2/5/2025

OpsFleet Infrastructure Docs

Assessment 1

Eugene Parker
DEVOPS ENGINEER

TABLE OF CONTENTS

Assessment 1: Terraform Scripting for AKS Cluster & Karpenter

1. Introduction

This document provides a comprehensive overview of the Terraform scripting implementation for deploying an AKS Cluster with Karpenter. It details the infrastructure components, including network design, resource deployment, and monitoring setups for different environments such as Development, Staging, and Production.

2. Infrastructure State Management

OpsFleet S3 Infrastructure Bucket

- A dedicated AWS S3 bucket to store Terraform state files securely.
- Ensures remote backend storage for consistency, collaboration, and disaster recovery.
- Enables state locking to prevent conflicting Terraform runs.

OpsFleet Infrastructure State Files

- Terraform state files store metadata about deployed resources.
- Helps track infrastructure changes and maintains version history.
- Enables automation and consistency across different deployments.
-

3. OpsFleet Network Environment Design

Development (Dev) Environment

- **Virtual Network**: Creates a private and secure network space for resources.
- **Web Tier Subnet**: Hosts frontend applications and public-facing services.
- **Public Tier Subnet**: Allocates public IP resources such as load balancers.
- **Data Tier Subnet**: Provides an isolated environment for databases and sensitive data.
- **Management Subnet**: Reserved for administrative, monitoring, and security-related services.
- **Route Table**: Defines routing rules to manage network traffic efficiently.
- **EC2 Private Endpoint**: Enables secure connectivity to AWS services without internet exposure.
- **Internet Gateway**: Provides external connectivity for public-facing resources.

Production (Prod) Environment

- **Virtual Network**: Secure, high-availability network for production workloads.
- **Web Tier Subnet**: Hosts live applications with load balancing and redundancy.
- **Public Tier Subnet**: Manages public resources such as API gateways.
- **Data Tier Subnet**: Stores critical database instances with encryption and access control.
- **Management Subnet**: Reserved for administrative functions and monitoring.

- **Route Table**: Defines network traffic routes and access control policies.
- **EC2 Private Endpoint**: Secure private access to AWS services.
- **Internet Gateway**: Facilitates external connectivity where required.
- **Flow Logs**: Captures and logs network activity for security audits and troubleshooting.

Staging Environment

- **Virtual Network**: Mirrors the production environment for testing and validation.
- **Web Tier Subnet**: Deploys staging versions of web applications.
- **Public Tier Subnet**: Manages external-facing services for staging tests.
- **Data Tier Subnet**: Stores testing data in a controlled environment.
- **Management Subnet**: Facilitates access for engineers and monitoring tools.
- **Route Table**: Controls network traffic within the staging environment.
- **EC2 Private Endpoint**: Ensures private connectivity without public exposure.
- **Internet Gateway**: Enables selective external access for testing.
- **Flow Logs**: Monitors network activity to detect performance issues or anomalies.

4. OpsFleet Dev Resources Deployment

Development Infrastructure Components

- **Dev Server (connects to cluster)**: Centralized server for managing the Kubernetes cluster.
- **Dev Cluster**: An AKS (Azure Kubernetes Service) cluster that manages containerized applications.
- **Node Groups**:
  - **OpsFleet Node-1**: Primary compute node for handling workloads.
  - **OpsFleet Node-2**: Secondary node ensuring redundancy and scalability.
- **Node Pools (Cluster Default)**:
  - **System Pool**: Manages critical system services for Kubernetes.
  - **General-purpose Pool**: Handles application workloads for development and testing.

5. OpsFleet Containers (Pod) Deployment

Karpenter Pods

Karpenter is an autoscaler designed to provision and optimize Kubernetes nodes dynamically.

- **Karpenter Pod-1**: Ensures automatic provisioning of required compute resources.
- **Karpenter Pod-2**: Provides additional scaling support to meet workload demands.

6. Logging & Monitoring

Logging Setup

- **OpsFleet Dev Network Flow Logs**: Captures IP traffic for security analysis and troubleshooting.
- **OpsFleet Dev EC2 Server Monitoring**: Tracks system performance metrics including CPU, memory, and disk usage.
- **OpsFleet Dev Cluster Monitoring**: Observes Kubernetes workloads, pod health, and resource consumption.

Monitoring Implementation

- **OpsFleet Dev EC2 Monitoring**: Provides insights into instance health, resource utilization, and performance trends.
- **OpsFleet Dev Cluster Monitoring**: Tracks Kubernetes cluster health, pod statuses, and system performance metrics.

7. Conclusion

This documentation provides a detailed blueprint for implementing a Terraform-based AKS cluster with Karpenter, covering network design, resource deployment, logging, and monitoring setups. By leveraging Terraform, organizations can achieve infrastructure automation, scalability, and maintainability across different environments.
