



Microservices Tracking Application Deployment on AWS EKS using Vault, IAM, and Docker, Angular, Maven

Objective

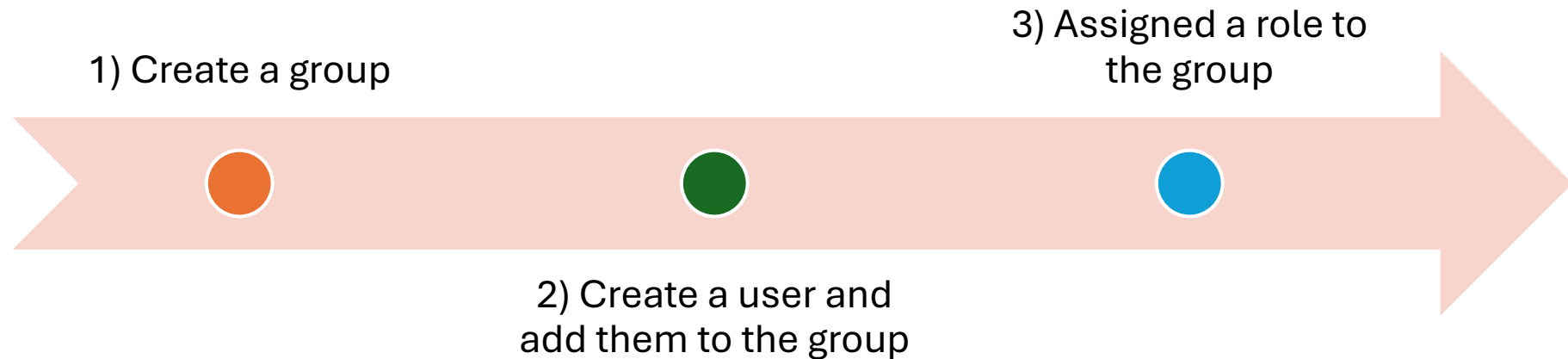
To design and deploy a microservices-based tracking application on AWS Elastic Kubernetes Service (EKS) utilizing Docker for containerization, Vault for secrets management, and IAM for access control, achieving a secure, scalable, and efficient deployment pipeline.

Tools & Technologies Used

- Cloud Platform: AWS (EKS, IAM, EC2)
- Containerization: Docker
- Orchestration: Kubernetes (EKS)
- Secret Management: HashiCorp Vault
- Version Control: Git & GitHub
- Operating System: Linux (RedHat)
- Build & Package Management: Maven
- Frontend Framework: Angular

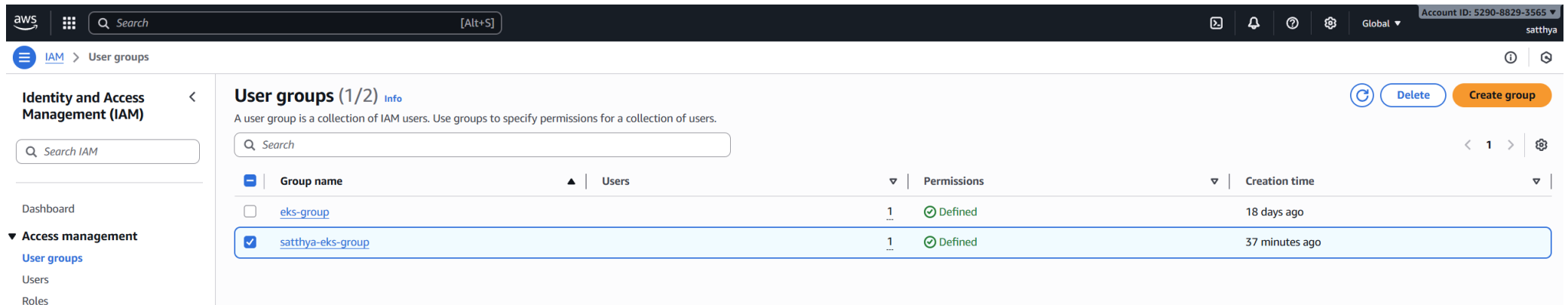
IAM

- Implemented AWS IAM roles and policies to manage access between EKS, Vault, and other AWS services, enabling secure deployment and operation of the microservices tracking application



IAM

- Created a group.



The screenshot shows the AWS IAM console interface. The top navigation bar includes the AWS logo, a search bar, and account information (Account ID: 5290-8829-3565, satthya). The left sidebar shows the 'Identity and Access Management (IAM)' section with a search bar and navigation links for 'User groups', 'Users', and 'Roles'. The main content area is titled 'User groups (1/2)' and includes a description: 'A user group is a collection of IAM users. Use groups to specify permissions for a collection of users.' Below this is a table of user groups.

Group name	Users	Permissions	Creation time
<input type="checkbox"/> eks-group	1	Defined	18 days ago
<input checked="" type="checkbox"/> satthya-eks-group	1	Defined	37 minutes ago

IAM

- Created a user and added them to the group.

The screenshot displays the AWS IAM console interface. On the left, the navigation pane shows 'Identity and Access Management (IAM)' with a search bar and a list of options including Dashboard, Access management (expanded), User groups (selected), Users, Roles, Policies, Identity providers, Account settings, Root access management, Access reports, and Access Analyzer. The main content area is titled 'satthya-eks-group' with an 'Info' link and 'Delete' and 'Edit' buttons. Below this is a 'Summary' section with three columns: 'User group name' (satthya-eks-group), 'Creation time' (October 15, 2025, 14:33 (UTC+08:00)), and 'ARN' (arn:aws:iam::529088293565:group/satthya-eks-group). A tabbed interface below the summary shows 'Users (1)', 'Permissions', and 'Access Advisor'. The 'Users (1)' tab is active, showing a table of users in the group. The table has columns for selection, User name, Groups, Last activity, and Creation time. One user, 'satthya-tracking-user', is listed with 1 group, no last activity, and a creation time of 'Now'. There are 'Remove' and 'Add users' buttons at the top right of the users table.

Identity and Access Management (IAM) <

Q Search IAM

Dashboard

▼ Access management

User groups

Users

Roles

Policies

Identity providers

Account settings

Root access management

▼ Access reports

Access Analyzer

Resource analysis New

Unused access

Analyzer settings

satthya-eks-group Info Delete Edit

Summary

User group name
satthya-eks-group

Creation time
October 15, 2025, 14:33 (UTC+08:00)

ARN
 arn:aws:iam::529088293565:group/satthya-eks-group

Users (1) | Permissions | Access Advisor

Users in this group (1) Remove Add users

An IAM user is an entity that you create in AWS to represent the person or application that uses it to interact with AWS.

