



VIRTUAL
CABLE

VDI with UDS Enterprise 3.6 and Amazon Web Services (AWS)



UDS
ENTERPRISE

3.6



Index

Introduction	2
UDS Enterprise on Amazon Web Services.....	3
Where do I begin?	3
Deploy UDS servers on AWS	5
▪ User creation in IAM module	5
▪ Bucket and Role Creation	9
▪ Import UDS servers	12
▪ Creation of UDS servers.....	14
▪ Configuration of UDS servers	19
▪ Create base machines or templates on AWS	22
UDS Enterprise Administration	26
AWS service provider integration	26
▪ Creation of base services	33
Creation of Pool of Services	36
Monitoring UDS Enterprise.....	41
Common errors and troubleshooting	41
About VirtualCable.....	42

Introduction

Amazon Web Services (AWS) is a proprietary Amazon platform that offers cloud services. Some of its advanced features include the ability to run virtual machines, virtual applications, databases, backups, and many other tasks. It integrates countless cloud services that are necessary to develop, test, deploy and manage virtual machines (VMs).

This VDI Guide with UDS Enterprise & AWS will help you to know the procedure to deploy and configure the UDS Enterprise components on said platform. This document shows, through real examples, how to create the necessary resources so that UDS Enterprise can deploy virtual desktops on AWS.

UDS Enterprise is made up of 3 elements that interact with each other.

- **UDS Server:** It is installed as a virtual machine (VM) and is provided in virtual appliance format
- **UDS Tunnel:** It is installed as a VM, and it is provided in virtual appliance format
- **UDS Dbserver:** It is installed as a VM and is provided in virtual appliance format, (Optional for UDS Free & Evaluation Edition)

The MySQL Database manager is supported as of version 5.6

MySQL security details:

- Protects from network-level security attacks with data-in-motion encryption through TLS/SSL support in MySQL Connectors.
- Protects data at rest using native encryption and optional external key management providers such as Amazo.

UDS Enterprise on Amazon Web Services

Before carrying out the integration, it is convenient to invest time in knowing the different configurable parts of UDS Enterprise (for more information visit our [Web](#). In the section [Documentation](#) You will find the UDS Enterprise installation, administration and user manual). One of them is the Service Providers, an extremely important element for the configuration of AWS in UDS Enterprise.

UDS Enterprise will allow the deployment of self-generated virtual desktops and virtual application sessions on the AWS platform. The UDS components (Server, Tunnel and Database) can be hosted in the AWS environment itself or can also be hosted in any other on-premise virtualization platform with connectivity to the AWS environment.

To import, install and configure UDS Enterprise within an AWS environment, you must request its specific components for this environment (UDS-Server, UDS-Tunnel and Database) and a serial number (Evaluation/Enterprise) from Virtual Cable.

You must have a valid AWS subscription with permissions on which to deploy the UDS Enterprise components, virtual desktops or Windows/Linux application servers.

Where do I begin?

First of all, you must have an account with administrator privileges (you can use the IAM account with permissions or a "Root" account, this last option is not recommended for security reasons) on the AWS platform.

If you already have it, Log in to the [portal](#)



Sign in

Root user

Account owner that performs tasks requiring unrestricted access. [Learn more](#)

IAM user

User within an account that performs daily tasks. [Learn more](#)

Account ID (12 digits) or account alias

Next

By continuing, you agree to the [AWS Customer Agreement](#) or other agreement for AWS services, and the [Privacy Notice](#). This site uses essential cookies. See our [Cookie Notice](#) for more information.

————— New to AWS? —————

Create a new AWS account

If you already have an active UDS environment (on a virtual platform on premise or on another cloud platform) and want to integrate with AWS, you must perform the necessary configurations at the network level so that there is communication between the UDS servers and the AWS environment.. In this case, you can go directly to the "UDS Enterprise Administration" section.

If you want to host the UDS components within the AWS environment, the Virtual Cable team will provide you with these components in a specific format and you must perform a series of tasks to import them.

To manage the service limits of AWS you can see the following link [AWS service quotas](#)

Deploy UDS servers on AWS

Below is an example of how to deploy the servers that make up the UDS environment on an AWS platform. This guide details the steps to upload and create the UDS Server component. The same tasks must be carried out for the UDS Tunnel server and the database.

The estimated time of the full deployment is around 4 hours

If the UDS version to be installed is Enterprise, it will be necessary to upload the database server to the platform. If you use the UDS Evaluation Edition versions, you can not deploy a database server and activate a local database included in the UDS server, although this configuration will not allow the environment to be updated.

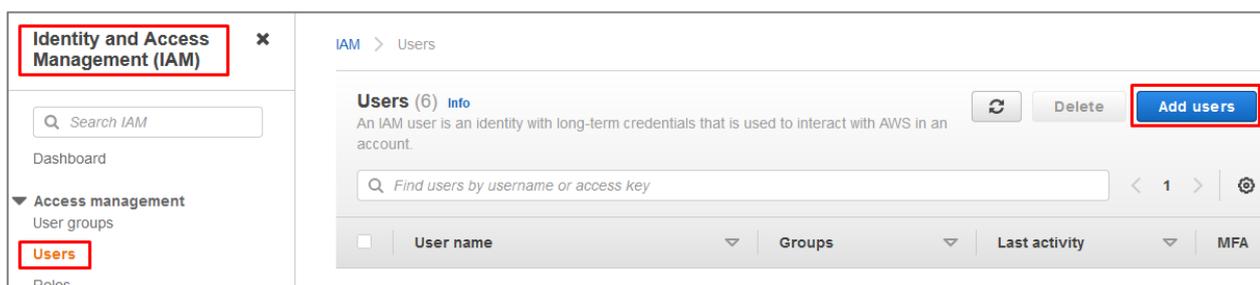
The UDS servers will be provided by the Virtual Cable team in disk image format (.ova).

▪ User creation in IAM module

To import the UDS components we will need to have a user account (where we have available the "**Access Key ID**" and the "**Secret Access Key**") with permissions, within the AWS IAM module. The necessary permits will be: "**IAMFullAccess**", "**AmazonEC2FullAccess**" and "**AmazonS3FullAccess**".

If you do not have any previously created user or you wish to configure a specific one to be used by UDS (recommended), the procedure shown below will be carried out:

Access the IAM module in our AWS environment (where we will have to have full permissions), select from the menu "**Access management**" the section "**users**" and "**add users**":



In the new user creation wizard, indicate a name and select "**access key - programmatic access**":

Add user 1 2 3 4 5

Set user details

You can add multiple users at once with the same access type and permissions. [Learn more](#)

User name*

+ Add another user

Select AWS access type

Select how these users will primarily access AWS. If you choose only programmatic access, it does NOT prevent users from accessing the console using an assumed role. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

Select AWS credential type* **Access key - Programmatic access**
Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.

Password - AWS Management Console access
Enables a **password** that allows users to sign-in to the AWS Management Console.

required
Cancel
Next: Permissions

In the next step of the wizard we will configure the necessary permissions that the user must have. We can create a group with specific permissions or assign them directly.

The permissions that the user must have will be: *"IAMFullAccess"*, *"AmazonEC2FullAccess"* and *"AmazonS3FullAccess"*.

The following screenshot shows how to assign them directly, using the option *"Attach existing policies directly"*. Through the policy search engine, we mark *"IAMFullAccess"*, *"AmazonEC2FullAccess"* and *"AmazonS3FullAccess"*:

Add user 1 2 3 4 5

Set permissions

 Add user to group

 Copy permissions from existing user

 Attach existing policies directly

Create policy ↻

Filter policies Showing 1 result

	Policy name	Type	Used as
<input checked="" type="checkbox"/>	 IAMFullAccess	AWS managed	Permissions policy (2)

Add user 1 2 3 4 5

▼ Set permissions

 Add user to group

 Copy permissions from existing user

 Attach existing policies directly

Create policy ↻

Filter policies Showing 1 result

	Policy name	Type	Used as
<input checked="" type="checkbox"/>	▶ AmazonEC2FullAccess	AWS managed	Permissions policy (4)

Add user 1 2 3 4 5

▼ Set permissions

 Add user to group

 Copy permissions from existing user

 Attach existing policies directly

Create policy ↻

Filter policies Showing 1 result

	Policy name	Type	Used as
<input checked="" type="checkbox"/>	▶ AmazonS3FullAccess	AWS managed	Permissions policy (2)

Follow the user creation wizard and check that all the data is correct:

Add user

1 2 3 4 5

Review

Review your choices. After you create the user, you can view and download the autogenerated password and access key.

User details

User name	UDS-import
AWS access type	Programmatic access - with an access key
Permissions boundary	Permissions boundary is not set

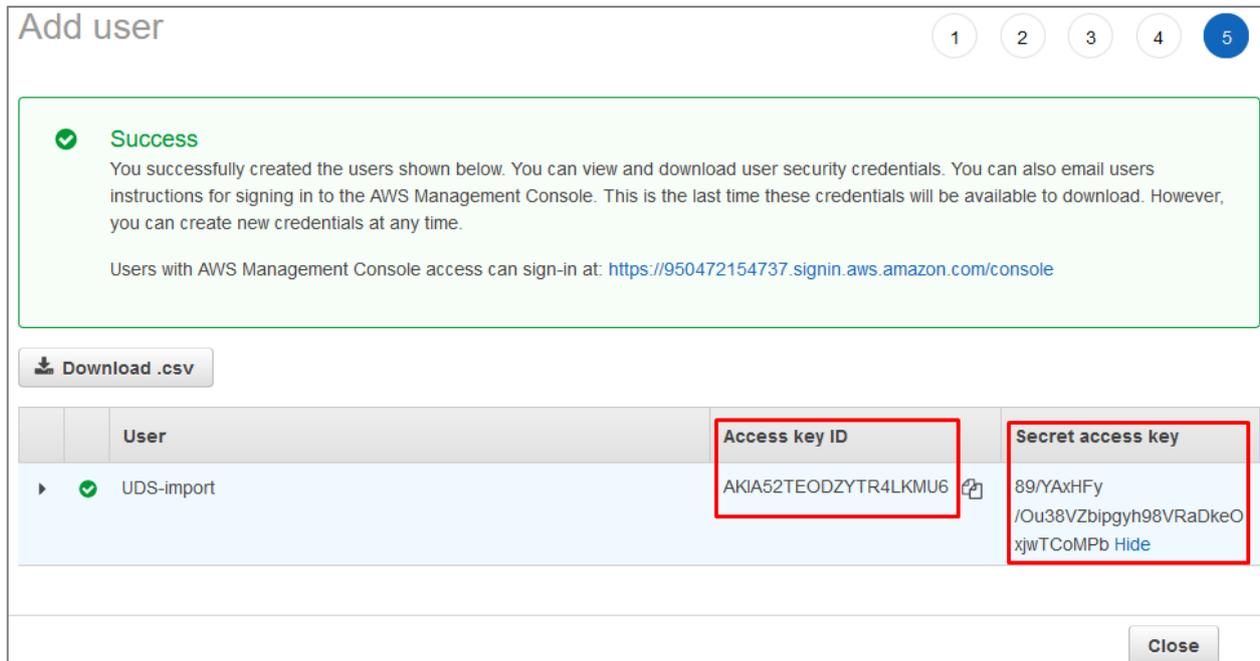
Permissions summary

The following policies will be attached to the user shown above.

