

AWS

Tutorial

- 1) What is AWS?
- 2) Creating an Amazon EC2 instance
- 3) Modifying the Amazon EC2 instance parameters
- 4) What is AWS Lambda?
- 5) AWS Certification Guide
- 6) Azure vs. AWS
- 7) Heroku Vs AWS
- 8) Digitalocean Vs AWS
- 9) Google Cloud Vs AWS
- 10) AWS Competitors & Alternatives

What is AWS?

Amazon Cloud

Services Tutorial

<https://www.guru99.com/what-is-aws.html>

What is Cloud Computing?

Cloud computing is a term referred to storing and accessing data over the internet. It doesn't store any data on the hard disk of your personal computer. In cloud computing, you can access data from a remote server.

What is AWS?

Amazon web service is a platform that offers flexible, reliable, scalable, easy-to-use and cost-effective cloud computing solutions.

AWS is a comprehensive, easy to use computing platform offered Amazon. The platform is developed with a combination of infrastructure as a service (IaaS), platform as a service (PaaS) and packaged software as a service (SaaS) offerings.

In this tutorial, you will learn,

- [What is Cloud Computing?](#)
- [What is AWS?](#)
- [History of AWS](#)
- [Important AWS Services](#)
- [Applications of AWS services](#)
- [Companies using AWS](#)
- [Advantages of AWS](#)
- [Disadvantages of AWS](#)

- Best practices of AWS

History of AWS

- 2002- AWS services launched
- 2006- Launched its cloud products
- 2012- Holds first customer event
- 2015- Reveals revenues achieved of \$4.6 billion
- 2016- Surpassed \$10 billion revenue target
- 2016- Release snowball and snowmobile
- 2019- Offers nearly 100 cloud services

Important AWS Services

Amazon Web Services offers a wide range of different business purpose global cloud-based products. The products include storage, databases, analytics, networking, mobile, development tools, enterprise applications, with a pay-as-you-go pricing model.



Important AWS Services

Here, are essential AWS services.

AWS Compute Services

Here, are Cloud Compute Services offered by Amazon:

1. **EC2(Elastic Compute Cloud)** - EC2 is a virtual machine in the cloud on which you have OS level control. You can run this cloud server whenever you want.

2. **LightSail**-This cloud computing tool automatically deploys and manages the computer, storage, and networking capabilities required to run your applications.
3. **Elastic Beanstalk**— The tool offers automated deployment and provisioning of resources like a highly scalable production website.
4. **EKS (Elastic Container Service for Kubernetes)**—The tool allows you to Kubernetes on Amazon cloud environment without installation.
5. **AWS Lambda**—This AWS service allows you to run functions in the cloud. The tool is a big cost saver for you as you to pay only when your functions execute.

Migration

Migration services used to transfer data physically between your datacenter and AWS.

1. **DMS (Database Migration Service)**-DMS service can be used to migrate on-site databases to AWS. It helps you to migrate from one type of database to another — for example, Oracle to MySQL.
2. **SMS (Server Migration Service)**-SMS migration services allows you to migrate on-site servers to AWS easily and quickly.
3. **Snowball**—Snowball is a small application which allows you to transfer terabytes of data inside and outside of AWS environment.

Storage

1. **Amazon Glacier**- It is an extremely low-cost storage service. It offers secure and fast storage for data archiving and backup.
2. **Amazon Elastic Block Store (EBS)**- It provides block-level storage to use with Amazon EC2 instances. Amazon Elastic Block Store volumes are network-attached and remain independent from the life of an instance.
3. **AWS Storage Gateway**- This AWS service is connecting on-premises software applications with cloud-based storage. It offers secure integration between the company's on-premises and AWS's storage infrastructure.

Security Services

1. **IAM (Identity and Access Management)**— IAM is a secure cloud security service which helps you to manage users, assign policies, form groups to manage multiple users.
2. **Inspector**—It is an agent that you can install on your virtual machines, which reports any security vulnerabilities.
3. **Certificate Manager**—The service offers free SSL certificates for your domains that are managed by Route53.
4. **WAF (Web Application Firewall)**— WAF security service offers application-level protection and allows you to block SQL injection and helps you to block cross-site scripting attacks.
5. **Cloud Directory**—This service allows you to create flexible, cloud-native directories for managing hierarchies of data along multiple dimensions.
6. **KMS (Key Management Service)**—It is a managed service. This security service helps you to create and control the encryption keys which allows you to encrypt your data.
7. **Organizations**—You can create groups of AWS accounts using this service to manage security and automation settings.
8. **Shield**—Shield is managed DDoS (Distributed Denial of Service protection service). It offers safeguards against web applications running on AWS.
9. **Macie**—It offers a data visibility security service which helps classify and protect your sensitive critical content.
10. **GuardDuty** —It offers threat detection to protect your AWS accounts and workloads.

Database Services

1. **Amazon RDS**- This Database AWS service is easy to set up, operate, and scale a relational database in the cloud.
2. **Amazon DynamoDB**- It is a fast, fully managed NoSQL database service. It is a simple service which allow cost-effective storage and retrieval of data. It also allows you to serve any level of request traffic.
3. **Amazon ElastiCache**- It is a web service which makes it easy to deploy, operate, and scale an in-memory cache in the cloud.
4. **Neptune**- It is a fast, reliable and scalable **graph database** service.
5. **Amazon RedShift**-It is Amazon's data warehousing solution which you can use to perform complex OLAP queries.

Analytics

1. **Athena**—This analytics service allows you to run SQL queries on your S3 bucket to find files.
2. **CloudSearch**—You should use this AWS service to create a fully managed search engine for your website.
3. **ElasticSearch**—It is similar to CloudSearch. However, it offers more features like application monitoring.
4. **Kinesis**—This AWS analytics service helps you to stream and analyze real-time data at massive scale.
5. **QuickSight**—It is a business analytics tool. It helps you to create visualizations in a dashboard for data in Amazon Web Services. For example, S3, DynamoDB, etc.
6. **EMR (Elastic Map Reduce)**—This AWS analytics service mainly used for big data processing like Spark, Splunk, Hadoop, etc.
7. **Data Pipeline**—Allows you to move data from one place to another. For example from DynamoDB to S3.

Management Services

1. **CloudWatch**—Cloud watch helps you to monitor AWS environments like EC2, RDS instances, and CPU utilization. It also triggers alarms based on various metrics.
2. **CloudFormation**—It is a way of turning infrastructure into the cloud. You can use templates for providing a whole production environment in minutes.
3. **CloudTrail**—It offers an easy method of auditing AWS resources. It helps you to log all changes.
4. **OpsWorks**—The service allows you to automate Chef/Puppet deployments on AWS environment.
5. **Config**—This AWS service monitors your environment. The tool sends alerts about changes when you break certain defined configurations.
6. **Service Catalog**—This service helps large enterprises to authorize which services will be used and which won't.
7. **AWS Auto Scaling**—The service allows you to automatically scale your resources up and down based on given CloudWatch metrics.
8. **Systems Manager**—This AWS service allows you to group your resources. It allows you to identify issues and act on them.

9. **Managed Services**—It offers management of your AWS infrastructure which allows you to focus on your applications.

Internet of Things

1. **IoT Core**— It is a managed cloud AWS service. The service allows connected devices like cars, light bulbs, sensor grids, to securely interact with cloud applications and other devices.
2. **IoT Device Management**—It allows you to manage your IoT devices at any scale.
3. **IoT Analytics**—This AWS IOT service is helpful to perform analysis on data collected by your IoT devices.
4. **Amazon FreeRTOS**—This real-time operating system for microcontrollers helps you to connect IoT devices in the local server or into the cloud.

Application Services

1. **Step Functions**—It is a way of visualizing what's going inside your application and what different microservices it is using.
2. **SWF (Simple Workflow Service)**—The service helps you to coordinate both automated tasks and human-led tasks.
3. **SNS (Simple Notification Service)**—You can use this service to send you notifications in the form of email and SMS based on given AWS services.
4. **SQS (Simple Queue Service)**—Use this AWS service to decouple your applications. It is a pull-based service.
5. **Elastic Transcoder**—This AWS service tool helps you to changes a video's format and resolution to support various devices like tablets, smartphones, and laptops of different resolutions.

Deployment and Management

1. **AWS CloudTrail:** The services records AWS API calls and send backlog files to you.
2. **Amazon CloudWatch:** The tools monitor AWS resources like Amazon EC2 and Amazon RDS DB Instances. It also allows you to monitor custom metrics created by user's applications and services.

3. **AWS CloudHSM:** This AWS service helps you meet corporate, regulatory, and contractual, compliance requirements for maintaining data security by using the Hardware Security Module(HSM) appliances inside the AWS environment.

Developer Tools

1. **CodeStar**—Codestar is a cloud-based service for creating, managing, and working with various software development projects on AWS.
2. **CodeCommit**— It is AWS's version control service which allows you to store your code and other assets privately in the cloud.
3. **CodeBuild**—This Amazon developer service help you to automates the process of building and compiling your code.
4. **CodeDeploy**—It is a way of deploying your code in EC2 instances automatically.
5. **CodePipeline**—It helps you create a deployment pipeline like testing, building, testing, authentication, deployment on development and production environments.
6. **Cloud9**—It is an Integrated Development Environment for writing, running, and debugging code in the cloud.

Mobile Services

1. **Mobile Hub**—Allows you to add, configure and design features for mobile apps.
2. **Cognito**—Allows users to signup using his or her social identity.
3. **Device Farm**—Device farm helps you to improve the quality of apps by quickly testing hundreds of mobile devices.
4. **AWS AppSync** —It is a fully managed GraphQL service that offers real-time data synchronization and offline programming features.

Business Productivity

1. **Alexa for Business**—It empowers your organization with voice, using Alexa. It will help you to Allows you to build custom voice skills for your organization.
2. **Chime**—Can be used for online meeting and video conferencing.
3. **WorkDocs**—Helps to store documents in the cloud
4. **WorkMail**—Allows you to send and receive business emails.

Desktop & App Streaming

1. **WorkSpaces**—Workspace is a VDI (Virtual Desktop Infrastructure). It allows you to use remote desktops in the cloud.
2. **AppStream** —A way of streaming desktop applications to your users in the web browser. For example, using MS Word in Google Chrome.

Artificial Intelligence

1. **Lex**—Lex tool helps you to build chatbots **quickly**.
2. **Polly**— It is AWS's text-to-speech service allows you to create audio versions of your notes.
3. **Rekognition** —It is AWS's face recognition service. This AWS service helps you to recognize faces and object in images and videos.
4. **SageMaker**—Sagemaker allows you to build, train, and deploy machine learning models at any scale.
5. **Transcribe**— It is AWS's speech-to-text service that offers high-quality and affordable transcriptions.
6. **Translate**—It is a very similar tool to Google Translate which allows you to translate text in one language to another.

AR & VR (Augmented Reality & Virtual Reality)

1. **Sumerian**—Sumerian is a set of tool for offering high-quality virtual reality (VR) experiences on the web. The service allows you to create interactive 3D scenes and publish it as a website for users to access.

Customer Engagement

1. **Amazon Connect**—Amazon Connect allows you to create your customer care center in the cloud.
2. **Pinpoint**—Pinpoint helps you to understand your users and engage with them.
3. **SES (Simple Email Service)**—Helps you to send bulk emails to your customers at a relatively cost-effective price.

Game Development

1. **GameLift**- It is a service which is managed by AWS. You can use this service to host dedicated game servers. It allows you to scale seamlessly without taking your game offline.

Applications of AWS services

Amazon Web services are widely used for various computing purposes like:

- Web site hosting
- Application hosting/SaaS hosting
- Media Sharing (Image/ Video)
- Mobile and Social Applications
- Content delivery and Media Distribution
- Storage, backup, and disaster recovery
- Development and test environments
- Academic Computing
- Search Engines
- Social Networking

Companies using AWS

- Instagram
- Zoopla
- Smugmug
- Pinterest
- Netflix
- Dropbox
- Etsy
- Talkbox
- Playfish
- Ftopia

Advantages of AWS

Following are the pros of using AWS services:

- AWS allows organizations to use the already familiar programming models, operating systems, databases, and architectures.

- It is a cost-effective service that allows you to pay only for what you use, without any up-front or long-term commitments.
- You will not require to spend money on running and maintaining data centers.
- Offers fast deployments
- You can easily add or remove capacity.
- You are allowed cloud access quickly with limitless capacity.
- Total Cost of Ownership is very low compared to any private/dedicated servers.
- Offers Centralized Billing and management
- Offers Hybrid Capabilities
- Allows you to deploy your application in multiple regions around the world with just a few clicks

Disadvantages of AWS

- If you need more immediate or intensive assistance, you'll have to opt for paid support packages.
- Amazon Web Services may have some common cloud computing issues when you move to a cloud. For example, downtime, limited control, and backup protection.
- AWS sets default limits on resources which differ from region to region. These resources consist of images, volumes, and snapshots.
- Hardware-level changes happen to your application which may not offer the best performance and usage of your applications.

Best practices of AWS

- You need to design for failure, but nothing will fail.
- It's important to decouple all your components before using AWS services.
- You need to keep dynamic data closer to compute and static data closer to the user.
- It's important to know security and performance tradeoffs.
- Pay for computing capacity by the hourly payment method.
- Make a habit of a one-time payment for each instance you want to reserve and to receive a significant discount on the hourly charge.

How to Create EC2 Instance in AWS: Step by Step Tutorial

<https://www.guru99.com/creating-amazon-ec2-instance.html>

What is Amazon EC2 Instance?

An **EC2 instance** is nothing but a virtual server in Amazon Web services terminology. It stands for **Elastic Compute Cloud**. It is a web service where an AWS subscriber can request and provision a compute server in AWS cloud.

An **on-demand** EC2 instance is an offering from AWS where the subscriber/user can rent the virtual server per hour and use it to deploy his/her own applications.

The instance will be charged per hour with different rates based on the type of the instance chosen. AWS provides multiple instance types for the respective business needs of the user.

Thus, you can rent an instance based on your own CPU and memory requirements and use it as long as you want. You can terminate the instance when it's no more used and save on costs. This is the most striking advantage of an on-demand instance- you can drastically save on your CAPEX.

In this tutorial, you will learn-

- [Login and access to AWS services](#)
- [Choose AMI](#)
- [Choose EC2 Instance Types](#)
- [Configure Instance](#)
- [Add Storage](#)
- [Tag Instance](#)
- [Configure Security Groups](#)
- [Review Instances](#)
- [Create a EIP and connect to your instance](#)
- [What is Spot Instance?](#)
- [Create a Spot Request](#)
 - [Find Instance Types](#)
 - [Configure the Spot instance](#)
 - [Review your Spot instance](#)

Let us see in detail how to launch an on-demand EC2 instance in AWS Cloud.

>Login and access to AWS services

Step 1) In this step,

- Login to your AWS account and go to the AWS Services tab at the top left corner.
- Here, you will see all of the AWS Services categorized as per their area viz. Compute, Storage, Database, etc. For creating an EC2 instance, we have to choose Computeà EC2 as in the next step.

The screenshot shows the AWS Services menu. A red box highlights the 'Compute' section under 'All AWS Services'. A red arrow points from the left towards the 'Compute' section. The 'Compute' section contains the following items: Storage & Content Delivery, Database, Networking, Developer Tools, Management Tools, and Security & Identity.

History

All AWS Services

Compute

Storage & Content Delivery

Database

Networking

Developer Tools

Management Tools

Security & Identity

- Open all the services and click on EC2 under Compute services. This will launch the dashboard of EC2.

Here is the EC2 dashboard. Here you will get all the information in gist about the AWS EC2 resources running.

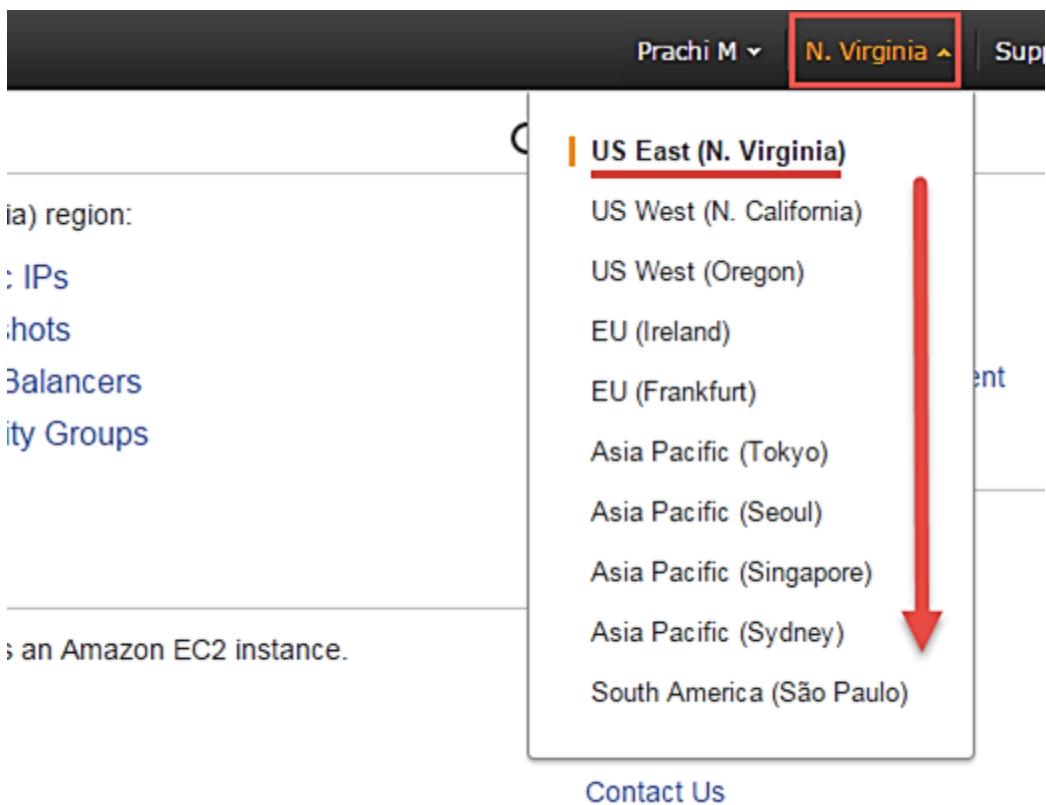
The screenshot shows the AWS EC2 Dashboard. The top navigation bar includes the AWS logo, a Services dropdown, and an Edit dropdown. The left sidebar has a red box around the 'EC2 Dashboard' link, and a red arrow points from this box to the 'Resources' section on the right. The sidebar also lists 'Events', 'Tags', 'Reports', and 'Limits'. Under 'INSTANCES', it lists 'Instances', 'Spot Requests', 'Reserved Instances', 'Scheduled Instances', 'Commands', and 'Dedicated Hosts'. Under 'IMAGES', it lists 'AMIs' and 'Bundle Tasks'. Under 'ELASTIC BLOCK STORE', it lists 'Volumes' and 'Snapshots'. The main content area is titled 'Resources' and displays a summary of Amazon EC2 resources in the US East (N. Virginia) region. A red box highlights this summary table. The table shows:

3	Running Instances	4 Elastic
0	Dedicated Hosts	17 Snap
12	Volumes	0 Load
22	Key Pairs	28 Secu
0	Placement Groups	

Below the summary, there is a message: 'Need fast, reliable, scalable, fully-managed message queuing? Try Amazon Simple Queue Service (SQS)'. The 'Create Instance' section features a large blue 'Launch Instance' button. A note below it states: 'Note: Your instances will launch in the US East (N. Virginia) region'.

Step 2) On the top right corner of the EC2 dashboard, choose the AWS Region in which you want to provision the EC2 server.

Here we are selecting N. Virginia. AWS provides 10 Regions all over the globe.



Step 3) In this step

- Once your desired Region is selected, come back to the EC2 Dashboard.
- Click on 'Launch Instance' button in the section of Create Instance (as shown below).

The screenshot shows the AWS EC2 Dashboard. On the left, there's a sidebar with links like EC2 Dashboard, Events, Tags, Reports, Limits, INSTANCES (with sub-links for Instances, Spot Requests, Reserved Instances, Scheduled Instances, Commands, Dedicated Hosts), IMAGES (with sub-links for AMIs, Bundle Tasks), and ELASTIC BLOCK STORE (with sub-links for Volumes). The main panel has a title 'Resources' and a summary of resources: 3 Running Instances, 0 Dedicated Hosts, 12 Volumes, 22 Key Pairs, and 0 Placement Groups. Below this is a 'Create Instance' section with a large blue 'Launch Instance' button, which is highlighted with a red box. A note below the button says 'To start using Amazon EC2 you will want to launch a'. At the bottom of the main panel is a 'Service Health' section.

- Instance creation wizard page will open as soon as you click 'Launch Instance'.

Choose AMI

Step 1) In this step we will do,

1. You will be asked to choose an AMI of your choice. (An AMI is an Amazon Machine Image. It is a template basically of an Operating System platform which you can use as a base to create your instance). Once you launch an EC2 instance from your preferred AMI, the instance will automatically be booted with the desired OS. (We will see more about AMIs in the coming part of the tutorial).
2. Here we are choosing the default Amazon [Linux](#) (64 bit) AMI.

The screenshot shows the AWS Step Functions console. At the top, there are navigation links: AWS, Services, and Edit. Below these, a horizontal bar indicates the current step: "1. Choose AMI" (highlighted with a red circle containing the number 1), "Choose Instance Type", "Configure Instance", "Add Storage", "Tag Instance", and "Create". The main content area is titled "Step 1: Choose an Amazon Machine Image (AMI)". A sub-section titled "Quick Start" contains three options: "My AMIs", "AWS Marketplace" (which is selected and highlighted with a black background and white text), and "Community AMIs". To the right of the "AWS Marketplace" section, there is a card for "Amazon Linux AMI 2015.09.1 (HVM), SSD Volume". The card includes the AMI icon, the name, a "Free tier eligible" badge, a description stating it's an EBS-backed, AWS-supported AMI with Docker, PHP, Ruby, and Java support, and details about the root device type (ebs) and virtualization type (hvm).

Choose EC2 Instance Types

Step 1) In the next step, you have to choose the type of instance you require based on your business needs.

1. We will choose t2.micro instance type, which is a 1vCPU and 1GB memory server offered by AWS.
2. Click on "Configure Instance Details" for further configurations

The screenshot shows the AWS EC2 instance creation wizard at Step 2: Choose an Instance Type. The 't2.micro' instance type is selected and highlighted with a red box and a red circle containing the number 1. A tooltip 'Free tier eligible' is visible over the 't2.micro' cell.

	Family	Type	vCPUs	Memory (GiB)	Instance Type
<input type="checkbox"/>	General purpose	t2.nano	1	0.5	
<input checked="" type="checkbox"/>	General purpose	t2.micro	1	1	Free tier eligible
<input type="checkbox"/>	General purpose	t2.small	1	2	
<input type="checkbox"/>	General purpose	t2.medium	2	4	
<input type="checkbox"/>	General purpose	t2.large	2	8	
<input type="checkbox"/>	General purpose	m4.large	2	8	

- In the next step of the wizard, enter details like no. of instances you want to launch at a time.
- Here we are launching one instance.

Configure Instance

Step 1) No. of instances- you can provision up to 20 instances at a time. Here we are launching one instance.

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request role to the instance, and more.

Number of instances Launch into Auto Scaling

Purchasing option Request Spot instances

Step 2) Under Purchasing Options, keep the option of 'Request Spot Instances' unchecked as of now. (This is done when we wish to launch Spot instances instead of on-demand ones. We will come back to Spot instances in the later part of the tutorial).

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request role to the instance, and more.

Number of instances Launch into Auto Scaling

Purchasing option Request Spot instances

Step 3) Next, we have to configure some basic networking details for our EC2 server.

- You have to decide here, in which VPC (Virtual Private Cloud) you want to launch your instance and under which subnets inside your VPC. It is better to determine and plan this prior to launching the instance. Your AWS architecture set-up should include IP ranges for your subnets etc. pre-planned for better management. (We will see how to create a new VPC in Networking section of the tutorial).

- Subnetting should also be pre-planned. E.g.: If it's a web server you should place it in the public subnet and if it's a DB server, you should place it in a private subnet all inside your VPC.

Below,

1. Network section will give a list of VPCs available in our platform.
2. Select an already existing VPC
3. You can also create a new VPC

Here I have selected an already existing VPC where I want to launch my instance.

The screenshot shows the AWS EC2 instance creation wizard at Step 3: Configure Instance Details. The 'Number of instances' is set to 1. Under 'Purchasing option', there is a checkbox for 'Request Spot instances' which is unchecked. The 'Network' section is highlighted with a red circle containing the number 1. It shows a dropdown menu with several VPC options listed:

- vpc-d5194fb0 (192.168.0.0/16) | Prachi_Test - VPC
- Launch into EC2-Classic
- vpc-621a5e07 (172.20.0.0/16) | POC_vpc
- vpc-d5194fb0 (192.168.0.0/16) | Prachi_Test - VPC** (This option is selected)
- vpc-8452bce0 (172.20.0.0/16) | POC_vpc
- vpc-823e39e7 (172.22.0.0/16) | TVPC
- vpc-4c51bf28 (10.0.0.0/16) | POC_vpc3

The 'Auto-assign Public IP' and 'IAM role' sections are also visible below the network dropdown.

Step 4) In this step,

- A VPC consists of subnets, which are IP ranges that are separated for restricting access.
- Below,

1. Under Subnets, you can choose the subnet where you want to place your instance.
2. I have chosen an already existing public subnet.
3. You can also create a new subnet in this step.

The screenshot shows the 'Step 3: Configure Instance Details' page in the AWS console. The top navigation bar includes 'AWS', 'Services', 'Edit', and tabs for '1. Choose AMI', '2. Choose Instance Type', '3. Configure Instance' (which is selected), '4. Add Storage', '5. Tag Instance', and '6. Co...'. The main section is titled 'Step 3: Configure Instance Details' with the sub-instruction 'Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request a role to the instance, and more.' Below this, there are several configuration fields:

- Number of instances:** Set to 1.
- Purchasing option:** Includes a checkbox for 'Request Spot instances' which is unchecked.
- Network:** Shows 'vpc-d5194fb0 (192.168.0.0/16) | Prachi_Test - VPC'. A red circle highlights the 'Subnet' dropdown menu, which lists four subnets:
 - subnet-b3e3d0ea(192.168.2.0/24) | Prachi_Test-Public
 - subnet-0eeef779(192.168.3.0/24) | Prachi_Test_Public_S
 - subnet-a94427de(192.168.1.0/24) | Prachi_Test- Public S
 - subnet-b3e3d0ea(192.168.2.0/24) | Prachi_Test-Public s
- Auto-assign Public IP:** Includes a checkbox which is unchecked.
- IAM role:** Set to 'None'.

- Once your instance is launched in a public subnet, AWS will assign a dynamic public IP to it from their pool of IPs.

Step 5) In this step,

- You can choose if you want AWS to assign it an IP automatically, or you want to do it manually later. You can enable/ disable 'Auto assign Public IP' feature here likewise.
- Here we are going to assign this instance a static IP called as EIP (Elastic IP) later. So we keep this feature disabled as of now.

AWS Services Edit

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Tag Instance 6. Co

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request role to the instance, and more.

Number of instances Launch into Auto Scaling

Purchasing option Request Spot instances

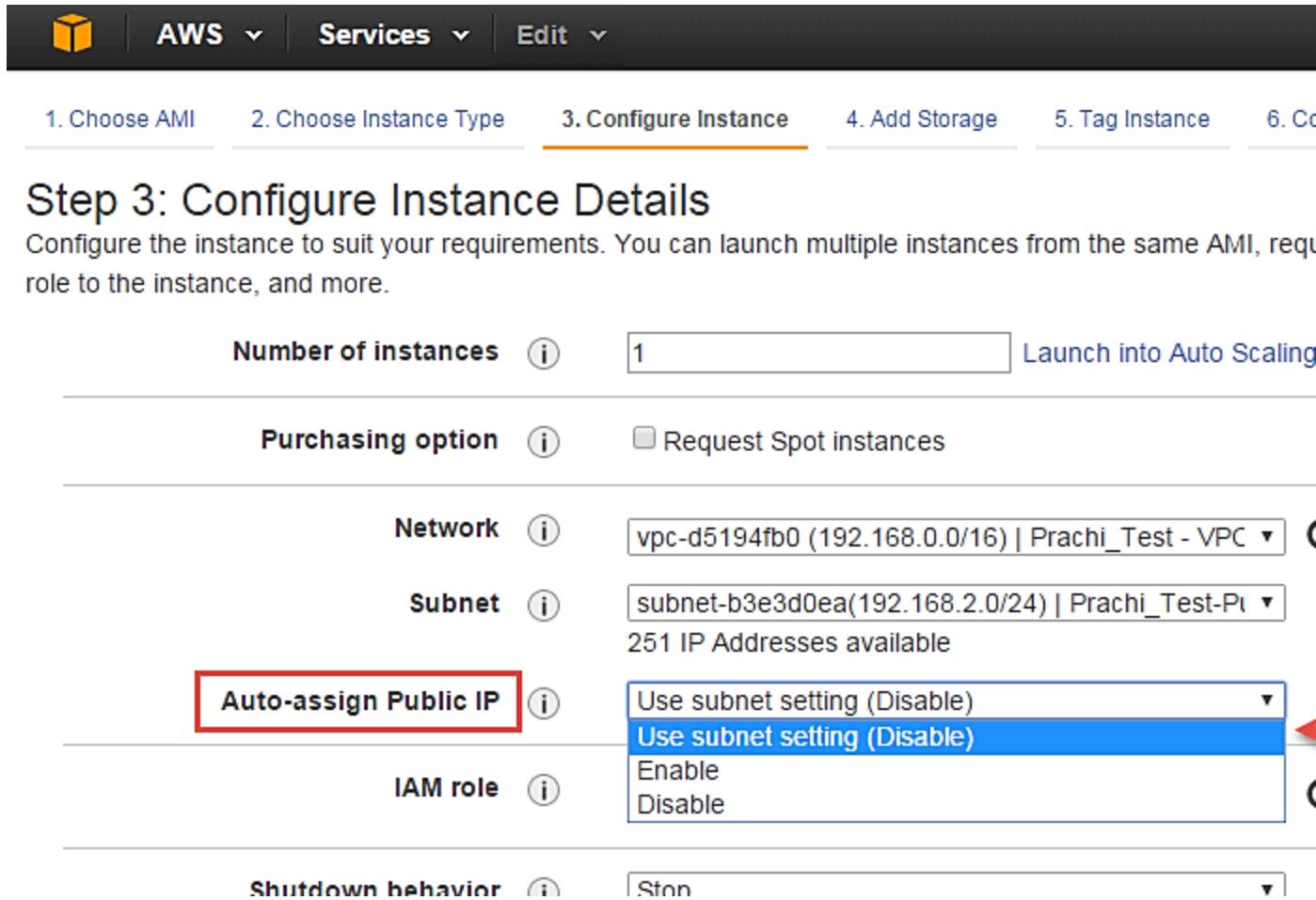
Network vpc-d5194fb0 (192.168.0.0/16) | Prachi_Test - VPC

Subnet subnet-b3e3d0ea(192.168.2.0/24) | Prachi_Test-PI
251 IP Addresses available

Auto-assign Public IP

IAM role

Shutdown behavior



Step 6) In this step,

- In the following step, keep the option of IAM role 'None' as of now. We will visit the topic of IAM role in detail in IAM services.

AWS Services Edit

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Tag Instance 6. Create instance

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request specific role to the instance, and more.

Number of instances (i) Launch into Auto Scaling

Purchasing option (i) Request Spot instances

Network (i) vpc-d5194fb0 (192.168.0.0/16) | Prachi_Test - VPC ▾

Subnet (i) subnet-b3e3d0ea(192.168.2.0/24) | Prachi_Test-PI ▾
251 IP Addresses available

Auto-assign Public IP (i) Use subnet setting (Disable)

IAM role (i) None 

Step 7) In this step, you have to do following things

- Shutdown Behavior – when you accidentally shut down your instance, you surely don't want it to be deleted but stopped.
- Here we are defining my shutdown behavior as Stop.

AWS Services Edit

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Tag Instance 6. Co

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request role to the instance, and more.

Number of instances Launch into Auto Scaling

Purchasing option Request Spot instances

Network

Subnet 251 IP Addresses available

Auto-assign Public IP

IAM role

Shutdown behavior

Enable termination protection

Monitoring Enable CloudWatch detailed monitoring

Step 8) In this step,

- In case, you have accidentally terminated your instance, AWS has a layer of security mechanism. It will not delete your instance if you have enabled accidental termination protection.
- Here we are checking the option for further protecting our instance from accidental termination.

The screenshot shows the AWS CloudFormation console during the 'Step 3: Configure Instance Details' phase. The top navigation bar includes 'AWS', 'Services', 'Edit', and tabs for 'Console Home', '1. Choose AMI', '2. Choose Instance Type', '3. Configure Instance' (which is active), '4. Add Storage', '5. Tag Instance', and '6. Create'. The main section is titled 'Step 3: Configure Instance Details'. It contains several configuration fields: 'IAM role' set to 'None'; 'Shutdown behavior' set to 'Stop'; 'Enable termination protection' (checkbox checked, highlighted with a red box and an arrow); and 'Monitoring' (checkbox unchecked). A note below monitoring says 'Additional charges apply.'

Step 9) In this step,

- Under Monitoring- you can enable Detailed Monitoring if your instance is a business critical instance. Here we have kept the option unchecked. AWS will always provide Basic monitoring on your instance free of cost. We will visit the topic of monitoring in AWS Cloud Watch part of the tutorial.
- Under Tenancy- select the option if shared tenancy. If your application is a highly secure application, then you should go for dedicated capacity. AWS provides both options.

The screenshot shows the AWS CloudFormation console during the 'Step 3: Configure Instance Details' process. At the top, there are tabs for '1. Choose AMI', '2. Choose Instance Type', '3. Configure Instance' (which is highlighted in orange), '4. Add Storage', '5. Tag Instance', and '6. Create'. Below the tabs, the title 'Step 3: Configure Instance Details' is displayed. Under the 'Tenancy' section, a red box highlights the 'Tenancy' button, and a red arrow points to the 'Shared - Run a shared hardware instance' option in the dropdown menu.

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Tag Instance 6. Create

Step 3: Configure Instance Details

IAM role ⓘ None

Shutdown behavior ⓘ Stop

Enable termination protection ⓘ Protect against accidental termination

Monitoring ⓘ Enable CloudWatch detailed monitoring
Additional charges apply.

Tenancy ⓘ

- Shared - Run a shared hardware instance
- Shared - Run a shared hardware instance**
- Dedicated - Run a Dedicated instance
- Dedicated host - Launch this instance on a Dedicated host

Step 10) In this step,

- Click on 'Add Storage' to add data volumes to your instance in next step.

AWS Services Edit

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Tag Instance 6. Launch

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request specific role to the instance, and more.

Number of instances	<input type="text" value="1"/>	Launch into Auto Scaling group
Purchasing option	<input checked="" type="checkbox"/> Request Spot instances	
Network	vpc-d5194fb0 (192.168.0.0/16) Prachi_Test - VPC	
Subnet	subnet-b3e3d0ea(192.168.2.0/24) Prachi_Test-Private Subnet 251 IP Addresses available	
Auto-assign Public IP	<input checked="" type="checkbox"/> Use subnet setting (Disable)	
IAM role	None	
Shutdown behavior	Stop	
Enable termination protection	<input checked="" type="checkbox"/> Protect against accidental termination	
Monitoring	<input type="checkbox"/> Enable CloudWatch detailed monitoring <small>Additional charges apply.</small>	
Tenancy	Shared - Run a shared hardware instance	

Add Storage

Step 1) In this step we do following things,

- In the Add Storage step, you'll see that the instance has been automatically provisioned a General Purpose SSD root volume of 8GB. (Maximum volume size we can give to a General Purpose volume is 16GB)

- You can change your volume size, add new volumes, change the volume type, etc.
- AWS provides 3 types of EBS volumes- Magnetic, General Purpose SSD, Provisioned IOPs. You can choose a volume type based on your application's IOPs needs.

The screenshot shows the AWS Step 4: Add Storage configuration page. At the top, there are tabs: 1. Choose AMI, 2. Choose Instance Type, 3. Configure Instance, 4. Add Storage (which is highlighted in orange), 5. Tag Instance, and 6. Create. Below the tabs, the heading "Step 4: Add Storage" is displayed. The main area shows a table with columns: Volume Type, Device, Snapshot, Size (GiB), and Volume Type. The first row is for the Root volume, which has a device of /dev/xvda, a snapshot of snap-a17f1036, and a size of 8 GiB. A red arrow points to the "Size (GiB)" input field. To the right of the table, there is a dropdown menu showing four options: General Purpose SSD, General Purpose SSD, Provisioned IOPS SSD, and Magnetic. Below the table, a note states: "Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. Learn more about usage restrictions." The "Add New Volume" button is also visible.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type
Root	/dev/xvda	snap-a17f1036	8	General Purpose SSD General Purpose SSD Provisioned IOPS SSD Magnetic
<input type="button" value="Add New Volume"/>				

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

Tag Instance

Step 1) In this step

- you can tag your instance with a key-value pair. This gives visibility to the AWS account administrator when there are lot number of instances.
- The instances should be tagged based on their department, environment like Dev/SIT/Prod. Etc. this gives a clear view of the costing on the instances under one common tag.

