Instruction manual

How to Install Openstack in Ubuntu 18.04 using Devstack

Preamble

On the following pages, you will find instructions on how to set-up a physical or virtual system with Devstack to create a working cloud environment. Devstack is a bundle of scripts that will install an compleet Openstack environment Devstack will install the latest version of openstack on your Ubuntu server, you should use devstack only in a test environment and not in a production environment. OpenStack is open source software that makes it possible to create a cloud Infrastructure, that offers an Infrastructure as a Service (IaaS) platform for both private and public clouds.

Important to note is that we assumed that you can make a system or virtual machine ready to install this software on. The prerequisites are documented in detail in the first chapter. afterwards in the second chapter you can find the steps on how to install Devstack. The chapters 3 till 7 will be about the configuration of the Devstack. In the Appendix the installation of Ubuntu can be found if you need this.

We strived towards a complete and easy-to-follow document for people who want to try out the functionalities of Devstack. We came to this idea because the already existing documentation and tutorials on the Internet were not complete, up to date, accurate or easy to follow.

Chapter 1 - Pre installation steps Devstack

Before you begin to read this manual, you must meet hardware requirements for Devstack which you will find below:

- Processor - 2 cores*
- Memory - 8GB
- Hard Drive - 60GB

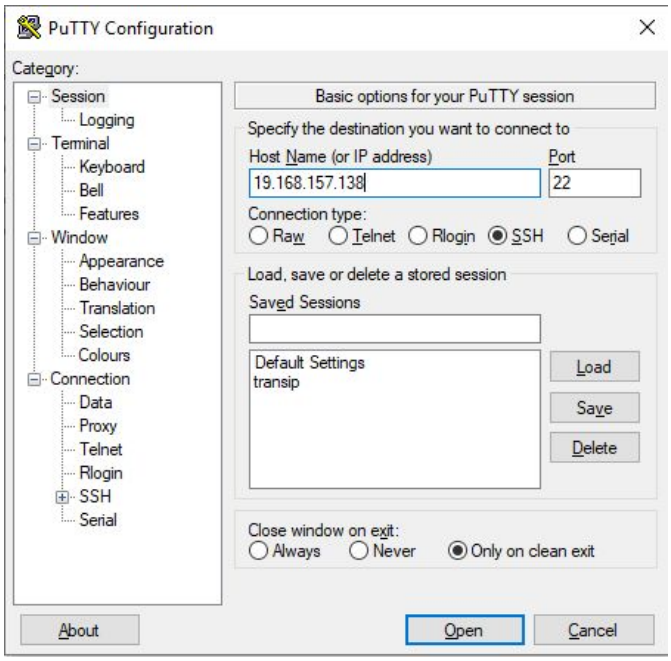
*Keep in mind that different processors have different per-core performance. It is advised to use hardware that is suitable for running complex tasks e.g. an Intel Xeon or Core i7 series CPU or equivalent.

*We assume that you know how to install Ubuntu server if this is not the case, it is not a problem in the appendix of this document you can find a step-by-step plan that will help you to install Ubuntu server 18.04.

*We assume that you can access the server with SSH or manage the server in another way and that you have an admin user and password.

*We assume that you have basic knowledge of Linux and SSH keys.

Step 1: Set up an SSH connection to the server. You can set up the SSH connection with a tool like putty or via the terminal when you use a Mac or Linux, you can also use Powershell to set up the connection.
When you use putty, enter the IP address and click on the "Open" button, after which you can log in with the username and password you provided during the installation.
Note: The IP addresses are only meant as examples in the images below.



In the case of a terminal or PowerShell, see the syntax to set up an ssh connection to the server in the image below. Login to the server with the specified username and password.

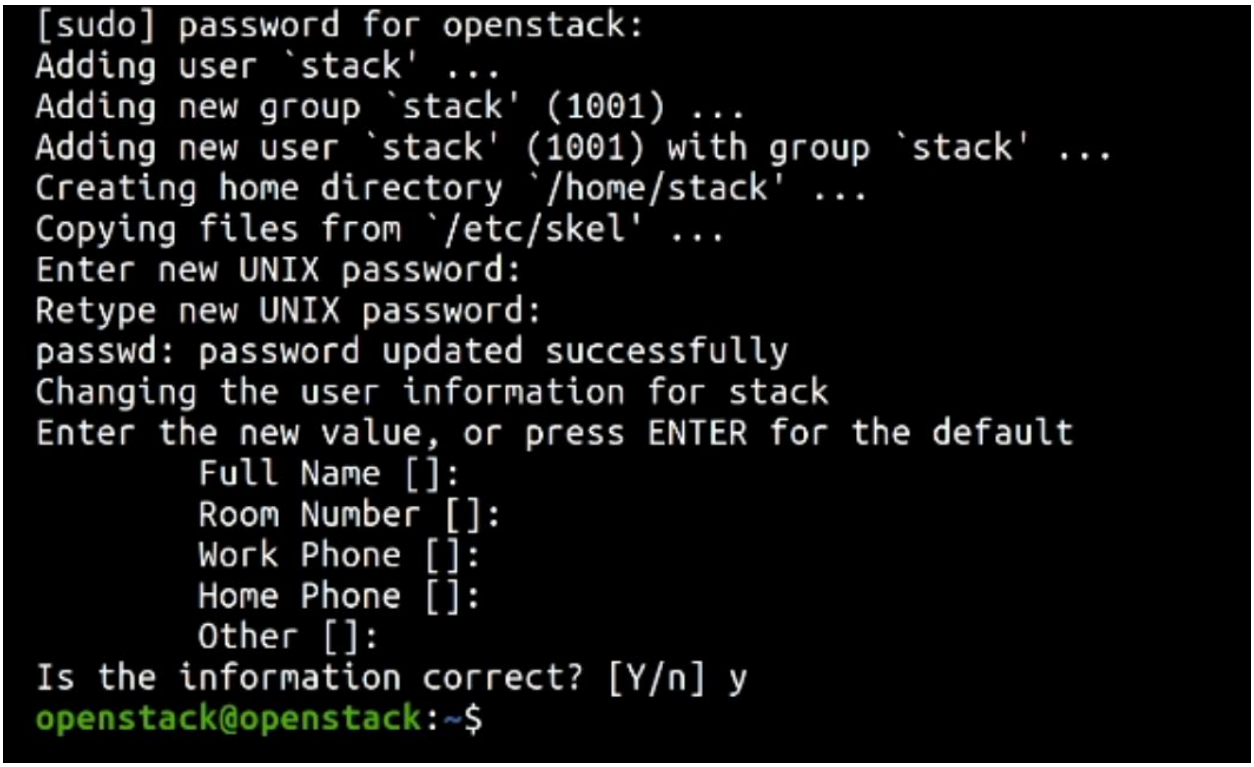
Step 2: The next step is to create a username and password for the devstack environment to run on. We recommend to make a user-specified for the use of devstack; so it only has the permissions it needs to run. Running it on the admin account is also possible, but keep in mind that running devstack on the admin account may have security risks.

To add a user in ubuntu: please type in the following lines in your terminal:

“sudo adduser stack”

Note: You will be automatically prompted to enter a password for user ‘’stack’’.

Note: Devstack should be run as a non-root user with sudo enabled.



Step 3: After you have created the user “stack” you have to enable sudo privileges for this user without need for a password. This is required for the Devstack installation, the command **“sudo -i”** will give you a root privilege terminal. To give the user “stack” privileges without asking for a password for the installation script of Devstack you have to execute the following command **echo "stack ALL=(ALL) NOPASSWD: ALL" >> /etc/sudoers.**

Step 4: Now you can log in as the user “stack” we use the commando **“su stack”** su stands for switch user. After you are logged in as the user “stack” you navigate to the home directory from stack, therefore you have to execute the command “cd ~”. To confirm you are in the home directory of the user “stack” we use the command “pwd”. The command will print the current directory pad. As you can see in the image below we are in the home directory of the user “stack”.

Step 5: To download devstack from Github, you will need the github cloner. With github cloner you will be able to download the installation files for devstack on your system.

To install github cloner; please type the following:
“**sudo apt install git”**

Ubuntu will automatically download and extract the installation files from github in the corresponding folders.

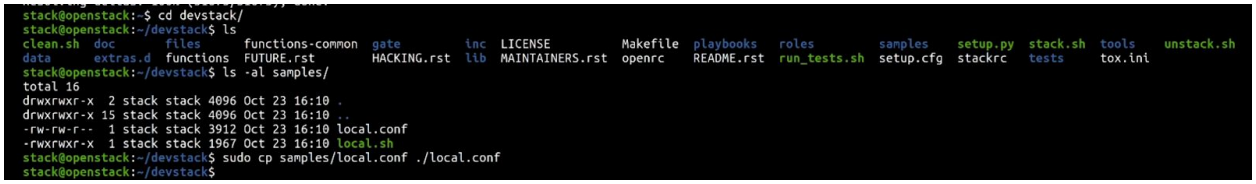
Note: You may be asked for your password before the installation will start, however if you are logged in as a sudo (administrator) this would not be a problem

Step 6: Now you need to download the devstack installation files. These can be obtained by typing the following in your terminal: git clone https://git.openstack.org/openstack-dev/devstack

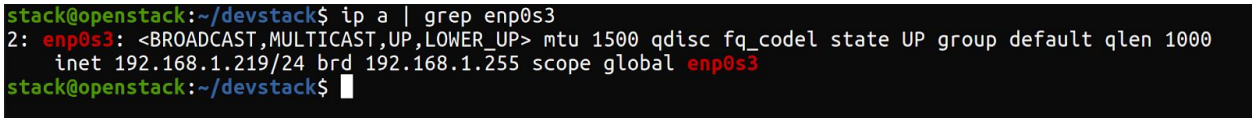
You will see some downloading files and if the command is executed correctly, you will see ‘’done’’ on your screen.
Note: The duration of the download may vary depending on your internet connection speed

Step 7: After you have downloaded the devstack installation files, you will need a configuration file. The easiest way to obtain a configuration file is to copy the sample configuration file in the /samples/ folder to the main directory. You can do this with the following command **"sudo cp samples/local.conf ./local.conf"** The ./local.conf part will copy the local.conf in the current directory, in this case devstack. The sample file will have you covered for most of the time.

Note: You can also make your own configuration file, you can Google on how to set parameters for your custom configuration files.



Step 8: Now you need the IP address of the server, we use the command "**ip a"** to find out the IP address. We use the grep command because we have multiple network cards in the server. The grep command filtered on the interface name that we specified (enp0s3) in our case.



Step 9: Next you have to edit the local.conf file, we use the nano text editor to open the local.conf file, but you can also use vi or vim for that. To open the local.conf file in the Devstack folder. We use the command **“sudo nano local.conf’.**

Note: If you use vi or vim replace nano in the command for vi or vim these are both text text editors.

In the local.conf file we edit the following ADMIN_PASSWORD, DATABASE_PASSWORD, RABBIT_PASSWORD, SERVICE_PASSWORD and HOST_IP

Note: Because this is a demo setup, we use a simple password. Enter the server's ip address at HOST_IP.

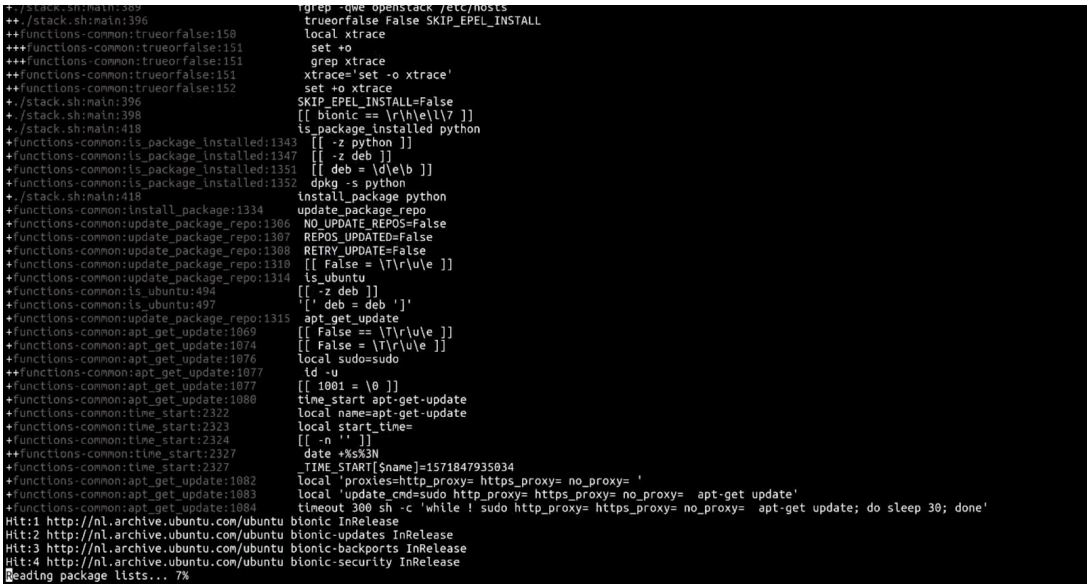
To save the local.conf file if you are using nano, press the key combination ctrl + x after which you will be asked if you want to save the file, to press it, the key combination of vi or vim is easy to look up via the Internet.

```
# Sample ``local.conf`` for user-configurable variables in ``stack.sh``

# NOTE: Copy this file to the root DevStack directory for it to work properly.

# ``local.conf`` is a user-maintained settings file that is sourced from ``stackrc``.
# This gives it the ability to override any variables set in ``stackrc``.
# Also, most of the settings in ``stack.sh`` are written to only be set if no
# value has already been set; this lets ``local.conf`` effectively override the
# default values.

# This is a collection of some of the settings we have found to be useful
# in our DevStack development environments. Additional settings are described
# in https://docs.openstack.org/devstack/latest/configuration.html#local-conf
# These should be considered as samples and are unsupported DevStack code.

# The ``localrc`` section replaces the old ``localrc`` configuration file.
# Note that if ``localrc`` is present it will be used in favor of this section.
[[local|localrc]]

# Minimal Contents
# ----------------

# While ``stack.sh`` is happy to run without ``localrc``, devlife is better when
# there are a few minimal variables set:

# If the ``*_PASSWORD`` variables are not set here you will be prompted to enter
# values for them by ``stack.sh``and they will be added to ``local.conf``.
ADMIN_PASSWORD=openstack
DATABASE_PASSWORD=openstack
RABBIT_PASSWORD=openstack
SERVICE_PASSWORD=openstack

# ``HOST_IP`` and ``HOST_IPV6`` should be set manually for best results if
# the NIC configuration of the host is unusual, i.e. ``eth1`` has the default
# route but ``eth0`` is the public interface.  They are auto-detected in
# ``stack.sh`` but often is indeterminate on later runs due to the IP moving
# from an Ethernet interface to a bridge on the host. Setting it here also
# makes it available for ``openrc`` to include when setting ``OS_AUTH_URL``.
# Neither is set by default.
HOST_IP=192.168.1.219
#HOST_IPV6=2001:db8::7
```

Chapter 2 - Installation of Devstack

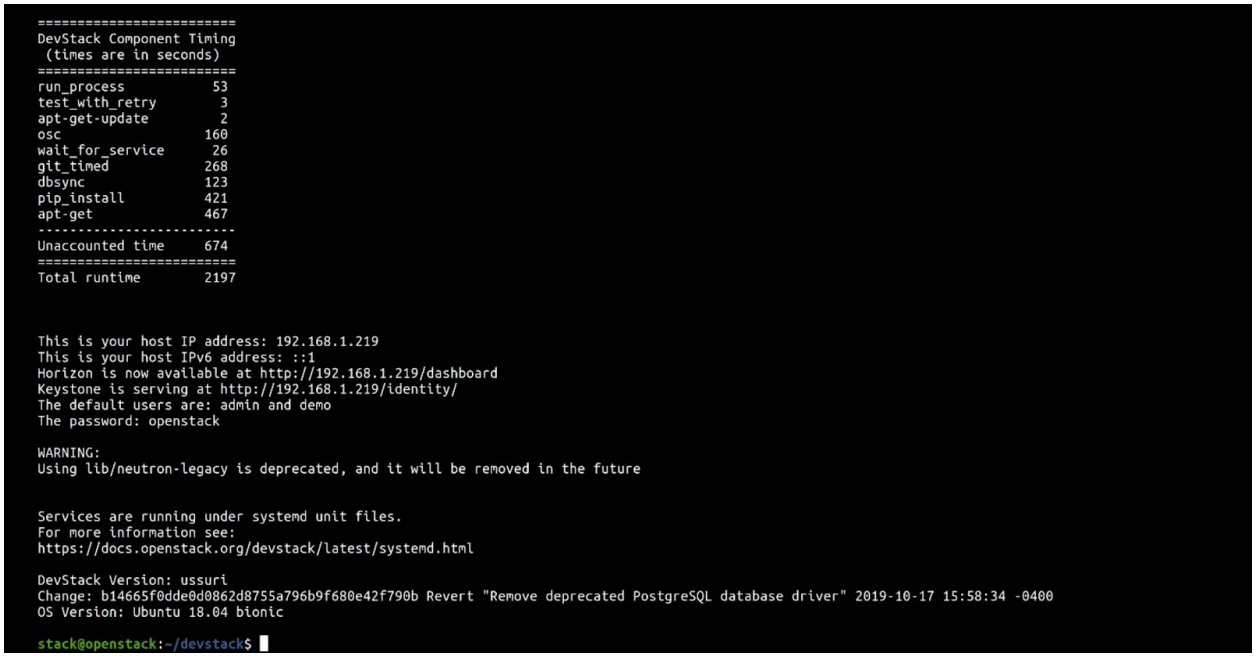
Step 1: After you have saved the local.conf document in the Devstack directory, it is time to start the installation of openstack through devstack. We use the stack.sh script for this, which can be found in the Devstack folder, to execute the script use ./stack.sh. As shown in the image below.

Note: If you have followed all the instructions in the manual you will not encounter any error messages.If you, however, if you encounter errors it is wise to look up the error message on the internet how to solve it.



Step 3: When the installation is finished you will see a screen like the one shown below. This screen will give you some general information e.g. Openstack version name, username, IP address for the web interface of Openstack and password.
Note: You can already log in but you cannot do anything yet so we do not log in to the Openstack website yet.

Note: If you want to log in to the openstack web interface, navigate with a browser to the IP address of your server. The condition is that your server is in the same network, if not, then port 80 must be forwarded to the internet, but this is a security risk. Login with username ”admin” and password “openstack”. As shown in the image below.



Chapter 3 - Add images

Images are files with a preloaded installation. You can use an image to create Virtual Machines. Images are equipped with specific drivers to boot the operating system within Openstack. The images can be downloaded via the internet. Because the server does not have a graphical user interface, we must use the tool wget. The tool will download the specified file from the internet and the file will be saved in the current folder. The tool is installed by default in Ubuntu 18.04. We will provide a link for six images:

Fedora Cloud Image
wget
https://download.fedoraproject.org/pub/fedora/linux/releases/30/Cloud/x86_64/images/Fedora-Cloud-Base-30-1.2.x86_64.qcow2

CentOS 7 Cloud Image
wget http://cloud.centos.org/centos/7/images/CentOS-7-x86_64-GenericCloud.qcow2

Ubuntu 18.04 Cloud Image
wget http://cloud-images.ubuntu.com/bionic/current/bionic-server-cloudimg-amd64.img

Debian 10 Cloud Image
wget
http://cdimage.debian.org/cdimage/openstack/current-10/debian-10-openstack-amd64.qcow2

CoreOS Cloud Image
wget
https://stable.release.core-os.net/amd64-usr/current/coreos_production_openstack_image.img.bz2

Note: As you can see the CoreOS Cloud Image is packed. You can easily unpack it with the command "bunzip2 coreos_production_openstack_image.img.bz2" remember that after extracting the file the extension changes from .bz2 to .img

Arch Linux Cloud Image

wget https://linuximages.de/openstack/arch/arch-openstack-LATEST-image-bootstrap.qcow2

Login credentials for the images
Debian: **debian**

Fedora: **fedora**

Ubuntu: **ubuntu**

RHEL: **cloud-user**

CentOS: **centos**

coreos: **core**

Arch Linux: **arch**

Gentoo: **gentoo**

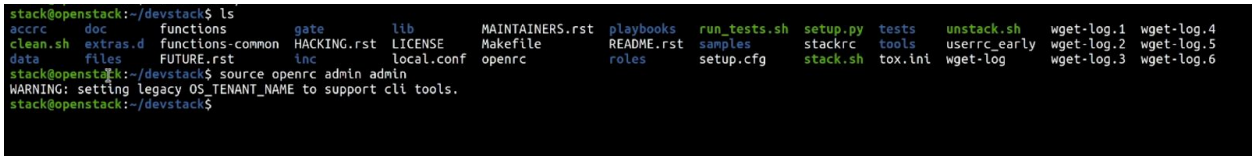
OpenSUSE: **root**

Cirros:

username: **cirros**

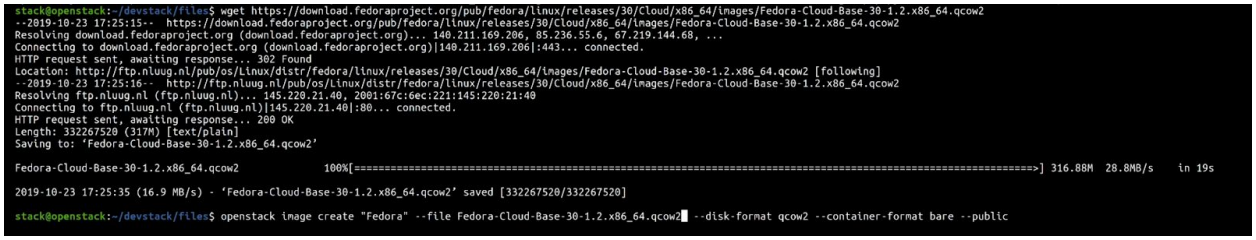
Password: **gocubsgo**

Step 1: Only the Openstack administrator can add images, to log in to the CLI interface as Openstack Administrator, navigate to the devstack directory that is in the home directory of the user stack. Then use the command source openrc admin admin. You will receive the notification “WARRING: setting legacy OS_TENANT_NAME to support cli tools”. This confirms that you are logged into the CLI tools of Openstack.



Step 2: Now navigate to the folder files you can do this with the command "cd files".

Step 3: Download the images that you want to use within openstack using wget. In the example below we downloaded the Fedora image. Then we used the command **“openstack image create "Fedora" --file Fedora-Cloud-Base-30-1.2.x86_64.qcow2 --disk-format qcow2 --container-format bare --public”**.

