



BLACKBUCKS INTERNSHIP REPORT

Global Request Distribution Architecture for Webpage Access with Autoscaling, EFS and SNS Integration.

SUBMITTED BY

SAIPRANAY GANTA(20B91A05Q8)
SAMPANGI VENKATA MANIKANTA(20B91A05R0)
SIRIGIRI SAI MAHESH (20B91A05S6)

UNDER THE GUIDANCE OF MR. AASHU DEV

**Blackbuck Engineers Pvt. Ltd
Road No 36, Jubilee Hills, Hyderabad**

BLACKBUCK INTERNSHIP WORK

Team Members:

- SAIPRANAY GANTA (20B91A05Q8)
- SAMPANGI VENKATA MANIKANTA (20B91A05R0)
- SIRIGIRI SAI MAHESH (20B91A05S6)

Title:

Global Request Distribution Architecture for Webpage Access with Autoscaling, EFS and SNS Integration.

Abstract:

The architecture described above focuses on distributing user requests in a proportional manner using an Elastic Load Balancer (ELB) across multiple instances created by an Auto Scaling group. Additionally, an Elastic File System (EFS) is set up to store web pages. The instances created by the Auto Scaling group are directly connected to the EFS, allowing them to access and serve the web pages. The ELB manages the incoming requests and directs them to the appropriate instances hosting the web pages stored in the EFS. Moreover, the architecture utilizes the SNS service to send email notifications to the administrator whenever an instance is created, terminated, or encounters failures during creation or termination within the Auto Scaling group. This comprehensive solution ensures efficient distribution of user requests, seamless web page hosting, and enables timely notifications to the administrator regarding instance management events.

AWS PROJECT WORK

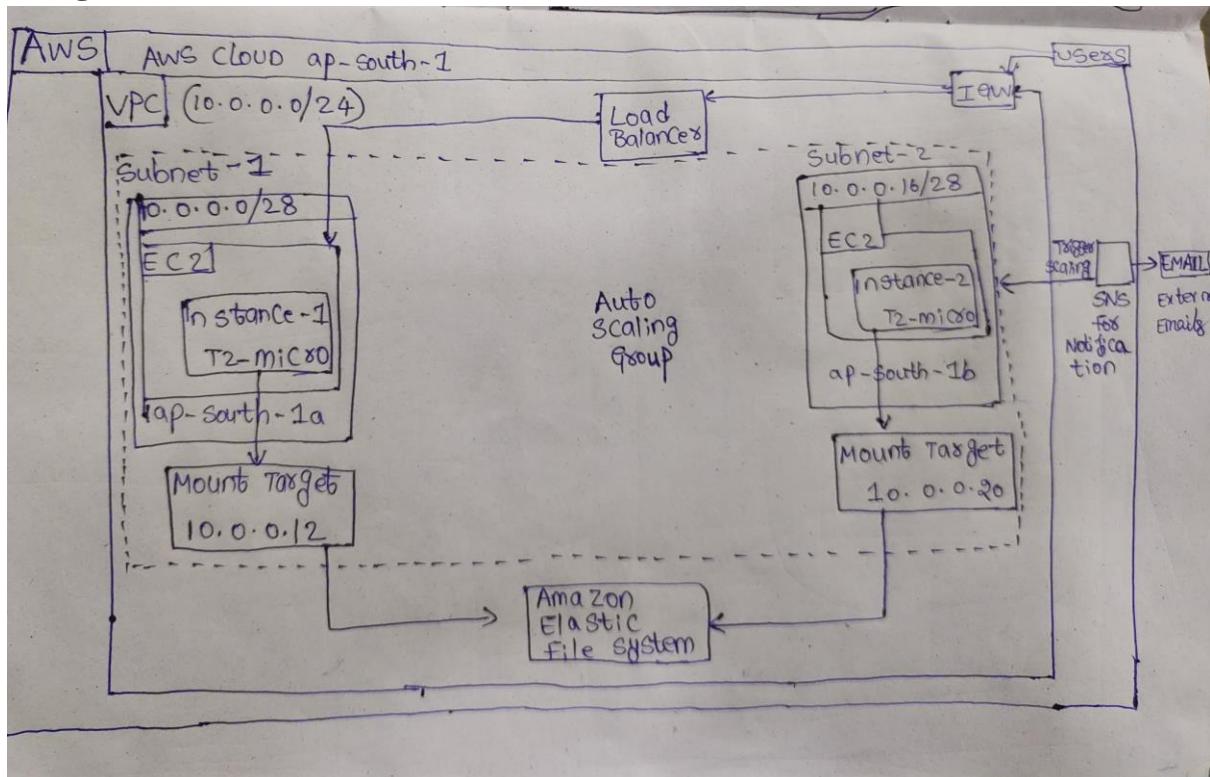
Abstract:

The architecture described above focuses on distributing user requests in a proportional manner using an Elastic Load Balancer (ELB) across multiple instances created by an Auto Scaling group. Additionally, an Elastic File System (EFS) is set up to store web pages. The instances created by the Auto Scaling group are directly connected to the EFS, allowing them to access and serve the web pages. The ELB manages the incoming requests and directs them to the appropriate instances hosting the web pages stored in the EFS. Moreover, the architecture utilizes the SNS service to send email notifications to the administrator whenever an instance is created, terminated, or encounters failures during creation or termination within the Auto Scaling group. This comprehensive solution ensures efficient distribution of user requests, seamless web page hosting, and enables timely notifications to the administrator regarding instance management events.

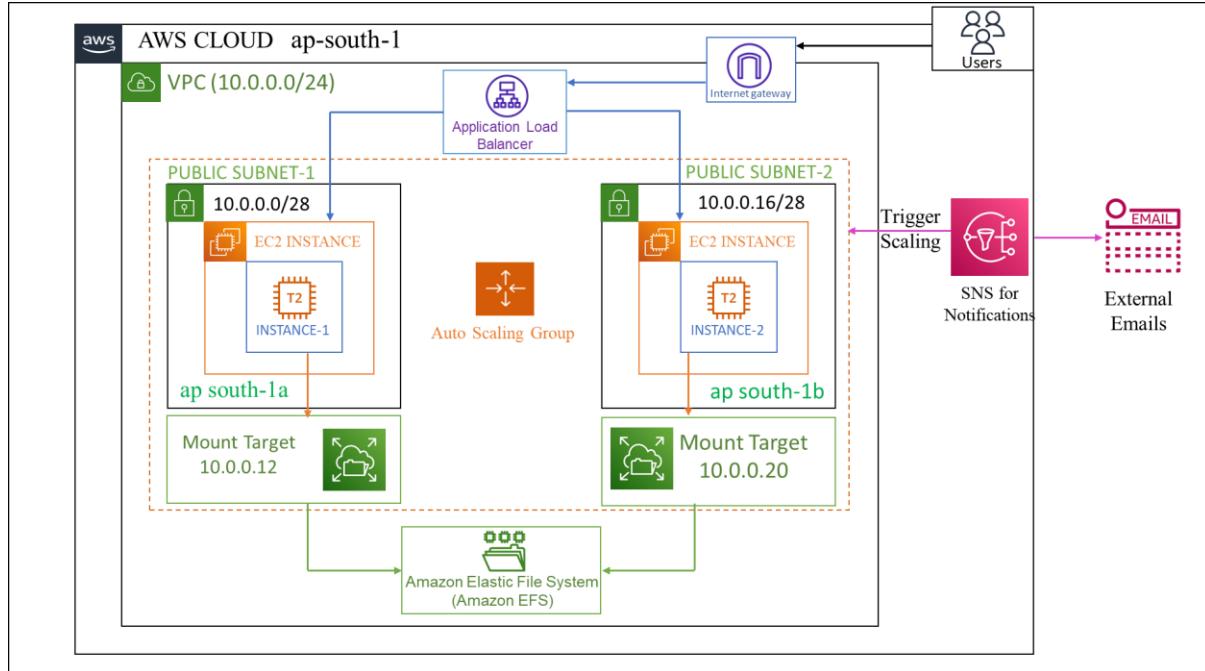
Services used:

- EC2 (Elastic Compute Cloud)
- VPC (Virtual Private Cloud)
- Amazon Elastic File System (Amazon EFS)
- Auto Scaling Group
- Application Load Balancer
- Simple Notification service (SNS)

Rough architecture:



Final architecture:



Cloud computing

Cloud computing is on-demand access, via the internet, to computing resources—applications, servers (physical servers and virtual servers), data storage, development tools, networking capabilities, and more—hosted at a remote data center managed by a cloud services provider (or CSP). The CSP makes these resources available for a monthly subscription fee or bills them according to usage.

Compared to traditional on-premises IT, and depending on the cloud services you select, cloud computing helps do the following:

- **Lower IT costs:** Cloud lets you offload some or most of the costs and effort of purchasing, installing, configuring, and managing your own on-premises infrastructure.
- **Improve agility and time-to-value:** With cloud, your organization can start using enterprise applications in minutes, instead of waiting weeks or months for IT to respond to a request, purchase and configure supporting hardware, and install software. Cloud also lets you empower certain users—specifically developers and data scientists.
- **Scale more easily and cost-effectively:** Cloud provides elasticity—instead of purchasing excess capacity that sits unused during slow periods, you can scale capacity up and down in response to spikes and dips in traffic. You can also take advantage of your cloud provider’s global network to spread your applications closer to users around the world.

The term ‘cloud computing’ also refers to the technology that makes cloud work. This includes some form of virtualized IT infrastructure—servers, operating system software, networking, and other infrastructure that’s abstracted, using special software, so that it can be pooled and divided irrespective of physical hardware boundaries. For example, a single hardware server can be divided into multiple virtual servers.

Cloud Computing Services:

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

are the three most common models of cloud services, and it’s not uncommon for an organization to use all three.

IaaS (Infrastructure-as-a-Service)

IaaS provides on-demand access to fundamental computing resources—physical and virtual servers, networking, and storage—over the internet on a pay-as-you-go basis. IaaS enables end users to scale and shrink resources on an as-needed basis, reducing the need for high, up-front capital expenditures or unnecessary on-premises or ‘owned’ infrastructure and for overbuying resources to accommodate periodic spikes in usage.

In contrast to SaaS and PaaS (and even newer PaaS computing models such as containers and serverless), IaaS provides the users with the lowest-level control of computing resources in the cloud.

IaaS was the most popular cloud computing model when it emerged in the early 2010s. While it remains the cloud model for many types of workloads, use of SaaS and PaaS is growing at a much faster rate.

PaaS (Platform-as-a-service)

PaaS provides software developers with on-demand platform—hardware, complete software stack, infrastructure, and even development tools—for running, developing, and managing applications without the cost, complexity, and inflexibility of maintaining that platform on-premises.

With PaaS, the cloud provider hosts everything—servers, networks, storage, operating system software, middleware, databases—at their data center. Developers simply pick from a

menu to ‘spin up’ servers and environments they need to run, build, test, deploy, maintain, update, and scale applications.

Today, PaaS is often built around *containers*, a virtualized compute model one step removed from virtual servers. Containers virtualize the operating system, enabling developers to package the application with only the operating system services it needs to run on any platform, without modification and without need for middleware.

SaaS (Software-as-a-Service)

SaaS—also known as cloud-based software or cloud applications—is application software that’s hosted in the cloud, and that user’s access via a web browser, a dedicated desktop client, or an API that integrates with a desktop or mobile operating system. In most cases, SaaS users pay a monthly or annual subscription fee; some may offer ‘pay-as-you-go’ pricing based on your actual usage.

In addition to the cost savings, time-to-value, and scalability benefits of cloud, SaaS offers the following:

- **Automatic upgrades:** With SaaS, users take advantage of new features as soon as the provider adds them, without having to orchestrate an on-premises upgrade.
- **Protection from data loss:** Because SaaS stores application data in the cloud with the application, users don’t lose data if their device crashes or breaks.

