Digital Forensics

Lab 9: Cloud Forensics

The goal of this lab is to learn about and practice with cloud forensics.

Tutorials / Videos

- DFIRScience
    - Explore and search for videos on Cloud forensics (e.g., AWS).
    - https://www.youtube.com/@DFIRScience/videos

Cloud Forensics Tutorials & Resources

- Artifacts of Google Drive Usage in Windows
- Cloud Storage Acquisition from Endpoint Devices
- Advanced Audit in Microsoft 365 (requires admin rights). You may also check this SANS methodology and this Investigating Office 365 Resource.
- AWS Digital Forensics (official page). You can also check this tutorial and video on AWS Incident Response and Forensics.
- Live forensics demo (includes attack from Kali, memory capture and analysis on Windows VM): https://www.youtube.com/watch?v=3FDmtq55QoI

Cloud Forensics Tools

- Industry Roundup Cloud Forensics. This article from Forensic Focus lists a number of commercial cloud forensic tools.
- Cado Cloud Collector is a free tool for acquiring images of AWS EC2 instances. Check the other free tools under Community tab.
- Kumodd is a Python tool for forensic acquisition of files/metadata from a Google Drive account (see Roussev et al., 2016 research paper on Moodle).
- Cloud Forensics Utils is a Google Python library for collecting info from GCP, Azure and AWS. Used by frameworks such as Turbinia and DFTimewolf.
- Velociraptor is a DFIR tool that that can be deployed in the cloud to monitor endpoints, collect information, etc.
- MemProcFS is a tool that can also be used for remote memory acquisition of Windows systems https://github.com/ufrisk/LeechCore/wiki/Device_Remote.
- Margarita Shotgun is a Python tool for remote memory acquisition of AWS EC2 Linux instances. Part of the ThreatResponse Incident Response Toolkit.

Practical Tasks

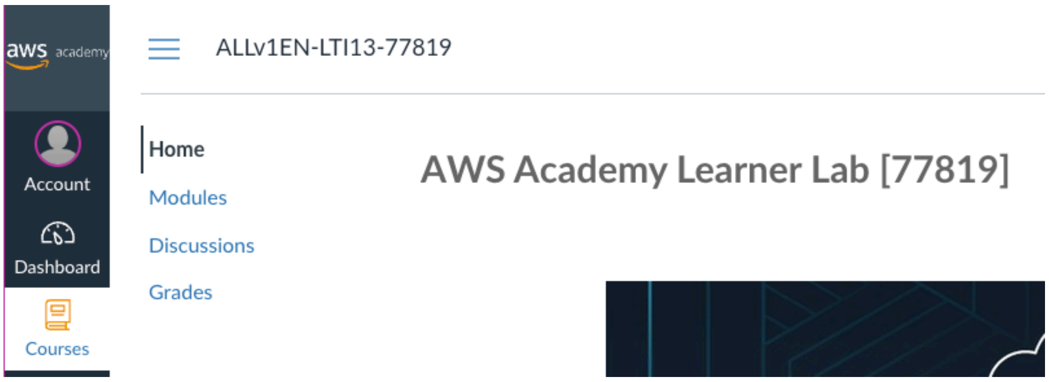
- For the lab practice we will capture forensic evidence from AWS EC2 instances.
- Kali Pre-requisite: Setup a Kali Linux VM on your laptop (I assume you already did that for the Penetration Testing module).
- The below steps show you how to:
    - Create a key pair and launch an AWS EC2 Linux instance.
    - Connect to the EC2 instance using SSH and create a LiME kernel object.
    - Install Margarita Shotgun in a local Kali Linux VM running in VirtualBox.
    - Create a memory dump of the remote EC2 instance.

Practical Tasks Steps and Screenshots

AWS Academy

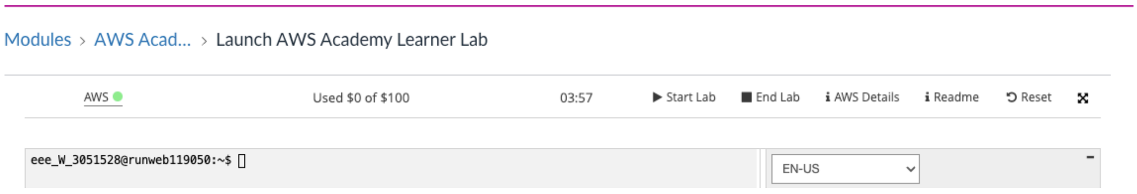
The lecturer has applied for an AWS Academy classroom:

- Please check your student e-mail account for an e-mail from AWS Academy with the link and instructions to join the classroom.
- Each student receives $100 credit, but learn to manage it efficiently.
- Take some time to explore the resources and guidelines under the Modules.



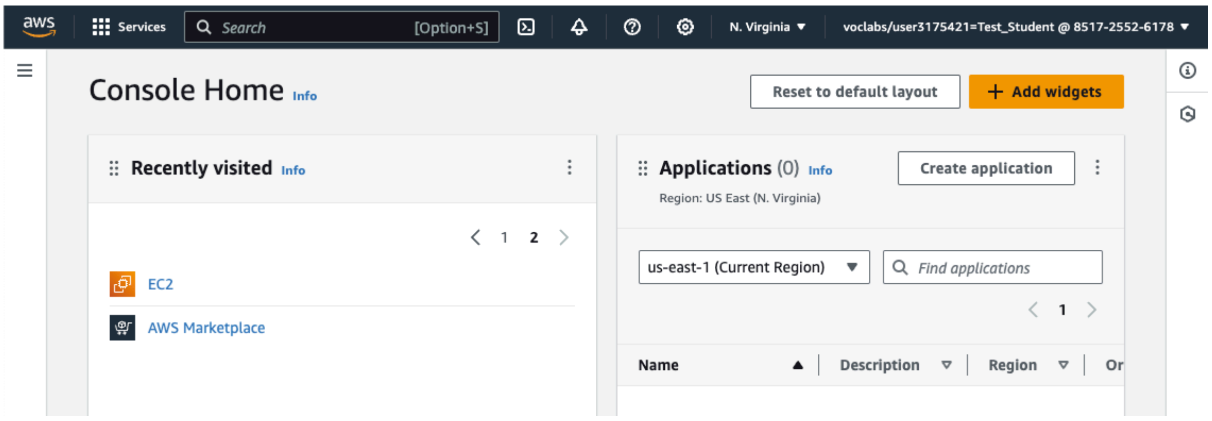
AWS Academy Learner Lab [77819]

- To start the lab navigate to:
  Modules → Launch AWS Academy Learner Lab → Start Lab
- The session lasts 4 hours, but if you need more time just click again on Start Lab to reset the timer.



```
eee_W_3051528@runweb119050:~$ 
```

- Launch the AWS web console by clicking on the AWS (see green circle above).
- Take some time to explore the AWS web console and services (but do not start services / resources that you do not plan to use as you will waste your credit).



IMPORTANT:

Some of the screenshots in the tutorial below might look slightly different from what you may see (as the AWS web console and Kali Linux are being updated over the years). The tutorial was tested multiple times over the years and it worked, just make sure you follow the steps carefully.

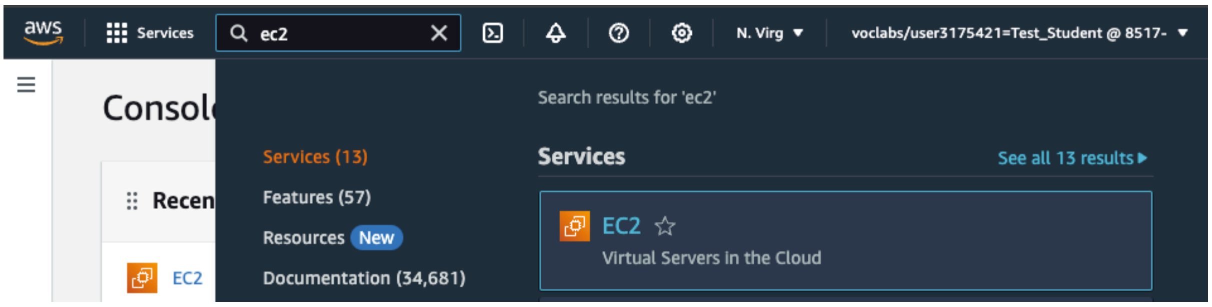
Part 1: Setup an AWS EC2 Key Pair

You must have an Amazon Elastic Compute Cloud (Amazon EC2) key pair to connect to the nodes in your cluster over a secure channel using the Secure Shell (SSH) protocol.