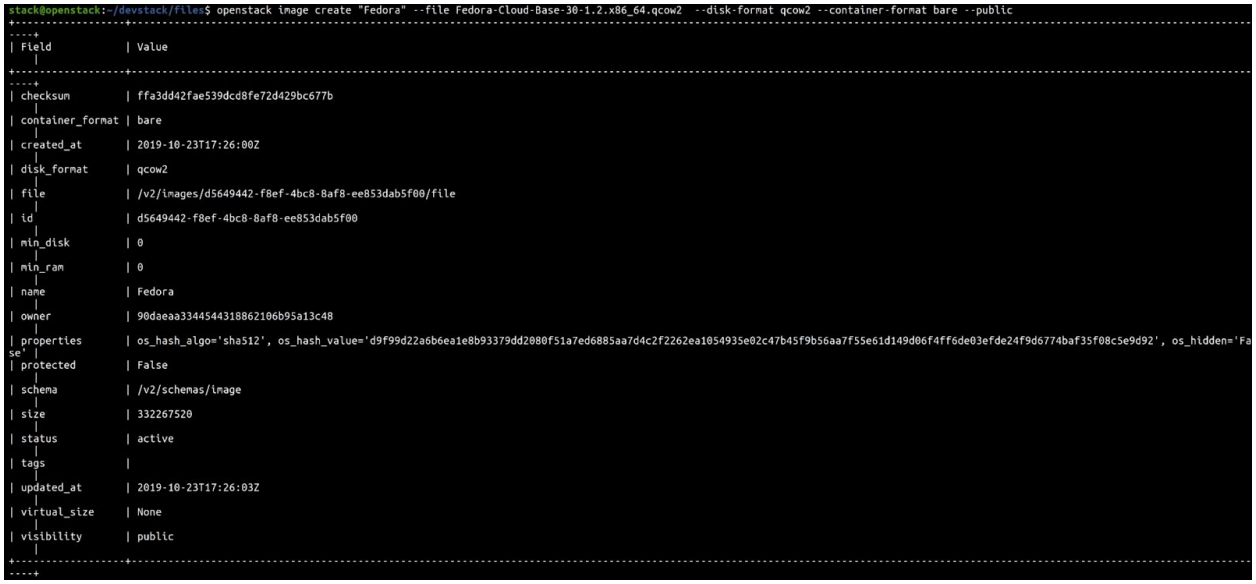


Note: openstack image create# tells openstack that we want to create an image
"Fedora"# tells openstack what name we want to give to the image within Openstack
--file Fedora-Cloud-Base-30-1.2.x86_64.qcow2#
--disk-format qcow2# tells openstack what imageformat we want to use to create the image
--container-format bare# tells openstack what container format we want to use to create the image. The container format is bare unless the image is packaged in a file format such as ovf that includes additional metadata related to the image.
--public# is needed to publish it to everyone

Repeat step 3 for each image that you want to add to openstack, you can easily download the file and execute the command **“openstack image create "name of the image" --file name of the**

image file. qcow2 (or img) --disk-format qcow2 --container-format bare --public”. Remember to always change the name and the image file name that you use.

Once the image has been added you will get the following output in the terminal as shown in the image below.



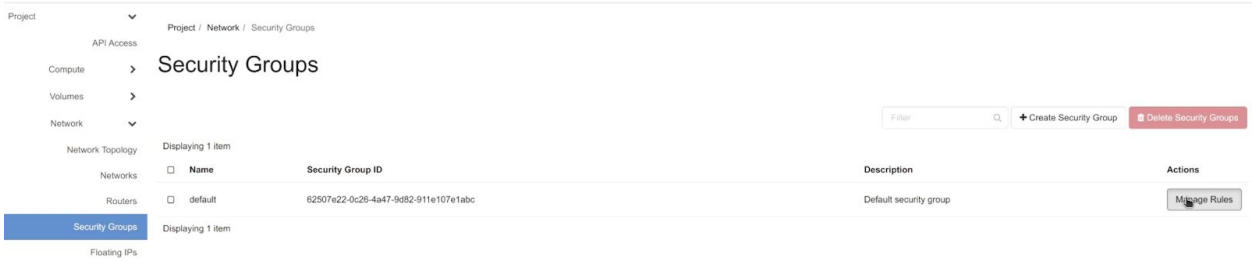
Step 4: Navigate with a web browser to the IP address and login with the login credentials. Verify that the images you have added are in the list, to check this, navigate to compute and then to images in the menu on the left.

Chapter 4 - Security Groups

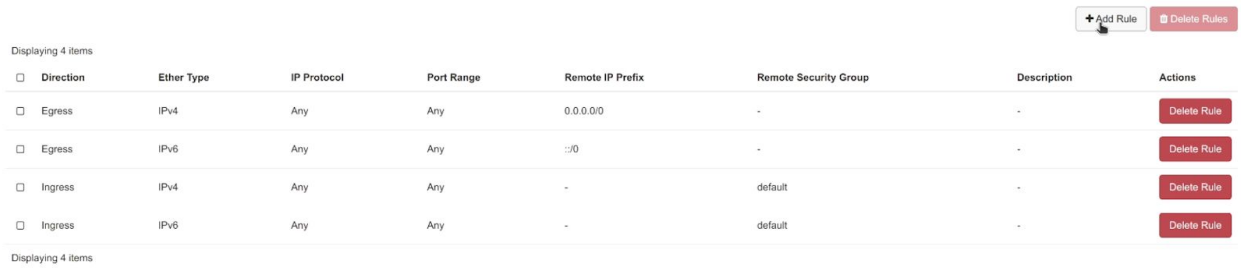
Openstack uses so-called security groups which are a set of firewall rules that are assigned by default to every vm that you create. If you want to manage the virtual server remotely, we have to change a few default settings. You can compare the firewall rules with the firewall rules from your (home) router.

Step 1: Navigate with a web browser to the IP address and login with the login credentials you have set.

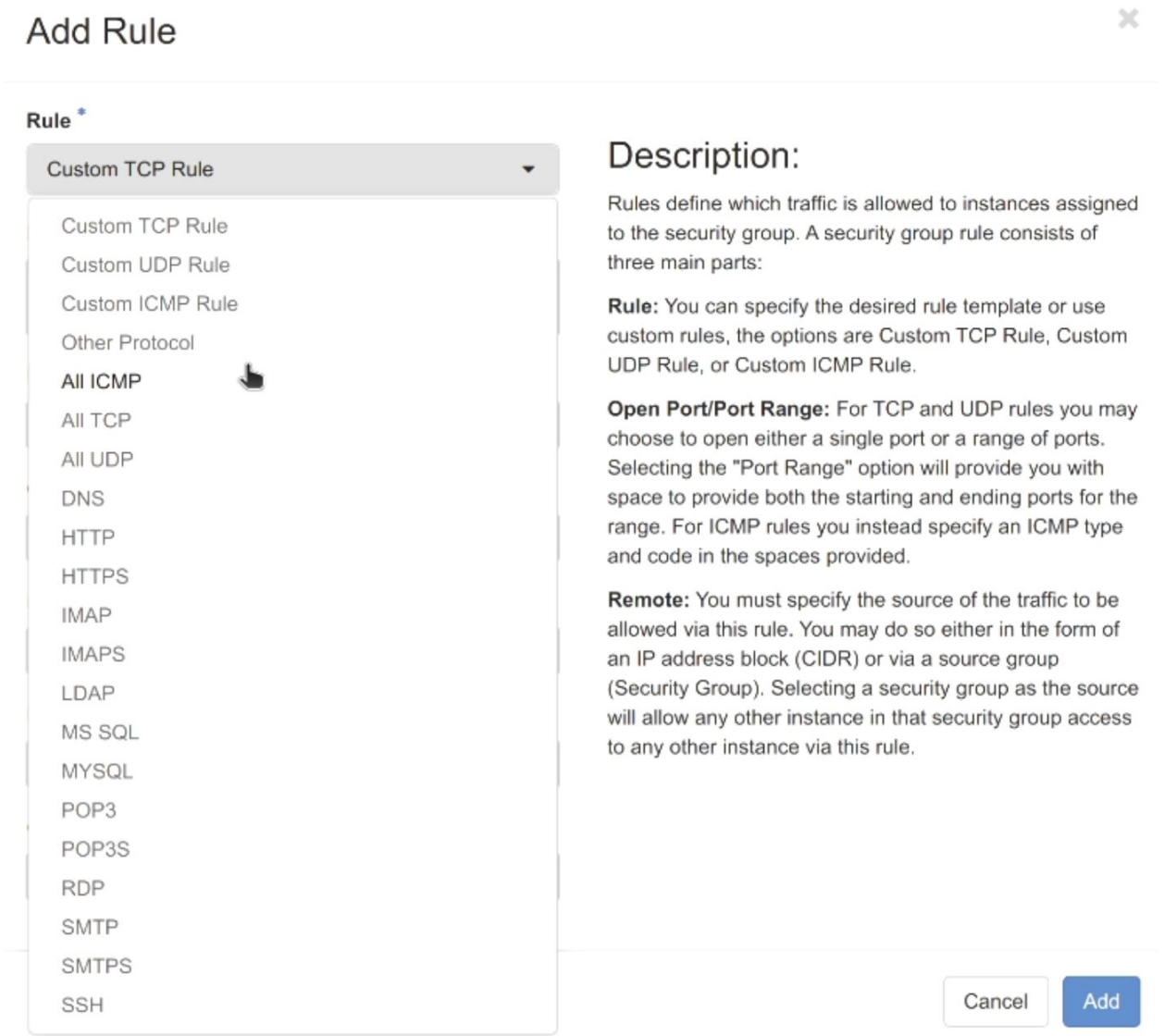
Step 2: Navigate to “Network” and then to “Security Groups” in the menu on the left, as seen in the image below. Click on "manage rules". Then click on “Default Security Group”, because we want all virtual machines to have the same firewall rules.



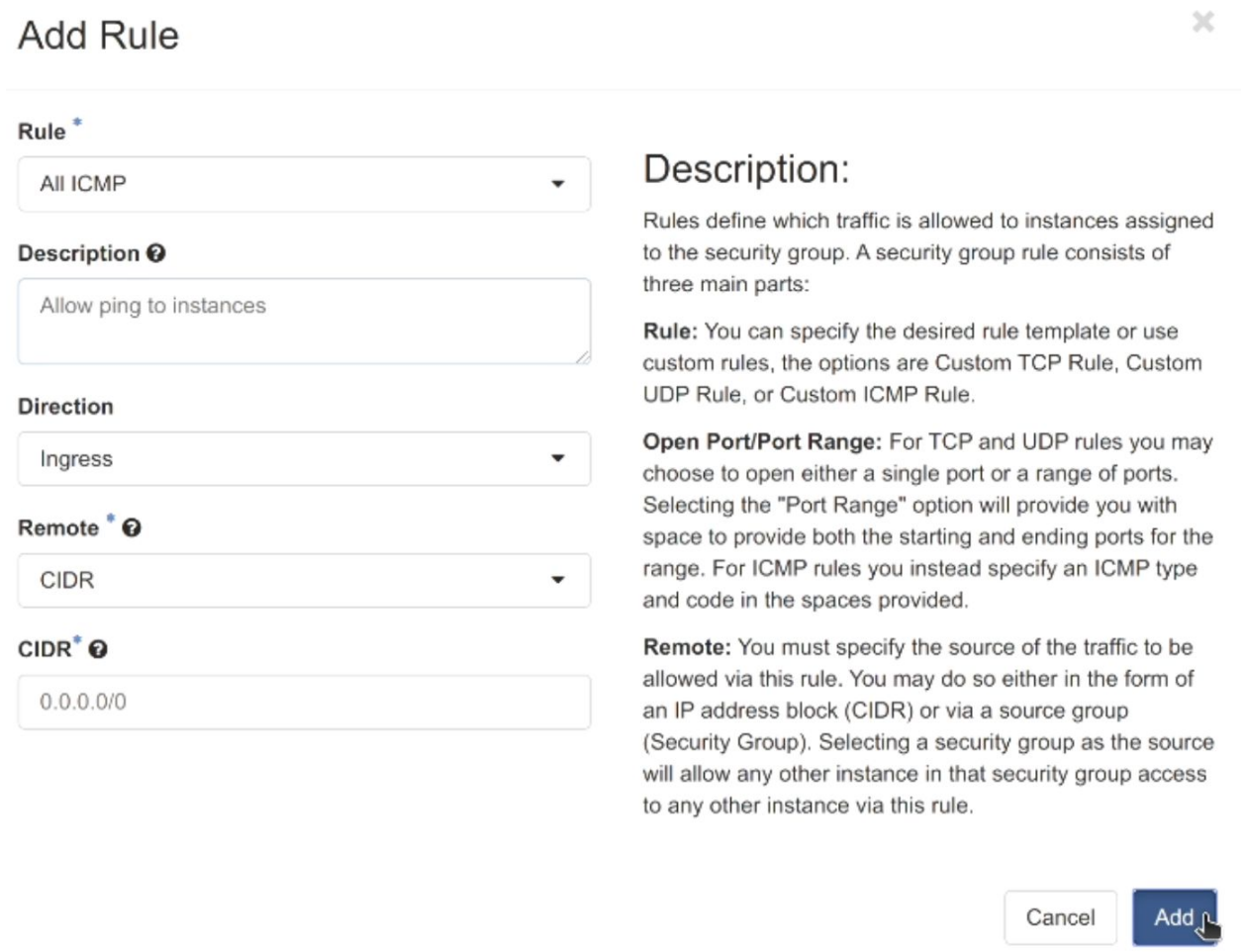
Step 3: In this window you choose "Add Rule"



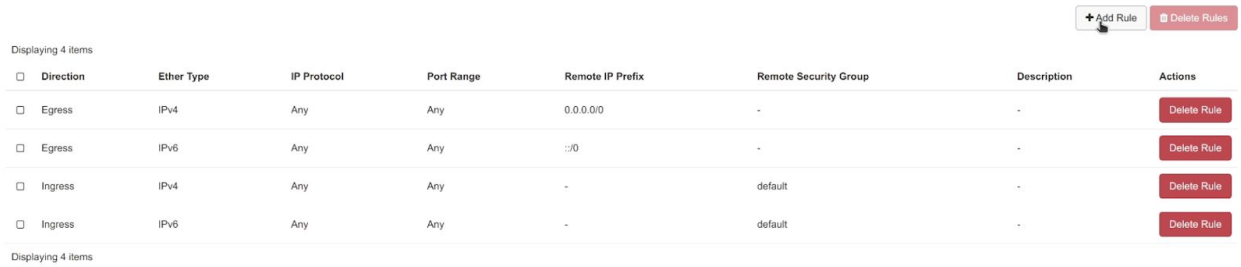
Step 4: This step is optional as we allow ping through the firewall here, we use this to test whether the virtual machine has already been started or not.



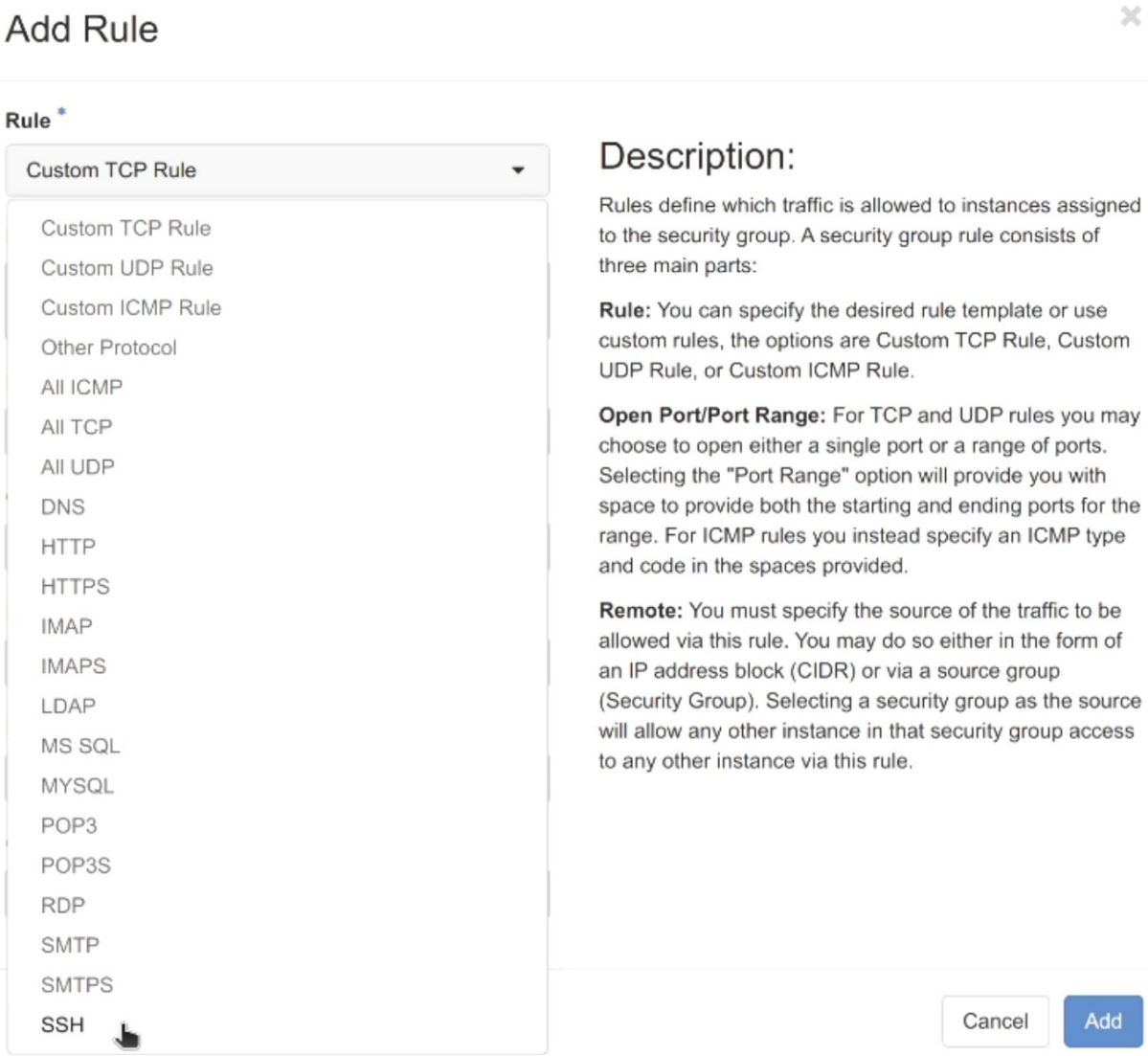
Then we give the firewall rule a description so that we know what the firewall rule is for. Then click on the button “Add”.



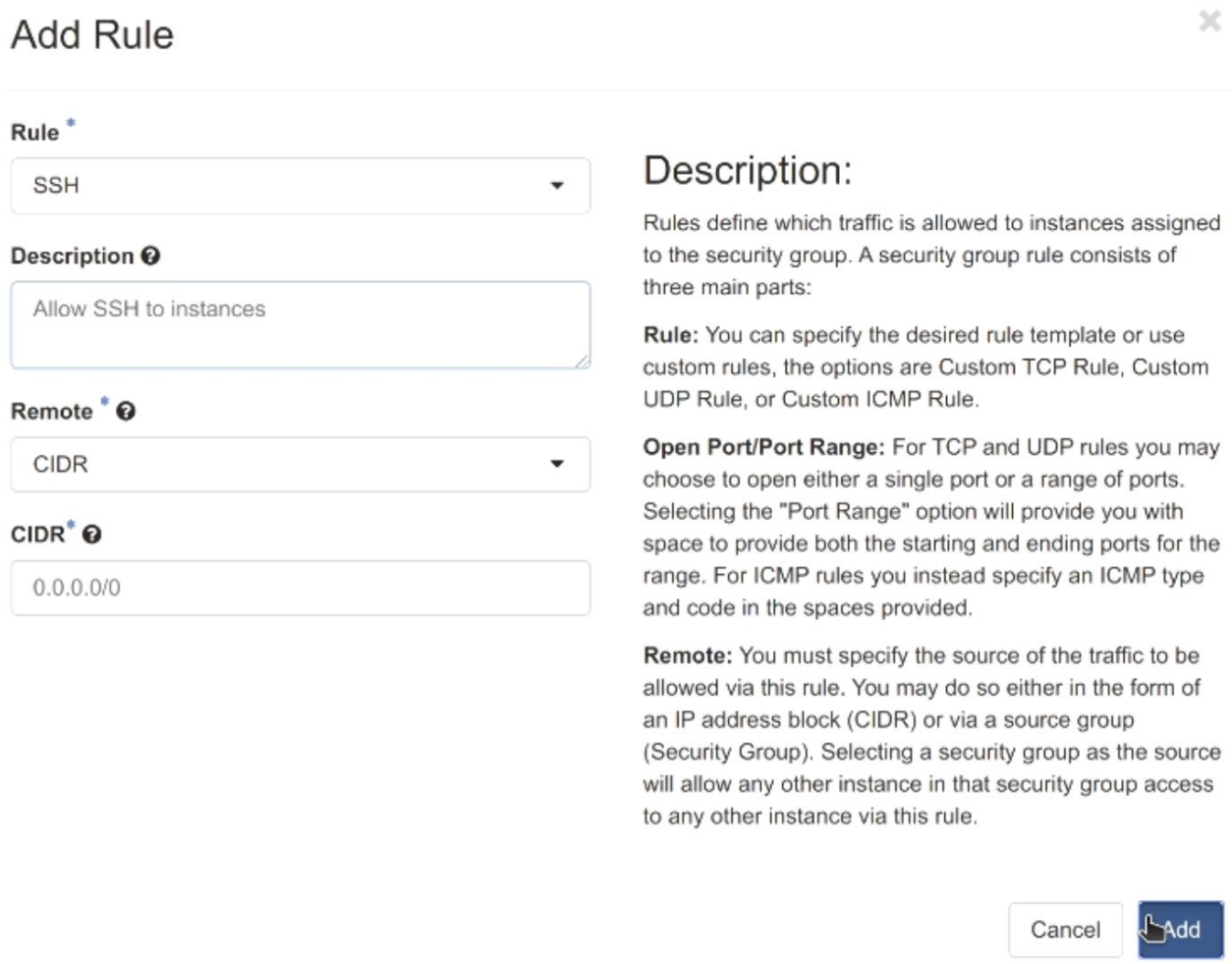
Step 5: click the "Add Rule" button again.



We ensure with the SSH rule that the SSH port [22] is open in the firewall, so the SSH connection can be set up.



Then we give the firewall rule a description so that we know what the firewall rule is for. Then click on the button “Add”.



