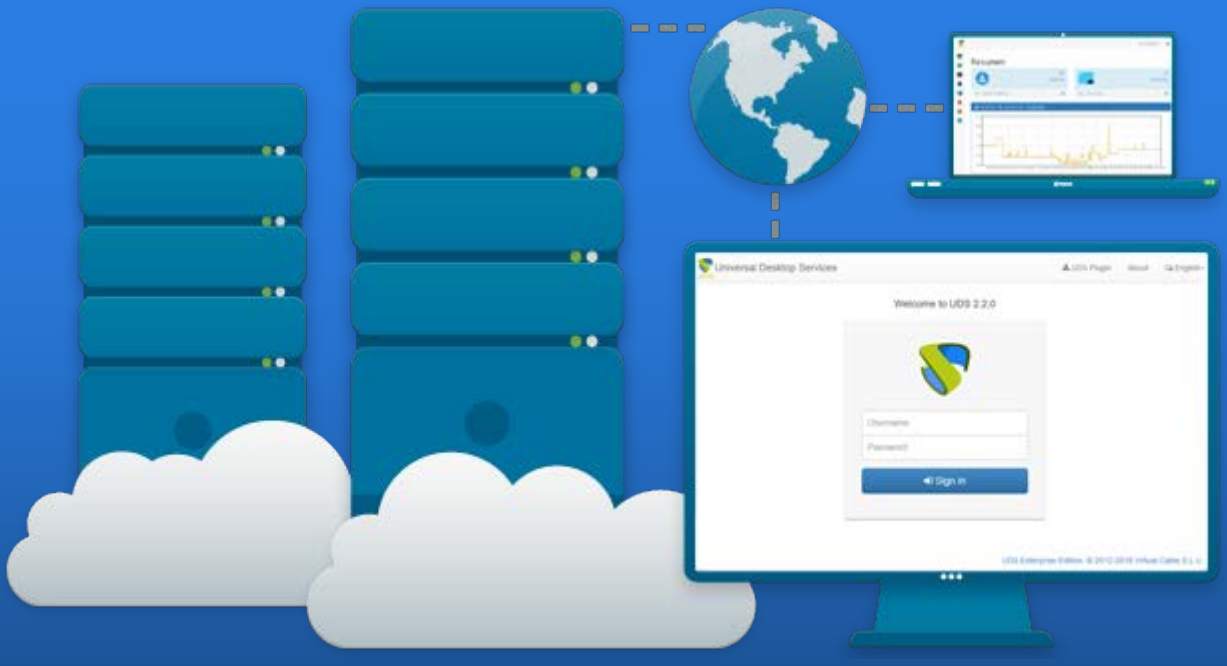
VDI with UDS Enterprise

&

Microsoft Azure



Introduction

Azure is a proprietary platform developed by Microsoft that offers services in the cloud. Some of its advanced features include the ability to run virtual machines, virtual applications, databases, backup copies and many other tasks. It integrates an infinity of services in the cloud that are necessary to develop, test, implement and manage virtual machines (VM).

The **VDI with UDS Enterprise & Microsoft Azure** guide will help you learn the procedure to implement and configure the UDS Enterprise components on such platform. This document shows real examples on how to create resource groups, storage accounts, containers and the necessary resources so that UDS Enterprise can implement virtual desktops in this platform.

In addition, we detail one of the procedures to create virtual machines (to be used as a machine or base template), the steps to migrate machines from an existing environment (VMware, Hyper-V, etc.) to Microsoft Azure and the simplest one: how to convert a VM disk to .vhd format (disk format recognized by Azure).

UDS Enterprise in Microsoft Azure

Before performing the integration, it is advisable to invest time in knowing the parts related to UDS Enterprise (for more information, visit our website, in the Documentation section you will find the Installation, administration and user manual of UDS Enterprise). Two of them are the **Service Providers** and the **Authenticators**, elements of great importance for the configuration of Azure in UDS Enterprise.

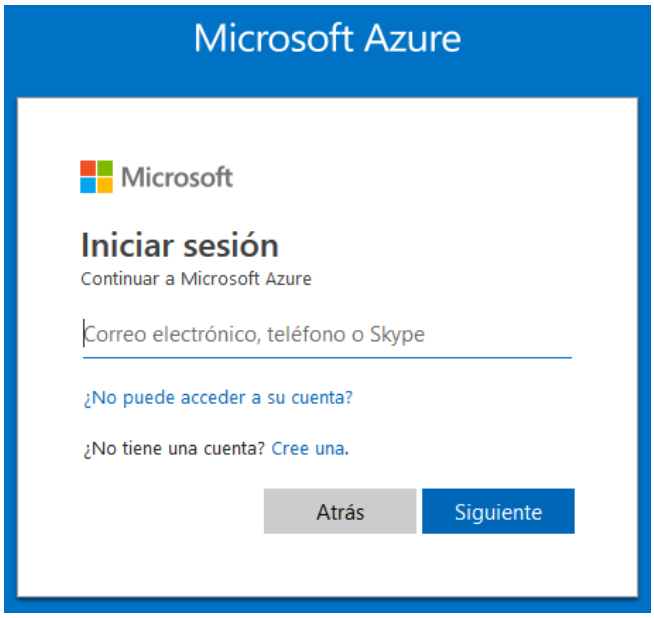
UDS Enterprise will allow the deployment of self-generated virtual desktops and virtual application sessions on the Microsoft Azure platform.

To install and configure UDS Enterprise you should request the Appliances (MySQL, Server and Tunneler) and the serial number (Free / Evaluation / Enterprise) to VirtualCable.

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

Where do I start?

First, you must have an account with administrator privileges on the Azure platform. If you already have it, log in to the portal.



When you log in and before loading the UDS Enterprise Appliances, you will need a series of items available on the Azure platform (Resource Groups, Storage Accounts, Container, Network Security Groups).

Below, we have included examples on how to create and configure these elements for the proper functioning of UDS Enterprise on an Azure platform.

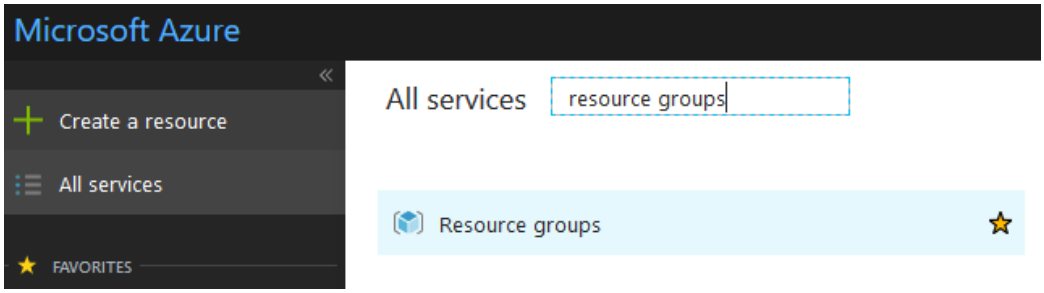
Required elements in Azure

- Resource Groups

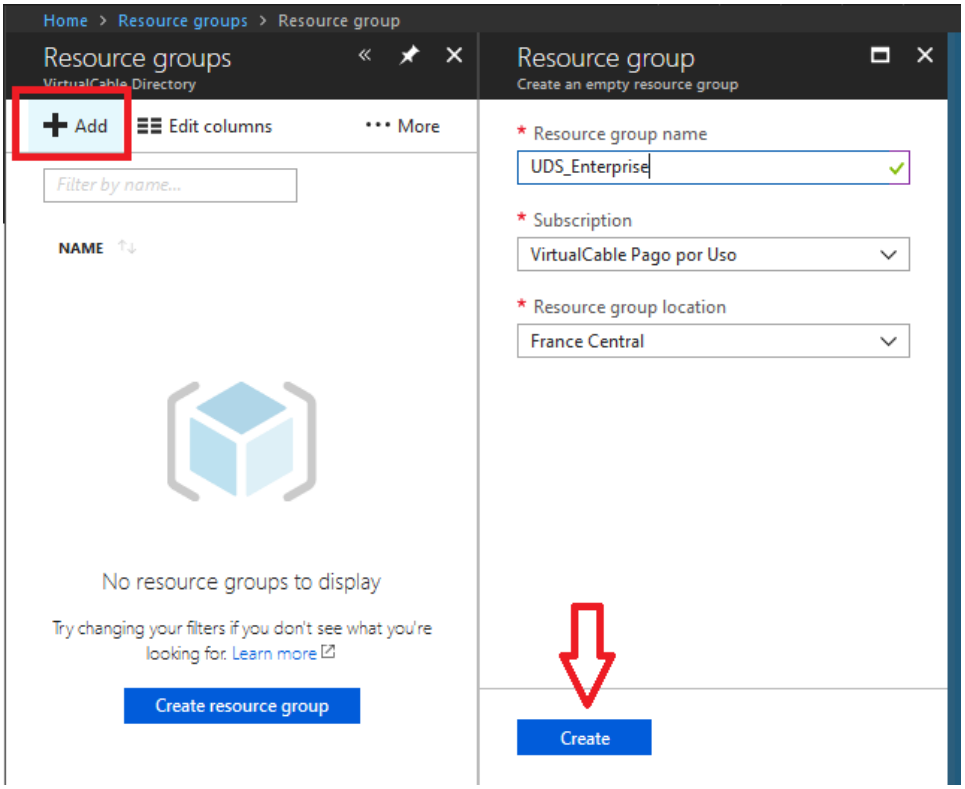
A “Resource Group” in Azure groups a collection of assets into logical groups for provisioning, monitoring and access control easily or even automatically, for more effective administration.

We will need to have at least one "Resource Group" on which deploying and configuring all UDS Enterprise requirements and components. If you do not have one, you can create it following these steps:

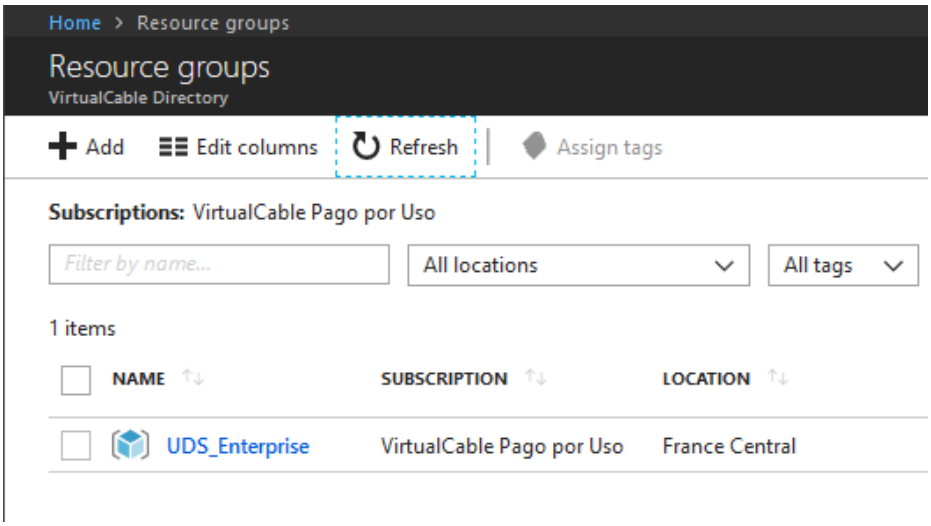
1. In the list of “Services”, we look for “Resources Group” (it is recommended to add it to your favorites list) and click on it.



2. Once inside, click “add” to create a new one. We indicate a descriptive name, we select the subscription in which it will be registered and we select a "Location of the group of resources". Finally, click on “Create”.



3. We confirm that the “Resource Group” has been correctly created.

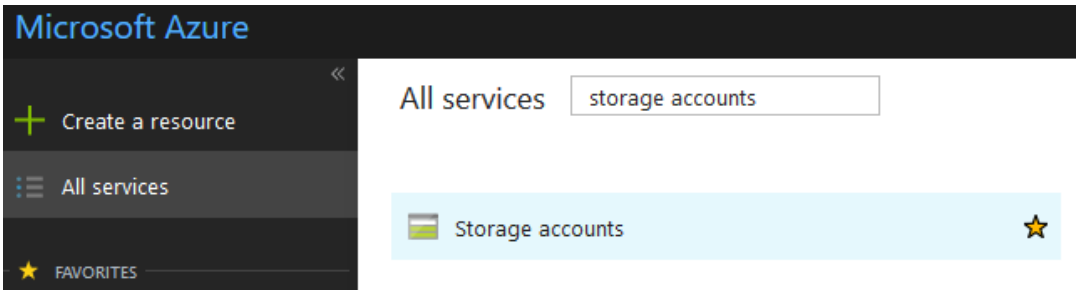


- Storage Accounts

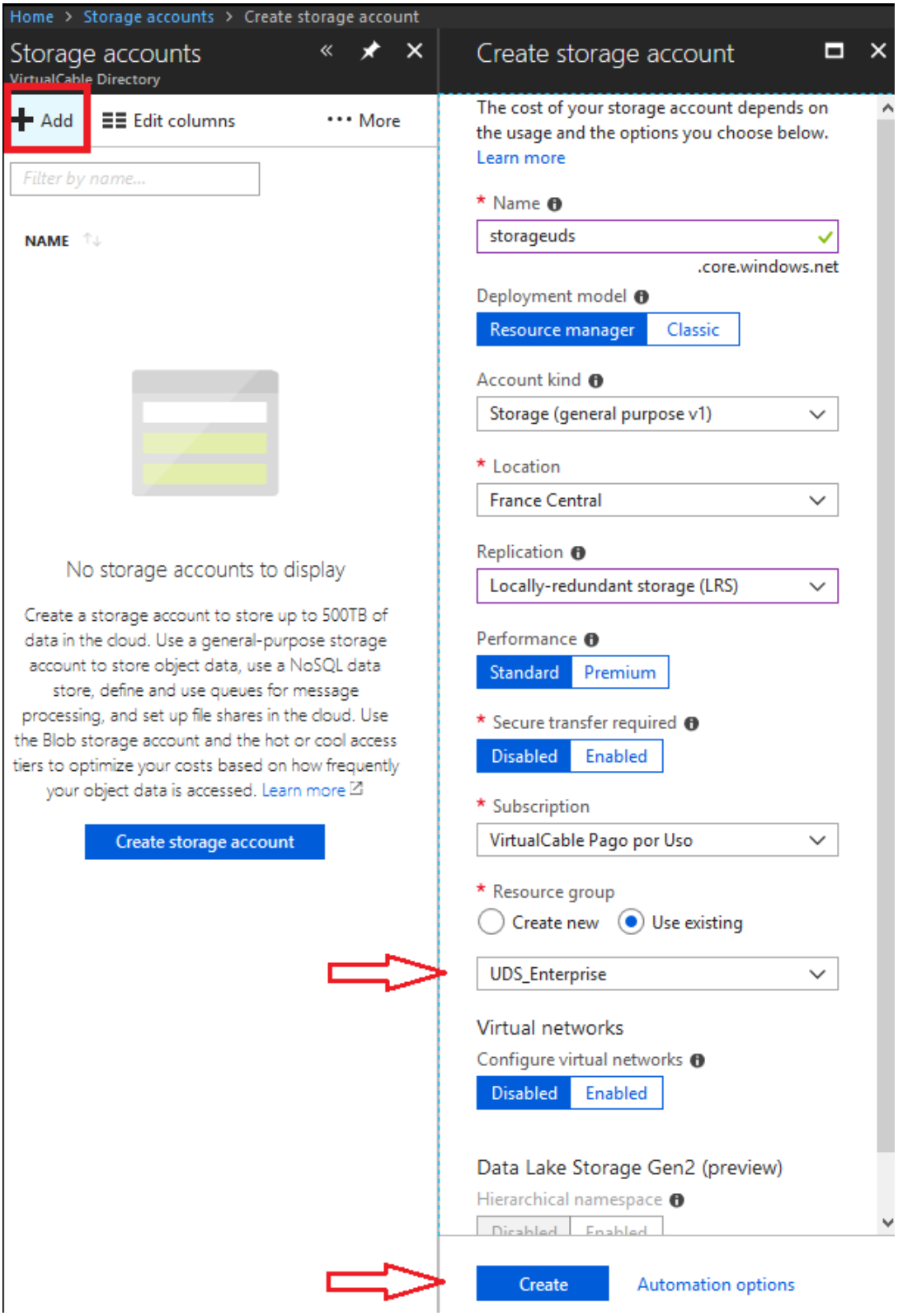
The next item we will need is a "Storage Accounts". This element will allow us to import the UDS components and generate the virtual disks to later implement the UDS virtual servers.

If you do not have one, you can create it by following these steps:

1. In the list of “Services”, we look for “Storage accounts” (it is recommended to add it to your favorites list) and click on it:

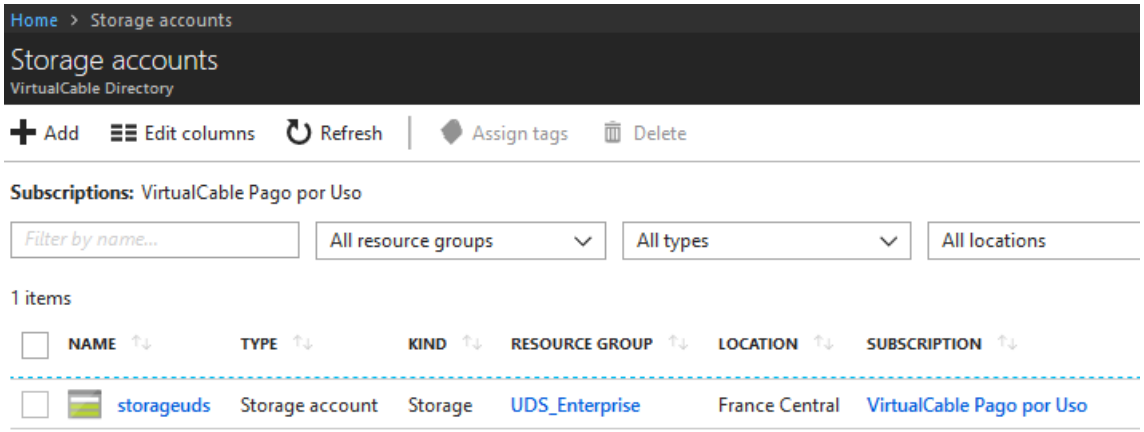


2. Once inside, click “add” to create a new one. We indicate a descriptive name, select the subscription in which it will be registered and select the "Resource Group" created above. Finally, click on “create”.



The choice of other options available when creating a “Storage Account” (“Account Type”, “Location”, “Replication” and “Performance”) does not affect the operation/implementation of UDS, but may affect the final cost.

3. We confirm that the “Storage account” has been correctly created.

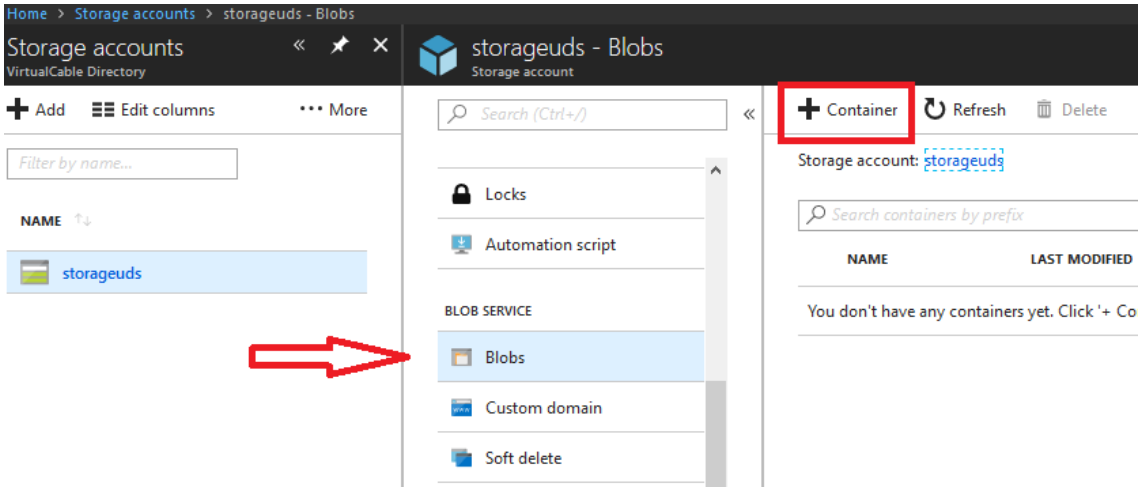


Container

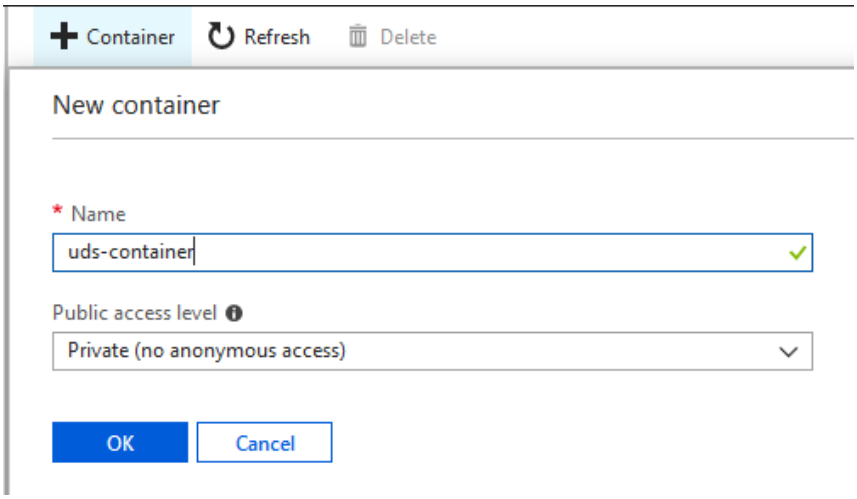
Once we have a valid “storage account”, we will need a “container” to load the disk images of the UDS servers.

If you do not have one, you can create it by following these steps:

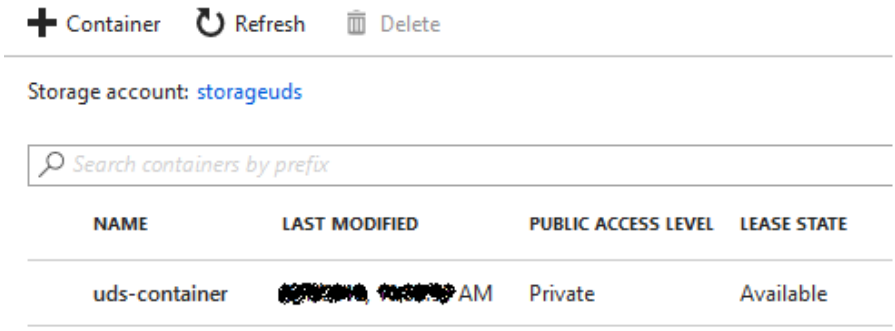
1. We select the “Storage account” where we’ll upload the UDS images. In the “BLOB SERVICE” menu, select “Blobs” and click on “Container”:



2. We indicate a descriptive name for the new “Container” and select the “Public access level” appropriate to our needs. Let's click on “ok” to finish the creation.



3. We confirm that the “Container” has been correctly created:

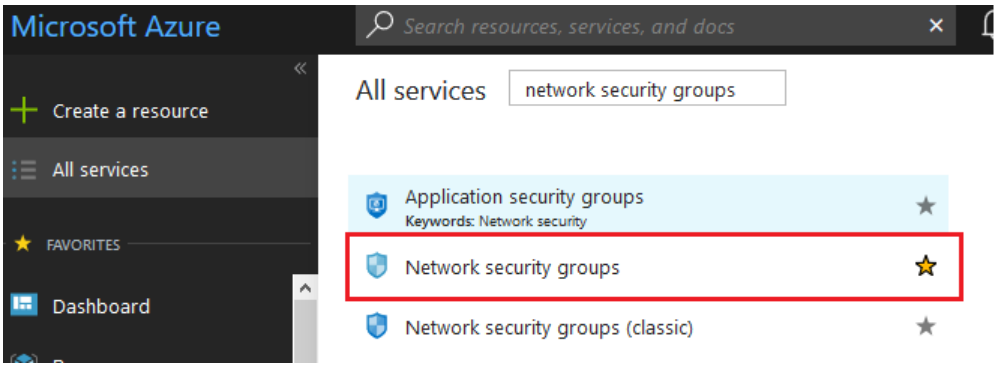


- Network security groups

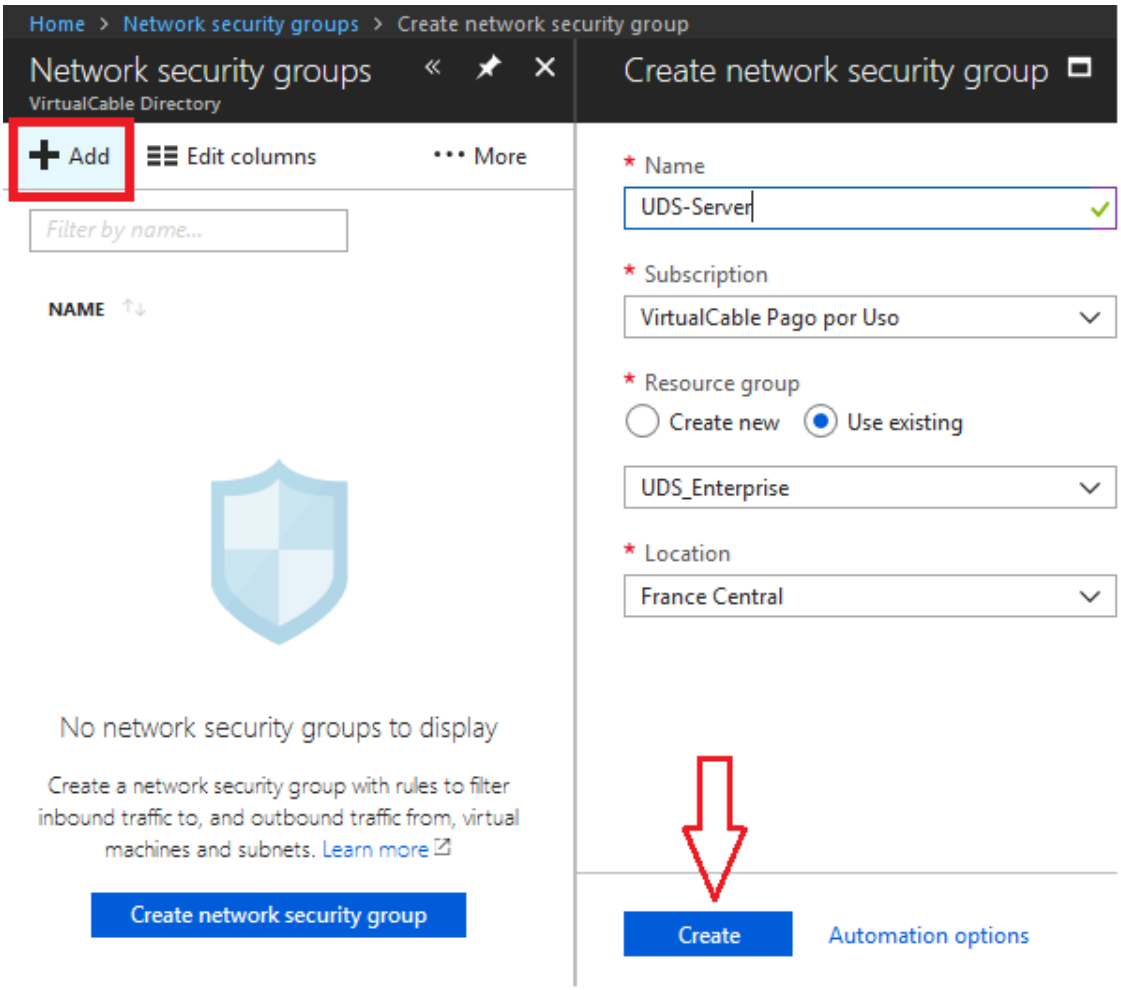
Another element necessary for the deployment of UDS will be the “Network Security Groups”, which will perform the firewall function.

For the different elements of UDS, specific ports will be required. Below are the ports that must be configured for the proper functioning of UDS:

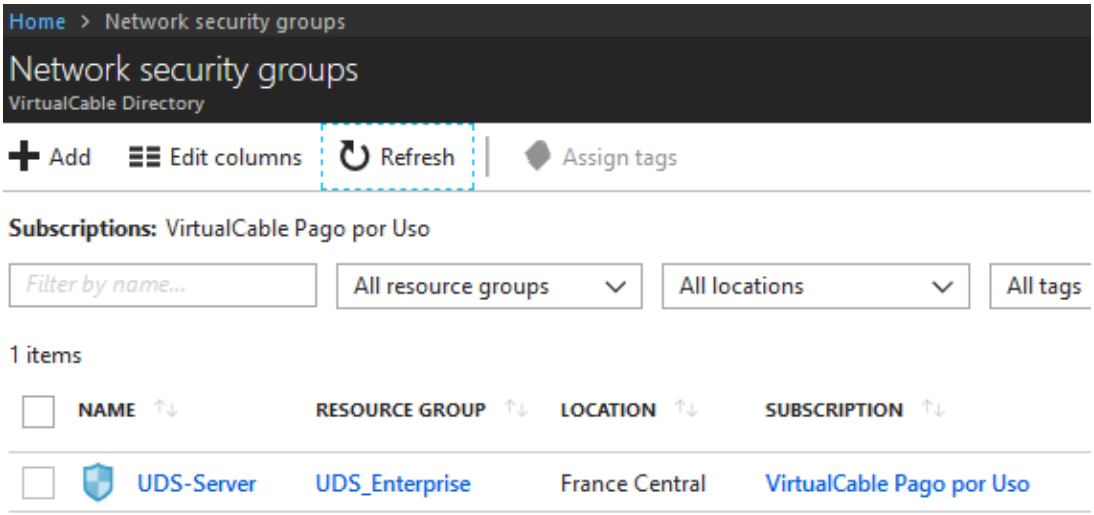
1. In the list of “Services”, we look for “Network security groups” (it is recommended to add it to your favorites list) and click on it:



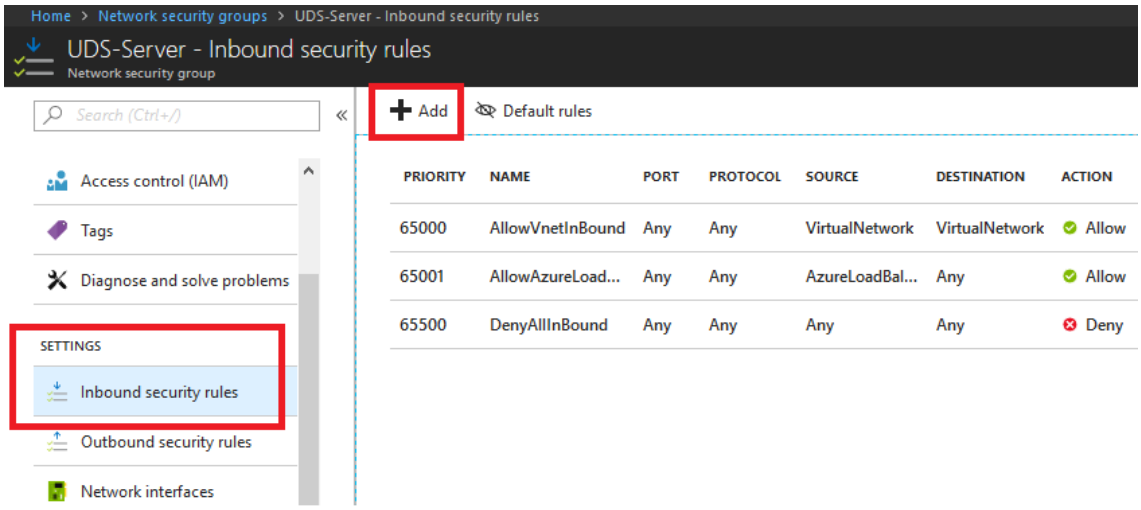
2. Once inside, click “add” to create a new one. We indicate a descriptive name, select the subscription where it will be registered, select the “Resource Group” previously created and the “Location”. Finally, click on “create”.



3. Confirm that the “Network Security Group” has been correctly created (it will be necessary to create two, one for the UDS server and another for the UDS Tunneler server):



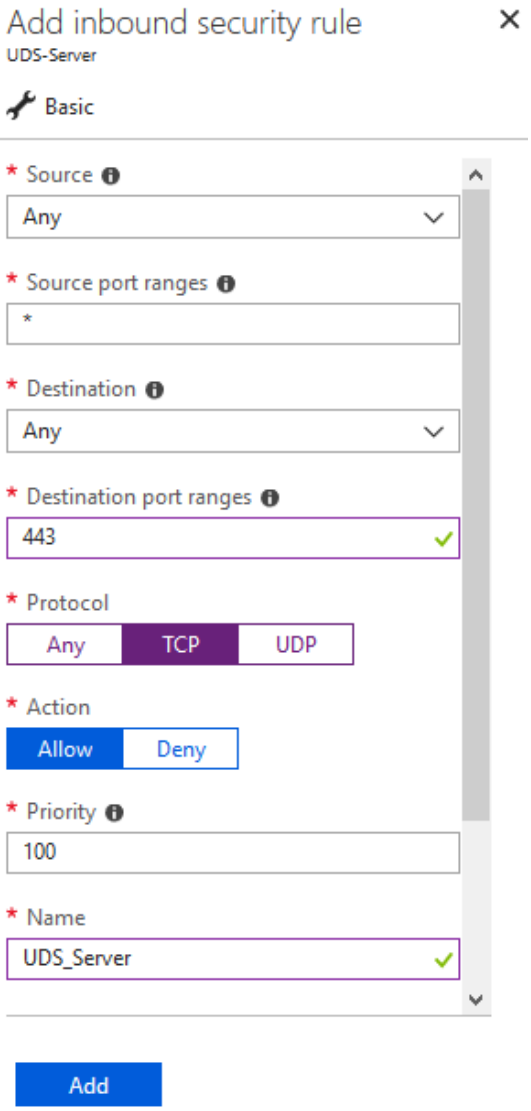
4. Access the new “Network Security Group” created by clicking on its name. In the “CONFIGURATION” menu, select “Entry security rules” and click “add” to create the necessary access rules:



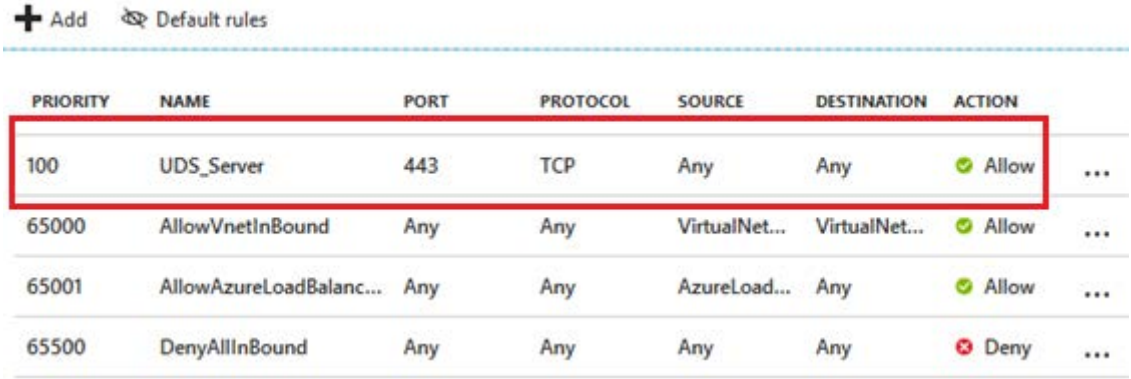
5. We will have to configure two “network security groups”; one for UDS Server and one for UDS Tunneler, each with its corresponding rule in “Rules of entry security”. In the following table you can verify the ports needed to access the UDS components and the service they will offer:

COMPONENT	PORT	FUNCTION
UDS SERVER	443	Access login panel
UDS TUNNELER	443, 10443	Access services and HTML5

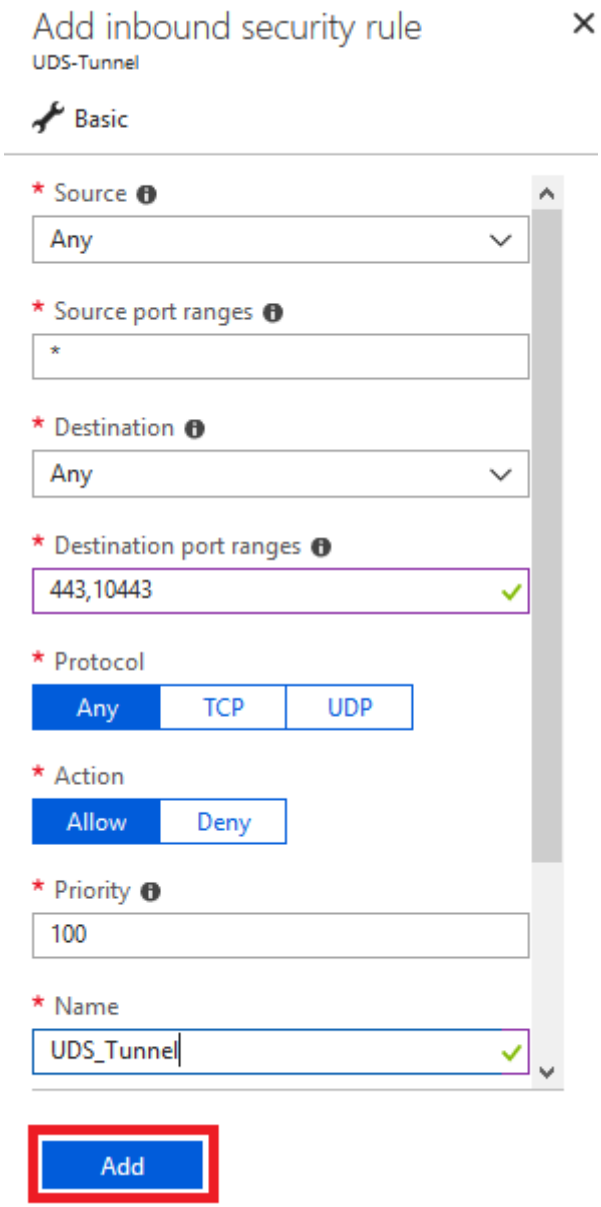
a) **UDS-Server:** We must create a UDS Server access rule where we allow traffic through port 443 TCP:



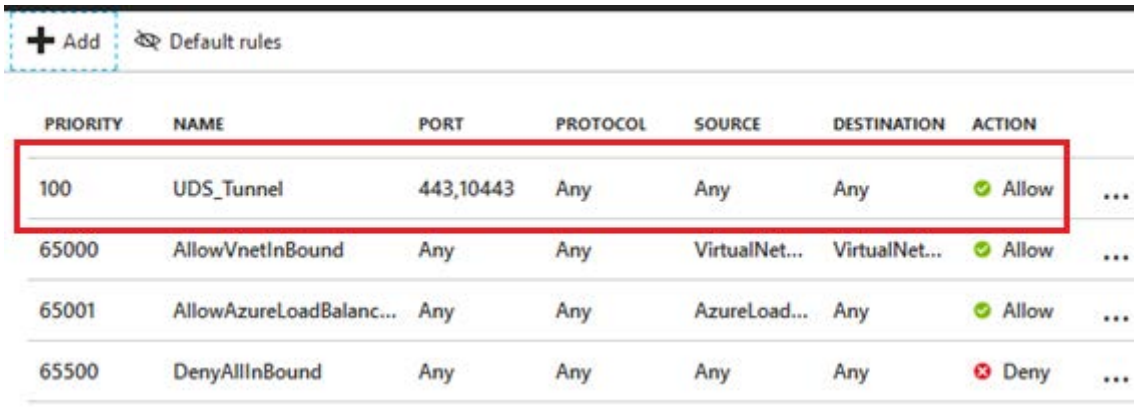
Once we have indicated the data as shown in the screenshot, we will click on “add” to create the rule and confirm its correct creation:



b) **UDS Tunneler:** We must create an access rule to access the UDS Tunneler server where we allow traffic through TCP port 443 and 10443:



Once we indicate the data shown in the screenshot, we will click on “add” to create the rule and confirm its correct creation:



Implementing UDS servers on the Azure platform

Below you can find an example of how to deploy the servers that make up the UDS Enterprise environment on an Azure platform. This guide details the steps to upload and create the UDS Server component. The same tasks must be performed for the UDS Tunneler server.