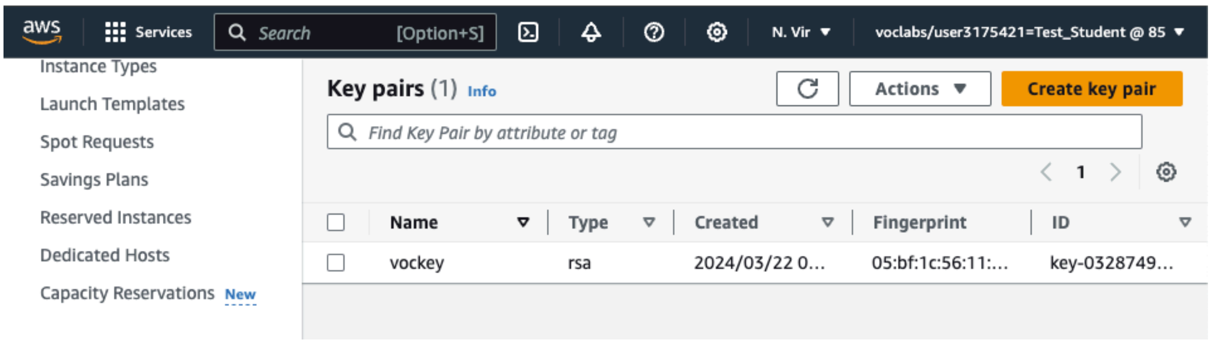
More detailed guide: Creating Your Key Pair Using Amazon EC2 in the *Amazon EC2 User Guide for Linux Instances*.

Simple steps:

1) Open the AWS web console and navigate to the EC2 service.

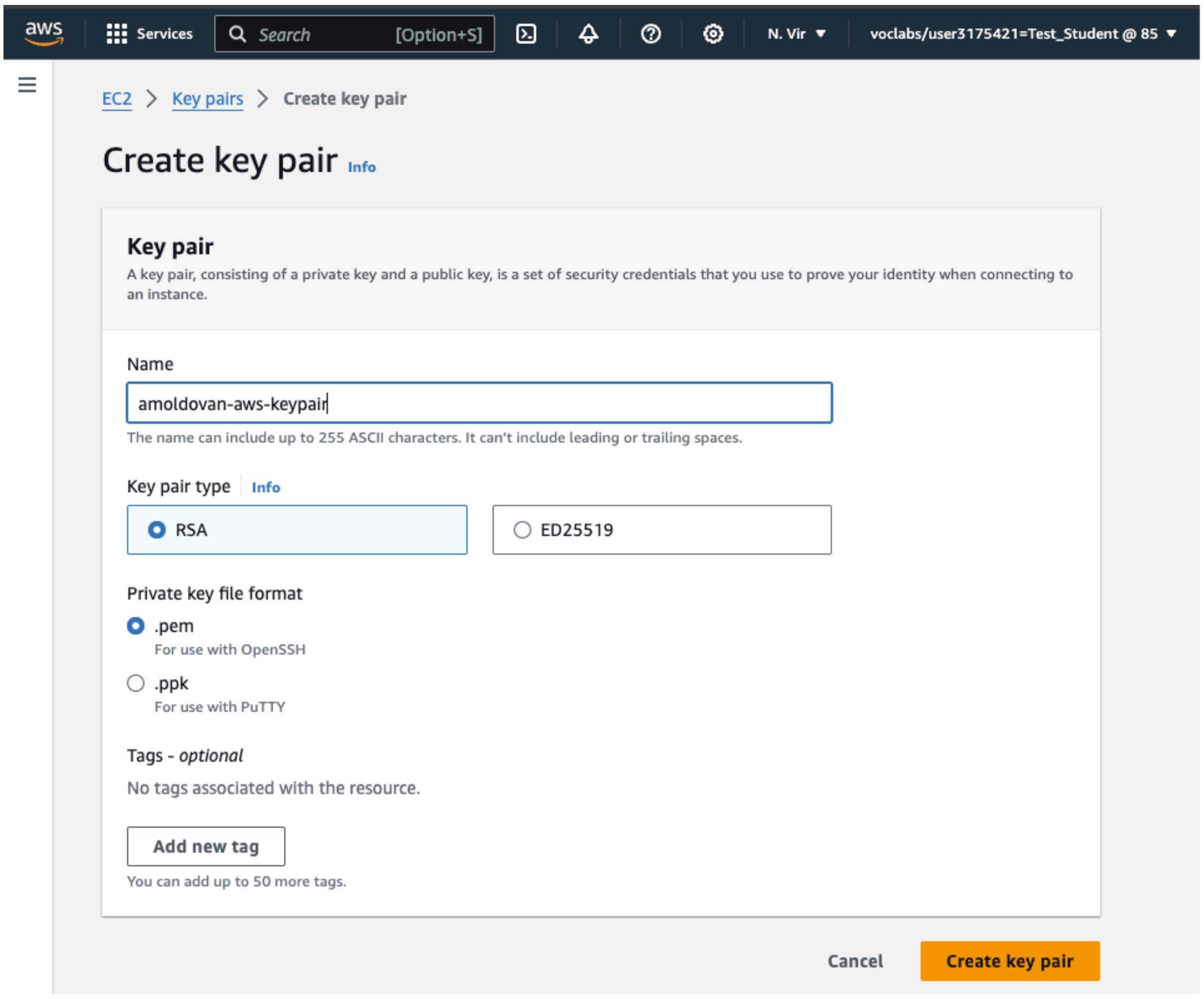


2) Navigate to 'Network & Security' → 'Key Pairs' → 'Create key pair'



Name	Type	Created	Fingerprint	ID
vockey	rsa	2024/03/22 0...	05:bf:1c:56:11:...	key-0328749...

3) Specify a meaningful name starting with your **student number**, select RSA and **.pem** file format (we can use it on Linux, Windows & MacOS), and click 'Create key pair'.


aws
Services
Search [Option+S]
N. Vir
voclabs/user3175421=Test_Student @ 85
EC2 > Key pairs > Create key pair

Create key pair Info

Key pair

A key pair, consisting of a private key and a public key, is a set of security credentials that you use to prove your identity when connecting to an instance.

Name

amoldovan-aws-keypair

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type Info

[✓ RSA] ED25519

Private key file format

[✓ .pem] For use with OpenSSH

.ppk For use with PuTTY

Tags - *optional*

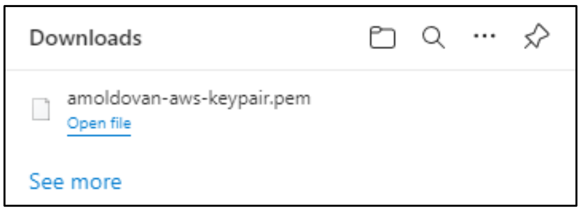
No tags associated with the resource.

Add new tag

You can add up to 50 more tags.

Cancel Create key pair

4) The key pair file will be downloaded automatically. Move it to your working folder on your local PC (e.g., …/DIGFOR/AWS), and keep it safe.



Downloads

amoldovan-aws-keypair.pem
Open file

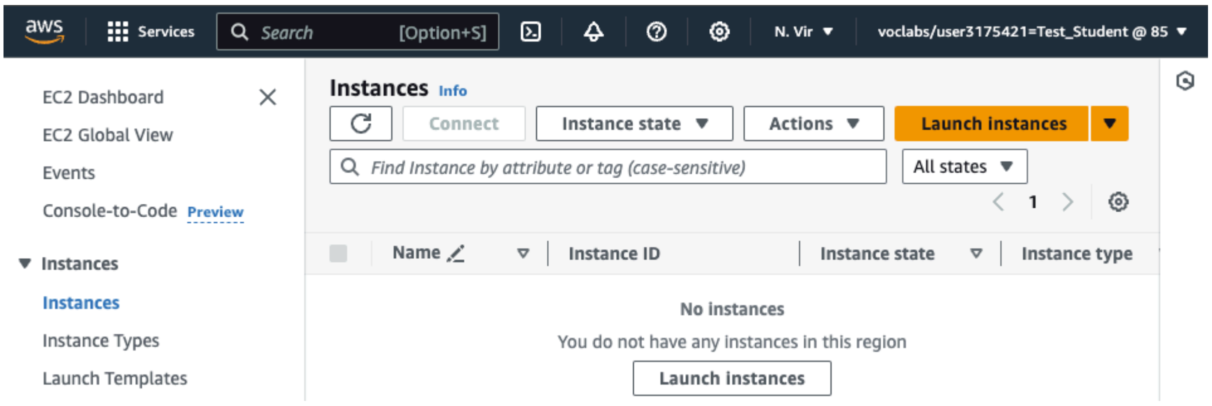
See more

5) Note that the key pair is specific to the selected region (e.g., N Virginia). If you change to a different region, you will have to re-create or import the key pair in that region.