Chapter 5 - SSH Keys

Step 1: Connect to the server where Openstack is installed through Devstack with SSH.
Note: Make sure that after logging in via SSH you switch to the user "stack", you do this with the command su "stack".

Step 2: You will need to generate a SSH key for the user you specified. With this SSH key you will not need to enter the password again for this user. This can be a good practice if you use a lot of users; because you do not need to remember the passwords for them. You can create an ssh key pair with the command **“ssh-keygen -t rsa”**

Using SSH keys also give you an extra layer of authentication security; since it is immensely hard to crack a SSH key compared to a normal password. Because the SSH key that we use is for this manual, we leave the SSH key password blank.

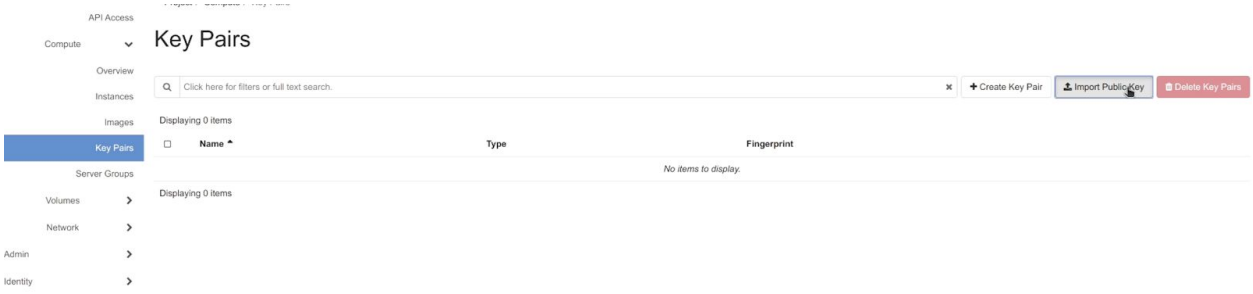
```
stack@openstack:~$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/stack/.ssh/id_rsa):
Created directory '/home/stack/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/stack/.ssh/id_rsa.
Your public key has been saved in /home/stack/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:VDGm7xmK372Aa+Z5CxXxEOhzF39Ooq7icAw+L3XAQ8w stack@openstack
The key's randomart image is:
+---[RSA 2048]----+
|       o .Oo     |
|        E+ =.    |
|       +o . .o   |
|       .*.... o o|
|      . S=+. . = |
|     . +.=.o.   .|
|      =.*.+.     |
|      .*+=.o.    |
|       *Bo+oo.   |
+----[SHA256]-----+
stack@openstack:~$
```

Now you have to import the public key to Openstack, we achieve this by copying the public key. Using the cat command the file will copied in /home/stack/.ssh/id_rsa.pub.

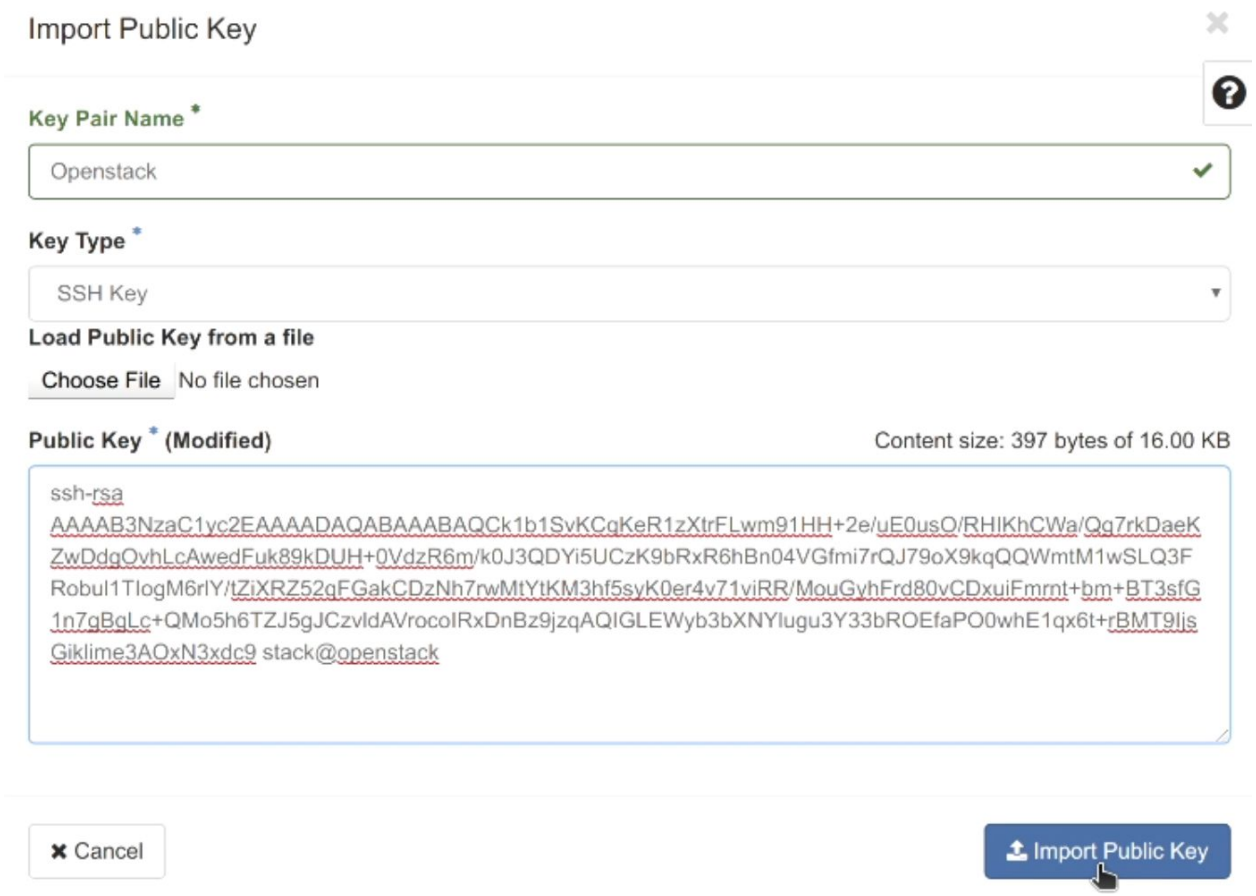
Note: Never gives the private key to anyone or imports it to Openstack!

Step 3: Navigate with a web browser to the IP address and login with the login credentials.

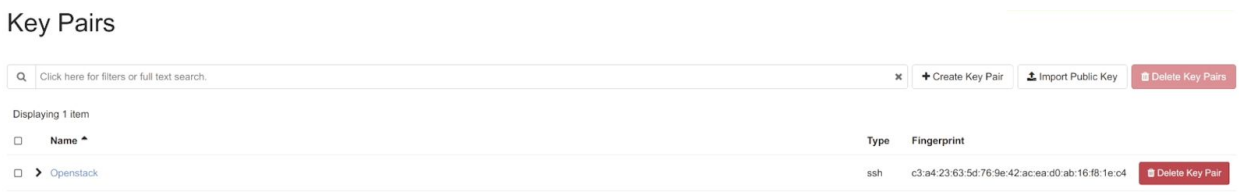
Step 4: Navigate to “Compute” and then to “Key Pair” in the menu on the left. Then choose import “Public Key” as seen in the image below.



Specify a name for the SSH key we use in the example below "Openstack" and select SSH key at the "key Type". Then paste the public key in the Public Key box. Then click on the "Import Public Key" button.



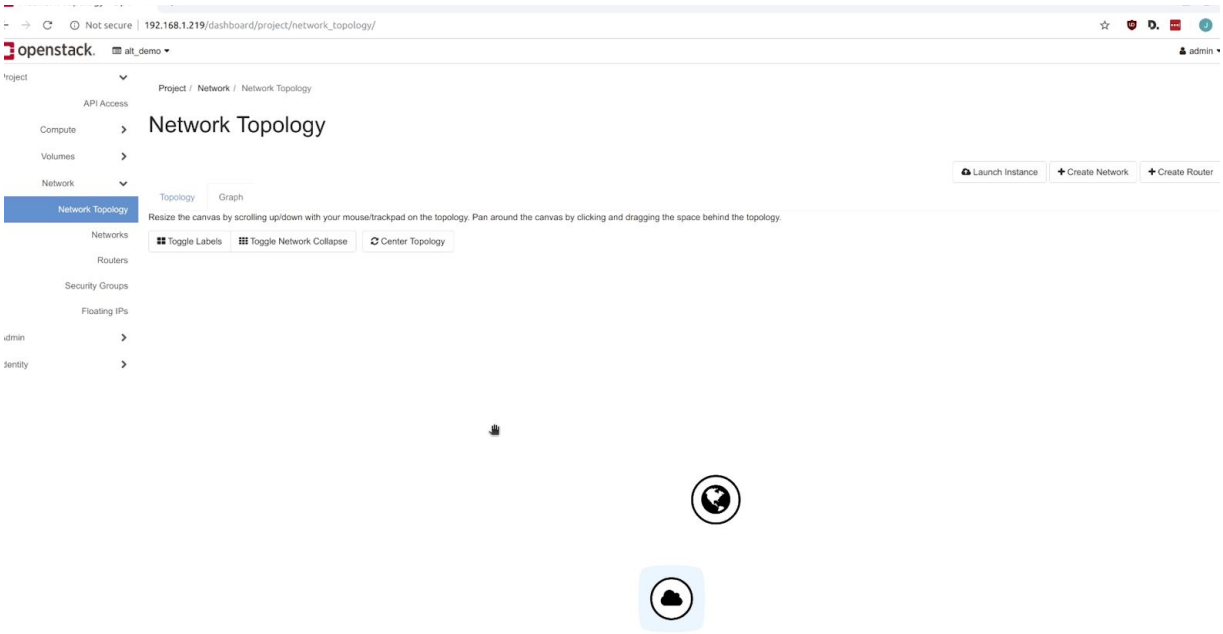
In the image below you can see that the SSH key has been successfully added.



Chapter 6 - Add router

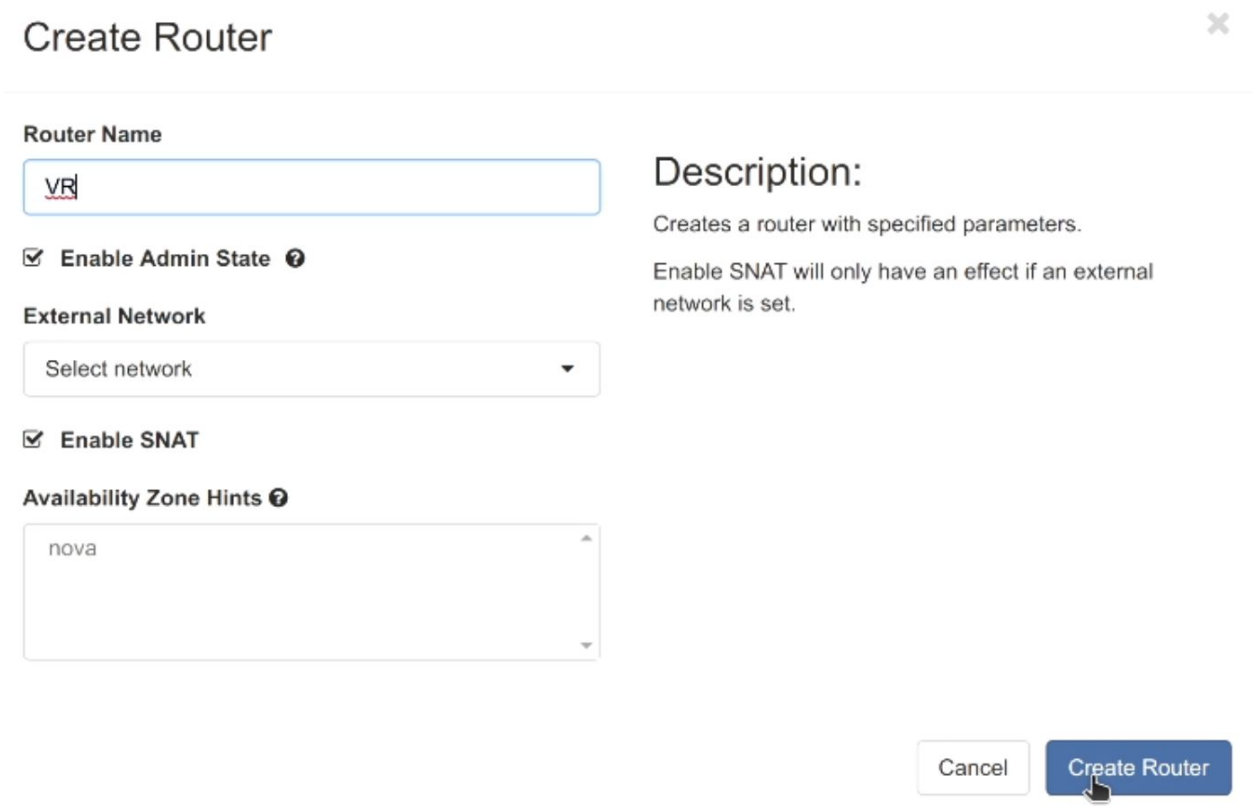
Step 1: Navigate with a web browser to the IP address and login with the login credentials. navigate to “Network” and then to “Network topology” in the menu on the left.

Note: The globe is the public network for the virtual machines. The cloud is the internal network for the virtual machines. The two networks must be connected to each other by with a router, you will do this in the next step.

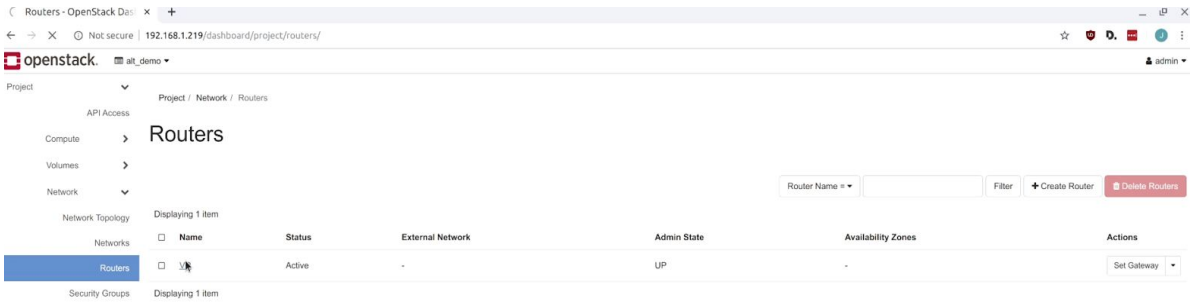


Step 2: You will then have to create and configure a router, to do this, navigate to Network and then to Routers in the menu on the left. We give the router the name “VR” and we click on the button “Create Router”.

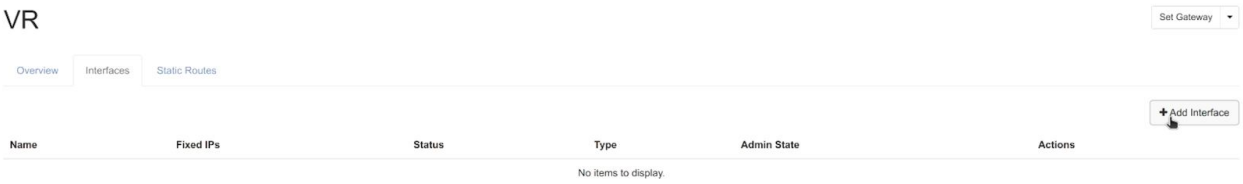
Note: you can determine the name of the router yourself.



Click on the (VR) router , as seen in the image below.



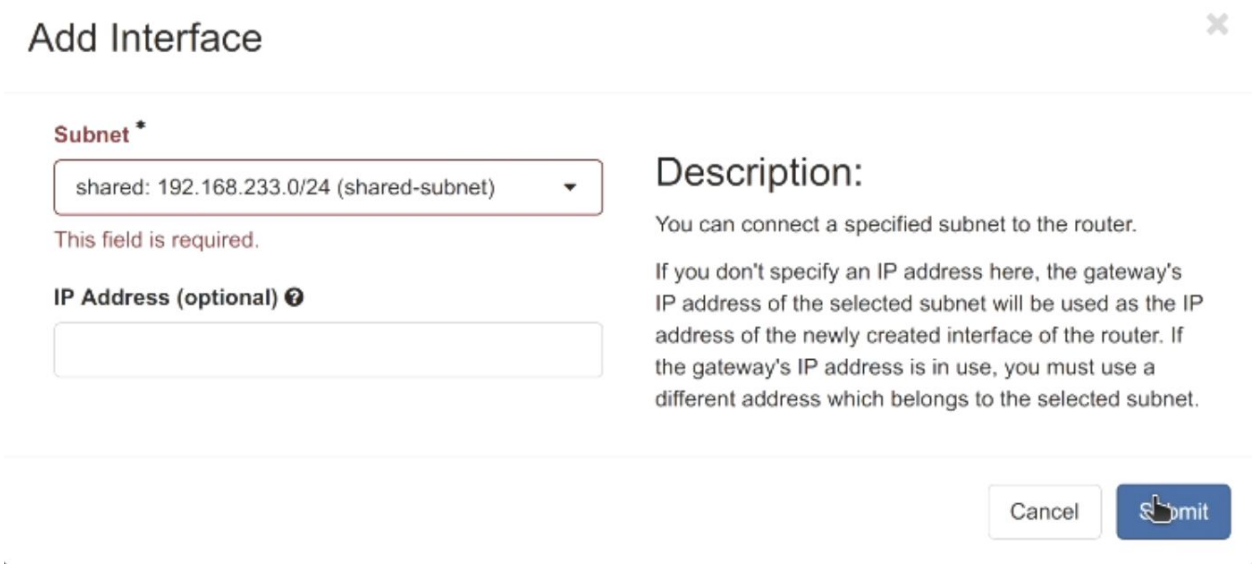
Click interfaces and then right again interfaces , as seen in the image below.



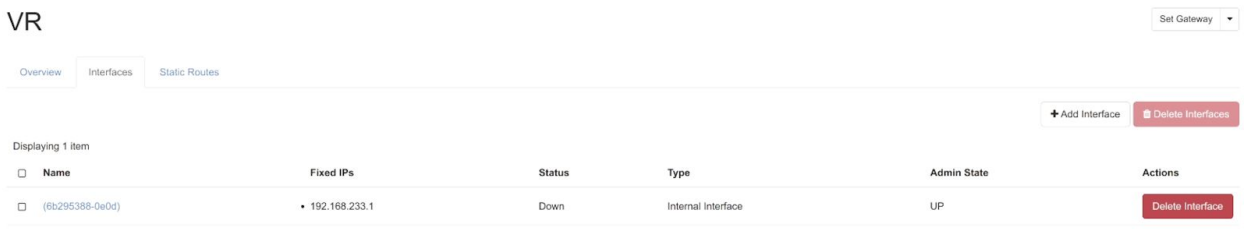
Select the subnet that we created then configure the IP address/ The subnet is in our case 192.168.233.0./24.

Then click on the button “Submit”.

Note: The gateway address will be automatically assigned to the interface.

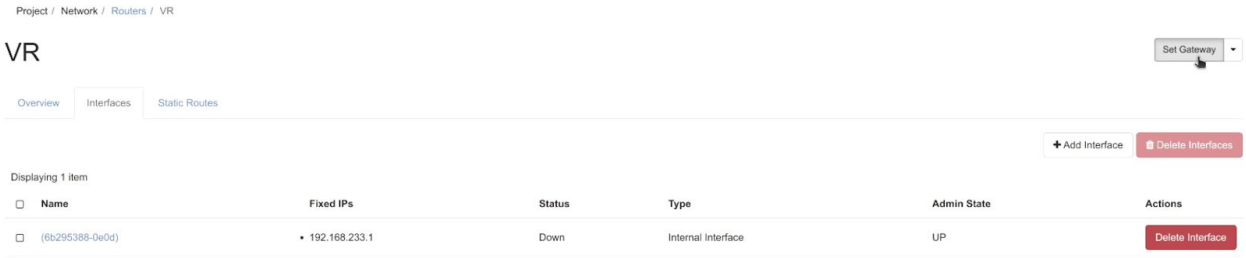


When the default gateway has been successfully added you will get the following output

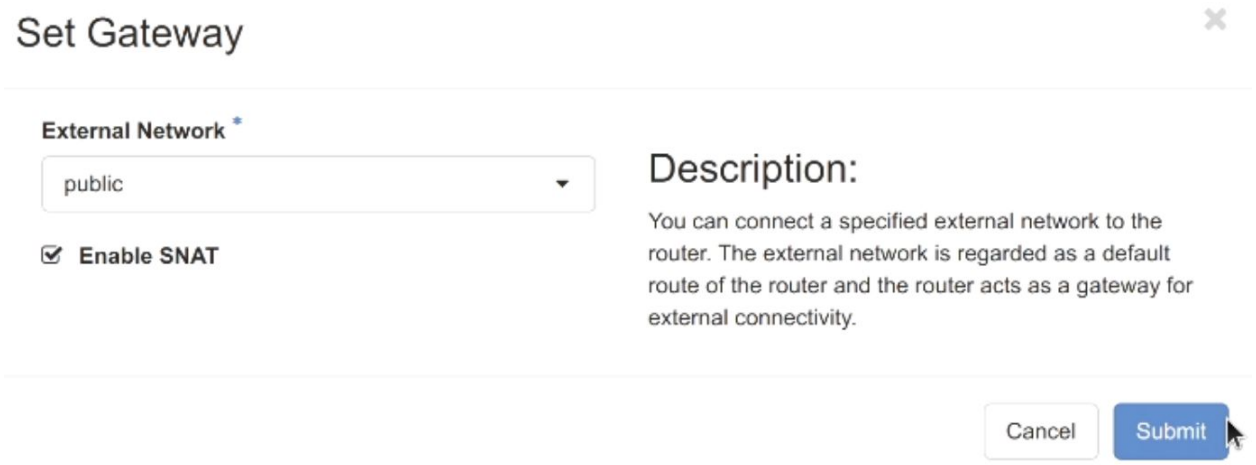


Note: Now our network needs another gateway to ‘reach’ the outside world or internet, we use the external network adapter for that.

