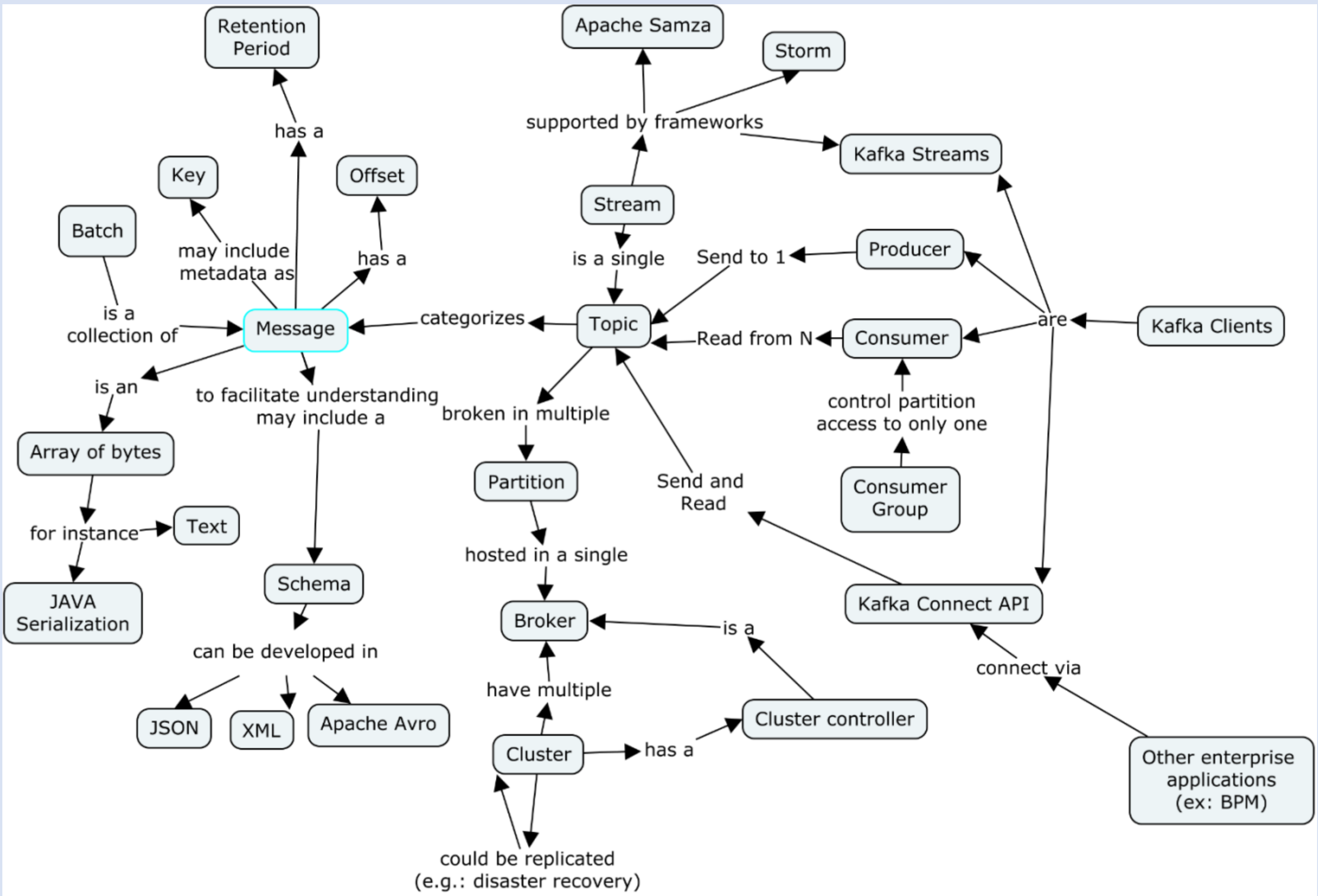
DEI
DEPARTAMENTO
DE ENGENHARIA INFORMÁTICA
TÉCNICO LISBOA

Tutorial – Deploying Kakfa service in Amazon AWS EC2

Instituto Superior Técnico
Universidade de Lisboa
Integração Empresarial / Enterprise Integration

The goal of this document is to show how to operate the basics of a remote Kafka service: installation, starting, accessing, testing, stopping and backup. Remote installation step by step is presented, where the remote installation requires an Amazon AWS account. The following concepts are contained in Kafka:



The following contents is presented in this document.

Contents

A. Creating AWS Academy account and AWS private key

The goal of this section is to activate your AWS account and create your private key that will allow you to access your EC2 instance.

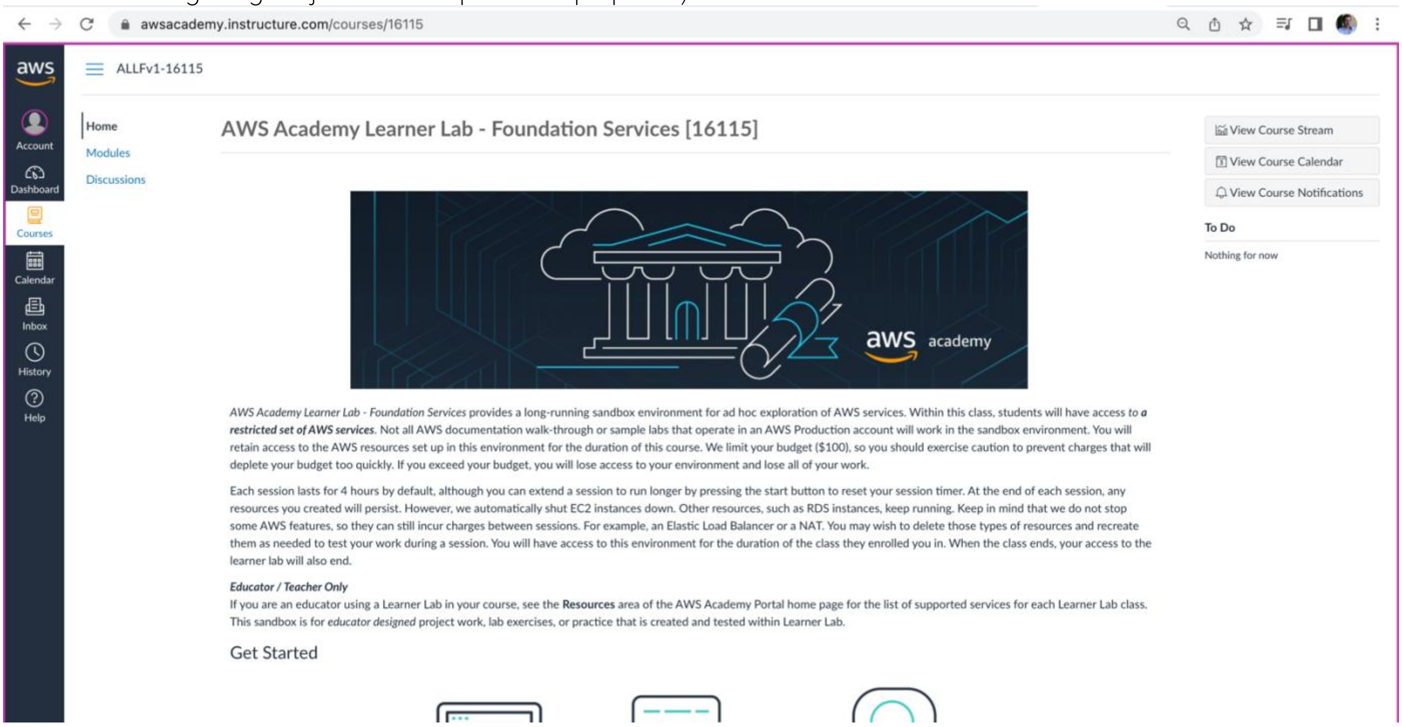
A.0. Create an amazon AWS Academy account using the invitation email sent by Amazon Academy on the faculties request.

While you are in the process of account creation be careful with your graduation date – define it, at least, accordingly with the end of semester date.

Important remark: *the usage of cloud resources implies costs. An EC2 AWS instance is a machine stored in an Amazon AWS facility that is using the usual computational resources: CPU, memory, storage, communications, energy, physical space, among others. This is a reality that any organization faces daily.*

Therefore, you should be aware that while your AWS instance is started, your AWS account also starts to be billed. We absolutely recommend you to always stop your instances when you are not working with them. Always manage your resources wisely!

A.1. After registration, accept the terms, and then you can access your account and have a similar canvas (the following image is just for exemplification purposes).



AWS Academy Learner Lab - Foundation Services [16115]

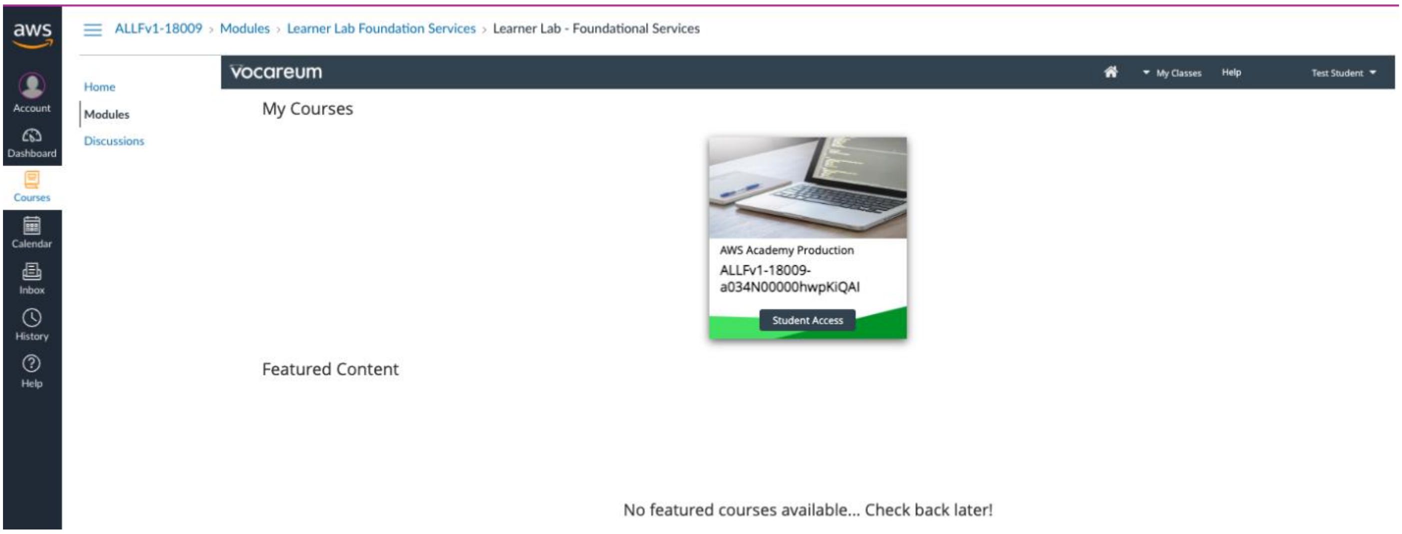
AWS Academy Learner Lab - Foundation Services provides a long-running sandbox environment for ad hoc exploration of AWS services. Within this class, students will have access *to* ***a restricted set of AWS services***. Not all AWS documentation walk-through or sample labs that operate in an AWS Production account will work in the sandbox environment. You will retain access to the AWS resources set up in this environment for the duration of this course. We limit your budget ($100), so you should exercise caution to prevent charges that will deplete your budget too quickly. If you exceed your budget, you will lose access to your environment and lose all of your work.

