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

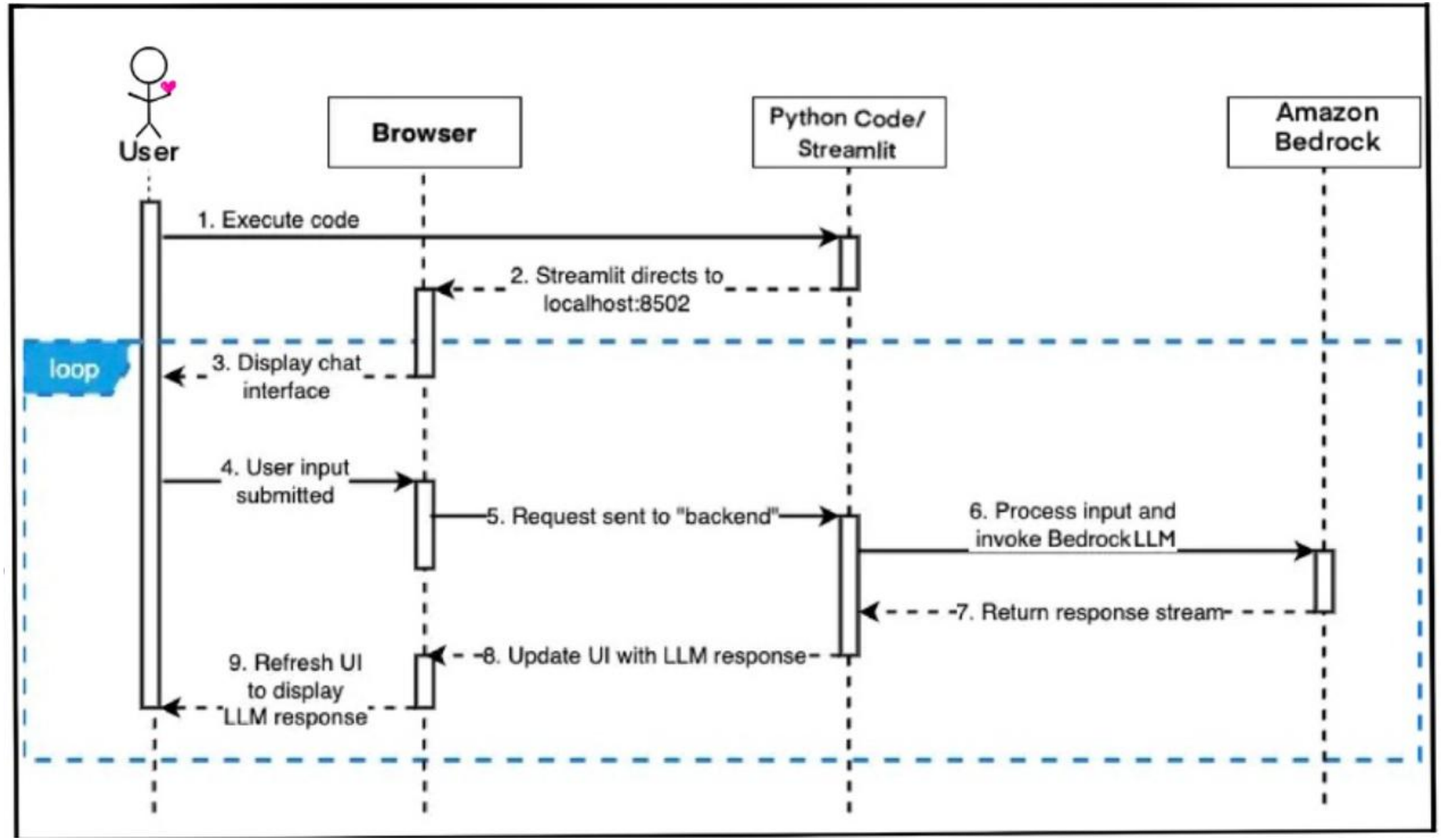
AWS Service Used – 1

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

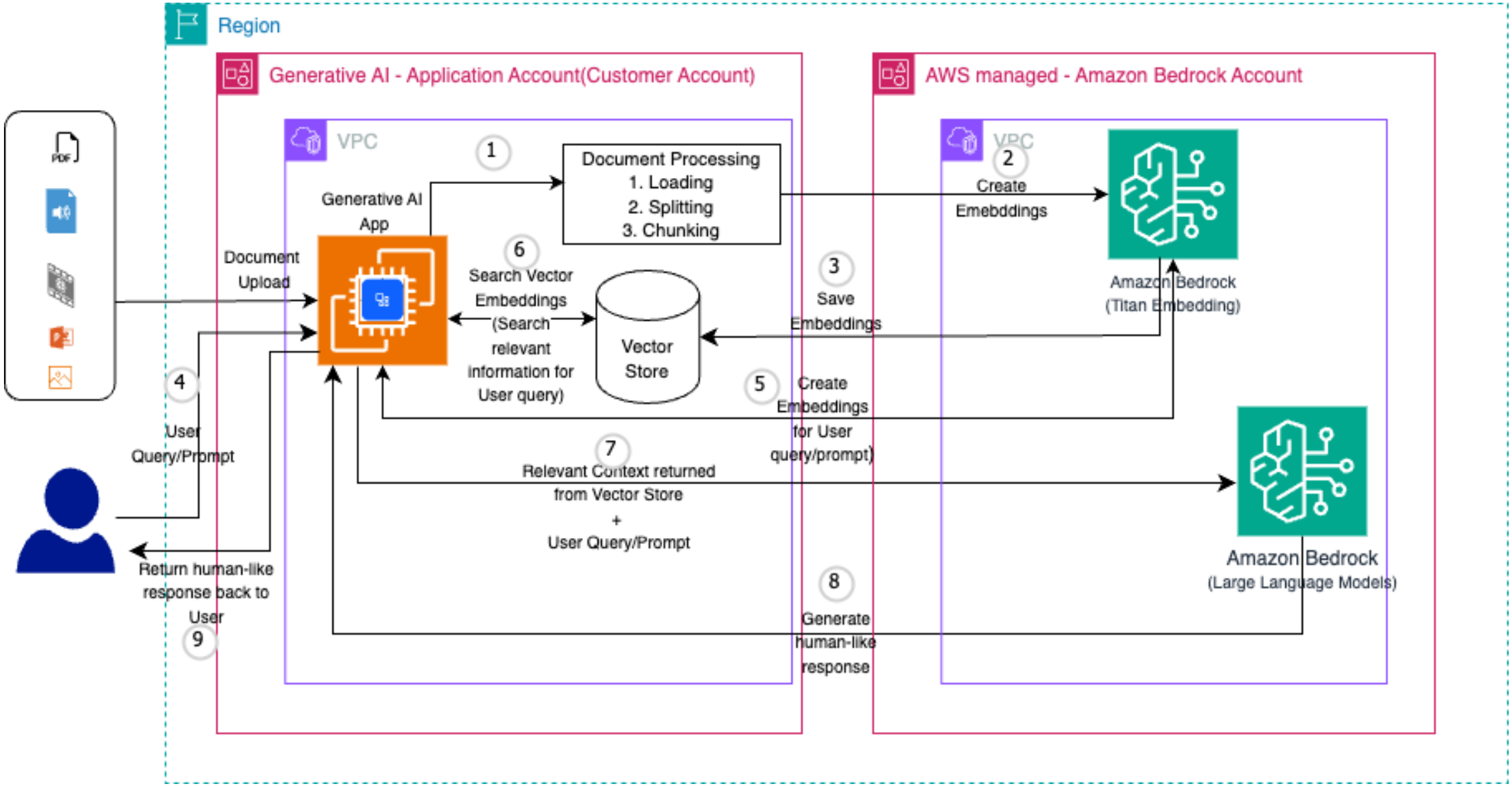
AWS Service	Purpose	Why Used
Application Load Balancer	Distributes traffic across EC2 instances.	Ensures high availability, auto-scaling, and failover support.
AWS Textract	Extracts text and data from documents.	Automates document processing in the RAG pipeline.
Amazon Transcribe	Converts speech in audio 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



