDS Actor will automatically log out of the desktop. Negative values and less than 300 seconds disable this option.



- Advanced:

Client software: Client software to be used

Membership software: Software utilizado durante la unión al dominio.

Machine clean: If enabled, UDS will delete the machine accounts from the virtual desktops in the indicated OU once the desktop is deleted. It is necessary that the user indicated in the "***Account***" field has permissions to perform this action on the indicated OU.

Server software: The type of domain to which machines join.

Use SSL: If enabled, SSL connection will be used to delete machine accounts from the domain.

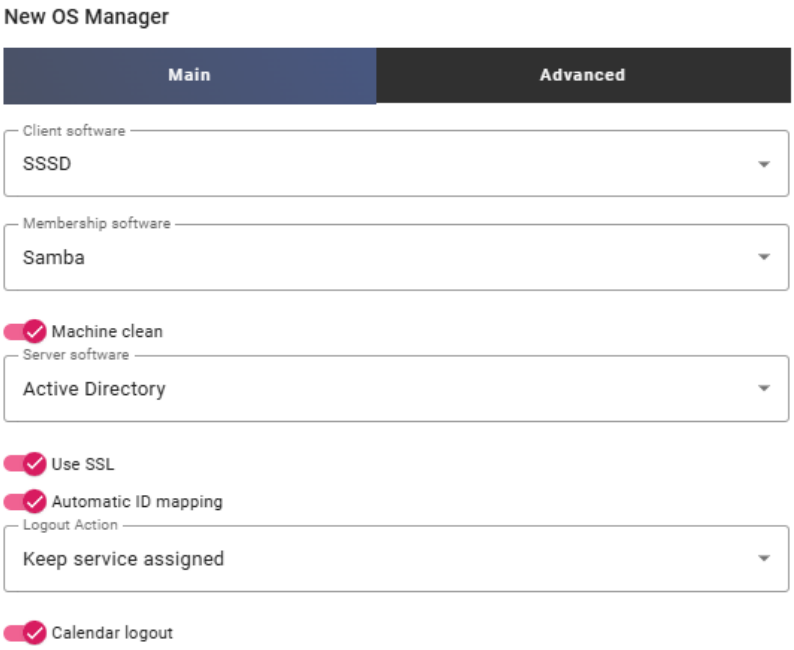
Logout Action: We will indicate the action that UDS will perform on the virtual desktop when a user's session is closed:

- **Keep service assigned (Semi-persistent virtual desktop):** When a user logs out of the virtual desktop, the system will not take any action on the virtual desktop. When you reconnect, you will be assigned the same desktop you had previously been working with. If a new "Service Pool" release is made, when the user logs out, their virtual desktop will be deleted and they will be connected to a new one generated in the new version.

- **Remove service (Non-persistent virtual desktop):** When a user logs out of the virtual desktop, the system deletes the virtual desktop. When you reconnect, you will be assigned a new one.

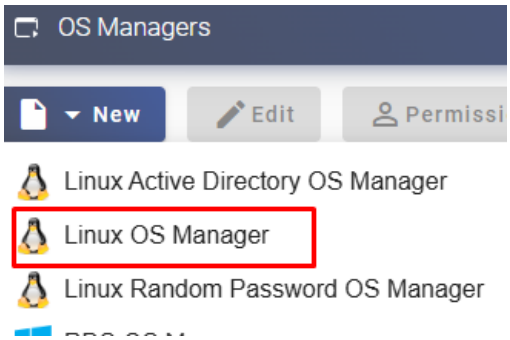
- **Keep service assigned even on new publication (Persistent Virtual Desktop):** When a user logs out of the virtual desktop, the system will not take any action on the virtual desktop. When you reconnect, you will be assigned the same desktop you had previously been working with. If a new publication of the "***Service Pool***" is made, when the user logs out, their virtual desktop will remain assigned and will only be deleted when the administrator indicates it.

Calendar logout: Allows you to select how the system will behave with a user's session when a calendar's end date is reached. If it is set to "***Yes***", when the end date of the calendar assigned to the service is reached, the system will disconnect the user from said service, if it is set to "***No***", the system will not disconnect the user, but if the user is disconnected, logs out or even the connection is cut off, they will no longer be able to connect again.


New OS Manager
Main
Advanced
Client software
SSSD
Membership software
Samba
Machine clean
Server software
Active Directory
Use SSL
Automatic ID mapping
Logout Action
Keep service assigned
Calendar logout

5.1.2 Linux

A "***Linux OS Manager***" is used for virtual desktops based on Linux systems. Performs renaming and session control on virtual desktops.

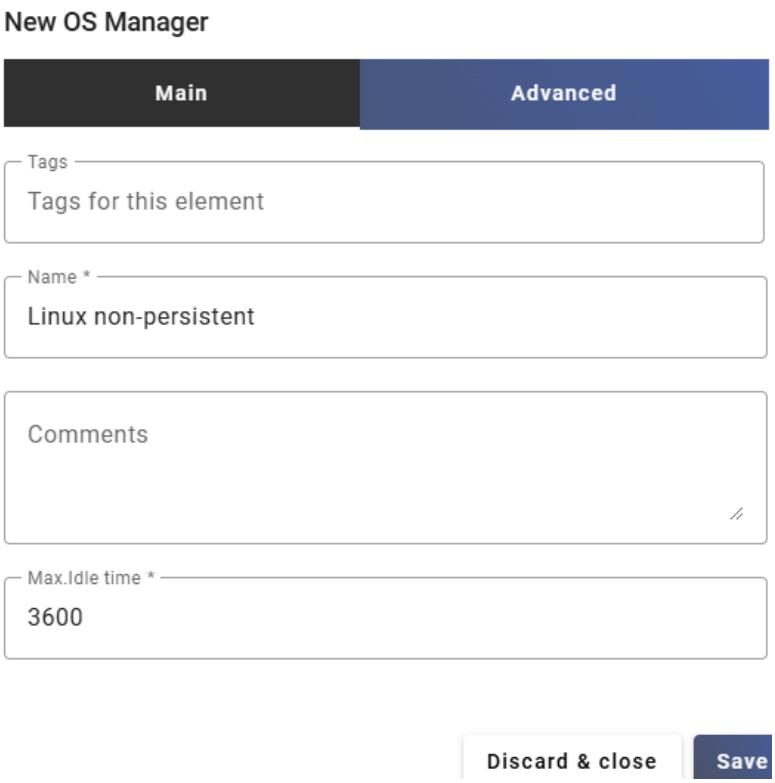


In a "***Linux OS Manager***" the minimum parameters to be configured are:

- Main:

Name: The descriptive name of the element.

Max. Idle time: Maximum idle time (indicated in seconds) on the virtual desktop. After this time, the UDS Actor will automatically log out of the desktop. Negative values and less than 300 seconds disable this option.



- Advanced:

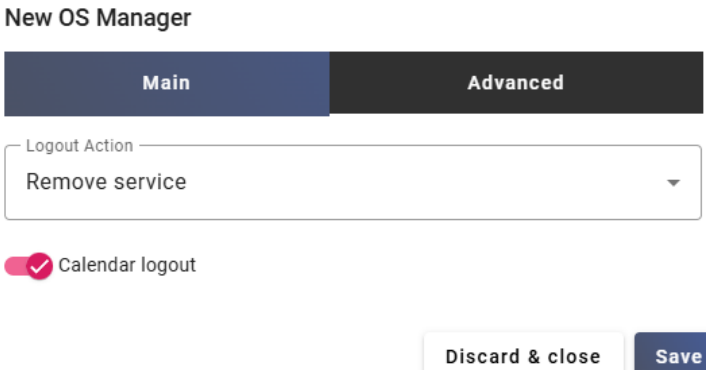
Logout Action: We will indicate the action that UDS will perform on the virtual desktop when a user's session is closedio:

- **Keep service assigned (Semi-persistent virtual desktop):** When a user logs off the virtual desktop, the system will take no action on it. When you reconnect, you will be assigned the same desktop you had previously been working with. If a new "Service Pool" release is made, when the user logs out, their virtual desktop will be deleted, and they will be connected to a new one generated in the new version.

- **Remove service (Non-persistent virtual desktop):** When a user logs off the virtual desktop, the system deletes the virtual desktop. When you reconnect, you will be assigned a new one.

- **Keep service assigned even on new publication (Persistent Virtual Desktop):** When a user logs out of the virtual desktop, the system will take no action on the virtual desktop. When you reconnect, you will be assigned the same desktop you had previously been working with. If a new publication of the "***Service Pool***" is made, when the user logs out, their virtual desktop will remain assigned and will only be deleted when the administrator indicates it.

Calendar logout: Allows you to select how the system will behave with a user's session when a calendar's end date is reached. If it is set to "***Yes***", when the end date of the calendar assigned to the service is reached, the system will disconnect the user from said service, if it is set to "***No***", the system will not disconnect the user, but if the user is disconnected, logs out or even the connection is cut off, they will no longer be able to connect again.



5.1.3 Linux Random Password

A "***Linux Random Password OS Manager***" is used for virtual desktops based on Linux systems and that require a higher level of security in user access. Performs the tasks of renaming, session control, and password change of an existing local user on virtual desktops.

By using it, a random password is assigned to an existing local user during the configuration of each new virtual desktop deployed, thus providing a higher level of security in access.

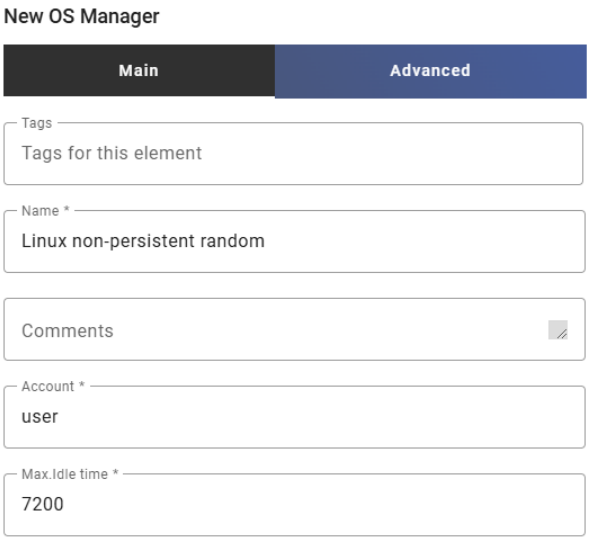
In a "***Linux Random Password OS Manager***" the minimum parameters to be configured are:

- Main:

Name: The descriptive name of the element.

Account: Name of a local user existing in the virtual desktop to which UDS will change the password for an autogenerated random one.

Max. Idle time: Maximum idle time (indicated in seconds) on the virtual desktop. After this time, the UDS Actor will automatically log out of the desktop. Negative values and less than 300 seconds disable this option.



- Advanced:

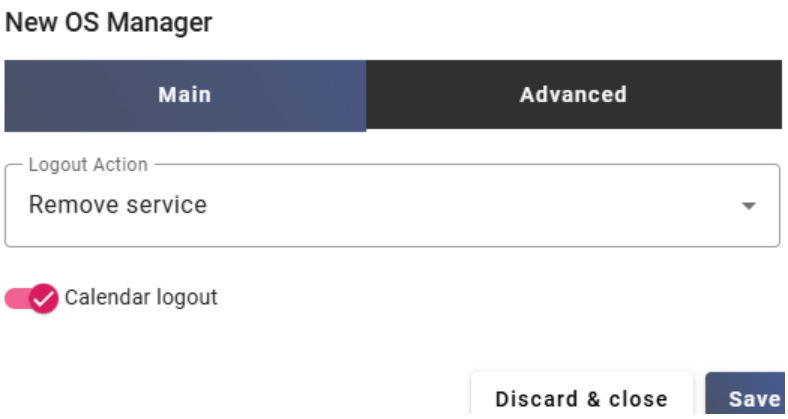
Logout Action: We will indicate the action that UDS will perform on the virtual desktop when a user's session is closed:

- **Keep service assigned (semi-persistent virtual desktop):** When a user logs out of the virtual desktop, the system will not take any action on it. When you reconnect, you will be assigned the same desktop you had previously been working with. If a new "Service Pool" release is made, when the user logs out, their virtual desktop will be deleted and they will be connected to a new one generated in the new version.

- **Remove service (Non-persistent virtual desktop):** When a user logs out of the virtual desktop, the system deletes the virtual desktop. When you reconnect, you will be assigned a new one.

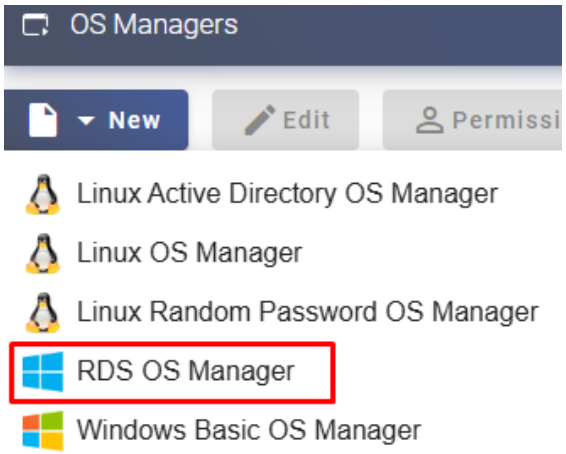
- **Keep service assigned even on new publication (Persistent Virtual Desktop):** When a user logs out of the virtual desktop, the system will take no action on the virtual desktop. When you reconnect, you will be assigned the same desktop you had previously been working with. If a new publication of the "***Service Pool***" is made, when the user logs out, their virtual desktop will remain assigned and will only be deleted when the administrator indicates it.

Calendar logout: Allows you to select how the system will behave with a user's session when a calendar's end date is reached. If it is set to "***Yes***", when the end date of the calendar assigned to the service is reached, the system will disconnect the user from said service, if it is set to "***No***", the system will not disconnect the user, but if the user is disconnected, logs out or even the connection is cut off, they will no longer be able to connect again.



5.1.4 RDS

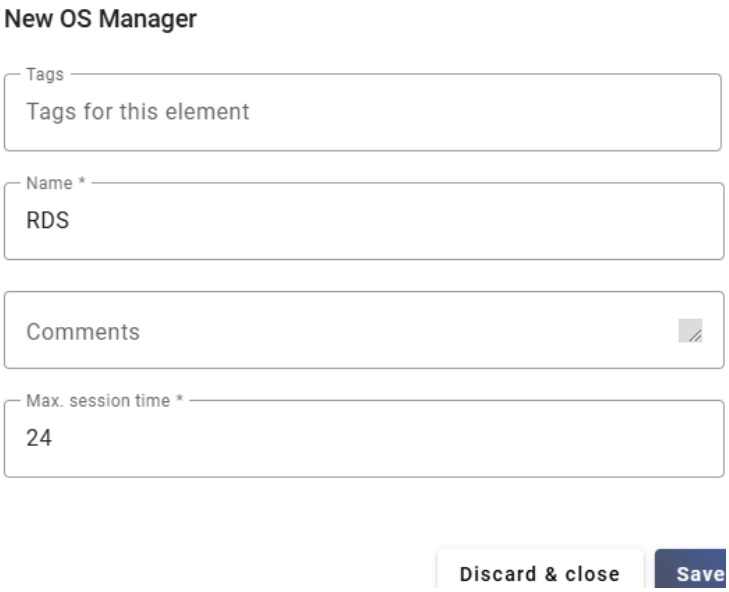
An "RDS ***OS Manager***" is used to configure a "***Service Pool***" that facilitates Windows virtual application sessions to users.



In an "***RDS OS Manager***" the minimum parameters to be configured are:

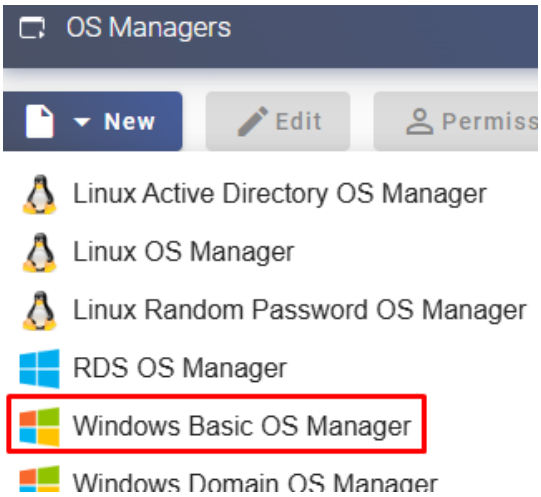
Name: The descriptive name of the element.

Max. Session time: Maximum time that the session of the user of an application will remain registered in the assigned services tab of a "***Service Pool***" expressed in hours (0 = unlimited). Under no circumstances will the user's application session end.



5.1.5 Windows Basic

A "***Windows Basic OS Manager***" is used for virtual desktops based on Windows systems that are not part of an AD domain. Performs renaming and session control on virtual desktops.

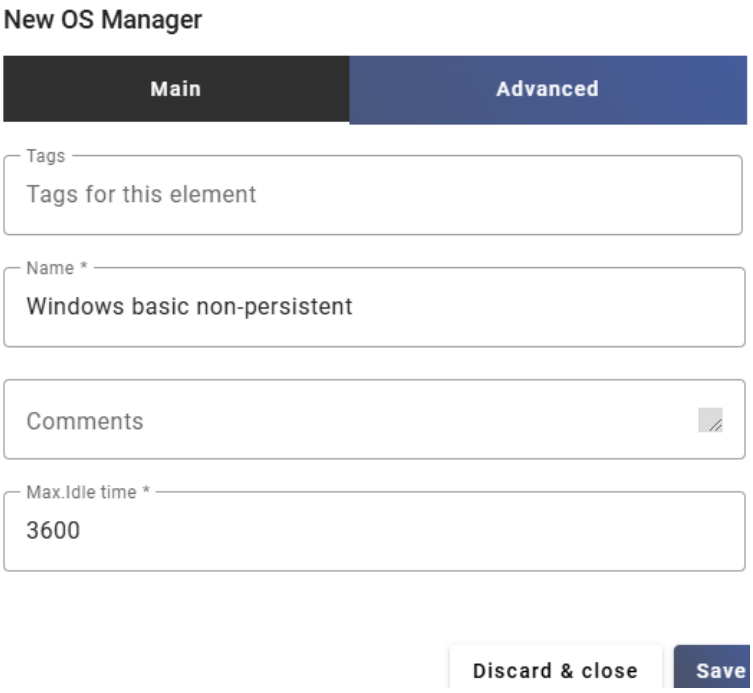


In a "***Windows Basic OS Manager***" the minimum parameters to be configured are:

- Main:

Name: The descriptive name of the element.

Max. Idle time: Maximum idle time (indicated in seconds) on the virtual desktop. After this time, the UDS Actor will automatically log out of the desktop. Negative values and less than 300 seconds disable this option.



- Advanced:

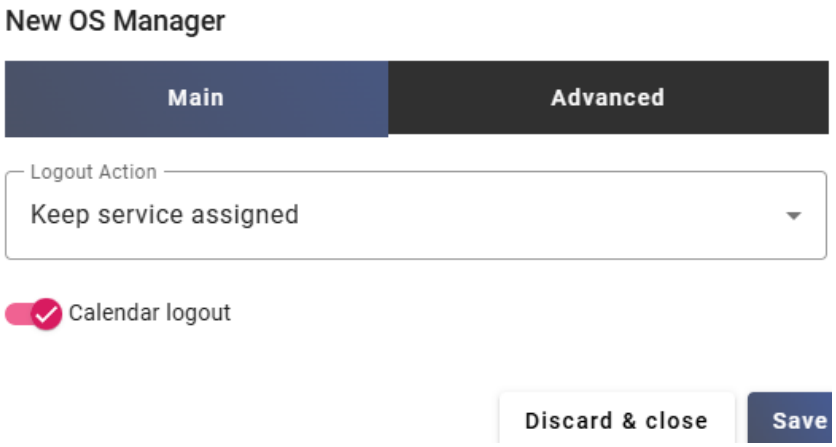
Logout Action: We will indicate the action that UDS will perform on the virtual desktop when a user's session is closed:

- **Keep service assigned (semi-persistent virtual desktop):** When a user logs out of the virtual desktop, the system will not take any action on it. When you reconnect, you will be assigned the same desktop you had previously been working with. If a new service pool release is made, when the user logs out, their virtual desktop will be deleted and they will be connected to a new one generated in the new version.

- **Remove service (Non-persistent virtual desktop):** When a user logs off the virtual desktop, the system deletes the virtual desktop. When you reconnect, you will be assigned a new one.

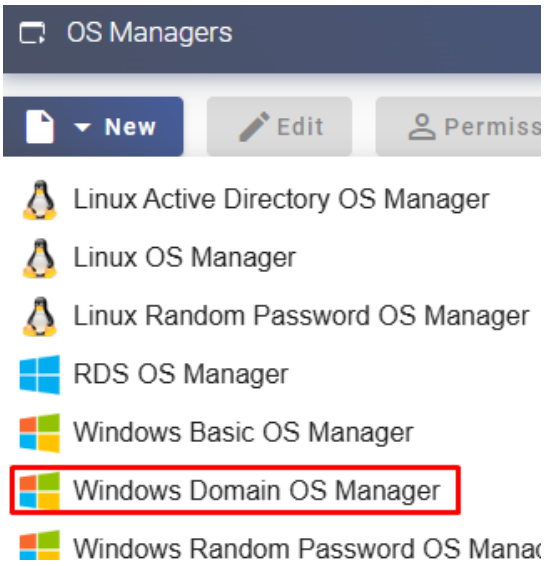
- **Keep service assigned even on new publication (Persistent Virtual Desktop):** When a user logs out of the virtual desktop, the system will take no action on the virtual desktop. When you reconnect, you will be assigned the same desktop you had previously been working with. If a new publication of the "***Service Pool***" is made, when the user logs out, their virtual desktop will remain assigned and will only be deleted when the administrator indicates it.

Calendar logout: Allows you to select how the system will behave with a user's session when a calendar's end date is reached. If it is set to "***Yes***", when the end date of the calendar assigned to the service is reached, the system will disconnect the user from said service, if it is set to "***No***", the system will not disconnect the user, but if the user is disconnected, logs out or even the connection is cut off, they will no longer be able to connect again.



5.1.6 Windows Domain

A "***Windows Domain OS Manager***" is used for virtual desktops based on Windows systems that are part of an AD domain. It performs the tasks of renaming, inclusion to an AD domain, and session control on virtual desktops.



In a "***Windows Domain OS Manager***" the minimum parameters to be configured are:

- Main:

Name: The descriptive name of the element.

Domain: The name of the AD domain to which the virtual desktops will be joined. It is required to use the FQDN format (For example: vc.local), NetBIOS is not supported.

Account: A username with permissions to join machines to the domain.

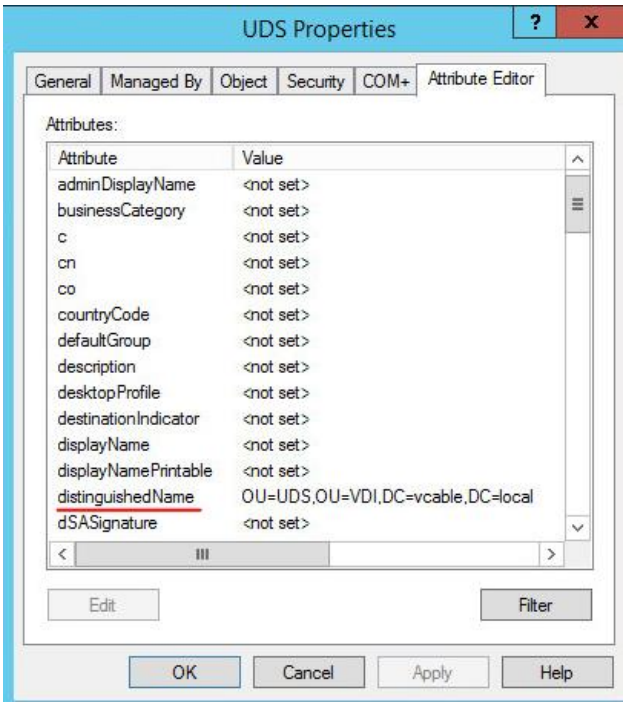
Password: User password in the "Account" field".

Safe use procedure: Passwords must be of sufficient length and include upper and lower case, numbers and special characters.

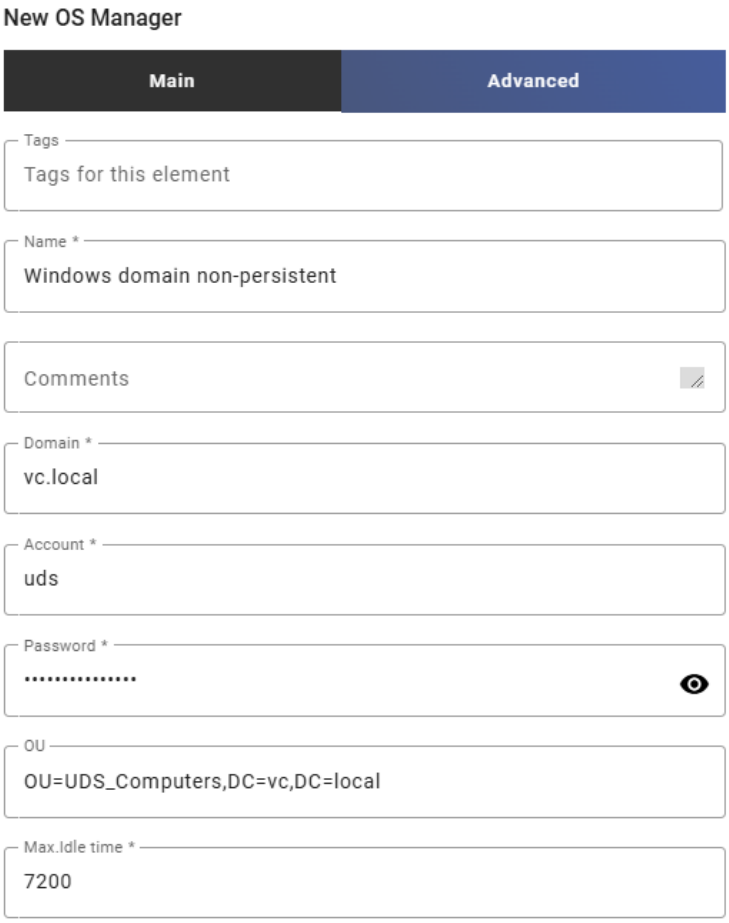
OU: The organizational unit where virtual desktops will be registered (if none is indicated, desktops will be registered in the default OU "***Computers***"). The supported OU format is:

OU=name_OU_last_level,OU=name_OU_first_level,DC=name_domain,DC=extension_domain

To avoid errors in formatting entry, it is recommended to query the "***distinguishedName***" attribute in the OU properties.



Max. Idle time: Maximum time (indicated in seconds) of inactivity on the virtual desktop. After this time, the UDS Actor will automatically log out of the desktop. Negative values and less than 300 seconds disable this option.



- Advanced:

Machine Group: Indicates to which machine group in an AD the UDS-generated virtual desktops will be added.

Machine clean: If enabled, UDS will delete the virtual desktop records in the indicated OU once the desktop is deleted. It is necessary that the user indicated in the "***Account***" field has permissions to perform this action on the indicated OU.

Server Hint: If there are several AD servers, an indication will be given as to which one to use preferably.

Logout Action: We will indicate the action that UDS will perform on the virtual desktop when a user's session is closed:

- **Keep service assigned (Semi-persistent virtual desktop):** When a user logs off the virtual desktop, the system will take no action on it. When you reconnect, you will be assigned the same desktop you had previously been working with. If a new "Service Pool" release is made, when the user logs out, their virtual desktop will be deleted and they will be connected to a new one generated in the new version.

- **Remove service (Non-persistent virtual desktop):** When a user logs off the virtual desktop, the system deletes the virtual desktop. When you reconnect, you will be assigned a new one.

- **Keep service assigned even on new publication (Persistent Virtual Desktop):** When a user logs out of the virtual desktop, the system will take no action on the virtual desktop. When you reconnect, you will be assigned the same desktop you had previously been working with. If a new publication of the "***Service Pool***" is made, when the user logs out, their virtual desktop will remain assigned and will only be deleted when the administrator indicates it.

Use SSL: If enabled, SSL connection will be used against the AD server.

Calendar logout: Allows you to select how the system will behave with a user's session when a calendar's end date is reached. If it is set to "***Yes***", when the end date of the calendar assigned to the service is reached, the system will disconnect the user from said service, if it is set to "***No***", the system will not disconnect the user, but if the user is disconnected, logs out or even the connection is cut off, they will no longer be able to connect again.

New OS Manager

Main | Advanced

Machine Group

Group to which add machines on creation. If empty, no group will be usec

Machine clean

Server Hint

In case of several AD servers, which one is preferred (only used for grouf

Logout Action

Remove service

Use SSL

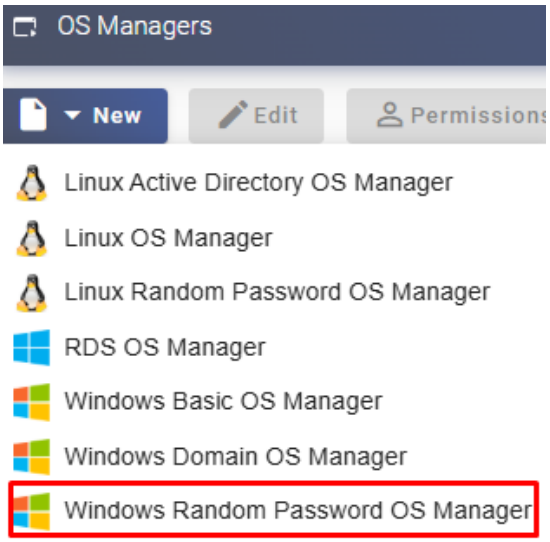
Timeout *

10

Calendar logout

5.1.7 Windows Random Password

A "***Windows Random Password OS Manager***" is used for virtual desktops that are based on Windows systems and require a higher level of security in user access. Performs the tasks of renaming, session control, and password change of an existing local user on virtual desktops.



By using it, an existing local user is assigned a random password during the configuration of each newly deployed virtual desktop, thus providing a higher level of security in access. The password is only known to a UDS administrator

In a "***Windows Random Password OS Manager***" the minimum parameters to be configured are:

- Main:

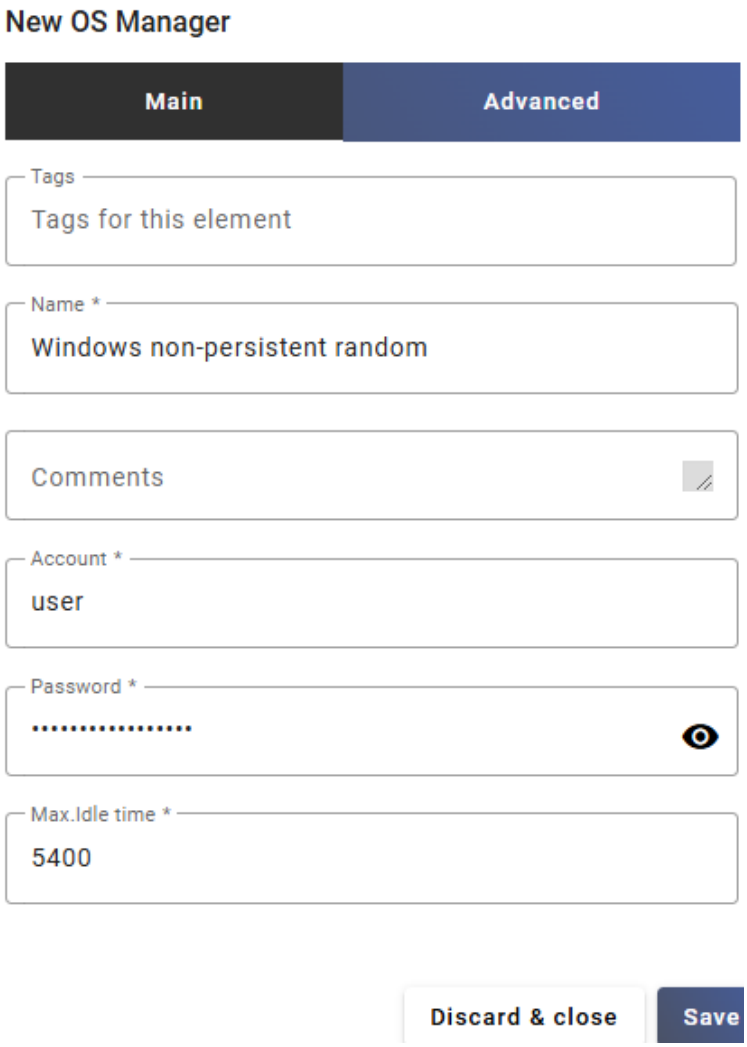
Name: The descriptive name of the element.

Account: Name of a local user existing on the template machine to which UDS will change the password for an autogenerated one in each virtual desktop created.

Password: Password of the user of the "***Account" field***".

Safe use procedure: Passwords must be of sufficient length and include upper and lower case, numbers and special characters.

Max. Idle time: Maximum idle time (indicated in seconds) on the virtual desktop. After this time, the UDS Actor will automatically log out of the desktop. Negative values and less than 300 seconds disable this option.



- Advanced:

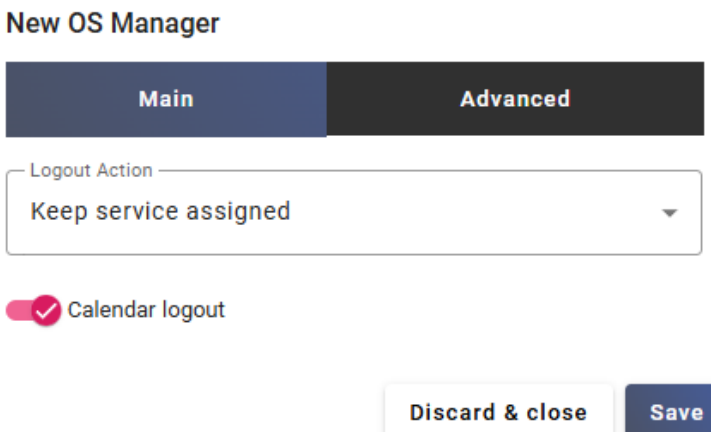
Logout Action: We will indicate the action that UDS will perform on the virtual desktop when a user's session is closed:

- **Keep service assigned (Semi-persistent virtual desktop):** When a user logs off the virtual desktop, the system will take no action on it. When you reconnect, you will be assigned the same desktop you had previously been working with. If a new "Service Pool" release is made, when the user logs out, their virtual desktop will be deleted and they will be connected to a new one generated in the new version.

- **Remove service (Non-persistent virtual desktop):** When a user logs off the virtual desktop, the system deletes the virtual desktop. When you reconnect, you will be assigned a new one.

- **Keep service assigned even on new publication (Persistent Virtual Desktop):** When a user logs out of the virtual desktop, the system will take no action on the virtual desktop. When you reconnect, you will be assigned the same desktop you had previously been working with. If a new publication of the "***Service Pool***" is made, when the user logs out, their virtual desktop will remain assigned and will only be deleted when the administrator indicates it.

Calendar logout: Allows you to select how the system will behave with a user's session when a calendar's end date is reached. If it is set to "***Yes***", when the end date of the calendar assigned to the service is reached, the system will disconnect the user from said service, if it is set to "***No***", the system will not disconnect the user, but if the user is disconnected, logs out or even the connection is cut off, they will no longer be able to connect again.