Each session lasts for 4 hours by default, although you can extend a session to run longer by pressing the start button to reset your session timer. At the end of each session, any resources you created will persist. However, we automatically shut EC2 instances down. Other resources, such as RDS instances, keep running. Keep in mind that we do not stop some AWS features, so they can still incur charges between sessions. For example, an Elastic Load Balancer or a NAT. You may wish to delete those types of resources and recreate them as needed to test your work during a session. You will have access to this environment for the duration of the class they enrolled you in. When the class ends, your access to the learner lab will also end.

Educator / Teacher Only
If you are an educator using a Learner Lab in your course, see the **Resources** area of the AWS Academy Portal home page for the list of supported services for each Learner Lab class. This sandbox is for *educator designed* project work, lab exercises, or practice that is created and tested within Learner Lab.

Get Started

A.2. Select the available course (the following image is just for exemplification purposes):



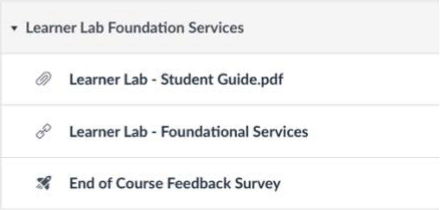
My Courses

AWS Academy Production
ALLFv1-18009-a034N00000hwpKiQAI

Featured Content

No featured courses available... Check back later!

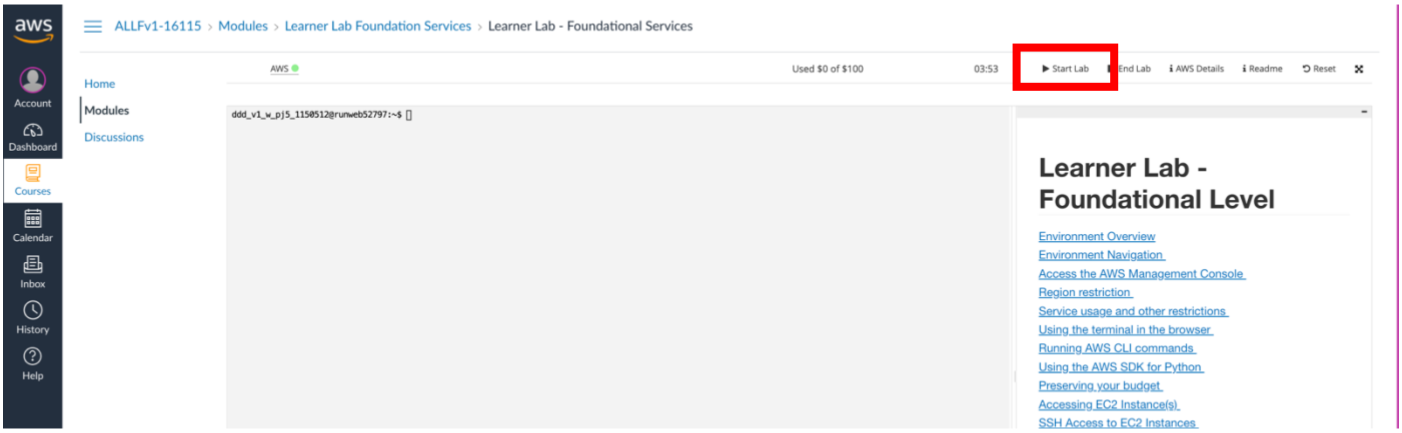
A.3. Select “Modules” and then “Learner Lab – Foundational Services”



Learner Lab Foundation Services

- Learner Lab - Student Guide.pdf
- Learner Lab - Foundational Services
- End of Course Feedback Survey

A.4. Click “**Start Lab**” on the right side

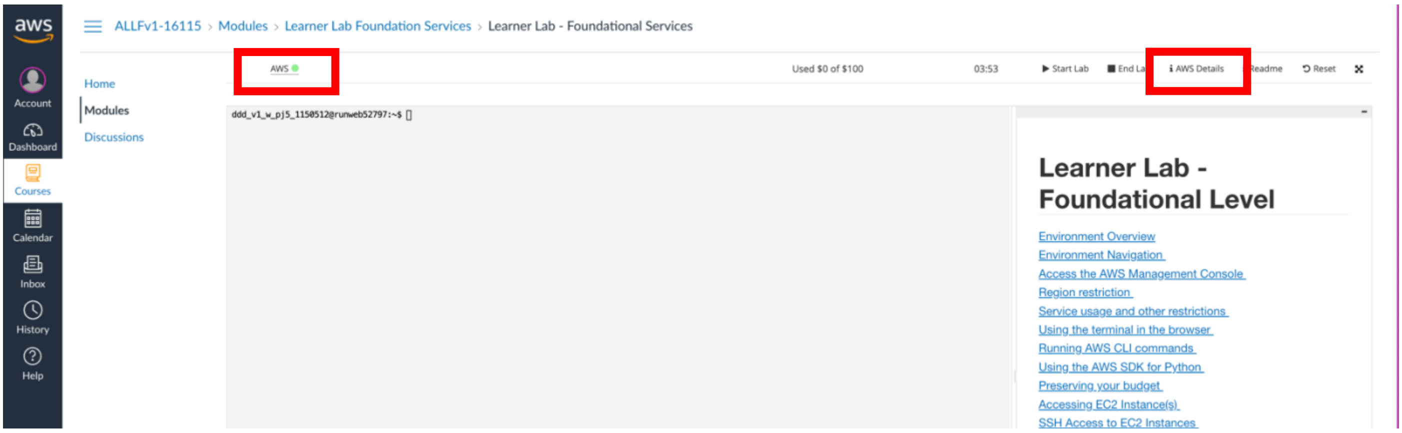


```
ddd_v1_w_pj5_1150512@runweb52797:~$ 
```

Learner Lab - Foundational Level

Environment Overview
Environment Navigation
Access the AWS Management Console
Region restriction
Service usage and other restrictions
Using the terminal in the browser
Running AWS CLI commands
Using the AWS SDK for Python
Preserving your budget
Accessing EC2 Instance(s)
SSH Access to EC2 Instances

A.5. Then, when the indicator is green, click “**AWS Details**” to access AWS Management Console on the right side. As indicated by the red boxes below. Store the returned key file in your computer for later use. You can choose .pem to use with OpenSSH and/or .ppk to use with PuTTY. If you are unsure store the keys in both formats on your computer.

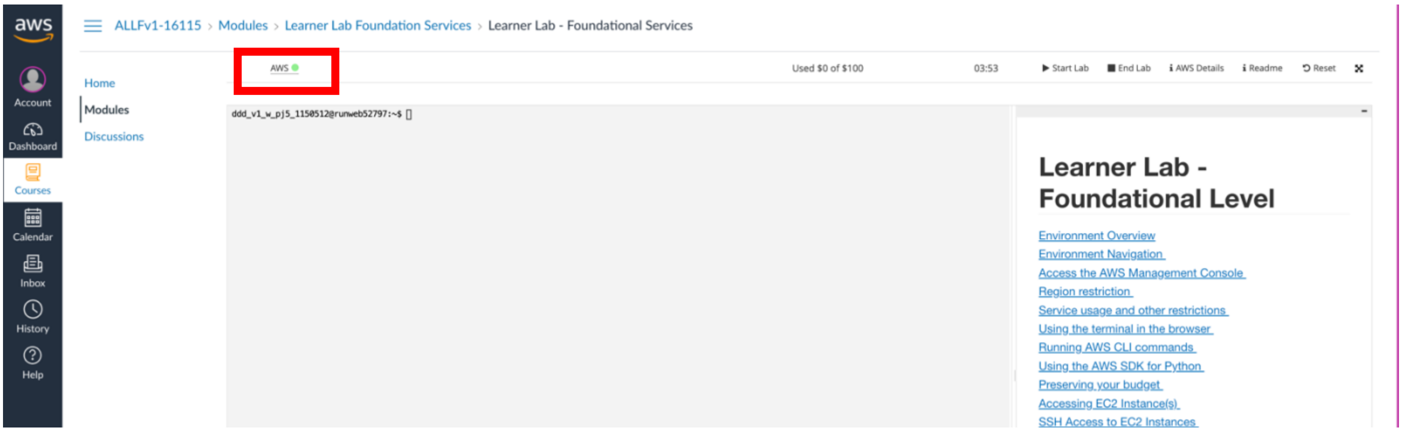


```
ddd_v1_w_pj5_1150512@runweb52797:~$ 
```

Learner Lab - Foundational Level

Environment Overview
Environment Navigation
Access the AWS Management Console
Region restriction
Service usage and other restrictions
Using the terminal in the browser
Running AWS CLI commands
Using the AWS SDK for Python
Preserving your budget
Accessing EC2 Instance(s)
SSH Access to EC2 Instances

A.6. Then, click “**AWS**” to access AWS Management Console on the left side. As indicated by the red box below.

