



# **BLACKBUCKS INTERNSHIPREPORT**

## **On Demand Based Scaling with Launch Template and Launch Configuration**

**Mr. AKULA DILEEP**

**RegdNo:21B91A0510**

**Mr. ALAPATI KARUNA KUMAR**

**RegdNo:21B91A0512**

**Mr. SHAIK FAYAZ**

**RegdNo:21B91A05S6**

## **UNDER THE GUIDANCE OF MR. AASHU DEV**

B.Tech, AWS solution Architect (2x)

certified,AWS Academy

Accredited Educator



**Blackbuck Engineers Pvt,Ltd**

**Road No 36, Jubilee Hills, Hyderabad**

# **BLACKBUCK INTERNSHIP WORK**

## **Team Members:**

- AKULA DILEEP (21B91A0510)
- ALAPATI KARUNA KUMAR (21B91A0512)
- SHAIK FAYAZ (21B91A05S6)

## **Title:**

On Demand Based Scaling with Launch Template and Launch Configuration .

## **Abstract:**

On-demand based scaling with launch templates and launch configurations is a feature offered by cloud computing providers like Amazon Web Services (AWS) to automatically adjust the number of instances in an auto-scaling group based on demand.

Here's how it works:

**Launch Template:** A launch template is a configuration template that defines the various parameters required to launch instances in the cloud environment. This includes specifications like the Amazon Machine Image (AMI), instance type, security groups, key pairs, and other instance launch settings.

**Launch Configuration:** A launch configuration is a deprecated way of defining the configuration for instances in an auto-scaling group. It served a similar purpose as a launch template but has been replaced by launch templates. However, some older systems might still be using launch configurations.

## **Table of Contents**

1. Introduction to Amazon web services (AWS)	1-2
2. Why AWS	2-5
3. AWS global Infrastructure	5-8
4. List of top AWS services	8-18
5. Implementation of the Architecture of Bastion Host with Auto Scaling	19
5.1. Introduction	19-20
5.2. AWS services used	20
5.3. Rough Architecture	20
5.4. Final Architecture	21
6. Implementation of the project	22
6.1. Service 1: VPC	22-23
6.2. Service 2: EC2	23-27
6.3. Service 3: AMI	28-30
6.4. Service 4: Launching a Template	30-34
6.5. Service 5: Creating an auto scaling group	34-49
6.6. Conclusion	50

## **Introduction To Amazon Web Services (AWS):**

- **Amazon Web Services, Inc. (AWS)** is a subsidiary of Amazon that provides on-demand cloud computing platforms and APIs to individuals, companies, and governments, on a metered, pay-as-you-go basis. Oftentimes, clients will use this in combination with autoscaling (a process that allows a client to use more computing in times of high application usage, and then scale down to reduce costs when there is less traffic). These cloud computing web services provide various services related to networking, compute, storage, middleware, IoT and other processing capacity, as well as software tools via AWS server farms. This frees clients from managing, scaling, and patching hardware and operating systems. One of the foundational services is Amazon Elastic Compute Cloud (EC2), which allows users to have at their disposal a virtual cluster of computers, with extremely high availability, which can be interacted with over the internet via REST APIs, a CLI or the AWS console. AWS's virtual computers emulate most of the attributes of a real computer, including hardware central processing units (CPUs) and graphics processing units (GPUs) for processing; local/RAM memory; hard-disk/SSD storage; a choice of operating systems; networking; and pre-loaded application software such as web servers, databases, and customer relationship management (CRM).
- AWS services are delivered to customers via a network of AWS server farms located throughout the world. Fees are based on a combination of usage (known as a "Pay-as-you-go" model), hardware, operating system, software, or networking features chosen by the subscriber required availability, redundancy, security, and service options. Subscribers can pay for a single virtual AWS computer, a dedicated physical computer, or clusters of either. Amazon provides select portions of security for subscribers (e.g., physical security of the data centers) while other aspects of security are the responsibility of the subscriber (e.g., account management, vulnerability scanning, patching). AWS operates in many global geographical regions including seven in North America.

- Amazon markets AWS to subscribers as a way of obtaining large-scale computing capacity more quickly and cheaply than building an actual physical server farm. All services are billed based on usage, but each service measures usage in varying ways. As of 2021 Q4, AWS has 33% market share for cloud infrastructure while the next two competitors Microsoft Azure and Google Cloud have 21%, and 10% respectively, according to Synergy Group.

Uses of AWS:

- A small manufacturing organization uses their expertise to expand their business by leaving their IT management to the AWS.
- A large enterprise spread across the globe can utilize the AWS to deliver the training to the distributed workforce.
- An architecture consulting company can use AWS to get the high compute rendering of construction prototype.
- A media company can use the AWS to provide different types of content such as ebox or audio files to the worldwide files.

## Why AWS?

There are several reasons why AWS has become a popular choice for cloud computing:

1. Broad and Comprehensive Service Offering: AWS offers a wide range of services to meet various computing needs. Whether you require computer power, storage, databases, machine learning, analytics, networking, or other capabilities, AWS provides a comprehensive set of services to fulfill these requirements.
2. Scalability and Flexibility: AWS allows users to scale their resources up or down based on demand. Whether you need to handle a sudden surge in traffic or want to reduce costs during

periods of lower activity, AWS provides the flexibility to adjust your resources accordingly. This scalability ensures that your applications can handle varying workloads effectively.

3. Global Infrastructure: AWS has a vast global infrastructure comprising numerous data centers and availability zones spread across different regions. This infrastructure enables users to deploy their applications and services closer to their target audience, resulting in reduced latency and improved performance.

4. Reliability and Availability: AWS has built a reputation for providing highly reliable and available services. With its multiple availability zones and data replication mechanisms, AWS ensures that your applications and data remain accessible even in the face of hardware failures or natural disasters. Service Level Agreements (SLAs) guarantee a certain level of uptime for many AWS services.

5. Security: AWS places a strong emphasis on security. It provides a wide range of security features and tools to help users protect their applications and data. This includes encryption options, network security controls, identity and access management, and compliance certifications. AWS adheres to industry best practices to maintain a secure environment for its customers.

6. Integration and Ecosystem: AWS integrates well with various third-party tools, technologies, and services. It offers extensive APIs and SDKs, making it easier to integrate AWS services into existing applications or build new solutions from scratch. The AWS ecosystem also includes a vibrant community, documentation, training resources, and support services, facilitating development and troubleshooting.

7. Cost-Effectiveness: AWS follows a pay-as-you-go pricing model, allowing users to pay only for the resources they consume. This eliminates the need for upfront investments in hardware and infrastructure. Additionally, AWS provides cost optimization tools and features to help users monitor and control their spending, ensuring cost-effectiveness.

8. Innovation and Continuous Improvement: AWS continues to innovate and expand its services, introducing new capabilities and features regularly. It invests heavily in research and development to stay at the forefront of cloud technology. This commitment to innovation ensures that users have access to the latest tools and advancements in cloud computing

These factors, among others, contribute to the popularity and success of AWS as a cloud computing provider. However, it's important to note that the choice of cloud provider should be based on your specific needs, requirements, and preferences. It's worth evaluating multiple cloud platforms to determine the best fit for your organization.

### **Advantages of AWS:**

1. Flexibility
2. Cost-effectiveness
3. Scalability/Elasticity
4. Security

#### **1) Flexibility**

- We can get more time for core business tasks due to the instant availability of new features and services in AWS.
- It provides effortless hosting of legacy applications. AWS does not require learning new technologies and migration of applications to the AWS provides advanced computing and efficient storage.
- AWS also offers a choice that whether we want to run the applications and services together or not. We can also choose to run a part of the IT infrastructure in AWS and the remaining part in data centers.

#### *2) Cost-effectiveness*

AWS requires no upfront investment, long-term commitment, and minimum expense when compared to traditional IT infrastructure that requires a huge investment.

#### **3) Scalability/Elasticity**

Through AWS, autoscaling and elastic load balancing techniques are automatically scaled up or down, when demand increases or decreases respectively. AWS techniques are ideal for handling

unpredictable or very high loads. Due to this reason, organizations enjoy the benefits of reduced cost and increased user satisfaction.

#### 4) *Security*

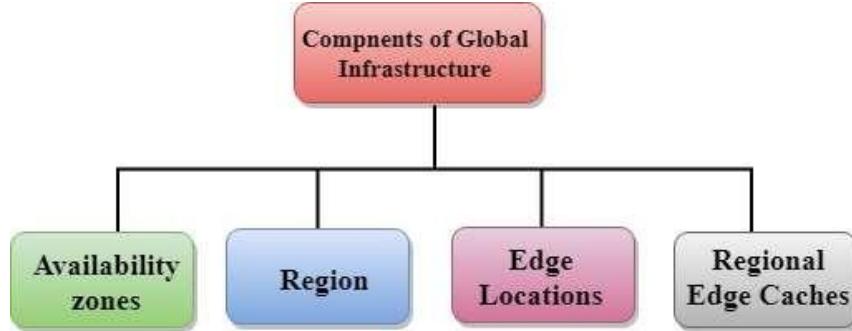
- AWS provides end-to-end security and privacy to customers.
- AWS has a virtual infrastructure that offers optimum availability while managing full privacy and isolation of their operations.
- Customers can expect high-level physical security because of Amazon's several years of experience in designing, developing, and maintaining large-scale IT operation centers.
- AWS ensures the three aspects of security, i.e., Confidentiality, integrity, and availability of users.

## AWS Global Infrastructure

- AWS is a cloud computing platform which is globally available.
- Global infrastructure is a region around the world in which AWS is based. Global infrastructure is a bunch of high-level IT services which is shown below:
- AWS is available in 19 regions, and 57 availability zones in December 2018 and 5 more regions 15 more availability zones for 2019.

The following are the components that make up the AWS infrastructure:

- Availability Zones
- Region
- Edge locations
- Regional Edge Caches



In Amazon Web Services (AWS), an Availability Zone (AZ) refers to a distinct, physically separate data center within a specific region. AZs are designed to provide fault tolerance and high availability by isolating failures and minimizing the impact of any disruption.

Each AZ is equipped with independent power, cooling, networking infrastructure, and is connected to other AZs within the same region through high-speed, low-latency links. They are strategically located to minimize the risk of natural disasters affecting multiple zones simultaneously.

By distributing resources across multiple AZs, you can design highly reliable and resilient architectures in AWS. When you launch resources like EC2 instances, databases, or storage volumes, you have the option to select the AZ in which they should be provisioned.

The primary benefits of utilizing Availability Zones in AWS include:

- Fault tolerance:** By deploying resources in different AZs, you protect your applications from single points of failure. If one AZ experiences an issue, your applications can continue running in other AZs, ensuring minimal downtime.

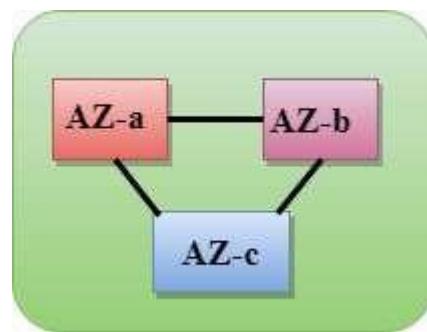
**High availability:** Distributing resources across AZs allows you to achieve high availability for your applications. Services like load balancers can be configured to route traffic across multiple AZs, automatically diverting traffic to healthy instances if one AZ becomes unavailable.

## Availability zone as a Data Center

- An availability zone is a facility that can be somewhere in a country or in a city. Inside this facility, i.e., Data Centre, we can have multiple servers, switches, load balancing, firewalls. The things which interact with the cloud sit inside the data centers.
- An availability zone can be a several data centers, but if they are close together, they are counted as 1 availability zone.

## Region

- A region is a geographical area. Each region consists of 2 more availability zones.
- A region is a collection of data centers which are completely isolated from other regions.
- A region consists of more than two availability zones connected to each other through links.



- Availability zones are connected through redundant and isolated metro fibers.

## Edge Locations

- Edge locations are the endpoints for AWS used for caching content.
- Edge locations consist of CloudFront and Amazon's Content Delivery Network (CDN).
- Edge locations are more than regions. Currently, there are over 150 edge locations.
- Edge location is not a region but a small location that AWS have. It is used for caching the content.
- Edge locations are mainly located in most of the major cities to distribute the content to end users with reduced latency.
- For example, some user accesses your website from Singapore; then this request would be redirected to the edge location closest to Singapore where cached data can be read.

### Regional Edge Cache

- AWS announced a new type of edge location in November 2016, known as a Regional Edge Cache.
- Regional Edge cache lies between CloudFront Origin servers and the edge locations.
- A regional edge cache has a larger cache than an individual edge location.
- Data is removed from the cache at the edge location while the data is retained at the Regional Edge Caches.
- When the user requests the data, then data is no longer available at the edge location. Therefore, the edge location retrieves the cached data from the regional edge cache instead of the Origin servers that have high latency.

## List of top AWS Services:

AWS is the widely used cloud platform worldwide, from start-ups to large enterprises. Though AWS services were introduced to the market by 2006, their revenue from Public Cloud SaaS has hit 145.5 billion USD by 2021. Presently, Amazon Web Services are a one-stop solution for all cloud services ranging from data storage to analytics. AWS services provide easy, simple, cost-effective cloud services, which drive businesses to achieve increased efficiency and performance. Besides, these services have many more features to serve customers in multiple ways.