Type	Name
Managed policy	AmazonEC2FullAccess
Managed policy	AmazonS3FullAccess
Managed policy	IAMFullAccess

[Cancel](#) [Previous](#) [Create user](#)

Proceed to create the new user with the assigned permissions. At this point it will be very important that we copy the user data: "**Access key ID**" and "**Secret Access key**" (especially the latter, since once the wizard window is closed this data will no longer be available, although it will be possible to generate a new "**Secret Access key**" if necessary).



Add user 1 2 3 4 5

Success
You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.

Users with AWS Management Console access can sign-in at: <https://950472154737.signin.aws.amazon.com/console>

[Download .csv](#)

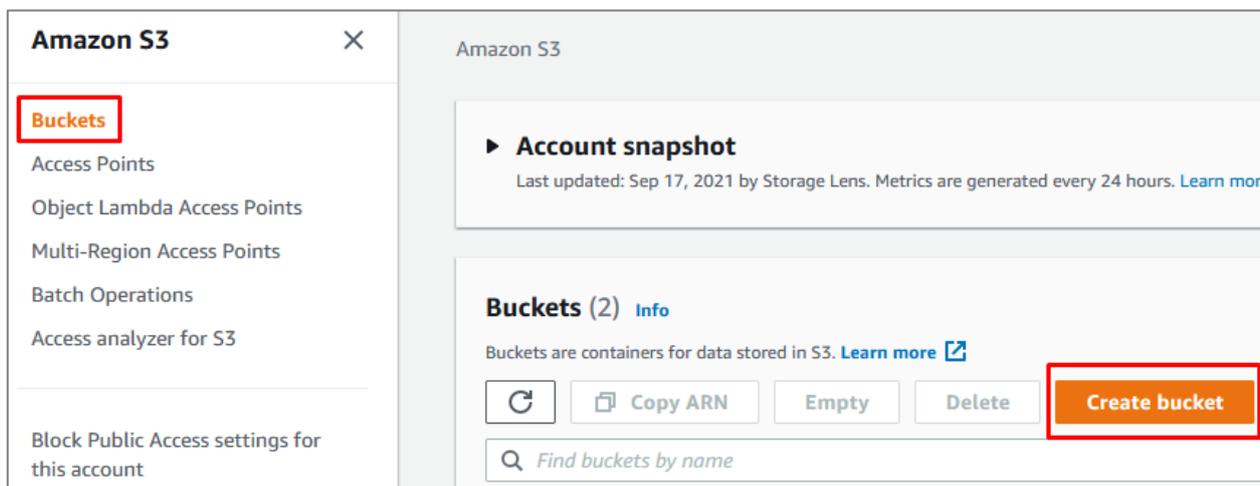
User	Access key ID	Secret access key
UDS-import	AKIA52TEODZYTR4LKMU6	89/YAxHFy /Ou38VZbipgyh98VRaDkeO xjwTCoMPb Hide

[Close](#)

■ Bucket and Role Creation

Once we have a user and their connection data, we must create a new bucket from the module "S3" of the AWS environment so that it can later be modified to allow it to contain the UDS components.

Access Amazon S3, we are located in "**buckets**" and click on "**create bucket**":



Amazon S3 X

Buckets

- Access Points
- Object Lambda Access Points
- Multi-Region Access Points
- Batch Operations
- Access analyzer for S3

Block Public Access settings for this account

Amazon S3

Account snapshot
Last updated: Sep 17, 2021 by Storage Lens. Metrics are generated every 24 hours. [Learn more](#)

Buckets (2) Info
Buckets are containers for data stored in S3. [Learn more](#)

[Refresh](#) [Copy ARN](#) [Empty](#) [Delete](#) [Create bucket](#)

In the wizard, we will indicate a name, select our region and leave the rest of the options by default:

Amazon S3 > Create bucket

Create bucket [Info](#)

Buckets are containers for data stored in S3. [Learn more](#) [↗](#)

General configuration

Bucket name

Bucket name must be unique and must not contain spaces or uppercase letters. [See rules for bucket naming](#) [↗](#)

AWS Region

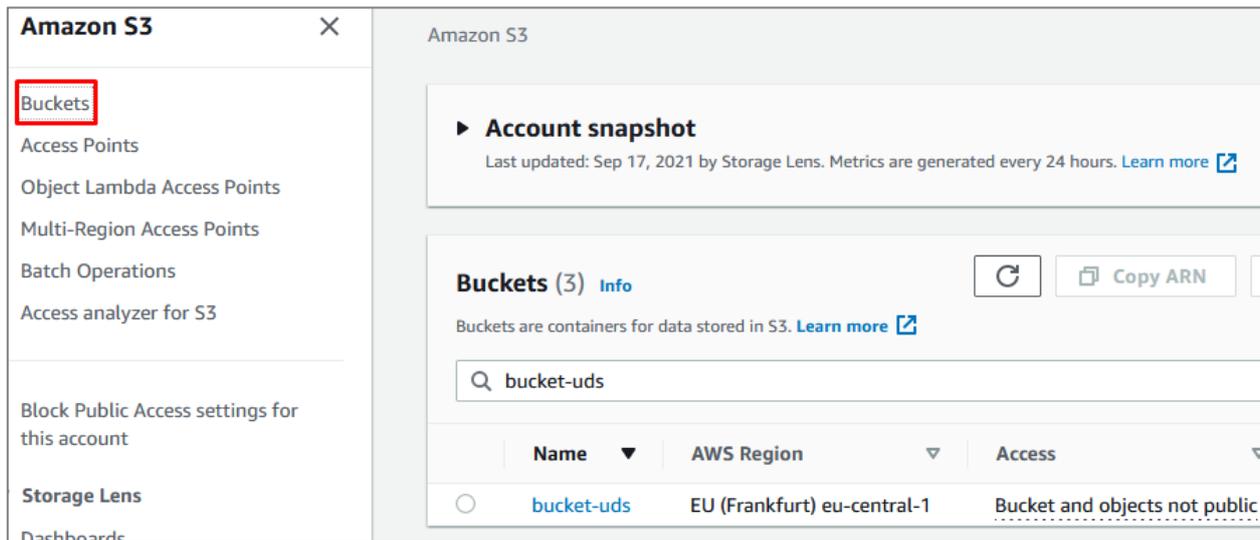
Copy settings from existing bucket - optional
Only the bucket settings in the following configuration are copied.

Object Ownership [Info](#)

Control ownership of objects written to this bucket from other AWS accounts and granted using access control list ownership determines who can specify access to objects.

<input checked="" type="radio"/> ACLs disabled (recommended) All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.	<input type="radio"/> ACLs enabled Objects in this bucket can be owned by other accounts. Access to this bucket and its objects is specified using ACLs.
--	--

Create the bucket that will host the UDS servers:



Now we will run an application that will allow us to create a role and modify the previously created bucket in the AWS environment, with all the necessary settings and permissions to import the UDS components.

You must download the following application:

https://images.udsenderprise.com/files/AWS/UDS_Import/setup_vmimport_role.zip

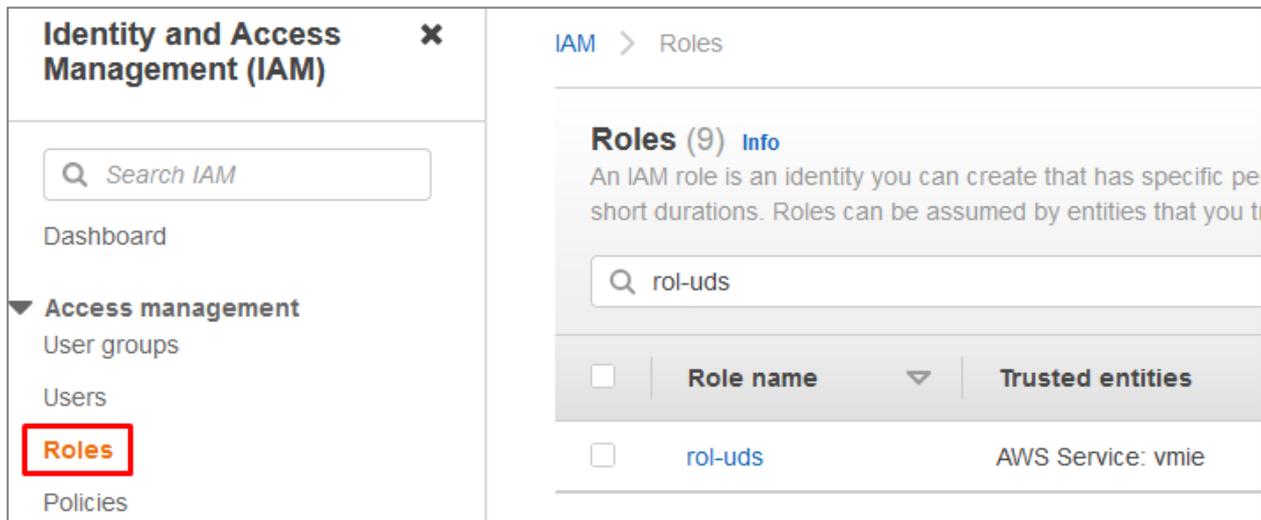
Extract the .zip file and execute it by command line on a computer with Windows OS with the following parameters:

- **-to**: Access key of the user with permissions indicated in the previous point.
- **-s**: Secret Access Key of the user.
- **-b**: Name of "*bucket*" that will be created in the AWS environment (S3) and that will serve us to store the UDS servers.
- **-n**: Name of the role that will be created in the AWS environment (IAM) to allow the import of UDS servers.

```

C:\Windows\system32\cmd.exe
C:\>
C:\>setup_vmimport_role.exe -a AKIA52TEODZYTR4LKMU6 -s 89/YAxHFy/Ou38VZbipgyh98VRaDkeOxjwTCoMPb
-b bucket-uds -n rol-uds
  
```

Once the command has been executed, we can see how the role has been created in the IAM module of the AWS environment:



■ Import UDS servers

To import the UDS components, we must have their images in .ova format:



Once downloaded, we will execute an application that will upload the UDS component to the indicated bucket in the AWS environment.

