

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

Objective

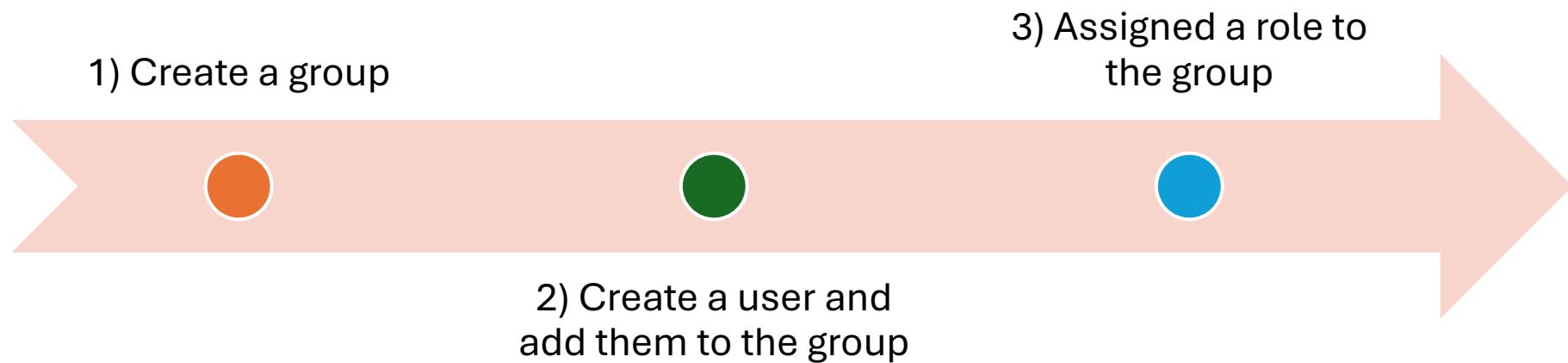
To design and deploy a microservices-based tracking application on AWS Elastic Kubernetes Service (EKS) utilizing Docker for containerization, Vault for secrets management, and IAM for access control, achieving a secure, scalable, and efficient deployment pipeline.

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

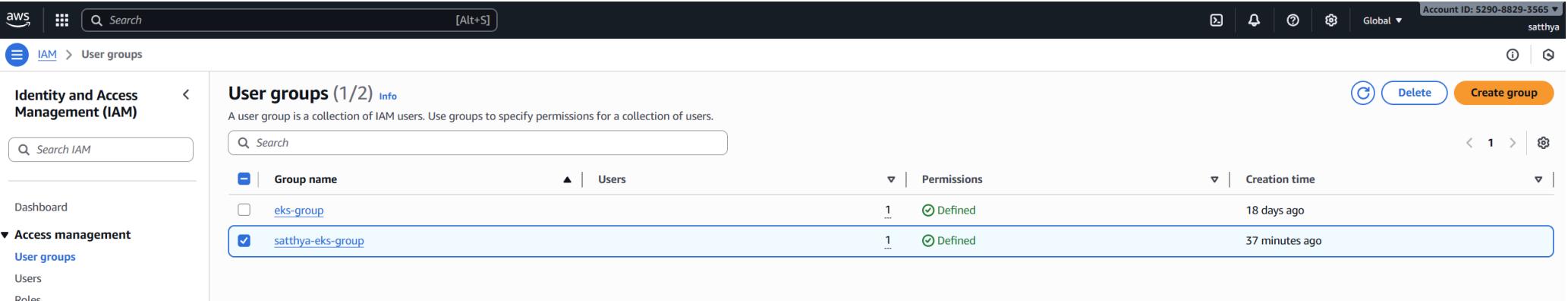
IAM

- 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 User groups page. The left sidebar has 'Identity and Access Management (IAM)' selected under 'Access management'. The main area displays 'User groups (1/2)'. A table lists two groups:

Group name	Users	Permissions	Creation time
eks-group	1	Defined	18 days ago
satthya-eks-group	1	Defined	37 minutes ago

IAM

- Created a user and added them to the group.

The screenshot shows the AWS IAM User Groups page. The top navigation bar includes links for IAM, User groups, and satthya-eks-group. The left sidebar has sections for Identity and Access Management (IAM) (Dashboard, Access management, User groups, Roles, Policies, Identity providers, Account settings, Root access management), Access reports (Access Analyzer, Resource analysis, Unused access, Analyzer settings), and a search bar. The main content area displays the 'satthya-eks-group' details:

- Summary**: User group name is satthya-eks-group. Creation time is October 15, 2025, 14:33 (UTC+08:00). ARN is arn:aws:iam::529088293565:group/satthya-eks-group.
- Users (1)**: Shows a single user named satthya-tracking-user. A 'Remove' button is available for this user.
- Permissions**: Not visible in the screenshot.
- Access Advisor**: Not visible in the screenshot.