Now, let's have a look at the most popular AWS services in 2023. In this blog, you can learn what is the objective, features, and benefits of each AWS service.

Here is the list of Top 30 AWS Services List:

### 1. Amazon EC2 [Elastic Compute Cloud]

Amazon EC2 is one of the fastest-growing cloud computing AWS services, which offers virtual servers to manage any kind of workload. It facilitates the computing infrastructure with the best suitable processors, networking facilities, and storage systems. As a result, it supports adapting to the workloads precisely. Amazon EC2 provides a highly secure, reliable, performing computing infrastructure meeting business demands. And it helps you to access resources quickly and dynamically scale capacities as per demands.



## 2. Amazon S3

Another popular addition to the AWS services list is Amazon S3, which is an object storage AWS service, which is highly scalable. It mainly helps users to access any quantity of data from anywhere. Here, data is stored in ‘storage classes’ to reduce costs without any extra investment and manage it comfortably. The data is highly secure and supports meeting audit and compliance requirements. You can handle any volume of data with Amazon S3’s robust access controls, replication tools, and higher visibility. Moreover, it supports maintaining data version controls and preventing accidental deletion.



## 3. AWS Aurora

Amazon Aurora is the next addition to this list of top AWS services in demand. Why? It is a MySQL and PostgreSQL compatible relational database with high performance. Believe it or not, it is five times faster than standard MySQL databases. And it allows for automating crucial tasks such as hardware provisioning, database setup and backups, and patching. Amazon Aurora is a distributed, fault-tolerant, self-healing storage system that could scale automatically as per needs. Besides, you can even reduce costs significantly and enhance databases' security, availability, and reliability.



## 4. Amazon DynamoDB

DynamoDB is a promising addition to this list of AWS services. DynamoDB is a fully managed and serverless NoSQL database AWS service. And it is a fast and flexible database system that provides innovative opportunities to developers at low costs. It gives you single-digit millisecond performance with unlimited throughput and storage. DynamoDB has in-built tools to generate actionable insights, useful analytics, and monitor traffic trends in applications.



## 5. Amazon RDS

Amazon RDS would be the next entry in this discussion on AWS services. Amazon RDS is the managed Relational Database AWS Service (RDS) for MySQL, PostgreSQL, Oracle, SQL Server, and MariaDB. It allows the setup, operation, and scale of a relational database in the cloud quickly. Also, it achieves high performance by automating the tasks such as hardware provisioning, database setup, patching, and backups. When you use Amazon RDS, you don't need to install and maintain the database software. Overall, you can optimize costs by embracing this service and achieve high availability, security, and compatibility for your resources.



## 6. Amazon Lambda

AWS Lambda is also a promising addition to the list of AWS services. Amazon Lambda is a serverless and event-driven computing AWS service. It helps to run codes automatically without worrying about servers and clusters. Simply put, codes can be uploaded directly to run without worrying about provisioning or managing infrastructure. So, this service automatically accepts 'code execution requests' irrespective of its scale. Besides, you can pay the price only for the computer time, so AWS Lambda makes effective cost-control.



## 7. Amazon VPC

Amazon VPC is the Virtual Private Cloud, which is an isolated cloud resource. It controls the virtual networking environment, such as resource placement, connectivity, and security. And it allows you to build and manage compatible VPC networks across cloud AWS resources and on-premise resources. Here, it improves security by applying rules for inbound and outbound connections. Also, it monitors VPC flow logs delivered to Amazon S3 and Amazon Cloudwatch to gain visibility over network dependencies and traffic patterns. Amazon VPC also detects anomalies in the patterns, prevents data leakage, and troubleshoots network connectivity and configuration issues.



## 8. Amazon CloudFront

Amazon CloudFront is another credible mention in the list of renowned Amazon Web Services. This AWS service delivers content globally, which offers high performance and security.

Mainly, it delivers data with high speed and low latency. Here, content is delivered to destinations successfully with automated network mapping and intelligent routing mechanisms. The security of data is enhanced with traffic encryption methods and access controls. Also, data can be transferred within milliseconds with its in-built data compression, edge computing capabilities, and field-level encryption. Besides, you gear up streaming high-quality video using AWS media services to any device quickly and consistently using Amazon CloudFront.



## 9. AWS Elastic Beanstalk

This AWS service supports running and managing web applications. Elastic Beanstalk allows for the easy deployment of applications from capacity provisioning, load balancing, and auto-scaling to application health monitoring. With its auto-scaling properties, this service simplifies demands in scaling to adjust to the needs of the business. It helps to manage peaks in workloads and traffic with minimum costs. Basically, AWS Elastic Beanstalk is a developer-friendly tool since it manages servers, load balancers, firewalls, and networks simply. As a result, this service allows developers to show much more focus on coding.



## 10. Amazon EC2 Auto-scaling

This AWS service scales computing capacity to meet the demands accurately. And it is achieved by adding or removing EC2 instances automatically. There are two types of scaling such as dynamic scaling and predictive scaling. Here, dynamic scaling responds to the presently changing demands, whereas predictive scaling responds based on predictions. Through Amazon EC2 Auto-scaling, you can identify the unhealthy EC2 instances, terminate them, and replace them with new instances.



## 11. Amazon ElastiCache

Amazon ElastiCache is a fully managed, flexible, in-memory caching AWS service. It supports increasing the performance of your applications and database. And this service helps to reduce the load in a database by caching data in memory. Amazon ElastiCache accesses data from in-memory with high speed, microsecond latency, and high throughput. With a self-managed cache service, you can reduce costs and eliminate the operational overhead of your business.



## 12. Amazon S3 Glacier

Amazon S3 Glacier is the archive storage in the cloud at a low cost. It is built with three storage classes such as S3 Glacier instant retrieval, flexible retrieval, and deep archive. Here, the instant class supports immediate access to data, and the flexible class allows flexible access within minutes to hours with no cost. The third one, deep archive, helps archive compliance data and digital media. Overall, they support you to access data from archives faster.



## 13. Amazon LightSail

Amazon LightSail is the website and applications building AWS service. This service offers Virtual Private Server instances, containers, databases, and storage. It allows a serverless computing service with AWS Lambda. With Amazon LightSail, you can create websites using pre-configured applications such as WordPress, Magento, Prestashop, and Joomla in a few clicks and at a low cost. In addition to this, it is the best tool for testing, so you can create, test, and delete sandboxes with your new ideas.



#### 14. Amazon Sagemaker

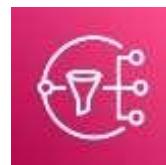
Amazon Sagemaker is the AWS service that allows building, training, and deploying Machine Learning (ML) models at a large capacity. It is an analytical tool that functions based on Machine Learning power to analyze data more efficiently. With its single toolset, you can build high-quality ML models quickly. Amazon Sagemaker not only generates reports but provides the purpose for generating predictions too. In addition, Amazon Ground Truth Plus creates datasets without labeling applications.



#### 15. Amazon SNS

It is the Amazon Simple Notification Service (SNS). It is a messaging service between Application to Application (A2P) and Application to Person (A2Person). Here, A2P helps many-to-many messaging between distributed systems, microservices, and event-driven serverless applications. And A2P supports applications to send messages to many users via mail, SMS, etc. For instance, you can send up to ten messages in a single API request. With effective filtering systems, subscribers will receive messages that they are interested in.

Besides, Amazon SNS works alongside Amazon SQS to deliver messages accurately and consistently.



#### 16. Amazon EBS

Amazon Elastic Block Store (EBS) is a block storage service. It supports scaling high-performance workloads such as SAP, Oracle, and Microsoft products. And it provides better

protection against failures up to 99.999%. It helps to resize clusters for big data analytics engines such as Hadoop and Spark. Also, you can build storage volumes, optimize storage performance, and reduce costs. Amazon EBS's lifecycle management creates policies that help create and manage backups effectively.



## 17. Amazon Kinesis

It is the AWS service that analyses video as well as data streams. Amazon Kinesis collects, processes, and analyzes all types of streaming data. Here, the data may be audio, video, application logs, website clickstreams, and IoT telemetry. Then, it generates real-time insights within seconds once the data has arrived. With the help of Amazon Kinesis, you could stream and process a large quantity of real-time data with low latencies, very simply.



## 18. Amazon Elastic File System (EFS)

Amazon EFS is the fully managed file system for Amazon EC2. And it is a simple and serverless elastic file system. You can create and configure file systems without provisioning, deploying, patching, and maintenance using Amazon EFS. Here, files can be added and deleted as per the scaling needs. Especially, you can pay only for the used space, hence this service helps to reduce costs.



## 19. AWS IAM

It is the Identity and Access Management (IAM) service offered by AWS to securely access the applications and resources. It regulates access to various resources based on roles and access policies; as a result, you can achieve a fine-grained access control on your resources. The AWS IAM access analyzer helps streamline permission management through setting, verifying, and refining. In addition, AWS IAM attribute-based access control helps create fine-grained permissions based on user attributes such as department, job role, team name, etc.



## 20. Amazon SQS

Amazon SQS is a fully managed message queuing service. There are two types of message queuing services: SQS Standard and SQS FIFO. Here, the SQS standard offers features such as maximum throughput, best-effort ordering, and quick delivery. And SQS FIFO processes messages only once in the same order by which they have been sent. Also, Amazon SQS allows decoupling or scaling microservices, distributed systems, and serverless applications. It helps you send, receive, and manage messages in a large volume. Moreover, there is no need to install and maintain other messaging software, reducing costs significantly. Besides, scaling is carried out quickly and automatically in this service.



## 21. Amazon RedShift

Amazon Redshift is a quick, simple, and cost-effective data warehousing service. You can gain insights about cloud data warehousing in an easy, faster, and more secure way. It allows analysis of all the data in operational databases, data lakes, data warehouses, and third-party data. And Amazon Redshift helps analyze a large volume of data and run complex analytical queries. With its automation capabilities, this service increases query speed and provides the best price performance.



## 22. Amazon Cloudwatch

This AWS service monitors the cloud resources and applications keenly. It is a single platform that helps to monitor all AWS resources and applications; it increases visibility to respond to issues quickly. Mainly, Amazon Cloudwatch provides actionable insights to optimize monitoring applications, systemwide performance changes, and resource utilization. And you can get a complete view of the health of AWS resources, applications, and services running on AWS and on-premises. In addition, Amazon CloudWatch helps to detect anomalies in the behavior of the cloud environment, set alarms, visualize logs and metrics, make automated actions, troubleshoot issues, and discover insights.



## 23. Amazon Chime

Amazon Chime is a communication service. It is a single solution that offers audio calling, video calling, and screen sharing capabilities. With the help of this service, you can make quality meetings, chat, and video calls both inside and outside of your organization. And more features can be added to this service as per your business needs. Mainly, you can set calls for a pre-defined time to automatically make calls on time. Amazon Chime helps you not to miss a meeting amidst your hectic schedule at work. Besides, you can pay as per the usage of resources by which you can reduce the costs significantly.



## 24. Amazon Cognito

It is the identity management AWS service. Amazon Cognito manages identities for accessing your applications and resources. Mainly, this service helps to add sign-in, sign-up, and access control the web and mobile apps quickly. It can support millions of users to sign in with familiar applications such as Apple, Facebook, Google, and Amazon. In Amazon Cognito, the feature ‘Cognito user pools’ can be set up quickly without any infrastructure, and the pool members will have a directory profile. It supports multi-factor authentication and encryption of data-at-rest and data-in-transit.



## 25. Amazon Inspector

Amazon Inspector is an automated vulnerability management service. This service offers continuous and automated vulnerability management for Amazon EC2 and Amazon ECR. It allows scanning AWS workloads for software vulnerabilities and unwanted network exposure. Amazon Inspector quickly identifies vulnerabilities, which helps to take immediate actions to resolve them before they worsen. Moreover, it supports meeting compliance requirements and reduces mean-time-to-remediate vulnerabilities. And it provides you with accurate risk scores and streamlined workflow.



## 26. AWS Firewall Manager

It is the central management service of firewall rules. The firewall manager supports managing firewall rules across all the applications and accounts. The common security rules help to manage new applications included over time. It is the one-time solution for consistently creating firewall rules and security policies and implementing them across the infrastructure. AWS firewall manager helps you audit VPC security groups for compliance requirements and control network traffic effectively.



## 27. Amazon Appflow

Amazon Appflow is a no-code service that allows the integration of SaaS applications and AWS services effortlessly. To be more precise, it securely automates dataflows integrating third-party applications and AWS services without using codes. You can transfer data between SaaS applications such as Salesforce, SAP, Zendesk, etc. since

Amazon Appflow can be integrated with other applications in a few clicks. Especially, a large volume of data can be moved without breaking it up into batches using this service.



