

Graded Project on Orchestration and Scaling

Objective: To automate the pipeline using ECR, EKS and Jenkins and deploy the app

GitHub Links: <https://github.com/SamDonald-A/StreamingApp>

Read file: <https://github.com/SamDonald-A/StreamingApp/blob/main/README.md>

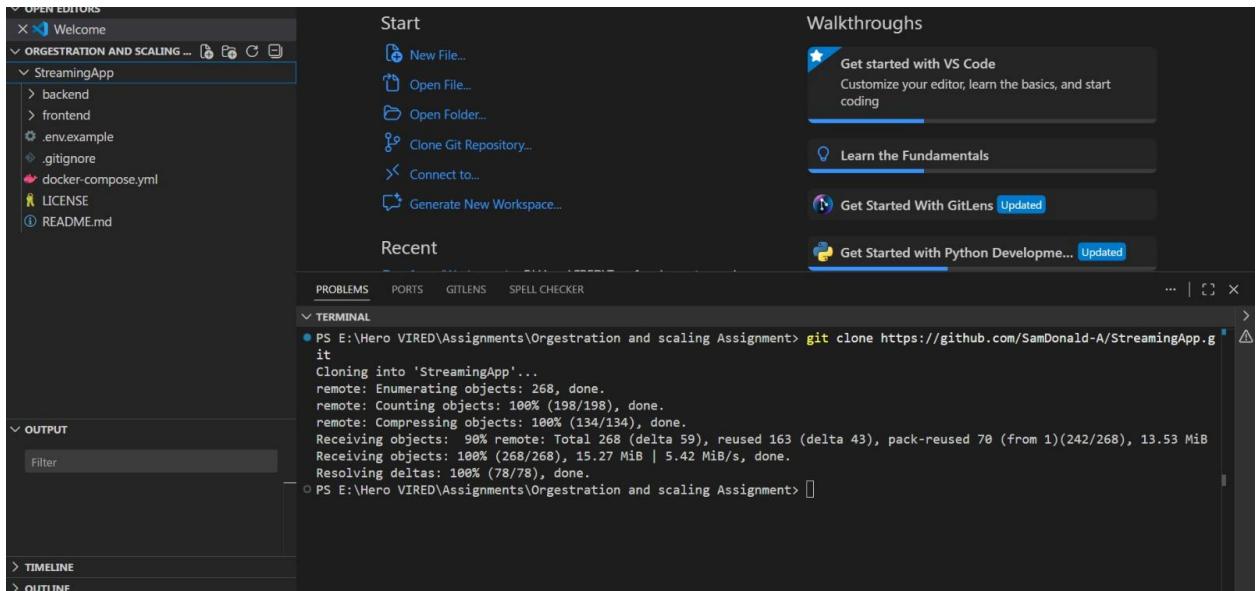
Step 1: Git and local code setup

- Fork the repo

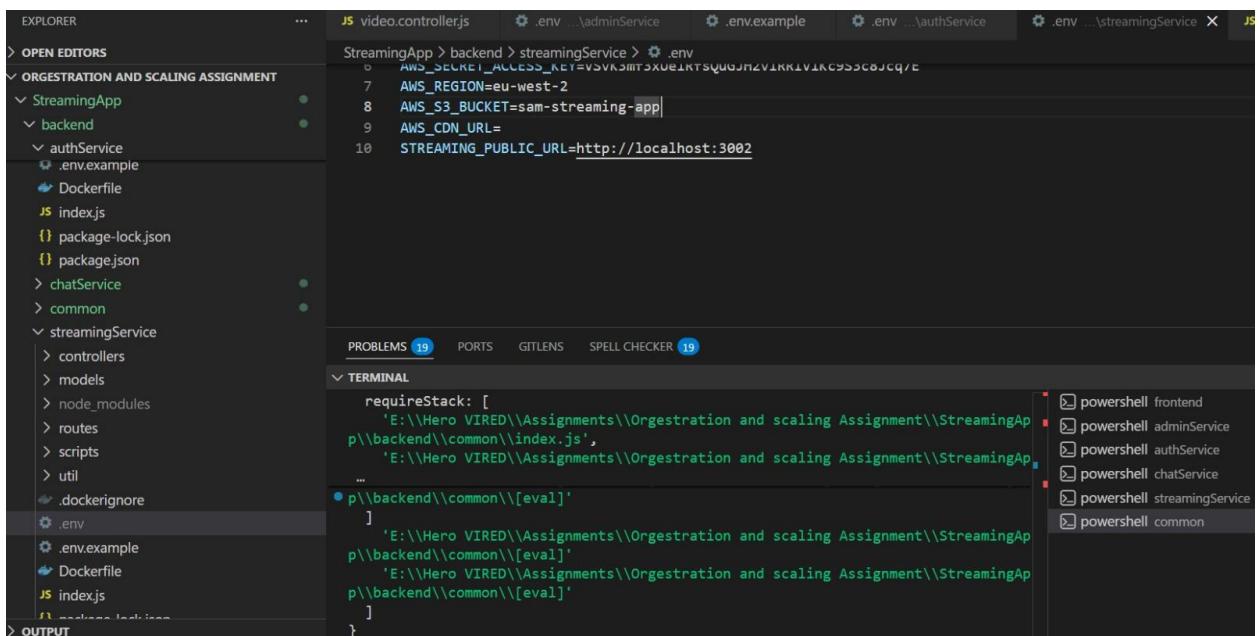
The screenshot shows a GitHub repository page for 'StreamingApp'. At the top, there's a search bar and navigation links for Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. Below that is the repository header with the name 'StreamingApp' (Public), a profile picture, and a note that it's forked from 'UnpredictablePrashant/StreamingApp'. There are buttons for Pin, Watch (0), and a dropdown menu. The main area shows a branch selector (main), a status message ('This branch is up to date with UnpredictablePrashant/StreamingApp:main .'), and a commit history. The commit history lists four commits by 'UnpredictablePrashant': 'WIP: adding the streaming codes' (f79b721 · 2 months ago), 'backend' (WIP: adding the streaming codes · 2 months ago), 'frontend' (WIP: adding the streaming codes · 2 months ago), '.env.example' (WIP: adding the streaming codes · 2 months ago), and '.gitignore' (Initial commit · 2 years ago). On the right side, there are sections for About, Collaborate, Read, Act, and Releases.

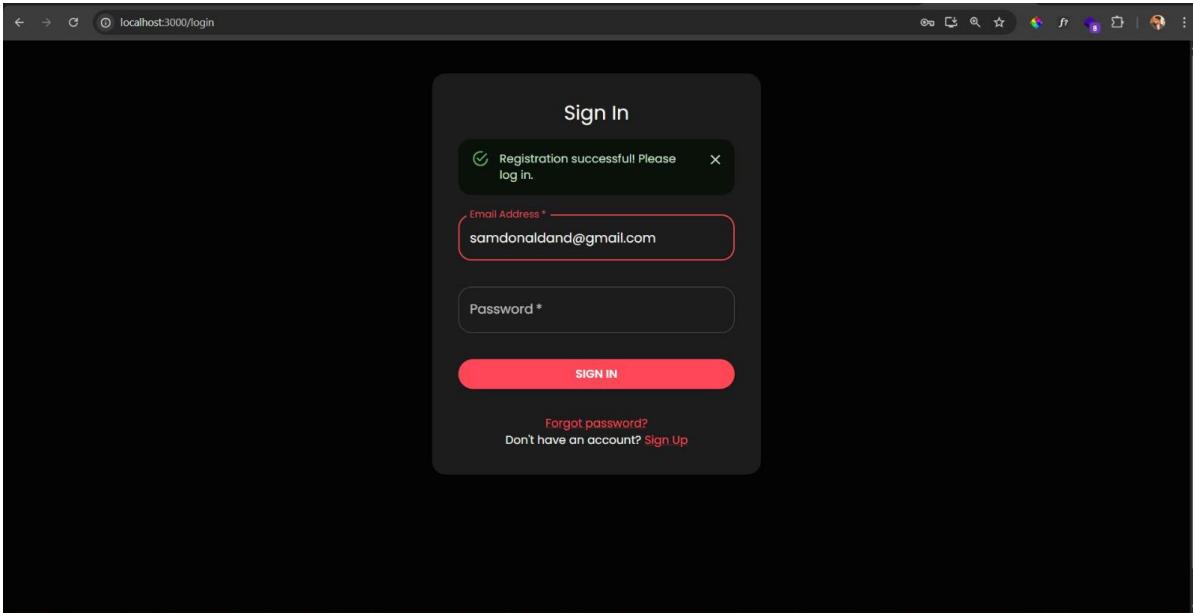
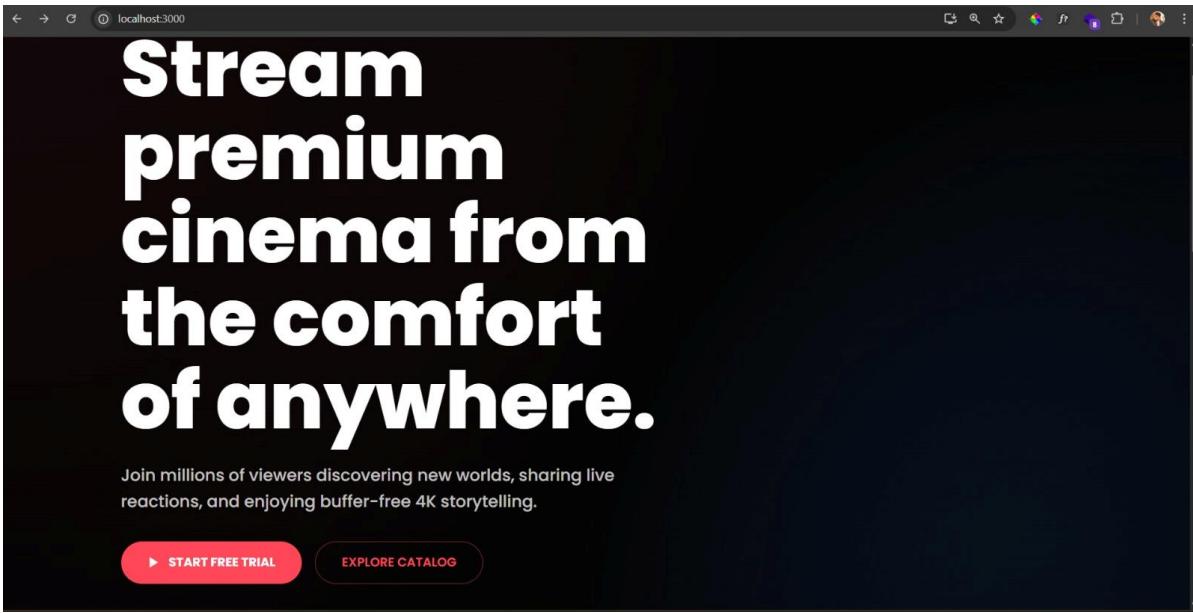
Commit	Message	Date
f79b721	WIP: adding the streaming codes	2 months ago
backend	WIP: adding the streaming codes	2 months ago
frontend	WIP: adding the streaming codes	2 months ago
.env.example	WIP: adding the streaming codes	2 months ago
.gitignore	Initial commit	2 years ago

- **Clone the repo in your local**



- **Create the .env in each service and in frontend and run the app for testing**





localhost:3000/register

Sign Up

Full Name *
Sam Donald

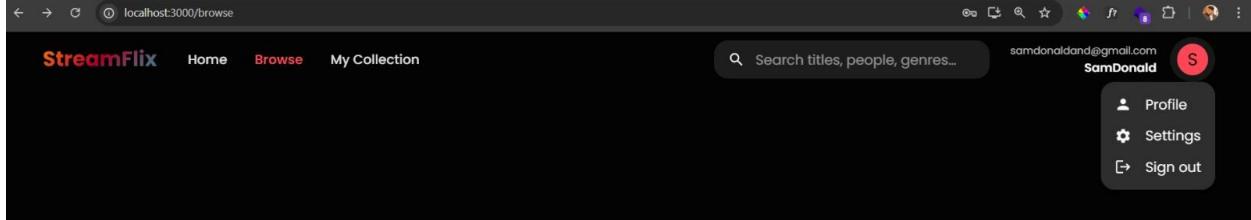
Email Address *
samdonaldand@gmail.com

Password *

Confirm Password *

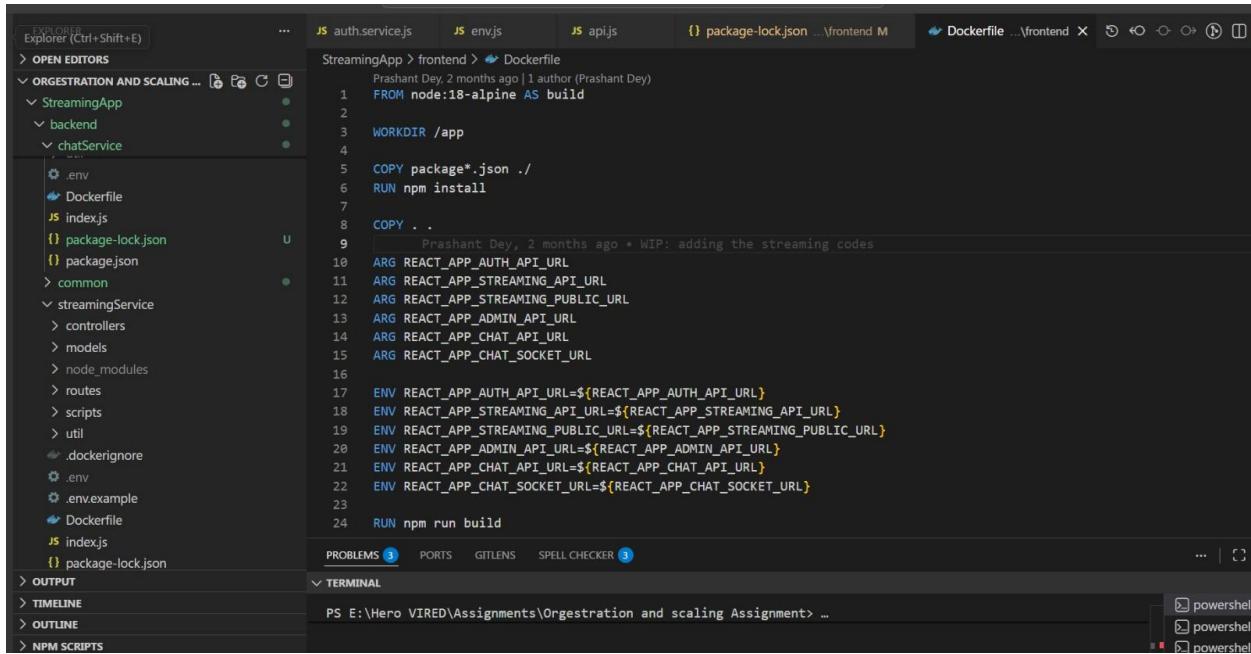
SIGN UP

Already have an account? [Sign In](#)



