

# AWS PROJECTS AND TASKS

1. Load balancing using spot instances in online shopping

Agenda:

- Spot instance
- Launch Template
- Auto Scaling Group
- Load Balancer
- Target Groups

Real life examples of Online shopping.:



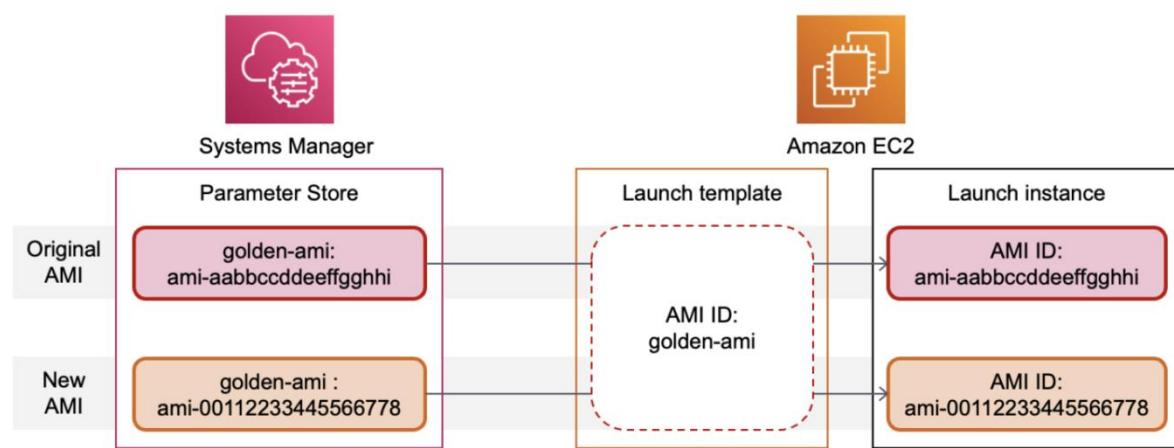
- **Spot instances:**

These instances are the cheapest one and good for small duration like few hours to few days only which works on bid pricing. If we bid for the instance then the instance should be available within the bid price or if bid pricing goes high then instance would be automatically terminated by aws. These instances are good for users who wants high processing power at low price and for minimum duration.



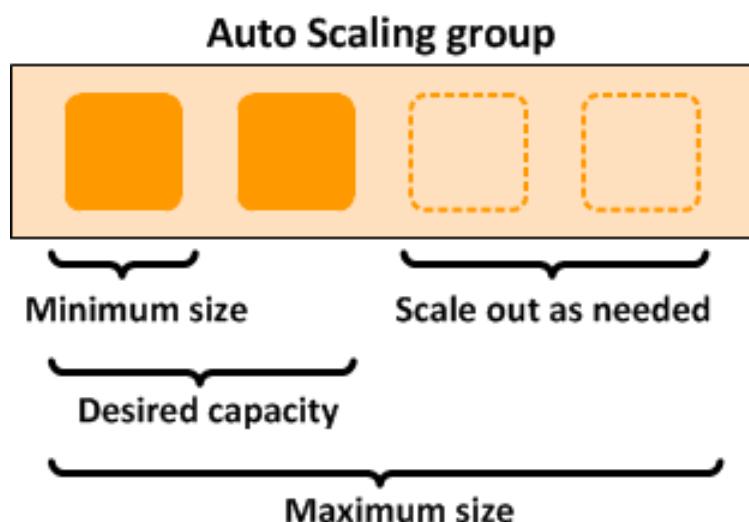
- Launch Templates:

Simplify and streamline the process of launching EC2 instances in AWS



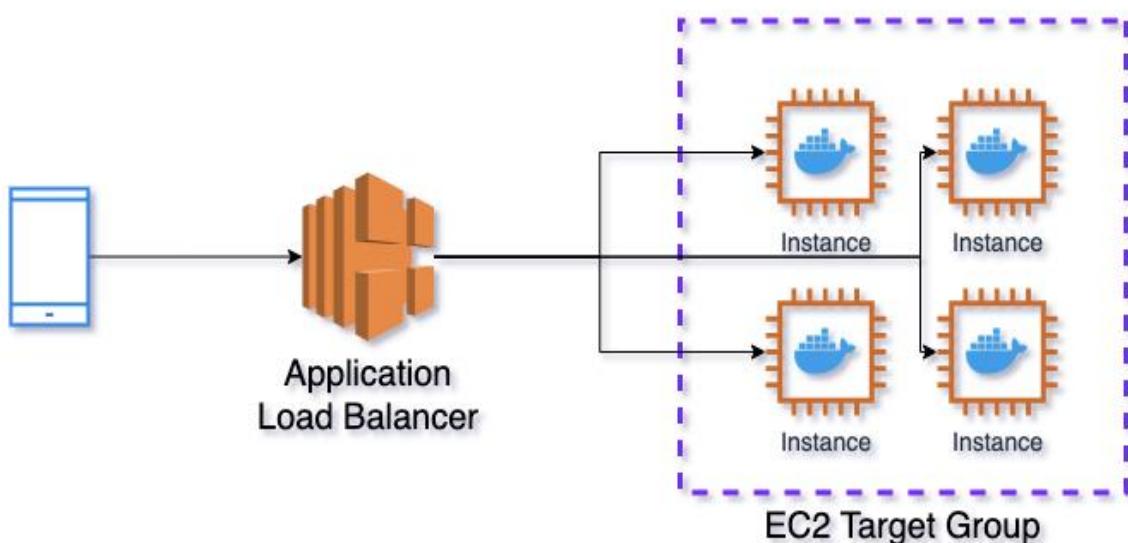
- Auto scaling group:

Auto scaling group can automatically add or remove instances based on traffic load and requirements.



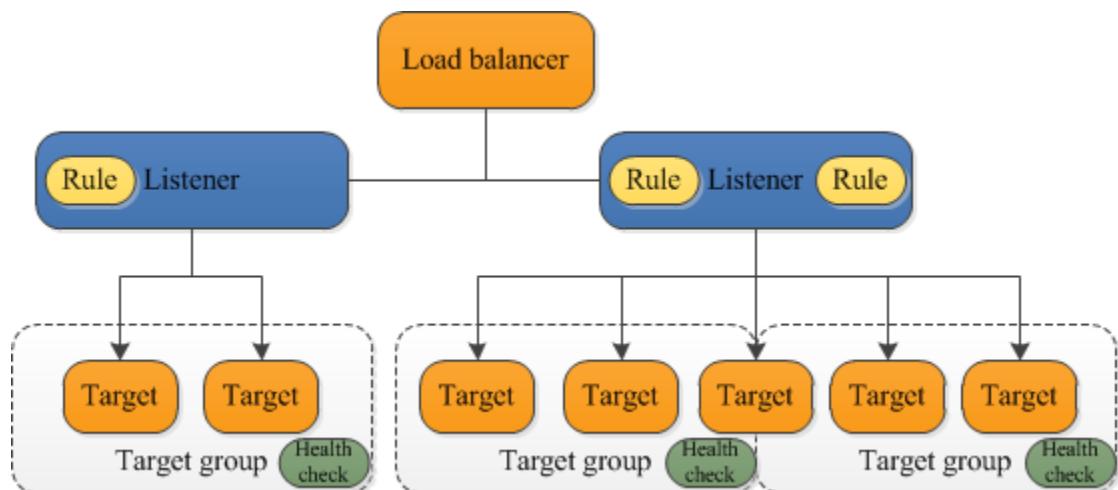
- Load Balancer:

Network device or software application that distributes incoming network traffic across multiple servers or resources to ensure that no single server becomes overloaded.



- Target groups:

are the instances which are part of the load balancing pool.  
It routes traffic from the load balancer to set of registered targets



\*Create the spot instances for load balancing in an online store can help reduce costs while ensuring high availability

**Instances (1/2) Info**

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
OnlineShopping	i-0031d6d3cc2f687cc	Running	t2.micro	2/2 checks passed	No alarms	us-east-1d
	i-06bdf90c5825c02dd	Terminated	t2.micro	-	No alarms	us-east-1a

**Instance: i-0031d6d3cc2f687cc (OnlineShopping)**

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

**Instance summary**

Instance ID	Public IPv4 address	Private IPv4 addresses
i-0031d6d3cc2f687cc (OnlineShopping)	3.85.14.55 [open address]	172.31.28.24
IPv6 address	Instance state	Public IPv4 DNS
-	Running	ec2-3-85-14-55.compute-1.amazonaws.com [open address]
Hostname type	Private IP DNS name (IPv4 only)	

\*Spot Instance VPC and Subnet configurations

#### Instance: i-0031d6d3cc2f687cc (OnlineShopping)

Answer private resource DNS name IPv4 (A)	Instance type t2.micro	Elastic IP addresses -
Auto-assigned IP address 3.85.14.55 [Public IP]	VPC ID vpc-04615e6acb7d30ef3	AWS Compute Optimizer finding <a href="#">Opt-in to AWS Compute Optimizer for Learn more</a>
IAM Role -	Subnet ID subnet-034f718e27409c4cc	Auto Scaling Group name -
IMDSv2 Required		

\*Spot instance created and the following key pair attached with it

#### Instance: i-0031d6d3cc2f687cc (OnlineShopping)

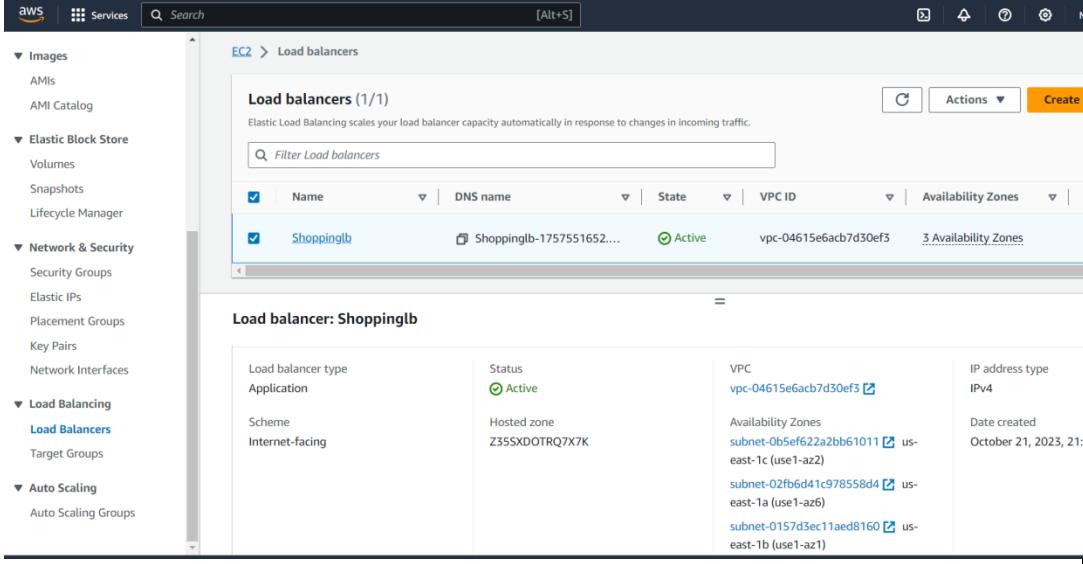
Platform details Linux/UNIX	AMI name al2023-ami-2023.2.20231016.0-kernel-6.1-x86_64	Termination protection Disabled
Stop protection Disabled	Launch time Sat Oct 21 2023 20:56:09 GMT+0530 (India Standard Time) (1 day)	AMI location amazon/al2023-ami-2023.2.202310x86_64
Instance auto-recovery Default	Lifecycle spot	Stop-hibernate behavior Disabled
AMI Launch index 0	Key pair assigned at launch shoppingkey	State transition reason -

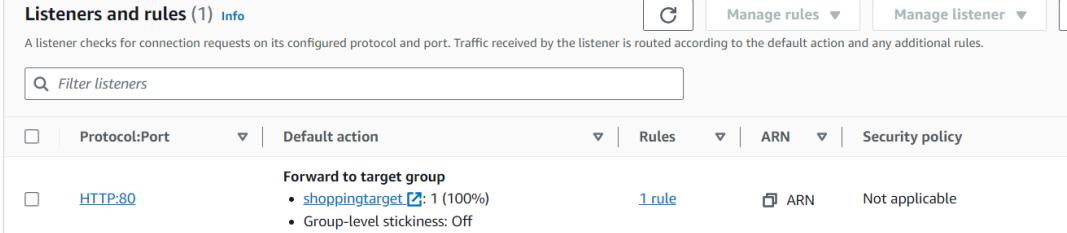
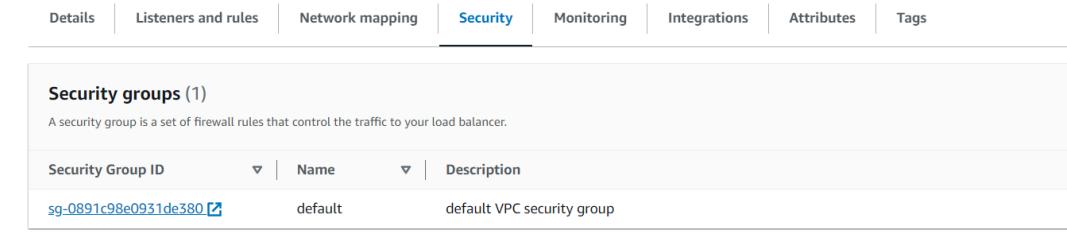
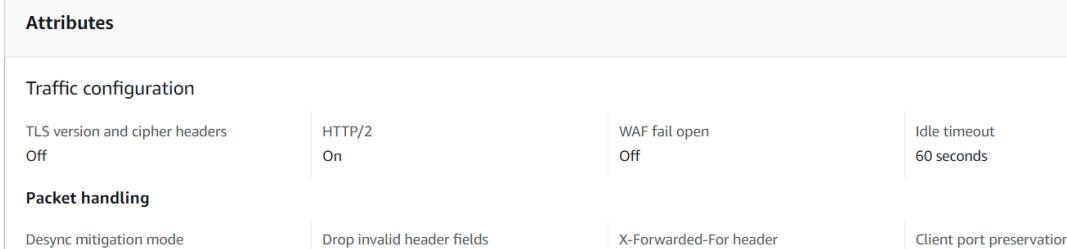
\*Inbound and outbound rules are typically associated with security groups and network access control lists (ACLs) in

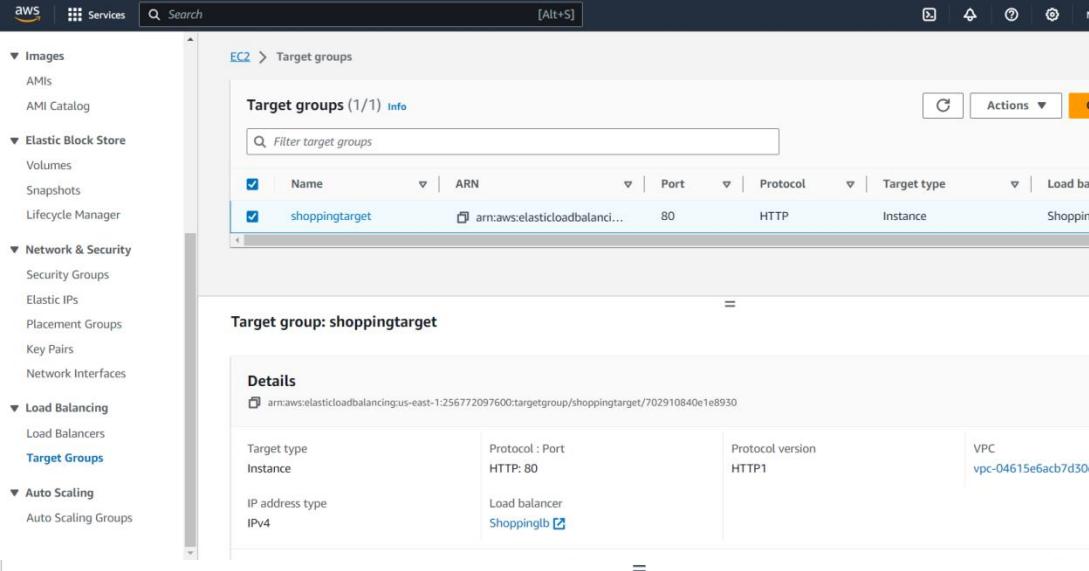
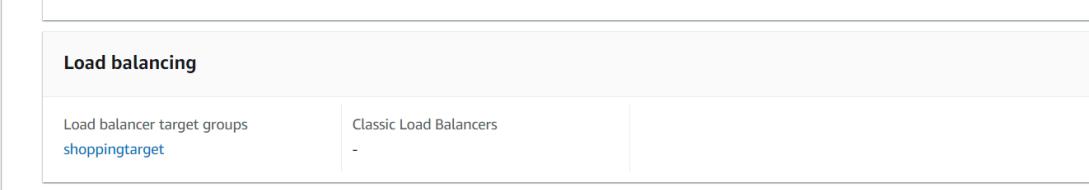
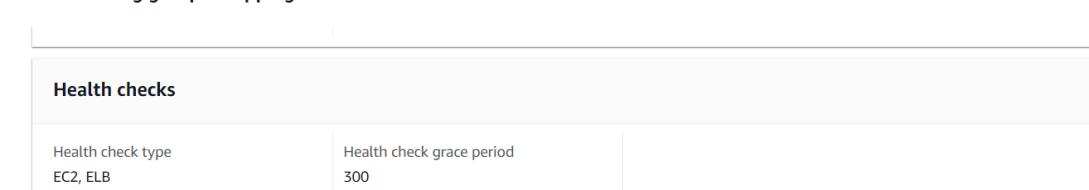
#### Instance: i-0031d6d3cc2f687cc (OnlineShopping)

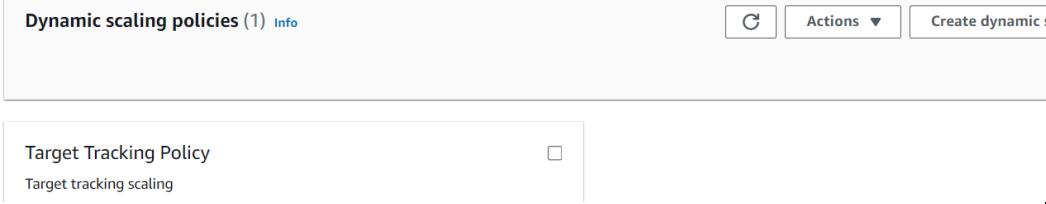
Security groups sg-0891c98e0931de380 (default)	<b>Inbound rules</b>										
<i>Filter rules</i>											
<table border="1"> <thead> <tr> <th>Name</th> <th>Security group rule ID</th> <th>Port range</th> <th>Protocol</th> <th>Source</th> </tr> </thead> <tbody> <tr> <td>-</td> <td>sgr-0f41dca8e534b4f42</td> <td>All</td> <td>All</td> <td>sg-0891c98e0931de380</td> </tr> </tbody> </table>		Name	Security group rule ID	Port range	Protocol	Source	-	sgr-0f41dca8e534b4f42	All	All	sg-0891c98e0931de380
Name	Security group rule ID	Port range	Protocol	Source							
-	sgr-0f41dca8e534b4f42	All	All	sg-0891c98e0931de380							