6. CONNECTIVITY

6.1 Transports

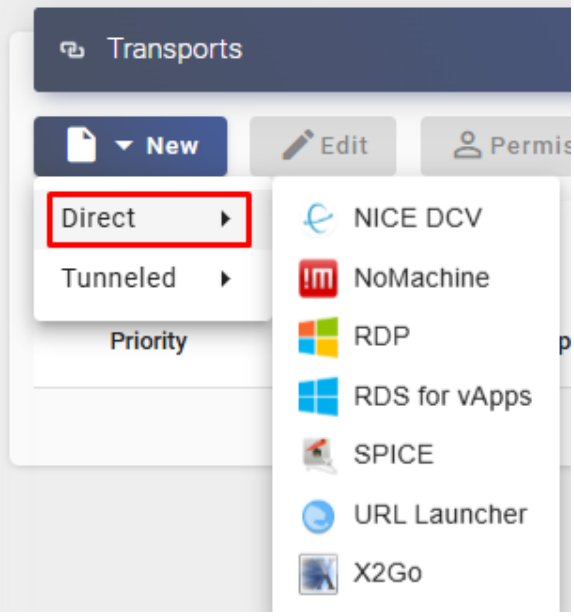
In order to be able to connect to the "***Service Pools***" it is necessary to create "***Transports***". These are applications that will be executed on the connection client and will be responsible for providing access to the implemented service.

Depending on the type of service we configure, the location and the way from which we connect to our virtual desktops, we will need to create different types of transports.

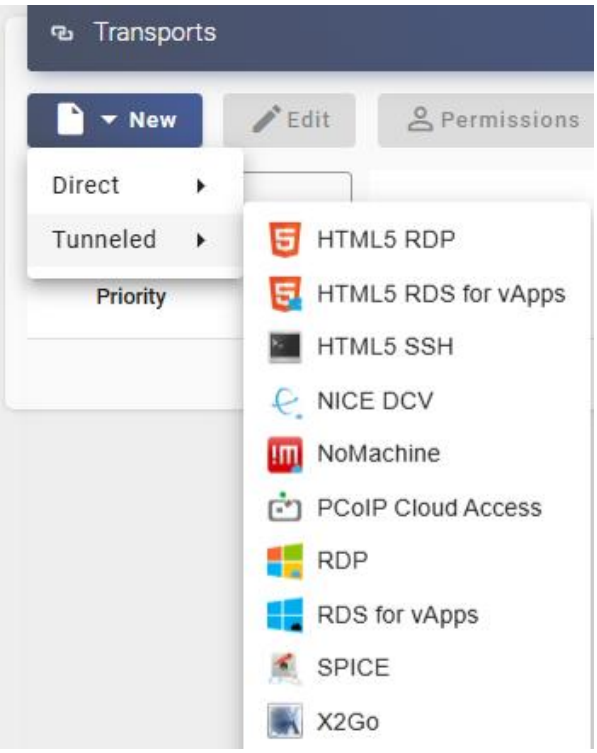
The connection client and the application desktop/server must have the connection protocol (client – server) used in the transport installed for proper transport operation.

To access the "***Transports***" section, we will hover over the "***Connectivity***" section and select "***Transports***". The following direct and tunneled tunnels are currently available:

- Direct:



- Tunneled:



The "***Transports***" that are indicated as "***direct***" will be used to access users to "***Service Pools***" from an internal LAN, VPN, LAN Extension, etc...

The "***Transports***" that are indicated as "***Tunneled***" will be used for user access to "***Service Pools***" from a WAN. These "***Transports***" will rely on the UDS Tunnel server to make the connection.

HTML5 "***Transport***" can be used to access users to "***Service Pools***" from all types of accesses (LAN, WAN, etc...). This "***Transport***" uses the UDS Tunnel server to make the connection against the virtual desktops and applications.

NOTE: If we need to access from a network that is not our LAN, we will have to use tunneled transports, which must have the public IP address of the UDS-Tunnel server indicated. The tunnel server will use two ports depending on the service to be offered. When it is a connection via HTML5, we will use port 10443 and when it is any tunneled connection (RDP, RDS, X2Go, etc...) it will use port 443.

Safe Employment Procedure:

UDS Enterprise configures the actions that a "client" that connects through it can perform on the "server"

In any case, when you install and/or enable the protocol on the "server" limit the actions that can be performed from the "client".

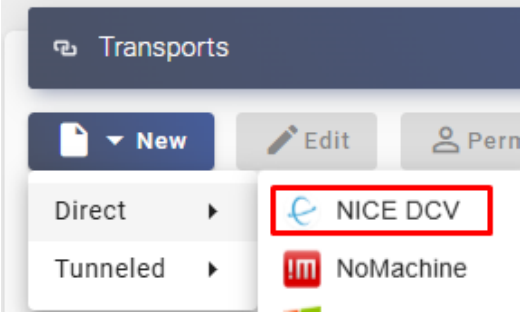
In this way, it will prevent a malicious "client" not managed by UDS that manages to access a "server" poorly protected by network policies from being able to perform on it.

6.1.1 NICE DCV

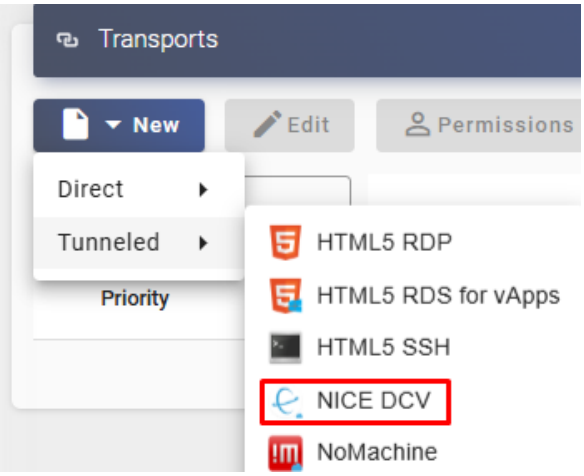
A "***NICE DCV***" transport allows users to access Windows, Linux desktops using the NICE DCV protocol. **Both connecting clients and desktops must have NICE DCV software installed and enabled.**

This transport can be configured directly and in a tunneled way. We will choose each type depending on the user's need for access.

Direct transport will be used when a user has IP connectivity with the desktop service, such as when they are on a LAN, there is connectivity via VPN, etc.…



We will use tunneled transport when the user does NOT have IP connectivity to the desktop service, such as when they are on a WAN. This type of transport uses the UDS Tunnel server to make the connection against the desktops, having to be previously configured for its correct operation.



In a "***NICE DCV***" transport the minimum parameters to be configured are:

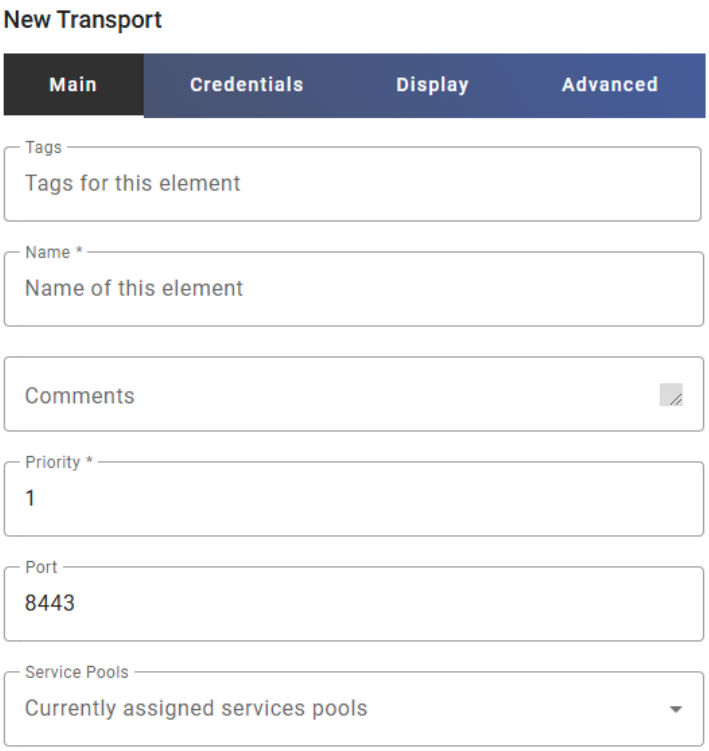
- Main:

Name: Name of the transport.

Priority: Priority that the transport will have. The lower that priority, the higher it appears in a service's list of available transports. The transport with the lowest priority will be the one used by default when you click on the image of a service.

Port: Port that will verify the UDS server (and the tunnel server when tunneling type is used) before allowing the user to connect. If this port is not accessible to the desktop to which the connection will be made, the connection will fail.

Service Pools: Allows you to assign this transport directly to one or more ***previously created*** Service Pools.

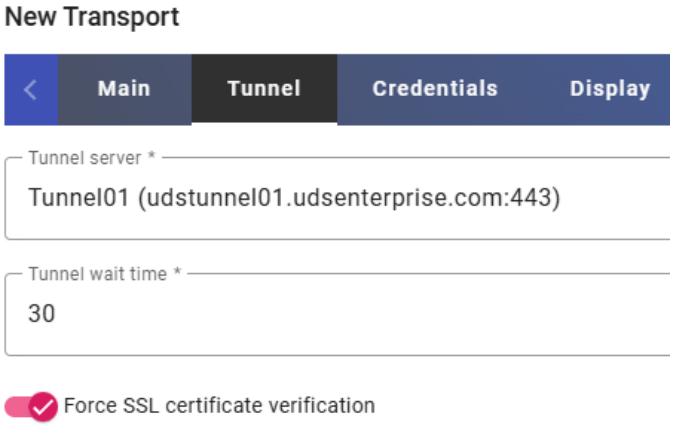


- Tunnel **(This tab will only be available in the tunneled type)**:

Tunnel Server: Select the tunnel that will make the connection. This element will be registered in "***Connectivity***" - "***Tunnels***".

Tunnel wait time: Validity time of the UDS Tunnel server ticket for the connection.

Force SSL certificate verification: Enables certificate verification. The UDS Tunnel server must have a valid certificate installed.



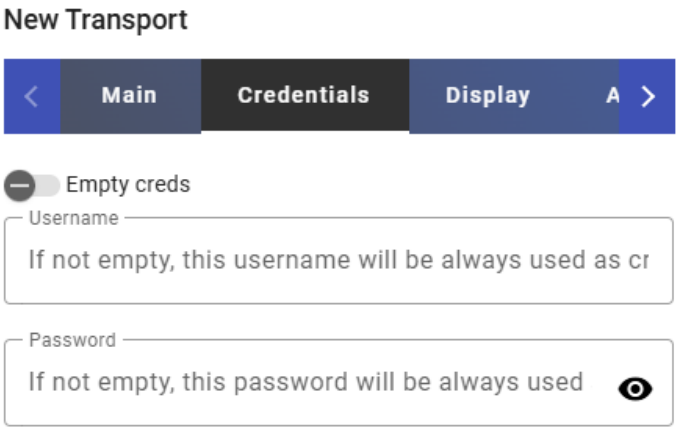
- Credentials:

Empty creds: if it is in "***Yes***", when we make the connection with the service it will ask us for the credentials to access the virtual desktop, in case it is in "***No***" the credentials entered in the UDS login portal will be redirected.

Username: Username that will be used to log in to the desktop (the user must exist on the desktop). If this field is empty, we will try to use the UDS portal login user if the "***Empty creds***" field is set to "***No***", or it will ask for credentials to indicate them manually if it is to "***Yes***".

Password: User password in the "***Username" field***".

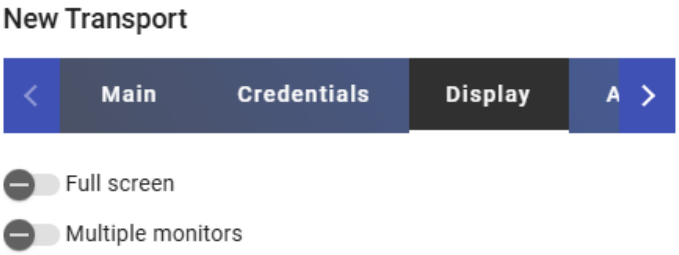
Safe use procedure: Passwords must be of sufficient length and include upper and lower case, numbers and special characters.



- Display:

Full screen: Indicates whether the connection to the service is in full screen.

Multiple monitors: Indicates whether multiple monitors will be used in connection to the service.



- Advanced:

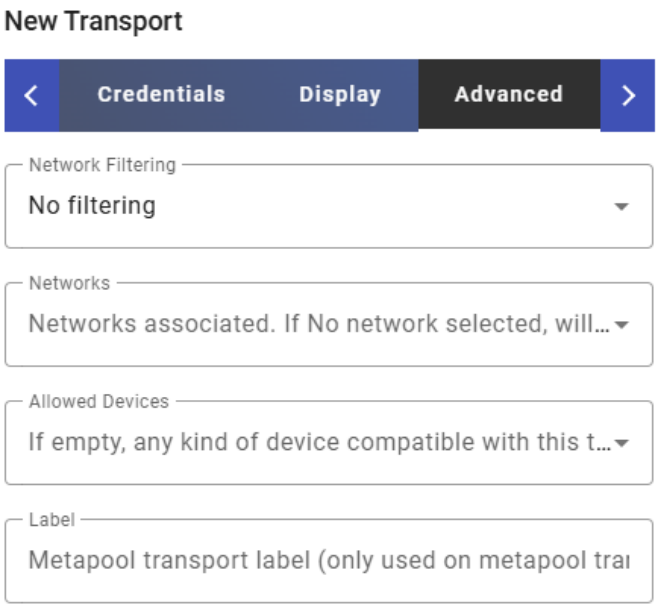
Network Filtering: We will indicate the action to be carried out on the networks indicated in the "***Networks***" section. We may not perform filtering: "***No filtering***", allow only the indicated networks: ***"Allow select networks***" or not allow only the indicated networks: "***Deny selected networks***".

Networks: Network ranges, subnets or IP addresses indicated in the "***Networks***" section of the "Connectivity" section. Used in conjunction with the "***Network Filtering***" field to allow or disable access to transport by users based on their network location.

NOTE: If none is selected, the transport will be visible on any network.

Allowed Devices: Enables access to the service only with selected devices. If none are checked, no filtering is performed.

Label: Allows you to group transports to be displayed, through the name of a label, in a meta pool. By using these tags, we can indicate that a metapool has several transports.

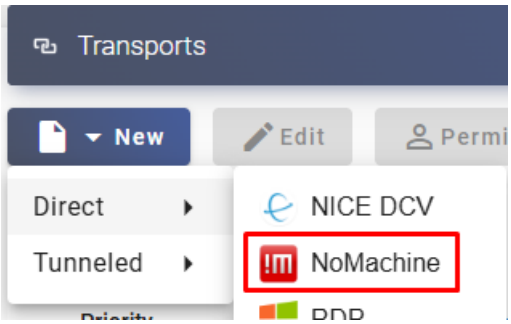


6.1.2 NoMachine

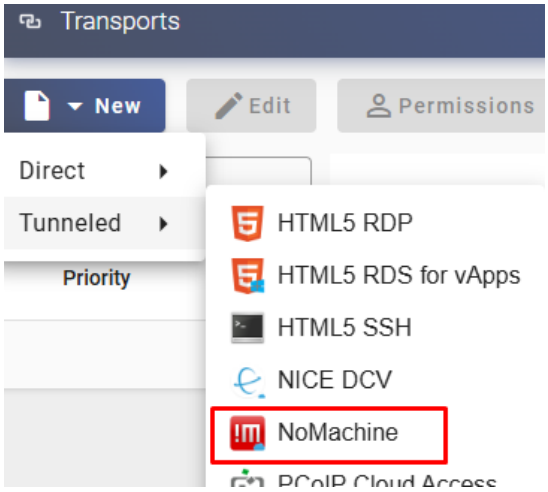
A "***NoMachine Protocol***" transport allows users to access Windows, Linux, and MacOS desktops using the NoMachine protocol. **Both connecting clients and desktops must have NoMachine software installed and enabled.**

This transport can be configured directly and in a tunneled way. We will choose each type depending on the user's need for access.

Direct transport will be used when a user has IP connectivity with the desktop service, such as when they are on a LAN, there is connectivity via VPN, etc.…



We will use tunneled transport when the user does NOT have IP connectivity to the desktop service, such as when they are on a WAN. This type of transport uses the UDS Tunnel server to make the connection against the desktops, having to be previously configured for its correct operation.



In a "***NoMachine***" transport the minimum parameters to be configured are:

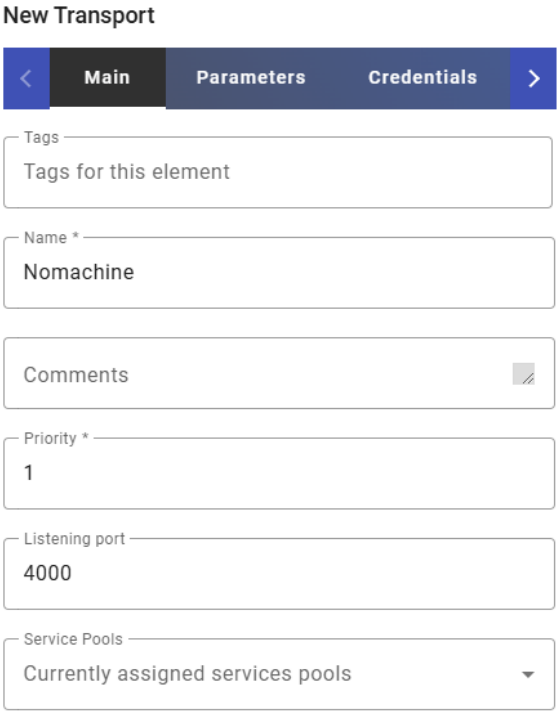
- Main:

Name: Name of the transport.

Priority: Priority that the transport will have. The lower that priority, the higher it appears in a service's list of available transports. The transport with the lowest priority will be the one used by default when you click on the image of a service.

Listening port: Port that will verify the UDS server (and the tunnel server when using the tunneled type) before allowing the user to connect. If this port is not accessible to the desktop to which the connection will be made, the connection will fail.

Service Pools: Allows you to assign this transport directly to one or more ***previously created*** Service Pools.

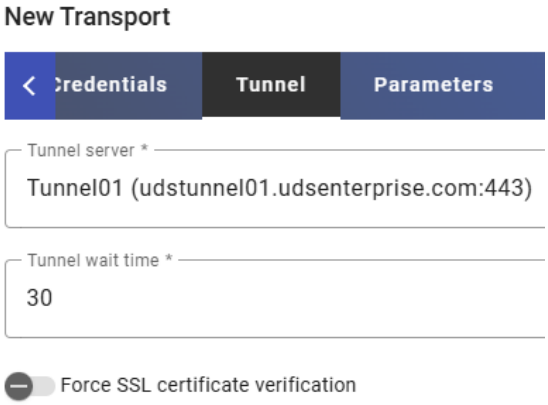


- Tunnel **(This tab will only be available in the tunneled type)**:

Tunnel Server: Select the tunnel that will make the connection. This element will be registered in "***Connectivity***" - "***Tunnels***".

Tunnel wait time: Validity time of the UDS Tunnel server ticket for the connection.

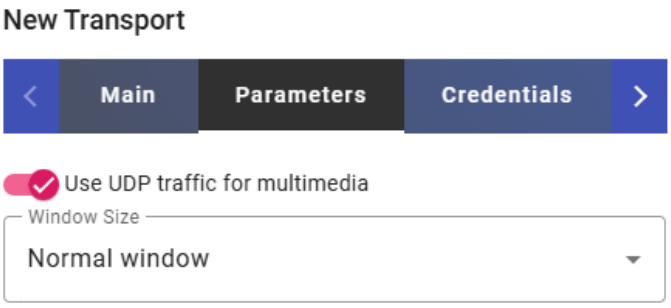
Force SSL certificate verification: Enables certificate verification. The UDS Tunnel server must have a valid certificate installed.



- Parameters:

Use UDP traffic for multimedia: Allows you to indicate whether the UDP protocol will be used in your media traffic.

Window Size: Indicates the size of the window on the connection to the service.



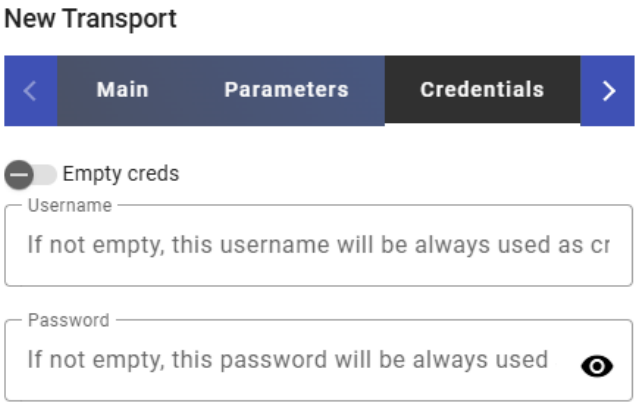
- Credentials:

Empty creds: if it is in "***Yes***", when we make the connection with the service it will ask us for the credentials to access the virtual desktop. If it is set to "***No***", the credentials entered in the UDS login portal will be redirected.

Username: Username that will be used to log in to the desktop (the user must exist on the desktop). If this field is empty, we will try to use the UDS portal login user if the "***Empty creds***" field is in "***No***", or it will ask for credentials to indicate them manually if it is in "***Yes***".

Password: User password in the "***Username" field***".

Secure use procedure: Passwords must be of sufficient length and include upper and lower case, numbers and special characters.



- Advanced:

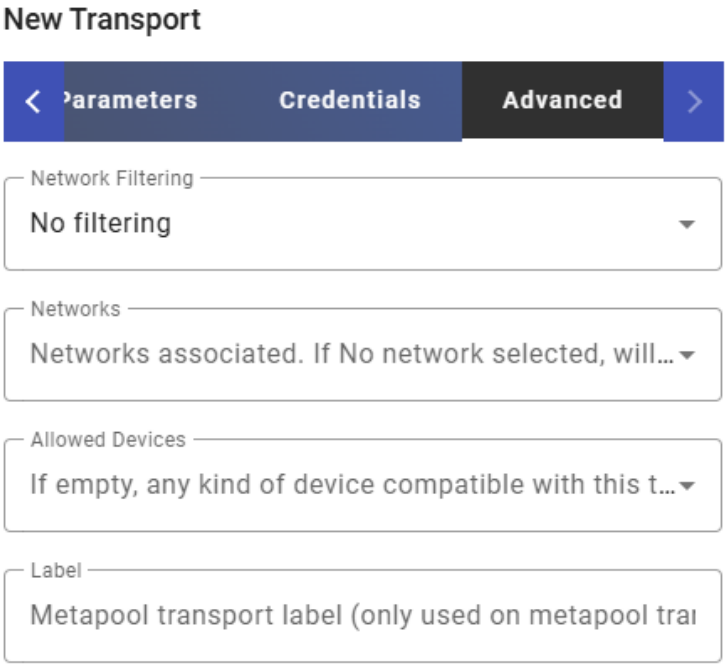
Network Filtering: We will indicate the action to be carried out on the networks indicated in the "***Networks***" section. We may not perform filtering: "***No filtering***", allow only the indicated networks: ***"Allow select networks***" or not allow only the indicated networks: "***Deny selected networks***".

Networks: Network ranges, subnets or IP addresses indicated in the "***Networks***" section of the "Connectivity" section. Used in conjunction with the "***Network Filtering***" field to allow or disable access to transport by users based on their network location.

NOTA: If none is selected, the transport will be visible on any network.

Allowed Devices: Enables access to the service only with selected devices. If none are checked, no filtering is performed.

Label: Allows you to group transports to be displayed, through the name of a label, in a meta pool. By using these tags, we can indicate that a metapool has several transports.

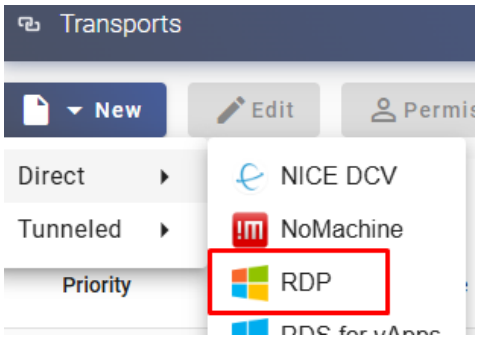


6.1.3 RDP

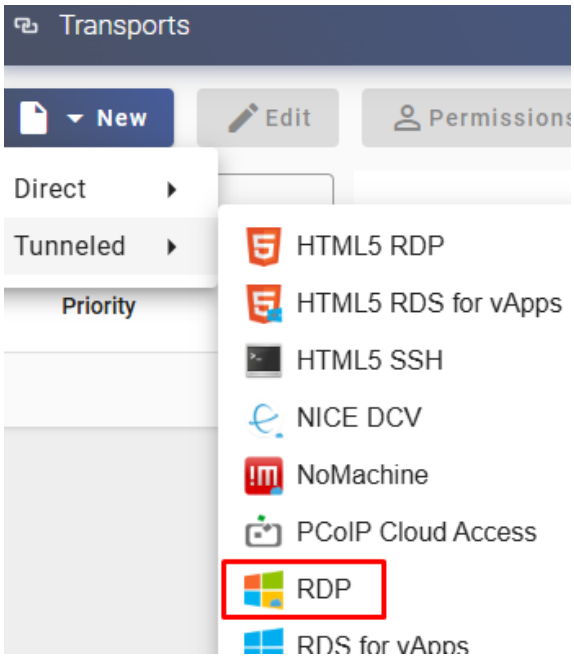
An ***"RDP"*** transport allows users to access Windows and Linux desktops using the Remote Desktop Protocol (RDP). **Both connecting clients and desktops must have RDP installed and enabled** (for Linux desktops, XRDP must be used).

This transport can be configured directly and in a tunneled way. We will choose each type depending on the user's need for access.

Direct transport will be used when a user has IP connectivity with the desktop service, such as when they are on a LAN, there is connectivity via VPN, etc.…



We will use tunneled transport when the user does NOT have IP connectivity to the desktop service, such as when they are on a WAN. This type of transport uses the UDS Tunnel server to make the connection against the desktops, having to be previously configured for its correct operation.



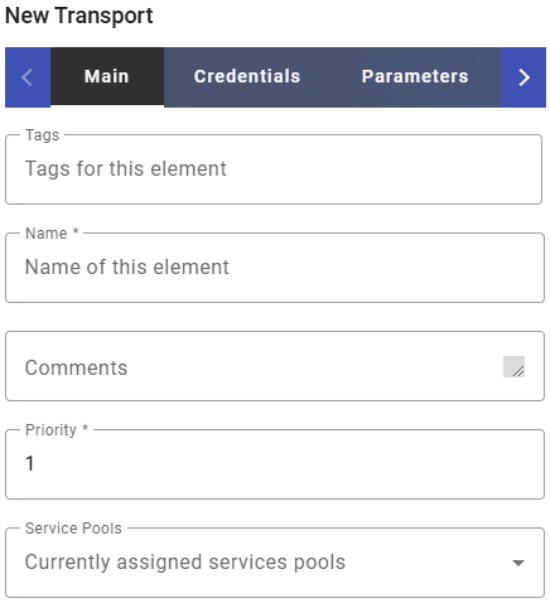
In an ***"RDP" transport*** the minimum parameters to be configured are:

- Main:

Name: Name of the transport.

Priority: Priority that the transport will have. The lower that priority, the higher it appears in a service's list of available transport. Transport with the lowest priority will be the one used by default when you click on the image of a service.

Service Pools: Allows you to assign this transport directly to one or more ***previously created*** Service Pools .

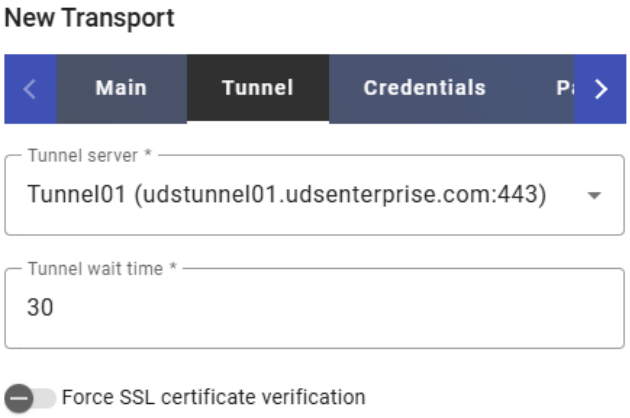


- Tunnel **(This tab will only be available in the tunneled type)**:

Tunnel Server: Select the tunnel that will make the connection. This element will be registered in "***Connectivity***" - "***Tunnels***".

Tunnel wait time: Validity time of the UDS Tunnel server ticket for the connection.

Force SSL certificate verification: Enables certificate verification. The UDS Tunnel server must have a valid certificate installed.



- Credentials:

Empty creds: If it is in "***Yes***", when we make the connection with the service it will ask us for the credentials to access the virtual desktop. If it is set to "***No***", the credentials entered in the UDS login portal will be redirected.

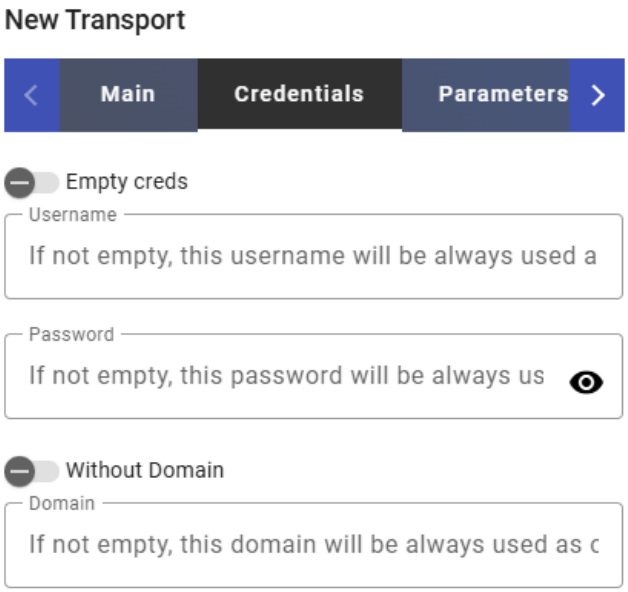
Username: Username that will be used to log in to the desktop (the user must exist on the desktop). If this field is empty, we will try to use the UDS portal login user if the "***Empty creds***" field is in "***No***", or it will ask for credentials to indicate them manually if it is in "***Yes***".

Password: User password in the "***Username" field***".

Safe use procedure: Passwords must be of sufficient length and include upper and lower case, numbers and special characters**.**

Without Domain: Indicates whether the domain name is redirected along with the user.

Domain: Name of the domain that will be sent with the user's credentials.



- Parameters:

Allow Smartcards: Enables Smartcards redirection.

Allow Printers: Enable printer redirection.

Local drives policy: Enable disk drive redirection:

- **Allow none:** No drive is redirected.
- **Allow PnP drives:** Only connected drives are redirected during an active session.
- **Allow any drive:** All units are redirected.

Force drives: Forces the redirection of specific units. Several can be indicated separated by commas. (Ex: F:,G:).

Allow Serials: Enable serial port forwarding.

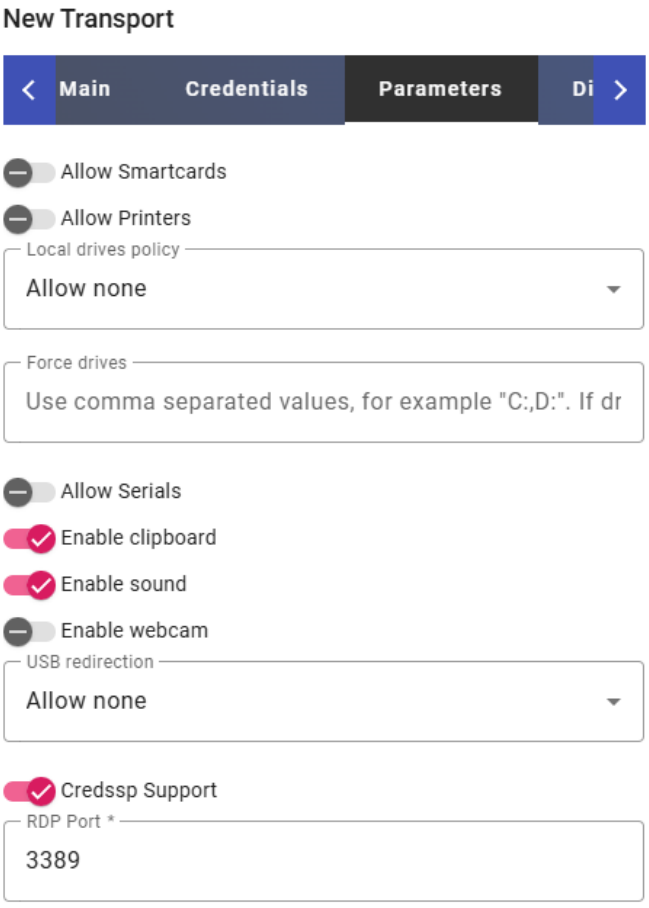
Enable clipboard: If enabled, it will allow copy/paste between the connection client and the desktop.

Enable Sound: If enabled, it will allow audio to be redirected from the desktop to the connecting client.

Enable webcam: If enabled, it will allow webcam redirection between the connection client and the desktop.

Credssp Support: Si está activado usará **“**Credential Security Support Provider”.

RDP Port: Port that will verify the UDS server (and the tunnel server when using the tunneled type) before allowing the user to connect. If this port is not accessible to the desktop to which the connection will be made, the connection will fail.



- Display

Screen Size: Determines the size of the desktop window.

Color depth: Indicates the depth of the color.

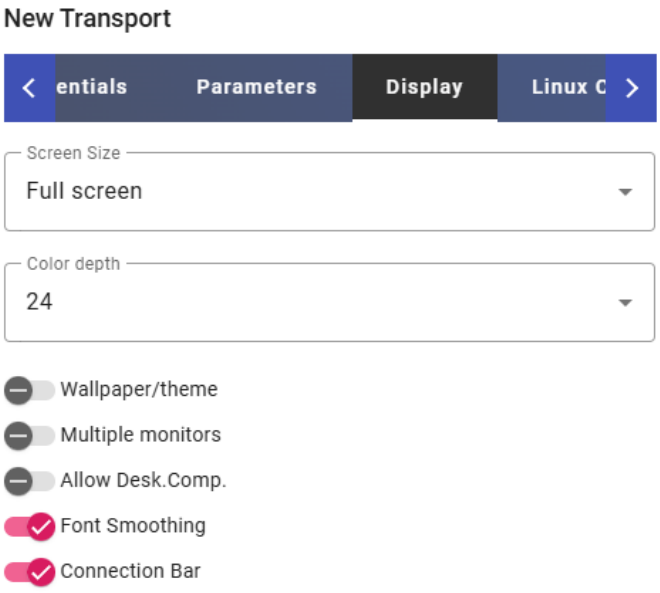
Wallpaper/theme: Displays the desktop wallpaper.

Multiple Monitors: Allows the use of multiple monitors.

Allow Desk. Comp.: Habilita “Desktop Composition”.

Font Smoothing: Enable font smoothing.

Connection Bar: Allows you to enable or disable the connection bar.