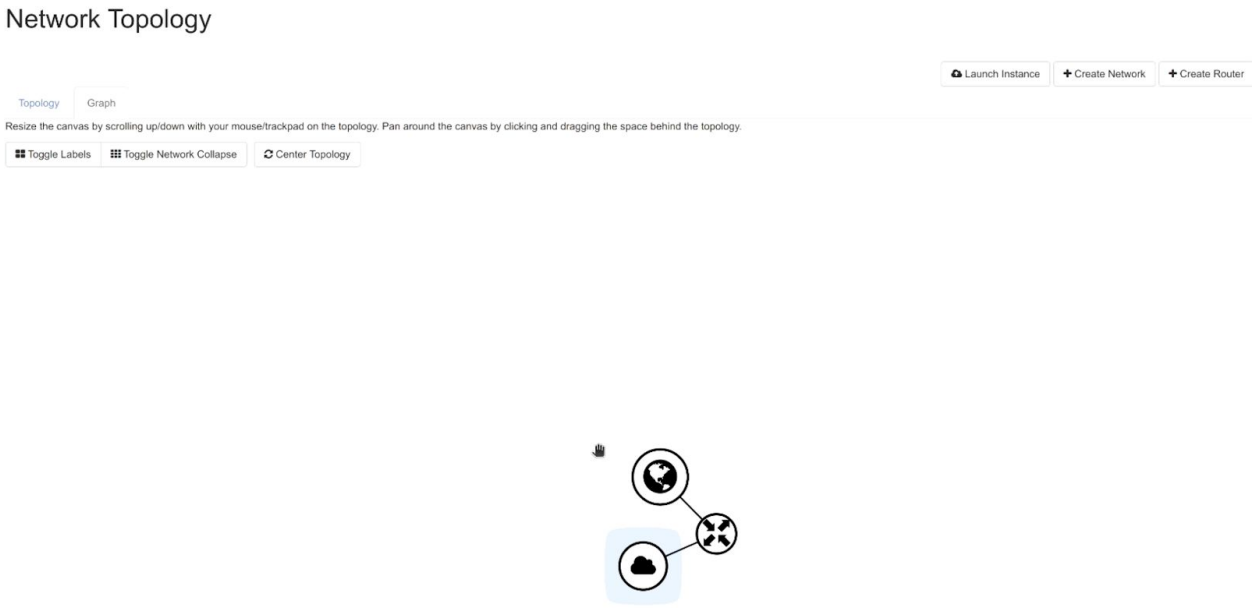
Step 4: Click on "Set Gateway" to set the gateway for the network.



Select the public network in the "External Network" menu, then click on the button “Submit”.



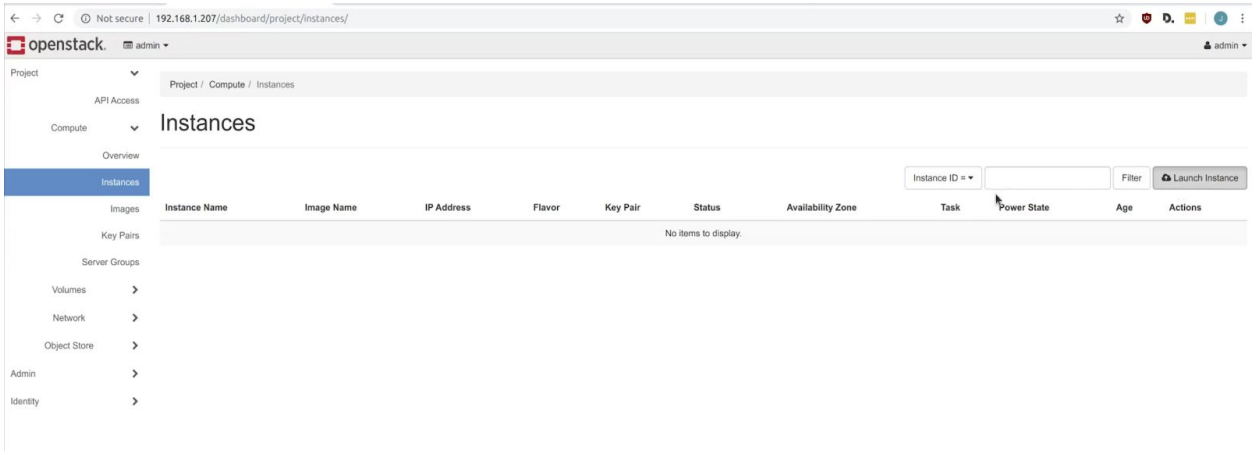
Step 5: Navigate to “Network” and then to “Network topology” in the menu on the left. Confirm that the internal and public are connected through the "VR" router.



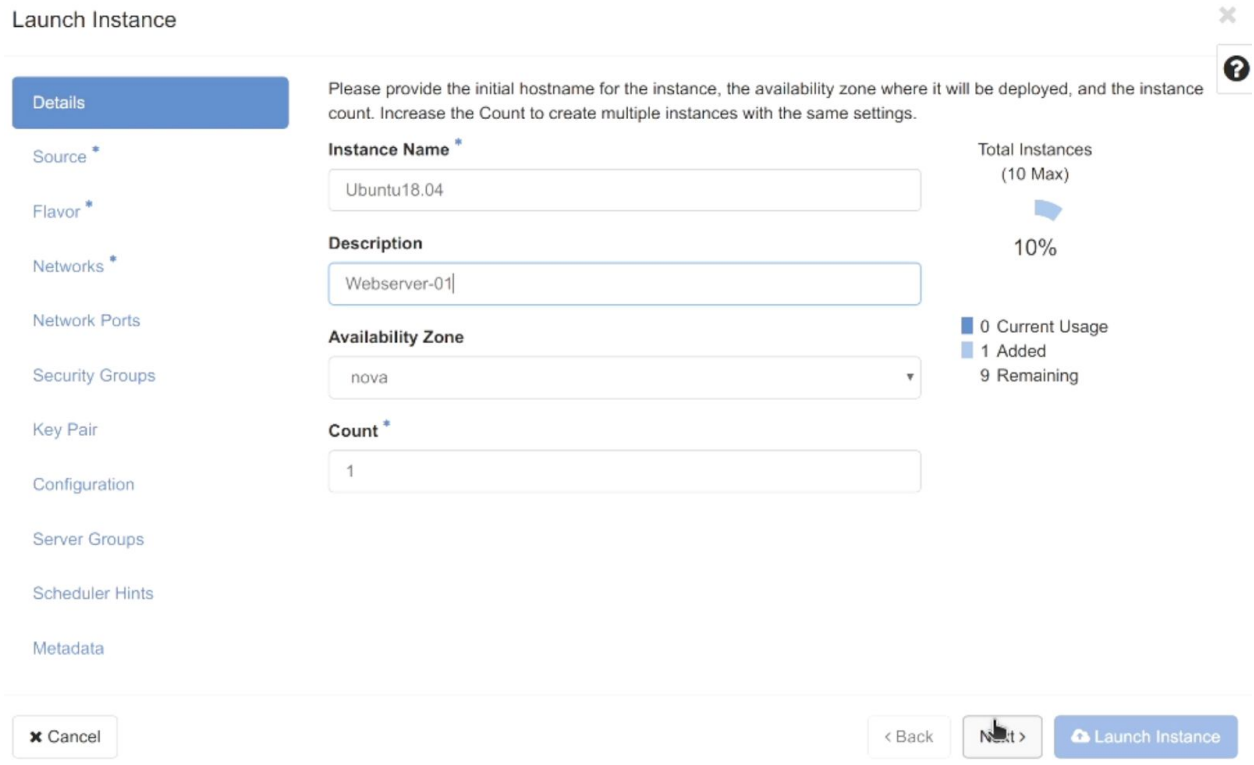
Chapter 7 - Create instances

Step 1: Navigate with a web browser to the IP address and login with the login credentials.

Step 2: Verify that there are no Instances, to check this, navigate to Compute and then to Instances in the menu on the left. As seen in the image below.

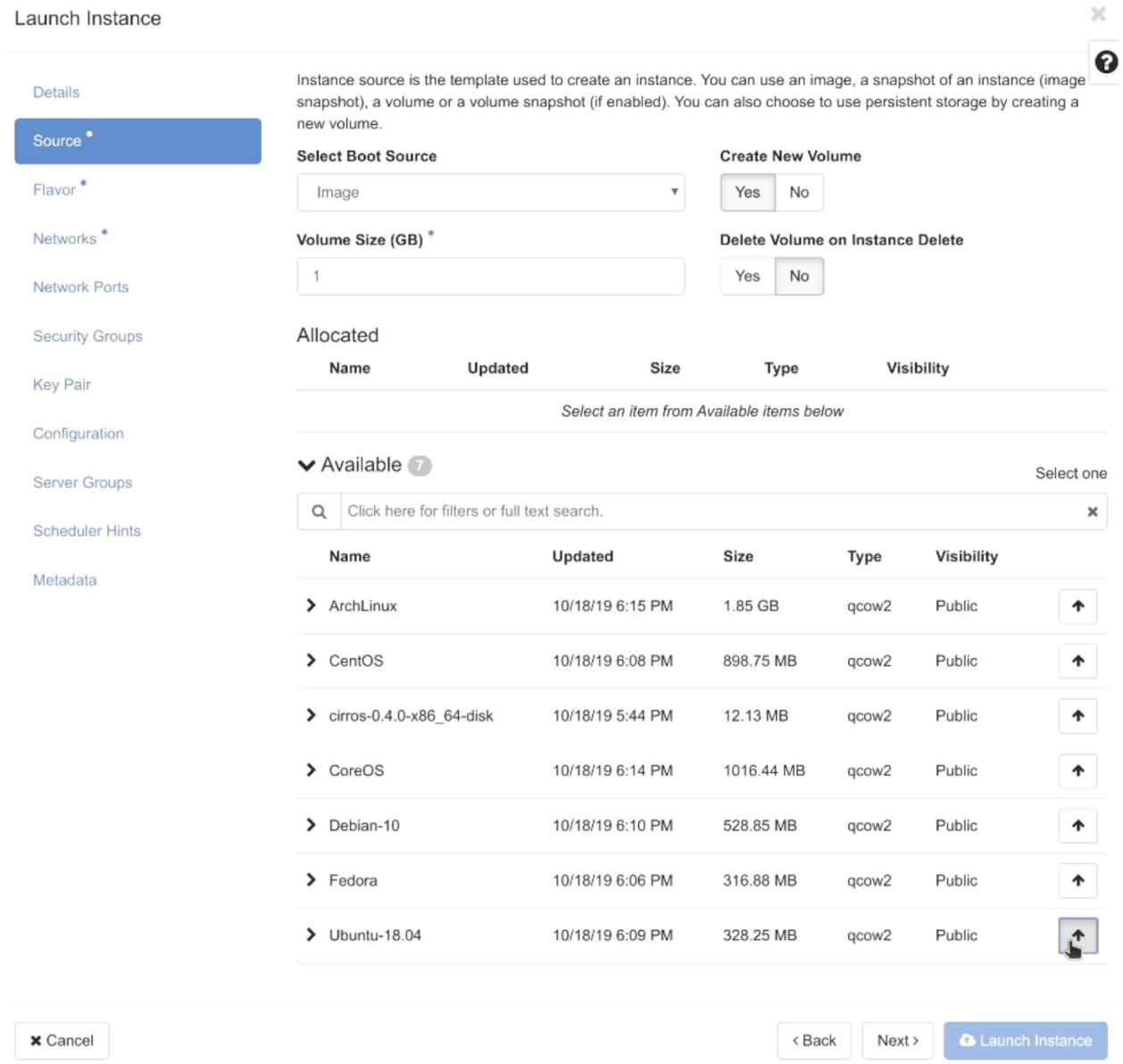


Step 3: Click on “Launch Instance”

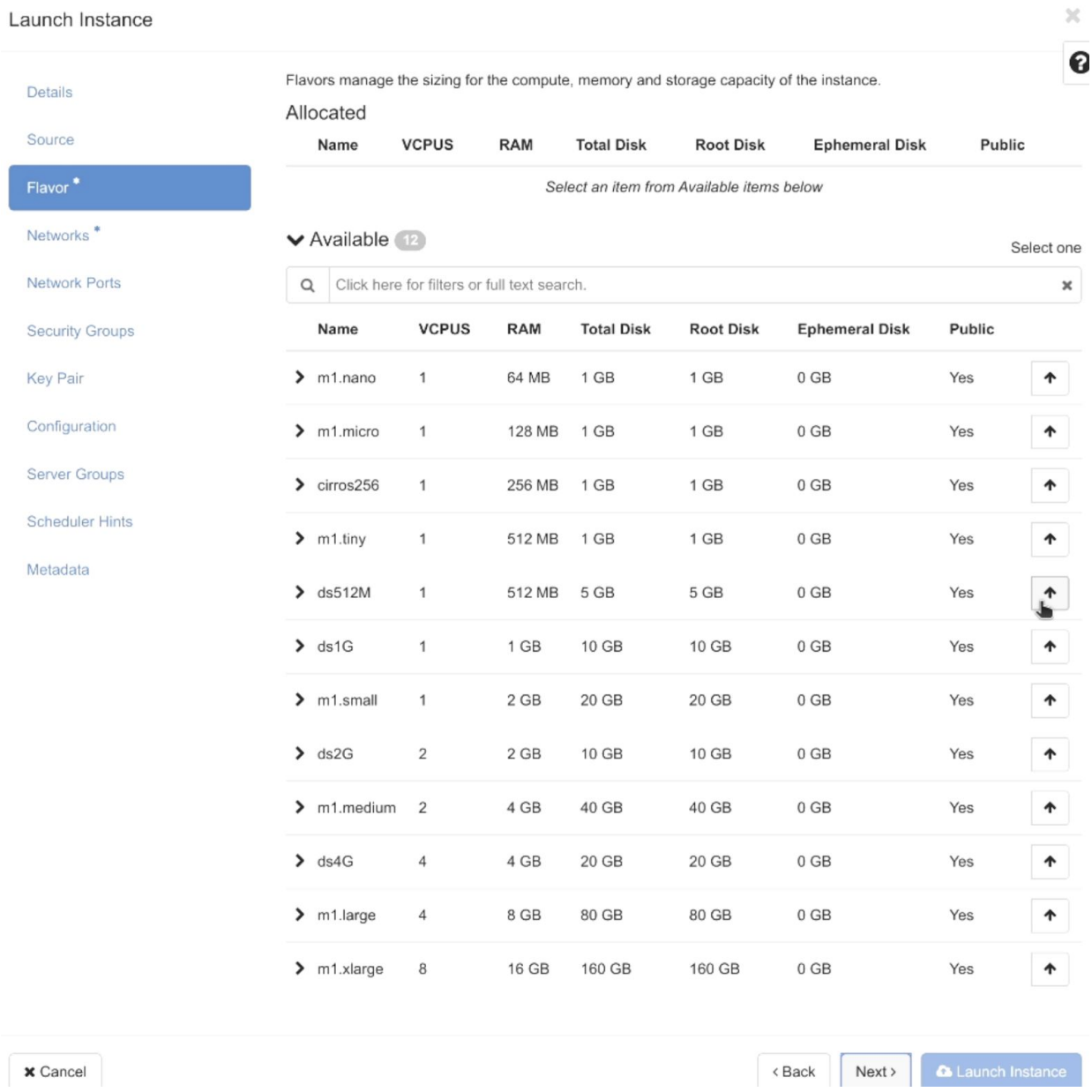


Now you have to give the instance a name. In this example the instance is called Ubuntu18.04, also you have to give a description for the instance because this instance will run a webserver you can you give the description “Webserver-01”. You want 1 copy of the instance so we choose the number 1. After that click on “Next”

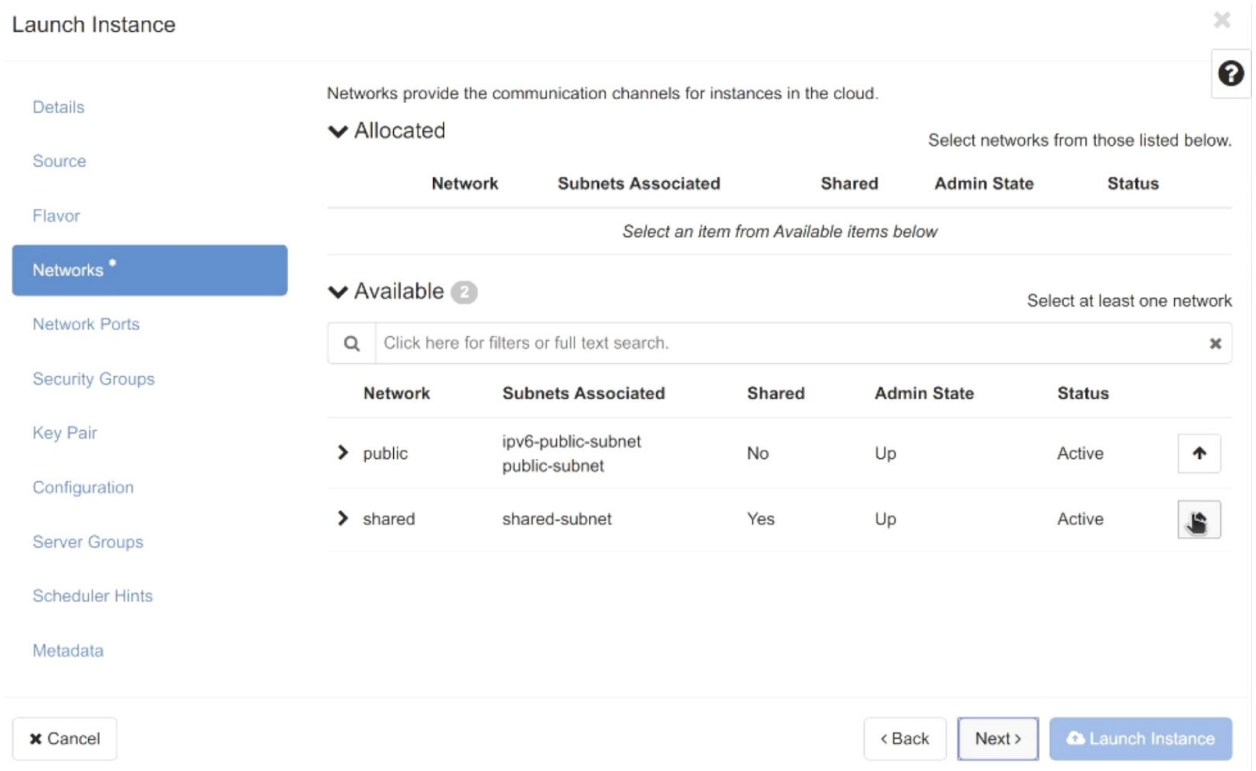
Then, you must choose the image and yes we choose the option "Delete Volume on Instance Delete ". And select the Ubuntu18.04 image, then click on the button “Next”.



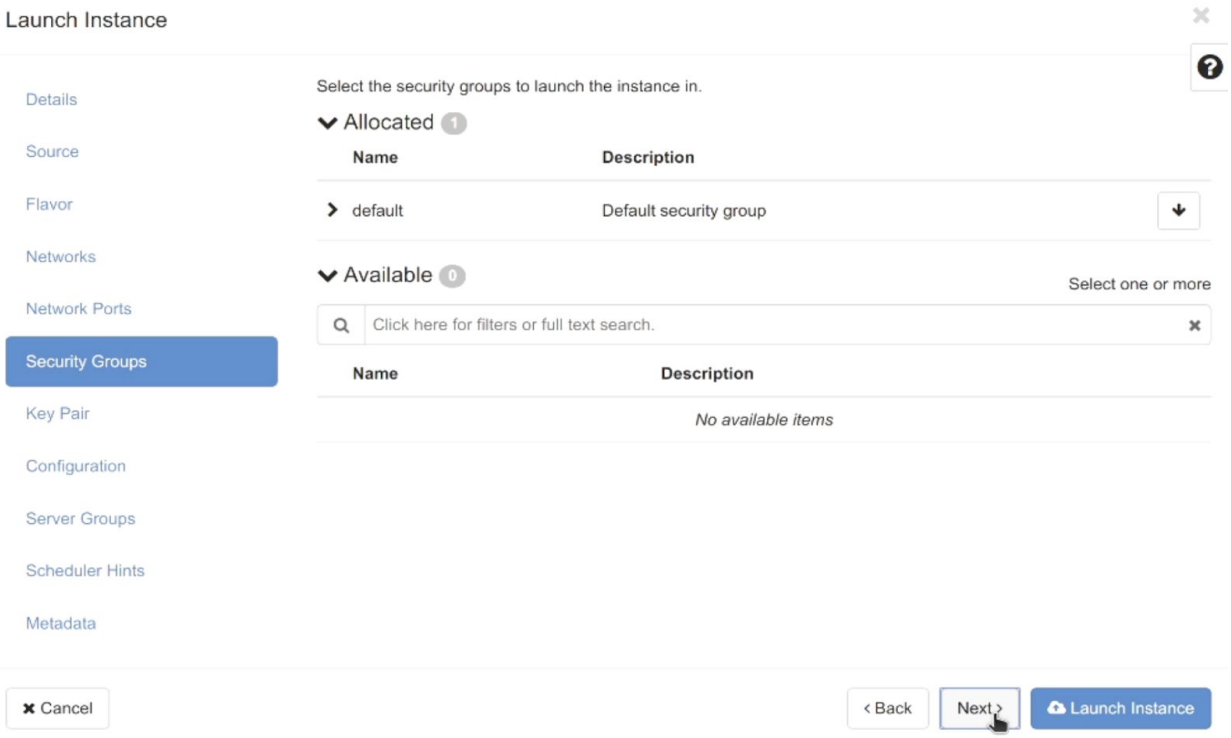
Next you have to choose the flavor of the instance, a flavor determines how many cores, RAM and how much hard disk space the virtual machine will have. In this example we will use “ds2G”. Then click on the button “Next”



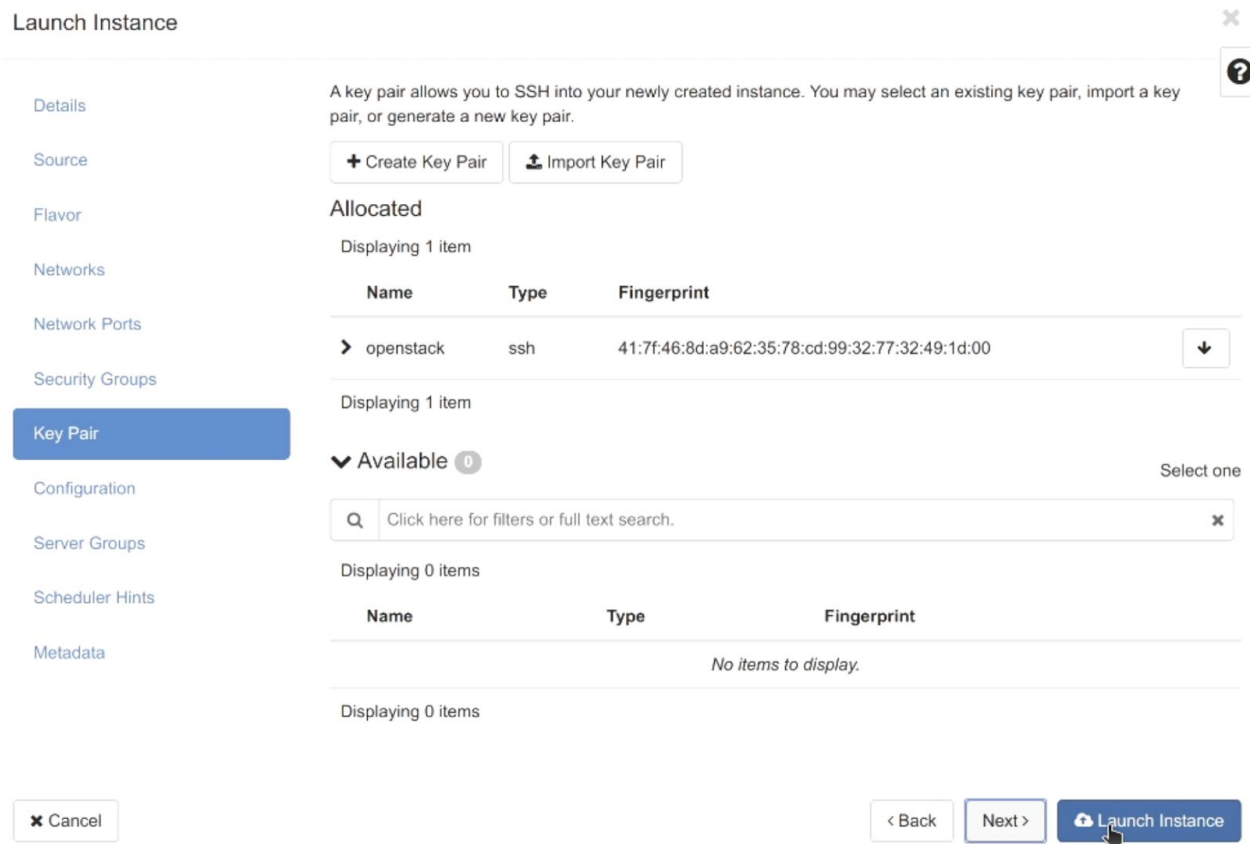
In this window we select the private network, and then click on “Next”.