Below the summary, there is a section titled 'Users in this group (1)' which lists the user satthya-tracking-user. The user details are as follows:

User name	Groups	Last activity	Creation time
satthya-tracking-user	1	None	Now

IAM

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

The screenshot shows the AWS IAM User Groups page. On the left, there's a navigation sidebar with options like Dashboard, Access management (User groups, Users, Roles, Policies, Identity providers, Account settings, Root access management), and Access reports (Access Analyzer, Resource analysis, Unused access, Analyzer settings, Credential report, Organization activity, Service control policies, Resource control policies). The main content area is titled "satthya-eks-group" and includes a "Summary" section with the user group name, creation time (October 15, 2025, 14:33 UTC+08:00), and ARN (arn:aws:iam::529088293565:group/satthya-eks-group). Below this, there are tabs for "Users (1)", "Permissions" (which is selected), and "Access Advisor". Under the "Permissions" tab, it says "Permissions policies (4)" and lists four AWS managed policies: AmazonEC2FullAccess, AmazonEKSLoadBalancingPolicy, AWSCloudFormationFullAccess, and IAMFullAccess, each with a count of 2 attached entities. There are buttons for "Simulate", "Remove", and "Add permissions".

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 'Create access key' wizard at Step 3: Retrieve access keys. The navigation bar at the top includes links for IAM, Users, sathya-tracking-user, and Create access key. A green 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." The main content area is titled 'Retrieve access keys' and contains the following sections:

- Access key**: A note stating: "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." It displays two fields: 'Access key' (AKIAJXWMA6VK67B7Q3YAA) and 'Secret access key' (*****). A 'Show' link is next to the secret key.
- Access key best practices**: A list of four items:
 - 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.A link to "best practices for managing AWS access keys" is provided.

At the bottom right are 'Download .csv file' and 'Done' buttons.

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 ~]$
```

The screenshot shows the HashiCorp Vault UI. On the left, there's a sidebar with icons for Vault, Dashboard, Secrets Engines (which is selected and highlighted in grey), and Access. The main area is titled "Secrets Engines" and contains two search bars: "Filter by engine type" and "Filter by engine name". Below these is a table with one row:

Type	Name	Description
aws	aws/	aws_16524973

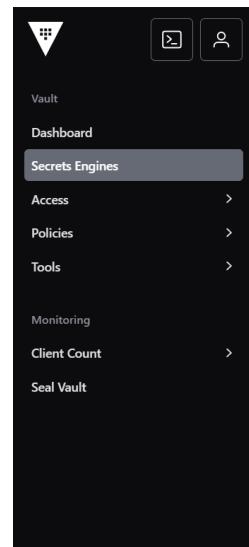
At the top right of the main area, there are buttons for "Enable new engine" and three dots for more options.

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
```



The screenshot shows the 'Configuration' tab for the AWS secrets engine. It includes fields for 'Identity token TTL' (set to 0), 'Access key' (containing the value 'AKIAJXWMA6VK67B7Q3YAA'), 'Region' (set to 'ap-southeast-1'), and 'Max retries' (set to -1). Below the configuration is a terminal window showing the command and its successful execution:

```
[ec2-user@ip-172-31-31-39 ~]$ vault write aws/config/root \
access_key=AKIAJXWMA6VK67B7Q3YAA \
secret_key=6kqhpPU8mQgtwF3q3xqMvdxa0emYgQUdaWlo77ROX \
region=ap-southeast-1
Success! Data written to: aws/config/root
[ec2-user@ip-172-31-31-39 ~]$
```

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          Value
access_key    AKIAIXWMA6VK67Q3YAA
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)}}{{ else }}{{ printf "vault-%s-%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          Value
access_key    AKIAIXWMA6VK67UQAC7UR
[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](#)