Q Search

<input type="checkbox"/>	User name	▲ Groups	Last activity ▼	Creation time ▼
<input type="checkbox"/>	satthya-tracking-user	1	None	Now

IAM

- Added a policy to the group. Any user in the group gets the same access.

The screenshot displays the AWS IAM console interface for the 'satthya-eks-group'. The left sidebar shows the navigation menu with 'Identity and Access Management (IAM)' selected. The main content area shows the 'satthya-eks-group' details, including a summary of the user group name, creation time, and ARN. Below this, the 'Permissions' tab is active, showing a list of four attached policies: AmazonEC2FullAccess, AmazonEKSLoadBalancingPolicy, AWSCloudFormationFullAccess, and IAMFullAccess. The table lists the policy name, type (AWS managed), and the number of attached entities (2 for each).

Identity and Access Management (IAM)

Search IAM

Dashboard

▼ Access management

- User groups
- Users
- Roles
- Policies
- Identity providers
- Account settings
- Root access management

▼ Access reports

- Access Analyzer
- Resource analysis [New](#)
- Unused access
- Analyzer settings
- Credential report
- Organization activity
- Service control policies
- Resource control policies

satthya-eks-group Info

Summary

User group name: satthya-eks-group

Creation time: October 15, 2025, 14:33 (UTC+08:00)

ARN: arn:aws:iam::529088293565:group/satthya-eks-group

Users (1) | **Permissions** | Access Advisor

Permissions policies (4) Info

You can attach up to 10 managed policies.

Filter by Type: All types

Policy name	Type	Attached entities
<input type="checkbox"/> AmazonEC2FullAccess	AWS managed	2
<input type="checkbox"/> AmazonEKSLoadBalancingPolicy	AWS managed	3
<input type="checkbox"/> AWSCloudFormationFullAccess	AWS managed	2
<input type="checkbox"/> IAMFullAccess	AWS managed	2

IAM

- Access keys are created for the user to enable secure programmatic access, which is used for automated deployment of the application

The screenshot shows the AWS IAM console interface for creating an access key for the user 'satthya-tracking-user'. The breadcrumb navigation at the top reads: IAM > Users > satthya-tracking-user > Create access key. A green warning banner at the top states: 'This is the only time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time.' On the left, a step indicator shows three steps: 'Step 1: Access key best practices & alternatives', 'Step 2 - optional: Set description tag', and 'Step 3: Retrieve access keys' (which is the active step). The main content area is titled 'Retrieve access keys' with an 'Info' link. It contains two sections: 'Access key' and 'Access key best practices'. The 'Access key' section includes a warning: 'If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.' Below this, there are two fields: 'Access key' with the value 'AKIAIWMA6VK67B7Q3YAA' and 'Secret access key' with a masked value '*****' and a 'Show' link. The 'Access key best practices' section lists four bullet points: 'Never store your access key in plain text, in a code repository, or in code.', 'Disable or delete access key when no longer needed.', 'Enable least-privilege permissions.', and 'Rotate access keys regularly.' It also includes a link to 'best practices for managing AWS access keys'. At the bottom right, there are two buttons: 'Download .csv file' and 'Done'.

IAM > Users > satthya-tracking-user > Create access key

✔ This is the only time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time.

Step 1
● Access key best practices & alternatives

Step 2 - optional
● Set description tag

Step 3
● **Retrieve access keys**

Retrieve access keys [Info](#)

Access key

If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.

Access key	Secret access key
AKIAIWMA6VK67B7Q3YAA	***** Show

Access key best practices

- Never store your access key in plain text, in a code repository, or in code.
- Disable or delete access key when no longer needed.
- Enable least-privilege permissions.
- Rotate access keys regularly.

For more details about managing access keys, see the [best practices for managing AWS access keys](#).

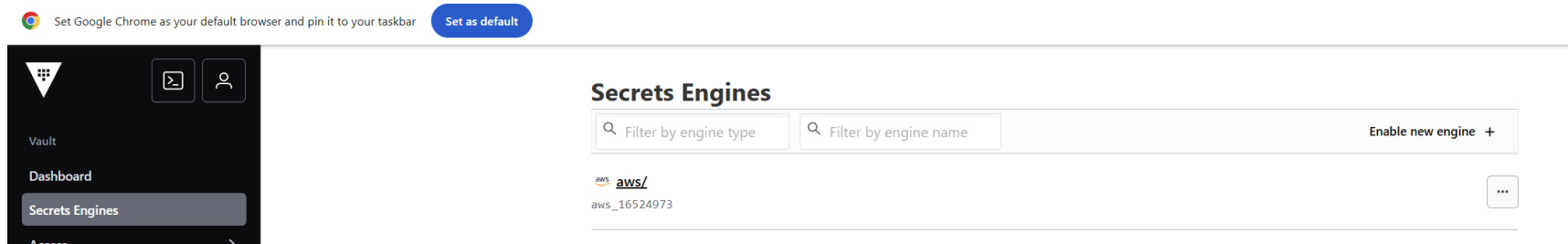
[Download .csv file](#) [Done](#)

VAULT CONFIGURATION

- Store the AWS Access Key ID and Secret Access Key in Vault
 - Enable the AWS secrets engine in Vault by running the command below

❏ vault secrets enable aws

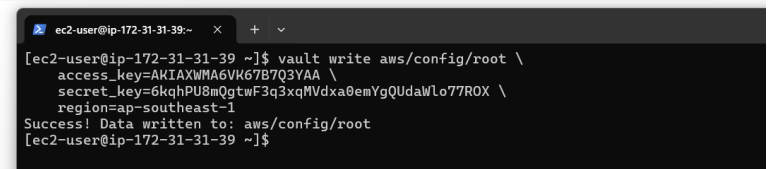
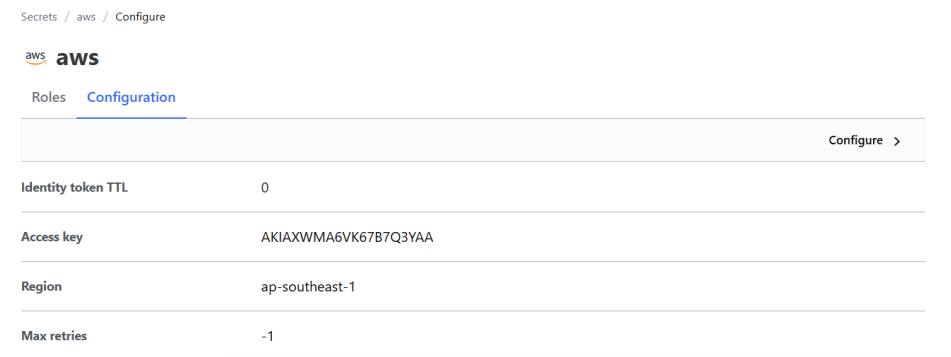
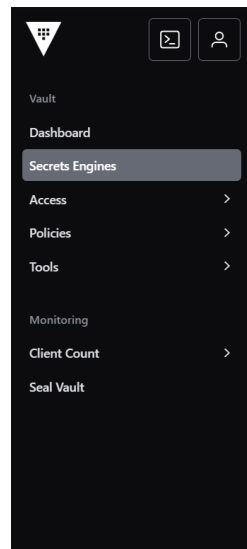
```
ec2-user@ip-172-31-31-39:~$ vault secrets enable aws
Success! Enabled the aws secrets engine at: aws/
ec2-user@ip-172-31-31-39:~$ vault secrets list
Path      Type      Accessor      Description
----      -
GitHub/   kv        kv_5dbe8771    n/a
aws/      aws       aws_b5335c4e   n/a
cubbyhole/ cubbyhole cubbyhole_728bab33 per-token private secret storage
identity/  identity  identity_0b4377d8 identity store
secrets/   kv        kv_aa499a1c    n/a
sys/      system    system_27514caa system endpoints used for control, policy and debugging
ec2-user@ip-172-31-31-39:~$
```



