

RAG Chatbot using AWS Bedrock and Streamlit Framework

A Step towards building an Executive AI Assistant: A Multi-LLM Chatbot for Summarizing Business Reports

Discussion 2
Group 4

8th May 2025

Team Members

- Aswin Karthik Panneer Selvam
- Billy Muchipisi
- Shaikh Umair Ahmed

Agenda

- Objective
- Problem Statement
- Solution Sequence Diagram
- Retrieval-augmented generation through AWS Services Lens
- AWS Services Used
- Architecture
- Overview of AWS Cloud Formation
- Comment on IAM Roles, Profile and Policies
- Demo of the Application
- Setup Steps
- Termination Steps
- Areas of Improvement
- References
- Appendix with some Output Snippets

Objective

Motivation

- In today's data-driven world, decision-makers are overwhelmed by the volume of unstructured information they must process daily.
- This project was inspired by the need to accelerate insight extraction, improve executive decision-making, and leverage the power of generative AI—all while gaining practical skills in cloud-based solution development.

Use Case Scenarios

- CEO Dashboard Assistant: Summarizes weekly reports, flags urgent insights
- Legal Document Analyzer: Extracts key clauses
- Academic Research Assistant: Summarizes literature
- Sales Enablement Bot: CRM and meeting summaries
- Medical Case Review Bot: Structured summaries from history PDFs

Problem Statement





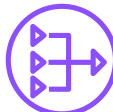

The CEO's Dilemma

- Receives dozens of departmental reports weekly
- Needs to make quick, informed decisions
- No time to read everything





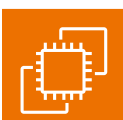
Our Solution

- A conversational chatbot that uses Retrieval-Augmented Generation (RAG) to answer questions using private document data
- Features
 - Enable multi-format document ingestion (PDF, Word, CSV, etc.)
 - Use reliable foundation models (FMs) for generative responses
 - Empower users to choose from multiple LLMs (for relevant Use cases)
 - Deliver secure, scalable, and accurate responses

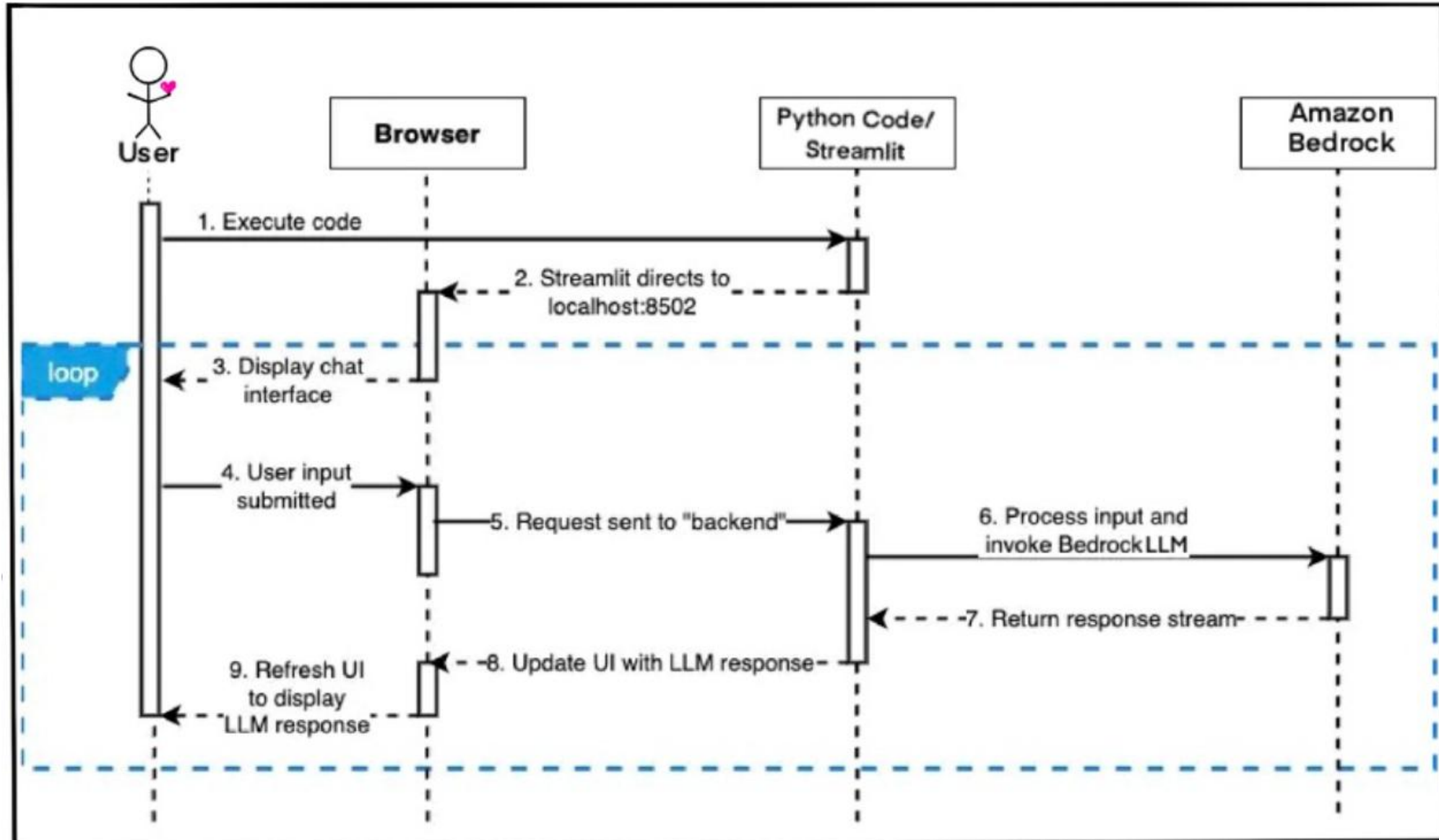
AWS Service Used – 1

	AWS Service	Purpose	Why Used
	AWS Bedrock	Provides access to foundation models for responses and embeddings.	Serverless access to GenAI models without managing infrastructure.
	AWS CloudFormation	Automates provisioning of AWS infrastructure as code.	Ensures reproducibility and consistency across environments.
	AWS S3	Stores uploaded files (e.g., PDFs, audio) for processing.	Scalable, durable, and integrates well with other AWS services.
	AWS VPC	Provides isolated networking for secure deployment.	Enables segmentation of public/private subnets and secure architecture.
	AWS NAT Gateway	Allows private subnet instances to access the internet securely.	Supports internet-bound traffic from private resources without exposing them.
	AWS Internet Gateway	Enables internet access for public subnet resources.	Required for internet connectivity for NAT Gateway and public EC2 instances.