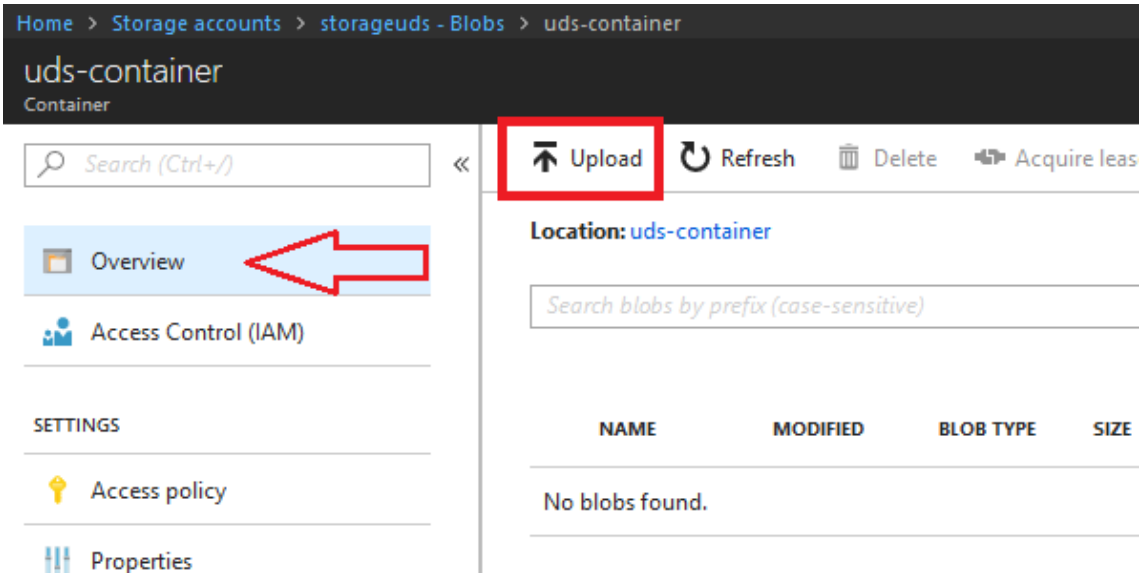
If the UDS version to be installed is Enterprise, you should also load the MySQL database server on the platform (remember that with UDS Free Edition and UDS Evaluation Edition it is not necessary to have a database server).

The VirtualCable team will provide the UDS servers in disk image format (.vhd)

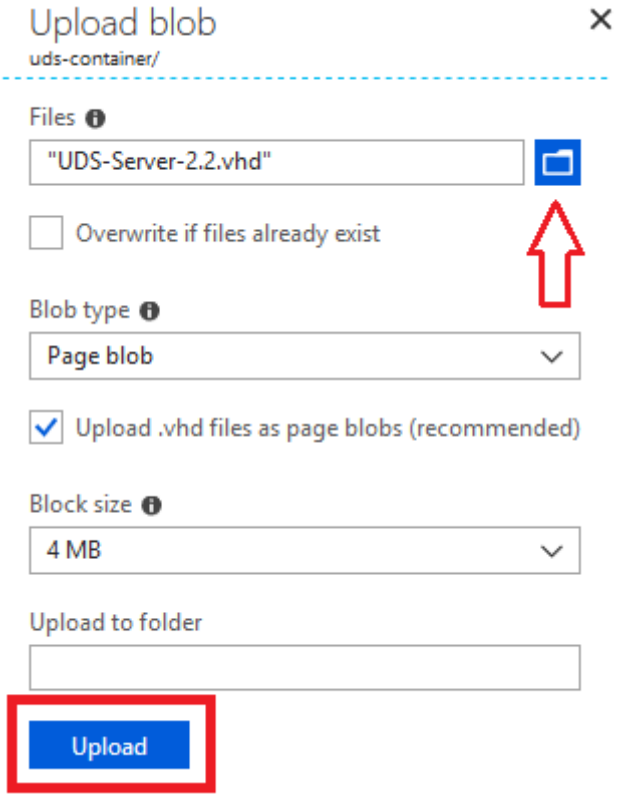
▪ Uploading disk images

The first task we will perform is importing the disk image of the UDS server. To do this, we must have a “Container” and the disk image of the UDS Server in .vhd format

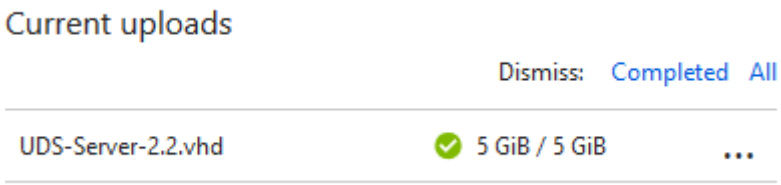
1. Access the “Container” (“Storage accounts”, in the “BLOB SERVICE” section, select “Blobs” and click on the “Container”) and click on “Upload”:



2. We indicate the image of the disk in the “Files” section. In “Blob type” we select “Page blob” and click on “Upload”



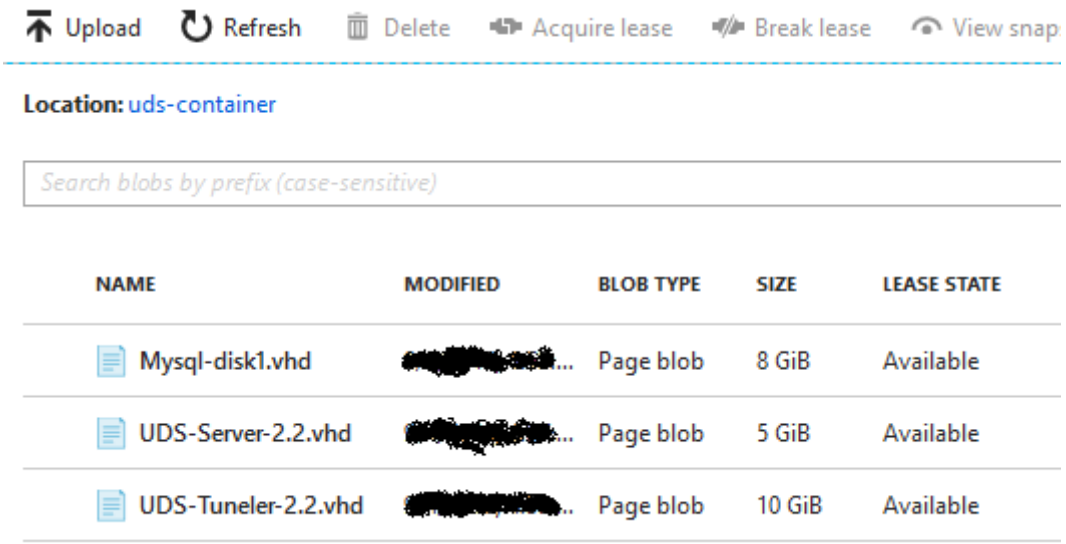
3. The image will begin to be imported and we will have to wait until the loading process finishes. Once finished, we will proceed to the next task, which will consist on the generation of a disk from the image.



Note: Depending on the size of the disk images and the speed of the connection, this process may take several minutes.

It will be necessary to repeat this process with the UDS Tunneler component and, in case of having an Enterprise version, with the MySQL database server.

Finally, we will see that we will have the UDS images available within the “Container”.

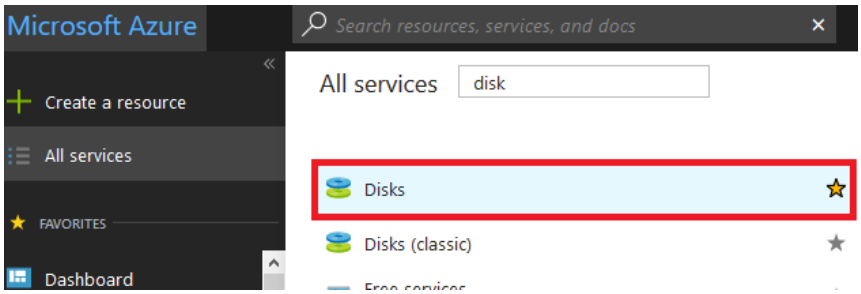


Creation of disks

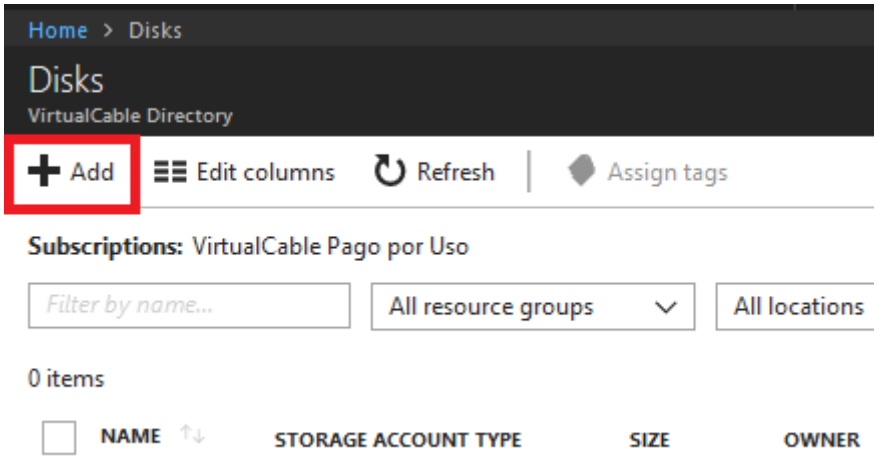
When we have the images of the different UDS components uploaded to the Azure platform, we will proceed to implement virtual disks based on these images.

From the virtual disks that we will create next, we will generate the virtual machines that will form the UDS environment.

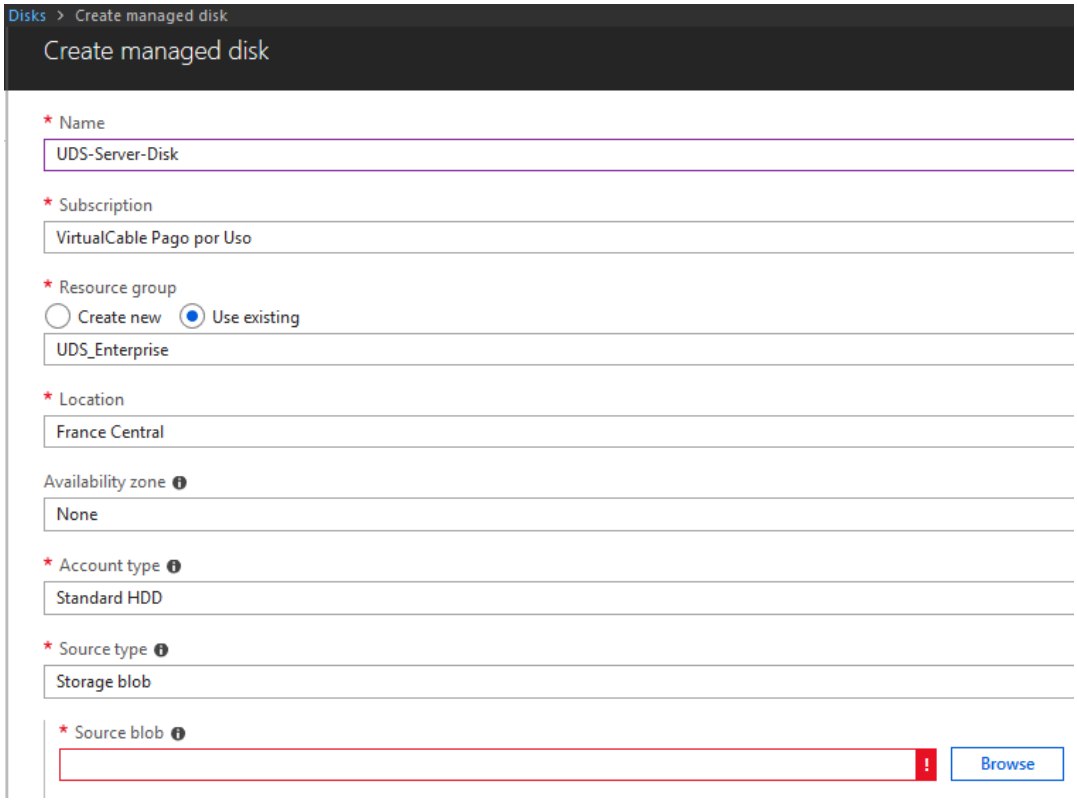
1. In the list of “Services”, we search for “disk” (it is recommended to add it to your favorites list) and click on it:



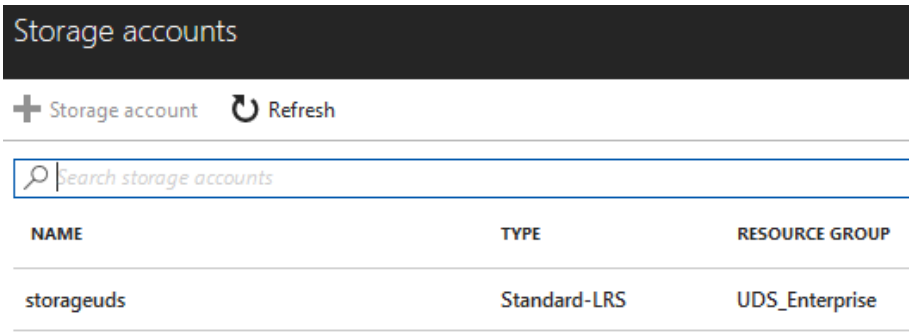
2. Click on “Add” to add a new disk.



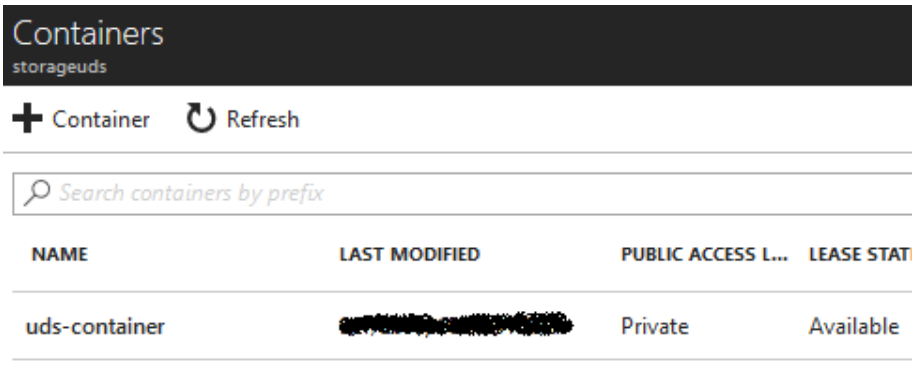
3. We indicate a descriptive name and a subscription, select the “Resource Group” for UDS, “Location” and the “Type of account” according to the required performance:



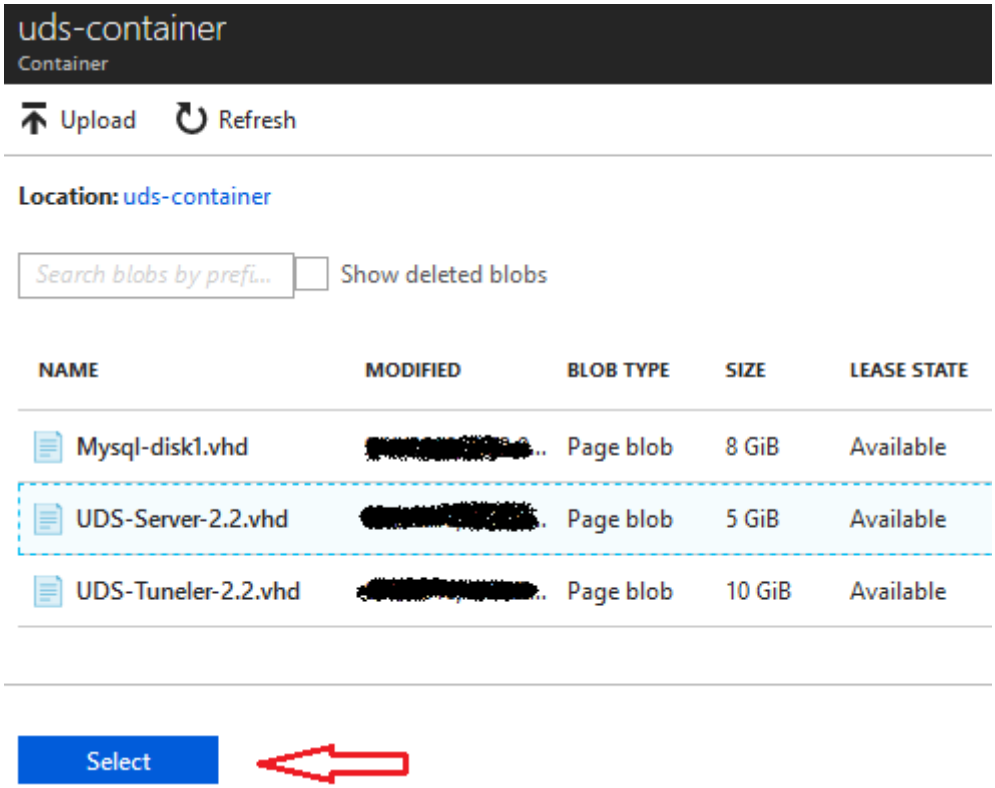
In “Source type” we should choose “Storage Blob”. To indicate the image, click on “Browse” and select the “Storage account” created above:



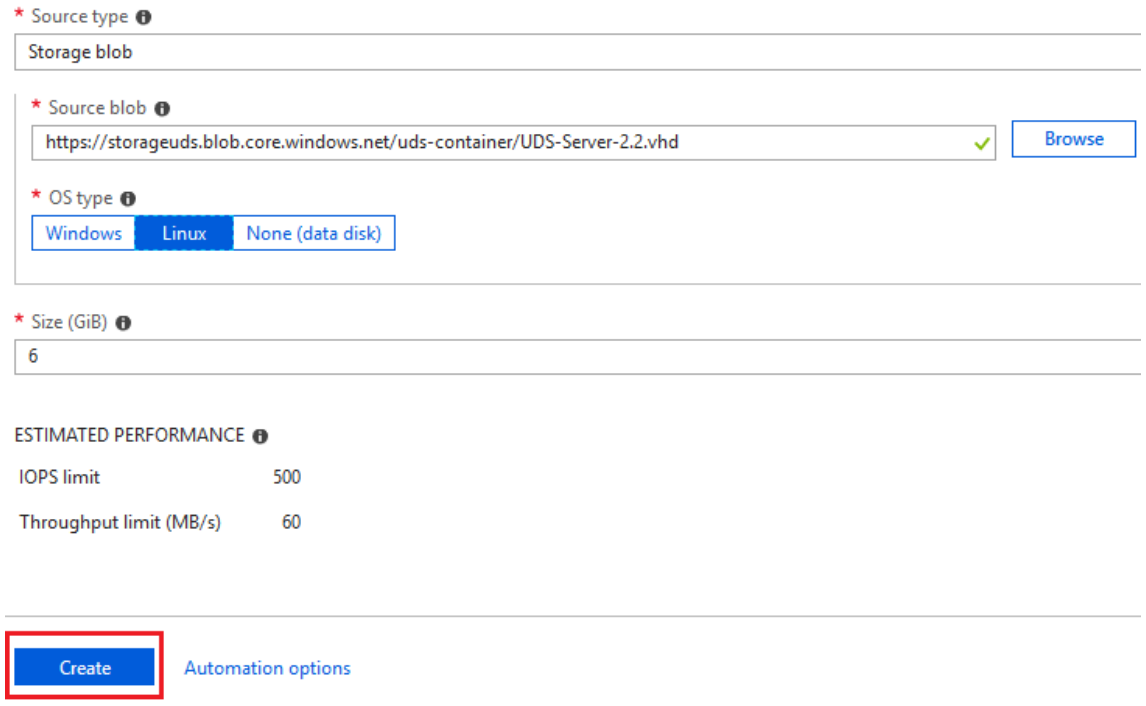
Once inside, we select the “Container” created above:



And finally we select the image imported in the previous step (in this case for the UDS server) and click on “Select”.



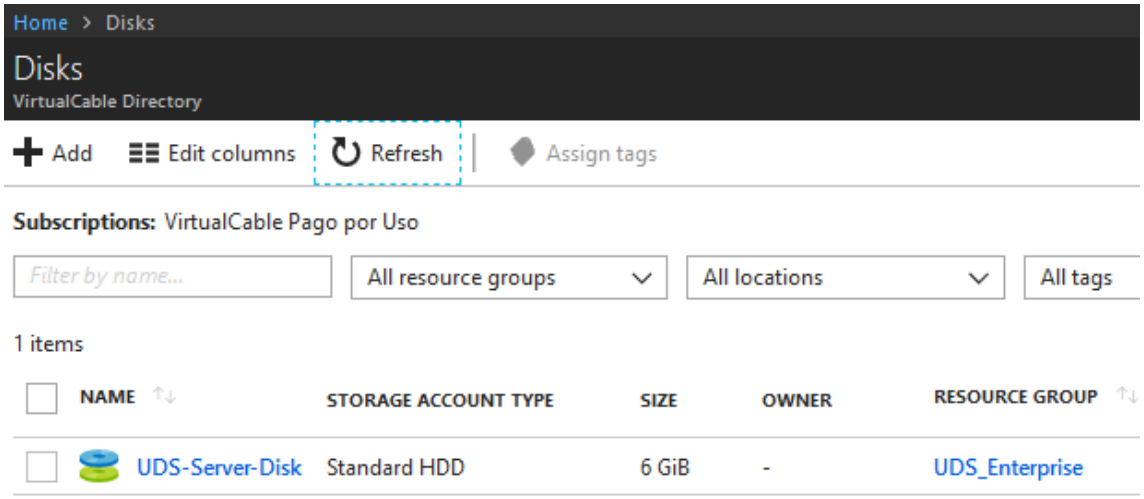
In “OS Type” we will indicate that it is “Linux” and we will enter the size of the disk in the “Size” field (we will always assign 1 GB more than the size of the image) and we will finish by clicking on “Create”.



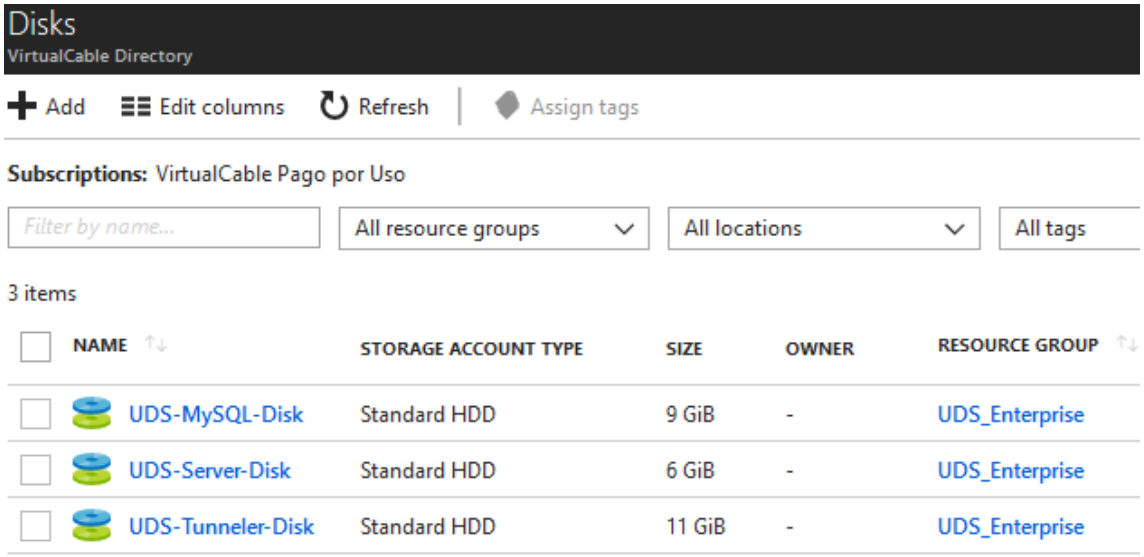
NOTE: The sizes of the disks for the different components of UDS Enterprise 2.2 will be the following:

COMPONENT	SIZE IN GB
UDS SERVER	5
UDS TUNNELER	11
MYSQL	9

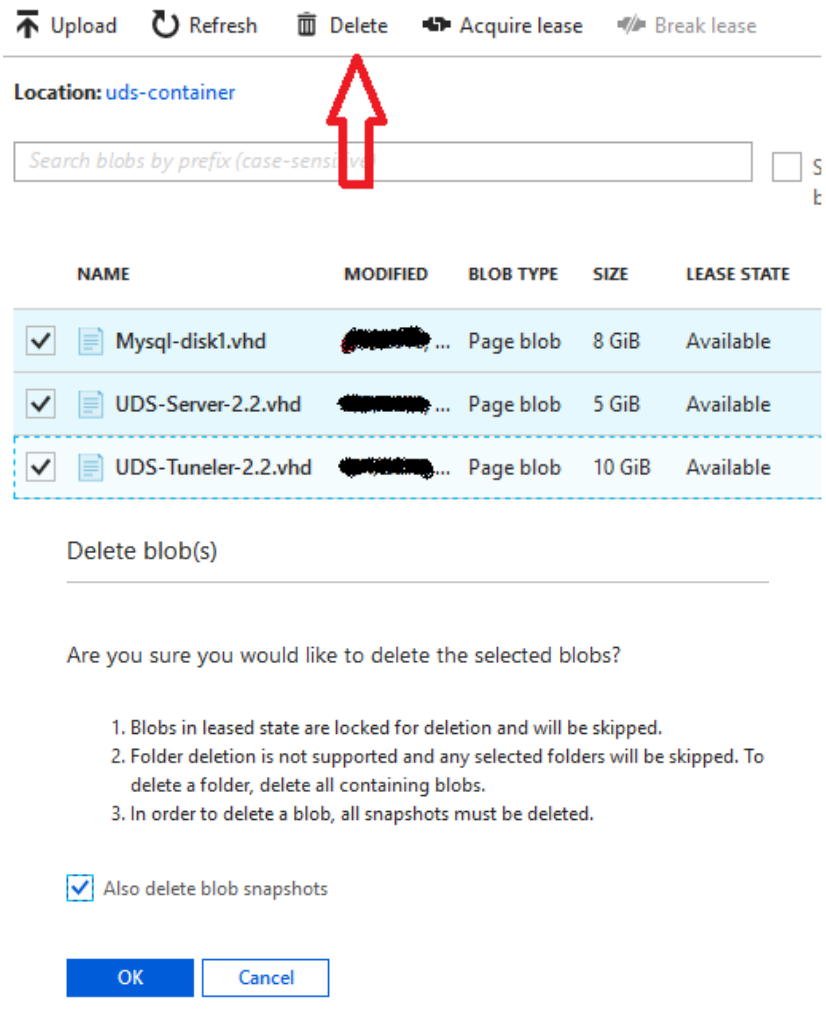
4. We will wait for the disk to be created and, once the task is finished, we will see what we have available to generate the virtual machines later:



5. Repeat the process with the UDS Tunneler component and, if you have an Enterprise version, also with the MySQL database.



NOTE: Once the disks are shown, we can remove the images from the "Container" to avoid unnecessary costs:

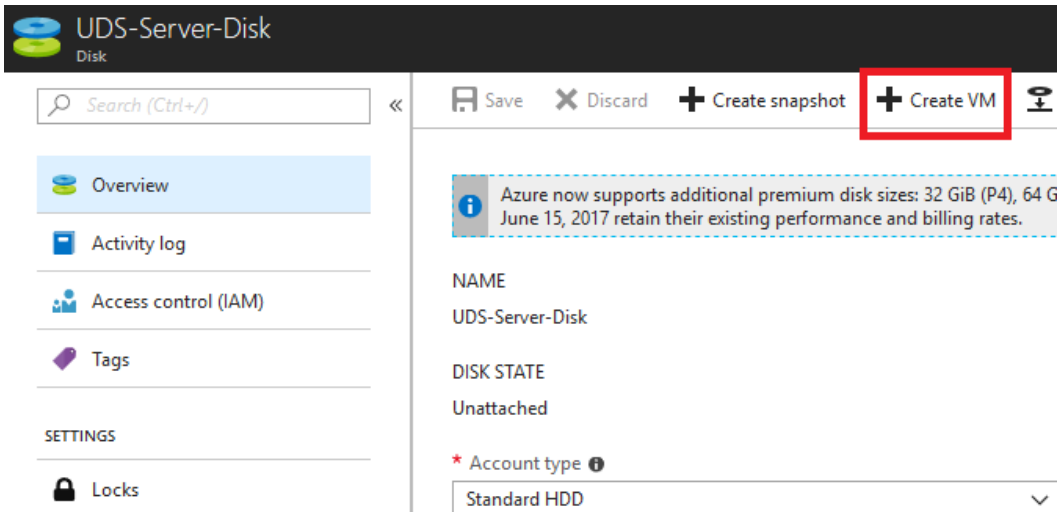


Creation of UDS virtual servers

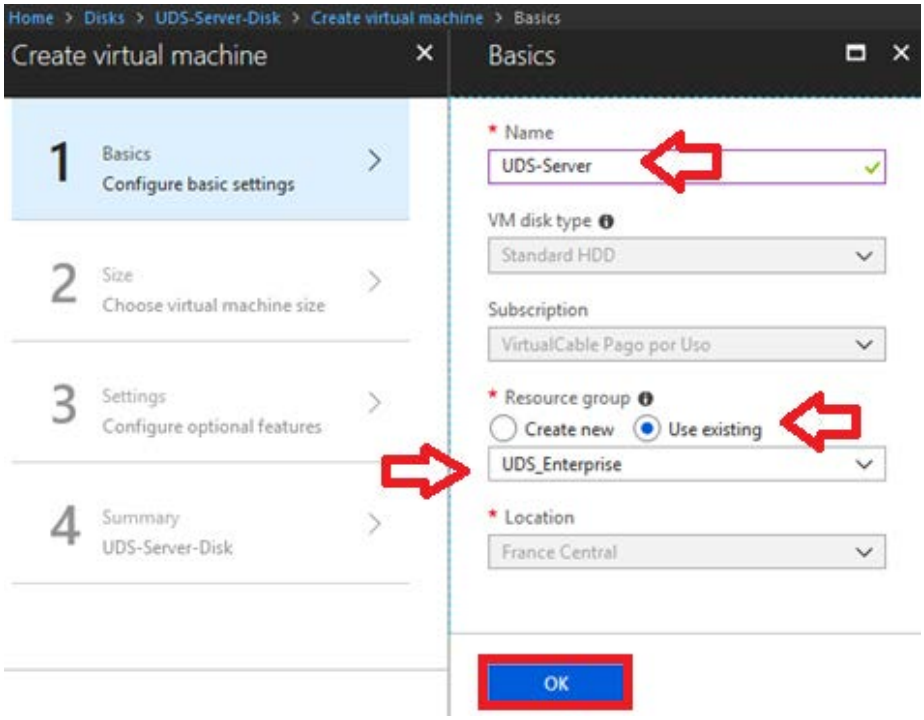
The last task we will perform in the process of importing / creating the UDS components will be the creation of the virtual machines based on the disks created in the previous step.

We will create the machines from the disks themselves:

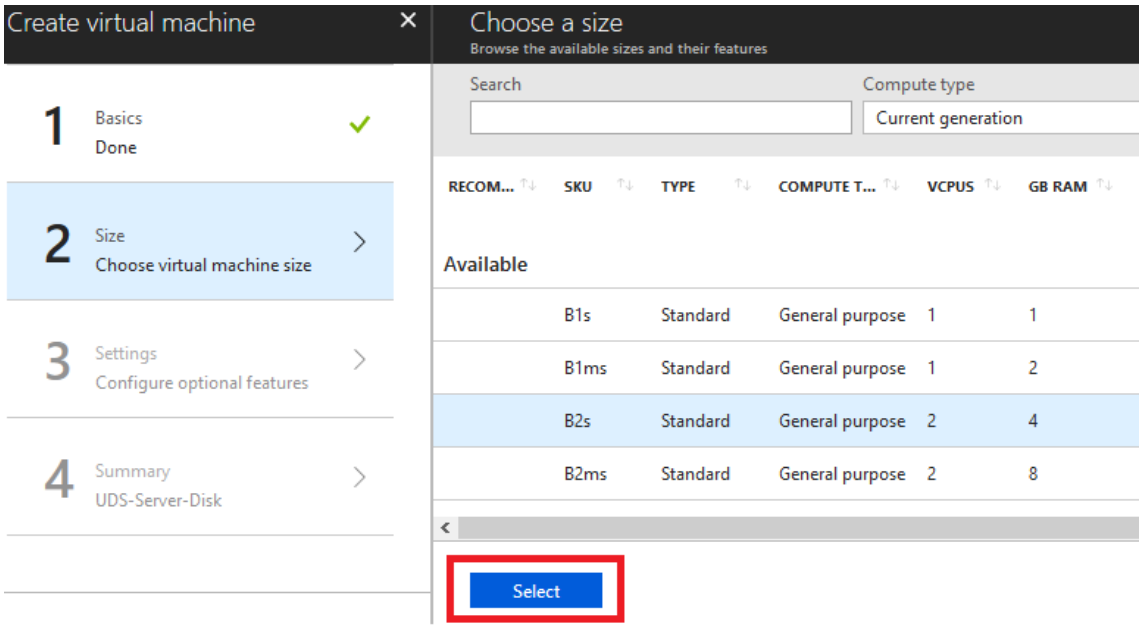
1. Select the previously created disk (from the “Disk” service) and click on “Create virtual machine”:



2. In the first step of the creation assistant, we will indicate the name of the virtual machine and select the “Resource Group” to which we have assigned all the UDS services. Click on “OK”.



3. In the second step we select the type of machine and the resources it will have. We mark the type and click on “Select” (in this example we will select for the UDS Server and UDS Tunneler the type B2s and for the MySQL server the type B1s).



