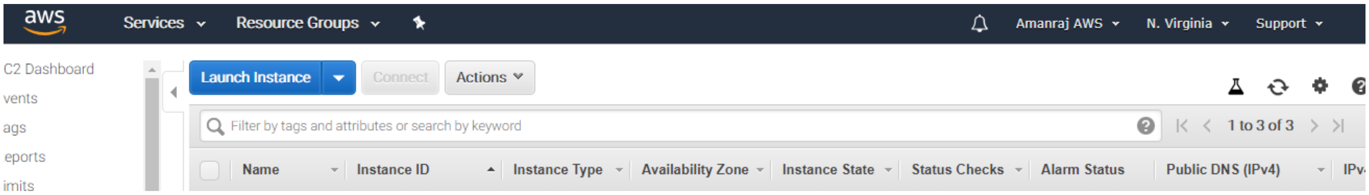
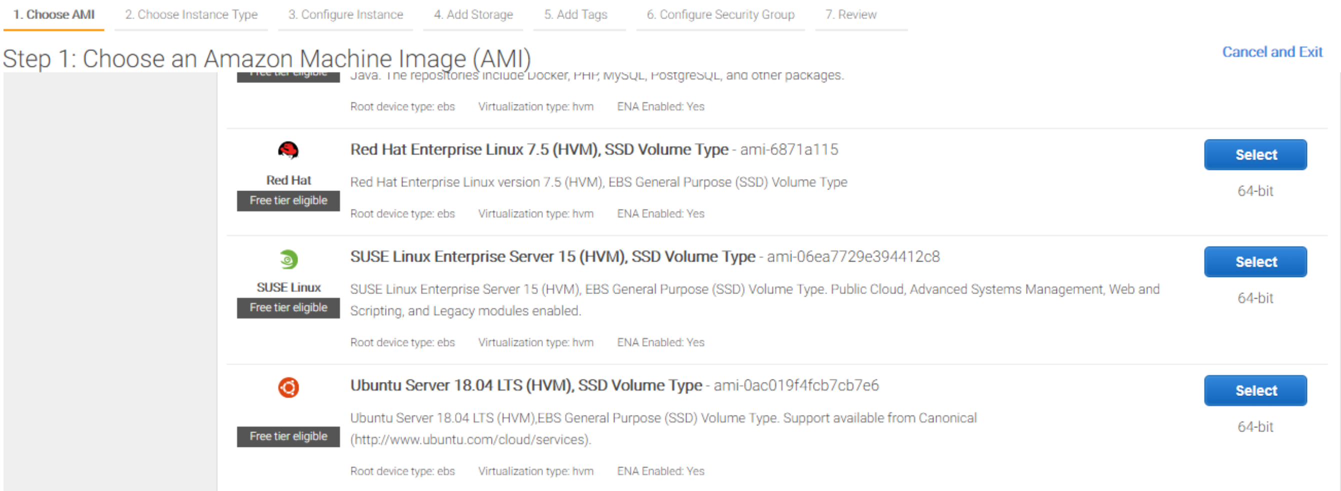
Launch EC2 Instance

Create an account on AWS using gwu.edu email and get 100$ credit free

1. On the EC2 dashboard click on launch instance



2. Select Ubuntu Server 18.04



Step 1: Choose an Amazon Machine Image (AMI)

Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages.

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

Red Hat Enterprise Linux 7.5 (HVM), SSD Volume Type - ami-6871a115

Red Hat Enterprise Linux version 7.5 (HVM), EBS General Purpose (SSD) Volume Type

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

SUSE Linux Enterprise Server 15 (HVM), SSD Volume Type - ami-06ea7729e394412c8

SUSE Linux Enterprise Server 15 (HVM), EBS General Purpose (SSD) Volume Type. Public Cloud, Advanced Systems Management, Web and Scripting, and Legacy modules enabled.

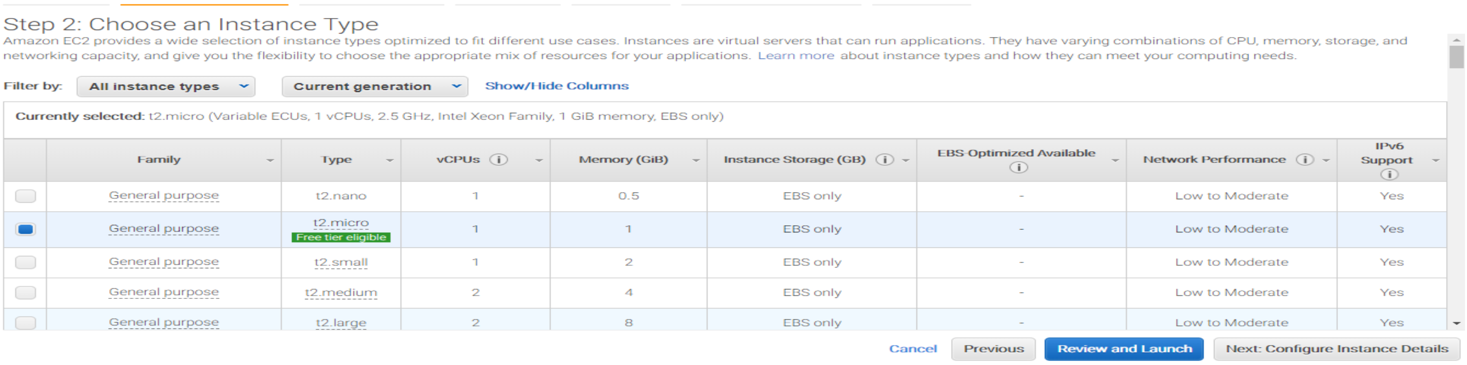
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

Ubuntu Server 18.04 LTS (HVM), SSD Volume Type - ami-0ac019f4fcb7cb7e6

Ubuntu Server 18.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

3. Select T2-micro



Step 2: Choose an Instance Type

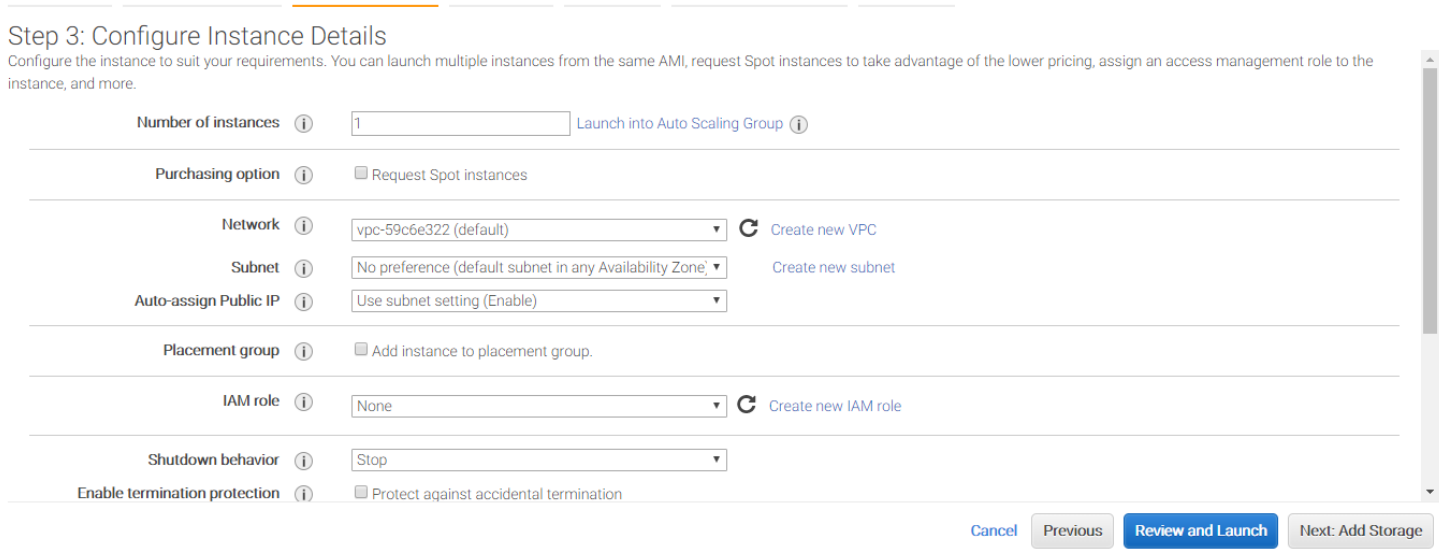
Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. Learn more about instance types and how they can meet your computing needs.

Filter by: All instance types · Current generation · Show/Hide Columns

Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS Optimized Available	Network Performance	IPv6 Support
□	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
■	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes
□	General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Yes
□	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
□	General purpose	t2.large	2	8	EBS only	-	Low to Moderate	Yes

4. Leave this page as it is and hit next to Add storage



Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances: 1 — Launch into Auto Scaling Group

Purchasing option: □ Request Spot instances

Network: vpc-59c6e322 (default) — Create new VPC

Subnet: No preference (default subnet in any Availability Zone) — Create new subnet

Auto-assign Public IP: Use subnet setting (Enable)

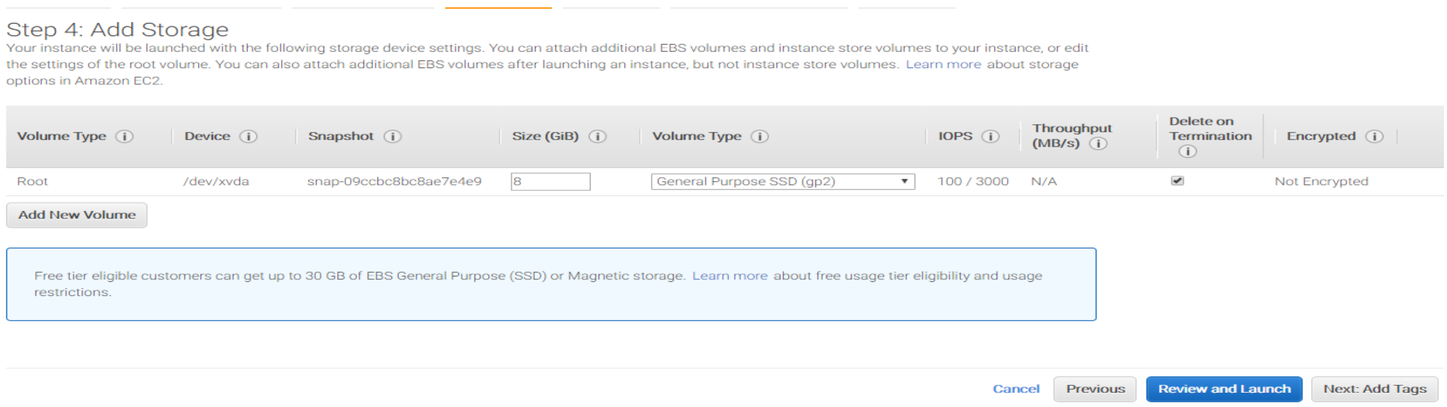
Placement group: □ Add instance to placement group.

IAM role: None — Create new IAM role

Shutdown behavior: Stop

Enable termination protection: □ Protect against accidental termination

5. Hit Next



Step 4: Add Storage

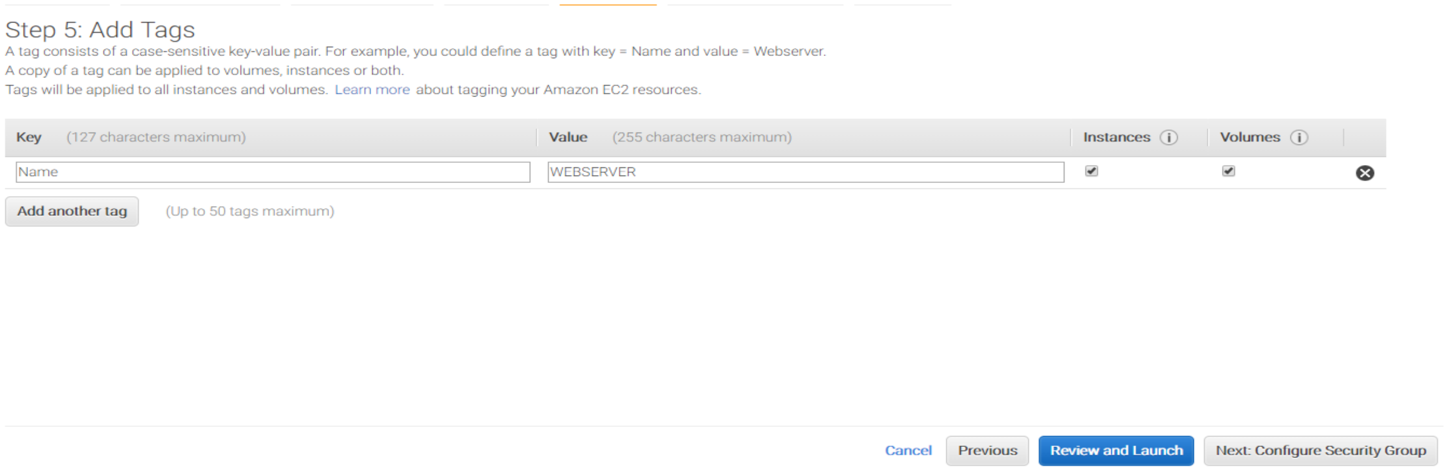
Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. Learn more about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encrypted
Root	/dev/xvda	snap-09ccbc8bc8ae7e4e9	8	General Purpose SSD (gp2)	100 / 3000	N/A	☑	Not Encrypted

Add New Volume

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. Learn more about free usage tier eligibility and usage restrictions.

