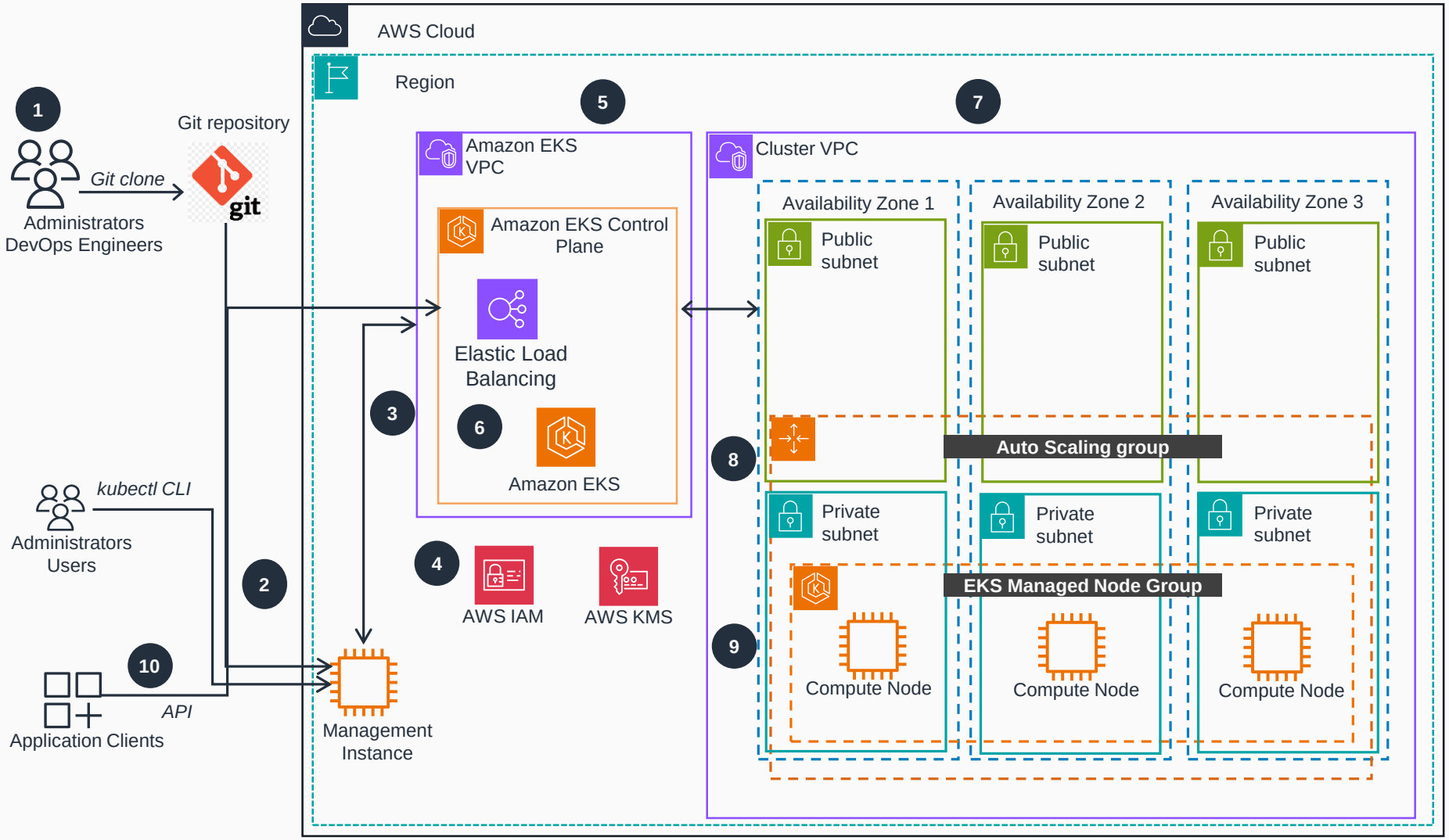
Guidance for Low Latency, High Throughput Inference Using Efficient Compute on Amazon EKS

(**OPTIONAL**) This architecture diagram shows how to setup an Amazon Elastic Kubernetes Service (Amazon EKS) cluster that is compatible with this Guidance. An Amazon EKS cluster is needed to deploy this Guidance.



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 using "provision" part of the project code.

1. Administrator or DevOps user obtains Infrastructure as Code (IaC) code with **Amazon EKS** specification from Git repository.

2. **Amazon Elastic Compute Cloud (Amazon EC2)** Management Instance provisioning is started by Admin/DevOps user via the **AWS CloudFormation** code obtained from the Git repo.

3. Management Instance *userData* script starts **Amazon EKS** cluster resource deployment processes against target AWS environment (using *eksctl* command and cluster specification).

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

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

6. The **Amazon EKS** cluster control plane components are deployed into the **Amazon EKS** VPC. The cluster control plane is provisioned across multiple Availability Zones and fronted by Elastic Load Balancing **(ELB**).

