

# Networking in GCP (Advanced topics)

# Topics to cover:

- Cloud Routing.
- Cloud NAT
- Load Balancing
- Identity Aware Proxy (IAP)
- Cloud DNS
- Virtual Private Network (VPN)
- VPC Service Controls (VPC-SC)
- VPC Peering, Shared VPC and [Cloud Interconnect](#)

# Routing in GCP

Every VPC network uses a scalable, distributed virtual routing mechanism. There is no physical device that's assigned to the network. Some routes can be applied selectively, but the [routing table](#) for a VPC network is defined at the VPC network level.

Each VM instance has a controller that is kept informed of all [applicable routes](#) from the network's routing table. Each packet leaving a VM is delivered to the appropriate next hop of an applicable route based on a routing order. When you add or delete a route, the set of changes is propagated to the VM controllers [by using an eventually consistent design](#).

[Routes overview | VPC | Google Cloud](#)

# Routes types

- System-generated routes:
  - Default routes
  - Subnet Routes
- Custom Routes:
  - Static routes
  - Dynamic routes
- Peering routes:
  - Peering subnet routes
  - Peering custom routes.

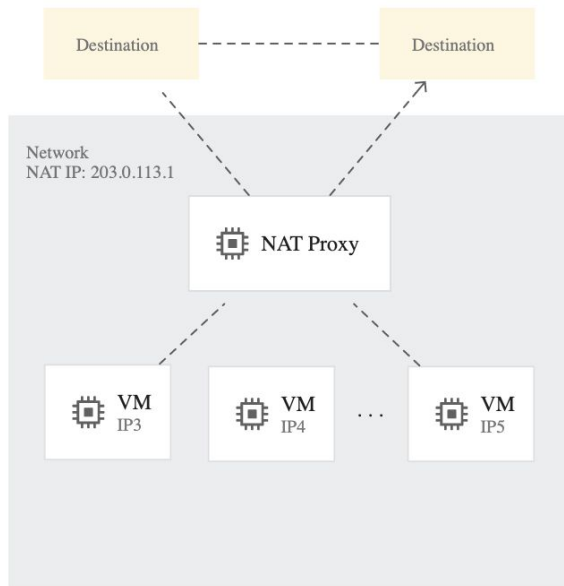
[https://cloud.google.com/vpc/docs/routes#types\\_of\\_routes](https://cloud.google.com/vpc/docs/routes#types_of_routes)

# Cloud Router

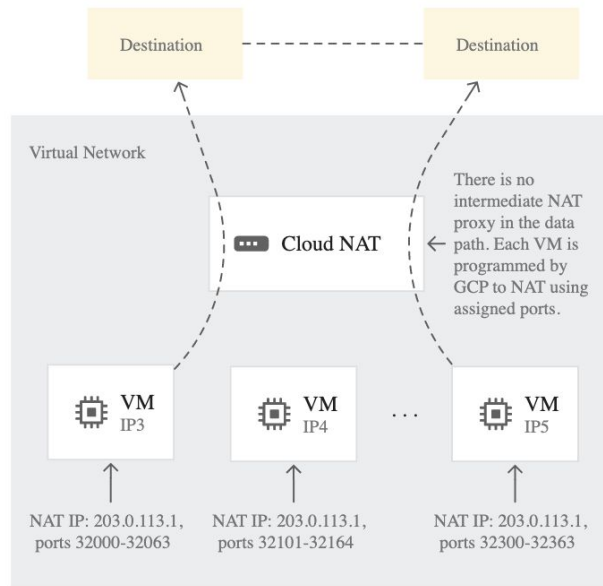
- Cloud Router is used to dynamically exchange routes between two VPCs or between VPC and on-premises networks.
- Cloud Router uses “Border Gateway Protocol” to exchange routing information between the networks.
- A Cloud Router also serves as the control plane for Cloud NAT.
- A Cloud Router is used with services like:
  - HA VPN
  - Classic VPN if your on-premises VPN gateway supports BGP.
  - Dedicated Interconnect & Partner Interconnect
- Direct Peering and Carrier Peering do not use Cloud Routers.