The minimum resources of the UDS components are shown in the following table:

COMPONENT	VCPU	VRAM
UDS SERVER	2	1
UDS TUNNELER	2	2
MYSQL	1	1

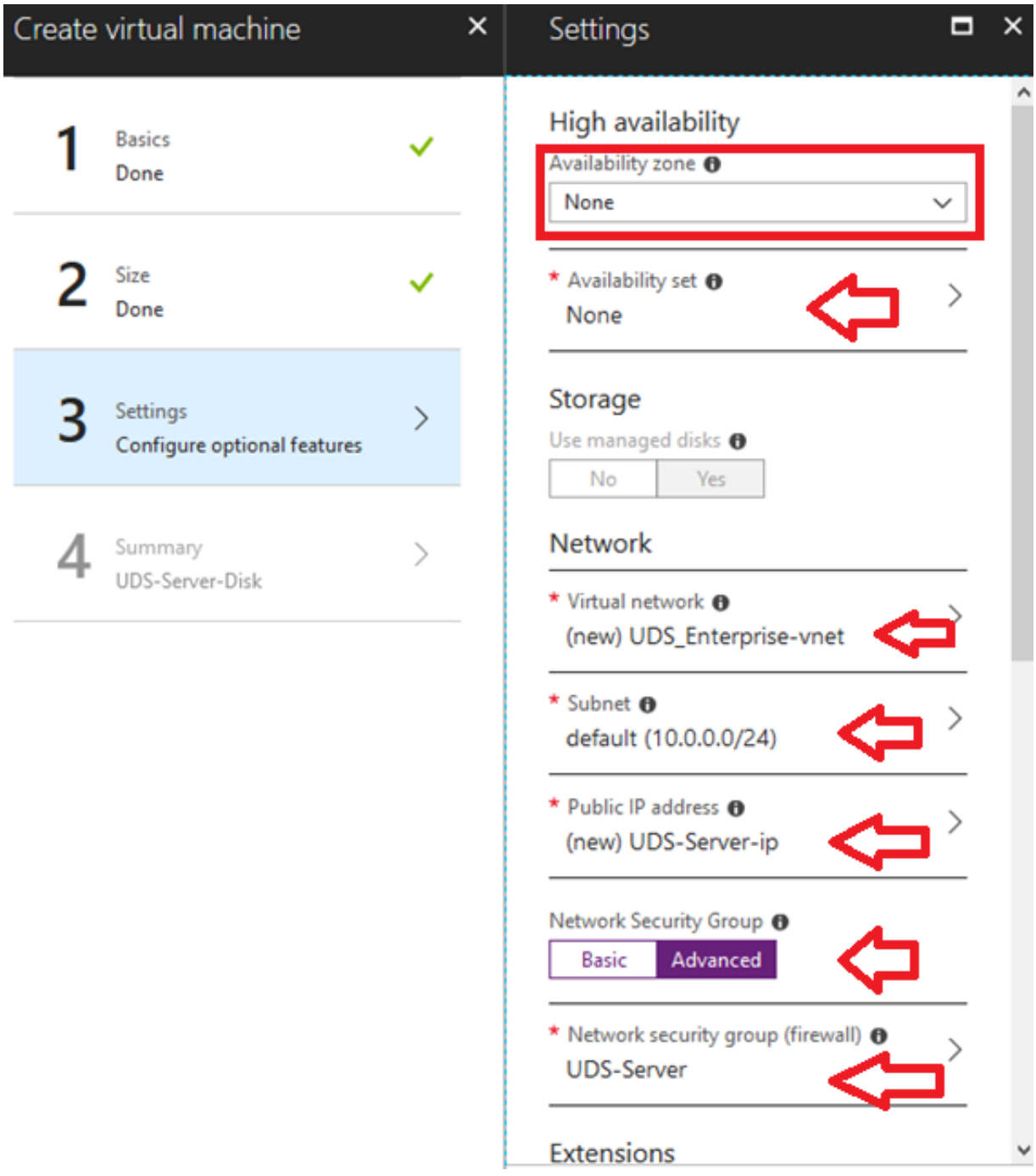
NOTE: UDS Tunneler server resources will be the only ones that tend to increase depending on the number of concurrent users (with the resources indicated in the table, a Tunneler server can reach between 30 and 50 users / concurrent connections with a medium profile use).

NOTE: We can assign the type B1s to the MySQL database.

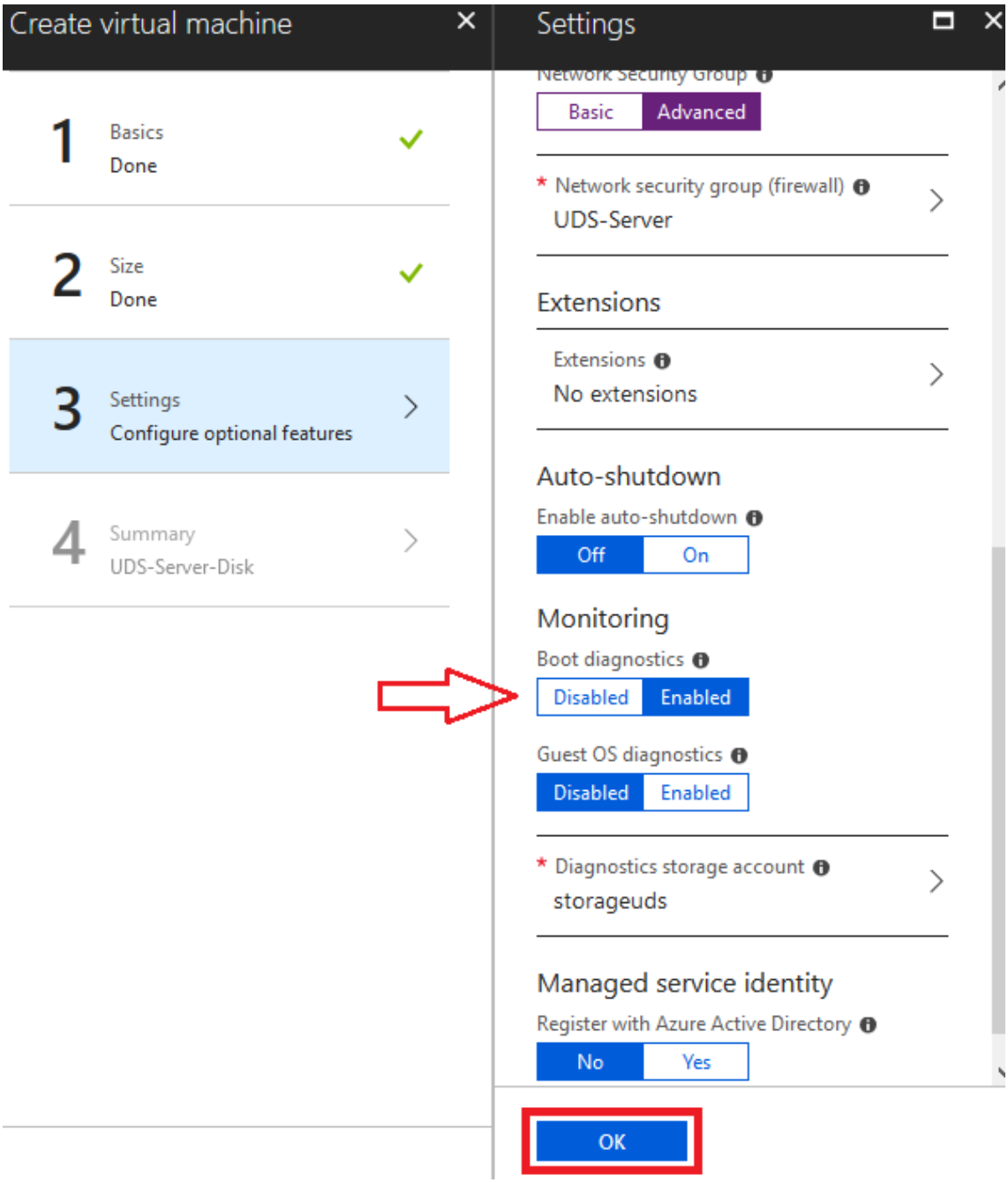
4. In the third step we will indicate the following necessary data in the Network section:
   a. **Virtual network:** If we already have a “virtual network” to connect the machines, we select it. Otherwise, we allow the system to create a new one (we can leave the name by default or indicate another). The rest of the components (UDS Tunneler and the MySQL database, must be assigned to the same network)
   b. **Subnet:** If we already have a “subnet” that assigns IP addresses to the virtual machines, we select it. Otherwise, we let the system create a new one (we can leave the “default” or indicate a specific one)

c. **Public IP address:** We will let the system assign a public IP address to the server. This default IP address will be dynamic, although once the VM is created, we can generate a DNS name associated to this IP or even force the machine to have a static public IP (the UDS Tunneler component must also have a public IP assigned, but for the MySQL database it won't be necessary, so in this case we will select "None")

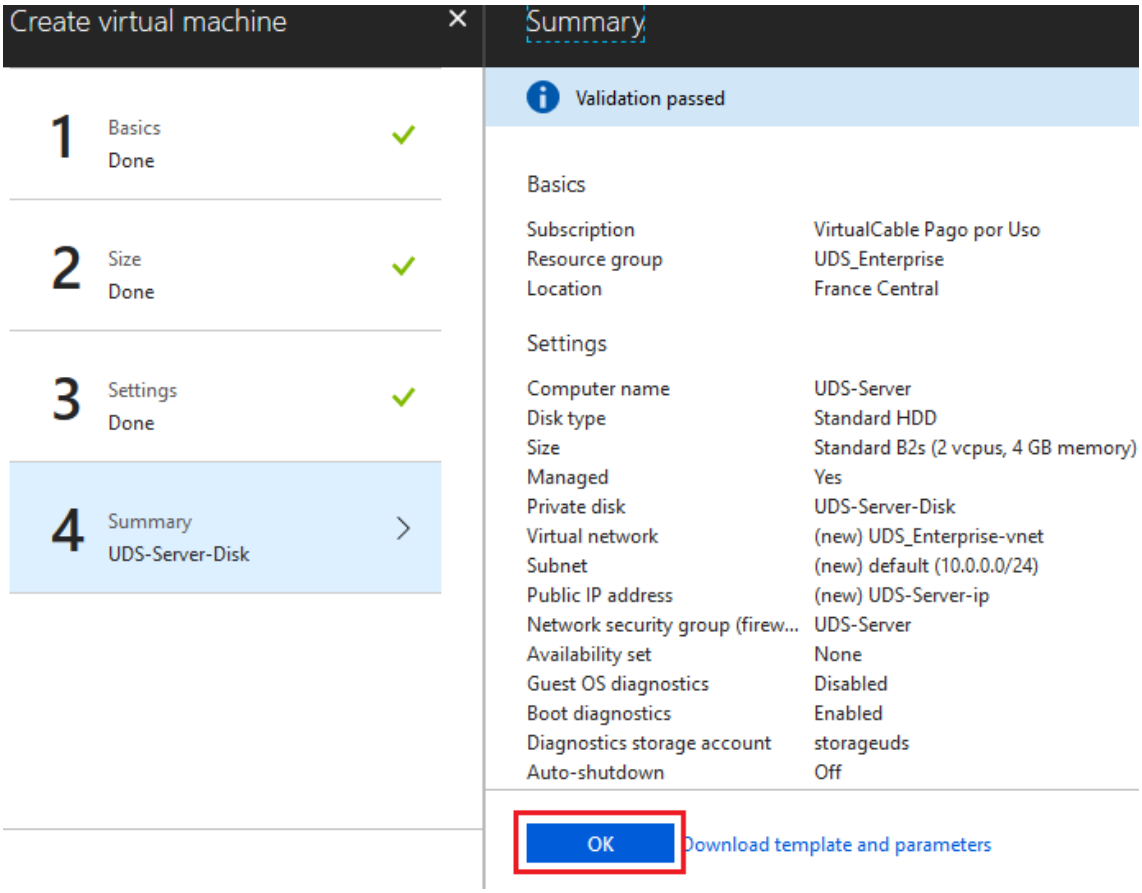
d. **Network Security Group:** We must select the "Advanced" option and assign the "Network Security Group" defined above for the UDS Server or UDS Tunneler component (in the case of the MySQL database server, we can select the "None" option, since access from outside the system is not necessary)



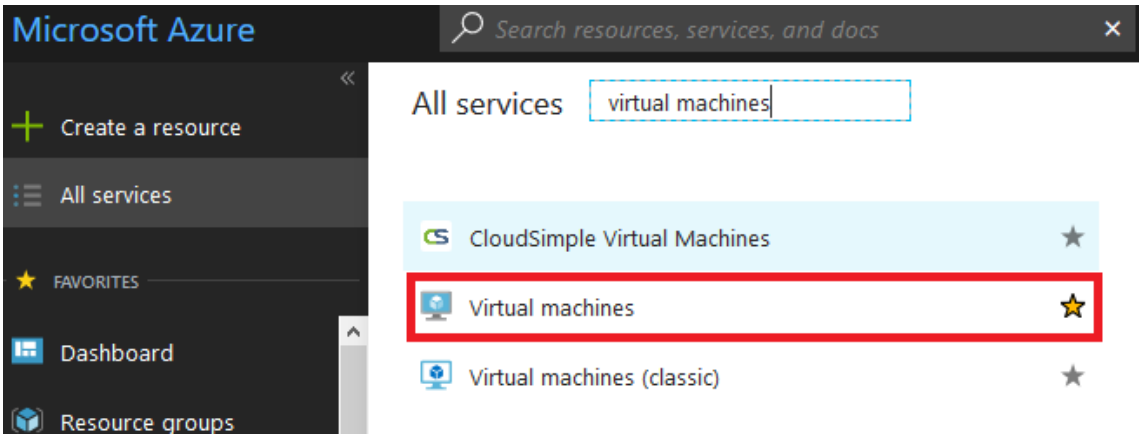
In step 3 it will also be necessary to activate the “Boot diagnostic” option in the “Monitoring” section, which will allow us to see a screenshot of the start and the subsequent status of the virtual machine. In addition, this option is necessary to access the “Serial console” (to enable “Boot diagnostics” it will be necessary to indicate a “Storage account”, in this case we select the one previously created to avoid the creation of a new one). Click “OK” to complete step 3:



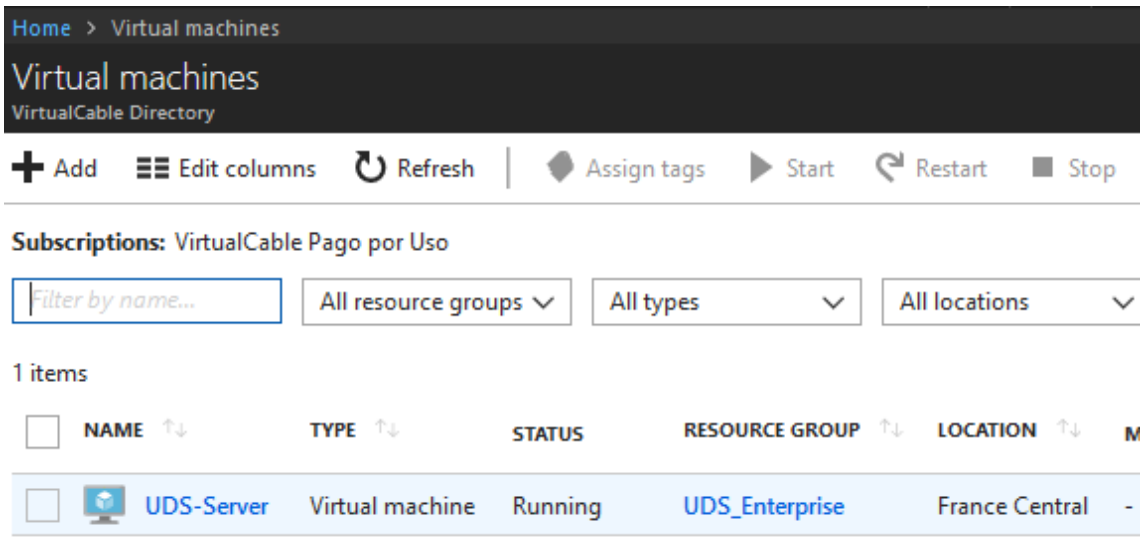
5. Finally, in step 4, we will verify that all the configuration is correct and click “OK” to create the virtual machine.



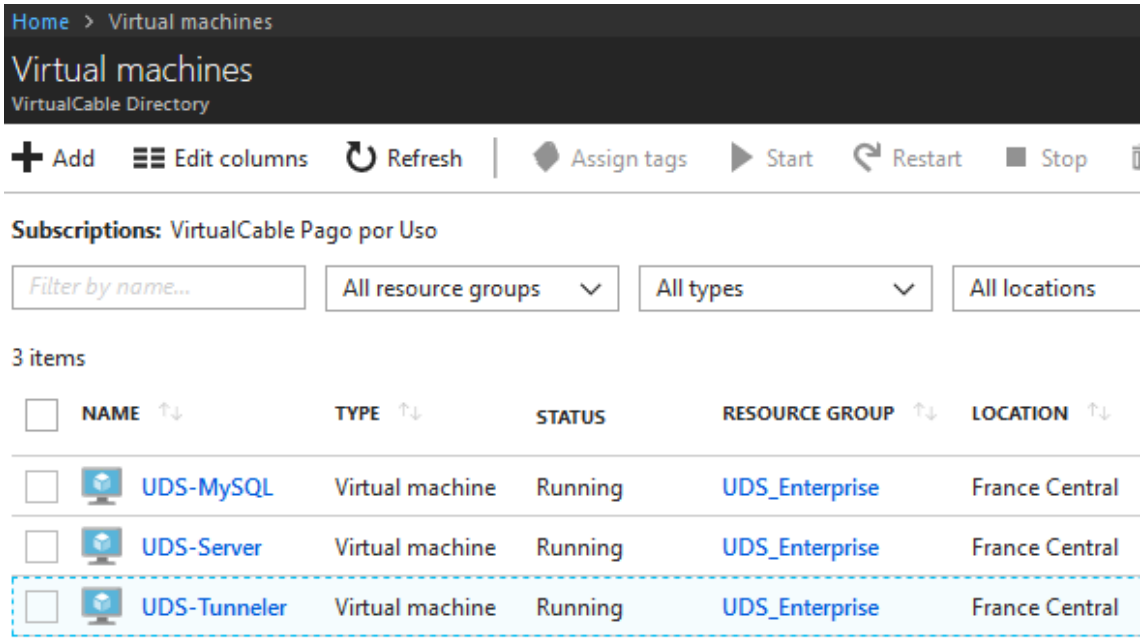
6. Once the creation process of the new virtual machine is finished, we will verify that we already have the new machine in the “Virtual Machines” service. In order to check it we will have to look in the list of “Services”, “Virtual machines” (it is recommended to add it to your favorites list) and click on it:



We will see the new virtual machine created and turned on:



7. Repeat the process with the UDS Tunneler component and if you have an Enterprise version also with the MySQL database.



UDS server configuration

Once we have all the UDS components deployed as virtual machines, we will proceed with their configuration.

To do this, we will access the “Virtual Machines” service and, in case of having an Enterprise version of UDS, we will start with the MySQL component (if we are going to implement a Free or Evaluation Edition, we will not need to have a database, since the UDS Server component automatically enables a local database).

o MySQL configuration

If you are using the MySQL database provided by the VirtualCable team, it will be pre-configured and you only need to verify that it has IP connectivity (by default the network is configured by DHCP).

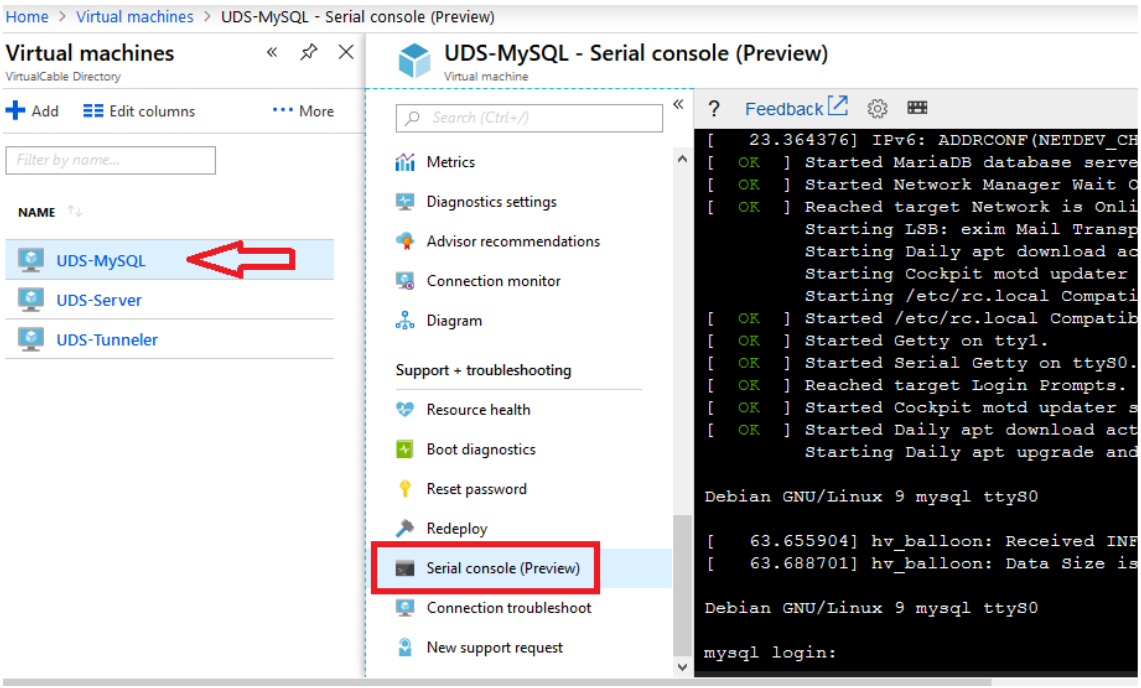
This MySQL server has created a BBDD instance ready to use with UDS Enterprise with the following data:

Instance name: uds

User: uds

Password: uds

In order to confirm that the server has a valid IP assigned via DHCP, we will have to connect via “Serial console”. We will access the “Virtual machines” service, select the virtual machine that contains the MySQL database and in the “Support + troubleshooting” menu we will select “Serial console (Preview)”.



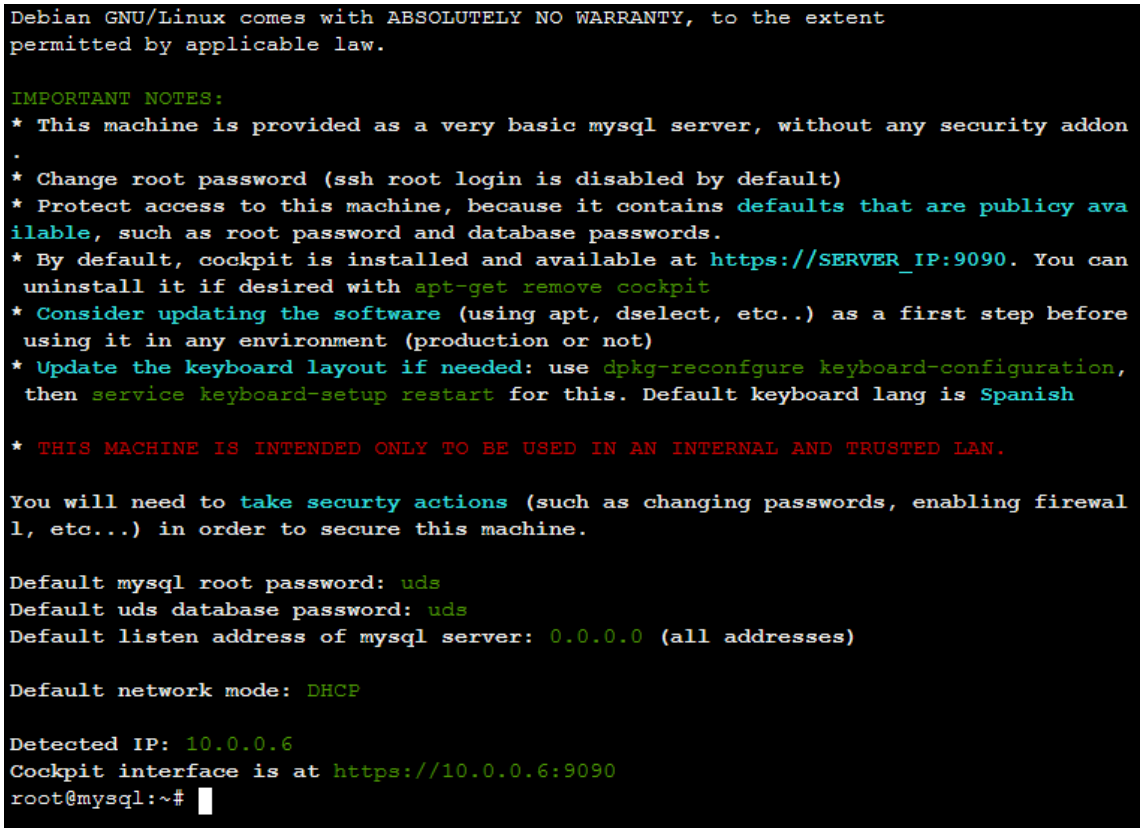
NOTE: The connection will take a few seconds and once connected, we must place the mouse inside and press the “enter” key.

We will validate in the MySQL server with the following data:

User: root

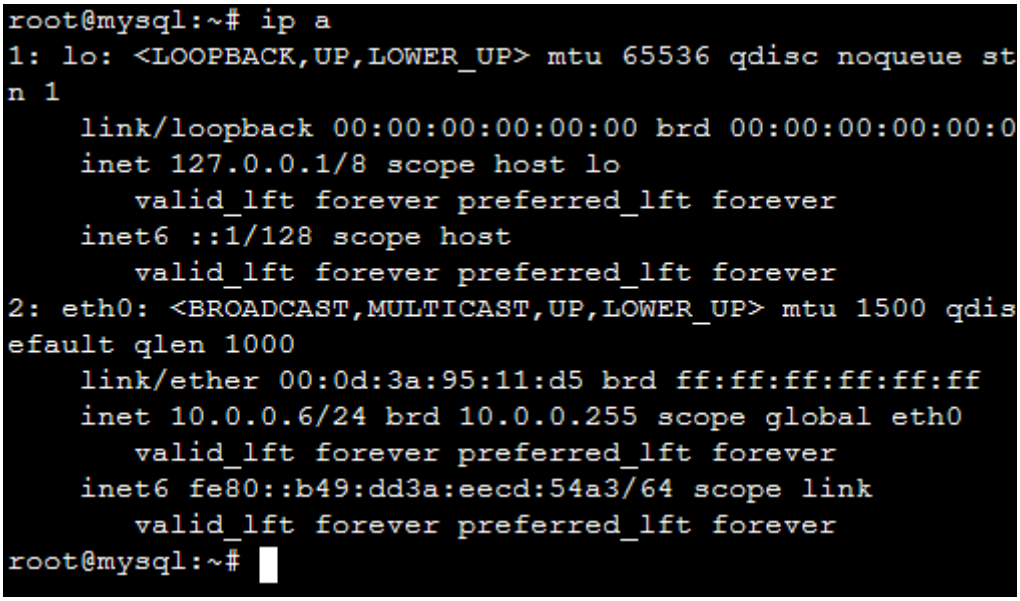
Password: uds

It will directly indicate the assigned IP address and relevant information about security and the configuration of the server itself.



If we want to confirm that the network configuration is correct, we can use the command:

ip a



Once we confirm that we have network connectivity, we will proceed to configure the UDS server component.

NOTE: The use of static IP addresses is recommended for all UDS components

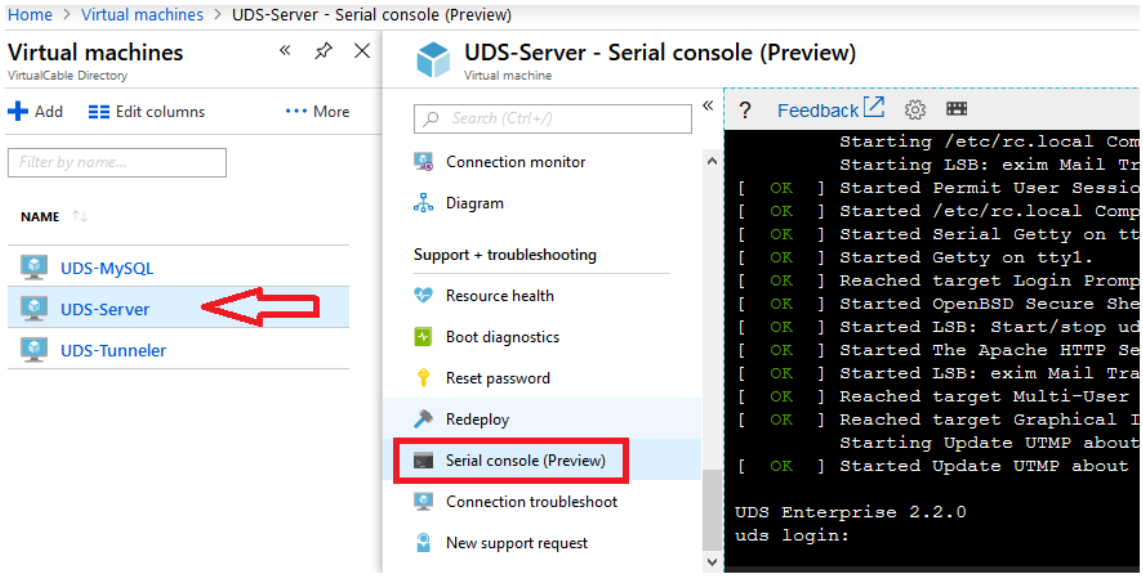
o UDS Server Configuration

The UDS Server component is the main element of the UDS environment. It has a configuration wizard that we will execute automatically. For this it will be necessary to indicate a series of data in the configuration file of the server.

NOTE: If you are configuring an Enterprise version of UDS, you must have a previously configured MySQL database server (with network connectivity through port 3306 and with a preconfigured database instance).

NOTE: To configure any version of UDS (Enterprise, Free or Evaluation) it is necessary to have a valid serial number provided by the VirtualCable team).

To access and execute the UDS server configuration wizard, we will have to connect through the “serial console”. We will access the service “Virtual machines”, select the virtual machine that contains the UDS server and in the “Support + troubleshooting” menu we will select “Serial console (preview)”.



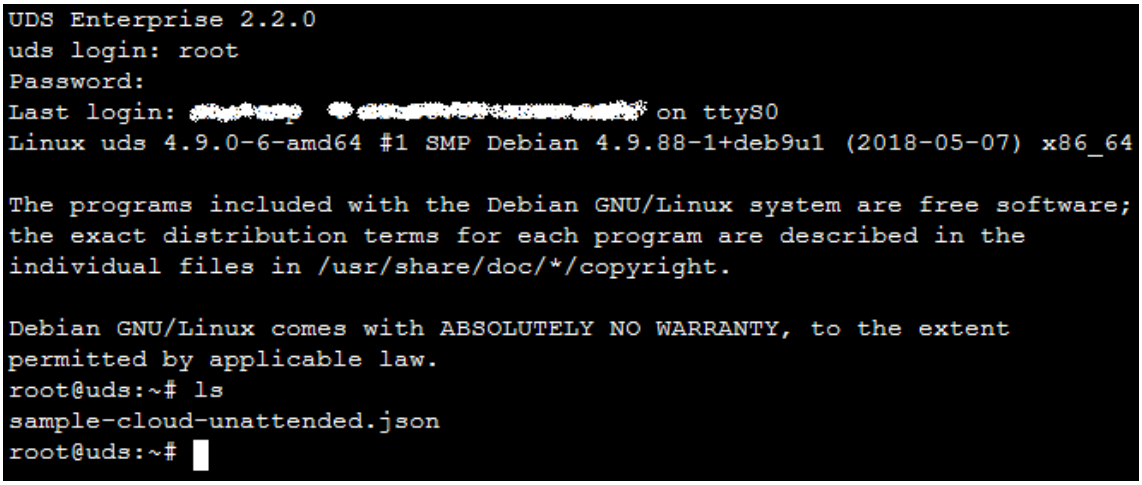
NOTE: The connection will take a few seconds and once connected, we must place the mouse inside and press the “enter” key.

We will validate on the UDS Server with the following credentials:

User: root

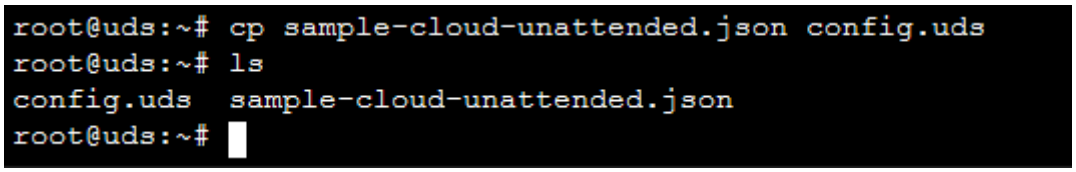
Password: uds

Once validated, we will confirm that in the /root directory we have an example configuration file with the name: “simple-cloud-unattended.json”



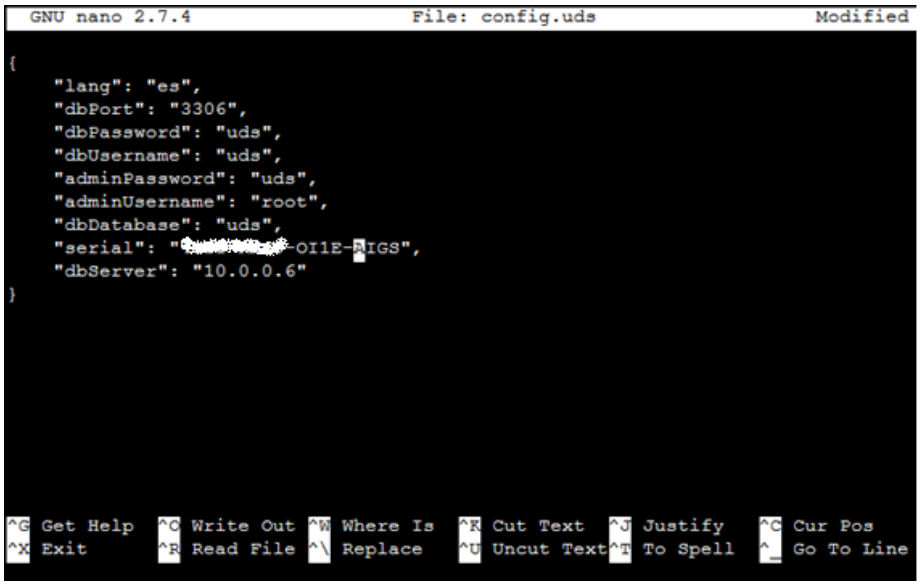
We will create a copy of this file and name it as: config.uds

cp sample-cloud-unattended.json config.uds



Once the new file is created, we will edit it and indicate the new configuration data:

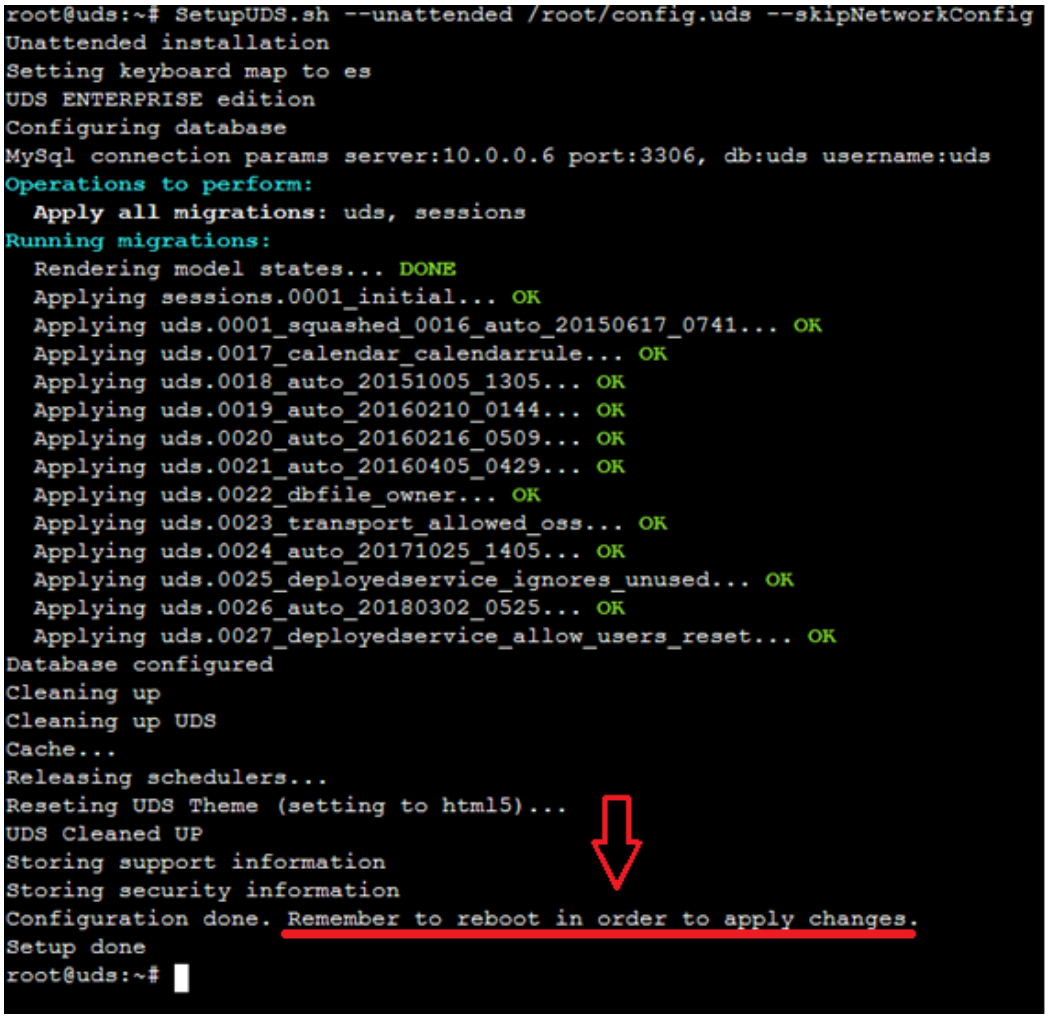
- **lang:** language of keyboard layout
- **dbPort:** connection port with the MySQL database (default 3306)
- **dbPassword:** password of the user configured in the MySQL database (default uds)
- **dbUsername:** user configured in MySQL database (default uds)
- **adminPassword:** password of the super-administrator of the UDS web environment and the root of the UDS Server virtual machine (default uds)
- **adminUsername:** name of the super-administrator of the UDS web environment (default: root)
- **dbDatabase:** name of the instance configured in MySQL (default: uds)
- **serial:** Serial number of the UDS Subscription (the serial can be for the UDS Enterprise version, UDS Enterprise Free Edition and UDS Enterprise Evaluation Edition)
- **dbServer:** IP address or name of the MySQL database



NOTE: If we are going to configure a UDS Enterprise Free or Evaluation Edition version, the data related to the MySQL database will not need to be configured (the system will ignore them).

We save the changes in the config.uds file and proceed to its execution. To do this we should launch the following command:

SetupUDS.sh --unattended /root/config.uds --skipNetworkConfig



Once the configuration process is correctly finished, **we must restart the server to apply the new configuration** and we can now configure the last UDS component, UDS Tunneler server.

