

Cloud

Cloud Computing is an essential tool to learn to become a DevOps Engineer. Many modern days software applications are deployed on Cloud Platforms like [Amazon Web Service](#), [Microsoft Azure](#), [Google Cloud Platform](#), etc as it provides scalability of the resources, elasticity, automation, tools integrations, and cost-optimizations which helps to manage software applications in the cloud.

Introduction to Amazon Web Services

Amazon Web Services (AWS) is a leading top platform in providing the web services of various domains. AWS follows the trends of digital IT and comes up needy services with optimized performances covering a wide range of services from Compute to Storage. It covers a wider range of customers of different domains to expand their business operations. This Article covers the fundamentals of AWS and its scope of IT business.

What Is AWS And Why Is It Used?

AWS stands for [Amazon Web Services](#), It is an expanded **cloud computing platform** provided by [Amazon](#) Company. AWS provides a wide range of services with a **pay-as-per-use pricing model** over the Internet such as Storage, Computing power, [Databases](#), [Machine Learning](#) services, and much more. AWS facilitates for both businesses and individual users with effectively hosting the applications, storing the data securely, and making use of a wide variety of tools and services improving management flexibility for IT resources.

History Of AWS

Then providing [Simple Storage Service \(Amazon S3\)](#) revolutionized with scalable management of Storage. Coming up with effective compute and storage services and providing them rental basis helped many startup companies and users with the cost of manual Hardware Infrastructure setup. Introducing the concept of [serverless computing](#) with [AWS lambda](#) services enhanced its business globally. It came up with services like [Elastic Beanstalk](#) made the deployment of applications much easier bringing large audiences. AWS always came with diverse array of services offering with technical innovations, updated services with current trends. AWS has emerged as a powerhouse in the world of [Cloud Computing](#).

How AWS Works?

AWS comes up with its own network infrastructure on establishing the datacenters in different regions mostly all over the world. Its global Infrastructure acts as a backbone for operations and services provided by AWS. It facilitates the users on creating secure environments using [Amazon VPCs](#) (Virtual Private Clouds). Essential services like [Amazon EC2](#) and Amazon S3 for utilizing the compute and storage service with elastic scaling. It supports the dynamic scaling of the applications with the services such as [Auto Scaling](#) and [Elastic Load Balancing \(AWS ELB \)](#). It provides a good user-friendly AWS Management Console facilitating seamless configuration and management of AWS services to the Users. Its Architecture ensures high availability , fault tolerance making AWS as a versatile powerful Cloud Computing Platform.

AWS Fundamentals

In the Journey of AWS, understanding the key concepts such as Regions, Availability Zones, Global Network Infrastructure, etc is crucial. The fundamentals of AWS keep on maintaining the

applications reliable and scalable with services globally with coming to a strategic deployment of resources for optimal performance and resilience. The following are the some of the main fundamentals of AWS:

- **Regions:** AWS provide the services with respective division of regions. The regions are divided based on geographical areas/locations and will establish data centers. Based on need and traffic of users, the scale of data centers is depended to facilitate users with low-latencies of servcies.
- **Availability Zones (AZ):** To prevent the Data centers for the Natural Calamities or any other disasters. The Datacenters are established as sub sections with isolated locations to enhance fault tolerance and disaster recovery management.
- **Global Network Infrastructure:** AWS ensures the reliability and scalability of services through setting up its own [AWS Network Infrastructure](#) globally. It helps in better management of data transmissions for optimized performance and security reliance.

Top AWS Services

In the rapid revolution of Cloud Computing, AWS facilitates with wide variety of services respect to the fields and needs. The following are the top AWS services that are in wide usage:

- **[Amazon EC2\(Elastic Compute Cloud\)](#)** : It provides the Scalable computing power via cloud allowing the users to run applications and manage the workloads over their remotely.
- **[Amazon S3 \(Simple Storage Service \)](#)**: It offers scalable object Storage as a Service with high durability for storing and retrieving any amount of data.
- **[AWS Lambda](#)**: It is a service in Serverless Architecture with Function as a Service facilitating serverless computing i.e., running the code on response to the events, the background environment management of servers is handled by aws automatically. It helps the developers to completely focus on the logic of code build.
- **[Amazon RDS \(Relational Database Service\)](#)**: This is an aws service that simplifies the management of database providing high available relational databases in the cloud.
- **[Amazon VPC \(Virtual Private Cloud\)](#)**: It enables the users to create isolated networks with option of public and private expose within the AWS cloud, providing safe and adaptable configurations of their resources.

To know more about refer the Article – [Top 25 AWS Service List](#)

Advantages Of Amazon Web Services

- AWS allows you to easily scale your resources up or down as your needs change, helping you to save money and ensure that your application always has the resources it needs.
- AWS provides a highly reliable and secure infrastructure, with multiple data centers and a commitment to 99.99% availability for many of its services.

- AWS offers a wide range of services and tools that can be easily combined to build and deploy a variety of applications, making it highly flexible.
- AWS offers a pay-as-you-go pricing model, allowing you to only pay for the resources you actually use and avoid upfront costs and long-term commitments.

Disadvantages Of Amazon Web Services

- AWS can be complex, with a wide range of services and features that may be difficult to understand and use, especially for new users.
- AWS can be expensive, especially if you have a high-traffic application or need to run multiple services. Additionally, the cost of services can increase over time, so you need to regularly monitor your spending.
- While AWS provides many security features and tools, securing your resources on AWS can still be challenging, and you may need to implement additional security measures to meet your specific requirements.
- AWS manages many aspects of the infrastructure, which can limit your control over certain parts of your application and environment.

Applications Of AWS

The AWS services are using by both startup and MNC companies as per their usecase. The startup companies are using overcome hardware infrasture cost and applications deployments effectively with cost and performance. Whereas large scale companies are using AWS cloud services for the management of their Infrastructure to completely focus on the development of products widely. The following the Real-world industrial use-cases of AWS services:

- **Netflix:** The Large streaming gaint using AWS for the storage and scaing of the applications for ensuring seamless content delivery with low latency without interruptions to millions of users globally.
- **Airbnb:** By utilizing AWS, Airbnb manages the various workloads and provides insurable and expandable infrastructure for its virtual marketplace and lodging offerings.
- **NASA's Jet Propulsion Laboratory:** It takes the help of AWS services to handle and analyze large-scale volumes of data related to vital scientific research missions and space exploration.
- **Capital One:** A financial Company that is utilizing AWS for its security and compliance while delivering innovative banking services to its customers.

AWS Global Infrastructure

The AWS global infrastructure is massive and is divided into geographical regions. The geographical regions are then divided into separate availability zones. While selecting the geographical regions for AWS, three factors come into play

- Optimizing Latency
- Reducing cost

- Government regulations (Some services are not available for some regions)

Each region is divided into at least two availability zones that are physically isolated from each other, which provides business continuity for the infrastructure as in a distributed system. If one zone fails to function, the infrastructure in other availability zones remains operational. The largest region North Virginia (US-East), has six availability zones. These availability zones are connected by high-speed fiber-optic networking.

There are over 100 edge locations distributed all over the globe that are used for the CloudFront (content delivery network). [CloudFront](#) can cache frequently used content such as images and videos(live streaming videos also) at edge locations and distribute it to edge locations across the globe for high-speed delivery and low latency for end-users. It also protects from DDOS attacks.

AWS Management Console

The AWS management console is a web-based interface to access AWS. It requires an AWS account and also has a smartphone application for the same purpose. So When you sign in for first time, you see the console home page where you see all the services provided by AWS. Cost monitoring is also done through the console.

AWS resources can also be accessed through various Software Development Kits (SDKs), which allows the developers to create applications as AWS as its backend. There are SDKs for all the major languages(e.g., [JavaScript](#), [Python](#), [Node.js](#), [.Net](#), [PHP](#), [Ruby](#), [Go](#), [C++](#)). There are mobile SDKs for Android, iOS, React Native, Unity, and Xamarin. AWS can also be accessed by making [HTTP calls](#) using the AWS-API. AWS also provides a [AWS Command Line Interface \(CLI\)](#) for remotely accessing the AWS and can implement scripts to automate many processes. This Console is also available as an app for Android and iOS. For mobile apps, you can simply download AWS console app.

AWS Cloud Computing Models

There are three [cloud computing models](#) available on AWS.

1. **Infrastructure as a Service (IaaS):** It is the basic building block of cloud IT. It generally provides access to data storage space, networking features, and computer hardware(virtual or dedicated hardware). It is highly flexible and gives management controls over the IT resources to the developer. For example, [VPC](#), [EC2](#), [EBS](#).
2. **Platform as a Service (PaaS):** This is a type of service where AWS manages the underlying infrastructure (usually operating system and hardware). This helps the developer to be more efficient as they do not have to worry about undifferentiated heavy lifting required for running the applications such as capacity planning, software maintenance, resource procurement, patching, etc., and focus more on deployment and management of the applications. For example, [RDS](#), [EMR](#), [ElasticSearch](#).
3. **[Software as a Service\(SaaS\)](#):** It is a complete product that usually runs on a browser. It primarily refers to end-user applications. It is run and managed by the service provider. The end-user only has to worry about the application of the software suitable to its needs. For example, Salesforce.com, Web-based email, Office 365 .

Amazon Web Services – FAQs

What Is AWS Used For?

The purpose of AWS is to provide a variety of services including storage, compute power, databases, and machine learning helping companies and users to build, run, and deploy their applications effectively with optimized performance, and cost-effectiveness.

Is AWS Good For A Career And What's The Salary?

Yes, A career in AWS Cloud is a great choice. It is a very competitive high-demand one, the one who gets expertise will receive competitive earnings.

Can I Learn AWS For Free, And Is It Easy To Learn?

Yes. AWS offers free-tier accounts for learning and doing practices with some resources with good limit. It is quite great for the beginner to have great practical learnings with implementations.

Does AWS Require Coding Skills?

Even though not familiar with coding skills one can learn and Use Cloud Service. But knowing how to code, particularly in scripting languages helps you to be more productive while using AWS.

Are AWS Certifications Necessary For A Career In Cloud Computing?

Yes, These AWS certifications are quite helpful for both getting exposure and chances of landing a job. It helps improve your career prospects in the competitive field of the cloud.

Amazon Web Services – Setting Up an AWS Account

Amazon web services is a cloud service platform that provides on-demand computational services, databases, storage space, and many more services. AWS allows its user to choose products from its wide variety of services and use them on-demand with no upfront payment for most of the services. Individually an AWS service may lack some functionality but, given the right AWS architecture, AWS services can be easily integrated to make highly complex and robust applications.

In this article, we will look into the process of creating & setting up an **AWS Free Tier Account**. Amazon is providing a number of various services in this AWS Free Tier account with some restrictions so that users can gain practical experience and a deeper understanding of AWS Cloud services. The AWS Free Tier's main objective is to give users a year(12 months) of free access to AWS Cloud Services so they may get experience in how to use the services. There is a limit on how much we may utilize of each service included in the AWS Free Tier account before being charged.

Note: You're not charged for any [AWS services](#) that you sign up for unless you're exceeding the free Tier limit. (Turn off or Delete the services once you have done with your practice)

Set Up A AWS Free Tier Account

Step 1: First Open your web browser and search for AWS Login Console and click on the first link. As shown in the picture below

Step 2: An AWS Login Console page will open now click on **Create an AWS Account**.

Step 3: A new AWS sign-in page will now open after selecting **Create an AWS Account**. Choose to **Create a new AWS account**. As shown in the image below

Step 4: In order to use the feature to log into an **AWS Free Tier Account**, we must validate the email address and have to provide the AWS account name in this stage. After clicking on **“Verify Email Address,”** you will receive a verification code at the address you provided. Next, you must create a **password** for this account. Finally, click **“Continue”** to move on to the next stage. The pictures below show every step of the process.

After clicking on **“Verify Email Address,”** you will receive a verification code at the address you provided.

Next, you must create a **password** for this account. Finally, click **“Continue”** to move on to the next stage.

Step 5: We must include all of our contact information in this phase to make it easier for Amazon support personnel to get in touch with us about our **AWS Account** and any feature references. As shown in the image below.

Step 6: We must provide the credit/debit card information in this step. There is no reason to panic at this time. AWS won't deduct any amount unless you pay it on your own. AWS may temporarily keep your identification that they will charge you only 2 Indian rupees.

Step 7: We have to verify our phone number in this phase. As seen in the image below, select **“TEXT or Voice call”** as the method for receiving your verification number, then complete the captcha by clicking on **“Send SMS.”** You will be sent to a screen where you must confirm the verification code you have received and click continue to proceed to the following stage. As seen in the pictures below.

Step 8: Enter the verification code you received on your mobile device, validate it, and then click Continue to move on to the following stage.

Step 9: Choose the support strategy you want to use. We are setting up an AWS Free Tier Account so select the Basic Support option, which is cost-free and which AWS also suggests for new customers. The Basic Support Plan includes following

- 24*7 self-service access to AWS resources
- Can access personal health dashboard
- It is free of cost

After selecting a plan, click **“Complete the sign up”** as shown in the image.

Step 10: **“Congratulation”** Upon the creation of your AWS account, you can sign in by clicking Click Sign into the console once more, input the email address that you provided, your password, and then click Sign in as shown in the accompanying image, where you can see AWS Management Console's home page for certain of its offering services.

Enter your **email address and previously-configured password**.

And this is the **Amazon Console Home page**, where you may access some of the most popular AWS services, including [EC2](#), [VPC](#), [AUTOSCALING](#), etc.

After setting up our AWS Free Tier account, we are now ready to begin using the services that AWS offers.

Create EC2 Instance in AWS (Amazon): Complete Tutorial

EC2 stands for Elastic Compute Cloud. EC2 is an on-demand computing service on the AWS cloud platform. Under computing, it includes all the services a computing device can offer to you along with the flexibility of a virtual environment. It also allows the user to configure their instances as per their requirements i.e. allocate the RAM, ROM, and storage according to the need of the current task.

Amazon EC2 is a short form of Elastic Compute Cloud (ECC) it is a cloud computing service offered by the Cloud Service Provider AWS. You can deploy your applications in EC2 servers without worrying about the underlying infrastructure. You configure the EC2-Instance in a very secure manner by using the VPC, Subnets, and Security groups. You can scale the configuration of the EC2 instance you have configured based on the demand of the application by attaching the autoscaling group to the EC2 instance. You can scale up and scale down the instance based on the incoming traffic of the application.

Create EC2 Instance in AWS (Amazon)

The following are the steps for creating an EC2 instance in AWS (Amazon):

Step 1: First, log into your [AWS account](#) and click on “**services**” present on the left of the AWS management console, i.e. the primary screen. From the drop-down menu of options, tap on “**EC2**”. To create an AWS free tier account refer to [Amazon Web Services \(AWS\) – Free Tier Account Set up](#).

Under Resources >> Click on “Instances running” — It will show if any [EC2 instances](#) are running or not.

Step 2: Click on the launch instance click on the launch instance, after clicking on it you will be redirected to a launch page where we can create an instance. Configure all the requirements to Create a new instance like the name of the instance as shown in the figure below.

Step 3: Select AMI – Required operating system from the available. There are different types of OS available select the OS as per your requirement.

Step 4: By default, it selects a free tier of storage. (IF YOU ARE ELIGIBLE FOR THE FREE TIER). From the available storage specifications, select a free tier-eligible storage service. The instance type includes the no.of CPUs required and the Memory required for your application. By default, the instance type is “t2.micro” which is a free tier-eligible service. Do not select any other which leads to the billing amount. To know more about instance types refer to [Amazon EC2 – Instance Types](#).

Step 4: Now, create a key-value pair, by clicking on “Create new key pair”. A window will pop up for creating key pair as shown below. The key value pair plays a major role while connecting to the EC2-Instance it will act as an [SSH-Key](#) to connect to the instance. Create Key-PairEnter

name>>Select “.pem” and create. Automatically key pair which was created will be downloaded. Select the created key pair.

Step 5: Keep the network settings as default settings and make changes if required. Storage As mentioned in the picture, Free tier eligible can get up to 30 GB of [EBS Storage](#). Keep it as default.

Step 6: Launching Instance At last, Check if all the selected are eligible for a free tier or not and click on “Launch instance”. That’s it, an instance will be created.

Steps To Connect Terminal Using SSH-Key

Step 1: Select the server to which you want to connect and click on the connect button at the top of that instance as shown in the image below.

Step 2: Copy the SSH key which is right following the example it will act as a [key-pair](#) to connect to EC2-Instance.

Step 3: Open the terminal and go to the folder where your .pem file is located and paste the key that you have copied in AWS and paste it in the [terminal](#).

To know whether you connected to EC2-Instance perfectly or not you can check the [IP-Adrees](#) of the instance if the IP is displaying then you have connected successfully.

How do I create AWS Resources Using Ansible?

[Ansible](#) provides set of built-in modules which are used to configure the AWS resources. BU using this ansible modules you can configure AWS resources like EC2 instance, S3 buckets, load balancer and autoscaling and son on.

Follow the steps mentioned below to create EC2 instance using Ansible

Step 1: Install the required modules in ansible to configure the AWS resources. You can install by using the following command.

```
ansible-galaxy install
```

Step 2: Create an playbook by using the yaml file and mention the configurations required to create AWS ec2 instance following is the sample YAML file to create AWS EC2 instance.

```
---
- hosts: localhost
  tasks:
    - name: Create an EC2 instance
      ec2:
        instance_type: t2.micro
```


ami: ami-0b7927eb9e3372e28
state: present
tags:
Name: MyEC2Instance

AWS EC2 Instance Types

Different Amazon EC2 instance types are designed for certain activities. Consider the unique requirements of your workloads and applications when choosing an instance type. This might include needs for computing, memory, or storage. To know more about EC2 instance types refer to the [Amazon EC2 instance types](#).

Creating AWS EC2 Snapshot

EBS Snapshots are point-in-time images or copies of your EBS Volume. These are stored on S3, which can be accessed through Elastic Cloud Computing APIs or AWS Console. While EBS volumes are availability zone (AZ's) specific but, Snapshots are Region-specific. Your Snapshot size must be either same or larger than the size of the original volume from which the snapshot is taken. As per Amazon, each AWS account can have a maximum of up to 5000 images or copies Volumes and up to 10,000 EBS Snapshots created. A snapshot, when created, shows a 'pending' status, which then converts into 'complete' once the snapshot creation is successful. To know more about AWS EC2 snapshot refer to the [How to create AWS snapshot](#).

Creating AWS EC2 AMI

An Amazon Machine Image(AMI) which contains the information to launch the Amazon EC2 instance. This AMI includes all the things which are required for the application like operating system, software and settings to create your own customized EMI. Creating your own customized AMI will make it easier to deploy the application in the EC2 instance. To know more about how to create AWS AMI refer to the [Amazon Web Services – Creating an Amazon Machine Image\(AMI\)](#).

EC2 Instance All-State in AWS

The common EC2 instance states are Pending, Running, Stopping, Stopped, Terminated, Shutting Down, and Rebooted. It is important to keep track of the state of your EC2 instances so that you can manage them properly. You can view the state of your instances in the EC2 Console, AWS CLI, or AWS SDKs. In AWS, EC2 (Elastic Compute Cloud) instances can have different states, which indicate what operations can be performed on them. Here are some of the common EC2 instance states:

1. **Pending:** When you launch an EC2 instance, it enters the pending state. This means that AWS is in the process of creating the instance and initializing all of the necessary components, such as the [virtual machine](#) and the associated [networking resources](#). During this time, you won't be able to access the instance, as it is not yet ready to be used.
2. **Running:** Once an EC2 instance has finished initializing, it enters the running state. This means that the instance is up and running and is ready to be used. In this state, you can log in to the instance and start using it to run your applications and services.

3. **Stopping:** If you manually stop an EC2 instance, or if it is part of an [auto-scaling group](#) and is being terminated, it enters the stopping state. During this state, AWS prepares the instance for shutdown by stopping any processes or applications running on the instance and disconnecting it from the network. However, the instance's configuration and data are preserved, so you can start the instance again later if you need to.
4. **Stopped:** Once an EC2 instance has been stopped, it enters the stopped state. In this state, the instance is not running and is not available for use. However, the instance's configuration and data are preserved, so you can start the instance again later if you need to. You might stop an instance if you don't need it for a period of time but don't want to terminate it entirely.
5. **Terminated:** If you manually terminate an EC2 instance, or if it is part of an auto-scaling group and is being terminated, it enters the terminated state. In this state, the instance is permanently deleted, and all of its configuration and data are lost. You might terminate an instance if you no longer need it, or if you want to replace it with a new instance.
6. **Shutting-down:** If AWS is retiring an instance, it goes into the "Shutting-down" state for a brief period before the instance is terminated. During this time, the instance is no longer available for use, and the data and configuration are preserved. This state is similar to the stopping state but with an added step of preparing the instance for retirement.
7. **Rebooting:** If you choose to reboot an EC2 instance, it enters the rebooting state. During this state, the instance's operating system is shut down and then restarted, but the instance's configuration and data are preserved. You might reboot an instance if you need to apply updates or make changes to the instance's configuration.

You can view the state of your EC2 instances in the EC2 Console, AWS CLI, or AWS SDKs. It is important to keep track of the state of your instances so that you can manage them properly, such as starting, stopping, or terminating instances as needed. When you use Amazon Web Services (AWS) to run virtual servers or instances, these instances can be in different states depending on what's happening with them. For example, an instance might be "running" when it's up and running properly, or "stopped" when it's not currently being used. [AWS CloudWatch](#) to monitor your EC2 instances and their associated resources in real time. CloudWatch provides a wealth of data on your instances, including CPU usage, disk activity, and network traffic, which can help you identify performance issues and other problems before they have a chance to impact your users.

Advantages of AWS EC2-Instances

- EC2 instances can be easily scaled up or down as per the requirement, providing a highly scalable and flexible infrastructure.
- EC2 instances are charged based on usage, making it cost-effective as you only pay for what you use.
- It can be easily deployed and managed using Amazon Web Services (AWS) management console, APIs, or CLI.

- It can be deployed in multiple availability zones to ensure high availability and data durability.
- It can be customized with different operating systems, applications, and network configurations.

Disadvantages of AWS EC2-Instances

- EC2 instances have limited customization options, which may not be sufficient for some applications.
- It can be expensive, especially when scaling up, and it can be challenging to control costs.
- This is vulnerable to security risks, such as unauthorized access, data breaches, and cyberattacks.
- EC2 instances can be complex to set up and manage, especially for non-technical users.
- It may experience latency due to the location of the instances and the data center, which can affect application performance.

Use cases of AWS EC2- Instances

- EC2 instances can be used to host websites, applications, and APIs in the cloud.
- It can be used to process large amounts of data using tools like Apache Hadoop and Apache Spark.
- It can be used to perform demanding computing tasks, such as scientific simulations and financial modeling.
- EC2 instances can be used to develop, test, and deploy software, allowing teams to quickly spin up resources as needed.

Best Practices For Amazon EC2 Instances

The best practices for launching or choosing the Amazon EC2-Instance are mentioned below:

1. **Choosing the Right OS:** While launching the EC2-instance you need to select the OS that suits your requirements which is where you want to deploy your application in which OS like Windows, Linux, and MacO.
2. **Cost saving:** When you are going to launch an instance you need to consider the cost and try to reduce the cost for the organization choose the type of instance wisely depending on the requirement.
3. **Attach Autoscaling:** When are you creating the EC2-Instance make sure that you attach it to the Autoscaling group by this if there is sudden traffic EC2-Instance will scale automatically depending on the load.

4. **Secure Instance:** Secure the EC2-instance by configuring it in the VPC and managing the inbound and outbound rules and also managing the incoming traffic to the EC2-Instance with the help of traffic routing.
5. **Snapshots:** Take automatic snapshots of the EBS volume automatically for certain intervals of time so you can back up the data without losing it.
6. **Attach EBS:** Attach the EBS volume to EC2-Instance without depending on the root volume of the server if anything happened to the root then your data will be safe in the EBS volume.
7. **Attach Elastic LoadBalancer:** Attach the [Elastic load balancer](#) to the EC2-Instances when there is sudden traffic the traffic will be distributed across multiple instances which will decrease the load.

Conclusion On Creating EC2 Instance in AWS (Amazon)

Another important aspect of managing EC2 instances is understanding the various instance types available in AWS. Different instance types have different performance characteristics and are optimized for different types of workloads. For example, some instances are optimized for CPU-intensive workloads, while others are better suited for memory-intensive applications. By choosing the right instance type for your workload, you can ensure that your applications are running efficiently and cost-effectively. Overall, understanding the different states of EC2 instances in AWS is just one aspect of effectively managing your infrastructure in the cloud. By taking advantage of tools like AWS CloudWatch, choosing the right instance types for your workloads, and following best practices for security and maintenance, you can ensure that your applications and services are always available to your users and that you're getting the most out of your investment in the cloud.