Step 2: Containerization - Docker setup

- Create Docker file in each service and in frontend



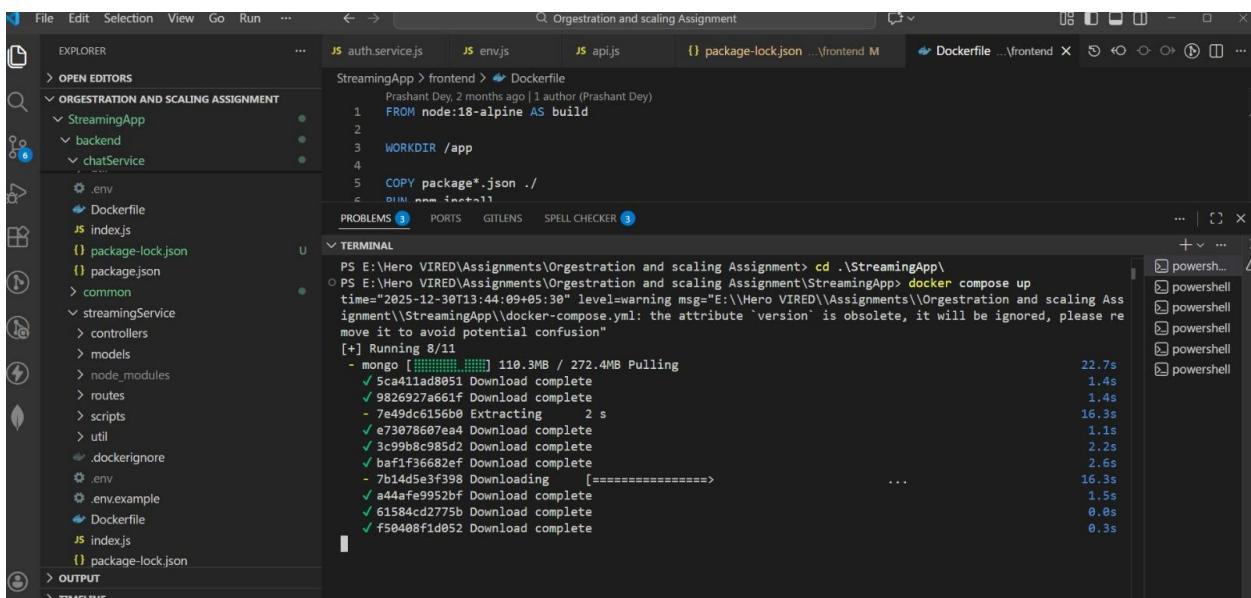
The screenshot shows the VS Code interface with the Dockerfile for the `StreamingApp` frontend service. The Dockerfile content is as follows:

```

FROM node:18-alpine AS build
WORKDIR /app
COPY package*.json .
RUN npm install
COPY . .
ARG REACT_APP_AUTH_API_URL
ARG REACT_APP_STREAMING_API_URL
ARG REACT_APP_STREAMING_PUBLIC_URL
ARG REACT_APP_ADMIN_API_URL
ARG REACT_APP_CHAT_API_URL
ARG REACT_APP_CHAT_SOCKET_URL
ENV REACT_APP_AUTH_API_URL=${REACT_APP_AUTH_API_URL}
ENV REACT_APP_STREAMING_API_URL=${REACT_APP_STREAMING_API_URL}
ENV REACT_APP_STREAMING_PUBLIC_URL=${REACT_APP_STREAMING_PUBLIC_URL}
ENV REACT_APP_ADMIN_API_URL=${REACT_APP_ADMIN_API_URL}
ENV REACT_APP_CHAT_API_URL=${REACT_APP_CHAT_API_URL}
ENV REACT_APP_CHAT_SOCKET_URL=${REACT_APP_CHAT_SOCKET_URL}
RUN npm run build

```

- Build image and run locally to check once again with containers using docker-compose.yaml



The screenshot shows the VS Code interface with the terminal output of running `docker compose up` for the `StreamingApp` service. The output shows the download and extraction of the MongoDB image.

```

PS E:\Hero VIRED\Assignments\Orchestration and scaling Assignment> cd .\StreamingApp
PS E:\Hero VIRED\Assignments\Orchestration and scaling Assignment\StreamingApp> docker compose up
time="2025-12-30T13:44:09+05:30" level=warning msg="E:\\Hero VIRED\\Assignments\\Orchestration and scaling Assignment\\StreamingApp\\docker-compose.yml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion"
[+] Running 8/11
- mongo [====] 118.3MB / 272.4MB Pulling
  ✓ 5ca411ad8851 Download complete
  ✓ 9826927a661f Download complete
  - 7e49dc6156b0 Extracting    2 s
  ✓ e73078607ea4 Download complete
  ✓ 3c99b8c985d2 Download complete
  ✓ baf1f36682ef Download complete
  - 7b14d5e3f398 Downloading    [=====]
  ✓ a44afe9952bf Download complete
  ✓ 61584cd2775b Download complete
  ✓ f50408f1d052 Download complete
22.7s
1.4s
1.4s
16.3s
1.1s
2.2s
2.6s
16.3s
1.5s
0.8s
0.3s

```

```

package.json
README.md
env.example
gitignore
docker-compose.yml
LICENSE
README.md

PUT
ELINE
LINE
SCRIPTS
    - SecretsUsedInArgOrEnv: Do not use ARG or ENV instructions for sensitive data (ENV "REACT_APP_AUTH_API_URL") (line 17)
)
PS E:\Hero VIRED\Assignments\Orchestration and scaling Assignment\StreamingApp> docker images
  IMAGE                   ID          DISK USAGE   CONTENT SIZE  EXTRA
  admin-service:latest   2102be6e4d44  201MB        0B
  auth-service:latest    94cbcd5f42ce  157MB        0B
  chat-service:latest    1fd8241d2916  190MB        0B
  front-end:latest       0bdd9f91fa2f   51.8MB       0B
  gcr.io/k8s-minikube/storage-provisioner:v5  6e38f40d628d  31.5MB       0B
  registry.k8s.io/coredns/coredns:v1.12.1  52546a367cc9  75MB        0B
  registry.k8s.io/etcd:3.6.4-0    5f1f5298c888  195MB        0B
  registry.k8s.io/kube-apiserver:v1.34.0  90550c43ad2b  88MB        0B
  registry.k8s.io/kube-controller-manager:v1.34.0  a0a7f2f2ec6d  74.9MB       0B
  registry.k8s.io/kube-proxy:v1.34.0   df0860106674  71.9MB       0B
  registry.k8s.io/kube-scheduler:v1.34.0  46169d968e92  52.8MB       0B
  registry.k8s.io/pause:13.10.1     cd073f4c5f6a   736KB        0B
  streaming-service:latest      3ae4926a7075  217MB        0B
  PS E:\Hero VIRED\Assignments\Orchestration and scaling Assignment\StreamingApp>

```

- Make sure your docker and minikube are running

	Name	Container ID	Image	Port(s)	CPU (%)	Memory usage...	Memory (%)	Disk read/write	Actions
<input type="checkbox"/>	minikube	8ca6c7cee415	k8s-minikube	55961:22	204.46%	614.6MB / 3GB	20.01%	701MB / 4.69G	
<input type="checkbox"/>	streamingapp	-	-	-	1.81%	373.3MB / 34.25G	6.39%	90.31MB / 9.7G	
<input type="checkbox"/>	mongo-1	87e2ccb8fe0	mongo:6	27017:27017	0.43%	143.6MB / 5.71Gi	2.46%	84.5MB / 9.7M	
<input type="checkbox"/>	streaming-1	f6f11ccc1f5a	streaminga	3002:3002	1.13%	60.11MB / 5.71Gi	1.03%	77.8KB / 4.1KE	
<input type="checkbox"/>	auth-1	2b610d20775b	streaminga	3001:3001	0%	50.46MB / 5.71Gi	0.86%	1.16MB / 4.1Ki	
<input type="checkbox"/>	admin-1	51b8abefb591	streaminga	3003:3003	0.1%	58.71MB / 5.71Gi	1%	229KB / 4.1KB	
<input type="checkbox"/>	chat-1	6a3e7e0637e7	streaminga	3004:3004	0.15%	50.18MB / 5.71Gi	0.86%	0B / 4.1KB	
<input type="checkbox"/>	frontend-1	2344049b0b61	streaminga	3000:80	0%	10.24MB / 5.71Gi	0.18%	4.35MB / 12.3i	

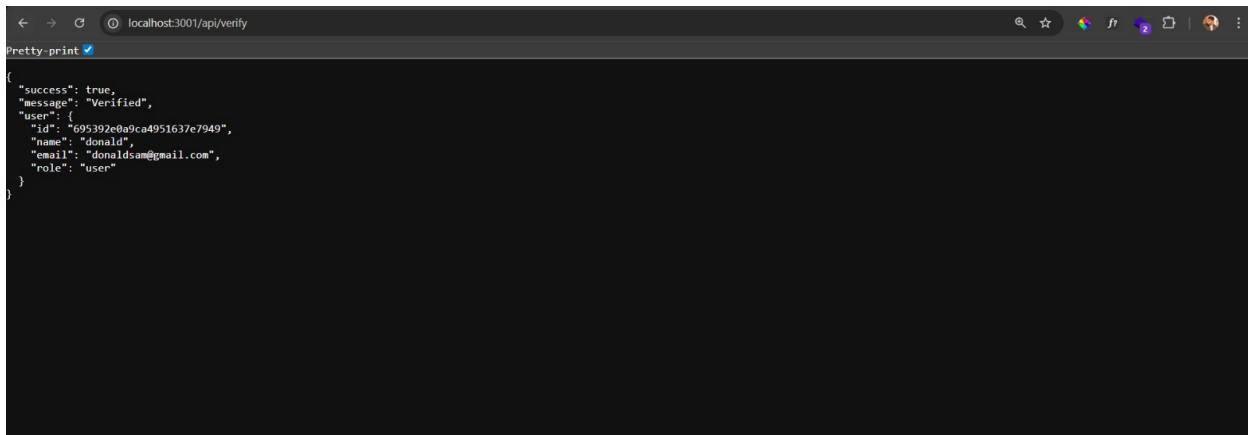
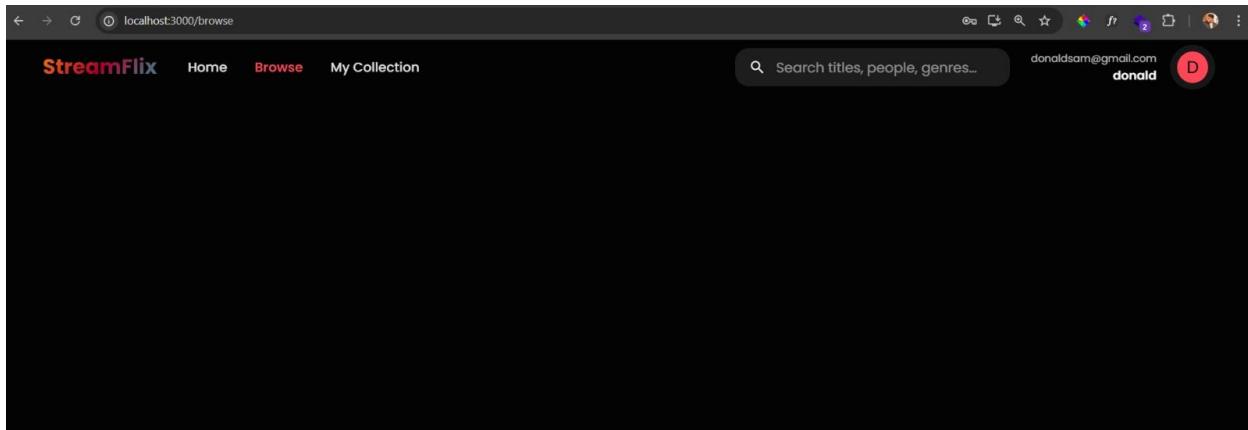
```

PS E:\Hero VIRED\Assignments\Orchestration and scaling Assignment> minikube start
  🎉 minikube v1.37.0 on Microsoft Windows 11 Home Single Language 10.0.26200.7462 Build 26200.7462
  🚀 Using the docker driver based on existing profile
  🔥 Starting "minikube" primary control-plane node in "minikube" cluster
  🚗 Pulling base image v0.0.48 ...
  ⚙️ Restarting existing docker container for "minikube" ...
  ! Failing to connect to https://registry.k8s.io/ from inside the minikube container
 💡 To pull new external images, you may need to configure a proxy: https://minikube.sigs.k8s.io/docs/reference/configuring/proxy/
  🛠️ Preparing Kubernetes v1.34.0 on Docker 28.4.0 ...
  🛡️ Verifying Kubernetes components...
    * Using image gcr.io/k8s-minikube/storage-provisioner:v5
  🌈 Enabled addons: default-storageclass, storage-provisioner
  🎉 Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
  PS E:\Hero VIRED\Assignments\Orchestration and scaling Assignment>

