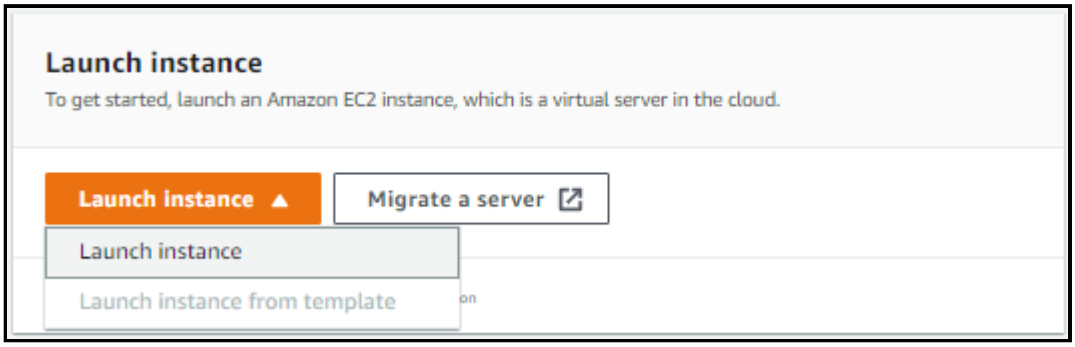
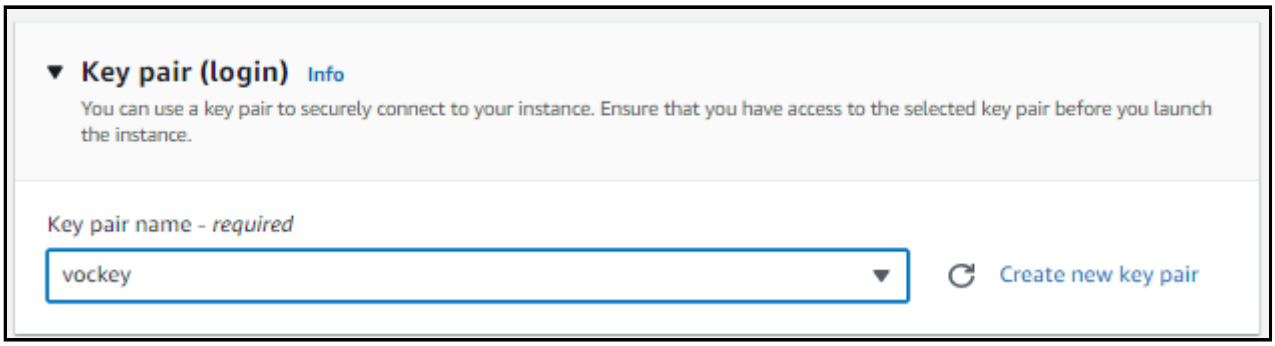