FAQs on AWS EC2 Create EC2 Instance in AWS (Amazon)

1. What Is The Difference Between S3 And EC2?

[S3](#) and EC2 are the two different services offered by the Amazon Web Services one is to store the data in the form of objects and another is to deploy the Web Application.

2. Is EC2 a PAAS or IaaS?

EC2 belongs to Infrastructure as a service where the underlying infrastructure is taken care by AWS.

3. Is EC2 a Virtual Machine?

Yes EC2 is a virtual machine where you can deploy the web applications.

Elastic Load Balancer in AWS

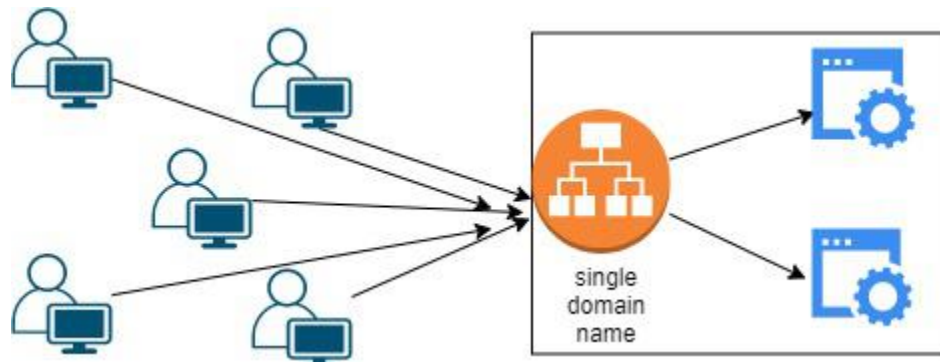
In simplest terms, **cloud computing** means storing and accessing the data and programs on remote servers that are hosted on the internet instead of the computer's hard drive or local server. It is also referred to as Internet-based computing.

What are Amazon Web Services?

Amazon Web Services is a subsidiary of Amazon.com that provides on-demand cloud computing platforms for individuals, companies, and governments, on a paid subscription basis, pay-as-you-go principle. [Amazon Web Services](#) offers a highly reliable, scalable, low-cost infrastructure platform in the cloud. You can automate the infrastructure with the tool called Terraform to know more about Terraform refer to [What is Terraform?](#)

How Elastic Load Balancing?

The elastic load balancer is a service provided by Amazon in which the incoming traffic is efficiently and automatically distributed across a group of backend servers in a manner that increases speed and performance. It helps to improve the scalability of your application and secures your applications. Load Balancer allows you to configure health checks for the registered targets. In case any of the registered targets ([Autoscaling group](#)) fails the health check, the load balancer will not route traffic to that unhealthy target. Thereby ensuring your application is highly available and fault tolerant. To know more about load balancing refer to [Load Balancing in Cloud Computing](#).



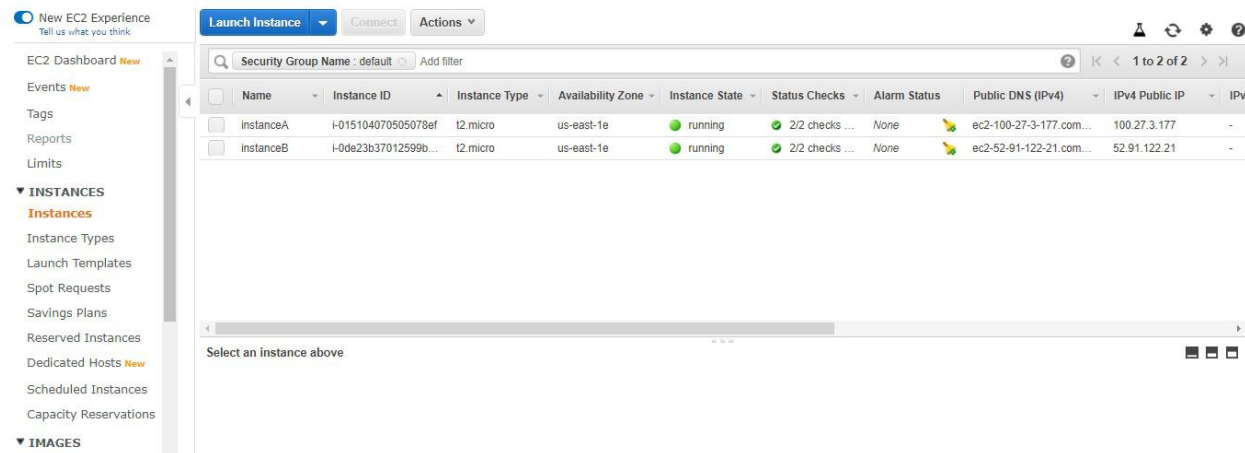
Types of Load Balancers

1. **Classic Load Balancer:** It is the traditional form of load balancer which was used initially. It distributes the traffic among the instances and is not intelligent enough to support host-based routing or path-based routing. It ends up reducing efficiency and performance in certain situations. It is operated on the connection level as well as the request level. Classic Load Balancer is in between the transport layer (TCP/SSL) and the application layer ([HTTP/HTTPS](#)).
2. **Application Load Balancer:** This type of Load Balancer is used when decisions are to be made related to HTTP and HTTPS traffic routing. It supports path-based routing and host-based routing. This load balancer works at the Application layer of the OSI Model. The load balancer also supports dynamic host port mapping.

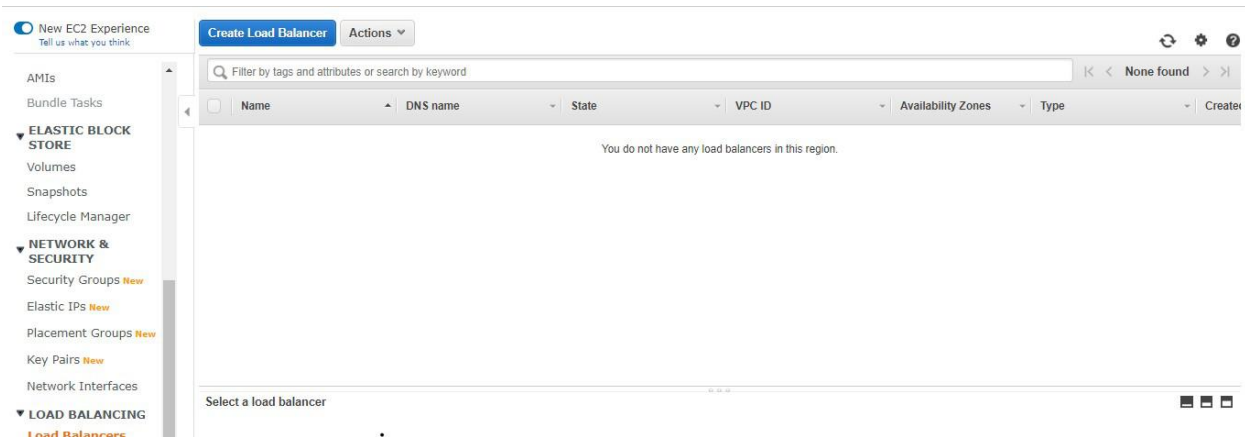
3. **Network Load Balancer:** This type of load balancer works at the transport layer(TCP/SSL) of the OSI model. It's capable of handling millions of requests per second. It is mainly used for load-balancing TCP traffic.
4. **Gateway Load Balancer:** Gateway Load Balancers provide you the facility to deploy, scale, and manage virtual appliances like firewalls. [Gateway Load Balancers](#) combine a transparent network gateway and then distribute the traffic.

Steps to configure an Application load balancer in AWS

Step 1: Launch the two instances on the AWS management console named Instance A and Instance B. Go to services and select the load balancer. To create AWS free tier account refer to [Amazon Web Services \(AWS\) – Free Tier Account Set up.](#)



Step 2: Click on Create the load balancer.




Step 3: Select Application Load Balancer and click on Create.

Select load balancer type

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

Application Load Balancer




[Create](#)

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

[Learn more >](#)

Network Load Balancer



[Create](#)

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

[Learn more >](#)

Classic Load Balancer

PREVIOUS GENERATION
for HTTP, HTTPS, and TCP

[Create](#)

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

[Learn more >](#)

Step 4: Here you are required to configure the load balancer. Write the name of the load balancer. Choose the scheme as internet facing.

1. Configure Load Balancer

2. Configure Security Settings

3. Configure Security Groups

4. Configure Routing

5. Register Targets

6. Review

Step 1: Configure Load Balancer

Basic Configuration

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

Name ⓘ

my-loadbalancer

Scheme ⓘ

☒ internet-facing
☐ internal

IP address type ⓘ

ipv4

Listeners

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

Load Balancer Protocol	Load Balancer Port
HTTP	80

[Add listener](#)

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

Step 5: Add at least 2 availability zones. Select us-east-1a and us-east-1b

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

Step 1: Configure Load Balancer

Availability Zones

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

VPC ⓘ vpc-c63330bc (172.31.0.0/16) (default) ▼

Availability Zones

☒ us-east-1a subnet-07a25158 ▼
IPv4 address ⓘ Assigned by AWS

☒ us-east-1b subnet-f6629590 ▼
IPv4 address ⓘ Assigned by AWS

☐ us-east-1c subnet-e8d22fc9 ▼

☐ us-east-1d subnet-919214dc ▼

☐ us-east-1e subnet-28f6cc16 ▼

☐ us-east-1f subnet-54e1495a ▼

Cancel Next: Configure Security Settings

Step 6: We don't need to do anything here. Click on Next: Configure Security Groups

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

Step 2: Configure Security Settings



Improve your load balancer's security. Your load balancer is not using any secure listener.

If your traffic to the load balancer needs to be secure, use the HTTPS protocol for your front-end connection. You can go back to the first step to add/configure secure listeners under [Basic Configuration](#) section. You can also continue with current settings.

Cancel Previous Next: Configure Security Groups

Step 7: Select the default security group. Click on Next: Configure Routing

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

Step 3: Configure Security Groups

A security group is a set of firewall rules that control the traffic to your load balancer. On this page, you can add rules to allow specific traffic to reach your load balancer. First, decide whether to create a new security group or select an existing one.

Assign a security group: ☐ Create a new security group ☒ Select an existing security group

Filter: VPC security groups

Security Group ID	Name	Description	Actions
<input type="checkbox"/> sg-0bb0a9bc3e885adfb	AutoScaling-Security-Group-1	AutoScaling-Security-Group-1 (2020-06-15 12:00:39.275+05:30)	Copy to new
<input type="checkbox"/> sg-0b3772fb578fb44ce	AutoScaling-Security-Group-2	AutoScaling-Security-Group-2 (2020-06-15 15:18:53.000+05:30)	Copy to new
<input checked="" type="checkbox"/> sg-103a4f3e	default	default VPC security group	Copy to new
<input type="checkbox"/> sg-0b13f451747da2fc2	launch-wizard-1	launch-wizard-1 created 2020-05-12T23:27:45.924+05:30	Copy to new
<input type="checkbox"/> sg-0458b504a37badf44	launch-wizard-10	launch-wizard-10 created 2020-06-13T14:16:46.319+05:30	Copy to new
<input type="checkbox"/> sg-0fd12e18e2b9c22d6	launch-wizard-11	launch-wizard-11 created 2020-06-15T11:38:34.722+05:30	Copy to new
<input type="checkbox"/> sg-04b735293b2ccb9a7	launch-wizard-12	launch-wizard-12 created 2020-06-15T15:10:02.695+05:30	Copy to new
<input type="checkbox"/> sg-0f3b470cd95160c71	launch-wizard-13	launch-wizard-13 created 2020-06-15T20:33:05.606+05:30	Copy to new
<input type="checkbox"/> sg-0d9a46000ea95453f	launch-wizard-2	launch-wizard-2 created 2020-05-13T05:34:24.807+05:30	Copy to new

Cancel
Previous
Next: Configure Routing

Step 8: Choose the name of the target group to be my target group. Click on Next: Register Targets.

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

Step 4: Configure Routing

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

Target group

Target group

Name

Target type ☒ Instance ☐ IP ☐ Lambda function

Protocol

Port

Health checks

Protocol

Path

Cancel
Previous
Next: Register Targets

Step 9: Choose instance A and instance B and click on Add to register. Click on Next: Review.

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

Step 5: Register Targets

Registered targets

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

Remove

<input type="checkbox"/>	Instance	Name	Port	State	Security groups	Zone
<input type="checkbox"/>	i-015104070505078ef	instanceA	80	● running	default	us-east-1e
<input type="checkbox"/>	i-0de23b37012599b85	instanceB	80	● running	default	us-east-1e

Instances

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

Add to registered on port 80

Search Instances							
<input type="checkbox"/>	Instance	Name	State	Security groups	Zone	Subnet ID	Subnet CIDR
<input checked="" type="checkbox"/>	i-015104070505078ef	instanceA	running	default	us-east-1e	subnet-28f6cc16	172.31.48.0/20
<input checked="" type="checkbox"/>	i-0de23b37012599b85	instanceB	running	default	us-east-1e	subnet-28f6cc16	172.31.48.0/20

Cancel Previous Next: Review

Step 10: Review all the configurations and click on create

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

Step 6: Review

Please review the load balancer details before continuing

▼ Load balancer

Name

my-loadbalancer

Scheme

internet-facing

Listeners

Port:80 - Protocol:HTTP

IP address type

ipv4

VPC

vpc-c63330bc

Subnets

subnet-07a25158, subnet-f6629590

Tags

▼ Security groups

Security groups

sg-103a4f3e

▼ Routing

Target group

New target group

Target group name

my-target-group

Port

80

Target type

instance

Protocol

HTTP

Health check protocol

HTTP

Path

/

Health check port

traffic port

Health threshold

5

Cancel Previous Create

Step 11: Congratulations!! You have successfully created a load balancer. Click on close.

Load Balancer Creation Status

✓

Successfully created load balancer
Load balancer `my-loadbalancer` was successfully created.
Note: It might take a few minutes for your load balancer to be fully set up and ready to route traffic, and for the targets to complete the registration process and pass the initial health checks.
Suggested next steps

- Discover other services that you can integrate with your load balancer. Visit the [Integrated services](#) tab within `my-loadbalancer`
- Consider using AWS Global Accelerator to further improve the availability and performance of your applications. [AWS Global Accelerator console](#)

Close

Step 12: This highlighted part is the [DNS name](#) which when copied in the URL will host the application and will distribute the incoming traffic efficiently between the two instances.

The screenshot shows the AWS Management Console interface for the 'my-loadbalancer' Elastic Load Balancing (ELB) instance. The 'Basic Configuration' tab is selected, displaying the following details:

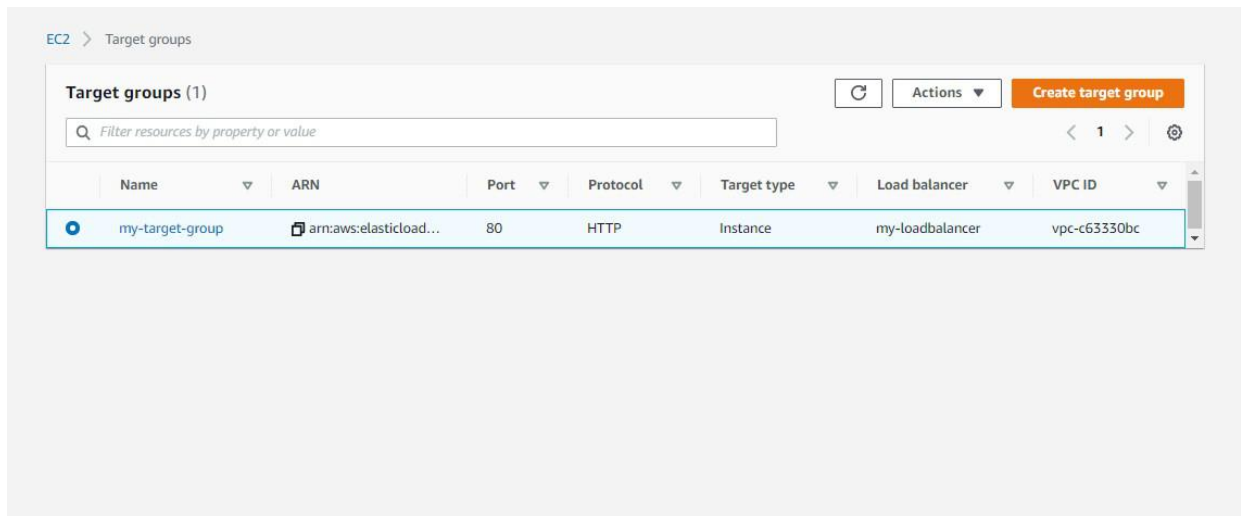
- Name:** my-loadbalancer
- ARN:** arn:aws:elasticloadbalancing:us-east-1:013179561180:loadbalancer/app/my-loadbalancer/ce4b8ceb87eb7694
- DNS name:** my-loadbalancer-2036300516 us-east-1.elb.amazonaws.com (A Record) - This text is highlighted in blue.
- State:** active
- Type:** application
- Scheme:** internet-facing
- IP address type:** ipv4
- VPC:** vpc-c63330bc
- Availability Zones:** subnet-07a25158 - us-east-1a

Step 13: This is the listener port 80 which listens to all the incoming requests

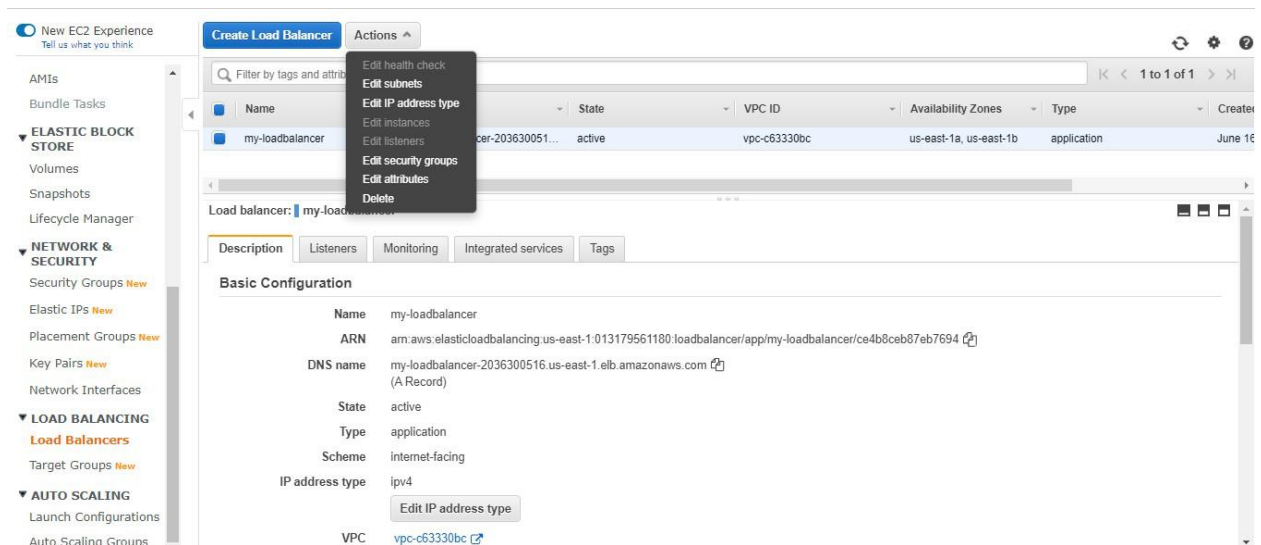
The screenshot shows the 'Listeners' tab for the 'my-loadbalancer' ELB instance. It displays a single listener configuration:

Listener ID	Security policy	SSL Certificate	Rules
arn:aws:elasticloadbalancing:us-east-1:013179561180:listener/app/my-loadbalancer/ce4b8ceb87eb7694/80	N/A	N/A	Default: forwarding to my-target-group View/edit rules

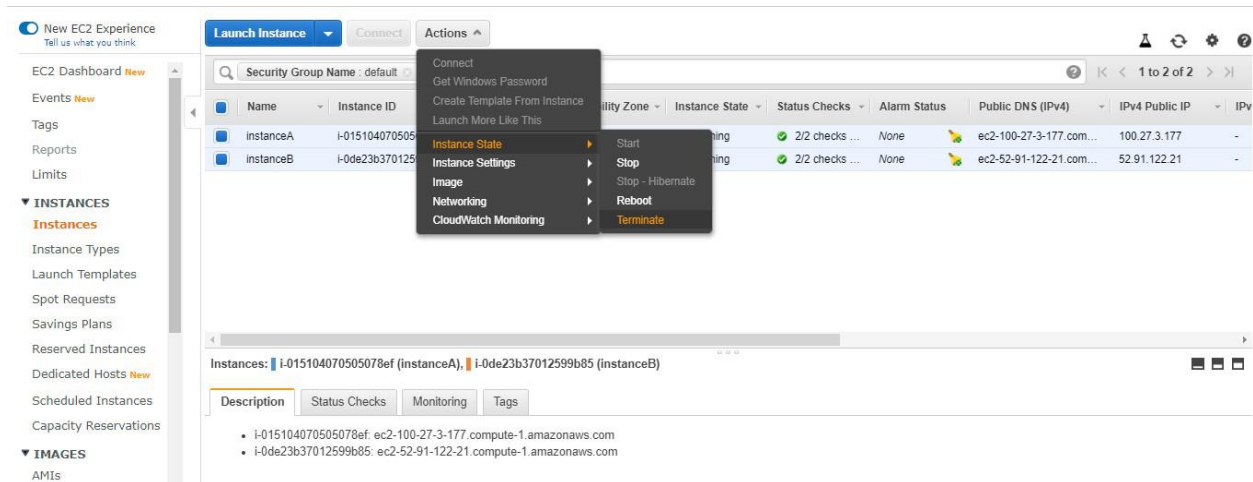
Step 14: This is the target group that we have created



Step 15: Now we need to delete the instance. Go to Actions -> Click on Delete.



Step 16: Also don't forget to terminate the instances.



Features of cloud

- No up-front investment
- Lowering operating cost
- Highly scalable and efficient
- Easy access
- Reducing business risks and maintenance expenses

Advantages of Elastic Load Balancer

- ELB automatically distributes incoming application traffic across multiple targets, such as [EC2 instances](#), [containers](#), and [IP addresses](#), to achieve high availability.
- It can automatically scale to handle changes in traffic demand, allowing you to maintain consistent application performance.
- It can monitor the health of its registered targets and route traffic only to the healthy targets.
- It evenly distributes traffic across all availability zones in a region, improving fault tolerance.

Disadvantages of Elastic Load Balancer

- ELB can add latency to your application, as traffic must pass through the load balancer before being routed to your targets.
- It has limited customization options, so you may need to use additional tools and services to fully meet your application's requirements.
- It can introduce additional complexity to your application architecture, requiring you to manage and maintain additional resources.
- It can increase your overall AWS costs, especially if you have high traffic volumes or require multiple load balancers.

FAQs On AWS Load Balancer

Q.1: What is the difference between EC2 and ELB?

Elastic load balancing will distribute the traffic to the application which is present in EC2-Instance.

Q.2: Which is faster NLB or ALB?

Network load balancer is faster than the Application load balancer when there is an sudden load.

Elastic Load Balancer in AWS

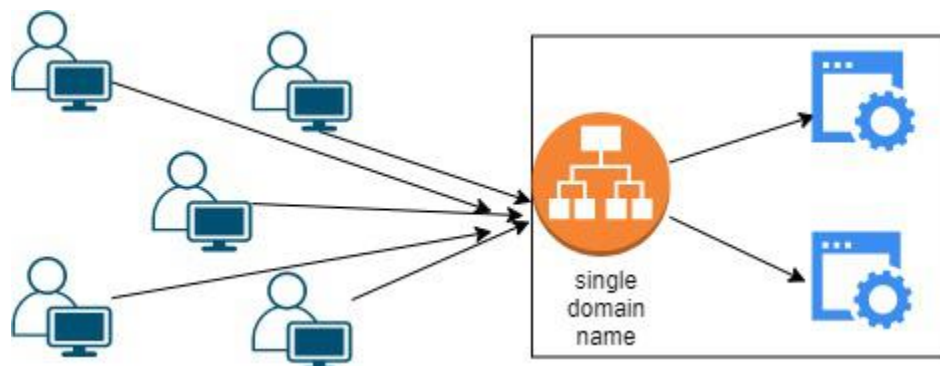
In simplest terms, **cloud computing** means storing and accessing the data and programs on remote servers that are hosted on the internet instead of the computer's hard drive or local server. It is also referred to as Internet-based computing.