[Cloud Router overview](#)

# Cloud NAT



1. Typical NAT Proxies



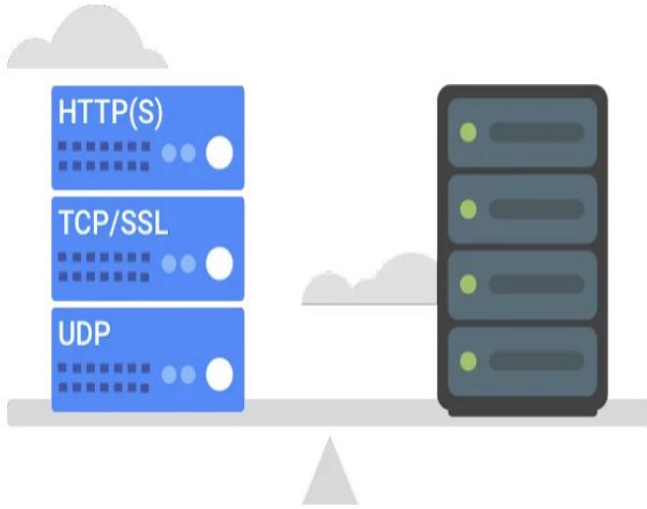
2. Google Cloud NAT

## [Cloud NAT overview](#)

## [Protect Your Network with Cloud NAT](#)

# Cloud Load Balancing

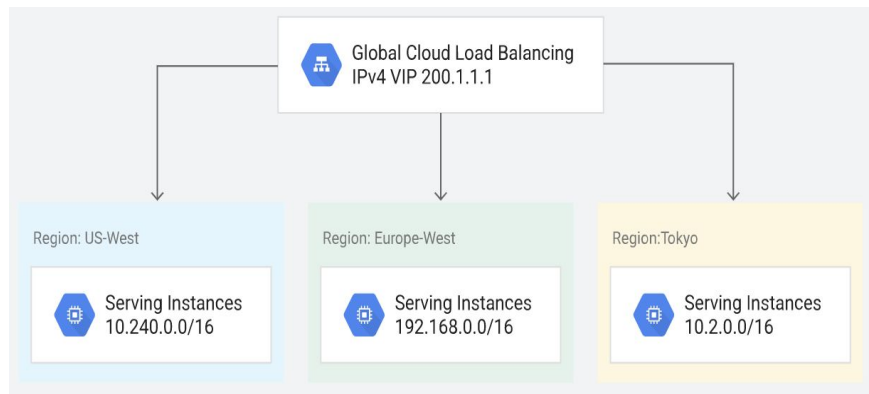
High performance, scalable load balancing on Google Cloud Platform.



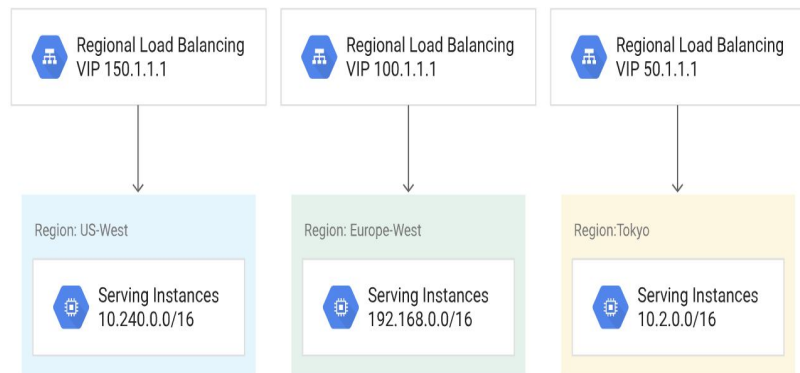
[Cloud Load Balancing documentation](#)