```

Ln 96, Col 1 Spaces: 2 UTF-8 CRLF {} JavaScript Finish Setup Go Live

- Now app is running via docker container



Step 3: Container orchestration – Kubernetes and Helm Setup

- Create k8s manifest files to run pods then trying with helm

```

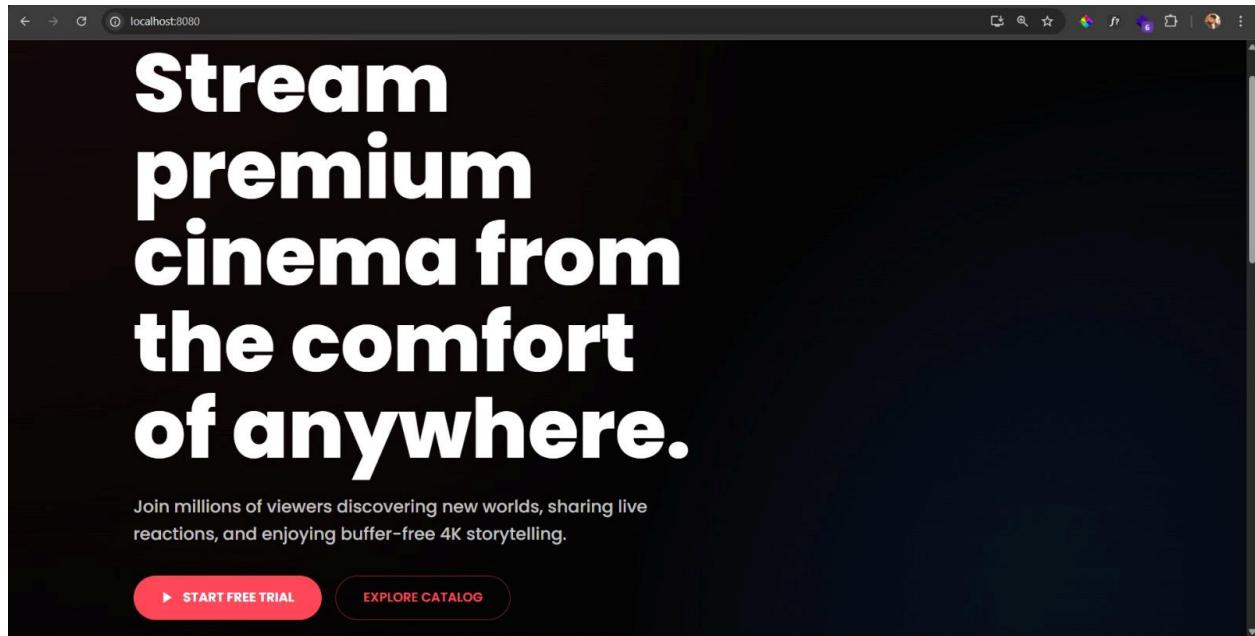
apiVersion: apps/v1
kind: Deployment
metadata:
  name: streaming
  namespace: streamingapp
spec:
  replicas: 1
  selector:
    matchLabels:
      app: streaming
  template:
    metadata:
      labels:
        app: streaming
    spec:
      containers:
        - name: streaming
          image: streaming-service:latest
          ports:
            - containerPort: 3002
          env:
            - name: PORT
              value: "3002"
            - name: MONGO_URI
              value: mongodb://mongo:27017/streamingapp
            - name: JWT_SECRET
              valueFrom:
                secretKeyRef:
                  name: streaming-secret

```

```
PS E:\Hero VIRED\Assignments\Orchestration and scaling Assignment\StreamingApp> kubectl get pods -n streamingapp
streaming-fd457b667-z2ck7  0/1  ContainerCreating   0          93s
PS E:\Hero VIRED\Assignments\Orchestration and scaling Assignment\StreamingApp> kubectl get svc -n streamingapp
● NAME           TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)        AGE
  admin          ClusterIP   10.196.125.40   <none>         3003/TCP     113s
  auth           ClusterIP   10.104.137.126  <none>         3001/TCP     113s
  chat            ClusterIP   10.184.182.33   <none>         3004/TCP     113s
  frontend        ClusterIP   10.96.7.247    <none>         80/TCP       113s
  mongo           ClusterIP   10.98.172.94   <none>         27017/TCP    113s
  streaming        ClusterIP   10.111.249.156  <none>         3002/TCP     113s
● PS E:\Hero VIRED\Assignments\Orchestration and scaling Assignment\StreamingApp> kubectl get ingress -n streamingapp
NAME          CLASS      HOSTS          ADDRESS        PORTS      AGE
streamingapp-ingress  nginx    streamingapp.local  192.168.49.2  80        2m10s
PS E:\Hero VIRED\Assignments\Orchestration and scaling Assignment\StreamingApp> kubectl get pods -n ingress-nginx
● NAME                  READY   STATUS    RESTARTS   AGE
  ingress-nginx-admission-create-ltlcc  0/1    Completed  0          14m
  ingress-nginx-admission-patch-crc8h  0/1    Completed  0          14m
  ingress-nginx-controller-9c49f96f-9wfsg 1/1    Running   0          14m
○ PS E:\Hero VIRED\Assignments\Orchestration and scaling Assignment\StreamingApp>
```

- Port forwarding for the local K8s deployment

```
U ingress-nginx-admission-patch-crc8h      0/1    Completed   0      23m
U ingress-nginx-controller-9cc49f96f-9wfsg  1/1    Running     0      23m
PS E:\Hero\VIRED\Assignments\Registration and scaling Assignment\StreamingApp> kubectl port-forward svc/frontend 8080:80 -n streamingapp
Forwarding from 127.0.0.1:8080 -> 80
Forwarding from [::1]:8080 -> 80
```



- Create EKS cluster and set context

```

PROBLEMS 3 PORTS GITLENS SPELL CHECKER 3
TERMINAL
PS E:\Hero VIRED\Assignments\Orgestration and scaling Assignment> eksctl create cluster --name sam-cluster-streaming --region eu-west-2 --version 1.33 --nodegroup-name ng-1 --node-type t3.medium --nodes 2 --nodes-min 2 --nodes-max 4 --with-oidc --managed
2025-12-29 14:03:29 [i] eksctl version 0.217.0
2025-12-29 14:03:29 [i] using region eu-west-2
2025-12-29 14:03:30 [i] setting availability zones to [eu-west-2a eu-west-2b eu-west-2c]
2025-12-29 14:03:30 [i] subnets for eu-west-2a - public:192.168.0.0/19 private:192.168.96.0/19
2025-12-29 14:03:30 [i] subnets for eu-west-2b - public:192.168.32.0/19 private:192.168.128.0/19
2025-12-29 14:03:30 [i] subnets for eu-west-2c - public:192.168.64.0/19 private:192.168.160.0/19
2025-12-29 14:03:30 [i] nodegroup "ng-1" will use "" [AmazonLinux2023/1.33]
2025-12-29 14:03:30 [!] Auto Mode will be enabled by default in an upcoming release of eksctl. This means managed node groups and managed networking add-ons will no longer be created by default. To maintain current behavior, explicitly set 'autoModeConfig.enabled: false' in your cluster configuration. Learn more: https://eksctl.io/usage/auto-mode/
2025-12-29 14:03:30 [i] using Kubernetes version 1.33
2025-12-29 14:03:30 [i] creating EKS cluster "sam-cluster-streaming" in "eu-west-2" region with managed nodes
2025-12-29 14:03:30 [i] will create 2 separate CloudFormation stacks for cluster itself and the initial man

```



```

2025-12-29 14:18:35 [i] nodegroup "ng-1" has 2 node(s)
2025-12-29 14:18:35 [i] node "ip-192-168-29-96.eu-west-2.compute.internal" is ready
2025-12-29 14:18:35 [i] node "ip-192-168-42-249.eu-west-2.compute.internal" is ready
2025-12-29 14:18:35 [✓] created 1 managed nodegroup(s) in cluster "sam-cluster-streaming"
2025-12-29 14:18:37 [i] creating addon: metrics-server
2025-12-29 14:18:38 [i] successfully created addon: metrics-server
2025-12-29 14:18:41 [i] kubectl command should work with "C:\\\\Users\\\\Sam Donald\\\\.kube\\\\config", try 'kubectl get nodes'
2025-12-29 14:18:41 [✓] EKS cluster "sam-cluster-streaming" in "eu-west-2" region is ready

```

- Setting-up the context to EKS

```

PROBLEMS 7 PORTS GITLENS SPELL CHECKER 7
TERMINAL
PS E:\Hero VIRED\Assignments\Orgestration and scaling Assignment\StreamingApp> eksctl version
0.217.0
PS E:\Hero VIRED\Assignments\Orgestration and scaling Assignment\StreamingApp> aws sts get-caller-identity
{
  "UserId": "AIDA6GBMCU7ZDOVXH2QLP",
  "Account": "975050024946",
  "Arn": "arn:aws:iam::975050024946:user/samdonaldand@gmail.com"
}

PS E:\Hero VIRED\Assignments\Orgestration and scaling Assignment\StreamingApp> aws eks update-kubeconfig --region eu-west-2 --name sam-cluster-streaming
Added new context arn:aws:eks:eu-west-2:975050024946:cluster/sam-cluster-streaming to C:/Users/Sam Donald/.kube/config
PS E:\Hero VIRED\Assignments\Orgestration and scaling Assignment\StreamingApp> kubectl config get-contexts
CURRENT NAME AUTHINFO NAMESPACE
* arn:aws:eks:eu-west-2:975050024946:cluster/sam-cluster-streaming arn:aws:eks:eu-west-2:975050024946:cluuster/sam-cluster-streaming docker-desktop docker-desktop
minikube minikube default
minikube minikube
samdonaldand@gmail.com@sam-cluster-streaming.eu-west-2.eksctl.io sam-cluster-streaming.eu-west-2.eksctl.io
samdonaldand@gmail.com@sam-cluster-streaming.eu-west-2.eksctl.io samdonaldand@gmail.com@sam-cluster-streaming.eu-west-2.eksctl.io

```

Step 4: Setup ECR

- Login to ECR

```
PS E:\Hero VIRED\Assignments\Orchestration and scaling Assignment\StreamingApp> aws ecr get-login-password --region eu-west-2 | docker login --username AWS --password-stdin 975050024946.dkr.ecr.eu-west-2.amazonaws.com
Login Succeeded
PS E:\Hero VIRED\Assignments\Orchestration and scaling Assignment\StreamingApp>
```

The terminal shows the command `aws ecr get-login-password --region eu-west-2 | docker login --username AWS --password-stdin 975050024946.dkr.ecr.eu-west-2.amazonaws.com` being run, followed by the message "Login Succeeded". The status bar at the bottom indicates "In 22, Col 1 Spaces: 2 UTF-8 CRLF {} YAML Finish Setup Go Live".

- Create repository and tag local images to the respected repo

The screenshot shows a code editor interface with two terminal panes. The left pane displays the Kubernetes and Helm configuration files for the StreamingApp. The right pane shows the AWS CLI commands being run to create ECR repositories and tag Docker images.

TERMINAL (Top):

```
PS E:\Hero VIRED\Assignments\Orchestration and scaling Assignment\StreamingApp> aws ecr create-repository --repository-name streaming --region eu-west-2
{
  "repository": {
    "repositoryArn": "arn:aws:ecr:eu-west-2:975050024946:repository/streaming",
    "registryId": "975050024946",
    "repositoryName": "streaming",
    "repositoryUri": "975050024946.dkr.ecr.eu-west-2.amazonaws.com/streaming",
    "createdAt": "2025-12-31T00:35:19.152000+05:30",
    "imageTagMutability": "MUTABLE",
    "imageScanningConfiguration": {
      "scanOnPush": false
    },
    "encryptionConfiguration": {
      "encryptionType": "AES256"
    }
  }
}

PS E:\Hero VIRED\Assignments\Orchestration and scaling Assignment\StreamingApp> aws ecr create-repository --repository-name frontend --region eu-west-2
{
  "repository": {
    "repositoryArn": "arn:aws:ecr:eu-west-2:975050024946:repository/frontend",
    "registryId": "975050024946",
    "repositoryName": "frontend",
    "repositoryUri": "975050024946.dkr.ecr.eu-west-2.amazonaws.com/frontend",
    "createdAt": "2025-12-31T00:35:48.059000+05:30",
    "imageTagMutability": "MUTABLE",
  }
}
```

TERMINAL (Bottom):

```
PS E:\Hero VIRED\Assignments\Orchestration and scaling Assignment\StreamingApp> docker tag streamingapp-admin:latest 975050024946.dkr.ecr.eu-west-2.amazonaws.com/admin:latest
PS E:\Hero VIRED\Assignments\Orchestration and scaling Assignment\StreamingApp> docker tag streamingapp-auth:latest 975050024946.dkr.ecr.eu-west-2.amazonaws.com/auth:latest
PS E:\Hero VIRED\Assignments\Orchestration and scaling Assignment\StreamingApp> docker tag streamingapp-chat:latest 975050024946.dkr.ecr.eu-west-2.amazonaws.com/chat:latest
PS E:\Hero VIRED\Assignments\Orchestration and scaling Assignment\StreamingApp> docker tag streamingapp-streaming:latest 975050024946.dkr.ecr.eu-west-2.amazonaws.com/streaming:latest
PS E:\Hero VIRED\Assignments\Orchestration and scaling Assignment\StreamingApp> docker tag streamingapp-frontend:latest 975050024946.dkr.ecr.eu-west-2.amazonaws.com/frontend:latest
PS E:\Hero VIRED\Assignments\Orchestration and scaling Assignment\StreamingApp>
```

The bottom terminal shows the results of the Docker tag command, listing the local images and their corresponding ECR URLs. The status bar at the bottom indicates "In 22, Col 1 Spaces: 2 UTF-8 CRLF {} YAML Finish Setup Go Live".