1. Here we have tagged the instance as a **Dev_Web server 01**

2. Go to configure Security Groups later

The screenshot shows the AWS Step 5: Tag Instance configuration screen. At the top, there are tabs: 1. Choose AMI, 2. Choose Instance Type, 3. Configure Instance, 4. Add Storage, 5. Tag Instance (which is highlighted), and 6. Create. Below the tabs, the section title is "Step 5: Tag Instance". A note says: "A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Dev_Web." The main area has a table with two columns: "Key" and "Value". In the "Key" column, there is a text input field with the placeholder "(127 characters maximum)". In the "Value" column, there is a text input field containing "Dev_Web". To the left of the "Value" field is a button labeled "Create Tag" with the note "(Up to 10 tags maximum)". A red arrow points from a circled '1' to the "Value" field.

Configure Security Groups

Step 1) In this next step of configuring Security Groups, you can restrict traffic on your instance ports. This is an added firewall mechanism provided by AWS apart from your instance's OS firewall.

You can define open ports and IPs.

- Since our server is a webserver=, we will do following things
 - 1. Creating a new Security Group
 - 2. Naming our SG for easier reference
 - 3. Defining protocols which we want enabled on my instance
 - 4. Assigning IPs which are allowed to access our instance on the said protocols
 - 5. Once, the firewall rules are set- Review and launch

AWS Services Edit

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Tag Instance 6. Co

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 traffic from the Internet to your server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP port, or add rules that restrict access to your instance.

[Learn more about Amazon EC2 security groups.](#)

Assign a security group: Create a new security group 1 Select an existing security group

Security group name: Web Server SG 2

Description: launch-wizard-7 created 2016-02-03T19:49:12Z 2016-02-03T19:49:12Z +05:30

Type	Protocol	Port Range
SSH	TCP	22
HTTP	TCP	80
HTTPS	TCP	443

Add Rule 3

Review Instances

Step 1) In this step, we will review all our choices and parameters and go ahead to launch our instance.

AWS Services Edit

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Tag Instance 6. Continue

Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to start launching your instance.

AMI Details

 **Amazon Linux AMI 2015.09.1 (HVM), SSD Volume Type - ami-60b6c60a**

Free tier eligible The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS Docker, PHP, MySQL, PostgreSQL, and other packages

Root Device Type: ebs Virtualization type: hvm

Instance Type

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)
t2.micro	Variable	1	1	EBS only

Security Groups

Security group name: Web Server SG
Description: launch-wizard-7 created 2016-02-03T19:49:12.288+05:30

Type (i)	Protocol (i)	Port (i)
----------	--------------	----------

Step 2) In the next step you will be asked to create a key pair to login to your instance. A key pair is a set of public-private keys.

AWS stores the private key in the instance, and you are asked to download the private key. Make sure you download the key and keep it safe and secured; if it is lost you cannot download it again.

1. Create a new key pair
2. Give a name to your key

3. Download and save it in your secured folder

Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click [Launch](#).

AMI Details

Amazon Linux AMI 2015.0

Free tier eligible

The Amazon Linux AMI is an Enterprise-grade Linux distribution for Amazon EC2. It includes Docker, PHP, MySQL, PostgreSQL, and other popular open-source software.

Root Device Type: ebs Virtualization: Amazon VPC

Instance Type

Instance Type	ECUs
t2.micro	Variable

Security Groups

Security group name: Web Server

Description: launch-vidya

Type: (i)

Select an existing key pair or create a new one

A key pair consists of a **public key** that AWS stores, and a **private key** that you store on your local machine. They allow you to connect to your instance securely. For Windows, you can download the private key file (.pem) to obtain the password used to log into your instance. For Linux, you can use the private key file to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys for this instance. You can always [remove](#) existing key pairs from a public AMI.

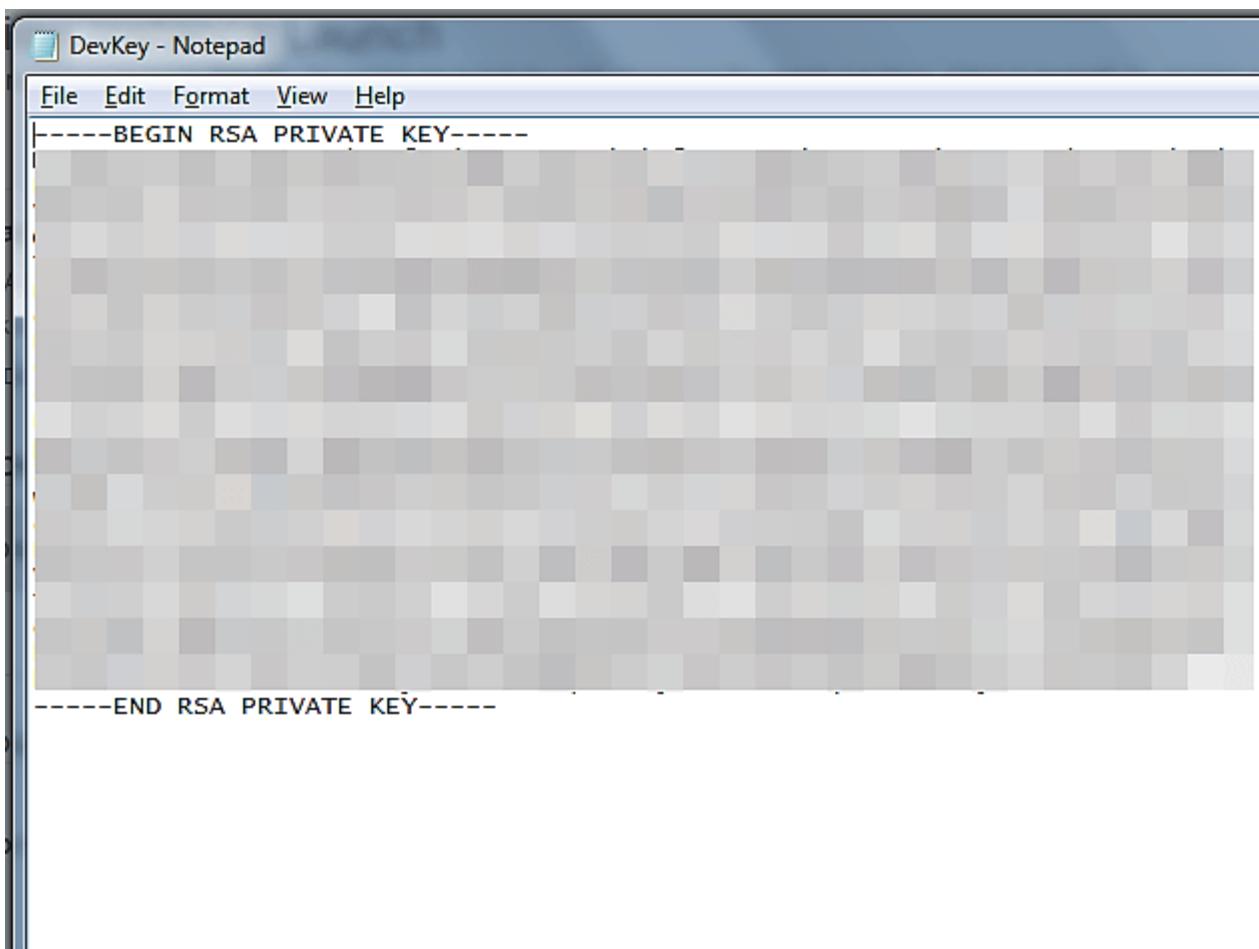
Create a new key pair

Key pair name

Dev Key

You have to download the **private key file** (*.pem) and store it in a secure and accessible location. You can download it again after it's created.

- When you download your key, you can open and have a look at your RSA private key.



Step 3) Once you are done downloading and saving your key, launch your instance.

Select an existing key pair or create a new key pair

X

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](#).

Create a new key pair

Key pair name

Dev Key

[Download Key Pair](#)



You have to download the **private key file** (*.pem file) before you can continue.

Store it in a secure and accessible location. You will not be able to download the file again after it's created.

[Cancel](#)

[Launch Instances](#)

- You can see the launch status meanwhile.

Launch Status



Initiating Instance Launches

Please do not close your browser while this is loading

Creating security groups... Successful

Authorizing inbound rules...

- You can also see the launch log.

Launch Status



Your instances are now launching

The following instance launches have been initiated: i-4c2c3cff [Hide launch log](#)

Creating security groups Successful (sg-62d7d21b)

Authorizing inbound rules Successful

Initiating launches Successful

Applying tags Successful

Launch initiation complete



Get notified of estimated charges

Create [billing alerts](#) to get an email notification when estimated charges on your AWS bill exceed

- Click on the 'Instances' option on the left pane where you can see the status of the instance as 'Pending' for a brief while.

The screenshot shows the AWS EC2 Instances page. The top navigation bar includes the AWS logo, a Services dropdown, and an Edit dropdown. Below the navigation is a toolbar with 'Launch Instance', 'Connect', and 'Actions' buttons. A search bar is present above the main table. The left sidebar has links for EC2 Dashboard, Events, Tags, Reports, Limits, and a collapsed 'INSTANCES' section which is expanded to show 'Instances', 'Spot Requests', and 'Reserved Instances'. The main content area displays a table of instances. The first row in the table is highlighted and shows the following data:

	Name	Instance ID	Type
	Dev_Web Server 01	i-4c2c3cff	t2.micro

- Once your instance is up and running, you can see its status as 'Running' now.
- Note that the instance has received a Private IP from the pool of AWS.

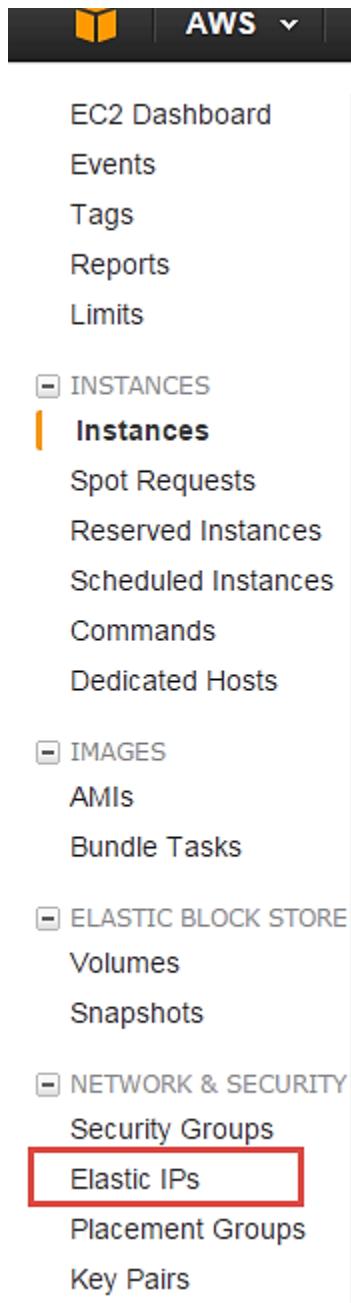
The screenshot shows the AWS EC2 Instances page. On the left sidebar, under the 'INSTANCES' section, 'Instances' is selected. The main area displays a table of instances. One instance is highlighted: 'Dev_Web_Server_01' (Instance ID: i-4c2c3cff, Instance Type: t2.micro). Below the table, detailed information for this instance is shown in a card:

Description	Status Checks	Monitoring	Tags
Instance ID	i-4c2c3cff		
Instance state	running		
Instance type	t2.micro		
Private DNS	ip-192-168-2-167.ec2.internal		
Private IPs	192.168.2.167		
Secondary private IPs			
VPC ID	vpc-d5194fb0		
Subnet ID	subnet-b3e3d0ea		
Network interfaces	eth0		
Source/dest. check	True		
ClassicLink	-		
EBS-optimized	False		

Create a EIP and connect to your instance

An EIP is a static public IP provided by AWS. It stands for Elastic IP. Normally when you create an instance, it will receive a public IP from the AWS's pool automatically. If you stop/reboot your instance, this public IP will change- it's dynamic. In order for your application to have a static IP from where you can connect via public networks, you can use an EIP.

Step 1) On the left pane of EC2 Dashboard, you can go to 'Elastic IPs' as shown below.



Step 2) Allocate a new Elastic IP Address.

The screenshot shows the AWS EC2 Dashboard. The left sidebar lists navigation options: EC2 Dashboard, Events, Tags, Reports, Limits, INSTANCES (with sub-options: Instances, Spot Requests, Reserved Instances, Scheduled Instances, Commands, Dedicated Hosts), and IMAGES (with sub-options: AMIs, Bundle Tasks). The main content area has a blue header bar with the text "Allocate New Address" and "Actions". A red arrow points to the "Allocate New Address" button. Below the header is a search bar with the placeholder "Filter by attributes or search by keyword". A table lists four Elastic IP addresses:

Elastic IP	Allocation ID	Instance
52.1.77.244	eipalloc-5a42cb3f	i-070a0dd4 (Aeri)
52.2.194.185	eipalloc-0a8cd76f	i-20787ddd (Pra)
52.6.162.134	eipalloc-5990e33c	i-1911cacb (S3F)
54.152.175.190	eipalloc-20138f45	i-7404e4fd (SUS)

Step 3) Allocate this IP to be used in a VPC scope.

The screenshot shows the AWS EC2 Dashboard with the 'Allocate New Address' dialog open. The left sidebar lists navigation options like EC2 Dashboard, Events, Tags, Reports, Limits, INSTANCES, IMAGES, and ELASTIC BLOCK STORE. The main area displays a table of existing Elastic IP addresses, and a modal dialog is centered over it.

Allocate New Address

Filter by attributes or search by keyword

Elastic IP	Allocation ID	Instance
52.1.77.244	eipalloc-5a42cb3f	i-070a0dd4 (Aeris)
52.2.194.185	eipalloc-0a8cd76f	i-20787ddd (Prac)
52.6.162.134	eipalloc-5990e33c	i-1911ca3ch (S3E)
54.152.175.190		

Select an address above.

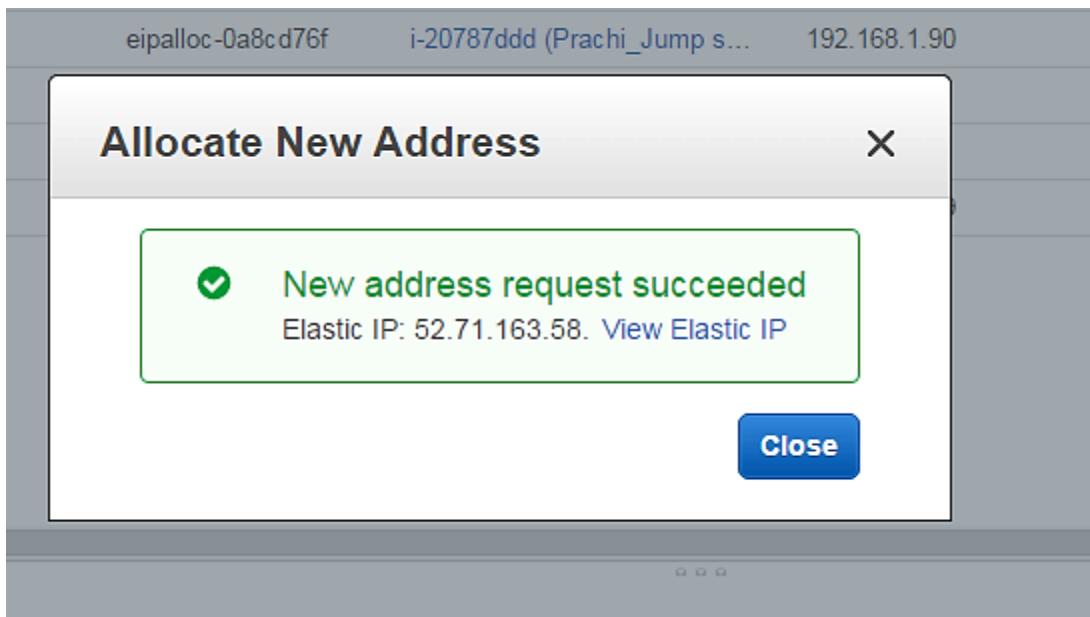
Allocate New Address

Are you sure you want to allocate a new EIP?

EIP used in: VPC ▾

EC2
VPC
Cancel

- Your request will succeed if you don't have 5 or more than 5 EIPs already in your account.



Step 4) Now assign this IP to your instance.

1. Select the said IP
2. Click on Actions -> Associate Address

A screenshot of the AWS EC2 Instances page. On the left sidebar, under the "INSTANCES" section, "Instances" is selected, indicated by a red circle with the number 1. In the main content area, there is a table of elastic IPs. One row for IP 52.71.163.58 has a blue selection box next to it, indicating it is selected. To the right of the table, an "Actions" dropdown menu is open, showing options: "Allocate New Address", "Release Addresses", "Associate Address", and "Dissociate Address". The "Associate Address" option is highlighted with a red circle and the number 2. Red arrows point from the numbered circles to their respective elements: one arrow points from circle 1 to the "Instances" link in the sidebar, and another points from circle 2 to the "Associate Address" option in the dropdown menu.

Step 5) In the next page,

1. Search for your instance and
2. Associate the IP to it.

Associate Address

Select the instance OR network interface to which you wish to associate this IP address (52.71.163.58)

Instance	dev
Network Interface	i-4c2c3cff (Dev_Web Server 01) (running)

Private IP Address	192.168.2.167*
---------------------------	----------------

Reassociation (i)

Warning
If you associate an Elastic IP address with your instance, your current public IP address is released.
(i)

Step 6) Come back to your instances screen, you'll see that your instance has received your EIP.

The screenshot shows the AWS EC2 Instances page. The left sidebar has options like EC2 Dashboard, Events, Tags, Reports, Limits, INSTANCES (with Instances selected), Spot Requests, Reserved Instances, Scheduled Instances, Commands, Dedicated Hosts, and IMAGES. The main area has buttons for Launch Instance, Connect, and Actions. A search bar shows 'search : i-4c2c3cf' with an 'Add filter' link. Below it is a table with columns: Name, Instance ID, and Instance Type. One row is selected for 'Dev_Web_Server_01' with Instance ID 'i-4c2c3cff' and Type 't2.micro'. At the bottom, tabs for Description, Status Checks, Monitoring, and Tags are shown, along with Instance ID 'i-4c2c3cff' and Instance state 'running'.

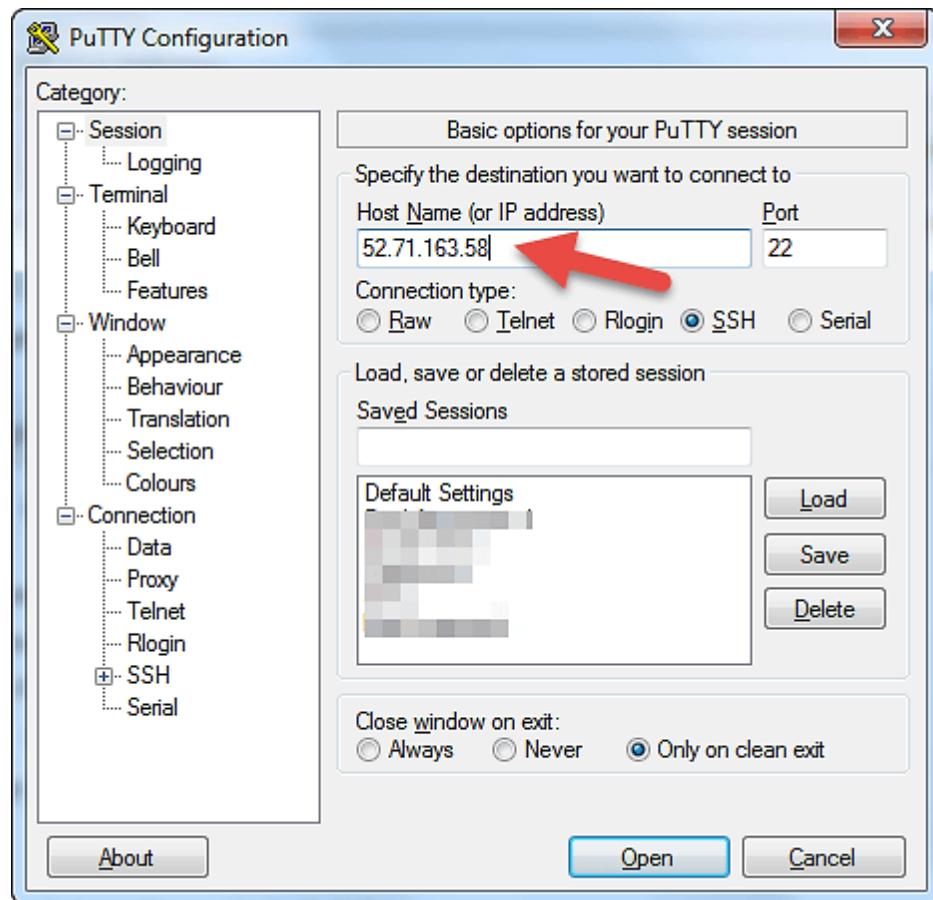
Name	Instance ID	Instance Type
Dev_Web_Server_01	i-4c2c3cff	t2.micro

Instance: i-4c2c3cff (Dev_Web_Server_01) Elastic IP: 52.71.163.58

Description Status Checks Monitoring Tags

Instance ID: i-4c2c3cff
Instance state: running

Step 7) Now open putty from your programs list and add your same EIP in there as below.

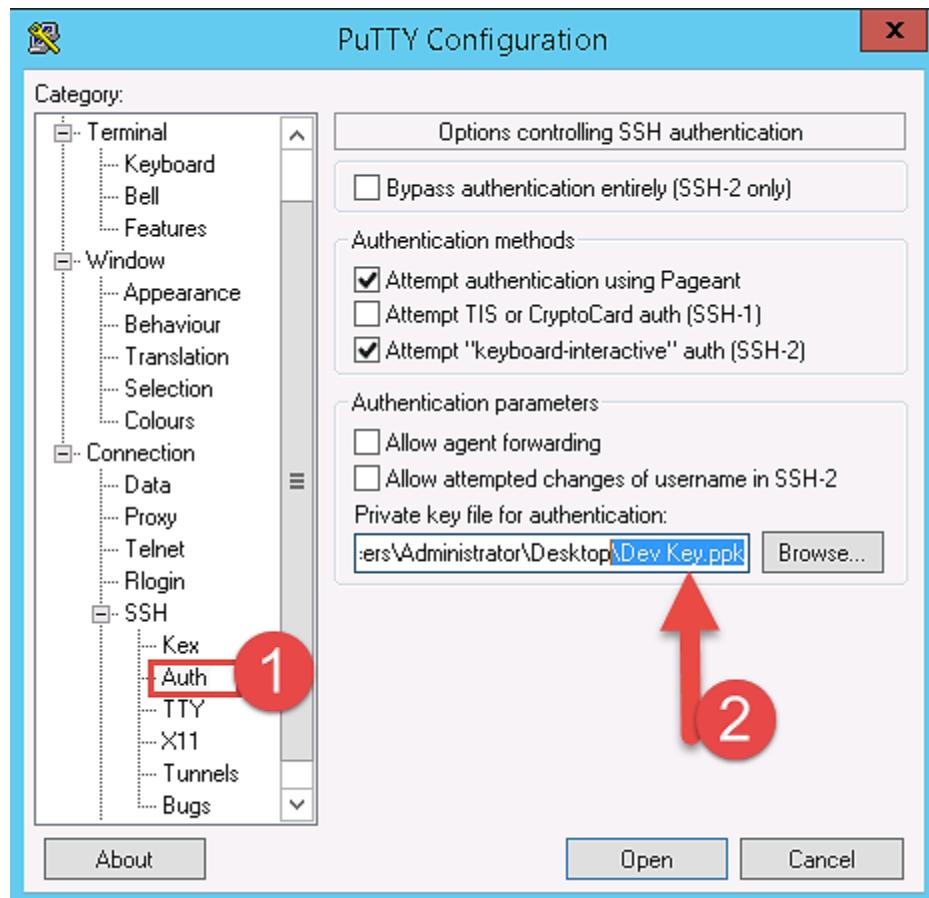


Step 8) In this step,

Add your private key in putty for secure connection

1. Go to Auth
2. Add your private key in .ppk (putty private key) format. You will need to convert pem file from AWS to ppk using puttygen

Once done click on "Open" button



- Once you connect, you will successfully see the [Linux](#) prompt.
- Please note that the machine you are connecting from should be enabled on the instance Security Group for SSH (like in the steps above).

A terminal window titled 'PuTTY' shows a successful SSH login. The title bar reads 'ec2-user@ip-192-168-2-167:~'. The session output is as follows:
login as: ec2-user ←
Authenticating with public key "imported-openssh-key"
[ec2-user@ip-192-168-2-167 ~]\$ █

Once you become familiar with the above steps for launching the instance, it becomes a matter of 2 minutes to launch the same!

You can now use your on-demand EC2 server for your applications.

What is Spot Instance?

A spot Instance is an offering from AWS; it allows an AWS business subscriber to bid on unused AWS compute capacity. The hourly price for a Spot instance is decided by AWS, and it fluctuates depending on the supply and demand for Spot instances.

Your Spot instance runs whenever your bid exceeds the current market price. The price of a spot instance varies based on the instance type and the Availability Zone in which the instance can be provisioned.

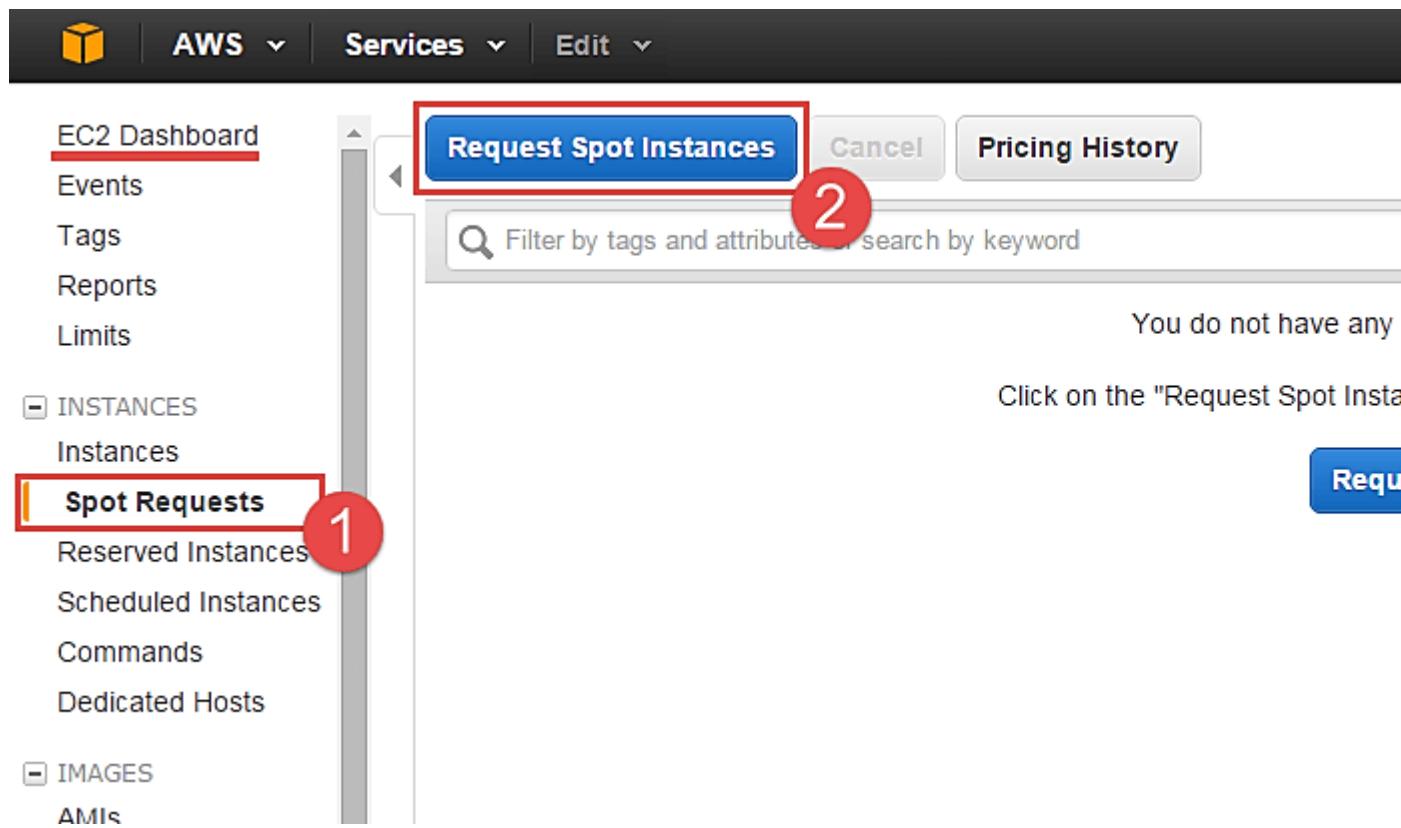
When your bid price exceeds the market spot price of the instance called as the 'spot price,' your instance stays running. When the spot price overshoots the bid price, AWS will terminate your instance automatically. Therefore, it is necessary to plan the spot instances in your application architecture carefully.

Create a Spot Request

In order to launch a spot instance, you have to first create a Spot Request.

Follow the steps below to create a Spot Request.

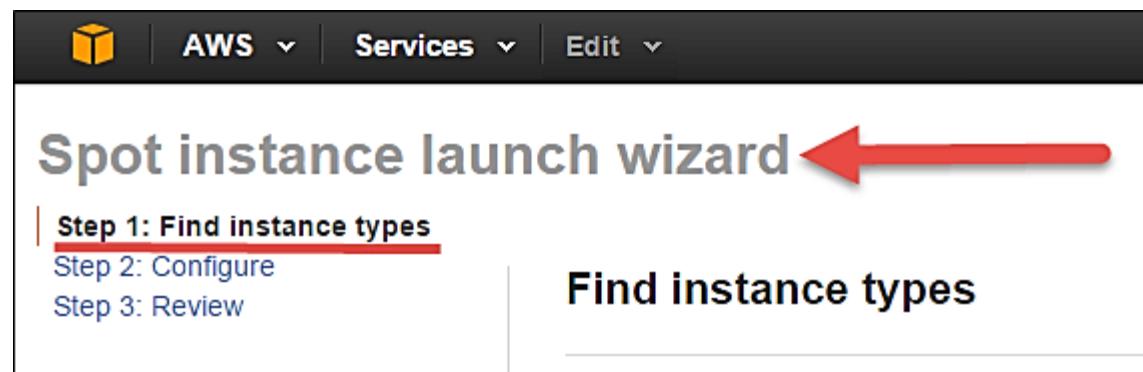
1. On the EC2 Dashboard select 'Spot Requests' from the left pane under Instances.
2. Click on the button 'Request Spot Instances" as shown below.



Spot instance launch wizard will open up. You can now go ahead with selecting the parameters and the instance configuration.

Find Instance Types

The first step for spot instance is to "Find instance types."



Step 1) Select an AMI- an AMI is a template consisting of the OS platform and software to be installed in the instance. Select your desired AMI from the existing list. We are selecting Amazon Linux AMI for this tutorial.

Spot instance launch wizard

Step 1: Find instance types

Step 2: Configure
Step 3: Review

Find instance types

AMI i

Capacity unit i

Target capacity i

Bid price i

% of On-Demand

Network filter i

omit VPC-only instance types

Amazon Linux AMI 2015.0
Linux/UNIX
Amazon Linux AMI 2015
SUSE Linux Enterprise
Ubuntu Server 14.04 LTS
Amazon Linux AMI 2015
Windows
Microsoft Windows Server
Custom
Use custom AMI

Select instance types: i

Step 2) Capacity Unit- a Capacity Unit is your application requirement. You may decide to launch an instance based on the instance type, vCPU or custom configuration like your choice of vCPU/memory/storage requirements. Here we are selecting an Instance.

Spot instance launch wizard

Step 1: Find instance types

Step 2: Configure

Step 3: Review

Find instance types

The screenshot shows the 'Find instance types' section of the AWS Spot instance launch wizard. It includes fields for AMI, Capacity unit, Target capacity, Bid price, and Network filter, along with a checkbox for omitting VPC-only instances.

AMI: Amazon Linux AMI 2015.09

Capacity unit: Instance (selected)

Target capacity: 1

Bid price: per instance/hour (selected)

Network filter: omit VPC-only instance types (unchecked)

If you wish to customize the capacity, you can add your choice of

1. vCPU,
2. Memory and
3. Instance storage as below.

The screenshot shows the "Spot instance launch wizard" on the AWS console. The left sidebar lists three steps: "Step 1: Find instance types" (selected), "Step 2: Configure", and "Step 3: Review". The main content area is titled "Find instance types". It includes fields for "AMI" (Amazon Linux AMI 2015.09), "Capacity unit" (Customize your unit...), "Minimum unit requirements" (vCPUs: 2, highlighted with a red box and circled with a red number 1), and "Target capacity" (1).

Step 1: Find instance types

Step 2: Configure

Step 3: Review

Find instance types

AMI

Amazon Linux AMI 2015.09

Capacity unit

Customize your unit...

Minimum unit requirements

vCPUs:

2

1

Current generation request

Target capacity

1

Step 3) Target Capacity depicts how many spot instances you wish to maintain in your request. Here we are selecting one.

Spot instance launch wizard

Step 1: Find instance types

Step 2: Configure

Step 3: Review

Find instance types

AMI

Amazon Linux AMI 2015.09

Capacity unit

Instance

Target capacity

1

Bid price

per instance/hour

% of On-Demand

Network filter

omit VPC-only instance types

Step 4) Bid Price – this is the maximum price we are ready to pay for the instance. We are going to set a particular price per instance/hour. This is the simplest to calculate based on our business requirement. We will see ahead how we should determine the bid price so that our bid price always remains high and doesn't exceed the spot price so that our instance keeps running.

Spot instance launch wizard

Step 1: Find instance types

Step 2: Configure

Step 3: Review

Find instance types

AMI

Amazon Linux AMI 2015.09

Capacity unit

Instance

Target capacity

1

Bid price

per instance/hour

% of On-Demand

Network filter

omit VPC-only instance types

just below the bid price you can see a button of Pricing History. Click on that as shown below.

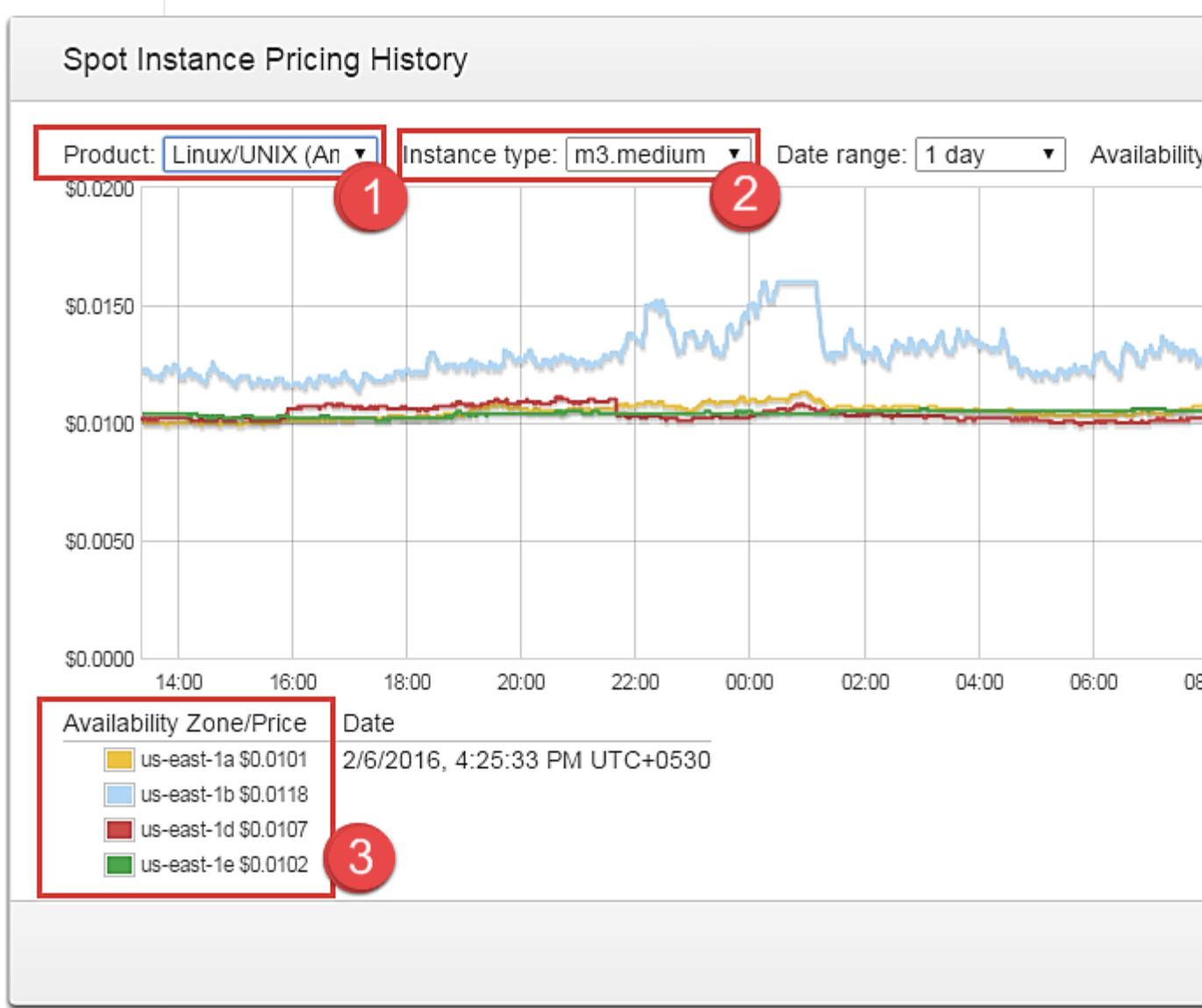
The screenshot shows the AWS Pricing History interface. At the top, there are two radio button options for 'Bid price': 'per instance/hour' (selected) and '% of On-Demand'. Below these are input fields for '\$' (0.07) and '%' (empty). A 'Network filter' checkbox is also present. The main section is titled 'Select instance types:' with a red border around the 'Pricing History' button. The table below lists instance types with columns for vCPUs, Memory (GiB), Storage (GB), Weighted capacity, and Total bid price.

