

# CAPSTONE PROJECT – AWS

## TASKFLOW APP

### INTRODUCTION

This document presents a comprehensive overview of the three-tier application, design and implement in a scalable, resilient, high available and automated infrastructure on AWS.

1. Infrastructure as Code (IaC): CloudFormation & Terraform
2. CI/CD Automation: Deploying the Application
3. Application Demo: The Task Flow App in Action
4. High Availability: Route 53 DNS Failover
5. Monitoring: CloudWatch for EKS
6. Automated Notifications: Lambda & Event Bridge

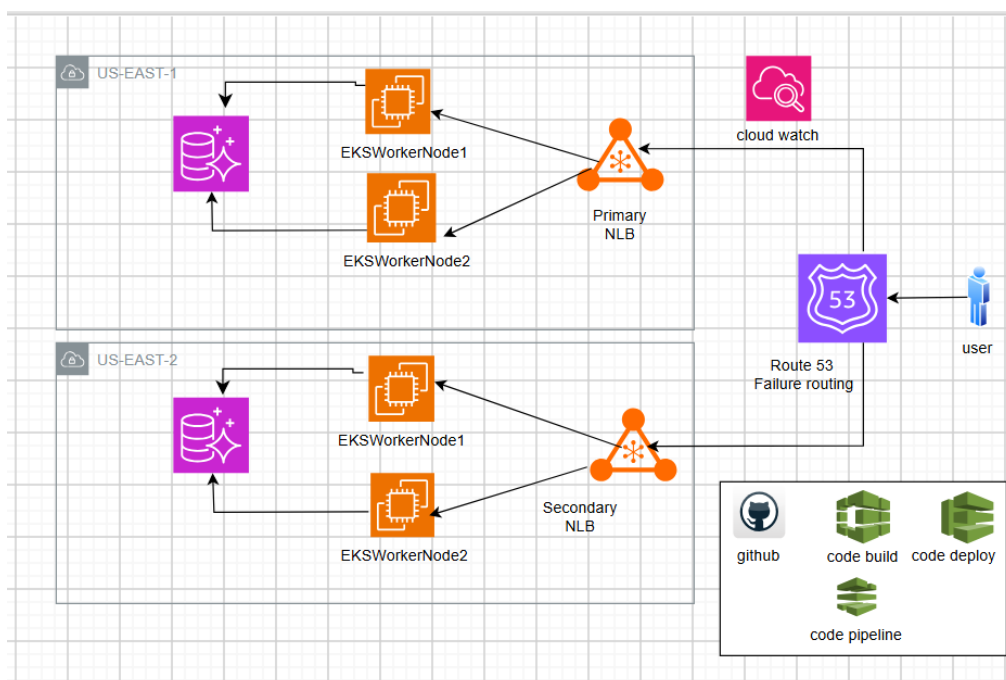
### **Three tier Architecture Overview :-**

The application is deployed on AWS, leveraging Amazon Elastic Kubernetes Service (EKS) for container orchestration, Amazon RDS for the relational database, and various networking and IAM components to ensure secure and scalable operations.

- Frontend: React application
- Backend: Node.js/Express API
- Database: AWS RDS PostgreSQL

### **ARCHITECTURE**

#### **High level architecture of infrastructure with route53 failover routing**



The diagram illustrates the relationships between various AWS Standard Components for EKS. On the left, components like **EKSNodeGroup**, **EKSNodeFromClusterIngress**, **EKSNodeSelfIngress**, and **DBInstance** are shown. These connect to **EKSCluster** and **EKSNodeSecurityGroup**. **EKSCluster** connects to **EKSNodeSecurityGroup** and **EKSClusterSecurityGroup**. **EKSNodeSecurityGroup** connects to **EKSClusterSecurityGroup**. **EKSClusterSecurityGroup** connects to the **VPC** component. The **VPC** component lists various subnets, routes, and gateways.

```

graph LR
    subgraph LeftColumn [ ]
        direction TB
        EKSNodeGroup[EKSNodeGroup]
        EKSNodeFromClusterIngress[EKSNodeFromClusterIngress]
        EKSNodeSelfIngress[EKSNodeSelfIngress]
        DBInstance[DBInstance]
    end

    subgraph MiddleColumn [ ]
        direction TB
        EKSCluster[EKSCluster]
        EKSNodeSecurityGroup[EKSNodeSecurityGroup]
    end

    subgraph RightColumn [ ]
        direction TB
        EKSClusterSecurityGroup[EKSClusterSecurityGroup]
        VPC[VPC]
    end

    EKSNodeGroup -.-> EKSCluster
    EKSNodeFromClusterIngress -.-> EKSCluster
    EKSNodeSelfIngress -.-> EKSCluster
    DBInstance -.-> EKSCluster
    EKSCluster -.-> EKSNodeSecurityGroup
    EKSCluster -.-> EKSClusterSecurityGroup
    EKSNodeSecurityGroup -.-> EKSClusterSecurityGroup
    EKSClusterSecurityGroup -.-> VPC
  