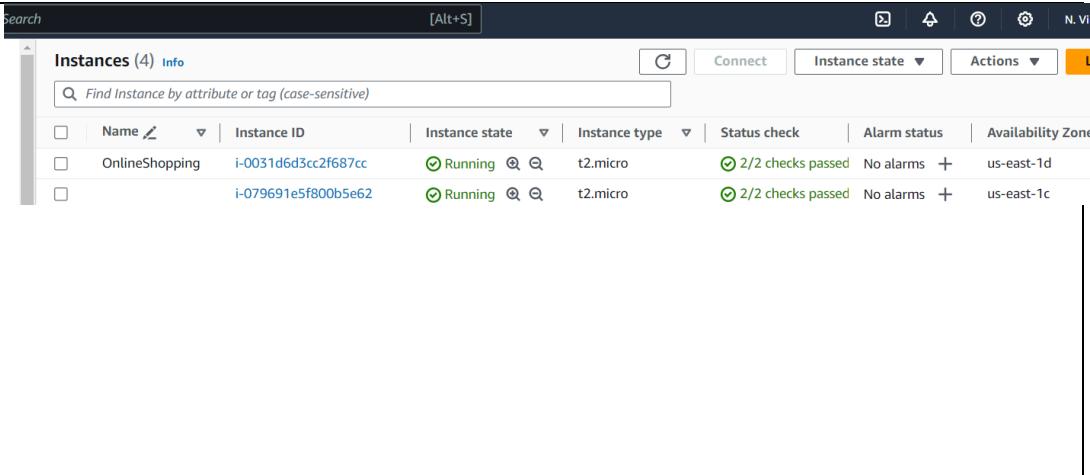
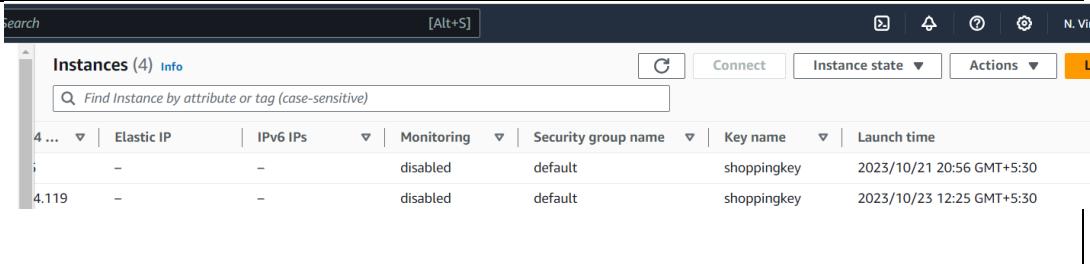
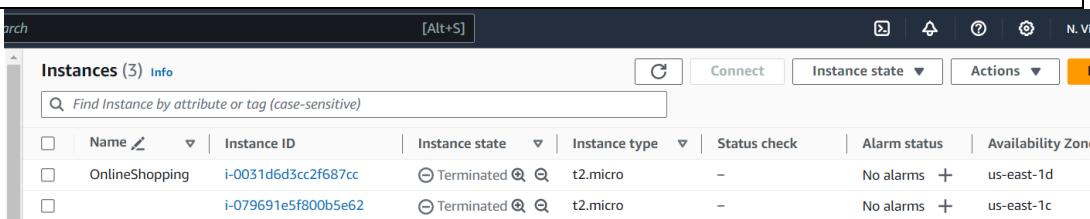
cloud computing environment																															
*Spot instance inbound rules and outbound rules	<p><b>Instance: i-0031d6d3cc2f687cc (OnlineShopping)</b></p> <p>The screenshot shows the AWS Security Groups interface. It displays two tables of security group rules.</p> <p><b>Inbound rules:</b></p> <table border="1"> <thead> <tr> <th>Name</th> <th>Security group rule ID</th> <th>Port range</th> <th>Protocol</th> <th>Source</th> </tr> </thead> <tbody> <tr> <td>-</td> <td>sgr-0f41dca8e534b4f42</td> <td>All</td> <td>All</td> <td>sg-0891c98e0931de380</td> </tr> </tbody> </table> <p><b>Outbound rules:</b></p> <table border="1"> <thead> <tr> <th>Name</th> <th>Security group rule ID</th> <th>Port range</th> <th>Protocol</th> <th>Destination</th> </tr> </thead> <tbody> <tr> <td>-</td> <td>sgr-03daad6c8f048733e</td> <td>All</td> <td>All</td> <td>0.0.0.0/0</td> </tr> </tbody> </table>	Name	Security group rule ID	Port range	Protocol	Source	-	sgr-0f41dca8e534b4f42	All	All	sg-0891c98e0931de380	Name	Security group rule ID	Port range	Protocol	Destination	-	sgr-03daad6c8f048733e	All	All	0.0.0.0/0										
Name	Security group rule ID	Port range	Protocol	Source																											
-	sgr-0f41dca8e534b4f42	All	All	sg-0891c98e0931de380																											
Name	Security group rule ID	Port range	Protocol	Destination																											
-	sgr-03daad6c8f048733e	All	All	0.0.0.0/0																											
*Create an AWS Launch Template that specifies the configuration for your spot instances. *You can define the instance type, security groups, key pair, and other settings.	<p>The screenshot shows the AWS Launch Templates interface. It displays a list of launch templates and the details for the Shoppinglt template.</p> <p><b>Launch Templates (1/1) Info</b></p> <table border="1"> <thead> <tr> <th>Launch Template ID</th> <th>Launch Template Name</th> <th>Default Version</th> <th>Latest Version</th> <th>Create Time</th> </tr> </thead> <tbody> <tr> <td>lt-0a0a131681e54fab5</td> <td>Shoppinglt</td> <td>1</td> <td>1</td> <td>2023-10-21T15:48:25.000Z</td> </tr> </tbody> </table> <p><b>Shoppinglt (lt-0a0a131681e54fab5)</b></p> <table border="1"> <thead> <tr> <th colspan="4">Instance details</th> </tr> <tr> <th>AMI ID</th> <th>Instance type</th> <th>Availability Zone</th> <th>Key pair name</th> </tr> </thead> <tbody> <tr> <td>ami-0df435f331839b2d6</td> <td>t2.micro</td> <td>-</td> <td>shoppingkey</td> </tr> <tr> <td>Security groups</td> <td>Security group IDs</td> <td colspan="2"></td> </tr> <tr> <td>-</td> <td>sg-0891c98e0931de380</td> <td colspan="2"></td> </tr> </tbody> </table>	Launch Template ID	Launch Template Name	Default Version	Latest Version	Create Time	lt-0a0a131681e54fab5	Shoppinglt	1	1	2023-10-21T15:48:25.000Z	Instance details				AMI ID	Instance type	Availability Zone	Key pair name	ami-0df435f331839b2d6	t2.micro	-	shoppingkey	Security groups	Security group IDs			-	sg-0891c98e0931de380		
Launch Template ID	Launch Template Name	Default Version	Latest Version	Create Time																											
lt-0a0a131681e54fab5	Shoppinglt	1	1	2023-10-21T15:48:25.000Z																											
Instance details																															
AMI ID	Instance type	Availability Zone	Key pair name																												
ami-0df435f331839b2d6	t2.micro	-	shoppingkey																												
Security groups	Security group IDs																														
-	sg-0891c98e0931de380																														
*Create an Auto Scaling group on your cloud provider (e.g., AWS, Google Cloud) that utilizes spot instances. *This group will automatically adjust the number of	<p>The screenshot shows the AWS Auto Scaling Groups interface. It displays a list of auto scaling groups and the details for the ShoppingASG group.</p> <p><b>Auto Scaling groups (1/1) Info</b></p> <table border="1"> <thead> <tr> <th>Name</th> <th>Launch template/configuration</th> <th>Instances</th> <th>Status</th> <th>Desired capacity</th> </tr> </thead> <tbody> <tr> <td>ShoppingASG</td> <td>Shoppinglt   Version Default</td> <td>2</td> <td>-</td> <td>1</td> </tr> </tbody> </table> <p><b>Auto Scaling group: ShoppingASG</b></p> <p><b>Launch template</b></p> <table border="1"> <thead> <tr> <th>Launch template</th> <th>AMI ID</th> <th>Instance type</th> <th>Owner</th> </tr> </thead> <tbody> <tr> <td>lt-0a0a131681e54fab5 Shoppinglt</td> <td>ami-0df435f331839b2d6</td> <td>t2.micro</td> <td>arn:aws:iam::25677209760</td> </tr> <tr> <td>Version</td> <td>Security groups</td> <td>Security group IDs</td> <td>Create time</td> </tr> </tbody> </table>	Name	Launch template/configuration	Instances	Status	Desired capacity	ShoppingASG	Shoppinglt   Version Default	2	-	1	Launch template	AMI ID	Instance type	Owner	lt-0a0a131681e54fab5 Shoppinglt	ami-0df435f331839b2d6	t2.micro	arn:aws:iam::25677209760	Version	Security groups	Security group IDs	Create time								
Name	Launch template/configuration	Instances	Status	Desired capacity																											
ShoppingASG	Shoppinglt   Version Default	2	-	1																											
Launch template	AMI ID	Instance type	Owner																												
lt-0a0a131681e54fab5 Shoppinglt	ami-0df435f331839b2d6	t2.micro	arn:aws:iam::25677209760																												
Version	Security groups	Security group IDs	Create time																												

instances based on traffic.	
<p>*The practice of distributing subnets across Availability Zones is a best practice for achieving high availability, fault tolerance, and efficient load balancing in cloud-based applications for online stores using spot instances in AWS.</p>	<p><b>Auto Scaling group: ShoppingASG</b></p>  <p>The screenshot shows the Network section of the Auto Scaling group configuration. It lists the Availability Zones (us-east-1a, us-east-1b, us-east-1c) and the Subnet ID (subnet-0b5ef622a2bb61011, subnet-02fb6d41c978558d4, subnet-0157d3ec11aed8160).</p>
<p>*Deploy a load balancer, such as an Application Load Balancer (ALB) on AWS to distribute incoming traffic efficiently across your spot instances. *This ensures even load distribution and fault</p>	 <p>The screenshot shows the Load balancers section of the EC2 service. It displays a single Application Load Balancer named Shoppinglb, which is active and associated with the VPC vpc-04615e6acb7d30ef3 and three availability zones (us-east-1a, us-east-1b, us-east-1c). The load balancer is configured with an Internet-facing scheme and is connected to the Z35SXDOTRQ7X7K hosted zone.</p>

tolerance.	
<ul style="list-style-type: none"> <li>*Load balancers perform health checks on the spot instances in the associated target group.</li> <li>* If a spot instance becomes unhealthy or is terminated, the load balancer will stop routing traffic to it.</li> <li>*This ensures that only healthy instances serve customer requests</li> </ul>	<p><b>Load balancer: Shoppinglb</b></p>  <p>The screenshot shows the 'Listeners and rules' section for the Shoppinglb load balancer. It displays a single listener for 'HTTP:80' with a default action of 'Forward to target group' pointing to a target group named 'shoppingtarget'. There is one rule defined and no ARN or security policy assigned.</p>
<ul style="list-style-type: none"> <li>*Load balancer security groups</li> </ul>	<p><b>Load balancer: Shoppinglb</b></p>  <p>The screenshot shows the 'Security' tab for the Shoppinglb load balancer. It lists a single security group named 'sg-0891c98e0931de380' which is the default VPC security group.</p>
<ul style="list-style-type: none"> <li>*The HTTP port is essential for efficiently routing web traffic and ensuring web</li> </ul>	<p><b>Load balancer: Shoppinglb</b></p>  <p>The screenshot shows the 'Attributes' tab for the Shoppinglb load balancer. It includes sections for 'Traffic configuration' (TLS version and cipher headers, WAF fail open, Idle timeout), 'Packet handling' (Desync mitigation mode, Drop invalid header fields, X-Forwarded-For header, Client port preservation), and 'Monitoring' (CloudWatch Metrics and CloudWatch Logs).</p>

application support	
*When traffic arrives at the load balancer, it distributes incoming requests across the spot instances in the target group	 <p>The screenshot shows the AWS EC2 Target Groups console. On the left, there's a navigation sidebar with options like Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, Network &amp; Security, Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces, Load Balancing, Load Balancers, Target Groups (which is selected), and Auto Scaling. The main area displays a table for 'Target groups (1/1)'. The table has columns for Name, ARN, Port, Protocol, Target type, and Load balancer. One row is selected, showing 'Name: shoppingtarget', 'ARN: arn:aws:elasticloadbalancing:us-east-1:256772097600:targetgroup/shoppingtarget/702910840e1e8930', 'Port: 80', 'Protocol: HTTP', 'Target type: Instance', and 'Load balancer: Shoppinglb'. Below the table, a section titled 'Target group: shoppingtarget' shows 'Details' including the ARN, Target type (Instance), Protocol (HTTP: 80), IP address type (IPv4), and Protocol version (HTTP1).</p>
*Configure your load balancer to work in conjunction with the Auto Scaling group. * This enables automatic addition or removal of spot instances as per the traffic demand	 <p>The screenshot shows the AWS Auto Scaling Groups console. It displays the configuration for the 'Auto Scaling group: ShoppingASG'. Under the 'Load balancing' section, it shows 'Load balancer target groups: shoppingtarget' and 'Classic Load Balancers: -'. There are also sections for 'Health checks' and 'Scaling policies'.</p>
*Set up health checks to monitor the status of your spot instances. *The load balancer can route traffic only	 <p>The screenshot shows the AWS Auto Scaling Groups console. It displays the configuration for the 'Auto Scaling group: ShoppingASG'. Under the 'Health checks' section, it shows 'Health check type: EC2, ELB' and 'Health check grace period: 300'. There are also sections for 'Scaling policies' and 'Metrics'.</p>

to healthy instances.	
<ul style="list-style-type: none"> <li>*Create auto scaling policies that define when and how instances should be added or removed based on metrics like CPU utilization or request count.</li> <li>*Dynamic scaling policies are crucial for managing the workload and costs in an online store that uses spot instances</li> </ul>	<p>Auto Scaling group: ShoppingASG</p>  <p>Dynamic scaling policies (1) <a href="#">Info</a> <a href="#">Edit</a> Actions ▾ <a href="#">Create dynamic scaling policy</a></p> <p>Target Tracking Policy <input type="checkbox"/> Target tracking scaling</p>
<ul style="list-style-type: none"> <li>*Target tracking scaling policy adjust resource capacity based on actual demand and handle fluctuations in traffic.</li> </ul>	<p><b>Auto Scaling group: ShoppingASG</b></p> <p><b>Target Tracking Policy</b> </p> <p>Target tracking scaling</p> <p>Enabled</p> <p>As required to maintain Application Load Balancer request count per target at 50</p> <p>Add or remove capacity units as required</p> <p>300 seconds to warm up before including in metric</p>

*The spot instance below online shopping launched based on auto scaling policies given to auto scaling group	
*The spot instance have been created with same configuration	
*The spot instances has been terminated	

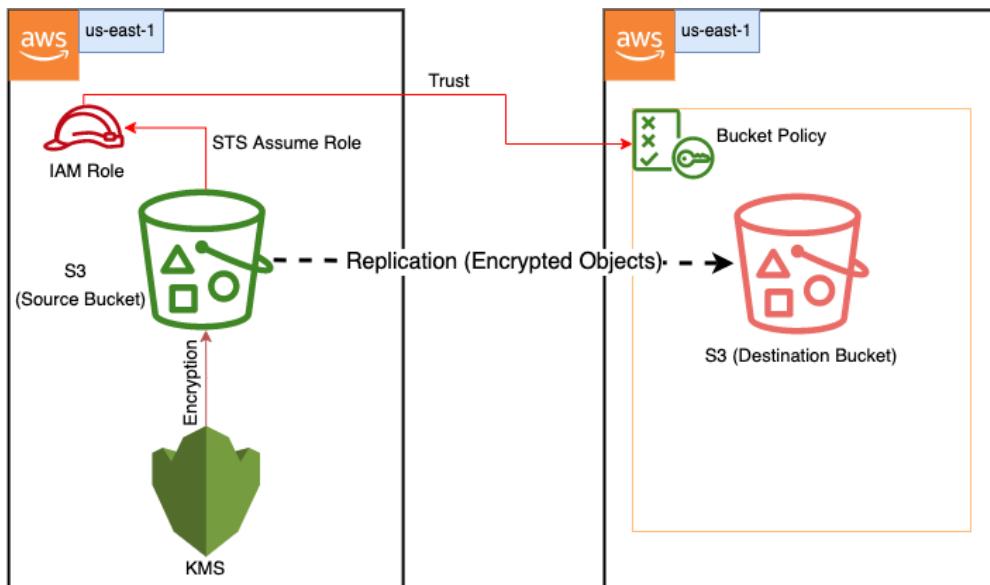
- Regularly monitor your system's performance and traffic patterns. Adjust your auto scaling and load balancing configurations as needed to optimize resource utilization and customer experience.
- By following these steps, you can ensure that your online store is highly available and responsive while benefiting from the cost advantages of spot instances.

## 2. Static website hosting in S3 bucket

Agenda:

- S3
- Objects
- Static Website Hosting
- ACL(Access Control Lists)

- S3: Amazon S3 (Simple Storage Service) is a scalable object storage service provided by Amazon Web Services (AWS). An S3 bucket is a container for storing objects which can consist of data files, images, videos, and any other type of binary or text data. S3 buckets are fundamental building blocks of Amazon S3, and they play a crucial role in cloud storage and web hosting on AWS. The data could be sent securely from one bucket in a region to another bucket in another region using users authentication and encryption.



- Objects: are the fundamental entity that you store in a bucket. An object consists of data, a key (unique within a bucket), and metadata. When you upload a file to an S3 bucket, you are creating an object. Objects can be anything from a simple text file to complex data, images, videos, or other binary data.

Amazon S3 > Buckets > demo-origin-bucket-1

## demo-origin-bucket-1 Info

**Objects** Properties Permissions Metrics Management Access Points

**Objects (4)**

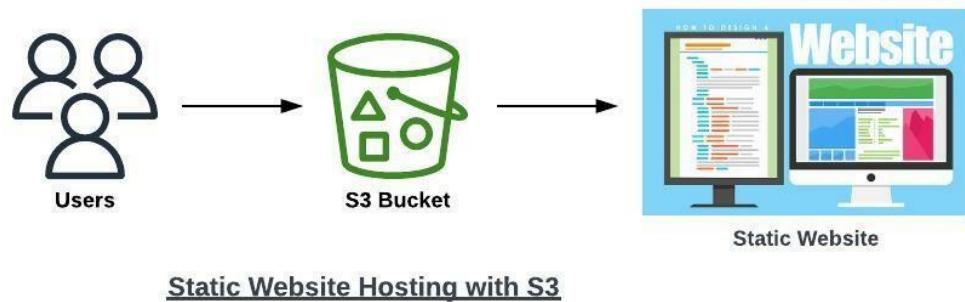
Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

[Actions ▾](#) [Create folder](#)

Find objects by prefix

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	app.py	py	December 10, 2022, 16:29:09 (UTC+05:30)	820.0 B	Standard
<input type="checkbox"/>	cdk.json	json	December 10, 2022, 16:29:08 (UTC+05:30)	194.0 B	Standard
<input type="checkbox"/>	GravitonDev-ec2.yml	yml	December 10, 2022, 16:29:06 (UTC+05:30)	2.8 KB	Standard
<input type="checkbox"/>	pythonLinting.yml	yml	December 10, 2022, 16:29:10 (UTC+05:30)	844.0 B	Standard

- Static Website Hosting: refers to the ability to host and serve static web content directly from Amazon S3, a scalable and durable object storage service. A static website consists of HTML, CSS, JavaScript, images, and other static assets that do not require server-side processing or a database. Hosting a static website in S3 is a cost-effective and straightforward way to make web content available to users.



- ACL: ACL stands for Access Control List. An S3 ACL is a set of permissions that define who can access objects and buckets in S3 and what actions they can perform. ACLs are used to manage access to S3 resources at a finer granularity than what is provided by bucket policies alone.

The screenshot shows the 'Edit Object Ownership' page in the Amazon S3 console. The navigation path is: Amazon S3 > Buckets > sample-bucket > Edit Object Ownership. The main title is 'Edit Object Ownership' with an 'Info' link. Below it is a section titled 'Object Ownership' with a description: 'Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.' There are two radio button options: 'ACLs disabled (recommended)' (selected) and 'ACLs enabled'. The 'ACLs disabled' option states: 'All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.' The 'ACLs enabled' option states: 'Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.' At the bottom, there are 'Cancel' and 'Save changes' buttons.

Amazon S3 > Buckets > sample-bucket > Edit Object Ownership

## Edit Object Ownership Info

**Object Ownership**

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

**ACLs disabled (recommended)**  
All objects in this bucket are owned by this account.  
Access to this bucket and its objects is specified using only policies.

**ACLs enabled**  
Objects in this bucket can be owned by other AWS accounts.  
Access to this bucket and its objects can be specified using ACLs.

Object Ownership  
Bucket owner enforced

Cancel **Save changes**

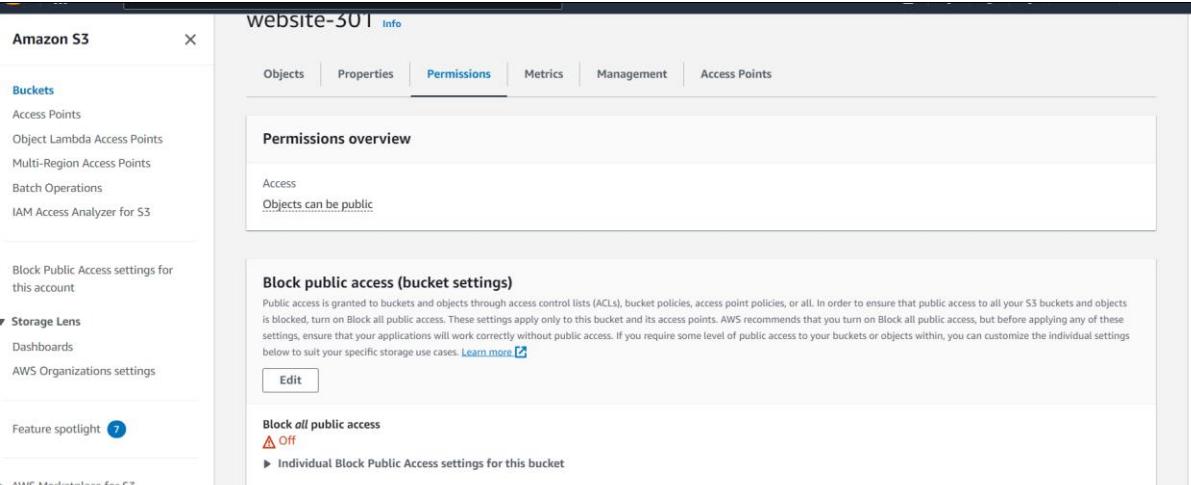
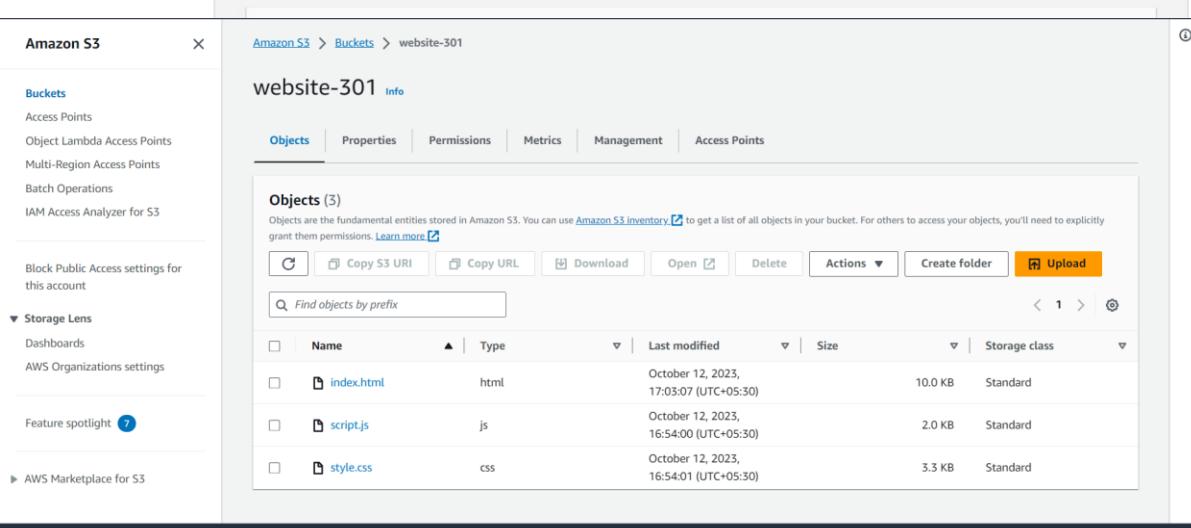
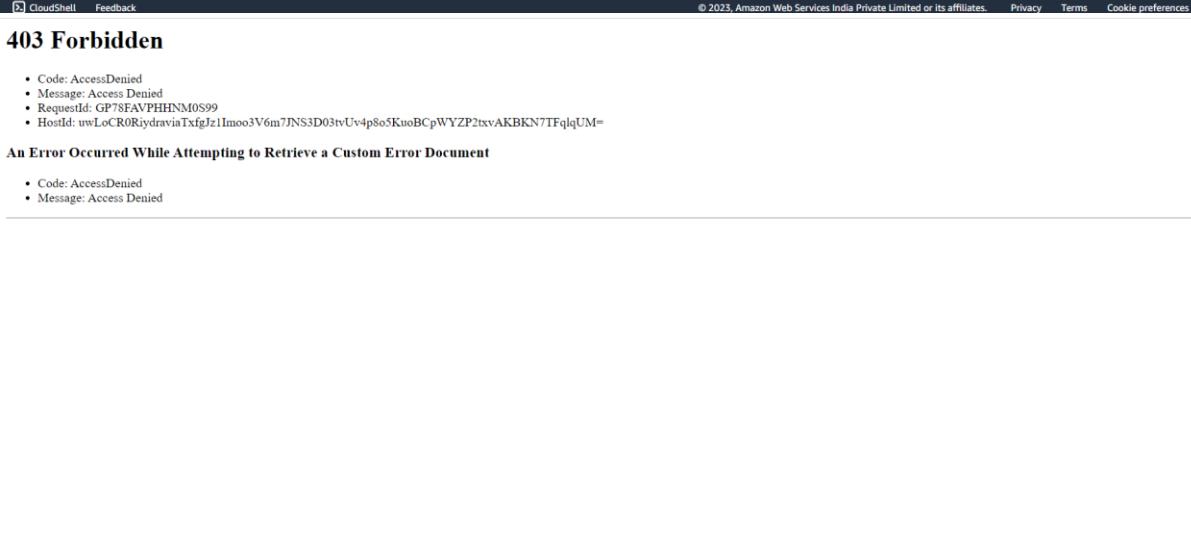
\*Manually Create a S3 bucket in aws cloud

The screenshot shows the AWS S3 Buckets page. On the left, there's a sidebar with navigation links like 'Buckets', 'Access Points', 'Object Lambda Access Points', etc. The main area displays an 'Account snapshot' with a link to 'View Storage Lens dashboard'. Below it, a table lists the bucket 'website-301' with details: Name (website-301), AWS Region (US East (N. Virginia) us-east-1), Access (Objects can be public), and Creation date (October 12, 2023, 16:20:54 (UTC+05:30)). A prominent orange 'Create bucket' button is at the top right of the table.