Part 2: Launch an EC2 Linux Instance

1) In the EC2 service navigate to ‘Instances’ → ‘Launch instances’

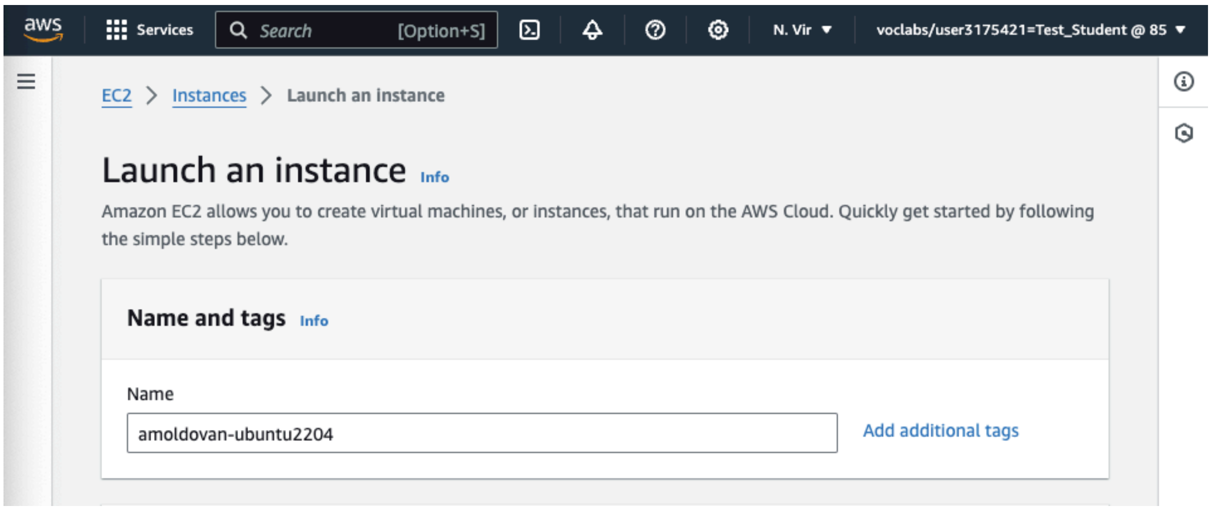
NOTE: If you face difficulties to launch an instance try to change the region (e.g., from Ireland to something else like London or N. Virginia).



No instances

You do not have any instances in this region

2) Give a meaningful name starting with your **student number**, and followed by the Linux OS version that you select.

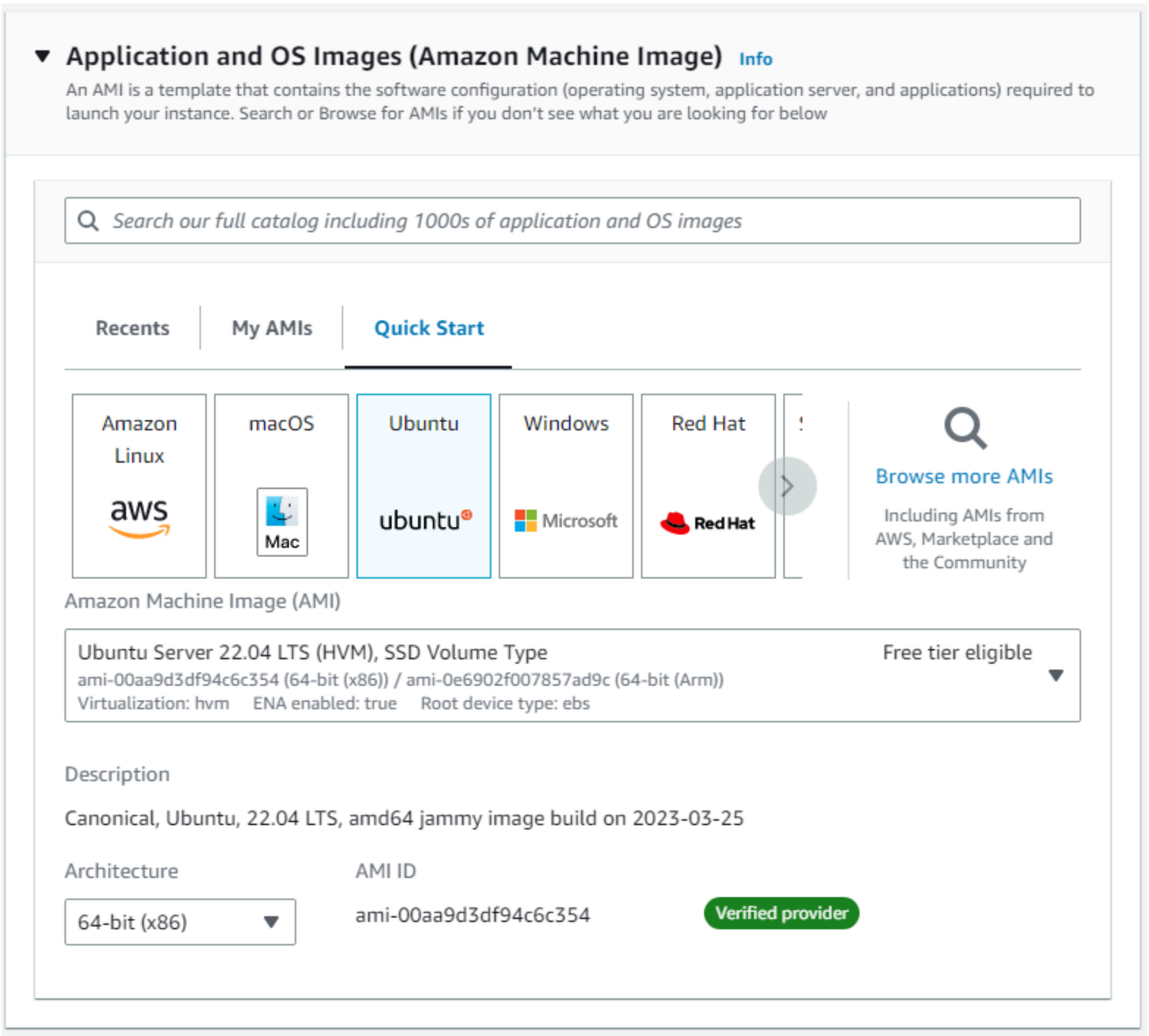


Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name: amoldovan-ubuntu2204

3) Select the Linux AMI (Amazon Machine Image).

NOTE: In this tutorial I try Ubuntu 22.04 LTS, but in the past I successfully acquired memory for other Linux OSs (e.g., Amazon Linux 2, Ubuntu 18.04, Ubuntu 20.04).



Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Amazon Machine Image (AMI)

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type — Free tier eligible
ami-00aa9d3df94c6c354 (64-bit (x86)) / ami-0e6902f007857ad9c (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Description

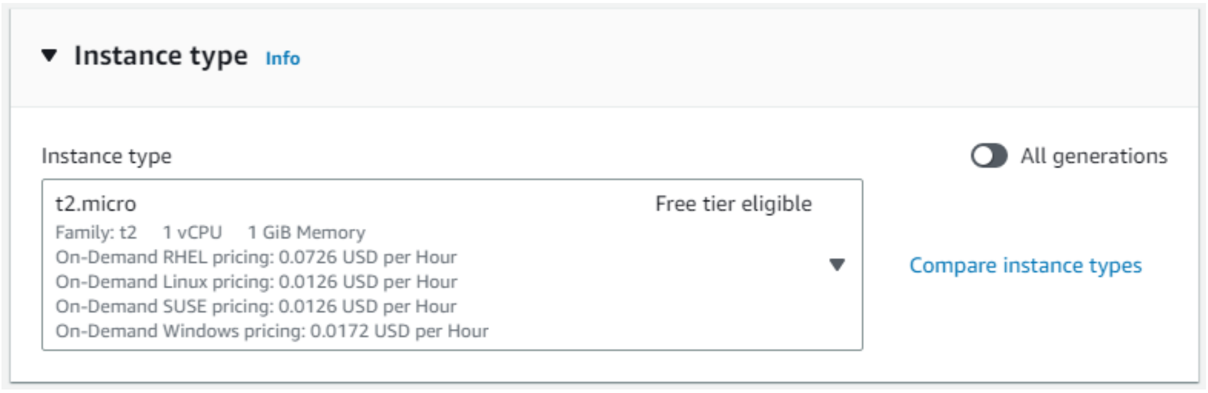
Canonical, Ubuntu, 22.04 LTS, amd64 jammy image build on 2023-03-25

Architecture: 64-bit (x86)

AMI ID: ami-00aa9d3df94c6c354 Verified provider

4) Select the instance type as **t2.micro** (as it is free tier eligible, and only has 1 GB of RAM so the memory acquisition and analysis would be faster).