You must download the following application:

https://images.udsenderprise.com/files/AWS/UDS_Import/import_uds_appliance.zip

Extract the .zip file and execute it by command line on a computer with Windows OS with the following parameters:

- **-to:** Access key of the user with permissions indicated in the previous point.
- **-s:** Secret Access Key of the user.
- **-b:** Name of the bucket that will be created in the AWS environment (S3) and that will be used to store the UDS servers.

- **-n:** Name of the role that will be created in the AWS environment (IAM) to allow the import of UDS servers.
- **-F:** Path of the UDS component to import (you can also indicate cloud repositories, for example: -f <https://images.udsenderprise.com/3.5/stable/aws/UDS-Server-aws.3.5.0.ova>).

```
C:\Users\Javier Gomez\Desktop>setup_vmimport_role.exe -a AKIA52TE0DZY3F2HCEY -s 69u5XusKwx3k7KMEhPwrW6PoDT7+32VvxJD7nqjN -b bucket-uds -n rol-uds -f "C:\Users\Javier Gomez\Downloads\UDS-Server-aws.3.6.0.ova"
```

Once executed, we will wait for it to upload:

```
C:\>import_uds_appliance.exe -a AKIA52TE0DZY3F2HCEY -s 69u5XusKwx3k7KMEhPwrW6PoDT7+32VvxJD7nqjN -b bucket-uds -n rol-uds -f "C:\Users\Javier Gomez\Downloads\UDS-Server-aws.3.6.0.ova"
Uploading UDS-Server-aws.3.6.0.ova [=====] 15%
```

And the machine is imported:

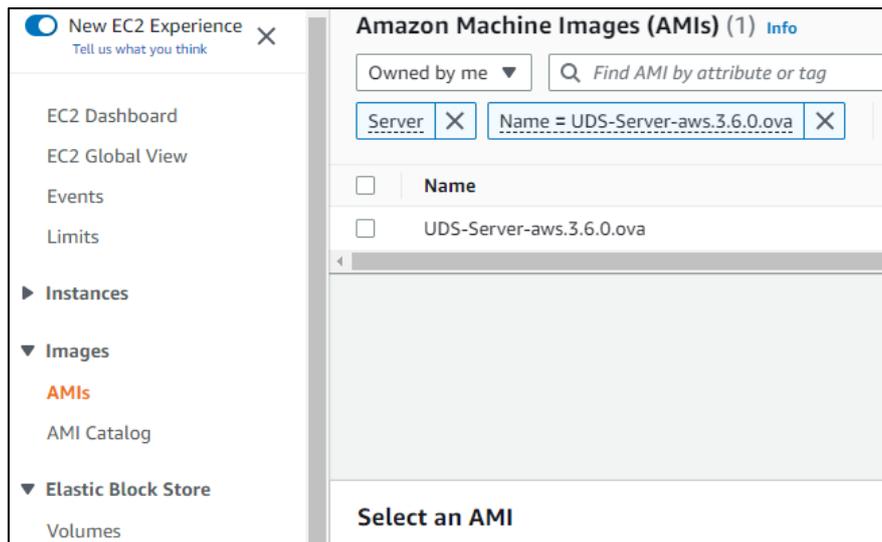
```
C:\>import_uds_appliance.exe -a AKIA52TE0DZY3F2HCEY -s 69u5XusKwx3k7KMEhPwrW6PoDT7+32VvxJD7nqjN -b bucket-uds -n rol-uds -f "C:\Users\Javier Gomez\Downloads\UDS-Server-aws.3.6.0.ova"
Uploading UDS-Server-aws.3.6.0.ova [=====] 100%
Task ID: import-ami-04815ed78c8479844
Importing [=====] 19% | (State: converting)
```

This last phase of the process can take several minutes. At this point the server imported into the bucket is converted and generates an AMI.

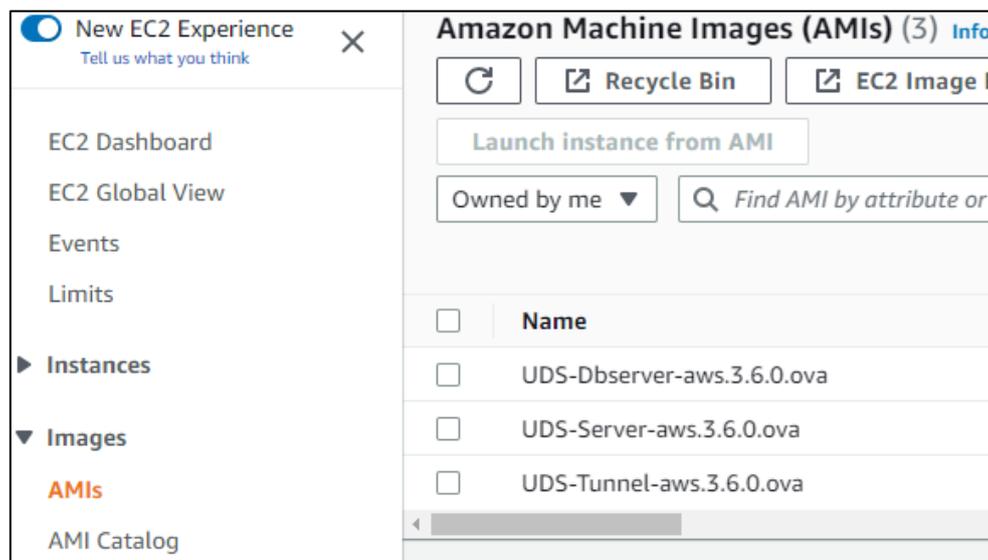
Once the process is finished, we will have the UDS server as AMI:

```
C:\>import_uds_appliance.exe -a AKIA52TE0DZY3F2HCEY -s 69u5XusKwx3k7KMEhPwrW6PoDT7+32VvxJD7nqjN -b bucket-uds -n rol-uds -f "C:\Users\Javier Gomez\Downloads\UDS-Server-aws.3.6.0.ova"
Uploading UDS-Server-aws.3.6.0.ova [=====] 100%
Task ID: import-ami-00b9736ef64cfe053
Importing [=====] 58% - (State: preparing ami)
File "UDS-Server-aws.3.6.0.ova" deleted from s3 bucket bucket-uds
AMI ID: ami-00d5f5e6e01582ec4
AMI name set to "UDS-Server-aws.3.6.0.ova"
Done
C:\>
```

It will be available in the panel **"EC2"** from the AWS environment:



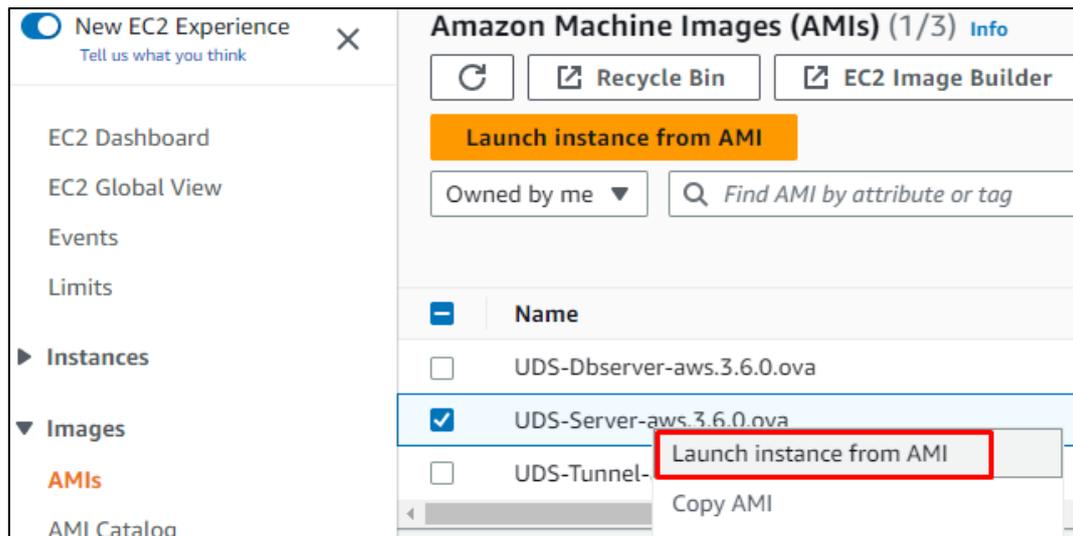
This import process must be repeated with the rest of the UDS components:



- Creation of UDS servers

The next step in the deployment of the UDS components is to create the virtual instance that will contain the UDS servers, based on the AMIs imported in the previous step.

Inside the module "EC2" from the AWS environment, we access the section "To my", we select the UDS component and click on "Launch instance from image".



In the instance creation wizard we must choose the appropriate type for each UDS component. The minimum requirements will be for all the components (Dbserver, UDS-Server and UDS-Tunnel) of 2vCPUs and 2 GB of RAM.



In the next step of the wizard we will configure the instance details. At least we must indicate a valid network and subnet, which allow us to communicate with other elements, and assign a public IP to have access to the outside.

Besides, create or select a "Security Group" with the necessary rules for each UDS component. Only the UDS-Server and UDS-Tunnel servers will need entry rules:

- **UDS server.** Port: 80/443 (for user and administrator access).
- **Tunnel server.** Port: 443 and 10443 (443 for tunneled connections and 10443 for HTML5 connections)
- **dbserver server:** We can allow all the traffic, since it will not be exposed to the internet.

▼ Network settings [Info](#)

VPC - *required* [Info](#)

vpc-0ef3cc126fb2dcb49 (VPC-10-16)
10.0.0.0/16

Subnet [Info](#)

subnet-062bacaefd3fa0088 Public subnet
VPC: vpc-0ef3cc126fb2dcb49 Owner: 950472154737
Availability Zone: eu-central-1c IP addresses available: 250 CIDR: 10.0.0.0/24

Auto-assign public IP [Info](#)

Enable

Firewall (security groups) [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group Select existing security group

Common security groups [Info](#)

Select security groups

UDS Server 3.6 sg-0b10a1aa6bbdd8f8a X
VPC: vpc-0ef3cc126fb2dcb49

Compare security group rules

Security groups that you add or remove here will be added to or removed from all your network interfaces.

▶ Advanced network configuration

NOTE:

The database server will not need to have a public IP.