6. Add Tag



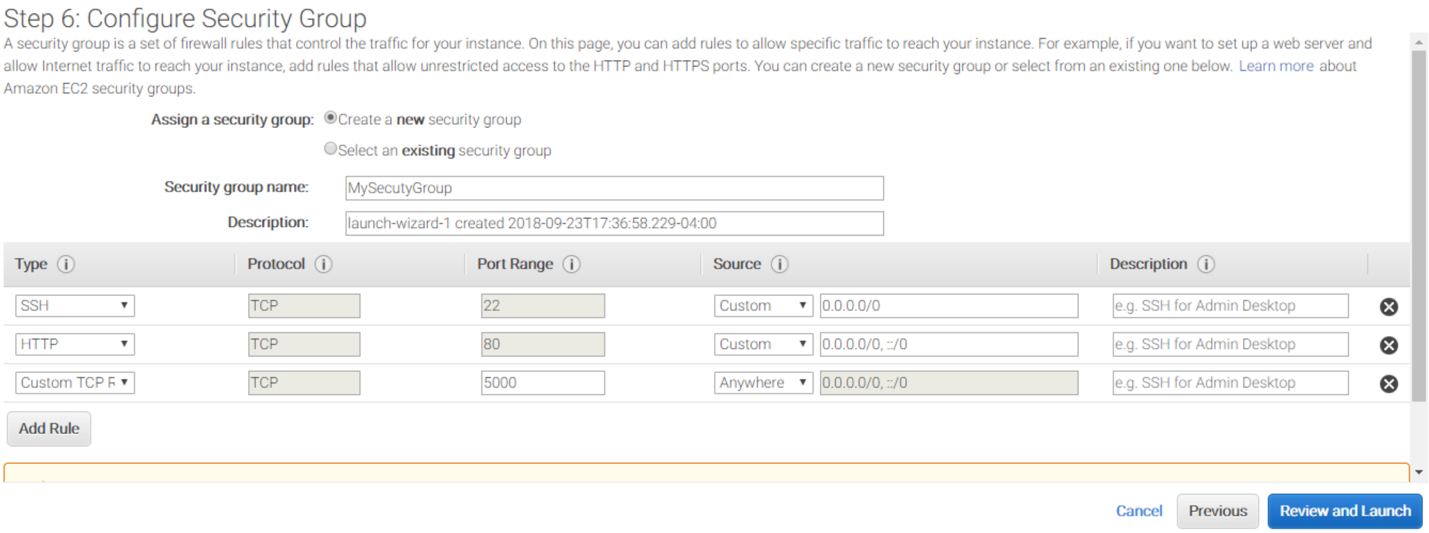
Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver.
A copy of a tag can be applied to volumes, instances or both.
Tags will be applied to all instances and volumes. Learn more about tagging your Amazon EC2 resources.

Key (127 characters maximum)	Value (255 characters maximum)	Instances	Volumes
Name	WEBSERVER	☑	☑

Add another tag (Up to 50 tags maximum)

7. Create a new security group and give it a name with ports 22, 80, 5000 and hit Review and Launch



Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. Learn more about Amazon EC2 security groups.

Assign a security group: ● Create a **new** security group

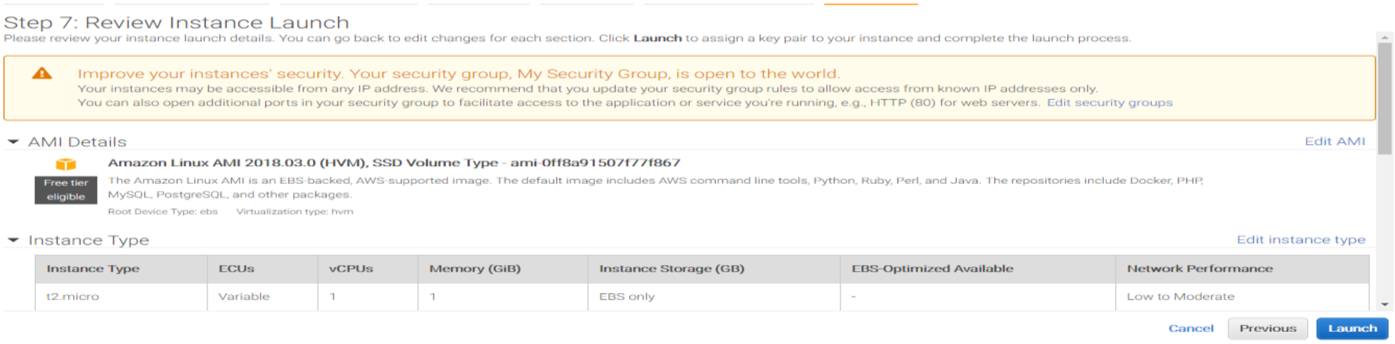
○ Select an **existing** security group

Security group name: MySecutyGroup

Description: launch-wizard-1 created 2018-09-23T17:36:58.229-04:00

Type	Protocol	Port Range	Source		Description
SSH	TCP	22	Custom	0.0.0.0/0	e.g. SSH for Admin Desktop
HTTP	TCP	80	Custom	0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop
Custom TCP R	TCP	5000	Anywhere	0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop

8. Hit Launch



Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

Improve your instances' security. Your security group, My Security Group, is open to the world.
Your instances may be accessible from any IP address. We recommend that you update your security group rules to allow access from known IP addresses only.
You can also open additional ports in your security group to facilitate access to the application or service you're running, e.g., HTTP (80) for web servers. Edit security groups

AMI Details — Edit AMI

Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type - ami-0ff8a91507f77f867

Free tier eligible

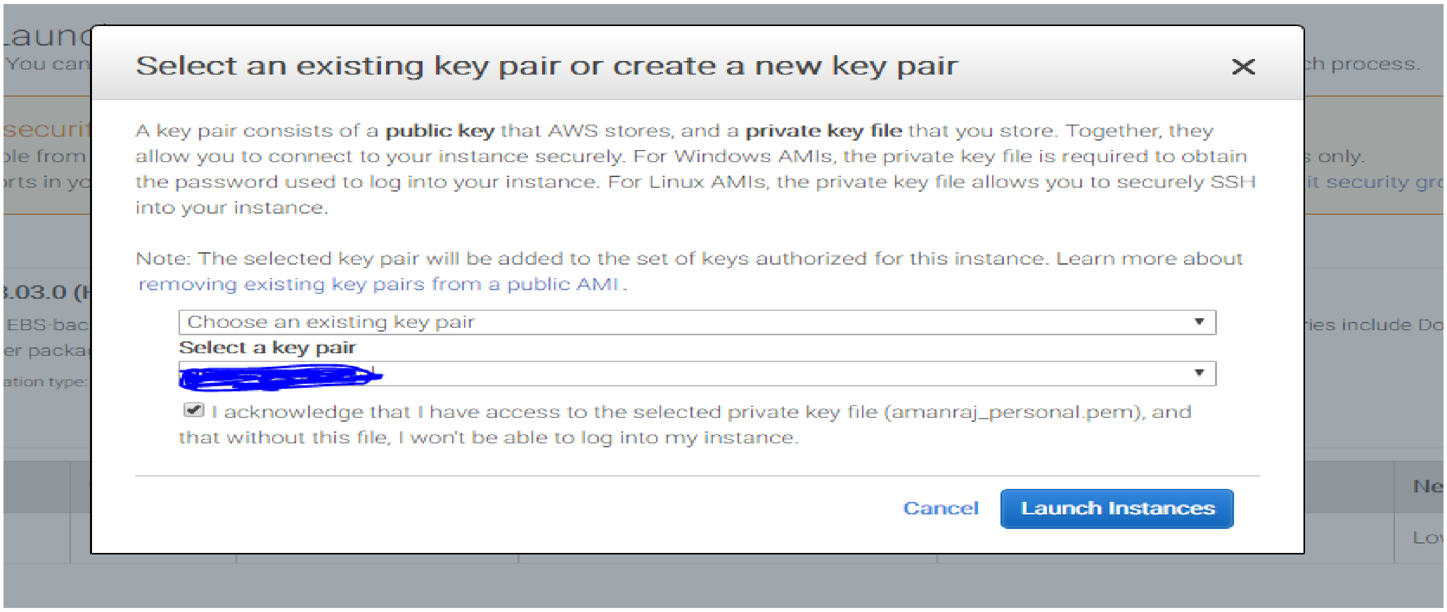
The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages.

Root Device Type: ebs   Virtualization type: hvm

Instance Type — Edit instance type

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

9. Create or use existing key pair



Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI.

Choose an existing key pair

Select a key pair

[redacted]

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

10. Connect to EC2 instance: Refer the below video
https://www.youtube.com/watch?v=bi7ow5NGC-U

Setting up Flask using EC2 Instance

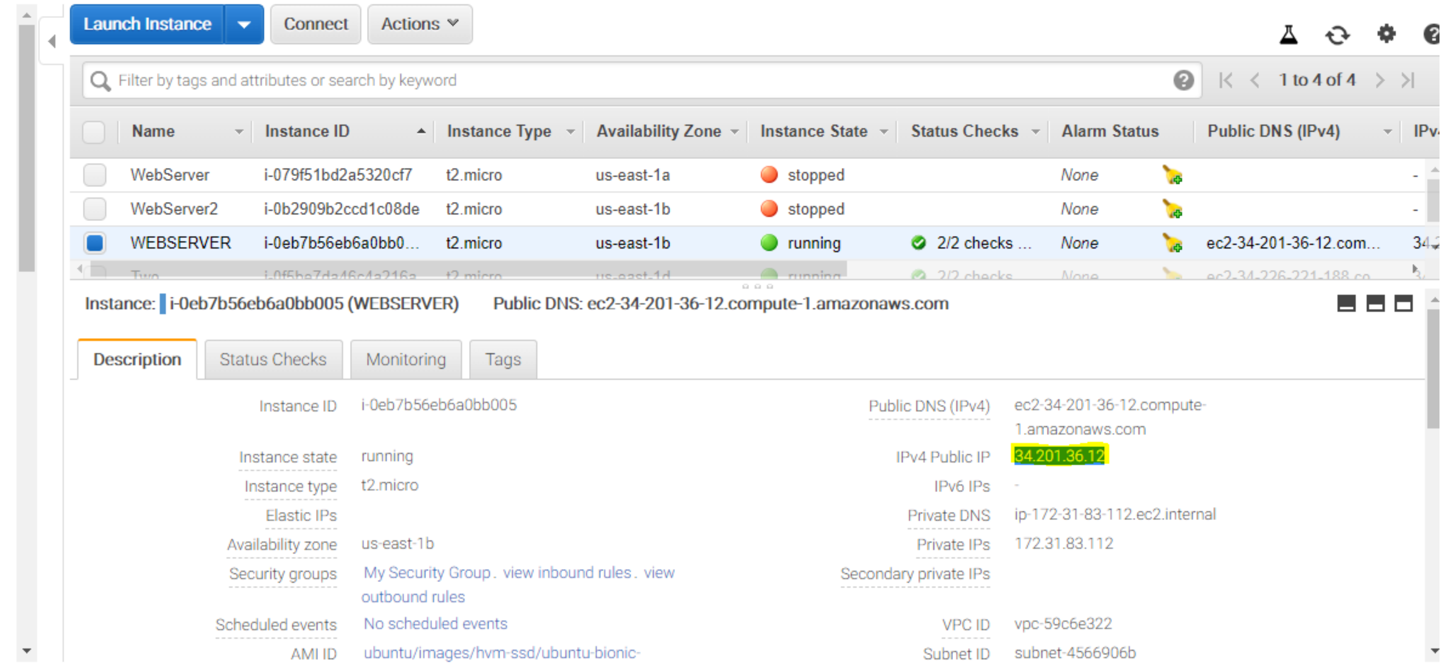
11. Once you are connected to EC2 and you are on your terminal type the following commands
    - Sudo apt update
    - Sudo su
    - sudo apt install python3-pip python3-dev build-essential libssl-dev libffi-dev python3-setuptools
    - sudo apt install python3-venv
    - mkdir ~/myproject
    - cd ~/myproject
    - python3.6 -m venv myprojectenv
    - source myprojectenv/bin/activate
    - pip install wheel
    - pip install uwsgi flask
    - vi ~/myproject/myproject.py
    - press i
    - copy and paste the below code

```
from flask import Flask
app = Flask(__name__)


@app.route("/")
def hello():
    return "<h1 style='color:blue'>Hello There!</h1>"


if __name__ == "__main__":
    app.run(host='0.0.0.0')
```

- to exit and save the above file press esc, colon, type wq and hit enter
- sudo ufw allow 5000
- python myproject.py
- your file should give the below output

```
Output
* Serving Flask app "myproject" (lazy loading)
 * Environment: production
   WARNING: Do not use the development server in a production environment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
```

- Once you see the below out on the terminal. Go back to the console and copy your public IP and go to the bowser and paste your public ip:5000

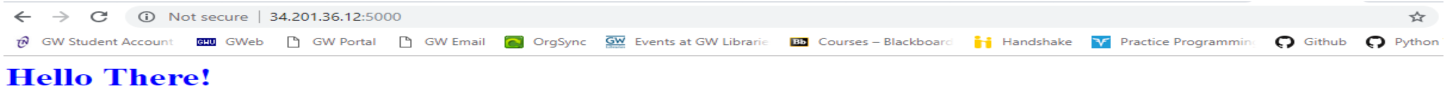


	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv
□	WebServer	i-079f51bd2a5320cf7	t2.micro	us-east-1a	● stopped		None		-
□	WebServer2	i-0b2909b2ccd1c08de	t2.micro	us-east-1b	● stopped		None		-
■	WEBSERVER	i-0eb7b56eb6a0bb0...	t2.micro	us-east-1b	● running	✓ 2/2 checks ...	None	ec2-34-201-36-12.com...	34.
□	Two	i-0f5ba7da46c4a216a	t2.micro	us-east-1d	● running	✓ 2/2 checks	None	ec2-34-226-221-188.co	3

Instance: i-0eb7b56eb6a0bb005 (WEBSERVER)   Public DNS: ec2-34-201-36-12.compute-1.amazonaws.com

Instance ID	i-0eb7b56eb6a0bb005	Public DNS (IPv4)	ec2-34-201-36-12.compute-1.amazonaws.com
Instance state	running	IPv4 Public IP	34.201.36.12
Instance type	t2.micro	IPv6 IPs	-
Elastic IPs		Private DNS	ip-172-31-83-112.ec2.internal
Availability zone	us-east-1b	Private IPs	172.31.83.112
Security groups	My Security Group. view inbound rules. view outbound rules	Secondary private IPs	
Scheduled events	No scheduled events	VPC ID	vpc-59c6e322
AMI ID	ubuntu/images/hvm-ssd/ubuntu-bionic-	Subnet ID	subnet-4566906b

12. Should display hello world



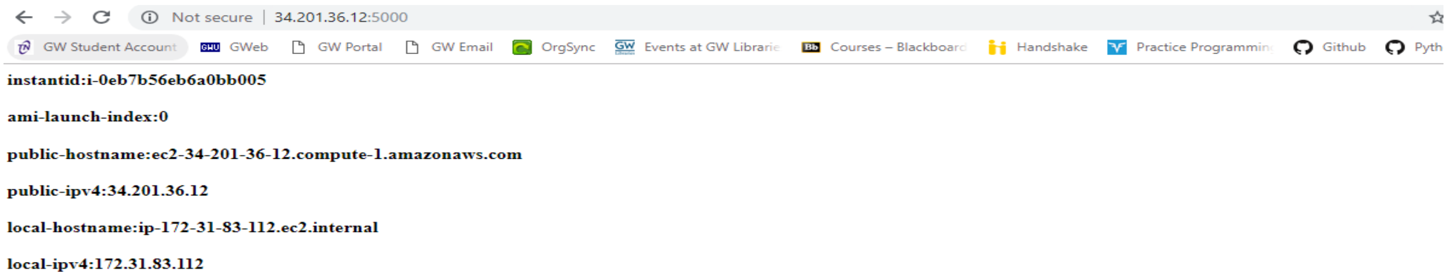
Hello There!

