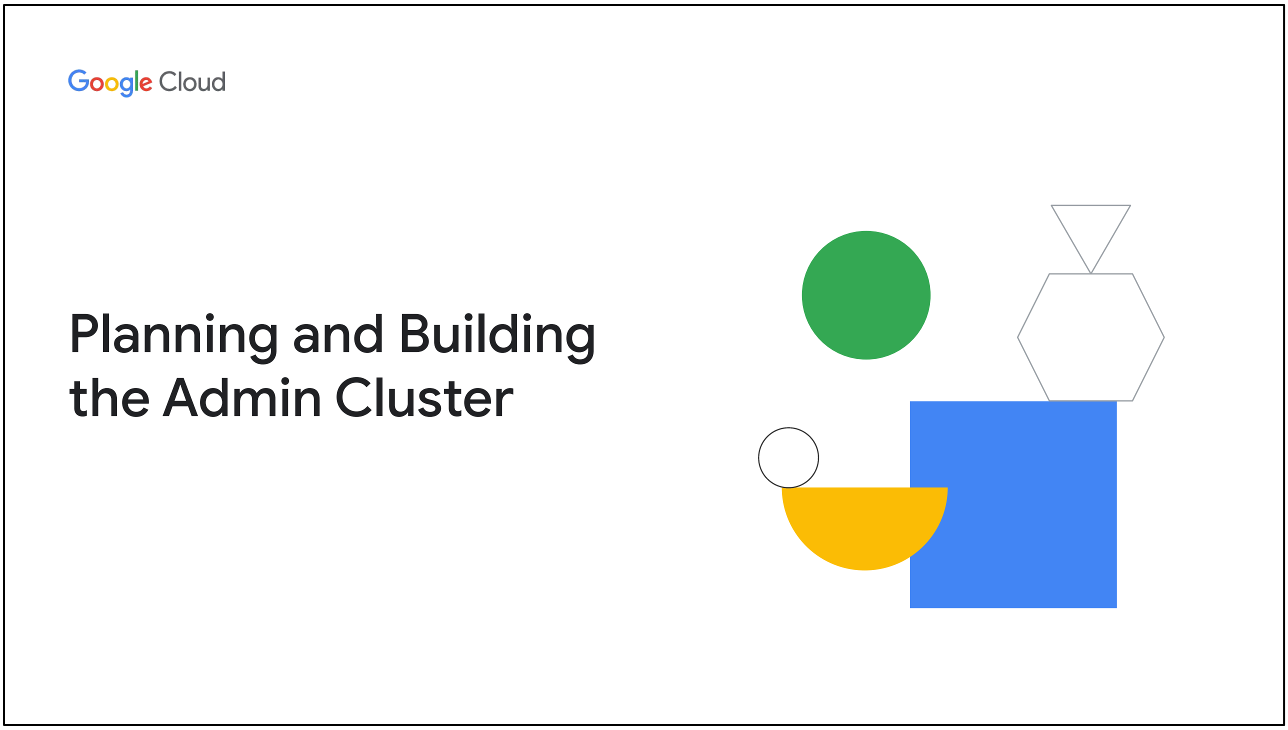
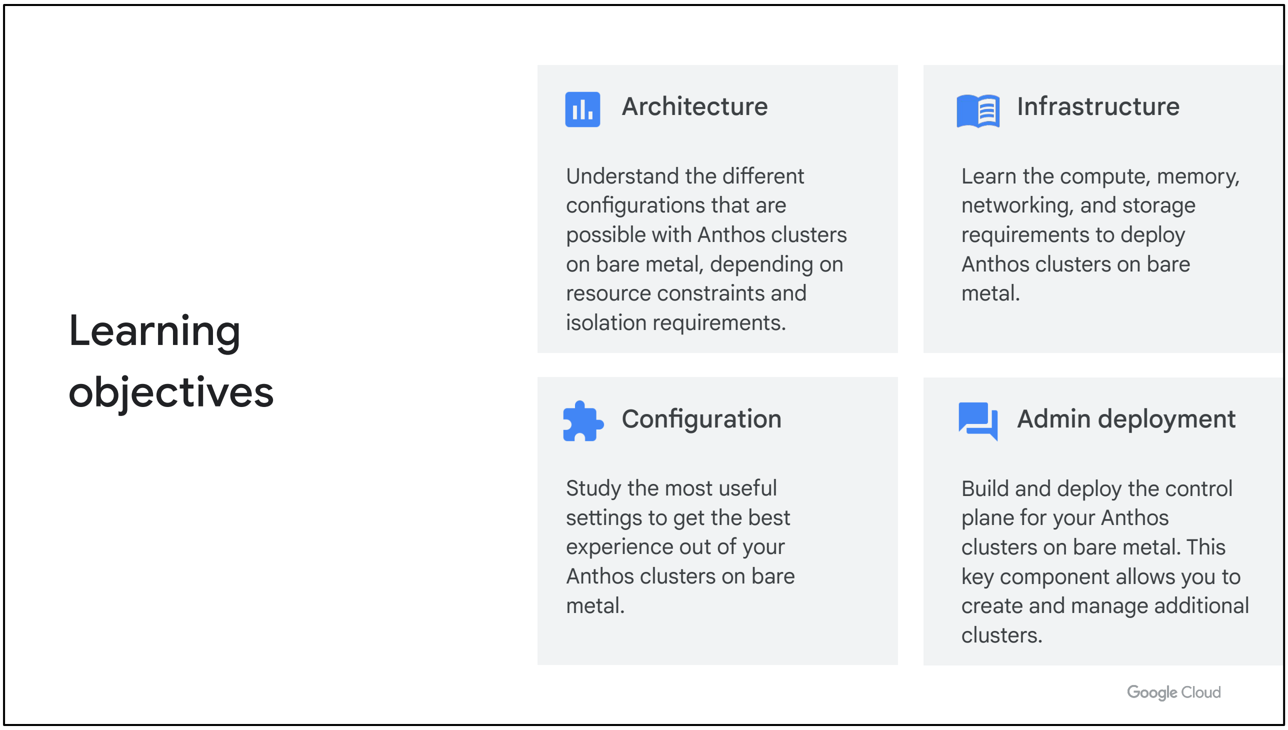


Planning and Building the Admin Cluster

Welcome to Planning and Building the Admin Cluster.



Learning objectives

Architecture

Understand the different configurations that are possible with Anthos clusters on bare metal, depending on resource constraints and isolation requirements.

Infrastructure

Learn the compute, memory, networking, and storage requirements to deploy Anthos clusters on bare metal.

Configuration

Study the most useful settings to get the best experience out of your Anthos clusters on bare metal.

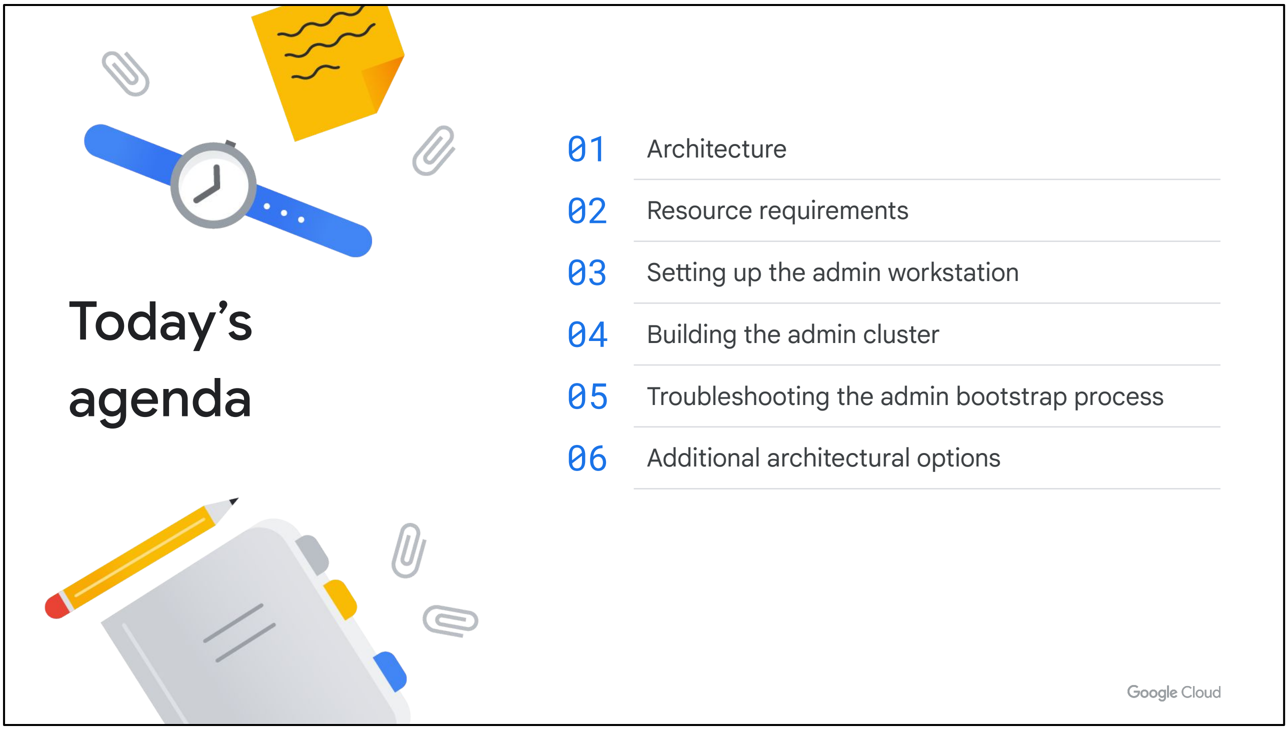
Admin deployment

Build and deploy the control plane for your Anthos clusters on bare metal. This key component allows you to create and manage additional clusters.

Google Cloud

In this module, you learn how to:

- Select from various cluster configurations, taking into account resource constraints and isolation requirements.
- Allocate appropriate resources for Anthos clusters.
- Choose cluster settings to optimize performance and functionality.
- Build the control plane cluster that is used to manage production clusters on bare metal.



Today's agenda

01 Architecture

02 Resource requirements

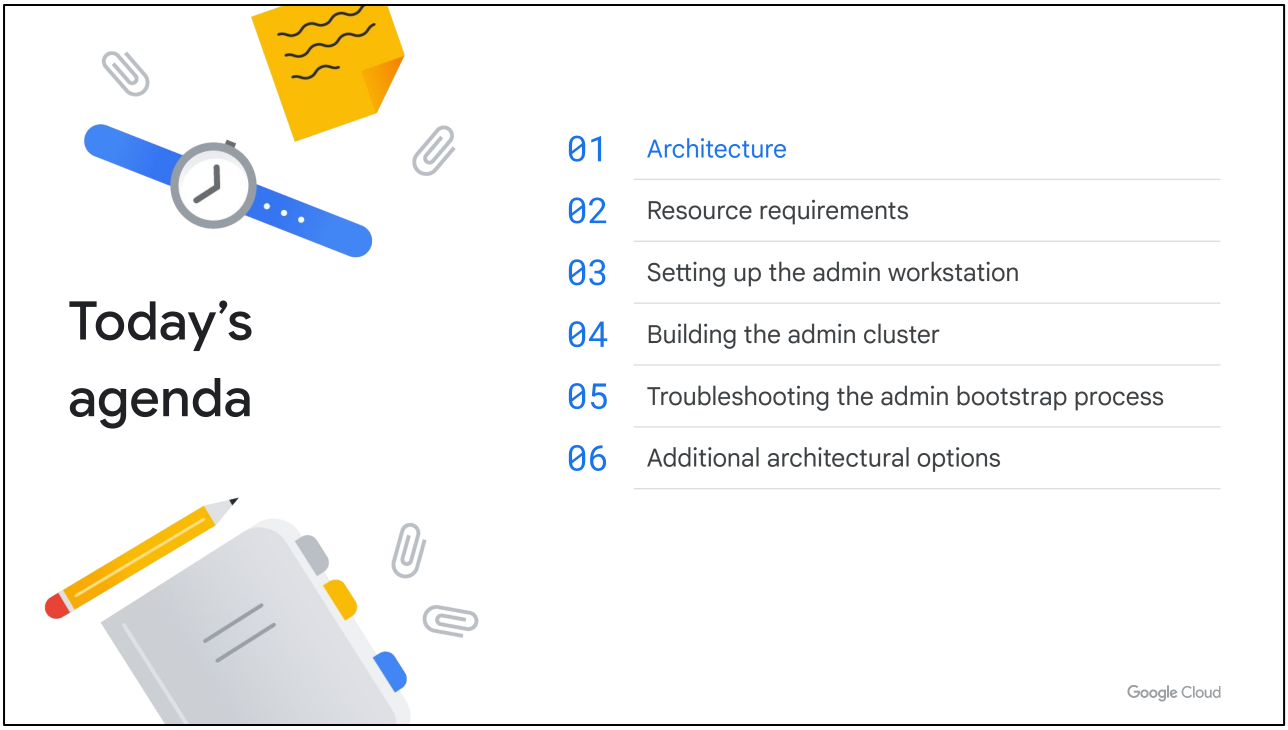
03 Setting up the admin workstation

04 Building the admin cluster

05 Troubleshooting the admin bootstrap process

06 Additional architectural options

Here is our agenda for the module.



Today's agenda

01 Architecture

02 Resource requirements

03 Setting up the admin workstation

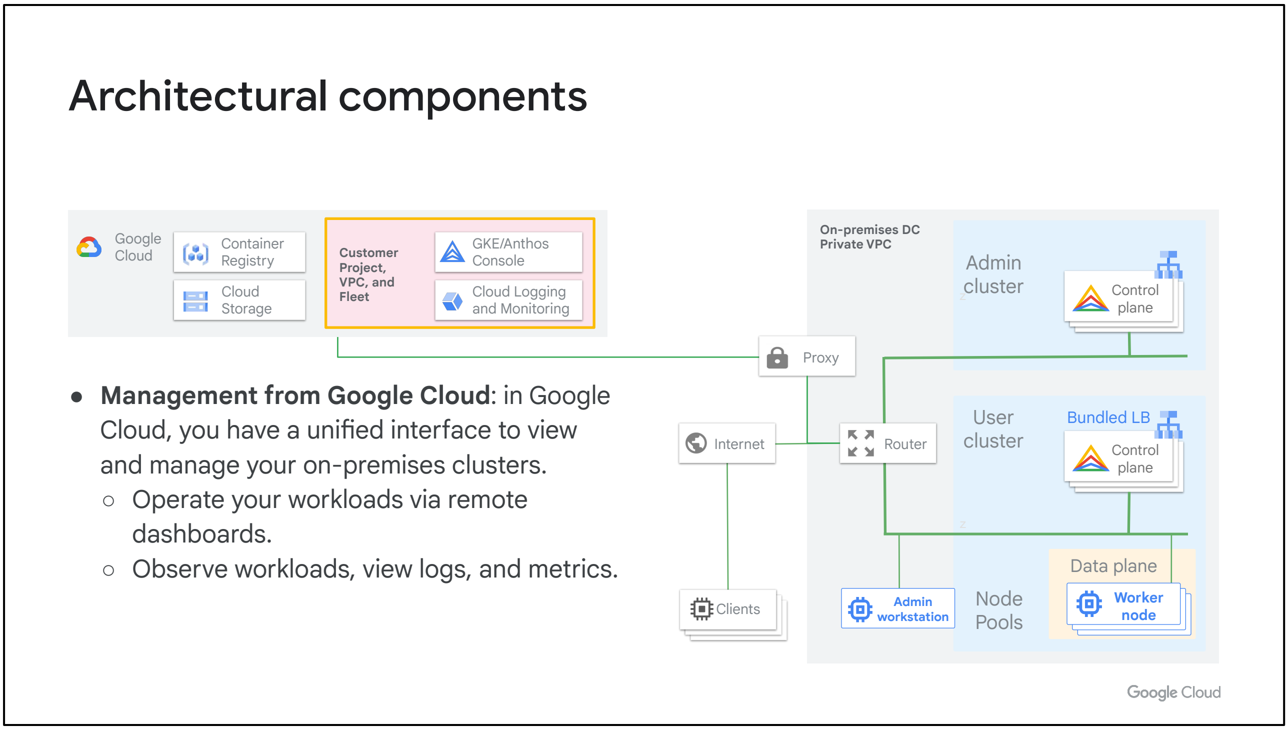
04 Building the admin cluster

05 Troubleshooting the admin bootstrap process

06 Additional architectural options

Google Cloud

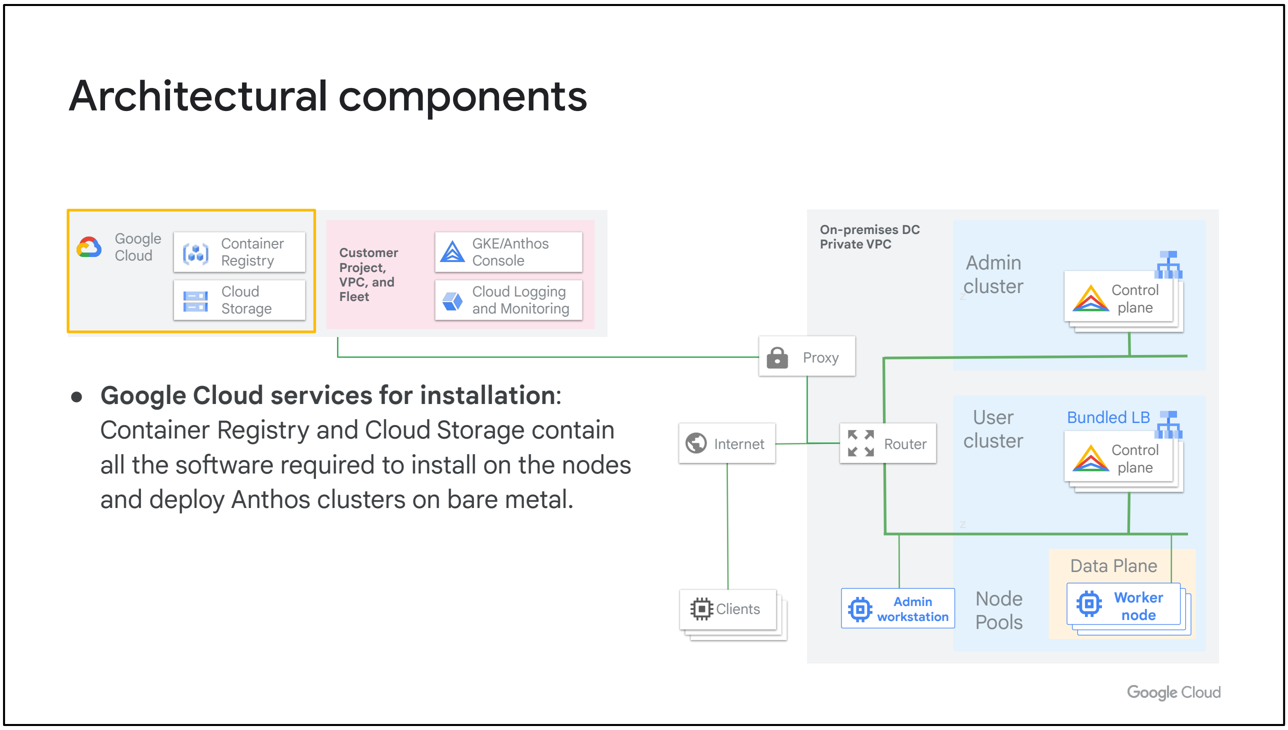
Let's start by getting a clear understanding of the architecture used for Anthos clusters on bare metal.