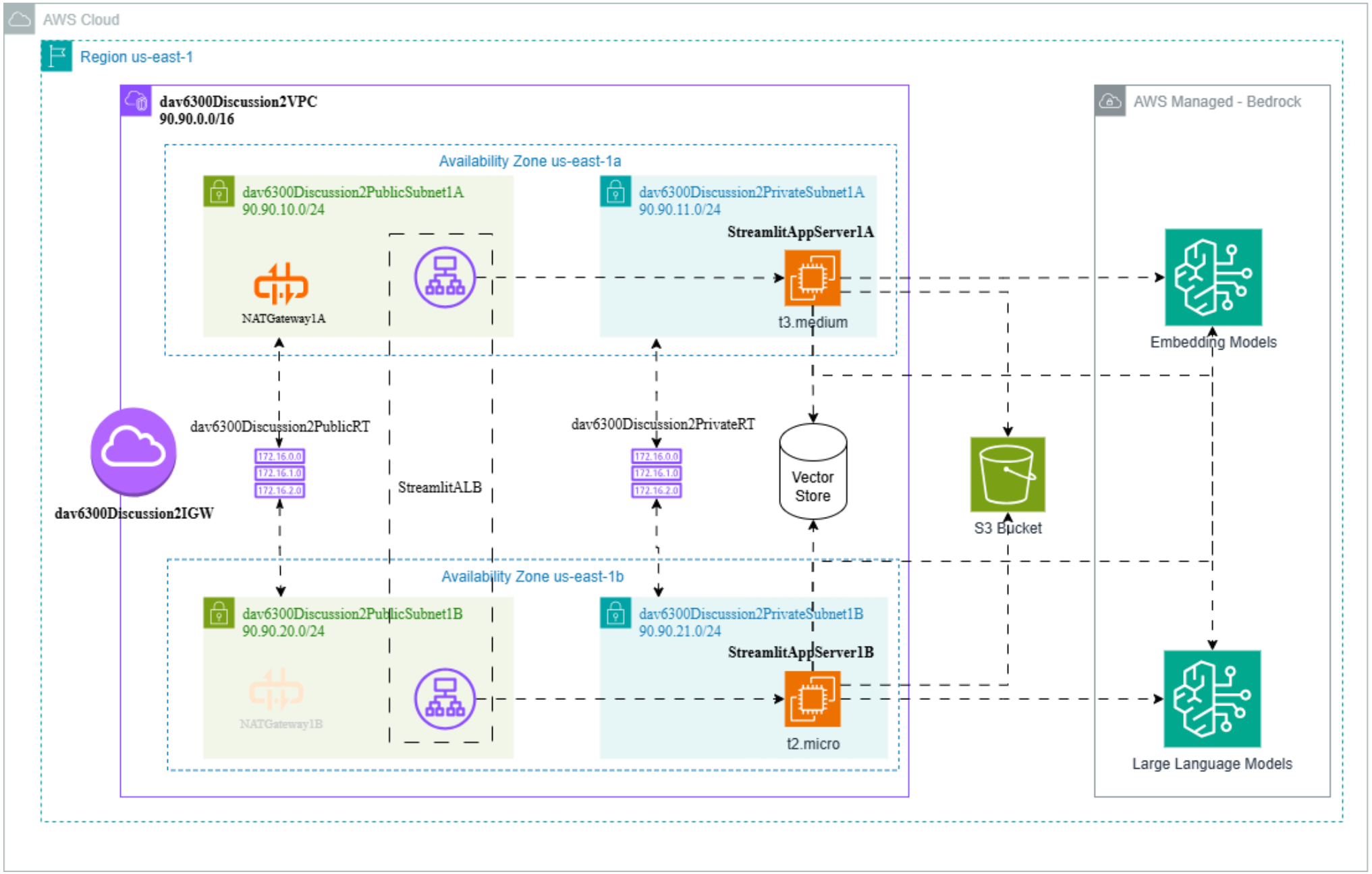
* Courtesy – Anand Mandilwar [1]

Retrieval-augmented generation through AWS Services Lens

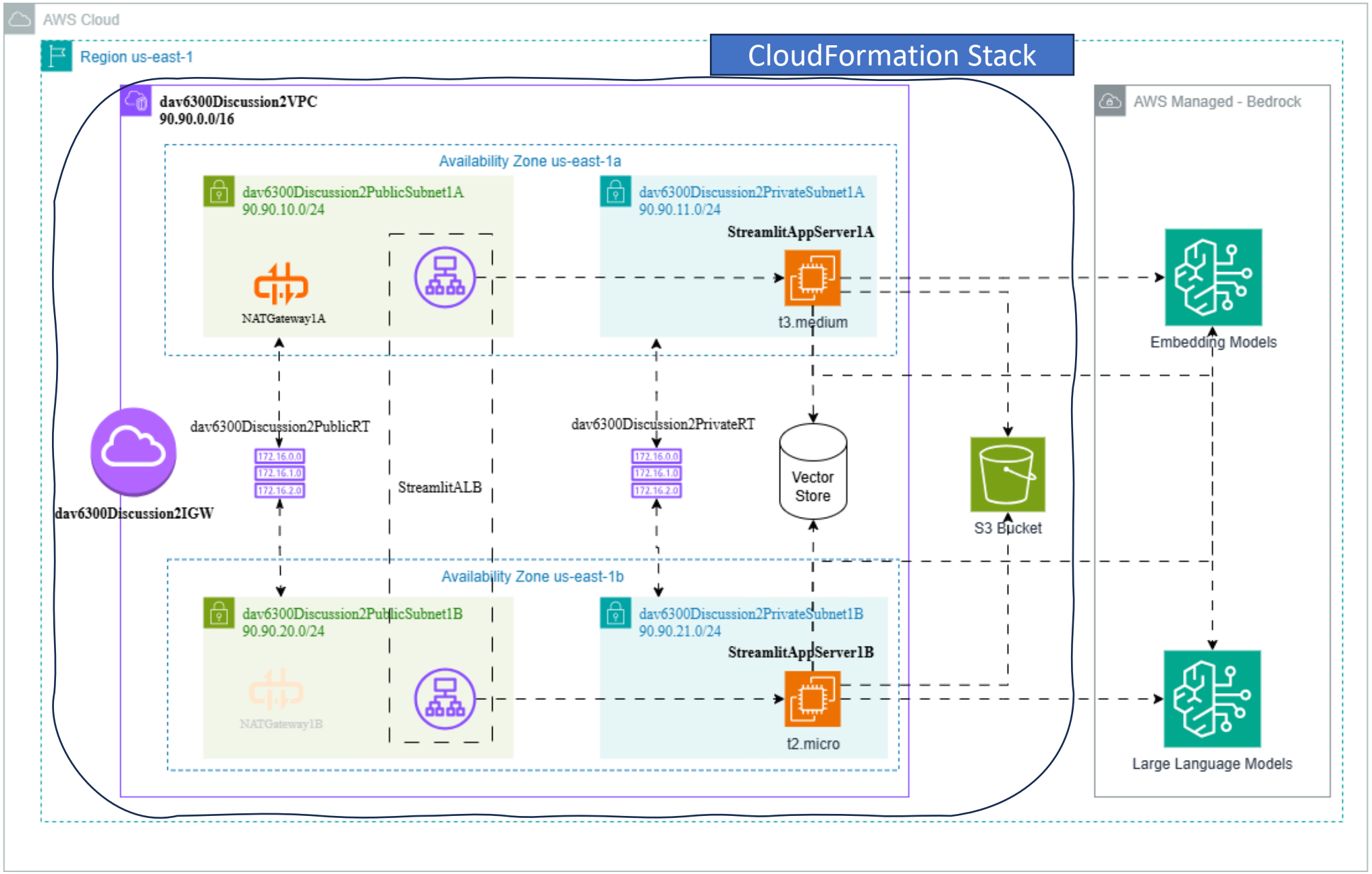


* Courtesy – Anand Mandilwar [1]

Architecture – 1



Architecture – 2



Overview of AWS Cloud Formation – 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.

```yaml
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
```

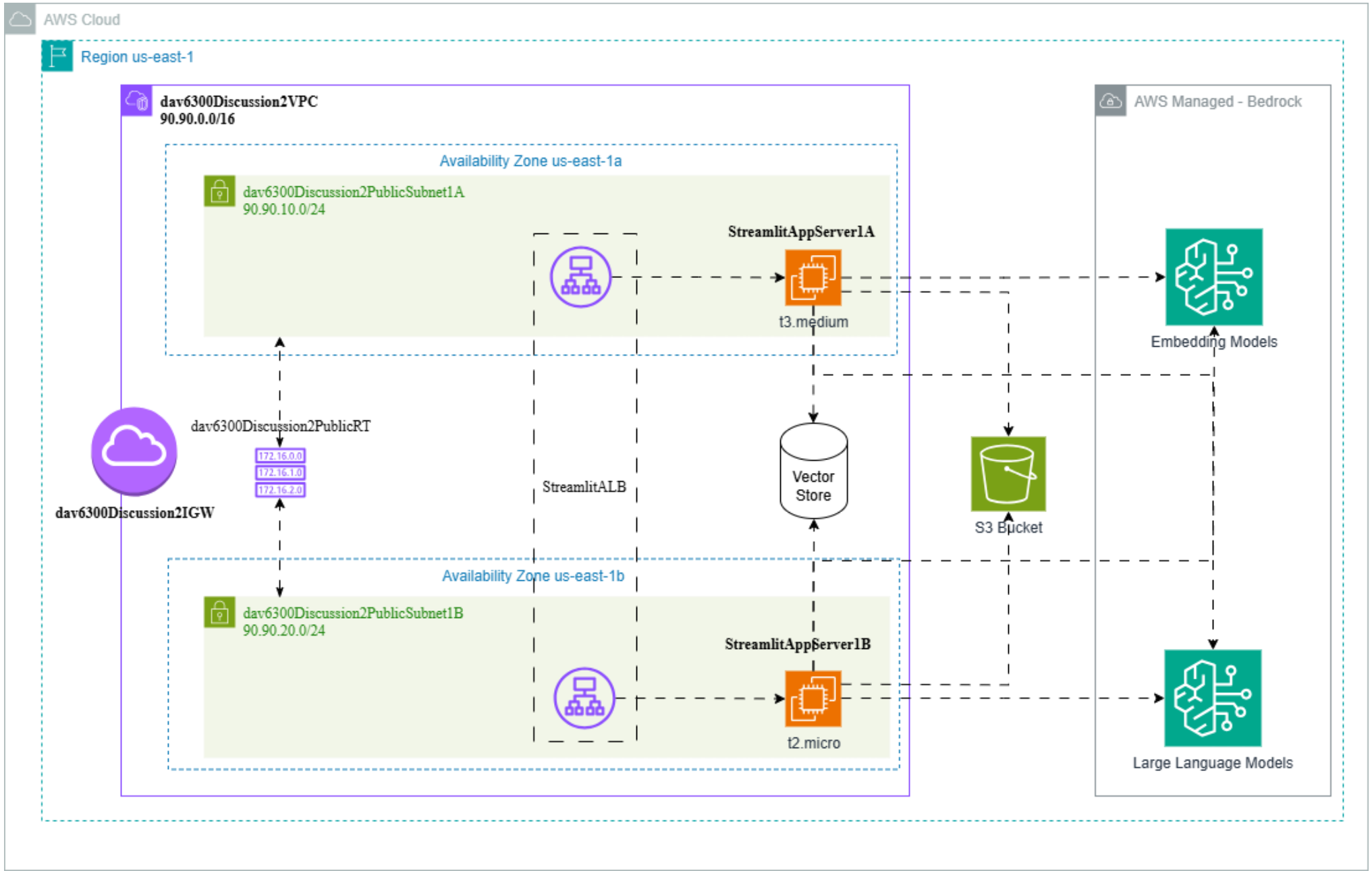
Format Version	Identifies the capabilities of the template
MetaData	Additional information about the template
Description	A description of what this template does
Parameters	Values to pass to your template at runtime
Mappings	A lookup table, maps keys to values so you can change your values
Conditions	Whether resources are created or properties are assigned
Transform	Applies macros
Resources	A resource you want to create
Outputs	Values that are returned

