

Abstract geometric lines in the top left corner of the slide, consisting of several overlapping, irregular polygons and lines in a light gray color.

TASK 5.1

SOAP, JETTY & REST DEPLOYMENT

DHRUBAJOTEE HOWLADER
NAHASAT NIBIR

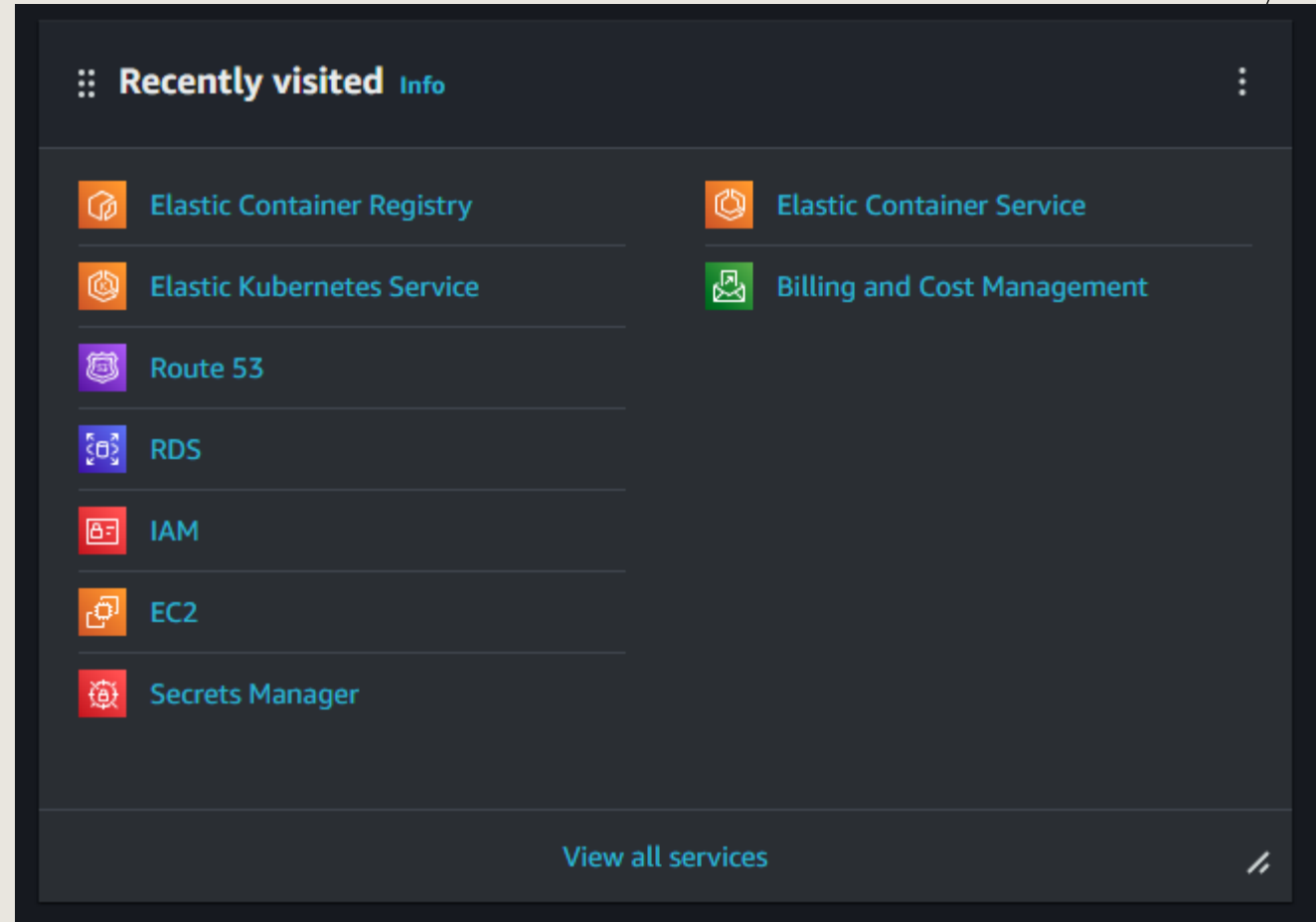
REST DEPLOYMENT

AW-Services Used For Deployment

- **Elastic Container Registry (ECR):** Used to store and manage Docker images for your application.
- **Elastic Kubernetes Service (EKS):** Managed Kubernetes service providing a platform for container orchestration and management.