VAULT CONFIGURATION

- Store the AWS Access Key ID and Secret Access Key in Vault
 - Configure the AWS secrets engine in Vault.

```
vault write aws/config/root \  
  access_key=NLNA \  
  secret_key=R4nm063hg \  
  region=ap-southeast-1
```



VAULT CONFIGURATION

- Store the AWS Access Key ID and Secret Access Key in Vault

➤ Rotate AWS Root Credentials in Vault.

❏ vault write -f aws/config/rotate-root

```
ec2-user@ip-172-31-31-39:~$ vault read /aws/config/root
Key
-----
access_key      AKIAXWMA6VK67B7Q3YAA
disable_automated_rotation  false
iam_endpoint    n/a
identity_token_audience    n/a
identity_token_ttl    0s
max_retries      -1
region           ap-southeast-1
role_arn          n/a
rotation_period   0s
rotation_schedule n/a
rotation_window   0
sts_endpoint      n/a
sts_fallback_endpoints []
sts_fallback_regions []
sts_region        n/a
username_template {{ if (eq .Type "STS") }}{{ printf "vault-%s-%s" (unix_time) (random 32) | truncate 32 }}
{{ else }}{{ printf "vault-%s-%s-%s" (printf "%s-%s" (.DisplayName) (.PolicyName) | truncate 42) (unix_time) (random 20)
| truncate 64 }}{{ end }}
[ec2-user@ip-172-31-31-39 ~]$ vault write -f aws/config/rotate-root
Key
-----
access_key      AKIAXWMA6VK67UQAC7UR
[ec2-user@ip-172-31-31-39 ~]$
```

Access keys (1)

Use access keys to send programmatic calls to AWS from the AWS CLI, AWS Tools for PowerShell, AWS SDKs, or direct AWS API calls. You can have a maximum of two access keys (active or inactive) at a time. [Learn more](#)

Create access key

AKIAXWMA6VK67UQAC7UR

Description

-

Last used

None

Last used region

None

Status

Active

Created

1 minute ago

Last used service

None

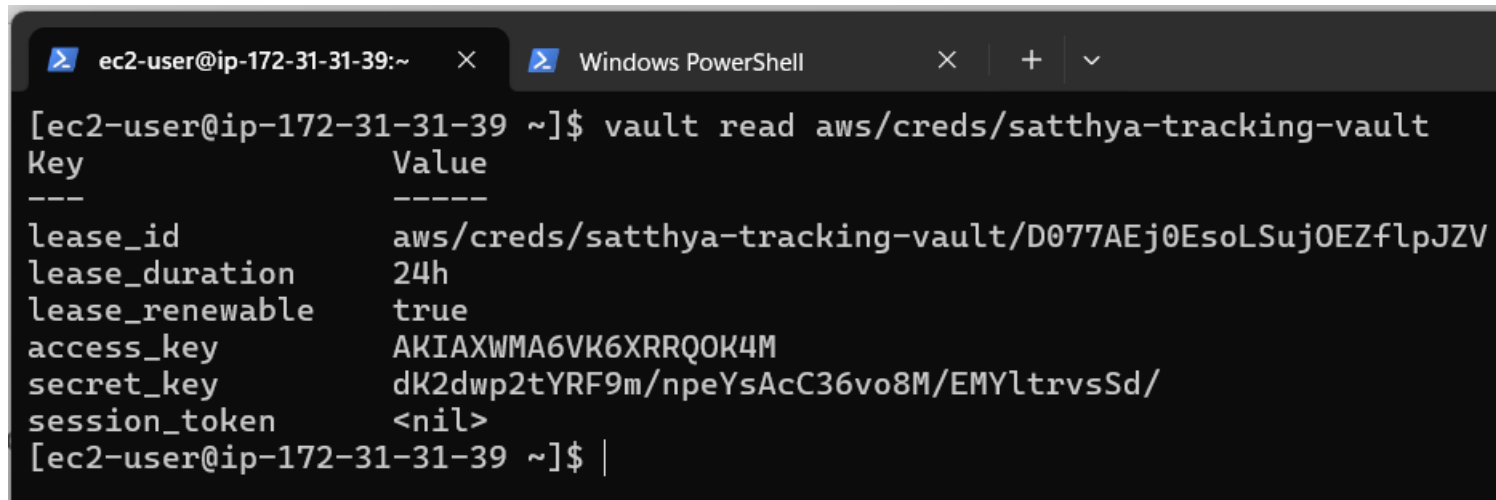
Actions

VAULT CONFIGURATION

- Store the AWS Access Key ID and Secret Access Key in Vault

➤ Read AWS Credentials via Vault.

❏ `vault read aws/creds/satthya-tracking-vault`



```
ec2-user@ip-172-31-31-39:~  Windows PowerShell
[ec2-user@ip-172-31-31-39 ~]$ vault read aws/creds/satthya-tracking-vault
Key          Value
---          -
lease_id     aws/creds/satthya-tracking-vault/D077AEj0EsoLSuj0EZflpJZV
lease_duration 24h
lease_renewable true
access_key    AKIAXWMA6VK6XRRQOK4M
secret_key    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
session_token <nil>
[ec2-user@ip-172-31-31-39 ~]$ |
```

IAM

- Created an IAM role to be attached to an EC2 instance for serving as the master node in an EKS cluster.