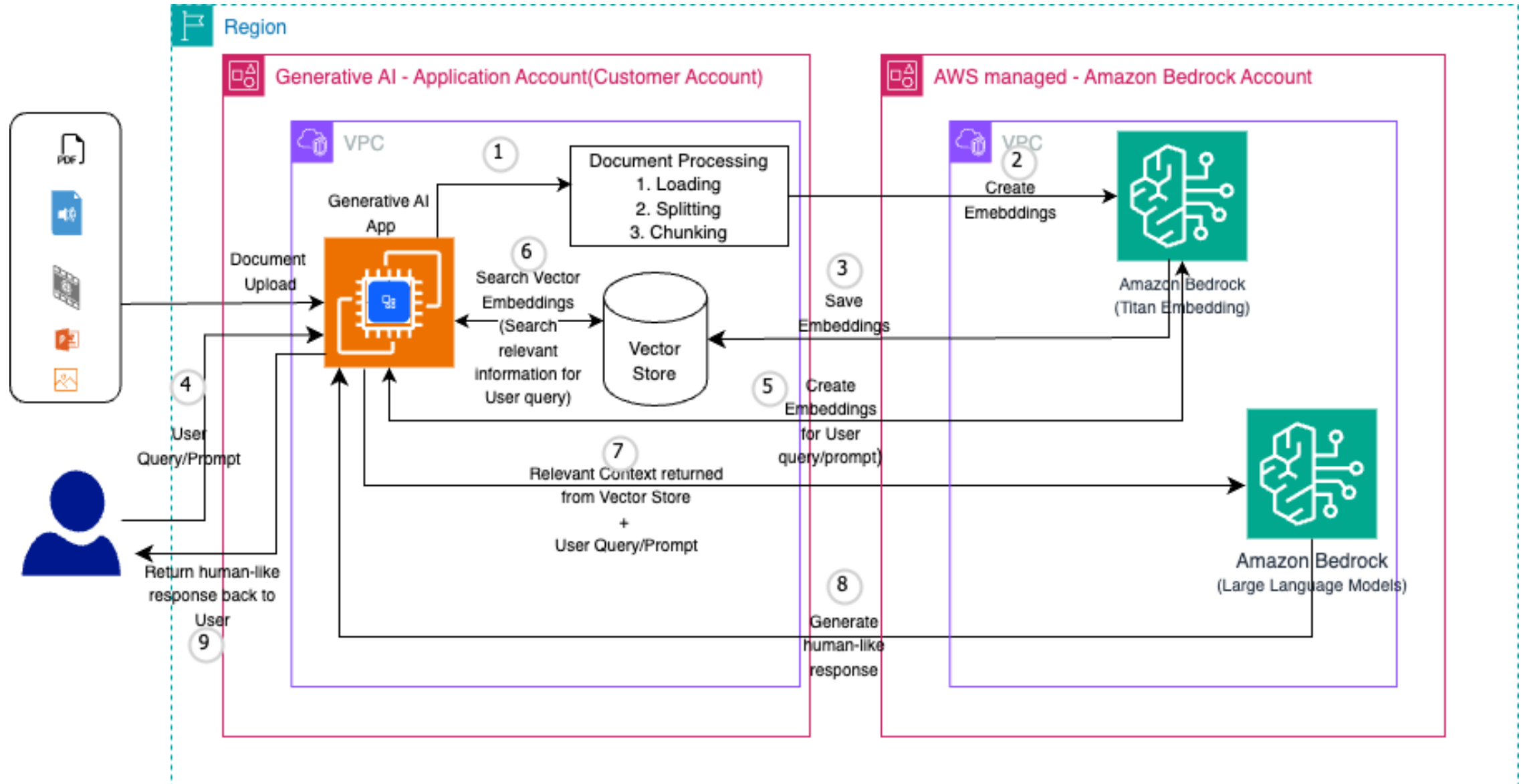
AWS Service Used – 2

	AWS Service	Purpose	Why Used
	Application Load Balancer	Distributes traffic across EC2 instances.	Ensures high availability, auto-scaling, and failover support.
	AWS Textract	Extracts text and data from documents.	Automates document processing in the RAG pipeline.
	Amazon Transcribe	Converts speech in audio files to text.	Enables audio input processing as part of multimodal input handling.
	AWS OpenSearch	Stores and retrieves embeddings via vector similarity search.	Enables fast, scalable semantic search for the retrieved documents.
	AWS EC2 (Streamlit App)	Hosts the Streamlit-based chat application interface.	Provides customizable and scalable environment for hosting the web app frontend.

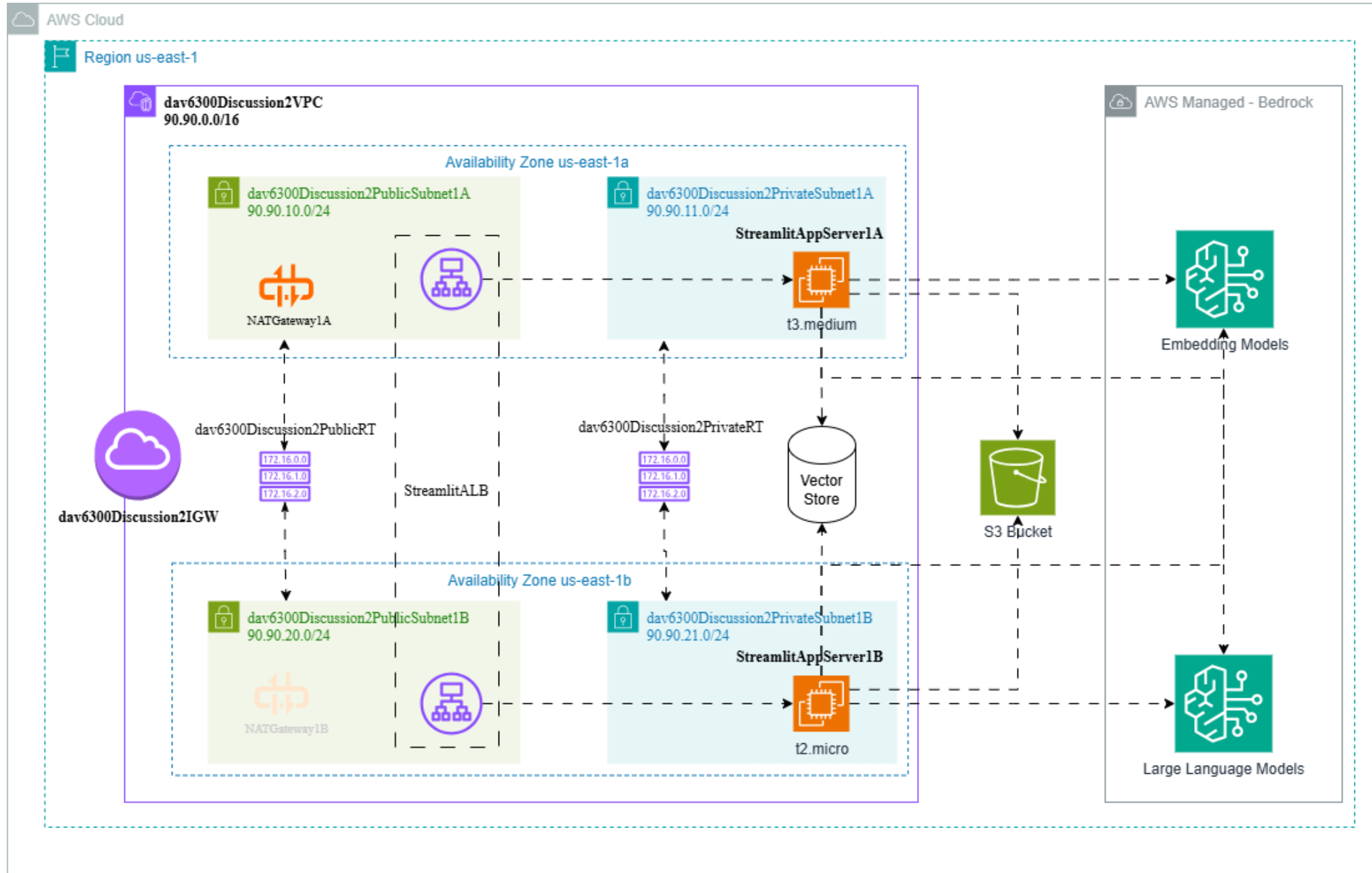
Solution Sequence Diagram



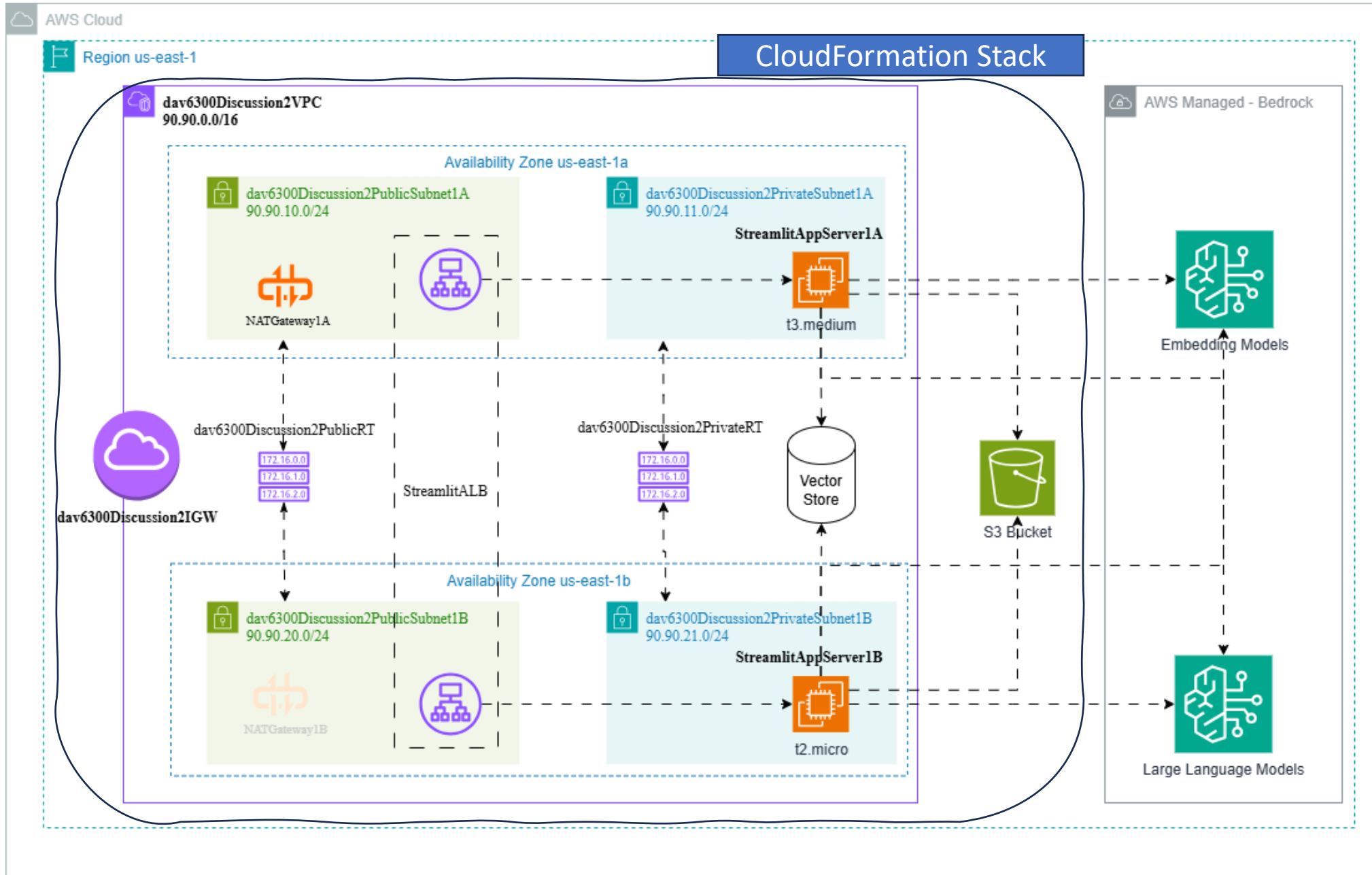
Retrieval-augmented generation through AWS Services Lens