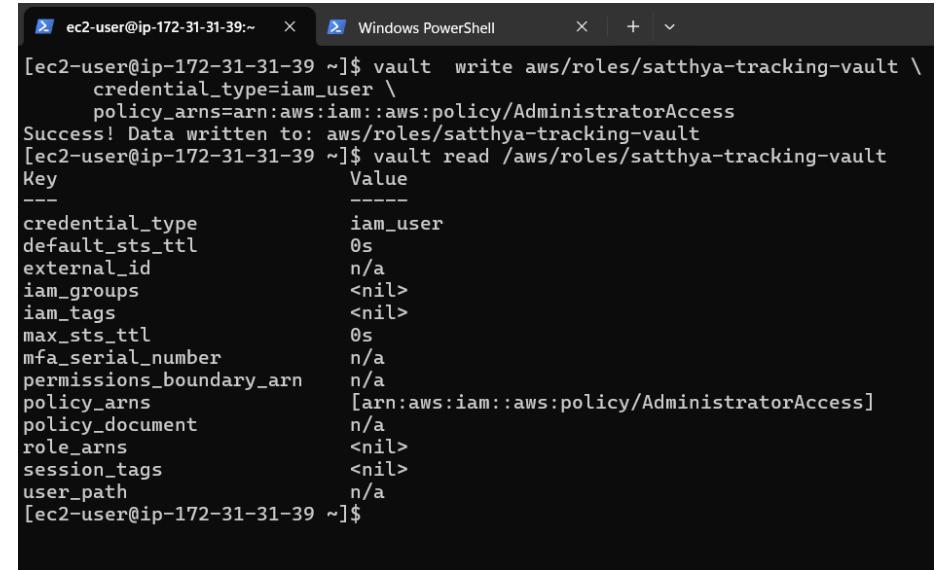
AKIAIXWMA6VK67UQAC7UR	Actions ▾
Description	Status Active
-	Created 1 minute ago
Last used	Last used service None
None	
Last used region	
None	

VAULT CONFIGURATION

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

➤ Vault Role Setup for AWS Access.

```
vault write aws/roles/satthya-tracking-vault \
  credential_type=iam_user \
  policy_arns=arn:aws:iam::aws:policy/AdministratorAccess
```



The screenshot shows a terminal window with two tabs: 'ec2-user@ip-172-31-31-39:~' and 'Windows PowerShell'. The 'ec2-user' tab contains the command and its execution details. The 'Windows PowerShell' tab shows the resulting key-value pairs for the role.

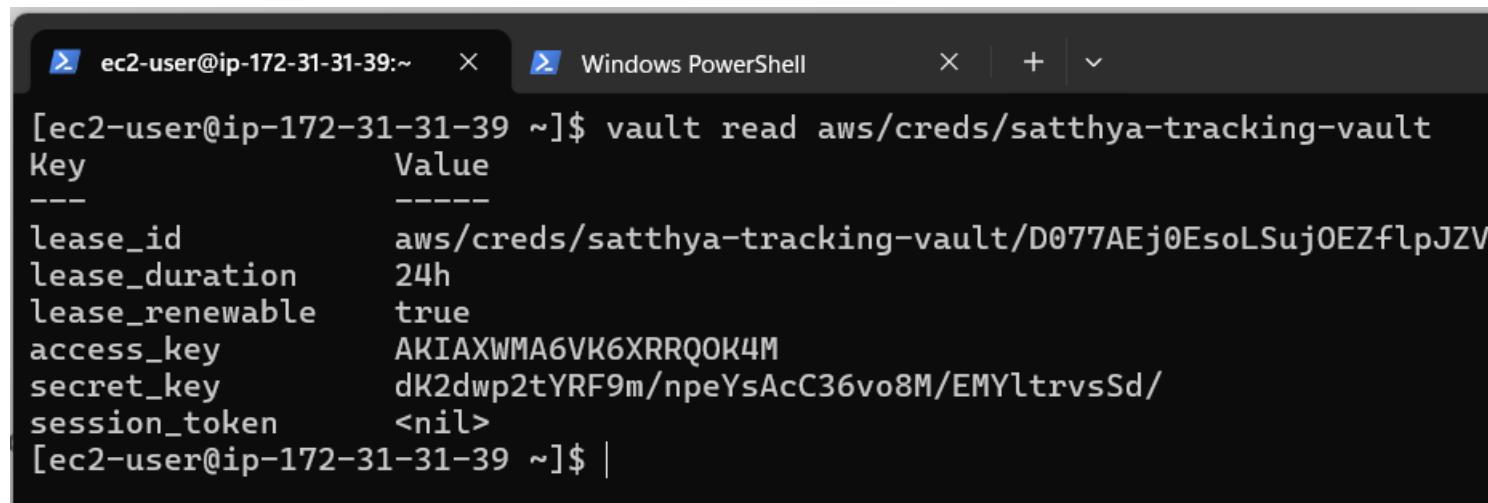
```
[ec2-user@ip-172-31-31-39 ~]$ vault write aws/roles/satthya-tracking-vault \
  credential_type=iam_user \
  policy_arns=arn:aws:iam::aws:policy/AdministratorAccess
Success! Data written to: aws/roles/satthya-tracking-vault
[ec2-user@ip-172-31-31-39 ~]$ vault read /aws/roles/satthya-tracking-vault
Key          Value
---          ---
credential_type    iam_user
default_sts_ttl   0s
external_id        n/a
iam_groups         <nil>
iam_tags           <nil>
max_sts_ttl       0s
mfa_serial_number n/a
permissions_boundary_arn [arn:aws:iam::aws:policy/AdministratorAccess]
policy_arns        n/a
policy_document   <nil>
role_arns         <nil>
session_tags      <nil>
user_path         n/a
[ec2-user@ip-172-31-31-39 ~]$
```