The screenshot displays the AWS IAM console interface for the 'satthya-EC2-EKS' role. The left sidebar shows the navigation menu with 'Access management' and 'Access reports' sections. The main content area shows the role's summary, including its creation date, ARN, and instance profile ARN. Below the summary, the 'Permissions' tab is active, showing a list of four attached managed policies.

Role satthya-EC2-EKS created. [View role] [Delete]

satthya-EC2-EKS [info]
Allows EC2 instances to call AWS services on your behalf. [Delete]

Summary [Edit]

Creation date	ARN	Instance profile ARN
October 15, 2025, 15:43 (UTC+08:00)	arn:aws:iam::529088293565:role/satthya-EC2-EKS	arn:aws:iam::529088293565:instance-profile/satthya-EC2-EKS

Last activity
-

Maximum session duration
1 hour

Permissions | Trust relationships | Tags | Last Accessed | Revoke sessions

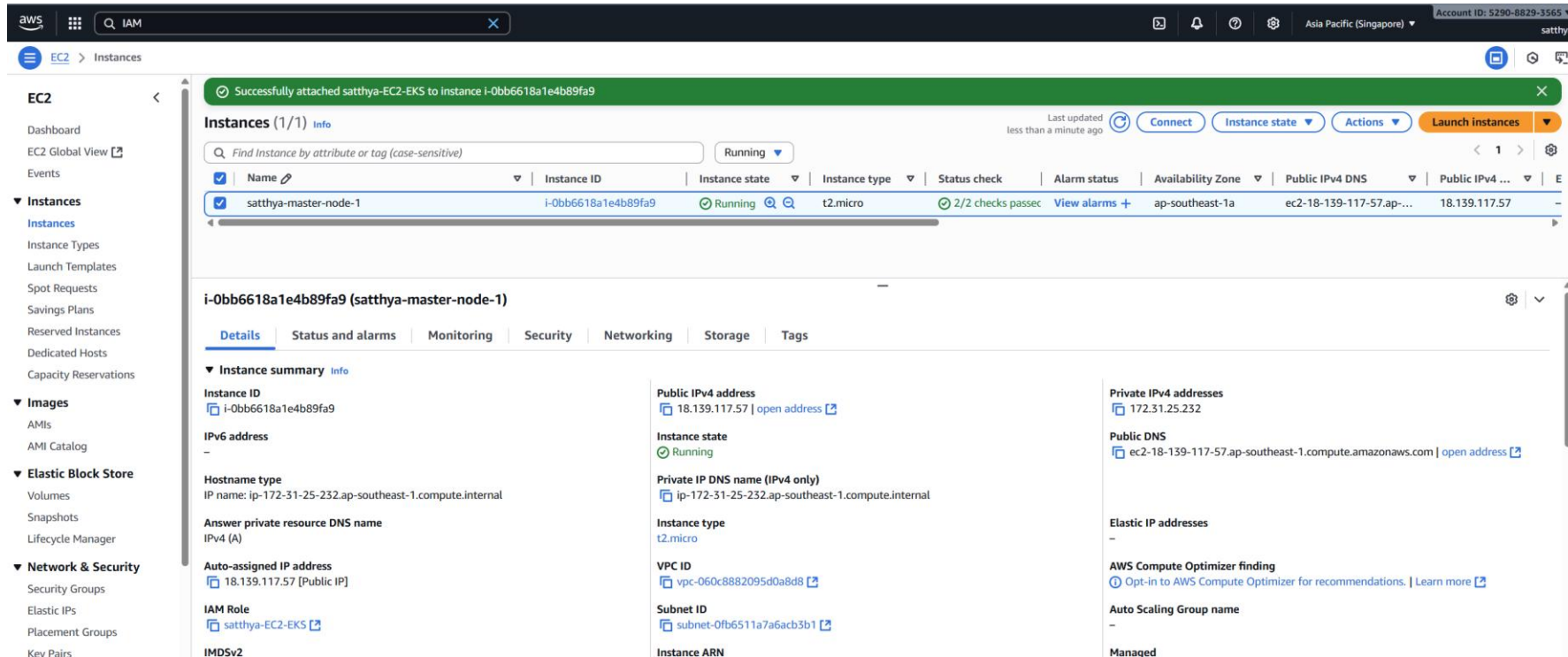
Permissions policies (4) [info]
You can attach up to 10 managed policies. [Simulate] [Remove] [Add permissions]

Search [Filter by Type: All types]

Policy name	Type	Attached entities
<input type="checkbox"/> AmazonEC2ContainerRegistryReadOnly	AWS managed	2
<input type="checkbox"/> AmazonEKS_CNI_Policy	AWS managed	3
<input type="checkbox"/> AmazonEKSClusterPolicy	AWS managed	3
<input type="checkbox"/> AmazonEKSWorkerNodePolicy	AWS managed	3

EKS Cluster Master Node on EC2

- Deployed an EC2 instance as the master node for an AWS EKS cluster and attached a custom IAM role to manage cluster resources and integrate with other AWS services.



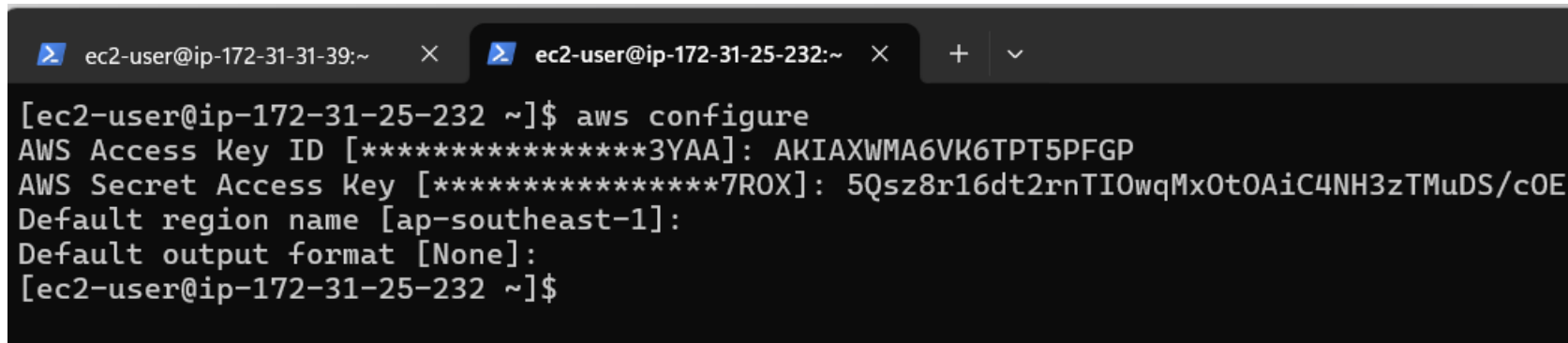
EKS Cluster Master Node on EC2

- Installed AWS CLI, eksctl, and kubectl on the master node.

```
ec2-user@ip-172-31-25-232:~ × + v
[ec2-user@ip-172-31-25-232 ~]$ aws --version
aws-cli/2.31.15 Python/3.13.7 Linux/6.12.0-55.25.1.el10_0.x86_64 exe/x86_64.rhel.10
[ec2-user@ip-172-31-25-232 ~]$ kubectl version --client
Client Version: v1.34.1
Kustomize Version: v5.7.1
[ec2-user@ip-172-31-25-232 ~]$ eksctl version
0.215.0
[ec2-user@ip-172-31-25-232 ~]$
```