Architecture - 1



Architecture - 2



Overview of AWS CloudFormation – an Infrastructure-as-Code tool

- **AWS CloudFormation** is a service that lets us define the AWS infrastructure resources in a declarative way
- It uses templates in JSON or YAML format.
- It automates the creation, configuration, and management of these resources.
- We can treat the infrastructure like code, making it easier to manage, version, and repeat deployments.

```
AWSTemplateFormatVersion: 2010-09-09
Description: CloudFormation template for s3 bucket
Resources:
  S3Bucket:
    DeletionPolicy: Retain
    Type: 'AWS::S3::Bucket'
    Description: Creating Amazon S3 bucket from CloudFormation
    Properties:
      AccessControl: Private
      PublicAccessBlockConfiguration:
        BlockPublicAcls: true
        BlockPublicPolicy: true
        IgnorePublicAcls: true
        RestrictPublicBuckets: true
      BucketEncryption:
        ServerSideEncryptionConfiguration:
          - ServerSideEncryptionByDefault:
              SSEAlgorithm: AES256
      VersioningConfiguration:
        Status: Enabled
Outputs:
  S3Bucket:
    Description: Bucket Created using this template.
    Value: !Ref S3Bucket
```

Format Version

Identifies the capabilities of the template

MetaData

Additional information about the template

Description

A description of what this template does

Parameters

Values to pass to your template at runtime

Mappings

A lookup table, maps keys to values so you can change your values

Conditions

Whether resources are created or properties are assigned

Transform

Applies macros

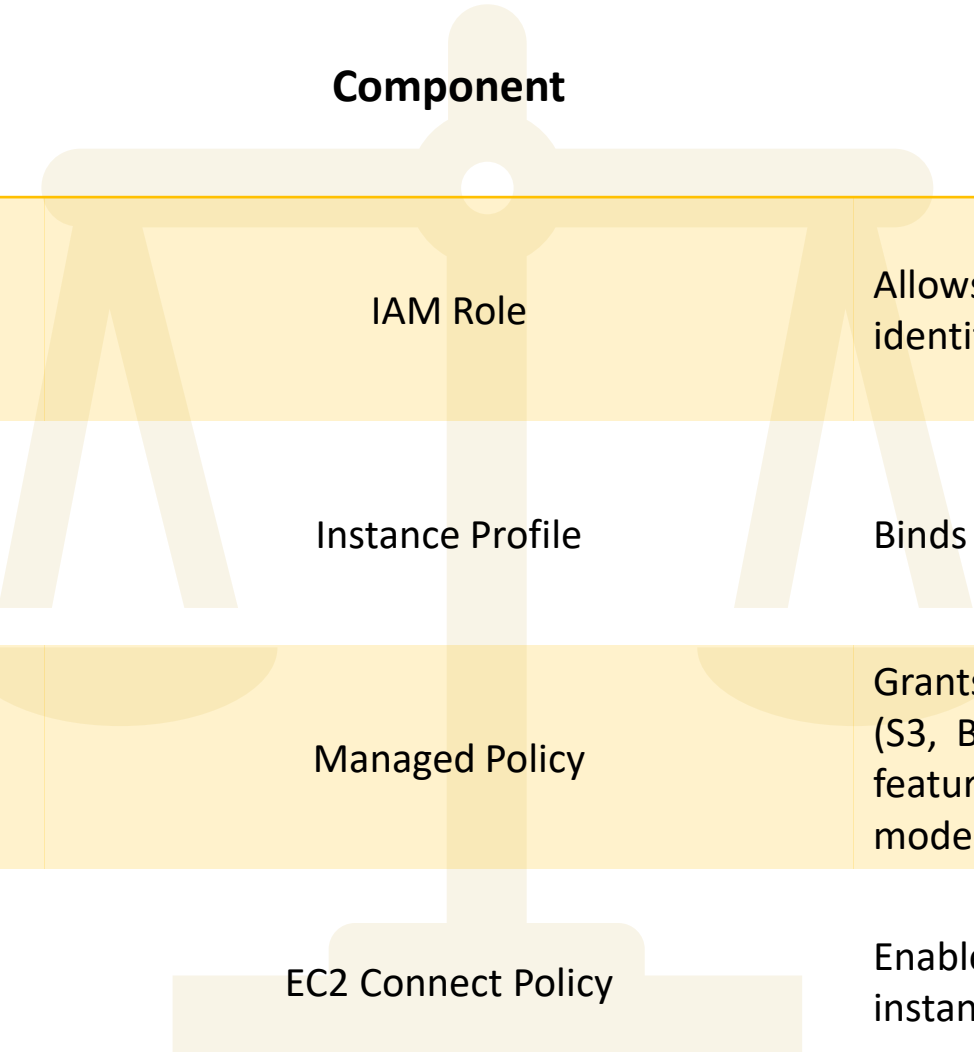
Resources

A resource you want to create

Outputs

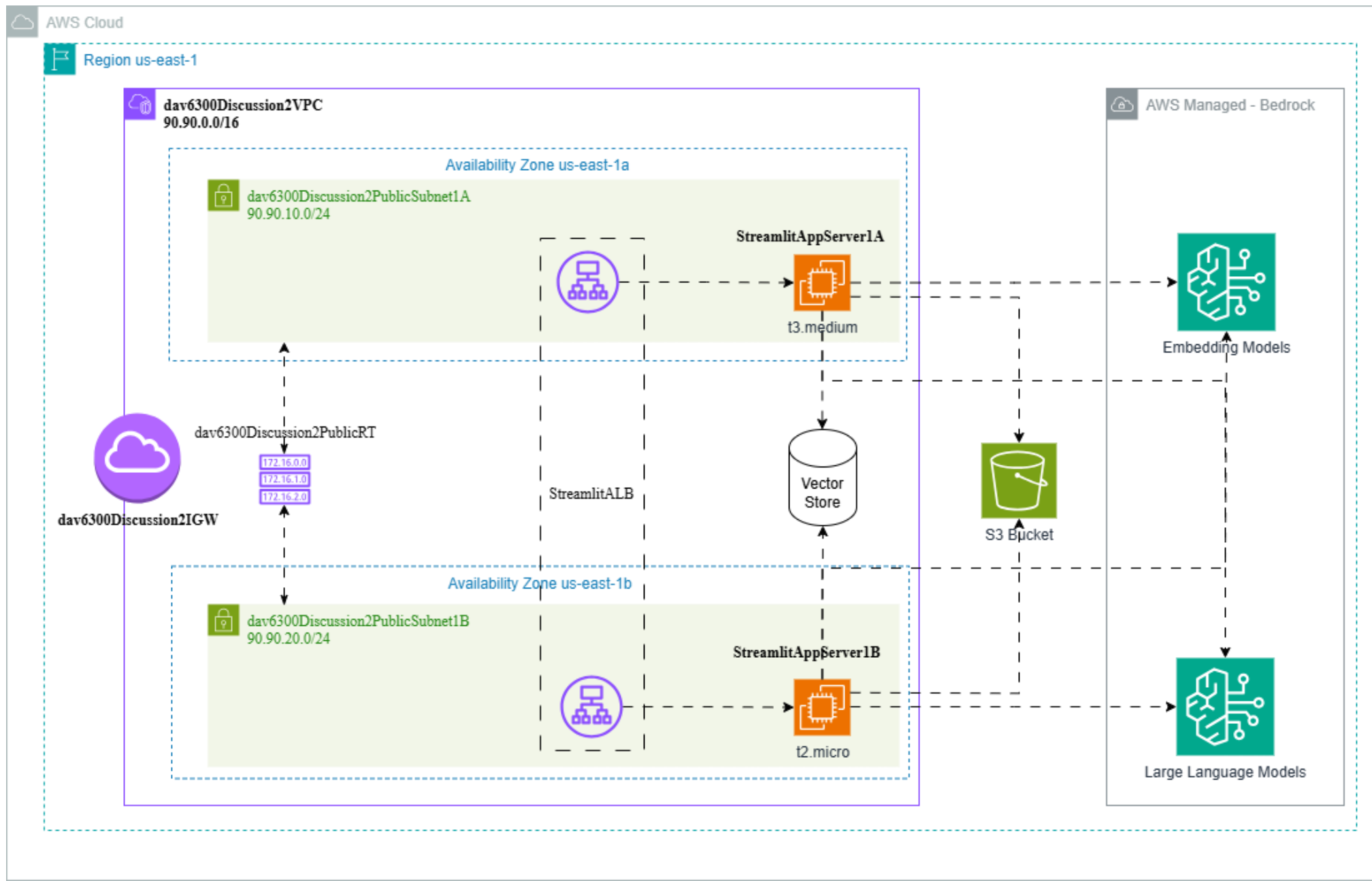
Values that are returned

Comment on IAM Roles, Profile and Policies



Component Name in Stack	Component	Role in the Application
StreamlitAppServerRole	IAM Role	Allows EC2 instance to assume a secure identity and interact with AWS services.
StreamlitAppServerProfile	Instance Profile	Binds the IAM role to the EC2 instance.
StreamlitAppManagedPolicy	Managed Policy	Grants necessary service-level permissions (S3, Bedrock, Textract, etc.) to enable app features like uploading data, invoking AI models, etc.
EC2ConnectCustomerManagedPolicy	EC2 Connect Policy	Enables secure remote access to the EC2 instance via EC2 Instance Connect.

Architecture – What's Running for the Demo



Demo of the Application

DNS Link of Application Load Balancer – **Streamlit ALB**

Running on Port **8501**

Note: Streamlit application doesn't handle all the exceptions as of now. Be mindful about the error/exception. If you encounter one, kindly report to us.

Github Repo - https://github.com/zaskap/rag_chatbot_dav_6300_discussion_2

Setup Steps- 1

The screenshot shows the AWS CloudFormation console in the us-east-1 region. The left sidebar contains navigation links for CloudFormation (Stacks, Stack details, Drifts, StackSets, Exports, Infrastructure Composer, IaC generator, Hooks overview, Hooks) and Registry (Public extensions, Activated extensions, Publisher, Spotlight). The main content area is titled 'Stacks (0)' and features a search bar 'Filter by stack name', a 'Filter status' dropdown set to 'Active', and a 'View nested' toggle. Below the filters is a table header with columns: Stack name, Status, Created time, and Description. The table is currently empty, displaying the message 'No stacks' and 'No stacks to display'. Action buttons at the top right include 'Delete', 'Update stack', 'Stack actions', and 'Create stack'. A 'Create stack' button is also prominently displayed in the center of the empty table area, along with a 'View getting started guide' button. The footer includes 'CloudShell', 'Feedback', and copyright information for Amazon Web Services, Inc. or its affiliates, with links to 'Privacy', 'Terms', and 'Cookie preferences'.

Stacks | CloudFormation | us-east-1

us-east-1.console.aws.amazon.com/cloudformation/home?region=us-east-1#/stacks?filteringStatus=active&filteringText=&stackId=arn%3Aaws%3Acloudformation%3Aus-east-1%3A086...

