

ECS Fargate Nginx Server Deployment with ALB And EFS using CodePipeline - Terraform (Task 8)



Zaeem Attique Ashar
Cloud Intern

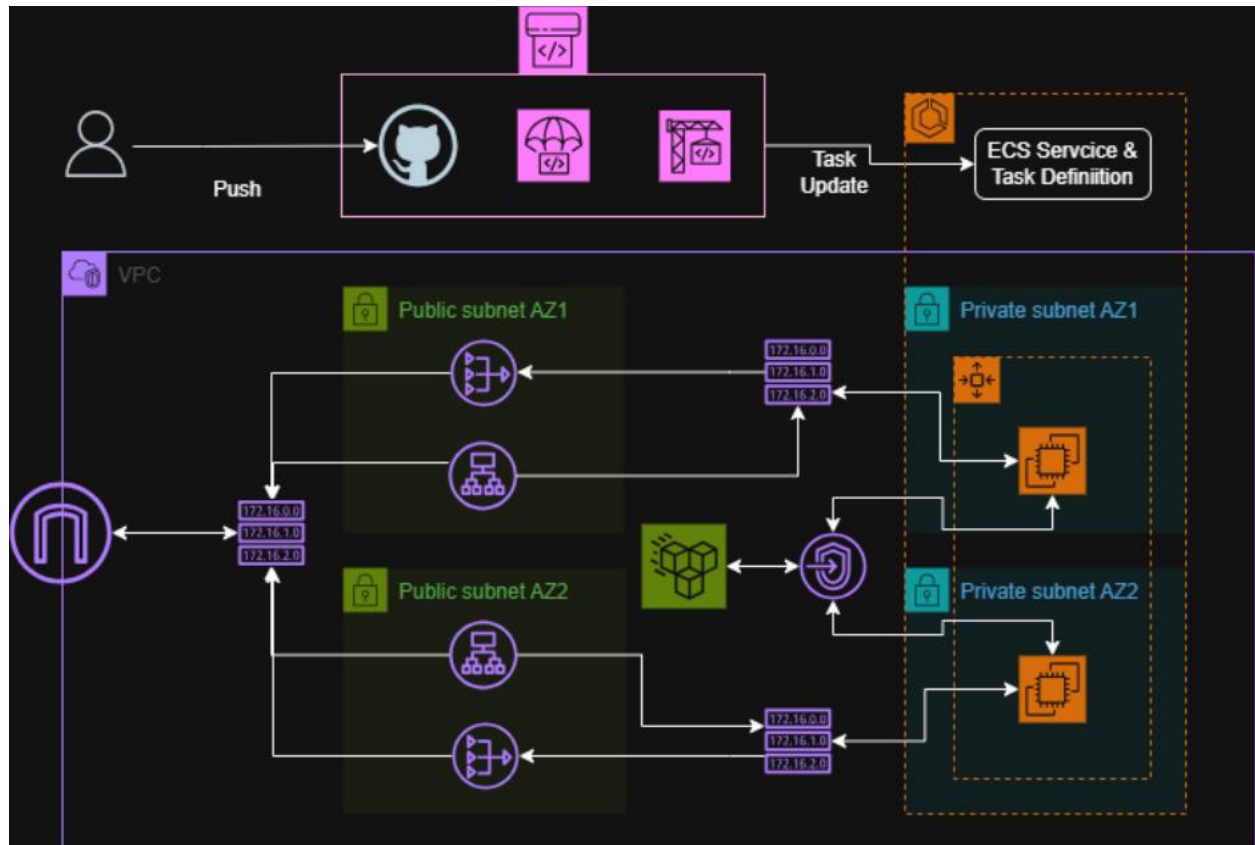
Task Description:

This task will be a guide on setting up a Node Application server with a highly available architecture on AWS ECS Cluster Spread over multiple availability zones. An internet facing Load Balancer will be set up in front of the ECS cluster to listen to traffic and balance it upon tasks. AWS EFS will be used for persistence in shared storage to prevent data loss. The application will automatically be deployed using the AWS CodePipeline. The complete infrastructure will be coded in Terraform modules.