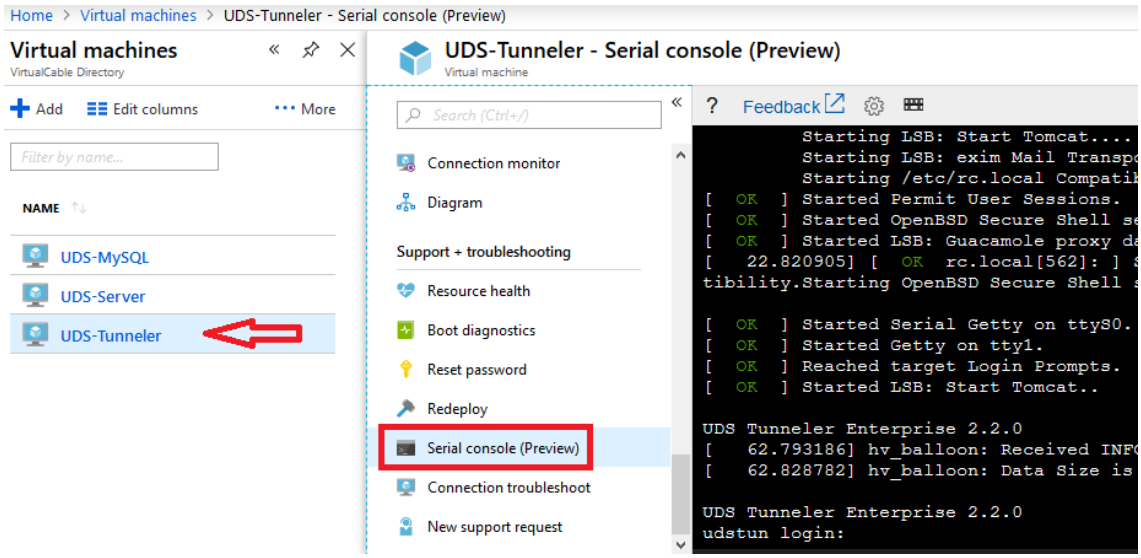
o UDS Tunneler configuration

The UDS Tunneler component is the element that will provide secure access to virtual desktops through the Internet. He will also be responsible for establishing the HTML5 connection (HTM5 Transport for desktops and vApps). It has a configuration wizard that we will run automatically. To do this it will be necessary to indicate a series of data in the configuration file of the server.

NOTE: To make a valid configuration of the UDS Tunneler server, it is necessary to have a valid UDS server with connectivity. The communication between these two components will be made through port 80 by default. If you want to use port 443, it will be necessary to have a valid certificate on both servers.

NOTE: The UDS Tunneler server can be connected against any version of UDS (Enterprise, Free or Evaluation).

In order to access and execute the configuration wizard of the UDS Tunneler we will have to connect via “Serial console”. We will access the “Virtual machines” service, we will select the virtual machine that contains UDS Tunneler and in the “Support + troubleshooting” menu we will select “Serial console (Preview)”.



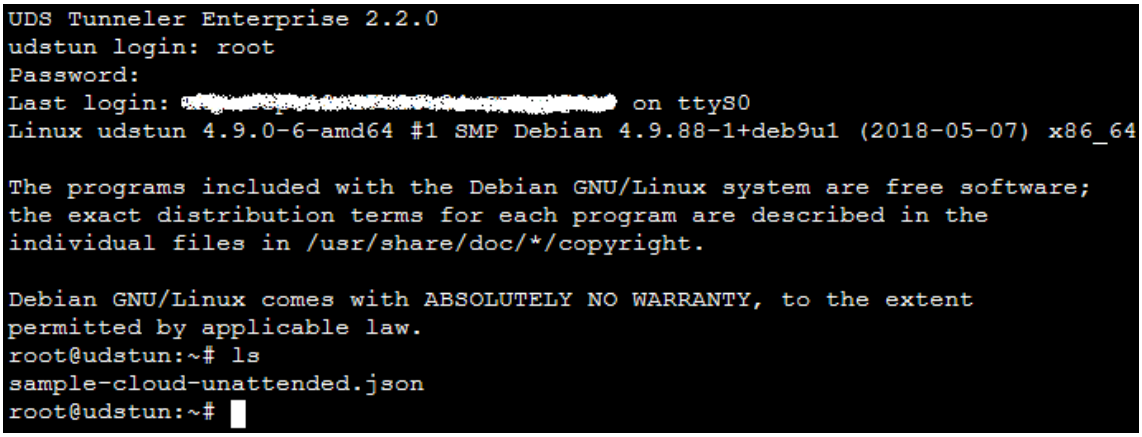
NOTE: The connection will take a few seconds and once connected, we must place the mouse inside and press the “enter” key.

We will validate on the UDS Server with the following credentials:

User: root

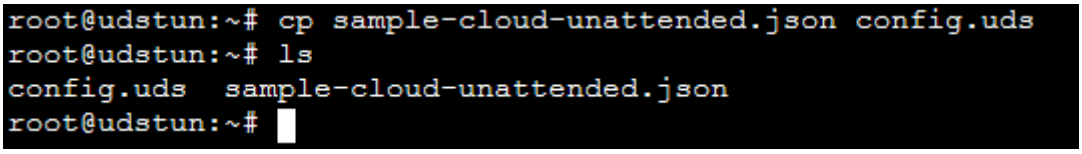
Password: uds

Once validated, we will confirm that in the /root directory we have an example configuration file with the name: “simple-cloud-unattended.json”.



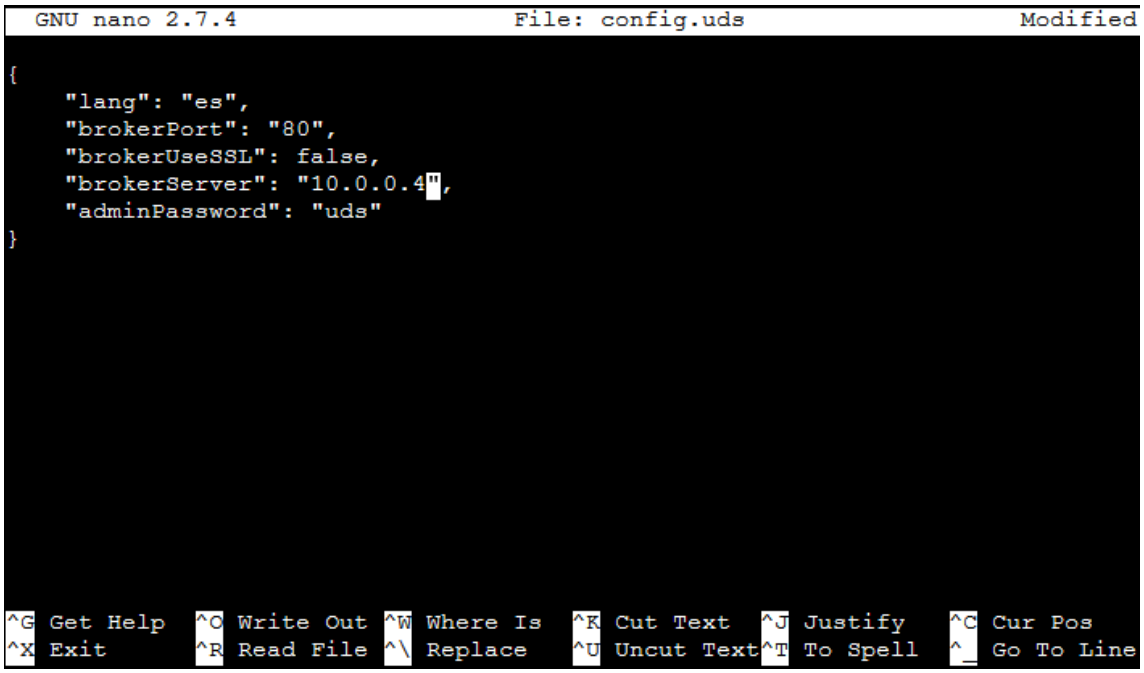
We will create a copy of this file and name it like: config.uds

cp sample-cloud-unattended.json config.uds



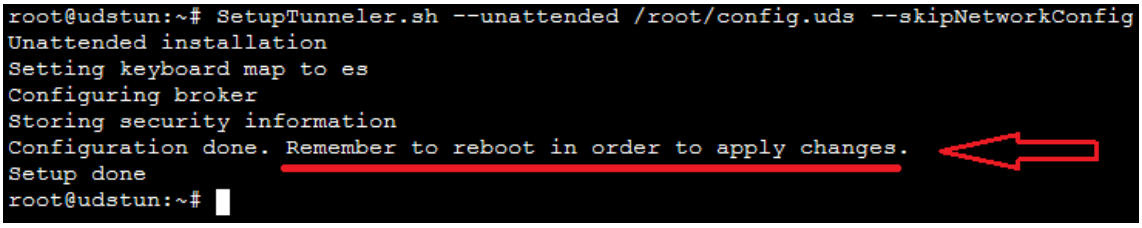
Once the new file is created, we will proceed to edit it and indicate the new configuration data:

- **lang:** Language of keyboard layout
- **brokerPort:** connection port with the UDS Server component (default 80)
- **brokerUseSSL:** enable or disable the use of SSL (default false)
- **brokerServer:** IP address or name of the UDS Server component
- **adminPassword:** UDS Tunneler virtual machine root password (default uds)



We save the changes in the file config.uds and proceed to its execution. To do this we must start the following command:

SetupUDS.sh --unattended /root/config.uds --skipNetworkConfig



Once the configuration process has been successfully completed, **we must restart the server to apply the new configuration**.

- ## Creating virtual machines as a base machine or template for UDS in Microsoft Azure:

In order for UDS to implement virtual desktops in the Azure platform, it is necessary to have a machine or base template on which the new self-generated desktops by UDS will be based. We can implement this base machine in different ways. Next, we will show a procedure that will allow us to migrate templates already installed and configured in other virtual platforms (vSphere, KVM, etc.) to the Azure platform.

The first thing to do is to have a disk image of the base virtual machine in .vhd format. There are many free tools. (like StarWind converter, qemu-img, etc…) that allow us to convert disks of different formats (vmdk de vmware, qcow2/raw de KVM, etc…) to .vhd format. It is very important to keep in mind that the disk image has to be with the total size (Fixed Size), format "Thin" is not supported (Dynamically Expanding).

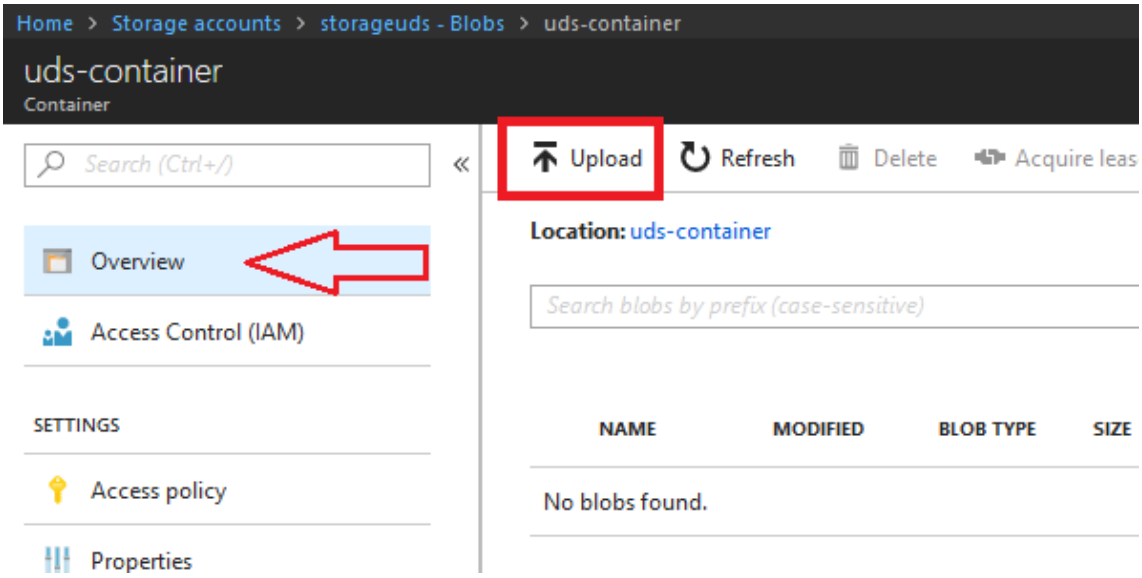
Before migrating the template machine, it is important that we make sure that it has a valid access mode (SSH or RDP type) in order to access it once it is hosted on the Azure platform (this platform does not have a console to be able to manage, configure, modify the machines). The base machine used in this example has enabled/installed access via SSH and RDP.

Another important point to keep in mind is the configuration of the network. It must be configured to carry the IP address through DHCP. In templates with S.O. Windows, it is necessary to have installed the valid network driver so that it is detected in the Azure platform (if the machine is exported from a Hyper-V platform it will already be integrated).

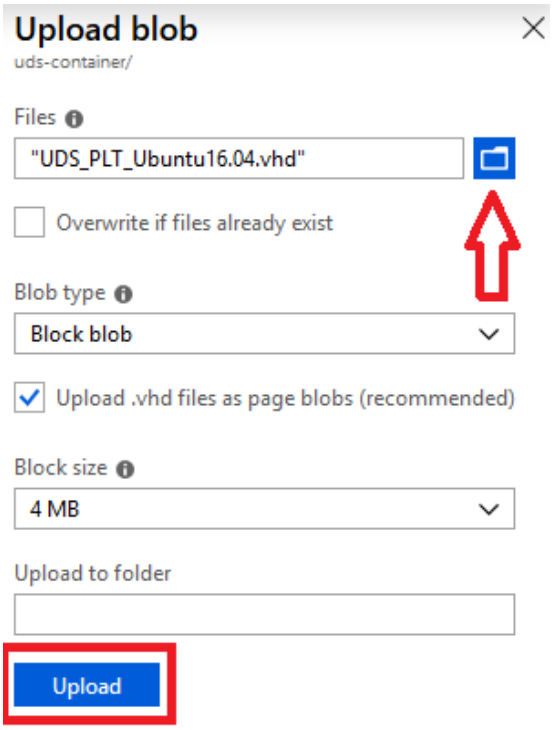
Once we have the disk image converted to the format supported by Azure (.vhd), we will load it into the platform and implement the new base machine. We will perform the following tasks described below (the procedure will be very similar to the one we use to implement the Appliances of UDS component):

- Uploading disk image .vhd to a “Container”

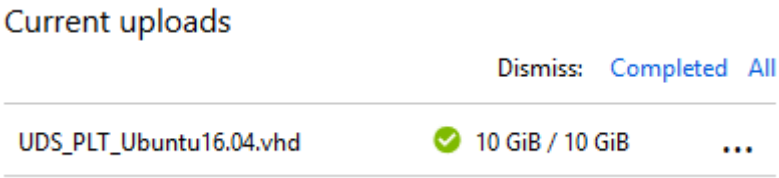
1. Access the “Container” (“Storage accounts”, in the “BLOB SERVICE” section, select “Blobs” and click on the “Container”) and click on “Upload”:



2. We indicate the disk image in the “Files” section. In “Blob type” we select “Page blob” and click on “Upload”.

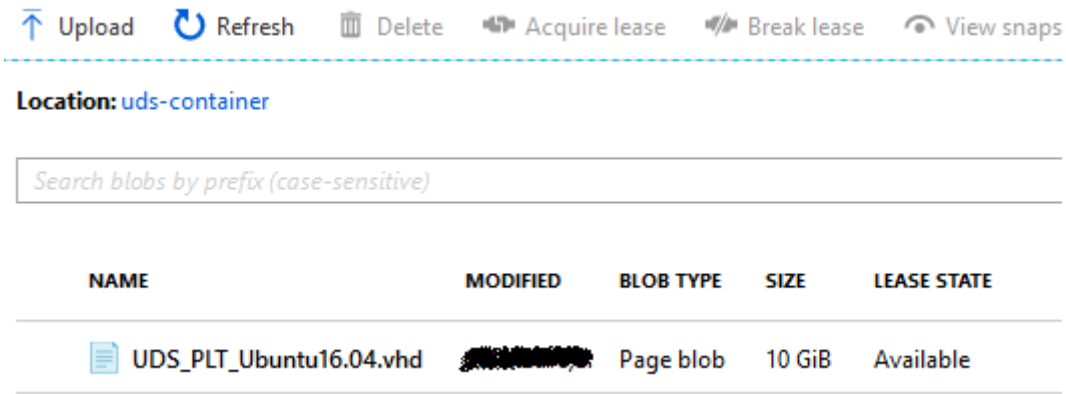


3. The image will start to be imported and we will have to wait until the upload process is finished. Once finished, we will proceed to the next task: the generation of a disk from the image.



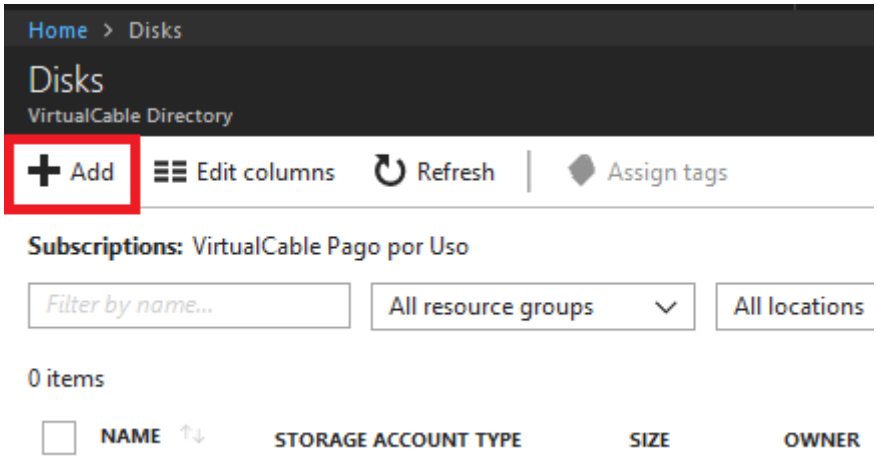
NOTE: Depending on the size of the disk images and the speed of the connection, this process could take several minutes.

Finally, we will see that within the “Container” we will have available the disk image of our base/template machine.

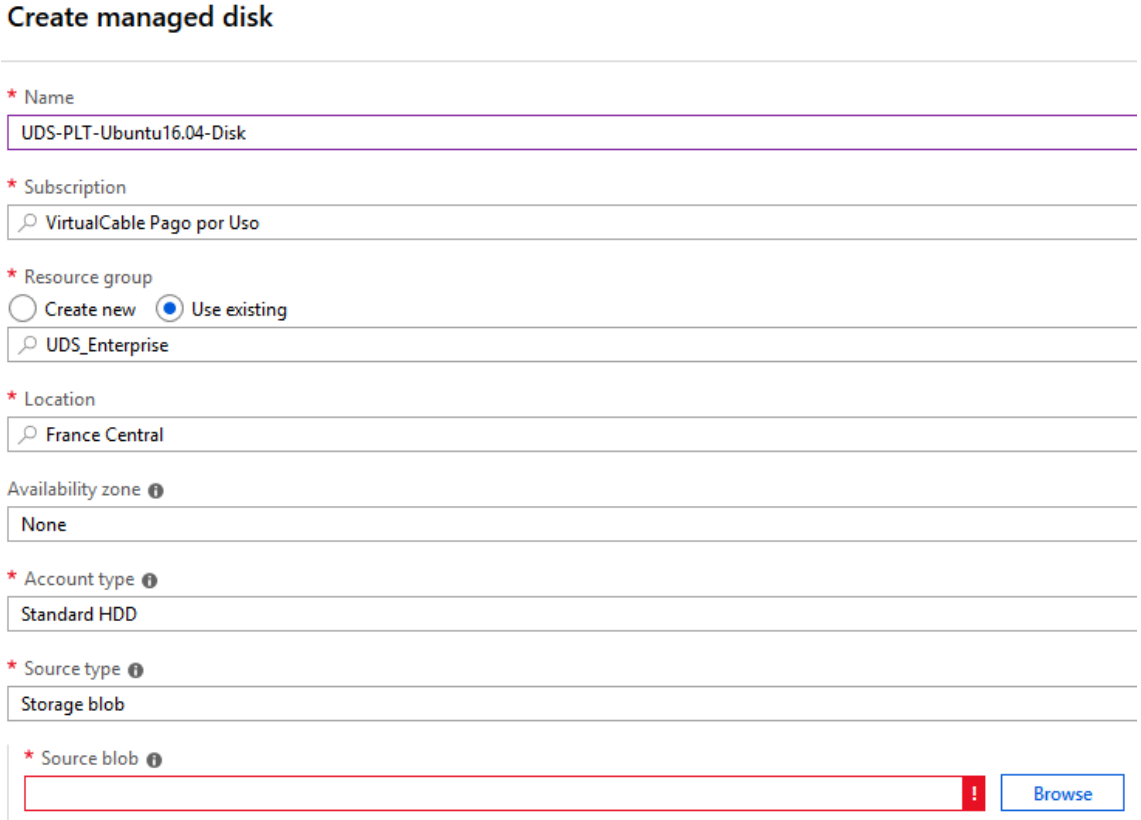


o Virtual disk creation

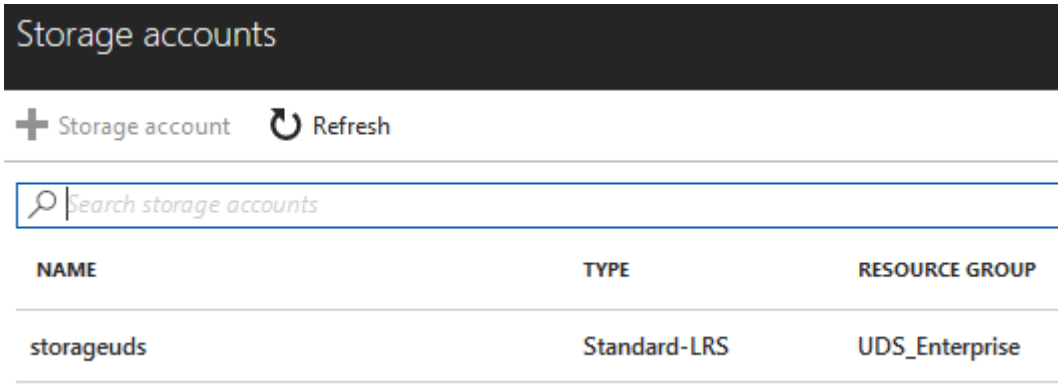
1. Access the “Disk” service and click on “add” to add a new disk.



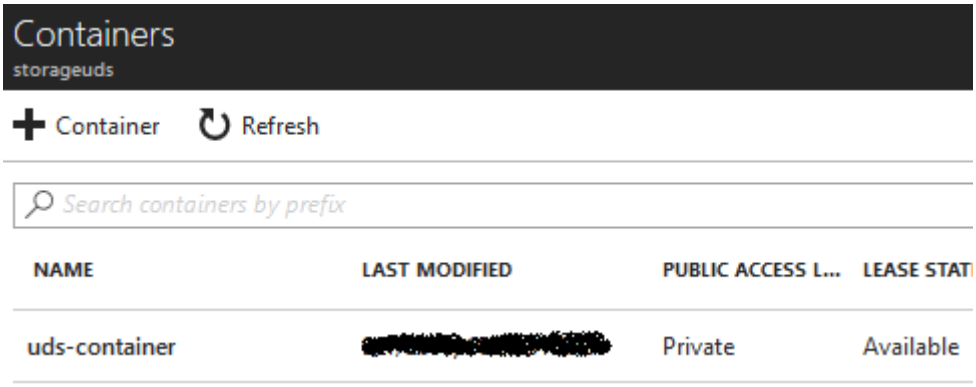
2. We indicate a descriptive name and a subscription, select the “Resource group” meant to UDS, “Location” and the “Account Type” based on the necessary performance:



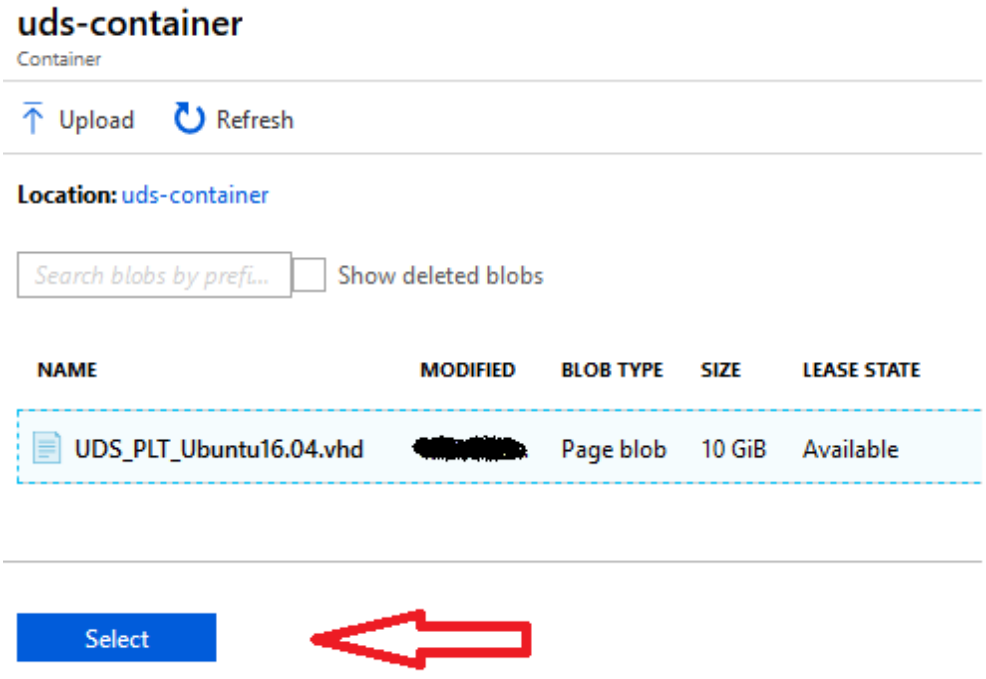
In “Source type” we must choose “Storage blob”. To indicate the image, click on “Browse” and select the “Storage accounts”:



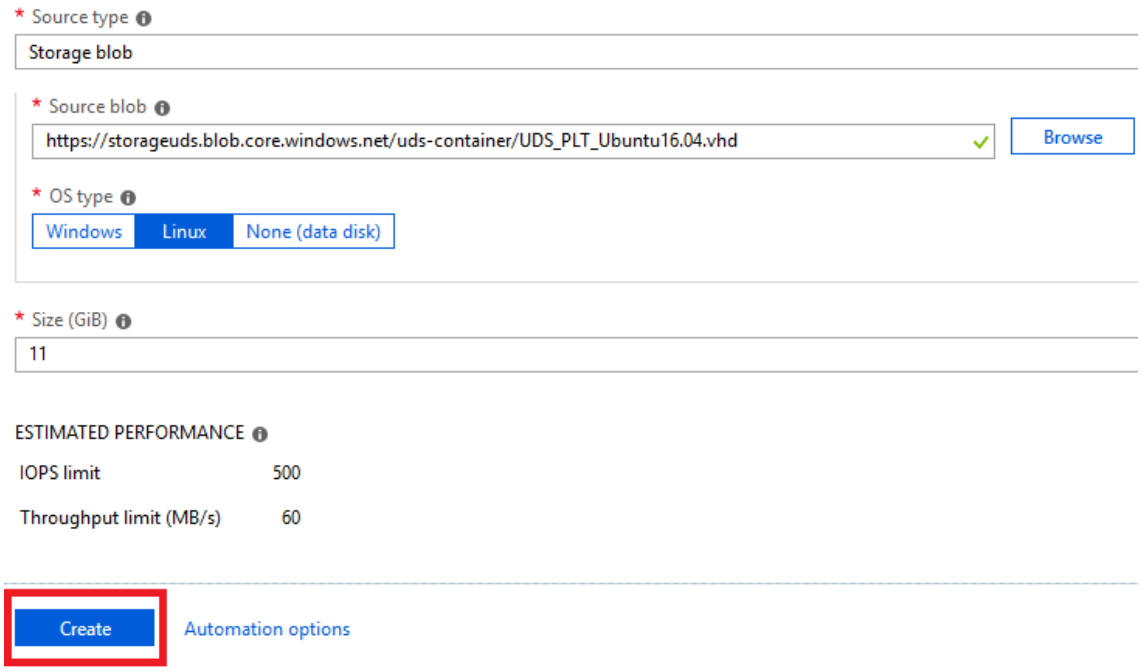
Once inside, we select the “Container” where we have uploaded the image of the template:



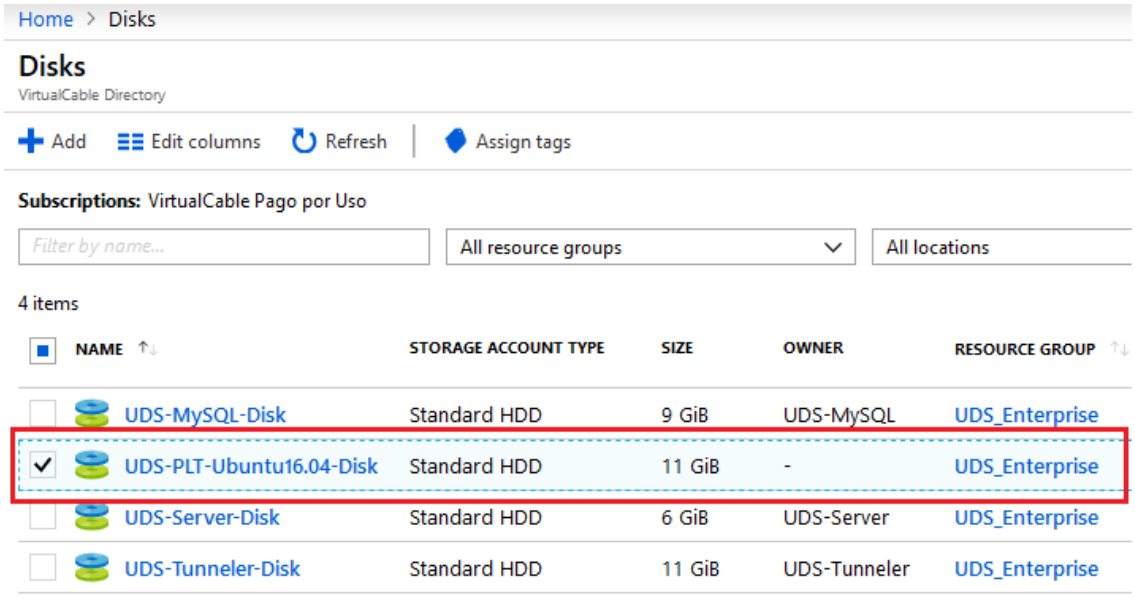
Finally we select the image of the template and click on “Select”.



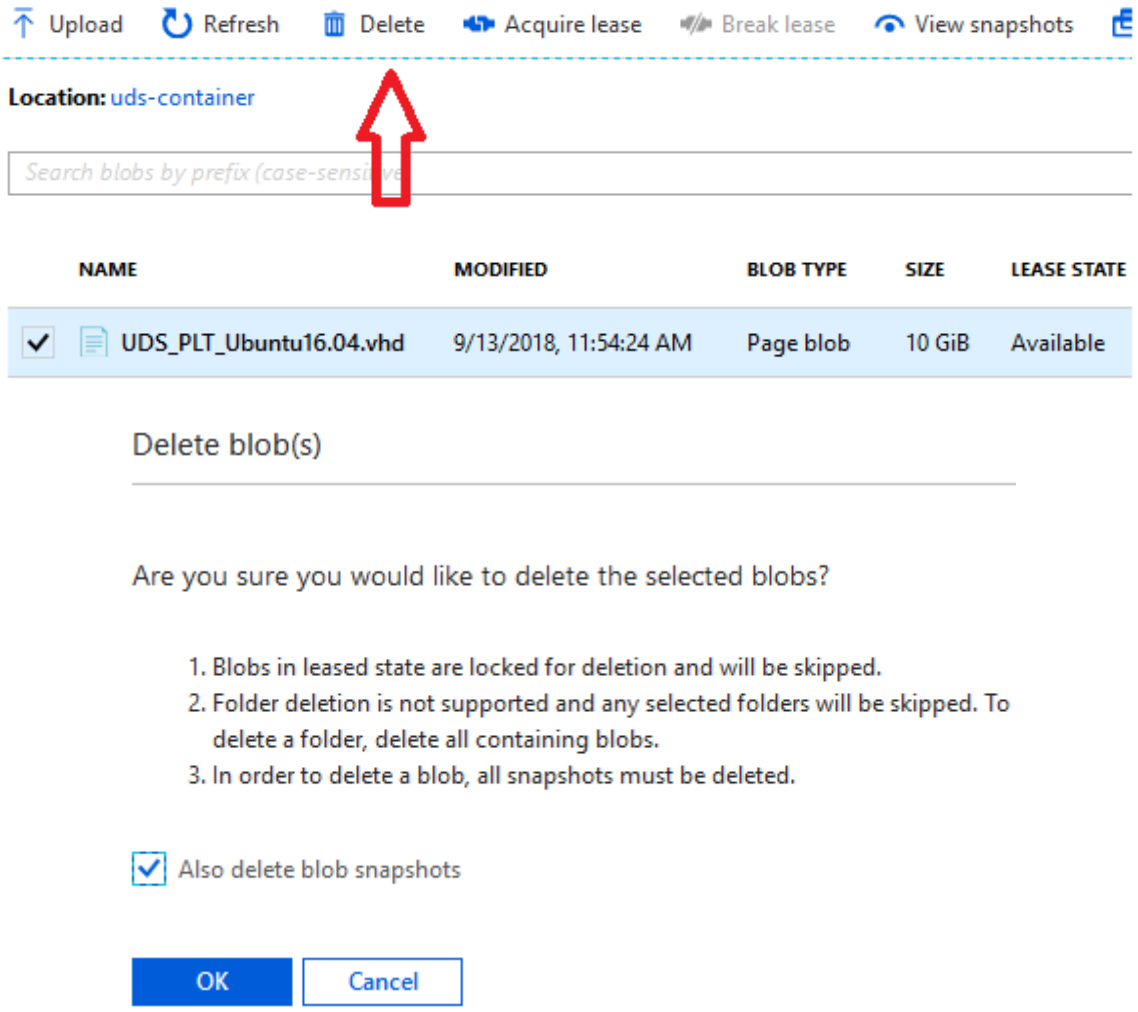
We will indicate “OS type” (in this case “Linux”) and enter the size of the disk in the “Size” field (we will always assign 1 GB more than size of the image) and finish by clicking on “Create”.



3. We wait for the disk to be created and, once the task is finished, we will see that we have it available to later generate the base/template machine using this virtual disk:



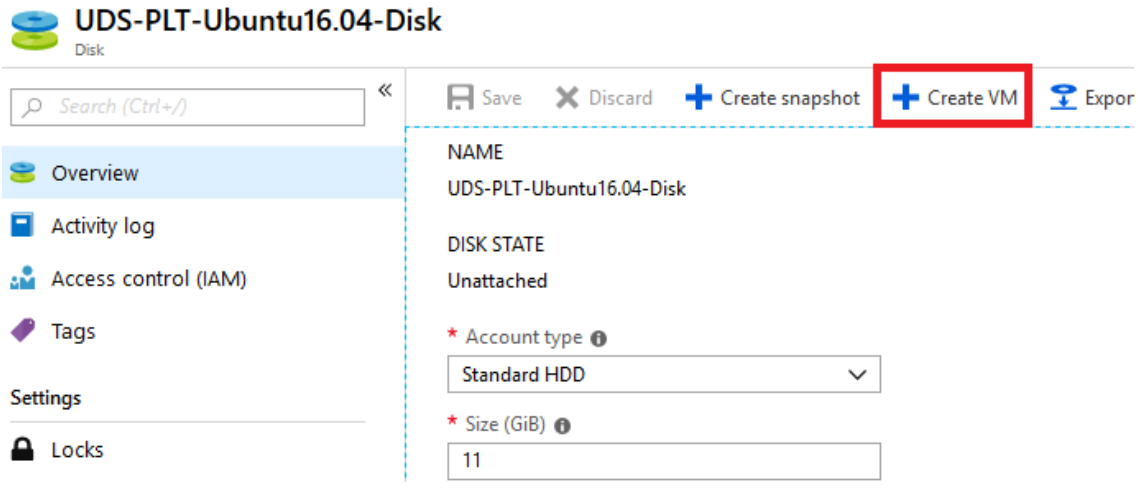
NOTE: Once the disk is deployed, we can delete the image of the “Container” to avoid generating unnecessary costs.