```

**Standard Component EKSNodeGroup**

- EKSNodeGroupRole
- EKSNodeGroup

**Standard Component EKSCluster**

- EKSClusterRole
- EKSCluster

**Standard Component EKSNodeFromClusterIngress**

- EKSNodeFromClusterIngress

**Standard Component EKSNodeFromClusterIngress**

- EKSNodeFromClusterIngress

**Standard Component EKSNodeSelfIngress**

- EKSNodeSelfIngress

**Standard Component DBInstance**

- DBSecurityGroup
- DBInstance
- DBSubnetGroup
- RDSEnhancedMonitoringRole

**Standard Component EKSNodeSecurityGroup**

- EKSNodeSecurityGroup

**Standard Component EKSClusterSecurityGroup**

- EKSClusterSecurityGroup

**Standard Component VPC**

- PrivateSubnet3
- PrivateSubnet3RTAssoc
- PrivateSubnet2
- PrivateSubnet2RTAssoc
- PrivateSubnet1
- PrivateSubnet1RTAssoc
- PrivateRouteTable
- PrivateRoute
- NatEIP
- NatGateway
- PublicSubnet3
- PublicSubnet1RTAssoc
- PublicSubnet2
- PublicSubnet2RTAssoc
- PublicSubnet1
- PublicSubnet1RTAssoc
- PublicRouteTable
- PublicRoute
- VPC
- AttachInternetGateway
- InternetGateway

<https://github.com/SelmiNazeeb/FinalProject-Devops.git>

## CREATING INFRASTRUCTURE THROUGH CLOUDFORMATION IN REGION1 (US-EAST-1)

- Cloud formation role – full admin access
- Pipeline role - [AmazonEKSClusterPolicy](#), [AWSCloudFormationFullAccess](#)

Developer Tools > [CodePipeline](#) > [Pipelines](#) > cftFinalPipe

## cftFinalPipe

Edit
Stop execution
Create trigger
Clone pipeline
Release change

Pipeline
Executions
Triggers
Settings
Tags
Stage

```

graph LR
    subgraph Pipeline [cftFinalPipe]
        direction LR
        subgraph Stage1 [ ]
            direction TB
            S1[Source]
        end
        subgraph Stage2 [ ]
            direction TB
            S2[Deploy]
        end
        Stage1 --> Stage2
    end
    S1 --> S2
  
```

**Source**  
a5fa5953-0f41-4c60-a2ff-5e290860050a  
0 of 1 action didn't run.

Source

6e334576 [Link](#) Source: cft2

**Deploy**  
a5fa5953-0f41-4c60-a2ff-5e290860050a  
All actions succeeded.

Deploy

[AWS CloudFormation](#)  
Jun 21, 2025 11:00 PM (UTC+5:30)

6e334576 [Link](#) Source: cft2

## Cloud formation stack created through pipeline

finalstack

Delete

Update stack

Stack actions

Create stack

Stack info

Events

Resources

Outputs

Parameters

Templat

Overview

Stack ID

arn:aws:cloudformation:us-east-1:390403857216:stack/finalstack/4319f810-4ec3-11f0-8462-12e25e1641f5

Description

Three-tier infra: VPC + EKS + RDS PostgreSQL with public/private subnets, IGW, NAT Gateway, route tables, security groups, IAM roles, etc.

Status

CREATE\_COMPLETE

Detailed status

-

Status reason

-

Root stack

-

Parent stack

-

Created time

2025-06-21 22:48:17 UTC+0530

## Output in cloud formation stack

finalstack

Delete

Update stack

Stack actions

Create stack

Stack info

Events

Resources

Outputs

Parameters

Template

Change sets

Git sync

Outputs (3)

Search outputs

< 1 >

Key	Value	Description	Export name
ClusterEndpoint	<a href="https://07EC244C5F63EFE19C7C8D99187E74F3.gr7.us-east-1.eks.amazonaws.com">https://07EC244C5F63EFE19C7C8D99187E74F3.gr7.us-east-1.eks.amazonaws.com</a>	-	finalstack-ClusterEndpoint
DBEndpoint	<a href="task-db-postgres.czs2i4meqfm1.us-east-1.rds.amazonaws.com">task-db-postgres.czs2i4meqfm1.us-east-1.rds.amazonaws.com</a>	-	finalstack-DBEndpoint
DBPort	5432	-	finalstack-DBPort