In step 4 of the wizard indicate the type of storage:

▼ Configure storage [Info](#) [Advanced](#)

1x GiB Root volume (Not encrypted)

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage X

Add new volume

0 x File systems [Edit](#)

Check that all the data is correct and launch the instance:

▼ **Summary**

Number of instances [Info](#)

Software Image (AMI)

AWS-VMImport service: Linux - ...[read more](#)
ami-00d5f5e6e01582ec4

Virtual server type (instance type)

t2.medium

Firewall (security group)

UDS Server 3.6

Storage (volumes)

1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet. ✕

[Cancel](#) [Launch instance](#)

[Review commands](#)

Don't forget to choose a "key pair":

▼ Key pair (login) [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

▼

↻ Create new key pair

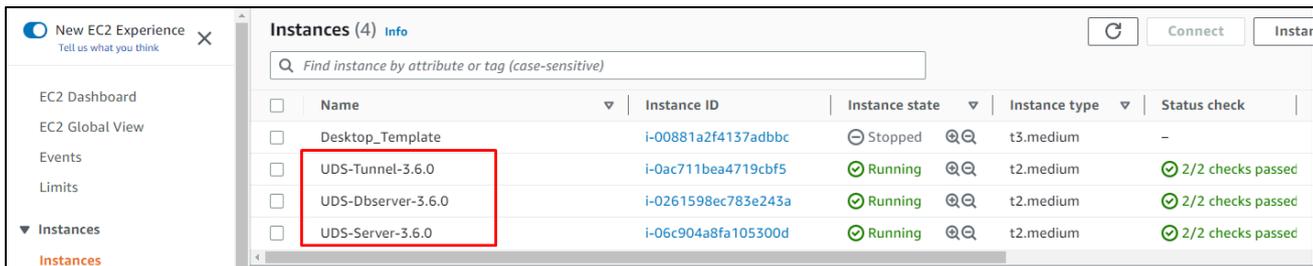
Once the instance is launched, we can access the section "instances" of the module "EC2" from the AWS environment and view the creation of the UDS component:

<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check
<input type="checkbox"/>	Desktop_Template	i-00881a2f4137adbcc	⊖ Stopped	t3.medium	-
<input type="checkbox"/>	UDS-Server-3.6.0	i-06c904a8fa105300d	⌚ Pending	t2.medium	-

Once started, indicate a descriptive name to the server:

<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type
<input type="checkbox"/>	Desktop_Template	i-00881a2f4137adbbc	Stopped	t3.medium
<input type="checkbox"/>	UDS-Server-3.6.0	i-06c904a8fa105300d	Running	t2.medium

Repeat the process with all UDS servers:



<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check
<input type="checkbox"/>	Desktop_Template	i-00881a2f4137adbbc	Stopped	t3.medium	-
<input type="checkbox"/>	UDS-Tunnel-3.6.0	i-0ac711bea4719cbf5	Running	t2.medium	2/2 checks passed
<input type="checkbox"/>	UDS-Dbserver-3.6.0	i-0261598ec783e243a	Running	t2.medium	2/2 checks passed
<input type="checkbox"/>	UDS-Server-3.6.0	i-06c904a8fa105300d	Running	t2.medium	2/2 checks passed

■ Configuration of UDS servers

Once we have all the instantiated UDS components, we will proceed with their configuration.

In this configuration example, we will rely on a virtual machine deployed on the same network as the UDS servers to have direct connectivity with them.

○ Database configuration

If you are using the database provided by the Virtual Cable team, it will already be pre-configured and you will only have to verify that you have IP connectivity (by default the network is configured by DHCP).

The default server credentials are:

- **User:**root
- **Password:**You

This server has created a database instance ready to be used with UDS Enterprise with the following data:

- **Instance name:**You
- **User:**You
- **Password:**You

By default, the server has its network settings via DHCP. It is recommended to always use static addressing in all UDS components.

- o UDS Server configuration

The UDS-Server component is the main element of the UDS environment. It has a configuration wizard accessible via web browser. Before accessing said configuration wizard, we will need to confirm that the server has been assigned an IP address via DHCP.

Once we know the IP address assigned to the server (selecting the instance, in the section "**Private IPv4 addresses**"), we will access via browser the IP address of the UDS server with port 9900

https://IP_Server:9900



From here indicate all the necessary data (IP data, serial to activate the subscription, credentials, etc...) to configure the server.

For more information on the configuration of the UDS server, consult the UDS Enterprise installation, administration and user manual available in the section [Documentation](#) from the web.

NOTE:

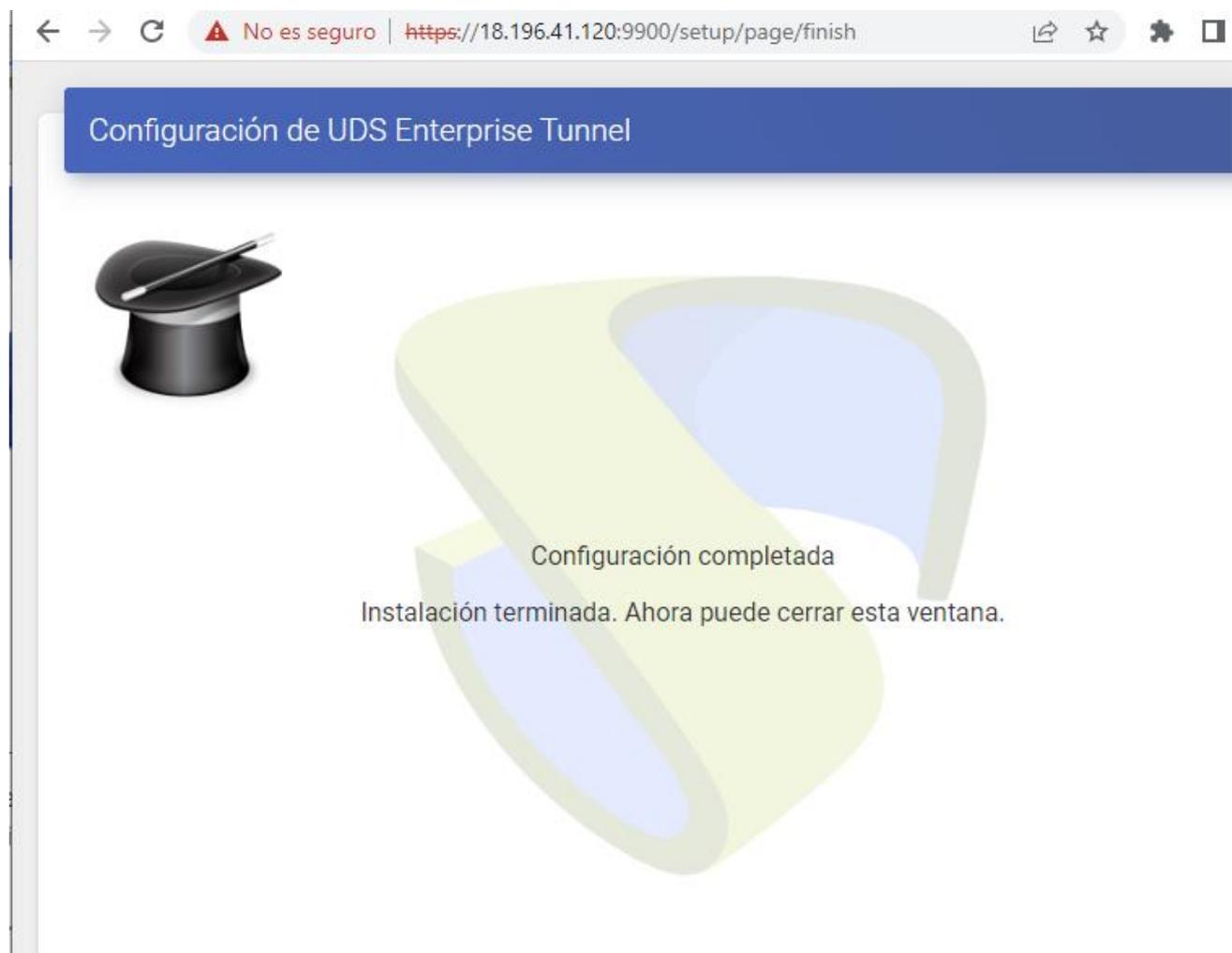
During the configuration procedure of the wizard, it will request the configuration data of the database server. In the case of using an external server, we must indicate the data of the previously configured database server (IP address, instance, user and password).

- o UDS Tunnel Configuration

The UDS Tunnel component is the element that will provide us with secure access to virtual desktops through the Internet. It will also take care of establishing the HTML5 connection (HTML5 Transport for desktops and vApps). It has a configuration wizard accessible via web browser. Before accessing said configuration wizard we will need to confirm that the server has been assigned an IP address.

Once we know the IP address assigned to the server (selecting the instance, in the section "**Private IPv4 addresses**"), we will access via browser the IP address of the UDS server with port 9900

`https://IP_Tunnel:9900`



From here we will indicate all the necessary data (IP data, credentials, certificates, etc...) to configure the server.

For more information on the configuration of the UDS Tunnel server, consult the UDS Enterprise installation, administration and user manual in the section [Documentation](#) from the web.

NOTE:

During the configuration procedure, the wizard will request the connection data of the UDS server.

- **Create base machines or templates on AWS**

For UDS to deploy virtual desktops on the AWS platform, it is necessary to have a base machine or template on which the new desktops self-generated by UDS will be based. This base machine can be deployed in different ways. Among them, it is possible to import an existing template on another platform (using the same applications that we have used to import the different UDS components) or rely on preconfigured machines (AMIs), which the AWS environment itself offers us.

If we choose to import a template, it is important that we make sure that it will have a valid access mode (SSH or RDP type), in order to access it once it is hosted on the AWS platform (this platform does not have a console to be able to manage, configure, and modify machines).

Another important point to take into account is the network configuration. It is necessary that it be configured to take IP addresses via DHCP.

