Anthos on AWS Deep Dive

Google Cloud

Agenda

5m	Prerequisites
15m	Architecture Components
10m	Implementation
5m	Operator Workflows
5m	Client Access to Services

Google Cloud

Prerequisites

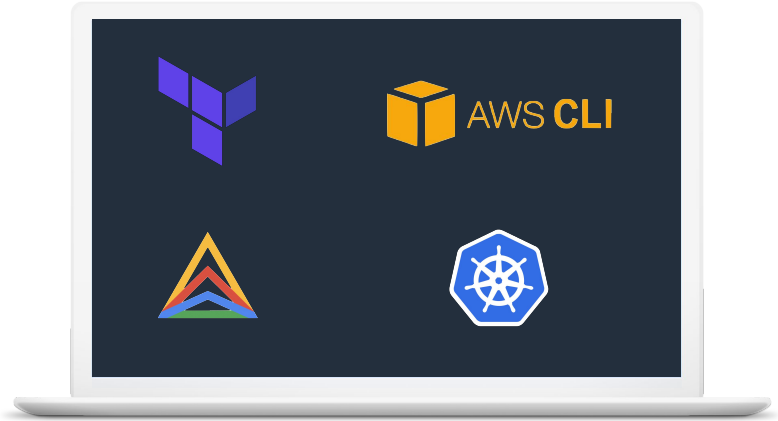
The following prerequisites are required to be performed prior to deploying Anthos on AWS:

- An active **Google Cloud Project**
- A user with a Google account with **Project Owner permissions**
- **Enable APIs** in the Google Cloud project
- GCP Service Account with **GKE Hub permissions**
- **AWS IAM user** with the required IAM Permissions

Required Tooling

The following tools are required for creating your Anthos on AWS environment. The tools require a Linux host with bash shell to issue commands.

- **aws-cli** - creates AWS KMS key
- **anthos-gke** - generates Terraform for the environment and gets GKE on AWS credentials
- **Terraform** - bootstrapping environment in AWS
- **kubectl** - creates and interacts with User Clusters

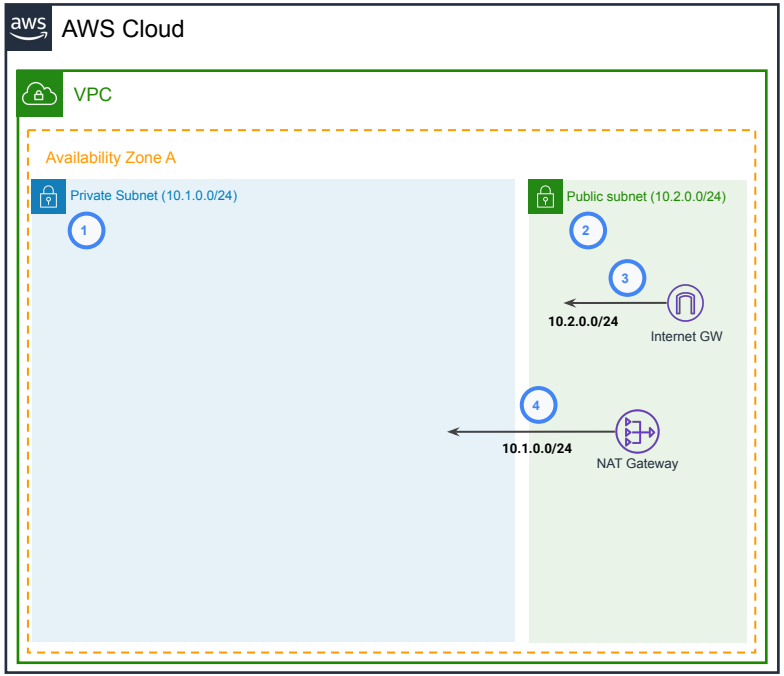


Architecture Components - New / Existing VPCs

A **new VPC** can be created using Terraform or alternatively an **existing VPC** can be defined.

To integrate with an existing VPC, the following must be defined:

- Required AWS IAM permissions
- An existing AWS VPC with:
  - (1) At least one public subnet
  - (2) At least one private subnet
  - (3) An internet gateway with a route to the public subnet
  - (4) A NAT gateway with a route to the private subnet
  - (5) DNS hostnames enabled



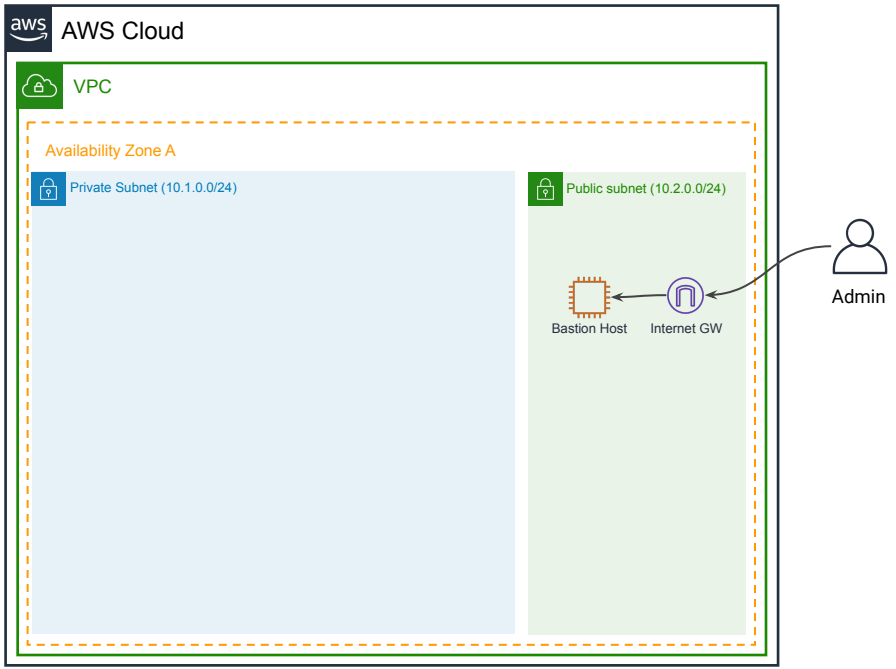
Architecture Components - Bastion Host

The Management Service uses a **private IP address** which isn't accessible from outside the AWS VPC.