## 28. Amazon Route 53

It is a scalable cloud Domain Name System (DNS) service. It allows end-users to connect with Amazon EC2, Elastic load balancers, Amazon S3 buckets, and even outside AWS. In this service, the feature 'Route 53 application recovery controllers' configure DNS health checks and helps to monitor the ability of systems to recover from failures. And 'Route 53 traffic flow' helps manage traffic across the globe using routing methods such as latency-based routing, Geo DNS, Geoproximity, and weighted round-robin.



## 29. AWS Cloud Formation

This AWS service creates and manages resources with templates. It is a single platform that can handle all AWS accounts across the globe. It automates resource management with AWS service integration and offers turnkey application distribution and governance controls. Also, AWS Cloud Formation can automate, test, and deploy infrastructure with continuous integration and delivery. And you can run applications right from AWS EC2 to complex multi-region applications using this service.



## 30. AWS Key Management Service (KMS)

AWS KMS manages the creation and control of encryption keys. It means that AWS KMS creates cryptographic keys and controls their uses across various applications. You can achieve a secure and resilient service using hardware resilient modules to protect keys. This service can be integrated with AWS CloudTrail to provide logs of all key usage to precisely fulfill compliance and regulatory requirements.



# Implementation of the Architecture of Bastion Host with Auto Scaling

## Introduction:

Amazon EC2 Auto Scaling allows you to automatically scale your EC2 instances up or down based on demand. You can use a launch template or a launch configuration to specify the configuration of your EC2 instances.

This project will explore the architecture of Autoscaling with launch templates and launch configurations. We will discuss the advantages and disadvantages of each approach, and we will create a sample application that uses Autoscaling with a launch template.

The goal of this project is to understand the different ways to use Autoscaling, and to choose the right approach for your specific needs.

## Auto Scaling Group:

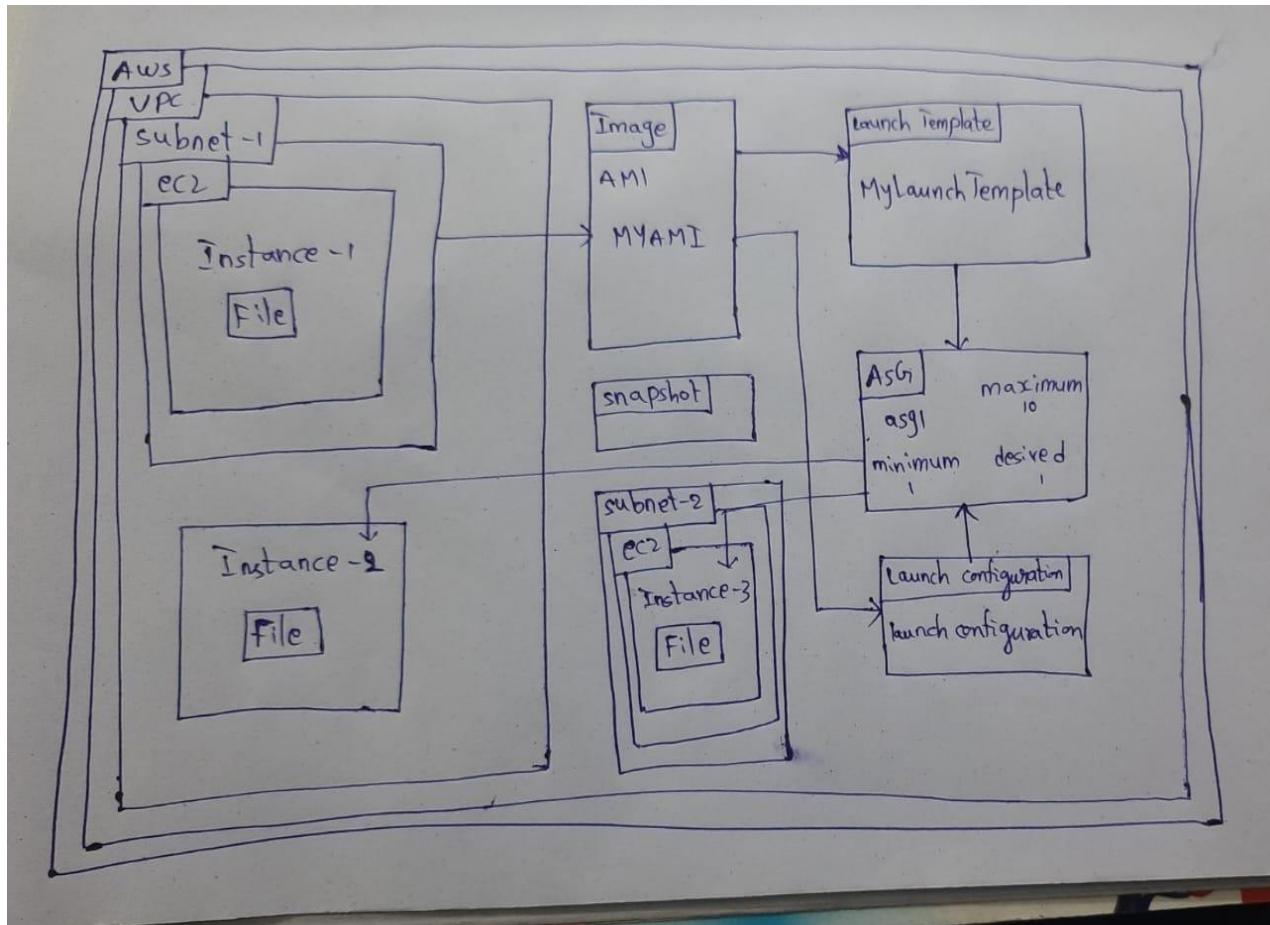
An Auto Scaling group contains a collection of EC2 instances that are treated as a logical grouping for the purposes of automatic scaling and management. An Auto Scaling group also lets you use Amazon EC2 Auto Scaling features such as health check replacements and scaling policies. Both maintaining the number of instances in an Auto Scaling group and automatic scaling are the core functionality of the Amazon EC2 Auto Scaling service. The size of an Auto Scaling group depends on the number of instances that you set as the desired capacity. You can adjust its size to meet demand, either manually or by using automatic scaling. An Auto Scaling group starts by launching enough instances to meet its desired capacity.

It maintains this number of instances by performing periodic health checks on the instances in the group. The Auto Scaling group continues to maintain a fixed number of instances even if an instance becomes unhealthy. If an instance becomes unhealthy, the group terminates the unhealthy instance and launches another instance to replace it.

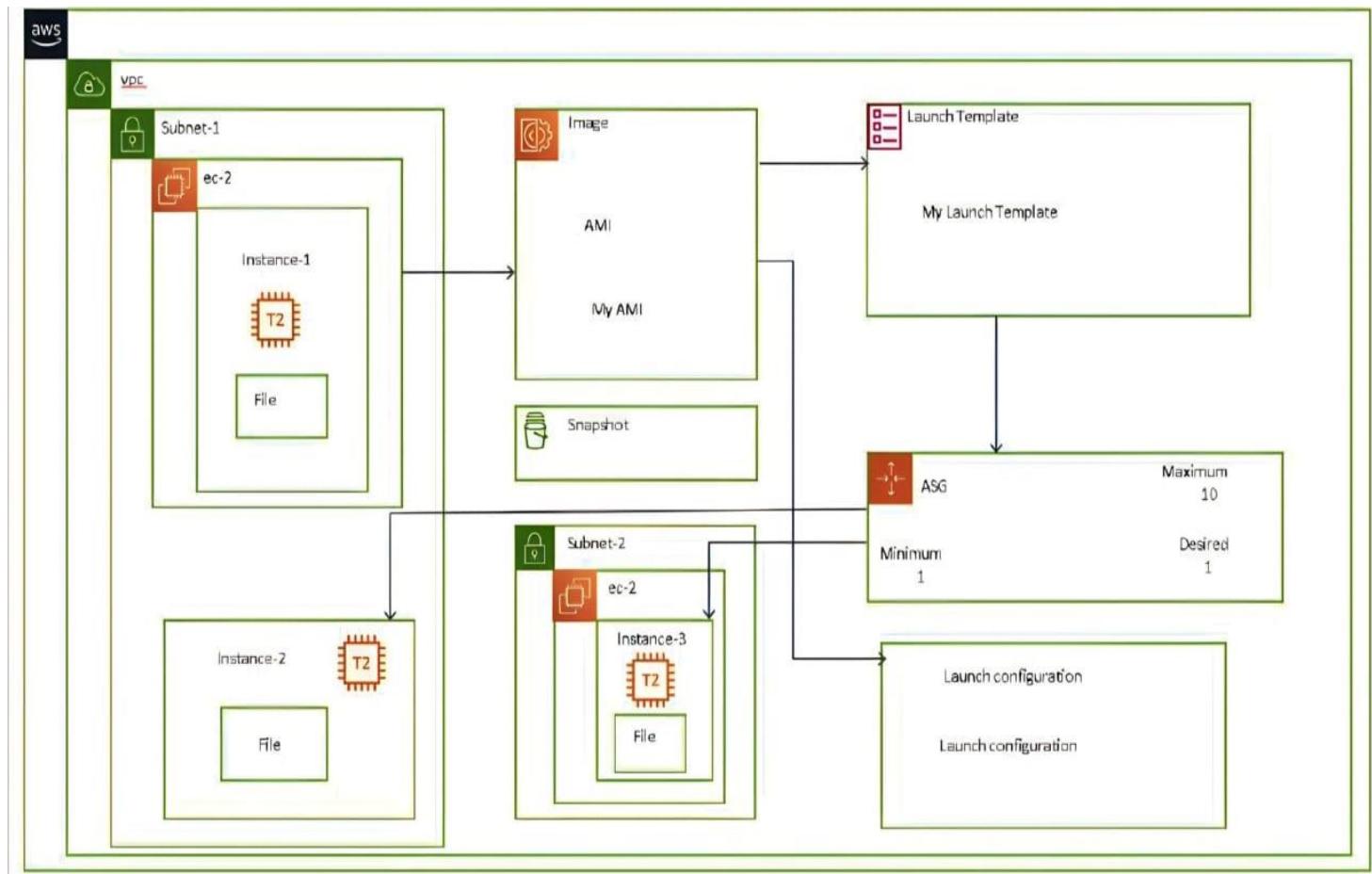
### AWS Services Used:

- a. Virtual Private Cloud(VPC)
- b. Elastic Compute Cloud(EC2)
- c. Identity and Access Management(IAM)
- d. Launching Template
- e. Auto Scaling Group

### Rough Architecture:



## Final Architecture:



## Implementation of the Project

1. Create an instance in the default VPC.
2. Create an AMI from the instance.
3. Go to the Launch Templates page and create a launch template.
4. In the launch template, specify the AMI that you created in step 2.
5. Create an Auto Scaling group using the launch template that you created in step 4.
6. Launch an instance from the Auto Scaling group.
7. Check whether the files that were present on the original instance are present on the new instance.
8. Launch an instance from the template in another subnet.

### Service 1: VPC

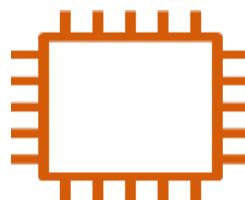


To create a VPC with the name "VPC1," follow these steps:

- Sign into the AWS Management Console.
- Open the Virtual Private Cloud Service.
- In this project we are going to use default vpc

The screenshot shows the AWS EC2 Management Console interface for launching a new instance. The 'Network settings' section is active, displaying two VPC options: 'vpc-061e03a25bd524fb3' (selected) and 'vpc-061e03a25bd524fb3' (default). The 'Auto-assign public IP' option is enabled. In the 'Firewall (security groups)' section, there is a note about security groups and traffic control. The 'Security group name' field is set to 'launch-wizard-3'. The 'Description' field contains the text 'launch-wizard-3 created 2023-07-23T13:02:37.040Z'. The 'Summary' section on the right shows the configuration: 1 instance, Amazon Linux 2023 AMI, t2.micro instance type, New security group, and 1 volume(s) - 8 GiB. A tooltip for the 'Free tier' is displayed, stating: 'Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable)'. The bottom right features the 'Launch instance' button.

## Service 2: EC2



Creating an EC2 instance in a public subnet as a Bastion Host:

- Name your instance as instance-1
- Select “Amazon Linux 2 AMI”.
- Instance type “t2. micro”.
- Select your existing key pair.
- Select default VPC, subnet and enable the auto-assign public IP.
- In the security group section, select availability zone as south-1a.
- create a new security group, add security groups that supports SSH and all the traffic.
- Launch your instance. And connect it

Here we name the instance as instance-1

**Name and tags**

Name: instance-1

Software image (AMI): Amazon Linux 2023 AMI 2023.1.2...  
ami-072ec8f4ea4a6f2cf

Virtual server type (instance type): t2.micro

Firewall (security group): New security group

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

**Quick Start**

Amazon | macOS | Ubuntu | Windows | Red Hat | SUSE Li |

**CloudShell** Feedback Language

© 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

- We choose “Amazon Linux 2 AMI(HVM) – Kernel 5.10.SSD Volume Type in AMI.

