Migrating Python Web Application on Amazon Virtual Private Cloud- Handbook

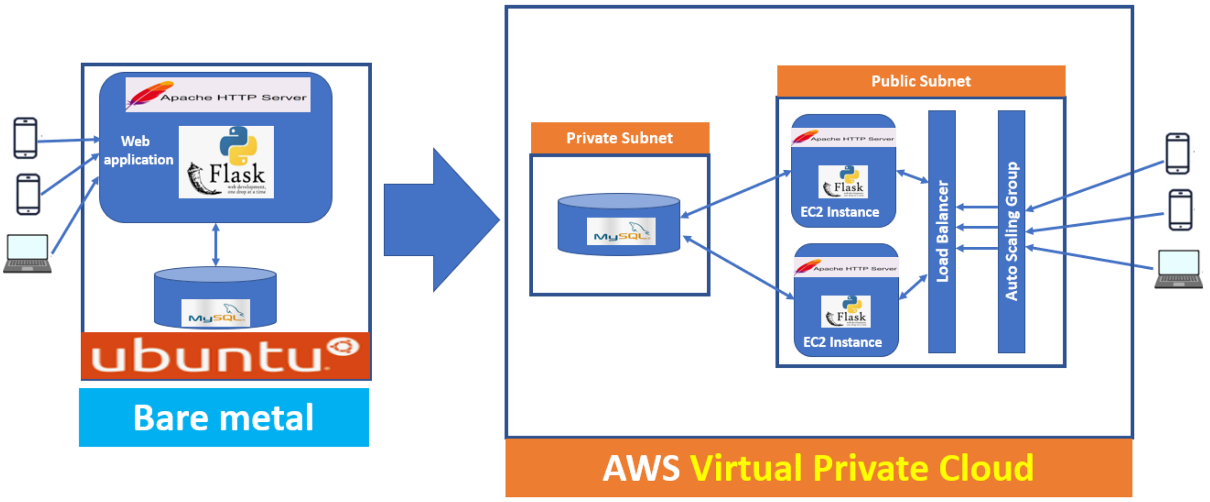
Author:
Pramod Singh
datasciencechampion@gmail.com

Table of contents

1. Objective

Objective is to migrate Python (Flask) web application from bare metal to Amazon VPC (Virtual Private Cloud)

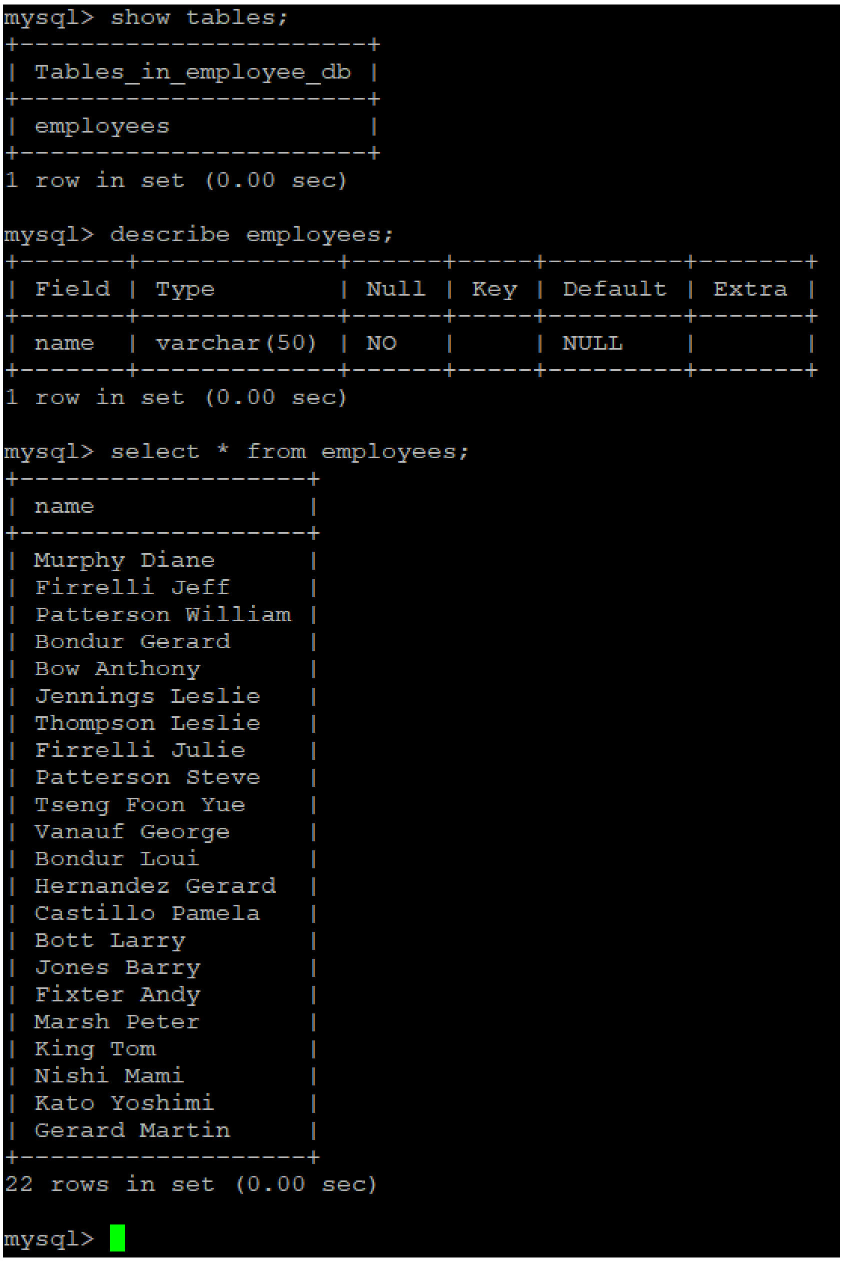
Application Migration : Bare metal to AWS VPC



1.1.1 About Existing Application

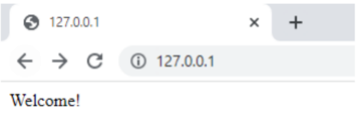
- Python application is running on the VM (Ubuntu server)
- Application is developed in Python using Flask framework.
- Using Apache webserver 2

- MYSQL is the backend database.

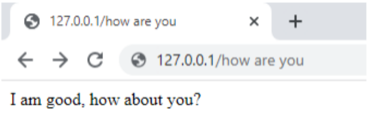


```
mysql> show tables;
+-----------------------+
| Tables_in_employee_db |
+-----------------------+
| employees             |
+-----------------------+
1 row in set (0.00 sec)

mysql> describe employees;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| name  | varchar(50) | NO   |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
1 row in set (0.00 sec)

mysql> select * from employees;
+--------------------+
| name               |
+--------------------+
| Murphy Diane       |
| Firrelli Jeff      |
| Patterson William  |
| Bondur Gerard      |
| Bow Anthony        |
| Jennings Leslie    |
| Thompson Leslie    |
| Firrelli Julie     |
| Patterson Steve    |
| Tseng Foon Yue     |
| Vanauf George      |
| Bondur Loui        |
| Hernandez Gerard   |
| Castillo Pamela    |
| Bott Larry         |
| Jones Barry        |
| Fixter Andy        |
| Marsh Peter        |
| King Tom           |
| Nishi Mami         |
| Kato Yoshimi       |
| Gerard Martin      |
+--------------------+
22 rows in set (0.00 sec)

mysql>
```

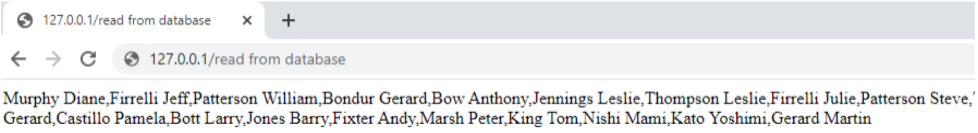
- End user access this application using web browser where he/she can send following web request and get the response from application/database.



Welcome!



I am good, how about you?



Murphy Diane,Firrelli Jeff,Patterson William,Bondur Gerard,Bow Anthony,Jennings Leslie,Thompson Leslie,Firrelli Julie,Patterson Steve,'
Gerard,Castillo Pamela,Bott Larry,Jones Barry,Fixter Andy,Marsh Peter,King Tom,Nishi Mami,Kato Yoshimi,Gerard Martin

2. Detailed Requirements

The application is developed in Python using Flask framework with MYSQL as the backend database. The application is running on-premises on an Ubuntu server which is made available as a virtual machine. Following are the tasks and requirements to migrate it to Amazon Cloud.

- Migrate the on-premises application
- Migrate the existing data to AWS cloud platform ensuring secure access to the data.
- The database should not be accessible from outside the VPC.
- Data should be accessible only from the EC2 instances where the Python application is installed.
- The application should be accessible from web-browser using standard HTTP port (80) so that end user does not experience any bad experience and uses the application in the similar way it was working on-premises.
- Enable autoscaling with load balancer so that it scales as per traffic load.

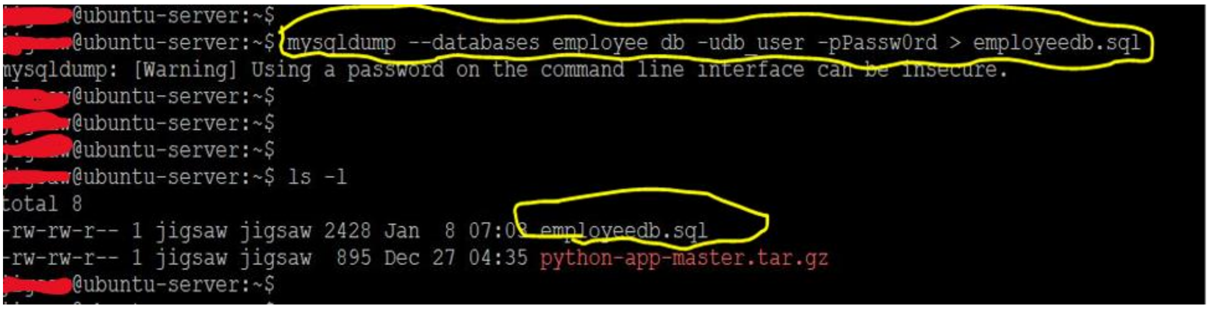
3. Migrating Steps

This section describes step by step implementation of this cloud migration project.

3.1 Exporting existing data using mysqldump

Here we are exporting existing data from mysql database which is running on-premises on an Ubuntu server.

3.1.1 Take dbdump using mysqldump



```
[redacted]@ubuntu-server:~$
[redacted]@ubuntu-server:~$ mysqldump --databases employee db -udb_user -pPassw0rd > employeedb.sql
mysqldump: [Warning] Using a password on the command line interface can be insecure.
[redacted]@ubuntu-server:~$
[redacted]@ubuntu-server:~$
[redacted]@ubuntu-server:~$
[redacted]@ubuntu-server:~$ ls -l
total 8
-rw-rw-r-- 1 jigsaw jigsaw 2428 Jan  8 07:08 employeedb.sql
-rw-rw-r-- 1 jigsaw jigsaw  895 Dec 27 04:35 python-app-master.tar.gz
[redacted]@ubuntu-server:~$
```

3.1.2 Transfer database dump to AWS.

We shall transfer the db dump created in 3.1.1 to AWS for loading into RDS. Steps mentioned in following chapters. Nothing for now, just move further.

3.2 Setting up RDS Database on AWS using RDS service

We shall setup RDS database on AWS using RDS services.

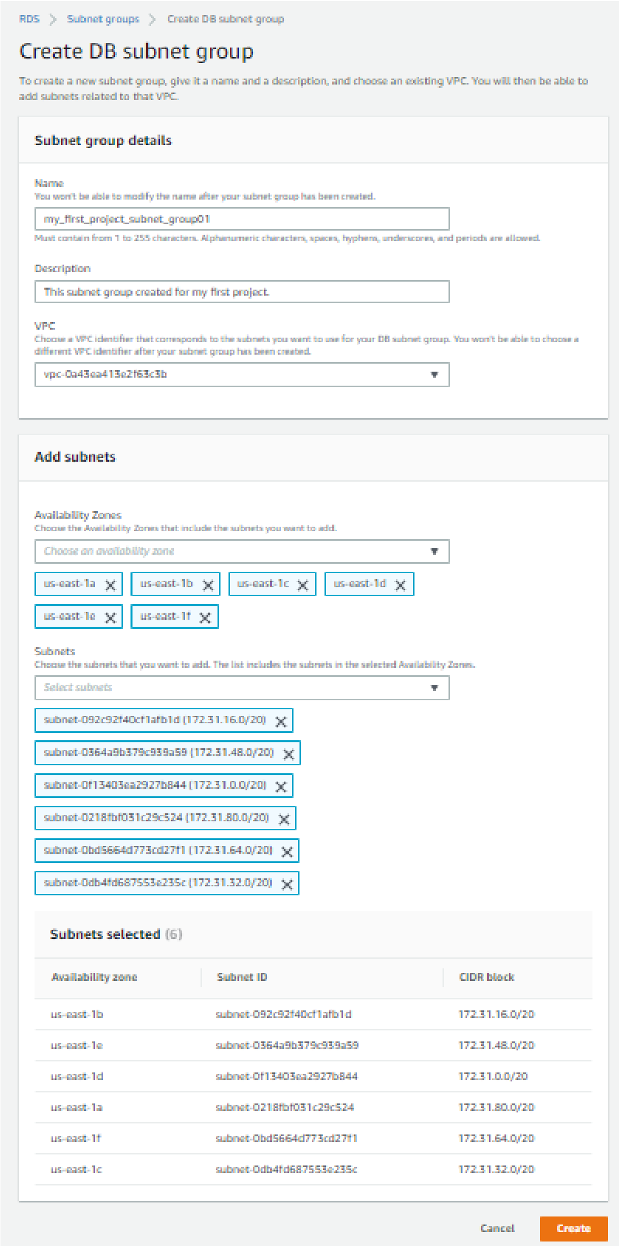
Following steps are involved in doing so.

3.2.1 Creating RDS Subnet Group using RDS service

An RDS Subnet Group is a collection of subnets that we can use to designate for our RDS database instance in a VPC.

Follow these steps to create a Security Group: my_first_project_subnet_group01

- Go to **RDS > Subnet groups > Create DB subnet group**
- Our VPC must have at least two subnets.
- These subnets must be in two different Availability Zones in the AWS Region where we want to deploy our DB instance.



Create DB subnet group

To create a new subnet group, give it a name and a description, and choose an existing VPC. You will then be able to add subnets related to that VPC.

Subnet group details

Name
You won't be able to modify the name after your subnet group has been created.
my_first_project_subnet_group01
Must contain from 1 to 255 characters. Alphanumeric characters, spaces, hyphens, underscores, and periods are allowed.

Description
This subnet group created for my first project.

VPC
Choose a VPC identifier that corresponds to the subnets you want to use for your DB subnet group. You won't be able to choose a different VPC identifier after your subnet group has been created.
vpc-0a43ea413e2f63c3b ▼

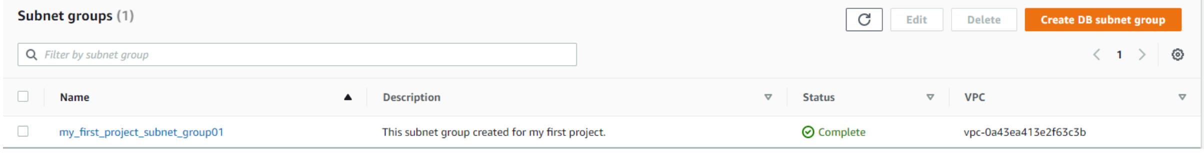
Add subnets

Availability Zones
Choose the Availability Zones that include the subnets you want to add.
Choose an availability zone ▼
us-east-1a ✕ us-east-1b ✕ us-east-1c ✕ us-east-1d ✕
us-east-1e ✕ us-east-1f ✕

Subnets
Choose the subnets that you want to add. The list includes the subnets in the selected Availability Zones.
Select subnets ▼
subnet-092c92f40cf1afb1d (172.31.16.0/20) ✕
subnet-0364a9b379c939a59 (172.31.48.0/20) ✕
subnet-0f13403ea2927b844 (172.31.0.0/20) ✕
subnet-0218fbf031c29c524 (172.31.80.0/20) ✕
subnet-0bd5664d773cd27f1 (172.31.64.0/20) ✕
subnet-0db4fd687553e235c (172.31.32.0/20) ✕

Subnets selected (6)

Availability zone	Subnet ID	CIDR block
us-east-1b	subnet-092c92f40cf1afb1d	172.31.16.0/20
us-east-1e	subnet-0364a9b379c939a59	172.31.48.0/20
us-east-1d	subnet-0f13403ea2927b844	172.31.0.0/20
us-east-1a	subnet-0218fbf031c29c524	172.31.80.0/20
us-east-1f	subnet-0bd5664d773cd27f1	172.31.64.0/20
us-east-1c	subnet-0db4fd687553e235c	172.31.32.0/20


Subnet groups (1)

☐	Name	Description	Status	VPC
☐	my_first_project_subnet_group01	This subnet group created for my first project.	✓ Complete	vpc-0a43ea413e2f63c3b

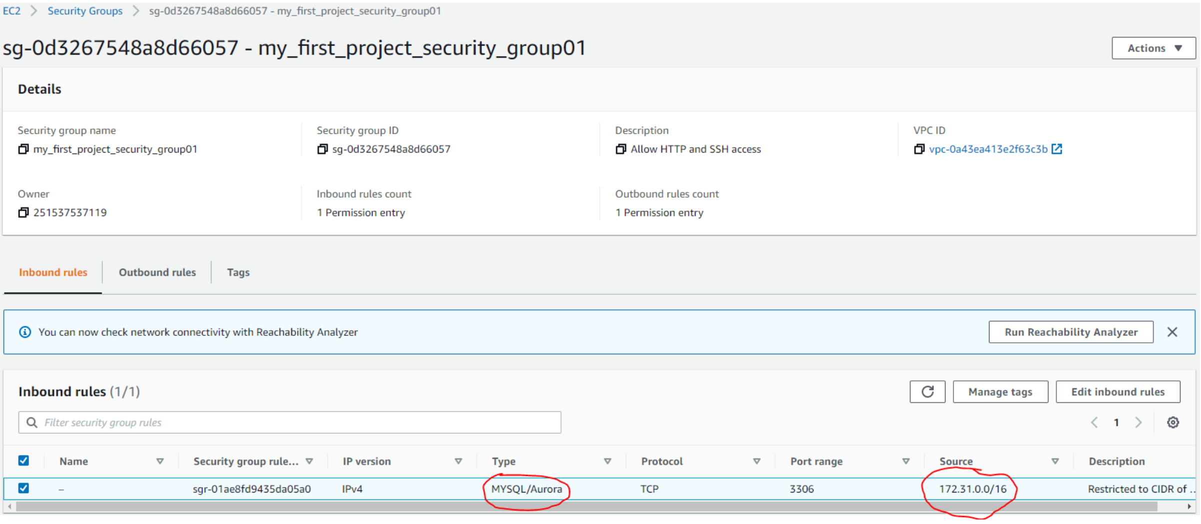
3.2.2 Creating RDS Security Group using EC2 Service

A *security group* acts as a virtual firewall for EC2 instances to control inbound and outbound traffic. When we launch an EC2 instance in a VPC, we can assign up to five security groups to the instance. Security groups act at the instance level, not the subnet level. Therefore, each instance in a subnet in our VPC can be assigned to a different set of security groups.

For each security group, we add *rules* that control the inbound traffic to instances, and a separate set of rules that control the outbound traffic.

Follow these steps to create a Security Group: my_first_project_security_group01

- Go to **EC2 > Security Groups > Create security group**
- Provide name and description
- Add Inbound rules
  - MYSQL/Aurora
  - Specify VPC CIDR range in the security group. This will restrict incoming traffic to RDS database to IPs which are in this CIDR range only.


