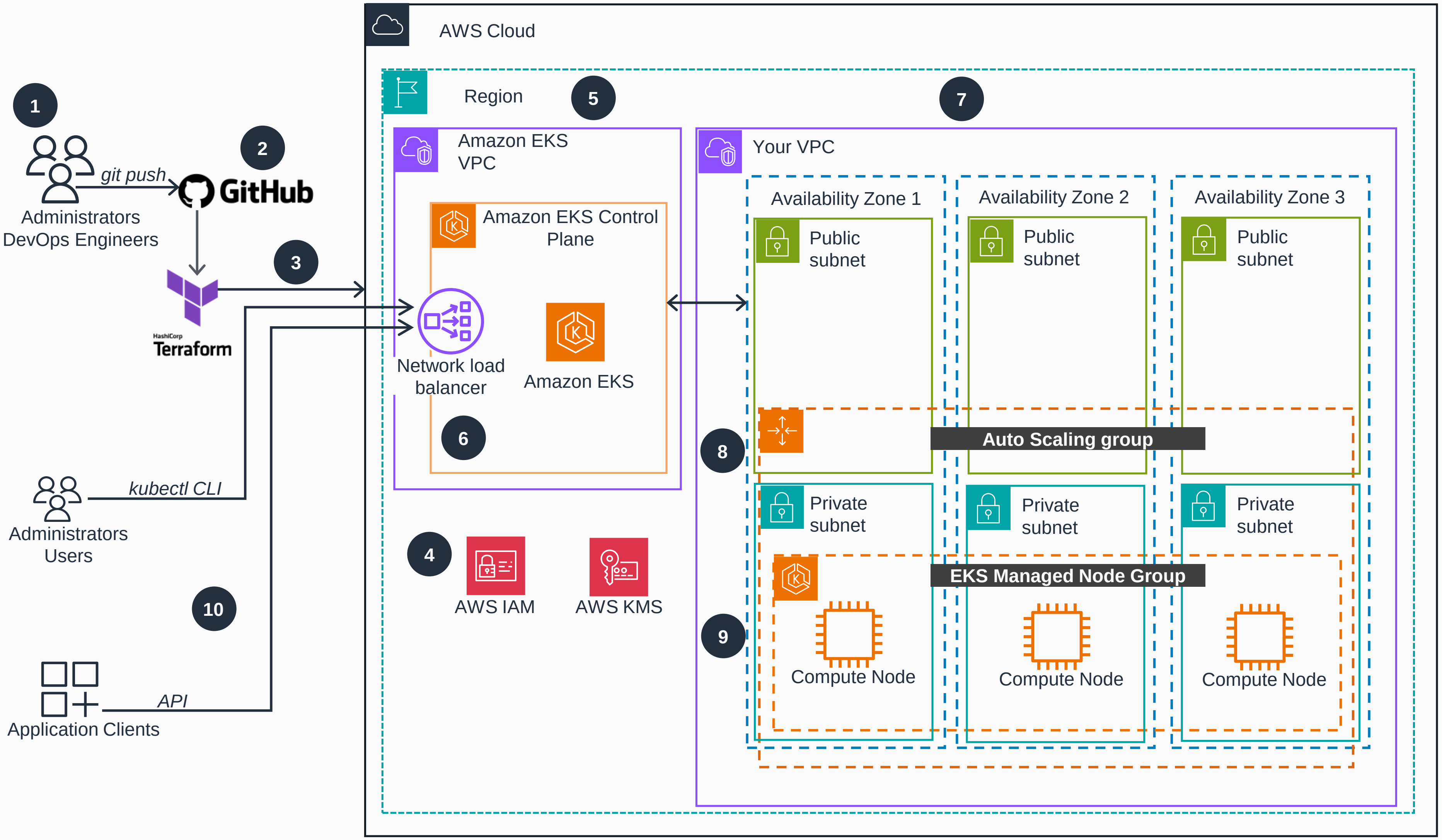
Guidance for Monitoring Amazon EKS Workloads using Amazon Managed Services for Prometheus & Grafana

(**OPTIONAL**) This architecture diagram demonstrates an Amazon Elastic Kubernetes Service (Amazon EKS) cluster provisioned through an Amazon EKS Blueprint for Terraform.



Optional
To deploy this Guidance, you need an **Amazon Elastic Kubernetes Service (Amazon EKS)** cluster provisioned. These steps show how to provision an **Amazon EKS** cluster with Amazon EKS Blueprint for Terraform.

1 Administrator or DevOps user commits Infrastructure as Code (IaC) code changes to **Amazon EKS** blueprint into Git repository.

2 Blueprint provisioning workflow is invoked upon code push to Git repo.

3 Terraform starts resource deployment processes against target AWS environment.

4 The required **AWS Identity and Access Management** (IAM) roles, polices, and **AWS Key Management Service** (AWS KMS) keys are created by Terraform.

5 The **Amazon EKS** virtual private cloud (VPC) for the control plane component is deployed by Terraform.

6 The **Amazon EKS** cluster control plane component is deployed into the **Amazon EKS** VPC by Terraform.

7 Your VPC is deployed for the compute plane by Terraform.

8 Subnets and other networking components are deployed into cluster VPCs by Terraform.

9 The **Amazon EKS** node group with compute plane nodes (**Amazon Elastic Compute Cloud (Amazon EC2)** instances in auto scaling group) is deployed into the cluster VPC by Terraform and joins the **Amazon EKS** cluster.