EKS Cluster Master Node on EC2

- Configured AWS CLI on an EC2 instance as a control node to manage AWS resources and Kubernetes clusters using dynamic AWS credentials retrieved from Vault.

A terminal window with two tabs. The active tab is titled 'ec2-user@ip-172-31-25-232:~'. The terminal shows the command 'aws configure' being executed, followed by prompts for AWS Access Key ID, AWS Secret Access Key, Default region name, and Default output format. The user has entered masked values for the keys and 'ap-southeast-1' for the region.

```
ec2-user@ip-172-31-31-39:~ × ec2-user@ip-172-31-25-232:~ × + v
[ec2-user@ip-172-31-25-232 ~]$ aws configure
AWS Access Key ID [*****3YAA]: AKIAXWMA6VK6TPT5PFGP
AWS Secret Access Key [*****7ROX]: 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
Default region name [ap-southeast-1]:
Default output format [None]:
[ec2-user@ip-172-31-25-232 ~]$
```

EKS Cluster Setup with Worker Nodes

- Created an EKS cluster satthya-trackmate with 3 t2.medium worker nodes for running Kubernetes workloads.

➤ eksctl create cluster \
--name satthya-trackmate \
--nodes-min=3 \
--node-type t2.medium

```
[ec2-user@ip-172-31-25-232 ~]$ eksctl create cluster \  
--name satthya-trackmate \  
--nodes-min=3 \  
--node-type t2.medium  
2025-10-15 09:15:00 [i] eksctl version 0.215.0  
2025-10-15 09:15:00 [i] using region ap-southeast-1  
2025-10-15 09:15:00 [i] setting availability zones to [ap-southeast-1b ap-southeast-1a ap-southeast-1c]  
2025-10-15 09:15:00 [i] subnets for ap-southeast-1b - public:192.168.0.0/19 private:192.168.0.0/19  
2025-10-15 09:15:00 [i] subnets for ap-southeast-1a - public:192.168.32.0/19 private:192.168.32.0/19  
2025-10-15 09:15:00 [i] subnets for ap-southeast-1c - public:192.168.64.0/19 private:192.168.64.0/19  
2025-10-15 09:15:00 [i] nodegroup "ng-01eabe70" will use "" [AmazonLinux2023/1.32]  
2025-10-15 09:15:00 [!] Auto Mode will be enabled by default in an upcoming release of eksctl. To opt-out of this  
current behavior, explicitly set 'autoModeConfig.enabled: false' in your cluster configuration.  
2025-10-15 09:15:00 [i] using Kubernetes version 1.32  
2025-10-15 09:15:00 [i] creating EKS cluster "satthya-trackmate" in "ap-southeast-1" region  
2025-10-15 09:15:00 [i] will create 2 separate CloudFormation stacks for cluster itself and nodegroup  
2025-10-15 09:15:00 [i] if you encounter any issues, check CloudFormation console or try  
2025-10-15 09:15:00 [i] Kubernetes API endpoint access will use default of {publicAccess=true} for this cluster  
2025-10-15 09:15:00 [i] CloudWatch logging will not be enabled for cluster "satthya-trackmate" - you can enable it with 'eksctl utils update-cluster-logging --enable  
2025-10-15 09:15:00 [i] default addons vpc-cni, kube-proxy, coredns, metrics-server were installed  
2025-10-15 09:15:00 [i]  
2 sequential tasks: { create cluster control plane "satthya-trackmate",  
  2 sequential sub-tasks: {  
    2 sequential sub-tasks: {  
      1 task: { create addons },  
      wait for control plane to become ready,  
    },  
    create managed nodegroup "ng-01eabe70",  
  },  
}  
2025-10-15 09:15:00 [i] building cluster stack "eksctl-satthya-trackmate-cluster"  
2025-10-15 09:15:00 [i] deploying stack "eksctl-satthya-trackmate-cluster"  
2025-10-15 09:15:30 [i] waiting for CloudFormation stack "eksctl-satthya-trackmate-cluster"  
2025-10-15 09:16:00 [i] waiting for CloudFormation stack "eksctl-satthya-trackmate-cluster"  
2025-10-15 09:17:01 [i] waiting for CloudFormation stack "eksctl-satthya-trackmate-cluster"  
2025-10-15 09:18:01 [i] waiting for CloudFormation stack "eksctl-satthya-trackmate-cluster"  
2025-10-15 09:19:01 [i] waiting for CloudFormation stack "eksctl-satthya-trackmate-cluster"  
2025-10-15 09:20:01 [i] waiting for CloudFormation stack "eksctl-satthya-trackmate-cluster"
```

EKS Cluster Deployment on AWS

- Amazon EKS cluster satthya-trackmate created in ap-southeast-1, ready to manage Kubernetes workloads.

The screenshot displays the Amazon Elastic Kubernetes Service (EKS) console for the 'satthya-trackmate' cluster in the ap-southeast-1 region. The interface includes a left-hand navigation menu with options like 'Dashboard', 'Clusters', 'Settings', 'Amazon EKS Anywhere', and 'Related services'. The main content area shows the cluster's status as 'Active' and provides a summary of its configuration.