What are Amazon Web Services?

Amazon Web Services is a subsidiary of Amazon.com that provides on-demand cloud computing platforms for individuals, companies, and governments, on a paid subscription basis, pay-as-you-go principle. [Amazon Web Services](#) offers a highly reliable, scalable, low-cost infrastructure platform in the cloud. You can automate the infrastructure with the tool called Terraform to know more about Terraform refer to [What is Terraform?](#)

How Elastic Load Balancing?

The elastic load balancer is a service provided by Amazon in which the incoming traffic is efficiently and automatically distributed across a group of backend servers in a manner that increases speed and performance. It helps to improve the scalability of your application and secures your applications. Load Balancer allows you to configure health checks for the registered targets. In case any of the registered targets ([Autoscaling group](#)) fails the health check, the load balancer will not route traffic to that unhealthy target. Thereby ensuring your application is highly available and fault tolerant. To know more about load balancing refer to [Load Balancing in Cloud Computing](#).



Types of Load Balancers

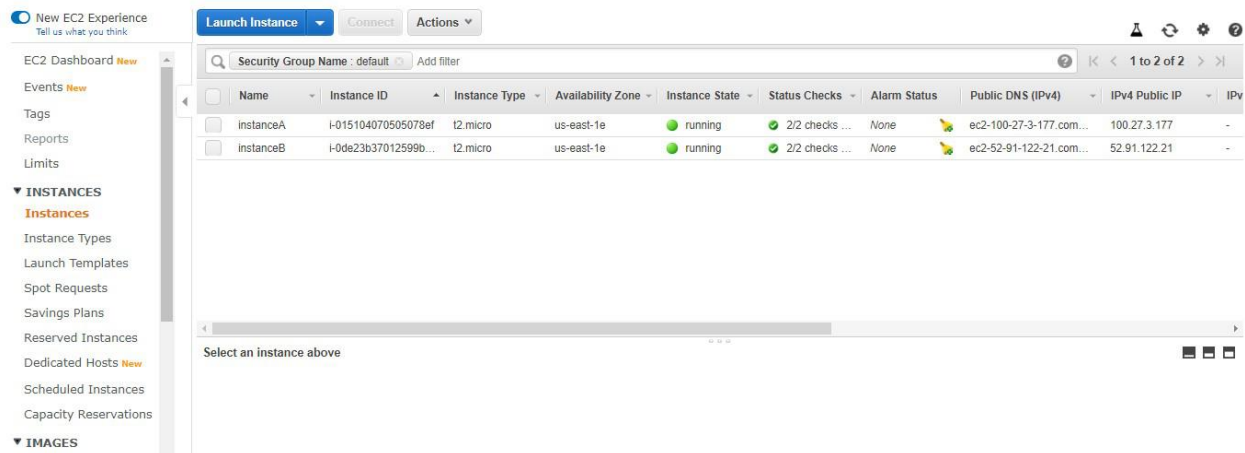
1. **Classic Load Balancer:** It is the traditional form of load balancer which was used initially. It distributes the traffic among the instances and is not intelligent enough to

support host-based routing or path-based routing. It ends up reducing efficiency and performance in certain situations. It is operated on the connection level as well as the request level. Classic Load Balancer is in between the transport layer (TCP/SSL) and the application layer ([HTTP/HTTPS](#)).

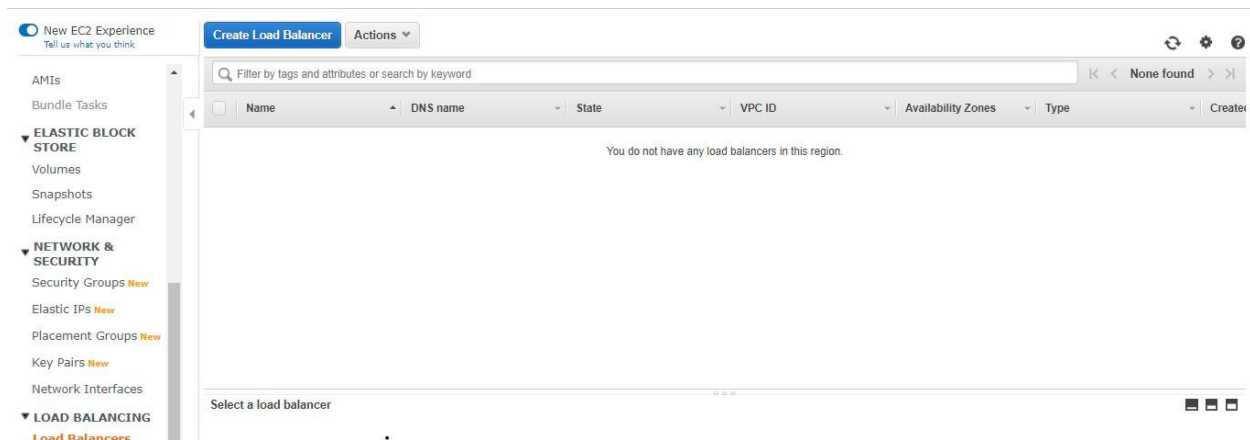
2. **Application Load Balancer:** This type of Load Balancer is used when decisions are to be made related to HTTP and HTTPS traffic routing. It supports path-based routing and host-based routing. This load balancer works at the Application layer of the OSI Model. The load balancer also supports dynamic host port mapping.
3. **Network Load Balancer:** This type of load balancer works at the transport layer(TCP/SSL) of the OSI model. It's capable of handling millions of requests per second. It is mainly used for load-balancing TCP traffic.
4. **Gateway Load Balancer:** Gateway Load Balancers provide you the facility to deploy, scale, and manage virtual appliances like firewalls. [Gateway Load Balancers](#) combine a transparent network gateway and then distribute the traffic.

Steps to configure an Application load balancer in AWS

Step 1: Launch the two instances on the AWS management console named Instance A and Instance B. Go to services and select the load balancer. To create AWS free tier account refer to [Amazon Web Services \(AWS\) – Free Tier Account Set up](#).



Step 2: Click on Create the load balancer.



Step 3: Select Application Load Balancer and click on Create.

Select load balancer type

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

Application Load Balancer

Create

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

[Learn more >](#)

Network Load Balancer

Create

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

[Learn more >](#)

Classic Load Balancer

PREVIOUS GENERATION
for HTTP, HTTPS, and TCP

Create

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

[Learn more >](#)

Step 4: Here you are required to configure the load balancer. Write the name of the load balancer. Choose the scheme as internet facing.

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

Step 1: Configure Load Balancer

Basic Configuration

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

Name ⓘ

Scheme ⓘ ☒ Internet-facing
☐ Internal

IP address type ⓘ

Listeners

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

Load Balancer Protocol	Load Balancer Port
HTTP	80

Add listener

Cancel
Next: Configure Security Settings

Step 5: Add at least 2 availability zones. Select us-east-1a and us-east-1b

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

Step 1: Configure Load Balancer

Availability Zones

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

VPC ⓘ vpc-c63330bc (172.31.0.0/16) (default) ▼

Availability Zones

☒ us-east-1a subnet-07a25158 ▼
IPv4 address ⓘ Assigned by AWS

☒ us-east-1b subnet-f6629590 ▼
IPv4 address ⓘ Assigned by AWS

☐ us-east-1c subnet-e8d22fc9 ▼

☐ us-east-1d subnet-919214dc ▼

☐ us-east-1e subnet-28f5cc16 ▼

☐ us-east-1f subnet-54e1495a ▼

Cancel Next: Configure Security Settings

Step 6: We don't need to do anything here. Click on Next: Configure Security Groups

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

Step 2: Configure Security Settings



Improve your load balancer's security. Your load balancer is not using any secure listener.

If your traffic to the load balancer needs to be secure, use the HTTPS protocol for your front-end connection. You can go back to the first step to add/configure secure listeners under [Basic Configuration](#) section. You can also continue with current settings.

Cancel Previous Next: Configure Security Groups

Step 7: Select the default security group. Click on Next: Configure Routing

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

Step 3: Configure Security Groups

A security group is a set of firewall rules that control the traffic to your load balancer. On this page, you can add rules to allow specific traffic to reach your load balancer. First, decide whether to create a new security group or select an existing one.

Assign a security group: ☐ Create a new security group ☒ Select an existing security group

Filter: VPC security groups

Security Group ID	Name	Description	Actions
<input type="checkbox"/> sg-0bb0a9bc3e885adfb	AutoScaling-Security-Group-1	AutoScaling-Security-Group-1 (2020-06-15 12:00:39.275+05:30)	Copy to new
<input type="checkbox"/> sg-0b3772fb578fb44ce	AutoScaling-Security-Group-2	AutoScaling-Security-Group-2 (2020-06-15 15:18:53.000+05:30)	Copy to new
<input checked="" type="checkbox"/> sg-103a4f3e	default	default VPC security group	Copy to new
<input type="checkbox"/> sg-0b13f451747da2fc2	launch-wizard-1	launch-wizard-1 created 2020-05-12T23:27:45.924+05:30	Copy to new
<input type="checkbox"/> sg-0458b504a37badf44	launch-wizard-10	launch-wizard-10 created 2020-06-13T14:16:46.319+05:30	Copy to new
<input type="checkbox"/> sg-0fd12e18e2b9c22d6	launch-wizard-11	launch-wizard-11 created 2020-06-15T11:38:34.722+05:30	Copy to new
<input type="checkbox"/> sg-04b735293b2ccb9a7	launch-wizard-12	launch-wizard-12 created 2020-06-15T15:10:02.695+05:30	Copy to new
<input type="checkbox"/> sg-0f3b470cd95160c71	launch-wizard-13	launch-wizard-13 created 2020-06-15T20:33:05.606+05:30	Copy to new
<input type="checkbox"/> sg-0d9a46000ea95453f	launch-wizard-2	launch-wizard-2 created 2020-05-13T05:34:24.807+05:30	Copy to new

Cancel
Previous
Next: Configure Routing

Step 8: Choose the name of the target group to be my target group. Click on Next: Register Targets.

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

Step 4: Configure Routing

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

Target group

Target group

Name

Target type ☒ Instance ☐ IP ☐ Lambda function

Protocol

Port

Health checks

Protocol

Path

Cancel
Previous
Next: Register Targets

Step 9: Choose instance A and instance B and click on Add to register. Click on Next: Review.

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

Step 5: Register Targets

Registered targets

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

Remove

<input type="checkbox"/>	Instance	Name	Port	State	Security groups	Zone
<input type="checkbox"/>	i-015104070505078ef	instanceA	80	● running	default	us-east-1e
<input type="checkbox"/>	i-0de23b37012599b85	instanceB	80	● running	default	us-east-1e

Instances

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

Add to registered on port 80

Search Instances							
<input type="checkbox"/>	Instance	Name	State	Security groups	Zone	Subnet ID	Subnet CIDR
<input checked="" type="checkbox"/>	i-015104070505078ef	instanceA	running	default	us-east-1e	subnet-28f6cc16	172.31.48.0/20
<input checked="" type="checkbox"/>	i-0de23b37012599b85	instanceB	running	default	us-east-1e	subnet-28f6cc16	172.31.48.0/20

Cancel Previous Next: Review

Step 10: Review all the configurations and click on create

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

Step 6: Review

Please review the load balancer details before continuing

▼ Load balancer

Name

my-loadbalancer

Scheme

internet-facing

Listeners

Port:80 - Protocol:HTTP

IP address type

ipv4

VPC

vpc-c63330bc

Subnets

subnet-07a25158, subnet-f6629590

Tags

▼ Security groups

Security groups

sg-103a4f3e

▼ Routing

Target group

New target group

Target group name

my-target-group

Port

80

Target type

instance

Protocol

HTTP

Health check protocol

HTTP

Path

/

Health check port

traffic port

Health threshold

5

Cancel Previous Create

Step 11: Congratulations!! You have successfully created a load balancer. Click on close.

Load Balancer Creation Status

✓

Successfully created load balancer
Load balancer `my-loadbalancer` was successfully created.
Note: It might take a few minutes for your load balancer to be fully set up and ready to route traffic, and for the targets to complete the registration process and pass the initial health checks.
Suggested next steps

- Discover other services that you can integrate with your load balancer. Visit the [Integrated services](#) tab within `my-loadbalancer`
- Consider using AWS Global Accelerator to further improve the availability and performance of your applications. [AWS Global Accelerator console](#)

Close

Step 12: This highlighted part is the [DNS name](#) which when copied in the URL will host the application and will distribute the incoming traffic efficiently between the two instances.

The screenshot shows the AWS Management Console interface for a newly created load balancer. The left sidebar contains navigation links for various AWS services. The main content area displays the 'Basic Configuration' tab for the load balancer 'my-loadbalancer'. The configuration details are as follows:

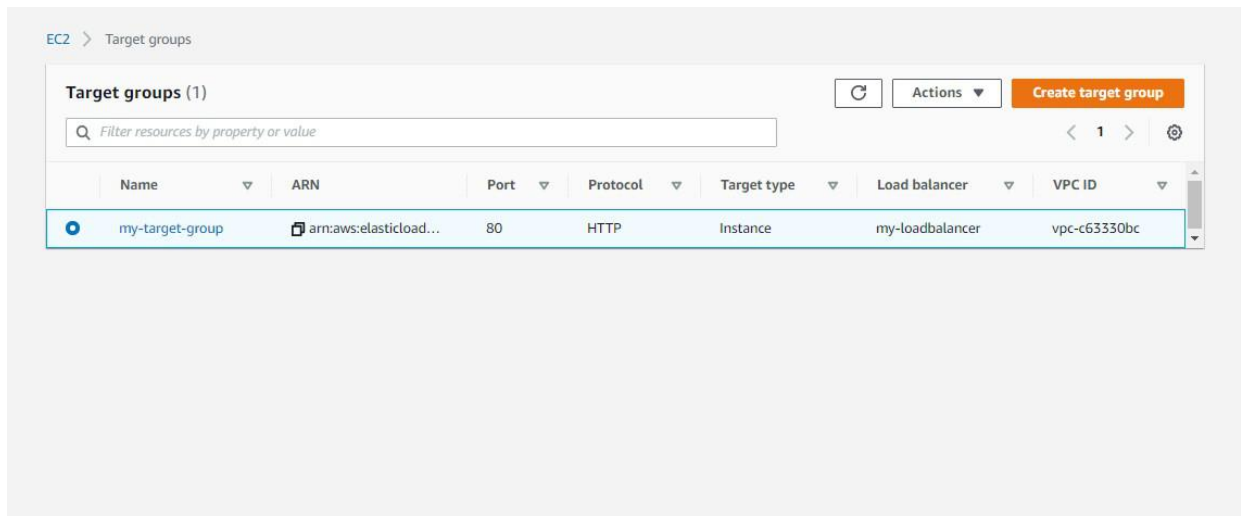
Property	Value
Name	my-loadbalancer
ARN	arn:aws:elasticloadbalancing:us-east-1:013179561180:loadbalancer/app/my-loadbalancer/ce4b8ceb87eb7694
DNS name	my-loadbalancer-2036300516.us-east-1.elb.amazonaws.com (A Record)
State	active
Type	application
Scheme	internet-facing
IP address type	ipv4
VPC	vpc-c63330bc
Availability Zones	subnet-07a25158 - us-east-1a

Step 13: This is the listener port 80 which listens to all the incoming requests

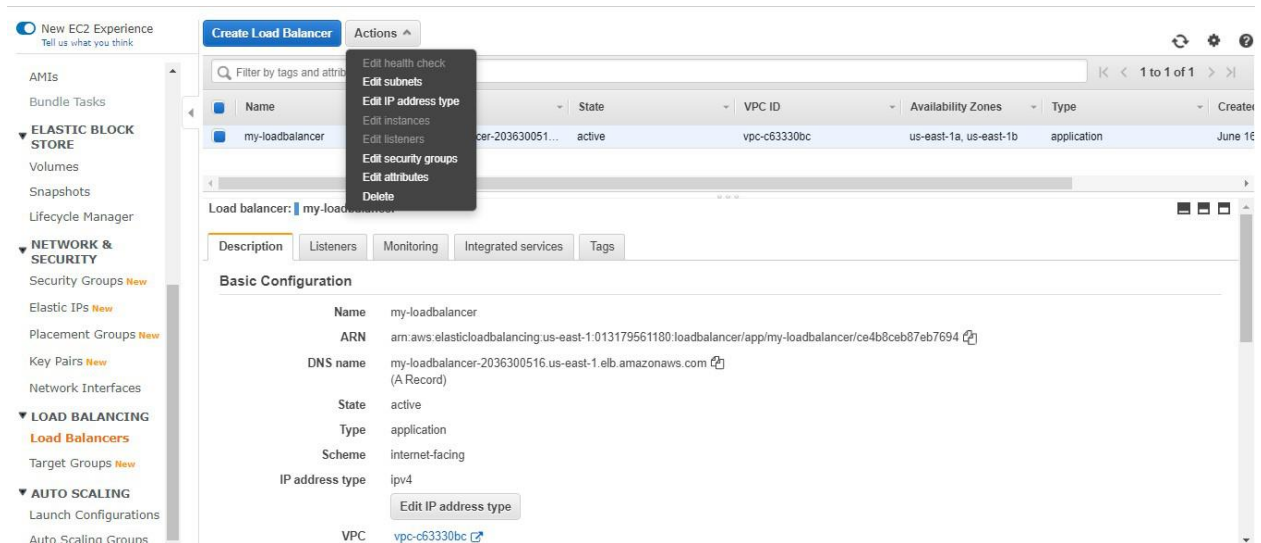
The screenshot shows the 'Listeners' tab for the load balancer 'my-loadbalancer'. It displays a table with listener information:

Listener ID	Security policy	SSL Certificate	Rules
arn:aws:elasticloadbalancing:us-east-1:013179561180:listener/app/my-loadbalancer/ce4b8ceb87eb7694/80	N/A	N/A	Default: forwarding to my-target-group

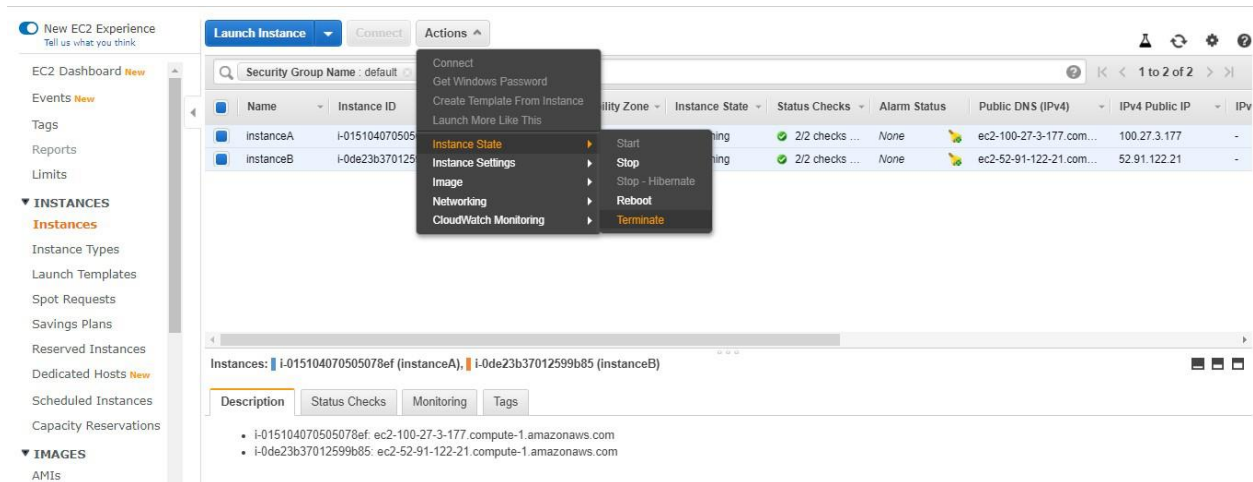
Step 14: This is the target group that we have created



Step 15: Now we need to delete the instance. Go to Actions -> Click on Delete.



Step 16: Also don't forget to terminate the instances.



Features of cloud

- No up-front investment
- Lowering operating cost
- Highly scalable and efficient
- Easy access
- Reducing business risks and maintenance expenses

Advantages of Elastic Load Balancer

- ELB automatically distributes incoming application traffic across multiple targets, such as [EC2 instances](#), [containers](#), and [IP addresses](#), to achieve high availability.
- It can automatically scale to handle changes in traffic demand, allowing you to maintain consistent application performance.
- It can monitor the health of its registered targets and route traffic only to the healthy targets.
- It evenly distributes traffic across all availability zones in a region, improving fault tolerance.

Disadvantages of Elastic Load Balancer

- ELB can add latency to your application, as traffic must pass through the load balancer before being routed to your targets.
- It has limited customization options, so you may need to use additional tools and services to fully meet your application's requirements.
- It can introduce additional complexity to your application architecture, requiring you to manage and maintain additional resources.
- It can increase your overall AWS costs, especially if you have high traffic volumes or require multiple load balancers.

FAQs On AWS Load Balancer

Q.1: What is the difference between EC2 and ELB?

Elastic load balancing will distribute the traffic to the application which is present in EC2-Instance.

Q.2: Which is faster NLB or ALB?

Network load balancer is faster than the Application load balancer when there is an sudden load.

Amazon Web Services – Auto Scaling Amazon EC2

Scalability refers to the capacity of a software solution to manage rising workloads. In simple terms, it is the ability of a system to readily add extra processing resources to handle the increased loads. Scaling [Amazon EC2](#) means you start with the resources you require at the time of starting your service and build your architecture to automatically scale in or out, in response to the changing demand. As a result, you only pay for the resources you really utilize. You don't have to be concerned about running out of computational power to satisfy your consumer's demand.

Auto Scaling

Auto Scaling is a feature in cloud computing that allows a cloud-based application to automatically adjust the resources it uses such as servers, compute instances based on demand. The goal of Auto Scaling is to ensure that the application has sufficient resources to meet performance goals and maintain availability, while also optimizing resource utilization and minimizing costs. To know the difference between Auto scaling and load balancer refer to the [Auto Scaling vs Load Balancer](#).

AWS(Amazon Web Services) Auto Scaling

AWS auto-scaling is used to scale up and scale down the EC2-instance by depending on the incoming traffic. You can scale up and scale down the applications in a few minutes based on the traffic which will decrease the latency of the application to the end-users. You can integrate the AWS Auto Scaling with multiple services provided by AWS like Amazon traffic, [Amazon DynamoDB](#) and [Amazon Aurora](#). You can also decrease the cost of an application because of dynamic scaling. When there is traffic, only maximum resources are used other it will use minimum resources.

Benefits of Auto Scaling

Here are some benefits of Auto Scaling discussed below:

- **Dynamical scaling:** AWS auto-scaling service doesn't require any type of manual intervention it will automatically scale the application down and up by depending up on the incoming traffic.
- **Pay For You Use:** In auto scaling the resource will be utilised in the optimised way where the demand is low the resource utilisation will be low and the demand will high the resource utilisation will increase so the AWS is going to charge you only for the amount of resources you really used.

- **Automatic Performance Maintenance:** AWS auto scaling maintains the optimal application performance with considering the workloads it will ensures that the application is running to desired level which will decrease the latency and also the capacity will be increased by based on your application

What is Amazon EC2 Auto Scaling?

Amazon EC2 auto-scaling will helps you to scale the resources of EC2 depending on the demand of incoming traffic. It will maintain the high availability and optimize the cost of AWS EC2.