Search [Alt+S] United States (N. Virginia) askap

CloudFormation > Stacks

Stacks (0)

Filter by stack name

Filter status: Active View nested

Stack name	Status	Created time	Description
No stacks			
No stacks to display			

Create stack

View getting started guide

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

Setup Steps- 2

CloudFormation – Create Stack

us-east-1.console.aws.amazon.com/cloudformation/home?region=us-east-1#/stacks/create

aws

Search

[Alt+S]

United States (N. Virginia)

askap

CloudFormation

Stacks

Create stack

CloudFormation

Stacks

StackSets

Exports

Infrastructure Composer

laC generator

Hooks overview

Hooks

Registry

Public extensions

Activated extensions

Publisher

Spotlight

Feedback

Step 1

Create stack

Step 2

Specify stack details

Step 3

Configure stack options

Step 4

Review and create

Create stack

Prerequisite – Prepare template

You can also create a template by scanning your existing resources in the [laC generator](#).

Prepare template

Every 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.

Choose an existing template

Upload or choose an existing template.

Build from Infrastructure Composer

Create a template using a visual builder.

Specify template

[Info](#)

This [GitHub repository](#) contains sample CloudFormation templates that can help you get started on new infrastructure projects. [Learn more](#)

Template source

Selecting a template generates an Amazon S3 URL where it will be stored. A template is a JSON or YAML file that describes your stack's resources and properties.

Amazon S3 URL

Provide an Amazon S3 URL to your template.

Upload a template file

Upload your template directly to the console.

Sync from Git

Sync a template from your Git repository.

Upload a template file

Choose file

ALB_Setup_for_Chatbot.yml

JSON or YAML formatted file

S3 URL: https://s3.us-east-1.amazonaws.com/cf-templates-12x2l4poj02yi-us-east-1/2025-05-07T210845.581Zps3-ALB_Setup_for_Chatbot.yml

View in Infrastructure Composer

Cancel

Next

CloudShell

Feedback

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

Setup Steps- 2

Infrastructure Composer | us-east-1

us-east-1.console.aws.amazon.com/composer/canvas?region=us-east-1&srcConsole=cloudformation&templatePath=cf-templates-12x2l4poj02yi-us-east-1%2F2025-05-07T210845...

CloudFormation > Infrastructure Composer

Unsaved changes 2025-05-07T210845.581Zps3-ALB_Setup_for_Chatbot.yml CloudFormation console mode Menu

Infrastructure Composer

List Resources

Search for a resource

Enhanced components (14)

- API Gateway
- Cognito UserPool
- Cognito UserPoolClient
- DynamoDB table
- EventBridge event rule
- EventBridge schedule
- Kinesis stream
- Lambda function

Canvas Template Arrange

Valid Create template

Standard component S3 bucket

Standard component StreamlitAppServer1A

- StreamlitAppServerRole
- StreamlitAppServerProfile
- StreamlitAppServer1A
- StreamlitAppServer1B

Standard component StreamlitAppManagedPolicy

- StreamlitAppManagedPolicy

Standard component EC2ConnectCustomerManagedPolicy

- EC2ConnectCustomerManagedPolicy

Standard component StreamlitListener

- StreamlitListener

Standard component StreamlitAppSecurityGroup

- StreamlitAppSecurityGroup

Standard component ALBSecurityGroup

- ALBSecurityGroup

Standard component StreamlitTargetGroup

- StreamlitTargetGroup

Standard component StreamlitALB

- StreamlitALB

Standard component dav6300Discussion2VPC

- dav6300Discussion2PublicSubnet1B
- AssociatePublicSubnet1B
- dav6300Discussion2PublicSubnet1A
- AssociatePublicSubnet1A
- dav6300Discussion2PublicRT
- PublicRoute
- dav6300Discussion2VPC
- AttachIGW
- dav6300Discussion2IGW

CloudShell Feedback

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

Setup Steps– 3

Infrastructure Composer | us-east-1

us-east-1.console.aws.amazon.com/composer/canvas?region=us-east-1&srcConsole=cloudformation&templatePath=cf-templates-12x2l4poj02yi-us-east-1%2F2025-05-07T210845...

