

3.2. Student Handout

Amazon EKS (Elastic Kubernetes Service) Student Handout

Overview

Amazon EKS is a managed service that simplifies running Kubernetes on AWS. It handles the Kubernetes control plane, allowing you to focus on deploying and managing applications.

Key Features of Amazon EKS

- 1. Managed Kubernetes Control Plane:** AWS manages the Kubernetes control plane, including the API server and etcd.
 - Example: AWS automatically updates and patches the control plane.
 - Example: AWS ensures high availability of the control plane across multiple availability zones.
 - Example: AWS handles the scaling of the control plane based on demand.
- 2. Integration with AWS Services:** EKS integrates with AWS services like IAM, CloudWatch, and ELB.
 - Example: Use IAM for secure access control to your Kubernetes resources.
 - Example: Monitor application performance with CloudWatch metrics.
 - Example: Balance traffic to your applications using Elastic Load Balancing.
- 3. Security:** EKS offers built-in security features such as IAM roles for service accounts and VPC isolation.
 - Example: Assign IAM roles to Kubernetes service accounts for secure AWS service access.
 - Example: Use VPC isolation to control network access to your EKS cluster.
 - Example: Encrypt data at rest using AWS Key Management Service (KMS).
- 4. Scalability:** EKS supports Kubernetes' native scaling features.

- Example: Use Horizontal Pod Autoscalers to adjust the number of pods based on CPU usage.
 - Example: Scale your application automatically during peak traffic times.
 - Example: Reduce resource usage during off-peak hours by scaling down.
-

Setting Up and Configuring an EKS Cluster

1. **Create an EKS Cluster:** Use AWS Management Console, AWS CLI, or eksctl.
 - Example: Create a cluster using the AWS Management Console with a few clicks.
 - Example: Use AWS CLI to script the creation of an EKS cluster.
 - Example: Automate cluster creation with eksctl for consistent environments.
 2. **Configure Networking:** Set up a VPC, subnets, and security groups.
 - Example: Use an existing VPC to integrate EKS with your current network setup.
 - Example: Create a new VPC tailored for your EKS cluster's needs.
 - Example: Define security groups to control traffic to and from your cluster.
 3. **IAM Roles:** Configure IAM roles for cluster and node interactions.
 - Example: Create an IAM role for the EKS control plane to access AWS services.
 - Example: Assign an IAM role to worker nodes for pulling container images from ECR.
 - Example: Use IAM roles for service accounts to grant applications specific permissions.
 4. **Install kubectl:** Use kubectl to interact with your Kubernetes cluster.
 - Example: Deploy applications using kubectl apply with YAML configuration files.
 - Example: Manage cluster resources with kubectl commands.
 - Example: Troubleshoot applications using kubectl logs and kubectl describe.
-

Deploying and Managing Applications on EKS

1. **Create a Deployment:** Define application replicas using Kubernetes Deployments.
 - Example: Deploy a web application with three replicas for high availability.
 - Example: Update an application with zero downtime using rolling updates.

- Example: Scale the deployment to handle increased load.
2. **Expose the Application:** Use Services to expose applications.
 - Example: Use a LoadBalancer service to expose your application to the internet.
 - Example: Use a ClusterIP service for internal communication within the cluster.
 - Example: Use a NodePort service to expose your application on a specific port.
 3. **Manage Services and Ingress Controllers:** Control access to your applications.
 - Example: Use an Ingress Controller to manage HTTP/HTTPS traffic.
 - Example: Configure path-based routing with Ingress resources.
 - Example: Secure your application with TLS termination at the Ingress level.
-

Scaling and Monitoring Applications

1. **Horizontal Pod Autoscalers (HPA):** Automatically scale pods based on resource usage.
 - Example: Configure HPA to scale pods between 2 and 10 based on CPU usage.
 - Example: Use HPA to maintain application performance during traffic spikes.
 - Example: Monitor HPA activity to optimize resource allocation.
 2. **Using CloudWatch for Monitoring and Logs:** Track metrics and logs.
 - Example: Send application logs to CloudWatch Logs for centralized management.
 - Example: Monitor CPU and memory usage with CloudWatch Metrics.
 - Example: Set up CloudWatch Alarms to notify you of performance issues.
 3. **Integrating AWS Monitoring Tools:** Enhance monitoring with AWS tools.
 - Example: Use AWS X-Ray for tracing requests through your application.
 - Example: Get best practice recommendations with AWS Trusted Advisor.
 - Example: Use AWS CloudTrail for auditing API calls to your EKS cluster.
-

Conclusion

Amazon EKS simplifies running Kubernetes on AWS by managing the control plane, allowing you to focus on deploying and scaling applications. With seamless integration with AWS

services, robust security features, and powerful scaling and monitoring capabilities, EKS is a powerful tool for managing containerized applications in the cloud.