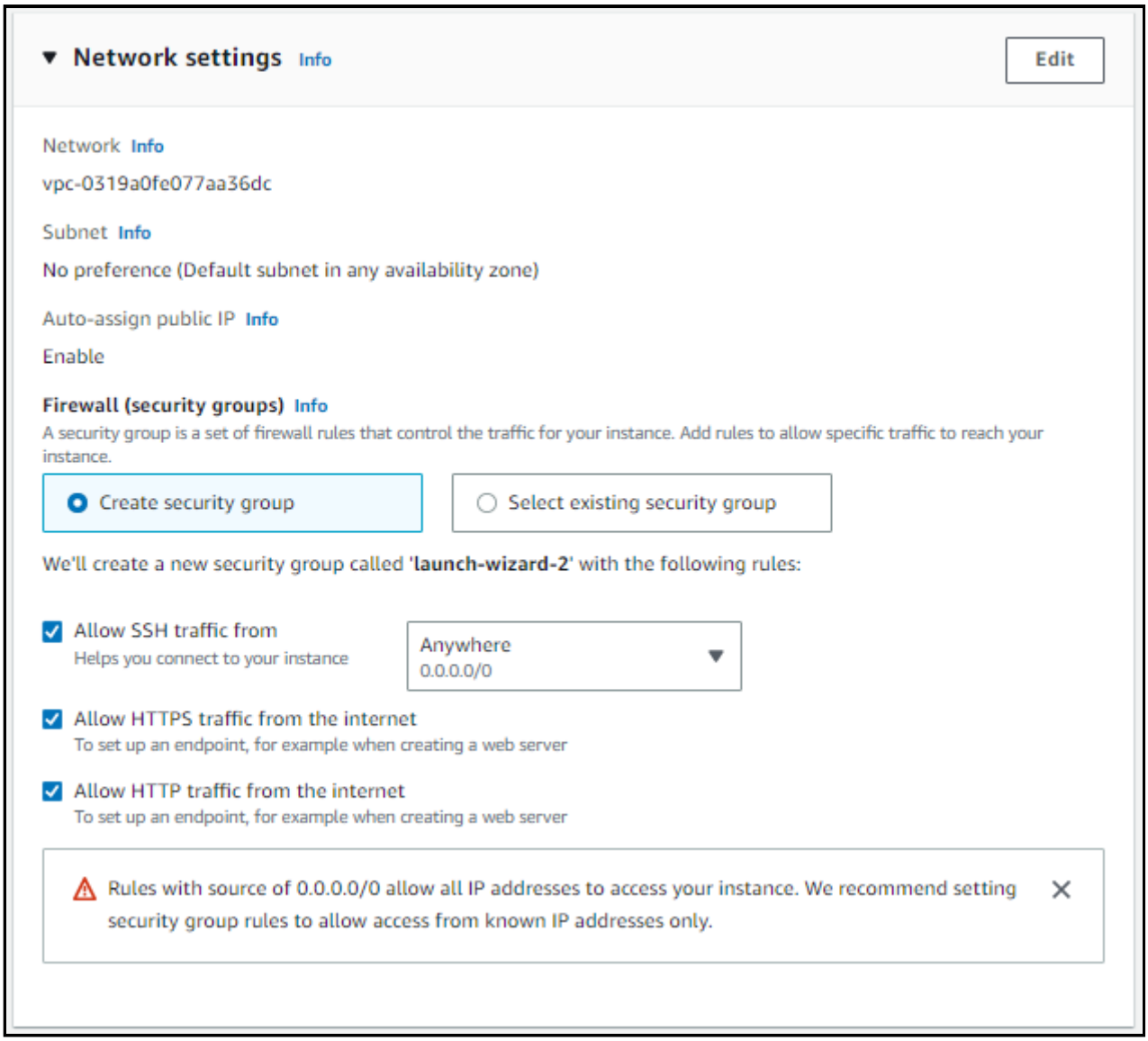
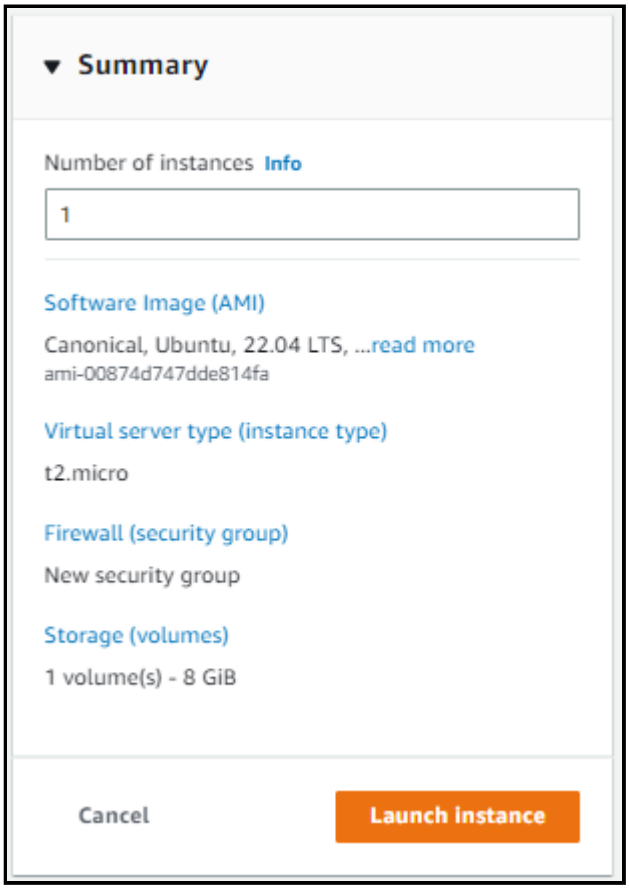
AWS - EXERCISE 1