EC2 > Security Groups > sg-0d3267548a8d66057 - my_first_project_security_group01
Actions ▼
Inbound rules
Outbound rules
Tags
Run Reachability Analyzer
Manage tags
Edit inbound rules
Filter security group rules
1

sg-0d3267548a8d66057 - my_first_project_security_group01

Details

Security group name	Security group ID	Description	VPC ID
my_first_project_security_group01	sg-0d3267548a8d66057	Allow HTTP and SSH access	vpc-0a43ea413e2f63c3b

Owner	Inbound rules count	Outbound rules count
251537537119	1 Permission entry	1 Permission entry

You can now check network connectivity with Reachability Analyzer

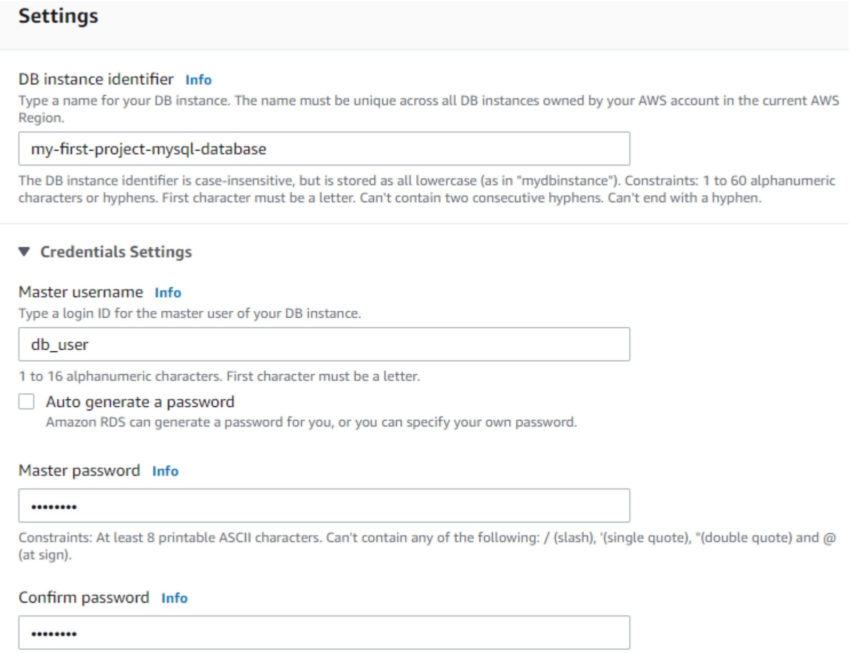
Inbound rules (1/1)

☑	Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description
☑	–	sgr-01ae8fd9435da05a0	IPv4	MYSQL/Aurora	TCP	3306	172.31.0.0/16	Restricted to CIDR of ..

3.2.3 Creating mysql database

Following these steps lets create database

- Go to **RDS > Databases > Create database**
- Choose
  - Standard create
  - MySQL
- Credentials Setting (Here we can set the user name and password we want to use to access this DB through our application)
  - DB instance name : my-first-project-mysql-database
  - Master username : db_user
  - Master password : Passw0rd



Settings

DB instance identifier Info

Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.

my-first-project-mysql-database

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

▼ Credentials Settings

Master username Info

Type a login ID for the master user of your DB instance.

db_user

1 to 16 alphanumeric characters. First character must be a letter.

□ Auto generate a password

Amazon RDS can generate a password for you, or you can specify your own password.

Master password Info

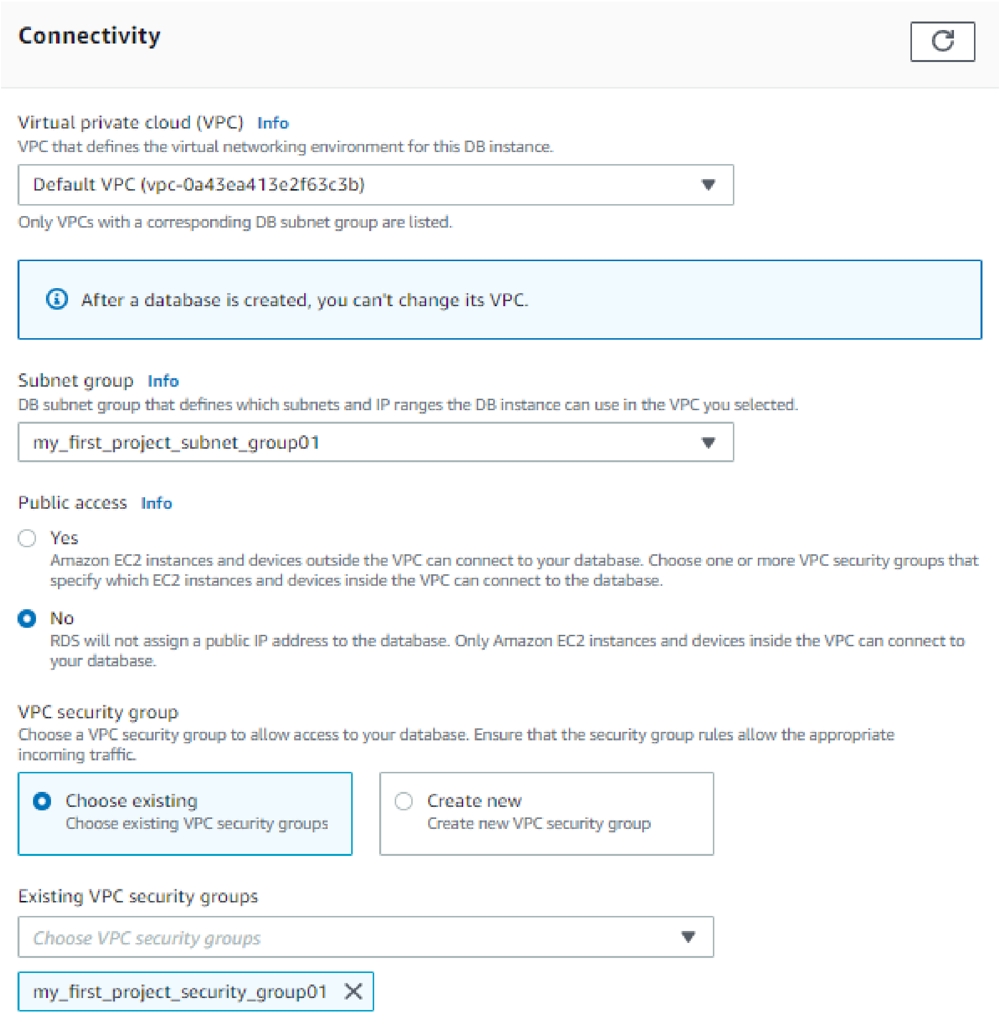
••••••••

Constraints: At least 8 printable ASCII characters. Can't contain any of the following: / (slash), '(single quote), "(double quote) and @ (at sign).

Confirm password Info

••••••••

- Connectivity
  - Subnet group : my_first_project_subnet_group01
  - Public access : No
  - VPC Security group
    - Choose existing : my_first_project_security_group01



Connectivity

Virtual private cloud (VPC) Info

VPC that defines the virtual networking environment for this DB instance.

Default VPC (vpc-0a43ea413e2f63c3b)

Only VPCs with a corresponding DB subnet group are listed.

After a database is created, you can't change its VPC.

Subnet group Info

DB subnet group that defines which subnets and IP ranges the DB instance can use in the VPC you selected.

my_first_project_subnet_group01

Public access Info

○ Yes
Amazon EC2 instances and devices outside the VPC can connect to your database. Choose one or more VPC security groups that specify which EC2 instances and devices inside the VPC can connect to the database.

● No
RDS will not assign a public IP address to the database. Only Amazon EC2 instances and devices inside the VPC can connect to your database.

VPC security group

Choose a VPC security group to allow access to your database. Ensure that the security group rules allow the appropriate incoming traffic.

● Choose existing
Choose existing VPC security groups

○ Create new
Create new VPC security group

Existing VPC security groups

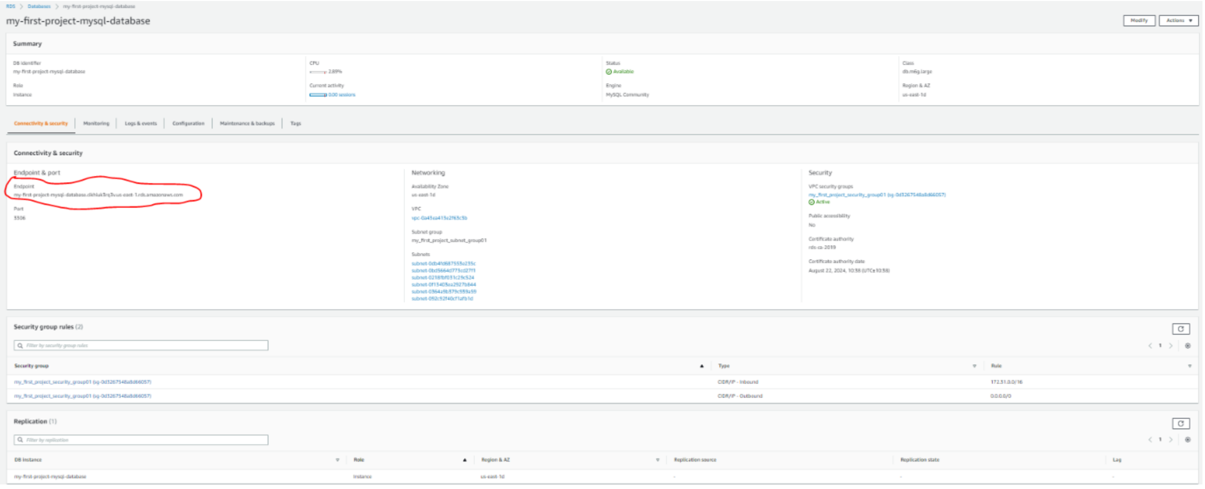
Choose VPC security groups

my_first_project_security_group01 ✕

- Finally click on Create database

Make a note of Endpoint. In this case it is:

my-first-project-mysql-database.cikhiuk3rq3v.us-east-1.rds.amazonaws.com



RDS > Databases > my-first-project-mysql-database

my-first-project-mysql-database

Summary

DB identifier
my-first-project-mysql-database

Role
Instance

CPU
2.89%

Current activity
0.00 sessions

Status
Available

Engine
MySQL Community

Class
db.m6g.large

Region & AZ
us-east-1d

Connectivity & security

Endpoint & port

Endpoint
my-first-project-mysql-database.cikhiuk3rq3v.us-east-1.rds.amazonaws.com

Port
3306

Networking

Availability Zone
us-east-1d

VPC
vpc-0a43ea413e2f63c3b

Subnet group
my_first_project_subnet_group01

Subnets
subnet-0db4fd687555e235c
subnet-0bd5664d773cd27f1
subnet-0218fbf031c29c524
subnet-0f13405ea2927b844
subnet-0364a9b379c939a59
subnet-092c92f40cf1afb1d

Security

VPC security groups
my_first_project_security_group01 (sg-0d3267548a8d66057)
Active

Public accessibility
No

Certificate authority
rds-ca-2019

Certificate authority date
August 22, 2024, 10:38 (UTC±10:38)

Security group rules (2)

Filter by security group rules

Security group	Type	Rule
my_first_project_security_group01 (sg-0d3267548a8d66057)	CIDR/IP - Inbound	172.31.0.0/16
my_first_project_security_group01 (sg-0d3267548a8d66057)	CIDR/IP - Outbound	0.0.0.0/0

Replication (1)

Filter by replication

DB instance	Role	Region & AZ	Replication source	Replication state	Lag
my-first-project-mysql-database	Instance	us-east-1d	-	-	-

Now we can create connection string to connect to our RDS mysql DB

Syntax: mysql -h<RDS Endpoint> -u<adminusername> -p<password>

$ mysql -hmy-first-project-mysql-database.cikhiuk3rq3v.us-east-1.rds.amazonaws.com -udb_user -pPassw0rd

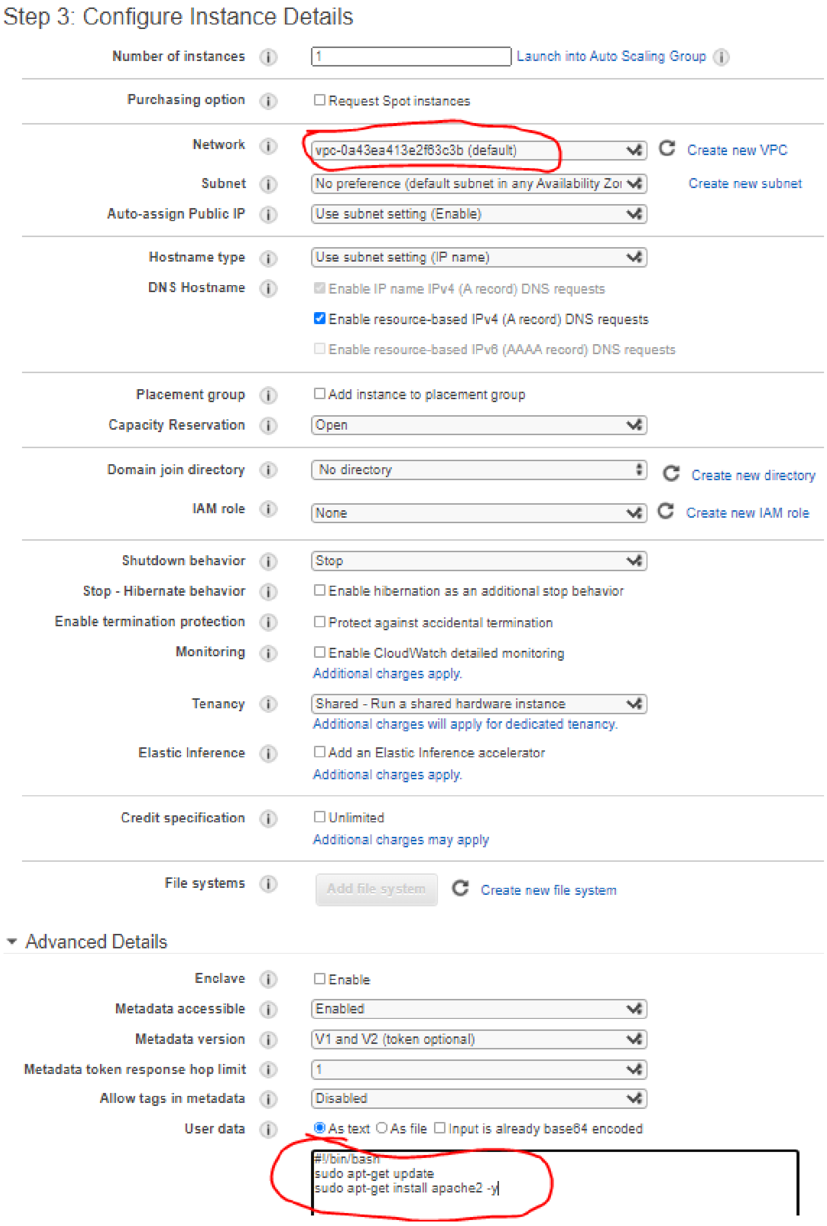
3.3 Creating EC2 instance

3.3.1 Setup Steps

Following these steps lets launch EC2 instance:

- Go to **EC2 Dashboard > Launch Instance**
- Choose an Amazon Machine Image (AMI)
  - Free tier eligible: Ubuntu Server 20.04 LTS (HVM), SSD Volume Type
- Choose an Instance Type
  - Free tier eligible
    - Family : t2
    - Type: t2.micro
    - vCPU :1
    - Memory(GiB) : 1
- Configure Instance Details
- User data – to update Linux and then install Web server (apache2) when instance is launched

```
#!/bin/bash
sudo apt-get update
sudo apt-get install apache2 -y
```



Step 3: Configure Instance Details

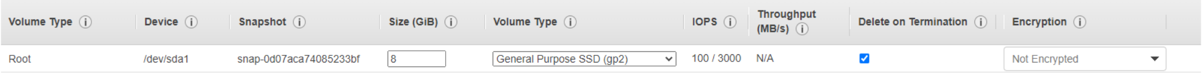
Number of instances ⓘ	1 — Launch into Auto Scaling Group ⓘ
Purchasing option ⓘ	□ Request Spot instances
Network ⓘ	vpc-0a43ea413e2f63c3b (default) — Create new VPC
Subnet ⓘ	No preference (default subnet in any Availability Zo — Create new subnet
Auto-assign Public IP ⓘ	Use subnet setting (Enable)
Hostname type ⓘ	Use subnet setting (IP name)
DNS Hostname ⓘ	☑ Enable IP name IPv4 (A record) DNS requests ☑ Enable resource-based IPv4 (A record) DNS requests □ Enable resource-based IPv6 (AAAA record) DNS requests
Placement group ⓘ	□ Add instance to placement group
Capacity Reservation ⓘ	Open
Domain join directory ⓘ	No directory — Create new directory
IAM role ⓘ	None — Create new IAM role
Shutdown behavior ⓘ	Stop
Stop - Hibernate behavior ⓘ	□ Enable hibernation as an additional stop behavior
Enable termination protection ⓘ	□ Protect against accidental termination
Monitoring ⓘ	□ Enable CloudWatch detailed monitoring Additional charges apply.
Tenancy ⓘ	Shared - Run a shared hardware instance Additional charges will apply for dedicated tenancy.
Elastic Inference ⓘ	□ Add an Elastic Inference accelerator Additional charges apply.
Credit specification ⓘ	□ Unlimited Additional charges may apply
File systems ⓘ	Add file system — Create new file system

Advanced Details

Enclave ⓘ	□ Enable
Metadata accessible ⓘ	Enabled
Metadata version ⓘ	V1 and V2 (token optional)
Metadata token response hop limit ⓘ	1
Allow tags in metadata ⓘ	Disabled
User data ⓘ	● As text ○ As file □ Input is already base64 encoded

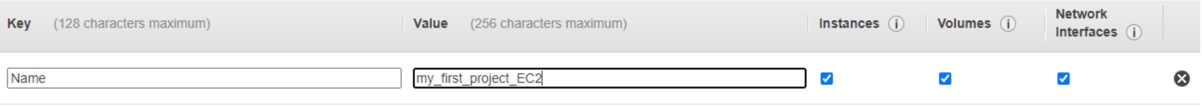
```
#!/bin/bash
sudo apt-get update
sudo apt-get install apache2 -y
```

- Add Storage



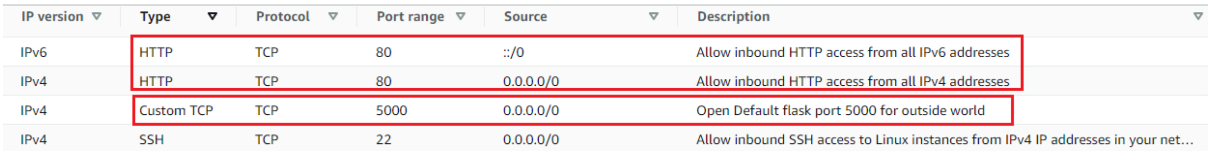
Volume Type ⓘ	Device ⓘ	Snapshot ⓘ	Size (GiB) ⓘ	Volume Type ⓘ	IOPS ⓘ	Throughput (MB/s) ⓘ	Delete on Termination ⓘ	Encryption ⓘ
Root	/dev/sda1	snap-0d07aca74085233bf	8	General Purpose SSD (gp2)	100 / 3000	N/A	☑	Not Encrypted

- Add Tag



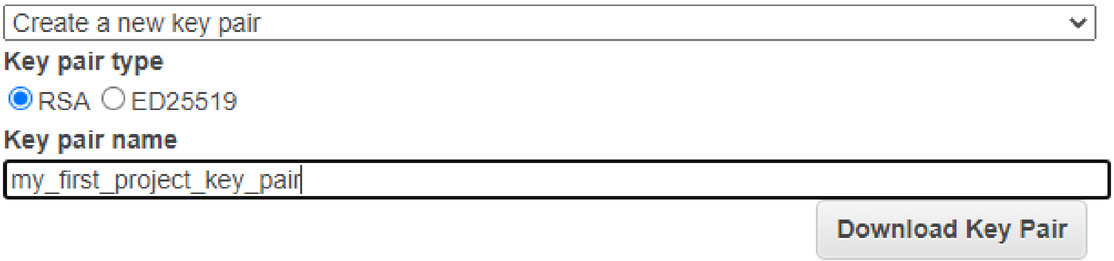
Key (128 characters maximum)	Value (256 characters maximum)	Instances ⓘ	Volumes ⓘ	Network Interfaces ⓘ	
Name	my_first_project_EC2	☑	☑	☑	⊗

- Configure Security Group
  Here we shall create a new security group "my_first_project_security_group02"
  This security group will allow following access on this EC2 instance:
  - SSH access on port 22
  - HTTP access on port 80
  - HTTP access on port 5000 (flask default testing port)