Cluster info

Cluster info	Kubernetes version	Support period	Provider
Status: Active	1.32	Standard support until March 23, 2026	EKS

Cluster health

Cluster health	Upgrade insights	Node health issues
0	0	0

Details

API server endpoint	OpenID Connect provider URL	Created
https://9AA40B18C087F633E80456FC5B1ABB07.y4.ap-southeast-1.eks.amazonaws.com	https://oidc.eks.ap-southeast-1.amazonaws.com/id/9AA40B18C087F633E80456FC5B1ABB07	18 minutes ago

Certificate authority

LS0tLS1CRUdJTIBDRVJUSUZJQ0FUR50tLS0tck1JSURCVENDQWUyZ0F3SUJBZ0Utm9Mc0YvemJucDh3RFFZSkVWklodmNOQVFFTEJRQXdGVEVUTUJFR0ExVUUKQXhNS2EzVmlaWEp1WlhSbGN6QWVGd

Cluster IAM role ARN

[arn:aws:iam::529088293565:role/eksctl-satthya-trackmate-cluster-ServiceRole-tN4Cztmg14X](#)

Cluster ARN

[arn:aws:eks:ap-southeast-1:529088293565:cluster/satthya-trackmate](#)

Platform version

eks.25

EKS Cluster Deployment on AWS

- Once the eksctl command ran, it provisioned 3 worker nodes in the cluster

✓ Successfully attached satthya-EC2-EKS to instance i-0bb6618a1e4b89fa9

Instances (4) [Info](#)

Last updated 10 minutes ago

Connect

Instance state ▾

Actions ▾

Launch instances ▾

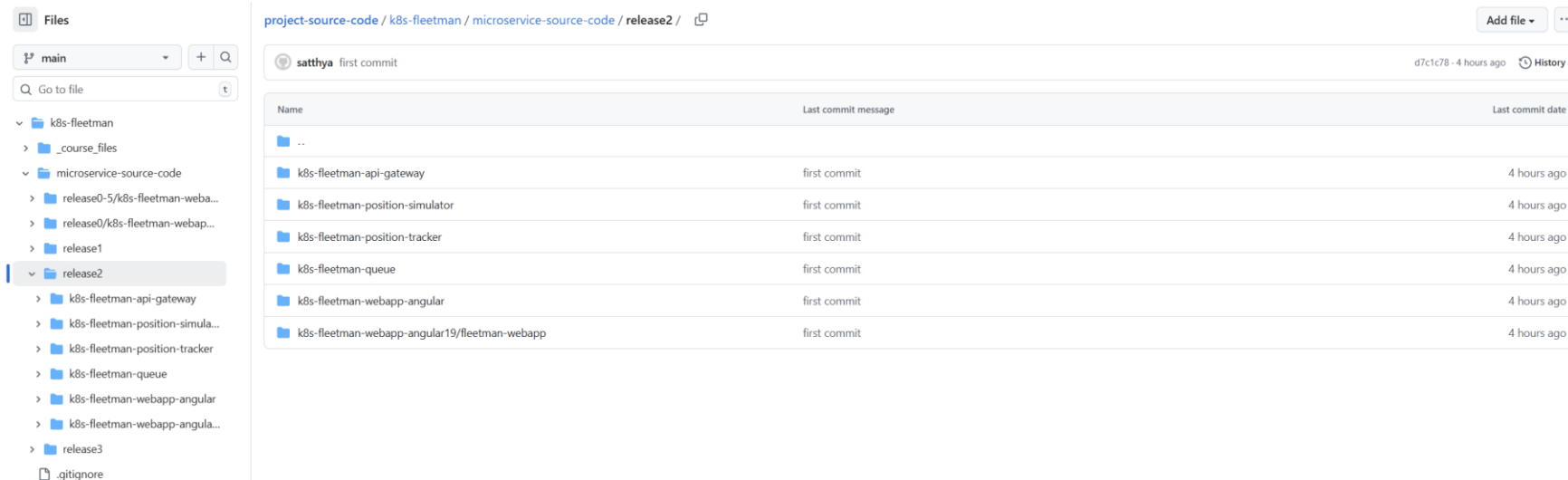
Running ▾

<input type="checkbox"/>	Name	Instance ID	Instance state ▾	Instance type ▾	Status check	Alarm status	Availability Zone ▾	Public IPv4 DNS ▾	Public IPv4 ... ▾	E
<input type="checkbox"/>	satthya-trackmate-ng-01eabe70-Node	i-08bf2a4292959e452	✓ Running	t2.medium	✓ 2/2 checks passec	View alarms +	ap-southeast-1c	ec2-18-138-51-205.ap-...	18.138.51.205	-
<input type="checkbox"/>	satthya-trackmate-ng-01eabe70-Node	i-07ef15095abfba4b6	✓ Running	t2.medium	✓ 2/2 checks passec	View alarms +	ap-southeast-1b	ec2-47-129-227-82.ap-...	47.129.227.82	-
<input type="checkbox"/>	satthya-master-node-1	i-0bb6618a1e4b89fa9	✓ Running	t2.micro	✓ 2/2 checks passec	View alarms +	ap-southeast-1a	ec2-18-139-117-57.ap-...	18.139.117.57	-
<input type="checkbox"/>	satthya-trackmate-ng-01eabe70-Node	i-0891a4adf27d444b7	✓ Running	t2.medium	✓ 2/2 checks passec	View alarms +	ap-southeast-1a	ec2-18-136-104-198.ap...	18.136.104.198	-

Download Source Code

- The project is on GitHub. Use git clone to download a complete copy locally, including all files and history, ready for building and running.

➤ `git clone https://github.com/satthya/project-source-code.git`



The image shows a side-by-side comparison of a local file explorer and a GitHub commit history page. On the left, a file explorer window displays a directory structure with folders like 'k8s-fleetman', 'microservice-source-code', and 'release2'. On the right, a GitHub commit history page for the 'project-source-code' repository shows a list of commits, all labeled 'first commit', with a table of commit details.

Name	Last commit message	Last commit date
..		
k8s-fleetman-api-gateway	first commit	4 hours ago
k8s-fleetman-position-simulator	first commit	4 hours ago
k8s-fleetman-position-tracker	first commit	4 hours ago
k8s-fleetman-queue	first commit	4 hours ago
k8s-fleetman-webapp-angular	first commit	4 hours ago
k8s-fleetman-webapp-angular19/fleetman-webapp	first commit	4 hours ago

Tracking App Build

- Build the Maven project, build the Angular front-end application, and create a Docker image for each microservice.

➤ Backend (Maven) :
mvn clean package

Run this in each microservice
directory if a pom.xml file exists.

