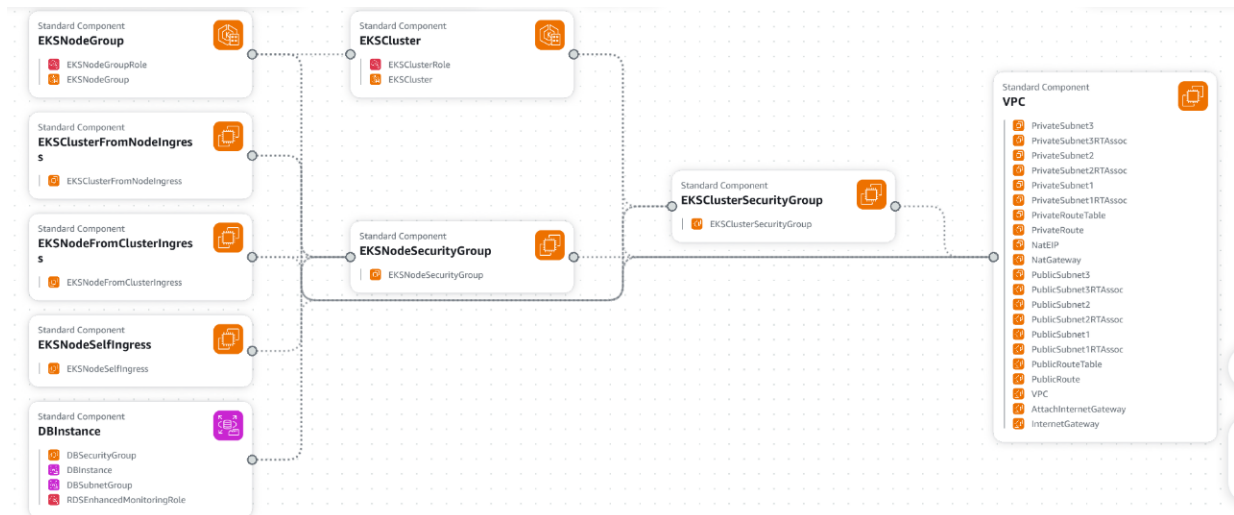
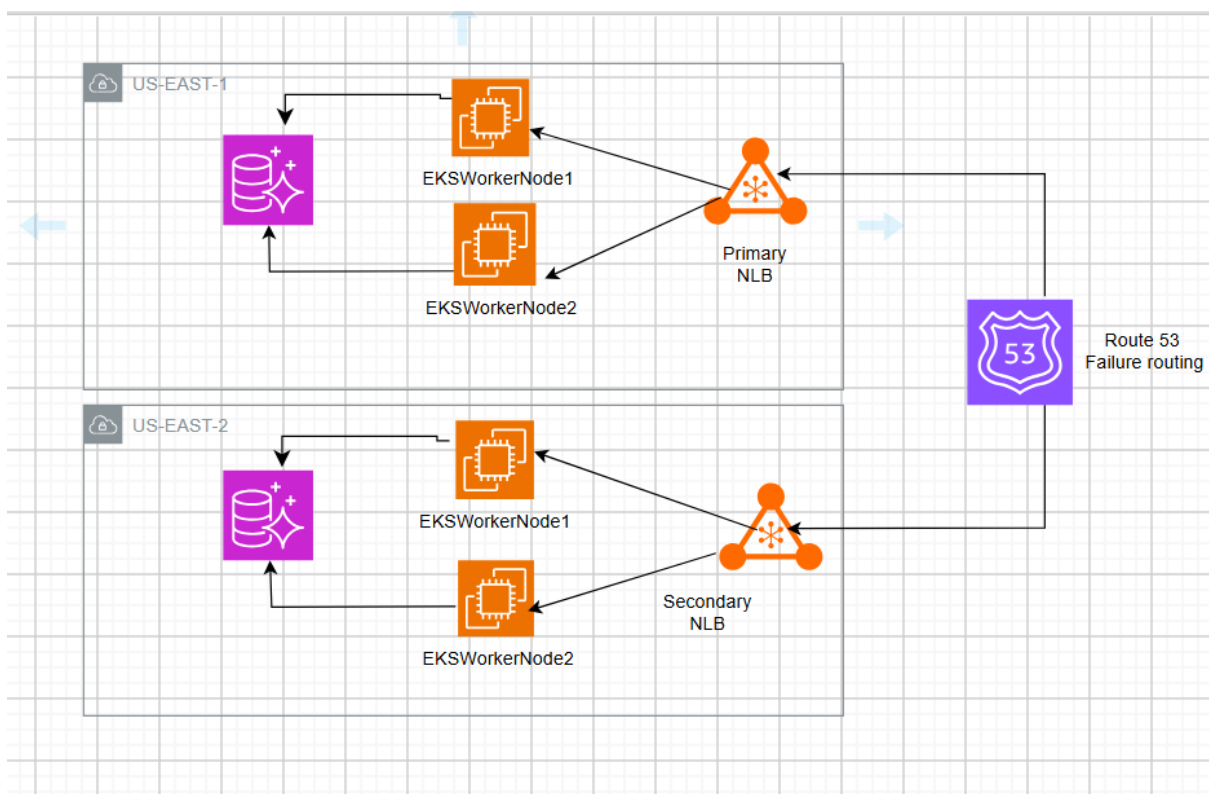