Web application to get instance metadata

13. Go back to console and do ctrl c
14. Type the following commands to start the process for getting the instance metadata
    - curl http:/test/meta-data/instance-id > a.txt
    - curl http:/test/meta-data/ami-launch-index > b.txt
    - curl http://169.254.169.254/latest/meta-data/public-hostname > c.txt
    - curl http://169.254.169.254/latest/meta-data/public-ipv4 > d.txt
    - curl http://169.254.169.254/latest/meta-data/local-hostname > e.txt
    - curl http://169.254.169.254/latest/meta-data/local-ipv4 > f.txt
15. Type vi myproject.py and delete the complete code and enter the below code.

```
from flask import Flask

application = Flask(__name__)


@application.route("/")

def hello():
    content = " "

    with open("a.txt",'r') as rd1:

        instantid = rd1.read()

    with open("b.txt",'r') as rd2:

        index = rd2.read()

    with open("c.txt",'r') as rd3:

        phostname = rd3.read()

    with open("d.txt",'r') as rd4:

        pip = rd4.read()

    with open("e.txt",'r') as rd5:

        lhostname = rd5.read()

    with open("f.txt",'r') as rd6:

        lip = rd6.read()

        content +="<h4>instantid:"+instantid+"</h4>"

        content +="<h4>ami-launch-index:"+index+"</h4>"

        content +="<h4>public-hostname:"+phostname+"</h4>"

        content +="<h4>public-ipv4:"+pip+"</h4>"

        content +="<h4>local-hostname:"+lhostname+"</h4>"

        content +="<h4>local-ipv4:"+lip+"</h4>"


    return content


if __name__ == "__main__":

  application.run(host='0.0.0.0')
```

16. Press esc, colon, and wq and run the file by typing python myproject.py.
17. Go back to browser and type the public ip:5000. It would display the instance metadata.

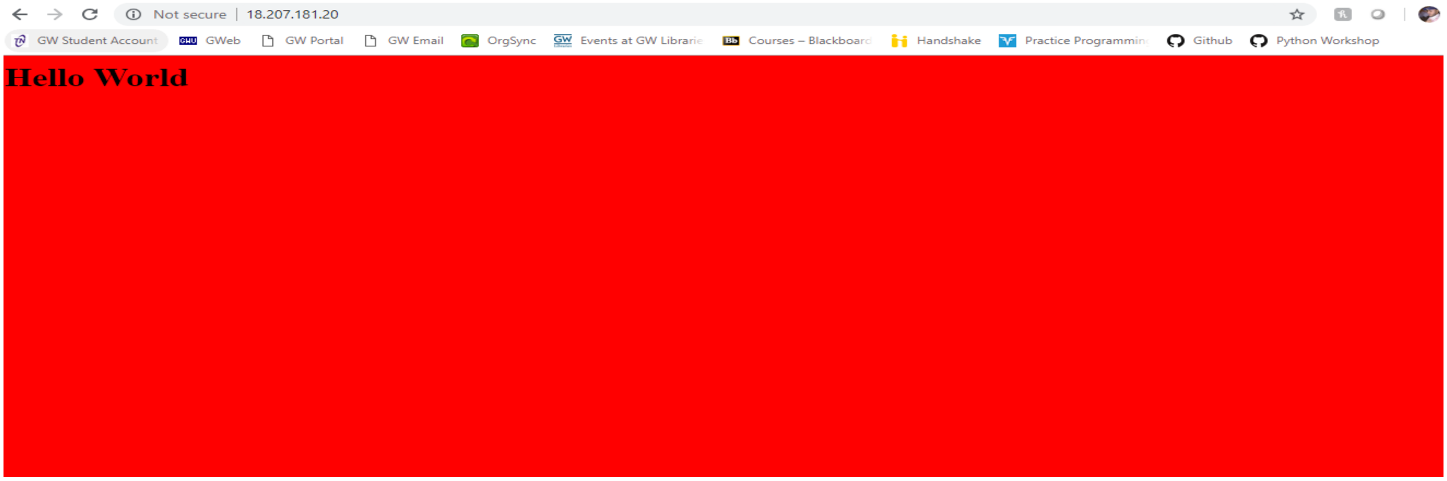


```
instantid:i-0eb7b56eb6a0bb005

ami-launch-index:0

public-hostname:ec2-34-201-36-12.compute-1.amazonaws.com

public-ipv4:34.201.36.12

local-hostname:ip-172-31-83-112.ec2.internal

local-ipv4:172.31.83.112
```

Create a Load Balancer and balance the load of web application

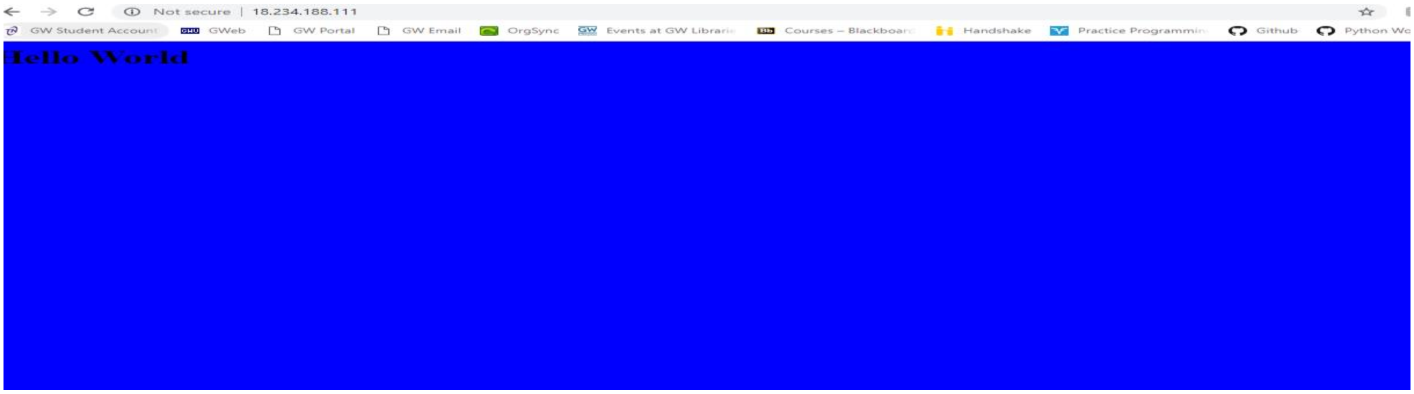
18. Delete the previous ec2 instance and create 2 new EC2 instances with Amazon Linux AMI 2018.03 and 2 different availability zones.
19. Connect your instances and go to terminal.
20. Type the below commands and make a webpage in first instance.
    - Sudo yum update
    - Sudo su
    - Sudo yum install httpd
    - Sudo service httpd start
    - sudo chkconfig httpd on
    - sudo vi /var/www/html/index.html
    - Hit i and Copy paste the below code

      ```
      <html>

      <body>
      <body bgcolor='red'>
      <h1>Hello World</h1>
      </body>

      </html>
      ```
    - Press esc, colon, wq.
    - Sudo service httpd restart
    - Copy the Public IP and paste in the browser and it should display the Hello World page



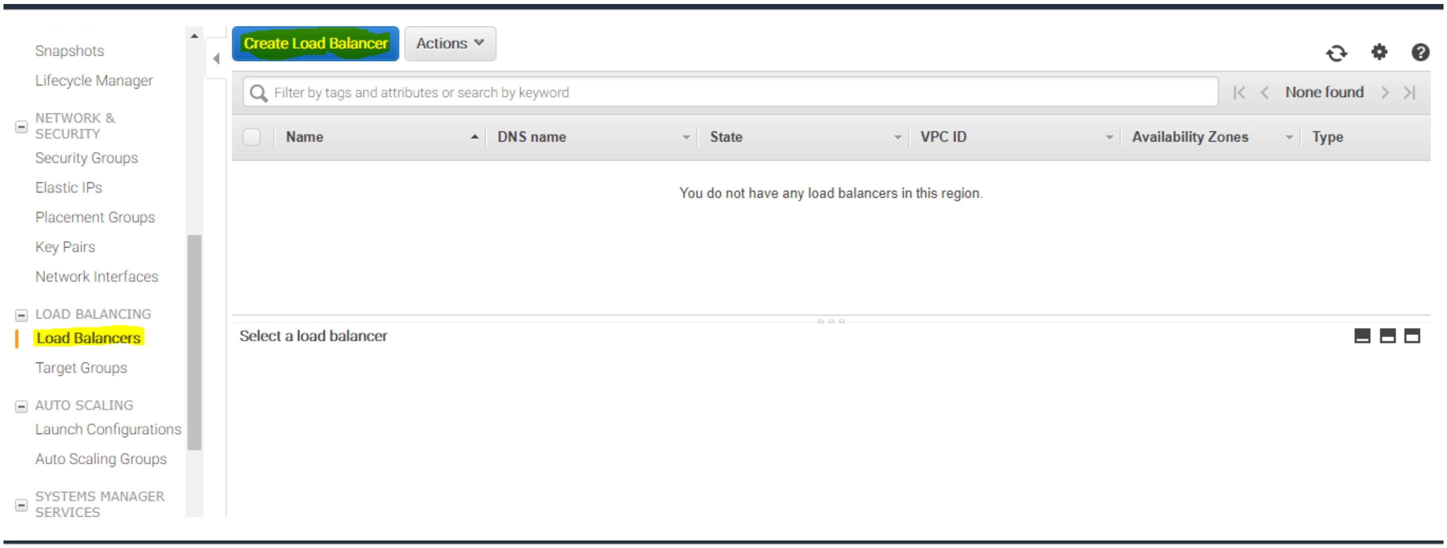
Hello World

21. Do the same for second instance. The only change would be the color. Change the color to blue and it should display Hello World like this.



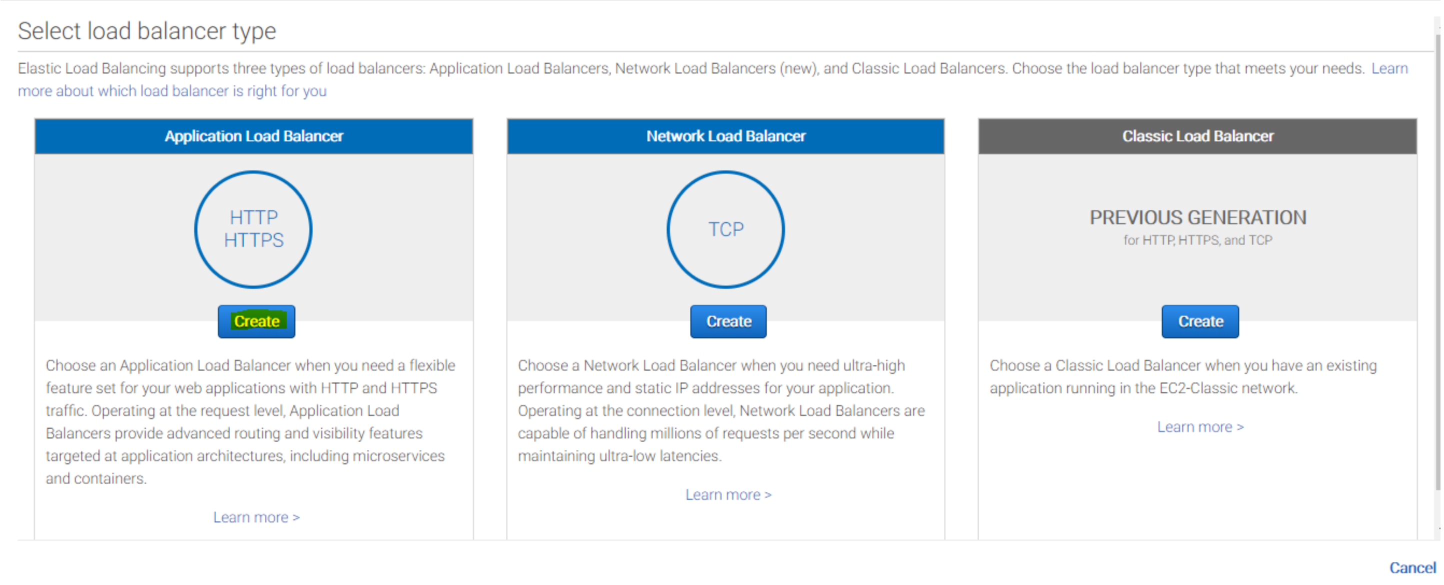
Hello World

22. Create a Load Balancer. On the console, in the left pane click on Load balancer.



You do not have any load balancers in this region.

Select a load balancer



Select load balancer type

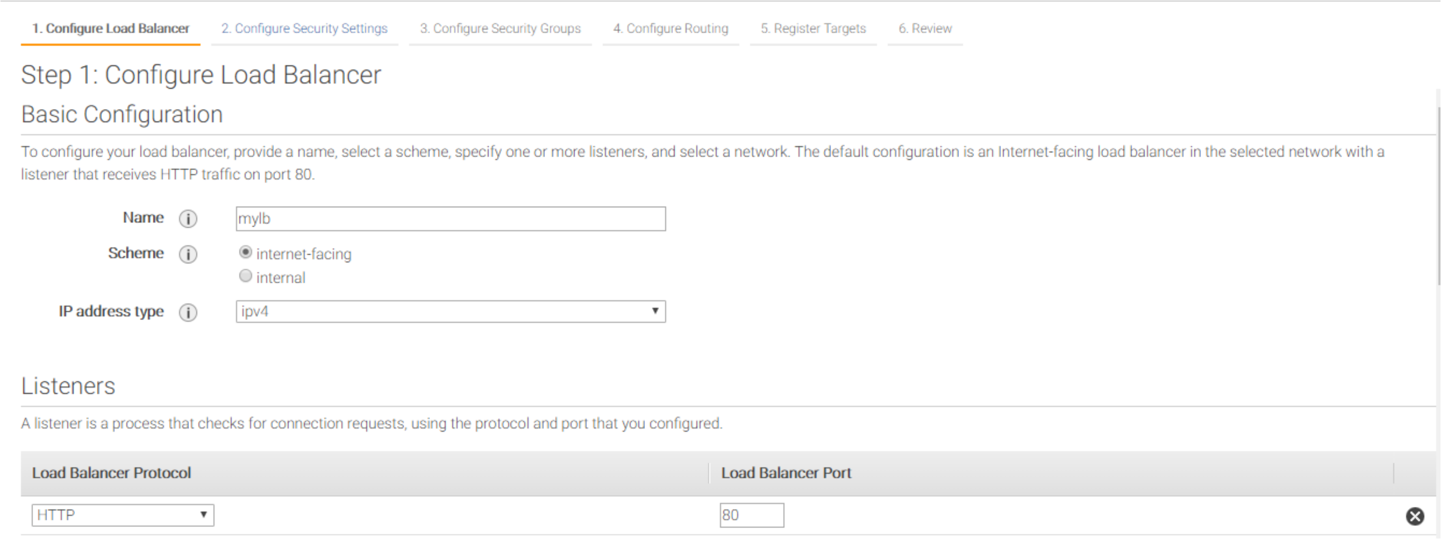
Elastic Load Balancing supports three types of load balancers: Application Load Balancers, Network Load Balancers (new), and Classic Load Balancers. Choose the load balancer type that meets your needs. Learn more about which load balancer is right for you

Choose an Application Load Balancer when you need a flexible feature set for your web applications with HTTP and HTTPS traffic. Operating at the request level, Application Load Balancers provide advanced routing and visibility features targeted at application architectures, including microservices and containers.

