

[Instruction manual](#)

How to Install Openstack in Ubuntu 18.04 using Devstack



Chapter 1 - Pre installation steps Devstack	4
Chapter 2 - Installation of Devstack	11
Chapter 3 - Add images	13
Chapter 4 - Security Groups	18
Chapter 5 - SSH Keys	23
Chapter 6 - Add router	26
Chapter 7 - Create instances	32
References list	42
Appendix - Installing Ubuntu 18.04	43

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 complete 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.

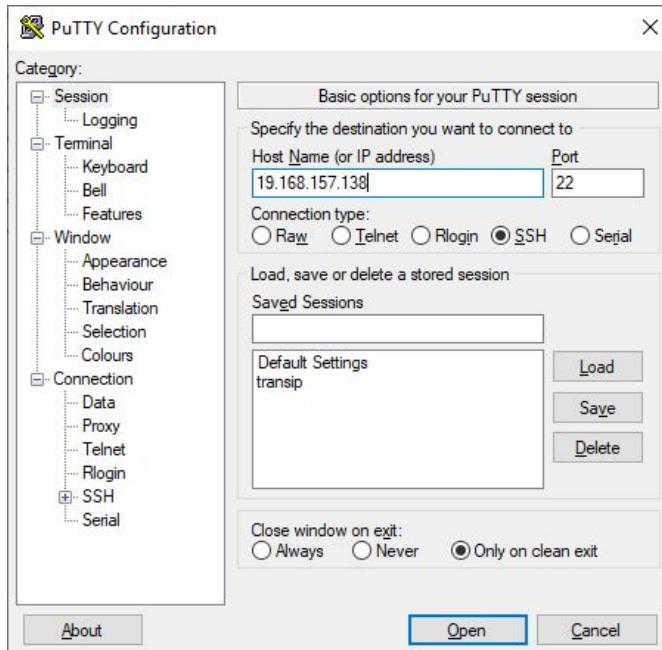
*We assume that you can access the server with SSH or manage the server in another way and that you have an admin user and password.

*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.

```
[sudo] password for openstack:  
Adding user `stack' ...  
Adding new group `stack' (1001) ...  
Adding new user `stack' (1001) with group `stack' ...  
Creating home directory `/home/stack' ...  
Copying files from `/etc/skel' ...  
Enter new UNIX password:  
Retype new UNIX password:  
passwd: password updated successfully  
Changing the user information for stack  
Enter the new value, or press ENTER for the default  
    Full Name []:  
    Room Number []:  
    Work Phone []:  
    Home Phone []:  
    Other []:  
Is the information correct? [Y/n] y  
openstack@openstack:~$
```

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.**

```
openstack@openstack:~$ sudo -i  
root@openstack:~# 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”.

```
root@openstack:~# cd ~  
root@openstack:~# su stack  
stack@openstack:/root$ cd ~  
stack@openstack:~$ pwd  
/home/stack  
stack@openstack:~$
```

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

```
stack@openstack:~$ sudo apt install git
Reading package lists... Done
Building dependency tree...
Reading state information... Done
git is already the newest version (1:2.17.1-1ubuntu0.4).
git set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
stack@openstack:~$
```

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

```
stack@openstack:~$ git clone https://git.openstack.org/openstack-dev/devstack
Cloning into 'devstack'...
warning: redirecting to https://opendev.org/openstack/devstack/
remote: Enumerating objects: 44358, done.
remote: Counting objects: 100% (44358/44358), done.
remote: Compressing objects: 100% (13344/13344), done.
remote: Total 44358 (delta 31675), reused 42564 (delta 30329)
Receiving objects: 100% (44358/44358), 8.40 MiB | 3.45 MiB/s, done.
Resolving deltas: 100% (31675/31675), done.
stack@openstack:~$
```

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.

```
stack@openstack:~/devstack$ cd devstack/
stack@openstack:~/devstack$ ls
clean.sh  doc      files    functions-common  gate      inc      LICENSE      Makefile   playbooks  roles      samples  setup.py  stack.sh  tools      unstack.sh
data      extras.d  functions  FUTURE.rst     HACKING.rst lib      MAINTAINERS.rst  openrc   README.rst run_tests.sh  setup.cfg  stackrc  tests      tox.ini
stack@openstack:~/devstack$ ls -al samples/
total 16
drwxrwxr-x  2 stack stack 4096 Oct 23 16:10 .
drwxrwxr-x 15 stack stack 4096 Oct 23 16:10 ..
-rw-rw-r--  1 stack stack 3912 Oct 23 16:10 local.conf
-rw-rw-r--  1 stack stack 1967 Oct 23 16:10 local.sh
stack@openstack:~/devstack$ sudo cp samples/local.conf ./local.conf
stack@openstack:~/devstack$
```

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.

```
stack@openstack:~/devstack$ ip a | grep enp0s3
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    inet 192.168.1.219/24 brd 192.168.1.255 scope global enp0s3
stack@openstack:~/devstack$
```

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.

```
stack@openstack:~/devstack$ ./stack.sh
```

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.

```
+ stack.sh:main:509    grep -qwe openstack /etc/hosts
++ ./stack.sh:main:396    truefalse False SKIP_EPEL_INSTALL
+functions-common:treeorfalse:150    local xtrace
++functions-common:treeorfalse:151    set +o
++functions-common:treeorfalse:151    grep xtrace
++functions-common:treeorfalse:151    xtrace='set -o xtrace'
++functions-common:treeorfalse:152    set +o xtrace
++functions-common:treeorfalse:152    SKIP_EPEL_INSTALL=False
+./stack.sh:main:399    [[ bionic == '\rhel\7' ]]
+./stack.sh:main:399    is_package_installed python
+./stack.sh:main:418    +Functions-common:is_package_installed:1343    [[ -z python ]]
+functions-common:is_package_installed:1347    [[ -z deb ]]
+functions-common:is_package_installed:1351    [[ deb = '\de\b' ]]
+functions-common:is_package_installed:1352    dpkg -s python
+./stack.sh:main:418    install_package python
+functions-common:install_package:1334    update_package_repo
+functions-common:update_package_repo:1306    NO_UPDATE_REPO=False
+functions-common:update_package_repo:1307    REPOS_UPDATED=False
+functions-common:update_package_repo:1308    RETRY_UPDATE=False
+functions-common:update_package_repo:1310    [[ False == '\!\!\!e' ]]
+functions-common:update_package_repo:1314    is_ubuntu
+functions-common:is_ubuntu:494    [[ -z deb ]]
+functions-common:is_ubuntu:497    [[ deb = deb ']]
+functions-common:update_package_repo:1315    apt_get_update
+functions-common:apt_get_update:1069    [[ False == '\!\!\!e' ]]
+functions-common:apt_get_update:1074    [[ False == '\!\!\!e' ]]
+functions-common:apt_get_update:1076    local sudo=sudo
++functions-common:apt_get_update:1077    id -u
+functions-common:apt_get_update:1077    [[ 1001 == '\0' ]]
+functions-common:apt_get_update:1080    time_start apt-get-update
+functions-common:time_start:2322    local name=apt-get-update
+functions-common:time_start:2323    local start_time=
[[ -n '' ]]
+functions-common:time_start:2324    date +%sX3N
+functions-common:time_start:2327    TIME_START[$name]=`date +%sX3N`
+functions-common:apt_get_update:1082    local 'proxies=http_proxy=https_proxy=no_proxy='
+functions-common:apt_get_update:1083    local 'update_cmd=sudo http_proxy=https_proxy=no_proxy= apt-get update'
+functions-common:apt_get_update:1084    timeout 300 sh -c 'while ! sudo http_proxy=https_proxy=no_proxy= apt-get update; do sleep 30; done'
Http:1 http://nl.archive.ubuntu.com/ubuntu bionic InRelease
Http:2 http://nl.archive.ubuntu.com/ubuntu bionic-updates InRelease
Http:3 http://nl.archive.ubuntu.com/ubuntu bionic-backports InRelease
Http:4 http://nl.archive.ubuntu.com/ubuntu bionic-security InRelease
Reading package lists... 7%
```

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.

```
=====
DevStack Component Timing
(times are in seconds)
=====
run_process      53
test_with_retry   3
apt-get-update    2
osc               160
wait_for_service  26
git_timed        268
dbsync            123
pip_install       421
apt-get           467
-----
Unaccounted time 674
=====
Total runtime     2197

This is your host IP address: 192.168.1.219
This is your host IPv6 address: ::1
Horizon is now available at http://192.168.1.219/dashboard
Keystone is serving at http://192.168.1.219/identity/
The default users are: admin and demo
The password: openstack

WARNING:
Using lib/neutron-legacy is deprecated, and it will be removed in the future

Services are running under systemd unit files.
For more information see:
https://docs.openstack.org/devstack/latest/systemd.html

DevStack Version: ussur
Change: b14665f0dde0d0862d8755a796b9f680e42f790b Revert "Remove deprecated PostgreSQL database driver" 2019-10-17 15:58:34 -0400
OS Version: Ubuntu 18.04 bionic

stack@openstack:~/devstack$
```

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.

```
stack@openstack:~/devstack$ ls
acrcr  doc  functions  gate  lib  MAINTAINERS.rst  playbooks  run_tests.sh  setup.py  tests  unstack.sh  wget-log.1  wget-log.4
clean.sh  extras.d  functions-common  HACKING.rst  LICENSE  Makefile  README.rst  samples  stackrc  tools  userrc_early  wget-log.2  wget-log.5
data  files  FUTURE.rst  inc  local.conf  openrc  roles  setup.cfg  stack.sh  tox.ini  wget-log  wget-log.3  wget-log.6
stack@openstack:~/devstack$ source openrc admin admin
WARNING: setting legacy OS_TENANT_NAME to support cli tools.
stack@openstack:~/devstack$
```

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”**.

```
stack@openstack:~/devstack/files$ wget https://download.fedoraproject.org/pub/fedora/linux/releases/30/Cloud/x86_64/images/Fedora-Cloud-Base-30-1.2.x86_64.qcow2
2019-10-23 17:25:35.453555  https://download.fedoraproject.org/pub/fedora/linux/releases/30/Cloud/x86_64/images/Fedora-Cloud-Base-30-1.2.x86_64.qcow2
Resolving download.fedoraproject.org (download.fedoraproject.org)... 140.211.169.206, 85.236.55.6, 67.219.144.68, ...
Connecting to download.fedoraproject.org (download.fedoraproject.org)|140.211.169.206|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: http://ftp.nluug.nl/pub/os/Linux/distr/fedora/linux/releases/30/Cloud/x86_64/images/Fedora-Cloud-Base-30-1.2.x86_64.qcow2 [following]
--2019-10-23 17:25:35.453555  https://ftp.nluug.nl/pub/os/Linux/distr/fedora/linux/releases/30/Cloud/x86_64/images/Fedora-Cloud-Base-30-1.2.x86_64.qcow2
Resolving ftp.nluug.nl (ftp.nluug.nl)|145.228.21.40|:2080... connected.
Connecting to ftp.nluug.nl (ftp.nluug.nl)|145.228.21.40|:2080... connected.
HTTP request sent, awaiting response... 200 OK
Length: 332267520 (317M) [text/plain]
Saving to: 'Fedora-Cloud-Base-30-1.2.x86_64.qcow2'

Fedora-Cloud-Base-30-1.2.x86_64.qcow2      100%[=====] 316.88M 28.8MB/s   in 19s
2019-10-23 17:25:35 (16.9 MB/s) - 'Fedora-Cloud-Base-30-1.2.x86_64.qcow2' saved [332267520/332267520]
stack@openstack:~/devstack/files$ 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.

```
stack@openstack:~/devstack/files$ openstack image create "Fedora" --file Fedora-Cloud-Base-30.1.2.x86_64.qcow2 --disk-format qcow2 --container-format bare --public
+---+
| Field      | Value
+---+
| checksum   | ffa3dd42fae5590cd0fe72d429bc677b
| container_format | bare
| created_at | 2019-10-23T17:26:00Z
| disk_format | qcow2
| file       | /v2/images/d5649442-f8ef-4bc8-8af8-ee853dab5f00/file
| id         | d5649442-f8ef-4bc8-8af8-ee853dab5f00
| min_disk   | 0
| min_ram   | 0
| name       | Fedora
| owner      | 90dsea334454318862106b95a13c48
| properties_se' | os_hash_algo='sha512', os_hash_value='d9f99d22a0b0ea1e8b93379dd2080f51a7ed6885aa7d4c2f2262ea1054935e02c47b45f9b56aa7f55e61d149d06f4ff6de03efde24f9d6774baf35f08c5e9d92', os_hidden='Fa
| protected  | False
| schema     | /v2/schemas/image
| size       | 332267520
| status     | active
| tags       |
| updated_at | 2019-10-23T17:26:03Z
| virtual_size | None
| visibility | public
+---+
```

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.

Type	Status	Visibility	Protected	Disk Format	Size
Image	Active	Shared	No	QCOW2	1.84 GB
Image	Active	Shared	No	QCOW2	898.75 MB
Image	Active	Shared	No	QCOW2	12.13 MB
Image	Active	Shared	No	QCOW2	1017.56 MB
Image	Active	Shared	No	QCOW2	316.88 MB
Image	Active	Shared	No	QCOW2	328.25 MB

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.

