

Phase 1

Contents

Team Information	2
1.IAM.....	2
2C3.Compute (EC2) C Storage (EBS) Layer	5
4. Simple Storage Service (S3)	12
5.Virtual Private Cloud (VPC)	23
1. Purpose of the VPC	23
2. VPC Configuration Requirements.....	23
2.1 Create VPC	23
2.2 Subnet Architecture.....	23
2.3 Internet Gateway Configuration	24
2.4 NAT Gateway Configuration.....	24
2.5 Route Tables Setup	24
2.6 Network ACLs.....	24
2.7 Security Groups	24
3. Deliverables.....	25
3.1 VPC Architecture Diagram.....	25
3.2 Subnet Allocation Table	25
3.3 Security Group Definitions	26
3.4 Network ACL Rules	26
4. Step-by-Step Implementation	26
6. Results	28
6. Lambda Functions.....	29
7. Elastic Load Balancer (ELB)	35
8. Relational Database Service (RDS)	46

Team Information

id	name	email
22101164	Merna Adel Abdelrahman	merna.ibrahim.2023@aiu.edu.eg
22101332	Remonda Rezq Azer	remonda.rezq.2023@Aiu.edu.eg
22101137	Abdallah Ahmed Mahmoud Lasheen	abdalla.lasheen.2023@Aiu.edu.eg
22101053	Antonius sameh	antonius.mosad.2023@Aiu.edu.eg
22100933	Noura adel abd el Rahman Mohamed	nora.mohamed.2023@Aiu.edu.eg
22100665	Nourhan Abdelhamid Ahmed Mohammed	nourhan.mohammed.2023@Aiu.edu.eg

1. IAM

1 Identity and Access Management (IAM)

1. Purpose s Strategy

The primary goal of this module is to secure access to the Cloud-Based Learning Platform resources. Our design philosophy follows the **Principle of Least Privilege**, ensuring strict isolation between the microservices as required in the project specifications.

Key Design Decisions:

- **Service Isolation:** Instead of a generic EC2 role, we created distinct IAM roles for each microservice (e.g., Quiz-Service-Role, TTS-Service-Role). This ensures that if the TTS service is compromised, the attacker cannot access the Quiz or Chat data.
- **Container Security:** We implemented IAM Roles for Service Accounts (IRSA) to map permissions directly to the container tasks rather than the underlying EC2 instances.
- **Admin Security:** We enforced strict Multi-Factor Authentication (MFA) for all users with administrative privileges.

2. Implementation Steps

We executed the following steps to meet the requirements listed in Section 2.1:

1. User Group Creation:

- a. Platform-Admins: Granted AdministratorAccess with a strictly attached Force_MFA policy.

- b. Backend-Devs: Granted access to development environments only, with restrictions on production S3 buckets.

2. Role Configuration:

- a. **Quiz-Generator-Role:** Configured for the Quiz Service to access quiz-service-storage-dev.
 - b. **Document-Reader-Role:** Configured for the Document Reader Service to access document-reader-storage-dev.
 - c. **Lambda-Cleanup-Role:** Created for the serverless maintenance functions.
3. **Policy Definition:** We authored custom JSON policies to restrict access to specific S3 buckets and RDS instances per service.

3. Configurations (Policy JSONs)

A. Quiz Service Policy

Rationale: This policy grants the Quiz Service permission to store generated quizzes and read processed document notes, as per the isolation requirements.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "QuizBucketAccess",
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:GetObject"
      ],
      "Resource": "arn:aws:s3:::quiz-service-storage-dev/* // Access limited strictly to the Quiz bucket"
    },
    {
      "Sid": "ReadProcessedNotes",
      "Effect": "Allow",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::document-reader-storage-dev/notes/*"
    },
    {
      "Sid": "ConnectToQuizDB",
      "Effect": "Allow",
      "Action": "rds-db:connect",
      "Resource": "arn:aws:rds-db:us-east-1:123456789:dbuser:quiz_db/app_user"
    }
  ]
}
```

B. Admin MFA Enforcement Policy

Rationale: This policy explicitly denies all actions for Admin users unless they are authenticated via MFA.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyAllExceptMFA",
      "Effect": "Deny",
      "NotAction": [
        "iam>CreateVirtualMFADevice",
        "iam:EnableMFADevice",
        "iam:ResyncMFADevice",
        "iam>ListMFADevices"
      ]
    }
  ]
}
```

```

],
"Resource": "*",
"Condition": {
    "BoolIfExists": {
        "aws:MultiFactorAuthPresent": "false"
    }
}
}
]
}
```

4. Access Control Matrix

The following table illustrates the permission boundaries established for the project resources:

Entity (Role/User)	Target Resource	Access Level	Rationale
Admin User	All Resources	Full Access	System Administration (MFA Required)
TTS-Service-Role	tts-service-storage-dev	Read/Write	Audio file generation C storage ¹²
STT-Service-Role	stt-service-storage-dev	Read/Write	Uploading audio for transcription ¹³
Chat-Service-Role	chat-service-storage-dev	Read/Write	Storing conversation history ¹⁴
Quiz-Service-Role	quiz-service-storage-dev	Read/Write	storing quiz templates C results ¹⁵
Lambda-Cleanup	All Buckets (Lifecycle)	Delete Only	Automated file archiving/cleanup ¹⁶

5. Deliverables Included

In accordance with the project requirements, the following files are attached in the IAM_Deliverables/ folder :

1. iam_role_definitions.pdf: Detailed documentation of all created roles.

2. policies/: Folder containing the raw .json policy files.
3. access_control_matrix.xlsx: Comprehensive permission matrix.
4. infrastructure/iam.tf: Terraform scripts used for provisioning the IAM layer.
5. screenshots/: Evidence of User creation, Group assignment, and MFA activation status.

2s3. Compute (EC2) & Storage (EBS) Layer

1. Overview

In this phase, I provisioned the compute and storage resources required to host the microservices, Kafka cluster, and Zookeeper ensemble. The implementation focuses on High Availability (HA) by distributing resources across two Availability Zones (us-east-1a, us-east-1b) and ensuring Data Security through full encryption.

2.1 SSH Key Pair Configuration

To facilitate secure remote access (SSH) to the EC2 instances for management and troubleshooting purposes, a dedicated RSA key pair was generated prior to launching the infrastructure.

Configuration Details:

- Name: **Project-Key**
- Type: **RSA (2048-bit)** – Chosen for standard security compatibility.
- Format: **.pem (Privacy Enhanced Mail)** – Selected for compatibility with OpenSSH clients and Linux/Mac environments.

This key is securely stored and is associated with all Launch Templates (Container-Host, Kafka-Broker, and Zookeeper-Node) to allow administrative access if required.

Key pair

A key pair, consisting of a private key and a public key, is a set of security credentials that you use to prove your identity when connecting to an instance.

Name

Project-Key

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type | [Info](#)

RSA

ED25519

Private key file format

.perm

For use with OpenSSH

.ppk

For use with PuTTY

Tags - *optional*

No tags associated with the resource.

[Add new tag](#)

You can add up to 50 more tags.

2.2 Launch Templates Configuration

To standardize the deployment of the microservices and message brokers, I created three distinct Launch Templates. These templates pre-define the Amazon Machine Image (Ubuntu Server 24.04 LTS), Instance Type (t3.medium), and network security settings, ensuring consistency across the environment.

Created Templates:

- Container-Host-Template: For the Docker/Kubernetes cluster.
- Kafka-Broker-Template: For the Kafka streaming cluster.
- Zookeeper-Node-Template: For the coordination ensemble.

The screenshot shows the AWS Cloud Computing interface for the CSE353 course. On the left, a sidebar menu for EC2 includes options like Dashboard, EC2 Global View, Events, Instances (Instances, Instance Types), Launch Templates (selected), Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, and Capacity Manager. Below these are Images (AMIs, AMI Catalog), Elastic Block Store (Volumes, Snapshots), and a 'New' button.

The main content area displays 'Launch Templates (1/3)' with a search bar and a table. The table has columns: Launch Template ID, Launch Template Name, Default Version, Latest Version, Create Time, and Created By. It lists three templates: 'Container-Host-Template' (selected), 'Zookeeper-Node-Template', and 'Kafka-Broker-Template'. The 'Container-Host-Template' row shows details: ID lt-04c45b15d9ee776e6, Name Container-Host-Template, Default Version 1, Latest Version 2, Create Time 2025-11-22T03:58:58.000Z, and Created By arn:aws:sts::755097.

A detailed view of the 'Container-Host-Template' is shown below. It includes sections for 'Launch template details' (Launch template ID, Launch template name, Default version), 'Owner' (arn:aws:sts::755097618049:assumed-role/vclabs/user4443494=merna.ibrahim.2023@Aiu.edu.eg), and tabs for 'Details' (selected), 'Versions', and 'Template tags'.

Create Container-Host-Template

The screenshot shows the 'Modify template (Create new version)' wizard. The top navigation bar includes EC2 > Launch templates > Modify template (Create new version).

Launch template name and version description

- Launch template name:** Container-Host-Template (lt-04c45b15d9ee776e6)
- Template version description:** Container-Host-Template (Max 255 chars)
- Auto Scaling guidance:** Select this if you intend to use this template with EC2 Auto Scaling. A checkbox is checked: 'Provide guidance to help me set up a template that I can use with EC2 Auto Scaling'.
- Source template:** (button)

Launch template contents

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

Application and OS Images (Amazon Machine Image) - required

An AMI contains the operating system, application server, and applications for your instance. If you don't see a suitable AMI below, use the search field or choose **Browse more AMIs**.

Summary

- Software Image (AMI):** Canonical, Ubuntu, 24.04, amd64... [read more](#) ami-0ecb62995f68bb549
- Virtual server type (instance type):** t3.medium
- Firewall (security group):** SG-containers
- Storage (volumes):** 1 volume(s) - 50 GiB

Buttons: Cancel, Create template version

Create Kafka-Broker-Template

EC2 > Launch templates > Modify template (Create new version)

Launch template name and version description

Launch template name
Kafka-Broker-Template (lt-03cd647d7fc0d24f2)

Template version description
Kafka-Broker-Template

Max 255 chars

Auto Scaling guidance | Info
Select this if you intend to use this template with EC2 Auto Scaling
 Provide guidance to help me set up a template that I can use with EC2 Auto Scaling

▶ Source template

Summary

Software Image (AMI)
Canonical, Ubuntu, 24.04, amd6... [read more](#)
ami-0ecb62995f6bb0549

Virtual server type (instance type)
t3.medium

Firewall (security group)
SG-Kafka

Storage (volumes)
1 volume(s) - 100 GiB

[Cancel](#) [Create template version](#)

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

▼ Application and OS Images (Amazon Machine Image) | Info
An AMI contains the operating system, application server, and applications for your instance. If you don't see a suitable AMI below, use the search field or choose [Browse more AMIs](#).

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

Create Zookeeper-Node-Template

EC2 > Launch templates > Modify template (Create new version)

Launch template name and version description

Launch template name
Zookeeper-Node-Template (lt-06bb71c93236bdc09)

Template version description
Zookeeper-Node-Template

Max 255 chars

Auto Scaling guidance | Info
Select this if you intend to use this template with EC2 Auto Scaling
 Provide guidance to help me set up a template that I can use with EC2 Auto Scaling

▶ Source template

Summary

Software Image (AMI)
Canonical, Ubuntu, 24.04, amd6... [read more](#)
ami-0ecb62995f6bb0549

Virtual server type (instance type)
t3.medium

Firewall (security group)
SG-Kafka

Storage (volumes)
1 volume(s) - 20 GiB

[Cancel](#) [Create template version](#)

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

▼ Application and OS Images (Amazon Machine Image) | Info
An AMI contains the operating system, application server, and applications for your instance. If you don't see a suitable AMI below, use the search field or choose [Browse more AMIs](#).

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

2.3 Auto Scaling Group (Container Cluster)

- For the application layer, I implemented an Auto Scaling Group (Container-Cluster-ASG) to ensure scalability and fault tolerance. The group is configured to launch instances into Private Subnets across two Availability Zones (us-east-1a, us-east-1b).

Configuration Details:

- Desired Capacity: 3 instances.
- Minimum Capacity: 3 instances.
- Maximum Capacity: 5 instances.

Health Checks: EC2-based health checks are enabled.

Last updated less than a minute ago ⟳

[Launch configurations](#) [Launch templates](#) [Actions](#) [Create Auto Scaling group](#)

Search your Auto Scaling groups

Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max
Container-Cluster-ASG	Container-Host-Template Version Latest	3	-	3	3	5

Create Container-Cluster-ASG

Edit Container-Cluster-ASG [Info](#)

Group size [Info](#)

Specify the size of the Auto Scaling group by changing the desired capacity. You can also specify minimum and maximum scaling limits.

Desired capacity type

Choose the unit of measurement for the desired capacity value. vCPUs and Memory(GiB) are only supported for mixed instances groups configured with a set of instance attributes.

Units (number of instances)

Desired capacity

Specify your group size.

3

Scaling limits

Set limits on how much your desired capacity can be increased or decreased.

Min desired capacity

3

Max desired capacity

5

Equal or less than desired capacity

Equal or greater than desired capacity

Instance type

t3.medium

Network

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

Availability Zones and subnets

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

Select Availability Zones and subnets ⟳

- us-east-1a | subnet-081bbc6179e84d6f7 (private-subnet-1a) X
10.0.10.0/24
- us-east-1b | subnet-0d680853d07e59d9c (private-subnet-1b) X
10.0.11.0/24

Create a subnet ✚

Availability Zone distribution - new

Auto Scaling automatically balances instances across Availability Zones. If launch failures occur in a zone, select a strategy.

Balanced best effort

If launches fail in one Availability Zone, Auto Scaling will attempt to launch in another healthy Availability Zone.

Balanced only

If launches fail in one Availability Zone, Auto Scaling will continue to attempt to launch in the unhealthy Availability Zone to preserve balanced distribution.

2.4 Instances s High Availability (Multi-AZ)

- A total of 9 EC2 instances were successfully deployed. To achieve High Availability (HA) as per the architecture requirements, the Kafka and Zookeeper brokers were manually distributed across distinct Availability Zones.
- Distribution:
- Containers: Automatically balanced by ASG.
- Kafka/Zookeeper: 1 node in us-east-1a and 2 nodes in us-east-1b.

Instances (9) Info								
	Name Filter	Instance ID	Instance state Filter	Instance type Filter	Status check Filter	Alarm status Filter	Availability Zone Filter	Public IP Filter
Find Instance by attribute or tag (case-sensitive)								
<input type="checkbox"/>	i-08997c2d62593fee5	Running View details Edit	t3.medium	OK 3/3 checks passed	OK View alarms +	us-east-1b	-	
<input type="checkbox"/>	i-0782600155e3ba3e6	Running View details Edit	t3.medium	OK 3/3 checks passed	OK View alarms +	us-east-1b	-	
<input type="checkbox"/>	i-012a8845320b9fc98	Running View details Edit	t3.medium	OK 3/3 checks passed	OK View alarms +	us-east-1b	-	
<input type="checkbox"/>	i-074e2007518d491a0	Running View details Edit	t3.medium	OK 3/3 checks passed	OK View alarms +	us-east-1b	-	
<input type="checkbox"/>	i-077a2ab4d099da7d0	Running View details Edit	t3.medium	OK 3/3 checks passed	OK View alarms +	us-east-1a	-	
<input type="checkbox"/>	i-0000940ed935a98ef	Running View details Edit	t3.medium	OK 3/3 checks passed	OK View alarms +	us-east-1a	-	
<input type="checkbox"/>	i-018c1f37e97e6fa7	Running View details Edit	t3.medium	OK 3/3 checks passed	OK View alarms +	us-east-1a	-	
<input type="checkbox"/>	i-0ddca5bdcecd82b4	Running View details Edit	t3.medium	OK 3/3 checks passed	OK View alarms +	us-east-1a	-	
<input type="checkbox"/>	i-065615fd55ad4091c	Running View details Edit	t3.medium	OK 3/3 checks passed	OK View alarms +	us-east-1a	-	

EC2 > Launch templates > Launch instance from template

Launch instance from template

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

Choose a launch template

Source template

Zookeeper-Node-Template
ID: It-06bb71c93236bdc09

1 (Default)

Instance details

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

Application and OS Images (Amazon Machine Image)

An AMI contains the operating system, application server, and applications for your instance. If you don't see a suitable AMI below, use the search field or choose [Browse more AMIs](#).

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

Summary

Number of instances [Info](#)

1

Software Image (AMI)
Canonical, Ubuntu, 24.04, amd64... [read more](#)
ami-0ccb62995f68bb549

Virtual server type (instance type)
t3.medium

Firewall (security group)
SG-Kafka

Storage (volumes)
1 volume(s) - 20 GiB

[Cancel](#) [Launch instance](#) [Preview code](#)

5. EBS Storage s Encryption

Persistent storage was provisioned using GP3 EBS volumes attached to all instances. To comply with strict

security requirements, Server-Side Encryption (AWS KMS) was enabled for all volumes holding sensitive

data.

Volume Specifications:

- Kafka Brokers: 100 GiB per instance.
- Container Hosts: 50 GiB per instance.
- Zookeeper Nodes: 20 GiB per instance.

Service Tier	Size	Type	Encryption	Mount Point
Kafka Brokers	100 GiB	gp3	Encrypted	/dev/sda1
Container Hosts	50 GiB	gp3	Encrypted	/dev/sda1
Zookeeper Nodes	20 GiB	gp3	Encrypted	/dev/sda1

Volumes (9) Info												
Last updated 1 minute ago Recycle Bin Actions ▾ Create volume												
Choose filter set ▾ Search												
	Name	Volume ID	Type	Size	IOPS	Throughput	Snapshot ID	Source volume ID	Created			
<input type="checkbox"/>	vol-0ee649bd8f541a8d3	gp3	50 GiB	3000	125	snapshot-01e432c...	-	-	2025/11/			
<input type="checkbox"/>	vol-0211dc1fc046b2739	gp3	20 GiB	3000	125	snapshot-01e432c...	-	-	2025/11/			
<input type="checkbox"/>	vol-0b2c1913f9e6a2642	gp3	100 GiB	3000	125	snapshot-01e432c...	-	-	2025/11/			
<input type="checkbox"/>	vol-00d02c50e1442de32	gp3	100 GiB	3000	125	snapshot-01e432c...	-	-	2025/11/			
<input type="checkbox"/>	vol-07a996db242c4c8a9	gp3	50 GiB	3000	125	snapshot-01e432c...	-	-	2025/11/			
<input type="checkbox"/>	vol-07fb77a2b9ca64831	gp3	20 GiB	3000	125	snapshot-01e432c...	-	-	2025/11/			
<input type="checkbox"/>	vol-0fc3837811cf93515	gp3	50 GiB	3000	125	snapshot-01e432c...	-	-	2025/11/			
<input type="checkbox"/>	vol-08c7493cd92693a0b	gp3	20 GiB	3000	125	snapshot-01e432c...	-	-	2025/11/			
<input type="checkbox"/>	vol-0529f8b3c33f6946d	gp3	100 GiB	3000	125	snapshot-01e432c...	-	-	2025/11/			

6. Snapshot Schedule and Retention Policy

- A comprehensive backup strategy was designed using Amazon Data Lifecycle Manager(DLM) to automate the creation and retention of EBS snapshots.
- Policy Specifications:
- Schedule Name: Project-Daily-Backup
- Target Resource: Instances tagged with Project: Phase1
- Frequency: Daily at 00:00 UTC.
- Retention Rule: Keep the last 7 snapshots (Weekly rotation) to optimize costs while ensuring data availability.
- Implementation Note: Due to lab account permissions restricting the `GetLifecyclePolicies` action, the automated policy could not be fully implemented.

activated. As a Proof of Concept (PoC), a manual snapshot was successfully created and verified for the critical Kafka data volumes.

Snapshots (3) Info							Last updated 2.2 minutes ago	Recycle Bin	Actions ▾	Create snapshot
<input type="checkbox"/>	Name 🔗	Snapshot ID	Full snapshot size	Volume size	Description	Storage tier	Snapshot status			
<input type="checkbox"/>	Automated-Backup-Simula...	snap-00449c2af8e28dfa9	10.91 GiB	100 GiB	Automated-Backup-Simula...	Standard	Completed	Edit	Delete	
<input type="checkbox"/>	Manual-Backup-Phase1-Pr...	snap-0d2f5615a83c9de27	8.38 GiB	20 GiB	Manual-Backup-Phase1-Pr...	Standard	Completed	Edit	Delete	
<input type="checkbox"/>	Manual-Backup-Phase1-Pr...	snap-04b71dfa4fd08e872	9.33 GiB	50 GiB	Manual-Backup-Phase1-Pr...	Standard	Completed	Edit	Delete	

7. Disaster Recovery Procedures

- Objective Recover data from EBS volume failures using manual snapshots created in Phase 1.
- Recovery Procedures
- Identify Failure:
- Detect instance failure via CloudWatch (StatusCheckFailed) or disk I/O errors in logs.
- Stop Instance:
- Stop the affected EC2 instance (do *not* terminate) to freeze data state.
- Note the volume mount point (e.g., /dev/sda1).
- Restore from Backup:
- Locate the latest snapshot tagged Manual-Backup-Proof.
- Create a new volume from the snapshot in the same Availability Zone as the instance.
- Ensure volume settings match (Type: gp3, Encryption: Enabled).
- Swap C Verify:
- Detach the corrupted volume and attach the new one using the correct mount point.

Start the instance and verify service health via SSH.

4. Simple Storage Service (S3)

1. S3 Bucket Naming Convention Document: confirms the naming pattern used for all S3 buckets.

Document Action	Details to be Included
Document Name	S3_Naming_Convention_Document
Naming Pattern	{service}-service-storage-{env}-001
Example Buckets	tts-service-storage-prod-001, stt-service-storage-prod-001, chat-service-storage-prod-001, document-reader-storage-prod-001, quiz-service-storage-prod-001, shared-assets-prod-001

The screenshot shows the AWS S3 console interface. On the left, there's a sidebar with navigation links like 'General purpose buckets', 'Directory buckets', etc. The main area displays a table titled 'General purpose buckets (6)' with columns for 'Name', 'AWS Region', and 'Creation date'. Each row contains a link to the bucket details page. To the right of the table, there are two callout boxes: 'Account snapshot' (updated daily) which provides visibility into storage usage and activity trends, and 'External access summary - new' (updated daily) which helps identify bucket permissions for public access or access from other AWS accounts. The top bar includes the AWS logo, search bar, and account information (Account ID: 7550-9761-8049, voclabs/user4443494=mema.ibrahim.2023@AiU.edu...).

2. Lifecycle Policy Configurations

Bucket Name	Rule Name	Transition	Expiration	Justification
Quiz / Chat / Document	ChatHistory-Archival-Deletion	After 90 days → Standard-IA	After 365 days	Critical data with 1-year retention requirement.
TTS / STT	TTS-Audio-Archival-Deletion	After 60 days → Standard-IA	After 180 days	Temporary files, deleted quickly for cost savings.
Shared Assets	SharedAssets-Cleanup-Deletion	None (remains Standard)	After 180 days	Static assets remain available; old backups removed.

document-reader-storage-prod-001

Lifecycle rule configuration

Review transition and expiration actions

Current version actions	Noncurrent versions actions
<ul style="list-style-type: none"> Day 0 <ul style="list-style-type: none"> Objects uploaded 	<ul style="list-style-type: none"> Day 0 <ul style="list-style-type: none"> No actions defined.

Lifecycle rule configuration

Review transition and expiration actions

Current version actions	Noncurrent versions actions
<ul style="list-style-type: none"> Day 0 <ul style="list-style-type: none"> Objects uploaded 	<ul style="list-style-type: none"> Day 0 <ul style="list-style-type: none"> No actions defined.

shared-assets-prod-001

Lifecycle rule configuration

Review transition and expiration actions

Current version actions	Noncurrent versions actions
<ul style="list-style-type: none"> Day 0 <ul style="list-style-type: none"> Objects uploaded 	<ul style="list-style-type: none"> Day 0 <ul style="list-style-type: none"> No actions defined.

stt-service-storage-prod-001

STT-UploadedAudio-Archival-Deletion

Lifecycle rule configuration

Review transition and expiration actions

Noncurrent versions actions

STT-UploadedAudio-Archival-Deletion

Review transition and expiration actions

Noncurrent versions actions

tts-service-storage-prod-001

TTS-Audio-Archival-Deletion

Lifecycle rule configuration

Review transition and expiration actions

Noncurrent versions actions

Review transition and expiration actions

Current version actions	Noncurrent versions actions
<ul style="list-style-type: none"> Day 0 <ul style="list-style-type: none"> Objects uploaded 	<ul style="list-style-type: none"> Day 0 <ul style="list-style-type: none"> No actions defined.
<p>↓</p>	
<ul style="list-style-type: none"> Day 60 <ul style="list-style-type: none"> Objects move to Standard-IA 	
<p>↓</p>	
<ul style="list-style-type: none"> Day 180 <ul style="list-style-type: none"> Objects expire 	

Delete expired object delete markers or incomplete multipart uploads

Expired object delete markers	Incomplete multipart uploads
-	-

Activate Windows
Go to Settings to activate Windows.

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

chat-service-storage-prod-001

Review transition and expiration actions

Current version actions	Noncurrent versions actions
<ul style="list-style-type: none"> Day 0 <ul style="list-style-type: none"> Objects uploaded 	<ul style="list-style-type: none"> Day 0 <ul style="list-style-type: none"> No actions defined.
<p>↓</p>	
<ul style="list-style-type: none"> Day 90 <ul style="list-style-type: none"> Objects move to Standard-IA 	
<p>↓</p>	
<ul style="list-style-type: none"> Day 365 <ul style="list-style-type: none"> Objects expire 	

Delete expired object delete markers or incomplete multipart uploads

Expired object delete markers	Incomplete multipart uploads
-	-

Activate Windows
Go to Settings to activate Windows.

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

quiz-service-storage-prod-001

ArchiveOldQuizResponses [Info](#)

[Edit](#) [Delete](#) [Actions ▾](#)

Lifecycle rule configuration

Lifecycle rule name ArchiveOldQuizResponses	Prefix -	Minimum object size -
Status Enabled	Object tags -	When no minimum object size is specified, the minimum object size for transitions is determined by the lifecycle configuration. Learn more
Scope Entire bucket		Maximum object size -

Review transition and expiration actions

Current version actions	Noncurrent versions actions
<ul style="list-style-type: none"> Day 0 <ul style="list-style-type: none"> Objects uploaded 	<ul style="list-style-type: none"> Day 0 <ul style="list-style-type: none"> No actions defined.
<p>↓</p>	

Activate Windows
Go to Settings to activate Windows.

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Bucket Policy JSON Documentation

- This section provides the final JSON code used to implement the Principle of Least Privilege and ensure Isolated Storage for each of the six S3 buckets.

Quiz Service Bucket Policy: IAM Role: QuizServiceRole Purpose: Restricts access to the quiz-service-storage-prod-001 bucket to only the Quiz Service Role.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowAccessOnlyForQuizServiceRole",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::[ACCOUNT-ID]:role/QuizServiceRole"
            },
            "Action": "s3:*",
            "Resource": [
                "arn:aws:s3:::quiz-service-storage-prod-001",
                "arn:aws:s3:::quiz-service-storage-prod-001/*"
            ]
        },
        {
            "Sid": "DenyAllOthers",
            "Effect": "Deny",
            "Principal": "*"
        }
    ]
}
```

```

"Principal": "*",
"Action": "s3:*",
"Resource": [
    "arn:aws:s3:::quiz-service-storage-prod-001",
    "arn:aws:s3:::quiz-service-storage-prod-001/*"
],
"Condition": {
    "StringNotLike": {
        "aws:PrincipalArn": "arn:aws:iam::[ACCOUNT-ID]:role/QuizServiceRole"
    }
}
}
]
}

```

2. TTS Service Bucket Policy: IAM Role: TtsServiceRole Purpose: Restricts access to the tts-service-storage-prod-001 bucket to only the TTS Service Role.

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowAccessOnlyForTtsServiceRole",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::[ACCOUNT-ID]:role/TtsServiceRole"
            },
            "Action": "s3:*",
            "Resource": [
                "arn:aws:s3:::tts-service-storage-prod-001",
                "arn:aws:s3:::tts-service-storage-prod-001/*"
            ]
        },
        {
            "Sid": "DenyAllOthers",
            "Effect": "Deny",
            "Principal": "*",

```

```

"Action": "s3:*",
"Resource": [
    "arn:aws:s3:::tts-service-storage-prod-001",
    "arn:aws:s3:::tts-service-storage-prod-001/*"
],
"Condition": {
    "StringNotLike": {
        "aws:PrincipalArn": "arn:aws:iam::[ACCOUNT-ID]:role/TtsServiceRole"
    }
}
}
]
}

```

3. STT Service Bucket Policy

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowAccessOnlyForSttServiceRole",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::[ACCOUNT-ID]:role/SttServiceRole"
            },
            "Action": "s3:*",
            "Resource": [
                "arn:aws:s3:::stt-service-storage-prod-001",
                "arn:aws:s3:::stt-service-storage-prod-001/*"
            ]
        },
        {
            "Sid": "DenyAllOthers",
            "Effect": "Deny",
            "Principal": "*",
            "Action": "s3:*,"
        }
    ]
}
```

```

"Resource": [
    "arn:aws:s3:::stt-service-storage-prod-001",
    "arn:aws:s3:::stt-service-storage-prod-001/*"
],
"Condition": {
    "StringNotLike": {
        "aws:PrincipalArn": "arn:aws:iam::[ACCOUNT-ID]:role/SttServiceRole"
    }
}
}
]
}
```

4. Chat Service Bucket Policy: IAM Role: ChatServiceRole Purpose: Restricts access to the chat-service-storage-prod-001 bucket to only the Chat Service Role.

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowAccessOnlyForChatServiceRole",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::[ACCOUNT-ID]:role/ChatServiceRole"
            },
            "Action": "s3:*",
            "Resource": [
                "arn:aws:s3:::chat-service-storage-prod-001",
                "arn:aws:s3:::chat-service-storage-prod-001/*"
            ]
        },
        {
            "Sid": "DenyAllOthers",
            "Effect": "Deny",
            "Principal": "*",
            "Action": "s3:*",
            "Resource": [

```

```

"arn:aws:s3:::chat-service-storage-prod-001",
"arn:aws:s3:::chat-service-storage-prod-001/*"

],
"Condition": {
    "StringNotLike": {
        "aws:PrincipalArn": "arn:aws:iam::[ACCOUNT-ID]:role/ChatServiceRole"
    }
}
}
]
```

5. Document Reader Bucket Policy:

- IAM Role: DocumentReaderRole
- Purpose: Restricts access to the document-reader-storage-prod-001 bucket to only the Document Reader Role.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowAccessOnlyForDocumentReaderRole",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::[ACCOUNT-ID]:role/DocumentReaderRole"
            },
            "Action": "s3:*",
            "Resource": [
                "arn:aws:s3:::document-reader-storage-prod-001",
                "arn:aws:s3:::document-reader-storage-prod-001/*"
            ]
        },
        {
            "Sid": "DenyAllOthers",
            "Effect": "Deny",
            "Principal": "*",

```

```

"Action": "s3:*",
"Resource": [
    "arn:aws:s3:::document-reader-storage-prod-001",
    "arn:aws:s3:::document-reader-storage-prod-001/*"
],
"Condition": {
    "StringNotLike": {
        "aws:PrincipalArn": "arn:aws:iam::[ACCOUNT-ID]:role/DocumentReaderRole"
    }
}
}
]
}

```

6. Shared Assets Bucket Policy

IAM Role: SharedAssetsRole

Purpose: Restricts access to the shared-assets-prod-001 bucket to only the Shared Assets Role.

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowAccessOnlyForSharedAssetsRole",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::[ACCOUNT-ID]:role/SharedAssetsRole"
            },
            "Action": "s3:*",
            "Resource": [
                "arn:aws:s3:::shared-assets-prod-001",
                "arn:aws:s3:::shared-assets-prod-001/*"
            ]
        },
        {
            "Sid": "DenyAllOthers",
            "Effect": "Deny",

```

```

"Principal": "*",
"Action": "s3:*",
"Resource": [
    "arn:aws:s3::::shared-assets-prod-001",
    "arn:aws:s3::::shared-assets-prod-001/*"
],
"Condition": {
    "StringNotLike": {
        "aws:PrincipalArn": "arn:aws:iam::[ACCOUNT-ID]:role/SharedAssetsRole"
    }
}
}
]
}
```

5. Virtual Private Cloud (VPC)

1. Purpose of the VPC

- Network isolation
- Traffic segmentation
- Controlled internet access
- Strong security boundaries for each tier

2. VPC Configuration Requirements

2.1 Create VPC

CIDR Block: 10.0.0.0/16

This CIDR allows up to 65,536 IP addresses, suitable for multi-tier distributed architecture.

2.2 Subnet Architecture

Subnet Tier	CIDR Block	Availability Zone
Public Subnet 1	10.0.1.0/24	AZ1
Public Subnet 2	10.0.2.0/24	AZ2
Private Subnet 1	10.0.10.0/24	AZ1
Private Subnet 2	10.0.11.0/24	AZ2

Data Subnet 1 (RDS)	10.0.20.0/24	AZ1
Data Subnet 2 (RDS)	10.0.21.0/24	AZ2
Kafka Subnet 1	10.0.30.0/24	AZ1
Kafka Subnet 2	10.0.31.0/24	AZ2

2.3 Internet Gateway Configuration

Attach an Internet Gateway (IGW) to the VPC to enable outbound and inbound internet traffic for public subnets.

2.4 NAT Gateway Configuration

Deploy one NAT Gateway per Availability Zone inside each public subnet. Private subnets route 0.0.0.0/0 traffic to NAT Gateways instead of the Internet Gateway.

2.5 Route Tables Setup

- Public route table → routes 0.0.0.0/0 to Internet Gateway
- Private route table → routes 0.0.0.0/0 to NAT Gateway
- Data subnets → no internet route (fully isolated)
- Kafka subnets → no internet route (internal cluster only)

2.6 Network ACLs

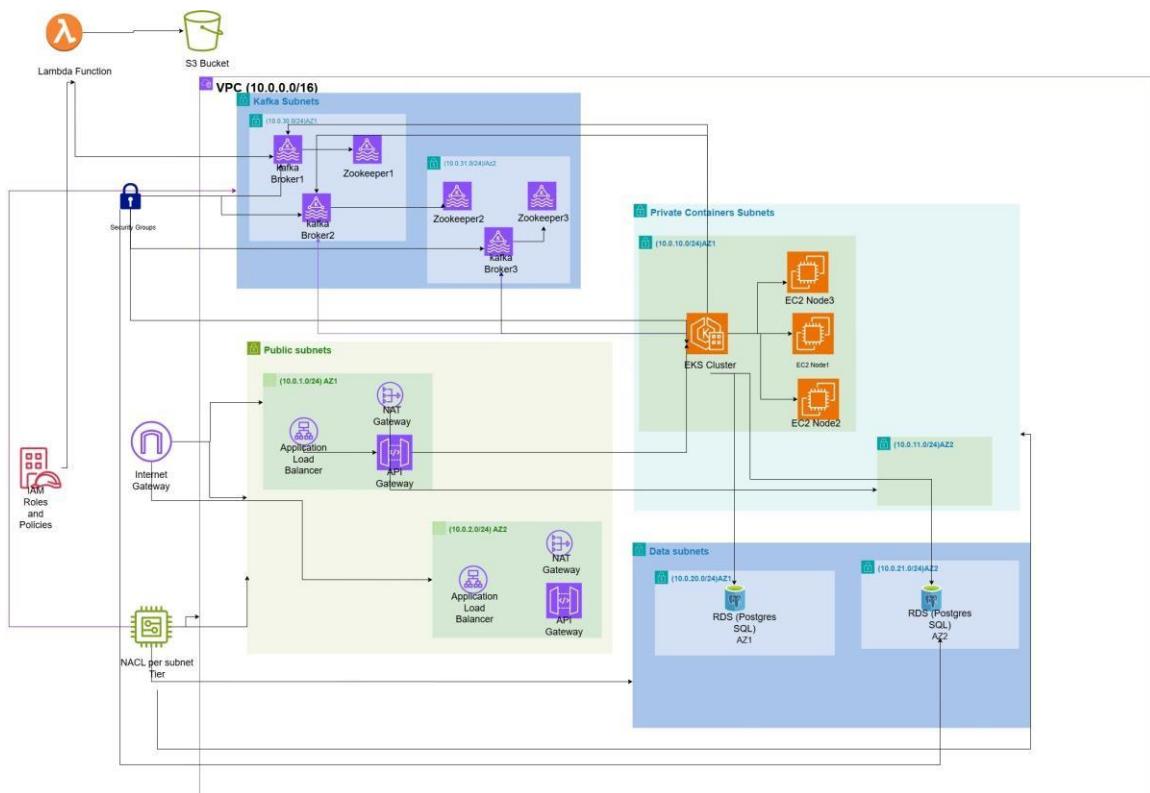
Network ACLs (NACLs) provide stateless packet filtering at the subnet level. Each subnet tier receives its own NACL for boundary-level protection.

2.7 Security Groups

Security Groups act as stateful firewalls for EC2, RDS, containers, load balancers, and Kafka brokers.

3. Deliverables

3.1 VPC Architecture Diagram



3.2 Subnet Allocation Table

Subnet Name	CIDR	AZ	Purpose
public-subnet-1a	10.0.1.0/24	us-east-1a	Load Balancers
public-subnet-1b	10.0.2.0/24	us-east-1b	Load Balancers
private-subnet-1a	10.0.10.0/24	us-east-1a	Containers
private-subnet-1b	10.0.11.0/24	us-east-1b	Containers
data-subnet-1a	10.0.20.0/24	us-east-1a	RDS Database
data-subnet-1b	10.0.21.0/24	us-east-1b	RDS Database
kafka-subnet-1a	10.0.30.0/24	us-east-1a	Kafka Cluster
kafka-subnet-1b	10.0.31.0/24	us-east-1b	Kafka Cluster

3.3 Security Group Definitions

Security Group	Inbound Rules	Outbound Rules	Description
SG-loadbalancer	HTTP 80, HTTPS 443 (0.0.0.0/0)	Allow all traffic	Allow Internet traffic to LB
SG-containers	TCP 3000, 8000 from SG-loadbalancer	Allow all traffic	App containers access
SG-RDS	PostgreSQL 5432 from SG-containers	Allow all traffic	DB access from containers
SG-Kafka	TCP 9092 from SG-containers	Allow all traffic	Kafka access from containers

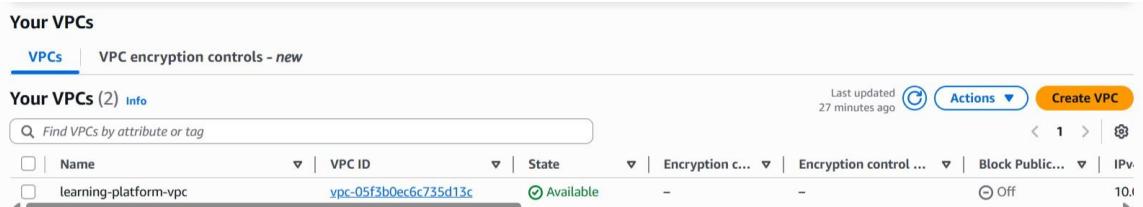
3.4 Network ACL Rules

NACL Name	Inbound Rules	Outbound Rules
nacl-public	Allow 80,443 from 0.0.0.0/0	Allow ephemeral (1024-65535)
nacl-private	Allow ephemeral from 10.0.0.0/16	Allow ephemeral (1024-65535)
nacl-data	Deny all from public subnets	Allow local traffic only
nacl-kafka	Allow 9092 from private subnets	Allow internal ephemeral only

4. Step-by-Step Implementation

1. Create the VPC

- Create a new VPC with CIDR block **10.0.0.0/16** to serve as the main isolated network.



2. Create the Subnets (8 total)

- Create **2 Public subnets** (10.0.1.0/24, 10.0.2.0/24) across two AZs for load balancers.
- Create **2 Private subnets** (10.0.10.0/24, 10.0.11.0/24) for application containers.

- Create **2 Data subnets** (10.0.20.0/24, 10.0.21.0/24) dedicated to RDS.
- Create **2 Kafka subnets** (10.0.30.0/24, 10.0.31.0/24) for the Kafka cluster.

Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR
kafka-subnet-1a	subnet-0a1f2b9c90e0f71c	Available	vpc-05f3b0ec6c735d13c learn...	Off	10.0.30.0/24
private-subnet-1b	subnet-0d680853d07e59d9c	Available	vpc-05f3b0ec6c735d13c learn...	Off	10.0.11.0/24
-	subnet-0d60d38ca1820-90f	Available	vpc-078609144298b253e	Off	172.31.80.0/2
data-subnet-1a	subnet-07c44a5aca6162980	Available	vpc-05f3b0ec6c735d13c learn...	Off	10.0.20.0/24
private-subnet-1a	subnet-081bbc6179e84d6f7	Available	vpc-05f3b0ec6c735d13c learn...	Off	10.0.10.0/24
-	subnet-054d162da155a8480	Available	vpc-078609144298b253e	Off	172.31.64.0/2
public-subnet-1a	subnet-0e9ae6a972070b954	Available	vpc-05f3b0ec6c735d13c learn...	Off	10.0.1.0/24
data-subnet-1b	subnet-009f793d72f9e3ef6	Available	vpc-05f3b0ec6c735d13c learn...	Off	10.0.21.0/24
kafka-subnet-1b	subnet-0414bb6881a731877	Available	vpc-05f3b0ec6c735d13c learn...	Off	10.0.31.0/24
public-subnet-1b	subnet-0f27d0f340bfaebd	Available	vpc-05f3b0ec6c735d13c learn...	Off	10.0.2.0/24
-	subnet-044d1d411c328h1f	Available	vnr-078609144298b253e	Off	172.31.16.0/2

3. Attach an Internet Gateway

- Create and attach an IGW to the VPC to enable internet access for public subnets.

Name	Internet gateway ID	State	VPC ID	Owner
learning-platform-igw	igw-08b3a22665ff0ab7e	Attached	vpc-05f3b0ec6c735d13c learning-plat...	755097618049
-	igw-0abd9e2f48428d687	Attached	vpc-078609144298b253e	755097618049

4. Create Public Route Table

- Add route **0.0.0.0/0 → Internet Gateway**.
- Associate the two public subnets with this table.

Name	Route table ID	Explicit subnet assoc...	Edge associations	Main	VPC
data-rt	rtb-0ff019ea106f24587	2 subnets	-	No	vpc-05f3b0ec6c735d13c learn.
-	rtb-06cd9fb724e35ea2	-	-	Yes	vpc-05f3b0ec6c735d13c learn.
-	rtb-060a3a3ae9a31c2ef	-	-	Yes	vpc-078609144298b253e
public-rt	rtb-0a388648951e2a8ae	2 subnets	-	No	vpc-05f3b0ec6c735d13c learn.
private-rt	rtb-011c3eb0ebcf4b260	2 subnets	-	No	vpc-05f3b0ec6c735d13c learn.
kafka-rt	rtb-0a1661d6c770602ed	2 subnets	-	No	vpc-05f3b0ec6c735d13c learn.

5. Create NAT Gateways

- Create one NAT Gateway in each public subnet.
- Allocate Elastic IPs for both NATs.

Name	NAT gateway ID	Connectivity...	State	State message	Availability ...	Route table ID	VPC
nat-gw-1a	nat-0fb1fbeeaa80f9038	Public	Available	-	Zonal	-	
nat-gw-1b	nat-069d6d3a6f33db02a	Public	Available	-	Zonal	-	

6. Create Private Route Tables

- For each AZ, create a private route table.
- Add route **0.0.0.0/0 → corresponding NAT Gateway**.
- Associate both private subnets.

Name	Route table ID	Explicit subnet assoc...	Edge associations	Main	VPC
data-rt	rtb-0ff019ea106f24587	2 subnets	-	No	vpc-05f3b0ec6c735d13c learn.
-	rtb-06cd9fb724e35ea2	-	-	Yes	vpc-05f3b0ec6c735d13c learn.
-	rtb-060a3a3ae9a31c2ef	-	-	Yes	vpc-078609144298b253e
public-rt	rtb-0a388648951e2a8ae	2 subnets	-	No	vpc-05f3b0ec6c735d13c learn.
private-rt	rtb-011c3eb0ebcf4b260	2 subnets	-	No	vpc-05f3b0ec6c735d13c learn.
kafka-rt	rtb-0a1661d6c770602ed	2 subnets	-	No	vpc-05f3b0ec6c735d13c learn.

7. Create Route Tables for Data & Kafka

- Create two additional route tables (data + kafka).
- **No internet routes** are added.
- Associate them with their respective subnets.

<input type="checkbox"/>	Name	Route table ID	Explicit subnet associations	Edge associations	Main	VPC
○	data-rt	rtb-0ff019ea106f24587	2 subnets	-	No	vpc-05f3b0ec6c735d13c learn.
○	-	rtb-06cd9fb724e35ea2	-	-	Yes	vpc-05f3b0ec6c735d13c learn.
○	-	rtb-060a3a3ae9a31c2ef	-	-	Yes	vpc-078609144298b253e
○	public-rt	rtb-0a388648951e2a8ae	2 subnets	-	No	vpc-05f3b0ec6c735d13c learn.
○	private-rt	rtb-011c3eb0ebcf4b260	2 subnets	-	No	vpc-05f3b0ec6c735d13c learn.
○	kafka-rt	rtb-0a1661d6c770602ed	2 subnets	-	No	vpc-05f3b0ec6c735d13c learn.

8. Create Security Groups

- Define SGs for each tier (Public, Private, Database, Kafka).
- Configure inbound/outbound rules based on required communication paths.

Security Groups (12) info					
<input type="checkbox"/>	Name	Security group ID	Security group name	VPC ID	Description
○	-	sg-0849158143a6fd693	SG-RDS	vpc-05f3b0ec6c735d13c	Allow PostgreSQL
○	-	sg-0479b2228ba84c18b	rds-postgres-sg	vpc-05f3b0ec6c735d13c	Security group for RDS
○	-	sg-0866a5c882ac1ff97	container-hosts-sg-dev	vpc-05f3b0ec6c735d13c	Security group for container hosts
○	-	sg-0ff6c301faa11bee9	Default security group1	vpc-05f3b0ec6c735d13c	Default security group
○	-	sg-03c7c12936d38238	default	vpc-078609144298b253e	default VPC security group
○	-	sg-021bd251e998c9f94	SG-containers	vpc-05f3b0ec6c735d13c	Allow TCP 3000, 5432
○	-	sg-0cc54f03e923084dc	MyDefaultSG	vpc-078609144298b253e	Default security group
○	-	sg-0de7cec3723401af3	SG-loadbalancer	vpc-05f3b0ec6c735d13c	Allow HTTP & HTTPS
○	-	sg-0af65e18b75f497c3	default	vpc-05f3b0ec6c735d13c	default VPC security group
○	-	sg-0898c0d62dda2db02	SG-Kafka	vpc-05f3b0ec6c735d13c	Allow TCP 9092 for Kafka
○	-	sg-0e82a99b873e5ab31	ec2-sg	vpc-05f3b0ec6c735d13c	launch-wizard-1 configuration

9. Create Network ACLs

- Create NACLs for each subnet tier.
- Define basic inbound/outbound rules to control traffic at subnet level.

Network ACLs (6) info						
<input type="checkbox"/>	Name	Network ACL ID	Associated with	Default	VPC ID	Inbound
○	-	acl-0400bcd8afc4a9cca	6 Subnets	Yes	vpc-078609144298b253e	2 Inbo
○	nacl-public	acl-060a54d853e0b0f5a	2 Subnets	No	vpc-05f3b0ec6c735d13c / learning-pla...	3 Inbo
○	nacl-private	acl-0afa7c632c7b9fe73	2 Subnets	No	vpc-05f3b0ec6c735d13c / learning-pla...	2 Inbo
○	nacl-kafka	acl-0f4e700e31e1620aa	2 Subnets	No	vpc-05f3b0ec6c735d13c / learning-pla...	4 Inbo
○	-	acl-0bb6b37989c4630ad	-	Yes	vpc-05f3b0ec6c735d13c / learning-pla...	2 Inbo
○	nacl-data	acl-024dedd43782dcf8b	2 Subnets	No	vpc-05f3b0ec6c735d13c / learning-pla...	4 Inbo

6. Results

The final VPC architecture was successfully deployed across two Availability Zones. All required subnets, route tables, NAT gateways, Internet gateway, security groups, and Network ACLs were configured and verified.

The environment achieves:

- **Full network isolation** between public, private, data, and Kafka tiers.
- **High availability** using two AZs for each subnet tier.

- **Secure outbound access** from private subnets using NAT Gateways.
- **No public exposure** for database and Kafka subnets.
- **Least-privilege security** enforced with tier-based Security Groups and NACLs.

6. Lambda Functions

AWS EC2 Dashboard:

- Account and Region Information
- **Active Region:** United States (N. Virginia).
- **Default VPC ID:** vpc-078609144298b253e.
- **Service Health Status:** The displayed status is "**This service is operating normally.**"
- Key Findings for Documentation
- The environment currently has **G running instances** supported by **3 active Load Balancers**.
- Network access is managed by **12 Security Groups**.
- The system health is reported as **normal** within the N. Virginia region.

The screenshot shows the AWS EC2 Dashboard for the United States (N. Virginia) region. The left sidebar includes links for Dedicated Hosts, Capacity Reservations, Capacity Manager, 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, Trust Stores), and Auto Scaling. The main content area displays the following information:

Resources					
Instances (running)	9	Auto Scaling Groups	1	Capacity Reservations	0
Dedicated Hosts	0	Elastic IPs	2	Instances	9
Key pairs	3	Load balancers	2	Placement groups	0
Security groups	12	Snapshots	1	Volumes	9

Launch instance: To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud. Buttons for "Launch instance" and "Migrate a server". Note: Your instances will launch in the United States (N. Virginia) Region.

Service health: AWS Health Dashboard. Status: This service is operating normally.

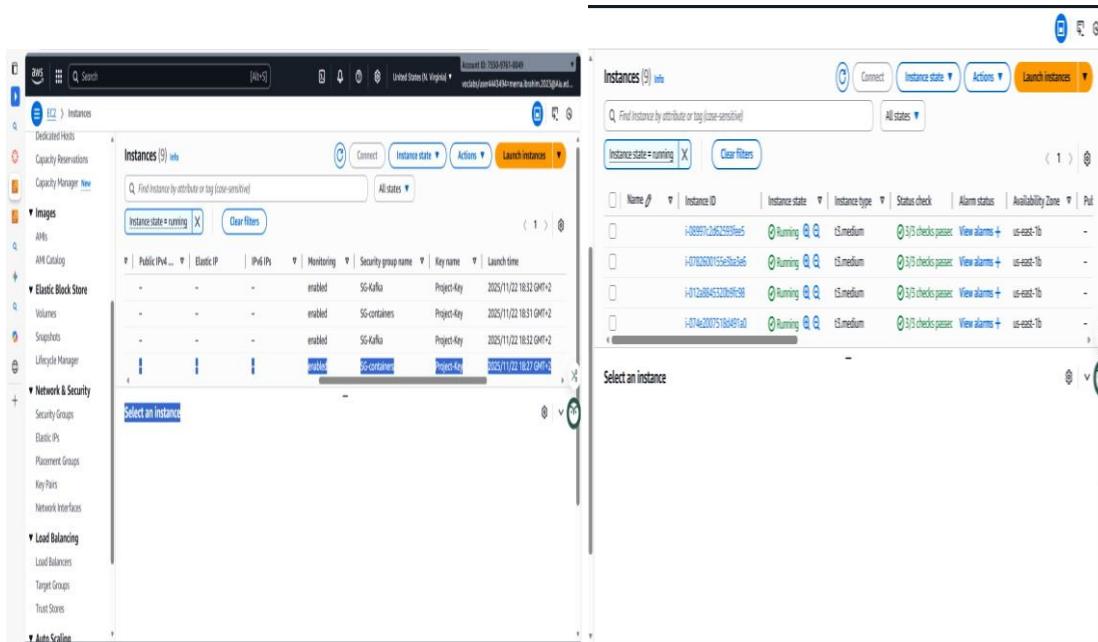
Account attributes: Default VPC (vpc-078609144298b253e). Settings include Data protection and security, Allowed AMIs, Zones, EC2 Serial Console, Default credit specification, and EC2 console preferences.

Explore AWS: 10 Things You Can Do Today to Reduce AWS Costs, Amazon GuardDuty Malware Protection.

AWS EC2 Infrastructure:

- EC2 Dashboard Overview
- The EC2 Dashboard confirms that the service is operating normally within the region.
- **Running Instances:** There are currently **G active EC2 virtual servers running**.

- **Load Balancers:** The account utilizes **3** active Load Balancers.
- **Security Groups:** A total of **12** Security Groups are defined for network control.
- Load Balancing Service Components
- The infrastructure features two key active Application Load Balancers (ALB-1 and ALB-4) and several Target Groups.
- A. Load Balancer Details: ALB-4
- **Type:** Application Load Balancer (ALB).
- **Scheme:** Internet-facing, accessible from the public internet.
- **Availability Zones (AZs):** Configured for high availability across us-east-1a and us-east-1b.
- **Listener Configuration:** Listens on **HTTP:3000** and forwards all traffic to the **target4** Target Group.



AWS EC2 Infrastructure Consolidated:

- Global EC2 Overview
- The EC2 Dashboard confirms the service is operating normally.
- **Running Instances:** There are currently **4** active EC2 virtual servers.
- **Load Balancers:** The account utilizes **3** active Load Balancers.
- **Security Groups:** A total of **12** Security Groups are defined for network control.
- Load Balancing Service Components
- The environment uses **Application Load Balancers (ALBs)** for traffic distribution.
- A. Load Balancer List

- **ALB-1:** Active, type application, scheme Internet-facing.
- **ALB-4:** Active, type application, scheme Internet-facing.
- B. Load Balancer Details: ALB-4
- **AZ Configuration:** Configured for high availability across us-east-1a and us-east-1b.
- **Listener Configuration:** Listens on **HTTP:3000** and forwards all traffic to the **target4** Target Group.

Name	State	Type	Scheme	IP address type	VPC ID	Availability Z
ALB-1	Active	application	Internet-facing	IPv4	vpc-05f3b0ec6c735d13c	2 Availability
ALB-4	Active	application	Internet-facing	IPv4	vpc-05f3b0ec6c735d13c	2 Availability

Network and Load Balancer:

- Load Balancer Details: ALB-4
- The configuration of ALB-4 indicates its role as a public-facing entry point for an application.
- **Load Balancer Type: Application** (Layer 7).
- **Status: Active.**
- **Scheme: Internet-facing**, meaning it has a public DNS name and is accessible from the internet.
- **Availability Zones (AZs):** Configured for high availability across two AZs: **us-east-1a** and **us-east-1b**.
- **Listener Configuration:** The visible listener is for **HTTP:3000**. This suggests the application or target group is accessed via port 3000.
- **DNS Name:** ALB-4-941778476.us-east-1.elb.amazonaws.com (Used to access the application).
- Security Group Configurations (Network Access Rules)

- These three Security Groups define the access policies for the different components in your infrastructure (web servers, Kafka, containers).
- A. Default Security Group 1 (sg-0ff6c301faa11bee9)
- This group typically serves as a baseline for general instances that need public access.
- Security Group Name:** Default security group 1
- Purpose:** Default security group for EC2 instances.
- Key Inbound Rules:**
 - Port 22 (SSH):** Allows secure remote administration access.
 - Port 80 (HTTP):** Allows standard web traffic (unencrypted).
- SG-containers (sg-021bd251e998c9f94)
- This group is designed to host containerized applications and restricts access to ensure traffic flows through the Load Balancer.
- Security Group Name:** SG-containers
- Description:** "Allow TCP 3000, 8000 from sg-loadbalancer."
- Key Inbound Rules:**
 - Port 80 (HTTP):** Standard web access.
 - Port 3000 (Custom TCP):** Used for application traffic (matches the ALB-4 listener port).
 - Port 8000 (Custom TCP):** Used for a secondary application or API service.

The screenshot shows the AWS CloudFormation console with the following details:

- Region:** United States (N. Virginia)
- Account ID:** 7550-9761-8049
- Load Balancer:** ALB-4
- Details:**
 - Load balancer type: Application
 - Status: Active
 - VPC: [vpc-05f3b0ec6c735d13c](#)
 - Load balancer IP address type: IPv4
 - Scheme: Internet-facing
 - Hosted zone: Z35XDOTRQ7X7K
 - Availability Zones:
 - subnet-0e9ae6a972070b954 us-east-1a (use1-az4)
 - subnet-0d680853d07e59d9c us-east-1b (use1-az6)
 - Date created: November 22, 2025, 18:35 (UTC+02:00)
- Listeners and rules:** (1)
 - Protocol:Port: Default action
 - Rules: ARN, Security policy, Default SSL

AWS Application Load Balancer (ALB) Target Group:

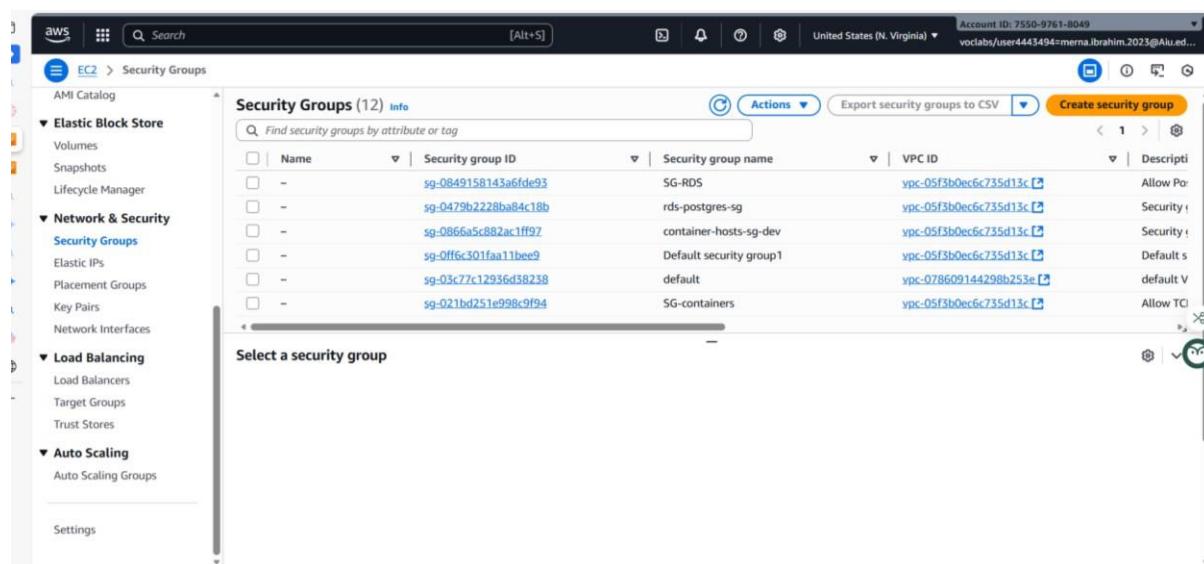
- Target Group Details
- This Target Group is configured to handle traffic using the **HTTP** protocol on port **3000** (HTTP: 3000), with the protocol version set to HTTP1. The targets registered within this group are specified by their **IP address** (Target Type: ip). This Target Group is associated with an Application Load Balancer identified as ALB - 4 within the specified VPC.
- Health Status Summary
- The health summary shows that the Target Group currently has a **Total of G targets** registered. Crucially, all **G targets are Healthy**, meaning they are successfully passing the configured health checks and are actively receiving traffic from the Load Balancer. There are **0 Unhealthy** targets, indicating that the backend services are fully operational and performing as expected. The status also confirms **0 Unused** and **0 Initial** targets, suggesting a stable operational environment.

The screenshot shows the AWS EC2 Target Groups console. The left sidebar navigation includes: Dedicated Hosts, Capacity Reservations, Capacity Manager (New), Images (AMIs, AMI Catalog), Elastic Block Store (Volumes, Snapshots, Lifecycle Manager), Network & Security (Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces), and Load Balancing (Load Balancers, Target Groups, Trust Stores). The main content area displays the details for 'target4'. The 'Details' section shows the ARN: arn:aws:elasticloadbalancing:us-east-1:755097618049:targetgroup/target4/cb9bb2dd22ef85a0, Target type: Instance, Protocol: Port, Port: 3000, Protocol version: HTTP, IP address type: IPv4, Load balancer: ALB-4, and VPC: vpc-05f3b0ec6c735d13c. Below this, a table summarizes target states: Total targets (9), Healthy (0), Unhealthy (9), Unused (0), Initial (0), and Draining (0). A note indicates 0 Anomalous targets. A link to 'Distribution of targets by Availability Zone (AZ)' is provided. At the bottom, tabs for Targets, Monitoring, Health checks, Attributes, and Tags are visible, along with buttons for Anomaly mitigation (Not applicable), Deregister, and Register targets. The footer includes links for CloudShell, Feedback, and various AWS terms like © 2025, Amazon Web Services, Inc. or its affiliates., Privacy, Terms, and Cookie preferences.

AWS Security Groups:

- Key Groups and Functionality
- The list highlights several specialized groups relevant to a typical application architecture:
- **SG-RDS and rds-postgres-sg:** These are dedicated security groups for database instances. Their primary function is to strictly limit inbound access to database ports (e.g., 5432) only from trusted application components, often achieved by referencing the Security Group of the application's EC2 instances or container hosts.
- **container-hosts-sg-dev and SG-containers:** These groups are used to secure the underlying hosts running containerized applications. They manage which ports are exposed to the outside world, and critically, which ports are accessible by the Load Balancer.

- Relevance to the Load Balancer (Loader) Component
- Security Groups are fundamental to the Load Balancer's operation and security posture:
- **Load Balancer Frontend:** A dedicated Security Group for the Application Load Balancer (ALB) must permit inbound client traffic (HTTP on port 80 and HTTPS on port 443) from the public internet.
- **Target Backend:** The Security Groups applied to the application targets (like the servers using SG-containers) must contain a specific **Inbound Rule** that allows traffic on the application's listening port (e.g., port 3000, as configured in the Target Group) **only** from the **Load Balancer's Security Group ID**. This mechanism ensures that backend servers are unreachable directly from the internet, enforcing that all traffic flows securely through the Load Balancer.



The screenshot shows the AWS EC2 Security Groups page. The left sidebar navigation includes '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, Trust Stores), and 'Auto Scaling' (Auto Scaling Groups). The main content area displays a table titled 'Security Groups (12)'. The columns are: Name, Security group ID, Security group name, VPC ID, and Description. The table lists the following entries:

Name	Security group ID	Security group name	VPC ID	Description
-	sg-0849158143a6fde93	SG-RDS	vpc-05f3b0ec6c735d13c	Allow Po...
-	sg-0479b2228ba84c18b	rds-postgres-sg	vpc-05f3b0ec6c735d13c	Security r...
-	sg-0866a5c882ac1ff97	container-hosts-sg-dev	vpc-05f3b0ec6c735d13c	Security r...
-	sg-0ff6c301faa11bee9	Default security group1	vpc-05f3b0ec6c735d13c	Default s...
-	sg-03c77c12936d38238	default	vpc-078609144298b253e	default V...
-	sg-021bd251e098c9f94	SG-containers	vpc-05f3b0ec6c735d13c	Allow TC...

Below the table, a modal window titled 'Select a security group' is open, showing a list of security groups: SG-RDS, rds-postgres-sg, container-hosts-sg-dev, Default security group1, default, and SG-containers. The 'SG-containers' entry is highlighted.

The screenshot shows the AWS EC2 Security Groups console. On the left, there's a navigation sidebar with sections like 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, Trust Stores), Auto Scaling (Auto Scaling Groups), and Settings. The main content area is titled "sg-021bd251e998c9f94 - SG-containers". It displays "Details" for the security group, including its name (SG-containers), ID (sg-021bd251e998c9f94), description ("Allow TCP 3000, 8000 from SG-loadbalancer"), VPC ID (vpc-05f3b0ec6c735d13c), owner (755097618049), inbound rules count (3 Permission entries), and outbound rules count (1 Permission entry). Below this, there are tabs for Inbound rules, Outbound rules, Sharing - new, VPC associations - new, and Tags. The "Inbound rules" tab is selected, showing a table with three entries:

Name	Security group rule ID	IP version	Type	Protocol	Port range
-	sgr-03932b9b172b7abb3	-	Custom TCP	TCP	8000
-	sgr-0ff6e10e70bd8b5b1	-	HTTP	TCP	80
-	sgr-00984ab80c7556093	-	Custom TCP	TCP	3000

At the bottom of the page, there are links for CloudShell and Feedback, and a footer with copyright information: © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences.

7. Elastic Load Balancer (ELB)

All Lambda Functions:

Sourc code Reposatoy: <https://github.com/Nourhannee/Lambda-Functions/tree/main>

The screenshot shows the AWS Lambda Functions console. The top navigation bar includes the AWS logo, a search bar, and account information (Account ID: 7550-9761-8049, United States (N. Virginia)). The main content area is titled "Functions (4/9)". It displays a table of functions with the following columns: Function name, Description, Package type, Runtime, and Last modified. The table lists the following functions:

Function name	Description	Package type	Runtime	Last modified
<input checked="" type="checkbox"/> S3EventProcessor	-	Zip	Python 3.14	4 hours ago
<input checked="" type="checkbox"/> KafkaMonitor	-	Zip	Python 3.14	4 hours ago
<input type="checkbox"/> RedshiftOverwatch	Deletes Redshift Cluster if the count is more than 2.	Zip	Python 3.9	1 day ago
<input type="checkbox"/> ModLabRole	updates LabRole to allow it to assume itself	Zip	Python 3.9	1 day ago
<input checked="" type="checkbox"/> AutoScaling	-	Zip	Python 3.14	4 hours ago
<input type="checkbox"/> RedshiftEventSubscription	Create Redshift event subscription to SNS Topic.	Zip	Python 3.9	1 day ago
<input type="checkbox"/> MainMonitoringFunction	-	Zip	Python 3.9	1 day ago
<input type="checkbox"/> RoleCreationFunction	Create SLR if absent	Zip	Python 3.9	1 day ago
<input checked="" type="checkbox"/> CleanupTasksScheduler	-	Zip	Python 3.14	32 minutes ago

At the bottom of the page, there are links for CloudShell and Feedback, and a footer with copyright information: © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences.

Steps to Create an AWS Lambda Function (Apply these steps on all lambda functions):

- **Open AWS Console** : Go to Lambda Service.
- **Create Function** : Select "Author from scratch".
- **Configure Basic Settings** : Set function name, runtime, architecture, and execution role.
- **Add Triggers** : (e.g., S3, CloudWatch Events, API Gateway, Kafka).
- **Write or Upload Code** : Use inline editor or upload a ZIP/package.
- **Set Environment Variables** : Add keys, configs, or connection strings.
- **Attach Lambda Layers** : Include shared libraries or dependencies.
- **Configure VPC Access (optional)** : If Lambda needs private resources (e.g., Kafka, RDS).
- **Monitor Logs & Metrics** : Use CloudWatch for performance and error tracking.

Common Lambda Layers – Steps for Using Shared Libraries

Create the Layer:

Prepare your shared libraries in a .zip file with folder structure:

/python/lib/pythonX.X/site-packages/

Upload the Layer to AWS Lambda:

Console → Layers → Create layer → Upload ZIP → Select runtime

Attach the Layer to a Lambda Function:

Function → Layers → Add a layer → Choose your created layer

Use Libraries in Your Function:

All libraries in the Layer are available to your Lambda without bundling them again

Update the Layer When Needed:

Upload a new version → Functions using the layer get the update automatically

1. Create Lambda function for S3 event processing:

- Function Identity:**

This is our S3EventProcessor Lambda function, which is a core component of the Serverless Layer in our infrastructure. It directly addresses the project requirement for S3 event processing.

The screenshot shows the AWS Lambda console interface. The top navigation bar includes the AWS logo, search bar, and account information (Account ID: 7550-9761-8049, vclabs/user443494=merna.ibrahim.2023@AiU.edu). The main title is "S3EventProcessor". Below it, there are tabs for "Function overview" (selected), "Info", "Diagram", and "Template". The "Diagram" tab shows a function node labeled "S3EventProcessor" with a "Layers" dependency. An "S3" trigger is attached to the function. Buttons for "+ Add destination" and "+ Add trigger" are visible. To the right, there are sections for "Description", "Last modified" (2 hours ago), "Function ARN" (arn:aws:lambda:us-east-1:755097618049:function:S3EventProcessor), and "Function URL" (Info). At the bottom, there are tabs for "Code" (selected), "Test", "Monitor", "Configuration", "Aliases", and "Versions".

I created the S3EventProcessor Lambda and configured its S3 trigger.

Whenever a file is uploaded to our S3 bucket, the event automatically invokes the Lambda, which reads the file, extracts metadata and a preview, and publishes a structured event into our Kafka pipeline.

Runtime Environment:

Here you can see the runtime configuration for our S3 event processor. We chose Python 3.14 for performance and compatibility, as Python is the base language for all our microservices.

Utilizing Lambda Layers:

The bottom section highlights the use of Lambda Layers. We have attached the **common-dependencies layer**.

"This Layer is used to bundle all shared dependencies—such as our Kafka client library or any common utility packages—avoiding the need to include them in the deployment package for every Lambda function. This makes our deployment process faster and our code base cleaner."

"By using a layer, we ensure **consistency** and make it easier to manage updates for all our different Lambda functions across the project."

This screenshot shows the AWS Lambda function configuration page for 'S3EventProcessor'. It includes sections for 'Code properties', 'Runtime settings', and 'Layers'.

- Code properties:** Shows package size (1.1 kB), SHA256 hash (1HdAdm26nLOUMQwnXZnIDVycx3r84RUyC0doBw47PYE=), and last modified (2 hours ago). It also indicates 'Encryption with AWS KMS customer managed KMS key'.
- Runtime settings:** Set to Python 3.14, Handler: lambda_function.lambda_handler, and Architecture: x86_64. Buttons for 'Edit' and 'Edit runtime management configuration' are present.
- Layers:** A single layer named 'common-dependencies' is listed, version 1, compatible with Python 3.14 and x86_64, with ARN arn:aws:lambda:us-east-1:755097618049:layer:common-dependencies:1. Buttons for 'Edit' and 'Add a layer' are shown.

Configuration ScreenShotes

This screenshot shows the 'Configuration' tab for the 'S3EventProcessor' Lambda function. The left sidebar lists various configuration options: General configuration, Triggers, Permissions, Destinations, Function URL, Environment variables, Tags, VPC, RDS databases, and Monitoring and operations tools. The 'Triggers' section is selected and displays one trigger named 'S3: document-reader-storage-prod-001'.

Trigger Type	Bucket ARN	Event Type	Is Complex Statement	Notification Name	Service Principal	Source Account	Statement ID
S3	arn:aws:s3:::document-reader-storage-prod-001	s3:ObjectCreated*	No	dc7d9f8b-0e23-4513-958b-f5af888815d	s3.amazonaws.com	755097618049	lambda-f0bd4aed-cbf7-409c-8578-95da28ad66ff

Security group ID	Protocol	Ports	Source
sg-0898c0d62dda2db02	Custom TCP	80	sg-0ff6c301faa11bee9
sg-0898c0d62dda2db02	Custom TCP	3000	sg-0ff6c301faa11bee9
sg-0898c0d62dda2db02	Custom TCP	9092	sg-021bd251e998c9f94

2. Cleanup tasks Lambda:

- **Function Identity's Purpose:**

This is our second required Lambda function: the CleanupTaskScheduler. This function is designed to handle all scheduled cleanup tasks, fulfilling the project requirement to Implement Lambda for cleanup tasks like deleting old files and archiving data.

- **Trigger Mechanism:**

Unlike the previous function which was event-driven (S3), this function is time-driven. As shown in the diagram, it is triggered by EventBridge (CloudWatch Events).

- **Scheduling and Frequency:**

We configure EventBridge using a Cron expression to define the exact schedule, for example, running every night at 2:00 AM

- **Code Function's Project Requirement:**

The Python code represents our CleanupTaskScheduler Lambda function. It directly addresses the project requirement to implement Lambda for cleanup tasks (delete old files).

This function is automatically triggered by EventBridge (CloudWatch Events) on a fixed schedule, ensuring regular maintenance without manual intervention.

Successfully updated the function CleanupTasksScheduler.



CleanupTasksScheduler

Throttle

Copy ARN

Actions ⏮

▼ Function overview Info

Export to Infrastructure Composer

Download ⏮

Diagram

Template

EventBridge (CloudWatch Events)

+ Add destination

+ Add trigger

Description

Last modified

4 hours ago

Function ARN

arn:aws:lambda:us-east-1:755097618049:function:CleanupTasksScheduler

Function URL Info

Code | Test | Monitor | Configuration | Aliases | Versions



Lambda > Functions > CleanupTasksScheduler

⋮ ⓘ ⓘ

Code properties Info

Package size
596 byte

SHA256 hash
 iAihpuivkCZlwZaWBae7jzes2oUCOSdpCsg+qc1aWOQ=

Last modified
4 hours ago

► Encryption with AWS KMS customer managed KMS key Info

Runtime settings Info

Runtime
Python 3.14

Handler Info
lambda_function.lambda_handler

Architecture Info
x86_64

Edit

Edit runtime management configuration

Layers Info

Merge order	Name	Layer version	Compatible runtimes	Compatible architectures	Version ARN
1	common-dependencies	1	python3.14	x86_64	arn:aws:lambda:us-east-1:755097618049:layer:common-dependencies:1

Edit

Add a layer

The screenshot shows two views of the AWS Lambda Functions configuration page for the 'CleanupTasksScheduler' function.

Top View (Triggers):

- General configuration:** Triggers, Permissions, Destinations, Function URL, Environment variables, Tags, VPC, RDS databases, Monitoring and operations tools.
- Configuration:** Triggers (1/1) Info
- Trigger:** EventBridge (CloudWatch Events): DailyCleanupSchedule
 - arn:aws:events:us-east-1:755097618049:rule/DailyCleanupSchedule
 - Rule state: ENABLED
 - Event bus: default
 - isComplexStatement: No
 - Schedule expression: cron(0 0 * ? *)
 - Service principal: events.amazonaws.com
 - Statement ID: lambda-d6e8c3c6-c3dc-4e7a-96a3-e3eb7218795b

Bottom View (VPC):

- General configuration:** Triggers, Permissions, Destinations, Function URL, Environment variables, Tags, VPC, RDS databases, Monitoring and operations tools.
- Configuration:** VPC Info
- VPC:** vpc-05f3b0ec6c735d13c (10.0.0.0/16) | learning-platform-vpc
 - Subnets:**
 - Allow IPv6 traffic = false
 - subnet-009f793d729e3ef6 (10.0.21.0/24) | us-east-1b, data-subnet-1b
 - subnet-07c44a5aca6162980 (10.0.20.0/24) | us-east-1a, data-subnet-1a
 - Security groups:**
 - sg-0866a5c882ac1ff97 (container-hosts-sg-dev)
- Security group rules (7):**

Inbound rules	Protocol	Ports	Source
sg-0866a5c882ac1ff97	Custom TCP	5432	sg-0849158143a6fde93
sg-0866a5c882ac1ff97	Custom TCP	8000	sg-021bd251e998c9f94
sg-0866a5c882ac1ff97	Custom TCP	5000	sg-021bd251e998c9f94

3. Kafka Monitor Lambda

- Function Identity's Purpose:**

This is our KafkaMonitor Lambda function, which fulfills the project requirement to Create Lambda for Kafka consumer monitoring.

Its main purpose is to continuously monitor the health, lag, and throughput of our microservices' consumers (like the Quiz Service and Chat Service) that are reading from the Kafka topics.

- Trigger Mechanism:**

We schedule this to run frequently—for example, every 5 minutes—to ensure timely detection of any processing slowdowns in our event stream.

- **Security's Permissions (if the IAM was working):**

The IAM role for this function is granted read-only access to the Kafka cluster's metrics (**via CloudWatch or the Kafka management service**) and permissions to send alerts and potentially trigger scaling actions, all strictly scoped for security.

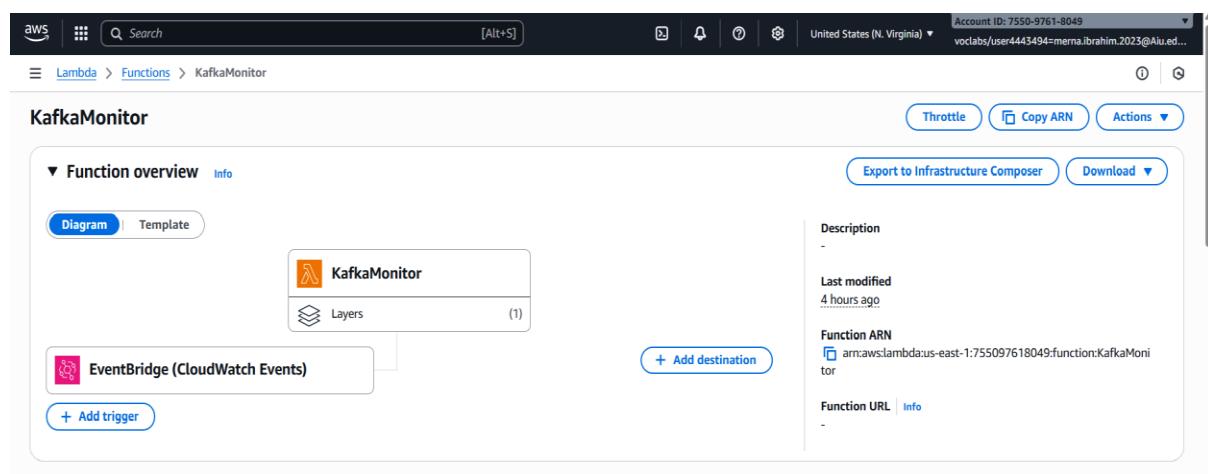
- **Code Function's Project Requirement:**

The Python code implements our KafkaMonitor Lambda function. It directly fulfills the project requirement to Create Lambda for Kafka consumer monitoring."

The function is triggered by EventBridge (CloudWatch Events) on a short schedule (e.g., every 5 minutes) to continuously assess the health of our event-driven system.

- **Configuration Details:**

The function uses environment variables (**KAFKA_BOOTSTRAP, TOPICS, CLOUDWATCH_NAMESPACE**) for flexible configuration, meaning we can reuse this single Lambda to monitor multiple topics just by changing environment settings in our IaC templates.



The screenshot shows the AWS Lambda Function Overview page for the function "KafkaMonitor". The function was last modified 4 hours ago. It has a Function ARN: arn:aws:lambda:us-east-1:755097618049:function:KafkaMonitor. A "Function URL" is also present. The function has one layer named "Layers" (1). It is triggered by "EventBridge (CloudWatch Events)". Buttons for "Throttle", "Copy ARN", and "Actions" are visible.

Code source (Info)

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences Account ID: 7550-9761-8049 vclabs/user4443494=merna.ibrahim.2023@AiU.edu...

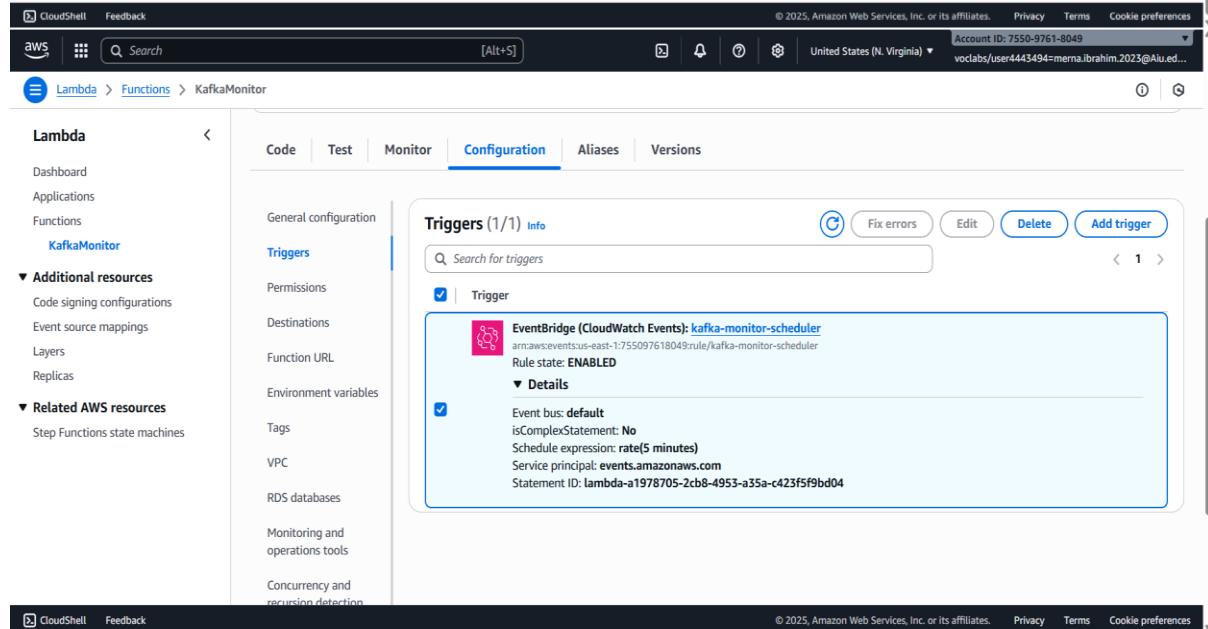
The code source section shows the SHA256 hash of the package: 360iphGNgnuTJAhwGz0eAm1bHGjawDFCm3baZe+Alko=. Last modified 4 hours ago. It uses AWS KMS customer managed KMS key for encryption.