Architectural components

- **Management from Google Cloud**: in Google Cloud, you have a unified interface to view and manage your on-premises clusters.
  - Operate your workloads via remote dashboards.
  - Observe workloads, view logs, and metrics.

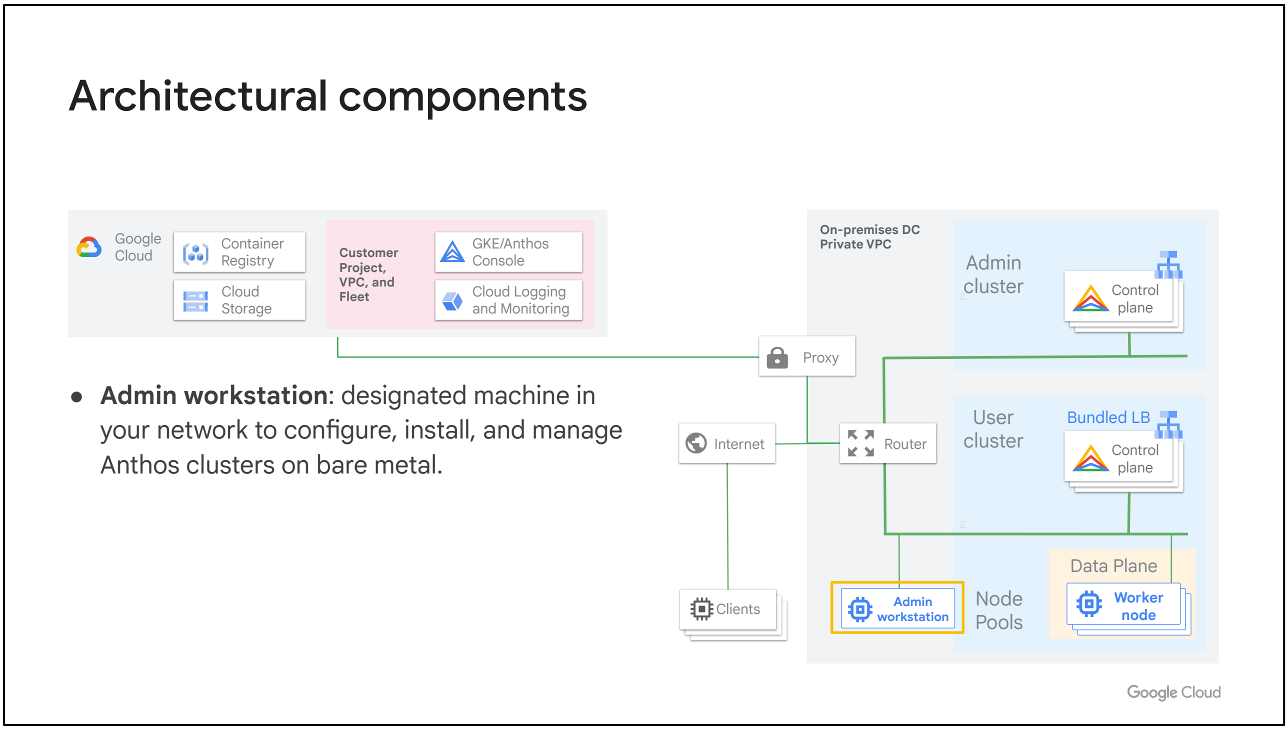
First, Anthos leverages Google Cloud operations tools to make managing your bare metal clusters easy.



Architectural components

- **Google Cloud services for installation**: Container Registry and Cloud Storage contain all the software required to install on the nodes and deploy Anthos clusters on bare metal.

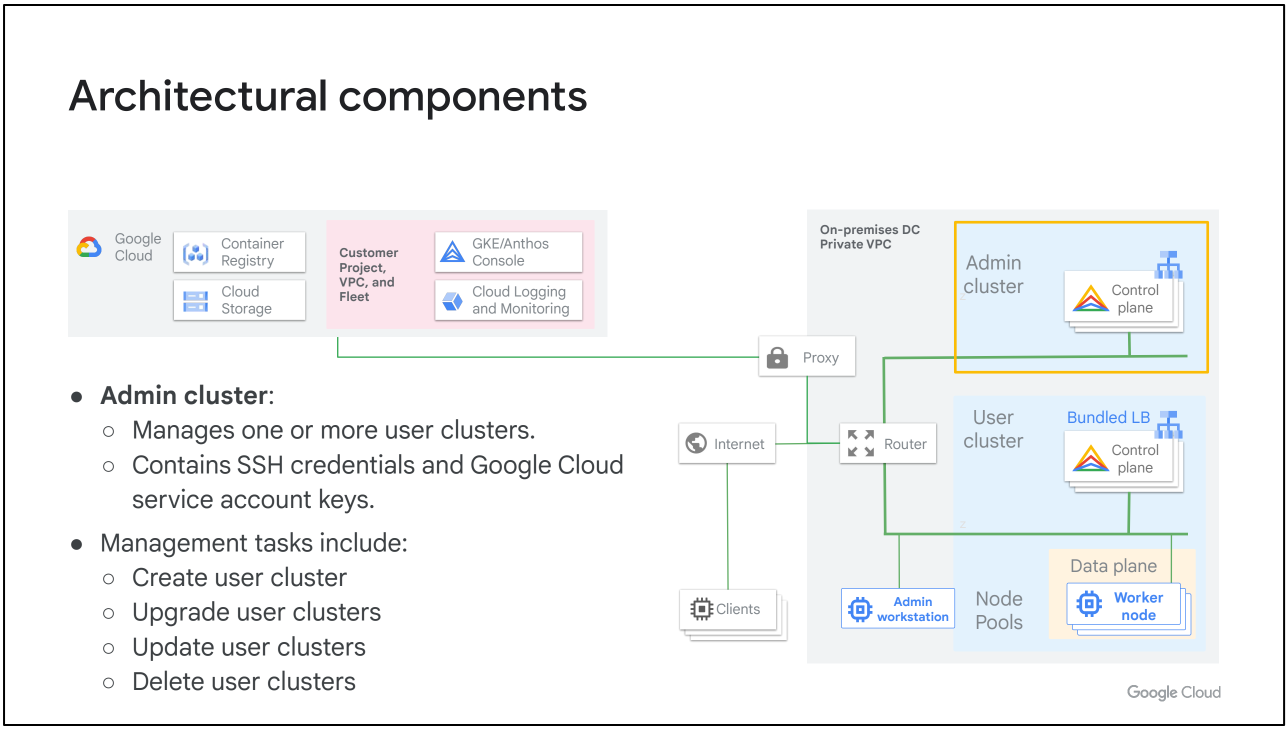
Second, all installation assets for Anthos on bare metal are shared via Container Registry and Cloud Storage.



Architectural components

- **Admin workstation**: designated machine in your network to configure, install, and manage Anthos clusters on bare metal.

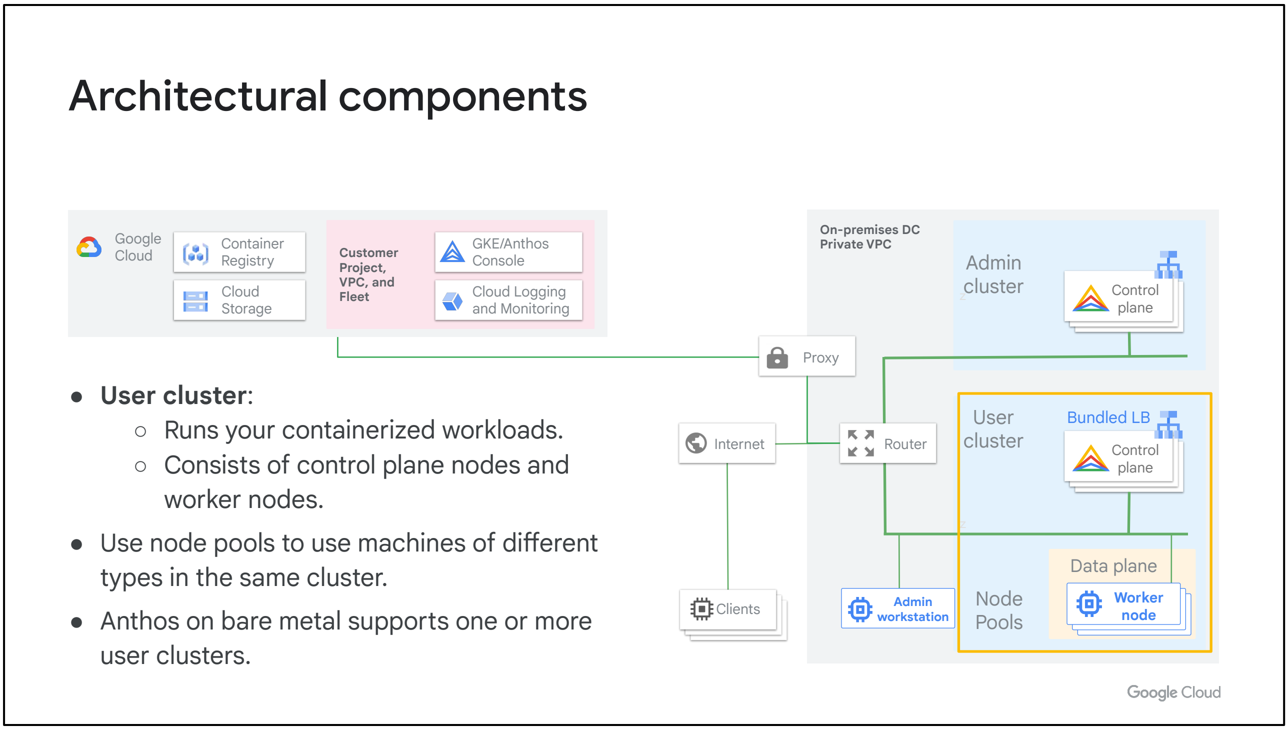
Third, operators create a special-purpose admin workstation which has all the tools necessary to install and manage your bare metal clusters. This workstation must have network access to all the machines that will be in admin and user clusters.



Architectural components

- **Admin cluster**:
  - Manages one or more user clusters.
  - Contains SSH credentials and Google Cloud service account keys.
- Management tasks include:
  - Create user cluster
  - Upgrade user clusters
  - Update user clusters
  - Delete user clusters

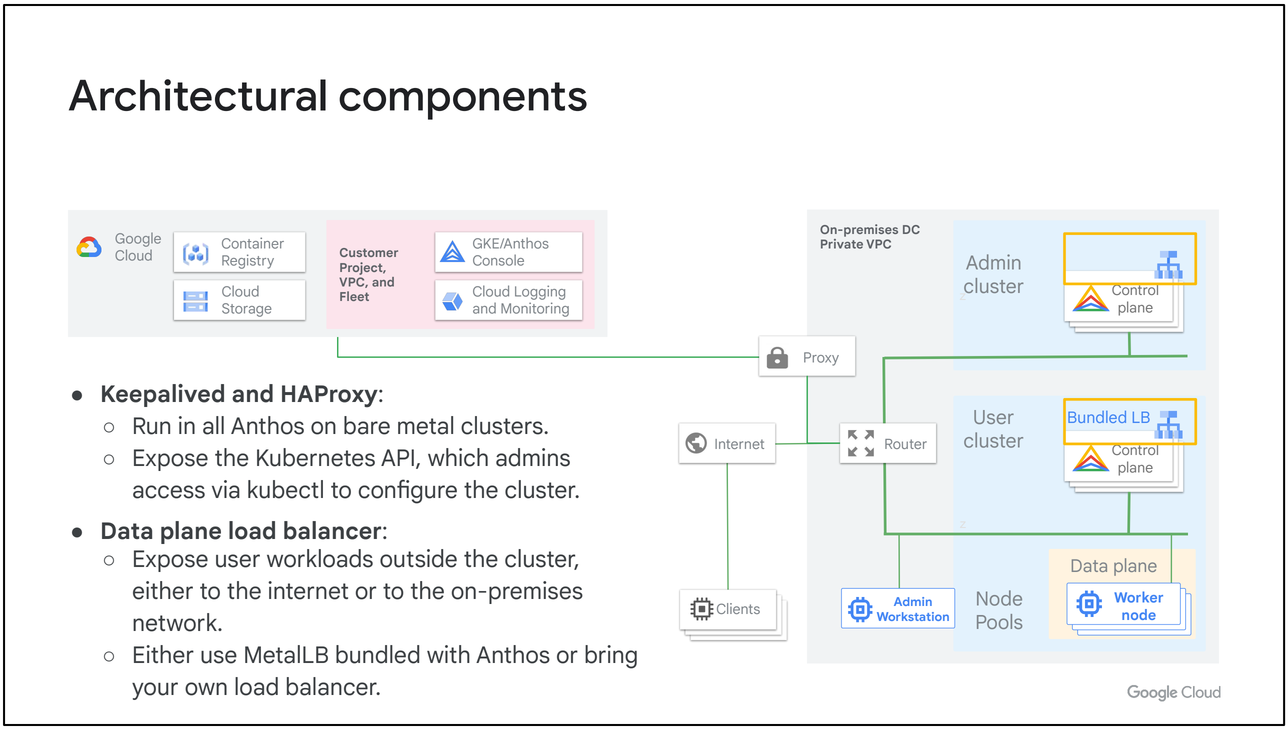
Fourth, operators create an admin cluster. This cluster is running software used to install and manage user clusters—the clusters to which you deploy application workloads. We discuss the software on this cluster in detail in a few minutes. Typically, admin clusters are comprised only of control plane nodes.



Architectural components

- **User cluster**:
  - Runs your containerized workloads.
  - Consists of control plane nodes and worker nodes.
- Use node pools to use machines of different types in the same cluster.
- Anthos on bare metal supports one or more user clusters.

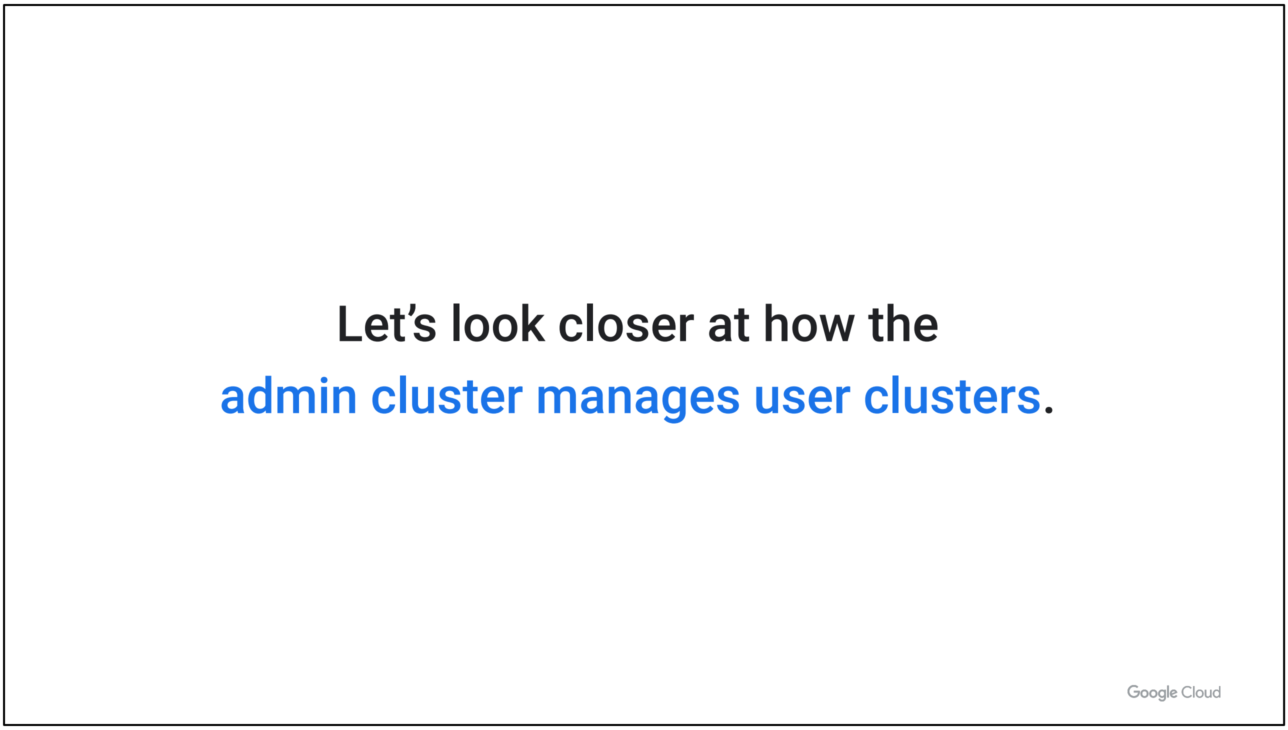
Fifth, with the admin cluster in place, operators can create user clusters, which run your containerized workloads. User clusters are comprised of control plane and worker nodes. You can build clusters using multiple machine types, with machines of the same type grouped into node pools.



Architectural components

- **Keepalived and HAProxy**:
  - Run in all Anthos on bare metal clusters.
  - Expose the Kubernetes API, which admins access via kubectl to configure the cluster.
- **Data plane load balancer**:
  - Expose user workloads outside the cluster, either to the internet or to the on-premises network.
  - Either use MetalLB bundled with Anthos or bring your own load balancer.

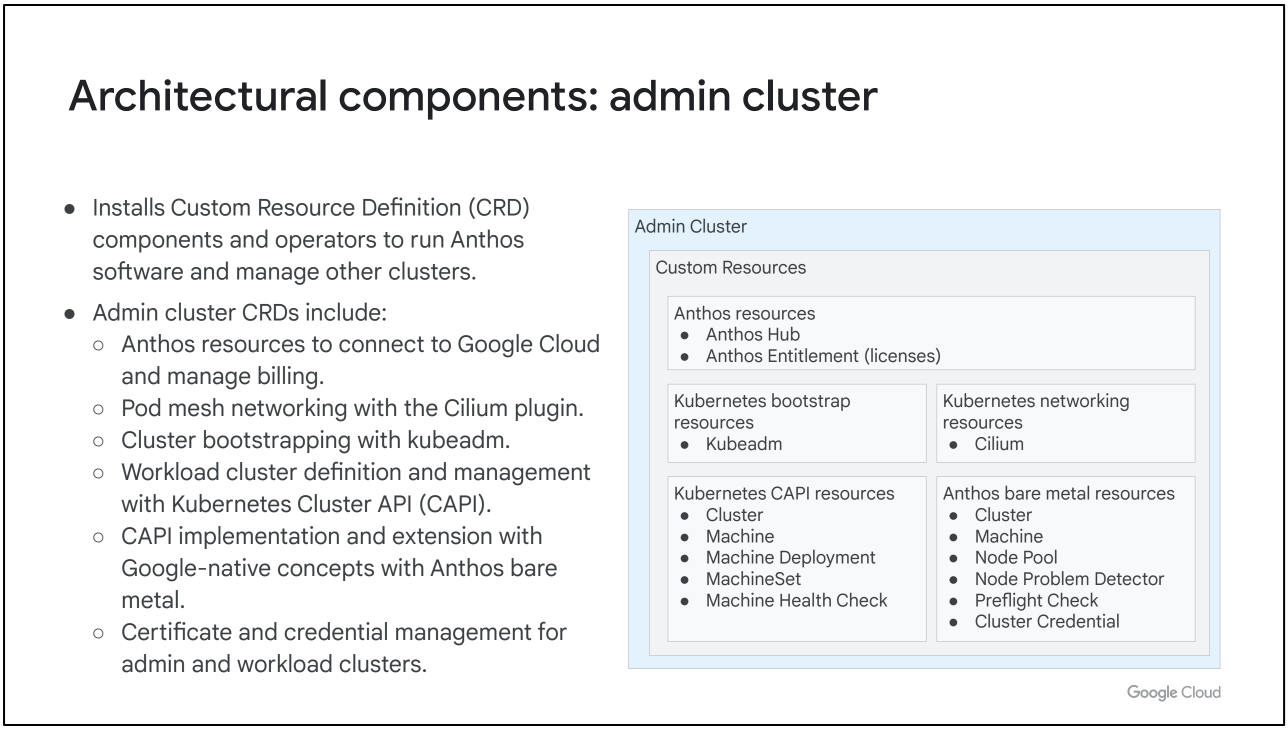
Lastly, you need load balancers in place for each cluster.
Keepalived and HAProxy: Run in all Anthos on bare metal clusters and expose the Kubernetes API, which admins access via kubectl to configure the cluster.
The Data plane load balancer is used to expose user workloads outside the cluster, either to the internet or to the on-premises network. You can either use MetalLB bundled with Anthos or bring your own load balancer.