1. Create a new EC2 instance. Name it "UbuntuDockerAWS".





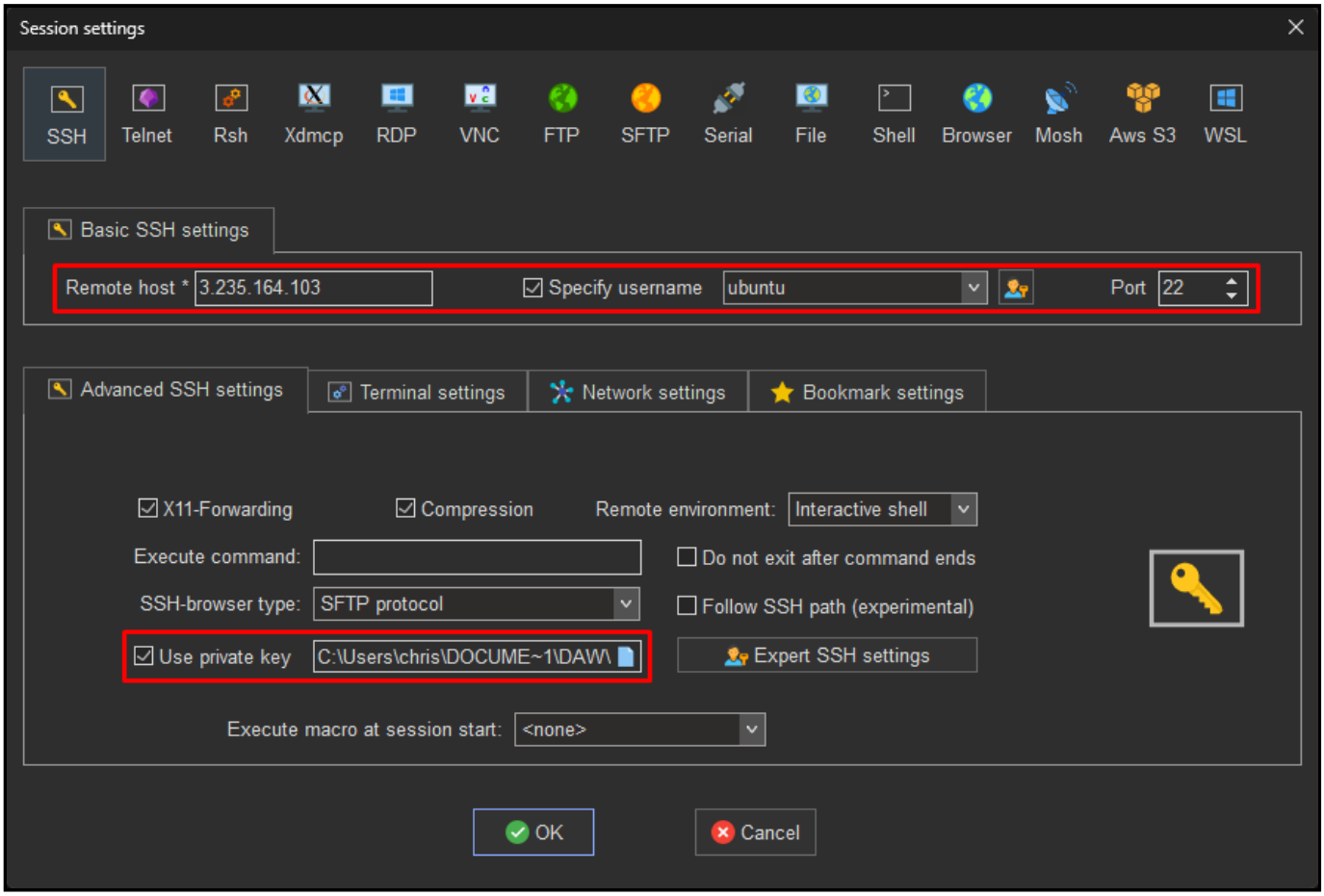

Network settings Info
Edit
Network Info
vpc-0319a0fe077aa36dc
Subnet Info
No preference (Default subnet in any availability zone)
Auto-assign public IP Info
Enable
Firewall (security groups) Info
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.
Create security group
Select existing security group
We'll create a new security group called 'launch-wizard-2' with the following rules:
Allow SSH traffic from
Helps you connect to your instance
Anywhere
0.0.0.0/0
Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server
Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server
Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

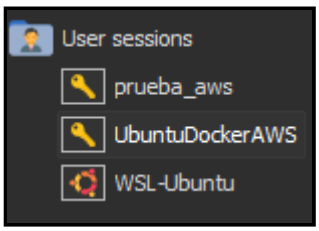

Summary
Number of instances Info
1
Software Image (AMI)
Canonical, Ubuntu, 22.04 LTS, ...read more
ami-00874d747dde814fa
Virtual server type (instance type)
t2.micro
Firewall (security group)
New security group
Storage (volumes)
1 volume(s) - 8 GiB
Cancel
Launch instance

2. Use this EC2 instance to do the following tasks:

a. Access to this instance using SSH client. Create a new profile.

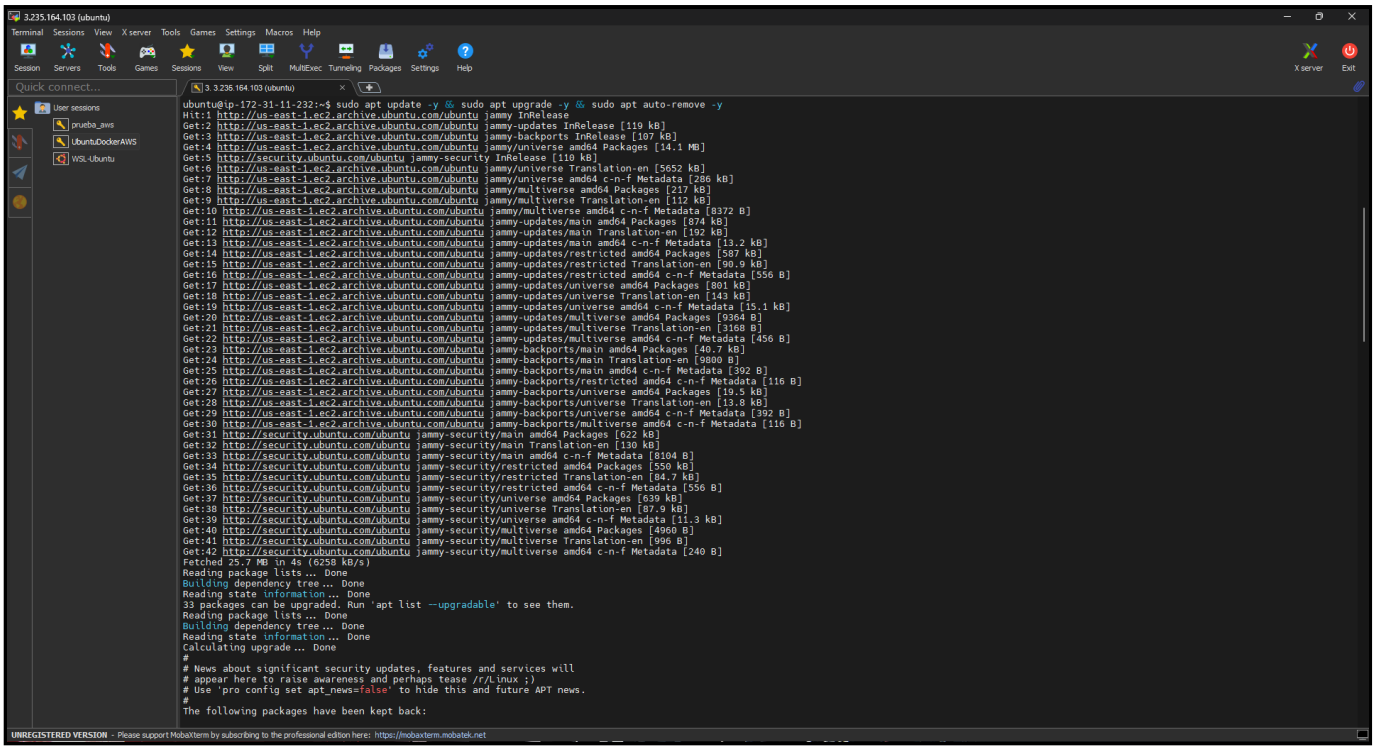
First, I create a new session and configure it to use my new instance's host and the default user "ubuntu". Also, I downloaded the "PPK" file, since I configured it to use a vockey.