SaaS is the primary delivery model for most commercial software today—there are hundreds of thousands of SaaS solutions available, from the most focused industry and departmental applications to powerful enterprise software database and AI (artificial intelligence) software.

Cloud Service Providers:

- Amazon Web Services
- Microsoft Azure
- Google Cloud Platform
- Oracle
- IBM cloud
- Salesforce

Amazon Web Services:

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, computing, 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 for 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.

Why AWS?

- **Easy to use:**

AWS is designed to allow application providers, ISVs, and vendors to host your applications quickly and securely – whether an existing application or a new SaaS-based application. You can use the AWS Management Console or well-documented web services APIs to access AWS's application hosting platform.

- **Flexible:**

AWS enables you to select the operating system, programming language, web application platform, database, and other services you need. With AWS, you receive a virtual environment that lets you load the software and services your application

requires. This eases the migration process for existing applications while preserving options for building new solutions.

- **Cost-effective:**

You pay only for the compute power, storage, and other resources you use, with no long-term contracts or up-front commitments. For more information on comparing the costs of other hosting alternatives with AWS, see the AWS Economics Center.

- **Reliable:**

With AWS, you take advantage of a scalable, reliable, and secure global computing infrastructure, the virtual backbone of Amazon.com's multi-billion-dollar online business that has been honed for over a decade.

- **Scalable and High performance:**

Using AWS tools, Auto Scaling, and Elastic Load Balancing, your application can scale up or down based on demand. Backed by Amazon's massive infrastructure, you have access to compute and storage resources when you need them.

- **Secure:**

Using AWS tools, Auto Scaling, and Elastic Load Balancing, your application can scale up or down based on demand. Backed by Amazon's massive infrastructure, you have access to compute and storage resources when you need them.

AWS Services:

Amazon EC2:



Amazon Elastic Compute Cloud (EC2) is a part of Amazon.com's cloud-computing platform, Amazon Web Services (AWS), that allows users to rent virtual computers on which to run their own computer applications. EC2 encourages scalable deployment of applications by providing a web service through which a user can boot an Amazon Machine Image (AMI) to configure a virtual machine, which Amazon calls an "instance", containing any software desired. A user can create, launch, and terminate server-instances as needed, paying by the second for active servers – hence the term "elastic". EC2 provides users with control over the geographical location of instances that allows for latency optimization and high levels of redundancy. In November 2010, Amazon switched its own retail website platform to EC2 and AWS.

Amazon announced a limited public beta test of EC2 on August 25, 2006, offering access on a first-come, first-served basis. Amazon added two new instance types (Large and Extra-Large) on October 16, 2007. On May 29, 2008, two more types were added, High-CPU Medium and High-CPU Extra Large. There were twelve types of instances available.

Amazon added three new features on March 27, 2008, static IP addresses, availability zones, and user selectable kernels. On August 20, 2008, Amazon added Elastic Block Store (EBS) This provides persistent storage, a feature that had been lacking since the service was introduced.

Instance types:

Initially, EC2 used Xen virtualization exclusively. However, on November 6, 2017, Amazon announced the new C5 family of instances that were based on a custom architecture around the KVM hypervisor, called Nitro. Each virtual machine, called an "instance", functions as a virtual private server. Amazon sizes instances based on "Elastic Compute Units". The performance of otherwise identical virtual machines may vary. On November 28, 2017, AWS announced a bare-metal instance type offering marking a remarkable departure from exclusively offering virtualized instance types.

As of January 2019, the following instance types were offered:

- General Purpose: A1, T3, T2, M5, M5a, M4, T3a
- Compute Optimized: C5, C5n, C4
- Memory Optimized: R5, R5a, R4, X1e, X1, High Memory, z1d
- Accelerated Computing: P3, P2, G3, F1
- Storage Optimized: H1, I3, D2

As of April 2018, the following payment methods by instance were offered:

- On-demand: pay by the hour without commitment.
- Reserved: rent instances with one-time payment receiving discounts on the hourly charge.

- Spot: bid-based service runs the jobs only if the spot price is below the bid specified by bidder. The spot price is claimed to be supply-demand based, however a 2011 study concluded that the price was generally not set to clear the market but was dominated by an undisclosed reserve price.

Amazon EBS (Elastic Block Store):



Amazon Elastic Block Store (Amazon EBS) provides block level storage volumes for use with EC2 instances. EBS volumes behave like raw, unformatted block devices. You can mount these volumes as devices on your instances. EBS volumes that are attached to an instance are exposed as storage volumes that persist independently from the life of the instance. You can create a file system on top of these volumes or use them in any way you would use a block device (such as a hard drive). You can dynamically change the configuration of a volume attached to an instance.

We recommend Amazon EBS for data that must be quickly accessible and requires long-term persistence. EBS volumes are particularly well-suited for use as the primary storage for file systems, databases, or for any applications that require fine granular updates and access to raw, unformatted, block-level storage. Amazon EBS is well suited to both database-style applications that rely on random reads and writes, and to throughput-intensive applications that perform long, continuous reads and writes.

With Amazon EBS, you pay only for what you use. For more information about Amazon EBS pricing, see the Projecting Costs Section of the Amazon Elastic Block Store page.

Features of Amazon EBS

- You create an EBS volume in a specific Availability Zone, and then attach it to an instance in that same Availability Zone. To make a volume available outside of the Availability Zone, you can create a snapshot and restore that snapshot to a new volume anywhere in that Region. You can copy snapshots to other Regions and then restore them to new volumes there, making it easier to leverage multiple AWS Regions for geographical expansion, data center migration, and disaster recovery.

- Amazon EBS provides the following volume types: General Purpose SSD, Provisioned IOPS SSD, Throughput Optimized HDD, and Cold HDD. For more information, see [EBS volume types](#).

The following is a summary of performance and use cases for each volume type.

- General Purpose SSD volumes (gp2 and gp3) balance price and performance for a wide variety of transactional workloads. These volumes are ideal for use cases such as boot volumes, medium-size single instance databases, and development and test environments.
 - Provisioned IOPS SSD volumes (io1 and io2) are designed to meet the needs of I/O-intensive workloads that are sensitive to storage performance and consistency. They provide a consistent IOPS rate that you specify when you create the volume. This enables you to predictably scale to tens of thousands of IOPS per instance. Additionally, io2 volumes provide the highest levels of volume durability.
 - Throughput Optimized HDD volumes (st1) provide low-cost magnetic storage that defines performance in terms of throughput rather than IOPS. These volumes are ideal for large, sequential workloads such as Amazon EMR, ETL, data warehouses, and log processing.
 - Cold HDD volumes (sc1) provide low-cost magnetic storage that defines performance in terms of throughput rather than IOPS. These volumes are ideal for large, sequential, cold-data workloads. If you require infrequent access to your data and are looking to save costs, these volumes provide inexpensive block storage.
-
- You can create your EBS volumes as encrypted volumes, in order to meet a wide range of data-at-rest encryption requirements for regulated/audited data and applications. When you create an encrypted EBS volume and attach it to a supported instance type, data stored at rest on the volume, disk I/O, and snapshots created from the volume are all encrypted. Encryption occurs on the servers that host EC2 instances, providing encryption of data-in-transit from EC2 instances to EBS storage. For more information, see [Amazon EBS encryption](#).
 - Performance metrics, such as bandwidth, throughput, latency, and average queue length, are available through the AWS Management Console. These metrics, provided by Amazon CloudWatch, allow you to monitor the performance of your volumes to make sure that you are providing enough performance for your applications without paying for resources you don't need.

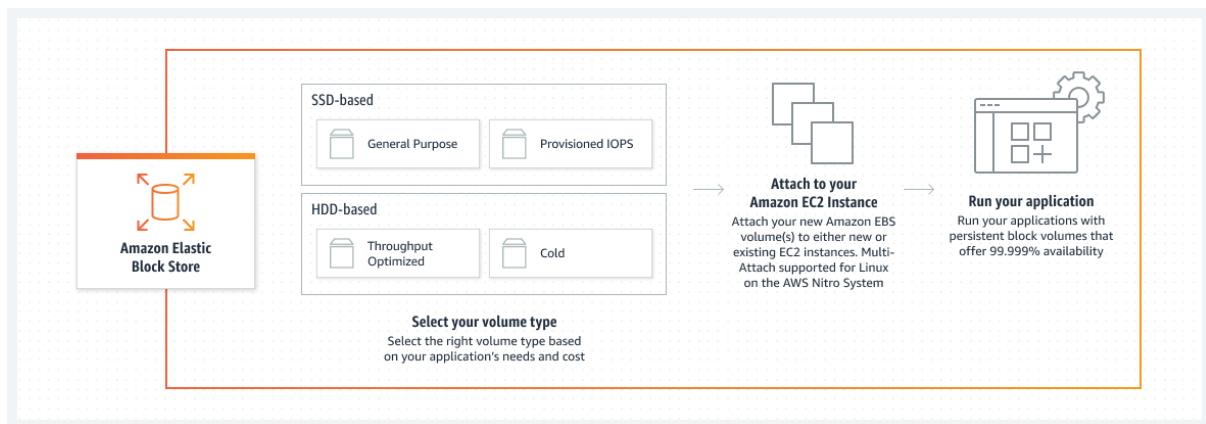
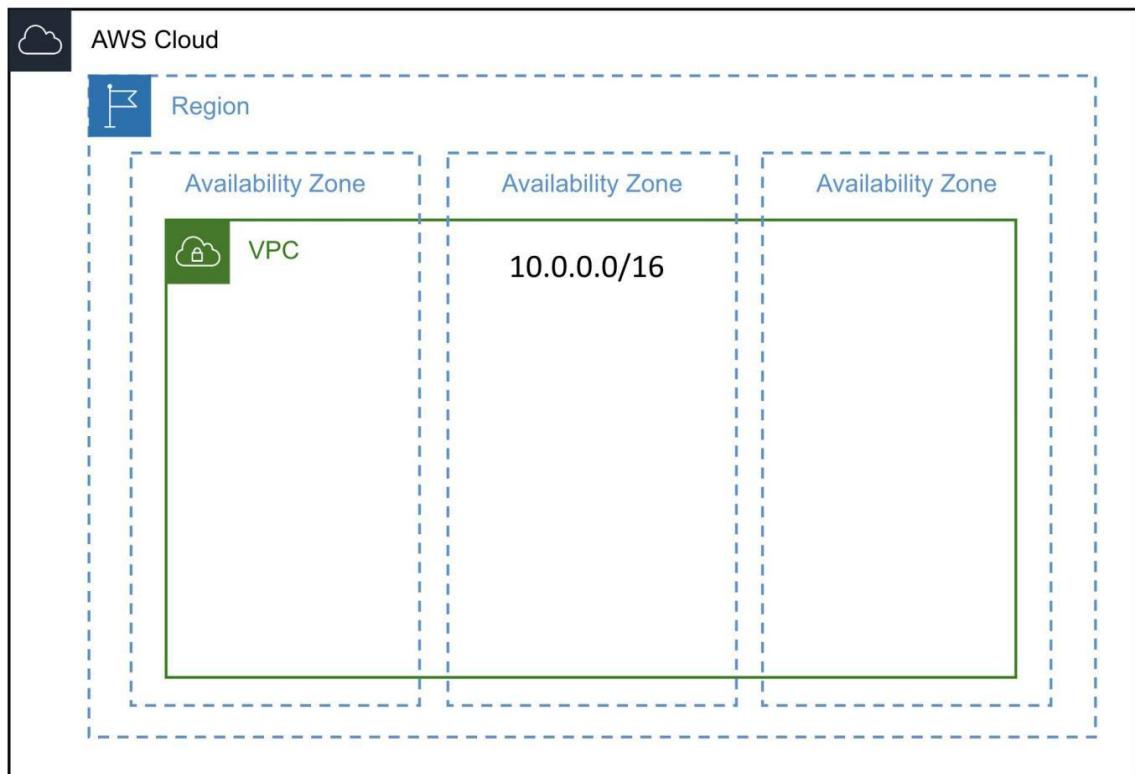


Fig. High-Performance Block Storage

Amazon VPC:



Amazon Virtual Private Cloud (VPC) is a commercial cloud computing service that provides a virtual private cloud, by provisioning a logically isolated section of Amazon Web Services (AWS) Cloud. Enterprise customers are able to access the Amazon Elastic Compute Cloud (EC2) over an IPsec based virtual private network. Unlike traditional EC2 instances which are allocated internal and external IP numbers by Amazon, the customer can assign IP numbers of their choosing from one or more subnets.



