

Graded Assignment on Container Orchestration

Objective: creating Kubernetes deployment files and a HELM chart for a MERN (MongoDB, Express.js, React.js, Node.js) based application.

GitHub Link: <https://github.com/SamDonald-A/ShopNow-Container-Orchestration>

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

Step 1: Git and local code setup

- Fork or Clone and Push it to your repo

The screenshot shows a GitHub repository page for 'ShopNow-Container-Orchestration'. The repository is public and was created by 'SamDonald-A'. It has 4 commits, 1 branch, and 0 tags. The main branch has 4 commits, all of which are 'Build app with Helm Commit' and were pushed 2 days ago. The repository description states: 'Tasked with creating Kubernetes deployment files and a HELM chart for a MERN (MongoDB, Express.js, React.js, Node.js) based application. The provided project consists of separate frontend (FE) and backend (BE) components.' The page also includes sections for About, Releases, and Packages.

About

Tasked with creating Kubernetes deployment files and a HELM chart for a MERN (MongoDB, Express.js, React.js, Node.js) based application. The provided project consists of separate frontend (FE) and backend (BE) components.

Releases

No releases published [Create a new release](#)

Packages

- Go to backend folder and do npm install then start the backend server to locally run the app for testing

The screenshot shows the VS Code interface with the terminal tab active. The terminal window displays the following command and its output:

```
You, 52 minutes ago | 1 author (you)
# shopNow backend (Node.js API)
FROM node:18-alpine
WORKDIR /app
# Install dependencies
COPY package*.json .
RUN npm ci --only=production --no-audit --prefer-offline
PS E:\Hero VIRED\Assignments\Container Orchestration Assignment\ShopNow-Container-Orchestrat...
See 'kubectl config -h' for help and examples
PS E:\Hero VIRED\Assignments\Container Orchestration Assignment\ShopNow-Container-Orchestrat...
config --region eu-west-2 --name sam-cluster-streaming
Updated context [arn:aws:eks:eu-west-2:975050024946:cluster/sam-cluster-streaming] in C:\Users\Sam Donald\.kube\confi...
npm notice New major version of npm available! 10.8.2 -> 11.7.0
npm notice Changelog: https://github.com/npm/cli/releases/tag/v11.7.0
npm notice To update run: npm install -g npm@11.7.0
npm notice
PS E:\Hero VIRED\Assignments\Container Orchestration Assignment\ShopNow-Container-Orchestrat...
> shopnow-backend@1.0.0 start
> node server.js

ShopNow API server running on port 5000
Health check: http://localhost:5000/api/health
MongoDB connected successfully
No products found, seeding database...
Database seeded with sample products
```

- Do the same for frontend as well

The screenshot shows the VS Code interface with the terminal tab active. The terminal window displays the following command and its output:

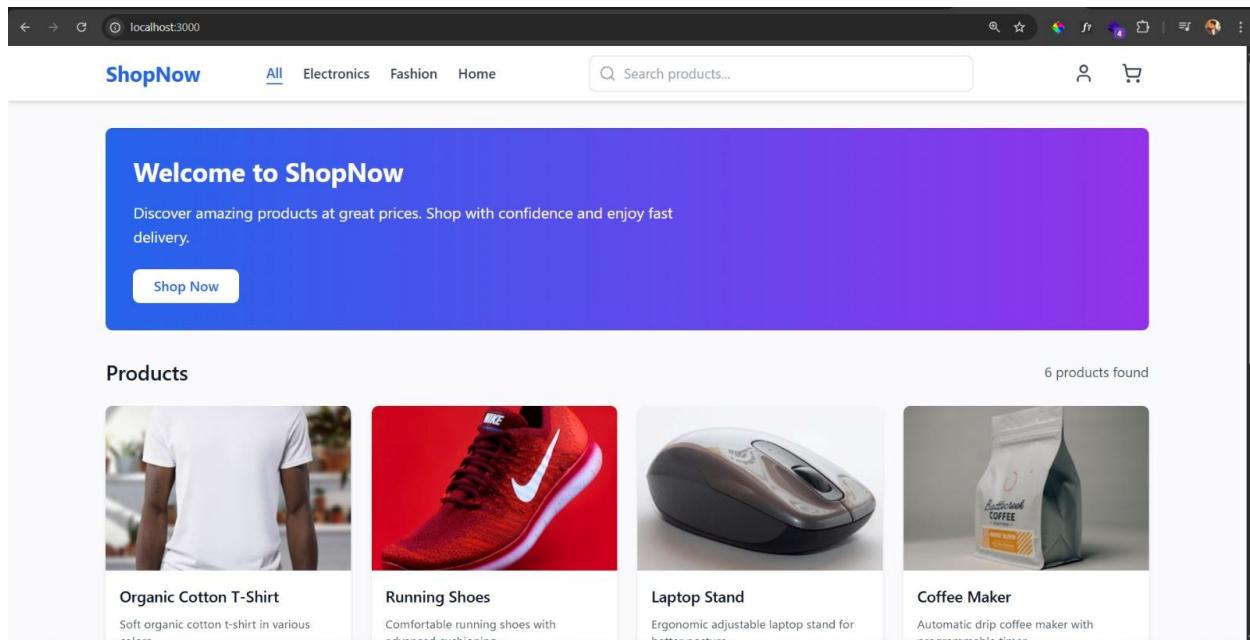
```
ShopNow-Container-Orchestrat... > backend > Dockerfile
# shopNow backend (Node.js API)
FROM node:18-alpine
WORKDIR /app
# Install dependencies
COPY package*.json .
RUN npm ci --only=production --no-audit --prefer-offline
PS E:\Hero VIRED\Assignments\Container Orchestration Assignment\ShopNow-Container-Orchestrat...
270 packages are looking for funding
  run 'npm fund' for details
15 vulnerabilities (4 moderate, 11 high)

To address issues that do not require attention, run:
  npm audit fix

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.
PS E:\Hero VIRED\Assignments\Container Orchestration Assignment\ShopNow-Container-Orchestrat...
> shopnow-frontend@1.0.0 start
> react-scripts start
```