b. Update the Ubuntu packages.

```
sudo apt update -y && sudo apt upgrade -y && sudo apt auto-remove -y
```



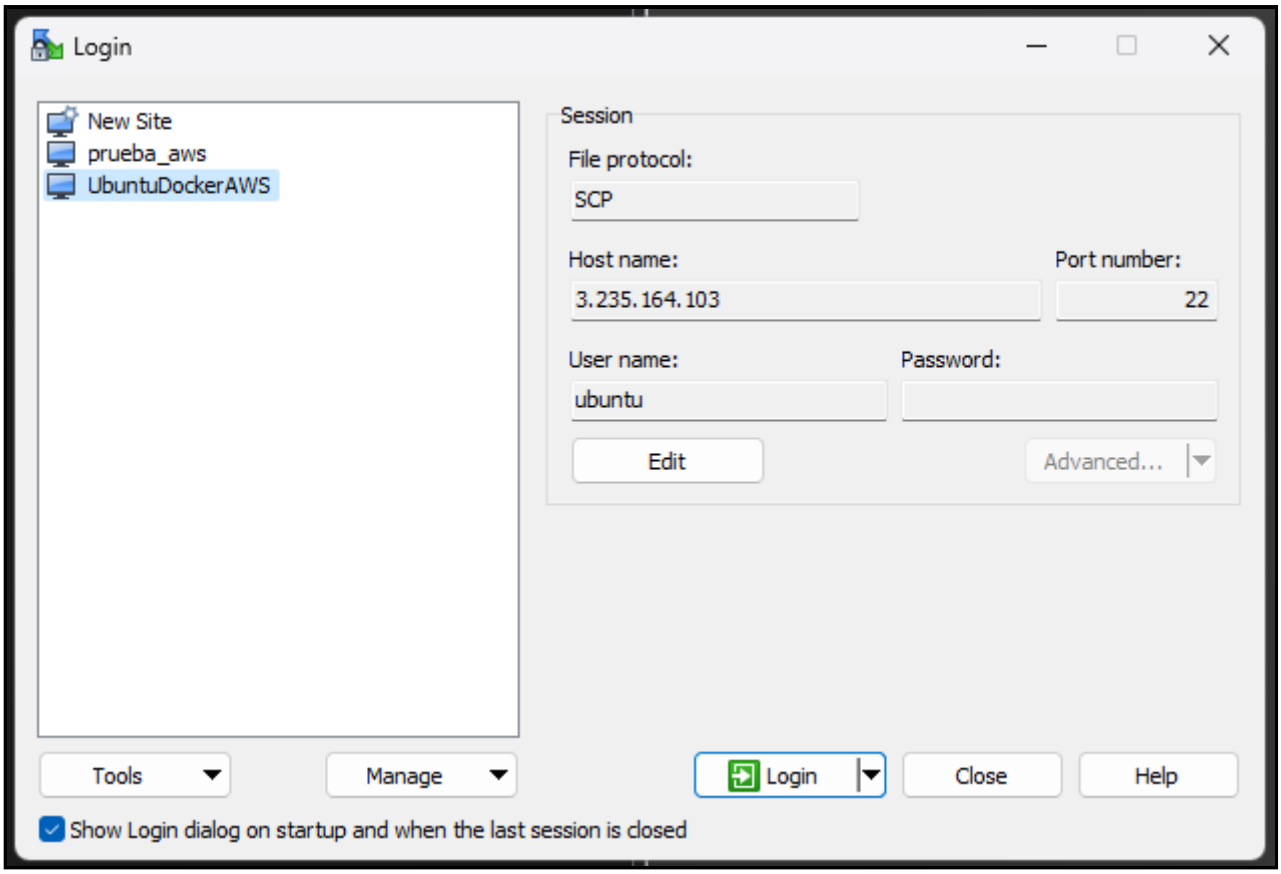
c. Install Docker in this server.

```
sudo apt install docker.io
```