Name	AWS Region	Access	Creation date
website-301	US East (N. Virginia) us-east-1	Objects can be public	October 12, 2023, 16:20:54 (UTC+05:30)

\*Enable static website hosting in S3 bucket which would assign url for website

This screenshot shows the 'Static website hosting' configuration for the 'website-301' bucket. It includes a note that 'Amazon S3 currently does not support enabling Object Lock after a bucket has been created. To enable Object Lock for this bucket, contact Customer Support'. The 'Requester pays' section is set to 'Disabled'. Under 'Static website hosting', 'Enabled' is selected, and the 'Bucket website endpoint' is listed as <http://website-301.s3-website-us-east-1.amazonaws.com>.

*Block public access should be disabled so that all users can access the website publicly	
*The html,javascript and css files are uploaded to S3 bucket as objects for hosting a website in public	
*The website shows an error because I didn't give the objects in public using ACL.	

\*Through actions I have to give all objects in public using ACL

Amazon S3 > Buckets > website-301

**website-301** [Info](#)

**Objects (3)**

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

<input checked="" type="checkbox"/>	Name	Type	Last modified
<input checked="" type="checkbox"/>	index.html	html	October 12, 2023, 17:03:07 (UTC+05:30)
<input checked="" type="checkbox"/>	script.js	js	October 12, 2023, 16:54:00 (UTC+05:30)
<input checked="" type="checkbox"/>	style.css	css	October 12, 2023, 16:54:01 (UTC+05:30)

[CloudShell](#) [Feedback](#) © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

\*All the objects have been selected to make it public

Amazon S3 > Buckets > website-301 > Make public

**Make public** [Info](#)

The make public action enables public read access in the object access control list (ACL) settings. [Learn more](#)

⚠️ When public read access is enabled and not blocked by Block Public Access settings, anyone in the world can access the specified objects.

**Specified objects**

<input checked="" type="checkbox"/>	Name	Type	Last modified	Size
<input checked="" type="checkbox"/>	index.html	html	October 12, 2023, 17:03:07 (UTC+05:30)	10.0 KB
<input checked="" type="checkbox"/>	script.js	js	October 12, 2023, 16:54:00 (UTC+05:30)	2.0 KB
<input checked="" type="checkbox"/>	style.css	css	October 12, 2023, 16:54:01 (UTC+05:30)	3.5 KB

Cancel **Make public**

\*Successfully edited the public access for users to access the website publicly

Successfully edited public access

**Make public: status**

The information below will no longer be available after you navigate away from this page.

**Summary**

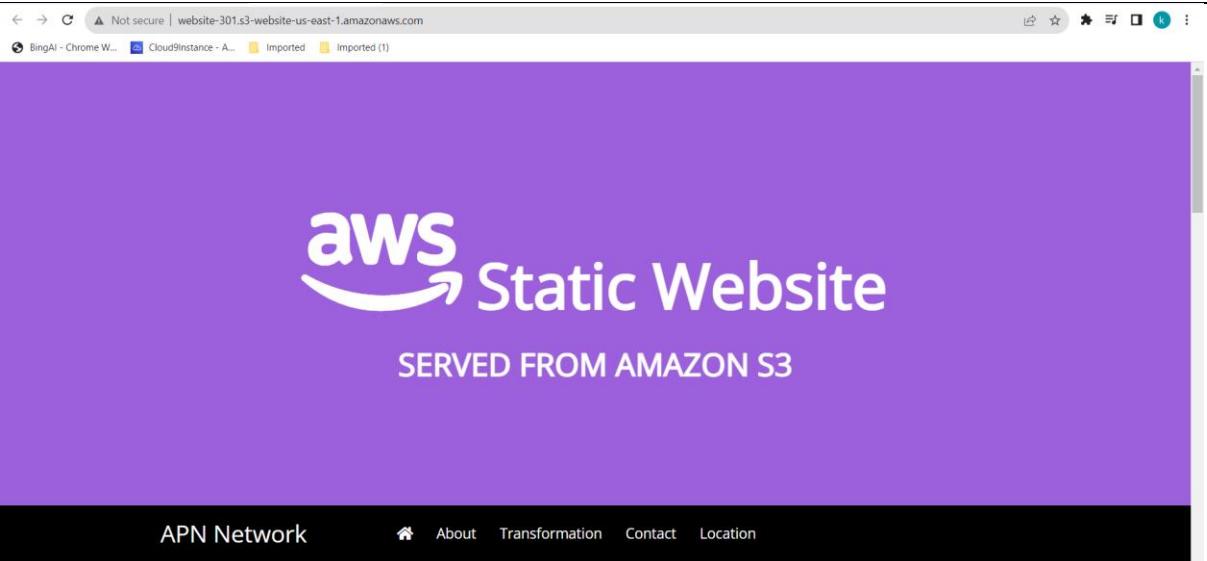
Source	Successfully edited public access	Failed to edit public access
s3://website-301	3 objects, 15.3 KB	0 objects

[Failed to edit public access](#) [Configuration](#)

**Failed to edit public access (0)**

Name	Folder	Type	Last modified	Size	Error

\*The static website created from S3 bucket objects in aws cloud



\*Now I am changing website content by modifying the code in index.html file.

\*The content served from amazon s3 will be replaced by created by kaushal.

\*After modifying code in index.html then I am replacing with old index.html and I am uploading file to s3 bucket and again we have to make public using acl to index.html object to access the static website.

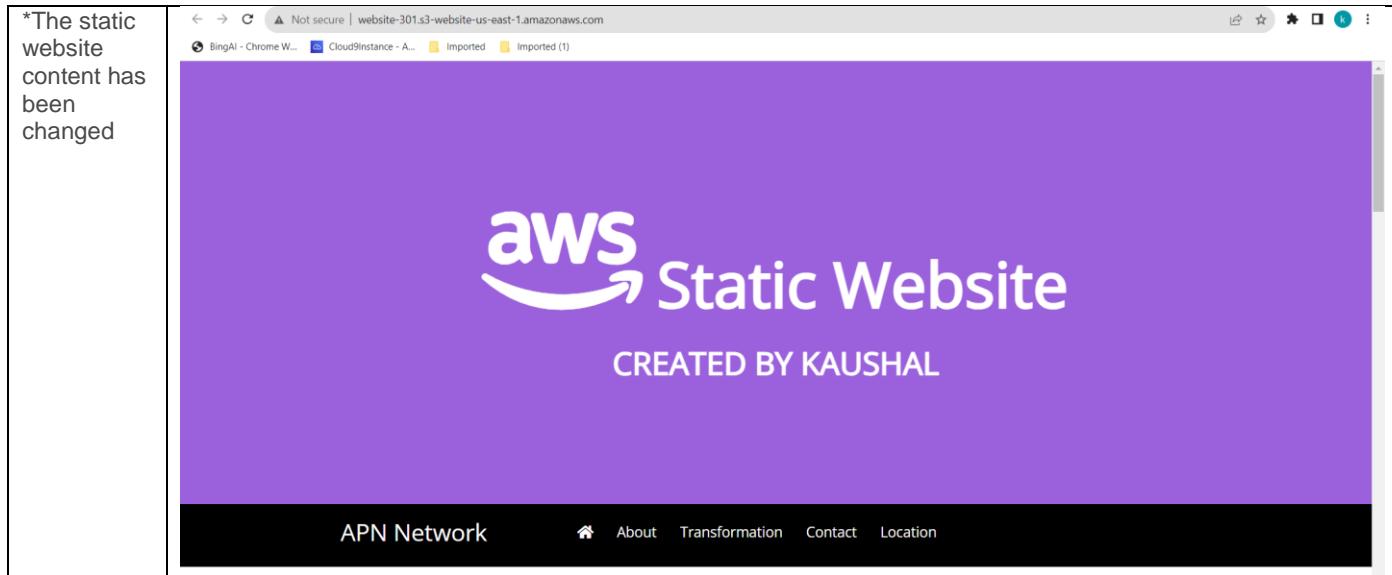
Destination	Succeeded	Failed
s3://website-301	1 file, 10.0 KB (100.00%)	0 files, 0 B (0%)

**Files and folders** (1 Total, 10.0 KB)

Name	Folder	Type	Size	Status	Error
index.html	-	text/html	10.0 KB	Succeeded	-

*The website shows error because the object in S3 bucket is not given public using ACL after replacing the file	<p>Not secure   website-301.s3-website-us-east-1.amazonaws.com</p> <p>BingAI - Chrome W... Cloud9Instance - A... Imported Imported (1)</p> <h2>403 Forbidden</h2> <ul style="list-style-type: none"> <li>Code: AccessDenied</li> <li>Message: Access Denied</li> <li>RequestID: JJAYKZ6GE5H59QJ4</li> <li>HostID: TLg4XVM0sUv3IbUejSueR0rNSEx+pXwR9Iu61jgXNrE08+zdS6nRn8S9wJEY94ocYYY5FdjMU=</li> </ul> <p>An Error Occurred While Attempting to Retrieve a Custom Error Document</p> <ul style="list-style-type: none"> <li>Code: AccessDenied</li> <li>Message: Access Denied</li> </ul>																																														
*The object has been successfully given public access using ACL	<p>aws Services Search [Alt+S]</p> <p>Successfully edited public access View details below.</p> <table border="1"> <thead> <tr> <th colspan="3">Summary</th> </tr> </thead> <tbody> <tr> <td>Source s3://website-301</td> <td>Successfully edited public access 1 object, 10.0 KB</td> <td>Failed to edit public access 0 objects</td> </tr> </tbody> </table> <p>Failed to edit public access Configuration</p> <table border="1"> <thead> <tr> <th colspan="10">Failed to edit public access (0)</th> </tr> <tr> <th colspan="10">Find objects by name</th> </tr> <tr> <th>Name</th> <th>▲</th> <th>Folder</th> <th>▼</th> <th>Type</th> <th>▼</th> <th>Last modified</th> <th>▼</th> <th>Size</th> <th>▼</th> </tr> </thead> <tbody> <tr> <td></td> </tr> </tbody> </table>	Summary			Source s3://website-301	Successfully edited public access 1 object, 10.0 KB	Failed to edit public access 0 objects	Failed to edit public access (0)										Find objects by name										Name	▲	Folder	▼	Type	▼	Last modified	▼	Size	▼										
Summary																																															
Source s3://website-301	Successfully edited public access 1 object, 10.0 KB	Failed to edit public access 0 objects																																													
Failed to edit public access (0)																																															
Find objects by name																																															
Name	▲	Folder	▼	Type	▼	Last modified	▼	Size	▼																																						

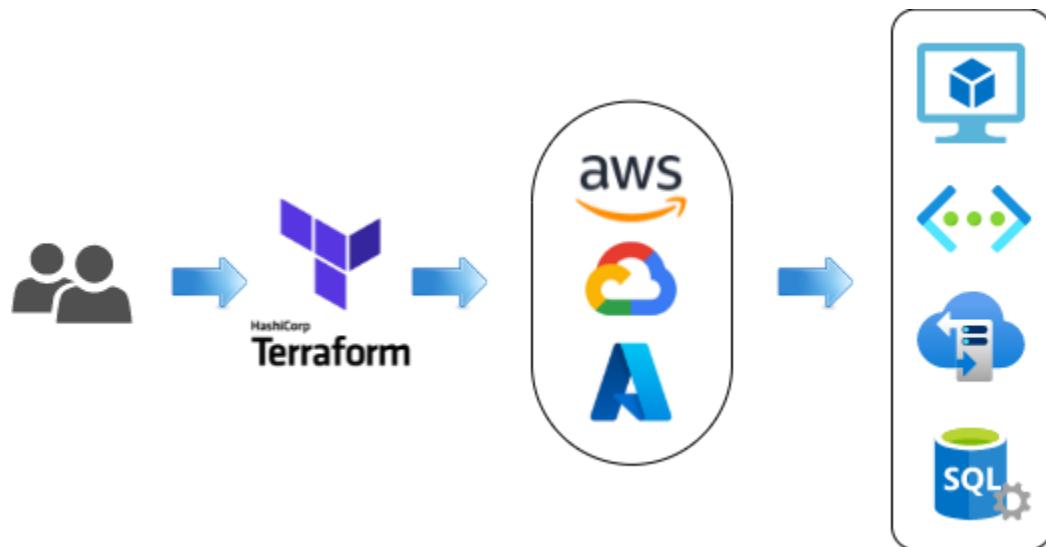
\*The static website content has been changed



### 3.Terraform Scripting:

#### Agenda:

- Terraform
- Aws cloud resources
- Terraform: is an open-source Infrastructure as Code (IaC) tool developed by HashiCorp. It enables users to define and provision infrastructure resources in a declarative configuration language. With Terraform we can describe the entire infrastructure stack—including compute instances, storage, networking, and then use that code to create and manage the infrastructure on various cloud providers or on-premises environments.



- Cloud resources: refer to the computing components and services provided by cloud computing platforms. These resources are made available over the internet and can be accessed and managed remotely. Cloud providers offer a variety of resources that organizations and individuals can leverage to build, deploy, and manage applications and infrastructure



\*Terraform  
script to create  
S3 bucket in  
cloud

```
GNU nano 6.2                                     s3bucket.tf
provider "aws" {
  region = "us-east-1"
}

resource "aws_s3_bucket" "kaushal" {
  bucket = "kaushalkorcomptenz"
  acl    = "private"

  versioning {
    enabled = true
  }
}
```

\*First the  
Terraform has  
to be initialized

```
The following dependency selections recorded in the lock file are
inconsistent with the current configuration:
- provider registry.terraform.io/hashicorp/aws: locked version selection 5.20.0 doesn't match the updated version constraints "5.20.1"

To update the locked dependency selections to match a changed configuration,
run:
  terraform init -upgrade

kaushal@kaushal-VirtualBox:~/Documents$ terraform init -upgrade

Initializing the backend...
Initializing provider plugins...
- Finding hashicorp/aws versions matching "5.20.1"...
- Installing hashicorp/aws v5.20.1...
- Installed hashicorp/aws v5.20.1 (signed by Hashicorp)

Terraform has made some changes to the provider dependency selections recorded
in the .terraform.lock.hcl file. Review those changes and commit them to your
version control system if they represent changes you intended to make.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.

kaushal@kaushal-VirtualBox:~/Documents$ terraform plan
```

\*We have to to give plan to read the configuration and the state of resources that we are going to create in cloud

```
kaushal@kaushal-VirtualBox:~/Documents$ terraform plan
Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

# aws_s3_bucket.kaushal will be created
+ resource "aws_s3_bucket" "kaushal" {
    acceleration_status      = (known after apply)
    acl                      = (known after apply)
    arn                      = (known after apply)
    bucket                   = "my-tf-test-bucket12312"
    bucket_domain_name       = (known after apply)
    bucket_prefix             = (known after apply)
    bucketRegionalDomainName = (known after apply)
    force_destroy            = false
    hostedZoneId             = (known after apply)
    id                       = (known after apply)
    objectLockEnabled        = (known after apply)
    policy                   = (known after apply)
    region                   = (known after apply)
    requestPayer             = (known after apply)
    tags                     = [
        + "Environment" = "Dev"
        + "Name"       = "My bucket"
    ]
    tags_all                 = [
        + "Environment" = "Dev"
        + "Name"       = "My bucket"
    ]
    website_domain           = (known after apply)
    websiteEndpoint          = (known after apply)
}

Plan: 1 to add, 0 to change, 0 to destroy.

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform apply" now.

kaushal@kaushal-VirtualBox:~/Documents$ terraform apply
Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
```

\*Finally we have to give apply to create resources in cloud

```
kaushal@kaushal-VirtualBox:~/Documents$ terraform apply
Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

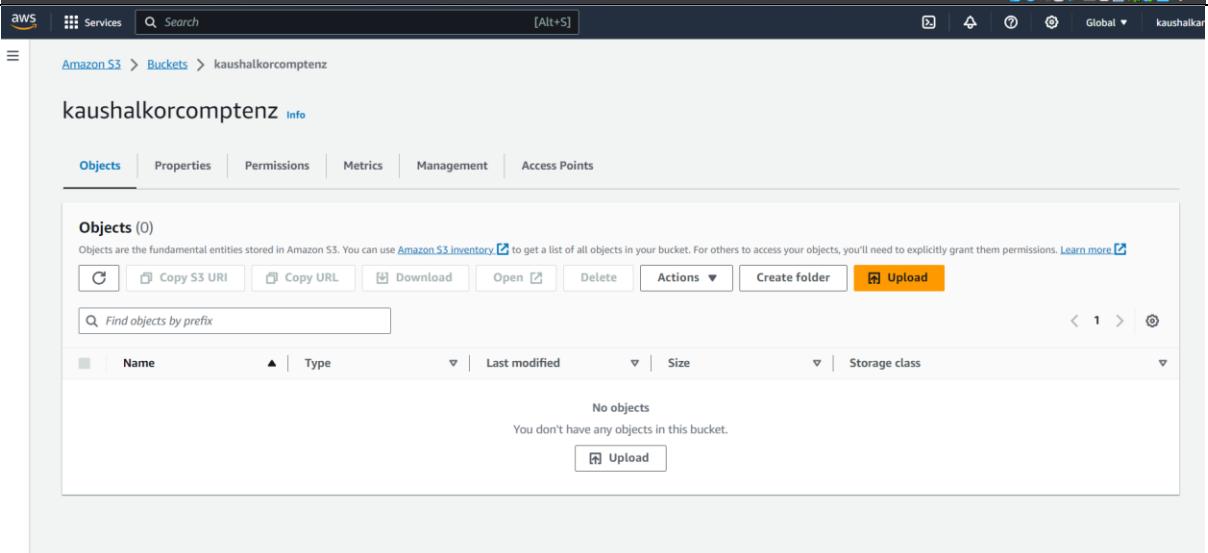
# aws_s3_bucket.kaushal will be created
+ resource "aws_s3_bucket" "kaushal" {
    acceleration_status      = (known after apply)
    acl                      = "private"
    arn                      = (known after apply)
    bucket                   = "my-tf-test-bucket12312"
    bucket_domain_name       = (known after apply)
    bucketPrefix              = (known after apply)
    bucketRegionalDomainName = (known after apply)
    forceDestroy             = false
    hostedZoneId             = (known after apply)
    id                       = (known after apply)
    objectLockEnabled        = (known after apply)
    policy                   = (known after apply)
    region                   = (known after apply)
    requestPayer             = (known after apply)
    tags                     = [
        + "Environment" = "Dev"
        + "Name"       = "My bucket"
    ]
    tags_all                 = [
        + "Environment" = "Dev"
        + "Name"       = "My bucket"
    ]
    websiteDomain           = (known after apply)
    websiteEndpoint          = (known after apply)

    + versioning {
        + enabled     = true
        + mfaDelete   = false
    }
}

Plan: 1 to add, 0 to change, 0 to destroy.

Warning: Argument is deprecated
```

\*The following S3 bucket would be created in cloud



\*Terraform  
script to create  
ec2 instance

```
GNU nano 6.2                                     ec2.tf
provider "aws" {
region = "us-east-1"
}
resource "aws_instance" "my_instance" {
ami           = "ami-0df435f331839b2d6"
instance_type = "t2.micro"
tags = [
  "Name" = "MyEC2Instance"
]
```

\*Terraform  
script to create  
instance in  
airticketbooking

```
GNU nano 6.2                                     airticketinstance.tf
provider "aws" {
region = "us-east-1"
}

resource "aws_security_group" "web" {
  name          = "air_ticket_booking_sg"
  description   = "Security group for Air Ticket Booking EC2 instance"

  ingress {
    from_port   = 80
    to_port     = 80
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }
}

resource "aws_instance" "air_ticket_booking" {
  ami           = "ami-055859c8e0f361065"
  instance_type = "t3.micro"
  key_name      = "key-09e0bb1691a4c7b2"
  subnet_id     = "subnet-0b5ef622a2bb61011"
  security_groups = ["sg-0891c98e0931de380"]
{
  Inbound rules = ["sgr-0f41dca8e534b4f42"]
  Outbound rules = ["sgr-03daad6c8f048733e"]
}
}

resource "aws_security_group" "allow_ssh" {
  name          = "allow-ssh"
  description   = "Allow SSH traffic"
```