IP version	Type	Protocol	Port range	Source	Description
IPv6	HTTP	TCP	80	::/0	Allow inbound HTTP access from all IPv6 addresses
IPv4	HTTP	TCP	80	0.0.0.0/0	Allow inbound HTTP access from all IPv4 addresses
IPv4	Custom TCP	TCP	5000	0.0.0.0/0	Open Default flask port 5000 for outside world
IPv4	SSH	TCP	22	0.0.0.0/0	Allow inbound SSH access to Linux instances from IPv4 IP addresses in your net…

- Review Instance
- Create a new key pair



Create a new key pair

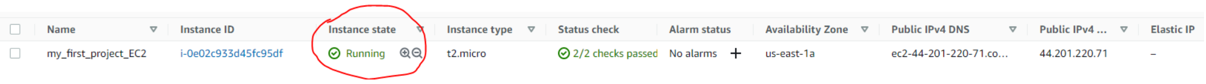
Key pair type

● RSA ○ ED25519

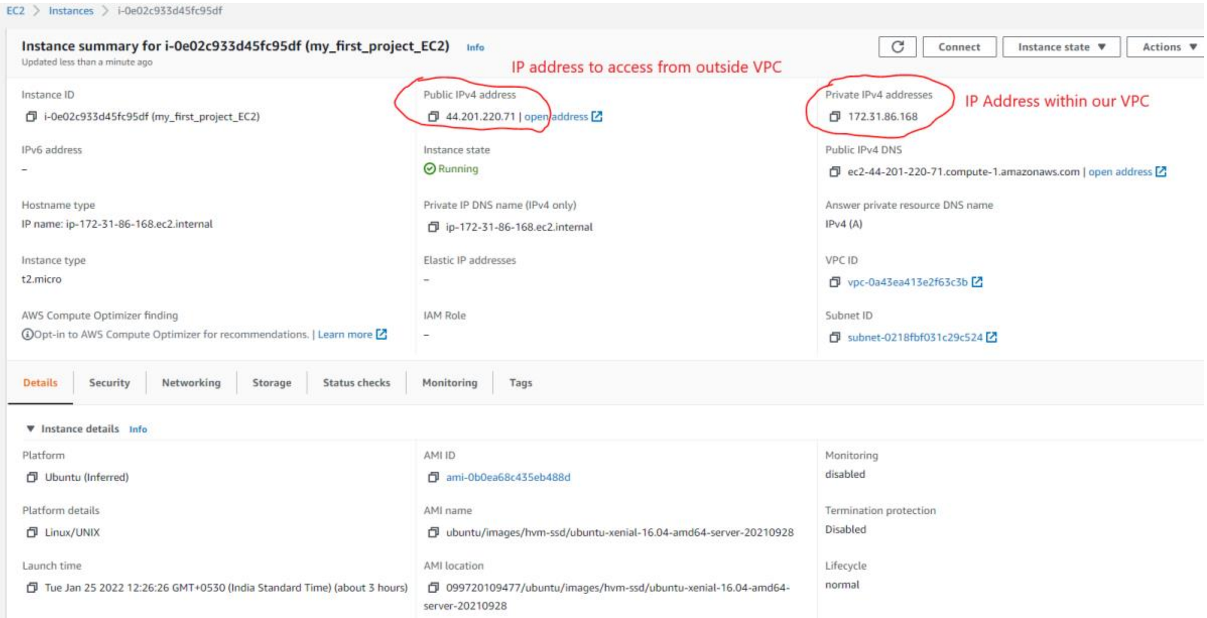
Key pair name

my_first_project_key_pair

- Launch



□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
□	my_first_project_EC2	i-0e02c933d45fc95df	Running	t2.micro	2/2 checks passed	No alarms +	us-east-1a	ec2-44-201-220-71.co...	44.201.220.71	–



Instance summary for i-0e02c933d45fc95df (my_first_project_EC2) Info
Updated less than a minute ago

IP address to access from outside VPC

IP Address within our VPC

Instance ID
i-0e02c933d45fc95df (my_first_project_EC2)

Public IPv4 address
44.201.220.71 | open address

Private IPv4 addresses
172.31.86.168

IPv6 address
–

Instance state
Running

Public IPv4 DNS
ec2-44-201-220-71.compute-1.amazonaws.com | open address

Hostname type
IP name: ip-172-31-86-168.ec2.internal

Private IP DNS name (IPv4 only)
ip-172-31-86-168.ec2.internal

Answer private resource DNS name
IPv4 (A)

Instance type
t2.micro

Elastic IP addresses
–

VPC ID
vpc-0a43ea413e2f63c3b

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

IAM Role
–

Subnet ID
subnet-0218fbf031c29c524

▼ Instance details Info

Platform
Ubuntu (Inferred)

AMI ID
ami-0b0ea68c435eb488d

Monitoring
disabled

Platform details
Linux/UNIX

AMI name
ubuntu/images/hvm-ssd/ubuntu-xenial-16.04-amd64-server-20210928

Termination protection
Disabled

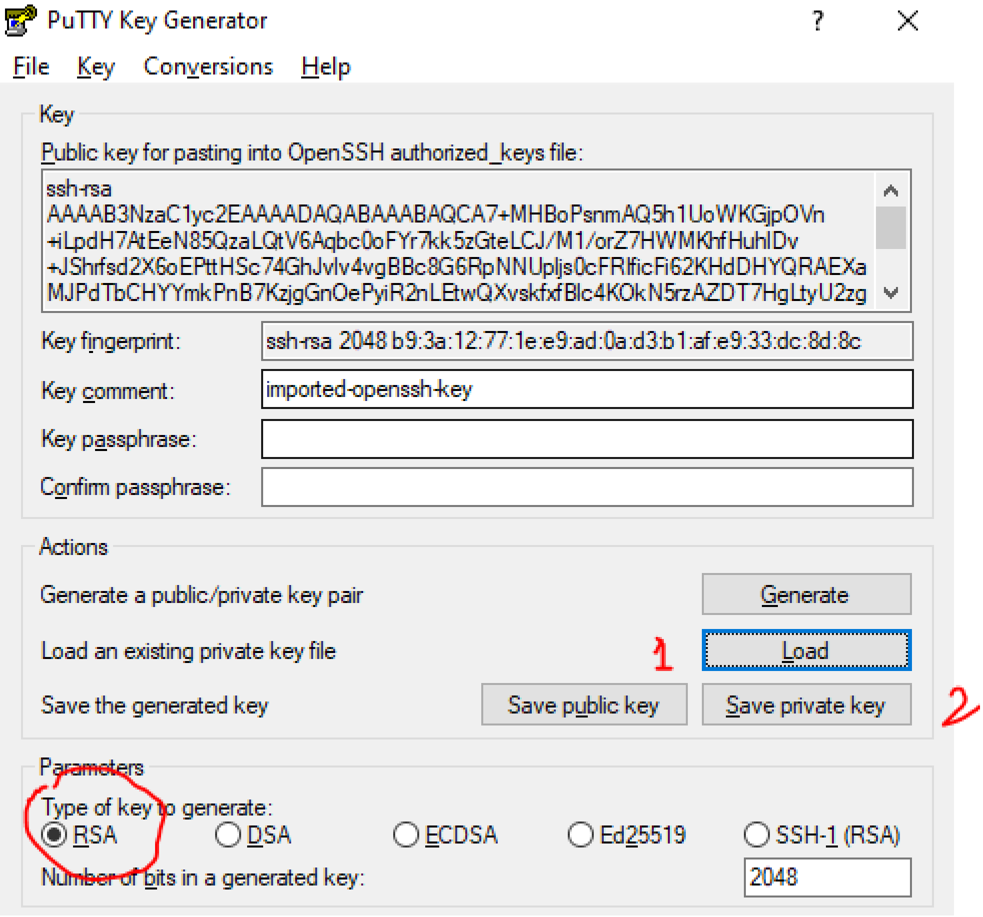
Launch time
Tue Jan 25 2022 12:26:26 GMT+0530 (India Standard Time) (about 3 hours)

AMI location
099720109477/ubuntu/images/hvm-ssd/ubuntu-xenial-16.04-amd64-server-20210928

Lifecycle
normal

3.3.2 Generating .ppk file for using Putty to connect to EC2 instance

- Go to window search > PuTTYgen > Open
- Load key-pair file my_first_project_key_pair.pem
- Save private key
- Select the same location as .pem file and provide the same file name with .ppk extension as my_first_project_key_pair.ppk



Public key for pasting into OpenSSH authorized_keys file:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQCA7+MHBoPsnmAQ5h1UoWKGjpOVn
+iLpdH7AtEeN85QzaLQtV6Aqbc0oFYr7kk5zGteLCJ/M1/orZ7HWMKhfHuhlDv
+JShrfsd2X6oEPttHSc74GhJvlv4vgBBc8G6RpNNUpljs0cFRlficFi62KHdDHYQRAEXa
MJPdTbCHYYmkPnB7KzjgGnOePyiR2nLEtwQXvskfxfBlc4KOkN5rzAZDT7HgLtyU2zg
```

Key fingerprint: ssh-rsa 2048 b9:3a:12:77:1e:e9:ad:0a:d3:b1:af:e9:33:dc:8d:8c

Key comment: imported-openssh-key

Key passphrase:

Confirm passphrase:

Generate a public/private key pair

Load an existing private key file

Save the generated key

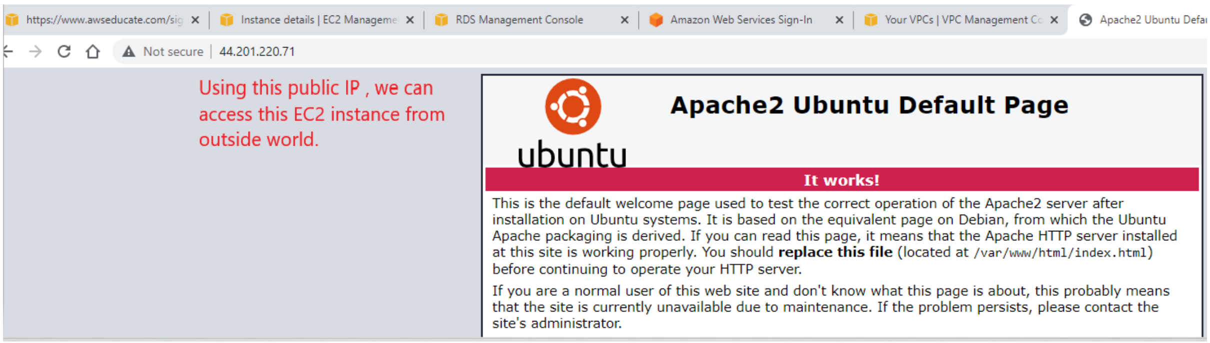
Type of key to generate:

● RSA ○ DSA ○ ECDSA ○ Ed25519 ○ SSH-1 (RSA)

Number of bits in a generated key: 2048

3.3.3 Verification Steps

Verify if EC2 instance accessible from outside world on HTTP using public IP.


https://www.awseducate.com/sig
Instance details | EC2 Manageme
RDS Management Console
Amazon Web Services Sign-In
Your VPCs | VPC Management C
Apache2 Ubuntu Defa
Not secure
44.201.220.71
ubuntu

Using this public IP , we can access this EC2 instance from outside world.

Apache2 Ubuntu Default Page

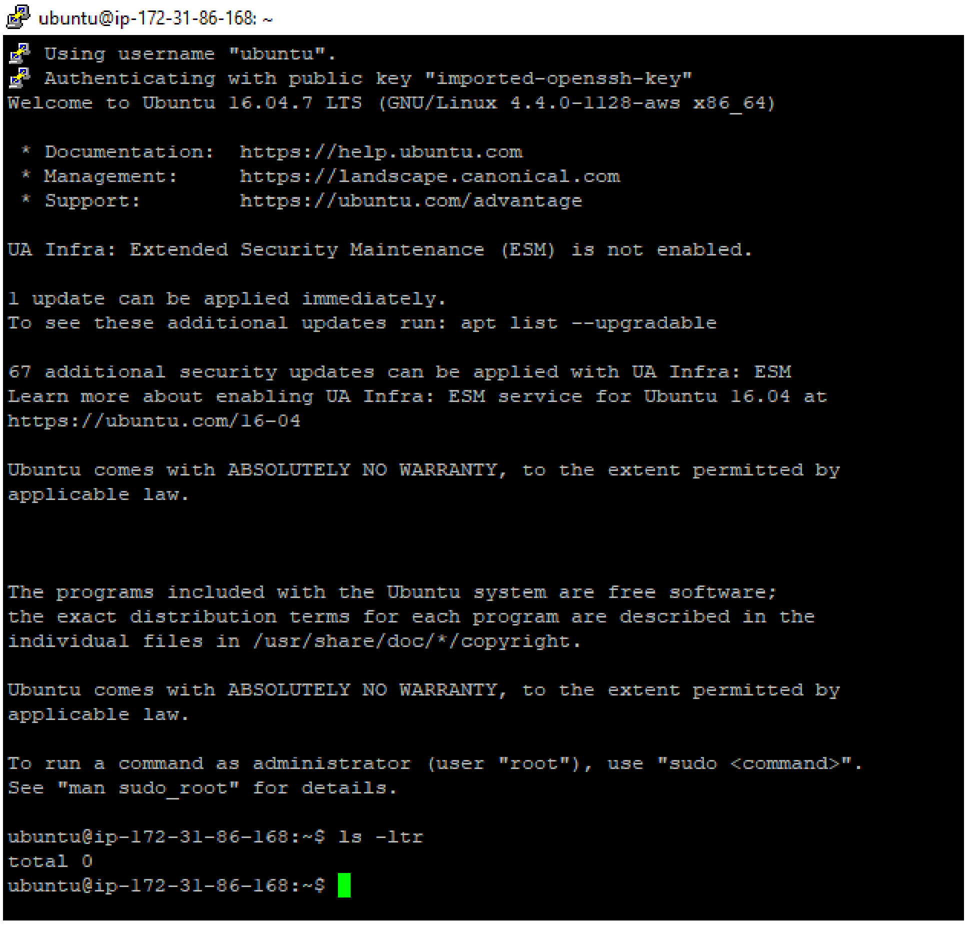
It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Verify if EC2 instance accessible from outside world on SSH(using PuTTY) using public IP.

- **Host name :** ubuntu@44.201.220.71
- **SSH > Auth > Browse** then select my_first_project_key_pair.ppk file
- Click on **Open**

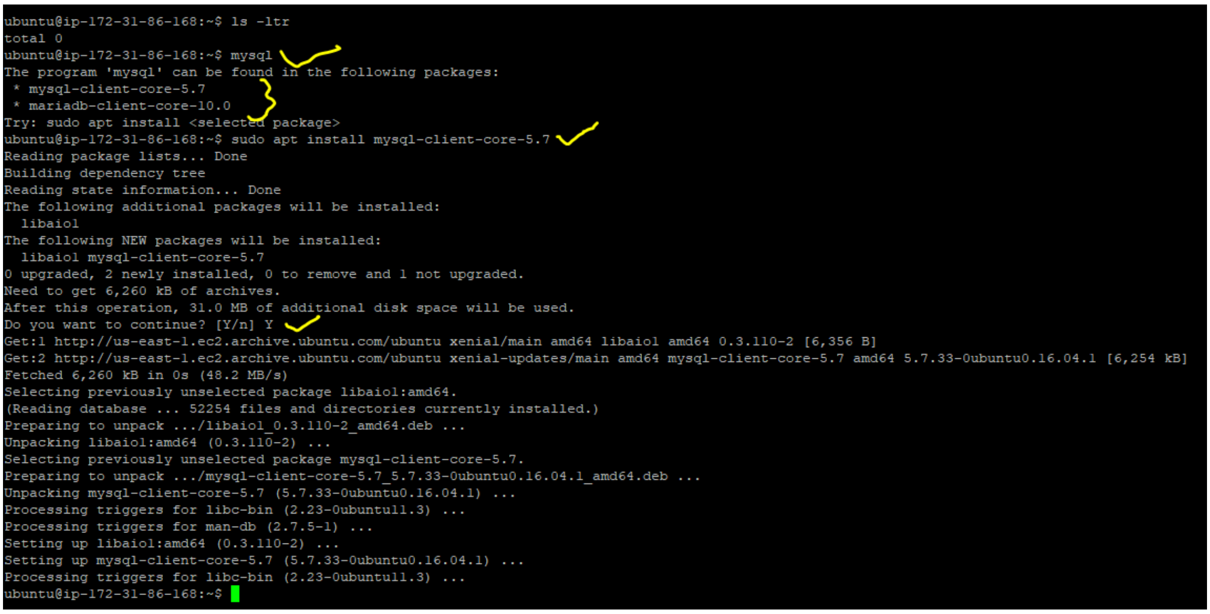


ubuntu@ip-172-31-86-168: ~

```
Using username "ubuntu".
Authenticating with public key "imported-openssh-key"
Welcome to Ubuntu 16.04.7 LTS (GNU/Linux 4.4.0-1128-aws x86_64)

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

UA Infra: Extended Security Maintenance (ESM) is not enabled.

1 update can be applied immediately.
To see these additional updates run: apt list --upgradable

67 additional security updates can be applied with UA Infra: ESM
Learn more about enabling UA Infra: ESM service for Ubuntu 16.04 at
https://ubuntu.com/16-04

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.



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@ip-172-31-86-168:~$ ls -ltr
total 0
ubuntu@ip-172-31-86-168:~$
```

3.3.4 Installing mysql client on EC2 instance

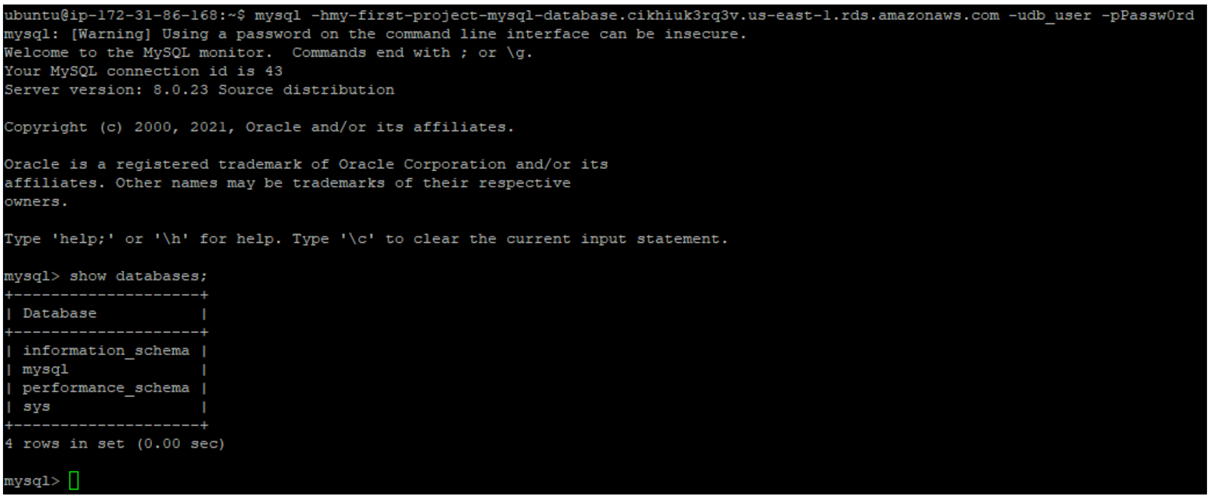
Once we are able to access our EC2 instance using public IP on SSH, next thing to install mysql client.



