**Objective:** Explore how EKS can work with serverless components.

**Tasks:**

1. Deploy an application in EKS.

2. Set up an AWS Lambda function.

3. Trigger the Lambda function based on EKS events.

**Documentation:**

- Event-driven architecture basics.

- EKS and Lambda integration.

- Real-world use cases.

**Requirements:**

1. An AWS account with required permissions to create and manage EKS clusters, EC2 instances, and EBS volumes.
2. AWS CLI installed and configured.
3. **kubectl** installed.
4. **eksctl** installed.
5. AWS Lambda Knowledge

**Tasks**

**Task 1: Deploy an Application in EKS**

1. **Setup EKS Cluster:** Create an EKS cluster using AWS Management Console, AWS CLI, or an Infrastructure as Code (IaC) tool like Terraform or AWS CloudFormation.
2. **Configure kubectl:** Configure kubectl to interact with your EKS cluster.
3. **Create a Deployment:** Write a Kubernetes deployment YAML file to describe your application’s deployment configuration.
4. **Deploy the Application:** Use kubectl apply to deploy your application to the EKS cluster.
5. **Verify Deployment:** Ensure that the application is running successfully.

**Task 2: Set Up an AWS Lambda Function**

1. **Create a Lambda Function:** Go to the AWS Lambda Console and create a new Lambda function.
2. **Choose a Runtime:** Select a runtime for your function. AWS Lambda supports Node.js, Python, Java, and more.
3. **Write the Function Code:** Write the code for your Lambda function.
4. **Set Up Triggers:** Configure the triggers for your Lambda function. You will set up an EKS event as a trigger in the next task.
5. **Deploy the Lambda Function:** Deploy your function by saving your changes.

**Task 3: Trigger the Lambda Function Based on EKS Events**

1. **Generate EKS Events:** Create events in EKS that should trigger the Lambda function. This could be a change in the deployment state, a pod failure, etc.
2. **Create an EventBridge Rule:** Go to the Amazon EventBridge console and create a new rule.
3. **Define the Event Pattern:** Define an event pattern that matches the EKS events you want to trigger the Lambda function.
4. **Set Up the Lambda Function as a Target:** Configure the Lambda function you created as the target for the EventBridge rule.
5. **Test the Integration:** Generate events in EKS and verify that they trigger the Lambda function.

**Documentation**

**Event-Driven Architecture Basics**

Event-driven architecture is a software architecture paradigm promoting the production, detection, consumption of, and reaction to events. An event can be defined as "a significant change in state," and this architecture relies heavily on the asynchronous transmission of events.

**EKS and Lambda Integration**

Amazon EKS and AWS Lambda can work together to create a seamless event-driven architecture. EKS can generate events based on the state of the Kubernetes cluster, and these events can be used to trigger AWS Lambda functions.

Real-World Use Cases

1. **Auto-Scaling:** Automatically scale EKS pods up or down based on demand.
2. **Error Monitoring and Alerts:** Trigger Lambda functions to send alerts or perform automated diagnostics when errors occur in EKS.
3. **Continuous Deployment:** Use Lambda to trigger deployment pipelines based on changes in the EKS cluster.
4. **Resource Cleanup:** Automatically clean up unused resources in EKS using Lambda functions.
5. **Audit Logging:** Use Lambda to log and audit changes in the EKS cluster for compliance and monitoring.