Let’s look closer at how the
admin cluster manages user clusters.

Google Cloud

OK, why exactly do we have an admin cluster—how does it facilitate creating and managing user clusters?



Architectural components: admin cluster

- Installs Custom Resource Definition (CRD) components and operators to run Anthos software and manage other clusters.
- Admin cluster CRDs include:
  - Anthos resources to connect to Google Cloud and manage billing.
  - Pod mesh networking with the Cilium plugin.
  - Cluster bootstrapping with kubeadm.
  - Workload cluster definition and management with Kubernetes Cluster API (CAPI).
  - CAPI implementation and extension with Google-native concepts with Anthos bare metal.
  - Certificate and credential management for admin and workload clusters.

Admin Cluster

Custom Resources

Anthos resources
- Anthos Hub
- Anthos Entitlement (licenses)

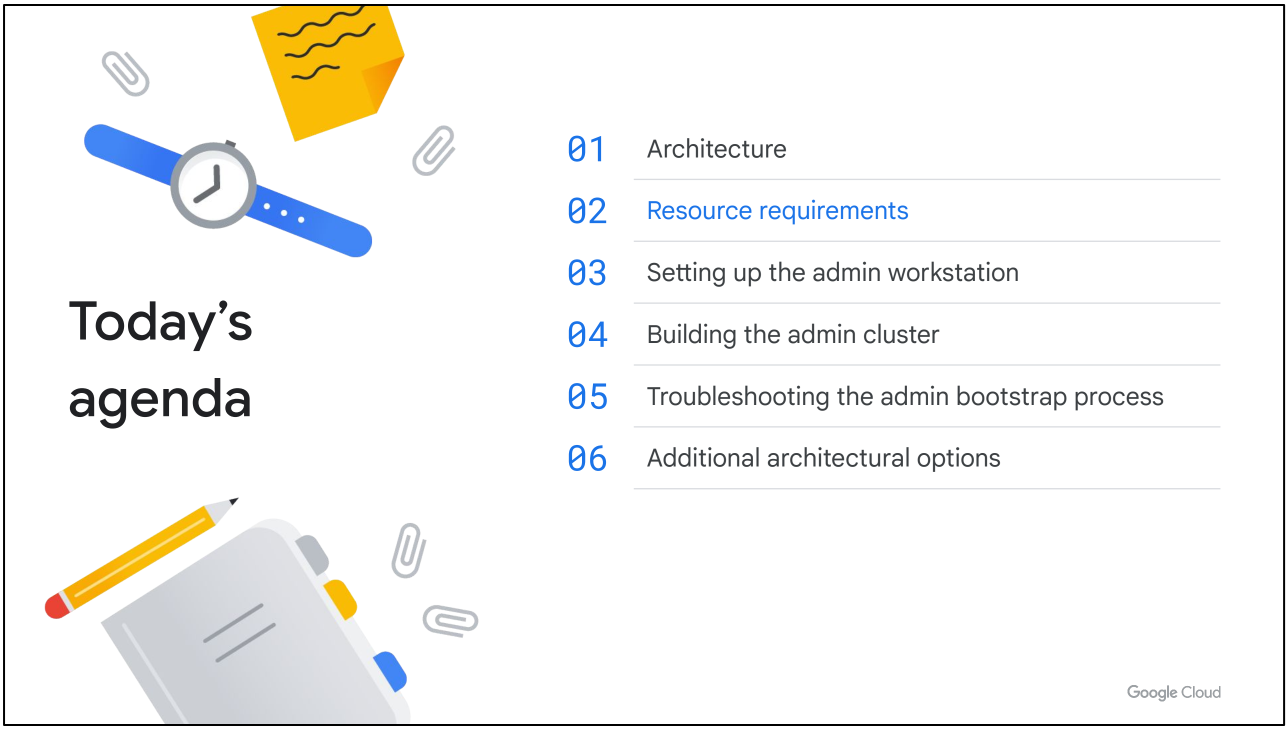
Kubernetes bootstrap resources
- Kubeadm

Kubernetes networking resources
- Cilium

Kubernetes CAPI resources
- Cluster
- Machine
- Machine Deployment
- MachineSet
- Machine Health Check

Anthos bare metal resources
- Cluster
- Machine
- Node Pool
- Node Problem Detector
- Preflight Check
- Cluster Credential

Well, I'm glad you asked! First, the admin cluster has a bunch of Custom Resource Definitions that describe Anthos cluster components. These include general Anthos resources, Cilium networking resources, cluster API resources, Anthos bare metal resources, etc. In addition, there are custom operators installed on the admin cluster, software which responds to changes in the custom resources and actually reaches out to configure and modify your cluster servers as needed so they form a functioning cluster.



Today's agenda

01 Architecture

02 Resource requirements

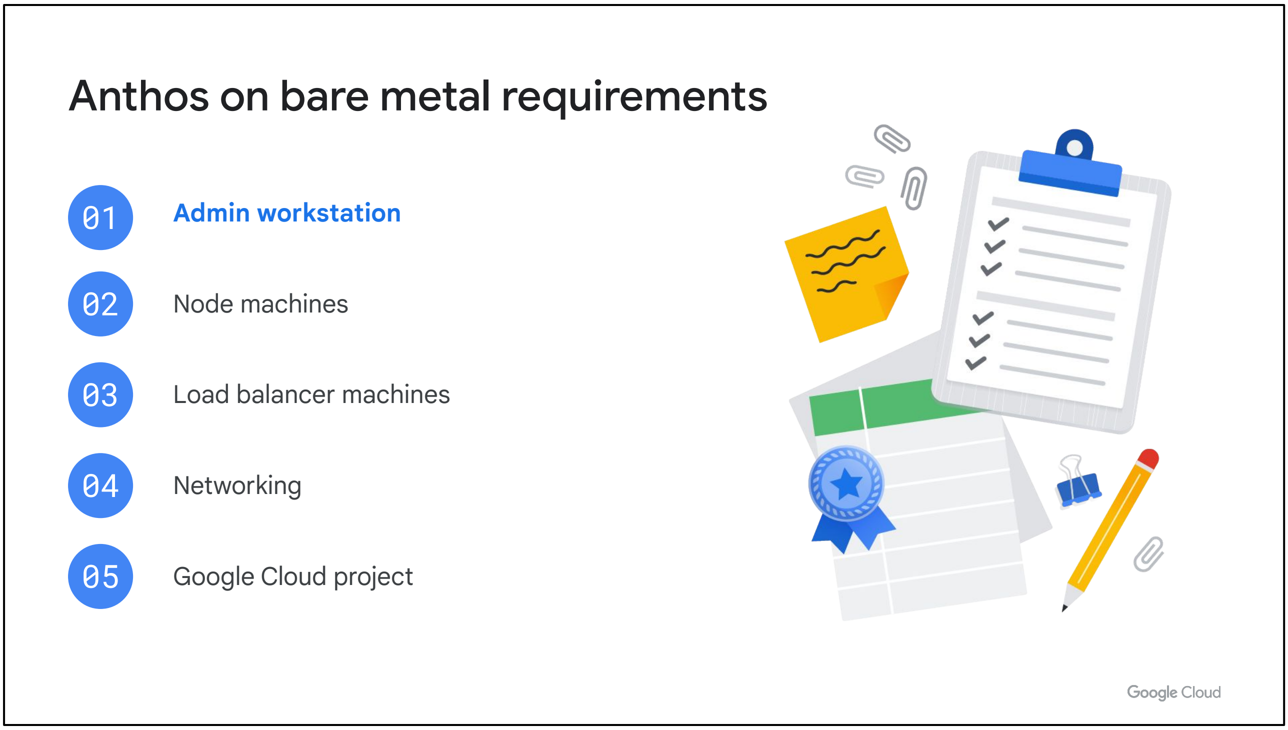
03 Setting up the admin workstation

04 Building the admin cluster

05 Troubleshooting the admin bootstrap process

06 Additional architectural options

You may be wondering about the requirements for each of these components—let's take a moment to explore what's needed.



Anthos on bare metal requirements

01 **Admin workstation**

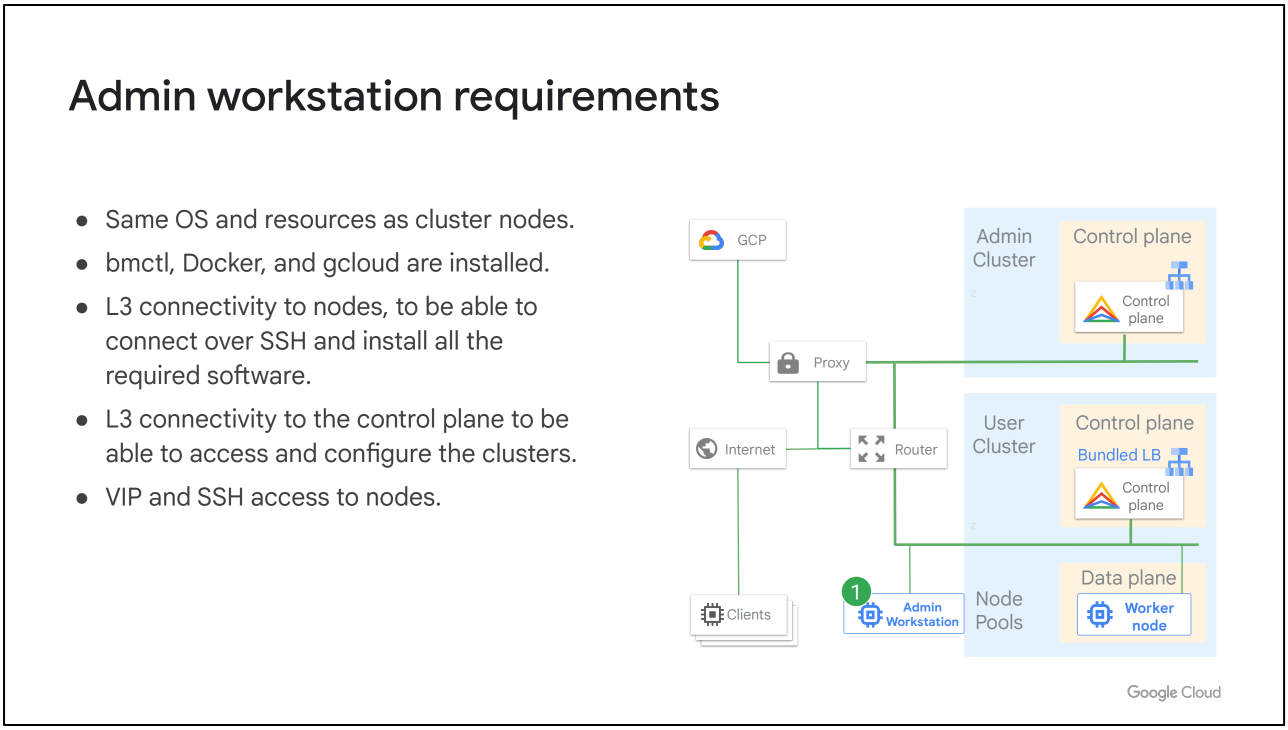
02 Node machines

03 Load balancer machines

04 Networking

05 Google Cloud project

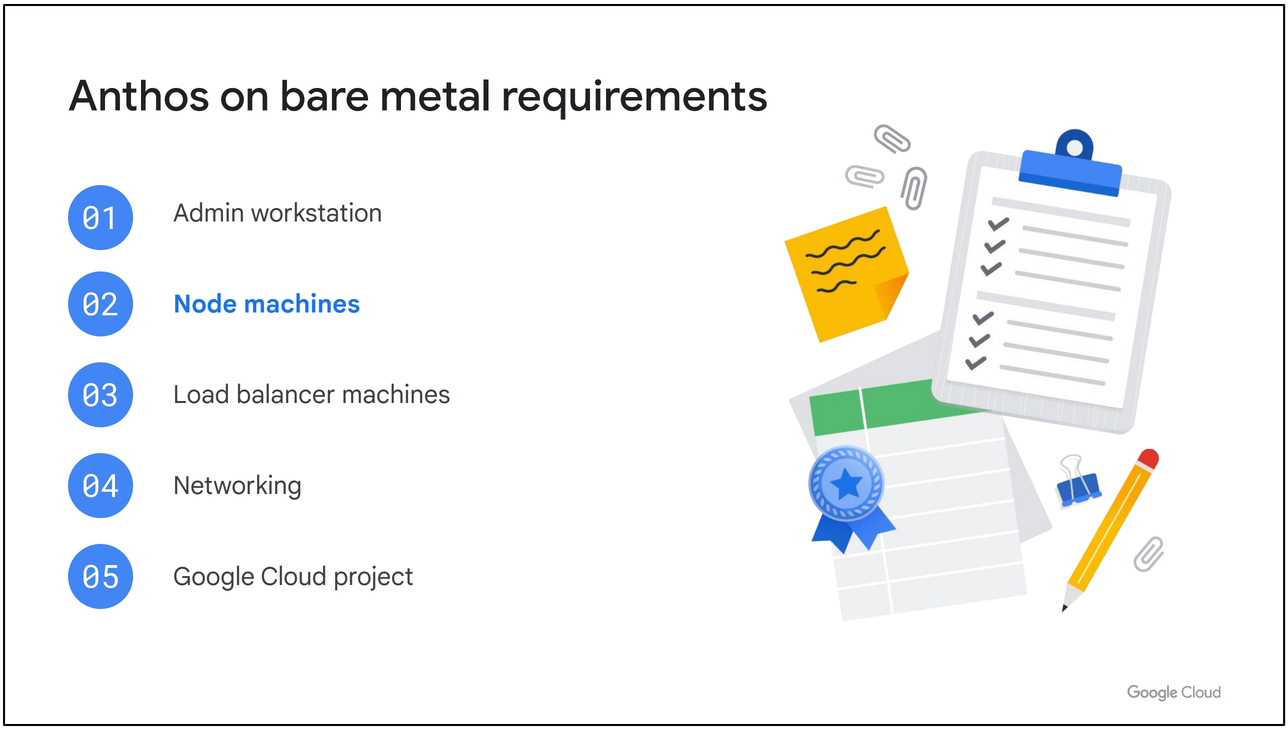
First, for the admin workstation ...



Admin workstation requirements

- Same OS and resources as cluster nodes.
- bmctl, Docker, and gcloud are installed.
- L3 connectivity to nodes, to be able to connect over SSH and install all the required software.
- L3 connectivity to the control plane to be able to access and configure the clusters.
- VIP and SSH access to nodes.

The server must be running the same operating system as the cluster nodes are running. The workstation must have network connectivity to the cluster and load balancer nodes, and must be able to SSH into those nodes. And you must manually install some tools on the workstation, including the bmctl administrative command-line utility.



Anthos on bare metal requirements

01 Admin workstation

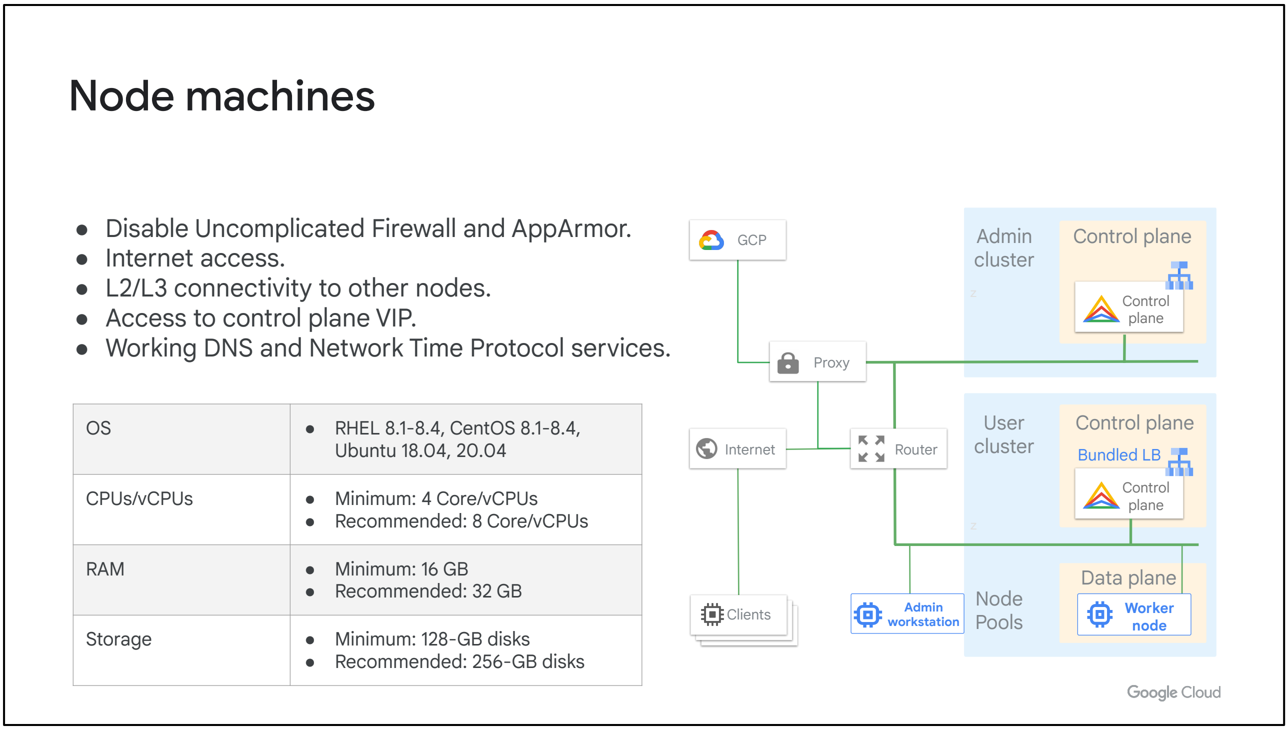
02 **Node machines**

03 Load balancer machines

04 Networking

05 Google Cloud project

For the cluster nodes ...



Node machines

- Disable Uncomplicated Firewall and AppArmor.
- Internet access.
- L2/L3 connectivity to other nodes.
- Access to control plane VIP.
- Working DNS and Network Time Protocol services.

OS	● RHEL 8.1-8.4, CentOS 8.1-8.4, Ubuntu 18.04, 20.04
CPUs/vCPUs	● Minimum: 4 Core/vCPUs ● Recommended: 8 Core/vCPUs
RAM	● Minimum: 16 GB ● Recommended: 32 GB
Storage	● Minimum: 128-GB disks ● Recommended: 256-GB disks

They must be running one of these supported operating systems.