## Postgresql Database

Databases (1)

Group resources

Modify

Actions

Create database

Filter by databases

< 1 >

DB identifier	Status	Role	Engine	Region ...	Size
<a href="#">task-db-postgres</a>	Available	Instance	PostgreSQL	us-east-1c	db.t3.micro

## ECR

Amazon ECR

Private registry

Repositories

Amazon Elastic Container Registry

Private registry

Public registry

Private repositories (5)

View push commands

Delete

Actions

Create repository

new

Repository name	URI	Created at	Tag immutability	Encryption type
<a href="#">backendnew</a>	<a href="https://390403857216.dkr.ecr.us-east-1.amazonaws.com/backendnew">390403857216.dkr.ecr.us-east-1.amazonaws.com/backendnew</a>	June 20, 2025, 16:13:08 (UTC+05.5)	Mutable	AES-256
<a href="#">frontendnew</a>	<a href="https://390403857216.dkr.ecr.us-east-1.amazonaws.com/frontendnew">390403857216.dkr.ecr.us-east-1.amazonaws.com/frontendnew</a>	June 20, 2025, 16:13:00 (UTC+05.5)	Mutable	AES-256

## EKS cluster

**three-tier-eks-cluster** [Delete cluster] [Upgrade version] [Monitor cluster]

⚠ Your cluster's Kubernetes version (1.28) will reach the end of extended support on November 26, 2025. If you don't upgrade your cluster to a later version before that date, it will be automatically upgraded to Kubernetes version 1.29. [Upgrade now]

**Cluster info** [Info](#)

<b>Status</b> Active	<b>Kubernetes version</b> 1.28	<b>Support period</b> Extended support until November 26, 2025	<b>Provider</b> EKS
<b>Cluster health</b> 0	<b>Upgrade insights</b> 6	<b>Node health issues</b> 0	

Overview | Resources | Compute | Networking | Add-ons | **Access** | Observability | Update history | Tags

**Access configuration** [Manage access](#)

Authentication mode  
EKS API and ConfigMap

## Permission to user to access EKS cluster

Select EKS cluster -> Access -> Add access entry -> give user arn -> next -> add “AmazonEKSClusterAdminPolicy” -> save

arn:aws:iam::390403857216:user/terraform-user      Standard      arn:aws:iam::390403857216:user/terraform-user      -      AmazonEKSClusterAdminPolicy

## TASK 2:

## DEPLOYING APPLICATION ON THE INFRASTRUCTURE CREATED BY CLOUD FORMATION

### Permissions needed

- pipeline role – Access entry in EKS cluster

Select EKS cluster -> Access -> Add access entry -> give pipeline role arn -> next -> add “AmazonEKSAdminPolicy” -> save

**IAM access entries (4)** [Info](#) [View details] [Edit] [Delete] [Create access entry]

Find access entry by property filtering

IAM principal ARN	Type	Username	Group names	Access policies
arn:aws:iam::390403857216:role/service-role/AWSCodePipelineServiceRole-us-east-1-buildpipelineeksRDS	Standard	arn:aws:sts::390403857216:assumed-role/AWSCodePipelineServiceRole-us-east-1-buildpipelineeksRDS/{{SessionName}}	-	AmazonEKSAdminPolicy

- Code build role - [AmazonEC2ContainerRegistryPowerUser](#)

## Success pipeline of application deployment

**buildpipelineeksRDS** [Edit] [Stop execution] [Create trigger] [Clone pipeline] [Release change]

Pipeline | Executions | Triggers | Settings | Tags | Stage

**Source** 961c718a-5e15-4243-96b7-aada45477ab9  
0 of 1 action didn't run.  
Source  
abf300d5 Source: config

**Build** 961c718a-5e15-4243-96b7-aada45477ab9  
All actions succeeded.  
Build  
AWS CodeBuild  
Jun 22, 2025, 1:15 PM (UTC+3:30)  
abf300d5 Source: config

**Deploy** 291e5175-9cc8-4b95-8fc7-88417e5f7be4  
All actions succeeded.  
Deploy  
Amazon EKS  
Jun 22, 2025, 1:27 PM (UTC+3:30)  
abf300d5 Source: config

>>> Now all containers, pods, deployment and services are running

## Load Balancer created for the application

Load balancers (2)								
Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.								
<input type="text" value="Filter load balancers"/>								
<div>&lt; 1 &gt; ⚙</div>								
<input type="checkbox"/>	Name	DNS name	State	VPC ID	Availability Zones	Type	Date created	
<input type="checkbox"/>	acac655f892824cb58fe...	acac655f892824cb58fe835...	Active	vpc-0a15dad6dfa654dd5	3 Availability Zones	network	June 22, 2025,	

In Browser – paste load balancer arn **(Application is working)**

**Frontend of Task management APP – Adding a new task** (task1 – give title and description -> add task)

TaskFlow Dashboard

Organize Your Workflow

Add New Task

Task deleted successfully!