* Courtesy – Sarah Lean [6]

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

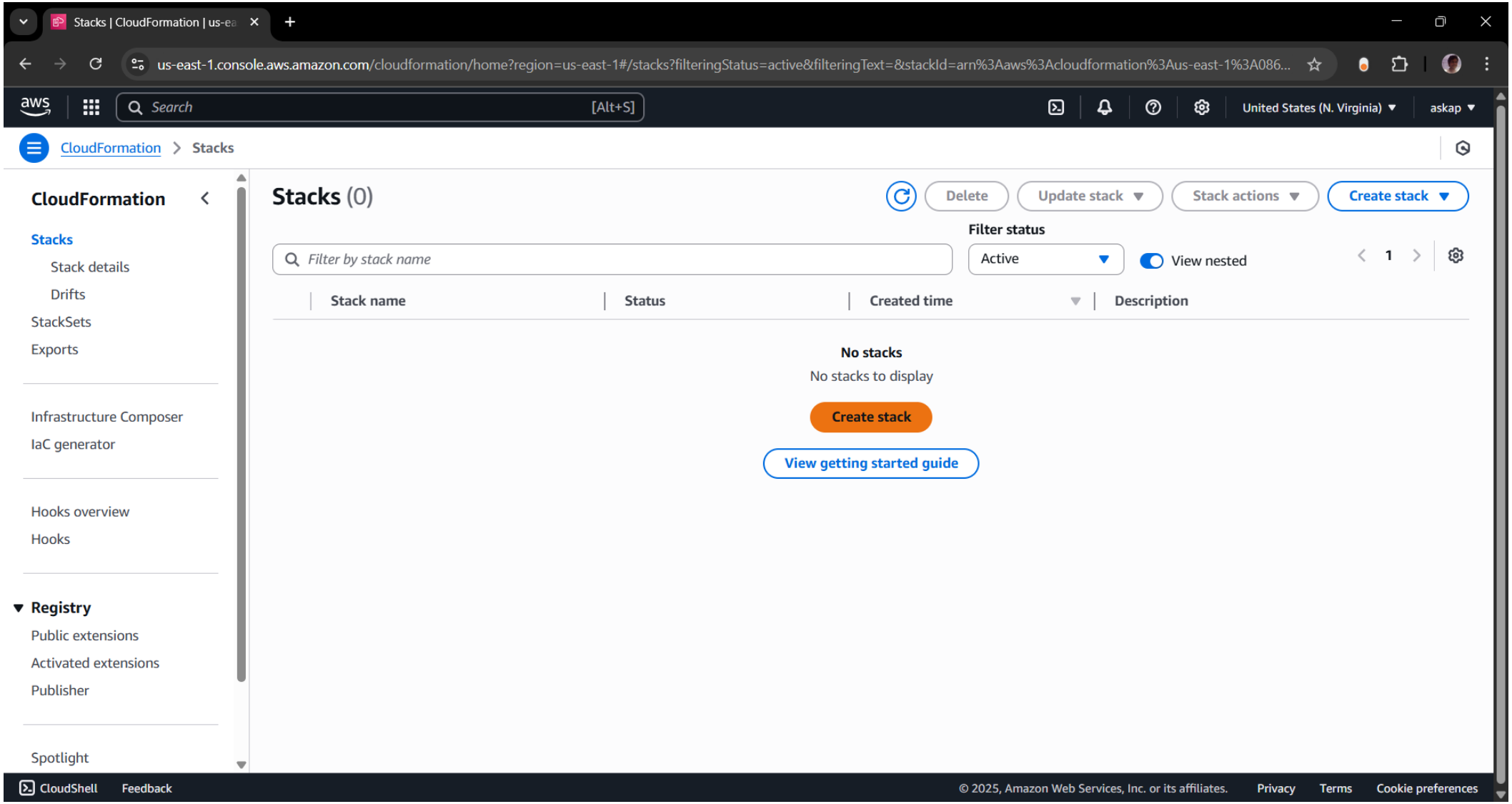
DNS Link of Application Load Balancer – **Streamlit ALB**

Running on Port **8501**

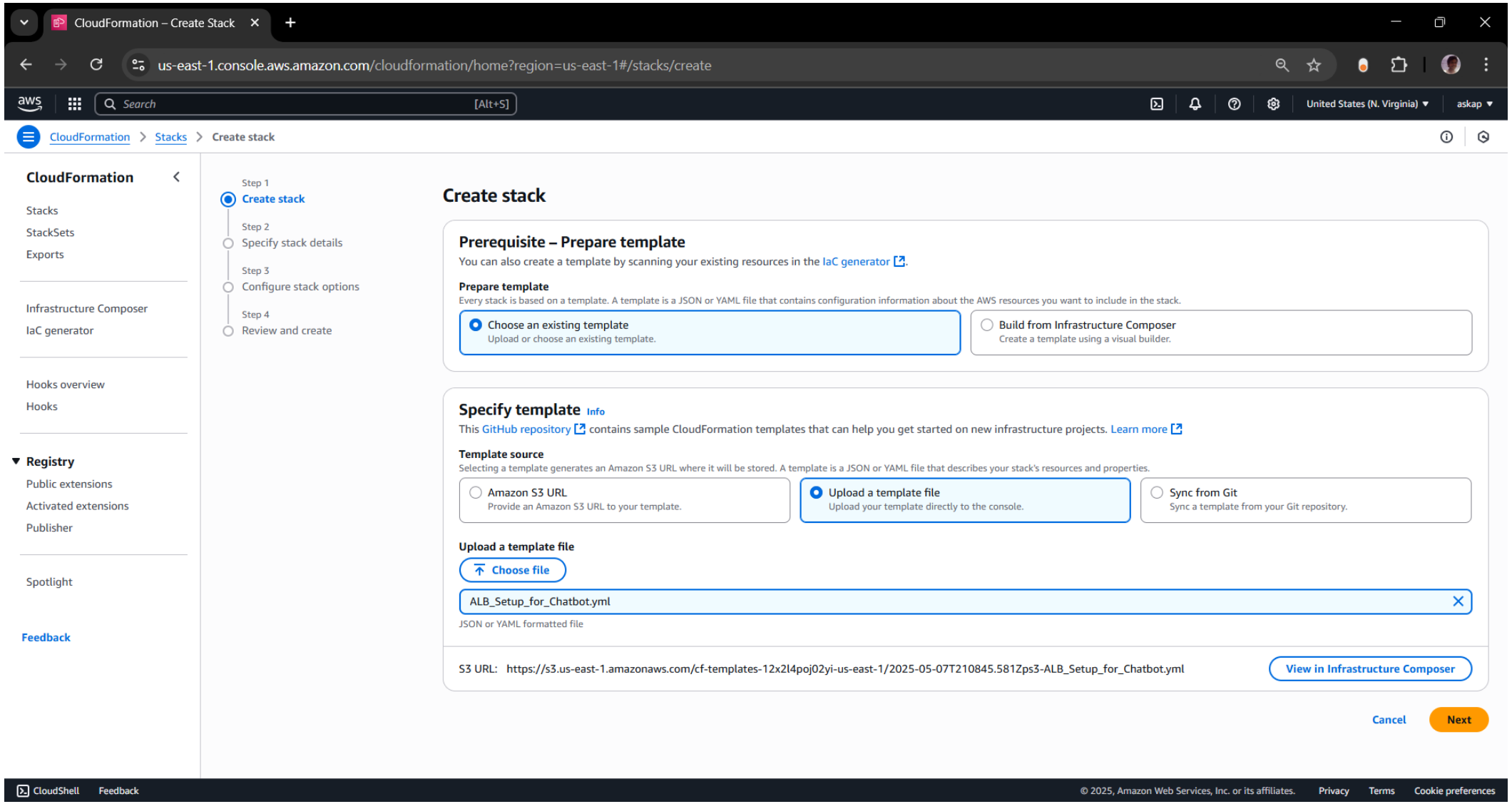
Note: Steamlit application doesn't handle all the exceptions as of now. Be mindful about the error/exception. If you encounter one, kindly report to us.