- App is running locally without Docker and Kubernetes



Step 2: Containerization - Docker setup

- Since we have Dockerfile Build it

```

OPEN EDITORS
CONTAINER ORCHESTRATION ASSIGNMENT...
ShopNow-Container-Orchestrator... 9
  > .vscode
  > admin
  > backend
  > docs
  > frontend
  > nginx
  > node_modules
  > public
  > src
  > .dockerignore
Dockerfile
{ package-lock.json
{ package.json
> jenkins
> kubernetes
> scripts
LICENSE
README.md
README.pdf
sprint_1.md
ShopNow Container Orchestration...
OUTPUT
TIMELINE
OUTLINE
NPM SCRIPTS
ShopNow-Container-Orchestrator... 10
  > # Install build deps if needed
  11
  12 # Copy package files (leverage cache)
  13 COPY package*.json .
  14 RUN npm install --no-audit --prefer-offline
  15
  16 # Copy source and build
  17 COPY .
  18 You, 1 hour ago * push code ...
  19 # Build argument for dynamic user path
PROBLEMS 95 PORTS GITLENS SPELL CHECKER 95
TERMINAL
PS E:\Hero VIRED\Assignments\Container Orchestration Assignment\ShopNow-Container-Orchestration\backend> docker build -t shopnow-backend:latest .
=> [internal] load build context
=> transferring context: 73.89kB
=> CACHED [2/7] WORKDIR /app
=> [3/7] COPY package*.json .
=> [4/7] RUN npm install --only=production --no-audit --prefer-offline
=> [5/7] COPY .
=> [6/7] RUN addgroup -S appgroup && adduser -S appuser -G appgroup
=> [7/7] RUN chown -R appuser:appgroup /app
=> exporting to image
=> exporting layers
=> exporting manifest sha256:c0cca42d2c94cd355347715aed22764a8784befc46a27ffcea4d428315ee7c80
=> => exporting config sha256:c18462dcfabf2d33838ea80414fb77f523bbfe2765656265adfe87a2af8c7b
=> => exporting attestation manifest sha256:8fcc7d4cb9803289cd3fc4e4390bd68540d05a477460cc70b0582f6759
=> => exporting manifest list sha256:54324087117e5a0db7614ba99bf47e84a20fb67da70b9310f5be621b4bb73d
=> => naming to docker.io/library/shopnow-backend:latest
=> => unpacking to docker.io/library/shopnow-backend:latest
PS E:\Hero VIRED\Assignments\Container Orchestration Assignment\ShopNow-Container-Orchestration\backend> docker

```

- Need to create repository for frontend and backend in DockerHub for pushing images

Docker Hub - hub.docker.com/repositories/samdonalda

Repositories

All repositories within the **samdonalda** namespace.

Name	Last Pushed	Contains	Visibility	Scout
samdonalda/gateway-service	21 days ago	IMAGE	Public	Inactive
samdonalda/order-service	21 days ago	IMAGE	Public	Inactive
samdonalda/product-service	21 days ago	IMAGE	Public	Inactive
samdonalda/user-service	21 days ago	IMAGE	Public	Inactive
samdonalda/sam_flask_app	about 1 month ago	IMAGE	Public	Inactive
samdonalda/nginx	2 months ago	IMAGE	Public	Inactive
samdonalda/sam-centos-python	3 months ago	IMAGE	Public	Inactive
samdonalda/sam_custom_image	3 months ago	IMAGE	Public	Inactive

[Create a repository](#)

Docker Hub - hub.docker.com/repositories/samdonalda

Create repository

Repository Name* **shopnow-backend**

Short description

A short description to identify your repository. If the repository is public, this description is used to index your content on Docker Hub and in search engines, and is visible to users in search results.

Visibility

Using 0 of 1 private repositories. [Get more](#)

Public Appears in Docker Hub search results

Private Only visible to you

You can push a new image to this repository using the CLI:

```
docker tag local-image:tagname new-repo:tagname
docker push new-repo:tagname
```

Make sure to replace `tagname` with your desired image repository tag.

[Cancel](#) [Create](#)

Docker Hub - hub.docker.com/repositories/samdonalda

Repositories

All repositories within the **samdonalda** namespace.

Name	Last Pushed	Contains	Visibility	Scout
samdonalda/shopnow-backend	1 minute ago	IMAGE	Public	Inactive
samdonalda/shopnow-frontend	1 minute ago	IMAGE	Public	Inactive
samdonalda/gateway-service	21 days ago	IMAGE	Public	Inactive

[Create a repository](#)

- Tag each images from local

```
shop-now:latest          48c/c57/0002      64MB     17MB   U
shopnow-backend:latest  5432408717e       237MB    54.8MB  U
streamingapp-admin:latest 45ec69a0285e      434MB    64.5MB
streamingapp-auth:latest  ffa3d699c90        238MB    55.9MB
streamingapp-chat:latest  aeb0ba14b8cf      299MB    62.1MB
streamingapp-frontend:latest e4be2c80c908     78.1MB   21.9MB
streamingapp-streaming:latest 0129aef2a3eca    377MB    80.1MB
PS E:\Hero VIRED\Assignments\Container Orchestration Assignment\ShopNow-Container-Orchestration\frontend> docker tag shop-now:lates
t sandomald/shopnow-frontend:latest
PS E:\Hero VIRED\Assignments\Container Orchestration Assignment\ShopNow-Container-Orchestration\frontend> docker tag shopnow-backen
d:latest sandomald/shopnow-backend:latest
PS E:\Hero VIRED\Assignments\Container Orchestration Assignment\ShopNow-Container-Orchestration\frontend>
```

- Push it to DockerHub

```
PS E:\Hero VIRED\Assignments\Container Orchestration Assignment\ShopNow-Container-Orchestration\frontend> docker tag shop-now:lates  
t samdonalda/shopnow-frontend:latest  
PS E:\Hero VIRED\Assignments\Container Orchestration Assignment\ShopNow-Container-Orchestration\frontend> docker push samdonalda/sh  
opnow-frontend:latest  
The push refers to repository [docker.io/samdonalda/shopnow-frontend]  
ed84c5dbdc2a: Pushed  
3c854c8cbf46: Pushed  
16eaaaf5f1c0: Pushed  
5e845cc16269: Pushed  
db2ec58da2ef: Pushed  
b407bcc80638: Pushed  
a0fbdb691d7c1: Pushed  
b8f4550756d7: Pushed  
da33b1ad0ac4: Pushed  
de5d475193dd: Pushed  
latest: digest: sha256:48c7c5770002b7084150c5380dd2d07d35d50d5e6d682722946070c8371ef180 size: 856  
PS E:\Hero VIRED\Assignments\Container Orchestration Assignment\ShopNow-Container-Orchestration\frontend>
```

Step 3: Container orchestration - Create Kubernetes manifest files

The image shows a code editor interface with a sidebar on the left containing a file tree and a main panel on the right displaying a YAML configuration file.

File Tree:

- > backend
- > docs
- > frontend
- > jenkins
- ✓ k8s
 - ✓ deployment
 - ! backend.yaml
 - ! frontend.yaml
 - ! mongo.yaml
 - ✓ ingress
 - ! ingress.yaml
 - ✓ namespace
 - ! namesapce.yaml
 - ✓ service
 - ! backend-service.yaml
 - ! frontend-service.yaml
 - ! mongo-service.yaml