- Then push the images to ECR

```

st 975050024946.dkr.ecr.eu-west-2.amazonaws.com/admin:latest
PS E:\Hero VIRED\Assignments\Orchestration and scaling Assignment\StreamingApp> docker tag streamingapp-auth:latest
PS E:\Hero VIRED\Assignments\Orchestration and scaling Assignment\StreamingApp> docker tag streamingapp-chat:latest
t 975050024946.dkr.ecr.eu-west-2.amazonaws.com/chat:latest ...
atest 975050024946.dkr.ecr.eu-west-2.amazonaws.com/frontend:latest
ates 975050024946.dkr.ecr.eu-west-2.amazonaws.com/frontend:latest
PS E:\Hero VIRED\Assignments\Orchestration and scaling Assignment\StreamingApp> docker push 975050024946.dkr.ecr.e
u-west-2.amazonaws.com/admin:latest
The push refers to repository [975050024946.dkr.ecr.eu-west-2.amazonaws.com/admin]
25ff2da83641: Pushed
668b0e31415c: Pushing [=====] 3.146MB/5.113MB
d3c88edb0654: Pushing [=====] 3.146MB/10.34MB
f18232174bc9: Pushing [=====] 3.642MB/3.642MB
dfec02fd755: Pushed
25aa8fccccb7a: Pushed
dd71dde834b5: Pushing [=====] 4.194MB/40.81MB
1e5a4c89ce5: Pushing [=====] 1.261MB/1.261MB
40922a90ac1b: Pushed
bee8f154fc8d: Pushing [=====] 3.146MB/4.058MB

```

- Replace image names with reference ID from AWS ECR image Repository

```

backend:
  services:
    auth:
      name: auth
      image: 975050024946.dkr.ecr.eu-west-2.amazonaws.com/auth:latest
      port: 3001
    streaming:
      name: streaming
      image: 975050024946.dkr.ecr.eu-west-2.amazonaws.com/streaming:latest
      port: 3002
    admin:
      name: admin
      image: 975050024946.dkr.ecr.eu-west-2.amazonaws.com/admin:latest
      port: 3003
    chat:
      name: chat
      image: 975050024946.dkr.ecr.eu-west-2.amazonaws.com/chat:latest
      port: 3004
  frontend:
    image: 975050024946.dkr.ecr.eu-west-2.amazonaws.com/frontend:latest

```

```

PS E:\Hero VIRED\Assignments\Orchestration and scaling Assignment\StreamingApp> docker push 975050024946.dkr.ecr.e
u-west-2.amazonaws.com/frontend:latest
34a64644ab756: Pushed
39c2ddfd6910: Pushing [=====] 15.52MB/15.52MB
d7e507024086: Pushed

```

- Deploy code via helm template – Create helm files and helm upgrade your project

```

nx ingress-nginx/ingress-nginx \ --namespace ingress-nginx \ --create-namespace \ --set controller.service.type=LoadBalancer
PS E:\Hero\VIRED\Assignments\Orchestration and scaling Assignment\StreamingApp> helm upgrade --install ingress-nginx-ingress-nginx-ingress-nginx --namespace ingress-nginx --create-namespace --set controller.service.type=LoadBalancer
Release "ingress-nginx" does not exist. Installing it now.
level=warn msg="unable to find exact version; falling back to closest available version" chart=ingress-nginx requested="" selected="4.14.1"
NAME: ingress-nginx
LAST DEPLOYED: Wed Dec 31 01:11:39 2025
NAMESPACE: ingress-nginx
STATUS: deployed
REVISION: 1
DESCRIPTION: Install complete
TEST SUITE: None
NOTES:
The ingress-nginx controller has been installed.
It may take a few minutes for the load balancer IP to be available.
You can watch the status by running 'kubectl get service --namespace ingress-nginx ingress-nginx-controller --output wide --watch'.
```

An example Ingress that makes use of the controller:

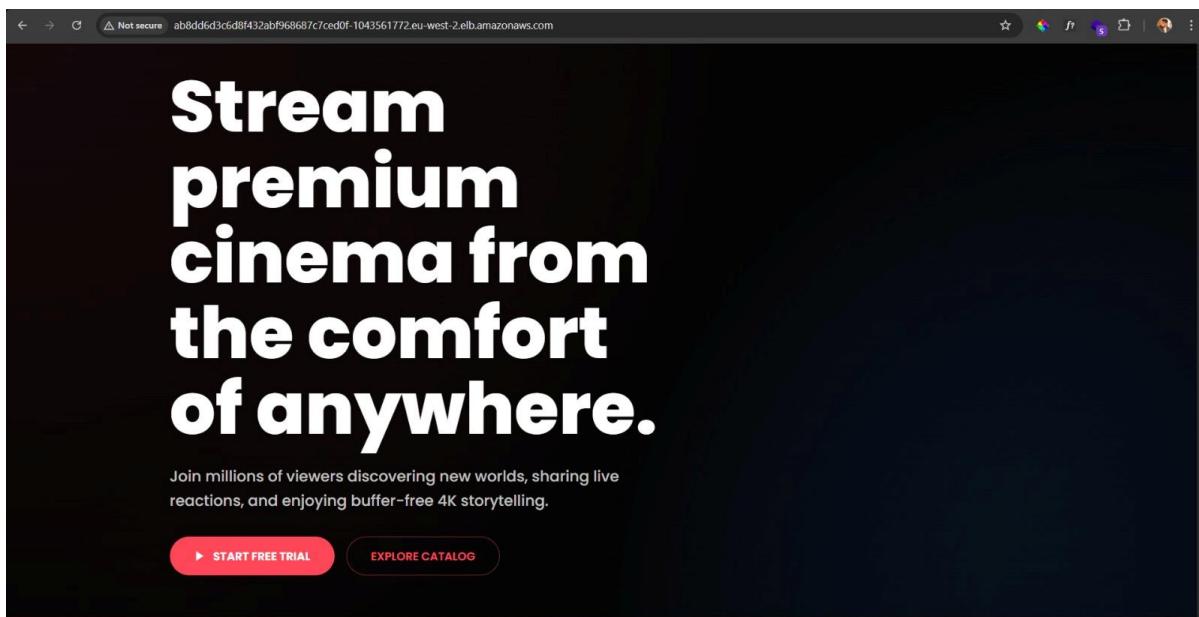
```

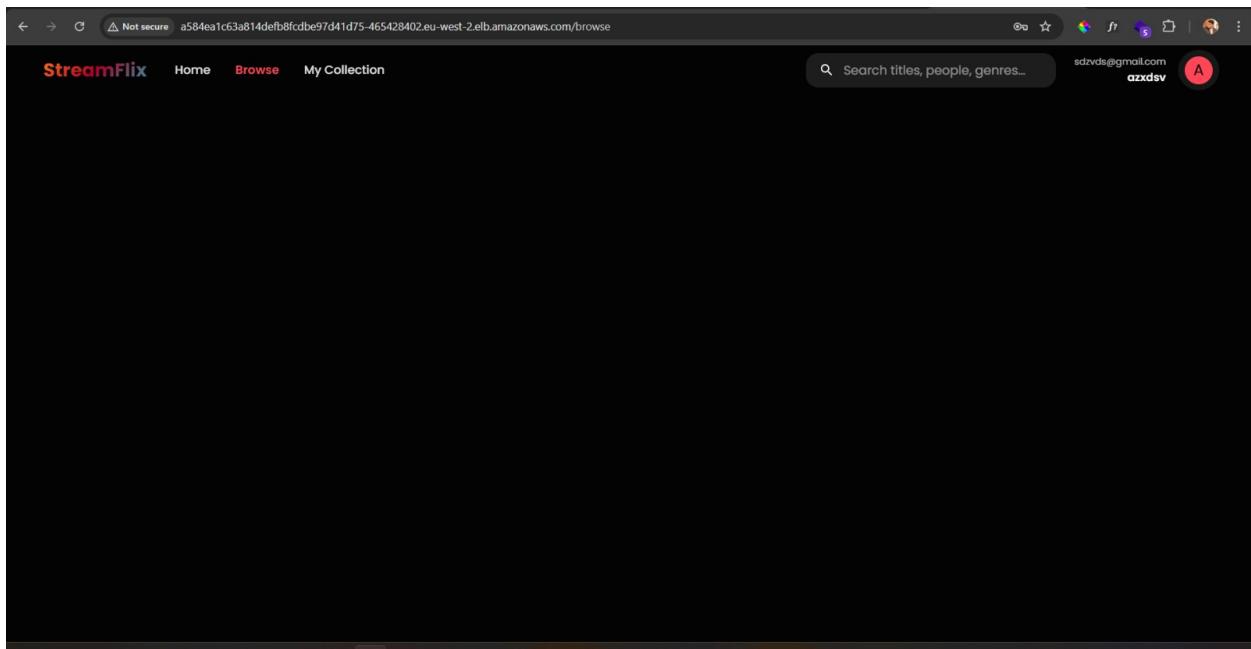
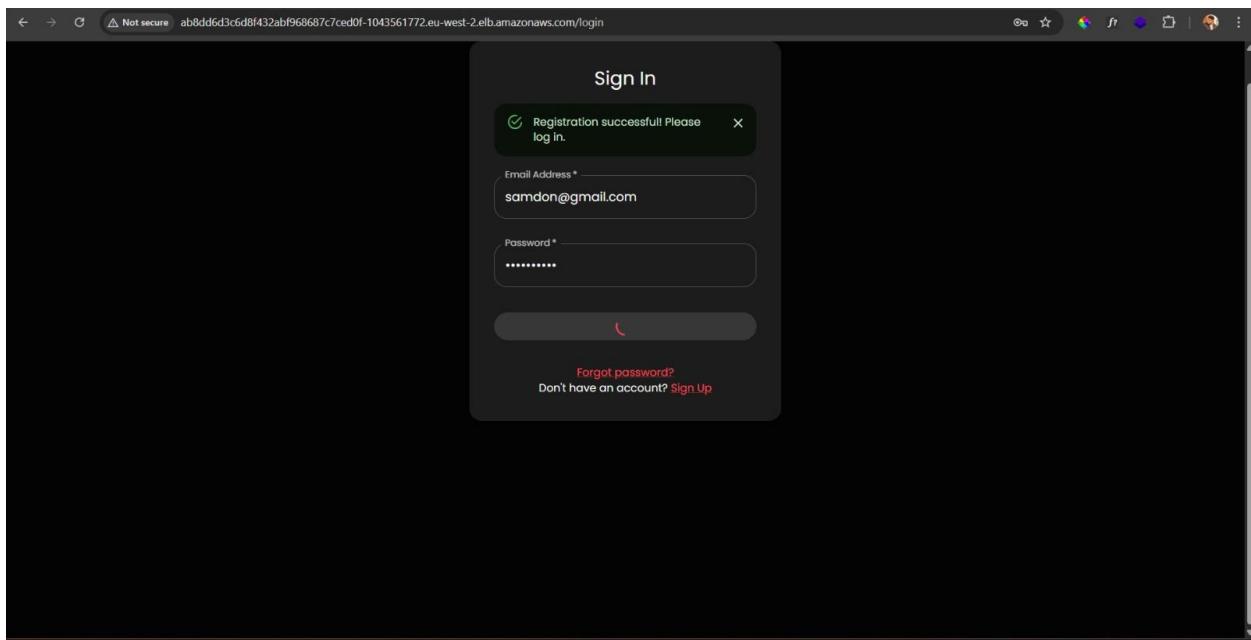
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: example
  namespace: foo
spec:
  ingressClassName: nginx
  rules:
    - host: www.example.com
      http:
        paths:
          - pathType: Prefix
```

- Check ingress created with load balancer

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP
ingress-nginx-controller	LoadBalancer	10.100.223.208	ab8dd6d3c6d8f432abf968687c7ced0f-1043561772.eu-west-2.elb.amazonaws.com
ingress-nginx-controller-admission	ClusterIP	10.100.183.195	<none>
	443/TCP		101s

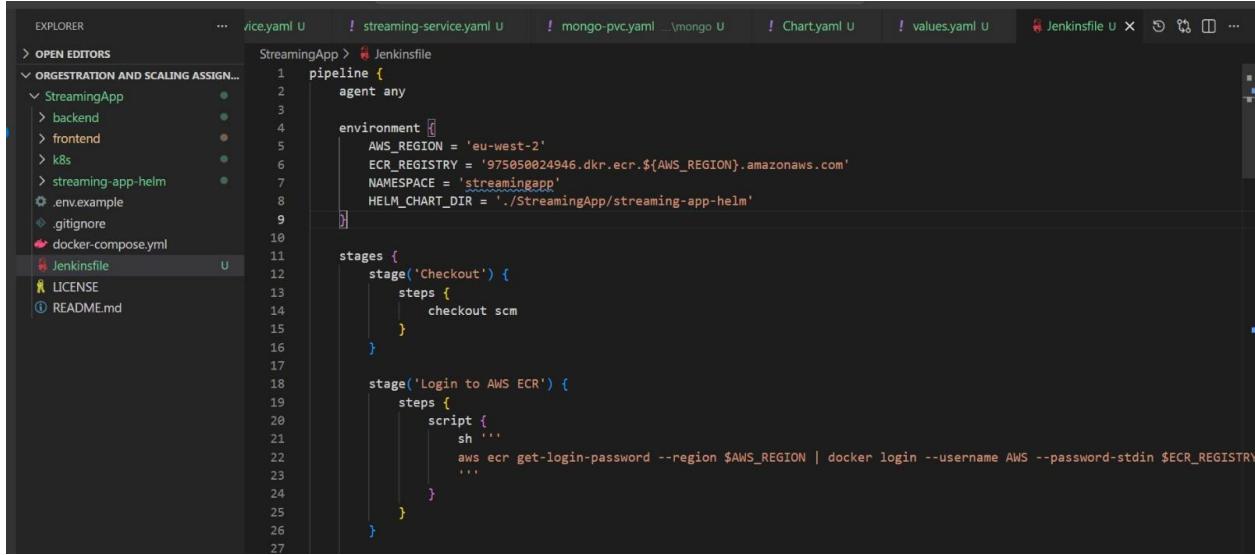
- Use the Loadbalancer and open the app in the browser





Step 5: Jenkinsfile and Jenkin setup

- Create Jenkins file



The screenshot shows a code editor with a Jenkinsfile open. The Jenkinsfile defines a pipeline with stages for checkout, logging into AWS ECR, and running a script to get login credentials.