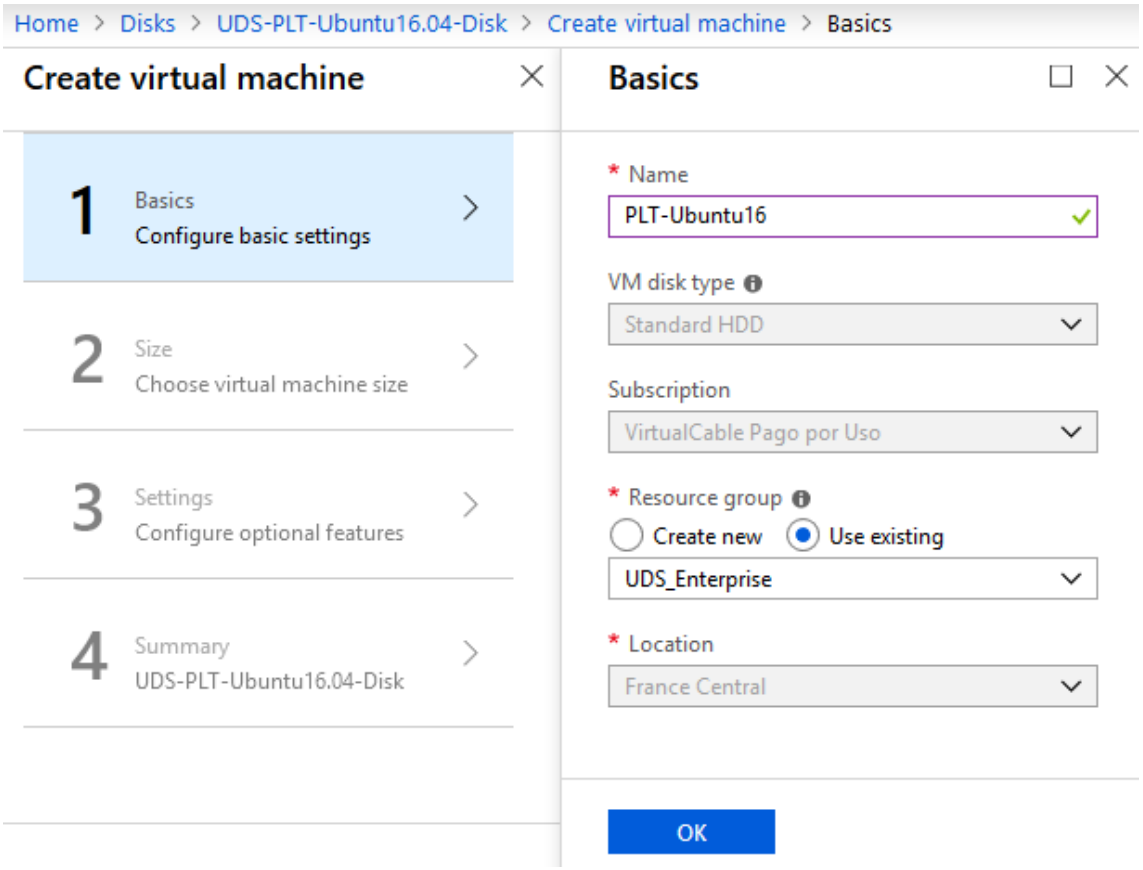
Creating virtual machine base

The creation of the base machine/template will be done from the disk itself:

1. Select the previously created disk (from the “Disk” service) and click on “Create VM”:

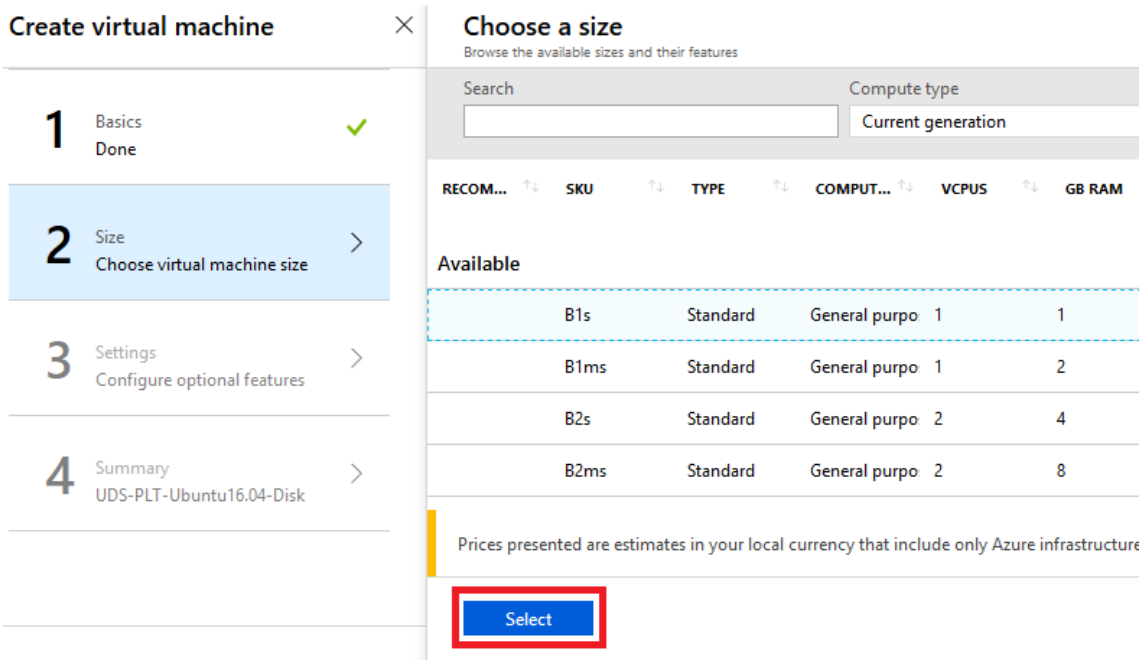


2. In the first step of the creation wizard we will indicate the name of the virtual machine (**it can not start with "UDS"**) and we will select the "Resource group" where we have assigned all UDS' services. Click on "OK".

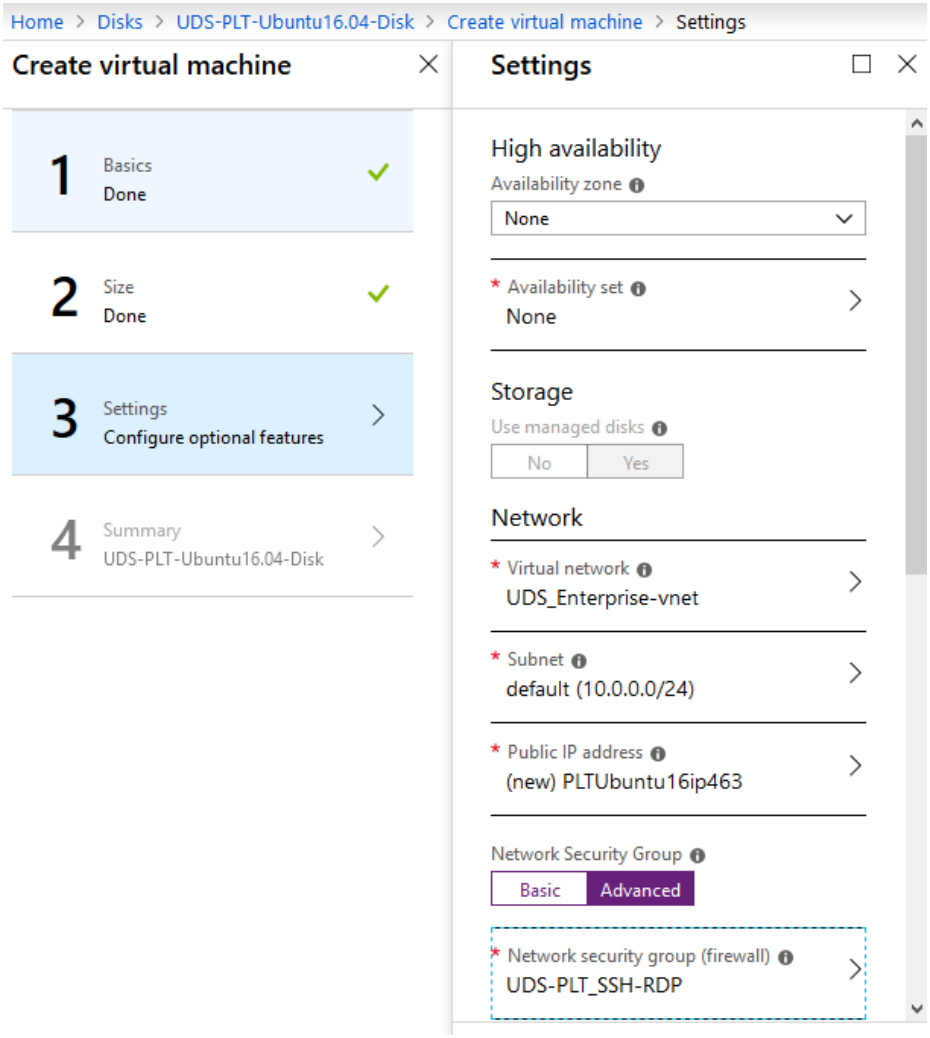


NOTE: The name of the template can't begin with the letters "UDS". If it starts with these letters, it will not be displayed nor will it be available in the UDS administration to be used as "base machine".

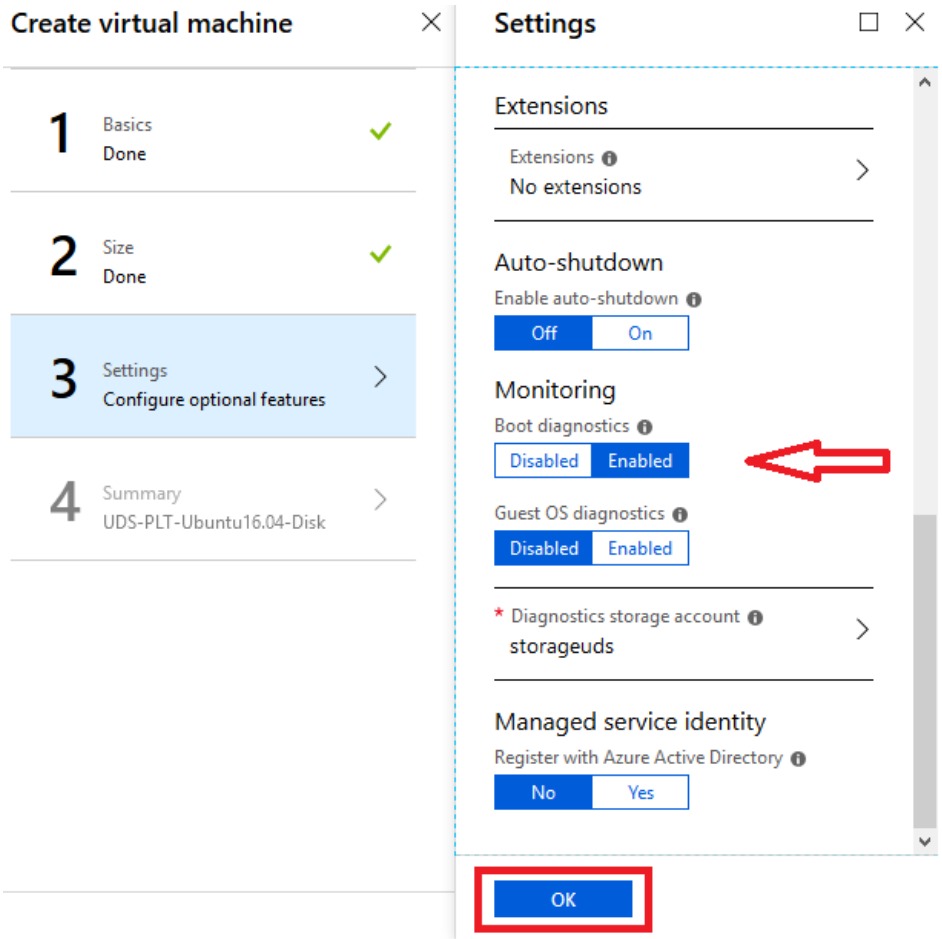
3. In the second step we select the type of the machine and the resources it will have. We choose the type and click on “Select” (in this example we will select for the template some minimum resources, B1s type).



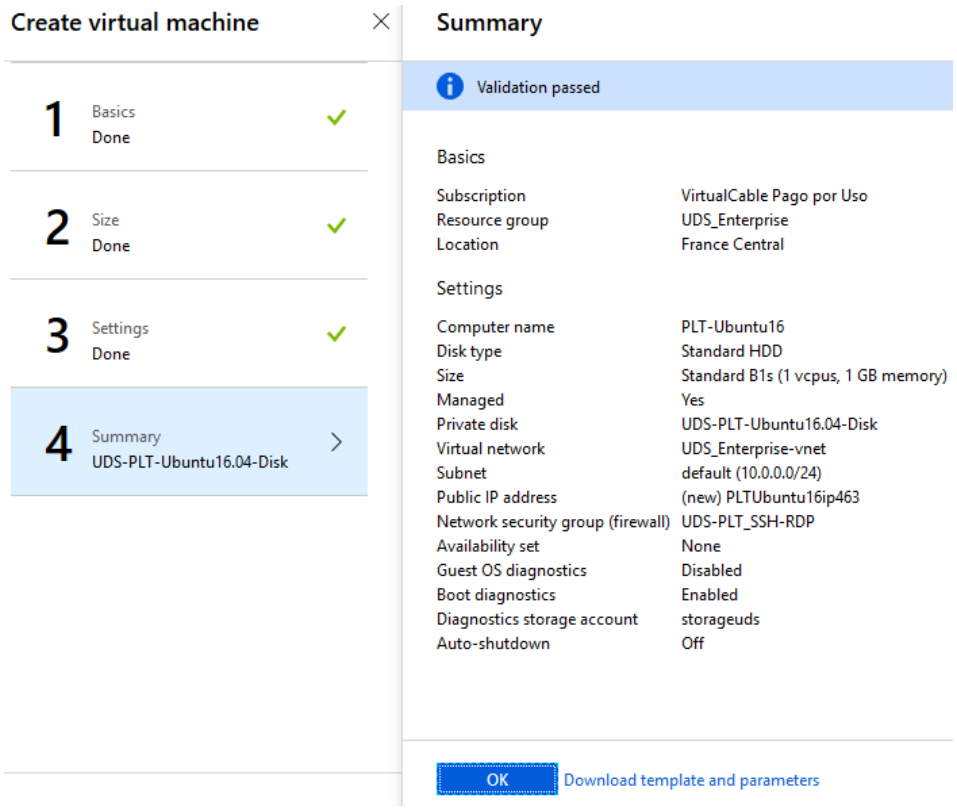
4. In the third step we will indicate the following necessary information in the Network section:
    a. **Virtual network:** If we already have a “virtual network” to connect the machines, we select it. Otherwise, we let the system create a new one (we can leave the name by default or indicate another).
    b. **Subnet:** If we have a “Subnet” that assigns IP addresses to the virtual machines, we select it. Otherwise, we let the system create a new one (we can leave the “default” or indicate a specific one).
    c. **Public IP address:** We will let the system assign a public IP address to the server. The main reason will be to be able to access the template via RDP or SSH, to be able to install new software (including the UDS Actor), maintenance or updates on the base machine. If the disk image already has the UDS Actor configured and you do not want to make any access, you can not indicate any public IP address by selecting “None”.
    d. **Network Security Group:** If we have assigned a “Public IP”. We must select the “Advanced” option and assign or create a “Network security group” to access the template. In this example, one that allows access through ports 22 (SSH) and 3389 (RDP) has been assigned.



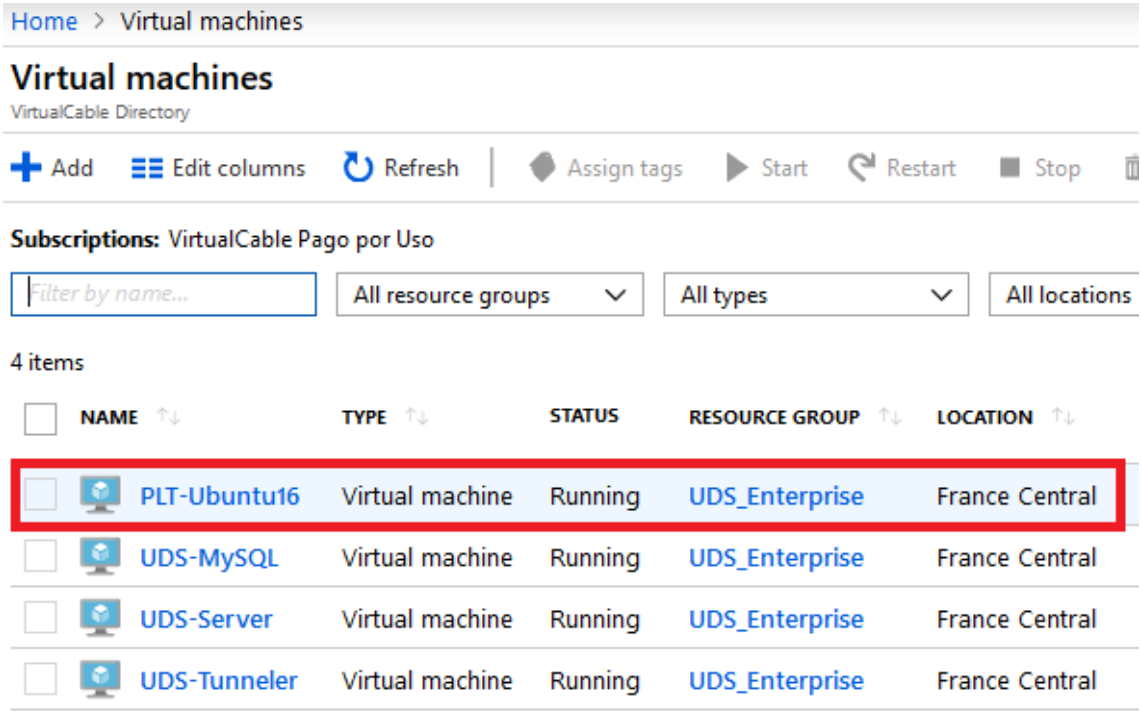
In step 3 we can also activate in the “Monitoring” section, the “Boot diagnostics” option, which will allow us to view a screenshot of the boot and subsequent status of the virtual machine. Click on “OK” to finish step 3:



5. In step 4 and last, we will check that all the configuration is correct and click “OK” to create the virtual machine.

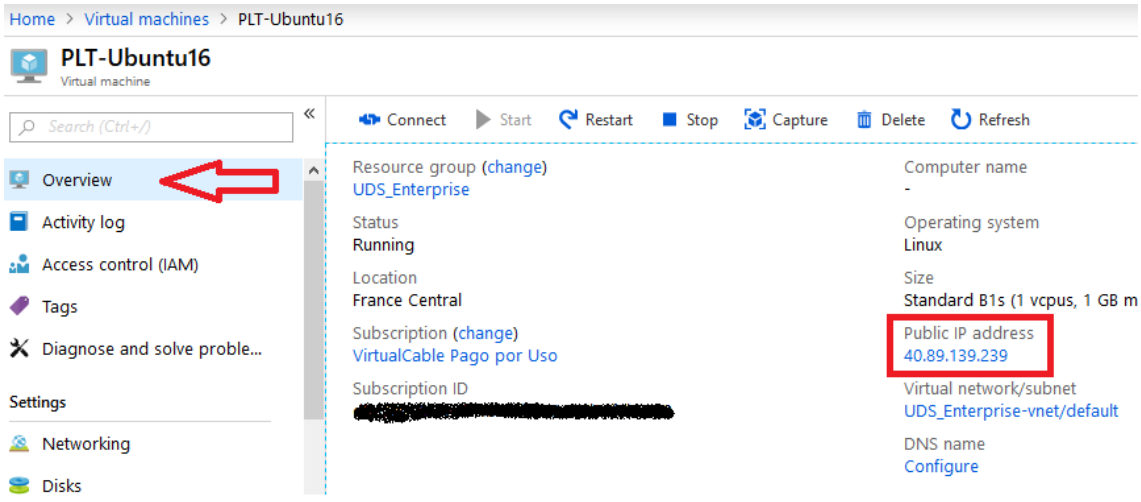


6. Once the process of creating the new VM is finished, we will check that we already have the new machine in the “Virtual machines” service. We will have to see the new virtual machine created and turned on:

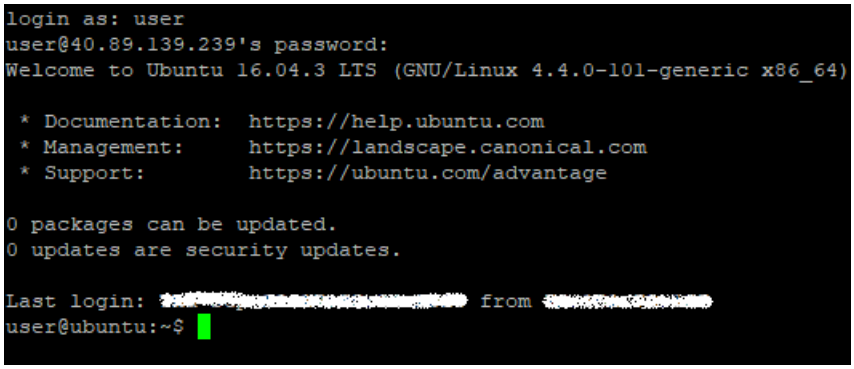


- Access and configuration of basic machine or template

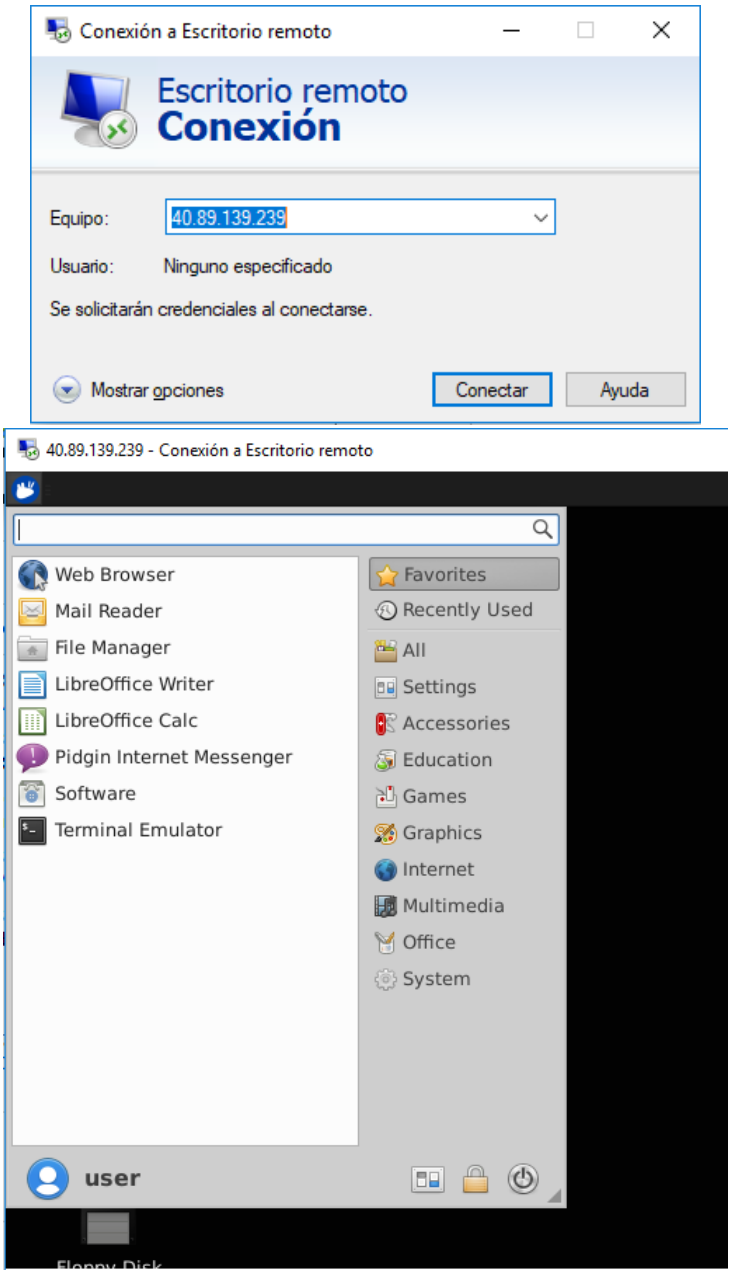
Once the virtual machine is deployed, we must be able to access it. To know what the public IP address of the machines is, we will click on it in the “Virtual machines” service. In the section “Overview” we will look at the value of “Public IP address”.



In this example we will connect through RDP to be able to perform the installation of the UDS Actor, although, as we indicated earlier, the “Network security group” assigned also allows us to connect through SSH:



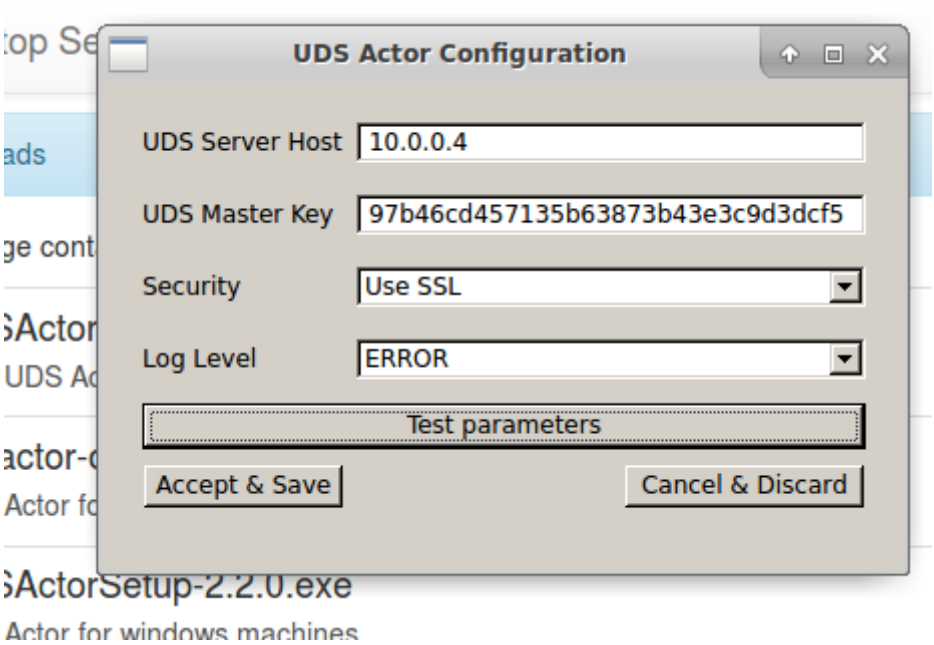
We access the template through RDP to install and configure the UDS Actor:



NOTE: You can consult the UDS Enterprise installation, administration and user manual in the documentation section of the UDS Enterprise website for more details on the installation of UDS Actor.

During the configuration of the UDS Actor, we can indicate in the connection data against UDS Server the address/local DNS name or also the IP or public DNS (in the case of using IP addresses instead of names, we must make sure that these addresses are not dynamic, since they can change with the on/off of the virtual machines).

In this example we will use the local IP address of the UDS Server:



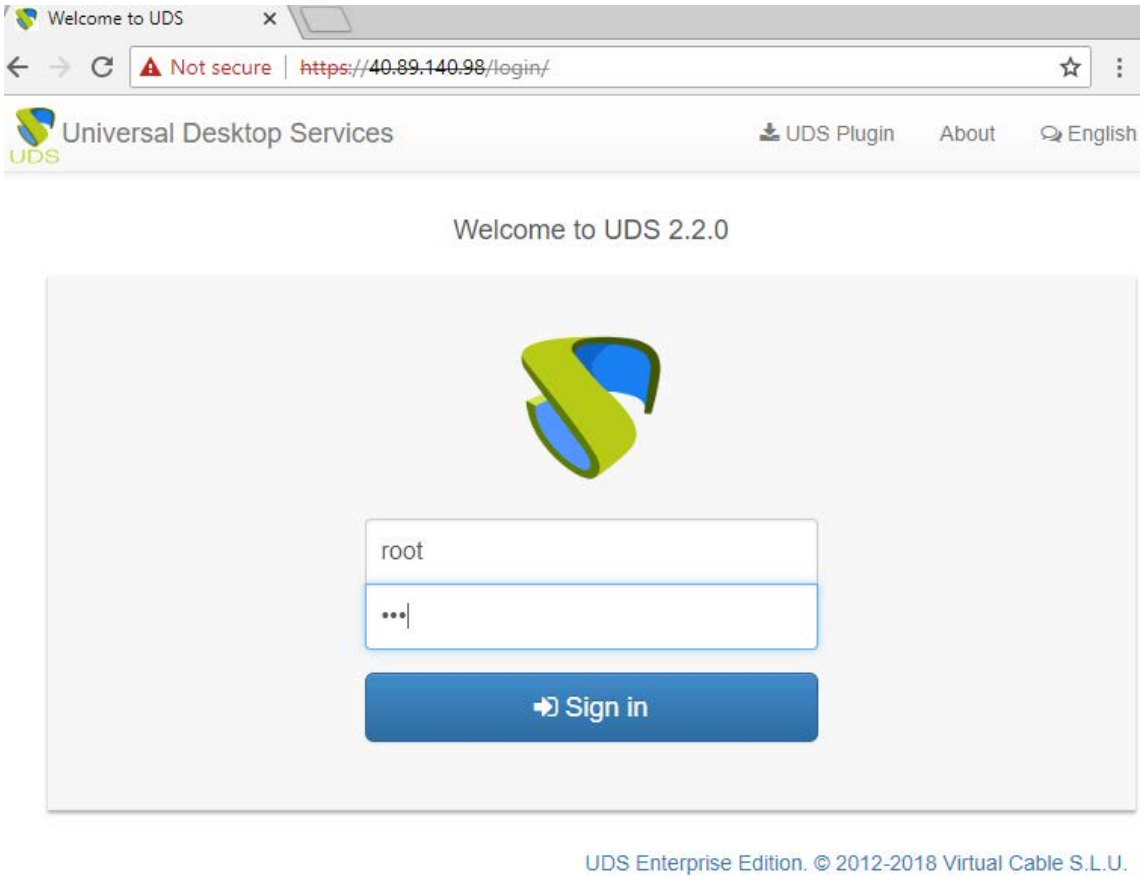
NOTE: It is necessary to verify through the “Test parameters” option that the connection data against UDS Server is correct.

Once all these tasks are completed, we can now **turn off the base machine or template** to be used with UDS Enterprise (it is not possible to publish a service if the base machine or template is turned on).

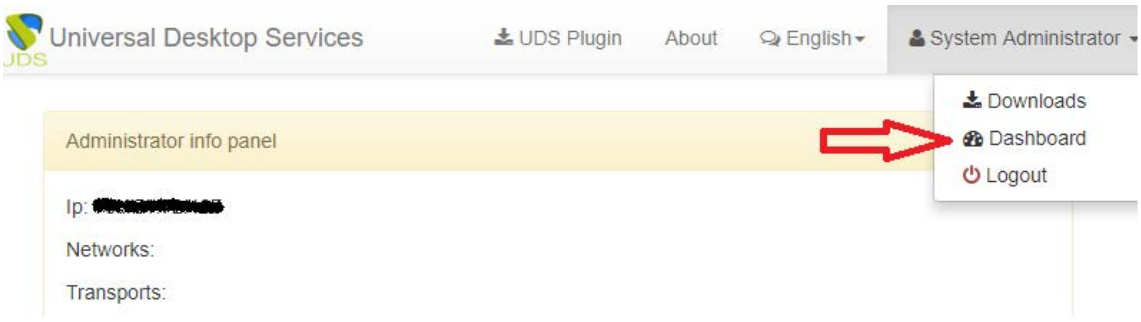
UDS Enterprise administration

Azure service provider integration

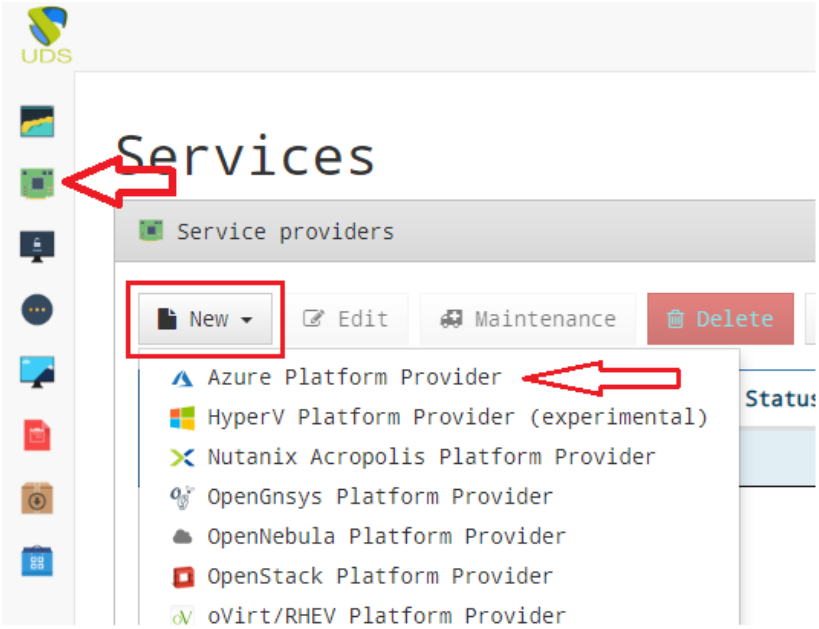
To perform the integration of Azure as a UDS Enterprise service provider, we should access the UDS administration dashboard. To do this, we access through web browser the public IP address or the name of the UDS Server using port 443. We log in with an administrator user (in this case we will use the administrator of the system user indicated in the configuration script of the UDS Server).



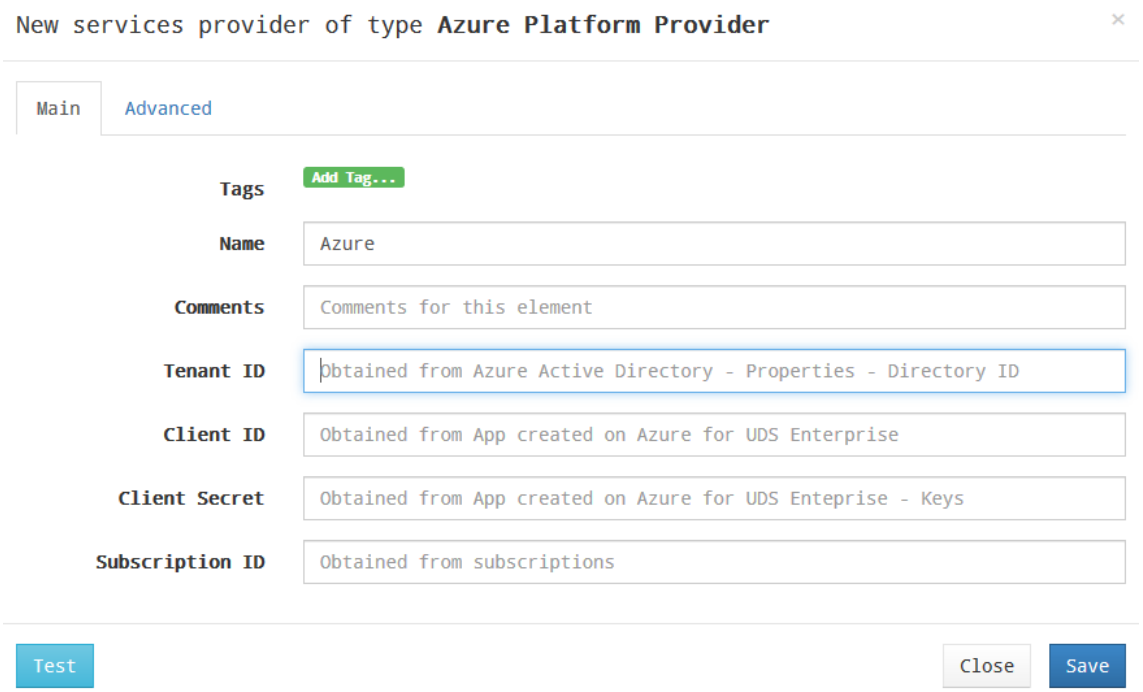
Once validated in the UDS access portal, we will access the “Dashboard” from the user's menu.



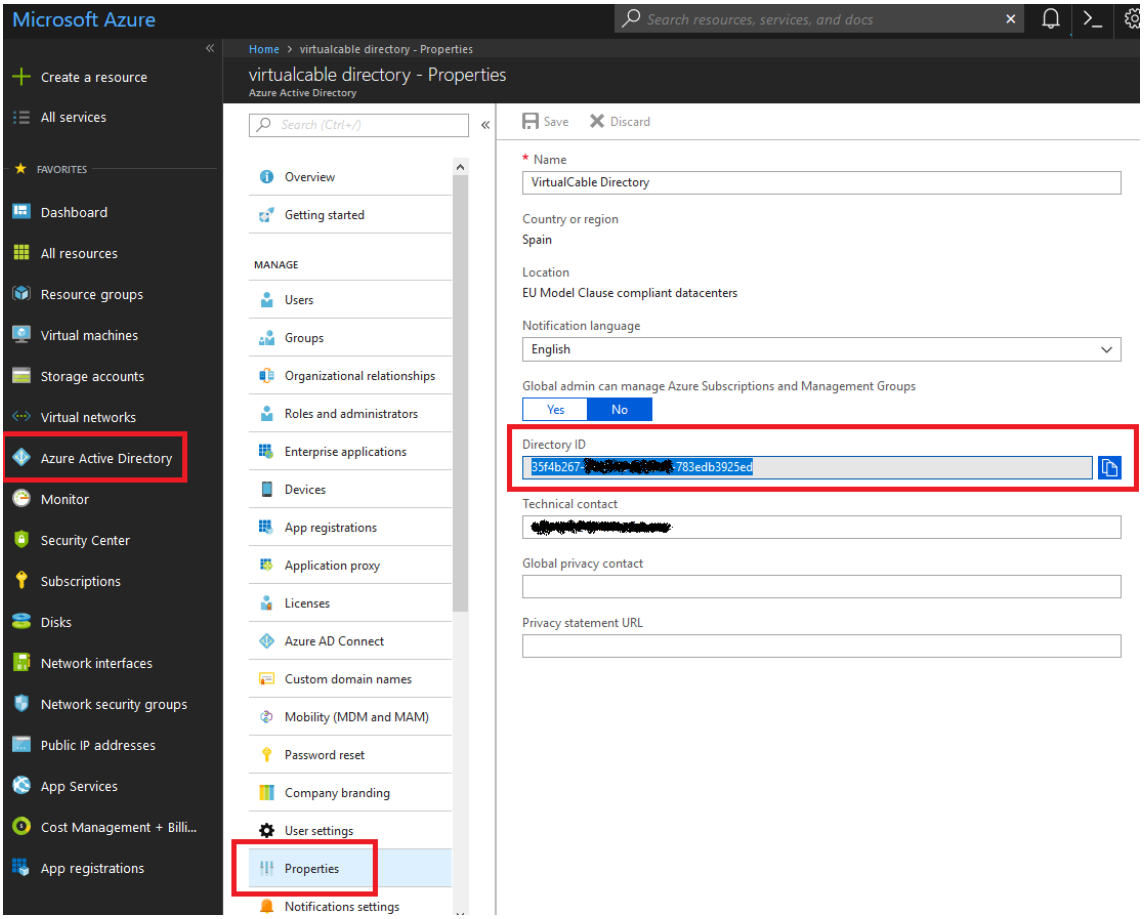
Within the UDS administration, we access the “Services” menu and click on “New” to register a new “Service provider”. We select “Azure Platform Provider”.