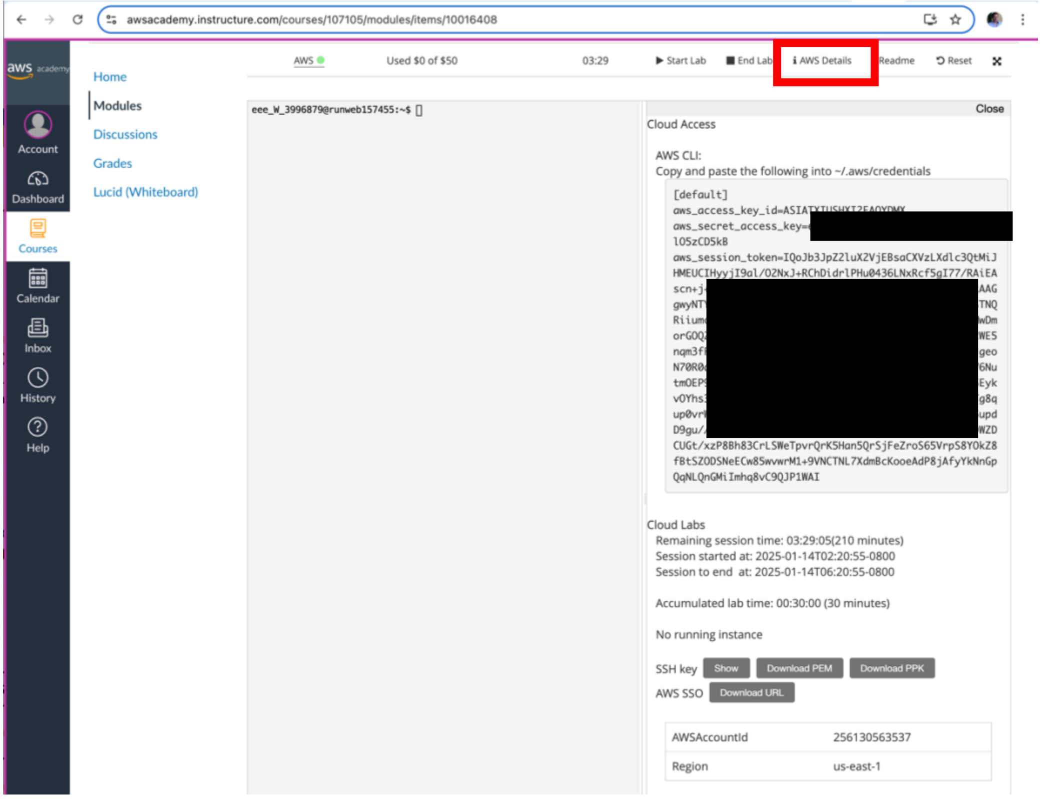


```
ddd_v1_w_pj5_1150512@runweb52797:~$ 
```

Learner Lab - Foundational Level

- Environment Overview
- Environment Navigation
- Access the AWS Management Console
- Region restriction
- Service usage and other restrictions
- Using the terminal in the browser
- Running AWS CLI commands
- Using the AWS SDK for Python
- Preserving your budget
- Accessing EC2 Instance(s)
- SSH Access to EC2 Instances

A.7. Then, click “**AWS Details**” to consult all the credentials to access your AWS area.



```
eee_W_3996879@runweb157455:~$ 
```

Cloud Access

AWS CLI:
Copy and paste the following into ~/.aws/credentials

```
[default]
aws_access_key_id=ASIA[redacted]
aws_secret_access_key=[redacted]
105zCD5kB
aws_session_token=IQoJb3JpZ2luX2VjEBsaCXVzLXdlc3QtMiJ
HMEUCIHyyjI9al/O2NxJ+RChDidrlPHu0436LNxRcf5gI77/RAiEA
scn+j[redacted]AAG
gwyNT[redacted]TNQ
Riium[redacted]wDm
orGOQ[redacted]WE5
nqm3f[redacted]geo
N70R0[redacted]6Nu
tmOEP[redacted]Eyk
vOYhs[redacted]g8q
up0vr[redacted]upd
D9gu/[redacted]WZD
CUGt/xzP8Bh83CrLSWeTpvrQrK5Han5QrSjFeZroS65VrpS8YOkZ8
fBtSZODSNeECw85wvwrM1+9VNCTNL7XdmBcKooeAdP8jAfyYkNnGp
QqNLQnGMiImhq8vC9QJP1WAI
```

Cloud Labs
Remaining session time: 03:29:05(210 minutes)
Session started at: 2025-01-14T02:20:55-0800
Session to end at: 2025-01-14T06:20:55-0800

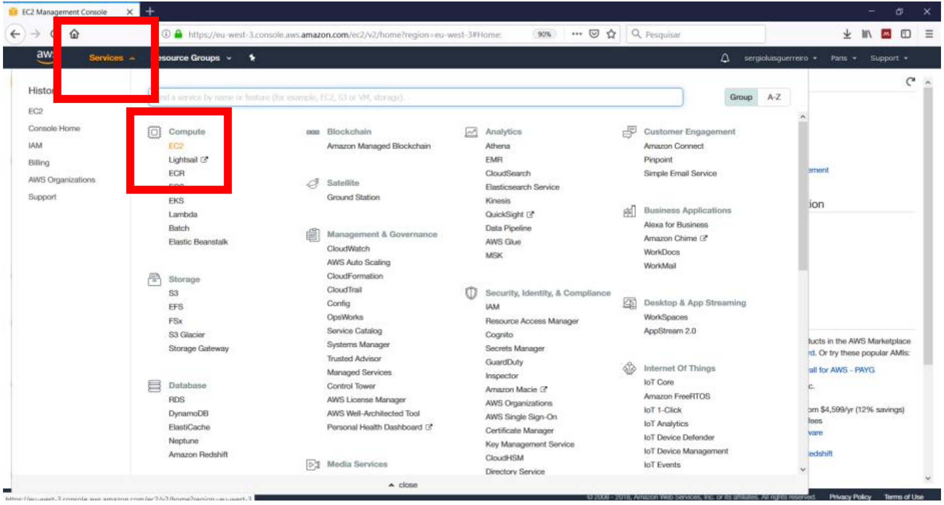
Accumulated lab time: 00:30:00 (30 minutes)

No running instance

SSH key [Show] [Download PEM] [Download PPK]
AWS SSO [Download URL]

AWSAccountId	256130563537
Region	us-east-1

A.8. In the left upper corner, select “`Services -> Compute (EC2)`”. *You can also search for EC2.*

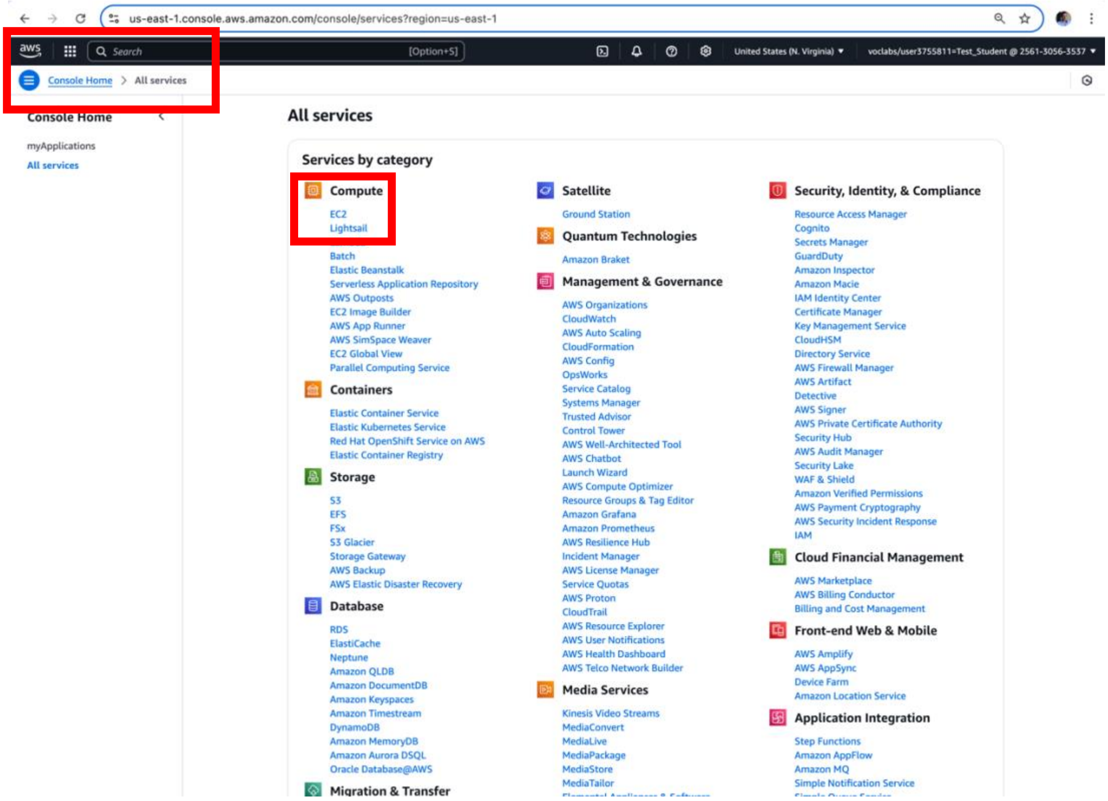


You will obtain the AWS EC2 dashboard. All the options are vertically organized in the left side.

B. Creating and launching an AWS EC2 instance

The goal of this section is to create your first EC2 instance using the available Amazon Machine Images (AMI) available. This requires the previous execution of section A.

B.1. Login in your account, then access the AWS console, and in the left upper corner, select "`All Services -> Compute (EC2)`".