Instance type	vCPUs	Memory (GiB)	Storage (GB)	Weighted capacity	Total bid price
All instance types	1	(Any)	(Any)	<input type="button" value="edit"/>	<input type="button" value="edit"/>
c3.large	2	3.75	2 x 16 SSD	1	\$0.07
c3.xlarge	4	7.5	2 x 40 SSD	1	\$0.07

Here in Pricing History, we can see a graph depicting instance pricing trends with historical data. You can select the parameters and get an idea of the pricing of our desired instance over a period of time.

1. Select the product. We have selected our Linux AMI.
2. Select the instance type. We have selected m3.medium.
3. Note the average prices for over a day here.

Thus, from the chart below, we can see that the instance type that we are planning to provision lies in the pricing range of \$0.01xx, and it seems that Availability Zone 'us-east 1a' has the lowest price.



cont. to step 4.

So let's come back to our step of quoting a bid price.

For the sake of maintaining our instance always available and if it falls within our budget, we can quote a higher bid price. Here we have quoted a slightly higher price of \$0.05.

Spot instance launch wizard

Step 1: Find instance types

Step 2: Configure

Step 3: Review

Find instance types

AMI

Amazon Linux AMI 2015.09

Capacity unit

Instance

Target capacity

1

Bid price

per instance/hour

% of On-Demand

Network filter

omit VPC-only instance type

You can see some trends in the wizard itself.

1. Note the instance types section
2. Select the instance type that we are planning to provision
3. Note the price that we are planning to bid. % of on-demand shows us that our quoted price is 75% of the on-demand price for the same instance type. This means we are saving 25% per hour as compared to an on-demand instance. You can further lower the price and save costs drastically.

Select instance types: i

1

Pricing History

Instance type	vCPUs	Memory (GiB)	Storage (GB)	Weighted capacity i	Total price i
General purpose	1	(Any)	(Any)	i	i
<input checked="" type="checkbox"/> m3.medium	1	3.75	1 x 4 SSD	1	\$0.05
<input type="checkbox"/> m3.large	2	7.5	1 x 32 SSD	1	\$0.05
<input type="checkbox"/> m3.xlarge	4	15	2 x 40 SSD	1	\$0.05
<input type="checkbox"/> m3.2xlarge	8	30	2 x 80 SSD	1	\$0.05

Step 5) Once we are done looking at the trends and quoting our bid price, click on next.

Select instance types: ⓘ

Pricing His

Instance type	vCPUs	Memory (GiB)	Storage (GB)	Weighted capacity ⓘ	Total price ⚡
General purpose	1	(Any)	(Any)		
<input checked="" type="checkbox"/> m3.medium	1	3.75	1 x 4 SSD	1	\$0.0
<input type="checkbox"/> m3.large	2	7.5	1 x 32 SSD	1	\$0.0
<input type="checkbox"/> m3.xlarge	4	15	2 x 40 SSD	1	\$0.0
<input type="checkbox"/> m3.2xlarge	8	30	2 x 80 SSD	1	\$0.0
<input type="checkbox"/> m4.large	2	8	EBS only	1	\$0.0
<input type="checkbox"/> m4.xlarge	4	16	EBS only	1	\$0.0
<input type="checkbox"/> m4.2xlarge	8	32	EBS only	1	\$0.0
<input type="checkbox"/> m4.4xlarge	16	64	EBS only	1	\$0.0
<input type="checkbox"/> m4.10xlarge	40	160	EBS only	1	\$0.0
view more					

Configure the Spot instance

Our next step is to configure the instance, in this step of the wizard, we'll configure instance parameters like VPC, subnets, etc.

Let's take a look.

Step 1) Allocation Strategy – it determines how your spot request is fulfilled from the AWS's spot pools. There are two types of strategies:

- Diversified – here, spot instances are balanced across all the spot pools

- Lowest price – here, spot instances are launched from the pool which has lowest price offers

For this tutorial, we'll select Lowest Price as our allocation strategy.

Spot instance launch wizard

Step 1: Find instance types

Step 2: Configure

Step 3: Review

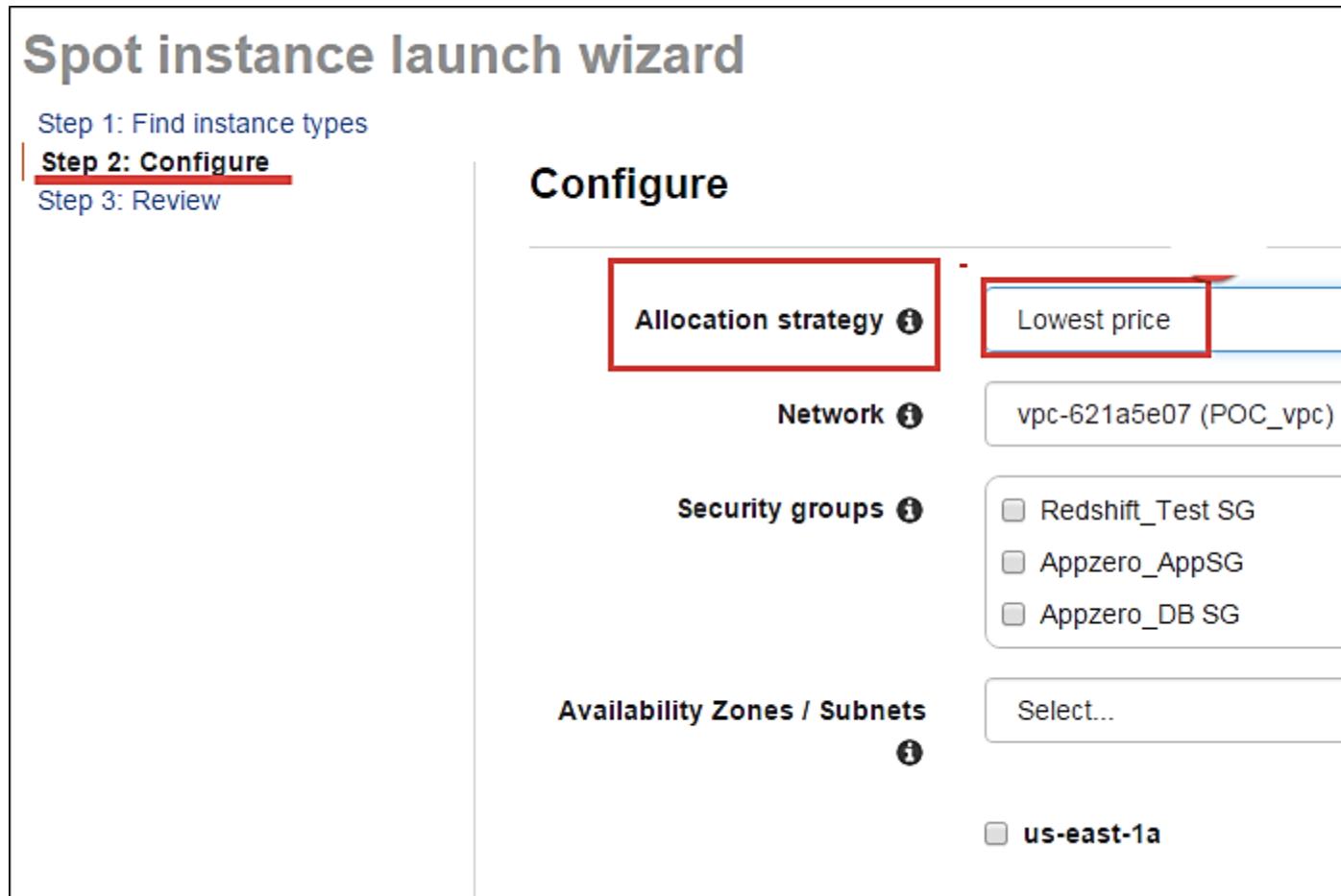
Configure

Allocation strategy i Lowest price

Network i vpc-621a5e07 (POC_vpc)

Security groups i Redshift_Test SG
 Appzero_AppSG
 Appzero_DB SG

Availability Zones / Subnets i Select...
 us-east-1a



Step 2) Select the VPC- we'll select from the list of available VPCs that we have created earlier. We can also create a new VPC in this step.

Spot instance launch wizard

Step 1: Find instance types

Step 2: Configure

Step 3: Review

Configure

Allocation strategy ⓘ

Lowest price

Network ⓘ

vpc-d5194fb0 (Prachi_Tes...
vpc-621a5e07 (POC_vpc)
vpc-d5194fb0 (Prachi_Tes...
vpc-84520ce0 (POC_vpc)
vpc-823e39e7 (TVPC)
vpc-4c51bf28 (POC_vpc3)
EC2-Classic

Security groups ⓘ

Availability Zones / Subnets

Select...

Step 3) Next we'll select the security group for the instance. We can select an already existing SG or create a new one.

Spot instance launch wizard

Step 1: Find instance types

Step 2: Configure

Step 3: Review

Configure

Allocation strategy i

Lowest price

Network i

vpc-d5194fb0 (Prachi_Tes

Security groups i

- Prachi_Public SG
- Web Server SG
- default
- launch-wizard-6

Availability Zones / Subnets

Select...

Step 4) Availability Zone- we'll select the AZ where we want to place our instance based on our application architecture. We are selecting AZ- us-east-1a.

Configure

Allocation strategy ⓘ

Lowest price

Network ⓘ

vpc-d5194fb0 (P)

Security groups ⓘ

- Prachi_Public
- Web Server S
- default
- launch-wizard-1

Availability Zones / Subnets ⓘ

Select...

us-east-1a

Step 5) Subnets- we are going to select the subnet from our list of already available list.

Configure

Allocation strategy i Lowest price

Network i vpc-d5194fb0 (Prachi_Test - VPC)

Security groups i

Prachi_Public SG
 Web Server SG
 default
 launch-wizard-6

Availability Zones / Subnets i

Select...

us-east-1a Subnet: i (Select...)

us-east-1b (Select...)

us-east-1d subnet-0eeef779 (Prachi_Test)

us-east-1e subnet-a94427de (Prachi_Test)

Step 6) Public IP- we'll choose to assign the instance a public IP as soon as it launches. In this step, you can choose if you want AWS to assign it an IP automatically, or you want to do it manually later. You can enable/ disable 'Auto assign Public IP' feature here likewise.

Availability Zones / Subnets

Select... ▾

us-east-1a Subnet: ⓘ subnet-a944 ▾

us-east-1b

us-east-1d

us-east-1e

Public IP ⓘ auto-assign at launch

Step 7) Key pair- A key pair is a set of public-private keys.

AWS stores the private key in the instance, and you are asked to download the private key. Make sure you download the key and keep it safe and secured; **if it is lost you cannot download it again.**

After selecting public IP, here we are selecting a key which we already have created in our last tutorial.

Availability Zones / Subnets

Select... ▾

us-east-1a Subnet: ⓘ subnet-a944 ▾

us-east-1b

us-east-1d

us-east-1e

Public IP ⓘ auto-assign at launch

Key pair name ⓘ Dev Key

Review your Spot instance

Once we are done configuring our spot instance request in the 2 steps earlier in our wizard, we'll take a look at the overall configuration.

Spot instance launch wizard

Step 1: Find instance types
Step 2: Configure
Step 3: Review

Review

▼ Spot request

IAM role	arn:aws:iam::018...
AMI	ami-60b6c60a
Target capacity	1 instances
Bid price	\$0.01 per unit
Instance type(s)	m3.medium: 1 un...
Allocation strategy	lowestPrice

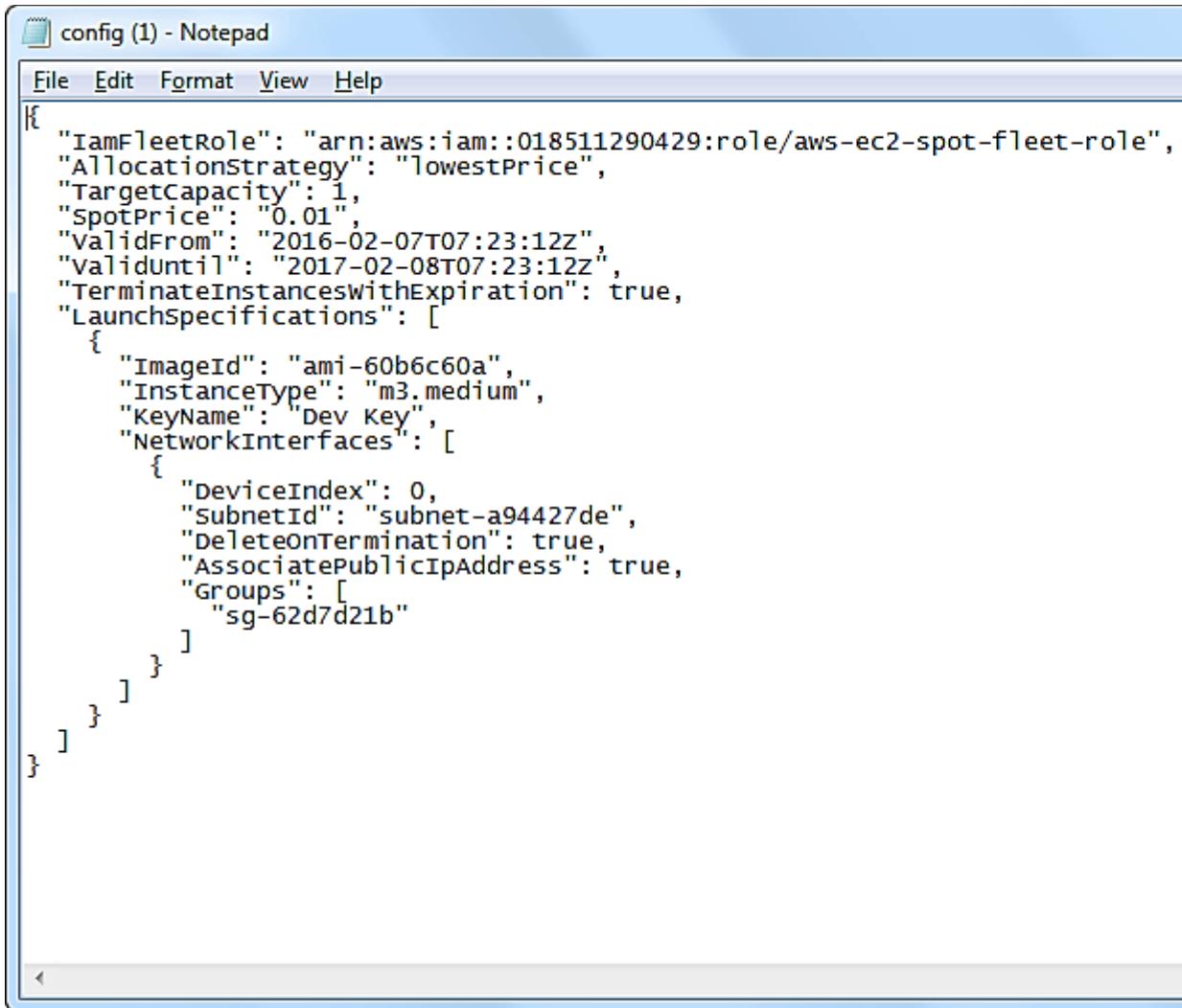
▼ Network

Network	vpc-d5194fb0
Security groups	sg-62d7d21b
Availability Zones / Subnets	us-east-1a (subn...

▼ Runtime duration

Request valid from	2/7/2016, 12:53:1...
Request valid until	2/8/2017, 12:53:1...

1. We can also download a JSON file with all the configurations. Below is our JSON file.



The screenshot shows a Windows Notepad window titled "config (1) - Notepad". The window contains the following JSON configuration code:

```
{  
    "IamFleetRole": "arn:aws:iam::018511290429:role/aws-ec2-spot-fleet-role",  
    "AllocationStrategy": "lowestPrice",  
    "TargetCapacity": 1,  
    "SpotPrice": "0.01",  
    "ValidFrom": "2016-02-07T07:23:12Z",  
    "ValidUntil": "2017-02-08T07:23:12Z",  
    "TerminateInstancesWithExpiration": true,  
    "LaunchSpecifications": [  
        {  
            "ImageId": "ami-60b6c60a",  
            "InstanceType": "m3.medium",  
            "KeyName": "Dev Key",  
            "NetworkInterfaces": [  
                {  
                    "DeviceIndex": 0,  
                    "SubnetId": "subnet-a94427de",  
                    "DeleteOnTermination": true,  
                    "AssociatePublicIpAddress": true,  
                    "Groups": [  
                        "sg-62d7d21b"  
                    ]  
                }  
            ]  
        }  
    ]  
}
```

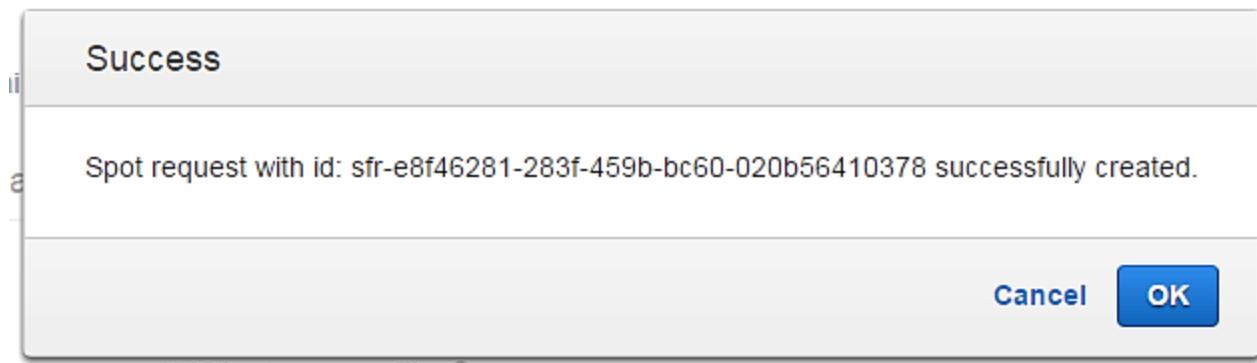
After we are done reviewing, we can proceed with the launching by clicking the Launch button as shown below.

Terminate instances with expiration i yes

▼ Instance details

User data i	
Key pair name i	Dev Key
IAM instance profile i	
Attach instance store i	no
EBS-optimized i	no
Monitoring enabled i	no

Once we select Launch, we can see a notification about the request getting created.



The spot request creation wizard will close, and the page will automatically direct back to the EC2 Dashboard.

You can see as shown below that the State of our request is 'open' which means that it is getting evaluated from the AWS's side. AWS EC2 will check if the required instance is available in its spot pool.

The screenshot shows the AWS EC2 Dashboard with the 'Request Spot Instances' interface open. On the left, a sidebar lists various EC2 services: EC2 Dashboard, Events, Tags, Reports, Limits, INSTANCES (with sub-options: Instances, Spot Requests, Reserved Instances, Scheduled Instances, Commands, Dedicated Hosts), IMAGES (with sub-options: AMIs, Bundle Tasks), ELASTIC BLOCK STORE (with sub-options: Volumes, Snapshots), and NETWORK & SECURITY (with sub-options: Security Groups, Elastic IPs, Placement Groups, Key Pairs). The 'Spot Requests' section is currently selected. The main panel displays a table of spot requests with columns: Name, Request ID, Max Price, and AMI ID. One row, 'sir-02g155kn', is highlighted with a red box. Below the table, a detailed view for 'Spot Request: sir-02g155kn' is shown, divided into 'Description' and 'Tags' tabs. The 'Description' tab contains the following configuration parameters:

Created	February 8, 2016 at 2:18:43 PM UTC+5:30
AMI ID	ami-60b6c60a
Product description	Linux/UNIX
Instance type	m3.medium
Availability Zone	-
Subnet	subnet-b3e3d0ea
IAM Role	-
Monitoring enabled	false
Security group(s)	default
Key pair name	Dev Key
RAM disk ID	-

After a couple of minutes, you can see that the state is changed to 'active', and now our spot request is successfully fulfilled. You can note the configuration parameters below.

	Name	Request ID	Max Price	AMI ID	Instance Type	Instance State
<input type="checkbox"/>		sir-02g6r657	\$0.01	ami-60b6c60a	i-53b6eed6	m3.medium
<input checked="" type="checkbox"/>		sir-02g155kn	\$0.05	ami-60b6c60a	i-cf967c7d	m3.medium

Spot Request: sir-02g155kn

Summary:

Thus, we saw in detail how to create an on-demand EC2 instance in this tutorial. Because it is an on-demand server, you can keep it running when in use and 'Stop' it when it's unused to save on your costs.

You can provision a Linux or Windows EC2 instance or from any of the available AMIs in AWS Marketplace based on your choice of OS platform.

If your application is in production and you have to use it for years to come, you should consider provisioning a reserved instance to drastically save on your CAPEX.

Here, we saw how to create a Spot Instance request successfully by determining our bid price.

Spot instances are a great way to save on costs for instances which are not application critical. A common example would be to create a fleet of spot instances for a task such as image processing or video encoding. In such cases, you can keep a cluster of instances under a load balancer.

If the bid price exceeds the spot price and your instance is terminated from AWS's side, you can have other instances doing the processing

job for you. You can leverage Auto scaling for this scenario. Avoid using Spot instances for business critical applications like databases etc.

AWS Certified Solutions Architect - Associate 2018

Change Instance Type, Security Group, Termination Protection: AWS EC2

EC2 stands for Elastic Compute Cloud. It is the compute service offering from the IaaS (Infrastructure as a Service) area of AWS.

Once an EC2 instance is provisioned, it is very handy to update/modify many of the instances configuration parameters using AWS Management Console.

Let's take a look at each of them.

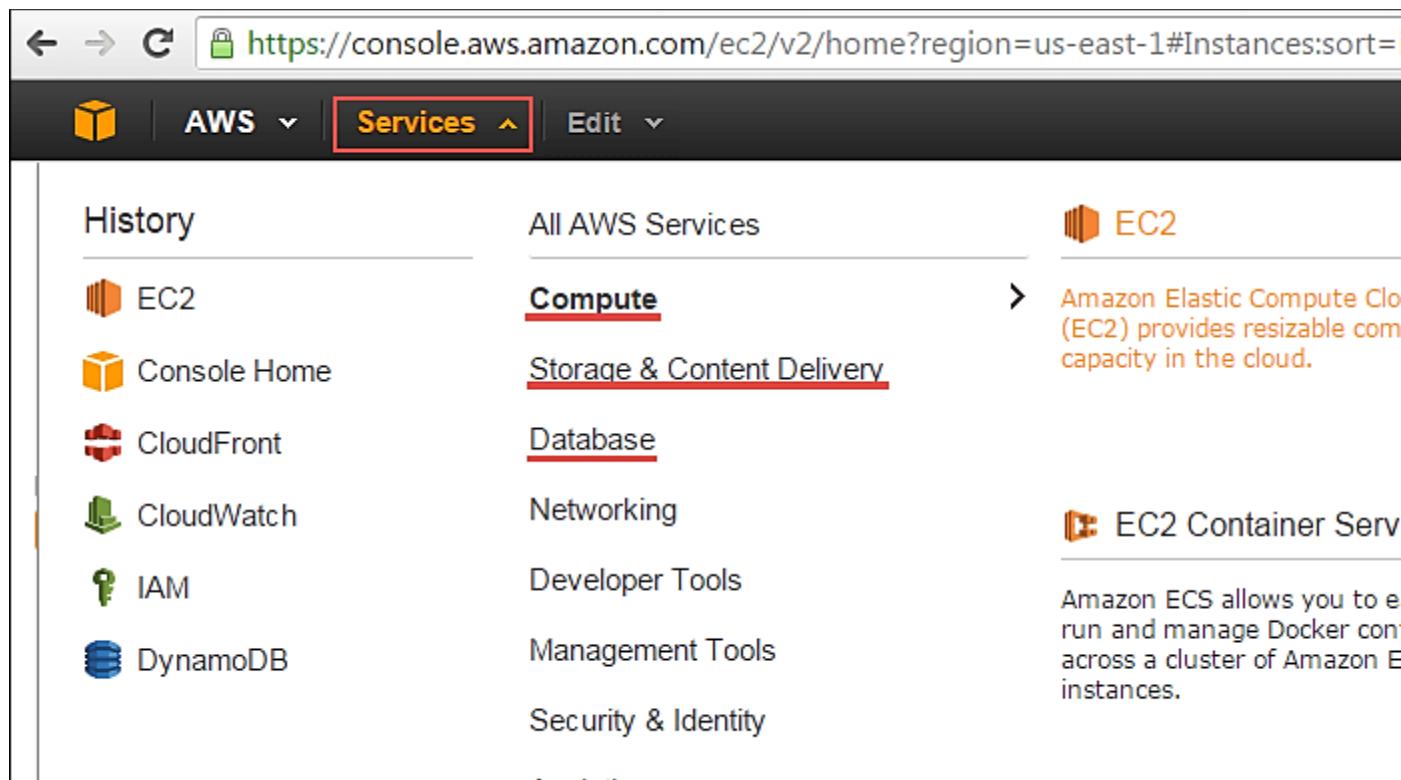
What You Will Learn: [hide]

- [Login and access to AWS services](#)
 - [Check the modification parameters](#)
 - [View the connection details](#)
 - [Launch multiple instances with the similar configuration](#)
 - [Change the instance state](#)
- [Change instance settings](#)
 - [Create tags](#)
 - [Attach to Auto Scaling Group](#)
 - [Change instance type](#)
 - [Enable termination protection](#)
 - [Change User Data](#)
 - [Change the shutdown behavior](#)
 - [View System Log](#)
- [Create an instance AMI](#)
- [Change the instance network settings](#)
 - [Change the Security Group](#)
 - [Add a Network Interface](#)
 - [Dissociating EIP](#)
 - [Change Source/Destination check](#)
 - [Manage private IP addresses](#)
 - [Enable/disable ClassicLink to a VPC](#)
 - [Enable detailed CloudWatch monitoring](#)

Login and access to AWS services

Step 1) In this step, you will do

- Login to your AWS account and go to the AWS Services tab at the top left corner.
- Here, you will see all of the AWS Services categorized as per their area viz. Compute, Storage, Database, etc. For creating an EC2 instance, we have to choose Compute & EC2 as in the next step.



Open all the services and click on EC2 under Compute services. This will launch the dashboard of EC2.

Here is the EC2 dashboard. Here you will get all the information in gist about the AWS EC2 resources running.

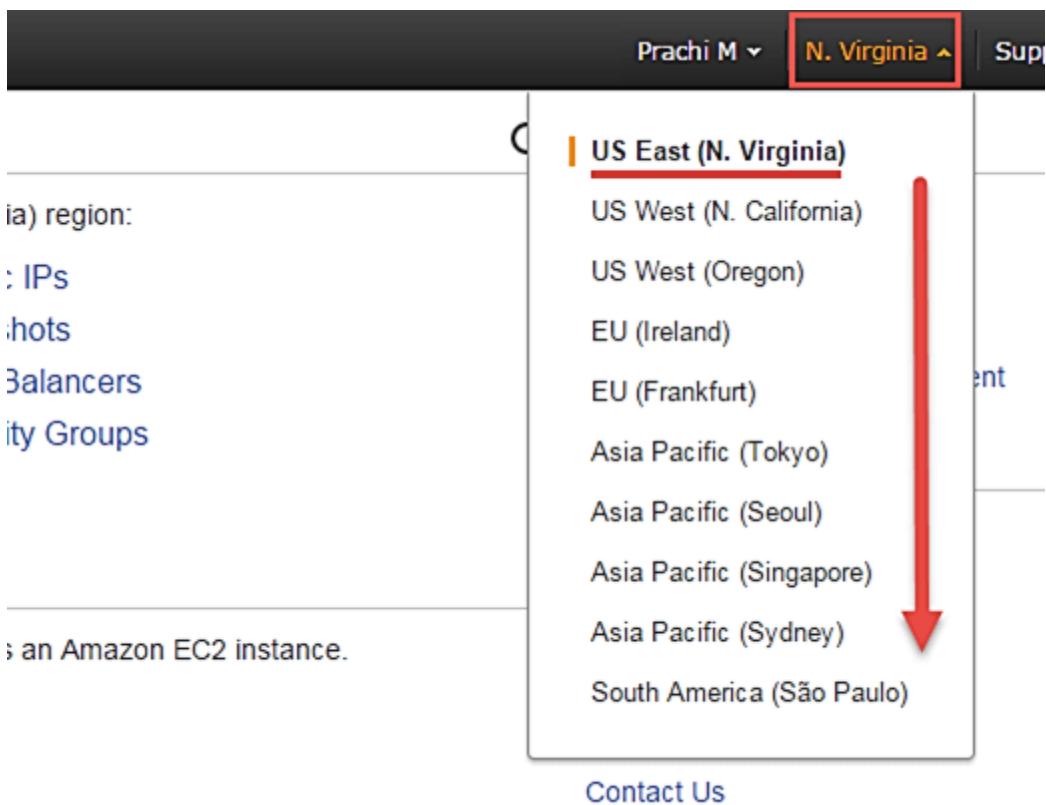
The screenshot shows the AWS EC2 Dashboard. At the top, there are navigation links: AWS, Services, and Edit. On the left, a sidebar menu includes: Events, Tags, Reports, Limits, INSTANCES (with sub-options: Instances, Spot Requests, Reserved Instances, Scheduled Instances, Commands, Dedicated Hosts), IMAGES (with sub-options: AMIs, Bundle Tasks), and ELASTIC BLOCK STORE (with sub-options: Volumes, Snapshots). A red arrow points from the 'INSTANCES' section towards the main content area. The main content area is titled 'Resources' and displays the following information: You are using the following Amazon EC2 resources in the US East (N. Virginia) region. The resource summary table is as follows:

3	Running Instances
0	Dedicated Hosts
12	Volumes
22	Key Pairs
0	Placement Groups

Below this, there is a promotional message: "Need fast, reliable, scalable, fully-managed message queuing? Try Amazon Simple Queue Service". The next section is titled "Create Instance" with a large blue "Launch Instance" button. A note below the button states: "Note: Your instances will launch in the US East (N. Virginia) region".

Step 2) On the top right corner of the EC2 dashboard, choose the AWS Region in which you want to provision the EC2 server.

Here we are selecting N. Virginia. AWS provides 10 Regions all over the globe.

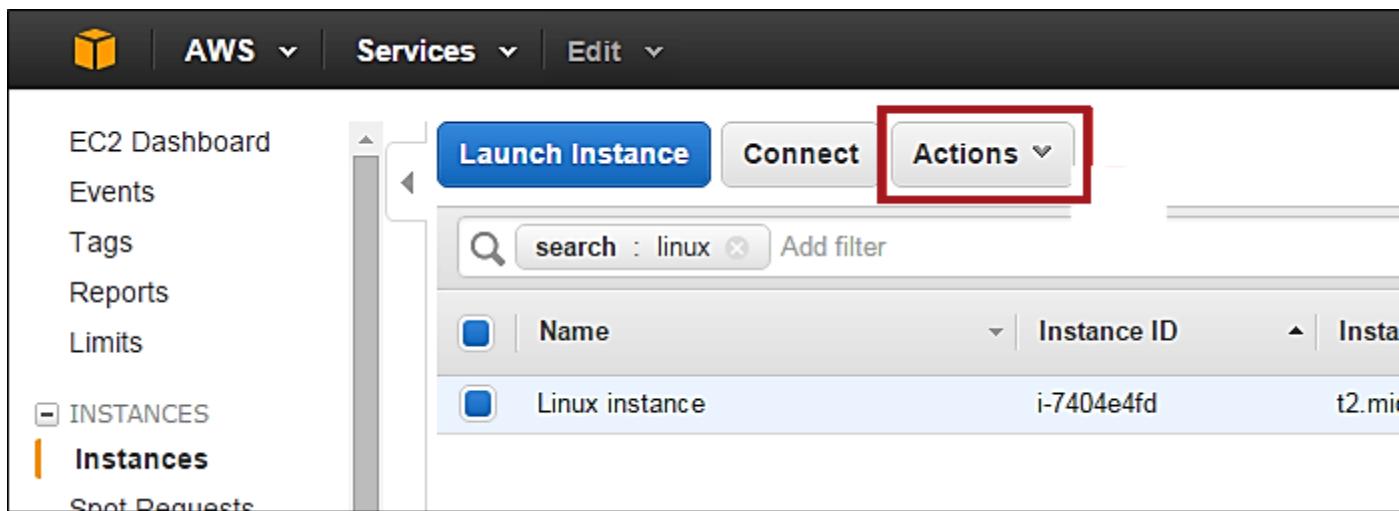


Step 3)

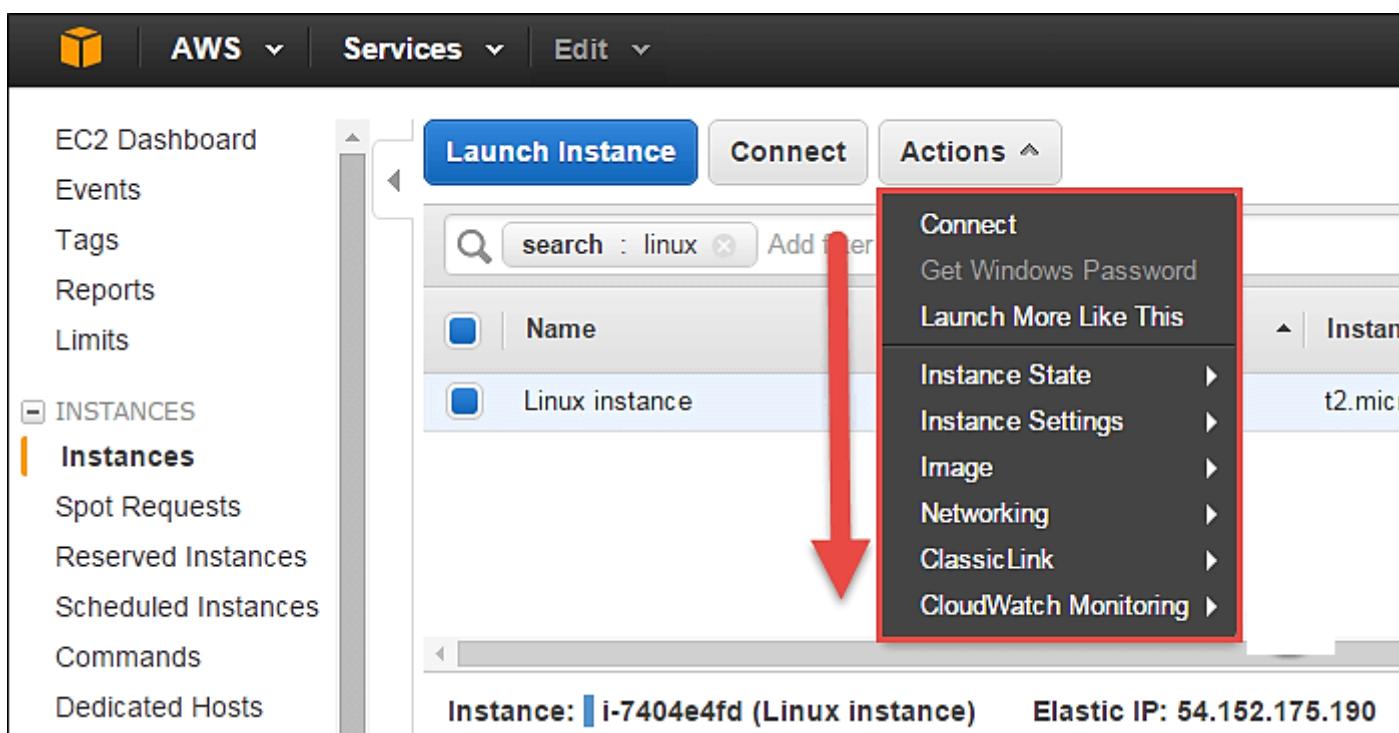
Once your desired Region is selected, come back to the EC2 Dashboard.

Check the modification parameters

Step 1) On the EC2 Dashboard, select the instance whose configuration parameters you want to modify and Click on the "Actions" button as shown below.