In order that UDS can connect to the Azure platform and be able to deploy virtual desktops automatically, it will be necessary to indicate a descriptive name and a series of data that we can obtain directly from this platform:

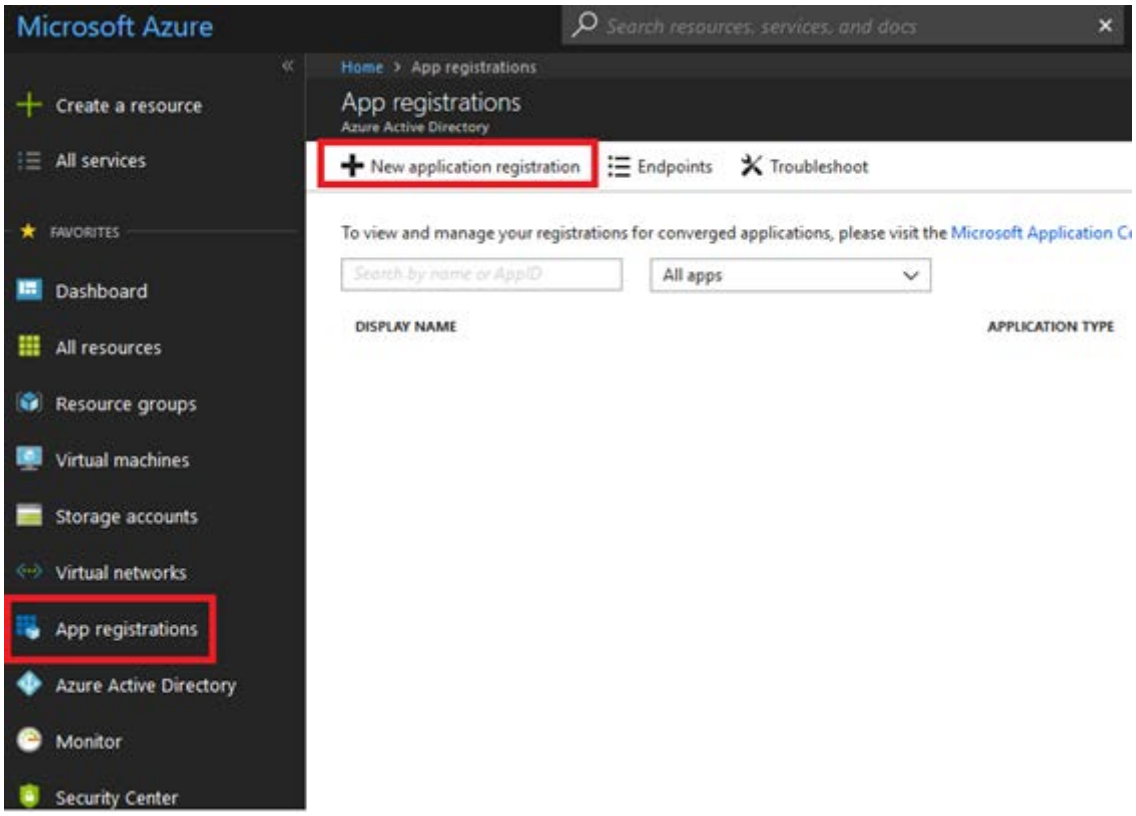


- **Tenant ID:** This value can be obtained from the service “Azure Active Directory”, “Properties”, “Directory ID”.



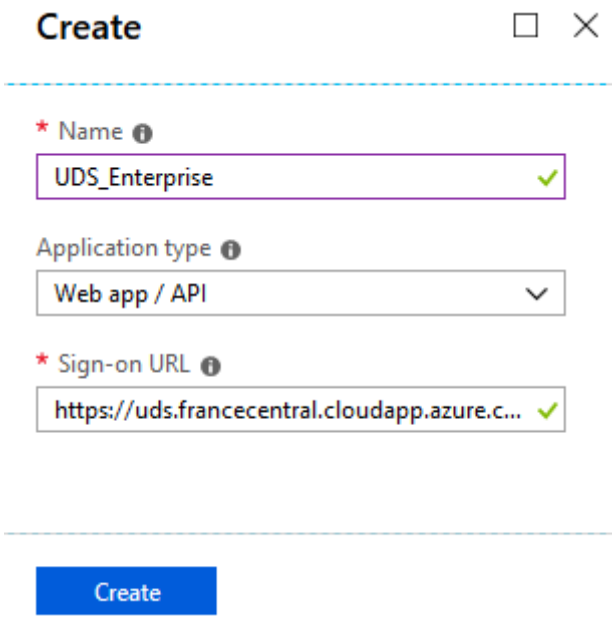
- **Client ID:** To obtain this value, it will be necessary to create a new “Application registrations” and give permission for our Azure subscription.

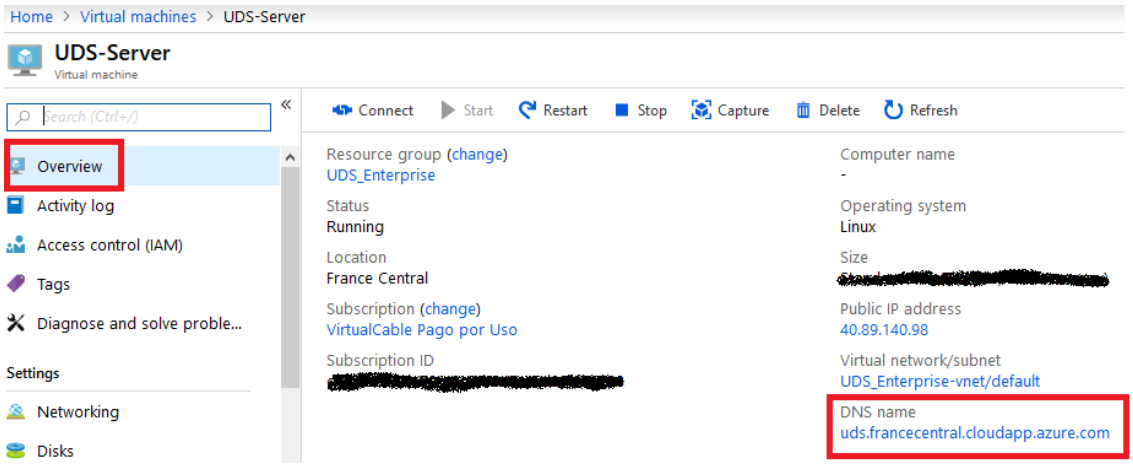
To register the application, we will go to the service “App registrations” and click on “New application registration”.



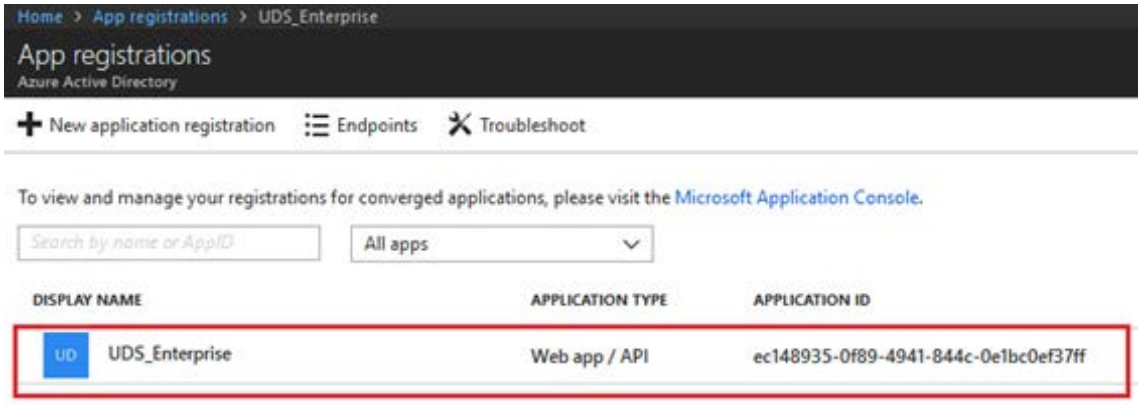
In the creation wizard we will indicate:

- **Name:** Name of the application
- **Application type:** We select “Web app / API”
- **Sign-on URL**: In this field we will indicate the access URL of UDS Server. We can obtain this URL from the virtual machine itself, in “Overview” – “DNS name”

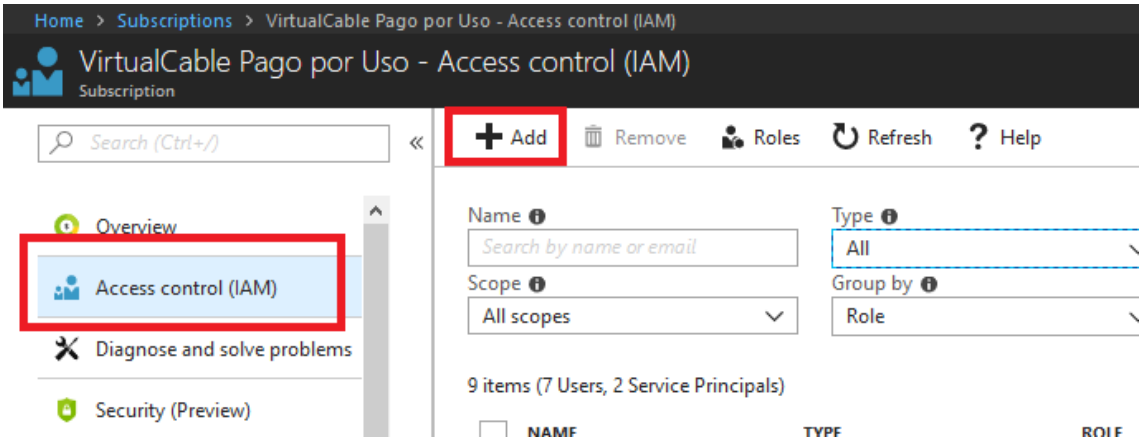




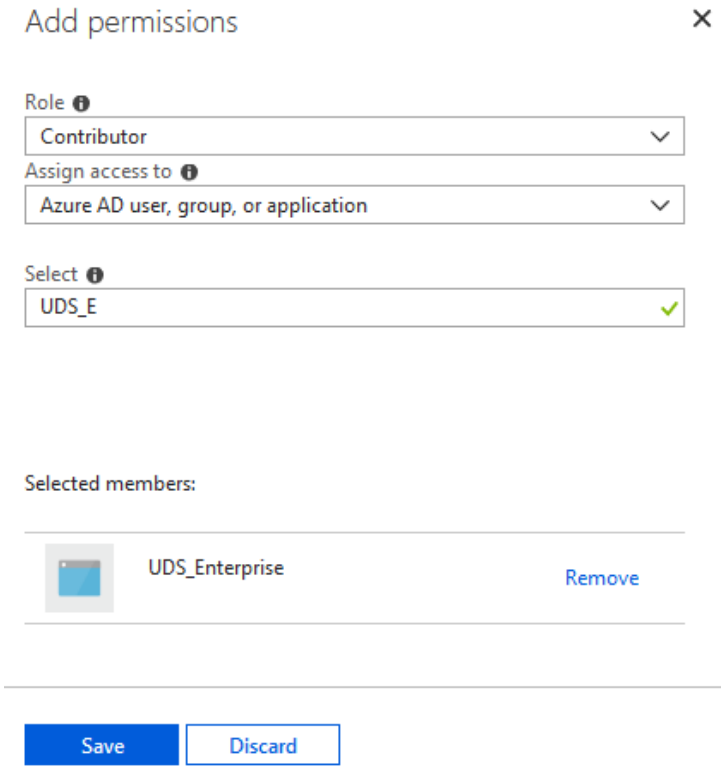
Once indicated all data of the application, we will click on “Create” and we will verify that it has been correctly created (if we do not see it, we will click on “View all applications”):



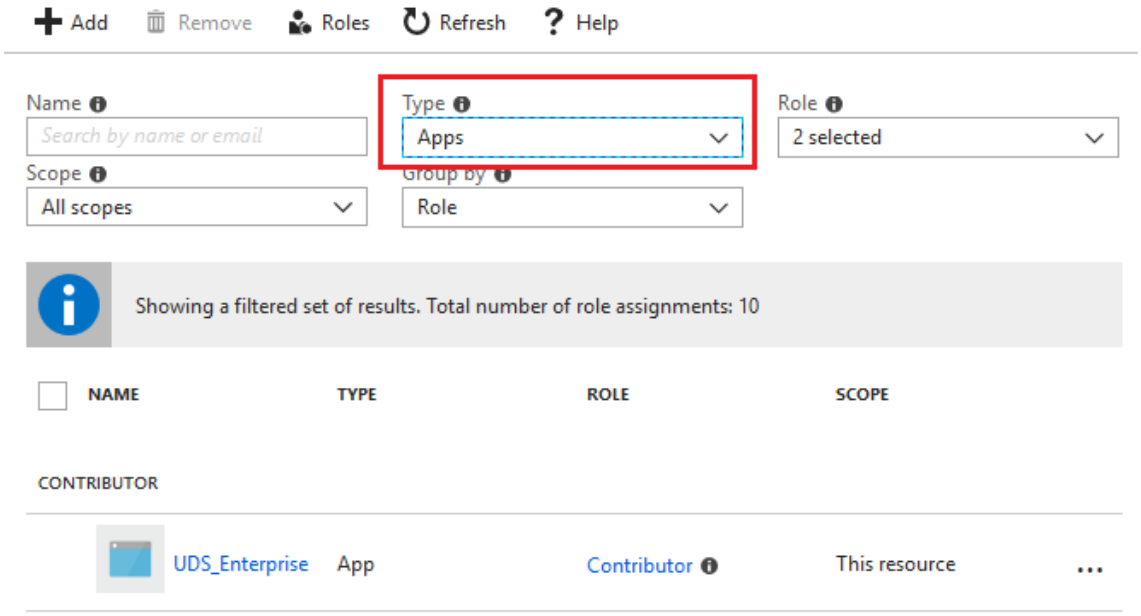
The column “APPLICATION ID” will indicate the “Client ID” that we will have to copy in UDS. In order to have a valid Client ID that can be used by UDS, we must give it permission on our subscription. To do this, we select our subscription of Azure (service “Subscriptions”) and in the option “Access control (IAM)” click on “add”.



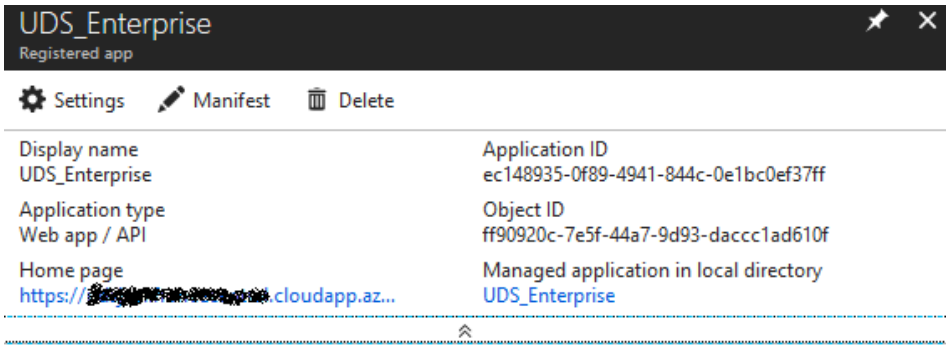
We indicate the role (in this case Contributor), we select that the access will be for “Azure AD user, group, or application” and we write the beginning of the application name registered in the previous step. Once it appears as available, we select it and click on “Save”



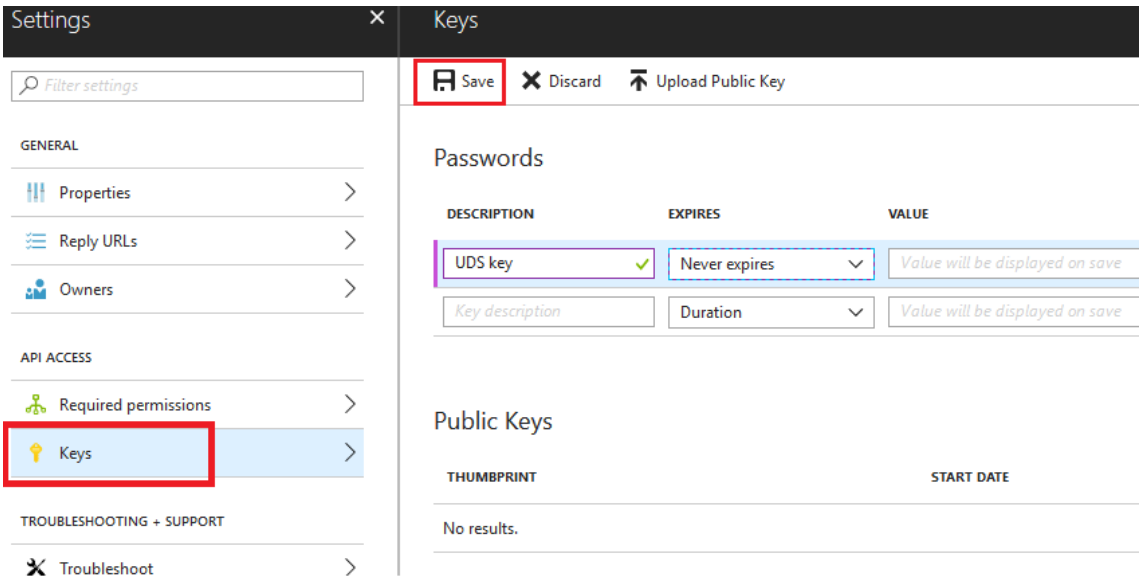
We can see the App with the assigned role:



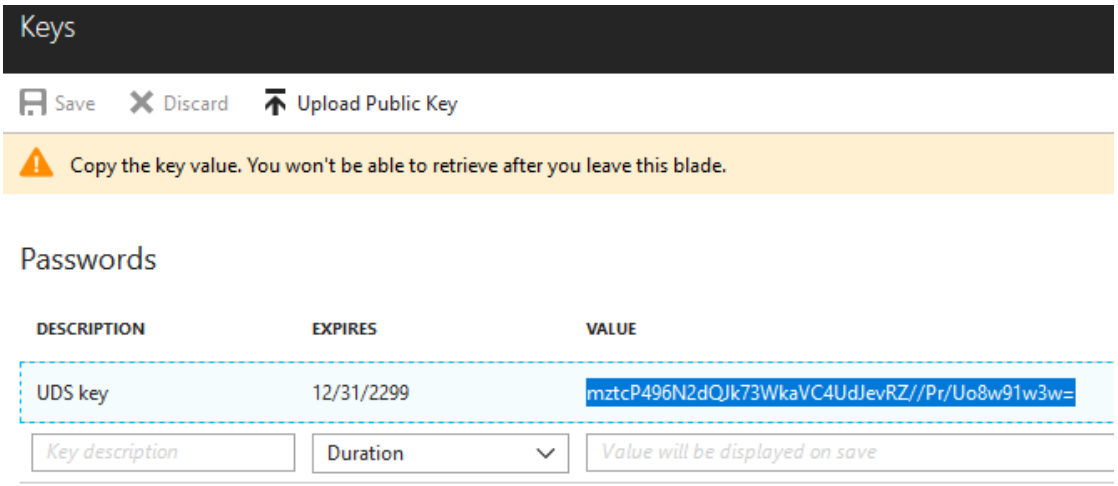
- **Client Secret:** This value will be obtained from the previously registered application. Click on it (in the “App registration”) and access “settings”.



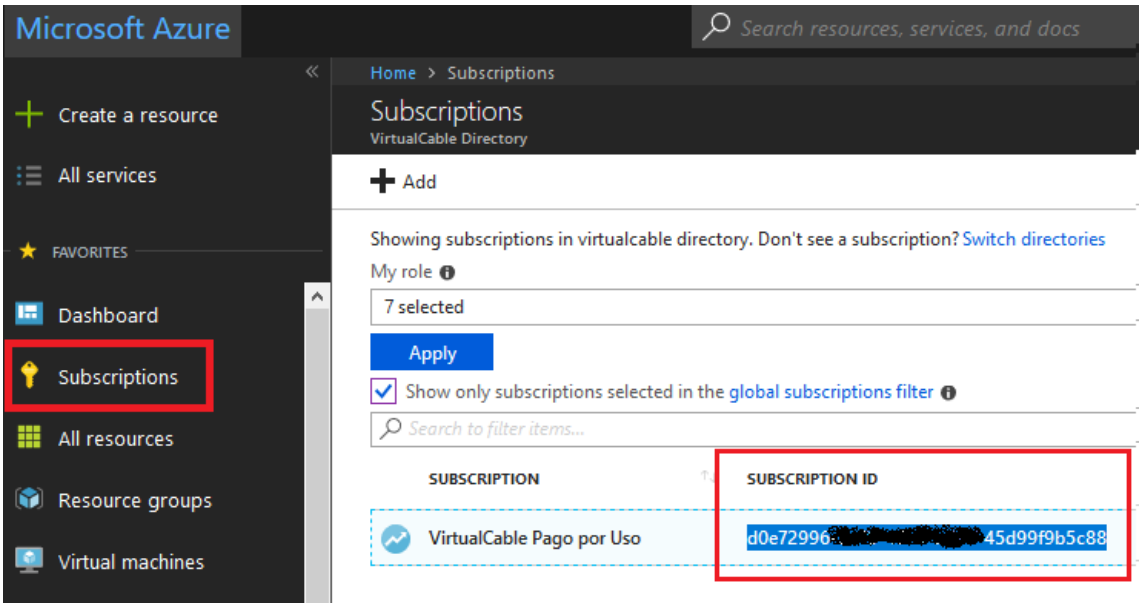
Inside “Settings” click on “Keys”. In the section “Passwords” we will indicate a description, we will select when it expires and we will click on “save” to copy the “key”:



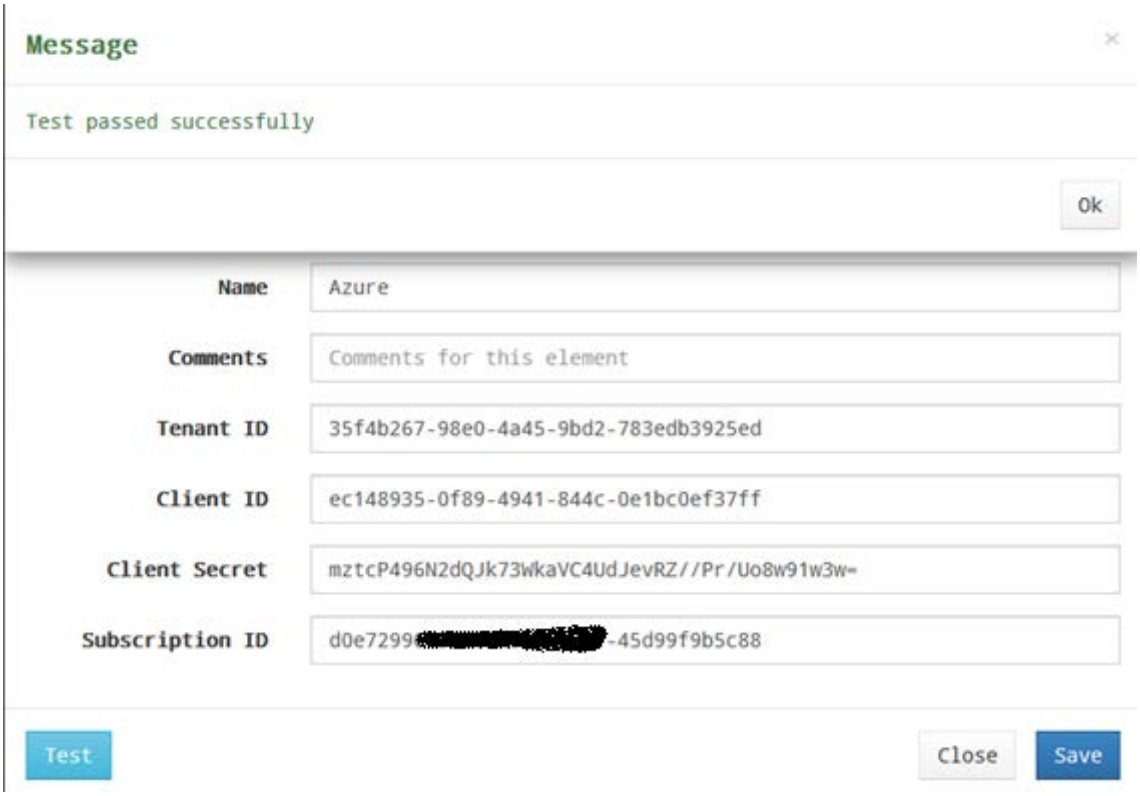
Once saved, it will allow us to copy the value (once this window is closed we will not be able to copy this value again, although we will be able to generate a new one if necessary) and we can use it as a Client Secret in UDS.



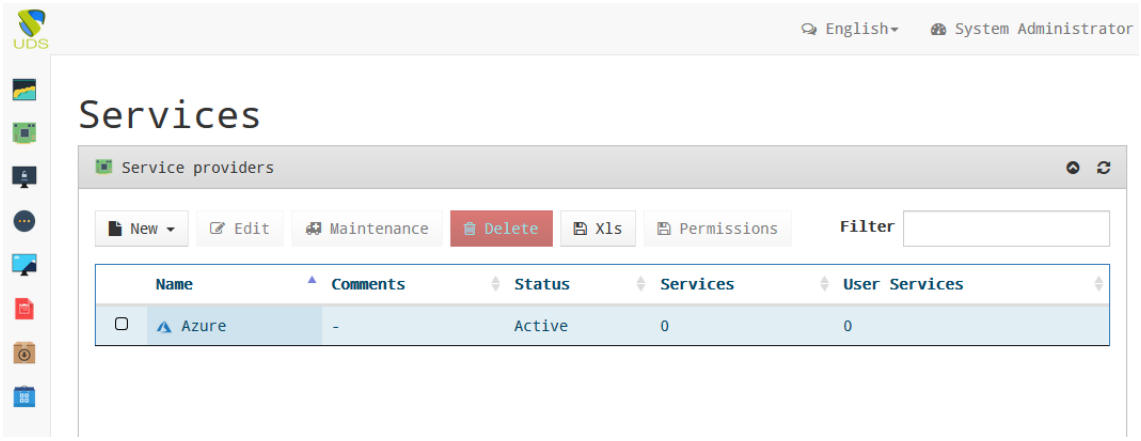
- **Subscription ID:** To obtain this value, we will access the “Subscriptions” service. We will identify our subscription and copy the value of “SUBSCRIPTION ID”:



Once all the fields are filled in, we will click on “Test” to verify that all the data are correct and we will save the parameters.

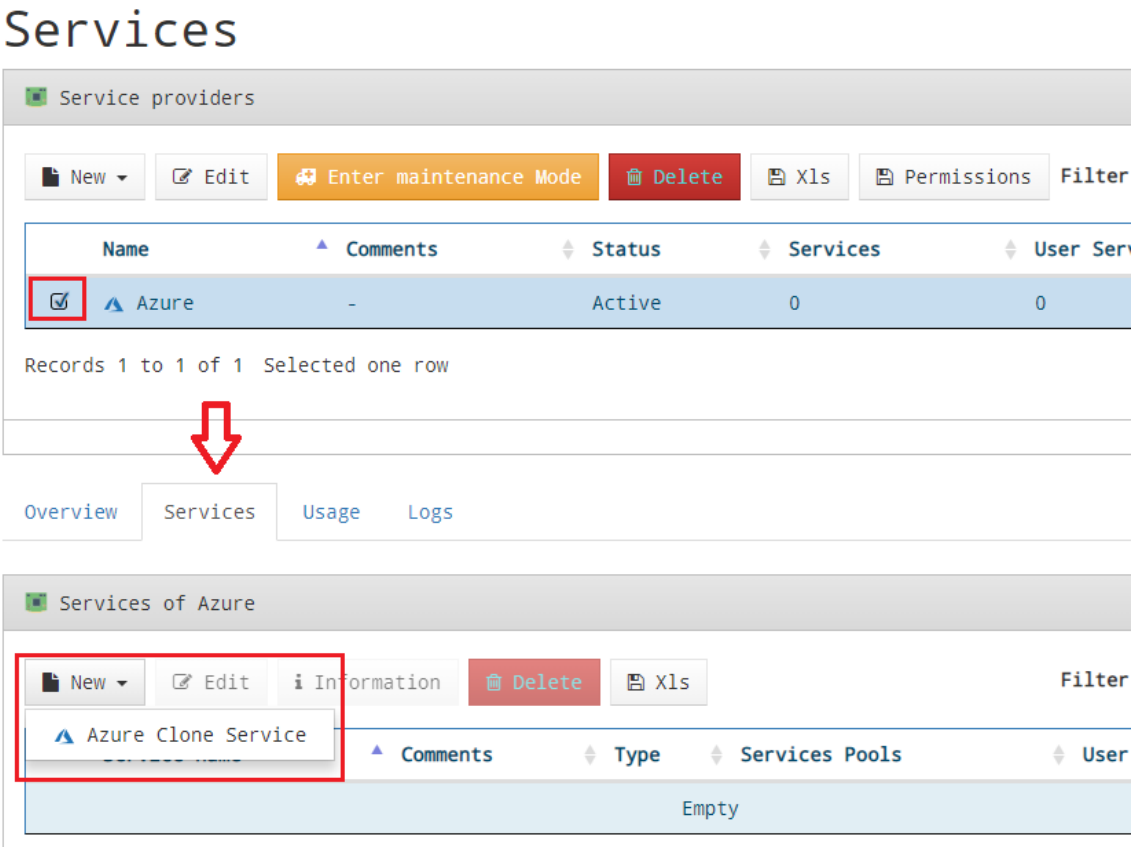


NOTE: Even if the test is not correct, we can save the provider and thus not lose the indicated data. Later you can check which of the values is the wrong one (the “Client Secret” will only be visible during its creation).



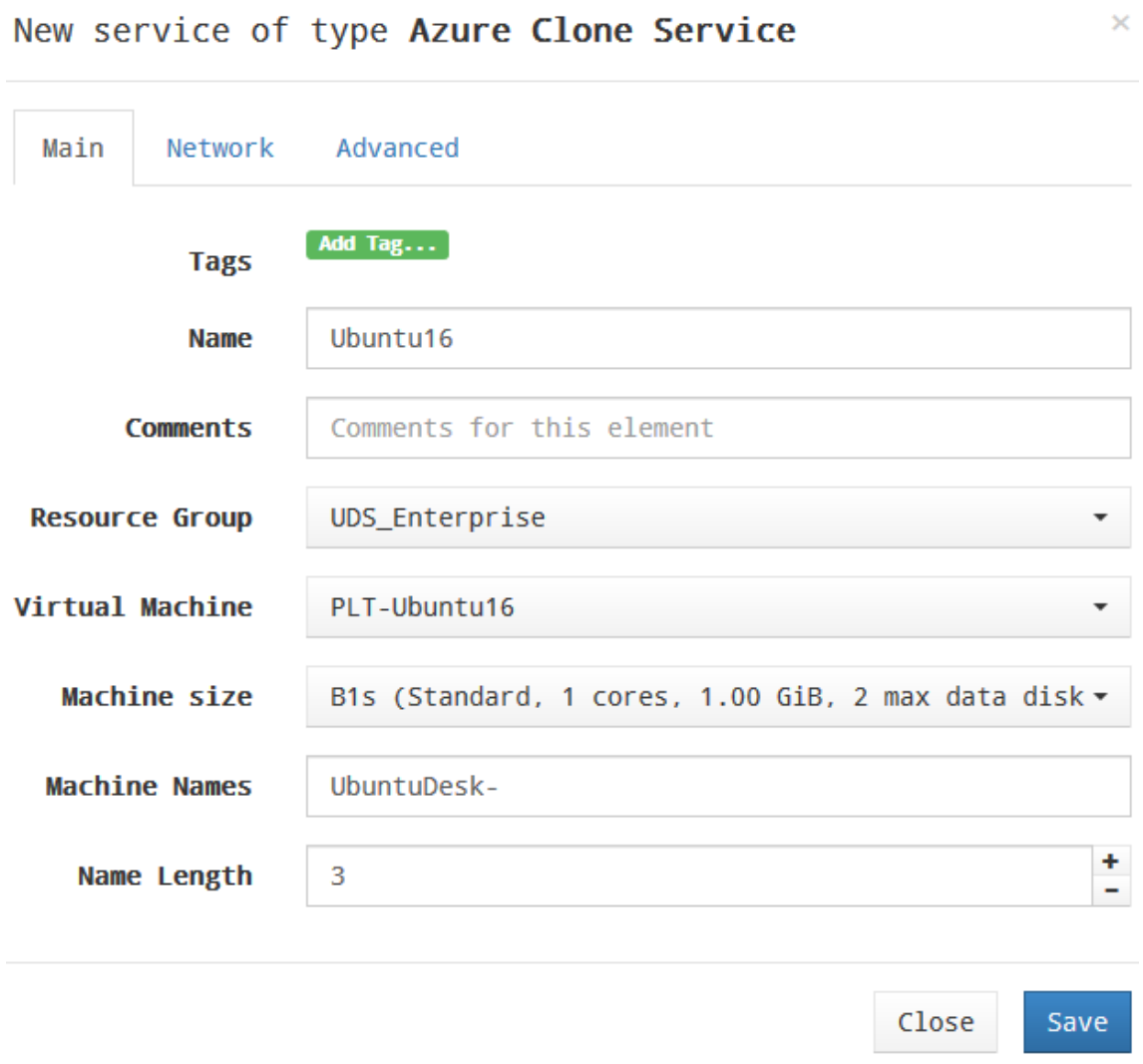
Creation of base service

When we have a valid “Service provider” connected to the Azure platform, we can create services based on templates. To do this, we select the provider, open the “Services” tab and click on “New” – “Azure Clone Service”.