INFRASTRUCTURE

Infrastructure resources created through cloud formation and terraform (infrastructure seen in cloud formation infrastructure composer)



Overall infrastructure with route53 failover routing



Git repo containing the whole code

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

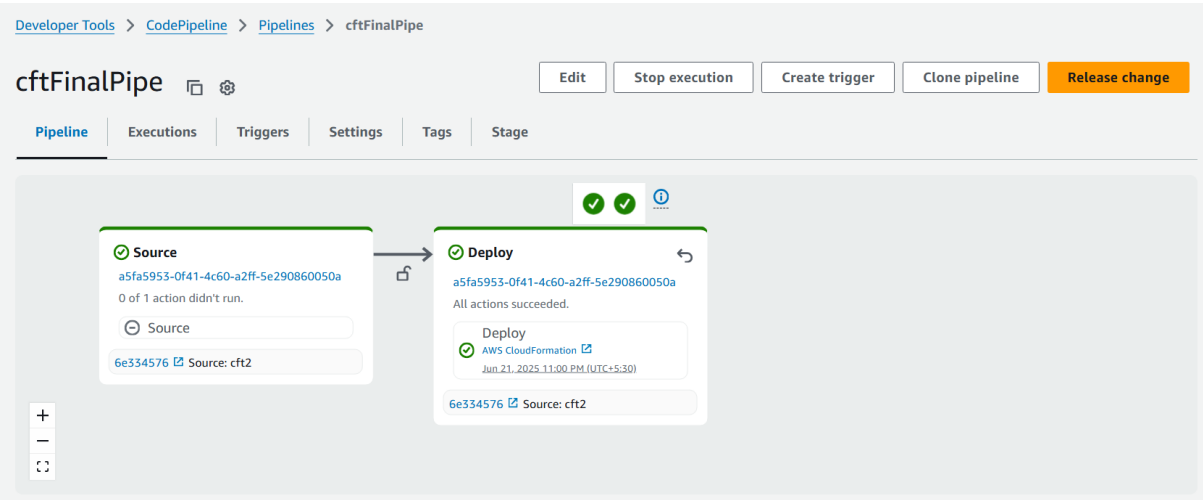
TASK1 :

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

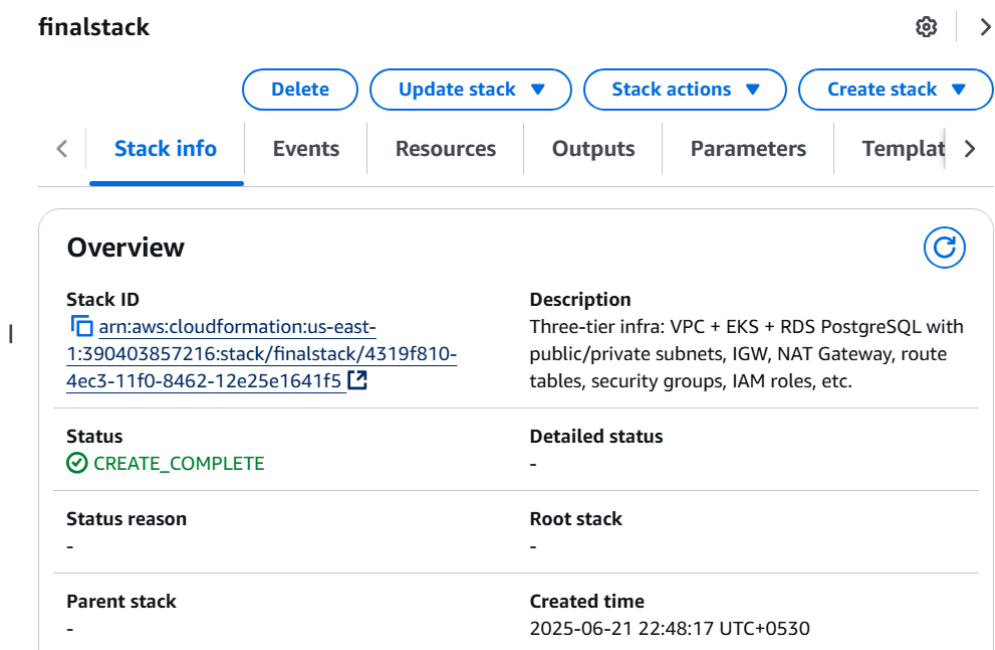
Permissions needed

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

Pipeline of infrastructure creation : success



Cloud formation stack created through pipeline



Resources created through stack

Resources (34)