Choose a Network Load Balancer when you need ultra-high performance and static IP addresses for your application. Operating at the connection level, Network Load Balancers are capable of handling millions of requests per second while maintaining ultra-low latencies.

Choose a Classic Load Balancer when you have an existing application running in the EC2-Classic network.

23. Give a name for your Load Balancer and select all availability zones.


1. Configure Load Balancer
2. Configure Security Settings
3. Configure Security Groups
4. Configure Routing
5. Register Targets
6. Review

Step 1: Configure Load Balancer

Basic Configuration

To configure your load balancer, provide a name, select a scheme, specify one or more listeners, and select a network. The default configuration is an Internet-facing load balancer in the selected network with a listener that receives HTTP traffic on port 80.

Name ⓘ mylb

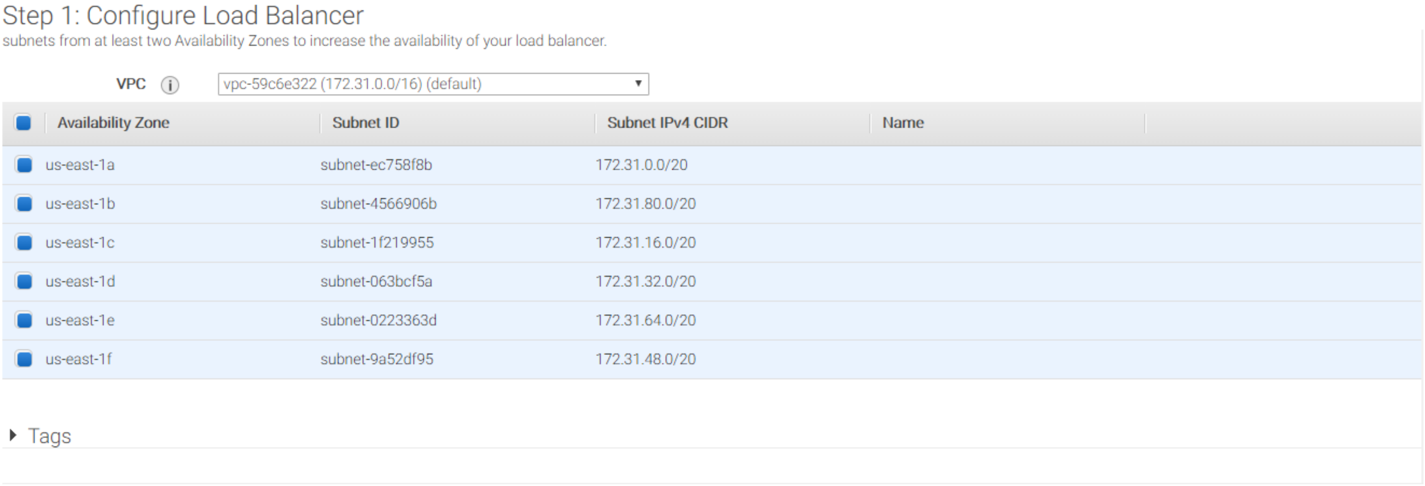
Scheme ⓘ ● internet-facing ○ internal

IP address type ⓘ ipv4

Listeners

A listener is a process that checks for connection requests, using the protocol and port that you configured.

Load Balancer Protocol	Load Balancer Port
HTTP	80



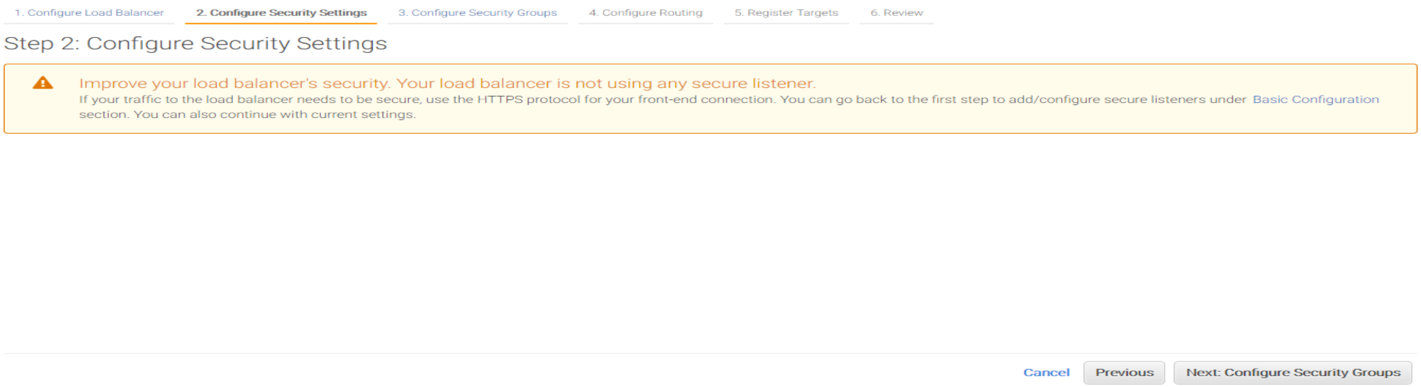
Step 1: Configure Load Balancer

subnets from at least two Availability Zones to increase the availability of your load balancer.

VPC ⓘ vpc-59c6e322 (172.31.0.0/16) (default)

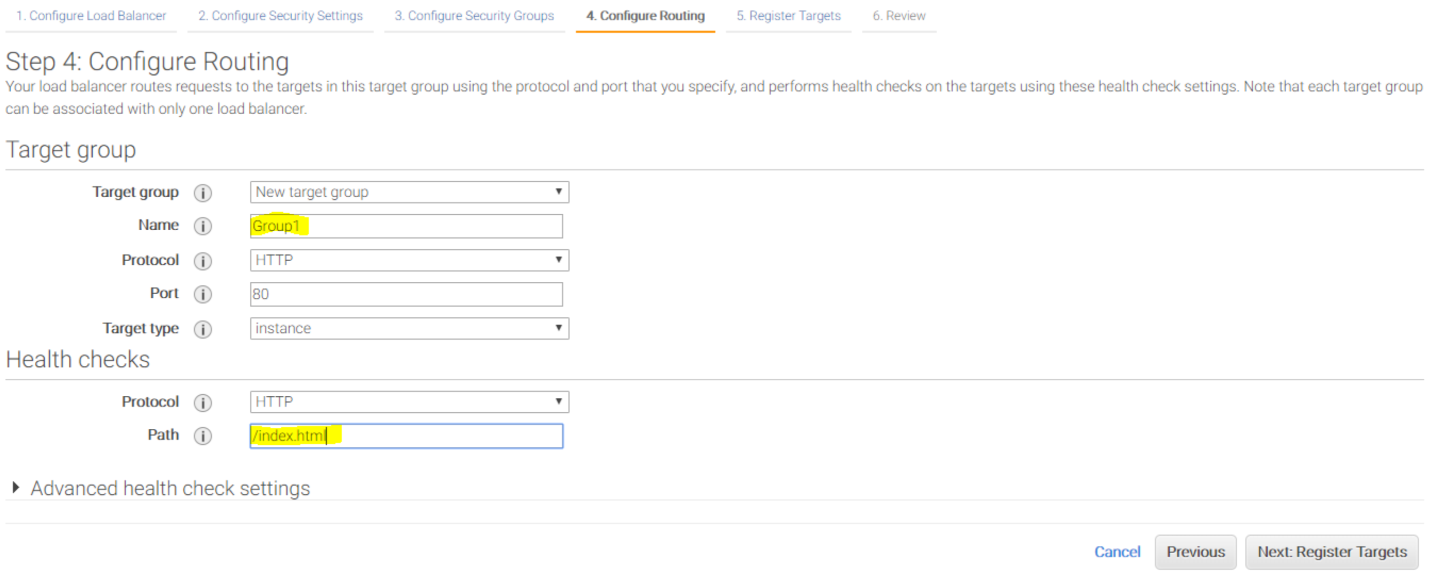
☑	Availability Zone	Subnet ID	Subnet IPv4 CIDR	Name
☑	us-east-1a	subnet-ec758f8b	172.31.0.0/20	
☑	us-east-1b	subnet-4566906b	172.31.80.0/20	
☑	us-east-1c	subnet-1f219955	172.31.16.0/20	
☑	us-east-1d	subnet-063bcf5a	172.31.32.0/20	
☑	us-east-1e	subnet-0223363d	172.31.64.0/20	
☑	us-east-1f	subnet-9a52df95	172.31.48.0/20	

24. Click Next


1. Configure Load Balancer
2. Configure Security Settings
3. Configure Security Groups
4. Configure Routing
5. Register Targets
6. Review

Step 2: Configure Security Settings

⚠ Improve your load balancer's security. Your load balancer is not using any secure listener.
If your traffic to the load balancer needs to be secure, use the HTTPS protocol for your front-end connection. You can go back to the first step to add/configure secure listeners under Basic Configuration section. You can also continue with current settings.


1. Configure Load Balancer
2. Configure Security Settings
3. Configure Security Groups
4. Configure Routing
5. Register Targets
6. Review

Step 4: Configure Routing

Your load balancer routes requests to the targets in this target group using the protocol and port that you specify, and performs health checks on the targets using these health check settings. Note that each target group can be associated with only one load balancer.

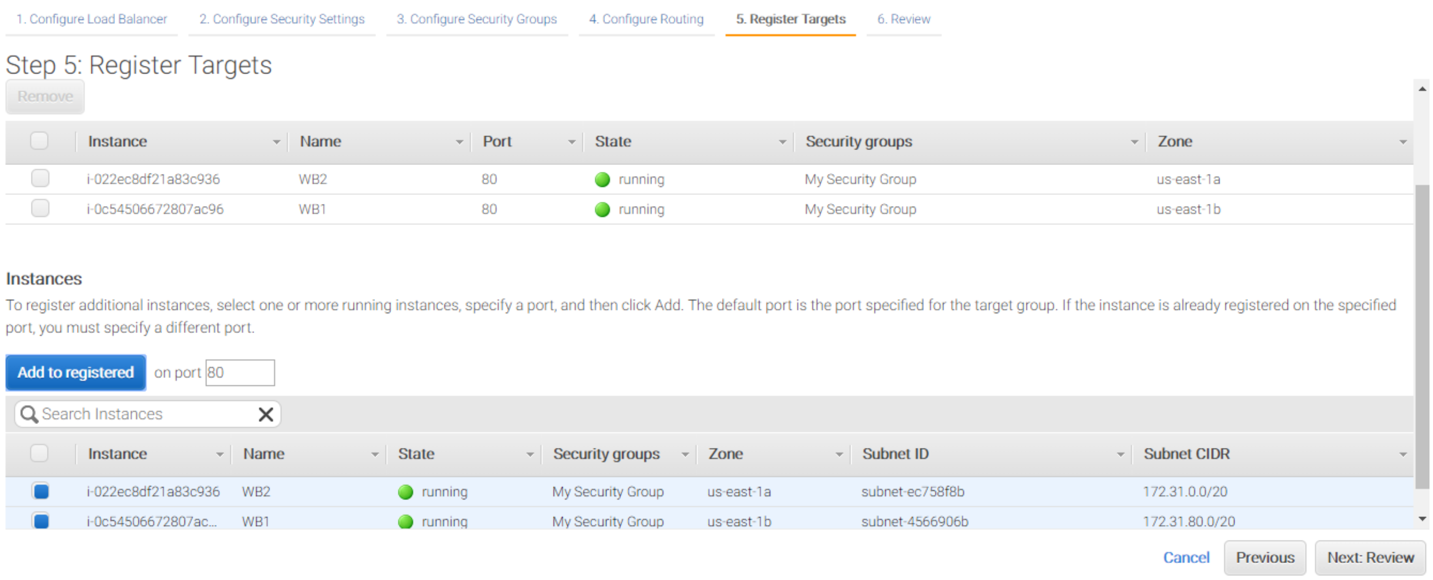
Target group

Target group	New target group
Name	Group1
Protocol	HTTP
Port	80
Target type	instance

Health checks

Protocol	HTTP
Path	/index.html

25. Click on instances and click on add to registered.


1. Configure Load Balancer
2. Configure Security Settings
3. Configure Security Groups
4. Configure Routing
5. Register Targets
6. Review
Remove

Step 5: Register Targets

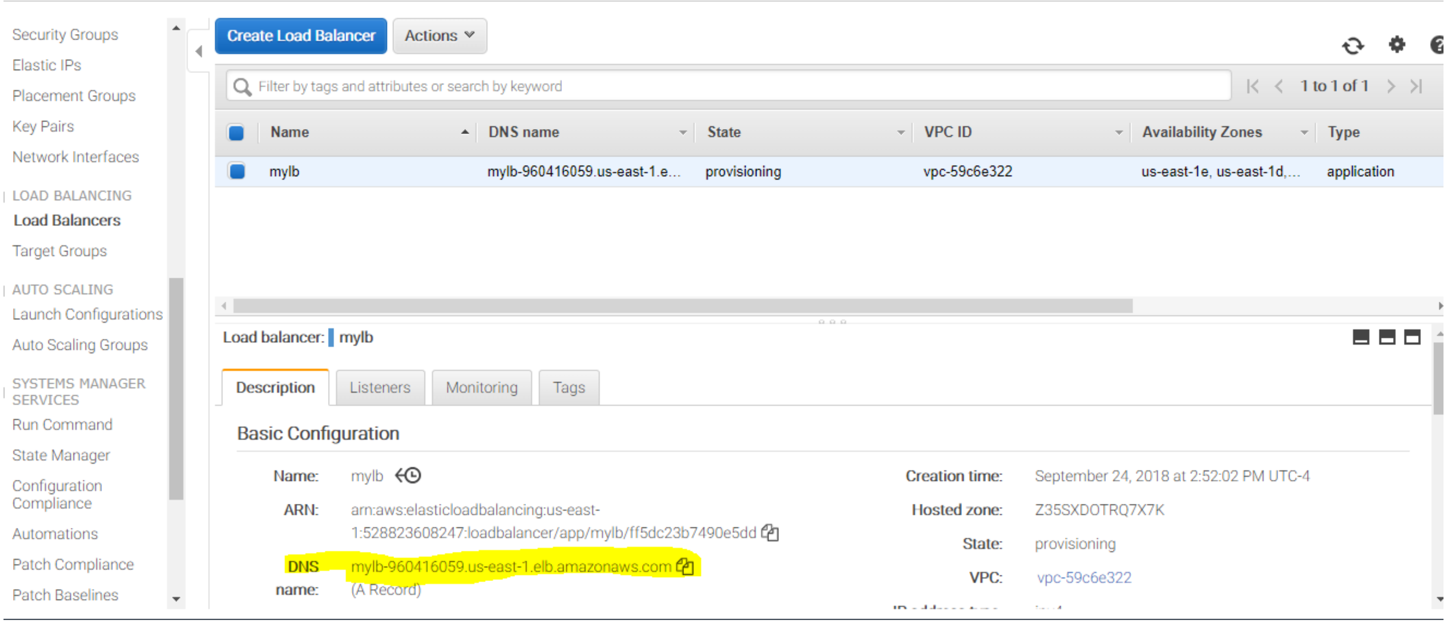
□	Instance	Name	Port	State	Security groups	Zone
□	i-022ec8df21a83c936	WB2	80	● running	My Security Group	us-east-1a
□	i-0c54506672807ac96	WB1	80	● running	My Security Group	us-east-1b

Instances

To register additional instances, select one or more running instances, specify a port, and then click Add. The default port is the port specified for the target group. If the instance is already registered on the specified port, you must specify a different port.