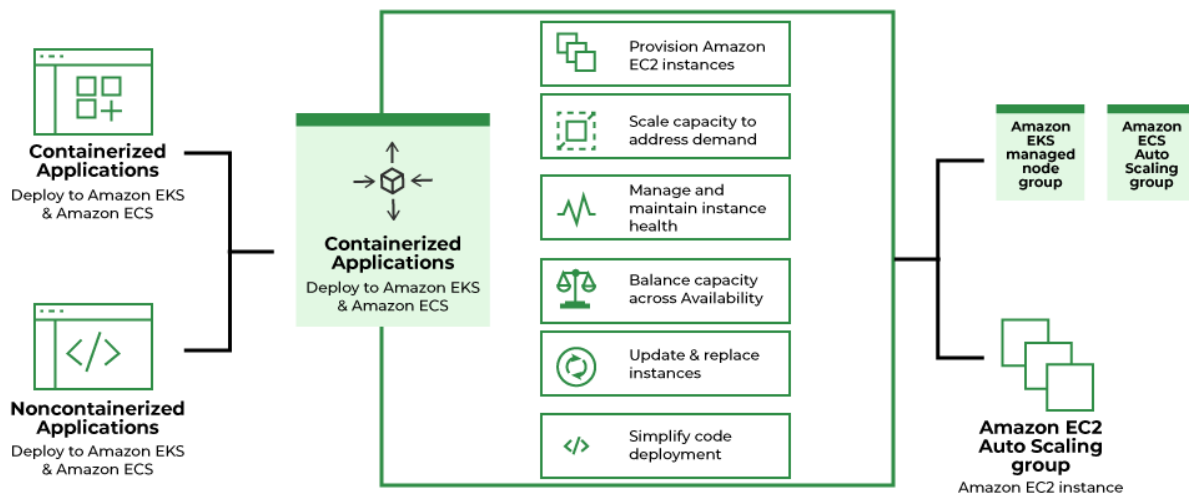
EC2 Auto Scaling is will helps to create collection of EC2 instances called an Autoscaling group where load balancer will transfer the load to this instances. The minimum, maximum, and preferred capacity for your Auto Scaling group can then be specified. To keep instances running at the appropriate capacity, EC2 Auto Scaling will start and stop them automatically.

EC2 auto scaling will offers you to configure the policies where you mention the details like at which percent of CPU utilization or memory usage you need to scale the instance based on the demand. They can be scaled automatically based on the traffic to the demand.

Auto Scaling Components

Following are the components of AWS Scaling Components.

- **Groups:**For scaling and managing the EC2 instances are grouped together so that they may be thought of as a single logical entity. You can mention the minimum and maximum no.of EC2 instance are required based up on the demand of the incoming traffic.
- **Configuration templates:** Configuration template or an launch template which is used by the EC2 autoscaling group for the EC2 instance. In which you can specify the Amazon Machine Image ID, keypair, security group and so on.
- **Scaling options:** Aws Autoscaling provides no.of options some of them are mentioned as following.
 - Dynamic scaling
 - Predictive scaling
 - Scheduled scaling
 - Manual scaling



That's the point where Amazon EC2 Autoscaling comes into the picture. You may use Amazon EC2 Auto Scaling in order to add or delete Amazon EC2 instances with respect to changes in your application demand. You can maintain a higher feeling of application availability by dynamically scaling your instances in and out as needed.

Features Of AWS (Amazon Web Services) Auto Scaling

You can use three scaling techniques within Amazon EC2 Auto Scaling i.e. Dynamic Scaling, Predictive Scaling, and Scheduled Scaling. They are explained in detail below:

- Dynamic Scaling:** Adapts to changing environments and responds with the EC2 instances as per the demand. It helps the user to follow the demand curve for the application, which ultimately helps the maintainer/user to scale the instances ahead of time. Target tracking scaling policies, for example, may be used to choose a loaded statistic for your application, such as CPU use. Alternatively, you might use Application Load Balancer's new "Request Count Per Target" measure, which is a load balancing option for the Elastic Load Balancing service. After that, Amazon EC2 Auto Scaling will modify the number of EC2 instances as needed to keep you on track.
- Predictive Scaling:** Helps you to schedule the right number of EC2 instances based on the predicted demand. You can use both dynamic and predictive scaling approaches together for faster scaling of the application. Predictive Scaling forecasts future traffic and allocates the appropriate number of EC2 instances ahead of time. Machine learning algorithms in Predictive Scaling identify changes in daily and weekly patterns and automatically update projections. In this way, the need to manually scale the instances on particular days is relieved.
- Scheduled Scaling:** As the name suggests allows you to scale your application based on the scheduled time you set. For e.g. A coffee shop owner may employ more baristas on weekends because of the increased demand and frees them on weekdays because of reduced demand.

Computing power is a programmed resource in the cloud, so you may take a more flexible approach to scale your applications. When you add Amazon EC2 Auto Scaling to an application, you may create new instances as needed and terminate them when they're no longer in use. In this way, you only pay for the instances you use, when they're in use.

Types Of AWS (Amazon Web Services) Autoscaling

- **Horizontal Scaling:** [Horizontal scaling](#) involves adding more instances to your application to handle increased demand. This can be done manually by launching additional instances, or automatically using Amazon EC2 Auto Scaling, which monitors your application's workload and adds or removes instances based on predefined rules.
- **Vertical Scaling:** [Vertical scaling](#) involves increasing the resources of existing instances, such as CPU, memory, or storage. This can be done manually by resizing instances, or automatically using Amazon EC2 Auto Scaling with launch configurations that specify instance sizes based on the workload.
- **Load Balancing:** Load balancing involves distributing incoming traffic across multiple instances to improve performance and availability. [Amazon Elastic Load Balancing \(ELB\)](#) is a service that automatically distributes incoming traffic across multiple instances in one or more Availability Zones.
- **Multi-Availability Zone Deployment:** Multi-Availability Zone (AZ) deployment involves launching instances in multiple AZs to improve availability and fault tolerance. Amazon EC2 Auto Scaling can be used to automatically launch instances in additional AZs to maintain availability in case of an AZ outage.
- **Containerization:** [Containerization](#) involves using containers to package and deploy applications, making them more portable and easier to manage. [Amazon Elastic Container Service \(ECS\)](#) is a service that makes it easy to run, stop, and manage Docker containers on a cluster of EC2 instances.

Benefits of AWS (Amazon Web Services) Auto Scaling EC2

Scaling as discussed should be implemented in an EC2 instance, in order to achieve more flexibility if the demand for application increases. Let's discuss what are the detailed benefits of Auto Scaling an EC2.

A method to make the most of AWS Cloud is to incorporate Amazon EC2 Auto Scaling into your application design. The applications benefit the following when you use Amazon EC2 Auto Scaling:

- Amazon EC2 Auto Scaling ensures that your application has enough capacity to handle current traffic demand at all times. This means your application can add or remove new and old instances respectively with respect to the demand of the application. The feature of auto adding and terminating the instances as per demand is termed as Better Availability of the application.
- Suppose an instance, becomes unhealthy by the time and is in use despite this fact. The chances of its crashing increase. Here comes another use case of Auto Scaling EC2. It will recognize which instance is not healthy or in technical terms which instance is slow, low efficient, etc, and automatically terminated the instance and replace it with a brand

new instance. Furthermore, a user can employ several availability zones with Amazon EC2 Auto Scaling. If one zone goes down or crashes, EC2 Auto Scaling compensates the same by launching instances in other zones. In this way, the traffic is can be migrated to the other zone in which new instances are added in order to manage traffic till the crashed zone gets healthy again.

- **Auto Scaling** is highly cost-efficient and must be employed if you're not sure about the traffic that your application will be receiving. As per the need, Amazon EC2 Auto Scaling can dynamically raise and reduce capacity. The user can save money by this as only according to the demand, new instances will be created and will be charged. As soon as the traffic to the application reduces, some instances get terminated and in this way, you use and pay for the instance that you really need.

Limitations of AWS (Amazon Web Services) EC2 Autoscaling

There are several limitations to consider when using Amazon EC2 Auto Scaling:

- **Number of instances:** Amazon EC2 Auto Scaling can support a maximum of 500 instances per Auto Scaling group.
- **Instance health checks:** Auto Scaling uses Amazon EC2 instance health checks to determine the health of an instance. If an instance fails a health check, Auto Scaling will terminate it and launch a new one. However, this process can take some time, which can impact the availability of your application.
- **Scaling policies:** Auto Scaling allows you to set scaling policies based on [CloudWatch metrics](#), but these policies can be complex to configure and may not always scale your application as expected.
- **Application dependencies:** If your application has dependencies on other resources or services, such as a database or cache, it may not scale as expected if those resources become overloaded or unavailable.
- **Cost:** Using Auto Scaling can increase the cost of running your application, as you may be charged for the additional instances that are launched.

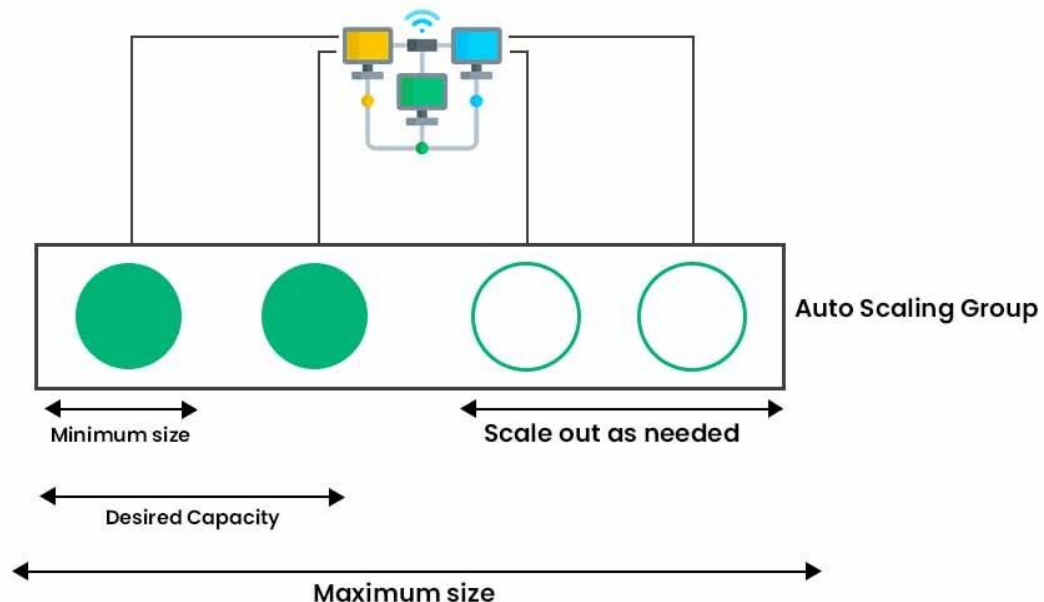
Overall, it's important to carefully consider the limitations of Amazon EC2 Auto Scaling and how they may impact your application when deciding whether to use this service. To know the difference between Auto scaling and load balancing refer to [Auto Scaling vs Load Balancer](#).

AWS (Amazon Web Services) Autoscaling For EC2 (Elastic Cloud Computing)

Amazon EC2 Autoscaling provides the liberty to automatically scale the instances as per the demand. Even if some problems are detected, the model replaces the unhealthy instances with ones that are fully functional. To automate fleet management for EC2 instances, Amazon EC2 Auto Scaling will perform three major functions:

- **Balancing the capacities across different Availability zones:** If your application has three availability zones, Amazon EC2 Autoscaling can help you balance the number of instances across the three zones. As a result, each zone receives no more or fewer instances than the others, resulting in a balanced distribution of traffic and burden.

- **Replacing and Repairing unhealthy instances:** If the instances fail to pass the health check, Autoscaling replaces them with healthy instances. As a result, the problem of instances crashing is reduced, and you won't have to manually verify their health or replace them if they're determined to be unhealthy.
- **Monitoring the health of instances:** While the instances are running, Amazon EC2 Auto Scaling ensures that they are healthy and that traffic is evenly allocated among them. It does health checks on the instances on a regular basis to see if they're experiencing any issues.



Use cases Of AWS (Amazon Web Services) AutoScaling

- **Automatic Scaling:** Application scaling can be done automatically based upon the incoming traffic if the load is increased then the application will scale up and the load decrease application will scale down automatically.
- **Schedule Scaling:** Based the data that previously available in at which particular point of time there going to be peak point and at which time there going to be less traffic we can schedule the auto scaling.
- **Integration:** You can integrate with other service in the AWS. Mainly the machine learning which will helps to predict the incoming traffic and can scale according to the traffic.

How To Configure AWS (Amazon Web Services) Auto Scaling Steps?

Auto Scaling is an Amazon Web Service it allows instances to scale when traffic or CPU load increases. Auto-scaling is a service that monitors all instances that are configured into the Auto Scaling group and ensures that loads are balanced in all instances. Depending on the load scaling group, increase the instance according to the configuration. When we created the auto-scaling group, we configured the Desired capacity, Minimum capacity, maximum capacity, and CPU utilization. If CPU utilization increases by 60% in all instances, one more instance is

created, and if CPU utilization decreases by 30% in all instances, one instance is terminated. These are totally up to us; what is our requirement. If any Instance fails due to any reason, then the Scaling group maintains the Desired capacity and starts another instance.

To know how to create autoscaling refer to [Create and Configure the Auto Scaling Group in EC2](#).

Pricing for Amazon EC2 Auto Scaling

Amazon autoscaling is free of cost there is no additional fee for using Amazon EC2 Auto Scaling. You will be charged only for the Amazon EC2 instances that you use. And also you will be charged for the resources such as CloudWatch alarms and Elastic Load Balancers.

What Is A Scaling Plan?

A blueprint for automatic Scale up or scale down of the your cloud resources in response to incoming traffic is called a scaling plan. It will give the complete outlook of resources you want to scale, the metrics you want to keep monitor, and the steps you want to take to scale those resources when their metrics rise or fall below certain levels. Many cloud resources, such as Amazon EC2 instances, [Elastic Load Balancing \(ELB\) instances](#), and [Amazon DynamoDB tables](#), can be scaled up and down by using of scaling plans. They can also be used to expand the resources of other cloud service providers, such [Google Cloud Platform](#) and [Microsoft Azure](#).

AWS Auto Scaling – FAQ'S

1. What Is The Difference Between AWS Auto Scaling And EC2 Auto Scaling?

AWS auto scaling is an service provided by the AWS which is used to scale the EC2 by depending up the in coming traffic.

2. What Are The Two Types Of Auto Scaling?

Auto scaling is mainly used to scale up and scale down the application based on the load. There are four main types of AWS autoscaling:

- 1. manual scaling,*
- 2. scheduled scaling,*
- 3. dynamic scaling, and*
- 4. predictive scaling*

3. What Are The 3 Components Of Auto Scaling Group?

The main components of autoscaling was mentioned below.

1. *Load Balancer.*
2. *Snapshot.*
3. *EC2 (Elastic Compute Cloud) Instance.*
4. *Autoscaling group.*

4. AWS Autoscaling Group Terraform

AWS Auto Scaling Group Terraform is a module that allows you to create and manage Auto Scaling groups using Terraform.

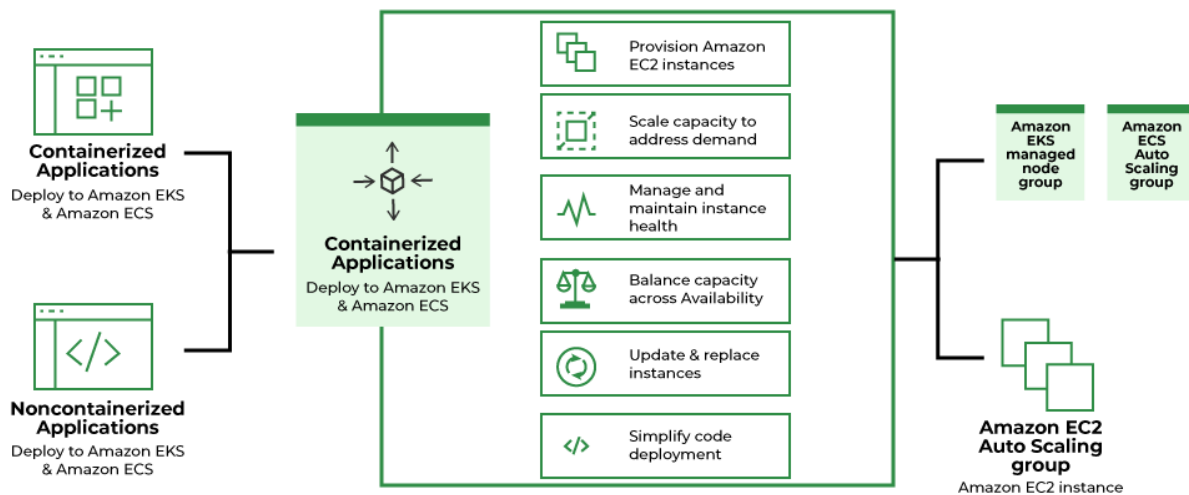
Create and Configure the Auto Scaling Group in EC2

[Auto Scaling](#) is an Amazon Web Service it allows instances to scale when traffic or CPU load increases. Auto-scaling is a service that monitors all instances that are configured into the Auto Scaling group and ensures that loads are balanced in all instances. Depending on the load scaling group, increase the instance according to the configuration. When we created the auto-scaling group, we configured the Desired capacity, Minimum capacity, maximum capacity, and CPU utilization. If CPU utilization increases by 60% in all instances, one more instance is created, and if CPU utilization decreases by 30% in all instances, one instance is terminated. These are totally up to us; what is our requirement. If any Instance fails due to any reason, then the Scaling group maintains the Desired capacity and starts another instance.

The auto-scaling group follows Horizontal Scaling. This service is very important for us nowadays because we do not need to create new instances manually and do not require manual monitoring.

AWS Auto Scaling

AWS auto scaling is used to scale up and scale down the [EC2-instance](#) by depending up on the incoming traffic. You can scale up and scale down the applications in few minutes based up on the traffic which will decrease the latency of the application to the end-users. You can integrate the AWS Auto Scaling with multiple services provided by the AWS like Amazon traffic,, [Amazon DynamoDB](#), and [Amazon Aurora](#). You can also decrease the cost of an application because of dynamic scaling. When there is traffic , only maximum resources are used other wise it will use minimum resources.

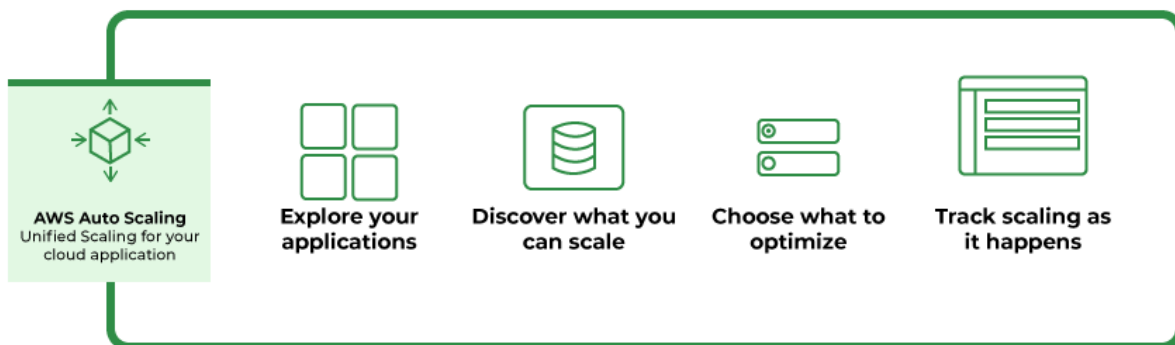


Benefits of Auto Scaling

1. **Dynamical scaling:** AWS auto-scaling service doesn't required any type of manual intervention it will automatically scale the application down and up by depending up on the incoming traffic.
2. **Pay For You Use:** Because of auto scaling the resource will be utilised in the optimised way where the demand is low the resource utilisation will be low and the demand will high the resource utilisation will increase so the AWS is going to charge you only for the amount of resources you really used.
3. **Automatic Performance Maintenance:** AWS autoscaling maintains the optimal application performance with considering the workloads it will ensures that the application is running to desired level which will decrease the latency and also the capacity will be increased by based on your application

How AWS Auto Scaling Works?

AWS autoscaling will scale the application based on the load of application. Instead of scaling manually AWS auto scaling will scale the application automatically when the incoming traffic is high it will scale up the application and when the traffic is low it will scale down the application.

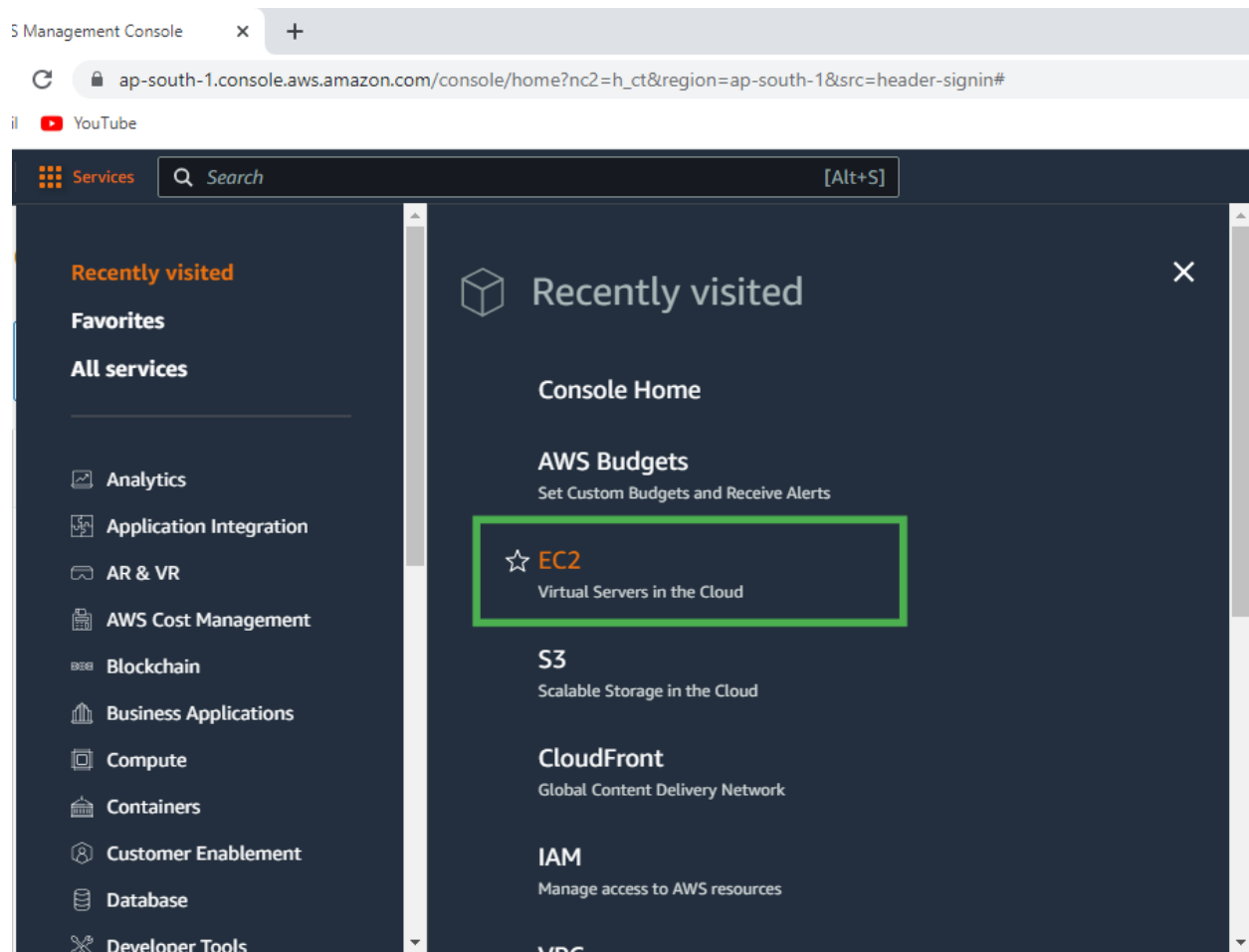


First you should choose which service or an application you want to scale then select the optimisation way like cost and performance and then keep track how the scaling is working.