Supporting Services:

- **Route 53:** DNS service for routing traffic to your application.
- **Relational Database Service (RDS):** Managed database service for storing and managing your application's data.
- **Identity and Access Management (IAM):** Service for managing user access and permissions.
- **EC2:** Elastic Compute Cloud for provisioning and managing virtual machines.
- **Secrets Manager:** Service for securely storing and retrieving secrets used by your application.



REST DEPLOYMENT

AWS – Elastic Container Registry (ECR)

Repository Name:

- Provide a concise and descriptive name.
- Use namespaces to group similar repositories.
- Adhere to naming conventions (letters, numbers, underscores, hyphens, periods).
- Character limit: 2-256 characters.

Image Tag Mutability:

- Choose between "Mutable" (tags can be overwritten) or "Immutable" (tags cannot be overwritten).
- Consider your tagging strategy and the need for immutability.
- Encryption Settings:

Encryption Configuration:

- Default: AES-256 encryption (industry standard).
- Optional: AWS Key Management Service (KMS) encryption for enhanced security.
- Note: Encryption settings cannot be changed after repository creation.

Create private repository

General settings

Repository name

Provide a concise name. Repository names support namespaces, which is recommended for grouping similar repositories.

339712865282.dkr.ecr.ap-south-1.amazonaws.com/

0 out of 256 characters maximum (2 minimum). The name must start with a letter and can only contain lowercase letters, numbers, and special characters `._-/`.

Image tag mutability [Info](#)

Specify the tag mutability setting to use. When tag immutability is turned on for a repository, tags are prevented from being overwritten.


☒ **Mutable**

Image tags can be overwritten.

☐ **Immutable**

Image tags are prevented from being overwritten.

Encryption settings

 The encryption settings for a repository can't be changed once the repository is created.

Encryption configuration [Info](#)

By default, repositories use the industry standard Advanced Encryption Standard (AES) encryption. You can optionally choose to use a key stored in the AWS Key Management Service (KMS) to encrypt the images in your repository.

☒ **AES-256**

Industry standard Advanced Encryption Standard (AES) encryption

☐ **AWS KMS**

AWS Key Management Service (KMS)

► Image scanning settings - *deprecated*

Cancel

Create

REST DEPLOYMENT

AWS – Elastic Container Registry (ECR) - GITHUB CI

Workflow Setup:

- **Trigger:** The workflow is triggered on pushes to the "main" branch.
- **Job:** A single job named "deploy" is defined.
- **Runner:** The job runs on an Ubuntu-latest runner.

Steps:

- **Checkout Repository:** The GitHub Actions checkout action is used to check out the repository code.
- **Configure AWS Credentials:** The AWS credentials are configured using secrets stored in GitHub.
- **Login to Amazon ECR:** The aws-actions/amazon-ecr-login@v2 action is used to log in to the ECR registry.
- **Load Secrets from Secrets Manager:** Secrets stored in AWS Secrets Manager are retrieved and saved to a local app.env file.

Build, Tag, and Push Docker Image:

- The Docker image is built using the docker build command.
- The image is tagged with the GitHub commit SHA.
- The image is pushed to the ECR registry using the docker push command.