NOTE: Check the pricing for better understanding: https://aws.amazon.com/ec2/pricing/

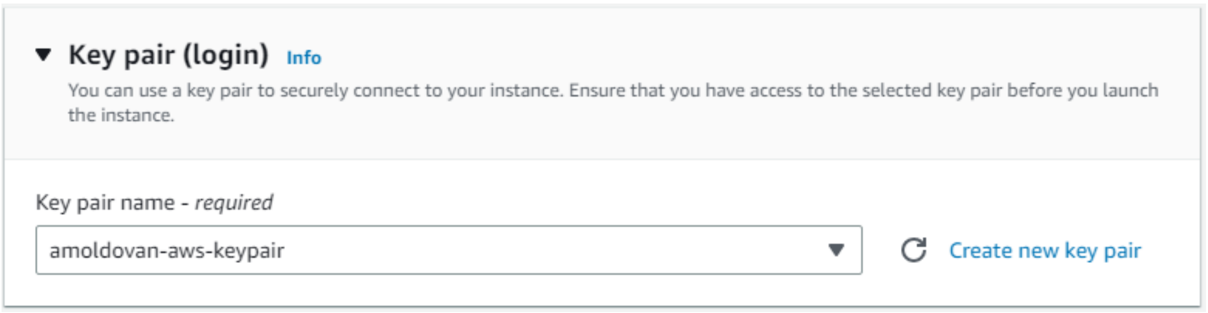


Instance type Info

Instance type

t2.micro — Free tier eligible
Family: t2 1 vCPU 1 GiB Memory
On-Demand RHEL pricing: 0.0726 USD per Hour
On-Demand Linux pricing: 0.0126 USD per Hour
On-Demand SUSE pricing: 0.0126 USD per Hour
On-Demand Windows pricing: 0.0172 USD per Hour

5) Select the EC2 key pair that you have created.



▼ **Key pair (login)** Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

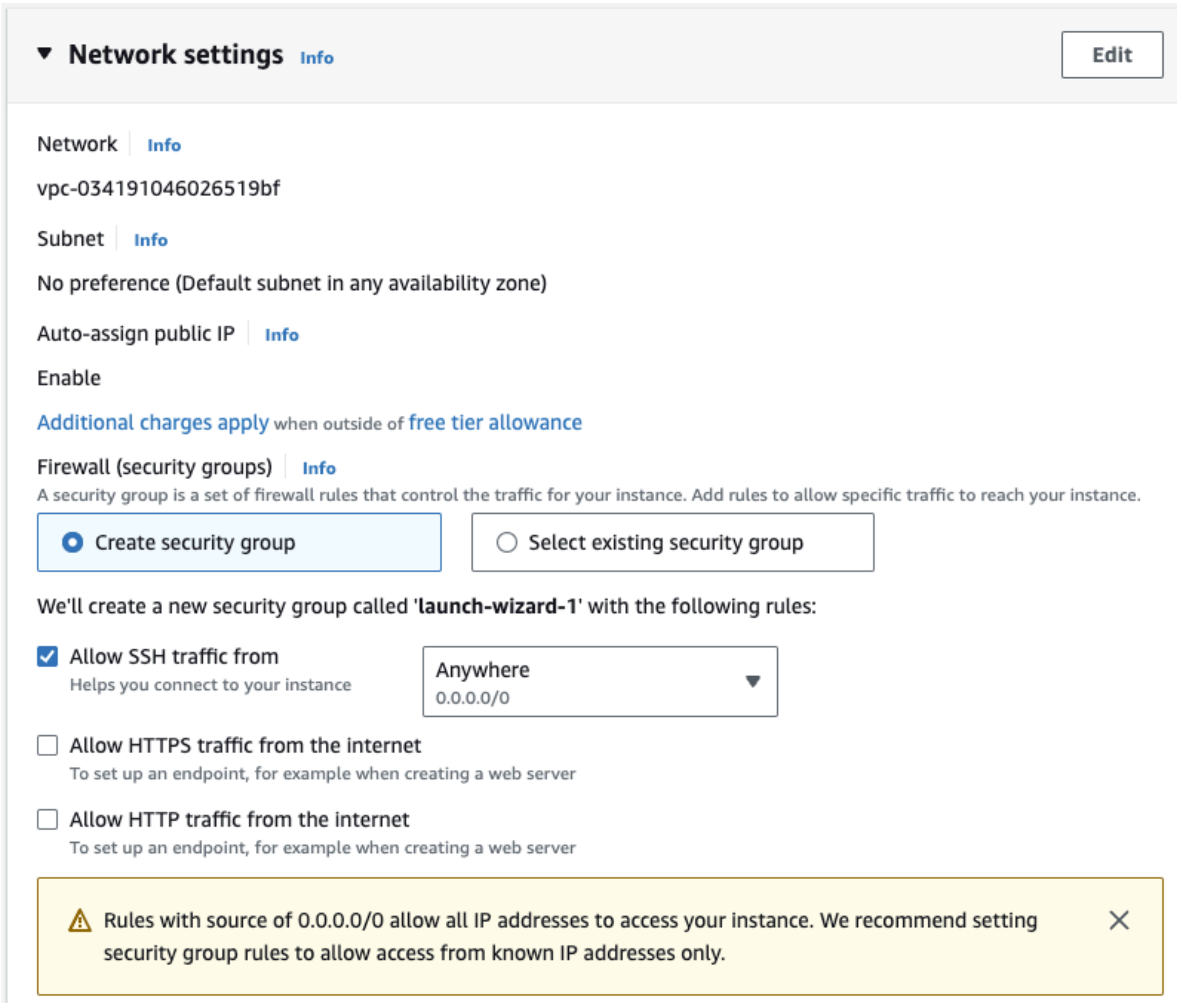
Key pair name - *required*

amoldovan-aws-keypair ▼

Create new key pair

6) Edit the 'Network settings', leave all settings as default except:

- 'Allow SSH traffic from' → Select **'My IP'**
  (**IMPORTANT**: Do not leave Anywhere as it is a security risk.)



▼ **Network settings** Info

Edit

Network Info

vpc-034191046026519bf

Subnet Info

No preference (Default subnet in any availability zone)

Auto-assign public IP Info

Enable

Additional charges apply when outside of free tier allowance

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-1**' with the following rules:

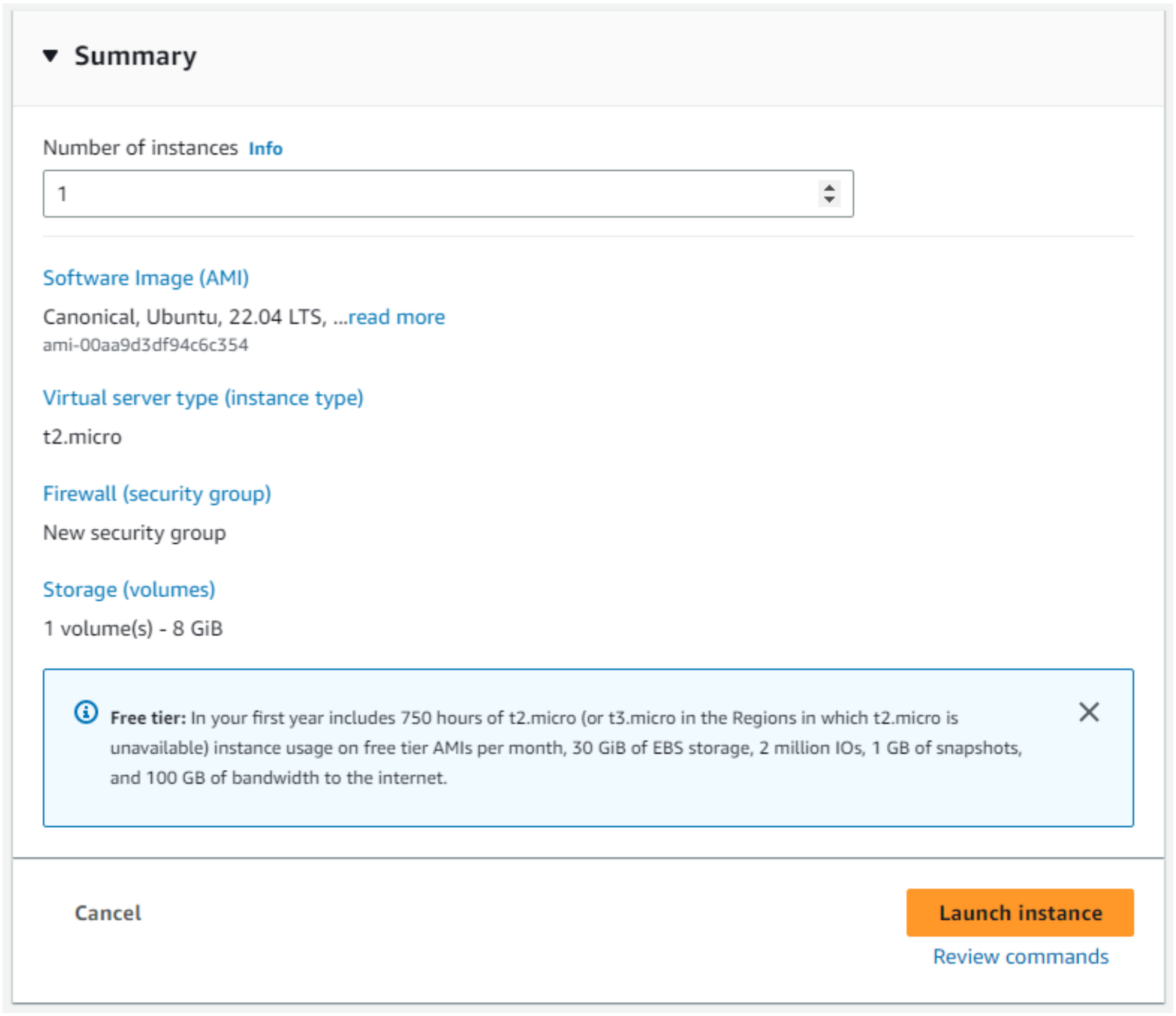
☑ Allow SSH traffic from — Anywhere 0.0.0.0/0
Helps you connect to your instance

☐ 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.

7) Leave the rest of the settings as they are. Scroll down and click on 'Launch instance'



▼ Summary

Number of instances Info

1

Software Image (AMI)

Canonical, Ubuntu, 22.04 LTS, ...read more

ami-00aa9d3df94c6c354

Virtual server type (instance type)

t2.micro

Firewall (security group)

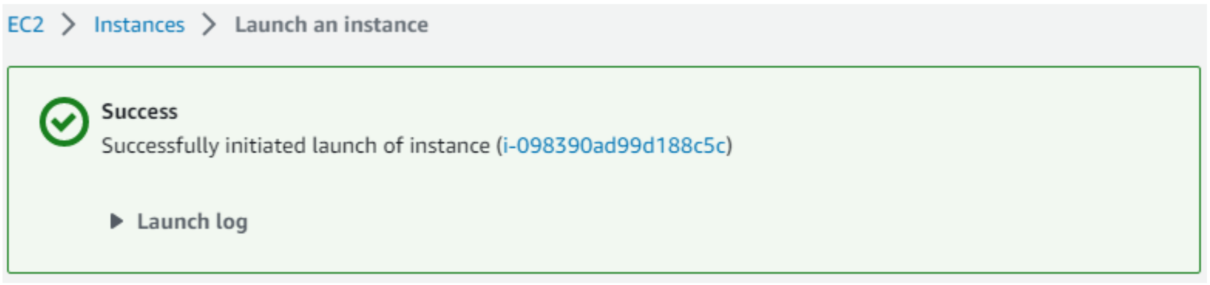
New security group

Storage (volumes)

1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

8) If everything works fine, you should see a success message.

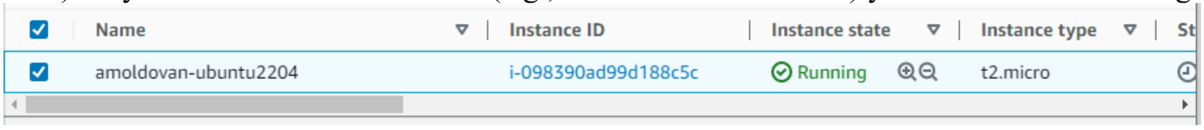


EC2 > Instances > Launch an instance

Success

Successfully initiated launch of instance (i-098390ad99d188c5c)

9) If you click on the instance ID (e.g., i-098390ad99d188c5c) you should see it running.

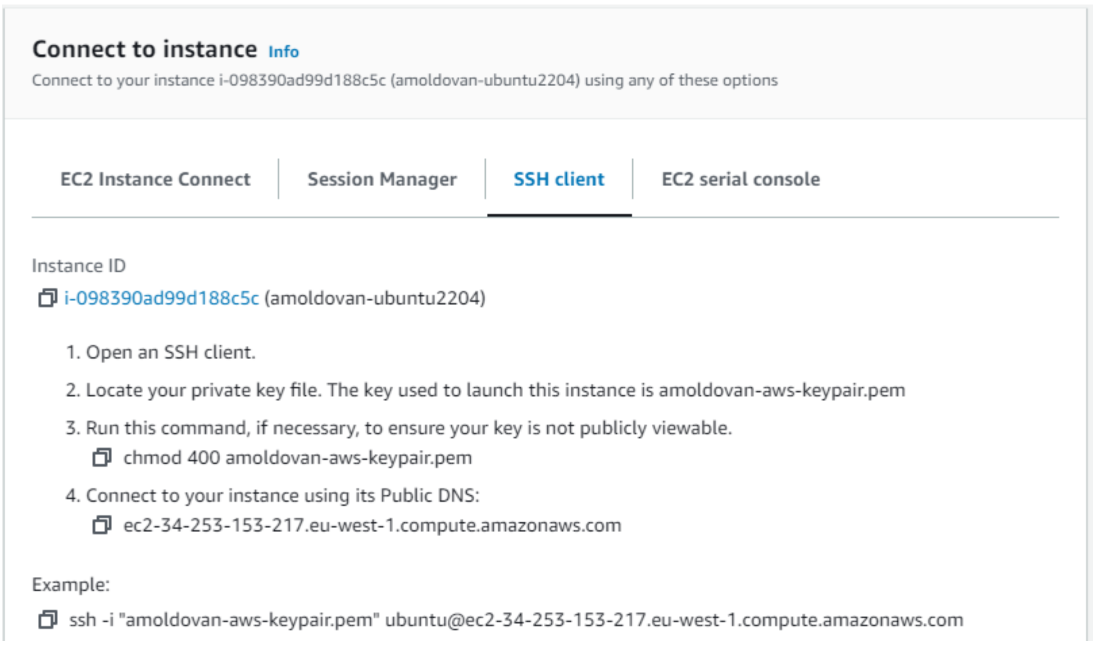


Name	Instance ID	Instance state	Instance type	St
amoldovan-ubuntu2204	i-098390ad99d188c5c	Running	t2.micro	

IMPORTANT: While we work with t2.micro that is free tier eligible, it is good practice to terminate resources that you do not use. Make sure to terminate the instance when you finished with the practice. You can create another instance later, if needed.

Part 3: Connect to the EC2 instance using SSH and create a LiME kernel object

1) To connect via SSH to the EC2 instance, click on the instance ID (see step 9 above) → then click on Connect button → Check the instructions how to.