Search [Alt+S]

United States (N. Virginia) askap

CloudFormation > Infrastructure Composer

Unsaved changes 2025-05-07T210845.581Zps3-ALB_Setup_for_Chatbot.yml CloudFormation console mode Menu

Infrastructure Composer

List Resources

Search for a resource

Enhanced components (14)

- API Gateway
- Cognito UserPool
- Cognito UserPoolClient
- DynamoDB table
- EventBridge event rule
- EventBridge schedule
- Kinesis stream
- Lambda function

Canvas Template Arrange

Valid Create template

Standard component S3 bucket

Standard component StreamlitAppServer1A

- StreamlitAppServerRole
- StreamlitAppServerProfile
- StreamlitAppServer1A
- StreamlitAppServer1B

Standard component StreamlitAppSecurityGroup

- StreamlitAppSecurityGroup

Standard component ALBSecurityGroup

- ALBSecurityGroup

Standard component StreamlitAppManagedPolicy

- StreamlitAppManagedPolicy

Standard component EC2ConnectCustomerManagedPolicy

- EC2ConnectCustomerManagedPolicy

Standard component StreamlitListener

- StreamlitListener

Standard component StreamlitTargetGroup

- StreamlitTargetGroup

Standard component StreamlitALB

- StreamlitALB

Standard component dav6300Discussion2VPC

- dav6300Discussion2PublicSubnet1B
- AssociatePublicSubnet1B
- dav6300Discussion2PublicSubnet1A
- AssociatePublicSubnet1A
- dav6300Discussion2PublicRT
- PublicRoute
- dav6300Discussion2VPC
- AttachIGW
- dav6300Discussion2IGW

CloudShell Feedback

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

Setup Steps– 4

CloudFormation | us-east-1

us-east-1.console.aws.amazon.com/cloudformation/home?region=us-east-1#/stacks/create

aws

Search

[Alt+S]

United States (N. Virginia)

askap

CloudFormation

Stacks

StackSets

Exports

Infrastructure Composer

laC generator

Hooks overview

Hooks

Registry

Public extensions

Activated extensions

Publisher

Spotlight

Feedback

Step 1

Create stack

Step 2

Specify stack details

Step 3

Configure stack options

Step 4

Review and create

Specify stack details

Provide a stack name

Stack name

StreamlitAppServer

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: 18/128.

Parameters

Parameters are defined in your template and allow you to input custom values when you create or update a stack.

EC2RoleName

Provide the Role Name used for this App Server

StreamlitAppRole

LatestAmild

Latest Amazon Linux 2023 AMI ID

/aws/service/ami-amazon-linux-latest/al2023-ami-kernel-default-x86_64

Cancel

Previous

Next

CloudShell

Feedback

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

Setup Steps– 5.1

The screenshot shows the AWS CloudFormation console in the 'us-east-1' region, specifically the 'Create stack' wizard. The browser address bar shows the URL: `us-east-1.console.aws.amazon.com/cloudformation/home?region=us-east-1#/stacks/create`. The console header includes the AWS logo, a search bar, and navigation links for CloudFormation, Stacks, and Create stack. The left sidebar contains navigation options: CloudFormation, Stacks, StackSets, Exports, Infrastructure Composer, IaC generator, Hooks overview, Hooks, Registry (Public extensions, Activated extensions, Publisher), Spotlight, and Feedback.

The main content area displays the 'Configure stack options' step, which is the third step in a four-step process. The steps are: Step 1: Create stack, Step 2: Specify stack details, Step 3: Configure stack options (current step), and Step 4: Review and create.

The 'Configure stack options' section includes the following options:

- Tags - optional**: Tags (key-value pairs) are used to apply metadata to AWS resources, which can help in organising, identifying and categorising those resources. You can add up to 50 unique tags for each stack. No tags associated with the stack. [Add new tag](#) (You can add 50 more tag(s)).
- Permissions - optional**: Specify an existing AWS Identity and Access Management (IAM) service role that CloudFormation can assume.
 - IAM role - optional**: Choose the IAM role for CloudFormation to use for all operations performed on the stack. The dropdown menu shows 'IAM role name' and 'Sample-role-name'. [Remove](#) and [Add](#) buttons are available.
- Stack failure options**:
 - Behaviour on provisioning failure**: Specify the roll-back behaviour for a stack failure. [Learn more](#).
 - ☒ **Roll back all stack resources**: Roll back the stack to the last known stable state.
 - ☐ **Preserve successfully provisioned resources**: Preserves the state of successfully provisioned resources, while rolling back failed resources to the last known stable state. Resources without a last known stable state will be deleted upon the next stack operation.
 - Delete newly created resources during a rollback**: Specify whether resources that were created during a failed operation should be deleted regardless of their deletion policy. [Learn more](#).
 - ☒ **Use deletion policy**: Retains or deletes created resources according to their attached deletion policy.
 - ☐ **Delete all newly created resources**: Deletes created resources during a rollback regardless of their attached deletion policy.
- Additional settings**: You can set additional options for your stack, like notification options and a stack policy. [Learn more](#).
 - Stack policy - optional**: Defines the resources that you want to protect from unintentional updates during a stack update.

The footer of the console shows 'CloudShell', 'Feedback', and copyright information: '© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences'.

Setup Steps– 5.2

The screenshot shows the AWS CloudFormation console in the 'us-east-1' region, specifically the 'Create stack' page. The browser address bar shows the URL: `us-east-1.console.aws.amazon.com/cloudformation/home?region=us-east-1#/stacks/create`. The AWS navigation bar at the top includes the 'aws' logo, a search bar, and the current region 'United States (N. Virginia)'.

The left-hand navigation pane shows the 'CloudFormation' menu with options like 'Stacks', 'StackSets', 'Exports', 'Infrastructure Composer', 'laC generator', 'Hooks overview', 'Hooks', 'Registry', 'Public extensions', 'Activated extensions', 'Publisher', 'Spotlight', and 'Feedback'.

The main content area is titled 'Create stack' and contains the following sections:

- Preserve successfully provisioned resources**: Preserves the state of successfully provisioned resources, while rolling back failed resources to the last known stable state. Resources without a last known stable state will be deleted upon the next stack operation.
- Delete newly created resources during a rollback**: Specify whether resources that were created during a failed operation should be deleted regardless of their deletion policy. [Learn more](#)
 - ☒ **Use deletion policy**: Retains or deletes created resources according to their attached deletion policy.
 - ☐ **Delete all newly created resources**: Deletes created resources during a rollback regardless of their attached deletion policy.
- Additional settings**: You can set additional options for your stack, like notification options and a stack policy. [Learn more](#)
 - Stack policy - optional**: Defines the resources that you want to protect from unintentional updates during a stack update.
 - Rollback configuration - optional**: Specify alarms for CloudFormation to monitor when creating and updating the stack. If the operation breaches an alarm threshold, CloudFormation rolls it back.
 - Notification options - optional**: Specify a new or existing Amazon Simple Notification Service topic where notifications about stack events are sent.
 - Stack creation options - optional**: Specify the timeout and termination protection options for stack creation.
- Capabilities**
 - The following resource(s) require capabilities: [AWS::IAM::Role]**: This template contains Identity and Access Management (IAM) resources. Check that you want to create each of these resources and that they have the minimum required permissions. In addition, they have customised names. Check that the customised names are unique within your AWS account. [Learn more](#)
 - ☒ I acknowledge that AWS CloudFormation might create IAM resources with customised names.

At the bottom right of the main content area, there are three buttons: 'Cancel', 'Previous', and 'Next'.

The footer of the console shows 'CloudShell', 'Feedback', and copyright information: '© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences'.

Setup Steps– 6.1

The screenshot shows the AWS CloudFormation console in the 'us-east-1' region, specifically the 'Review and create' step for creating a new stack. The breadcrumb navigation shows 'CloudFormation > Stacks > Create stack'. The left sidebar contains navigation links for CloudFormation, Stacks, StackSets, Exports, Infrastructure Composer, IaC generator, Hooks overview, Hooks, Registry, Public extensions, Activated extensions, Publisher, Spotlight, and Feedback.

Review and create

Step 1: Specify template [Edit](#)

Prerequisite – Prepare template

Template

Template URL
https://s3.us-east-1.amazonaws.com/cf-templates-12x2l4poj02yi-us-east-1/2025-05-07T211115.875Zapa-ALB_Setup_for_Chatbot.yml

Stack description
CloudFormation template to create a VPC, subnets, security groups, and an Application Load Balancer (ALB) for a Streamlit application.