Environment Variables:

- **REGISTRY:** The ECR registry URL.
- **REPOSITORY:** The name of the ECR repository.
- **IMAGE_TAG:** The tag for the Docker image (GitHub commit SHA).

```
name: Deploy to production

on:
  push:
    branches: [ "main" ]

jobs:

  deploy:
    name: Build image
    runs-on: ubuntu-latest

    steps:
      - name: Checkout repo
        uses: actions/checkout@v3

      - name: Configure AWS credentials
        uses: aws-actions/configure-aws-credentials@v4
        with:
          aws-access-key-id: ${ secrets.AWS_ACCESS_KEY_ID }
          aws-secret-access-key: ${ secrets.AWS_SECRET_ACCESS_KEY }
          aws-region: ap-south-1

      - name: Login to Amazon ECR
        id: login-ecr
        uses: aws-actions/amazon-ecr-login@v2

      - name: Load secrets and save to app.env
        run: aws secretsmanager get-secret-value --secret-id banking_system --query SecretString --output text | jq -r 'to_entries|map("\(key)=\(.value)"|join("\n")'

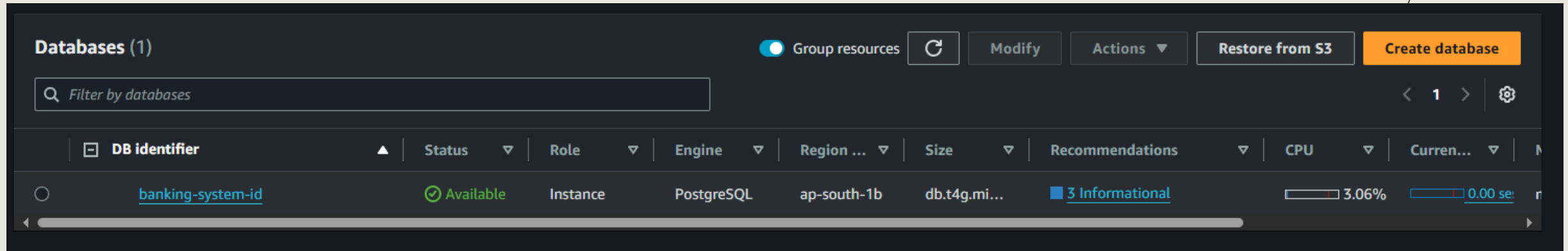
      - name: Build, tag, and push docker image to Amazon ECR
        env:
          REGISTRY: ${ steps.login-ecr.outputs.registry }
          REPOSITORY: banking_system
          IMAGE_TAG: ${ github.sha }
        run: |
          docker build -t $REGISTRY/$REPOSITORY:$IMAGE_TAG .
          docker push $REGISTRY/$REPOSITORY:$IMAGE_TAG
```

REST DEPLOYMENT

AWS – RDS PostgreSQL as Database

Engine Version:

- **PostgreSQL 16.3-R2:** The latest supported version of PostgreSQL for RDS.



REST DEPLOYMENT

AWS – IAM

IAM User:

- **Username:** github-ci
- **ARN:**
arn:aws:iam::339712865282:user/github-ci
- **Console Access:** Disabled
- **Access Keys:** Two active access keys (AKIAU6GDW6QBFFVEJBPQ and

Permissions:

- **Permissions Policies:** Three policies attached to the user through a group.
 - AmazonEC2ContainerRegistryFullAccess
 - EKSFullAccess
 - SecretsManagerReadWrite

The screenshot displays the AWS IAM console interface for the 'github-ci' user. The top navigation bar shows 'IAM > Users > github-ci'. The main content area is titled 'github-ci Info' and includes a 'Delete' button. Below this is a 'Summary' section with a table containing the following information:

ARN	Console access	Access key 1
arn:aws:iam::339712865282:user/github-ci	Disabled	AKIAU6GDW6QBFFVEJBPQ - Active Used 19 hours ago. Yesterday old.
Created October 20, 2024, 13:35 (UTC+06:00)	Last console sign-in -	Access key 2 AKIAU6GDW6QBFBMPCNE - Active Used Yesterday. Yesterday old.