Task Title

task1

Task Description

cloud formation success

+ ADD TASK

Your Tasks (1)

new

yes

## Added task and stored in Database

TaskFlow Dashboard

Organize Your Workflow

Add New Task

Task added successfully!

Task Title

Task Description

+ ADD TASK

Your Tasks (2)

task1

cloud formation success

new

yes

## TASK3 :

# CREATING INFRASTRUCTURE THROUGH TERRAFORM IN REGION2 (US-EAST-2)

## Permissions needed

- Code build role - policy should be attached (added in github -> permissions/terraform-build-policy)  
[AmazonS3FullAccess](#), [AmazonVPCFullAccess](#)

## Infrastructure creation : success

The screenshot shows the AWS CodeBuild console for a project named 'terraform-build'. The configuration section indicates the source provider is GitHub, the primary repository is 'SelmiNazeem/FinalProject-Devops', and the service role is 'arn:aws:iam::390403857216:role/service-role/codebuild-terraform-build-service-role'. The build history section shows a single successful build run with the following details:

Build run	Status	Build number	Source version	Submitter	Duration	Completed
terraform-build:e843e6de-8558-4b41-80b0-595f8f7acb80	Succeeded	43	c5c9cbe868531e a361b2b2a840f6 2df68996e6c1	GitHub- Hookshot/fb6e1 5e	38 seconds	12 hours ago

## Dynamo DB table and S3 bucket for storing terraform tf state storage of code build of infrastructure

## Dynamo DB

The screenshot shows the AWS DynamoDB console for a table named 'terraform-lock-table'. The 'Scan or query items' section shows a successful scan with the following results:

LockID (String)	Digest
selmicapstones3/terr...	b0dba6e44719e3886ff8fb1dd33e67c

## S3 bucket

**selmicapstones3** 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](#)

☒ Show versions < 1 > [Settings](#)

<input type="checkbox"/>	Name	Type	Last modified	Size
<input type="checkbox"/>	<a href="#">terraform.tfstate</a>	tfstate	June 23, 2025, 11:55:56 (UTC+05:30)	134.8 KB

## Outputs

Outputs:

```
eks_cluster_endpoint = "https://7627E37233A3CF883ECD615F9D063276.gr7.us-east-2.eks.amazonaws.com"
eks_cluster_security_group_id = "sg-020b2b5f3047cfc66"
rds_endpoint = "task-db-postgres.czus4mukos81.us-east-2.rds.amazonaws.com:5432"
vpc_id = "vpc-02b566a9f4f50daec"
```

## TASK 4:

## DEPLOYING APPLICATION ON THE INFRASTRUCTURE CREATED BY TERRAFORM

### Permissions needed

- pipeline role – Access entry in EKS cluster

Select EKS cluster -> Access -> Add access entry -> give pipeline role arn -> next -> add “AmazonEKSAdminPolicy” -> save

**IAM access entries (4)** Info [View details](#) [Edit](#) [Delete](#) [Create access entry](#)

< 1 >

IAM principal ARN	Type	Username	Group names	Access policies
<a href="#">arn:aws:iam::390403857216:role/service-role/AWSCodePipelineServiceRole-us-east-1-buildpipelineeksRDS</a>	Standard	<a href="#">arn:aws:sts::390403857216:assumed-role/AWSCodePipelineServiceRole-us-east-1-buildpipelineeksRDS/{{SessionName}}</a>	-	AmazonEKSAdminPolicy

- Code build role - [AmazonEC2ContainerRegistryPowerUser](#)

### Success pipeline of application deployment

Developer Tools > [CodePipeline](#) > [Pipelines](#) > terra-app-deploy-pipeline

**terra-app-deploy-pipeline** [Edit](#) [Stop execution](#) [Create trigger](#) [Clone pipeline](#) [Release change](#)

[Pipeline](#) [Executions](#) [Triggers](#) [Settings](#) [Tags](#) [Stage](#)

✓ ✓ ✓ ⓘ

```
graph LR
    Source[Source] --> Build[Build]
    Build --> Deploy[Deploy]
```

**Source** [ed1e6c3d-8e0a-40a5-98d6-a57e3bdc204](#)  
All actions succeeded.

Source  
✓ [GitHub \(via GitHub App\)](#)  
8 hours ago  
[d2ee709a](#) [Source: add](#)