This will create war/jar artifact

```
[satthya@localhost k8s-fleetman-api-gateway]$ ll
total 32
-rw-r--r--. 1 satthya satthya 251 Oct 19 11:31 Dockerfile
-rw-r--r--. 1 satthya satthya 1083 Oct 19 11:31 LICENSE
-rwxr-xr-x. 1 satthya satthya 7058 Oct 19 11:31 mvnw
-rw-r--r--. 1 satthya satthya 5006 Oct 19 11:31 mvnw.cmd
-rw-r--r--. 1 satthya satthya 2277 Oct 19 11:31 pom.xml
-rw-r--r--. 1 satthya satthya 247 Oct 19 11:31 README.md
drwxr-xr-x. 4 satthya satthya 30 Oct 19 11:31 src
[satthya@localhost k8s-fleetman-api-gateway]$ mvn clean package
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.virtualpairprogrammers:fleetman >-----
[INFO] Building fleetman 0.0.1-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- maven-clean-plugin:3.2.0:clean (default-clean) @ fleetman ---
[INFO]
[INFO] --- maven-resources-plugin:3.3.0:resources (default-resources) @ fleetman ---
[INFO] Copying 3 resources
[INFO]
[INFO] --- maven-compiler-plugin:3.10.1:compile (default-compile) @ fleetman ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 6 source files to /home/satthya/fleetman-project/project-source-code/k8s-fleetman-api-gateway/target/classes
```

Tracking App Build

- Build the Maven project, build the Angular front-end application, and create a Docker image for each microservice.

➤ Frontend (Angular) :
ng build

Run this in the front-end web directory.

Turns your Angular code into a deployable front-end application

```
14 vulnerabilities (6 low, 8 moderate)

To address all issues, run:
  npm audit fix

Run 'npm audit' for details.
[satthya@localhost fleetman-webapp]$ ng build
Initial chunk files | Names | Raw size | Estimated transfer size
main-G47HUJXS.js | main | 469.72 kB | 124.99 kB
styles-SDKLAYOD.css | styles | 242.92 kB | 24.68 kB
polyfills-EQXJKH7W.js | polyfills | 35.81 kB | 11.76 kB
| Initial total | 748.45 kB | 161.43 kB

Application bundle generation complete. [16.351 seconds]

▲ [WARNING] bundle initial exceeded maximum budget. Budget 500.00 kB was not met by 248.45 kB with a total of 748.45 kB.

▲ [WARNING] Module 'leaflet' used by 'src/app/map/map.component.ts' is not ESM
CommonJS or AMD dependencies can cause optimization bailouts.
For more information see: https://angular.dev/tools/cli/build#configuring-commonjs-dependencies

▲ [WARNING] 9 rules skipped due to selector errors:
.table>>>* -> Did not expect successive traversals.
.table-sm>>>* -> Did not expect successive traversals.
.table-bordered>>>* -> Did not expect successive traversals.
```

Tracking App Build

- Build the Maven project, build the Angular front-end application, and create a Docker image for each microservice.

➤ Docker image :

docker build -t "name"

docker push satthya04/fleetman-webapp .

Run this in each directory containing a Dockerfile, replace name with your image name, and push each image to Docker Hub

```
drwxr-xr-x. 564 satthya satthya 16384 Oct 19 14:38 node_modules
-rwxr-xr-x. 1 satthya satthya 1438 Oct 19 14:36 package.json
-rwxr-xr-x. 1 satthya satthya 528124 Oct 19 14:38 package-lock.json
drwxr-xr-x. 2 satthya satthya 4096 Oct 19 14:36 public
-rwxr-xr-x. 1 satthya satthya 1536 Oct 19 14:36 README.md
drwxr-xr-x. 3 satthya satthya 68 Oct 19 14:36 src
-rwxr-xr-x. 1 satthya satthya 472 Oct 19 14:36 tsconfig.app.json
-rwxr-xr-x. 1 satthya satthya 1003 Oct 19 14:36 tsconfig.json
-rwxr-xr-x. 1 satthya satthya 477 Oct 19 14:36 tsconfig.spec.json
[satthya@localhost fleetman-webapp]$ docker build -t fleetman-webapp .
[+] Building 2.7s (11/11) FINISHED
=> [internal] load build definition from Dockerfile                                docker:default 0.1s
=> => transferring dockerfile: 413B                                              0.0s
=> WARN: MaintainerDeprecated: Maintainer instruction is deprecated in favor of using label (line 3) 0.1s
=> [internal] load metadata for docker.io/library/nginx:1.14.0-alpine            2.0s
=> [auth] library/nginx:pull token for registry-1.docker.io                     0.0s
=> [internal] load .dockerignore                                                 0.1s
=> => transferring context: 113B                                                 0.0s
=> [1/5] FROM docker.io/library/nginx:1.14.0-alpine@sha256:8976218be775f4244df2a60a169d44606b6978bac4375192074ce 0.0s
=> [internal] load build context                                                0.1s
=> => transferring context: 905.57kB                                             0.1s
=> CACHED [2/5] RUN apk add --update bash && rm -rf /var/cache/apk/*             0.0s
=> CACHED [3/5] RUN rm -rf /usr/share/nginx/html/*                             0.0s
=> CACHED [4/5] COPY /dist/fleetman-webapp/browser/ /usr/share/nginx/html       0.0s
=> CACHED [5/5] COPY nginx.conf /etc/nginx/nginx.conf                          0.0s
=> exporting to image                                                           0.1s
=> exporting layers                                                             0.0s
=> writing image sha256:2f20f81eb1a63e82e0f83e7c46597fb5c553bc4ba891a55215c0546080a8700d 0.0s
=> naming to docker.io/library/fleetman-webapp                                0.0s
```


Tracking App Deployment

- Deployed a microservices tracking app with Docker & Kubernetes using YAML files and exposed the frontend via LoadBalancer.

```
[ec2-user@ip-172-31-25-232 tracking-app]$ cat deployment.yml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: fleetman-webapp
spec:
  replicas: 1
  selector:
    matchLabels:
      app: webapp
  template:
    metadata:
      labels:
        app: webapp
    spec:
      containers:
        - name: webapp
          image: sathya04/fleetman-webapp:release2
          env:
            - name: SPRING_PROFILES_ACTIVE
              value: production-microservice
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: fleetman-api-gateway
spec:
  replicas: 1
  selector:
    matchLabels:
      app: api-gateway
  template:
    metadata:
      labels:
        app: api-gateway
    spec:
      containers:
        - name: api-gateway
          image: sathya04/fleetman-api-gateway:latest
          env:
            - name: SPRING_PROFILES_ACTIVE
              value: production-microservice
```

```
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: fleetman-position-simulator
spec:
  replicas: 1
  selector:
    matchLabels:
      app: position-simulator
  template:
    metadata:
      labels:
        app: position-simulator
    spec:
      containers:
        - name: position-simulator
          image: sathya04/fleetman-position-simulator:latest
          env:
            - name: SPRING_PROFILES_ACTIVE
              value: production-microservice
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: fleetman-queue
spec:
  replicas: 1
  selector:
    matchLabels:
      app: queue
  template:
    metadata:
      labels:
        app: queue
    spec:
      containers:
        - name: queue
          image: sathya04/fleetman-queue:latest
```

```
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: fleetman-position-tracker
spec:
  replicas: 1
  selector:
    matchLabels:
      app: position-tracker
  template:
    metadata:
      labels:
        app: position-tracker
    spec:
      containers:
        - name: position-tracker
          image: sathya04/fleetman-position-tracker:latest
          env:
            - name: SPRING_PROFILES_ACTIVE
              value: production-microservice
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: mongodb
spec:
  replicas: 1
  selector:
    matchLabels:
      app: mongodb
  template:
    metadata:
      labels:
        app: mongodb
    spec:
      containers:
        - name: mongodb
          image: mongo:3.6.5-jessie
          volumeMounts:
            - name: mongo-persistent-storage
              mountPath: /data/db
          volumes:
            - name: mongo-persistent-storage
              persistentVolumeClaim:
                claimName: mongo-pvc
```