The screenshot shows the Openstack Network interface. In the top navigation bar, 'Project' is selected. Under 'Compute', 'Volumes', and 'Network' (which is expanded), there are sub-options: 'Network Topology', 'Networks', and 'Routers'. The 'Security Groups' option under 'Network' is highlighted with a blue background. On the right side, there is a search bar with 'Filter' and a magnifying glass icon, a red button '+ Create Security Group', and a red button 'Delete Security Groups'. Below the navigation, the title 'Security Groups' is displayed. A table lists one item: 'default' (Security Group ID: 62507e22-0c26-4a47-9d82-911e107e1abc). The table has columns for 'Name', 'Security Group ID', 'Description' (Default security group), and 'Actions' (Manage Rules). At the bottom of the table, it says 'Displaying 1 item'.

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

The screenshot shows the 'Manage Rules' window for the 'Default Security Group'. At the top, there is a red button '+ Add Rule' and a red button 'Delete Rules'. Below this, a table displays four items. The table has columns: 'Direction', 'Ether Type', 'IP Protocol', 'Port Range', 'Remote IP Prefix', 'Remote Security Group', 'Description', and 'Actions'. The rows are: 1. Egress, IPv4, Any, Any, 0.0.0.0/0, -, -. 2. Egress, IPv6, Any, Any, ::/0, -, -. 3. Ingress, IPv4, Any, Any, -, default, -. 4. Ingress, IPv6, Any, Any, -, default, -. At the bottom of the table, it says 'Displaying 4 items'.

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.

Add Rule



Rule *

Custom TCP Rule

Custom TCP Rule
Custom UDP Rule
Custom ICMP Rule
Other Protocol

All ICMP



All TCP

All UDP

DNS

HTTP

HTTPS

IMAP

IMAPS

LDAP

MS SQL

MYSQL

POP3

POP3S

RDP

SMTP

SMTPS

SSH

Description:

Rules define which traffic is allowed to instances assigned to the security group. A security group rule consists of three main parts:

Rule: You can specify the desired rule template or use custom rules, the options are Custom TCP Rule, Custom UDP Rule, or Custom ICMP Rule.

Open Port/Port Range: For TCP and UDP rules you may choose to open either a single port or a range of ports. Selecting the "Port Range" option will provide you with space to provide both the starting and ending ports for the range. For ICMP rules you instead specify an ICMP type and code in the spaces provided.

Remote: You must specify the source of the traffic to be allowed via this rule. You may do so either in the form of an IP address block (CIDR) or via a source group (Security Group). Selecting a security group as the source will allow any other instance in that security group access to any other instance via this rule.

Cancel

Add

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

Add Rule

Rule *

All ICMP

Description ?

Allow ping to instances

Direction

Ingress

Remote * ?

CIDR

CIDR * ?

0.0.0.0/0

Description:

Rules define which traffic is allowed to instances assigned to the security group. A security group rule consists of three main parts:

Rule: You can specify the desired rule template or use custom rules, the options are Custom TCP Rule, Custom UDP Rule, or Custom ICMP Rule.

Open Port/Port Range: For TCP and UDP rules you may choose to open either a single port or a range of ports. Selecting the "Port Range" option will provide you with space to provide both the starting and ending ports for the range. For ICMP rules you instead specify an ICMP type and code in the spaces provided.

Remote: You must specify the source of the traffic to be allowed via this rule. You may do so either in the form of an IP address block (CIDR) or via a source group (Security Group). Selecting a security group as the source will allow any other instance in that security group access to any other instance via this rule.

Cancel

Add 

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

Direction	Ether Type	IP Protocol	Port Range	Remote IP Prefix	Remote Security Group	Description	Actions
Egress	IPv4	Any	Any	0.0.0.0/0	-	-	<button>Delete Rule</button>
Egress	IPv6	Any	Any	::/0	-	-	<button>Delete Rule</button>
Ingress	IPv4	Any	Any	-	default	-	<button>Delete Rule</button>
Ingress	IPv6	Any	Any	-	default	-	<button>Delete Rule</button>

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

Add Rule

Rule *

Custom TCP Rule

- Custom TCP Rule
- Custom UDP Rule
- Custom ICMP Rule
- Other Protocol
- All ICMP
- All TCP
- All UDP
- DNS
- HTTP
- HTTPS
- IMAP
- IMAPS
- LDAP
- MS SQL
- MYSQL
- POP3
- POP3S
- RDP
- SMTP
- SMTPS
- SSH

Description:

Rules define which traffic is allowed to instances assigned to the security group. A security group rule consists of three main parts:

Rule: You can specify the desired rule template or use custom rules, the options are Custom TCP Rule, Custom UDP Rule, or Custom ICMP Rule.

Open Port/Port Range: For TCP and UDP rules you may choose to open either a single port or a range of ports. Selecting the "Port Range" option will provide you with space to provide both the starting and ending ports for the range. For ICMP rules you instead specify an ICMP type and code in the spaces provided.

Remote: You must specify the source of the traffic to be allowed via this rule. You may do so either in the form of an IP address block (CIDR) or via a source group (Security Group). Selecting a security group as the source will allow any other instance in that security group access to any other instance via this rule.

Cancel

Add

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

Add Rule

Rule *

SSH

Description ?

Allow SSH to instances

Remote * ?

CIDR

CIDR* ?

0.0.0.0/0

Description:

Rules define which traffic is allowed to instances assigned to the security group. A security group rule consists of three main parts:

Rule: You can specify the desired rule template or use custom rules, the options are Custom TCP Rule, Custom UDP Rule, or Custom ICMP Rule.

Open Port/Port Range: For TCP and UDP rules you may choose to open either a single port or a range of ports. Selecting the “Port Range” option will provide you with space to provide both the starting and ending ports for the range. For ICMP rules you instead specify an ICMP type and code in the spaces provided.

Remote: You must specify the source of the traffic to be allowed via this rule. You may do so either in the form of an IP address block (CIDR) or via a source group (Security Group). Selecting a security group as the source will allow any other instance in that security group access to any other instance via this rule.

Cancel

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+Z5CxXxE0hzF390oq7icAw+L3XAQ8w stack@openstack
The key's randomart image is:
+---[RSA 2048]----+
|   o .o
|   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!

```
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's fingerprint is:
SHA256:VDGn7xMk372a+25CxXxE0hzF390oq7lcAw+L3XAQ8w stack@openstack
The key's randomart image is:
+---[RSA 2048]---+
|          .0.   |
|         .E. .   |
|        +o . .o   |
|       .*.... o o   |
|      . S+=.. . =   |
|     . +=.o. .   |
|    =*+.o. .   |
|   .+o.oo. .   |
+---[SHA256]---+
stack@openstack:~$ cat /home/stack/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAQADQABAAQD054g1esy9PZ905bhv3XZC7M8RpPhsnw1meljsd4Ujs4Un5/l/qcDBCxdyGebtqGxqgesZoMACKzAPzu0Eulo5Ujc60wn85DhzuQqGvlpvsjVQX+fvm4IzquUy50RrbXVsyzSLu80cJ07n00a8eNSukkqTjbZxZr7ys8CXLxpg+mon5ZCXYC57/1DH3zwLzcjlAWqrqmO20tE2skc7JKRsFSC8GJhJALbJ3PhBF9KubhZIdtyckP1FFh6WqlzQSgc357Ql2YC95Syu92ttc0w+HJ0l/kwyoklSkfHqNg3p8u94kRnxEzyLMrz9h2MSJAigApMe+AkeFu6+jZ stack@openstack
stack@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.

Import Public Key

Key Pair Name *

Key Type *

SSH Key

Load Public Key from a file

Choose File No file chosen

Public Key * (Modified)

Content size: 397 bytes of 16.00 KB