- Linux Client (only applies to Linux connection clients):

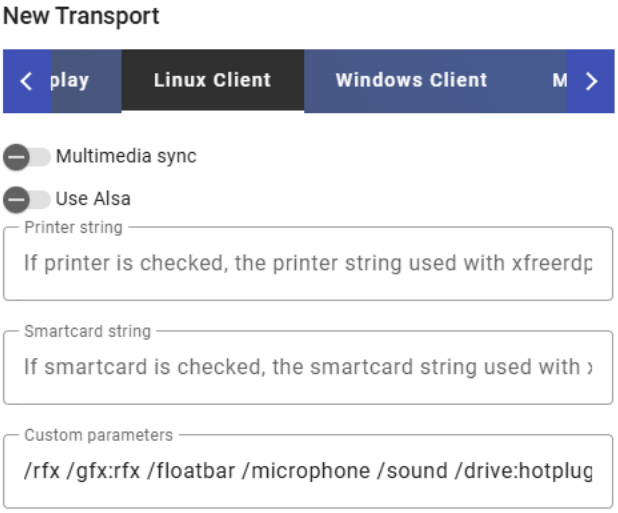
Multimedia sync: Enables the media parameter in the FreeRDP client.

Use Alsa: Enable audio usage through Alsa.

Printer string: ejemplo: "Zebra","ZDesigner TM400 200 dpi (ZPL)"
("Zebra" is the name of the local printer, "ZDesigner TM400 200 dpi (ZPL)" is the exact name of the printer driver in Windows).

Smartcard string: example: "Gemalto PC Twin Reader 00 00"
("Gemalto PC Twin Reader 00 00" *is the name of the smartcard).*

Custom parameters: You can specify any parameter that the FreeRDP client supports (if you indicate several, you have to separate them by spaces). They will be applied when you establish the connection to the desktop. (e.g. /rfx /gfx:rfx /floatbar /microphone /sound /drive:hotplug,* /gfx-h264:avc444)

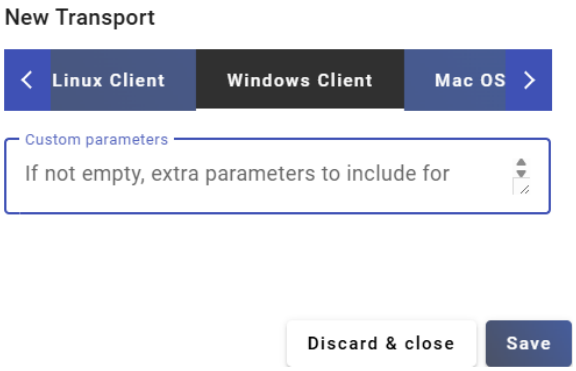


NOTE:
In the screenshot, some important parameters are indicated such as: /gfx:rfx greatly improves the quality of video and audio, /drive:media,/media allows you to redirect the local units connected to the connection client, /smartcard allows you to redirect a smartcard…

- Windows Client (only applies to Windows connection clients):

Custom parameters: List where parameters can be added to the RDP connection.

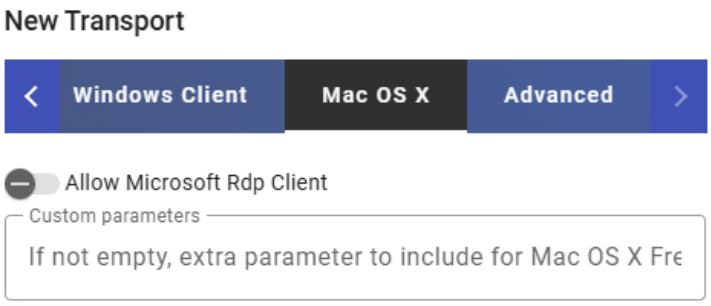
Optimize Teams: If enabled, the UDS client will NOT call the RDP client "remote desktop" installed on Windows by default and it will require the manual installation of a special RDP client that provides an optimization for the "Teams" conferencing application. If this RDP client is not installed, the connection will fail.



- Mac OS X (only applies to MacOS connection clients):

Allow Microsoft Rdp Client: Allows you to use the Microsoft RDP client instead of the FreeRDP client. To be able to use it, it must be enabled and also the FreeRDP client must not be installed on the MacOS connection client computer.

Custom parameters: You can specify any parameter that the FreeRDP client supports (if you specify several, you have to separate them by spaces). They will be applied when establishing the connection to the desktop.



- Advanced:

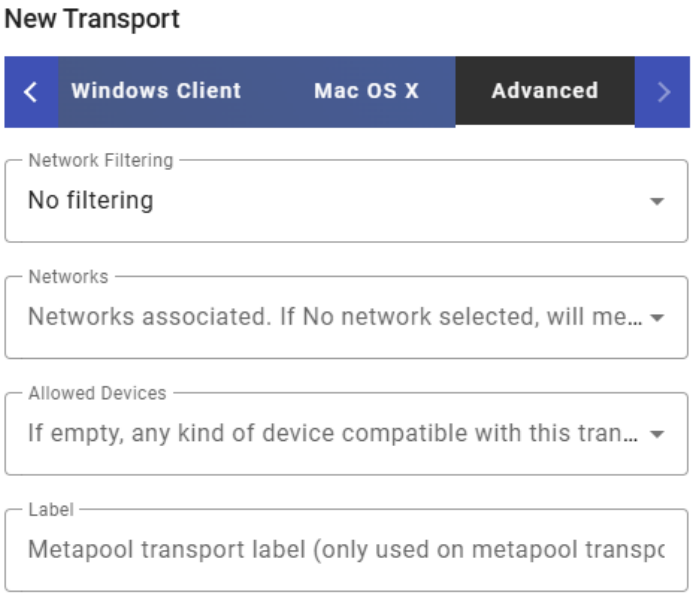
Network Filtering: We will indicate the action to be carried out on the networks indicated in the "***Networks***" section. We may not perform filtering: "***No filtering***", allow only the indicated networks: ***"Allow select networks***" or not allow only the indicated networks: "***Deny selected networks***".

Networks: Network ranges, subnets or IP addresses indicated in the "***Networks***" section of the "Connectivity" section. Used in conjunction with the "***Network Filtering***" field to allow or disable access to transport by users based on their network location.

NOTE: If none is selected, the transport will be visible on any network.

Allowed Devices: Enables access to the service only with selected devices. If none are checked, no filtering is performed.

Label: Allows you to group transports to be displayed, through the name of a label, in a meta pool. By using these tags, we can indicate that a metapool has several transports.



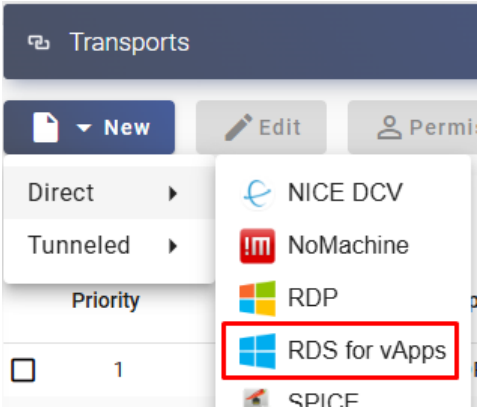
6.1.4 RDS For vApps

An "***RDS for vAPPs***" transport allows users to access Windows virtual applications using RemoteAPP (RDP). Connecting clients can be Windows or Linux systems.

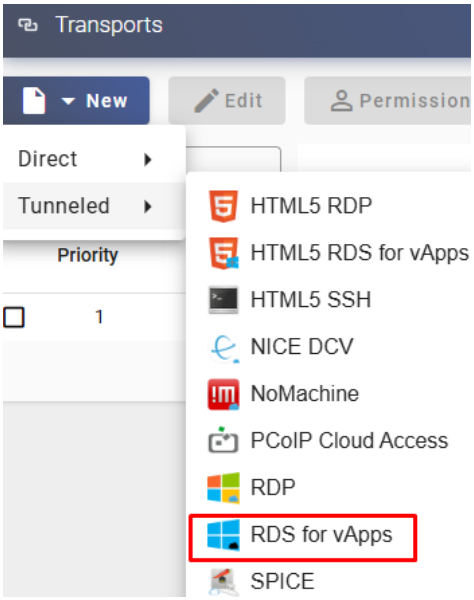
A Windows connection client must have RemoteAPP to open virtual applications. A Linux connection client must have the FreeRDP RDP client to open virtual applications.

This transport can be configured directly and in a tunneled way. We will choose each type depending on the user's need for access.

Direct transport will be used when a user has IP connectivity with the application server, such as when they are on a LAN, there is connectivity via VPN, etc.…



We will use tunneled transport when the user does NOT have IP connectivity to the application server, such as when they are on a WAN. This type of transport uses the UDS Tunnel server to make the connection against the application servers, having to be previously configured for its correct operation.



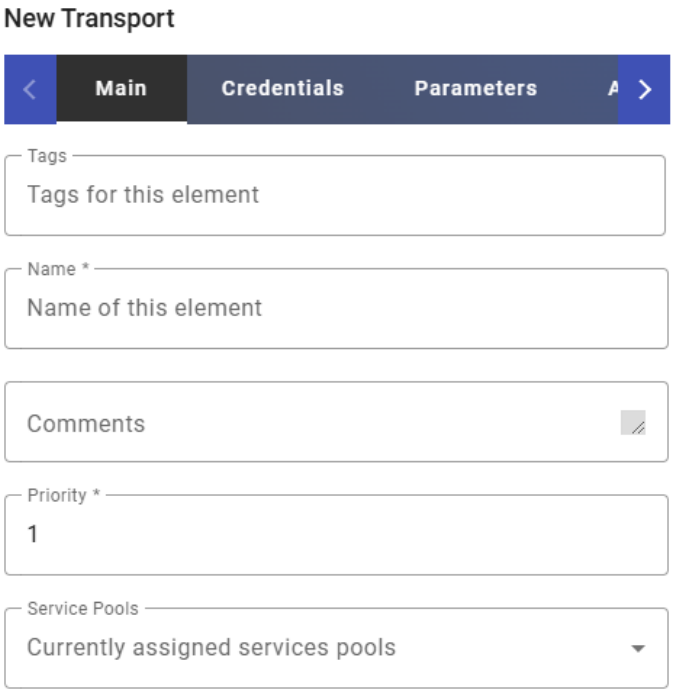
In an "***RDS for vAPPs***" transport, the minimum parameters to be configured are:

- Main:

Name: Name of the transport.

Priority: Priority that the transport will have. The lower that priority, the higher it appears in a service's list of available transports. The transport with the lowest priority will be the one used by default when you click on the image of a service.

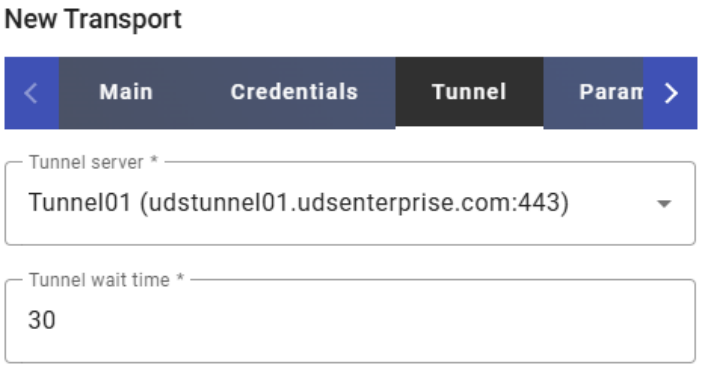
Service Pools: Allows you to assign this transport directly to one or more ***previously created*** Service Pools.



- Tunnel **(This tab will only be available in the tunneled type)**:

Tunnel Server: Select the tunnel that will make the connection. This element will be registered in "***Connectivity***" - "***Tunnels***".

Tunnel wait time: UDS Tunnel server ticket validity time for the connection.

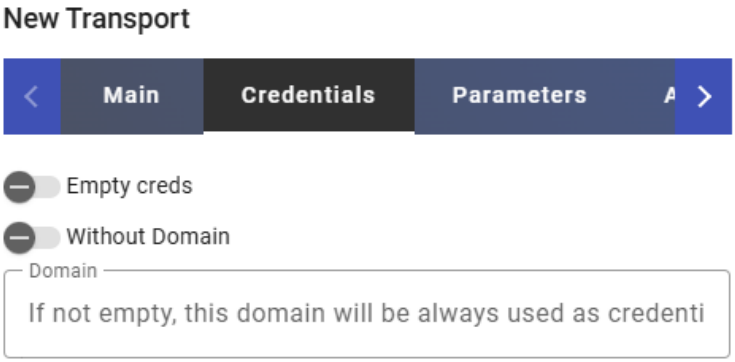


- Credentials:

Empty creds: If it is in "***Yes***", when we connect with the service it will ask us for the credentials to access the virtual application. If it is set to "***No***", the credentials entered in the UDS login portal will be redirected.

Without Domain: Indicates whether the domain name is redirected along with the user.

Domain: Name of the domain that will be sent with the user's credentials.



- Parameters:

Allow Smartcards: Enable Smartcard redirection.

Allow Printers: Enable printer redirection.

Local drives policy: Enable disk drive redirection:

- **Allow none:** No se redirecciona ninguna unidad.
- **Allow PnP drives:** Only connected drives are redirected during an active session.
- **Allow any drive:** All units are redirected.

Force drives: Force redirection of specific drives. Several can be indicated separated by commas. (Ex: F:,G:).

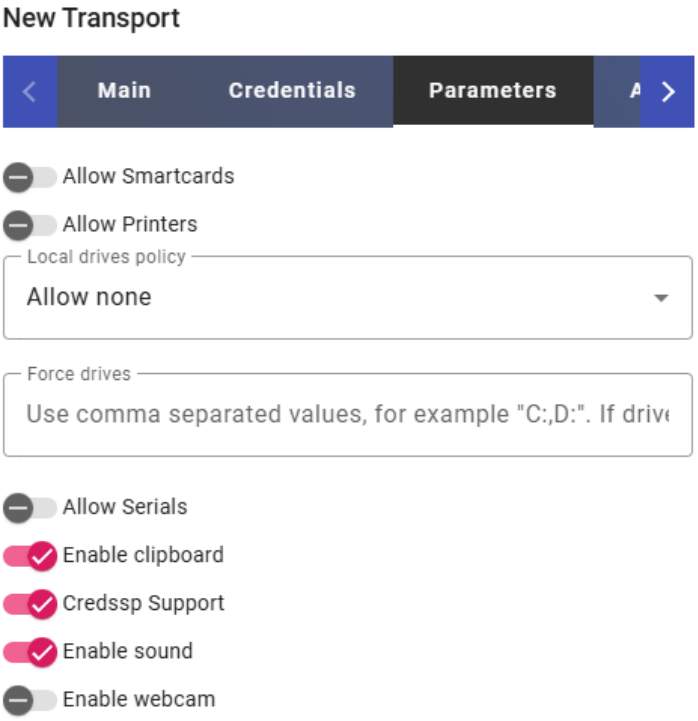
Allow Serials: Enables serial port forwarding.

Enable clipboard: If enabled, it will allow copy/paste between the connection client and the virtual application.

Credssp Support: Si está activado usará **"**Credential Security Support Provider".

Enable Sound: If enabled, it will allow the redirection of audio from the virtual application to the connecting client.

Enable webcam: If enabled it will allow webcam redirection between the connection client and the virtual application



- Advanced:

RDP Port: Connection port against the RDP server.

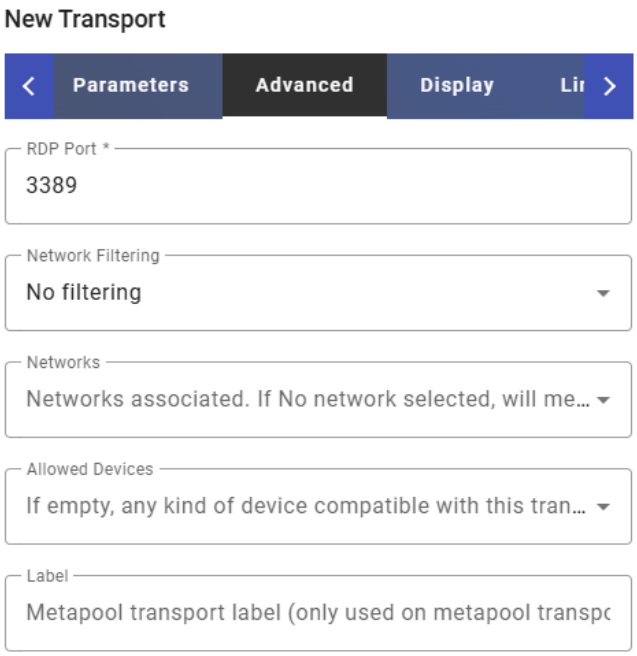
Network Filtering: We will indicate the action to be carried out on the networks indicated in the "***Networks***" section. We may not perform filtering: "***No filtering***", allow only the indicated networks: ***"Allow select networks***" or not allow only the indicated networks: "***Deny selected networks***".

Networks: Network ranges, subnets or IP addresses indicated in the "***Networks***" section of the "Connectivity" section. Used in conjunction with the "***Network Filtering***" field to allow or disable access to transport by users based on their network location.

NOTE: If none is selected, the transport will be visible on any network.

Allowed Devices: Enables access to the service only with selected devices. If none are checked, no filtering is performed.

Label: Allows you to group transports to be displayed, through the name of a label, in a meta pool. By using these tags, we can indicate that a metapool has several transports.

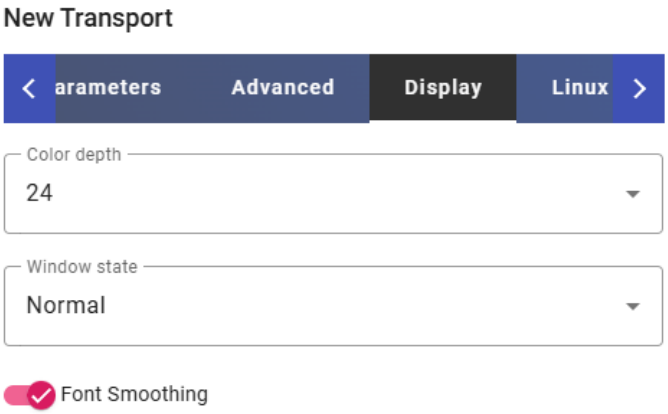


- Display:

Color depth: Indicates color depth.

Windows state: Indicates the window mode in which the application will launch.

Font Smoothing: Turn on font smoothing.



- Linux Client:

Execute as shell: Runs an application in session mode offering better visualization.

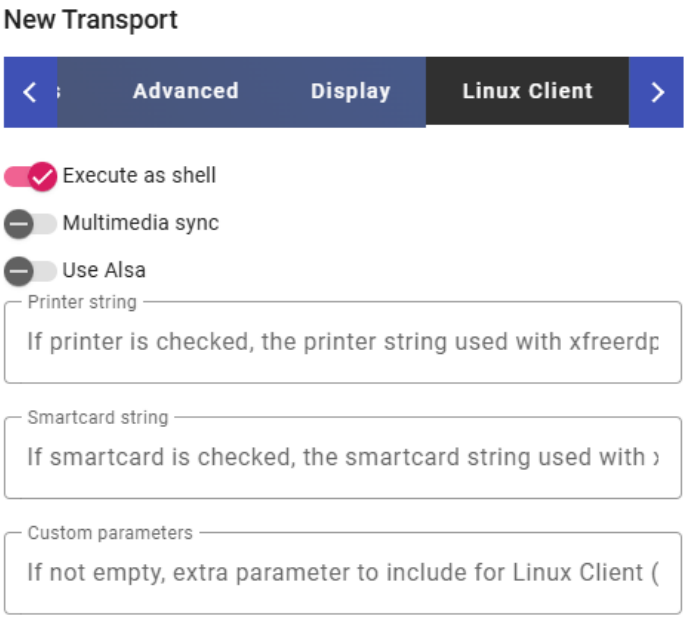
Multimedia sync: Enables the multimedia parameter in freerdp.

Use Alsa: Enable audio usage through Alsa.

Printer string: Ejemplo: "Zebra","ZDesigner TM400 200 dpi (ZPL)"
("Zebra" is the name of the local printer, "ZDesigner TM400 200 dpi (ZPL)" is the exact name of the printer driver in Windows).

Smartcard string: Example: "Gemalto PC Twin Reader 00 00"
("Gemalto PC Twin Reader 00 00" *is the name of the smartcard).*

Custom parameters: Any parameter that the FreeRDP client supports can be specified. They will be applied when establishing the connection to the virtual application.

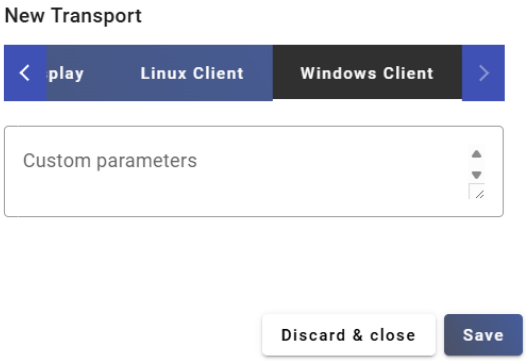


NOTE: For example, parameters such as: /t:"nombre_aplicación" allows you to force a title to the window (if it is not indicated, the name will be that of the service pool), /dynamic-resolution +toggle-fullscreen allows you to maximize the application window…

- Windows Client (only applies to Windows connection clients):

Custom parameters: A list where parameters can be added to the RDP connection.

Optimize Teams: If enabled, the UDS client will NOT call the RDP client "remote desktop" installed on Windows by default and it will require the manual installation of a special RDP client that provides an optimization for the "Teams" conferencing application. If this RDP client is not installed, the connection will fail.



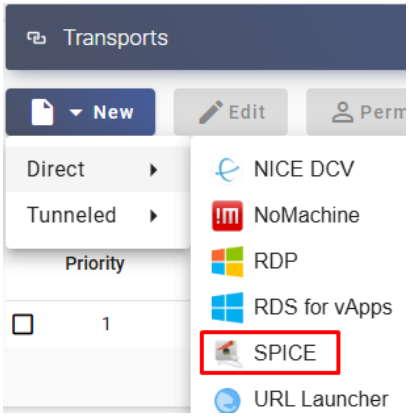
6.1.5 SPICE

A "***SPICE"*** transport allows users to access Windows/Linux desktops using the "***SPICE"*** protocol. **It is necessary for the connecting clients to have the "*SPICE"* (Virt-Manager) client installed**.

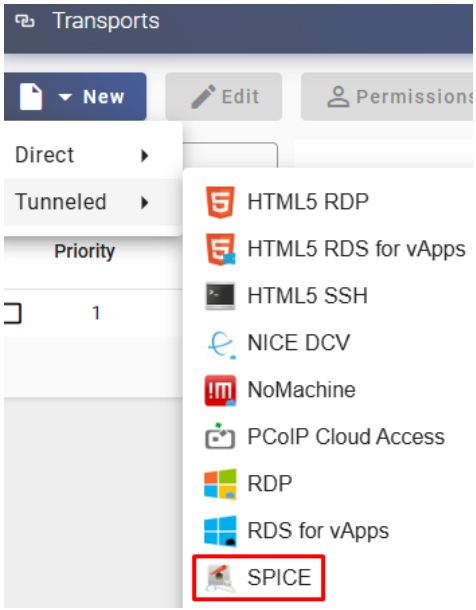
The "***SPICE"*** transport can only be used with a service provider such as Proxmox, oVirt and OpenNebula.

This transport can be configured directly and in a tunneled way. We will choose each type depending on the user's need for access.

Direct transport will be used when a user has IP connectivity with the desktop service, such as when they are on a LAN, there is connectivity via VPN, etc.…



We will use tunneled transport when the user does NOT have IP connectivity to the desktop service, such as when they are on a WAN. This type of transport uses the UDS Tunnel server to make the connection against the desktops, having to be previously configured for its correct operation.



In a "***SPICE" transport*** the minimum parameters to be configured are:

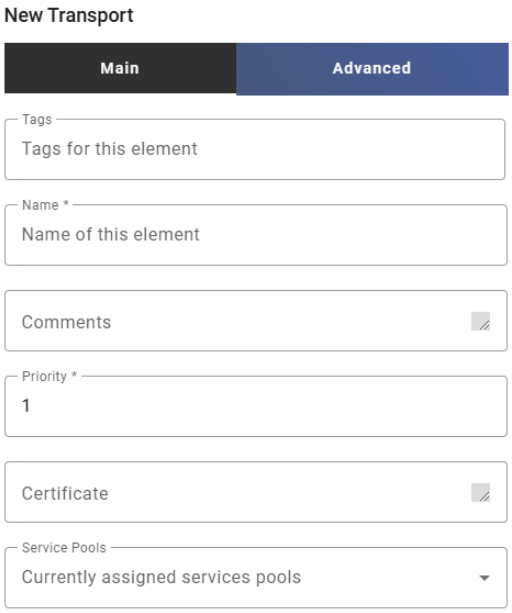
- Main:

Name: Name of the transport.

Priority: Priority that the transport will have. The lower that priority, the higher it appears in a service's list of available transports. The transport with the lowest priority will be the one used by default when you click on the image of a service.

Certificate: Certificate generated in ovirt-engine or OpenNebula. It is required to connect to desktops (usually hosted on **/etc/pki/ovirt-engine/certs/ca.cer**).

Service Pools: Allows you to assign this transport directly to one or more ***previously created*** Service Pools.

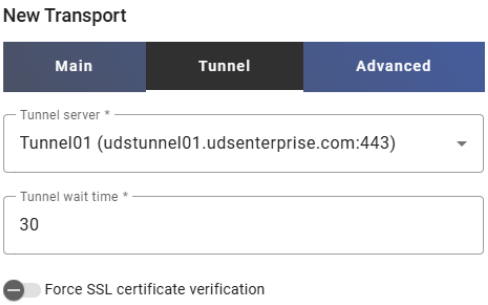


- Tunnel **(This tab will only be available in the tunneled type)**:

Tunnel Server: Select the tunnel that will make the connection. This element will be registered in "***Connectivity***" - "***Tunnels***".

Tunnel wait time: Validity time of the UDS Tunnel server ticket for the connection.

Force SSL certificate verification: Enables certificate verification. The UDS Tunnel server must have a valid certificate installed.



- Advanced:

Fullscreen Mode: Enables full screen on the desktop connection.

Smartcard Redirect: Enables smartcard redirection on the desktop.

Enable USB: Allows redirection of devices connected to a USB port.

New USB Auto Sharing: Allows the redirection of PnP devices connected to a USB port.

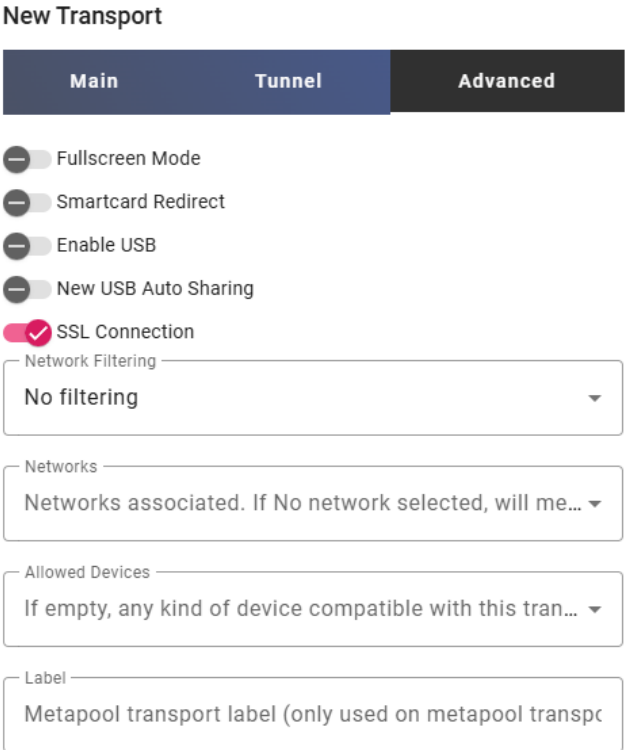
Network Filtering: We will indicate the action to be carried out on the networks indicated in the "***Networks***" section. We may not perform filtering: "***No filtering***", allow only the indicated networks: ***"Allow select networks***" or not allow only the indicated networks: "***Deny selected networks***".

Networks: Network ranges, subnets or IP addresses indicated in the "***Networks***" section of the "Connectivity" section. Used in conjunction with the "***Network Filtering***" field to allow or disable access to transport by users based on their network location.

NOTE: If none is selected, the transport will be visible on any network.

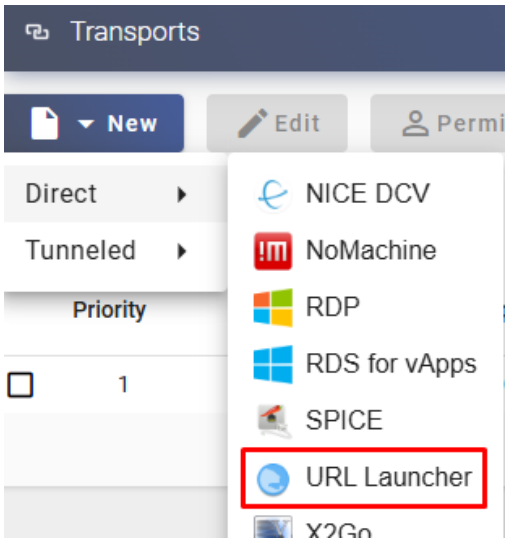
Allowed Devices: Enables access to the service only with selected devices. If none are checked, no filtering is performed.

Label: Allows you to group transports to be displayed, through the name of a label, in a meta pool. By using these tags, we can indicate that a metapool has several transports.



6.1.6 URL Launcher

A direct transport "***URL Launcher***" allows users to access a specific URL that will run in the browser (the default one) of their connecting client computer.



In a "***URL Launcher***" transport the minimum parameters to be configured are:

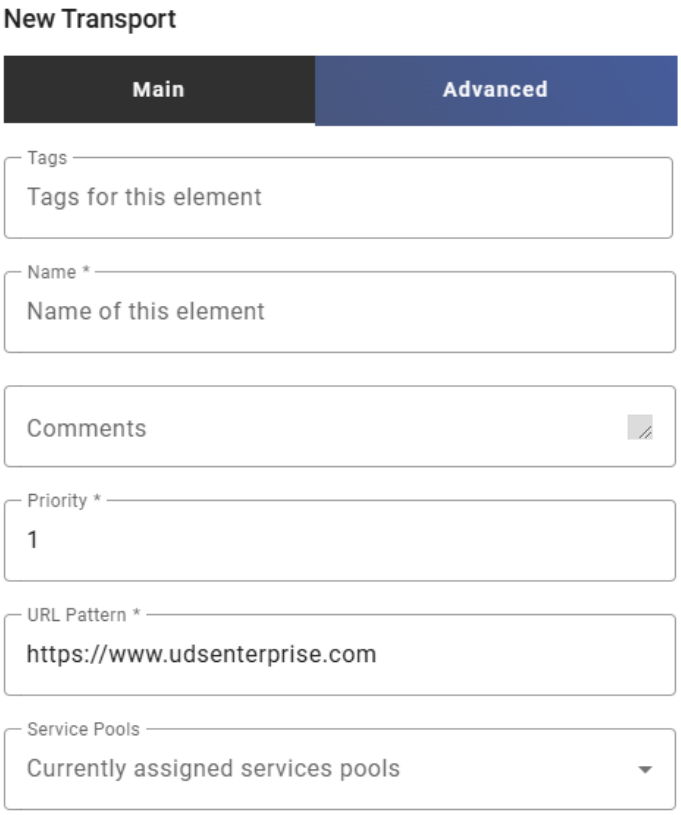
- Main:

Name: Name of the transport.

Priority: Priority that the transport will have. The lower that priority, the higher it appears in a service's list of available transports. The transport with the lowest priority will be the one used by default when you click on the image of a service.

URL Pattern: URL that will launch the browser when the service is accessed. _IP_ can be used to send the IP address specified in the "**Machine IP**" field of the base service of type "**Static Single IP**" or _USER_ can also be used to send the name of the user who started the service.

Service Pools: Allows you to assign this transport directly to one or more previously created Service Pools.



- Advanced:

Force new HTML Window: We can indicate whether the URL indicated in the "**URL Pattern" field** will be launched in a new browser tab, the next ones that are executed will be overwritten, or the current one will be reused.

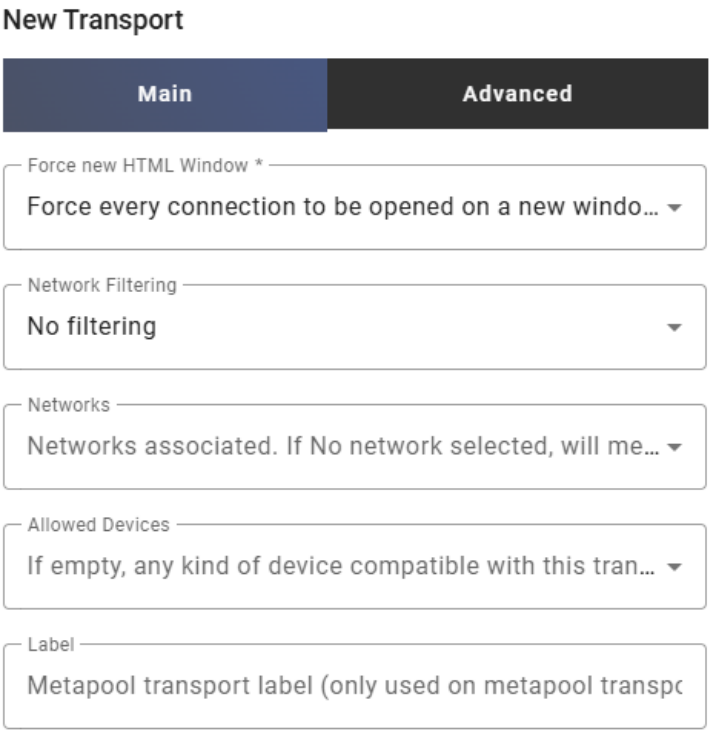
Network Filtering: We will indicate the action to be carried out on the networks indicated in the "***Networks***" section. We may not perform filtering: "***No filtering***", allow only the indicated networks: ***"Allow select networks***" or not allow only the indicated networks: "***Deny selected networks***".

Networks: Network ranges, subnets or IP addresses indicated in the "***Networks***" section of the "Connectivity" section. Used in conjunction with the "***Network Filtering***" field to allow or disable access to transport by users based on their network location.

NOTE: If none is selected, the transport will be visible on any network.

Allowed Devices: Enables access to the service only with selected devices. If none are checked, no filtering is performed.

Label: Allows you to group transports to be displayed, through the name of a label, in a meta pool. By using these tags, we can indicate that a metapool has several transports.



NOTE:

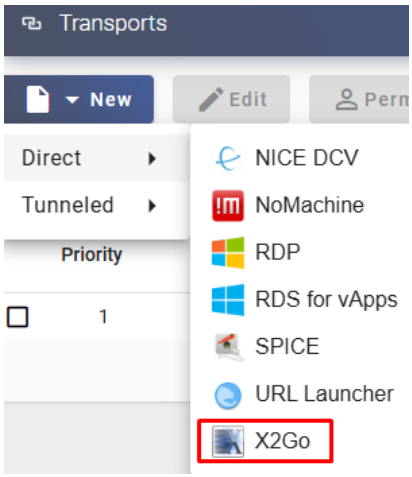
To publish a service pool and associate this transport, it is recommended to use the service provider type "**Static IP Machine Provider**" with a base service of type "**Static Single IP**".