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

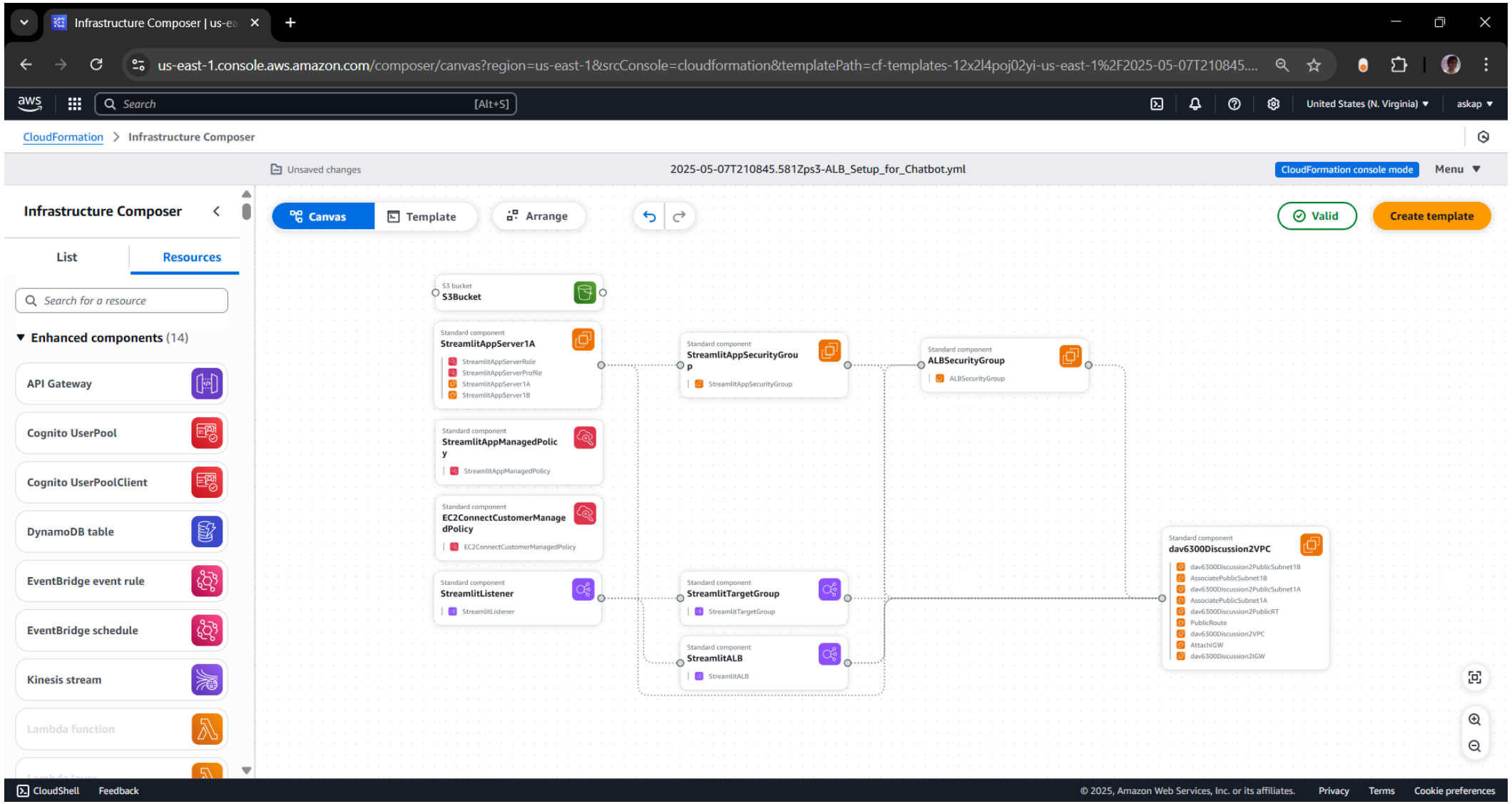
Setup Steps- 1



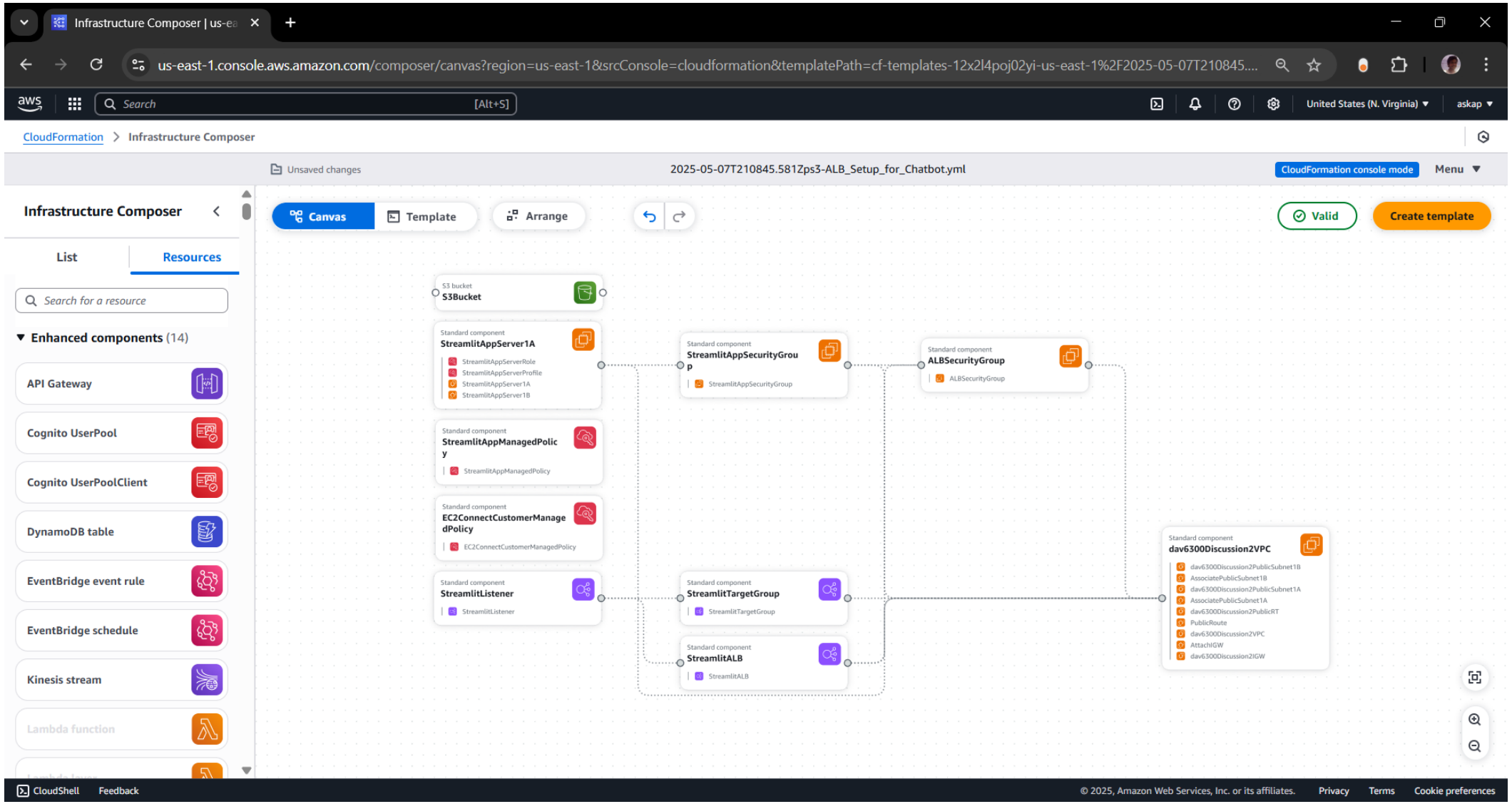
Setup Steps- 2



Setup Steps- 2



Setup Steps– 3



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 > Create stack

CloudFormation

- Stacks
- StackSets
- Exports
- Infrastructure Composer
- IaC 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

LatestAmiId
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

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 > Create stack

CloudFormation

- Stacks
- StackSets
- Exports

- Infrastructure Composer
- IaC 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

Configure stack 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.

IAM role name | *Sample-role-name* | Remove

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.

CloudShell | Feedback

 Privacy | Terms | Cookie preferences

Setup Steps– 5.2

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 > Create stack

CloudFormation

- Stacks
- StackSets
- Exports

- Infrastructure Composer
- IaC generator

- Hooks overview
- Hooks

- **Registry**
  - Public extensions
  - Activated extensions
  - Publisher

- Spotlight

Feedback

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.**

Cancel | Previous | Next

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

Setup Steps– 6.1

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 > Create stack

CloudFormation

- Stacks
- StackSets
- Exports
- Infrastructure Composer
- IaC 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

