

## Assignment no #2

### 1. Create an EC2 instance

Instances (1) <a href="#">Info</a>		<a href="#">C</a>	Connect	Instance state ▾	Actions ▾	<a href="#">Launch instances</a>	▼
<input type="text"/> Filter instances							
<input type="text"/> search: i-01ddf37d7379ea999 <a href="#">X</a> <a href="#">Clear filters</a>							
□	Name ▾	Instance ID	Instance state ▾	Instance type ▾	Status check	Alarm status	▼
<input type="checkbox"/>	FirstInst	i-01ddf37d7379ea999	<span>Running</span> <a href="#">Q</a> <a href="#">Q</a>	t2.micro	<span>2/2 checks passed</span>	No alarms	-

### 2. Connect to an instance and run system update.

```
Verifying : 1:grub2-2.06-2.amzn2.0.1.x86_64
Verifying : systemtap-runtime-4.4-1.amzn2.0.1.x86_64
Verifying : 1:grub2-tools-minimal-2.06-2.amzn2.0.1.x86_64
Verifying : curl-7.61.1-12.amzn2.0.4.x86_64
Verifying : 1:grub2-pc-2.06-2.amzn2.0.1.x86_64
Verifying : kernel-tools-4.14.238-182.422.amzn2.x86_64
Verifying : libcurl-7.61.1-12.amzn2.0.4.x86_64
Verifying : 1:grub2-pc-modules-2.06-2.amzn2.0.1.noarch
Verifying : grubby-8.28-23.amzn2.0.1.x86_64

Installed:
  grub2.x86_64 1:2.06-2.amzn2.0.3
  grub2-tools.x86_64 1:2.06-2.amzn2.0.3
  grub2-tools-extra.x86_64 1:2.06-2.amzn2.0.3
  kernel.x86_64 0:4.14.243-185.433.amzn2

Updated:
  curl.x86_64 0:7.76.1-4.amzn2.0.1
  grub2-common.noarch 1:2.06-2.amzn2.0.3
  grub2-pc-modules.noarch 1:2.06-2.amzn2.0.3
  kernel-tools.x86_64 0:4.14.243-185.433.amzn2
  systemtap-runtime.x86_64 0:4.4-1.amzn2.0.2

Replaced:
  grub2.x86_64 1:2.06-2.amzn2.0.1
  grub2-tools.x86_64 1:2.06-2.amzn2.0.1

Complete!
[ec2-user@ip-172-31-36-248 ~]$
```

### 3. Enable termination protection and test.

<span>⌚ Failed to terminate an instance: The instance 'i-01ddf37d7379ea999' may not be terminated. Modify its 'disableApiTermination' instance attribute and try again.</span>	X						
<hr/>							
Instances (1/1) <a href="#">Info</a>		<a href="#">C</a>	Connect	Instance state ▾	Actions ▾	<a href="#">Launch instances</a>	▼
<input type="text"/> Filter instances							
✓	Name ▾	Instance ID	Instance state ▾	Instance type ▾	Status check	Alarm status	▼
<input checked="" type="checkbox"/>	FirstInst	i-01ddf37d7379ea999	<span>Running</span> <a href="#">Q</a> <a href="#">Q</a>	t2.micro	<span>2/2 checks passed</span>	No alarms	-

### 4. Disable termination protection and terminate the instance.

<span>⌚ Disabled termination protection for i-01ddf37d7379ea999</span>	X						
<span>⌚ Successfully terminated i-01ddf37d7379ea999</span>	X						
<hr/>							
Instances (1/1) <a href="#">Info</a>		<a href="#">C</a>	Connect	Instance state ▾	Actions ▾	<a href="#">Launch instances</a>	▼
<input type="text"/> Filter instances							
✓	Name ▾	Instance ID	Instance state ▾	Instance type ▾	Status check	Alarm status	▼
<input checked="" type="checkbox"/>	FirstInst	i-01ddf37d7379ea999	<span>Shutting-down</span> <a href="#">Q</a> <a href="#">Q</a>	t2.micro	<span>2/2 checks passed</span>	No alarms	-