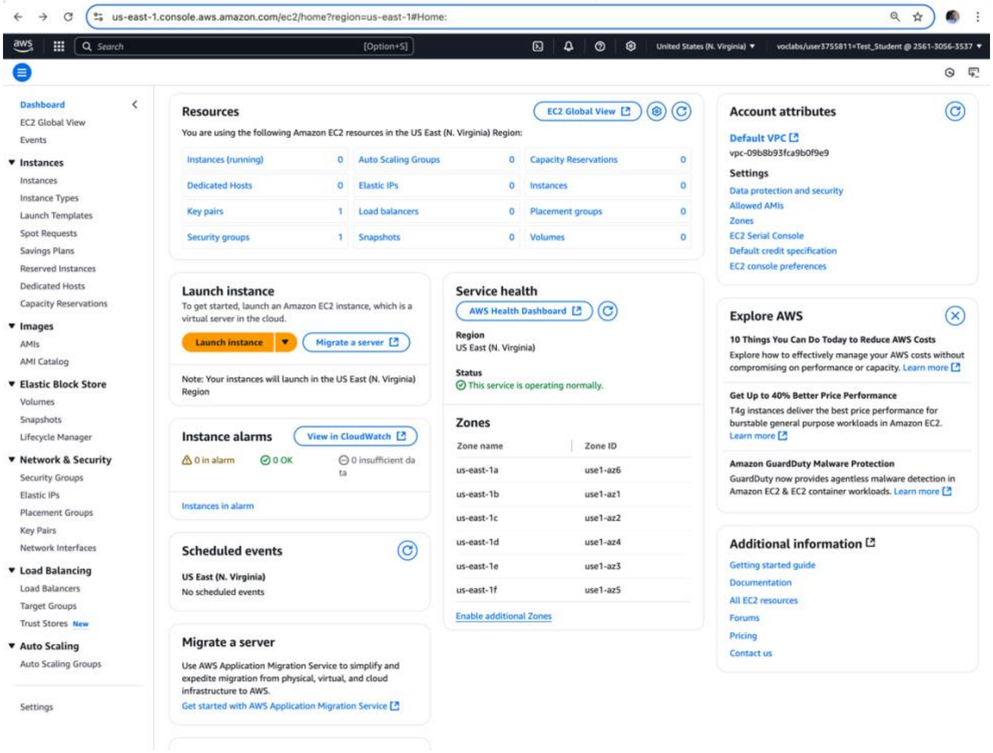
us-east-1.console.aws.amazon.com/console/services?region=us-east-1

United States (N. Virginia) ▾ voclabs/user3755811=Test_Student @ 2561-3056-3537 ▾

All services

Services by category

B.2. Select "`Instances -> Instances`". The following appearance will be shown.



us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Home:

United States (N. Virginia) ▾ voclabs/user3755811=Test_Student @ 2561-3056-3537 ▾

Resources

You are using the following Amazon EC2 resources in the US East (N. Virginia) Region:

Instances (running)	0	Auto Scaling Groups	0	Capacity Reservations	0
Dedicated Hosts	0	Elastic IPs	0	Instances	0
Key pairs	1	Load balancers	0	Placement groups	0
Security groups	1	Snapshots	0	Volumes	0

Launch instance

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

Note: Your instances will launch in the US East (N. Virginia) Region

Instance alarms

⚠ 0 in alarm ✓ 0 OK ⊖ 0 insufficient data

Instances in alarm

Scheduled events

US East (N. Virginia)
No scheduled events

Migrate a server

Use AWS Application Migration Service to simplify and expedite migration from physical, virtual, and cloud infrastructure to AWS.
Get started with AWS Application Migration Service

Service health

Region
US East (N. Virginia)

Status
✓ This service is operating normally.

Zones

Zone name	Zone ID
us-east-1a	use1-az6
us-east-1b	use1-az1
us-east-1c	use1-az2
us-east-1d	use1-az4
us-east-1e	use1-az3
us-east-1f	use1-az5

Enable additional Zones

Account attributes

Default VPC
vpc-09b8b93fca9b0f9e9

Settings
Data protection and security
Allowed AMIs
Zones
EC2 Serial Console
Default credit specification
EC2 console preferences

Explore AWS

10 Things You Can Do Today to Reduce AWS Costs
Explore how to effectively manage your AWS costs without compromising on performance or capacity. Learn more

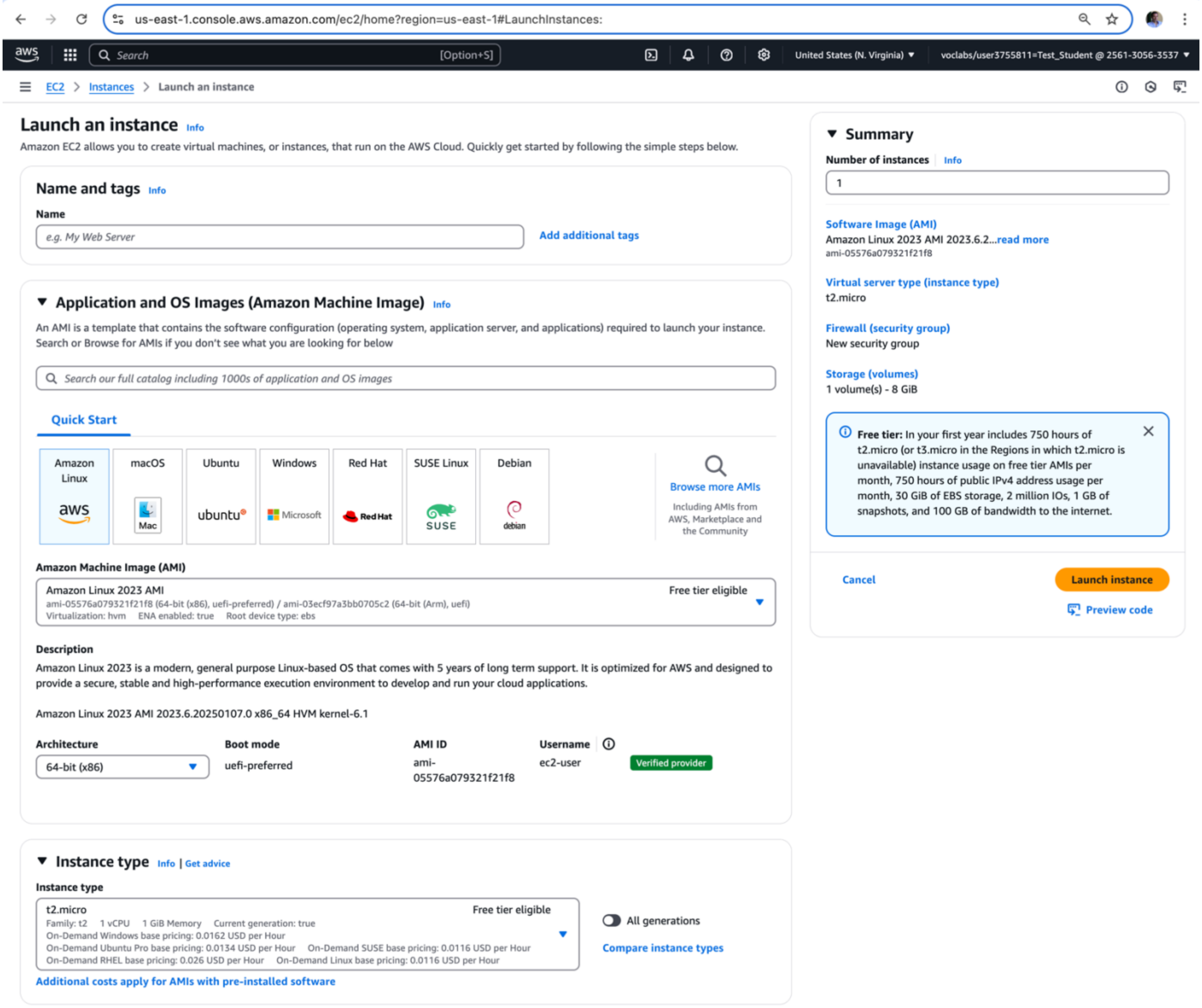
Get Up to 40% Better Price Performance
T4g instances deliver the best price performance for burstable general purpose workloads in Amazon EC2. Learn more

Amazon GuardDuty Malware Protection
GuardDuty now provides agentless malware detection in Amazon EC2 & EC2 container workloads. Learn more

Additional information

Getting started guide
Documentation
All EC2 resources
Forums
Pricing
Contact us

B.3. Select “`Launch Instance`” and a similar interface will be presented as shown below. The goal is to define the Amazon Machine Image (AMI) that will contain all the software configuration required to launch your instance.



us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LaunchInstances:

aws
Search
[Option+S]
United States (N. Virginia) ▼
voclabs/user3755811=Test_Student @ 2561-3056-3537 ▼
EC2 > Instances > Launch an instance

Launch an instance Info

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 and tags Info

Name: e.g. My Web Server — Add additional tags

▼ 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)

Amazon Linux 2023 AMI — Free tier eligible
ami-05576a079321f21f8 (64-bit (x86), uefi-preferred) / ami-03ecf97a3bb0705c2 (64-bit (Arm), uefi)
Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Amazon Linux 2023 is a modern, general purpose Linux-based OS that comes with 5 years of long term support. It is optimized for AWS and designed to provide a secure, stable and high-performance execution environment to develop and run your cloud applications.

Amazon Linux 2023 AMI 2023.6.20250107.0 x86_64 HVM kernel-6.1

Architecture	Boot mode	AMI ID	Username
64-bit (x86)	uefi-preferred	ami-05576a079321f21f8	ec2-user Verified provider

▼ Instance type Info | Get advice

Instance type

t2.micro — Free tier eligible
Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand Windows base pricing: 0.0162 USD per Hour
On-Demand Ubuntu Pro base pricing: 0.0134 USD per Hour On-Demand SUSE base pricing: 0.0116 USD per Hour
On-Demand RHEL base pricing: 0.026 USD per Hour On-Demand Linux base pricing: 0.0116 USD per Hour

All generations — Compare instance types

Additional costs apply for AMIs with pre-installed software

▼ Summary

Number of instances Info: 1

Software Image (AMI): Amazon Linux 2023 AMI 2023.6.2...read more
ami-05576a079321f21f8

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, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

B.4. Use the following configurations:

Step 1 – Choose an Amazon Machine Image (AMI)	select “Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type” (available on January 2025) with “64-bit (x86)”
Step 2 – Choose an Instance Type	select “t2.micro”
Step 3 – Configure Instance Details	check if number of instances = 1 Keep all the remaining options as default
Step 4 – Storage	keep the default size
Step 5 – Key Pair (login)	Key Pair name = vockey

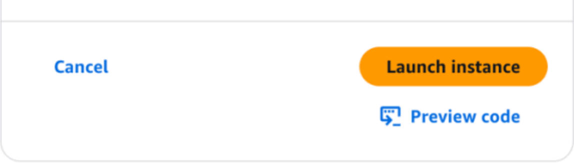
Step 6 – Add Tags	Key = application
	Value = kafka

Step 7 – Configure Security Group *(it could be configured later in the instance console or*	Select “Create a new security group” Security group name: launch-Kafka Select “Add rule” Type = “Custom TCP Rule” Protocol = TCP Port range = 2181

since the creation using terraform)	Source = Anywhere (0.0.0.0, ::/0)
	Description = zookeeper
	Select "Add rule"
	Type = "Custom TCP Rule"
	Protocol = TCP
	Port range = 9092
	Source = Anywhere (0.0.0.0, ::/0)
	Description = kafka
	
	Select "Add rule"
	Type = "SSH"
	Protocol = TCP
	Port range = 22
	Source = Anywhere (0.0.0.0, ::/0)
	Description = remote access