Runtime settings (Info)

Runtime Python 3.14 Handler lambda_function.lambda_handler Architecture x86_64

Layers (Info)

Merge order	Name	Layer version	Compatible runtimes	Compatible architectures	Version ARN
1	common-dependencies	1	python3.14	x86_64	arn:aws:lambda:us-east-1:755097618049:layer:common-dependencies:1



The configuration page shows the "Configuration" tab selected. Under "Triggers", there is one trigger named "EventBridge (CloudWatch Events): kafka-monitor-scheduler" which is enabled. The trigger details show it uses the default event bus, has no complex statements, and a schedule expression of "rate(5 minutes)". The service principal is events.amazonaws.com and the statement ID is lambda-a1978705-2cb8-4953-a35a-c423f5f9bd04.

4. AutoScaling Lambda:

- **Function Identity's Purpose:**

This is our final Lambda function for the infrastructure layer: the AutoScaling function. It fulfills the project requirement to Implement Lambda for automated scaling decisions.

Its primary purpose is to provide custom, proactive scaling logic that goes beyond the standard CloudWatch CPU or Network metrics.

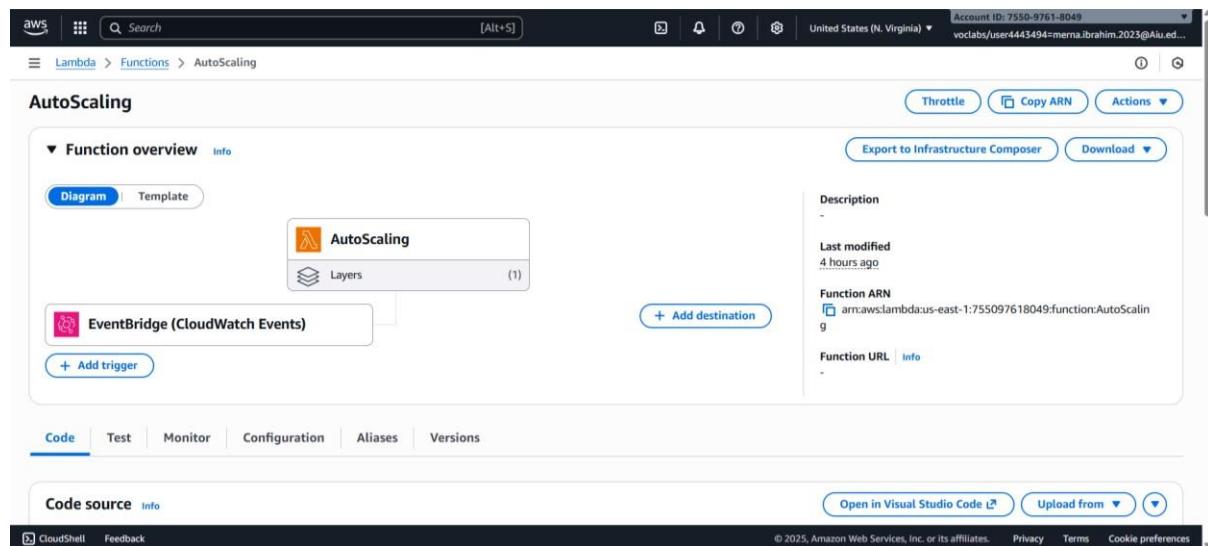
- **Trigger Mechanism:**

However, this function is designed to be triggered specifically by a CloudWatch Alarm. For instance, when the TopicLag metric (generated by our KafkaMonitor Lambda) exceeds a critical threshold, the alarm triggers this AutoScaling function.

- **Code Function's Project Requirement:**

This Python code implements a custom scaling solution that meets the requirement for automated scaling decisions.

Unlike native Auto Scaling, this Lambda allows us to implement complex, multi-metric logic or react to custom metrics like the Kafka Lag we discussed earlier.



The screenshots show the configuration for an AutoScaling Lambda function. The top screenshot shows the 'Code properties' section with a package size of 857 bytes, SHA256 hash, and last modified 4 hours ago. It also shows runtime settings for Python 3.14 and Handler lambda_function.lambda_handler, with architecture x86_64. The middle screenshot shows the 'Configuration' tab selected, displaying triggers for CloudWatch Events (auto-scaling). The bottom screenshot shows the VPC configuration, including subnets (allow IPv6 traffic = false), security groups (SG-RDS, SG-Kafka, ec2-sg, SG-containers), and security group rules (Inbound rules for Custom TCP ports 5432, 80, 8000).

Code properties

- Package size: 857 byte
- SHA256 hash: tGRCQ22OvY3Cm1sEgK+LsNEx8NZWxRa25V0vg0cFzns=
- Last modified: 4 hours ago

Runtime settings

- Runtime: Python 3.14
- Handler: lambda_function.lambda_handler
- Architecture: x86_64

Layers

Merge order	Name	Layer version	Compatible runtimes	Compatible architectures	Version ARN
1	common-dependencies	1	python3.14	x86_64	arn:aws:lambda:us-east-1:755097618049:layer:common-dependencies:1

Configuration

Triggers (1/1)

- Trigger (EventBridge (CloudWatch Events): auto-scaling)
- Rule state: ENABLED
- Event bus: default
- isComplexStatement: No
- Schedule expression: rate(5 minutes)
- Service principal: events.amazonaws.com
- Statement ID: lambda-f018ef04-3099-4f3b-b82f-39bb5201e146

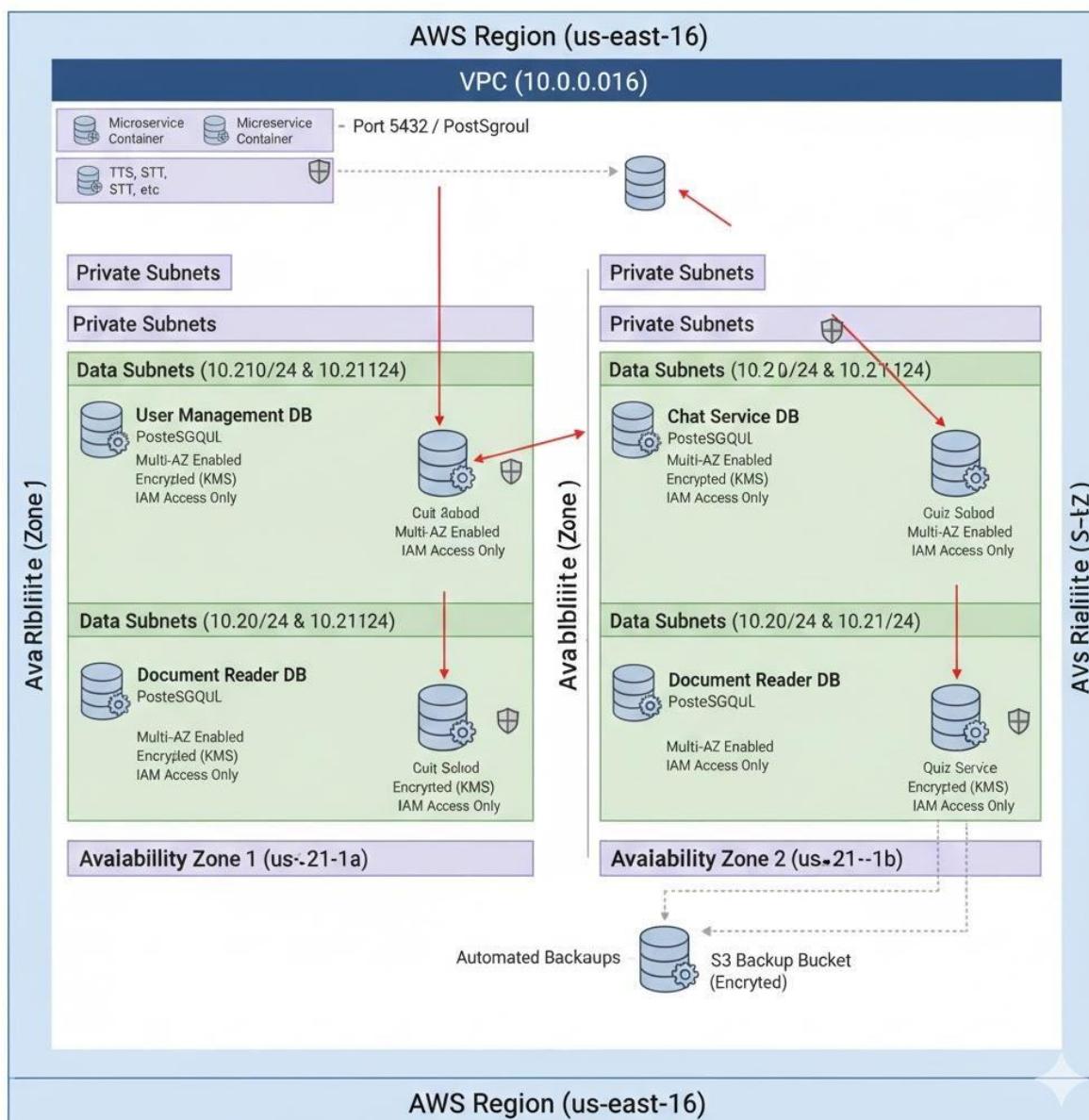
VPC

- VPC: vpc-05f3b0ec6c735d13c (10.0.0.0/16) | learning-platform-vpc
- Subnets:
 - Allow IPv6 traffic = false
 - subnet-0d680853d07e59d9c (10.0.11.0/24) | us-east-1b, private-subnet-1b
 - subnet-081bc6179e84d6f7 (10.0.10.0/24) | us-east-1a, private-subnet-1a
- Security groups:
 - sg-0849158143a6fdde93 (SG-RDS)
 - sg-0898c0df62dd2db02 (SG-Kafka)
 - sg-0e82a99b873e5ab51 (ec2-sg)
 - sg-021bd251e998c9ff4 (SG-containers)

Security group rules (8)

Inbound rules	Security group ID	Protocol	Ports	Source
	sg-0849158143a6fdde93	Custom TCP	5432	sg-021bd251e998c9ff4
	sg-021bd251e998c9ff4	Custom TCP	80	sg-0ff6c301faa11bee9
	sg-031bd251e998c9ff4	Custom TCP	8000	sg-021bd251e998c9ff4

8. Relational Database Service (RDS)



RDS Instance Deployed.

This screenshot confirms the successful provisioning of the **user-mgmt-db** instance. It is **Available** and uses the **PostgreSQL** engine, meeting the project's requirement for a dedicated database per service (**Isolation**).

Databases (1)		Group resources		Modify	Actions	Create database	
DB identifier		Status	Role	Engine	Upgrade rollout order	Region ...	Size
user-mgmt-db		Available	Instance	PostgreSQL	SECOND	us-east-1b	db.t3.medium

DB Credentials Management.

This screenshot shows the "**Modify DB instance**" page, specifically the **Credentials management** section for the user-mgmt-db. It confirms the choice of **Self managed** credentials over integrating with AWS Secrets Manager. The **Master password** field is visible, where the developer sets a custom, secure password for the database. This step is critical for **security setup** and managing access credentials.

Modify DB instance: user-mgmt-db

The screenshot shows the "Modify DB instance" page for the database "user-mgmt-db". In the "Credentials management" section, the "Self managed" option is selected, indicated by a blue radio button. The "Master password" field contains a partially obscured password. Other options like "Managed in AWS Secrets Manager - most secure" and "Auto generate password" are also shown but not selected.

Instance Sizing Confirmed.

This image confirms the **DB instance class** selection within the configuration settings. The chosen class is **db.t3.medium**, categorized under **Burstable classes**. This selection adheres to the project's **minimum requirement** specification, providing **2 vCPUs** and **4 GB RAM** for the database instance.

Instance configuration

The screenshot shows the "Instance configuration" section of the RDS instance setup. Under "DB instance class", the "db.t3.medium" option is selected, highlighted with a blue radio button. Below it, the specifications "2 vCPUs", "4 GiB RAM", "EBS Bandwidth: Up to 2,085 Mbps", and "Network: Up to 5 Gbps" are listed.

Storage Type Specified.

This screenshot confirms the **Storage configuration** for the RDS instance. The **Storage type** is set to **General Purpose SSD (gp3)**, offering high performance and independent scaling. The **Allocated storage** is set to **20 GiB**, and the default **Provisioned IOPS (3000)** and **Storage throughput (125 MiBps)** are utilized. This adheres to using modern, performant SSD storage.

Storage

Storage type [Info](#)
Provisioned IOPS SSD (io2) storage volumes are now available.

General Purpose SSD (gp3)
Performance scales independently from storage

Allocated storage [Info](#)
20 GiB

Minimum: 20 GiB. Maximum: 32,768 GiB

Provisioned IOPS [Info](#)
3000 IOPS

Baseline IOPS of 3,000 IOPS is included for allocated storage less than 400 GiB.

Storage throughput [Info](#)
125 MiBps

Baseline storage throughput of 125 MiBps is included for allocated storage less than 400 GiB.

To provision additional IOPS and throughput, increase the allocated storage to 400 GiB or greater.

Storage Autoscaling Enabled.

This image shows the **Additional storage configuration** where **Storage autoscaling** is **Enabled**. This feature allows the database storage to dynamically increase when needed, up to the set **Maximum storage threshold** of **1000 GiB**. This ensures the database can handle growth without manual intervention.

▼ Additional storage configuration

Storage autoscaling [Info](#)
Provides dynamic scaling support for your database's storage based on your application's needs.

Enable storage autoscaling
Enabling this feature will allow the storage to increase after the specified threshold is exceeded.

Maximum storage threshold [Info](#)
Charges will apply when your database autoscales to the specified threshold

1000 GiB

Allocated storage value must be 22 GiB to 32,768 GiB

Connectivity Settings Chosen.

This screenshot displays the **Connectivity** configuration. It shows the **IPv4** network type selected and highlights the assignment of a **DB subnet group** (default VPC subnet group is used here). Critically, a **Security group** named rds-postgres-sg has been attached, which will control network access to the database instance. **Multi-AZ deployment** is currently set to **Do not create a standby instance**, but the project requires this to be enabled for production.

Availability & durability

Multi-AZ deployment [Info](#)

- Create a standby instance (recommended for production usage)
Creates a standby in a different Availability Zone (AZ) to provide data redundancy, eliminate I/O freezes, and minimize latency spikes during system backups.
- Do not create a standby instance

Connectivity

Network type [Info](#)
To use dual-stack mode, make sure that you associate an IPv6 CIDR block with a subnet in the VPC you specify.

IPv4
Your resources can communicate only over the IPv4 addressing protocol.

Dual-stack mode
Your resources can communicate over IPv4, IPv6, or both.

DB subnet group
default-vpc-05f3b0ec6c735d13c

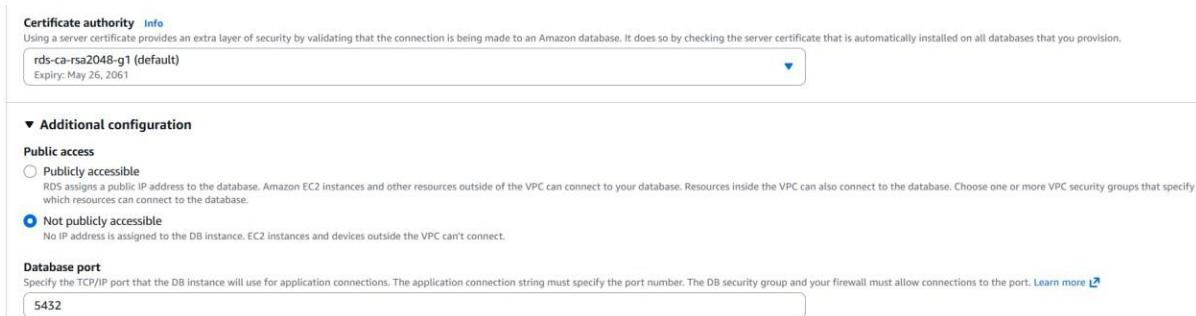
Security group
List of DB security groups to associate with this DB instance.

Choose security groups

rds-postgres-sg

Security and Network Access.

This image focuses on the **Additional configuration** for network security. It confirms that **Public access** is set to **Not publicly accessible**, ensuring the database is isolated within the VPC. The default **Database port** is correctly set to **5432 (PostgreSQL)**, and a **Certificate authority** is utilized for secure, validated connections (TLS/SSL).



Certificate authority [Info](#)
Using a server certificate provides an extra layer of security by validating that the connection is being made to an Amazon database. It does so by checking the server certificate that is automatically installed on all databases that you provision.

rds-ca-rsa2048-g1 (default)
Expiry: May 26, 2061

▼ Additional configuration

Public access

Publicly accessible
RDS assigns a public IP address to the database. Amazon EC2 instances and other resources outside of the VPC can connect to your database. Resources inside the VPC can also connect to the database. Choose one or more VPC security groups that specify which resources can connect to the database.

Not publicly accessible
No IP address is assigned to the DB instance. EC2 instances and devices outside the VPC can't connect.

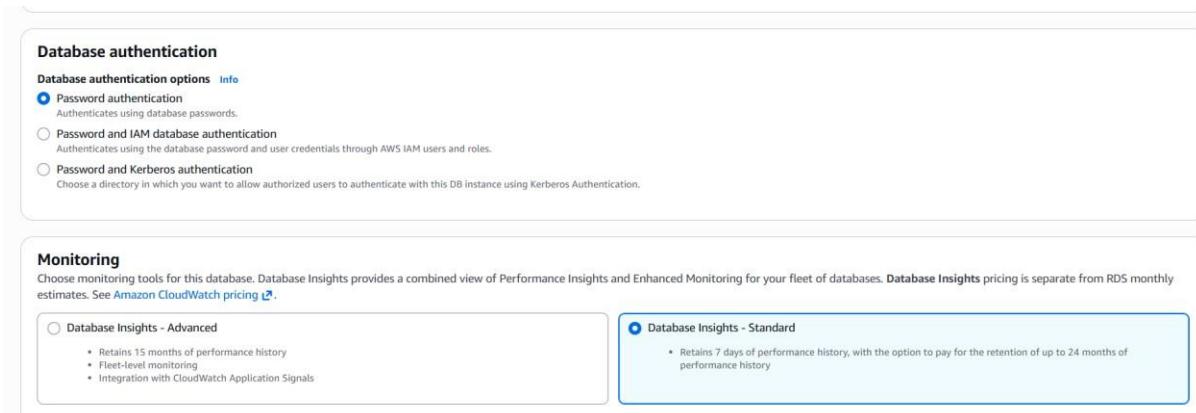
Database port
Specify the TCP/IP port that the DB instance will use for application connections. The application connection string must specify the port number. The DB security group and your firewall must allow connections to the port. [Learn more](#)

5432

Authentication and Monitoring Chosen.

This image confirms two key configurations:

- 1. Database Authentication:** **Password authentication** is selected, meaning users authenticate using traditional database passwords (managed manually or via Secrets Manager).
- 2. Monitoring:** **Database Insights - Standard** is chosen. This option retains **7 days of performance history**, providing necessary metrics for basic monitoring and troubleshooting.



Database authentication

Database authentication options [Info](#)

Password authentication
Authenticates using database passwords.

Password and IAM database authentication
Authenticates using the database password and user credentials through AWS IAM users and roles.

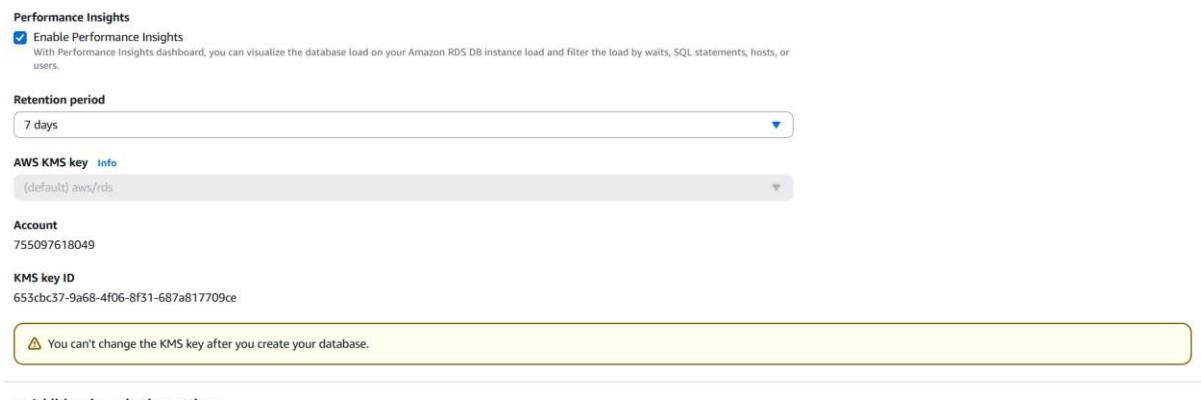
Password and Kerberos authentication
Choose a directory in which you want to allow authorized users to authenticate with this DB instance using Kerberos Authentication.

Monitoring
Choose monitoring tools for this database. Database Insights provides a combined view of Performance Insights and Enhanced Monitoring for your fleet of databases. **Database Insights** pricing is separate from RDS monthly estimates. See [Amazon CloudWatch pricing](#).

<input type="radio"/> Database Insights - Advanced <ul style="list-style-type: none"> Retains 15 months of performance history Fleet-level monitoring Integration with CloudWatch Application Signals 	<input checked="" type="radio"/> Database Insights - Standard <ul style="list-style-type: none"> Retains 7 days of performance history, with the option to pay for the retention of up to 24 months of performance history
--	---

Performance Insights Enabled.

This screenshot confirms the configuration of **Performance Insights**, which is **Enabled** to visualize database load. The **Retention period** is set to **7 Days**. Importantly, it shows the database is configured to use an **AWS KMS key ((default) aws/rds)**, ensuring that the database is **encrypted at rest** as required by security best practices.



Performance Insights

Enable Performance Insights
With Performance Insights dashboard, you can visualize the database load on your Amazon RDS DB instance load and filter the load by waits, SQL statements, hosts, or users.

Retention period
7 days

AWS KMS key [Info](#)
(default) aws/rds

Account
755097618049

KMS key ID
653cbc37-9a68-4f06-8f31-687a817709ce

⚠ You can't change the KMS key after you create your database.

▼ Additional monitoring settings
Enhanced Monitoring, CloudWatch Logs and DevOps Guru

Additional Monitoring Settings.

This image displays the **Additional monitoring settings** for the RDS instance. It shows that **Enhanced Monitoring** is currently **disabled**. Below this, it provides options for **Log exports** (IAM, PostgreSQL, Upgrade) to Amazon CloudWatch Logs, none of which are currently selected. This section documents the default or chosen settings for advanced observability and logging



▼ Additional monitoring settings
Enhanced Monitoring, CloudWatch Logs and DevOps Guru

Enhanced Monitoring
 Enable Enhanced monitoring
Enabling Enhanced Monitoring metrics are useful when you want to see how different processes or threads use the CPU.

Log exports
Select the log types to publish to Amazon CloudWatch Logs

iam-db-auth-error log
 PostgreSQL log
 Upgrade log

IAM role
The following service-linked role is used for publishing logs to CloudWatch Logs.

RDS service-linked role

Backup and Retention Settings.

This image shows the **Backup configuration** under **Additional configuration**. It specifies that the **automated backup** is **not yet enabled** (checkbox is unchecked), though the **Backup retention period** is correctly set to **7 days** (meeting the project's minimum requirement). A custom **Backup window** is defined to start at **10:15 UTC** for a duration of **0.5 hours**. To comply with the project's requirements, automated backup **must be enabled** here.

▼ Additional configuration

Database options, backup turned on, maintenance, delete protection turned off

Database options

DB parameter group [Info](#)

default.postgres16



Backup

Enable automated backup

Creates a point-in-time snapshot of your database.

Backup retention period [Info](#)

The number of days (1-35) for which automatic backups are kept.

7



days

Backup window [Info](#)

The daily time range (in UTC) during which RDS takes automated backups.

Choose a window

No preference

Start time

10



:

15



UTC

Duration

0.5



hours

Copy tags to snapshots

Maintenance and Protection Settings.

This final configuration image shows the **Maintenance** options. **Auto minor version upgrade** is **Enabled**, ensuring the database automatically receives small patches. The **DB instance maintenance window** is defined for **Wednesday at 06:49 UTC** for a **0.5-hour** duration. Finally, **Delete protection** is currently **disabled** (checkbox unchecked), meaning the database can be accidentally or intentionally deleted without an extra confirmation layer.

Backup replication [Info](#)

Enable replication in another AWS Region

Enabling replication automatically creates backups of your DB instance in the selected Region, for disaster recovery, in addition to the current Region.

Maintenance

Auto minor version upgrade [Info](#)

Enable auto minor version upgrade

Enabling auto minor version upgrade will automatically upgrade your database minor version. For limitations and more details, see Automatically upgrading the minor engine version [documentation](#).

DB instance maintenance window

The weekly time range during which system maintenance can occur.

Start day

Wednesday

Start time

06



49



UTC

Duration

0.5



hours

Enable deletion protection

Protects the database from being deleted accidentally. While this option is enabled, you can't delete the database.

DB Subnet Group Verification.

This final screenshot confirms the existence of a **DB subnet group** named default-vpc-05f3b0ec6c735d13c with a status of **Complete**. A DB Subnet Group is essential for RDS to launch instances (especially Multi-AZ deployments) within your specified **VPC's private subnets** (Data Subnets, as per your project's architecture). This verifies the successful networking preparation needed for launching highly available databases.

The screenshot shows the AWS RDS Subnet Groups page. On the left, there's a sidebar with 'Aurora and RDS' selected. The main area is titled 'Subnet groups (1)' and shows a table with one row. The row contains the name 'default-vpc-05f3b0ec6c735d13c', a description 'Created from the RDS Management Console', a status 'Complete', and a VPC ID 'vpc-05f3b0ec6c735d13c'. There are buttons for 'Edit' and 'Delete' at the top right, and a 'Create DB subnet group' button.

Subnet Group Details.

This screenshot shows the detailed information for the **DB Subnet Group**. The **Subnet group name** is displayed, confirming its association with the specified **VPC ID (vpc-05f3b0ec6c735d13c)**. This group is essential for launching the RDS instance into the correct **private data subnets** within your VPC, a foundational requirement for securing the database and enabling **Multi-AZ deployments**.

The screenshot shows the 'Subnet group details' page for the 'default-vpc-05f3b0ec6c735d13c' subnet group. It includes sections for 'Subnet group name' (default-vpc-05f3b0ec6c735d13c), 'VPC ID' (vpc-05f3b0ec6c735d13c), 'ARN' (arn:aws:rds:us-east-1:755097618049:subgrp:default-vpc-05f3b0ec6c735d13c), 'Description' (Created from the RDS Management Console), and a note about ARN format: 'arn:aws:rds:us-east-1:755097618049:subgrp:default-vpc-05f3b0ec6c735d13c e.g.:arn:aws:kms:<region><accountID>/key/<key-id>'.

Subnets Summary in DB Group.

This final screenshot provides a detailed table summarizing the 8 subnets selected for the **DB Subnet Group**. This is crucial documentation for the Configurations and Results sections, as it verifies the entire network architecture for the RDS deployment.

It shows that all four required subnet types (**kafka**, **private**, **data**, **public**) are correctly included across two Availability Zones (**us-east-1a** and **us-east-1b**), along with their associated **CIDR blocks**. This confirms that the RDS instance can be launched into the **private data subnets (10.0.20.0/24 and 10.0.21.0/24)** and supports the **Multi-AZ requirement** for high availability.

Subnets selected (8)

Availability zone	Subnet name	Subnet ID	CIDR block
us-east-1a	kafka-subnet-1a	subnet-0a1f2b9c9c0e0f71c	10.0.30.0/24
us-east-1b	private-subnet-1b	subnet-0d680853d07e59d9c	10.0.11.0/24
us-east-1a	data-subnet-1a	subnet-07c44a5aca6162980	10.0.20.0/24
us-east-1a	private-subnet-1a	subnet-081bbc6179e84d6f7	10.0.10.0/24
us-east-1a	public-subnet-1a	subnet-0e9ae6a972070b954	10.0.1.0/24
us-east-1b	data-subnet-1b	subnet-009f793d72f9e3ef6	10.0.21.0/24
us-east-1b	kafka-subnet-1b	subnet-0414bb6881a731877	10.0.31.0/24
us-east-1b	public-subnet-1b	subnet-0f27d0f340b3faebd	10.0.2.0/24

Subnets Selected for DB Group.

This screenshot shows the "**Add subnets**" configuration step for the DB Subnet Group. It confirms that subnets across **two Availability Zones (us-east-1a and us-east-1b)** have been selected. Specifically, the list includes all four required subnet types, categorized by AZ and CIDR blocks:

- **Kafka:** 10.0.30.0/24 and 10.0.31.0/24
- **Data:** 10.0.20.0/24 and 10.0.21.0/24
- **Private:** 10.0.10.0/24 and 10.0.11.0/24
- **Public:** 10.0.1.0/24 and 10.0.2.0/24

This step successfully collects the necessary subnets within the VPC to allow RDS to launch the database instance in the highly available **Data Subnets** across multiple AZs.

Add subnets

Availability Zones
Choose the Availability Zones that include the subnets you want to add.

Choose an availability zone ▾

us-east-1a ✖ us-east-1b ✖

Subnets
Choose the subnets that you want to add. The list includes the subnets in the selected Availability Zones.

Select subnets ▾

kafka-subnet-1a Subnet ID: subnet-0a1f2b9c9c0e0f71c	private-subnet-1b Subnet ID: subnet-0d680853d07e59d9c
data-subnet-1a Subnet ID: subnet-07c44a5aca6162980	private-subnet-1a Subnet ID: subnet-081bbc6179e84d6f7
public-subnet-1a Subnet ID: subnet-0e9aa6a972070b954	data-subnet-1b Subnet ID: subnet-009f793d72f9e3ef6
kafka-subnet-1b Subnet ID: subnet-0414bb6881a731877	public-subnet-1b Subnet ID: subnet-0f27d0f540b3faebd

For Multi-AZ DB clusters, you must select 3 subnets in 3 different Availability Zones.

Phase 2

3. Apache Kafka

3.1 Kafka Cluster Architecture

The screenshot shows the AWS Amazon MSK Create cluster wizard at Step 2: Networking. The user has selected "Multi-VPC private connectivity". They have chosen a VPC (vpc-05d2c059fa6b57964) and created three zones: us-east-1a, us-east-1b, and us-east-1c. Each zone is associated with a specific subnet (subnet-0502400db014b8d..., subnet-09e49a2e465066b..., and subnet-042dd7cd6450299... respectively). The "Public access" section is set to "Off". Under "Security groups in Amazon EC2", the user has assigned the "kafka-sg" security group. Navigation buttons at the bottom right include "Cancel", "Previous", and "Next".

Networking settings for Amazon MSK cluster, including VPC, subnets across three zones, and security group assignment

Security

Access control methods Info
The method that you want Amazon MSK to use to authenticate clients and allow or deny actions. A cluster can have one or more methods selected. Your choice of access methods will determine the encryption options that you can select.

Unauthenticated access
No authentication is required for clients and all actions are allowed.

IAM role-based authentication
Use IAM to authenticate the identity of clients that connect to an MSK cluster. Use the IAM console to create and deploy IAM user or role-based policies.

IAM access control is now supported for Java and non-Java clients
IAM access control is supported for Java and non-Java clients. To review the client changes required to configure IAM authentication, review our documentation [documentation](#).

SASL/SCRAM authentication
SCRAM provides a username-and-password authentication method using the SASL framework. Amazon MSK uses AWS Secrets Manager to enable SASL/SCRAM. You can link secrets after you have created a cluster.

TLS client authentication through AWS Certificate Manager (ACM)
Use AWS Private Certificate Authority (CA) to authenticate the identity of clients that connect to an MSK cluster.

Encryption Info
Choose data encryption options that you want to use. Data management requires encryption.

Encrypt data in transit

TLS encryption enabled
When the access control method is IAM, SASL/SCRAM or TLS, clients must use TLS encryption to communicate with the brokers.

Between clients and brokers

TLS encryption
Required for IAM, SASL/SCRAM and TLS access control methods.

Plaintext
Plaintext traffic isn't possible with SASL/SCRAM or IAM access control methods.

Within the cluster

TLS encryption

Encrypt data at rest
Amazon MSK uses AWS KMS keys to encrypt your data at rest. You can use AWS Key Management Services (KMS) to create and manage CMKS.

Use AWS-managed key
The AWS-managed key (aws/kafka) is a KMS key in your account that is created, managed and used on your behalf by Amazon MSK.

Use customer-managed key
Customer-managed keys are KMS keys in your AWS account that you create, own and manage.

Cancel **Previous** **Next**

MSK cluster configured with private subnets in us-east-1a, 1b, and 1c zones.

Monitoring and tags

Monitoring

Amazon CloudWatch metrics for this cluster Info
Enhanced metrics are available at an additional cost.

Basic monitoring
Includes basic cluster-level and broker-level monitoring. Available free.

Enhanced broker-level monitoring
Also includes basic monitoring. Available at an additional cost.

Enhanced topic-level monitoring
Also includes basic and enhanced broker-level monitoring. Available at an additional cost.

Enhanced partition-level monitoring
Also includes basic, enhanced broker-level monitoring and topic-level monitoring. Available at an additional cost.

Open monitoring with Prometheus Info
Also includes basic and enhanced broker-level monitoring, topic-level monitoring and partition-level monitoring. Available at an additional cost.

Enable open monitoring with Prometheus
When you enable open monitoring with Prometheus, you can expose metrics using the JMX Exporter, the Node Exporter or both. These metrics include cluster-level, broker-level, topic-level and partition-level information. Open monitoring is available at no charge, but charges apply for the transfer of data across Availability Zones.

Broker log delivery Info
Broker logs enable you to troubleshoot your Apache Kafka applications and analyse communications with your MSK cluster. Amazon MSK doesn't charge for sending the logs. However, ingestion and storage charges apply based on the destination.

Deliver to Amazon CloudWatch Logs
Analyze, query and set alarms on the logs.

Log group
To create a new log group, visit [Amazon CloudWatch Logs console](#).
 arn:aws:log:us-east-1:234690396635:log-group:/aws/msk/kafka-dev-logs:
Use the format arn:aws:logs:[region]:[account-id]:log-group:[log-group-name].*

Deliver to Amazon S3
Store and retrieve raw logs in object storage.

Deliver to Amazon Data Firehose
Capture, transform, and deliver logs to Amazon OpenSearch Service or other Amazon Data Firehose destinations.

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

Key	Value - optional
<input type="text"/> kafka	<input type="text"/> true X Remove
<input type="text"/> Project	<input type="text"/> Cloud-Learning-Platform X Remove

Add new tag
You can add up to 48 more tags.

Cancel **Previous** **Next**

Basic CloudWatch monitoring enabled with tagged MSK cluster for project tracking.

The screenshot shows the AWS Amazon MSK Create Cluster wizard at Step 5: Review and create. The configuration details are as follows:

- Step 1: Cluster settings**
 - Cluster name:** my-kafka-cluster
 - Cluster type:** Provisioned
 - Apache Kafka version:** 3.8.x
 - Metadata mode:** ZooKeeper
- Brokers**
 - Storage:** EBS storage volume per broker: 1000 GiB; Provisioned storage throughput per broker: Not enabled; Cluster storage mode: EBS storage
 - Cluster configuration:** Cluster configuration: MSK default configuration
- Step 2: Networking**
 - Networking settings:** Number of zones: 3; Subnets: subnet-0502400b9014b8d1d, subnet-09e09a2e465066b1a, subnet-042d87cd6450299d9; Security groups applied: sg-0fc5574ddab0a959
- Step 3: Security**
 - Security settings:** Access control methods: Unauthenticated access (Not enabled), IAM role-based authentication (Enabled).
 - Configure IAM:** To use IAM access control, configure your client properties and provide IAM policies that allow or deny actions. Learn more.
 - Encryption:** Encrypt data in transit: TLS client authentication through AWS Certificate Manager (ACM) (Not enabled). Between clients and brokers: TLS encryption (Enabled), Plaintext (Not enabled).
 - Encrypt data at rest:** AWS KMS keys that Amazon MSK uses to encrypt your data at rest. You can use AWS Key Management Service (KMS) to create and manage CMKs. KMS key: (default) aws/kafka
- Step 4: Monitoring and tags**
 - Monitoring metrics:** Amazon CloudWatch metrics for this cluster: Monitoring level: Basic.
 - Open monitoring with Prometheus:** Open monitoring: Not enabled.
 - Broker log delivery:** Amazon CloudWatch logs: /aws/msk/kafka-dev-logs; Amazon S3: Not enabled; Amazon Data Firehose: Not enabled.
 - Cluster tags (2):**

Key	Value
kafka	true
Project	Cloud-Learning-Platform
- Time to create a provisioned cluster:** A typical provisioned cluster takes up to 15 minutes to create.

At the bottom right are buttons for **Cancel**, **Previous**, and **Create cluster**.

Final MSK cluster review: 3-zone setup, IAM security, TLS encryption, and basic monitoring.

The screenshot shows the AWS Amazon MSK Create Cluster wizard. The steps are:

- Step 1: Cluster settings**
 - Cluster name:** my-kafka-cluster
 - Cluster type:** Provisioned
 - Apache Kafka version:** 3.8.x
 - Metadata mode:** ZooKeeper
- Step 2: Networking**
 - Networking settings:** Number of zones: 3. Subnets: subnet-0502400db014bd1d, subnet-05e49a2e465066b1a, subnet-042dd7cd6450299d9. Security groups applied: sg_0fc3574ddab0a059.
- Step 3: Security**
 - Security settings:**
 - Access control methods:** Unauthenticated access: Not enabled; IAM role-based authentication: Enabled.
 - Configure IAM:** To use IAM access control, configure your client properties and provide IAM policies that allow or deny actions. Learn more.
 - SASL/SCRAM authentication:** Not enabled.
 - TLS client authentication through AWS Certificate Manager (ACM):** Not enabled.
 - Encryption:**
 - Encrypt data in transit:** Within the cluster: TLS encryption: Enabled. Between clients and brokers: TLS encryption: Enabled; Plaintext: Not enabled.
 - Encrypt data at rest:** AWS KMS keys that Amazon MSK uses to encrypt your data at rest. You can use AWS Key Management Service (KMS) to create and manage CMKs. KMS key (default): aws/kafka.
- Step 4: Monitoring and tags**
 - Monitoring metrics:** Amazon CloudWatch metrics for this cluster: Monitoring level: Basic.
 - Open monitoring with Prometheus:** Open monitoring: Not enabled.
 - Broker log delivery:**
 - Amazon CloudWatch logs: /aws/msk/kafka-dev-logs.
 - Amazon S3: Not enabled.
 - Amazon Data Firehose: Not enabled.
 - Cluster tags (2):**

Key	Value
kafka	true
Project	Cloud-Learning-Platform
- Step 5: Review and create**
 - Time to create a provisioned cluster:** A typical provisioned cluster takes up to 15 minutes to create.
 - Buttons:** Cancel, Previous, Create cluster.

Final MSK cluster review: 3-zone setup, IAM security, TLS encryption, and basic monitoring.