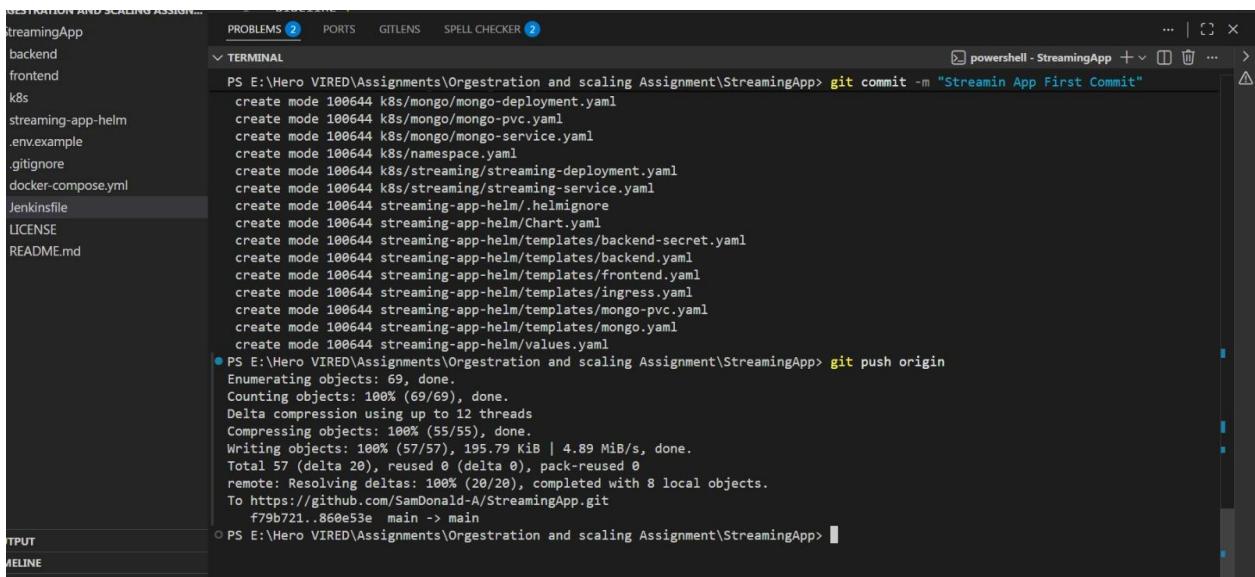
```
pipeline {
    agent any

    environment {
        AWS_REGION = 'eu-west-2'
        ECR_REGISTRY = '975050024946.dkr.ecr.${AWS_REGION}.amazonaws.com'
        NAMESPACE = 'streamingapp'
        HELM_CHART_DIR = './StreamingApp/streaming-app-helm'
    }

    stages {
        stage('Checkout') {
            steps {
                checkout scm
            }
        }

        stage('Login to AWS ECR') {
            steps {
                script {
                    sh '''
                        aws ecr get-login-password --region $AWS_REGION | docker login --username AWS --password-stdin $ECR_REGISTRY
                    '''
                }
            }
        }
    }
}
```

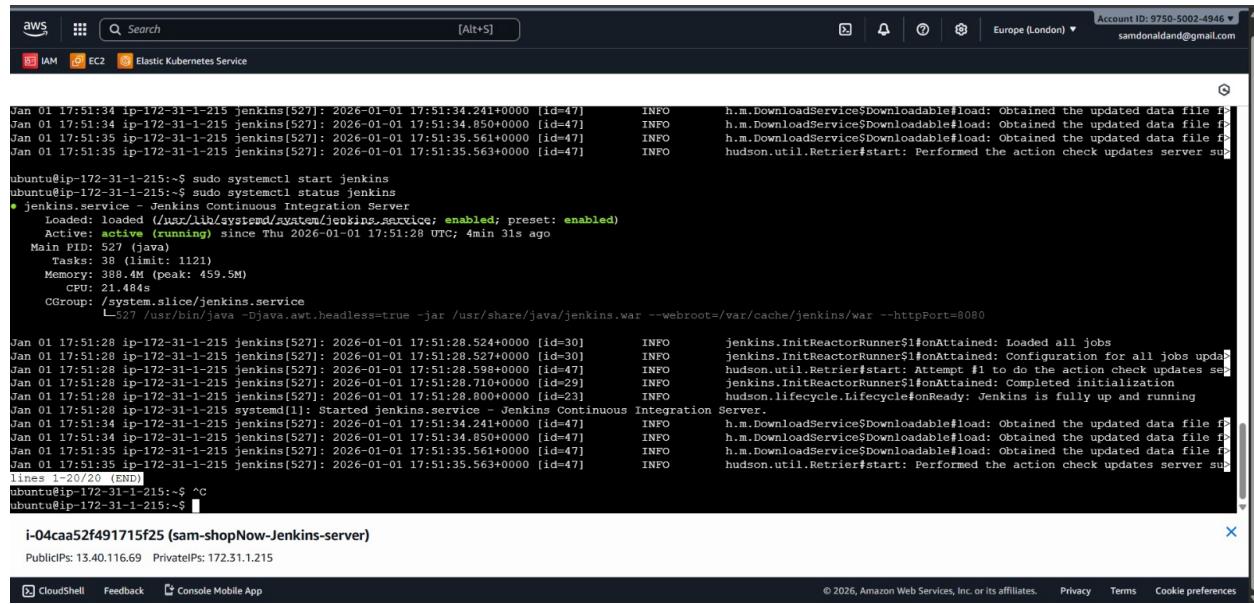
- And Push the complete code to the Git



The screenshot shows a terminal window in a code editor. It displays the output of a git commit command followed by a git push origin command. The commit message is "Streamin App First Commit". The push command shows the progress of pushing local changes to a GitHub repository.

```
PS E:\Hero\VIRED\Assignments\Orgestration and scaling Assignment\StreamingApp> git commit -m "Streamin App First Commit"
create mode 100644 k8s/mongo/mongo-deployment.yaml
create mode 100644 k8s/mongo/mongo-pvc.yaml
create mode 100644 k8s/mongo/mongo-service.yaml
create mode 100644 k8s/namespace.yaml
create mode 100644 k8s/streaming/streaming-deployment.yaml
create mode 100644 k8s/streaming/streaming-service.yaml
create mode 100644 streaming-app-helm/.helmignore
create mode 100644 streaming-app-helm/Chart.yaml
create mode 100644 streaming-app-helm/templates/backend-secret.yaml
create mode 100644 streaming-app-helm/templates/backend.yaml
create mode 100644 streaming-app-helm/templates/frontend.yaml
create mode 100644 streaming-app-helm/templates/ingress.yaml
create mode 100644 streaming-app-helm/templates/mongo-pvc.yaml
create mode 100644 streaming-app-helm/templates/mongo.yaml
create mode 100644 streaming-app-helm/values.yaml
PS E:\Hero\VIRED\Assignments\Orgestration and scaling Assignment\StreamingApp> git push origin
Enumerating objects: 69, done.
Counting objects: 100% (69/69), done.
Delta compression using up to 12 threads
Compressing objects: 100% (55/55), done.
Writing objects: 100% (57/57), 195.79 KiB | 4.89 MiB/s, done.
Total 57 (delta 20), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (20/20), completed with 8 local objects.
To https://github.com/SamDonald-A/StreamingApp.git
    f79b721..860e53e  main -> main
PS E:\Hero\VIRED\Assignments\Orgestration and scaling Assignment\StreamingApp>
```

- Create Jenkins server in the EC2 and open it in the browser via IP address



```

aws | ⚡ AM | EC2 | Elastic Kubernetes Service | Search | [Alt+S] | Europe (London) | Account ID: 9750-5002-4946 | samdonaldand@gmail.com

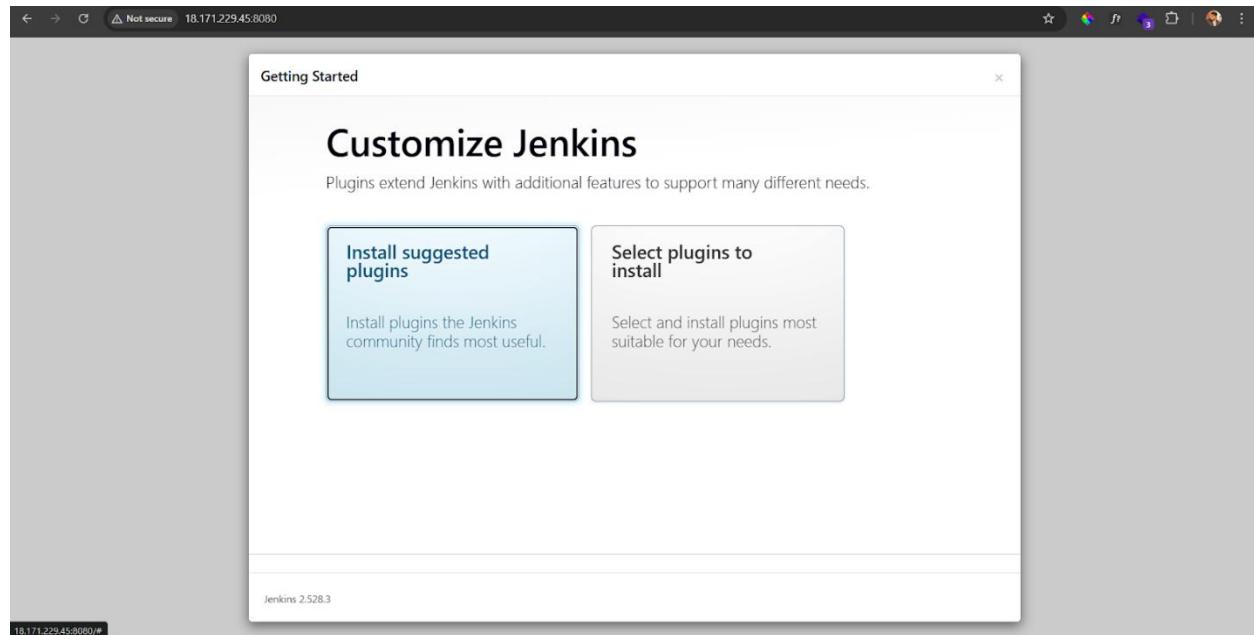
Jan 01 17:51:34 ip-172-31-1-215 jenkins[527]: 2026-01-01 17:51:34.241+0000 [id=47] INFO h.m.DownloadService$Downloadable$load: Obtained the updated data file fo...
Jan 01 17:51:34 ip-172-31-1-215 jenkins[527]: 2026-01-01 17:51:34.850+0000 [id=47] INFO h.m.DownloadService$Downloadable$load: Obtained the updated data file fo...
Jan 01 17:51:35 ip-172-31-1-215 jenkins[527]: 2026-01-01 17:51:35.561+0000 [id=47] INFO h.m.DownloadService$Downloadable$load: Obtained the updated data file fo...
Jan 01 17:51:35 ip-172-31-1-215 jenkins[527]: 2026-01-01 17:51:35.563+0000 [id=47] INFO hudson.util.Retrier$start: Performed the action check updates server su...

ubuntu@ip-172-31-1-215:~$ sudo systemctl start jenkins
ubuntu@ip-172-31-1-215:~$ sudo systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
   Loaded: loaded (/lib/systemd/system/jenkins.service; enabled; preset: enabled)
     Active: active (running) since Thu 2026-01-01 17:51:28 UTC; 4min 31s ago
       Main PID: 527 (java)
          Tasks: 38 (limit: 1121)
        Memory: 388.4M (peak: 459.5M)
           CPU: 21.484s
          CGroup: /system.slice/jenkins.service
                  └─527 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war --httpPort=8080

Jan 01 17:51:28 ip-172-31-1-215 jenkins[527]: 2026-01-01 17:51:28.524+0000 [id=30] INFO jenkins.InitReactorRunner$1#onAttained: Loaded all jobs
Jan 01 17:51:28 ip-172-31-1-215 jenkins[527]: 2026-01-01 17:51:28.527+0000 [id=30] INFO jenkins.InitReactorRunner$1#onAttained: Configuration for all jobs upda...
Jan 01 17:51:28 ip-172-31-1-215 jenkins[527]: 2026-01-01 17:51:28.598+0000 [id=47] INFO hudson.util.Retrier$start: Attempt #1 to do the action check updates se...
Jan 01 17:51:28 ip-172-31-1-215 jenkins[527]: 2026-01-01 17:51:28.710+0000 [id=29] INFO jenkins.InitReactorRunner$1#onAttained: Completed initialization
Jan 01 17:51:28 ip-172-31-1-215 jenkins[527]: 2026-01-01 17:51:28.800+0000 [id=23] INFO hudson.lifecycle.Lifecycle$onReady: Jenkins is fully up and running
Jan 01 17:51:28 ip-172-31-1-215 jenkins[527]: 2026-01-01 17:51:28.803+0000 [id=23] INFO hudson.lifecycle.Lifecycle$onReady: Jenkins is fully up and running
Jan 01 17:51:34 ip-172-31-1-215 jenkins[527]: 2026-01-01 17:51:34.241+0000 [id=47] INFO h.m.DownloadService$Downloadable$load: Obtained the updated data file fo...
Jan 01 17:51:34 ip-172-31-1-215 jenkins[527]: 2026-01-01 17:51:34.850+0000 [id=47] INFO h.m.DownloadService$Downloadable$load: Obtained the updated data file fo...
Jan 01 17:51:35 ip-172-31-1-215 jenkins[527]: 2026-01-01 17:51:35.561+0000 [id=47] INFO h.m.DownloadService$Downloadable$load: Obtained the updated data file fo...
Jan 01 17:51:35 ip-172-31-1-215 jenkins[527]: 2026-01-01 17:51:35.563+0000 [id=47] INFO hudson.util.Retrier$start: Performed the action check updates server su...
lines 1-20/20 (END)
ubuntu@ip-172-31-1-215:~$ ^C
ubuntu@ip-172-31-1-215:~$ [  ]
```

i-04caa52f491715f25 (sam-shopNow-Jenkins-server)
PublicIPs: 13.40.116.69 PrivateIPs: 172.31.1.215

CloudShell Feedback Console Mobile App © 2026, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences



Not secure 18.171.229.45:8080

Getting Started

Getting Started

✓ Folders	✓ OWASP Markup Formatter	Build Timeout	Credentials Binding
Timestamper	Workspace Cleanup	Ant	Gradle
Pipeline	GitHub Branch Source	Pipeline: GitHub Groovy Libraries	Pipeline Graph View
Git	SSH Build Agents	Matrix Authorization Strategy	LDAP
Email Extension	Mailer	Dark Theme	

commons-lang3 v3.x Jenkins API
** Jenkins API
Folders
OWASP Markup Formatter
** ASR API
** JSON Path API
** Structs

** - required dependency

Jenkins 2.528.3

This screenshot shows the Jenkins 'Getting Started' page. It displays a grid of available plugins: Folders, OWASP Markup Formatter, Build Timeout, Credentials Binding, Timestamper, Workspace Cleanup, Ant, Gradle, Pipeline, GitHub Branch Source, Pipeline: GitHub Groovy Libraries, Pipeline Graph View, Git, SSH Build Agents, Matrix Authorization Strategy, LDAP, Email Extension, Mailer, and Dark Theme. A tooltip for 'OWASP Markup Formatter' indicates it has dependencies on commons-lang3 v3.x Jenkins API, Jenkins API, and Folders. A note at the bottom right states '** - required dependency'. The Jenkins version is listed as 2.528.3.