For external access, the default configuration includes a bastion host in a public subnet.

- Source CIDR ranges are whitelisted by Security Groups, defined during the initial bootstrap

The **Bastion host is optional**, access to the management service can transit via a private connection e.g. Direct Connect.

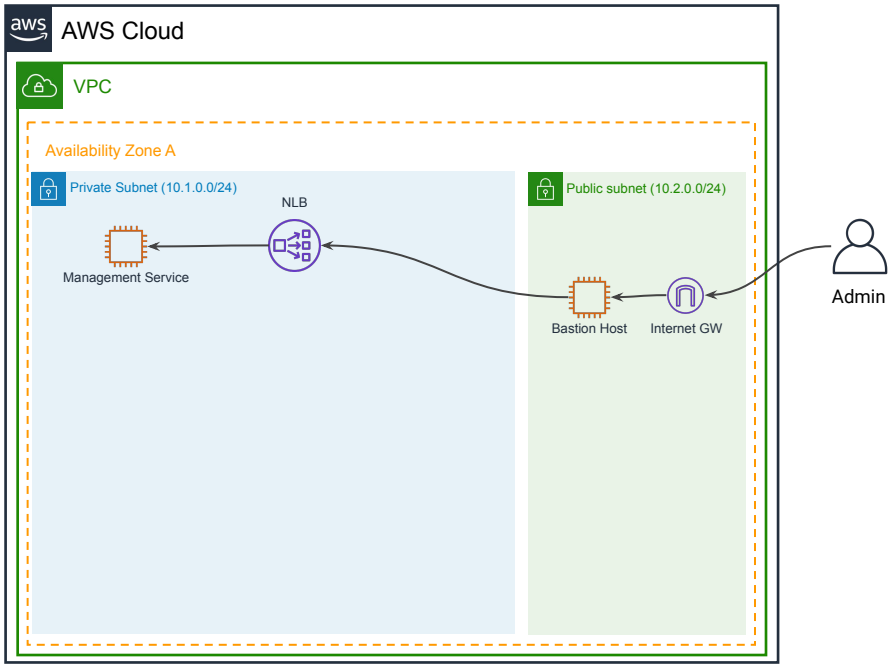


Architecture Components - Management Service

The GKE Management Service makes requests to the AWS API to **provision AWS resources** for clusters.

- Deployed in a **single availability zone**, in the same VPC as the clusters it manages
- The management service instance is wrapped in an ASG of size 1 for resiliency

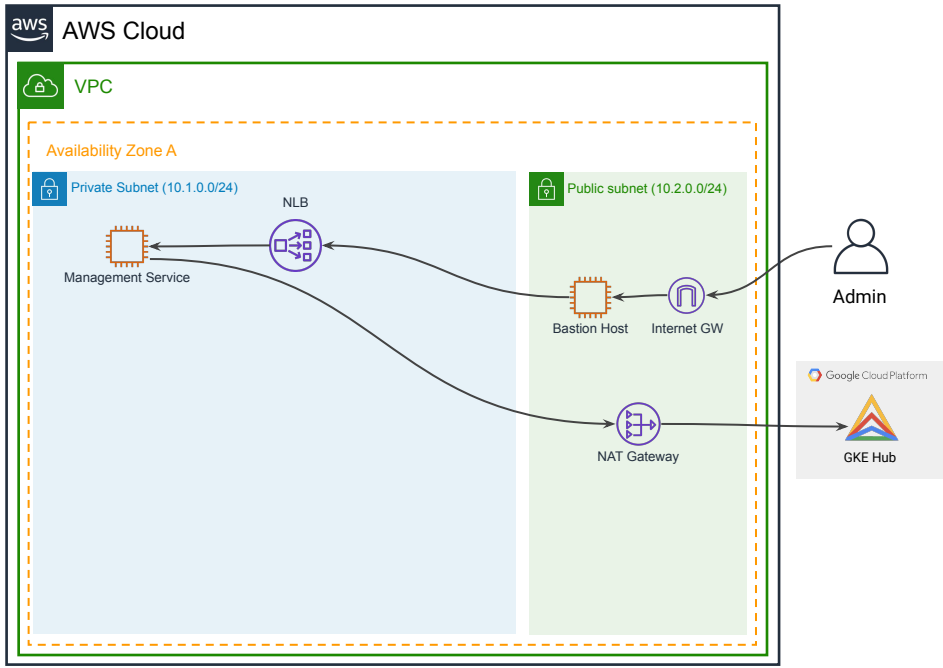
Note: User Cluster operation is not dependant on the Management Service.



Architecture Components - NAT Gateway

Traffic transits the NAT Gateway for outbound internet connectivity.

A proxy can be leveraged for outbound internet traffic.



Architecture Components - User Cluster

The **User Cluster is a GKE cluster** where you run your workloads.