On the next screen choose the default security group, and click on “Next”.

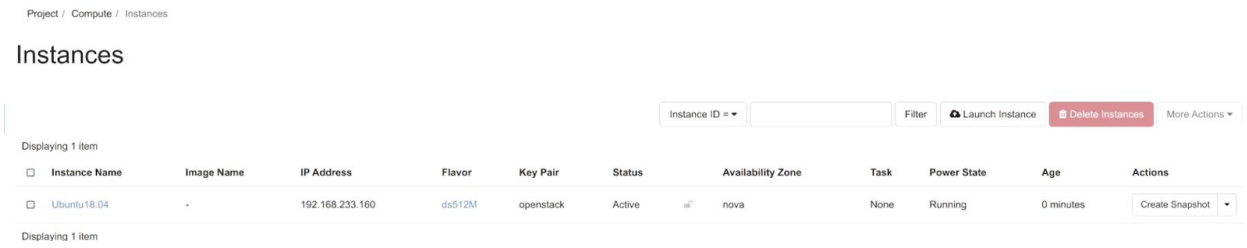


On the next screen choose the SSH key, and click on “Launch Instance”.

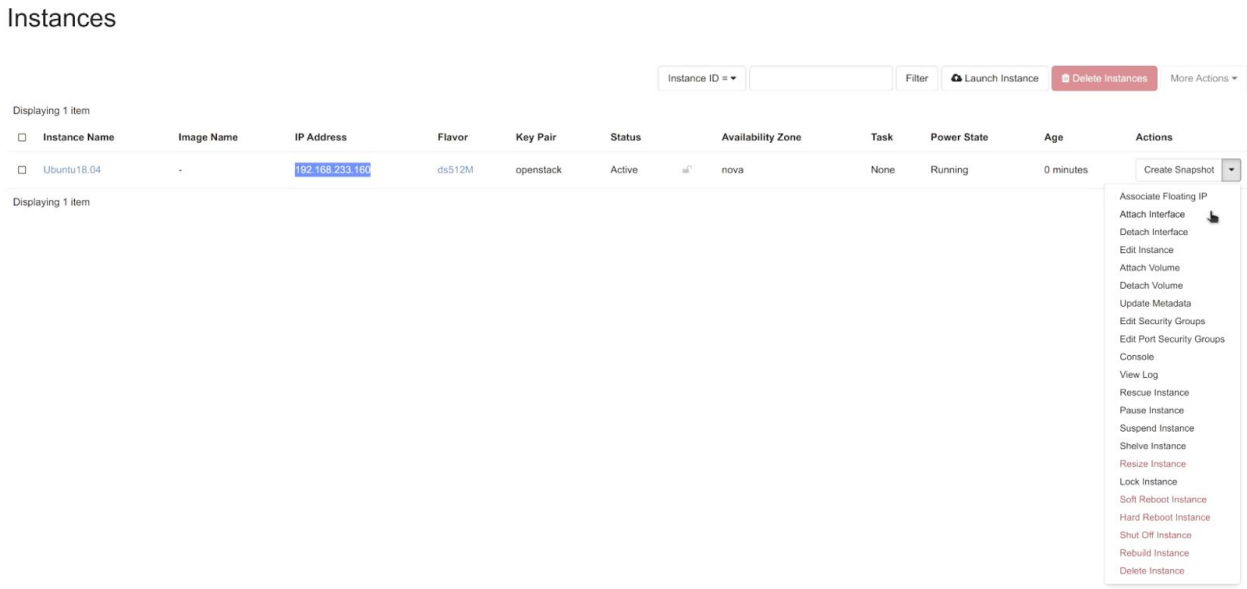


The virtual machine will be created on the server with the flavor and will be started.

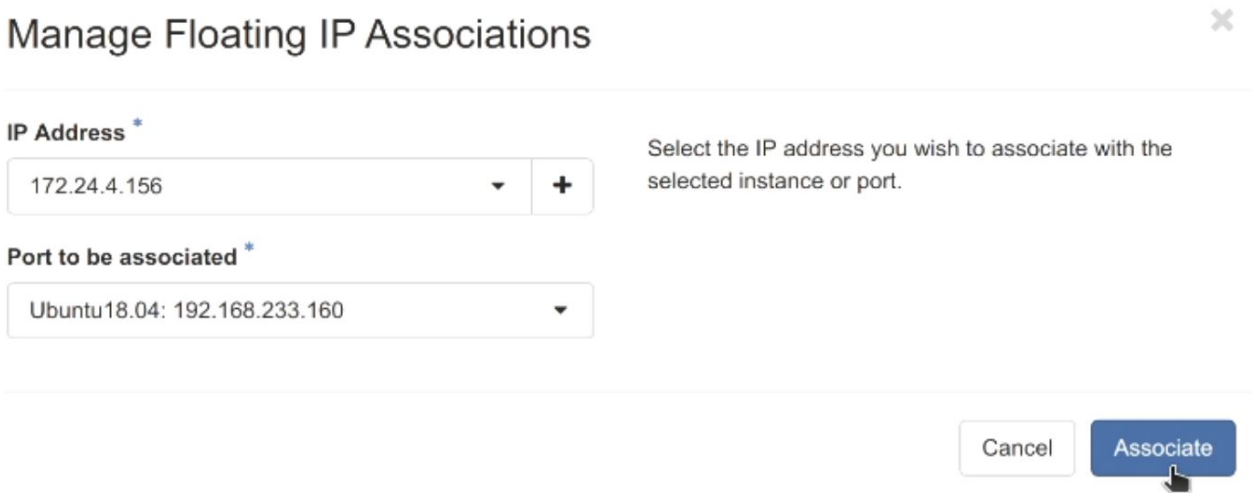
If the status changes to "Running", you can click on the name of the virtual machine and then open the console, you should see the console as a virtual monitor. As seen in the image below.



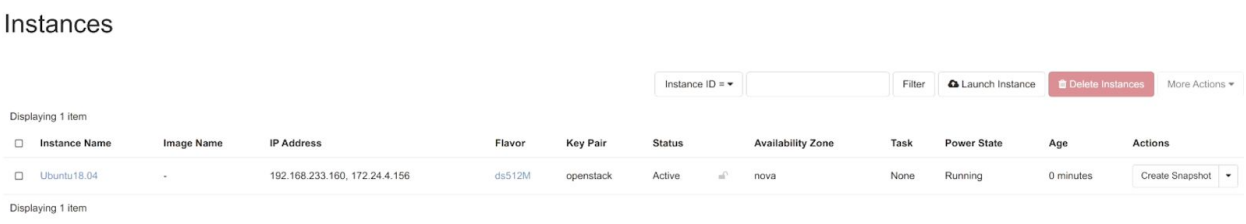
Step 4: Openstack works with so-called floating IP addresses which can be compared with external IP addresses. To assign a floating IP address to a virtual machine, click on associate floating IP as shown on the image below.



In the menu of the floating IP addresses we select the public IP range in this example this is 172.24.4.156 and then select the virtual machine, in this example that is “Ubuntu18.04” then choose associate.



We see that the public IP address 172.24.4.156 is assigned to Ubuntu18.04.



Step 5: We are now going to manage the virtual machine with SSH, connect to the server where Openstack is installed through Devstack with SSH.
Note: Make sure that after logging in via SSH you switch to the user "stack", you do this with the command su "stack".
With the command SSH and the user ubuntu we can set up an SSH connection to the virtual machine, for that you type ssh [user] in this case ubuntu @ [ip address] in this case 172.24.4.156 after which the system ask if we want to save the fingerprint you answer yes to the question. The SSH connection is then set up and you can manage the virtual machine from this point. You can also manage other virtual machines in the same way.

```
stack@openstack:~$ ssh ubuntu@172.24.4.156
The authenticity of host '172.24.4.156 (172.24.4.156)' can't be established.
ECDSA key fingerprint is SHA256:mIQtwSUBwB0okHYaMQyOV5YmBYyRTTbgzkmZTgP1swo.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '172.24.4.156' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-65-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Wed Oct 23 22:35:08 UTC 2019

  System load:  0.0               Processes:           78
  Usage of /:   20.5% of 4.67GB   Users logged in:     0
  Memory usage: 23%               IP address for ens3: 192.168.233.160
  Swap usage:   0%

0 packages can be updated.
0 updates are security updates.



The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ubuntu18:~$
```

final thoughts

If you did follow the tutorial, you can now create for example a Openstack-Centos instance. The next step you can take is to configure your server and run some tests. A nice way to test your server is to install a load balancer. If you like this idea, click on the link below and select the first search result. https://lmgtfy.com/?q=centos+load+balancer&s=g

References list

How To add Glance Cloud images to OpenStack - Computing for Geeks. (2019, 26 juni). Geraadpleegd op 25 september 2019, van https://computingforgeeks.com/adding-images-openstack-glance/

How to Install Your Own Cloud Platform with OpenStack in RHEL/CentOS 7. (2016, 25 augustus). Geraadpleegd op 25 september 2019, van https://www.tecmint.com/openstack-installation-guide-rhel-centos/

OpenStack Docs: Disk and Container Formats. (2019, 21 augustus). Geraadpleegd op 25 september 2019, van https://docs.openstack.org/glance/pike/user/formats.html

OpenStack User Tip: Add a new Image to OpenStack | Mirantis. (z.d.). Geraadpleegd op 25 september 2019, van https://www.mirantis.com/blog/openstack-user-tip-add-new-image-openstack/

OpenStack Docs: Setup DevStack. (z.d.). Geraadpleegd op 22 oktober 2019, van https://docs.openstack.org/sahara/latest/contributor/devstack.html

Appendix - Installing Ubuntu 18.04

Before you begin to read this manual, you must meet hardware requirements for Devstack which you will find below:

- Processor - 2 cores*
- Memory - 8GB
- Hard Drive - 60GB

*Keep in mind that different processors have different per-core performance. It is advised to use hardware that is suitable for running complex tasks e.g. a Intel Xeon or Core i7 series CPU or equivalent.

Step 1: To install the required software you first off need the ISO of Ubuntu 18.04 server, which you can download on the Ubuntu website. For convenience, we will share the URL where you can download the Ubuntu server: https://ubuntu.com/download/server. Save the ISO somewhere where the ISO can easily be found on your computer. Now you can create the VM with the specific hardware requirements above. Or you can make the ISO you downloaded bootable for use on bare metal hardware. This can be arranged by burning the ISO file to a DVD or flashing it onto a USB stick. You can find a lot of manuals and tutorials on how to do this on the Internet .

Having basic knowledge of networking is not compulsory but is advised since it will help you understand more how the network configuration works for Devstack. You should however be good to go in following this manual with no to barely any knowledge of networking.

Step 2 : You need to have a VirtualBOX environment or physical system ready. We presume you know how to install VirtualBOX as it is fairly simple to deploy on your system. In the unlikely event that you do not know how to install VirtualBOX we provide you with the following URL: https://www.wikihow.com/Install-VirtualBox

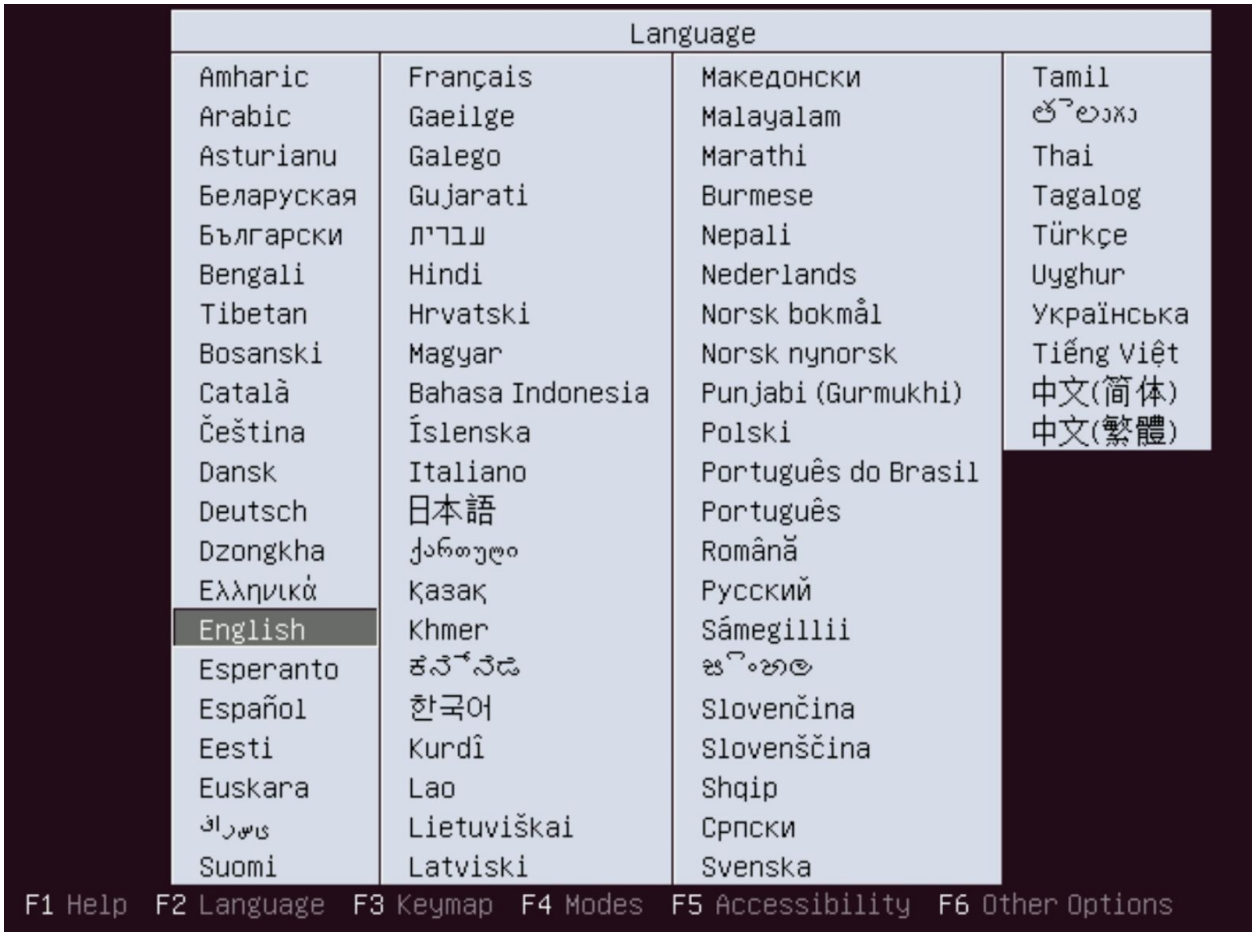
If you want to install this on a physical system, the steps will be depending on your system specifications. We advise you to follow the instructions from your system manufacturer how to startup the Ubuntu installation via USB or DVD.

Important notice!
Make sure you backup all your data if you are installing this on a physical system; since the Ubuntu installation will format the hard disks of your system. This may cause irreversal data loss.

If you are ready with doing the steps documented above and your system meets the minimal requirements of Devstack, we can proceed to installing Ubuntu 18.04.

Step 3: Now you need to install Ubuntu 18.04 on your virtual machine or bare metal hardware. Boot from the ISO file or DVD / USB if you are installing this on your physical machine.

Step 4: When you boot up your (virtual) machine with the boot from USB/ DVD / ISO properly configured, this menu will be shown on-screen. We will select ‘English’ as preferred language. This option is totally free of choice. You can choose whatever language you desire. Since we will only use the command-line interface in Ubuntu, the commands and command structures will be the same, the language setting will be irrelevant.

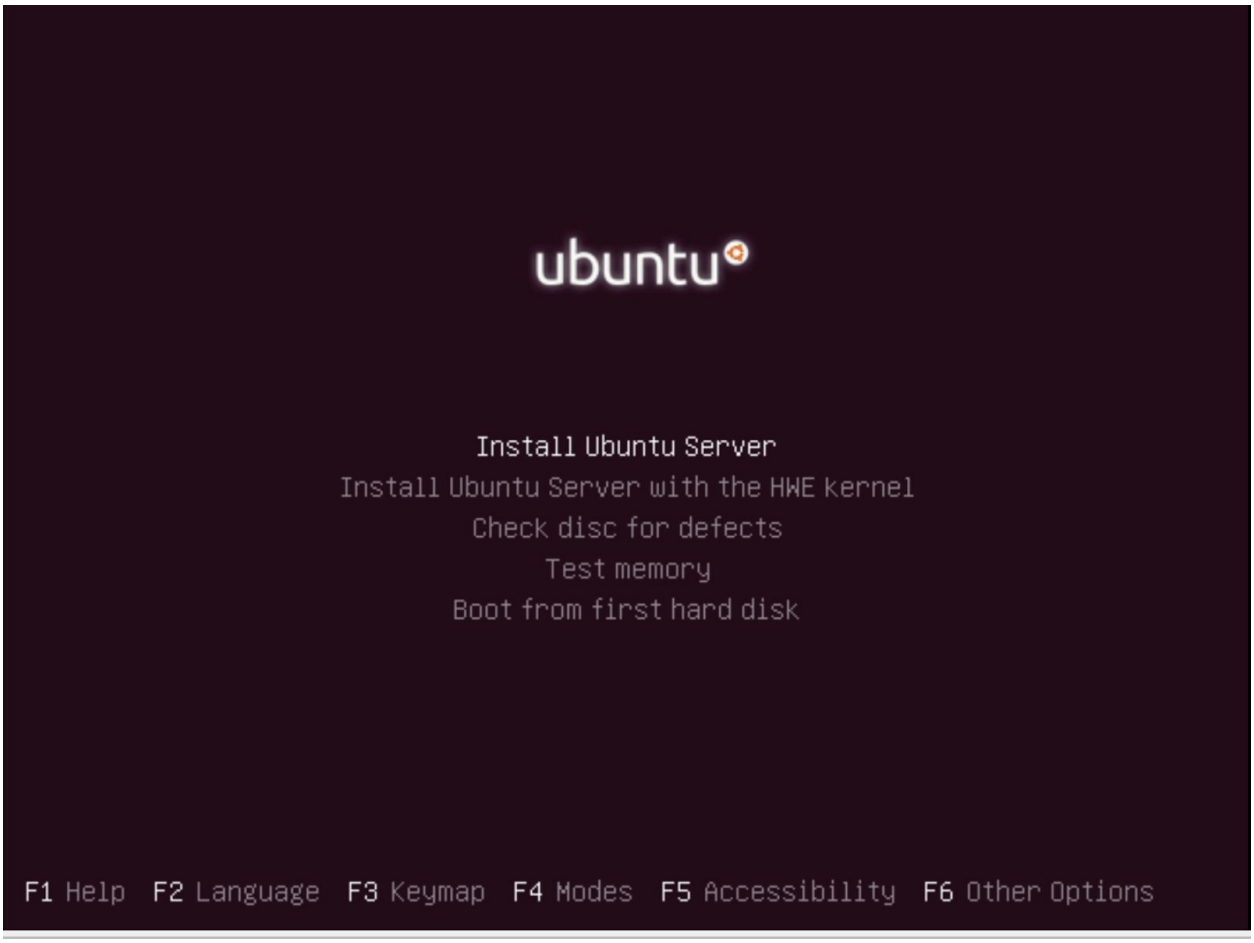


Step 5: After the language selection menu, the main install menu will be shown on screen. Make sure “Install Ubuntu Server” is showing in bold; this means the installation of Ubuntu is selected. You can use the arrow keys on your keyboard to change the selection if this is not the case.

After you made sure “Install Ubuntu Server” is selected, press “ENTER” on your keyboard.