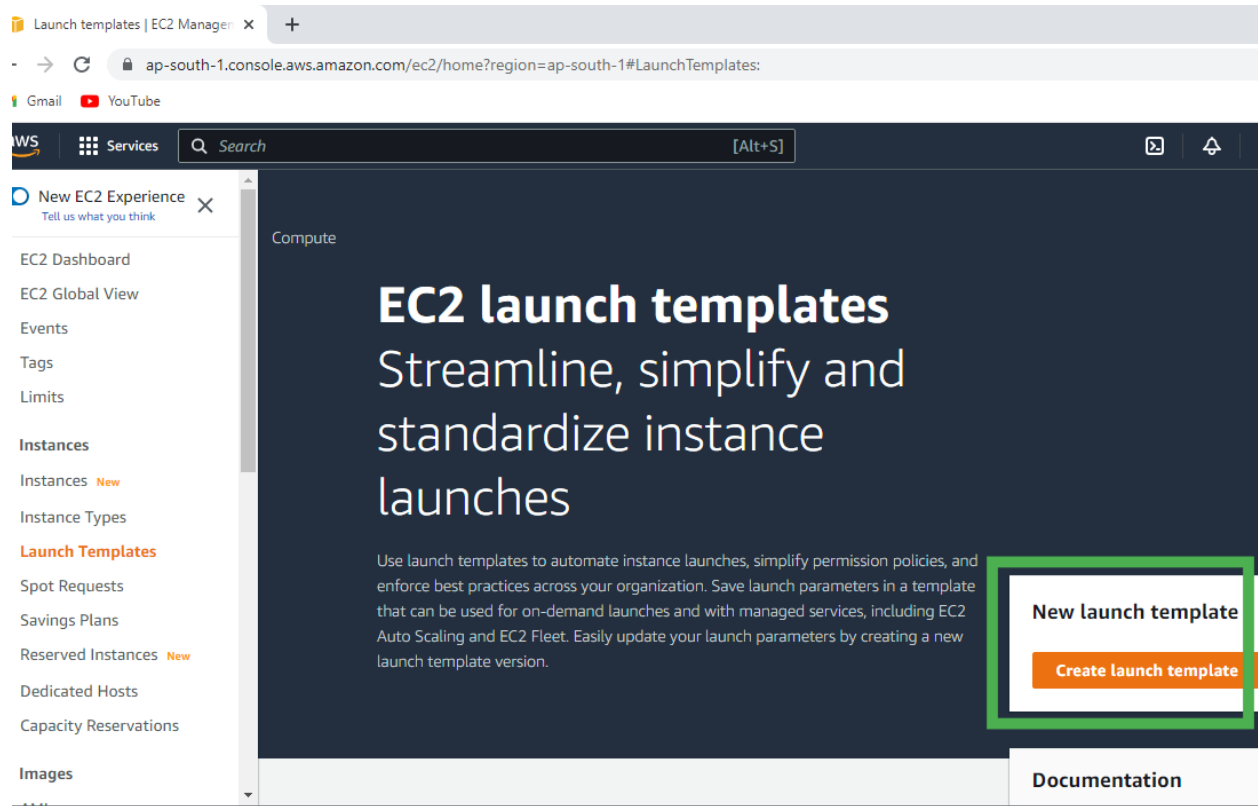
Steps To create Auto Scaling Launch Template

Step 1: Click on the All Services.

Step 2: Click on the **EC2(Elastic Cloud Computing)**.



Step 3: Scroll Down and click on the **Launch Templates** and click on the **Create launch template**



Step 4: Type the Template name.

Create launch template | EC2 Ma x +

← → ↺

ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#CreateTemplate:

Gmail YouTube

Services [Alt+S]

≡

EC2 > Launch templates > Create launch template

Create launch template

Creating a launch template allows you to create a saved instance configuration that can be reused, shared and launched at a later time. Templates can have multiple versions.

Launch template name and description

Launch template name - *required*

Must be unique to this account. Max 128 chars. No spaces or special characters like '&', '*', '@'.

Template version description

Max 255 chars

Auto Scaling guidance [Info](#)

Select this if you intend to use this template with EC2 Auto Scaling

☐ Provide guidance to help me set up a template that I can use with EC2 Auto Scaling

Step 5: Select the Amazon Machine Image.

Create launch template | EC2 Ma x +

← → ↻ ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#CreateTemplate:

Gmail YouTube

aws Services Search [Alt+S]

Specify the details of your launch template below. Leaving a field blank will result in the field not being included in the launch template.

▼ Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Recents Quick Start

Don't include in launch template


Amazon Linux
aws

macOS
Mac

Ubuntu
ubuntu

Windows
Microsoft

>


Browse more AMIs
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2 AMI (HVM) - Kernel 5.10 SSD Volume Type
ami-074dc0a6f6c764218 (64-bit (x86)) / ami-074e5caffd1685 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible ▼

Step 6: Select the Instance Type and Key pair.

Create launch template | EC2 Ma x +

← → ↺

ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#CreateTemplate:

Gmail YouTube

aws Services Search [Alt+S]

☰

▼ Instance type Info Advanced

Instance type

t2.micro Free tier eligible

Family: t2 1 vCPU 1 GiB Memory

On-Demand Linux pricing: 0.0124 USD per Hour

On-Demand Windows pricing: 0.017 USD per Hour

▼

Compare instance types

▼ Key pair (login) Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name

linux1 ▼

↻ Create new key pair

Step 7: Select the **Security Group** or Create the new one.

Create launch template | EC2 Ma x +

← → ↺

ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#CreateTemplate:

Gmail YouTube

aws Services Search [Alt+S]

≡

Subnet Info

Don't include in launch template ▼

Create new subnet [↗](#)

When you specify a subnet, a network interface is automatically added to your template.

Firewall (security groups) Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☒ Select existing security group

☐ Create security group

Security groups Info

Select security groups ▼

default sg-030f4c4f8280b6b37 ✕

VPC: vpc-066d8b2c4a30ca9f3

▶ Advanced network configuration

Step 8: Click on the Create Launch Template.

Launch template | EC2 Ma x +

ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#CreateTemplate:

YouTube

Services Search [Alt+S]

▼ Storage (volumes) Info

EBS Volumes [Hide details](#)

▶ Volume 1 (AMI Root) (8 GiB, EBS, General purpose SSD (gp2))
AMI Volumes are not included in the template unless modified

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage

Add new volume

▼ Resource tags Info

No resource tags are currently included in this template. Add a resource tag to include it in the launch template.

Add tag

50 remaining (Up to 50 tags maximum)

▼ Summary


Software Image (AMI)
Amazon Linux 2 Kernel 5.10 AMI...read n
ami-074dc0a6f6c764218

Virtual server type (instance type)
t2.micro

Firewall (security group)
default

Storage (volumes)
1 volume(s) - 8 GiB

Cancel Create launch template



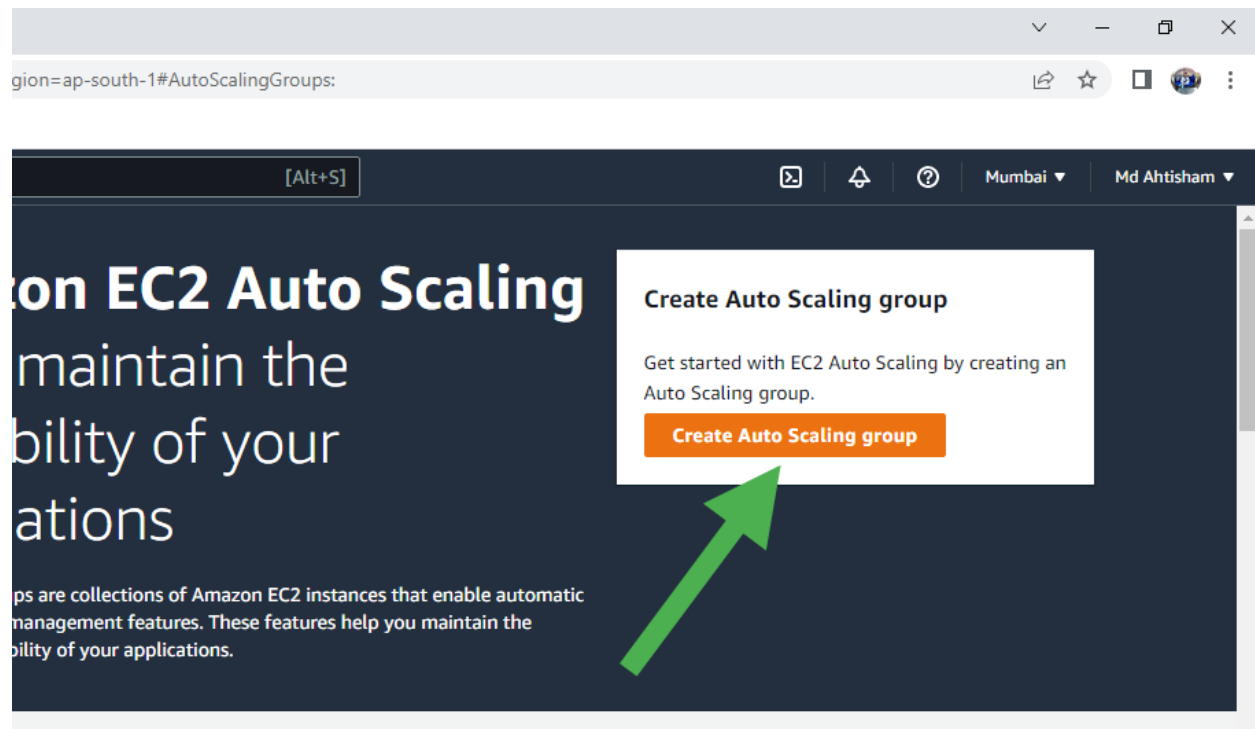
Step 9: Now you can see the template is **created**. Now, scroll down and click on the **Auto Scaling Groups**.

The screenshot shows the AWS Management Console interface. The browser tab is 'Launch templates | EC2 Manager' and the URL is 'ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#LaunchTemplates:'. The left-hand navigation menu is expanded, showing categories like 'Elastic Block Store', 'Network & Security', 'Load Balancing', and 'Auto Scaling'. Under 'Auto Scaling', 'Launch Configurations' and 'Auto Scaling Groups' are listed. A green arrow points from 'Auto Scaling Groups' to the main content area. The main content area is titled 'EC2 > Launch templates' and shows 'Launch templates (1) Info'. A table lists the launch templates:

Launch template ID	Launch template name
lt-03ecedf31ae45baeb	my-template

Create An Auto Scaling Group Using a Launch Template

Step 1: Click on the **Create Auto Scaling group**.



Step 2: Type the Auto Scaling group name.

Auto Scaling group | EC2 x +

ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#CreateAutoScalingGroup:

YouTube

Services Search [Alt+S]

Step 1

Choose launch template or configuration

Step 2

Choose instance launch options

Step 3 (optional)

Configure advanced options

Step 4 (optional)

Configure group size and scaling policies

Step 5 (optional)

Add notifications

Step 6 (optional)

Add tags

Step 7

Choose launch template or configuration [Info](#)

Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group. If you currently use launch configurations, you might consider migrating to launch templates.

Name

Auto Scaling group name

Enter a name to identify the group.

my-scaling

Must be unique to this account in the current Region and no more than 255 characters.

Launch template [Info](#)

[Switch to launch configuration](#)

Launch template

Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.

Select a launch template

[Create a launch template](#)

Step 3: Select your Template.

ite Auto Scaling group | EC2 x +

ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#CreateAutoScalingGroup:

YouTube

Services Search [Alt+S]


Step 5 (optional)
Add notifications


Step 6 (optional)
Add tags


Step 7
Review



Launch template [Info](#) [Switch to launch configuration](#)

Launch template
Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.

my-template ▼ 

Version
Default (1) ▼ 

[Create a launch template version](#) 

Description	Launch template	Instance type
-	my-template 	t2.micro
	lt-03ecedf31ae45baeb	
AMI ID	Security groups	Request Spot Instances
ami-074dc0a6f6c764218	-	No
Key pair name	Security group IDs	
linux1	sg-030f4c4f8280b6b37 	

Step 4: Select the **VPC** or go with the default VPC and also select the **Availability zone**.

Create Auto Scaling group | EC2 | x +

→ ↻ ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#CreateAutoScalingGroup:

mail YouTube

Services Search [Alt+S]

Step 2
Choose instance launch options

Step 3 (optional)
Configure advanced options

Step 4 (optional)
Configure group size and scaling policies

Step 5 (optional)
Add notifications

Step 6 (optional)
Add tags

Step 7
Review

Network Info

For most applications, you can use multiple Availability Zones and let EC2 Auto Scaling balance your instances across the zones. The default VPC and default subnets are suitable for getting started quickly.

VPC
Choose the VPC that defines the virtual network for your Auto Scaling group.

vpc-066d8b2c4a30ca9f3
172.31.0.0/16 Default

[Create a VPC](#)

Availability Zones and subnets
Define which Availability Zones and subnets your Auto Scaling group can use in the chosen VPC.

Select Availability Zones and subnets

ap-south-1a | subnet-0aea47f5b636494fe X
172.31.32.0/20 Default

ap-south-1b | subnet-05c3cc56dfcf12289 X
172.31.0.0/20 Default

ap-south-1c | subnet-0e268ddbe523359c4 X
172.31.16.0/20 Default

[Create a subnet](#)

back Looking for language selection? Find it in the new Unified Settings

Type here to search

© 2022, Amazon Internet Services Private Ltd. or its affiliates.

Step 5: Configure the **Group size** and **Scaling policies**.

Select as per your requirement:

- Desired: 4
- Minimum: 4
- Maximum: 8

The screenshot displays the AWS Management Console interface for configuring an Auto Scaling group. The browser address bar shows the URL: `ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#CreateAutoScalingGroup`. The console header includes the 'Services' menu, a search bar, and utility icons. On the left, a sidebar lists the configuration steps: Step 1 (Choose launch template or configuration), Step 2 (Choose instance launch options), Step 3 (optional) (Configure advanced options), Step 4 (optional) (Configure group size and scaling policies), Step 5 (optional) (Add notifications), Step 6 (optional) (Add tags), and Step 7. The main content area is titled 'Configure group size and scaling policies' with an 'Info' link. Below the title, a description states: 'Set the desired, minimum, and maximum capacity of your Auto Scaling group. You can optionally add a scaling policy to dynamically scale the number of instances in the group.' A section titled 'Group size - optional' with an 'Info' link contains instructions: 'Specify the size of the Auto Scaling group by changing the desired capacity. You can also specify minimum and maximum capacity limits. Your desired capacity must be within the limit range.' This section includes three input fields: 'Desired capacity' (text box with '4'), 'Minimum capacity' (text box with '4'), and 'Maximum capacity' (spinner box with '8'). A green rectangular box highlights these three input fields. The footer of the console shows the copyright notice: '© 2022, Amazon Internet Services Private Ltd. or its affiliates.' The Windows taskbar at the bottom of the screen shows the search bar and various application icons.

Step 6: Select the Target tracking scaling policy.

Auto Scaling group | EC2 | x +

ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#CreateAutoScalingGroup:

YouTube

Services Search [Alt+S]

Step 7
Review

Scaling policies - optional

Choose whether to use a scaling policy to dynamically resize your Auto Scaling group to meet changes in demand. [Info](#)

☒ **Target tracking scaling policy**
Choose a desired outcome and leave it to the scaling policy to add and remove capacity as needed to achieve that outcome.

☐ None

Scaling policy name
Target Tracking Policy

Metric type
Average CPU utilization

Target value
50

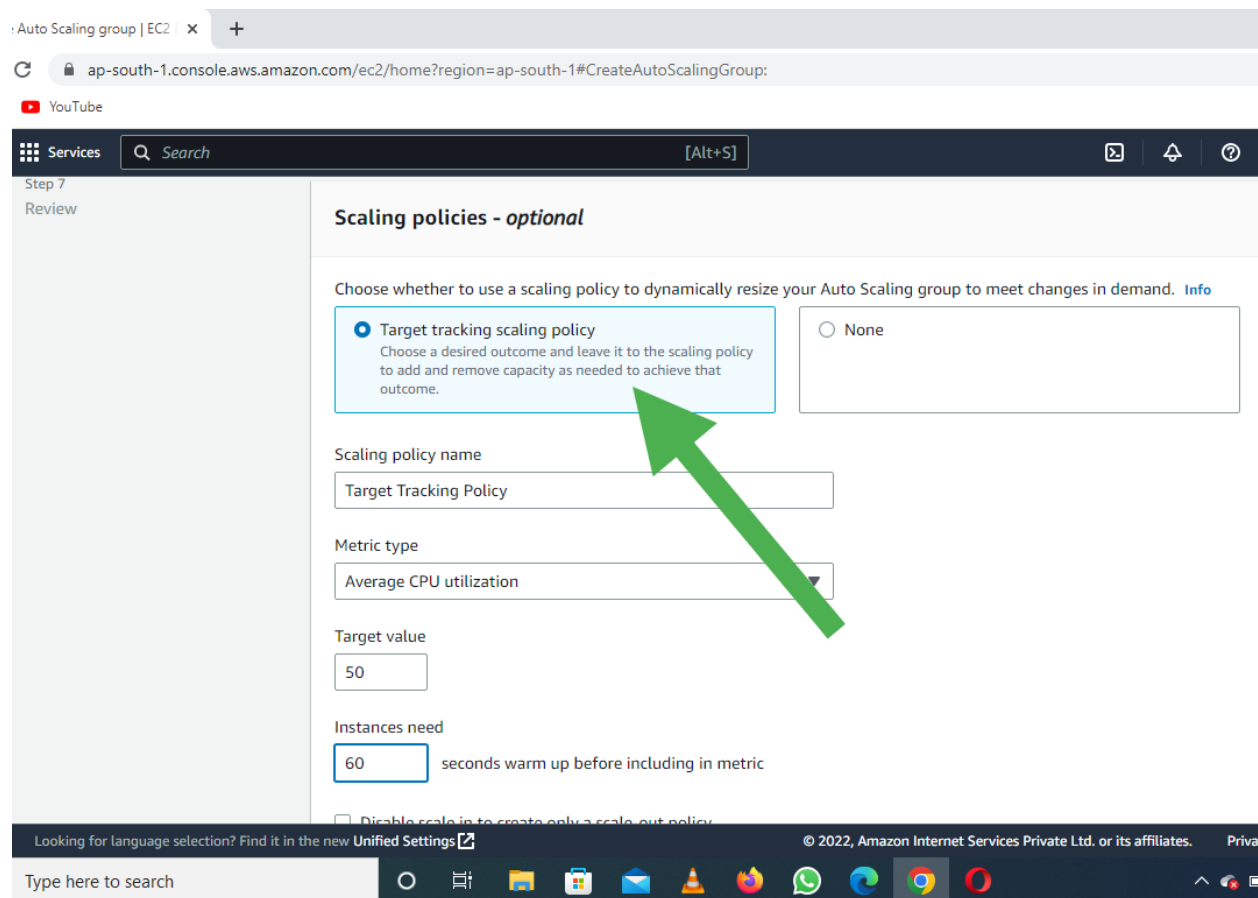
Instances need
60 seconds warm up before including in metric

☐ Disable scale in to create only a scale out policy.

Looking for language selection? Find it in the new [Unified Settings](#)

© 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy

Type here to search



Step 7: Click on the Create Auto Scaling Group.

The screenshot shows the AWS Management Console interface for creating an Auto Scaling Group. The browser address bar shows the URL: `amazon.com/ec2/home?region=ap-south-1#CreateAutoScalingGroup:`. The console header includes a search bar with `[Alt+S]`, notification icons, and the region `Mumbai`.

Step 5: Add notifications Edit

Notifications

No notifications

Step 6: Add tags Edit

Tags (0)

Key	Value	Tag new instances
No tags		

Cancel **Create Auto Scaling group**

© 2022, Amazon Internet Services Private Ltd. or its affiliates. [Privacy](#) [Terms](#)

- Now you can see the **Auto Scaling is creating** and it is also creating the desired state of the EC2 Instance

Scaling groups | EC2 Manag x +

ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#AutoScalingGroups:

YouTube

Services Search [Alt+S]

my-scaling, 1 Scaling policy created successfully

EC2 > Auto Scaling groups

Auto Scaling groups (1) Info

Search your Auto Scaling groups

<input type="checkbox"/>	Name	Launch template/configuration	Instances	Status	Desired capacity
<input type="checkbox"/>	my-scaling	my-template Version Default	0	Updating capacity...	4

0 Auto Scaling groups selected

Looking for language selection? Find it in the new Unified Settings

Type here to search

© 2022, Amazon Internet Services Private Ltd. or its affiliates. Private

- We selected the **Desired state equal to 4** and you can see the **4 Instance is Running**

The screenshot displays the AWS Management Console for EC2 instances. At the top, there's a navigation bar with the AWS logo, a search bar, and user information (Mumbai, Md Ahtishar). Below this, the 'Instances (4)' section is active, showing a table of four EC2 instances. Each instance is in the 'Running' state, has a 'Status check' of 'Initializing', and 'No alarms' are present. The instances are t2.micro type and are located in the ap-south-1 region. The table columns are: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, and Availability zone.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability zone
-	i-0cec1bd4b24be62c5	Running	t2.micro	Initializing	No alarms	ap-south-1
-	i-0c2b1fd19e7935cde	Running	t2.micro	Initializing	No alarms	ap-south-1
-	i-0acfc6f7d4b3de8be	Running	t2.micro	Initializing	No alarms	ap-south-1
-	i-096b25590ff5d1293	Running	t2.micro	Initializing	No alarms	ap-south-1

Below the table, there's a 'Select an instance' section. At the bottom of the console, there's a footer with copyright information (© 2022, Amazon Internet Services Private Ltd. or its affiliates), links to Privacy, Terms, and Cookie preferences, and a system clock showing 00:32 on 25-11-2022.

FAQs On Create And Configure The Auto Scaling Group In EC2

1. What Is The Difference Between EC2 Auto Scaling And AWS Auto Scaling?

AWS Auto-scaling is used to scale the AWS EC2 instance for better availability and productivity

2. Why Do We Need Auto Scaling?

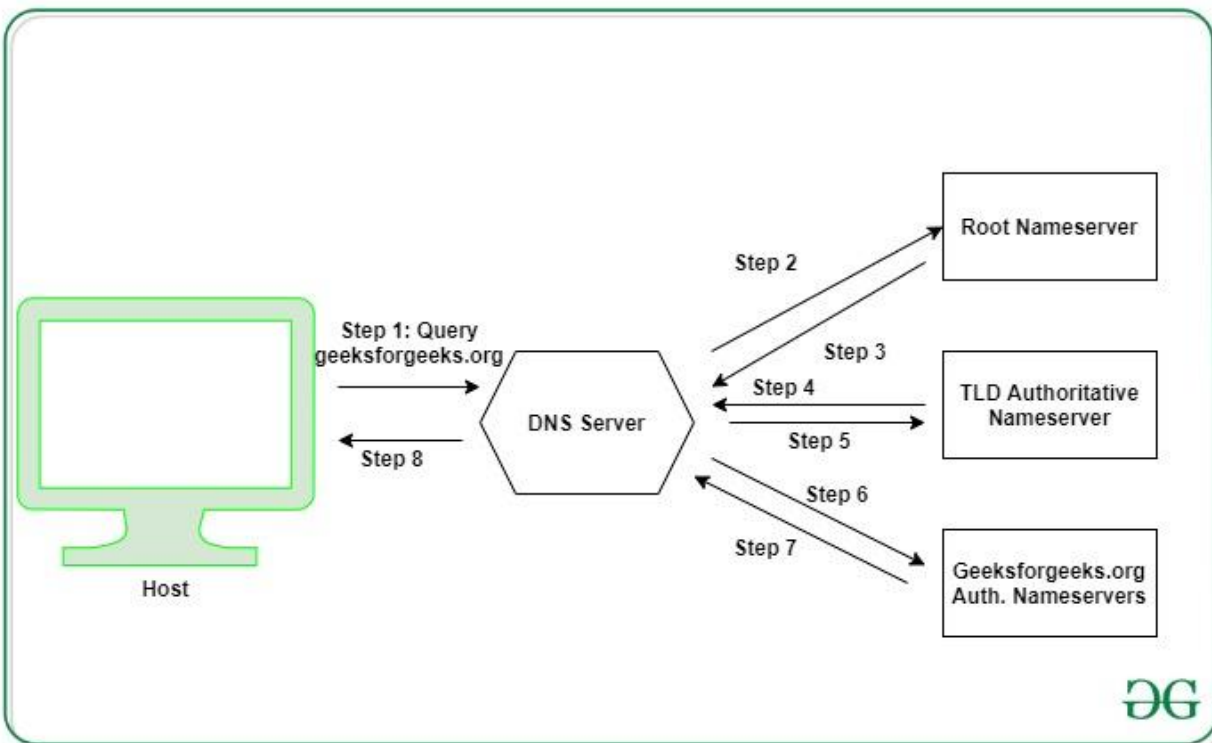
Auto scaling is essential to efficiently manage resources in response to varying workloads. It optimizes costs, ensures performance, and maintains availability by automatically adjusting resources up or down based on demand.

Amazon Web Services – Working with DNS

In this article, we will look into how DNS works, and how you can troubleshoot partial DNS failures when using AWS services. This can be done using available online tools and the command-line interface.

Let's start with a brief introduction of how DNS works. The **Domain Name System** is built using a distributed architecture. When the host needs to resolve the IP address of a domain name, the host device hands over this process to a DNS server. The DNS server finds the IP address of the domain name and returns it back to the host.

Let us walk through the process of a simple DNS query. When a customer is trying to resolve the DNS record Amazon.com this happens.