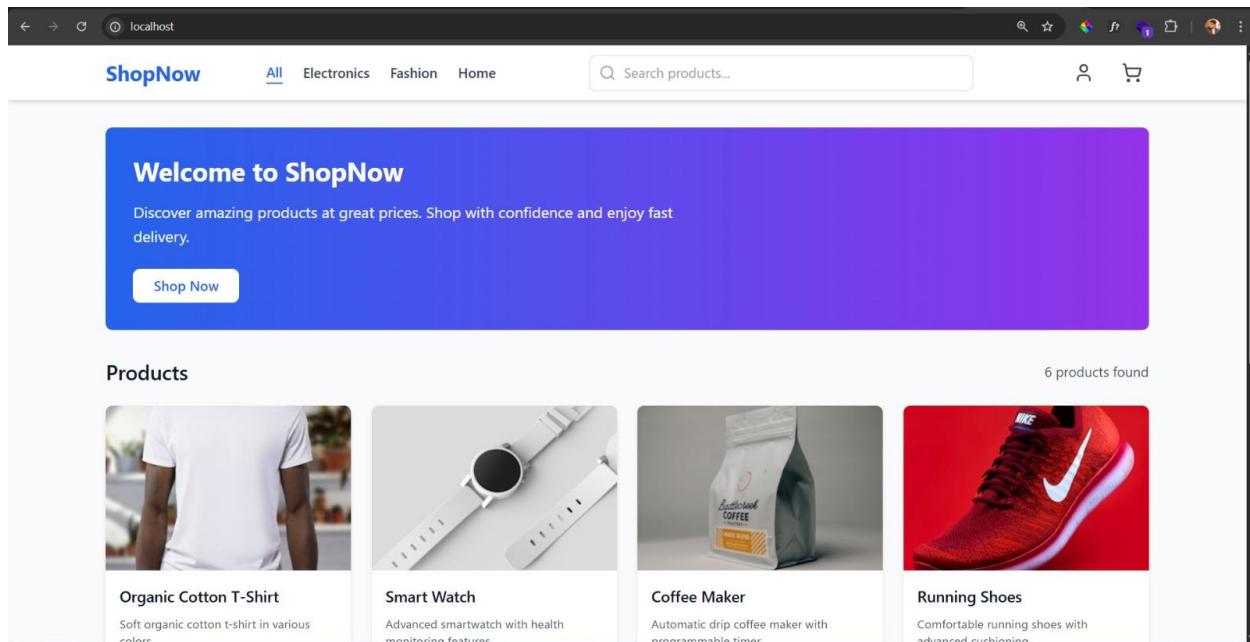
YAML File Content (ingress.yaml):

```
8 spec:  
9   ingressClassName: nginx  
10  rules:  
11    - http:  
12      paths:  
13        - path: /aryan  
14          pathType: Prefix  
15          backend:  
16            service:  
17              name: frontend  
18              port:  
19                number: 80  
20  
21
```

Bottom Status Bar:

- PROBLEMS 8
- PORTS
- GITLENS
- SPELL CHECKER 8
- TERMINAL

- Now app is running in Minikube from Kubernetes – We can Access it via minikube tunnel



Step 4: Create Helm and EKS

- Create Helm folder inside your project root

```

    shopnow-helm
    ├── charts
    ├── templates
    └── ...
        └── Chart.yaml
        └── values.yaml
        └── LICENSE
        └── README.md
        └── README.pdf
        └── sprint_1.md
        └── ShopNow Container Orch...
    └── ...
        └── ...
            └── ...
                └── ...
                    └── ...
                        └── ...
                            └── ...
                                └── ...
                                    └── ...
                                        └── ...
                                            └── ...
                                                └── ...
                                                    └── ...
                                                        └── ...
                                                            └── ...
                                                                └── ...
                                                                    └── ...
                                                                        └── ...
                                                                            └── ...
                                                                                └── ...
                                                                                    └── ...
                                                                                        └── ...
                                                                                            └── ...
                                                                                                └── ...
                                                                ................................................................

```

The terminal output shows the following steps:

- Stopped tunnel for service streamingapp-ingress.
- PS E:\Hero VIRED\Assignments\Container Orchestration Assignment\ShopNow-Container-Orchestration\k8s\ingress> minikube tunnel
- PS E:\Hero VIRED\Assignments\Container Orchestration Assignment\ShopNow-Container-Orchestration\k8s\ingress> kubectl logs deployment/backend -n shopnow-app
- PS E:\Hero VIRED\Assignments\Container Orchestration Assignment\ShopNow-Container-Orchestration\k8s\ingress> helm create shopnow-helm
- Creating shopnow-helm
- PS E:\Hero VIRED\Assignments\Container Orchestration Assignment\ShopNow-Container-Orchestration\k8s\ingress> cd ..
- PS E:\Hero VIRED\Assignments\Container Orchestration Assignment\ShopNow-Container-Orchestration\k8s> cd ..
- PS E:\Hero VIRED\Assignments\Container Orchestration Assignment\ShopNow-Container-Orchestration> helm create shopnow-helm
- Creating shopnow-helm
- PS E:\Hero VIRED\Assignments\Container Orchestration Assignment\ShopNow-Container-Orchestration>