**Build** [ed1e6c3d-8e0a-40a5-98d6-a57e3bdc204](#)  
All actions succeeded.


Build  
✓ [AWS CodeBuild](#)  
7 hours ago  
[d2ee709a](#) [Source: add](#)

**Deploy** [ed1e6c3d-8e0a-40a5-98d6-a57e3bdc204](#)  
All actions succeeded.

Deploy  
✓ [Amazon EKS](#)  
7 hours ago  
[d2ee709a](#) [Source: add](#)

>>> Now all containers, pods, deployment and services are running

## Load Balancer created for the application

Load balancers (1)								Actions	Create load balancer
Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.									
<input type="text" value="Filter load balancers"/>									
<input type="checkbox"/>	Name	DNS name	State	VPC ID	Availability Zones	Type			
<input type="checkbox"/>	ae42791e1b72140ccb6...	ae42791e1b72140ccb6d58...	Active	vpc-02b566a9f4f50daec	2 Availability Zones	network			

In Browser – paste load balancer arn **(Application is working)**

**Frontend of Task management APP – Adding a new task** (task2 – give title and description -> add task)

TaskFlow Dashboard

Organize Your Workflow

Add New Task

Task deleted successfully!

Task Title

task2



Task Description

terraform deploy success

+ ADD TASK

Your Tasks (1)

task  
new



## Added task and stored in Database

TaskFlow Dashboard

Organize Your Workflow

Add New Task

Task added successfully!



Task Title

Task Description



+ ADD TASK

Your Tasks (2)

task2  
terraform deploy success



hello  
added route53





## TASK 5 :

### ROUTE 53 FOR FAILURE ROUTING

To ensure high availability of application, implemented a DNS failover strategy using Amazon Route 53.

- Configuration: An 'A' record was created with a Failover routing policy.
- Primary Target: The Network Load Balancer in us-east-1
- Secondary Target: The Network Load Balancer in us-east-2
- Function: If the primary NLB becomes unhealthy, Route 53 automatically reroutes all traffic to the secondary region.

Domain Name : selmithreetier.com

Registered domains <a href="#">Info</a>				
<input type="text" value="Search domains by name"/>				
Domain name	Expiration date	Auto-renew	Transfer lock	
<a href="#">selmithreetier.com</a>	June 23, 2026	On	Off	

### Creating a failure routing A record to public hosted zone

selmithreetier.com

Hosted zone details

Edit hosted zone

Records (4)

DNSSEC signing

Hosted zone tags (0)

Records (1/4)

Info

Automatic mode is the current search behavior optimized for best filter results. [To change modes go to settings.](#)

Type

Routing p...

Alias

< 1 >

⚙️

<input type="checkbox"/>	Record ...	Type	Routin...	Differ...	Alias	Value/Route traffic to	TTL (s)
<input type="checkbox"/>	selmithre...	NS	Simple	-	No	ns-1447.awsdns-52.org. ns-717.awsdns-25.net. ns-216.awsdns-27.com. ns-1827.awsdns-36.co.uk.	17280
<input type="checkbox"/>	selmithre...	SOA	Simple	-	No	ns-1447.awsdns-52.org. aws...	900
<input checked="" type="checkbox"/>	app.selmi...	A	Failover	Primary	Yes	acac655f892824cb58fe8359...	-
<input type="checkbox"/>	app.selmi...	A	Failover	Secondary	Yes	ae42791e1b72140ccb6d586...	-

Record details

⚙️

>

Edit record

Record name

app.selmithreetier.com

Record type

A

Value

acac655f892824cb58fe835909  
f65563-  
e574172a6ac64731.elb.us-east-  
1.amazonaws.com.

Alias

Yes

TTL (seconds)

-

Routing policy

Failover

Failover record tune

Browse – “https://app.selmithreetier.com”

**Frontend of Task management APP – Adding a new task** (task3 – give title and description -> add task)

TaskFlow Dashboard

Organize Your Workflow

Add New Task

Task deleted successfully!

Task Title

task3

Task Description

Route 53 added succesfully

+ ADD TASK

Your Tasks (1)

task2

terraform deploy success

**Added task and stored in Database**

TaskFlow Dashboard

Organize Your Workflow

Add New Task

Task added successfully!

Task Title

Task Description

+ ADD TASK

Your Tasks (2)

task3

Route 53 added succesfully

task2

terraform deploy success

**Deleted one stored value**

TaskFlow Dashboard

Organize Your Workflow

Add New Task

Task deleted successfully!