Architecture Diagram: 3

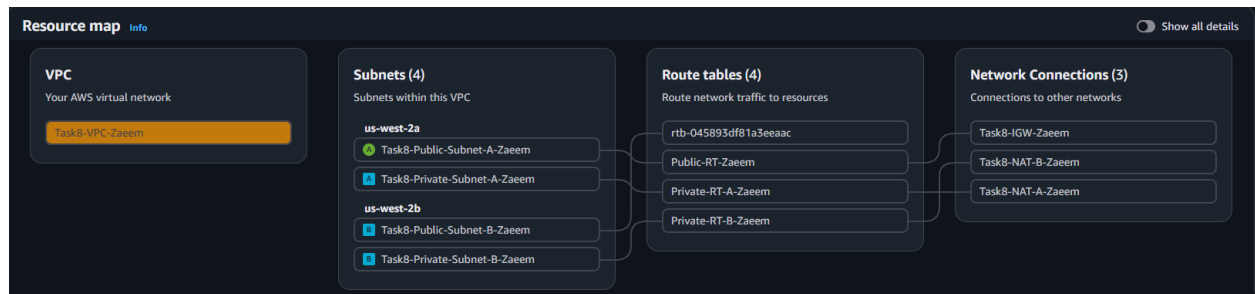
Task8.1: Create basic networking infrastructure	4
Task8.2: Create EFS File System and Mount Points	4
Task8.3: Build Docker Image and push to ECR.....	6
Task8.4: Create an ECS + EC2 Cluster and dependencies	6
Task8.5: Creating the CodePipeline	11
Task8.6: Verification and Testing	12
Task8.7: Problems Faced	13

Architecture Diagram:



Task8.1: Create basic networking infrastructure

- Create and configure a VPC
 - CIDR Block: 10.0.0.0/16
- Create and configure Subnets
 - Public Subnet A (us-west-2a), CIDR: 10.0.1.0/24
 - Private Subnet A (us-west-2a), CIDR: 10.0.2.0/24
 - Public Subnet B (us-west-2b), CIDR: 10.0.3.0/24
 - Private Subnet A (us-west-2a), CIDR: 10.0.4.0/24
- Create and configure NAT Gateways
 - NAT Gateway A in Public Subnet A
 - NAT Gateway B in Public Subnet B
- Create and configure Internet Gateway
 - Create and attach to the project's VPC
- Create and configure Route Tables
 - Public Route Table, Outbound rule: 0.0.0.0/0 -> IGW, attach to Public SN A&B
 - Private Route Table A, Outbound Rule: 0.0.0.0/0 -> NGW attach to Private SN A
 - Private Route Table B, Outbound Rule: 0.0.0.0/0 -> NGW attach to Private SN B



Task8.2: Create EFS File System and Mount Points

- Create File System:
 - Name: Task8-EFS-Zaeem
 - Encryption: Yes
- Create Access Points:
 - File System: Task8-EFS-Zaeem

- o Name: Task8-EFS-AP-Zaeem
- o Root Directory: /
- o POSIX UID 1000, GID 1000
- o Owner UID 1000, Owner GID: 1000
- o AP Permissions: 0755
- Create Mount Points:
 - o AZ us-west-2a, Subnet Private, SG Task8-EFS-SG-Zaeem
 - o AZ us-west-2b, Subnet Private, SG Task8-EFS-SG-Zaeem

Task8-EFS-Zaeem (fs-047d1f61669f299a9)

DeleteAttach

General

Edit

Amazon resource name (ARN)

arn:aws:elasticfilesystem:us-west-2:880958245574:file-system/fs-047d1f61669f299a9