```
ssh-rsa
AAAAB3NzaC1yc2EAAAQABAAQCK1b1SvKCqKeR1zXtrFLwm91HH+2e/uE0usO/RHIKhCWa/Qq7rkDaeK
ZwDdgOvhLcAwedFuk89kDUH+0VdzR6m/k0J3QDYi5UCzK9bRxR6hBn04VGfmi7rQJ79oX9kqQQWmtM1wSLQ3F
Robul1TlogM6rlY/lZiXRZ52gFGakCDzNh7rwMtYtKM3hf5syK0er4v71viRR/MouGvhFrd80vCDxuiFmrnt+bm+BT3sfG
1n7qBqLc+QM05h6TZJ5gJCzvldAVrocoIRxDnBz9jzqAQIGLEWyb3bXNYlugu3Y33bROEfPO0whE1qx6t+rBMT9lis
Giklime3AOxN3xdc9 stack@openstack
```

Cancel **Import Public Key**

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

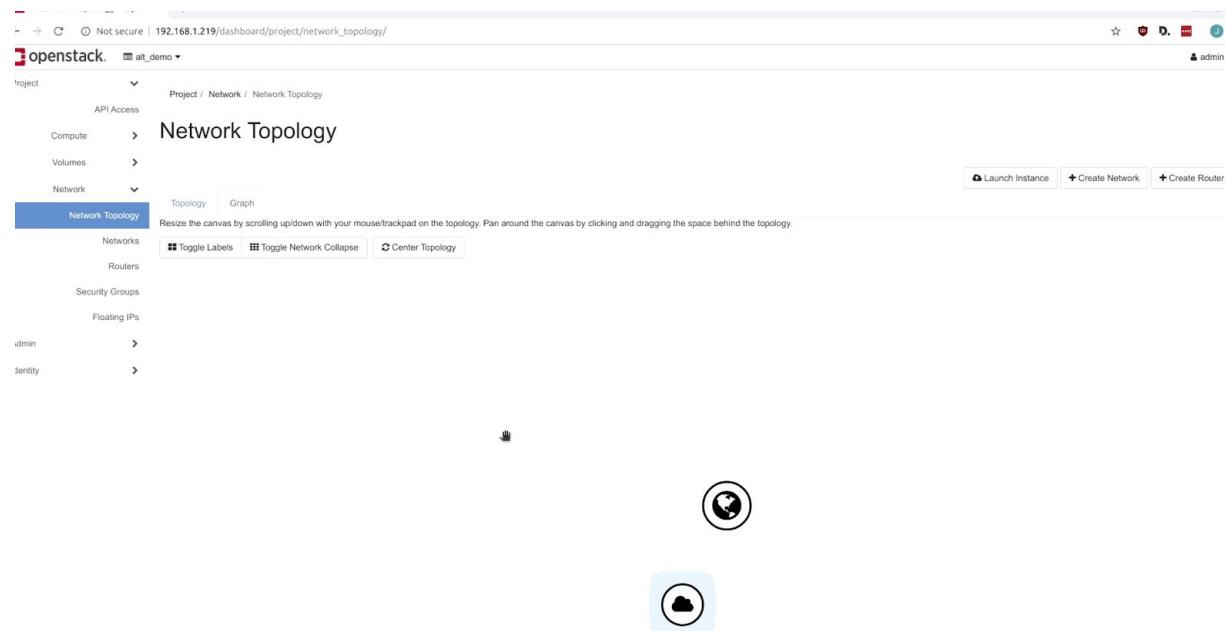
Key Pairs

Click here for filters or full text search.		+ Create Key Pair	Import Public Key	Delete Key Pairs
Displaying 1 item				
□	Name ▲	Type	Fingerprint	
□	Openstack	ssh	c3:a4:23:63:5d:76:9e:42:ac:ea:d0:ab:16:f8:1e:c4	Delete Key Pair

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.

Create Router

Router Name

Enable Admin State ?

External Network

Select network ▾

Enable SNAT

Availability Zone Hints ?

nova

Description:

Creates a router with specified parameters.

Enable SNAT will only have an effect if an external network is set.

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

The screenshot shows the OpenStack Dashboard with the URL 192.168.1.219/dashboard/project/routers/. The page title is "Routers - OpenStack Dashboard". The left sidebar has "Project" selected under "Compute". The main content area shows a table titled "Routers" with one item displayed. The table columns are: Name, Status, External Network, Admin State, Availability Zones, and Actions. The single row shows "vr" as the Name, "Active" as the Status, "-" as the External Network, "UP" as the Admin State, and "-" as the Availability Zones. The Actions column contains a "Set Gateway" dropdown menu. At the top right, there are buttons for "Create Router" and "Delete Router".

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

The screenshot shows a configuration screen for a VR (Virtual Router). The title bar says "VR". Below it is a navigation bar with tabs: "Overview", "Interfaces" (which is selected), and "Static Routes". On the right side, there is a button labeled "+ Add Interface". The main content area displays a table with the following columns: Name, Fixed IPs, Status, Type, Admin State, and Actions. A message at the bottom states "No items to display."

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.

Add Interface

Subnet *

shared: 192.168.233.0/24 (shared-subnet) ▾

This field is required.

IP Address (optional) ⓘ

Cancel Submit

Description:

You can connect a specified subnet to the router.

If you don't specify an IP address here, the gateway's IP address of the selected subnet will be used as the IP address of the newly created interface of the router. If the gateway's IP address is in use, you must use a different address which belongs to the selected subnet.

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

VR

Set Gateway ▾

OverviewInterfacesStatic Routes

+ Add InterfaceDelete Interfaces

Displaying 1 item					
<input type="checkbox"/>	Name	Fixed IPs	Status	Type	Admin State
<input type="checkbox"/>	(6b295388-0e0d)	• 192.168.233.1	Down	Internal Interface	UP

Delete Interface

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.