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 ~]$ 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   AKIAIXWMA6VK6XRRQOK4M
secret_key   dK2dwp2tYRF9m/npeYsAcC36vo8M/EMYltrvsSd/
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 shows the AWS IAM Role details page for 'satthya-EC2-EKS'. The top navigation bar includes 'IAM > Roles > satthya-EC2-EKS'. A green success message states 'Role satthya-EC2-EKS created.' Below the message, the role name 'satthya-EC2-EKS' is displayed with a 'View role' button and a 'Delete' button. A description below the role name indicates it allows EC2 instances to call AWS services on behalf of the user. The 'Summary' section provides details like creation date (October 15, 2025), ARN, and maximum session duration (1 hour). The 'Permissions' tab is selected, showing four managed policies attached to the role: 'AmazonEC2ContainerRegistryReadOnly', 'AmazonEKS_CNI_Policy', 'AmazonEKSClusterPolicy', and 'AmazonEKSWorkerNodePolicy'. Each policy is listed with its type (AWS managed) and the number of entities it is attached to (2, 3, 3, and 3 respectively). Buttons for 'Simulate', 'Remove', and 'Add permissions' are also present.

Identity and Access Management (IAM)

satthya-EC2-EKS

Role satthya-EC2-EKS created.

View role X

Delete

satthya-EC2-EKS Info

Allows EC2 instances to call AWS services on your behalf.

Summary

Creation date: October 15, 2025, 15:43 (UTC+08:00)

Last activity: -

ARN: arn:aws:iam::529088293565:role/satthya-EC2-EKS

Instance profile ARN: arn:aws:iam::529088293565:instance-profile/satthya-EC2-EKS

Maximum session duration: 1 hour

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

Permissions policies (4) Info

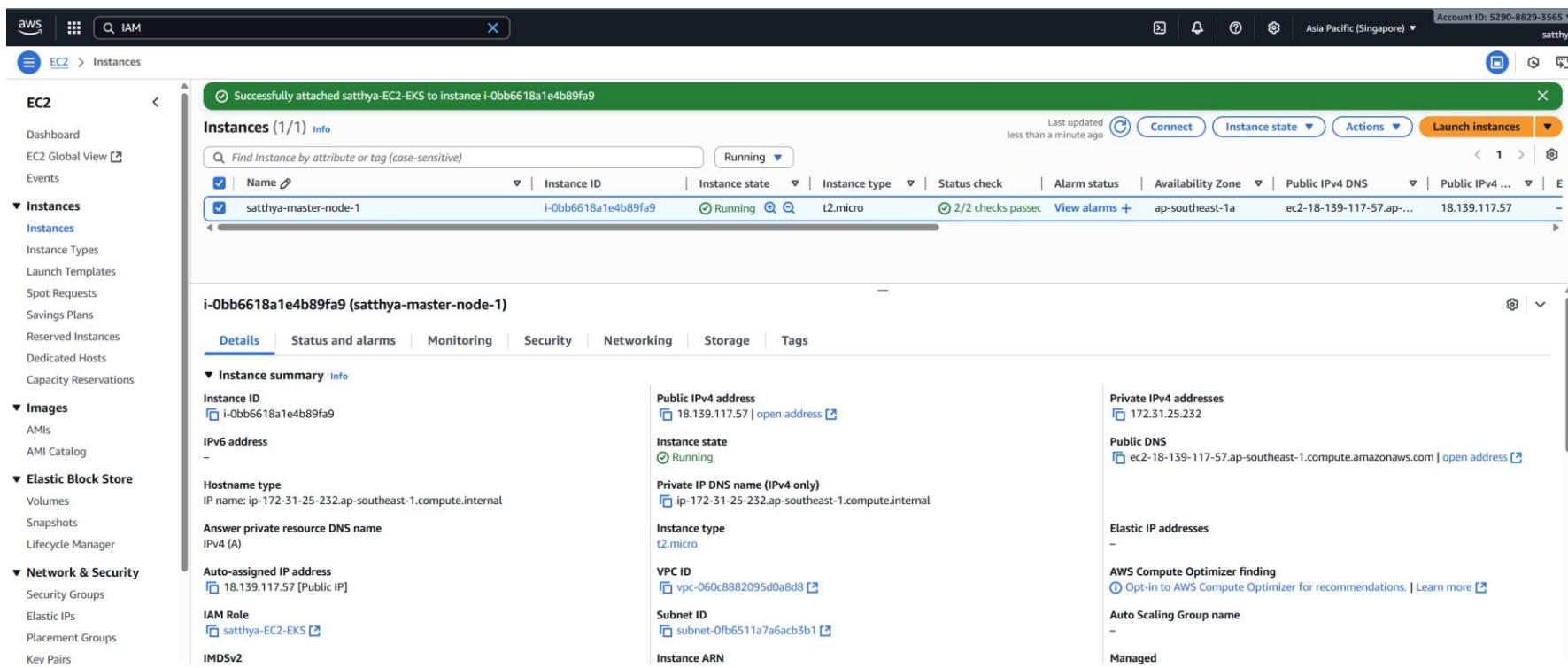
You can attach up to 10 managed policies.