Task Title

Task Description

+ ADD TASK

Your Tasks (1)

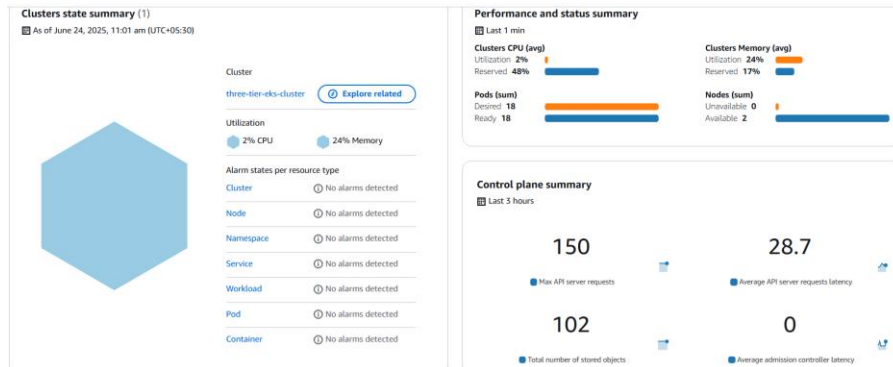
task3

Route 53 added succesfully

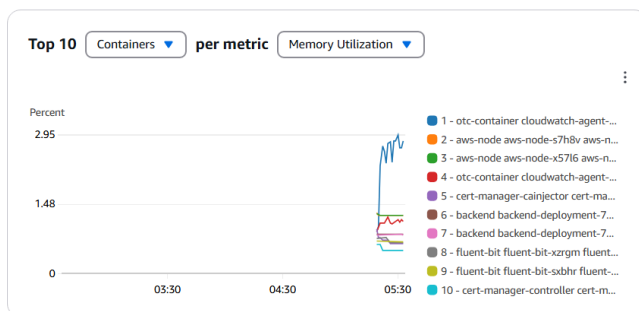
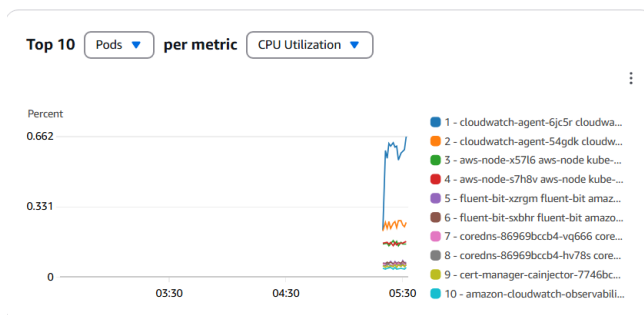
## TASK 6:

# CLOUD WATCH – MONITORING EKS CLUSTER THROUGH CLOUD WATCH

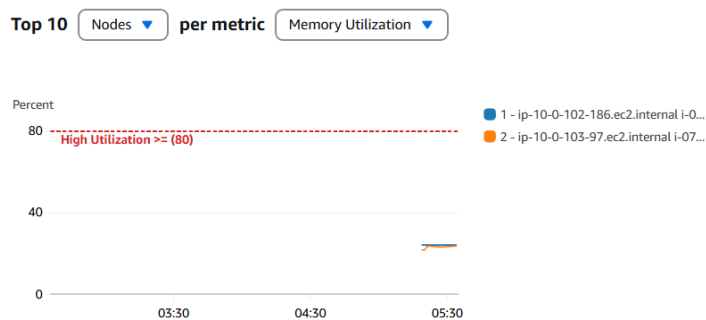
CloudWatch dashboard showing the "Clusters state summary" and graphs for CPU and Memory Utilization



## Pod cpu utilization monitoring



## nodes memory utilization monitoring



## TASK 7:

### LAMBDA AND EVENT BRIDGE

This solution ensures that whenever a pipeline execution fails, an automated email notification is sent using Amazon SNS, triggered by Amazon EventBridge and handled via an AWS Lambda function.

#### Workflow:

1. A Code Pipeline execution fails.
2. Amazon Event Bridge detects the state change.
3. An AWS Lambda function is triggered.
4. The Lambda function sends a message to an SNS topic.
5. SNS sends a formatted email notification to subscribers.

#### Lambda function

The screenshot shows the AWS Lambda console for a function named 'projectlambda'. The 'Function overview' tab is active, displaying a diagram of the function with no layers. To the right, the 'Description' section shows the function's last modified time as '5 minutes ago' and its ARN as 'arn:aws:lambda:us-east-1:390403857216:function:projectlambda'. Below the overview, the 'Code source' tab is selected, showing an 'Upload from' button.