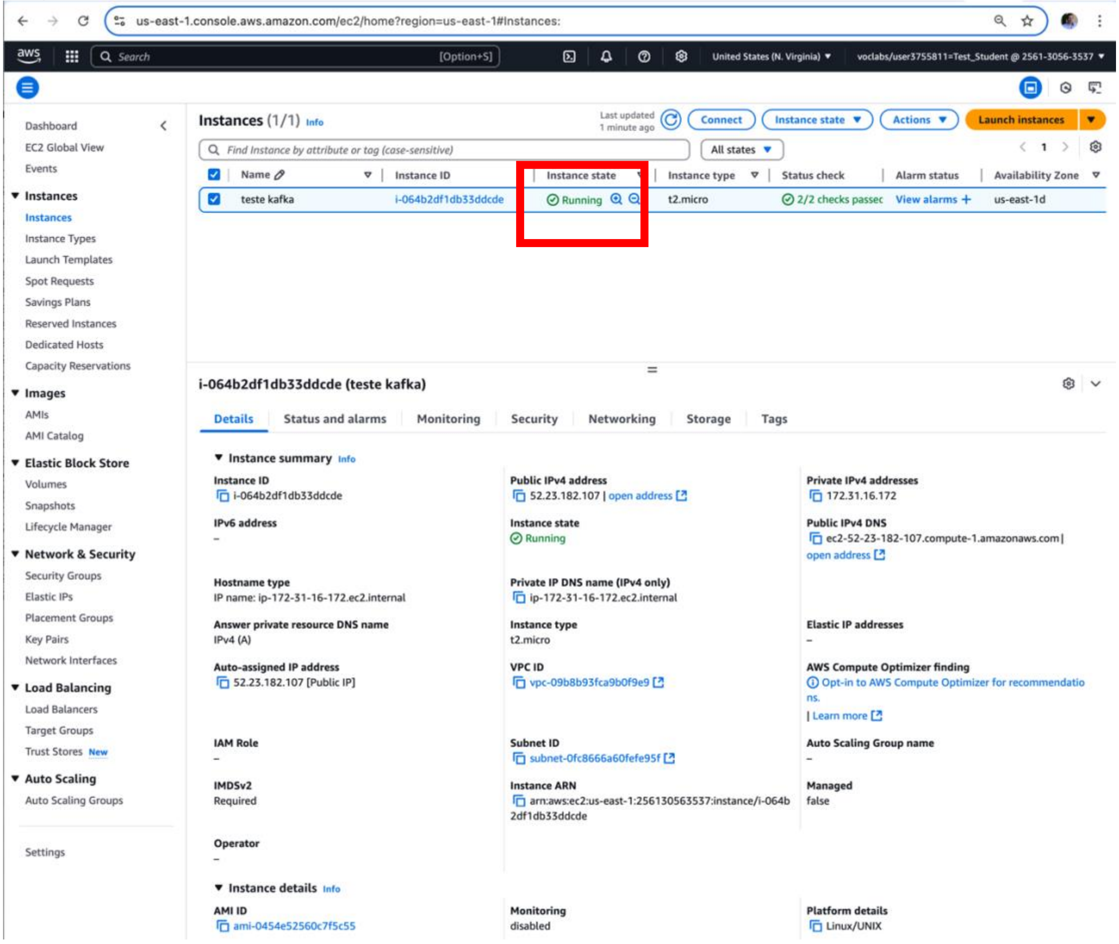
Step 8 – Review Instance Launch	

B.5. Press "Launch" in the bottom of the page



Your AWS account starts to be billed at this point.

B.6. Go to EC2 dashboard as explained in A.1., and you will find an appearance similar with the following image.



Instances (1/1) Info

☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
☑	teste kafka	i-064b2df1db33ddcde	Running	t2.micro	2/2 checks passec	View alarms +	us-east-1d

i-064b2df1db33ddcde (teste kafka)

Instance summary Info

Instance ID
i-064b2df1db33ddcde

IPv6 address
–

Hostname type
IP name: ip-172-31-16-172.ec2.internal

Answer private resource DNS name
IPv4 (A)

Auto-assigned IP address
52.23.182.107 [Public IP]

IAM Role
–

IMDSv2
Required

Operator
–

Public IPv4 address
52.23.182.107 | open address

Instance state
Running

Private IP DNS name (IPv4 only)
ip-172-31-16-172.ec2.internal

Instance type
t2.micro

VPC ID
vpc-09b8b93fca9b0f9e9

Subnet ID
subnet-0fc8666a60fefe95f

Instance ARN
arn:aws:ec2:us-east-1:256130563537:instance/i-064b2df1db33ddcde

Private IPv4 addresses
172.31.16.172

Public IPv4 DNS
ec2-52-23-182-107.compute-1.amazonaws.com | open address

Elastic IP addresses
–

AWS Compute Optimizer finding
Opt-in to AWS Compute Optimizer for recommendations. | Learn more

Auto Scaling Group name
–

Managed
false

Instance details Info

AMI ID
ami-0454e52560c7f5c55

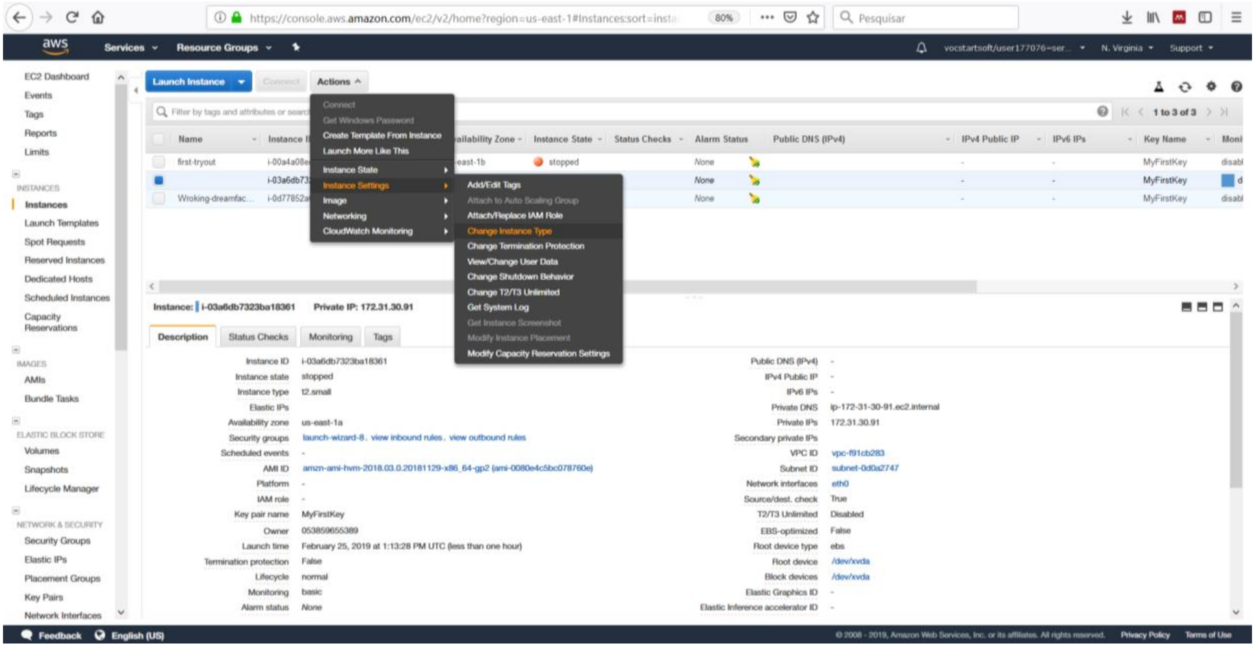
Monitoring
disabled

Platform details
Linux/UNIX

You can check the IPv4 and v6 public address, and the state of the instance. In this example it is *running*.

B.7. Stop the instance, checking the row with the desired instance and then pressing "Instance state -> Stop". Wait for the instance State to change to stopped.

B.8. To allow a proper future execution of Kafka, change the type of instance, by choosing the following options, and then choose **t2.small.**



Instance: i-03a6db7323ba18361 Private IP: 172.31.30.91

Instance ID	i-03a6db7323ba18361
Instance state	stopped
Instance type	t2.small
Elastic IPs	
Availability zone	us-east-1a
Security groups	launch-wizard-8. view inbound rules. view outbound rules
Scheduled events	-
AMI ID	amzn-ami-hvm-2018.03.0.20181129-x86_64-gp2 (ami-0080e4c5bc078760e)
Platform	-
IAM role	-
Key pair name	MyFirstKey
Owner	053859655389
Launch time	February 25, 2019 at 1:13:28 PM UTC (less than one hour)
Termination protection	False
Lifecycle	normal
Monitoring	basic
Alarm status	None

Public DNS (IPv4)	-
IPv4 Public IP	-
IPv6 IPs	-
Private DNS	ip-172-31-30-91.ec2.internal
Private IPs	172.31.30.91
Secondary private IPs	
VPC ID	vpc-f91cb283
Subnet ID	subnet-0d0a2747
Network interfaces	eth0
Source/dest. check	True
T2/T3 Unlimited	Disabled
EBS-optimized	False
Root device type	ebs
Root device	/dev/xvda
Block devices	/dev/xvda
Elastic Graphics ID	-
Elastic Inference accelerator ID	-

C. Access the AWS EC2 instance using PuTTY (for windows only)

The goal of this section is to describe how your AWS EC2 instance can be accessed using an SSH client. It will be useful to install, configure and manage the Kafka server.

C.1. Install PuTTY from https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html

C.2. *(optional – only if the key is a .pem file)* Convert your AWS private key created in section A, using PuTTYgen. For that, execute PuTTYgen from start menu.

C.3. *(optional – only if the key is a .pem file)* Then, in type of key to generate choose: RSA

C.4. *(optional – only if the key is a .pem file)* Press “Load” and locate your AWS private key stored in step A.4. (with .pem file extension)

C.5. *(optional – only if the key is a .pem file)* Press “Save private key” and choose “Yes” when asking to save without passphrase

C.6. *(optional – only if the key is a .pem file)* Store with the same name as the AWS private key (with .ppk file extension)

C.7. Start PuTTY from start menu

C.8. Fill hostname with “user_name@public_dns_name”, where user_name = ec2-user and public_dns_name= <as presented in B.7.>

⚠ Public DNS name is changed whenever an AWS EC2 instance reboot is performed.