Tracking App Deployment

- Created a Service YAML file in Kubernetes to expose the microservices and route traffic to the Pods through a stable endpoint.

```
ec2-user@ip-172-31-25-232:~ X ec2
kind: Service
apiVersion: v1
metadata:
  name: fleetman-webapp
spec:
  ports:
    - name: webapp
      port: 80

  selector:
    app: webapp

  type: LoadBalancer
---
```

```
kind: Service
apiVersion: v1
metadata:
  name: fleetman-api-gateway
spec:
  ports:
    - name: api-gateway
      port: 8080

  selector:
    app: api-gateway

  type: ClusterIP
```

```
---
kind: Service
apiVersion: v1
metadata:
  name: fleetman-queue
spec:
  ports:
    - name: queue
      port: 8161

    - name: endpoint
      port: 61616

  selector:
    app: queue

  type: ClusterIP
---
```

```
kind: Service
apiVersion: v1
metadata:
  name: fleetman-position-tracker
spec:
  ports:
    - name: position-tracker
      port: 8080

  selector:
    app: position-tracker

  type: ClusterIP
```

```
kind: Service
apiVersion: v1
metadata:
  name: fleetman-mongodb
spec:
  selector:
    app: mongodb
  ports:
    - name: mongoport
      port: 27017

  type: ClusterIP
```

Tracking App Deployment

- Created a Storage YAML file in Kubernetes to provide persistent storage for the database, ensuring data remains intact even if the database Pod restarts or is replaced.

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: mongo-pvc
spec:
  storageClassName: cloud-ssd
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 7Gi
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: cloud-ssd
provisioner: kubernetes.io/aws-ebs
parameters:
  type: gp2
~
```

Tracking App Deployment

- Deploy the YAML file using the command below:
 - `kubectl apply -f deployment.yml`
 - `kubectl apply -f service.yml`
 - `kubectl apply -f storage.yml`

```
[ec2-user@ip-172-31-25-232 tracking-app]$ ll
total 12
-rw-r--r--. 1 ec2-user ec2-user 2479 Oct 24 13:21 deployment.yml
-rw-r--r--. 1 ec2-user ec2-user  957 Oct 24 13:25 service.yml
-rw-r--r--. 1 ec2-user ec2-user  330 Oct 24 13:29 storage-aws.yml
[ec2-user@ip-172-31-25-232 tracking-app]$ kubectl apply -f deployment.yml
deployment.apps/fleetman-webapp created
deployment.apps/fleetman-api-gateway created
deployment.apps/fleetman-position-simulator created
deployment.apps/fleetman-queue created
deployment.apps/fleetman-position-tracker created
error: error parsing deployment.yml: error converting YAML to JSON: yaml: line 19: found a tab character that violates indentation
[ec2-user@ip-172-31-25-232 tracking-app]$ kubectl apply -f service.yml
service/fleetman-webapp created
service/fleetman-api-gateway created
service/fleetman-queue created
service/fleetman-position-tracker created
service/fleetman-mongodb created
[ec2-user@ip-172-31-25-232 tracking-app]$ kubectl apply -f storage-aws.yml
persistentvolumeclaim/mongo-pvc created
storageclass.storage.k8s.io/cloud-ssd created
[ec2-user@ip-172-31-25-232 tracking-app]$
```

Tracking App Deployment

- View all Kubernetes resources in the current namespace, including Pods, Services, Deployments, and ReplicaSets.

```
[ec2-user@ip-172-31-25-232 tracking-app]$ kubectl get all
```

NAME	READY	STATUS	RESTARTS	AGE
pod/fleetman-api-gateway-9d595b467-7npkg	1/1	Running	0	18m
pod/fleetman-position-simulator-7bcf87d74b-hhktm	1/1	Running	0	18m
pod/fleetman-position-tracker-5b5dc8bd88-6298z	1/1	Running	0	18m
pod/fleetman-queue-69c5d887cc-n6llq	1/1	Running	0	18m
pod/fleetman-webapp-77c9f67b8f-x7jjx	1/1	Running	0	18m
pod/mongodb-5979dc7894-fcv7w	1/1	Running	0	21s

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/fleetman-api-gateway	ClusterIP	10.100.224.210	<none>	8080/TCP	18m
service/fleetman-mongodb	ClusterIP	10.100.115.51	<none>	27017/TCP	18m
service/fleetman-position-tracker	ClusterIP	10.100.241.238	<none>	8080/TCP	18m
service/fleetman-queue	ClusterIP	10.100.121.129	<none>	8161/TCP, 61616/TCP	18m
service/fleetman-webapp	LoadBalancer	10.100.144.201	a105c8f15bd324f13b06f85e6e7802ee-361513463.ap-southeast-1.elb.amazonaws.com	80:32054/TCP	18m
service/kubernetes	ClusterIP	10.100.0.1	<none>	443/TCP	28m

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
deployment.apps/fleetman-api-gateway	1/1	1	1	18m
deployment.apps/fleetman-position-simulator	1/1	1	1	18m
deployment.apps/fleetman-position-tracker	1/1	1	1	18m
deployment.apps/fleetman-queue	1/1	1	1	18m
deployment.apps/fleetman-webapp	1/1	1	1	18m
deployment.apps/mongodb	1/1	1	1	21s

NAME	DESIRED	CURRENT	READY	AGE
replicaset.apps/fleetman-api-gateway-9d595b467	1	1	1	18m
replicaset.apps/fleetman-position-simulator-7bcf87d74b	1	1	1	18m
replicaset.apps/fleetman-position-tracker-5b5dc8bd88	1	1	1	18m
replicaset.apps/fleetman-queue-69c5d887cc	1	1	1	18m
replicaset.apps/fleetman-webapp-77c9f67b8f	1	1	1	18m
replicaset.apps/mongodb-5979dc7894	1	1	1	21s

```
[ec2-user@ip-172-31-25-232 tracking-app]$
```

Tracking App UI