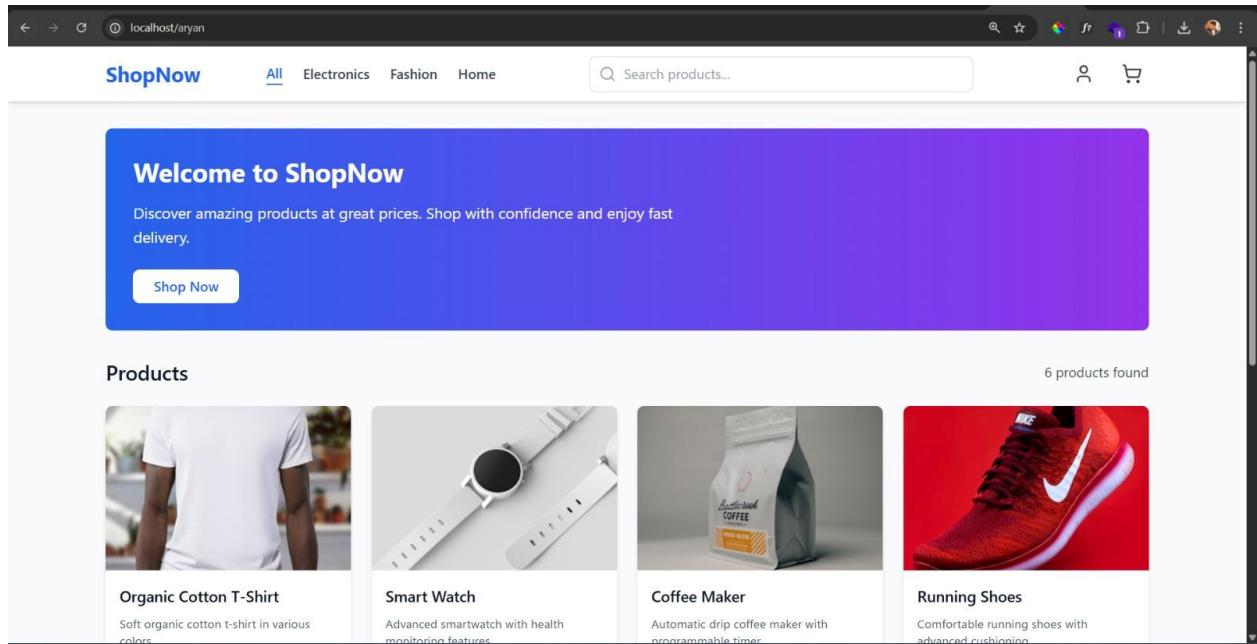
- **Install and deploy helm to create resources – Containers and services are running now**
- **Make sure you delete all the resources that are created before for testing k8s manifest files because helm will create everything fresh again.**

```

TERMINAL
PS E:\Hero VIRED\Assignments\Container Orchestration Assignment\ShopNow-Container-Orchestration> kubectl create namespace shopnow-a
PP
● PS E:\Hero VIRED\Assignments\Container Orchestration Assignment\ShopNow-Container-Orchestration> helm install shopnow ./shopnow-helm
m -n shopnow-app
NAME: shopnow
LAST DEPLOYED: Tue Jan  6 20:31:07 2026
NAMESPACE: shopnow-app
STATUS: deployed
REVISION: 1
DESCRIPTION: Install complete
TEST SUITE: None
● PS E:\Hero VIRED\Assignments\Container Orchestration Assignment\ShopNow-Container-Orchestration> helm list -n shopnow-app
NAME           NAMESPACE   REVISION   UPDATED             STATUS      CHART
shopnow        shopnow-app  1          2026-01-06 20:31:07.5460791 +0530 IST  deployed   shopnow-helm-0.1.0    1.0
● PS E:\Hero VIRED\Assignments\Container Orchestration Assignment\ShopNow-Container-Orchestration> kubectl get all -n shopnow-app
NAME                           READY   STATUS    RESTARTS   AGE
pod/backend-7775c884bf-vvjkf  1/1    Running   0          2m41s
pod/frontend-b54d97c9b-7nkqh  1/1    Running   0          2m41s
pod/mongo-7b9f858cbb-brf21   1/1    Running   0          2m41s
service/backend-service        ClusterIP  10.109.182.231 <none>    5000/TCP  2m41s

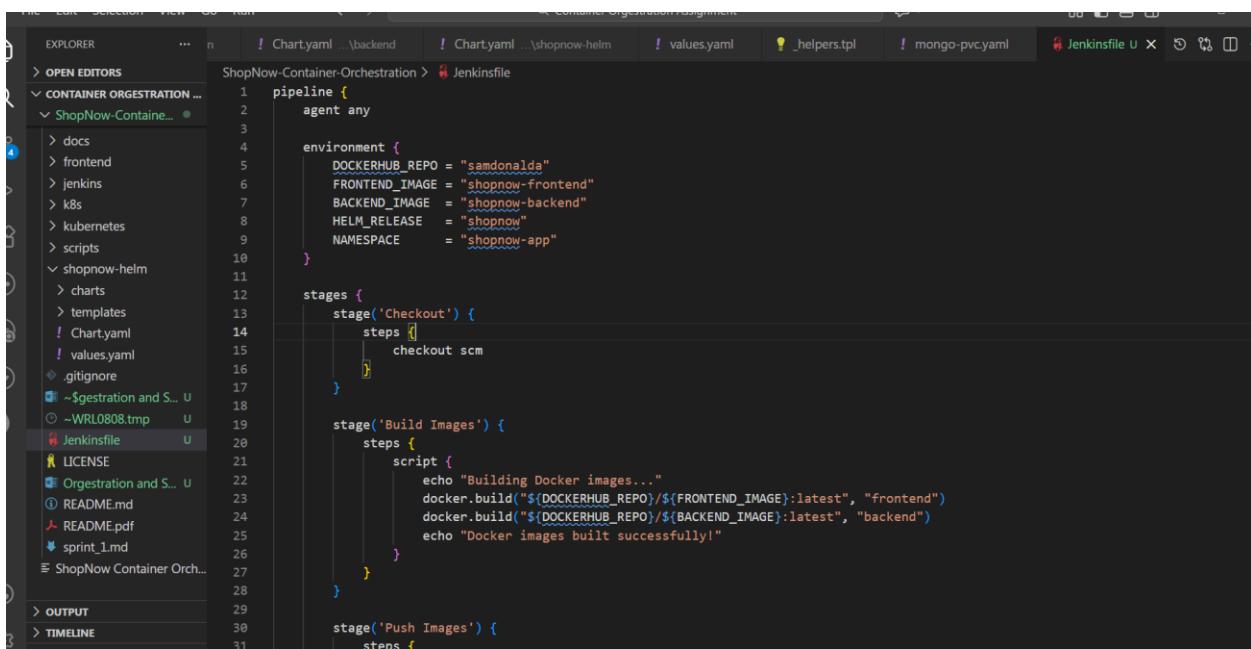
```