\* Terraform  
script to create  
instance in  
airticketbooking

```
GNU nano 6.2                                     airticketinstance.tf

}

resource "aws_instance" "air_ticket_booking"
{
  ami          = "ami-055859c8e0f361065"
  instance_type = "t3.micro"
  key_name      = "key-09e0eb81691a4c7b2"
  subnet_id     = "subnet-0b5ef622a2bb61011"
  security_groups = ["sg-0891c98e0931de380"]
}

Inbound rules  = ["sgr-0f41dca8e534b4f42"]
Outbound rules = ["sgr-03daad6c8f048733e"]
}

resource "aws_security_group" "allow_ssh" {
  name          = "allow-ssh"
  description   = "Allow SSH traffic"
  vpc_id        = "vpc-04615e6acb7d30ef3"

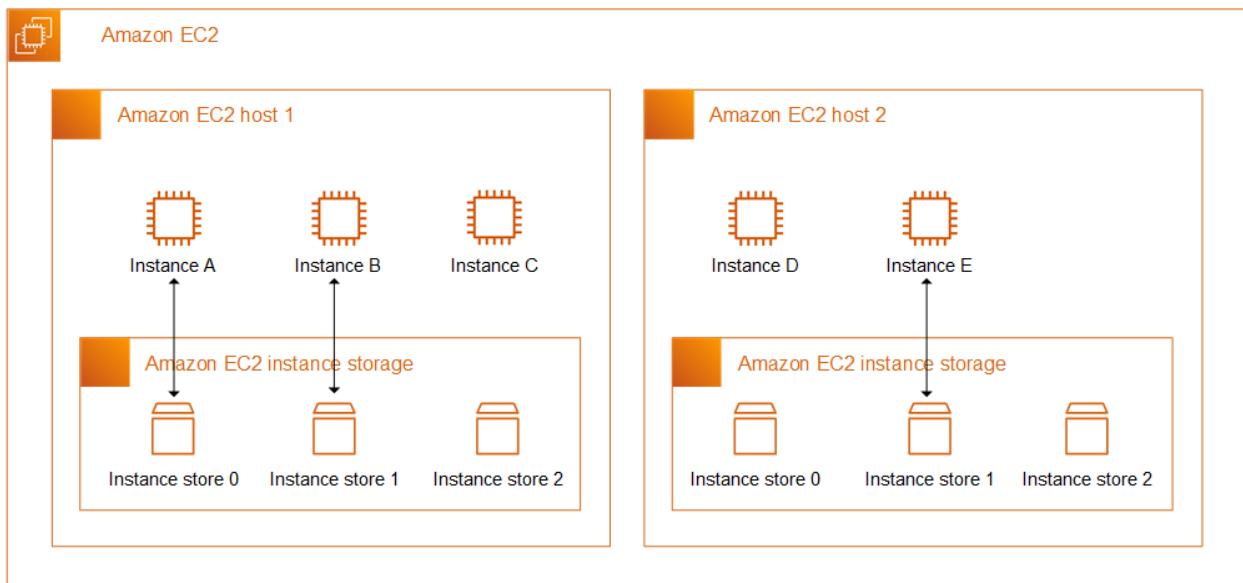
  ingress {
    from_port   = 22
    to_port     = 22
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }
}

output "instance_public_ip" {
  value = aws_instance.air_ticket_booking.public_ip
}
|
```

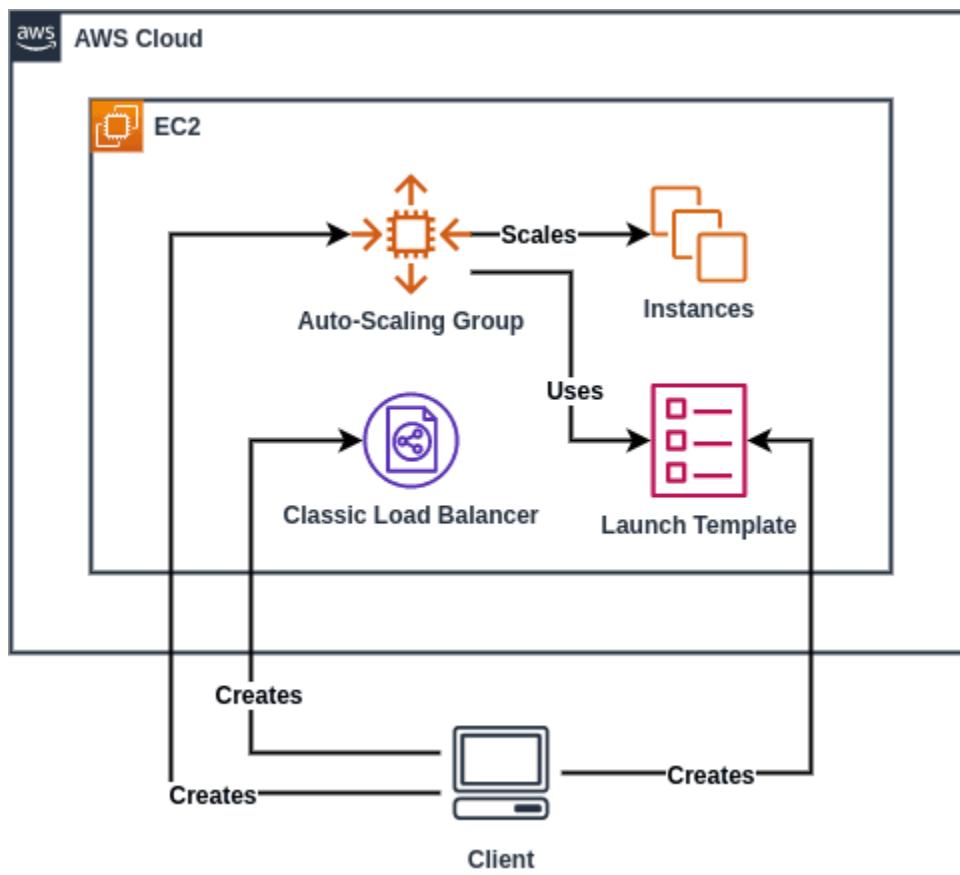
## 4.Launch templates on ec2 instance with role associated

### Agenda:

- EC2 Instance
- Launch Templates
- IAM Role
- EC2 Instance: a web service provided by Amazon Web Services (AWS) that allows users to rent virtual servers known as instances in the cloud. EC2 instances provide scalable computing capacity and are designed to run various types of applications from simple to complex workloads. Users can choose from a variety of instance types where each instance type optimized for different use cases.



- Launch Templates: are a feature that allows users to define a set of parameters and configurations for launching EC2 instances. Launch Templates streamline the process of launching instances by providing a template that includes various specifications, such as the Amazon Machine Image (AMI), instance type, key pair, security groups, block device mappings, and other settings.



- IAM Role: are a fundamental component of the AWS security model. IAM roles define a set of permissions that determine what actions users, applications, or AWS services can perform within your AWS environment. Unlike IAM users or groups, IAM roles are not associated with a specific identity. Instead roles can be assumed by IAM users, AWS services, or federated identities.



\*Create a instance with instance type,vpc,subnets and iam role to be associated with it

**Instance summary for i-0851bb22d6fdef695 (kkinstance) [Info]**

Updated less than a minute ago

Instance ID	i-0851bb22d6fdef695 (kkinstance)	Public IPv4 address	18.232.176.3 [open address]
IPv6 address	-	Instance state	Running
Hostname type	IP name: ip-172-31-28-228.ec2.internal	Private IP DNS name (IPv4 only)	ip-172-31-28-228.ec2.internal
Answer private resource DNS name	IPv4 (A)	Instance type	t2.large
Auto-assigned IP address	18.232.176.3 [Public IP]	VPC ID	vpc-04615e6acb7d30ef3
IAM Role	kaushalrole	Subnet ID	subnet-034f718e27409c4cc
IMDSv2	Required		

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

\*Once the instance have been created then create template from the instance

**Instances (1/3) [Info]**

Find Instance by attribute or tag (case-sensitive)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status
kkinstance	i-0851bb22d6fdef695	Running	t2.large	2/2 checks passed	No alarms
kkinstance	i-0e26c723968570629	Terminated	t2.micro	-	No alarms
kkinstance	i-0d679d7af84fe3fb	Running	t2.micro	2/2 checks passed	No alarms

**Actions ▾ | Launch instances ▾**

- Connect
- View details
- Manage instance state
- Instance settings
- Networking
- Security
- Image and templates
- Monitor and troubleshoot

**Create image**  
**Create template from instance**  
**Launch more like this**

\*Specify the version of template that we have to launch from the instance

**EC2 > Launch templates > ourt2microinstance**

**ourt2microinstance (lt-04e74df6bf1f53673)**

**Actions ▾ | Delete template**

**Launch template details**

Launch template ID	lt-04e74df6bf1f53673	Launch template name	ourt2microinstance	Default version	1	Owner	arn:aws:iam::256772097600:root
--------------------	----------------------	----------------------	--------------------	-----------------	---	-------	--------------------------------

**Details | Versions | Template tags**

**Launch template version details**

Version	2	Description	v1	Date created	2023-10-11T07:50:24.000Z	Created by	arn:aws:iam::256772097600:root
	1 (Default)	Resource tags	Network interfaces	Advanced details			

**Spot purchasing options**

IAM instance profile	Instance auto-recovery	Shutdown behavior	Stop-hibernate behavior
default	default	stop	false

\*The template has been launched with its latest version.

\*We can also launch instance from template

\*These are the instances launched from the template

The screenshot shows two screenshots of the AWS EC2 console side-by-side.

**Left Screenshot (Launch Templates):**

- Shows the "Launch Templates (1/1)" page.
- Table header: Launch Template ID, Launch Template Name, Default Version, Latest Version.
- Single entry: lt-04e74df6bf1f53673, ourt2microinstance, 1, 2.
- Details pane for "ourt2microinstance (lt-04e74df6bf1f53673)":
  - Launch template name: lt-04e74df6bf1f53673
  - Default version: 1
  - Owner: arn:aws:iam::256772097600:root

**Right Screenshot (Instances):**

- Shows the "Instances (3) Info" page.
- Table header: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, Public IPv4 DNS.
- Three instances listed:
  - kkinstance (i-0851bb22d6fdef695), Running, t2.large, 2/2 checks passed, No alarms, us-east-1d, ec2-18-232-176-
  - kkinstance (i-0e26c723968570629), Terminated, t2.micro, -, No alarms, us-east-1d, -
  - kkinstance (i-0d679d7af84fe3fbc), Running, t2.micro, 2/2 checks passed, No alarms, us-east-1d, ec2-54-144-202-

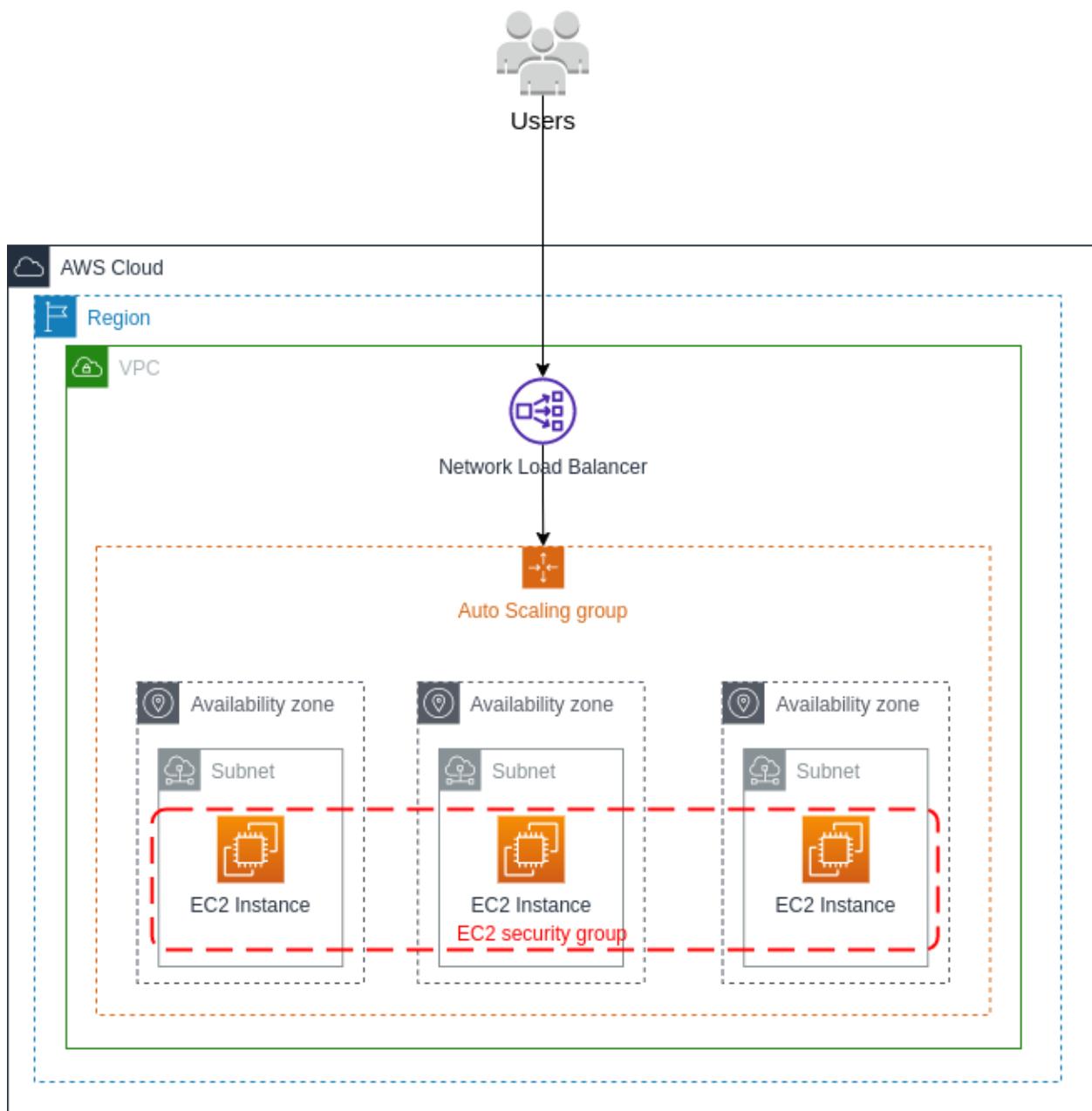
## 5.Air ticket booking system project in aws cloud

When building an air ticket booking system in the AWS Cloud. We would typically use a combination of AWS services rather than traditional "instances." Here are key components and AWS services to consider for such a project:

Agenda:

- EC2 Instances
- Relational Database Service(RDS)
- S3
- Route 53
- Elastic Beanstalk
- CloudFront
- Lambda
- Cognito
- Simple Email Service(SES)
- Step Functions
- CloudWatch
- Identity and Access Management (IAM)
- API Gateway
- DynamoDB
- Simple Notification Service(SNS)
- Simple Queue Service(SQS)
- Redshift
- Polly

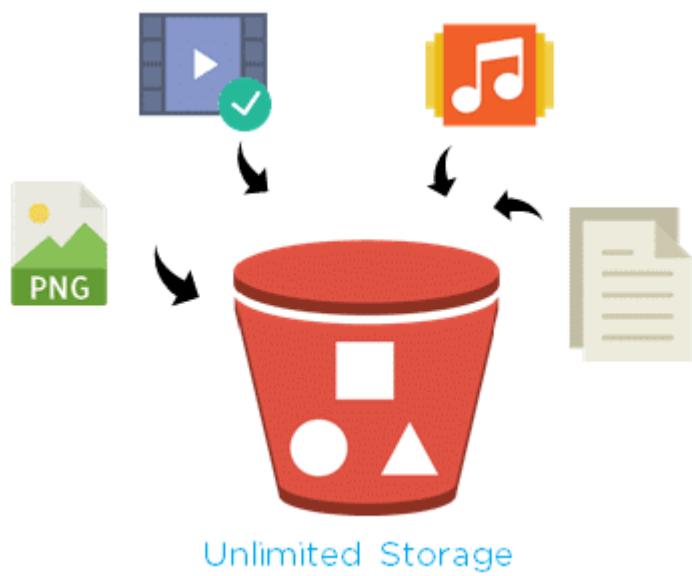
- EC2 Instances: can play a crucial role in providing the necessary computing resources to support the various components of the system. It can host web servers to serve the booking system's front-end application for users to get access to it.



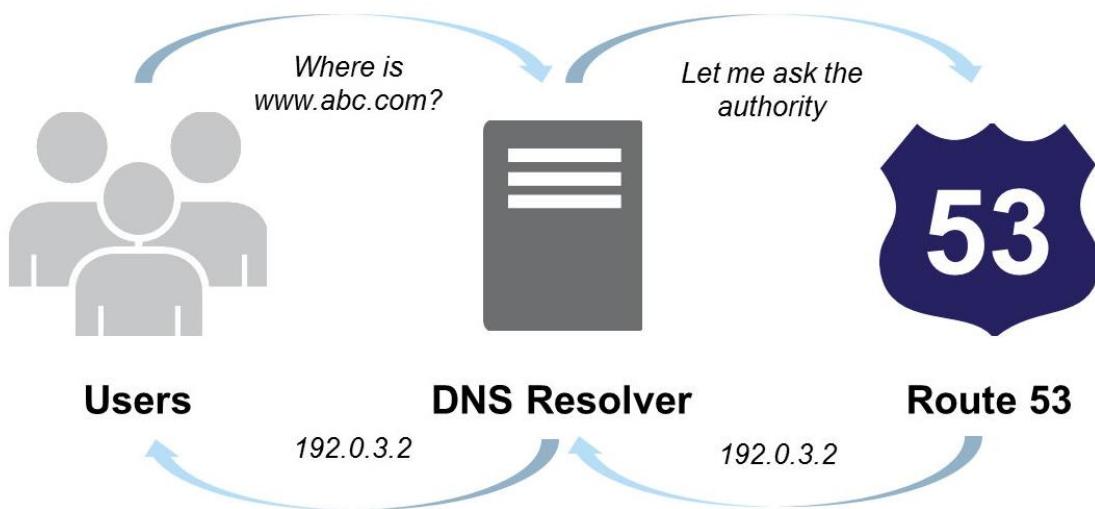
- Relational Database Service(RDS): is a managed database service provided by Amazon Web Services (AWS). It simplifies the process of setting up, operating, and scaling a relational database in the cloud. RDS supports several popular database engines, making it easier for users to deploy and manage databases without the need to handle the administrative tasks associated with database management.



- S3: is designed to store and retrieve any amount of data from anywhere on the web and is commonly used for a variety of purposes including data storage, backup and recovery, data archiving, content distribution and as a foundation for various AWS services.



- Route 53: is a scalable and highly available Domain Name System (DNS) web service provided by Amazon Web Services (AWS). DNS is a fundamental component of the internet that translates user-friendly domain names (like www.example.com) into IP addresses that computers use to identify each other on the network. Route 53 not only offers domain registration services but also provides DNS and health checking services.



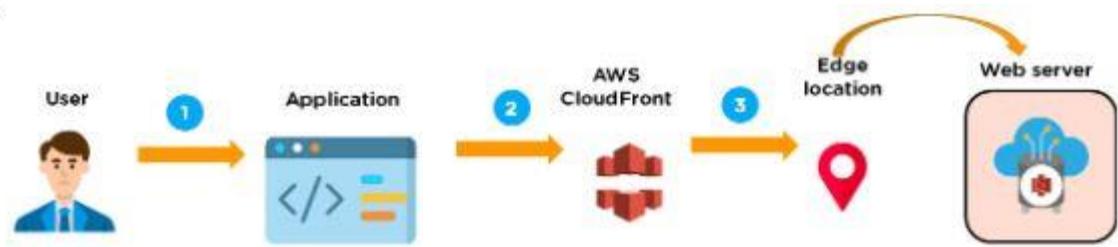
- Elastic Beanstalk: is a fully managed service provided by Amazon Web Services (AWS) that makes it easy to deploy and run applications in multiple programming languages, including Java, .NET, Python, Node.js, Ruby, Go, and more. Elastic Beanstalk abstracts the underlying infrastructure and automates the deployment, scaling, and management of applications, allowing developers to focus on writing code rather than managing the underlying infrastructure.

## AWS Elastic Beanstalk (EB)

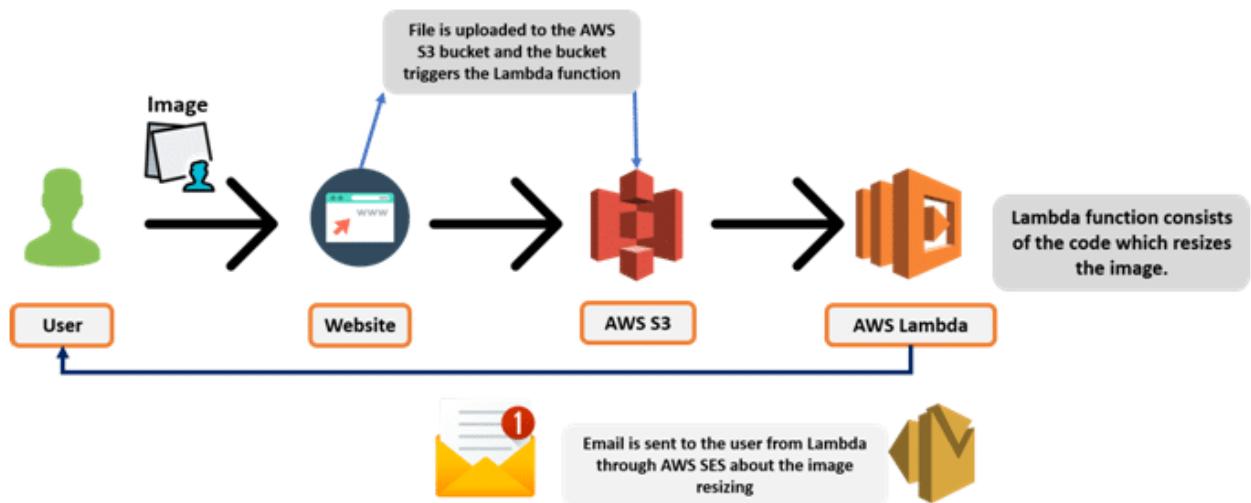
- Easily deploy, monitor, and scale three-tier applications
- Infrastructure provisioned and managed by EB
  - you maintain complete control.
- Preconfigured application containers
  - easily customizable.
- Support for these platforms:

The diagram illustrates the concept of deploying an application. A green cube represents the application code, which is being placed onto a grey three-tier application stack. The stack consists of three interconnected layers: a bottom layer of four grey cubes, a middle layer of three grey cubes, and a top layer of one grey cube. To the left of the stack, there is a collection of application platform logos: Microsoft .NET (blue square), PHP (blue oval), Java (red square with a flame), Ruby (red gem), Docker (blue whale icon), Node.js (green hexagon), and Python (yellow Python logo).