Performance mode

General Purpose

Throughput mode

Bursting

Lifecycle management

Transition into Infrequent Access (IA): None

Transition into Archive: None

Transition into Standard: None

Availability zone

Regional

Automatic backups

Disabled

Encrypted

No

File system state

Available

DNS name

fs-047d1f61669f299a9.efs.us-west-2.amazonaws.com

Replication overwrite protection

Enabled

Task8.3: Build Docker Image and push to ECR

- Build Docker Image:
 - o Got to source code directory.
 - o Create DockerFile.yaml and write instructions
 - o Use command to build image: *docker image -t nodejs:latest -f DockerFile.yaml .*
 - o Use to tag image: *docker tag nodejs:latest <ecr uri>/zaeem/Task8:latest*
 - o Use command to push to ECR: *docker push <ecr uri>/zaeem/Task8:latest*

```
PS C:\Users\zaeem\Documents\Innovation Lab - Cloudelligent\Task7\node-js-sample-master> aws ecr get-login-password --region us-west-2 | docker login --username AWS --password-stdin 504649076991.dkr.ecr.us-west-2.amazonaws.com
Login Succeeded
PS C:\Users\zaeem\Documents\Innovation Lab - Cloudelligent\Task7\node-js-sample-master> docker build -t zaeem/task7 -f .\DockerFile.yaml .
[+] Building 1.9s (10/10) FINISHED
-> [internal] load build definition from DockerFile.yaml
-> => transferring dockerfile: 199B
-> [internal] load metadata for docker.io/library/node:18-alpine
-> [internal] load .dockerignore
-> => transferring context: 2B
-> [1/5] FROM docker.io/library/node:18-alpine@sha256:8d6421d663bd28fd3ebc480332f249011d118945588d0a15cb9bc4b8ca09d9e
-> => resolve docker.io/library/node:18-alpine@sha256:8d6421d663bd28fd3ebc480332f249011d118945588d0a15cb9bc4b8ca09d9e
-> [internal] load build context
-> => transferring context: 41.69kB
-> CACHED [2/5] WORKDIR /usr/src/app
-> CACHED [3/5] COPY package*.json ./
-> CACHED [4/5] RUN npm install --production
-> CACHED [5/5] COPY . .
-> exporting to image
-> => exporting layers
-> => exporting manifest sha256:5fd261775c0ca9c4c083335a6079530c6dcfc9660818527c9fb229c3dbffed2c
-> => exporting config sha256:80c04d551154129d48c506c4448c332b7bd21b027897c9f4e30c77794dec
-> => exporting attestation manifest sha256:1fd7b12ccae809c20d9395a03d33e51fc308c5e6d41313dackeeefb92ae836a6
-> => exporting manifest list sha256:730e95da8cf21f547b178ffa99695da5132466766b5617c8d81335f30b9942e08
-> => naming to docker-desktop/finchboat/build/desktop-linux/desktop-linux/9mli3thrz9wfbzrn42bncqdm
-> => unpacking to docker-desktop/finchboat/build/desktop-linux/desktop-linux/9mli3thrz9wfbzrn42bncqdm (id + chd)
View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/9mli3thrz9wfbzrn42bncqdm
PS C:\Users\zaeem\Documents\Innovation Lab - Cloudelligent\Task7\node-js-sample-master> docker tag zaeem/task7:latest 504649076991.dkr.ecr.us-west-2.amazonaws.com/zaeem/task7:latest
PS C:\Users\zaeem\Documents\Innovation Lab - Cloudelligent\Task7\node-js-sample-master> docker push 504649076991.dkr.ecr.us-west-2.amazonaws.com/zaeem/task7:latest
The push refers to repository [504649076991.dkr.ecr.us-west-2.amazonaws.com/zaeem/task7]
79b00165671e: Pushed
a2a0fd26c4ff: Pushed
1e5ak489cae5: Pushed
25f72a83641: Pushed
96805a352ee2: Pushed
dd71dde834b5: Pushed
779b8e4cf363: Pushed
8c0c2c8931c7: Pushed
f18232174bc9: Pushed
latest: digest: sha256:730e95da8cf21f547b178ffa99695da5132466766b5617c8d81335f30b9942e08 size: 856
PS C:\Users\zaeem\Documents\Innovation Lab - Cloudelligent\Task7\node-js-sample-master>
```

Task8.4: Create an ECS + EC2 Cluster and dependencies

- Create and Configure a Launch Template
 - o Name: Task8-EC2-LT-Zaeem
 - o Container instance AMI: Amazon Linux 2023
 - o Instance Type: t3.micro
 - o SSH Key pair: Task8-EC2
 - o Subnet: Do not include
 - o Availability Zone: Do not include
 - o Security Group: Task8-EC2-SG-Zaeem
 - o Storage Volume: 8GiB Default EBS volume