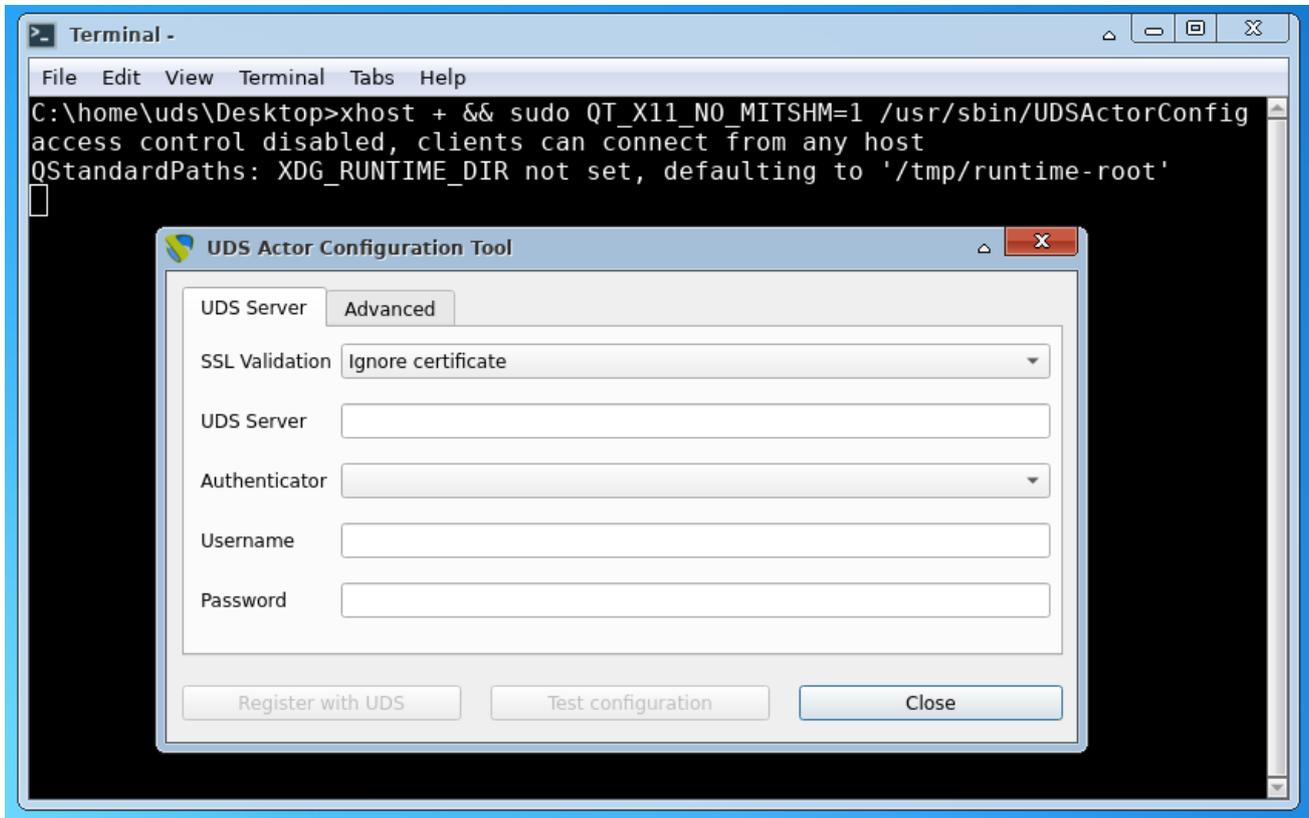
- **Base machine access and configuration**

Once the base machine or template is deployed and accessible via RDP, for example, we must install all the software that we need to have available in the virtual desktops deployed by UDS, perform the template optimization configurations, which are very important for the good performance of our machines (disable unnecessary services, startup time optimization, etc...) and finally install the UDS Actor.

NOTE:

You can consult the [UDS Enterprise Installation, Administration and User manual](#) in the section on [Documentation](#) from the UDS Enterprise website for more details on the installation of UDS Actor.

During the configuration of the UDS Actor we must indicate in the connection data against UDS Server the local DNS address/name or IP or public DNS depending on the type of deployment (in the case of using IP addresses instead of names, it must be ensured that these addresses are not dynamic and change when the servers are turned off/on).



NOTE:

If we want to view the configuration of the UDS Actor in an Ubuntu OS through RDP, we will have to execute the following command from a console:

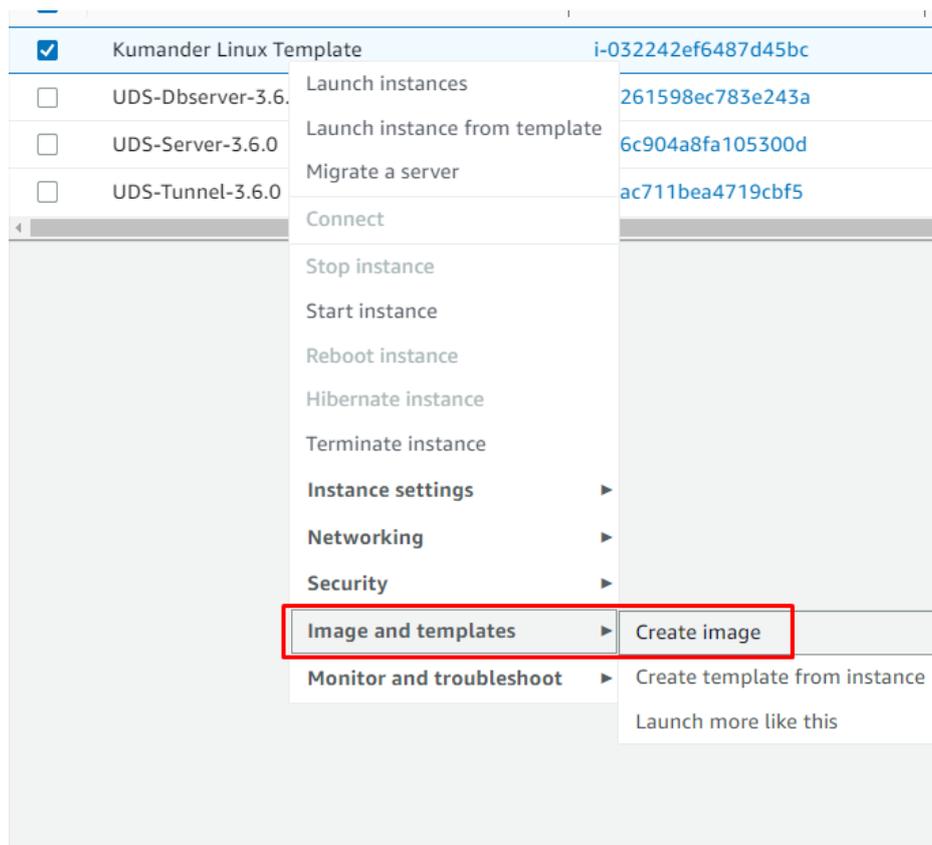
Ubuntu 18: `sudo QT_X11_NO_MITSHM=1 /usr/sbin/UDSActorConfig`

Ubuntu 20: `xhost + && sudo QT_X11_NO_MITSHM=1 /usr/sbin/UDSActorConfig`

- o AMI Creation

Once the configuration and installation of the UDS Actor has been completed, we can turn off the base machine or template and create the AMI that we will use in UDS to generate the virtual desktops to which the users will make the connection.

After turning off the base machine or template, we will select it, click on **"Actions"**, **"Image and templates"** and **"create image"**:



In the image creation wizard, we indicate a descriptive name for the AMI, (it will be the one that we see from the UDS administration console) and check the option **"Delete on termination"**. Click on **"Create Image"** to generate the AMI:

Instance ID
 i-032242ef6487d45bc (Kumander Linux Template)

Image name

 Maximum 127 characters. Can't be modified after creation.

Image description - optional

 Maximum 255 characters

No reboot
 Enable

Instance volumes

Storage type	Device	Snapshot	Size	Volume type	IOPS	Throughput	Delete on termination	Encrypted
EBS	/dev/...	Create new snapshot fr...	32	EBS General Purpose S...	100		<input checked="" type="checkbox"/> Enable	<input type="checkbox"/> Enable

NOTE:

To prevent orphaned volumes from being left on the platform, we must select the option “Delete on termination”.

After finishing the creation of the image, you will have it available in the AMIs section and we can rename it with a descriptive name:

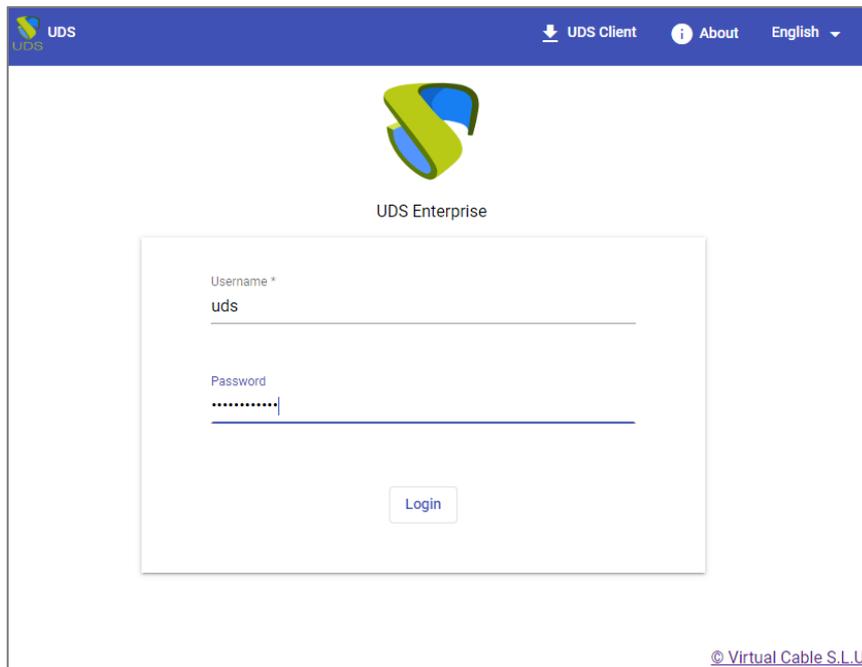
Name	AMI ID
<input type="checkbox"/> UDS-Server-aws.3.6.0.ova	ami-00d5f5e6e01582ec4
<input type="checkbox"/> UDS-Tunnel-aws.3.6.0.ova	ami-072cf809d1eeec68b
<input checked="" type="checkbox"/> Kumander-Linux-img	ami-01ba52eb27fac2512
<input type="checkbox"/> UDS-Dbserver-aws.3.6.0.ova	ami-0614f6c4b926e26cc
<input type="checkbox"/> KumanderLinux.ova	ami-094e281f9499a3906

Once you have the image (AMI) we can access the UDS administration to continue with the process of configuration and deployment of VDI services.