Q Search resources

< 1 >

Logical ID	Physical ID	Type	Status	Module
AttachInternetGateway	IGWVpc-0a15dad6dfa654dd5	AWS::EC2::VPCGatewayAttachment	CREATE_COMPLETE	-
DBInstance	task-db-postgres	AWS::RDS::DBInstance	CREATE_COMPLETE	-
DBSecurityGroup	sg-0b884508425083b44	AWS::EC2::SecurityGroup	CREATE_COMPLETE	-
DBSubnetGroup	task-db-postgres-subnet-group	AWS::RDS::DBSubnetGroup	CREATE_COMPLETE	-
EKSCluster	three-tier-eks-cluster	AWS::EKS::Cluster	CREATE_COMPLETE	-
EKSClusterFromNodeIngress	sgr-05c25fc400cd957fb	AWS::EC2::SecurityGroupIngress	CREATE_COMPLETE	-
EKSClusterRole	finalstack-EKSClusterRole-3nVJOcCjuPVL	AWS::IAM::Role	CREATE_COMPLETE	-
EKSClusterSecurityGroup	sg-03d366fdb055a8abf	AWS::EC2::SecurityGroup	CREATE_COMPLETE	-
EKSNodeFromClusterIngress	sgr-0a9b238f8163ef5b0	AWS::EC2::SecurityGroupIngress	CREATE_COMPLETE	-
EKSNodeGroup	three-tier-eks-cluster/app-node-group	AWS::EKS::Nodegroup	CREATE_COMPLETE	-
EKSNodeGroupRole	finalstack-EKSNodeGroupRole-pgB2ZjUW4tVs	AWS::IAM::Role	CREATE_COMPLETE	-
EKSNodeSecurityGroup	sg-079b9e86378fbb1af	AWS::EC2::SecurityGroup	CREATE_COMPLETE	-
EKSNodeSelfIngress	sgr-0aa0fad27f71bd69e	AWS::EC2::SecurityGroupIngress	CREATE_COMPLETE	-
InternetGateway	lgw-021f3dab7cfea1f0b	AWS::EC2::InternetGateway	CREATE_COMPLETE	-
PrivateSubnet2RTAssoc	rtbassoc-0ed8252df0690f5f0	AWS::EC2::SubnetRouteTableAssociation	CREATE_COMPLETE	-
PrivateSubnet3	subnet-0bce62677f3a6f609	AWS::EC2::Subnet	CREATE_COMPLETE	-
PrivateSubnet3RTAssoc	rtbassoc-00c3d839b8c3b5875	AWS::EC2::SubnetRouteTableAssociation	CREATE_COMPLETE	-
PublicRoute	rtb-04f9c8025c03e2860 0.0.0.0/0	AWS::EC2::Route	CREATE_COMPLETE	-
PublicRouteTable	rtb-04f9c8025c03e2860	AWS::EC2::RouteTable	CREATE_COMPLETE	-
PublicSubnet1	subnet-06ee189155b033a24	AWS::EC2::Subnet	CREATE_COMPLETE	-
PublicSubnet1RTAssoc	rtbassoc-0ea7d0da7a5b0ca1f	AWS::EC2::SubnetRouteTableAssociation	CREATE_COMPLETE	-
PublicSubnet2	subnet-04b6cd1ed6eacdcbd	AWS::EC2::Subnet	CREATE_COMPLETE	-
PublicSubnet2RTAssoc	rtbassoc-0fe1c507ce20148ed	AWS::EC2::SubnetRouteTableAssociation	CREATE_COMPLETE	-
PublicSubnet3	subnet-02e0c106269a659a6	AWS::EC2::Subnet	CREATE_COMPLETE	-
PublicSubnet3RTAssoc	rtbassoc-0aee15f2ed39589a5	AWS::EC2::SubnetRouteTableAssociation	CREATE_COMPLETE	-
RDSEnhancedMonitoringRole	finalstack-RDSEnhancedMonitoringRole-m7vEBzEzbQLT	AWS::IAM::Role	CREATE_COMPLETE	-
VPC	vpc-0a15dad6dfa654dd5	AWS::EC2::VPC	CREATE_COMPLETE	-