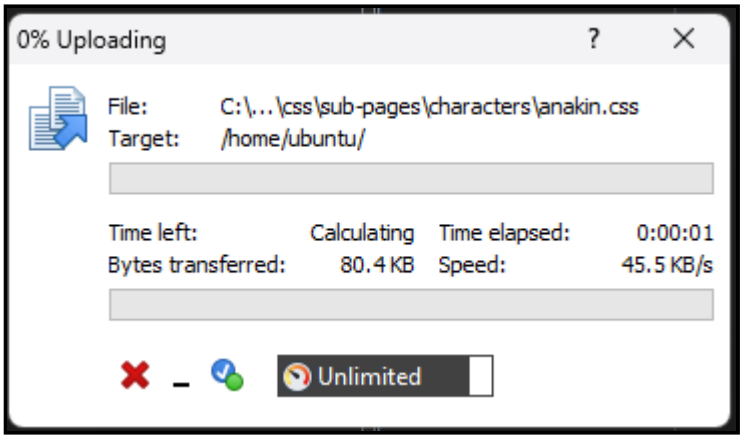
```
ubuntu@ip-172-31-11-232:~$ sudo apt install docker.io
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
The following additional packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools cgroupfs-mount | cgroup-lite debootstrap docker-doc rinse zfs-fuse | zfsutils
The following NEW packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base docker.io pigz runc ubuntu-fan
0 upgraded, 8 newly installed, 0 to remove and 6 not upgraded.
Need to get 66.8 MB of archives.
After this operation, 287 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 pigz amd64 2.6-1 [63.6 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 bridge-utils amd64 1.7-1ubuntu3 [34.4 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 runc amd64 1.1.0-0ubuntu1.1 [4242 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 containerd amd64 1.5.9-0ubuntu3.1 [28.1 MB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 dns-root-data all 2021011101 [5256 B]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 dnsmasq-base amd64 2.86-1.1ubuntu0.1 [354 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 docker.io amd64 20.10.12-0ubuntu4 [34.0 MB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 ubuntu-fan all 0.12.16 [35.2 kB]
Fetched 66.8 MB in 2s (40.3 MB/s)
Preconfiguring packages ...
Selecting previously unselected package pigz.
(Reading database ... 63571 files and directories currently installed.)
Preparing to unpack .../0-pigz_2.6-1_amd64.deb ...
Unpacking pigz (2.6-1) ...
Selecting previously unselected package bridge-utils.
Preparing to unpack .../1-bridge-utils_1.7-1ubuntu3_amd64.deb ...
Unpacking bridge-utils (1.7-1ubuntu3) ...
Selecting previously unselected package runc.
Preparing to unpack .../2-runc_1.1.0-0ubuntu1.1_amd64.deb ...
Unpacking runc (1.1.0-0ubuntu1.1) ...
Selecting previously unselected package containerd.
Preparing to unpack .../3-containerd_1.5.9-0ubuntu3.1_amd64.deb ...
Unpacking containerd (1.5.9-0ubuntu3.1) ...
Selecting previously unselected package dns-root-data.
Preparing to unpack .../4-dns-root-data_2021011101_all.deb ...
Unpacking dns-root-data (2021011101) ...
Selecting previously unselected package dnsmasq-base.
Preparing to unpack .../5-dnsmasq-base_2.86-1.1ubuntu0.1_amd64.deb ...
Unpacking dnsmasq-base (2.86-1.1ubuntu0.1) ...
Selecting previously unselected package docker.io.
Preparing to unpack .../6-docker.io_20.10.12-0ubuntu4_amd64.deb ...
Unpacking docker.io (20.10.12-0ubuntu4) ...
Selecting previously unselected package ubuntu-fan.
Preparing to unpack .../7-ubuntu-fan_0.12.16_all.deb ...
Unpacking ubuntu-fan (0.12.16) ...
Setting up dnsmasq-base (2.86-1.1ubuntu0.1) ...
Setting up runc (1.1.0-0ubuntu1.1) ...
Setting up dns-root-data (2021011101) ...
Setting up bridge-utils (1.7-1ubuntu3) ...
Setting up pigz (2.6-1) ...
Setting up containerd (1.5.9-0ubuntu3.1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /lib/systemd/system/containerd.service.
```