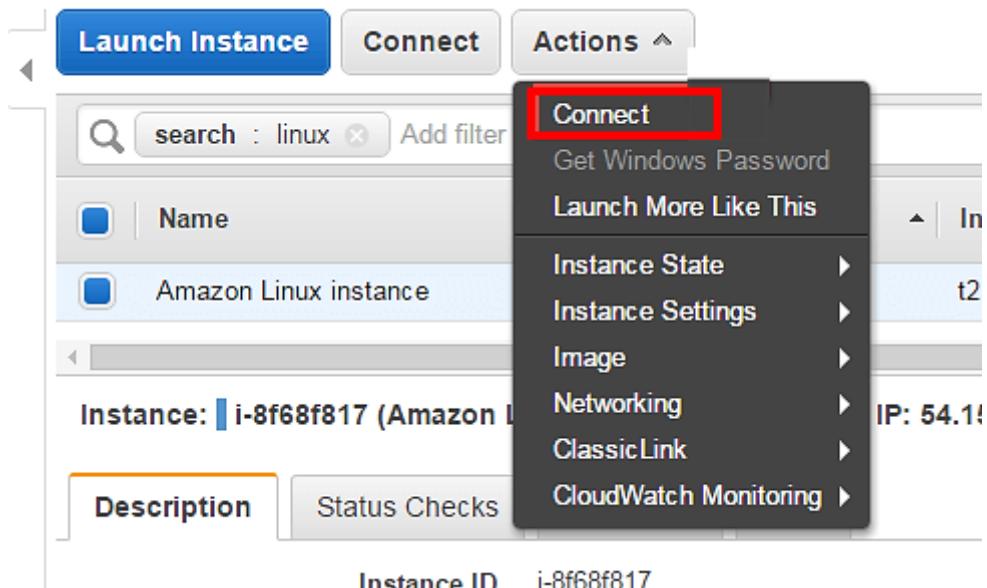
Step 2) As you click the button, the drop- down will show us all the areas where we can modify the instance characteristics.



View the connection details

Connect option below will show us ways in which we can connect to an EC2 instance.

Step 1) Click on option 'Connect.'



You may choose to connect with a standalone SSH client or a [Java](#) client. You will get a step-by-step procedure on how you can connect to your instance.

For this tutorial, we can see the connection methods for a [Linux](#) instance.

Connect To Your Instance

- I would like to connect with A standalone SSH client A Java SSH Client directly from my browser (Java required)

To access your instance:

1. Open an SSH client. (find out how to [connect using PuTTY](#))
2. Locate your private key file (Prachi_Jump server.pem). The wizard automatically detects the key you used to launch the instance.
3. Your key must not be publicly viewable for SSH to work. Use this command if needed:

```
chmod 400 Prachi_Jump server.pem
```

4. Connect to your instance using its Public DNS:

`ec2-54-152-175-190.compute-1.amazonaws.com`

Example:

```
ssh -i "Prachi_Jump server.pem" ec2-user@ec2-54-152-175-190.compute-1.amazonaws.com
```

Please note that in most cases the username above will be correct, however please ensure that you read your AMI usage instructions to ensure that the AMI owner has not changed the default AMI username.

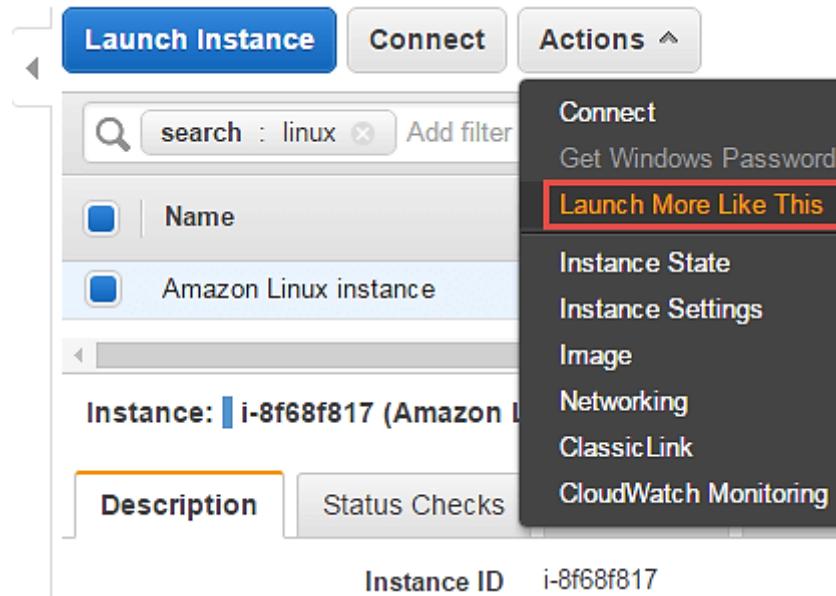
If you need any assistance connecting to your instance, please see our [connection documentation](#).

Close

Launch multiple instances with the similar configuration

If you have a single EC2 instance running with a particular configuration, and you wish to quickly launch another instance in a one-click deployment, then 'Launch More Like This' option helps us do that.

Step 1) Click on 'Launch More Like This.'



You will be straightaway directed to the review instance details page of the launch instance wizard. Here we can verify all the details once more.

Step 2) On review instance details page Click on button 'Launch.'

1. Choose AMI

2. Choose Instance Type

3. Configure Instance

4. Add Storage

5. Tag Instance

6. C

Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch Instance** to start the launch process.



⚠ Improve your instances' security. Your security group, Prachi_Public SG, is currently set to allow traffic from anywhere.

Your instances may be accessible from any IP address. We recommend that you update your security group to restrict access to your instances.

You can also open additional ports in your security group to facilitate access to the application or service running on your instances.

AMI Details



amzn-ami-hvm-2015.09.1.x86_64-gp2 - ami-60b6c60a

Amazon Linux AMI 2015.09.1 x86_64 HVM GP2

Root Device Type: ebs Virtualization type: hvm

Instance Type

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GiB)
t2.micro	Variable	1	1	EBS only

Security Groups

Step 3) In this window,

1. Select an existing key pair
2. Click on "Launch Instance."

Select an existing key pair or create a new key pair

X

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

Dev Key

1

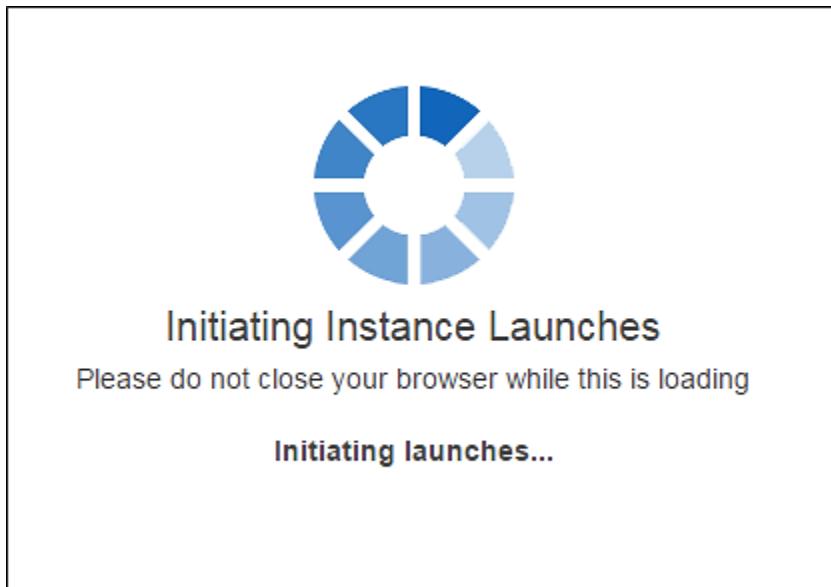
I acknowledge that I have access to the selected private key file (Dev Key.pem), and that without this file, I won't be able to log into my instance.

2

Cancel

Launch Instances

Instance launch progress can be seen as below.



You can see below that a new instance is in a pending state before creation.

Screenshot of the AWS CloudFormation console showing the details of a new instance.

The top navigation bar includes "Launch Instance", "Connect", and "Actions".

The search bar shows "search : i-93da490b" and an "Add filter" button.

The table header includes columns for Name, Instance ID, Instance Type, and Availability Zone.

A row in the table is highlighted with a red border, showing:

Name	Instance ID	Instance Type	Availability Zone
Amazon Linux instance	i-93da490b	t2.micro	us-east-1b

Below the table, the instance details are displayed:

Instance: i-93da490b (Amazon Linux instance) Public DNS: ec2-52-90-225-238.compute-1.amazonaws.com

Description tab is selected.

Attribute	Value
Instance ID	i-93da490b
Instance state	pending
Instance type	t2.micro
Private DNS	ip-192-168-2-189.ec2.internal
Private IPs	192.168.2.189
Secondary private IPs	
VPC ID	vpc-d5194fb0
Subnet ID	subnet-b3e3d0ea
Network interfaces	eth0
Source/dest. check	True
ClassicLink	-
EBS-optimized	False

You can see that the new instance has the same tag as well.

Name	Instance ID	Instance Type	Availability Zone
Amazon Linux instance	i-8f68f817	t2.micro	us-east-1b
Amazon Linux instance	i-93da490b	t2.micro	us-east-1b

Change the instance state

You can change the instance state on the fly from the Management Console on a single click.

Step 1) In this step, Click on 'Instance State' under actions.

- Stop – you can stop the running instance
- Reboot – you can reboot the instance
- Terminate – you can delete the instance permanently

Name	Instance Type	Availability Zone
Linux instance		us-east-1a

Instance: i-7404e4fd (Linux instance) Elastic IP: 54.152.175.190

Change instance settings

Here you can change a lot of instance settings like security groups, termination protection, etc.

Let's see each one in detail.

Create tags

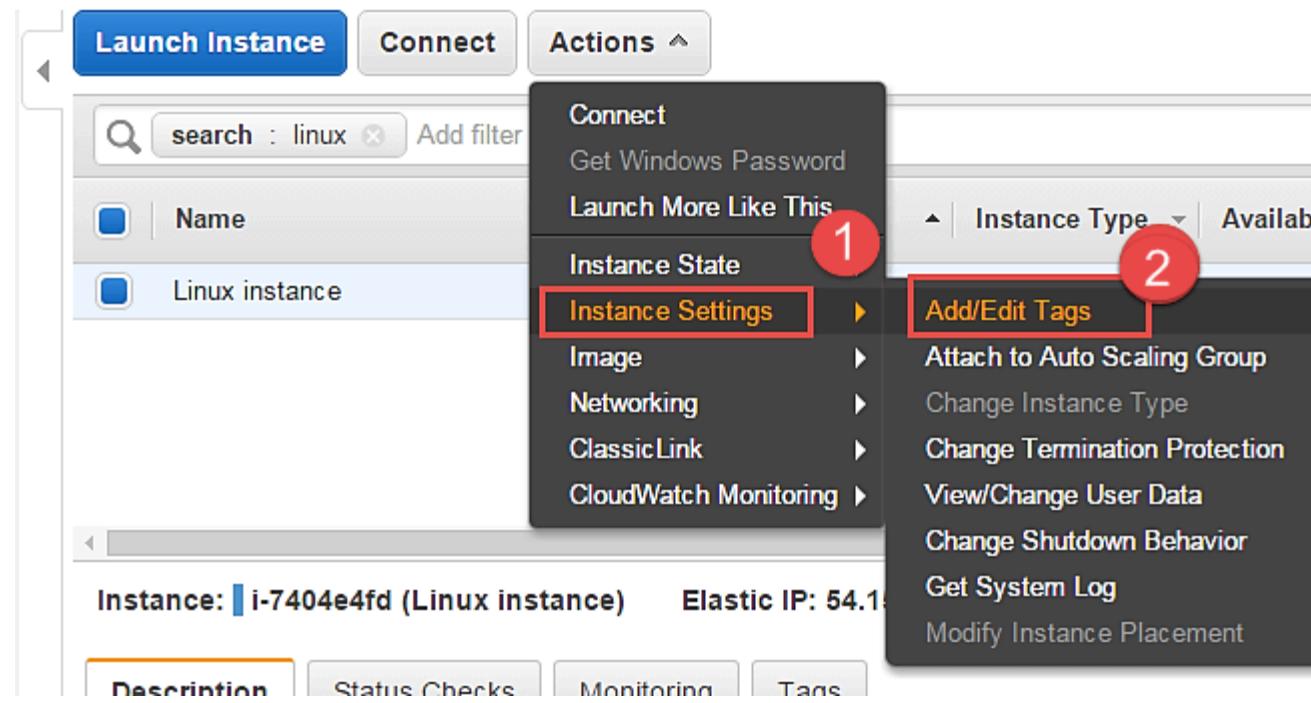
Add/Edit Tags – You can add or edit the tags assigned to the instance. Tagging makes it easier for the business owner of the AWS account to keep a track of the instances especially if there are multiple environments.

AWS admins should assign each instance a tag based on the segregation e.g.: tagging all the instances in the production environment as 'Prod' or tagging the instances belonging to a department with the department initials etc. Tagging is a very effective method to track the costing of the instances as well.

Let's see how to change tags

Step 1) In this step,

1. Click on instance setting
2. Click on 'Add/Edit Tags.'



Step 2) A tag is just a key-value pair.

1. So we have assigned a new tag as Department and added its value as Cloud.
2. Click on Save

The screenshot shows the 'Add/Edit Tags' dialog box. The title bar says 'Add/Edit Tags' and has a close button. Below it is a message: 'Apply tags to your resources to help organize and identify them.' A note explains that a tag consists of a case-sensitive key-value pair. The dialog contains a table with two rows:

Key	Value
Name	Linux instance
Department	Cloud

At the bottom left is a 'Create Tag' button. On the right side, there are 'Cancel' and 'Save' buttons. The 'Save' button is highlighted with a red box and circled '2'. The 'Department' row is also highlighted with a red box and circled '1'.

Step 3) Come back to the EC2 Dashboard and

1. Select your instance again
2. Select the tab of 'Tags'

Note that the new tag as "Department" with value as Cloud has appeared under Tags.

The screenshot shows the AWS EC2 Instance Details page. At the top, there are buttons for 'Launch Instance', 'Connect', and 'Actions'. Below that is a search bar with 'search : linux' and an 'Add filter' button. A table lists instances based on Name, Instance ID, Instance Type, and Availability Zone. One instance, 'Linux instance' (Instance ID i-7404e4fd), is highlighted with a red box and a red circle with the number 1. Below the table, the instance details are shown: 'Instance: i-7404e4fd (Linux instance)' and 'Elastic IP: 54.152.175.190'. Underneath, there are tabs for 'Description', 'Status Checks', 'Monitoring', and 'Tags'. The 'Tags' tab is selected and highlighted with a red box and a red circle with the number 2. An 'Add/Edit Tags' button is visible. A table below shows existing tags: one tag named 'Name' with value 'Linux instance' and another tag named 'Department' with value 'Cloud', both highlighted with red boxes.

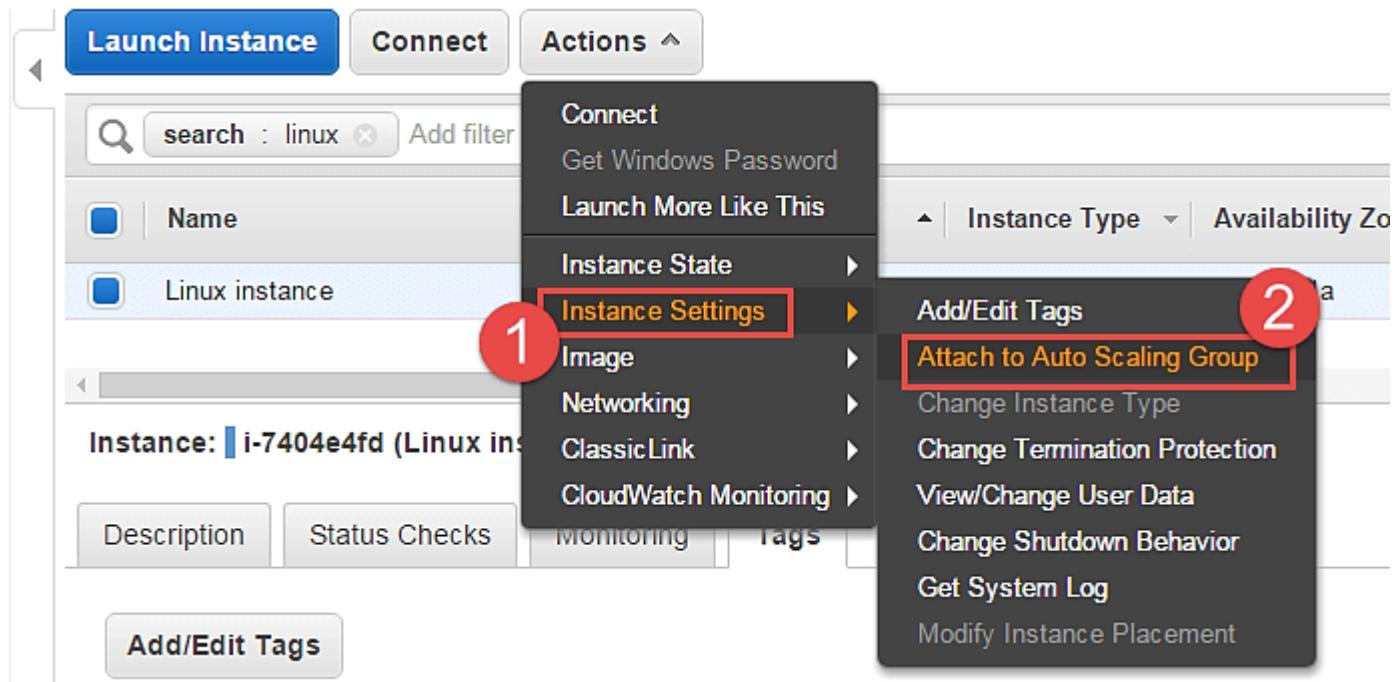
Key	Value
Name	Linux instance
Department	Cloud

Attach to Auto Scaling Group

An EC2 instance can be attached to an Auto Scaling Group on the fly.

Step 1) In this step, we do following things

1. Click on 'Instance Settings'
2. Click on 'Attach to Auto Scaling Group.'



Step 2) In this step,

1. Attach an instance to an existing AS group. You can also create a new AS group in this step.
2. Select one AS group from the list of already existing groups.
3. Click on 'Attach'.

This action will attach your instance to an auto-scaling group in your environment.

Attach to Auto Scaling Group

Attach an instance to:

- a new Auto Scaling group
- an existing Auto Scaling group

1

The screenshot shows a search interface for 'Auto Scaling Group'. The search bar contains 'Auto Scaling Group'. Below it is a table with two columns: 'Name' and 'Availability Zones'. One row is highlighted with a red box and a red number '2' over it, corresponding to the 'Test AS grp' entry. This row also has a blue checkbox next to it. The 'Availability Zones' column shows 'us-east-1a' for this specific group.

Name	Availability Zones
Test AS grp	us-east-1a

Proceeding with this action will:

- Add the instance to the group **Test AS grp**
- Automatically increment the desired capacity of the group **Test AS grp**

Are you sure you want to attach i-7404e4fd (ec2-54-152-175-190.compute-1.amazonaws.com)?

Cancel **Attach**

Change instance type

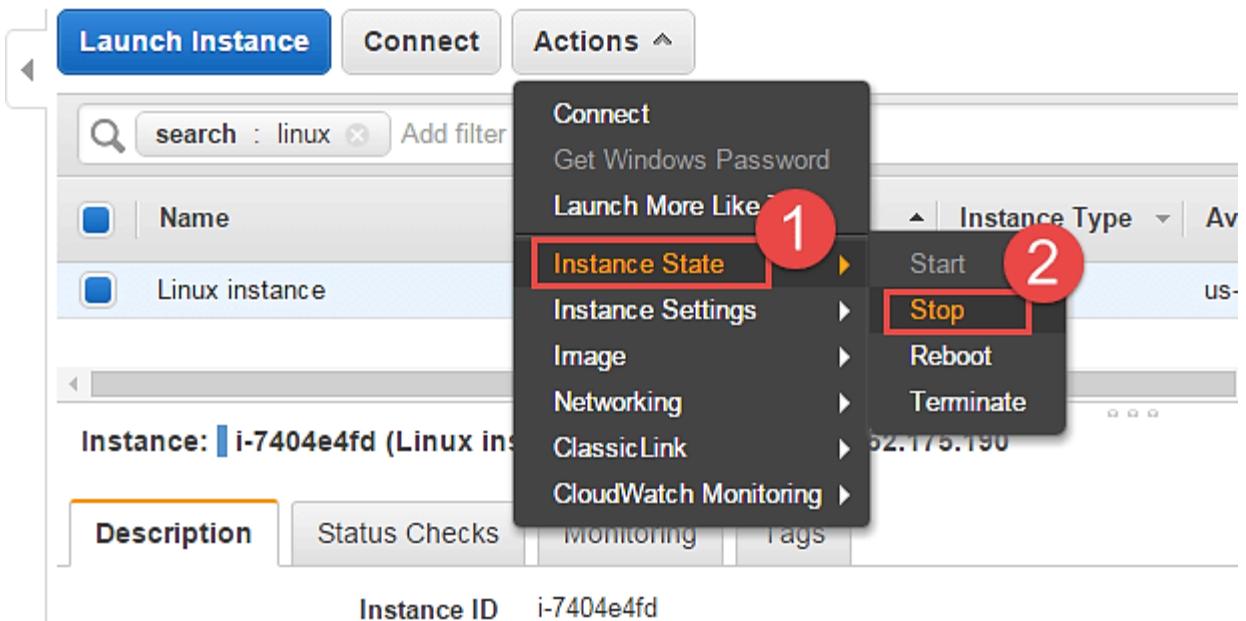
You can change the instance-type of your instance if you desire higher configuration instance as per your application requirement. This can be done to vertically scale your instance and provide you with more compute/memory capacity.

Let's see how to do this.

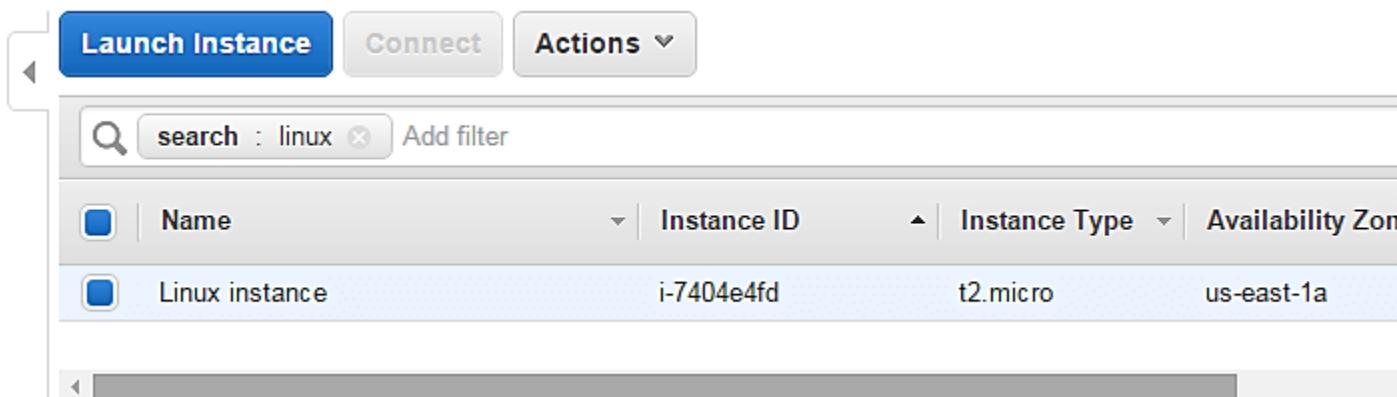
You cannot change an instance type if it's a running server. You have to stop it before doing so.

Step 1) In this step,

1. Go to 'InstanceState'
2. Click on 'Stop'. This will stop the instance.

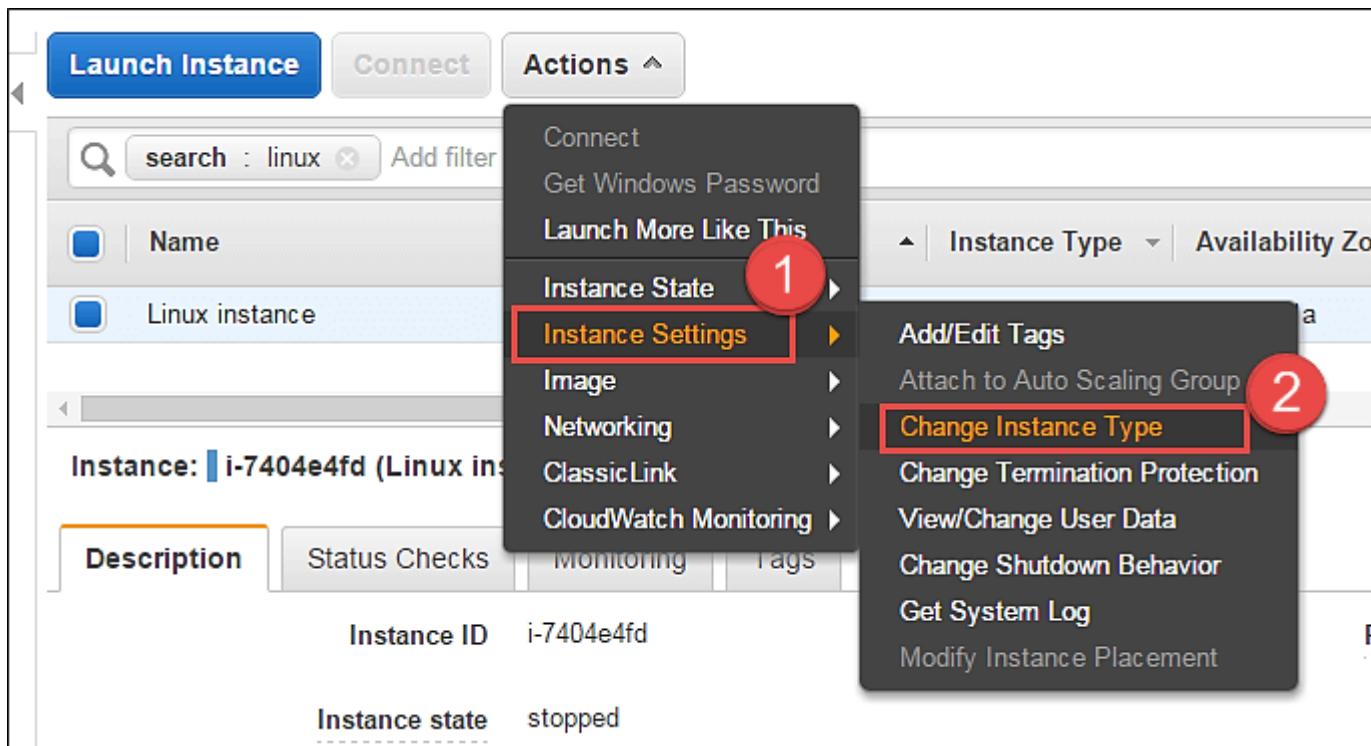


Note that the instance state is now in "stopping" mode on the EC2 Dashboard. You change an instance type now.

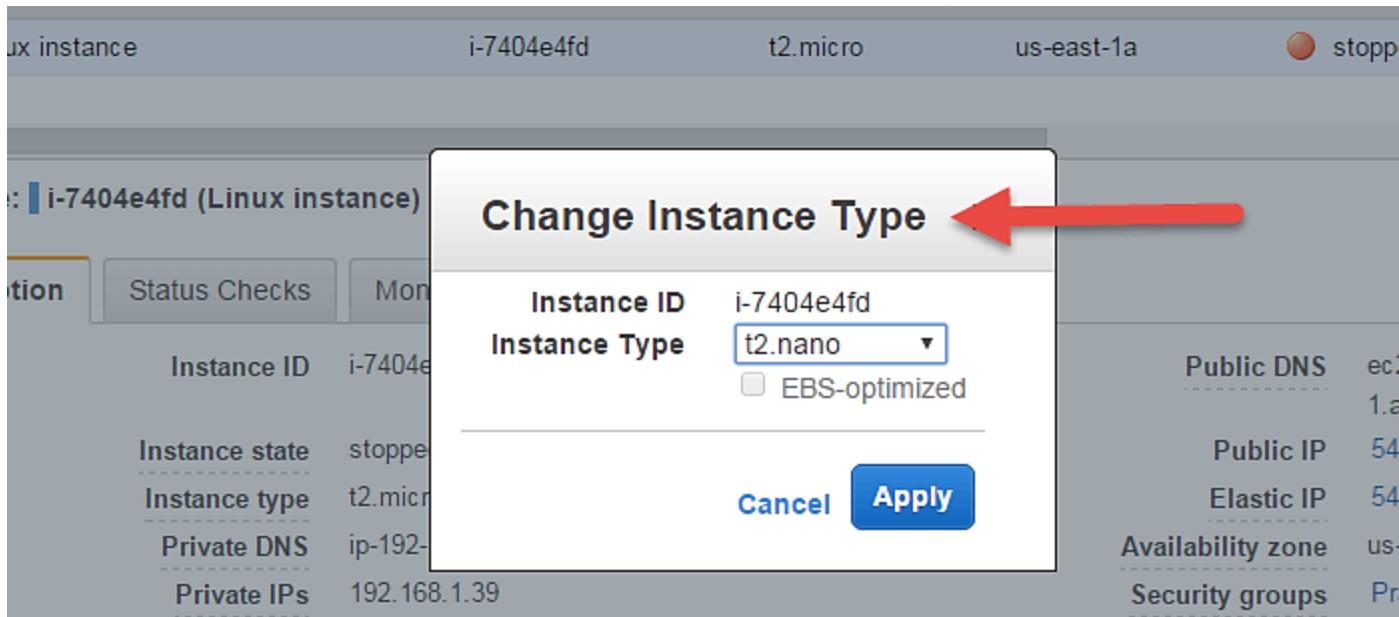


Step 2) In this step,

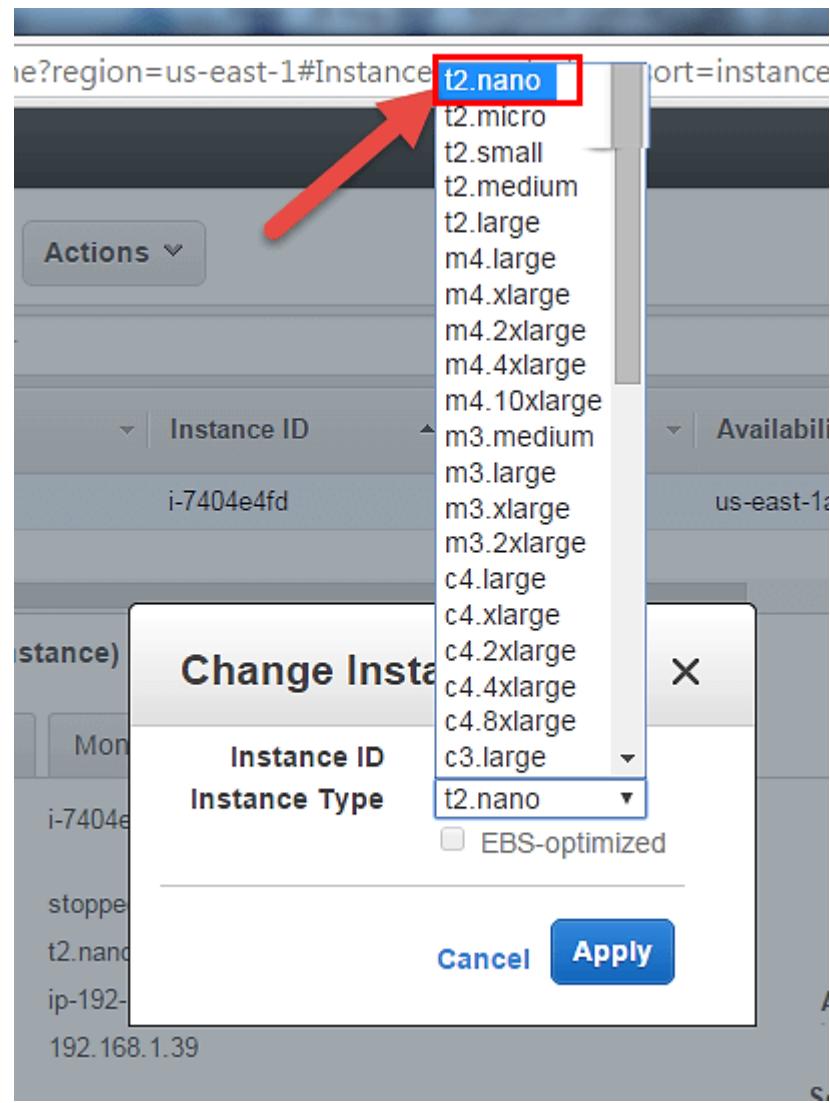
1. Go to 'Instance Settings'
2. Click on 'Change Instance Type'



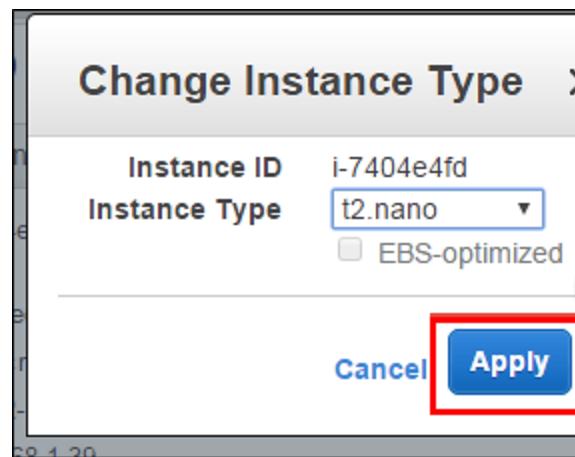
A Change Instance Type pop-up will appear.



Step 3) You can select from a range of EC2 available instance types. For this tutorial, we are changing it to t2.nano just for the sake of demonstration.



Step 4) Select t2.nano and hit 'Apply'.



Notice on the EC2 Dashboard, your instance type has been changed to the said type automatically.

The screenshot shows the AWS EC2 Instance Details page for an instance named 'Linux instance' with ID i-7404e4fd. The instance is currently stopped. The 'Instance type' field is highlighted with a red box and contains the value 't2.nano'. Other visible details include the instance state as 'stopped', private DNS as 'ip-192-168-1-39.ec2.internal', private IP as '192.168.1.39', and VPC ID as 'vpc-d5194fb0'. The page also includes tabs for Description, Status Checks, Monitoring, and Tags.

Name	Instance ID	Type
Linux instance	i-7404e4fd	t2.nano

Instance: i-7404e4fd (Linux instance) Elastic IP: 54.152.175.190

Description Status Checks Monitoring Tags

Instance ID: i-7404e4fd

Instance state: stopped
Instance type: t2.nano

Private DNS: ip-192-168-1-39.ec2.internal
Private IPs: 192.168.1.39

Secondary private IPs

VPC ID: vpc-d5194fb0

You can now start your instance and continue on the operations on that. There will be no change in other configuration parameters and also your existing installations on the server will remain intact.

Enable termination protection

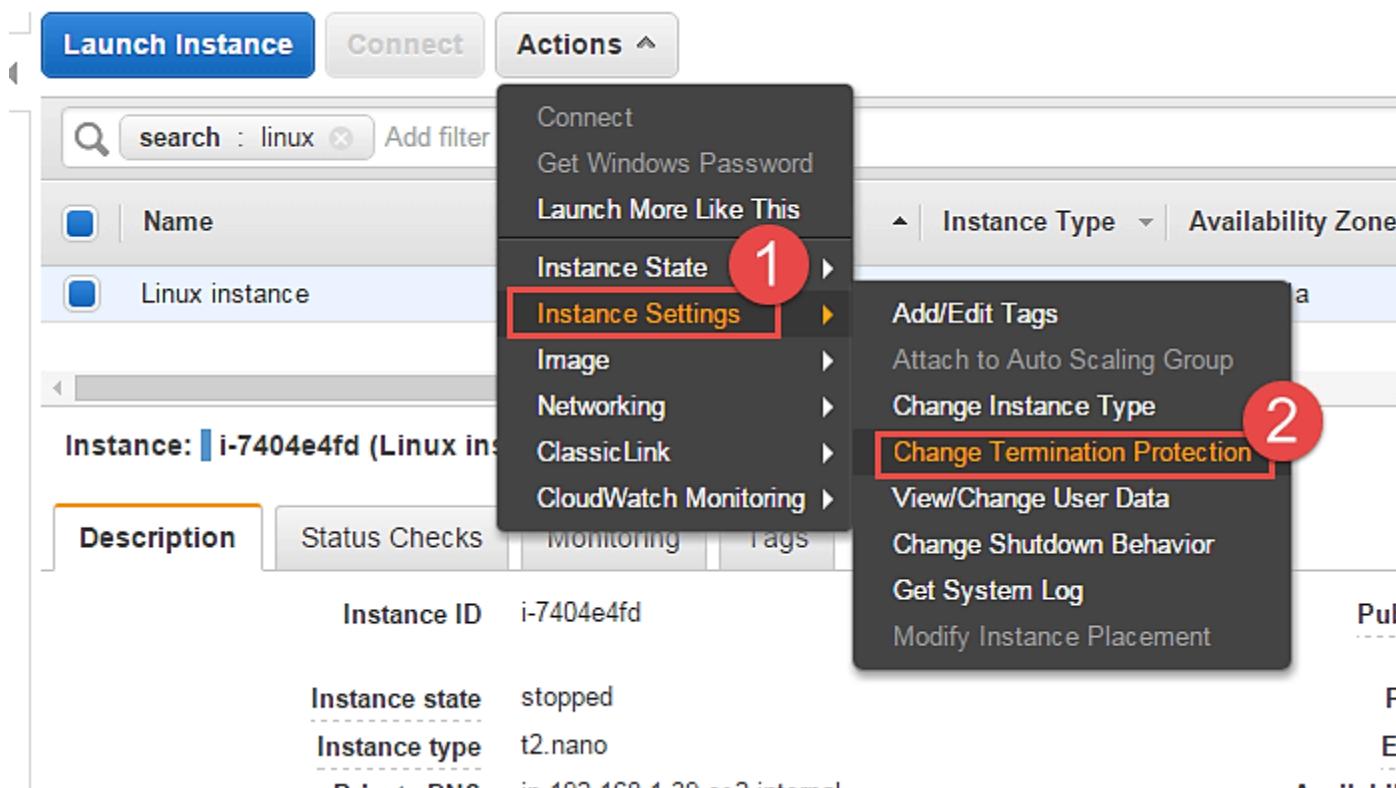
An instance should always have termination protection enabled especially on production servers. This will ensure that your EC2 instance is not getting accidentally terminated.