Important notice:

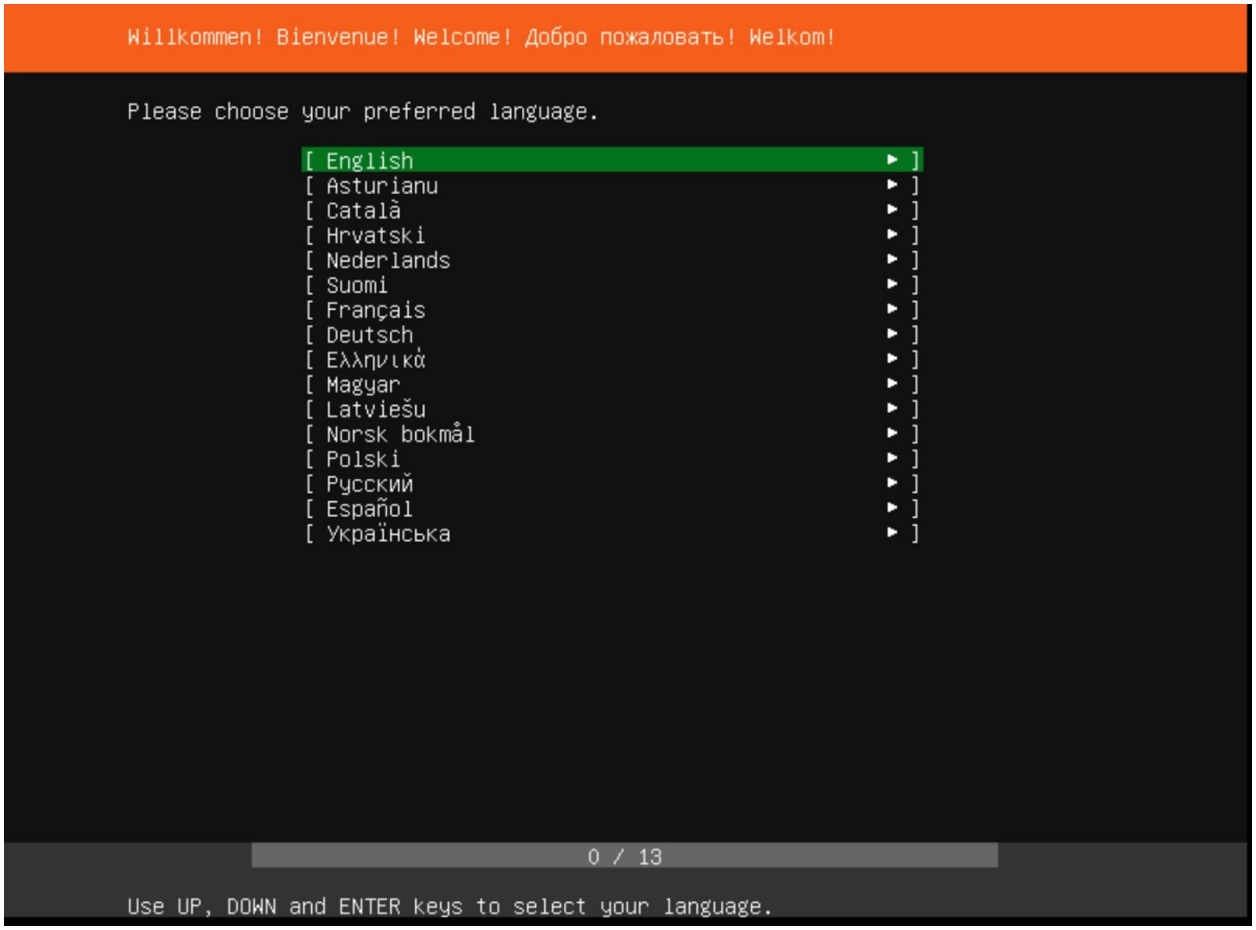
It can take a while for the next screen will be shown, depending on your system performance. Make sure the hard disk icon or hard disk activity led (when using a physical system) is flickering.
If your system does not load the next screen and no hard disk activity is present, please redownload and/or reflash/reburn your Ubuntu installation image. It can be faulty.



Step 6: After selecting the correct option in the main menu, you will be presented with the install language selection.

We will select ‘English’ as preferred language. This option is totally free of choice. You can choose whatever language you desire. Since we will only use the command-line interface in Ubuntu, the commands and command structures will be the same, the language setting will be irrelevant.

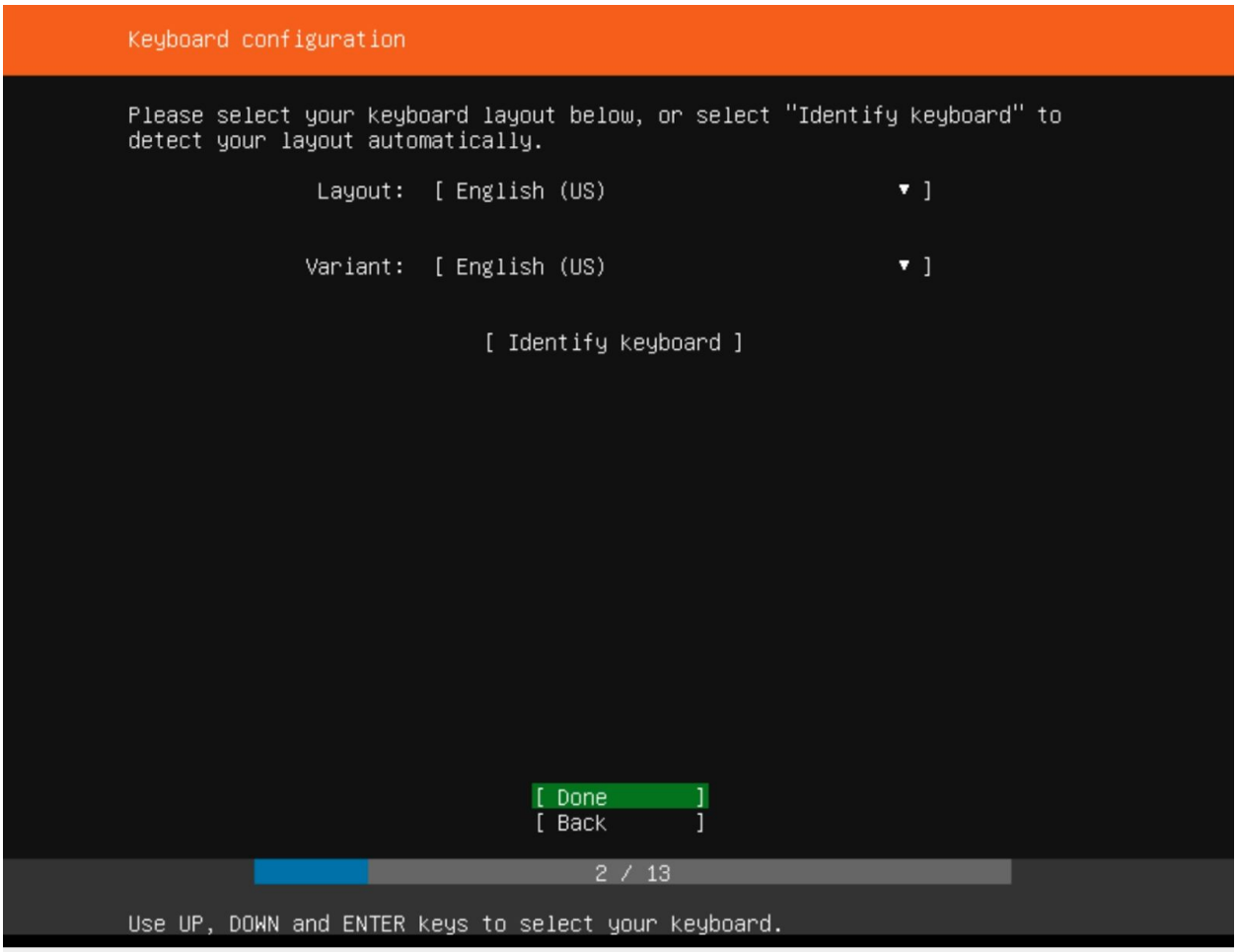
Press “**ENTER**” to continue the installation



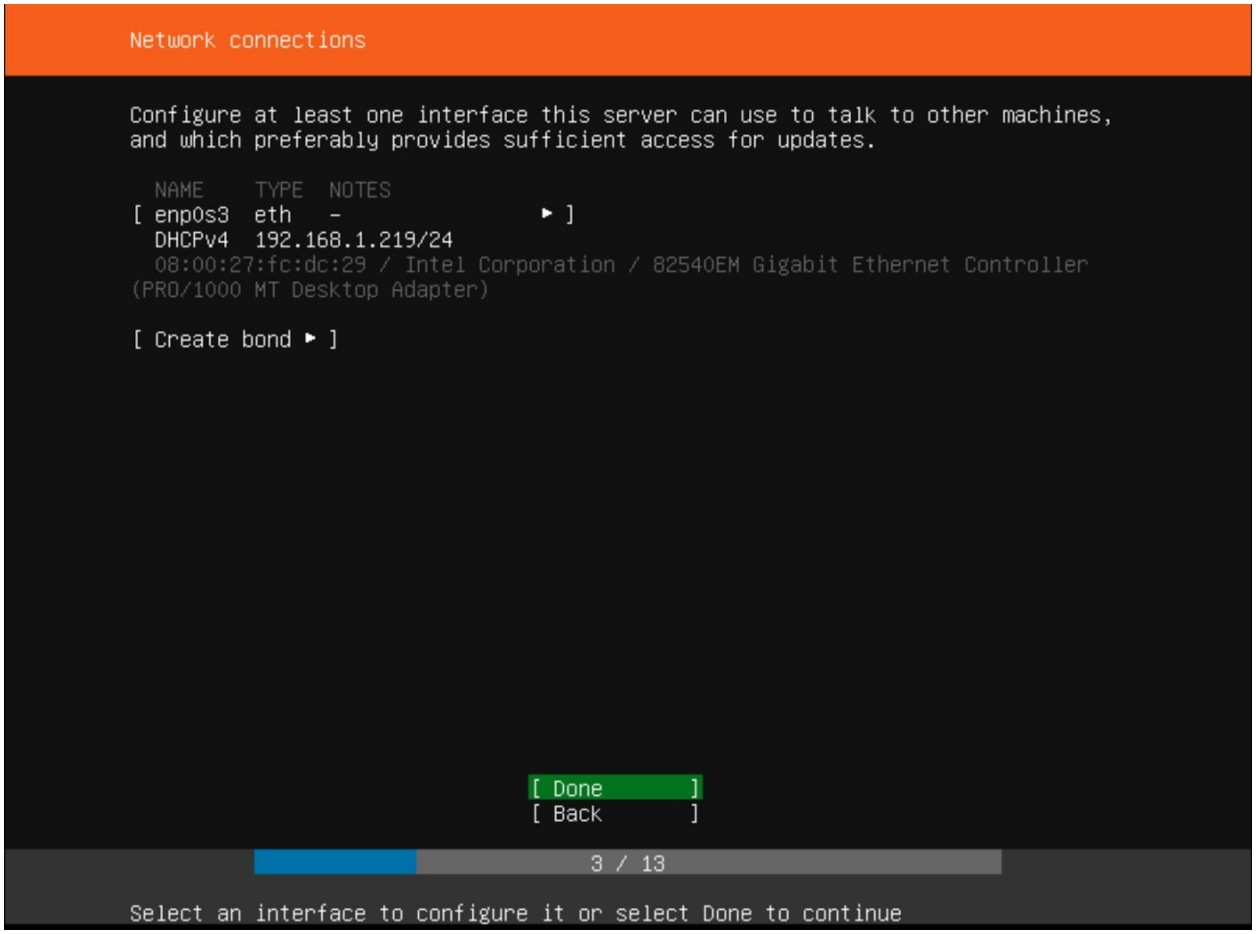
Step 7: After the install language selection is completed you'll need to set up your keyboard layout. We strongly advise to set the correct keyboard layout for your computer, which may vary depending on your country and region. Setting an incorrect keyboard layout may result in difficulties later on when executing commands in the command line interface.
Use the arrow keys on your keyboard to set the correct layout. When ready; select 'Done' and press "**ENTER**" to continue.

Tip:

If you are unsure which keyboard layout you can also let the installer determine which keyboard you have; you can use the automatic detect function by selecting "Identify Keyboard" and pressing "**ENTER**".



Step 8: After setting up the keyboard layout, you will need to set up your Network Interface Card. Usually the default settings are enough to get you going.
You can also manually set up your network interface. However you would need to know how to set this up properly.
In this manual we use the default setting which automatically grabs an IP address from the router.
Tip:
Make sure when using a virtual machine that your network interface card is set to “Bridged” mode.
Check if the network interface gets a number. If it does not or it gets you a 169. address, please check your router or/and dhcp server if its working correctly.

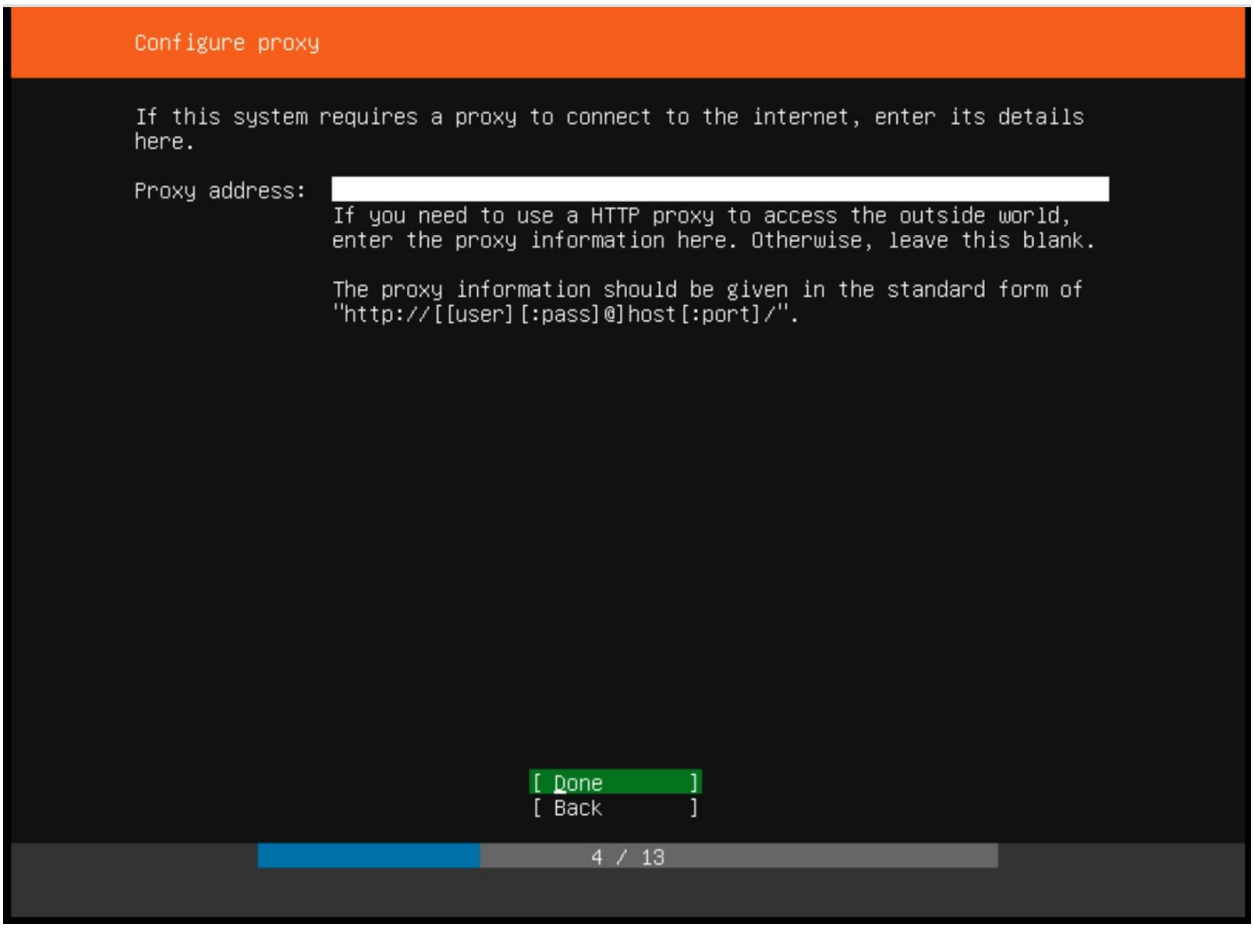


Press “**Enter**” to continue

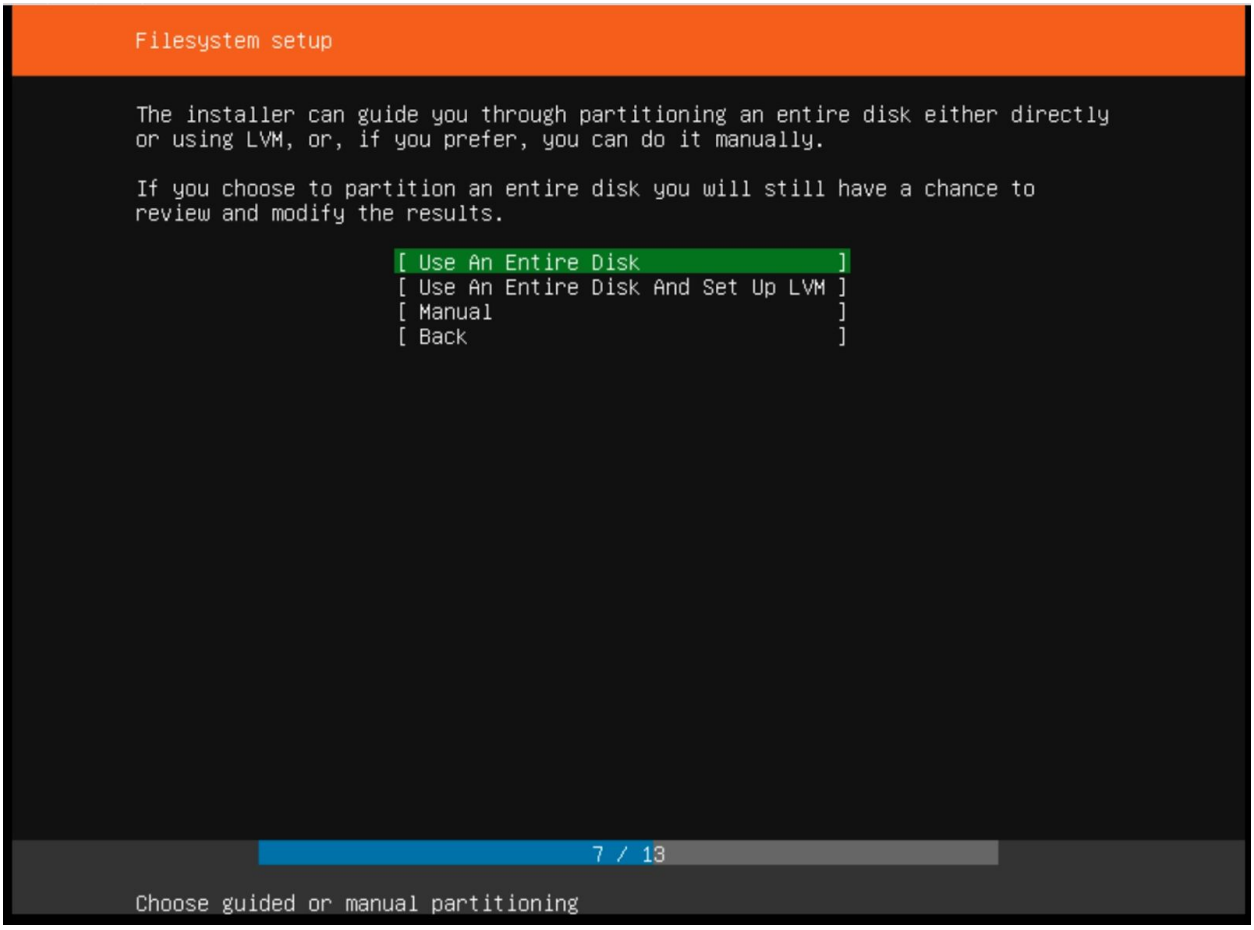
Step 9:

The default setting is normally good to go. You usually do not need to configure any proxy addresses. If you do however, you can specify it here.
For this manual we leave it blank because our network configuration does not need a proxy address.