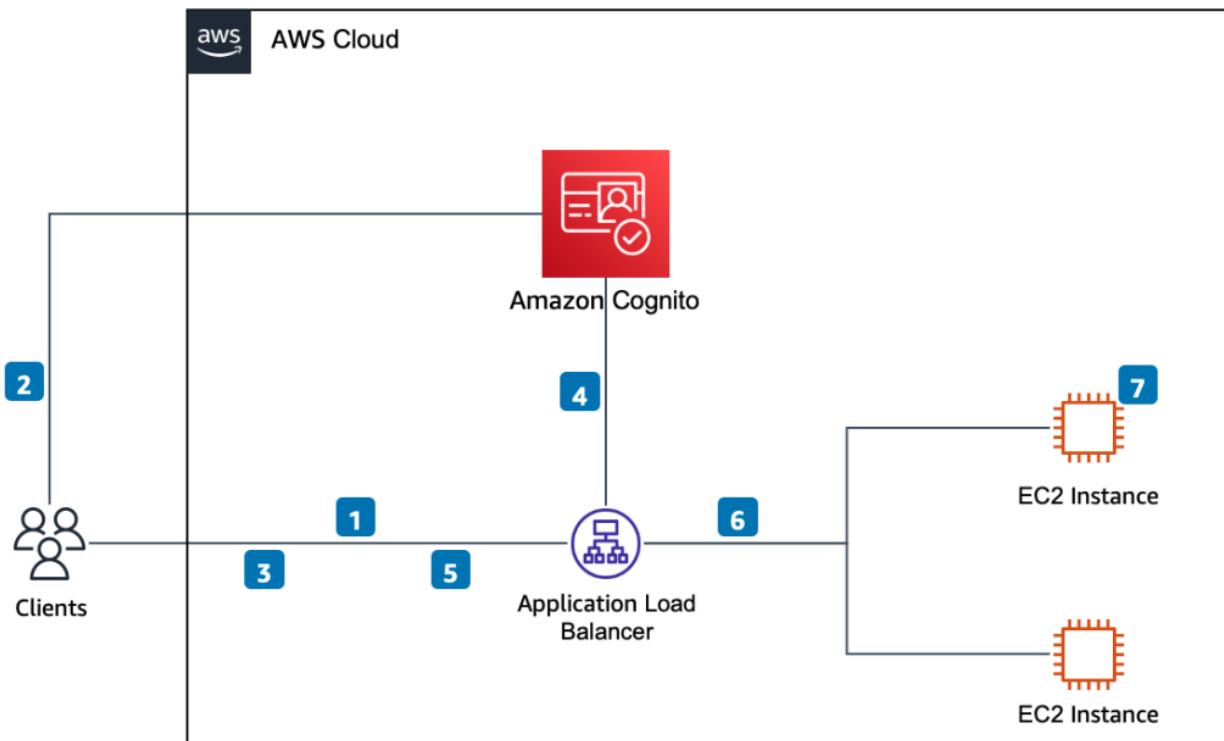
- Cloudfront: is a content delivery network (CDN) service provided by Amazon Web Services (AWS). CloudFront helps deliver static and dynamic web content, including images, videos, scripts and other files to users with low latency and high transfer speeds. It is designed to enhance the performance, reliability and scalability of web applications and websites by caching content at edge locations distributed around the world.



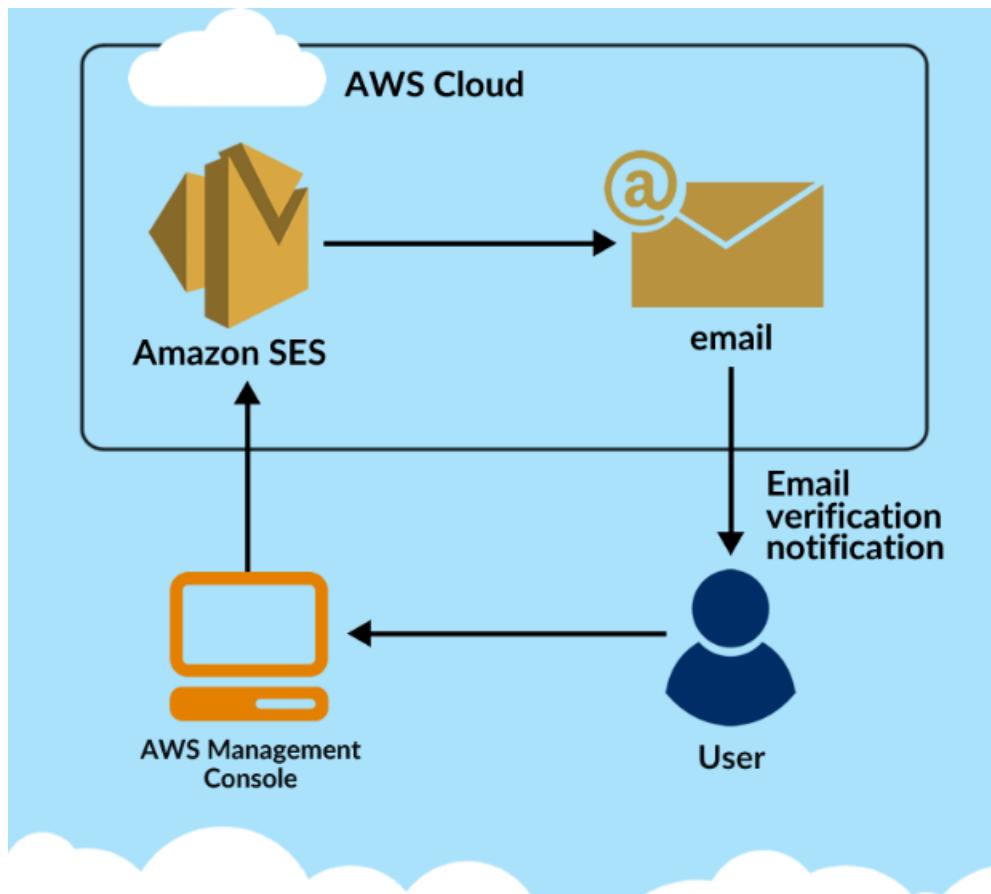
- Lambda: is a serverless compute service provided by Amazon Web Services (AWS). Lambda allows you to run code without provisioning or managing servers. With Lambda, you can execute code in response to events and automatically manage the compute resources needed to run that code. This serverless model eliminates the need to worry about infrastructure, allowing developers to focus solely on their application code.



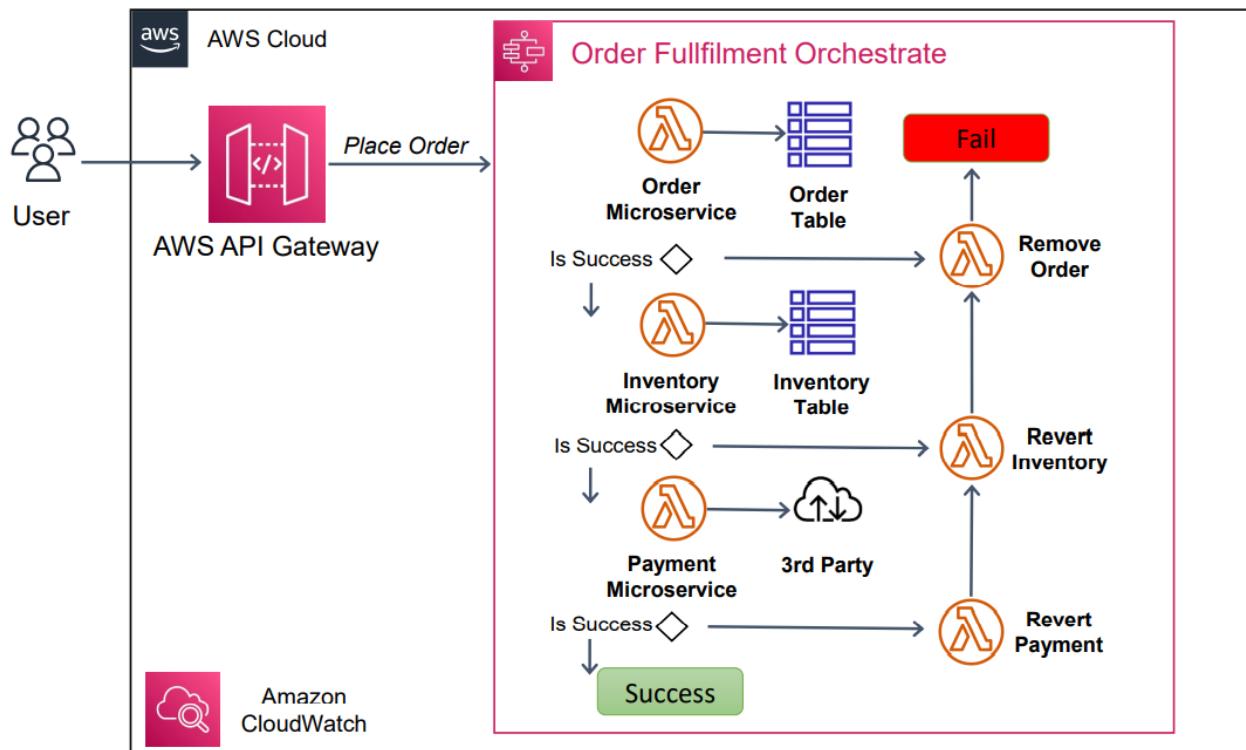
- Cognito: is a fully managed identity service provided by Amazon Web Services (AWS) that makes it easy to add authentication, authorization and user management to your applications. It enables you to securely authenticate users and manage their identities for both web and mobile applications.



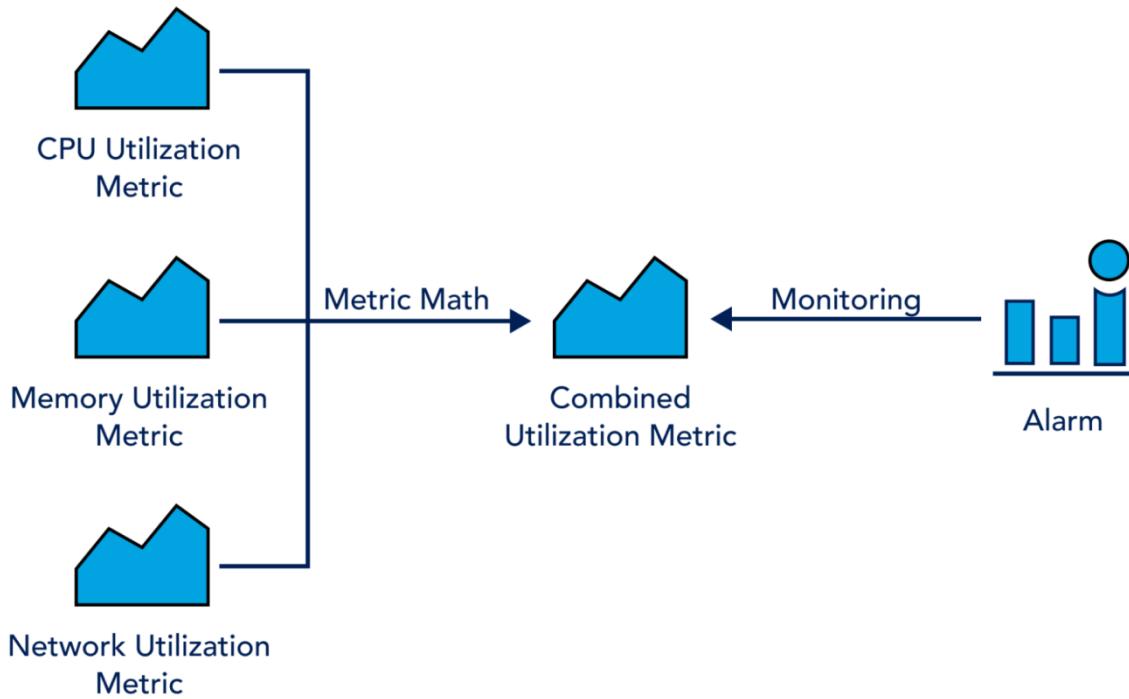
- Simple Email Service(SES): is a fully managed email sending service provided by Amazon Web Services (AWS). SES is designed to help businesses and developers send emails securely, reliably, and at scale. It allows you to send marketing emails, transactional messages, and other types of communications to your customers.



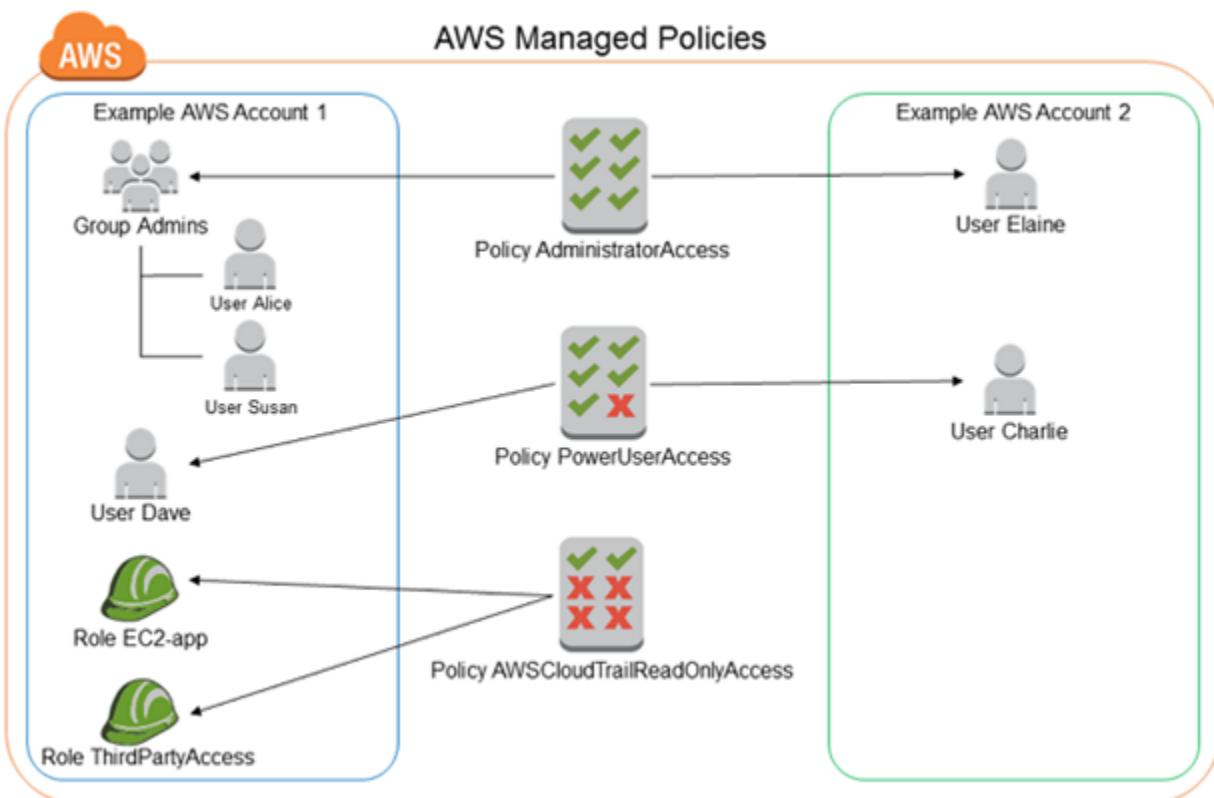
- Step Functions: is a fully managed AWS service that makes it easy to coordinate the components of distributed applications and microservices using visual workflows. Step Functions allows you to design and run workflows that integrate services such as AWS Lambda, AWS Fargate, and more.



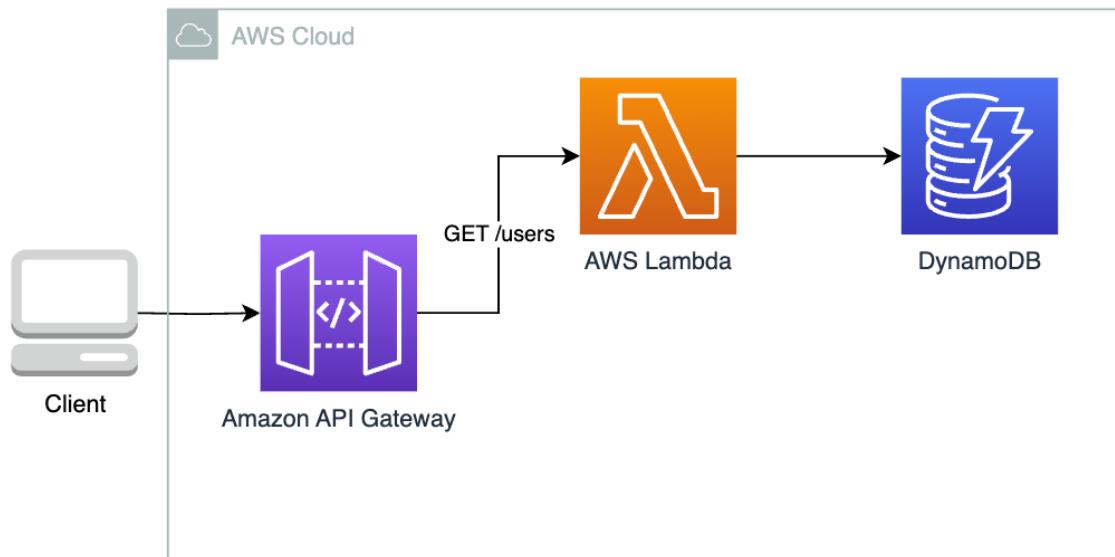
- CloudWatch: is a monitoring and observability service provided by Amazon Web Services (AWS). It is designed to collect and track various metrics, collect and monitor log files and set alarms. CloudWatch allows you to gain insights into your applications, systems and AWS resources helping you ensure the performance, availability and reliability of your applications.



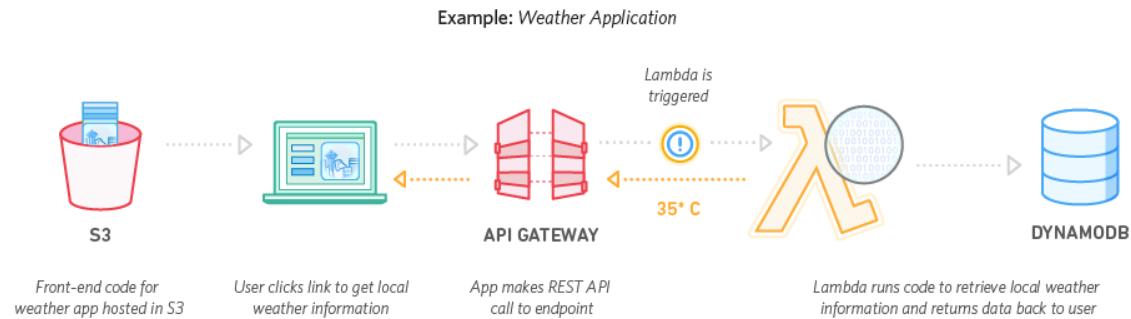
- Identity and Access Management(IAM) role: in AWS (Amazon Web Services) are a fundamental part of AWS security. IAM roles define a set of permissions for making AWS service requests and are not associated with a specific user or group. Instead IAM roles are meant to be assumed by trusted entities such as AWS services, applications or users.



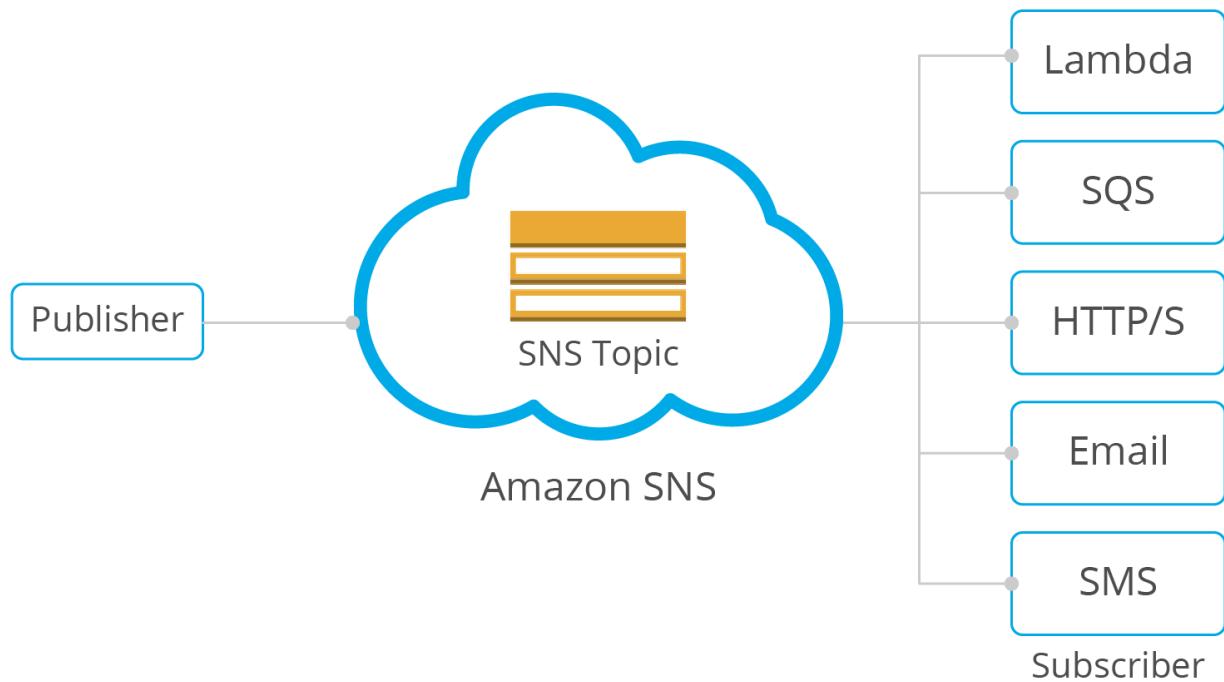
- API Gateway: is a fully managed service in AWS that makes it easy for developers to create, publish, maintain, monitor and secure APIs (Application Programming Interfaces) at any scale. It acts as a gateway for applications to connect with backend services and allows you to create, deploy, and manage APIs quickly and efficiently.