d. Use WinSCP to transfer the content of a static web page to home directory.

I create a new SCP connection with the host in WinSCP.



And now I can transfer my static project to the server home directory.

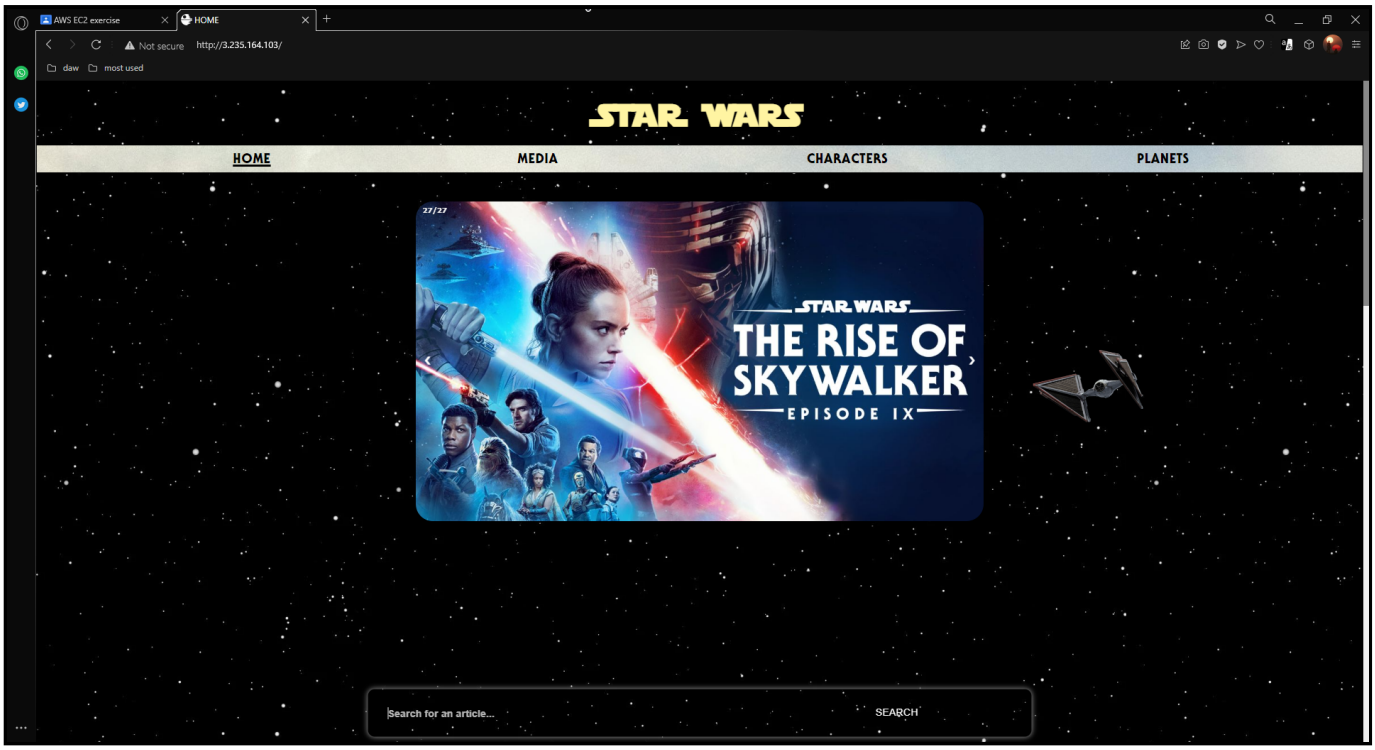


e. Run a "nginx:1.22.1" docker to publish this static web page (port 80).

```
sudo docker run -d -p 80:80 -v ~/fin_1er_trimestre:/usr/share/nginx/html
nginx:1.22.1
```

```
ubuntu@ip-172-31-11-232:~$ sudo docker run -d -p 80:80 -v ~/fin_1er_trimestre:/usr/share/nginx/html nginx:1.22.1
Unable to find image 'nginx:1.22.1' locally
1.22.1: Pulling from library/nginx
bb263680fed1: Pull complete
c13df232596d: Pull complete
2b1f6cfd11a1: Pull complete
cef27009be5d: Pull complete
9317a4486630: Pull complete
023932eaae7e: Pull complete
Digest: sha256:6b9d1c6e9826964d65710927416b526ec5939545e66ad42326ccb338880f2c5d
Status: Downloaded newer image for nginx:1.22.1
00e308bf23a5d4e5731d1f58345314df83074606ce599cdffc3c8ea476e37256
```

It is now possible to visit "3.235.164.103:80" and visit the page, located on the docker nginx server.



3. Now consider you are not using Windows but an Ubuntu console:

a. Access to previous "UbuntuDockerAWS" EC2 instance.

To be able to do this, I first need to download the "PEM" vockey file for the instance.

```
chmod 400 labsuser.pem
```

```
christian@christianms13:/mnt/c/Users/chris/Documents/DAW/.extra/aws_ej1$ chmod 400 labsuser.pem
christian@christianms13:/mnt/c/Users/chris/Documents/DAW/.extra/aws_ej1$ ls -la