AWS will add an additional level of security in case you happen to accidentally hit the instance terminate option.

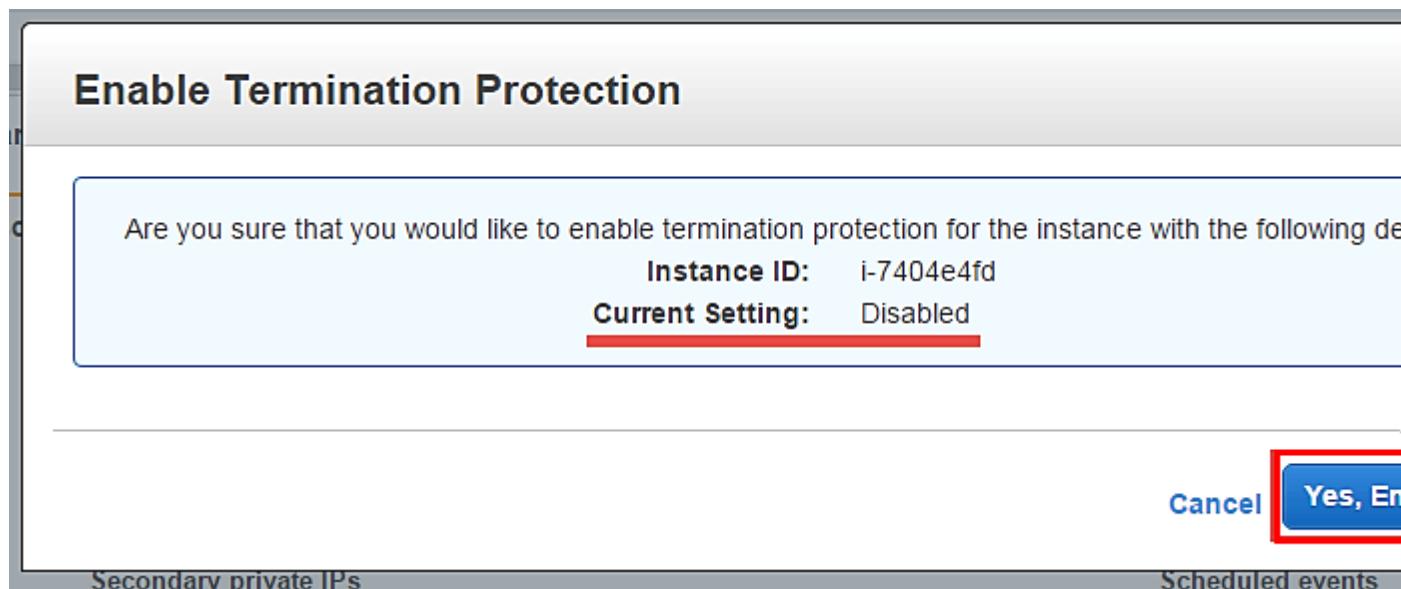
Let's see how to enable termination protection.

Step 1) In this step,

1. Go to 'Instance Settings.'
2. Click on 'Change Termination Protection.'



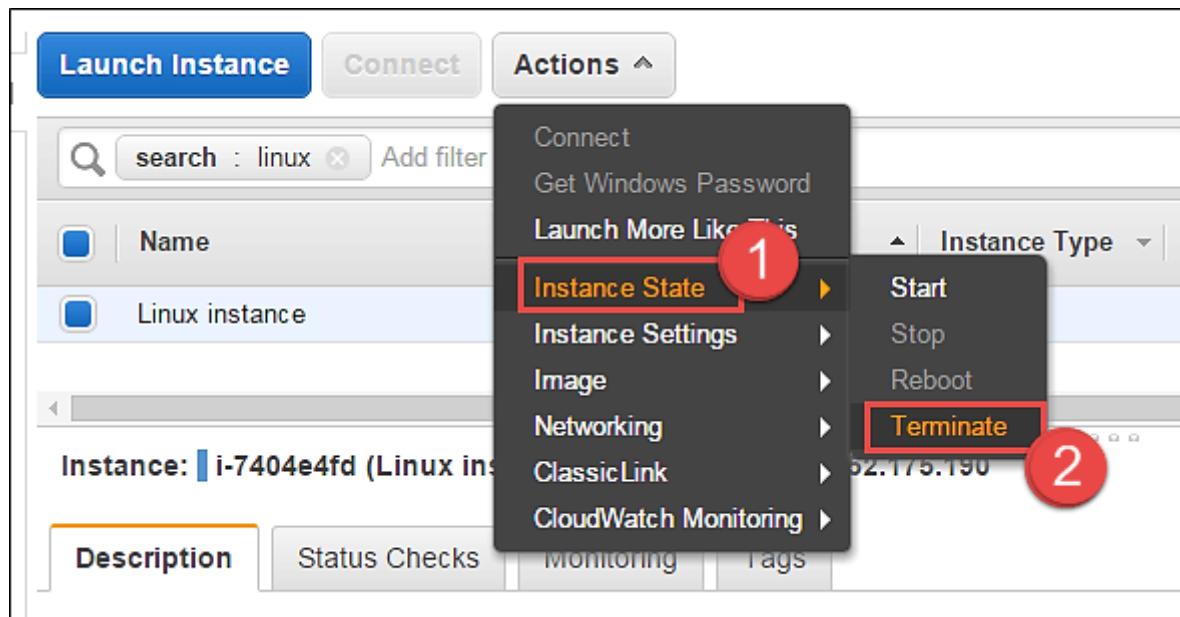
Step 2) Notice that the current setting on our instance is disabled. Click on "Yes,Enable".



This has enabled Termination protection on our instance. We'll check to see if our instance gets deleted when we hit Terminate.

Step 3) In this step,

1. Select option 'Instance State' and then
2. Click on 'Terminate.'



AWS will immediately notify you that the EC2 instance has "termination protection enabled" and you will not be able to delete it. The 'Terminate' button below is disabled.

Terminate Instances



Warning

On an EBS-backed instance, the default action is for the root EBS volume to be deleted when the instance is terminated. Storage on any local drives will be lost.

Are you sure you want to terminate these instances?

These instances have Termination Protection and will not be terminated. Use the Change Termination Protection option from the Instances screen Actions menu to allow termination of these instances.

- i-7404e4fd (Linux instance, ec2-54-152-175-190.compute-1.amazonaws.com)

Cancel

Yes, Terminat

Change User Data

When you launch a new EC2 instance, you have the option to pass user data to an instance to run tasks at boot time automatically e.g. common configuration tasks, init scripts, etc.

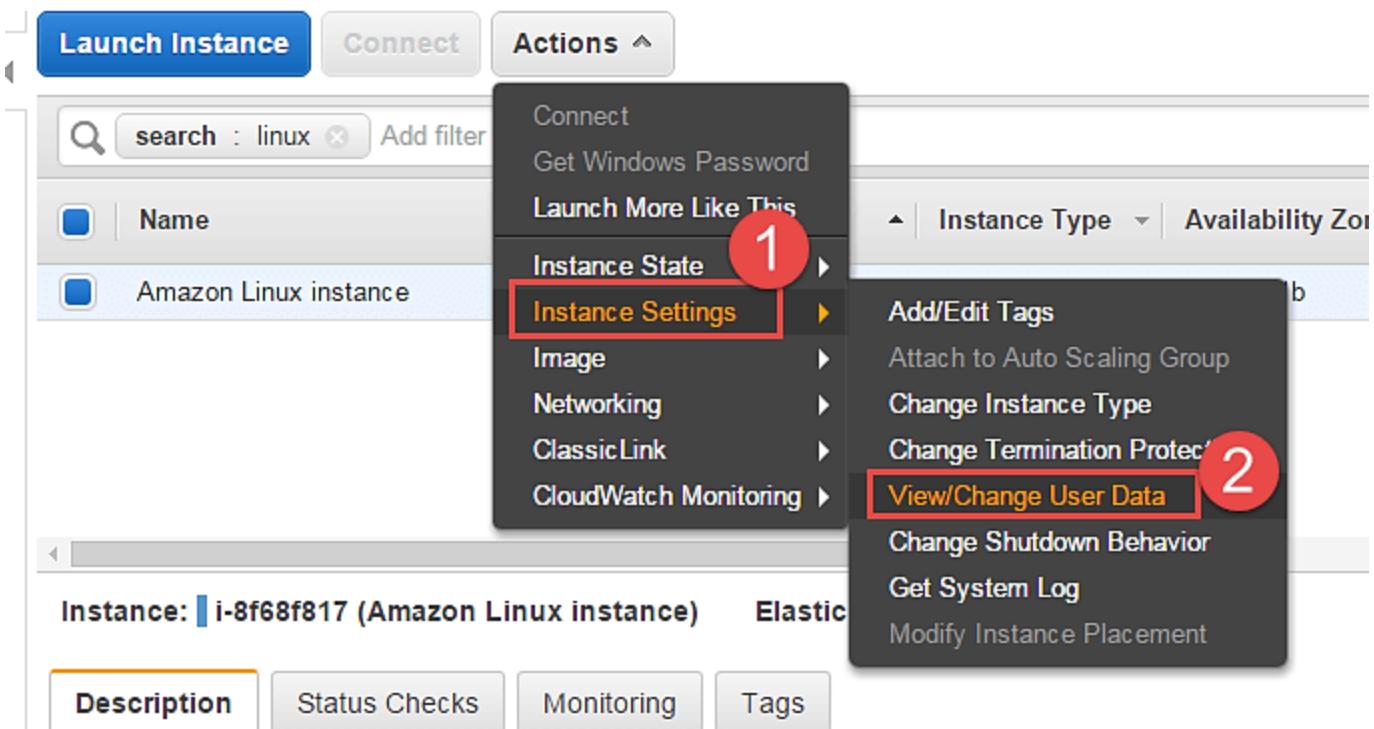
You can pass the user data in the form of shell scripts or cloud-init directives. This can be either plain text, as a file or as base64 encoded text for API calls.

Here we will see how we can edit these scripts.

You will have to stop the instance first, you will not be able to edit the instance userdata if it's running. On a stopped instance, perform below steps.

Step 1) In this step, do the following things

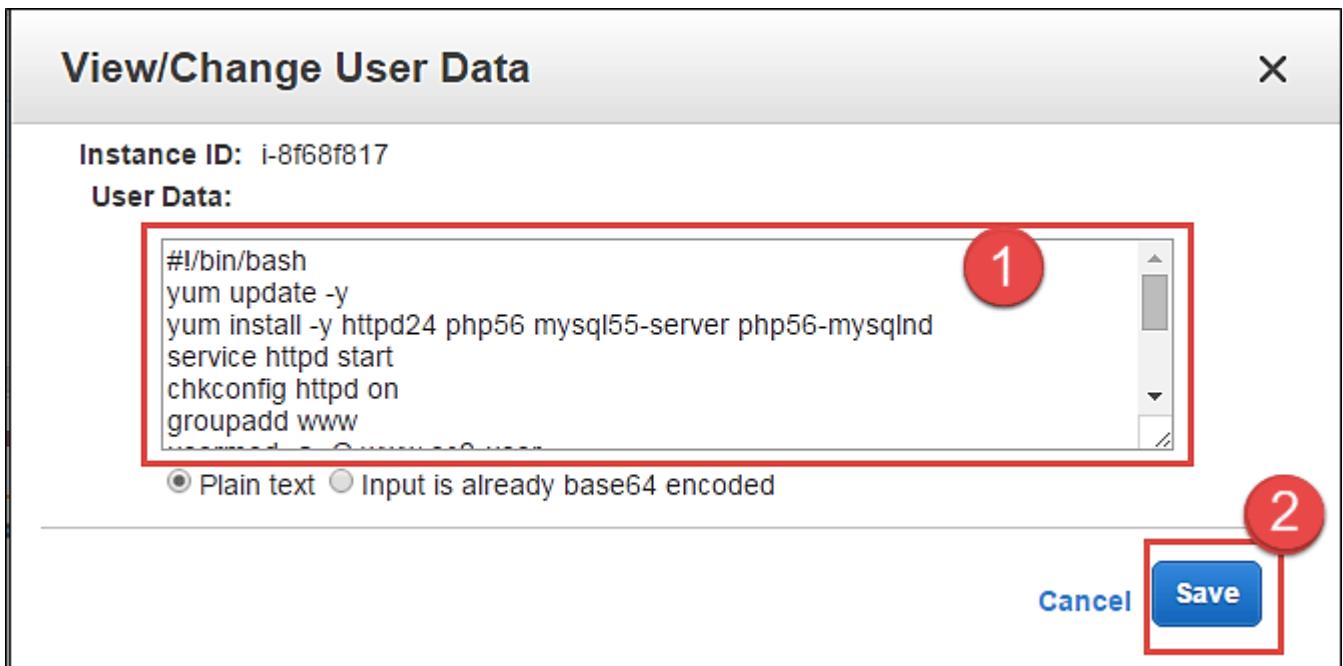
1. Go to 'Instance Settings'.
2. Click on 'View/Change User Data'.



Here for the purpose of demonstration, we have a shell script which installs LAMP stack on the server.

Step 2) In this step,

1. View/ modify your user data field.
2. Click on "Save" tab.



Change the shutdown behavior

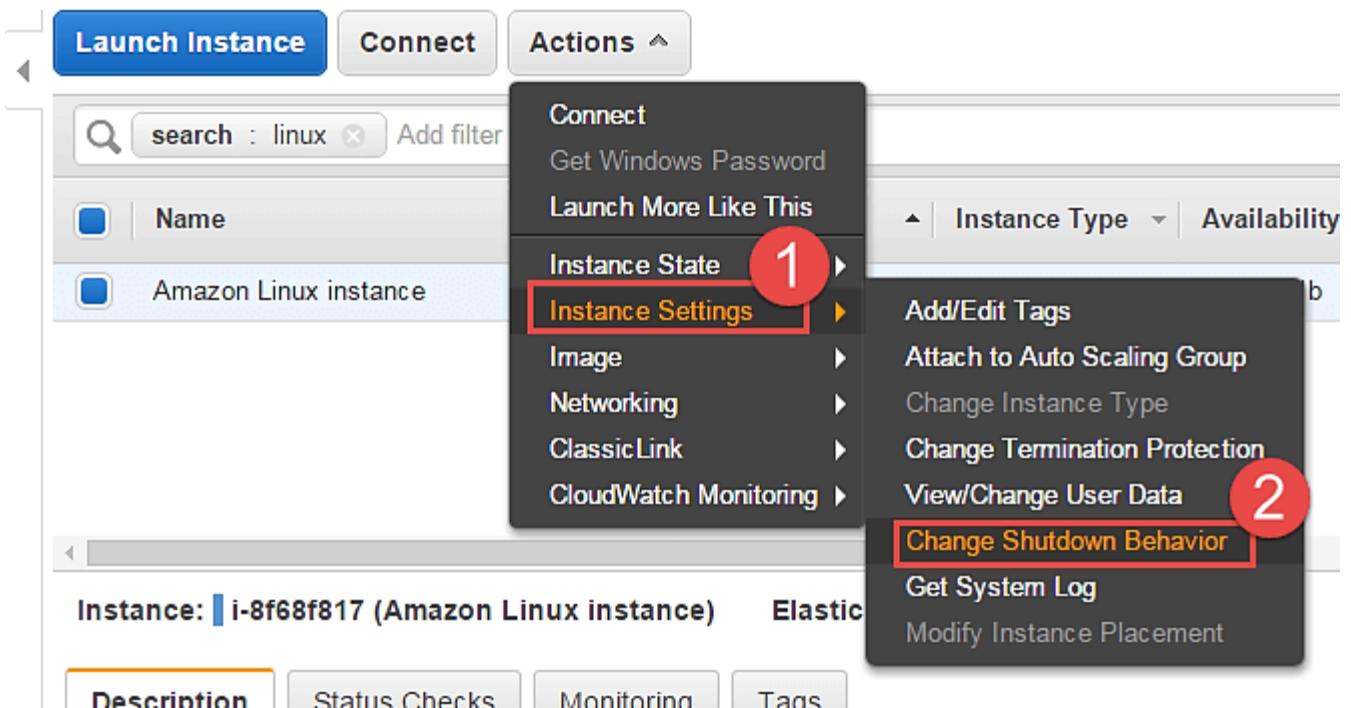
If ever you have accidentally shutdown the instance via the OS console, you don't want AWS EC2 to actually terminate the instance.

For that, we can set up the shutdown behavior as 'Stop' instead of 'Terminate'. We can also do vice versa if the application requirement is as such.

Let's see how to achieve this.

Step 1) In this step,

1. Go to 'Instance Settings'.
2. Click on 'Change Shutdown Behavior.'



Step 2) In this step, click on 'Stop' and then hit apply. The setting will be applied to the instance accordingly.



Step 3) Now when "stop" shutdown is initiated in the instance console via putty, it will not get terminated. It will simply shutdown normally.

The screenshot shows a terminal window with the following text:

```
[ec2-user@ip-192-168-2-119 ~]$ shutdown now
shutdown: Need to be root
[ec2-user@ip-192-168-2-119 ~]$ sudo shutdown now

Broadcast message from ec2-user@ip-192-168-2-119
(/dev/pts/0) at 10:12 ...

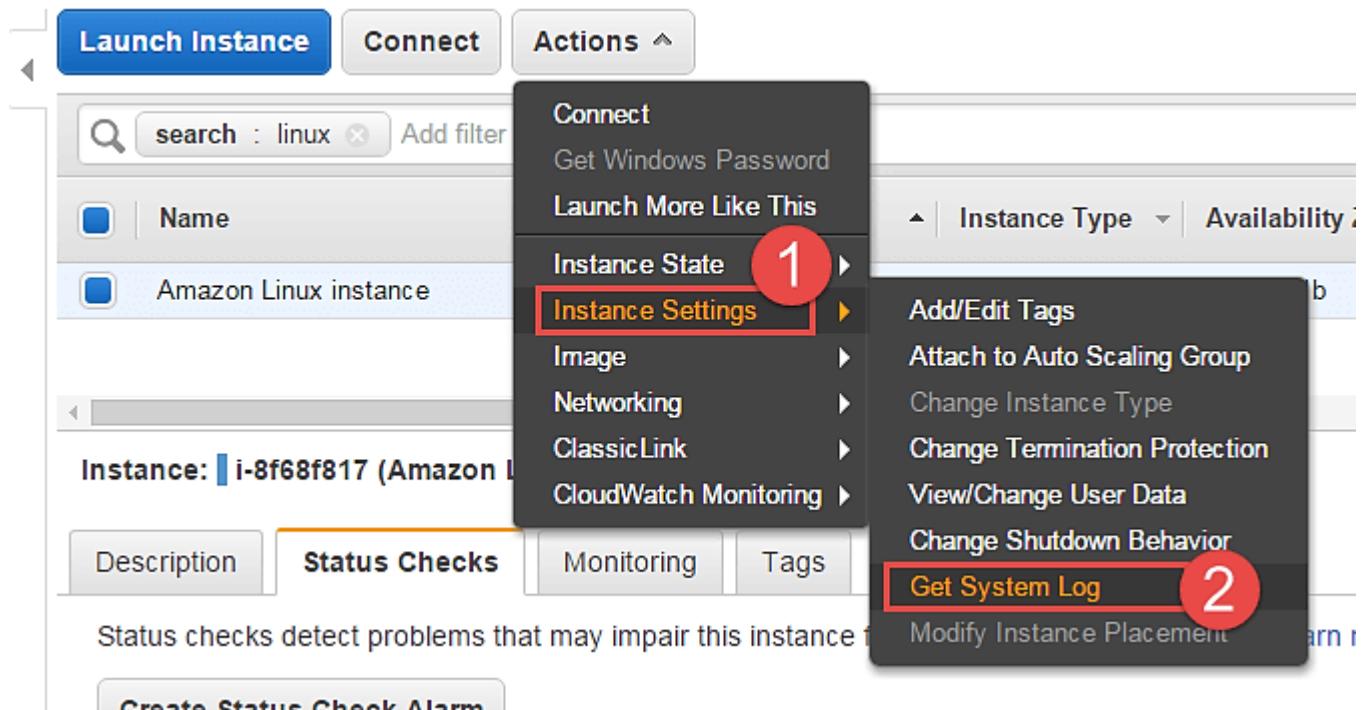
The system is going down for maintenance NOW!
[ec2-user@ip-192-168-2-119 ~]$
```

View System Log

You can see the system log for any EC2 instance for troubleshooting purposes etc.

Step 1) In this step,

1. Go to 'Instance Settings'.
2. Click on 'Get System Log'.



You can see a separate window depicting the instance log details. Here we can see a snap of log when the instance was restarted.

System Log: i-8f68f817 (Amazon Linux instance)

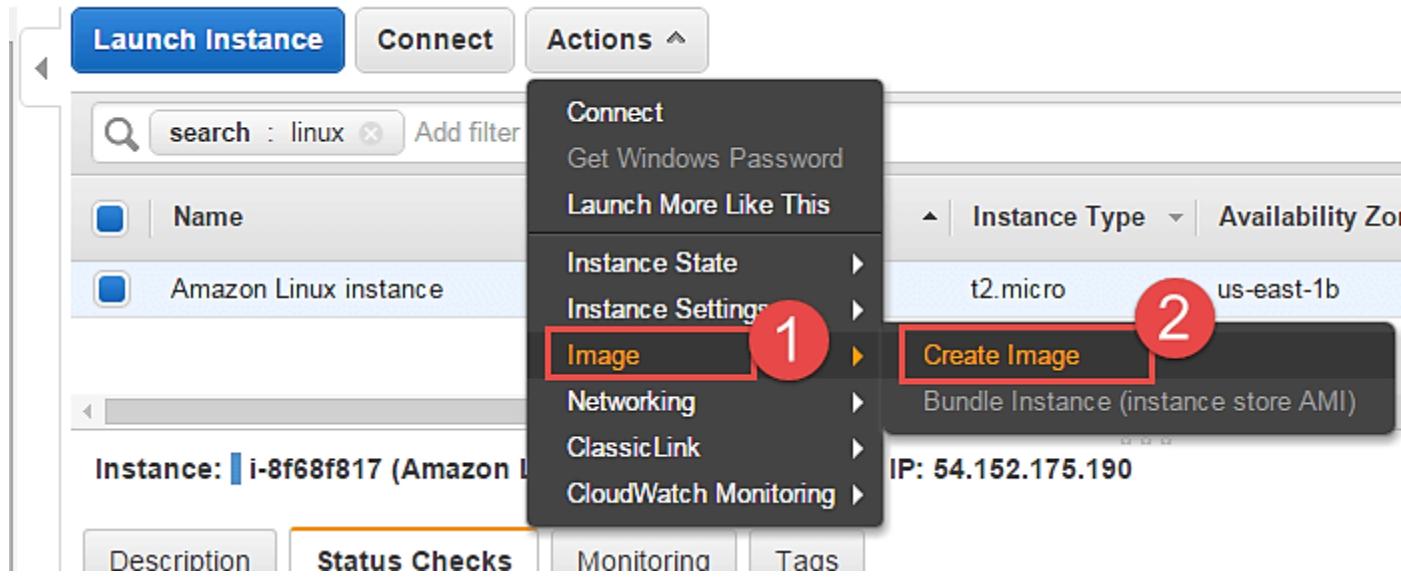
```
Stopping auditd: [ 338.466244] audit: type=1305 audit(1455099123.120:35): audit_piu=0
[ OK ]
[ 338.558605] audit: type=1305 audit(1455099123.212:36): auid=4294967295 ses=4294967295
[ 338.565151] audit: type=1305 audit(1455099123.220:37): audit_enabled=0 old=1 auid=4294967295
Shutting down system logger: [ OK ]
Shutting down interface eth0: [ OK ]
Shutting down loopback interface: [ OK ]
Stopping rngd: [ OK ]
Retrigger failed udev events--type=failed is deprecated and will be removed from a future udev release
udevadm[2629]: --type=failed is deprecated and will be removed from a future udev release
[ OK ]
[H]Telling INIT to go to single user mode.
init: rc main process (2386) killed by TERM signal
/dev/fd/10: line 2: plymouth: command not found
[root@ip-192-168-2-119 /]# /dev/fd/9: line 1: /sbin/plymouthd: No such file or directory
init: plymouth-shutdown main process (2665) terminated with status 1
initctl: Event failed
init: splash-manager main process (2661) terminated with status 1
Stopping block device availability: Deactivating block devices:
[ OK ]
Sending all processes the TERM signal... [ OK ]
Sending all processes the KILL signal... init: rcS-sulogin main process (2638) killed
[ OK ]
Saving random seed: [ OK ]
Turning off quotas: [ OK ]
init: Re-executing /sbin/init
[ 1402.297137] EXT4-fs (xvda1): re-mounted. Opts: (null)
Please stand by while rebooting the system...
[ 1402.456859] xenbus: xenbus_dev_shutdown: device/vfb/0: Initialising != Connected, skipping
[ 1402.462662] reboot: Restarting system
[ 1402.465001] reboot: machine restart
```

Create an instance AMI

You can create an AMI of your EC2 instance for backup.

Step 1) In this step,

1. Go to 'Image'.
2. Click on 'Create Image'.



An image creation wizard will open.

Step 2) In this step,

1. Add the image name
2. Give some friendly description for the AMI
3. Check the volumes and then hit 'Create Image' button.

Create Image

Instance ID: i-8f68f817

Image name: **Backup AMI** 1

Image description: image for backing up the instance 2

No reboot:

Instance Volumes

Volume Type	Device	Snapshot	Size (GiB)	Volume Type
Root	/dev/xvda	snap-a17f1036	8	Magnetic

[Add New Volume](#)

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

AWS will receive your create image request and will send a notification immediately.

Create Image

✓ Create Image request received.

[View pending image ami-4d0a2027](#)

Any snapshots backing your new EBS image can be managed on the [snapshots screen](#) after the image has been created.

You can check the status of the request on the EC2 dashboard as 'pending' just like what is shown below.

The screenshot shows the AWS EC2 Dashboard with the 'AMIs' section selected. The main pane displays the details of an AMI named 'ami-4d0a2027'. The 'Status' field is highlighted with a red box and a green arrow pointing to it, indicating its current state. The AMI details are as follows:

AMI ID	ami-4d0a2027
Owner	018511290429
Status	pending
Creation date	February 10, 2016 at 4:03:03 PM UTC+5:30
Architecture	x86_64

After a while the status is "available" and you will have your AMI ready as a backup.

Screenshot of the AWS EC2 Dashboard showing the AMI details for 'Backup AMI'. The 'Status' field is highlighted with a red box and a green arrow pointing to it.

AMI ID: ami-4d0a2027

Owner: 018511290429

Status: available

Creation date: February 10, 2016 at 4:03:03 PM UTC+5:30

You can also de-register it from the dashboard once the backup is old.

Screenshot of the AWS EC2 Dashboard showing the Actions menu for an AMI. The 'Deregister' option is highlighted with a red box.

- Launch
- Actions ▾
- Deregister
- Register New AMI
- Copy AMI
- Modify Image Permissions
- Add/Edit Tags
- Modify Boot Volume Setting

Change the instance network settings

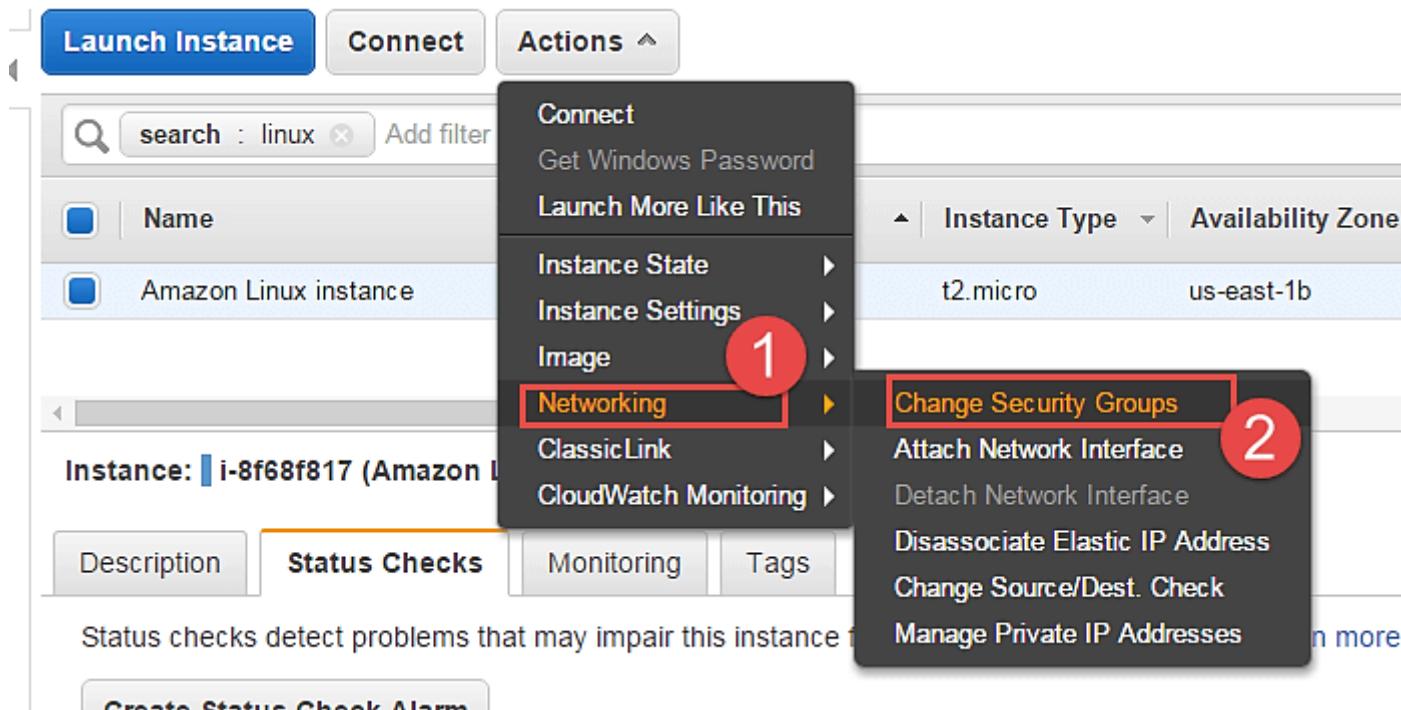
Change the Security Group

You can change the SG (Security Group) of an instance anytime. If you have another security group with different firewall rules, you can easily do so using the console.

Let's see how.

Step 1) In this step,

1. Go to 'Networking'.
2. Click on 'Change Security Groups'.



Step 2) In the change security groups wizard, it will show the already existing SG on the instance along with a list of all the security groups in the region.

Change Security Groups

Instance ID:i-8f68f817

Interface ID:eni-90afecce

Select Security Group(s) to associate with your instance

Security Group ID	Name	Description
<input type="checkbox"/> sg-099b466e	Aeris_SG	launch-wizard-6 created 201
<input type="checkbox"/> sg-21bbde45	default	default VPC security group
<input type="checkbox"/> sg-12433e75	launch-wizard-6	launch-wizard-6 created 201
<input checked="" type="checkbox"/> sg-1bbadf7f	Prachi_Public SG	launch-wizard-2 created 201
<input type="checkbox"/> sg-4a5aac2c	quick-create-1	quick-create-1 created on T
<input type="checkbox"/> sg-470d3621	rds-launch-wizard	Created from the RDS Mana
<input type="checkbox"/> sg-29003b4f	rds-launch-wizard-1	Created from the RDS Mana

Canc

Step 3) In this step,

1. Tick the box against your desired SG
2. Click on 'Assign Security Groups' button.

Change Security Groups

Instance ID:i-8f68f817

Interface ID:eni-90afecce

Select Security Group(s) to associate with your instance

Security Group ID	Name	Description
<input type="checkbox"/> sg-21bbde45	default	default VPC security group
<input type="checkbox"/> sg-12433e75	launch-wizard-6	launch-wizard-6 created 20
<input type="checkbox"/> sg-1bbadf7f	Prachi_Public SG	launch-wizard-2 created 20
<input type="checkbox"/> sg-4a5aac2c	quick-create-1	quick-create-1 created on T
<input type="checkbox"/> sg-470d3621	rds-launch-wizard	Created from the RDS Man
<input type="checkbox"/> sg-29003b4f	rds-launch-wizard-1	Created from the RDS Man
<input checked="" type="checkbox"/> sg-62d7d21b	Web Server SG	launch-wizard-7 created 20

1

Canc

Step 4) On the EC2 Dashboard, you can see that the SG of the instance has been changed. The instance will now send/receive traffic based on the new SG settings.

The screenshot shows the AWS EC2 Instances page. At the top, there are three buttons: 'Launch Instance' (blue), 'Connect', and 'Actions'. Below this is a search bar with 'search : linux' and a 'Add filter' button. A table lists instances, with one row selected: 'Amazon Linux instance' (Instance ID: i-8f68f817, Instance Type: t2.micro, Availability Zone: us-east-1b). Below the table, the instance details are shown: Instance ID: i-8f68f817, Instance state: running, Instance type: t2.micro, Private DNS: ip-192-168-2-119.ec2.internal, Private IPs: 192.168.2.119, Secondary private IPs: (empty), VPC ID: vpc-d5194fb0. To the right, there are tabs for 'Description' (selected), 'Status Checks', 'Monitoring', and 'Tags'. On the far right, there are several other tabs: 'Public IP', 'Private IP', 'EBS', 'Availability Zone', 'Security Groups' (which is highlighted with a red box), and 'Scheduled Tasks'.

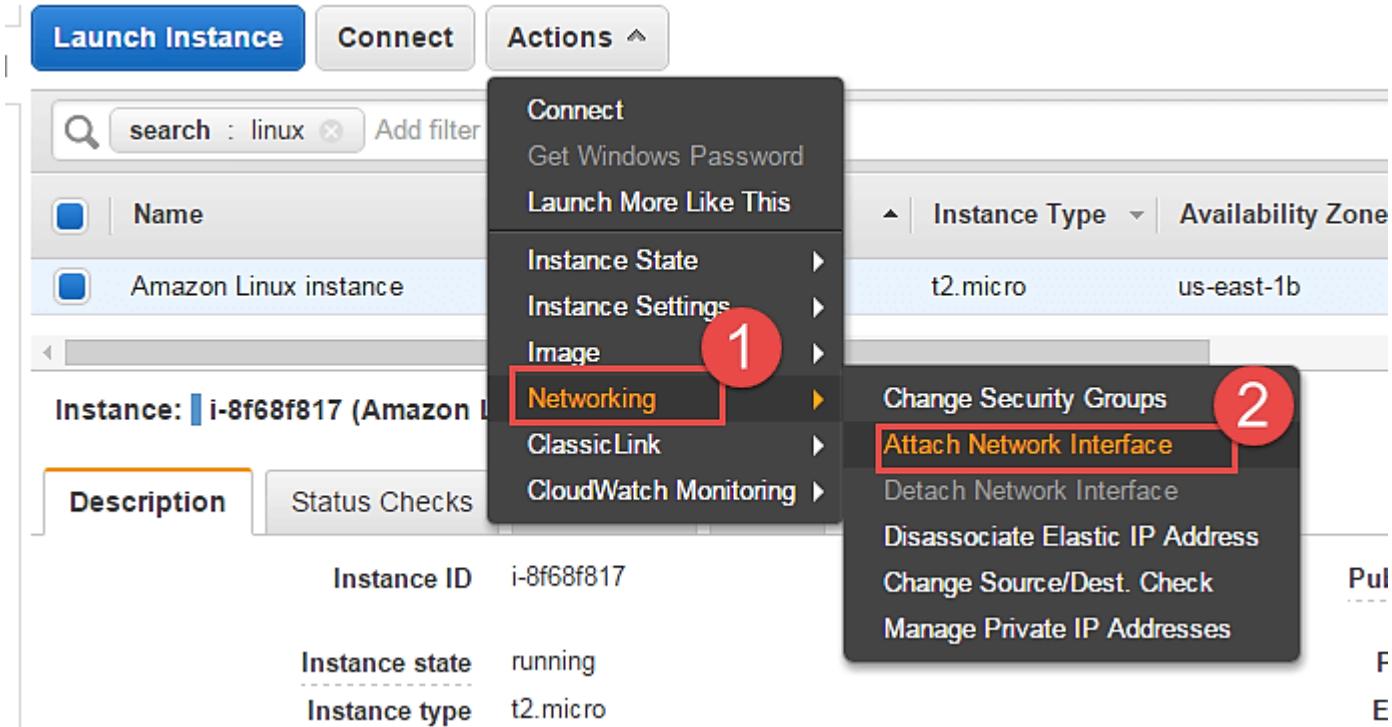
You can also add multiple security groups.