Review and create

Step 1: Specify template

Edit

Prerequisite – Prepare template

Template
Template is ready

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 | 1

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

Step 3: Configure stack options

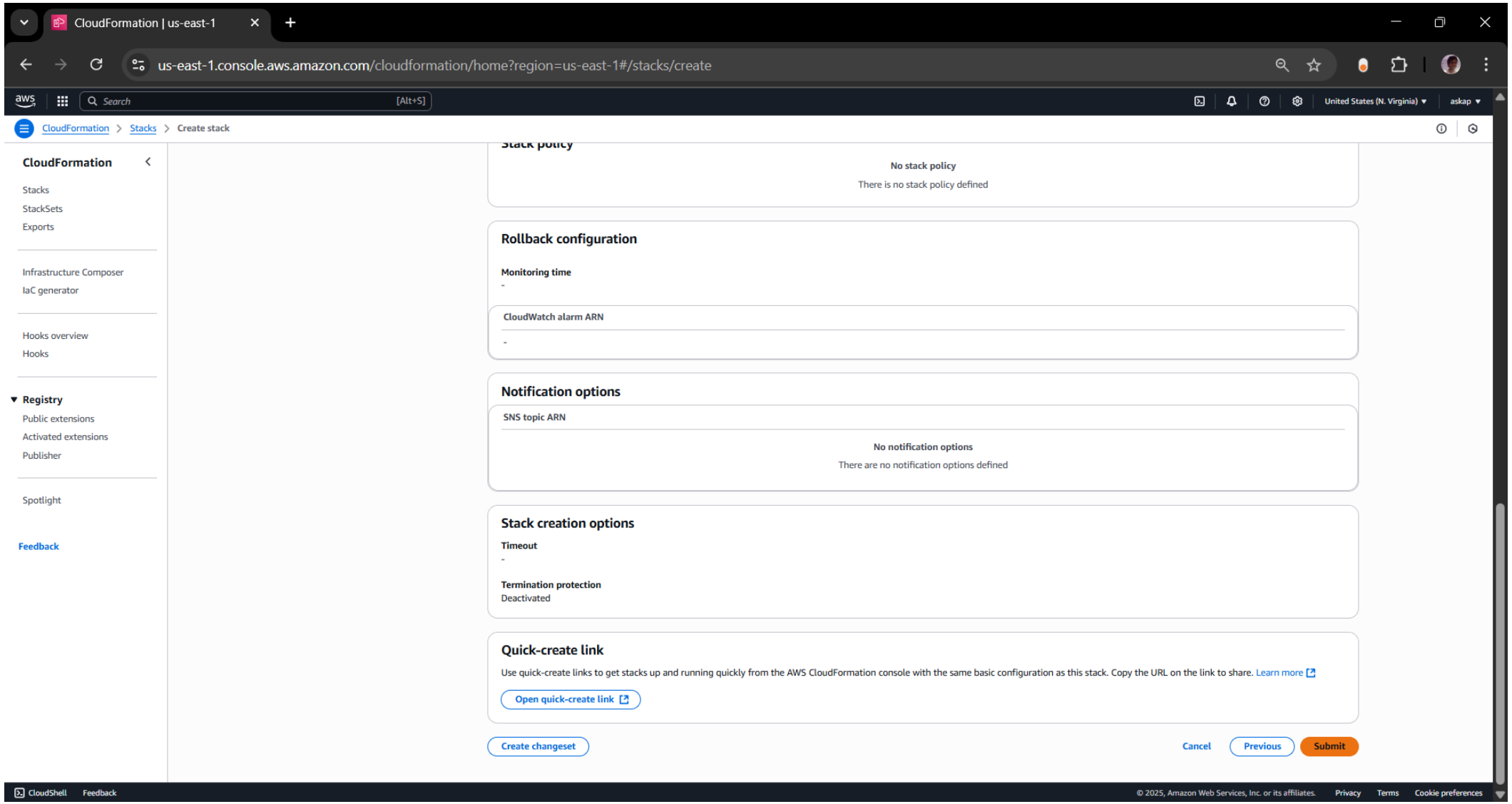
Edit

Tags

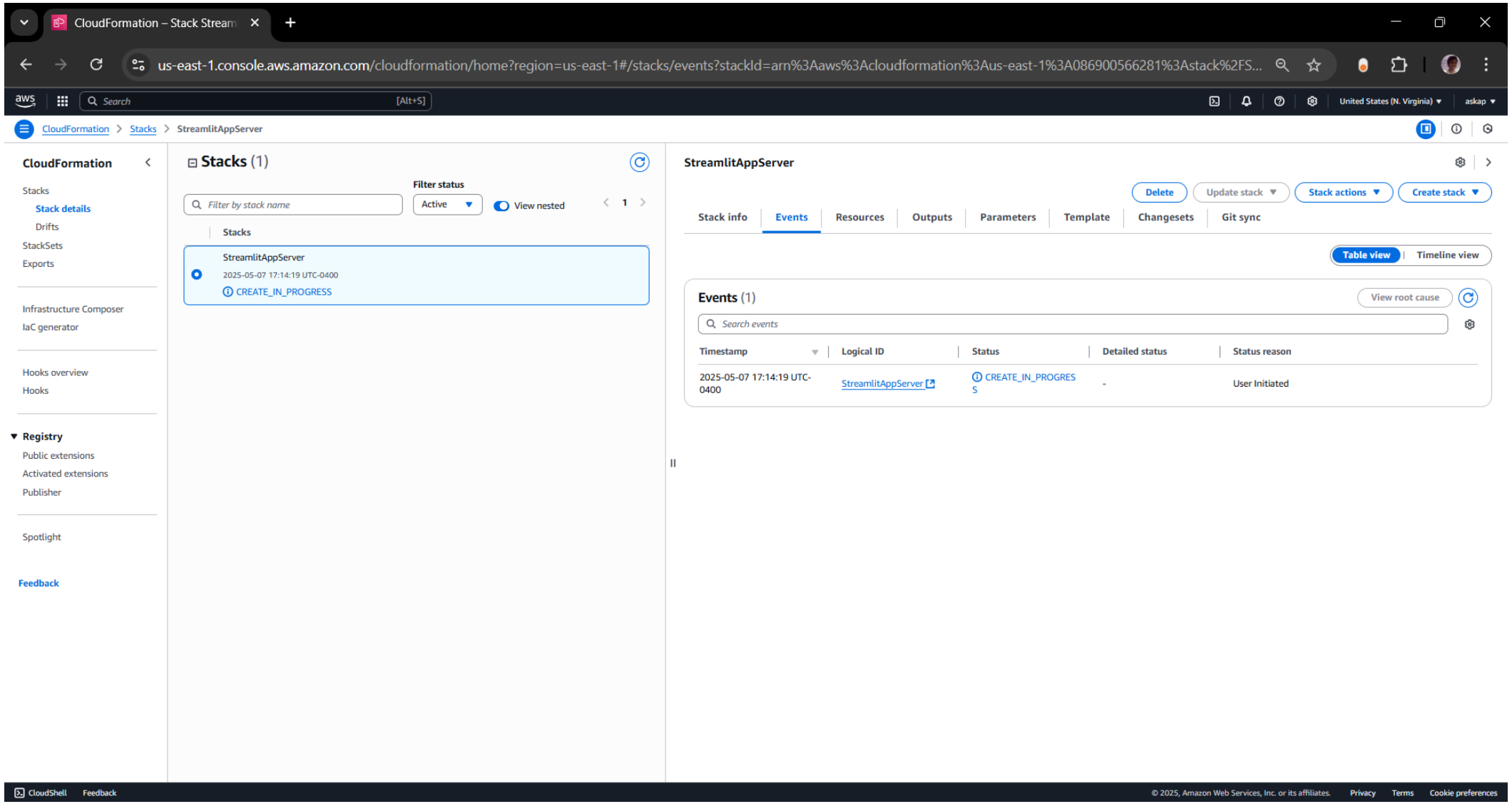
CloudShell Feedback

 Privacy Terms Cookie preferences

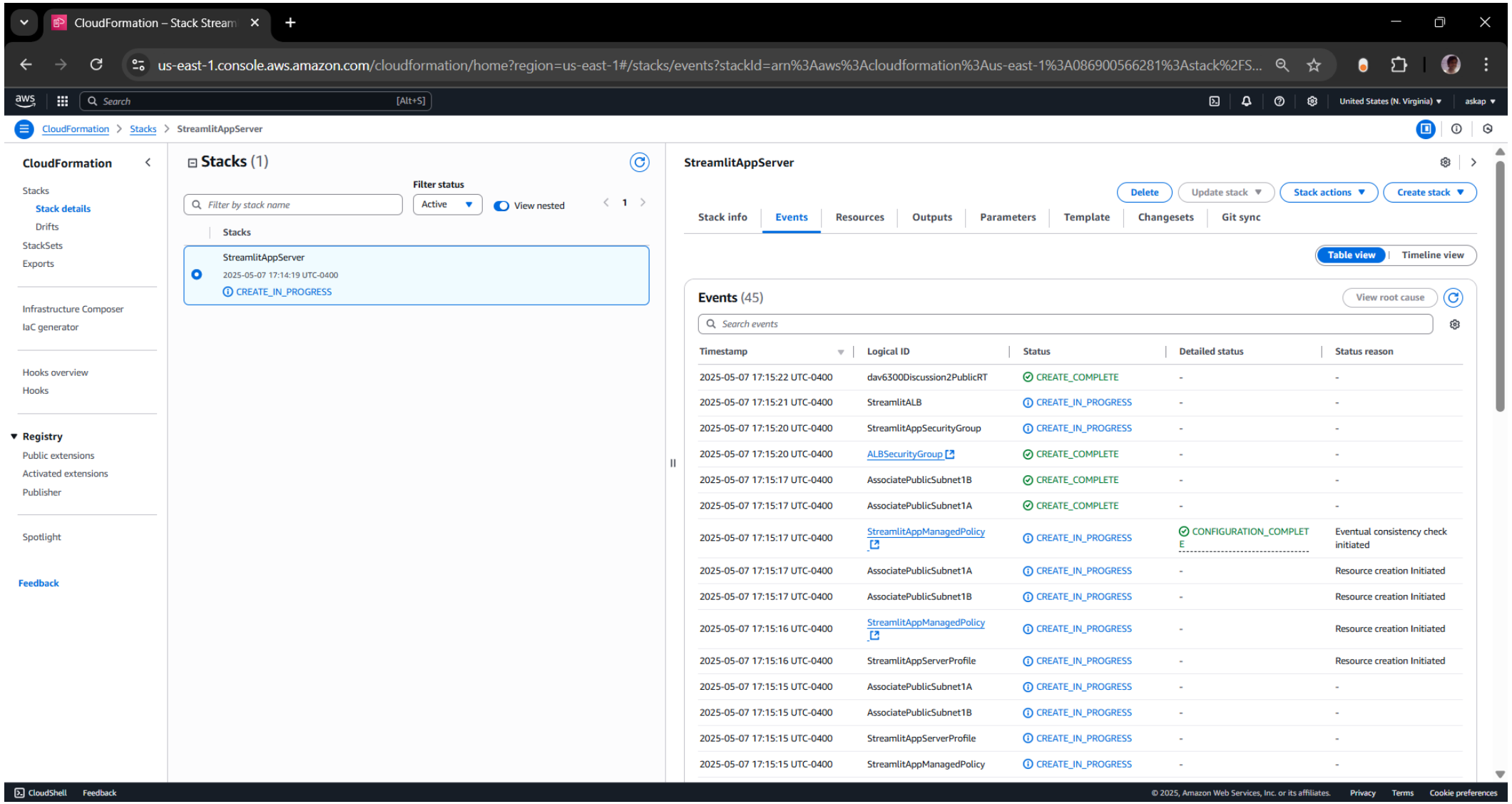
Setup Steps– 6.2



Setup Steps– 7



Setup Steps– 8



Setup Steps– 9

CloudFormation – Stack Stream

us-east-1.console.aws.amazon.com/cloudformation/home?region=us-east-1#/stacks/resources?stackId=arn%3Aaws%3Acloudformation%3Aus-east-1%3A086900566281%3Astack%...