```
ubuntu@ip-172-31-86-168:~$ ls -ltr
total 0
ubuntu@ip-172-31-86-168:~$ mysql
The program 'mysql' can be found in the following packages:
 * mysql-client-core-5.7
 * mariadb-client-core-10.0
Try: sudo apt install <selected package>
ubuntu@ip-172-31-86-168:~$ sudo apt install mysql-client-core-5.7
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libaiol
The following NEW packages will be installed:
  libaiol mysql-client-core-5.7
0 upgraded, 2 newly installed, 0 to remove and 1 not upgraded.
Need to get 6,260 kB of archives.
After this operation, 31.0 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 xenial/main amd64 libaiol amd64 0.3.110-2 [6,356 B]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu xenial-updates/main amd64 mysql-client-core-5.7 amd64 5.7.33-0ubuntu0.16.04.1 [6,254 kB]
Fetched 6,260 kB in 0s (48.2 MB/s)
Selecting previously unselected package libaiol:amd64.
(Reading database ... 52254 files and directories currently installed.)
Preparing to unpack .../libaiol_0.3.110-2_amd64.deb ...
Unpacking libaiol:amd64 (0.3.110-2) ...
Selecting previously unselected package mysql-client-core-5.7.
Preparing to unpack .../mysql-client-core-5.7_5.7.33-0ubuntu0.16.04.1_amd64.deb ...
Unpacking mysql-client-core-5.7 (5.7.33-0ubuntu0.16.04.1) ...
Processing triggers for libc-bin (2.23-0ubuntu11.3) ...
Processing triggers for man-db (2.7.5-1) ...
Setting up libaiol:amd64 (0.3.110-2) ...
Setting up mysql-client-core-5.7 (5.7.33-0ubuntu0.16.04.1) ...
Processing triggers for libc-bin (2.23-0ubuntu11.3) ...
ubuntu@ip-172-31-86-168:~$
```

Connecting to our RDS database using this mysql client using RDS checkpoint.

$ mysql -hmy-first-project-mysql-database.cikhiuk3rq3v.us-east-1.rds.amazonaws.com -udb_user -pPassw0rd



```
ubuntu@ip-172-31-86-168:~$ mysql -hmy-first-project-mysql-database.cikhiuk3rq3v.us-east-1.rds.amazonaws.com -udb_user -pPassw0rd
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 43
Server version: 8.0.23 Source distribution

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
4 rows in set (0.00 sec)

mysql>
```

3.3.5 Installing Python packages

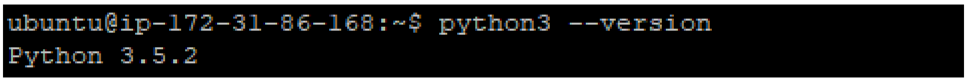
Application which we are migrating is written in python.

We need to install:

- pip3 – to install dependencies

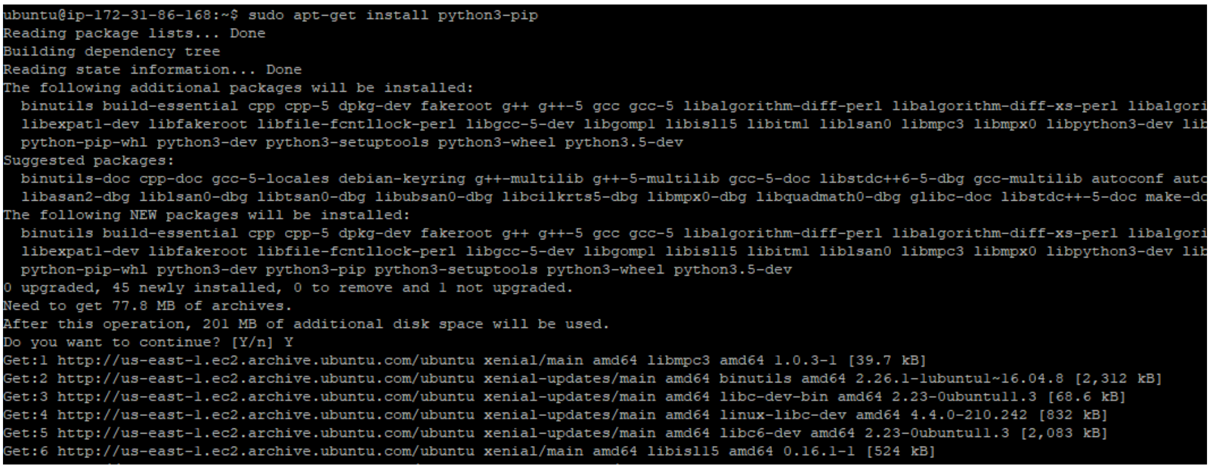
- Dependency packages
    - flask
    - flask-mysql

Check python version



```
ubuntu@ip-172-31-86-168:~$ python3 --version
Python 3.5.2
```

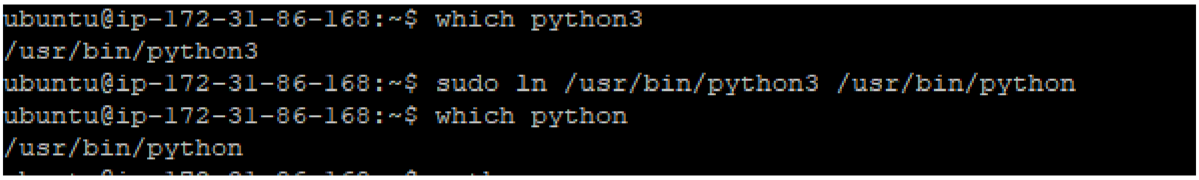
- Install pip3



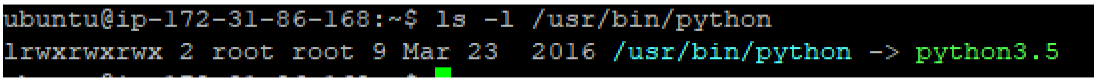
```
ubuntu@ip-172-31-86-168:~$ sudo apt-get install python3-pip
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  binutils build-essential cpp cpp-5 dpkg-dev fakeroot g++ g++-5 gcc gcc-5 libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgori
  libexpat1-dev libfakeroot libfile-fcntllock-perl libgcc-5-dev libgomp1 libisl15 libitm1 liblsan0 libmpc3 libmpx0 libpython3-dev lib
  python-pip-whl python3-dev python3-setuptools python3-wheel python3.5-dev
Suggested packages:
  binutils-doc cpp-doc gcc-5-locales debian-keyring g++-multilib g++-5-multilib gcc-5-doc libstdc++6-5-dbg gcc-multilib autoconf autc
  libasan2-dbg liblsan0-dbg libtsan0-dbg libubsan0-dbg libcilkrts5-dbg libmpx0-dbg libquadmath0-dbg glibc-doc libstdc++-5-doc make-dc
The following NEW packages will be installed:
  binutils build-essential cpp cpp-5 dpkg-dev fakeroot g++ g++-5 gcc gcc-5 libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgori
  libexpat1-dev libfakeroot libfile-fcntllock-perl libgcc-5-dev libgomp1 libisl15 libitm1 liblsan0 libmpc3 libmpx0 libpython3-dev lib
  python-pip-whl python3-dev python3-pip python3-setuptools python3-wheel python3.5-dev
0 upgraded, 45 newly installed, 0 to remove and 1 not upgraded.
Need to get 77.8 MB of archives.
After this operation, 201 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 xenial/main amd64 libmpc3 amd64 1.0.3-1 [39.7 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu xenial-updates/main amd64 binutils amd64 2.26.1-1ubuntu1~16.04.8 [2,312 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu xenial-updates/main amd64 libc-dev-bin amd64 2.23-0ubuntu11.3 [68.6 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu xenial-updates/main amd64 linux-libc-dev amd64 4.4.0-210.242 [832 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu xenial-updates/main amd64 libc6-dev amd64 2.23-0ubuntu11.3 [2,083 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu xenial/main amd64 libisl15 amd64 0.16.1-1 [524 kB]
```

- Installing dependencies flask & flask-mysql
    - $ sudo pip3 install flask
    - $ sudo pip3 install flask-mysql

3.3.6 Creating soft link for python from python3



```
ubuntu@ip-172-31-86-168:~$ which python3
/usr/bin/python3
ubuntu@ip-172-31-86-168:~$ sudo ln /usr/bin/python3 /usr/bin/python
ubuntu@ip-172-31-86-168:~$ which python
/usr/bin/python
```



```
ubuntu@ip-172-31-86-168:~$ ls -l /usr/bin/python
lrwxrwxrwx 2 root root 9 Mar 23  2016 /usr/bin/python -> python3.5
```

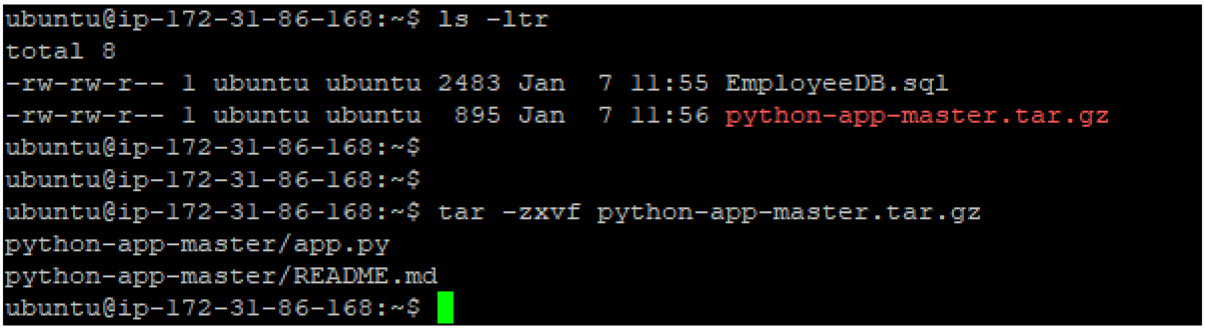
3.3.7 Transfer the project files/packages to EC2 instance

Using WinSCP lets transfer the project file/packages on EC2 instance.

- Go to **WinSCP**
- Host name: 44.201.220.71
- User name: ubuntu

- Password: click on Advance > SSH > Authentication and then select this file: my_first_project_key_pair.ppk
- Click on Login

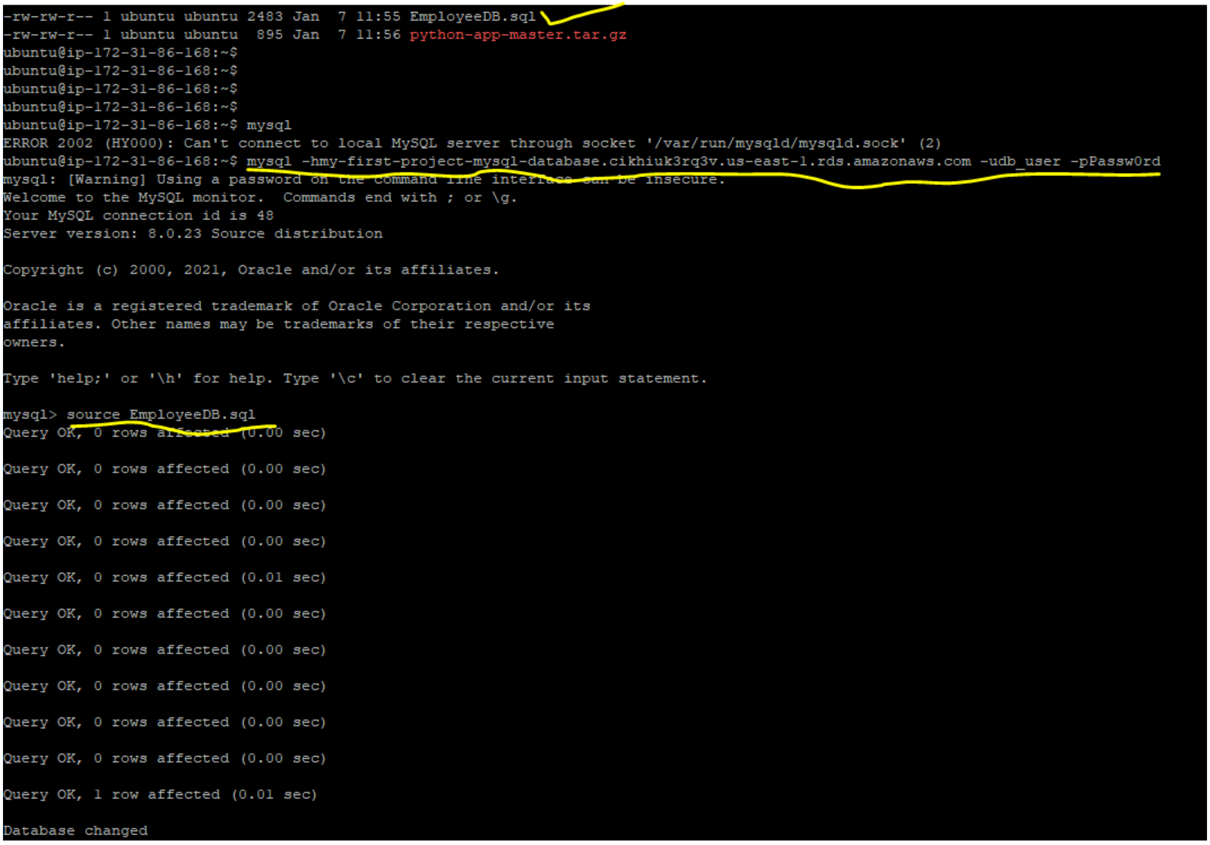
3.3.8 Extract project files/packages on EC2 instance



```
ubuntu@ip-172-31-86-168:~$ ls -ltr
total 8
-rw-rw-r-- 1 ubuntu ubuntu 2483 Jan  7 11:55 EmployeeDB.sql
-rw-rw-r-- 1 ubuntu ubuntu  895 Jan  7 11:56 python-app-master.tar.gz
ubuntu@ip-172-31-86-168:~$
ubuntu@ip-172-31-86-168:~$
ubuntu@ip-172-31-86-168:~$ tar -zxvf python-app-master.tar.gz
python-app-master/app.py
python-app-master/README.md
ubuntu@ip-172-31-86-168:~$
```

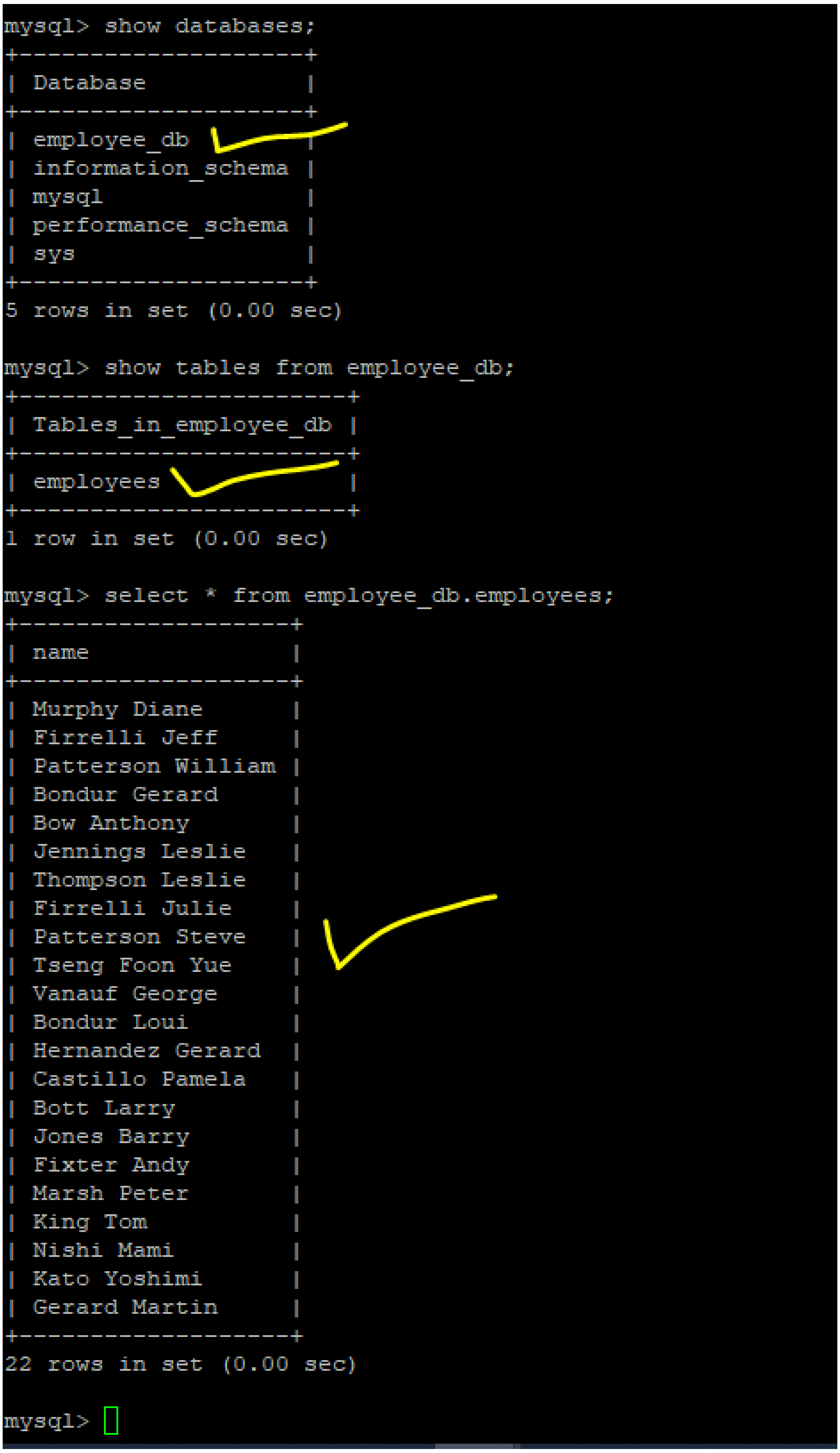
3.3.9 Load database dump into RDS database

Now from EC2 instance using mysql client, we shall load our database dump file (EmployeeDB.sql) to RDS database.



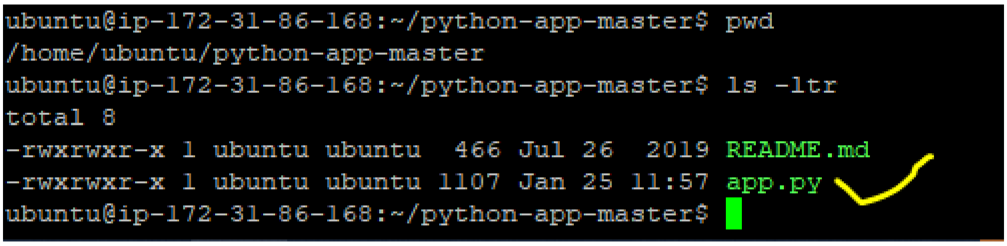
```
-rw-rw-r-- 1 ubuntu ubuntu 2483 Jan  7 11:55 EmployeeDB.sql
-rw-rw-r-- 1 ubuntu ubuntu  895 Jan  7 11:56 python-app-master.tar.gz
ubuntu@ip-172-31-86-168:~$
ubuntu@ip-172-31-86-168:~$
ubuntu@ip-172-31-86-168:~$
ubuntu@ip-172-31-86-168:~$
ubuntu@ip-172-31-86-168:~$ mysql
ERROR 2002 (HY000): Can't connect to local MySQL server through socket '/var/run/mysqld/mysqld.sock' (2)
ubuntu@ip-172-31-86-168:~$ mysql -hmy-first-project-mysql-database.cikhiuk3rq3v.us-east-1.rds.amazonaws.com -udb_user -pPassw0rd
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 48
Server version: 8.0.23 Source distribution

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> source EmployeeDB.sql
Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.01 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 1 row affected (0.01 sec)

Database changed
```

3.3.10Verify if data loading is successful

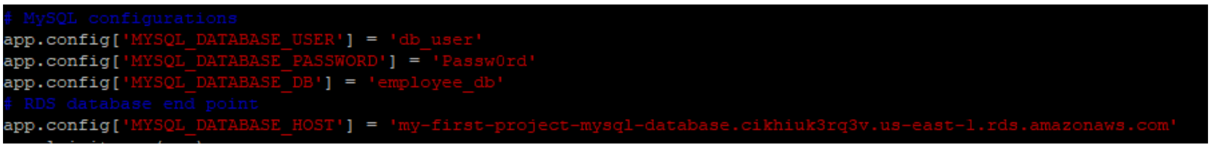