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	https://07EC244C5F63FE19C7C8D99187E74F3.gr7.us-east-1.eks.amazonaws.com	-	finalstack-ClusterEndpoint
DBEndpoint	task-db-postgres.czs2i4megfm1.us-east-1.rds.amazonaws.com	-	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
task-db-postgres	🟢 Available	Instance	PostgreSQL	us-east-1c	db.t3.micro

ECR

AWS

🔍 Search

[Alt+S]

United States (N. Virginia)

terraform-user @ 390403857216-mm1 ▾

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
backendnew	390403857216.dkr.ecr.us-east-1.amazonaws.com/backendnew	June 20, 2025, 16:13:08 (UTC+05.5)	Mutable	AES-256
frontendnew	390403857216.dkr.ecr.us-east-1.amazonaws.com/frontendnew	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

Status

Active

Kubernetes version

1.28

Support period

⚠️ Extended support until November 26, 2025

Provider

EKS

Cluster health

🟢 0

Upgrade insights

🟢 6

Node health issues

🟢 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

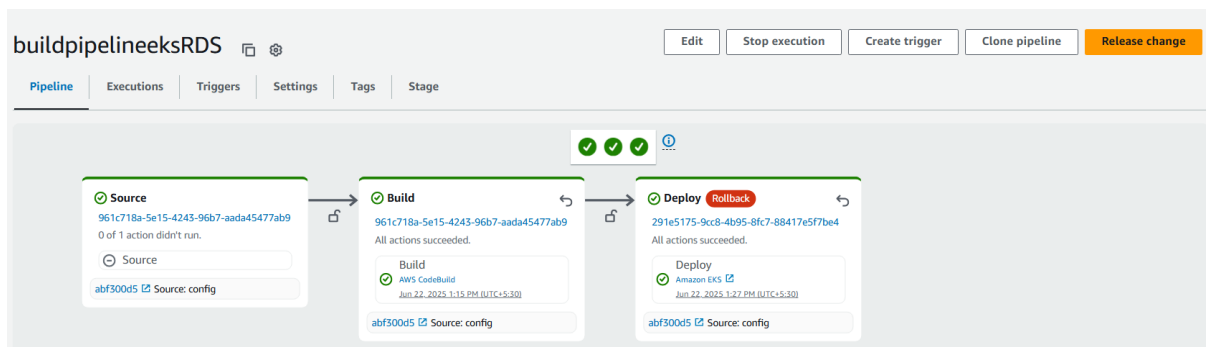
- pipeline role – Access entry in EKS cluster

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						< 1 >			
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



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

Load Balancer created for the application

Load balancers (2)								Actions	Create load balancer
Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.									
Filter load balancers								< 1 >	
<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,		

acac655f892824cb58fe835909f65563

Actions

▼ Details

Load balancer type

Network

Status

Active

VPC

[vpc-0a15dad6dfa654dd5](#)

Load balancer IP address type

IPv4

Scheme

Internet-facing

Hosted zone

Z26RL4JYFTOTI

Availability Zones

[subnet-06ec189155b033a24](#) us-east-1a (use1-az4)

[subnet-04b6cd1ed6eacdbbd](#) us-east-1b (use1-az6)

[subnet-02e0c106269a659a6](#) us-east-1c (use1-az1)

Date created

June 22, 2025, 12:11 (UTC+05:30)

Load balancer ARN

[arn:aws:elasticloadbalancing:us-east-1:390403857216:loadbalancer/net/acac655f892824cb58fe835909f65563/c574172a6ac64731](#)

DNS name

[acac655f892824cb58fe835909f65563-c574172a6ac64731.elb.us-east-1.amazonaws.com](#) (A Record)

Listeners

Network mapping

Resource map

Security

Monitoring

Integrations

Attributes

Capacity

Tags

Listeners (1)

Actions

Add listener

A listener checks for connection requests using the protocol and port that you configure. Traffic received by a Network Load Balancer listener is forwarded to the selected target group.

Filter listeners

< 1 >

☐

Protocol:Port

▼

☐

Default action

▼

☐

ARN

▼

☐

Security policy

▼

☐

Default SSL/TLS certificate

▼

☐

ALPN policy

▼

☐

Tags

▼

☐

[TCP-80](#)

Forward to target group

[k8s-tasksn-frontend-f6ae5deda7](#)

[ARN](#)

Not applicable

Not applicable

None

[2 tags](#)

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 top navigation bar includes 'Developer Tools', 'CodeBuild', 'Build projects', and 'terraform-build'. Below the navigation bar, there are buttons for 'Actions', 'Create trigger', 'Edit', 'Clone', 'Debug build', 'Start build with overrides', and 'Start build'.