Add a Network Interface

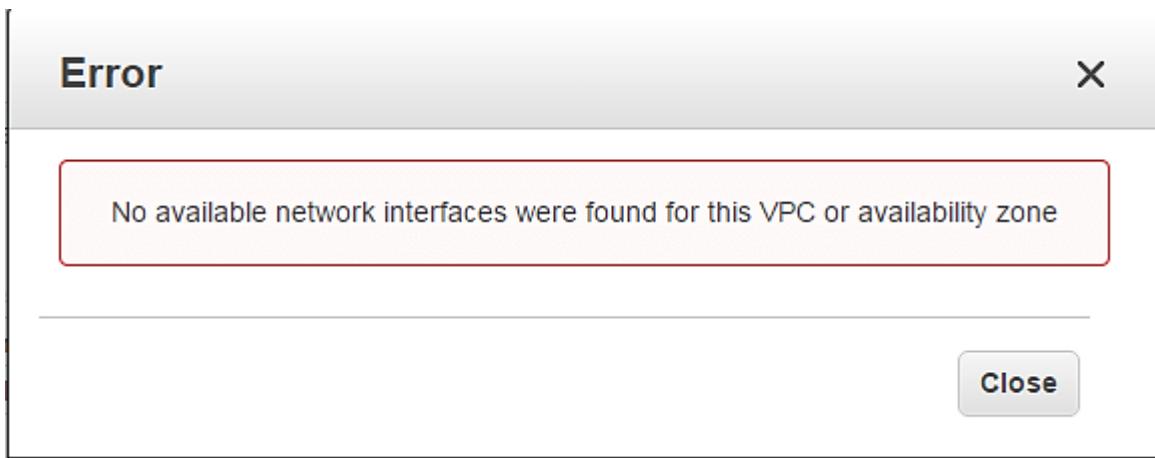
A network interface is like another NIC card to an instance. It will have another set of IPs additional to the already existing primary Network Interface.

Step 1) In this step,

1. Go to 'Networking'.
2. Click on 'Attach Network Interface'.



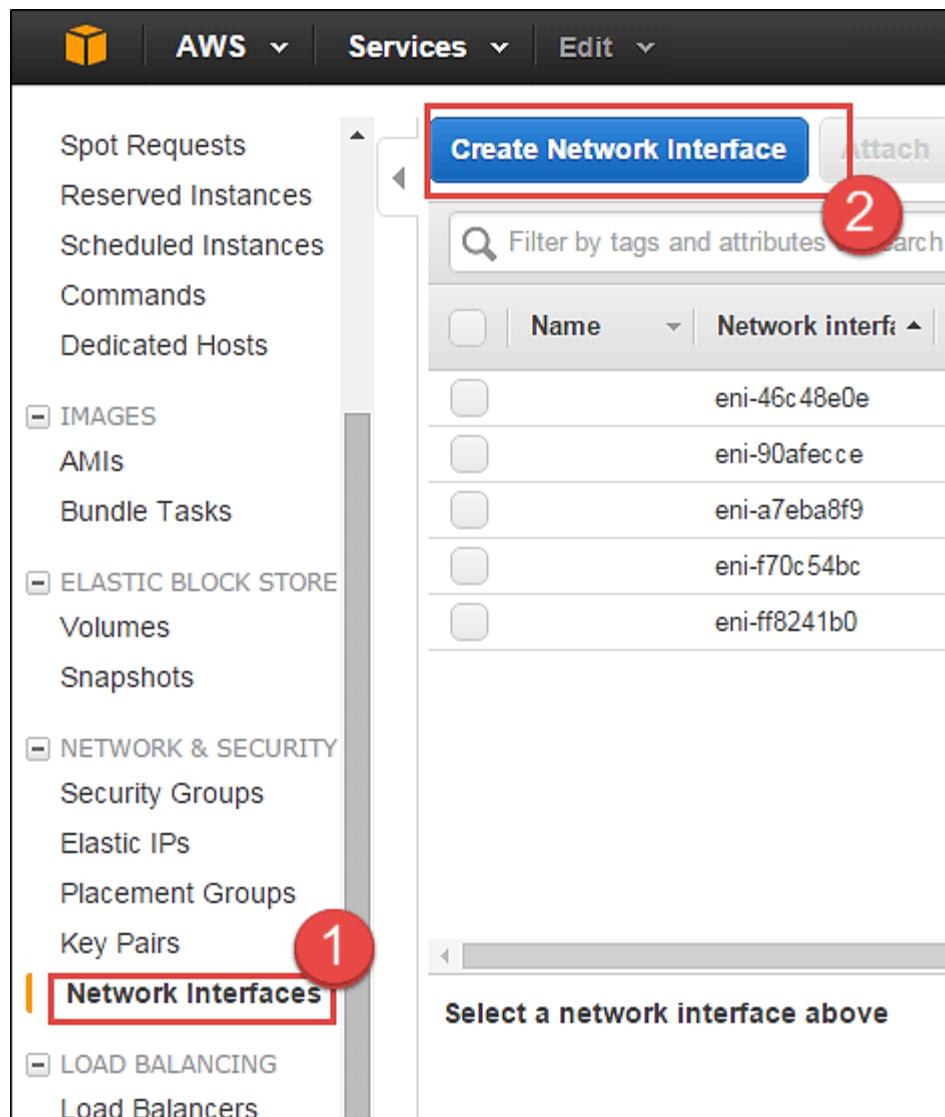
You will get an error prompt if you don't have a Network Interface already created.



Let's see how to create a Network Interface quickly.

Step 2) In this step,

1. Go to EC2 Dashboard, and click on 'Network Interfaces' on the left pane.
2. Click on 'Create Network Interface' button.



Step 3) In this step,

1. Add a description for your network interface
2. Select the subnet where you want to create your network interface. Keep the auto assign the private IP option default
3. Security groups are applied to a network interface of an instance, so here you will get an option for the same. Select your desired SG
4. Once you're done entering the details, click on 'Create.'

Create Network Interface

1

Description	(i)	Test Network interface
Subnet	(i)	subnet-b3e3d0ea (192.168.2.0/24) us-east-1b Prachi_Test-Public
Private IP	(i)	auto assign
Security groups	(i)	sq-099b466e - Aeris SG - launch-wizard-6 created 2015-07-21T14: sq-1bbadf7f - Prachi Public SG - launch-wizard-2 created 2015-03-11 sg-62d7d21b - Web Server SG - launch-wizard-7 created 2016-02-01 sg-21bbde45 - default - default VPC security group

Now you can come back to the EC2 Dashboard and check that your network interface is getting created.

	Name	Network interface	Subnet ID	VPC ID	Zone	Security groups
		eni-46c48e0e	subnet-a94427de	vpc-d5194fb0	us-east-1a	Prachi_Public
		eni-90afecce	subnet-b3e3d0ea	vpc-d5194fb0	us-east-1b	Web Server
<input checked="" type="checkbox"/>		eni-a7eba8f9	subnet-b3e3d0ea	vpc-d5194fb0	us-east-1b	Prachi_Public
		eni-f70c54bc	subnet-a94427de	vpc-d5194fb0	us-east-1a	Aeris_SG
		eni-ff8241b0	subnet-a94427de	vpc-d5194fb0	us-east-1a	Prachi_Public

Now come back on Step 2) and go ahead with selecting your available interface which we just created and attach it to the instance.

Now as you can see the network interface which we just created is enlisted below automatically.

Attach Network Interface

X

Instance ID:i-8f68f817

Network Interface: eni-a7eba8f9 (Test Network interface) ▾

If you attach another network interface to your instance, your current public IP address is released when you restart your instance. Learn more about [public IP addresses](#).

[Cancel](#)

[Attach](#)

Your network interface will be attached to the instance immediately.

We can come back to the EC2 Dashboard and check our instance now. Note that the instance has 2 private IPs belonging to 2 network interfaces.

Name	Instance ID	Instance Type	Availability Zone
Amazon Linux instance	i-8f68f817	t2.micro	us-east-1b

Description [Status Checks](#) [Monitoring](#) [Tags](#)

Instance ID i-8f68f817

Instance state running

Instance type t2.micro

Private DNS ip-192-168-2-119.ec2.internal

Private IPs 192.168.2.119, 192.168.2.81

Secondary private IPs

VPC ID vpc-d5194fb0

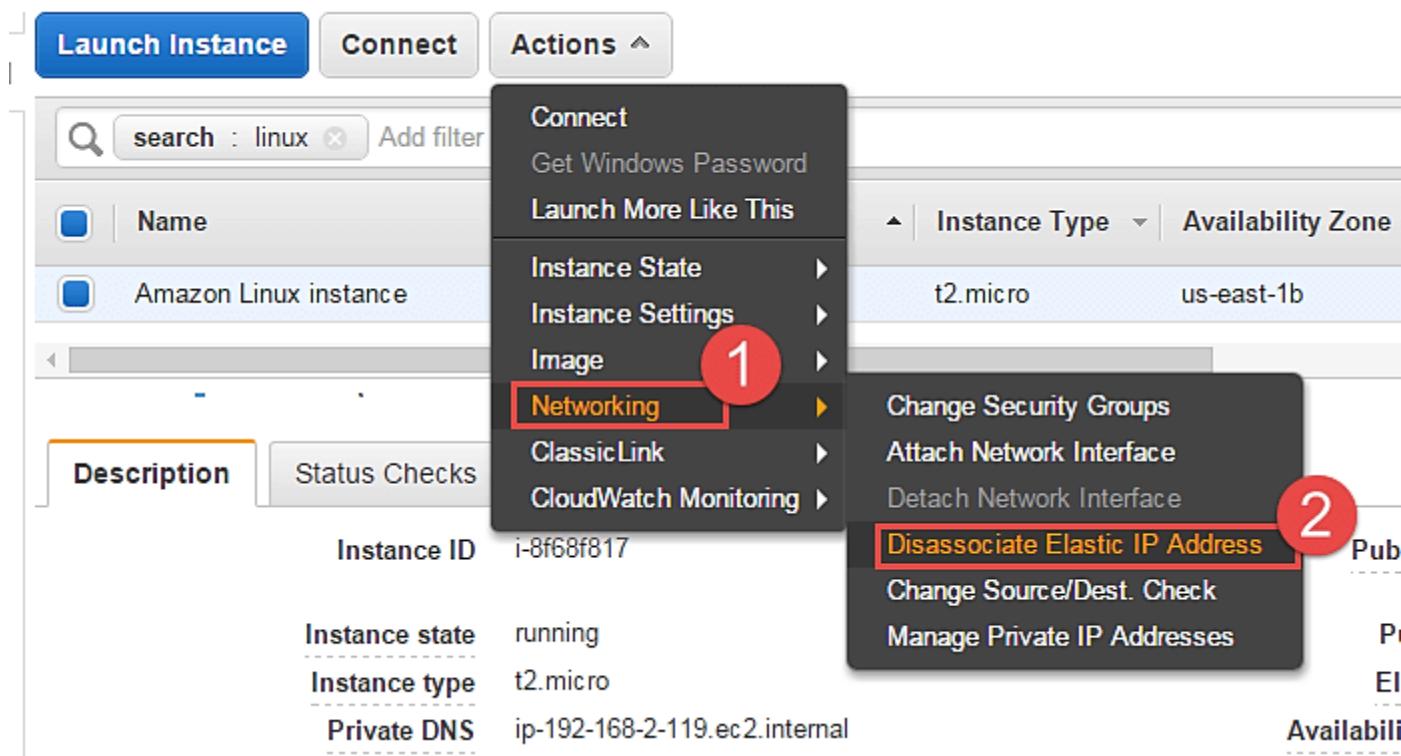
Dissociating EIP

An Elastic IP is a static Public IP.

You can dissociate an EIP directly from the instance dashboard.

Step 1) In this step

1. Click on 'Networking.'
2. Click on 'Dissociate Elastic IP Address.'



Step 2) Click on the button of dissociate, once we have verified the instance id and the EIP.

Disassociate Elastic IP Address

Are you sure that you wish to disassociate this Elastic IP Address?

Public IP 54.152.175.190

Instance ID i-8f68f817

Network interface ID eni-90afecce

[Cancel](#)

[Yes, Disassociate](#)

Check below that the instance dashboard now shows the EIP field blank.

The screenshot shows the AWS CloudWatch Metrics interface. At the top, there are three buttons: 'Launch Instance' (blue), 'Connect', and 'Actions'. Below these are search and filter fields. A table lists instances, with one row selected: 'Amazon Linux instance' (Instance ID: i-8f68f817, Instance Type: t2.micro, Availability Zone: us-east-1b). Below the table, tabs for 'Description', 'Status Checks', 'Monitoring', and 'Tags' are visible. The 'Description' tab is active, showing detailed instance information:

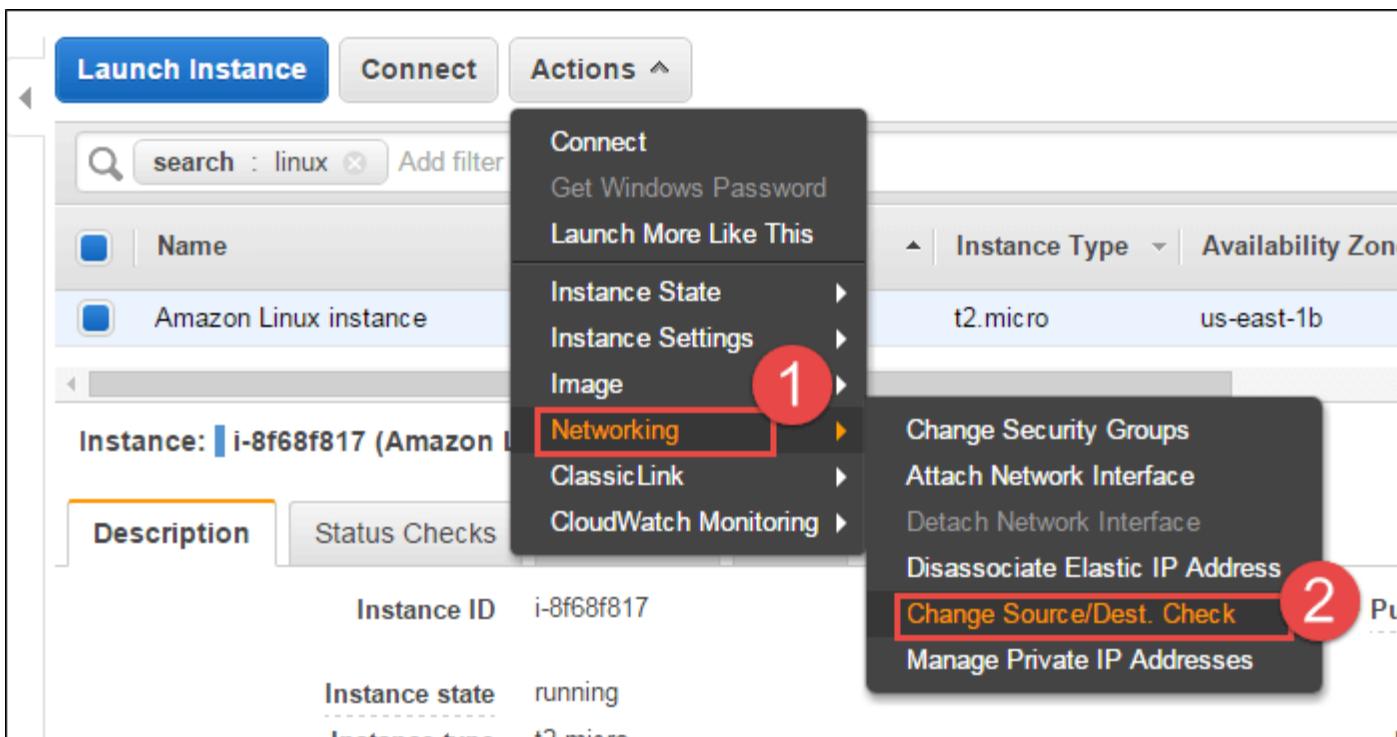
	Value
Instance ID	i-8f68f817
Instance state	running
Instance type	t2.micro
Private DNS	ip-192-168-2-119.ec2.internal
Private IPs	192.168.2.119
Secondary private IPs	
VPC ID	vpc-d5194fb0
Subnet ID	subnet-b3e3d0ea
Network interfaces	eth0
Source/dest. check	True

Change Source/Destination check

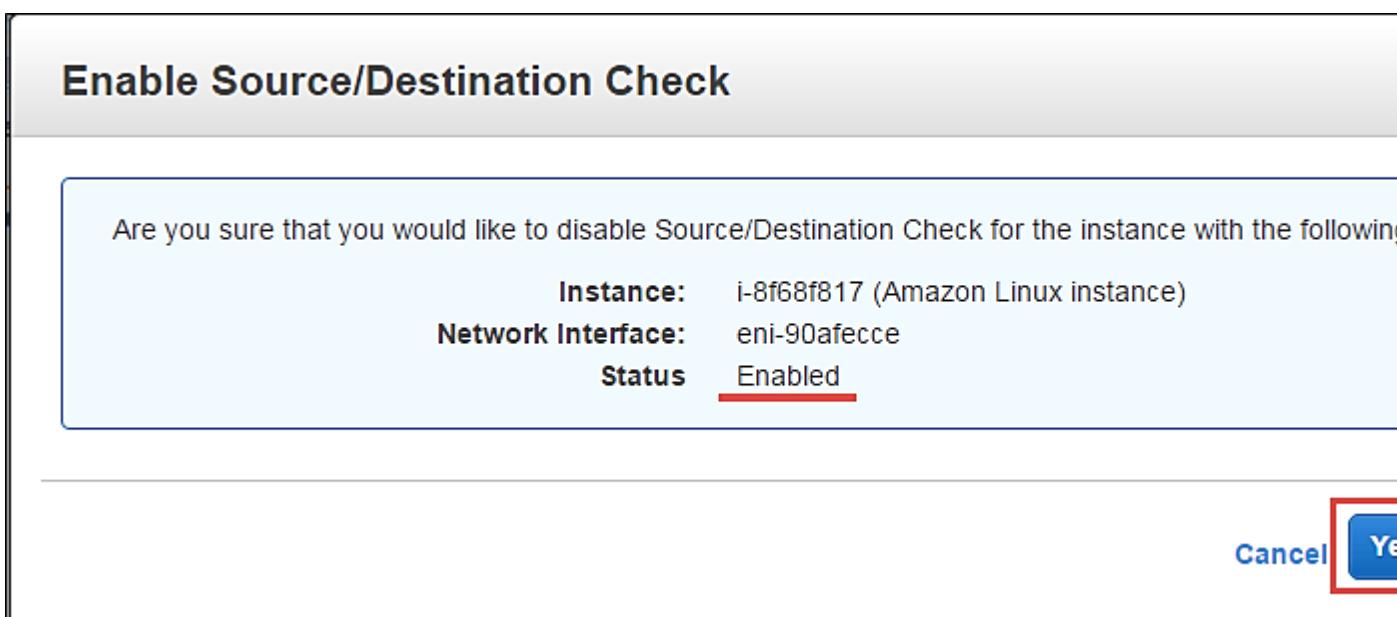
The Source/Destination Check attribute controls whether source/destination checking is enabled on the instance. Disabling this attribute enables an instance to handle network traffic that isn't specifically destined for the instance. For example, instances running services such as network address translation, routing, or a firewall should set this value to disabled.

Step 1) In this step,

1. Click on 'Networking.'
2. Click on 'change Source/Dust. Check'



Step 2) Click on 'Disable'. If it is disabled already, you can enable it in this step.

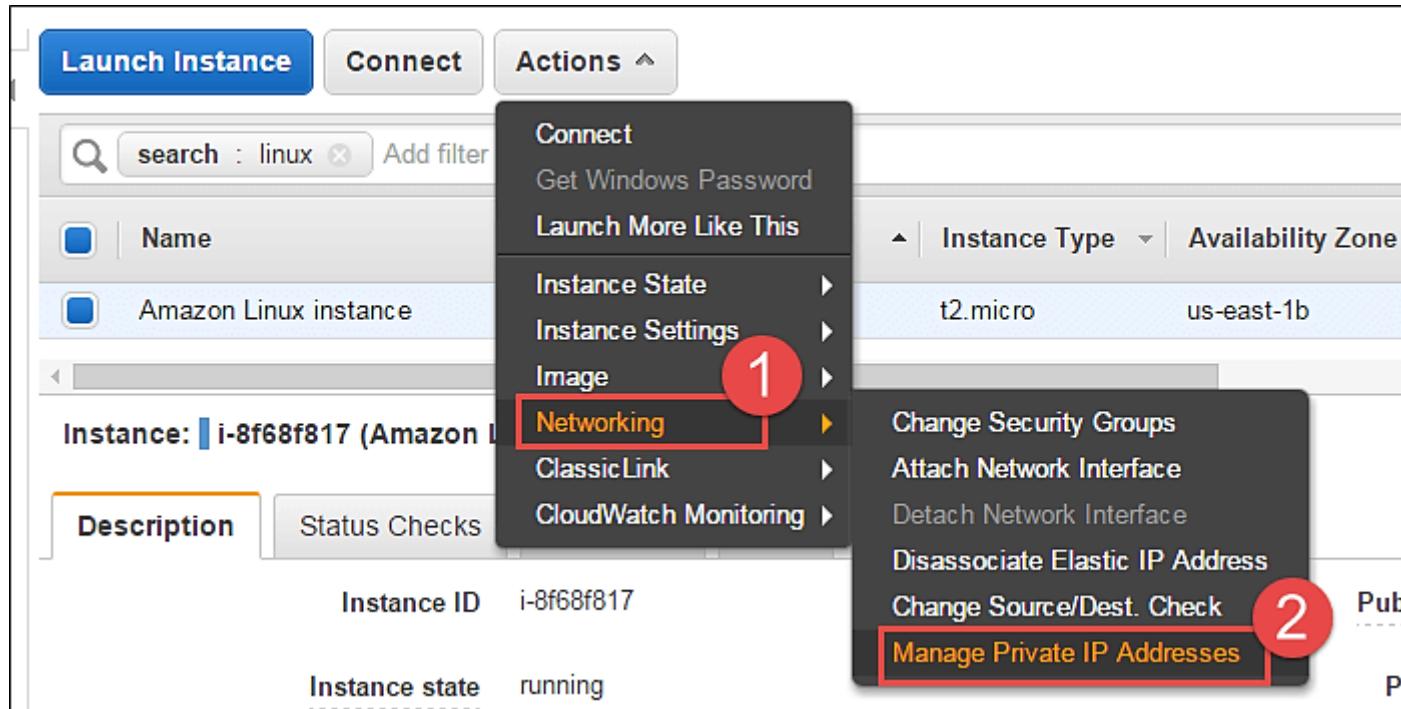


Manage private IP addresses

You can assign multiple private IP addresses to a single instance if that is your application architecture's design. The maximum no of IPs you can assign of course depends on the EC2 instance type.

Step 1) In this step,

1. Click on 'Networking.'
2. Click on 'Manage Private IP addresses.'



You will be redirected to a new window to assign a secondary IP address to your instance.

Step 2) In this step,

1. Here we are leaving the field blank. This will enable AWS to auto-assign any available private IP to our instance.
2. Click on 'Update.'

Manage Private IP Addresses

X

You can assign and unassign secondary private IP addresses on each network interface. Leave the address field blank and an available address will be assigned or enter an IP address that you want to assign.

▼ eth0: eni-90afecce - Primary network interface - 192.168.2.0/24

Private IP	Public IP
192.168.2.119	54.152.175.190
<input type="button" value="Auto-assign"/> 1	<input type="button" value="Undo"/>

[Assign new IP](#)

Allow reassignment [\(i\)](#)

Are you sure you want to perform the following changes:

- 1 unspecified private IP addresses will be assigned to eni-90afecce

[Cancel](#)

2

Note that an IP has been automatically assigned here.

Manage Private IP Addresses

X

You can assign and unassign secondary private IP addresses on each network interface. Leave the address field blank and an available address will be assigned or enter an IP address that you want to assign.

▼ eth0: eni-90afecce - Primary network interface - 192.168.2.0/24

Private IP	Public IP
192.168.2.119	54.152.175.190
192.168.2.163	Unassign

[Assign new IP](#)

Allow reassignment [\(i\)](#)

[Cancel](#)

[Yes, Update](#)

Also, come back to the EC2 dashboard and notice the 2 private IPs assigned. These are 2 IPs on a single network interface.

The screenshot shows the AWS EC2 Instances page. At the top, there are three buttons: 'Launch Instance' (blue), 'Connect', and 'Actions'. Below this is a search bar with 'search : linux' and an 'Add filter' link. A table lists instances, with one row selected: 'Amazon Linux instance' (Instance ID: i-8f68f817, Instance Type: t2.micro, Availability Zone: us-east-1b). Below the table, the instance details for 'i-8f68f817 (Amazon Linux instance)' are shown. The 'Description' tab is selected, showing the following information:

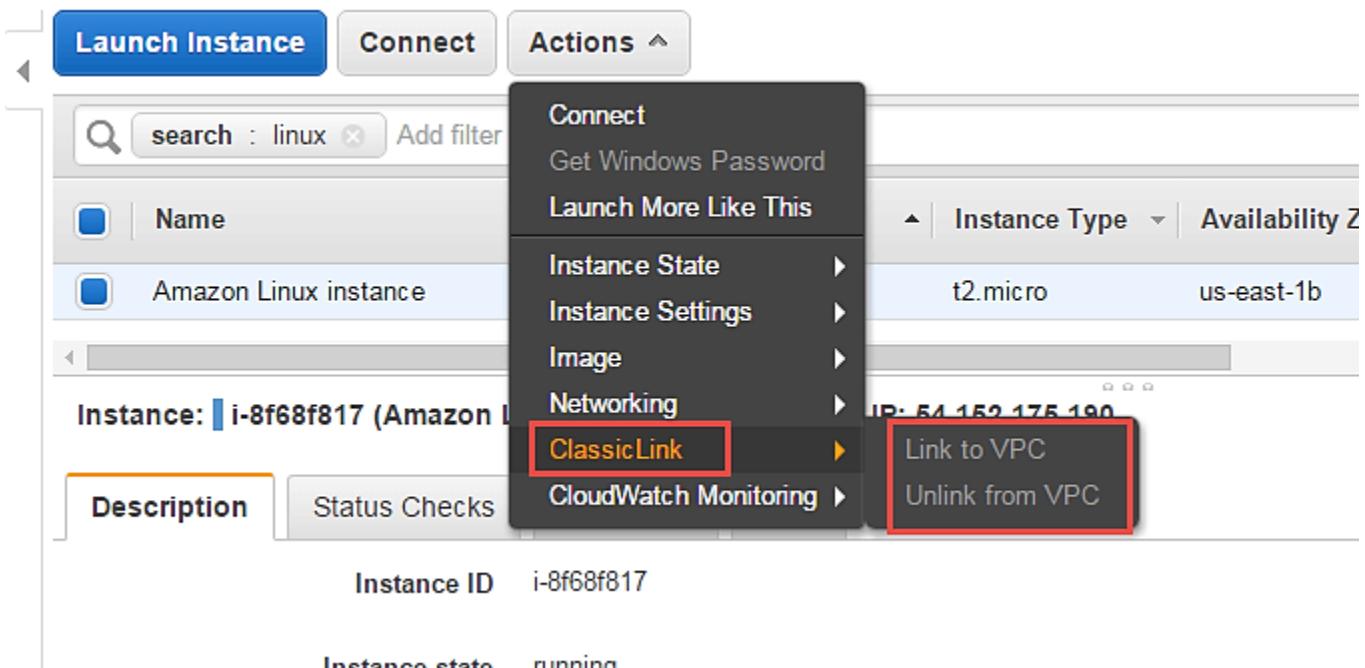
Instance ID	i-8f68f817
Instance state	running
Instance type	t2.micro
Private DNS	ip-192-168-2-119.ec2.internal
Private IPs	192.168.2.119
Secondary private IPs	192.168.2.163
VPC ID	vpc-d5194fb0
Subnet ID	subnet-b3e3d0ea
Network interfaces	eth0

The 'Private IPs' and 'Secondary private IPs' rows are highlighted with a red box.

Enable/disable ClassicLink to a VPC

If your instance is provisioned in EC2 – Classic, which is a deployment mode in AWS where resources are provisioned out of a VPC; then you can link your instance to a VPC environment as shown below.

The options below are disabled for us as our instance is already in a VPC.



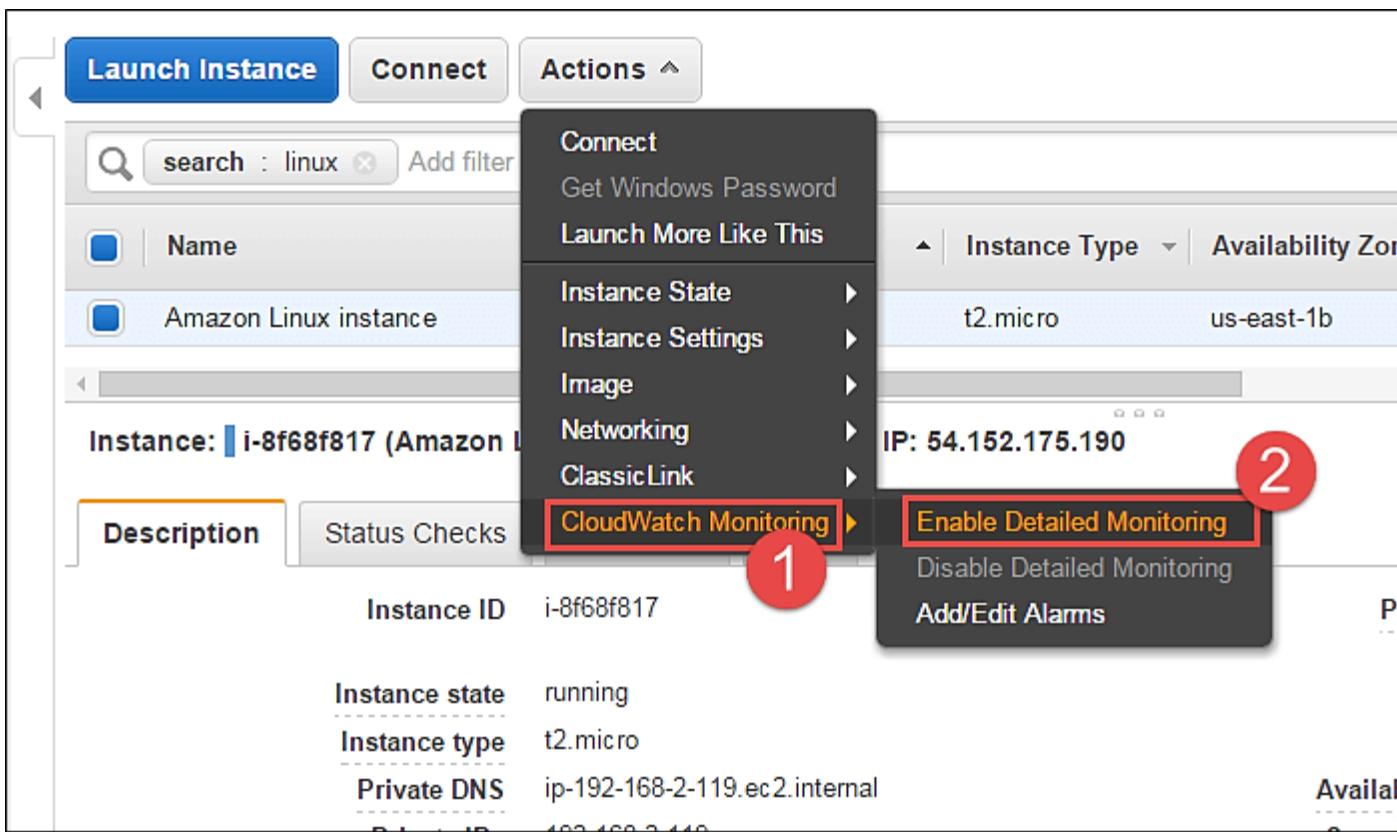
Enable detailed CloudWatch monitoring

AWS will by default have basic CloudWatch monitoring enabled on all its resources. However, if our instances are production instances, we may wish to enable detailed monitoring on them with additional costs of course.

Step 1) In this step,

1. Click on 'CloudWatch Monitoring'
2. Click on 'Enable Detailed Monitoring'

You can also add/edit alarms to alert you for attributes in your CloudWatch monitoring metrics.



Summary

Thus, we saw in this tutorial, how to enable/modify various attributes in AWS for the instance configuration from the Management Console after it is launched.

AWS provides many more configuration options via CLI/API.

AWS Certified Solutions Architect - Associate 2018

What is AWS Lambda? Lambda

Function with Examples

Before AWS Lambda, let's understand:

What is Serverless?

Serverless is a term that generally refers to serverless applications. Serverless applications are ones that don't need any server provision and do not require to manage servers.

What is AWS Lambda?

AWS LAMBDA is an event-driven, serverless computing platform provided by Amazon as a part of Amazon Web Services. Therefore you don't need to worry about which AWS resources to launch, or how will you manage them. Instead, you need to put the code on Lambda, and it runs.

In AWS Lambda the code is executed based on the response of events in AWS services such as add/delete files in S3 bucket, HTTP request from Amazon API gateway, etc. However, a Lambda can only be used to execute background tasks.

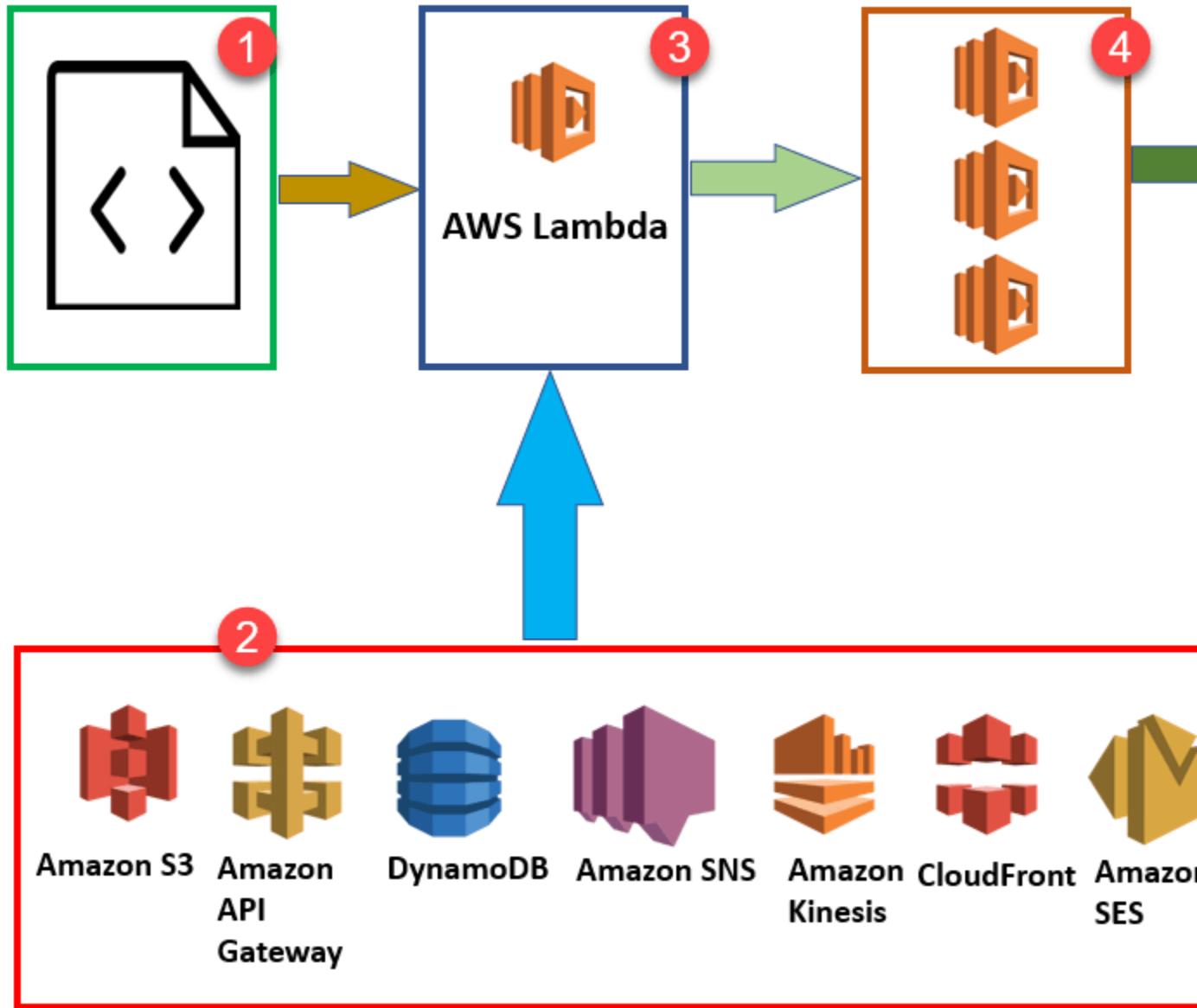
AWS Lambda helps you to focus on your core product and business logic instead of managing operating system (OS) access control, OS patching, right-sizing, provisioning, scaling, etc.