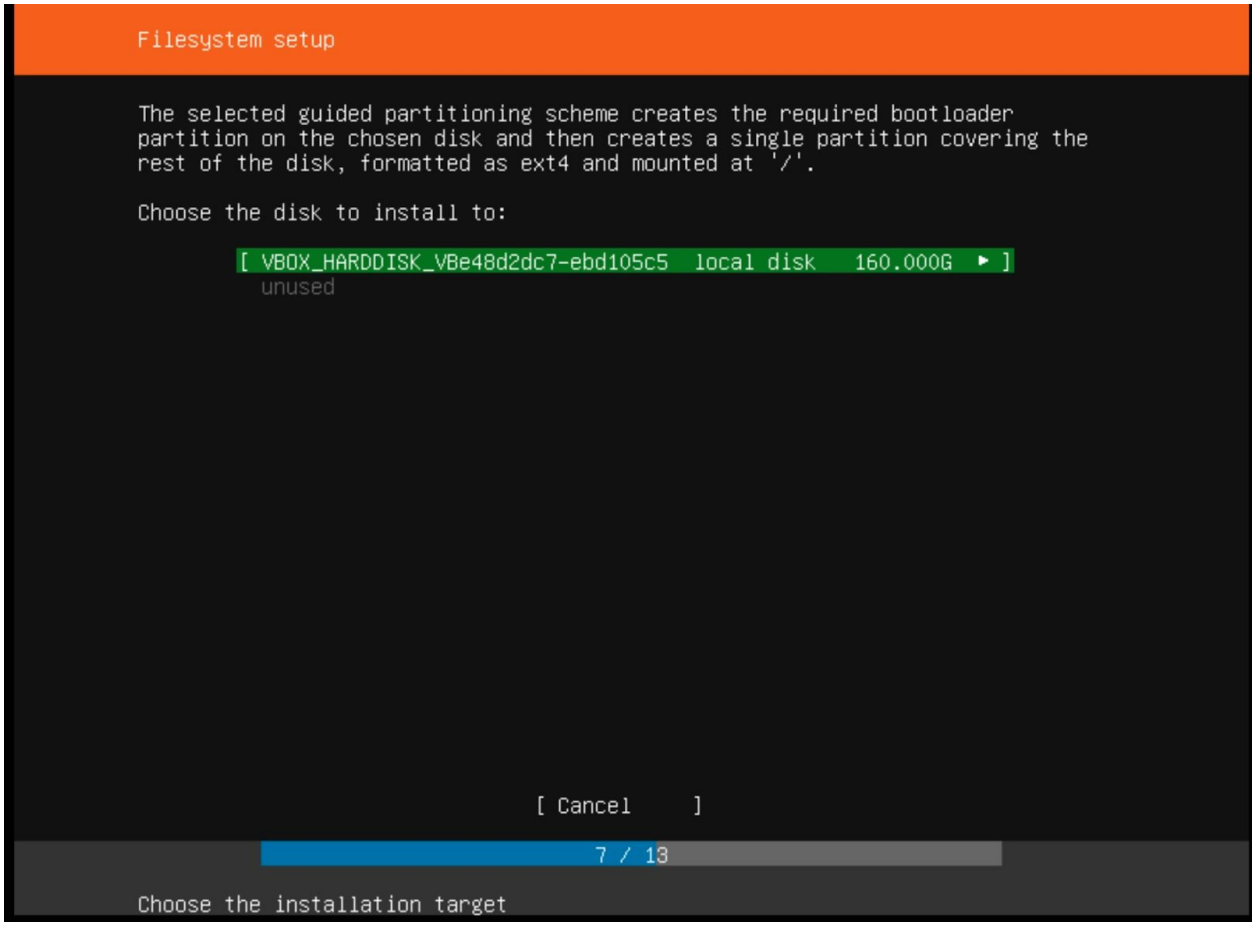
Press “**Enter**” to continue



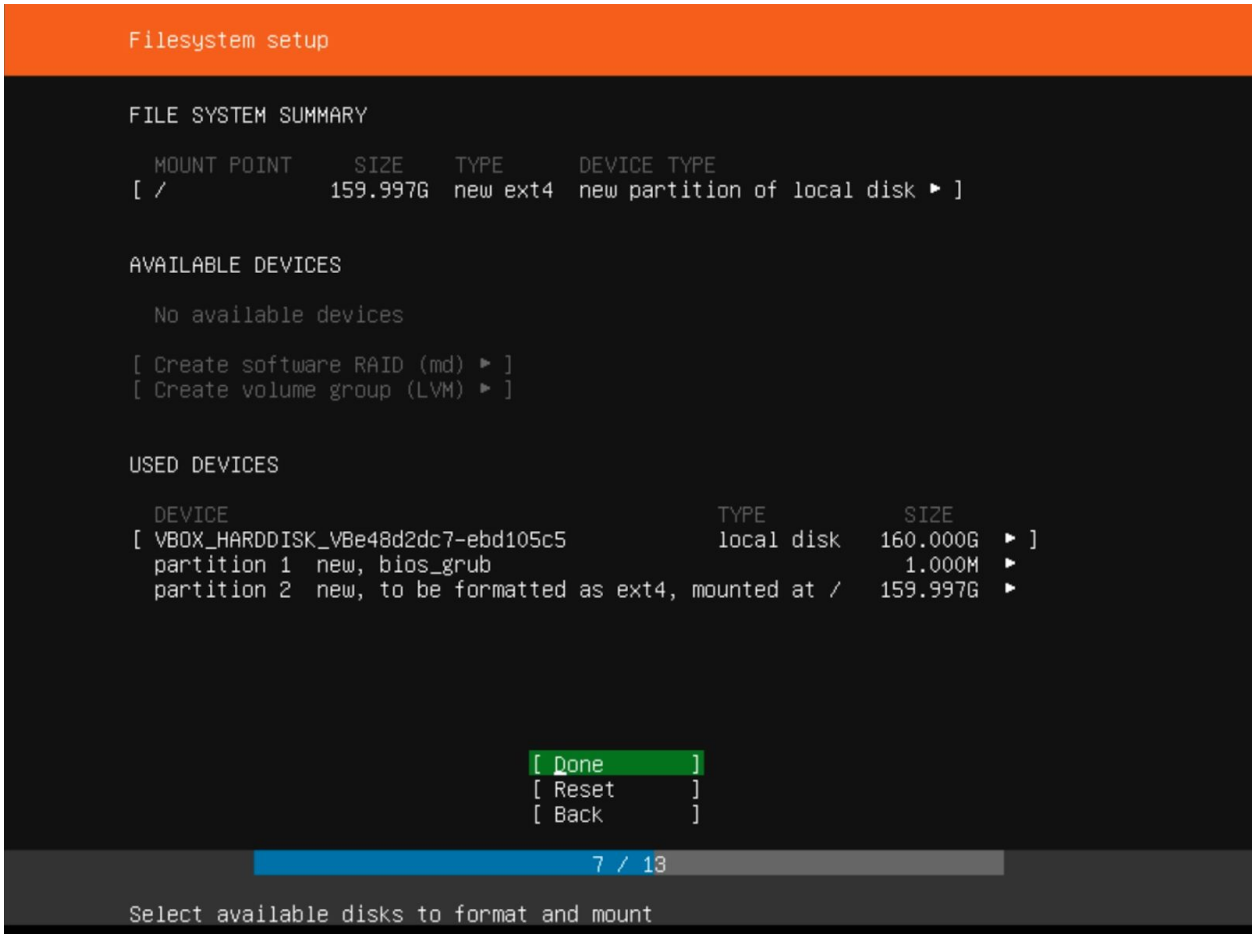
Step 10: In this step we choose to use the entire hard disk for the installation of Ubuntu server 18.04. For the next installation step click on “**Enter**”.



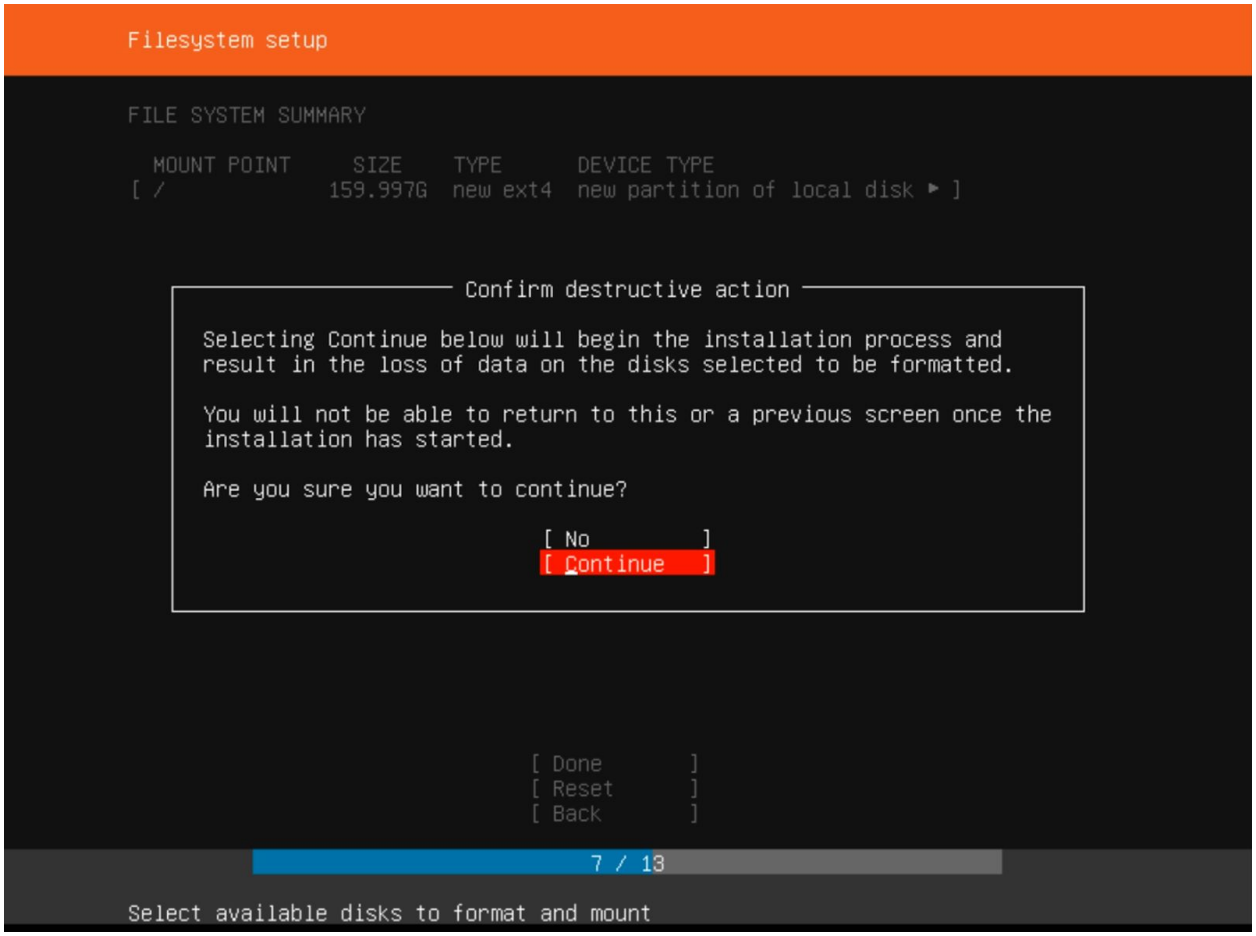
We select the only hard disk in the computer, the hard disk is need for coping the installation files in to the hard disk. For the next installation step click on “**Enter**”



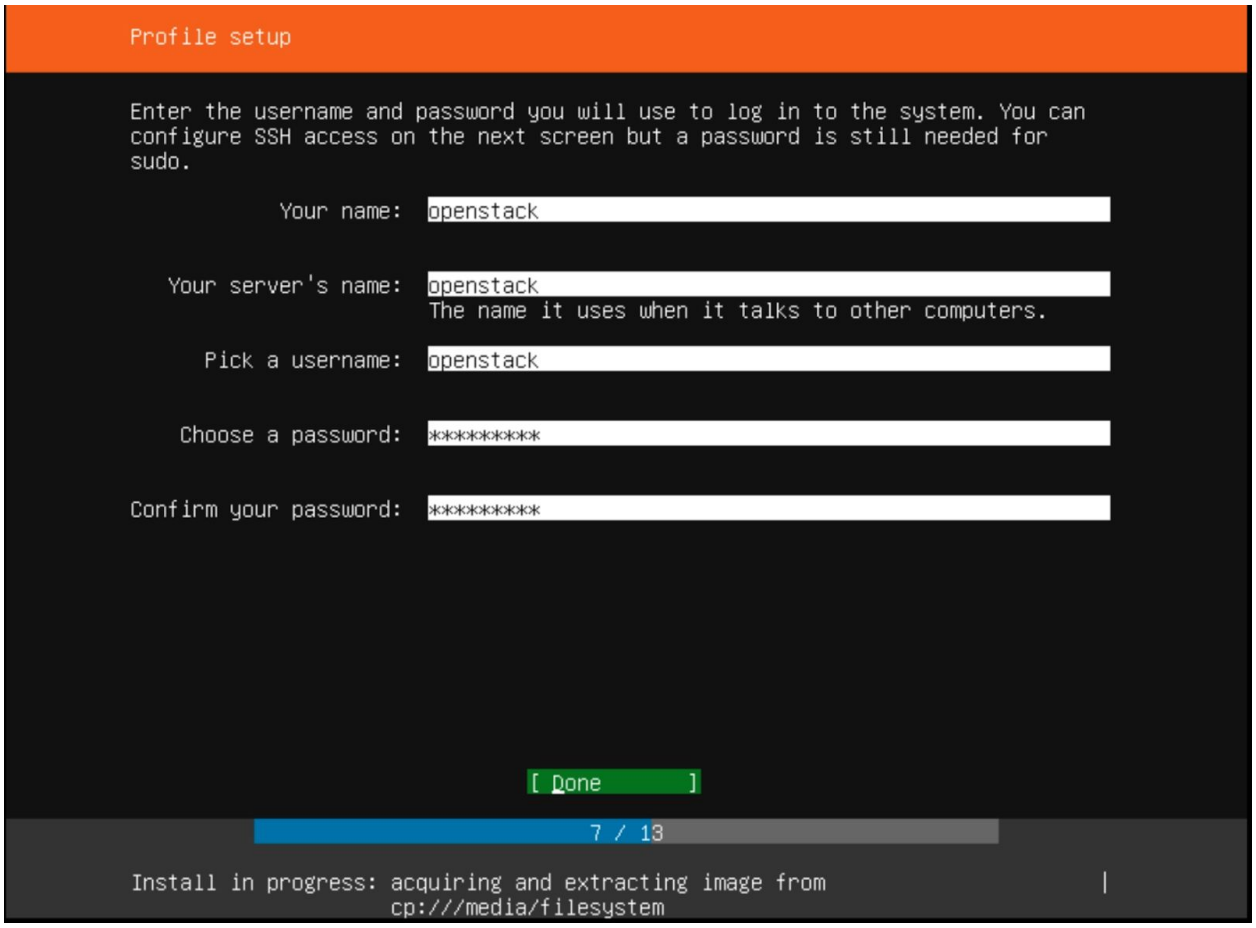
Step 11: In this step the installation of Ubuntu server gives an overview of the options we selected during the previous steps, the installation make sure that you don't make any configuration errors before continuing with the installation of Ubuntu server 18.04. For the next installation step click on “**Enter**”



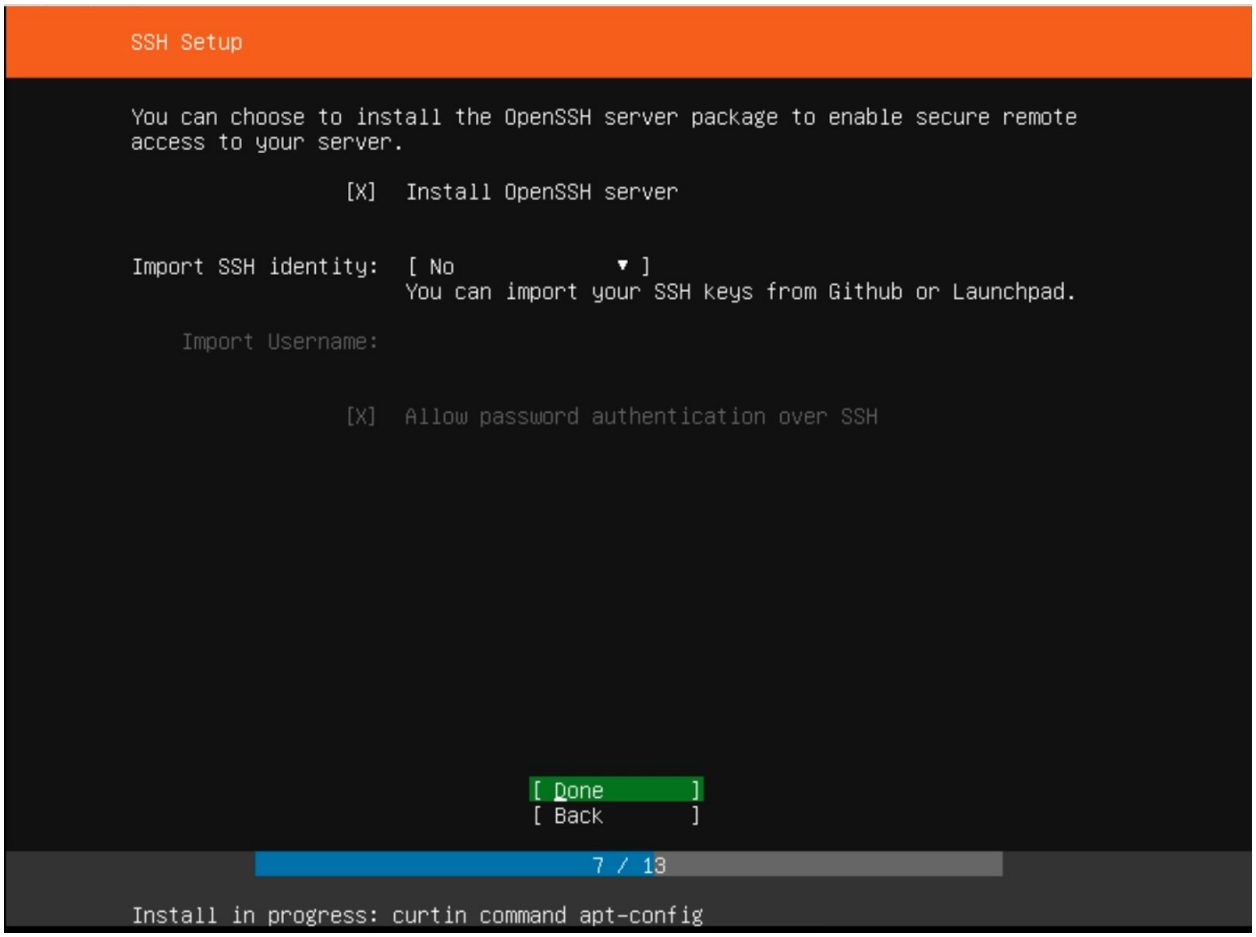
Step 12: At this step you must confirm that you want to make the configuration. Click on “**Continue**”. The settings will now be applied to your system and the installation will start. You have crossed the point of no return here. You will not be able to reconfigure the previous steps from here on.



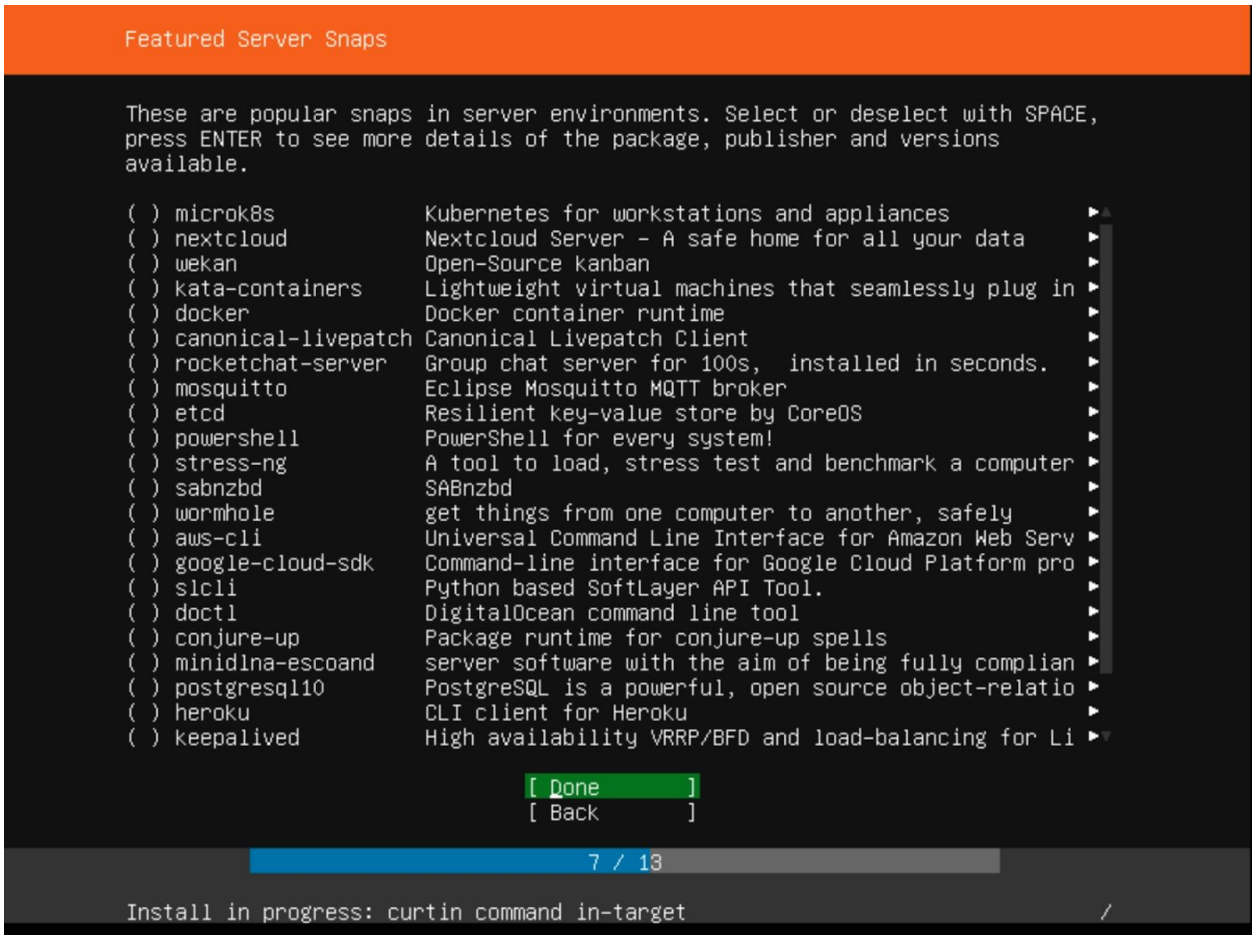
Step 13: After the main installation of Ubuntu 18.04 is complete. You will need to specify an admin user. We will use the username as “openstack” and password “openstack” in this example, however you can specify this differently if you wish to.



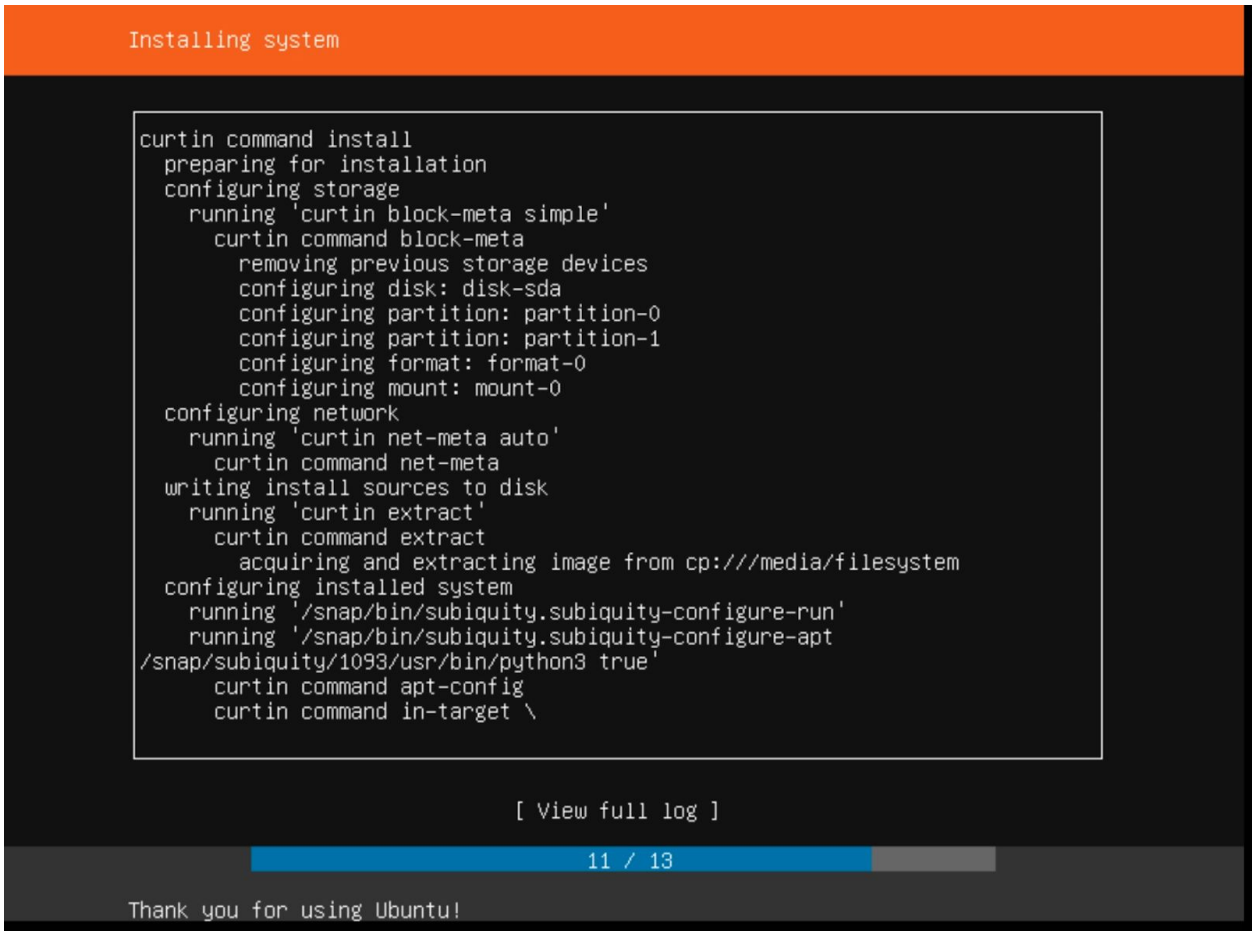
Step 14: To manage the server remotely, choose to install an SSH server on the server. This allows you to manage the server remotely. Select “**Install OpenSSH server**” at Ubuntu installer.



