



### Assignment no #3

#### 1. Create a VPC

✓ You successfully created vpc-01b977220f16dc184 / myvpc

VPC > Your VPCs > vpc-01b977220f16dc184

## vpc-01b977220f16dc184 / myvpc

Actions ▾

**Details** Info

VPC ID	State	DNS hostnames	DNS resolution
📄 vpc-01b977220f16dc184	✓ Available	Disabled	Enabled
Tenancy	DHCP options set	Main route table	Main network ACL
Default	dopt-14b5f67f	rtb-0afaf8c5ed2a90999	acl-054485a259429fb27

#### 2. Create an 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 ✕

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

VPC > Internet gateways > igw-0226c491e9df0de38

## igw-0226c491e9df0de38 / myigw

Actions ▾

**Details** Info

Internet gateway ID	State	VPC ID	Owner
📄 igw-0226c491e9df0de38	✓ Attached	vpc-01b977220f16dc184   myvpc	📄 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 ✕

**Details** Info

Route table ID	Main	Explicit subnet associations	Edge associations
📄 rtb-0e2f6d9ae5938d018	Yes	—	—

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 ▾

ℹ You can now check network connectivity with Reachability Analyzer Run Reachability Analyzer ✕

**Details** Info

Route table ID	Main	Explicit subnet associations	Edge associations
📄 rtb-0e2f6d9ae5938d018	📄 Yes	–	–

5. Create the subnet.

✔ You have successfully created 1 subnet: subnet-0f8e502aec5bbaaad ✕

**Subnets (1/1)** Info 🔄 Actions ▾ Create subnet

🔍 Filter subnets < 1 > ⚙

Subnet ID: subnet-0f8e502aec5bbaaad ✕ Clear filters

<input checked="" type="checkbox"/>	Name ▾	Subnet ID ▾	State ▾	VPC ▾	IPv4
<input checked="" type="checkbox"/>	mysubnet1	subnet-0f8e502aec5bbaaad	✔ Available	vpc-01b977220f16dc184   my...	40.0

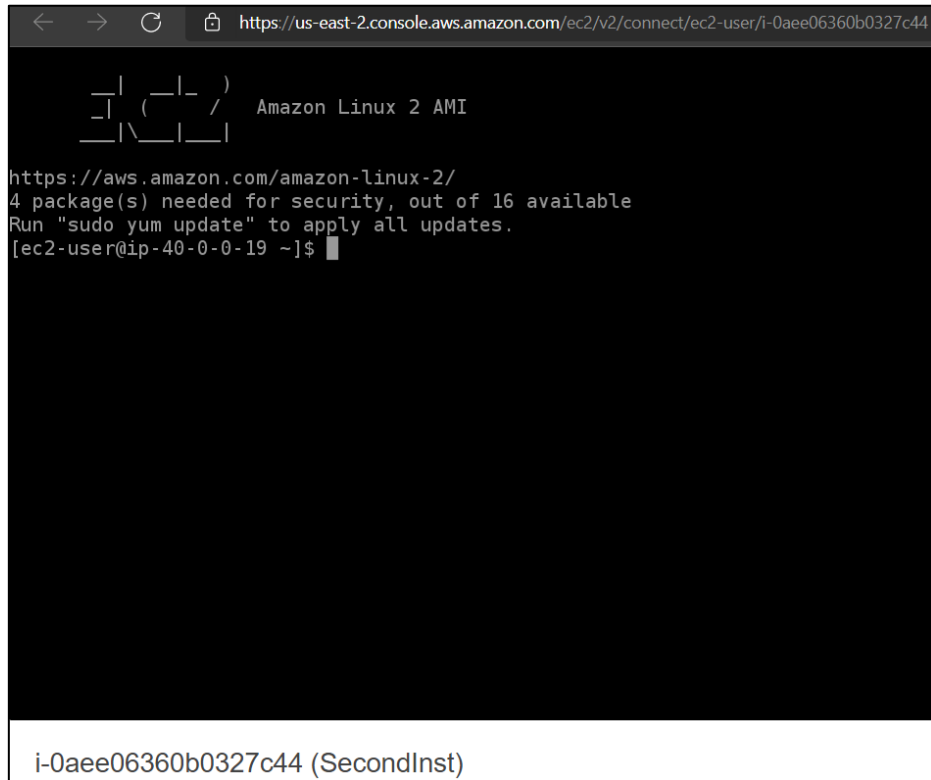
6. Modify auto assign IP settings for the subnet.

## subnet-0f8e502aec5bbaaad / mysubnet1

**Details**

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

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



The screenshot shows a terminal window from the AWS Management Console. The browser address bar at the top displays the URL: `https://us-east-2.console.aws.amazon.com/ec2/v2/connect/ec2-user/i-0aee06360b0327c44`. The terminal output shows the Amazon Linux 2 AMI boot sequence, including the ASCII art logo and the message: "4 package(s) needed for security, out of 16 available. Run 'sudo yum update' to apply all updates." The prompt is `[ec2-user@ip-40-0-0-19 ~]$`. At the bottom of the terminal window, the instance ID `i-0aee06360b0327c44` is displayed with the label `(SecondInst)`.

```

  _ _ | _ _ | _ _ )
  _ | ( _ | /      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 ~]$ █

i-0aee06360b0327c44 (SecondInst)
```

## Assignment no #4

### 1. Create an EBS application.

Elastic Beanstalk > Applications

### All applications

Filter results matching the display values

	Application name ▲	Environments ▼	Date created ▼	Last modified ▼	ARN
<input type="radio"/>	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.

Elastic Beanstalk > Environments

### All environments

Filter results matching the display values

	Environment name ▲	Health ▼	Application name ▼	Date created ▼	Last modified ▼	URL
<input type="radio"/>	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.

← → ↻ ⚠ Not secure | myapp1-env.eba-b97mpimr.us-east-2.elasticbeanstalk.com

# Congratulations

Your first AWS Elastic Beanstalk Python Application is now running on your own dedicated environment in the AWS Cloud

This environment is launched with Elastic Beanstalk Python Platform

### What's Next?

- [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](#)