Amazon EKS Use Case

Amazon EKS Client Project: Streamlining Kubernetes Management

GENERAL CHARACTERISTICS

Intent	To simplify Kubernetes management and ensure scalable, secure, and compliant operations for a healthcare company using Amazon Elastic Kubernetes Service (EKS).
Scope	Deployment of Amazon EKS clusters with integrated security and monitoring solutions, enabling seamless scalability for containerized workloads.
Level	System-level.
Client	Confidential (Healthcare Company).
Last Update	[Today's Date]
Status	Finalized.
Stage	Implementation and Validation.

ACTORS

Primary Actor	DevOps Engineer.
Secondary Actors	Solution Architect, Security Engineer, Compliance Officer.

PREREQUISITES

Static Preconditions	- AWS account with required permissions and services enabled (Amazon EKS, IAM, CloudWatch and Secrets Manager). - VPC configured with subnets for EKS worker nodes and associated resources.
Dynamic Preconditions	- Containerized application ready for deployment. - Role-based access policies configured for teams and applications.
Assumptions	- Kubernetes workloads can be scaled using Horizontal Pod Autoscaler (HPA). - The client's regulatory compliance needs (e.g., HIPAA) can be met using AWS native tools.

TRIGGERS

Trigger Event	The client needs to modernize infrastructure for their patient data analysis platform, ensuring scalability, compliance, and simplified Kubernetes operations.

Copyright © 2024-25 spireMinds Limited.

Amazon EKS Use Case

EXPECTED OUTCOME

Success Postcondition	The client achieves dynamic scalability, enhanced security, and regulatory compliance, with reduced operational overhead for Kubernetes management.
Failed Postcondition	Workloads experience performance bottlenecks or fail to meet compliance requirements.

OPERATIONS AND CONCEPTS

Operations	1. Deployed Amazon EKS clusters to host containerized applications securely. 2. Configured Kubernetes Horizontal Pod Autoscaler (HPA) for dynamic workload scaling. 3. Implemented AWS Secrets Manager for secure handling of sensitive application credentials. 4. Integrated Amazon CloudWatch and Prometheus for monitoring and alerting. 5. Enabled IAM roles for Kubernetes Service Accounts (IRSA) for secure access control. 6. Conducted regular compliance and security audits.
Concepts	- Amazon EKS: Fully managed Kubernetes service for deploying and managing containerized workloads. - Kubernetes HPA: Dynamically adjusts resources based on workload demands. - AWS Secrets Manager: Manages secure storage and retrieval of application credentials. - Amazon CloudWatch: Provides monitoring, logging, and alerting for infrastructure and applications. - Prometheus: Open-source tool for monitoring Kubernetes clusters and visualizing metrics.

MAIN SUCCESS SCENARIO

Step 1	Assessed the client's requirements and designed a multi-region EKS architecture for high availability.
Step 2	Deployed EKS clusters with managed node groups to simplify operations and enhance reliability.

Copyright © 2024-25 spireMinds Limited.

Amazon EKS Use Case

Step 3	Configured HPA to automatically scale application pods based on traffic and resource demands.
Step 4	Implemented Secrets Manager for managing sensitive data securely.
Step 5	Set up CloudWatch and Prometheus for monitoring application performance and generating alerts.
Step 6	Applied IAM roles to Kubernetes service accounts to secure access to AWS services.
Step 7	Conducted penetration testing and compliance audits to validate the solution against regulatory requirements.
Step 8	Delivered a fully functional, scalable, and secure Kubernetes management solution using Amazon EKS.

spireMinds

Copyright © 2024-25 spireMinds Limited.