The screenshot shows the 'VR' router configuration page. At the top, there are tabs for 'Overview', 'Interfaces', and 'Static Routes'. Below these, a table displays one interface entry:

Name	Fixed IPs	Status	Type	Admin State	Actions
(6b295388-0e0d)	• 192.168.233.1	Down	Internal Interface	UP	<button>Delete Interface</button>

At the top right of the interface list, there are buttons for '+ Add Interface' and 'Delete Interfaces'. Above the table, a 'Set Gateway' button is highlighted with a mouse cursor.

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

Set Gateway

The screenshot shows the 'Set Gateway' configuration dialog. It includes fields for 'External Network' (set to 'public'), 'Enable SNAT' (checked), and a 'Description' section with explanatory text. At the bottom are 'Cancel' and 'Submit' buttons, with 'Submit' being the target of a mouse cursor.

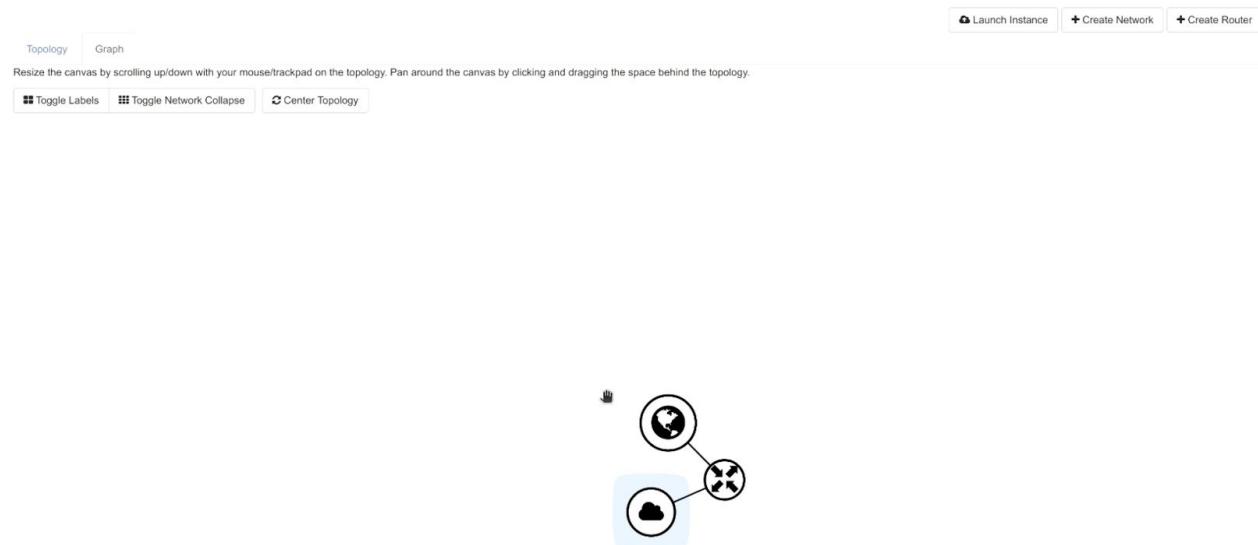
External Network *

Enable SNAT

Description:
You can connect a specified external network to the router. The external network is regarded as a default route of the router and the router acts as a gateway for external connectivity.

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.

Network Topology



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.

The screenshot shows the OpenStack dashboard interface. The URL in the browser is 192.168.1.207/dashboard/project/instances/. The top navigation bar includes links for Home, Project, API Access, Compute, Instances, Overview, Images, Instance Name, Image Name, IP Address, Flavor, Key Pair, Status, Availability Zone, Task, Power State, Age, Actions, and Launch Instance. The Instances link is highlighted in blue. On the left, a sidebar lists categories like Project, Compute, Storage, Network, and Admin. The main content area displays a table with the following columns: Instance ID, Power State, Age, and Actions. A message at the bottom of the table says "No items to display." The status bar at the bottom right shows "admin" and a timestamp "2014-07-10 10:45:45".

Step 3: Click on “Launch Instance”

Launch Instance

Details

Please provide the initial hostname for the instance, the availability zone where it will be deployed, and the instance count. Increase the Count to create multiple instances with the same settings.

Instance Name * Ubuntu18.04

Description Webserver-01

Total Instances (10 Max) 10%

Availability Zone nova

Count * 1

0 Current Usage
1 Added
9 Remaining

Source *
Flavor *
Networks *
Network Ports
Security Groups
Key Pair
Configuration
Server Groups
Scheduler Hints
Metadata

Cancel **Back** **Next >** **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”.

Launch Instance

Source *

Details

Flavor *

Networks *

Network Ports

Security Groups

Key Pair

Configuration

Server Groups

Scheduler Hints

Metadata

Allocated

Select Boot Source

Create New Volume

Image

Volume Size (GB) *

1

Delete Volume on Instance Delete

Yes No

Yes No

Select an item from Available items below

▼ Available 7 Select one

Click here for filters or full text search.

Name	Updated	Size	Type	Visibility	
ArchLinux	10/18/19 6:15 PM	1.85 GB	qcow2	Public	
CentOS	10/18/19 6:08 PM	898.75 MB	qcow2	Public	