Not secure 18.171.229.45:8080

Getting Started

Create First Admin User

Username

Password

Confirm password

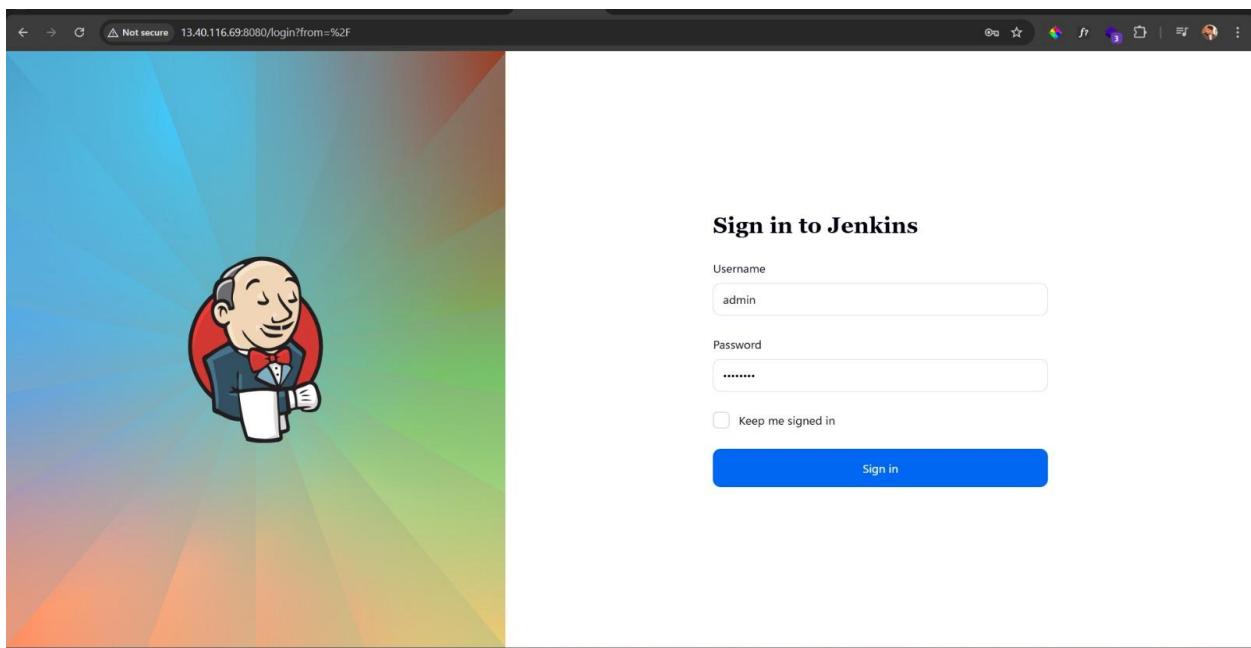
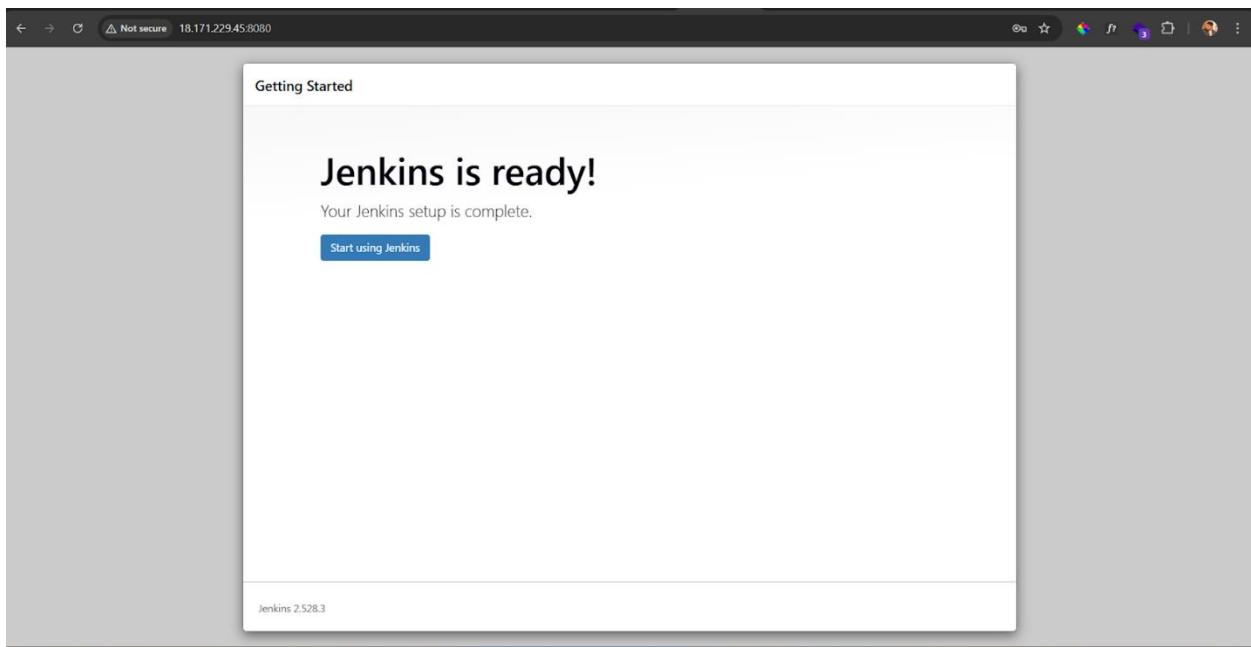
Full name

E-mail address

Skip and continue as admin

Jenkins 2.528.3

This screenshot shows the 'Create First Admin User' step of the Jenkins setup wizard. It contains five input fields: 'Username', 'Password', 'Confirm password', 'Full name', and 'E-mail address'. Below the fields are two buttons: 'Skip and continue as admin' and 'Save and Continue'. The Jenkins version is listed as 2.528.3.



- Click create New Item on the top left

The screenshot shows the Jenkins dashboard at the URL 13.40.116.69:8080. The main header says "Not secure". On the left, there's a sidebar with "Jenkins" and a "+ New Item" button. The main area has tabs for "Build History" and "All". Below that is a table for the "Build Queue" with one entry: "Jenkinspipe" (Status: S, Last Success: N/A, Last Failure: N/A, Last Duration: N/A). At the bottom, there are icons for S, M, and L. The address bar at the bottom shows 13.40.116.69:8080/view/all/newJob.

- Give name & Select Pipeline

The screenshot shows the "New Item" creation dialog at the URL 13.40.116.69:8080/view/all/newJob. The title is "New Item". It asks for an item name ("sam-streaming-app") and lists item types: "Freestyle project", "Pipeline" (selected), "Multi-configuration project", "Folder", and "Multibranch Pipeline". A blue "OK" button is at the bottom.

- Select GitHub Hook trigger for GITScm polling (We need to setup webhook in the git repo)

The screenshot shows the Jenkins job configuration page for a job named "sam-streaming-app". The "General" tab is selected. Under the "Triggers" section, the "GitHub hook trigger for GITScm polling" checkbox is checked. Other options like "Build after other projects are built", "Build periodically", "Poll SCM", and "Trigger builds remotely" are unchecked.

The screenshot shows the Jenkins job configuration page for a job named "sam-streaming-app". The "Pipeline" tab is selected. Under the "Definition" section, the dropdown menu is set to "Pipeline script from SCM". Below this, the "SCM" dropdown is set to "Git". A modal dialog is open, showing the "Repositories" section with a "Repository URL" input field that has a red error message: "Please enter Git repository." There is also a "Credentials" dropdown set to "- none -" and a "+ Add" button. At the bottom of the dialog are "Save" and "Apply" buttons.

- Provide that Git repo link here and select your branch

The screenshot shows a pipeline configuration interface. At the top, there's a dropdown menu labeled "Pipeline script from SCM". Below it, under "SCM", a "Git" provider is selected. The "Repositories" section contains a single repository entry with a "Repository URL" of "https://github.com/SamDonald-A/StreamingApp.git". Under "Credentials", there is a dropdown set to "- none -" and a "+ Add" button. An "Advanced" button is also present. A "+ Add Repository" button is located at the bottom left of this section. In the "Branches to build" section, a "Branch Specifier" is set to "*/*main". There is a "+ Add Branch" button and a "Repository browser" dropdown set to "(Auto)".

- Setup Webhook in Github Repository – Goto settings of your repo and click webhook

The screenshot shows the "Webhooks" section of a GitHub repository settings page. The sidebar includes "Actions", "Models", "Webhooks" (which is highlighted), "Copilot", "Environments", "Codespaces", and "Pages". The main content area is titled "Default branch" and contains the text: "The default branch is considered the ‘base’ branch in your repository automatically made, unless you specify a different branch." Below this is a "main" branch selector and an edit icon. At the bottom right, there is a "Releases" section.

- Click add New then add Payload and select Json

The screenshot shows the GitHub settings interface for a repository named 'StreamingApp'. The 'Webhooks' tab is selected. On the left, there's a sidebar with 'Moderation options' expanded, showing 'Code and automation' and 'Webhooks' (which is currently selected). At the top right, there's a 'Add webhook' button.

This screenshot shows the 'Webhooks' configuration form. The 'Payload URL *' field contains 'http://13.40.116.69:8080/github-webhook/'. The 'Content type *' dropdown is open, showing 'application/x-www-form-urlencoded' (selected), 'application/json' (highlighted in blue), and 'application/x-www-form-urlencoded' again. The 'SSL verification' section below is collapsed.

This screenshot shows the 'Webhooks' list. It displays one webhook entry: 'http://13.40.116.69:8080/github-webhook...' (push). To the right of the URL are 'Edit' and 'Delete' buttons. Below the list, it says 'This hook has never been triggered.'

- Now push the code to check the webhook trigger

```

0   <Grid container spacing={0} alignItems="center" >
1     <Grid item xs={12} md={7}>
2       <Typography
3         variant={isMobile ? 'h3' : 'h1'}
4         sx={{ fontWeight: 800, lineHeight: 1.05, mb: 3 }}
5       >
6         Stream premium cinema from anywhere. You, now + Uncommitted changes
7       </Typography>
8       <Typography variant="h6" color="#rgba(255,255,255,0.75)" sx={{ mb: 4 }}>
9         Join millions of viewers discovering new worlds, sharing live reactions, and enjoying buffer-free 4K
10      </Typography>
11      <Stack direction={{ xs: 'column', sm: 'row' }} spacing={2}>
12        <Button
13          variant="contained"
14          size="large"

```

```

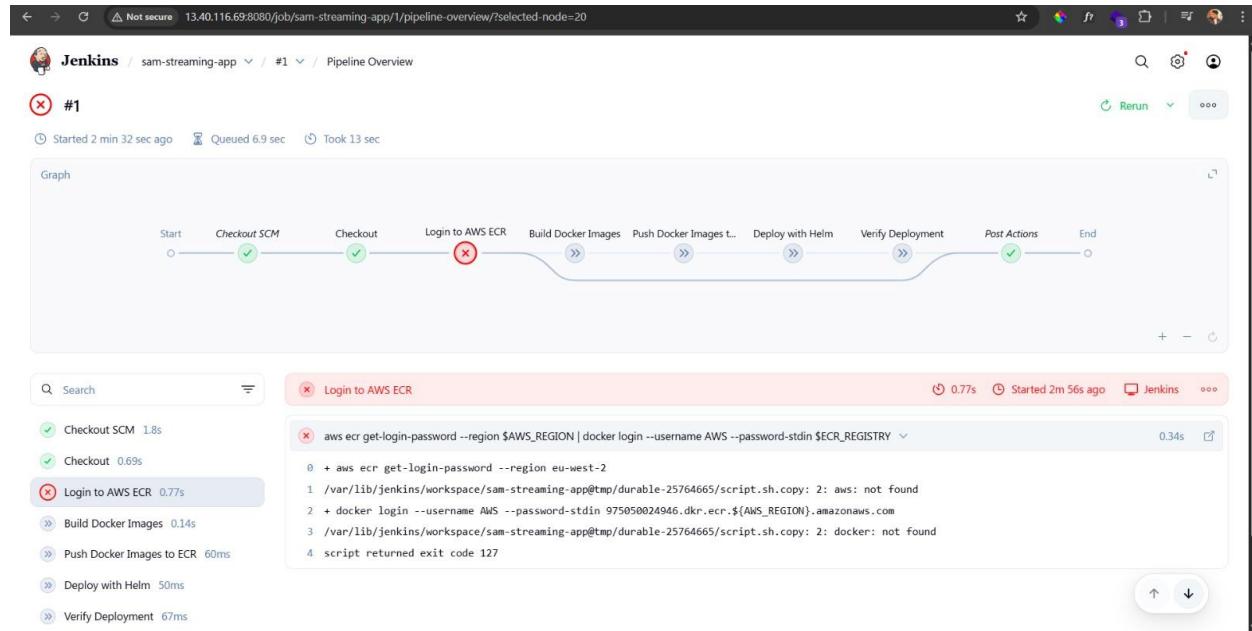
> public
> src
> components
> config
> contexts
> hooks
> pages
JS AdminDashboard.js
JS Browse.js
JS ForgotPassword.js
JS LandingPage.js
JS Login.js
JS Profile.js
JS Register.js
JS Settings.js
JS StreamingPage.js
> services
> styles
# App.css
> OUTPUT
> TERMINAL
create mode 100644 streaming-app-helm/templates/frontend.yaml
create mode 100644 streaming-app-helm/templates/ingress.yaml
create mode 100644 k8s/namespace.yaml
create mode 100644 k8s/streaming/streaming-deployment.yaml
create mode 100644 k8s/streaming/streaming-service.yaml ...
To https://github.com/SamDonald-A/StreamingApp.git
    f79b721..860e53e main -> main
PS E:\Hero VIRED\Assignments\Orchestration and scaling Assignment\StreamingApp> git add .
PS E:\Hero VIRED\Assignments\Orchestration and scaling Assignment\StreamingApp> git commit -m "Streaming App second commit - Jenkins check"
[main 05db669] Streaming App second commit - Jenkins check
  1 file changed, 1 insertion(+), 1 deletion(-)
PS E:\Hero VIRED\Assignments\Orchestration and scaling Assignment\StreamingApp> git push origin
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 12 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 528 bytes | 528.00 KiB/s, done.
Total 6 (delta 5), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (5/5), completed with 5 local objects.
To https://github.com/SamDonald-A/StreamingApp.git
    860e53e..05db669 main -> main
PS E:\Hero VIRED\Assignments\Orchestration and scaling Assignment\StreamingApp>

```

- And we see the Webhook triggered the pipeline

The screenshot shows the Jenkins job configuration page for 'sam-streaming-app'. The top navigation bar includes links for 'Status', 'Changes', 'Build Now', 'Configure', 'Delete Pipeline', 'Stages', 'Rename', 'Pipeline Syntax', and 'GitHub Hook Log'. The main content area displays the 'Permalinks' section, which lists the 'Last build (#1), 0.92 sec ago'. Below this is a 'Builds' section with a dropdown menu.