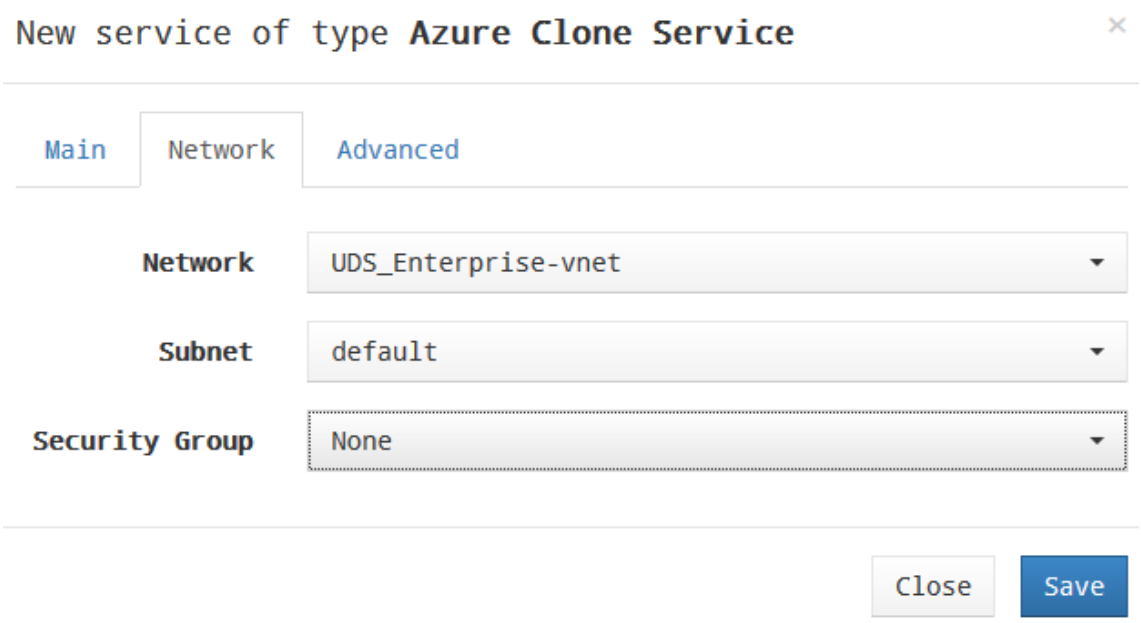


To create a base service of type “Azure Clone Service” we must indicate:

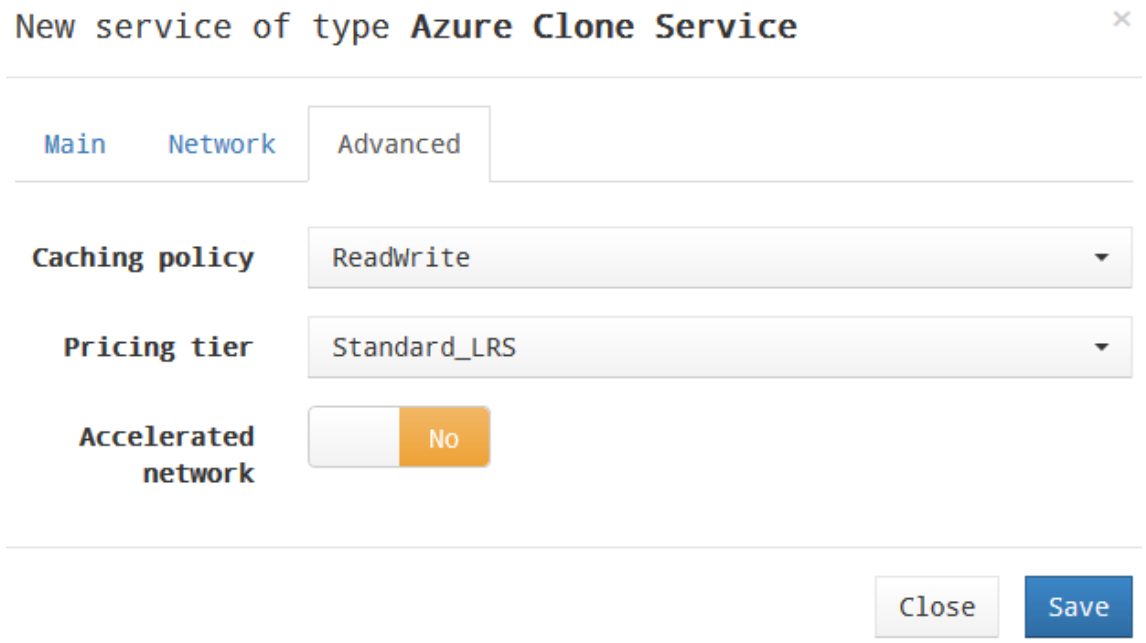
- Main:
  - **Name:** Descriptive name of the base service
  - **Resource Group:** We select the Azure “Resource Group” where we have our base machine or template
  - **Virtual Machine:** Base machine or template that we will use to deploy virtual desktops
  - **Machine Size:** Amount of resources that will have the virtual desktops automatically implement by UDS (this list will show all types of machines available in Azure, therefore, we must make sure that our Azure subscription supports the chosen type)
  - **Machine Names:** Root name of the virtual desktops generated by UDS
  - **Name Length:** Number of digits of the counter for UDS machines. These digits will be joined 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…)



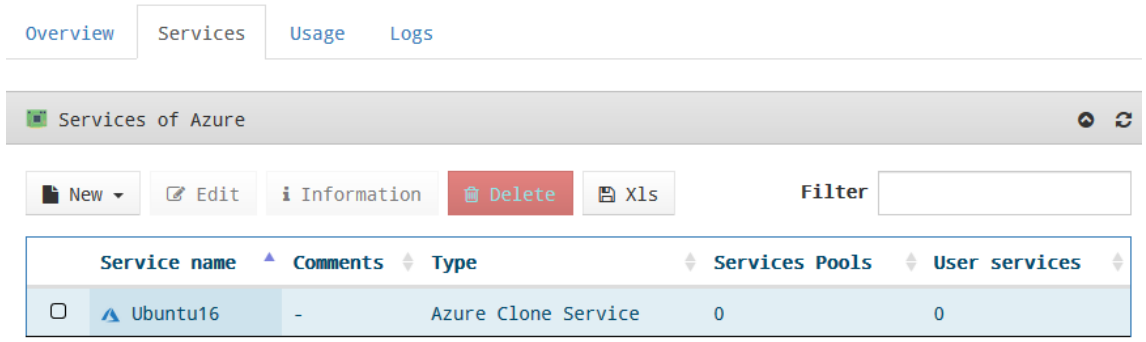
- Network:
  - **Network:** Existing virtual network in the Azure environment to which the virtual desktops will connect (there must be connectivity to the UDS Server component)
  - **Subnet:** Existing subnet of the Azure environment to which the virtual desktops will connect
  - **Security Group:** We can indicate a “Security Group” to be assigned to the virtual desktops. In this example, when the UDS components and the automatically generated desktops are in the same network, we will select “None”, since we don’t want to apply any of them



- Advanced:
  - **Caching policy:** Disk cache configuration
  - **Pricing tier:** Level of applied redundancy
  - **Accelerated network:** Enables the use of this technology (can not be used with most machines, only with: D/DSv3, E/ESv3, Fsv2 and Ms/Mms and S.O. Linux)



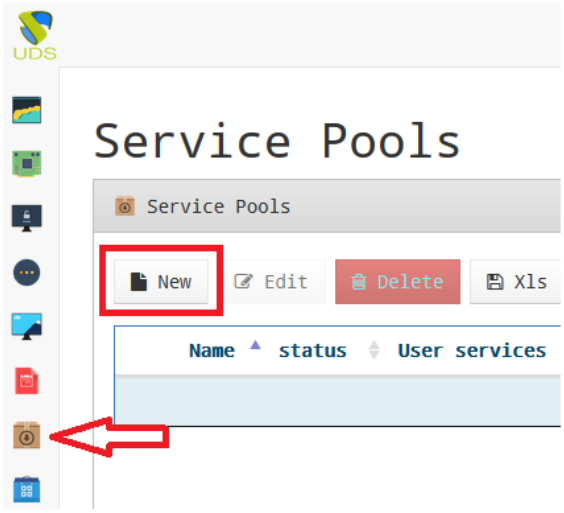
Click on “Save” and we will have a valid base service to automatically deploy virtual desktops:



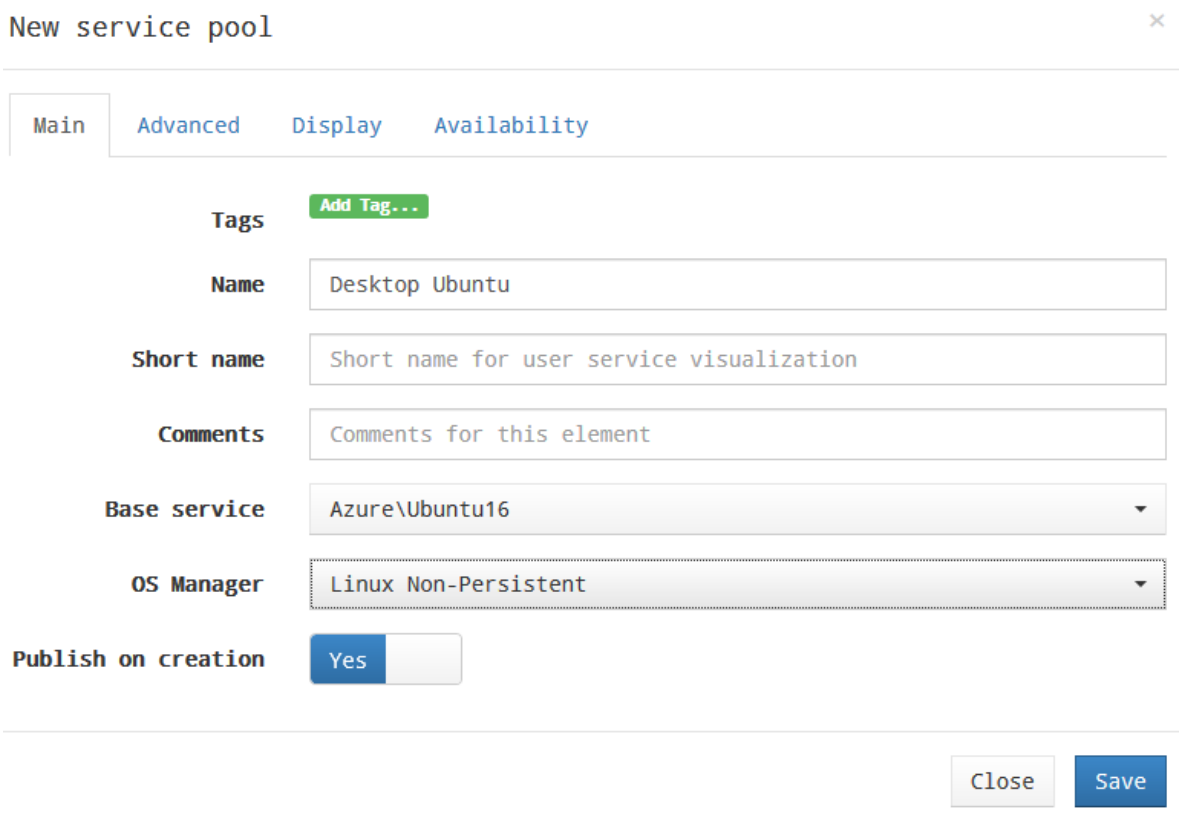
Service Pools creation

Before proceeding to create a group of services (to publish virtual desktops), it will be necessary to have at least one “Authenticator” (to validate and be able to assign services to users), an “OS Manager” (to indicate the operating system and the policy of persistence of the generated desktops) and a “Transport” (to make the connection to the desktops) 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 documentation section of our website.

When the elements discussed above are ready (Authenticator, OS Manager and Transport) we can create “Service Pools”. To do this, we open the “Service Pools” tab and click on “New”.

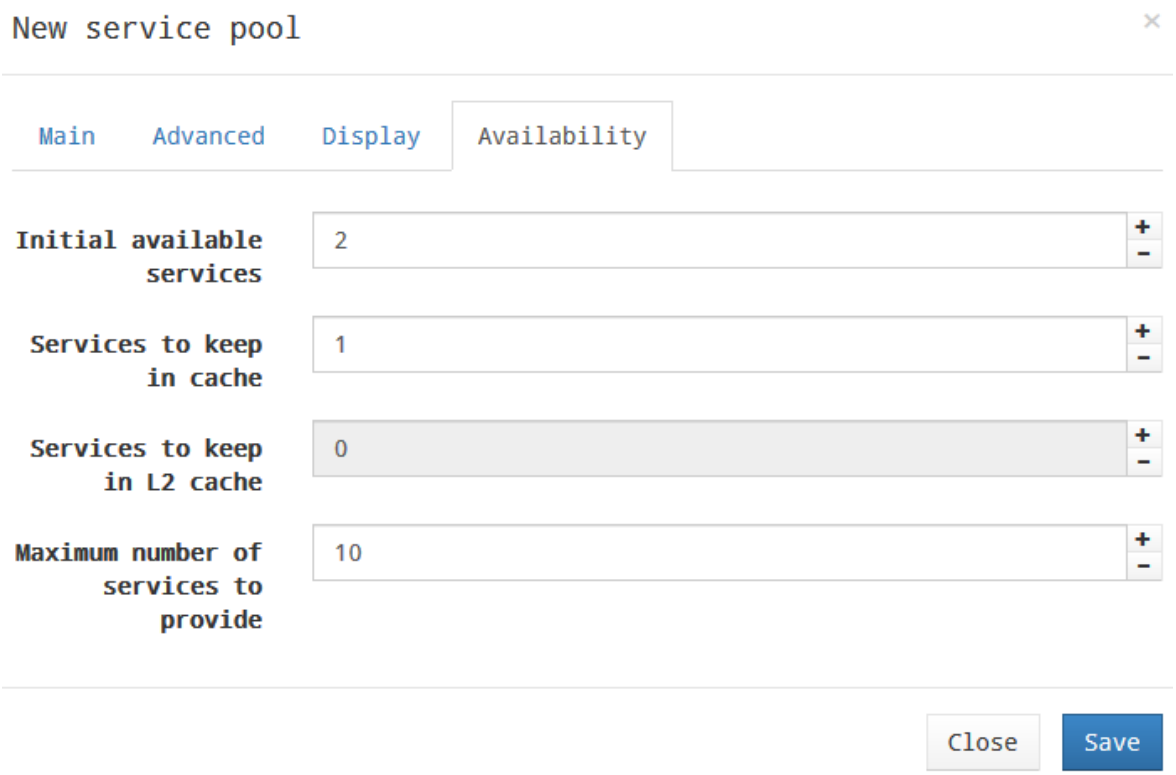


In the “Main” tab we will indicate the name of the service (this name will be visible to users) and we will select the base service previously created (in this case, the Azure platform and the Ubuntu16 base service) and an existing “OS Manager” (in this example we will use one for the Linux operating system and the non-persistent type).



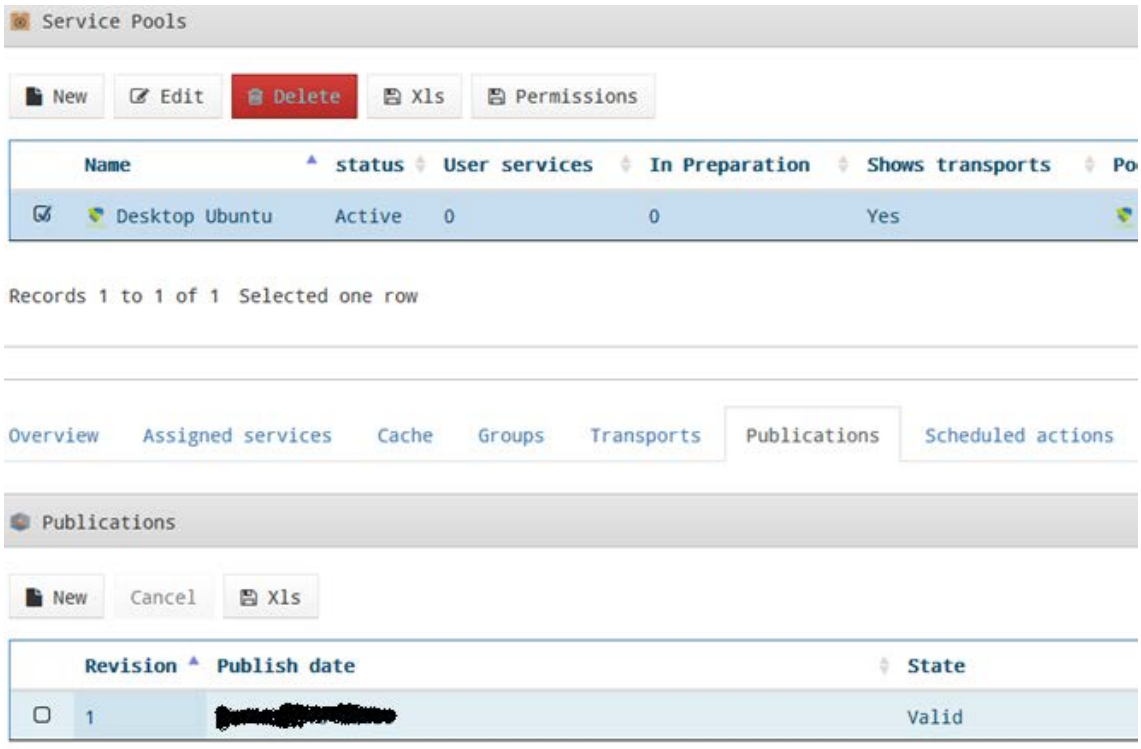
In the parameters of the “Advanced” and “Display” tabs you can left the by default values. In the “Availability” tab we will indicate the initial desktops that UDS will generate and the ones in cache level 1 (In Azure the use of the L2 cache is not available).

In this example we will indicate that UDS is going to automatically create 2 desktops and we’ll always have at least 1 available in cache.



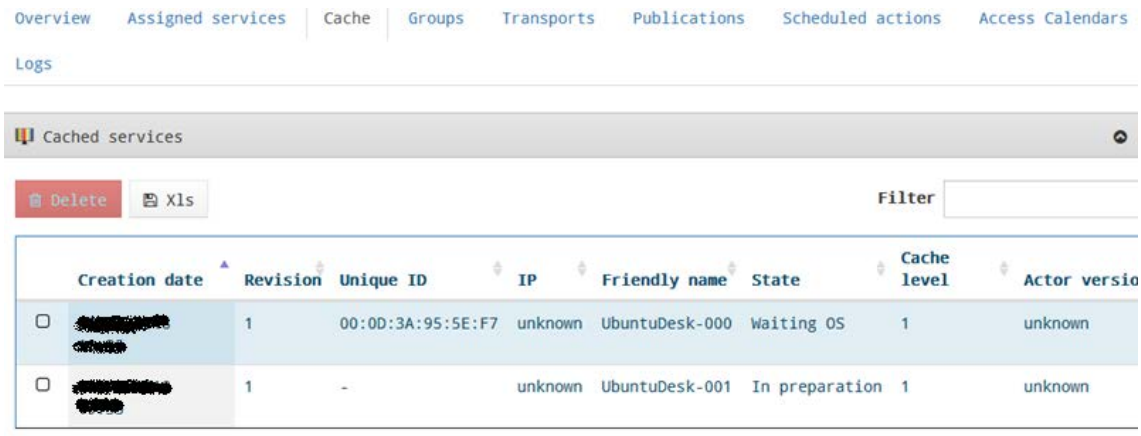
NOTE: At the time of saving the configuration or publishing a new version, the base machine or template **must be turned off**.

When selecting “Service Pool” and opening the “Publications” tab, it will be verified if the publication has been generated correctly. When in a “Valid” state, the system will automatically start generating the virtual desktops indicated in the cache parameters.

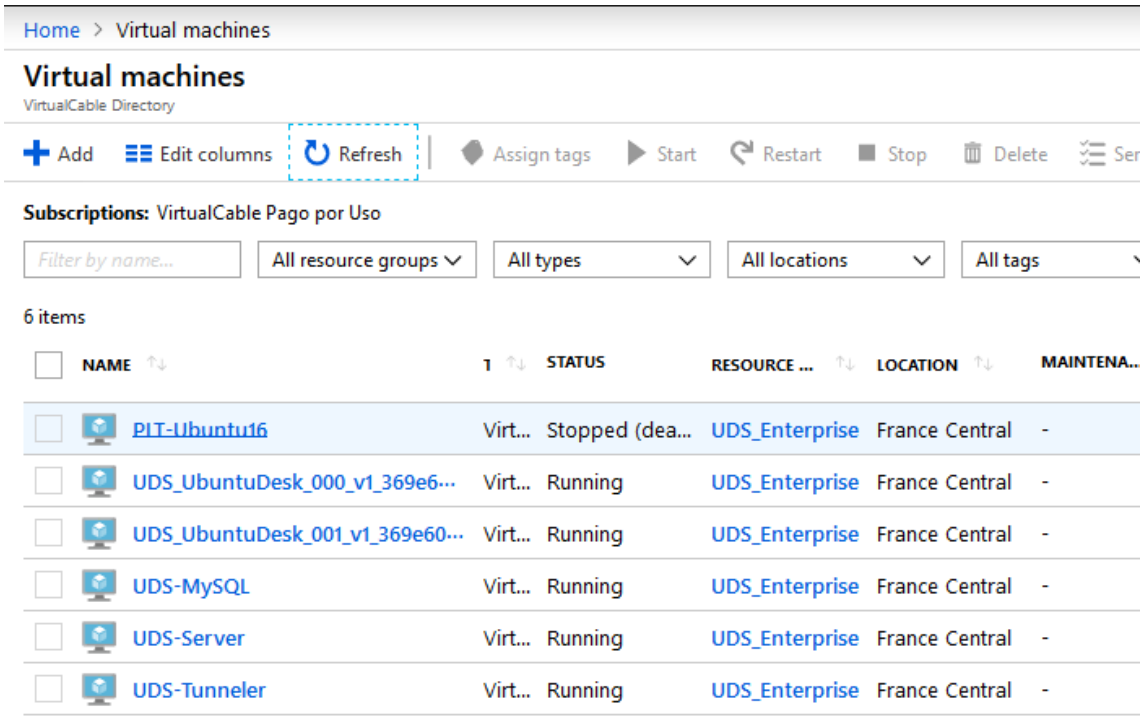


In the “Cache” tab we can see how the desktops start to be generated.

In this example, we have indicated that initially the system creates 2 virtual desktops:

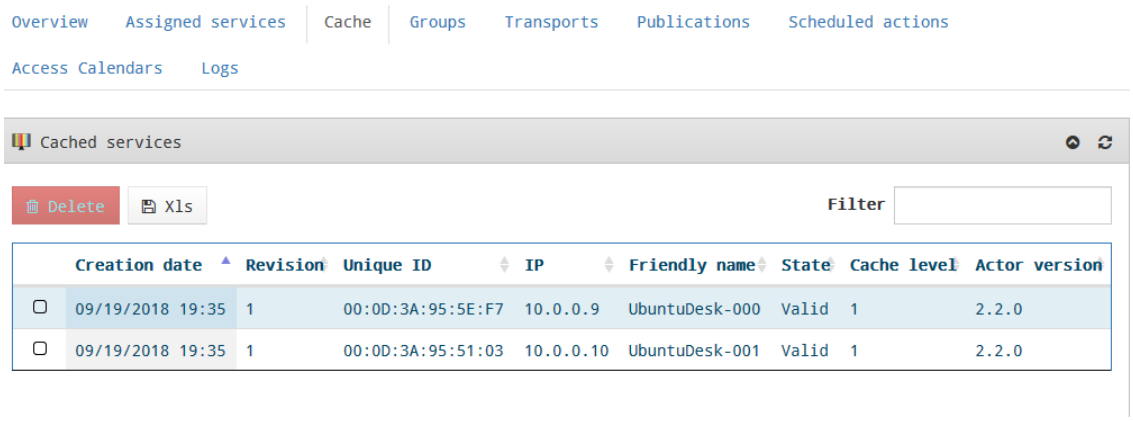


In the Azure environment we will also see how virtual desktops are being created:



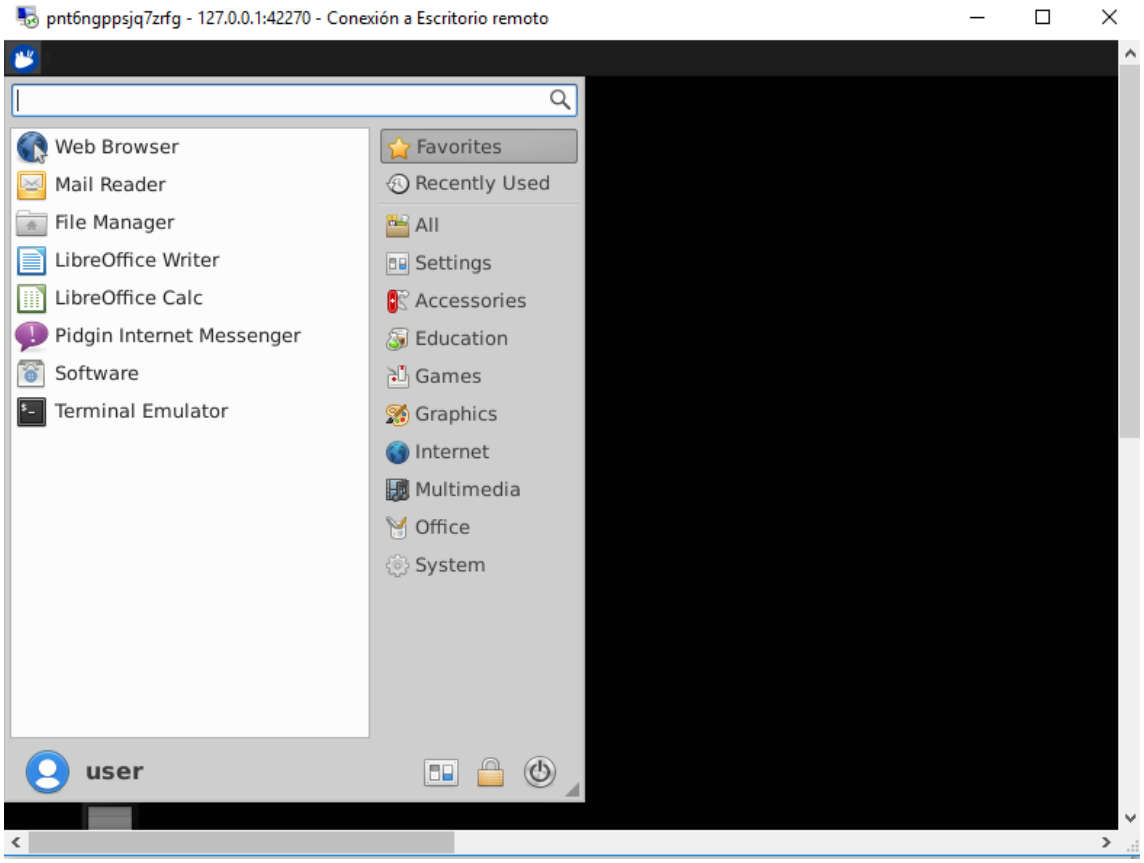
Once the desktops are in a “Valid” state (that is, the UDS Actor installed in the template has finished applying the necessary configurations), they will be available for users to access them.

NOTE: In order for users to see the service, the “Service Pool” created must have an assigned user group (“Groups” tab) and a transport (“Transports” tab).



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

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



NOTE: If we are outside the network configured in Azure, it will be necessary to use a Tunnelered transport (as you can see in the screenshot of the connection example, it is connecting to 127.0.0.1 since the connection is made through a Tunneler).

Integration of Azure AD as “Authenticator” of UDS Enterprise

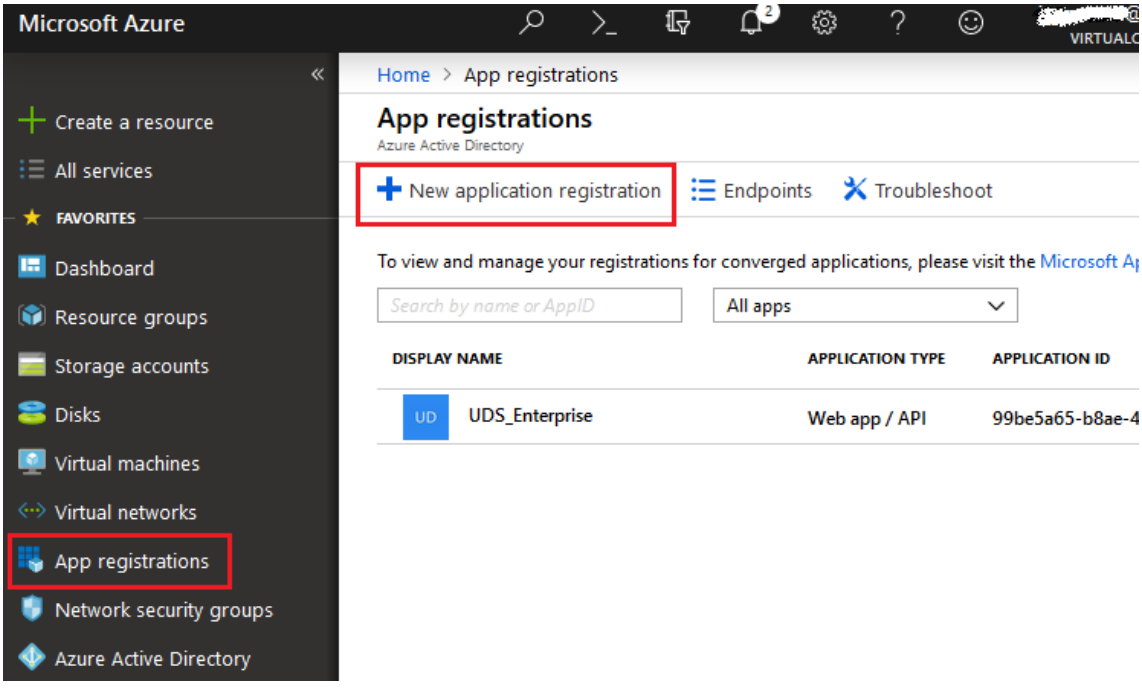
UDS allows integration with the Azure authentication system, called “Azure Active Directory”. Through this integration, it will be possible to validate the users registered in this authenticator in the UDS login portal and allow their access to desktop services and virtual applications.

To allow the correct integration between UDS and “Azure Active Directory” it will be necessary to perform some previous tasks in Azure platform.

Tasks to be done in Azure

The first task we will perform in the Azure environment will be to create valid “App registrations” to allow UDS to access “Azure Active Directory”.

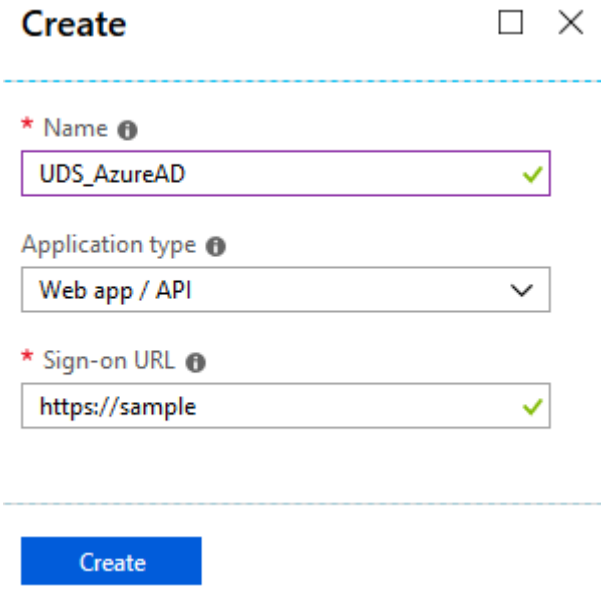
To register the application, we will go to the service “App registrations” and click on “New application registration”.



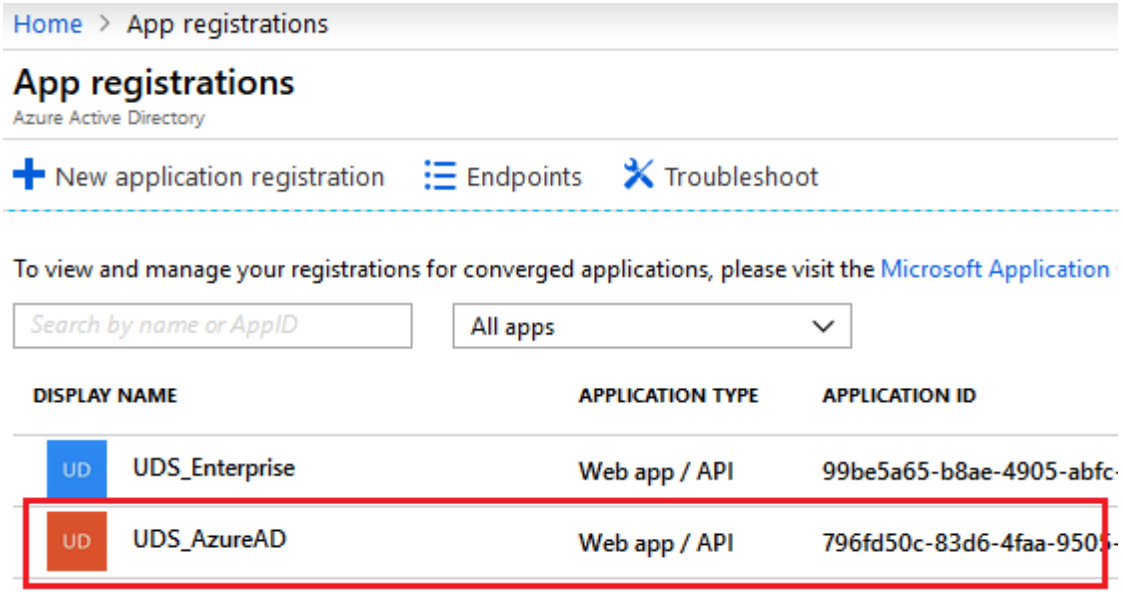
NOTE: In some cases it will be necessary to click on “View all applications” to view all existing ones.

In the creation assistant we will indicate:

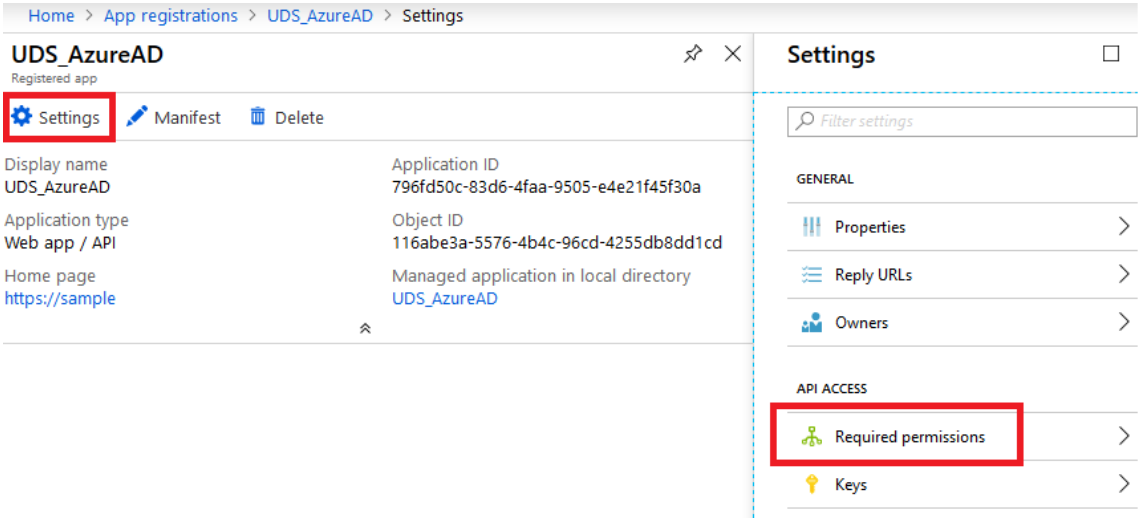
- **Name:** Name of the application
- **Application type:** We select “Web app / API”
- **Sign-on URL:** In this field we can indicate any URL (it is not necessary that it exists), it will not be used by UDS



Once all the application data is indicated, click on “create” and verify that it was created correctly (if we do not see it, click on “View all applications”):

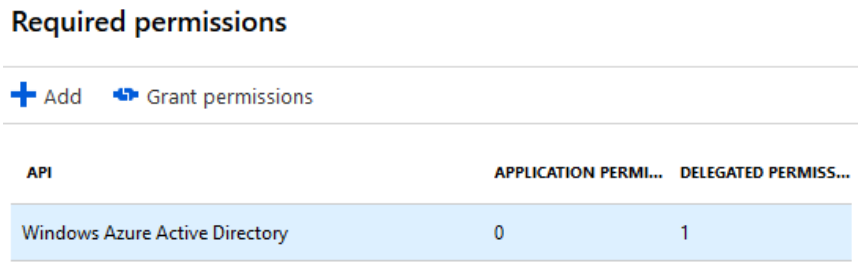


After checking that it has been correctly created, we will have to select the App, click on “Setting” and select in the menu “API ACCESS” the option “Required permissions”.

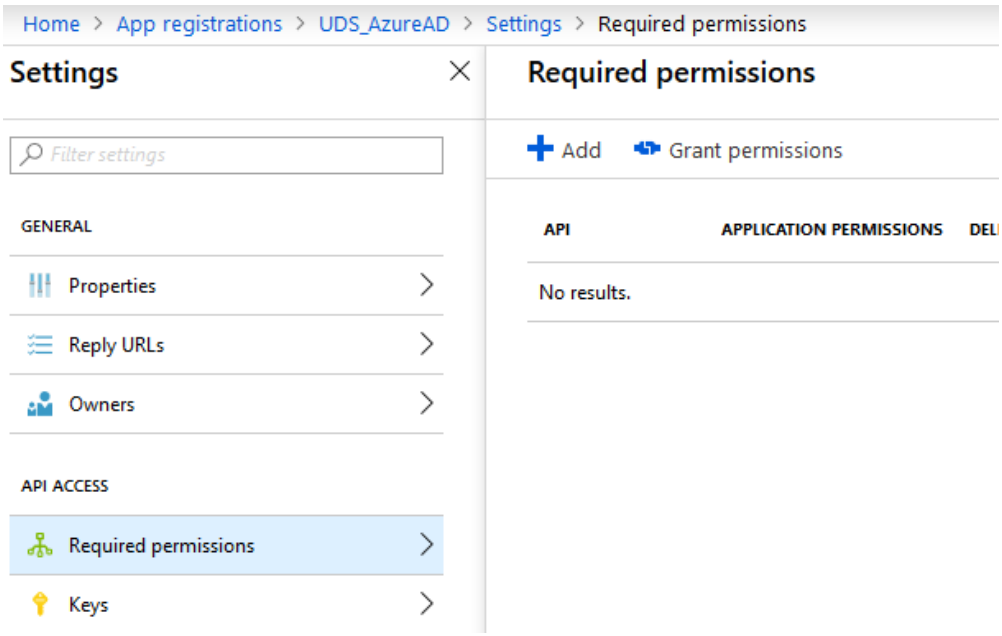


Now we will assign the necessary permissions so that UDS can read users and groups of the “Azure Active Directory” authenticator.

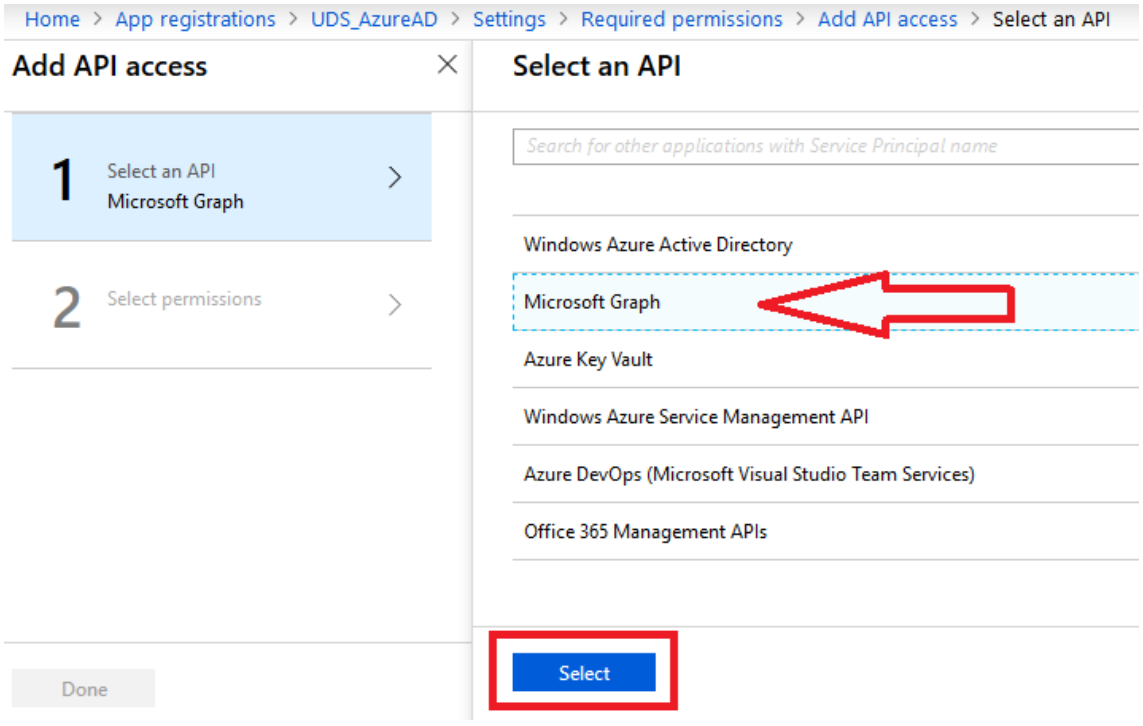
By default, one called “Windows Azure Active Directory” is added in the creation of the App. We can select it and delete it.