□	Instance	Name	State	Security groups	Zone	Subnet ID	Subnet CIDR
☑	i-022ec8df21a83c936	WB2	● running	My Security Group	us-east-1a	subnet-ec758f8b	172.31.0.0/20
☑	i-0c54506672807ac...	WB1	● running	My Security Group	us-east-1b	subnet-4566906b	172.31.80.0/20

26. Click Review and Launch
27. Go back to the load balancer and copy the DNS and paste in browser. It should display webpage from your first instance and when you refresh the page it should display the web page from second instance.



Name	DNS name	State	VPC ID	Availability Zones	Type
mylb	mylb-960416059.us-east-1.e...	provisioning	vpc-59c6e322	us-east-1e, us-east-1d,...	application

Load balancer: mylb

Basic Configuration

Name: mylb

ARN: arn:aws:elasticloadbalancing:us-east-1:528823608247:loadbalancer/app/mylb/ff5dc23b7490e5dd

DNS name: mylb-960416059.us-east-1.elb.amazonaws.com (A Record)

Creation time: September 24, 2018 at 2:52:02 PM UTC-4

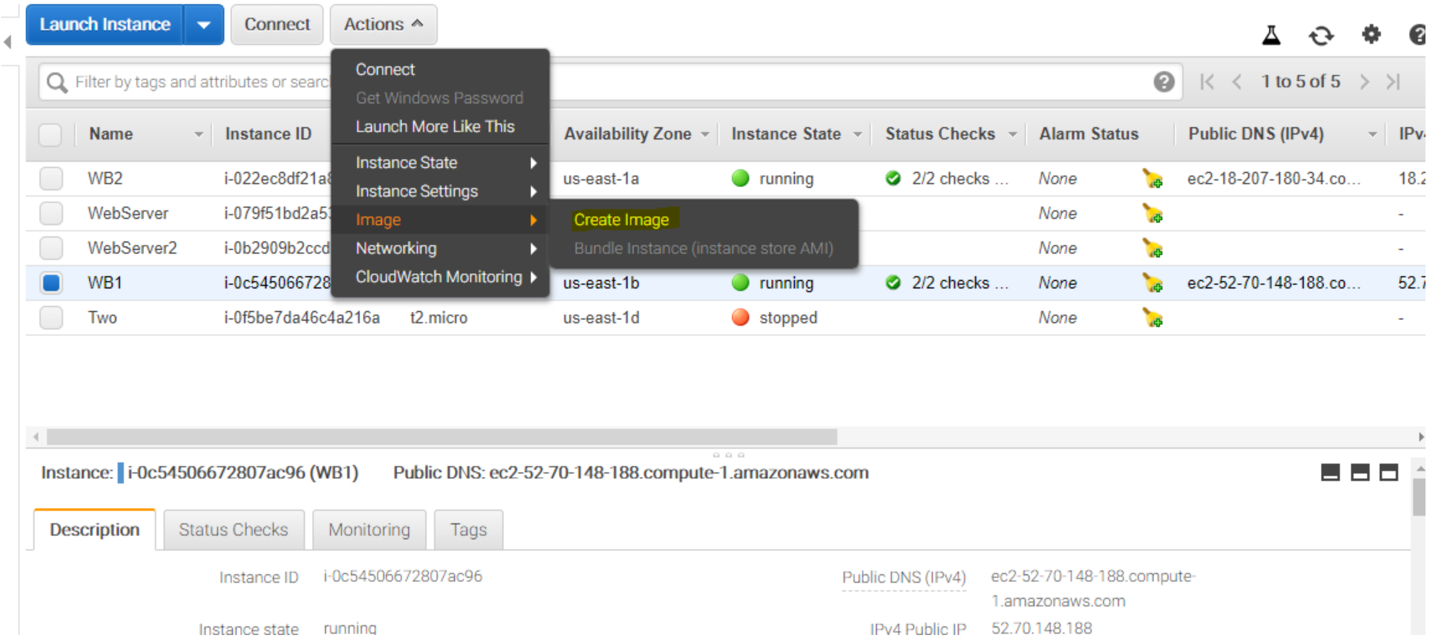
Hosted zone: Z35SXDOTRQ7X7K

State: provisioning

VPC: vpc-59c6e322

Set up the boto python library and custom AMI

28. Delete the Load balancer you created and the target groups.
29. Go to the Windows powershell and type the below command
    - Pip install boto
30. Create AMI for one of the two instances. Go to the console, click on first instance, go to Actions and create image



Name	Instance ID		Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv
WB2	i-022ec8df21a		us-east-1a	running	2/2 checks ...	None	ec2-18-207-180-34.co...	18.2
WebServer	i-079f51bd2a5					None		-
WebServer2	i-0b2909b2ccd					None		-
WB1	i-0c545066728		us-east-1b	running	2/2 checks ...	None	ec2-52-70-148-188.co...	52.7
Two	i-0f5be7da46c4a216a	t2.micro	us-east-1d	stopped		None		-

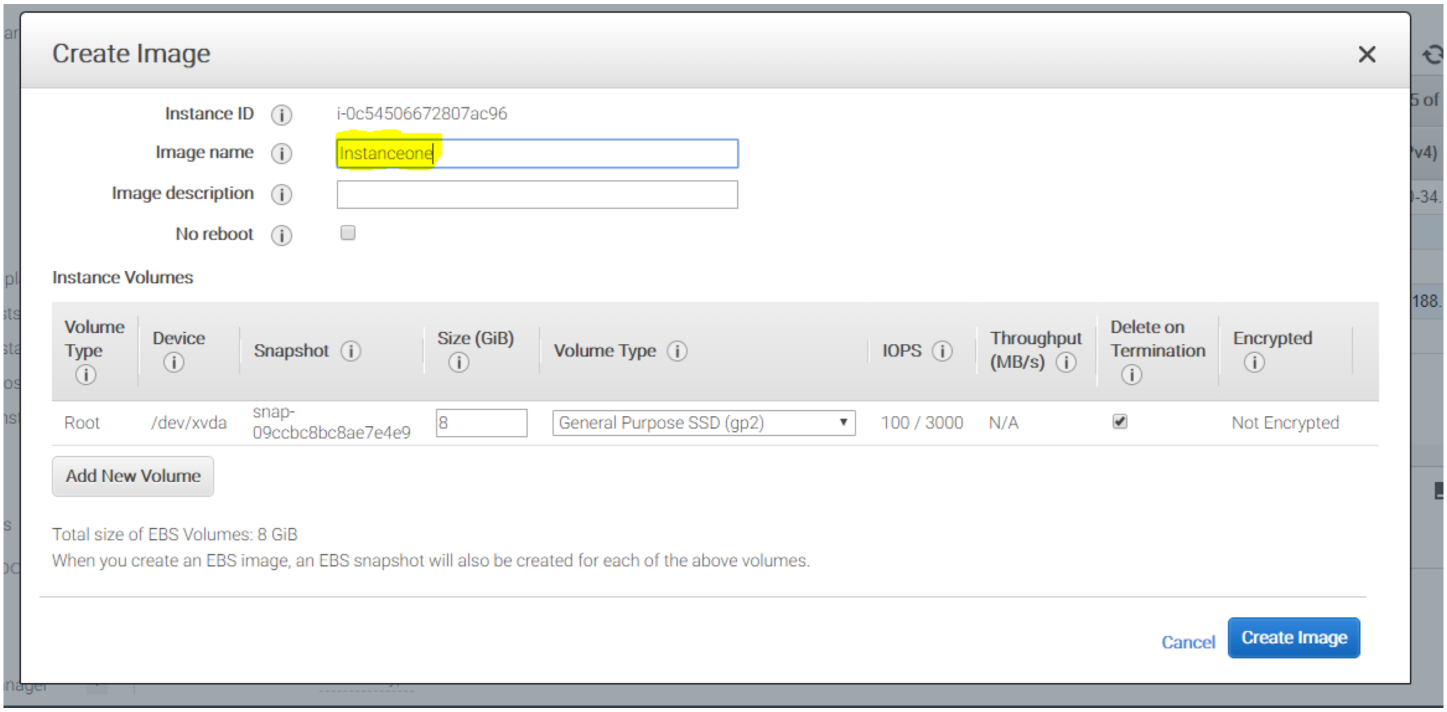
Instance: i-0c54506672807ac96 (WB1) Public DNS: ec2-52-70-148-188.compute-1.amazonaws.com

Instance ID i-0c54506672807ac96

Instance state running

Public DNS (IPv4) ec2-52-70-148-188.compute-1.amazonaws.com

IPv4 Public IP 52.70.148.188



Create Image

Instance ID: i-0c54506672807ac96

Image name: Instanceone

Image description:

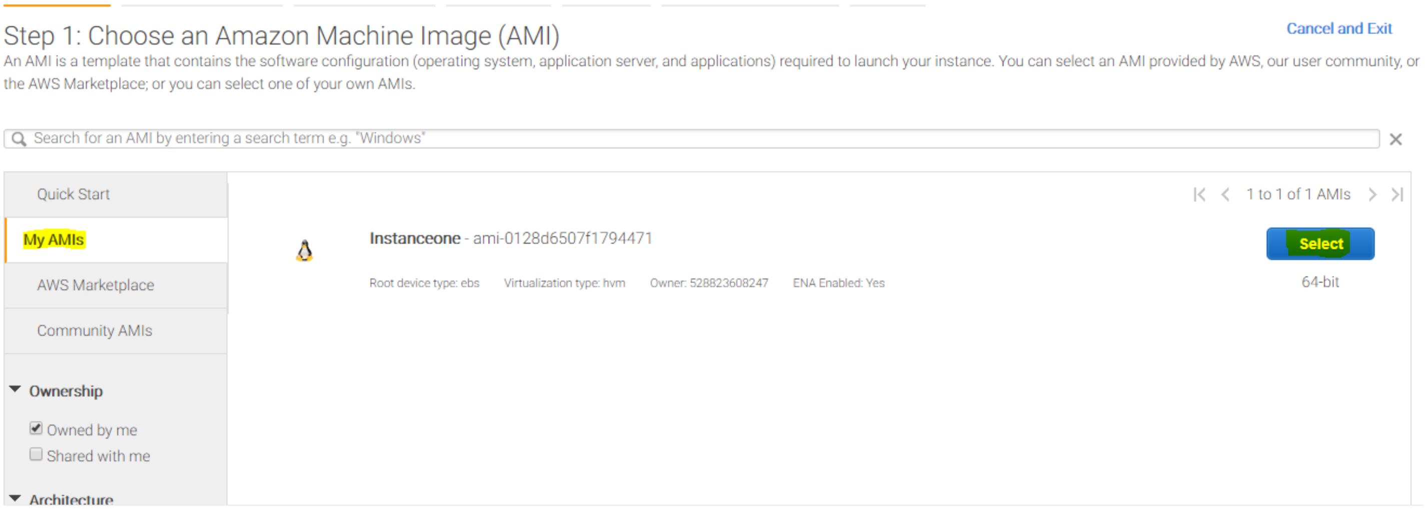
No reboot: □

Instance Volumes

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encrypted
Root	/dev/xvda	snap-09ccbc8bc8ae7e4e9	8	General Purpose SSD (gp2)	100 / 3000	N/A	☑	Not Encrypted

Total size of EBS Volumes: 8 GiB
When you create an EBS image, an EBS snapshot will also be created for each of the above volumes.

31. Once the image the created, delete the instance for which you created the image.
32. Go to Load balancer section and create a new load balancer as we did in the previous steps and create a new target group and add the second instance you have.
33. After this Go to the console and create the new instance using AMI that you just created as per the below screenshot.



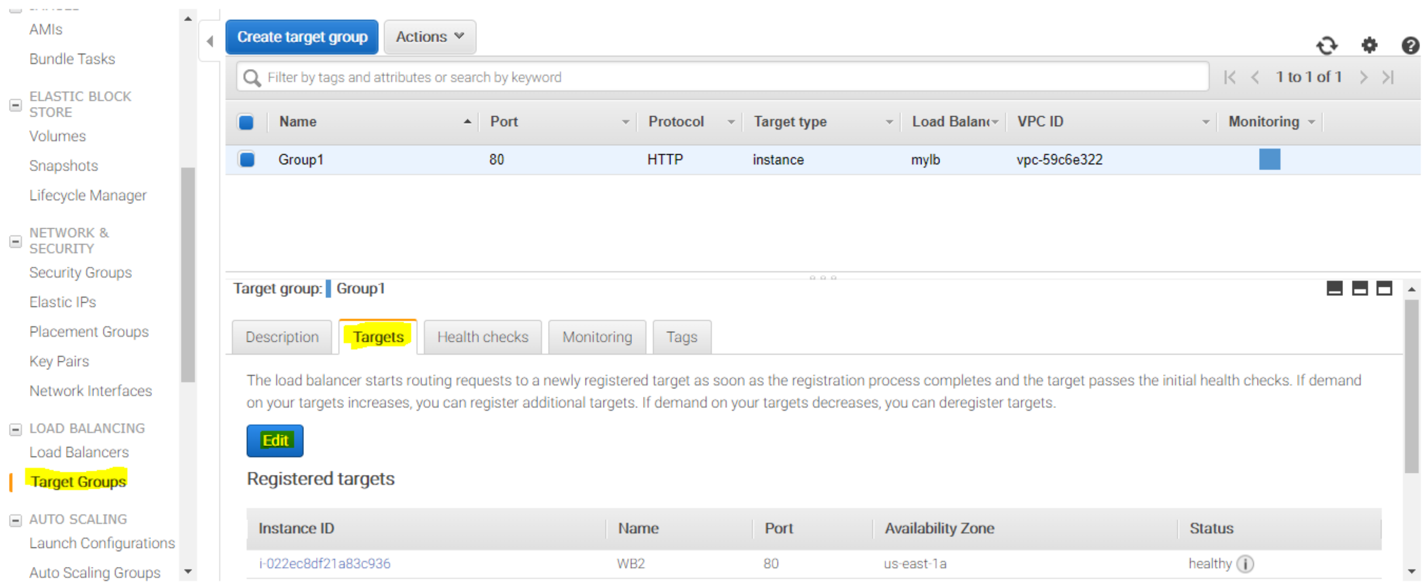
Step 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

Instanceone - ami-0128d6507f1794471

Root device type: ebs   Virtualization type: hvm   Owner: 528823608247   ENA Enabled: Yes

34. Select the AMI and proceed with normal instance creation. Select the security group that we created before and launch the instance. Also select a different availability zone than your previous instance
35. Once the instance has been provisioned, go to Load balancer section and open the target group and edit it.