cirros-0.4.0-x86_64-disk	10/18/19 5:44 PM	12.13 MB	qcow2	Public	
CoreOS	10/18/19 6:14 PM	1016.44 MB	qcow2	Public	
Debian-10	10/18/19 6:10 PM	528.85 MB	qcow2	Public	
Fedora	10/18/19 6:06 PM	316.88 MB	qcow2	Public	
Ubuntu-18.04	10/18/19 6:09 PM	328.25 MB	qcow2	Public	

✗ Cancel

< Back

Next >

Launch Instance

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”

Launch Instance

Details		Flavors manage the sizing for the compute, memory and storage capacity of the instance.							
Source		Allocated							
		Name	VCPUS	RAM	Total Disk	Root Disk	Ephemeral Disk	Public	
Flavor *		Select an item from Available items below							
Networks *		▼ Available (12) Select one							
Network Ports		<input type="text"/> Click here for filters or full text search.							
Security Groups		Name	VCPUS	RAM	Total Disk	Root Disk	Ephemeral Disk	Public	
Key Pair		➤ m1.nano	1	64 MB	1 GB	1 GB	0 GB	Yes	
Configuration		➤ m1.micro	1	128 MB	1 GB	1 GB	0 GB	Yes	
Server Groups		➤ cirros256	1	256 MB	1 GB	1 GB	0 GB	Yes	
Scheduler Hints		➤ m1.tiny	1	512 MB	1 GB	1 GB	0 GB	Yes	
Metadata		➤ ds512M	1	512 MB	5 GB	5 GB	0 GB	Yes	
		➤ ds1G	1	1 GB	10 GB	10 GB	0 GB	Yes	
		➤ m1.small	1	2 GB	20 GB	20 GB	0 GB	Yes	
		➤ ds2G	2	2 GB	10 GB	10 GB	0 GB	Yes	
		➤ m1.medium	2	4 GB	40 GB	40 GB	0 GB	Yes	
		➤ ds4G	4	4 GB	20 GB	20 GB	0 GB	Yes	
		➤ m1.large	4	8 GB	80 GB	80 GB	0 GB	Yes	
		➤ m1.xlarge	8	16 GB	160 GB	160 GB	0 GB	Yes	

Cancel
 < Back
 Next >
 Launch Instance

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

Launch Instance

Details Networks * Source Flavor Networks * Network Ports Security Groups Key Pair Configuration Server Groups Scheduler Hints Metadata

Allocated Networks provide the communication channels for instances in the cloud. Select networks from those listed below.

Available Select at least one network

Click here for filters or full text search.

Network	Subnets Associated	Shared	Admin State	Status
public	ipv6-public-subnet public-subnet	No	Up	Active
shared	shared-subnet	Yes	Up	Active

Cancel Back Next Launch Instance

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

Launch Instance

Details Select the security groups to launch the instance in.

Source Allocated 1

Name	Description
default	Default security group

Flavor Available 0 Select one or more

Name	Description
No available items	

Networks

Network Ports

Security Groups

Key Pair

Configuration

Server Groups

Scheduler Hints

Metadata

x Cancel < Back Next > Launch Instance

The screenshot shows the 'Launch Instance' wizard in progress, specifically the 'Security Groups' step. On the left, a sidebar lists various configuration tabs: Details, Source, Flavor, Networks, Network Ports, Security Groups (which is highlighted in blue), Key Pair, Configuration, Server Groups, Scheduler Hints, and Metadata. The main content area is titled 'Select the security groups to launch the instance in.' It shows two sections: 'Allocated' (containing one item, 'default' with a description of 'Default security group') and 'Available' (containing zero items). Below these sections is a search bar with placeholder text 'Click here for filters or full text search.' At the bottom right are navigation buttons: '< Back', 'Next >' (with a hand cursor icon indicating it's being clicked), and 'Launch Instance'.