```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| employee_db        |
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.00 sec)

mysql> show tables from employee_db;
+-----------------------+
| Tables_in_employee_db |
+-----------------------+
| employees             |
+-----------------------+
1 row in set (0.00 sec)

mysql> select * from employee_db.employees;
+-------------------+
| name              |
+-------------------+
| Murphy Diane      |
| Firrelli Jeff     |
| Patterson William |
| Bondur Gerard     |
| Bow Anthony       |
| Jennings Leslie   |
| Thompson Leslie   |
| Firrelli Julie    |
| Patterson Steve   |
| Tseng Foon Yue    |
| Vanauf George     |
| Bondur Loui       |
| Hernandez Gerard  |
| Castillo Pamela   |
| Bott Larry        |
| Jones Barry       |
| Fixter Andy       |
| Marsh Peter       |
| King Tom          |
| Nishi Mami        |
| Kato Yoshimi      |
| Gerard Martin     |
+-------------------+
22 rows in set (0.00 sec)

mysql> 
```

3.3.11Configure database detail in application file app.py



```
ubuntu@ip-172-31-86-168:~/python-app-master$ pwd
/home/ubuntu/python-app-master
ubuntu@ip-172-31-86-168:~/python-app-master$ ls -ltr
total 8
-rwxrwxr-x 1 ubuntu ubuntu  466 Jul 26  2019 README.md
-rwxrwxr-x 1 ubuntu ubuntu 1107 Jan 25 11:57 app.py
ubuntu@ip-172-31-86-168:~/python-app-master$
```

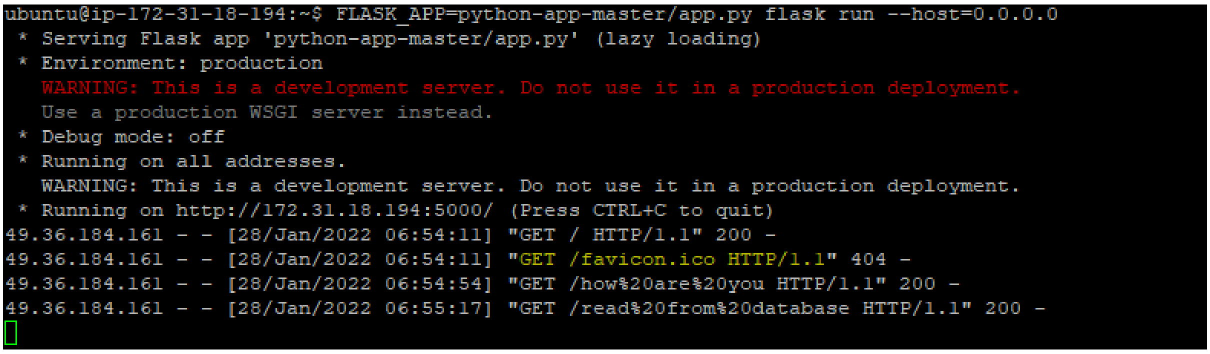


```
# MySQL configurations
app.config['MYSQL_DATABASE_USER'] = 'db_user'
app.config['MYSQL_DATABASE_PASSWORD'] = 'Passw0rd'
app.config['MYSQL_DATABASE_DB'] = 'employee_db'
# RDS database end point
app.config['MYSQL_DATABASE_HOST'] = 'my-first-project-mysql-database.cikhiuk3rq3v.us-east-1.rds.amazonaws.com'
```

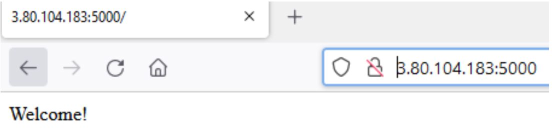
3.4 Running application on port 5000

Now let’s run our application.

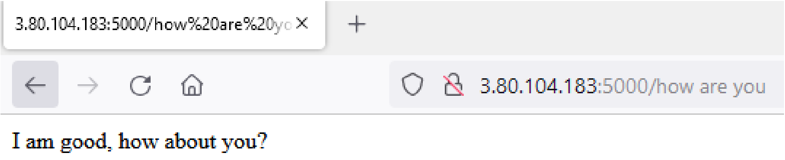
$ FLASK_APP=python-app-master/app.py flask run --host=0.0.0.0



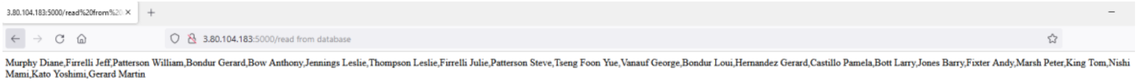
```
ubuntu@ip-172-31-18-194:~$ FLASK_APP=python-app-master/app.py flask run --host=0.0.0.0
 * Serving Flask app 'python-app-master/app.py' (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on all addresses.
   WARNING: This is a development server. Do not use it in a production deployment.
 * Running on http://172.31.18.194:5000/ (Press CTRL+C to quit)
49.36.184.161 - - [28/Jan/2022 06:54:11] "GET / HTTP/1.1" 200 -
49.36.184.161 - - [28/Jan/2022 06:54:11] "GET /favicon.ico HTTP/1.1" 404 -
49.36.184.161 - - [28/Jan/2022 06:54:54] "GET /how%20are%20you HTTP/1.1" 200 -
49.36.184.161 - - [28/Jan/2022 06:55:17] "GET /read%20from%20database HTTP/1.1" 200 -
```



Welcome!


3.80.104.183:5000/how%20are%20yo
3.80.104.183:5000/how are you

I am good, how about you?


3.80.104.183:5000/read%20from%20
3.80.104.183:5000/read from database

Murphy Diane,Firrelli Jeff,Patterson William,Bondur Gerard,Bow Anthony,Jennings Leslie,Thompson Leslie,Firrelli Julie,Patterson Steve,Tseng Foon Yue,Vanauf George,Bondur Loui,Hernandez Gerard,Castillo Pamela,Bott Larry,Jones Barry,Fixter Andy,Marsh Peter,King Tom,Nishi Mami,Kato Yoshimi,Gerard Martin

3.5 Running application on port HTTP 80

Our flask-based web application is running on dec/test port 5000 successfully. Now we need to configure our application to run on standard http port 80 so that end user does not experience any change in their app usage. To do this we need to setup WSGI (Web Server Gateway Interface Module)

3.5.1 Installing and Enabling Web Server Gateway Interface Module (WSGI)

Follow these steps to setup WSGI

- Install wsgi
  ```
  $ sudo apt-get install libapache2-mod-wsgi-py3 python-dev
  ```
- Module wsgi gets enabled by default after installation. We can give the following command to make sure Module wsgi is enabled.
  ```
  $ sudo a2enmod wsgi
  ```

3.5.2 Setup flask application

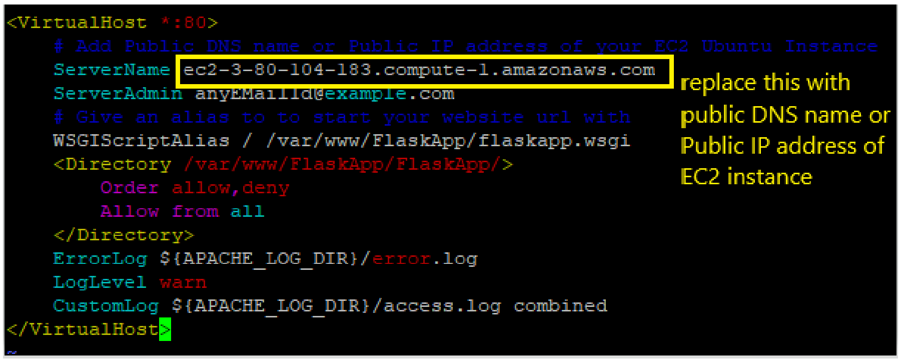
- List the contents of the default web pages directory.
  ```
  $ ls -l /var/www
  ```
- Create the required sub-directories in the above.
  ```
  $ sudo mkdir -p /var/www/FlaskApp/FlaskApp
  ```
- Create an empty __init__.py file and copy the python application.
  ```
  $ sudo touch /var/www/FlaskApp/FlaskApp/__init__.py
  $ sudo cp python-app-master/app.py /var/www/FlaskApp/FlaskApp/
  ```

- Configure and Enable a New Virtual Host at port number 80. To do this we need to create a configuration file for the FlaskApp.
  ```
  $ sudo vi /etc/apache2/sites-available/FlaskApp.conf
  ```

  Then copy paste the follwing lines.

  ```
  <VirtualHost *:80>
     # Add Public DNS name or Public IP address of your EC2 Ubuntu Instance
     ServerName Here Add Public DNS name or Public IP address of your EC2 Ubuntu Instance
     ServerAdmin anyEMailId@example.com
     # Give an alias to to start your website url with
     WSGIScriptAlias / /var/www/FlaskApp/flaskapp.wsgi
     <Directory /var/www/FlaskApp/FlaskApp/>
        Order allow,deny
        Allow from all
     </Directory>
     ErrorLog ${APACHE_LOG_DIR}/error.log
  ```

```
    LogLevel warn
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```



```
<VirtualHost *:80>
    # Add Public DNS name or Public IP address of your EC2 Ubuntu Instance
    ServerName ec2-3-80-104-183.compute-1.amazonaws.com
    ServerAdmin anyEMailId@example.com
    # Give an alias to to start your website url with
    WSGIScriptAlias / /var/www/FlaskApp/flaskapp.wsgi
    <Directory /var/www/FlaskApp/FlaskApp/>
        Order allow,deny
        Allow from all
    </Directory>
    ErrorLog ${APACHE_LOG_DIR}/error.log
    LogLevel warn
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
~
```

replace this with public DNS name or Public IP address of EC2 instance

- Enable the virtual host with the command below

  ```
  $ sudo a2ensite FlaskApp
  ```

Note: In case we get with the following message

```
--
Enabling site FlaskApp.
To activate the new configuration, you need to run:
  systemctl reload apache2
--
```

- Then run the following command:

  ```
  $ sudo systemctl reload apache2
  ```

- Create the .wsgi File

Run the following command and create a configuration file flaskapp.wsgi in the FlaskApp directory

```
$ sudo vi /var/www/FlaskApp/flaskapp.wsgi
```

```
#!/usr/bin/python
import sys
import logging
logging.basicConfig(stream=sys.stderr)
sys.path.insert(0,"/var/www/FlaskApp/")

from FlaskApp.app import app as application
application.secret_key = 'Add your secret key or a Dummy string'
```

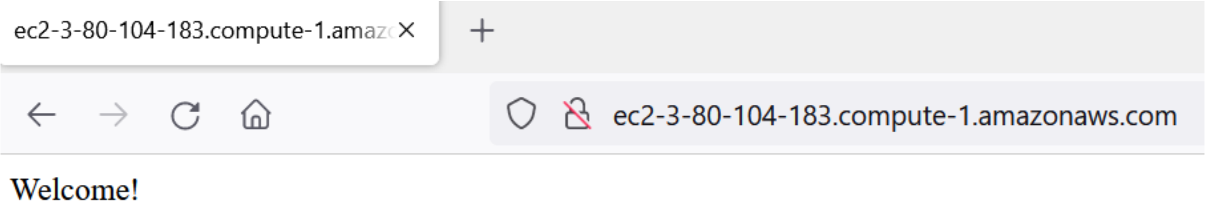
- Finally restart apache2 server.

```
$ sudo service apache2 restart
```

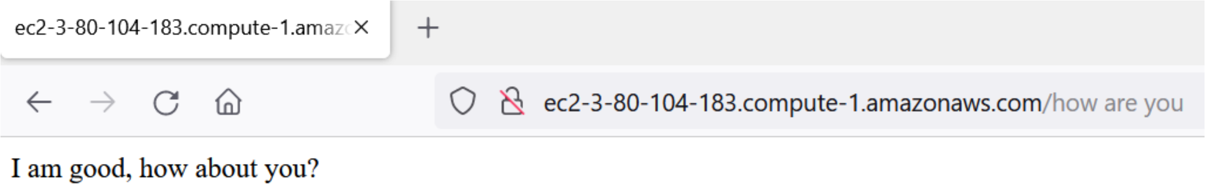
If all well OK till here then our flask application is ready to accept HTTP request from anywhere on port 80 using the Public IP address or Public DNS name of our EC2 instance.

Our EC2 Public IPv4 DNS is : **ec2-3-80-104-183.compute-1.amazonaws.com**

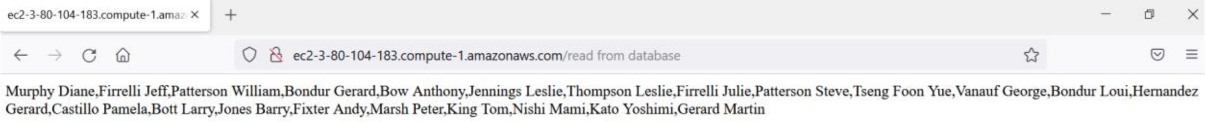
http://ec2-3-80-104-183.compute-1.amazonaws.com/ => Welcome
http://ec2-3-80-104-183.compute-1.amazonaws.com/how%20are%20you => I am good, how about you?
http://ec2-3-80-104-183.compute-1.amazonaws.com/read%20from%20database => <List of employee names>


ec2-3-80-104-183.compute-1.amaz
ec2-3-80-104-183.compute-1.amazonaws.com

Welcome!


ec2-3-80-104-183.compute-1.amaz
ec2-3-80-104-183.compute-1.amazonaws.com/how are you

I am good, how about you?


ec2-3-80-104-183.compute-1.amaz
ec2-3-80-104-183.compute-1.amazonaws.com/read from database

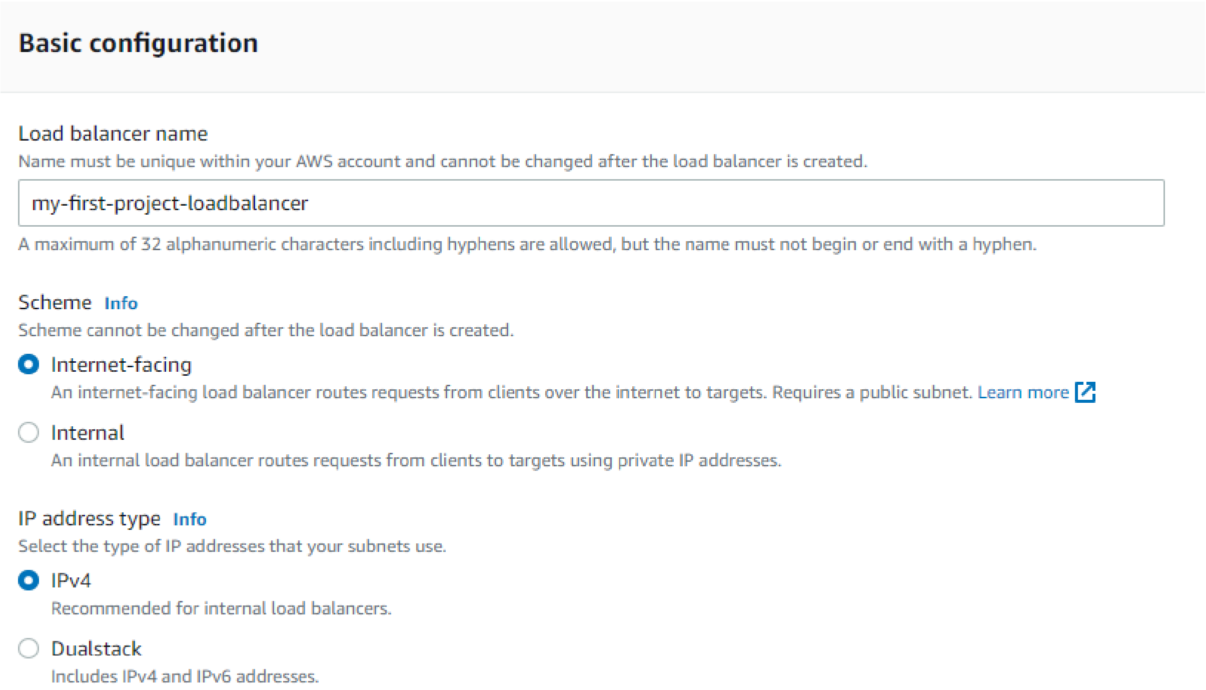
Murphy Diane,Firrelli Jeff,Patterson William,Bondur Gerard,Bow Anthony,Jennings Leslie,Thompson Leslie,Firrelli Julie,Patterson Steve,Tseng Foon Yue,Vanauf George,Bondur Loui,Hernandez Gerard,Castillo Pamela,Bott Larry,Jones Barry,Fixter Andy,Marsh Peter,King Tom,Nishi Mami,Kato Yoshimi,Gerard Martin

This indicates that we have installed Flask and configured a sample test Flask application to use port number 80 properly.

3.6 Creating Load balancer

3.6.1 Create load balancer

- Go to **EC2 > Load balancers > Application Load Balancer**



Basic configuration

Load balancer name

Name must be unique within your AWS account and cannot be changed after the load balancer is created.

my-first-project-loadbalancer

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Scheme Info

Scheme cannot be changed after the load balancer is created.

● Internet-facing
An internet-facing load balancer routes requests from clients over the internet to targets. Requires a public subnet. Learn more

○ Internal
An internal load balancer routes requests from clients to targets using private IP addresses.

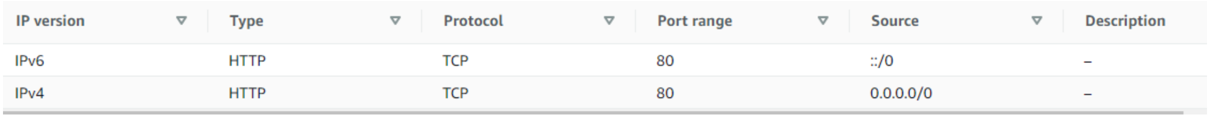
IP address type Info

Select the type of IP addresses that your subnets use.

● IPv4
Recommended for internal load balancers.

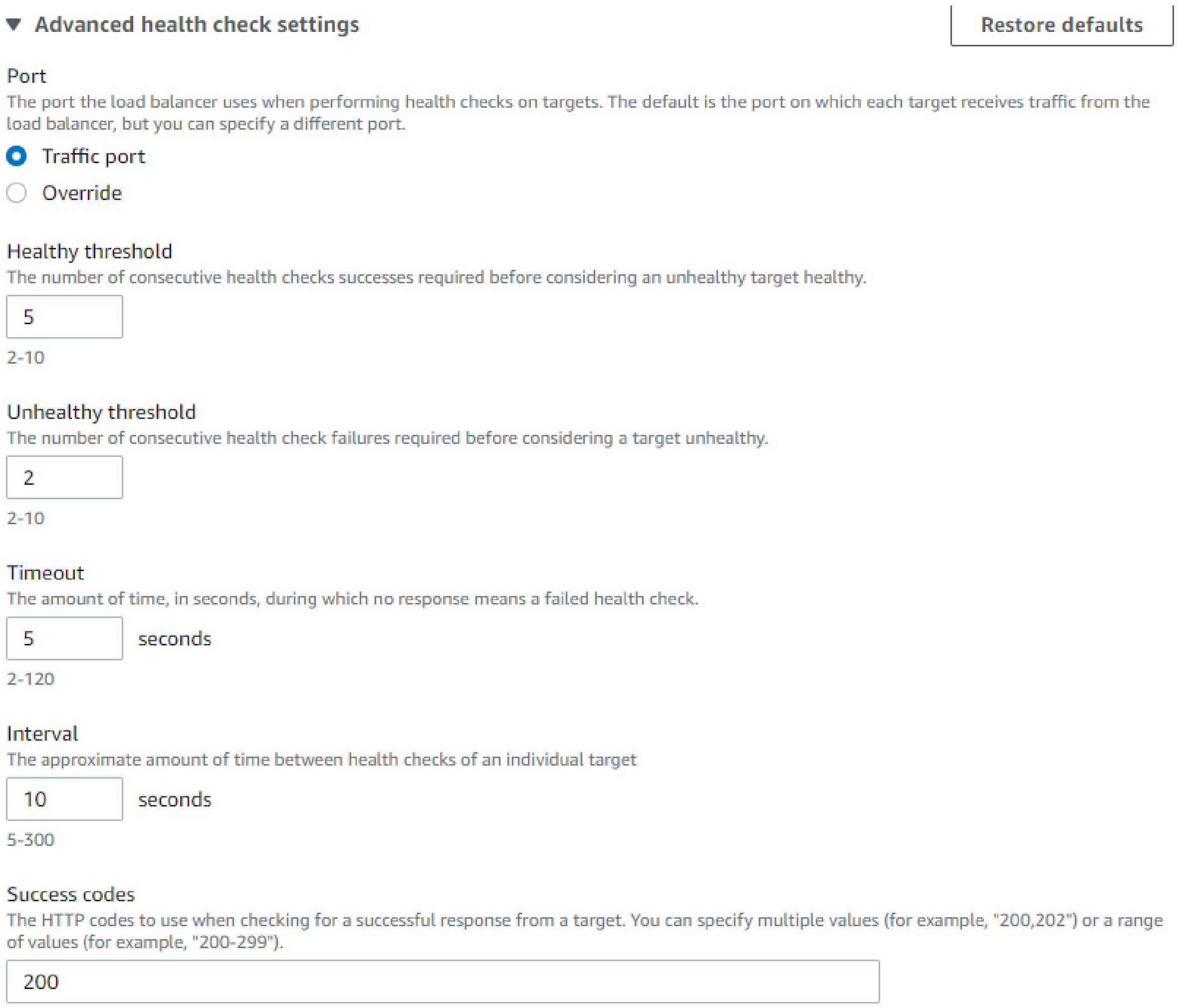
○ Dualstack
Includes IPv4 and IPv6 addresses.