- Study the log error



Step 6: Check Jenkins Host server requirements for EKS to run the app

Make sure all the services are installed

- Docker
- Helm
- Aws CLI
- Kubectl
- At least t3.medium in EC2 for running the pipeline
- At least 20gb storage for the npm and other installation process on the machine
- EKS IAM Role permissions
- Jenkins
- Jenkins credentials
- Jenkins plugins
- Email Notification setup

```

inflating: aws/dist/wheel-0.45.1.dist-info/INSTALLER
inflating: aws/dist/wheel-0.45.1.dist-info/RECORD
inflating: aws/dist/wheel-0.45.1.dist-info/entry_points.txt
inflating: aws/dist/wheel-0.45.1.dist-info/WHEEL
inflating: aws/dist/wheel-0.45.1.dist-info/METADATA
inflating: aws/dist/wheel-0.45.1.dist-info/REQUESTED
inflating: aws/dist/wheel-0.45.1.dist-info/direct_url.json
ubuntu@ip-172-31-1-215:~$ sudo ./aws install
You can now run /usr/local/bin/aws --version
ubuntu@ip-172-31-1-215:~$ aws --version
aws-cli/2.32.26 Python/3.13.11 Linux/6.14.0-1018-aws exe/x86_64/ubuntu.24
ubuntu@ip-172-31-1-215:~$ sudo apt install -y docker.io
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
bridge-utils containerlnd dns-root-data dnsmasq-base pigz runc ubuntu-fan
Suggested packages:
ifupdown aufs-tools cgroups-mount cgroup-lite debbootstrap docker-buildx docker-compose-v2 docker-doc rinse zfs-fuse | zfsutils
The following NEW packages will be installed:
bridge-util containerlnd dns-root-data dnsmasq-base docker.io pigz runc ubuntu-fan
0 upgraded, 8 newly installed, 0 to remove and 44 not upgraded.
Need to get 76.0 MB of archives.
After this operation, 288 MB of additional disk space will be used.
Get:1 http://eu-west-2.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 pigz amd64 2.8-1 [65.6 kB]
Get:2 http://eu-west-2.ec2.archive.ubuntu.com/ubuntu noble/main amd64 bridge-utils amd64 1.7.1-lubuntu2 [33.9 kB]
Get:3 http://eu-west-2.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 runc amd64 1.3.3-0ubuntu1~24.04.3 [8815 kB]

```

- Add all required plugins in Jenkins

The screenshot shows the Jenkins plugin manager interface. The search bar at the top contains the text "AWS Cred". Below the search bar, there are tabs for "Updates", "Available plugins", "Installed plugins" (which is selected), and "Advanced settings". A sidebar on the left lists "Updates" (10), "Available plugins", "Installed plugins" (96), and "Advanced settings". The main content area displays the "AWS Credentials Plugin" details, including its name, version, description, and status (Health: green, Enabled). There is also a link to report an issue.

- Add credentials

The screenshot shows the Jenkins global configuration page. At the top, there is a search bar and a "Manage Jenkins" header. Below the header, there are several management sections: "Nodes" (Add, remove, control and monitor nodes), "Clouds" (Add, remove, and configure cloud instances), "Appearance" (Configure look and feel), "Security" (Secure Jenkins), "Users" (Create/delete/modifying users), "Status Information" (System Information, System Log, Load Statistics), and "About Jenkins" (See version and license information). The "Credentials" section is highlighted, indicating it is currently selected.

Not secure 13.40.116.69:8080/manage/credentials/store/system/domain/_/newCredentials

Jenkins / Manage Jenkins / Credentials / System / Global credentials (unrestricted...)

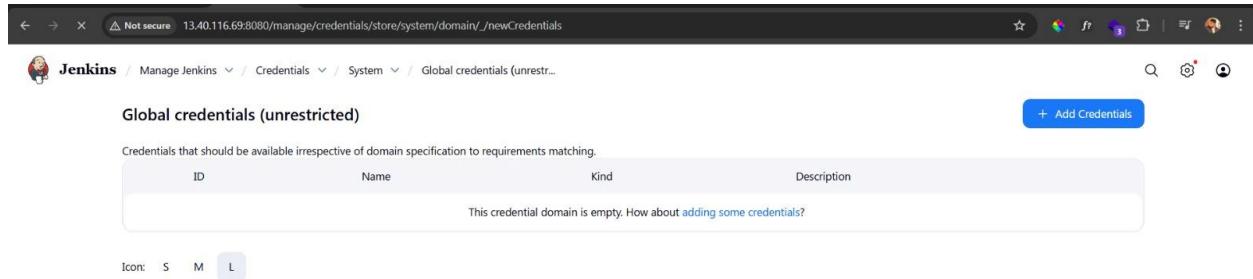
Global credentials (unrestricted)

Credentials that should be available irrespective of domain specification to requirements matching.

ID	Name	Kind	Description
This credential domain is empty. How about adding some credentials?			

Icon: S M L

+ Add Credentials



Not secure 13.40.116.69:8080/manage/credentials/store/system/domain/_/newCredentials

Jenkins / Manage Jenkins / Credentials / System / Global credentials (unrestricted...)

New credentials

Kind

- Username with password
- Username with password
- AWS Credentials
- GitHub App
- SSH Username with private key
- Secret file
- Secret text
- X.509 Client Certificate
- Certificate

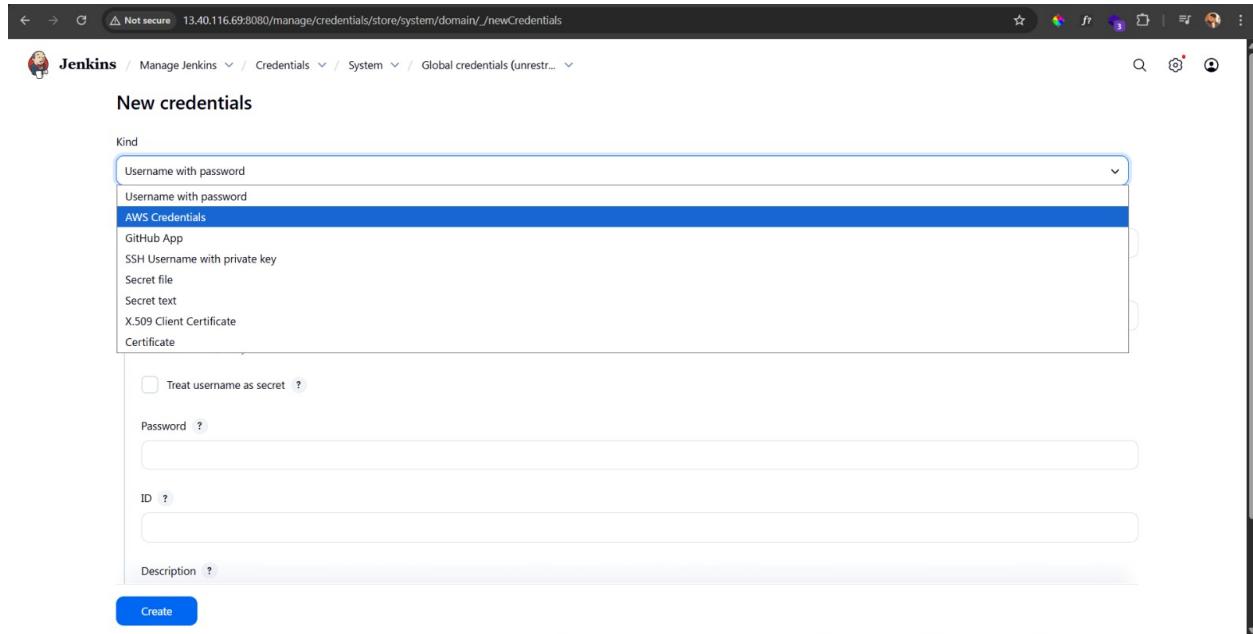
Treat username as secret ?

Password ?

ID ?

Description ?

Create



- Add your AWS Secret and ID

Jenkins / Manage Jenkins / Credentials / System / Global credentials (unrestr... / New credentials

New credentials

Kind: AWS Credentials

Scope: Global (Jenkins, nodes, items, all child items, etc)

ID: aws-creds

Description: Sam AWS ECR access

Access Key ID: AKIA6GBCMU7ZOEMPTF4J

Secret Access Key:

Please specify the Secret Access Key

Create

- Set up Email notification – Got to system and find email notification and setup then test the email sent as a notification

Not secure 13.40.116.69:8080/manage/

Jenkins / Manage Jenkins

Manage Jenkins

Building on the built-in node can be a security issue. You should set up distributed builds. See the documentation.

Set up agent Set up cloud Dismiss

Java 17 end of life in Jenkins

You are running Jenkins on Java 17, support for which will end on or after Mar 31, 2026. Refer to the documentation for more details.

More Info Ignore

System Configuration

System: Configure global settings and paths.

Tools: Configure tools, their locations and automatic installers.

Nodes: Add, remove, control and monitor the various nodes that Jenkins runs jobs on.

Clouds: Add, remove, and configure cloud instances to provision agents on-demand.

Plugins: Add, remove, disable or enable plugins that can extend the functionality of Jenkins.

Appearance: Configure the look and feel of Jenkins.

Security

Security: Secure Jenkins; define who is allowed to access/use the system.

Credentials: Configure credentials.

Credential Providers: Configure the credential providers and types.

13.40.116.69:8080/manage/configure

Not secure 13.40.116.69:8080/manage/configure

Jenkins / Manage Jenkins / System

E-mail Notification

SMTP server: smtp.gmail.com

Default user e-mail suffix:

Advanced ▾ Edited

Use SMTP Authentication ?
User Name: samdonaldand@gmail.com
⚠️ For security when using authentication it is recommended to enable either TLS or SSL.

Password:
 Use SSL ? Use TLS

Save Apply

Not secure 13.40.116.69:8080

Jenkins / Manage Jenkins / System

Charset: UTF-8

Test configuration by sending test e-mail
Test e-mail recipient: samdonaldand@gmail.com

Email was successfully sent Test configuration

GitHub Pull Requests

Published Jenkins URL:

Actualise local repo on factory creation

Save Apply

Saved

REST API Jenkins 2.528.3

Compose

Test email #1 Inbox x

to me address not configured yet <samdonaldand@gmail.com>

This is test email #1 sent from Jenkins

Reply Forward ...

1 of 10,508 1:36 AM (0 minutes ago) ⭐ ⓘ ↵ ⏺

ox 5,108
red
oized
it
fts
chases 11
dates 5,709

- Then Change the Jenkins pipeline flow according to your requierments and push the code to the repository

```

PROBLEMS 16 PORTS GITLENS SPELL CHECKER 16
TERMINAL
Counting objects: 100% (11/11), done.
Delta compression using up to 12 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 528 bytes | 528.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0 ...
PS E:\Hero VIRED\Assignments\Orgestration and scaling Assignment\StreamingApp> git commit -m "Streaming App third commit - Jenkins check"
[main e32688b] Streamg App third commit - Jenkins check
 1 file changed, 29 insertions(+), 7 deletions(-)
PS E:\Hero VIRED\Assignments\Orgestration and scaling Assignment\StreamingApp> git push origin
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 646 bytes | 646.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/SamDonald-A/StreamingApp.git
 05db669..e32688b main -> main
PS E:\Hero VIRED\Assignments\Orgestration and scaling Assignment\StreamingApp>

```

- Loadbalancer also created

Name	State	Type	Scheme	IP address type	VPC ID	Available
aabff10116cf140c587d7fc5a1096116	Active	network	Internet-facing	IPv4	vpc-019ffc70e219434a	3 Available

- Deployment Success and Notification also sent to the mail on success and check the loadbalancer on the browser

52

SUCCESS: sam-streaming-app #49 Inbox

 address not configured yet <samdonaldand@gmail.com>
to me ▾

Build Successful

Job: sam-streaming-app
Build: #49
URL: <http://18.175.248.105:8080/job/sam-streaming-app/49/>

11

28

Reply Forward Smiley

Jenkins / sam-streaming-app #49 / Pipeline Overview

Graph

```

graph LR
    Start((Start)) --> SCM((Checkout SCM))
    SCM --> Checkout((Checkout))
    Checkout --> Login((Login to AWS ECR))
    Login --> Configure((Configure kubectl for...))
    Configure --> Build((Build Docker Images))
    Build --> Push((Push Images to ECR))
    Push --> Deploy((Deploy Application...))
    Deploy --> Verify((Verify Deployment))
    Verify --> Print((Print Application URL))
    Print --> PostActions((Post Actions))

```

Summary:

- Checkout SCM: 0.82s
- Checkout: 0.37s

Overall: Started 2m 15s ago | Jenkins

- Check All Pods and services are running

Jenkins / sam-streaming-app #49 / Console Output

Timestamps View as plain text

System clock time Use browser timezone Elapsed time None

```

20:57:52 + set -e
20:57:52 + kubectl get pods -n streamingapp
20:57:53 NAME READY STATUS RESTARTS AGE
20:57:53 admin-59b7497844-vm95k 1/1 Running 0 36m
20:57:53 auth-8f6778cbf-22cd9 1/1 Running 0 36m
20:57:53 chat-58758b6455-56qgx 1/1 Running 0 36m
20:57:53 frontend-7dc7b49b65-hs2rh 1/1 Running 0 36m
20:57:53 mongo-77d8f9b55-cc98k 1/1 Running 0 36m
20:57:53 streaming-65d68f4455-wmth4 1/1 Running 0 36m
20:57:53 + kubectl get svc -n streamingapp
20:57:53 NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
20:57:53 admin ClusterIP 10.100.231.178 <none> 3003/TCP 36m
20:57:53 auth ClusterIP 10.100.247.53 <none> 3001/TCP 36m
20:57:53 chat ClusterIP 10.100.52.151 <none> 3004/TCP 36m
20:57:53 frontend ClusterIP 10.100.36.54 <none> 80/TCP 36m
20:57:53 mongo ClusterIP 10.100.202.187 <none> 27017/TCP 36m
20:57:53 streaming ClusterIP 10.100.123.171 <none> 3002/TCP 36m
20:57:53 + kubectl get ingress -n streamingapp
20:57:55 NAME CLASS HOSTS ADDRESS PORTS AGE
20:57:55 streamingapp-ingress nginx * aabff10116cf140c587d7fca51096116-9ec49cd106ddae6b.elb.eu-west-2.amazonaws.com 80 36m
20:57:55 + kubectl get deployment -n streamingapp
20:57:55 + grep -v NAME
20:57:56 admin 1/1 1 1 36m
20:57:56 auth 1/1 1 1 36m
20:57:56 chat 1/1 1 1 36m
20:57:56 frontend 1/1 1 1 36m
20:57:56 mongo 1/1 1 1 36m
20:57:56 streaming 1/1 1 1 36m
20:57:56 + kubectl wait --for-condition=Available deployment --all -n streamingapp --timeout=5m
20:57:57 deployment.apps/admin condition met