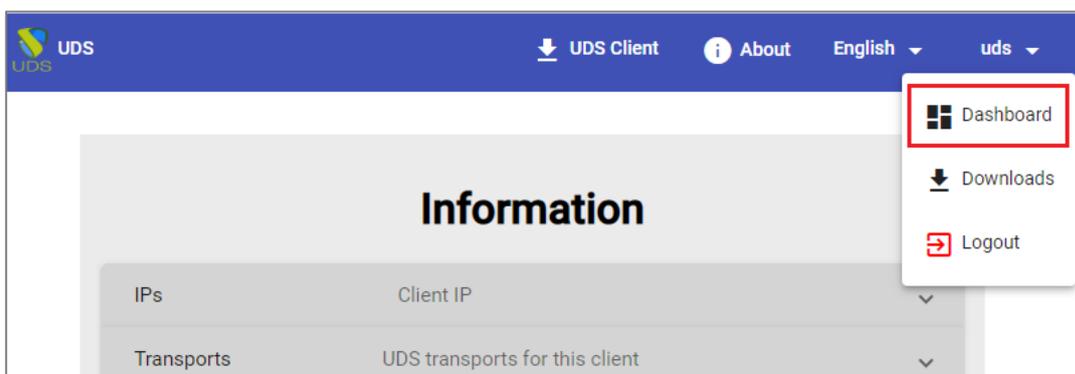
UDS Enterprise Administration

AWS service provider integration

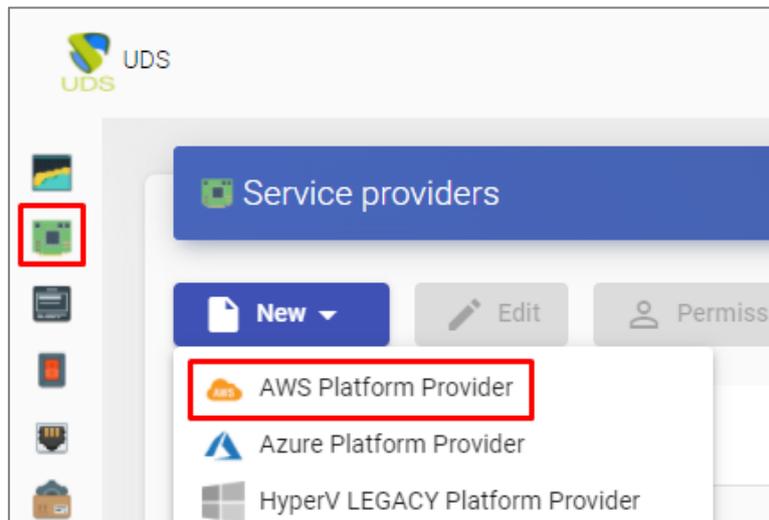
To integrate AWS as a UDS Enterprise service provider, we must access the UDS administration. To do this, we access the IP address or name of the UDS Server component via a web browser and authenticate ourselves with an administrator user (in the first access we will use the system administrator user indicated in the UDS server configuration wizard).



Once validated in the UDS login portal, we will access the "**dashboard**" from the user menu.



Within the UDS administration, we access the menu “*Services*” and click on “*New*” to register a new “*service provider*”. We select “*AWS Platform Provider*”:



In order for UDS to connect to the AWS platform and to be able to automatically deploy virtual desktops, it will be necessary to indicate the following information:

New provider

Main
Advanced

Tags

Tags for this element

Name *

Amazon Web Services

Comments

Comments for this element

Access Key ID *

Obtained from user created on AWS IAM for UDS Enterprise

Secret Access Key *

Obtained from user created on AWS IAM for UDS Enterprise - Keys

Test

Discard & close

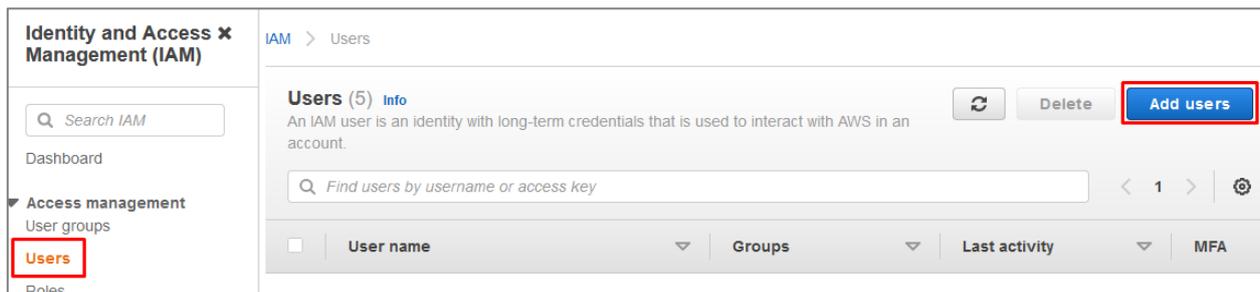
Save

- **Main:** We will indicate a descriptive name, and the connection data with the AWS account. To get a "**Access Key ID**" and a "**Secret Access Key**" it will be necessary to create a new user (or use an existing one) in the IAM module of the AWS console.

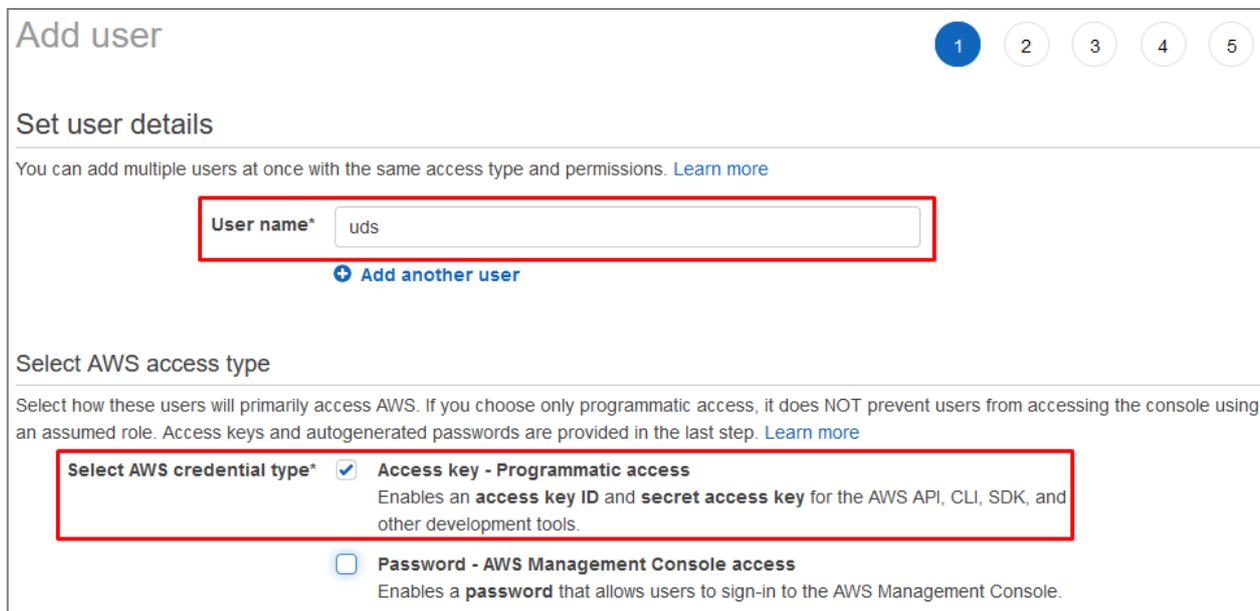
NOTE:

It is possible to use the same user that we have used to import the UDS machines, as long as we have all your data. In this example we will create a new user.

To create a new user, we access the IAM module and select in the menu "**users**" and click on "**add users**":



In the new user creation wizard, we indicate a name and we will select in "**Select AWS credential type**" the option "**Access key - Programmatic access**":



Add user 1 2 3 4 5

Set user details

You can add multiple users at once with the same access type and permissions. [Learn more](#)

User name* uds

[+ Add another user](#)

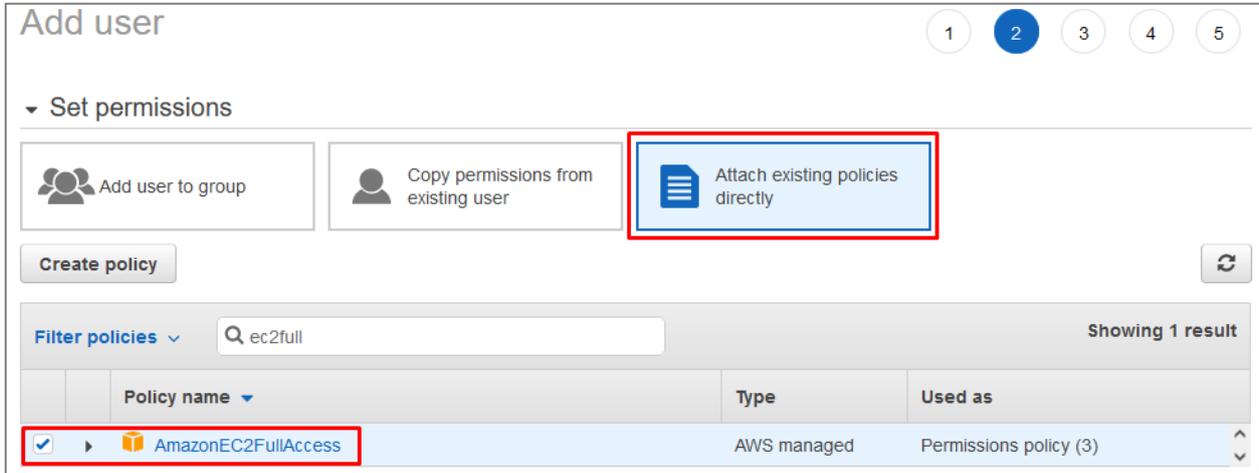
Select AWS access type

Select how these users will primarily access AWS. If you choose only programmatic access, it does NOT prevent users from accessing the console using an assumed role. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

Select AWS credential type*

- Access key - Programmatic access**
Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.
- Password - AWS Management Console access**
Enables a **password** that allows users to sign-in to the AWS Management Console.