Connect to instance Info

Connect to your instance i-098390ad99d188c5c (amoldovan-ubuntu2204) using any of these options

Instance ID

i-098390ad99d188c5c (amoldovan-ubuntu2204)

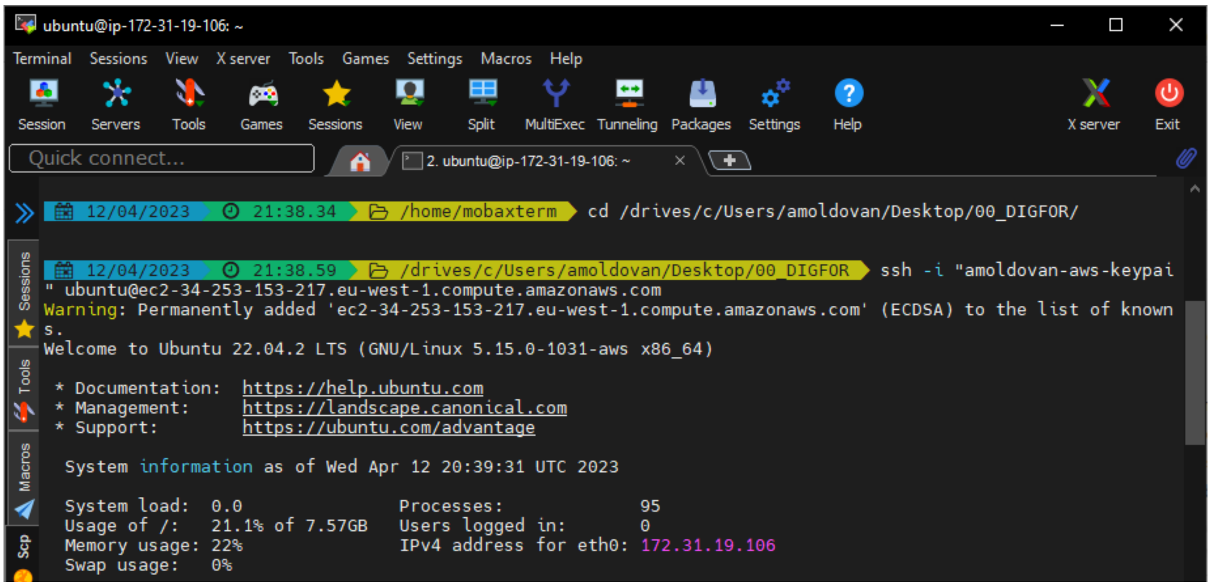
1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is amoldovan-aws-keypair.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
   `chmod 400 amoldovan-aws-keypair.pem`
4. Connect to your instance using its Public DNS:
   `ec2-34-253-153-217.eu-west-1.compute.amazonaws.com`

Example:

```
ssh -i "amoldovan-aws-keypair.pem" ubuntu@ec2-34-253-153-217.eu-west-1.compute.amazonaws.com
```

OPTIONAL: If you want to try it, you should be able to connect to the EC2 instance and complete this part from your laptop PC host OS also. But note the following:

- If your PC runs MacOS / Linux, you just need to open a Terminal, use **cd** to navigate to the folder with your key pair, then run the **chmod** and **ssh** commands.
- If your PC runs Windows, you may face difficulties with changing permissions for the key pair (e.g., you cannot use the chmod command in Windows CMD). The easy solution is to install a software that includes a Linux Bash such as:
  - Git for Windows: https://git-scm.com/
  - MobaXterm (see below screenshot): https://mobaxterm.mobatek.net/

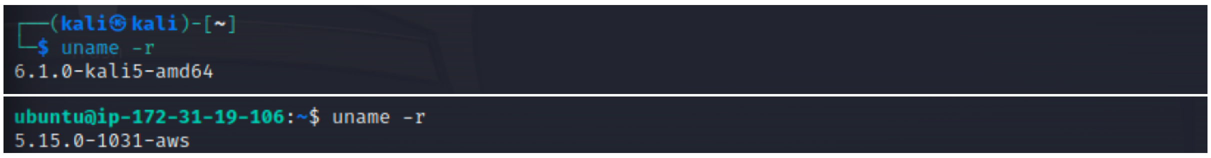


ubuntu@ip-172-31-19-106: ~
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help X server Exit
Quick connect...
2. ubuntu@ip-172-31-19-106: ~
Sessions Tools Macros Sftp

```
 12/04/2023  21:38.34  /home/mobaxterm  cd /drives/c/Users/amoldovan/Desktop/00_DIGFOR/

 12/04/2023  21:38.59  /drives/c/Users/amoldovan/Desktop/00_DIGFOR  ssh -i "amoldovan-aws-keypai
" ubuntu@ec2-34-253-153-217.eu-west-1.compute.amazonaws.com
Warning: Permanently added 'ec2-34-253-153-217.eu-west-1.compute.amazonaws.com' (ECDSA) to the list of known
s.
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-1031-aws x86_64)

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

  System information as of Wed Apr 12 20:39:31 UTC 2023

  System load:  0.0               Processes:             95
  Usage of /:   21.1% of 7.57GB   Users logged in:       0
  Memory usage: 22%               IPv4 address for eth0: 172.31.19.106
  Swap usage:   0%
```

NOTE: To keep it simple we will connect to the AWS EC2 target from inside a local Kali Linux VM which will act as our forensic VM (I assume you have already set up a Kali VM on your laptop and used it for the Penetration Testing module. If not get a pre-made VM for VirtualBox or VMware from https://www.kali.org/get-kali/).

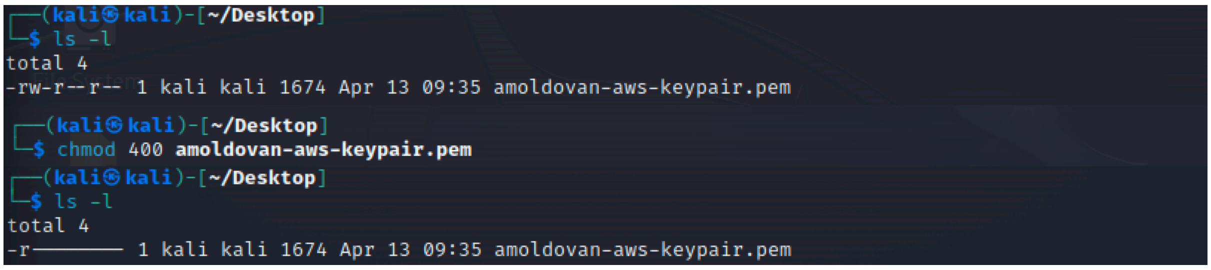
- The rationale is because we can only install Margarita Shotgun on Linux (so while we can do part 3 on Windows, we can only do part 4 on Linux).
- An alternative would be to use another AWS EC2 instance as the forensic VM. If you select the same Linux OS and kernel version as for the target VM, you could build the LiME kernel object on the forensic VM directly.
- Note that in a real DFIR context you should built the LiME kernel objects for all of your Linux cloud instances in advance, so that you are prepared to take memory dumps quickly when needed. That would also avoid running commands on the target and changing its memory state.
- The below screenshots show you the kernel version for the forensic VM (e.g., Kali 2023.1) and the target VM (e.g., Ubuntu 22.04).
- As they are different, we will built the memory profile on the target and then copy it over to the forensic VM.



```
┌──(kali㉿kali)-[~]
└─$ uname -r
6.1.0-kali5-amd64
```

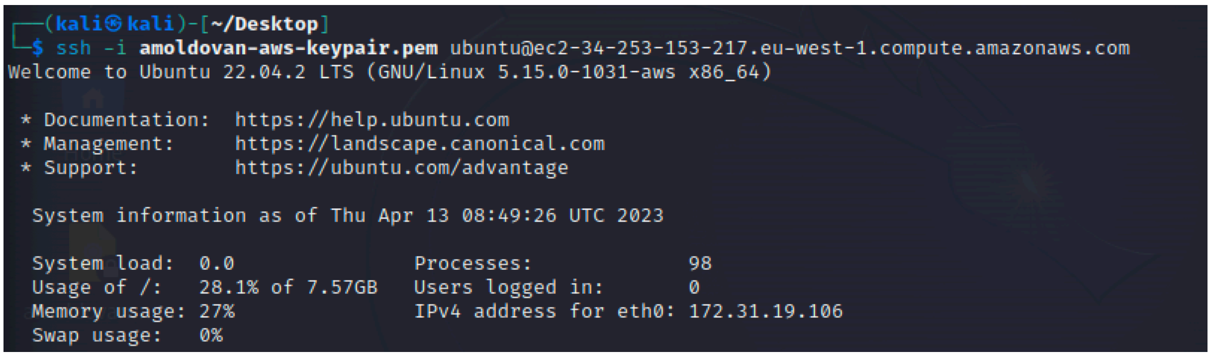
```
ubuntu@ip-172-31-19-106:~$ uname -r
5.15.0-1031-aws
```

2) Remember to copy (e.g., drag and drop) the key pair file to the forensic VM. When you try to connect via ssh, you may get a "WARNING: UNPROTECTED PRIVATE KEY FILE!". You need to restrict permissions using the **chmod** command so that the key pair file cannot be accessed by other users. The below screenshot shows the permissions before and after running the chmod command.



```
┌──(kali㉿kali)-[~/Desktop]
└─$ ls -l
total 4
-rw-r--r-- 1 kali kali 1674 Apr 13 09:35 amoldovan-aws-keypair.pem

┌──(kali㉿kali)-[~/Desktop]
└─$ chmod 400 amoldovan-aws-keypair.pem
┌──(kali㉿kali)-[~/Desktop]
└─$ ls -l
total 4
-r-------- 1 kali kali 1674 Apr 13 09:35 amoldovan-aws-keypair.pem
```

3) Connect to your Linux instance with the ssh command. If successful you should see a welcome message. Can also use the instance public IP instead of domain name.