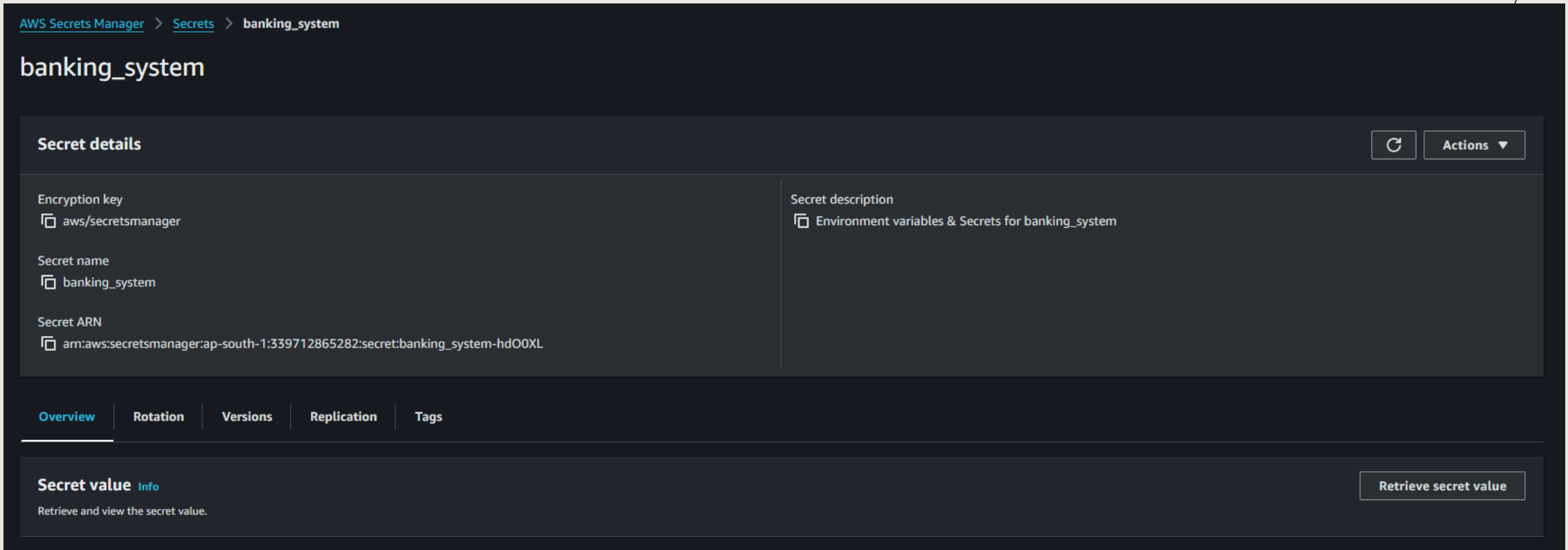
Below the summary is a tabbed interface with 'Permissions' selected. It shows 'Permissions policies (3)' with a search bar and a 'Filter by Type' dropdown set to 'All types'. The table below lists the attached policies:

Policy name	Type	Attached via
AmazonEC2ContainerRegistryFullAccess	AWS managed	Group deployment
EKSFullAccess	Customer inline	Group deployment
SecretsManagerReadWrite	AWS managed	Group deployment

REST DEPLOYMENT

AWS – Secrets Manager

For Safety Purpose We are using AWS Secrets Manager to Store Environment Variables



The screenshot displays the AWS Secrets Manager console interface. At the top, the breadcrumb navigation shows 'AWS Secrets Manager > Secrets > banking_system'. The main heading is 'banking_system'. Below this, the 'Secret details' section is visible, containing the following information:

- Encryption key:** aws/secretsmanager
- Secret name:** banking_system
- Secret ARN:** arn:aws:secretsmanager:ap-south-1:339712865282:secret:banking_system-hd00XL
- Secret description:** Environment variables & Secrets for banking_system

On the right side of the 'Secret details' section, there are two buttons: a refresh icon and an 'Actions' dropdown menu. Below the details section, there is a horizontal tab bar with the following tabs: 'Overview' (selected), 'Rotation', 'Versions', 'Replication', and 'Tags'. At the bottom of the console, the 'Secret value' section is visible, with a sub-label 'Info'. It contains the text 'Retrieve and view the secret value.' and a button labeled 'Retrieve secret value'.