Amazon Web Services launched Amazon Virtual Private Cloud on 26 August 2009, which allows the Amazon Elastic Compute Cloud service to be connected to legacy infrastructure over an IPsec VPN. In AWS, the basic VPC is free to use, with users being charged by usage for additional features. EC2 and RDS instances running in a VPC can also be purchased using Reserved Instances, however will have a limitation on resources being guaranteed. [citation needed]

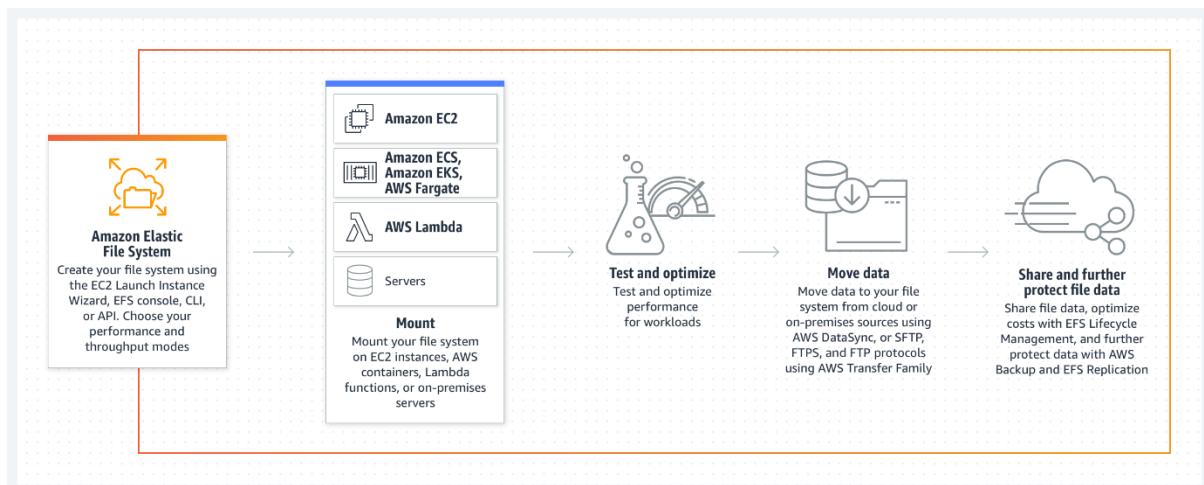
IBM Cloud launched IBM Cloud VPC on 4 June 2019, provides an ability to manage virtual machine-based compute, storage, and networking resources. Pricing for IBM Cloud Virtual Private Cloud is applied separately for internet data transfer, virtual server instances, and block storage used within IBM Cloud VPC.

Google Cloud Platform resources can be provisioned, connected, and isolated in a virtual private cloud (VPC) across all GCP regions. With GCP, VPCs are global resources and subnets within that VPC are regional resources. This allows users to connect zones and regions without the use of additional networking complexity as all data travels, encrypted in transit and at rest, on Google's own global, private network. Identity management policies and security rules allow for private access to Google's storage, big data, and analytics managed services. VPCs on Google Cloud Platform leverage the security of Google's data centers.

Elastic File System:



Amazon Elastic File System (EFS) is a fully managed, scalable, and highly available file storage service provided by Amazon Web Services (AWS). EFS is designed to provide shared file storage for EC2 instances, allowing multiple instances to access and modify files concurrently. It offers a simple and scalable solution for applications that require shared file storage, such as content management systems, web servers, and Big Data analytics. EFS automatically scales storage capacity as needed, without any upfront provisioning, ensuring that applications have the required storage resources to handle growing workloads. It supports the Network File System (NFS) protocol, making it compatible with a wide range of Linux-based applications and services. EFS provides high durability and availability, replicating data across multiple availability zones within a region, thereby protecting against data loss and ensuring continuous access to files. With its elastic scalability, ease of use, and strong reliability, EFS simplifies file storage management and provides a flexible solution for shared file access in AWS environments.



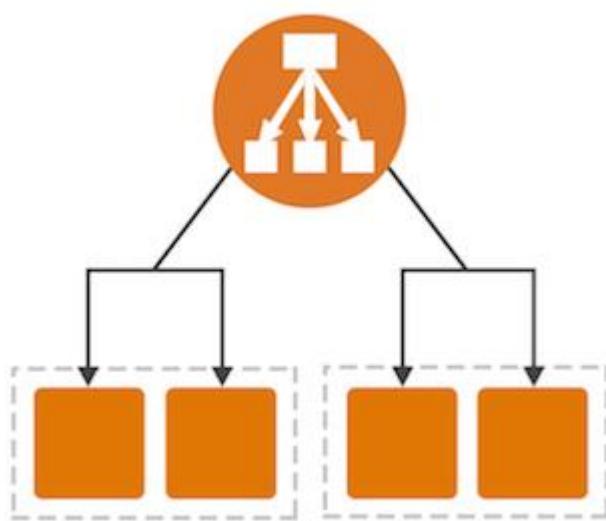
Load Balancer:



AWS Elastic Load Balancing

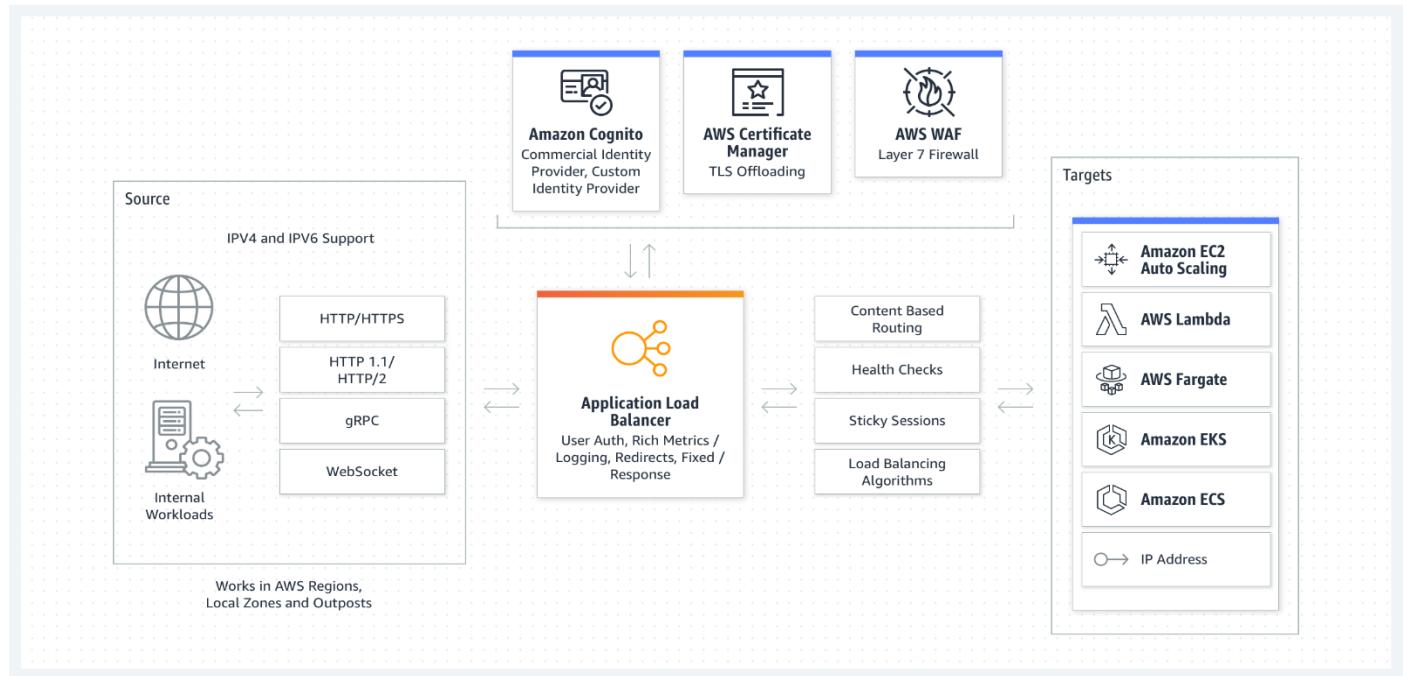
Load Balancer in AWS is a service that evenly distributes incoming network traffic across multiple instances or services, ensuring high availability and optimal resource utilization. With Classic Load Balancer (CLB) and Application Load Balancer (ALB) options, it can operate at the transport or application layer, respectively. Load balancers automatically handle traffic routing, detect and redirect requests from unhealthy instances to healthy ones, and provide advanced features like content-based routing and SSL termination. By distributing traffic and providing redundancy, load balancers enhance scalability, availability, and performance of applications deployed in AWS, making them a crucial component for achieving reliable and efficient workload management.

Application Load Balancer:

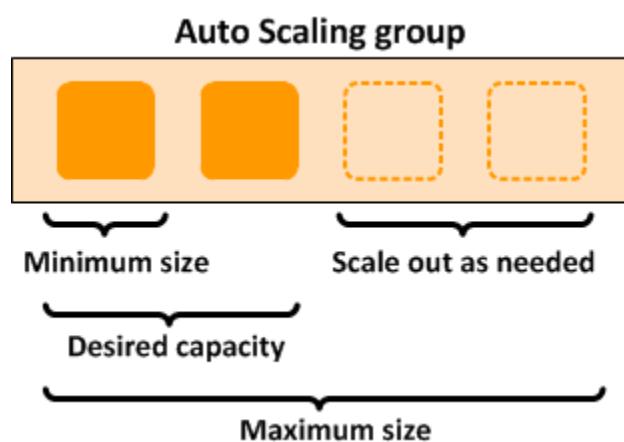


The Application Load Balancer (ALB) is an advanced load balancing solution provided by Amazon Web Services (AWS). It operates at the application layer (Layer 7) of the network stack, allowing it to intelligently route traffic based on content and application-specific

requirements. ALB supports HTTP and HTTPS protocols, enabling it to handle web traffic and provide secure connections. It offers features like content-based routing, allowing requests to be directed to different backend services based on specific URL patterns or headers. ALB also supports advanced health checks, SSL/TLS termination, and sticky sessions for session persistence. With its flexible and powerful routing capabilities, ALB is an essential component for building highly scalable, fault-tolerant, and performant applications in AWS.