6.1.7 X2Go

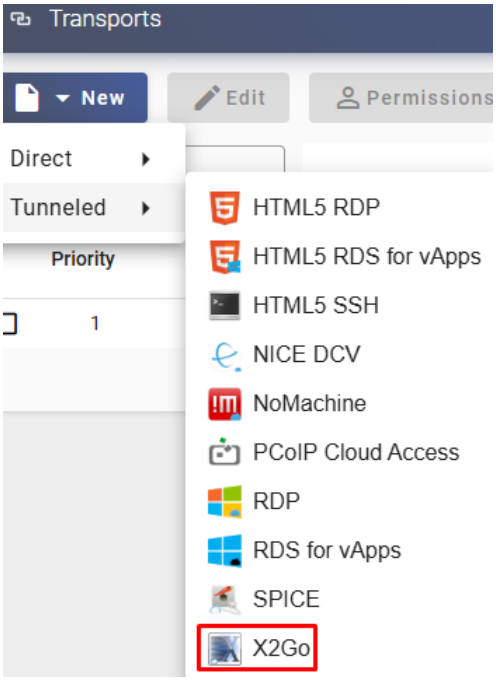
An "*X2Go*" transport allows users to access Linux desktops and applications using the "*X2Go*" software. **It is necessary that both the connection clients (client) and the vAPPs linux desktops or server (server) have "*X2Go*" installed and enabled.**

This transport can be configured directly and in a tunneled way. We will choose each type depending on the user's need for access.

Direct transport will be used when a user has IP connectivity with the desktop service, such as when they are on a LAN, there is connectivity via VPN, etc.



We will use tunneled transport when the user does NOT have IP connectivity to the desktop service, such as when they are on a WAN. This type of transport uses the UDS Tunnel server to make the connection against the desktops, having to be previously configured for its correct operation.



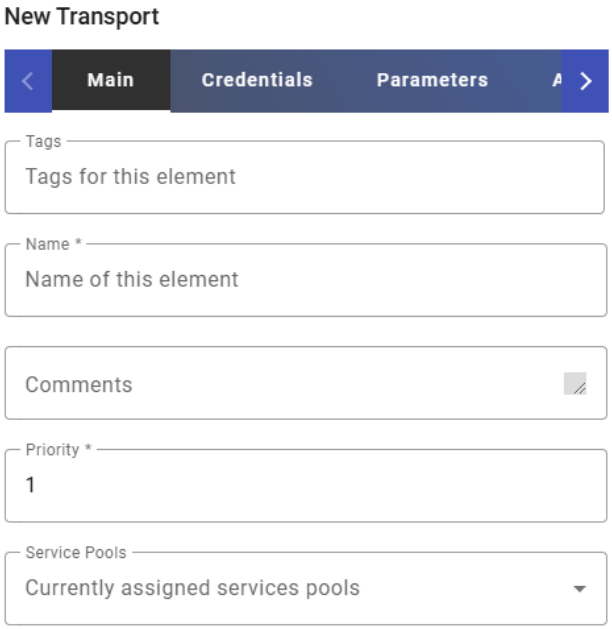
In an "*X2Go*" transport the minimum parameters to be configured are:

- Main:

Name: Nombre del transporte.

Priority: Prioridad que tendrá el transporte. Cuanto menor sea esa prioridad, más arriba aparecerá en la lista de transportes disponibles de un servicio. El transporte con la prioridad más baja será el que se utilice por defecto al pulsar sobre la imagen de un servicio.

Service Pools: Permite asignar este transporte directamente a uno o varios "*Service Pools*" creados previamente.

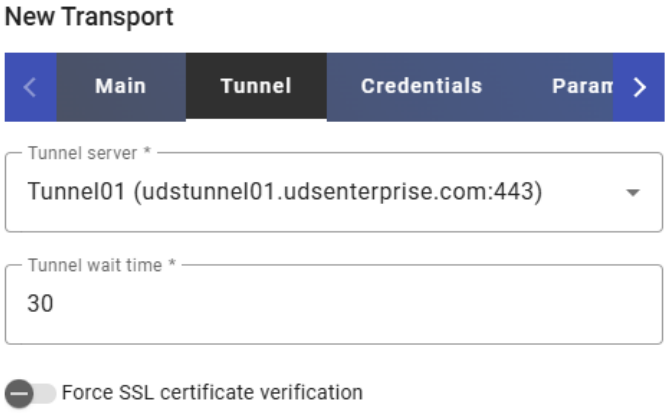


- Tunnel **(This tab will only be available in the tunneled type)**:

Tunnel Server: Select the tunnel that will make the connection. This element will be registered in "*Connectivity*" - "*Tunnels*".

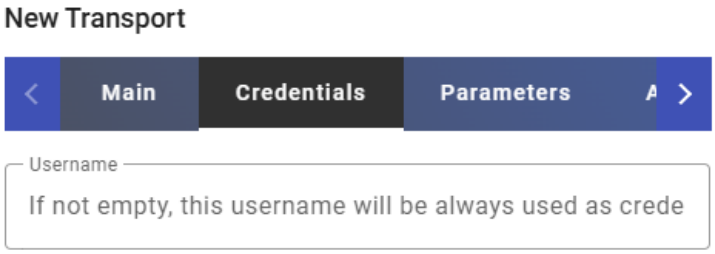
Tunnel wait time: Validity time of the UDS Tunnel server ticket for the connection.

Force SSL certificate verification: Enables certificate verification. The UDS Tunnel server must have a valid certificate installed.



- Credentials:

Username: Username that can be used to log in to the desktop.



- Parameters:

Screen size: Resolution of the connection window.

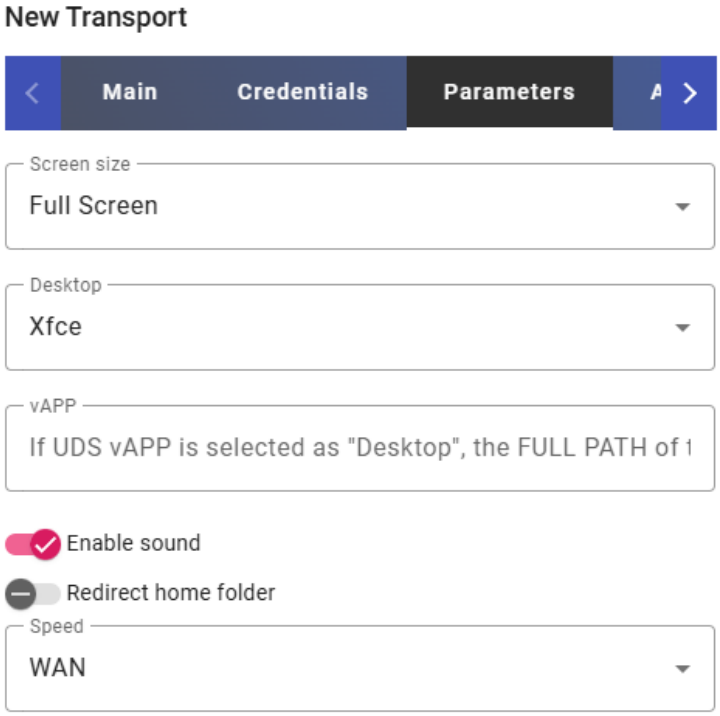
Desktop: Selection of the desktop manager (xfce, Mate, Gnome, etc...) or virtualization of Linux applications (UDS vAPP).

vAPP: Execution path of the application to be virtualized (only applies if "***UDS vAPP***" is selected in the "***Desktop" parameter***).

Enable sound: Enable sound.

Redirect home folder: Redirect the user's /home.

Speed: Connection Optimization.



- Advanced:

Sound: Selecting the type of sound server.

Keyboard: Language of the keyboard.

Pack: A method used to compress images.

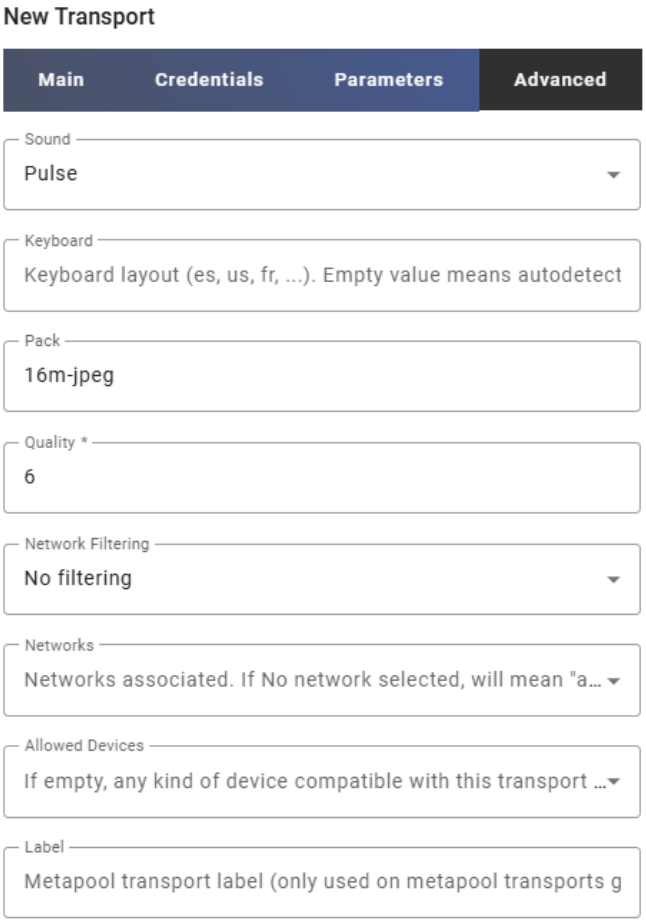
Quality: Image quality (0-9).

Network Filtering: We will indicate the action to be carried out on the networks indicated in the "***Networks***" section. We may not perform filtering: "***No filtering***", allow only the indicated networks: ***"Allow select networks***" or not allow only the indicated networks: "***Deny selected networks***".

Networks: Network ranges, subnets or IP addresses indicated in the "***Networks***" section of the "Connectivity" section. Used in conjunction with the "***Network Filtering***" field to allow or disable access to transport by users based on their network location.

NOTE: If none is selected, the transport will be visible on any network.

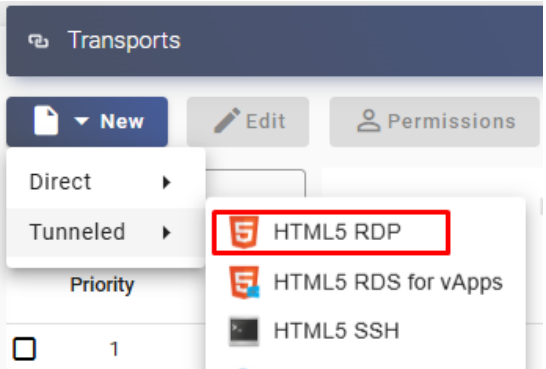
Allowed Devices: Enables access to the service only with selected devices. If none are checked, no filtering is performed.
Label: Allows you to group transports to be displayed, through the name of a label, in a meta pool. By using these tags, we can indicate that a metapool has several transports.



6.1.8 HTML5 RDP

An "***HTML5 RDP***" transport allows access to Windows and Linux desktops using the RDP protocol through a web browser that supports HTML5 (for Linux desktops it is necessary for the machines to have the XRDP package installed. For Windows desktops, RDP access must be enabled.)

This transport uses the UDS Tunnel server to make the connection against the desktops, having to be previously configured for its correct operation.



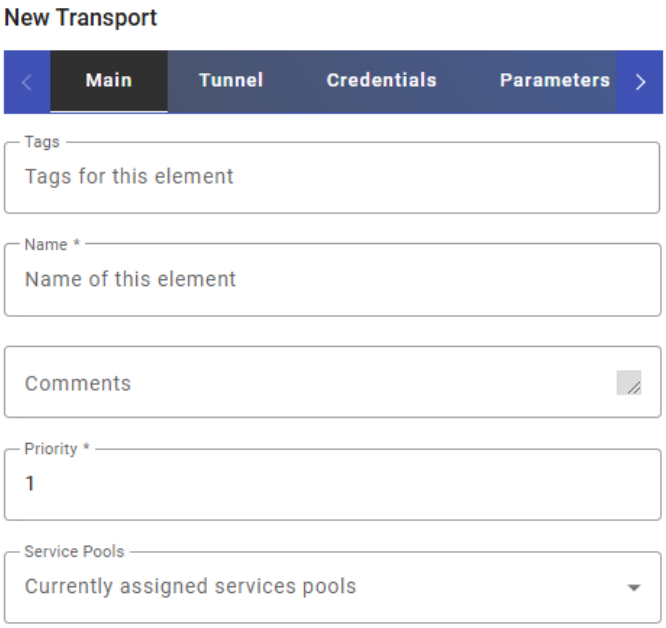
In an "***HTML5 RDP***" transport the minimum parameters to be configured are:

- Main:

Name: Name of the transport.

Priority: Priority that the transport will have. The lower that priority, the higher it appears in a service's list of available transport. Transport with the lowest priority will be the one used by default when you click on the image of a service.

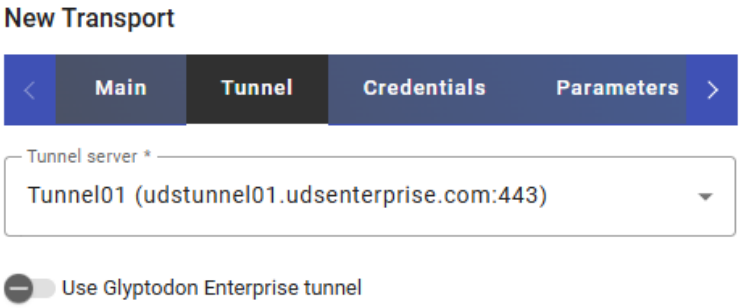
Service Pools: Allows you to assign this transport directly to one or more ***previously created*** Service Pools.



- Tunnel:

Tunnel Server: Select the tunnel that will make the connection. This element will be registered in "*Connectivity*" - "*Tunnels*".

Use Glyptodon Enterprise tunnel: Allows you to use the Glyptodon Enterprise Server tunnel for HTML5 connections



- Credentials:

Empty creds: If it is in "*Yes*", when we make the connection with the service it will ask us for the credentials to access the virtual desktop. If it is "*No*", the credentials entered in the UDS login portal will be redirected.

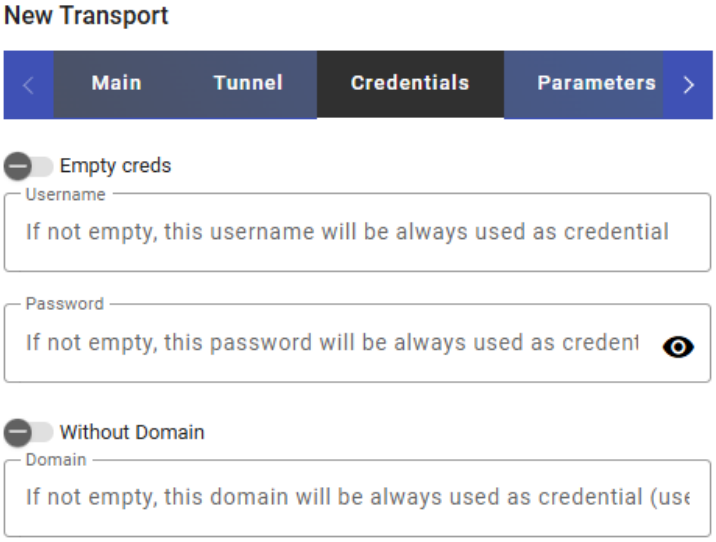
Username: Username that will be used to log in to the desktop (the user must exist on the desktop). If this field is empty, we will try to use the UDS portal login user if the "*Empty creds*" field is set to "*No*", or it will ask for credentials to indicate them manually if it is "*Yes*".

Password: User password in the "*Username" field*".

Safe use procedure: Passwords must be of sufficient length and include upper and lower case, numbers and special characters.

Without Domain: Indicates whether the domain name is redirected along with the user.

Domain: Name of the domain that will be sent with the user's credentials.



- Parameters:

Show wallpaper: Shows the wallpaper.

Allow Desk.Comp.: Habilita "***Desktop Composition***".

Font Smoothing: Enable font smoothing (improves display).

Enable Audio: Enables audio (speaker) redirection on the desktop.

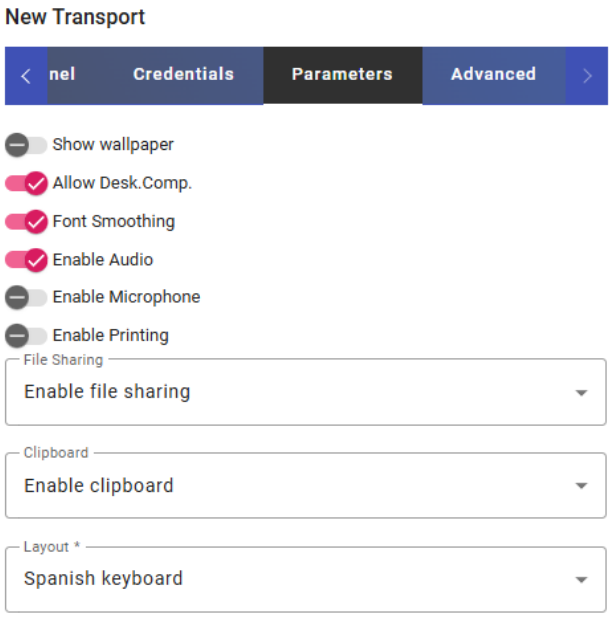
Enable Microphone: Enables audio (Microphone) redirection on the desktop.

Enable Printing: Enables desktop printing. A printer will automatically be enabled to print documents in pdf format and download them to the connecting client computer.

File Sharing: Enables a temporary repository (located on the UDS Tunnel server) to be able to upload or download files between the desktop and the connection client.

Clipboard: Enables copying and pasting text between the client connection and the desktop.

Layout: Keyboard language to be enabled on the desktop.



- Advanced:

Ticket Validity: Indicates the time (in seconds) of validity of the ticket generated by UDS in the connection client for the HTML5 connection. After this time, it will no longer be possible to use the reconnect button to recover a session in the event of a cut (it will be necessary to click on the service again to access).

Force new HTML Window: Forces every connection that starts to be in a new window. It is reused or the service portal is overwritten.

Security: Indicates the security level of the connection. The "Any" option will choose the one that is enabled on the virtual desktop to which we connect. If we want to apply NLA and it is not possible to redirect credentials, we will have to enable the "empty creds" check of the "Credentials" section".

RDP Port: Port that will verify the UDS server and the tunnel server before allowing the user to connect. If this port is not accessible to the desktop to which the connection will be made, the connection will fail.

Glyptodon Enterprise context path: In case of using the Glyptodon Enterprise tunnel and it was not in the default route, we must indicate the new path.

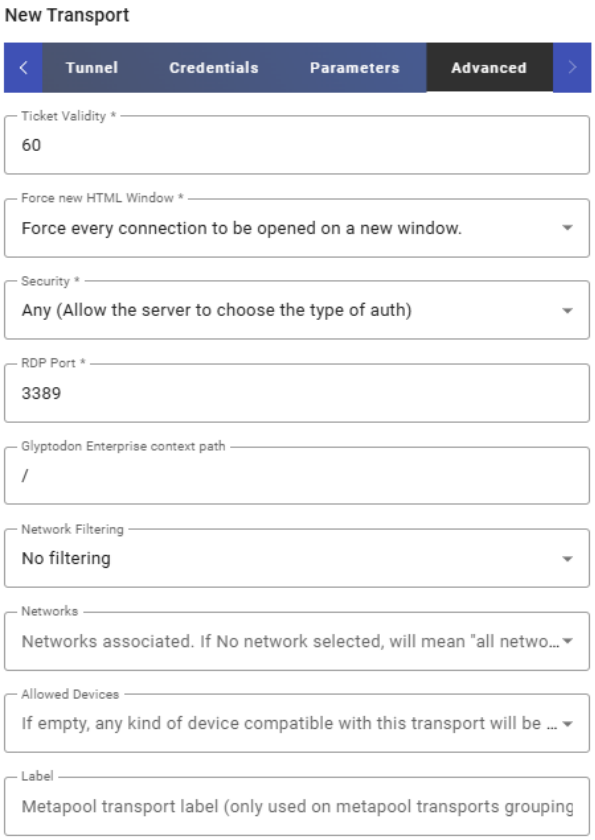
Network Filtering: We will indicate the action to be carried out on the networks indicated in the "***Networks***" section. We may not perform filtering: "***No filtering***", allow only the indicated networks: ***"Allow select networks***" or not allow only the indicated networks: "***Deny selected networks***".

Networks: Network ranges, subnets or IP addresses indicated in the "***Networks***" section of the "Connectivity" section. Used in conjunction with the "***Network Filtering***" field to allow or disable access to transport by users based on their network location.

NOTE: If none is selected, the transport will be visible on any network.

Allowed Devices: Enables access to the service only with selected devices. If none are checked, no filtering is performed.

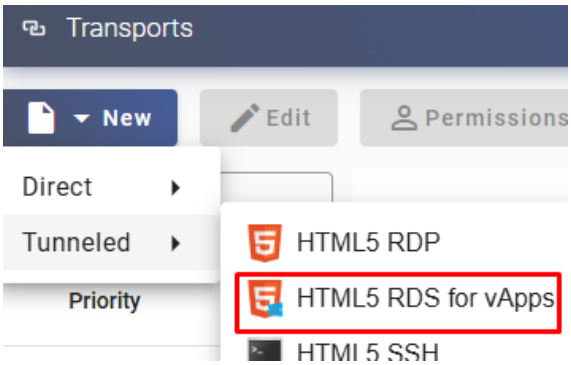
Label: Allows you to group transports to be displayed, through the name of a label, in a meta pool. By using these tags, we can indicate that a metapool has several transports.



6.1.9 HTML5 RDP For vApps

An "***HTML5 RDP for vAPPs***" transport allows access to Windows virtual applications using RemoteAPP through a web browser that supports HTML5. The connection client can have any OS that includes a browser with HTML5 support.

This transport uses the UDS Tunnel server to make the connection, having to be previously configured for its correct operation.



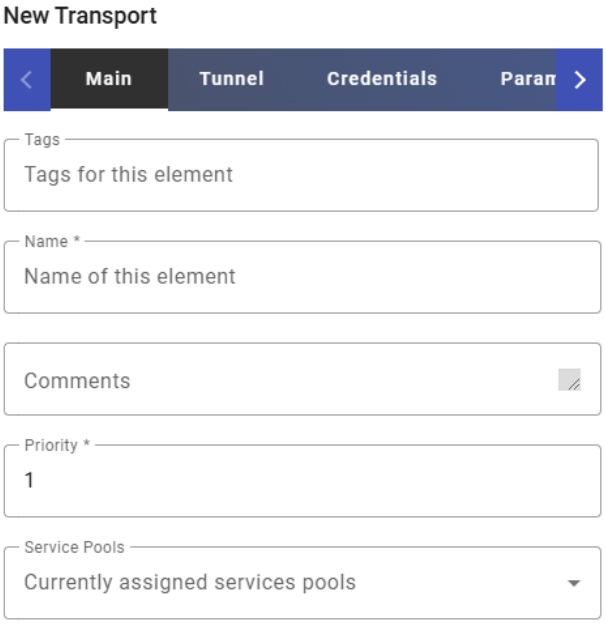
In a "***HTML5 RDP***" transport the minimum parameters to be configured are:

- Main:

Name: Name of the transport.

Priority: Priority that the transport will have. The lower that priority, the higher it appears in a service's list of available transports. The transport with the lowest priority will be the one used by default when you click on the image of a service.

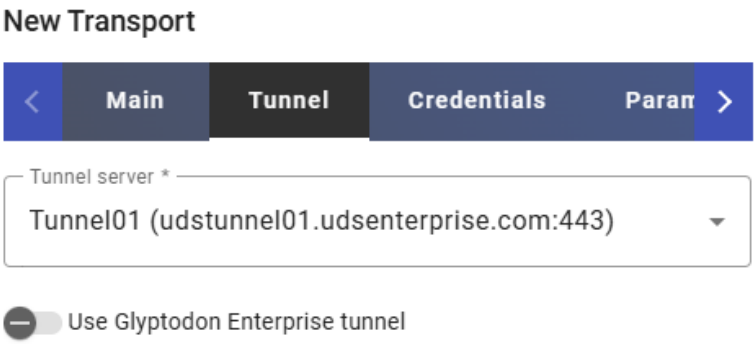
Service Pools: Allows you to assign this transport directly to one or more ***previously created*** Service Pools.



- Tunnel:

Tunnel Server: Select the tunnel that will make the connection. This element will be registered in "***Connectivity***" - "***Tunnels***".

Use Glyptodon Enterprise tunnel: Allows you to use the Glyptodon Enterprise Server tunnel for HTML5 connections.

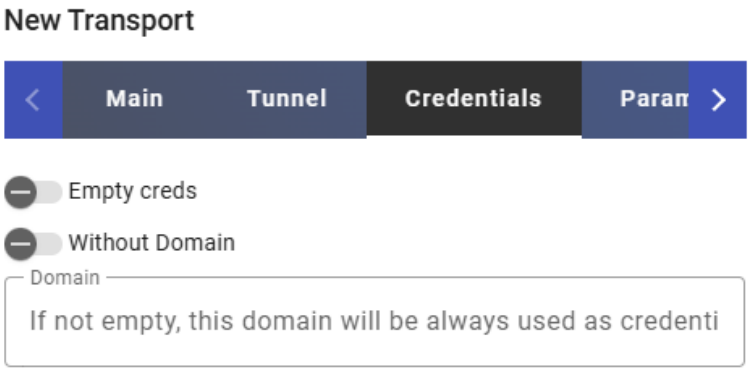


- Credentials:

Empty creds: If it is in "***Yes***", when we make the connection with the service it will ask us for the credentials. If it is in "***No***", the credentials entered in the UDS login portal or those indicated in the "mapping users" will be redirected.

Without Domain: Indicates whether the domain name is redirected along with the user.

Domain: Name of the domain that will be sent with the user's credentials (usually it is not necessary to indicate the domain since it is already defined along with the user's name).



- Parameters:

Show wallpaper: Shows the wallpaper.

Allow Desk.Comp.: Habilita "***Desktop Composition***".

Font Smoothing: Enable font smoothing (improves display).

Enable Audio: Enables audio (speaker) redirection in the app.

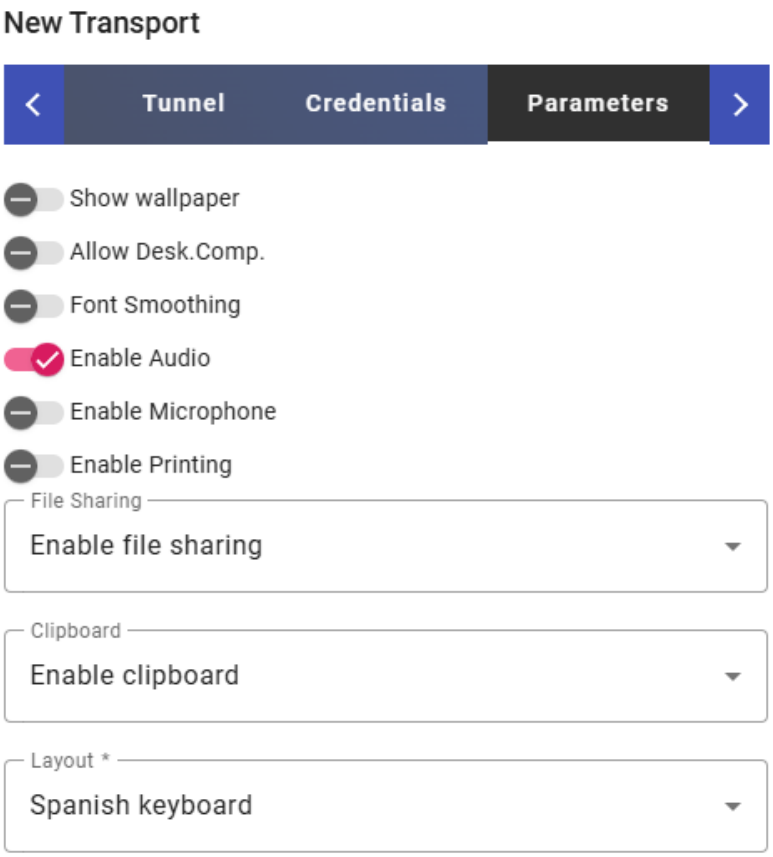
Enable Microphone: Enables audio (Microphone) redirection in the app.

Enable Printing: Enables in-app printing. A printer will be automatically enabled to print documents in pdf format and download them to the connecting client computer.

File Sharing: Enables a temporary repository (located on the UDS Tunnel server) to be able to upload or download files between the application and the connection client.

Clipboard: Enables copying and pasting text between the connection client and the application.

Layout: Keyboard language that will be enabled in the app.



- Advanced:

Ticket Validity: Indicates the time (in seconds) of validity of the ticket generated by UDS in the connection client for the HTML5 connection. After this time, it will no longer be possible to use the reconnect button to recover a session in the event of a cut (it will be necessary to click on the service again to access).

Force new HTML Window: Forces every connection that starts to be in a new window. It is reused or the service portal is overwritten.

Security: Indicates the security level of the connection. The "Any" option will choose the one that is enabled on the virtual desktop to which we connect. If we want to apply NLA and it is not possible to redirect credentials, we will have to enable the "empty creds" check of the "Credentials" section.

RDP Port: Port that will verify the UDS server and the tunnel server before allowing the user to connect. If this port is not accessible to the desktop to which the connection will be made, the connection will fail.

Glyptodon Enterprise context path: In case of using the Glyptodon Enterprise tunnel and it was not in the default route, we must indicate the new path.

Network Filtering: We will indicate the action to be carried out on the networks indicated in the "***Networks***" section. We may not perform filtering: "***No filtering***", allow only the indicated networks: ***"Allow select networks***" or not allow only the indicated networks: "***Deny selected networks***".

Networks: Network ranges, subnets or IP addresses indicated in the "***Networks***" section of the "Connectivity" section. Used in conjunction with the "***Network Filtering***" field to allow or disable access to transport by users based on their network location.

NOTE: If none is selected, the transport will be visible on any network.

Allowed Devices: Enables access to the service only with selected devices. If none are checked, no filtering is performed.

Label: Allows you to group transports to be displayed, through the name of a label, in a meta pool. By using these tags, we can indicate that a metapool has several transports.

New Transport

Credentials | Parameters | Advanced | Gateway

Ticket Validity *
60

Force new HTML Window *
Force every connection to be opened on a new window.

Security *
Any (Allow the server to choose the type of auth)

RDP Port *
3389

Glyptodon Enterprise context path
/

Load Balancing Info
Load Balancing Info. Normally, this field is blank. Check de documenta

Network Filtering
No filtering

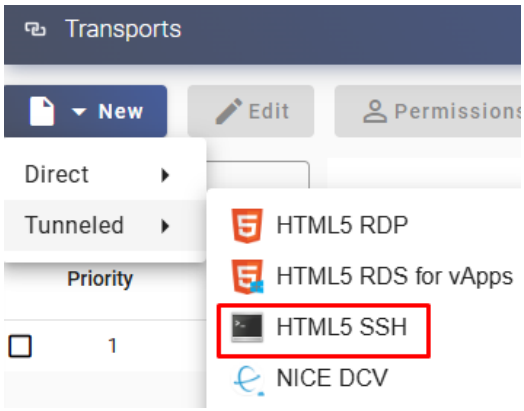
Networks
Networks associated. If No network selected, will mean "all networ...

Allowed Devices
If empty, any kind of device compatible with this transport will be al...

6.1.10 HTLM5 SSH

An "***HTML5 SSH***" transport allows access to a computer via the SSH protocol using a browser. The SSH protocol ensures secure and encrypted communication between the client and the server, using different types of cryptography.

This transport uses the UDS Tunnel server to make the connection against the desktops, having to be previously configured for its correct operation.

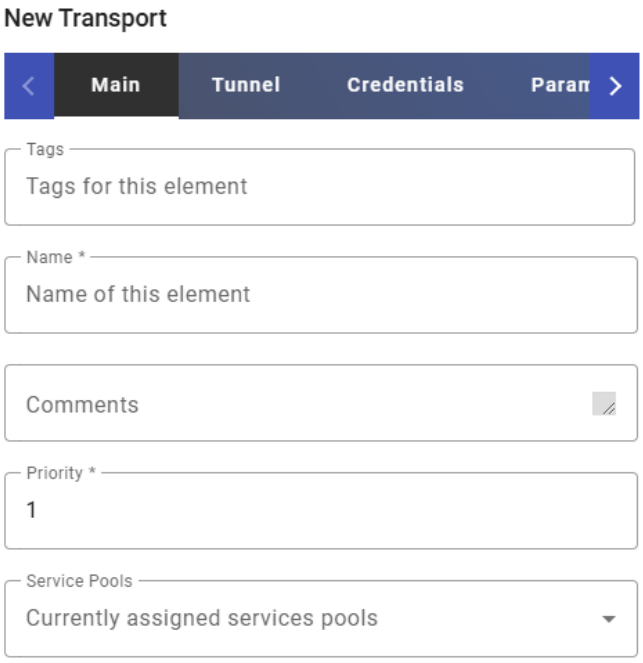


- Main:

Name: Name of the transport.

Priority: Priority that the transport will have. The lower that priority, the higher it appears in a service's list of available transports. The transport with the lowest priority will be the one used by default when you click on the image of a service.

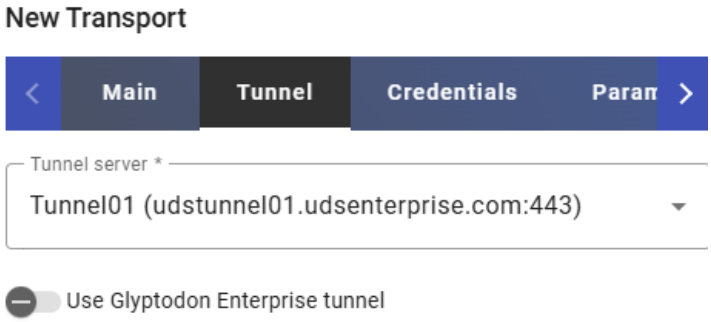
Service Pools: Allows you to assign this transport directly to one or more ***previously created*** Service Pools.



- Tunnel:

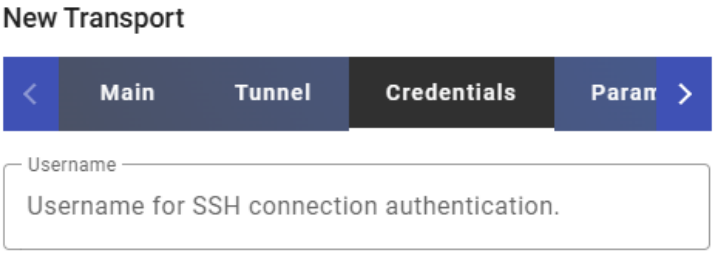
Tunnel Server: Select the tunnel that will make the connection. This element will be registered in "*Connectivity*" - "*Tunnels*".

Use Glyptodon Enterprise tunnel: Allows you to use the Glyptodon Enterprise Server tunnel for HTML5 connections.



- Credentials

Username: User for SSH connection.