(CSE353) Cloud Computing

The screenshot shows the AWS Amazon MSK Create Cluster wizard. The process is divided into five steps:

- Step 1: Cluster settings**
 - Cluster name:** my-kafka-cluster
 - Cluster type:** Provisioned
 - Apache Kafka version:** Apache Kafka version 5.0.0
 - Metadata mode:** ZooKeeper
- Step 2: Networking**
 - Networking settings:** Number of zones: 3, Subnets: subnet-05024000, subnet-042d07c7, subnet-050e59a2, Security groups applied: sg-0fb5574cdebae659
- Step 3: Security**
 - Access control methods:** Unauthenticated access: Not enabled, IAM role-based authentication: Enabled
 - Configure IAM:** A callout box for IAM access control.
 - SASL/SCRAM authentication:** Not enabled
 - TLS client authentication through AWS Certificate Manager (ACM):** Not enabled
 - Encryption:**
 - Encrypt data in transit:** Between clients and brokers: TLS encryption Enabled, Plaintext: Not enabled
 - Encrypt data at rest:** KMS key: (default) vmm/kafka
- Step 4: Monitoring and tags**
 - Monitoring metrics:** Amazon CloudWatch metrics for this cluster: Monitoring level: Basic, Open monitoring with Prometheus: Open monitoring Not enabled
 - Broker log delivery:** Amazon CloudWatch logs: /aws/elastic/kafka/cluster-logs
 - Amazon S3:** Not enabled
 - Amazon Data Firehose:** Not enabled
- Cluster tags (2):**

Key	Value
kafka	true
Project	Cloud-Learning-Platform

Time to create a provisioned cluster: A typical provisioned cluster takes up to 15 minutes to create.

Final MSK cluster review: 3-zone setup, IAM security, TLS encryption, and basic monitoring.

Review and create

Step 1: Cluster settings

Cluster name
Name: my-kafka-cluster

Cluster type
Cluster type: Provisioned

Apache Kafka version
Apache Kafka version: 3.8.x

Metadata mode
ZooKeeper

Brokers

Storage
EBS storage volume per broker: 1000 GiB | Provisioned storage throughput per broker: Not enabled | Cluster storage mode: EBS storage

Cluster configuration
Cluster configuration: MSK default configuration

Step 2: Networking

Networking settings
Number of zones: 3 | Subnets: subnet-05024000, subnet-09e49a2e, subnet-042dd7cd | Security groups applied: sg-0fc83f74d8ab0019

Step 3: Security

Security settings
Access control methods:
Unauthenticated access: Not enabled
IAM role-based authentication: Enabled

Configure IAM
To use IAM access control, configure your client properties and provide IAM policies that allow or deny actions. [Learn more](#)

SASL/SCRAM authentication: Not enabled
TLS client authentication through AWS Certificate Manager (ACM): Not enabled

Encryption
Encrypt data in transit: Encrypt options for data in transit within cluster and between clients and cluster.
Within the cluster: TLS encryption: Enabled
Between clients and brokers: TLS encryption: Enabled, Plaintext: Not enabled

Encrypt data at rest: AWS KMS keys that Amazon MSK uses to encrypt your data at rest. You can use AWS Key Management Service (KMS) to create and manage CMKs.
KMS key: (default) aws/kafka

Step 4: Monitoring and tags

Monitoring metrics
Amazon CloudWatch metrics for this cluster: Basic
Open monitoring with Prometheus: Open monitoring: Not enabled

Broker log delivery
Amazon CloudWatch logs: /aws/msk/kafka-dev-logs
Amazon S3: Not enabled
Amazon Data Firehose: Not enabled

Cluster tags (2)

Key	Value
kafka	true
Project	Cloud-Learning-Platform

Time to create a provisioned cluster
A typical provisioned cluster takes up to 15 minutes to create.

Create cluster

Final MSK cluster review: 3-zone setup, IAM security, TLS encryption, and basic monitoring.

```
sh-5.2$ CLASSPATH=/home/ssm-user/aws-msk-iam-auth-1.1.9-all.jar \
/home/ssm-user/kafka_2.13-3.5.1/bin/kafka-topics.sh \
--list \
--bootstrap-server b-1.mykafkacluster.x5xpry.c21.kafka.us-east-1.amazonaws.com:9098 \
--command-config /home/ssm-user/client.properties
__amazon_msk_canary
__consumer_offsets
audio.generation.completed
audio.generation.requested
audio.transcription.completed
audio.transcription.requested
chat.message
document.processed
document.uploaded
notes.generated
quiz.generated
quiz.requested
sh-5.2$
```

Kafka topics via MSK cluster using IAM and config file reference

Session ID: root-c5u5u5fhf35cfqzhodpric4y84 Shortcuts Instance ID: i-05dc18abf3d2df716 Terminate

```
sh-5.2$ JAR="/home/ssm-user/aws-msk-iam-auth-1.1.9-all.jar"
MSK_BIN="/home/ssm-user/kafka_2.13-3.5.1/bin"
BOOT="b-1.mykafkacluster.x5xpry.c21.kafka.us-east-1.amazonaws.com:9098"
CFG="/home/ssm-user/client.properties"

echo "--- All Topics Details ---"
for topic in document.uploaded document.processed notes.generated quiz.requested quiz.generated audio.transcription.completed audio.transcription.requested audio.generation.completed audio.generation.requested; do
    echo ""
    echo "--- Topic: $topic ---"
    CLASSPATH="$JAR" "$MSK_BIN/kafka-topics.sh" --describe --topic "$topic" --bootstrap-server "$BOOT" --command-config "$CFG"
done
--- All Topics Details ---

--- Topic: document.uploaded ---
TopicId: K_froIX8Rb-oJHmQUpjLfQ PartitionCount: 3      ReplicationFactor: 2      Configs: min.insync.replicas=2,message.format.version=3.0-IV1
,unclean.leader.election.enable=true
    Topic: document.uploaded      Partition: 0      Leader: 3      Replicas: 3,2      Isr: 3,2
    Topic: document.uploaded      Partition: 1      Leader: 1      Replicas: 1,3      Isr: 1,3
    Topic: document.uploaded      Partition: 2      Leader: 2      Replicas: 2,1      Isr: 2,1

--- Topic: document.processed ---
TopicId: gJaTcjrSsy9V_HIoT5jew PartitionCount: 3      ReplicationFactor: 2      Configs: min.insync.replicas=2,message.format.version=3.0-IV1
,unclean.leader.election.enable=true
    Topic: document.processed      Partition: 0      Leader: 1      Replicas: 1,2      Isr: 1,2
    Topic: document.processed      Partition: 1      Leader: 2      Replicas: 2,3      Isr: 2,3
    Topic: document.processed      Partition: 2      Leader: 3      Replicas: 3,1      Isr: 3,1

--- Topic: notes.generated ---
Topic: notes.generated TopicId: pxtJ0vdk9CwMfI2lnioJg PartitionCount: 2      ReplicationFactor: 2      Configs: min.insync.replicas=2,message.format.version=3.0-IV1,unclean
.leader.election.enable=true
    Topic: notes.generated      Partition: 0      Leader: 1      Replicas: 1,2      Isr: 1,2
    Topic: notes.generated      Partition: 1      Leader: 2      Replicas: 2,3      Isr: 2,3

--- Topic: quiz.requested ---
Topic: quiz.requested TopicId: 2couly6nS80Ryt14ohlZvQ PartitionCount: 2      ReplicationFactor: 2      Configs: min.insync.replicas=2,message.format.version=3.0-IV1,unclean
```

Described Kafka topics with partition count, replication factor, and in-sync replica configuration

Session ID: root-c5u5u5fhf35cfqzhodpric4y84 Shortcuts Instance ID: i-05dc18abf3d2df716 Terminate

```
Topic: quiz.generated      Partition: 0      Leader: 3      Replicas: 3,1      Isr: 3,1
Topic: quiz.generated      Partition: 1      Leader: 1      Replicas: 1,2      Isr: 1,2

--- Topic: audio.transcription.requested ---
Topic: audio.transcription.requested TopicId: VkyAIIe2RA2rTIK4-DQffA PartitionCount: 2      ReplicationFactor: 2      Configs: min.insync.replicas=2,message.format.version
=3.0-IV1,unclean.leader.election.enable=true
    Topic: audio.transcription.requested      Partition: 0      Leader: 1      Replicas: 1,3      Isr: 1,3
    Topic: audio.transcription.requested      Partition: 1      Leader: 2      Replicas: 2,1      Isr: 2,1

--- Topic: audio.transcription.completed ---
Topic: audio.transcription.completed TopicId: D7WGO-GASqy3DvygscTUhw PartitionCount: 2      ReplicationFactor: 2      Configs: min.insync.replicas=2,message.format.version
=3.0-IV1,unclean.leader.election.enable=true
    Topic: audio.transcription.completed      Partition: 0      Leader: 1      Replicas: 1,3      Isr: 1,3
    Topic: audio.transcription.completed      Partition: 1      Leader: 2      Replicas: 2,1      Isr: 2,1

--- Topic: audio.generation.requested ---
Topic: audio.generation.requested TopicId: t3xL-Do9RsgK7oM2AlOMWA PartitionCount: 2      ReplicationFactor: 2      Configs: min.insync.replicas=2,message.format.version
=3.0-IV1,unclean.leader.election.enable=true
    Topic: audio.generation.requested      Partition: 0      Leader: 3      Replicas: 3,1      Isr: 3,1
    Topic: audio.generation.requested      Partition: 1      Leader: 1      Replicas: 1,2      Isr: 1,2

--- Topic: audio.generation.completed ---
Topic: audio.generation.completed TopicId: hyDRhhNASYGwTvUV8DUYg PartitionCount: 2      ReplicationFactor: 2      Configs: min.insync.replicas=2,message.format.version
=3.0-IV1,unclean.leader.election.enable=true
    Topic: audio.generation.completed      Partition: 0      Leader: 2      Replicas: 2,1      Isr: 2,1
    Topic: audio.generation.completed      Partition: 1      Leader: 3      Replicas: 3,2      Isr: 3,2

--- Topic: chat.message ---
Topic: chat.message TopicId: WctIJ9V8Ti6WhUGLIB0lg PartitionCount: 6      ReplicationFactor: 2      Configs: min.insync.replicas=2,message.format.version=3.0-IV1,unclean
.leader.election.enable=true
    Topic: chat.message      Partition: 0      Leader: 3      Replicas: 3,1      Isr: 3,1
    Topic: chat.message      Partition: 1      Leader: 1      Replicas: 1,2      Isr: 1,2
    Topic: chat.message      Partition: 2      Leader: 2      Replicas: 2,3      Isr: 2,3
    Topic: chat.message      Partition: 3      Leader: 3      Replicas: 3,2      Isr: 3,2
    Topic: chat.message      Partition: 4      Leader: 1      Replicas: 1,3      Isr: 1,3
    Topic: chat.message      Partition: 5      Leader: 2      Replicas: 2,1      Isr: 2,1
```

Kafka CLI output confirms topic configurations with 3 partitions, 2 replicas, and active ISR sets

```
sh-5.2$ CLASSPATH="$JAR" "$KAFKA_BIN/kafka-console-consumer.sh" \
--bootstrap-server "$BOOT" \
--topic quiz.requested \
--consumer.config "$CFG" \
--group my-test-group \
--from-beginning
hello test 1764863412
test message
hello-from-producer-1764862415
hello test 1764862775
hello test 1764863431
hello from producer
hi from producer
```

Messages published to 'quiz.requested' topic using MSK console producer

```
sh-5.2$ CLASSPATH="$JAR" "$KAFKA_BIN/kafka-console-producer.sh" \
  --broker-list "$BOOT" \
  --topic quiz.requested \
  --producer.config "$CFG"
>hello from producer
>hi from producer 
>
```

MSK console consumer connected to topic 'quiz.requested' and received messages successfully.

3.2 Kafka Integration Patterns

1. Event Sourcing

Purpose:

Track every state change as an immutable sequence of events instead of storing only the final state.

How it is implemented:

- Services produce events to Kafka (e.g., document.events, quiz.requested).
 - Consumers rebuild state from these events.

Benefits:

- Full audit/history
 - Debugging and replay
 - Supports rebuilding materialized views

2 C0RS

Purpose:

Separate write operations (commands) from read operations (queries)

How it is implemented:

- Write side publishes events (e.g., `audio.transcription.requested`).
 - Read side consumes and builds query models.

Benefits:

- Scalable reads
- Independent scaling
- Cleaner boundaries

3. Saga Pattern

Purpose:

Handle multi-step transactions across microservices.

How it is implemented:

- Orchestrator uses saga.requests.
- Services respond with success/failure events.
- Compensating events undo failures.

Benefits:

- Reliable workflows
- Rollback supported
- Loose coupling

4. Event Notification

Purpose:

Notify services of significant events.

How it is implemented:

- Services publish notifications (e.g., notifications.document).

Benefits:

- Loose coupling
- Real-time updates
- Easy extensibility

4. API Gateway Implementation

4.1 API Gateway Architecture

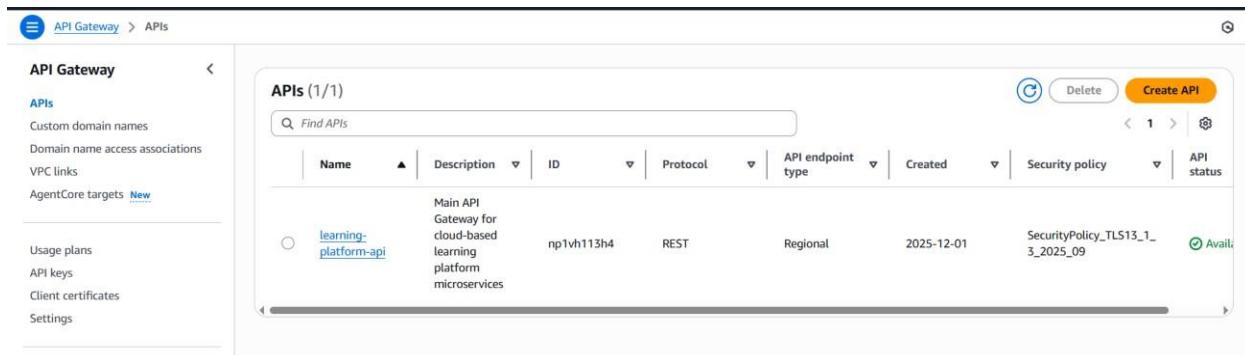
This guide outlines the chronological sequence of steps executed to establish a secure, scalable API Gateway using AWS services, fulfilling all technical and documentation requirements for the project.

I. Gateway Foundation and Routing Setup

This phase established the single entry point and defined the mechanism for routing incoming traffic to the backend microservices (running behind the ALB).

Step 1: API Gateway Creation and Stage Setup

1. Implementation commenced with the deployment of the AWS API Gateway (REST API), named learning-platform-api.
2. Purpose: This API was established as the central point of control for the entire application traffic.



The screenshot shows the AWS API Gateway console. On the left, there's a sidebar with 'API Gateway' selected, followed by 'APIs', 'Custom domain names', 'Domain name access associations', 'VPC links', 'AgentCore targets', 'Usage plans', 'API keys', 'Client certificates', and 'Settings'. The main area is titled 'APIs (1/1)' with a search bar. A table lists the single API: Name (learning-platform-api), Description (Main API Gateway for cloud-based learning platform microservices), ID (np1vh113h4), Protocol (REST), API endpoint type (Regional), Created (2025-12-01), Security policy (SecurityPolicy_TLS13_1_3_2025_09), and API status (Available). There are buttons for 'Create API' and 'Delete' at the top right of the table.

Step 2: Resource and Proxy Configuration

1. All primary service path structures (e.g., /api/tts, /api/documents, /api/chat, etc.) were established as resources under the main /apiroot.
2. To enable comprehensive routing, a child resource /{proxy+} was created for each service path, which is mandatory for **Proxy Integration** to ensure all underlying requests (e.g., /api/tts/synthesize) are handled correctly.
3. **CORS (Cross-Origin Resource Sharing)** was globally enabled on all resources to facilitate frontend connectivity.

API Gateway

- APIs
- Custom domain names
- Domain name access associations
- VPC links
- AgentCore targets [New](#)

▼ API: learning-platform-api

- Resources
- Stages
- Authorizers
- Gateway responses
- Models
- Resource policy
- Documentation
- Dashboard
- API settings

Usage plans
API keys

Resources

Create resource

- /quiz
 - OPTIONS
 - /{proxy+}
 - ANY
 - OPTIONS
- /stt
 - OPTIONS
 - /{proxy+}
 - ANY
 - OPTIONS
- /tts
 - OPTIONS
 - /{proxy+}
 - ANY
 - OPTIONS

Resource details

Path /api/tts
Resource ID 8j3djt

Methods (1)

Method type	Integration type	Authorization	API key
OPTIONS	Mock	None	Not required

API actions **Deploy API**

API Gateway

- APIs
- Custom domain names
- Domain name access associations
- VPC links
- AgentCore targets [New](#)

▼ API: learning-platform-api

- Resources
- Stages
- Authorizers
- Gateway responses
- Models
- Resource policy
- Documentation
- Dashboard
- API settings

Usage plans
API keys

Resources

Create resource

- /quiz
 - OPTIONS
 - /{proxy+}
 - ANY
 - OPTIONS
- /stt
 - OPTIONS
 - /{proxy+}
 - ANY
 - OPTIONS
- /tts
 - OPTIONS
 - /{proxy+}
 - ANY
 - OPTIONS

Resource details

Path /api/stt
Resource ID 7un4q2

Methods (1)

Method type	Integration type	Authorization	API key
OPTIONS	Mock	None	Not required

API actions **Deploy API**

⌚ Successfully created resource '/api/quiz/{proxy+}' ×

Resources

Create resource

- /api
 - OPTIONS
 - /chat
 - OPTIONS
 - /documents
 - OPTIONS
 - /quiz
 - OPTIONS
 - /{proxy+}
 - ANY
 - OPTIONS
- /stt

Resource details

Path /api/quiz/{proxy+}
Resource ID 9qk8ts

Methods (2)

Method type	Integration type	Authorization	API key
ANY	Not setup	None	Not required
OPTIONS	Mock	None	Not required

API actions **Deploy API**

Successfully created resource '/api/documents/{proxy+}'

② 0 △ 0 ② 6 ① 0 ③ 0 X

Resources

Resource details													
Path	/api/documents/{proxy+}												
Resource ID	qw5mhw												
Methods (2) <table border="1"> <thead> <tr> <th>Method type</th> <th>Integration type</th> <th>Authorization</th> <th>API key</th> </tr> </thead> <tbody> <tr> <td>ANY</td> <td>Not setup</td> <td>None</td> <td>Not required</td> </tr> <tr> <td>OPTIONS</td> <td>Mock</td> <td>None</td> <td>Not required</td> </tr> </tbody> </table>		Method type	Integration type	Authorization	API key	ANY	Not setup	None	Not required	OPTIONS	Mock	None	Not required
Method type	Integration type	Authorization	API key										
ANY	Not setup	None	Not required										
OPTIONS	Mock	None	Not required										

Create resource

- ✓ /
- ✓ /api
 - OPTIONS
 - ✓ /chat
 - OPTIONS
 - ✓ /documents
 - OPTIONS
 - ✓ /{proxy+}
 - ANY
 - OPTIONS
 - ✓ /quiz
 - OPTIONS
 - ✓ /{proxy+}

Successfully created resource '/api/chat/{proxy+}'

② 0 △ 0 ② 8 ① 0 ③ 0 X

Resources

Resource details													
Path	/api/chat/{proxy+}												
Resource ID	39lqcj												
Methods (2) <table border="1"> <thead> <tr> <th>Method type</th> <th>Integration type</th> <th>Authorization</th> <th>API key</th> </tr> </thead> <tbody> <tr> <td>ANY</td> <td>Not setup</td> <td>None</td> <td>Not required</td> </tr> <tr> <td>OPTIONS</td> <td>Mock</td> <td>None</td> <td>Not required</td> </tr> </tbody> </table>		Method type	Integration type	Authorization	API key	ANY	Not setup	None	Not required	OPTIONS	Mock	None	Not required
Method type	Integration type	Authorization	API key										
ANY	Not setup	None	Not required										
OPTIONS	Mock	None	Not required										

Create resource

- ✓ /
- ✓ /api
 - OPTIONS
 - ✓ /chat
 - OPTIONS
 - ✓ /{proxy+}
 - ANY
 - OPTIONS
 - ✓ /documents
 - OPTIONS
 - ✓ /{proxy+}
 - ANY
 - OPTIONS

Step 3: Setting HTTP Proxy Integration

1. The **Integration Request** for the ANY method under each /{proxy+}resource was configured as **HTTP Proxy**.
2. The **Endpoint URL** was explicitly set to the **Application Load Balancer (ALB)** DNS name, including the required service path (e.g., <http://dev-public-alb.../api/tts/{proxy}>).
3. The **Content Handling** setting was maintained as **Passthrough** to ensure the gateway transmits raw data (file uploads, complex JSON) directly to the backend services without alteration.
4. **Output:** This step successfully linked the public API endpoint to the private backend ALB.

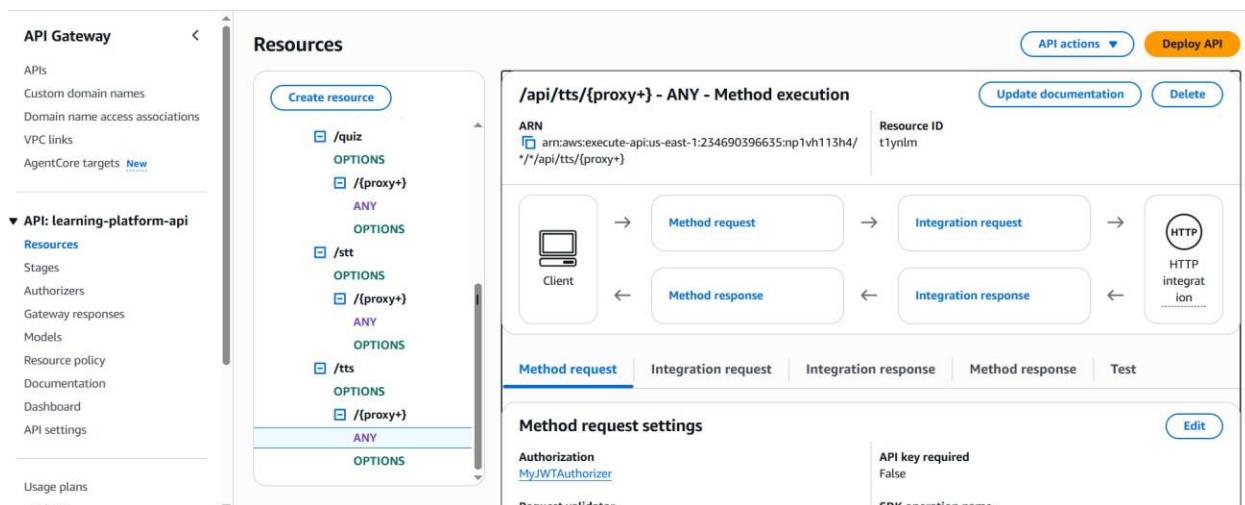
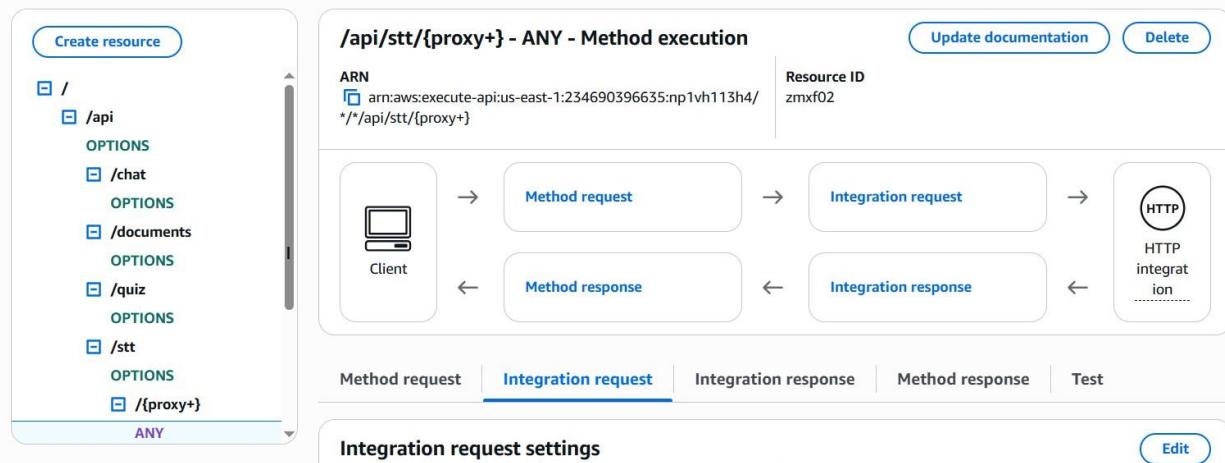
✓ Successfully updated method request settings for 'ANY' in '{proxy+}'. Redeploy your API for the update to take effect.

0 0 3 0 0 ✓

Resources

API actions ▾

Deploy API

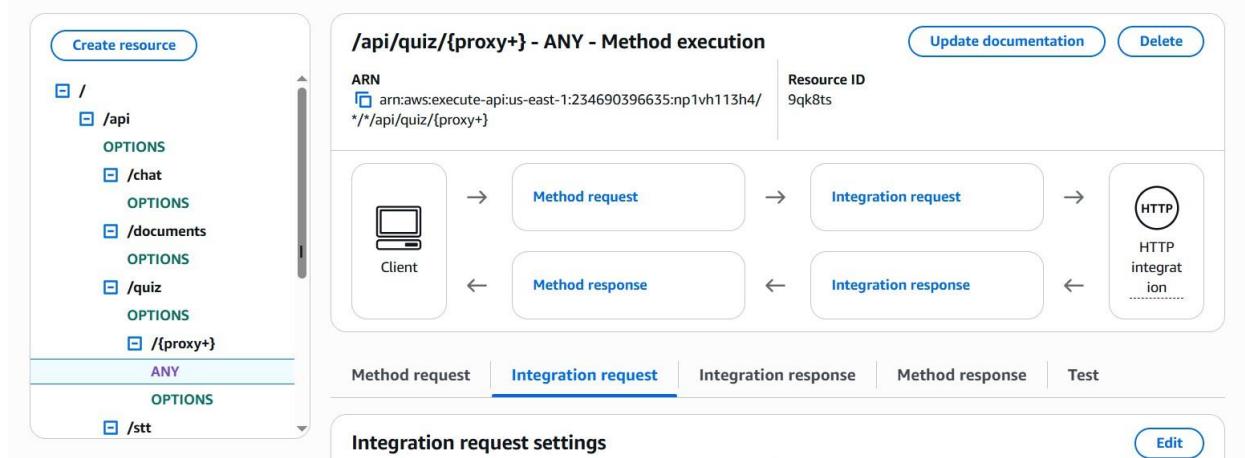


✓ Successfully updated method request settings for 'ANY' in '{proxy+}'. Redeploy your API for the update to take effect.

01 00 05 00 00

Resources

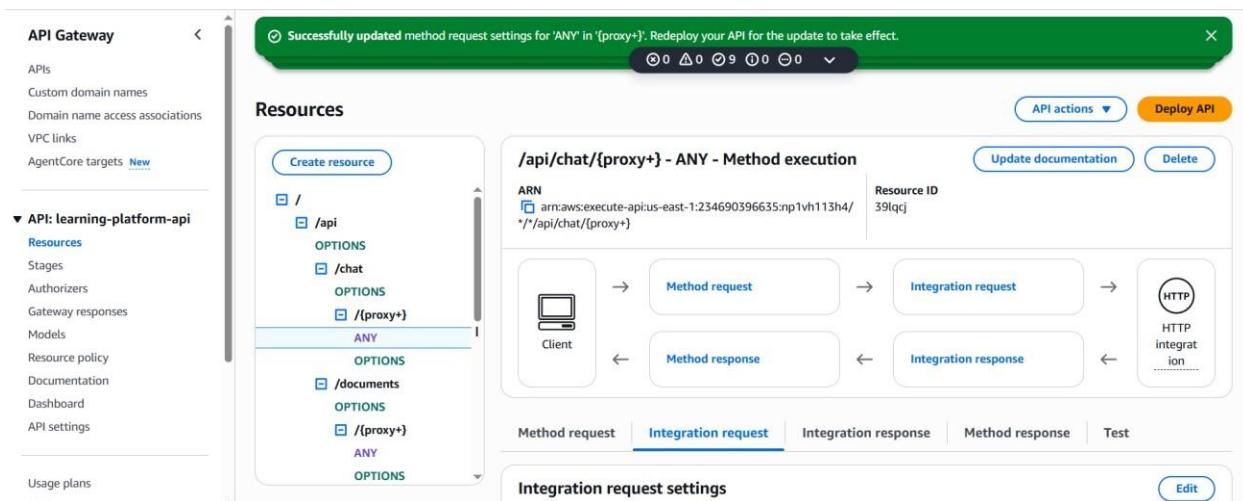
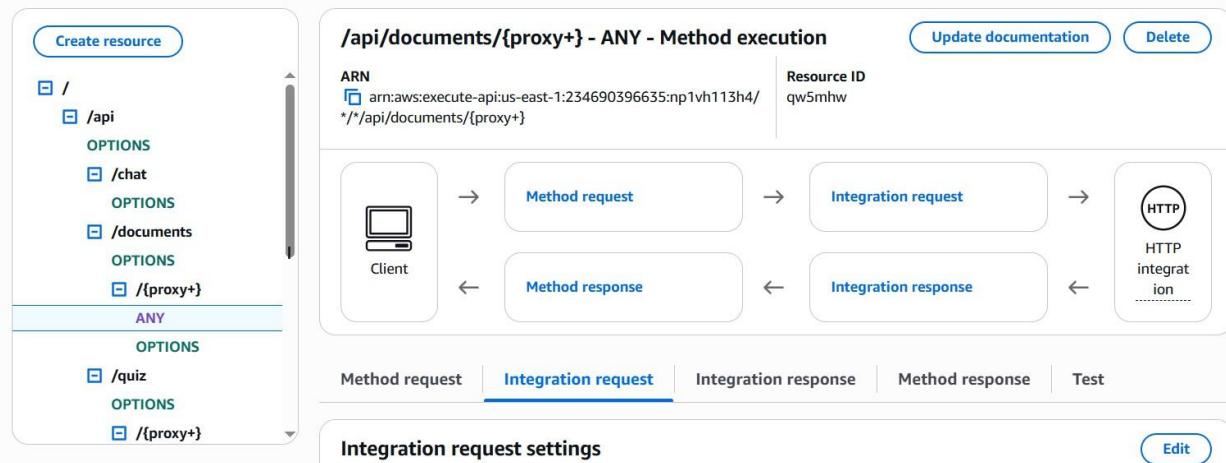
Deploy API



Successfully updated method request settings for 'ANY' in '{proxy+}'. Redeploy your API for the update to take effect.

0 0 0 7 0 0

Resources



II. Security and Control Implementation

Step 4: Creating the Lambda Authorizer Function

- A dedicated Lambda function named **JWTAuthorizerFunc** was created in the AWS Lambda console. This function was deployed with the necessary code to contain the **JWT validation logic**.
- **Purpose:** This function serves as the external security policy engine for the gateway.

Create function Info

Choose one of the following options to create your function.

- Author from scratch
Start with a simple Hello World example.
- Use a blueprint
Build a Lambda application from sample code and configuration presets for common use cases.
- Container image
Select a container image to deploy for your function.

Basic information

Function name

Enter a name that describes the purpose of your function.

Function name must be 1 to 64 characters, must be unique to the Region, and can't include spaces. Valid characters are a-z, A-Z, 0-9, hyphens (-), and underscores (_).

Runtime Info

Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.



Architecture Info

Choose the instruction set architecture you want for your function code.

- arm64
- x86_64

The screenshot shows the AWS Lambda Function Overview page for a function named "JWTAuthorizerFunc". The "Function overview" tab is selected. On the left, there's a "Diagram" button (which is currently active) and a "Template" button. The main area displays the function name "JWTAuthorizerFunc" with a small Lambda icon, "Layers (0)", and buttons for "+ Add trigger" and "+ Add destination". To the right, there are sections for "Description", "Last modified" (12 seconds ago), "Function ARN" (arn:aws:lambda:us-east-1:254690396635:function:JWTAuthorizerFunc), and "Function URL" (info). Below these are tabs for "Code", "Test", "Monitor", "Configuration", "Aliases", and "Versions". At the bottom, there are buttons for "Open in Visual Studio Code" and "Upload from". On the right side of the page, there's a sidebar titled "Tutorials" with a section for "Create a simple web app". It includes a brief description and a bulleted list of steps: "Build a simple web app, consisting of a Lambda function with a function URL that outputs a webpage" and "Invoke your function through its function URL". There are also "Learn more" and "Start tutorial" buttons.

Step 5: Defining the Authorizer in API Gateway

- A new **Lambda Authorizer** instance, named **MyJWTAuthorizer**, was defined in the API Gateway console and successfully linked to the **JWTAuthorizerFunc** from Step 4.
- The **Token source** was explicitly configured to read the **Authorization** request header, which is the standard location for receiving the JWT from the client.
- **Output:** The Gateway is now configured to invoke the Lambda function for authentication checks.

Create authorizer Info

Authorizer details

Authorizer name
MyJWTAuthorizer

Authorizer type Info
Choose to authorize your API calls using one of your Lambda functions or a Cognito User Pool.
 Lambda
 Cognito

Lambda function
Provide the Lambda function name or alias. You can also provide an ARN from another account.
 us-east-1 arn:aws:lambda:us-east-1:234690396635:function:JWTAuthorizerFunc

Grant API Gateway permission to invoke your Lambda function
When you save your changes, API Gateway updates your Lambda function's resource-based policy to allow this API to invoke it.

Lambda invoke role - optional
Specify an optional role API Gateway will use to make requests to your authorizer. For optimal API performance it is strongly recommended to activate Regional STS in the region where your API is located.

Lambda event payload
Choose token to send a single header that contains an authorization token. Choose request to send all request parameters.

≡ API Gateway > APIs > learning-platform-api (np1vh113h4) > Authorizers

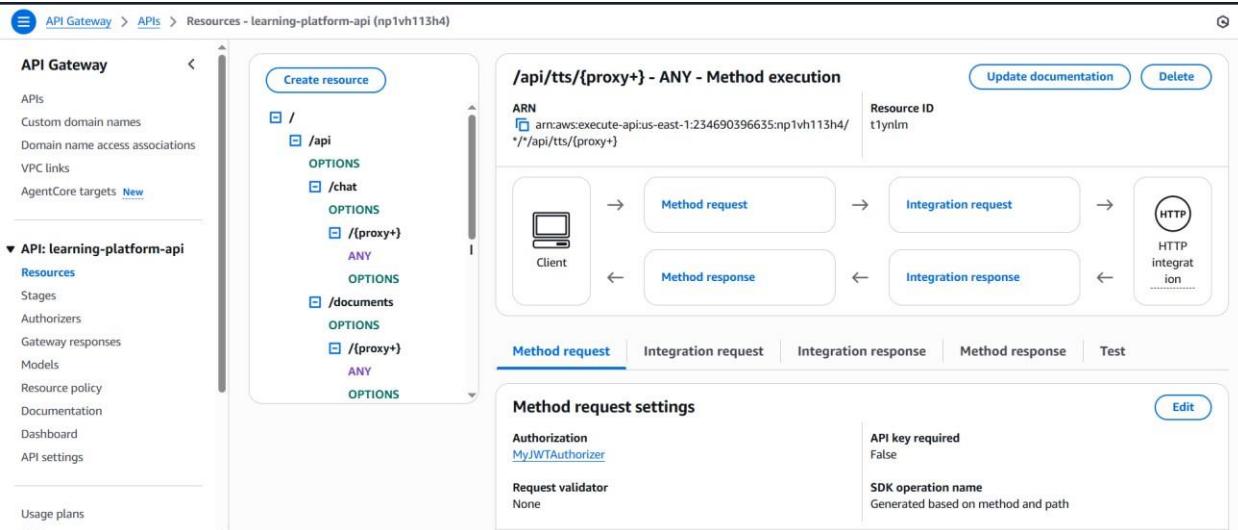
⋮ ⌂ ⌃

⌚ Successfully created authorizer 'MyJWTAuthorizer'.

Authorizers (1) <small>Info</small>	
MyJWTAuthorizer	Edit Delete Create authorizer <p>Authorizer ID jdjhjh</p> <p>Lambda function JWTAuthorizerFunc (us-east-1)</p> <p>Lambda invoke role - optional -</p> <p>Lambda event payload Token</p> <p>Token source Authorization</p> <p>Token validation - optional None</p> <p>Authorization caching</p>

Step 6: Enforcing Security on Endpoints

- The **Authorization** setting in the Method Request tab for all proxy methods (ANY) was changed from NONE to **MyJWTAuthorizer**.
- (**Crucial Check**): The **API key required** checkbox was left **unchecked**, as authentication relies solely on the JWT mechanism.
 - **Output:** All service endpoints are now secured, fulfilling the **JWT-based authentication** requirement.



Step 7: Configure Logging Permissions (IAM Prerequisite)

1. To grant API Gateway permission to write execution logs, a dedicated **IAM Role** named **APIGatewayCloudWatchLogsRole** was created.
2. The managed policy **AmazonAPIGatewayPushToCloudWatchLogs** was attached to this role, granting the permissions required to write logs to CloudWatch.
3. The ARN of this new role was then copied and pasted into the global **API Gateway Settings** under the **CloudWatch log role ARN** field, successfully linking the permissions to the service.
4. **Output:** This step resolved the prerequisite error for enabling logging on the deployment stage.

Identity and Access Management (IAM)

APIGatewayCloudWatchLogsRole [Info](#)

Allows API Gateway to push logs to CloudWatch Logs.

Summary		Edit
Creation date	December 04, 2025, 20:10 (UTC+02:00)	ARN
Last activity	-	Maximum session duration
AWS managed		

Permissions [Trust relationships](#) [Tags](#) [Last Accessed](#) [Revoke sessions](#)

Permissions policies (1) [Info](#)

You can attach up to 10 managed policies.

Filter by Type	
Search	All types
<input type="checkbox"/> Policy name AmazonAPIGatewayPushToCloudWatchLogsPolicy	Type
<input type="checkbox"/> AmazonAPIGatewayPushToCloudWatchLogsPolicy	AWS managed
Attached entities	
1	

[Simulate](#) [Remove](#) [Add permissions](#)

API Gateway > Settings > Edit logging settings

Edit logging settings

CloudWatch log role [Info](#)

CloudWatch log role ARN

arn:aws:iam::234690396635:role/APIGatewayCloudWatchLogsRole

[Cancel](#) [Save changes](#)

API Gateway > Settings

Settings

Logging [Info](#)

To write to CloudWatch logs in your account, create an Identity and Access Management (IAM) role and add the required permissions for CloudWatch.

CloudWatch log role ARN

arn:aws:iam::234690396635:role/APIGatewayCloudWatchLogsRole

Throttling [Info](#)

To change account-level throttling limits, you can request a service quota increase in the Service Quotas console or the AWS Support Center Console.

Account-level throttling

Your current account-level throttling rate is 10000 requests per second with a burst of 5000 requests. [Learn more](#)

Step 8: Configuring Logging and Monitoring

- In the Stages settings for the prod stage, the **CloudWatch logs** dropdown was set to **Errors and Info logs**.
 - The **Detailed metrics** checkbox was also activated to enable granular monitoring for latency and error rates.
- Output:** This fulfills the **Request Logging and Monitoring** requirement.

The screenshot shows the AWS API Gateway interface. A modal window titled "Deploy API" is open. It displays a success message: "Successfully updated". Below it, there's a section for creating a new stage with a dropdown menu set to "New stage" and a text input field containing "prod". A note says: "A new stage will be created with the default settings. Edit your stage settings on the Stage page." At the bottom right of the modal are "Cancel" and "Deploy" buttons. The background shows the main API Gateway dashboard with various resources and stages listed.

The screenshot shows the "Edit logs and tracing" page for the "learning-platform-api". Under the "Logs and tracing settings" section, several options are listed:

- CloudWatch logs:** Errors and info logs (selected)
- Data tracing:** (disabled)
- Detailed metrics:** (enabled)
- Custom access logging:** (disabled)
- X-Ray tracing:** (disabled)

Below this, a note states: "When you turn on X-Ray tracing, API Gateway will create an IAM role with permissions: 'xray:PutTraceSegments', 'xray:PutTelemetryRecords'. Learn more" with a link icon.

The screenshot shows the "Stages" page for the "learning-platform-api". The "prod" stage is selected. Key configuration details shown include:

- Default method-level caching:** Inactive
- Invoke URL:** https://np1vh113h4.execute-api.us-east-1.amazonaws.com/prod
- Active deployment:** tjmgnx on December 04, 2025, 18:20 (UTC+02:00)
- Logs and tracing:**
 - CloudWatch logs: Error and info logs (Active)
 - Detailed metrics: Active
 - Data tracing: Inactive
 - X-Ray tracing: Inactive
 - Custom access logging: Inactive

At the bottom, tabs for "Stage variables", "Deployment history", "Documentation history", "Canary", and "Tags" are visible.