☐	Name	Port	Protocol	Target type	Load Balanc	VPC ID	Monitoring
☑	Group1	80	HTTP	instance	mylb	vpc-59c6e322	▨

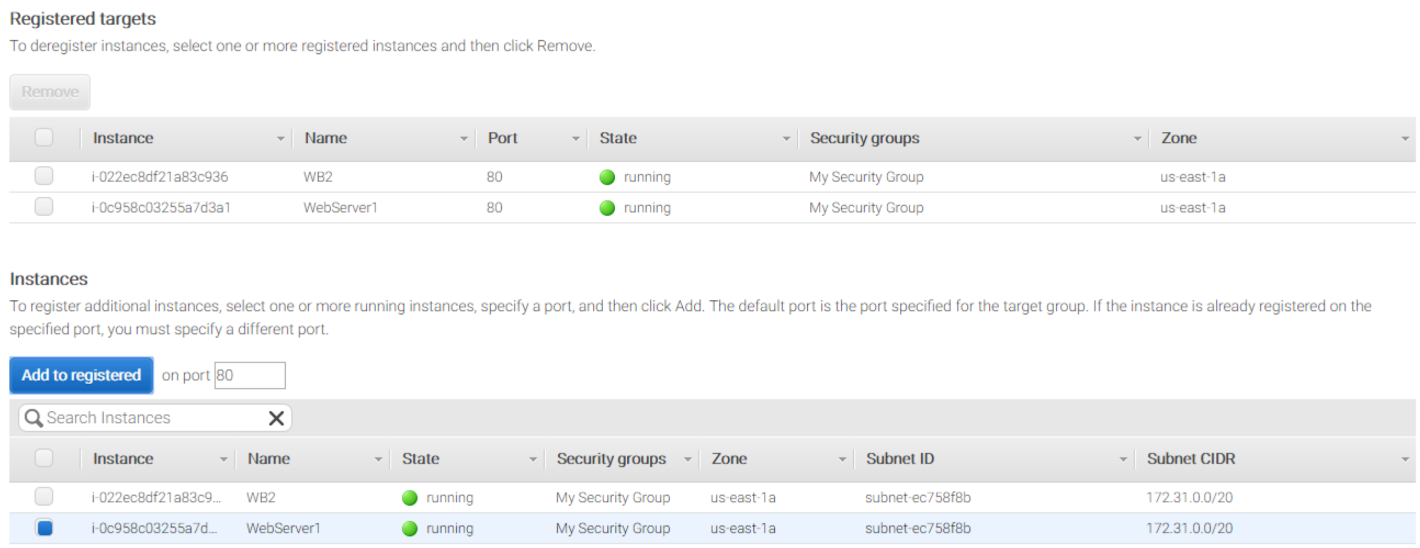
Target group: Group1

The load balancer starts routing requests to a newly registered target as soon as the registration process completes and the target passes the initial health checks. If demand on your targets increases, you can register additional targets. If demand on your targets decreases, you can deregister targets.

Registered targets

Instance ID	Name	Port	Availability Zone	Status
i-022ec8df21a83c936	WB2	80	us-east-1a	healthy ⓘ

36. Select the instance you just created and add to registered and click save.



Registered targets

To deregister instances, select one or more registered instances and then click Remove.

☐	Instance	Name	Port	State	Security groups	Zone
☐	i-022ec8df21a83c936	WB2	80	● running	My Security Group	us-east-1a
☐	i-0c958c03255a7d3a1	WebServer1	80	● running	My Security Group	us-east-1a

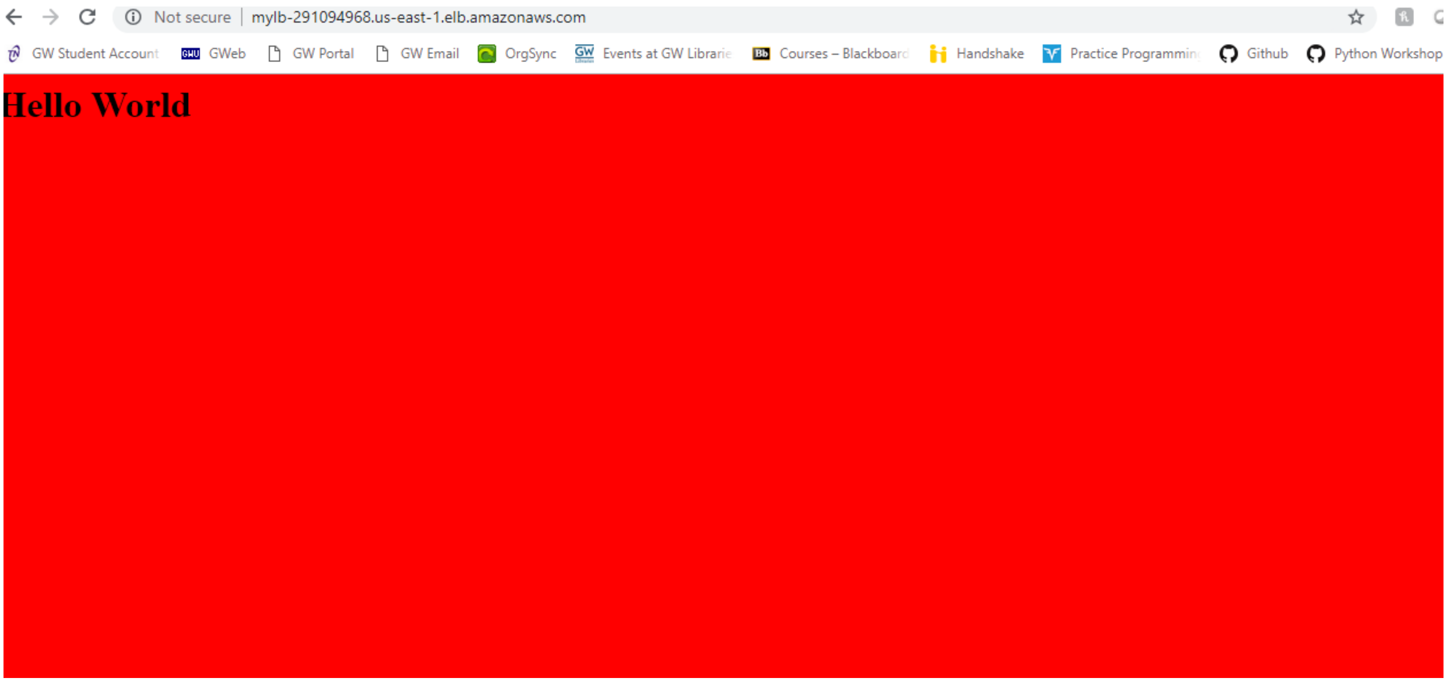
Instances

To register additional instances, select one or more running instances, specify a port, and then click Add. The default port is the port specified for the target group. If the instance is already registered on the specified port, you must specify a different port.

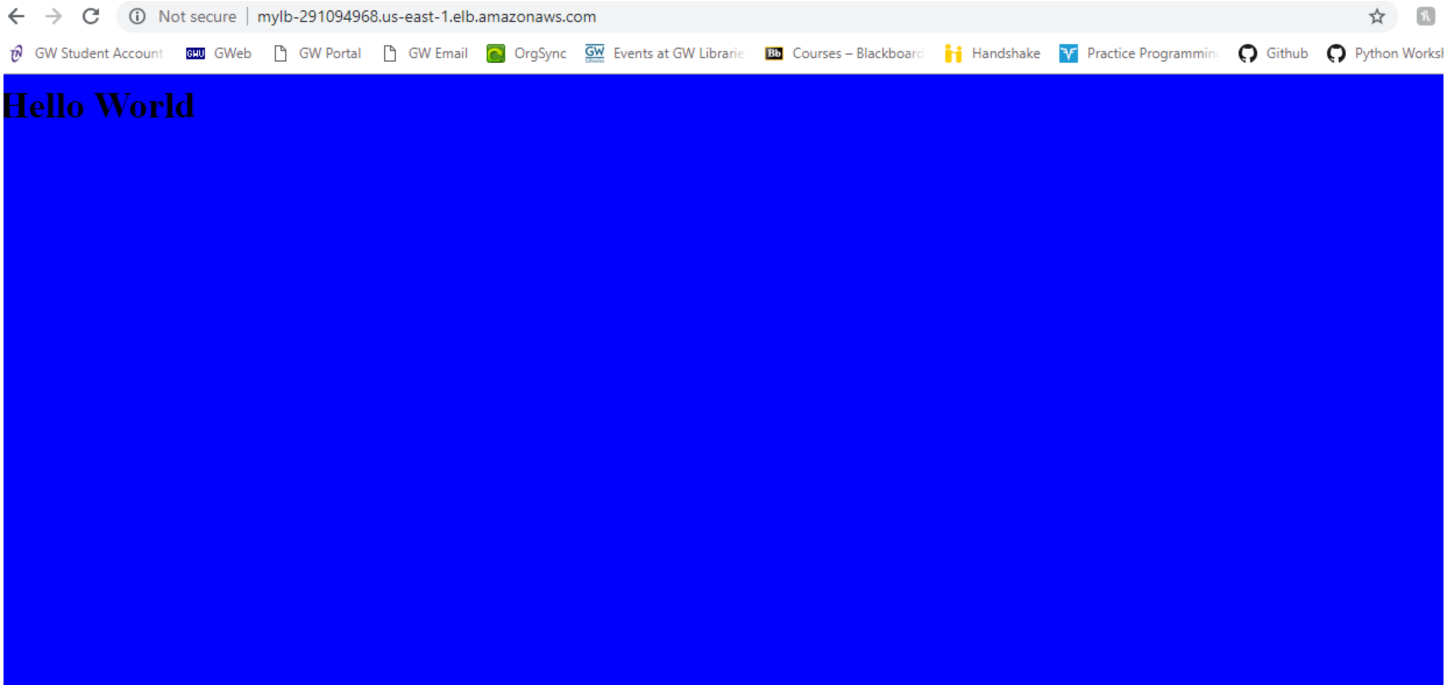
on port 80

☐	Instance	Name	State	Security groups	Zone	Subnet ID	Subnet CIDR
☐	i-022ec8df21a83c9...	WB2	● running	My Security Group	us-east-1a	subnet-ec758f8b	172.31.0.0/20
☑	i-0c958c03255a7d...	WebServer1	● running	My Security Group	us-east-1a	subnet-ec758f8b	172.31.0.0/20

37. Go back to browser and test the load balancer and it should display both web pages if you refresh it.



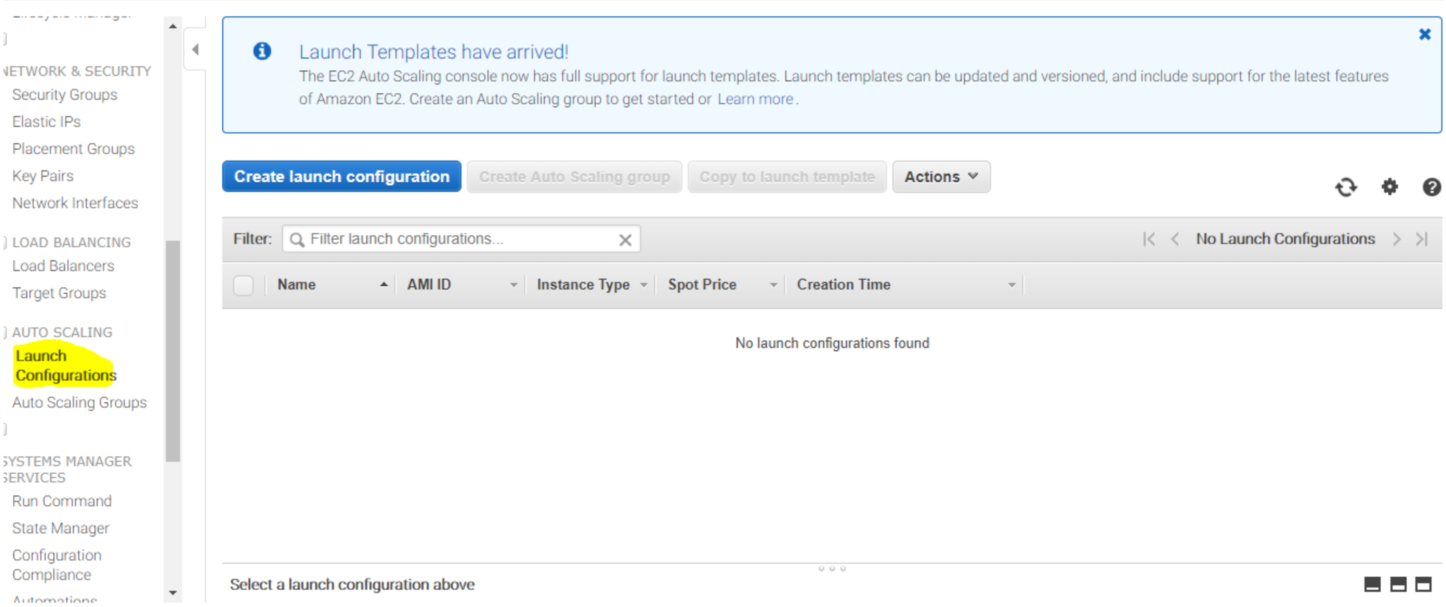
Hello World



Hello World

Autoconfiguration and Autoscaling

38. Go to Auto scaling, launch configuration section and create launch configuration



Launch Templates have arrived!

The EC2 Auto Scaling console now has full support for launch templates. Launch templates can be updated and versioned, and include support for the latest features of Amazon EC2. Create an Auto Scaling group to get started or Learn more.