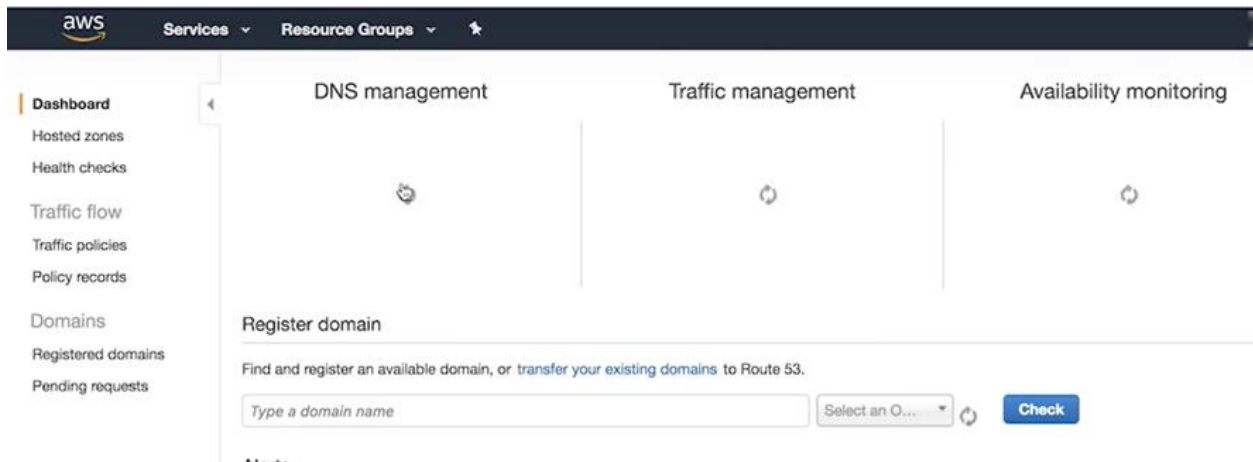


- First, the host sends the requested DNS query to the local DNS server.
- Second, the DNS server which is pre-configured with the list of root name servers randomly selects one of these root name servers and sends an interactive DNS query for the record Amazon.com.
- Third, the root name server responds with a list of authoritative name servers for the dot com zone as well as the IP addresses.
- Fourth the DNS server randomly selects one of the main returned in step three and sends another DNS query for the record Amazon.com.
- Fifth the top-level domain name server responds with a list of name servers that are authoritative for the domain Amazon.com.
- Sixth the DNS server randomly selects one of the authoritative name servers returned in step five and sends another DNS query for Amazon.com.
- Seventh since the name server receiving the query in step six is authoritative for the domain Amazon.com the name server responds to the DNS server with the value of the record Amazon.com which is an IP address.
- Finally, the DNS server sends this DNS response to the host.

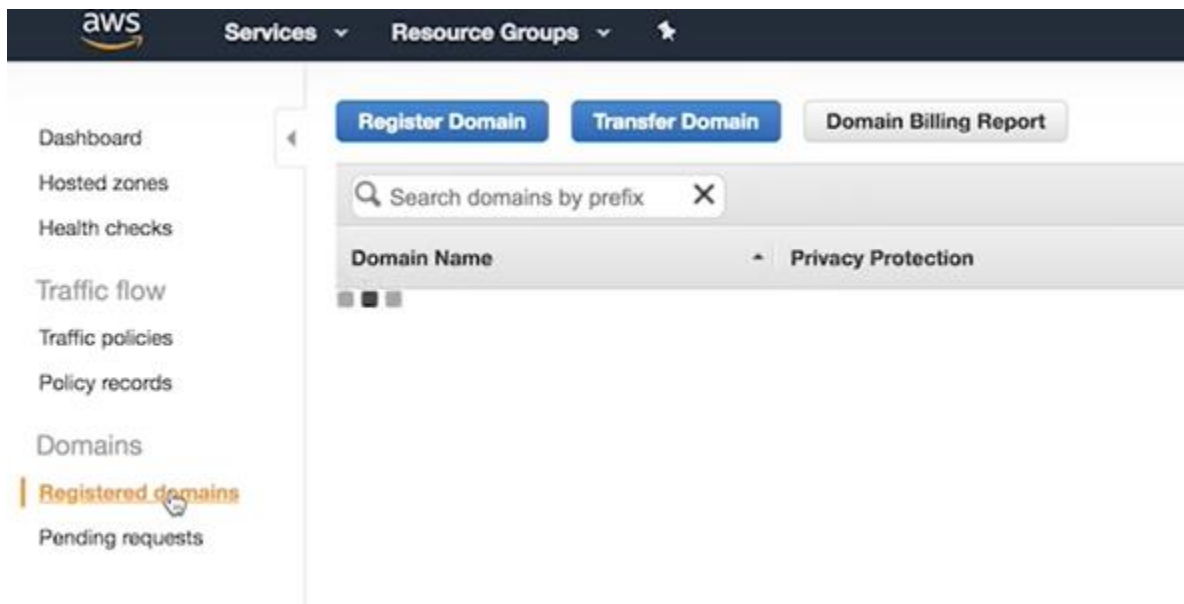
Let us walk through the process of troubleshooting common DNS issues. For example, a customer just transferred the domain "awskcvideos.com" to Route53 and cannot resolve

records in their hosted zone after completing the transfer. For DNS records to resolve properly after transferring or registering a domain on Route53, the Route53 name servers on your hosted zone need to match the name servers on the Registered Domain section on the Route53 console. Here's how you verify this.

Step 1: After you sign an AWS management console, navigate to the **Amazon Route53** console.



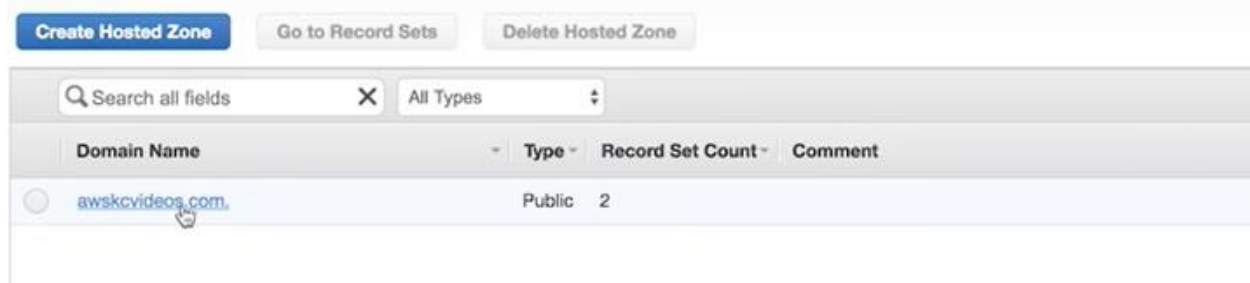
Step 2: In the navigation panel, choose Registered Domains.



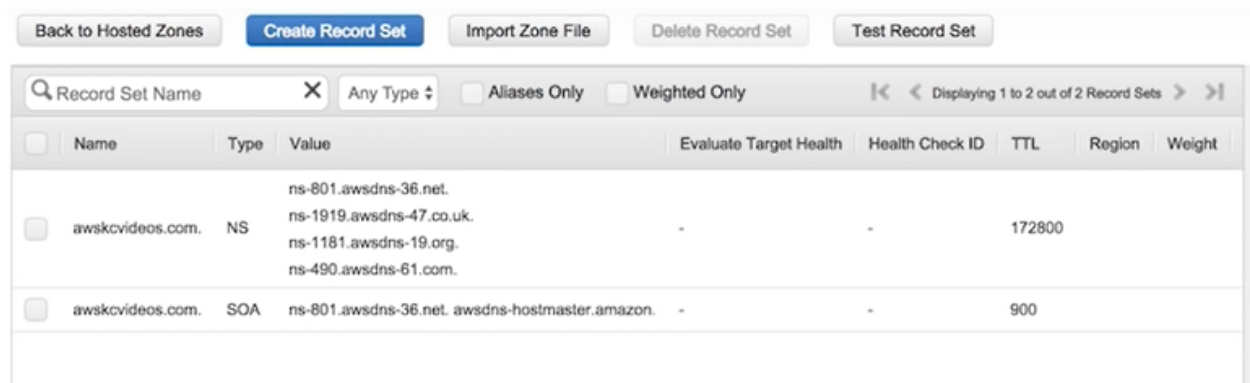
Step 3: Click the desired domain name.

Step 4: Take note of the four nameservers.

Step 5: In the navigation panel choose the hosted zone and click the domain name.



Step 6: Take note of the four name servers in the hosted zone and verify the name servers listed in both sections marked.



In this case, the customer with the domain “awskcvideos.com” will not be able to resolve any record in their hosted zone. The solution is to update the nameservers in the registered domain section with the nameservers in their Route53 hosted zone.

Other common DNS errors include SERVFAIL and REFUSED. A SERVFAIL DNS response indicates the DNS server was unable to process this query due to a problem with the authoritative nameserver. A SERVFAIL response is also a common response for certain DNSSEC validations that are unsuccessful. To fix this, verify the DNS service provider supports DNSSEC and the authoritative nameserver on the Registered Domain section are reachable and valid for the domain. A REFUSED response indicates the nameserver is not authoritative for that domain, meaning it does not have the records for that domain in its zone file. To fix this, verify the correct nameservers in the hosted zone updated on the Registered Domain section of the Route53 console.

Introduction to Amazon Route53

In Simplest terms, cloud computing means storing and accessing the data and programs on remote servers that are hosted on the internet instead of the computer's hard drive or local server. It is also referred to as Internet-based computing.

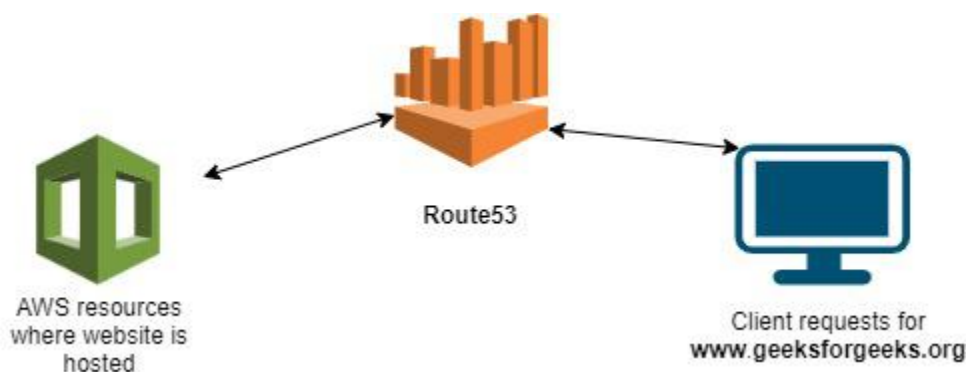
Features of cloud

- No up-front investment
- Lowering operating cost
- Highly scalable
- Easy access
- Reducing business risks and maintenance expenses
- No need to guess the capacity
- Flexible

Amazon Web Services is a subsidiary of Amazon.com that provides on-demand cloud computing platforms to individuals, companies, and governments, on a paid subscription basis.

Amazon Route53

Amazon Route 53 is a highly available and scalable cloud Domain Name System (DNS) web service. It is basically designed for developers and corporate to route the end users to Internet applications by translating human-readable names like www.geeksforgeeks.org into the numeric IP addresses like 192.0.1.1 that computers use to connect to each other. You cannot use Amazon Route 53 to connect your on-premises network with AWS Cloud.



Functions of Route53

- If a web application requires a domain name, Route53 service helps to register the name for the website (i.e domain name).
- Whenever a user enters the domain name, Route53 helps to connect the user to the website.
- If any failure is detected at any level, it automatically routes the user to a healthy resource.

- Amazon Route 53 is cost effective, secure and scalable.
- Amazon Route 53 is flexible, highly available and reliable.

Methodologies related to Route53

- **Records:** Records are created to route internet traffic to the resources. They are the objects present in the hosted zone which determines how the internet traffic has to be routed for a domain name so that it finally reaches the resources. The name of each record in a hosted zone must end with the name of the hosted zone.
- **Hosted zone:** When the domain name is registered, Route53 creates a public hosted zone that has the same name as the domain name. It is a collection of records that contains information about how to route traffic of its domains and all of its subdomains.
- **DNS query:** It is a request for information sent from DNS client to the DNS server.
- **Alias records:** Alias records helps in routing internet traffic to AWS resources like S3 bucket, Amazon CloudFront, etc. It is created at the top node of the DNS namespace.
- **Name servers:** They are the servers in the DNS that translates the domain name into IP address so that internet traffic can be routed to the resources.
- **DNS failover:** A method for routing the traffic from unhealthy resources to healthy resources, whenever a failure is detected.
- **Routing policy:** Routing policy determines how Amazon Route53 responds to queries.

Types of Routing Policy

- **Simple routing policy:** It is a simple Route53 routing technique that can be used to route internet traffic to a single resource. For example; Web server to a website. Using this, routing multiple records with the same name cannot be created but multiple values (such as multiple IP addresses) can be specified in the same record.
- **Failover routing policy:** Whenever a resource goes unhealthy, this policy allows to route the traffic from unhealthy resource to healthy resource.
- **Geolocation routing policy:** This routing policy routes the traffic to resources on the basis of the geographic location of the user. Geographic locations can be specified by continent, country, or state. For example; A person residing in France will be redirected to the website in the French language while a person from the US will be redirected to the website in the English language.
- **Geoproximity routing policy:** It routes traffic on the basis of the geographical location of the user and the type of content user wants to access. The user can optionally shift traffic from resources at one location to resource at another location. Using this policy, a user can shift more traffic to one location compared to another location by specifying a value known as **bias**.
- **Latency routing policy:** If a website has to be hosted in multiple regions then a latency based routing policy is used. To improve performance for the users, this policy helps in

serving requests from the AWS region that provides the lowest latency. To use this policy the latency records for the resources are created in multiple AWS regions.

- **Multivalue routing policy:** It is used when users want Route53 to return multiple values in response to DNS queries. It first checks the health of resources and then returns the multiple values only for the health resources.
- **Weighted routing policy:** This routing policy routes traffic to multiple resources with a single domain name according to the proportion decided by the user.

Benefits of Route53

- **Highly Reliable:** Route53 is built using AWS's highly available and reliable infrastructure. The distributed nature of the AWS DNS servers helps ensure a consistent ability to route the end-users to the web application.
- **Scalable:** It automatically scales the resources during large traffic and also handles large queries without the user's intervention.
- **Easy to use:** Very user-friendly and easy to configure DNS settings. It can start to answer your DNS queries within minutes. Can be mapped easily to any resource.
- **Health Check:** Route 53 monitors the health of the application. If any failure is detected, it automatically redirects the user to a healthy resource before the customer can identify the problem.
- **Flexible:** You can decide which policy you want to use at given time.
- **Simple:** Using routing types, Route53 helps to manage traffic globally.
- **Cost-effective:** Payment is done only according to the services used.
- **Secure:** By integrating it with IAM, the access to Amazon Route53 is secured by giving its permissions to only the authorized users.
- **Mapped with various AWS services:** It can be used to map domain names to Amazon EC2 instances, S3 buckets, and other AWS resources.

Amazon VPC – Introduction to Amazon Virtual Private Cloud

Amazon VPC or **Amazon Virtual Private Cloud** is a service that allows its users to launch their virtual machines in a protected as well as isolated virtual environment defined by them. You have complete control over your VPC, from creation to customization and even deletion. It's applicable to organizations where the data is scattered and needs to be managed well. In other words, VPC enables us to select the virtual address of our private cloud and we can also define all the sub-constituents of the VPC like subnet, subnet mask, availability zone, etc on our own.

- We can place the necessary resources and manage access to those resources in the VPC, a private area of Amazon that we control.
- A default “VPC” will be generated when we register an [AWS account](#), allowing us to manage the virtual networking environment, the [IP address](#), the construction of subnets, route tables, and gateways.

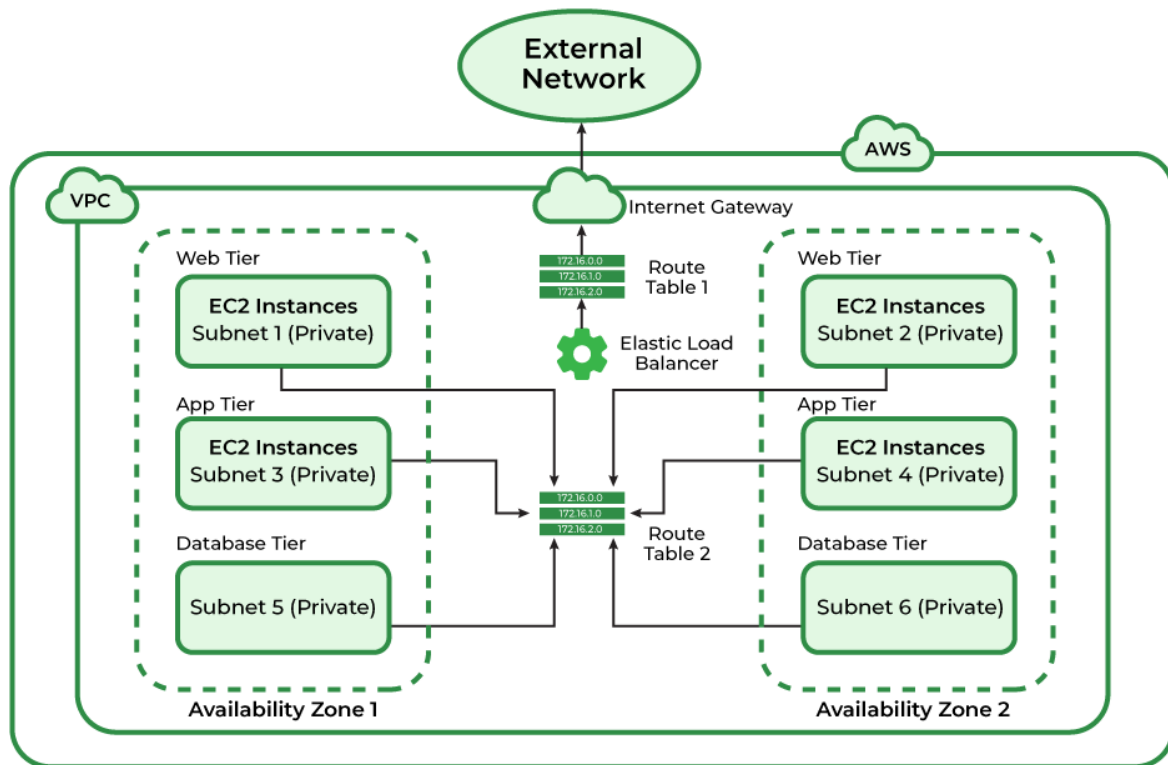
What is Amazon VPC(Virtual Private Cloud)?

[Amazon](#) VPC can be referred to as the private cloud inside the cloud. It is a logical grouping of servers in a specified network. The servers that you are going to deploy in the Virtual Private Cloud(VPC) will be completely isolated from the other servers that are deployed in the Amazon Web Services. You can have complete control of the IP address to the virtual machines and route tables and gateways to the VPC. With the help of security groups and network access control lists, you can protect your application more.

Amazon VPC (Virtual Private Cloud) Architecture

The basic architecture of a properly functioning VPC consists of many distinct services such as [Gateway](#), [Load Balancer](#), [Subnets](#), etc. Altogether, these resources are clubbed under a VPC to create an isolated virtual environment. Along with these services, there are also security checks on multiple levels.

It is initially divided into subnets, connected with each other via route tables along with a load balancer.



Amazon VPC (Virtual Private Cloud) Components

VPC

You can launch AWS resources into a defined virtual network using Amazon Virtual Private Cloud (Amazon VPC). With the advantages of utilizing the scalable infrastructure of AWS, this virtual network closely mimics a conventional network that you would operate in your own data center. /16 user-defined address space maximum (65,536 addresses)

Subnetes

To reduce traffic, the subnet will divide the big network into smaller, connected networks. Up to /16, 200 user-defined [subnets](#).

Route Tables

[Route Tables](#) are mainly used to Define the protocol for traffic routing between the subnets.

Network Access Control Lists

[Network Access Control Lists \(NACL\)](#) for VPC serve as a firewall by managing both inbound and outbound rules. There will be a default NACL for each VPC that cannot be deleted.

Internet Gateway(IGW)

he [Internet Gateway \(IGW\)](#) will make it possible to link the resources in the VPC to the Internet.

Network Address Translation (NAT)

[Network Address Translation \(NAT\)](#) will enable the connection between the private subnet and the internet.

Amazon VPC (Virtual Private Cloud) Fundamentals

- If the subnet has internet access then it is called PublicSubnet.
- If the subnet doesn't have internet access then it is called PrivateSubnet.
- A subnet must reside entirely within one Availability Zone.
- An entire subnet must be contained within a single Availability Zone.
- Access between instances is managed by VPC Security Groups for both inbound and outgoing traffic (EC2 Security Groups can only define inbound rules).
- We can specify Subnet IP Routing with the aid of the Route Table.
- If a server/instance which is in a private subnet wants to reach the internet then it must have NAT in a public subnet.

Subnet

- A subnet is a smaller portion of the network that typically includes all the machines in a certain area.
- We can add as many as subnets we need in one availability zone. Each subnet must reside entirely within one availability zone.
- The public subnets will be attached to Internet Gateway which enables Internet access.
- The private subnets will not have internet access.
- Each and every subnet which is presented in VPC must be associated with the routing table.

Internet Gateway

- With the help of **IGW** (Internet Gateway), the resources present (e.g: [EC2](#)) in the VPC will enable to access the Internet.
- One VPC can't have more than one IGW
- If resources are running in a certain VPC then IGW can not be detached from that particular VPC.

Route Table

- Route Table contains a set of rules, called route which helps us to route the network traffic.
- A single VPC can have as many as route tables it requires.
- If the dependencies are attached to the [route table](#) then they can't be deleted.

NACL Network Access Control Lists

- The NACL security layer for VPC serves as a firewall to manage traffic entering and leaving one or more subnets.
- The NACL for the default VPC is active and connected to the default subnets.

Classless Inter-Domain Routing (CIDR)

- A technique for allocating IP addresses and for IP routing is called classless [Inter-Domain Routing \(CIDR\)](#), and its range is 0-32.
- When setting up a VPC, we must specify a set of IPv4 addresses using [classless Inter-Domain Routing \(CIDR\)](#), for (**Example:**10.0.0.0/16 For our VPC, this will serve as the main CIDR block).

RFC1918 Address(Private address)

- An enterprise organization will give an internal host an IP address known as an RFC1918 address. These IP addresses are employed in private networks that cannot be accessed or accessed through the internet.

The following networks are included in the RFC1918 address(Private address)

10.0.0.0 -10.255.255.255 (10/8 prefix)

172.16.0.0 - 172.31.255.255 (172.16/12 prefix)

192.168.0.0 - 192.168.255.255 (192.168/16 prefix)

Amazon VPC Network Address Translation (NAT)

- RFC1918 address is a workable solution to IPv4 address exhaustion issues thanks to [Network Address Translation \(NAT\)](#).
- An internal host can communicate with an internet server with help of NAT.
- The internet and a private network are separated by a NAT device.

Use cases of Amazon VPC

- Using VPC, you can host a public-facing website, a single-tier basic web application, or just a plain old website.
- The connectivity between our web servers, application servers, and database can be limited by VPC with the help of [VPC peering](#).
- By managing the inbound and outbound connections, we can restrict the incoming and outgoing security of our application.

Amazon VPC (Virtual Private Cloud) Working

Follow the Steps Mentioned Below To Configure Virtual Private Cloud(VPC)

Step 1: Login into AWS Console and navigate to the VPC as shown below.

Step 2: After navigating to the AWS VPC console click on create VPC.

✔ You successfully deleted vpc-098d155bcac43e21e / GFG-VPC

Create VPC

Launch EC2 Instances

Note: Your Instances will launch in the US East region.