Filter by Type: All types

Policy name	Type	Attached entities
AmazonEC2ContainerRegistryReadOnly	AWS managed	2
AmazonEKS_CNI_Policy	AWS managed	3
AmazonEKSClusterPolicy	AWS managed	3
AmazonEKSWorkerNodePolicy	AWS managed	3

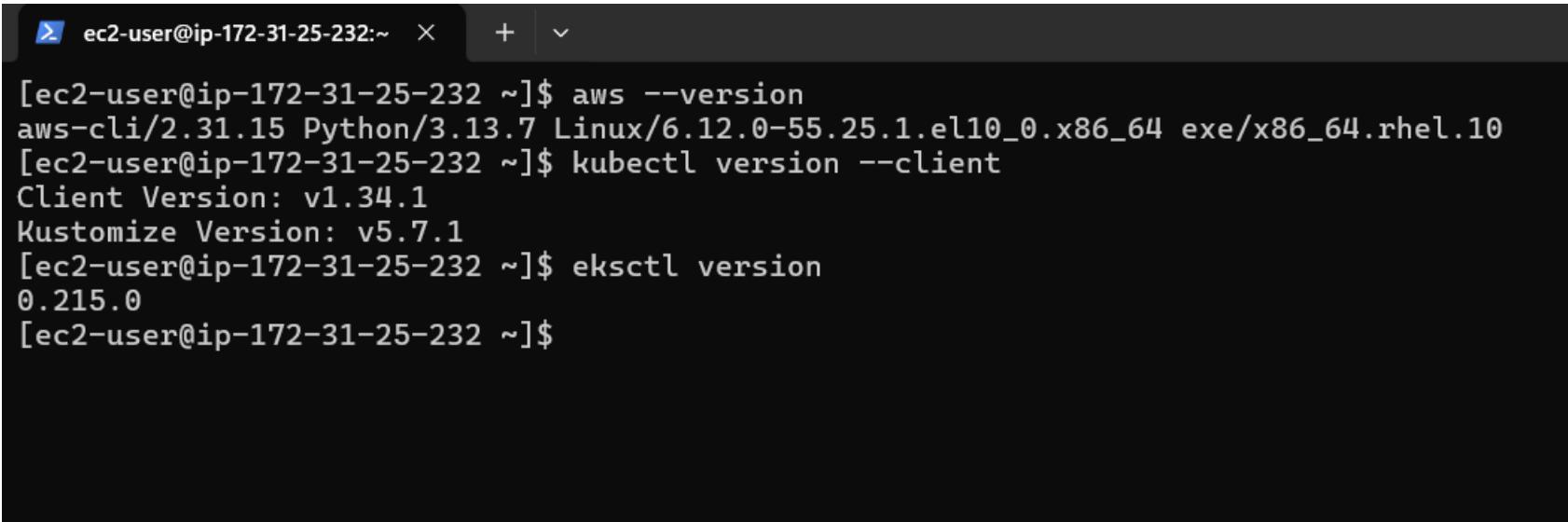
EKS Cluster Master Node on EC2

- 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

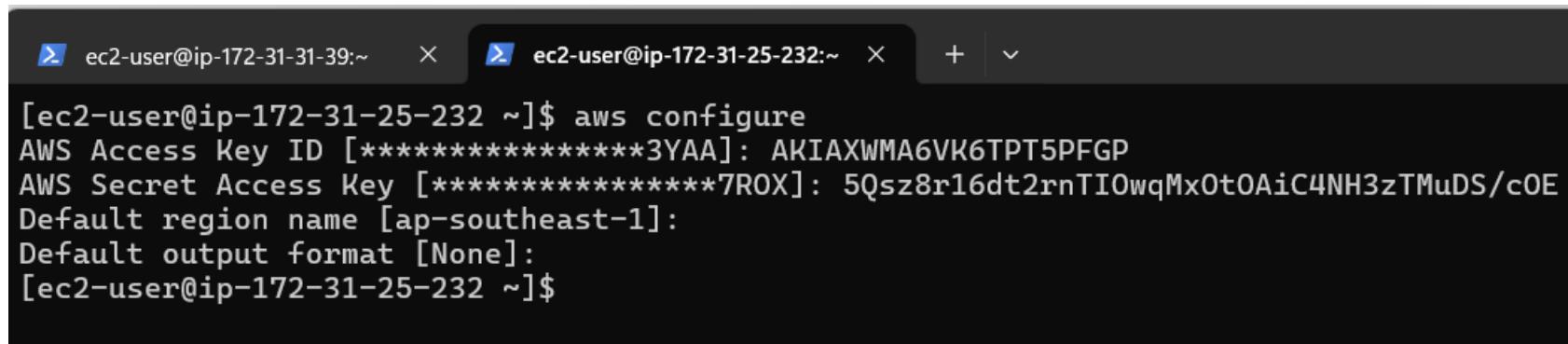
- Installed AWS CLI, eksctl, and kubectl on the master node.



```
[ec2-user@ip-172-31-25-232 ~]$ 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.



The screenshot shows a terminal window with two tabs open. The active tab is titled 'ec2-user@ip-172-31-25-232:~' and displays the output of the 'aws configure' command. The command prompts for AWS Access Key ID, AWS Secret Access Key, Default region name, and Default output format. The AWS Access Key ID is set to 'AKIAJWMA6VK6TPT5PFGP', the AWS Secret Access Key is set to '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', the Default region name is set to 'ap-southeast-1', and the Default output format is set to 'None'. The second tab, 'ec2-user@ip-172-31-31-39:~', is visible in the background.