Configuration

Source provider	Primary repository	Artifacts upload location	Service role
GitHub	SelmiNazeeb/FinalProject-Devops	-	arn:aws:iam::390403857216:role/service-role/codebuild-terraform-build-service-role

Public builds: Disabled

Build history

Build run	Status	Build number	Source version	Submitter	Duration	Completed
terraform-build:e843e6de-8558-4b41-80b0-595f8f7acb80	Succeeded	43	c5c9cbe868531e-a361b2b2a840f6-2df68996e6c1	GitHub-Hookshot/fb6e15e	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 left sidebar includes 'DynamoDB', 'Dashboard', 'Tables', 'Explore items', ' PartiQL editor', 'Backups', 'Exports to S3', 'Imports from S3', 'Integrations', 'Reserved capacity', and 'Settings'.

terraform-lock-table

Scan or query items: ☒ Scan ☐ Query

Select a table or index: Table - terraform-lock-table

Select attribute projection: All attributes

Filters - optional:

Completed - Items returned: 1 - Items scanned: 1 - Efficiency: 100% - RCUs consumed: 2

Table: terraform-lock-table - Items returned (1)

Scan started on June 24, 2025, 00:02:50

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](#)

Find objects by prefix

Show versions

< 1 >

<input type="checkbox"/>	Name	Type	Last modified	Size
<input type="checkbox"/>	terraform.tfstate	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"
```

Resources created

RDS

Databases (1)

Group resources

Modify

Actions

Create database

Filter by databases

< 1 >

<input type="checkbox"/>	DB identifier	Status	Role	Engine	Region ...	Size
<input type="radio"/>	task-db-postgres	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 (2)

View push commands

Delete

Actions

Create repository

Search by repository substring

Repository name

URI

Created at