**Amazon Machine Image (AMI)**

Amazon Linux 2023 AMI  
Free tier eligible  
ami-072ec8f4ea4a6f2cf (64-bit (x86)) / ami-072ec8f4ea4a6f2cf (64-bit (Arm))  
Virtualization: hvm EHA enabled: true Root device type: ebs

**Description**

Amazon Linux 2023 AMI 2023.1.20230719.0 x86\_64 HVM kernel-6.1

**Architecture** 64-bit (x86) **AMI ID** ami-072ec8f4ea4a6f2cf **Verified provider**

**Instance type**

**t2.micro** Family: t2, 1 vCPU, 1 GB Memory, Current generation: true, On-Demand Linux pricing: 0.0124 USD per Hour, On-Demand Windows pricing: 0.0127 USD per Hour, On-Demand RHEL pricing: 0.0724 USD per Hour, On-Demand SUSE pricing: 0.0124 USD per Hour

All generations Compare instance types

**Summary**

Number of instances: 1

Software image (AMI): Amazon Linux 2023 AMI 2023.1.2...  
ami-072ec8f4ea4a6f2cf

Virtual server type (instance type): t2.micro

Firewall (security group): New security group

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

**Free tier:** In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable)

Cancel Launch instance Review commands

© 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

**Instance type**

t2.micro      Free tier eligible

Family: t2   1 vCPU   1 GB Memory   Current generation: true  
On-Demand Linux pricing: 0.0124 USD per Hour  
On-Demand Windows pricing: 0.017 USD per Hour  
On-Demand RHEL pricing: 0.0724 USD per Hour  
On-Demand SUSE pricing: 0.0124 USD per Hour

All generations      Compare instance types

**Key pair (login)**

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 - required

Dileep1      Create new key pair

**Network settings**

VPC - required

vpc-061e03a25bd524fb3 (default)

Subnet Info

subnet-0ce7b5ac31d74cb59  
VPC: vpc-061e03a25bd524fb3   Owner: 516802179317  
Availability Zone: ap-south-1a   IP addresses available: 4091   CIDR: 172.31.32.0/20

Create new subnet

Auto-assign public IP

Enable

**Firewall (security groups)**

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.

Create security group      Select existing security group

Security group name - required

Dileepsrg

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and \_-!@#\$%^&\_

Description - required

launch-wizard-3 created 2023-07-23T11:55:19.294Z

Inbound Security Group Rules

Security group rule 1 [TCP, 22, 0.0.0.0/0]      Remove

**Summary**

Number of instances

1

Software image (AMI)

Amazon Linux 2023 AMI 2023.1.2...read more  
ami-072ec8f4ea4a6f2cf

Virtual server type (instance type)

t2.micro

Firewall (security group)

New security group

Storage (volumes)

1 volume(s) - 8 GiB

Free tier: in your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable)

Cancel      Launch instance      Review commands

**Network settings**

VPC - required

vpc-061e03a25bd524fb3 (default)

Subnet Info

subnet-0ce7b5ac31d74cb59  
VPC: vpc-061e03a25bd524fb3   Owner: 516802179317  
Availability Zone: ap-south-1a   IP addresses available: 4091   CIDR: 172.31.32.0/20

Create new subnet

Auto-assign public IP

Enable

**Firewall (security groups)**

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.

Create security group      Select existing security group

Security group name - required

Dileepsrg

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and \_-!@#\$%^&\_

Description - required

launch-wizard-3 created 2023-07-23T11:55:19.294Z

Inbound Security Group Rules

Security group rule 1 [TCP, 22, 0.0.0.0/0]      Remove

**Summary**

Number of instances

1

Software image (AMI)

Amazon Linux 2023 AMI 2023.1.2...read more  
ami-072ec8f4ea4a6f2cf

Virtual server type (instance type)

t2.micro

Firewall (security group)

New security group

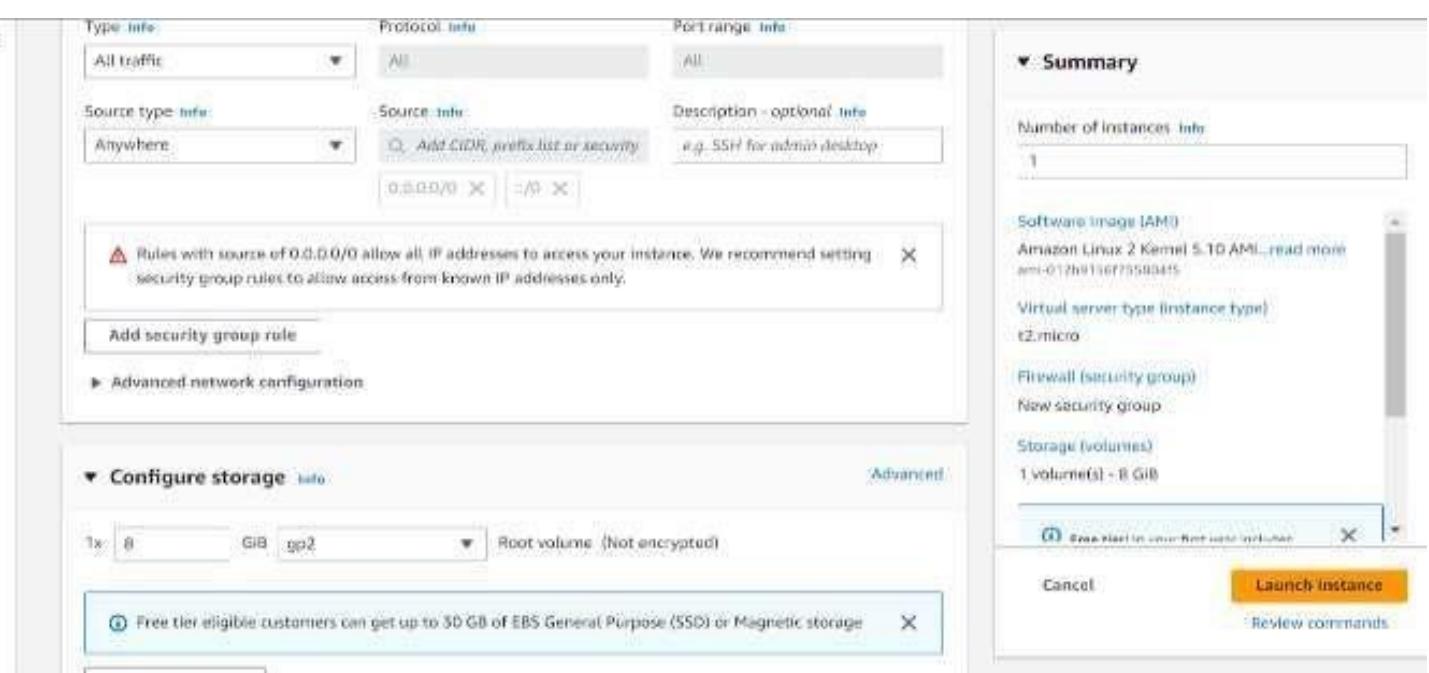
Storage (volumes)

1 volume(s) - 8 GiB

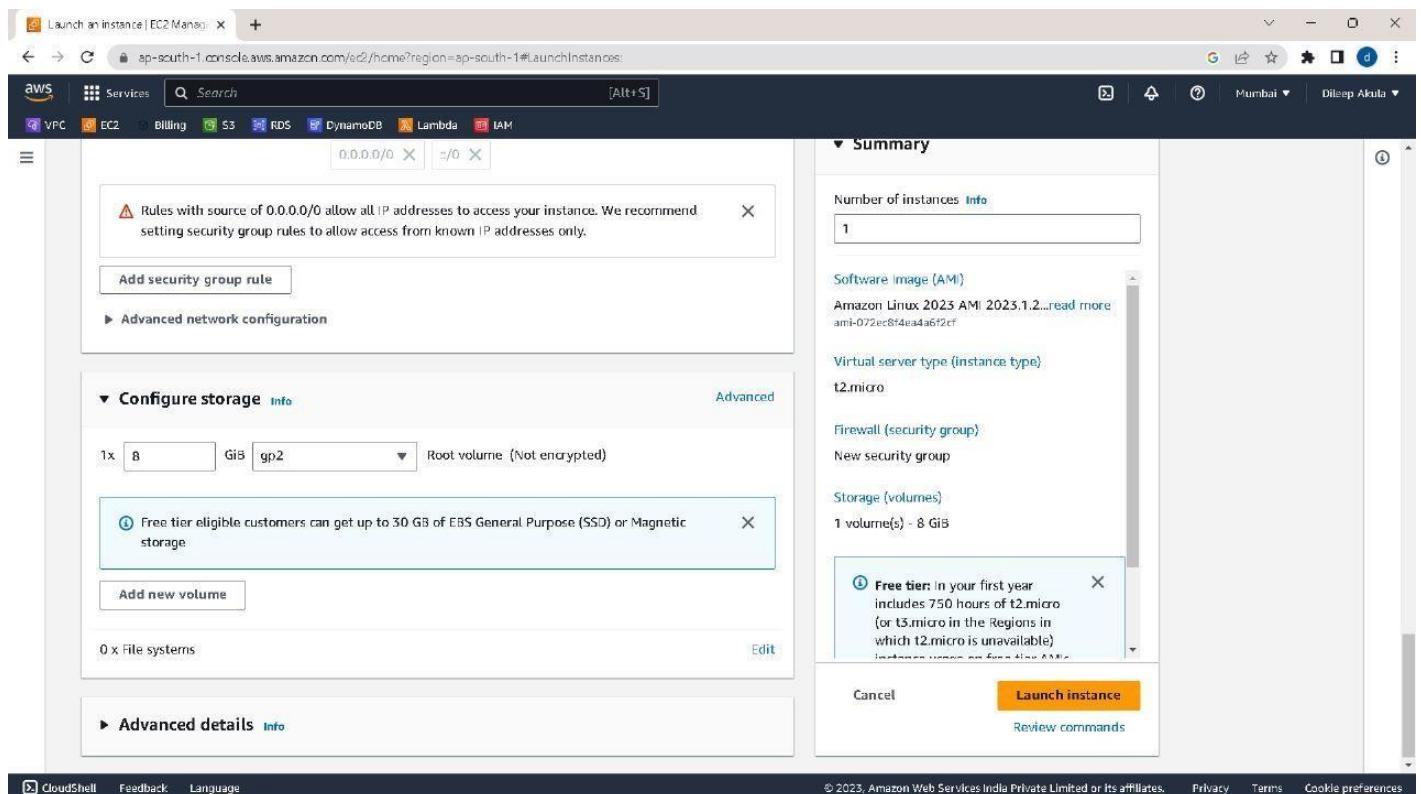
Free tier: in your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable)

Cancel      Launch instance      Review commands

- Add “Type: All traffic”, “source type: Anywhere” in inboundsecurity Group rules.



- Make sure that in configure storage GiB : 8 and route volume : gp2 are present.



Now “instance-1” is successfully launched. and it is connected

The screenshot shows the AWS EC2 Management Console. On the left, there's a sidebar with options like 'New EC2 Experience', 'EC2 Dashboard', 'EC2 Global View', 'Events', 'Instances', 'Instance Types', 'Launch Templates', 'Spot Requests', 'Savings Plans', 'Reserved instances', 'Dedicated Hosts', 'Capacity Reservations', 'Images', 'AMIs', 'AMI Catalog', 'Elastic Block Store', 'Volumes', 'Snapshots', 'Lifecycle Manager', 'CloudShell', 'Feedback', and 'Language'. The main area is titled 'Instances (1/1) info' and shows a table with one row for 'instance-1'. The table columns include Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IPv4 DNS. The instance details show it's running, t2.micro, with no alarms, in ap-south-1a, and has a public IP of ec2-13-233-142. Below this, a detailed view for 'instance-1' is expanded, showing its instance ID (i-0bebe46ca99ef79e9), Public IPv4 address (13.233.142.126), Private IP4 address (172.31.40.38), and Public IPv4 DNS (ec2-13-233-142.ap-south-1.compute.amazonaws.com).

The screenshot shows a terminal session in the AWS CloudShell. The title bar says 'Connect to instance | EC2 Manager | EC2 Instance Connect'. The terminal window displays a series of commands run on the EC2 instance:

```
Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023

[ec2-user@ip-172-31-40-38 ~]$ sudo su
[root@ip-172-31-40-38 ec2-user]# vmstat
procs -----------memory----- swap-- -----io--- system-- -----cpu-----
r b swpd free buff cache si so bi bo in cs us sy id wa st
0 0 0 581580 1876 249764 0 0 2081 1620 185 541 30 5 59 0 6
[root@ip-172-31-40-38 ec2-user]# cat > f1
Hello World
^C
[root@ip-172-31-40-38 ec2-user]# mkdir Dileep
[root@ip-172-31-40-38 ec2-user]# mv f1 Dileep
[root@ip-172-31-40-38 ec2-user]# ls
Dileep
[root@ip-172-31-40-38 ec2-user]# ls Dileep
f1
[root@ip-172-31-40-38 ec2-user]# history
 1  vmstat
 2  cat > f1
 3  mkdir Dileep
 4  mv f1 Dileep
 5  ls
 6  ls Dileep
 7  history
[root@ip-172-31-40-38 ec2-user]#
```

At the bottom of the terminal, it shows the instance ID (i-0bebe46ca99ef79e9), public IP (13.233.142.126), and private IP (172.31.40.38).