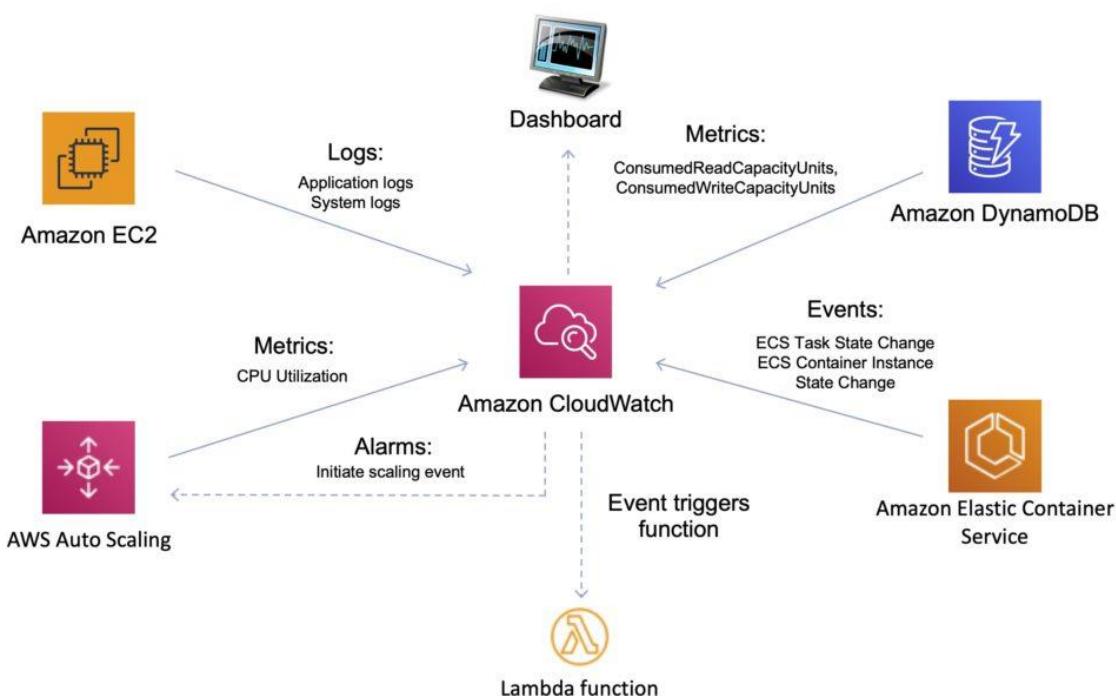
AWS Autoscaling:



Amazon EC2 Auto Scaling helps you ensure that you have the correct number of Amazon EC2 instances available to handle the load for your application. You create collections of EC2 instances, called *Auto Scaling groups*. You can specify the minimum number of instances in each Auto Scaling group, and Amazon EC2 Auto Scaling ensures that your group never goes below this size. You can specify the maximum number of instances in each Auto Scaling group, and Amazon EC2 Auto Scaling ensures that your group never goes above this size. If you specify the desired capacity, either when you create the group or at any time thereafter, Amazon EC2 Auto Scaling ensures that your group has this many instances. If you specify scaling policies, then Amazon EC2 Auto Scaling can launch or terminate instances as demand on your application increases or decreases.

For example, the following Auto Scaling group has a minimum size of one instance, a desired capacity of two instances, and a maximum size of four instances. The scaling policies that you define adjust the number of instances, within your minimum and maximum number of instances, based on the criteria that you specify.

Amazon CloudWatch:



Amazon CloudWatch monitors your Amazon Web Services (AWS) resources and the applications you run on AWS in real time. You can use CloudWatch to collect and track metrics, which are variables you can measure for your resources and applications.

The CloudWatch home page automatically displays metrics about every AWS service you use. You can additionally create custom dashboards to display metrics about your custom applications and display custom collections of metrics that you choose.

You can create alarms that watch metrics and send notifications or automatically make changes to the resources you are monitoring when a threshold is breached. For example, you can monitor the CPU usage and disk reads and writes of your Amazon EC2 instances and then use that data to determine whether you should launch additional instances to handle increased load. You can also use this data to stop underused instances to save money.

With CloudWatch, you gain system-wide visibility into resource utilization, application performance, and operational health.

AWS Simple Notification Service :



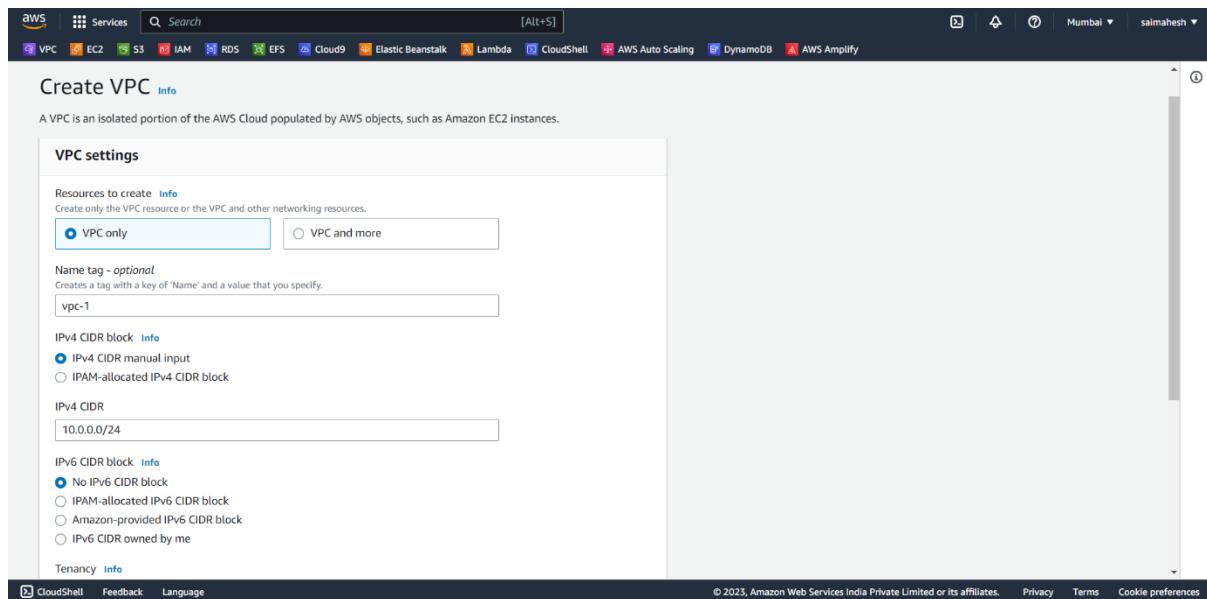
The Simple Notification Service (SNS) in AWS is a fully managed messaging service that enables the sending and receiving of notifications and messages. SNS provides a highly scalable and flexible solution for pub/sub messaging, allowing publishers to send messages to multiple subscribers. It supports various communication protocols, including HTTP, email, SMS, and mobile push notifications, ensuring seamless delivery across multiple channels. SNS simplifies the process of building distributed systems by decoupling components and enabling asynchronous communication. It offers features like message filtering, topic-based message filtering, and message attributes for granular control over message delivery. With its ease of use, scalability, and versatility, SNS is a reliable service for building event-driven architectures and real-time messaging applications in AWS.



IMPLEMENTATION

Steps to perform:

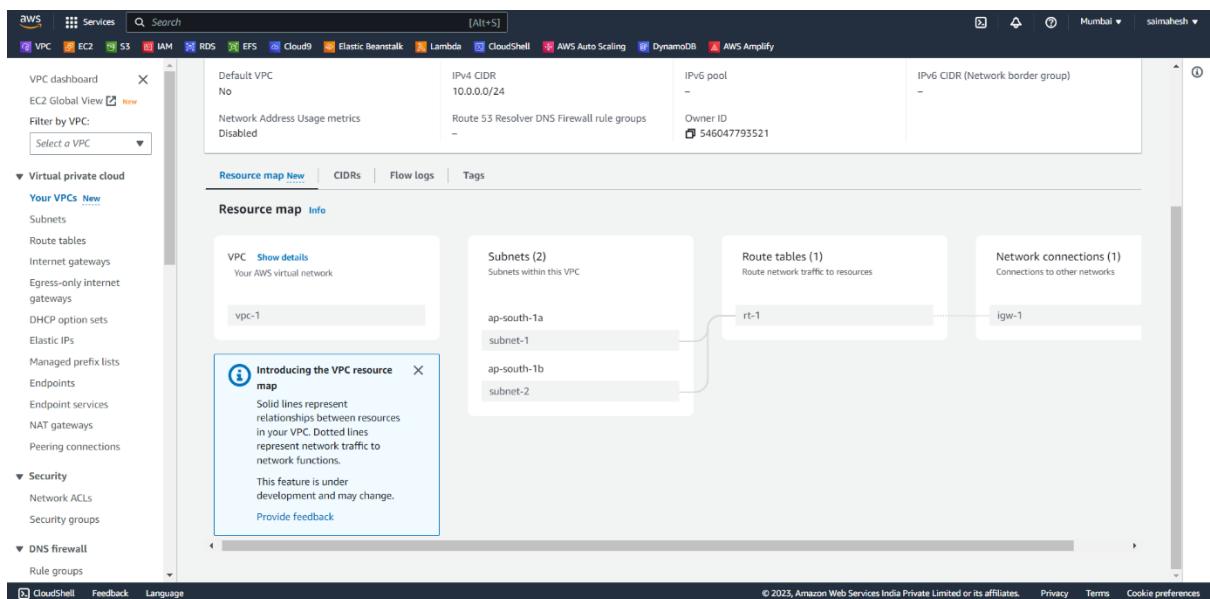
VPC: Creating a custom VPC



Give a name to the Custom VPC and an IPV4 address

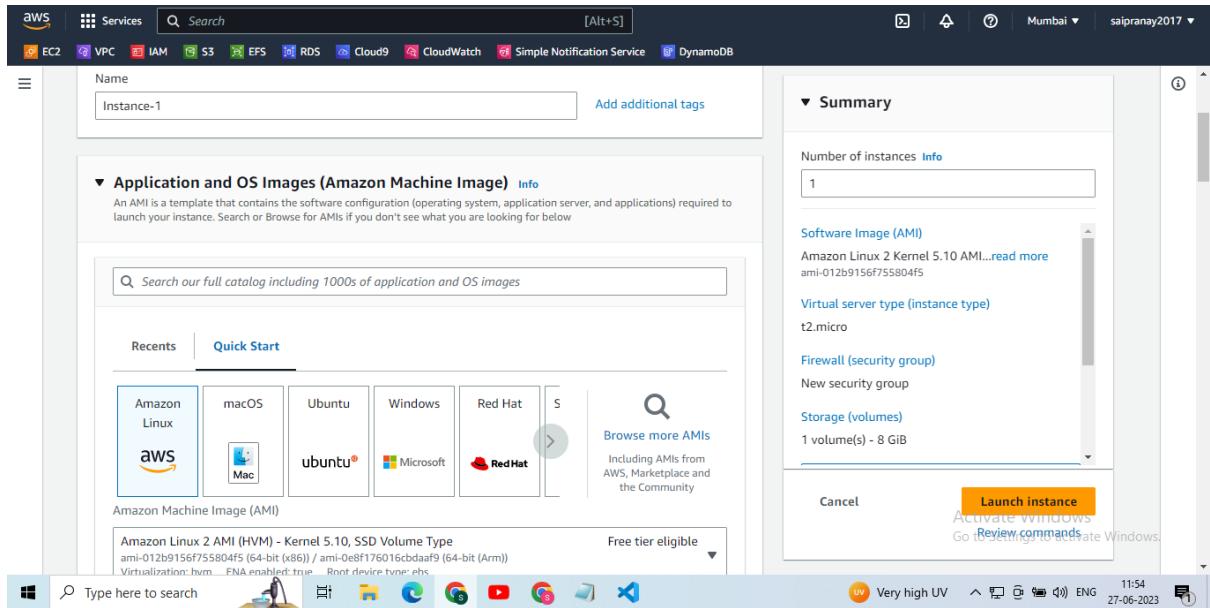
The screenshot shows the AWS VPC Details page for a newly created VPC. The VPC ID is `vpc-0ec62dc2eb8179800`. The state is **Available**. The Default VPC is set to **No**. The IPv4 CIDR is `10.0.0.0/24`. The Network Address Usage metrics are **Disabled**. The DNS resolution is **Enabled**. The Main route table is `rtb-09c929ec94f1778f3 / rt-1`. The IPv6 pool is empty. The Route 53 Resolver DNS Firewall rule groups are empty. The Owner ID is `546047793521`. The Main network ACL is `acl-0c2d56615dd9d57c1`. The IPv6 CIDR (Network border group) is empty.