In this AWS Lambda tutorial for beginners, you will learn:

- [How does AWS Lambda work?](#)
- [Events that Trigger AWS Lambda](#)
- [AWS Lambda Concepts](#)
- [AWS Lambda VS AWS EC2](#)
- [AWS Lambda VS AWS Elastic Beanstalk](#)
- [Use Cases of AWS Lambda](#)
- [Best practices of Lambda function](#)
- [When not to use AWS Lambda](#)
- [Advantages of using AWS Lambda](#)
- [Limitations of AWS Lambda](#)

How does AWS Lambda work?

The following block diagram explains the working of AWS Lambda in a few easy steps:



Step 1: First upload your AWS Lambda code in any language supported by AWS Lambda. Java, Python, Go, and C# are some of the languages that are supported by AWS lambda.

Step 2: These are some AWS services which allow you to trigger AWS Lambda.

Step 3: AWS Lambda helps you to upload code and the event details on which it should be triggered.

Step 4: Executes AWS Lambda Code when it is triggered by AWS services:

Step 5: AWS charges only when the AWS lambda code executes, and not otherwise.

This will happen in the following scenarios:

- Upload files in an S3 bucket
- When HTTP get/post endpoint URL is hit
- For adding/modifying and deleting Dynamo DB tables
- In the process of data streams collection
- Push notification
- Hosting of website
- Email sending

Note: You should remember that you will charge for AWS services only when the AWS Lambda code executes, else you don't need to pay anything.

Events that Trigger AWS Lambda

Here, are Events which will be triggered when you use AWS Lambda.

- Insert, updating and deleting data Dynamo DB table
- To include push notifications in SNS
- To search for log history in CloudTrail
- Entry into an S3 object
- DynamoDB can trigger AWS Lambda whenever there is data added, modified, and deleted in the table.
- Helps you to schedule the event to carry out the task at regular time pattern.
- Modifications to objects in S3 buckets
- Notifications sent from Amazon SNS.
- AWS Lambda can be used to process the CloudTrail logs
- API Gateway allows you to trigger AWS Lambda on GET/POST methods.

AWS Lambda Concepts

Function:

A function is a program or a script which runs in AWS Lambda. Lambda passes invocation events into your function, which processes an event and returns its response.

Runtimes:

Runtime allows functions in various languages which runs on the same base execution environment. This helps you to configure your function in runtime. It also matches your selected programming language.

Event source:

An event source is an AWS service, such as Amazon SNS, or a custom service. This triggers function helps you to executes its logic.

Lambda Layers:

Lambda layers are an important distribution mechanism for libraries, custom runtimes, and other important function dependencies. This AWS component also helps you to manage your development function code separately from the unchanging code and resources that it uses.

Log streams:

Log stream allows you to annotate your function code with custom logging statements which helps you to analyse the execution flow and performance of your Lambda functions.

How to use AWS Lambda

Step 1) Goto <https://aws.amazon.com/lambda/> and Get Started

AWS Lambda

Run code without thinking about servers. Pay only for the compute time you use.

Get started with AWS Lambda



TECH TALK

Optimizing Your Serverless Applications

Step 2) Create an account or sign in with your existing account

Step 3) In the next Lambda page,

1. Edit the code
2. Click Run

COMPUTE

AWS Lambda

lets you run code without thinking about servers.

You pay only for the compute time that you consume — there is no charge when your code is not running. With Lambda, you can run code for virtually any type of application or backend service, all with zero administration.

How it works

```
1 exports.handler = (event, context, callback) => {  
2     // Succeed with the string "Hello world!"  
3     callback(null, "Hello Guru99!");  
4 };
```

1

2

Step 4) You will see output

AWS Lambda

lets you run code without thinking about servers.

You pay only for the compute time that you consume — there is no charge when your function is not running. With Lambda, you can run code for virtually any type of application or backend service, without provisioning or managing servers.

How it works

① "Hello Guru99!"

```
1 exports.handler = (event, context, callback) => {  
2     // Succeed with the string "Hello world!"  
3     callback(null, 'Hello Guru99!');  
4 };
```

AWS Lambda VS AWS EC2

Here, are some major differences between AWS Lambda and EC2.

Parameters	AWS Lambda	AWS EC2
Definition	AWS Lambda is a Platform as a Service (PaaS). It helps you to run and execute your backend code.	AWS EC2 is a general purpose computing service.
Flexibility	Does not offer any flexibility to log in to compute instances. It allows you to choose a customized operating system or language runtime.	Offers the widest range of systems, storage, and networking options.

Installation process	You need to select your environment where you want to run the code and push the code into AWS Lambda.	For the first software release
Environment restrictions	It is restricted to few languages.	No environment restrictions

AWS Lambda VS AWS Elastic Beanstalk

Here, are some major differences between AWS Lambda and Elastic Beanstalk.

Parameters	AWS Elastic Beanstalk	AWS Lambda
Main task	Deploy and manage the apps on AWS Cloud without worrying about the infrastructure which runs those applications.	AWS Lambda
Selection of AWS resources	It gives you a freedom to select AWS resources; For example, you can choose EC2 instance which is optimal according to your application.	You can choose Lambda
Type of system	It is a stateful system.	It is a stateless system.

Use Cases of AWS Lambda

AWS Lambda used for a wide range of applications like:

- Helps you for ETL process
- Allows you to perform real-time file processing and real-time stream processing
- Use for creating web applications
- Use in Amazon products like Alexa Chatbots and Amazon Echo/Alexa
- Data processing (real-time streaming analytics)
- Automated Backups of everyday tasks
- Scalable back ends (mobile apps, IoT devices)
- Helps you to execute server-side backend logic
- Allows you to filter and Transform data

Best practices of Lambda function

Here, are important best practices of Lambda functions:

- Use the right "timeout."
- Utilize the functions of local storage which is 500MB in size in the /temp folder
- Minimizing the use of start-up code which is not directly related to processing the current event.
- You should use built-in CloudWatch monitoring of your Lambda functions to view and optimize request latencies.

When not to use AWS Lambda

Following are the situation where Lambda is surely not an ideal option:

- It is not appropriate to use AWS Lambda software packages or applications which rely on calling underlying Windows RPCs
- If is used for custom software applications with licensing agreements like MS-Office document processing, Oracle databases, etc.
- AWS Lambda should not be used for custom hardware process such as GPU acceleration, hardware affinity.

Advantages of using AWS Lambda

Here, are pros/benefits of using AWS lambda:

- AWS Lambda is a highly flexible tool to use
- It helps you to grant access to resources, including VPCs
- Author directly with WYSIWYG editor in console.
- You can use it as a plugin for Eclipse and Visual Studio.
- As it is serverless architecture, you don't need to worry about managing or provisioning servers.
- You do not need to set up any Virtual Machine.
- Helps developers to run and execute the code's response to events without building any infrastructure.
- You just need to for the compute time taken, only when your code runs.
- You can monitor your code performance in real time through CloudWatch.
- It allows you to run your code without provisioning or to manage any other server
- Helps you to execute the code only when needed
- You can scale it automatically to handle a few requests per day and even support more than thousands of requests per second.
- AWS Lambda can be configured with the help of external event timers to perform scheduled tasks.
- AWS Lambda should be configured with external event and timers so; it can be used for scheduling.
- Lambda functions are stateless so that it can be scaled quickly.
- AWS Lambda is fast so it will execute your code within milliseconds.

Limitations of AWS Lambda

Here are the cons/disadvantages of using AWS Lambda:

- AWS Lambda tool is not suitable for small projects.
- AWS Lambda entirely relies on AWS for the infrastructure, so you can't install any additional software if your code demands it.
- Concurrent execution is limited to 100
- AWS Lambda completely depended on AWS for the infrastructure; you cannot install anything additional software if your code demands it.
- Its memory volume can vary between 128 to 1536 MB.
- Event request should not exceed 128 KB.
- Lambda functions help you to write their logs only in CloudWatch. This is the only tool that allows you to monitor or troubleshoot your functions.
- Its code execution timeout is just 5 minutes.

Summary

- Serverless is a term that generally refers to serverless applications.
- AWS Lambda is one such serverless compute service. Therefore, you don't need to worry about which AWS resources to launch, or how will they manage them.
- A function is a program or a script which runs in AWS Lambda.
- Runtime allows functions in various languages which runs on the same base execution environment.
- An event source is an AWS service, such as Amazon SNS, or a custom service.
- Lambda layers are an important distribution mechanism for libraries, custom runtimes, and other important function dependencies.
- Log stream allows you to annotate your function code with custom logging statements which helps you to analyse the execution flow and performance of your Lambda functions.
- AWS Lambda is a Platform as a Service (PaaS). It helps you to run and execute your backend code.
- AWS EC2 Is an Infrastructure as a Service (IaaS). It provides virtualized computing resources.
- Deploy and manage the apps on AWS Cloud without worrying about the infrastructure which runs those applications.
- AWS Lambda is used for running and executing your Back-end code. You can't use it to deploy an application.
- AWS Lambda helps you for the ETL process.
- The best practice of Lambda function is to use the right "timeout".
- It is not appropriate to use AWS Lambda software packages or applications which rely on calling underlying Windows RPCs
- AWS Lambda is a highly flexible tool.
- AWS Lambda tool is not suitable for small projects.
- A common event which will be triggered when you use AWS Lambda is Insert, updating and deleting data Dynamo DB table.

• [Prev](#)

[Report a Bug](#)

• [Next](#)

AWS Certification Guide: Cost, Courses, Salary, Exam Details

What is AWS certification?

AWS Certification helps professionals to build credibility and confidence by validating their cloud expertise with an industry-recognized credential. It helps skilled professionals to obtain the various type of AWS certificate according to his/her skill.

Amazon offers a certification for different IT professionals like Cloud Practitioner, Architect, Developer, and Operations roles. Moreover, it also provides certain Specialty certifications to validate advanced skills in specific technical areas.

In this tutorial, you will learn:

- [What is AWS certification?](#)
- [Benefits Of AWS Certificate](#)
- [Types Of AWS Certifications Courses](#)
 - [AWS Certified Cloud Practitioner](#)
 - [AWS Certified Solutions Architect – Associate](#)
 - [AWS Developer -Associate](#)
 - [AWS sysops Administrator- Associate](#)
 - [AWS solution Architect- Professional](#)
 - [AWS Certified DevOps Engineer – Professional](#)

- [AWS Certified Big Data – Specialty](#)
 - [AWS Certified Advanced Networking – Specialty](#)
 - [AWS Certified Security – Specialty](#)
- [Learning Objective Of AWS Certification](#)
- [How To Start For AWS Certification?](#)
 - [Exam Content](#)
- [Current Salary for AWS Certified Professionals](#)

Benefits Of AWS Certificate

Here are pros/benefits of AWS certificate.

- AWS Certification helps any professional to demonstrates their technical expertise and advance their career
- It helps employers find skilled cloud professionals.
- AWS certification allows you to verify your technical understanding and skills.
- Helps to gain access to the AWS Certified LinkedIn Community.
- Allows you to gather latest knowledge about your domain
- Owning AWS certification open door for new opportunities
- Allows you to boost your self-esteem
- You can validate your skills and knowledge in the preeminent cloud computing platform.
- Helps you to demonstrate credibility and dedication to your cloud computing career path.
- It provides access to a network of like-minded peers and AWS thought-leaders.
- Allows you to leverage the AWS shared security responsibility model.

Types Of AWS Certifications Courses

Here, are some most important AWS certification which helps you boost your career.

- AWS Certified Cloud Practitioner
- AWS Certified Developer – Associate
- AWS Certified SysOps Administrator – Associate
- AWS Certified Solutions Architect – Associate
- AWS Certified Solutions Architect – Professional
- AWS Certified DevOps Engineer – Professional
- AWS Certified Big Data – Specialty
- AWS Certified Advanced Networking – Specialty
- AWS Certified Security – Specialty

Let's see each of AWS certification along with with exam its exam details:

[AWS Certified Cloud Practitioner](#)

This certification course helps you to gain knowledge about various types of cloud technology roles with a method to validate their AWS Cloud knowledge to enhance your credibility as an IT professional.

Exam details:

- Prerequisites: Minimum of six months of general AWS cloud experience in any role is recommended.
- Exam Format: Multiple choice questions
- Exam Duration: 90 minutes
- Exam Language: Available in English, Japanese, Korean, and Simplified Chinese
- Cost: 100 USD

After the End, of this course, you will learn,

- Understanding of most fundamental AWS architectural principles
- The value proposition of the AWS cloud
- Important AWS cloud services and their applications
- Underlying security and compliance and shared responsibility model for security
- Core cloud deployment and operating principles

AWS Certified Solutions Architect – Associate

It is an associate certification course which helps you to validate your ability to effectively demonstrate knowledge of how to build and deploy a secure and robust application using AWS technologies.

- Prerequisites: Knowledge and some experience in designing distributed applications
- Format: Multiple choice questions and multiple answers
- Exam Duration: 130 minutes
- Cost: 150 \$

After the End, of course, you will learn,

- About the network technologies and how they work in AWS and how client interfaces connect to the AWS platform.
- AWS-related data security practices, disaster recovery methods, and troubleshooting.
- You will also know the concept of to build secure and reliable applications on the AWS platform.
- Deploying hybrid systems on-premises data center and AWS components
- This exam AWS course includes the design of highly available and scalable systems and its implementation and deployment in AWS server.

AWS Developer -Associate

The AWS certified Developer is an associate exam. The course for this exam teaches you how you can developer and maintaining AWS-based applications. You will able to learn how to write actual code which used AWS software to in your business.

- Prerequisite: One or more years of hands-on experience using AWS.
- Format: Multiple choice and multiple answers
- Exam Duration: 130 minutes
- Exam Cost: \$150

At the End of this course, you will learn,

- You will understand the underlying AWS architecture and the core AWS service
- Hands-on experience in designing, developing, deploying, and maintaining AWS applications.
- AWS services like AWS databases, notifications, workflow services, and services for storage and change management services.

AWS sysops Administrator- Associate

The AWS Certified SysOps Administrator Associate is only a certification exam which is entirely geared toward system administrators. Passing this exam is quite tricky as you need both technical expertise as well as conceptual understanding about the operational aspect of the AWS platform.

Exam Details:

- Prerequisite: Previous experience as a Linux or Windows administrator will be a plus.
- Format: Multiple choice and multiple answers
- Duration: 130 minutes
- Exam Cost: \$150
- Exam Language: Available in English, Japanese, Korean, and Simplified Chinese

After the End of this course, you will learn,

- How to deploy applications for the AWS platform
- Learn how you can send and receive data between data centers and AWS
- Selecting the appropriate AWS services to meet the requirement of your business
- Provisioning, managing, maintaining, and securing systems in an AWS environment

AWS solution Architect- Professional

A professional AWS architect is an AWS certification course for someone who needs to evaluate an organization's demands and make architectural recommendations for implementing and deploying applications on AWS.

Exam Details:

- Prerequisites: You should be Certified Solutions Architect – Associate to attain this exam.
- At least two years of hands-on experience designing and deploying cloud architecture on AWS and best practice knowledge of multi-application architectural design is strongly recommended.
- Format: Multiple-choice, multiple-answer
- Time: 170 minutes
- Cost: 300 USD

After the End of this course, you will learn,

- Learn about best practices for architecting and the designing of applications on AWS
- Techniques for migrating complex application systems to AWS
- Picking the right AWS service for the requirements of an application
- Knowledge of cost optimization strategies

AWS Certified DevOps Engineer – Professional

The DevOps Engineer certification course is all about operating, managing, and build on the AWS platform. The course mainly two fundamental concepts of Develops continuous delivery and automation of processes.

Exam Details:

- Prerequisites: It is an associate-level certification and a minimum of five years of relevant experience
- Format: Multiple-choice, multiple-answer
- Exam time: 170 Minutes
- Cost: 300 USD

At the End of this AWS certification, you will learn,

- The concept of continuous developer and modern CD methodologies
- Techniques for implementing CD systems
- Set up, monitoring, and logging and maintaining systems on AWS server

AWS Certified Big Data – Specialty

The AWS certified Big Data- specialty certification is suited for those who have a background in data analytics and experience using various types of AWS service for designing the most reliable big data solutions.

- Prerequisite: At least two years of hands-on experience using Amazon web services.
- Exam Duration: 170 minutes
- Format: Multiple choice questions and multiple-answer
- Exam Fees: 300 USD

At the End of this AWS certification, you will learn,

- AWS Architecting best practices for implementing big data services solutions offered by AWS
- AWS tools for automating data analysis
- You will learn how to access control to secure the data.
- The course covers how to design and to maintain big data applications.
- AWS services covered include: Kinesis, Athena, Quick sight, and Recognition

AWS Certified Advanced Networking – Specialty

This AWS certification course is designed to validate a candidate's skills and experience in connection with performing complex networking tasks on AWS.

Exam Details:

- Prerequisite: Candidates should have a background in architecting and implementing network solutions.
- Format: Multiple-choice and multiple-answer
- Duration: 170 minutes
- Exam Language: Available in English, Japanese, Korean, and Simplified Chinese
- Exam Cost: 300 USD

At the End of this AWS certification, you will learn,

- Designing, developing, and deploying AWS cloud solutions
- Implementing core services as per the architectural best practices
- Automation of AWS tasks for network deployments
- Security and compliance design and implementation
- Learn about network optimization and troubleshooting

AWS Certified Security – Specialty

The AWS security -the course covers topics that security pros and teams need to master security fundamentals. It follows best practices and builds in-depth knowledge of key services unique to the AWS platform.

It covers topics like protection and encryption, infrastructure security, access management, monitoring, and logging, etc.

Exam Details:

- Prerequisite: At least two years hand-of-experience sourcing AWS workloads. Security control for the workload on AWS. At least five years of experience of designing and implementing security solutions
- Exam Format: Multiple choices, Multiple-answer
- Exam time; 170 minutes
- Exam Cost: 300 USD
- Exam Language: Available in English, Japanese, Korean, and Simplified Chinese

At the End of this AWS certification, you will learn,

- Concept of specialized data classifications and AWS data protection mechanisms.
- Helps you to understand data encryption methods and AWS mechanisms to implement them.
- How to manage secure Internet protocols and AWS mechanisms for implementation.
- You will able to take a right with regard to cost, security, and deployment complexity given a set of application requirements.
- Implement logging and monitoring to detect and analyze security vulnerabilities within your infrastructure

Learning Objective Of AWS Certification

Here, are the most important objective of learning AWS course:

- Allows you to Identify the security and compliance benefits of by using the AWS Cloud.
- Allows you to understand the access control and management features of AWS.
- You can learn about data encryption methods to secure all types of sensitive data.
- Get knowledge about the important steps for managing various AWS resources.
- You can use AWS services to protect network security.
- You will get the knowledge about how to audit an AWS environment.
- Explain the AWS compliance and assurance programs.
- Formulate solution plans and learn about AWS architectural best practices

How To Start For AWS Certificaiton?

While there are not standard define steps to start AWS certification, below-given steps are the most straight-forward.

Step 1)

- First of all, you need to Enroll yourself in an AWS training class.
- Select the desired module that you wants to take.

Step 2) Review all the available study materials and Exam Guides related to selected AWS module.

Step 3) Read multiple AWS whitepapers. It offers plenty of crucial information regarding topics. These hold some useful information, which may answer your questions.

Step 4) Next, you need to take regular practice. A practice test will help you to become free of stress about the AWS certification exams.

Step 5) Schedule the final AWS certification exam once you are ready. It generally takes around 80-120 hours of practice/studying to be prepared for the exam. However, it depends on your experience and the certification course that you have selected.

Exam Preparation

These training courses and materials will help with exam preparation:

- Architecting on AWS instructor-led, live or virtual 3-day course
- AWS Whitepapers faws.amazon.com/wltepap, Kindle.pdf, and Other Materials
- Identify AWS services which help you to automate, monitor, and manage security operations on AWS.
- AWS Well-Architected web page (various whitepapers linked)

Exam Content

There are mainly two types of questions on the examination:

- Multiple-choice: It has one correct and three incorrect responses
- Multiple-response: Has two correct responses out of five options.

Current Salary for AWS Certified Professionals

Here, is the latest salary by AWS certificate in USA

- AWS Certified Solutions Architect – Associate: \$121,292
- AWS Certified Solutions Architect – Professional: \$142,160
- AWS Certified Developer – Associate: \$114,148
- AWS Certified DevOps Engineer – Professional: \$118,395
- AWS Certified SysOps Administrator – Associate: \$142,160

• [Prev](#)

[Report a Bug](#)

• [Next](#)

Azure vs. AWS: Key Differences

What is Azure?

Azure is an open source and flexible cloud platform which helps in development, service hosting, service management, and data storage. The Azure cloud computing tool hosts web applications over the internet with the help of Microsoft data centers.

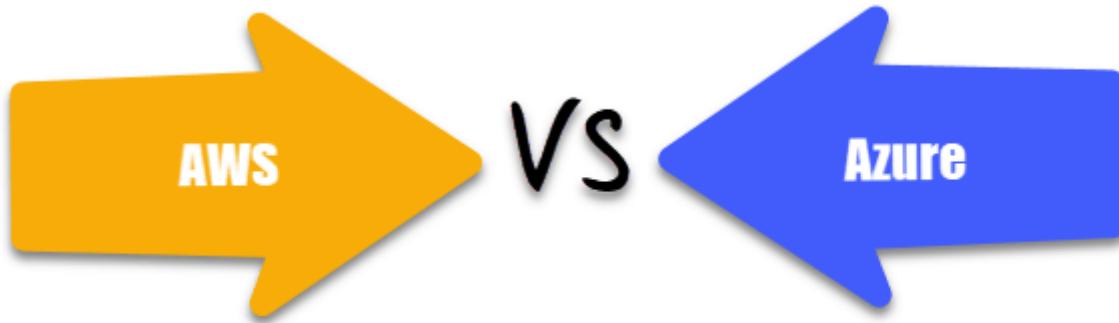
In this tutorial, you will learn

- [What is Azure?](#)
- [What is Aws?](#)
- [Comparison between Azure and AWS](#)
- [Popularity Index with Market Share](#)
- [Advantages of AWS](#)
- [Advantages of Azure](#)
- [Disadvantages of AWS](#)
- [Disadvantages of Azure](#)
- [Which one is better?](#)

What is Aws?

Amazon Web Services is widely used secure cloud services platform, offering computing power, content delivery, database storage, and other functionality to help businesses scale and grow.

Comparison between Azure and AWS

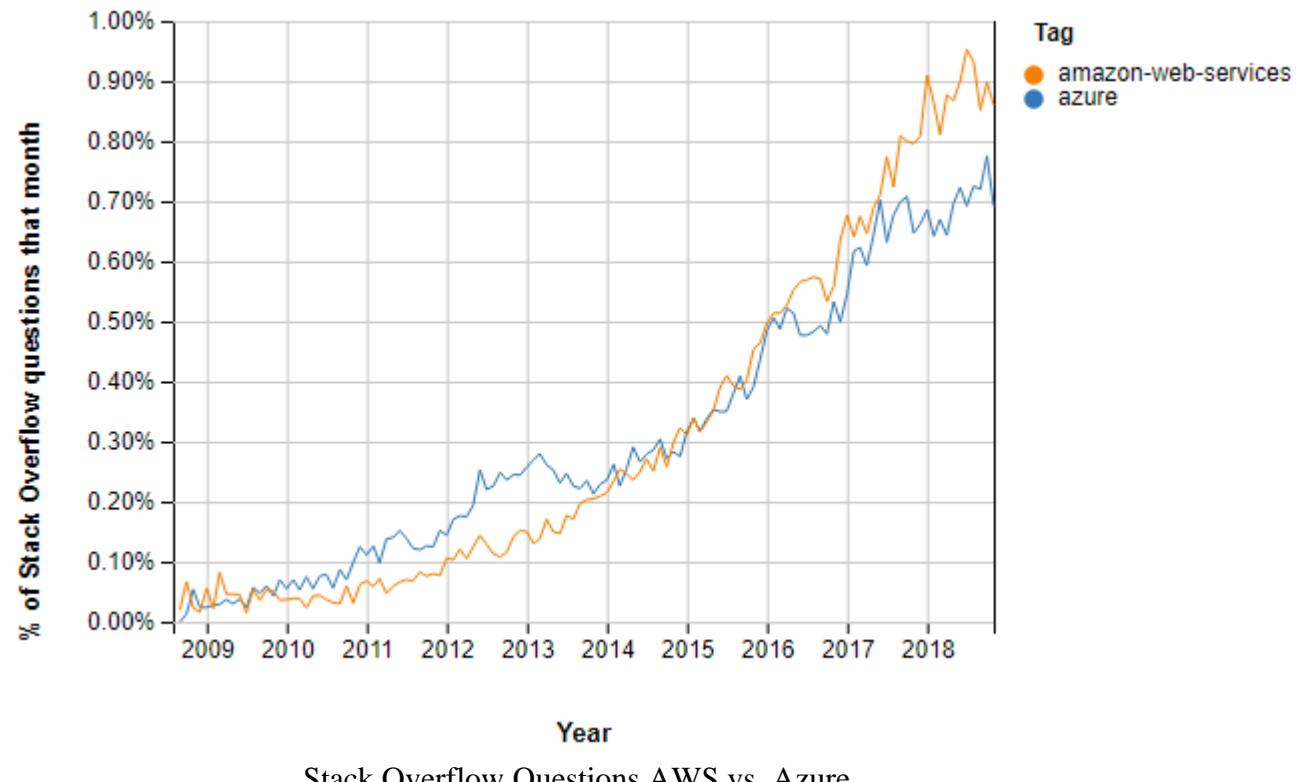


Parameters	AWS	Azure
Launched Date	Launched in 2006.	Launched In 2011
Market Share	31% Share of the global computing market	11% Share from
Availability Zone	61 Availability Zone	140 Availability
Storage services	<ul style="list-style-type: none"> • S3 • Buckets • EBS • SDB • domains • Easy to use • SQS • CloudFront • AWS Import/Export 	<ul style="list-style-type: none"> • Blob Storage • Containers • Azure Data Lake • Table Storage • Tables • Storage Services
Databases Services	<ul style="list-style-type: none"> • MySQL • Oracle • DynamoDB 	<ul style="list-style-type: none"> • MS SQL • SQL Server
Deployment Services	<ul style="list-style-type: none"> • Amazon Web Services • Amazon Machine Instance (AMI) • Traditional Deployment Models • Fine-grained updates • Elastic Beanstalk 	Cspkg (fancy zip file) for fine-grained updates

	<ul style="list-style-type: none"> • Cloud Formation 	
Networking Services	<ul style="list-style-type: none"> • IP/Elastic IP/ELB • Virtual Private Cloud • Route 53 • ELB • Firewall heavily configurable 	<ul style="list-style-type: none"> • Automatic scaling • Load-balancing • Azure Cloud Services • Balancing • Endpoint
Price	Per hour- rounded up	On-demand reservation
Customers	Adobe, Airbnb, Expedia, Yelp, Nokia, Netflix, Novartis.	Pearson, 3M, Toyota
Type of Cloud	Virtual Private Cloud (VPC)	Virtual Network
Connection type	Direct Connect	ExpressRoute
Pricing models	<ul style="list-style-type: none"> • Free Tier • Per Hour • Free Trial Per Minute • No charge for stopped instances • Pay for EBS volume 	<ul style="list-style-type: none"> • Free Trial • Per Minute
Government Cloud	AWS has an edge as far as government cloud offerings.	Limited reach for government
Support for Hybrid cloud	Does not offer the best of hybrid cloud support.	With Hybrid Cloud instances.
Ecosystem	AWS has a software marketplace with an extensive partner ecosystem.	With very few Local partners
Support for Big Data	EBS storage is ideal for handling big data.	Standard storage and premium storage
Maturity	More mature cloud environment for big data.	The less mature environment
Machine access	In AWS machine can be accessed separately.	Machines are grouped by name with various regions
Salary	The average salary for "AWS engineer" is approximately \$141,757 per year for Software Architect.	The average salary for "AWS architect" is approximately \$113,582 per year

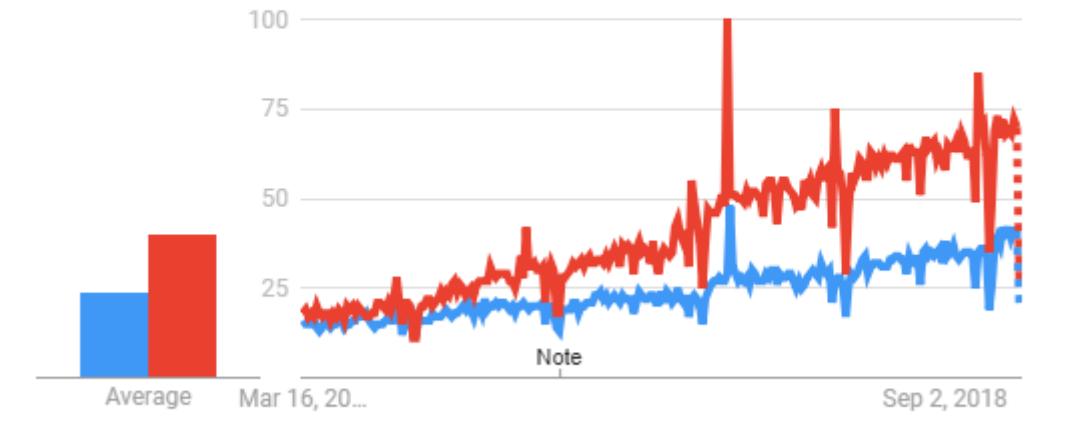
Key features	Zero setups, Detail Monitoring, Auto-scaling groups.	Startup friendly,
Long term data archiving	Allows long term data archiving and retrieval.	Does not offer a
Security	Security is provided using defined roles with permission control feature.	Provides security

Popularity Index with Market Share



AWS continues to dominate a global cloud-infrastructure services industry which is likely to hit proximately \$70 billion last year. Today, it enjoys market share which is better than some top public cloud providers.

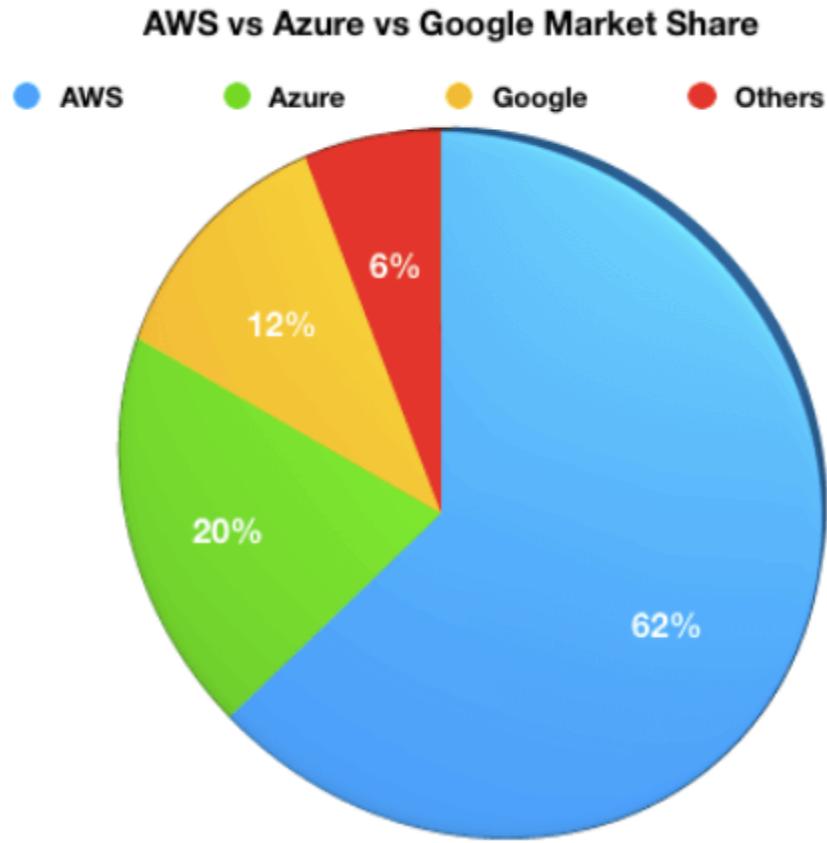
● azure ● aws



United States. Past 5 years. Web Search.

Google Trends Azure vs. AWS

In the past year, Azure's cloud adoption rate is almost 85 percent that of AWS', up from 70 percent last year.



Advantages of AWS

Here, are significant advantages of adopting AWS cloud services:

- Compute Cloud allows you to increase or decrease storage according to the need of your organization
- AWS enables you to select an operating system, programming language, database of your choice.
- Broad & deep service offerings
- Robust partner ecosystem
- Trusted by high-profile customers
- High Transfer Stability
- Minimal information is lost during server and storage transfer
- Offers more data centers for availability and low latency
- Better DevOps support
- Simpler licensing method
- Stronger support for BI and analytics

Advantages of Azure

Here, are some major advantages of using Azure cloud services:

- Capability for developers and users to create, maintain and deploy applications
- Fully scalable cloud computing platform offers open access across multiple languages, frameworks, and tools
- Total support for Microsoft legacy apps
- Greater awareness of enterprise needs
- Easy one-click migrations in many cases
- Conversion of on-prem licenses to the cloud
- Support for mixed Linux/Windows environments
- Offers inbuilt tool like Azure stack to help the organization deliver Azure service from the own data center

Disadvantages of AWS

Here, are few drawbacks of Amazon Web Services:

- Less hybrid- cloud-friendly
- AWS elastic load balancer is not equipped to handle as many requests as it receives
- AWS lacks customer support, so it more suitable for a technically savvy group of consumers and those companies who have their inbuild tech support team
- The number of choices offered by AWS is confusing to those who may not speak the language of technology.
- Incompatible and Weak Hybrid Strategy
- AWS is a less open private cloud. This makes it an unpopular storage option for sensitive industries like banking
- AWS has too many products which makes the selection process much harder

Disadvantages of Azure

The major Drawbacks of Azure cloud services are:

- Customer service is not transparent, and data is hosted globally. So, if you have data restrictions where it must be stored in a specific country, at that time you need to verify/specify with Microsoft
- You will be charged extra for paying as you go
- Azure cloud-based services are full of glitches. To fix these bugs, you will need to spend additional money
- Less flexibility about non-Windows server platforms, when compared to AWS

Which one is better?

Microsoft Azure has increased its market share in the last couple of years, but not to an extent where there is a real contest between the two companies at least for the near future.

Moreover, both companies introduce new products, new integrations, and new pricing structures. Therefore, the final selection will be depend on the need of your organization.

KEY DIFFERENCE

- Both Azure and AWS supports hybrid cloud but Azure supports hybrid cloud better.
- Azure offers express routes while AWS offers direct connection.
- Azure provides security by offering permissions on the whole account whereas AWS security is provided using defined roles with permission control feature.
- Azure machines are grouped into cloud service and respond to the same domain name with various ports whereas the AWS machine can be accessed separately.
- Azure has a virtual network cloud whereas AWS has Virtual Private Cloud.
- Azure has 140 availability zone whereas AWS has 61 availability zone.

• [Prev](#)

[Report a Bug](#)

• [Next](#)

Heroku vs AWS: 10 Most Important Differences You Must Know!

What is AWS?

Amazon web service is a platform that offers flexible, reliable, scalable, easy-to-use and cost-effective cloud computing solutions.

AWS has a massive collection of cloud services that build up a fully-fledged platform. It is known as a powerhouse of storage, databases, analytics, networking and deployment/delivery options offered to developers. AWS Cloud is available in 16 different geographic regions, and the number is increasing.

In this tutorial, we will learn,

- [What is AWS?](#)
- [What is Heroku?](#)
- [Why use Heroku?](#)
- [Why choose Amazon Web Services?](#)
- [AWS vs. Heroku comparison table](#)
- [Disadvantages of Heroku](#)
- [Disadvantages of AWS](#)
- [The Verdict](#)

What is Heroku?

Heroku is a cloud service provider and software development platform which facilitates fast and effective building, deploying and scaling of web applications. It has 140 inbuilt add-ons, ranging from alerts, analytic tools security services which are used for purpose like monitoring, caching and mailing or networking add-ons.

The tool can provide you with built-in instant run-time application services. Moreover, you don't need to think about infrastructure because it managed automatically by the software itself. Heroku is owned by Salesforce.