- **Resources are created and app deployed using helm chart**



Step 5: Jenkinsfile and Jenkins setup

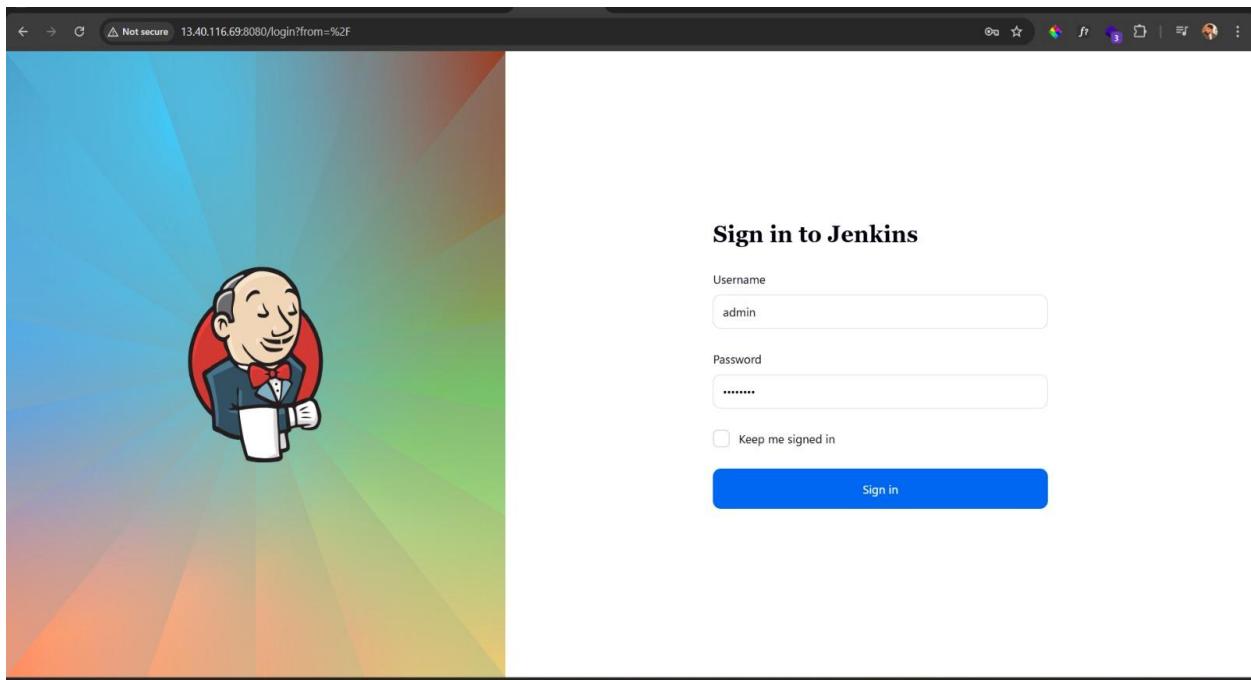
- Create Jenkins file



The screenshot shows a code editor with a Jenkinsfile open. The file defines a pipeline with stages for checkout, building Docker images, and pushing them. It uses environment variables like DOCKERHUB_REPO, FRONTEND_IMAGE, BACKEND_IMAGE, HELM_RELEASE, and NAMESPACE. The Jenkinsfile is part of a project named 'ShopNow-Container-Orchestration'.

```
1 pipeline {
2     agent any
3
4     environment {
5         DOCKERHUB_REPO = "samdonalda"
6         FRONTEND_IMAGE = "shopnow-frontend"
7         BACKEND_IMAGE = "shopnow-backend"
8         HELM_RELEASE = "shopnow"
9         NAMESPACE = "shopnow-app"
10    }
11
12    stages {
13        stage('Checkout') {
14            steps {
15                checkout scm
16            }
17        }
18
19        stage('Build Images') {
20            steps {
21                script {
22                    echo "Building Docker images..."
23                    docker.build("${DOCKERHUB_REPO}/${FRONTEND_IMAGE}:latest", "frontend")
24                    docker.build("${DOCKERHUB_REPO}/${BACKEND_IMAGE}:latest", "backend")
25                    echo "Docker images built successfully!"
26                }
27            }
28        }
29
30        stage('Push Images') {
31            steps {
32            }
33        }
34    }
35}
```

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



- Click create New Item on the top left

The screenshot shows the Jenkins dashboard at the URL 18.175.58.101:8080. The top navigation bar indicates "Not secure". The dashboard features a central table displaying build statistics for two projects: "Jenkinspipe" and "sam-streaming-app". Below the table are sections for "Build Queue" and "Build Executor Status". A bottom navigation bar includes icons for S, M, and L.

- Give name & Select Pipeline

The screenshot shows the "New Item" creation dialog. In the "Enter an item name" field, the text "shopnow-app-assignment" is entered. Under "Select an item type", the "Pipeline" option is selected, described as "Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.". Other options shown include "Freestyle project", "Multi-configuration project", and "Folder". At the bottom of the dialog is an "OK" button.

- 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 "shopnow-app-assignment". Under the "Triggers" section, the "GitHub hook trigger for GITScm polling" checkbox is checked. The "Pipeline" section shows the "Definition" set to "Pipeline script from SCM" and the "SCM" dropdown set to "Git". The "Save" and "Apply" buttons are visible at the bottom.

- Provide that Git repo link here and select your branch

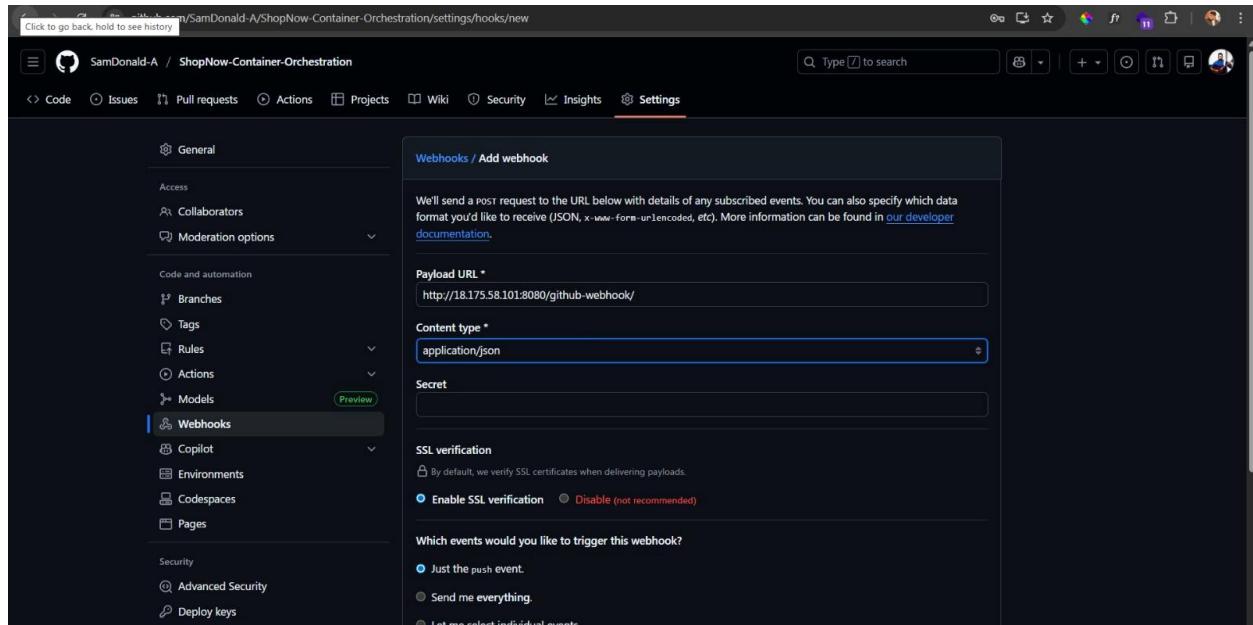
The screenshot shows a GitHub repository page for "ShopNow-Container-Orchestration". The repository has 1 branch and 0 tags. A context menu is open over a commit message, showing options like "Clone", "HTTPS", "SSH", and "GitHub CLI". The "Clone" option is highlighted, showing the URL "https://github.com/SamDonald-A/ShopNow-Container-Orchestration". The repository details on the right include an "About" section describing it as a MERN application with separate frontend and backend components, and sections for "Readme", "MIT license", "Activity", "Releases", and "Create a new release".

The screenshot shows the Jenkins Pipeline configuration page for a job named "shopnow-app-assignment". The "Pipeline" tab is selected in the sidebar. The main area is titled "Definition" and shows "Pipeline script from SCM". Under "SCM", "Git" is selected. The "Repositories" section contains one repository with the URL "https://github.com/SamDonald-A/ShopNow-Container-Orchestration.git" and no credentials defined. There is an "Advanced" button and a "+ Add Repository" link. Below the repositories is a "Branches to build" dropdown. At the bottom are "Save" and "Apply" buttons.

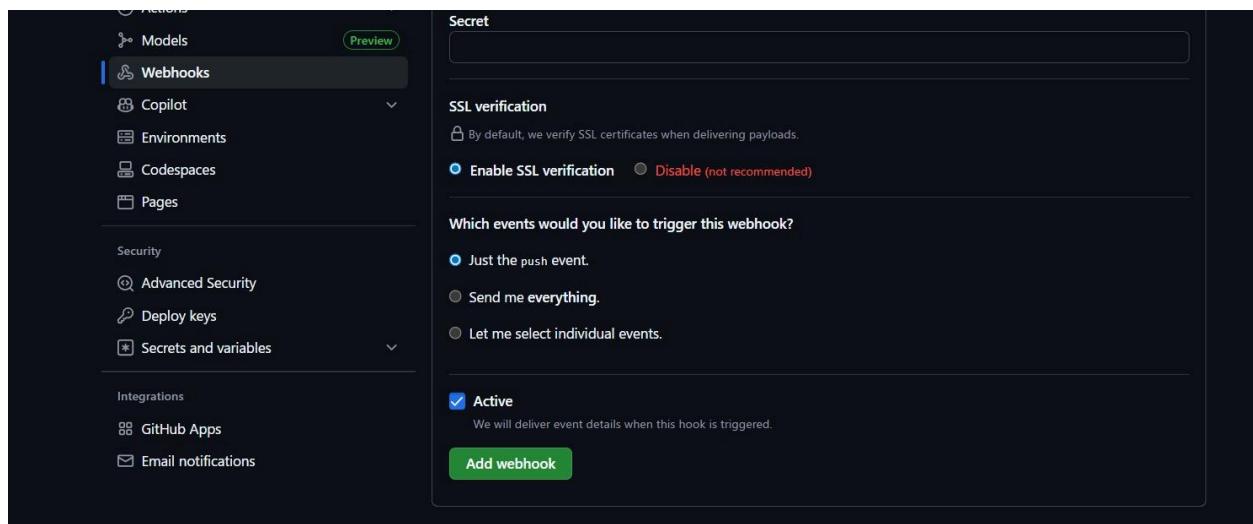
- **Setup Webhook in Github Repository – Goto settings of your repo and click webhook**