Task8-Launch-Template-Zaeem-20251202141903412000000005 (lt-03dadf894321fd96f)

Actions
Delete template

Launch template details

Launch template ID

lt-03dadf894321fd96f

Launch template name

Task8-Launch-Template-Zaeem-20251202141903412000000005

Default version

1

Owner

arn:aws:iam::880958245574:user/Zaeem

Details
Versions
Template tags

Launch template version details

Version

1 (Default)

Description

-

Date created

2025-12-02T14:19:03.000Z

Created by

arn:aws:iam::880958245574:user/Zaeem

Instance details

Storage
Resource tags
Network interfaces
Advanced details

AMI ID

ami-07b09ad3acff075f

Instance type

t3.micro

Availability Zone

-

Availability Zone Id

-

Key pair name

-

Security groups

-

Security group IDs

sg-07c3ab8569e31bb5d

- Create and Configure Application Load Balancer
 - o Name: Task8-ALB-Zaeem
 - o Scheme: Internet Facing
 - o Load Balancer IP: IPv4
 - o VPC: Task8-VPC-Zaeem
 - o AZ and Subnets: AZ1 Public SN & AZ2 Public SN
 - o Security Groups: Task8-ALB-SG-Zaeem
 - o Listener Protocol HTTP, Port 80
 - o Routing Action: Forward to TG
 - o Register Targets: None

Task8-ALB-A-Zaeem

Details

Load balancer type

Application

Status

Active

VPC

vpc-08e12832b2d654c42

Load balancer IP address type

IPv4

Scheme

Internet-facing

Hosted zone

Z1H1FL5HABSF5

Availability Zones

[subnet-0eefe88e1cad33b17](#) us-west-2b (usw2-az1)
[subnet-097056f4ad974d766](#) us-west-2a (usw2-az2)

Date created

December 2, 2025, 19:18 (UTC+05:00)

Load balancer ARN

arn:aws:elasticloadbalancing:us-west-2:880958245574:loadbalancer/app/Task8-ALB-A-Zaeem/3330680984198999

DNS name info

Task8-ALB-A-Zaeem-1969410025.us-west-2.elb.amazonaws.com (A Record)

Listeners and rules

Network mapping

Resource map

Security

Monitoring

Integrations

Attributes

Capacity

Tags

Listeners and rules (1)

Manage rules

Manage listener

Add listener

A listener checks for connection requests on its configured protocol and port. Traffic received by the listener is routed according to the default action and any additional rules.

Filter listeners

< 1 >

Protocol:Port

Default action

Rules

ARN

Security policy

Default SSL/TLS

HTTP:5000	<ul style="list-style-type: none"> Forward to target group 	Task8-ALB-Target-Group-Zaeem : 1 (100%) Target group stickiness: Off	1 rule	ARN	Not applicable	Not applicable
-----------	---	---	--------	-----	----------------	----------------

- Create and Configure Auto Scaling Group
 - o Name: Task8-ASG-Zaeem
 - o Launch Template: Task8-EC2-LT-Zaeem
 - o VPC: Task8-VPC-Zaeem
 - o AZ: Private SN A & Private SN B
 - o Load Balancer: Attach to existing
 - o Attach to an Existing Target Group
 - o Group Size: Desired capacity 2, min 1, max 3
 - o No scaling policies

Task8-ECS-Capacity-Provider-Zaeem

Capacity provider overview

Capacity provider name

Task8-ECS-Capacity-Provider-Zaeem

Scaling type

EC2 Auto Scaling

Update status

-

Status

Active

Cluster name

Task8-ECS-Cluster-Zaeem

CloudFormation stack

-

Capacity provider ARN

arn:aws:ecs:us-west-2:880958245574:capacity-provider/Task8-ECS-Capacity-Provider-Zaeem

- Create and Configure ECS Cluster
 - o Name: Task8-ECS-Cluster-Zaeem
 - o Infrastructure: Fargate and self-managed instances
 - o Select Task8-ASG-Zaeem Auto Scaling Group

Task8-ECS-Cluster-Zaeem

ASG

Last updated

December 2, 2025, 20:47 (UTC+5:00)