total 8
drwxrwxrwx 1 christian christian  512 Feb  9 17:57 .
drwxrwxrwx 1 christian christian  512 Feb  9 17:24 ..
-r-xr-xr-x 1 christian christian 1678 Feb  9 17:57 labsuser.pem
-rwxrwxrwx 1 christian christian 1438 Feb  9 17:24 labsuser.ppk
```

Once I have this file, I need to change it's permissions:

```
ssh -i "labuser.pem" ubuntu@ec2-3-235-164-103.compute-1.amazonaws.com
```

```
christian@christianms13:~$ ssh -i "vockey.pem" ubuntu@ec2-3-235-164-103.compute-1.amazonaws.com
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-1028-aws x86_64)

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

  System information as of Thu Feb  9 17:40:54 UTC 2023

  System load:  0.0               Processes:               106
  Usage of /:   29.1% of 7.57GB   Users logged in:         0
  Memory usage: 27%               IPv4 address for docker0: 172.17.0.1
  Swap usage:   0%                IPv4 address for eth0:    172.31.11.232


0 updates can be applied immediately.


Last login: Thu Feb  9 16:39:40 2023 from 46.37.76.3
ubuntu@ip-172-31-11-232:~$
```

b. Copy using "scp" command a directory with another static web page to the "UbuntuDockerAWS" EC2 instance.

First of all, since I am going tu use the same web page, I need to create an additional directory in the user home of "UbuntuDockerAWS" to locate the page in there.

```
mkdir auxiliar
```

Now that I have a place to copy the page in, I can use the SCP command:

```

```