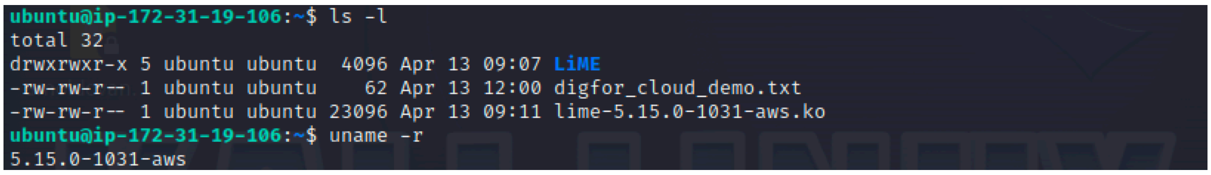
```
┌──(kali㉿kali)-[~/Desktop]
└─$ ssh -i amoldovan-aws-keypair.pem ubuntu@ec2-34-253-153-217.eu-west-1.compute.amazonaws.com
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-1031-aws x86_64)

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

  System information as of Thu Apr 13 08:49:26 UTC 2023

  System load:  0.0               Processes:             98
  Usage of /:   28.1% of 7.57GB   Users logged in:       0
  Memory usage: 27%               IPv4 address for eth0: 172.31.19.106
  Swap usage:   0%
```

4) **OPTIONAL**: Run some Linux commands to understand the target environment.

```
# Check Linux OS and kernel version
cat /etc/os-release
uname -r

# Check the username, path and contents of the user folder
whoami
pwd
ls -l
ls -al

# Check CPU, RAM and Disk
lscpu
cat /proc/meminfo
df -h

# Check the environment PATH
echo $PATH
```

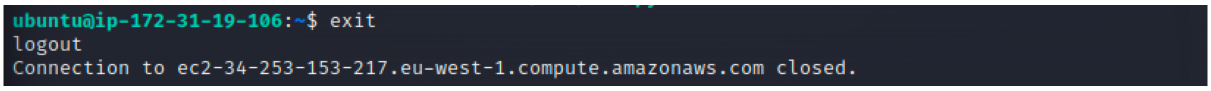
5) Create a LiME kernel object by running the below commands.

```
# Install Linux headers (may already be installed)
sudo apt update
sudo apt install linux-headers-$(uname -r)

# Get LiME and compile it
git clone https://github.com/504ensicsLabs/LiME.git
cd LiME/src/
make

# You may need to install make and gcc
sudo apt install make
sudo apt install gcc
make

# Check the LiME kernel object file .ko
ls -l

# Copy the ko file to the user folder
cp lime-5.4.0-1060-aws.ko ~
cd ~

# Optional: Echo some message to search for during memory analysis
echo "This is just a test file for the DIGFOR cloud forensics demo."
> digfor_cloud_demo.txt
cat digfor_cloud_demo.txt
```



```
ubuntu@ip-172-31-19-106:~$ ls -l
total 32
drwxrwxr-x 5 ubuntu ubuntu  4096 Apr 13 09:07 LiME
-rw-rw-r-- 1 ubuntu ubuntu    62 Apr 13 12:00 digfor_cloud_demo.txt
-rw-rw-r-- 1 ubuntu ubuntu 23096 Apr 13 09:11 lime-5.15.0-1031-aws.ko
ubuntu@ip-172-31-19-106:~$ uname -r
5.15.0-1031-aws
```

6) Close the SSH connection with the exit command or Ctrl+D shortcut.



```
ubuntu@ip-172-31-19-106:~$ exit
logout
Connection to ec2-34-253-153-217.eu-west-1.compute.amazonaws.com closed.
```

Part 4: Install Margarita Shotgun in Kali VM and take a remote memory dump

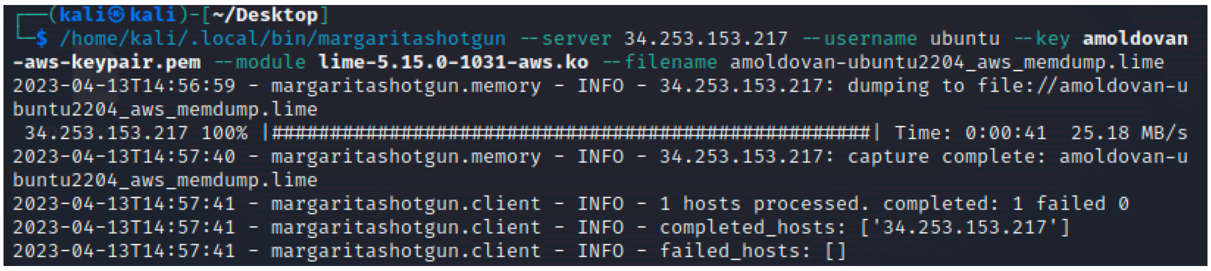
For the detailed documentation see: https://margaritashotgun.readthedocs.io/en/latest/

Check the repository contents (we had to create our LiME kernel object as the repository was not updated in many years): https://threatresponse-lime-modules.s3.amazonaws.com/

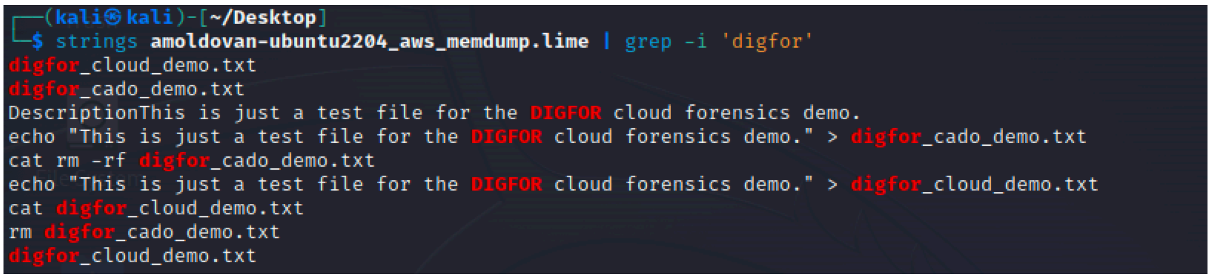
Run the below commands on the local Kali VM to complete this part.

```
# Check Linux OS and kernel version
cat /etc/os-release
uname -r

# Check Python version
python --version
python2 --version
python3 -version
pip --version
pipx --version

# Install pipx if needed
sudo apt update
sudo apt install pipx

# Clone repo
git clone https://github.com/ThreatResponse/margaritashotgun.git
cd margaritashotgun
# Comment a line to avoid install failure with error "use_2to3 is invalid"
sed -i 's/use_2to3/#use_2to3/g' setup.py
# Use pipx (as pip fails with "error: externally-managed-environment")
pipx install .
# Check that Margarita Shotgun was installed
margaritashotgun -h

# Try to capture the memory with the --repository option
# This would fail for newer kernel versions as the repo was not updated
margaritashotgun --server 34.253.153.217 --username ubuntu --key amoldovan-
aws-keypair.pem --repository --filename amoldovan-
ubuntu2204_aws_memdump.lime

# Copy the LiME kernel object from the target VM to the local VM
cd Desktop
scp -i amoldovan-aws-keypair.pem ubuntu@34.253.153.217:/home/ubuntu/*.ko
/home/kali/Desktop
ls -l

# Capture the memory with the created kernel object
margaritashotgun --server 34.253.153.217 --username ubuntu --key amoldovan-
aws-keypair.pem --module lime-5.15.0-1031-aws.ko --filename amoldovan-
ubuntu2204_aws_memdump.lime

# Check the memory dump size and log
ls -l
cat memory-capture.log

# Do some basic analysis of the memory dump using strings
strings amoldovan-ubuntu2204_aws_memdump.lime | grep -i 'Linux version'
strings amoldovan-ubuntu2204_aws_memdump.lime | grep -i 'digfor'
```