```

Jenkins / sam-streaming-app #49 / Console Output

Timestamps View as plain text

System clock time Use browser timezone Elapsed time None

```
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Print Application URL)
[Pipeline] sh
20:57:57 + kubectl get svc ingress-nginx-controller -n ingress-nginx -o jsonpath=[.status.loadBalancer.ingress[0].hostname]
20:57:58 + INGRESS_LB=aabff10116cf140c587d7fc51096116-9ec49cd106ddae6b.elb.eu-west-2.amazonaws.com
20:57:58 + [ -n aabff10116cf140c587d7fc51096116-9ec49cd106ddae6b.elb.eu-west-2.amazonaws.com ]
20:57:58 + echo Application URL: http://aabff10116cf140c587d7fc51096116-9ec49cd106ddae6b.elb.eu-west-2.amazonaws.com
20:57:58 Application URL: http://aabff10116cf140c587d7fc51096116-9ec49cd106ddae6b.elb.eu-west-2.amazonaws.com
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Declarative: Post Actions)
[Pipeline] mail
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // timestamps
[Pipeline] }
[Pipeline] // timeout
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

- Cross Check after the pipeline end

aws Search Account ID: 9750-5002-4946 samdonaldand@gmail.com

AM EC2 Elastic Kubernetes Service

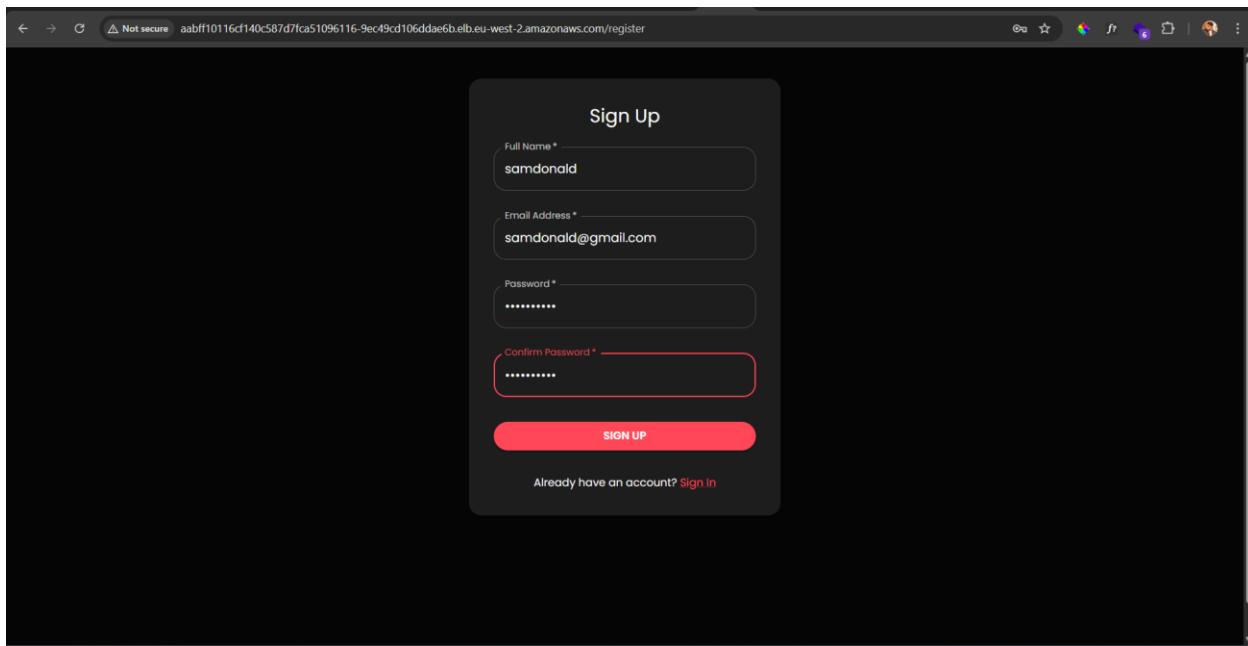
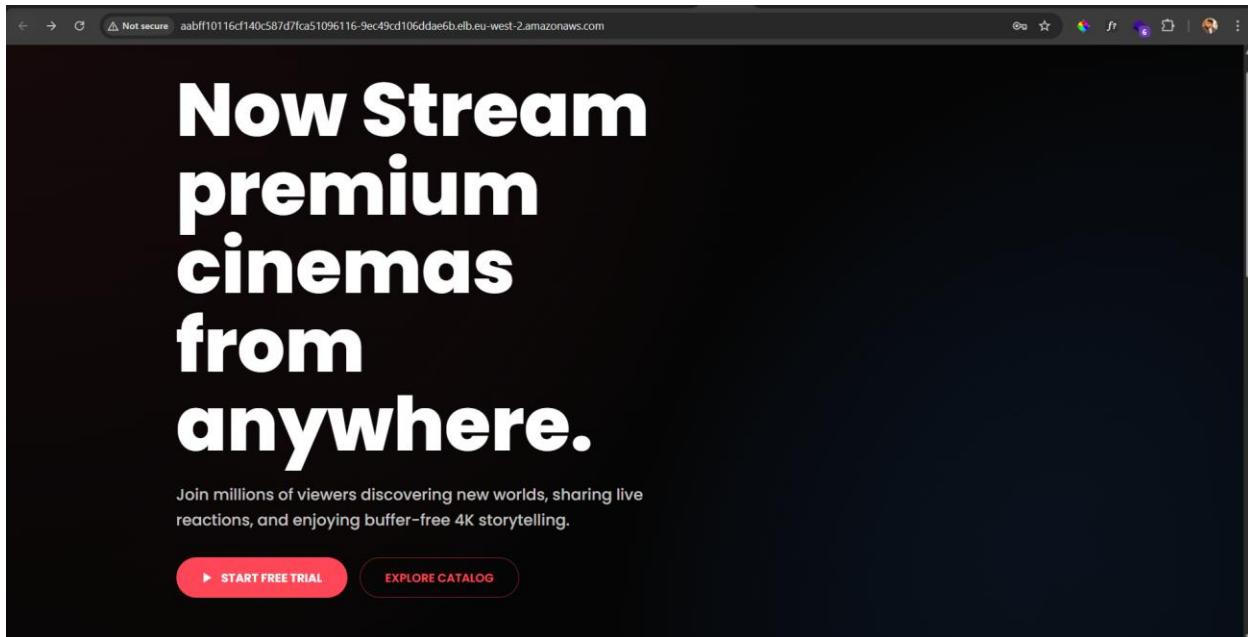
```
2) [ ] 192.168.116.180:3002 169 0:021 404 a8469d67d0b826f7ecd4997352284cdd
20:42:92.153 - - [04/Jan/2026:15:02:13 +0000] "GET / HTTP/1.1" 200 644 "-" "Mozilla/5.0 zgrab/0.x" 110 0.001 [streamingapp-frontend-80] [ ] 192.168.58.153:80 644 0.000 200 873
ubuntu@ip-172-31-1-215:~$ kubectl get pods -n streamingapp
NAME          READY   STATUS    RESTARTS   AGE
admin-59d7497844-vm95k   1/1    Running   0          15m
auth-8f6778cbf-22cd0   1/1    Running   0          15m
chat-58758b6455-56gpx   1/1    Running   0          15m
frontend-7dc7b49b65-hs2rh  1/1    Running   0          15m
mongo-77d8f9bc55-cc98k   1/1    Running   0          15m
streaming-65d6bf4455-vmt4  1/1    Running   0          15m
ubuntu@ip-172-31-1-215:~$ kubectl get svc -n streamingapp
NAME           TYPE        CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
admin           ClusterIP   10.100.231.178   <none>          3003/TCP       16m
auth            ClusterIP   10.100.247.53    <none>          3001/TCP       16m
chat             ClusterIP   10.100.52.151   <none>          3004/TCP       16m
frontend         ClusterIP   10.100.36.54    <none>          80/TCP         16m
mongo            ClusterIP   10.100.202.187   <none>          27017/TCP     16m
streaming        ClusterIP   10.100.123.171   <none>          3002/TCP       16m
ubuntu@ip-172-31-1-215:~$ kubectl get pods -n ingress-nginx
NAME          READY   STATUS    RESTARTS   AGE
ingress-nginx-controller-54d99f99df-c8n2p  1/1    Running   2 (45m ago)  45m
ubuntu@ip-172-31-1-215:~$ kubectl get svc -n ingress-nginx
NAME           TYPE        CLUSTER-IP      EXTERNAL-IP      PORT(S)
ingress-nginx-controller   LoadBalancer   10.100.159.229   aabff10116cf140c587d7fc51096116-9ec49cd106ddae6b.elb.eu-west-2.amazonaws.com  80:31596/TCP, 443:31786/TC
ingress-nginx-controller-admission   ClusterIP   10.100.37.153   <none>          443/TCP
ubuntu@ip-172-31-1-215:~$ 
```

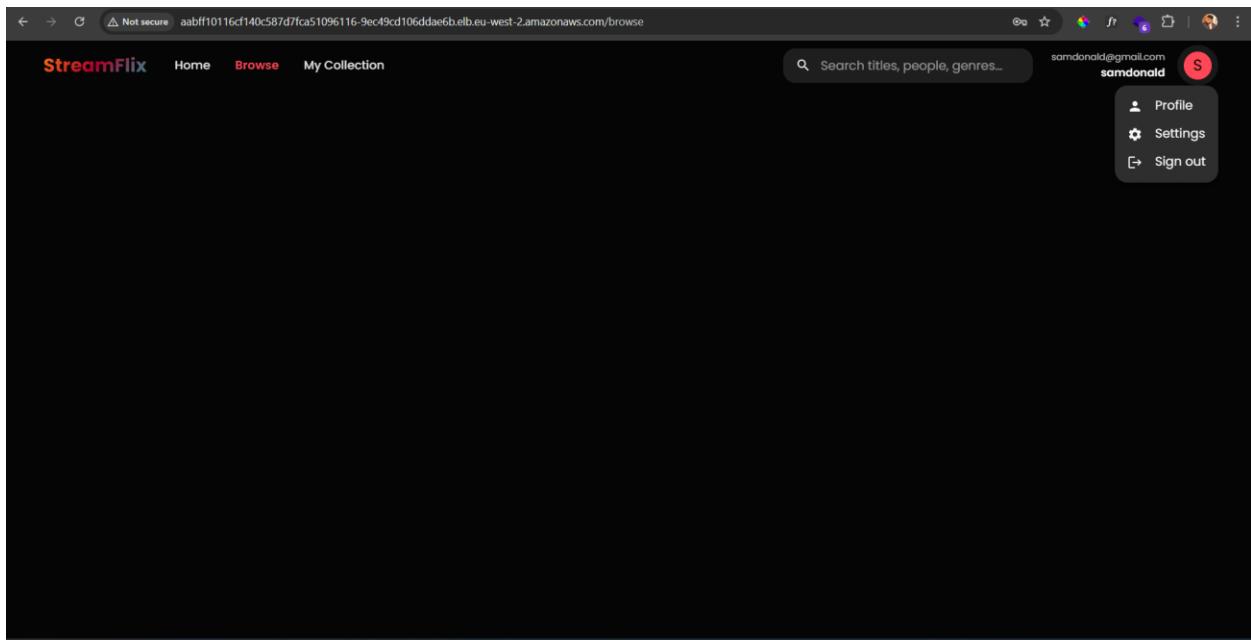
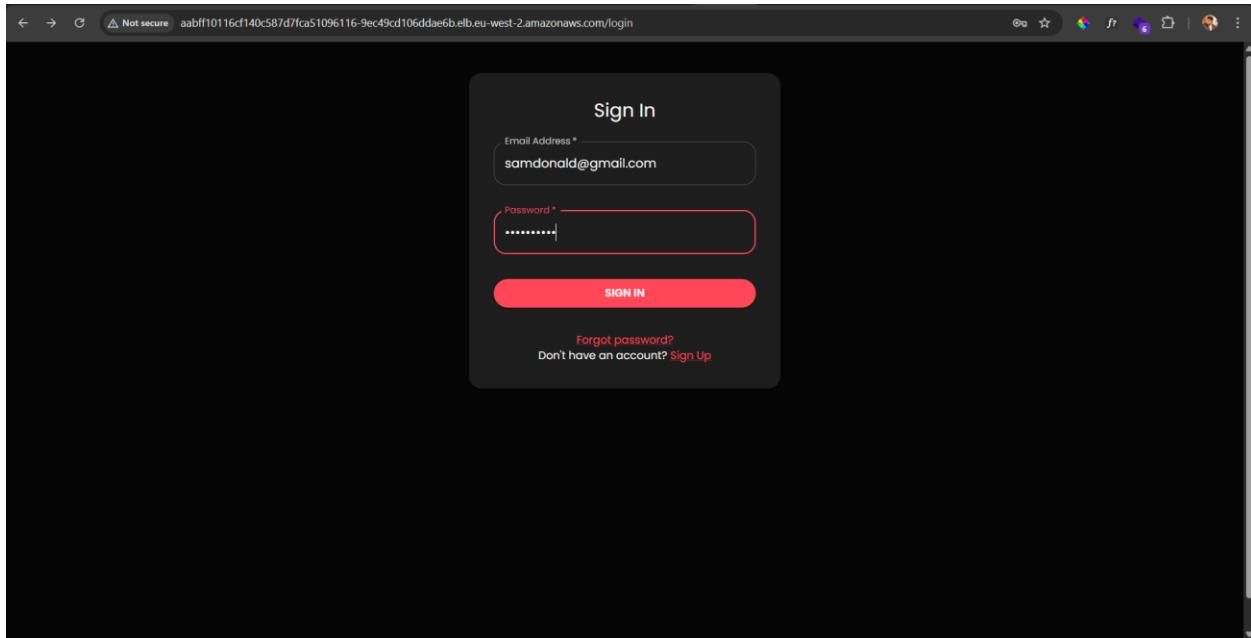
i-04caa52f491715f25 (sam-shopNow-Jenkins-server)

PublicIPs: 18.175.248.105 PrivateIPs: 172.31.1.215

CloudShell Feedback Console Mobile App © 2026, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

- Check the Load balancer on the browser – Create user and see if its works





User Created successfully which means the micro services are able to communicate between them

Documentation by: Sam Donald A
Email: samdonaldand@gmail.com
GitHub: <https://github.com/SamDonald-A>
Website: www.samdonald.in