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

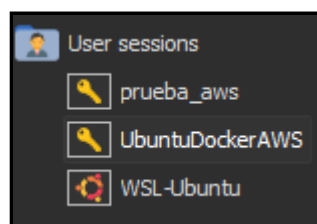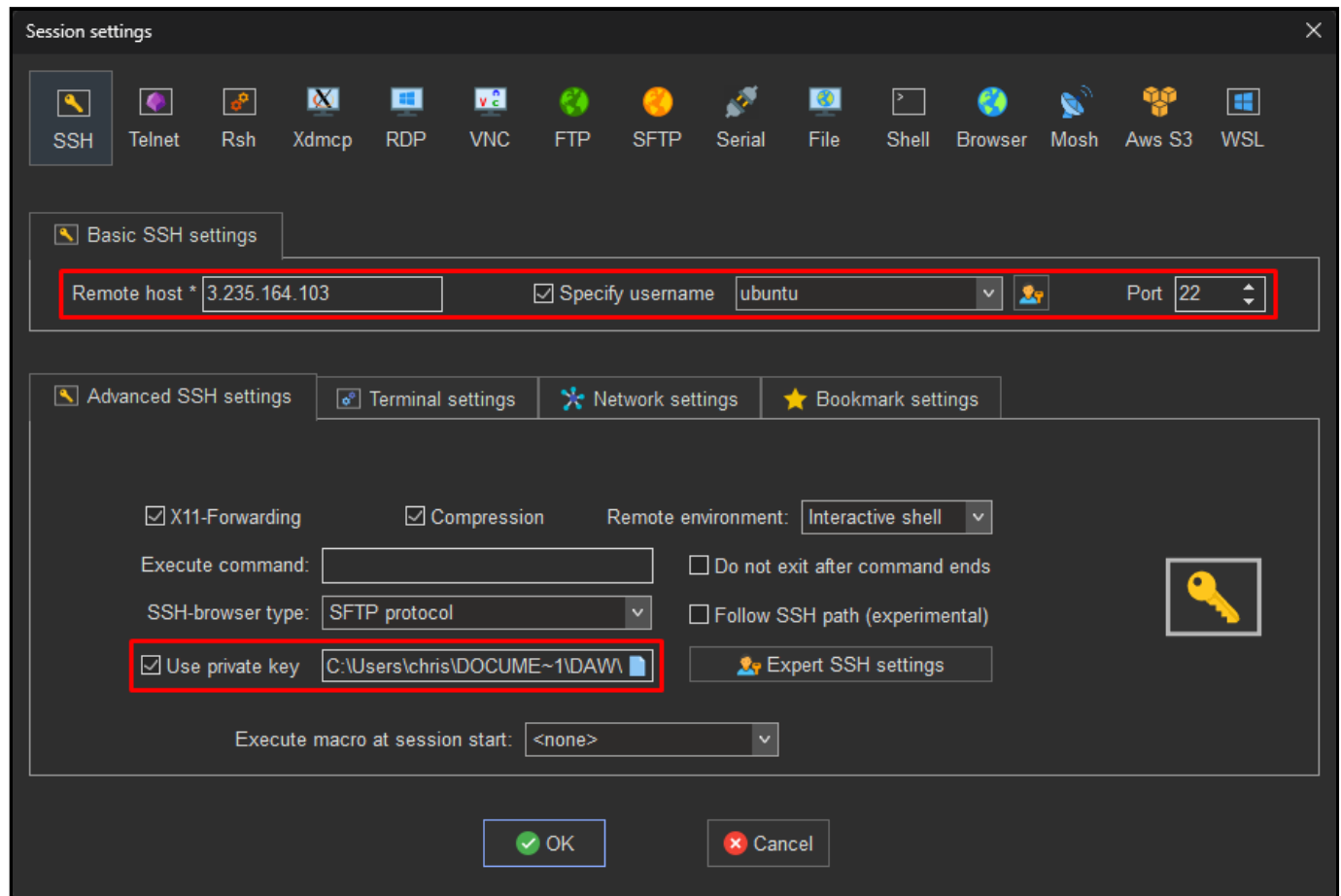Storage (volumes)
1 volume(s) - 8 GiB

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.
```