C.9. Choose SSH connection type and check if port = 22

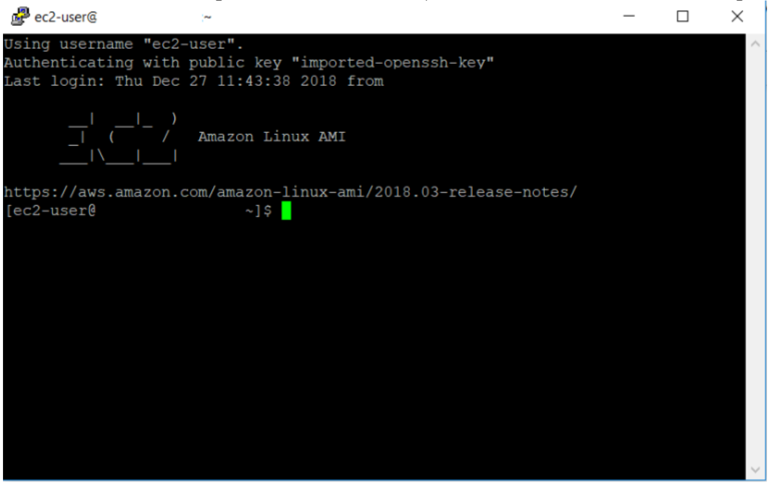
C.10. You can configure PuTTY to automatically send "keepalive" data at regular intervals to keep the session active. This is useful to avoid disconnecting the instance from session inactivity. In the Category panel, choose Connection and enter the required range in the Seconds between keepalives field. For example, if the session disconnects after 10 minutes of inactivity, enter 180 to configure PuTTY to send keepalive data every 3 minutes.

C.11. In the Category panel, choose Connection, choose SSH, choose Auth, select “Browse” and choose the .ppk file from step C.6.

⚠ Start your AWS instance.

C.12. In the Category panel, choose Session, and then select “Open“

C.13. Confirm the certificate in the dialog box. You will be presented with the following similar screen:



```
Using username "ec2-user".
Authenticating with public key "imported-openssh-key"
Last login: Thu Dec 27 11:43:38 2018 from

       __|  __|_  )
       _|  (     /   Amazon Linux AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/
[ec2-user@                ~]$ 
```

C.14. Update your AWS instance with the command:

```
sudo yum update
```

C.15. Check JDK packages available with the command:

```
sudo yum search "java-17"
```

C.16. Execute the command (or another equivalent package)

```
sudo yum install java-17-amazon-corretto-devel.x86_64
```

C.17. Check if JAVA is currently version 17 with the command:

```
java -version
```

If any other version is referred you may remove it, e.g., first check the installed java versions with the command:

```
yum list installed |grep java
```

and then remove the ones that you don't need with the command

```
sudo yum remove java-1.7.0-openjdk
```

C.18. Execute the command:

```
exit
```

⚠ Stop your AWS instance.

D. Access the AWS EC2 instance using ssh (for linux/macOS) and scp (for linux/macOS)

The goal of this section is to describe how your AWS EC2 instance can be accessed using an SSH client in linux. It will be useful to install, configure and manage the Kafka server.

Start your AWS instance.

Public DNS name is changed whenever an AWS EC2 instance reboot is performed.

D.1. In a new terminal session, change directories to the location of the private key file that you created when you created the AWS instance, for instance:

```
[oracle@soabpm-vm ~]$ cd /media/sf_SharedFolderForOracleSOA
```

C.19. Use the following command to set the permissions of your private key file so that only you can read it.

```
[oracle@soabpm-vm sf_SharedFolderForOracleSOA]$ chmod u=rwx,g=,o= myKeyAWS.pem
```

C.20. Use the ssh command to connect to the instance. You specify the private key (.pem) file and user_name@public_dns_name. For example, if you used Amazon Linux 2 or the Amazon Linux AMI, the user name is ec2-user. In

```
[oracle@soabpm-vm sf_SharedFolderForOracleSOA]$ ssh -i myKeyAWS.pem ec2-user@YOURIP
The authenticity of host 'X.X.X.X (X.X.X.X)' can't be established.
RSA key fingerprint is 25:1c:ef:ee:c1:87:61:e9:ea:f6:e3:45:1e:3a:63:73.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'X.X.X.X' (RSA) to the list of known hosts.
Last login: Fri Feb  8 14:59:07 2019 from Y.Y.Y.Y

       __|  __|_  )
       _|  (     /   Amazon Linux AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/
5 package(s) needed for security, out of 6 available
Run "sudo yum update" to apply all updates.
ZooKeeper JMX enabled by default
Using config: /usr/local/zookeeper/bin/../conf/zoo.cfg
Starting zookeeper ... already running as process 11211.
[ec2-user@ip-X-X-X-X ~]$
```

C.21. Update your AWS instance with the command:

```
sudo yum update
```

C.22. Check JDK packages available with the command:

```
sudo yum search "java-17"
```

C.23. Execute the command (or another equivalent package)

```
sudo yum install java-17-amazon-corretto-devel.x86_64
```

C.24. Check if JAVA is currently version 17 with the command:

```
java -version
```

If any other version is referred you may remove it, e.g., first check the installed java versions with the command:

```
yum list installed |grep java
```

and then remove the ones that you don't need with the command

```
sudo yum remove java-1.7.0-openjdk
```

C.25. Execute the command:

```
exit
```

C.26. To transfer files from your machine to your AWS instance, use the scp application in linux. For instance:

```
scp -i my-key-pair.pem /path/SampleFile.txt ec2-user@c2-198-51-100-1.compute-1.amazonaws.com:~
```

C.27. To transfer files from your AWS instance to your machine, use the scp application in linux. For instance:

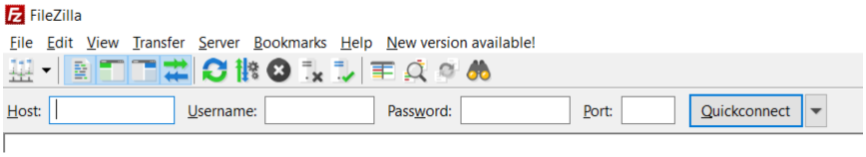
```
scp -i my-key-pair.pem ec2-user@ec2-198-51-100-1.compute-1.amazonaws.com:~/SampleFile.txt ~/SampleFile2.txt
```

⚠ Stop your AWS instance.

E. Access the AWS EC2 instance using FileZilla (for windows and macOS)

The goal of this section is to describe how your AWS EC2 instance can be accessed using an SFTP client. It will be useful to transfer files between your computer and AWS EC2 instance and back.

E.1. Install FileZilla from https://filezilla-project.org/download.php?type=client
E.2. Open FileZilla.
E.3. Go to the FileZilla Edit -> settings, and on the left, click SFTP.
E.4. Add a new private key. (Your .ppk key)
E.5. If you are using a .pem key you must convert it accordingly with steps C.2. to C.6., otherwise it will not work
E.6. At the top in the Quickconnect bar (similar with the image below), put your Public DNS in the host, ec2-user, port 22 (Port 22 is SFTP rather than FTP, AWS will kick back FTP.), and NO PASSWORD.


FileZilla
File Edit View Transfer Server Bookmarks Help New version available!
Host:
Username:
Password:
Port:
Quickconnect

⚠ Start your AWS instance.

E.7. Click Quickconnect.
E.8. File transfer to AWS EC2 instance is only a matter of dragging files from the left to right side.

⚠ Stop your AWS instance.

F. Installing manually Kafka in the AWS EC2 instance

The goal of this section is to describe the required steps to install the Zookeeper and Kafka servers in your AWS EC2 instance.

Start your AWS instance.

F.1. Access your AWS EC2 instance using previous steps of this tutorial.

F.2. To change for your home directory, type the command:

```
cd
```

F.3. Download the Kafka binary version from https://kafka.apache.org/downloads. Recommended version is kafka_2.13-3.9.0.tgz (in January 2025).

```
wget https://downloads.apache.org/kafka/3.9.0/kafka_2.13-3.9.0.tgz
```

F.4. Download the ZooKeeper binary version from https://zookeeper.apache.org/releases.html. Recommended version is apache-zookeeper-3.9.3-bin.tar.gz (in January 2025).

```
wget https://dlcdn.apache.org/zookeeper/zookeeper-3.9.3/apache-zookeeper-3.9.3-bin.tar.gz
```

F.5. Then, type the command:

```
ls -ltr
```

to check if you have both kafka and zookeeper files available.

F.6. Update your AWS instance with the command:

```
sudo yum update
```

F.7. Execute the command (or another equivalent package)

```
sudo yum install java-17-amazon-corretto-devel.x86_64
```

F.8. To extract the zookeeper files, type the command:

```
tar -zxf apache-zookeeper-3.9.3-bin.tar.gz
```

F.9. Move the new zookeeper directory to system typing the command:

```
sudo mv apache-zookeeper-3.9.3-bin /usr/local/zookeeper
```

F.10. Create a new directory using the command:

```
sudo mkdir -p /var/lib/zookeeper
```

F.11. Create the baseline zookeeper server configuration with the command:

```
cat > /usr/local/zookeeper/conf/zoo.cfg << EOF
```

and then write the following content to the file directly in the command line:

```
tickTime=2000
dataDir=/var/lib/zookeeper
clientPort=2181
EOF
```

F.12. Start ZooKeeper server with the command:

```
sudo /usr/local/zookeeper/bin/zkServer.sh start
```

F.13. To extract the kafka server, type the command:

```
tar -zxf kafka_2.13-3.9.0.tgz
```

F.14. Move the new kafka directory to system typing the command:

```
sudo mv kafka_2.13-3.9.0 /usr/local/kafka
```

F.15. Create a new directory using the command:

```
sudo mkdir /tmp/kafka-logs
```

F.16. Start Kafka with the command:

```
sudo /usr/local/kafka/bin/kafka-server-start.sh -daemon /usr/local/kafka/config/server.properties
```

Hint 1: if you are short in the instance memory reduce the JVM usage with the command:

```
export KAFKA_HEAP_OPTS="-Xmx256M -Xms256M"
```

Hint 2: you can always check if Zookeeper and Kafka servers are started correctly with the command:

```
ps -ef |grep java
```

⚠ Stop your AWS instance.

Exercise 1: To automate the start of Zookeeper and Kafka servers with instance login write both starting commands in the file:

```
.bash_profile
```

G. Testing Kafka locally in the AWS EC2 instance

The goal of this section if to test if the previous installation in section E. was executed successfully. The production and consumption of Kafka messages are tested. It is considered that in this environment the servers and the clients are installed in the same machine.

Start your AWS instance.