Now the VPC is created

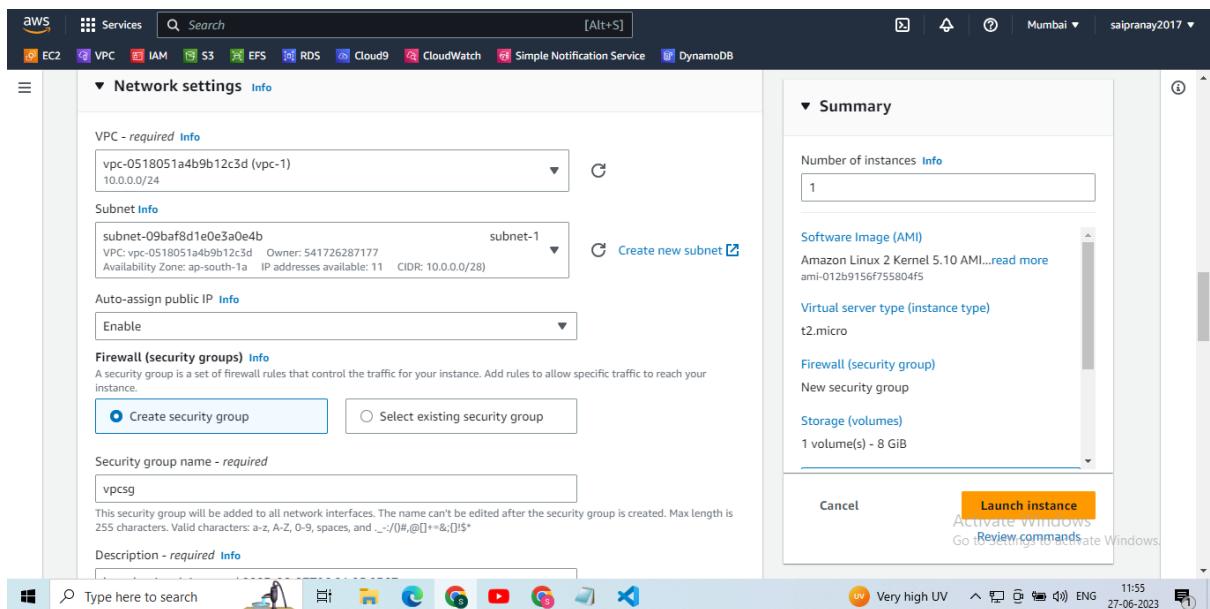


This is the flow diagram of how a vpc and the subnets created are connected to VPC and the VPC connected to the internet gateway.

EC2:



Giving name to the instance and choosing the AMI



Selecting the Custom VPC - the VPC we created in the above steps
Selecting the subnet-1 present in the availability zone 1a
Creating a new security group called - vpcsg

Security group rule 1 (TCP, 22, Anywhere)

- Type: ssh
- Protocol: TCP
- Port range: 22
- Source type: Anywhere
- Description: e.g. SSH for admin desktop

Security group rule 2 (TCP, 2049, Anywhere)

- Type: NFS
- Protocol: TCP
- Port range: 2049
- Source type: Anywhere
- Description: e.g. SSH for admin desktop

Security group rule 3 (TCP, 80, Anywhere)

- Type: HTTP
- Protocol: TCP
- Port range: 80
- Source type: Anywhere

Summary

- Number of instances: 1
- Software Image (AMI): Amazon Linux 2 Kernel 5.10 AMI... (read more)
- Virtual server type (instance type): t2.micro
- Firewall (security group): New security group
- Storage (volumes): 1 volume(s) - 8 GiB

Buttons:

- Cancel
- Launch instance
- Activate Windows
- Review commands

Adding rules:

1. SSH
2. All traffic
3. NFS - to mount the Elastic File System
4. HTTP - to host the web page

Instances (1) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status
Instance-1	i-0c5d01bb1ac7dbedd	Running	t2.micro	2/2 checks passed	No alarms

Select an instance

Activate Windows
Go to Settings to activate Windows.

The instance is created now on hitting the launch instance button.

EFS:

The screenshot shows the 'File system settings' step of the EFS creation wizard. The 'General' tab is selected. The 'Name' field contains 'efs-1'. The 'Storage class' section has 'Standard' selected (radio button is checked). Below it, there's a note about automatic backups and a warning message: '⚠ We recommend that you create a backup policy for your file system'. Under 'Lifecycle management', it says 'EFS Intelligent-Tiering uses Lifecycle Management to automatically achieve the right price and performance blend for your application by moving your files between the Standard and Standard-Infrequent Access storage classes.' The 'Transition into IA' dropdown is set to '30 day(s) since last access'. The 'Transition out of IA' dropdown is set to 'None'. The bottom of the screen shows the AWS navigation bar.

First we create name of EFS and rest constrains as set to default.

The screenshot shows the 'Network access' step of the EFS creation wizard. The 'Network' section is active. It shows a VPC selection dropdown with 'vpc-0ec62dc2eb8179800' and 'vpc-1'. The 'Mount targets' section lists two availability zones: 'ap-south-1a' and 'ap-south-1b'. Each zone has a corresponding subnet ID: 'subnet-075c0007aff...' and 'subnet-0db019290...'. Security groups are assigned to each mount target: 'sg-058589d6257f49783' for ap-south-1a and 'sg-058589d6257f49783' for ap-south-1b. A 'Remove' button is available for each security group assignment. At the bottom, there's a 'Add mount target' button. The bottom of the screen shows the AWS navigation bar.

Now we Select Network access i.e VPC and Mount Target.

The screenshot shows the 'Review and create' step for creating an EFS file system. On the left, a sidebar lists steps: Step 1 File system settings, Step 2 Network access, Step 3 - optional File system policy, and Step 4 Review and create. The main area is titled 'Review and create' and 'Step 1: File system settings'. It contains a table titled 'File system' with the following data:

Field	Value	Is editable?
Name	efs-1	Yes
Performance mode	General Purpose	No
Throughput mode	Elastic	Yes
Encrypted	Yes	No
KMS Key ID	-	No
Lifecycle management	Transition into IA: 30 day(s) since last access Transition out of IA: None	Yes
Automatic backups	No	Yes
VPC ID	vpc-0ec62dc2eb8179800 (vpc-1)	Yes
Availability Zone	Standard	No

Below the table is a 'Tags' section with a placeholder 'Add tag' and a 'Create file system' button.

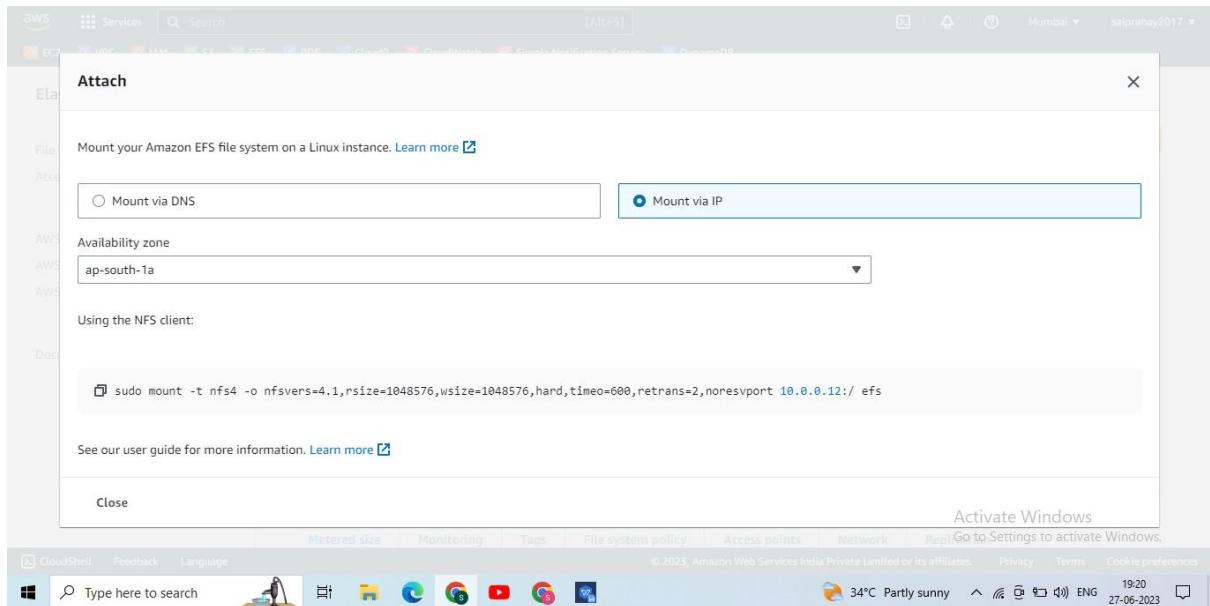
Reviewing and creating the EFS

The screenshot shows the 'Amazon EFS > File systems' page. The left sidebar has links for 'File systems', 'Access points', 'AWS Backup', 'AWS DataSync', 'AWS Transfer', and 'Documentation'. The main area shows a table titled 'File systems (1)' with one entry:

Name	File system ID	Encrypted	Total size	Size in Standard / One Zone	Size in Standard-IA / One Zone-IA	Provisioned Throughput (MiB/s)	File system state	Creation time
efs-1	fs-0014a0765ef63a9c3	Encrypted	6.00 KiB	6.00 KiB	0 Bytes	-	Available	Tue, 27 2023 05:46:00 GMT

At the bottom of the table are buttons for 'View details', 'Delete', and 'Create file system'.

The EFS File System is created now.



Copy the mount url to mount the file system on EC2

```

Installed:
httpd.x86_64 0:2.4.57-1.amzn2

Dependency Installed:
apr.x86_64 0:1.7.2-1.amzn2
generic-logos-httd.noarch 0:18.0.0-4.amzn2
mailcap.noarch 0:2.1.41-2.amzn2

Complete!
[root@ip-10-0-0-4 ec2-user]# sudo service httpd start
Redirecting to /bin/systemctl start httpd.service
[root@ip-10-0-0-4 ec2-user]# mkdir /var/www/html/efs-mount-point
[root@ip-10-0-0-4 ec2-user]# sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport 10.0.0.12:/ /var/www/html/efs-mount-point
[root@ip-10-0-0-4 ec2-user]# cd /var/www/html/efs-mount-point
[root@ip-10-0-0-4 efs-mount-point]# vi aws.html
[root@ip-10-0-0-4 efs-mount-point]# history
 1  sudo yum -y install httpd
 2  sudo service httpd start
 3  mkdir /var/www/html/efs-mount-point
 4  sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport 10.0.0.12:/ /var/www/html/efs-mount-point
 5  cd /var/www/html/efs-mount-point
 6  vi aws.html
 7  history
[root@ip-10-0-0-4 efs-mount-point]# 

i-0c5d01bb1ac7dbedd (Instance-1)
PublicIPs: 43.205.138.9 PrivateIPs: 10.0.0.4

```

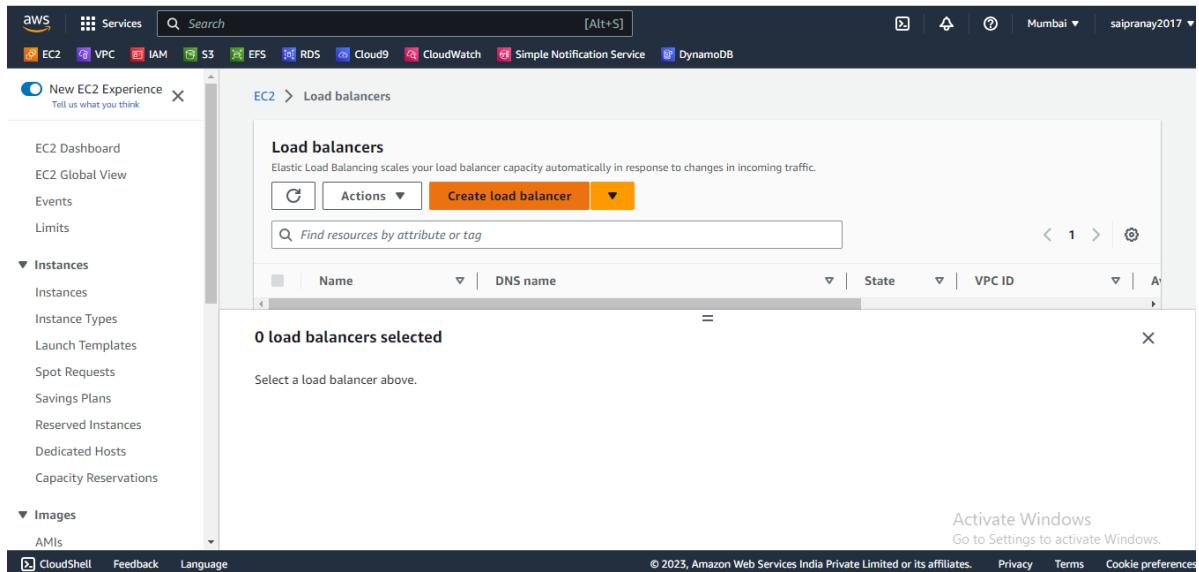
Mounting File System on EC2:

Using the above commands :

we can mount the file system we created to the ec2 instance.