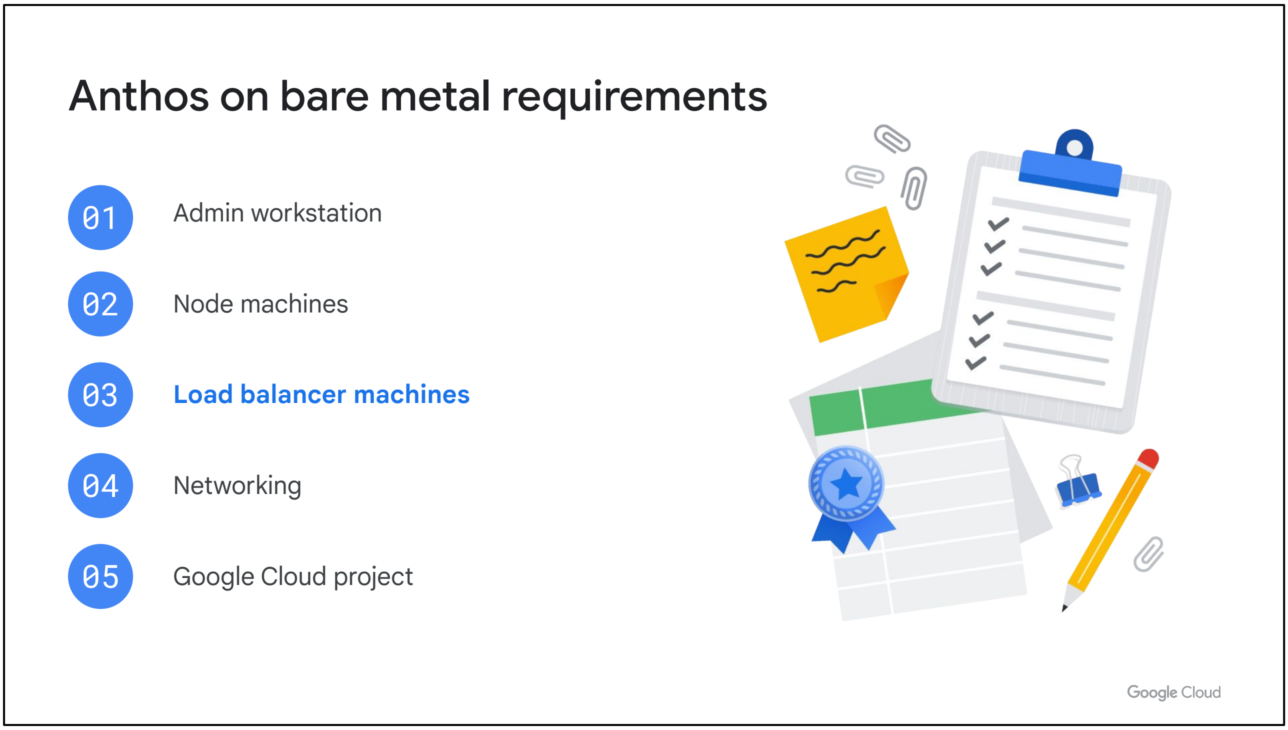
Each machine should have a minimum of eight virtual CPUs.

And it's recommended that the machines have 32GB of RAM.

Google recommends 256GB disks.

For operating systems that use Uncomplicated Firewall and AppArmor, you will need to disable those services.

You will want to confirm the configuration of DNS, NTP, and L3 connectivity to the VIP you are setting aside for the admin cluster control plane.



Anthos on bare metal requirements

01 Admin workstation

02 Node machines

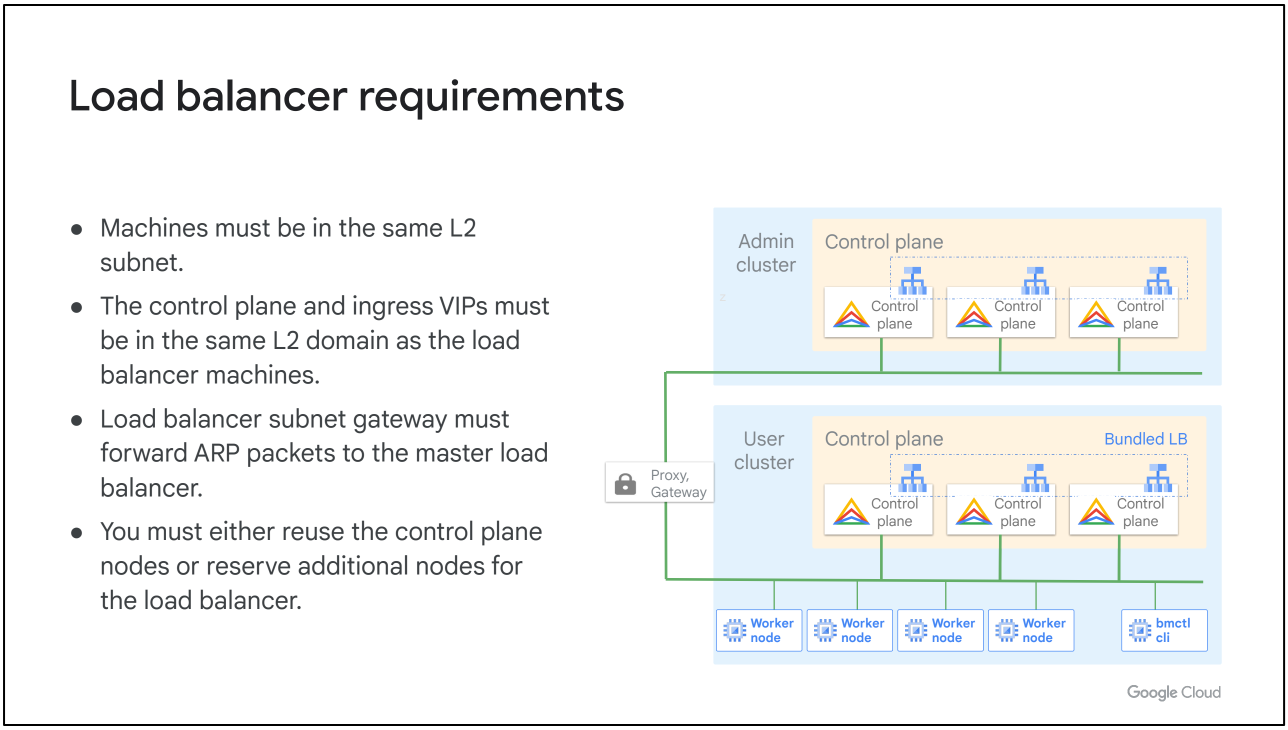
03 **Load balancer machines**

04 Networking

05 Google Cloud project

Google Cloud

For load balancer machines ...



Load balancer requirements

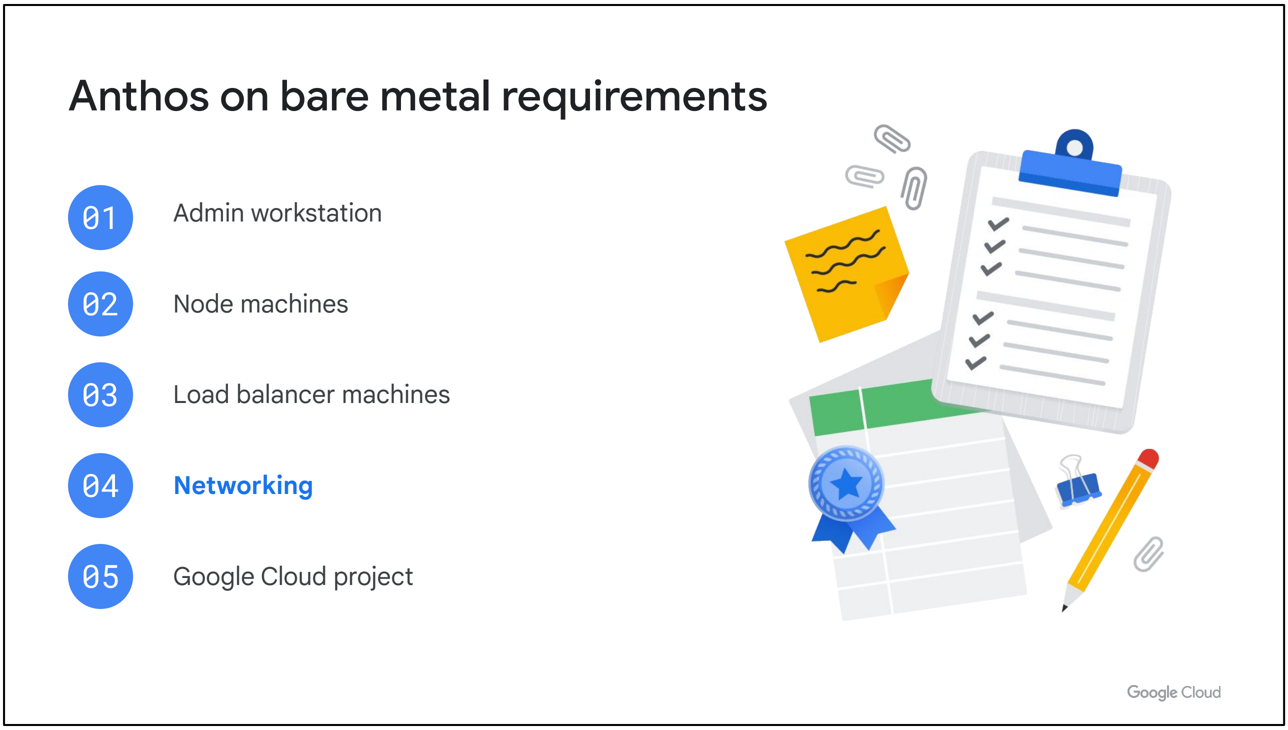
- Machines must be in the same L2 subnet.
- The control plane and ingress VIPs must be in the same L2 domain as the load balancer machines.
- Load balancer subnet gateway must forward ARP packets to the master load balancer.
- You must either reuse the control plane nodes or reserve additional nodes for the load balancer.

Machines running the load balancing software must be in the same L2 domain.

The control plane and ingress VIPs must be in the same L2 domain as the load balancer machines.

The gateway to the subnet used by the load balancer machines must listen to and forward gratuitous ARP messages.

You install the load balancing software either on cluster control plane nodes, or on nodes in a separate, load balancing node pool.



Anthos on bare metal requirements

01 Admin workstation

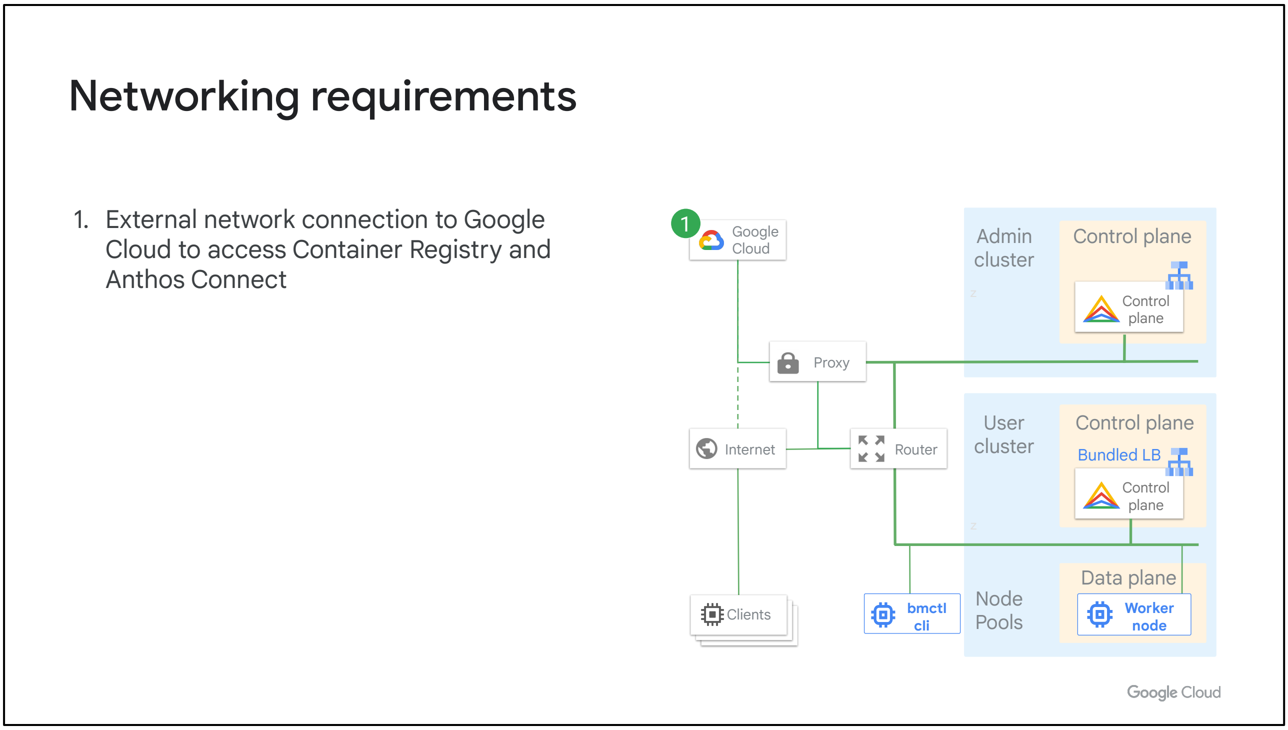
02 Node machines

03 Load balancer machines

04 **Networking**

05 Google Cloud project

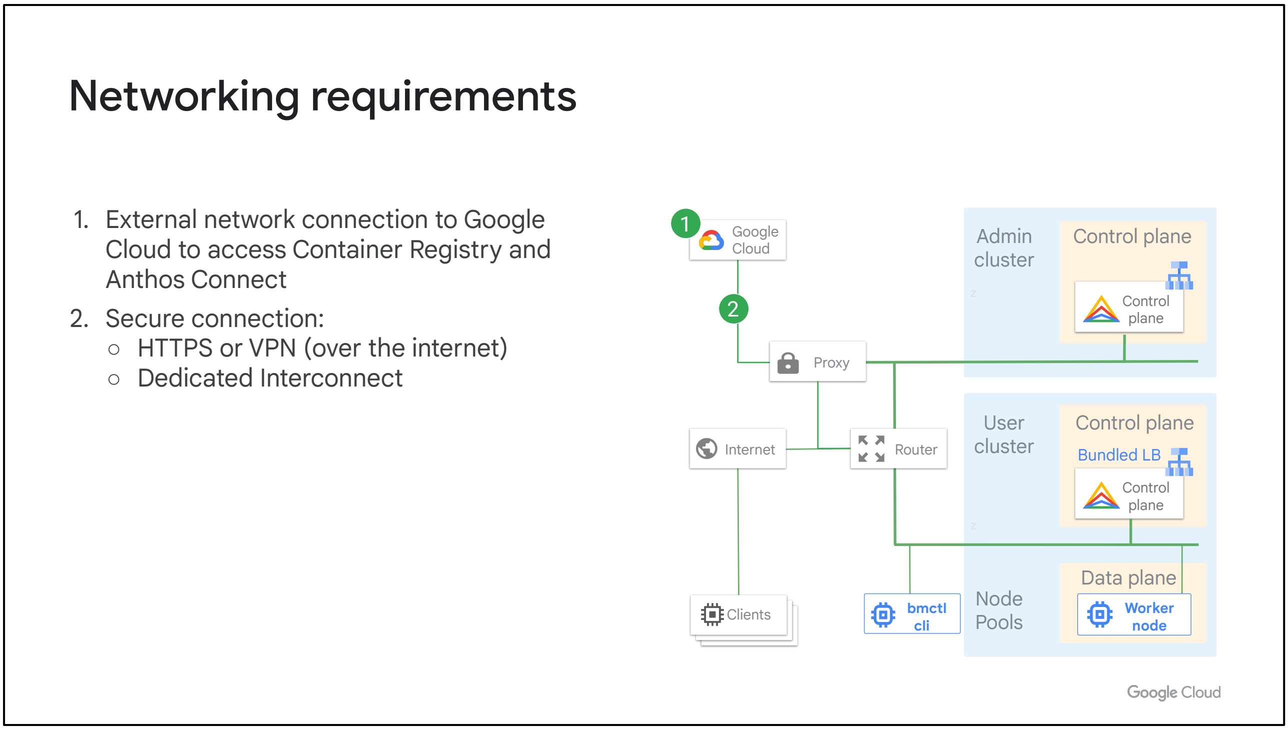
Beyond the load balancers, you will need to make the following configuration changes on your networks.



Networking requirements

1. External network connection to Google Cloud to access Container Registry and Anthos Connect

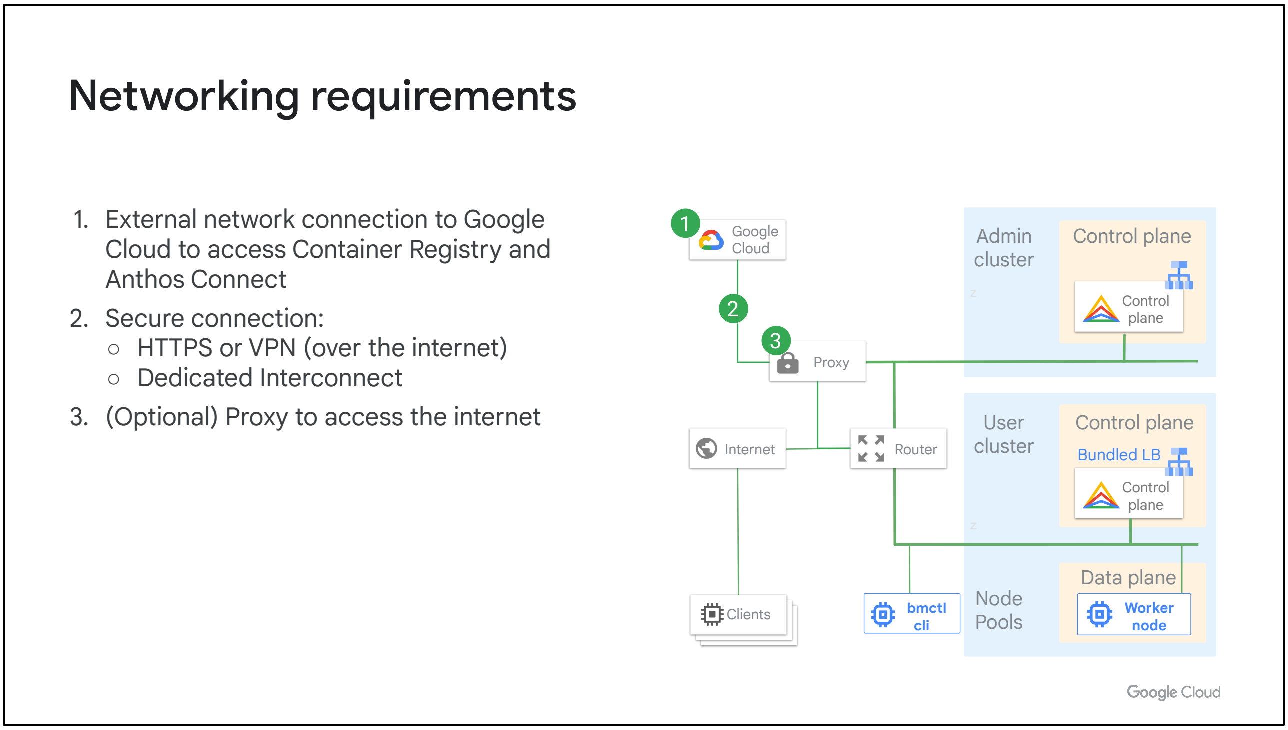
Your clusters will need to talk to several Google Cloud services, including Anthos Connect.