```
[ec2-user@ip-172-31-25-232 ~]$ aws configure
AWS Access Key ID [*****3YAA]: AKIAJWMA6VK6TPT5PFGP
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
2025-10-15 09:15:00 [i] subnets for ap-southeast-1b - public:192.168.0.0/19 private:192.
2025-10-15 09:15:00 [i] subnets for ap-southeast-1a - public:192.168.32.0/19 private:192.
2025-10-15 09:15:00 [i] subnets for ap-southeast-1c - public:192.168.64.0/19 private:192.
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 ek
rrent behavior, explicitly set 'autoModeConfig.enabled: false' in your cluster configurati
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" regi
2025-10-15 09:15:00 [i] will create 2 separate CloudFormation stacks for cluster itself a
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=
2025-10-15 09:15:00 [i] CloudWatch logging will not be enabled for cluster "satthya-track
2025-10-15 09:15:00 [i] you can enable it with 'eksctl utils update-cluster-logging --ena
2025-10-15 09:15:00 [i] default addons vpc-cni, kube-proxy, coredns, metrics-server were
2025-10-15 09:15:00 [i] 2 sequential tasks: { create cluster control plane "satthya-trackmate",
2025-10-15 09:15:00 [i]   2 sequential sub-tasks: {
2025-10-15 09:15:00 [i]     1 task: { create addons },
2025-10-15 09:15:00 [i]       wait for control plane to become ready,
2025-10-15 09:15:00 [i]     },
2025-10-15 09:15:00 [i]     create managed nodegroup "ng-01eabe70",
2025-10-15 09:15:00 [i]   }
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-cluste
2025-10-15 09:16:00 [i]   waiting for CloudFormation stack "eksctl-satthya-trackmate-cluste
2025-10-15 09:17:01 [i]   waiting for CloudFormation stack "eksctl-satthya-trackmate-cluste
2025-10-15 09:18:01 [i]   waiting for CloudFormation stack "eksctl-satthya-trackmate-cluste
2025-10-15 09:19:01 [i]   waiting for CloudFormation stack "eksctl-satthya-trackmate-cluste
2025-10-15 09:20:01 [i]   waiting for CloudFormation stack "eksctl-satthya-trackmate-cluste
```

EKS Cluster Deployment on AWS

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

The screenshot shows the EKS console interface for the 'sathya-trackmate' cluster. The left sidebar includes navigation links for Dashboard, Clusters, Settings (Dashboard settings, Console settings), Amazon EKS Anywhere (Enterprise Subscriptions), and Related services (Amazon ECR, AWS Batch). The main content area displays the cluster details:

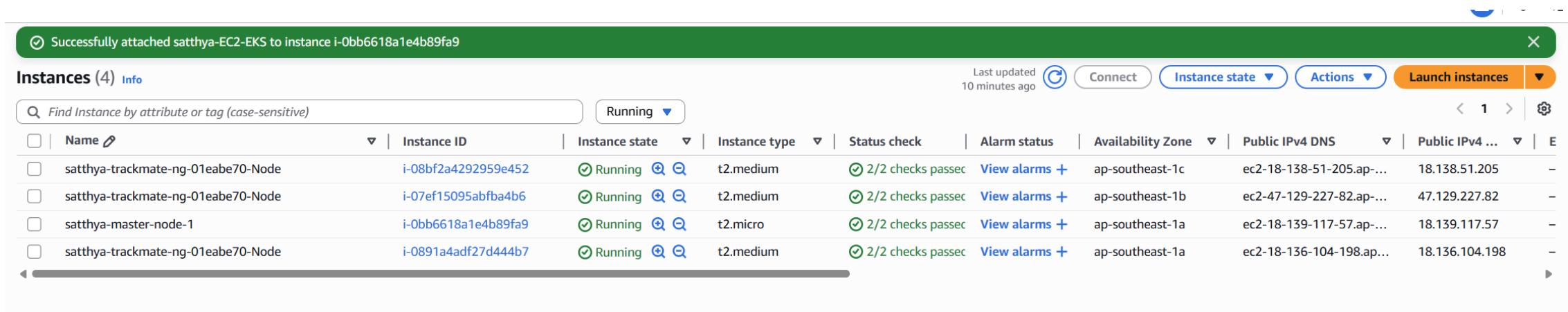
- Cluster info:** Status: Active, Kubernetes version: 1.32, Support period: Standard support until March 23, 2026, Provider: EKS.
- Cluster health:** 0 issues.
- Upgrade insights:** 0 issues.
- Node health issues:** 0 issues.

The 'Overview' tab is selected. Below it, the 'Details' section contains the following information:

- API server endpoint:** <https://9AA40B18C087F633E80456FC5B1ABB07.yl4.ap-southeast-1.eks.amazonaws.com>
- OpenID Connect provider URL:** <https://oidc.eks.ap-southeast-1.amazonaws.com/id/9AA40B18C087F633E80456FC5B1ABB07>
- Certificate authority:** A long base64 encoded string: LS0tLS1CRUJTiBDRVJUSUZJQ0FURS0tLS0tCK1JSURCVENDQWUyZ0F3SUJB20Utm9Mc0YvemJucd3RFZSkvtWklodmNOQVFTEJRQXdGVEVUTUJFROExVUUKQXhNS2EzVmlaWEp1WhSbGN6QWVGd
- Cluster IAM role ARN:** arn:aws:iam::529088293565:role/eksctl-sathya-trackmate-ServiceRole-tN4CtZTmgI4X
- Created:** 18 minutes ago
- Cluster ARN:** arn:aws:eks:ap-southeast-1:529088293565:cluster/sathya-trackmate
- Platform version:** eks.25

EKS Cluster Deployment on AWS

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



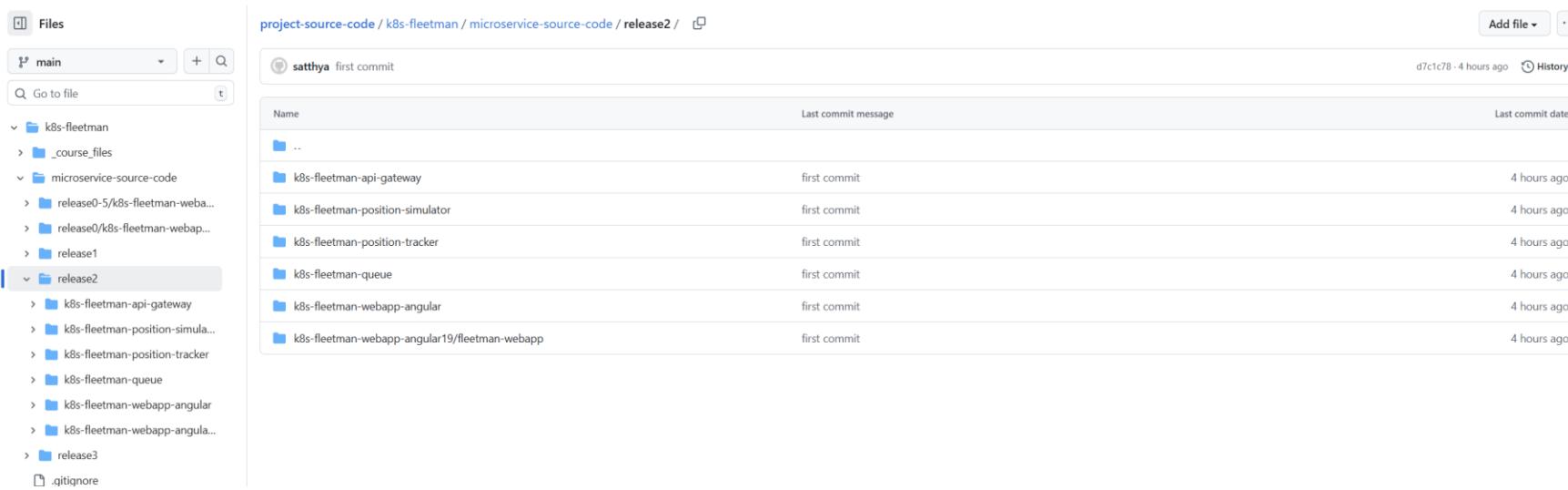
A screenshot of the AWS CloudWatch Metrics console. A green notification bar at the top says "Successfully attached sathya-EC2-EKS to instance i-0bb6618a1e4b89fa9". Below it is a table titled "Instances (4) Info" showing four EC2 instances:

Instances (4) Info											
Last updated 10 minutes ago											
Actions Launch instances											
Find Instance by attribute or tag (case-sensitive)											
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...			E
sathya-trackmate-ng-01eabe70-Node	i-08bf2a4292959e452	Running	t2.medium	2/2 checks passed	View alarms +	ap-southeast-1c	ec2-18-138-51-205.ap...	18.138.51.205	–	–	
sathya-trackmate-ng-01eabe70-Node	i-07ef15095abfba4b6	Running	t2.medium	2/2 checks passed	View alarms +	ap-southeast-1b	ec2-47-129-227-82.ap...	47.129.227.82	–	–	
sathya-master-node-1	i-0bb6618a1e4b89fa9	Running	t2.micro	2/2 checks passed	View alarms +	ap-southeast-1a	ec2-18-139-117-57.ap...	18.139.117.57	–	–	
sathya-trackmate-ng-01eabe70-Node	i-0891a4adf27d444b7	Running	t2.medium	2/2 checks passed	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`