# Cloud Load Balancing - Cont.

## Premium Networking Tier



## Standard Networking Tier





# Cloud Load Balancing - Cont.

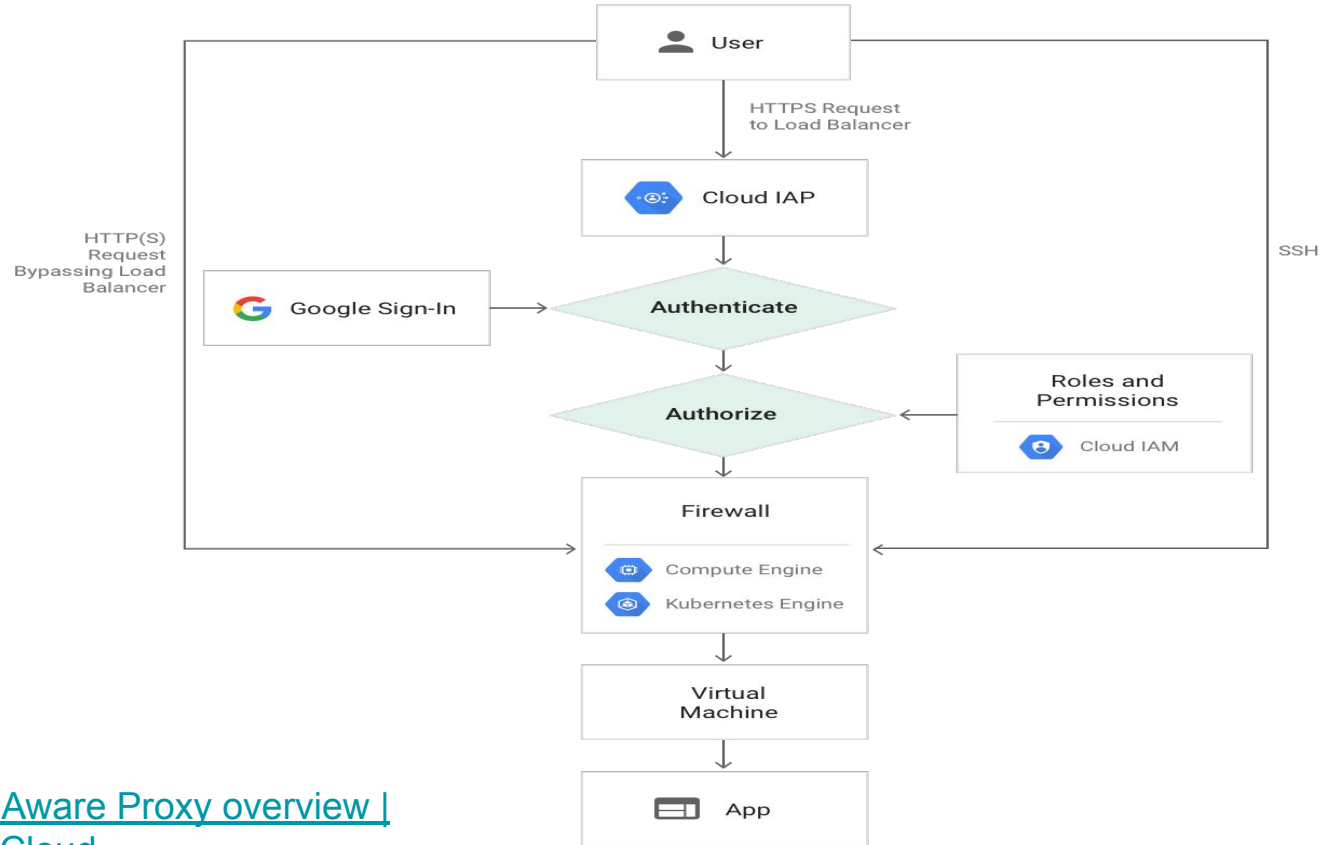
Load balancer types:

- External HTTP(S) Load Balancing (global and regional modes)
- Internal HTTP(S) Load Balancing
- External TCP/UDP Network Load Balancing
- Internal TCP/UDP Load Balancing
- SSL Proxy Load Balancing
- TCP Proxy Load Balancing

[Choosing a load balancer](#)

[Decision tree for choosing a load balancer](#)

# Identity Aware Proxy (IAP)



[Identity-Aware Proxy overview | Google Cloud](#)

# Lab 3.1

1. In previous lab you created the VPC “auto-vpc”, How many routes created for this VPC? Can you delete any of these routes?
2. In previous lab you created a VPC named “custom-vpc” How many routes created for this VPC?
3. How would you block internet access from you vpc using routes?
4. Add a NAT gateway on any of the subnets in your VPC.

# Lab 3.2

1. In previous lab you created an MIG of a template using the custom image “custom-img-nginx”, Create a Global (or Regional) HTTP Load balancer to access your MIGs Nginx setup.
2. Try to configure IAP at the load balancer level to protect your ingress access. Is it possible to have IAP enabled for HTTP resources?