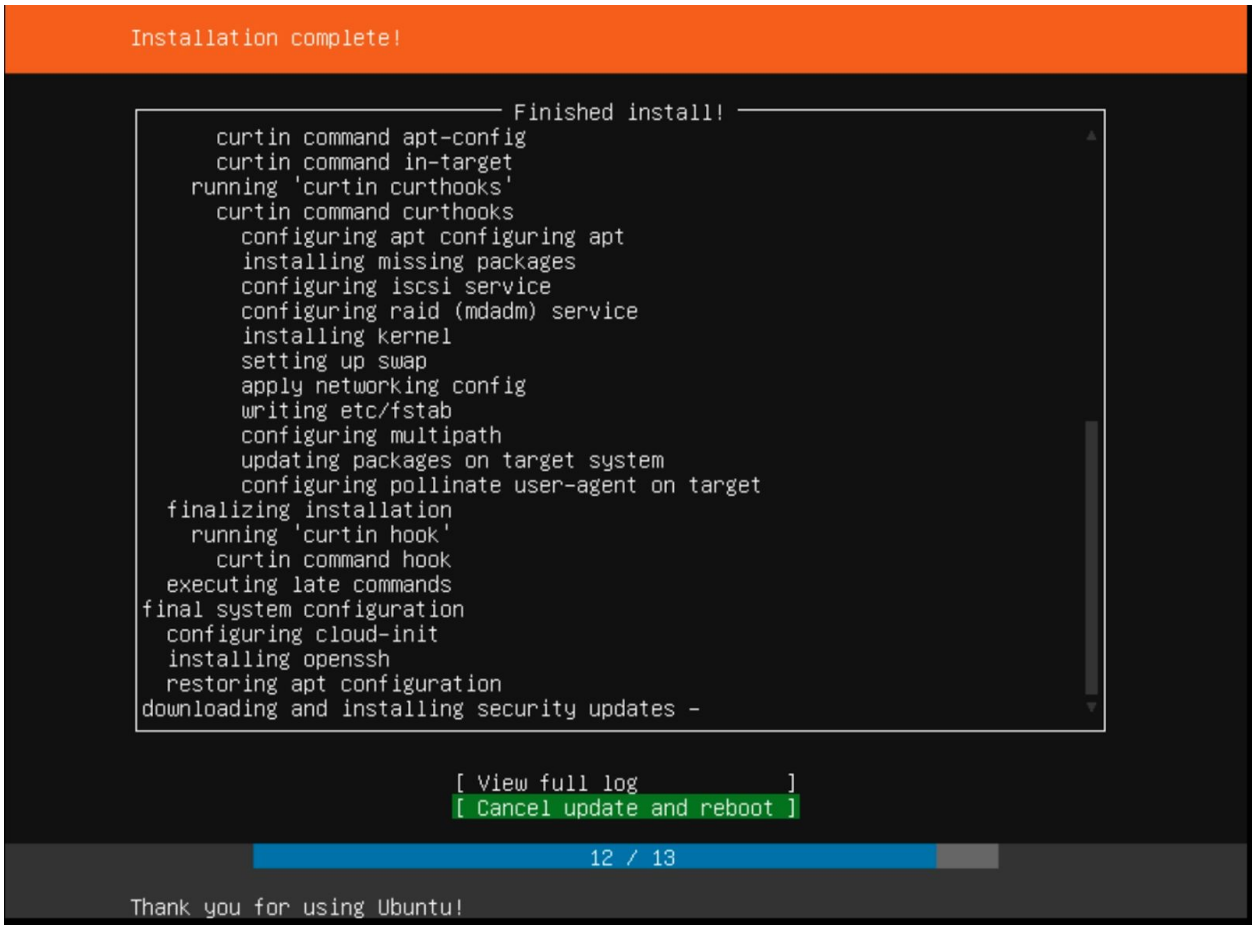
Step 15: During this step you can install software packages, we do not do this in this tutorial because we are going to use the server for specific purposes



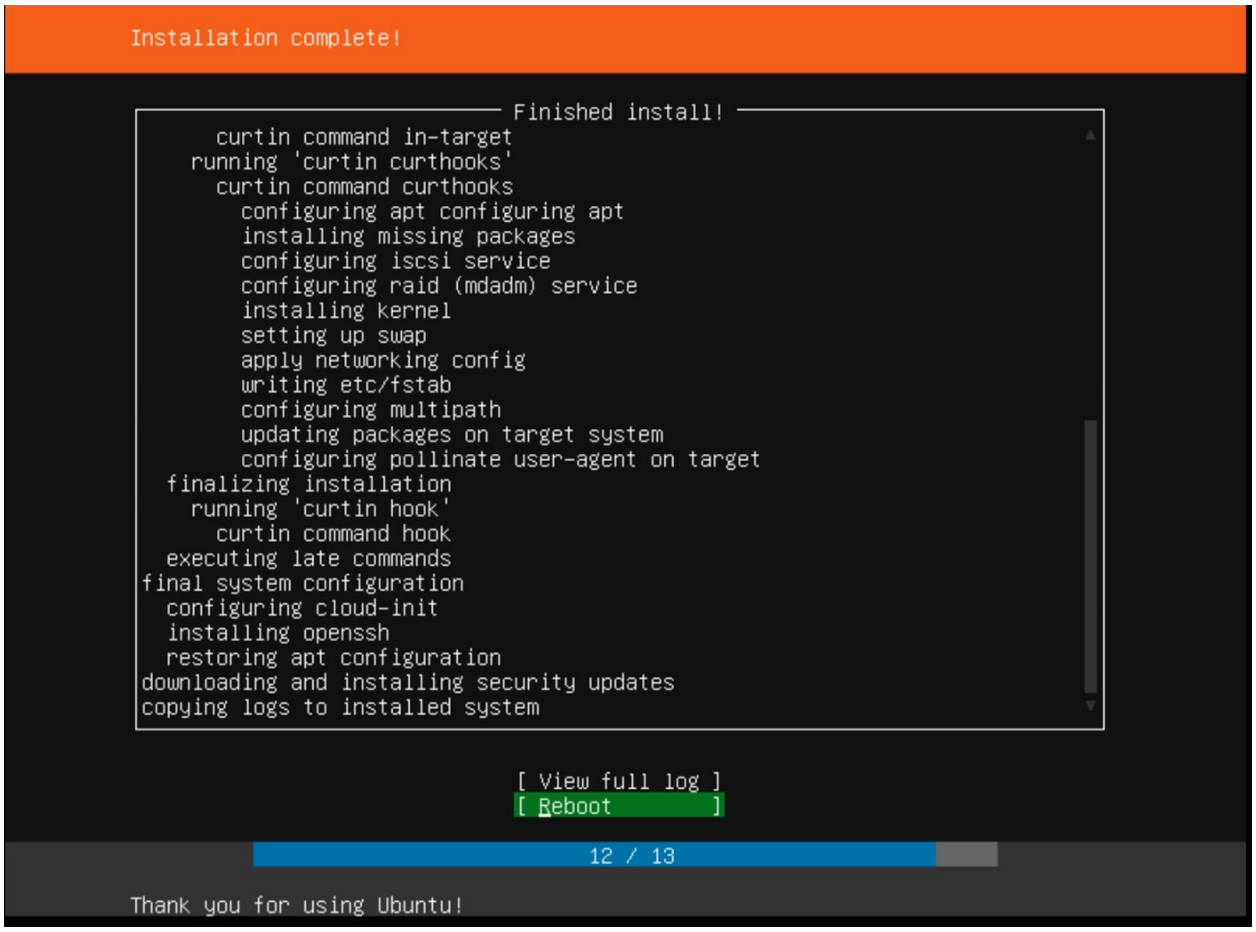
Step 16: Now you will have to wait until the system is fully installed on your server.



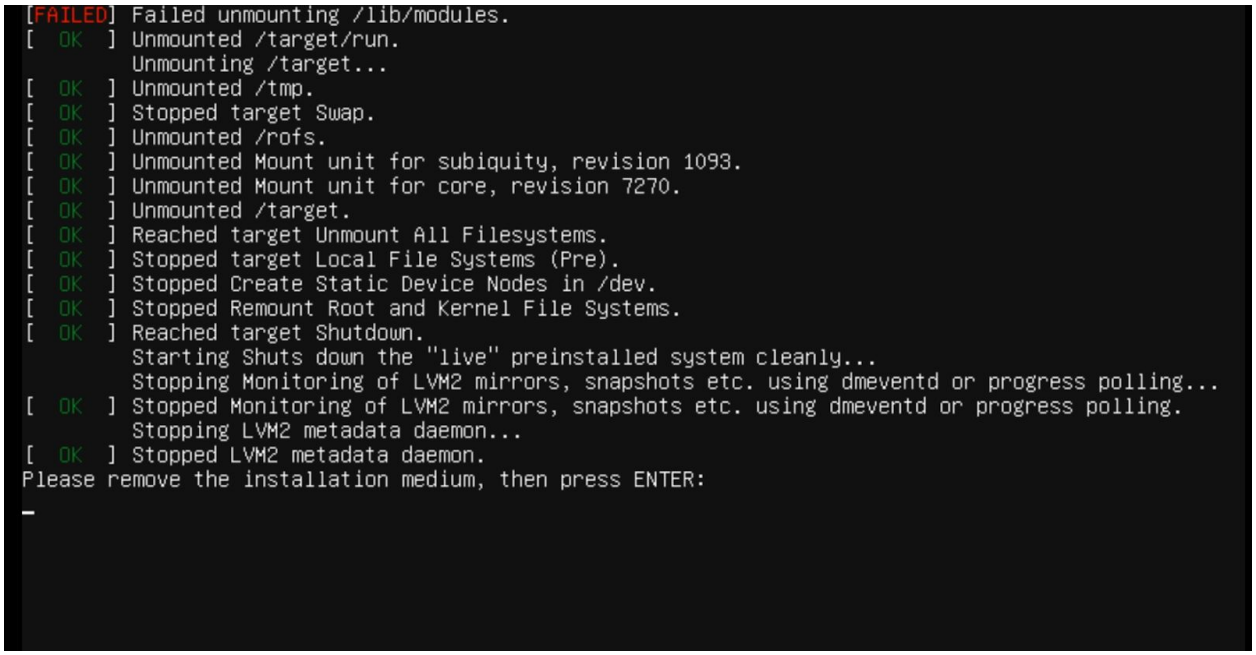
Step 17: After the installation is complete, it will automatically security updates for your system. It will reboot automatically after the installing of the update is completed.



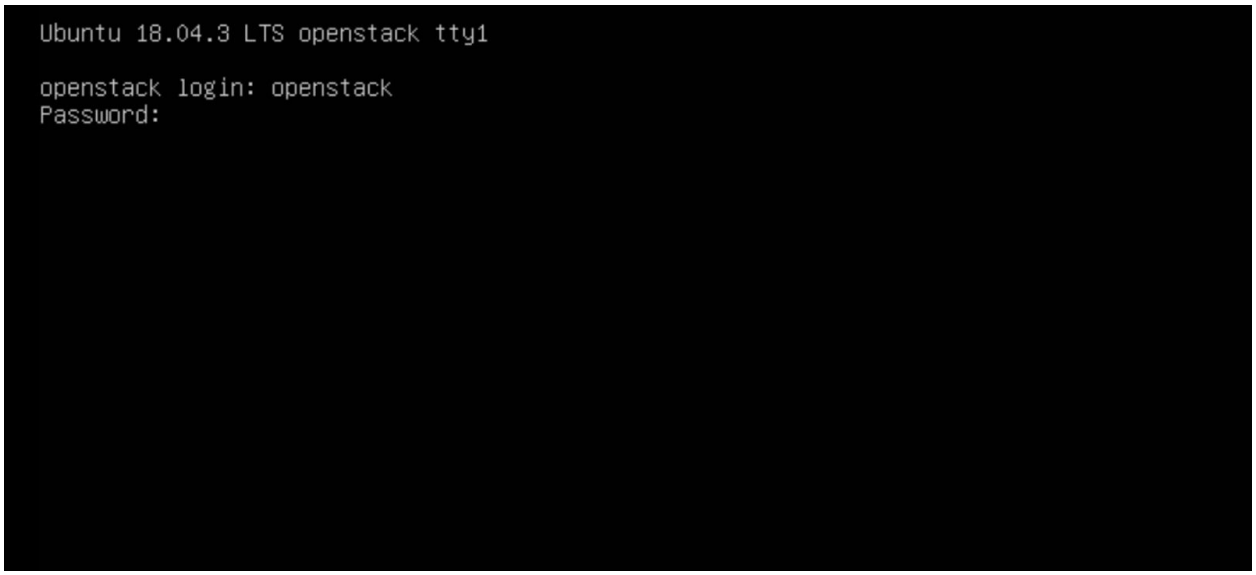
Step 18 :After the updates have been completely downloaded and installed, you can restart the server and click on "reboot".



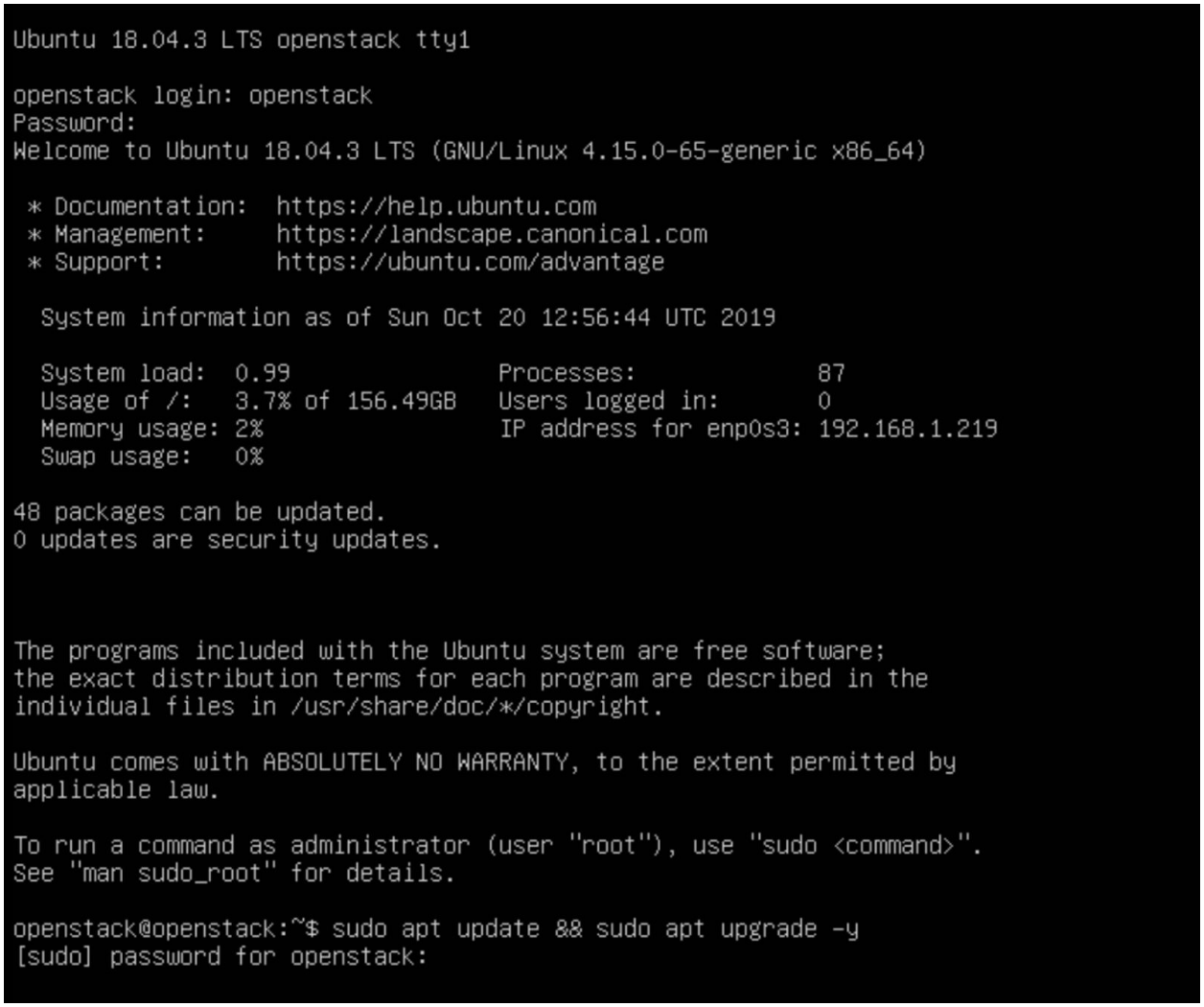
Step 19: The Ubuntu installation will ask you to remove the installation medium, after you have done this click on "Enter". For example, remove your USB, DVD or the ISO from your virtual machine



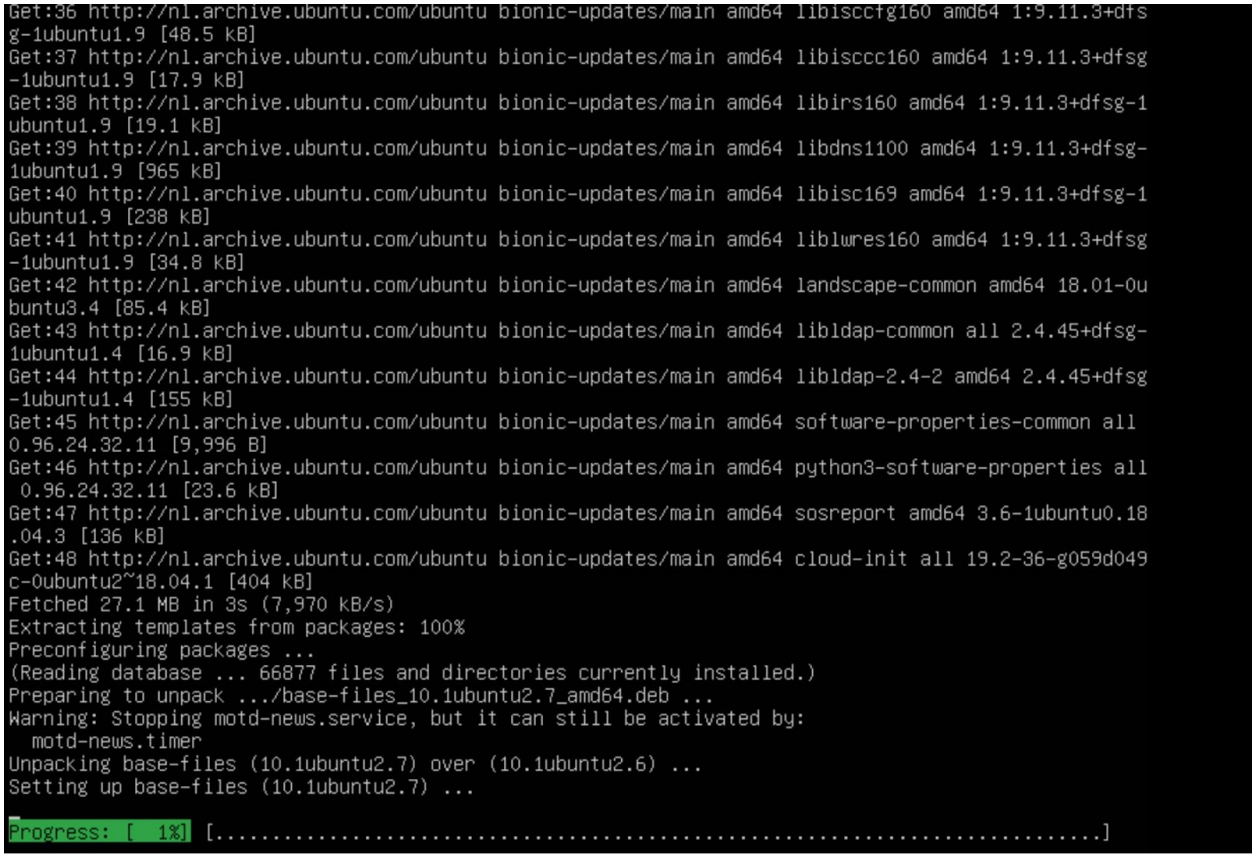
Step 20: After you have removed the installation medium from the system, your server will restart, after which you will see the login screen as shown in the image below. Login using your admin defined user and password.



Step 21: To be sure you check your server for updates. You do this with the command "**sudo update && sudo upgrade -y**". Your server will now check if new updates are available, if there are any, they will be installed automatically without you having to answer "yes" to install the updates.



Step 21: As you can see on the image below, there are still updates available for your server that will be installed. You will see a progress bar at the bottom of the screen in the color green. You will also see how far your server is with installing the updates. The update progress will start at 1% and this goes up to 100%.



Step 22: We suggest that you reboot your system when the updates are completed, this is to make sure your system saves and applies all the changes due to updates on your system. You can reboot by typing “sudo reboot”.