**LAB# 3**

**Securing EKS Applications with AWS Secrets**

**Manager**

**\*\*Tasks\*\*:**

**1. Store a secret in AWS Secrets Manager.**

**2. Access the secret from an EKS pod.**

**3. Rotate the secret and validate updates.**

**\*\*Documentation\*\*:**

**- Kubernetes security basics.**

**- AWS Secrets Manager overview.**

**- Securely managing and accessing secrets**

**Prerequisites:**

**Install kubectl**

**Install eks**

**Install awscli**

**Add access key**

**Add roles for cluster and node permissions**

**Before you can access a secret from an EKS pod, you need to store it securely in AWS Secrets Manager. Follow these steps to store a secret:**

* **Log in to the AWS Management Console.**
* **Navigate to the AWS Secrets Manager service.**
* **Click "Store a new secret."**
* **Choose the "Credentials for RDS database, Redshift cluster, or Aurora cluster" option or any other suitable option depending on your use case.**
* **Enter the secret values, such as username and password.**
* **Define the rotation policy if you plan to rotate the secret regularly.**
* **Review and confirm the settings, and then click "Store" to create the secret.**

[**https://docs.aws.amazon.com/secretsmanager/latest/userguide/integrating\_csi\_driver.html**](https://docs.aws.amazon.com/secretsmanager/latest/userguide/integrating_csi_driver.html)

| apiVersion: v1 kind: Pod metadata:  name: my-app-pod spec:  containers:  - name: my-app-container  image: my-app-image  volumeMounts:  - name: secret-volume  mountPath: /mnt/secrets-store  volumes:  - name: secret-volume  csi:  driver: secrets-store.csi.k8s.io  readOnly: true  volumeAttributes:  secretProviderClass: "aws-secrets" |
| --- |

To rotate a secret stored in AWS Secrets Manager, follow these steps:

* Go to the Secrets Manager in the AWS Management Console.
* Select the secret you want to rotate.
* Click "Rotate secret."
* Configure the rotation settings and choose a Lambda function or an AWS Fargate task to perform the rotation.
* Monitor the rotation process and ensure that your application continues to function correctly with the updated secret values.

It's important to thoroughly test the rotation process and validate that your application can handle secret updates without any disruptions.

**Conclusion of the lab:**

**The process of securing EKS applications with AWS Secrets Manager underscores the critical role of managing sensitive information securely within Kubernetes environments**.

* **Secret Storage in AWS Secrets Manager: Storing secrets in AWS Secrets Manager provides a centralized and highly secure repository for sensitive data such as API keys, passwords, or encryption keys. This ensures a robust security posture by preventing direct exposure within EKS clusters.**
* **Accessing Secrets from EKS Pods: Implementing secure access mechanisms allows EKS pods to retrieve secrets from AWS Secrets Manager securely. This ensures that only authorized applications or services can access sensitive information, bolstering data confidentiality.**
* **Rotating Secrets and Validation: Rotating secrets at regular intervals enhances security by minimizing the risk of prolonged exposure to static credentials. Validating the update process ensures that applications can seamlessly access updated secrets without disruptions.**