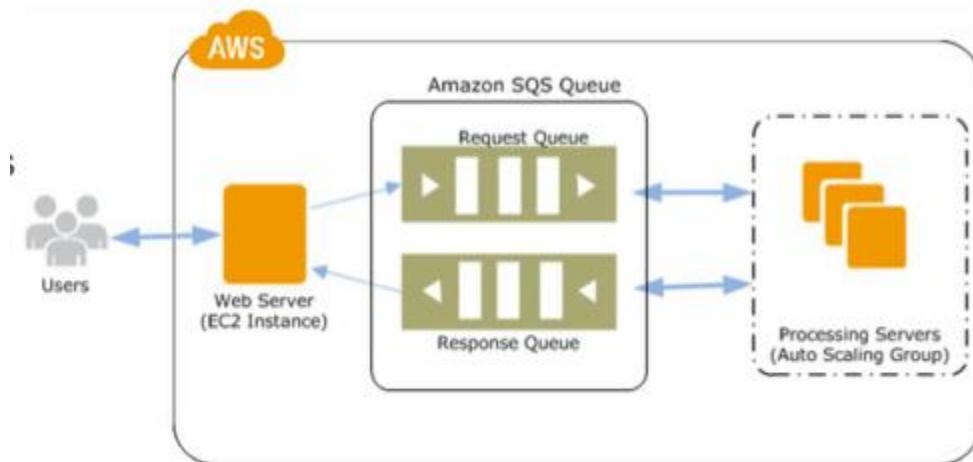
- DynamoDB: is a fully managed NoSQL database service provided by Amazon Web Services (AWS). It is designed to provide fast and predictable performance with seamless scalability. DynamoDB is a key-value and document database that can be used for a wide range of applications, from simple web and mobile applications to complex enterprise-level solutions.



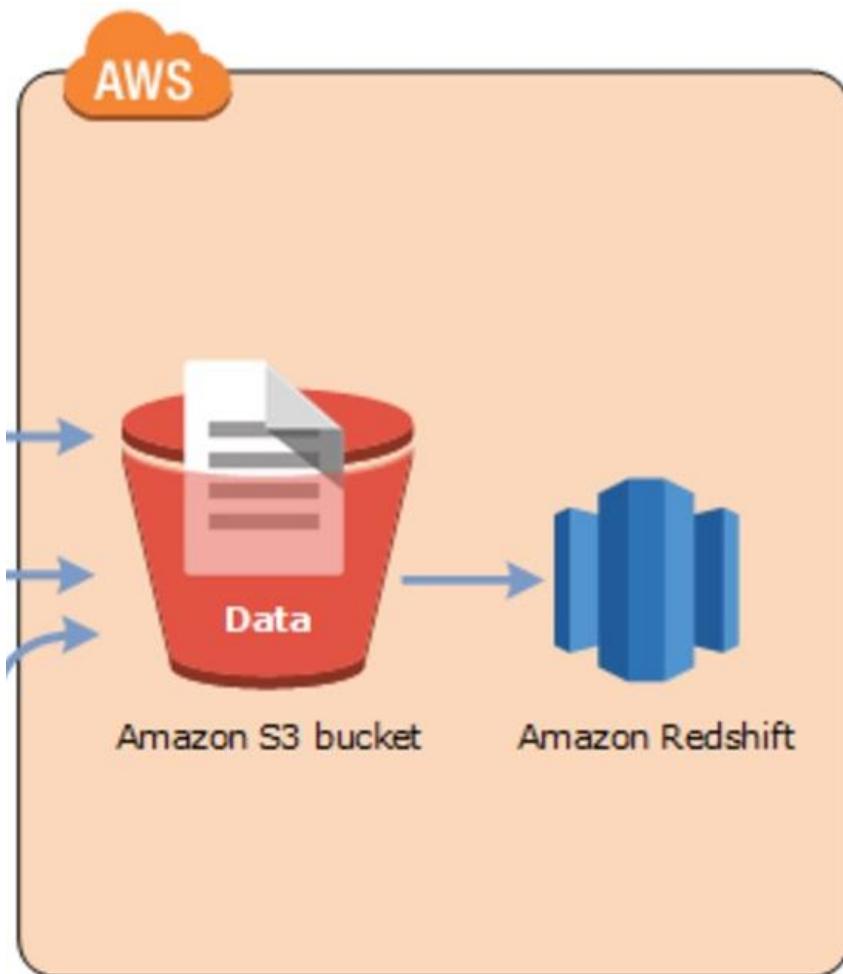
- Simple Notification Service(SNS): is a fully managed messaging service provided by Amazon Web Services (AWS). It enables you to send messages or notifications to a distributed set of recipients via various communication protocols, such as SMS, email, Amazon Simple Queue Service (SQS), AWS Lambda, and more. SNS simplifies the process of sending messages to a large number of subscribers or endpoints.



- Simple Queue Service (SQS): is a fully managed message queuing service provided by Amazon Web Services (AWS). SQS enables decoupling of the components of a cloud application allowing them to operate independently and asynchronously. It provides a reliable and scalable way to transmit any volume of data between distributed systems, applications and microservices.



- Redshift: is a fully managed petabyte-scale data warehouse service in the cloud provided by Amazon Web Services (AWS). It is designed to handle large datasets and perform complex queries for analytics and business intelligence purposes. Redshift is based on a scalable, columnar storage architecture that allows for efficient querying and analysis of structured data.



- Polly: is a service provided by Amazon Web Services (AWS) that turns text into lifelike speech. It is a text-to-speech (TTS) service that uses advanced deep learning technologies to convert written text into spoken words. Polly supports a variety of languages and voices allowing developers to create applications with natural-sounding and expressive speech.



\*You can use EC2 instances for running the backend application and handling business logic.

The screenshot shows the AWS EC2 Instances page. The left sidebar includes options like EC2 Dashboard, EC2 Global View, Events, Instances (selected), Instances Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images (AMIs, AMI Catalog), and Elastic Block Store (Volumes, Snapshots, Lifecycle Manager). The main content area displays a table titled 'Instances (1/3) Info' with three rows. The first row is for an unnamed instance (i-017761d5eac50d4ee) in us-east-1c, the second for 'Airticketapp-env' (i-055b19cc20d854d06) in us-east-1c, and the third for 'airticket' (i-03ba285cfa23ee008) in us-east-1a. The 'airticket' instance is currently selected. The details for this instance are shown in the expanded view below, including its Instance ID, Public IPv4 address (34.195.218.100), Private IP address (172.31.103.124), and Public IPv4 DNS (ec2-34-195-218-100.compute-1.amazonaws.com).

\*Store data such as flight schedules, customers information and bookings in RDS (Relational Database Service).

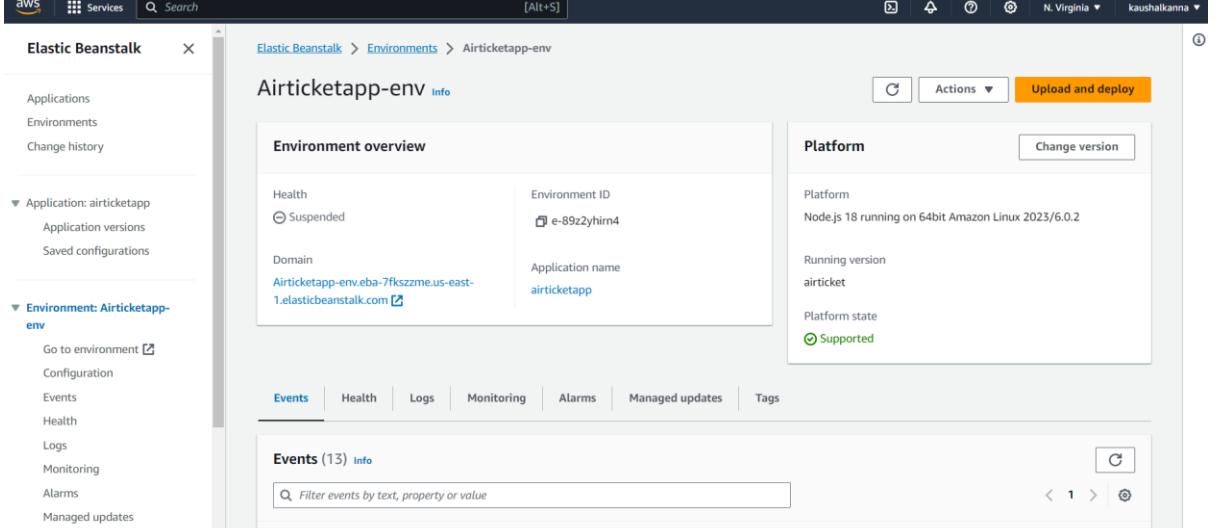
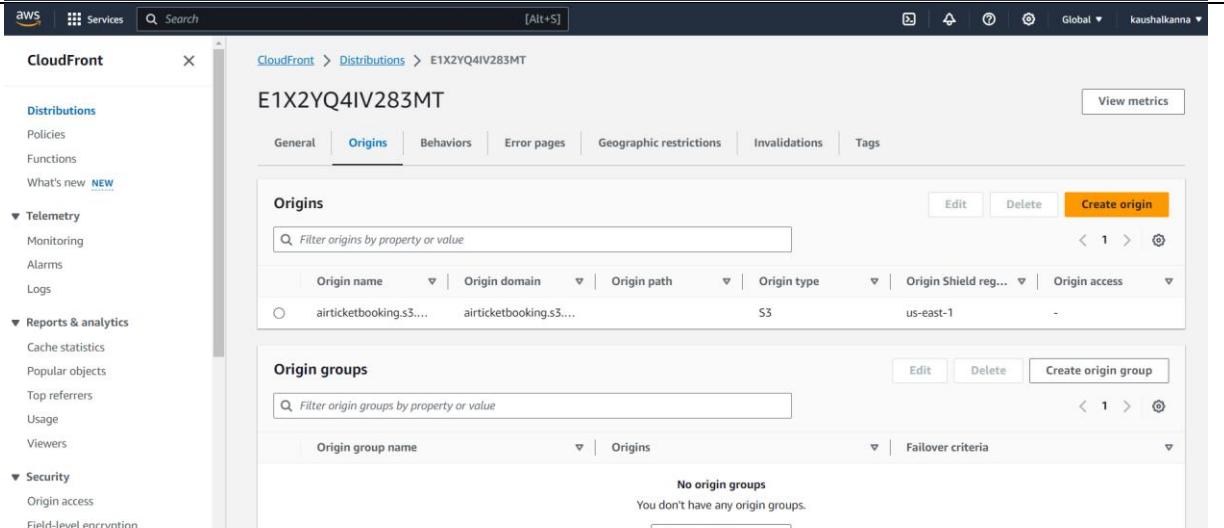
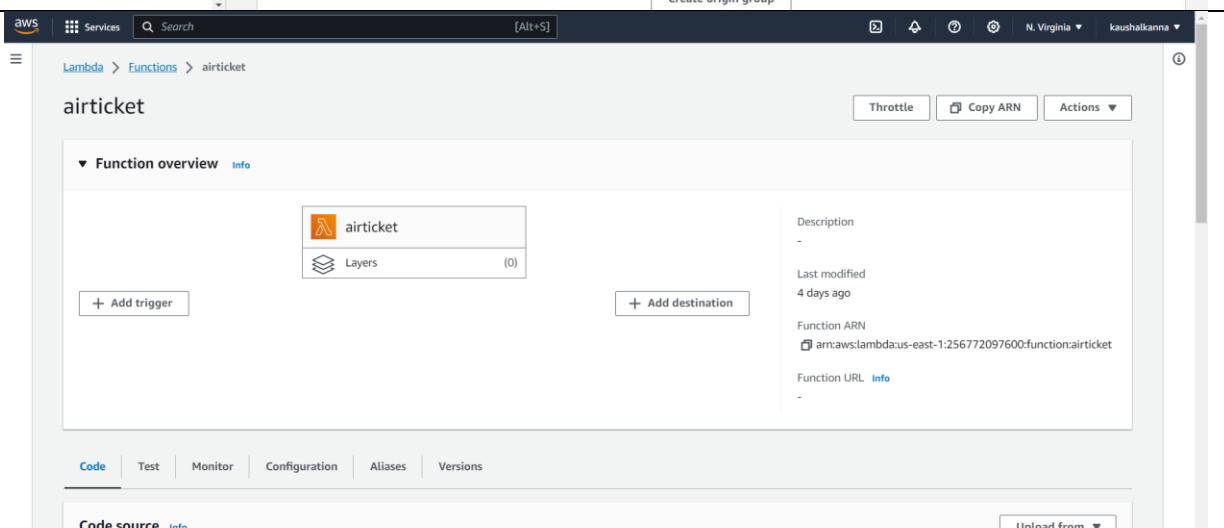
The screenshot shows the AWS RDS Databases page. The left sidebar lists options: Dashboard, Databases (selected), Query Editor, Performance insights, Snapshots, Exports in Amazon S3, Automated backups, Reserved instances, Proxies, Subnet groups, Parameter groups, Option groups, Custom engine versions, Zero-ETL integrations (New), Events, and Event subscriptions. The main content area shows the 'airticketdatabase' instance under the 'Summary' tab. Key details include: DB identifier (airticketdatabase), CPU usage (2.66%), Status (Available), Class (db.t3.micro), Role (Instance), Current activity (0 Connections), Engine (MySQL Community), Region & AZ (us-east-1a), and VPC security group (default sg-0891c98e0931de580). Other tabs available include Connectivity & security, Monitoring, Logs & events, Configuration, Maintenance & backups, and Tags.

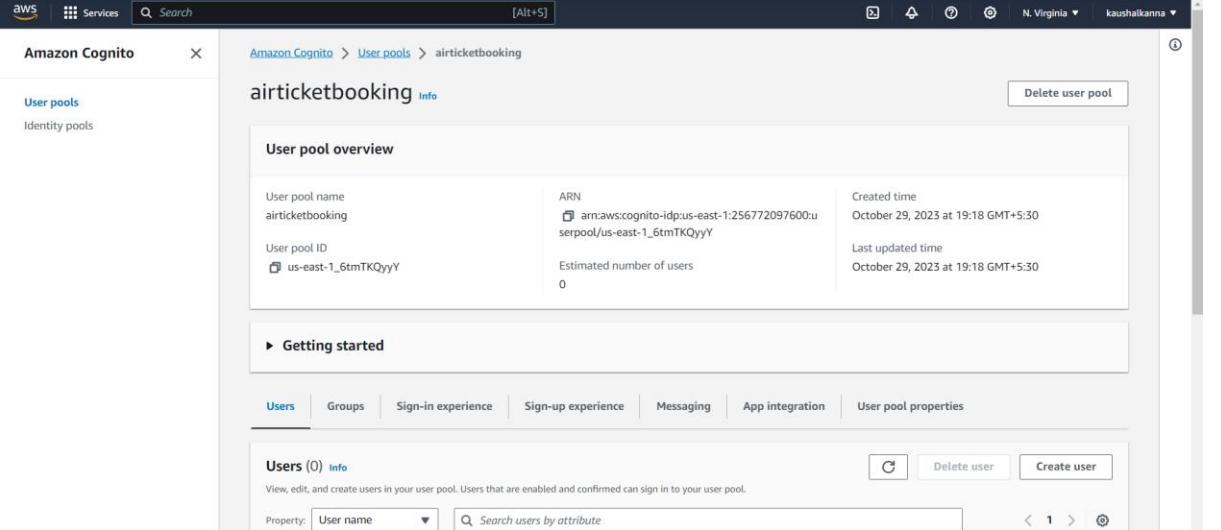
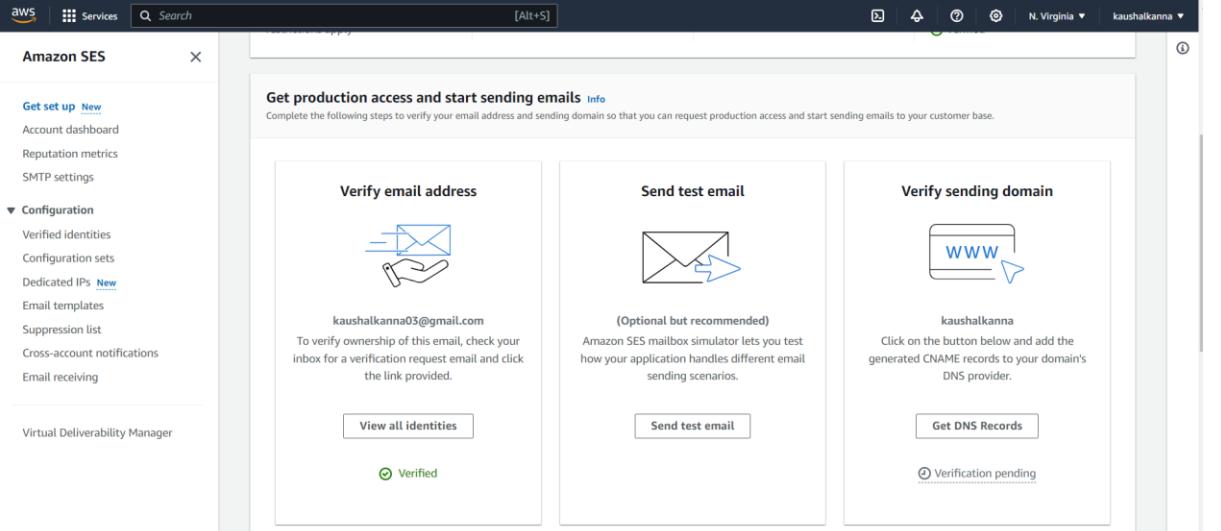
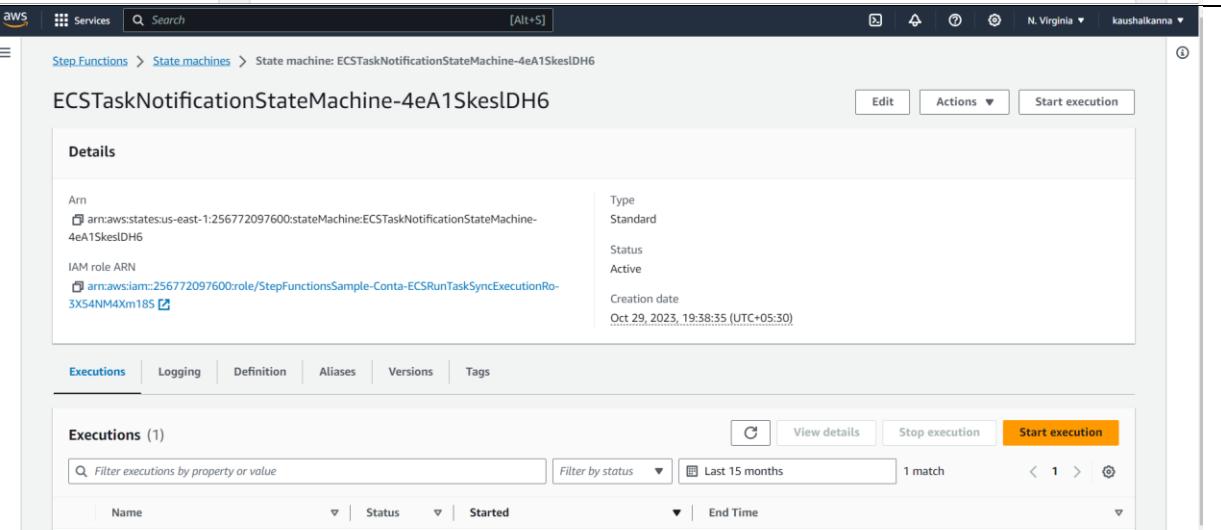
\*Store static assets like images and styles in S3 for efficient content delivery.

Name	Type	Last modified	Size	Storage class
airticket.css	css	October 25, 2023, 13:41:36 (UTC+05:30)	1.2 KB	Standard
airticket.html	html	October 25, 2023, 13:41:38 (UTC+05:30)	1.0 KB	Standard
airticket.js	js	October 25, 2023, 13:41:37 (UTC+05:30)	1.0 KB	Standard
ea8bbf9b-56c6-49d5-9d02-5cadff62b5387.mp3	mp3	October 29, 2023, 20:54:26 (UTC+05:30)	20.4 KB	Standard

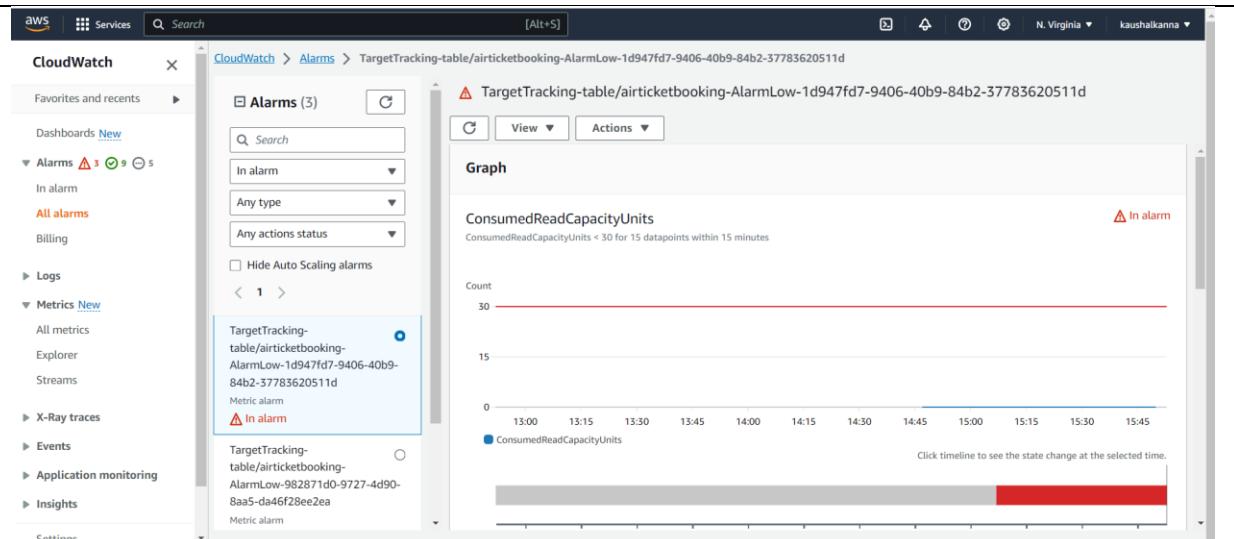
\*Use Route 53 for domain registration and DNS routing to your application

Record ...	Type	Routing policy	Alias	TTL (s...)	Health ...
airticket...	NS	Simple	-	No	ns-1549.awsdns-01.co.uk. ns-107.awsdns-13.com. ns-1319.awsdns-36.org. ns-678.awsdns-20.net.
airticket...	SOA	Simple	-	No	ns-1549.awsdns-01.co.uk. a... 900

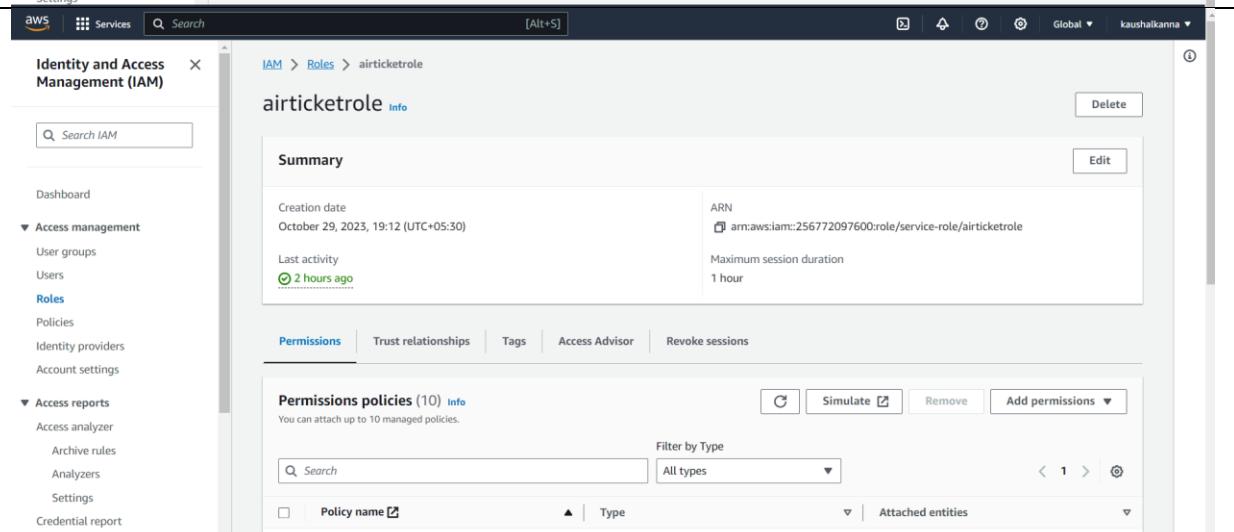
*Deploy the application using Elastic Beanstalk for easy management of the application stack.	
*Set up CloudFront for a Content Delivery Network to improve content delivery and reduce latency	
*Use Lambda for serverless functions that can be triggered for specific tasks such as sending email confirmations	

*Implement Cognito for user authentication and authorization	
*Use SES(Simple Email Service) For sending transactional and promotional emails.	
*Use Step Functions to create workflows for tasks like booking confirmation and payment processing.	