## Service 3: AMI



- To create an AMI first we need to create an image, for that look at the following procedure:
- Select the public instance.
- Go to action.
- Go to image and templates.
- Click on create image, give image name, and create the image.

The screenshot shows the 'Create Image' page in the AWS EC2 Management console. The URL is <https://ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#CreateImage:instanceId=i-0bebe46ca99ef79e9>. The page displays the configuration for creating an image from the selected instance (i-0bebe46ca99ef79e9). The 'Image name' field is set to 'MYAMI'. The 'Image description - optional' field contains 'AMI'. The 'No reboot' checkbox is unchecked. Under 'Instance volumes', there is one volume entry: an EBS volume of size 8 GB, type EBS General Purpose, with Throughput of 100 and Delete on termination checked. The 'Encrypted' checkbox is also checked. At the bottom, there are links for CloudShell, Feedback, Language, and a footer with copyright information and links for Privacy, Terms, and Cookie preferences.

The screenshot shows the 'CreateImage' wizard in the AWS Management Console. The current step is 'Configure instance volumes'. It lists one volume: an EBS General Purpose volume of size 8 GiB, with IOPS set to 100 and Throughput to 100. The 'Delete on termination' checkbox is checked. Below the volume list, a note states: 'During the image creation process, Amazon EC2 creates a snapshot of each of the above volumes.' Under 'Tags - optional', there are two radio button options: 'Tag image and snapshots together' (selected) and 'Tag image and snapshots separately'. A key 'MYAMI' is added with a value 'optional'. The 'Create image' button is at the bottom right.

- After images have been created go to AMI, there we can find an AMI.
- Whenever an AMI is created a snapshot gets created along with it.

The screenshot shows the 'Schemas' section of the AWS EC2 Management Console. On the left, the navigation pane includes 'EC2 Global View', 'Events', 'Instances', 'Launch Templates', 'Spot Requests', 'Savings Plans', 'Reserved Instances', 'Dedicated Hosts', 'Capacity Reservations', 'Images', 'AMIs', 'AMI Catalog', 'Elastic Block Store', 'Volumes', 'Schemas' (selected), and 'Lifecycle Manager'. The main area displays a table titled 'Snapshots (1/1) Info' with one entry: 'MYAMI' (snapshot ID: snap-0a33179c835d05151, 8 GiB, Standard storage, Completed status, started on 2023/07/23 17:41). Below the table is a detailed view for 'Snapshot ID: snap-0a33179c835d05151 (MYAMI)', showing details like Volume size (8 GiB), Progress (Available 100%), and Snapshot status (Completed).

The screenshot shows the AWS Management Console interface for the EC2 service. The left sidebar navigation includes 'EC2 Global View', 'Instances' (with sub-options like Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations), 'Images' (AMIs, AMI Catalog), 'Elastic Block Store' (Volumes, Snapshots), and 'Network & Security' (Security Groups, Elastic IPs). The main content area is titled 'Snapshots (1/1)'. A table lists one snapshot entry:

Name	Snapshot ID	Volume size	Description	Storage t...	Snapshot status	Started
MYAMI	snap-0a33179c835d05151	8 GiB	Created by CreateImage(i...	Standard	Completed	2023/07/23 17:41 G

Below the table, a detailed view for 'Snapshot ID: snap-0a33179c835d05151 (MYAMI)' is shown. The 'Details' tab is selected, displaying the following information:

Snapshot ID snap-0a33179c835d05151 (MYAMI)	Volume size 8 GiB	Progress Available (100%)	Snapshot status Completed
Owner 516802179317	Volume ID vol-0e2fb1b2ad7b2d720	Started Sun Jul 23 2023 17:41:57 GMT+0530 (India Standard Time)	Product codes -
Encryption Not encrypted	KMS key ID -	KMS key alias -	KMS key ARN -
Fast snapshot restore	Description Created by CreateImage(i...		

## Service 4:Launch template

To launch a template, follow these steps:

- Open the AWS Management Console in your web browser.
- Navigate to the launch template.
- Click on launch template
- Give the required name and template version descriptions
- In application and os images select my AMI'S
- Select the required instance type, key pair and in network settings select subnet as ap-south-1a
- Select the necessary security group
- Make sure that the storage volume is 8gb
- Now launch the template.

The screenshot shows the AWS EC2 Launch Templates landing page. The left sidebar navigation includes 'Launch templates' under the 'Instances' section. The main content area features a large heading 'EC2 launch templates' and sub-sections 'Streamline, simplify and standardize instance launches' and 'Benefits and features'. A call-to-action button 'Create launch template' is prominently displayed. The footer contains copyright information and links to CloudShell, Feedback, and Language.

The screenshot shows the 'Create launch template' wizard. The left panel contains fields for 'Launch template name and description' (name: 'MyLaunchTemplate', version: 'webserver'), 'Template version description' (description: 'webserver'), 'Auto Scaling guidance' (checkbox: 'Provide guidance to help me set up a template that I can use with EC2 Auto Scaling'), and 'Launch template contents' (links: 'Template tags' and 'Source template'). The right panel is titled 'Summary' and lists configuration sections: 'Software image (AMI)', 'Virtual server type (instance type)', 'Firewall (security group)', and 'Storage (volumes)'. A callout box highlights the 'Free tier' benefit: 'In your first year includes 750 hours of t2.micro (or t5.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 50 GiB of EBS storage, 2 million IOPS, 1 GiB of snapshots, and 100 GiB of bandwidth to the internet.' The bottom right has 'Cancel' and 'Create launch template' buttons.

Screenshot of the AWS EC2 Create launch template interface showing the 'Application and OS Images (Amazon Machine Image)' section.

**Software Image (AMI):** ami-026ead46aa7e8f9a3

**Virtual server type (instance type):** t2.micro

**Firewall (security group):** Not specified

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

**Description:** AMI

**Architecture:** x86\_64

**AMI ID:** ami-026ead46aa7e8f9a3

**Free tier information (t2.micro):**

- Includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

**Create launch template** button is visible.

Screenshot of the AWS EC2 Create launch template interface showing the 'Instance type' section.

**Instance type:** t2.micro

**Key pair (login):** Dileep1

**Network settings:** Subnet: Don't include in launch template

**Free tier information (t2.micro):**

- Includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

**Create launch template** button is visible.

The screenshot shows the 'Create launch template' wizard in the AWS EC2 console. The 'Network settings' section is active, displaying a subnet selection dropdown containing 'subnet-Oce7b5ac31d74cb59'. Below it, there's a note about automatically adding a network interface. Under 'Firewall (security groups)', there are two options: 'Select existing security group' (radio button selected) and 'Create security group'. A dropdown menu shows a single entry: 'Dileepsg sg-0d7772f5a24afc548'. The 'Storage (volumes)' section shows one EBS volume of 8 GiB. On the right, a summary panel displays the AMI ('ami-026ead45aa7e8f9a3'), instance type ('t2.micro'), security group ('Dileepsg'), and storage ('1 volume(s) - 8 GiB'). A tooltip for the 'Free tier' is visible, stating: 'In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.' At the bottom right is a 'Create launch template' button.

The screenshot shows the 'Launch templates' page in the AWS EC2 console. The left sidebar is expanded to show 'New EC2 Experience' and various navigation links like EC2 Dashboard, Instances, Launch Templates, Images, and Elastic Block Store. The main area shows a table titled 'Launch templates (1/1) Info' with one entry: 'lt-0e0280bcc8982171a' (Launch template ID), 'MyLaunchTemplate' (Launch template name), '1' (Default version), and '1' (Latest version). Below the table, the details for 'MyLaunchTemplate (lt-0e0280bcc8982171a)' are shown, including its ID, name, version, and owner ('arn:aws:iam::516802179317:root'). At the bottom, there are tabs for 'Details', 'Versions', and 'Template tags', and a 'Launch template version details' section with an 'Actions' button.

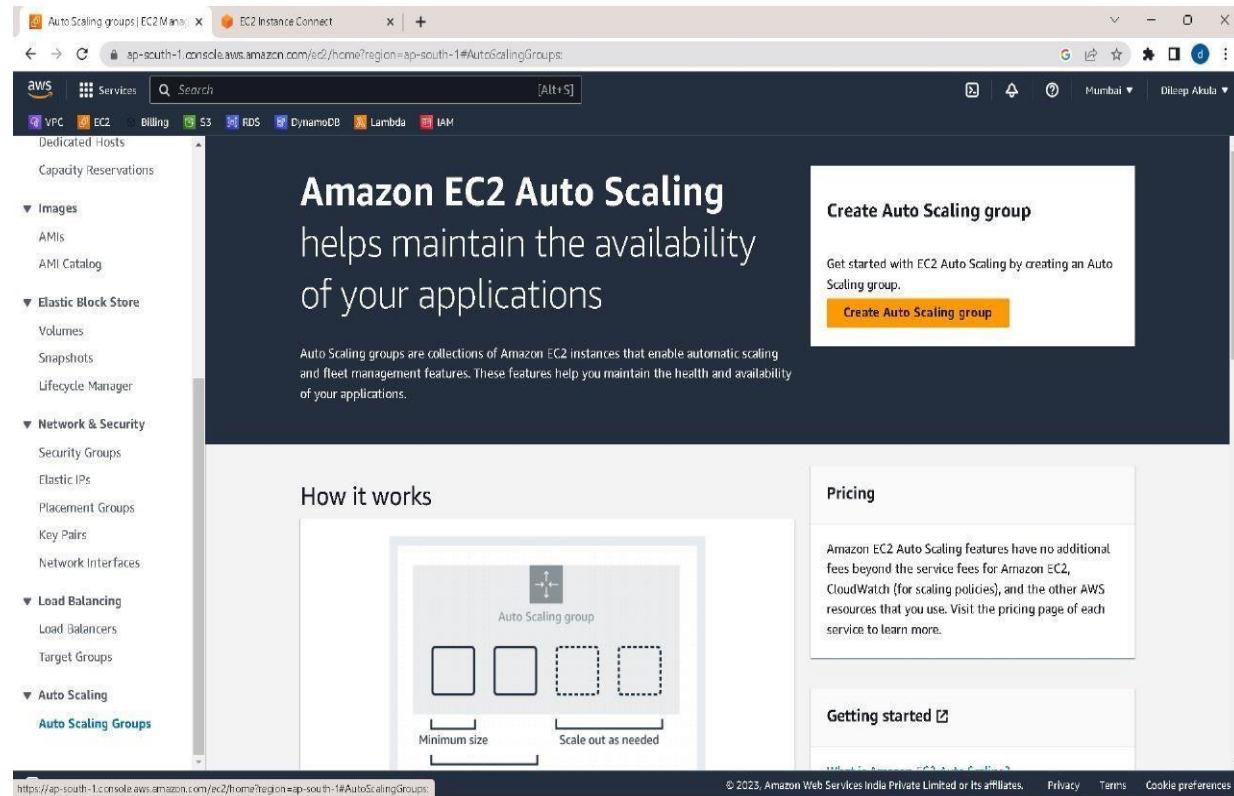
- Choose instance name as “instance-private”.

- Select “Amazon Linux 2 AMI”.
- Instance type “t2. micro”.
- Select your existing key pair which is used in instance-public.
- Select your custom VPC, public subnet and disable the auto-assign public IP.
- In the security group section, select availability zone as south-1b.
- Choose existing security group that is used in instance-public.
- Launch your instance.

## Service 5:Auto Scaling Group

### Creating an Auto Scaling Group

- An auto scaling group creates a running instance itself which reduces the CPU utilization and provides users to access an instance with all the features.
- So, we create a template and auto scaling group through existing AMI.



Create Auto Scaling group | EC2 | EC2 Instance Connect

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

Services Search [Alt+S]

VPC EC2 Billing S3 RDS DynamoDB Lambda IAM

EC2 > Auto Scaling groups > Create Auto Scaling group

Step 1 Choose launch template

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

**Choose launch template** Info

Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group.

**Name**

**Auto Scaling group name**  
Enter a name to identify the group.  
 Must be unique to this account in the current Region and no more than 255 characters.

**Launch template** Info

For accounts created after May 31, 2023, the EC2 console only supports creating Auto Scaling groups with launch templates. Creating Auto Scaling groups with launch configurations is not recommended but still available via the CLI and API until December 31, 2023.

**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.

CloudShell Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

Create Auto Scaling group | EC2 | EC2 Instance Connect

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

Services Search [Alt+S]

VPC EC2 Billing S3 RDS DynamoDB Lambda IAM

Step 7 Review

**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.

**Version**  
Default (1)

Description	Launch template	instance type
webserver	<input type="text" value="MyLaunchTemplate"/> <input type="button" value="sg-0e0280bcc8982171a"/>	t2.micro
AMI ID	Security groups	Request Spot Instances
ami-026ead46aa7e8f9a3	-	No
Key pair name	Security group IDs	
Dileep1	<input type="text" value="sg-0d7772f5a24afc548"/> <input type="button" value="sg-0d7772f5a24afc548"/>	

**Additional details**

Storage (volumes)	Date created
-	Sun Jul 23 2023 17:46:21 GMT+0530 (India Standard Time)

CloudShell Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

Create Auto Scaling group | EC2 | EC2 Instance Connect

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

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-061e03a25bd524fb3  
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-0ce7b5ac31d74cb59  
172.31.32.0/20 Default

ap-south-1b | subnet-0a2581cab7b36bcc8  
172.31.0.0/20 Default

ap-south-1c | subnet-0c2eaa02d9180f9ba  
172.31.16.0/20 Default

Create a subnet

**Instance type requirements**

Override launch template

You can keep the same instance attributes or instance type from your launch template, or you can choose to override the launch template by specifying different instance attributes or manually adding instance types.