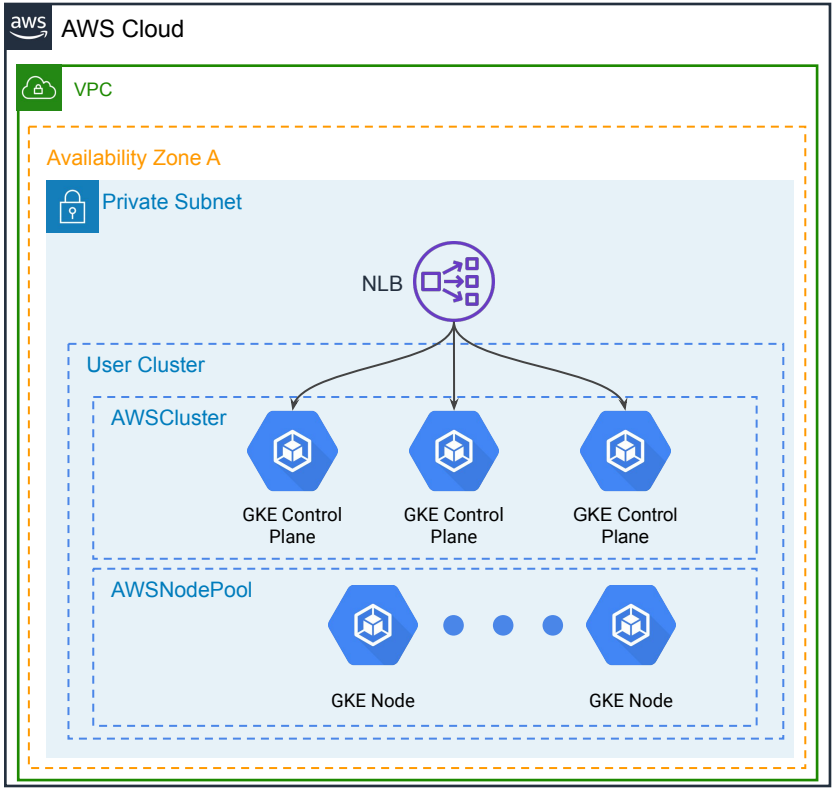
The default deployment creates an `AWSCluster` with three control plane replicas in the same availability zones.

- An AWS NLB is used for the Kubernetes API endpoint

A node pool is a group of nodes within a cluster that all have the same configuration.

- Node pools use a `AWSNodePool` specification
- Each node pool can only span **a single availability zone**

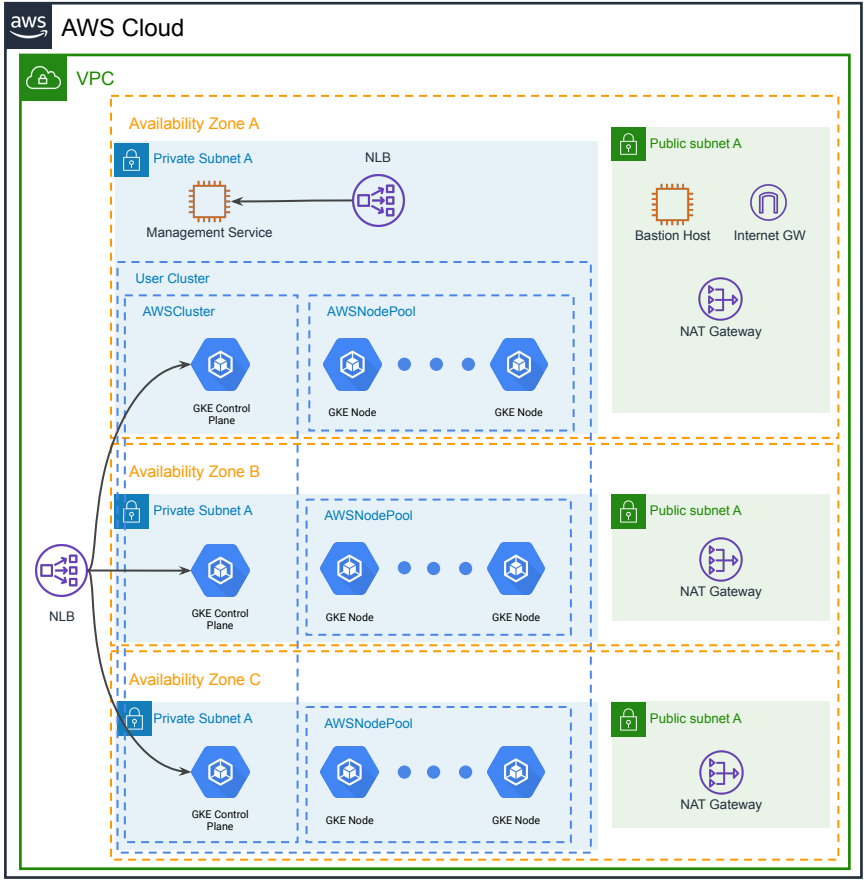
User Cluster instances are wrapped in an ASG for scalability and resiliency.



Architecture Components - High Availability User Cluster

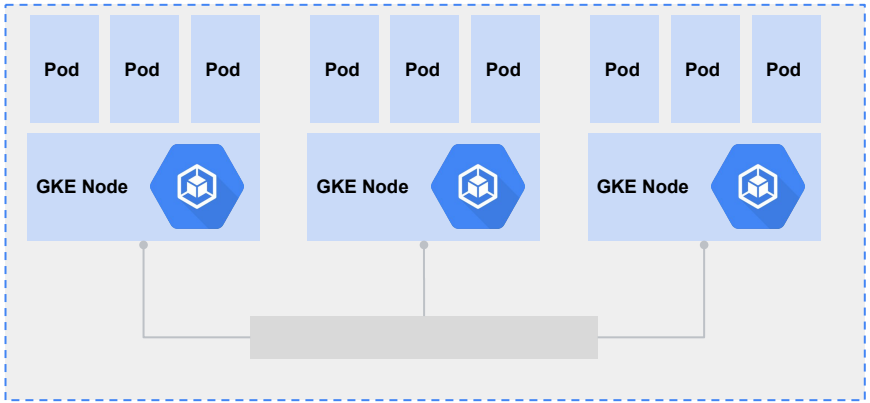
Anthos GKE on AWS supports high availability User Clusters within a VPC.