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

CloudFormation > Stacks > StreamlitAppServer

CloudFormation

- Stacks
  - Stack details
  - Drifts
- StackSets
- Exports

- Infrastructure Composer
- IaC generator

- Hooks overview
- Hooks

- **Registry**
  - Public extensions
  - Activated extensions
  - Publisher

- Spotlight

Feedback

Stacks (1)

Filter by stack name | Filter status: Active | View nested

Stacks

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

StreamlitAppServer

Delete | Update stack | Stack actions | Create stack

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

Resources (17)

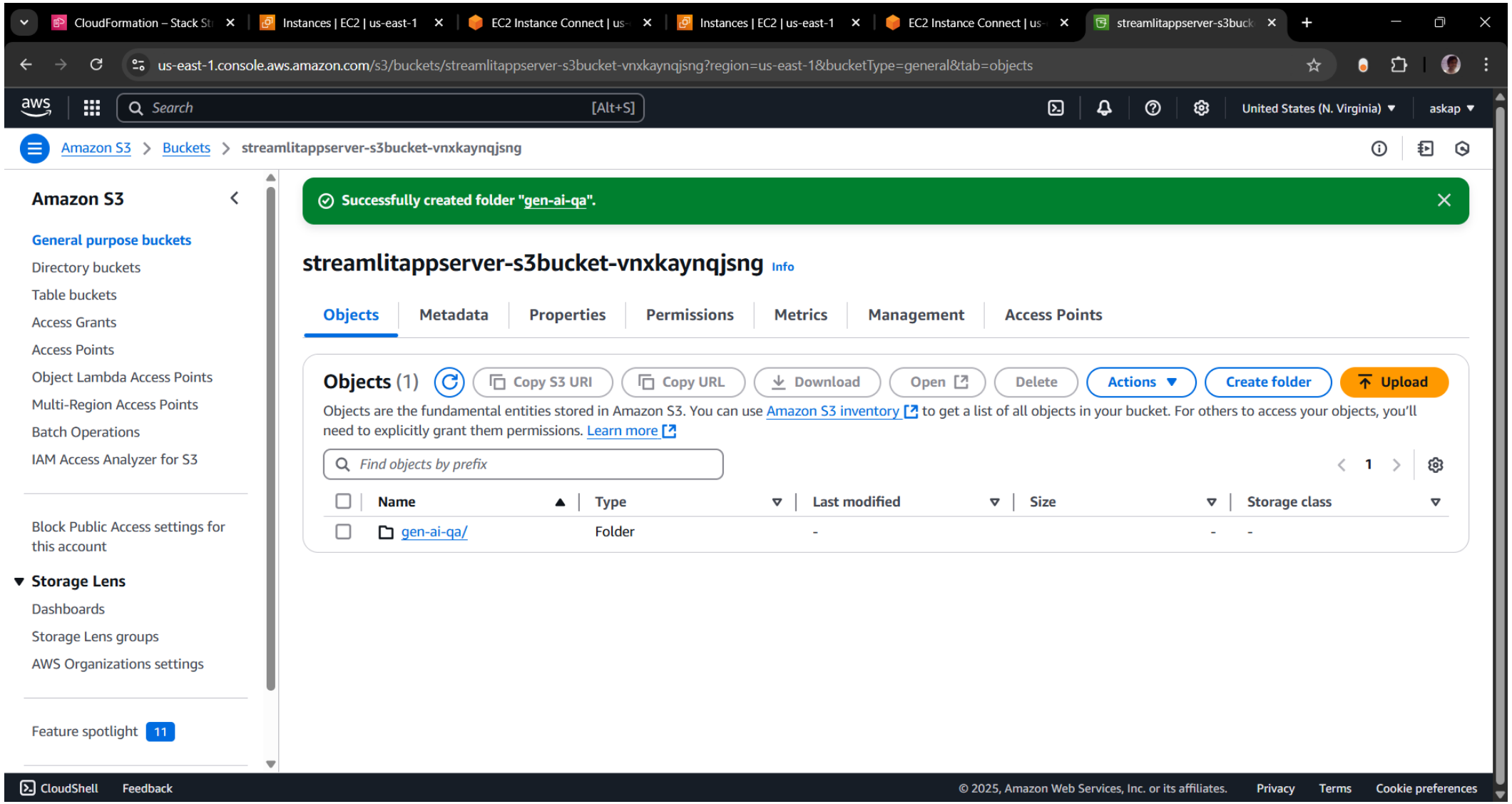
Search resources

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|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	-
dav6300Discussion2PublicSubnet1B	subnet-03b3ccb1a50f96f8b	AWS::EC2::Subnet	CREATE_COMPLETE	-
dav6300Discussion2VPC	vpc-0a2114b1fcd1774ff	AWS::EC2::VPC	CREATE_COMPLETE	-
PublicRoute	rtb-051b45fcc5696e20a|0.0.0.0/0	AWS::EC2::Route	CREATE_COMPLETE	-
S3Bucket	streamlitappserver-s3bucket-rphbf3mvj42b	AWS::S3::Bucket	CREATE_COMPLETE	-
StreamlitALB	arn:aws:elasticloadbalancing:us-east-1:086900566281:loadbalancer/app/Stream-Strea-SiYyJC8EstMt/4372d23fd181c816	AWS::ElasticLoadBalancingV2::LoadBalancer	CREATE_IN_PROGRESS	-