CloudShell Feedback Language

© 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

Create Auto Scaling group | EC2 | EC2 Instance Connect

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

Step 5 - optional Add notifications

Step 6 - optional Add tags

Step 7 Review

**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-0ce7b5ac31d74cb59  
172.31.32.0/20 Default

ap-south-1b | subnet-0a2581cab7b36bcc8  
172.31.0.0/20 Default

ap-south-1c | subnet-0c2eaa02d9180f9ba  
172.31.16.0/20 Default

Create a subnet

**Instance type requirements**

Override launch template

You can keep the same instance attributes or instance type from your launch template, or you can choose to override the launch template by specifying different instance attributes or manually adding instance types.

Launch template	Version	Description
MyLaunchTemplate	Default	webserver

Launch template  
MyLaunchTemplate  
lt-0e0280bcc8982171a

Instance type  
t2.micro

Cancel Skip to review Previous Next

CloudShell Feedback Language

© 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

Create Auto Scaling group | EC2 | EC2 Instance Connect

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

Services Search [Alt+S]

VPC EC2 Billing S3 RDS DynamoDB Lambda IAM

EC2 > Auto Scaling groups > Create Auto Scaling group

Step 1 Choose launch template

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

### Configure advanced options - *optional*

Choose a load balancer to distribute incoming traffic for your application across instances to make it more reliable and easily scalable. You can also set options that give you more control over health check replacements and monitoring.

#### Load balancing

Use the options below to attach your Auto Scaling group to an existing load balancer, or to a new load balancer that you define.

No load balancer Traffic to your Auto Scaling group will not be fronted by a load balancer.

Attach to an existing load balancer Choose from your existing load balancers.

Attach to a new load balancer Quickly create a basic load balancer to attach to your Auto Scaling group.

**Attach to a new load balancer**

Define a new load balancer to create for attachment to this Auto Scaling group.

**Load balancer type**

Choose from the load balancer types offered below. Type selection cannot be changed after the load balancer is created. If you need a different type of load balancer than those offered here, visit the Load Balancing console.

Application Load Balancer HTTP, HTTPS

Network Load Balancer TCP, UDP, TLS

Load balancer name

CloudShell Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

Create Auto Scaling group | EC2 | EC2 Instance Connect

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

Services Search [Alt+S]

VPC EC2 Billing S3 RDS DynamoDB Lambda IAM

Step 7 Review

Choose from the load balancer types offered below. Type selection cannot be changed after the load balancer is created. If you need a different type of load balancer than those offered here, visit the Load Balancing console.

Application Load Balancer HTTP, HTTPS

Network Load Balancer TCP, UDP, TLS

Load balancer name

Name cannot be changed after the load balancer is created.

asg1-1

Load balancer scheme

Scheme cannot be changed after the load balancer is created.

Internal

Internet-facing

Network mapping

Your new load balancer will be created using the same VPC and Availability Zone selections as your Auto Scaling group. You can select different subnets and add subnets from additional Availability Zones.

VPC

vpc-061e05a25bd524fb5

Availability Zones and subnets

You must select a single subnet for each Availability Zone enabled. Only public subnets are available for selection to support DNS resolution.

ap-south-1c	subnet-0c2eaa02d9180f9ba
ap-south-1a	subnet-0ce7b5ac31d74cb59
ap-south-1b	subnet-0a2581cab7b36bcc8

Databases and services

CloudShell Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

Create Auto Scaling group | EC2 | EC2 Instance Connect

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

AWS Services Search [Alt+S]

VPC EC2 Billing S3 RDS DynamoDB Lambda IAM

Listeners and routing

If you require secure listeners, or multiple listeners, you can configure them from the Load Balancing console after your load balancer is created.

Protocol	Port	Default routing (forward to)
HTTP	80	Create a target group

New target group name  
An instance target group with default settings will be created...

asg1-1

Tags - optional

Consider adding tags to your load balancer. Tags enable you to categorize your AWS resources so you can more easily manage them.

Add tag

50 remaining

Health checks

Health checks increase availability by replacing unhealthy instances. When you use multiple health checks, all are evaluated, and if at least one fails, instance replacement occurs.

EC2 health checks

Always enabled

Additional health check types - optional Info

Turn on Elastic Load Balancing health checks Recommended

Elastic Load Balancing monitors whether instances are available to handle requests. When it reports an unhealthy instance, EC2 Auto Scaling can replace it on its next periodic check.

Health check grace period Info

© 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

Create Auto Scaling group | EC2 | EC2 Instance Connect

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

AWS Services Search [Alt+S]

VPC EC2 Billing S3 RDS DynamoDB Lambda IAM

Listeners and routing

If you require secure listeners, or multiple listeners, you can configure them from the Load Balancing console after your load balancer is created.

Protocol	Port	Default routing (forward to)
HTTP	80	Create a target group

New target group name  
An instance target group with default settings will be created...

asg1-1

Tags - optional

Consider adding tags to your load balancer. Tags enable you to categorize your AWS resources so you can more easily manage them.

Add tag

50 remaining

Health checks

Health checks increase availability by replacing unhealthy instances. When you use multiple health checks, all are evaluated, and if at least one fails, instance replacement occurs.

EC2 health checks

Always enabled

Additional health check types - optional Info

Turn on Elastic Load Balancing health checks Recommended

Elastic Load Balancing monitors whether instances are available to handle requests. When it reports an unhealthy instance, EC2 Auto Scaling can replace it on its next periodic check.

Health check grace period Info

© 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

Create Auto Scaling group | EC2 | EC2 Instance Connect

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

EC2 Services Search [Alt+S]

VPC EC2 Billing S3 RDS DynamoDB Lambda IAM

Mumbai Dilip Akula

EC2 > Auto Scaling groups > Create Auto Scaling group

Step 1 Choose launch template

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

### Configure group size and scaling policies - optional Info

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.

#### Group size - optional Info

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.

Desired capacity

Minimum capacity

Maximum capacity

#### 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

Metric type

Target value

Instances need  seconds warm up before including in metric

Disable scale in to create only a scale-out policy

Instance scale-in protection - optional

CloudShell Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

Create Auto Scaling group | EC2 | EC2 Instance Connect

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

EC2 Services Search [Alt+S]

VPC EC2 Billing S3 RDS DynamoDB Lambda IAM

Mumbai Dilip Akula

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

Metric type

Target value

Instances need  seconds warm up before including in metric

Disable scale in to create only a scale-out policy

### Instance scale-in protection - optional

CloudShell Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

The screenshot shows the AWS Cloud console with the URL <https://ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#CreateAutoScalingGroup>. The page is titled "Create Auto Scaling group | EC2". The "Scaling policy name" field contains "Target Tracking Policy". The "Metric type" dropdown is set to "Average CPU utilization". The "Target value" input field has "60" entered. The "Instances need" input field has "50" entered, with the note "seconds warm up before including in metric". A checkbox for "Disable scale in to create only a scale-out policy" is unchecked. Below this is a section titled "Instance scale-in protection - optional" with a checkbox for "Enable instance scale-in protection" which is also unchecked. At the bottom are "Cancel", "Skip to review", "Previous", and "Next" buttons.

There is no need to add the notifications.

The screenshot shows the AWS Cloud console with the URL <https://ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#CreateAutoScalingGroup>. The page is titled "Create Auto Scaling group | EC2". On the left, a sidebar lists steps: "Step 1 Choose launch template", "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", and "Step 6 - optional Add tags", with "Step 7 Review" at the bottom. The main content area is titled "Add notifications - optional" with a link to "Info". It says "Send notifications to SNS topics whenever Amazon EC2 Auto Scaling launches or terminates the EC2 instances in your Auto Scaling group." A "Add notification" button is present. At the bottom are "Cancel", "Skip to review", "Previous", and "Next" buttons.

Screenshot of the AWS EC2 Auto Scaling group creation wizard Step 4: Add tags - optional.

The page shows a sidebar with steps: Step 1 (Choose launch template), Step 2 (Choose instance launch options), Step 3 (Configure advanced options), Step 4 (Configure group size and scaling policies), Step 5 (Add notifications), Step 6 (Add tags), and Step 7 (Review). The main content area displays a "Tags (1)" section with one tag named "Name" with value "Autoasg". A checkbox "Tag new instances" is checked. A note says: "You can optionally choose to add tags to instances (and their attached EBS volumes) by specifying tags in your launch template. We recommend caution, however, because the tag values for instances from your launch template will be overridden if there are any duplicate keys specified for the Auto Scaling group." Buttons at the bottom include "Cancel", "Previous", and "Next".

Screenshot of the AWS EC2 Auto Scaling group creation wizard Step 5: Add notifications.

The page shows sections for "Instance scale-in protection" (with an unchecked checkbox "Enable instance protection from scale in") and "Step 5: Add notifications" (with a "Notifications" section showing "No notifications").

**Step 6: Add tags**

The "Tags (1)" section shows one tag named "Name" with value "Autoasg". A checkbox "Tag new instances" is checked. Buttons at the bottom include "Cancel", "Previous", and a prominent orange "Create Auto Scaling group" button.

The screenshot shows the AWS Auto Scaling Groups page. At the top, a green banner indicates: "asg1, 1 Scaling policy, 1 Load balancer, 1 Target group, 1 Listener created successfully. 1 new target group has been attached to ASG." Below this, the "Auto Scaling groups (1/1) Info" section shows a table with one row for "asg1". The table columns include Name, Launch template/configuration, Instantiated, Status, Desired capacity, Min, Max, and Availability zone. The "Status" column shows "Updating capacity...". The "Actions" dropdown menu is open, with "Edit" highlighted. The "Launch configurations" and "Launch templates" tabs are also visible.

Auto Scaling group: asg1

Details Activity Automatic scaling Instance management Monitoring Instance refresh

Group details

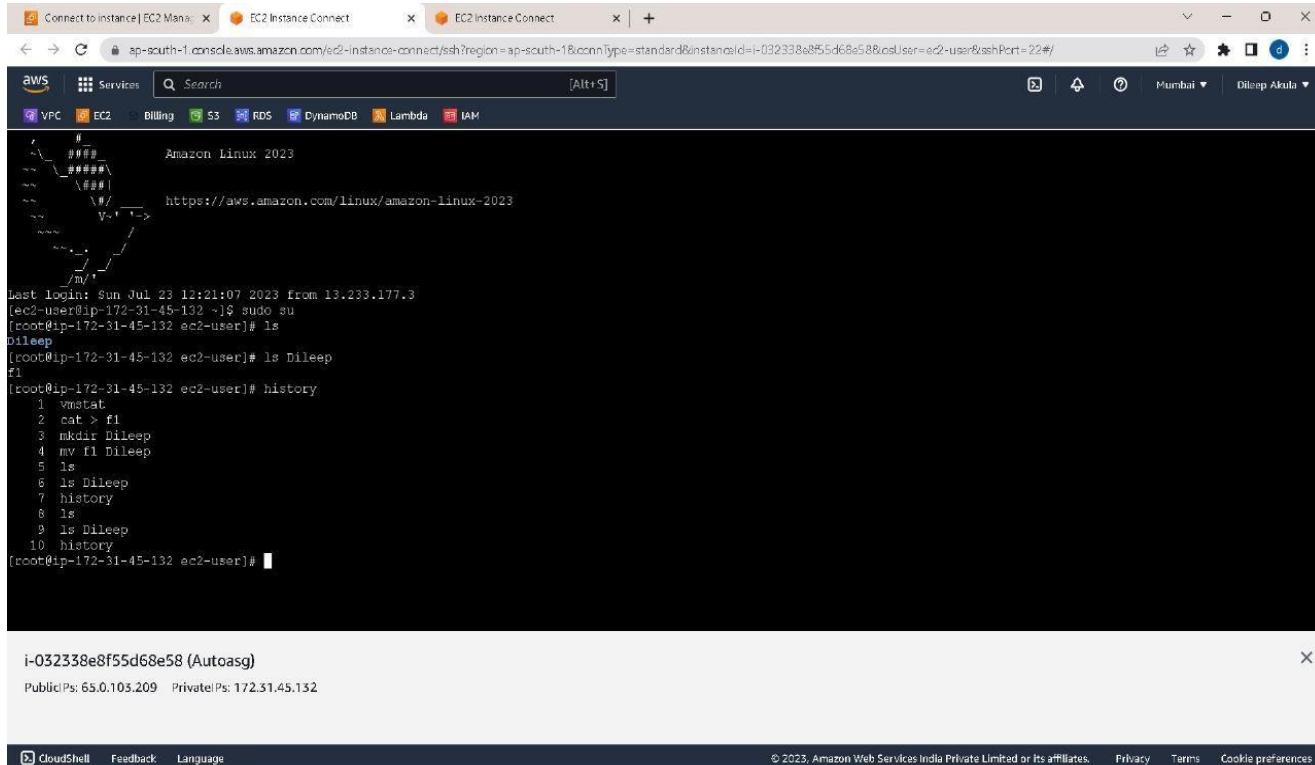
Auto Scaling group name: asg1 Desired capacity: 1 Status: Updating capacity Amazon Resource Name (ARN): arn:aws:autoscaling:ap-south-1:516802179317:autoScalingGroup:78712f8d-0dfc-4cf6-9bf

Therefore Auto Scaling Group is created.