- `AWSCluster` control plane nodes are deployed **across multiple Availability Zones**.
- `AWSNodePools` are deployed **into a single Availability Zones**. Multiple Node Pools are required for availability and resiliency.



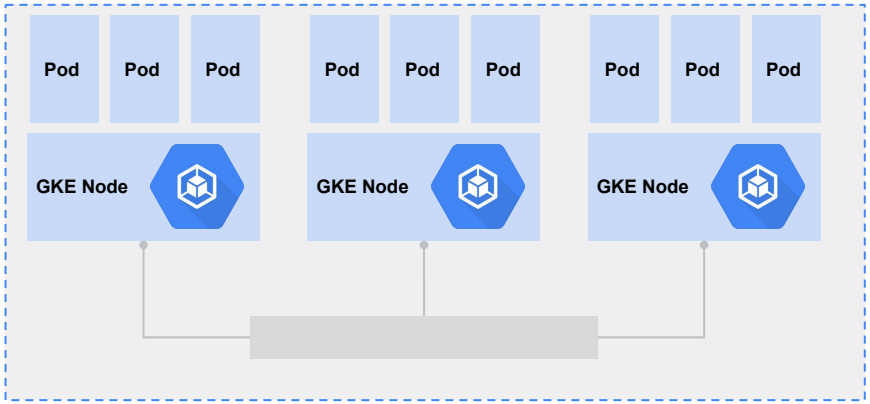
Implementation - Connectivity and Network Policy

- **Island Mode** - Cluster IPs (Pod and Service IPs) form an island within your VPC.
- Cluster IPs (Pods and Services) form a cluster wide **node-to-node mesh using BGP**.
- **Network Policy** is supported via the Calico CNI plugin.



Implementation - IP Subnet Allocation

- CIDR ranges for AWS resources are defined within your Terraform configuration.
- Pods and Services CIDR range are defined within the `AWSCluster` CRD.
- It is important to allocate an IP Range with **adequate IPs for current and future use.**



Implementation - Storage

GKE on AWS provides a number of options for providing Persistent Volumes for your workloads.

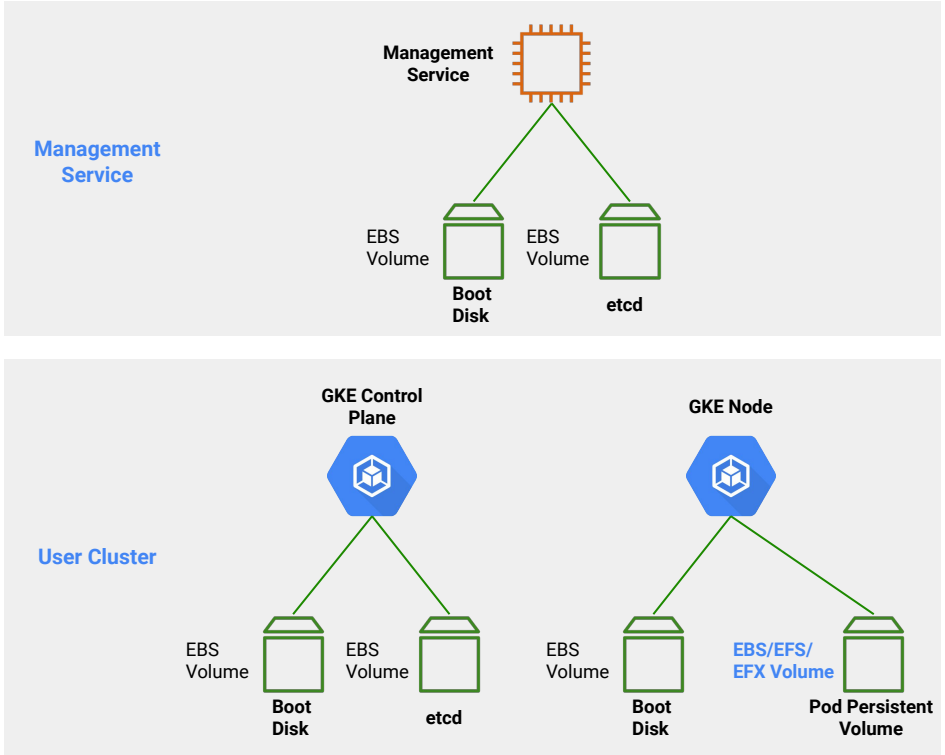
By default AWS EBS volumes are provisioned with the `aws-ebs-csi-driver` with either:

- `standard-rwo` StorageClass with `gp2` volumes
- `premium-rwo` StorageClass with `io1` volumes

Alternate storage volumes such as AWS EFS and FSX are available.

Existing EBS volumes can be imported into GKE on AWS.

All volumes are encrypted by default.