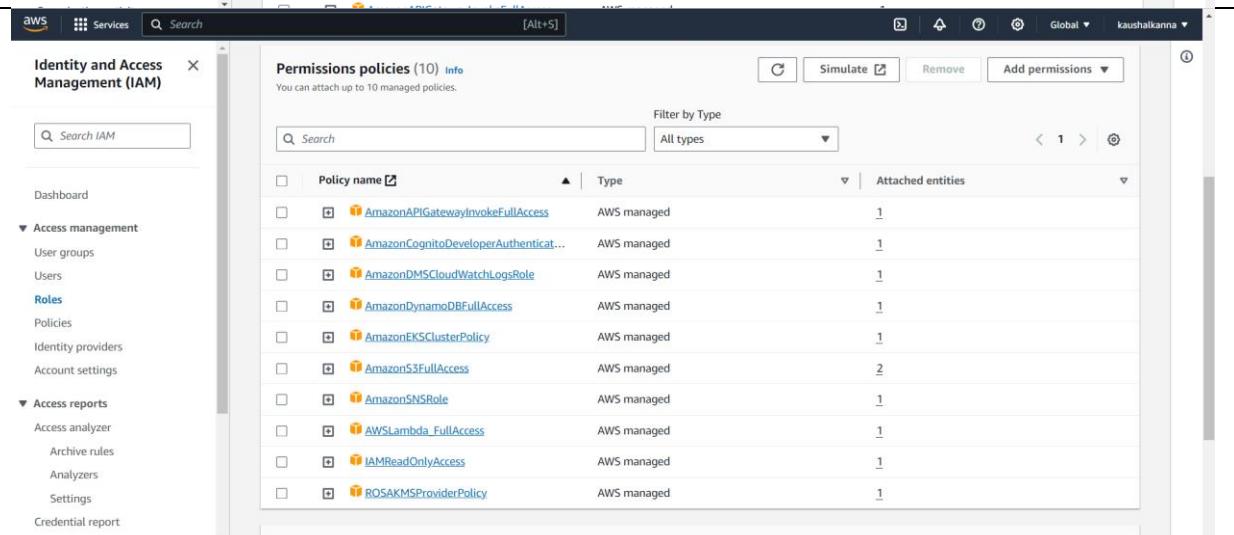
\*Set up CloudWatch for monitoring and logging.

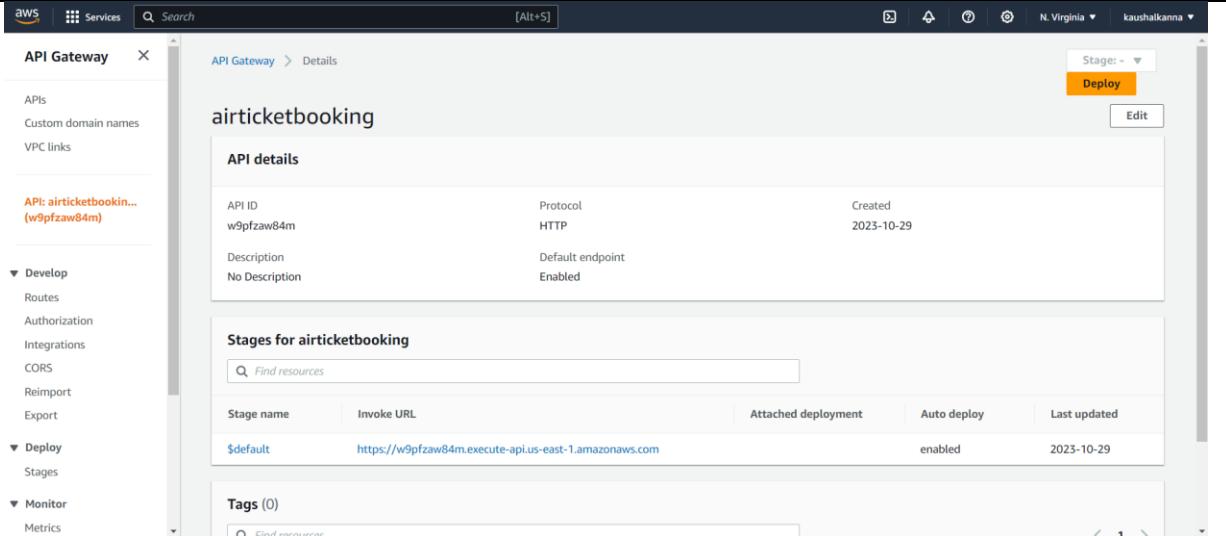
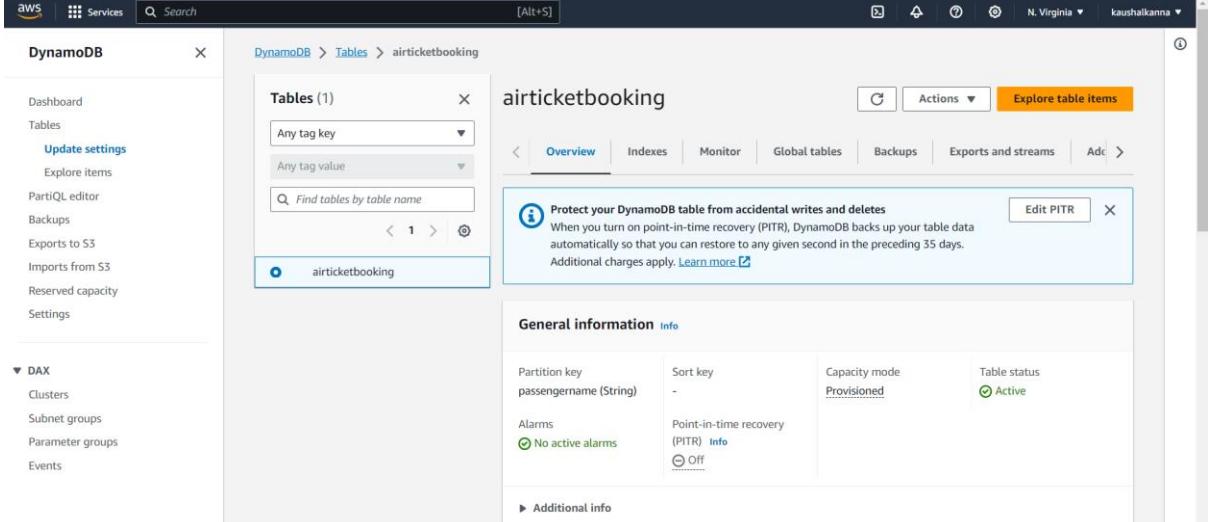
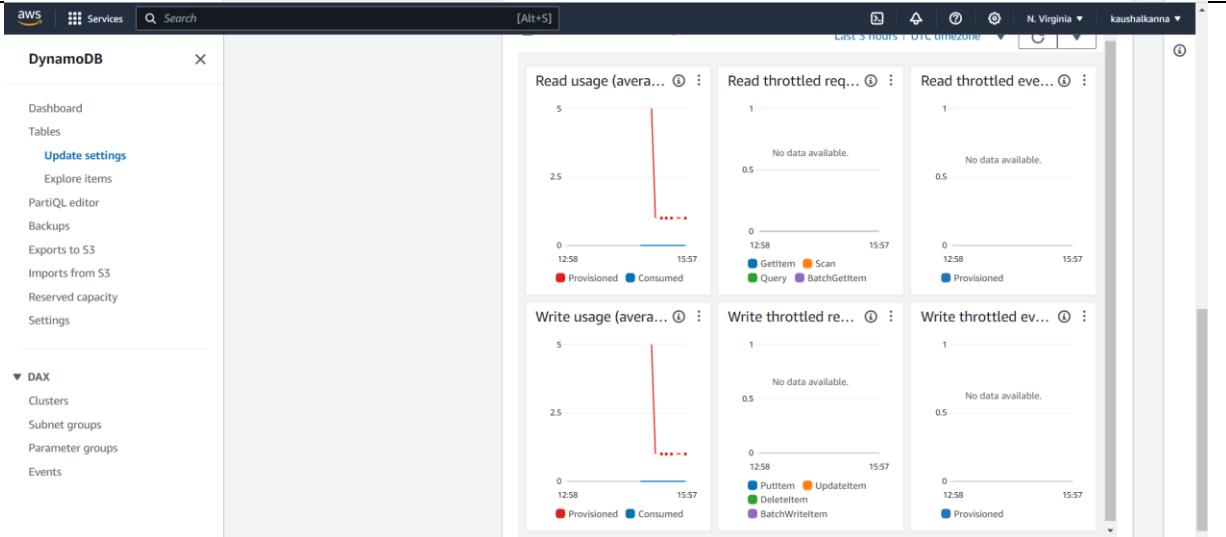


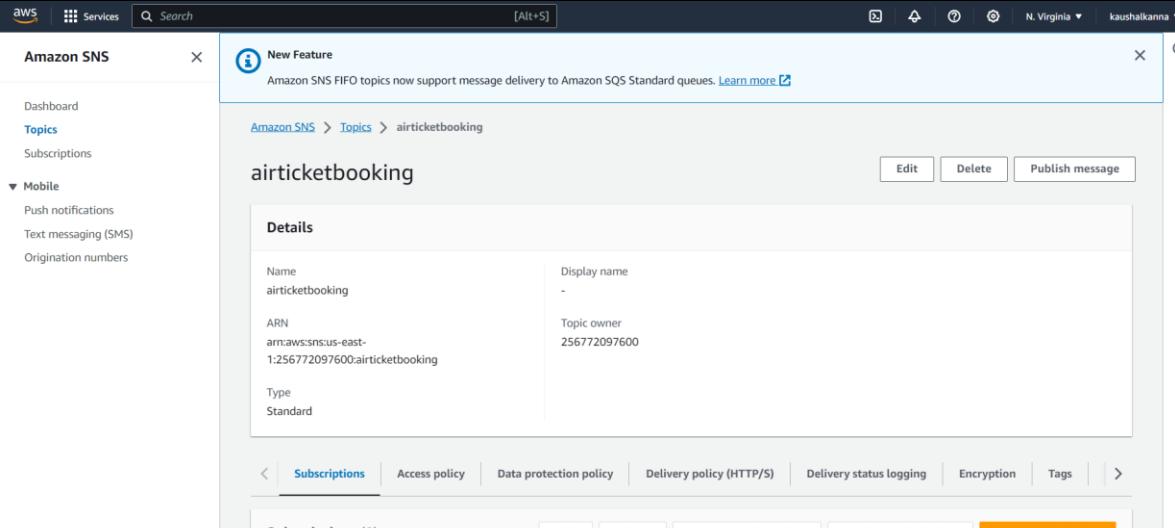
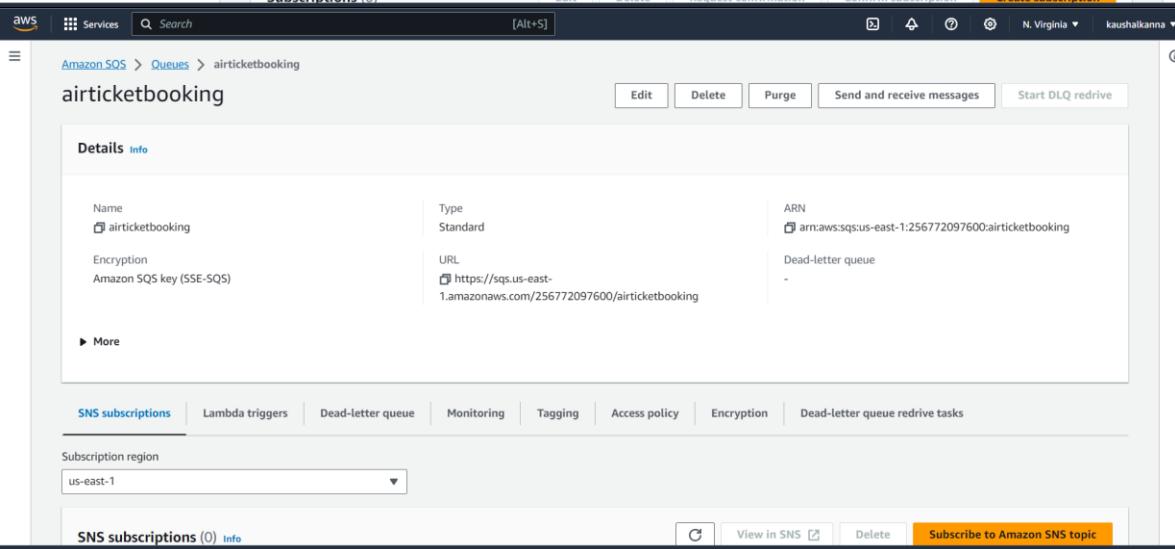
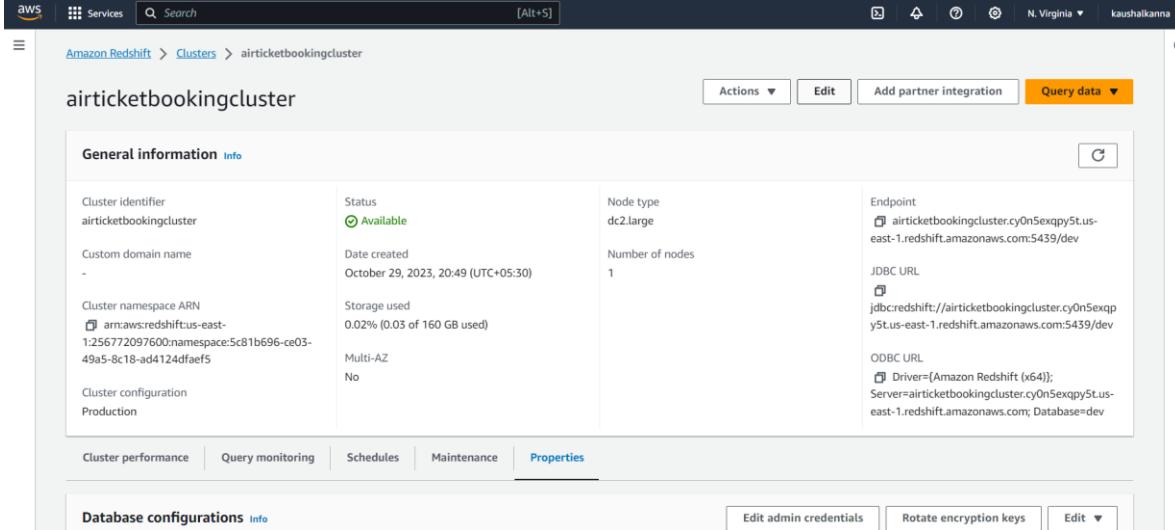
\*Define IAM roles and permissions for various components



\*The following iam permission policies to access these services



*Use API Gateway to create restful APIs for flight search and booking	
*Use DynamoDB If you require a NoSQL database for specific use cases	
*These are the following database usage by users	

*Use SNS(Simple Notification Service)to push notifications for users.	 The screenshot shows the Amazon SNS console. A banner at the top indicates a new feature: "Amazon SNS FIFO topics now support message delivery to Amazon SQS Standard queues." The main view displays a topic named "airticketbooking". The "Details" section shows the topic's name, ARN, and type. Below it are tabs for "Subscriptions", "Access policy", "Data protection policy", "Delivery policy (HTTP/S)", "Delivery status logging", "Encryption", and "Tags".
*Use SQS(Simple Query Service)If you need a message queuing service for asynchronous processing	 The screenshot shows the Amazon SQS console. It displays a queue named "airticketbooking". The "Details" section includes fields for Name (airticketbooking), Type (Standard), ARN, URL (https://sqs.us-east-1.amazonaws.com/256772097600/airticketbooking), and Dead-letter queue (-). Below the details are tabs for "SNS subscriptions", "Lambda triggers", "Dead-letter queue", "Monitoring", "Tagging", "Access policy", "Encryption", and "Dead-letter queue redrive tasks". A dropdown for "Subscription region" is set to "us-east-1".
*Use Redshift for data analytics and reporting.	 The screenshot shows the Amazon Redshift console. It displays a cluster named "airticketbookingcluster". The "General information" section provides details like Cluster identifier (airticketbookingcluster), Status (Available), Node type (dc2.large), Endpoint (airticketbookingcluster.cyOn5exqpy5.us-east-1.redshift.amazonaws.com:5439/dev), and JDBC URL (jdbc:redshift://airticketbookingcluster.cyOn5exqpy5.us-east-1.redshift.amazonaws.com:5439/dev). Other properties include Custom domain name (-), Date created (October 29, 2023, 20:49 (UTC+05:30)), Storage used (0.02% (0.03 of 160 GB used)), Number of nodes (1), and ODBC URL (Driver={Amazon Redshift (x64)}; Server=airticketbookingcluster.cyOn5exqpy5.us-east-1.redshift.amazonaws.com; Database=dev). Below the general information are tabs for "Cluster performance", "Query monitoring", "Schedules", "Maintenance", and "Properties".

\*Use Polly for text-to-speech functionality in your application.

The screenshot shows the Amazon Polly interface. On the left, there's a sidebar with links: Text-to-Speech, Lexicons, and S3 synthesis tasks. The main area is titled "S3 synthesis tasks (1) Info". It says "Synthesis tasks reflect all your speeches saved to S3 buckets. To create a new task, choose Text-to-Speech". Below this is a table with one row of data:

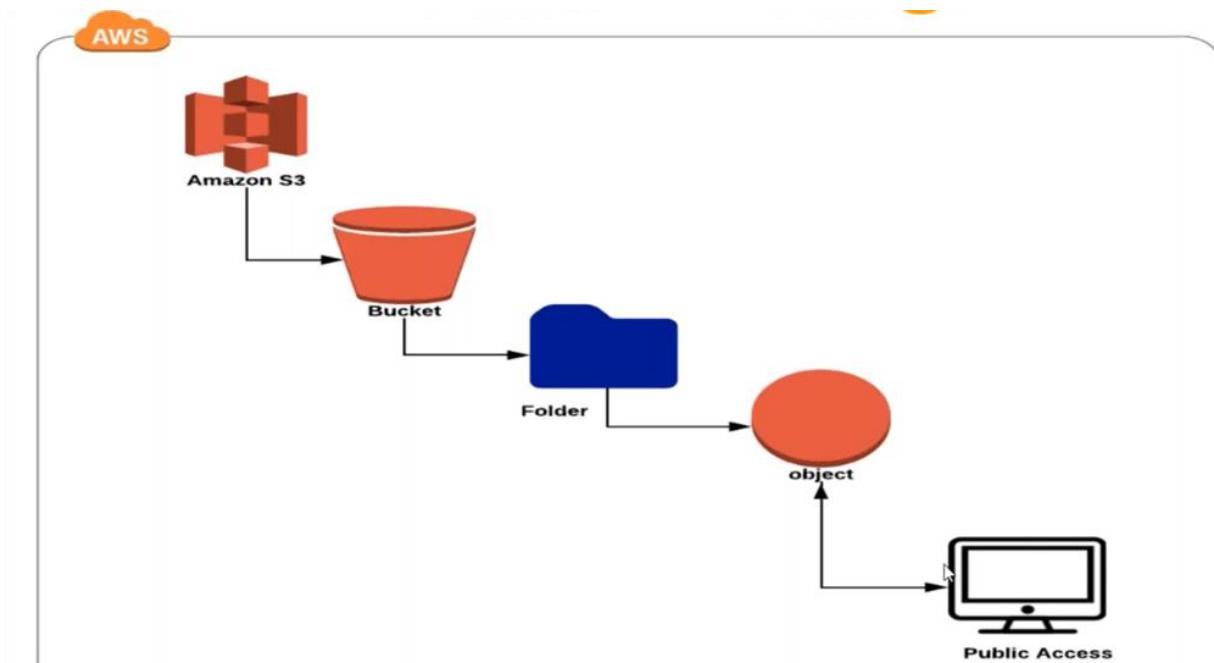
Task ID	Status	Voice ID	Engine	Number of characters	Creation time	S3 URL
<a href="#">ea8bbf9b-56c6-49d5-9d02-5cadff62b5387</a>	Completed	Matthew	Standard	51	15:24:21 UTC+05:30 10/29/2023	<a href="s3://airticketbooking/ea8bbf9b-56c6-49d5-9d02-5cadff62b5387.mp3">s3://airticketbooking/ea8bbf9b-56c6-49d5-9d02-5cadff62b5387.mp3</a>

## 6.Object file code that are uploaded in S3 bucket

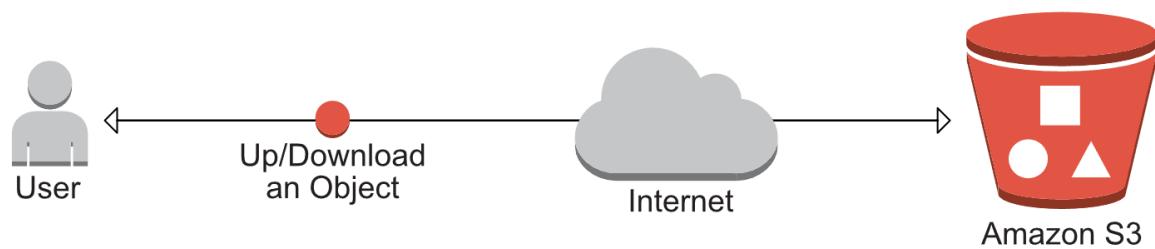
Agenda:

- S3
- Objects
- HTML scripts
- Style scripts

- S3: Simple Storage Service is a scalable and highly durable object storage service provided by Amazon Web Services (AWS). It is designed to store and retrieve any amount of data from anywhere on the web and is widely used for various purposes including data storage, backup, content distribution and hosting static websites.