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-fleetma
ode/release2/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.
[satthy@localhost fleetman-webapp]$ ng build
Initial chunk files    Names          Raw size | Estimated transfer size
main-G47HUXJS.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
-rw-rxr-xr-x. 1 satthya satthya 1438 Oct 19 14:36 package.json
-rwrxr-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
-rwrxr-xr-x. 1 satthya satthya 1536 Oct 19 14:36 README.md
drwxr-xr-x. 3 satthya satthya 68 Oct 19 14:36 src
-rwrxr-xr-x. 1 satthya satthya 472 Oct 19 14:36 tsconfig.app.json
-rwrxr-xr-x. 1 satthya satthya 1003 Oct 19 14:36 tsconfig.json
-rwrxr-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
=> => transferring dockerfile: 413B
=> WARN: MaintainerDeprecated: Maintainer instruction is deprecated in favor of using label (line 3)
=> [internal] load metadata for docker.io/library/nginx:1.14.0-alpine
=> [auth] library/nginx:pull token for registry-1.docker.io
=> [internal] load .dockerignore
=> => transferring context: 113B
=> [1/5] FROM docker.io/library/nginx:1.14.0-alpine@sha256:8976218be775f4244df2a60a169d44606b6978bac4375192074ce
=> [internal] load build context
=> => transferring context: 905.57kB
=> CACHED [2/5] RUN apk add --update bash && rm -rf /var/cache/apk/*
=> CACHED [3/5] RUN rm -rf /usr/share/nginx/html/*
=> CACHED [4/5] COPY ./dist/fleetman-webapp/browser/ /usr/share/nginx/html
=> CACHED [5/5] COPY nginx.conf /etc/nginx/nginx.conf
=> exporting to image
=> => exporting layers
=> => writing image sha256:2f20f81eb1a63e82e0f83e7c46597fb5c553bc4ba891a55215c0546080a8700d
=> => naming to docker.io/library/fleetman-webapp
```

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: satthya04/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: satthya04/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: satthya04/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: satthya04/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: satthya04/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:~ % 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-hhkrm 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-x7jx           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