#### Event bridge rule

The screenshot shows the AWS EventBridge console for a rule named 'projectrule'. The 'Rule details' tab is active, displaying the rule's status as 'Enabled', its ARN as 'arn:aws:events:us-east-1:390403857216:rule/projectrule', and its event bus ARN as 'arn:aws:events:us-east-1:390403857216:event-bus/default'. Below the details, the 'Event pattern' tab is selected, showing a JSON pattern that triggers on a 'CodePipeline Stage Execution State Change' with a state of 'FAILED'. The pattern is as follows:

```
1 {
2   "source": ["aws.codepipeline"],
3   "detail-type": ["CodePipeline Stage Execution State Change"],
4   "detail": {
5     "state": ["FAILED"]
6   }
7 }
```

Pipeline failed

Success

The most recent change will re-run through the pipeline. It might take a few moments for the status of the run to show in the pipeline view.

Developer Tools

CodePipeline

Pipelines

3-tier-new-eks

3-tier-new-eks

Edit

Stop execution

Create trigger

Clone pipeline

Release change

Pipeline

Executions

Triggers

Settings

Tags

Stage

Source

aad0cba5-27e0-4d9a-8511-935472037f44

All actions succeeded.

Source

GitHub (via GitHub App)

1 minute ago

4dd9f184

Source: add

Build

aad0cba5-27e0-4d9a-8511-935472037f44

In progress: 1

Build

AWS CodeBuild

Just now

4dd9f184

Source: add

Deploy

e0165303-6cc4-4796-935b-a73b3199022b

All actions succeeded.

Deploy

Amazon EKS

Jun 17, 2025 6:08 PM (UTC+5:30)

4dd9f184

Source: add

Lambda test success

Code

Test

Monitor

Configuration

Aliases

Versions

Executing function: succeeded (logs)

Details

Test event

Info

To invoke your function without saving an event, configure the JSON event, then choose Test.

CloudWatch Logs Live Tail

Save

Test

Test event action

Create new event

Edit saved event

Event name

test1

Maximum of 25 characters consisting of letters, numbers, dots, hyphens and underscores.

Event sharing settings

Private

This event is only available in the Lambda console and to the event creator. You can configure a total of 10. Learn more

Notification mail

[FAILED] Pipeline 3-tier-new-eks

AN

AWS Notifications<no-reply@sns.amazonaws.com>

To: Selmi Nazeeb Khan Seena Beegum(UST,IN)

Tue 6/24/2025 6:19 PM

External Email

Do not click any links or open any attachments unless you trust the sender and know the content is safe.

✖ Pipeline '3-tier-new-eks' has FAILED.

Execution ID: aad0cba5-27e0-4d9a-8511-935472037f44

Failed Stage: Error fetching stage: 'stageStates'

State: FAILED

--

If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe:

https://urldefense.com/v3/

https://sns.us-east-1.amazonaws.com/unsubscribe.html?SubscriptionArn=arn:aws:sns:us-east-1:390403857216:projectsns:75a43bfa-e330-490d-86d5-08cd8403ff8b&Endpoint=Selmi.NS@ust.com

!!PdM5GIU!Qx7YdsrwHE5U\_bF3OwSCTsxLYh8zilh9YDIkX48UFsdFMkJClqYcwzDS4gUPFIJ\_7QUvDXzSLaNRaQAvFo\$

Please do not reply directly to this email. If you have any questions or

[FAILED] Pipeline 3-tier-ne...

(No subject)

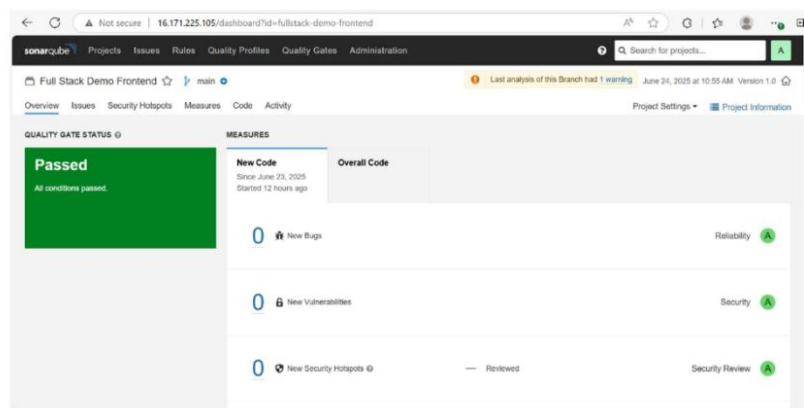
## TASK 8:

### SONARQUBE INTEGRATION

#### Create a Project in SonarQube

1. Log into the SonarQube UI
2. Go to **Projects > Create project**
3. Give it a name, like Full\_Stack\_backend
4. Choose "**Manually**" when asked how you'll set it up
5. Copy the generated token and save it securely — you'll use this in the buildspec.yml
6. Note down the **Project Key** and **SonarQube Server URL**

```
- sonar-scanner \  
  -Dsonar.projectKey=$SONAR_PROJECT_KEY \  
  -Dsonar.sources=. \  
  -Dsonar.host.url=$SONAR_HOST_URL \  
  -Dsonar.login=$SONAR_TOKEN
```



## TASK 9:

### CODE COMMIT

Code commit is service inside aws which can be used instead of github in source stage of code pipeline.