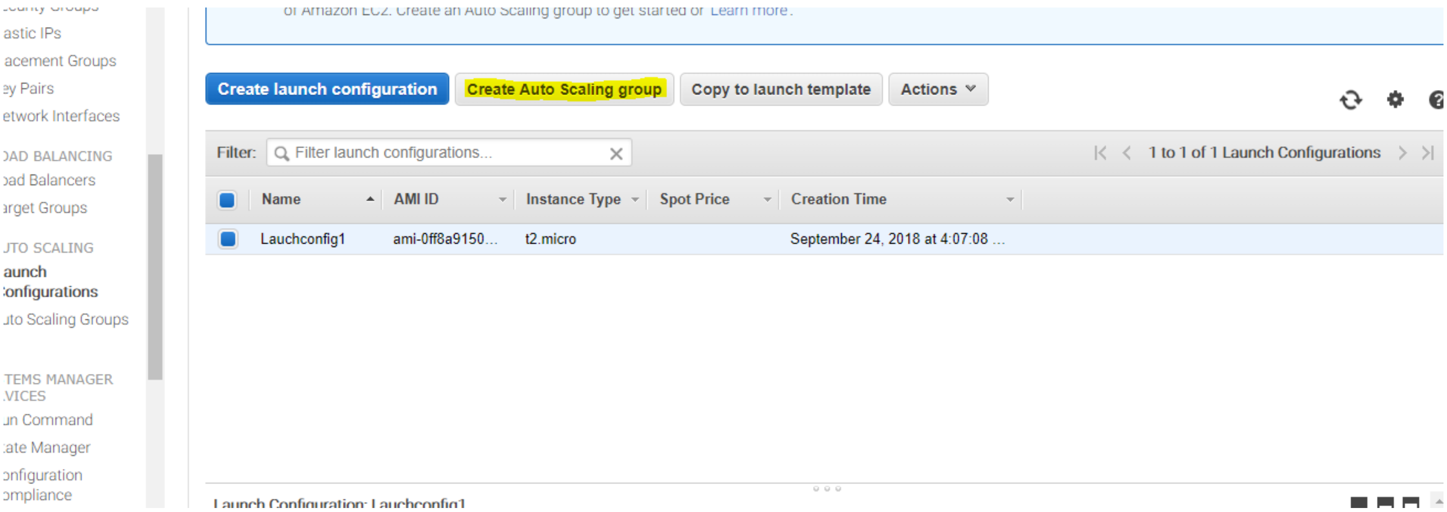
Filter: Filter launch configurations...

Name	AMI ID	Instance Type	Spot Price	Creation Time

No launch configurations found

Select a launch configuration above

39. Configure the instance in the same way as you would do the normal instance.
40. Once launch configuration is created go to autoscaling groups and create one.

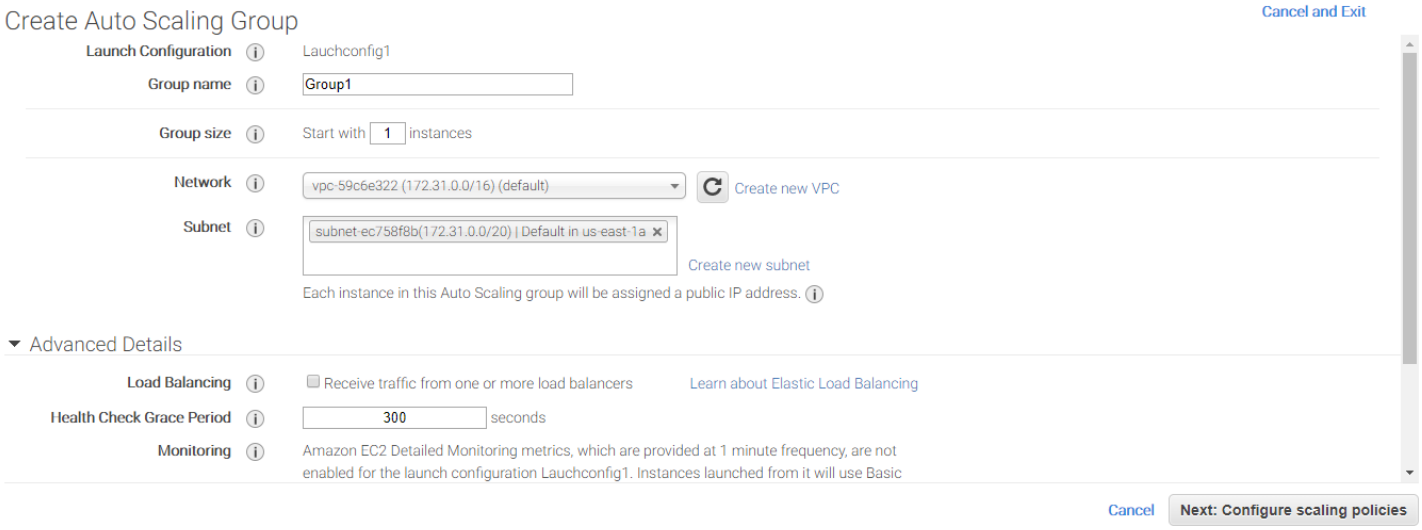


of Amazon EC2. Create an Auto Scaling group to get started or Learn more.

Filter: Filter launch configurations...

Name	AMI ID	Instance Type	Spot Price	Creation Time
Lauchconfig1	ami-0ff8a9150...	t2.micro		September 24, 2018 at 4:07:08 ...

Launch Configuration: Lauchconfig1


Cancel and Exit
Create new VPC
Create new subnet
Learn about Elastic Load Balancing
Cancel
Next: Configure scaling policies

Create Auto Scaling Group

Launch Configuration: Lauchconfig1

Group name: Group1

Group size: Start with 1 instances

Network: vpc-59c6e322 (172.31.0.0/16) (default)

Subnet: subnet-ec758f8b(172.31.0.0/20) | Default in us-east-1a

Each instance in this Auto Scaling group will be assigned a public IP address.

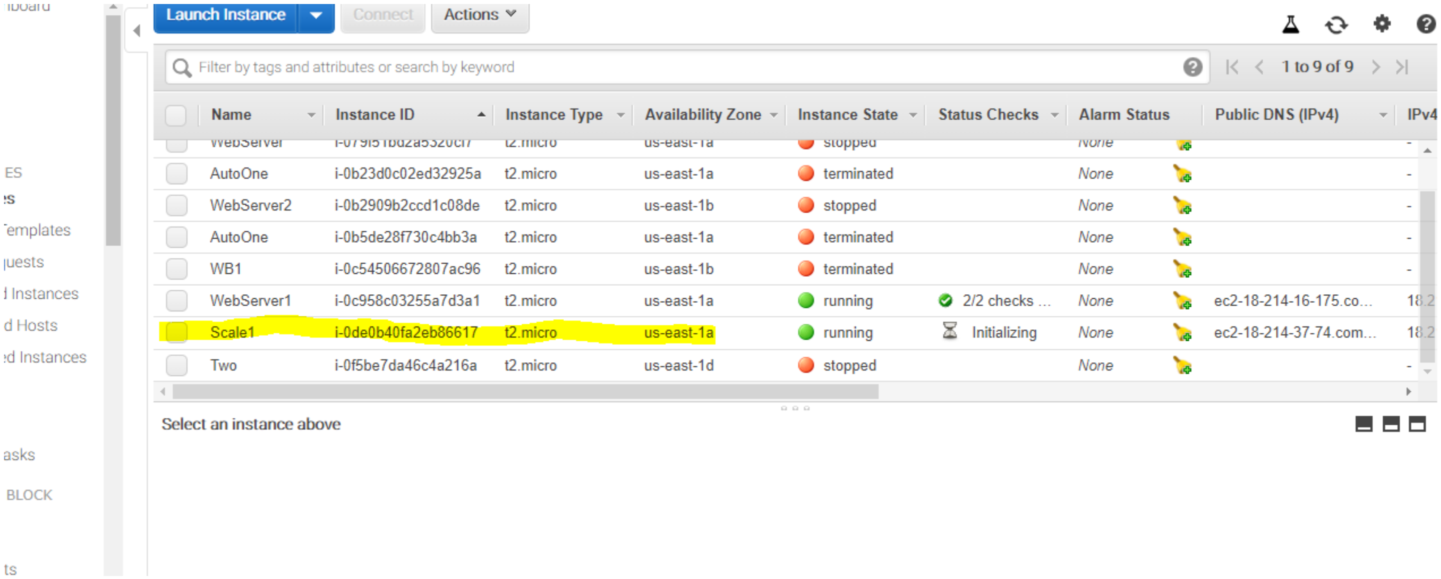
Advanced Details

Load Balancing: □ Receive traffic from one or more load balancers

Health Check Grace Period: 300 seconds

Monitoring: Amazon EC2 Detailed Monitoring metrics, which are provided at 1 minute frequency, are not enabled for the launch configuration Lauchconfig1. Instances launched from it will use Basic

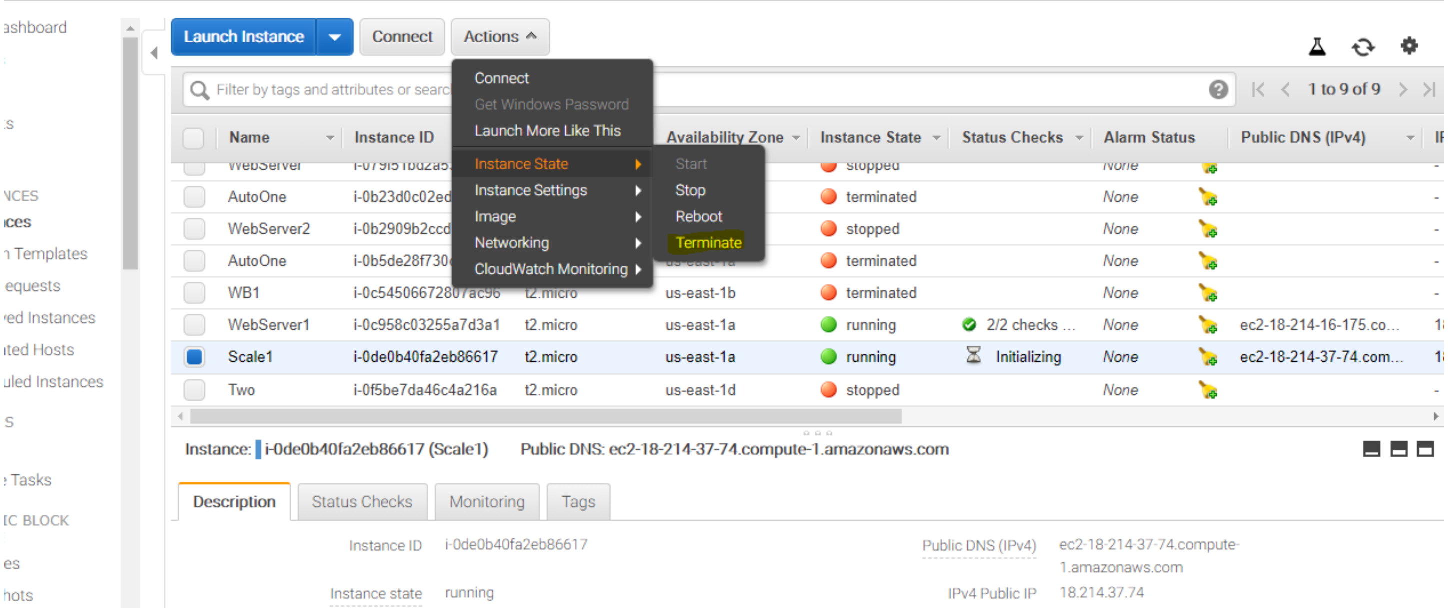
41. Click next several times and give the tag to your auto scaling group.
42. Once created, go to instance tab and you will see the autoscaling instance in your instance tab.



Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4
WebServer	i-079f51bd2a5320cf7	t2.micro	us-east-1a	stopped		None		-
AutoOne	i-0b23d0c02ed32925a	t2.micro	us-east-1a	terminated		None		-
WebServer2	i-0b2909b2ccd1c08de	t2.micro	us-east-1b	stopped		None		-
AutoOne	i-0b5de28f730c4bb3a	t2.micro	us-east-1a	terminated		None		-
WB1	i-0c54506672807ac96	t2.micro	us-east-1b	terminated		None		-
WebServer1	i-0c958c03255a7d3a1	t2.micro	us-east-1a	running	2/2 checks ...	None	ec2-18-214-16-175.co...	18.2
Scale1	i-0de0b40fa2eb86617	t2.micro	us-east-1a	running	Initializing	None	ec2-18-214-37-74.com...	18.2
Two	i-0f5be7da46c4a216a	t2.micro	us-east-1d	stopped		None		-

Select an instance above

43. In order to test auto scaling delete this instance and wait for 5 min and the auto scaling group will provision the instance on your own within 5 min.