### Assignment no #3

#### 1. Create a VPC

✓ You successfully created vpc-01b977220f16dc184 / myvpc X

VPC > Your VPCs > vpc-01b977220f16dc184

### vpc-01b977220f16dc184 / myvpc

Actions ▾

Details		Info	
VPC ID vpc-01b977220f16dc184	State <span style="color: green;">Available</span>	DNS hostnames Disabled	DNS resolution Enabled
Tenancy Default	DHCP options set dopt-14b5f67f	Main route table rtb-0afaf8c5ed2a90999	Main network ACL acl-054485a259429fb27

#### 2. Create a internet gateway and attach to VPC

✓ The following internet gateway was created: igw-0226c491e9df0de38 . You can now attach to a VPC to enable the VPC to communicate with the internet. Attach to a VPC X

✓ Internet gateway igw-0226c491e9df0de38 successfully attached to vpc-01b977220f16dc184 X

VPC > Internet gateways > igw-0226c491e9df0de38

### igw-0226c491e9df0de38 / myigw

Actions ▾

Details		Info	
Internet gateway ID igw-0226c491e9df0de38	State <span style="color: green;">Attached</span>	VPC ID vpc-01b977220f16dc184   myvpc	Owner 847145281302

#### 3. Create a route table and add route to igw.

✓ Route table rtb-0e2f6d9ae5938d018 | myroutetable1 was created successfully.

✓ Updated routes for rtb-0e2f6d9ae5938d018 / myroutetable1 successfully Details

VPC > Route tables > rtb-0e2f6d9ae5938d018

### rtb-0e2f6d9ae5938d018 / myroutetable1

Actions ▾

You can now check network connectivity with Reachability Analyzer Run Reachability Analyzer X

Details		Info	
Route table ID rtb-0e2f6d9ae5938d018	Main <span style="color: green;">Yes</span>	Explicit subnet associations -	Edge associations -

**4. Make custom route table the main route table.**

You successfully set the route table rtb-0e2f6d9ae5938d018 / myroutetable1 as main.

VPC > Route tables > rtb-0e2f6d9ae5938d018

## rtb-0e2f6d9ae5938d018 / myroutetable1

Actions ▾

You can now check network connectivity with Reachability Analyzer

Run Reachability Analyzer X

Details		Info	
Route table ID rtb-0e2f6d9ae5938d018	Main Yes	Explicit subnet associations -	Edge associations -

**5. Create the subnet.**

You have successfully created 1 subnet: subnet-0f8e502aec5bbaaad

Subnets (1/1) Info Actions ▾ Create subnet

Filter subnets

Subnet ID: subnet-0f8e502aec5bbaaad X Clear filters

Name	Subnet ID	State	VPC	IPv4
mysubnet1	subnet-0f8e502aec5bbaaad	Available	vpc-01b977220f16dc184   my...	40.0

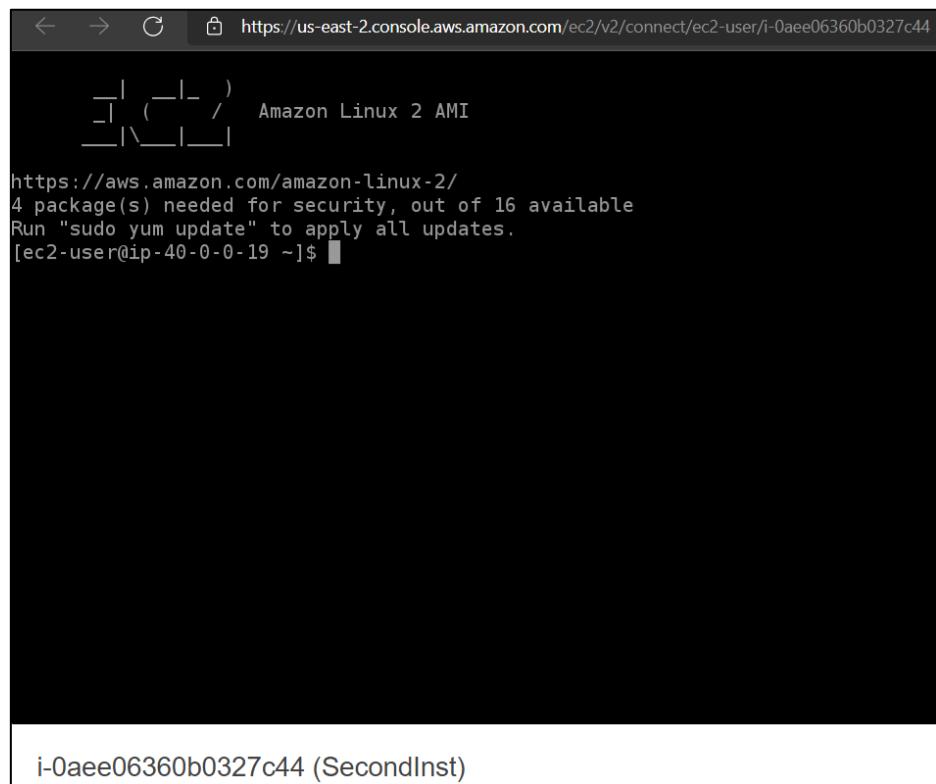
**6. Modify auto assign IP settings for the subnet.**