In the next step of the new user creation wizard, we must select their permissions. We can assign the user to a group with the assigned permission: **"AmazonEC2FullAccess"** or directly assign this permission to the user as shown in the following screenshot:



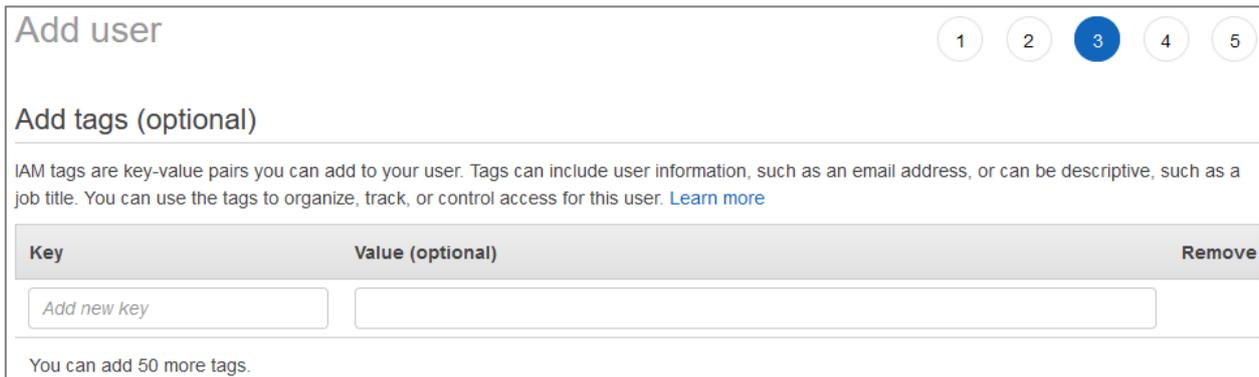
Add user 1 2 3 4 5

▼ **Set permissions**

Filter policies Showing 1 result

	Policy name	Type	Used as
<input checked="" type="checkbox"/>	AmazonEC2FullAccess	AWS managed	Permissions policy (3)

In step 3, if necessary, we could add tags for the user.



Add user 1 2 3 4 5

Add tags (optional)

IAM tags are key-value pairs you can add to your user. Tags can include user information, such as an email address, or can be descriptive, such as a job title. You can use the tags to organize, track, or control access for this user. [Learn more](#)

Key	Value (optional)	Remove
<input type="text" value="Add new key"/>	<input type="text"/>	<input type="button" value="Remove"/>

You can add 50 more tags.

Finally, check that all the data is correct and we proceed to the creation of the new user.

1 2 3 **4** 5

Add user

Review

Review your choices. After you create the user, you can view and download the autogenerated password and access key.

User details

User name	uds
AWS access type	Programmatic access - with an access key
Permissions boundary	Permissions boundary is not set

Permissions summary

The following policies will be attached to the user shown above.

Type	Name
Managed policy	AmazonEC2FullAccess

Tags

No tags were added.

Cancel Previous Create user

1 2 3 4 **5**

Add user

✔ **Success**

You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.

Users with AWS Management Console access can sign-in at: <https://950472154737.signin.aws.amazon.com/console>

Download .csv

User	Access key ID	Secret access key
▶ ✔ uds	AKIA52TEODZY3OHFILVM	***** Show

Once created, you will have the "*Access key ID*" and the "*Secret Access key*" to add it in the Service Provider settings.

	User	Access key ID	Secret access key
▶ ✓	uds	AKIA52TEODZY3OHFILVM	K7hCH+NdFQuhuOab52/k9+r67SRhqXkwyg+ZmkHf Hide

It must be taken into account that the "*Secret Access key*" we will not be able to consult it again, although we will be able to generate a new one.

New provider

[Main](#) [Advanced](#)

Tags
Tags for this element

Name *
Amazon Web Services

Comments
Comments for this element

Access Key ID *
AKIA52TEODZY3OHFILVM

Secret Access Key *
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

◀ ▶

[Test](#) [Discard & close](#) [Save](#)

- **Advanced:** indicate the creation and deletion concurrency, the connection timeout, if necessary a proxy server (for communication between the UDS and AWS server) and the region of our EC2 environment.

New provider

Main
Advanced

Creation concurrency *
30

Removal concurrency *
15

Timeout *
30

Proxy
Proxy used for connection to AWS (use PROTOCOL://host:port, i.e. htt

Default region *
eu-central-1

Test
Discard & close
Save

Perform a connection test with the service provider to confirm the correct integration and save.

New provider

Main
Advanced

Tags
Tags for this element

Name *
Amazon Web Services

Comments
Comments for this element

Access Key ID *
AKIA52TEODZY30HFILVM

Secret Access Key *
K7hCH+NdFQuhu0ab52/k9+r67SRhqXkwyg+ZmkHf

Test
Discard & close
Save

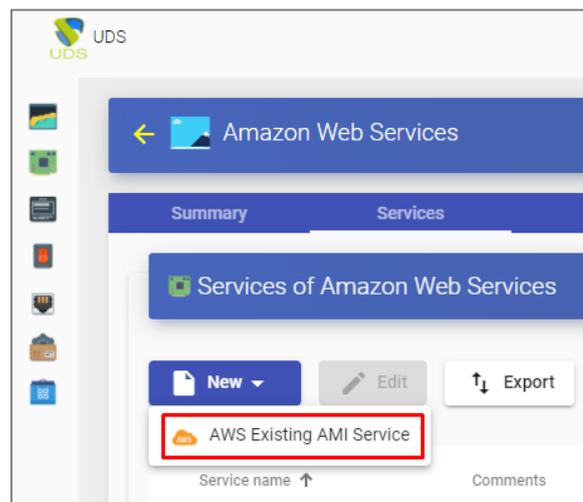
Test passed successfully
dismiss

NOTE:

Even if the test does not come out correct, we can save the provider and thus not lose the indicated data.

- Creation of base services

When we have a *service provider* valid connected with the AWS platform, we will be able to create services based on "Amazon Machine Images" (AMIs). To do this, access the provider (with double click or right button - "*details*") and in the tab "*Services*" click on "*New*" - "*AWS Existing AMI Service*".



To create a base service of type "*AWS Existing AMI Service*" we will need to indicate:

- o Main:
 - **Yam:** Friendly name of the base service.
 - **Region:** Location of the Amazon EC2 environment on which to work. All Regions are supported.
 - **AMI:** Base machine image or template that we will use to deploy virtual desktops (with the UDS Actor installed and configured).
 - **Instance type :** Amount of resources that the virtual desktops automatically deployed by UDS will have (this list will show all the types of machines available. Therefore, it must be ensured that the type chosen is appropriate for the service to be deployed).
 - **Keypair:** A set of security credentials used to prove your identity when connecting to an Amazon EC2 instance.

- **Machine Names:** Name root that the virtual desktops generated by UDS will have.
- **Name Length:** Number of digits of the counter for UDS machines. These digits will be attached to the "*machine names*" to form the DNS name of the virtual desktops (with 1 digit you can create 9 machines, with 2, 99, with 3, 999, etc...).

New service

Main Network

Tags
Tags for this element

Name *
Kummander

Comments
Comments for this element

Region *
eu-central-1

AMI *
Kumander-Linux-img (Kummander-Linux)

Instance type *
t2.small (1 cpus, 2048 MB, i386,x86_64, 2.5 GHz)

Key pair *
UDS36 (a8:11:8a:87:28:85:70:7a:ce:0f:b7:c2:01:...

Machine Names *
Kumman-

- network:
 - **VPC:** Existing virtual network of the AWS environment and to which the virtual desktops will connect.
 - **Subnetwork:** Existing subnet to which the virtual desktops will connect.
 - **Safety Group:** Security group to be assigned to virtual desktops.

New service

Main
Network

VPC *

VPC-10-16 (10.0.0.0/16) ▼

Subnetwork *

Public subnet/subnet-062bacaefd3fa0088 (10.0.0.... ▼

Security groups *

VDI36 (RDP + SSH) ▼

Discard & close
Save

Save and You will already have a valid base service to automatically deploy virtual desktops in AWS:

Services of Amazon web Services

New ▼
Edit
Export
Delete

Filter 1 - 1 of 1

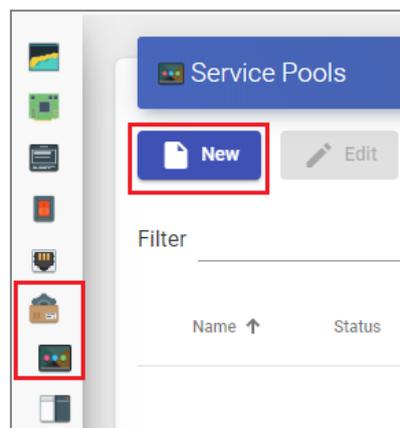
|<
<
>
>|
↻

Service name ↑	Comments	Type	Services Pools
<input type="checkbox"/> Kummander		AWS Existing AMI Service	0

Creation of Pool of Services

Before proceeding to create a pool of services (to publish virtual desktops), it will be necessary to have at least one **"Authenticator"** with user groups (to validate and be able to assign the service to users), a **"OS Manager"** (to indicate the OS and the persistence policy of the generated desktops) and a **"Transportation"** (to connect to the desktop) previously configured. To see more details on how to configure these elements, you can access the UDS Enterprise Installation, Administration and User Manual in the section on [Documentation](#) from our website.

When we have the elements mentioned above (**"Authenticator"**, **"OS Manager"** and **"Transportation"**) we can now create **"Service Pools"**. To do this, we access the section **"Pools"**, we open the tab **"Service Pools"** and click on **"New"**.



In the tab **"Main"** indicate the name of the service (this name will be visible to the users) and we will select the previously created base service (in this case the AWS platform and the xUbuntu20 base service) and a **"OS Manager"** existing (in this example one will be used for Linux OS and of non-persistent type).

New service Pool

< Main
Display >

Tags
Tags for this element

Name *
Kummander-Linux

Short name
Short name for user service visualization

Comments
Comments for this element

Base service
Amazon web Services\Kummander

OS Manager
OS Manager linux

Publish on creation
 Yes

Discard & close
Save

The parameters of the tabs **"Advanced"** and **"display"** you can leave them by default. In the tab **"Availability"** We will indicate the initial desktops that UDS will generate and the cache ones.

In this example we will indicate that UDS automatically creates 2 desktops and we always have at least 1 available in cache.