The screenshot shows the GitHub repository settings page for "SamDonald-A / ShopNow-Container-Orchestration". The "Settings" tab is active. On the left, the "Webhooks" option under "Code and automation" is highlighted. The main content area is titled "Webhooks" and contains a sub-section "General" with a note about what webhooks do. A "Add webhook" button is visible at the top right of this section. The left sidebar also includes sections for "Access", "Collaborators", "Moderation options", "Branches", "Tags", "Rules", "Actions", "Models", "Copilot", "Environments", "Codespaces", and "Pages".

- Click add New then add Payload and select Json



- Click Add webhook



Now we can push the and the Jenkins will be triggered

- Now setup the Docker Credentials in the Jenkins Global secret

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

New credentials

Kind: Username with password

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

Username: samdonalda

Treat username as secret

Password:

ID: docker-hub-cred

Description: Docker Credentials for Jenkins Pipeline

Create

- Create EKS Cluster

```

aws
[Alt+S] Search
Account ID: 9750-5002-4946
Europe (London) samdonaldand@gmail.com

[AM] [EC2] [Elastic Kubernetes Service]

Kustomize Version: v5.7.1
ubuntu@ip-172-31-1-215:~$ eksctl create cluster --name sam-project-cluster --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
2026-01-07 04:18:28 [i] eksctl version 0.22.10
2026-01-07 04:18:28 [i] using region eu-west-2
2026-01-07 04:18:28 [i] setting availability zones to [eu-west-2a eu-west-2b eu-west-2a]
2026-01-07 04:18:28 [i] subnets for eu-west-2a - public:192.168.0.0/19 private:192.168.96.0/19
2026-01-07 04:18:28 [i] subnets for eu-west-2b - public:192.168.32.0/19 private:192.168.128.0/19
2026-01-07 04:18:28 [i] subnets for eu-west-2a - public:192.168.64.0/19 private:192.168.160.0/19
2026-01-07 04:18:28 [i] nodegroup "ng-1" will use "" [AmazonLinux2023/1.33]
2026-01-07 04:18:28 [i] 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/
2026-01-07 04:18:28 [i] using Kubernetes version 1.33
2026-01-07 04:18:28 [i] creating EKS cluster "sam-project-cluster" in "eu-west-2" region with managed nodes
2026-01-07 04:18:28 [i] will create 2 separate CloudFormation stacks for cluster itself and the initial managed nodegroup
2026-01-07 04:18:28 [i] if you encounter any issues, check CloudFormation console or try 'eksctl utils describe-stacks --region=eu-west-2 --cluster=sam-project-cluster'
2026-01-07 04:18:28 [i] Kubernetes API endpoint access will use default of (publicAccess=true, privateAccess=false) for cluster "sam-project-cluster" in "eu-west-2"
2026-01-07 04:18:28 [i] CloudWatch logging will now be enabled for cluster "sam-project-cluster" in "eu-west-2"
2026-01-07 04:18:28 [i] you can enable it with 'eksctl utils update-cluster-logging --enable-types=(SPECIFY-YOUR-LOG-TYPES-HERE (e.g. all)) --region=eu-west-2 --cluster=sam-project-cluster'
2026-01-07 04:18:28 [i] default addons metrics-server, vpc-cni, kube-proxy, coredns were not specified, will install them as EKS addons
2026-01-07 04:18:28 [i] 1 sequential tasks: ( create cluster control plane "sam-project-cluster",
2026-01-07 04:18:28 [i]   2 sequential sub-tasks: (
2026-01-07 04:18:28 [i]     5 sequential sub-tasks: (
2026-01-07 04:18:28 [i]       1 task: ( create addons ),
2026-01-07 04:18:28 [i]       wait for control plane to become ready,
2026-01-07 04:18:28 [i]       associate IAM OIDC provider,
2026-01-07 04:18:28 [i]       no tasks,
2026-01-07 04:18:28 [i]       update VPC CNI to use IRSA if required,
2026-01-07 04:18:28 [i]     )
2026-01-07 04:18:28 [i]   )
2026-01-07 04:18:28 [i] )
2026-01-07 04:18:28 [i] i-04caa52f491715f25 (sam-shopNow-Jenkins-server)
2026-01-07 04:18:28 [i] PublicIPs: 18.175.58.101 PrivateIPs: 172.31.1.215

```

```

2026-01-07 04:32:59 [i] creating addon: metrics-server
2026-01-07 04:33:00 [i] successfully created addon: metrics-server
2026-01-07 04:33:01 [i] kubectl command should work with "/home/ubuntu/.kube/config", try 'kubectl get nodes'
2026-01-07 04:33:01 [V] EKS cluster "sam-project-cluster" in "eu-west-2" region is ready
ubuntu@ip-172-31-1-215:~$ 

```

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

PublicIPs: 18.175.58.101 PrivateIPs: 172.31.1.215

Clusters (2) Info						Create cluster
		Cluster name	Status	Kubernetes version	Support period	Upgrade policy
<input type="radio"/>	sam-project-cluster	Active	1.33	Upgrade now	Standard support until July 29, 2026	Extended support 12 minutes ago

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
- eksctl

```

ubuntu@ip-172-31-1-215:~$ curl -sLO "https://github.com/weaveworks/eksctl/releases/latest/download/eksctl_Linux_amd64.tar.gz"
ubuntu@ip-172-31-1-215:~$ tar -xzf eksctl_Linux_amd64.tar.gz
ubuntu@ip-172-31-1-215:~$ sudo mv eksctl /usr/local/bin
ubuntu@ip-172-31-1-215:~$ eksctl version
0.221.0
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:~$ kubectl version --client
Client Version: v1.35.0
Kustomize Version: v5.7.1

```

- Also make sure all required plugins installed in Jenkins

Jenkins / Manage Jenkins / Plugins

Plugins

AWS Cred

Upgrades

Available plugins

Installed plugins

Advanced settings

Download progress

Name : AWS-Credential-Plugin 254.v978a.5e206a.d7

Allows storing Amazon IAM credentials within the Jenkins Credentials API. Store Amazon IAM access keys (AWSAccessKeyId and AWSSecretKey) within the Jenkins Credentials API. Also support IAM Roles and IAM MFA Token.