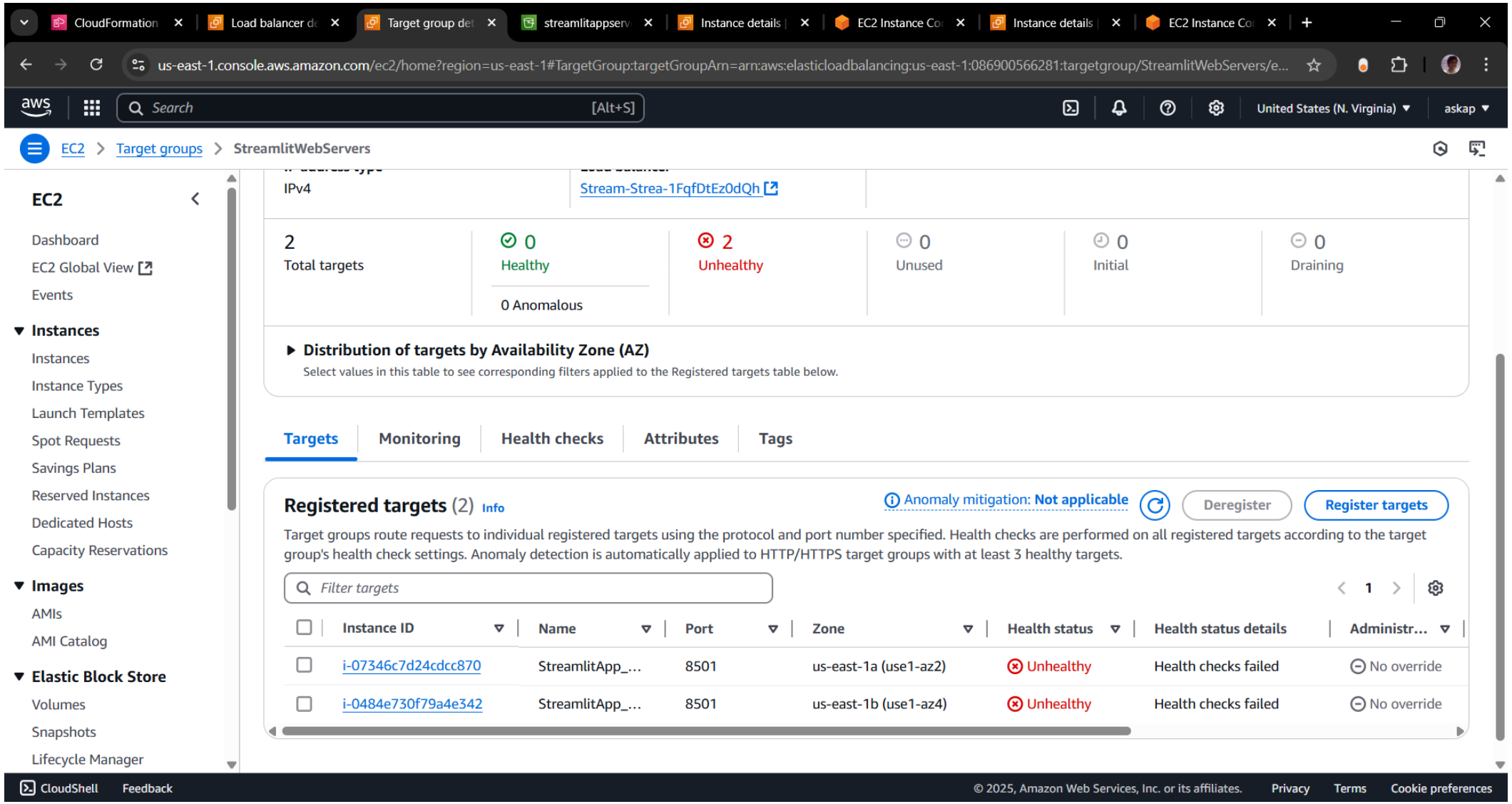
CloudShell | Feedback

 Privacy | Terms | Cookie preferences

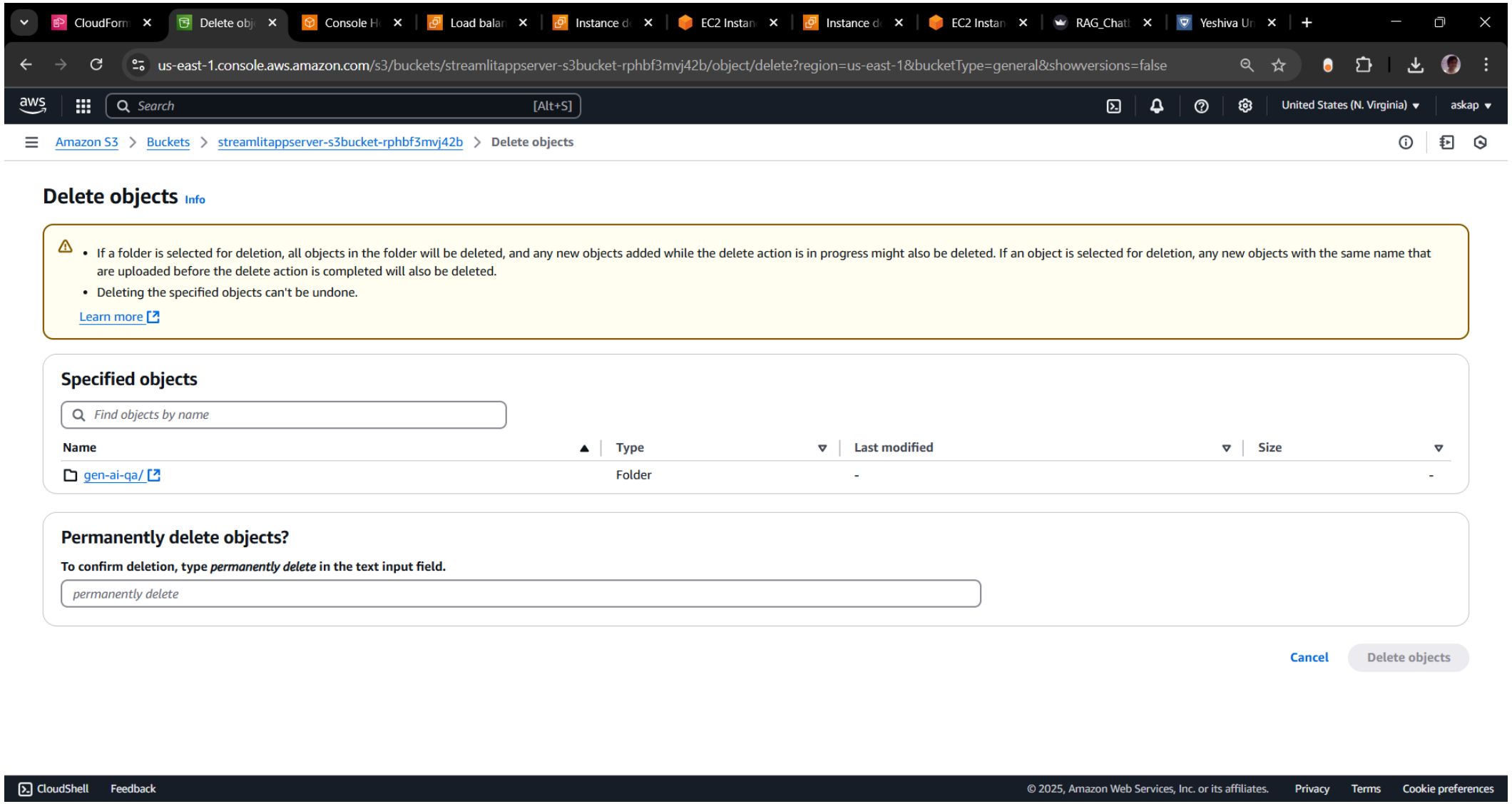
Setup Steps– 10



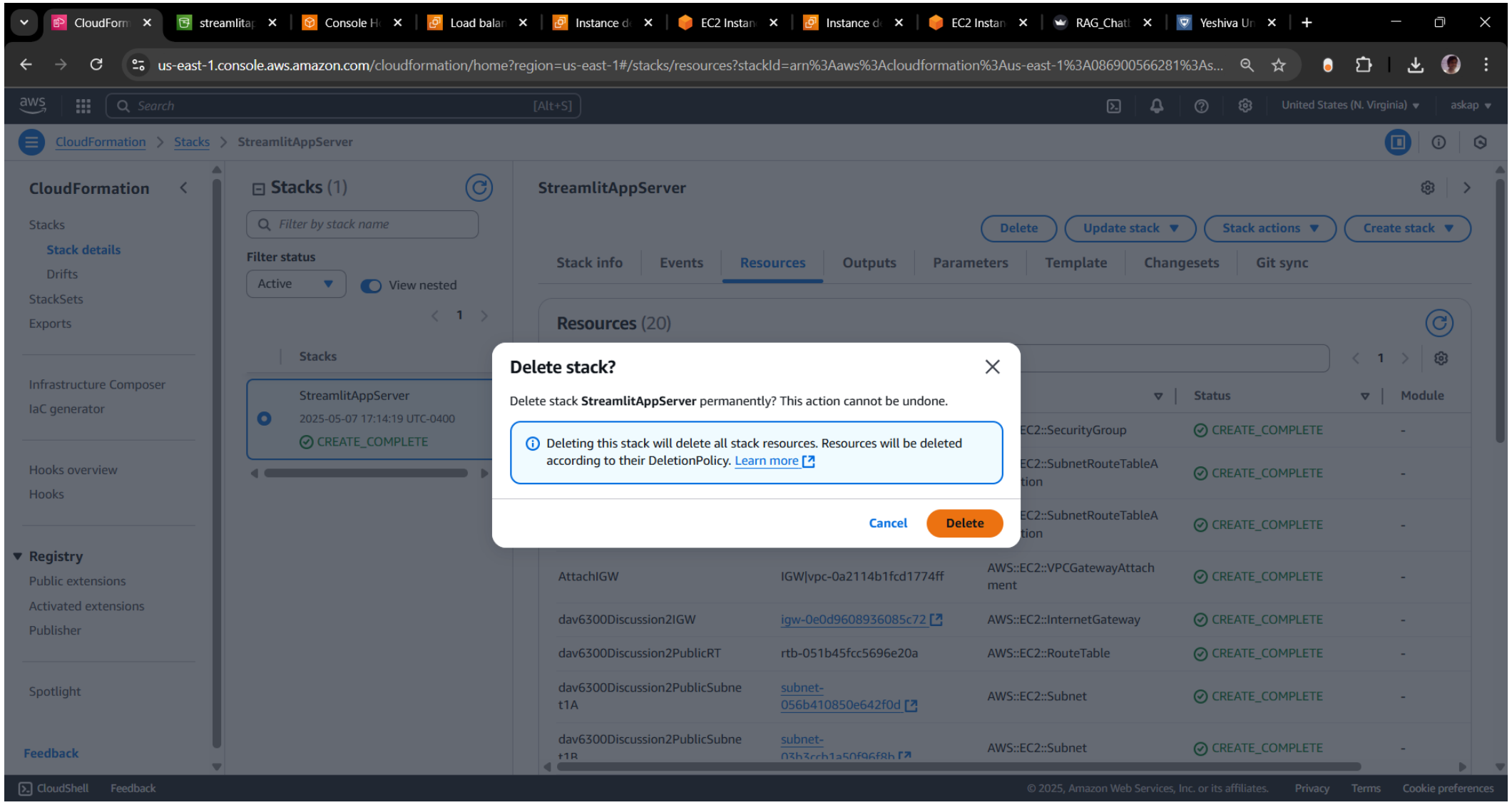
Setup Steps– 11



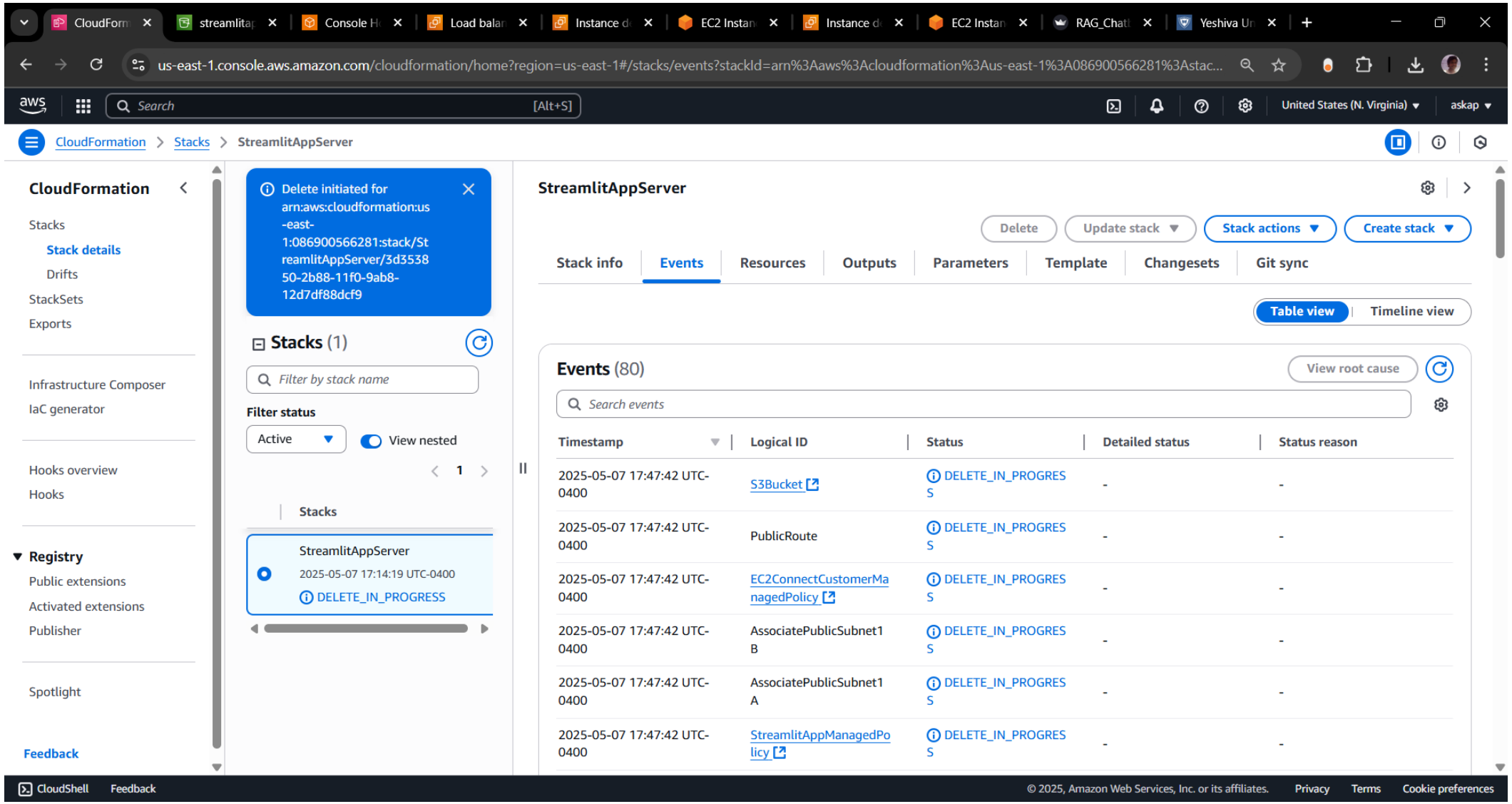
Termination Steps – 1



Termination Steps – 2



Termination Steps – 3

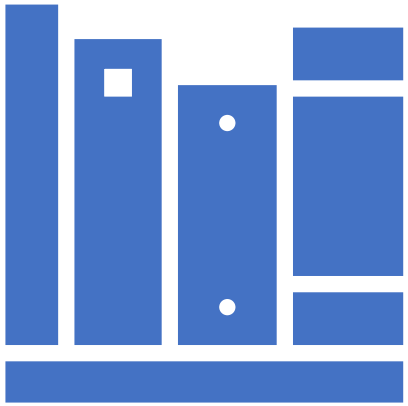


Areas of Improvement