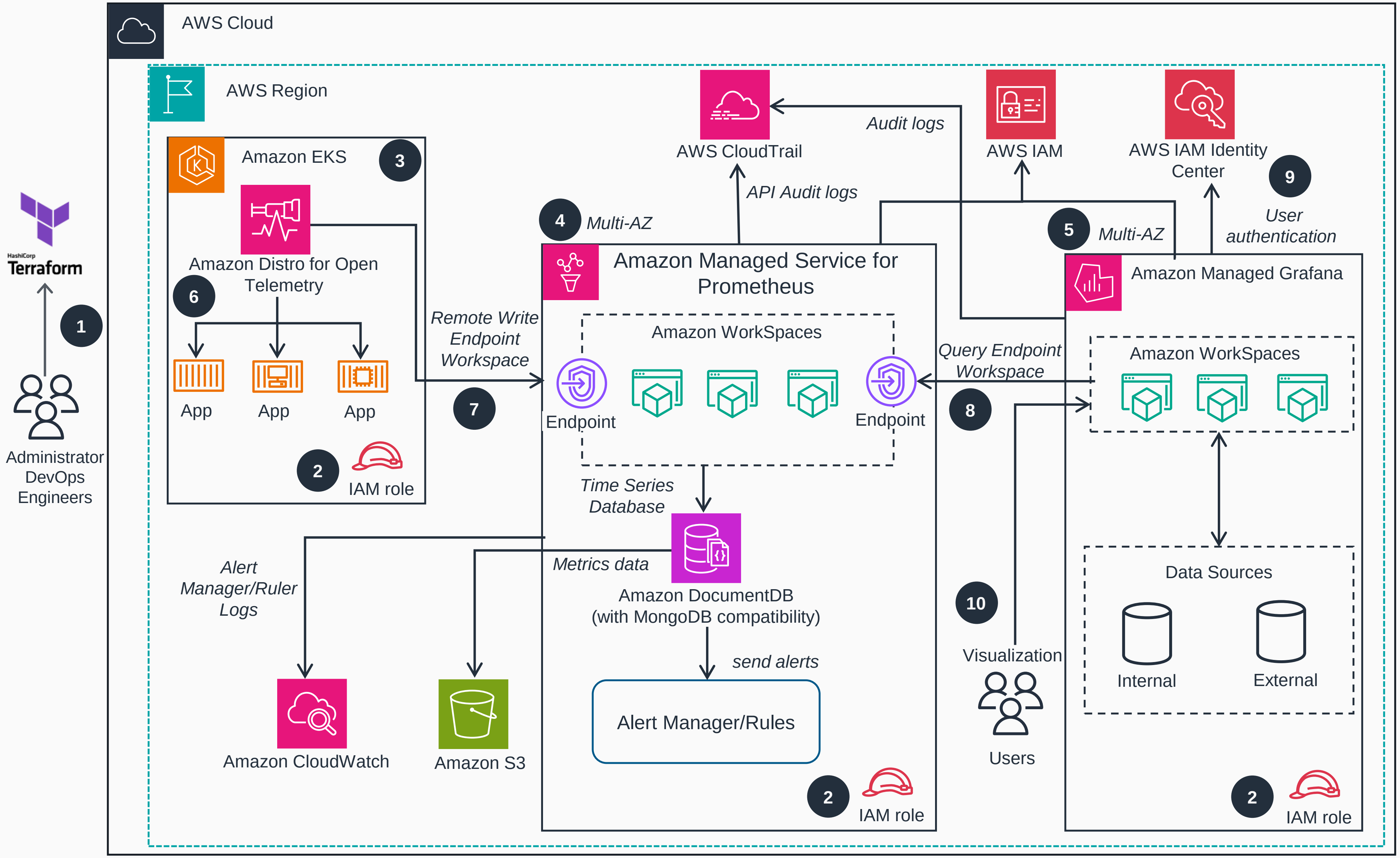
10 The **Amazon EKS** c**luster** is available for application deployment. The Kubernetes API is accessible for the command line interface (CLI) clients and applications through a Network Load Balancer (NLB).

Reviewed for technical accuracy December 13, 2023

AWS Reference Architecture

Guidance for Monitoring Amazon EKS Workloads using Amazon Managed Services for Prometheus & Grafana

This slide demonstrates the deployment of AWS Observability Accelerator on an Amazon Elastic Kubernetes Service (Amazon EKS) cluster.



1 The administrator or DevOps team users initiate the installation of the AWS Observability Accelerator through a Terraform blueprint.

2 The required **IAM** roles and polices are created by Terraform.

3 **AWS Distro for OpenTelemetry** collector resources are deployed into an **Amazon EKS** cluster by Terraform.

4 **Amazon Managed Service for Prometheus** is deployed in multiple Availability Zones (multi-AZ) and configured with alerts and rules by Terraform.

5 **Amazon Managed Grafana** is deployed in multi-AZ mode. It's integrated with **Amazon Managed Service for Prometheus** and other services by Terraform.

6 Metrics are collected from microservices, pods, or jobs running on the **Amazon Elastic Kubernetes Service** (Amazon EKS) cluster by the **Distro for OpenTelemetry** collector.

7 Collected metrics are exported to **Amazon Managed Service for Prometheus** through the writer endpoint, and stored in a time-series database. Metrics can be exported to **Amazon Simple Storage Service** (Amazon S3). Alert rules are created in **Amazon Managed Service for Prometheus** based on metric thresholds.

8 Imported metrics are available for queries to **Amazon Managed Grafana** through the query endpoint of the data source.

9 Users authenticate to **Amazon Managed Grafana** through **AWS IAM Identity Center** (or another Single Sign-on provider).

10 Metrics and metadata are available to **IAM** authenticated and authorized users in the **Amazon Managed Grafana** user interface through dashboards.

Reviewed for technical accuracy December 13, 2023

AWS Reference Architecture