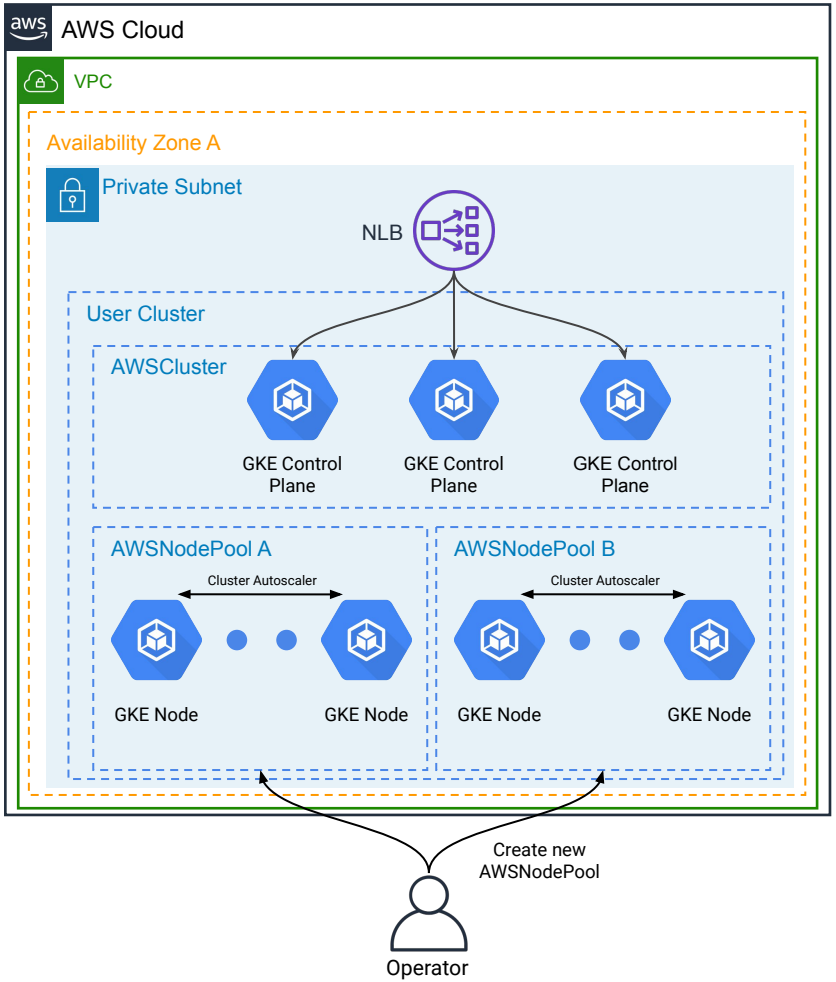


Implementation - Scaling

Automatically scale out your `AWSNodePools` using Cluster Autoscaler.

- Cluster Autoscaler works on a per-node pool basis and **scales based on resource requests**.
- If the node cannot be drained gracefully after a timeout period (10 minutes), the node is forcibly terminated.

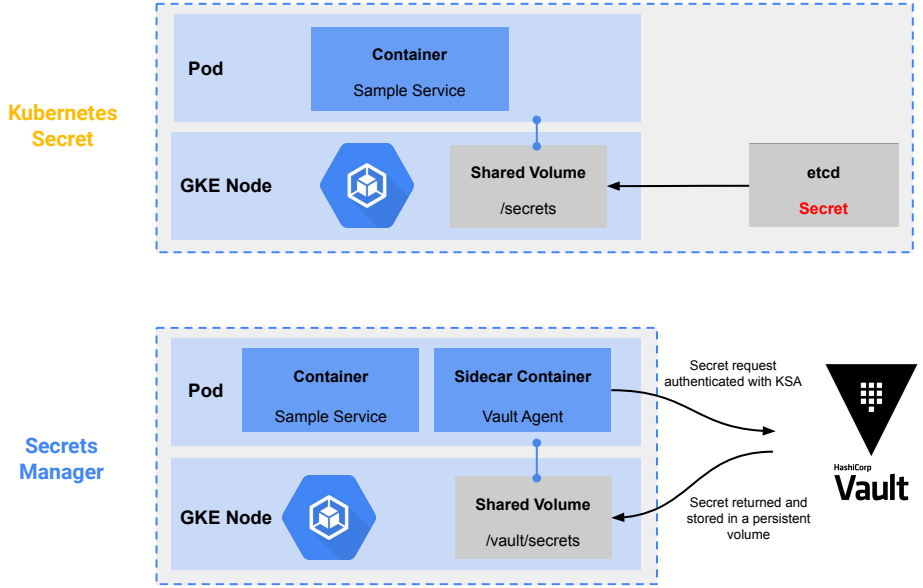
Manually create and delete `AWSNodePools` to scale up, down or across AWS Availability Zones.



Implementation - Secret Management

Secrets can be securely managed in a variety of ways.

- The **Kubernetes Secret** object stores a base-64 encoded representation of your secret in etcd, encrypted using the AWS KMS service (aws-encryption-provider)
- A **Secrets Manager** such as Vault can be used, to access Vault secrets inside Pods, an Agent Sidecar injector is used



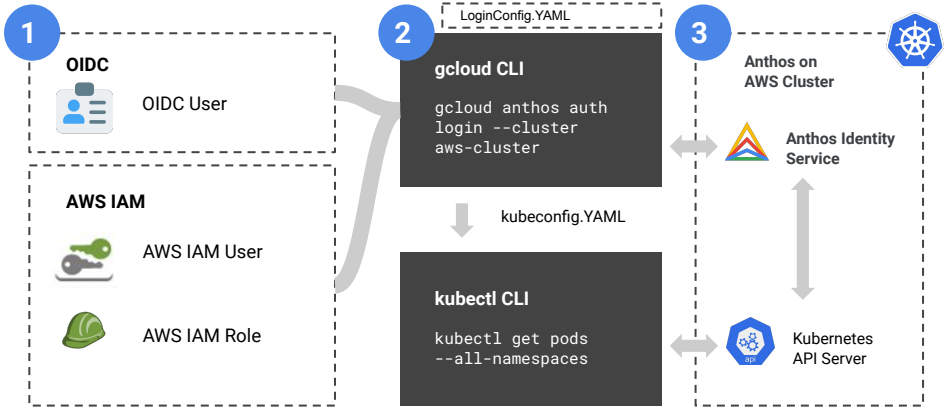
Implementation - User Cluster Authentication

Authenticate using your **existing OIDC provider**, such as ADFS or Okta, to access Anthos GKE clusters on AWS.

- Leverage on the same identity provider as the **single source of truth** across all environments

Authenticate using your **AWS IAM credentials** and maintain consistency with your AWS environment.