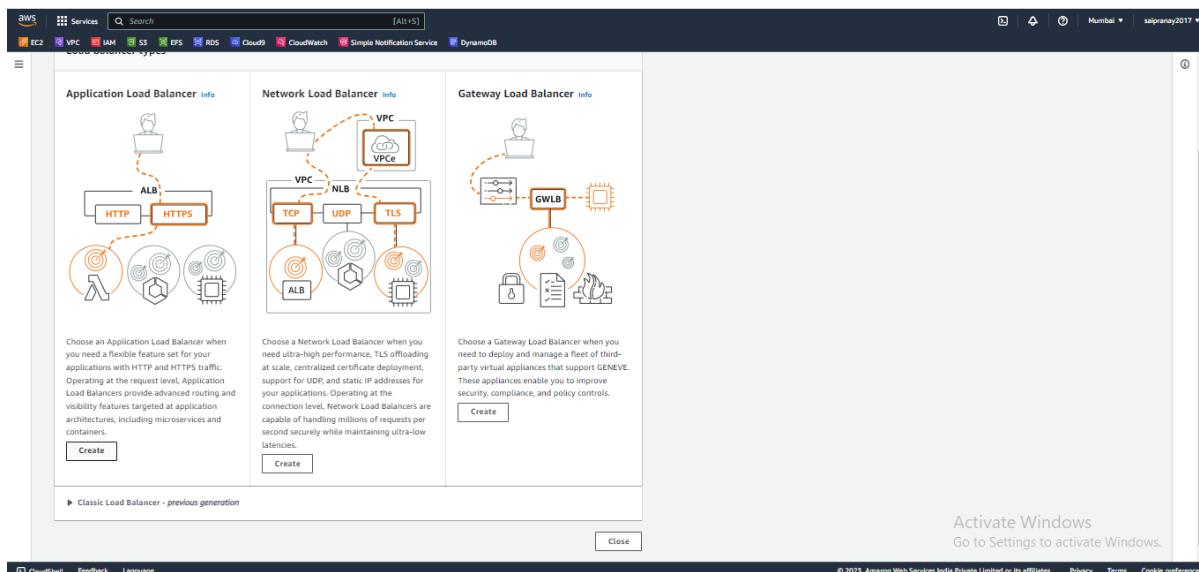
Now a HTML file aws.html is created which is present in the file system we created.

Load Balancer:



The screenshot shows the AWS EC2 dashboard with the 'Load balancers' section selected. The top navigation bar includes services like EC2, VPC, IAM, S3, RDS, Cloud9, CloudWatch, Simple Notification Service, and DynamoDB. A search bar and a user profile 'saipranay2017' are at the top right. On the left, a sidebar lists EC2-related options such as Instances, Images, and AMIs. The main content area is titled 'Load balancers' and contains a table header for 'Name', 'DNS name', 'State', 'VPC ID', and 'Actions'. Below the table, it says '0 load balancers selected' and 'Select a load balancer above.' There is also an 'Activate Windows' message at the bottom.

Now to create a load balancer click on create load balancer.



The screenshot shows the 'Create Load Balancer' dialog box with three tabs: 'Application Load Balancer', 'Network Load Balancer', and 'Gateway Load Balancer'. Each tab has a diagram and a brief description. The 'Application Load Balancer' tab shows a flow from a client to an ALB, then to Lambda functions, and finally to EC2 instances. It supports HTTP and HTTPS. The 'Network Load Balancer' tab shows a flow from a client to an NLB, then to a VPC, and finally to EC2 instances. It supports TCP, UDP, and TLS. The 'Gateway Load Balancer' tab shows a flow from a client to a GWLB, which then routes traffic to various services like a database, file storage, and security. Each tab has a 'Create' button at the bottom. At the bottom of the dialog, there is a link to 'Classic Load Balancer - previous generation' and a 'Close' button.

Creating an application load balancer as it is an elastic load balancer. It can divide the requests proportionally for also the new instances created by the auto scaling group.

Load balancer name
Name must be unique within your AWS account and can't be changed after the load balancer is created.
ELB-1

Scheme [Info](#)
Scheme can't be changed after the load balancer is created.
 Internet-facing
An internet-facing load balancer routes requests from clients over the internet to targets. Requires a public subnet. [Learn more](#)
 Internal
An internal load balancer routes requests from clients to targets using private IP addresses.

IP address type [Info](#)
Select the type of IP addresses that your subnets use.
 IPv4
Recommended for internal load balancers.
 Dualstack
Includes IPv4 and IPv6 addresses.

Activate Windows
Go to Settings to activate Windows.

Giving name to the application load balancer and keeping the rest of the things as default.

Network mapping [Info](#)
The load balancer routes traffic to targets in the selected subnets, and in accordance with your IP address settings.

VPC [Info](#)
Select the virtual private cloud (VPC) for your targets or you can [create a new VPC](#). Only VPCs with an internet gateway are enabled for selection. The selected VPC can't be changed after the load balancer is created. To confirm the VPC for your targets, view your [target groups](#).

vpc-1
vpc-0518051a4b9b12c3d
IPv4: 10.0.0.0/24

Mappings [Info](#)
Select at least two Availability Zones and one subnet per zone. The load balancer routes traffic to targets in these Availability Zones only. Availability Zones that are not supported by the load balancer or the VPC are not available for selection.

ap-south-1a (aps1-az1)

Subnet
subnet-09baf8d1e0e3a0e4b

IPv4 address
Assigned by AWS

Activate Windows
Go to Settings to activate Windows.

Selecting the custom VPC, sub-nets.

The screenshot shows the AWS CloudFront console. At the top, there's a navigation bar with the AWS logo, a search bar, and links for Services, Mumbai, and saipranay2017. Below the navigation bar, there's a sidebar with a tree view and a main content area. In the main content area, there's a section titled 'ap-south-1b (aps1-az3)' which includes fields for Subnet (subnet-06cee25c0198f668b) and IPv4 address (Assigned by AWS). Below this, there's a 'Security groups' section with a dropdown menu showing 'Select up to 5 security groups' and a list containing 'vpcsg sg-047c3890b23e84ccb X' and 'VPC: vpc-0518051a4b9b12c3d'. At the bottom of the page, there are links for CloudShell, Feedback, Language, and a footer with copyright information.

Selecting the security group that we created during the instance launch.

The screenshot shows the AWS CloudFront console. At the top, there's a navigation bar with the AWS logo, a search bar, and links for Services, Mumbai, and saipranay2017. Below the navigation bar, there's a sidebar with a tree view and a main content area. In the main content area, there's a section titled 'Listeners and routing' with a sub-section for 'Listener HTTP:80'. This section includes fields for Protocol (HTTP), Port (80), and Default action (Forward to 'vpcsg sg-047c3890b23e84ccb'). There are also sections for Listener tags (optional) and a 'Create target group' link. At the bottom of the page, there are links for CloudShell, Feedback, Language, and a footer with copyright information.

Creating a target group

The screenshot shows the AWS CloudFront console with the following interface elements:

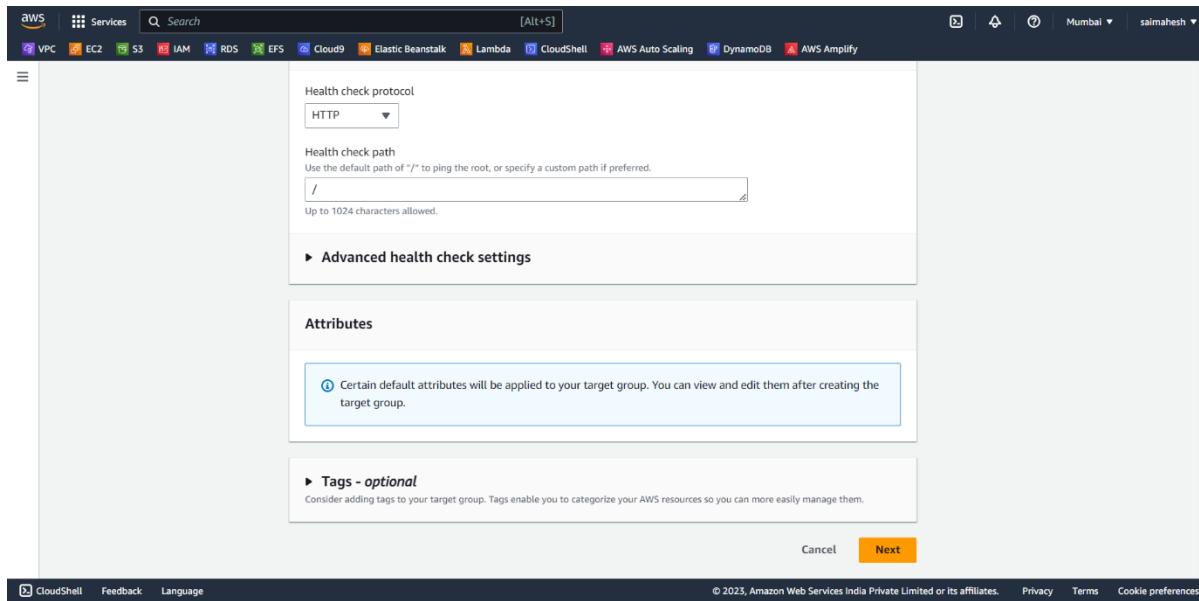
- Top Bar:** AWS logo, Services menu, Search bar, [Alt+S] key shortcut, Mumbai region, and user saipranay2017.
- Left Sidebar:** Step 1: Specify group details; Step 2: Register targets.
- Main Content Area:**
 - Section Title:** Basic configuration
 - Description:** Settings in this section can't be changed after the target group is created.
 - Target Type Selection:** A radio button labeled "Instances" is selected. Other options include "IP addresses" and "Lambda function".
 - Instances Description:** Supports load balancing to instances within a specific VPC and facilitates the use of Amazon EC2 Auto Scaling to manage and scale your EC2 capacity.
 - IP addresses Description:** Supports load balancing to VPC and on-premises resources, facilitates routing to multiple IP addresses and network interfaces on the same instance, offers flexibility with microservice based architectures, and supports IPv6 targets.
 - Lambda function Description:** Facilitates routing to a single Lambda function.
- Bottom Bar:** CloudShell, Feedback, Language, © 2023, Amazon Web Services India Private Limited or its affiliates., Privacy, Terms, Cookie preferences.