Report an issue with this plugin

Health Enabled

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

```
create mode 100644 Documentation\ContainerOrchestration.DOCKERFILE
create mode 100644 Jenkinsfile
PS E:\Hero\VIRED\Assignments\ContainerOrchestration Assignment\ShopNow-Container-Orchestration> ...
PS E:\Hero\VIRED\Assignments\ContainerOrchestration Assignment\ShopNow-Container-Orchestration> git commit -m "Jenkins & EKS Ready - Commit"
[main dcb55ec] Jenkins & EKS Ready - Commit
 1 file changed, 1 insertion(+), 1 deletion(-)
PS E:\Hero\VIRED\Assignments\ContainerOrchestration Assignment\ShopNow-Container-Orchestration> git push origin
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 12 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (7/7), 3.91 MiB | 106.00 KiB/s, done.
Total 7 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), completed with 1 local object.
To https://github.com/SamDonaldA/ShopNow-Container-Orchestration.git
 50c01ae..dcb55ec main -> main
PS E:\Hero\VIRED\Assignments\ContainerOrchestration Assignment\ShopNow-Container-Orchestration>
```

- Pipeline triggered

The screenshot shows a Jenkins pipeline configuration page. The left sidebar contains navigation links: Status, Changes, Build Now, Configure, Delete Pipeline, Stages, Rename, Pipeline Syntax, and GitHub Hook Log. The main content area is titled "shopnow-app-assignment" and includes a "Permalinks" section with a link to the last build. A "Build Now" button is visible at the top right. The bottom section displays a "Builds" table with one entry: "#1 5:47 AM".

Build	Time
#1	5:47 AM

- Loadbalancer also created

Load balancers (1) What's new?

Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.

Name	State	Type	Scheme	IP address type	VPC ID	Avg.
af041d23e8b5947efa57236f17ba16fb	Active	network	Internet-facing	IPv4	vpc-0ddccdc774a72caa	3 A

- Study the log on the pipeline failure and Fix

Jenkins / shopnow-app-assignment #1 Pipeline Overview

#1

Started 1 min 52 sec ago Queued 6.4 sec Took 1 min 52 sec and counting

Graph

```

graph LR
    Start((Start)) --> CSCM[Checkout SCM]
    CSCM --> Checkout[Checkout]
    Checkout --> BI[Build Images]
    BI --> PI[Push Images]
    PI --> DEK8S[Deploy to EKS using...]
    DEK8S --> VD[Verify Deployment]
    VD --> SPL[Show Pod Logs...]
    SPL --> End((End))
    
```

Deploy to EKS using Helm

1.8s Started 4.6s ago Jenkins

Checkout SCM 1.1s	Checkout 0.62s	Build Images 1m 33s	Push Images 16s	Deploy to EKS using Helm 1.8s	Verify Deployment 0.15s	Show Pod Logs (Optional) 57ms
-------------------	----------------	---------------------	-----------------	-------------------------------	-------------------------	-------------------------------

```

echo "Deploying Helm chart..." helm upgrade --install ${HELM_RELEASE} ./shopnow-helm --namespace ${NAMESPACE} --create-namespace ...
+ echo Deploying Helm chart...
1 Deploying Helm chart...
2 + helm upgrade --install shopnow ./shopnow-helm --namespace shopnow-app --create-namespace
3 Error: Kubernetes cluster unreachable: Get "https://8F700DF3FCAA62A9A10843B047D8C1F9.gr7.eu-west-2.eks.amazonaws.com/version": dial tcp: lookup 8F700DF3FCAA62A9A10843B047D8C1F9.gr7.eu-west-2.eks.amazonaws.com on 127.0.0.53:53: no such host
4 script returned exit code 1

```

- Make sure you have this role in your EC2 IAM of Jenkins Server

Identity and Access Management (IAM)

Sam-Jenkins-EKS-Access-Role

Permissions

Permissions policies (3) Info

You can attach up to 10 managed policies.

Policy name	Type	Attached entities
AmazonEKSWorkerNodePolicy	AWS managed	115
AmazonEKSServicePolicy	AWS managed	6
AmazonEKSClusterPolicy	AWS managed	91

Permissions boundary (not set)

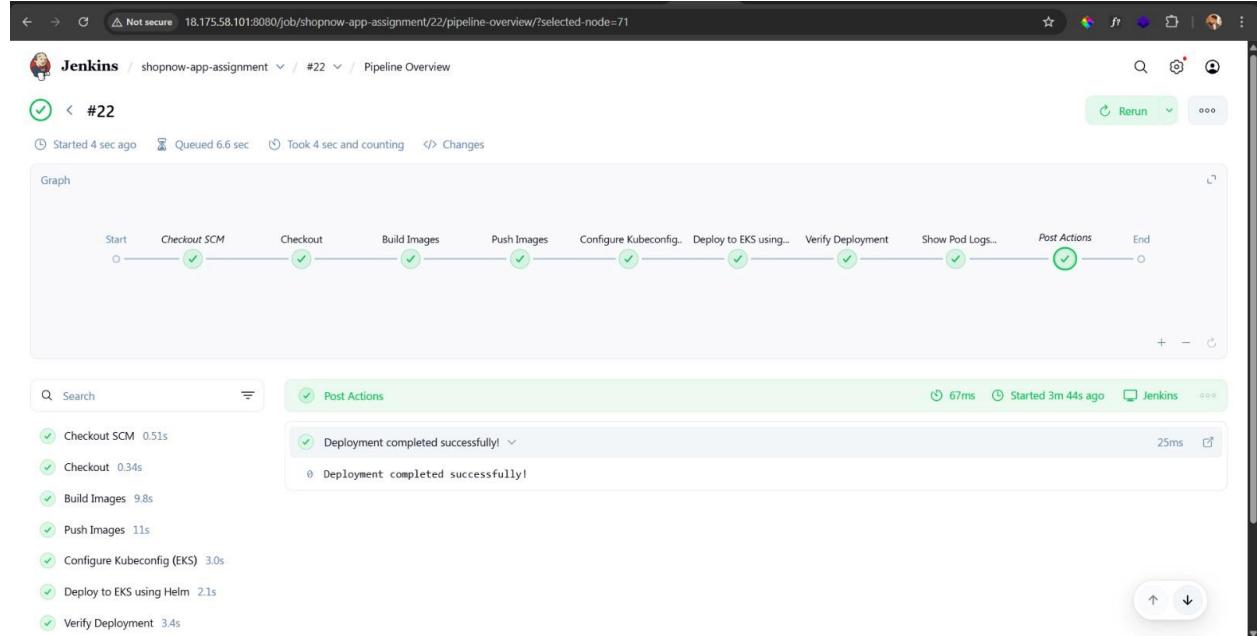
Generate policy based on CloudTrail events

You can generate a new policy based on the access activity for this role, then customize, create, and attach it to this role. AWS uses your CloudTrail events to identify the services and actions used and generate a policy. [Learn more](#)