#### STEPS:

##### 1. Create a repository through CLI

- `aws codecommit create-repository --repository-name cloudops-demo --repository-description "My first CodeCommit repo"`

##### 2. Clone using Git:

- `git clone ssh://git-codecommit.us-east-1.amazonaws.com/v1/repos/cloudops-demo`

Repositories Info

Notify ▼

Clone URL ▼

View repository

Delete repository

Create repository

Q

< 1 > ⚙

Name ▼	Description	Last modified ▼	Clone URL	AWS KMS Key
<div></div> <div>cloudops-demo</div>	-	4 hours ago	<div> <div>HTTPS</div> <div>SSH</div> <div>HTTPS (GRC)</div> </div>	arn:aws:kms:us-east-1:365591124323:key/ee874354-d669-4782-9212-68ad7b425130

```

gdud9JCpFX59q3pnFULB4IdAKOBbrChIMZQZxfD/4gZ2oW7uvXLReWjy3AAVjzzWf39m2dLLiYp7vpL2bAxqCDstHXJ8K288x/f81b/BFGYJpBf1xWKO7g0Nzg8V/ewToKq3lSgcFz
TeYzmtBLgsVDG6hhTJksrIjotWtGfh8fVXxaEFckdIkM7IohAOZUVYPlbILfUvsnZlIdLWFxPU6u+FmMYFxl0hycV93721a5s000Zv/bMyMoNywQ007PMKOTKrmqp/8prAQSPb/PTY
nSpjIoLvabryunk88VuK4qay7hwpB7zeY70yx5tgf7oKgVpx7HGFNvuZwaVPfwc8U31xRM/79SASQGWt1LFsr+5dZAJ0K78qWvCV6YUfG6AveNX2d8zbYguzjItvYyPEmZMry1Z0g36X
1zf425gj27pDx/mXchtQEQ== codecommit-access
root@ip-172-31-86-56:~/.ssh# cd
root@ip-172-31-86-56:~# cd /home/ubuntu/
root@ip-172-31-86-56:/home/ubuntu# nano ~/.ssh/config
root@ip-172-31-86-56:/home/ubuntu# chmod 600 ~/.ssh/config
root@ip-172-31-86-56:/home/ubuntu# git clone ssh://git-codecommit.us-east-1.amazonaws.com/v1/repos/cloudops-demo
Cloning into 'cloudops-demo'...
The authenticity of host 'git-codecommit.us-east-1.amazonaws.com (52.94.226.180)' can't be established.
RSA key fingerprint is SHA256:eLMYlj0DKA4uvDZcl/KgtIayZANwX6t8+8isPtotBoY.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'git-codecommit.us-east-1.amazonaws.com' (RSA) to the list of known hosts.
Warning: You appear to have cloned an empty repository.
root@ip-172-31-86-56:/home/ubuntu# ssh git-codecommit.us-east-1.amazonaws.com
You have successfully authenticated over SSH. You can use Git to interact with AWS CodeCommit. Interactive shells are not supported. Connecti
on to git-codecommit.us-east-1.amazonaws.com closed by remote host.
Connection to git-codecommit.us-east-1.amazonaws.com closed.
root@ip-172-31-86-56:/home/ubuntu#

```

## TASK 10:

### CREATE S3 BUCKET, CLONE YOUR GITHUB REPO AND UPLOAD IN S3 THROUGH TERRAFORM

```
provider "aws" {
  region = "us-east-1"
}

resource "aws_s3_bucket" "project_bucket" {
  bucket = "selmi-bucket-123456"
  force_destroy = true

  tags = {
    Name          = "ProjectCodeBucket"
    Environment = "Dev"
  }
}

# Clone GitHub repo locally and upload to S3 using AWS CLI
resource "null_resource" "fetch_and_upload_code" {
  provisioner "local-exec" {
    command = <<EOT
      rm -rf /tmp/my-project
      git clone https://github.com/SelmiNazeeb/FinalProject-Devops.git /tmp/my-project
      aws s3 cp /tmp/my-project s3://${aws_s3_bucket.project_bucket.bucket}/ --recursive
    EOT
  }

  depends_on = [aws_s3_bucket.project_bucket]
}
```