## subnet-0f8e502aec5bbaaad / mysubnet1

Details

Subnet ID subnet-0f8e502aec5bbaaad	Subnet ARN arn:aws:ec2:us-east-2:847145281302:subnet/subnet-0f8e502aec5bbaaad	State Available
Available IPv4 addresses 251	IPv6 CIDR -	Availability Zone us-east-2a
VPC vpc-01b977220f16dc184   myvpc	Route table rtb-0e2f6d9ae5938d018   myroutetable1	Network ACL acl-054485a259429fb27
Auto-assign public IPv4 address Yes	Auto-assign IPv6 address No	Auto-assign customer-owned IPv4 address No
		IPv6 CIDR reservations

**7. Launch an EC2 instance in custom VPC.**



A screenshot of a terminal window from the AWS CloudWatch Metrics Insights interface. The URL in the address bar is <https://us-east-2.console.aws.amazon.com/ec2/v2/connect/ec2-user/i-0aee06360b0327c44>. The terminal shows the following output:

```
Amazon Linux 2 AMI
https://aws.amazon.com/amazon-linux-2/
4 package(s) needed for security, out of 16 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-40-0-0-19 ~]$
```

The terminal window has a black background and white text. The bottom right corner of the terminal window is highlighted with a white border.

i-0aee06360b0327c44 (SecondInst)

## Assignment no #4

### 1. Create an EBS application.

The screenshot shows the 'All applications' list in the AWS Elastic Beanstalk console. The table has columns for Application name, Environments, Date created, Last modified, and ARN. One entry is visible:

Application name	Environments	Date created	Last modified	ARN
myapp1	Myapp1-env	2021-08-30 19:00:14 UTC+0530	2021-08-30 19:00:14 UTC+0530	arn:aws:elasticbeanstalk:us-east-2:847145281302:application/myapp1

### 2. Create web server environment.

The screenshot shows the 'All environments' list in the AWS Elastic Beanstalk console. The table has columns for Environment name, Health, Application name, Date created, Last modified, and URL. One entry is visible:

Environment name	Health	Application name	Date created	Last modified	URL
Myapp1-env	Ok	myapp1	2021-08-30 19:01:57 UTC+0530	2021-08-30 19:05:22 UTC+0530	Myapp1-env.eba-b97mpimr.us-east-2.elasticbeanstalk.com

### 3. Deploying a sample application.

The screenshot shows a browser window displaying the deployment confirmation for the AWS Elastic Beanstalk Python application. The main content area says 'Congratulations' and provides the URL 'myapp1-env.eba-b97mpimr.us-east-2.elasticbeanstalk.com'. The status bar at the top indicates 'Not secure'. To the right, there is a sidebar titled 'What's Next?' with a list of links:

- [AWS Elastic Beanstalk overview](#)
- [AWS Elastic Beanstalk concepts](#)
- [Deploy a Django Application to AWS Elastic Beanstalk](#)
- [Deploy a Flask Application to AWS Elastic Beanstalk](#)
- [Customizing and Configuring a Python Container](#)
- [Working with Logs](#)