Step 2: Specify stack details [Edit](#)

Provide a stack name

Stack name
StreamlitAppServer

Parameters (2)

Search

Key	Value
EC2RoleName	StreamlitAppRole
LatestAmiId	/aws/service/ami-amazon-linux-latest/al2023-ami-kernel-default-x86_64

Step 3: Configure stack options [Edit](#)

Tags

CloudShell Feedback

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

Setup Steps– 6.2

CloudFormation | us-east-1

us-east-1.console.aws.amazon.com/cloudformation/home?region=us-east-1#/stacks/create

aws

Search

[Alt+S]

CloudFormation

Stacks

Create stack

CloudFormation

Stacks

StackSets

Exports

Infrastructure Composer

laC generator

Hooks overview

Hooks

Registry

Public extensions

Activated extensions

Publisher

Spotlight

Feedback

Stack policy

No stack policy

There is no stack policy defined

Rollback configuration

Monitoring time

-

CloudWatch alarm ARN

-

Notification options

SNS topic ARN

No notification options

There are no notification options defined

Stack creation options

Timeout

-

Termination protection

Deactivated

Quick-create link

Use quick-create links to get stacks up and running quickly from the AWS CloudFormation console with the same basic configuration as this stack. Copy the URL on the link to share. [Learn more](#)

Open quick-create link

Create changeset

Cancel

Previous

Submit

CloudShell

Feedback

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

Setup Steps– 7

The screenshot displays the AWS CloudFormation console interface. The browser address bar shows the URL: `us-east-1.console.aws.amazon.com/cloudformation/home?region=us-east-1#/stacks/events?stackId=arn%3Aaws%3Acloudformation%3Aus-east-1%3A086900566281%3Astack%2FS...`. The console header includes the AWS logo, a search bar, and the region set to "United States (N. Virginia)".

The left-hand navigation pane shows the "CloudFormation" section with links to "Stacks", "Stack details", "StackSets", "Exports", "Infrastructure Composer", "laC generator", "Hooks overview", and "Hooks". The "Registry" section is expanded, showing "Public extensions", "Activated extensions", "Publisher", and "Spotlight". A "Feedback" link is at the bottom.

The main content area is divided into two panels. The left panel, titled "Stacks (1)", shows a list of stacks with a filter for "Active" status and a "View nested" toggle. A single stack, "StreamlitAppServer", is listed with a timestamp of "2025-05-07 17:14:19 UTC-0400" and a status of "CREATE_IN_PROGRESS".

The right panel, titled "StreamlitAppServer", displays the stack's details. It includes tabs for "Stack info", "Events", "Resources", "Outputs", "Parameters", "Template", "Changesets", and "Git sync". The "Events" tab is selected, showing a table of events. The table has columns for "Timestamp", "Logical ID", "Status", "Detailed status", and "Status reason". One event is listed with a timestamp of "2025-05-07 17:14:19 UTC-0400", a logical ID of "StreamlitAppServer", a status of "CREATE_IN_PROGRESS", and a status reason of "User Initiated".

The footer of the console shows "CloudShell" and "Feedback" links on the left, and copyright information "© 2025, Amazon Web Services, Inc. or its affiliates." along with "Privacy", "Terms", and "Cookie preferences" links on the right.

Setup Steps– 8

The screenshot displays the AWS CloudFormation console interface. The top navigation bar shows the 'CloudFormation' service and the 'Stacks' page for the 'StreamlitAppServer' stack. The region is set to 'us-east-1'. The left sidebar contains navigation links for 'Stacks', 'Stack details', 'StackSets', 'Exports', 'Infrastructure Composer', 'laC generator', 'Hooks overview', 'Hooks', 'Registry', 'Public extensions', 'Activated extensions', 'Publisher', 'Spotlight', and 'Feedback'.

The main content area is divided into two panels. The left panel, titled 'Stacks (1)', shows a list of stacks. The 'StreamlitAppServer' stack is highlighted, with a status of 'CREATE_IN_PROGRESS' and a creation time of '2025-05-07 17:14:19 UTC-0400'. The right panel, titled 'StreamlitAppServer', shows the 'Events' tab. The events table lists 45 events, with the following columns: Timestamp, Logical ID, Status, Detailed status, and Status reason.

Timestamp	Logical ID	Status	Detailed status	Status reason
2025-05-07 17:15:22 UTC-0400	dav6300Discussion2PublicRT	CREATE_COMPLETE	-	-
2025-05-07 17:15:21 UTC-0400	StreamlitALB	CREATE_IN_PROGRESS	-	-
2025-05-07 17:15:20 UTC-0400	StreamlitAppSecurityGroup	CREATE_IN_PROGRESS	-	-
2025-05-07 17:15:20 UTC-0400	ALBSecurityGroup	CREATE_COMPLETE	-	-
2025-05-07 17:15:17 UTC-0400	AssociatePublicSubnet1B	CREATE_COMPLETE	-	-
2025-05-07 17:15:17 UTC-0400	AssociatePublicSubnet1A	CREATE_COMPLETE	-	-
2025-05-07 17:15:17 UTC-0400	StreamlitAppManagedPolicy	CREATE_IN_PROGRESS	CONFIGURATION_COMPLETE	Eventual consistency check initiated
2025-05-07 17:15:17 UTC-0400	AssociatePublicSubnet1A	CREATE_IN_PROGRESS	-	Resource creation Initiated
2025-05-07 17:15:17 UTC-0400	AssociatePublicSubnet1B	CREATE_IN_PROGRESS	-	Resource creation Initiated
2025-05-07 17:15:16 UTC-0400	StreamlitAppManagedPolicy	CREATE_IN_PROGRESS	-	Resource creation Initiated
2025-05-07 17:15:16 UTC-0400	StreamlitAppServerProfile	CREATE_IN_PROGRESS	-	Resource creation Initiated
2025-05-07 17:15:15 UTC-0400	AssociatePublicSubnet1A	CREATE_IN_PROGRESS	-	-
2025-05-07 17:15:15 UTC-0400	AssociatePublicSubnet1B	CREATE_IN_PROGRESS	-	-
2025-05-07 17:15:15 UTC-0400	StreamlitAppServerProfile	CREATE_IN_PROGRESS	-	-
2025-05-07 17:15:15 UTC-0400	StreamlitAppManagedPolicy	CREATE_IN_PROGRESS	-	-

The bottom of the console shows the 'CloudShell' and 'Feedback' buttons. The footer contains the copyright notice: '© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences'.

Setup Steps– 9

The screenshot displays the AWS CloudFormation console interface. The browser address bar shows the URL: `us-east-1.console.aws.amazon.com/cloudformation/home?region=us-east-1#/stacks/resources?stackId=arn%3Aaws%3Acloudformation%3Aus-east-1%3A086900566281%3Astack%...`. The console header includes the AWS logo, a search bar, and the region "United States (N. Virginia)".

The left sidebar contains navigation options: CloudFormation, Stacks, StreamlitAppServer, Stack details, Drifts, StackSets, Exports, Infrastructure Composer, laC generator, Hooks overview, Hooks, Registry, Public extensions, Activated extensions, Publisher, Spotlight, and Feedback.

The main content area is divided into two panels. The left panel, titled "Stacks (1)", shows a list of stacks with a filter for "Active" and a "View nested" toggle. The right panel, titled "StreamlitAppServer", displays the "Resources (17)" tab. The resources are listed in a table with columns: Logical ID, Physical ID, Type, Status, and Module.

Logical ID	Physical ID	Type	Status	Module
ALBSecurityGroup	sg-098bcf61bdf200c61	AWS::EC2::SecurityGroup	CREATE_COMPLETE	-
AssociatePublicSubnet1A	rtbassoc-09f45fa5856eafc6b	AWS::EC2::SubnetRouteTableAssociation	CREATE_COMPLETE	-
AssociatePublicSubnet1B	rtbassoc-0115d0537af3d1fc1	AWS::EC2::SubnetRouteTableAssociation	CREATE_COMPLETE	-
AttachIGW	IGW\yvc-Oa2114b1fcd1774ff	AWS::EC2::VPCGatewayAttachment	CREATE_COMPLETE	-
dav6300Discussion2IGW	igw-0e0d9608936085c72	AWS::EC2::InternetGateway	CREATE_COMPLETE	-
dav6300Discussion2PublicRT	rtb-051b45fcc5696e20a	AWS::EC2::RouteTable	CREATE_COMPLETE	-
dav6300Discussion2PublicSubnet1A	subnet-056b410850e642f0d	AWS::EC2::Subnet	CREATE_COMPLETE	-
dav6300Discussion2PublicSubnet1B	subnet-03b3ccb1a50f96f8b	AWS::EC2::Subnet	CREATE_COMPLETE	-
dav6300Discussion2VPC	vpc-Oa2114b1fcd1774ff	AWS::EC2::VPC	CREATE_COMPLETE	-
PublicRoute	rtb-051b45fcc5696e20a 0.0.0.0/0	AWS::EC2::Route	CREATE_COMPLETE	-
S3Bucket	streamlittappserver-s3bucket-rphbf3mvj42b	AWS::S3::Bucket	CREATE_COMPLETE	-
StreamlitALB	arn:aws:elasticloadbalancing:us-east-1:086900566281:loadbalancer/app/Stream-StreamlitAppServer-4372d23fd181	AWS::ElasticLoadBalancingV2::LoadBalancer	CREATE_IN_PROGRESS	-

The footer of the console shows "CloudShell", "Feedback", and copyright information: "© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences".