# Google Kubernetes Engine (GKE)

# Topics to cover:

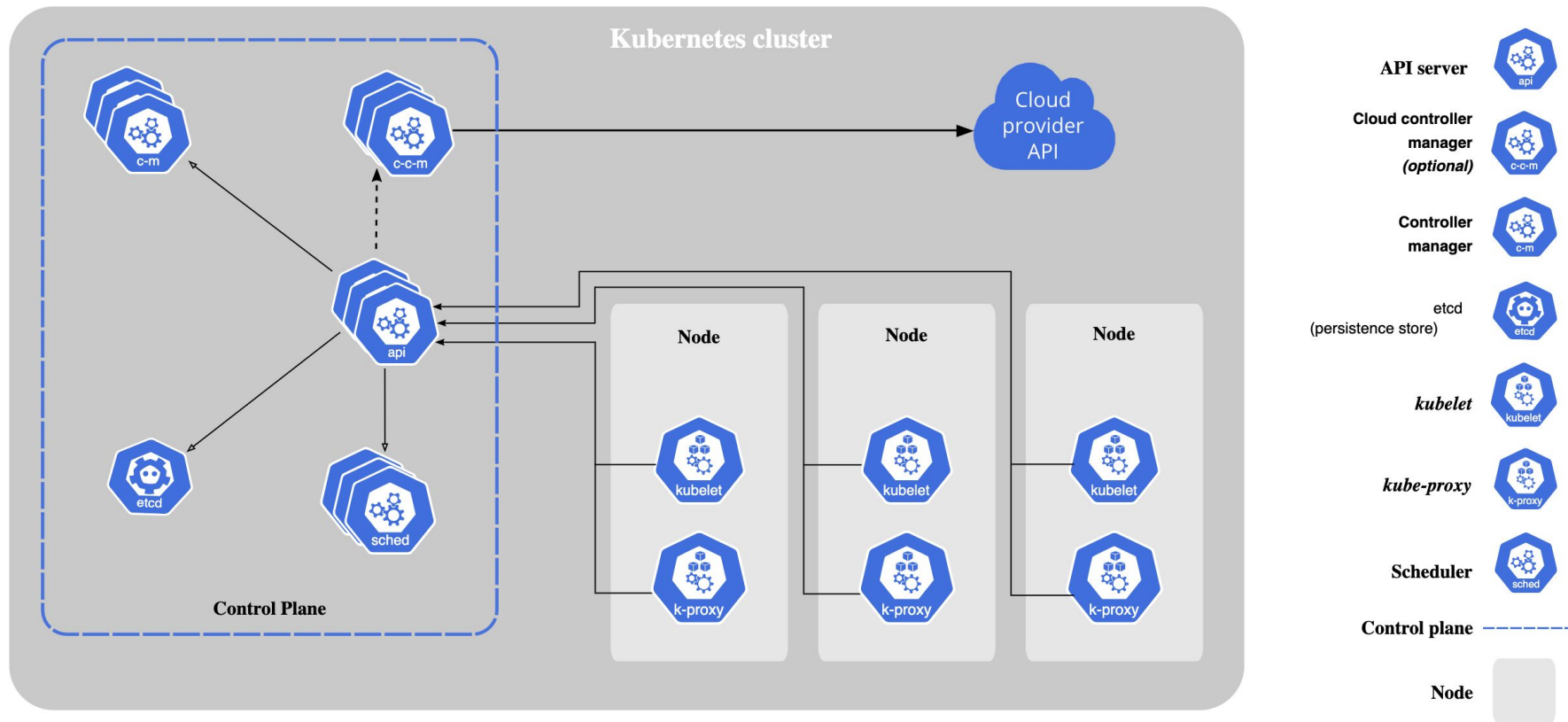
- What is GKE?
- Create and Manage GKE clusters.
- Installing kubectl and connect to GKE cluster
- Viewing GKE cluster state.
- Monitoring deployments on GKE.

# What is GKE? (Cont.)

GKE is a managed service for deploying kubernetes clusters with the benefit of:

- Google Cloud [load-balancing](#) for Compute Engine instances
- [Node pools](#) to designate subsets of nodes within a cluster for additional flexibility
- [Automatic scaling](#) of your cluster's node instance count
- [Automatic upgrades](#) for your cluster's node software
- [Node auto-repair](#) to maintain node health and availability
- [Logging and monitoring](#) with Google Cloud's operations suite for visibility into your cluster

# What is GKE?





# What is GKE? (Cont.)

## Types of GKE clusters:

- Zonal clusters (Single-Zone Or Multi-Zone):
  - A single-zone cluster has a single control plane running in one zone. This control plane manages workloads on nodes running in the same zone.
  - A multi-zonal cluster has a single replica of the control plane running in a single zone, and has nodes running in multiple zones.
- Regional clusters:
  - A regional cluster has multiple replicas of the control plane, running in multiple zones within a given region. Nodes also run in each zone where a replica of the control plane runs.