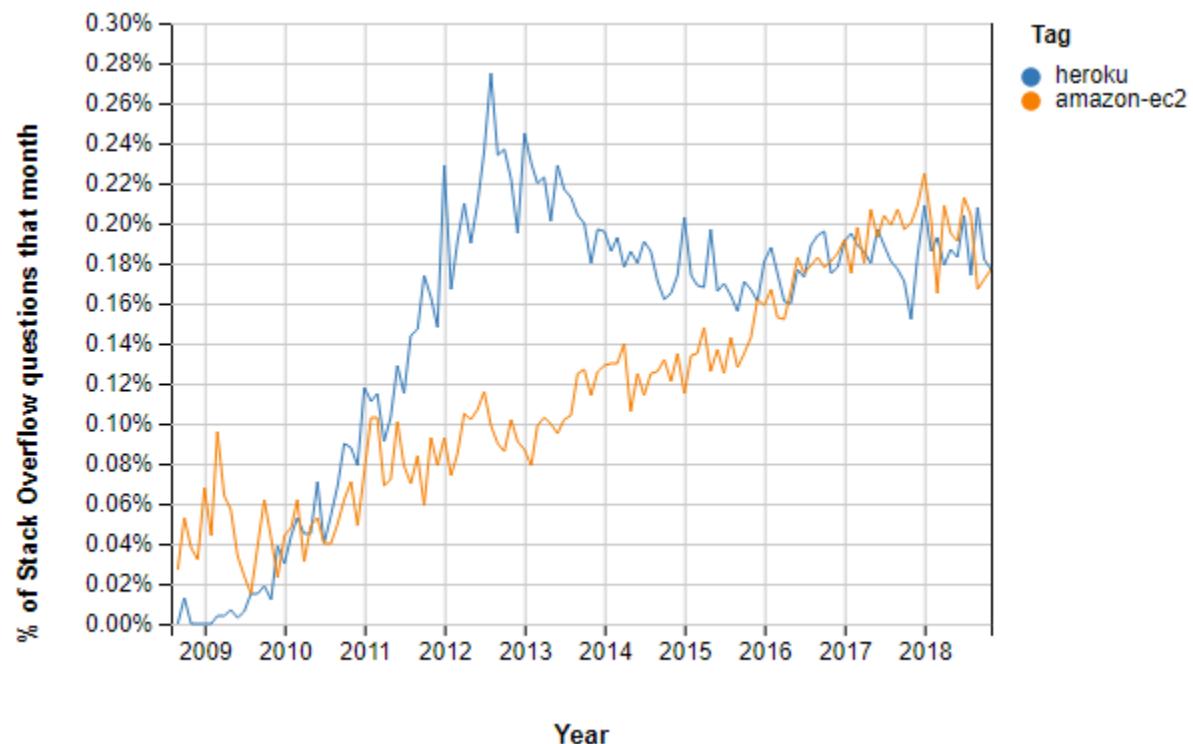
Heroku provides

- Predictability and insight into application maintenance support elements
- Clear accountability and SLAs
- Decreased risk
- Ability to focus on innovation over operations

You don't need to manage the AWS environment – it's all done transparently as part of the Heroku experience

Heroku is a Platform as a Service built on Top of AWS

Why use Heroku?

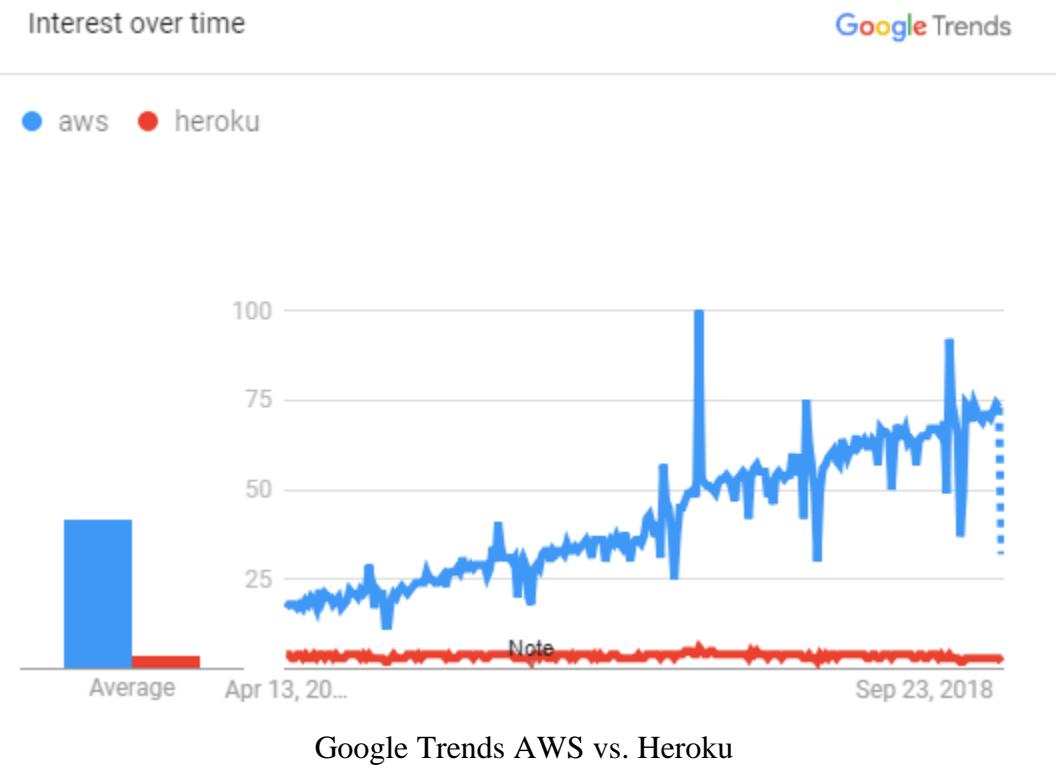


Stack Overflow Questions Heroku vs Amazon

Here are pros/benefits of using Heroku:

- Allows the developer to focus on code instead of infrastructure
- Enhance the productivity of cloud app development team
- Offers single billing for all projects broken down by team
- Monitor and enhance performance though rich application monitoring
- Helps your development, QA, and business stakeholders create a unified dashboard.
- Support form Modern Open Source Languages
- High-performance Salesforce integration
- Simple Horizontal & Vertical Scalability
- Heroku operation and security team is instantly ready to help you 24/7
- Leading Platform tools and Services Ecosystem
- Helps you to focus on innovation, not operations
- The Heroku Enterprise architecture offers minimal or no downtime during the system updates.
- Fast application lifecycle management and permissions
- Allows you to remove friction from the development
- Offers a powerful dashboard and CLI
- Integrates with familiar developer workflows
- Predictability and insight into the cost of application development and maintenance
- A bunch of supportive tools
- Beginner and startup-friendly
- It allows you to create a new server in just 10 seconds by using the interface of Heroku Command Line.
- This cloud computing platform takes care of patching systems and keeping everything healthy.
- A range of automated functionalities including the scaling, configuration, setup, and others
- Easy integration with other AWS products
- Medium learning curve
- Offers best-in-class Developer Experience
- Secure connectivity to Salesforce data to build a single view of the customer
- Flexibility to customize and support unique DevOps workflow needs

Why choose Amazon Web Services?

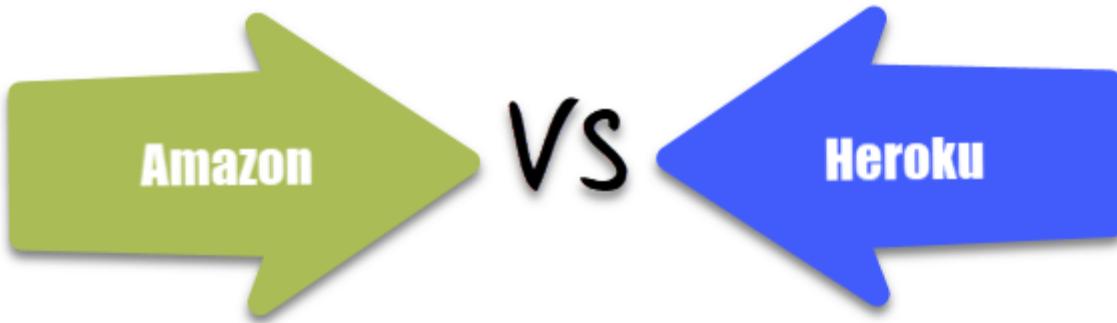


Here are the pros/benefits of selecting AWS web services:

- You should opt for AWS when you have DevOps teams who can configure and manage the infrastructure
- You have very little time to spend on the deployment of a new version of your web or mobile app.
- AWS offers easy deployment process for an app
- AWS web service is an ideal option when your project needs high computing power
- Helps you to improve the productivity of the application development team
- A range of automated functionalities including the configuration, scaling, setup, and others
- Use AWS you have full control over the resources.
- AWS allows organizations to use the already familiar programming models, operating systems, databases, and architectures.
- It is a cost-effective service that allows you to pay only for what you use, without any up-front or long-term commitments.
- You will not require to spend money on running and maintaining data centers.
- Offers fast deployments
- You can easily add or remove capacity.
- You are allowed cloud access quickly with limitless capacity.
- Total Cost of Ownership is very low compared to any private/dedicated servers.
- Offers Centralized Billing and management
- Offers Hybrid Capabilities

- Allows you to deploy your application in multiple regions around the world with just a few clicks

AWS vs. Heroku comparison table



Parameters	Heroku
Owner	Salesforce.com
Hosted on	Amazon's data centers
Type of Service	PaaS
Pricing	Heroku costs \$0.05 per hour.
Languages	Node.js, Java, Ruby, PHP, Python, Go, Scala, Clojure
Geographic Regions	Europe, USA, Australia, Japan, etc.
Features	<ul style="list-style-type: none"> • Fully flexible runtime environment with smart containers (dynos) system • Allows manual horizontal and vertical scaling • Allows you to roll back your database or code in no time. • App monitoring system to keep track of metrics, like response time, throughput, memory, etc.
Best for	Startups, Medium Businesses, Large Enterprises
Companies using	Toyota, Citrix, Westfield, Yesware, etc.
Complexity	It's software some time too simple, even for professional developers.
Built-in Tools for Management and Monitoring	<ul style="list-style-type: none"> • Heroku Command Line • Heroku Application Metrics • Heroku Connect • Heroku Status
Heroku Service Level Agreements	Available for Enterprise projects
Rapid deployment	Heroku offers you a ready-to-use environment which allows you to push your code and make a few configuration changes to get your application running.

Parameters	Heroku
Need DevOps Engineer	Not at all
Development of server	The creation of a server is a simple process.
Computational demands	Low

Disadvantages of Heroku

Here, are cons/drawbacks of using Heroku:

- Heroku dynos are frequently unreachable for various reasons
- Inbound and outbound latency is high.
- It offers low network performance.
- Heroku does not allow you to run any other services on dynos.
- To purchase additional dynos/workers, you need to pay \$35 a month which is quite costly.
- It proves to be expensive for large and high-traffic apps.
- Limited in types of instances
- Not ideally suited for heavy-computing projects.

Disadvantages of AWS



- Unreliable deployment with no error description
- AWS deployment process is not easy and very lengthy which can take up to 15 to 20 minute for a simple website
- AWS is not an ideal option for start-ups that are **not** tech-savvy
- Less frequent updates and new stack versions.
- You need to deploy your application yourself using recipes, Capistrano, or manually.
- Launching multiple app instances is not an easy process in AWS.

The Verdict

Heroku and AWS both are excellent platforms. However, you need to understand what kind of feature your organization needs and how much you are ready to pay for them. You also need to ascertain the skills sets (especially DevOps) in your team before you

make a choice. The right platform is one which helps you achieve your business goal and at the same time keeps your developers productive.

KEY DIFFERENCE

- Heroku is container-base cloud platform offering (PaaS) whereas AWS is a secure cloud services platform providing IaaS, PaaS and SaaS.
- Heroku offers you a ready-to-use environment that allows you to deploy your code fast whereas the deployment process of AWS service is quite complicated.
- Heroku is best suitable for Startups, Medium Businesses whereas AWS is mainly focused on Medium Businesses and Large Enterprises.
- Heroku can meet low computational demands whereas AWS can meet high/very high computational demands.
- Heroku doesn't need infrastructure maintenance whereas AWS needs a dedicated DevOps guy.
- Comparatively, Heroku supports less Geographical Regions.

• [Prev](#)

[Report a Bug](#)

• [Next](#)

Digitalocean vs AWS: 10 Most Important Differences You Must Know!

What is DigitalOcean?

DigitalOcean is a simple cloud service provider. It has a simple set-up and very affordable in price. It allows developers to accomplish a task like spinning up a server(droplet) in a fraction of time it takes on other platforms.

In this tutorial, you will learn,

- [What is DigitalOcean?](#)
- [What is AWS?](#)
- [Why DigitalOcean?](#)
- [Why Amazon?](#)
- [History of DigitalOcean](#)
- [History of Amazon](#)
- [DigitalOcean Vs. Amazon](#)
- [Disadvantages of DigitalOcean](#)
- [Disadvantages of AWS](#)
- [Who is the winner?](#)

What is AWS?

Amazon Web Service(AWS) is a platform which offers flexible, reliable, scalable, easy-to-use, and cost-effective cloud computing solutions.

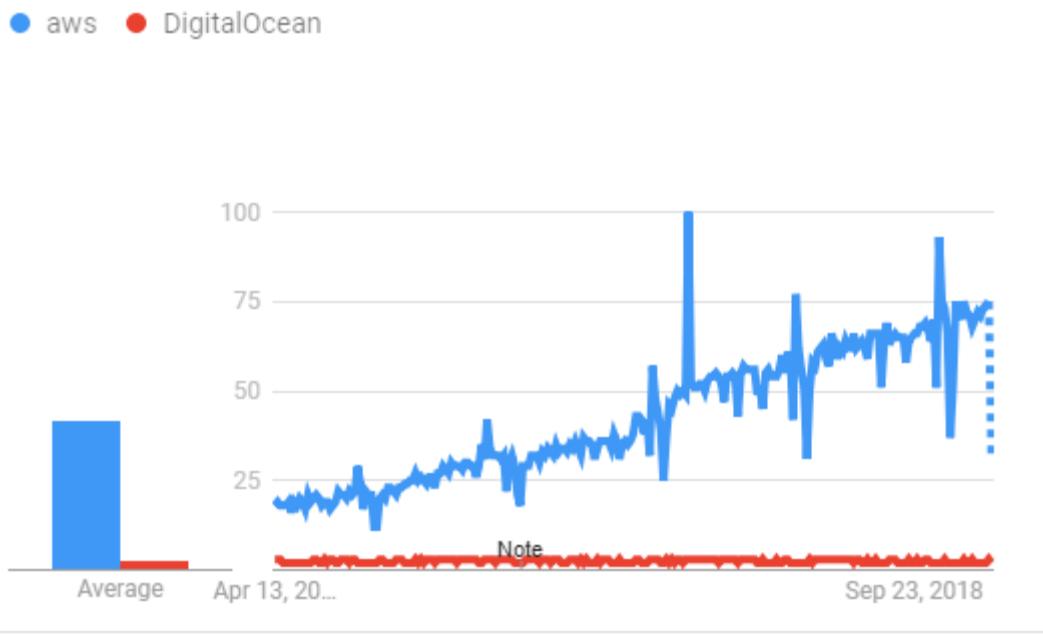
Amazon Web Services is a huge collection of cloud-computing services that build up a fully-fledged platform. It is known as a powerhouse of storage, databases, analytics, networking, and deployment/delivery options for developers. AWS Cloud is available in 16 different geographic regions.

Why DigitalOcean?

Here, are the most prominent advantages/benefits for using DigitalOcean:

- Allows you to create multiple server instances using the same account
- Offers great performing servers
- Easy to set up and provides support of various operating systems
- DigitalOcean is an ideal option for those people who provide managed hosting services for web applications or websites in shared environments
- Great tool for tech-savvy start-ups
- Works on What You See Is What You Pay model
- User-friendly management interface
- Pricing is very affordable and scalable
- Offers well documented FAQ and tutorials

Why Amazon?



United States. Past 5 years. Web Search.

Google Trends Digital ocean vs. AWS

- AWS Compute Cloud allows you to increase or decrease storage as per the requirement of your organization
- AWS enables you to select an operating system, programming language, database of your choice.
- Broad and deep service offerings
- Robust partner ecosystem
- Simple and transparent licensing method
- High network reliability with low latency
- Minimal information is lost during the process of server and storage transfer
- Better DevOps support
- Support for Business Intelligence and analytics

History of DigitalOcean

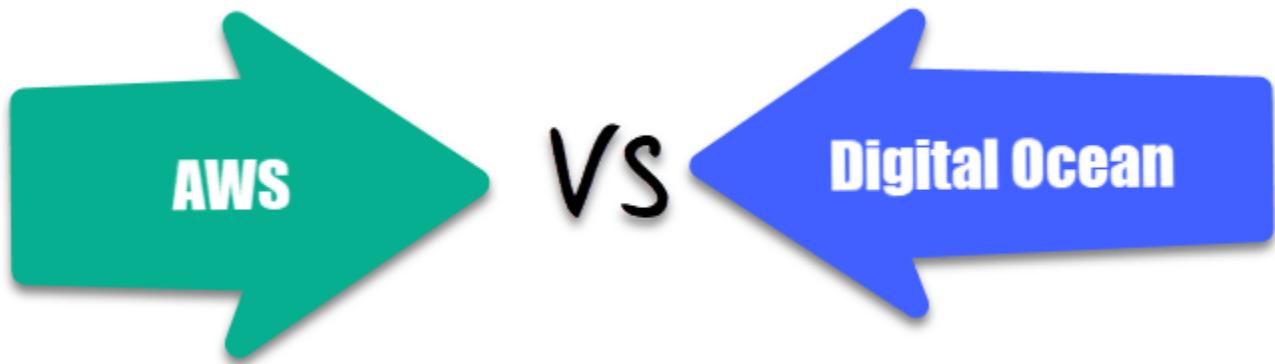
- In 2003, Moisey Uretsky and Ben founded ServerStack, a managed web hosting business.
- In 2011 the Uretskys founded DigitalOcean. This firm offered server provisioning and cloud hosting services.
- In August 2012, the company got traction and launched around 10,000 cloud server instances.

- By May 2015, DigitalOcean expanded further with a new data center in Toronto, Canada.
- In May 2018, the DigitalOcean announced the launch of Kubernetes-based container service.

History of Amazon

- 2002- A launch of AWS services
- 2006- Launched its cloud products
- 2015- Able to achieve revenues of \$4.6 billion
- 2016- Surpassed 10 billion dollar revenue target
- 2016- Release snowball and snowmobile
- 2019- Offers More than 100 cloud services

DigitalOcean Vs. Amazon



Parameters	DigitalOcean	Amazon
Focus area	Developers and small applications	Everybody works
Best feature	Best pricing and quicker server deployments	A high variety
Security	DigitalOcean offers physical security in their data centers, which is a plus. Their centers are secured infrastructures protected from physical threat to prevent unauthorized entry.	AWS has plenty GuardDuty which also helps you
Type of Cloud	DigitalOcean is Infrastructure as a Service (IaaS). It allows you to manage security, databases, and even operating system.	AWS is Platform as a Service. It contains
Compatibility With Windows Operating System	DigitalOcean is not compatible with Microsoft Windows OS. It works only with predefined Linux installation.	Amazon EC2
Who it for?	If you want a MySQL instance, then you should opt for DigitalOcean.	If you are a developer, have spare time, and need a right option for your project.
Accepted forms of Payment	Credit cards, Paypal.	Only Credit Cards
Mobile Friendly UI	Yes	No

Parameters	DigitalOcean	Amazon
Company using the Technology	Airbnb, Medium, Pinterest, Reddit, etc.	
Key Features	Auto-scaling, reliable, easy management.	Simple dashboard
Pricing	Pricing starts from \$5 to \$640 per month for entry-level server	Pricing starts

Disadvantages of DigitalOcean

Here, are cons/drawbacks of using DigitalOcean:

- You can't install the system by yourself or provide your ISO
- Security issues may occur with SSH host keys.
- Does not have a SAN but instead uses local storage in RAID
- DigitalOcean has fewer regions, compared to AWS.
- DigitalOcean does not offer any cloud computing instances that have GPUs.
Useful for AI, machine learning, and Data sciences training.
- Errors and bugs were not appropriately addressed.

Disadvantages of AWS

Here, are cons/drawbacks of using AWS services:

- AWS is not an ideal option for start-ups that are **not** tech-savvy
- Lots of hidden costs
- AWS best suited for companies who want a datacentre, not a server
- Less hybrid-cloud friendly. Incompatible and weak hybrid strategy
- Unreliable deployment with no error description
- Launching multiple app instances is not an easy process in AWS.
- AWS is a less open private cloud. This makes it an unpopular storage option for sensitive industries like finance and banking.
- Customer Support is expensive.

Who is the winner?

In reality, DigitalOcean can't be considered a competitor of Amazon as its target market is a small development team who wants to stage up small high-performance instance quickly. Many select AWS because of its popularity.

However, If you are looking for a cost-effective cloud computing solution, then DigitalOcean is undoubtedly the best option for you. But, if you want multiple tools for your business, then you should opt for AWS services.

KEY DIFFERENCE

- DigitalOcean provides Infrastructure as a Service (IaaS) whereas AWS provides Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS).
- DigitalOcean is a simple cloud service provider while Amazon Web Service(AWS) is a platform that offers flexible, reliable, scalable, easy-to-use, and cost-effective cloud computing solutions.
- DigitalOcean offers Auto-scaling, reliable, easy management, on the other hand, AWS offers Simple dashboard, Great community, Easy Configuration and many more.
- DigitalOcean is more suitable for developers and small applications while AWS is suitable for big scalable applications.

• [Prev](#)

[Report a Bug](#)

• [Next](#)

Google Cloud vs AWS: (2020 Comparison)

What is AWS?

Amazon web service is a platform that offers flexible, reliable, scalable, easy-to-use, and cost-effective cloud computing solutions.

AWS cloud computing platform offers a massive collection of cloud services that build up a fully-fledged platform. It is known as a powerhouse of storage, databases, analytics, networking, and deployment/delivery options offered to developers.

In this tutorial, you will learn

- [What is AWS?](#)
- [What is Google Cloud?](#)

- Why AWS?
- Why Google Cloud?
- Features of AWS
- Features of Google Cloud
- AWS vs. Google Cloud Platform (GCP)
- Disadvantages of AWS
- Disadvantages of Google Cloud
- How to Pick Your Cloud

What is Google Cloud?

Google launched the Google Cloud Platform(GCP) in 2011. This cloud computing platform helps a business to grow and thrive. It also helps you to take advantage of Google's infrastructure and providing them with services that is intelligent, secure, and highly flexible.

Why AWS?



Here are the important pros/benefits of selecting AWS web services:

- AWS offers easy deployment process for an app

- You should opt for AWS when you have DevOps teams who can configure and manage the infrastructure
- You have very little time to spend on the deployment of a new version of your web or mobile app.
- AWS web service is an ideal option when your project needs high computing power
- Helps you to improve the productivity of the application development team
- A range of automated functionalities including the configuration, scaling, setup, and others
- It is a cost-effective service that allows you to pay only for what you use, without any up-front or long-term commitments.
- AWS allows organizations to use the already familiar programming models, operating systems, databases, and architectures.
- You are allowed cloud access quickly with limitless capacity.

Why Google Cloud?



Here are the pros/benefits of selecting Google cloud services:

- Offers higher productivity gained through Quick Access to innovation
- Employees can work from Anywhere
- Future-Proof infrastructure

- It provides a serverless environment which allows you to connect cloud services with a large focus mainly on the microservices architecture.
- Offers Powerful Data Analytics
- Cost-efficiency due to long-term discounts
- Big Data and Machine Learning products
- Offers Instance and payment configuration

Features of AWS

Important features of AWS are:

- Total Cost of Ownership is very low compared to any private/dedicated servers.
- Offers Centralized Billing and management
- Offers Hybrid Capabilities
- Allows you to deploy your application in multiple regions around the world with just a few clicks

Features of Google Cloud

Important features of Google Cloud are:

- Constantly including more Language & OS.
- A better UI helps you to improves user experience.
- Offers an on-demand self-service
- Broad network access
- Resource pooling and Rapid elasticity

AWS vs. Google Cloud Platform (GCP)



Here are major differences between AWS and Google Cloud.

Parameters	AWS	Google Cloud
Age	11 years	6 years
Pricing	Per hour basis	Per-minute basis

Parameters	AWS	Google Cloud
What is?	AWS is a secure, cloud service the platform developed and managed by Amazon.	Google Cloud Platform is a suite of services.
Containers	Docker, Kubernetes	Kubernetes Only
Best feature	It dominates the public cloud market by offering a range of cloud-based products and services.	It specializes in high compute scene.
Main service	The flagship compute service of AWS is Elastic Compute Cloud, or EC2.	Google's primary service is Google Cloud Functions.
Cost	It's on a little higher end in terms of compute and storage costs.	It's a clear winner with its cost efficiency.
Object Storing	Amazon Simple Storage Services also called (AWS S3)	Google Cloud Storage
VM disk storage	Amazon Elastic Block Store (Amazon EBS)	Persistent Disk (both HDD and SSD)
File storage	Amazon Elastic File System (Amazon EFS)	Cloud Filestore
Archive Storage	S3 One Zone- Infrequent Access. Amazon Glacier offers data querying capabilities.	Archival Cloud Storage.
Bulk Data Transfer	AWS Import/Export Disk: Shipping disk drives AWS Snowball (device-based) AWS SnowMobile: Exabyte- scale data transfer via ruggedized shipping container AWS is the clear winner here.	Nearline (low frequency) Coldline (lowest frequency)
Hybrid support	AWS Storage Gateway offers managed virtual tape infrastructure for a hybrid environment.	Relies on partners such as EMC and NetApp.
Disaster Recovery Management	Offers cloud-based disaster recovery services.	Offers out-of-the-box DR options.
Backup	Amazon S3 is used for secondary backup storage	Has its own built-in Google Cloud Backup.
Serverless computing	AWS Lambda	Container Engine
Volume Sizes	500GB to 16TB	1 GB to 64 TB
Max IOPS per volume	500	3000 read 15000 write
AI service and Machine Learning	AWS has its own AI service, called SageMaker. Compared to Google, the AI offering of AWS is limited.	Google cloud used AI chips and libraries. The speed and performance is better than Google.
Networking framework	AWS has no tiered networking framework.	GCP has its tiered networking framework.
Data transmission format	In general format.	In a fully encrypted format.
Big data support	AWS has its own big data analysis tool which is called AWS Lambda.	Google Cloud Platform has Big Data services.
Available zones	AWS services are available in 21 different zones.	Google Cloud Platform services are available in many regions.
Companies using	AppDirect, Eat with Ava, Icarros, Valera, etc.	Bugsnag, Atomcert, Policygenius, and more.

Disadvantages of AWS

Here, are drawbacks/cons of using AWS cloud:

- AWS deployment process is not easy and very lengthy which can take up to 15 to 20 minute for a simple website
- Unreliable deployment with no error description
- AWS is not an ideal option for start-ups that are **not** tech-savvy
- You need to deploy your application yourself using recipes, capistrano, or manually.
- Launching multiple app instances is a very difficult process in AWS.

Disadvantages of Google Cloud

Here, are drawbacks/cons of using Google Cloud:

- Small components, difficult to start
- Out of Free Tier, everything costs.
- Lacks features compared to AWS.

How to Pick Your Cloud

Google Cloud and AWS both are good platforms. So, before deciding you should understand what type of feature your organization needs and how much you want to pay for them. Remember that, the right cloud service provider helps you achieve your business goal by enhancing the functioning of your organization. Since, AWS is the market leader in features and reliability our vote will tilt towards them.

KEY DIFFERENCE

- Google Cloud is a suite of Google's public cloud computing resources & services whereas AWS is a secure cloud service developed and managed by Amazon.
- Google Cloud offers Google Cloud Storage, while AWS offers Amazon Simple Storage Services.
- In Google cloud services, data transmission is a fully encrypted format on the other hand, in AWS, data transmission is in the general format.
- Google Cloud volume size is 1 GB to 64 TB while AWS volume size is 500 GB to 16 TB.
- Google Cloud provides backup services, but AWS offers cloud-based disaster recovery services.

- [Next](#)

20 BEST AWS Competitors & Alternatives in 2020

AWS is Amazon's cloud computing platform that offers fast, flexible, reliable, and cost-effective solutions. It also offers a service in the form of building blocks which can be used to create and deploy various types of applications in the cloud.

However, AWS services set default limits on a resource which differ from region to region. Also, hardware-level changes happen to your application, which might not offer the best performance and usage of your application.

Here, is a list of Top 20 AWS alternatives which are good enough to replace AWS cloud computing services. This list consists of paid and open-source tools with important features and the latest link.

1) Kamatera



A cloud server tool developed by [Kamatera](#) is very much similar to a physical server. It operates in a virtual infrastructure cloud, making it highly flexible and cost-effective. This cloud server pricing is based on pay as you use model a standard in the industry.

Features:

- 13 Data Centers across four continents for ultimate performance and availability
- Customized and Tailored Made VPS Hosting to fit your needs
- Scalability: Allows you to quickly add load balancers, firewalls, private networks and apps such as: pfSense, Docker, CPanel, Drupal, Jenkins, WordPress, Magento, node.JS and many more.

- All SSDs with UNLIMITED TRAFFIC. 99.95% Up-Time Guaranteed
- Scale across hundreds of servers in seconds
- Billing options – Per Month or Per Day
- 24/7/365 Tech Human Support
- 30 Day Free Trial to test the services

More Information >>

2) DigitalOcean



DigitalOcean's droplet is a scalable computer service. It is more than just virtual machines. This cloud platform offers add-on storage, security, and monitoring capabilities to run production applications easily.

Features:

- Allows you to deploy your custom image, one-click app, or standard distribution
- You can deploy Droplets and get a reliable connection and flat pricing across 8 data center regions
- Option to select Standard Plans or Performance Plans according to your business needs

More Information >>

3) Rackspace



Rackspace is another useful cloud computer service tool. It offers services like hosting web applications, cloud files, cloud backup, database, and cloud server, etc.

Features:

- Fast-migrating to the Cloud
- Helps you to prepare your business for the worst-case scenario
- Work on pay as you go model, so you are charged base on your usage
- It helps you to use a combination of solid-state drives and hard drives to deliver high performance

[More Information >>](#)

4) MassiveGrid



[MassiveGrid](#) offers Virtual and Dedicated Private Clouds. With Virtual Private Clouds, helps users to have the flexibility to manage their resources in their environment according to their business needs.

Features:

- Offers fast & reliable Network Infrastructure
- Private cloud clients a secure web control panel, it can be used 24x7x365 to manage their clouds
- Offers high availability services with building a state of the art infrastructure
- Extremely Fast and dedicated Hardware

[**More Information >>**](#)

5) [Alibaba Cloud](#)



[Alibaba](#) is the largest Chinese cloud computing company. It is a new platform which created a global footprint with over 1500 CDN Nodes worldwide of 19 regions and 56 availability zones across more than 200 countries.

Features:

- Helps you to achieve faster results
- Helps you to protect and backup your data
- Full management permissions and multiple management methods
- Highly stable applications and reliable data storage

[More Information >>](#)

6) LiquidWeb



The liquid web offers cloud Sites which is a managed hosting platform which offers creatives freedom to build and launch websites without the need to learn cPanel or server management.

Features:

- It allows you to manage your sites quickly and effortlessly
- Host Unlimited Sites & Apps with a single Account
- Not require any server management skill
- The tool can easily be integrated with WordPress, Drupal, Joomla, etc.

[More Information >>](#)

7) Microsoft Azure



Azure is a cloud computing platform which is launched by Microsoft in February 2010. This open source and flexible cloud platform which helps in development, data storage, service management & hosting solutions.

Features:

- Windows Azure offers the most effective solution for your data needs
- Provides scalability, flexibility, and cost-effectiveness
- Offers consistency across clouds with familiar tools and resources
- Allow you to scale your IT resources up and down according to your business needs

Download link: <https://azure.microsoft.com/en-in/>

8) Google Cloud Platform



Google Cloud Platform

Google Cloud is a set of solution and products which includes GCP & G suite. It helps you to solve all kind of business challenges with ease.

Features:

- Allows you to scale with open, flexible technology
- Solve issues with accessible AI & data analytics
- Eliminate the need for installing costly servers
- Allows you to transform your business with a full suite of cloud-based services

Download link: <https://cloud.google.com/>

9) VMware



VMware is a comprehensive cloud management platform. It helps you to manage a hybrid environment running anything from traditional to container workloads. The tools also allow you to maximize the profits of your organization.

Features:

- Enterprise-ready Hybrid Cloud Management Platform
- Offers Private & Public Clouds
- Comprehensive reporting and analytics which improve the capacity of forecasting & planning
- Offers additional integrations with 3rd parties and custom applications, and tools.
- Provides flexible, Agile services

Download link: <https://www.vmware.com/in/cloud-services/infrastructure.html>

10) Salesforce



Salesforce cloud computing offers multiple cloud services like Sales Cloud, Service Cloud, Marketing Cloud, etc. Helps you to accelerate production of your environment.

Features:

- Salesforce Service Cloud offers 24 * 7 support
- Allows you to take a right and decisive decisions about your business
- Helps in managing the customer's contact information, automating the business processes, etc.

<https://www.salesforce.com/in/cloudcomputing/>

11) Oracle Cloud



Oracle Cloud offers innovative and integrated cloud services. It helps you to build, deploy, and manage workloads in the cloud or on premises. Oracle Cloud also helps companies to transform their business and reduce complexity.

Features:

- Oracle offers more options for where and how you make your journey to the cloud
- Oracle helps you realize the importance of modern technologies including Artificial intelligence, chatbots, machine learning, and more
- Offers Next-generation mission-critical data management in the cloud
- Oracle provides better visibility to unsanctioned apps and protects against sophisticated cyber attacks

Download link: <https://www.oracle.com/cloud/>

12) Verizon Cloud



Verizon Cloud computing platform allows you to control your infrastructure with advanced set-up and customization options from a single user interface.

Features:

- Expand any workload quickly to help grow your business with less risk
- Helps you to build the right cloud with performance, support, and flexibility to make your business successful
- Allows you to select flexible service need according to your organizations
- You can trim down the risk and retain the data integrity across the apps

Download link: <http://www.verizonenterprise.com/welcome-to-verizon-cloud/>

13) Navisite



NaviSite provides cloud services for enterprises and mid-sized businesses by using the best IT technologies.

It offers a range of cloud service solutions like Cloud Infrastructure services, Cloud desktop, and hosting services.

Features:

- NaviSite simplifies application management services which include Managed Office 365 services
- It offers cloud-based Infrastructure-as-a-Service (IaaS) solutions that include managed cloud and self-service cloud solutions
- It helps you to simplify desktop management and administration

Download link: <https://www.navisite.com/>

14) IBM Cloud



IBM cloud is a full stack cloud platform which spans public, private and hybrid environments. It is built with a robust suite of advanced and AI tools.

Features:

- IBM cloud offers infrastructure as a service (IaaS), software as a service (SaaS) and platform as a service (PaaS)
- IBM Cloud is used to build pioneering which helps you to gain value for your businesses
- It offers high performing cloud communications and services into your IT environment

Download link: <https://www.ibm.com/cloud/>

15) OpenNebula



OpenNebula is a cloud computing platform. It allows you to manage heterogeneously distributed data center infrastructures. It helps you to

manages the data center's virtual infrastructure to build private, public and hybrid implementations.

Features:

- Easy to install, use, maintain, and operate
- Provides greater functionality for private & hybrid clouds
- Highly-scalable, reliable, and commercially supported

Download link: <https://opennebula.org/>

16) Pivotal



Pivotal cloud foundry which is shortly known as PCF is a proven digital solution for businesses. It helps you to move faster toward a software-driven future.

Features:

- Accelerate feature delivery
- Specially designed tool for zero-downtime deployments
- Helps you to reduce risk in your app portfolio
- Deliver enterprise SLAs (Service level agreement) at scale

Download link: <https://pivotal.io/>

17) CloudSigma



Cloudsigma is a flexible cloud server, and virtual private server hosting solutions. It offers a straightforward and transparent approach for pricing. You can easily stream at multiple gigabit speed from their cloud servers.

Features:

- Helps you to achieve complete control and flexibility over your cloud environment
- Allows you to mix & match all SSD and magnetic storage
- This cloud computing service tool is certified as compliant with the highest ISO 27001 requirements for security and data privacy

Download link: <https://www.cloudsigma.com/>

18) Dell Cloud



Dell offers a cloud platform, cloud-enabled infrastructure, models, and serves in a single place. It allows your own or selects from reference architecture, integrated and public cloud platforms.

Features:

- Cloud that works with your existing operations
- Cloud consumption using Dell Financial Services
- Accelerate your transformation with expert cloud services

Download link: <https://www.dell.com/en-us/cloud/hybrid-cloud-computing/index.htm>

19) LimeStone



OnePortal Rapid is built with the latest open source technology to offer fast, feature rich, highly scalable cloud platform.

Features:

- Helps you to build and deploy applications with standard OpenStack based API libraries
- OpenStack Horizon web dashboard allows easy tracking and managing your cloud
- Flexible billing method ensures you only pay for the resources you use
- Scale quickly with additional compute and storage resource

Download link: <https://www.limestonenetworks.com/cloud/servers.html>

20) Quadrant:



If a cloud which was developed by Quadrant is fully scalable and reliable cloud infrastructure, the tool is billed hourly based on your line resource size which you can view a break down of the cost associated with each resource.

Features:

- QuadraNet's uptime SLA comes default with all cloud configurations
- The InfraCloud supports a wide variety of OS like CentOS to FreeBSD to Windows
- The custom interface helps you to manage your InfraCloud instances

Download link: <https://www.quadranet.com/infracloud>

• [Prev](#)

[Report a Bug](#)

AWS Tutorial

- [1\) What is AWS?](#)
- [2\) Creating an Amazon EC2 instance](#)
- [3\) Modifying the Amazon EC2 instance parameters](#)
- [4\) What is AWS Lambda?](#)
- [5\) AWS Certification Guide](#)
- [6\) Azure vs. AWS](#)
- [7\) Heroku Vs AWS](#)
- [8\) Digitalocean Vs AWS](#)
- [9\) Google Cloud Vs AWS](#)

- 10) AWS Competitors & Alternatives