Actions

Create with Express Mode

Cluster overview

ARN

arn:aws:ecs:us-west-2:880958245574:cluster/Task8-ECS-Cluster-Zaeem

Status

Active

CloudWatch monitoring

Default

Registered container instances

2

Services

Draining

-

Active

1

Tasks

Pending

-

Running

2

Services

Tasks

Infrastructure

Metrics

Scheduled tasks

Configuration

Event history

Tags

Services (1) Info

Last updated

December 2, 2025, 20:47 (UTC+5:00)

Manage tags

Update

Delete service

Create

Filter services by value

Filter launch type

Any launch type

Filter scheduling strategy

Any scheduling strategy

Filter resource management type

Any resource management type

< 1 >

Service name

ARN

Status

Schedu...

La...

Task de...

Deployments and tasks

Task8-ECS-Service-Zaeem

arn:aws:ecs:us-v

Active

REPLICA

EC2

Task8-ECS...

2/2 Ta

- Create a Task Definition for the ECS Cluster
 - o Name: Task8-NodeJS-Zaeem
 - o Infrastructure Requirements: Amazon EC2 Instances
 - o OS&Arch: Linux/x64, Network Mode: bridge
 - o vCPU: 1, Memory: 3GB
 - o Select Task role and Task Execution Roles
 - o Container 1 (essential): Name NodeJS-App, ECR Image URI, Port Mapping 3000 to 80 for accessing the nodejs application
 - o Storage: Configure at task definition, Vol Type: EFS, Enter EFS ID, Enter AP ID

Task8-ECS-Task-Definition-Zaeem:17

Last updated

December 2, 2025, 20:51 (UTC+5:00)

Deploy

Actions

Create new revision

Overview

Info

ARN

am:aws:ecs:us-west-2:880958245574:task-definition/Task8-ECS-Task-Definition-Zaeem:17

Status

ACTIVE

Time created

December 2, 2025, 20:34 (UTC+5:00)

App environment

EC2

Task role

Task8-ecs-task-role-Zaeem

Task execution role

Task8-ecs-task-execution-role-Zaeem

Operating system/Architecture

-

Network mode

bridge

Fault injection

-

Containers

JSON

Task placement

Volumes (1)

Requires attributes

Tags

Task size

Task CPU

256 units (0.25 vCPU)

Task CPU maximum allocation for containers

CPU (unit)

0

20

40

60

80

100

120

140

160

180

200

220

240

NodeJS-App

Shared task CPU

Task memory

256 MiB (0.25 GB)

Task memory maximum allocation for container memory reservation

Memory (MiB)

0

20

40

60

80

100

120

140

160

180

200

220

240

NodeJS-App

Shared task memory

- Create ECS Service to run Task Definition:
 - o Family: Task8-NodeJS-Zaeem
 - o Svc Name: Task8-NodeJS-Zaeem-Service
 - o Compute option: Capacity Provider Strategy, custom: base 2, weight 1
 - o Scheduling Strategy: Replica
 - o Desired Task: 2
 - o Load Balancer: Task8-VPC-Zaeem, Type: ALB, Container: NodeJS-App, Load Balancer: Task8-ALB-Zaeem

Task8-ECS-Service-Zaeem

Info

Last updated

December 2, 2025, 20:50 (UTC+5:00)

Delete service

Update service

Service overview

Status

Active

Tasks (2 Desired)

0 Pending | 2 Running

Task definition: revision

Task8-ECS-Task-Definition-Zaeem:17

Deployment status

Success

Tasks

Logs

Deployments

Events

Configuration and networking

Service auto scaling

Event history

Tags

Service configuration

Service ARN

arn:aws:ecs:us-west-2:880958245574:service/Task8-ECS-Cluster-Zaeem/Task8-ECS-Service-Zaeem