Step G: Rate Limiting and Final Deployment

1. A Usage Plan (e.g., StudentPlan) was created with set limits for **Rate** and **Burst** and associated with the prod stage.
2. The plan was explicitly configured with the following constraints to limit usage:
 - a. **Request Rate:** 50 requests per second.
 - b. **Burst Capacity:** 100 requests.
 - c. **Quota:** 1000 requests per day.
3. The StudentPlan was then associated with the deployed **prod** stage of the learning-platform-api, ensuring all incoming traffic adheres to these limits.
4. The entire API was finalized and deployed by clicking "**Deploy API**" and selecting the **prod** stage.
5. **Final Deliverables:** The **API Gateway configuration files** and **API documentation** were generated by exporting the OpenAPI JSON from the final deployment stage.

The screenshot shows the configuration of a usage plan in an API Gateway. The URL in the address bar is [API Gateway > Usage plans > StudentPlan > Edit usage plan](#). The page title is "Usage plan details". The "Name" field is filled with "StudentPlan". The "Description - optional" field contains the text: "Initial production deployment routing all API traffic (TTS, STT, Chat, Docs, Quiz) to the backend Application Load Balancer via proxy integration." Under the "Throttling" section, the "Rate" is set to 50 and the "Burst" is set to 100. Under the "Quota" section, the "Requests" field is set to 1000 with a dropdown menu showing "Per day".

aws [Alt+S]

Search United States (N. Virginia) Account ID: 2546-9039-6635 Antonius sameh

[API Gateway](#) > Usage plans

⌚ Successfully created usage plan 'StudentPlan'.

Name	ID	Request rate	Burst	Quota
StudentPlan	zqajqx	50 per second	100	1000 per day

[API Gateway](#) > Usage plans > StudentPlan

⌚ Successfully added stage "prod" to usage plan.

StudentPlan Last updated December 05, 2025, 15:20 (UTC+02:00)

Usage plan details	Rate 50 requests per second
Usage plan ID zqajqx	Burst 100 requests
Description Initial production deployment routing all API traffic (TTS, STT, Chat, Docs, Quiz) to the backend Application Load Balancer via proxy integration.	Quota 1000 requests per day
AWS Marketplace product code -	

[Associated stages](#) [Associated API keys](#) [Tags](#)

API	Stage	Method throttling
learning-platform-api	prod	-

[API Gateway](#) > APIs > learning-platform-api (np1vh113h4) > Stages

⌚ Successfully created deployment for learning-platform-api. This deployment is active for prod.

Stages

prod	Stage details Edit
	Stage name prod
	Rate Info 10000
	Cache cluster Info Inactive
	Burst Info 5000
	Default method-level caching Inactive
	Invoke URL https://np1vh113h4.execute-api.us-east-1.amazonaws.com/prod
	Active deployment 00a0gu on December 05, 2025, 15:28 (UTC+02:00)
	Logs and tracing Info Edit

III. API Configuration and Documentation

This requirement is met by a single generated file containing the complete technical specification of the deployment.

- **Deliverable: OpenAPI (Swagger) Specification File (JSON).**
 - **Proof of Configuration:** The file was exported using the **API Gateway Extensions** option. This ensures the output includes all underlying integration logic, such as the direct links to the **Application Load Balancer (ALB)** and the invocation setup for the **Lambda Authorizer**.

```
{
  "swagger" : "2.0", "info" : {
    "description" : "Main API Gateway for cloud-based learning platform microservices\n",
    "version" : "2025-12-04T15:27:00Z",
    "title" : "learning-platform-api"
  },
  "host" : "np1vh113h4.execute-api.us-east-1.amazonaws.com", "basePath" :
  "/prod",
  "schemes" : [ "https" ], "paths" : {
    "/api" : {
      "options" : {
        "consumes" : [ "application/json" ], "produces" : [
          "application/json" ],
        "tags" : [
          "Learning Platform"
        ]
      }
    }
  }
}
```

```

"responses" : {
    "200" : {
        "description" : "200 response",
        "schema" : {
            "$ref" : "#/definitions/Empty"
        },
        "headers" : {
            "Access-Control-Allow-Origin" : { "type" :
                "string"
            },
            "Access-Control-Allow-Methods" : { "type" :
                "string"
            },
            "Access-Control-Allow-Headers" : { "type" :
                "string"
            }
        }
    },
    "x-amazon-apigateway-integration" : { "responses" : {
        "default" : { "statusCode" :
            "200",
            "responseParameters" : {
                "method.response.header.Access-Control-Allow-Methods" :
                "'DELETE,GET,HEAD,OPTIONS,PATCH,POST,PUT'",
                "method.response.header.Access-Control-Allow-Headers" :
                "'Content-Type,Authorization,X-Amz-Date,X-Api-Key,X-Amz-Security-Token'",
                "method.response.header.Access-Control-Allow-Origin" : "'*'"
            }
        }
    },
    "requestTemplates" : {
        "application/json" : "{\"statusCode\": 200}"
    },
    "passthroughBehavior" : "when_no_match", "type" :
    "mock"
}
},
"/api/chat" : {
    "options" : {
        "consumes" : [ "application/json" ],
        "produces" : [ "application/json" ],
        "responses" : {
            "200" : {
                "description" : "200 response",
                "schema" : {

```



```

},
"headers" : {
    "Access-Control-Allow-Origin" : { "type" :
        "string"
    },
    "Access-Control-Allow-Methods" : { "type" :
        "string"
    },
    "Access-Control-Allow-Headers" : { "type" :
        "string"
    }
}
},
"x-amazon-apigateway-integration" : { "responses" : {
    "default" : { "statusCode" :
        "200",
        "responseParameters" :{
            "method.response.header.Access-Control-Allow-Methods" :
                "'DELETE,GET,HEAD,OPTIONS,PATCH,POST,PUT'",
            "method.response.header.Access-Control-Allow-Headers" : "'Content-
Type,Authorization,X-Amz-Date,X-Api-Key,X-Amz-Security-Token'",
            "method.response.header.Access-Control-Allow-Origin" : "'*'"
        }
    }
},
"requestTemplates" : {
    "application/json" : "{\"statusCode\": 200}"
},
"passthroughBehavior" : "when_no_match", "type" :
"mock"
}
},
"/api/chat/{proxy+}" : {
    "options" : {
        "consumes" : [ "application/json" ],
        "produces" : [ "application/json" ],
        "parameters" : [ {
            "name" : "proxy",
            "in" : "path", "required"
            : true, "type" : "string"
        }],
        "responses" : {
            "200" : {
                "description" : "200 response",
                "schema" : {

```

```

"$ref" : "#/definitions/Empty"
},
"headers" : {
    "Access-Control-Allow-Origin" : { "type" :
        "string"
    },
    "Access-Control-Allow-Methods" : { "type" :
        "string"
    },
    "Access-Control-Allow-Headers" : { "type" :
        "string"
    }
}
},
"x-amazon-apigateway-integration" : { "responses" : {
    "default" : { "statusCode" :
        "200",
        "responseParameters" : {
            "method.response.header.Access-Control-Allow-Methods" :
                "DELETE,GET,HEAD,OPTIONS,PATCH,POST,PUT",
            "method.response.header.Access-Control-Allow-Headers" : "'Content-
Type,Authorization,X-Amz-Date,X-Api-Key,X-Amz-Security-Token'",
            "method.response.header.Access-Control-Allow-Origin" : "'*'"
        }
    }
},
    "requestTemplates" : {
        "application/json" : "{\"statusCode\": 200}"
    },
    "passthroughBehavior" : "when_no_match", "type" :
        "mock"
}
},
"x-amazon-apigateway-any-method" : { "produces" : [
    "application/json" ], "parameters" : [ {
        "name" : "proxy",
        "in" : "path", "required"
        : true, "type" : "string"
    }],
    "responses" : { },
    "x-amazon-apigateway-integration" : {
        "uri" : "http://dev-public-alb-954135140.us-east-1.elb.amazonaws.com/api/chat/{proxy}",
        "httpMethod" : "ANY",
        "responses" : {

```

```

"default" : { "statusCode" :
    "200"
},
},
"requestParameters" : {
    "integration.request.path.proxy" : "method.request.path.proxy"
},
"passthroughBehavior" : "when_no_templates",
"type" : "http"
}
},
"/api/documents" : {
    "options" : {
        "consumes" : [ "application/json" ],
        "produces" : [ "application/json" ],
        "responses" : {
            "200" : {
                "description" : "200 response",
                "schema" : {
                    "$ref" : "#/definitions/Empty"
                },
                "headers" : {
                    "Access-Control-Allow-Origin" : { "type" :
                        "string"
                    },
                    "Access-Control-Allow-Methods" : { "type" :
                        "string"
                    },
                    "Access-Control-Allow-Headers" : { "type" :
                        "string"
                    }
                }
            }
        }
    },
    "x-amazon-apigateway-integration" : { "responses" : {
        "default" : { "statusCode" :
            "200",
            "responseParameters" : {
                "method.response.header.Access-Control-Allow-Methods" :
                "'DELETE,GET,HEAD,OPTIONS,PATCH,POST,PUT'",
                "method.response.header.Access-Control-Allow-Headers" : "'Content-Type,Authorization,X-Amz-Date,X-Api-Key,X-Amz-Security-Token'",
                "method.response.header.Access-Control-Allow-Origin" : "'*'"
            }
        }
    }
}

```

"requestTemplates" : {

```

        "application/json" : "{\"statusCode\": 200}"
    },
    "passthroughBehavior" : "when_no_match", "type" :
    "mock"
}
},
"/api/documents/{proxy+}" : {
    "options" : {
        "consumes" : [ "application/json" ],
        "produces" : [ "application/json" ],
        "parameters" : [ {
            "name" : "proxy",
            "in" : "path", "required"
            : true, "type" : "string"
        } ],
        "responses" : {
            "200" : {
                "description" : "200 response",
                "schema" : {
                    "$ref" : "#/definitions/Empty"
                },
                "headers" : {
                    "Access-Control-Allow-Origin" : { "type" :
                        "string"
                    },
                    "Access-Control-Allow-Methods" : { "type" :
                        "string"
                    },
                    "Access-Control-Allow-Headers" : { "type" :
                        "string"
                    }
                }
            },
            "x-amazon-apigateway-integration" : { "responses" : {
                "default" : { "statusCode" :
                    "200",
                    "responseParameters" : {
                        "method.response.header.Access-Control-Allow-Methods" :
                        "'DELETE,GET,HEAD,OPTIONS,PATCH,POST,PUT'",
                        "method.response.header.Access-Control-Allow-Headers" : "'Content-Type,Authorization,X-Amz-Date,X-Api-Key,X-Amz-Security-Token'",
                        "method.response.header.Access-Control-Allow-Origin" : "'*'"
                    }
                }
            }
        }
    }
}

```

```
"requestTemplates" : {
    "application/json" : "{\"statusCode\": 200}"
},
"passthroughBehavior" : "when_no_match", "type" :
"mock"
}
},
"x-amazon-apigateway-any-method" : { "produces" : [
"application/json" ], "parameters" : [ {
"name" : "proxy",
"in" : "path", "required"
: true, "type" : "string"
} ],
"responses" : { },
"x-amazon-apigateway-integration" : {
"uri" : "http://dev-public-alb-954135140.us-east- 1.elb.amazonaws.com/api/documents/{proxy}",
"httpMethod" : "ANY",
"responses" : {
"default" : { "statusCode" :
"200"
}
},
"requestParameters" : {
"integration.request.path.proxy" : "method.request.path.proxy"
},
"passthroughBehavior" : "when_no_templates",
"type" : "http"
}
}
},
"/api/quiz" : {
"options" : {
"consumes" : [ "application/json" ],
"produces" : [ "application/json" ],
"responses" : {
"200" : {
"description" : "200 response",
"schema" : {
"$ref" : "#/definitions/Empty"
},
"headers" : {
"Access-Control-Allow-Origin" : { "type" :
"string"
},
"Access-Control-Allow-Methods" : { "type" :
"string"
}
}
}
}
}
```

```

},
"Access-Control-Allow-Headers" : { "type" :
    "string"
}
}
}
},
"x-amazon-apigateway-integration" : { "responses" : {
"default" : { "statusCode" :
    "200",
"responseParameters" : {
        "method.response.header.Access-Control-Allow-Methods" :
"\"DELETE,GET,HEAD,OPTIONS,PATCH,POST,PUT\"",
        "method.response.header.Access-Control-Allow-Headers" : "\"Content-
Type,Authorization,X-Amz-Date,X-Api-Key,X-Amz-Security-Token\"",
        "method.response.header.Access-Control-Allow-Origin" : "*"
    }
}
},
"requestTemplates" : {
    "application/json" : "{\"statusCode\": 200}"
},
"passthroughBehavior" : "when_no_match", "type" :
"mock"
}
},
"/api/quiz/{proxy+}" : {
"options" : {
    "consumes" : [ "application/json" ],
    "produces" : [ "application/json" ],
    "parameters" : [ {
        "name" : "proxy",
        "in" : "path", "required" :
true, "type" : "string"
    }],
    "responses" : {
        "200" : {
            "description" : "200 response",
            "schema" : {
                "$ref" : "#/definitions/Empty"
            },
            "headers" : {
                "Access-Control-Allow-Origin" : { "type" :
                    "string"
            },
            "Access-Control-Allow-Methods" : {

```

```

        "type" : "string"
    },
    "Access-Control-Allow-Headers" : { "type" :
        "string"
    }
}
},
"x-amazon-apigateway-integration" : { "responses" : {
    "default" : { "statusCode" :
        "200",
        "responseParameters" : {
            "method.response.header.Access-Control-Allow-Methods" :
            "'DELETE,GET,HEAD,OPTIONS,PATCH,POST,PUT'",
            "method.response.header.Access-Control-Allow-Headers" : "'Content-
Type,Authorization,X-Amz-Date,X-Api-Key,X-Amz-Security-Token'",
            "method.response.header.Access-Control-Allow-Origin" : "*"
        }
    }
},
"requestTemplates" : {
    "application/json" : "{\"statusCode\": 200}"
},
"passthroughBehavior" : "when_no_match", "type" :
"mock"
}
},
"x-amazon-apigateway-any-method" : { "produces" : [
    "application/json" ], "parameters" : [ {
        "name" : "proxy",
        "in" : "path", "required"
        : true, "type" : "string"
    } ],
"responses" : { },
"x-amazon-apigateway-integration" : {
    "uri" : "http://dev-public-alb-954135140.us-east-1.elb.amazonaws.com/api/quiz/{proxy}",
    "httpMethod" : "ANY",
    "responses" : {
        "default" : { "statusCode" :
            "200"
        }
    },
    "requestParameters" : {
        "integration.request.path.proxy" : "method.request.path.proxy"
    },

```

```

        "passthroughBehavior" : "when_no_templates",
        "type" : "http"
    }
},
"/api/stt" : {
    "options" : {
        "consumes" : [ "application/json" ],
        "produces" : [ "application/json" ],
        "responses" : {
            "200" : {
                "description" : "200 response",
                "schema" : {
                    "$ref" : "#/definitions/Empty"
                },
                "headers" : {
                    "Access-Control-Allow-Origin" : { "type" :
                        "string"
                    },
                    "Access-Control-Allow-Methods" : { "type" :
                        "string"
                    },
                    "Access-Control-Allow-Headers" : { "type" :
                        "string"
                    }
                }
            }
        }
    },
    "x-amazon-apigateway-integration" : { "responses" : {
        "default" : { "statusCode" :
            "200",
            "responseParameters" : {
                "method.response.header.Access-Control-Allow-Methods" :
                "'DELETE,GET,HEAD,OPTIONS,PATCH,POST,PUT'",
                "method.response.header.Access-Control-Allow-Headers" : "'Content-
Type,Authorization,X-Amz-Date,X-Api-Key,X-Amz-Security-Token'",
                "method.response.header.Access-Control-Allow-Origin" : "'*'"
            }
        }
    }
},
    "requestTemplates" : {
        "application/json" : "{$statusCode": 200}"
    },
    "passthroughBehavior" : "when_no_match", "type" :
    "mock"
}
}
}
```



```
"/api/stt/{proxy+}" : {
  "options" : {
    "consumes" : [ "application/json" ],
    "produces" : [ "application/json" ],
    "parameters" : [ {
      "name" : "proxy",
      "in" : "path", "required" :
        true, "type" : "string"
    }],
    "responses" : {
      "200" : {
        "description" : "200 response",
        "schema" : {
          "$ref" : "#/definitions/Empty"
        },
        "headers" : {
          "Access-Control-Allow-Origin" : { "type" :
            "string"
          },
          "Access-Control-Allow-Methods" : { "type" :
            "string"
          },
          "Access-Control-Allow-Headers" : { "type" :
            "string"
          }
        }
      }
    },
    "x-amazon-apigateway-integration" : { "responses" : {
      "default" : { "statusCode" :
        "200",
        "responseParameters" : {
          "method.response.header.Access-Control-Allow-Methods" :
            "'DELETE,GET,HEAD,OPTIONS,PATCH,POST,PUT'",
          "method.response.header.Access-Control-Allow-Headers" :
            "'Content-Type,Authorization,X-Amz-Date,X-Api-Key,X-Amz-Security-Token'",
          "method.response.header.Access-Control-Allow-Origin" : "*"
        }
      }
    },
    "requestTemplates" : {
      "application/json" : "{\"statusCode\": 200}"
    },
    "passthroughBehavior" : "when_no_match", "type" :
      "mock"
  }
}
```

```

"x-amazon-apigateway-any-method" : { "produces" : [
    "application/json" ], "parameters" : [ {
        "name" : "proxy",
        "in" : "path", "required"
        : true, "type" : "string"
    } ],
    "responses" : { },
    "x-amazon-apigateway-integration" : {
        "uri" : "http://dev-public-alb-954135140.us-east-1.elb.amazonaws.com/api/stt/{proxy}",
        "httpMethod" : "ANY",
        "responses" : {
            "default" : { "statusCode" :
                "200"
            }
        },
        "requestParameters" : {
            "integration.request.path.proxy" : "method.request.path.proxy"
        },
        "passthroughBehavior" : "when_no_templates",
        "type" : "http"
    }
},
"/api/tts" : {
    "options" : {
        "consumes" : [ "application/json" ],
        "produces" : [ "application/json" ],
        "responses" : {
            "200" : {
                "description" : "200 response",
                "schema" : {
                    "$ref" : "#/definitions/Empty"
                }
            },
            "headers" : {
                "Access-Control-Allow-Origin" : { "type" :
                    "string"
                },
                "Access-Control-Allow-Methods" : { "type" :
                    "string"
                },
                "Access-Control-Allow-Headers" : { "type" :
                    "string"
                }
            }
        }
    }
},

```

```
"x-amazon-apigateway-integration" : { "responses" : {
    "default" : { "statusCode" :
        "200",
        "responseParameters" : {
            "method.response.header.Access-Control-Allow-Methods" :
            "\"DELETE,GET,HEAD,OPTIONS,PATCH,POST,PUT\"",
            "method.response.header.Access-Control-Allow-Headers" : "\"Content-Type,Authorization,X-Amz-Date,X-Api-Key,X-Amz-Security-Token\"",
            "method.response.header.Access-Control-Allow-Origin" : "***"
        }
    }
},
"requestTemplates" : {
    "application/json" : "{\"statusCode\": 200}"
},
"passthroughBehavior" : "when_no_match", "type" :
"mock"
}
},
"/api/tts/{proxy+}" : {
    "options" : {
        "consumes" : [ "application/json" ],
        "produces" : [ "application/json" ],
        "parameters" : [ {
            "name" : "proxy",
            "in" : "path", "required"
            : true, "type" : "string"
        } ],
        "responses" : {
            "200" : {
                "description" : "200 response",
                "schema" : {
                    "$ref" : "#/definitions/Empty"
                },
                "headers" : {
                    "Access-Control-Allow-Origin" : { "type" :
                        "string"
                    },
                    "Access-Control-Allow-Methods" : { "type" :
                        "string"
                    },
                    "Access-Control-Allow-Headers" : { "type" :
                        "string"
                    }
                }
            }
        }
    }
}
```

```

},
"x-amazon-apigateway-integration" : {
    "responses" : {
        "default" : {
            "statusCode" : "200",
            "responseParameters" : {
                "method.response.header.Access-Control-Allow-Methods" :
                    "'DELETE,GET,HEAD,OPTIONS,PATCH,POST,PUT'",
                "method.response.header.Access-Control-Allow-Headers" : "'Content-
Type,Authorization,X-Amz-Date,X-Api-Key,X-Amz-Security-Token'",
                "method.response.header.Access-Control-Allow-Origin" : "*"
            }
        }
    },
    "requestTemplates" : {
        "application/json" : "{$statusCode": 200}"
    },
    "passthroughBehavior" : "when_no_match", "type" :
        "mock"
    }
},
"x-amazon-apigateway-any-method" : {
    "produces" : [
        "application/json"
    ],
    "parameters" : [
        {
            "name" : "proxy",
            "in" : "path", "required" :
                true, "type" : "string"
        }
    ],
    "responses" : {}
},
"x-amazon-apigateway-integration" : {
    "uri" : "http://dev-public-alb-954135140.us-east- 1.elb.amazonaws.com/api/tts/{proxy}",
    "httpMethod" : "ANY",
    "responses" : {
        "default" : {
            "statusCode" :
                "200"
        }
    },
    "requestParameters" : {
        "integration.request.path.proxy" : "method.request.path.proxy"
    },
    "passthroughBehavior" : "when_no_templates",
    "type" : "http"
}
},
"definitions" : {
}

```

```

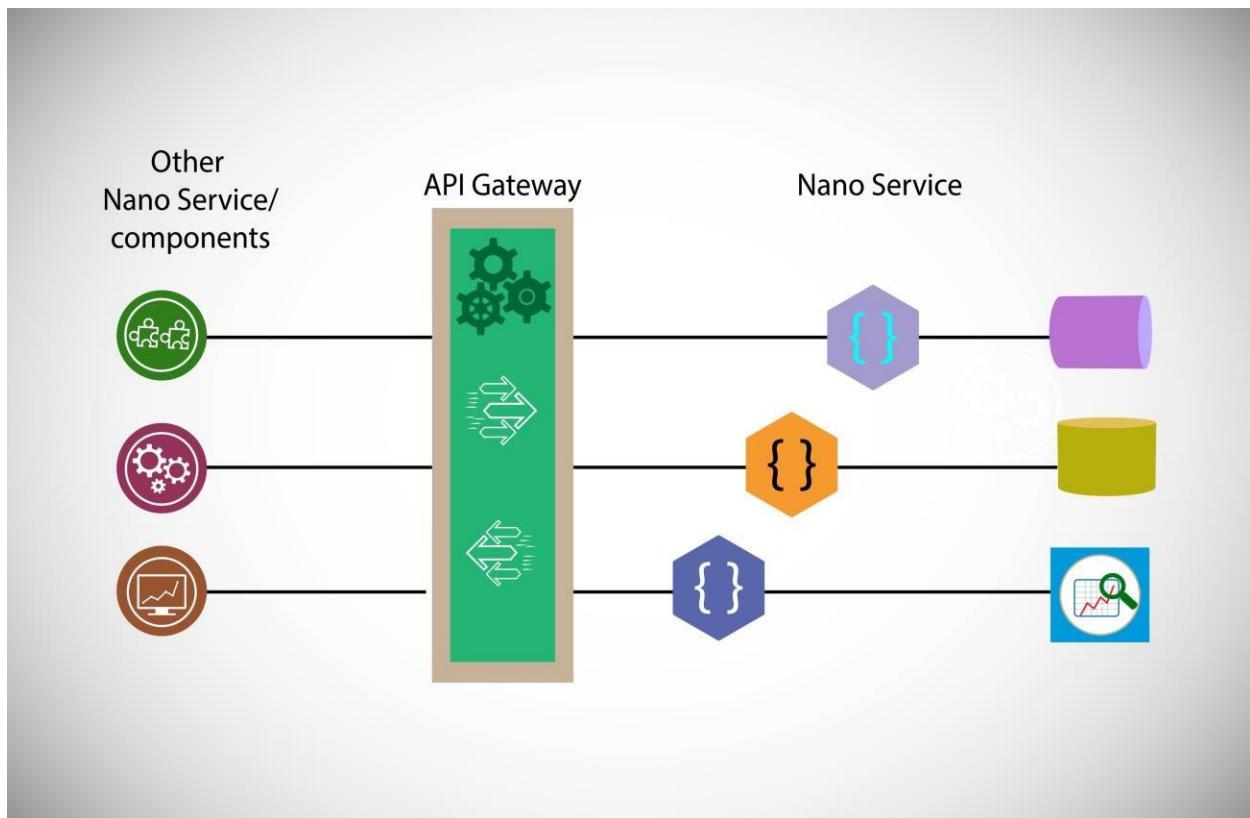
"Empty" : {
    "type" : "object", "title" :
    "Empty Schema"
}
},
"x-amazon-apigateway-security-policy" : "SecurityPolicy_TLS13_1_3_2025_09", "x-
amazon-apigateway-endpoint-access-mode" : "BASIC"
}

```

VI. Authentication Flow Diagrams

- **Deliverable: Flowchart or Detailed Narrative.**

- **Proof of Security Mechanism:** This document outlines the security architecture where the **Lambda Authorizer** intercepts the request and validates the JSON Web Token (JWT) provided in the Authorization header, generating an **IAM Allow Policy** as the prerequisite for backend access.



V. Final Deployment Artifact

- **Production Endpoint (Invoke URL):**
<https://np1vh113h4.execute-api.us-east-1.amazonaws.com/prod>
- *This URL acts as the live entry point for all client requests, verifying that the API Gateway is successfully deployed and accessible on the internet.*

5. Microservices Specifications

5.1 Text-to-Speech (TTS) Service

Overview of the TTS Microservice

Goal: Build a scalable Text-to-Speech service that converts text into audio, stores it, and integrates with Kafka for event streaming.

Key Features

- Generate natural speech (MP3/WAV/OGG)
 - Multi-language support
 - REST APIs for synthesize / retrieve / delete
 - Kafka event-driven pipeline
 - Audio storage isolation using S3
 - Containerized + Fully deployed on AWS
-

Step 1: Infrastructure Setup (S3 & Networking)

First, you need to set up the **storage and isolation** environment on AWS.

- **Create S3 Bucket:** Provision a dedicated S3 bucket named **tts-service-storage-env-bucket**.
 - **Purpose:** This will store all generated audio files to ensure storage isolation.
 - **Networking:** Ensure the service runs within the VPC to communicate with the Kafka cluster and other services
-

Configure the S3 Bucket

Name	AWS Region	Creation date
amzn-sc3	US East (N. Virginia) us-east-1	December 4, 2025, 14:08:47 (UTC+02:00)
dynamic-docs-bucket	US East (N. Virginia) us-east-1	December 4, 2025, 14:36:44 (UTC+02:00)
dynamic-docs-bucket1	US East (N. Virginia) us-east-1	December 4, 2025, 15:23:28 (UTC+02:00)
my-document-bucket5	US East (N. Virginia) us-east-1	December 4, 2025, 19:44:39 (UTC+02:00)
stt-service-storage-dev-1	US East (N. Virginia) us-east-1	December 5, 2025, 13:43:24 (UTC+02:00)
tts-service-storage-env-bucket	US East (N. Virginia) us-east-1	December 4, 2025, 18:24:06 (UTC+02:00)

Step 2: Microservice Development (Python & Logic)

Develop the core logic using the specified tech stack.

- **Language & Environment:** Use **Python 3.11** as the base runtime.
- **TTS Engine Integration:** Implement the speech synthesis logic using the AWS

Polly SDK, gTTS, or Coqui TTS.

Prerequisite: IAM Permissions

- Before writing the code, the identity (Role) running your Docker container must have permissions to access Polly and S3.
- **Policy Required:** **AmazonPollyFullAccess** and **AmazonS3FullAccess**

(to save the file).

- **API Implementation:** Develop the following **REST** endpoints:
 - POST /api/tts/synthesize: To trigger speech generation.
 - GET /api/tts/audio/{id}: To retrieve the file.
 - DELETE /api/tts/audio/{id}: To remove audio files

IAM Role

ttt-service-role

Summary

ARN: arn:aws:iam::234690396635:role/tts-service-role

Last activity: 1 hour ago

Maximum session duration: 1 hour

Permissions | Trust relationships | Tags | Last Accessed | Revoke sessions

Permissions policies (4)

You can attach up to 10 managed policies.

Policy name	Type	Attached entities
AmazonPollyFullAccess	AWS managed	2
ECRPullImagesPolicy	Customer managed	6
ECSCloudWatchLogsPolicy	Customer managed	6
TTSServiceS3Access	Customer managed	1

Permissions boundary (not set)

APIs

API Gateway

Resources

- /ttt**
- /sts**
- /ts**

Resource details

Path: /api/tts

Resource ID: 8j3djt

Methods (1)

Method type	Integration type	Authorization	API key
OPTIONS	Mock	None	Not required

TTS-POST

Create method

Method details

Method type: POST

Integration type:

- Lambda function: Integrate your API with a Lambda function. (Icon: Lambda)
- HTTP: Integrate with an existing HTTP endpoint. (Icon: HTTP)
- Mock: Generate a response based on API Gateway mappings and transformations. (Icon: Mock)
- AWS service: Integrate with an AWS Service. (Icon: AWS)
- VPC link: Integrate with a resource that isn't accessible over the public internet. (Icon: VPC)

HTTP proxy integration
Send the request to your HTTP endpoint without customizing the integration request or integration response.

Create method

HTTP proxy integration
Send the request to your HTTP endpoint without customizing the integration request or integration response.

Response transfer mode: [Info](#)
 Buffered: Wait to receive the complete response before beginning transmission.
 Stream: Send portions of the response without waiting for the complete response.

HTTP method: POST

Endpoint URL: http://35.175.107.104:5000/api/tts/synthesize

Content handling: [Learn more](#)
 Passthrough

Integration timeout: [Info](#)
By default, you can enter an integration timeout of 50 - 29,000 milliseconds. You can use Service Quotas to raise the integration timeout to greater than 29,000 ms

Method request settings

TTS-GET

Create method

Method details

Method type: GET

Integration type:

- Lambda function: Integrate your API with a Lambda function. (Icon: Lambda)
- HTTP: Integrate with an existing HTTP endpoint. (Icon: HTTP)
- Mock: Generate a response based on API Gateway mappings and transformations. (Icon: Mock)
- AWS service: Integrate with an AWS Service. (Icon: AWS)
- VPC link: Integrate with a resource that isn't accessible over the public internet. (Icon: VPC)

HTTP proxy integration
Send the request to your HTTP endpoint without customizing the integration request or integration response.

Response transfer mode: [Info](#)

The screenshot shows the AWS API Gateway interface. In the left sidebar, under 'APIs', 'Custom domain names', 'Domain name access associations', 'vPC links', and 'AgentCore targets' are listed. Under 'API: learning-platform-api', 'Resources' is selected, showing 'Stages', 'Authorizers', 'Gateway responses', 'Models', 'Resource policy', 'Documentation', 'Dashboard', and 'API settings'. The main panel displays a 'Resources' section with a 'Create resource' button and a tree view of resources. A green banner at the top indicates 'Successfully created method: DELETE' for the '/ttc' endpoint. Below this, the 'Resource details' section shows the path '/api/tts' and a 'Methods (4)' table:

Method type	Integration type	Authorization	API key
DELETE	HTTP	None	Not required
GET	HTTP	None	Not required
OPTIONS	Mock	None	Not required
POST	HTTP	None	Not required

The screenshot shows the 'Enable CORS' configuration page for the '/ttc' endpoint. It includes sections for 'CORS settings' (info), 'Gateway responses' (checkboxes for Default 4XX, Default 5XX), 'Access-Control-Allow-Methods' (checkboxes for DELETE, GET, OPTIONS, POST), 'Access-Control-Allow-Headers' (text input: Content-Type,X-Amz-Date,Authorization,X-Api-Key,X-Amz-Security-Token), and 'Access-Control-Allow-Origin' (text input: Enter an origin that can access the resource. Use a wildcard '*' to allow any origin to access the resource.).

Build Docker Image

Use Python 3.11 base image

Install libraries:

boto3 (Polly/S3)

Kafka client (kafka-python)

Optional: gTTS

Expose port 5000

Push the Docker image to ECR (tts-service-repo) using:

```
docker build -t tts-service .
```

```
docker push <account>.dkr.ecr.<region>.amazonaws.com/tts-service-  
repo
```

```
[ec2-user@ip-172-31-74-1 ~]$ nano main.py
Last login: Fri Dec  5 00:16:20 2025 from 169.150.196.141
[ec2-user@ip-172-31-74-1 ~]$ nano main.py
[ec2-user@ip-172-31-74-1 ~]$ nano main.py
[ec2-user@ip-172-31-74-1 ~]$ nano requirements.txt
[ec2-user@ip-172-31-74-1 ~]$ nano dockerfile
[ec2-user@ip-172-31-74-1 ~]$ docker build -t tts-service .
[!] Building 0.4s (1/1) FINISHED                                            docker:default
[internal] load build definition from dockerfile                         0.0s
=> => transferring dockerfile: 544B                                         0.0s
[internal] load metadata for docker.io/library/python:3.11-slim          0.1s
[internal] load image交代: dockerignore                                    0.0s
[internal] load image交代: Dockerfile                                       0.0s
[internal] load image交代: requirements.txt                                0.0s
[internal] load image交代: Dockerfile                                       0.0s
[internal] load build context                                           0.0s
=> => rendering context: 20.13kB                                         0.0s
=> CACHED [2/6] WORKDIR /app                                           0.0s
=> CACHED [3/6] RUN apt-get update && apt-get install -y ffmpeg && r 0.0s
=> CACHED [4/6] COPY requirements.txt                                     0.0s
=> CACHED [5/6] RUN pip install --no-cache-dir -r requirements.txt    0.0s
[6/6] COPY . .                                                       0.0s
=> exporting to image                                                 0.0s
=> saving layers                                                       0.0s
=> writing image sha256:e8b30adic2fef6ea0bbdb2e28223f358b1b7d9e9004 0.0s
=> naming to docker.io/library/tts-service                            0.0s
[ec2-user@ip-172-31-74-1 ~]$ docker rm -f tts-container
tts-container removed
[ec2-user@ip-172-31-74-1 ~]$ docker run -d -p 5000:5000 \
-e TTS_STORAGE_BUCKET=tts-service-storage-env-bucket \
--name tts-container \
tts-service
4fb900c1fdfe0192dce27d452407b760a0ffce17f6ed3958b94a6cb9bb5db7762e
[ec2-user@ip-172-31-74-1 ~]$ docker logs tts-container
INFO:  Started server process [1]
INFO:  Waiting for application startup.
INFO:  Application startup complete.
INFO:  Uvicorn running on http://0.0.0.0:5000 (Press CTRL+C to quit)
[ec2-user@ip-172-31-74-1 ~]$
```

Push image into ECR

```
Microsoft Windows [Version 10.0.26100.7171]
(c) Microsoft Corporation. All rights reserved.

C:\Users\nour>aws ecr get-login-password --region us-east-1 | docker login --username AWS --password-stdin 234690396635.dkr.ecr.us-east-1.amazonaws.com
Login Succeeded

C:\Users\nour>docker tag tts-service:latest 234690396635.dkr.ecr.us-east-1.amazonaws.com/tts-service-repo:latest

C:\Users\nour>docker push 234690396635.dkr.ecr.us-east-1.amazonaws.com/tts-service-repo:latest
The push refers to repository [234690396635.dkr.ecr.us-east-1.amazonaws.com/tts-service-repo]
7dcfc7f2afab6: Pushed
52a0443eef709: Pushed
88941405be2f: Pushed
c9a903001313: Pushed
171a560c130d: Pushed
22b63e76fdde1: Pushed
1df5296944ef: Pushed
4a81ea28dc6c: Pushed
0e4bc2bd6656: Pushed
b3dd773c3296: Pushed
latest: digest: sha256:c472c3cf1167800850b9d833107cf801ec0cb1216ea2af5d0f0abfc394647ad7 size: 856

C:\Users\nour>
```

TTS image after pushing it into ECR

Amazon ECR > Private registry > Repositories > Images

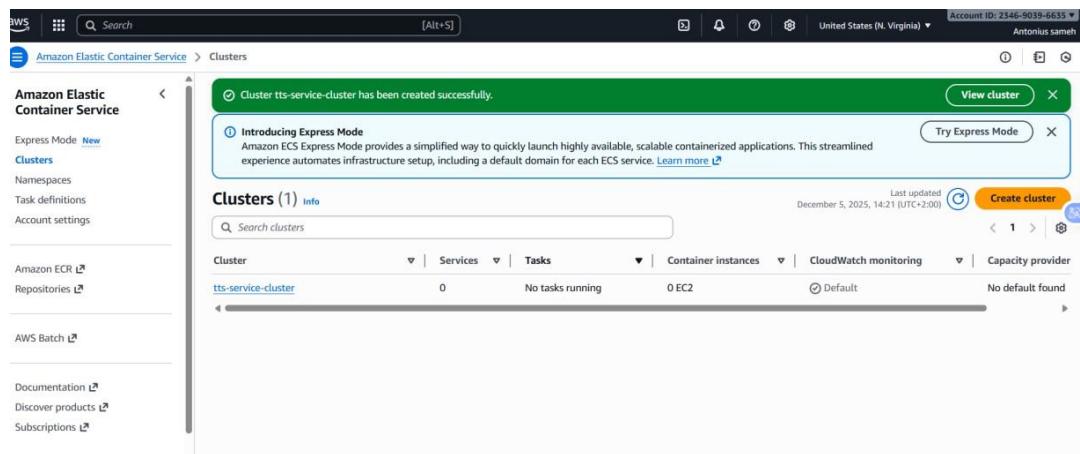
tts-service-repo

Summary **Images**

Images (3)

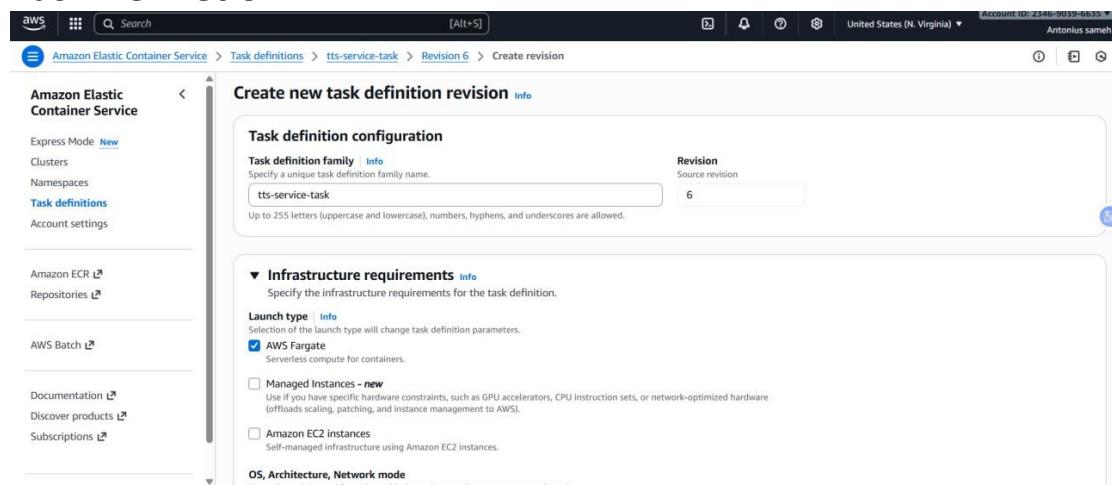
Image tags	Type	Created at	Image size	Image digest	Last pulled at
<input checked="" type="checkbox"/> latest	Image Index	December 05, 2025, 13:53:45 (UTC+02)	314.12	sha25...	-
	Image	December 05, 2025, 13:53:44 (UTC+02)	0.00	sha25...	December 05, 2025, 13:53:51 (UTC+02)
	Image	December 05, 2025, 13:53:44 (UTC+02)	314.12	sha25...	-

Create ECS Cluster



The screenshot shows the AWS Elastic Container Service (ECS) Clusters page. A green success message at the top states: "Cluster tts-service-cluster has been created successfully." Below this, a section titled "Introducing Express Mode" provides information about the streamlined infrastructure setup. The main table lists one cluster, "tts-service-cluster", with details: 0 services, 0 tasks, 0 EC2 container instances, CloudWatch monitoring is off, and the capacity provider is Default. The page also includes a search bar and navigation links for Express Mode, Clusters, Namespaces, Task definitions, Account settings, Amazon ECR, Repositories, AWS Batch, Documentation, Discover products, and Subscriptions.