Selecting the target type - Instances

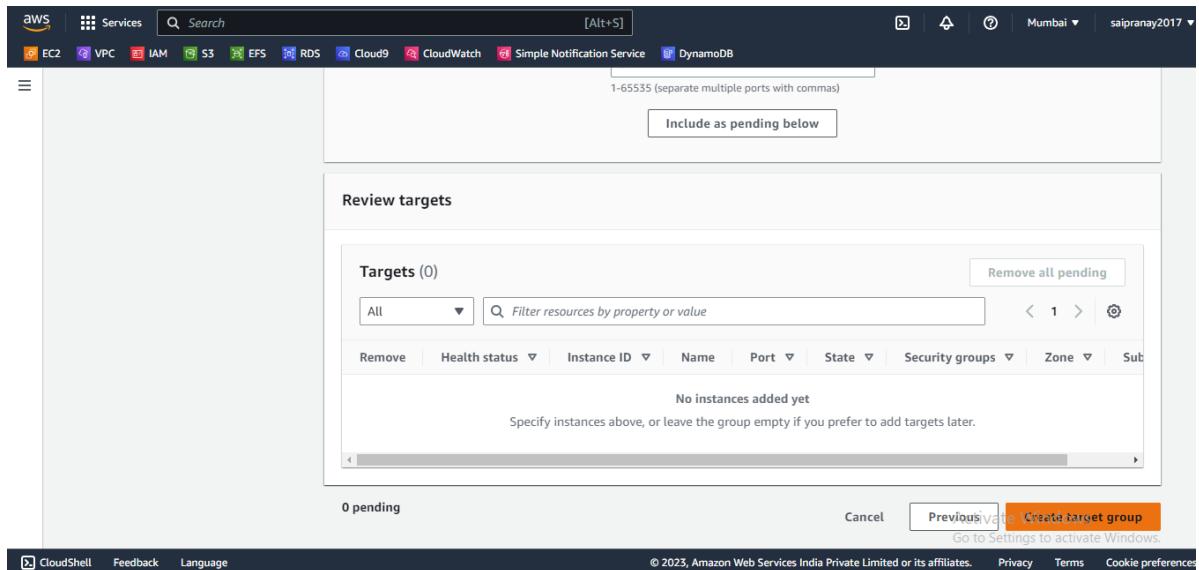
The screenshot shows the AWS CloudFront console with the following interface elements:

- Top Bar:** AWS logo, Services menu, Search bar, [Alt+S] key shortcut, Mumbai region, and user saipranay2017.
- Left Sidebar:** Application Load Balancer.
- Main Content Area:**
 - Target group name:** ProjectTarget
 - Protocol:** HTTP (selected) and Port: 80
 - VPC:** Select the VPC with the instances that you want to include in the target group. The dropdown shows "vpc-1" and its details: vpc-0518051a4b9b12c3d, IPv4: 10.0.0.0/24.
 - Protocol version:** HTTP1.1 (selected)
- Bottom Bar:** CloudShell, Feedback, Language, © 2023, Amazon Web Services India Private Limited or its affiliates., Privacy, Terms, Cookie preferences.

- Giving the target group name.
- Selecting the custom VPC.



Keeping rest of the things as default click on next



Now click on create the target group.

The screenshot shows the AWS EC2 Target Groups page. At the top, a green banner says "Successfully created target group: ProjectTarget". Below it, the "Target groups (1) Info" section displays a table with one row for "ProjectTarget". The table columns are Name, ARN, Port, Protocol, Target type, and Load balancer. The "Load balancer" column shows "None associated". A message below the table says "0 target groups selected" and "Select a target group above." At the bottom right of the page, there are links for "CloudShell", "Feedback", and "Language", along with copyright information and privacy terms.

Now the target group is created successfully
Select this target group to create the load balancer

The screenshot shows the "Create load balancer" wizard, Step 1: Set Load Balancer Properties. It has several sections: "Load balancer tags - optional" (with a note about adding tags), "Summary" (with a note about reviewing configurations and estimating cost), "Basic configuration" (with fields for Load balancer name, VPC, and Subnet), "Security groups" (with a note about security groups), "Network mapping" (with a note about VPC and Subnet), "Listeners and routing" (with a note about HTTP-80 and target groups), "Add-on services" (with a note about none), "Tags" (with a note about none), and "Attributes" (with a note about certain default attributes). At the bottom right, there are "Cancel" and "Create load balancer" buttons.

Now click on create load balancer.

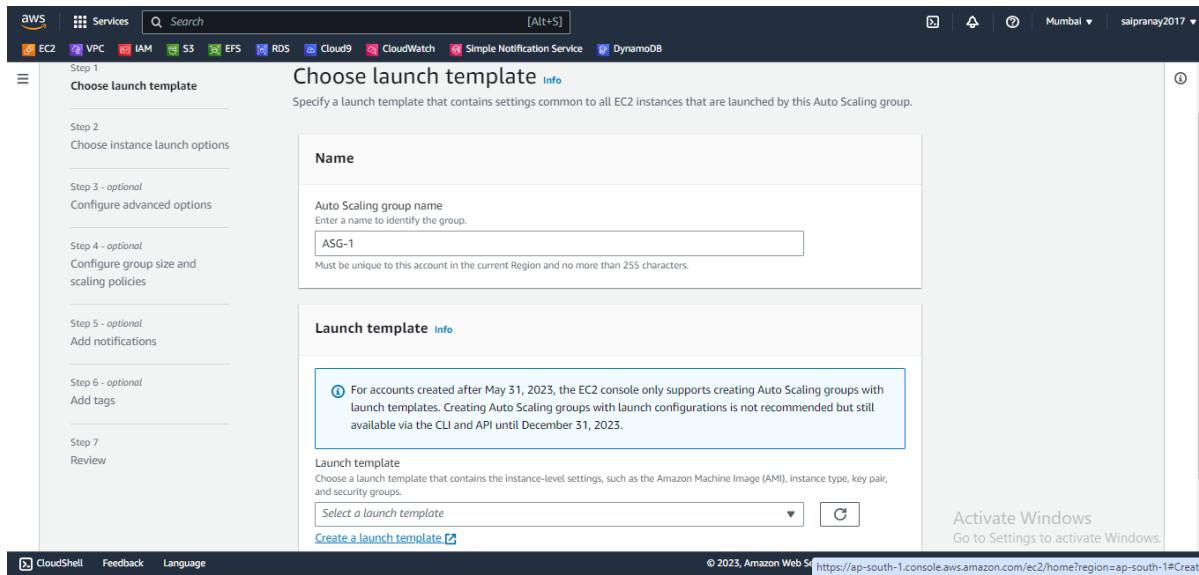
The screenshot shows the AWS EC2 Load Balancers page. The top navigation bar includes services like EC2, VPC, IAM, S3, EFS, RDS, Cloud9, CloudWatch, Simple Notification Service, and DynamoDB. The main content area displays a table titled "Load balancers (1)". The table has columns for Name, DNS name, State, and VPC ID. One row is listed: "ELB-1" with "ELB-1-1909890956.ap-south-1.elb.amazonaws.com" as the DNS name, "Active" state, and "vpc-0518051a4b9b12c3d" VPC ID. Below the table, a message says "0 load balancers selected". A note at the bottom says "Select a load balancer above." The left sidebar contains links for EC2 Dashboard, EC2 Global View, Events, Limits, Instances (with sub-links for Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations), Images (with AMIs), and Elastic Block Store (with CloudShell, Feedback, Language). The bottom right corner shows copyright information: "© 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences".

The application load balancer is created now.

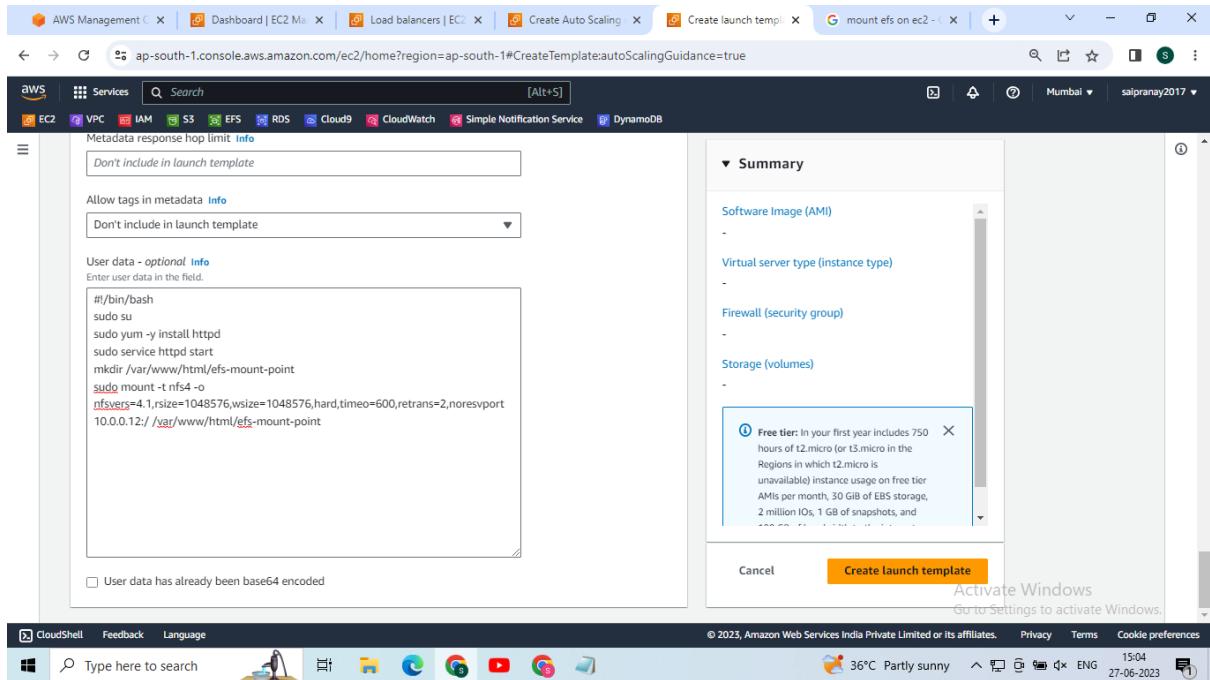
Auto Scaling Group:

The screenshot shows the AWS Auto Scaling Groups page. The top navigation bar includes services like EC2, VPC, IAM, S3, EFS, RDS, Cloud9, CloudWatch, Simple Notification Service, and DynamoDB. The main content area features a large banner with the text "Amazon EC2 Auto Scaling helps maintain the availability of your applications". Below the banner, a section titled "How it works" shows a diagram of an "Auto Scaling group" containing four squares, one of which is dashed. To the right, a "Pricing" section states that there are no additional fees beyond service fees for Amazon EC2, CloudWatch, and other AWS resources. A call-to-action button "Create Auto Scaling group" is prominently displayed. The left sidebar contains links for EC2 Dashboard, EC2 Global View, Events, Limits, Instances, Images (AMIs), and Elastic Block Store (CloudShell, Feedback, Language). The bottom right corner shows copyright information: "© 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences".

Now creating an auto scaling group to launch new instances.



- Give the autoscaling group name
- Now giving a template to launch an instance.



Giving the template name **Presentation**.

While launching the template all of options are set like the instances launched before but :-

In the user data section of launching a template the COMMANDS are given so that when the ASG launches an instance, the instance is directly mounted to the Elastic File System.

The screenshot shows the AWS CloudFormation console interface. The user is creating a launch template named 'Presentation'. The configuration includes:

Description	Launch template	Instance type
-	Presentation lt-09e839404fe6e91b3	t2.micro
AMI ID	ami-049a62eb90480f276	Security groups
Key pair name	pranaykey	Security group IDs sg-047c3890b23e84ccb
Additional details		
Storage (volumes)	-	Date created Tue Jun 27 2023 13:13:18 GMT+0530 (India Standard Time)

At the bottom right, there is an 'Activate Windows' message with a link to Settings.