G.1.Access your AWS EC2 instance using the steps C.7. to C.12.
G.2.First, install telnet package in your AWS EC2 instance typing the command:

```
sudo yum install telnet.x86_64
```

G.3.**Test ZooKeeper** installation typing the command:

```
telnet localhost 2181
```

then write the command:

```
srvr
```

and it is expected to receive the following output:

```
[ec2-user@ ~]$ telnet localhost 2181
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
srvr
Zookeeper version: 3.4.12-e5259e437540f349646870ea94dc2658c4e44b3b, built on 03/27/2018 03:55 GMT
Latency min/avg/max: 0/0/0
Received: 1
Sent: 0
Connections: 1
Outstanding: 0
Zxid: 0x0
Mode: standalone
Node count: 4
Connection closed by foreign host.
```

G.4.**Test Kafka** installation, with the creation of a new topic named "test", typing the command:

```
sudo /usr/local/kafka/bin/kafka-topics.sh --create --bootstrap-server localhost:9092 --replication-factor 1 --partitions 1 --topic test
```

G.5.Type the command to **check if the topic "test" is created successfully**:

```
sudo /usr/local/kafka/bin/kafka-topics.sh --bootstrap-server localhost:9092 --describe --topic test
```

G.6.Type the command to **produce messages to the topic "test"**:

```
sudo /usr/local/kafka/bin/kafka-console-producer.sh --broker-list localhost:9092 --topic test
```

And write some text messages, one per line. Then, to finish press Ctrl+D.

G.7.Type the command to **consume messages from the topic "test"**:

```
sudo /usr/local/kafka/bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic test --from-beginning
```

All the sent messages from topic "test" will be presented. Then, to finish press Ctrl+C.

Exercise 2: Try creating different PuTTY sessions for producer and consumer and check the run-time production and consumption of messages.

Stop your AWS instance.

H. Testing Kafka in the AWS EC2 instance remotely

The goal of this section if to test if the previous installation in section E. was executed successfully. The production and consumption of Kafka messages are tested. It is considered that in this environment the servers and the clients are installed in different machines.

This section requires the previous installation of Kafka, as described in section E.

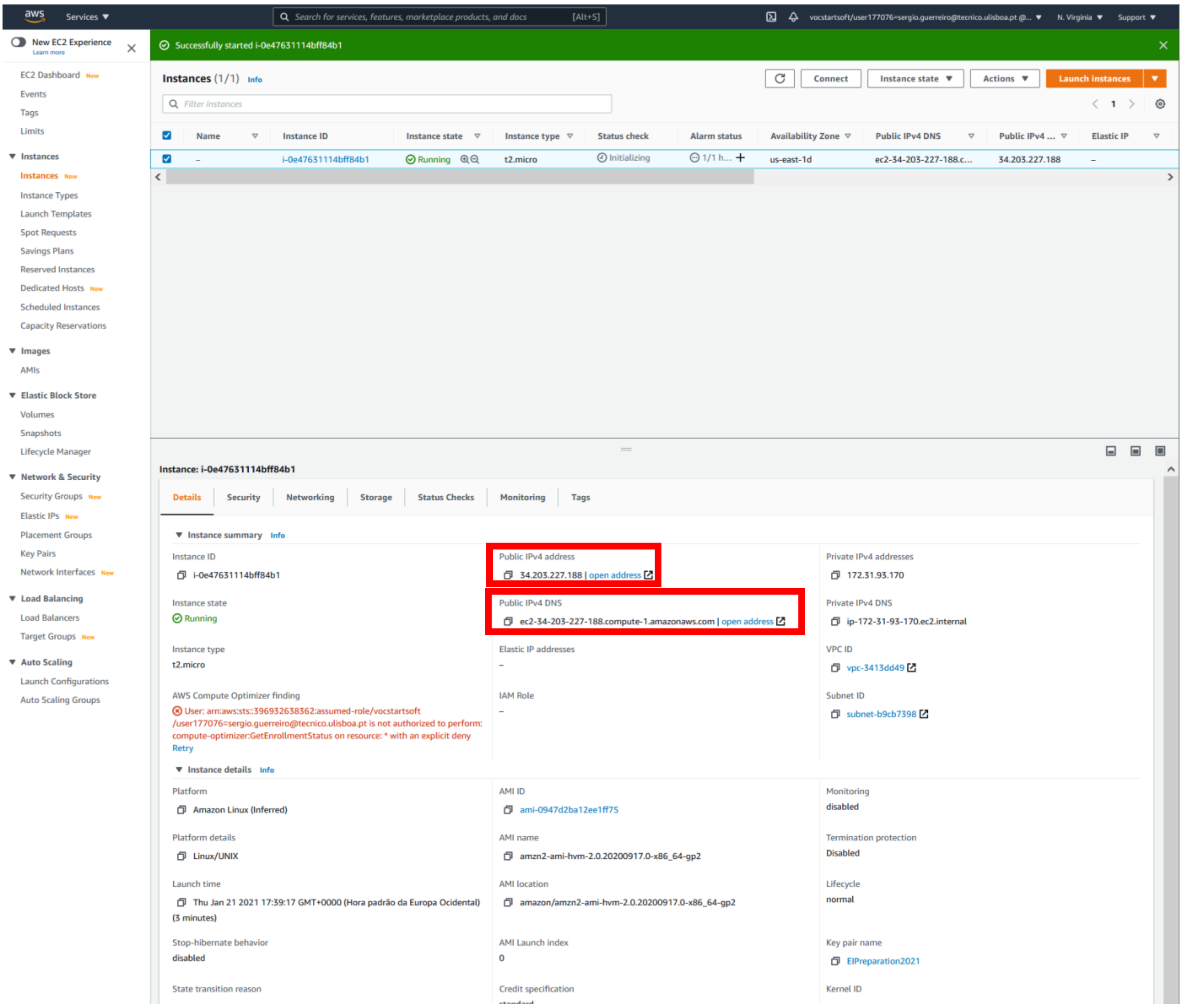
⚠ Start your AWS instance.

H.0. Access your AWS EC2 instance using the steps C.7. to C.12.

H.1. From your local docker container, from local macOS, local linux, or other AWS EC2 linux environment, check if you can connect to the AWS EC2 instance Zookeeper server using the following command:

```
telnet YourPublicIP 2181
```

remember that *YourPublicIP* can be found in the AWS EC2 dashboard as presented below:



Successfully started i-0e47631114bff84b1

Instances (1/1) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
–	i-0e47631114bff84b1	Running	t2.micro	Initializing	1/1 h...	us-east-1d	ec2-34-203-227-188.c...	34.203.227.188	–

Instance: i-0e47631114bff84b1

▼ Instance summary Info

Instance ID
i-0e47631114bff84b1

Public IPv4 address
34.203.227.188 | open address

Private IPv4 addresses
172.31.93.170

Instance state
Running

Public IPv4 DNS
ec2-34-203-227-188.compute-1.amazonaws.com | open address

Private IPv4 DNS
ip-172-31-93-170.ec2.internal

Instance type
t2.micro

Elastic IP addresses
–

VPC ID
vpc-3413dd49

AWS Compute Optimizer finding
User: arn:aws:sts::396932638362:assumed-role/vocstartsoft/user177076=sergio.guerreiro@tecnico.ulisboa.pt is not authorized to perform: compute-optimizer:GetEnrollmentStatus on resource: * with an explicit deny
Retry

IAM Role
–

Subnet ID
subnet-b9cb7398

▼ Instance details Info

Platform
Amazon Linux (Inferred)

AMI ID
ami-0947d2ba12ee1ff75

Monitoring
disabled

Platform details
Linux/UNIX

AMI name
amzn2-ami-hvm-2.0.20200917.0-x86_64-gp2

Termination protection
Disabled

Launch time
Thu Jan 21 2021 17:39:17 GMT+0000 (Hora padrão da Europa Ocidental) (3 minutes)

AMI location
amazon/amzn2-ami-hvm-2.0.20200917.0-x86_64-gp2

Lifecycle
normal

Stop-hibernate behavior
disabled

AMI Launch index
0

Key pair name
EIPreparation2021

State transition reason

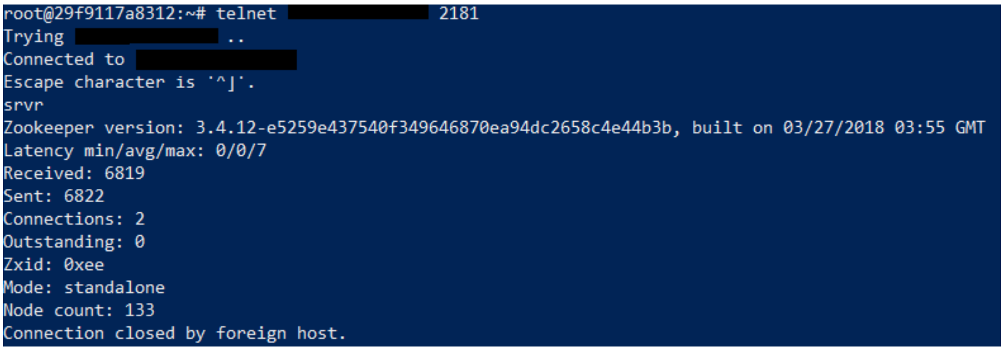
Credit specification

Kernel ID

After that, send the command:

```
srvr
```

You should be able to see an interaction similar with the following:



```
root@29f9117a8312:~# telnet [redacted] 2181
Trying [redacted] ..
Connected to [redacted]
Escape character is '^]'.
srvr
Zookeeper version: 3.4.12-e5259e437540f349646870ea94dc2658c4e44b3b, built on 03/27/2018 03:55 GMT
Latency min/avg/max: 0/0/7
Received: 6819
Sent: 6822
Connections: 2
Outstanding: 0
Zxid: 0xee
Mode: standalone
Node count: 133
Connection closed by foreign host.
```

Hint 3: Public IP or Public DNS Name could be.

H.2. Using the FileZilla, download the file /usr/local/kafka/config/server.properties from your AWS EC2 instance.

H.3. Edit the previous file adding the following uncommented text. Where *Your Public DNS Name* can be found in the AWS EC2 dashboard as presented above.

```
…
############################# Socket Server Settings #############################

# The address the socket server listens on. It will get the value returned from
# java.net.InetAddress.getCanonicalHostName() if not configured.
#   FORMAT:
#     listeners = listener_name://host_name:port
#   EXAMPLE:
#     listeners = PLAINTEXT://your.host.name:9092
#listeners=PLAINTEXT://:9092
listeners=PLAINTEXT://Your Public DNS Name:9092
…
```

Hint 4: Public DNS name is changed whenever an AWS EC2 instance reboot is performed, therefore reconfiguration of this file could be needed often.