- Objects: are the fundamental storage entity in S3. Objects are essentially files that can be stored in S3.

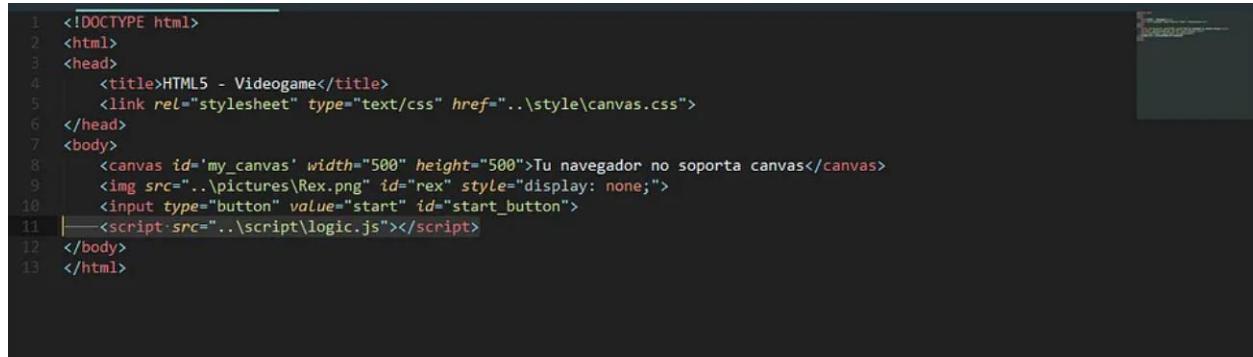


- HTML Scripts(Hypertext markup language): typically refers to the code written in the HTML language to define the structure and content of a web page. HTML is a markup language used to create the structure of a webpage by using tags to define elements such as headings, paragraphs, images, links, and more. HTML scripts consist of a series of HTML tags that are interpreted by web browsers to render the content on a webpage.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Code Example</title>
</head>
<body>
  <!-- Load external script with integrity check and crossorigin attribute -->
  <script src="https://HappyCoding.com/scripts/example.js"
    integrity="sha123-BEF6380502"
    crossorigin="anonymous"></script>

  <!-- Rest of HTML content goes here -->
  <h1>Welcome HAPPY CODING!</h1>
  <p>Check out this example showing how to use the script tag with the integrity attribute</p>
</body>
</html>
```

- Style scripts: typically refer to the code written in CSS (Cascading Style Sheets) to define the visual presentation and layout of HTML elements on a webpage. CSS is a stylesheet language used to describe the look and formatting of a document written in HTML or XML. CSS allows developers to control aspects such as colors, fonts, spacing, positioning, and responsiveness of HTML elements. Style scripts written in CSS are used to apply styles to specific HTML elements or groups of elements.



```

1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>HTML5 - Videogame</title>
5      <link rel="stylesheet" type="text/css" href="../style/canvas.css">
6  </head>
7  <body>
8      <canvas id='my_canvas' width="500" height="500">Tu navegador no soporta canvas</canvas>
9      
10     <input type="button" value="start" id="start_button">
11     <script src="../script\logic.js"></script>
12 </body>
13 </html>

```

- Html code which has been used to create static website in S3 bucket

```

<!DOCTYPE html>
<!--[if lt IE 7 ]><html class="ie ie6" lang="en"> <![endif]-->
<!--[if IE 7 ]><html class="ie ie7" lang="en"> <![endif]-->
<!--[if IE 8 ]><html class="ie ie8" lang="en"> <![endif]-->
<!--[if (gte IE 9)|(IE)]><!-->
<html lang="en">
<!--<![endif]-->

<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1">
    <meta name="description" content="">
    <meta name="author" content="">
    <!--[if IE]>
        <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
    <![endif]-->

```

```

<title>My Static Website</title>
<!--REQUIRED STYLE SHEETS-->
<!-- BOOTSTRAP CORE STYLE CSS -->
<link href="https://stackpath.bootstrapcdn.com/bootstrap/3.1.1/css/bootstrap.min.css"
rel="stylesheet" />
<!-- FONTAWESOME STYLE CSS -->
<link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.5.0/css/all.css"
crossorigin="anonymous">
<!-- CUSTOM STYLE CSS -->
<link href="style.css" rel="stylesheet" />
<!-- GOOGLE FONT -->
<link href="https://fonts.googleapis.com/css?family=Open+Sans" rel='stylesheet'
type='text/css'>
<!-- HTML5 shim and Respond.js IE8 support of HTML5 elements and media queries -->
<!--[if lt IE 9]>
<script src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></script>
<script src="https://oss.maxcdn.com/libs/respond.js/1.3.0/respond.min.js"></script>
<![endif]-->
</head>

<body>
<!--Header section -->
<div class="container" id="home">
<div class="row text-center">
<div class="col-md-12">
<h1 class="head-main" > <i class="fab fa-aws fa-2x"></i> Static Website</h1>
<h2 class="head-sub-main" > Created by Kaushal </h2>
</div>
</div>
</div>
<!--End Header section -->

<!-- Navigation -->
<nav class="navbar-inverse" role="navigation">
<div class="container">
<div class="navbar-header">
<button type="button" class="navbar-toggle" data-toggle="collapse" data-
target=".navbar-ex1-collapse">
<span class="sr-only">Toggle navigation</span>
<span class="icon-bar"></span>
<span class="icon-bar"></span>
<span class="icon-bar"></span>
</button>
<a class="navbar-brand" href="http://aws.amazon.com/partners/">APN Network </a>
</div>
<!-- Collect the nav links for toggling -->
<div class="collapse navbar-collapse navbar-ex1-collapse">
<ul class="nav navbar-nav">
<li><a href="#"><i class="fas fa-home"></i></a>
</li>

```

```

<li><a href="#about">About</a>
</li>
<li><a href="#work-sec">Transformation</a>
</li>
<li><a href="#contact-sec">Contact</a>
</li>
<li><a href="#map-sec">Location</a>
</li>
</ul>
</div>
<!-- /.navbar-collapse -->
</div>
<!-- /.container -->
</nav>
<!--End Navigation -->

<!--About Section-->
<section class="color-white " id="about">
<div class="container">
<div class="row text-center">
<div class="col-md-8 col-md-offset-2 ">
<h2 style="padding-top:50px;">About Us</h2>
<h4>
<strong>
    Lorem ipsum dolor sit amet, consectetur adipiscing elit.
    Curabitur nec nisl odio. Mauris vehicula at nunc id posuere.
</strong>
</h4>
</div>
</div>
</div>
</div>
</section>

<section class=" color-light">
<div class="container">
<div class="row text-center">
<div class="col-md-12">
<h2> Our exclusive services</h2>
<p>
    Lorem ipsum dolor sit amet, consectetur adipiscing elit.
    Curabitur nec nisl odio. Mauris vehicula at nunc id posuere.
    Lorem ipsum dolor sit amet, consectetur adipiscing elit.
    Curabitur nec nisl odio. Mauris vehicula at nunc id posuere.
    Lorem ipsum dolor sit amet, consectetur adipiscing elit.
    Curabitur nec nisl odio. Mauris vehicula at nunc id posuere.
    Lorem ipsum dolor sit amet, consectetur adipiscing elit.
    Curabitur nec nisl odio. Mauris vehicula at nunc id posuere.
    Lorem ipsum dolor sit amet, consectetur adipiscing elit.
    Curabitur nec nisl odio. Mauris vehicula at nunc id posuere.
    Lorem ipsum dolor sit amet, consectetur adipiscing elit.
    Curabitur nec nisl odio. Mauris vehicula at nunc id posuere.
    
```

```

Curabitur nec nisl odio. Mauris vehicula at nunc id posuere.
Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Curabitur nec nisl odio. Mauris vehicula at nunc id posuere.
Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Curabitur nec nisl odio. Mauris vehicula at nunc id posuere.

</p>
</div>
</div>
</div>
</div>
</section>
<!--End About Section-->

<!-- Transformation Section -->
<section class="color-white " id="work-sec">
<div class="container">
<div class="row text-center">
<div class="col-md-8 col-md-offset-2 ">
<h2>Transformation</h2>
<h4>
<strong>
    Lorem ipsum dolor sit amet, consectetur adipiscing elit.
    Curabitur nec nisl odio. Mauris vehicula at nunc id posuere.
</strong>
</h4>
</div>
</div>
</div>

<div class="row text-center g-pad-bottom">
<div class="col-md-4">
<div class="work-div">
    <i class="fas fa-box-open fa-5x"></i>
    <h3>Package </h3>
</div>
</div>

<div class="col-md-4 ">
<div class="work-div">
    <i class="fas fa-arrows-alt-h fa-5x"></i>
</div>
</div>

<div class="col-md-4 ">
<div class="work-div">
    <i class="fas fa-cloud fa-5x"></i>
    <h3>The Cloud</h3>
</div>
</div>

</div>
</div>

```

```

</section>
<!--End Transformation Section -->
<!-- Contact Section -->
<section class="color-light " id="contact-sec">
  <div class="container">
    <div class="row text-center">
      <div class="col-md-8 col-md-offset-2 ">
        <h2>Locate Us here</h2>
        <h4>
          <strong>
            Lorem ipsum dolor sit amet, consectetur adipiscing elit.
            Curabitur nec nisl odio. Mauris vehicula at nunc id posuere.
          </strong>
        </h4>
      </div>
    </div>
  </div>

  <div class="row">
    <div class="col-md-5 contact-cls">
      <h3>Our Address</h3>
      <div>
        <span><i class="fa fa-home"> </i> Address: 123/56, Your City, USA </span>
        <br />
        <span><i class="fa fa-phone"> </i> Phone: 82-230-555-8899</span>
      </div>
    </div>
    <div class="col-md-7">
      <br />
      <div id="social-icon">
        <a href="#"><i class="fab fa-facebook fa-2x"></i></a>
        <a href="#"><i class="fab fa-twitter fa-2x"></i></a>
        <a href="#"><i class="fab fa-linkedin fa-2x"></i></a>
        <a href="#"><i class="fab fa-google-plus fa-2x"></i></a>
        <a href="#"><i class="fab fa-pinterest fa-2x"></i></a>
      </div>
    </div>
  </div>
</section>
<!--End Contact Section -->

<!--Map Section -->
<section class="color-light " id="map-sec">
  <iframe class="cnt"
src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d2999841.293321206!2d75.8092040499999!3d42.7559420499997!2m3!1f0!2f0!3f0!3m2!1i1024!2i768!4f13.1!3m3!1m2!1s0x4ccc4bf0f123a5a9%3A0xddcf6c1de189567!2sNew+York!5e0!3m2!1sen!2s!4v139531308825"></iframe>
</section>

```

```

<!--End Map Section -->
<!--footer Section -->
<div class="for-full-back " id="footer">
    CSS Template from <a href="https://www.free-css.com/free-css-templates/page203/image-less">Free CSS</a>
</div>
<!--End footer Section -->
<!-- JAVASCRIPT FILES PLACED AT THE BOTTOM TO REDUCE THE LOADING TIME -->
<!-- CORE JQUERY -->
<script src="https://code.jquery.com/jquery-1.10.2.min.js"></script>
<!-- BOOTSTRAP CORE SCRIPT -->
<script src="https://stackpath.bootstrapcdn.com/bootstrap/3.1.1/js/bootstrap.min.js"></script>
<!-- CUSTOM SCRIPTS -->
<script src="script.js"></script>

<script defer
src="https://static.cloudflareinsights.com/beacon.min.js/v8b253dfea2ab4077af8c6f58422dfbfd1689876627854" integrity="sha512-bjgnUKX4azu3dLTie9u6TKqgx29RBwfj3QXYt5EKfWM/9hPSAI/4qcV5NACjwAo8UtTeWefx6Zq5PHcMm7Tg==" data-cf-beacon='{"rayId":"814ef2a9294440bc","token":"a73834a4a1444e9ab89e8da06da41720","version":"2023.8.0","si":100}' crossorigin="anonymous"></script>
</body>

</html>

```

- This is the style script which has been used to create static website

```

/*=====
GENERAL STYLES
=====*/
body {
    font-family:'Open Sans', sans-serif;
    font-size:14px;
    background-color:#9B60DB;
}

.nav a {

```

```
color:#ffffff !important;  
}  
  
.navbar-header a {  
color:#ffffff !important;  
padding-right:100px;  
}  
  
.text-center {  
text-align:center;  
}  
  
h1, h2, h3, h4, h5, h6 {  
font-family:'Open Sans', sans-serif;  
}  
  
h1 {  
line-height:80px;  
font-weight:900;  
font-size:60px;  
padding:30px 20px 10px 20px;  
}  
  
h2 {  
line-height:40px;  
font-weight:900;  
font-size:30px;  
padding:20px 20px 10px 20px;
```

```
text-transform:uppercase;  
}  
  
h3 {  
    line-height:30px;  
    padding-bottom:20px;  
}  
  
h4 {  
    line-height:40px;  
    padding-bottom:15px;  
}  
  
p {  
    font-weight:300;  
    line-height:30px;  
    padding-bottom:20px;  
}  
  
.space-free {  
    height:100px;  
}  
  
section {  
    padding-top:50px;  
    padding-bottom:30px;  
}
```

```
/*=====
 COLOR CODES
=====*/
.color-white {
    background-color:#ffff!important;
}

.color-dark {
    background-color:#696969!important;
}

.color-light {
    background-color:#E9E9E9!important;
}

/*=====
 HEADER STYLES
=====*/
.head-main {
    font-size:80px !important;
    font-weight:900!important;
    color:#ffff!important;
    padding:150px 20px 10px 20px;
}

.head-sub-main {
    font-size:40px !important;
```

```
font-weight:600!important;  
color:#fff!important;  
padding:5px 20px 150px 20px;  
}  
  
/*=====  
 NAVIGATION STYLES  
=====*/  
  
nav {  
    position: absolute;  
    width: 100%;  
    background: #fff;  
    z-index:99;  
}  
.fixed {  
    position: fixed;  
    top: 0;  
    min-height: 50px;  
    z-index: 99;  
}  
  
.navbar-inverse {  
background-color: #000000;
```

```
border-color: #000000;  
min-height: 80px;  
padding-top: 10px;  
font-size: 18px;  
}  
  
.navbar-brand {  
  
font-size: 30px;  
}  
  
/*=====
```

#### WORK FLOW STYLES

```
=====*/  
  
.work-div {  
  
width: 100%;  
border: 0;  
padding: 20px;  
margin-bottom: 30px;  
}  
  
/*=====
```

#### CONTACT STYLES

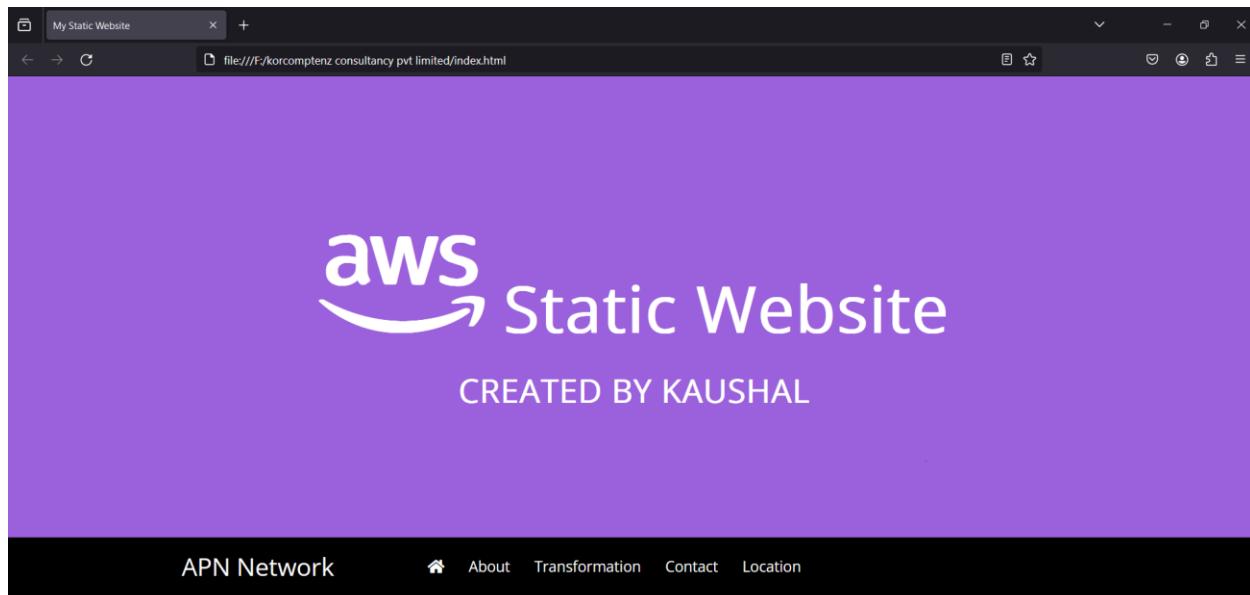
```
=====*/  
  
#social-icon a {
```

```
color:#616161!important;  
margin-right:10px;  
}  
  
.contact-cls {  
font-size:20px;  
line-height:40px;  
}  
  
.cnt {  
width: 100%;  
min-height: 350px;  
border: 1px solid #e1e1e1;  
}  
  
/*=====
```

#### FOOTER STYLES

```
=====*/  
  
#footer {  
background-color:#000;  
color:#fff;  
padding:20px 50px 20px 50px;  
text-align:right;  
}
```

- This is the following static website created from above html and style scripts in S3 bucket for static website hosting



- Html code to create airticketbooking website in S3 bucket

```
<!DOCTYPE html>
<html>
<head>
    <title>Flight Booking Form</title>
</head>
<body>
    <h1>Flight Booking</h1>
    <form action="book_flight.php" method="post">
        <label for="from">From:</label>
        <input type="text" id="from" name="from" required><br><br>

        <label for="to">To:</label>
        <input type="text" id="to" name="to" required><br><br>
```

```

<label for="departure">Departure Date:</label>
<input type="date" id="departure" name="departure" required><br><br>

<label for="return">Return Date:</label>
<input type="date" id="return" name="return"><br><br>

<label for="passengers">Passengers:</label>
<select id="passengers" name="passengers">
    <option value="1">1</option>
    <option value="2">2</option>
    <option value="3">3</option>
    <option value="4">4</option>
    <option value="5">5</option>
</select><br><br>

    <input type="submit" value="Book Now">
</form>
</body>
</html>

```

- This is the style script used for airticketbooking website in S3 bucket

```
/* Reset some default styles for consistency */
```

```

body, html {
    margin: 0;
    padding: 0;
    font-family: Arial, sans-serif;
}

```

```
/* Style the container for the booking system */
```

```
.
.booking-container {
    width: 80%;
    margin: 0 auto;
    padding: 20px;
    background-color: #f0f0f0;
    border: 1px solid #ccc;
    border-radius: 5px;
}
```

```
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);  
}  
  
/* Style the header */
```

```
.header {  
    text-align: center;  
    font-size: 24px;  
    margin-bottom: 20px;  
}
```

```
/* Style flight listings */
```

```
.flight-list {  
    list-style: none;  
    padding: 0;  
}
```

```
.flight-item {  
    border: 1px solid #ddd;  
    border-radius: 5px;  
    padding: 10px;  
    margin: 10px 0;  
    background-color: #fff;  
}
```

```
/* Style buttons */
```

```
.button {  
    background-color: #007BFF;  
    color: #fff;
```

```
border: none;  
padding: 10px 20px;  
border-radius: 5px;  
cursor: pointer;  
}  
  
.button:hover {  
background-color: #0056b3;  
}  
  
/* Style the footer */  
.footer {  
text-align: center;  
margin-top: 20px;  
color: #777;  
}  
  
/* Media query for responsiveness (adjust breakpoints as needed) */  
@media (max-width: 768px) {  
.booking-container {  
width: 100%;  
}  
}
```

- This is the airticket booking website that been created from html and style scripts in S3 bucket.

**Flight Booking**

From:

To:

Departure Date:  dd / mm / yyyy

Return Date:  dd / mm / yyyy

Passengers:

- This is the Hello World linux user datascript to display hello world website

```
#install httpd(Linux 2 Version)

yum update -y

yum install -y httpd

systemctl start httpd

systemctl enable httpd

echo "<h1> Hello World from $(hostname -f)</h1>">/var/www/html/index.html
```



**Certified**

APR 2023 - APR 2024

INDIA

TM



**Korcomptenz**

Korcomptenz is a technology transformation provider that partners with clients to improve their digital experience and insights. We unlock the power of technology in the areas of ERP, CRM, infrastructure management, and cloud to empower our clients with intelligent and experiential solutions. We #FocusOnYou

Discover how we #FocusOnYou at [www.korcomptenz.com](http://www.korcomptenz.com)

**Get in touch with us:**

+97142285285 | [sales@korcomptenz.com](mailto:sales@korcomptenz.com)