Setup Steps– 10

The screenshot shows the AWS Management Console interface. The browser tabs at the top include 'CloudFormation – Stack St...', 'Instances | EC2 | us-east-1', 'EC2 Instance Connect | us...', 'Instances | EC2 | us-east-1', 'EC2 Instance Connect | us...', and 'streamlitappserver-s3buck...'. The address bar shows the URL: 'us-east-1.console.aws.amazon.com/s3/buckets/streamlitappserver-s3bucket-vnxkaynqjsng?region=us-east-1&bucketType=general&tab=objects'. The AWS logo and a search bar are visible in the top navigation bar. The left sidebar shows the 'Amazon S3' menu with options like 'General purpose buckets', 'Directory buckets', 'Table buckets', 'Access Grants', 'Access Points', 'Object Lambda Access Points', 'Multi-Region Access Points', 'Batch Operations', 'IAM Access Analyzer for S3', 'Storage Lens', and 'Feature spotlight 11'. The main content area displays a green success message: 'Successfully created folder "gen-ai-qa".' Below this, the bucket name 'streamlitappserver-s3bucket-vnxkaynqjsng' is shown with an 'Info' link. The 'Objects' tab is selected, showing a list of objects. The list contains one item: a folder named 'gen-ai-qa/'. The table has columns for Name, Type, Last modified, Size, and Storage class. The bottom of the console shows 'CloudShell' and 'Feedback' links, along with copyright information and links to 'Privacy', 'Terms', and 'Cookie preferences'.

CloudFormation – Stack St... Instances | EC2 | us-east-1 EC2 Instance Connect | us... Instances | EC2 | us-east-1 EC2 Instance Connect | us... streamlitappserver-s3buck...

us-east-1.console.aws.amazon.com/s3/buckets/streamlitappserver-s3bucket-vnxkaynqjsng?region=us-east-1&bucketType=general&tab=objects

aws Search [Alt+S] United States (N. Virginia) askap

Amazon S3 > Buckets > streamlitappserver-s3bucket-vnxkaynqjsng

Amazon S3

General purpose buckets

Directory buckets

Table buckets

Access Grants

Access Points

Object Lambda Access Points

Multi-Region Access Points

Batch Operations

IAM Access Analyzer for S3

Block Public Access settings for this account

▼ Storage Lens

Dashboards

Storage Lens groups

AWS Organizations settings

Feature spotlight 11

Successfully created folder "gen-ai-qa".

streamlitappserver-s3bucket-vnxkaynqjsng Info

Objects Metadata Properties Permissions Metrics Management Access Points

Objects (1) Copy S3 URI Copy URL Download Open Delete Actions Create folder Upload

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

Find objects by prefix

	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	gen-ai-qa/	Folder	-	-	-

CloudShell Feedback

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

Setup Steps– 11

The screenshot displays the AWS Management Console interface for configuring a target group. The breadcrumb navigation shows the path: **EC2** > **Target groups** > **StreamlitWebServers**.

Summary Metrics:

- Total targets: 2
- Healthy: 0
- Unhealthy: 2
- Unused: 0
- Initial: 0
- Draining: 0
- Anomalous: 0

Distribution of targets by Availability Zone (AZ)

Select values in this table to see corresponding filters applied to the Registered targets table below.

Navigation Tabs: Targets (selected), Monitoring, Health checks, Attributes, Tags

Registered targets (2) [Info](#)

Anomaly mitigation: **Not applicable** [Refresh](#) [Deregister](#) [Register targets](#)

Target groups route requests to individual registered targets using the protocol and port number specified. Health checks are performed on all registered targets according to the target group's health check settings. Anomaly detection is automatically applied to HTTP/HTTPS target groups with at least 3 healthy targets.

<input type="checkbox"/>	Instance ID	Name	Port	Zone	Health status	Health status details	Administr...
<input type="checkbox"/>	i-07346c7d24cdcc870	StreamlitApp_...	8501	us-east-1a (use1-az2)	Unhealthy	Health checks failed	No override
<input type="checkbox"/>	i-0484e730f79a4e342	StreamlitApp_...	8501	us-east-1b (use1-az4)	Unhealthy	Health checks failed	No override

Footer: CloudShell Feedback | © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Termination Steps – 1

CloudFormDelete objConsole HLoad balanInstance dEC2 InstanEC2 InstanRAG_ChatlYeshiva Un+us-east-1.console.aws.amazon.com/s3/buckets/streamlitappserver-s3bucket-rphbf3mvj42b/object/delete?region=us-east-1&bucketType=general&showversions=falseSearch[Alt+S]United States (N. Virginia)askap

Amazon S3> Buckets> streamlitappserver-s3bucket-rphbf3mvj42b> Delete objects


Delete objects [Info](#)

⚠

- If a folder is selected for deletion, all objects in the folder will be deleted, and any new objects added while the delete action is in progress might also be deleted. If an object is selected for deletion, any new objects with the same name that are uploaded before the delete action is completed will also be deleted.
- Deleting the specified objects can't be undone.

[Learn more](#)

Specified objects

Name	Type	Last modified	Size
 gen-ai-qa/	Folder	-	-

Permanently delete objects?

To confirm deletion, type *permanently delete* in the text input field.

Cancel

Delete objects

CloudShellFeedback

© 2025, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

Termination Steps – 2

The screenshot shows the AWS CloudFormation console in the 'us-east-1' region. The 'StreamlitAppServer' stack is selected, and the 'Resources' tab is active. A modal dialog titled 'Delete stack?' is displayed, asking for confirmation to delete the stack permanently. The dialog includes a warning message and a link to learn more about deletion policies. The background shows the stack's resources, including various AWS services like EC2, S3, and IAM.

Delete stack?

Delete stack **StreamlitAppServer** permanently? This action cannot be undone.

Deleting this stack will delete all stack resources. Resources will be deleted according to their DeletionPolicy. [Learn more](#)

[Cancel](#) [Delete](#)

Resource Name	Physical ID	Resource Type	Status	Module
AttachIGW	IGW/vpc-0a2114b1fcd1774ff	AWS::EC2::VPCGatewayAttachment	CREATE_COMPLETE	-
dav6300Discussion2IGW	igw-0e0d9608936085c72	AWS::EC2::InternetGateway	CREATE_COMPLETE	-
dav6300Discussion2PublicRT	rtb-051b45fcc5696e20a	AWS::EC2::RouteTable	CREATE_COMPLETE	-
dav6300Discussion2PublicSubnet1A	subnet-056b410850e642f0d	AWS::EC2::Subnet	CREATE_COMPLETE	-
dav6300Discussion2PublicSubnet1R	subnet-03h3cch1a50f96f8h	AWS::EC2::Subnet	CREATE_COMPLETE	-

Termination Steps – 3

The screenshot shows the AWS CloudFormation console interface. The browser address bar indicates the user is in the us-east-1 region, viewing the Stack events for the StreamlitAppServer stack. The left sidebar contains navigation options for CloudFormation, including Stacks, Stack details, Drifts, StackSets, Exports, Infrastructure Composer, IaC generator, Hooks overview, and Hooks. The Registry section is also visible, showing Public extensions, Activated extensions, and Publisher. The main content area displays the StreamlitAppServer stack details, including a notification that deletion has been initiated for the stack. The Events tab is selected, showing a list of events with columns for Timestamp, Logical ID, Status, Detailed status, and Status reason. The events list shows that the stack is in a DELETED state, with the status reason being 'DELETED_IN_PROGRESS'.

CloudFormation > **Stacks** > StreamlitAppServer

Stacks (1)

Filter by stack name

Filter status: Active View nested

Stacks

StreamlitAppServer

2025-05-07 17:14:19 UTC-0400

DELETED_IN_PROGRESS

StreamlitAppServer

Delete Update stack Stack actions Create stack

Stack info **Events** Resources Outputs Parameters Template Changesets Git sync

Table view Timeline view

Events (80)

View root cause

Search events

Timestamp	Logical ID	Status	Detailed status	Status reason
2025-05-07 17:47:42 UTC-0400	S3Bucket	DELETED_IN_PROGRESS	-	-
2025-05-07 17:47:42 UTC-0400	PublicRoute	DELETED_IN_PROGRESS	-	-
2025-05-07 17:47:42 UTC-0400	EC2ConnectCustomerManagedPolicy	DELETED_IN_PROGRESS	-	-
2025-05-07 17:47:42 UTC-0400	AssociatePublicSubnet1B	DELETED_IN_PROGRESS	-	-
2025-05-07 17:47:42 UTC-0400	AssociatePublicSubnet1A	DELETED_IN_PROGRESS	-	-
2025-05-07 17:47:42 UTC-0400	StreamlitAppManagedPolicy	DELETED_IN_PROGRESS	-	-