An instance autoasg is created in instance page. Connect it.

The screenshot shows the AWS Instances page. On the left sidebar, under the "Instances" section, "New EC2 Experience" is selected. The main table lists two instances: "instance-1" and "Autoasg". The "Autoasg" instance is selected, showing its details in the right panel. The instance summary table includes fields for Instance ID (i-032338e8f55d68e58), Public IPv4 address (65.0.103.209), Private IPv4 addresses (172.31.45.132), Instance state (Running), and Public IPv4 DNS (ec2-65-0-103-209.ap-south-1.compute.amazonaws.com).

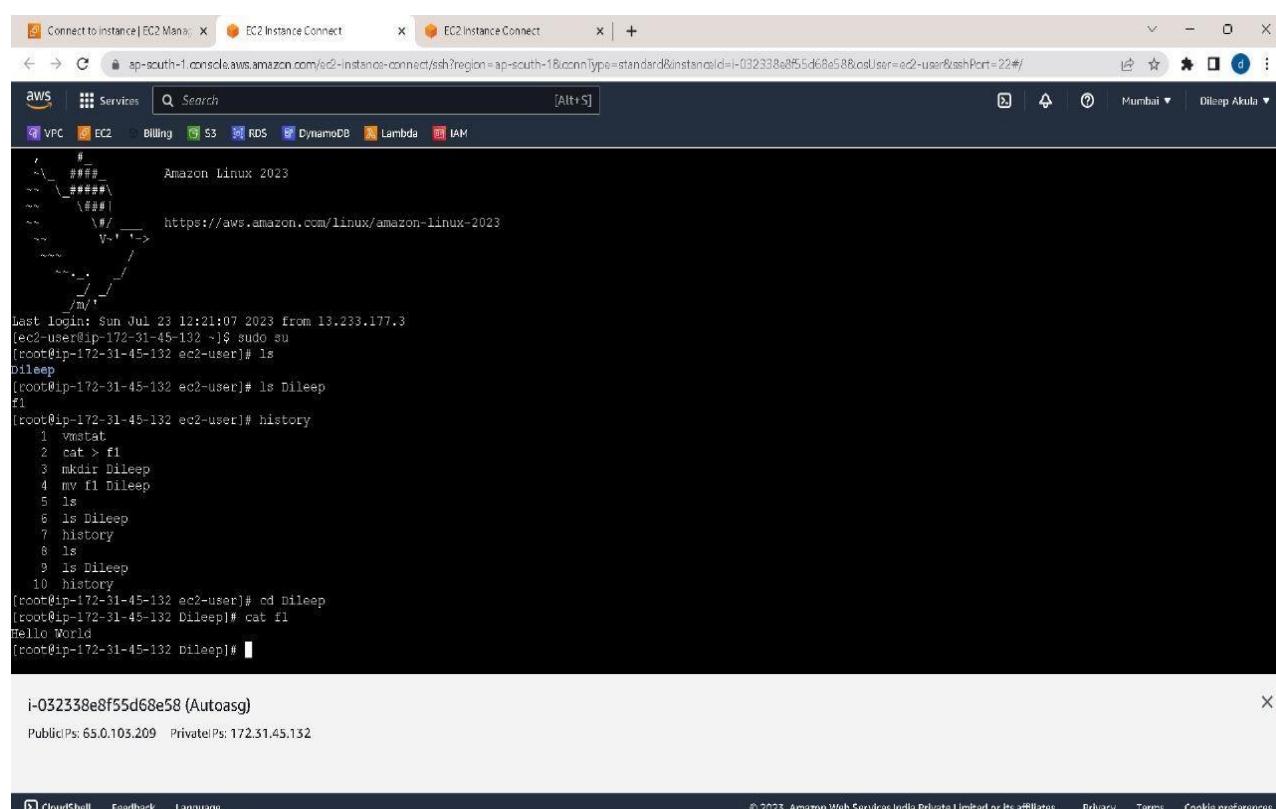
## Execute the following commands



The screenshot shows a terminal window in AWS CloudShell. The session history is displayed, starting with a root shell on an Amazon Linux 2023 instance. The user runs several commands including 'vmstat', 'cat > f1', 'mkdir Dileep', 'mv f1 Dileep', 'ls', and 'history'. The session ends with a root command to exit.

```
Last login: Sun Jul 23 12:21:07 2023 from 13.233.177.3
[ec2-user@ip-172-31-45-132 ~]$ sudo su
[root@ip-172-31-45-132 ec2-user]# ls
Dileep
[root@ip-172-31-45-132 ec2-user]# ls Dileep
f1
[root@ip-172-31-45-132 ec2-user]# history
 1 vmstat
 2 cat > f1
 3 mkdir Dileep
 4 mv f1 Dileep
 5 ls
 6 ls Dileep
 7 history
 8 ls
 9 ls Dileep
10 history
[root@ip-172-31-45-132 ec2-user]#
```

i-032338e8f55d68e58 (Autoasg)  
PublicIPs: 65.0.105.209 PrivateIPs: 172.31.45.132



This screenshot shows a second terminal session in AWS CloudShell. It starts with a root shell on an Amazon Linux 2023 instance. The user runs 'ls' to see the directory contents, which include a file named 'Dileep'. They then run 'cd Dileep' to change into that directory. Finally, they run 'cat f1' to print the contents of the file 'f1', which contains the text 'Hello World'.

```
Last login: Sun Jul 23 12:21:07 2023 from 13.233.177.3
[ec2-user@ip-172-31-45-132 ~]$ sudo su
[root@ip-172-31-45-132 ec2-user]# ls
Dileep
[root@ip-172-31-45-132 ec2-user]# cd Dileep
[root@ip-172-31-45-132 Dileep]# cat f1
Hello World
[root@ip-172-31-45-132 Dileep]#
```

i-032338e8f55d68e58 (Autoasg)  
PublicIPs: 65.0.105.209 PrivateIPs: 172.31.45.132

The file that is created in instance1 is present in autoASG instance

Now launch instance instance-2 in the subnet same as that of instance-1 from template MyLaunchTemplate.

This screenshot shows the 'Launch instance from template' wizard in the AWS Management Console. The 'Summary' section on the right indicates 1 instance will be launched using AMI ami-026ead46aa7e8f9a3, t2.micro instance type, and security group Dileepsg. The 'Free tier' information states it includes 750 hours of t2.micro usage in the first year. The 'Launch instance' button is highlighted in orange at the bottom right.

**Launch instance from template**

Launching from a template allows you to launch from an instance configuration that you would have saved in the past. These saved configurations can be reused and shared with other users to standardize launches across an organisation.

**Choose a launch template**

Source template: MyLaunchTemplate (ID: lt-0a0260bcc8982171a)

Source template version: 1 (Default) webserver

**Instance details**

Your instance details are listed below. Any fields that are not specified as part of the configuration below will use the template or default values for those fields. Ensure that you have permissions to override these parameters or your instance launch will fail.

**Application and OS Images (Amazon Machine Image)**

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're looking for below.

Cancel      Launch instance      Review commands

This screenshot shows the 'Application and OS Images (Amazon Machine Image)' search results page. It displays a search bar, filter tabs (AMI from catalog, Recents, My AMIs, Quick Start), and a list of AMIs. The 'My AMIs' tab is selected, showing the user's own AMI named 'MYAMI'. The summary on the right is identical to the previous screenshot, indicating 1 instance will be launched using the same configuration.

**Application and OS Images (Amazon Machine Image)**

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

AMI from catalog    Recents    **My AMIs**    Quick Start

Template or default value    Owned by me    Shared with me

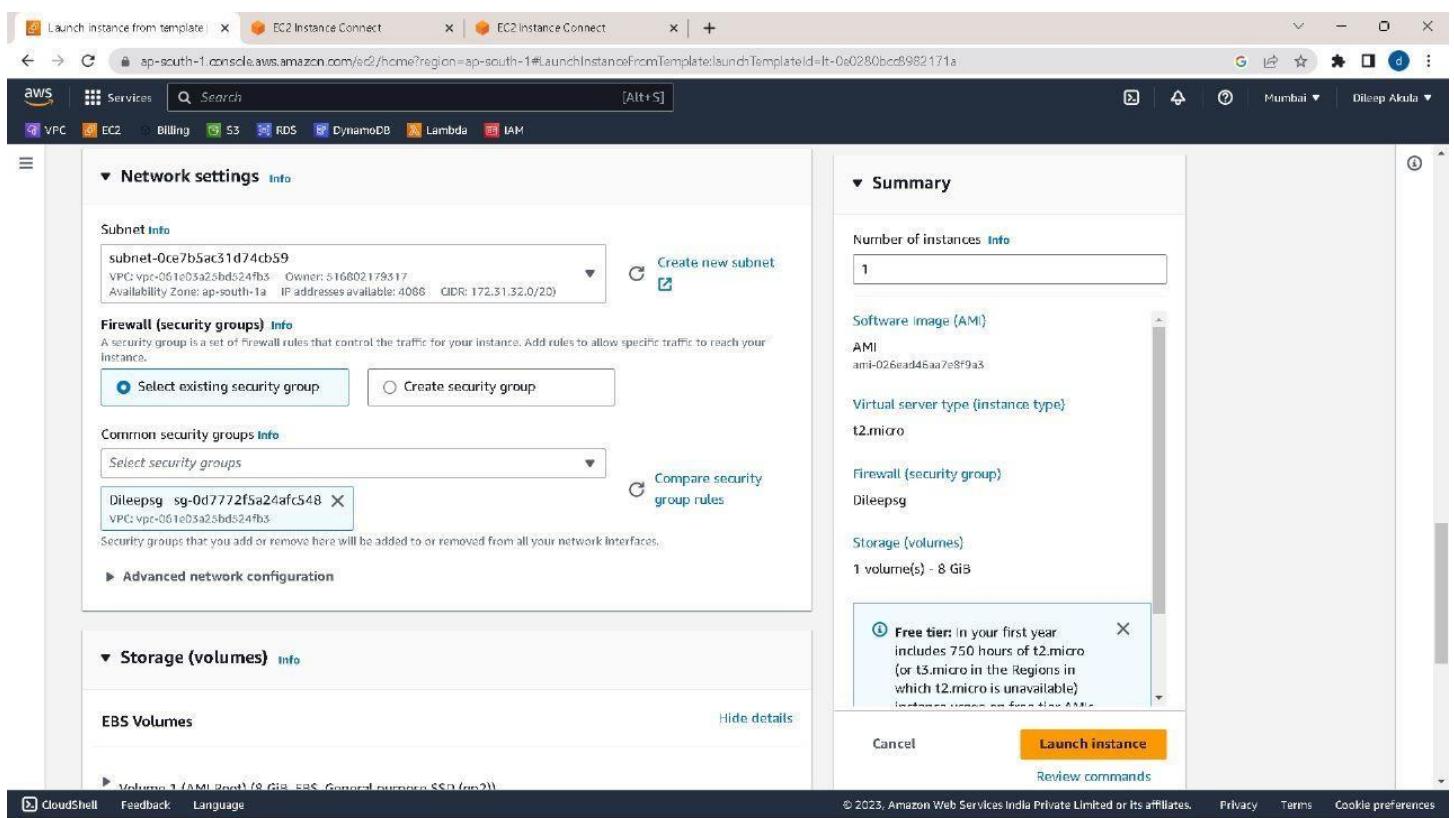
**Amazon Machine Image (AMI)**

MYAMI  
ami-026ead46aa7e8f9a3  
2023-07-23T12:10:27.000Z    Virtualization:hvm    ENA enabled:true    Root device type: ebs    boot mode: uefi-preferred

Description  
AMI

Architecture    AMI ID  
x86\_64    ami-026ead46aa7e8f9a3

Cancel      Launch instance      Review commands



Now launch the instance and connect it

Instances (1/3) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
instance-1	i-0bebe46ca99ef79e9	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1a	ec2-13-233-142
Autoasg	i-032338ef55d68e58	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1a	ec2-65-0-103-2
<b>instance-2</b>	<b>i-0e850281f28d70d3f</b>	<b>Running</b>	<b>t2.micro</b>	-	No alarms	ap-south-1a	ec2-65-0-132-9

**Instance: i-0e850281f28d70d3f (instance-2)**

**Details** | Security | Networking | Storage | Status checks | Monitoring | Tags

**Instance summary**

instance ID	Public IPv4 address	Private IPv4 addresses
i-0e850281f28d70d3f (instance-2)	65.0.132.95   open address	172.31.34.68
IPv6 address	Instance state	Public IPv4 DNS
-	Running	ec2-65-0-132-95.ap-south-1.compute.amazonaws.com   open address
Hostname type	Private IP DNS name (IPv4 only)	
IP name: ip-172-31-34-68.ap-south-1.compute.internal	ip-172-31-34-68.ap-south-1.compute.internal	