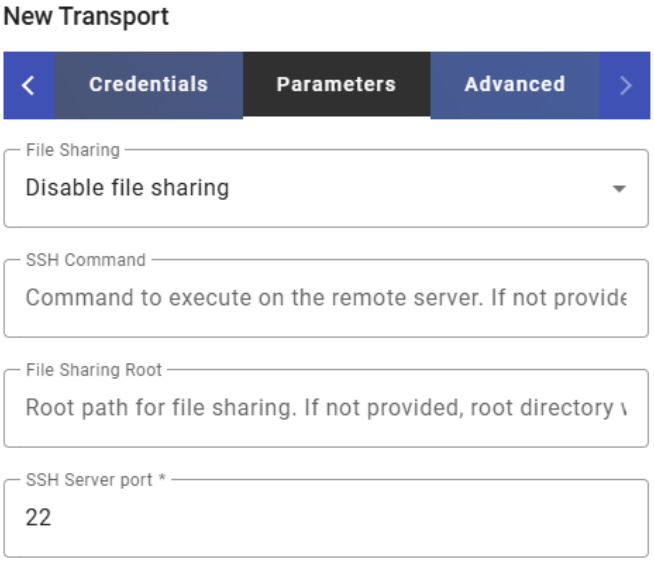


o Parameters

SSH Command: Command to run on the remote server.

File Sharing: Enable or disable file sharing.

SSH Server port: Port to use on the SSH connection



- Advanced

Ticket Validity: Indicates the time (in seconds) of validity of the ticket generated by UDS in the connection client for the HTML5 connection. After this time, it will no longer be possible to use the reconnect button to recover a session in the event of a cut (it will be necessary to click on the service again to access).

Force new HTML Window: Forces every connection that starts to be in a new window. It is reused or the service portal is overwritten.

Glyptodon Enterprise context path: In case of using the Glyptodon Enterprise tunnel and it was not in the default route, we must indicate the new path.

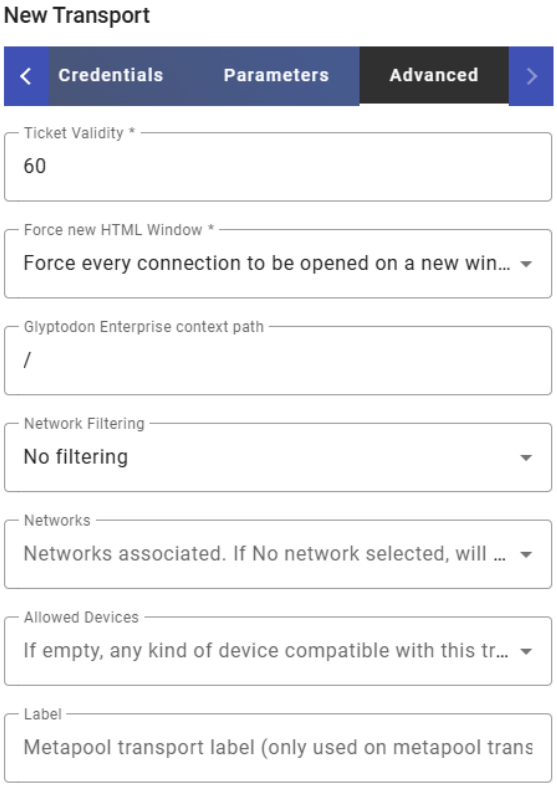
Network Filtering: We will indicate the action to be carried out on the networks indicated in the "***Networks***" section. We may not perform filtering: "***No filtering***", allow only the indicated networks: ***"Allow select networks***" or not allow only the indicated networks: "***Deny selected networks***".

Networks: Network ranges, subnets or IP addresses indicated in the "***Networks***" section of the "Connectivity" section. Used in conjunction with the "***Network Filtering***" field to allow or disable access to transport by users based on their network location.

NOTE: If none is selected, the transport will be visible on any network.

Allowed Devices: Enables access to the service only with selected devices. If none are checked, no filtering is performed.

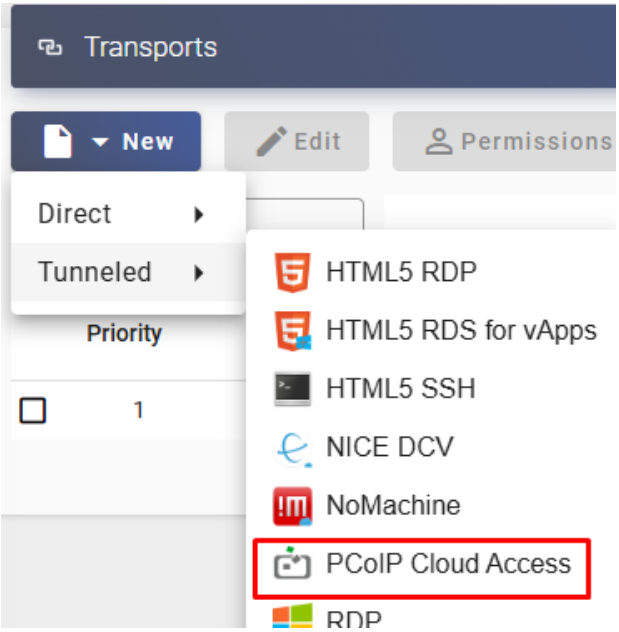
Label: Allows you to group transports to be displayed, through the name of a label, in a meta pool. By using these tags, we can indicate that a metapool has several transports.



6.1.11 PCoIP Cloud Access

A "***PCoIP Cloud Access***" transport allows access to Windows/Linux desktops by users using HP's PCoIP (HP Anyware) protocol. **Both desktops and connection clients must have the server and PCoIP client installed and enabled, respectively.**

In addition, it will be necessary to have and have previously configured the rest of the elements that make up the PCoIP Cloud Access suite (License Server, Connection Manager and Security Gateway).



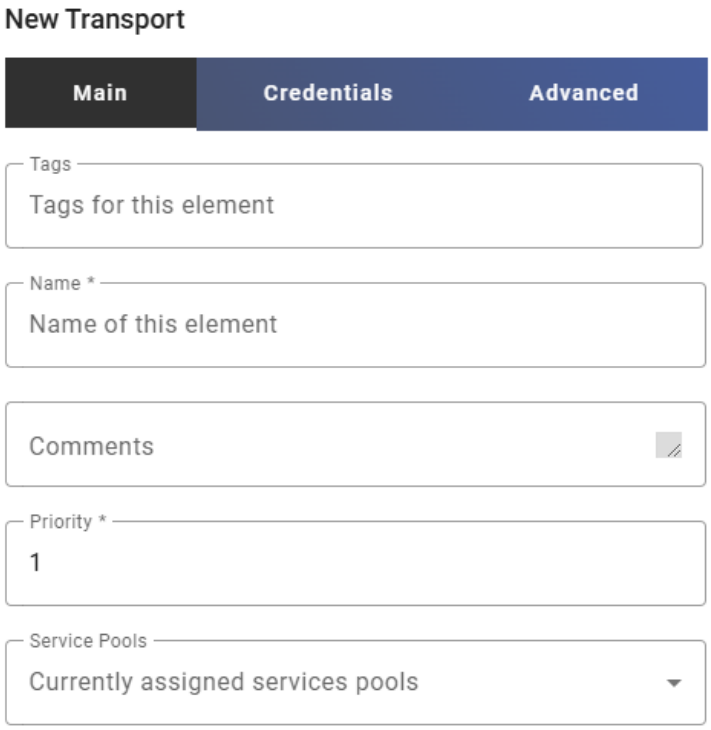
En un "***PCoIP Transport***" los parámetros mínimos a configurar son:

- Main:

Name: Name of the transport.

Priority: Priority that the transport will have. The lower that priority, the higher it appears in a service's list of available transports. The transport with the lowest priority will be the one used by default when you click on the image of a service.

Service Pools: Allows you to assign this transport directly to one or more ***previously created*** Service Pools.



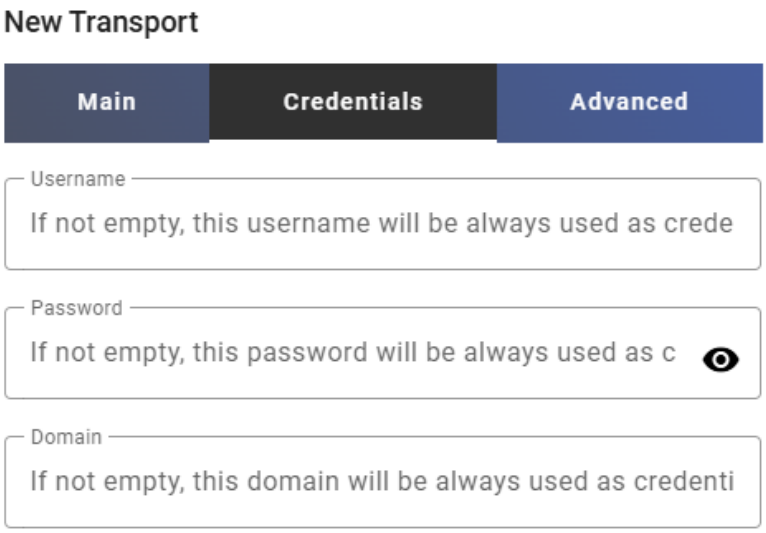
- Credentials:

Username: Username that will be used to log in to the desktop. If this field is empty, the user's name will be requested when making the connection.

Password: User password in the "***Username" field***".

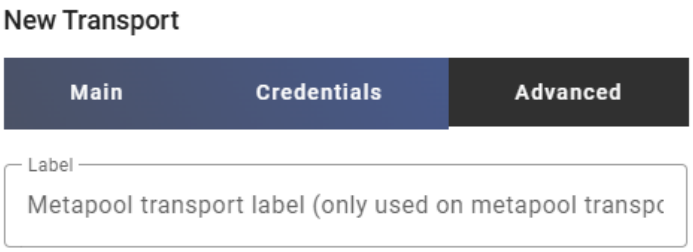
Safe use procedure: Passwords must be of sufficient length and include upper and lower case, numbers and special characters.

Domain: Name of the domain that will be sent with the user's credentials.



- Advanced:

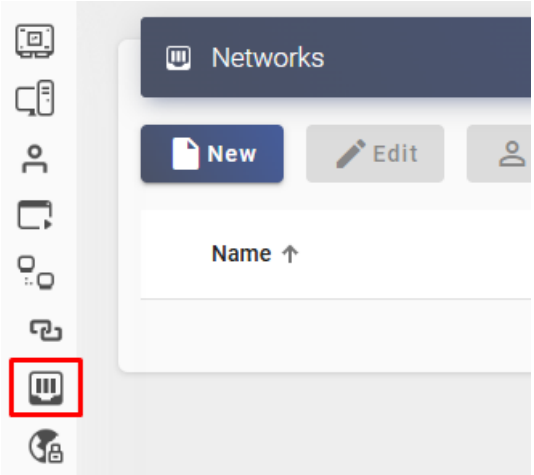
Label: It allows you to group transports to be displayed, through the name of a tag, in a meta pool. By using these tags, we can indicate that a metapool has several transports.



6.2 Networks

UDS allows you to register different networks that will be used by different elements of the administration such as Transport, Authenticators, MFAs, etc... to perform advanced filtering depending on the access network of the users that are validated in the UDS login portal.

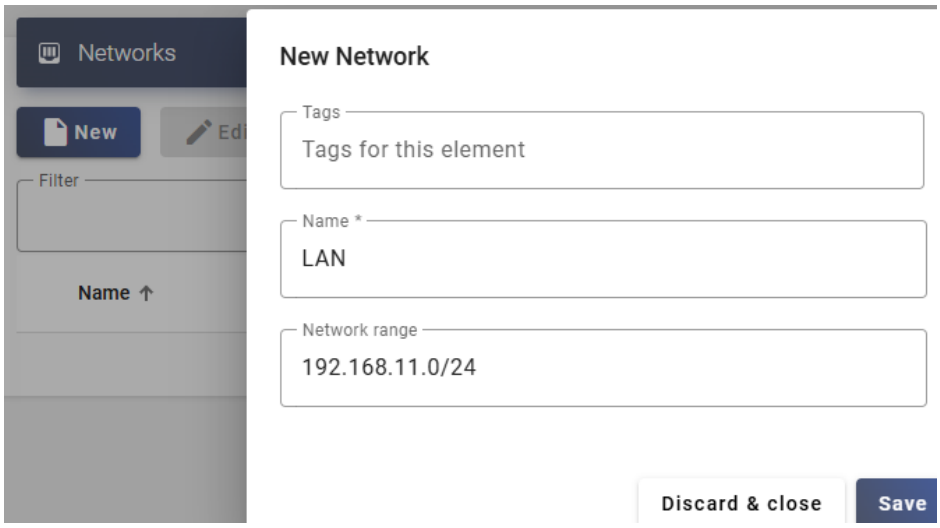
To add a network, we will hover over the "***Connectivity***" section and select "***Networks***".



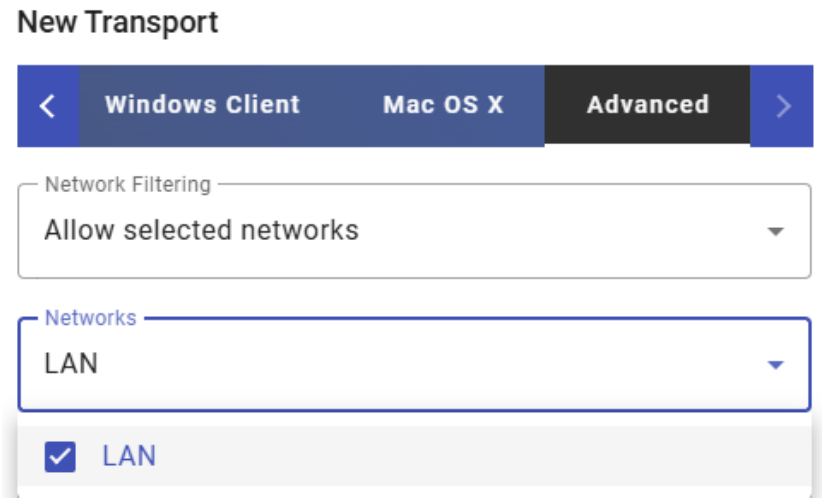
Press "***New***" and enter a descriptive name and a network range, complete subnet or IP address.

The following formats are supported:

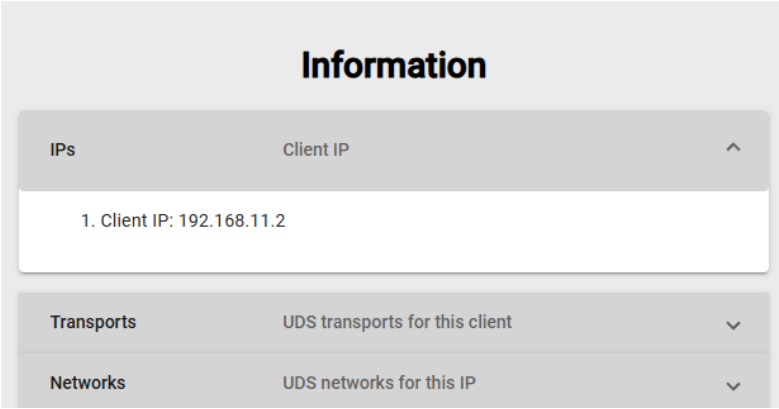
- Unique IP address: xxx.xxx.xxx.xxx (For example: 192.168.11.33).
- Full Subnet: xxx.xxx.xxx.xxx/MASK (For example: 192.168.11.0/24).
- Address Range IP: xxx.xxx.xxx.xxx-xxx.xxx.xxx.xxx (For example: 192.168.11.1-192.168.11.155).



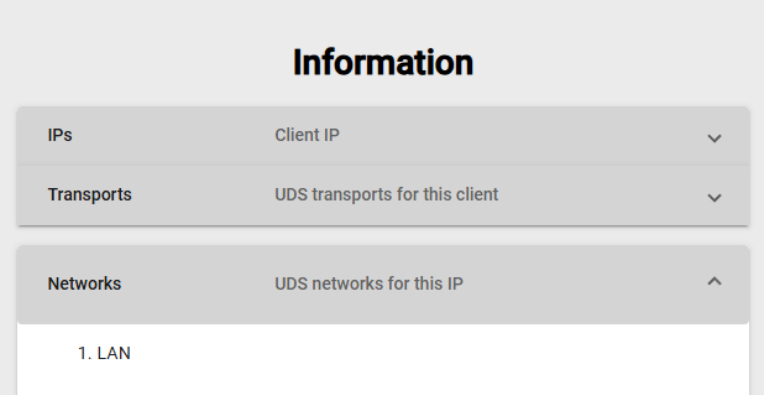
These defined networks will appear available in transports, authenticators, MFA systems, etc. and we can configure whether or not a connection client that is on that network displays the:



On a user's services screen ("***user mode***"), if the user has administration permissions over UDS, the IP address detected by UDS in the connection client can be displayed:



In addition to the networks to which it belongs:



Safe Employment Procedure:

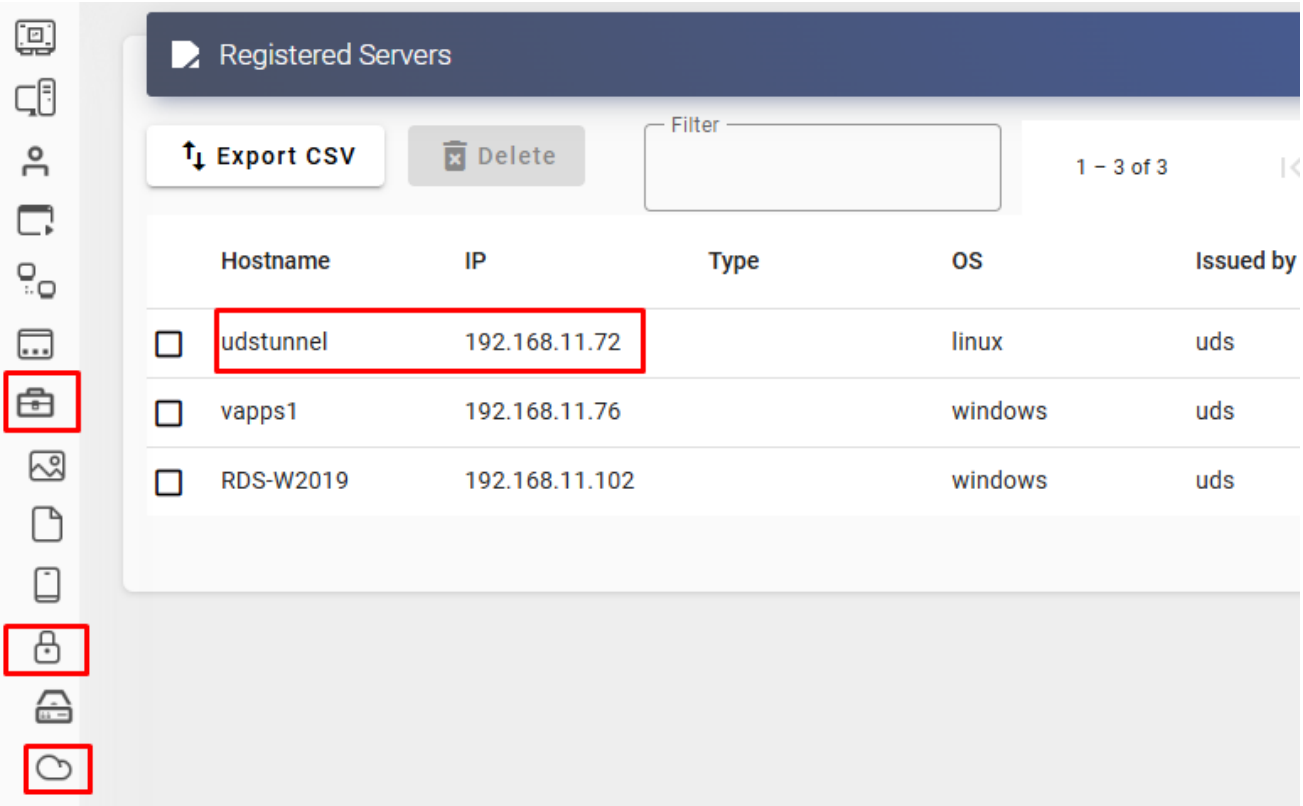
Limit access to services to only known networks.

- If you need to provide access to certain services from the internet:
- If the connections are site-to-site, limit the public IPs of your locations.
- If you can narrow down the geographic locations from where connections can be made through your subnets, do so.
- If you can avoid using insecure protocols from the internet, do so.
- If you can prevent access from subnets that you may find insecure for whatever reason, do so.

6.3 Tunnels

A tunnel will allow us to define tunnel servers previously configured and registered with the UDS server, which can be used in the transports. These tunnel servers will allow a transport to make a tunneled connection of a protocol or an HTML5 connection.

In order for a tunnel server to be used, it must be configured and registered on a UDS server. Once registered, we can view it in the "***Tools***" – "***Tokens***" section and within "***Servers***" it will be registered:

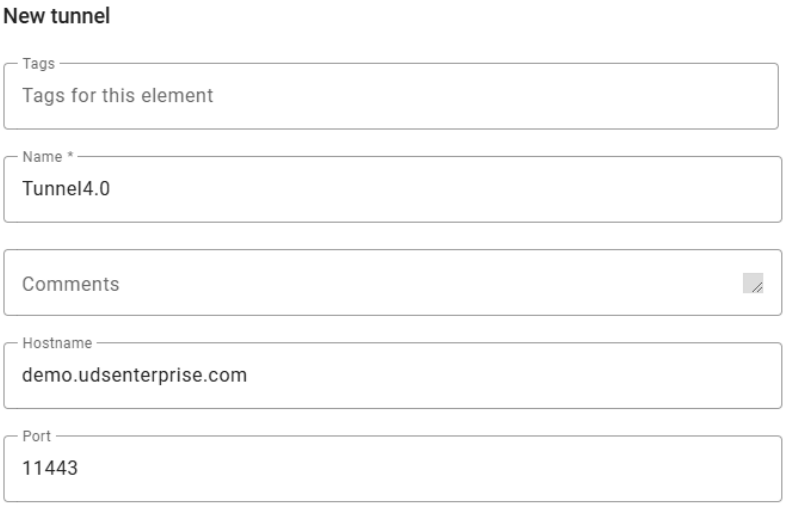


To register a Tunnel server, click on "***New***" and enter the following data:

Name: Name that we will give to the element and that will be visible for selection within a transport.

Hostname: IP address or name of the Tunnel server to be used (if there are several tunnel servers and they are behind a balancer, it will be necessary to indicate the IP or name of the balancer's balancer)

Port: Port that will be used on the tunnel server to connect to it. By default, the tunnel server listens on two ports for the two types of services it offers: 443 and 10443 (although in this section it is possible to indicate a different port and then redirect to the internal IP of the tunnel)

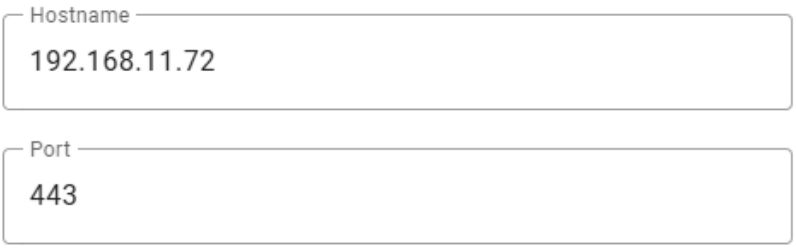


NOTE: In the image, port 11443 is used with the public name since port 443 is being used by the UDS server. The firewall has a policy that everything that enters the 11443 forwards it to the 443 of the tunnel's local IP.

The IP or server name (hostname field) or port (port field), may vary depending on how the service is offered, being able to use the default IPs or internal names and ports (443 and 10443) or the IPs or public names with ports different from the default ones, but then there is a redirection to the default listening ports of the Tunnel server.

For the two types of connections offered by the tunnel server, we would have:

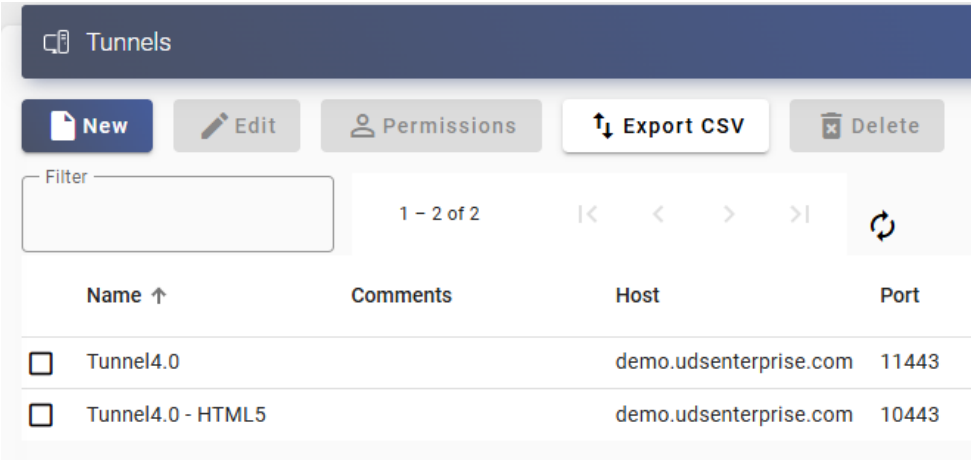
- **Tunneled connections:** By default the tunnel listens on port 443 (**it is NOT HTTP traffic**, it is SSL traffic with a custom header, therefore, if elements such as firewalls, balancers, etc. are used, do not use https templates for this service). It is possible to use any other port and then perform a redirection to port 443 of the tunnel, for example, when a single public IP address is used for the entire service (Server and Tunnel) and port 443 is already being used by the UDS Server for the login portal, in this case we would indicate port 11443 with the public IP and internally a redirection of all incoming traffic through 11443 to 443 of the tunnel's local IP).



- **HTML5 connections**: By default, the tunnel listens on port 10443. As in the case of tunneled connections, you can also indicate a different port and then redirect to the default listening port of the local IP of the Tunnel server, although in this case as the default port, 10443 is a port that is not usually busy, you can avoid being redirected.

NOTE: Connection clients must be able to reach the IP/name and port indicated in this element in order for the connection to be established.

We save and we will already have a tunnel server available to use in transports. We will be able to register all the tunnel servers that are necessary:



NOTE: Although we can enter each of the tunnel servers created and assign it a registered server, this function is not yet operational and is not necessary. In future versions of UDS it will apply.

7. POOLS

7.1 Service pools

The creation of a "***Service Pool***" will allow the deployment of desktop services or virtual applications, which will be available for access by different groups of users.

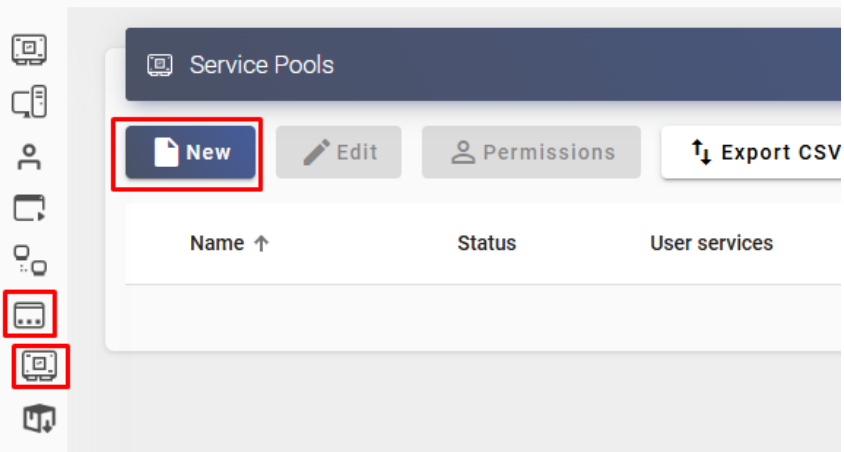
The elements needed to create a "***Service Pools***" will be a "***Base Service***" (composed of a "***Service Providers***" + a service created in it) and an "***OS Manager***". Once created, we will have to assign one or more groups of users and one or more transports to enable user access.

Some types of ***service pools*** may not require an ***OS Manager***".

Safe Employment Procedure:

Limit access to the "Service Pool" to the necessary users/groups, through the necessary protocols, from the necessary networks and, although we will see it later, at the necessary times. In a nutshell, narrow down as much as possible who, how, and when you can access a service provided by a Service Pool"

To create a "***Service Pool***" we will go to the "***Pools***" section and select "***Service Pool***".



To configure a "***Service Pool***" it will be necessary to indicate:

- Main:

Name: Name of the "***Service Pool***" (this name will be the one that is shown to a user to access your service).

In this section, variables can be added to be able to display information about the services:

- **{use}**: Indicates the % of pool usage (calculated based on the maximum services to be provided and the assigned services).
- **{total}**: Total number of computers (data extracted from the maximum services to be provided from the service pool)).

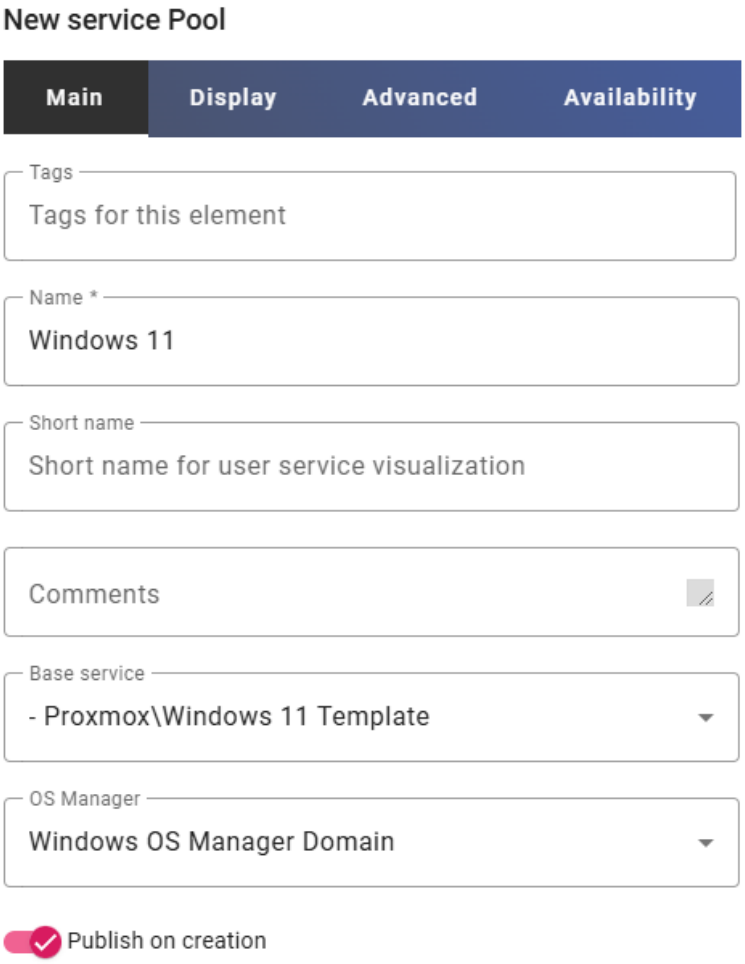
- **{usec}**: Number of computers used by users in a service pool.
- **{left}**: Number of computers available in the pool for user connection.

Short name: If indicated, it will be the name of the service that will be displayed to the user. When you hover over it, the contents of the "Name" field will appear".

Base Service: Base service to be used (desktop or virtual application). It is composed of a service provider and a base service previously configured in the "***Services***" section.

OS Manager: "***OS Manager***" previously created whose configuration will be applied to each of the generated virtual desktops. In case of publishing a vAPP service it will also be required, but in other services such as connection to fixed machines or RDS sessions, this field will not be used.

Publish on creation: If it is enabled and virtual desktop services are deployed, when we save the service pool the system will automatically launch the first publication. If it is in "***No***", it will be necessary to launch the publication of the service manually (from the "***Publications***" tab)).



- Display:

Visible: If disabled, the "***Service Pool***" will not be shown as available to users on the UDS services page ("***User mode***").

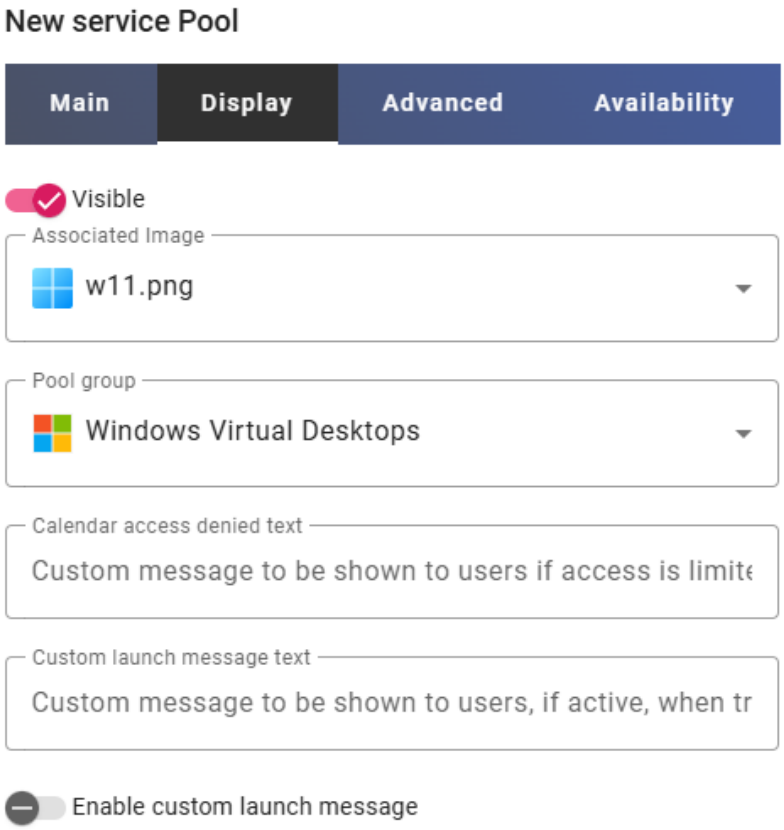
Associated Image: Image associated with the service. It must first be added to the image repository, accessible from the "***Tools***" – "***Gallery***" section.

Pool group: Allows you to group different services. In order to assign a "***Pool group***", it must be previously created in the "***Pools***" – "***Groups***" section.

Calendar Access denied text: Text that will be displayed when a service has access denied by the application of an access calendar.

Custom launch massage text: If the "***Enable custom launch message***" check is enabled, this message will be displayed when a user accesses this pool of services.

Enable custom launch message: Enables the appearance of a customizable message, which if accepted will launch the service or if not its execution will be canceled.



- Advanced:

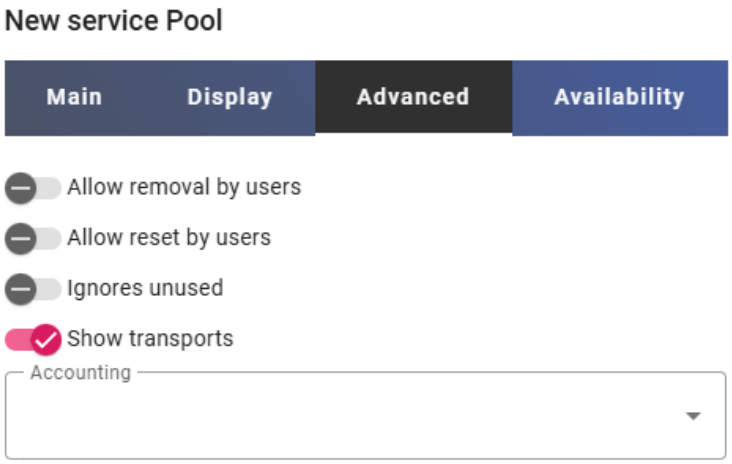
Allow removal by users: If enabled, users will be able to delete the services assigned to them. If the service is an auto-generated virtual desktop by UDS, it will be deleted and a new one will be assigned on the next connection. If it is another type of service (vAPP / Static IP), only the assignment will be removed and a new one will be assigned on the next connection.