Task definition: revision

Task8-ECS-Task-Definition-Zaeem:17

Launch type

EC2

Scheduling strategy

REPLICA

Created by

arn:aws:iam::880958245574:user/Zaeem

Amazon ECS managed tags

Turned off

Propagate tags from

None

Availability Zone rebalancing

Turned on

CloudFormation stack

-

ECS Exec

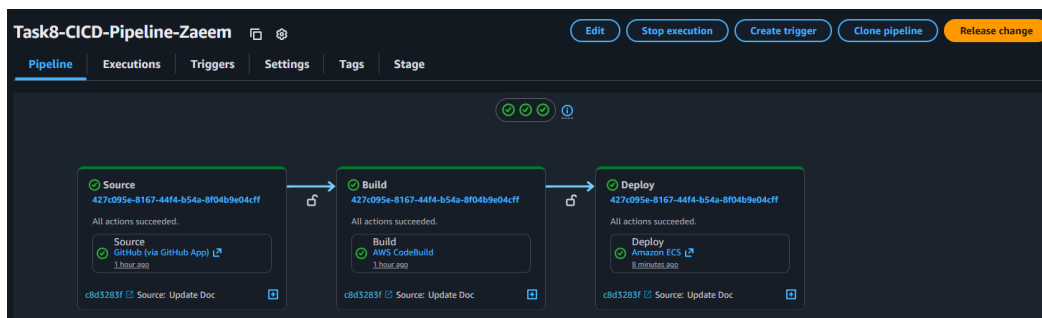
Turned off

Task placement strategy and constraints

Task8.5: Creating the CodePipeline

- Basic Pipeline Configuration:
 - Use the Build Pipeline button to create a new pipeline.
 - Category: Custom Pipeline
 - Pipeline Name: Task8-Pipeline-Zaeem
 - Execution mode: Queued
 - Service Role: Create new
- Source Stage:
 - Source Provider: GitHub (OAuth via app)
 - Connect to GitHub and allow the pipeline connection
- Build Stage:
 - Build Provider: Other (AWS CodeBuild)
 - Create a new project
 - Set Environment Variables:
 - REGION="us-west-2"
 - ACCOUNT_ID="504649076991"
 - REPO_NAME="zaeem/Task8"
 - IMAGE_TAG="latest"
 - Build Type: Single Build
 - Region: United States Oregon

- o Input Artifacts: SourceArtifact
- o Place the buildspec.yml file in the root of repo
- Deploy Stage (skipping the test stage):
 - o Deploy Provider: Amazon ECS
 - o Region: United States (Ohio)
 - o Input Artifacts: BuildArtifact
 - o Place the buildspec.yml file in the root of repo
 - o Cluster Name: Task8-ECS-Cluster-Zaeem
 - o Service Name: Task8-NodeJS-Zaeem-Service
 - o Deployment Time Out: 15 min
 - o Create Pipeline



Task8.6: Verification and Testing

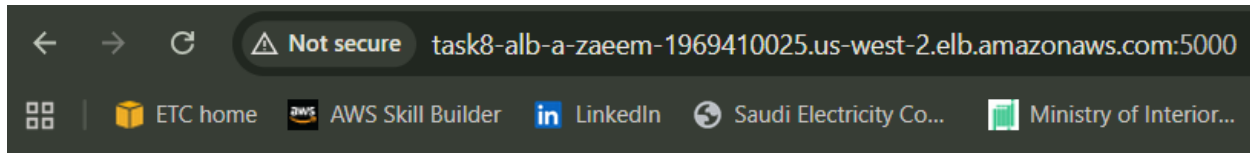
- Any commits to the GitHub code repository will trigger the pipeline



- The tasks deployed can be seen under the ECS Service

Service name	ARN	Status	Scheduling strategy	Task de...	Deployments and tasks
Task8-ECS-Service-Zaeem	arn:aws:ecs:us-v...	Active	REPLICA	EC2	Task8-ECS... 2/2 Ta

- The webpage of the NodeJS application can be accessed at the DNS of the ALB



Hello World!

Task8.7: Problems Faced

- The resources of the containers in the task definition should be less than that of the resources set in the Launch Template for the instances otherwise the containers will not be placed and will be stuck in the pending state.
- We must specify artifacts output in the buildspec.yml file to make sure that they are saved in the S3 bucket and passed onto the Deploy Stage otherwise we get an S3 permission error which is not related to permissions at all.
- The pipeline role must have the correct policies attached to carry out all the stages.
- We need to attach the proper permissions to be able to pass the Exec and Task roles to the deploy stage of the codepipeline, as well as the s3 permissions to read the artifact files.
- The name of the container in the appspec.yml file should match exactly the name of the container in the task definition otherwise the project is not deployed.