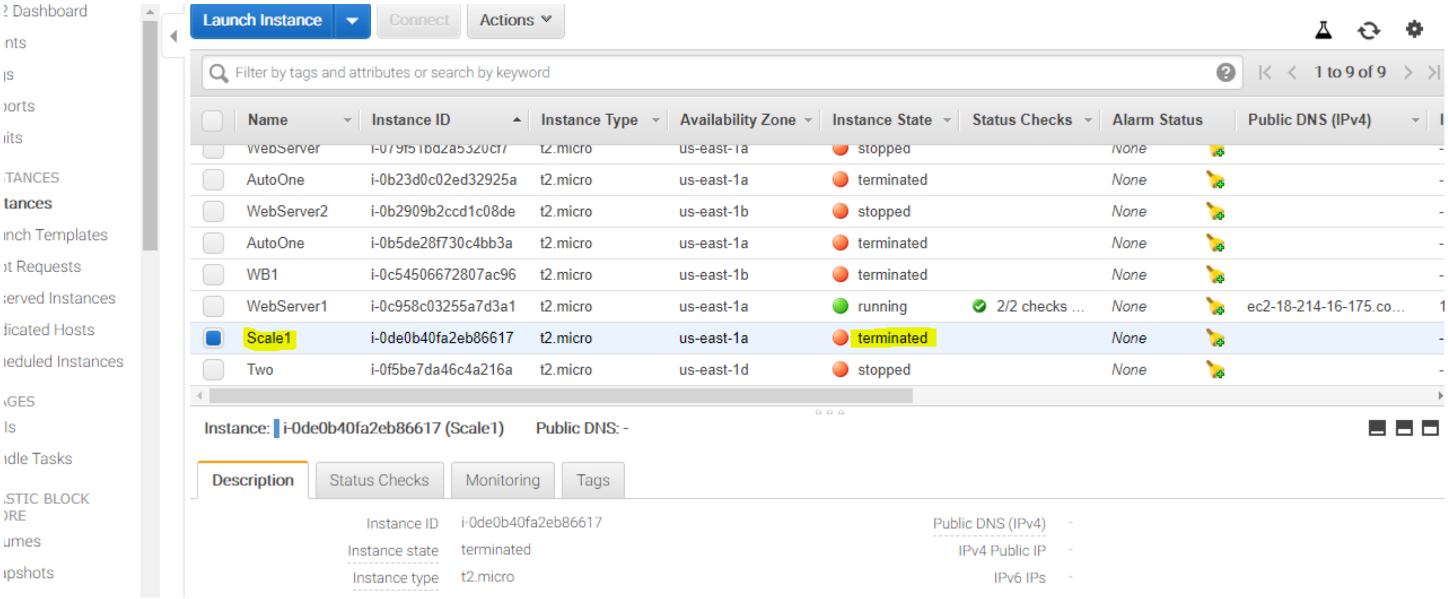


Name	Instance ID		Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IF
WebServer	i-079f51bd2a5			stopped		None		-
AutoOne	i-0b23d0c02ed			terminated		None		-
WebServer2	i-0b2909b2ccd			stopped		None		-
AutoOne	i-0b5de28f730		us-east-1a	terminated		None		-
WB1	i-0c54506672807ac96	t2.micro	us-east-1b	terminated		None		-
WebServer1	i-0c958c03255a7d3a1	t2.micro	us-east-1a	running	2/2 checks ...	None	ec2-18-214-16-175.co...	1
Scale1	i-0de0b40fa2eb86617	t2.micro	us-east-1a	running	Initializing	None	ec2-18-214-37-74.com...	1
Two	i-0f5be7da46c4a216a	t2.micro	us-east-1d	stopped		None		-

Instance: i-0de0b40fa2eb86617 (Scale1) Public DNS: ec2-18-214-37-74.compute-1.amazonaws.com

Description | Status Checks | Monitoring | Tags

Instance ID	i-0de0b40fa2eb86617	Public DNS (IPv4)	ec2-18-214-37-74.compute-1.amazonaws.com
Instance state	running	IPv4 Public IP	18.214.37.74



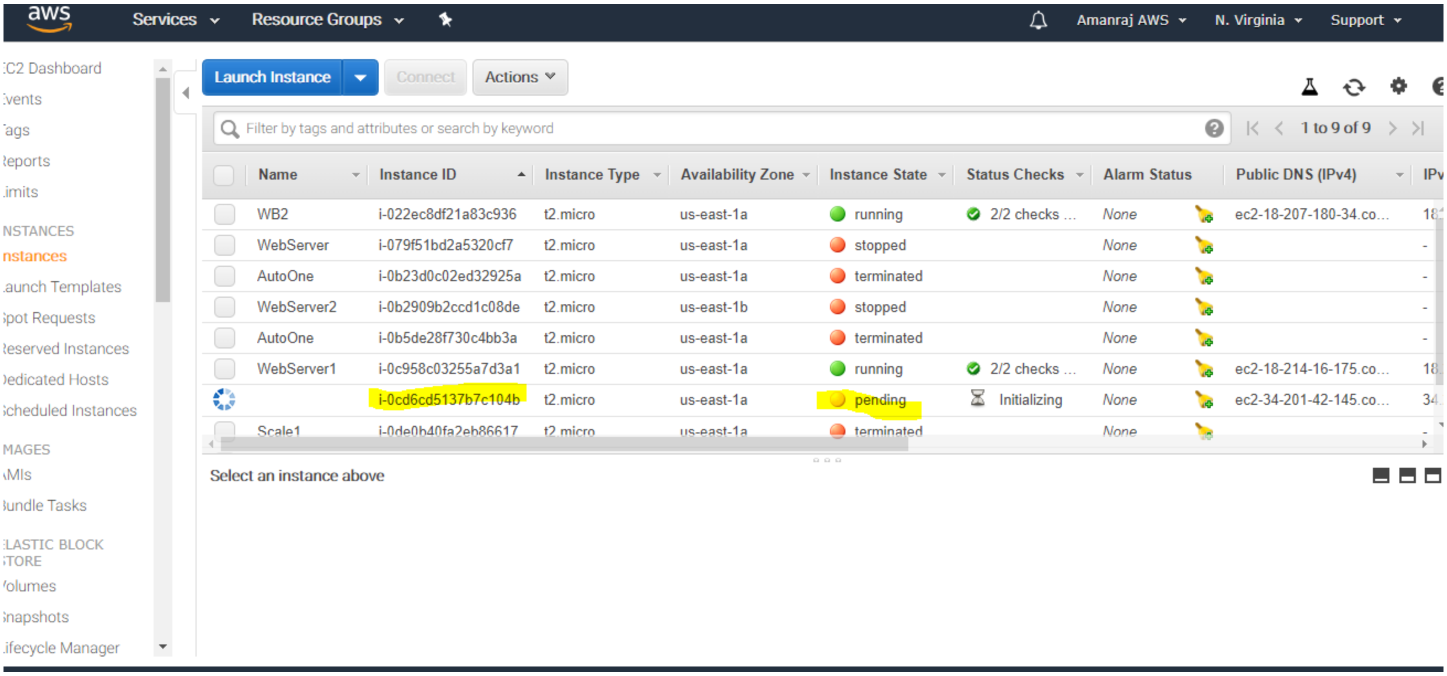
	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	I
□	WebServer	i-079f51bd2a5320cf7	t2.micro	us-east-1a	🟥 stopped		None		-
□	AutoOne	i-0b23d0c02ed32925a	t2.micro	us-east-1a	🟥 terminated		None		-
□	WebServer2	i-0b2909b2ccd1c08de	t2.micro	us-east-1b	🟥 stopped		None		-
□	AutoOne	i-0b5de28f730c4bb3a	t2.micro	us-east-1a	🟥 terminated		None		-
□	WB1	i-0c54506672807ac96	t2.micro	us-east-1b	🟥 terminated		None		-
□	WebServer1	i-0c958c03255a7d3a1	t2.micro	us-east-1a	🟩 running	✓ 2/2 checks ...	None	ec2-18-214-16-175.co...	1
☑	Scale1	i-0de0b40fa2eb86617	t2.micro	us-east-1a	🟥 terminated		None		-
□	Two	i-0f5be7da46c4a216a	t2.micro	us-east-1d	🟥 stopped		None		-

Instance: i-0de0b40fa2eb86617 (Scale1) Public DNS: -

Instance ID i-0de0b40fa2eb86617
Instance state terminated
Instance type t2.micro

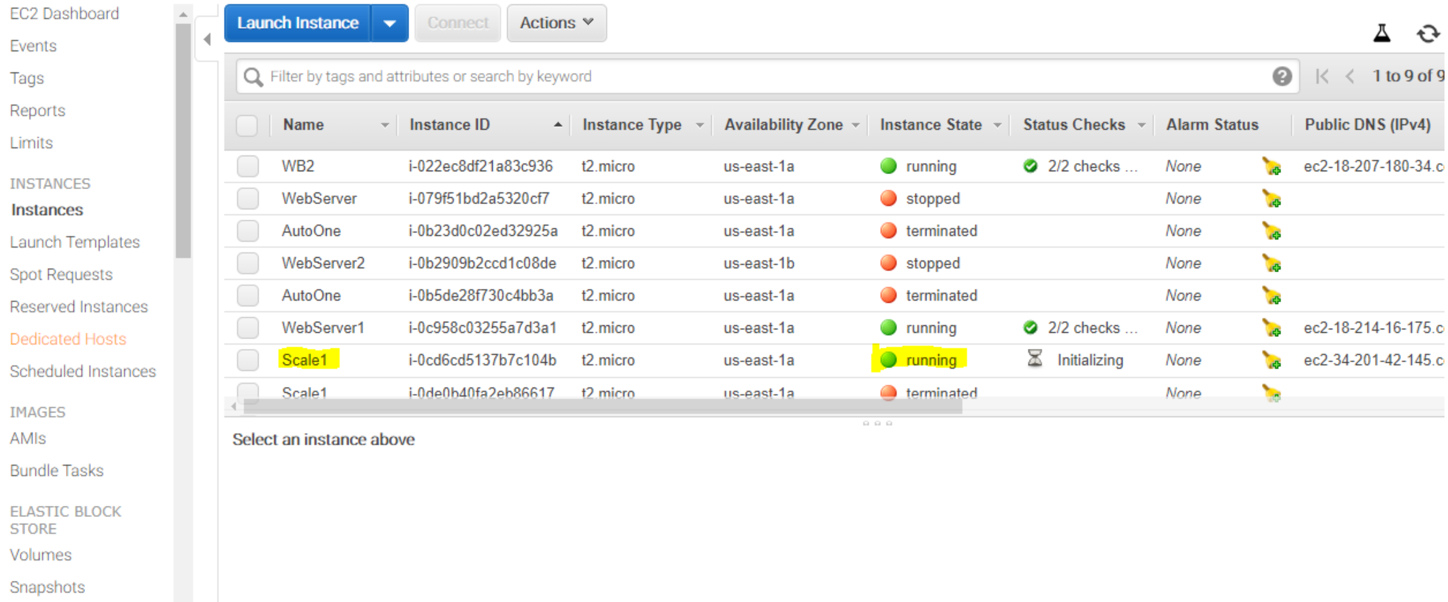
Public DNS (IPv4) -
IPv4 Public IP -
IPv6 IPs -

44. Provisioned automatically within 5 min



	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv
□	WB2	i-022ec8df21a83c936	t2.micro	us-east-1a	🟩 running	✓ 2/2 checks ...	None	ec2-18-207-180-34.co...	18
□	WebServer	i-079f51bd2a5320cf7	t2.micro	us-east-1a	🟥 stopped		None		-
□	AutoOne	i-0b23d0c02ed32925a	t2.micro	us-east-1a	🟥 terminated		None		-
□	WebServer2	i-0b2909b2ccd1c08de	t2.micro	us-east-1b	🟥 stopped		None		-
□	AutoOne	i-0b5de28f730c4bb3a	t2.micro	us-east-1a	🟥 terminated		None		-
□	WebServer1	i-0c958c03255a7d3a1	t2.micro	us-east-1a	🟩 running	✓ 2/2 checks ...	None	ec2-18-214-16-175.co...	18
		i-0cd6cd5137b7c104b	t2.micro	us-east-1a	🟨 pending	⌛ Initializing	None	ec2-34-201-42-145.co...	34
□	Scale1	i-0de0b40fa2eb86617	t2.micro	us-east-1a	🟥 terminated		None		-

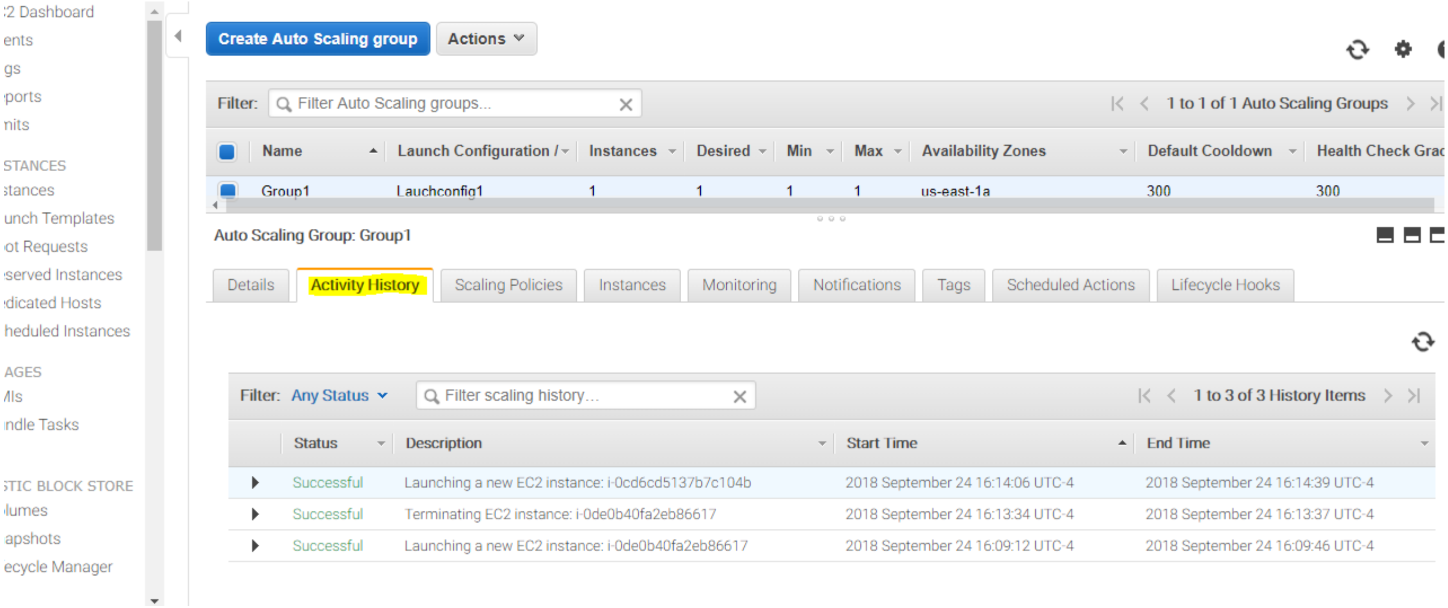
Select an instance above



Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)
WB2	i-022ec8df21a83c936	t2.micro	us-east-1a	running	2/2 checks ...	None	ec2-18-207-180-34.c
WebServer	i-079f51bd2a5320cf7	t2.micro	us-east-1a	stopped		None	
AutoOne	i-0b23d0c02ed32925a	t2.micro	us-east-1a	terminated		None	
WebServer2	i-0b2909b2ccd1c08de	t2.micro	us-east-1b	stopped		None	
AutoOne	i-0b5de28f730c4bb3a	t2.micro	us-east-1a	terminated		None	
WebServer1	i-0c958c03255a7d3a1	t2.micro	us-east-1a	running	2/2 checks ...	None	ec2-18-214-16-175.c
Scale1	i-0cd6cd5137b7c104b	t2.micro	us-east-1a	running	Initializing	None	ec2-34-201-42-145.c
Scale1	i-0de0b40fa2eb86617	t2.micro	us-east-1a	terminated		None	

Select an instance above

45. If you go to the autoscaling group tab, you can see the history also.



Name	Launch Configuration /	Instances	Desired	Min	Max	Availability Zones	Default Cooldown	Health Check Grac
Group1	Lauchconfig1	1	1	1	1	us-east-1a	300	300

Auto Scaling Group: Group1

Status	Description	Start Time	End Time
Successful	Launching a new EC2 instance: i-0cd6cd5137b7c104b	2018 September 24 16:14:06 UTC-4	2018 September 24 16:14:39 UTC-4
Successful	Terminating EC2 instance: i-0de0b40fa2eb86617	2018 September 24 16:13:34 UTC-4	2018 September 24 16:13:37 UTC-4
Successful	Launching a new EC2 instance: i-0de0b40fa2eb86617	2018 September 24 16:09:12 UTC-4	2018 September 24 16:09:46 UTC-4