Resources by Region

↻ Refresh Resources

You are using the following Amazon VPC resources

<div>VPCs</div> <div>See all regions ▼</div> <div>US East 1</div>	<div>NAT Gateways</div> <div>See all regions ▼</div> <div>US East 0</div>
<div>Subnets</div> <div>See all regions ▼</div> <div>US East 6</div>	<div>VPC Peering Connections</div> <div>See all regions ▼</div> <div>US East 0</div>
<div>Route Tables</div> <div>See all regions ▼</div> <div>US East 1</div>	<div>Network ACLs</div> <div>See all regions ▼</div> <div>US East 1</div>
<div>Internet Gateways</div> <div>See all regions ▼</div> <div>US East 1</div>	<div>Security Groups</div> <div>See all regions ▼</div> <div>US East 1</div>
<div>Egress-only Internet Gateways</div> <div>See all regions ▼</div> <div>US East 0</div>	<div>Customer Gateways</div> <div>See all regions ▼</div> <div>US East 0</div>
<div>DHCP option sets</div> <div>US East 1</div>	<div>Virtual Private Gateways</div> <div>US East 0</div>

Step 3: Configure all the details required to create as shown in the image below. Some of the most required settings to configure VPC was as follows

- Name of the Network.
- IPv4 CIDR.
- And tags of VPC after that click on create VPC.

Create VPC Info

A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances.

VPC settings

Resources to create Info
Create only the VPC resource or the VPC and other networking resources.

☒ VPC only ☐ VPC and more

Name tag - optional
Creates a tag with a key of 'Name' and a value that you specify.

GFG-VPC

IPv4 CIDR block Info
☒ IPv4 CIDR manual input
☐ IPAM-allocated IPv4 CIDR block

IPv4 CIDR
10.0.0.0/24
CIDR block size must be between /16 and /28.

IPv6 CIDR block Info
☒ No IPv6 CIDR block
☐ IPAM-allocated IPv6 CIDR block
☐ Amazon-provided IPv6 CIDR block
☐ IPv6 CIDR owned by me

Tenancy Info
Default

Tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key: Name Value - optional: GFG-VPC

Remove tag

Add tag

You can add 40 more tags.

Cancel Create VPC

Step 4: Virtual Private Cloud Created successfully with the required setting to us.

Step 6: Check the VPC dashboard whether the VPC created is available to use as shown in the image below GFG-VPC.

<input type="checkbox"/>	Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR
<input type="checkbox"/>	-	vpc-08d4ac89e9417b4bb	Available	172.31.0.0/16	-
<input type="checkbox"/>	GFG-VPC	vpc-098d155bcac43e21e	Available	10.0.0.0/24	-

What Is AWS VPC (Virtual Private Cloud) Peering?

Amazon Virtual Private Cloud (VPC) is a private cloud where you can deploy your AWS Virtual machines with controlled access. It is completely isolated from each other; the servers which are in one VPC cannot communicate with the other virtual machines in another virtual private network.

VPC peering can establish the connection between two Virtual Private Clouds, which enables you to route the traffic between two VPCs using the IP address. The virtual servers which are in the same network can communicate with each other without VPC peering connections, but the servers which are in two different networks can't communicate with each other without [VPC peering](#).

AWS VPC (Virtual Private Cloud) Console

We can create and manage VPCs using the AWS Management Console. Log in to your AWS account. Once you are redirected to the AWS management console, click on “**Services**”, and a list of options will be visible. Under “**Networking & Content Delivery**” there is an option named **VPC**, and there is the navigation pane, which consists of various services as options. Refer to the image attached ahead for a better understanding. We have discussed some of the important resources.

AWS Command Line Interface (AWS CLI)

We may issue commands on our own (OS) by using [Windows](#), [Mac](#), and [Linux](#) computers using AWS command line tools (OS). By using the command line, we can make it more expedient and quick than the console.

FAQs On Amazon VPC(Virtual Private Cloud)

1. Amazon VPC Full Form

The full form of Amazon VPC is Virtual Private Cloud which is isolated cloud within the Amazon Web Service Cloud.

2. Amazon VPC Traffic Mirroring

Amazon VPC traffic mirroring is a feature provided by Amazon by which you can replicate the traffic from source instance to the target instance for the analysis or troubleshooting.

3. Amazon VPC Lattice

Amazon VPC is a fully managed application networking service which streamlines the process of connecting, securing and monitoring applications across multiple AWS accounts and VPCs which will reduce the efforts of developers and can focus on the developing part.

4. Amazon VPC Flow Logs

Amazon VPC flow logs will help you to monitor carefully the in and out of the traffic through the network. The logs will be stored in the CloudWatch, Amazon S3 or Amazon Kinesis data firehose.

Introduction to Microsoft Azure | A cloud computing service

Designed by Microsoft in 2010, Microsoft Azure is one of the widely used cloud computing platforms. Azure provides a wide variety of services such as cloud storage, compute services, network services, cognitive services, databases, analytics, and IoT. It makes building, deploying, and managing applications very easy. All the Microsoft Azure fundamentals are also described for a better understanding of readers.

What is Azure?

[Azure is Microsoft's](#) cloud platform, just like Google has its Google Cloud and Amazon has its Amazon Web Service or AWS. Generally, it is a platform through which we can use Microsoft's resources. For example, to set up a huge server, we will require huge investment, effort, physical space, and so on. In such situations, Microsoft Azure comes to our rescue. It will provide us with virtual machines, fast processing of data, analytical and monitoring tools, and so on to make our work simpler. The pricing of Azure is also simpler and cost-effective. Popularly termed as "Pay As You Go", which means how much you use, pay only for that.

How Does Microsoft Azure Work?

It is a private and public cloud platform that helps developers and IT professionals to build, deploy, and manage the application. It uses the technology known as virtualization. Virtualization separates the tight coupling between the hardware and the operating system using an abstraction layer called a hypervisor. Hypervisor emulates all the functions of a computer in the virtual machine; it can run multiple virtual machines at the same time, and each virtual machine can run any operating system such as Windows or Linux. Azure takes this virtualization technique and repeats it on a massive scale in the data center owned by Microsoft. Each data center has many racks filled with servers, and each server includes a hypervisor to run multiple virtual machines. The network switch provides connectivity to all those servers.

Azure will provide the Microsoft Azure is a cloud computing platform which offers

- Infrastructure as a service (IaaS).
- Platform as a service (PaaS).
- Software as a service (SaaS).

Infrastructure as a service (IaaS)

Virtual machines, storage, and networking will come under the category of infrastructure as a service, but the users have to do manually the build and deploy of the applications. Azure will support a wide range of operating systems because of its Hyper-hypervisor.

Platform as a service (PaaS)

Azure app service, Azure functions, and logic apps are some services that are offered by Azure under the platform as a service. This service will provide autoscaling and load balancing, and also there will be a pre-configured environment for the application.

Software as a service (SaaS)

Office 365, Dynamics 365, and Azure Active Directory are some of the services provided by Microsoft Azure under Software as a Service (SaaS); the complete application will be managed by the Microsoft Azure, including deploying, scaling, and load balancing.

What is a public cloud? Everything you need to know

Computing in which service provider makes all resources public over the internet. It is connected to the public Internet. Service provider serves resources such as virtual machines, applications, storage, etc. to the general public over the internet. It may be free of cost or with minimal pay-per-usage. It is available for public display; Google uses the cloud to run some of its applications like Google Docs, Google Drive, or YouTube, etc.

It is the most common way of implementing cloud computing. The external cloud service provider owns, operates, and delivers it over the public network. It is best for the companies which need an infrastructure to accommodate a large number of customers and work on projects which have diverse organizations i.e. research institutions and NGOs, etc.

Key Characteristics of Public Clouds:

- Accessibility
- Scalability
- Cost-effectiveness
- Security
- Reliability

What Is Microsoft Azure Used For?

Following are the some the use cases that Microsoft Azure Used.

- **Deployment Of applications:** You can develop and deploy the application in the azure cloud by using the service called Azure App Service and Azure Functions after deploying the applications end users can access it.
- **Identity and Access Managment:** The application and data which is deployed and stored in the Microsoft Azure can be secured with the help of Identity and Access Managment. It's commonly used for single sign-on, multi-factor authentication, and identity governance.
- **Data Storage and Databases:** You can store the data in Microsoft azure in service like blob storage for unstructured data, table storage for NoSQL data, file storage, and Azure SQL Database for relational databases. The service can be scaled depending on the amount of data we are getting.
- **DevOps and Continuous Integration/Continuous Deployment (CI/CD):** Azure DevOps will provide some tools like ncluding version control, build automation, release management, and application monitoring.

Azure for DR and Backup

A full range of disaster recovery (DR) and backup services are available from Microsoft Azure to help shield your vital data and apps from interruptions. With the help of these services, you may quickly restore your data and applications in the event of a disaster by replicating them to a secondary cloud site. Azure backup services also protect your data from ransomware attacks, unintentional deletion, and corruption.

Key Azure DR and Backup Services

- **Azure Site Recovery:** Your on-premises virtual machines (VMs) can be replicated to Azure more easily with the help of this solution. You may easily failover your virtual machines (VMs) to Azure in the event of a disaster and keep your business running. Azure VM replication to an alternative Azure region is also supported by Azure Site Recovery.
- **Azure Backup:** If you want to protect the data which is present in the cloud then you need to use the Azure Backup service. It offers a single area to monitor backup jobs, manage backup policies, and recover data. Azure pricing and costs.

Azure competition

Following are the some of the competitors of Microsoft Azure:

- Amazon Web Services (AWS).
- Google Cloud Platform (GCP).
- IBM Cloud.
- Alibaba Cloud.
- Oracle Cloud Infrastructure (OCI).

Azure History

Microsoft unveiled Windows Azure in early October 2008 but it went to live after February 2010. Later in 2014, Microsoft changed its name from Windows Azure to Microsoft Azure. Azure provided a service platform for .NET services, SQL Services, and many Live Services. Many people were still very skeptical about “the cloud”. As an industry, we were entering a brave new world with many possibilities. Microsoft Azure is getting bigger and better in the coming days. More tools and more functionalities are being added. It has two releases as of now. It's a famous version of **Microsoft Azure v1** and later **Microsoft Azure v2**. Microsoft Azure v1 was more JSON script-driven than the new version v2, which has interactive UI for simplification and easy learning. Microsoft Azure v2 is still in the preview version.

How Azure can help in business?

Azure can help our business in the following ways-

- **Capital less:** We don't have to worry about the capital as Azure cuts out the high cost of hardware. You simply pay as you go and enjoy a subscription-based model that's kind to your cash flow. Also, setting up an Azure account is very easy. You simply register in Azure Portal and select your required subscription and get going.
- **Less Operational Cost:** Azure has a low operational cost because it runs on its servers whose only job is to make the cloud functional and bug-free, it's usually a whole lot more reliable than your own, on-location server.
- **Cost Effective:** If we set up a server on our own, we need to hire a tech support team to monitor them and make sure things are working fine. Also, there might be a situation where the tech support team is taking too much time to solve the issue incurred in the server. So, in this regard is way too pocket-friendly.
- **Easy Back-Up and Recovery options:** Azure keeps backups of all your valuable data. In disaster situations, you can recover all your data in a single click without your business getting affected. Cloud-based backup and recovery solutions save time, avoid large up-front investments and roll up third-party expertise as part of the deal.
- **Easy to implement:** It is very easy to implement your business models in Azure. With a couple of on-click activities, you are good to go. Even there are several tutorials to make you learn and deploy faster.
- **Better Security:** Azure provides more security than local servers. Be carefree about your critical data and business applications. As it stays safe in the Azure Cloud. Even, in

natural disasters, where the resources can be harmed, Azure is a rescue. The cloud is always on.

- **Work from anywhere:** Azure gives you the freedom to work from anywhere and everywhere. It just requires a network connection and credentials. And with most serious Azure cloud services offering mobile apps, you're not restricted to which device you've got to hand.
- **Increased collaboration:** With Azure, teams can access, edit and share documents anytime, from anywhere. They can work and achieve future goals hand in hand. Another advantage of Azure is that it preserves records of activity and data. Timestamps are one example of Azure's record-keeping. Timestamps improve team collaboration by establishing transparency and increasing accountability.

What are the Various Azure Services and How does Azure Work?

Following are some of the services Microsoft Azure offers:

1. **Compute:** Includes Virtual Machines, Virtual Machine Scale Sets, Functions for serverless computing, Batch for containerized batch workloads, Service Fabric for microservices and container orchestration, and Cloud Services for building cloud-based apps and APIs.
2. **Networking:** With Azure, you can use a variety of networking tools, like the Virtual Network, which can connect to on-premise data centers; Load Balancer; Application Gateway; VPN Gateway; Azure DNS for domain hosting, Content Delivery Network, Traffic Manager, ExpressRoute dedicated private network fiber connections; and Network Watcher monitoring and diagnostics
3. **Storage:** Includes Blob, Queue, File, and Disk Storage, as well as a Data Lake Store, Backup, and Site Recovery, among others.
4. **Web + Mobile:** Creating Web + Mobile applications is very easy as it includes several services for building and deploying applications.
5. **Containers:** Azure has a property that includes Container Service, which supports Kubernetes, DC/OS or Docker Swarm, and Container Registry, as well as tools for microservices.
6. **Databases:** Azure also included several SQL-based databases and related tools.
7. **Data + Analytics:** Azure has some big data tools like HDInsight for Hadoop Spark, R Server, HBase, and Storm clusters
8. **AI + Cognitive Services:** With Azure developing applications with artificial intelligence capabilities, like the Computer Vision API, Face API, Bing Web Search, Video Indexer, and Language Understanding Intelligent.
9. **Internet of Things:** Includes IoT Hub and IoT Edge services that can be combined with a variety of machine learning, analytics, and communications services.
10. **Security + Identity:** Includes Security Center, Azure Active Directory, Key Vault, and Multi-Factor Authentication Services.

11. **Developer Tools:** Includes cloud development services like Visual Studio Team Services, Azure DevTest Labs, HockeyApp mobile app deployment and monitoring, Xamarin cross-platform mobile development, and more.

Difference between AWS (Amazon Web Services), Google Cloud, and Azure

	AWS	Google Cloud	Azure
Technology	EC2 (Elastic Compute Cloud)	Google Compute Engine(GCE)	VHD (Virtual Hard Disk)
Databases Supported	AWS fully supports relational and NoSQL databases and Big Data.	Technologies pioneered by Google, like Big Query, Big Table, and Hadoop, are databases, and Big Data,naturally fully supported.	Azure supports both relational and NoSQL through Windows AzureTable and HDInsight.
Pricing	Per hour — rounded up.	Per minute — rounded up	Per minute — rounded up.
Models	On demand, reserved spot.	On demand — sustained use.	Per minute- rounded up commitments(Pre-paid or monthly)
Difficulties	Many enterprises find it difficult to understand the company cost structure.	Fewer features and services.	Less “Enterprise-ready.
Storage Services	<ul style="list-style-type: none"> • Simple Storage Service(S3) • Elastic Block Storage. 	<ul style="list-style-type: none"> • Blob Storage • Queue Storage. • File Storage 	<ul style="list-style-type: none"> • Cloud storage. • Persistent Disk • Transfer appliance.

	AWS	Google Cloud	Azure
	<ul style="list-style-type: none"> • Elastic File storage. 	<ul style="list-style-type: none"> • Disk Storage. • Data Lake Store 	
Machine Learning	<ul style="list-style-type: none"> • Sage maker. • Lex. • polly.And many more 	<ul style="list-style-type: none"> • Machine learning • Azure Bot service • Cognitive service 	<ul style="list-style-type: none"> • Cloud speech AI • Cloud Video Intelligence. • Cloud Machine learning engine.

FAQs On Microsoft Azure

1. Microsoft Azure Student

If you are an student then Microsoft azure will provide the free access to the variety of service like.

- *compute.*
- *storage.*
- *networking.*

2. Microsoft Azure Certification

The [Microsoft Azure Fundamentals](#) certification covers various topics related to cloud computing, cloud concepts, and Azure services. The exam tests your understanding of Azure architecture, management, governance, and security. The AZ-900 certification, also known as Microsoft Azure Fundamentals, is an entry-level certification that validates foundational knowledge of cloud concepts and Microsoft Azure services. To obtain the certificate, you will need to pass the AZ-900 exam. An exam can be taken by anyone interested in gaining a basic understanding of cloud computing(Microsoft Azure), regardless of their technical background or job role. It is also useful for particular individuals who are new to Azure and want to start their journey in cloud computing and DevOps.

3. Microsoft Azure Storage Explorer

[Azure Storage Account](#) is a storage account that is a resource that acts as a container that groups all the data services from Azure storage (Azure blobs, Azure files, Azure Queues, and Azure Tables). This helps us manage all of them as a group. The policies we specify while creating the storage account or making changes after the creation applies to all the services

inside the account. Deleting a storage account deletes all the storage services deployed and the data stored inside it.

4. Microsoft Azure Storage Explorer

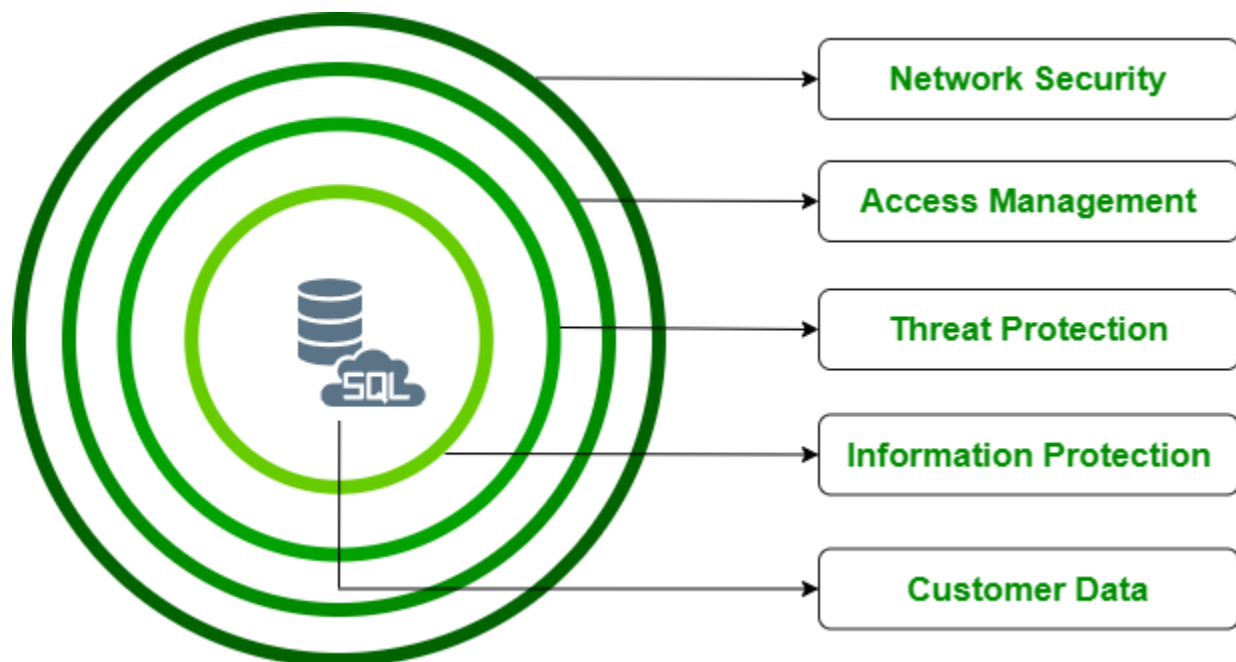
Microsoft Azure Storage Explorer is a standalone application which provide the user GUI to manage the azure storage. It supports

- [Blob Storage](#),
- [Table Storage](#),
- [Queue Storage](#), and
- [File Storage](#).

What Are Azure Data Security Features?

Data security is a huge topic and it's one of the most important types of security in space today. Lots of hackers and world governments are trying to break into databases because Data has got a lot of financial and other value.

Azure has a multi-layer view of security that you can't just have one single fence around your data. You need to have multiple layers of security that each one of those has to be violated effectively in order to get to the customer data sitting at the center.



1. Network Security: The topmost layer is Network security.

- **Block by default:** Azure SQL database has a firewall off by default. So you cannot connect to it unless the IP is white listed. You have to explicitly allow other Azure services even to connect to it. And it's not just open to the world even if you had credentials like your username and password. You need to have the firewall enabled.
- **Protect the whole server or protect individual databases:** Once you white list an IP, the database DV 1 and the data warehouse also will be able to be accessed. But we can actually go into the database and put server level firewalls as well. So there are SQL commands effectively that allow you to create firewalls to allow and block. So if you have multiple databases but you only want them to access one, you can allow the IP address through the server but block that IP on many of the databases.

You can allow or restrict other Azure Services and you can even add an endpoint into a virtual network that allows you to control traffic through the network security group through typical virtual network security protections

2. Access management: It deals with identity management and authentication effectively. There are two ways to achieve this:

- SQL authentication (username and password)
- Azure Active Directory (Azure AD)

All servers have a root username and password to the main user that we create during setup. But then we can enable the Azure Active Directory. We have to enable a root ID for that, and then we can create Azure Active Directory users that can then be granted access as well. So once you've set up the admin user then you're opening the door for other users to authenticate through Azure Active Directory and not SQL Server authentication. It allows you to manage your security centralized location instead of having SQL Server has its own authentication database. Once you are logged in with the user ID and password. You have certain levels of access:

- **Principle of The Least Privilege:** Microsoft recommends the principle of the least privilege which goes on to state that you should not give people excessive permissions. i.e Everyone should not be admin, Admin account should not be used to do your day-to-day work and Your applications should not be running in DB owner permissions. Creating the right levels, right users, and roles for the permissions can save you if there is a compromise that account doesn't have excessive privileges.
- **Role-Based Access Control (RBAC):** Azure Role-Based Access Control (Azure RBAC) helps you manage who has access to Azure 's resources, what they can do with those resources, and the areas they have access to. RBAC is another way that protects people from getting access to things they are not supposed to have access to.
- **Row Level Security:** SQL server itself has security in things such as Row-level security where you can allow users to access specific regions. It's in the same database and it's just a column filter that's going to determine whether you have access to it or not. So you can do all the way down to the data level types of authorization which is fine-grained.

3. Threat Protection: Azure Monitor is the centralized source for alerts, log files, monitoring, and things like that.

- **Advanced-Data Security:** If you sign up for advanced data security, there's a free trial for that and then it costs around \$20-\$30 a month per server. After that, you get these three cool features in terms of threat protection.
- **Data Discovery and Classification:** Data discovery and classification is pretty cool. It will actually go and examine your data to an audit effectively and determine which of your data fields are potentially and personally identifiable information subject to deep GDP restrictions subject at API. You might want to implement some security rules around personally identifiable information. So you can actually tag these columns as potentially sensitive information and again implement certain rules based on the tags on the columns.
- **Vulnerability Assessment:** If people can read people's names and addresses without a certain level of authorization, vulnerability assessment will look at your server and determine if you've got too many IP addresses enabled. If you're set to based on your firewall settings, your server if you've got roles in users that are of excessive privileges or don't have any use and it's going to tell you important security things about your setup.
- **Advanced Threat Protection:** Advanced threat protection is more like protecting against SQL injection attacks and some of those common things where we've got hackers actively trying to hack into your server guessing the password multiple times.

4. Information Security

- **Data Security – At Rest:** Data is encrypted by default in Azure- Transparent Data Encryption(TDE). Azure Controls the keys, and often the encryption is transparent to you. You can control the keys using the Azure Key Vault. If someone breaks into an Azure Data Center, they can't read your data. But if they pass network, identify, and authorization checks, then they can.
- **Data Security – In Transit:** This is an important one because the Internet is a series of connected nodes. Anyone sitting in between one server and another can read the data so it is important to encrypt it using SSL/TLS(i.e. HTTPS). All data traveling outside of Azure should travel encrypted
- **Always Encrypted:** Some Azure Database services support an "always encrypted" mode. The data is encrypted at the client and is stored in the encrypted state. Nothing can read it without the key, and only the client has the key. If the client is hacked, they can read the data.
- **Data Masking – Anonymizing:** Sometimes, you don't need access to sensitive data fields all the time. i.e You might need to know the order dollar amounts, order id, order date, the product ordered, etc. You have the ability to restrict access to customer personally identifiable information to only some accounts. Data masking returns ***** for certain fields that are needed by a query.
- **Store Data Encrypted:** You can always choose to have the application do the encryption. If you are storing passwords, hash them using a good hashing algorithm and a salt. Instead of having your password being sent across in plain text and encrypted at

the server side you can do the hashing and use salts in the proper algorithm.

5. Customer Data: SQL Database and SQL Managed Instance secure customer data by means of Transport Layer Security (TLS) encryption in motion.