Networking requirements

1. External network connection to Google Cloud to access Container Registry and Anthos Connect
2. Secure connection:
   - HTTPS or VPN (over the internet)
   - Dedicated Interconnect

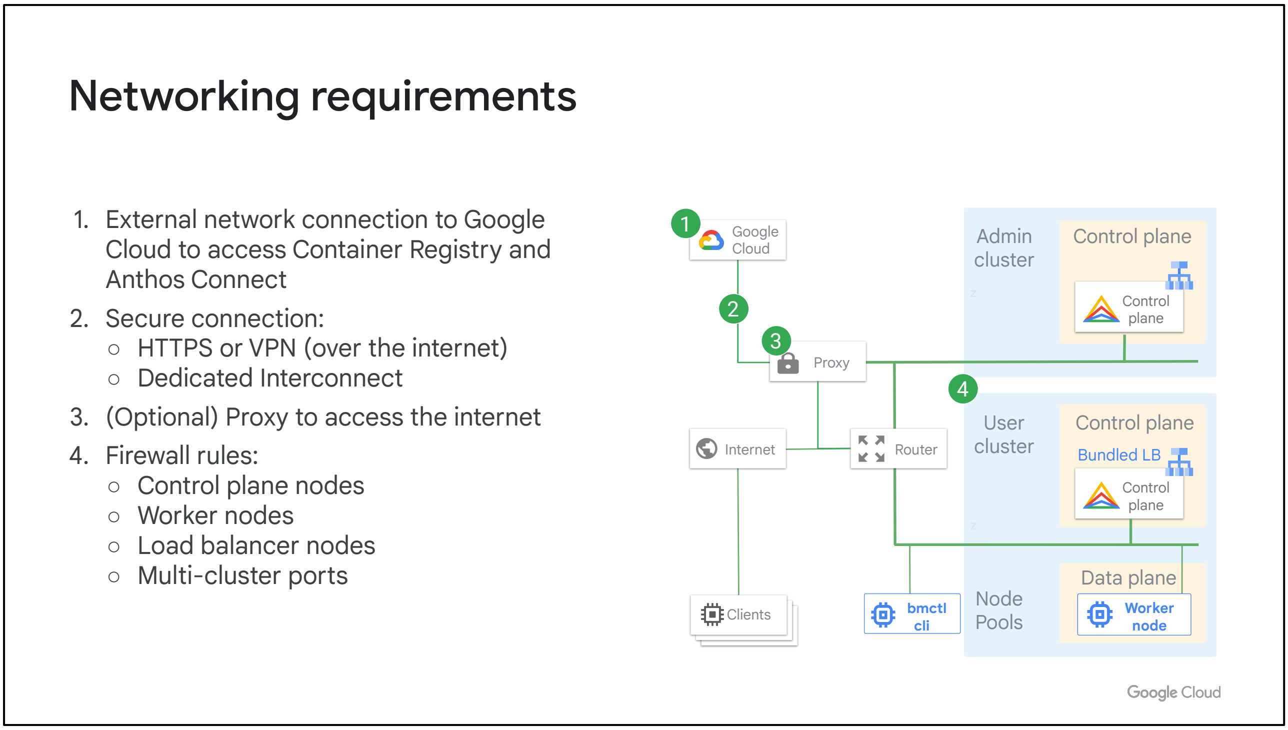
They may connect using a TLS channel over the open internet, or may use a VPN or Dedicated Interconnect connection.



Networking requirements

1. External network connection to Google Cloud to access Container Registry and Anthos Connect
2. Secure connection:
   - ○ HTTPS or VPN (over the internet)
   - ○ Dedicated Interconnect
3. (Optional) Proxy to access the internet

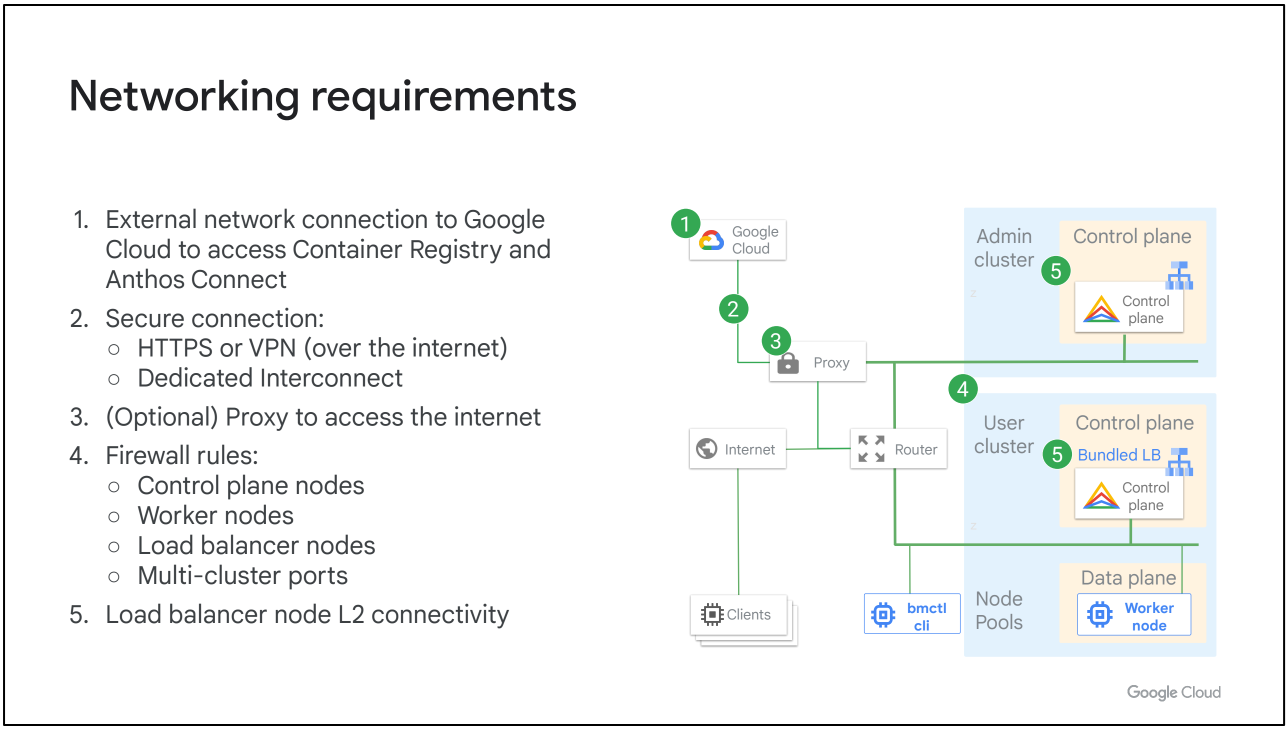
If your on-premises machines go through a proxy to reach the internet, you will need to configure your nodes to use the proxy successfully.



Networking requirements

1. External network connection to Google Cloud to access Container Registry and Anthos Connect
2. Secure connection:
   - HTTPS or VPN (over the internet)
   - Dedicated Interconnect
3. (Optional) Proxy to access the internet
4. Firewall rules:
   - Control plane nodes
   - Worker nodes
   - Load balancer nodes
   - Multi-cluster ports

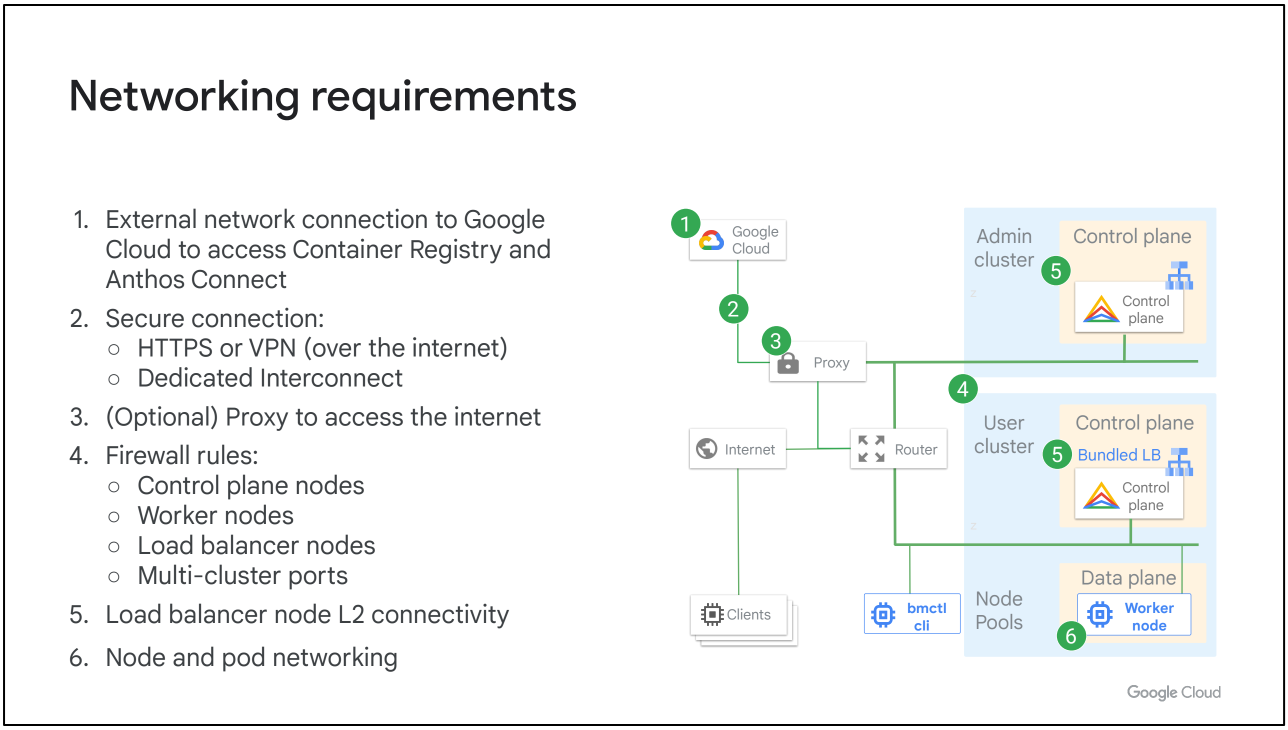
Obviously, you'll need to configure firewall rules to allow traffic to flow to machines as needed. Different rules will control traffic to control plane nodes, worker nodes, and load balancer nodes. Most rules are required for basic bare metal cluster operation, though a few are specific to multi-cluster deployments.



Networking requirements

1. External network connection to Google Cloud to access Container Registry and Anthos Connect
2. Secure connection:
   - HTTPS or VPN (over the internet)
   - Dedicated Interconnect
3. (Optional) Proxy to access the internet
4. Firewall rules:
   - Control plane nodes
   - Worker nodes
   - Load balancer nodes
   - Multi-cluster ports
5. Load balancer node L2 connectivity

As noted earlier, the nodes running your load balancing software require L2 connectivity amongst themselves.

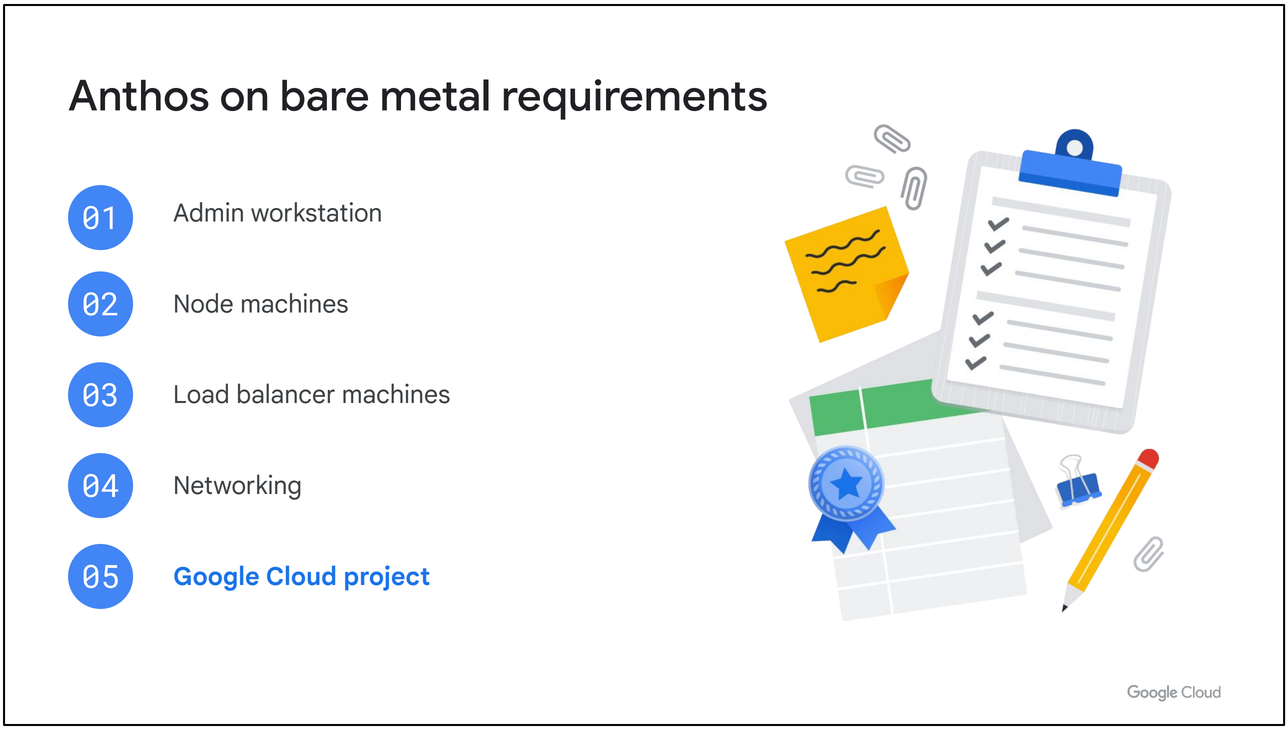


Networking requirements

1. External network connection to Google Cloud to access Container Registry and Anthos Connect
2. Secure connection:
   - HTTPS or VPN (over the internet)
   - Dedicated Interconnect
3. (Optional) Proxy to access the internet
4. Firewall rules:
   - Control plane nodes
   - Worker nodes
   - Load balancer nodes
   - Multi-cluster ports
5. Load balancer node L2 connectivity
6. Node and pod networking

Google Cloud

And lastly, you'll need to make sure you have IP addresses and L3 connectivity between nodes to support node and pod networking.



Anthos on bare metal requirements

01 Admin workstation

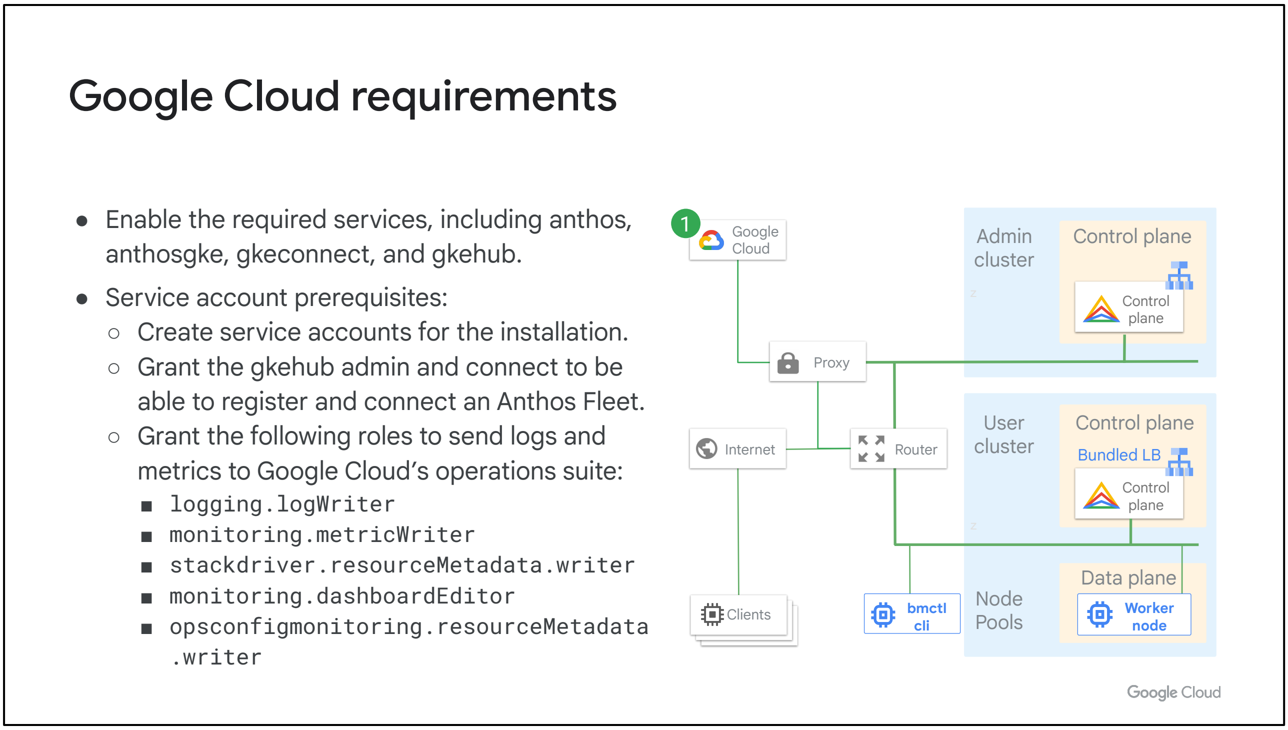
02 Node machines

03 Load balancer machines

04 Networking

05 **Google Cloud project**

In addition to all the on-premises resources, you will need to set up a Google Cloud project.