Allow reset by users: If it is activated, the user will be able to restart or reset the services assigned to them (only applies to virtual desktops autogenerated by UDS).

Ignores unused: If enabled, non-persistent user services that are not in use will not be deleted. Recommended option if there are problems detecting machine usage by the actor.

Show transports: With this option activated, all transports assigned to the service will be displayed. If it is not enabled, only the default transport with the highest priority (lowest number in the "***priority***" field of a transport) will be displayed.

Accounting: Assignment of a service to a previously created "***Accounts***" ("***Pools***" – "***Accounts***"). Allowing you to monitor the time of use of a service.



- Availability:

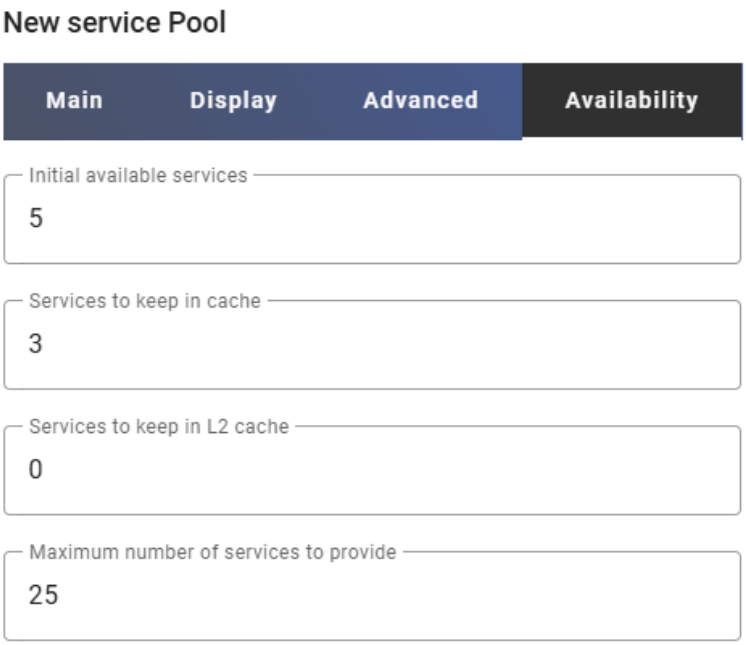
Initial available services: Minimum number of virtual desktops that will exist in the service pool, these can be assigned to users or cached (L2 cached desktops are not counted). If the system detects that this number of desktops does not exist, it will automatically create them.

Services to keep in cache: Number of virtual desktops always available. These will be configured and ready to be assigned to a user (they will be automatically generated until the maximum number of machines indicated in the "***Maximum number of services to provide***" field is reached).

Services to keep in L2 cache: Number of virtual desktops in a powered off state. These desktops will be configured and ready for assignment when the system demands new cached desktops.

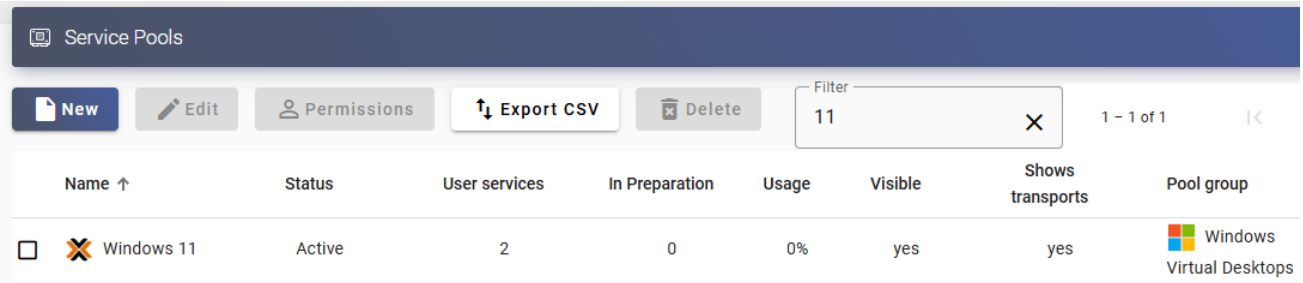
Virtual desktops generated at the L2 cache level will be cached at the time the system demands them. They will never be assigned directly to users.

Maximum number of services to provide: Maximum number of virtual desktops created by the system in the "***Service Pool***" (L2 cached desktops will not be counted towards this parameter).

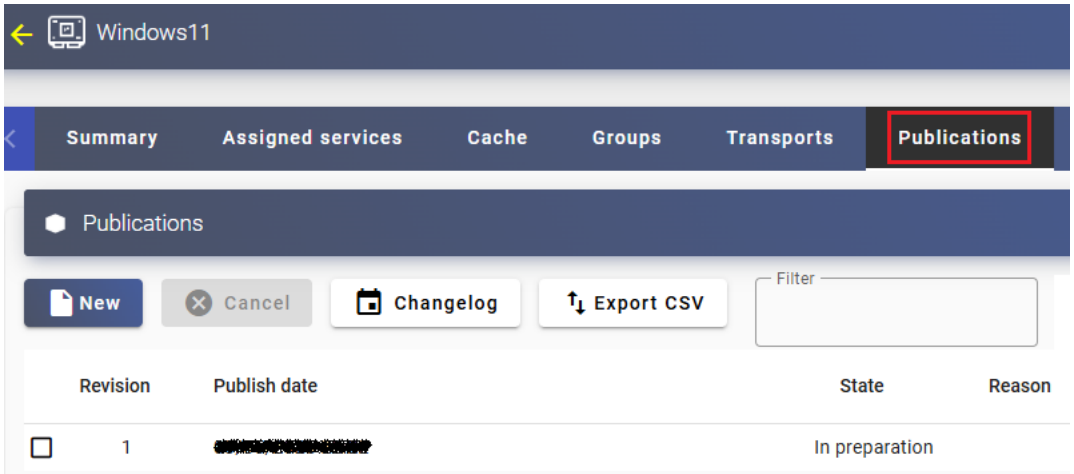


Save the "***Service Pool***" and if we have checked the "***publish on creation***" check, the system will start generating virtual desktops based on the configured cache (Availabiliy tab).

By clicking on the ***"Delete***" button, we can completely delete a Service ***Pool***, and by "***Editing***" we can modify it.

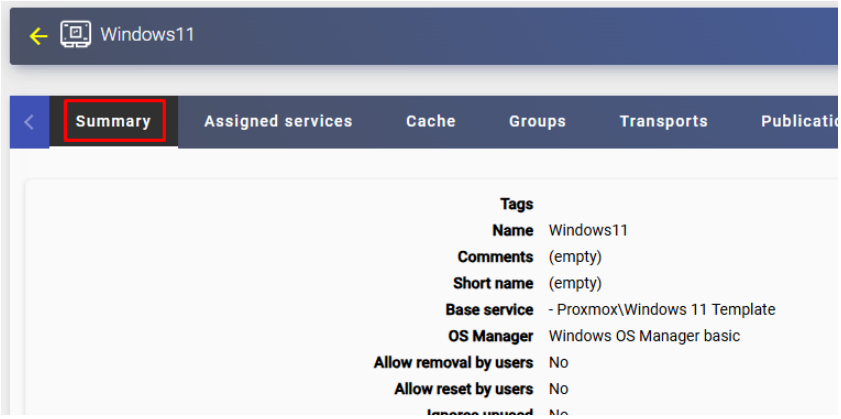


If we access the "***Service Pool***" created, in the "***publications***" section (if we have checked the "***Publish on creation***" option), the system will start with the publication of the service generating the base machine (publication) on which the virtual desktops will be based. As soon as the status is valid, the creation of the virtual desktops will begin.

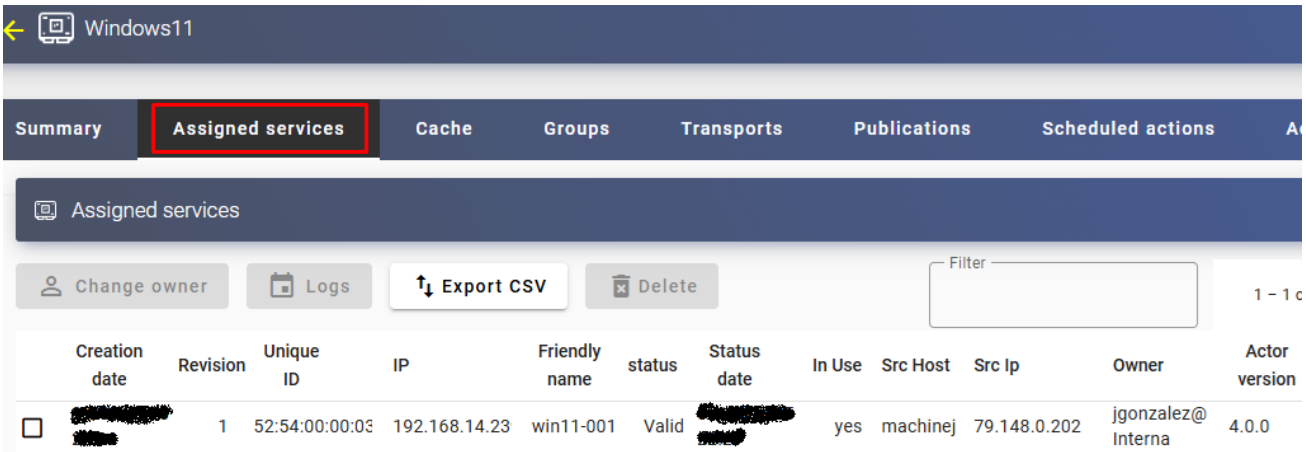


Once we have created a "***Service Pool***", we will access it and we will have the following control and configuration menus:

- **Summary:** Summary of all settings applied to the service pool.



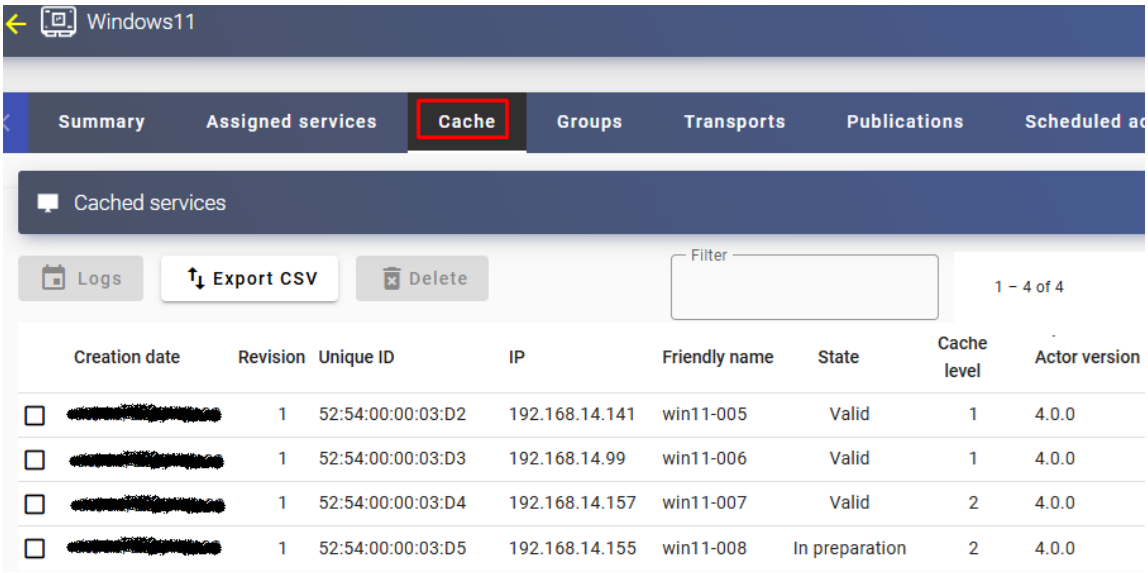
- **Assigned Services:** Virtual desktops that have been assigned to users. Displays information about the desktop creation date, the revision (or publish) number on which the desktop is generated, the MAC address of the VM network card, the DNS and IP name of the virtual desktop, the status of the desktop, if in use, the name and IP of the connecting client, the machine owner and the version of the UDS Actor installed on the machine template.



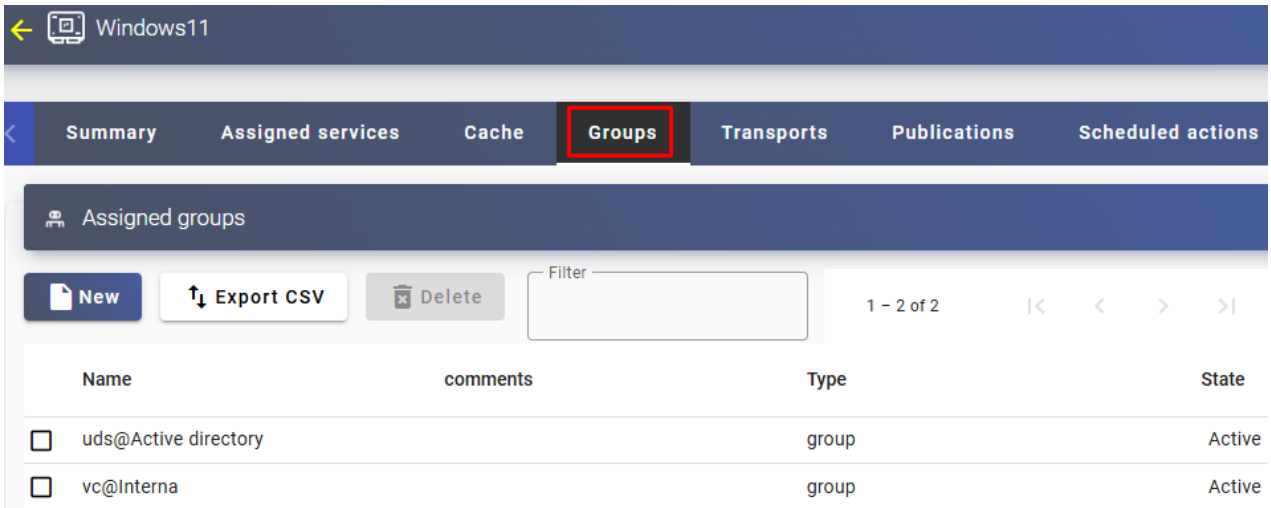
By marking the virtual desktop and clicking on "***Change owner***", we can change the user assigned to the desktop.

By clicking on "***Delete***" we can delete it manually and on "***Logs***" we will see all the information reported by the "***UDS Actor***" installed on the desktop.

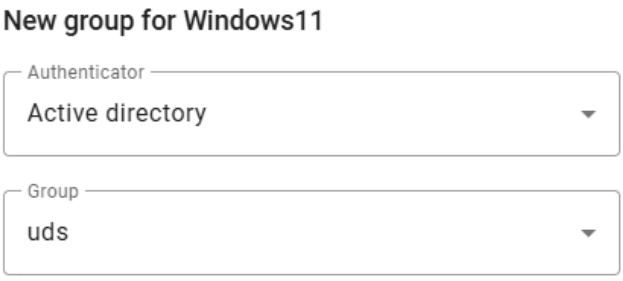
- **Cache**: Virtual desktops available for connecting users, (including L2 cache machines). These desktops will go through different states:
  - **In Preparation**: In this state, virtual desktops are being built on the virtualization platform.
  - **Waiting OS:** In this state, the virtual desktops are being configured with the parameters indicated in the "***OS Manager***" (rename, inclusion in the domain, etc…)
  - **Valid:** When a virtual desktop is in this state, it means that it is available for a user to access.



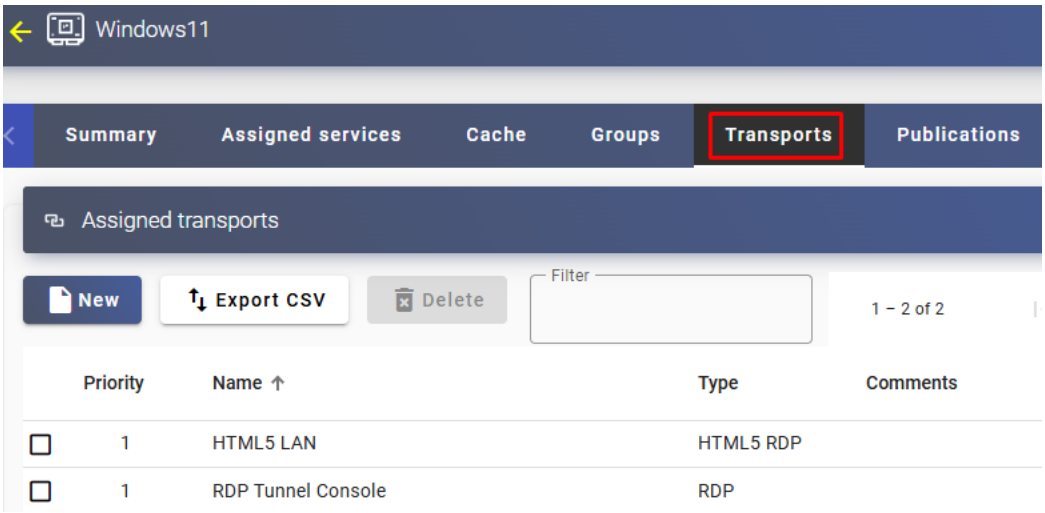
- Groups: To allow users to connect, it is necessary to assign access groups or metagroups. These groups or metagroups must be created in the "Authenticators" section and we can assign one or more access groups or metagroups to each "Service Pool"".



We select the "Authenticator" and based on your choice we choose the "Group Name"".



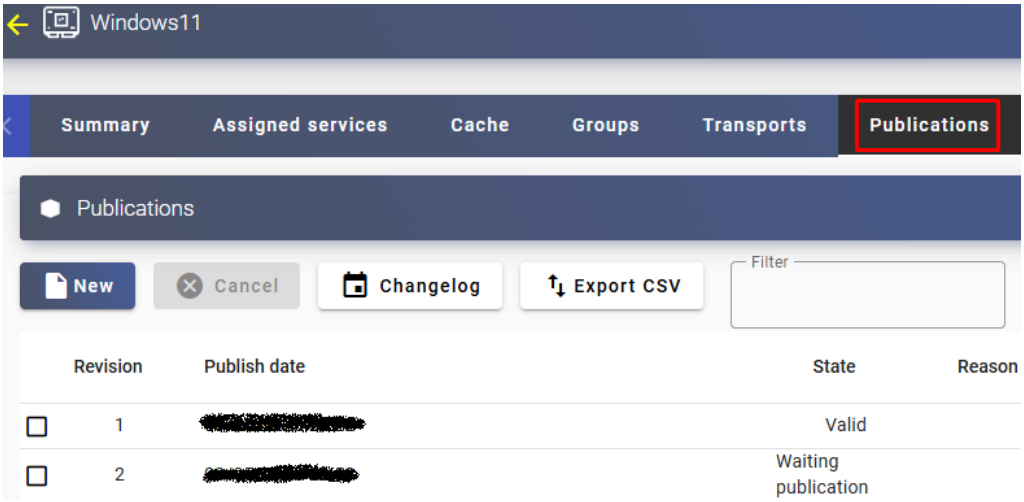
- **Transports:** The "Transports" to make the connection with the virtual desktop (previously added in the "Transports" section) will be indicated. The "Transport" with the lowest priority will be the one that the system configures by default. To use the rest of the transports, the user will have to open the drop-down menu on the virtual desktop access screen (drop-down menu) and select the one that corresponds.



Select the "Transport" we want to use in this "Service Pool" and save.

- **Publications:** From this menu we can make a new publication of the service (for example, if we have updated our base machine with new applications or OS patches and we want all our virtual desktops to take these changes). After the publishing process is complete, the entire system cache will be regenerated with the new desktops based on this latest release.

If we make a new publication, a new base machine will be generated and, once available, the system will proceed to eliminate the virtual desktops of the previous version and generate new ones based on the new publication.

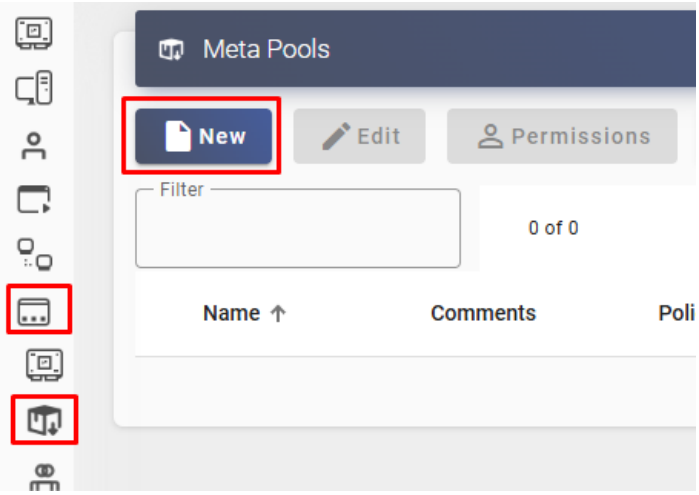


7.2 Meta Pools

The creation of a "***Meta Pool***" will allow access to desktop services or vApps composed of different "***Services Pools***". These Pools will work together providing different services in a completely transparent way for users.

The Service ***Pools*** that form a Meta ***Pool*** will work under a policy that will allow services to be provisioned according to the needs of the Pool. Currently, the supported policies will be defined by priorities, platform capacity, and platform usage.

To create a "***Meta Pool***" we will go to the "***Pools***" section and select "***Meta Pools***".



To set up a "***Meta Pool***" it will be necessary to indicate:

- Main:

Name: Name of the "***Meta Pool***" (this name will be the one that is shown to a user to access its service: desktop or virtual application).

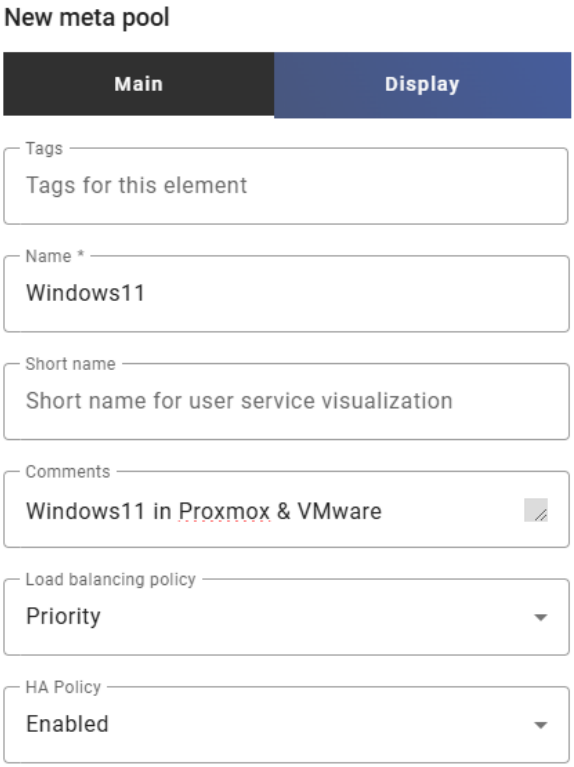
Short name: If indicated, it will be the name of the service that will be shown to the user. When you hover over it, the contents of the "Name" field will appear".

Load balancing Policy: Policy that will be applied when assigning the services of the ***"Services Pools***" that are part of the "***Meta Pool"***".

- **Evenly distributed**: Services will be consumed equally in all the Services ***Pools*** that make up the Meta ***Pool***".
- **Priority**: The services will be consumed from the "***Service Pool***" that has the highest priority (priority is defined by the "***priority***" field). The lower the value of this field, the higher priority the item has.) When the Service ***Pool*** reaches the maximum number of services, the following services will be consumed.

- **Greater % available**: The services will be created and consumed from the "***Service Pool***" that has the highest percentage of free use.

HA Policy: If enabled, the service pool will be checked before assigning services to the user, if it is not available, it will move to the next.



- Display:

Associated Image: Image associated with the "***Meta Pool***". It must first be added to the image repository, accessible from the "***Tools***" – "***Gallery***" section.

Pool group: Enables the ability to group different "***Meta Pools***" to be able to assign a "***Pool group***". It must be previously created in the "***Pools***" – "***Groups***" section.

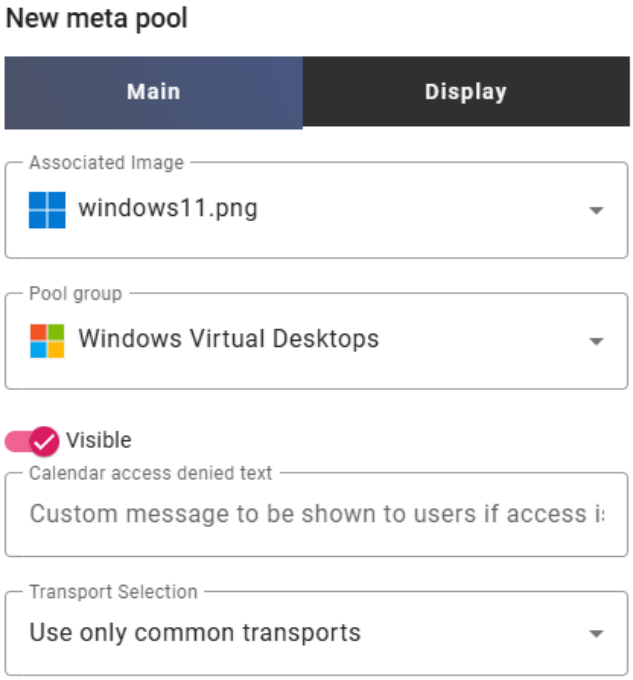
Visible: If disabled, the "***Meta Pool***" will not be shown as available to users on the UDS services page ("***User mode***").

Calendar Access denied text: Text that will be displayed when the "***Meta Pool***" has access denied by the application of an access calendar.

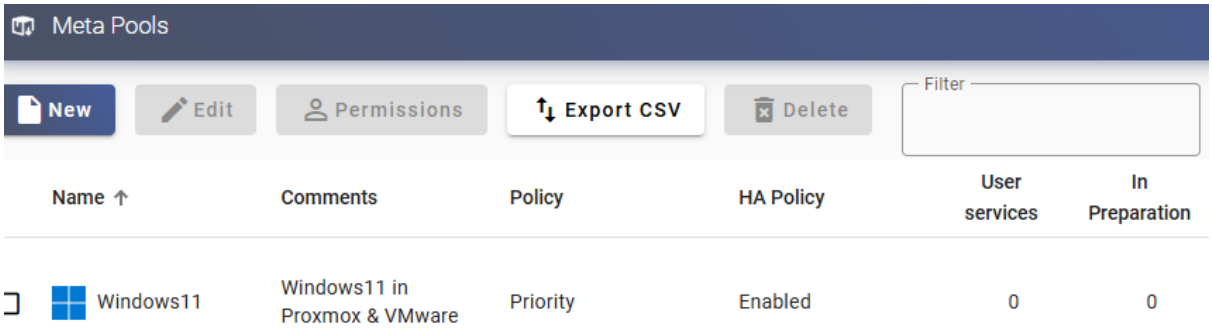
Transport Selection: We will indicate how the transports will be assigned to the "***meta pool'"***":

- **Automatic selection**: The available and lowest priority transport assigned to the Service ***Pool*** will be available in the ***meta pool***. Transport selection is not allowed.

- **Use only common transports**: Existing transports that are shared by all ***Service Pools*** will be available in the ***meta pool***".
- **Group Transports by label**: Those transports that have different and ***grouped labels*** will be available in the "***meta pool***" (this field is within each "***Transport***" in the "***advanced***" tab).).

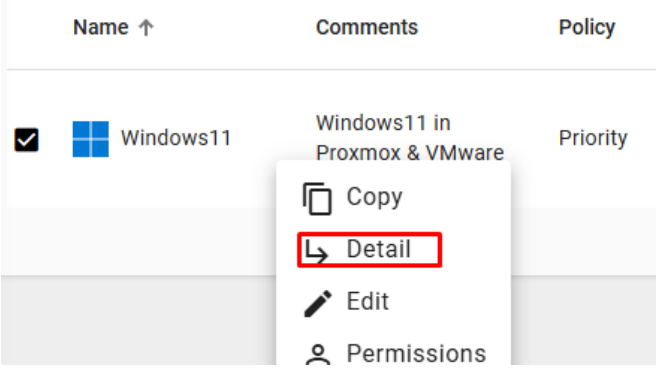


Save the configuration and we will have a valid "***Meta Pool***" to start registering "***Services Pools***"".

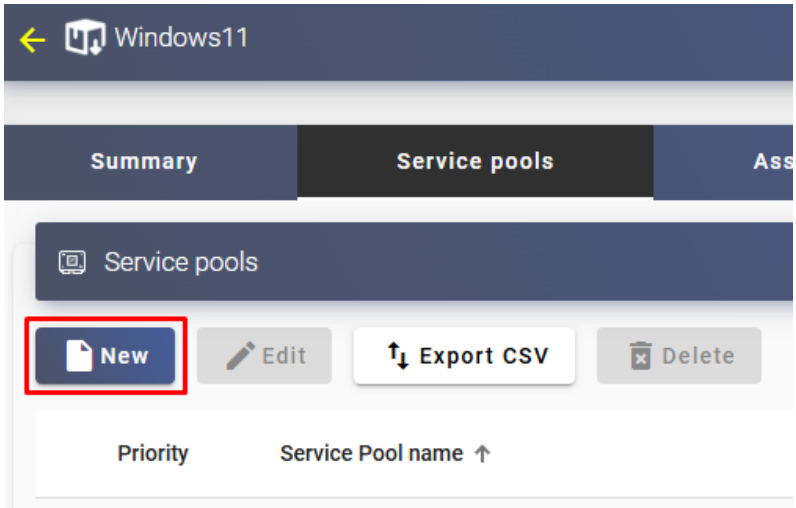


To modify a parameter in an existing ***"Meta Pool***", select it and click on "***Edit***".

Once created, we must add "***Services Pools***". To do this, we double-click on the "***Meta Pool***" created or in the provider's menu select "***Detail***":



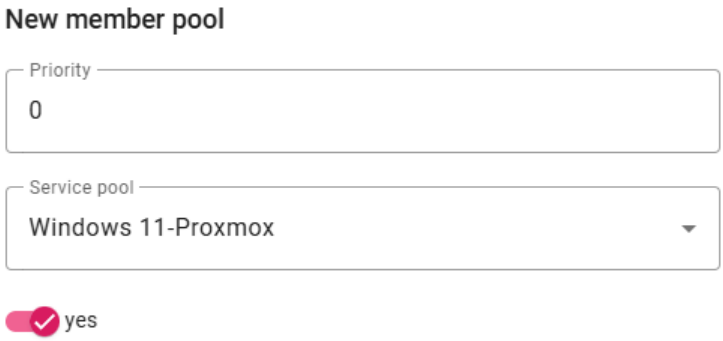
Click on "***New***" to add all the "***Services Pools***" that the "***Meta Pool***" will contain. We can add as many as we need, combining services hosted on different virtualization platforms (VMware, Proxmox, Azure, etc...), application servers and static computers.



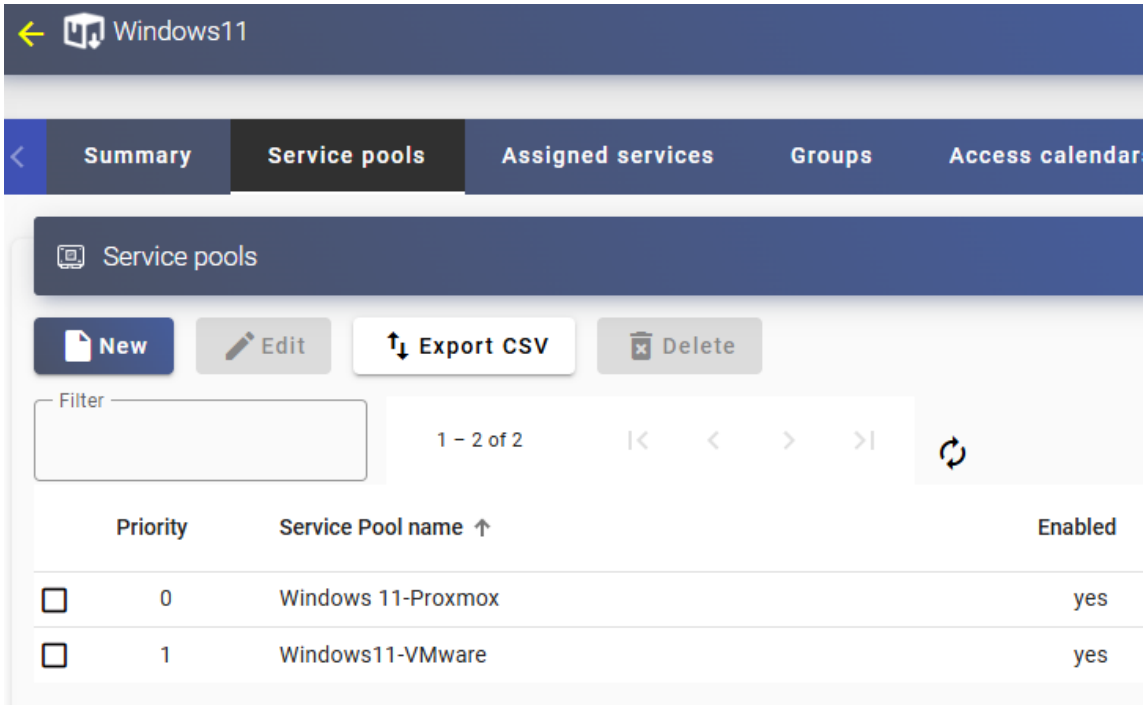
To add a "***Service Pool***" we must indicate the following parameters:

Priority: Priority that the ***"Service Pool***" will have in the "***Meta Pool***", the lower the value, the more priority it will have with respect to the rest of the elements. Applies to the "***Priority***" balancing policy

Service pool: Name of the "***Service Pool***" we want to add. It must be previously created.
Enabled?: Enables or disables the use of the service pool in the "***Meta Pool"***".



We can add as many as we need, combining services hosted on different virtualization platforms (VMware, KVM, Azure, etc...), application servers and static equipment.



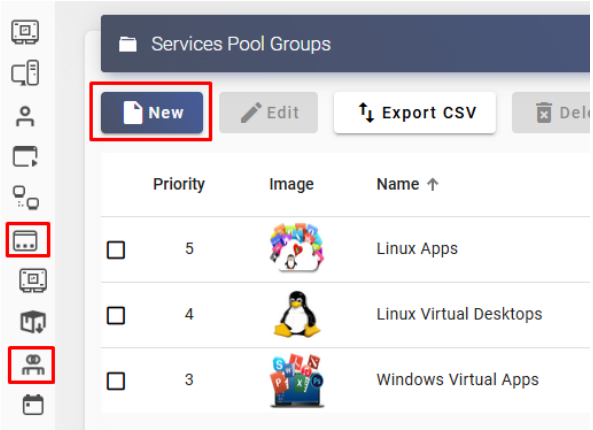
As in a "***Service Pool***", here we will also have the following information and configuration tabs:

- **Assigned services:** Displays services assigned to users, allowing them to be manually deleted and reassigned to another user.
- **Groups:** Indicates which groups of users of the different authenticators registered in the system will have access to the service.
- **Access calendars:** Allows you to apply a previously created access calendar.
- **Logs:** Displays all events that occurred in the "***Meta Pool'***".