- Use the **same credentials to access Anthos GKE clusters on AWS and AWS services**

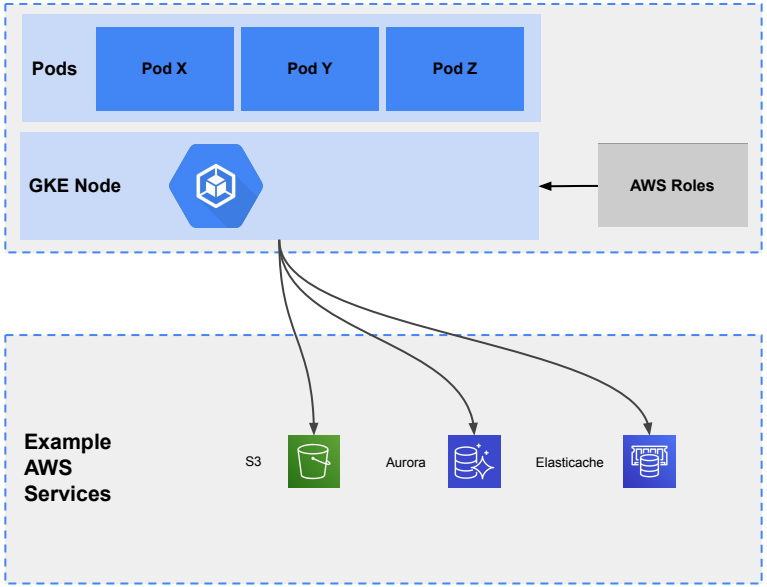


Implementation - Assigning AWS Roles

AWS Roles can be assigned at the **EC2 instance level.**

- Permissions at this level are inherited by all of the Pods running on the Node(s)
- For workloads with differing risk profiles, a separate Node Pool can be used with restrictive permissions

Future releases of Anthos on AWS will target a **workload identity** model, where permissions are applied at the **Pod level.**

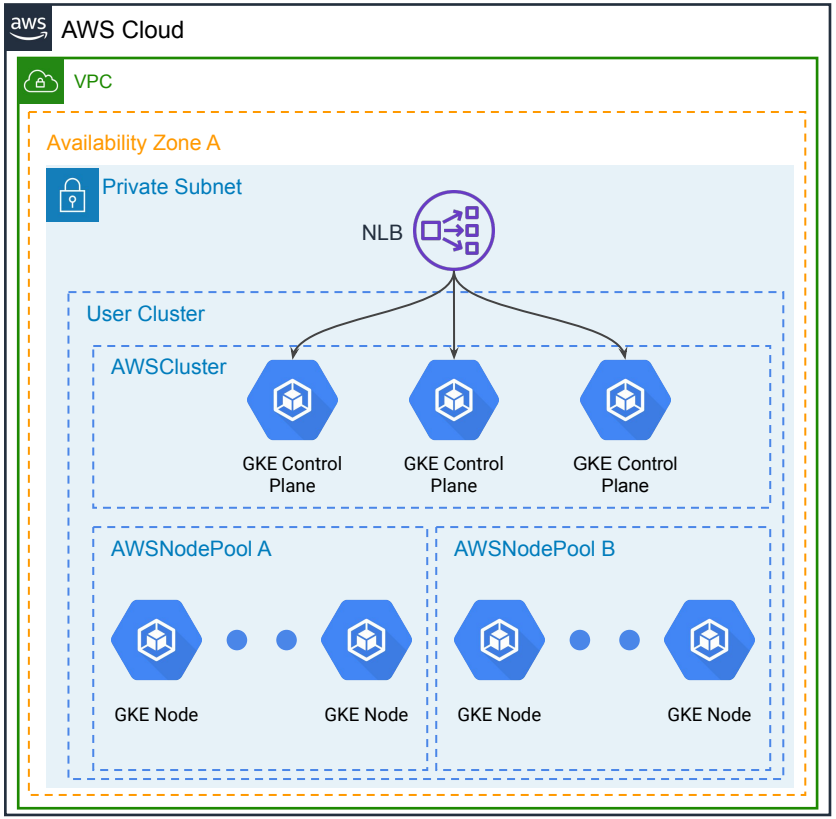


Implementation - Upgrades

Upgrades to Anthos GKE on AWS User Clusters can be performed.

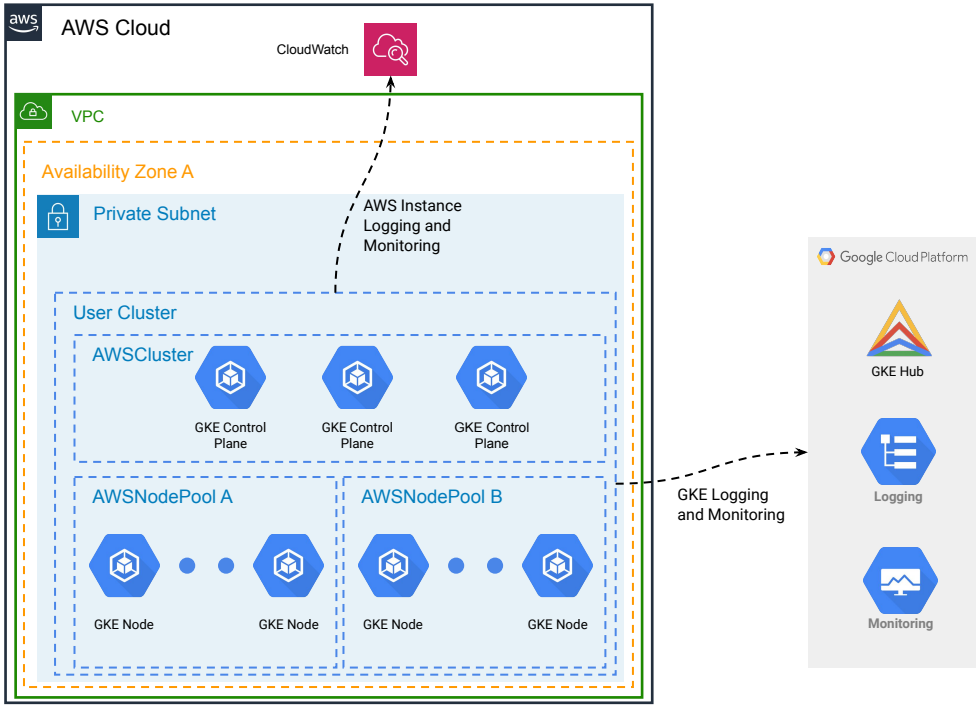
- `AWSCluster` to a new version of GKE on AWS without updating your `AWSNodePools`
- An `AWSNodePool` will not update to a version newer than your `AWSCluster`
- To update your `AWSNodePools`, you must first update your `AWSCluster`