New service Pool

< Advanced
Availability >

Initial available services
2

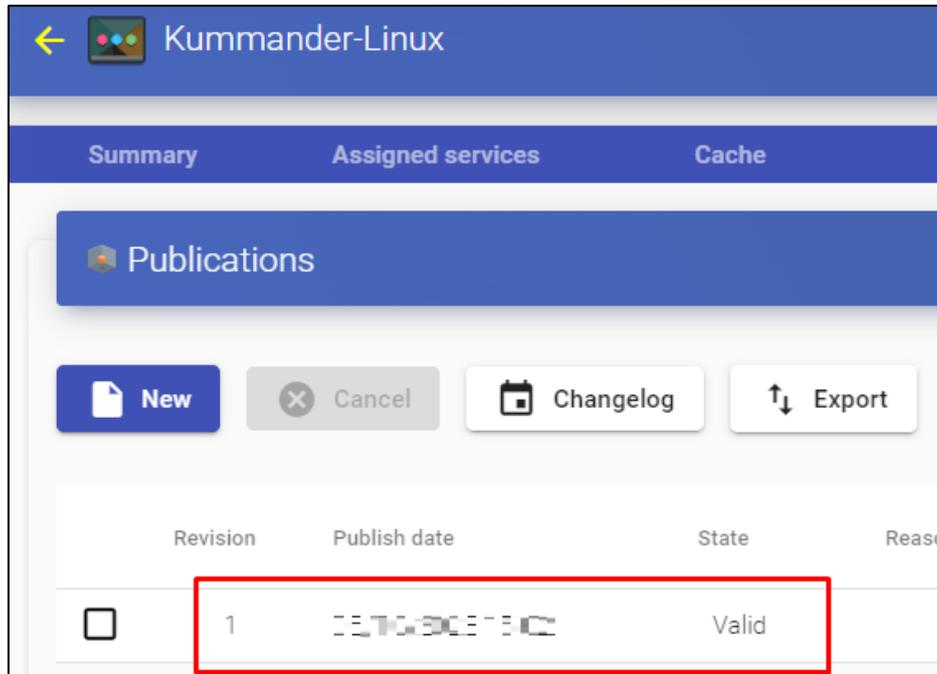
Services to keep in cache
1

Services to keep in L2 cache
1

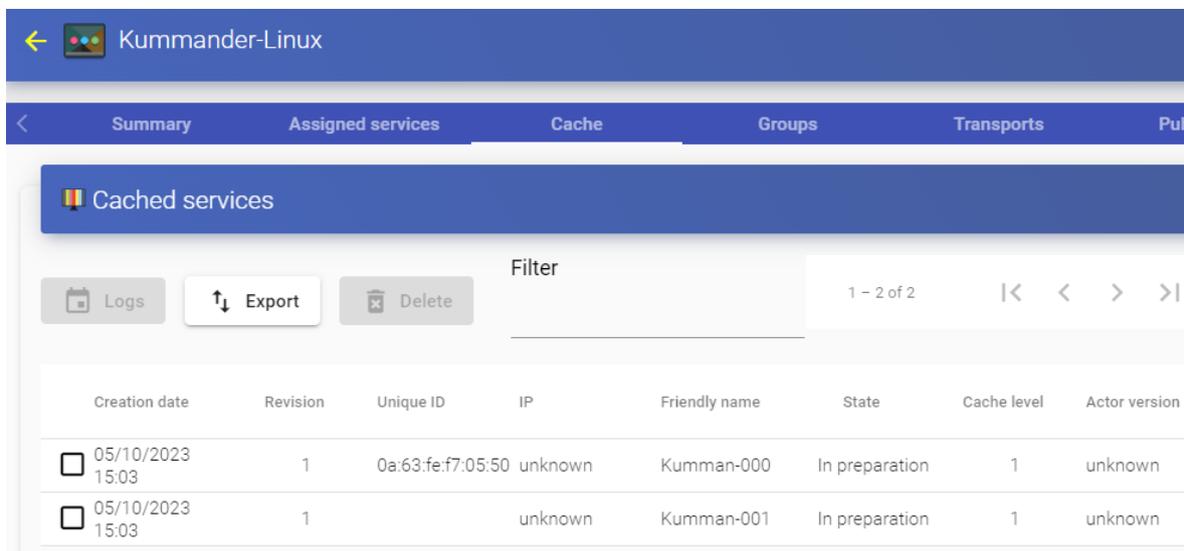
Maximum number of services to provide
10

Discard & close
Save

Selecting the **"Service Pool"** and opening the tab **"publications"** we will check if the publication has been generated correctly. When in a state **valid**, the system will start to auto-generate the virtual desktops indicated in the cache parameters.



In the tab **"Cache"** you will be able to see how the desktops begin to be generated.



In the AWS environment you will also see how virtual desktops are generated:

Instances (7) [Info](#)

Find instance by attribute or tag (case-sensitive)

<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type
<input type="checkbox"/>	Kumander Linux Template	i-032242ef6487d45bc	⊖ Stopped	t2.medium
<input type="checkbox"/>	Kumman-000	i-094fd0f5398042fb3	✔ Running	t2.small
<input type="checkbox"/>	Kumman-001	i-02bbc65f837027aab	✔ Running	t2.small
<input type="checkbox"/>	Kumman-002	i-0ac3cdc64544b7bbc	✔ Running	t2.small
<input type="checkbox"/>	UDS-Dbserver-3.6.0	i-0261598ec783e243a	✔ Running	t2.medium
<input type="checkbox"/>	UDS-Server-3.6.0	i-06c904a8fa105300d	✔ Running	t2.medium
<input type="checkbox"/>	UDS-Tunnel-3.6.0	i-0ac711bea4719cbf5	✔ Running	t2.medium

Once the desktops are in the state **“valid”** (that is, the UDS Actor installed in the template has finished applying the necessary configuration), they will be available for users to access.

← Kummander-Linux

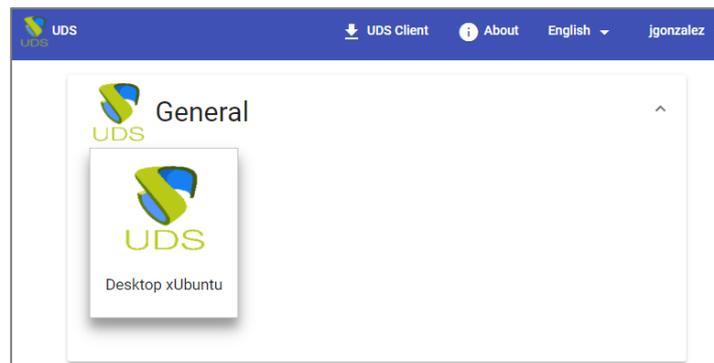
Summary Assigned services Cache Groups Transports

🇺🇦 Cached services

📅 Logs Export Delete Filter 1 - 3 of 3

<input type="checkbox"/>	Creation date	Revision	Unique ID	IP	Friendly name	State	Cache level	Actor version
<input type="checkbox"/>	2023-10-26 10:00:00	1	0a:63:fe:f7:05:50	10.0.0.183	Kumman-000	Valid	1	3.6.0
<input type="checkbox"/>	2023-10-26 10:00:00	1	0a:f7:8f:97:bf:92	10.0.0.5	Kumman-001	Valid	1	3.6.0
<input type="checkbox"/>	2023-10-26 10:00:00	1	0a:3c:88:cd:09:66	10.0.0.64	Kumman-002	In preparation	2	3.6.0

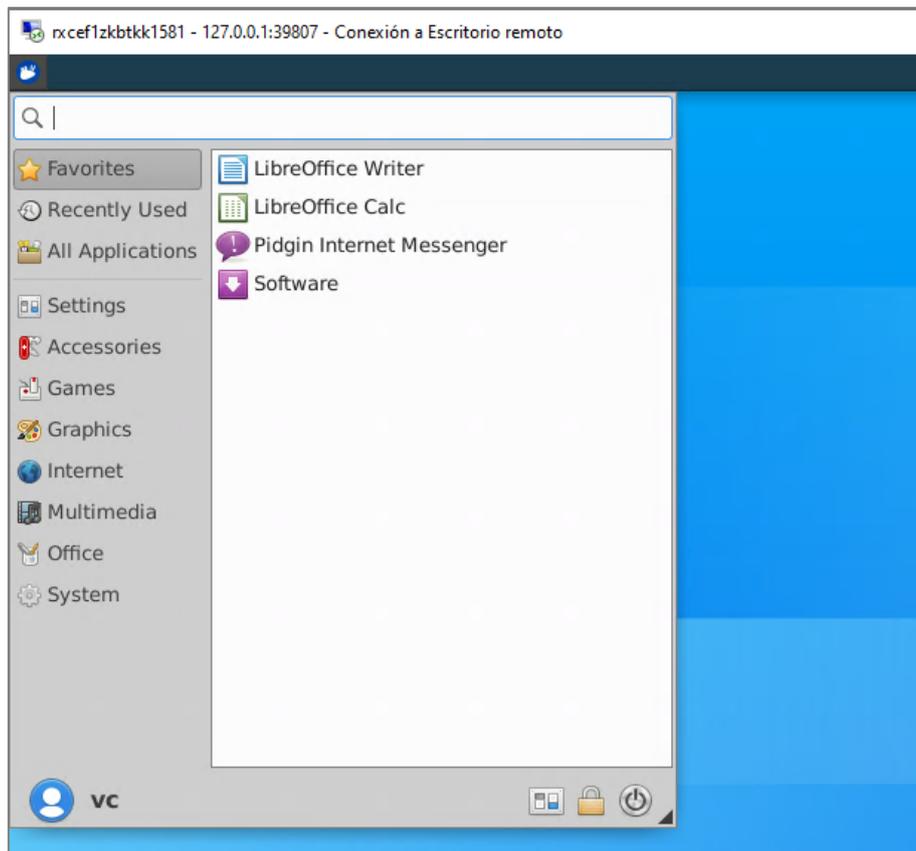
We will access the services window with a user (it is not possible to use the system administrator super-user) and we will see the available service.



NOTE:

In order for the service to be viewed by users, the “Service Pool” created must be assigned a user group (“Groups” tab) and a transport (“Transports” tab).

Access it by clicking on the image (in this example an RDP type transport has been configured).



NOTE:

If we are outside the VPC network configured in AWS, it will be necessary to use the tunneled transport (as can be seen in the screenshot of the connection example, it is connecting to 127.0.0.1 since the connection is made via Tunnel).

Monitoring UDS Enterprise

With UDS Enterprise you can use any third-party monitoring tool, for example you can use “[OpenUDS monitor](#)” or [LoginVSI](#) etc etc.

Common errors and troubleshooting

- Connection problems with UDS Login page:

If you can't access the UDS login page, the first thing that you verify is the state of the connection with the UDS Server. If there is no connection, you should verify the connection parameters.

- Creating AWS AMI:

It is necessary to have created the AMI of the template in order to be able to deploy it, if this step is not done, you won't be able to deploy any machines.

About VirtualCable

[Virtual Cable](#) is a company specialized in the digital transformation of the workplace. The company develops, supports and markets UDS Enterprise. Its team of experts has designed **VDI** solutions 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 in virtualization technologies. **Millions of Windows and Linux virtual desktops with UDS Enterprise are deployed all over the world every day.**