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



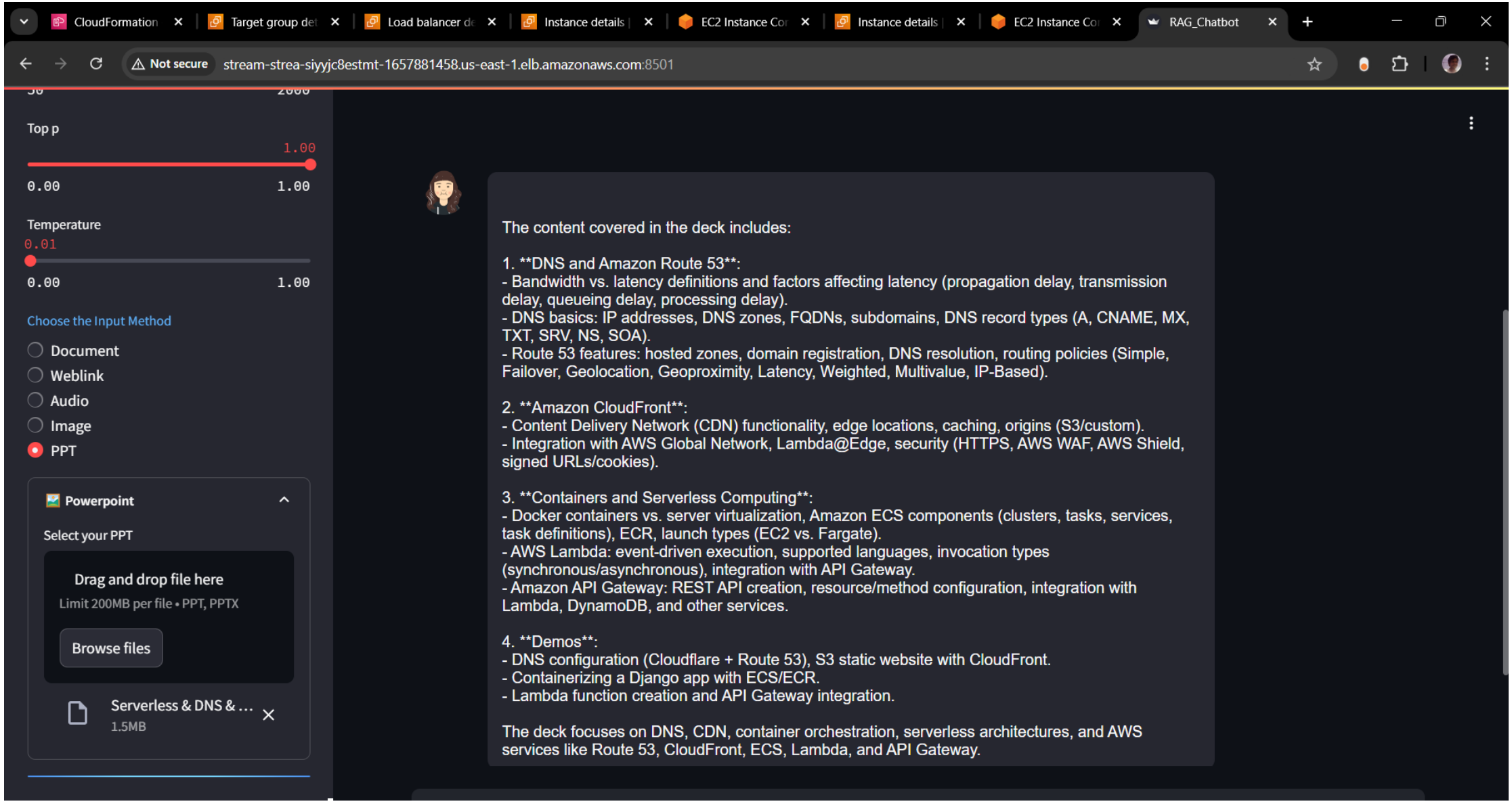

Thank You!

Any Questions

Appendix

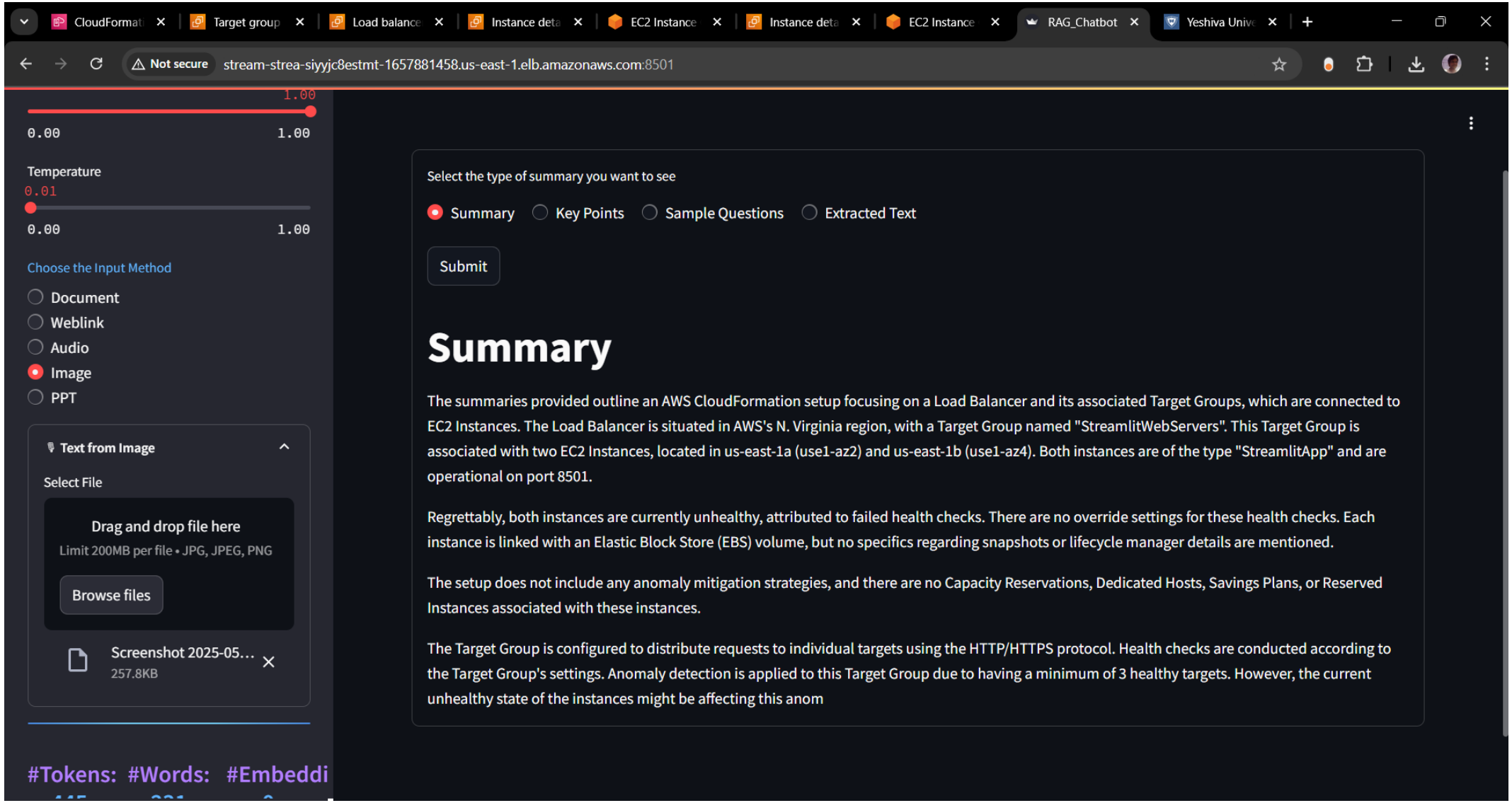
Some Output Snippets–1

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



Some Output Snippets–2

Image of AWS Architecture was uploaded and asked to summarize



Some Output Snippets–2

Key Points from an uploaded PDF document was summarized