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

Launch Instance

Details A key pair allows you to SSH into your newly created instance. You may select an existing key pair, import a key pair, or generate a new key pair.

Source [+ Create Key Pair](#) [Import Key Pair](#)

Flavor

Networks

Network Ports

Security Groups

Key Pair

Allocated

Displaying 1 item

Name	Type	Fingerprint
openstack	ssh	41:7f:46:8d:a9:62:35:78:cd:99:32:77:32:49:1d:00

Available **0** Select one

Configuration

Server Groups

Scheduler Hints

Metadata

Displaying 0 items

No items to display.

Displaying 0 items

[Cancel](#) [Back](#) [Next >](#) [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.

Project / Compute / Instances

Instances

Displaying 1 item

Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone	Task	Power State	Age	Actions
Ubuntu18.04	-	192.168.233.160	ds512M	openstack	Active	az1 nova	None	Running	0 minutes	Create Snapshot

Displaying 1 item

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.

Instances

Displaying 1 item

Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone	Task	Power State	Age	Actions
Ubuntu18.04	-	192.168.233.160	ds512M	openstack	Active	az1 nova	None	Running	0 minutes	Create Snapshot

Displaying 1 item

- Associate Floating IP
- Attach Interface
- Detach Interface
- Edit Instance
- Attach Volume
- Detach Volume
- Update Metadata
- Edit Security Groups
- Edit Port Security Groups
- Console
- View Log
- Rescue Instance
- Pause Instance
- Suspend Instance
- Shelve Instance
- Resize Instance
- Lock Instance
- Soft Reboot Instance
- Hard Reboot Instance
- Shut Off Instance
- Rebuild Instance
- Delete Instance

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.

Manage Floating IP Associations

IP Address *

172.24.4.156



Select the IP address you wish to associate with the selected instance or port.

Port to be associated *

Ubuntu18.04: 192.168.233.160



Cancel

Associate



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

Instances

Instances											
	Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone	Task	Power State	Age	Actions
<input type="checkbox"/>	Ubuntu18.04	-	192.168.233.160, 172.24.4.156	ds512M	openstack	Active	us	nova	None	Running	0 minutes
Displaying 1 item											

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:mIQtwSUBwB0okHYaMQuOV5YmBYyRTTbgzkmZTgP1swo.
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=q>

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.

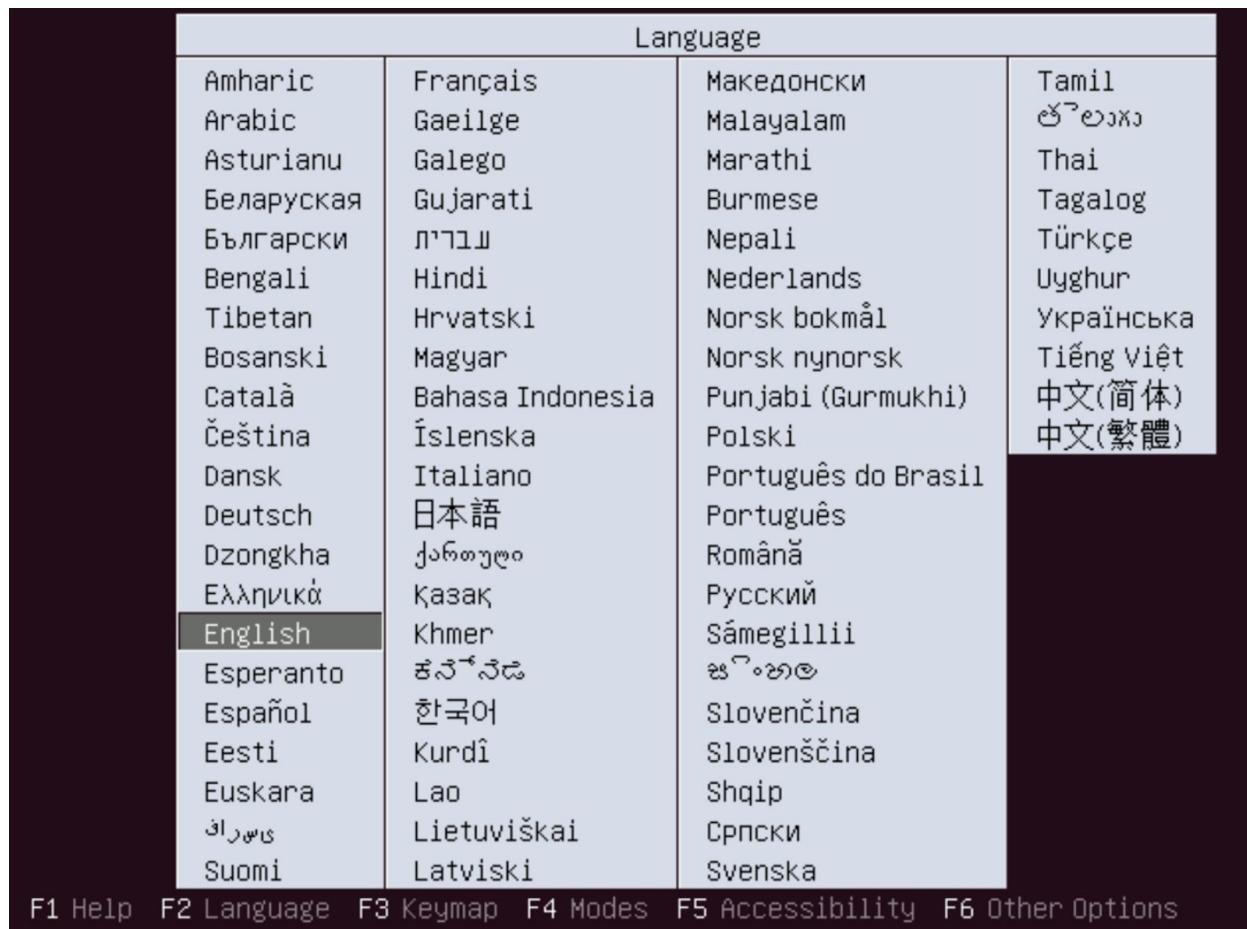
Important notice!

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.



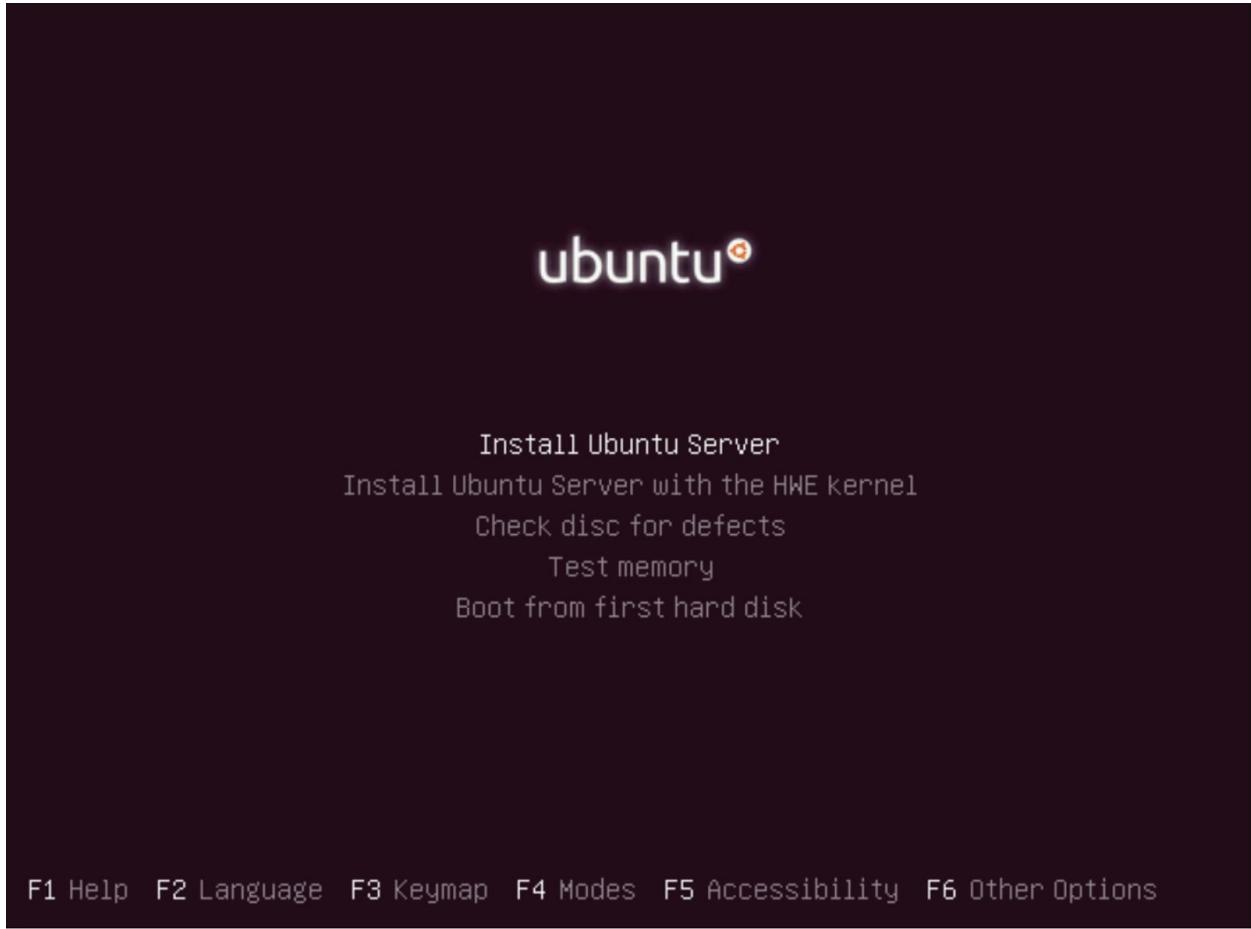
Step 5: After the language selection menu, the main install menu will be shown on screen. Make sure “Install Ubuntu Server” is showing in bold; this means the installation of Ubuntu is selected. You can use the arrow keys on your keyboard to change the selection if this is not the case.

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.

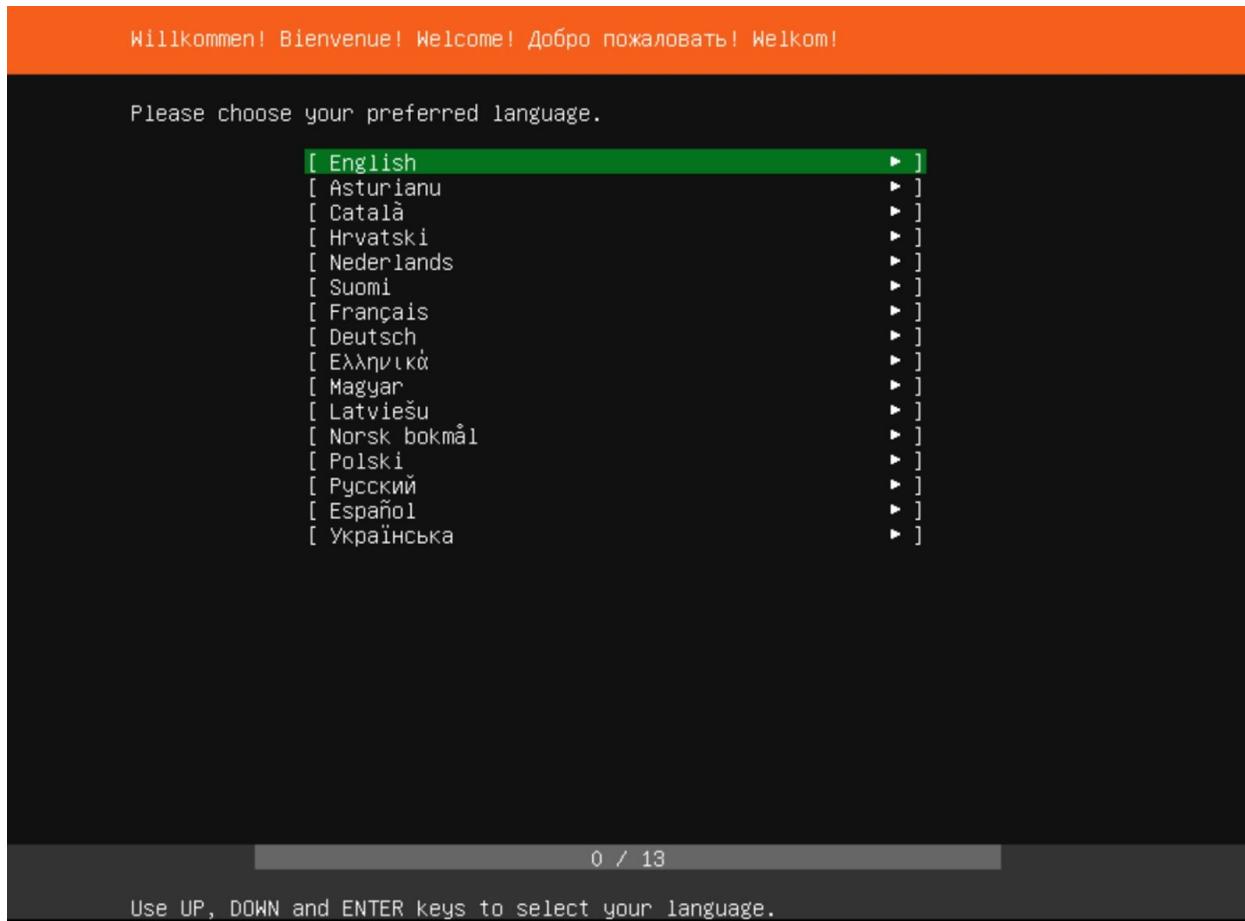
If your system does not load the next screen and no hard disk activity is present, please redownload and/or reflash/reburn your Ubuntu installation image. It can be faulty.