Tag immutability

Encryption type

<input type="radio"/>	backendnew	390403857216.dkr.ecr.us-east-2.amazonaws.com/backendnew	June 23, 2025, 15:28:03 (UTC+05.5)	Mutable	AES-256
<input type="radio"/>	frontendnew	390403857216.dkr.ecr.us-east-2.amazonaws.com/frontendnew	June 23, 2025, 15:27:51 (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](#)

Status
🟢 Active

Cluster health
🟢 0

Kubernetes version [Info](#)
1.28

Upgrade insights
🟢 6

Support period
⚠️ Extended support until November 26, 2025

Node health issues
🟢 0

Provider
EKS

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

Access configuration [Manage access](#)

Authentication mode
EKS API and ConfigMap

IAM access entries (4) [Info](#)

[View details](#) [Edit](#) [Delete](#) [Create access entry](#)

< 1 >

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 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](#)

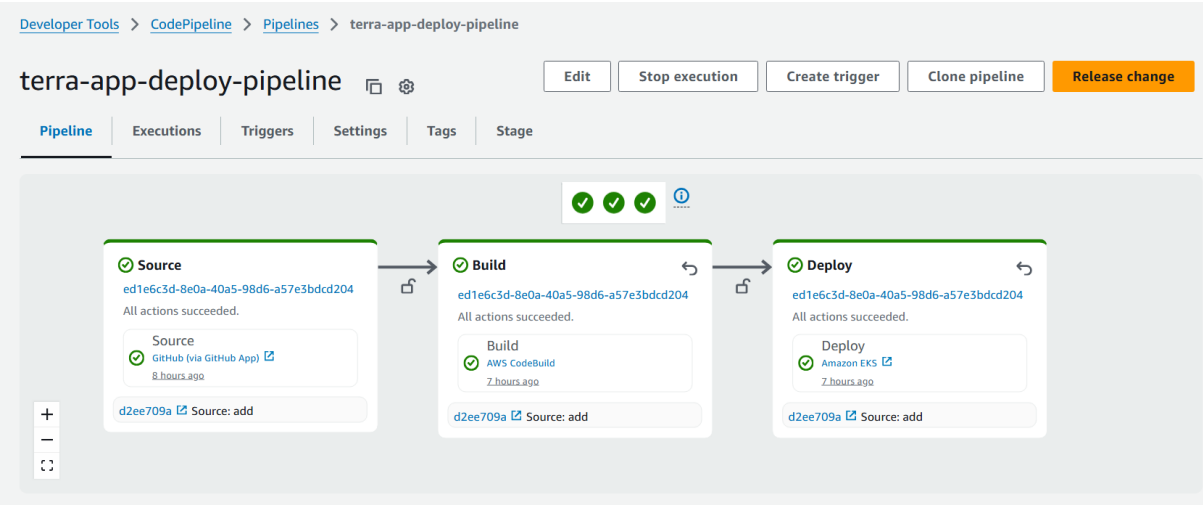
[View details](#) [Edit](#) [Delete](#) [Create access entry](#)

< 1 >

IAM principal ARN	Type	Username	Group names	Access policies
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



>>> 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="checkbox"/>	Name	DNS name	State	VPC ID	Availability Zones	Type
<input type="checkbox"/>	ae42791e1b72140ccb6...	ae42791e1b72140ccb6d58...	Active	vpc-02b566a9f4f50daec	2 Availability Zones	network

ae42791e1b72140ccb6d586e27168a6b [Actions](#)

Details

Load balancer type Network	Status Active	VPC vpc-02b566a9f4f50daec	Load balancer IP address type IPv4
Scheme Internet-facing	Hosted zone ZLMOA37VPKANP	Availability Zones subnet-08dd5ceaf6d578809 us-east-2b (use2-az2) subnet-052ebb6f8c08320a7 us-east-2a (use2-az1)	Date created June 23, 2025, 15:41 (UTC+05:30)