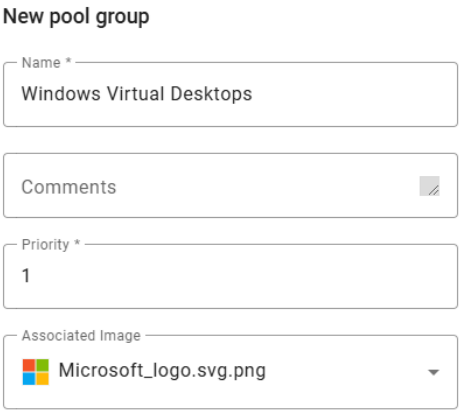
7.3 Groups

UDS allows services to be grouped together to facilitate their access and location. In addition, each service grouping can be assigned a name and image. If "Groups" are not defined, the services will be located in the default system, which is created by the system.

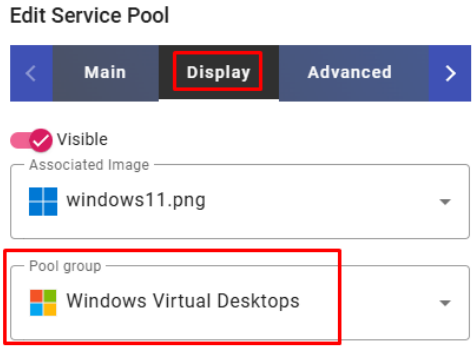
To create "***Groups***", we will hover over the "***Pools***" section and select "***Groups***":



Select "***New***" and indicate a descriptive name. We assign the priority of the pool group (the lower the value, the more priority it will have with respect to the rest of the elements.) and associate an image (it must exist in the image gallery, section "***Tools***" – "***Gallery***").



Once created, it will be available for assignment to a "***Service Pool***", within the "***Display***" section in the field: Pool Group:



7.4 Access calendars and scheduled tasks

UDS Enterprise incorporates a system to allow or deny access through calendars.

They allow you to enable or restrict user access to virtual desktop services and applications by dates and time slots.

With the use of calendars, it is also possible to schedule and automate certain tasks on a Service ***Pool***, such as making new publications, adjusting system cache values, adding or removing groups and transports, or changing the maximum number of services.

Safe Employment Procedure:

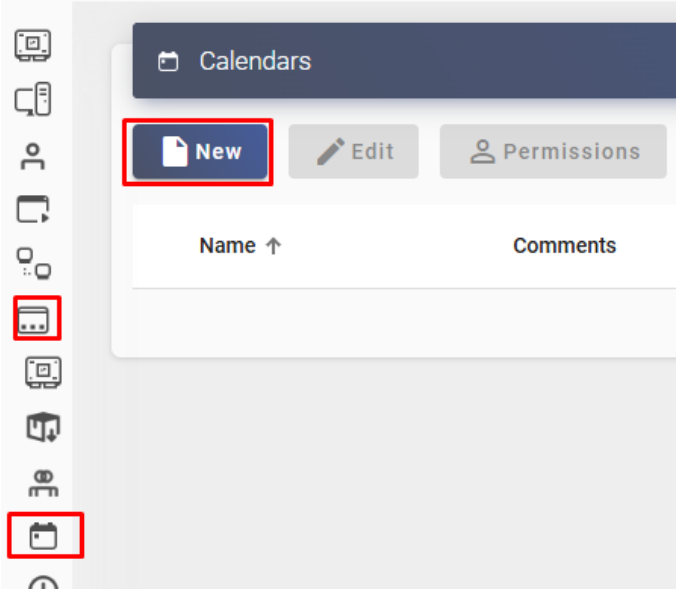
If users, or certain users, can only access the services, or certain services, restrict such users' access to those services to the period in which they can access them.

If access through certain protocols, or from certain networks, to certain services can only be done in certain periods of time, combine them in different "Service pools" in such a way that you can limit them through calendars.

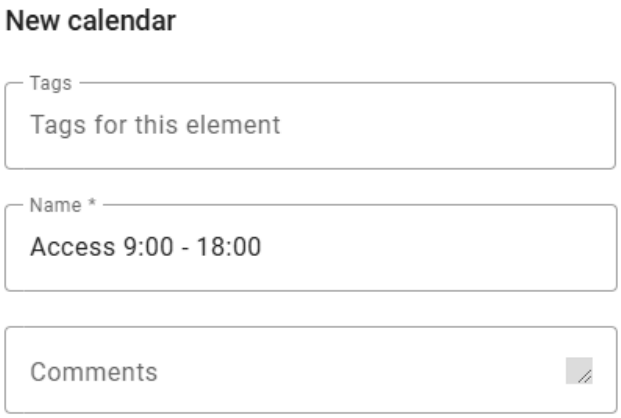
If you are scheduling a task, check its operation first in real time, in this way you will prevent unwanted effects from occurring without its monitoring.

7.4.1 Calendars

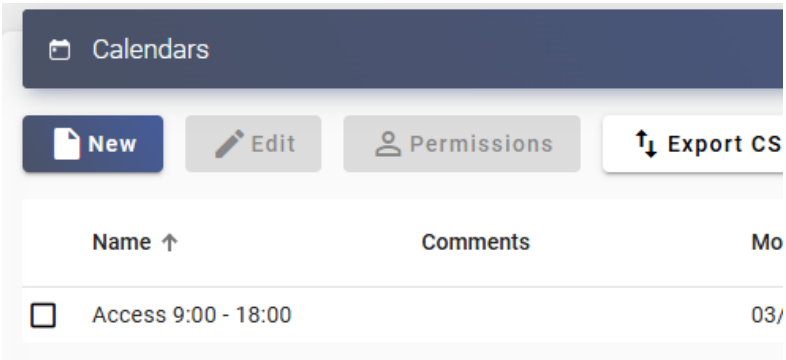
To create "***Calendars***" we will hover over the "***Pools***" section and select "***Calendars***".



We indicate a descriptive name to identify the calendar.

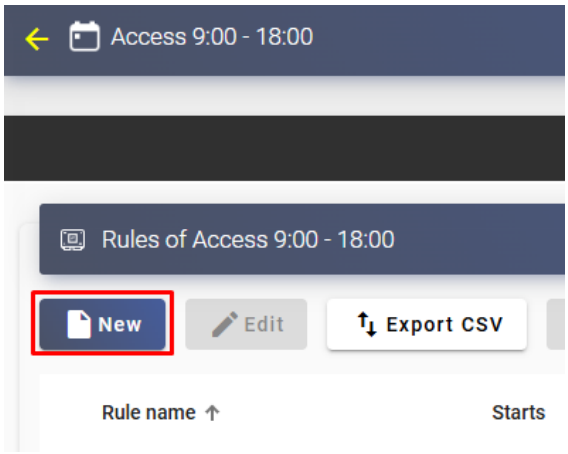


Save and we will already have a valid calendar to start creating the rules that we will later apply to a service through the "***Services Pool***"".



In a Calendar we can register several types of rules in which we could schedule the availability of services at certain times.

To create a rule, access the calendar and click on "***New***".



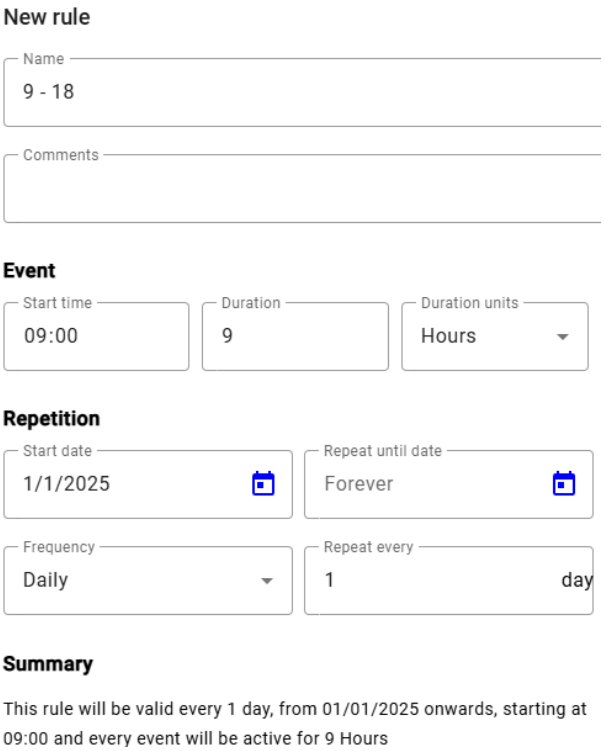
The minimum parameters to be configured in a Rule are:

Name: Name of the rule.

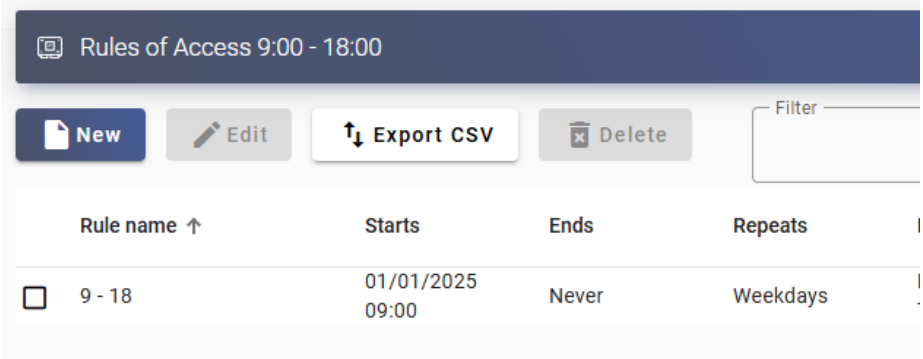
Event: Setting up run periods. To do this, we indicate the start time and duration of the event (in minutes, time, days and months).

Repetition: In this section we can configure that the rule repeats in days, weeks, months, years, and even allows us to specify working days. Finally, we can indicate repetition intervals per day.

Summary: Displays a summary of all previously performed configuration.



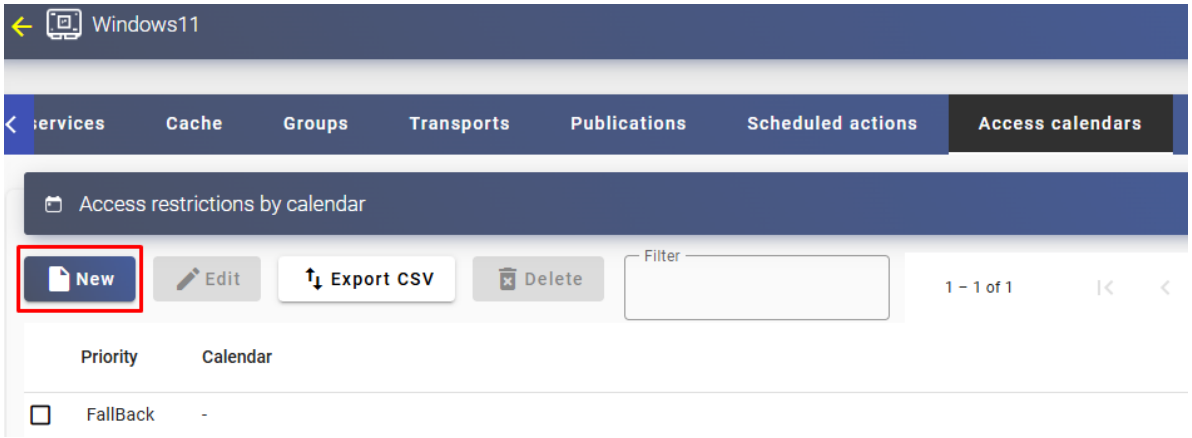
Save and have a valid rule to assign to a "***Service Pool***" (virtual desktop and/or application).



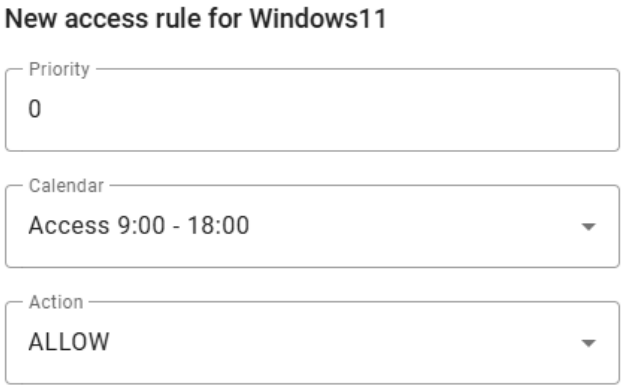
7.4.1.1 Enable or deny user access

Once the rules are configured in the calendars, we can use them to enable or deny users access to desktop services and virtual applications.

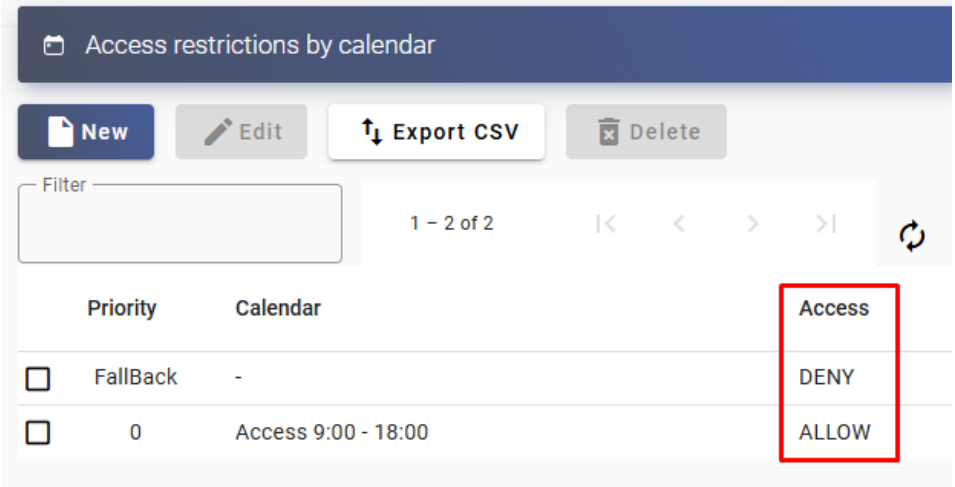
To apply these calendars with their rules, select a "***Service Pool***", go to the "***Access Calendars***" tab and click on "***New***":



We indicate the access priority, select an existing calendar and mark the action that we will apply in the access to the service.



Save and we will have a "***Service Pool***" with a configured access calendar.

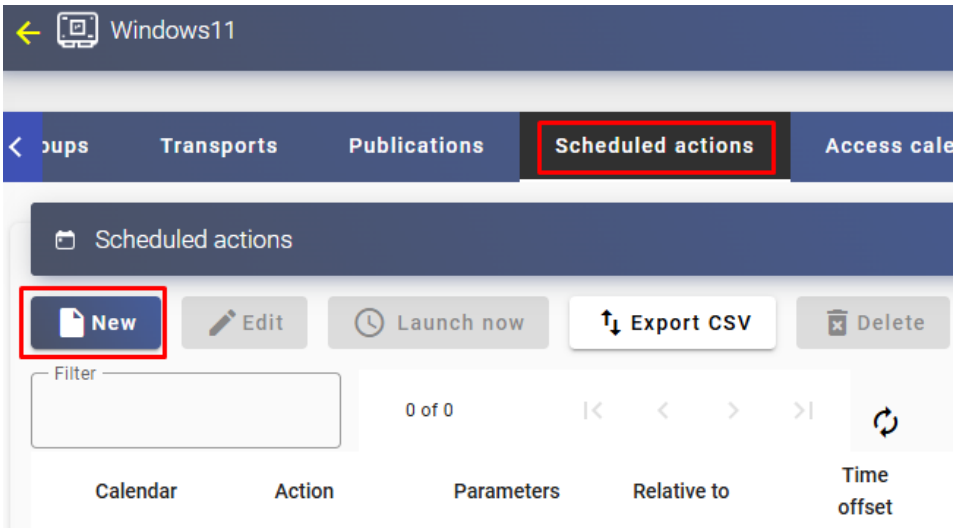


NOTE: The default rule "FallBack" will have to be adjusted based on the needs of the service to allow or deny access to the service when the calendar does not apply.

7.4.2 Scheduled Actions:

Once the rules have been set up in the calendars, we can use them to schedule certain tasks on a Service ***Pool***.

To apply these calendars with their rules, select a "***Service Pool***", go to the "***Scheduled actions***" tab and click on "***New***".



We indicate an existing calendar, the time during which the action will be executed and select the action to be carried out:

Set initial services: Rebasing of the minimum virtual desktops created and configured.

Set cache size: Readjustment of the virtual desktops available in the system cache.

Maximum number of services set: Modifies the maximum number of virtual desktops.

Publish: Creation of a new publication.

Add an transport: Add existing transport.

Remove a transport.

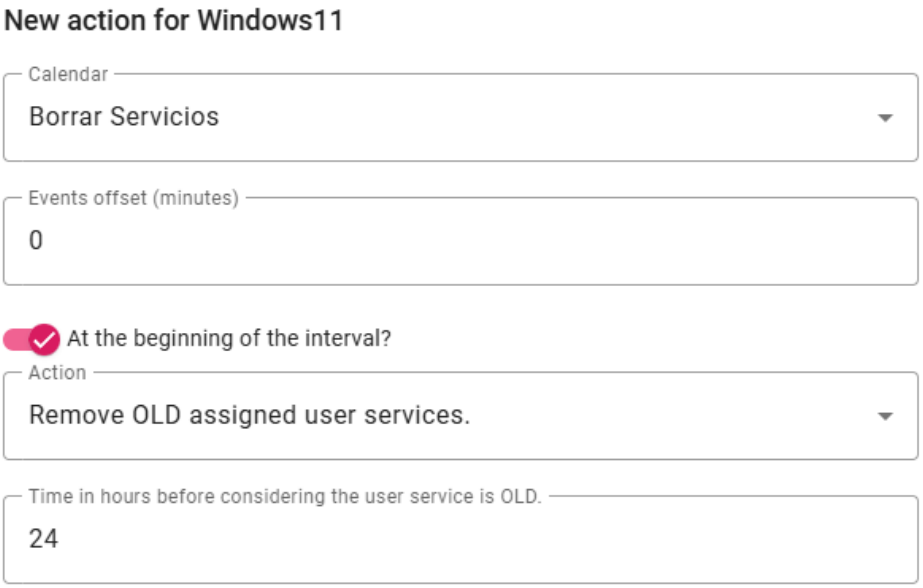
Add a group: Add an existing user group.

Remove a group: Delete user group.

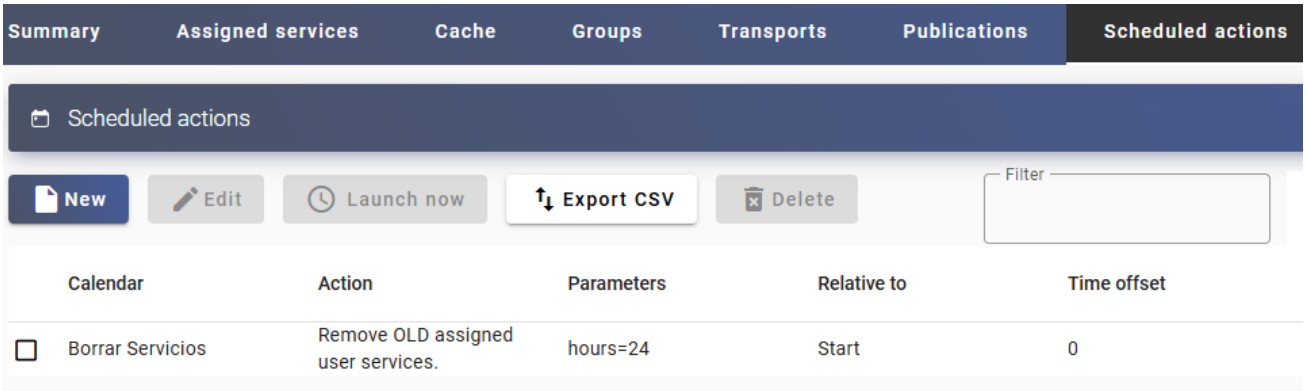
Sets the ignore unused: Set the "***Ignores unused***" option.

Remove ALL assigned user service: Deletes all services assigned to users in the Service ***Pool***.

Remove OLD assigned user services: Deletes services with an allocation time greater than the specified value (in hours).



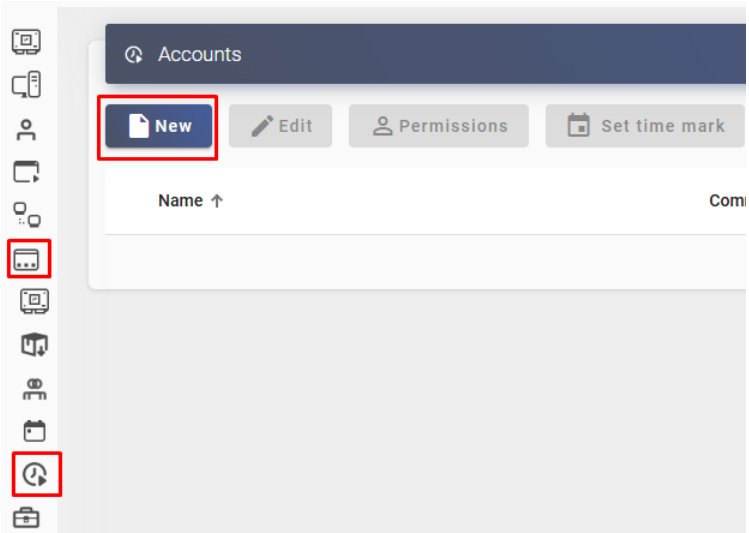
We save and we will have a scheduled task that performs a specific action on a "***Service Pool'***".



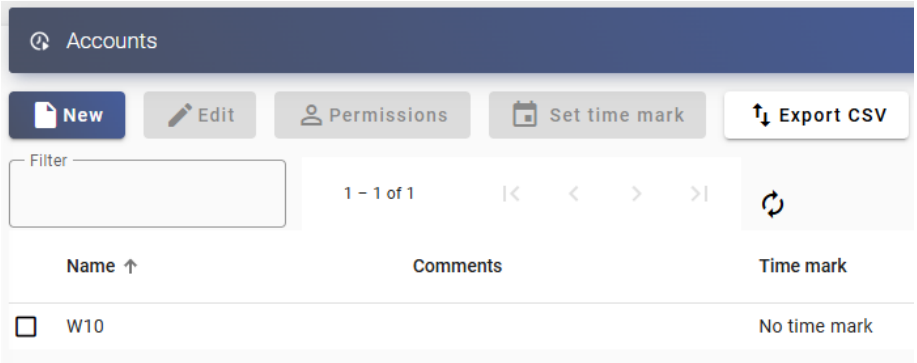
NOTE: If we click on "Launch now" we can execute the action at that very moment.

7.5 Accounting

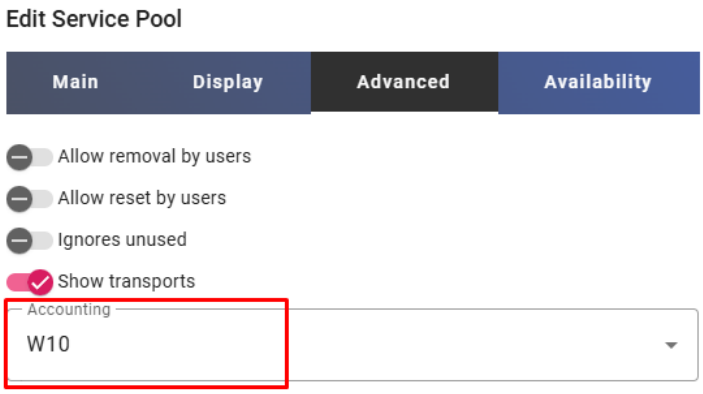
UDS allows you to apply a usage counter to a service pool, in order to account for the time of use of a service by a user. To create a counter we will access the "***Pools***" menu in the "***Accounting***" section and click on "***new***".



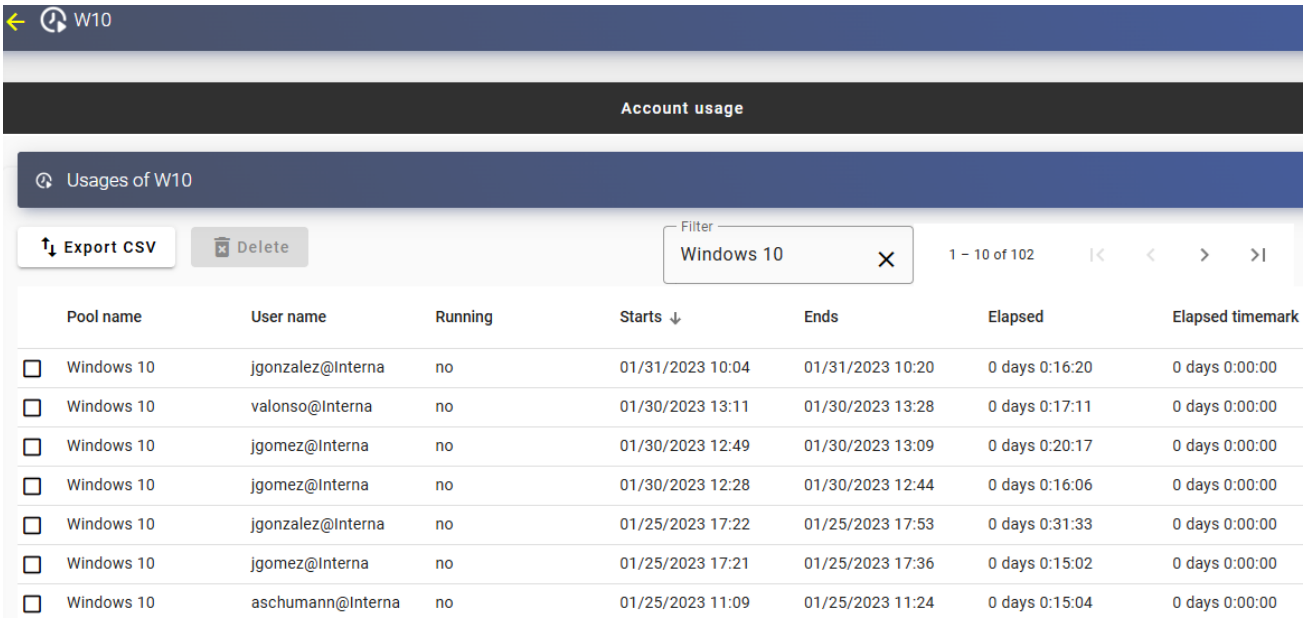
We will indicate a name for the element and save:



Once the counter has been created, we must assign it to the Pool of services that we are going to post. We edit the pool, tab "***Advanced***" parameter "***Accounting***".

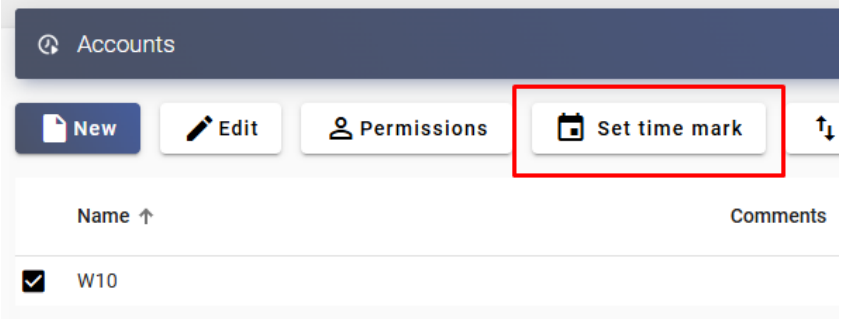


Each time a user accesses a service in the pool, it will be recorded indicating whether it is currently in use, start and end date and the total connection time made.



Pool name	User name	Running	Starts	Ends	Elapsed	Elapsed timemark
Windows 10	jgonzalez@Interna	no	01/31/2023 10:04	01/31/2023 10:20	0 days 0:16:20	0 days 0:00:00
Windows 10	valonso@Interna	no	01/30/2023 13:11	01/30/2023 13:28	0 days 0:17:11	0 days 0:00:00
Windows 10	jgomez@Interna	no	01/30/2023 12:49	01/30/2023 13:09	0 days 0:20:17	0 days 0:00:00
Windows 10	jgomez@Interna	no	01/30/2023 12:28	01/30/2023 12:44	0 days 0:16:06	0 days 0:00:00
Windows 10	jgonzalez@Interna	no	01/25/2023 17:22	01/25/2023 17:53	0 days 0:31:33	0 days 0:00:00
Windows 10	jgomez@Interna	no	01/25/2023 17:21	01/25/2023 17:36	0 days 0:15:02	0 days 0:00:00
Windows 10	aschumann@Interna	no	01/25/2023 11:09	01/25/2023 11:24	0 days 0:15:04	0 days 0:00:00

If it is necessary to add timestamps to perform calculations from a certain point in time, we can apply it to the counter using the "***Set time mark" option***".



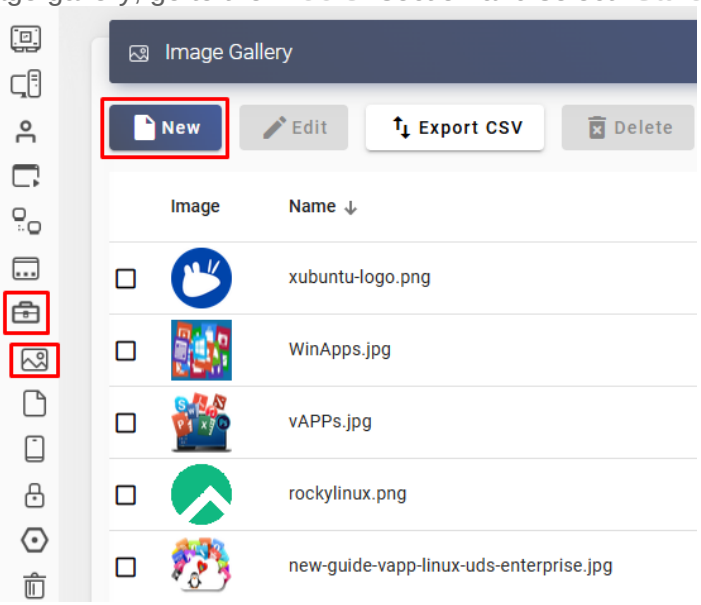
8. TOOLS

UDS Enterprise provides a series of tools that allow you to provide greater flexibility to the system, providing customization, usage reports, advanced parameters, etc…

8.1 Gallery

UDS has a repository of images that can be associated with a "***Service Pool***" or "***Pool Group***" to facilitate the identification of the service by the user. Accepted formats are: PNG, JPEG, and GIF. If the size of the image is larger than 128x128, it will be resized to these values (it is recommended to use square images to avoid deformations in these when performing the conversion).

To access the UDS image gallery, go to the "***Tools***" section and select "***Gallery***".



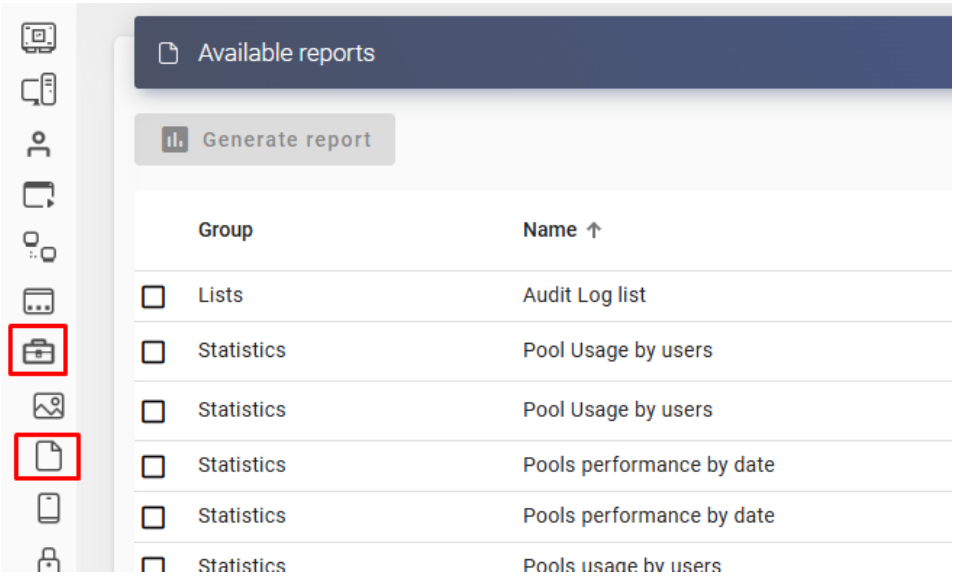
Select "***New***" to add a new image to the repository. It will be necessary to indicate a name and through the "***Select image***" button we will search for the image that we want to upload to the gallery.

8.2 Reports

UDS includes a series of predetermined reports that allow us to know the use of the platform, these reports can be generated mainly to know the interaction of users with the different services, grouped by service pool, although we will also have reports more oriented to audits, lists of users registered on the platform, access to the login portal, etc...

Most reports will be available in pdf and csv formats.

To generate a report, go to the "***Tools***" section and select "***Reports***":



These reports offer a simple and quick way to obtain information on how users use the UDS environment, but it is not the only one, customizable reports can also be generated using:

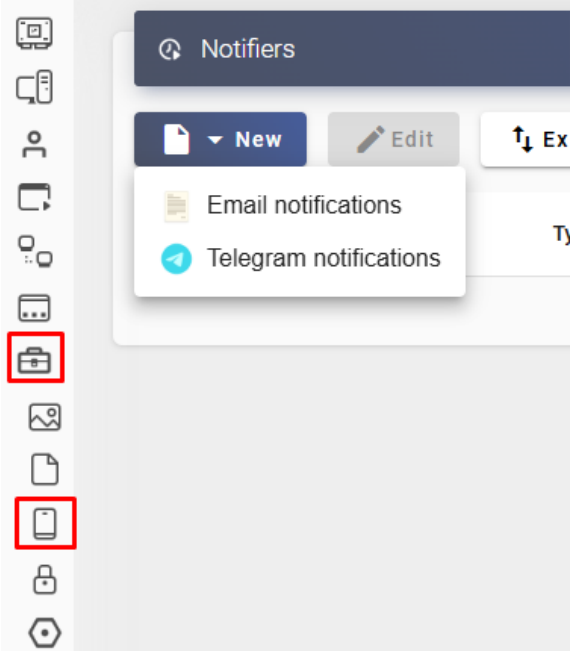
- UDS server log files: /var/log/uds/use.log and /var/log/uds/trace.log
- Making SQL requests to the database server, where all the system information is stored.

Additionally, it is possible to integrate UDS with real-time monitoring systems, such as software such as Grafana or Zabbix. In this link you can see an example of UDS integration with open monitoring tools.

8.3 Notifiers

It is possible to register two types of notifiers in UDS, so that events of certain types (warning, error and critical) are sent automatically.