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.

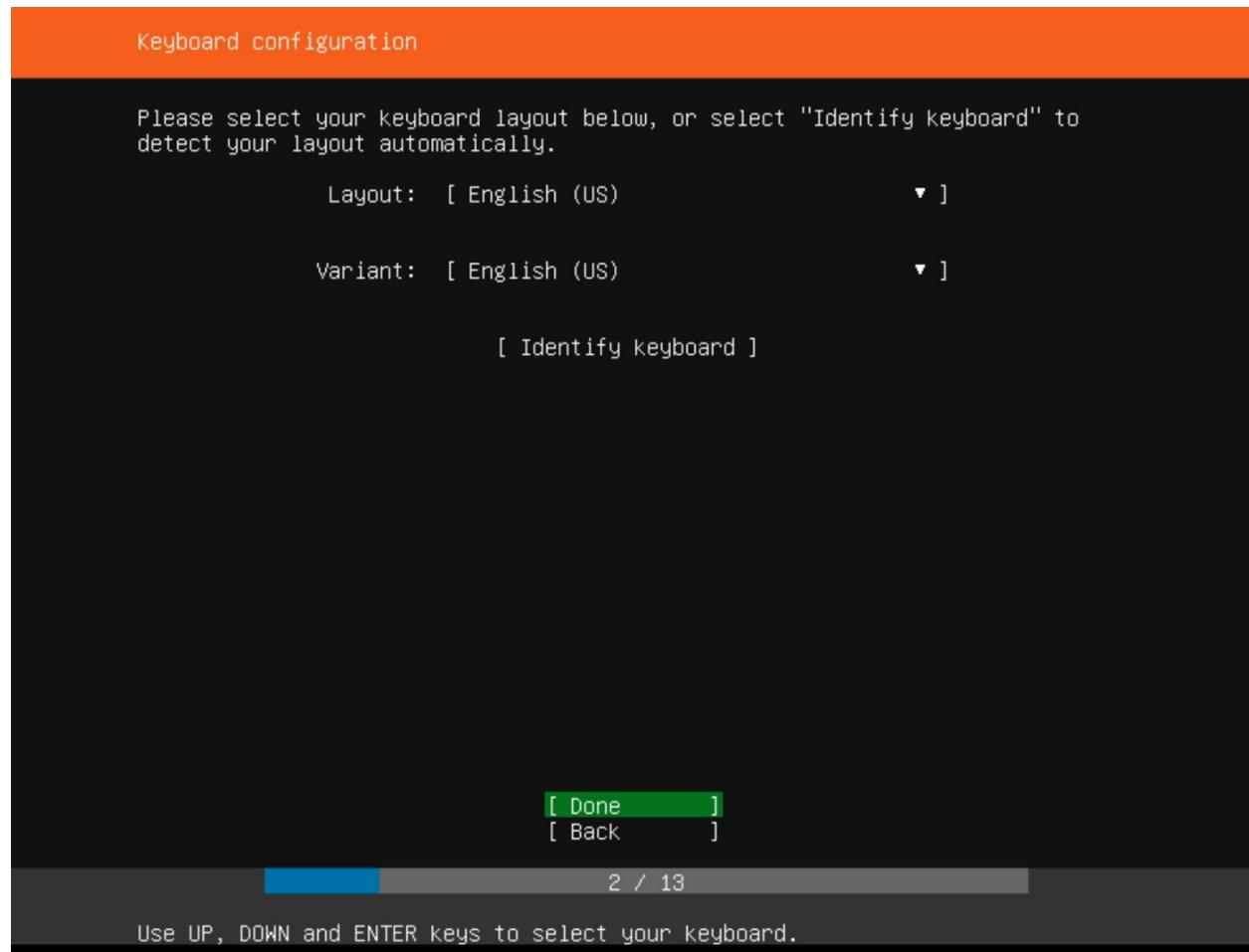
Press “**ENTER**” to continue the installation



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.

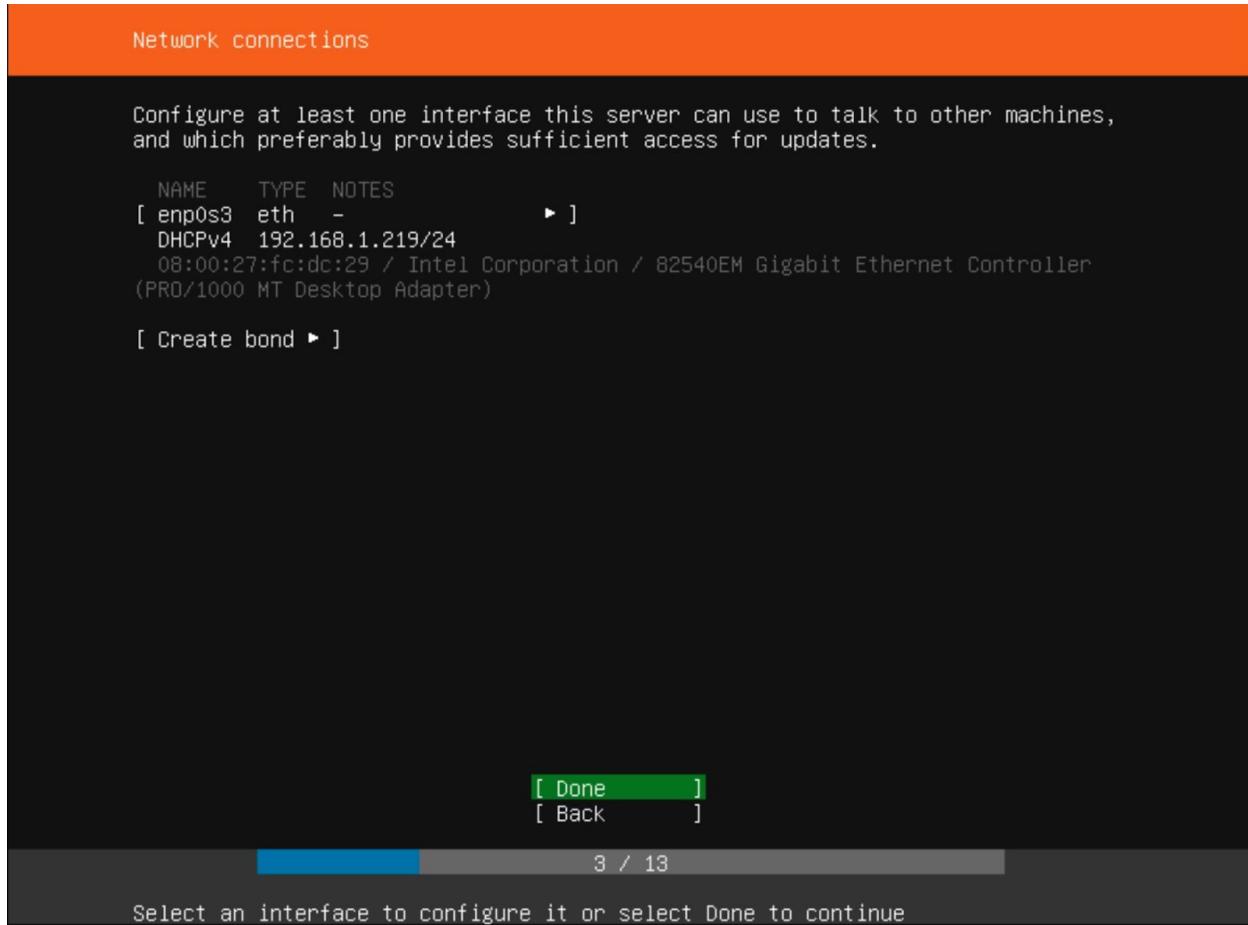
You can also manually set up your network interface. However you would need to know how to set this up properly.

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.



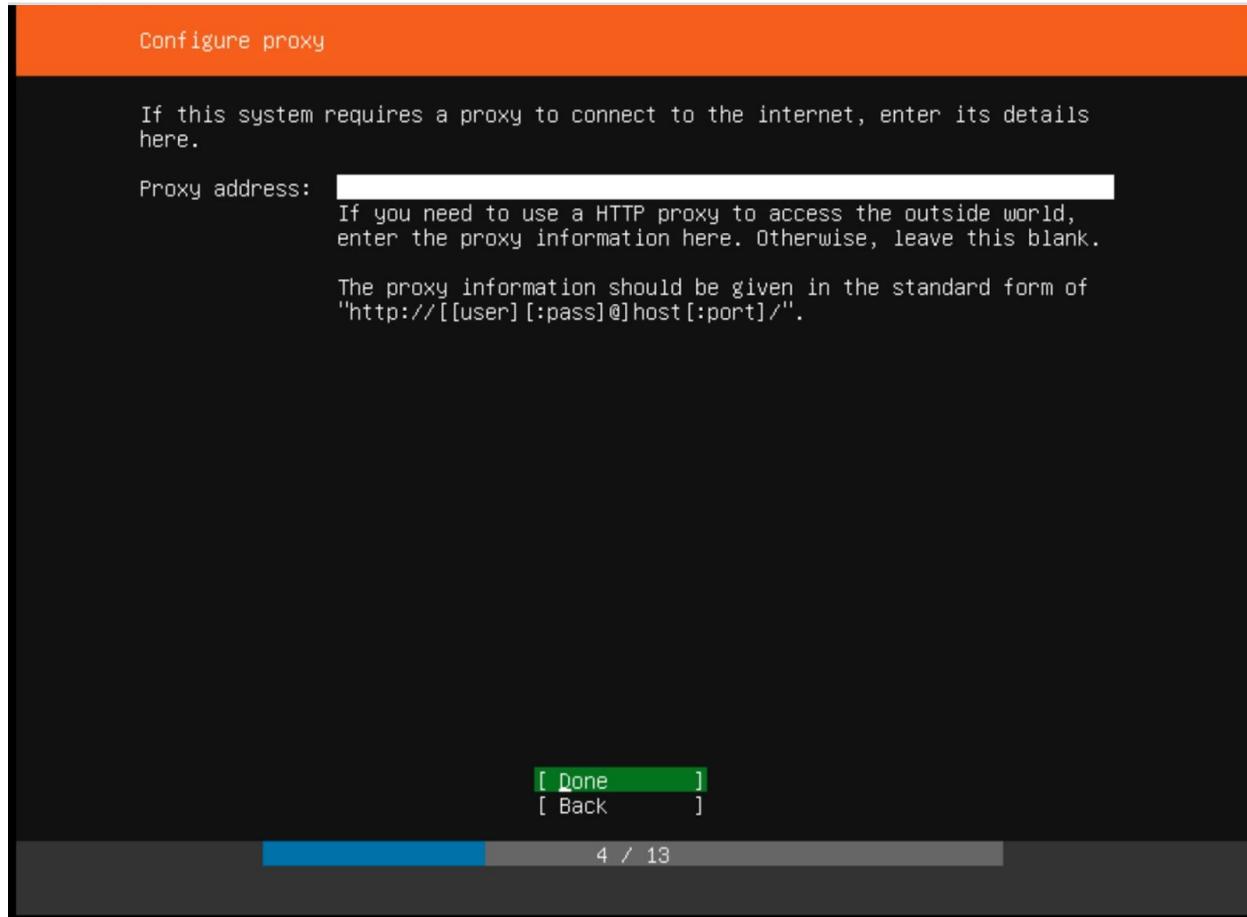
Press “**Enter**” to continue

Step 9:

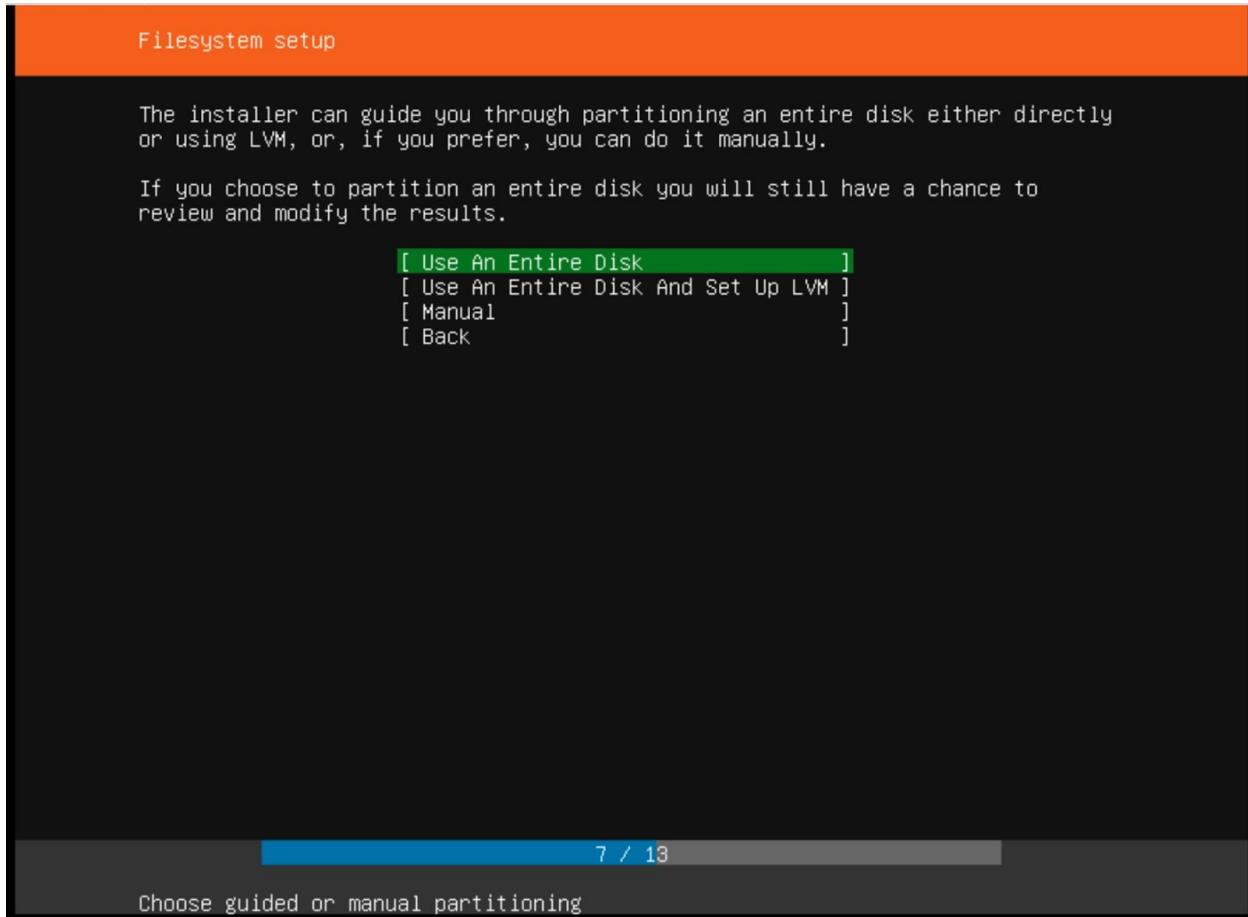
The default setting is normally good to go. You usually do not need to configure any proxy addresses. If you do however, you can specify it here.

For this manual we leave it blank because our network configuration does not need a proxy address.

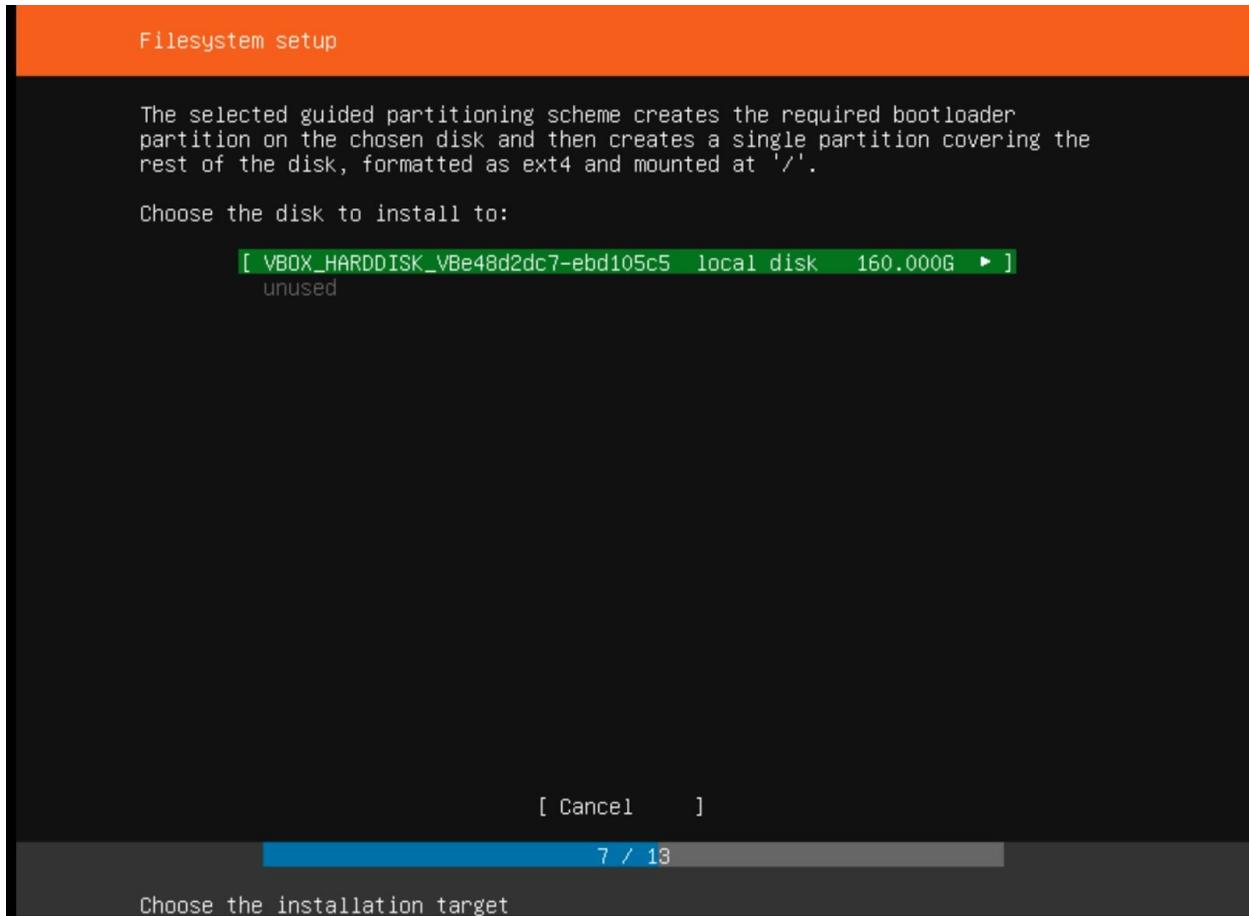
Press “**Enter**” to continue



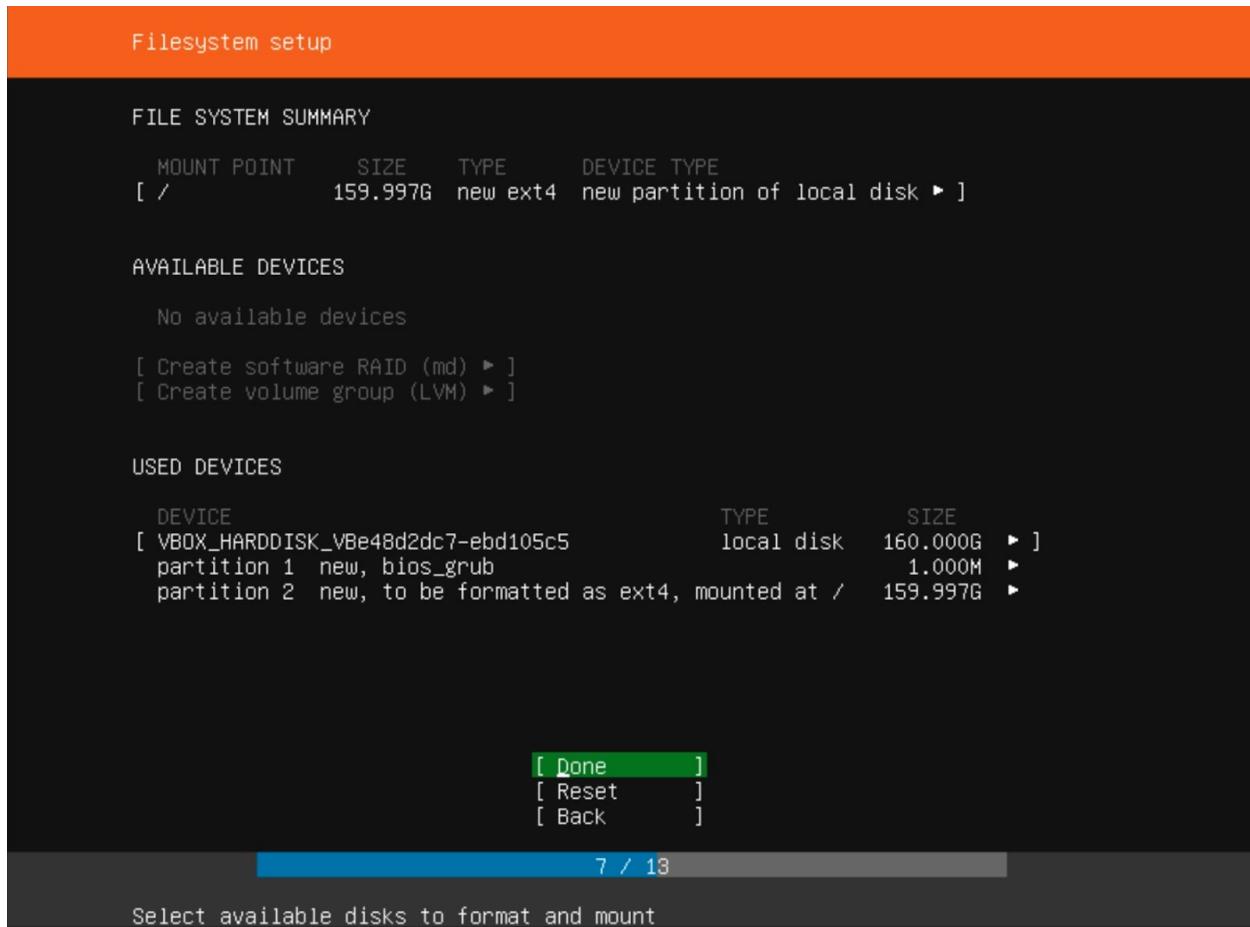
Step 10: In this step we choose to use the entire hard disk for the installation of Ubuntu server 18.04. For the next installation step click on “**Enter**”.



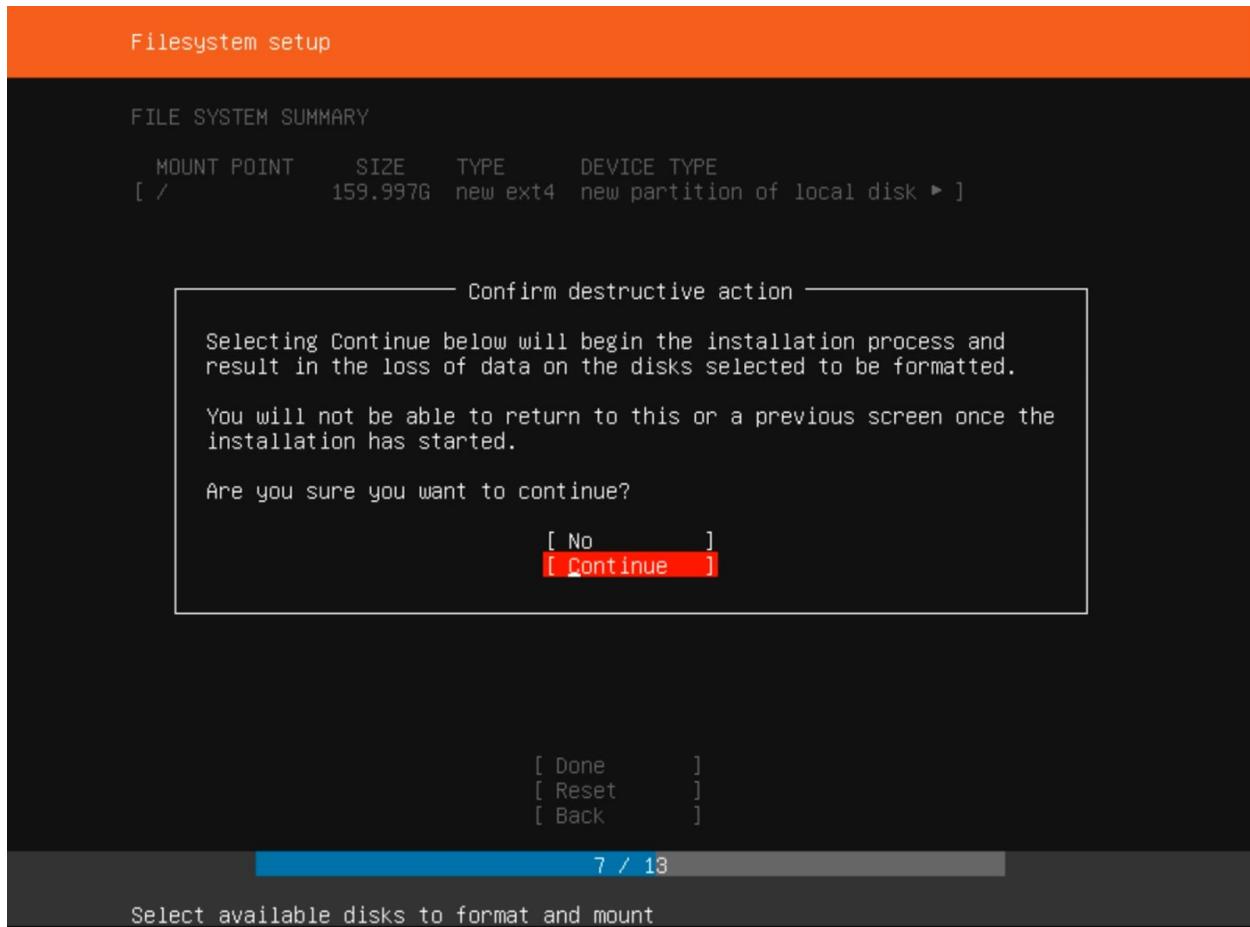
We select the only hard disk in the computer, the hard disk is need for coping the installation files in to the hard disk. For the next installation step click on “**Enter**”



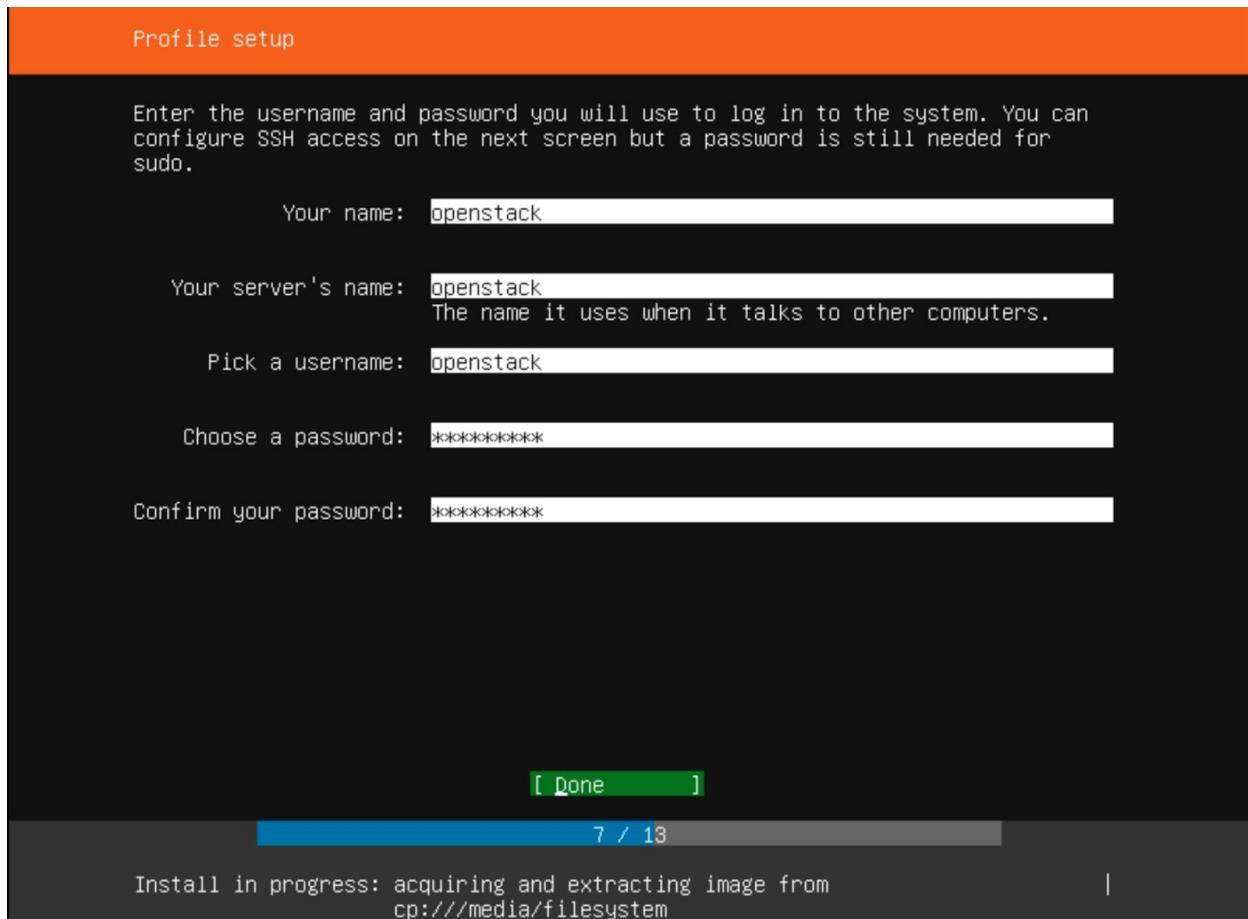
Step 11: In this step the installation of Ubuntu server gives an overview of the options we selected during the previous steps, the installation make sure that you don't make any configuration errors before continuing with the installation of Ubuntu server 18.04. For the next installation step click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



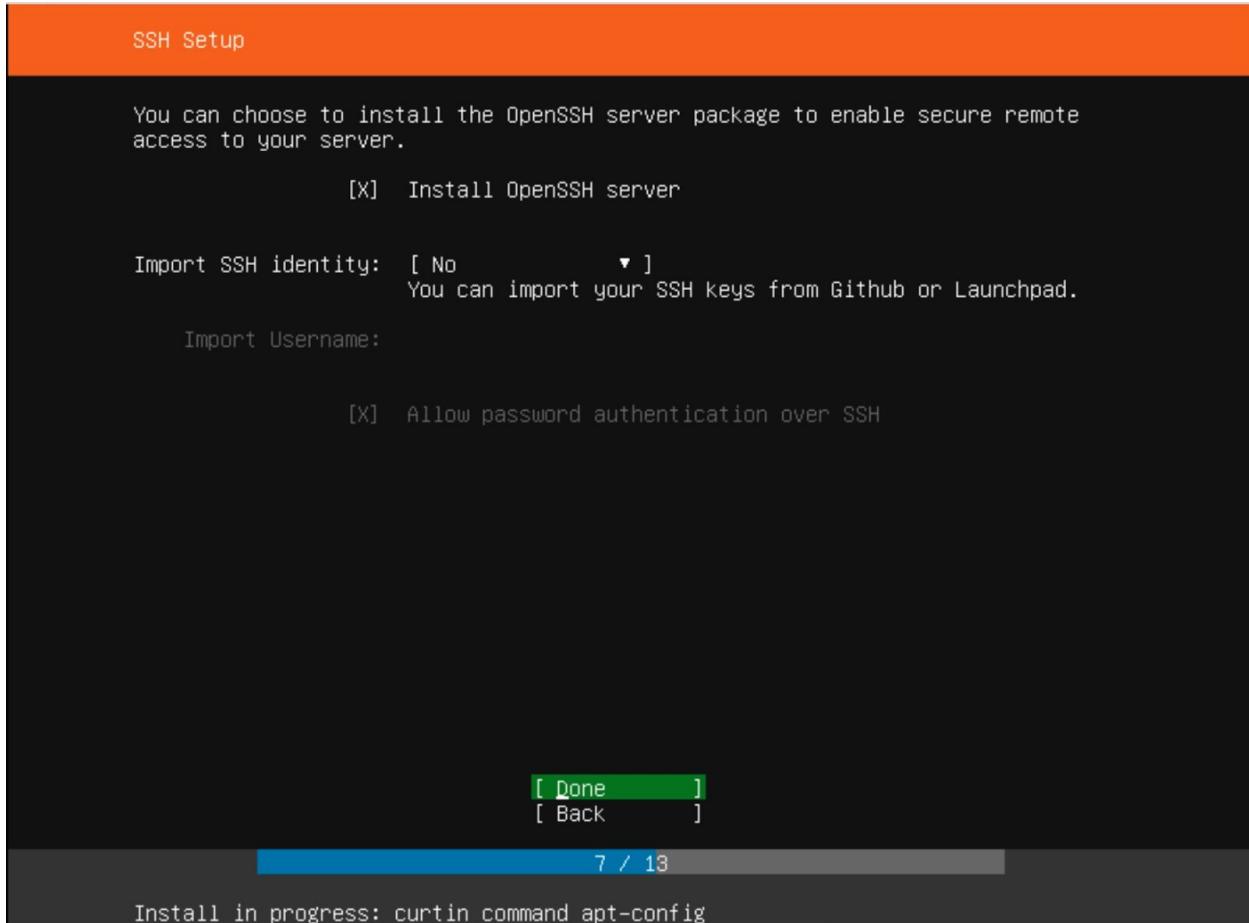
Step 12: At this step you must confirm that you want to make the configuration. Click on “**Continue**”. The settings will now be applied to your system and the installation will start. You have crossed the point of no return here. You will not be able to reconfigure the previous steps from here on.



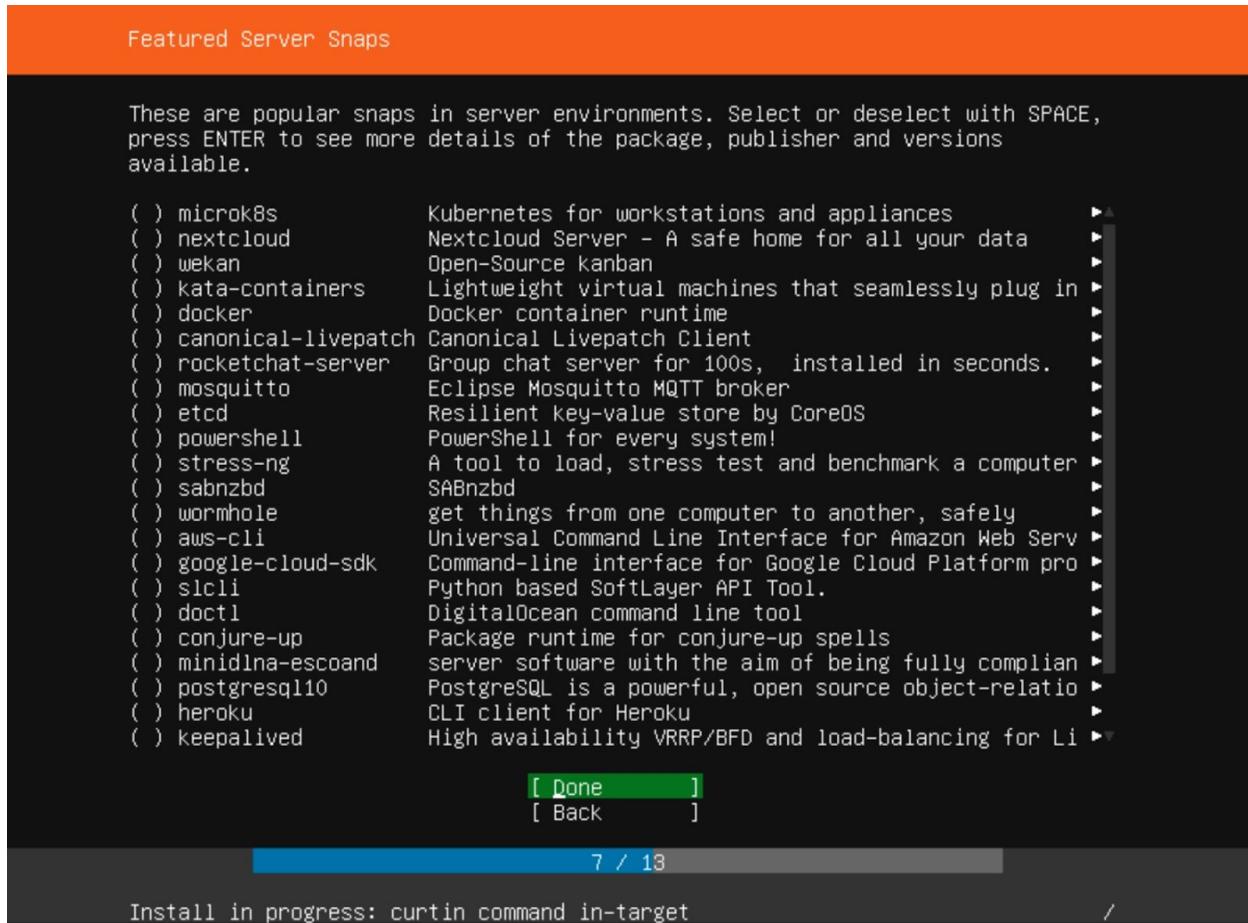
Step 13: After the main installation of Ubuntu 18.04 is complete. You will need to specify an admin user. We will use the username as “openstack” and password “openstack” in this example, however you can specify this differently if you wish to.



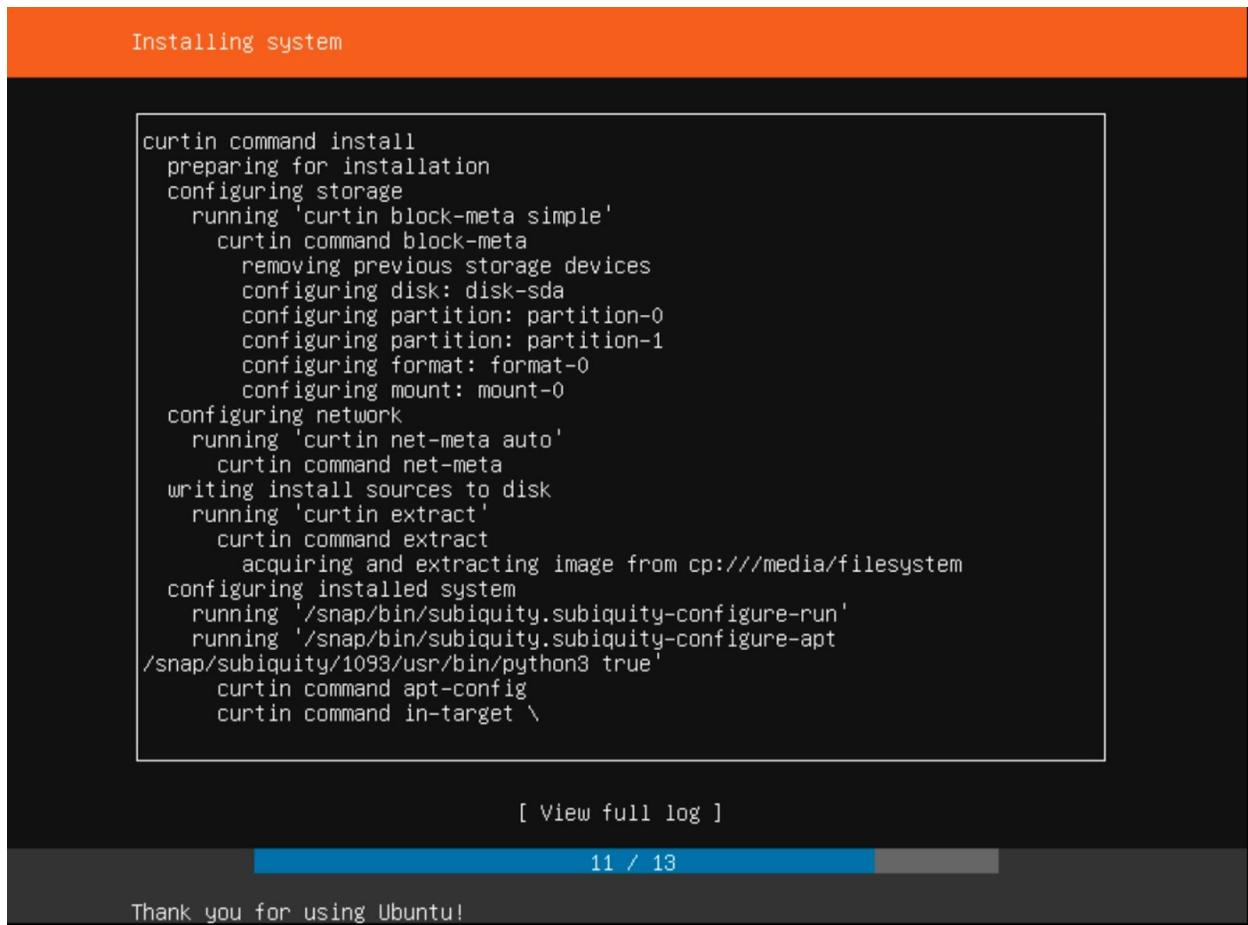
Step 14: To manage the server remotely, choose to install an SSH server on the server. This allows you to manage the server remotely. Select “**Install OpenSSH server**” at Ubuntu installer.



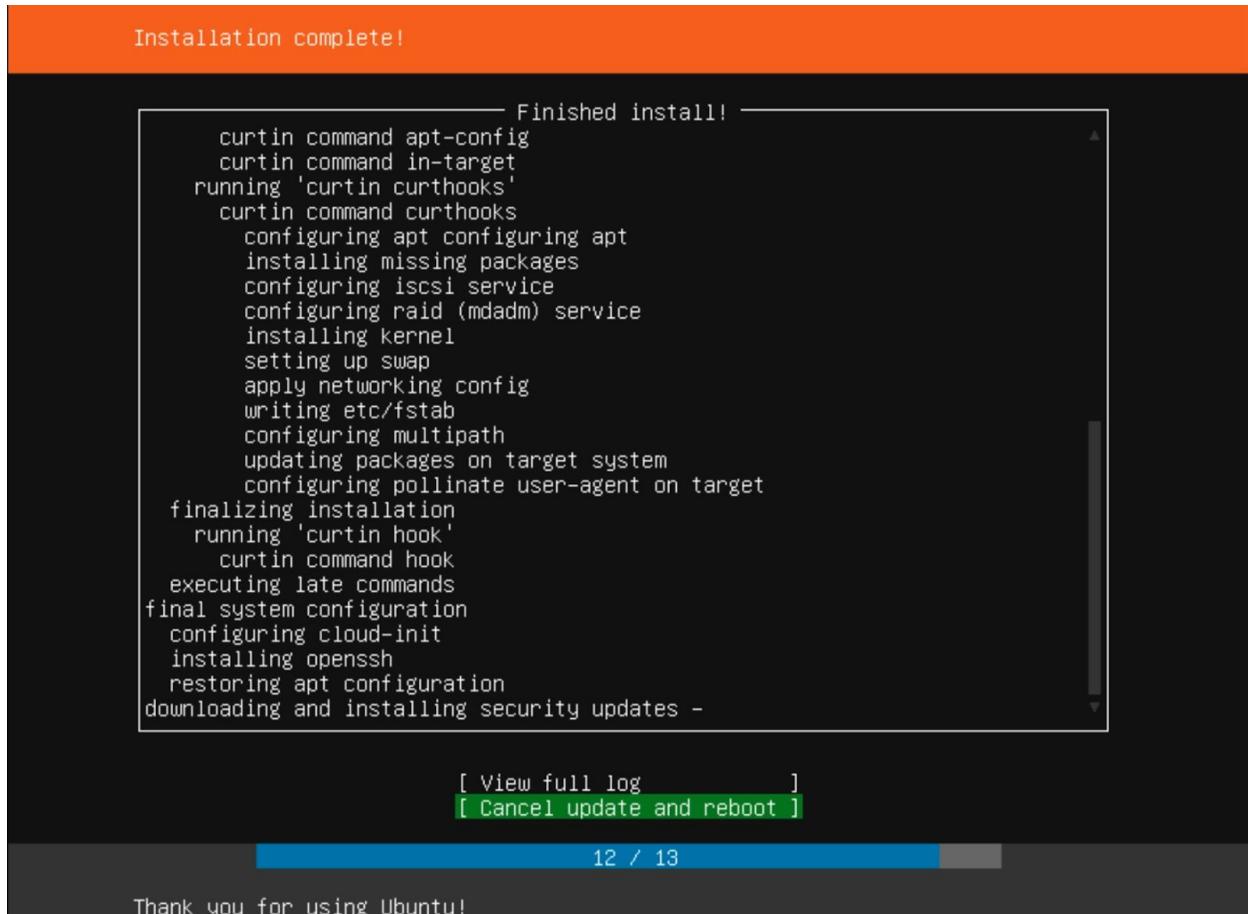
Step 15: During this step you can install software packages, we do not do this in this tutorial because we are going to use the server for specific purposes



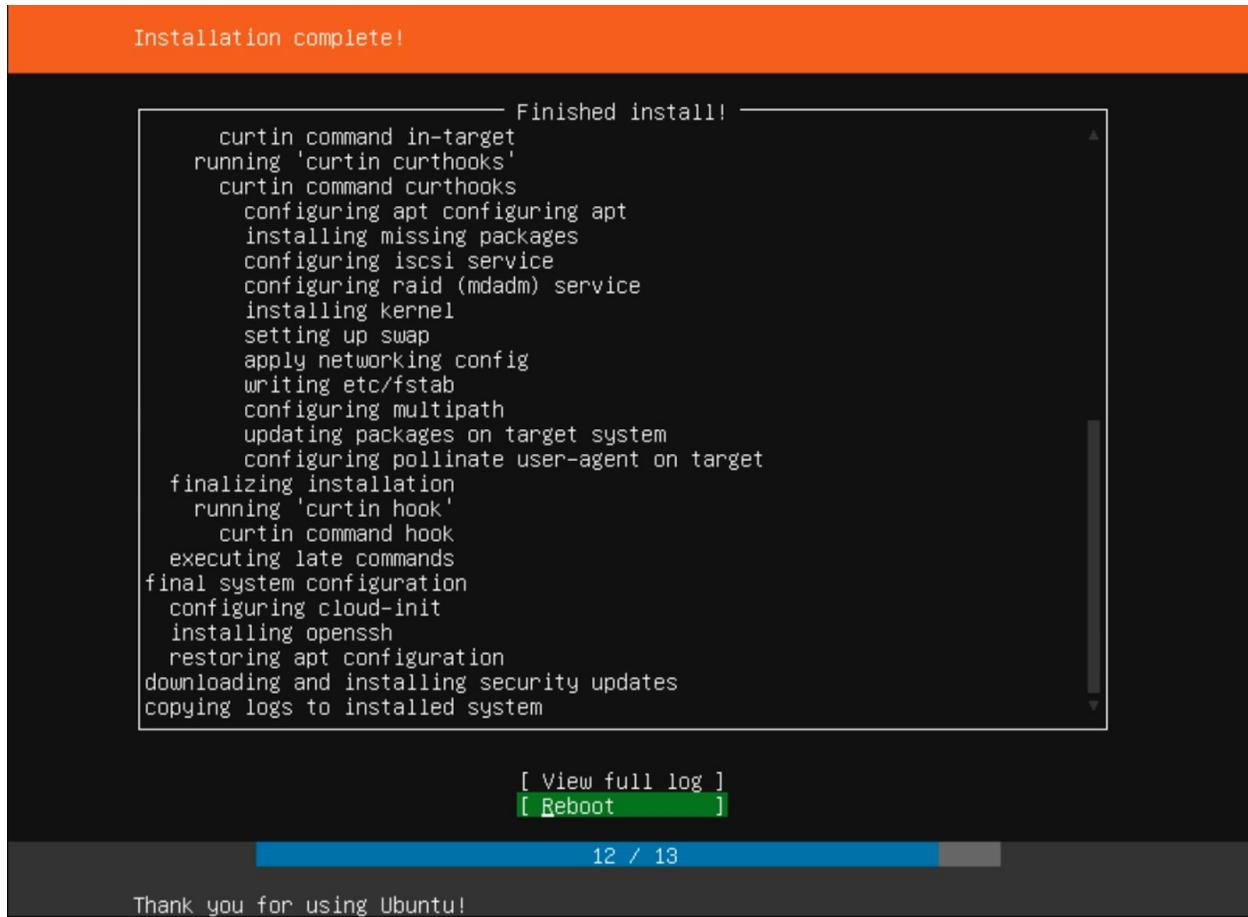
Step 16: Now you will have to wait until the system is fully installed on your server.



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.



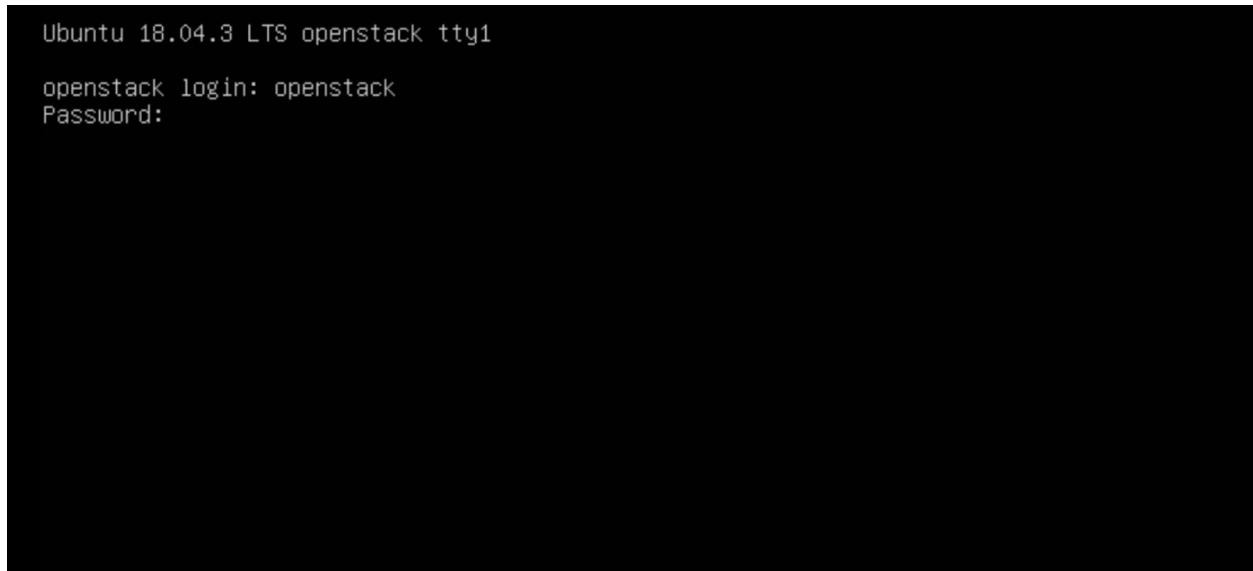
Step 18 :After the updates have been completely downloaded and installed, you can restart the server and click on "reboot".



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