`AWSNodePools` version must be no less than minor versions behind your `AWSCluster` version



Implementation - Logging, Monitoring and Agents

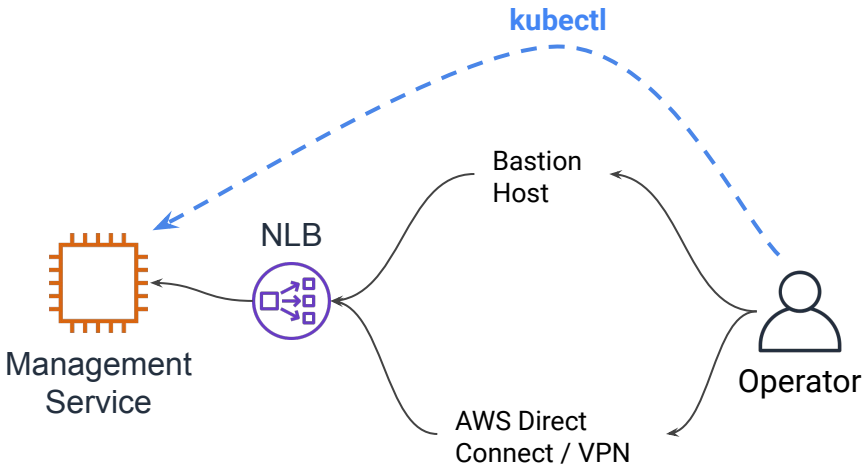
- **Google Cloud Logging** and **Monitoring** can be leveraged to monitor your Anthos GKE clusters
- CloudWatch can be leveraged for **Logging and Monitoring of AWS resources**
- **Existing agents** running on VMs can typically be deployed to Nodes via a Kubernetes DaemonSet



Operator Workflow - Access to Management Service

Operators interact with the Management Service to perform CRUD operations on User Clusters.

- Use `anthos-gke` to connect and authenticate to your GKE on AWS Management Service
- A kubeconfig file is generated (`gke_aws_management.conf`) for connectivity and authentication to the Management Service
- Interacting with the management service is performed using `kubectl`



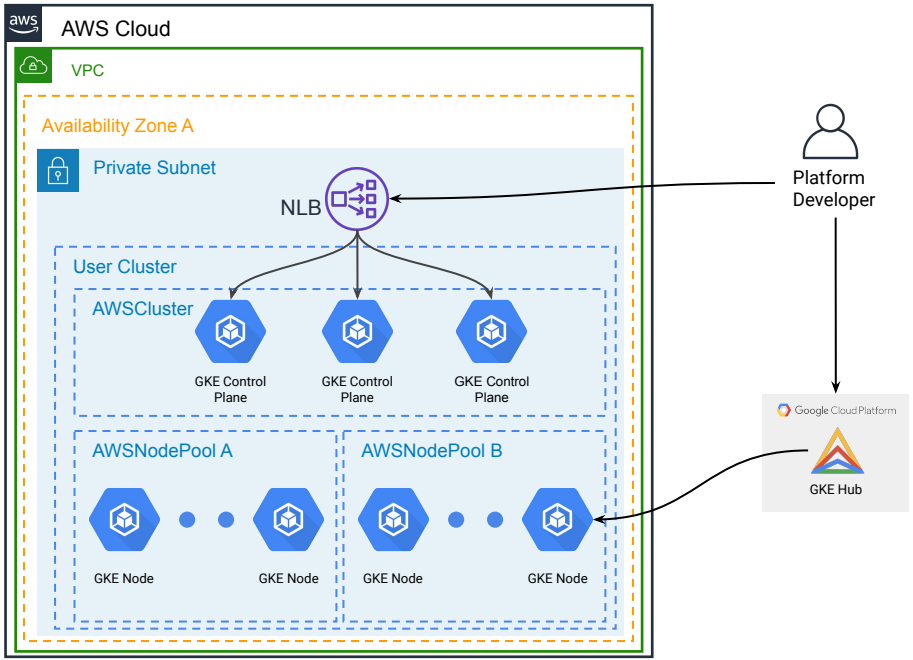
Operator Workflow - Access to User Clusters

The GCP Console can be used to interact with the Kubernetes resources in the User Cluster

- Anthos Connect provides connectivity between the User Cluster and Google Cloud

A Kubernetes API e.g. `kubectl` client can interact with the User Cluster

- GKE on AWS creates a kubeconfig for each user cluster
- By default this file is named `gke_aws_default_<cluster-name>.conf`