- Create new Security Group: my_first_project_security_group03



IP version	Type	Protocol	Port range	Source	Description
IPv6	HTTP	TCP	80	::/0	–
IPv4	HTTP	TCP	80	0.0.0.0/0	–

- **Listeners and routing**
- Click on Create target Group and create a target group as :
  - Choose a target type: Instance
  - Protocol version: HTTP1
  - Health check protocol
    - HTTP
  - Advance health check settings



▼ **Advanced health check settings**

Port

The port the load balancer uses when performing health checks on targets. The default is the port on which each target receives traffic from the load balancer, but you can specify a different port.

● Traffic port

○ Override

Healthy threshold

The number of consecutive health checks successes required before considering an unhealthy target healthy.

5

2-10

Unhealthy threshold

The number of consecutive health check failures required before considering a target unhealthy.

2

2-10

Timeout

The amount of time, in seconds, during which no response means a failed health check.

5 seconds

2-120

Interval

The approximate amount of time between health checks of an individual target

10 seconds

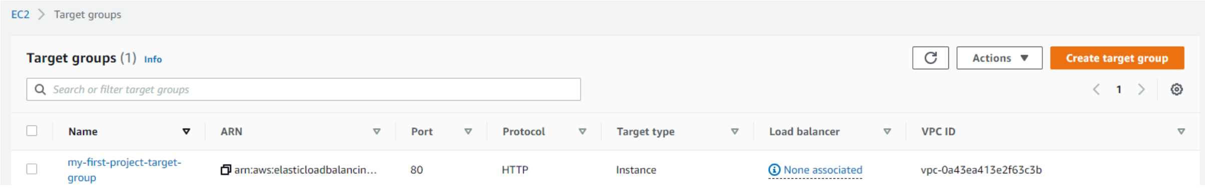
5-300

Success codes

The HTTP codes to use when checking for a successful response from a target. You can specify multiple values (for example, "200,202") or a range of values (for example, "200-299").

200

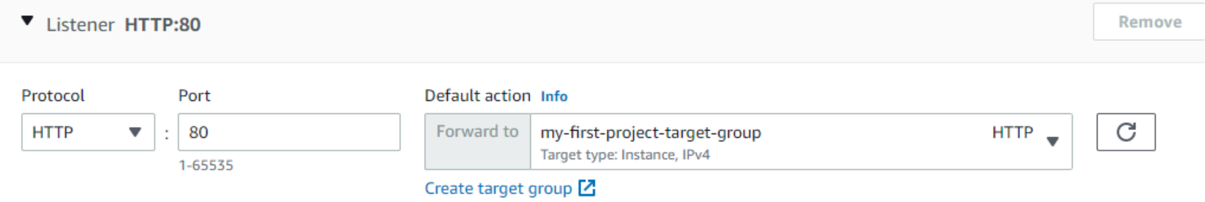
- Click Next
- Click Create target group



Target groups (1) Info

□	Name	ARN	Port	Protocol	Target type	Load balancer	VPC ID
□	my-first-project-target-group	arn:aws:elasticloadbalancin...	80	HTTP	Instance	None associated	vpc-0a43ea413e2f63c3b

- Go back to Load balancer creation screen and select this new target group.



▼ Listener **HTTP:80**

Protocol: HTTP

Port: 80

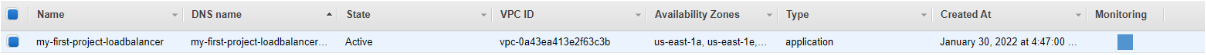
1-65535

Default action Info

Forward to my-first-project-target-group HTTP

Target type: Instance, IPv4

- Click on Create load balancer

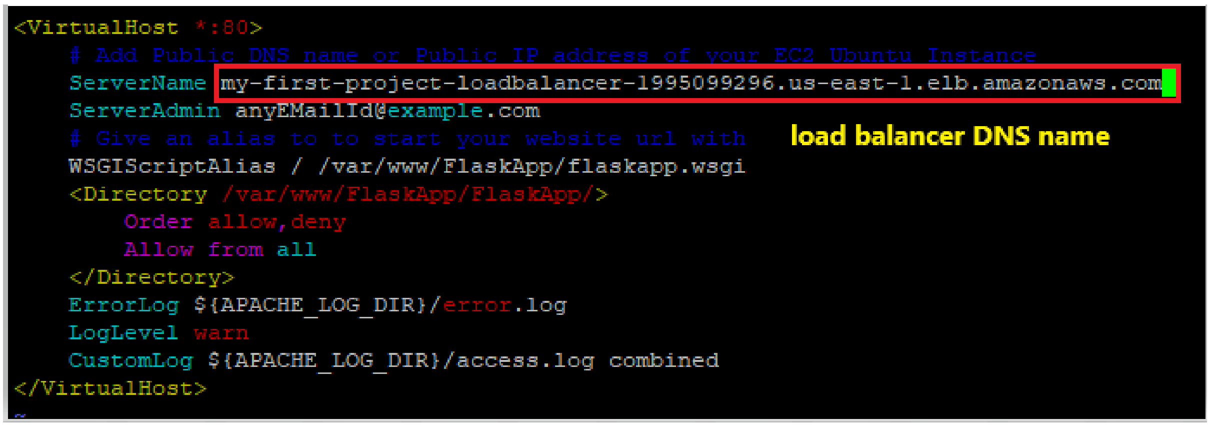


■	Name	DNS name	State	VPC ID	Availability Zones	Type	Created At	Monitoring
■	my-first-project-loadbalancer	my-first-project-loadbalancer...	Active	vpc-0a43ea413e2f63c3b	us-east-1a, us-east-1e,...	application	January 30, 2022 at 4:47:00 ...	■

3.6.2 Update WSGI configuration file

- Update the DNS name with load balancer DNS name

sudo vi /etc/apache2/sites-available/FlaskApp.conf



```
<VirtualHost *:80>
    # Add Public DNS name or Public IP address of your EC2 Ubuntu Instance
    ServerName my-first-project-loadbalancer-1995099296.us-east-1.elb.amazonaws.com
    ServerAdmin anyEMailId@example.com
    # Give an alias to to start your website url with
    WSGIScriptAlias / /var/www/FlaskApp/flaskapp.wsgi
    <Directory /var/www/FlaskApp/FlaskApp/>
        Order allow,deny
        Allow from all
    </Directory>
    ErrorLog ${APACHE_LOG_DIR}/error.log
    LogLevel warn
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
~
```

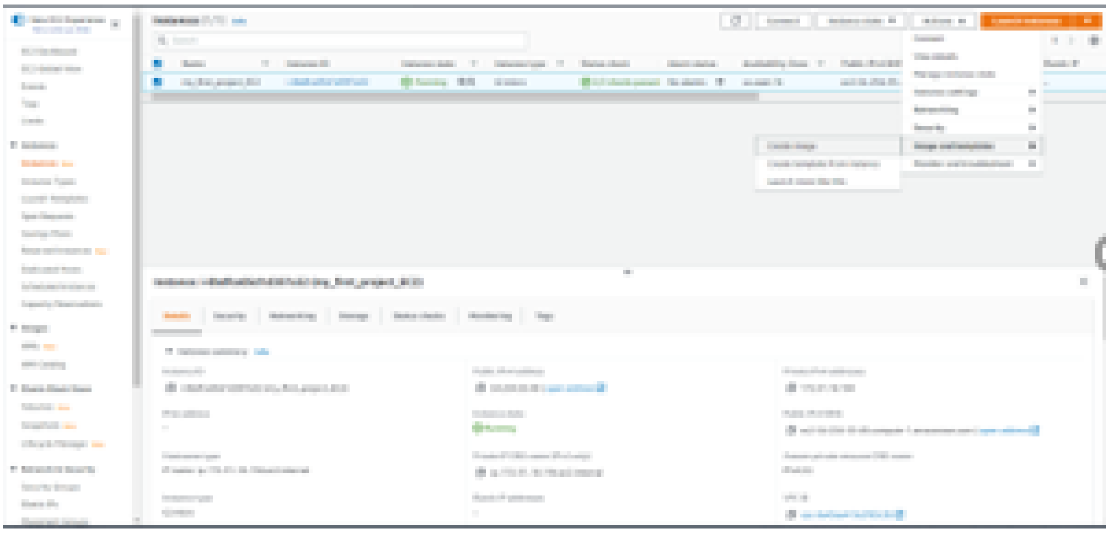
- Enable the virtual host with the command below after configuration change

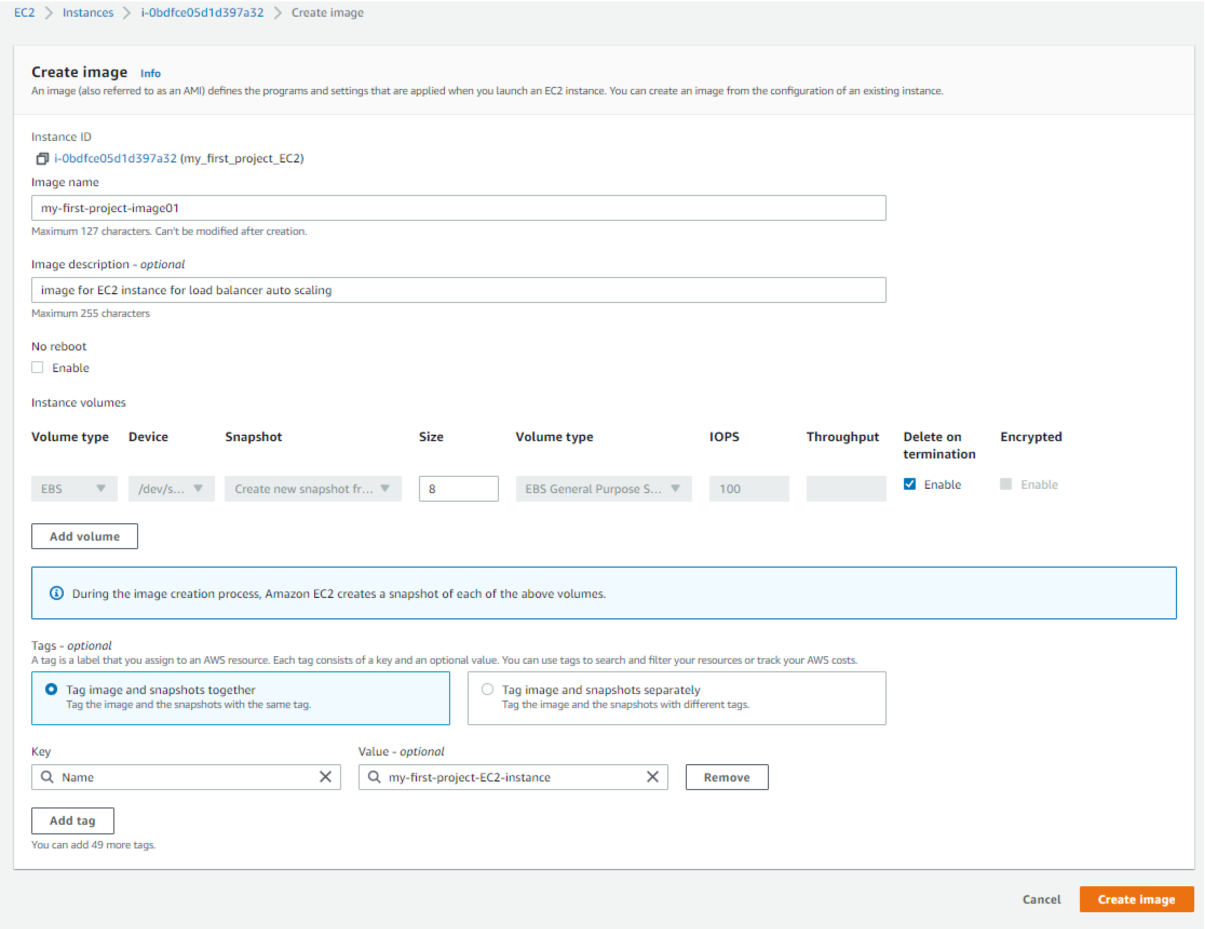
$ sudo a2ensite FlaskApp

3.6.3 Creating auto scaling group

After this get EC2 instance gets launched, Create an AMI (image) with any name for example my-first-project-img-01

Select EC2 instance > Action > Image and templates > Create image





EC2 > Instances > i-0bdfce05d1d397a32 > Create image

Create image Info

An image (also referred to as an AMI) defines the programs and settings that are applied when you launch an EC2 instance. You can create an image from the configuration of an existing instance.

Instance ID

i-0bdfce05d1d397a32 (my_first_project_EC2)

Image name

my-first-project-image01

Maximum 127 characters. Can't be modified after creation.

Image description - *optional*

image for EC2 instance for load balancer auto scaling

Maximum 255 characters

No reboot

□ Enable

Instance volumes

Volume type	Device	Snapshot	Size	Volume type	IOPS	Throughput	Delete on termination	Encrypted
EBS	/dev/s...	Create new snapshot fr...	8	EBS General Purpose S...	100		☑ Enable	□ Enable

Add volume

ⓘ During the image creation process, Amazon EC2 creates a snapshot of each of the above volumes.

Tags - *optional*

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

● Tag image and snapshots together
Tag the image and the snapshots with the same tag.

○ Tag image and snapshots separately
Tag the image and the snapshots with different tags.

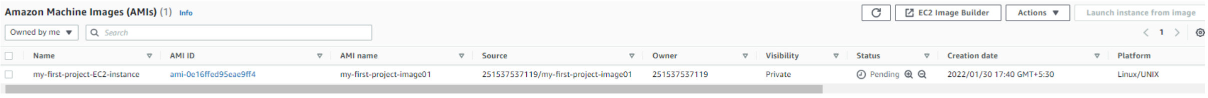
Key	Value - *optional*	
Name	my-first-project-EC2-instance	Remove

Add tag

You can add 49 more tags.

Cancel Create image

New Image is created for EC2 instance. (Go to Images > AMI)



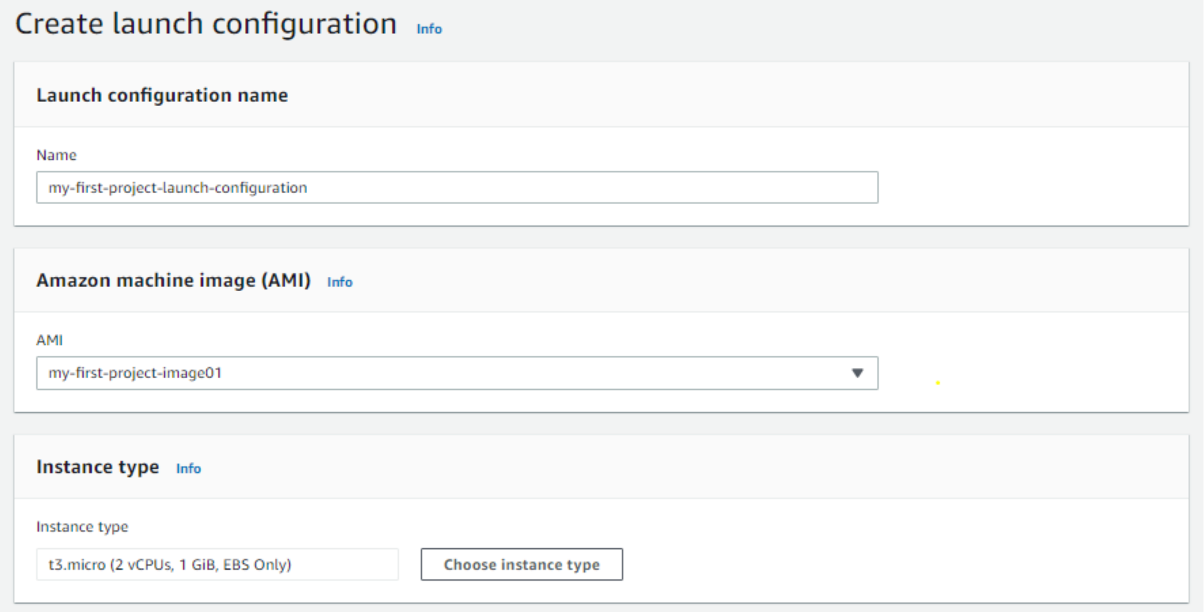
Amazon Machine Images (AMIs) (1) Info

Name	AMI ID	AMI name	Source	Owner	Visibility	Status	Creation date	Platform
my-first-project-EC2-instance	ami-0e16ffed95eae9ff4	my-first-project-image01	251537537119/my-first-project-image01	251537537119	Private	Pending	2022/01/30 17:40 GMT+5:30	Linux/UNIX

3.6.4 Creating launch configuration

If something goes down in running EC2 instance, then Auto scaling group must start a new instance. For this we need to create a **launch configuration.**

Go to **EC2 > Launch configurations > Create launch configuration**



Create launch configuration Info

Launch configuration name

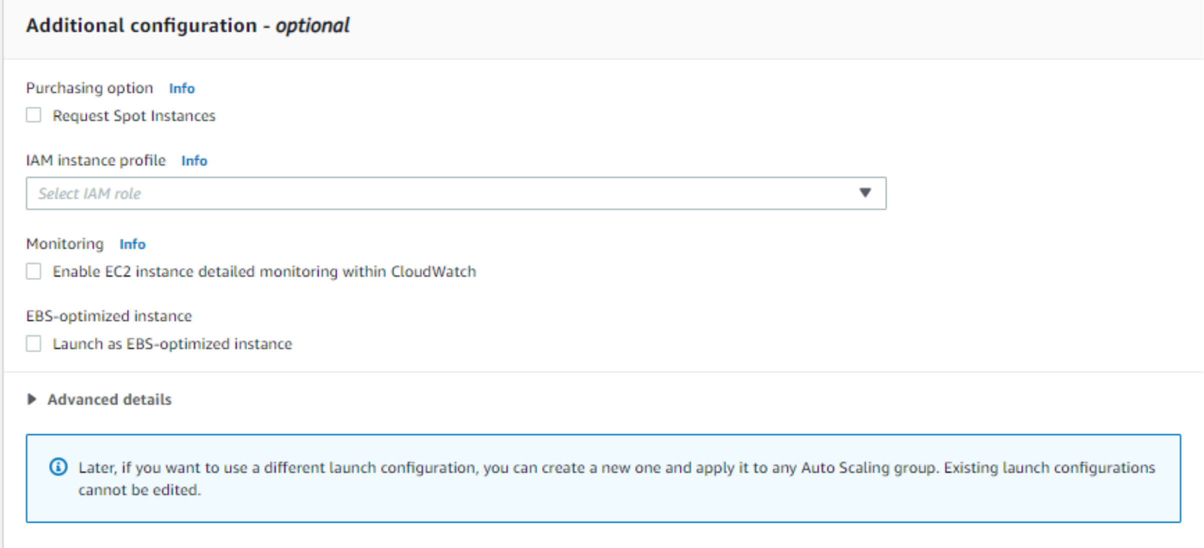
Name: my-first-project-launch-configuration

Amazon machine image (AMI) Info

AMI: my-first-project-image01 ▼

Instance type Info

Instance type: t3.micro (2 vCPUs, 1 GiB, EBS Only)



Additional configuration - *optional*

Purchasing option Info

□ Request Spot Instances

IAM instance profile Info

Select IAM role ▼

Monitoring Info

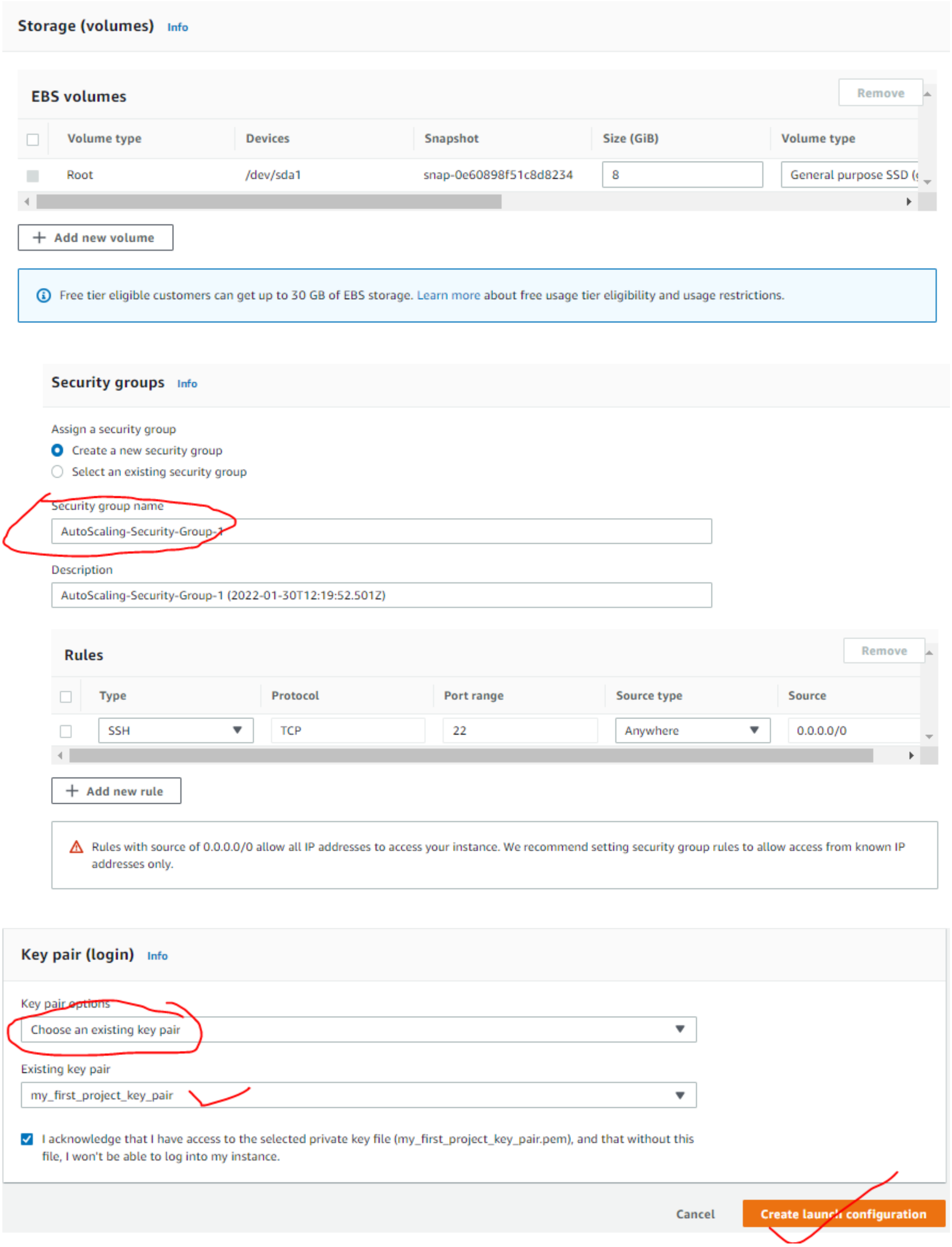
□ Enable EC2 instance detailed monitoring within CloudWatch

EBS-optimized instance

□ Launch as EBS-optimized instance

▸ Advanced details

ⓘ Later, if you want to use a different launch configuration, you can create a new one and apply it to any Auto Scaling group. Existing launch configurations cannot be edited.