7. Cluster VPC is deployed for the **Amazon EKS** compute plane.

8. Public and Private subnets and other networking components are deployed in cluster VPCs.

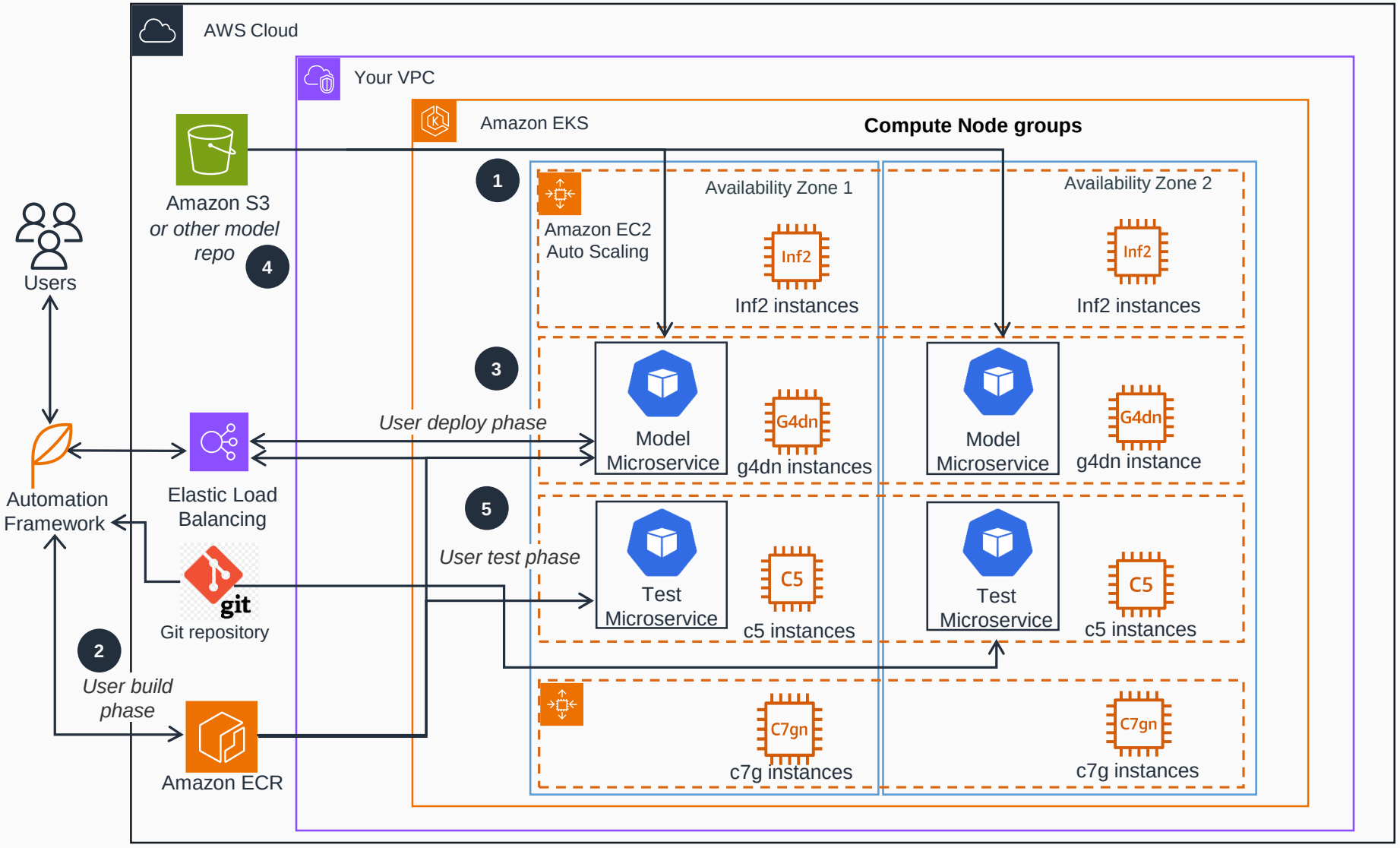
9. The **Amazon EKS** compute plane node groups. containing **Amazon Elastic Compute Cloud** (Amazon EC2) node instances in auto scaling groups, are deployed into the cluster VPC and join the **Amazon EKS** cluster.

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 an **ELB**.

AWS Reference Architecture

Guidance for Low Latency, High Throughput Inference Using Efficient Compute on Amazon EKS

This Guidance demonstrates a simple, scalable, and highly available architecture for running machine learning (ML) inference pipelines on AWS. It uses a standard Amazon Elastic Kubernetes Service (Amazon EKS) infrastructure that can be deployed across multiple Availability Zones for high availability.



1. The **Amazon Elastic Kubernetes Service (Amazon EKS)** cluster has several compute node groups with one **Amazon Elastic Compute Cloud (Amazon EC2)** instance family per node group. Each node group can support different instance types, such as AWS Graviton Processors (c7g) or AWS Inferentia processors (inf2)-based instances deployed across Availability Zones (AZs).

2. The natural language processing (NLP) models, serving application and machine learning (ML) framework dependencies, are built by users as container images using the automation framework. These images are uploaded to **Amazon Elastic Container Registry (Amazon ECR)**. Decoupling the model container images from the model data reduces the size of the model container images.

3. Using the automation framework, the model container images customized for each compute node instance, are obtained from the respective **Amazon ECR** repositories. They are deployed to the **Amazon EKS** cluster using generated deployment manifests via Kubernetes API exposed through **Elastic Load Balancing (ELB)**.

4. ML model application containers download the model artifacts from the model repository, such as **Amazon Simple Storage Service (Amazon S3)**, or other repositories upon their initialization. This component of the architecture decouples the model data from its service definition. ML inference services are available in the **Amazon EKS** cluster.

5. Load testing of the deployed ML inference services is performed using containerized test clients deployed using images from **Amazon ECR** repository by the automation framework. Test clients send simultaneous requests to the ML model service pool running in the **Amazon EKS** cluster. Performance Test results are obtained and aggregated.

Reviewed for technical accuracy February 5, 2024

AWS Reference Architecture