Load balancer ARN arn:aws:elasticloadbalancing:us-east-2:390403857216:loadbalancer/net/ae42791e1b72140ccb6d586e27168a6b/64186227eb34f237		DNS name Info ae42791e1b72140ccb6d586e27168a6b-64186227eb34f237.elb.us-east-2.amazonaws.com (A Record)	

Listeners (1) [Actions](#) [Add listener](#)

A listener checks for connection requests using the protocol and port that you configure. Traffic received by a Network Load Balancer listener is forwarded to the selected target group.

<input type="checkbox"/>	Protocol:Port	Default action	ARN	Security policy	Default SSL/TLS certificate	ALPN policy	Tags
<input type="checkbox"/>	TCP:80	Forward to target group k8s-tasks-frontend-c16bab7844	ARN	Not applicable	Not applicable	None	2 tags

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)

The screenshot shows the TaskFlow Dashboard interface. The header is dark green with 'TaskFlow Dashboard' on the left and 'Organize Your Workflow' on the right. The main content area is light blue. On the left, the 'Add New Task' section has a green success message 'Task deleted successfully!'. Below it, the 'Task Title' field contains 'task2' and the 'Task Description' field contains 'terraform deploy success'. A green '+ ADD TASK' button is at the bottom. On the right, the 'Your Tasks (1)' section shows a single task: 'task' with description 'new', accompanied by edit and delete icons.

Added task and stored in Database

This screenshot shows the TaskFlow Dashboard after adding a new task. The 'Add New Task' section now displays a green success message 'Task added successfully!'. The 'Task Title' field is empty, and the 'Task Description' field is empty. The '+ ADD TASK' button remains. The 'Your Tasks (2)' section now lists two tasks: 'task2' with description 'terraform deploy success' and 'hello' with description 'added route53'. Each task has edit and delete icons.