REST DEPLOYMENT

AWS – Elastic Kubernetes Service (EKS)

We are using EKS to contain the images that we can deploy

The screenshot displays the AWS Management Console interface for an EKS cluster named 'banking_system'. The breadcrumb navigation at the top shows 'EKS > Clusters > banking_system'. The cluster name 'banking_system' is prominently displayed at the top left of the console, with a refresh icon and a 'Delete cluster' button to its right. Below this, a 'Cluster info' section is expanded, showing a table with the following details:

Status	Kubernetes version	Support period	Provider
Active	1.31	Standard support until November 26, 2025	EKS

Below the cluster info section, a horizontal navigation bar contains tabs for 'Overview', 'Resources', 'Compute', 'Networking', 'Add-ons', 'Access', 'Observability', 'Upgrade insights', 'Update history', and 'Tags'. The 'Overview' tab is currently selected. Under the 'Details' section, several key attributes are listed:

- API server endpoint:** `https://1496501B3C284BAEE7FB39026914F606.gr7.ap-south-1.eks.amazonaws.com`
- OpenID Connect provider URL:** `https://oidc.eks.ap-south-1.amazonaws.com/id/1496501B3C284BAEE7FB39026914F606`
- Created:** October 18, 2024, 23:59 (UTC+06:00)
- Certificate authority:** `LS0tLS1CRUdJTiBDRVJUSUZJQ0FURSB0tLS0tCk1JSURCVEN DQWUyZ0F3SUJBZ0U3paMnRMUVRrb2t3RFFZSkVWklod mNOQVFFTEJRQXdGVEVUTUJFR0ExVUUKQXhNS2EzVmlaW`
- Cluster IAM role ARN:** `arn:aws:iam::339712865282:role/AWSEKSClusterRole` (with a 'View in IAM' link)
- Cluster ARN:** `arn:aws:eks:ap-south-1:339712865282:cluster/banking_system`
- Platform version:** eks.6

REST DEPLOYMENT

AWS – Elastic Kubernetes Service (EKS)

We need **aws-auth.yaml**, **deployment.yaml**, **service.yaml** for deploying to AWS-EKS

aws-auth.yaml

```
eks > aws-auth.yaml
1  apiVersion: v1
2  kind: ConfigMap
3  metadata:
4    name: aws-auth
5    namespace: kube-system
6  data:
7    mapUsers: |
8      - userarn: arn:aws:iam::339712865282:user/github-ci
9        username: github-ci
10     groups:
11     - system:masters
```

service.yaml

```
eks > service.yaml
1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: banking-system-api-service
5  spec:
6    selector:
7      app: banking-system-api
8    ports:
9      - protocol: TCP
10        port: 80
11        targetPort: 8080
12    type: LoadBalancer
```

deployment.yaml

```
eks > deployment.yaml
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: banking-system-api-deployment
5  labels:
6    app: banking-system-api
7  spec:
8    replicas: 1
9    selector:
10     matchLabels:
11       app: banking-system-api
12  template:
13    metadata:
14     labels:
15       app: banking-system-api
16    spec:
17     containers:
18     - name: banking-system-api
19       image: 339712865282.dkr.ecr.ap-south-1.amazonaws.com/banking_system:e776812bf9cd66310899647978e35a46da4d8ae0
20       imagePullPolicy: Always
21     ports:
22     - containerPort: 8080
23
```

REST DEPLOYMENT

AWS – Elastic Kubernetes Service (EKS)

Console Commands To Use

```
# ----- Start
# kubectl version --client
# ls -l ~/.aws
# cat ~/.kube/config
# cat ~/.aws/credentials
# aws sts get-caller-identity
# vi ~/.aws/credentials = To edit the credentials
# kubectl cluster-info
# export AWS_PROFILE=default
# kubectl apply -f eks/aws-auth.yaml --> Make sure your in the default profile when executing this
# kubectl get service
# kubectl get pods
# use k9s for better kubernetes cluster usage
# kubectl apply -f eks/deployment.yaml
# kubectl apply -f eks/service.yaml
configawsEKS:
    aws eks update-kubeconfig --name banking_system --region ap-south-1
# To connect kubectl to aws eks cluster
connectawsEKSCluster:
    kubectl config use-context arn:aws:eks:ap-south-1:339712865282:cluster/banking_system
# ----- End
```