H.4. Save the file and upload, using FileZilla, for your AWS EC2 instance in /usr/local/kafka/config/server.properties.

Hint 5 regarding H.2., H.3. and H.4.: another option to change the content of the server.properties file is using the vi application directly in the AWS EC2 environment.

H.5. Stop Kafka server using the command:

```
sudo /usr/local/kafka/bin/kafka-server-stop.sh -daemon /usr/local/kafka/config/server.properties
```

H.6. Check that Kafka server is stopped with:

```
ps -ef |grep java|grep kafka
```

H.7. Restart Kafka with the command, as in F.15.:

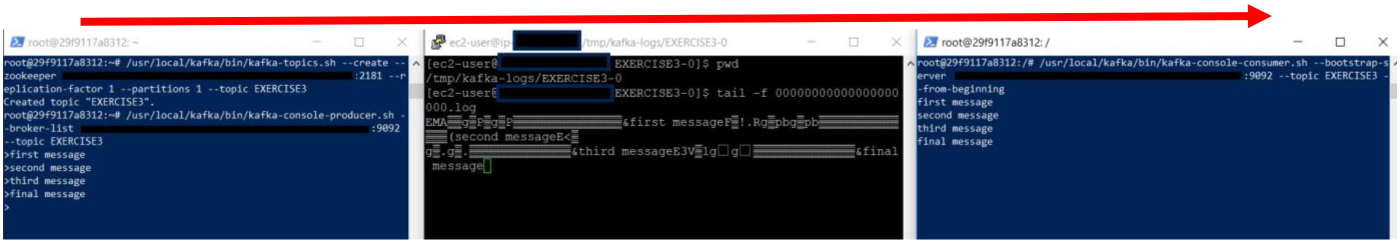
```
sudo /usr/local/kafka/bin/kafka-server-start.sh -daemon /usr/local/kafka/config/server.properties
```

H.8. To test the connection, from your local docker container, from local macOS, local linux, or other AWS EC2 linux environment, type the command to **check if the topic “test” is created successfully in the AWS EC2 instance**:

```
/usr/local/kafka/bin/kafka-topics.sh --bootstrap-server YourPublicDNSName:9092 --describe --topic test
```

H.9. You are now able to communicate with the AWS EC2 instance. Any other command from section G. can now be tested.

Exercise 3: Create an environment where the messages are produced from one AWS EC2 machine, sent to Kafka server in AWS EC2 instance, and then consumed by another AWS EC2 machine. Check if the flow of messages is executing correctly. Screenshot of the solution:



```
root@29f9117a8312:~# /usr/local/kafka/bin/kafka-topics.sh --create --
zookeeper [redacted] :2181 --r
eplication-factor 1 --partitions 1 --topic EXERCISE3
Created topic "EXERCISE3".
root@29f9117a8312:~# /usr/local/kafka/bin/kafka-console-producer.sh -
-broker-list [redacted] :9092
--topic EXERCISE3
>first message
>second message
>third message
>final message
>
```

```
[ec2-user@[redacted] EXERCISE3-0]$ pwd
/tmp/kafka-logs/EXERCISE3-0
[ec2-user@[redacted] EXERCISE3-0]$ tail -f 0000000000000000
000.log
EMA▒g▒P▒g▒P▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒&first messageF▒!.Rg▒pbg▒pb▒▒▒▒▒▒▒▒▒▒
▒▒▒(second messageE<▒
g▒.g▒.▒▒▒▒▒▒▒▒▒▒▒▒▒▒&third messageE3V▒lg□g□▒▒▒▒▒▒▒▒▒▒▒▒▒▒&final
 message
```

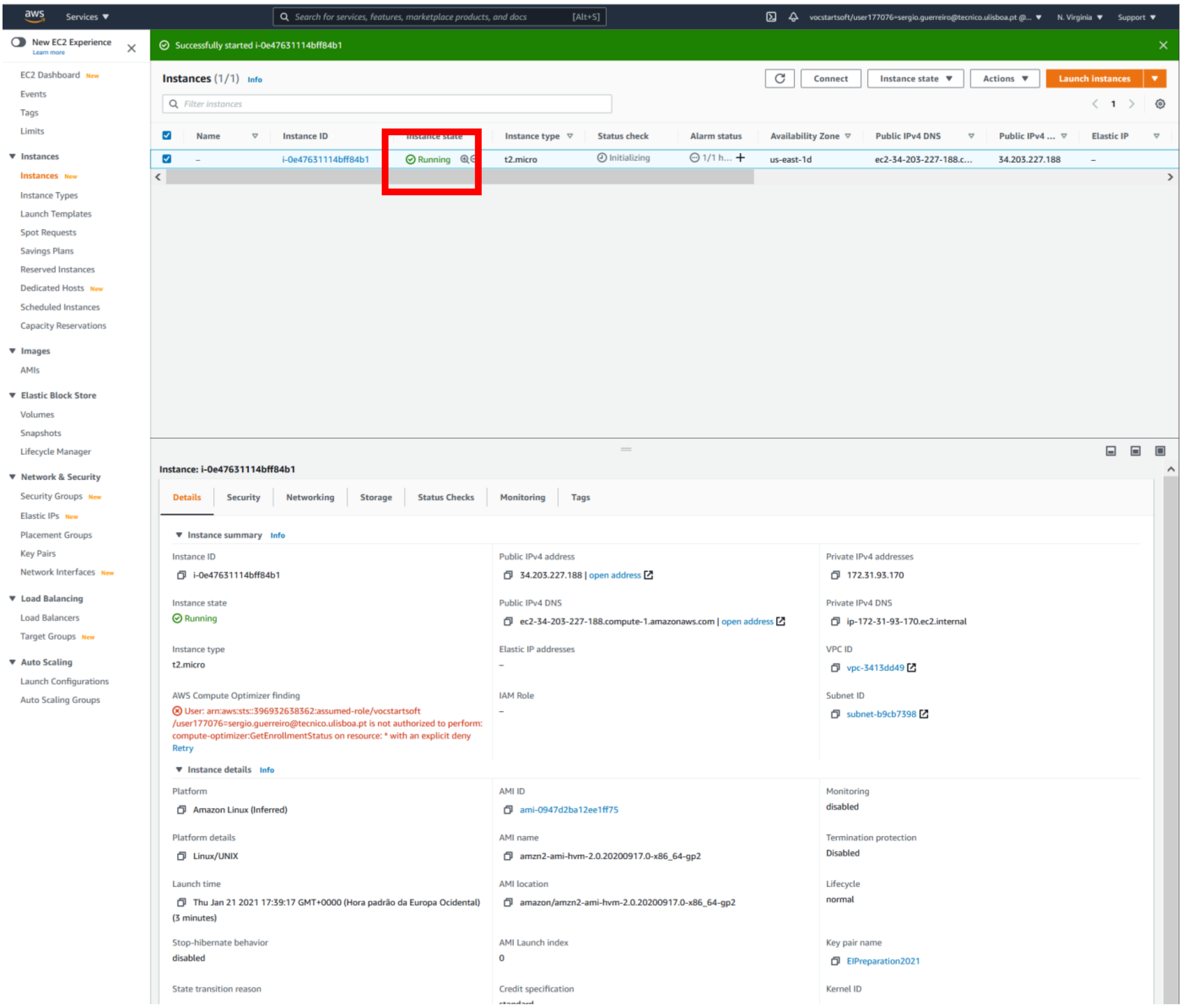
```
root@29f9117a8312:/# /usr/local/kafka/bin/kafka-console-consumer.sh --bootstrap-s
erver [redacted] :9092 --topic EXERCISE3 -
-from-beginning
first message
second message
third message
final message
```

⚠ Stop your AWS instance.

I. Stopping the AWS EC2 instance

The goal of this section is to describe how an AWS EC2 instance can be stopped. This section requires the execution of section B.

I.1. Go to AWS EC2 dashboard as explained in A.1., and you will find an appearance similar with the following image.



Successfully started i-0e47631114bff84b1

Instances (1/1) Info

☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
☑	–	i-0e47631114bff84b1	Running	t2.micro	Initializing	1/1 h...	us-east-1d	ec2-34-203-227-188.c...	34.203.227.188	–

Instance: i-0e47631114bff84b1

▼ Instance summary Info

Instance ID i-0e47631114bff84b1	Public IPv4 address 34.203.227.188 | open address	Private IPv4 addresses 172.31.93.170
Instance state Running	Public IPv4 DNS ec2-34-203-227-188.compute-1.amazonaws.com | open address	Private IPv4 DNS ip-172-31-93-170.ec2.internal
Instance type t2.micro	Elastic IP addresses –	VPC ID vpc-3413dd49
AWS Compute Optimizer finding User: arn:aws:sts::396932638362:assumed-role/vocstartsoft/user177076=sergio.guerreiro@tecnico.ulisboa.pt is not authorized to perform: compute-optimizer:GetEnrollmentStatus on resource: * with an explicit deny Retry	IAM Role –	Subnet ID subnet-b9cb7398

▼ Instance details Info

Platform Amazon Linux (Inferred)	AMI ID ami-0947d2ba12ee1ff75	Monitoring disabled
Platform details Linux/UNIX	AMI name amzn2-ami-hvm-2.0.20200917.0-x86_64-gp2	Termination protection Disabled
Launch time Thu Jan 21 2021 17:39:17 GMT+0000 (Hora padrão da Europa Ocidental) (3 minutes)	AMI location amazon/amzn2-ami-hvm-2.0.20200917.0-x86_64-gp2	Lifecycle normal
Stop-hibernate behavior disabled	AMI Launch index 0	Key pair name EIPreparation2021
State transition reason	Credit specification standard	Kernel ID

I.2. Stop the instance, checking the row with the desired instance and then pressing “Instance state -> Stop instance state”. Wait for the instance State to change to stopped.

Other references

- Connecting to Your Linux Instance Using SSH, https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AccessingInstancesLinux.html
- To create a Maven project in eclipse: https://www.tech-recipes.com/rx/39279/create-a-new-maven-project-in-eclipse/
- About Kafka group-ids: https://stackoverflow.com/questions/35561110/can-multiple-kafka-consumers-read-same-message-from-the-partition
- Where to download apache maven https://maven.apache.org/download.cgi
- How to install apache maven https://maven.apache.org/install.html