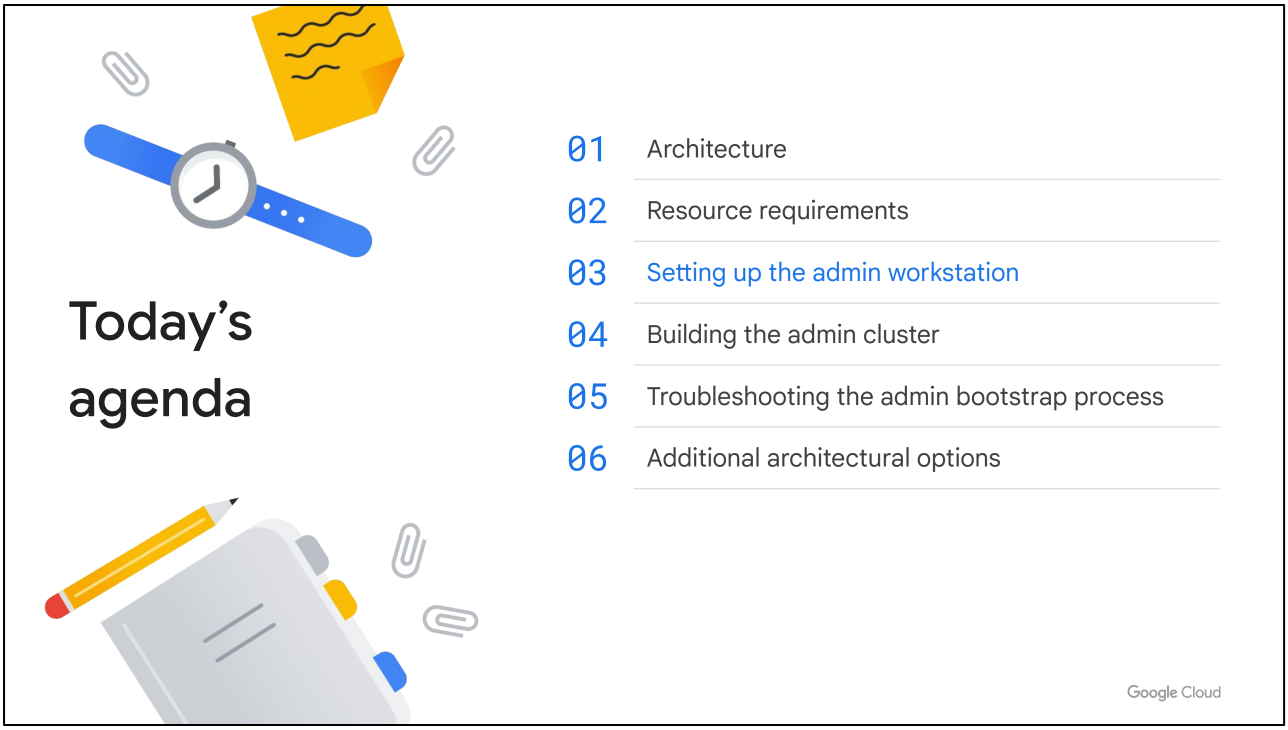


Google Cloud requirements

- Enable the required services, including anthos, anthosgke, gkeconnect, and gkehub.
- Service account prerequisites:
  - Create service accounts for the installation.
  - Grant the gkehub admin and connect to be able to register and connect an Anthos Fleet.
  - Grant the following roles to send logs and metrics to Google Cloud's operations suite:
    - `logging.logWriter`
    - `monitoring.metricWriter`
    - `stackdriver.resourceMetadata.writer`
    - `monitoring.dashboardEditor`
    - `opsconfigmonitoring.resourceMetadata.writer`

Google Cloud

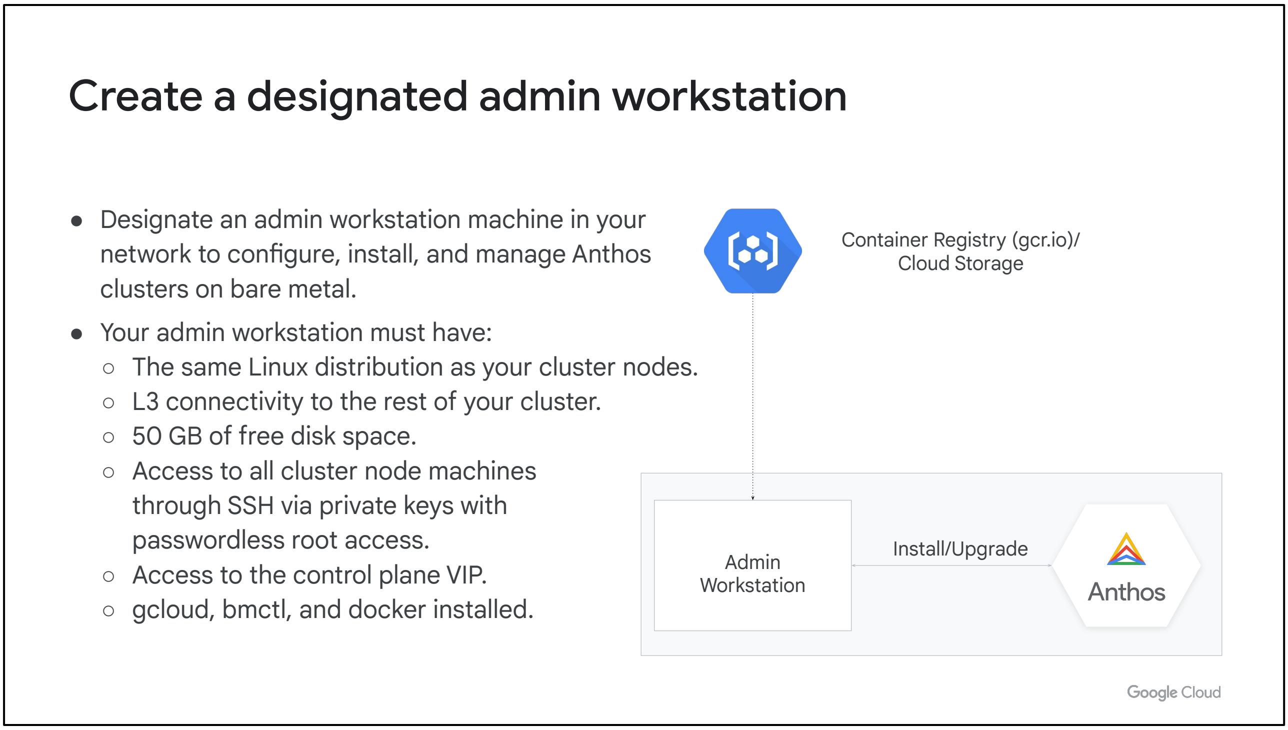
This entails enabling the required services …



Today's agenda

- 01 Architecture
- 02 Resource requirements
- 03 Setting up the admin workstation
- 04 Building the admin cluster
- 05 Troubleshooting the admin bootstrap process
- 06 Additional architectural options

Let's discuss setting up the admin workstation.



Create a designated admin workstation

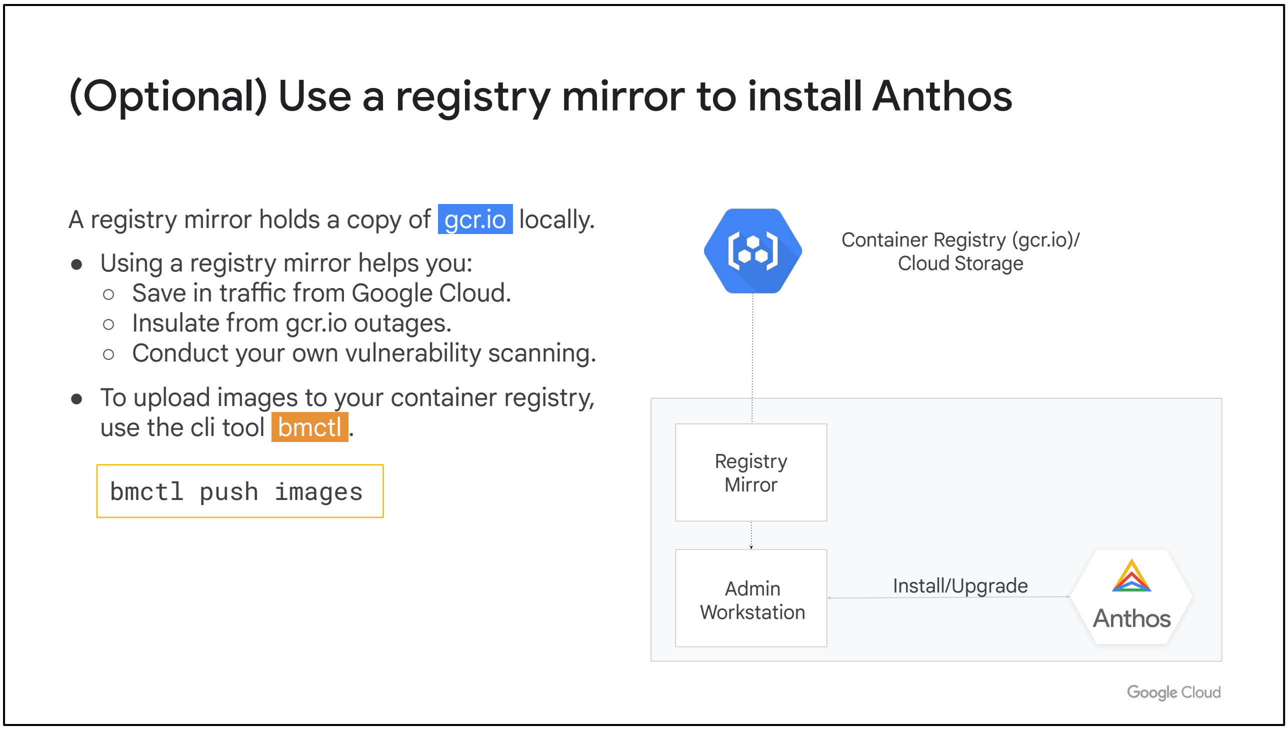
- Designate an admin workstation machine in your network to configure, install, and manage Anthos clusters on bare metal.
- Your admin workstation must have:
  - The same Linux distribution as your cluster nodes.
  - L3 connectivity to the rest of your cluster.
  - 50 GB of free disk space.
  - Access to all cluster node machines through SSH via private keys with passwordless root access.
  - Access to the control plane VIP.
  - gcloud, bmctl, and docker installed.

Select a computer on the network where you plan to deploy your clusters.

Make sure your workstation is running the appropriate operating system, has network connectivity to the machines you will be using in your cluster, and has at least 50 GB of free disk space.

You will need to install the Google Cloud SDK, the bmctl command, and Docker.

You will also need to configure the workstation with SSH private keys that allow software running on the workstation to connect to remote servers as root and perform installations and configuration.



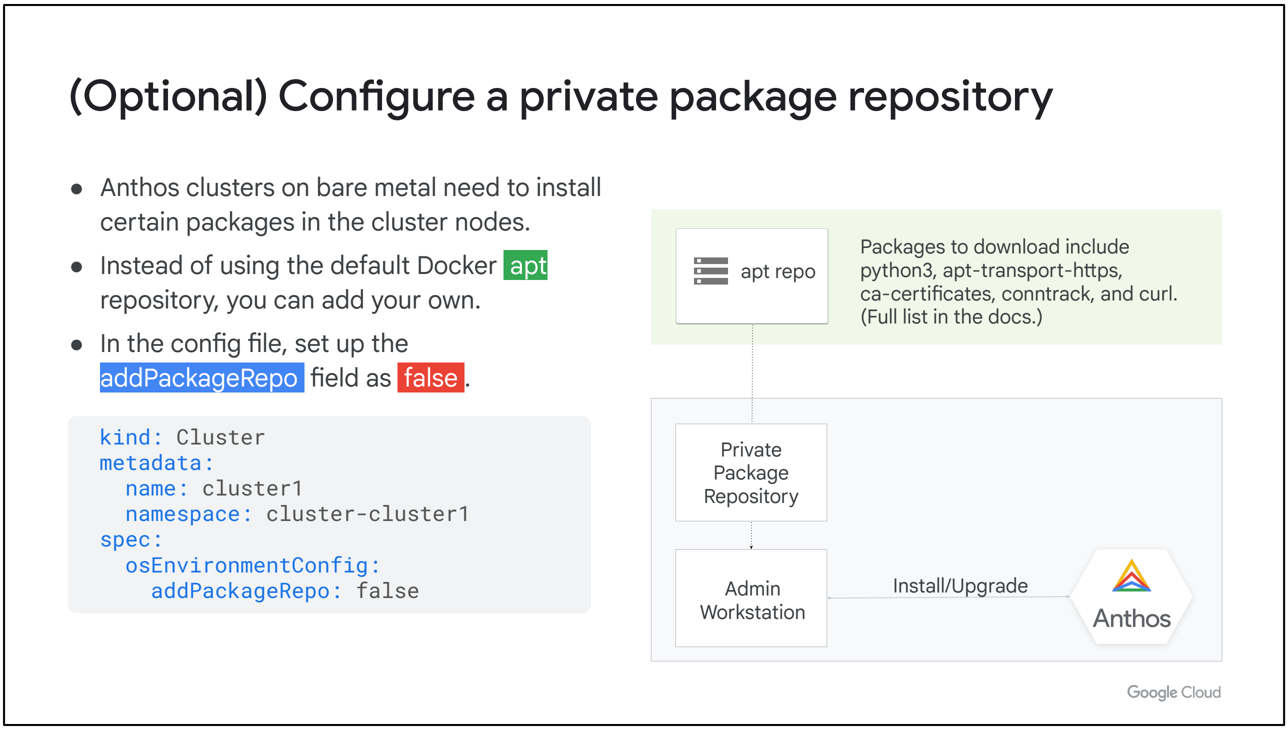
(Optional) Use a registry mirror to install Anthos

A registry mirror holds a copy of gcr.io locally.

- Using a registry mirror helps you:
  - Save in traffic from Google Cloud.
  - Insulate from gcr.io outages.
  - Conduct your own vulnerability scanning.
- To upload images to your container registry, use the cli tool bmctl.

```
bmctl push images
```

Some organizations will want to use a registry mirror, installing from a local registry rather than pulling images from gcr.io. Using a mirror can reduce internet traffic, lower latencies, and improve your security posture. You can download the Anthos images from gcr.io, and use the bmctl utility to load them into your private registry.



(Optional) Configure a private package repository

- Anthos clusters on bare metal need to install certain packages in the cluster nodes.
- Instead of using the default Docker apt repository, you can add your own.
- In the config file, set up the addPackageRepo field as false.

```
kind: Cluster
metadata:
  name: cluster1
  namespace: cluster-cluster1
spec:
  osEnvironmentConfig:
    addPackageRepo: false
```

apt repo

Packages to download include python3, apt-transport-https, ca-certificates, conntrack, and curl. (Full list in the docs.)

Private Package Repository

Admin Workstation

Install/Upgrade

Anthos

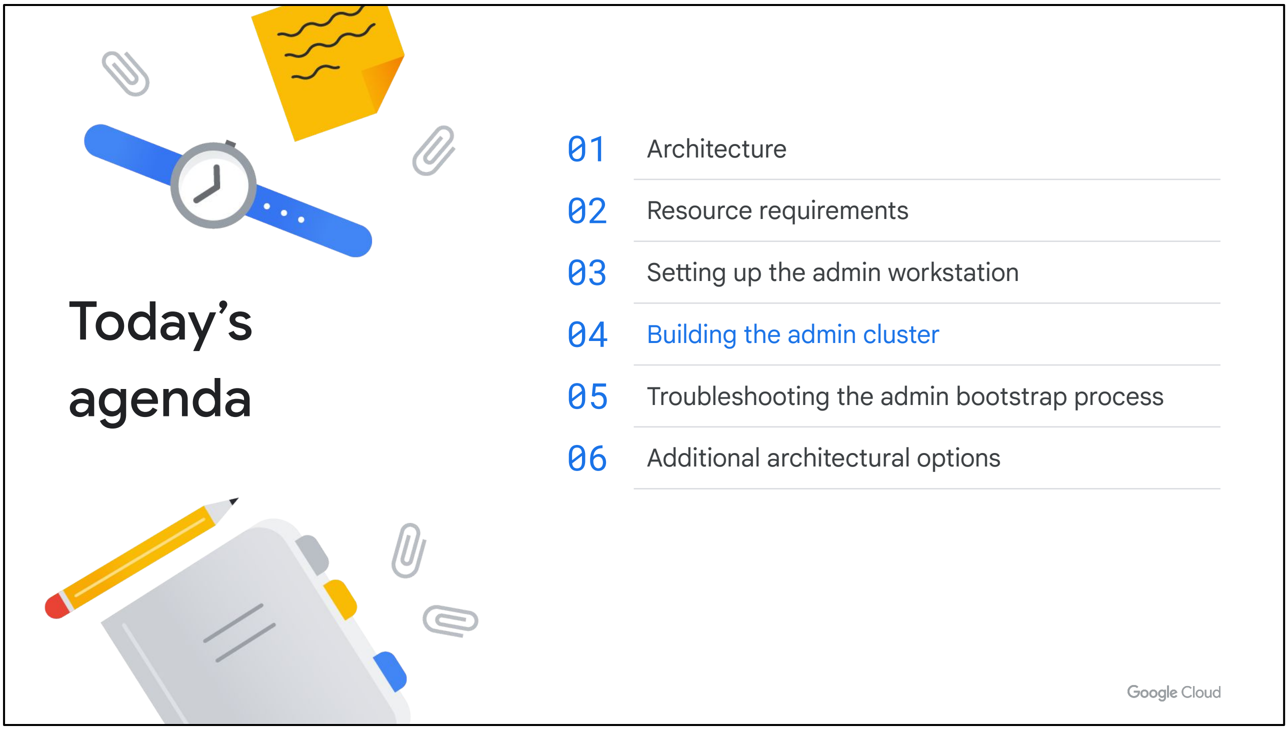
Google Cloud

Similarly, you might want to use a private package registry rather than installing the Docker apt repository on each cluster node.

Your cluster nodes will need to install packages such as those listed in the diagram to the right.

By default, Anthos installs the Docker apt repository on each node; you can change this so that nodes use your own package repository.

All you have to do to enable use of the private repository is modify the addPackageRepo field in the config file.



Today's agenda

01 Architecture

02 Resource requirements

03 Setting up the admin workstation

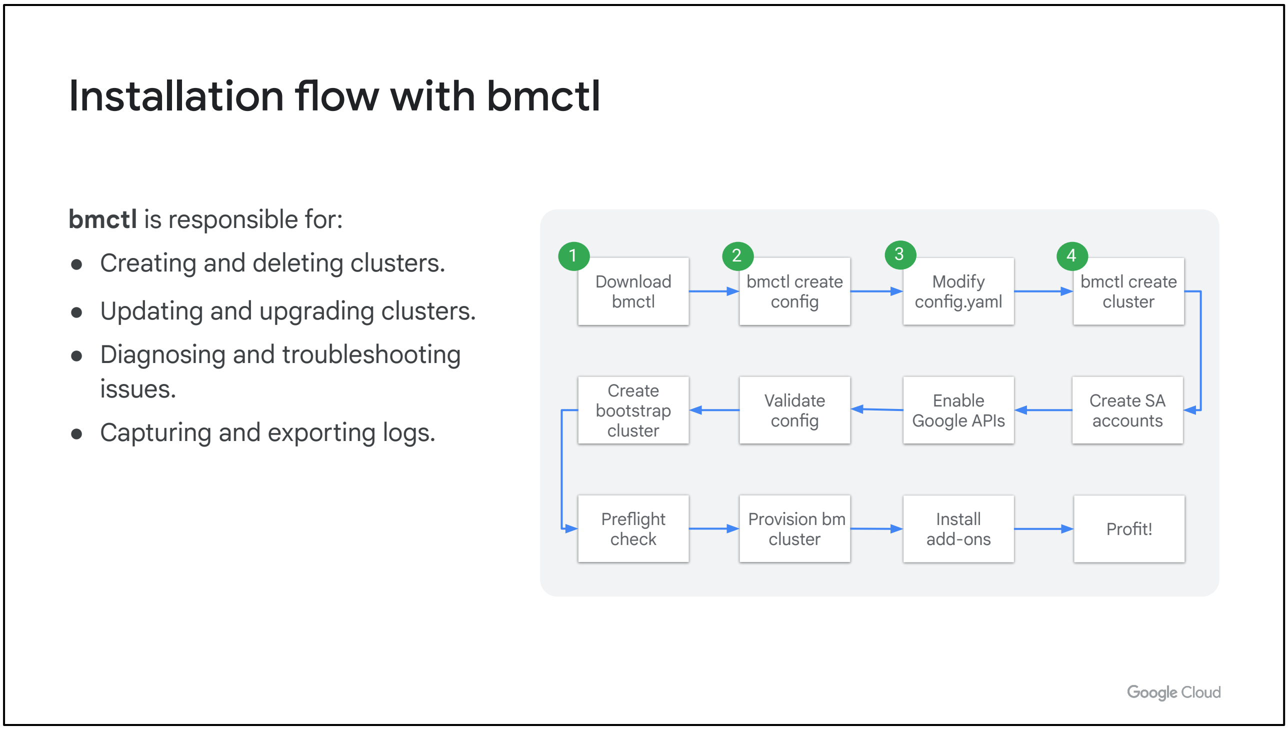
04 Building the admin cluster

05 Troubleshooting the admin bootstrap process

06 Additional architectural options

Google Cloud

Once you have a working admin workstation, you can use it to create an admin cluster.