CloudShell Feedback

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

Areas of Improvement

Topic	Area	Improvement Suggestion	Benefit
Cloud Formation Template Design	Modularization	Split the template into nested stacks for VPC, EC2, IAM, etc.	Enhances maintainability, reusability, and clarity.
	Parameterization	Use parameters for instance types, subnet IDs, AMI IDs, etc.	Enables flexibility and reuse across environments.
	Use of Mappings & Conditions	Use mapping for region-specific values, conditions for optional resources.	Helps adapt template to different regions and use cases.
	Outputs & Exports	Add meaningful outputs (e.g., ALB DNS, EC2 IPs) for downstream stacks.	Improves integration with other stacks or tools.
Services Used	EC2 (Streamlit)	Consider using Fargate with ECS or Elastic Beanstalk for better scalability and less ops overhead.	Reduces EC2 maintenance, auto-scales more easily.
	Vector Store (OpenSearch)	Optionally use Amazon Aurora PostgreSQL + pgvector if relational metadata is needed.	Better for hybrid use cases (structured + vector).
	Textract & Transcribe	Use event-driven processing (e.g., with S3 + Lambda) to make it more scalable.	Decouples architecture, improves resilience.
Latency and Optimization	CloudWatch	Use invocation logging and latency metrics from CloudWatch to monitor spikes.	Helps detect slow responses or overuse.
	ALB to EC2 Communication	Use Connection Draining and Health Checks to reduce user-facing latency.	Smooth traffic transitions and avoids cold EC2s.
	Private Subnet Access	Use VPC Endpoints (Interface) for Bedrock/Textract if supported.	Avoids public internet latency and saves NAT cost.
	Caching	Add Amazon ElastiCache (Redis) to cache frequent queries or document embeddings.	Reduces repetitive Bedrock/OpenSearch calls.

References

1. [Build a conversational chatbot using different LLMs within single interface – Part 1](#)
2. [Deploy a Dynamic Website on AWS with CloudFormation](#)
3. [Building a Secure and Scalable Django Blog on AWS: The Ultimate Guide](#)
4. [AWS CloudFormation User Guide](#)
5. [Anatomy of efficient CloudFormation templates for large-scale automated testing, AIOps, MLOps, etc.](#)
6. [AWS CloudFormation template explained](#)



Thank You!
Any Questions

Appendix

Some Output Snippets–1

Serverless & DNS & CF.pptx Presentation from Cloud Computing Class

The content covered in the deck includes:

- **DNS and Amazon Route 53**:**
 - Bandwidth vs. latency definitions and factors affecting latency (propagation delay, transmission delay, queueing delay, processing delay).
 - DNS basics: IP addresses, DNS zones, FQDNs, subdomains, DNS record types (A, CNAME, MX, TXT, SRV, NS, SOA).
 - Route 53 features: hosted zones, domain registration, DNS resolution, routing policies (Simple, Failover, Geolocation, Geoproximity, Latency, Weighted, Multivalue, IP-Based).
- **Amazon CloudFront**:**
 - Content Delivery Network (CDN) functionality, edge locations, caching, origins (S3/custom).
 - Integration with AWS Global Network, Lambda@Edge, security (HTTPS, AWS WAF, AWS Shield, signed URLs/cookies).
- **Containers and Serverless Computing**:**
 - Docker containers vs. server virtualization, Amazon ECS components (clusters, tasks, services, task definitions), ECR, launch types (EC2 vs. Fargate).
 - AWS Lambda: event-driven execution, supported languages, invocation types (synchronous/asynchronous), integration with API Gateway.
 - Amazon API Gateway: REST API creation, resource/method configuration, integration with Lambda, DynamoDB, and other services.
- **Demos**:**
 - DNS configuration (Cloudflare + Route 53), S3 static website with CloudFront.
 - Containerizing a Django app with ECS/ECR.
 - Lambda function creation and API Gateway integration.

The deck focuses on DNS, CDN, container orchestration, serverless architectures, and AWS services like Route 53, CloudFront, ECS, Lambda, and API Gateway.

Some Output Snippets-2

Image of AWS Architecture was uploaded and asked to summarize

The screenshot shows a web browser window with multiple tabs. The active tab is titled "RAG_Chatbot". The address bar shows a URL: "stream-strea-siyyjc8estmt-1657881458.us-east-1.elb.amazonaws.com:8501". The page has a dark theme. On the left side, there is a sidebar with a progress bar at the top (0.00 to 1.00) and a "Temperature" slider (0.01 to 1.00). Below these, there is a section titled "Choose the Input Method" with radio buttons for "Document", "Weblink", "Audio", "Image" (selected), and "PPT". At the bottom of the sidebar, there is a "Text from Image" section with a "Select File" button and a "Browse files" button. A file named "Screenshot 2025-05..." (257.8KB) is shown. The main content area has a heading "Select the type of summary you want to see" with radio buttons for "Summary" (selected), "Key Points", "Sample Questions", and "Extracted Text". Below this is a "Submit" button. The main content area also has a large heading "Summary" and three paragraphs of text. The first paragraph describes the AWS CloudFormation setup, the second paragraph describes the health status of the instances, and the third paragraph describes the Target Group configuration. At the bottom of the page, there is a footer with the text "#Tokens: #Words: #Embeddi".

CloudFormati x Target group x Load balance x Instance deta x EC2 Instance x Instance deta x EC2 Instance x RAG_Chatbot x Yeshiva Univ x

Not secure stream-strea-siyyjc8estmt-1657881458.us-east-1.elb.amazonaws.com:8501

1.00
0.00 1.00

Temperature
0.01
0.00 1.00

Choose the Input Method

☐ Document
☐ Weblink
☐ Audio
☒ Image
☐ PPT

Text from Image

Select File

Drag and drop file here
Limit 200MB per file • JPG, JPEG, PNG

Browse files

Screenshot 2025-05...
257.8KB

Select the type of summary you want to see

☒ Summary ☐ Key Points ☐ Sample Questions ☐ Extracted Text

Submit

Summary

The summaries provided outline an AWS CloudFormation setup focusing on a Load Balancer and its associated Target Groups, which are connected to EC2 Instances. The Load Balancer is situated in AWS's N. Virginia region, with a Target Group named "StreamlitWebServers". This Target Group is associated with two EC2 Instances, located in us-east-1a (use1-az2) and us-east-1b (use1-az4). Both instances are of the type "StreamlitApp" and are operational on port 8501.

Regrettably, both instances are currently unhealthy, attributed to failed health checks. There are no override settings for these health checks. Each instance is linked with an Elastic Block Store (EBS) volume, but no specifics regarding snapshots or lifecycle manager details are mentioned.

The setup does not include any anomaly mitigation strategies, and there are no Capacity Reservations, Dedicated Hosts, Savings Plans, or Reserved Instances associated with these instances.

The Target Group is configured to distribute requests to individual targets using the HTTP/HTTPS protocol. Health checks are conducted according to the Target Group's settings. Anomaly detection is applied to this Target Group due to having a minimum of 3 healthy targets. However, the current unhealthy state of the instances might be affecting this anom

#Tokens: #Words: #Embeddi

Some Output Snippets-2

Key Points from an uploaded PDF document was summarized

The screenshot displays a web browser window with multiple tabs. The active tab is titled "RAG_Chatbot" and shows a web application interface. The browser's address bar indicates a "Not secure" connection to "stream-strea-siyyjc8estmt-1657881458.us-east-1.elb.amazonaws.com:8501".

The application interface has a dark theme. On the left side, there is a sidebar with a "Choose the Input Method" section containing radio buttons for "Document" (selected), "Weblink", "Audio", "Image", and "PPT". Below this is a "Documents" section with a "Select File" button and a "Drag and drop file here" area. A file named "Assignment 10 - Mor..." (70.2KB) is shown as uploaded. At the bottom of the sidebar, there are counters for "#Tokens: 672", "#Words: 499", and "#Embeddi: 0", along with a "Contact" button.

The main content area features a navigation bar with "QnA", "Document Summary" (highlighted), and "About RAG_Chatbot". Below the navigation bar, there is a section titled "Select the type of summary you want to see" with radio buttons for "Summary", "Key Points" (selected), "Sample Questions", and "Extracted Text". A "Submit" button is located below these options.

The "Key Points" section displays a list of bullet points summarizing the document's content:

- Paper focuses on AI transparency in financial credit risk assessment
- Combines Shapley values with Minimal Spanning Trees for model interpretation
- Uses model-agnostic approach in post-processing phase
- Dataset includes 15,045 SMEs from Southern Europe (2015-2016)
- 10.9% default rate in dataset
- 80/20 training-testing split
- XGBoost achieved AUROC of 0.93
- Logistic Regression achieved AUROC of 0.81
- Defaulted and non-defaulted companies form distinct clusters