Storage (volumes) Info

EBS volumes

☐	Volume type	Devices	Snapshot	Size (GiB)	Volume type
☐	Root	/dev/sda1	snap-0e60898f51c8d8234	8	General purpose SSD ((

Free tier eligible customers can get up to 30 GB of EBS storage. Learn more about free usage tier eligibility and usage restrictions.

Security groups Info

Assign a security group

● Create a new security group

○ Select an existing security group

Security group name: AutoScaling-Security-Group-1

Description: AutoScaling-Security-Group-1 (2022-01-30T12:19:52.501Z)

Rules

☐	Type	Protocol	Port range	Source type	Source
☐	SSH	TCP	22	Anywhere	0.0.0.0/0

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.

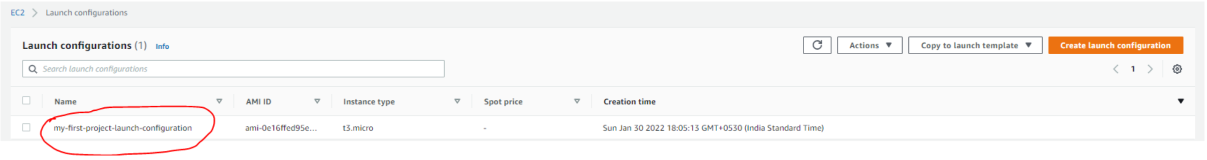
Key pair (login) Info

Key pair options: Choose an existing key pair

Existing key pair: my_first_project_key_pair

☑ I acknowledge that I have access to the selected private key file (my_first_project_key_pair.pem), and that without this file, I won't be able to log into my instance.

Remove
+ Add new volume
Remove
+ Add new rule
Cancel
Create launch configuration



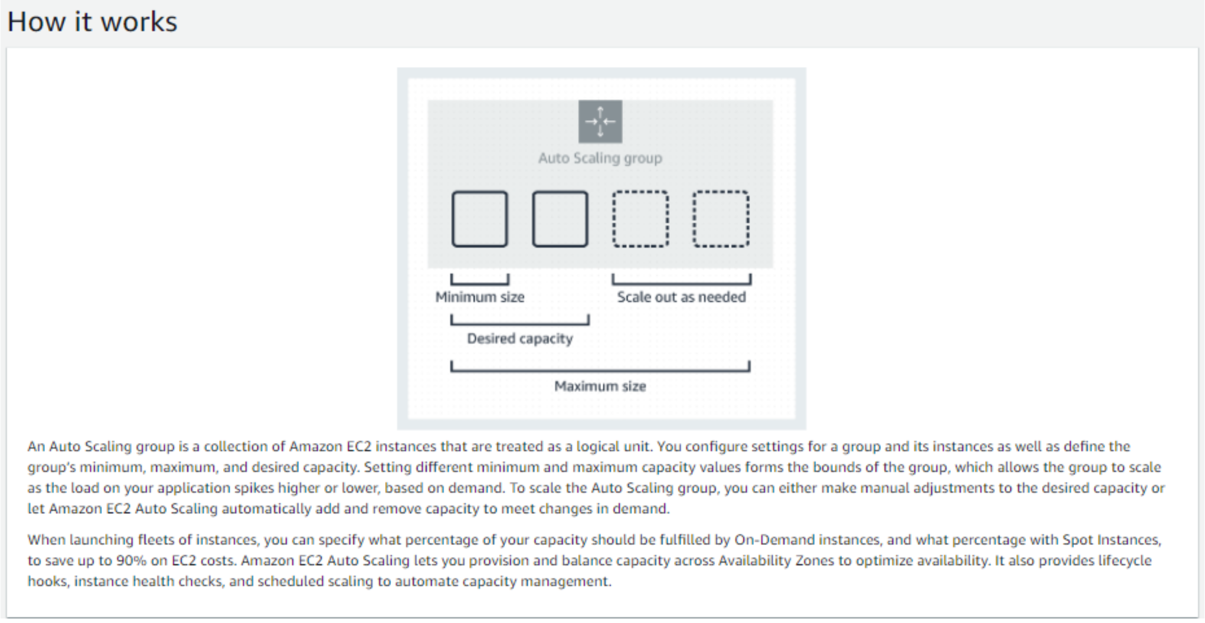
EC2 > Launch configurations

Launch configurations (1) Info

☐	Name	AMI ID	Instance type	Spot price	Creation time
☐	my-first-project-launch-configuration	ami-0e16ffed95e...	t3.micro	-	Sun Jan 30 2022 18:05:13 GMT+0530 (India Standard Time)

Actions
Copy to launch template
Create launch configuration
Search launch configurations
1

3.6.5 Auto Scaling group

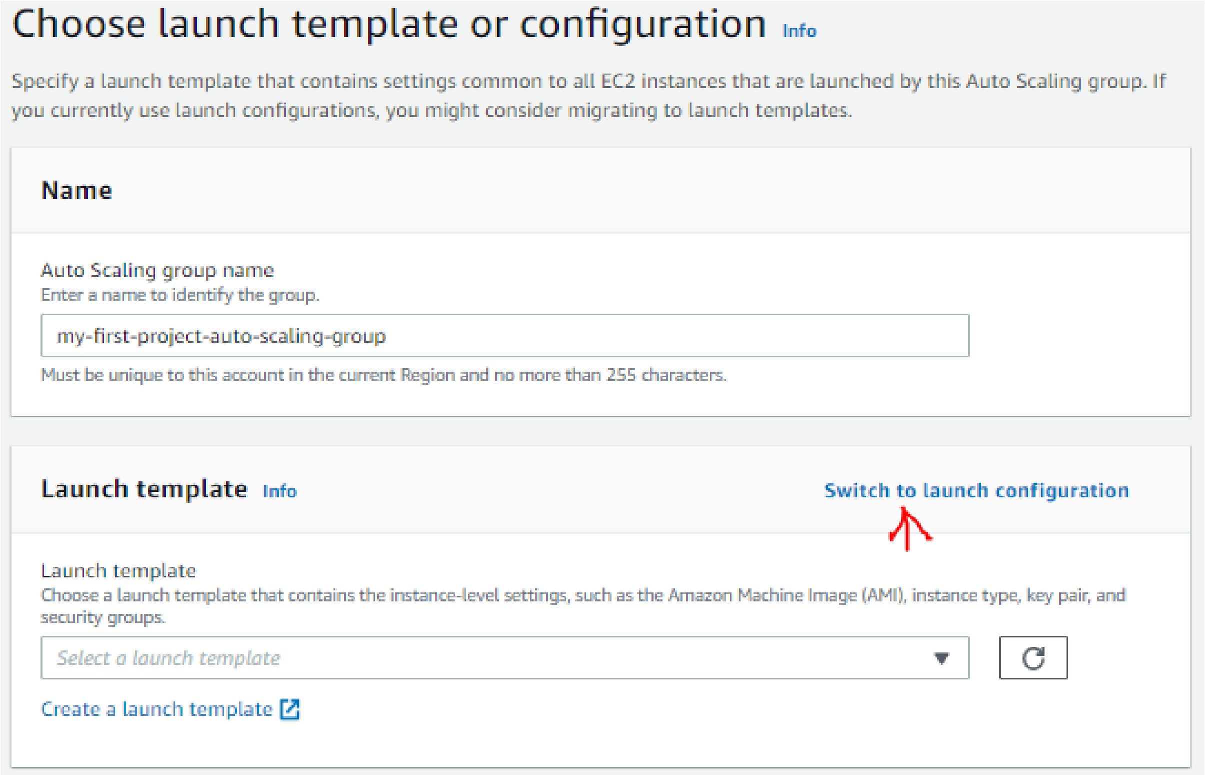


How it works

An Auto Scaling group is a collection of Amazon EC2 instances that are treated as a logical unit. You configure settings for a group and its instances as well as define the group's minimum, maximum, and desired capacity. Setting different minimum and maximum capacity values forms the bounds of the group, which allows the group to scale as the load on your application spikes higher or lower, based on demand. To scale the Auto Scaling group, you can either make manual adjustments to the desired capacity or let Amazon EC2 Auto Scaling automatically add and remove capacity to meet changes in demand.

When launching fleets of instances, you can specify what percentage of your capacity should be fulfilled by On-Demand instances, and what percentage with Spot Instances, to save up to 90% on EC2 costs. Amazon EC2 Auto Scaling lets you provision and balance capacity across Availability Zones to optimize availability. It also provides lifecycle hooks, instance health checks, and scheduled scaling to automate capacity management.

Go to **EC2 > Auto Scaling groups > Create Auto Scaling group**



Choose launch template or configuration Info

Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group. If you currently use launch configurations, you might consider migrating to launch templates.

Name

Auto Scaling group name
Enter a name to identify the group.

my-first-project-auto-scaling-group

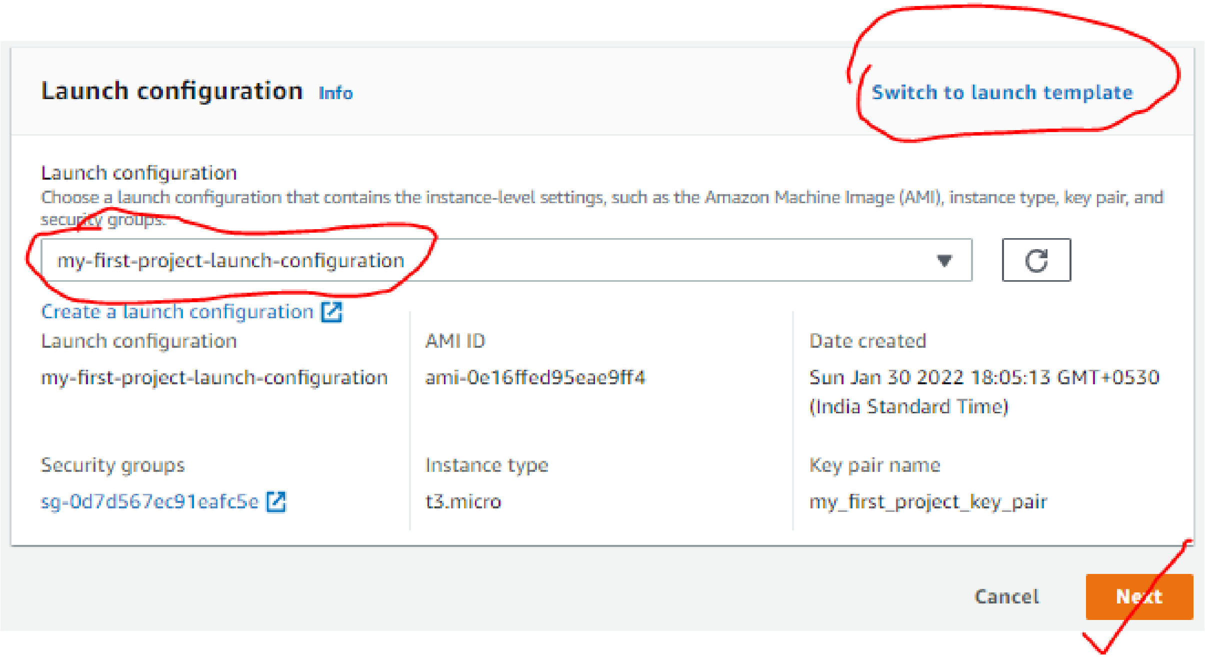
Must be unique to this account in the current Region and no more than 255 characters.

Launch template Info

Launch template
Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.

Select a launch template

Choose Switch to launch configuration and choose the launch configuration that we created. Click on Next.



Launch configuration

Choose a launch configuration that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.

my-first-project-launch-configuration

Launch configuration: my-first-project-launch-configuration

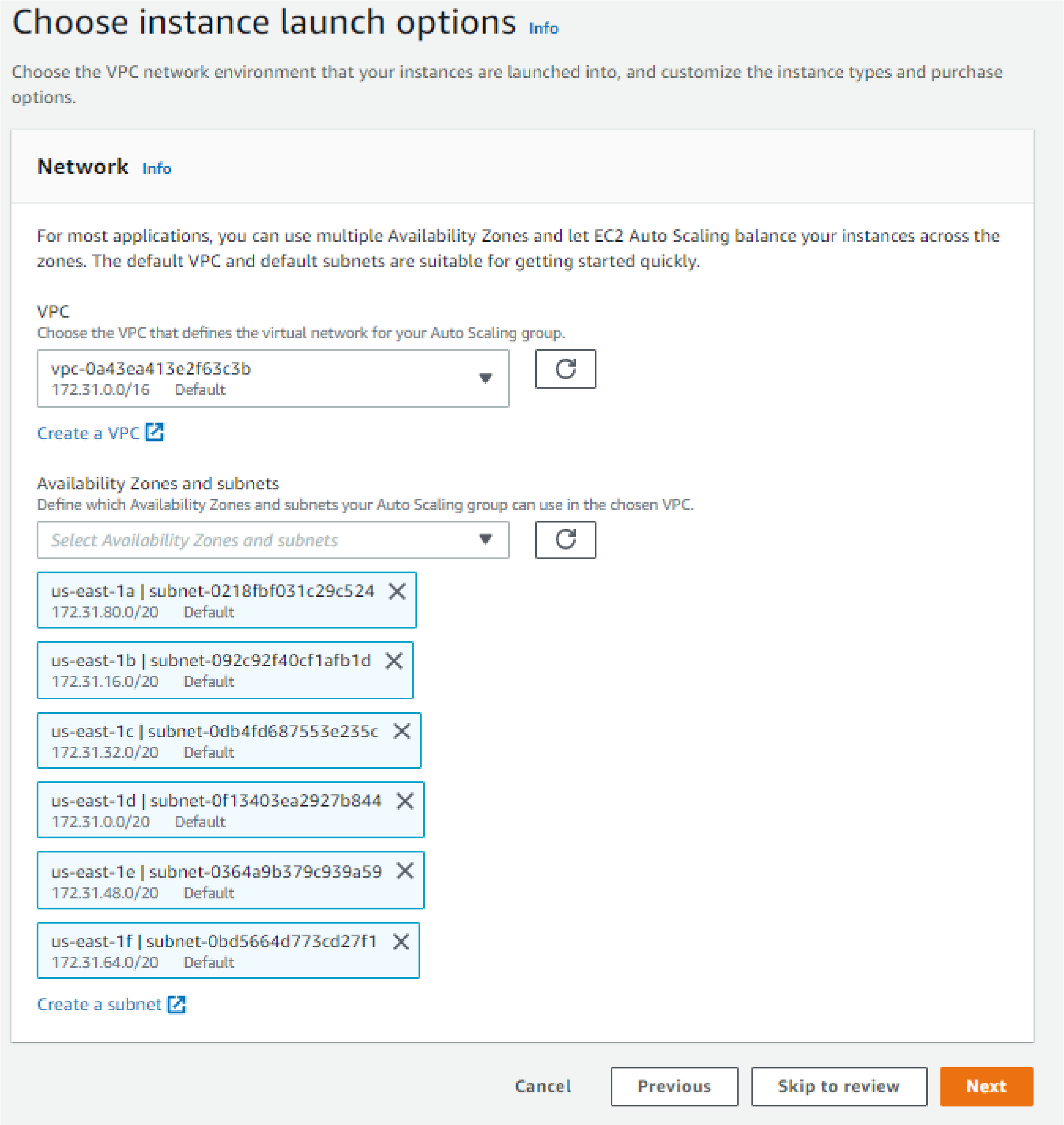
AMI ID: ami-0e16ffed95eae9ff4

Date created: Sun Jan 30 2022 18:05:13 GMT+0530 (India Standard Time)

Security groups: sg-0d7d567ec91eafc5e

Instance type: t3.micro

Key pair name: my_first_project_key_pair



Choose instance launch options Info

Choose the VPC network environment that your instances are launched into, and customize the instance types and purchase options.

Network Info

For most applications, you can use multiple Availability Zones and let EC2 Auto Scaling balance your instances across the zones. The default VPC and default subnets are suitable for getting started quickly.

VPC

Choose the VPC that defines the virtual network for your Auto Scaling group.

vpc-0a43ea413e2f63c3b
172.31.0.0/16 Default

Create a VPC

Availability Zones and subnets

Define which Availability Zones and subnets your Auto Scaling group can use in the chosen VPC.