Run the following commands

```

# Last login: Sun Jul 23 12:00:31 2023 from 13.233.177.3
[ec2-user@ip-172-31-34-68 ~]$ sudo su
[root@ip-172-31-34-68 ec2-user]# ls
dileep
[root@ip-172-31-34-68 ec2-user]# ls Dileep
[root@ip-172-31-34-68 ec2-user]# cd Dileep
[root@ip-172-31-34-68 Dileep]# cat f1
Hello World
[root@ip-172-31-34-68 Dileep]# history
1  vmstat
2  cat > f1
3  mkdir Dileep
4  mv f1 Dileep
5  ls
6  ls Dileep
7  history
8  ls
9  ls Dileep
10 cd Dileep
11 cat f1
12 history
[root@ip-172-31-34-68 Dileep]#

```

i-0e850281f28d70d3f (instance-2)  
PublicIPs: 65.0.132.95 PrivateIPs: 172.31.34.68

The files that are present in instance-1 reflects in instance-2 It is due to auto scaling group

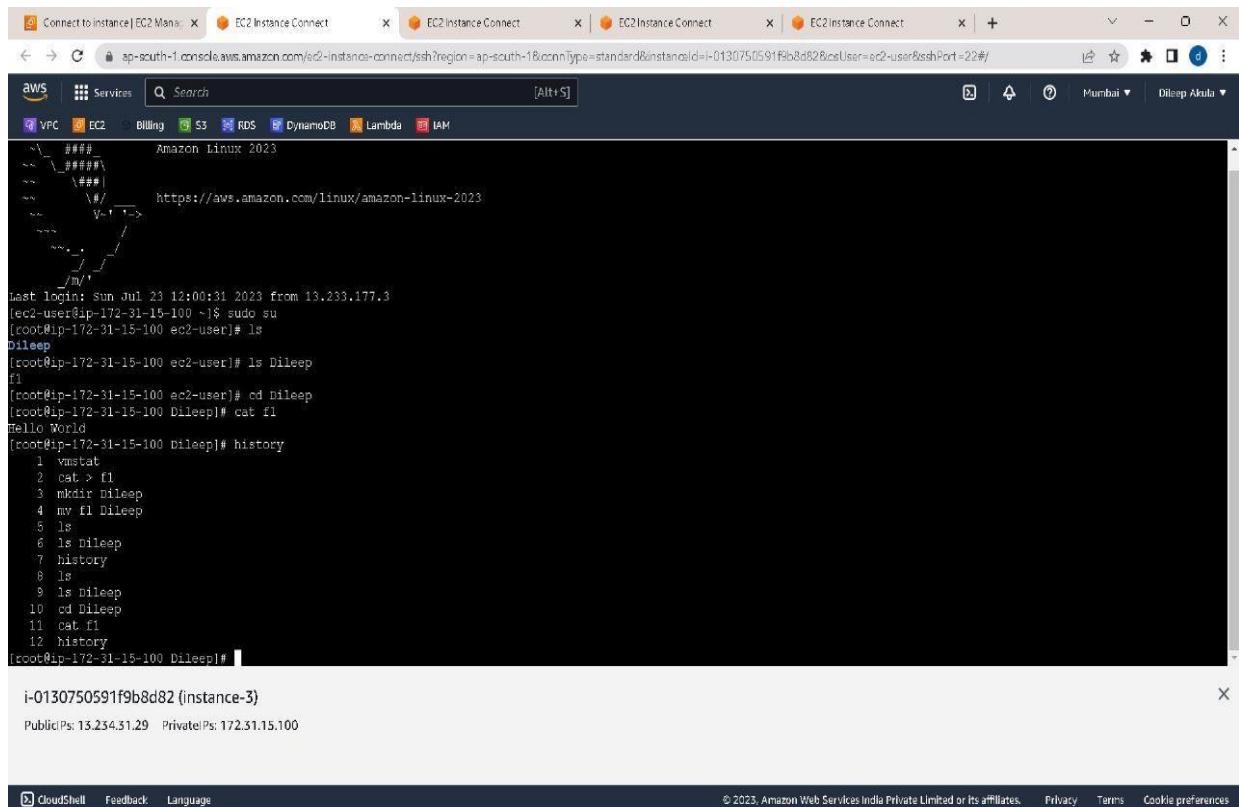
Now create another instance instance-3 from template My LaunchTemplate bit in different subnet select ap-south-1b

The screenshot shows the AWS EC2 Manager interface with the 'Launch templates' section selected. A modal window titled 'MyLaunchTemplate (lt-0e0280bcc8982171a)' is open, displaying the 'Launch template details' and 'Launch template version details' sections. The 'Actions' dropdown menu is open, showing options like 'Launch instance from template', 'Modify template (Create new version)', and 'Delete template'. The 'Launch template ID' is lt-0e0280bcc8982171a, the 'Launch template name' is MyLaunchTemplate, and the 'Default version' is 1. The 'Owner' is arn:aws:iam::516802179317:root.

The screenshot shows the 'Launch instance from template' wizard in the AWS Management Console. The current step is 'Choose a launch template'. It displays a dropdown for 'Source template' set to 'MyLaunchTemplate' and a dropdown for 'Source template version' set to '1 (Default) webserver'. To the right, the 'Summary' section shows the configuration: 1 instance, AMI 'ami-026ead4d4aa7e8f9a3', instance type 't2.micro', security group 'Dileepsg', and 1 volume (8 GiB). A tooltip for the 'Free tier' is visible, stating: 'In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable)'. At the bottom are 'Cancel', 'Launch instance', and 'Review commands' buttons.

The screenshot shows the 'Launch instance from template' wizard in the AWS Management Console. The current step is 'Network settings'. It shows a subnet selection dropdown with 'subnet-0a2581cab7b36bcc8' selected, and a 'Create new subnet' button. Below it, a 'Firewall (security groups)' section shows 'Dileepsg sg-0d7772f5a24afc548' selected, with a 'Select existing security group' and 'Create security group' radio buttons. A 'Common security groups' dropdown lists 'Dileepsg sg-0d7772f5a24afc548'. A 'Compare security group rules' button is also present. To the right, the 'Summary' section shows the configuration: 1 instance, AMI 'ami-026ead4d4aa7e8f9a3', instance type 't2.micro', security group 'Dileepsg', and 1 volume (8 GiB). A tooltip for the 'Free tier' is visible, stating: 'In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable)'. At the bottom are 'Cancel', 'Launch instance', and 'Review commands' buttons.

Run the following commands



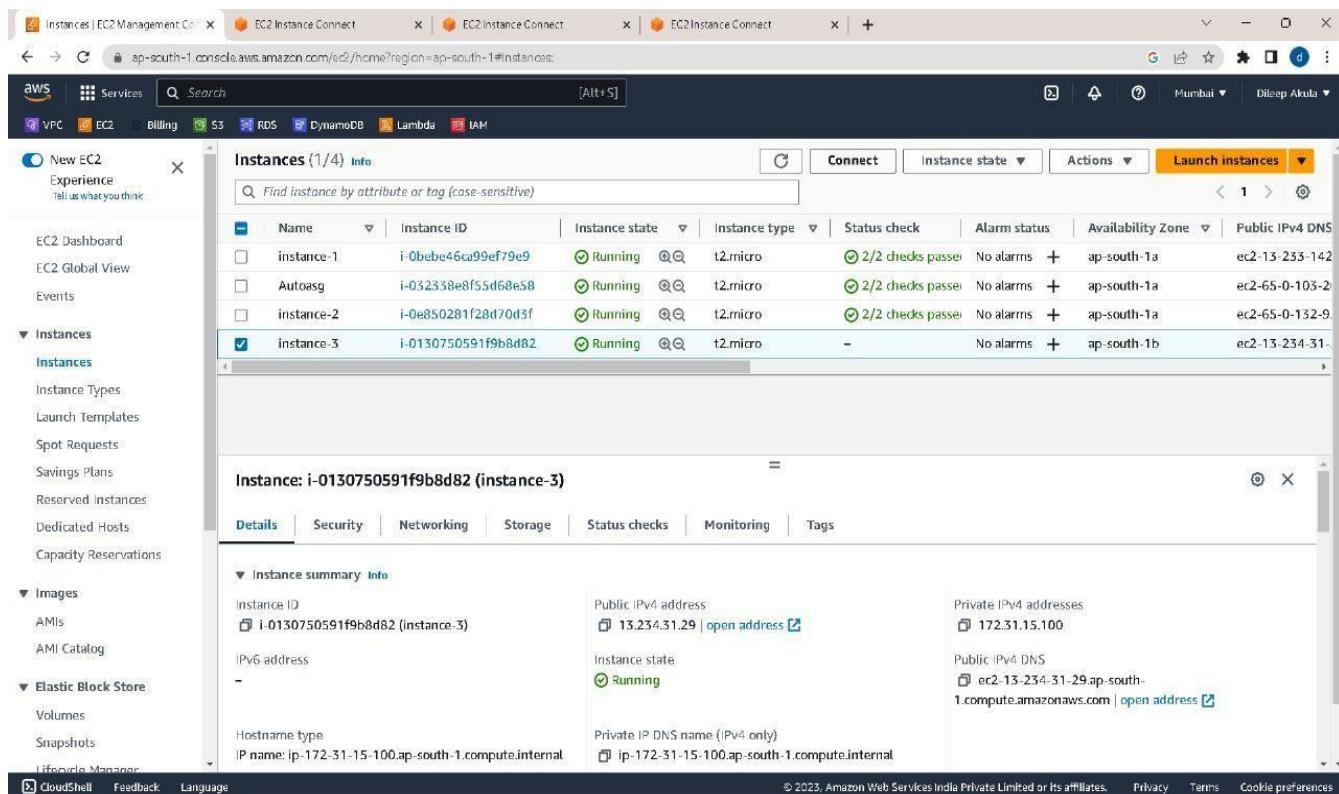
The screenshot shows a terminal session in AWS CloudShell. The user has run several commands to create a directory named 'Dileep' and a file named 'f1' containing 'Hello World'. The session history is as follows:

```
Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023

Last login: Sun Jul 23 12:00:31 2023 from 13.233.177.3
[ec2-user@ip-172-31-15-100 ~]$ sudo su
[root@ip-172-31-15-100 ec2-user]# ls Dileep
[root@ip-172-31-15-100 ec2-user]# cd Dileep
[root@ip-172-31-15-100 Dileep]# cat f1
Hello World
[root@ip-172-31-15-100 Dileep]# history
 1 vmsstat
 2 cat > f1
 3 mkdir Dileep
 4 mv f1 Dileep
 5 ls
 6 ls Dileep
 7 history
 8 ls
 9 ls Dileep
10 cd Dileep
11 cat f1
12 history
[root@ip-172-31-15-100 Dileep]#
```

i-0130750591f9b8d82 (instance-3)

Public IPs: 13.234.31.29 Private IPs: 172.31.15.100



The screenshot shows the AWS EC2 Management Console. The left sidebar is collapsed, and the main area displays the 'Instances' page. It lists four instances: 'instance-1', 'Autoasg', 'instance-2', and 'instance-3'. The fourth instance, 'instance-3', is selected. The details for 'instance-3' are shown in the main pane.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
instance-1	i-0bebe46ca99ef79e9	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1a	ec2-13-233-142
Autoasg	i-032338e8f55d68e58	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1a	ec2-65-0-103-2
instance-2	i-0e850281f28d70d3f	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1a	ec2-65-0-132-9
instance-3	i-0130750591f9b8d82	Running	t2.micro	-	No alarms	ap-south-1b	ec2-13-234-31-1

**Instance: i-0130750591f9b8d82 (instance-3)**

**Details** | Security | Networking | Storage | Status checks | Monitoring | Tags

**Instance summary**

instance ID	i-0130750591f9b8d82 (instance-3)	Public IPv4 address	13.234.31.29   open address	Private IPv4 addresses	172.31.15.100
IPv6 address	-	Instance state	running	Public IPv4 DNS	ec2-13-234-31-29.ap-south-1.compute.amazonaws.com   open address
Hostname type	IP name: ip-172-31-15-100.ap-south-1.compute.internal	Private IP DNS name (IPv4 only)	ip-172-31-15-100.ap-south-1.compute.internal		

The same data available in instance-1 reflects in instance-3

**Conclusion:**

In this project, we explored the architecture of Autoscaling with launch templates and launch configurations. The goal of this project was to understand the different ways to use Autoscaling, and to choose the right approach for your specific needs. We learned that launch templates offer more flexibility and scalability than launch configurations, but they are also more complex to create and manage. Ultimately, the best choice for you will depend on your specific needs. If you need more flexibility and want to be able to easily update your configuration, then you should use a launch template. If you need a simpler solution and are compatible with older versions of Amazon EC2, then you should use a launch configuration. We hope that this project has been helpful in understanding the different ways to use Autoscaling. Thank you for your time!