Task Definition



The screenshot shows the AWS Elastic Container Service (ECS) Task Definitions page. It is navigating through "Task definitions > tts-service-task > Revision 6 > Create revision". The main form is titled "Create new task definition revision" and contains sections for "Task definition configuration" (with a task definition family of "tts-service-task" and a source revision of "6") and "Infrastructure requirements". Under "Launch type", the "AWS Fargate" option is selected. Other options like "Managed Instances - new" and "Amazon EC2 instances" are available but not selected. At the bottom, there is a note about "OS, Architecture, Network mode". The left sidebar mirrors the structure of the previous screenshot, listing Express Mode, Clusters, Namespaces, Task definitions (which is currently selected), Account settings, and other AWS services.

The screenshot shows the AWS Elastic Container Service (ECS) Task Definitions page. The left sidebar includes links for Express Mode, Clusters, Namespaces, Task definitions (which is selected), Account settings, Amazon ECR, Repositories, AWS Batch, Documentation, Discover products, and Subscriptions. The main content area is titled "Amazon Elastic Container Service" and shows a "Task definitions" section for "tts-service-task". It details the task's configuration: "OS, Architecture, Network mode" (Network mode is selected), "Task size" (1 vCPU, 2 GB Memory), and "Task roles - conditional" (Task role: tts-service-role). It also lists "Task execution role" (tts-service-role) and "Task placement - optional" and "Fault injection - optional".

The screenshot shows the AWS Elastic Container Service (ECS) Task Definitions page. The left sidebar includes links for Express Mode, Clusters, Namespaces, Task definitions (selected), Account settings, Amazon ECR, Repositories, AWS Batch, Documentation, Discover products, and Subscriptions. The main content area is titled "Amazon Elastic Container Service" and shows a "Container - 1" section for "tts-service-task". It details the container's configuration: "Container details" (Name: tts-service-container, Essential container: Yes), "Image URI" (234690396635.dkr.ecr.us-east-1.amazonaws.com/tts-service-repo@sha256:c472c3cf1167800850b9d833107cf801), "Private registry" (Private registry authentication selected), "Port mappings" (Container port: 5000, Protocol: TCP, Port name: tts-service, App protocol: HTTP), and "Read only root file system" (Info).

The screenshot shows the AWS ECS Task Definitions interface. On the left, there's a sidebar with navigation links like 'Amazon Elastic Container Service' (selected), 'Task definitions' (underlined), and 'Create revision'. The main content area is titled '(CSE353) Cloud Computing' and shows a 'Task definitions' page for a task named 'tts-service-task'. A sub-section titled 'Revision 6' is selected. The page displays environment variables under the heading 'Environment variables - optional'. It includes sections for 'Environment variables' (with a 'Info' link), 'Add individually' (with a 'Key' dropdown set to 'KAFKA_BOOTSTRAP' and 'Value' 'my-kafka-cluster'), 'Add from file' (with a 'Add environment file' button), and 'Logging - optional' (with a 'Log driver' dropdown set to 'awslogs').

Configure Load Balancer

- Listener port: 5000
- Target group:tts-tg
- Type: IP
- Port: 5000
- Health check: /health
- ECS tasks automatically register as targets.

Load Balancer

The screenshot shows the AWS EC2 Load Balancers interface. The left sidebar includes 'Classic Load Balancer' (selected), 'Network & Security', 'Load Balancing', and 'Auto Scaling'. The main content area shows a 'dev-public-alb' load balancer. A modal window titled 'Introducing token validation of JWTs for ALB' is open, providing information about token validation. Below the modal, the 'Details' section for 'dev-public-alb' is visible, showing details such as 'Load balancer type: Application', 'Status: Active', 'Scheme: Internet-facing', 'Hosted zone: Z355XDOTRQ7X7K', 'VPC: vpc-05d2c059fa6b57964', 'Availability Zones: subnet-056d0000e56b97c3 us-east-1a (use1-az1), subnet-08ff9c7049f7e5b0d us-east-1b (use1-az2)', 'Load balancer IP address type: IPv4', and 'Date created: December 1, 2025, 17:36 (UTC+02:00)'. At the bottom, tabs for 'Listeners and rules', 'Network mapping', 'Resource map', 'Security', 'Monitoring', 'Integrations', 'Attributes', 'Capacity', and 'Tags' are present.

Listener

HTTP:5000 info

Protocol:Port: HTTP:5000

Load balancer: dev-public-alb

Default actions:

- Forward to target group tss-tg: 1 (100%)

Target group stickiness: Off

Listener ARN: arn:aws:elasticloadbalancing:us-east-1:234690396635:listener/app/dev-public-alb/a54fc5c1cfed3ec/afde15abc027794f

Listener rules (1) info

Traffic received by the listener is routed according to the default action and any additional rules. Rules are evaluated in priority order from the lowest value to the highest value.

Priority	Name tag	Conditions (If)	Transforms	Actions (Then)	Actions
----------	----------	-----------------	------------	----------------	---------

Listeners and rules (1/3) info

Protocol:Port: Default action

Protocol:Port	Default action	Rules	ARN	Security policy	Default SSL/TLS certificate
HTTPS-443	Not reachable	0 rules	ARN	ELBSecurityPolicy-TLS13-1-2...	myapi.local (Certificate)
HTTP-80	Not reachable	0 rules	ARN	Not applicable	Not applicable
HTTP-5000	Not reachable	1 rule	ARN	Not applicable	Not applicable

Kafka Integration for TTS Microservice

- Create Kafka Cluster (Amazon MSK) Steps:
- Open Amazon MSK in AWS Console
- Click Create

Cluster Select:

- Cluster Type: Provisioned
- Number of Brokers: 3
- Instance Type: kafka.m5.large
- Storage: 100–200 GB
- Create or choose a VPC, subnets, and security groups
- Wait for cluster status → "Active"

Create Kafka Topics

You need two topics:

Consumer (TTS) :audio.generation.requested to Receive text generation jobs.

Producer (TTS): audio.generation.completed to Send event after audio

is generated

5.2 Speech-to-Text (STT) Service

FastAPI 0.1.0 OAS 3.1

/openapi.json

default ^

POST /api/stt/transcribe Upload And Transcribe

GET /api/stt/transcription/{id} Get Transcription

GET /api/stt/transcriptions List Transcriptions

Schemas ^

Body_upload_and_transcribe_api_stt_transcribe_post > Expand all object

HTTPValidationError > Expand all object

ValidationError > Expand all object

FastAPI 0.1.0 OAS 3.1

/openapi.json

default

POST /api/stt/transcribe Upload And Transcribe

Parameters

No parameters

Request body required

file * required audio.mp3 string(\$binary)

Execute Clear

Responses

Curl

```
curl -X 'POST' \
  'http://3.87.226.61:8000/api/stt/transcribe' \
  -H 'accept: application/json' \
  -H 'Content-Type: multipart/form-data' \
  -F 'file=audio.mp3;type=audio/mpeg'
```

Request URL

http://3.87.226.61:8000/api/stt/transcribe

Server response

Code	Details	Links
200	<p>Response body</p> <pre>{ "id": "3868d8e3-f9d4-4f9d-8f5a-c88f9999bcec", "status": "Transcription Started", "job_name": "transcription-3868d8e3-f9d4-4f9d-8f5a-c88f9999bcec" }</pre> <p>Download</p> <p>Response headers</p> <pre> content-length: 142 content-type: application/json date: Fri, 05 Dec 2025 21:07:14 GMT server: uvicorn</pre>	No links
422	<p>Validation Error</p> <p>Media type</p> <p>application/json</p> <p>Example Value Schema</p> <pre>"string"</pre>	No links

Responses

Code	Description	Links
200	Successful Response	No links

Schemas

Body_upload_and_transcribe_api_stt_transcribe_post > Expand all object

HTTPValidationError > Expand all object

ValidationError > Expand all object

FastAPI

default

/api/v1/transcribe (Default transcript)

Parameters

No parameters

Request body (optional)

File (optional) (Choose File...)

Content-Type: multipart/form-data

Results

Clear

Responses

Code Details

200 Response body

```
[{"id": "1", "text": "Hello, how are you?", "type": "Transcription", "status": "Completed"}, {"id": "2", "text": "I'm fine, thank you.", "type": "Transcription", "status": "Completed"}]
```

Response headers

Content-Length: 102 Content-Type: application/json Date: Fri, 01 Dec 2023 09:00:00 GMT

Requirements

Code Description

200 Successful Response

Content-type: application/json

Example Value: {"transcripts": [{"id": "1", "text": "Hello, how are you?", "type": "Transcription", "status": "Completed"}, {"id": "2", "text": "I'm fine, thank you.", "type": "Transcription", "status": "Completed"}]}

422 Validation Error

Content-type: application/json

Example Value: {"error": "Validation error: file is required"}

Code Details

200 Response body

```
"Hello! How are you?"
```

Response headers

Content-Length: 102 Content-Type: application/json Date: Fri, 01 Dec 2023 09:00:00 GMT

Requirements

Code Details

200 Response body

```
"Hello! How are you?"
```

Response headers

Content-Length: 102 Content-Type: application/json Date: Fri, 01 Dec 2023 09:00:00 GMT

Responses

Code Details

200 Response body

```
"Hello! How are you?"
```

Response headers

Content-Length: 102 Content-Type: application/json Date: Fri, 01 Dec 2023 09:00:00 GMT

Requirements

Code Details

200 Successful Response

Content-type: application/json

Example Value: {"transcripts": [{"id": "1", "text": "Hello, how are you?", "type": "Transcription", "status": "Completed"}, {"id": "2", "text": "I'm fine, thank you.", "type": "Transcription", "status": "Completed"}]}

422 Validation Error

Content-type: application/json

Example Value: {"error": "Validation error: file is required"}

Code Details

200 Response body

```
"Hello! How are you?"
```

Response headers

Content-Length: 102 Content-Type: application/json Date: Fri, 01 Dec 2023 09:00:00 GMT

Responses

Code Details

200 Response body

```
"Hello! How are you?"
```

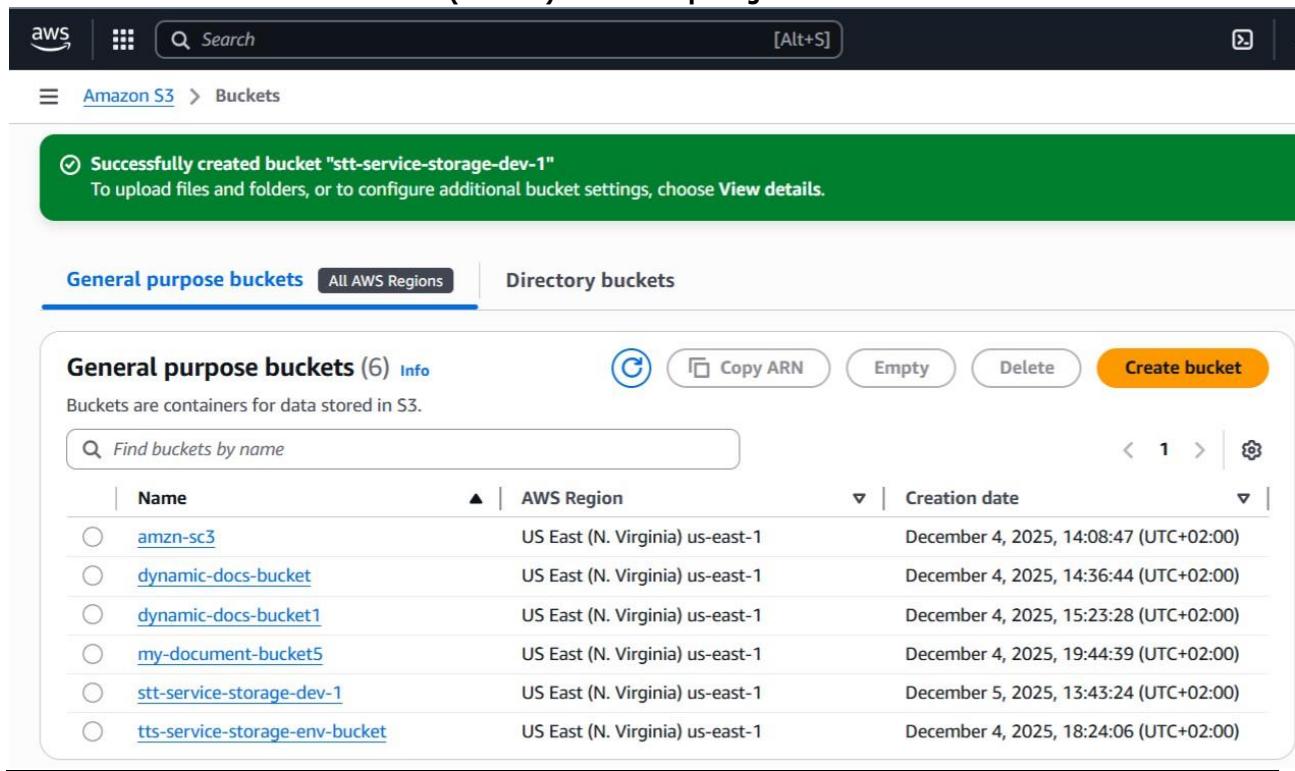
Response headers

Content-Length: 102 Content-Type: application/json Date: Fri, 01 Dec 2023 09:00:00 GMT

Requirements

Bundles

- Body_upload_and_transcribe_api_v1_transcribe_post >
- HTTPValidationError >
- ValidationError >



The screenshot shows the AWS S3 console. At the top, there's a success message: "Successfully created bucket 'stt-service-storage-dev-1'. To upload files and folders, or to configure additional bucket settings, choose View details." Below this, there are two tabs: "General purpose buckets" (selected) and "All AWS Regions". Under "General purpose buckets", there's a sub-tab "Directory buckets". A "Create bucket" button is visible. The main table lists six buckets:

Name	AWS Region	Creation date
amzn-sc3	US East (N. Virginia) us-east-1	December 4, 2025, 14:08:47 (UTC+02:00)
dynamic-docs-bucket	US East (N. Virginia) us-east-1	December 4, 2025, 14:36:44 (UTC+02:00)
dynamic-docs-bucket1	US East (N. Virginia) us-east-1	December 4, 2025, 15:23:28 (UTC+02:00)
my-document-bucket5	US East (N. Virginia) us-east-1	December 4, 2025, 19:44:39 (UTC+02:00)
stt-service-storage-dev-1	US East (N. Virginia) us-east-1	December 5, 2025, 13:43:24 (UTC+02:00)
tts-service-storage-env-bucket	US East (N. Virginia) us-east-1	December 4, 2025, 18:24:06 (UTC+02:00)

5.3 Chat Completion Service

5.4 Document Reader Service

Bedrock Service

1. Accessing the Amazon Bedrock Service

The first step is to locate and navigate to the Amazon Bedrock service within your AWS Management Console.

- **Action:** In the AWS search bar, type bedrock.
- **Result :** The top result will be **Amazon Bedrock**, described as "The easiest way to build and scale generative AI applications with foundation models..."
- **Navigation:** Click on the **Amazon Bedrock** service link to open its main console page.

- **Overview and Initial Setup**

- Once you are on the Amazon Bedrock page, you will see an overview of the service and necessary setup steps.

- The Overview page provides information on **New feature announcements** (like reinforcement fine-tuning), the **Model catalog** (stating Bedrock supports over 100 foundation models), and how to **Get started by using API Keys**.
- **Next Step:** To begin testing models, navigate to the **Test** section in the left-hand menu.

- **3. Navigating to the Text Playground**

- The playground is where you can interact directly with the foundation models.

- **Navigation:** In the left-hand navigation pane, under the **Test** section, click on **Chat / Text playground**.

- **4. Configuring the Text Playground Mode**

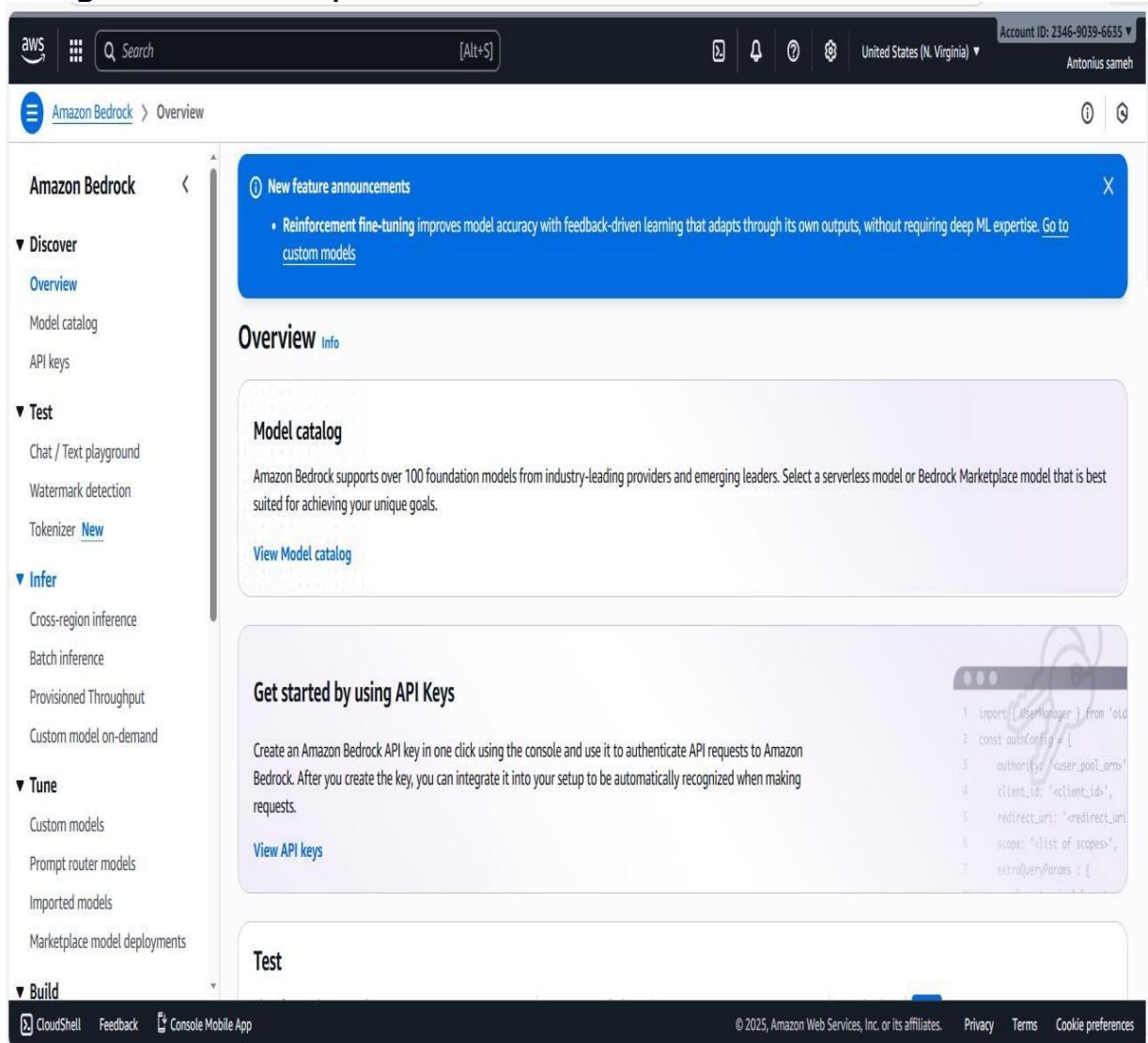
- The Text Playground allows you to choose how you want to interact with the models.

- **Mode Selection :** You have two main modes:

- **Chat:** Generate a conversation by iteratively submitting prompts and replying to each response. (This is selected in the image).

- **Single prompt:** Generate a single response from a model.

- **Recommended Action:** For an interactive conversation, select the Chat mode.
- **5. Selecting a Model to Start Inferencing**
- After setting the mode, the next crucial step is selecting which Foundation Model (FM) you want to use.
 - **Current State :** The playground is ready, but you need to select a model. The prompt reads: **Select a model to get started.**
 - **Action:** Click the **Select model** button at the top of the playground. A list of available models (e.g., from Anthropic, AI21 Labs, Cohere, Amazon) will appear, provided you have requested and been granted **Model access** in the settings.
 - **Final Step:** Once a model is selected, you can type your prompt into the text box at the bottom and click **Run** to generate a response.



The screenshot shows the Amazon Bedrock console interface. At the top, there's a navigation bar with the AWS logo, a search bar, and account information (Account ID: 2346-9039-6635, Antonius sameh). Below the navigation bar, the left sidebar has a tree view with categories like 'Discover' (Overview, Model catalog, API keys), 'Test' (Chat / Text playground, Watermark detection, Tokenizer), 'Infer' (Cross-region inference, Batch inference, Provisioned Throughput, Custom model on-demand), 'Tune' (Custom models, Prompt router models, Imported models, Marketplace model deployments), and 'Build' (CloudShell, Feedback, Console Mobile App). The main content area is titled 'Overview' and contains sections for 'New feature announcements' (about Reinforcement fine-tuning), 'Model catalog' (describing support for over 100 foundation models and linking to View Model catalog), 'Get started by using API Keys' (with a code snippet for creating an API key), and a 'Test' section. The bottom of the screen includes links for CloudShell, Feedback, and Console Mobile App, along with copyright information (© 2023, Amazon Web Services, Inc. or its affiliates.) and links for Privacy, Terms, and Cookie preferences.

The screenshot shows the Amazon Bedrock interface. On the left, there's a sidebar with sections for Discover, Test, Infer, and Tune. Under Test, 'Chat / Text playground' is selected. The main area has a 'Mode' dropdown set to 'Chat'. Under 'Select', 'Chat' is highlighted with a blue border. Below it, 'Single prompt' is listed. At the bottom, there's a large input text area with a 'Run' button.

Selecting the Foundation Model (FM)

After clicking "Select model," a dialog box appears, allowing you to choose the specific Foundation Model you want to use for inference.

- **Model Selection Dialog :** The selection process is broken down into three columns:
 - **Categories (Model Providers):** Here you choose the company that created the model (e.g., AI21 Labs, Amazon, Anthropic, Meta).
 - **Models:** After selecting a provider (e.g., **Meta** is highlighted), you choose the specific model version (e.g., **Llama 3.1 70B Instruct v1** is selected in the image). The token limits are also displayed here (e.g., **128K tokens**).
 - **Inference:** This section typically confirms the chosen inference profile or region (e.g., **US Meta Llama 3.1 70B Instruct**).
- **Action:** Select your desired model and click the **Apply** button to load it into the playground.

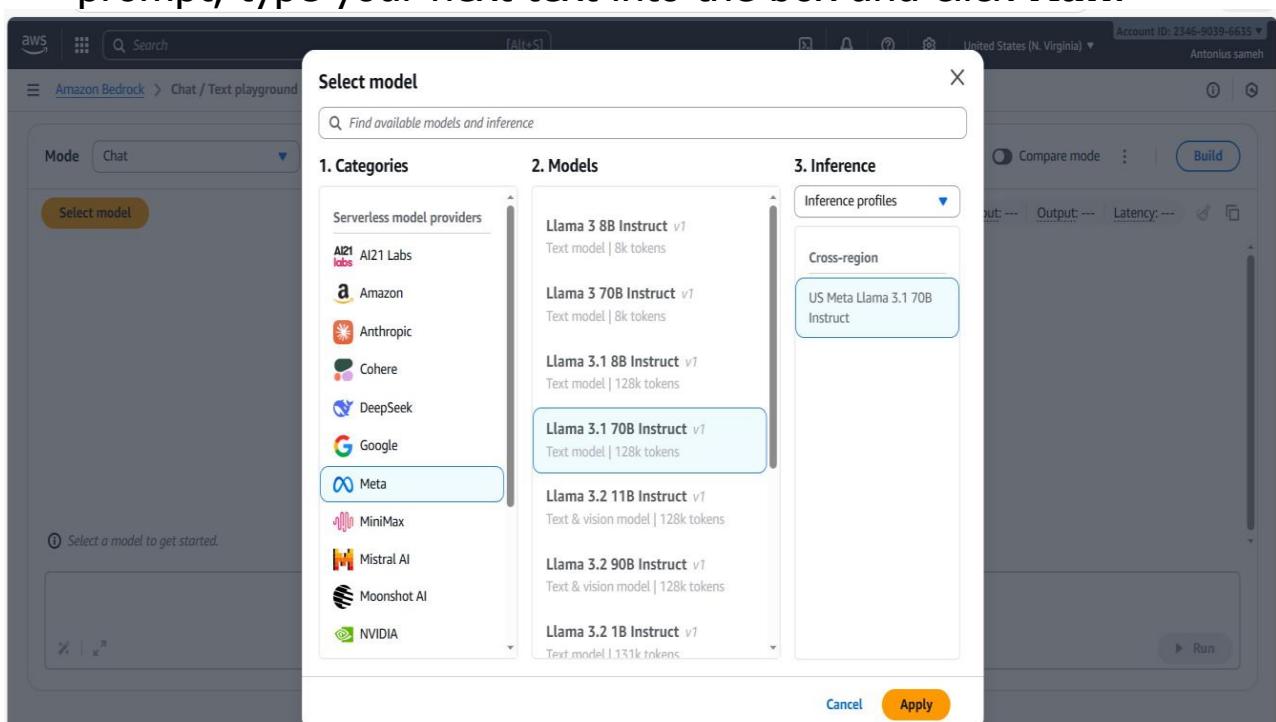
7. Interacting with the Selected Model

Once the model is loaded, the playground is ready for you to send prompts and receive generated output.

- **Loaded Model :** The selected model, **Llama 3.1 70B Instruct v1**, is

now displayed under **Configurations**.

- **Model Configurations:** On the left, you can adjust inference parameters like:
- **System prompts:** To provide context or instructions to guide the model's behavior.
 - **Length (Maximum output tokens):** To set a limit on the length of the model's response (e.g., **512** tokens).
- **Prompting and Generation:**
 - **Input:** The user has provided a prompt: "Write a simple Python function that checks if a number is prime, with explanation."
 - **Output:** The model has generated the requested Python code (def is_prime(n):...) along with the explanation (implied by the presence of the response box).
 - **Metrics:** The top right shows performance metrics: **Input** (30 tokens), **Output** (99 tokens), and **Latency** (3533 ms).
 - **Action:** To continue the conversation or run a new prompt, type your next text into the box and click **Run**.



The screenshot shows the AWS Lambda function configuration interface. On the left, there's a sidebar with 'Configurations' and 'Info' tabs, and a 'System prompts' section containing 'Add system prompts'. Below that is a 'Length' section with a 'Maximum output tokens' slider set to 512. The main area shows a code editor with a Python script for checking if a number is prime:

```

def is_prime(n):
    if n <= 1:
        return False
    elif n <= 3:
        return True
    elif n % 2 == 0 or n % 3 == 0:
        return False
    i = 5
    while i * i <= n:
        ...

```

At the bottom right of the code editor is a yellow 'Run' button. At the top right, there are status metrics: Input: 30, Output: 99, Latency: 3533 ms. The top bar also shows the AWS logo, search bar, and account information: Account ID: 2346-9039-6635, United States (N. Virginia), Antonius sameh.

Building the Docker Image

The service is prepared and built on an EC2 instance (i-05c76a35a0565ff2c).

- **Setup**

- The necessary project files, including a requirements.txt file listing Python dependencies like fastapi, uvicorn, boto3, and PyPDF2, are created.
- A Dockerfile is prepared to define the container environment.

Building the Image :

- The command docker build -t document-reader-service . is executed.
- This command builds the Docker image locally, tagging it as document-reader-service.
- The build steps include loading the Dockerfile, pulling the base Python image (python:3.11-slim), installing dependencies (e.g., gcc, libpq-dev via apt-get), and copying requirements.
- **Result:** The build process completes successfully .

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

Last login: Thu Dec  4 20:44:05 2025 from 18.206.107.27
[ec2-user@ip-10-0-1-55 ~]$ cd document-service
[ec2-user@ip-10-0-1-55 document-service]$ nano requirements.txt
[ec2-user@ip-10-0-1-55 document-service]$ nano requirements.txt
[ec2-user@ip-10-0-1-55 document-service]$ cat requirements.txt
cat: requirements.txt: No such file or directory
[ec2-user@ip-10-0-1-55 document-service]$ nano requirements.txt
[ec2-user@ip-10-0-1-55 document-service]$ cat requirements.txt
fastapi==0.109.0
uvicorn==0.27.0
python-multipart==0.0.6
boto3==1.34.14
psycopg2-binary==2.9.9
kafka-python==2.0.2
PyPDF2==3.0.1
python-docx==1.1.0
openai==1.10.0
python-dotenv==1.0.0
[ec2-user@ip-10-0-1-55 document-service]$ nano Dockerfile
[ec2-user@ip-10-0-1-55 document-service]$ nano Dockerfile
[ec2-user@ip-10-0-1-55 document-service]$ nano Dockerfile
[ec2-user@ip-10-0-1-55 document-service]$ nano .env
[ec2-user@ip-10-0-1-55 document-service]$ nano main.py
[ec2-user@ip-10-0-1-55 document-service]$ docker build -t document-reader-service .
(*) Building 31.3s (11/11) FINISHED
      docker:default          0.0s
-> [internal] load build definition from Dockerfile           0.0s
=> => transferring dockerfile: 40KB                         0.0s
-> [internal] load metadata for docker.io/library/python:3.11-slim   0.3s
=> [internal] load .dockerrcignore                           0.0s
=> => transferring context: 2B                            0.0s
=> [1/6] FROM docker.io/library/python:3.11-slim@sha256:193fd4bbcb3d2ae612bd6cc3548d2f7e78d65b549fcda8a75624c47474444d  3.2s
=> => resolve docker.io/library/python:3.11-slim@sha256:193fd4bbcb3d2ae612bd6cc3548d2f7e78d65b549fcda8a75624c47474444d  0.0s
=> => sha256:193fd4bbcb3d2ae612bd6cc3548d2f7e78d65b549fcda8a75624c47474444d 10.37KB / 10.37KB  0.0s
=> => sha256:c4116d4d7ea9320db352f6516001262753529edf1e20b2c609a6b9a49cc6be4 1.75KB / 1.75KB  0.0s
=> => sha256:e040af38f5bce9a239b7e739e86304d26964d1d55ad56b9297a3d691e91634d 5.48KB / 5.48KB  0.0s
=> => sha256:0e1ebc2bd656e66a004e3c749af70e5650bac225824de0949dea31cb6b7863db 29.78MB / 29.78MB  0.45s
=> => sha256:33d773c329649f22e467ae63d1c612a039a0559dec99fffb9ada904ab5c60c55 14.38MB / 14.38MB  0.2s
=> => sha256:22b63e76fde1e200371ed9f3ce91161d192063beff65c9abfbff3819910a974 1.29MB / 1.29MB  0.1s
=> => sha256:1771569cc1299aa898e762fe441953a721b11a3927ef2f69aa69b0044a88f2da 251B / 251B  0.2s
=> => extracting sha256:0e1ebc2bd656e66a004e3c749af70e5650bac225824de0949dea31cb6b7863db  1.4s
=> => extracting sha256:22b63e76fde1e200371ed9f3ce91161d192063beff65c9abfbff3819910a974  0.1s
=> => extracting sha256:3dd773c329649f22e467ae63d1c612a039a0559dec99fffb9ada904ab5c60c55  1.0s
=> => extracting sha256:1771569cc1299aa898e762fe441953a721b11a3927ef2f69aa69b0044a88f2da  0.0s
=> [internal] load build context  0.0s
=> => transferring context: 4.84KB  0.0s
=> [2/6] WORKDIR /app  0.1s
=> [3/6] RUN apt-get update && apt-get install -y gcc libpq-dev  13.2s
=> [4/6] COPY requirements.txt .  0.1s
```

j-05c76a35a0565ff2c (My-Deploy-Server)

PublicIP: 3.239.149.35 PrivateIP: 10.0.1.55

```
[ec2-user@ip-10-0-1-55 document-service]$ docker stop document-service
document-service
[ec2-user@ip-10-0-1-55 document-service]$ docker rm document-service
document-service
[ec2-user@ip-10-0-1-55 document-service]$ docker rm document-service
Error response from daemon: No such container: document-service
[ec2-user@ip-10-0-1-55 document-service]$ rm main.py
[ec2-user@ip-10-0-1-55 document-service]$ nano main.py
[ec2-user@ip-10-0-1-55 document-service]$ nano main.py
[ec2-user@ip-10-0-1-55 document-service]$ docker build -t document-reader-service .
[*] Building 0.4s (11/11) FINISHED
    => [internal] Load build definition from Dockerfile          docker:default
    => => transferring dockerfile: 406B                         0.0s
    => [internal] load metadata for docker.io/library/python:3.11-slim   0.0s
    => [internal] load .dockerignore                                0.0s
    => => transferring context: 2B                               0.0s
    => [1/6] FROM docker.io/library/python:3.11-slim@sha256:193fad1bb31d2ae112bd6cd334862f7c78d65b549fcaa3a175624c17474444d  0.0s
    => [internal] load build context                            0.0s
    => => transferring context: 7.39kB                          0.0s
    => CACHED [2/6] WORKDIR /app                             0.0s
    => CACHED [3/6] RUN apt-get update && apt-get install -y gcc libpq-dev && rm -rf /var/lib/apt/lists/*  0.0s
    => CACHED [4/6] COPY requirements.txt                     0.0s
    => CACHED [5/6] RUN pip install --no-cache-dir -r requirements.txt  0.0s
    => [6/6] COPY .                                         0.0s
    => exporting to image                                     0.0s
    => => exporting layers                                    0.0s
    => => writing image sha256:5479c39f01027066a2ea257a1d68d4da12b68577d936b63d1fc67d1e03c3175  0.0s
    => => naming to docker.io/library/document-reader-service 0.0s
[ec2-user@ip-10-0-1-55 document-service]$ docker run -d --name document-service -p 8000:8000 --env-file .env document-reader-service
```

i-05c76a35a0565ff2c (My-Deploy-Server)

PublicIPs: 3.239.149.35 PrivateIPs: 10.0.1.55

Running the Container :

- The command docker run --name document-service -p 8000:8000 --env-file .env document-reader-service is used to:
 - Run the image tagged document-reader-service.
 - Map port 8000 on the container to port 8000 on the EC2 host.
 - Set the container name as document-service.
 - Load environment variables from a local .env file.

• Testing the API :

- The command curl <http://localhost:8000/openapi.json> is executed to hit

the service's API documentation endpoint.

- **Result:** The command successfully returns the full OpenAPI

JSON specification, confirming the service is running and accessible.

. Creating the ECR Repository

The ECR repository is a secure location to store the Docker image on AWS.

- **Initial Push Attempt & Failure :**

- An initial attempt to run docker push fails with the error "The repository with name 'document-reader-service' does not exist in the registry..."

- **Creating the Repository :**

- The command `aws ecr create-repository --repository-name document-reader-service --region us-east-1` is executed.
 - **Result:** The repository is successfully created, returning a JSON object confirming the **repositoryUri** (`234690396635.dkr.ecr.us-east-1.amazonaws.com/document-reader-service`).

```
[ec2-user@ip-10-0-1-55 document-service]$ docker push 234690396635.dkr.ecr.us-east-1.amazonaws.com/document-reader-service:latest
The push refers to repository [234690396635.dkr.ecr.us-east-1.amazonaws.com/document-reader-service]
319d4dca9597: Preparing
f260ccb4ff6bf: Preparing
3734a5ac2887: Preparing
5377a646f4a0: Preparing
3464b0b77e61: Preparing
720ee7aae3ad: Waiting
655ff69eb9c8: Waiting
6c988eaa0862: Waiting
70a290c5e58b: Waiting
name unknown: The repository with name 'document-reader-service' does not exist in the registry with id '234690396635'
[ec2-user@ip-10-0-1-55 document-service]$ aws ecr create-repository --repository-name document-reader-service --region us-east-1
```

```
"repository": {  
    "repositoryArn": "arn:aws:ecr:us-east-1:234690396635:repository/document-reader-service",  
    "registryId": "234690396635",  
    "repositoryName": "document-reader-service",  
    "repositoryUri": "234690396635.dkr.ecr.us-east-1.amazonaws.com/document-reader-service",  
    "createdAt": "2025-12-04T21:30:07.733000+00:00",  
    "imageTagMutability": "MUTABLE",  
    "imageScanningConfiguration": {  
        "scanOnPush": false  
    },  
    "encryptionConfiguration": {  
        "encryptionType": "AES256"
```

i-05c76a35a0565ff2c (My-Deploy-Server)

Public IPs: 3.239.149.35 Private IPs: 10.0.1.55

i-05c76a35a0565ff2c (My-Deploy-Server)

Public IPs: 3.239.149.35 Private IPs: 10.0.1.55

```
[ec2-user@ip-10-0-1-55 document-service]$ curl http://localhost:8000/openapi.json | grep "regenerate-notes"
% Total    % Received % Xferd  Average Speed   Time   Time  Current
          Dload  Upload Total Spent   Left Speed
100  3443  100  3443    0      0  1450k  0:--:--:--:--:--:--:-- 1681k
["openapi": "3.1.0", "info": {"title": "Document Reader Service", "version": "0.1.0"}, "paths": {""/api/documents/upload": {"post": {"summary": "Upload Document", "operationId": "upload_document", "parameters": [{"name": "api_documents_upload_post", "in": "path", "required": true}, {"name": "requestBody", "in": "body", "schema": {"$ref": "#/components/schemas/Body_upload_document_api_documents_upload_post"}], "responses": {"200": {"description": "Successful Response", "content": {"application/json": {"schema": {}}}}, "422": {"description": "Validation Error", "content": {"application/json": {"schema": {"$ref": "#/components/schemas/HTTPValidationError"}}}}}, "/api/documents/{doc_id)": {"get": {"summary": "Get Document Detail", "operationId": "get_document_details", "parameters": [{"name": "doc_id", "in": "path", "required": true}, {"name": "schema", "in": "query", "type": "String", "title": "Doc Id"}], "responses": {"200": {"description": "Successful Response", "content": {"application/json": {"schema": {}}}}, "422": {"description": "Validation Error", "content": {"application/json": {"schema": {"$ref": "#/components/schemas/HTTPValidationError"}}}}}, "delete": {"summary": "Delete Document", "operationId": "delete_document", "parameters": [{"name": "doc_id", "in": "path", "required": true}, {"name": "schema", "in": "query", "type": "String", "title": "Doc Id"}], "responses": {"200": {"description": "Successful Response", "content": {"application/json": {"schema": {}}}}, "422": {"description": "Validation Error", "content": {"application/json": {"schema": {"$ref": "#/components/schemas/HTTPValidationError"}}}}}, "/api/documents/{doc_id}/notes": {"get": {"summary": "Get Document Notes", "operationId": "get_document_notes", "parameters": [{"name": "doc_id", "in": "path", "required": true}, {"name": "schema", "in": "query", "type": "String", "title": "Doc Id"}], "responses": {"200": {"description": "Successful Response", "content": {"application/json": {"schema": {}}}}, "422": {"description": "Validation Error", "content": {"application/json": {"schema": {"$ref": "#/components/schemas/HTTPValidationError"}}}}}, "/api/documents/{doc_id}/regenerate-notes": {"post": {"summary": "Regenerate Notes", "operationId": "regenerate_notes", "parameters": [{"name": "doc_id", "in": "path", "required": true}, {"name": "schema", "in": "query", "type": "String", "title": "Doc Id"}], "responses": {"200": {"description": "Successful Response", "content": {"application/json": {"schema": {}}}}, "422": {"description": "Validation Error", "content": {"application/json": {"schema": {"$ref": "#/components/schemas/HTTPValidationError"}}}}}, "/api/documents": {"get": {"summary": "List Documents", "operationId": "list_documents", "parameters": [{"name": "schema", "in": "query", "type": "String", "title": "File"}], "responses": {"200": {"description": "Successful Response", "content": {"application/json": {"schema": {}}}}, "422": {"description": "Validation Error", "content": {"application/json": {"schema": {"$ref": "#/components/schemas/HTTPValidationError"}}}}}, "components": {"schemas": {"Body_upload_document_api_documents_upload_post": {"properties": {"file": {"type": "string", "format": "binary", "title": "File"}}, "type": "object", "required": ["file"]}, "Body_upload_document_api_documents_upload_post": {"properties": {"detail": {"items": {"$ref": "#/components/schemas/ValidationError"}, "type": "array", "title": "Detail"}}, "type": "object", "title": "HTTPValidationError"}, "ValidationError": {"properties": {"loc": {"items": {"anyOf": [{"type": "string"}, {"type": "integer"}]}, "type": "array", "title": "Location"}, "msg": {"type": "string", "title": "Message"}, "type": "string", "title": "Error Type"}}, "object": {"type": "object", "title": "ValidationError"}], "responses": {"200": {"description": "Successful Response", "content": {"application/json": {"schema": {}}}}, "422": {"description": "Validation Error", "content": {"application/json": {"schema": {"$ref": "#/components/schemas/ValidationError"}}}}}
```

i-05c76a35a0565ff2c (My-Deploy-Server)

PublicIPs: 3.239.149.35 PrivateIPs: 10.0.1.55

Logging In and Pushing the Image to ECR

The final steps involve authenticating with ECR and uploading the built image.

- Authentication Issues :
 - An initial attempt to authenticate fails with an AccessDeniedException because the IAM user (noura-kafka-admin) lacks the necessary permissions to call GetAuthorizationToken.
 - Successful Login :
 - A subsequent successful login is shown after the permission issue is presumably resolved: Login Succeeded.
 - A warning about unencrypted passwords in the Docker configuration is displayed, with a link to configure a credential helper.
 - Tagging the Image :
 - The local image is tagged with the ECR repository URI: docker tag document-reader-service:latest 234690396635.dkr.ecr.us-east-1.amazonaws.com/document-reader-service:latest.
 - Pushing the Image :
 - The command docker push 234690396635.dkr.ecr.us-

east- 1.amazonaws.com/document-reader-service:latest is executed.

- **Result:** The layers of the image are uploaded successfully (Preparing, Waiting, and implicitly Pushed once all layers are done).