TASK 5 :

ROUTE 53 FOR FAILURE ROUTING

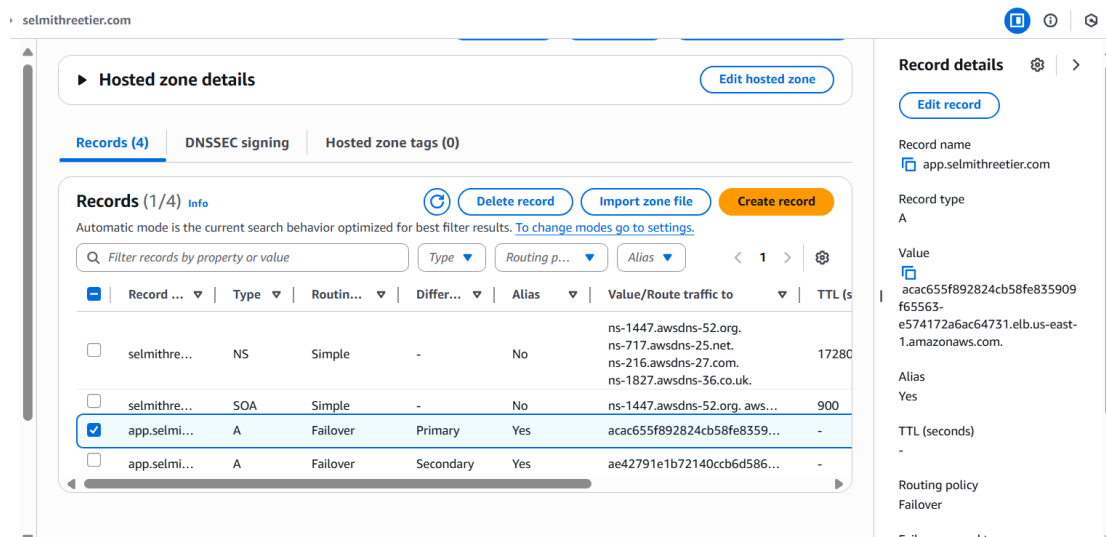
Domain Name : selmithreetier.com

The screenshot shows the 'Registered domains' page in the Amazon Route 53 console. At the top, there are buttons for 'Download billing report', 'Transfer in', and 'Register domains'. Below these is a search bar with the placeholder 'Search domains by name'. The main table lists the registered domains. The table has columns for 'Domain name', 'Expiration date', 'Auto-renew', and 'Transfer lock'. The only domain listed is 'selmithreetier.com'.

Domain name	Expiration date	Auto-renew	Transfer lock
selmithreetier.com	June 23, 2026	On	Off

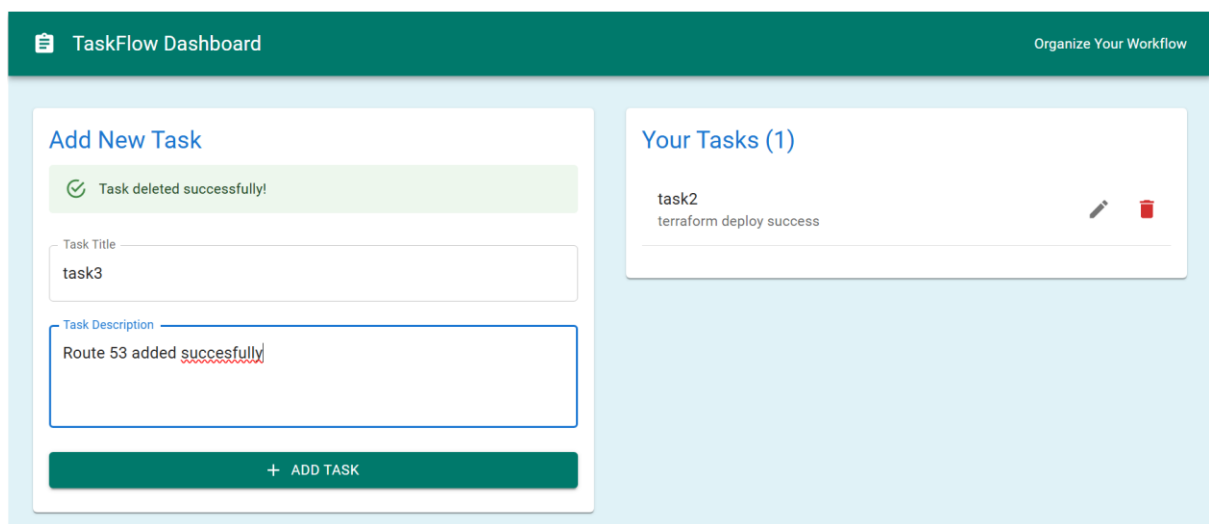
Creating a failure routing A record to public hosted zone

- Set primary load balancer as LB in region1 and secondary as LB in region2
- If primary load balancer fails routes the traffic to secondary load balancer

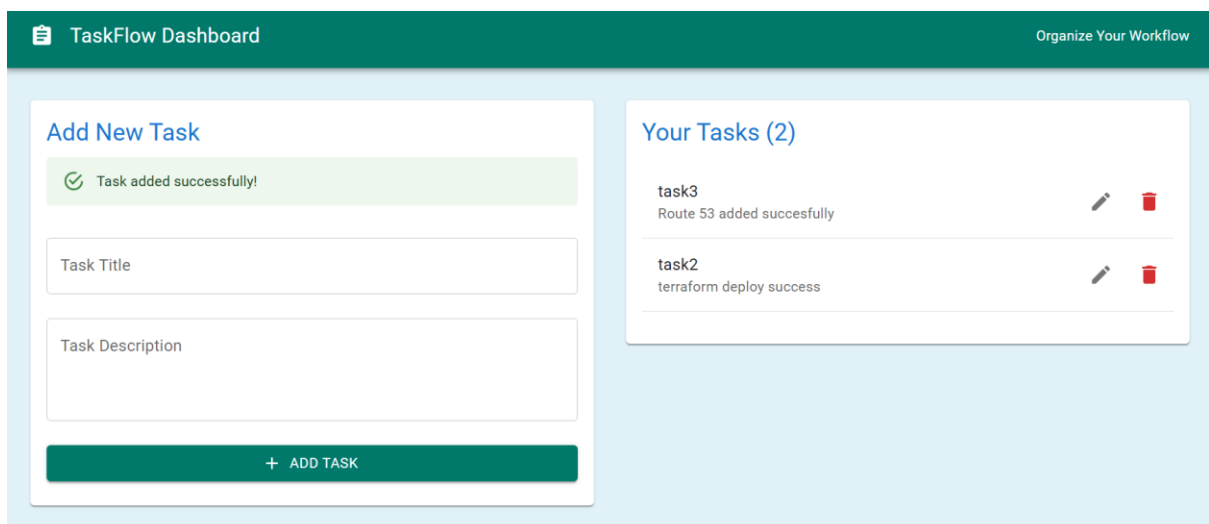


Browse – “https://app.selmithreetier.com”

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



Added task and stored in Database



Deleted one stored value

TaskFlow Dashboard

Organize Your Workflow

Add New Task

Task deleted successfully!

Task Title

Task Description

+ ADD TASK

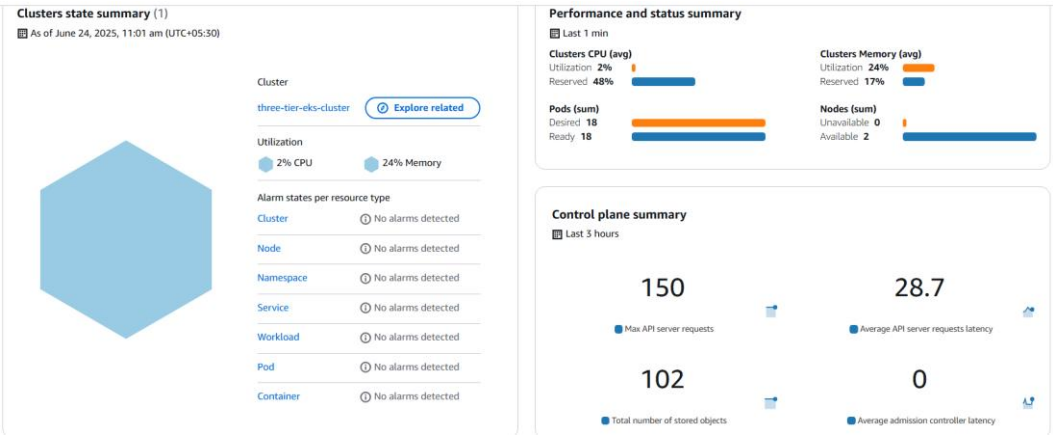
Your Tasks (1)

task3

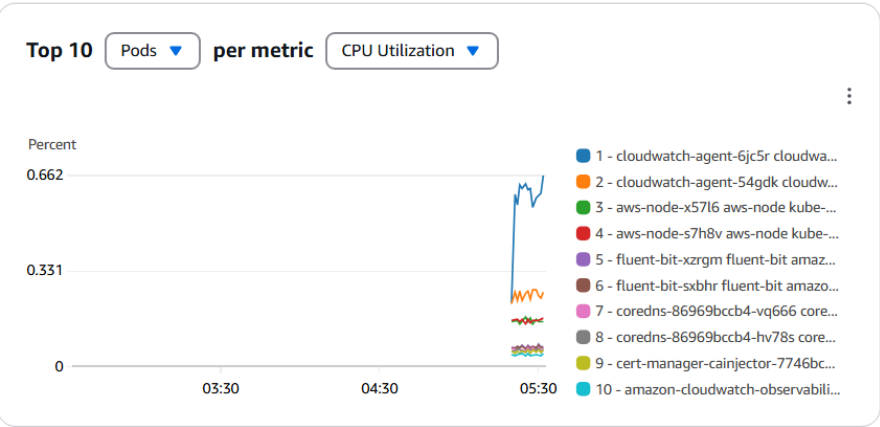
Route 53 added succesfully

TASK 6:

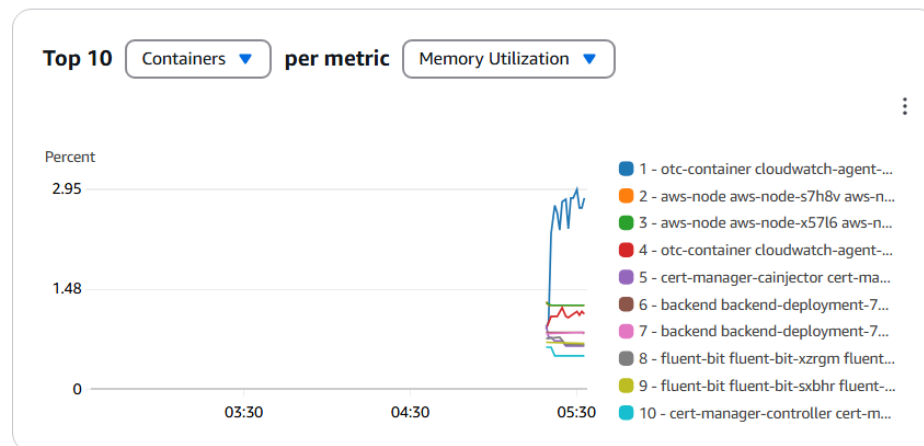
CLOUD WATCH – MONITORING EKS CLUSTER THROUGH CLOUD WATCH



Pod cpu utilization monitoring



Container memory utilization monitoring



nodes memory utilization monitoring