Installation flow with bmctl

bmctl is responsible for:

- Creating and deleting clusters.
- Updating and upgrading clusters.
- Diagnosing and troubleshooting issues.
- Capturing and exporting logs.

1 Download bmctl

2 bmctl create config

3 Modify config.yaml

4 bmctl create cluster

Create SA accounts

Enable Google APIs

Validate config

Create bootstrap cluster

Preflight check

Provision bm cluster

Install add-ons

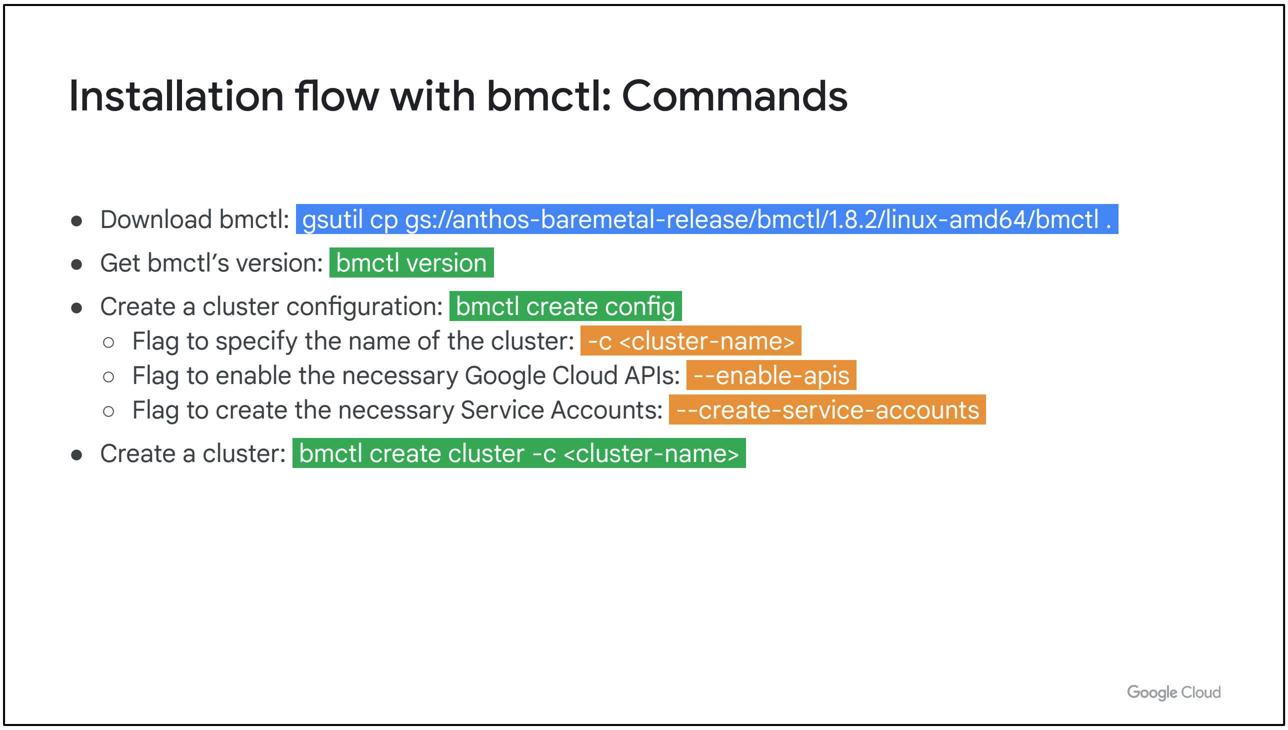
Profit!

Operators use the bmctl tool to create and manage clusters.

The diagram shows how bmctl operates to prepare resources and start the installation of the admin cluster.

The items highlighted with green numbers show the steps performed by the operator. The operator installs bmctl on the admin workstation, runs the create config command to generate a configuration file, modifies the config file as appropriate, then uses bmctl create cluster to build the cluster using the config file.

The bmctl utility then creates necessary service accounts, enables APIs, validates the config file, builds the bootstrap cluster, and uses the bootstrap cluster to create the admin cluster.

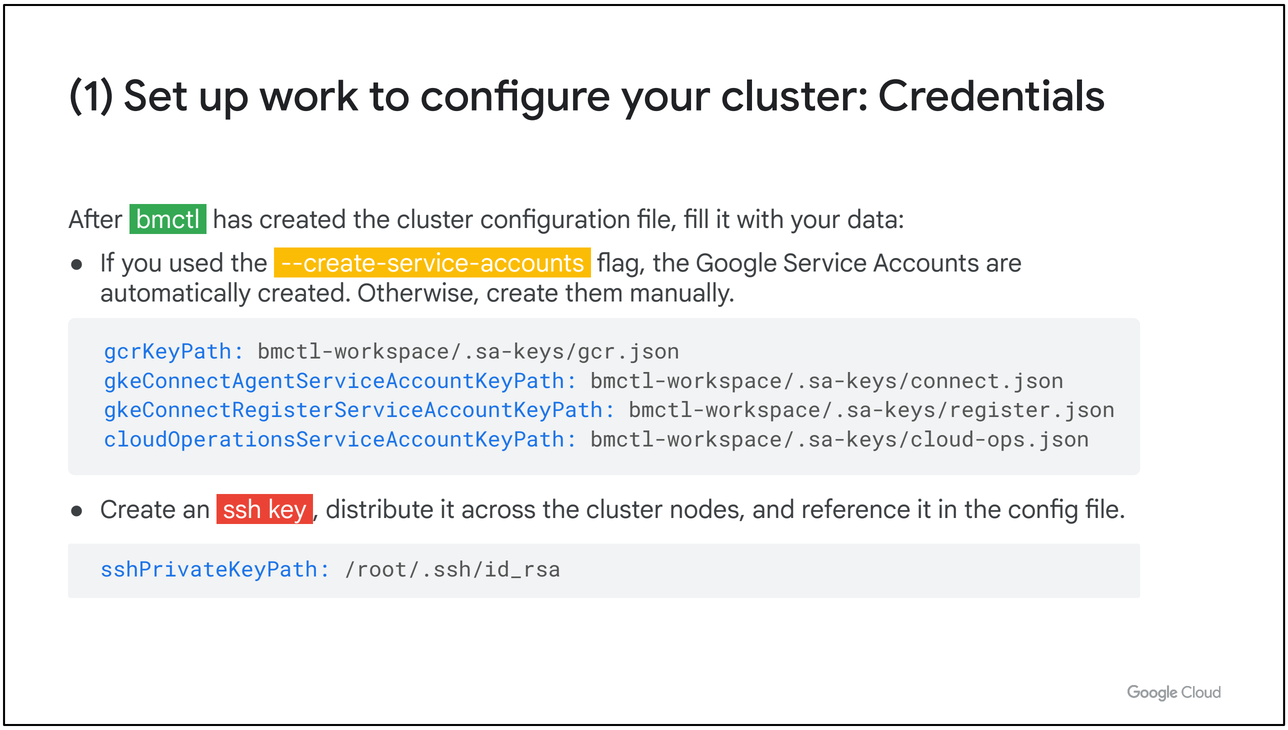


Installation flow with bmctl: Commands

- Download bmctl: `gsutil cp gs://anthos-baremetal-release/bmctl/1.8.2/linux-amd64/bmctl .`
- Get bmctl's version: `bmctl version`
- Create a cluster configuration: `bmctl create config`
  - Flag to specify the name of the cluster: `-c <cluster-name>`
  - Flag to enable the necessary Google Cloud APIs: `--enable-apis`
  - Flag to create the necessary Service Accounts: `--create-service-accounts`
- Create a cluster: `bmctl create cluster -c <cluster-name>`

Google Cloud

These are the commands that you would use to do the interactive steps from the previous diagram. You download the utility from Google Cloud Storage, create a config file, then create the cluster.



(1) Set up work to configure your cluster: Credentials

After bmctl has created the cluster configuration file, fill it with your data:

- If you used the --create-service-accounts flag, the Google Service Accounts are automatically created. Otherwise, create them manually.

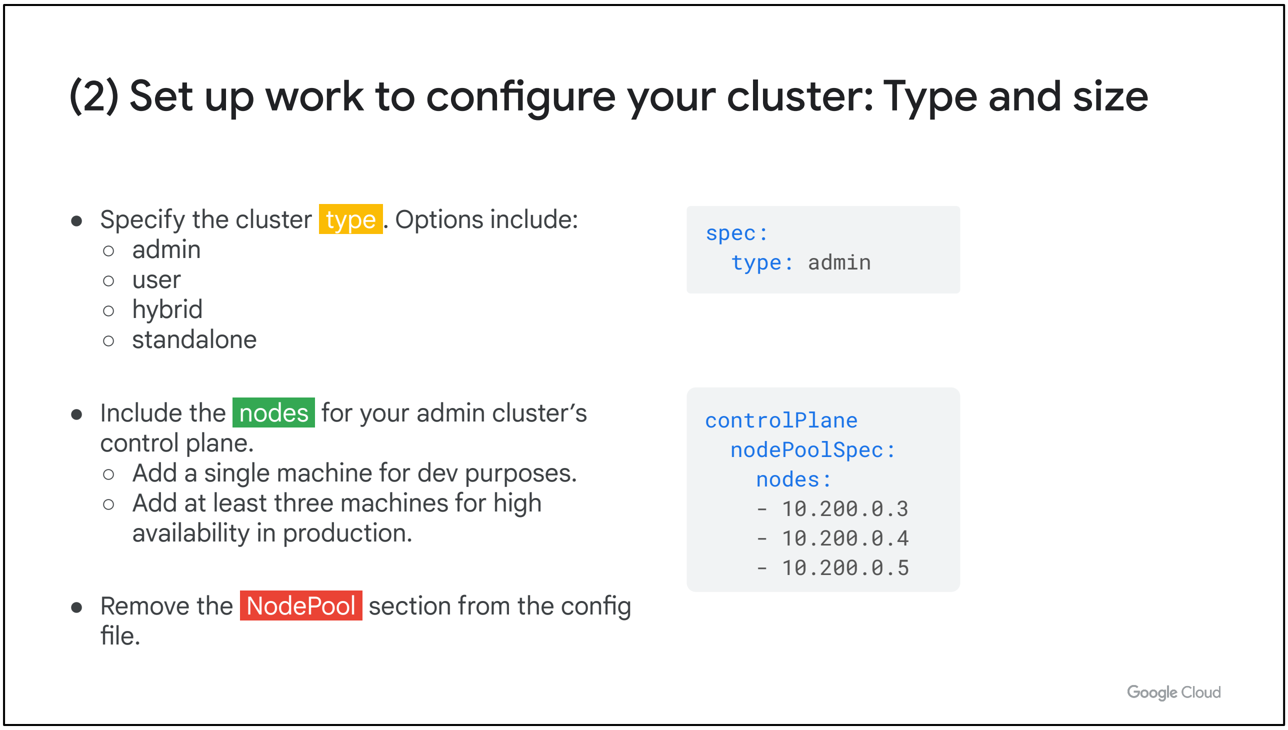
```
gcrKeyPath: bmctl-workspace/.sa-keys/gcr.json
gkeConnectAgentServiceAccountKeyPath: bmctl-workspace/.sa-keys/connect.json
gkeConnectRegisterServiceAccountKeyPath: bmctl-workspace/.sa-keys/register.json
cloudOperationsServiceAccountKeyPath: bmctl-workspace/.sa-keys/cloud-ops.json
```

- Create an ssh key, distribute it across the cluster nodes, and reference it in the config file.

```
sshPrivateKeyPath: /root/.ssh/id_rsa
```

Google Cloud

The bmctl utility will generate a generic configuration file for your cluster, but you will need to update the file as appropriate for your installation. The configuration file will require paths to various keys used for Service Account authentication and SSH connections to cluster nodes. If you used the --create-service-accounts flag, the key files will have been generated for you and the paths in the config file populated. You will need to copy a usable SSH private key to the admin workstation and populate the path value in the config file yourself.



(2) Set up work to configure your cluster: Type and size

- Specify the cluster type. Options include:
  - admin
  - user
  - hybrid
  - standalone

```
spec:
  type: admin
```

- Include the nodes for your admin cluster's control plane.
  - Add a single machine for dev purposes.
  - Add at least three machines for high availability in production.

```
controlPlane
  nodePoolSpec:
    nodes:
    - 10.200.0.3
    - 10.200.0.4
    - 10.200.0.5
```

- Remove the NodePool section from the config file.

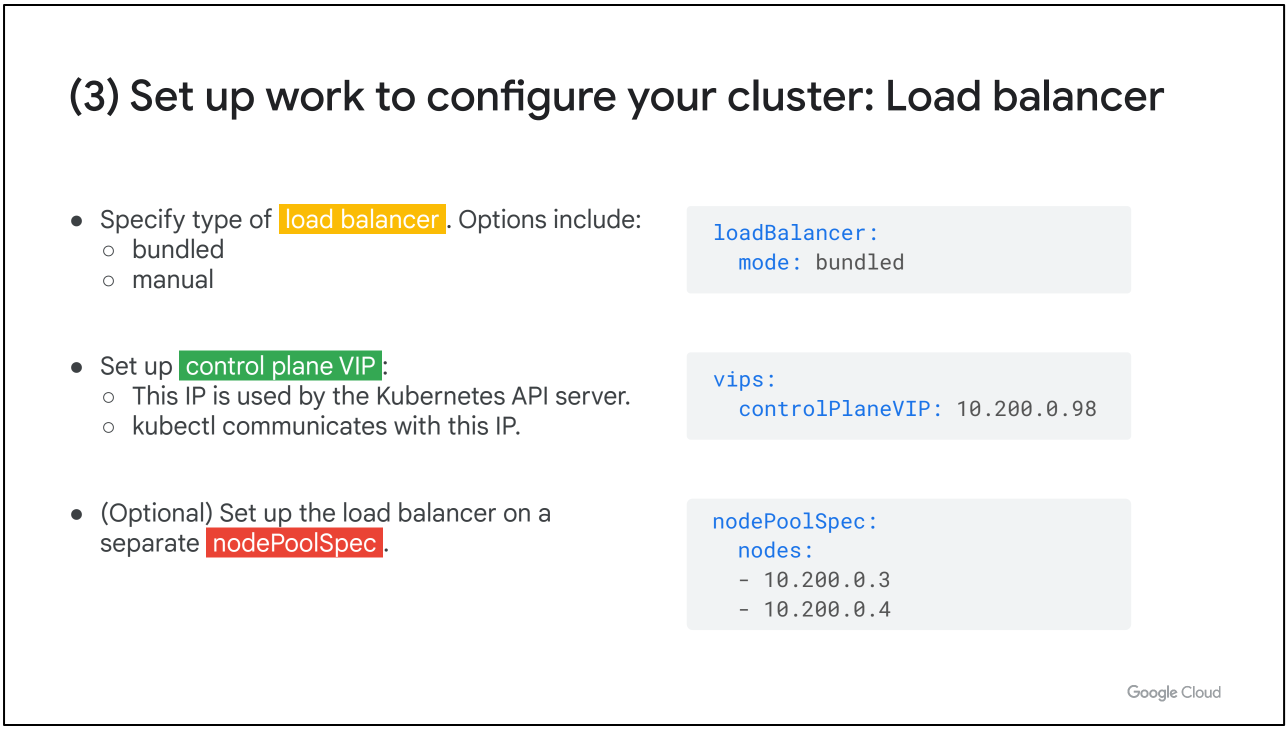
Google Cloud

When creating a new cluster, you need to specify the cluster type and the size of the cluster.

New clusters can be admin clusters – the clusters used to build and manage other clusters. They can be user clusters – the clusters you use to deploy your workloads. They can be hybrid clusters, which are basically admin clusters that also run application workloads. Or they can be standalone clusters, which run workloads but are self-contained, and are not managed by an admin cluster.

In the controlPlane section of the config file, you specify the IP addresses of the control plane nodes. For production clusters, you should use a High Availability configuration which would use at least three control plane machines.

Most admin clusters don't need worker nodes, so you can simply remove the NodePool section at the bottom of the config file which specifies worker node addresses.



(3) Set up work to configure your cluster: Load balancer

- Specify type of load balancer. Options include:
  - bundled
  - manual

```
loadBalancer:
  mode: bundled
```

- Set up control plane VIP:
  - This IP is used by the Kubernetes API server.
  - kubectl communicates with this IP.

```
vips:
  controlPlaneVIP: 10.200.0.98
```

- (Optional) Set up the load balancer on a separate nodePoolSpec.

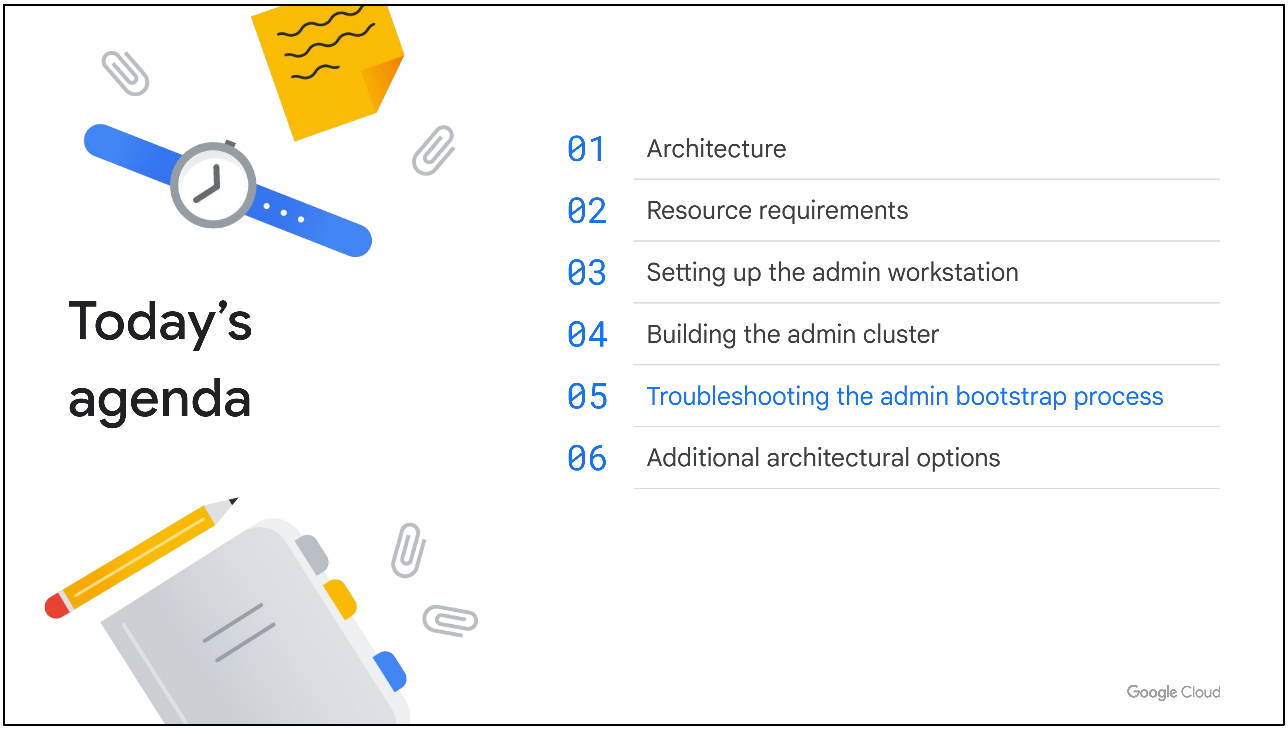
```
nodePoolSpec:
  nodes:
  - 10.200.0.3
  - 10.200.0.4
```

Bare metal clusters require a load balancing solution, and you will configure that during installation.