Select Availability Zones and subnets

us-east-1a | subnet-0218fbf031c29c524 ✕
172.31.80.0/20 Default

us-east-1b | subnet-092c92f40cf1afb1d ✕
172.31.16.0/20 Default

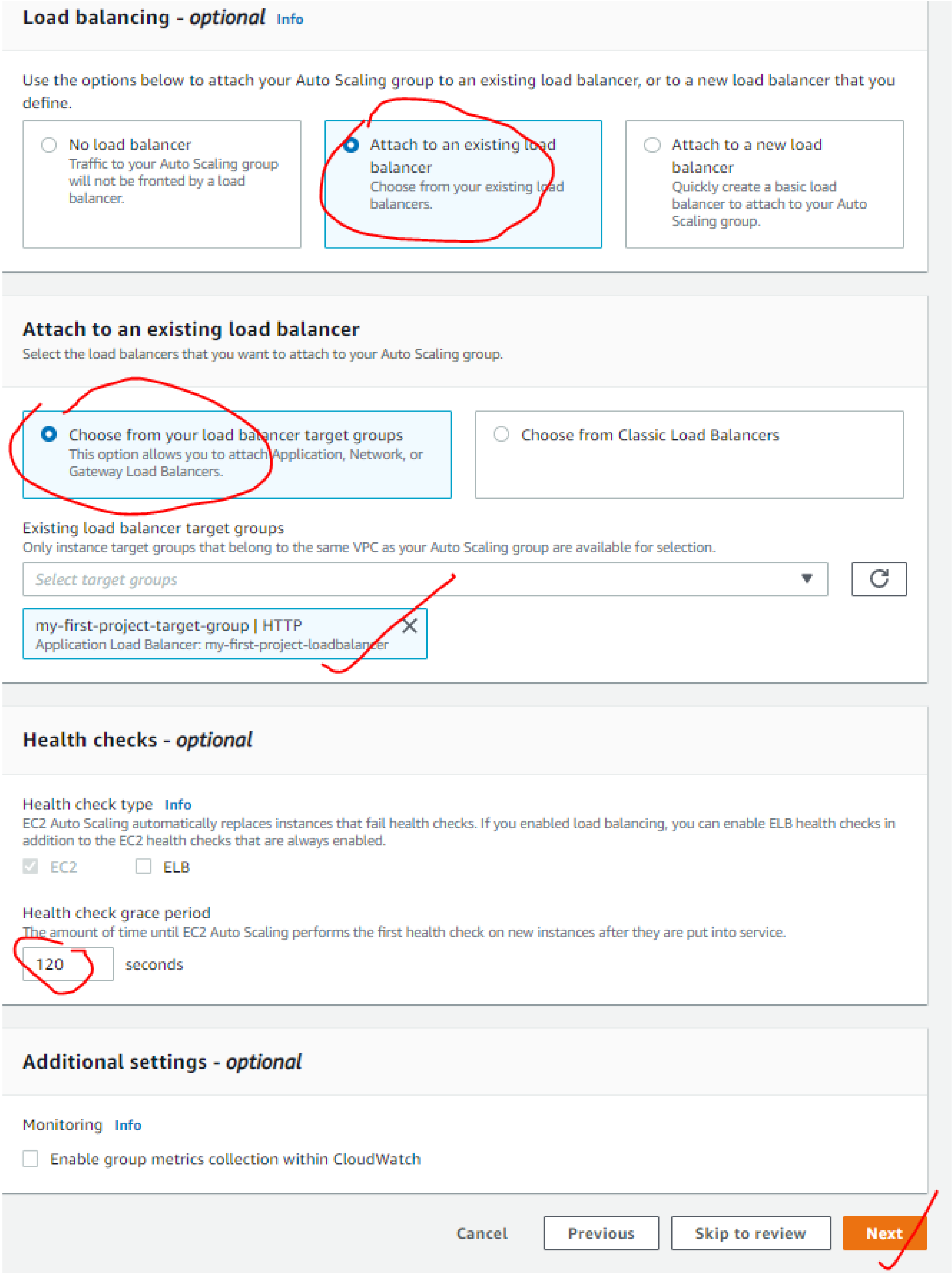
us-east-1c | subnet-0db4fd687553e235c ✕
172.31.32.0/20 Default

us-east-1d | subnet-0f13403ea2927b844 ✕
172.31.0.0/20 Default

us-east-1e | subnet-0364a9b379c939a59 ✕
172.31.48.0/20 Default

us-east-1f | subnet-0bd5664d773cd27f1 ✕
172.31.64.0/20 Default

Create a subnet



Load balancing - *optional* Info

Use the options below to attach your Auto Scaling group to an existing load balancer, or to a new load balancer that you define.

○ No load balancer
Traffic to your Auto Scaling group will not be fronted by a load balancer.

● Attach to an existing load balancer
Choose from your existing load balancers.

○ Attach to a new load balancer
Quickly create a basic load balancer to attach to your Auto Scaling group.

Attach to an existing load balancer

Select the load balancers that you want to attach to your Auto Scaling group.

● Choose from your load balancer target groups
This option allows you to attach Application, Network, or Gateway Load Balancers.

○ Choose from Classic Load Balancers

Existing load balancer target groups
Only instance target groups that belong to the same VPC as your Auto Scaling group are available for selection.

Select target groups ▼

my-first-project-target-group | HTTP ✕
Application Load Balancer: my-first-project-loadbalancer

Health checks - *optional*

Health check type Info
EC2 Auto Scaling automatically replaces instances that fail health checks. If you enabled load balancing, you can enable ELB health checks in addition to the EC2 health checks that are always enabled.

☑ EC2 □ ELB

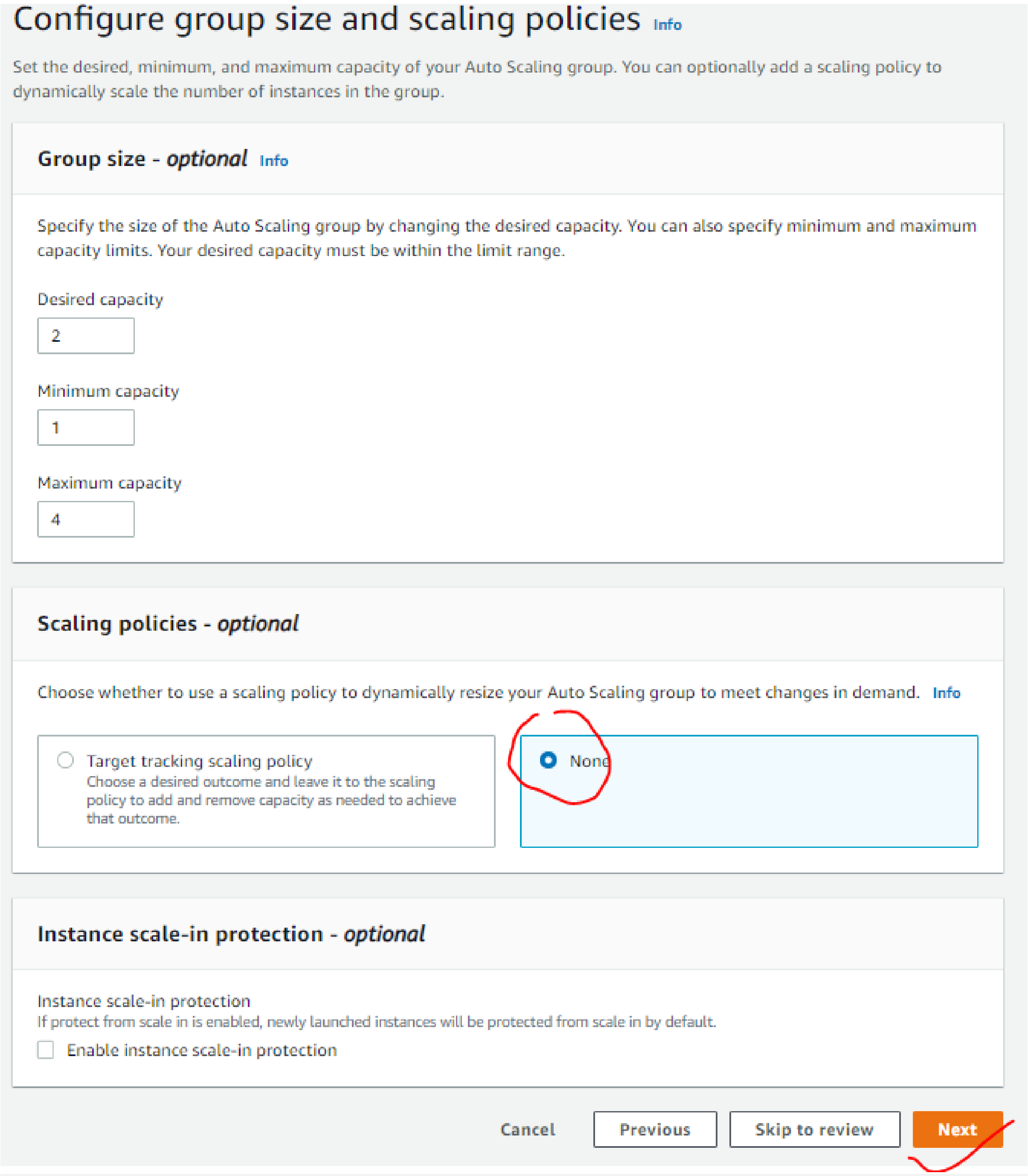
Health check grace period
The amount of time until EC2 Auto Scaling performs the first health check on new instances after they are put into service.

120 seconds

Additional settings - *optional*

Monitoring Info

□ Enable group metrics collection within CloudWatch



Configure group size and scaling policies

Set the desired, minimum, and maximum capacity of your Auto Scaling group. You can optionally add a scaling policy to dynamically scale the number of instances in the group.

Group size - *optional*

Specify the size of the Auto Scaling group by changing the desired capacity. You can also specify minimum and maximum capacity limits. Your desired capacity must be within the limit range.

Desired capacity
2

Minimum capacity
1

Maximum capacity
4

Scaling policies - *optional*

Choose whether to use a scaling policy to dynamically resize your Auto Scaling group to meet changes in demand.

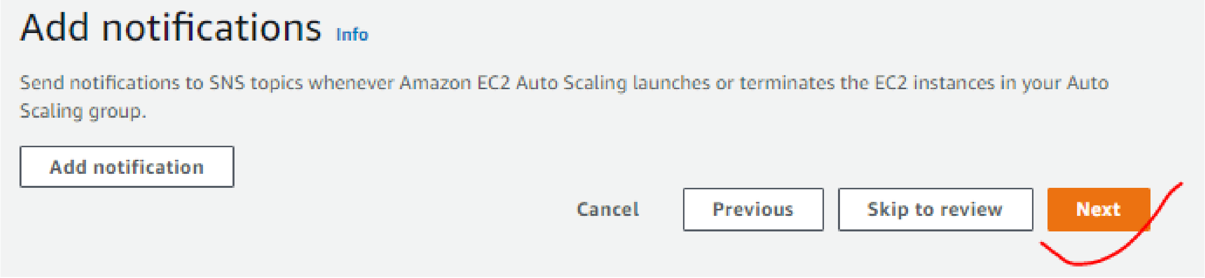
○ Target tracking scaling policy
Choose a desired outcome and leave it to the scaling policy to add and remove capacity as needed to achieve that outcome.

● None

Instance scale-in protection - *optional*

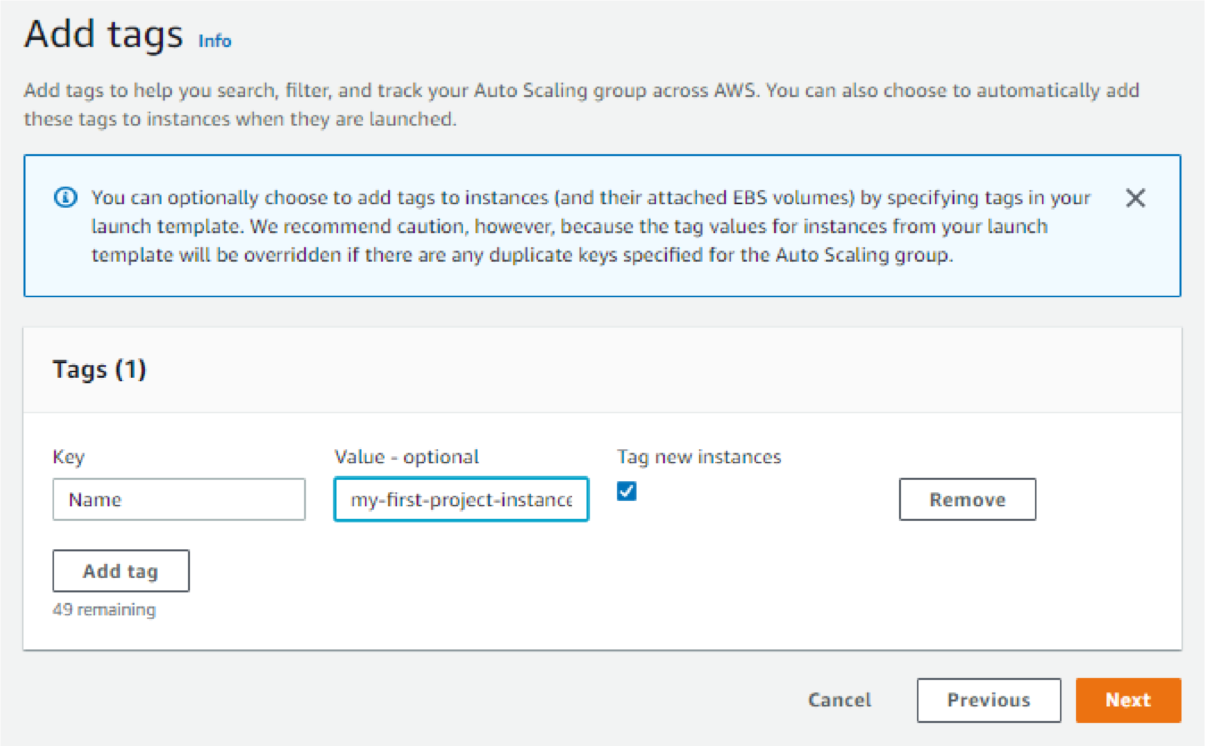
Instance scale-in protection
If protect from scale in is enabled, newly launched instances will be protected from scale in by default.

□ Enable instance scale-in protection



Add notifications

Send notifications to SNS topics whenever Amazon EC2 Auto Scaling launches or terminates the EC2 instances in your Auto Scaling group.



Add tags Info

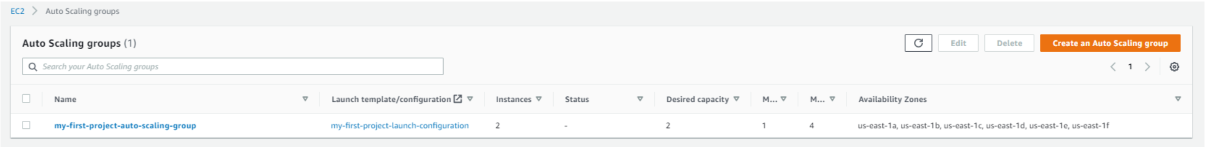
Add tags to help you search, filter, and track your Auto Scaling group across AWS. You can also choose to automatically add these tags to instances when they are launched.

You can optionally choose to add tags to instances (and their attached EBS volumes) by specifying tags in your launch template. We recommend caution, however, because the tag values for instances from your launch template will be overridden if there are any duplicate keys specified for the Auto Scaling group.

Tags (1)

Key	Value - optional	Tag new instances	
Name	my-first-project-instance	☑	Remove

After this Review & Click on Create Auto Scaling group.

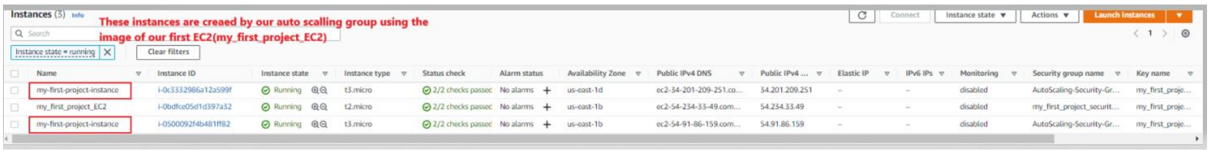


Auto Scaling groups (1)

Search your Auto Scaling groups

Name	Launch template/configuration	Instances	Status	Desired capacity	M...	M...	Availability Zones
my-first-project-auto-scaling-group	my-first-project-launch-configuration	2	-	2	1	4	us-east-1a, us-east-1b, us-east-1c, us-east-1d, us-east-1e, us-east-1f

Now go to EC2 instances and we can find that auto scalling group has created two new instances using our image: my-first-project-img-01



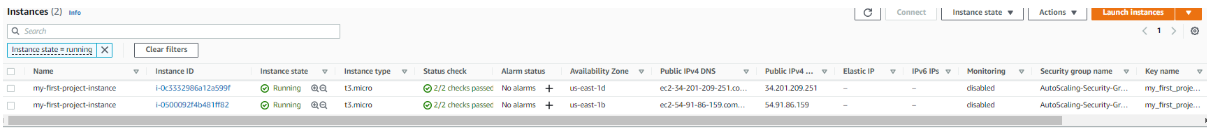
Instances (3) Info

These instances are creaed by our auto scalling group using the image of our first EC2(my_first_project_EC2)

Instance state = running ✕ | Clear filters

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP	IPv6 IPs	Monitoring	Security group name	Key name
my-first-project-instance	i-0c3332986a12a599f	Running	t3.micro	2/2 checks passec	No alarms	us-east-1d	ec2-34-201-209-251.co...	34.201.209.251	–	–	disabled	AutoScaling-Security-Gr...	my_first_proje...
my_first_project_EC2	i-0bdfce05d1d397a32	Running	t2.micro	2/2 checks passec	No alarms	us-east-1b	ec2-54-234-33-49.com...	54.234.33.49	–	–	disabled	my_first_project_securit...	my_first_proje...
my-first-project-instance	i-0500092f4b481ff82	Running	t3.micro	2/2 checks passec	No alarms	us-east-1b	ec2-54-91-86-159.com...	54.91.86.159	–	–	disabled	AutoScaling-Security-Gr...	my_first_proje...

Now we shall terminate our original instance (my_first_project_EC2) because we need only 2 EC under our auto scaling group.



Instances (2) Info

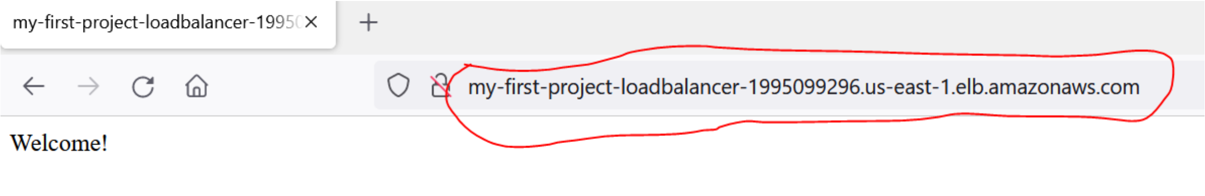
Instance state = running ✕ | Clear filters

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP	IPv6 IPs	Monitoring	Security group name	Key name
my-first-project-instance	i-0c3332986a12a599f	Running	t3.micro	2/2 checks passed	No alarms	us-east-1d	ec2-34-201-209-251.co...	34.201.209.251	–	–	disabled	AutoScaling-Security-Gr...	my_first_proje...
my-first-project-instance	i-0500092f4b481ff82	Running	t3.micro	2/2 checks passed	No alarms	us-east-1b	ec2-54-91-86-159.com...	54.91.86.159	–	–	disabled	AutoScaling-Security-Gr...	my_first_proje...

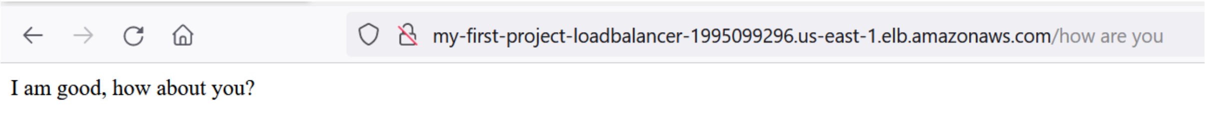
3.6.6 Running application from load balancer

Go to the Load Balancing console by choosing Load Balancer on left navigation panel. Find the DNS name for the Load Balancer created in the earlier Step.

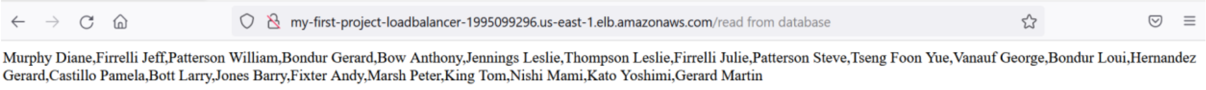
Click on the copy icon next to the A record and then paste it into a new web browser tab.
Open a browser and enter the URL
http://my-first-project-loadbalancer-1995099296.us-east-1.elb.amazonaws.com/
http://my-first-project-loadbalancer-1995099296.us-east-1.elb.amazonaws.com/how%20are%20you
http://my-first-project-loadbalancer-1995099296.us-east-1.elb.amazonaws.com/read%20from%20database


my-first-project-loadbalancer-1995
my-first-project-loadbalancer-1995099296.us-east-1.elb.amazonaws.com

Welcome!


my-first-project-loadbalancer-1995099296.us-east-1.elb.amazonaws.com/how are you

I am good, how about you?


my-first-project-loadbalancer-1995099296.us-east-1.elb.amazonaws.com/read from database

Murphy Diane,Firrelli Jeff,Patterson William,Bondur Gerard,Bow Anthony,Jennings Leslie,Thompson Leslie,Firrelli Julie,Patterson Steve,Tseng Foon Yue,Vanauf George,Bondur Loui,Hernandez Gerard,Castillo Pamela,Bott Larry,Jones Barry,Fixter Andy,Marsh Peter,King Tom,Nishi Mami,Kato Yoshimi,Gerard Martin

End of this Handbook