# What is GKE? (Cont.)

## Modes of GKE clusters:

- Standard mode:

You manage the cluster's underlying infrastructure, giving you node configuration flexibility

- Autopilot mode:

GKE provisions and manages the cluster's underlying infrastructure, including nodes and node pools, giving you an optimized cluster with a hands-off experience.

[GKE Standard vs Autopilot](#)

[Introducing GKE Autopilot](#)

# Create and Manage GKE clusters:

## Create a zonal cluster:

```
gcloud container clusters create example-cluster \  
  --zone us-central1-a \  
  --node-locations us-central1-a,us-central1-b,us-central1-c
```

## Create a regional cluster:

```
gcloud container clusters create my-regional-cluster --region us-west1
```

- [how-to: create zonal cluster](#)
- [how-to: create regional cluster](#)


# Installing kubectl and connect to GKE cluster:

## 1- Install kubectl using gcloud tool:

```
→ .ssh gcloud components install kubectl
```

## 2- Connect to the cluster:

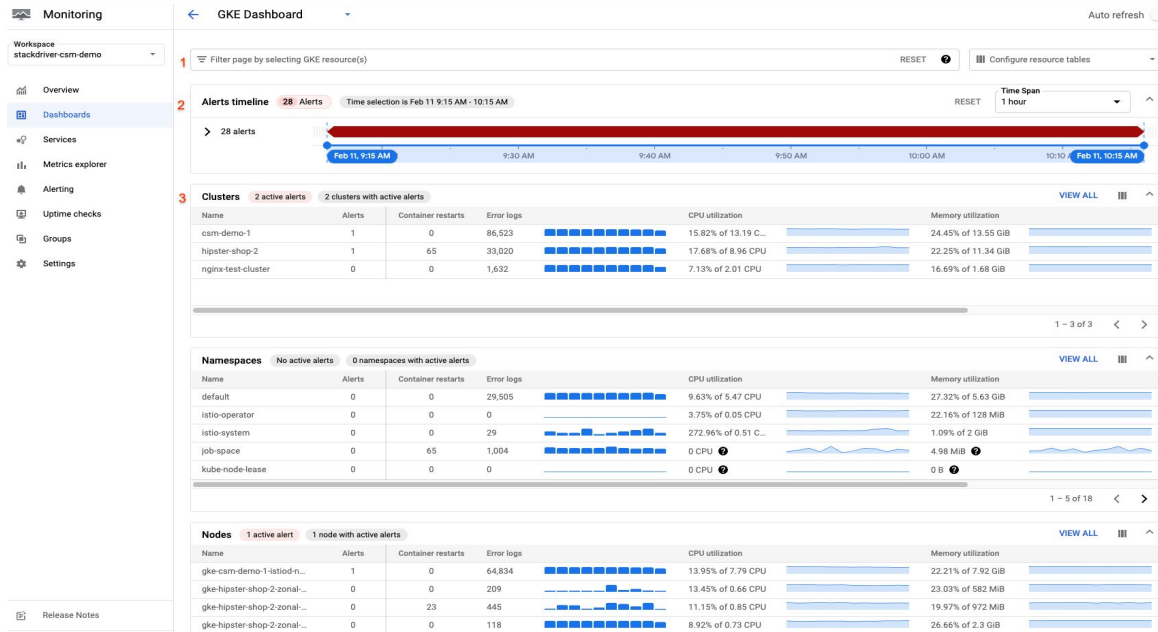
Generating a "kubeconfig" entry

```
gcloud container clusters get-credentials cluster-name 
```



[gcloud components install | Cloud SDK Documentation](#)

# Viewing GKE cluster stats and Monitoring your deployments.



[Observing your GKE clusters | Operations Suite | Google Cloud](#)

# Lab 3.3

1. Create a **private** GKE cluster.
2. Deploy Nginx as a deployment using latest Nginx docker image on Docker Hub.
3. Expose your Nginx deployment using Kubernetes LoadBalancer Service.
4. What is the type of GCP Load Balancer that is created for your LB service?
5. Use kubectl to view container logs.
6. Use cloud logging service to view container logs. [hint: search about cloud logging service for gke]
7. (Bonus) setup a HTTP load balancer for your deployment using the kubernetes ingress resource. (hint: [link](#))
8. Create an autopilot GKE cluster with public control plane.
9. Enforce the cluster's control plane to accept only connections from your local machine.
10. Install kubectl on local machine and use it to connect to the cluster.