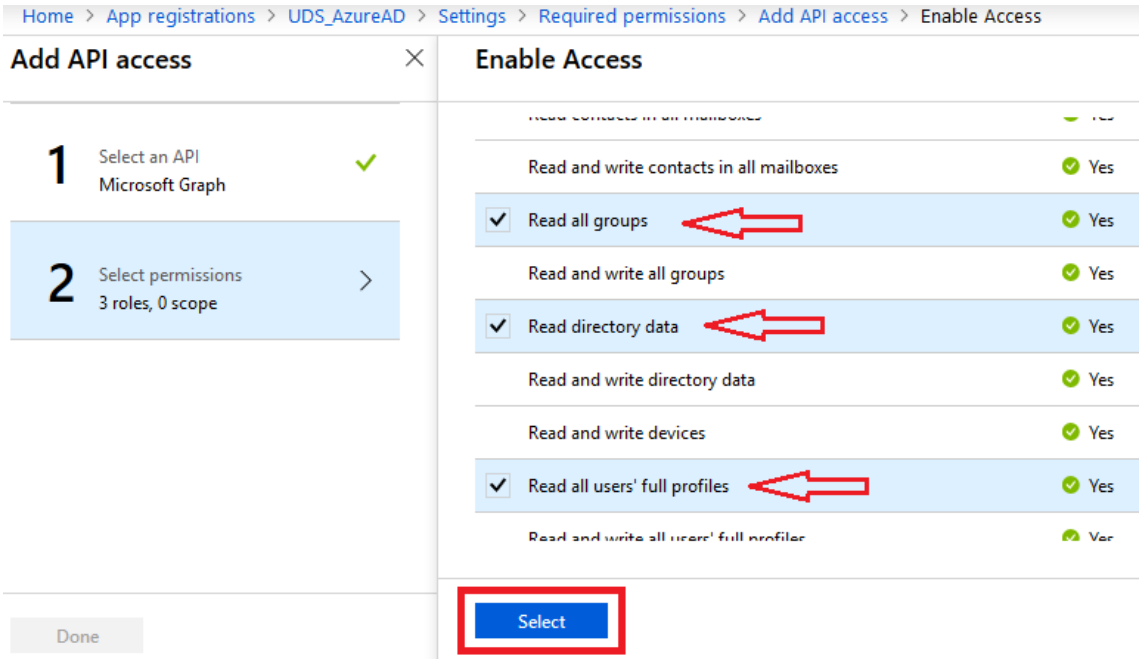
Once deleted, we will click on “add” to add the permissions that UDS needs for its correct integration.



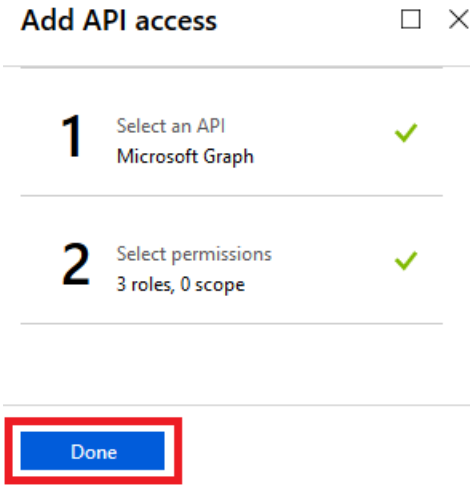
In step 1 of the wizard, we will select “Microsoft Graph” and click on “Select”.



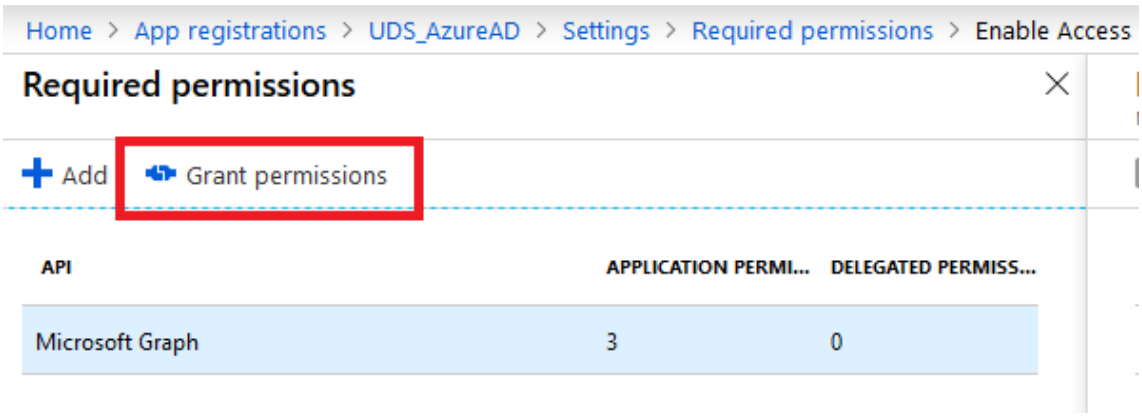
In step 2 we will indicate the necessary permissions that will be within the “APPLICATION PERMISSIONS” section. They will be: “Read all groups”, “Read directory data” and “Read all users’ full profiles”. Click on “select” once we have selected them.



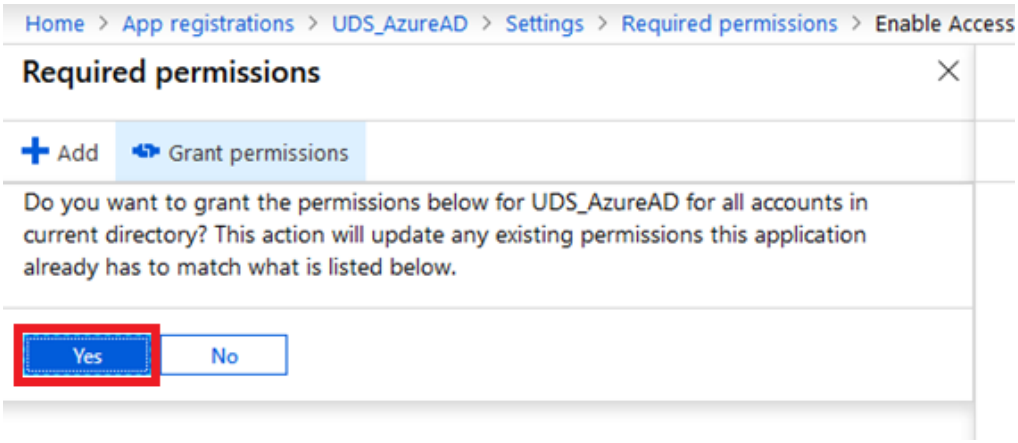
Click on “Done” to finish the assistant.



Finally, we will select “Microsoft Graph” and click on “Grant permissions” to apply them correctly.



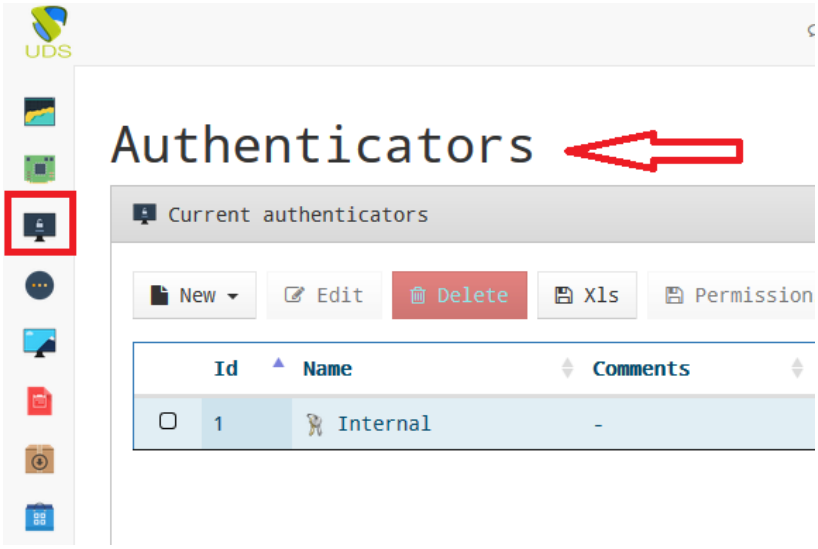
We will click on “yes” to apply.



The next task of the integration process of “Azure Active Directory” with UDS will be done from the UDS administration itself.

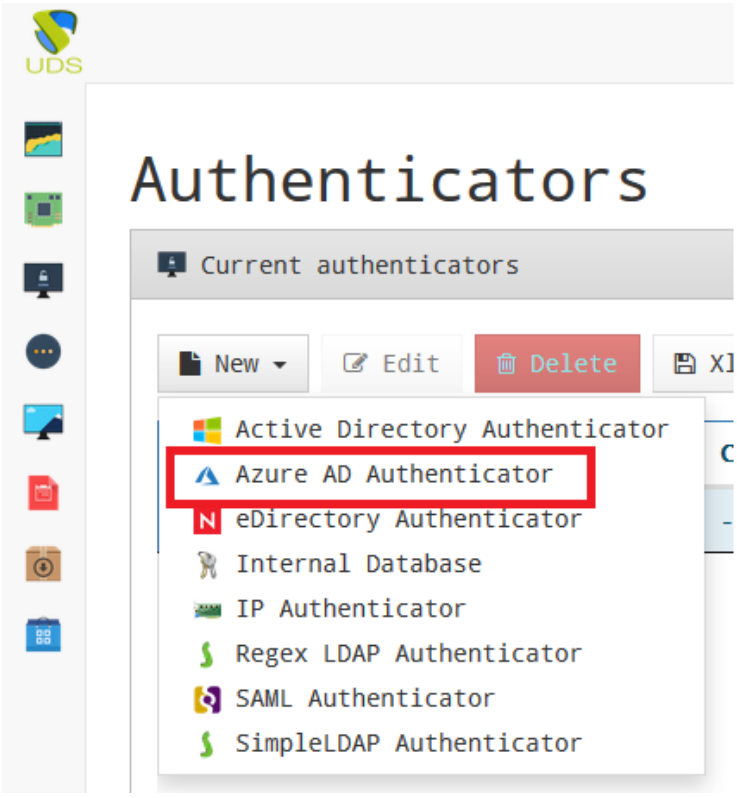
Tasks to be done in UDS Enterprise

From the UDS Enterprise administration dashboard, we will proceed to register the new authenticator “Azure Active Directory”. In order to do this, we will validate in the UDS login portal with a user with administration permissions and we will access the “Authenticators” section.



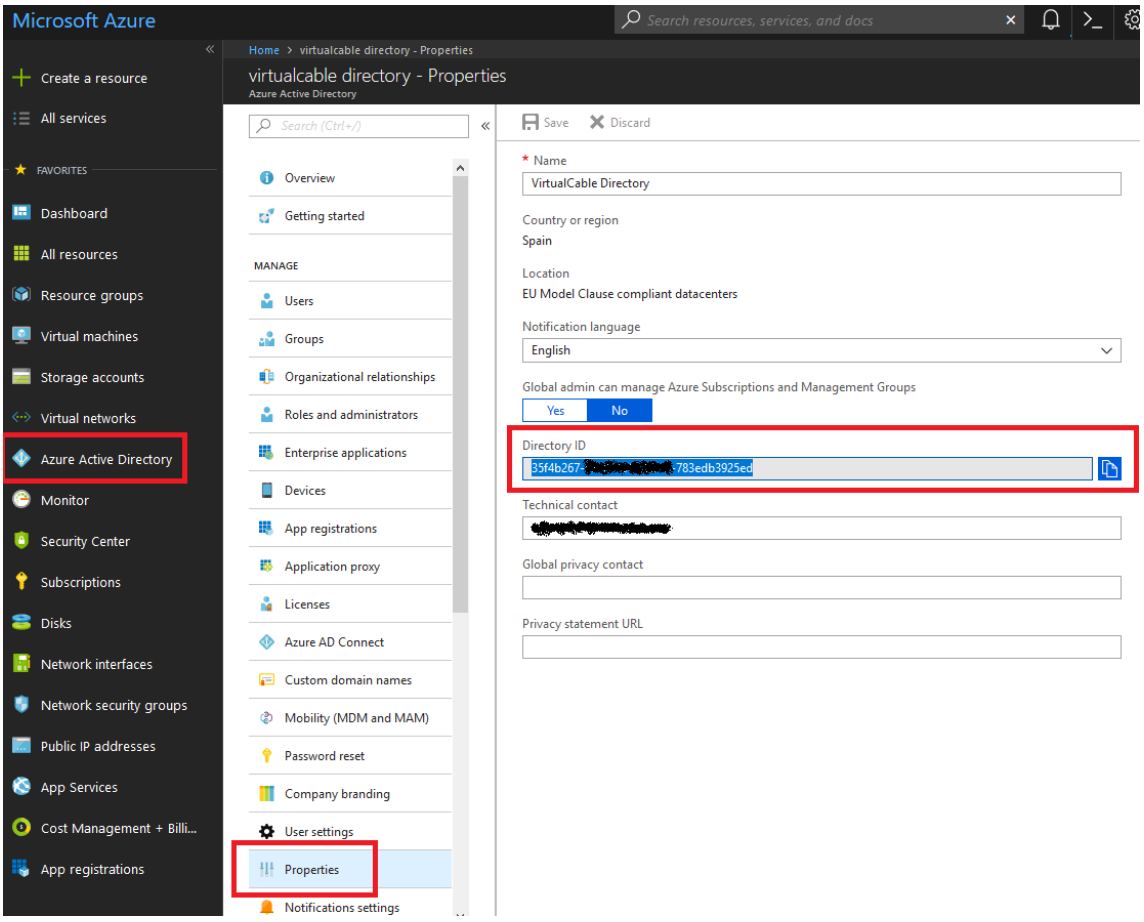
NOTE: In UDS we can have different types of authenticators registered in the system, the priority field will define which will be the authenticator that will be shown to the users by default.

Click on “New” and select “Azure AD Authenticator”.

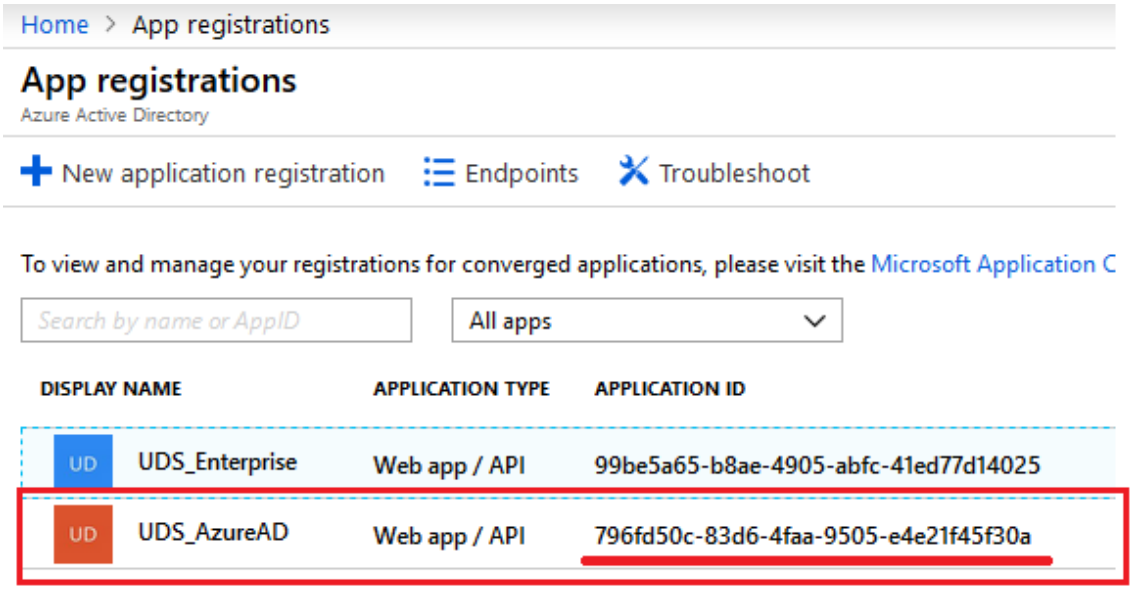


Within the wizard we must indicate a series of necessary data:

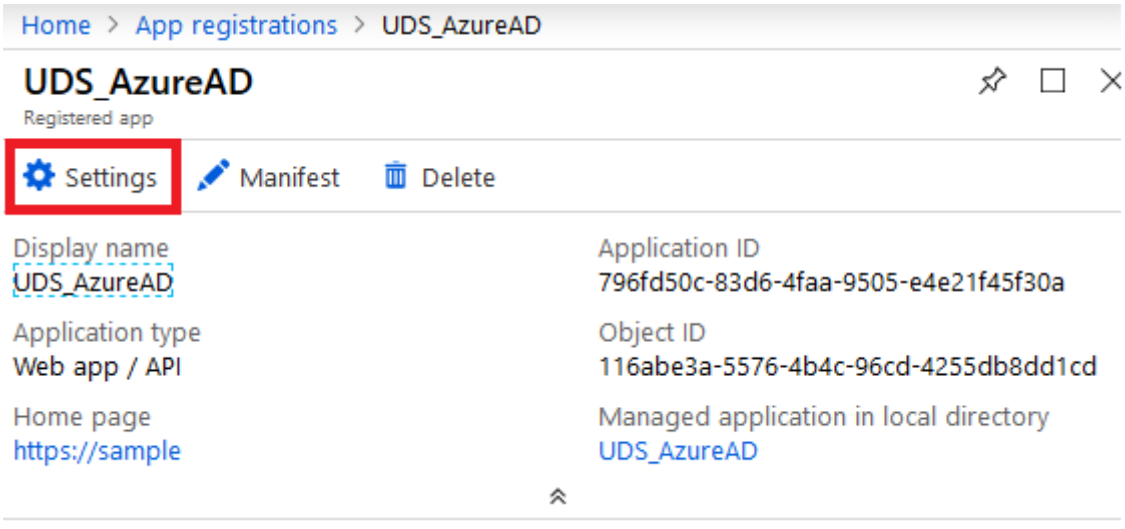
- **Main:**
  - **Name:** Authenticator name
  - **Priority:** Priority of this authenticator in the list of available authenticators. The lower the priority, the greater will be in the list of available authenticators (of all the authenticators, the one with the lowest priority, including the negative values, will be the default authenticator)
  - **Label:** Label assigned to this authenticator. You must place it in the login URL to perform a direct validation without having to use the list of authenticators
  - **Tenant ID:** This value can be obtained from the service “Azure Active Directory”, “Properties” , “Directory ID”



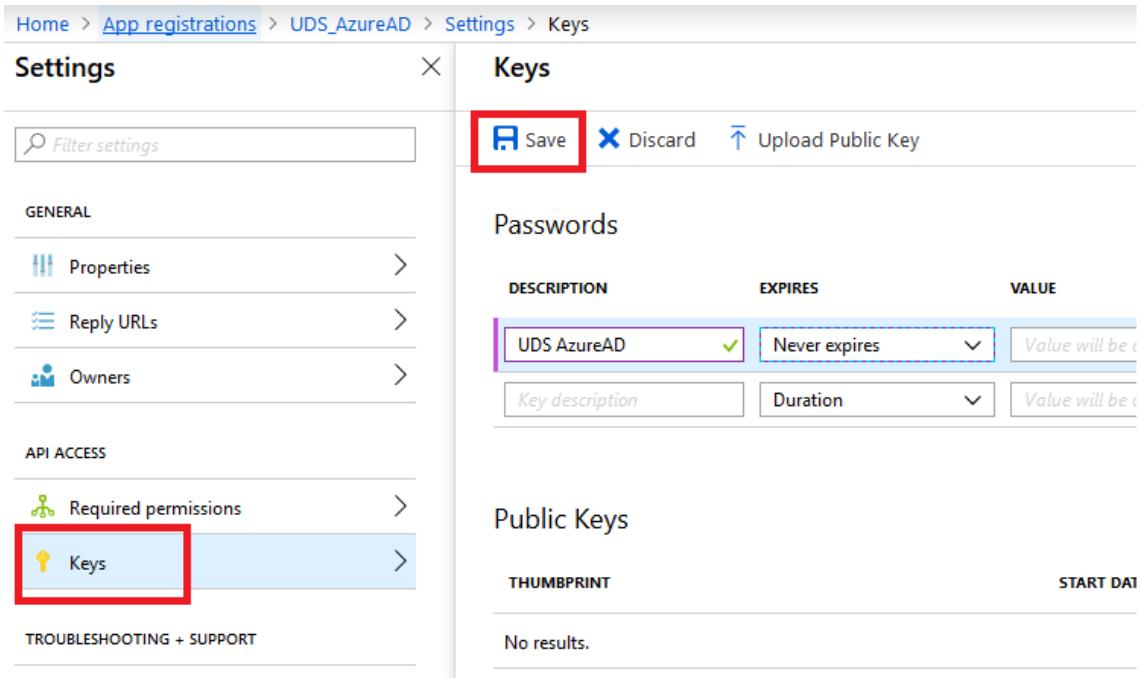
- **Client ID:** To obtain this value, it will be necessary to access the “Application registration” created above and copy the value of “Application ID”.



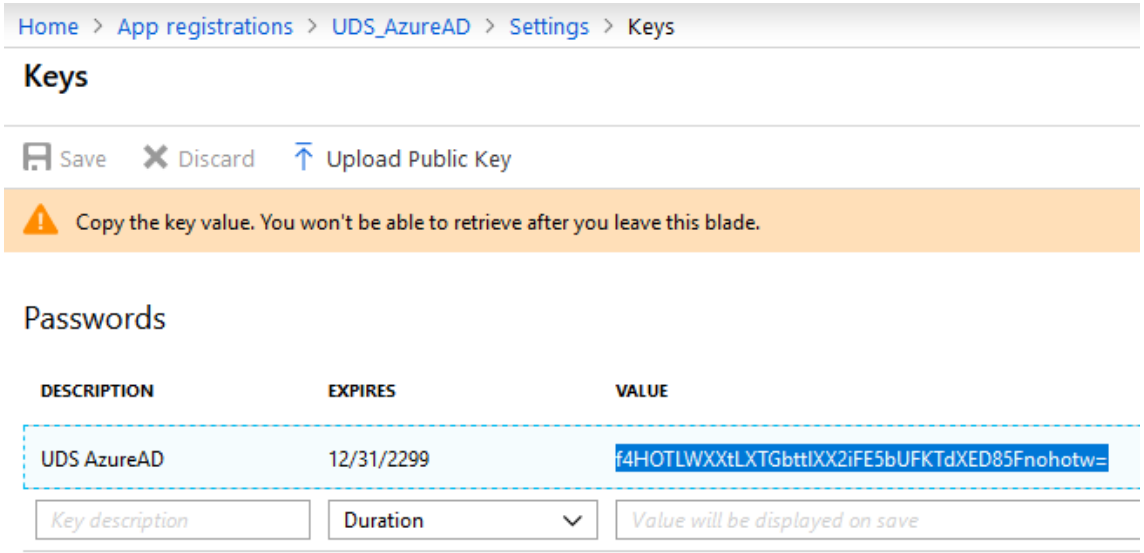
- **Client Secret:** We will obtain this value from the previously registered application. Click on it (in the “App registrations” service) and access “settings”.



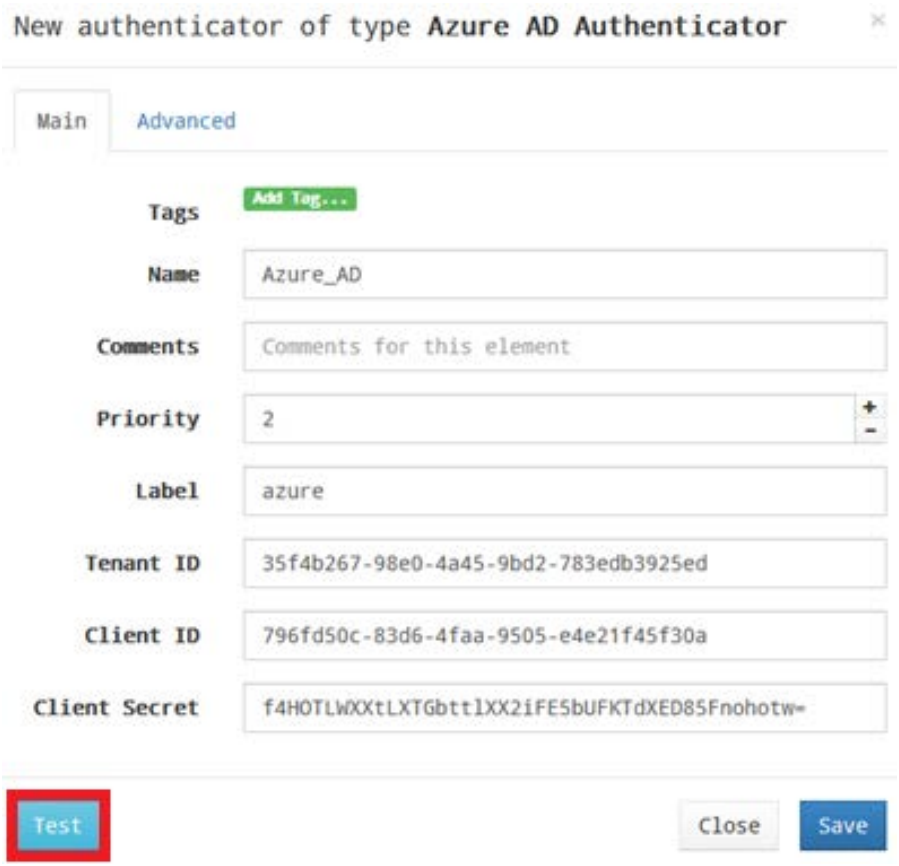
Within “Settings” click on “Keys”. In the section “Passwords” we will indicate a description, the date it expires and click on “save” to copy the “key”:



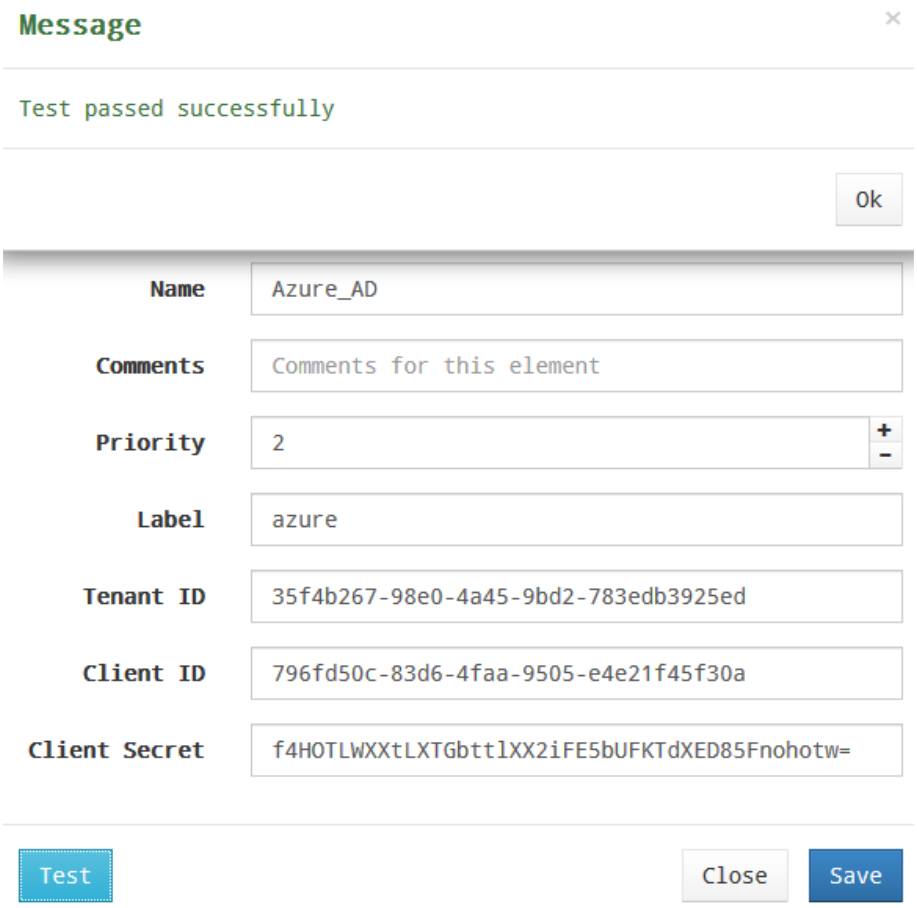
Once saved, it will allow us to copy the value (once this window is closed, we can’t copy this value again, although we can generate a new one if necessary) and we can use it as a Client Secret in UDS Enterprise.



Once all the fields are filled in, we will click on “Test” to verify the correct integration.



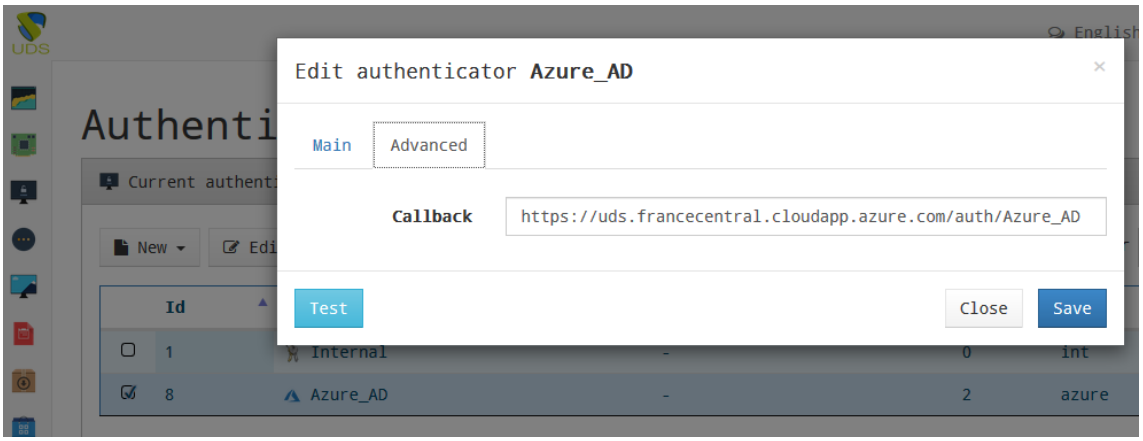
Once verified the correct connection, press on “save” to save it.



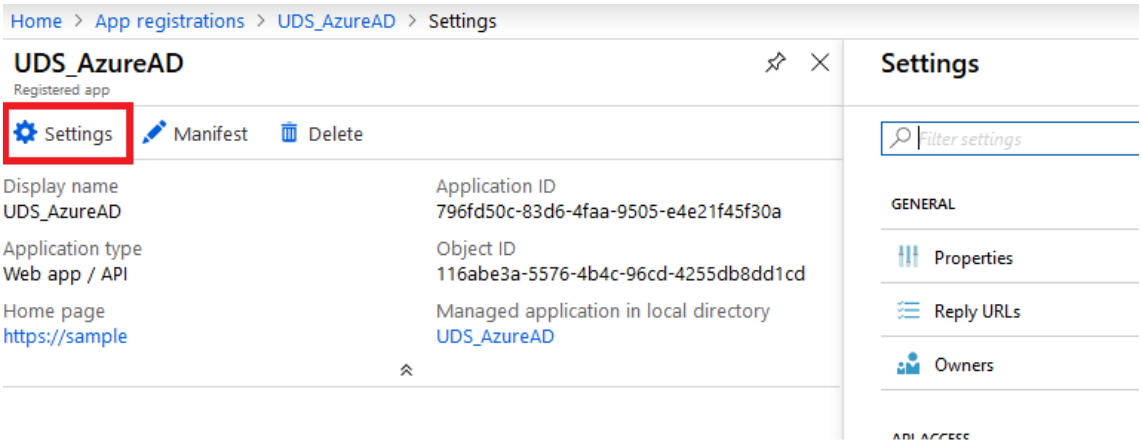
NOTE: If the test indicates that an error has occurred, you can save the connector by clicking “Save” to avoid losing data such as “Client Secret” and then review the causes of the connection error.

The last task to be performed in order to complete the integration of UDS with “Azure Active Directory” authenticator will be to indicate the access URL allowed in the Azure environment.

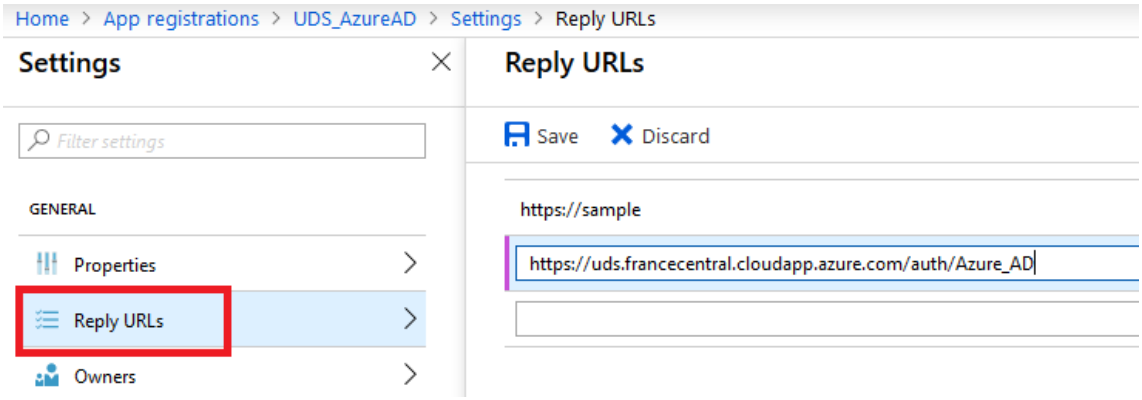
In the “Authenticators” section of the UDS administration dashboard, we select the authenticator previously created to edit it, accessing the ”Advanced” tab. We will need to copy the value of the “Callback” field.



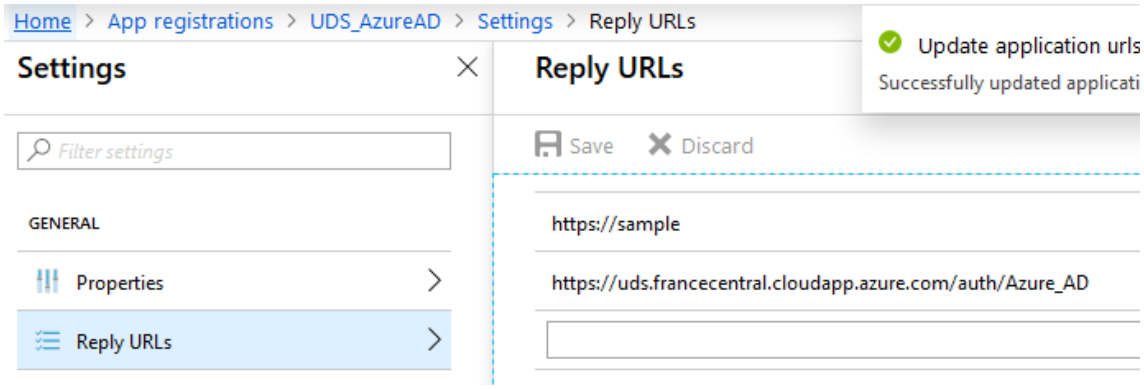
Once we the value has been copied, we will access the Azure platform. In “App Registrations”, we select the application previously created for the integration of Azure AD with UDS and click on “Settings”.



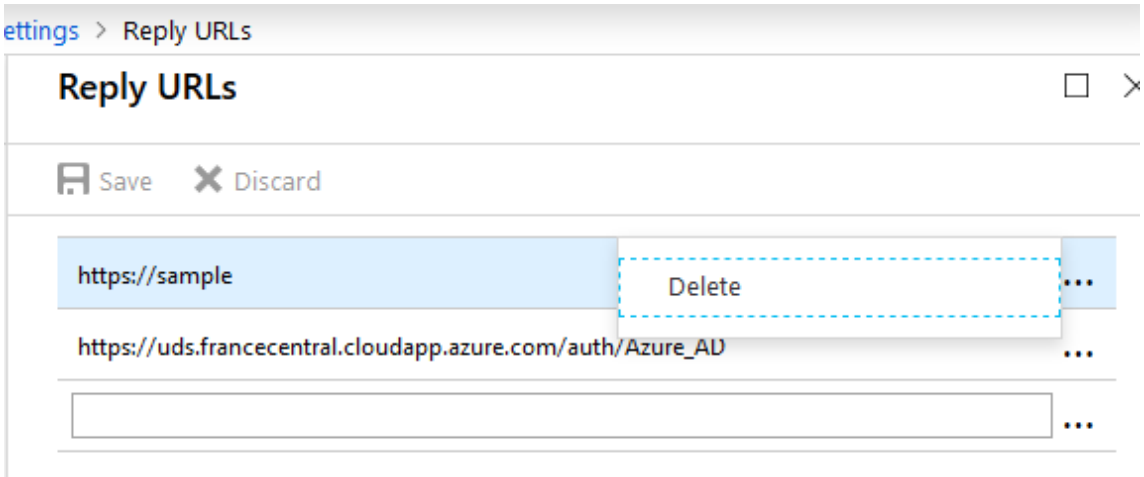
Within “Settings” and under the “General” menu, we select “Reply URLs”. There we will have to paste the value copied from the UDS administration in the “Callback” field.



Click on “Save” to save the new “Reply URL”.



NOTE: The URL indicated in the creation of the application can be deleted. We select it, click on the access points to the action menu, select “Delete” and then “Save”.



Once these steps are completed, users can now authenticate with the user credentials configured in an “Azure Active directory” authenticator.

About VirtualCable

VirtualCable markets UDS Enterprise through a subscription model, including support and updates, depending on the number of users.

In addition, VirtualCable offers professional services to install and configure UDS Enterprise and other virtualization technologies.

For more information, visit www.udsenterprise.com or send us an email to info@udsenterprise.