Some screenshots of outputs:



```
┌──(kali㉿kali)-[~/Desktop]
└─$ /home/kali/.local/bin/margaritashotgun --server 34.253.153.217 --username ubuntu --key amoldovan
-aws-keypair.pem --module lime-5.15.0-1031-aws.ko --filename amoldovan-ubuntu2204_aws_memdump.lime
2023-04-13T14:56:59 - margaritashotgun.memory - INFO - 34.253.153.217: dumping to file://amoldovan-u
buntu2204_aws_memdump.lime
 34.253.153.217 100% |####################################################| Time: 0:00:41  25.18 MB/s
2023-04-13T14:57:40 - margaritashotgun.memory - INFO - 34.253.153.217: capture complete: amoldovan-u
buntu2204_aws_memdump.lime
2023-04-13T14:57:41 - margaritashotgun.client - INFO - 1 hosts processed. completed: 1 failed 0
2023-04-13T14:57:41 - margaritashotgun.client - INFO - completed_hosts: ['34.253.153.217']
2023-04-13T14:57:41 - margaritashotgun.client - INFO - failed_hosts: []
```

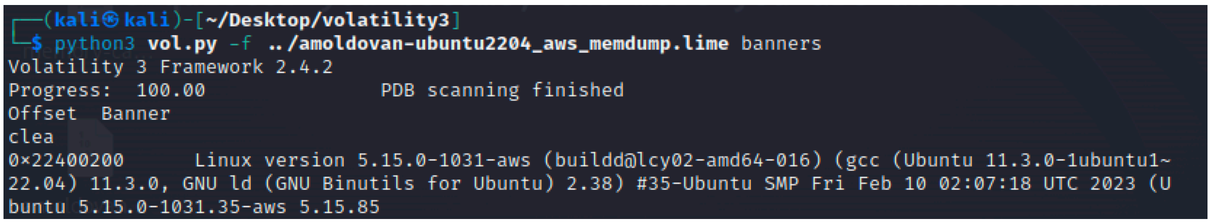


```
┌──(kali㉿kali)-[~/Desktop]
└─$ strings amoldovan-ubuntu2204_aws_memdump.lime | grep -i 'digfor'
digfor_cloud_demo.txt
digfor_cado_demo.txt
DescriptionThis is just a test file for the DIGFOR cloud forensics demo.
echo "This is just a test file for the DIGFOR cloud forensics demo." > digfor_cado_demo.txt
cat rm -rf digfor_cado_demo.txt
echo "This is just a test file for the DIGFOR cloud forensics demo." > digfor_cloud_demo.txt
cat digfor_cloud_demo.txt
rm digfor_cado_demo.txt
digfor_cloud_demo.txt
```

OPTIONAL: Analyse the memory dump with Volatility 3. Note that most commands would not work as we do not have a symbol table for Ubuntu 22.04 kernel version. This would need to be created (but would require more time and troubleshooting). Some guidelines:

- https://volatility3.readthedocs.io/en/latest/symbol-tables.html#
- https://medium.com/mii-cybersec/memory-forensic-linux-kernel-confusion-8f711a4ed4d1

```
# Get and test Volatility 3
git clone https://github.com/volatilityfoundation/volatility3.git
cd volatility3
python3 vol.py -h | grep linux
python3 vol.py isfinfo.IsfInfo
python3 vol.py -f ../amoldovan-ubuntu2204_aws_memdump.lime banners.Banners

# We can try to get some Linux symbol tables, but they are old
wget
https://downloads.volatilityfoundation.org/volatility3/symbols/linux.zip
mv linux.zip volatility3/volatility3/symbols

# Other commands would require a symbol table for this kernel version
python3 vol.py -f ../amoldovan-ubuntu2204_aws_memdump.lime linux.bash.Bash
python3 vol.py -f ../amoldovan-ubuntu2204_aws_memdump.lime linux.lsof.Lsof
python3 vol.py -f ../amoldovan-ubuntu2204_aws_memdump.lime linux.proc.Maps
python3 vol.py -f ../amoldovan-ubuntu2204_aws_memdump.lime
linux.pslist.PsList
```



```
┌──(kali㉿kali)-[~/Desktop/volatility3]
└─$ python3 vol.py -f ../amoldovan-ubuntu2204_aws_memdump.lime banners
Volatility 3 Framework 2.4.2
Progress:  100.00               PDB scanning finished
Offset  Banner
clea
0×22400200      Linux version 5.15.0-1031-aws (buildd@lcy02-amd64-016) (gcc (Ubuntu 11.3.0-1ubuntu1~
22.04) 11.3.0, GNU ld (GNU Binutils for Ubuntu) 2.38) #35-Ubuntu SMP Fri Feb 10 02:07:18 UTC 2023 (U
buntu 5.15.0-1031.35-aws 5.15.85
```

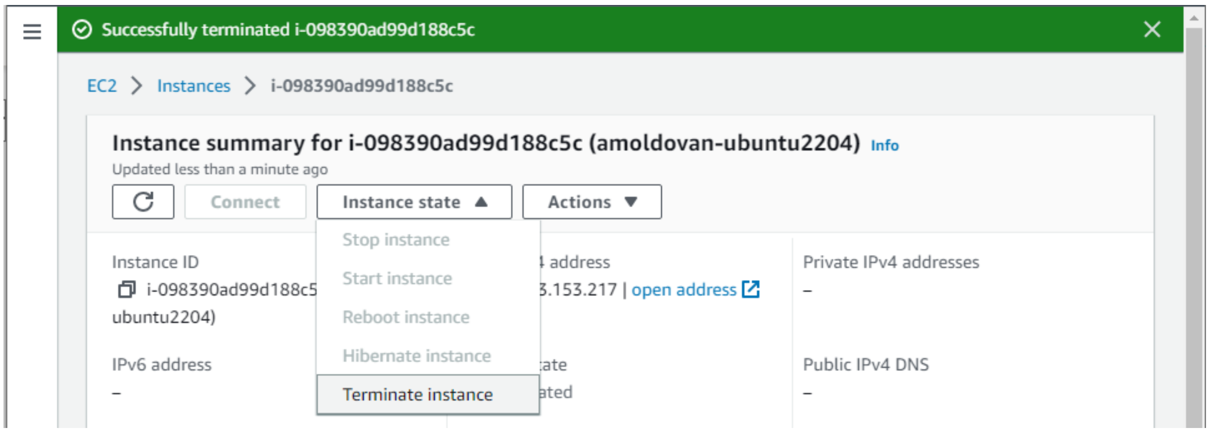
AWS Solution: https://aws.amazon.com/blogs/security/how-to-automatically-build-forensic-kernel-modules-for-amazon-linux-ec2-instances/

Part 5: Terminate the EC2 Instance

IMPORTANT: While we work with t2.micro that is free tier eligible, it is good practice to terminate resources that you do not use. Make sure to terminate the instance when you finished with the practice. You can create another instance later, if needed

In EC2 find your instance and click on it.

Navigate to 'Instance state' → 'Terminate instance'



Successfully terminated i-098390ad99d188c5c

EC2 > Instances > i-098390ad99d188c5c

Instance summary for i-098390ad99d188c5c (amoldovan-ubuntu2204) Info

Updated less than a minute ago

Instance ID: i-098390ad99d188c5 ubuntu2204)

address: 3.153.217 | open address

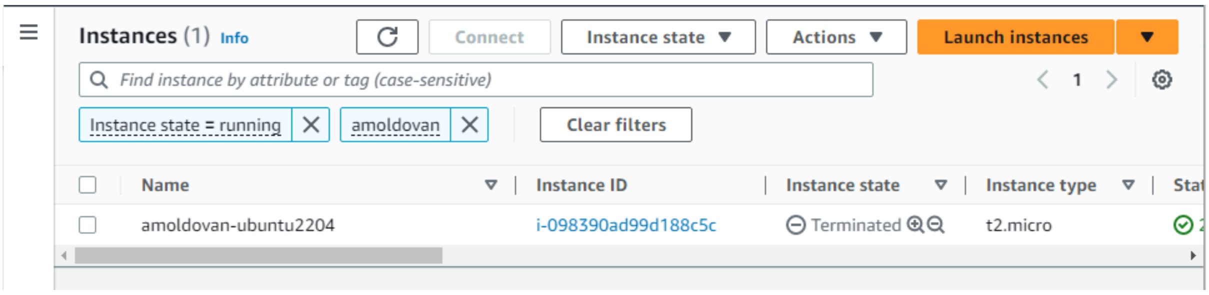
Private IPv4 addresses: –

IPv6 address: –

ate: ated

Public IPv4 DNS: –

Go back to Instances and check it was terminated



Instances (1) Info

Name	Instance ID	Instance state	Instance type	Stat
amoldovan-ubuntu2204	i-098390ad99d188c5c	Terminated	t2.micro	