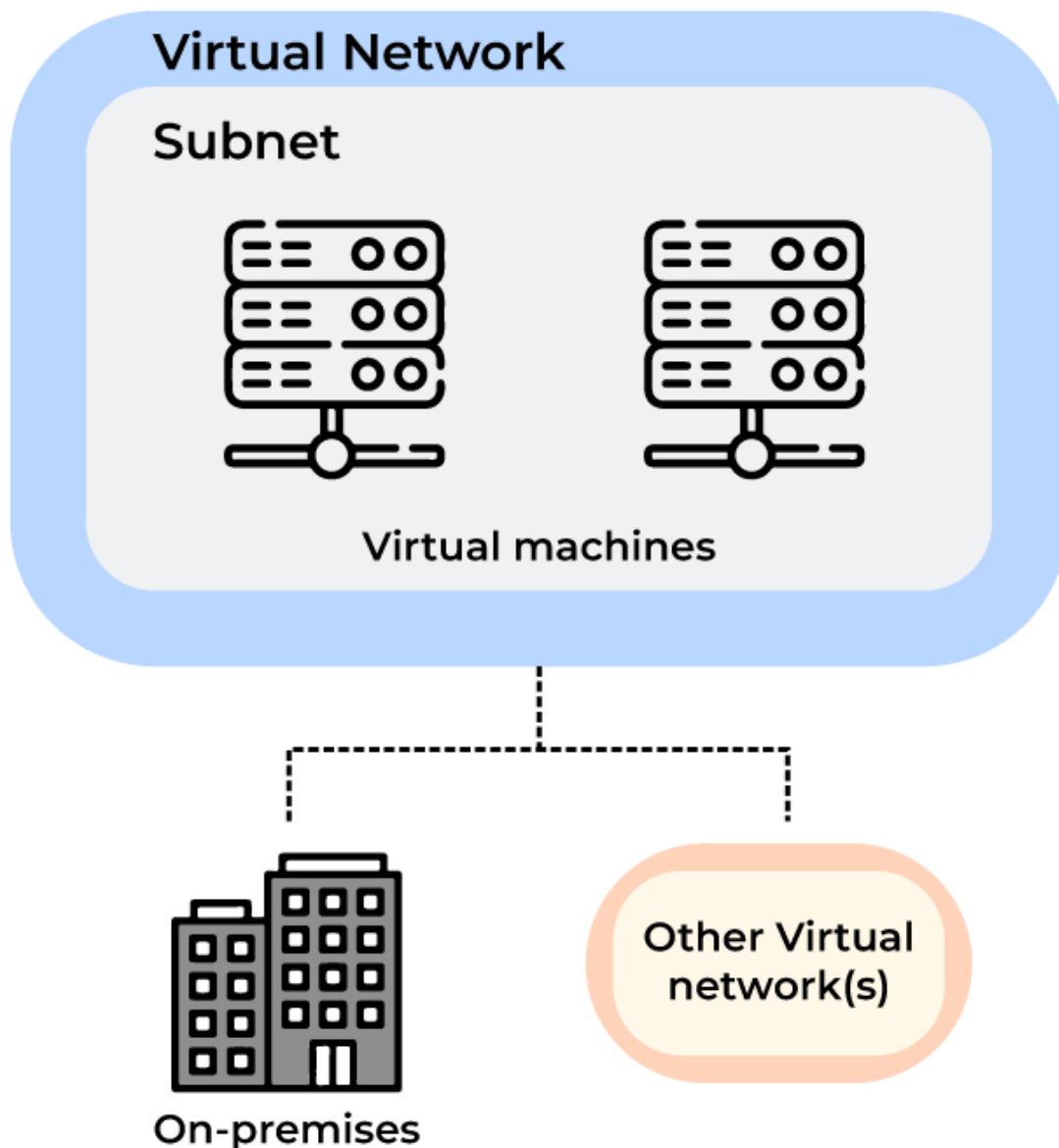
Microsoft Azure – Virtual Network

Whenever an organization moves from on-premises to a cloud-like Azure, they require the same networking functionality as they had in on-premises deployment with some level of network isolation. Azure provides different networking components to carry out these functionalities and services. One of them is the Azure virtual network (VNet).

What is Azure Virtual Network?

It is a representation of the on-premises network on the cloud. It helps us logically isolate the Azure cloud network dedicated to our subscription. It helps us to manage and provision virtual private networks in Azure, link the virtual networks with other virtual networks in Azure, or with on-premises IT infrastructure and networks that help us create hybrid or cross-premises solutions.

Each virtual network we create has its own [Classless Inter-Domain Routing \(CIDR\)](#) block and can be linked with other virtual and on-premises networks of the CIDR blocks that do not overlap. We also have control of [DNS](#) server settings and segmentation of the virtual networks into the subnets.



How Does Virtual Networking Work?

Virtual networking is used to make a connection between resources and services within the Azure cloud environment. Virtual networking is like an isolated environment. Like it can offer capabilities like private IP address space, network security groups, and routing tables apart from your on-premises network.

How Virtual Network Works in Azure:

1. **Virtual Networks:** Virtual network is a logically isolated network where we maintain our [virtual machines](#) and other resources.

2. **Subnets:** [Subnets](#) are part of a virtual network where we can group all of them based on our requirements like functionality, [security](#), and based on other requirements. With the subnets, we can control the incoming traffic to our application.
3. **Routing:** Routing is used to direct the traffic between virtual machines and resources placed in a virtual network.
4. **VPN and Express Route:** We connect to a virtual network in Azure from on-premises in two ways by using [VPN\(Virtual Private Network\)](#) for secure connection or we can use Express route for dedicated private connection.

You can build a flexible, scalable, and secure cloud architecture in Azure that is tailored to your company's needs by leveraging virtual networking.

Examples of Virtual Networking

A virtual network is a fundamental component of Azure Cloud that provides flexibility, scalability, and secure network connectivity. Some of them are mentioned below

1. **Azure Load Balancer:** The load balancer can distribute the traffic across multiple virtual machines in a virtual network. To distribute traffic to various apps or microservices, the load balancer and Azure Application Gateway can function together.
2. **Azure VPN Gateway:** A virtual device called Azure VPN Gateway offers a safe connection between your on-premises network and an Azure Virtual Network. A secure tunnel is built between the [on-premises](#) network and the Azure Virtual Network using VPN Gateway and the IPsec protocol.
3. **Azure ExpressRoute:** With the help of ExpressRoute we can establish a dedicated, private connection between on-premises infrastructure and Azure virtual network. ExpressRoute routes public traffic faster and more securely.
4. **Azure Firewall:** A virtual firewall called Azure Firewall gives resources inside an Azure Virtual Network network security. In accordance with application- and network-level policies, Azure Firewall is able to examine and filter traffic.

Usage of Virtual Networks

Virtual networks help us in many ways. Some of its use cases are as follows:

- **Create a Dedicated Private Cloud-Only Virtual Network:** This solution can be used when we don't require a hybrid configuration. When we create a virtual network, our virtual machines and services within it can communicate directly and securely with each other in the cloud. We can still configure endpoint connections for our services and virtual machines which requires [internet](#) communication.
- **Extend Our Data Center with Security:** Site-to-site virtual private networks use IPSEC to provide us with a secure connection between our corporate [virtual private network gateway](#) and Azure. We can use this to securely scale our data center's capacity.
- **Create Cross-Premises Scenarios:** Virtual networks help us securely connect cloud-based applications to any on-premises system such as [Unix systems](#) and mainframes. It gives us the flexibility to support a range of [hybrid or cross-premises solutions](#).

Subnets: Subnets help us get logical divisions within our network which help us increase performance, improve security and make it easier to manage the network. A virtual network can be segmented into one or more subnets. Each and every subnet contains a range of IP addresses that fall within the virtual network address space. This range should be unique within the address space and **not overlap** with other subnet address ranges. The address space should be specified using CIDR.

Elements of Azure Virtual Network

1. **Subnets:** Subnets can be used to split up a virtual network into smaller, simpler networks. It is possible to designate each subnet to a different security zone and control traffic between subnets using network security groups.
2. **Network Security Groups:** Network traffic between subnets or between virtual machines (VMs) inside a subnet is filtered using NSGs. The source or destination IP address, port number, or protocol can all be used to define rules in an NSG that allow or refuse communication.
3. **Virtual Network Interface Cards:** [Virtual Network Interface Cards](#) will help our VMs to other resources via the internet. Each VM can have more than one NIC.
4. **Virtual Private Network Gateway:** Virtual Private Network Gateway (VPNG) technology can link on-premises resources to cloud resources.
5. **Azure Application Gateway:** You may manage and scale web applications using the load balancer for web traffic known as Azure Application Gateway. It offers SSL offloading, URL-based routing, cookie-based session affinity, and web application firewall (WAF) features.

Creating Virtual Networks

We can create new virtual networks at any time or add virtual networks whenever we create a [virtual machine](#). We have to define the address space and at least one subnet when creating the virtual network.

By default, we can create up to **50 virtual networks per region in a subscription**. We can also increase the limit up to 500 by contacting Microsoft Azure support.

Step-By-Step Demo of Creating Azure Virtual Machine And Virtual Network

Step 1: Navigate to your Azure portal.

Step 2: Click on the [Create resource](#) option. You will be listed with different resources that can be deployed on Azure.

Step 3: Type in a **virtual network** in the search bar. Click on Create option.

Step 4: You will be prompted to fill in details about the virtual network we wish to create.

Step 5: In the **basics, tab** fill in the required details

- **Subscription:** The subscription in which you wish to create the virtual network.
- **Resource Group:** Choose the resource group where you wish to create the virtual network. You can create a new resource by clicking on create a new option.
- **Name:** Enter the name of your virtual network.
- **Region:** Choose the region for your virtual network.

Step 6: Click on **the Next: IP Addresses** button. In the IP Addresses tab enter the following details

- **IPv4 address space:** Enter the IPV4 address range in CIDR notation (e.g. 192.208.9.0/32).
- **Subnet:** Create a subnet for your virtual network. There should be at least one subnet.

Step 7: Click on **'Next ': Security tab** and choose the security settings you wish for your virtual network.

Microsoft Azure

Search resources, services, and docs (G+/)

Home > Create a resource > Virtual network >

Create virtual network ...

Basics

IP Addresses

Security

Tags

Review + create

BastionHost ⓘ

☒ Disable

☐ Enable

DDoS Protection Standard ⓘ

☒ Disable

☐ Enable

Firewall ⓘ

☒ Disable

☐ Enable

Review + create

< Previous

Next : Tags >

[Download a template for automation](#)

Step 8: Click on the **review + create** option. Wait for all the validations to pass.

Microsoft Azure

Search resources, services, and docs (G+)

Home > Create a resource > Virtual network >

Create virtual network ...

✓ Validation passed

BasicsIP AddressesSecurityTagsReview + create

Basics

Subscription

Azure

Resource group

(new) hv

Name

Region

Central India

IP addresses

Address space

10.0.0.0/16

Subnet

default (10.0.0.0/24)

Tags

None

Security

BastionHost

Disabled

Create< PreviousNext >Download a template for automation

Step 9: Click on **create** option to create your virtual network.

This is how we can create an Azure virtual network. It serves as the fundamental building block for our private network in Azure. The virtual network is similar to a traditional network that we operate in our data centers with additional benefits of cloud infrastructure like scalability, isolation, and availability.

Azure IP addressing is critical in ensuring that all the resources are accessible. Private IP addresses are used to communicate between resources in Azure while public IP addresses are used to access Azure resources directly through the internet.

Advantages of Using Azure Virtual Network

- You may build a secure, segregated network environment with Azure Virtual Network that only permits authorized access. To manage network traffic and stop illegal access, you can employ network security groups, firewall rules, and virtual private networks (VPNs).
- It is possible to link Azure Virtual Network with other Microsoft Azure services including the [load balancer](#), autoscaling, azureVPN gateway, and application gateway.
- We may modify our network in accordance with our needs thanks to Azure Virtual Network. We can adjust our IP addresses, subnetting, and routing to meet the needs of our applications.
- We can create a multi-region network that is spanned across the multi-region.
- You can scale your network resources up or down with Azure Virtual Network according to your business demands without experiencing any downtime.

Getting Familiar With IP Addressing And IP Subnetting

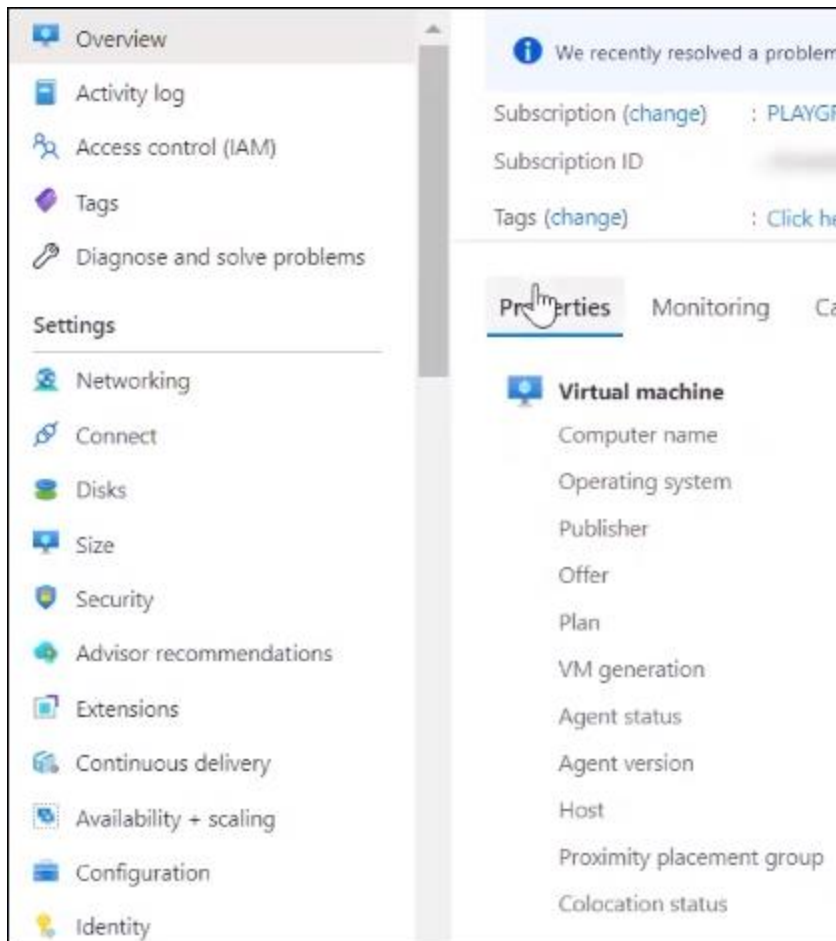
To Know more about IP addresses please refer to the [What is an IP Address?](#) and [Introduction To Subnetting](#).

Microsoft Azure – Simpler Management of Virtual Machine

In the given article we will learn how to easily configure your virtual machines to make the most out of Azure. Azure has allowed you to quickly and seamlessly manage your virtual machine right within the Azure Portal.


Implementation:


In the VM overview, the first we've done is categorize your information into different tabs. In the first tab, which is the properties tab, you can see the important properties of your virtual machine all at a glance.
















If you want to delve deeper into a particular property or change it, you can click on the section header, where you'll be brought right to the blade where you can do that.

Size

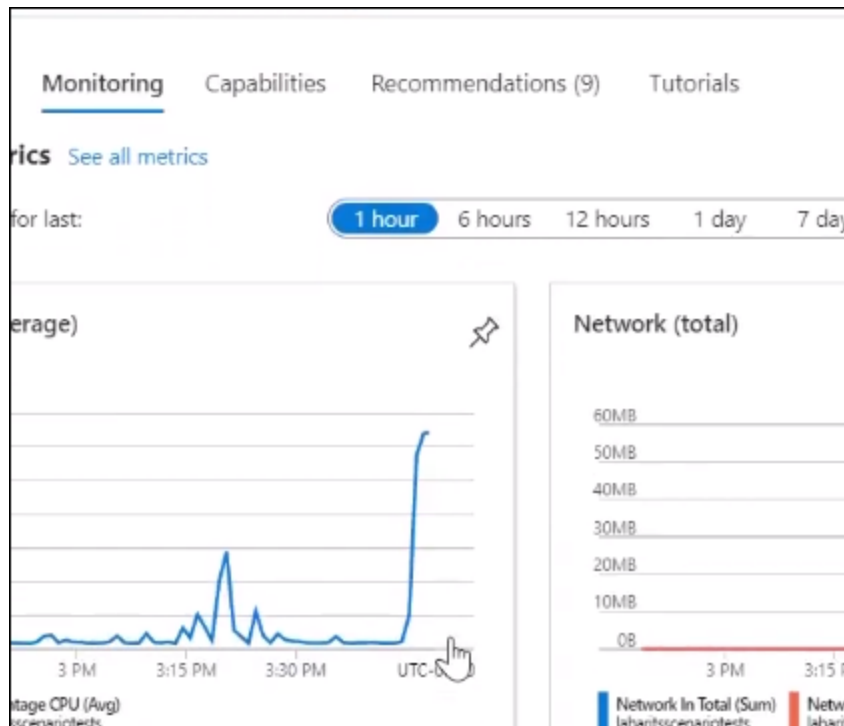
 If the virtual machine is currently running, changing its size will cause it to be restarted. Stopping the virtual machine may reveal additional sizes.

Display cost : Monthly **vCPUs : All** **RAM (GiB) : All**  Add filter

Showing 176 VM sizes. | Subscription: PLAYGROUND - IaaSExp - Team 02 | Region: South Central US | Current size: Standard

VM Size 	Family 	vCPUs 	RAM (GiB) 	Data disks 
 Most used by Azure users  The most used sizes by users in Azure				
DS1_v2  	General purpose	1	3.5	4
D2s_v3  	General purpose	2	8	4
B2s  	General purpose	2	4	4

In the monitoring tab, you get out-of-the-box, informative metrics about your host machine. You can also see this data tracked over time by clicking on the selector.




Scrolling down, you also get additional monitoring tools. With Insights, you can get even more out-of-the-box metrics that inform you about your virtual machine's health and performance.

Additional Azure Monitor tools

**Monitor at scale with Insights**

Get visibility into the resource's performance and health, accounting for different processes and interconnected dependencies.

By setting up alerts, you can quickly become aware of health, usage, and cost issues.

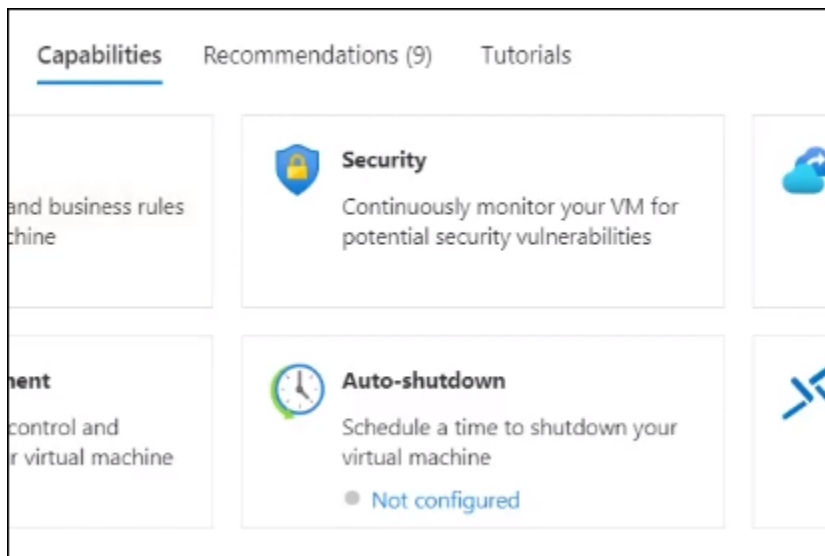
**Get alerted to issues**

Create alerts to monitor resource health, usage, cost and more.

And with Log Analytics, you can query data collected by Azure Monitor logs and perform your own data analysis.



You can access all these functionalities by clicking on the respective cards. In the capabilities tab, you get an overview of some of the most used services on Azure, can see the respective onboarding status and also get a simplified onboarding process all within the tab.



For example, with Azure Policy, you see that you have already configured policies. You click on it, and you can see a contact screen pops up offering a simplified onboarding process for assigning recommended policies. You can choose one of the other recommended policies, click assign, and quickly start onboarding more to the service without having to leave this tab.

Assign recommended policies

These policies assist with the adoption of industry best practices.

Each selected policy creates an assignment specifically for this virtual machine. Use the Policy page to update or remove created assignments. Grayed-out policies are already assigned to a scope that includes this virtual machine.

[Learn more about the recommended policies.](#)

POLICY NAME
<input checked="" type="checkbox"/> Audit virtual machines without disaster recovery configured
<input type="checkbox"/> Azure Backup should be enabled for Virtual Machines
<input type="checkbox"/> Audit VMs that do not use managed disks

If you want more granular control and more advanced configurations, click on the card itself, where you'll be brought to the experience.

Azure is working with the different teams that you see here to make sure that in the future, all these services can have this simplified onboarding process as well. In the Azure Advisor Recommendations tab, you can get proactive, actionable, and personalized best practices from Azure.

Recommendations (9)
Tutorials

Recommendations to reduce costs, increase security, optimize performance and reliability and achieve operational excellence

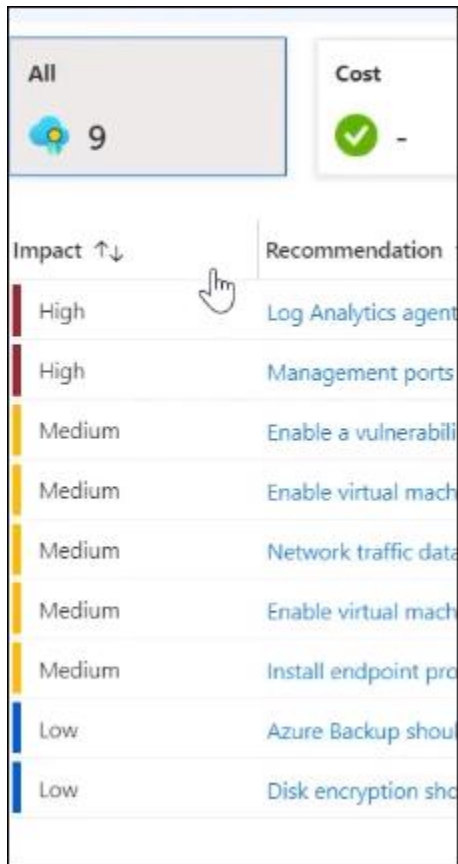
Security
7

Reliability
2

Performance
-

↑↓	Category ↑↓	Impacted resource
Just-in-time VM protection should be installed on your virtual machine	Security	1 Virtual machine
Just-in-time VM protection of virtual machines should be protected with just-in-time ne...	Security	1 Virtual machine
Security assessment solution on virtual machines	Security	1 Virtual machine
Configure backup to protect your data from corruption and accide...	Reliability	1 Virtual machine

The recommendations are also sorted by impact, so you can prioritize what you want to act on first. With this tab, you can easily see how to improve the performance, security, reliability, and cost of the virtual machines and make sure you're following Azure's best practices.



The screenshot shows the Azure Recommendations interface. At the top, there are two tabs: 'All' (selected) and 'Cost'. The 'All' tab shows a blue lightbulb icon and the number '9'. The 'Cost' tab shows a green checkmark icon and a minus sign. Below the tabs, there is a table with two columns: 'Impact' and 'Recommendation'. The 'Impact' column has a sort arrow and a list of impact levels: High, High, Medium, Medium, Medium, Medium, Medium, Low, and Low. The 'Recommendation' column lists the corresponding actions: Log Analytics agent, Management ports, Enable a vulnerability, Enable virtual machine, Network traffic data, Enable virtual machine, Install endpoint protection, Azure Backup should, and Disk encryption should. A hand cursor is pointing at the 'High' impact level in the first row.

Impact ↑↓	Recommendation
High	Log Analytics agent
High	Management ports
Medium	Enable a vulnerability
Medium	Enable virtual machine
Medium	Network traffic data
Medium	Enable virtual machine
Medium	Install endpoint protection
Low	Azure Backup should
Low	Disk encryption should

Lastly, in the tutorials tab, you have quick access to free training from Microsoft and video tutorials to help you get started with the virtual machine experience and also make the most out of it. This is especially helpful for customers who are new to virtual machines, new to Azure, or even new to the cloud. With these new additions, we make it easier than ever to easily and quickly get started on managing your virtual machine in Azure.

Tutorials

Application with the
re Linux VM

MEAN-based web
Azure Linux virtual



Build a scalable application with virtual machine scale sets

8 units • 57 min

Enable your application to automatically adjust to changes in load while minimizing costs with virtual machine scale sets.

[Start](#)



Configure network for your virtual machines

8 units • 1 hr 34 min

Learn how to configure networking in a secure way for your Azure virtual machines.

[Start](#)