```
[FAILED] Failed unmounting /lib/modules.  
[ OK ] Unmounted /target/run.  
      Unmounting /target...  
[ OK ] Unmounted /tmp.  
[ OK ] Stopped target Swap.  
[ OK ] Unmounted /rofs.  
[ OK ] Unmounted Mount unit for subiquity, revision 1093.  
[ OK ] Unmounted Mount unit for core, revision 7270.  
[ OK ] Unmounted /target.  
[ OK ] Reached target Unmount All Filesystems.  
[ OK ] Stopped target Local File Systems (Pre).  
[ OK ] Stopped Create Static Device Nodes in /dev.  
[ OK ] Stopped Remount Root and Kernel File Systems.  
[ OK ] Reached target Shutdown.  
      Starting Shuts down the "live" preinstalled system cleanly...  
      Stopping Monitoring of LVM2 mirrors, snapshots etc. using dmeventd or progress polling...  
[ OK ] Stopped Monitoring of LVM2 mirrors, snapshots etc. using dmeventd or progress polling.  
      Stopping LVM2 metadata daemon...  
[ OK ] Stopped LVM2 metadata daemon.  
Please remove the installation medium, then press ENTER:  
-
```

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.

```
Ubuntu 18.04.3 LTS openstack tty1

openstack login: openstack
Password:
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 Sun Oct 20 12:56:44 UTC 2019

 System load:  0.99           Processes:      87
 Usage of /:   3.7% of 156.49GB  Users logged in:  0
 Memory usage: 2%              IP address for enp0s3: 192.168.1.219
 Swap usage:   0%

48 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.

openstack@openstack:~$ sudo apt update && sudo apt upgrade -y
[sudo] password for openstack:
```

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%.

```
Get:36 http://nl.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libiscctg160 amd64 1:9.11.3+dfsg-1ubuntu1.9 [48.5 kB]
Get:37 http://nl.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libisccc160 amd64 1:9.11.3+dfsg-1ubuntu1.9 [17.9 kB]
Get:38 http://nl.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libirs160 amd64 1:9.11.3+dfsg-1ubuntu1.9 [19.1 kB]
Get:39 http://nl.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libdns1100 amd64 1:9.11.3+dfsg-1ubuntu1.9 [965 kB]
Get:40 http://nl.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libisc169 amd64 1:9.11.3+dfsg-1ubuntu1.9 [238 kB]
Get:41 http://nl.archive.ubuntu.com/ubuntu bionic-updates/main amd64 liblwres160 amd64 1:9.11.3+dfsg-1ubuntu1.9 [34.8 kB]
Get:42 http://nl.archive.ubuntu.com/ubuntu bionic-updates/main amd64 landscape-common amd64 18.01-0ubuntu3.4 [85.4 kB]
Get:43 http://nl.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libldap-common all 2.4.45+dfsg-1ubuntu1.4 [16.9 kB]
Get:44 http://nl.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libldap-2.4-2 amd64 2.4.45+dfsg-1ubuntu1.4 [155 kB]
Get:45 http://nl.archive.ubuntu.com/ubuntu bionic-updates/main amd64 software-properties-common all 0.96.24.32.11 [9,996 B]
Get:46 http://nl.archive.ubuntu.com/ubuntu bionic-updates/main amd64 python3-software-properties all 0.96.24.32.11 [23.6 kB]
Get:47 http://nl.archive.ubuntu.com/ubuntu bionic-updates/main amd64 sosreport amd64 3.6-1ubuntu0.18.04.3 [136 kB]
Get:48 http://nl.archive.ubuntu.com/ubuntu bionic-updates/main amd64 cloud-init all 19.2-36-g059d049c-0ubuntu2~18.04.1 [404 kB]
Fetched 27.1 MB in 3s (7,970 kB/s)
Extracting templates from packages: 100%
Preconfiguring packages ...
(Reading database ... 66877 files and directories currently installed.)
Preparing to unpack .../base-files_10.1ubuntu2.7_amd64.deb ...
Warning: Stopping motd-news.service, but it can still be activated by:
  motd-news.timer
Unpacking base-files (10.1ubuntu2.7) over (10.1ubuntu2.6) ...
Setting up base-files (10.1ubuntu2.7) ...
Progress: [ 1%] [.....]
```

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”.

```
update-initramfs: deferring update (trigger activated)
Setting up libldap-2.4-2:amd64 (2.4.45+dfsg-1ubuntu1.4) ...
Setting up nplan (0.98-0ubuntu1~18.04.1) ...
Setting up sosreport (3.6-1ubuntu0.18.04.3) ...
Setting up libdns-export1100 (1:9.11.3+dfsg-1ubuntu1.9) ...
Setting up python3-software-properties (0.96.24.32.11) ...
Setting up libdns1100:amd64 (1:9.11.3+dfsg-1ubuntu1.9) ...
Setting up liblures160:amd64 (1:9.11.3+dfsg-1ubuntu1.9) ...
Setting up cloud-init (19.2-36-g059d049c-0ubuntu2~18.04.1) ...
Installing new version of config file /etc/cloud/templates/ntp.conf.debian tmpl ...
Setting up software-properties-common (0.96.24.32.11) ...
Setting up libpam-systemd:amd64 (237-3ubuntu10.31) ...
Setting up libiscfg160:amd64 (1:9.11.3+dfsg-1ubuntu1.9) ...
Setting up snapd (2.40+18.04) ...
Installing new version of config file /etc/apparmor.d/usr.lib.snapd.snap-confine.real ...
md5sum: /etc/apparmor.d/usr.lib.snapd.snap-confine: No such file or directory
snapd.failure.service is a disabled or a static unit, not starting it.
snapd.snap-repair.service is a disabled or a static unit, not starting it.
Setting up libirs160:amd64 (1:9.11.3+dfsg-1ubuntu1.9) ...
Setting up libbind9-160:amd64 (1:9.11.3+dfsg-1ubuntu1.9) ...
Setting up bind9-host (1:9.11.3+dfsg-1ubuntu1.9) ...
Setting up dnsutils (1:9.11.3+dfsg-1ubuntu1.9) ...
Processing triggers for systemd (237-3ubuntu10.31) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for dbus (1.12.2-1ubuntu1.1) ...
Processing triggers for rsyslog (8.32.0-1ubuntu4) ...
Processing triggers for mime-support (3.60ubuntu1) ...
Processing triggers for ureadahead (0.100.0-21) ...
Processing triggers for install-info (6.5.0.dfsg.1-2) ...
Processing triggers for plymouth-theme-ubuntu-text (0.9.3-1ubuntu7.18.04.2) ...
update-initramfs: deferring update (trigger activated)
Processing triggers for libc-bin (2.27-3ubuntu1) ...
Processing triggers for initramfs-tools (0.130ubuntu3.8) ...
update-initramfs: Generating /boot/initrd.img-4.15.0-65-generic
openstack@openstack:~$ sudo reboot_
```