```

⇒ [3/6] RUN apt-get update && apt-get install -y    gcc      libpq-dev    && rm -rf /var/lib/apt/lists/*
⇒ [4/6] COPY requirements.txt .
⇒ [5/6] RUN pip install --no-cache-dir -r requirements.txt
⇒ [6/6] COPY . .
⇒ exporting to image
⇒ => exporting layers
⇒ => writing image sha256:67fc03336578108a364842897f0ed350dd6c425f92e1ff73982059ed36d26377
⇒ => naming to docker.io/library/document-reader-service
[ec2-user@ip-10-0-1-55 document-service]$ aws ecr get-login-password --region us-east-1 | docker login --username AWS --password-stdin 234690396635.dkr.ecr.us-east-1.amazonaws.com

An error occurred (AccessDeniedException) when calling the GetAuthorizationToken operation: User: arn:aws:iam::234690396635:user/abdullah-kafka-admin is not authorized to perform: ecr:GetAuthorizationToken on resource: * because no identity-based policy allows the ecr:GetAuthorizationToken action
Error: Cannot perform an interactive login from a non TTY device
[ec2-user@ip-10-0-1-55 document-service]$ aws ecr get-login-password --region us-east-1 | docker login --username AWS --password-stdin 234690396635.dkr.ecr.us-east-1.amazonaws.com

WARNING! Your password will be stored unencrypted in /home/ec2-user/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
[ec2-user@ip-10-0-1-55 document-service]$ docker tag document-reader-service:latest 234690396635.dkr.ecr.us-east-1.amazonaws.com/document-reader-service:latest
[ec2-user@ip-10-0-1-55 document-service]$ docker push 234690396635.dkr.ecr.us-east-1.amazonaws.com/document-reader-service:latest
The push refers to repository [234690396635.dkr.ecr.us-east-1.amazonaws.com/document-reader-service]
b19d4dca9597: Preparing
f268ceb4f6bf: Preparing
3734a5ac2887: Preparing
5377a646f4a0: Preparing
8464b0b77e61: Preparing

```

i-05c76a35a0565ff2c (My-Deploy-Server)

PublicIPs: 3.239.149.35 PrivateIPs: 10.0.1.55

Verifying the Image in the AWS Console

The final step is to confirm the image is present in the ECR console.

- **ECR Console View :**
 - The **Private repositories** page shows the document-reader-service repository and its URI.
 - Navigating to the repository details under the **Images** tab confirms the upload.
 - **Verification :** The image with the tag **latest** is listed, showing the creation date (December 04, 2025), size (160.32 MB), and image digest.

The screenshot shows the AWS ECR Private registry Repositories page. On the left sidebar, under 'Private registry', 'Repositories' is selected. The main content area displays a single repository named 'document-reader-service'. A blue banner at the top right says 'Managed signing now available'.

Repository name	URI	Created at	Tag immutability	Encryption type
document-reader-service	234690396635.dkr.ecr.us-east-1.amazonaws.com/document-reader-service	December 04, 2025, 23:30:07 (UTC+02)	Mutable	AES-256

The screenshot shows the AWS ECR Private registry Repository 'document-reader-service' Images page. Under the 'Images' tab, there is one image named 'latest'. The table below provides details about this image.

Image tags	Type	Created at	Image size	Image digest	Last pulled at
latest	Image	December 04, 2025, 23:30:45 (UTC+02)	160.32	sha256:10c6936...	-

User name	Path	Group	Last activity	MFA	Password age	Console last sign-in	Action
abdullah-kafka-admin	/	0	1 hour ago	-	12 hours	-	Act
antonios-quiz-docker-dev	/	0	-	-	12 hours	-	-
developer-0	/	1	-	-	-	-	-
developer-1	/	1	-	-	-	-	-
merna-api-stt-dev	/	0	12 hours ago	-	12 hours	12 hours ago	-
nora-chat-docs-dev	/	0	8 minutes ago	-	12 hours	-	Act
nourhan-tts-dev	/	0	-	-	12 hours	-	-
remonda-k8s-admin	/	0	-	-	12 hours	-	-
terraform-user	/	0	Yesterday	-	-	-	Act

Document Reader Service

/openapi.json

AI-powered document processing service with AWS integration

Health

GET / Welcome endpoint

GET /health Health check

Documents

POST /api/documents/upload Upload a document

GET /api/documents/{doc_id} Get document details

DELETE /api/documents/{doc_id} Delete a document

GET /api/documents/{doc_id}/notes Get generated notes

POST /api/documents/{doc_id}/regenerate-notes Regenerate document notes

GET /api/documents List all documents

<http://localhost:8000/docs>

5.5 Quiz and Exercise Service

The **Quiz and Exercise Service** is a web application designed to generate quizzes and exercises based on input text or predefined content. This service is powered by advanced machine learning models and is typically integrated into educational platforms to assess the understanding of students or users.

Key Benefits of Docker Implementation:

- Isolation:** Each service (e.g., the web application, database, or caching system) can run in its container, ensuring that they do not interfere with each other.
- Consistency:** Docker ensures that the application runs in the same environment across different machines, preventing environment-specific bugs.
- Scalability:** Docker allows for easy scaling of services. For instance, if the quiz service experiences high traffic, more containers can be spun up to handle the load.
- Efficiency:** Docker containers are lightweight, so they consume fewer resources compared to traditional virtual machines.

Docker Compose for Multi-Container Setup:

To manage the various services such as the web application, database, and message brokers, **Docker Compose** is used. This tool helps define and run multi-container Docker applications.

- **docker-compose.yml** defines all the services that need to run in containers.
- **FastAPI Service**: The main service running the quiz generation logic.
- **PostgreSQL Database**: Stores quiz data, questions, and user responses.
- **Kafka**: Used for message streaming (e.g., real-time quiz generation).

FastAPI 0.1.0 OAS 3.1

/openapi.json

default	
GET	/healthz Health
POST	/api/quiz/generate Generate Quiz
GET	/api/quiz/{quiz_id} Get Quiz
DELETE	/api/quiz/{quiz_id} Delete Quiz
POST	/api/quiz/submit Submit Quiz
GET	/api/quiz/{quiz_id}/results Get Results
GET	/api/quiz/history/all Get History

Generate Quiz

POST /api/quiz/generate Generate Quiz

Parameters

No parameters

Request body required

application/json

Edit Value | Schema

```
{
  "notes_text": "AI makes machines learn and think; neural networks help AI learn; ML is about learning from data."
}
```

Execute Clear

Code Details

Response body

```
{
  "message": "Quiz generated successfully",
  "notes_received": "AI makes machines learn and think; neural networks help AI learn; ML is about learning from data.",
  "quiz": [
    {
      "id": 91,
      "type": "mcq",
      "question": "What is AI?",
      "options": [
        "AI",
        "Math",
        "Bio",
        "History"
      ],
      "correct_answer": "AI"
    },
    {
      "id": 92,
      "type": "tf",
      "question": "AI uses neural networks.",
      "answer": true
    },
    {
      "id": 93,
      "type": "short",
      "question": "Define ML.",
      "expected_answer": "Machine learning."
    }
  ]
}
```

Download

Get Quiz

GET /api/quiz/{quiz_id} Get Quiz

Parameters

Name	Description
quiz_id * required	integer (path) 39

Responses

200 Response body

```
{
  "quiz_id": 39,
  "notes_text": "AI makes machines learn and think; neural networks help AI learn; ML is about learning from data.",
  "quiz": [
    {
      "id": 91,
      "type": "mcq",
      "question": "What is AI?",
      "options": [
        "AI",
        "Math",
        "Bio",
        "History"
      ],
      "correct_answer": "AI"
    },
    {
      "id": 92,
      "type": "tf",
      "question": "AI uses neural networks.",
      "answer": true
    },
    {
      "id": 93,
      "type": "short",
      "question": "Define ML.",
      "expected_answer": "Machine learning."
    }
  ]
}
```

Download

delete quiz

DELETE /api/quiz/{quiz_id} Delete Quiz

Parameters

Name	Description
quiz_id * required	integer (path) quiz_id

Responses

Submit Quiz

POST /api/quiz/submit Submit Quiz

Parameters

No parameters

Request body required

application/json

Edit Value | Schema

```
{
  "quiz_id": 39,
  "answers": {
    "91": "AI",
    "92": "true",
    "93": "Machine learning."
  }
}
```

Execute Clear

http://127.0.0.1:8000/api/quiz/submit

Server response

Code	Details
200	Response body

```
{
  "quiz_id": 39,
  "score": 3,
  "total_questions": 3,
  "details": [
    {
      "question_id": 91,
      "correct_answer": "AI",
      "user_answer": "AI",
      "is_correct": true
    },
    {
      "question_id": 92,
      "correct_answer": "True",
      "user_answer": "true",
      "is_correct": true
    },
    {
      "question_id": 93,
      "correct_answer": "Machine learning.",
      "user_answer": "Machine learning.",
      "is_correct": true
    }
  ]
}
```

Get Results

GET /api/quiz/{quiz_id}/results Get Results

Parameters

Name	Description
quiz_id * required	integer (path)

Responses

Code	Details
200	Response body

```
{
  "quiz_id": 39,
  "score": 3,
  "total_questions": 3,
  "details": [
    {
      "question_id": 91,
      "correct_answer": "AI",
      "user_answer": "AI",
      "is_correct": true
    },
    {
      "question_id": 92,
      "correct_answer": "True",
      "user_answer": "true",
      "is_correct": true
    },
    {
      "question_id": 93,
      "correct_answer": "Machine learning.",
      "user_answer": "Machine learning.",
      "is_correct": true
    }
  ]
}
```

Get History

Code Details

Code	Details
200	Response body

```
{
  "history": [
    {
      "submission_id": 8,
      "quiz_id": 39,
      "score": 3
    },
    {
      "submission_id": 7,
      "quiz_id": 38,
      "score": 3
    },
    {
      "submission_id": 6,
      "quiz_id": 31,
      "score": 3
    },
    {
      "submission_id": 5,
      "quiz_id": 28,
      "score": 3
    },
    {
      "submission_id": 4,
      "quiz_id": 27,
      "score": 3
    }
  ]
}
```

Response headers

```
content-length: 355
content-type: application/json
date: Fri, 08 Dec 2025 21:13:08 GMT
server: uvicorn
```


Docker is a platform that allows developers to package applications and dependencies into isolated containers, making it easy to develop, ship, and run applications in any environment. Implementing **Docker** for the **Quiz and Exercise Service** ensures a smooth and efficient deployment process.

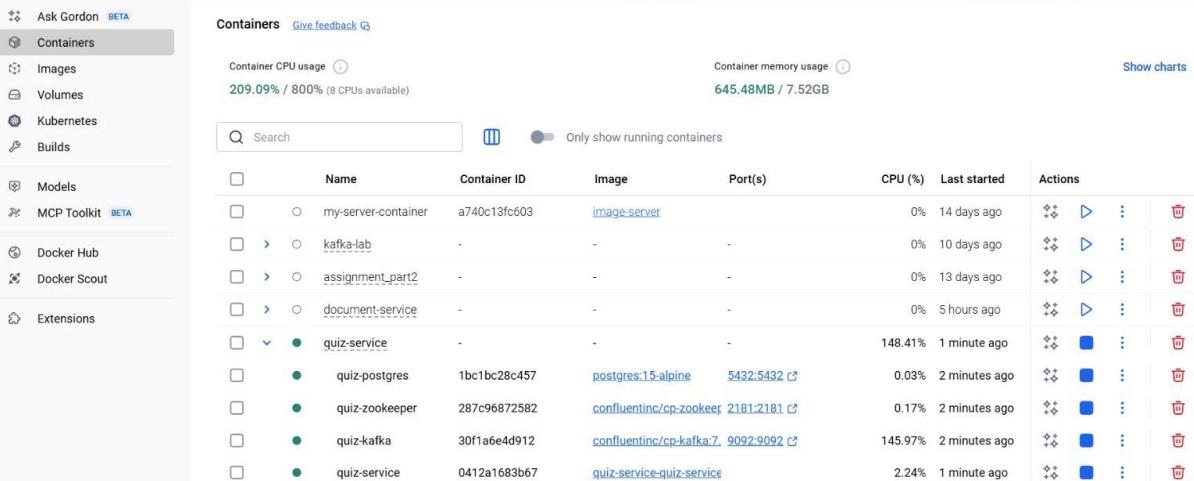
Key Benefits of Docker Implementation:

1. **Isolation:** Each service (e.g., the web application, database, or caching system) can run in its container, ensuring that they do not interfere with each other.
2. **Consistency:** Docker ensures that the application runs in the same environment across different machines, preventing environment-specific bugs.
3. **Scalability:** Docker allows for easy scaling of services. For instance, if the quiz service experiences high traffic, more containers can be spun up to handle the load.
4. **Efficiency:** Docker containers are lightweight, so they consume fewer resources compared to traditional virtual machines.

Docker Compose for Multi-Container Setup:

To manage the various services such as the web application, database, and message brokers, **Docker Compose** is used. This tool helps define and run multi-container Docker applications.

- **docker-compose.yml** defines all the services that need to run in containers.
- **FastAPI Service:** The main service running the quiz generation logic.
- **PostgreSQL Database:** Stores quiz data, questions, and user responses.
- **Kafka:** Used for message streaming (e.g., real-time quiz generation).



6. Containerization Requirements

6.1 Docker Implementation

6.2 Container Orchestration

6.3 Container Registry

6. Containerization and Orchestration Requirements

This section documents the implementation of containerization strategies using Docker and the orchestration of microservices using Kubernetes (K8s) manifests. The architecture ensures scalability, self-healing, and efficient resource management.

6.3 Kubernetes Orchestration

The production environment utilizes Kubernetes manifests to define the desired state of the system, covering deployments, networking, auto-scaling, and storage.

A. Workload Management (Deployments)

The Deployment manifests define the application workloads, including replica counts, resource quotas (CPU/Memory), and health checks (Liveness/Readiness probes).

- **api-gateway-deployment.yaml**

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: api-gateway-deployment
  labels:
    app: api-gateway
spec:
  replicas: 3
  strategy:
    type: RollingUpdate
  selector:
    matchLabels:
      app: api-gateway
  template:
    metadata:
      labels:
        app: api-gateway
    spec:
      containers:
        - name: api-gateway-container
          image: 234690396635.dkr.ecr.us-east-1.amazonaws.com/api-gateway-repo:latest
          ports:
            - containerPort: 8080

      resources:
        requests:
          cpu: "500m"
          memory: "1Gi"
        limits:
          cpu: "500m"
          memory: "1Gi"

      livenessProbe:
        httpGet:
          path: /health
```

```

    port: 8080
    initialDelaySeconds: 15
    timeoutSeconds: 5
  readinessProbe:
    httpGet:
      path: /ready
      port: 8080
    initialDelaySeconds: 5
    timeoutSeconds: 3
  
```

- **chat-service-deployment.yaml**

```

apiVersion: apps/v1
kind: Deployment
metadata:
  name: chat-service-deployment
  labels:
    app: chat-service
spec:
  replicas: 2
  strategy:
    type: RollingUpdate
  selector:
    matchLabels:
      app: chat-service
  template:
    metadata:
      labels:
        app: chat-service
    spec:
      containers:
        - name: chat-service-container
          image: 234690396635.dkr.ecr.us-east-1.amazonaws.com/chat-service-repo:latest
          ports:
            - containerPort: 8080
          resources:
            requests:
              cpu: "2"
              memory: "4Gi"
            limits:
              cpu: "2"
              memory: "4Gi"
          livenessProbe:
            httpGet:
              path: /health
              port: 8080
            initialDelaySeconds: 15
            timeoutSeconds: 5
          readinessProbe:
            httpGet:
              path: /ready
              port: 8080
            initialDelaySeconds: 5
            timeoutSeconds: 3
  
```

```

volumeMounts:
- name: db-persistent-storage
  mountPath: /app/data/db
volumes:
- name: db-persistent-storage
  persistentVolumeClaim:
    claimName: chat-service-pvc

```

- **quiz-service-deployment.yaml**

```

apiVersion: apps/v1
kind: Deployment
metadata:
  name: quiz-service-deployment
  labels:
    app: quiz-service
spec:
  replicas: 2
  strategy:
    type: RollingUpdate
  selector:
    matchLabels:
      app: quiz-service
  template:
    metadata:
      labels:
        app: quiz-service
    spec:
      containers:
        - name: quiz-service-container
          image: 234690396635.dkr.ecr.us-east-1.amazonaws.com/quiz-service-repo:latest
          ports:
            - containerPort: 8080
          resources:
            requests:
              cpu: "1"
              memory: "2Gi"
            limits:
              cpu: "1"
              memory: "2Gi"
      livenessProbe:
        httpGet:
          path: /health
          port: 8080
        initialDelaySeconds: 15
        timeoutSeconds: 5
      readinessProbe:
        httpGet:
          path: /ready
          port: 8080
        initialDelaySeconds: 5
        timeoutSeconds: 3
      volumeMounts:
        - name: db-persistent-storage

```

mountPath: /app/data/quiz

volumes:

- name: db-persistent-storage
- persistentVolumeClaim:
- claimName: quiz-service-pvc

o **document-reader-deployment.yaml**

```

apiVersion: apps/v1
kind: Deployment
metadata:
  name: document-reader-deployment
  labels:
    app: document-reader-service
spec:
  replicas: 2
  strategy:
    type: RollingUpdate
  selector:
    matchLabels:
      app: document-reader-service
  template:
    metadata:
      labels:
        app: document-reader-service
    spec:
      containers:
        - name: document-reader-container
          image: 234690396635.dkr.ecr.us-east-1.amazonaws.com/document-reader-repo:latest
          ports:
            - containerPort: 8080
          resources:
            requests:
              cpu: "1"
              memory: "2Gi"
            limits:
              cpu: "1"
              memory: "2Gi"
      livenessProbe:
        httpGet:
          path: /health
          port: 8080
        initialDelaySeconds: 15
        timeoutSeconds: 5
      readinessProbe:
        httpGet:
          path: /ready
          port: 8080
        initialDelaySeconds: 5
        timeoutSeconds: 3
      volumeMounts:
        - name: db-persistent-storage

```

```

mountPath: /app/data/docs
volumes:
- name: db-persistent-storage
  persistentVolumeClaim:
    claimName: document-reader-pvc

```

- o **stt-service-deployment.yaml**

```

apiVersion: apps/v1
kind: Deployment
metadata:
  name: stt-service-deployment
  labels:
    app: stt-service
spec:
  replicas: 2
  strategy:
    type: RollingUpdate
  selector:
    matchLabels:
      app: stt-service
  template:
    metadata:
      labels:
        app: stt-service
    spec:
      containers:
        - name: stt-service-container
          image: 234690396635.dkr.ecr.us-east-1.amazonaws.com/stt-service-repo:latest
          ports:
            - containerPort: 8080
          resources:
            requests:
              cpu: "2"
              memory: "4Gi"
            limits:
              cpu: "2"
              memory: "4Gi"
          livenessProbe:
            httpGet:
              path: /health
              port: 8080
            initialDelaySeconds: 15
            timeoutSeconds: 5
          readinessProbe:
            httpGet:
              path: /ready
              port: 8080
            initialDelaySeconds: 5
            timeoutSeconds: 3

```

- o **tts-service-deployment.yaml**

```
apiVersion: apps/v1
```

```

kind: Deployment
metadata:
  name: tts-service-deployment
  labels:
    app: tts-service
spec:
  replicas: 2
  strategy:
    type: RollingUpdate
  selector:
    matchLabels:
      app: tts-service
  template:
    metadata:
      labels:
        app: tts-service
    spec:
      containers:
        - name: tts-service-container
          image: 234690396635.dkr.ecr.us-east-1.amazonaws.com/tts-service-repo:latest
          ports:
            - containerPort: 8080

      resources:
        requests:
          cpu: "1"
          memory: "2Gi"
        limits:
          cpu: "1"
          memory: "2Gi"
      livenessProbe:
        httpGet:
          path: /health
          port: 8080
        initialDelaySeconds: 15
        timeoutSeconds: 5
      readinessProbe:
        httpGet:
          path: /ready
          port: 8080
        initialDelaySeconds: 5
        timeoutSeconds: 3

```

B. Service Discovery & Networking

Kubernetes Services are configured to expose pods internally or externally.

- Internal Communication: The document-reader-service uses ClusterIP for secure internal communication.
- External Access: The api-gateway-service uses LoadBalancer to expose the application to the internet.
 - **document-reader-service.yaml**

```
apiVersion: v1
kind: Service
metadata:
  name: document-reader-service
  labels:
    app: document-reader-service
spec:
  type: ClusterIP
  selector:
    app: document-reader-service
  ports:
    - protocol: TCP
      port: 8080
      targetPort: 8080
```

- **quiz-service-service.yaml**

```
apiVersion: v1
kind: Service
metadata:
  name: quiz-service
  labels:
    app: quiz-service
spec:
  type: ClusterIP
  selector:
    app: quiz-service
  ports:
    - protocol: TCP
      port: 8080
      targetPort: 8080
```

- **api-gateway-service.yaml**

```
apiVersion: v1
kind: Service
metadata:
  name: api-gateway-service
  labels:
    app: api-gateway
spec:
  type: LoadBalancer
  selector:
    app: api-gateway
  ports:
    - protocol: TCP
      port: 80
      targetPort: 8080
```

- **stt-service-service.yaml**

```
apiVersion: v1
kind: Service
```

```
metadata:
  name: stt-service
  labels:
    app: stt-service
spec:
```

```
  type: ClusterIP
  selector:
```

```
    app: stt-service
  ports:
    - protocol: TCP
      port: 8080
      targetPort: 8080
```

- **tts-service-service.yaml**

```
apiVersion: v1
kind: Service
metadata:
  name: tts-service
  labels:
    app: tts-service
spec:
  type: ClusterIP
  selector:
    app: tts-service
  ports:
    - protocol: TCP
      port: 8080
      targetPort: 8080
```

- **chat-service-service.yaml**

```
apiVersion: v1
kind: Service
metadata:
  name: chat-service
  labels:
    app: chat-service
spec:
  type: ClusterIP
  selector:
    app: chat-service
  ports:
    - protocol: TCP
      port: 8080
      targetPort: 8080
```

- **Auto-scaling Strategies (HPA)**

Horizontal Pod Autoscalers (HPA) are implemented to automatically adjust the number of pods based on CPU utilization. This ensures the system handles traffic spikes efficiently.

- **chat-service-hpa.yaml**

```
apiVersion: autoscaling/v2
kind: HorizontalPodAutoscaler
metadata:
  name: chat-service-hpa
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: chat-service-deployment
  minReplicas: 2
  maxReplicas: 10
  metrics:
    - type: Resource
      resource:
        name: cpu
        target:
          type: Utilization
          averageUtilization: 75
```

- **document-reader-hpa.yaml**

```
apiVersion: autoscaling/v2
kind: HorizontalPodAutoscaler
metadata:
  name: document-reader-hpa
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: document-reader-deployment
  minReplicas: 2
  maxReplicas: 10
  metrics:
    - type: Resource
      resource:
        name: cpu
        target:
          type: Utilization
          averageUtilization: 75
```

- **tts-service-hpa.yaml**

```
apiVersion: autoscaling/v2
kind: HorizontalPodAutoscaler
metadata:
  name: tts-service-hpa
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: tts-service-deployment
  minReplicas: 2
  maxReplicas: 10
```

metrics:

- type: Resource

resource:

 name: cpu

 target:

 type: Utilization

 averageUtilization: 75

○ **quiz-service-hpa.yaml**

```
apiVersion: autoscaling/v2
kind: HorizontalPodAutoscaler
metadata:
  name: quiz-service-hpa
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: quiz-service-deployment
  minReplicas: 2
  maxReplicas: 10
  metrics:
    - type: Resource
      resource:
        name: cpu
        target:
          type: Utilization
          averageUtilization: 75
```

○ **stt-service-hpa.yaml**

```
apiVersion: autoscaling/v2
kind: HorizontalPodAutoscaler
metadata:
  name: stt-service-hpa
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: stt-service-deployment
  minReplicas: 2
  maxReplicas: 10
  metrics:
    - type: Resource
      resource:
        name: cpu
        target:
          type: Utilization
          averageUtilization: 75
```

○ **api-gateway-hpa.yaml**

```
apiVersion: autoscaling/v2
kind: HorizontalPodAutoscaler
```

```

metadata:
  name: api-gateway-hpa
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: api-gateway-deployment
  minReplicas: 3
  maxReplicas: 15
  metrics:
    - type: Resource
      resource:
        name: cpu
        target:
          type: Utilization
          averageUtilization: 75

```

- **Persistent Storage (State Management)**

For services requiring data persistence (like the Chat Service which may cache vector data locally), Persistent Volume Claims (PVC) are used.

- **chat-service-pvc.yaml**

```

apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: chat-service-pvc
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 5Gi

```

- **document-reader-pvc.yaml**

```

apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: document-reader-pvc
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 5Gi
  # storageClassName: "standard"

```

- **quiz-service-pvc.yaml**

```

apiVersion: v1
kind: PersistentVolumeClaim

```

metadata:

name: quiz-

service-pvc spec:

accessModes:

-
ReadWrite

Once

resources:

requests:

storage: 3Gi

storageClassName: "standard"

Phase 3

7.1 Storage Isolation Principles

S3 Isolation:

- Separate S3 bucket per service

Name	AWS Region	Creation date
chat-service-storage-dev-976289921590	US East (N. Virginia) us-east-1	December 20, 2025, 19:44:01 (UTC+02:00)
document-reader-storage-dev-976289921590	US East (N. Virginia) us-east-1	December 20, 2025, 19:44:00 (UTC+02:00)
quiz-service-storage-dev-976289921590	US East (N. Virginia) us-east-1	December 20, 2025, 19:44:01 (UTC+02:00)
shared-assets-dev-976289921590	US East (N. Virginia) us-east-1	December 20, 2025, 19:44:00 (UTC+02:00)
stt-service-storage-dev-976289921590	US East (N. Virginia) us-east-1	December 20, 2025, 19:44:00 (UTC+02:00)
tts-service-storage-dev-976289921590	US East (N. Virginia) us-east-1	December 20, 2025, 19:44:00 (UTC+02:00)

- Bucket policies enforce service-specific IAM role access only

Policy name	Type	Description
document-cloud-watc...	Customer man...	-
document-s3-policy	Customer man...	-
document-texttract-po...	Customer man...	-
EC2AccessS3RDSPolicy	Customer man...	Permis...
ForceMFA	Customer man...	Permis...
LambdaServicePolicy	Customer man...	Permis...
quiz-s3-policy	Customer man...	None
stt-s3-policy	Customer man...	None
stt-transcribe-policy	Customer man...	None
tts-cloudwatch-policy	Customer man...	None

- Created IAM TTS Policies

Policy name	Type	Description
tts-cloudwatch-policy	Customer managed	-
tts-polly-policy	Customer managed	-
tts-s3-policy	Customer managed	-

- Created IAM Quiz Policies

The screenshot shows the 'Add permissions' step of the IAM role creation wizard. On the left, a vertical navigation bar lists three steps: 'Select trusted entity', 'Add permissions' (which is currently selected), and 'Name, review, and create'. The main area displays a table titled 'Permissions policies (1/1123)' with one item listed: 'quiz-s3-policy'. A search bar at the top of the table allows filtering by policy name. Below the table, there's a section for setting a 'permissions boundary' with the note 'optional'. At the bottom right, there are 'Cancel', 'Previous', and 'Next' buttons.

This screenshot shows the Windows taskbar with several application icons visible, including the AWS Lambda console, Google Chrome, and other productivity tools. The system tray shows the date and time as 12:38 AM on 12/21/2025.

- Created IAM Document Policies

This screenshot shows the 'Add permissions' step of the IAM role creation wizard, similar to the previous one but with three policies selected: 'document-cloud-watch-policy', 'document-s3-policy', and 'document-text-extract-policy'. The interface includes a search bar, filter options, and a note about optional permissions boundaries. The taskbar at the bottom remains consistent with the previous screenshot.

- Created IAM STT Policies

Current permissions policies (0)

Other permissions policies (2/1123)

Policy name	Type	Description
stt-s3-policy	Customer managed	-
stt-transcribe-policy	Customer managed	-

Add permissions

- Created IAM Chat Policies

Current permissions policies (0)

Other permissions policies (1/1124)

Policy name	Type	Description
chat-policy	Customer managed	-
SageMakerStudioBedrockChatAgentUserRolePolicy	AWS managed	Provides access to an Amazon Bedrock c...

Add permissions

- Separate encryption keys per bucket (KMS)

The screenshot shows the AWS KMS console interface. On the left, there's a navigation sidebar with 'Key Management Service (KMS)' selected. Under 'Customer managed keys', five keys are listed:

Aliases	Key ID	Status	Key type	Key spec
chat-s3-kms-key	2b25aaa1-6252-406a-bf7b...	Enabled	Symmetric	SYMMETRIC_DEFAULT
tts-s3-kms-key	c6af3ae8-0aab-4fc6-a970-2...	Enabled	Symmetric	SYMMETRIC_DEFAULT
stt-s3-key	78c1bbc0-7837-4d9b-a652...	Enabled	Symmetric	SYMMETRIC_DEFAULT
quiz-s3-key	b42f589b-9108-4337-be88...	Enabled	Symmetric	SYMMETRIC_DEFAULT
document-s3-key	d4f0e962-6976-47ea-9ef1-2...	Enabled	Symmetric	SYMMETRIC_DEFAULT

At the bottom of the page, there are links for 'CloudShell', 'Feedback', and 'Console Mobile App'. The status bar at the bottom right shows '12:02 AM 13/21/2025'.

- TTS Bucket Policy

The screenshot shows the AWS S3 console. In the left sidebar, under 'Buckets', the 'General purpose buckets' section is expanded, showing 'Directory buckets', 'Table buckets', and 'Vector buckets'. The 'Edit bucket policy - S3 bucket t...' tab is active. The policy document is displayed as JSON code:

```

1  {
2    "Version": "2012-10-17",
3    "Statement": [
4      {
5        "Sid": "AllowTTSRoleAccess",
6        "Effect": "Allow",
7        "Principal": [
8          "AWS: "arn:aws:iam:976289921590:role/tts-service-role"
9        ],
10       "Action": [
11         "s3:GetObject",
12         "s3:PutObject",
13         "s3:DeleteObject",
14         "s3:ListBucket"
15       ],
16       "Resource": [
17         "arn:aws:s3:::tts-service-bucket",
18         "arn:aws:s3:::tts-service-bucket/*"
19       ],
20       "Condition": [
21         "StringEquals": {
22           "aws:SourceArn": "arn:aws:kms:REGION:ACCOUNT_ID:key/TTS_KMS_KEY_ID"
23         }
24       ]
25     }
26   ]

```

To the right of the policy editor, there are two sections: 'Edit statement' and 'Select a statement'. The 'Edit statement' section contains a text input field and a 'Save' button. The 'Select a statement' section has a note: 'Select an existing statement in the policy or add a new statement.' and a 'Add new statement' button.

At the bottom of the page, there are links for 'CloudShell', 'Feedback', and 'Console Mobile App'. The status bar at the bottom right shows '10:30 AM 13/21/2025'.

- STT Bucket Policy

```

1  {
2    "Version": "2012-10-17",
3    "Statement": [
4      {
5        "Sid": "AllowSTTRoleAccess",
6        "Effect": "Allow",
7        "Principal": {
8          "AWS": "arn:aws:iam::976289921590:role/stt-service-role"
9        },
10       "Action": [
11         "s3:GetObject",
12         "s3:PutObject",
13         "s3:DeleteObject",
14         "s3>ListBucket"
15       ],
16       "Resource": [
17         "arn:aws:s3:::stt-service-storage-dev-976289921590",
18         "arn:aws:s3:::stt-service-storage-dev-976289921590/*"
19       ],
20       "Condition": {
21         "StringEquals": {
22           "aws:SourceArn": "arn:aws:kms:us-east-1:976289921590:key/78c1bbc0-7837-4d9b-a652-e962d58fa98b"
23         }
24       }
25     ]
26   }
27 }
28

```

- Quiz Bucket Policy

```

1  {
2    "Version": "2012-10-17",
3    "Statement": [
4      {
5        "Sid": "AllowQuizRoleAccess",
6        "Effect": "Allow",
7        "Principal": {
8          "AWS": "arn:aws:iam::976289921590:role/quiz-service-role"
9        },
10       "Action": [
11         "s3:GetObject",
12         "s3:PutObject",
13         "s3:DeleteObject",
14         "s3>ListBucket"
15       ],
16       "Resource": [
17         "arn:aws:s3:::quiz-service-storage-dev-976289921590",
18         "arn:aws:s3:::quiz-service-storage-dev-976289921590/*"
19       ],
20       "Condition": {
21         "StringEquals": {
22           "aws:SourceArn": "arn:aws:kms:us-east-1:976289921590:key/b42f589b-9108-4337-be88-72338154d6c2"
23         }
24       }
25     ]
26   }
27 }
28

```

- Document Bucket Policy

```

1▼ {
2  "Version": "2012-10-17",
3  "Statement": [
4    {
5      "Sid": "AllowDocumentRoleAccess",
6      "Effect": "Allow",
7      "Principal": {
8        "AWS": "arn:aws:iam::976289921590:role/document-service-role"
9      },
10     "Action": [
11       "s3:GetObject",
12       "s3:PutObject",
13       "s3:DeleteObject",
14       "s3>ListBucket"
15     ],
16     "Resource": [
17       "arn:aws:s3:::document-service-bucket",
18       "arn:aws:s3:::document-service-bucket/*"
19     ],
20     "Condition": {
21       "StringEquals": {
22         "aws:SourceArn": "arn:aws:kms:us-east-1:976289921590:key/d4f0e962-6976-47ea-9ef1-25864fcda2aa"
23       }
24     }
25   }
26 ]
27 }
28

```

- Chat Bucket Policy

```

1▼ {
2  "Version": "2012-10-17",
3  "Statement": [
4    {
5      "Sid": "AllowInternalAccountAccess",
6      "Effect": "Allow",
7      "Principal": {
8        "AWS": "arn:aws:iam::976289921590:role/chat-service-role"
9      },
10     "Action": [
11       "s3:GetObject",
12       "s3:PutObject",
13       "s3:DeleteObject",
14       "s3>ListBucket"
15     ],
16     "Resource": [
17       "arn:aws:s3:::chat-service-storage-dev-976289921590",
18       "arn:aws:s3:::chat-service-storage-dev-976289921590/*"
19     ],
20     "Condition": {
21       "StringEquals": {
22         "aws:SourceArn": "arn:aws:kms:REGION:ACCOUNT_ID:key/DOCUMENT_KMS_KEY_ID"
23       }
24     }
25   }
26 ]
27 }
28

```

Database Isolation:

Databases (4)

DB identifier	Status	Role	Engine	Upgrade rollout order	Region ...	Size
chat-service-db	Available	Instance	PostgreSQL	SECOND	us-east-1b	db.t3.micro
document-reader-db	Available	Instance	PostgreSQL	SECOND	us-east-1a	db.t3.micro
quiz-service-db	Available	Instance	PostgreSQL	SECOND	us-east-1b	db.t3.micro
user-management-db	Available	Instance	PostgreSQL	SECOND	us-east-1b	db.t3.micro

General purpose buckets (6) [Info](#)

[Create bucket](#)

Buckets are containers for data stored in S3.

Name	AWS Region	Creation date
chat-service-storage-dev-976289921590	US East (N. Virginia) us-east-1	December 20, 2025, 19:44:01 (UTC+02:00)
document-reader-storage-dev-976289921590	US East (N. Virginia) us-east-1	December 20, 2025, 19:44:00 (UTC+02:00)
quiz-service-storage-dev-976289921590	US East (N. Virginia) us-east-1	December 20, 2025, 19:44:01 (UTC+02:00)
shared-assets-dev-976289921590	US East (N. Virginia) us-east-1	December 20, 2025, 19:44:00 (UTC+02:00)
stt-service-storage-dev-976289921590	US East (N. Virginia) us-east-1	December 20, 2025, 19:44:00 (UTC+02:00)
tts-service-storage-dev-976289921590	US East (N. Virginia) us-east-1	December 20, 2025, 19:44:00 (UTC+02:00)

Each bucket is for different services like chat, document reader, quiz, shared assets, STT (speech-to-text), and TTS (text-to-speech)

document-reader-storage-dev-976289921590

ⓘ Public access is blocked because Block Public Access settings are turned on for this bucket

To determine which settings are turned on, check your Block Public Access settings for this bucket. Le

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "AllowInternalAccountAccess",  
            "Effect": "Allow",  
            "Principal": {  
                "AWS": "arn:aws:iam::976289921590:root"  
            },  
            "Action": [  
                "s3:GetObject",  
                "s3:PutObject",  
                "s3:DeleteObject",  
                "s3>ListBucket"  
            ],  
            "Resource": [  
                "arn:aws:s3:::document-reader-storage-dev-976289921590",  
                "arn:aws:s3:::document-reader-storage-dev-976289921590/*"  
            ]  
        }  
    ]  
}
```

This policy allows full internal access (get, put, delete, list) to the specified AWS account for this bucket and its contents, while public access is completely blocked

document-reader-storage-dev-976289921590



✓ Successfully edited bucket policy.

i Public access is blocked because Block Public Access settings are turned on for this bucket. To determine which settings are turned on, check your Block Public Access settings for this bucket.

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "AllowDocumentReaderService",  
      "Effect": "Allow",  
      "Principal": {  
        "AWS": "arn:aws:iam::976289921590:role/document-service-role"  
      },  
      "Action": [  
        "s3:GetObject",  
        "s3:PutObject",  
        "s3:DeleteObject",  
        "s3>ListBucket"  
      ],  
      "Resource": [  
        "arn:aws:s3:::document-reader-storage-dev-976289921590",  
        "arn:aws:s3:::document-reader-storage-dev-976289921590/*"  
      ]  
    }  
  ]  
}
```

The bucket policy now grants full access (get, put, delete, list) to the **document-service-role** within the same AWS account, while public access remains blocked.

quiz-service-storage-dev-976289921590

Public access is blocked because Block Public Access settings are turned on for this bucket.
To determine which settings are turned on, check your Block Public Access settings.

```
"Statement": [
    {
        "Sid": "AllowInternalAccountAccess",
        "Effect": "Allow",
        "Principal": {
            "AWS": "arn:aws:iam::976289921590:root"
        },
        "Action": [
            "s3:GetObject",
            "s3:PutObject",
            "s3:DeleteObject",
            "s3>ListBucket"
        ],
        "Resource": [
            "arn:aws:s3:::quiz-service-storage-dev-976289921590",
            "arn:aws:s3:::quiz-service-storage-dev-976289921590/*"
        ]
    }
]
```

S

This policy grants full internal access (get, put, delete, list) to the root account for the quiz-service bucket, while public access remains blocked

quiz-service-storage-dev-976289921590



✓ Successfully edited bucket policy.