- **Create and Attach EBS setup in your EC2 Jenkins Host since we need PVC for MongoDB (EKS always Need EBS if PVC required – In AWS PVC means EBS)**
- **And make sure your node group also has the IAM Role policy to access EBS, So that It can create and attach EBS-CSI Volume on the deployment**

```
ubuntu@ip-172-31-1-215:~$ aws eks create-addon --cluster-name sam-project-cluster --addon-name aws-ebs-csi-driver --region eu-west-2
{
  "addon": {
    "addonName": "aws-ebs-csi-driver",
    "clusterName": "sam-project-cluster",
    "status": "CREATING",
    "addonVersion": "v1.54.0-eksbuild.1",
    "health": {
      "issues": []
    },
    "addonArn": "arn:aws:eks:eu-west-2:975050024946:addon/sam-project-cluster/aws-ebs-csi-driver/9ecdcc3c-cd52-d790-08ea-2cc4a41d800c",
    "createdAt": "2026-01-07T12:41:05.162000+00:00",
    "modifiedAt": "2026-01-07T12:41:05.182000+00:00",
    "tags": {},
    "namespaceConfig": {
      "namespace": "kube-system"
    }
  }
}
ubuntu@ip-172-31-1-215:~$ kubectl get pods -n kube-system | grep ebs
ebs-csi-controller-65649bb665-24b2q   6/6   Running  0          35s
ebs-csi-controller-65649bb665-hchjj   6/6   Running  0          35s
ebs-csi-node-g9sn4                   3/3   Running  0          35s
ebs-csi-node-lpsgr                  3/3   Running  0          35s
ubuntu@ip-172-31-1-215:~$ i-04caa52f491715f25 (sam-shopNow-Jenkins-server)
PublicIPs: 18.175.58.101 PrivateIPs: 172.31.1.215
```

- **After debugging, push the code and check the pipeline then Check All Pods and services are running on the success**



Not secure 18.175.58.101:8080/job/shopnow-app-assignment/22/pipeline-overview/?selected-node=61

Jenkins / shopnow-app-assignment #22 Pipeline Overview

Search

echo "Pods:" kubectl get pods -n \${NAMESPACE} -o wide echo "Services:" kubectl get svc -n \${NAMESPACE} echo "Ingress / LoadBalancer:" k... 3.4s

Checkout SCM 0.51s
Checkout 0.34s
Build Images 9.8s
Push Images 11s
Configure Kubeconfig (EKS) 3.0s
Deploy to EKS using Helm 2.1s
Verify Deployment 3.4s
Show Pod Logs (Optional) 4.3s
Post Actions 67ms

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE	NOMINATED NODE
backend-5cc6f9b68f-kzxkm	1/1	Running	0	22m	192.168.31.134	ip-192-168-0-133.eu-west-2.compute.internal	<none>
frontend-65b47d98dd-9rl58	1/1	Running	0	22m	192.168.77.236	ip-192-168-82-14.eu-west-2.compute.internal	<none>
mongo-6989cc89cb-9dv7v	1/1	Running	1 (22m ago)	22m	192.168.12.142	ip-192-168-0-133.eu-west-2.compute.internal	<none>

+ echo Services:
Services:
+ kubectl get svc -n shopnow-app

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
backend-service	ClusterIP	10.100.253.249	<none>	5000/TCP	6h14m
frontend	ClusterIP	10.100.63.116	<none>	80/TCP	6h14m
mongo-service	ClusterIP	10.100.170.151	<none>	27017/TCP	6h14m

+ echo Ingress / LoadBalancer:
Ingress / LoadBalancer:
+ kubectl get ingress -n shopnow-app

NAME	CLASS	HOSTS	ADDRESS	PORTS	AGE
shopnow-ingress	nginx	*	af041d23e8b5947efa57236f17ba16fb-470a35341a4769a4.elb.eu-west-2.amazonaws.com	80	6h14m

- Check the Load balancer on the browser – /aryan after ELB is able to fetch products on the Landing page

Not secure af041d23e8b5947efa57236f17ba16fb-470a35341a4769a4.elb.eu-west-2.amazonaws.com/aryan

ShopNow

All Electronics Fashion Home

Search

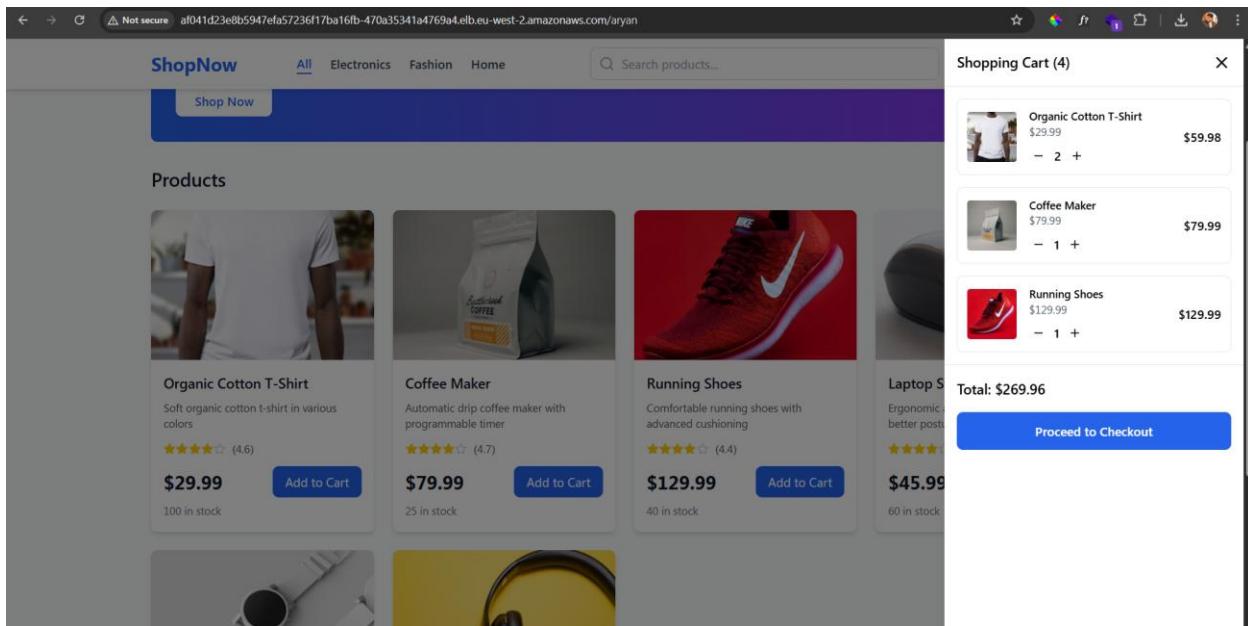
Not secure af041d23e8b5947efa57236f17ba16fb-470a35341a4769a4.elb.eu-west-2.amazonaws.com/aryan

Welcome to sam-ShopNow
Discover amazing products at great prices. Shop with confidence and enjoy fast delivery.

Shop Now

Products 6 products found

Organic Cotton T-Shirt Soft organic cotton t-shirt in various colors 	Coffee Maker Automatic drip coffee maker with programmable timer 	Running Shoes Comfortable running shoes with advanced cushioning 	Laptop Stand Ergonomic adjustable laptop stand for better posture



App Successfully Deployed using Helm, Jenkins and AWS EKS Cluster

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