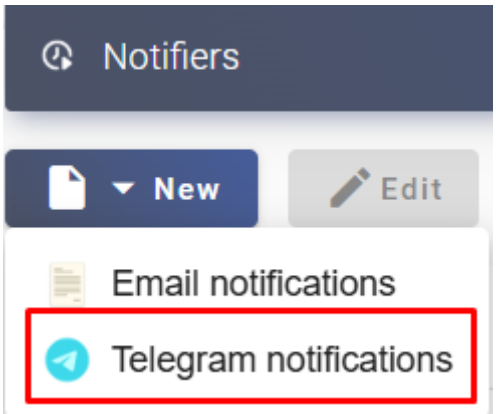
To create a new notifier, go to the "***Tools***" section and select "***Notifiers***". We will be able to configure a notifier via email and another via Telegram:



8.3.1 Telegram

This notifier will send messages via the Telegram application and will be able to send events categorized as: "***Warning***", "***Error***" and "***Critical***".

To create the notifier, click on "***New***" and select "***Telegram notifications***".



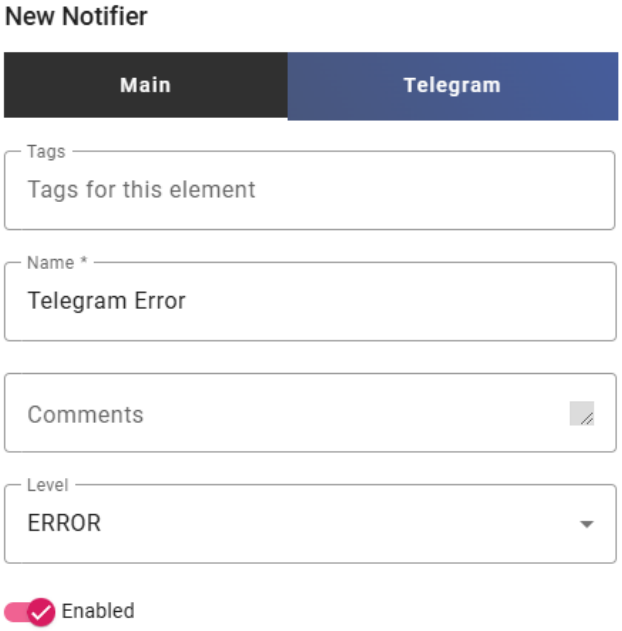
We will indicate the following data:

- Main

Name: A descriptive name of the element

Level: The events generated by the system and that are cataloged in this level will be sent. We can choose between events: Warning, Error or Critical.

Enabled: If active, the events will be sent.

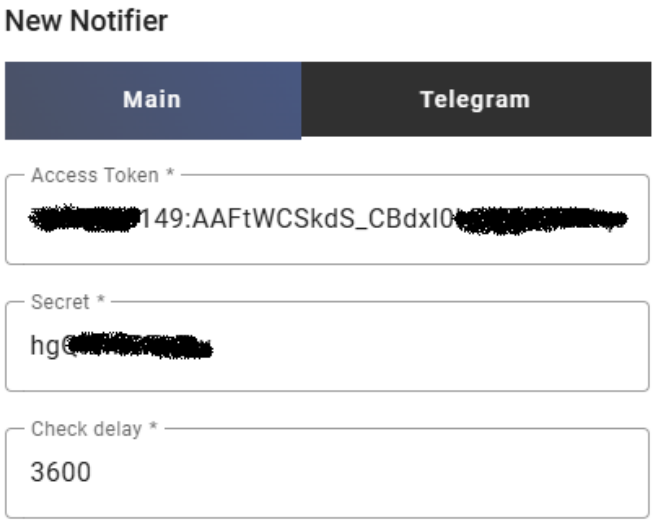


- Telegram

Access Token: Token de acceso generado en Telegram.

Secret: Validation key used.

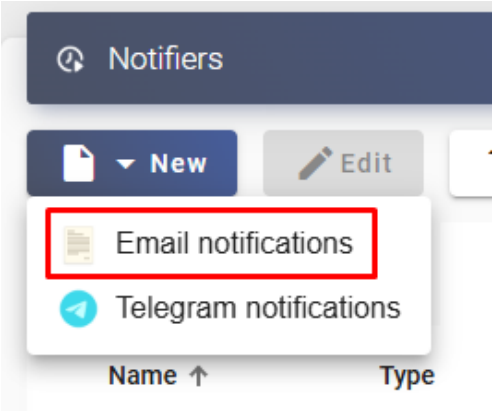
Check delay: Time between checks in seconds.



8.3.2 Email

This notifier will send messages via email and can send events categorized as: "***Warning***", "***Error***" and "***Critical***".

To create the notifier click on "***New***" and select "***Email notifications***".



We will indicate the following data:

- Main

Name: A descriptive name of the element

Level: The events generated by the system and that are cataloged in this level will be sent. We can choose between events: Warning, Error or Critical.

Enabled: If active, the events will be sent.

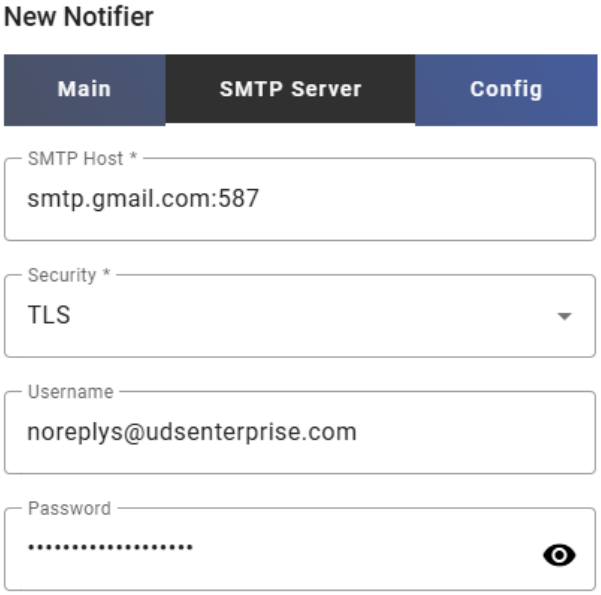
- SMTP Server

SMTP Host: Mail sending server.

Security: Type of security applied.

Username: User account with permissions to send emails on the indicated server.

Password: The user's password that will be used to send emails.

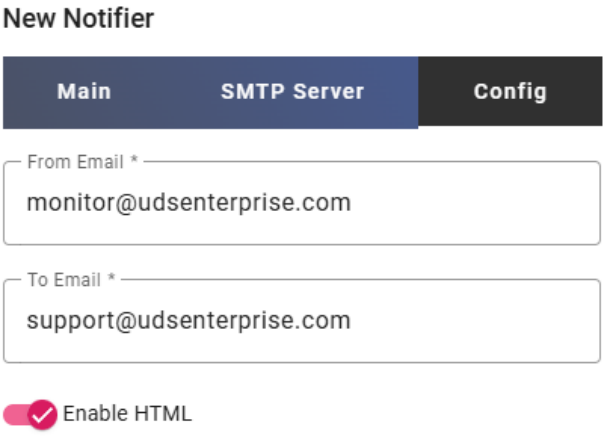


- Config

From Email: An email account that will be displayed in the send.

To Email: The email account that receives the messages sent.

Enable HTML: If enabled, HTML code will be enabled in the message.



8.4 Configuration

UDS Enterprise provides a series of parameters that will define certain aspects of the system's operation. These parameters will be responsible for defining aspects such as safety, mode of operation, connectivity, etc... both of the UDS system itself and of its communication with the virtual platforms registered in UDS.

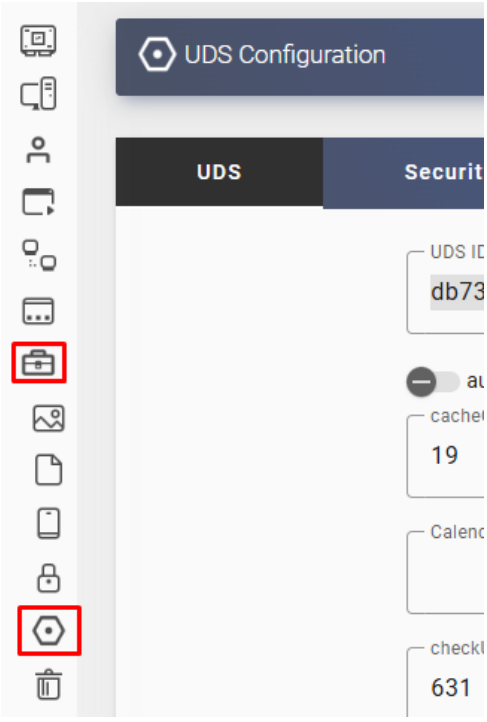
This manual shows only some of the variables of the system, which are considered to be the most useful for the management of virtual desktops.

In the rest of the variables it is recommended not to modify the default values, an incorrect modification of a parameter can cause the system to stop completely or not work correctly.

NOTE: Once the values of any of the UDS advanced configuration variables have been modified, it will be necessary to restart the UDS Server to apply the changes.

In case you want to modify any value not documented in this section, it is recommended to contact the UDS Enterprise support team to verify such change and confirm that it does not negatively affect the operation of the UDS system.

To access the parameters of the UDS advanced configuration, access the "***Tools***" section and select "***Configuration***":



Procedimiento de Empleo Seguro:

Read very carefully each of the parameters, their default value and the consequences of choosing one value or another.

There are some that affect security directly, grouped under the heading "security" and many others that can affect distributed under other headings.

Remember that the modification of any of these parameters may have consequences or generate other collateral problems.

As we said in the introduction to this document, the application of many of them entails a very deep knowledge of the systems that are being managed, many times there are multiple dependencies and/or incompatibilities between the systems.

It is not recommended to carry out any intervention on the appliances unless you are absolutely certain of what is being done and how it will affect the operation of the software.

In any case, it is advisable to have at least one backup of the appliances before undertaking any modification and, as they are virtual machines, to take a snapshot beforehand to be able to revert the modifications immediately if necessary.

It is advised that any changes be made first, performing all the necessary tests, in a non-production environment, as is obvious.

It should also be taken into account that the application of certain security mechanisms affects:

- Performance, such as the encryption and decryption of communications,
- To abandon legacy systems, because they demand high minimum supported levels of security.
- To assume additional costs, such as having to contract recognized certificates.

As in any other project for the implementation of security measures, the risks must be assessed, identifying the threats, vulnerabilities, impact and their treatment. A platform in a private environment is not the same as one exposed to the internet, a platform for sporadic work is not the same as a general use, etc.

8.4.1 UDS

UDS ID: Identification code of the installed instance of UDS Enterprise.

AutorunService: If enabled, automatically makes the connection to the user's service (via "default" transport) after the user is validated in the login portal. The activation of this parameter affects all users registered on the platform.

By activating this parameter, the user will not see the user services page. **If the user validated in the login portal has more than one service available, this parameter will not be applied**, and the user services page will be displayed as normal.

Default: Disabled.

CacheCheckDelay: Indicates how often (in seconds) the system cache data will be checked to generate or delete machines. With this value, the speed with which machines are generated and destroyed can be accelerated or decreased (very low values can saturate a hypervisor platform).

For example, with the default, the cache system will create a machine every 19 seconds

Default: 19

Calendar Access denied text: Generic text displayed to all users when they access a service that is limited by a calendar.

If there is any text within a service pool, in the "Calendar Access Denied Text" field, it will overwrite the text of this parameter.

CheckUnusedTime: Indicates how much time has to pass (in seconds) for the system to delete a service that is in use "no" and also has an OS Manager of type "Remove service" (non-persistent). If you have an OS manager of any other type, it will not apply this check.

For example, if a VDI pool with a non-persistent OS manager has a service assigned to a user, but the user has not logged in (the connection has been cancelled, the machine has had an error, the actor has not run correctly and has not detected the usage, etc...), after the time of this variable (it can be the time of this variable x2) it will be automatically deleted because it is considered erroneous.

Default: 631

DisallowGlobalLogin: If enabled, it does not display the global list of authenticators in the login portal.

If enabled, users will be validated on the authenticator "by default" or with higher priority.
To validate with other authenticators and allow users access to the system, it will be necessary to use the "label" in the access URL (defined in the authenticator).

Default: Disabled.

Experimental Features: Habilita características no verificadas y que pueden producir mal funcionamientos en el entorno. No se recomienda habilitarlo.

KeepInfoTime: Defines the time (in seconds) that the completed events of a Service Pool remain visible. These events may be already completed or failed.

Once these events are deleted, for example, the names of machines, IP addresses or MACs of the services will be reused.

Default: 14401 seconds (4 hours)

Max entries for general UDS logs:

MaxInitTime: Time that a service can be in a configuration state or preparing, after this time (in seconds) it will be considered to have failed and will be deleted.

Default: 3601

MaxLogPerElement: The maximum number of logs to be kept in the log of an item in the administration.

Default: 100

MaxRemovalTime: The maximum time (in seconds) that an item in the deletion state will remain. After this time, it will be considered that some error has occurred in the deletion and the task will be deleted.

Default: 14400

Notify on new publication: If it is active, when a new publication is made and there are users with assigned services from previous publications, a notice will appear indicating that a new publication exists, and their current service is going to be deleted.

If this parameter is not active, the service will still be deleted, but the user will not be notified. The deletion time will be defined in the "SessionExpireTime" variable.

Default: Disabled

Page reload Time: Parametro legacy. Modification is not recommended.

Redirect to tag on logout: If enabled, when a user who accesses the UDS portal via URL with a tag logs out, they will be redirected to the portal with the same tag.

Default: Disabled

RemovalCheck: The check time (in seconds) that the system will use to check the cache items that it needs to mark for deletion. Modification is not recommended.

Default: 31

RestrainCount: The number of errors that have to occur within the time indicated in the "Restrain Time" parameter for a service pool to enter restricted mode. When this number is reached and restricted mode is activated, any operations performed in the service pool are stopped.

In some environments it is advisable to increase the default value as it is initially very conservative to avoid overloading the virtualization platform.

Default: 3

RestrainTime: The time that has to elapse before a service pool enters restricted mode according to the number of errors indicated in the "RestrainCount" parameter.

With the defaults of this parameter and "RestrainCount", if 3 errors occur in 600 seconds in a service pool, it will enter restricted mode and when that condition is no longer met, the pool will leave restricted mode and return to normal operation.

Default: 600

SessionExpireTime = Indicates the maximum time a user session will be open after a new post has been made. After that time, the system will log out the user and proceed to delete the service. If the service has an OS Manager with "***Keep service assigned even on new publication***" as the persistence policy on the virtual desktop, this value does not apply.

Default: 24 hours.

StatsAccumFrequency: The frequency (in seconds) with which statistics for services will be collected. Modification is not recommended.

Default: 3600

StatsDuration = Time (in days) that the system will save the statistics. If this value is increased, the required space in the database can be increased.

Default: 365 days.

UserServiceCleanNumber: Maximum number of services that will be marked for simultaneous deletion for the overall service pools.

Default: 24

8.4.2 Security

Allowed IP Forwarders: IP addresses of forwarders that will be accepted by the system, for example when there is an HA configuration and there are load balancers ahead, we can indicate the IPs of these balancers so that they are accepted.

The value "*" (default) indicates that all forwarder IPs are accepted.

AllowRootWebAccess = Allows the login of the super-user (user created in the UDS-Server configuration wizard) in the UDS login portal.

Modifying this variable will not affect root user access through the Linux OS console.

Behind a proxy = Tells the system that UDS servers are "behind" a proxy (e.g., a UDS environment in high availability with an HA Proxy-type load balancer).

When a UDS environment is in HA, it is recommended to enable this parameter.

Block ip on login failure: Enables that in addition to blocking a user who has failed several times in the login portal, the IP address of their connection client is also blocked.

Enforce Zero-Trush Mode: Instructs the system not to travel any password on any connection and also no type of credential will be stored in the database.

LoginBlockTime: Time that a user will be blocked (in seconds) after entering their password incorrectly the times indicated in the variable "***maxLoginTries***". Default: 300 seconds (5 minutes).

MaxLoginTries: The number of attempts a user will have to enter their password before the system locks them out.

RootPass = Super-user password created in the UDS-Server configuration wizard.

Safe use procedure: Passwords must be of sufficient length and include upper and lower case, numbers and special characters.

SuperUser = Name of the super-user created in the UDS Server configuration wizard.

Session timeout for Admin = Time in seconds until an admin who takes no action is logged out.

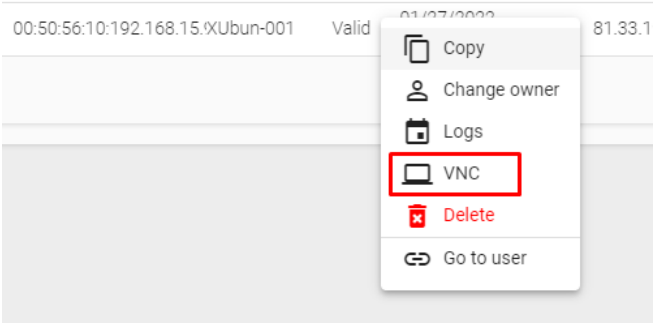
Session timeout for User = Time in seconds until a user who does not take any action is logged out.

Trusted Hosts = Hosts that you consider to be secure. These hosts can make "sensitive" requests to UDS, such as tunnel servers. Separated by commas.

> Supports full subnets, ips range, and specific public ips
> Default: "*" (all allowed).

8.4.3 Admin

Enable VNC for user services = If enabled, a new option will appear in the assigned services of a Service Pool to allow connection over VNC.



By clicking on "VNC" a file will be generated with all the information to connect to the service through a VNC client (the client must be previously installed and there must also be direct network connectivity with the service)

Default: Disabled

List page size = Number of items to display. Applies to all sections of the administration.

Default: 10

Trusted Hosts for Admin: Filters from which IP addresses it is possible to manage UDS (including from web access to administration with the API) separated by commas.

Supports full subnets, ips range, and specific public ips.

8.4.4 Custom

Parameters related to the graphic customization of UDS (login and user services portal) are described:

Cookies consent enabled: If enabled, the cookie use notice will be displayed. UDS does NOT use third-party cookies, only its own and the necessary for the operation of the environment.

Cookies consent text: Text for the use of cookies that will replace the default one.

CSS: Supports css code to modify the login portal pages and UDS default services.

Logo name: Text displayed next to the top left image of the user menu bar.

Min. Services to show filter: The minimum number of services that must exist in a user's services window (user mode) for the filter option to be displayed.

Show Filter on Top: Allows you to modify the location of the service search bar on the user services page (user mode).

Site copyright info: Text that will appear at the bottom right of the login and user services page.

Site copyright link: Web address of the text in the "***Site copyright info***" section.

Site information: HTML code for partial customization of UDS login page.

The code entered will appear below the users' login box in the UDS login portal.

Site name: Text that will appear at the top of the users' login box in the UDS login portal.

8.4.5 WYSE

Parameters related to connecting to Dell Wyse clients are described:

Autoconnect = Allows the device to be auto-connected.

Default: no.

Colors = Defines the quality of the colors offered during connection.

Default High.

DefaultUser = Default user redirected to the device.

Default: UDS.

Language = Device language.

Default: us.

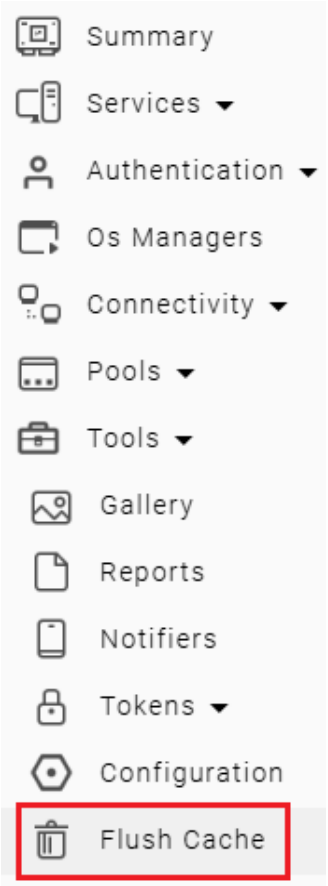
Privilege = A user's privilege level.

Default: NONE.

For more details on these parameters see the official Dell Wyse documentation.

8.5 Flush Cache

To empty the UDS system cache, go to the "***Tools***" section and select "***Flush cache***":



The most common reasons for clearing the system cache are:

- **Blocking a user:** When a user enters his password incorrectly the times indicated in the variable "maxloginTries" (security section in the UDS configuration), the system blocks that user, to unlock him immediately, it will be necessary to empty the system cache.

- **Actualización de inventario:** It is possible that when editing a "Service" some elements such as datastores, networks, base machines, etc... are not available (as these have been cached to avoid unnecessary requests). To be able to view them we will have to empty the system cache, in this way the Broker will make the request to the hypervisor again and the data will be updated.

9. PERMISSIONS

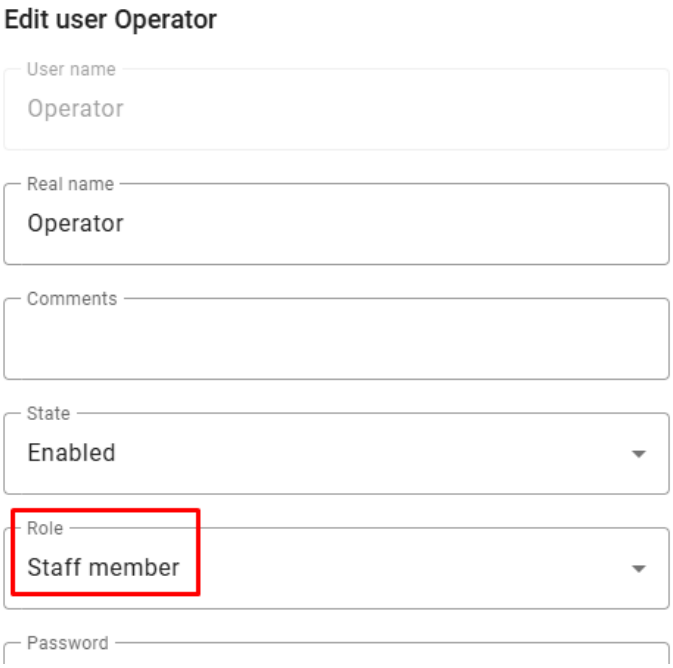
Within the administration of UDS Enterprise it is possible to assign access and management permissions to the different elements, to users and groups of users. Permissions will be assigned directly to each element and will apply to its sub-elements as well.

Safe Employment Procedure:

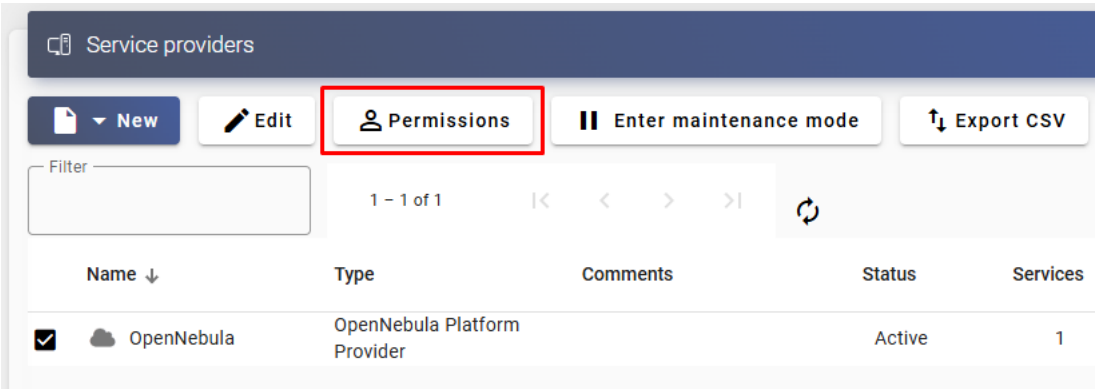
Do not authorize more users/groups, or assign more permissions than strictly necessary.

If you want to give access to a single user, it is better to create a group with that user than to register the user directly.

To allow a user to access the administration and to have these permissions applied to them, the user must have the "***Staff member" option enabled***":

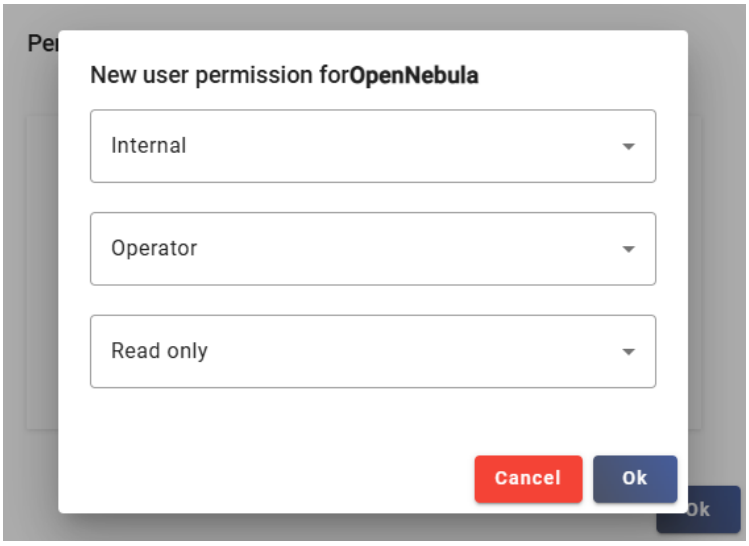


To enable permissions in the different elements of the administration we will have to select the element and click on "***Permissions***". For example, in a "***Service Provider***".

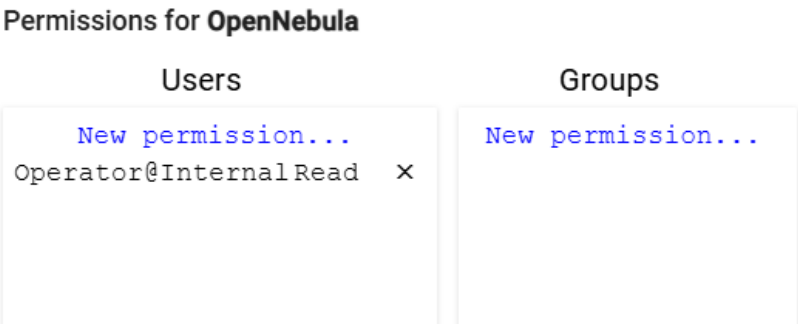


In the permissions window, click on "***New permission...***" for groups and for users and select the authenticator and group/user on which the permission will be applied.

We will have to specify whether this user or group will have read access to the element ("***Read Only***") or full control ("***Full Access")***").



Once applied, users who have the "***Staff member" option enabled*** will be able to access this element of the administration with the permissions assigned to them.

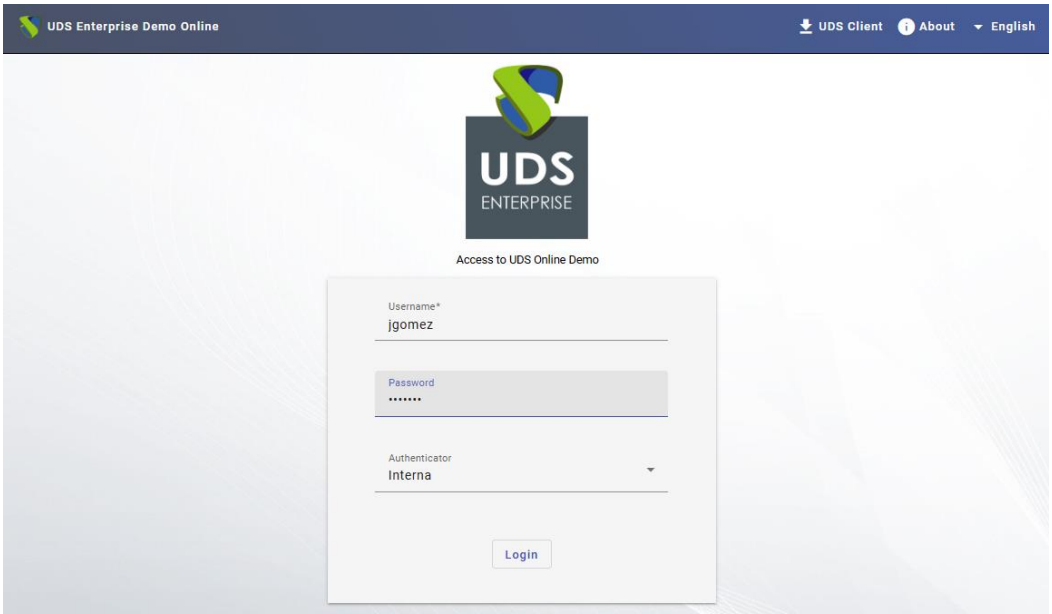


To remove permissions from a group or user, click on the "***X***" to remove it.

"***Full Access***" ("***Manage***") permissions can only be applied to elements that have a second level ("***Services***", "***Calendars***", "***Service Pools***", etc...).

10. ACCESSING SERVICES

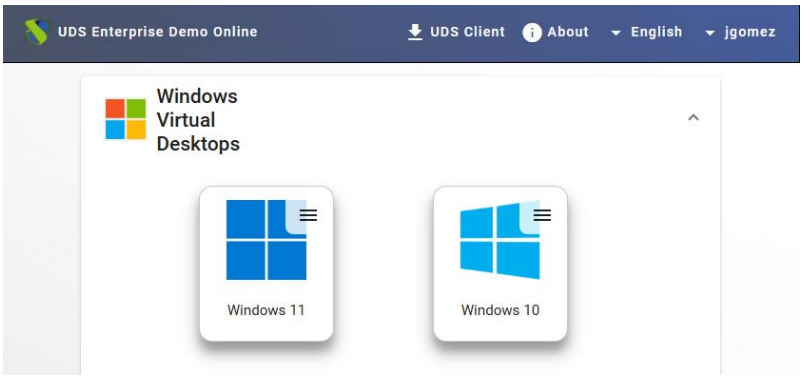
Once we have one or more "***Service Pools***" available with services published in valid state, we can initiate a connection with a virtual desktop. We will access via web browser the address or name of the UDS server (UDS-Server). We indicate a valid username and password and select the authenticator if you have more than one.



En la pantalla de servicios disponibles nos aparecerán los servicios a los que tiene acceso el usuario con el que hemos iniciado sesión en el sistema UDS. Hacemos clic sobre el que queremos iniciar la conexión.

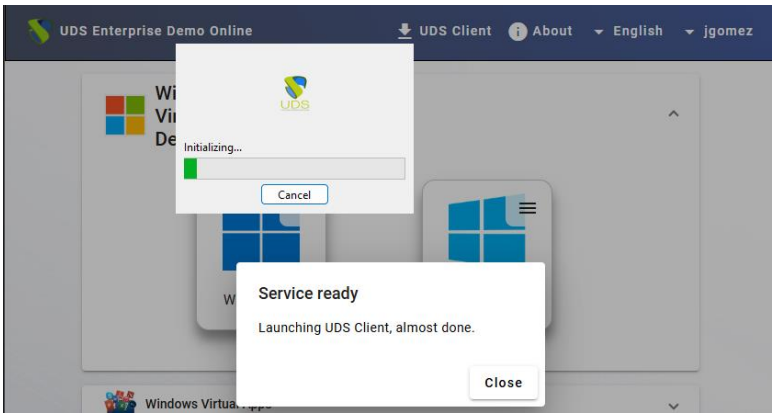
Por defecto, si hacemos clic sobre la imagen del servicio directamente, realizaremos la conexión con el "***Transport***" que tenga la prioridad más baja. Si hemos configurado varios, aparecerá un menú desplegable donde podremos seleccionar el "***Transport***" por el que nos conectaremos al escritorio virtual.

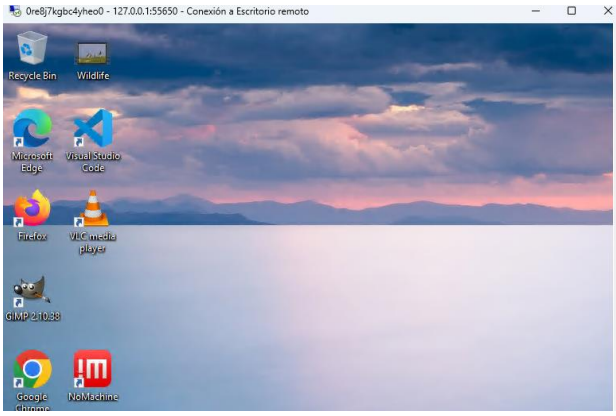
Si es necesario, o en caso de disponer de varios tipos de servicios (aplicaciones, escritorios Linux o escritorios Windows) podemos agrupar los servicios para facilitar el acceso a los usuarios:



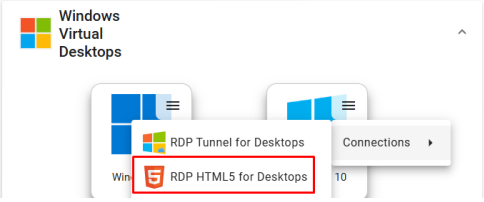
To initiate the connection to the virtual machine, it is necessary to have the "***UDS Client***" installed on the connecting client machine. This is required to be able to initiate the connection to all transports except HTML5.

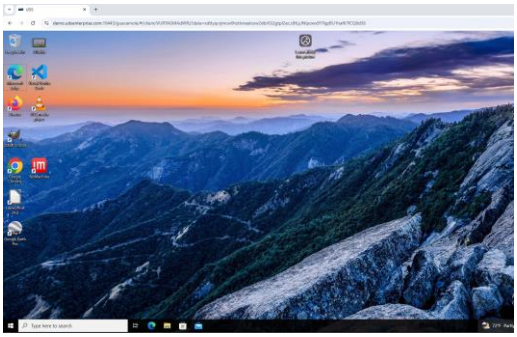
In order to connect with the desktop or virtual application, it is necessary to have the clients of each protocol used installed (RDP, NX, RGS, SPICE client, etc…).





HTML5 connection example:





11. ABOUT VIRTUAL CABLE

UDS ENTERPRISE, THE SOFTWARE VDI OF VIRTUAL CABLE

About UDS Enterprise

UDS Enterprise is a new software concept for creating a **fully customized virtualization** platform **for the workplace**. It provides **secure 24x7 access** from **any location and device** to all applications and software of an organization or educational center.

It allows you to combine Windows and Linux **desktop and application virtualization** in a single console, as well **as remote access** to Windows, Linux and macOS computers. Its Open Source base guarantees **compatibility with any third-party technology**. It can be deployed on premise, in a public, private, hybrid or **multicloud**. You can even combine several environments at the same time and perform automatic and **intelligent overflows** to optimize performance and efficiency. All with a **single subscription**.

About Virtual Cable

Virtual Cable is a company specialized in the digital **transformation of the workplace**. The company develops, supports and markets UDS Enterprise. It has recently been recognized as an **IDC Innovator in Virtual Client Computing** worldwide. Its team of experts has designed digital **workplace solutions** (**VDI, vApp and remote access to physical computers**) tailored to each sector to provide a unique user experience fully adapted to the needs of each user profile. Virtual Cable professionals have **more than 30 years** of experience in IT and software development and more than 15 years in virtualization technologies. **Everyday millions of Windows and Linux virtual desktops** are deployed with UDS Enterprise around the world.

All proper names of programs, operating systems, hardware equipment, etc. appearing herein are trademarks of their respective companies or organizations.

All rights reserved. The content of this work is protected by law, which establishes prison sentences and/or fines, in addition to the corresponding compensation for damages, for those who reproduce, plagiarize, distribute or publicly communicate, in whole or in part, a literary, artistic or scientific work, or its transformation, interpretation or artistic execution fixed in any type of support or communicated through any means, without the required authorization.

-END OF DOCUMENT-