To determine which settings are turned on, check your Bucket's Access Settings.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "AllowQuizService",  
            "Effect": "Allow",  
            "Principal": {  
                "AWS": "arn:aws:iam::976289921590:role/quiz-service-policy"  
            },  
            "Action": [  
                "s3:GetObject",  
                "s3:PutObject",  
                "s3:DeleteObject",  
                "s3>ListBucket"  
            ],  
            "Resource": [  
                "arn:aws:s3:::quiz-service-storage-dev-976289921590",  
                "arn:aws:s3:::quiz-service-storage-dev-976289921590/*"  
            ]  
        }  
    ]  
}
```

15

The bucket policy now grants full access (get, put, delete, list) to the **quiz-service IAM role**, replacing the previous root account access, while public access remains blocked

chat-service-storage-dev-976289921590

To determine which settings are turned on, check your Block Public Access settings.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "AllowInternalAccountAccess",  
            "Effect": "Allow",  
            "Principal": {  
                "AWS": "arn:aws:iam::976289921590:role/chat-service-role"  
            },  
            "Action": [  
                "s3:GetObject",  
                "s3:PutObject",  
                "s3:DeleteObject",  
                "s3>ListBucket"  
            ],  
            "Resource": [  
                "arn:aws:s3:::chat-service-storage-dev-976289921590",  
                "arn:aws:s3:::chat-service-storage-dev-976289921590/*"  
            ],  
            "Condition": {}  
        }  
    ]  
}
```

This policy grants full access (get, put, delete, list) to the **chat-service IAM role**, while public access remains blocked

chat-service-storage-dev-976289921590

✓ Successfully edited bucket policy.

i Public access is blocked because Block Public Access settings are turned on.
To determine which settings are turned on, check your Block Public Access settings.

```
"Statement": [  
    {  
        "Sid": "AllowChatService",  
        "Effect": "Allow",  
        "Principal": {  
            "AWS": "arn:aws:iam::976289921590:role/chat-service-role"  
        },  
        "Action": [  
            "s3:GetObject",  
            "s3:PutObject",  
            "s3:DeleteObject",  
            "s3>ListBucket"  
        ],  
        "Resource": [  
            "arn:aws:s3:::chat-service-storage-dev-976289921590",  
            "arn:aws:s3:::chat-service-storage-dev-976289921590/*"  
        ]  
    }  
]
```

The bucket policy now grants full access (get, put, delete, list) to the **chat-service IAM role**, while public access remains blocked

stt-service-storage-dev-976289921590

ⓘ Public access is blocked because Block Public Access settings are turned on for this bucket
To determine which settings are turned on, check your Block Public Access settings for this bucket. Learn more about [using](#)

```
{  
    "Sid": "AllowSTTRoleAccess",  
    "Effect": "Allow",  
    "Principal": {  
        "AWS": "arn:aws:iam::976289921590:role/stt-service-role"  
    },  
    "Action": [  
        "s3:GetObject",  
        "s3:PutObject",  
        "s3:DeleteObject",  
        "s3>ListBucket"  
    ],  
    "Resource": [  
        "arn:aws:s3:::stt-service-storage-dev-976289921590",  
        "arn:aws:s3:::stt-service-storage-dev-976289921590/*"  
    ],  
    "Condition": {  
        "StringEquals": {  
            "aws:SourceArn": "arn:aws:kms:us-east-1:976289921590:key/78c1bbc0-7837-4d9b-a652-e962d58fa98b"  
        }  
    }  
}
```

This policy grants full access (get, put, delete, list) to the **STT service IAM role**, but only when requests originate from the specified KMS key, while public access remains blocked

stt-service-storage-dev-976289921590



✓ Successfully edited bucket policy.

ⓘ Public access is blocked because Block Public Access settings are turned on.
To determine which settings are turned on, check your Block Public Acc

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "AllowSTTService",  
            "Effect": "Allow",  
            "Principal": {  
                "AWS": "arn:aws:iam::976289921590:role/stt-service-role"  
            },  
            "Action": [  
                "s3:GetObject",  
                "s3:PutObject",  
                "s3:DeleteObject",  
                "s3>ListBucket"  
            ],  
            "Resource": [  
                "arn:aws:s3:::stt-service-storage-dev-976289921590",  
                "arn:aws:s3:::stt-service-storage-dev-976289921590/*"  
            ]  
        }  
    ]  
}
```

The bucket policy now grants full access (get, put, delete, list) to the **STT service IAM role**, while public access remains blocked

tts-service-storage-dev-976289921590

ⓘ Public access is blocked because Block Public Access settings are turned on.
To determine which settings are turned on, check your Block Public Ac

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "AllowInternalAccountAccess",  
            "Effect": "Allow",  
            "Principal": {  
                "AWS": "arn:aws:iam::976289921590:root"  
            },  
            "Action": [  
                "s3:GetObject",  
                "s3:PutObject",  
                "s3:DeleteObject",  
                "s3>ListBucket"  
            ],  
            "Resource": [  
                "arn:aws:s3:::tts-service-storage-dev-976289921590",  
                "arn:aws:s3:::tts-service-storage-dev-976289921590/*"  
            ]  
        }  
    ]  
}
```

This policy grants full internal access (get, put, delete, list) to the root account for the TTS service bucket, while public access remains blocked

tts-service-storage-dev-976289921590

 Successfully edited bucket policy.

To determine which settings are turned on, check your Block Public Access setting.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "AllowTTSService",  
            "Effect": "Allow",  
            "Principal": {  
                "AWS": "arn:aws:iam::976289921590:role/tts-service-role"  
            },  
            "Action": [  
                "s3:GetObject",  
                "s3:PutObject",  
                "s3:DeleteObject",  
                "s3>ListBucket"  
            ],  
            "Resource": [  
                "arn:aws:s3:::tts-service-storage-dev-976289921590",  
                "arn:aws:s3:::tts-service-storage-dev-976289921590/*"  
            ]  
        }  
    ]  
}
```

S

The bucket policy now grants full access (get, put, delete, list) to the **TTS service IAM role**, while public access remains blocked

shared-assets-dev-976289921590

ⓘ Public access is blocked because Block Public Access settings are turned on.
To determine which settings are turned on, check your Block Public Access settings.

```
"Version": "2012-10-17",
"Statement": [
    {
        "Sid": "AllowInternalAccountAccess",
        "Effect": "Allow",
        "Principal": {
            "AWS": "arn:aws:iam::976289921590:root"
        },
        "Action": [
            "s3:GetObject",
            "s3:PutObject",
            "s3:DeleteObject",
            "s3>ListBucket"
        ],
        "Resource": [
            "arn:aws:s3:::shared-assets-dev-976289921590",
            "arn:aws:s3:::shared-assets-dev-976289921590/*"
        ]
    }
],
```

This policy grants full internal access (get, put, delete, list) to the root account for the shared-assets bucket, while public access remains blocked

shared-assets-dev-976289921590

 Successfully edited bucket policy.

To determine which settings are turned on, check your Block Public Access settings.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "AllowAllServiceRoles",  
            "Effect": "Allow",  
            "Principal": {  
                "AWS": [  
                    "arn:aws:iam::976289921590:role/document-service-role",  
                    "arn:aws:iam::976289921590:role/tts-service-role",  
                    "arn:aws:iam::976289921590:role/quiz-service-policy",  
                    "arn:aws:iam::976289921590:role/stt-service-role",  
                    "arn:aws:iam::976289921590:role/chat-service-role"  
                ]  
            },  
            "Action": [  
                "s3:GetObject",  
                "s3:PutObject"  
            ],  
            "Resource": [  
                "arn:aws:s3:::shared-assets-dev-976289921590/*"  
            ]  
        }  
    ]  
}
```

The shared-assets bucket now allows **read and write access** to all service-specific IAM roles (document, TTS, quiz, STT, chat), while public access remains blocked

SQS fair queues is now available for standard queues.
Automatically manage noisy neighbors in your queues with fair queues, a new feature that limits noisy neighbor impact across all message groups. Add a group identifier to your messages, and SQS re-orders messages to ensure no single tenant impact the time in queue for any other tenants.

[Learn more](#)

Create queue

Details

Type

Choose the queue type for your application or cloud infrastructure.

Standard info
At-least-once delivery, message ordering isn't preserved

- At-least once delivery
- Best-effort ordering

FIFO info
First-in-first-out delivery, message ordering is preserved

- First-in-first-out delivery
- Exactly-once processing

Info You can't change the queue type after you create a queue.

Name

quiz-service-queue

A queue name is case-sensitive and can have up to 80 characters. You can use alphanumeric characters, hyphens (-), and underscores (_).

Configuration [Info](#)
Set the maximum message size, visibility to other consumers, and message retention.

CloudShell Feedback Console Mobile App

Seconds Days

Should be between 0 seconds and 12 hours.

Delivery delay [Info](#)
0 Seconds

Should be between 0 seconds and 15 minutes.

Receive message wait time [Info](#)
0 Seconds

Should be between 0 and 20 seconds.

Encryption [Info](#)
Amazon SQS provides in-transit encryption by default. To add at-rest encryption to your queue, enable server-side encryption.

Server-side encryption
 Disabled
 Enabled

Encryption key type
 Amazon SQS key (SSE-SQS)
An encryption key that Amazon SQS creates, manages, and uses for you.

AWS Key Management Service key (SSE-KMS)
An encryption key protected by AWS Key Management Service (AWS KMS).

Access policy [Info](#)
Define who can access your queue.

Choose method
 Basic
Use simple criteria to define a basic access policy.

Advanced
Use a JSON object to define an advanced access policy.

JSON (read-only)

```
{
  "Version": "2012-10-17",
  "Id": "..._default_policy_ID",
  "Statement": [
    {
      "Sid": "..._owner_statement",
      "Effect": "Allow",
      "Principal": {
        "AWS": "976289921590"
      },
      "Action": [
        "SQS:*"
      ],
      "Resource": "arn:aws:sqs:us-east-1:976289921590:quiz-service-queue"
    }
  ]
}
```

Redrive allow policy - Optional [Info](#)
Identify which source queues can use this queue as the dead-letter queue.

Select which source queues can use this queue as the dead-letter queue.
 Disabled
 Enabled

Dead-letter queue - Optional [Info](#)
Send undeliverable messages to a dead-letter queue.

Set this queue to receive undeliverable messages.
 Disabled
 Enabled

Tags - Optional [Info](#)
A tag is a label assigned to an AWS resource. Use tags to search and filter your resources or track your AWS costs.

Key Value - optional

Add new tag

You can add 49 more tags.

[Cancel](#) [Create queue](#)

This configuration creates a **Standard SQS queue named quiz-service-queue** with:

- No encryption,
- 4-day message retention,
- Open access policy (anyone can send/receive messages),
- No dead-letter queue,
- Tag: Owner = Merna.

The screenshot shows the 'Create subscription' wizard in the AWS SNS console. The 'Protocol' is set to 'Amazon SQS' and the 'Endpoint' is 'arn:aws:sqs:us-east-1:976289921590:quiz-service-queue'. A note indicates that raw message delivery is disabled. A confirmation message states: 'After your subscription is created, you must confirm it.' Below the main form, there's a section for 'Subscription filter policy - optional' and a 'Redrive policy (dead-letter queue) - optional' section. At the bottom right are 'Cancel' and 'Create subscription' buttons.

This configuration creates an SNS subscription that sends messages from the **document-processed-topic** SNS topic to the **quiz-service-queue** SQS queue using the **Amazon SQS protocol**

The screenshot shows the AWS Lambda function editor for a function named 'document_reader'. The 'Code source' tab is selected, showing the 'index.py' file content:

```

def lambda_handler(event, context):
    document_id = "12345"
    extracted_text = "Some extracted text"

    event_payload = {
        'document_id': document_id,
        'extracted_text': extracted_text
    }

    sns.publish(
        TopicArn='arn:aws:sns:us-east-1:976289921590:document-processed-topic',
        Message=json.dumps(event_payload)
    )

    return {
        'statusCode': 200,
        'body': json.dumps('Event published successfully!')
    }

```

The left sidebar shows the function structure with 'EXPLORER', 'DOCUMENT.READER', 'DEPLOY', and 'TEST EVENTS [NONE SELECTED]' sections. Buttons for 'Deploy' and 'Test' are visible.

This Lambda function simulates document processing and sends an SNS notification to the document-processed-topic with extracted text details

The screenshot shows the AWS SNS 'Create topic' interface. The 'Type' dropdown is set to 'Standard'. The 'Name' field contains 'document-processed-topic'. Other optional settings like 'Encryption', 'Access policy', 'Data protection policy', 'Delivery policy', 'Message delivery status logging', 'Tags', and 'Active tracing' are shown but not modified. At the bottom right are 'Cancel' and 'Create topic' buttons.

This configuration creates a **Standard SNS topic named document-processed-topic** with default settings (no encryption, no custom access policy, no tags).

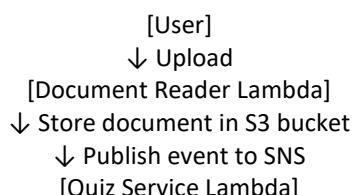
AWS Data Sharing Mechanism Implementation Guide

1. Introduction

In this project, services are designed to have **completely isolated storage** and **share data only via events**. Main services:

- Document Reader Service
- Quiz Service

2. System Architecture



↓ Consume from SQS
 ↓ Store relevant data in its own DB/S3

3. Document Reader Service

3.1 S3 Bucket

- Each service has a **dedicated bucket**.
- Example: document-reader-storage-dev-976289921590
- Stores uploaded documents.

3.2 Lambda Function

- Runtime: **Python 3.11**
- Code:

```
import boto3
import json
```

```
sns = boto3.client("sns")
s3 = boto3.client("s3")
```

```
TOPIC_ARN = "arn:aws:sns:us-east-1:976289921590:document-processed-topic"
BUCKET_NAME = "document-reader-storage-dev-976289921590"
```

```
def lambda_handler(event, context):
```

```
    # Store document in S3
    document_id = "12345"
    extracted_text = "Some extracted text"
    document_content = b"Sample file content"
```

```
s3.put_object(
    Bucket=BUCKET_NAME,
    Key=f"{document_id}.txt",
    Body=document_content
)
```

```
# Publish event to SNS
```

```
payload = {
    "document_id": document_id,
    "extracted_text": extracted_text
}
```

```
sns.publish(
    TopicArn=TOPIC_ARN,
    Message=json.dumps(payload)
)
```

```
return {
    "statusCode": 200,
    "body": "Event published to SNS and file stored in S3"
}
```

3.3 Permissions

- Lambda Execution Role must include:
 - **AmazonS3FullAccess** (or limited to bucket)
 - **AmazonSNSFullAccess**
 - **AWSLambdaBasicExecutionRole**

4. SNS Topic

- Type: **Standard**
- Name: document-processed-topic
- Purpose: Publish events when a document is processed.

5. Quiz Service

5.1 SQS Queue

- Receives events from SNS:
 - Queue Name: quiz-service-queue
 - Subscribed to SNS Topic: document-processed-topic
 - Each service has its own dedicated queue.

5.2 Lambda Function

- Handles events from SQS
- Stores relevant data in **S3 or service-specific database**

Example Code:

```
import boto3
import json
```

```
sqs = boto3.client("sqs")
s3 = boto3.client("s3")
```

```
QUEUE_URL = "https://sqs.us-east-1.amazonaws.com/976289921590/quiz-service-queue"
BUCKET_NAME = "quiz-service-storage-dev-976289921590"
```

```
def lambda_handler(event, context):
    for record in event['Records']:
        message = json.loads(record['body'])
        document_id = message['document_id']
        extracted_text = message['extracted_text']

        # Store in service-specific S3
        s3.put_object(
            Bucket=BUCKET_NAME,
            Key=f"{document_id}.txt",
            Body=extracted_text.encode('utf-8')
        )

    return {
        'statusCode': 200,
        'body': 'Data processed and stored successfully'
    }
```

5.3 Permissions

- Lambda Execution Role must include:
 - **AmazonS3FullAccess** (or limited to bucket)
 - **AmazonSQSFullAccess**
 - **AWSLambdaBasicExecutionRole**

6. Key Principles

1. **Isolated storage:** Each service has its own S3 bucket and database.
2. **Event-driven data sharing:** No direct database queries or S3 access between services.
3. **Decoupling:** Services communicate only via SNS/SQS events.

Implementing Infrastructure as Code on AWS

Infrastructure as Code (IaC) is an approach used to manage and provision cloud infrastructure through machine-readable configuration files instead of manual processes. On AWS, this approach ensures consistency, scalability, and automation across different environments.

2. Tool Selection: AWS CloudFormation

AWS CloudFormation is used as the Infrastructure as Code tool because it is a native AWS service that allows infrastructure to be defined using YAML or JSON templates. CloudFormation manages the creation, update, and deletion of AWS resources in a controlled and automated manner.

3. Step 1: Access AWS CloudFormation

- Log in to the AWS Management Console.
- Select the required AWS Region.
- Search for CloudFormation from the AWS services menu.
- Open the CloudFormation dashboard.

4. Step 2: Define Infrastructure Using Templates

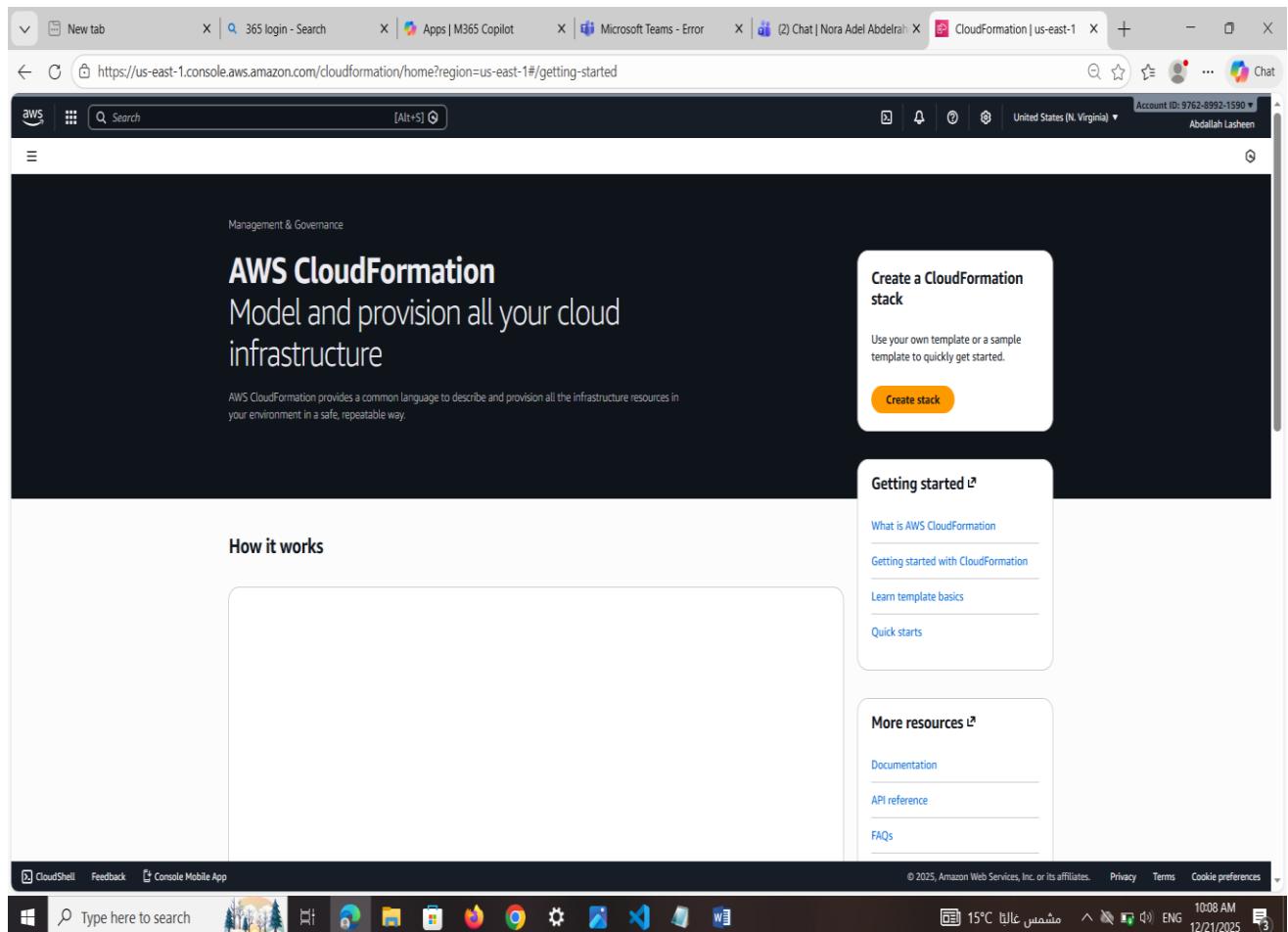
Infrastructure is defined using CloudFormation templates written in YAML. Each template represents a specific infrastructure component.

Typical components include:

The screenshot shows the AWS CloudFormation console interface. On the left, a sidebar lists services like CloudFormation, Infrastructure Composer, and Public extensions. The main content area displays the CloudFormation service with a summary card: "Create and Manage Resources with Templates". Below this are sections for "Features" (IaC Generator, Spotlight, Public extensions), "Resources in us-east-1" (with a note about cross-Region search), and "Cost and usage". A central panel shows application details for "No applications" in "AWS East (N. Virginia)". The bottom navigation bar includes links for CloudShell, Feedback, and Console Mobile App, along with system status and time information.

- Compute (EC2)
- Security (Security Groups, IAM roles)

This code-based definition ensures repeatable and version-controlled infrastructure.



5. Step 3: Implement Modular Infrastructure Components

To implement modular infrastructure:

- Each infrastructure component is defined in a separate template.

Examples:

- VPC template
- EC2 template
- Database template

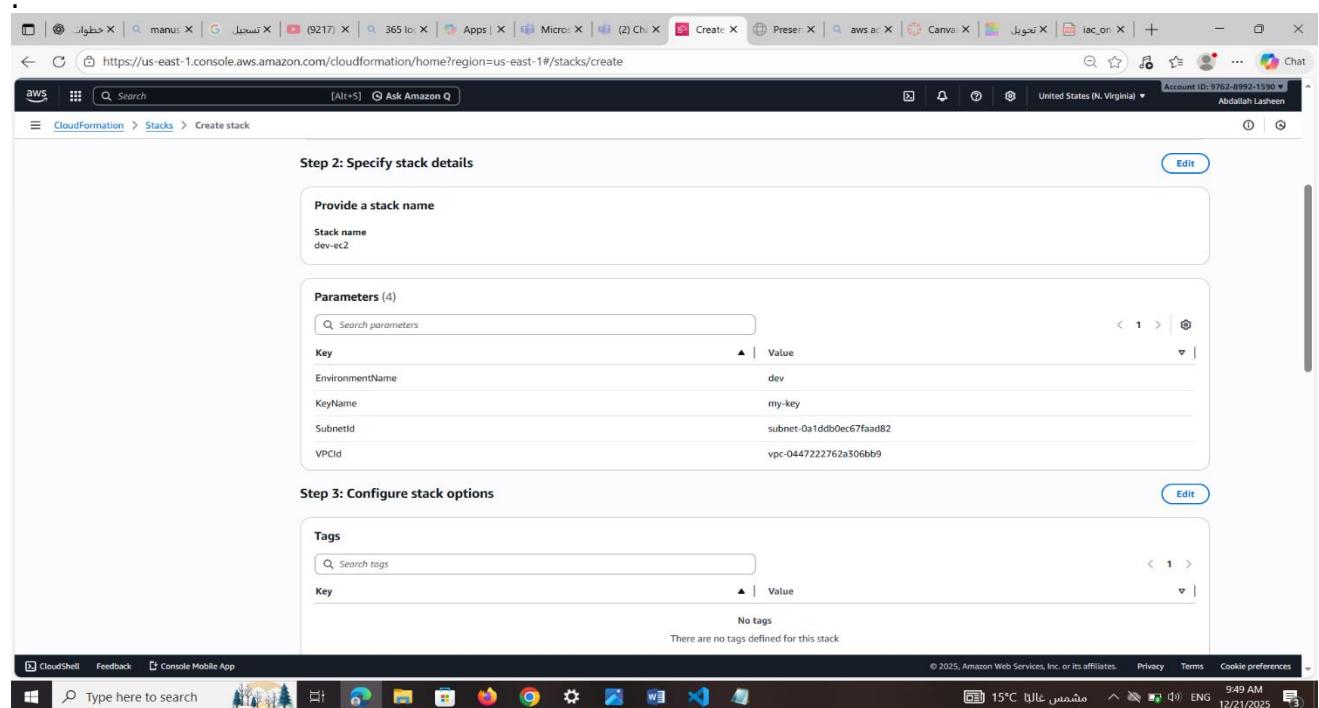
Each module can be reused independently, making the infrastructure easier to manage and extend.

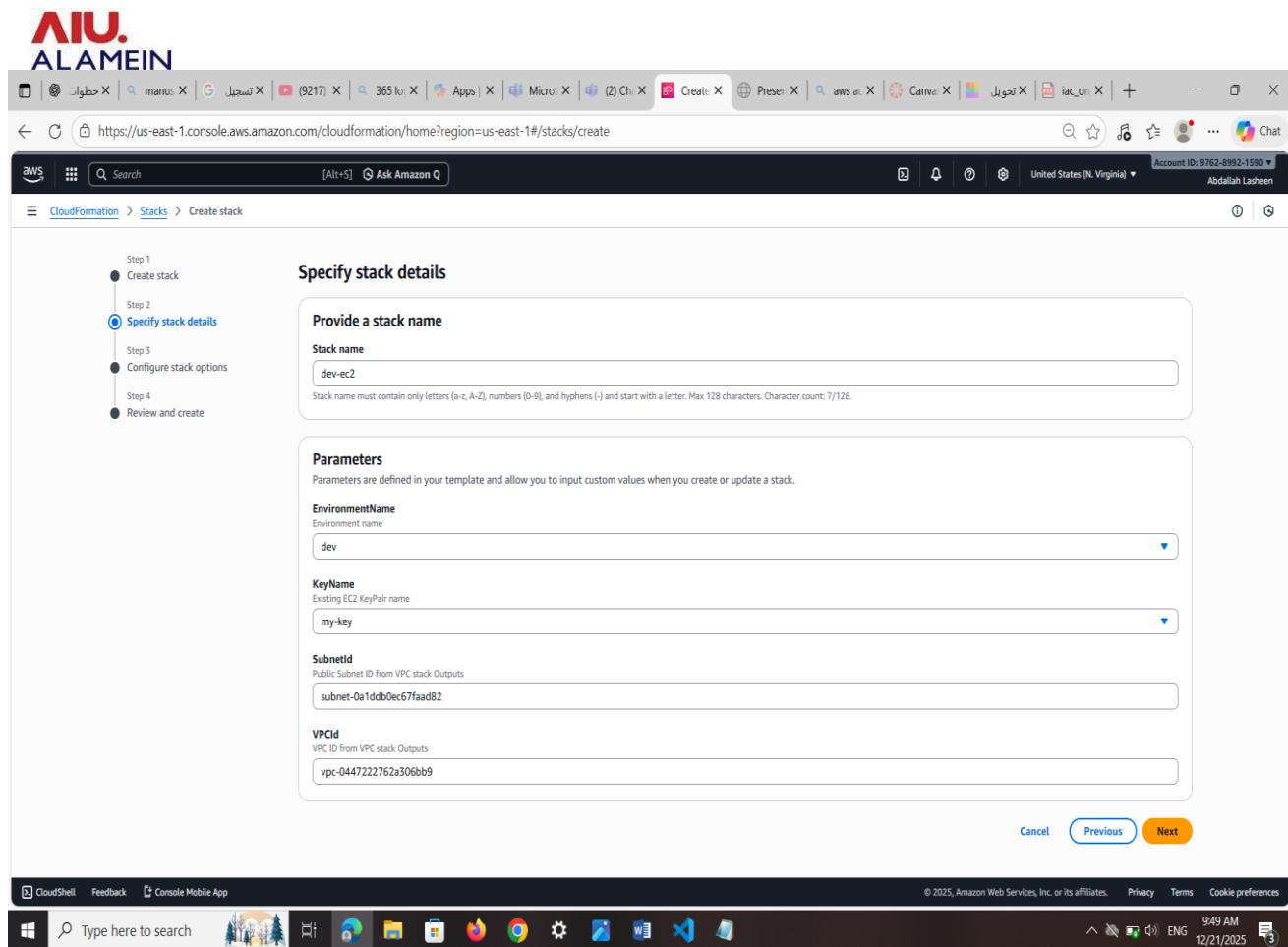
6. Step 4: Create a Stack for Each Module

In CloudFormation, each template is deployed as a stack. Steps:

- Click Create stack.
- Upload the module template (for example, the VPC template).
- Provide required parameters.
- Create the stack.

This modular approach improves flexibility and maintenance



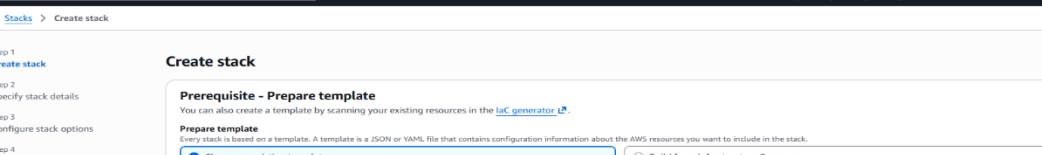


7. Step 5: Separate Environments Using Parameters

Environment separation is achieved by using parameters in the templates. Example environments:

- Development (dev)
 - Staging
 - Production (prod) Each environment uses:
 - The same template
 - Different parameter values
 - A separate CloudFormation stack

This ensures complete isolation between environments.



The screenshot shows the AWS CloudFormation 'Create stack' wizard at Step 1: Create stack. The left sidebar lists steps: Step 1 (Create stack, highlighted), Step 2 (Specify stack details), Step 3 (Configure stack options), and Step 4 (Review and create). The main area is titled 'Create stack' and contains a 'Prerequisite - Prepare template' section. It says you can also create a template by scanning your existing resources in the [LaC generator](#). Below is a radio button group for choosing a template: 'Choose an existing template' (selected) and 'Build from Infrastructure Composer'. A note states: 'Your stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.' At the bottom, there's a 'Specify template' section with options for 'Template source': 'Amazon S3 URL' (radio button), 'Upload a template file' (radio button selected, with a 'Choose file' button and a 'vpc.yaml' file listed), and 'Sync from Git'.

The screenshot shows the AWS CloudFormation 'Create stack' wizard. The left sidebar lists steps: Step 1 (Create stack) is completed (solid blue dot), Step 2 (Specify stack details) is active (outline blue dot), Step 3 (Configure stack options) is pending (outline grey dot), Step 4 (Review and create) is pending (outline grey dot). The main area is titled 'Specify stack details' and contains a 'Provide a stack name' section with a text input field containing 'Dev-vpc'. Below it is a note: 'Stack name must contain only letters (a-z, A-Z), numbers (0-9), and hyphens (-) and start with a letter. Max 128 characters. Character count: 7/128.' Further down are 'Parameters' and 'EnvironmentName' sections, each with dropdown menus. At the bottom are 'Cancel', 'Previous', and 'Next' buttons.

CloudShell Feedback Console Mobile App © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 15°C مقتبس ١٢/٢١/٢٠٢٥ 10:10 AM ENG عالیہ

8. Step 6: Deploy the Development Environment

For the development environment:

Create a stack named dev-vpc. (CSE353) Cloud Computing

- Set the parameter EnvironmentName to dev.
- Deploy the stack.

Development resources are used for testing and experimentation.

The screenshots show the AWS CloudFormation 'Create stack' wizard. The first screenshot shows the 'Configure stack options' step, where the user is setting up tags, permissions, and stack failure options. The second screenshot shows the 'Review and create' step, where the user is reviewing the template and parameters before deployment. The third screenshot shows the 'Review and create' step again, with the 'Edit' button visible.

9. Step 7: Deploy the Staging Environment

For the staging environment:

- Create a stack named staging-vpc.
- Set the parameter EnvironmentName to staging.
- Deploy the stack.

The screenshot shows the AWS CloudFormation console with the 'staging-vpc' stack selected. The left sidebar lists three stacks: 'staging-vpc' (CREATE_COMPLETE), 'prod-vpc' (CREATE_IN_PROGRESS), and 'dev-vpc' (CREATE_IN_PROGRESS). The main panel displays the 'Overview' tab for the 'staging-vpc' stack, which was created on 2025-12-21 at 08:16:50 UTC+0200. The stack ID is arn:aws:cloudformation:us-east-1:976289921590:stack/staging-vpc/a2d85e80-de34-11f0-ac01-12e67062d135. The status is CREATE_COMPLETE. The stack has no parent stack and no deleted time. The last drift check time was 2025-12-21 08:16:50 UTC+0200. Termination protection is deactivated. The stack has a detailed status and a description: "VPC Template with environment separation (dev, staging, prod)". The bottom of the screen shows the Windows taskbar with various pinned icons and the system tray indicating 15°C, 10:39 AM, and 12/21/2025.

10. Step 8: Deploy the Production Environment

For the production environment:

- Create a stack named prod-vpc.
- Set the parameter EnvironmentName to prod.
- Deploy the stack.

Production infrastructure is isolated, secure, and stable.

The screenshot shows the AWS CloudFormation console with the 'prod-vpc' stack selected. The left sidebar lists three stacks: prod-vpc, staging-vpc, and dev-vpc, all in 'CREATE_COMPLETE' status. The main panel displays the 'Overview' section for the prod-vpc stack, including its Stack ID, Status (CREATE_COMPLETE), and various metadata fields like Description, Root stack, and Drift status.

11. Step 9: Review and Create the Stack

Before deployment:

- Review all stack details.
- Acknowledge required AWS permissions.
- Create the stack.

The screenshot shows the AWS CloudFormation console with the URL <https://us-east-1.console.aws.amazon.com/cloudformation/home?region=us-east-1#/stacks?filteringText=&filteringStatus=active&reviewNested=true>. The page displays a list of three stacks:

Stack name	Status	Created time	Description
prod-vpc	CREATE_COMPLETE	2025-12-21 09:25:56 UTC+0200	VPC Template with environment separation (dev, staging, prod)
staging-vpc	CREATE_COMPLETE	2025-12-21 08:16:50 UTC+0200	VPC Template with environment separation (dev, staging, prod)
dev-vpc	CREATE_COMPLETE	2025-12-21 08:01:20 UTC+0200	VPC Template with environment separation (dev, staging, prod)

12. Step 10:code

AWSTemplateFormatVersion: '2010-09-09'

Description: EC2 Instance using existing VPC and Subnet

Parameters:

EnvironmentName:

Type: String

AllowedValues:

- dev
- staging
- prod

Description: Environment name

VPCId:

Type: String

Description: VPC ID from VPC stack Outputs

SubnetId:

Type: String

Description: Public Subnet ID from VPC stack Outputs

KeyName:

Type: AWS::EC2::KeyPair::KeyName

Description: Existing EC2 KeyPair name

Resources:

InstanceSecurityGroup:

Type: AWS::EC2::SecurityGroup

Properties:

GroupDescription: Allow SSH access

VpcId: !Ref VPCId

SecurityGroupIngress:

- IpProtocol: tcp

- FromPort: 22

- ToPort: 22

- CidrIp: 0.0.0.0/0

Tags:

- Key: Name

- Value: !Sub "\${EnvironmentName}-sg"

MyInstance:

Type: AWS::EC2::Instance

Properties:

 InstanceType: t2.micro

 ImageId: ami-0c02fb55956c7d316 # Amazon Linux 2 (us-east-1)

 KeyName: !Ref KeyName

 SubnetId: !Ref SubnetId

 SecurityGroupIds:

 - !Ref InstanceSecurityGroup

Tags:

 - Key: Name

 Value: !Sub "\${EnvironmentName}-ec2"

Outputs:

InstanceId:

Description: EC2 Instance ID

Value: !Ref MyInstance

1. Network Security

1.1 Private Subnets

- All services are deployed in **private subnets**.
- Direct internet access is restricted.

1.2 Internet Exposure

- Only the **Application Load Balancer (ALB)** is exposed to the internet.
- All other resources remain in private subnets.

1.3 Security Groups

- Implement **least privilege** rules.
- Only allow required inbound/outbound traffic.

1.4 Monitoring

- Enable **VPC Flow Logs** to monitor traffic.

1.5 Web Protection

- Enable **AWS WAF** on ALB to block malicious requests.
- Implement **DDoS protection** with **AWS Shield**.

2. Data Security

2.1 Encryption at Rest

- Encrypt all data in **S3, EBS, and RDS**.
- Use **AWS KMS** for encryption key management.
- Maintain **separate keys per service**.
- Implement **regular key rotation**.

2.2 Encryption in Transit

- Use **TLS 1.3** for all communication.

2.3 Secrets Management

- Store sensitive credentials in **AWS Secrets Manager**.
- Avoid hard-coding secrets in code or containers.

4. Access Control

Service	IAM Role	Trusted Entity	AWS Service	Allowed Actions
---------	----------	----------------	-------------	-----------------

Chat Service	chat-service-role	ecs-tasks.amazonaws.com	S3	GetObject, PutObject, DeleteObject, ListBucket
Chat Service	chat-service-role	ecs-tasks.amazonaws.com	CloudWatch Logs	CreateLogGroup, CreateLogStream, PutLogEvents
Lambda Service	lambda-exec-role	lambda.amazonaws.com	S3	GetObject, PutObject
Lambda Service	lambda-exec-role	lambda.amazonaws.com	CloudWatch Logs	CreateLogGroup, CreateLogStream, PutLogEvents
EC2 Service	ec2_platform_role	ec2.amazonaws.com	RDS	DescribeDBInstances, Connect
Service	IAM Role	Trusted Entity	AWS Service	Allowed Actions
Document Reader	document_reader-role-gpgftvdt	lambda.amazonaws.com	SNS	sns:*

Document Reader	document_reader-role-gpgftvdt	lambda.amazonaws.com	SMS via SNS	sms-voice:SMS via SNS voice:Verify
Document Reader	document_reader-role-gpgftvdt	lambda.amazonaws.com	CloudWatch Logs	CreateLogG PutLogEvent
Service	IAM Role	Trusted Entity	AWS Service	Allowed Actions
TTS Service	tts-service-role	ecs-tasks.amazonaws.com	S3	GetObject, ListBucket
TTS Service	tts-service-role	ecs-tasks.amazonaws.com	CloudWatch Logs	CreateLogG PutLogEvent

TTS Service

tts-service-role

ecs-tasks.amazonaws.com

Polly

SynthesizeS

Service

IAM Role

Trusted Entity

AWS Service

Allowed Act

Quiz Service

quiz-service-role

ecs-tasks.amazonaws.com

S3

GetObject, ListBucket

Quiz Service

quiz-service-role

ecs-tasks.amazonaws.com

Secrets Manager

GetSecretValue

Service

IAM Role

Trusted Entity

AWS Service

Allowed Act

Container Service

container_service_account_role

ecs-tasks.amazonaws.com

Elastic Container Registry

GetAuthorizationResult, BatchCheckImage, GetDownloadUrl

Service	IAM Role	Trusted Entity	AWS Service	Allowed Actions
Document Reader	document_reader-role-gpgftvdt	lambda.amazonaws.com	SNS	* (All SNS actions) voice:Describe CreateVerification SendTextMessage DeleteVerification VerifyDestination DescribeAccruedFee DescribeSpeechMarkings DescribePhoneNumber SetTextMessage DescribeOptInStatus DeleteOptInStatus
Document Reader	document_reader-role-gpgftvdt	lambda.amazonaws.com	CloudWatch Logs	CreateLogGroup PutLogEvent
Service	IAM Role	Trusted Entity	AWS Service	Allowed Actions
Lambda Service	lambda-exec-role	lambda.amazonaws.com	SNS	* (Full access)

Service	IAM Role	Trusted Entity	AWS Service	Allowed Actions
Lambda Service	lambda-exec-role	lambda.amazonaws.com	End User Messaging SMS	DescribeVerification, CreateVerification, SendDestination, SendTextMessage, DeleteVerification, VerifyDestination, DescribeAccount, DescribeSpecification, DescribePhoneNumber, SetTextMessage, DescribeOptInStatus, DeleteOptedOutStatus
EC2 Service	terraform-ec2-role	ec2.amazonaws.com	EC2	*
EC2 Service	terraform-ec2-role	ec2.amazonaws.com	S3	Get*, List*
EC2 Service	terraform-ec2-role	ec2.amazonaws.com	CloudWatch	*

**EC2
Service**

terraform-ec2-role

ec2.amazonaws.com

**Auto
Scaling**

*

**EC2
Service**

terraform-ec2-role

ec2.amazonaws.com

**ELB / ELB
v2**

*

**EC2
Service**

terraform-ec2-role

ec2.amazonaws.com

IAM

CreateService