Choosing the instance template

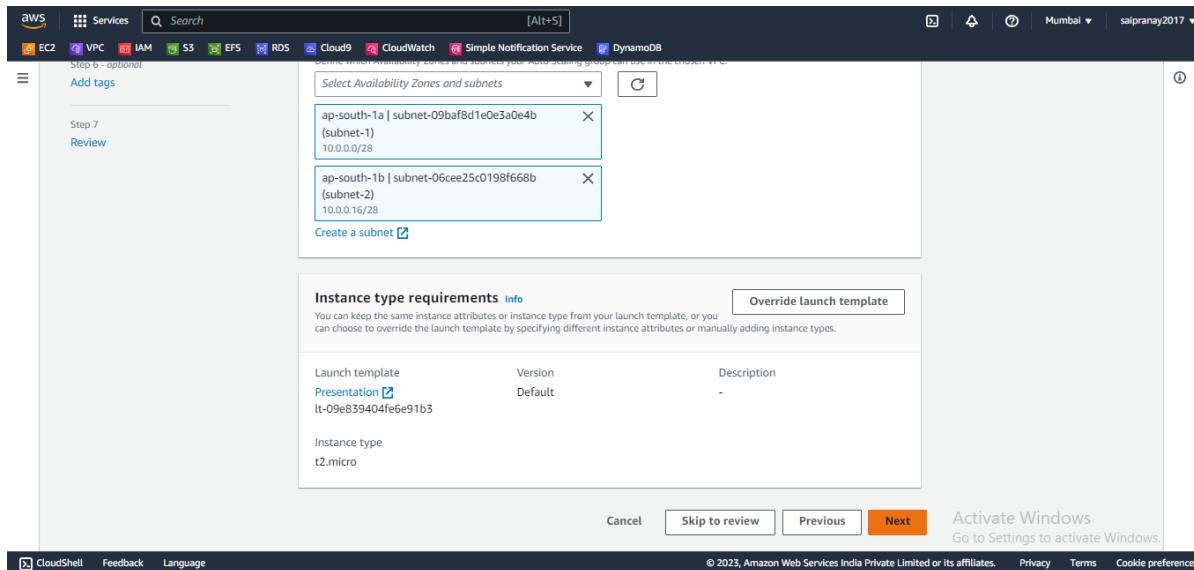
The screenshot shows the second step of the CloudFormation wizard, titled 'Step 2: Choose instance launch options'. The 'Network' tab is selected. The user has chosen a VPC named 'vpc-0518051a4b9b12c3d (vpc-1)'. Under 'Availability Zones and subnets', two subnets are selected:

- ap-south-1a | subnet-09baf8d1e0e3a0e4b (subnet-1)
10.0.0.0/24
- ap-south-1b | subnet-06cee25c0198f668b (subnet-2)
10.0.0.16/28

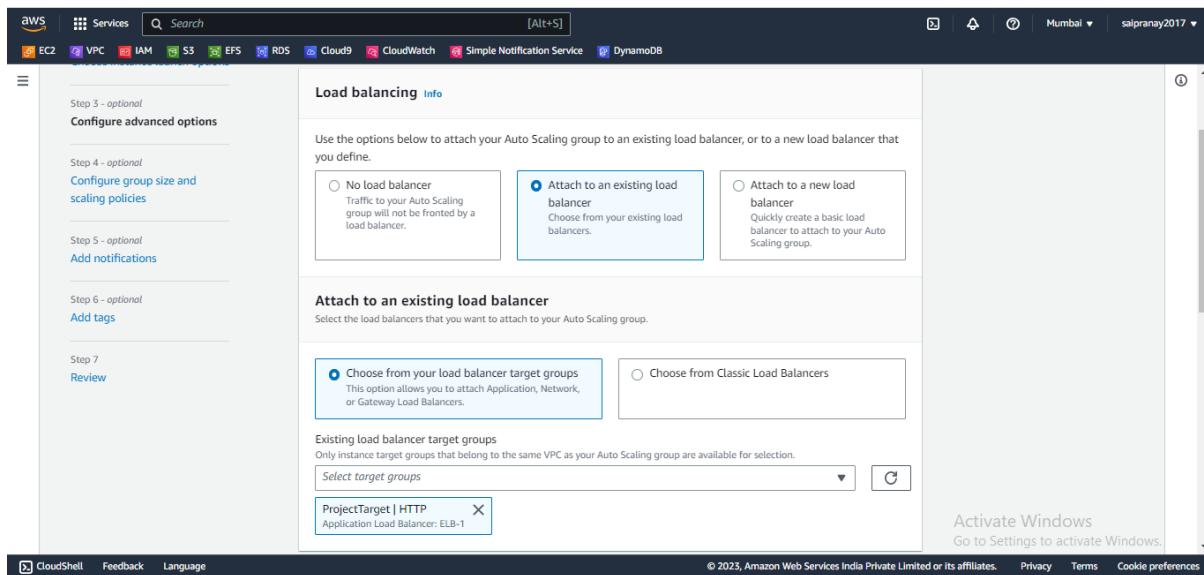
At the bottom right, there is an 'Activate Windows' message with a link to Settings.

Selecting the custom VPC

Selecting the Subnets present in different availability zones



Click on Next



Attaching the auto scaling group to the elastic load balancer.

The screenshot shows the AWS Auto Scaling Groups creation wizard at Step 4: **Configure group size and scaling policies - optional**. The page title is "Configure group size and scaling policies - optional". A sub-instruction says: "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." Below this, there's a section titled "Group size - optional" with fields for Desired capacity (set to 3), Minimum capacity (set to 1), and Maximum capacity (set to 10). At the bottom, there's a "Scaling policies - optional" section with a radio button for "Target tracking scaling policy" (which is selected) and another for "None". Other optional steps like "Choose launch template", "Choose instance launch options", "Configure advanced options", "Add notifications", and "Add tags" are listed on the left.

Giving the scaling policies

-Different capacity of instances requirements

The screenshot shows the AWS Auto Scaling Groups creation wizard at Step 5: **Scaling policies - optional**. The page title is "Scaling policies - optional". A sub-instruction says: "Choose whether to use a scaling policy to dynamically resize your Auto Scaling group to meet changes in demand." Below this, there's a section titled "Scaling policies - optional" with a radio button for "Target tracking scaling policy" (which is selected) and another for "None". Other fields include "Scaling policy name" (set to "Target Tracking Policy"), "Metric type" (set to "Average CPU utilization"), "Target value" (set to 30), and "Instances need" (set to 300 seconds warm up before including in metric). A checkbox for "Disable scale in to create only a scale-out policy" is also present. The "None" option is shown in a separate panel.

Selecting a target Scaling policy

And click on next

Using the SNS Service :

The screenshot shows the 'Add notifications - optional' step of the EC2 Auto Scaling group creation wizard. On the left, a sidebar lists steps from 1 to 7. Step 6 is 'Add notifications'. The main area shows a 'Notification 1' configuration. It includes fields for 'Send a notification to' (set to 'Project-Topic'), 'With these recipients' (set to 'saipranayganta2003@gmail.com'), and a 'Use existing topic' button. Under 'Event types', four checkboxes are checked: 'Launch', 'Terminate', 'Fail to launch', and 'Fail to terminate'. A large 'Add notification' button is at the bottom. Navigation buttons at the bottom right include 'Cancel', 'Skip to review', 'Previous', and a highlighted 'Next' button.

Creating a topic

- Giving the topic name
- Giving an email address to whom the email should be sent when an instance is created or terminated.

The screenshot shows the 'Notifications' section of the EC2 Auto Scaling group creation wizard. It displays 'Notification 1' with the SNS Topic 'Project-Topic' and the recipient 'saipranayganta2003@gmail.com'. The 'Event types' for this notification are checked: Launch, Terminate, Fail to launch, and Fail to terminate. Below this, the 'Step 6: Add tags' section is shown, which is currently empty. Navigation buttons at the bottom right include 'Cancel', 'Previous', a highlighted 'Create Auto Scaling group' button, and 'Windows'.

Reviewing the selected conditions and click on create

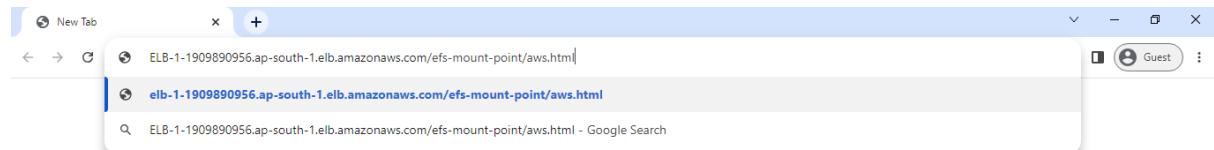
The screenshot shows the AWS Auto Scaling Groups page. At the top, there is a navigation bar with the AWS logo, a search bar, and links for various services like EC2, VPC, IAM, S3, EFS, RDS, Cloud9, CloudWatch, Simple Notification Service, and DynamoDB. The location is set to Mumbai, and the user is saipranay2017. A green banner at the top indicates "ASG-1, 1 Scaling policy, 1 Notification created successfully". Below the banner, the page title is "EC2 > Auto Scaling groups". The main content area has a header "Auto Scaling groups (1) Info" with tabs for "Launch configurations", "Launch templates", "Actions", and "Create Auto Scaling group". There is a search bar and a table with one row for "ASG-1". The table columns include Name, Launch template/configuration, Instances, Status, Desired capacity, Min, and Max. The "ASG-1" row shows "Presentation | Version Default", 3 instances, and values for Desired capacity, Min, and Max. At the bottom, it says "0 Auto Scaling groups selected".

Now the auto-scaling group is created.

The screenshot shows the AWS Load Balancers page. On the left, there is a sidebar with navigation links for Images, AMIs, AMI Catalog, Elastic Block Store (Volumes, Snapshots, Lifecycle Manager), Network & Security (Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces), Load Balancing (Load Balancers, Target Groups), and Auto Scaling. The "Load Balancers" link is highlighted in red. The main content area has a header "Load balancers (1)". It says "Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic." There is a "Create load balancer" button and a search bar. A table lists one load balancer named "ELB-1". The table columns include Name, State, VPC ID, and Actions. The "ELB-1" row shows "ELB-1-1909890956.ap-south-1.elb.amazonaws.com", "Active", and "vpc-0518051a4b9b12c3d". At the bottom, it says "0 load balancers selected" and "Select a load balancer above." A green message "DNS name copied" is displayed above the table.

Now copy the DNS name of the Elastic load balancer.

Web Page Hosting :



You're browsing as a Guest

Pages you view in this window won't appear in the browser history
and they won't leave other traces, like cookies, on the computer
after you close all open Guest windows. Any files you download will
be preserved, however.

[Learn more](#)

Activate Windows
Go to Settings to activate Windows.

- Copy the link and add /efs-mount-point/aws.html
- As the file is present in the EFS

AWS Architect Webpage Not secure elb-1-1909890956.ap-south-1.elb.amazonaws.com/efs-mount-point/aws.html

AWS Architect Webpage Sai Pranay Sai Mahesh Manikanta

Welcome to My AWS Architect

This is a simple webpage built to explain my architect.

The architecture described above focuses on distributing user requests in a proportional manner using an Elastic Load Balancer (ELB) across multiple instances created by an Auto Scaling group. Additionally, an Elastic File System (EFS) is set up to store web pages. The instances created by the Auto Scaling group are directly connected to the EFS, allowing them to access and serve the web pages. The ELB manages the incoming requests and directs them to the appropriate instances hosting the web pages stored in the EFS. Moreover, the architecture utilizes the SNS service to send email notifications to the administrator whenever an instance is created, terminated, or encounters failures during creation or termination within the Auto Scaling group. This comprehensive solution ensures efficient distribution of user requests, seamless web page hosting, and enables timely notifications to the administrator regarding instance management events.

Thank You Aashudev Sir

Activate Windows
Go to Settings to activate Windows.

Finally the web page is hosted :)