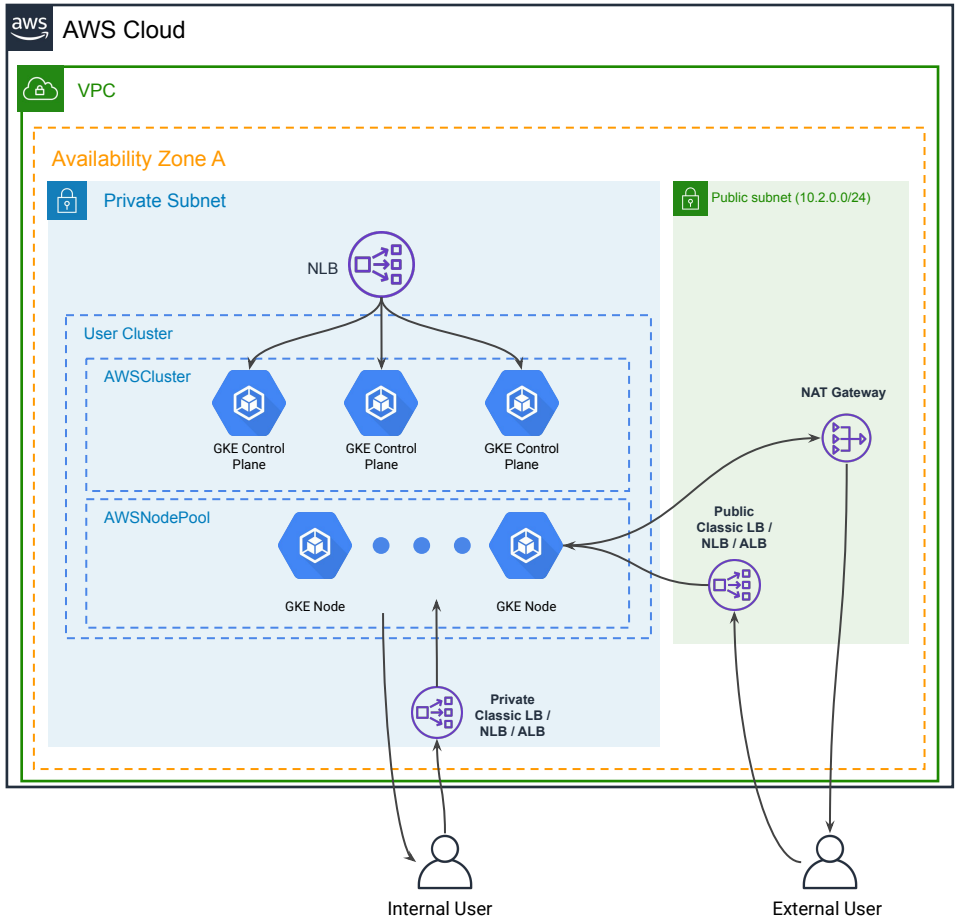
Client Access to Services

A **public or private** Kubernetes **LoadBalancer** can be configured depending on an annotation in your object.

- AWS SGs and the network ACLs control access to a public load balancer
- Classic Load Balancer (Classic ELB), Network Load Balancer (NLB) or Application Load Balancer (ALB) are supported

Ingress is configurable via the use of an Istio Ingress Controller.

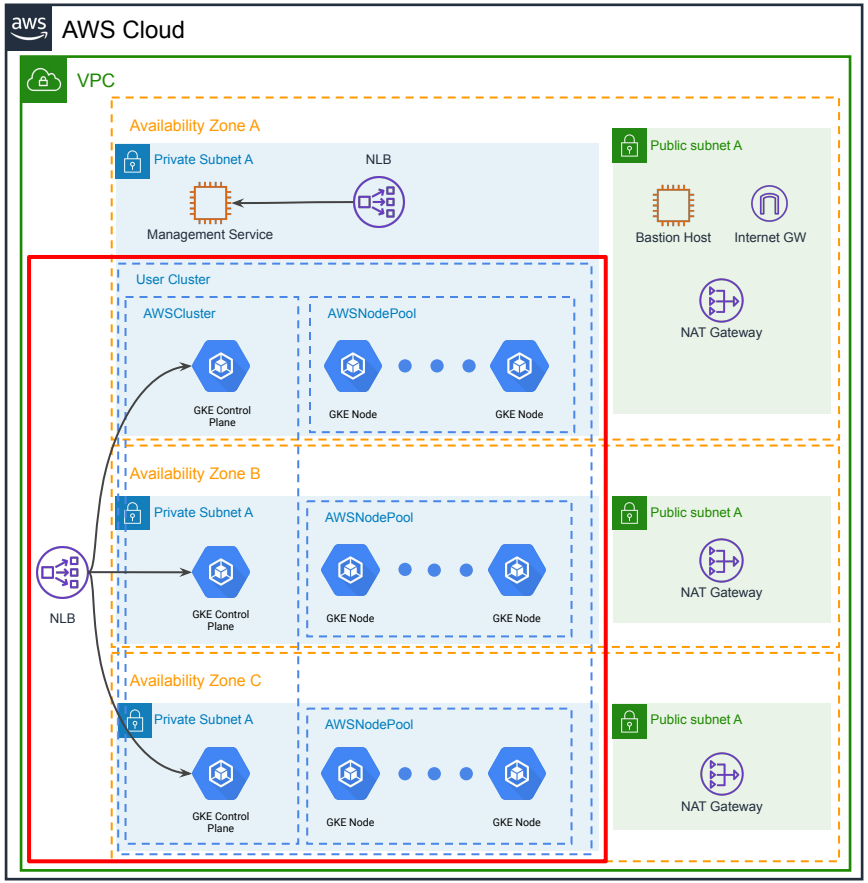
- Deployment is supported into Public and Private subnets



Demo

- Bootstrap Anthos on AWS Environment
- Create User Cluster
- Deploy a Sample Application

Confidential + Proprietary


AWS Cloud
VPC
Availability Zone A
Private Subnet A
NLB
Management Service
Public subnet A
Bastion Host
Internet GW
NAT Gateway
User Cluster
AWSCluster
AWSNodePool
GKE Control Plane
GKE Node
GKE Node
Availability Zone B
Private Subnet A
AWSNodePool
NLB
GKE Control Plane
GKE Node
GKE Node
Public subnet A
NAT Gateway
Availability Zone C
Private Subnet A
AWSNodePool
GKE Control Plane
GKE Node
GKE Node
Public subnet A
NAT Gateway

Anthos

Google Cloud