In the loadBalancer section of the configuration file, you specify whether to use the bundled load balancing software or an alternative, manually installed and configured load balancer.

You'll need to reserve a virtual IP for accessing the Kubernetes API server, which is done by setting the controlPlaneVIP value in the config file.

If you are using the bundled load balancer and want to run the load balancing software on machines that are separate from the cluster control plane, you can populate the nodePoolSpec section of the loadBalancer section of the config file.



Today's agenda

01 Architecture

02 Resource requirements

03 Setting up the admin workstation

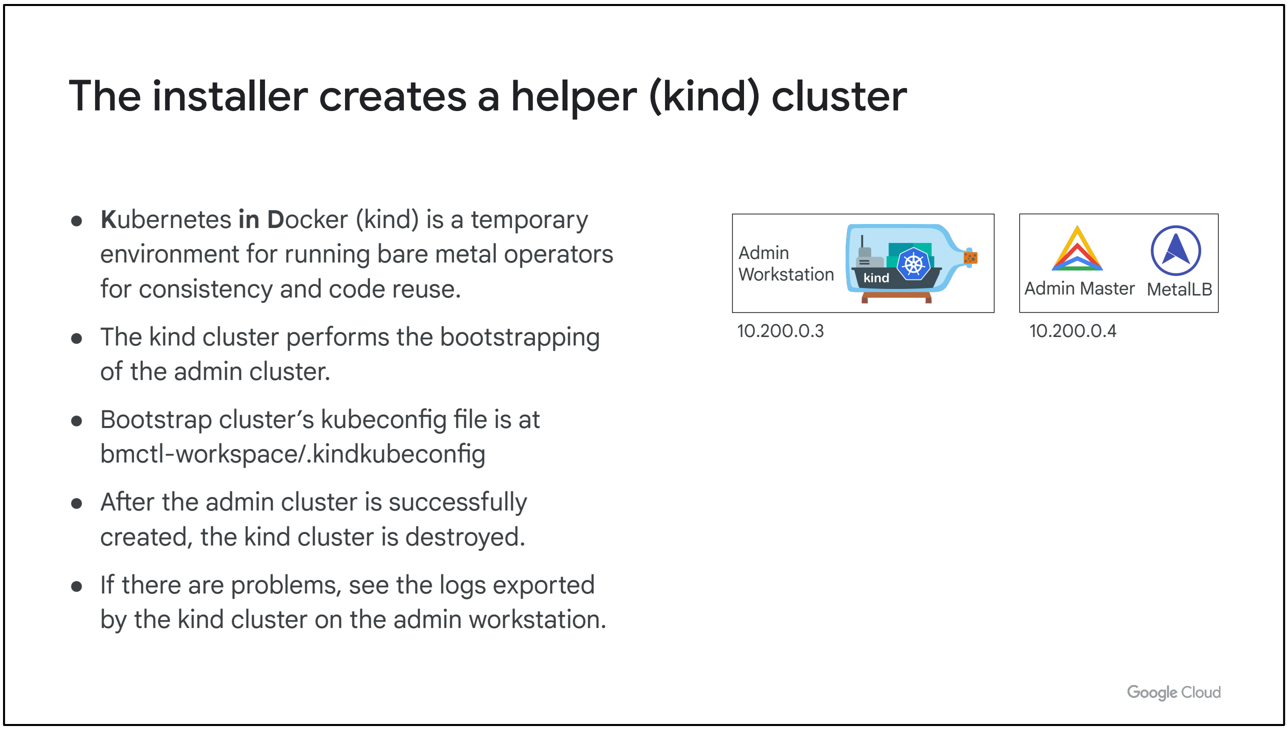
04 Building the admin cluster

05 Troubleshooting the admin bootstrap process

06 Additional architectural options

Google Cloud

Now, most of the time admin cluster creation goes smoothly. But let's look at some information that could be helpful in troubleshooting just in case you run into a problem.



The installer creates a helper (kind) cluster

- **K**ubernetes **in D**ocker (kind) is a temporary environment for running bare metal operators for consistency and code reuse.
- The kind cluster performs the bootstrapping of the admin cluster.
- Bootstrap cluster's kubeconfig file is at bmctl-workspace/.kindkubeconfig
- After the admin cluster is successfully created, the kind cluster is destroyed.
- If there are problems, see the logs exported by the kind cluster on the admin workstation.

Admin Workstation

kind

10.200.0.3

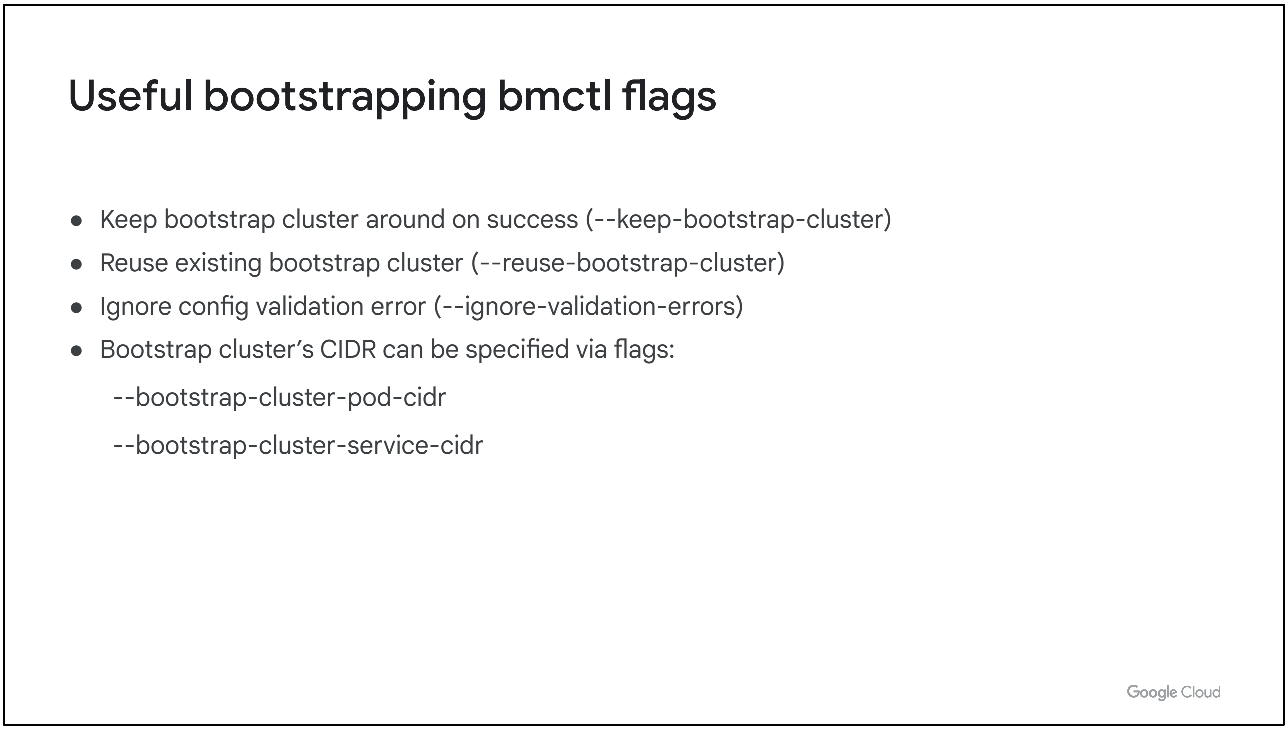
Admin Master

MetalLB

10.200.0.4

Google Cloud

When you use bmctl to create an admin cluster, it actually starts by creating a Kubernetes in Docker deployment. This is a temporary Kubernetes deployment running on the admin workstation, which has the software necessary to build the admin workstation. The config file for the kind cluster can be found in the bmctl-workspace directory, and kind logs are written locally for troubleshooting.



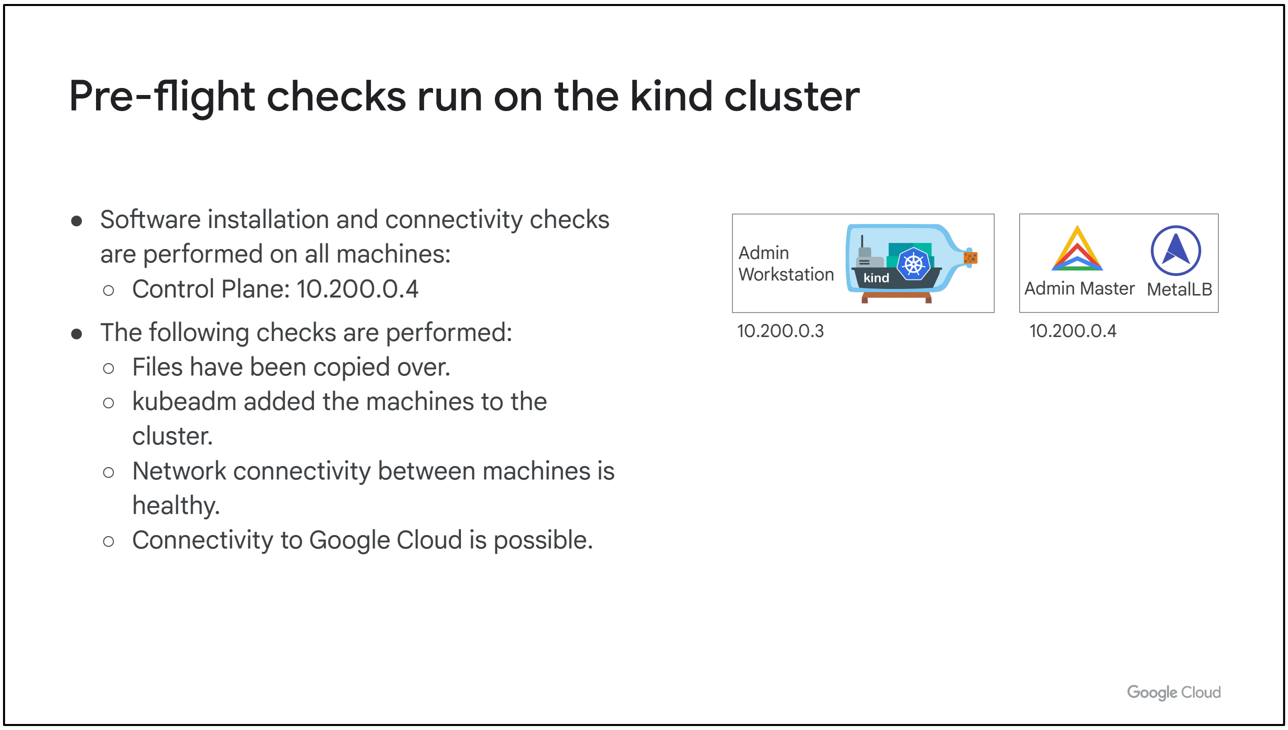
Useful bootstrapping bmctl flags

- Keep bootstrap cluster around on success (--keep-bootstrap-cluster)
- Reuse existing bootstrap cluster (--reuse-bootstrap-cluster)
- Ignore config validation error (--ignore-validation-errors)
- Bootstrap cluster’s CIDR can be specified via flags:

  --bootstrap-cluster-pod-cidr

  --bootstrap-cluster-service-cidr

You can modify the behavior of the kind environment with various bmctl flags. If you are doing multiple repetitive operations, you can avoid tearing down and rebuilding the kind environment. You can also tell bmctl to ignore configuration validation issues.

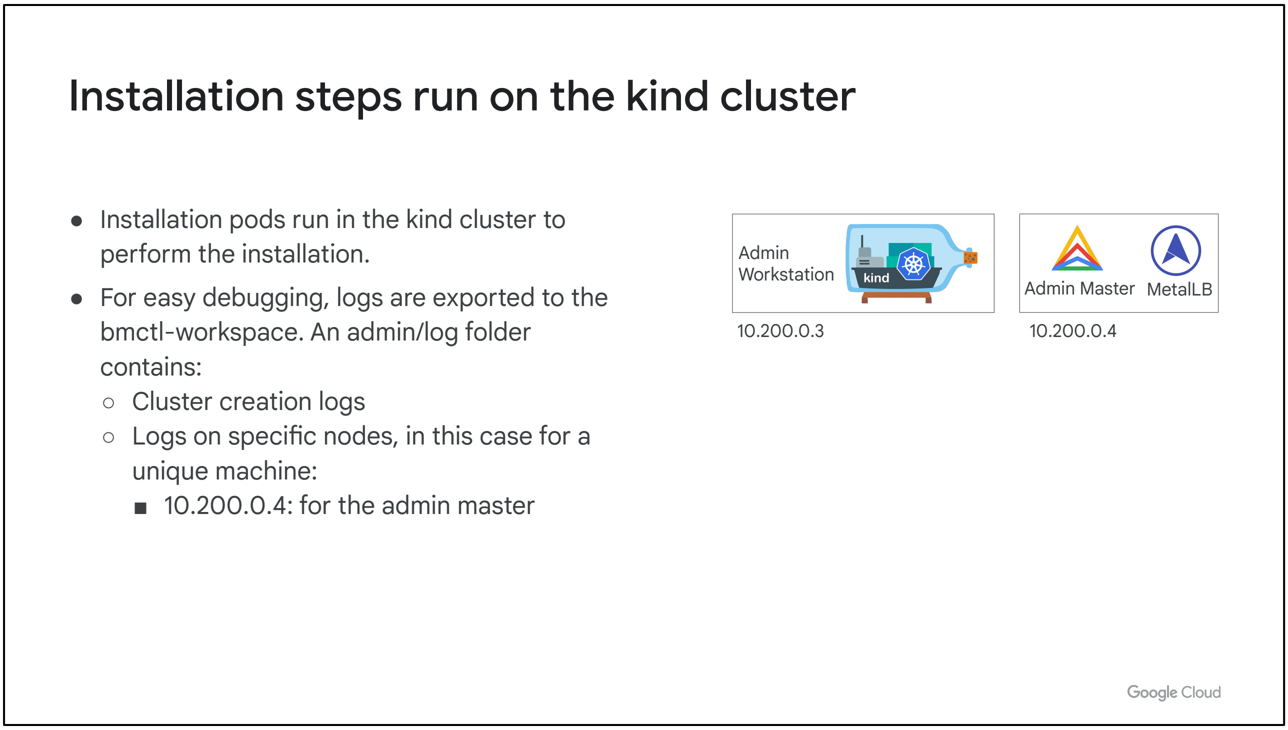


Pre-flight checks run on the kind cluster

- Software installation and connectivity checks are performed on all machines:
  - Control Plane: 10.200.0.4
- The following checks are performed:
  - Files have been copied over.
  - kubeadm added the machines to the cluster.
  - Network connectivity between machines is healthy.
  - Connectivity to Google Cloud is possible.

Admin Workstation
kind
10.200.0.3
Admin Master
MetalLB
10.200.0.4
Google Cloud

Prior to installing components, pre-flight checks are run on the kind cluster. It will make sure files have been copied into place, that network connectivity is good, that the machines can connect to Google Cloud services, etc.



Installation steps run on the kind cluster

- Installation pods run in the kind cluster to perform the installation.
- For easy debugging, logs are exported to the bmctl-workspace. An admin/log folder contains:
  - Cluster creation logs
  - Logs on specific nodes, in this case for a unique machine:
    - 10.200.0.4: for the admin master

Admin Workstation
kind
10.200.0.3
Admin Master
MetalLB
10.200.0.4

The actual admin cluster installation process runs in pods running in the kind cluster. As noted earlier, the logs are written locally into the bmctl-workspace directory. You will find logs for the overall cluster creation process, as well as for the setup of each node.

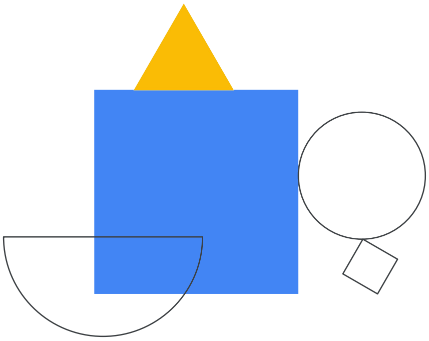
Lab intro

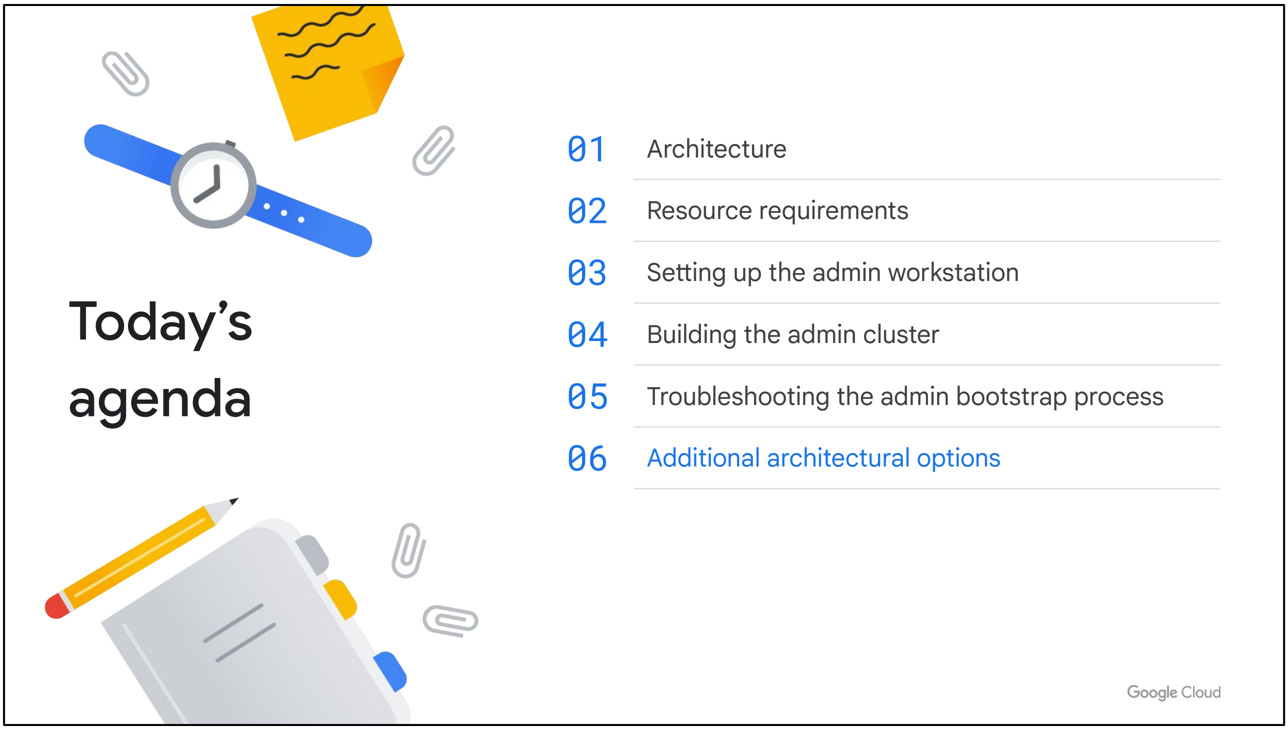
75 min

Lab 091: Creating Infrastructure and Deploying Anthos Clusters on Bare Metal

Come back at 12:10

In task 3: change the environment variable export to WS_MACHINE_TYPE=n1-standard-4





Today's agenda

01 Architecture

02 Resource requirements

03 Setting up the admin workstation

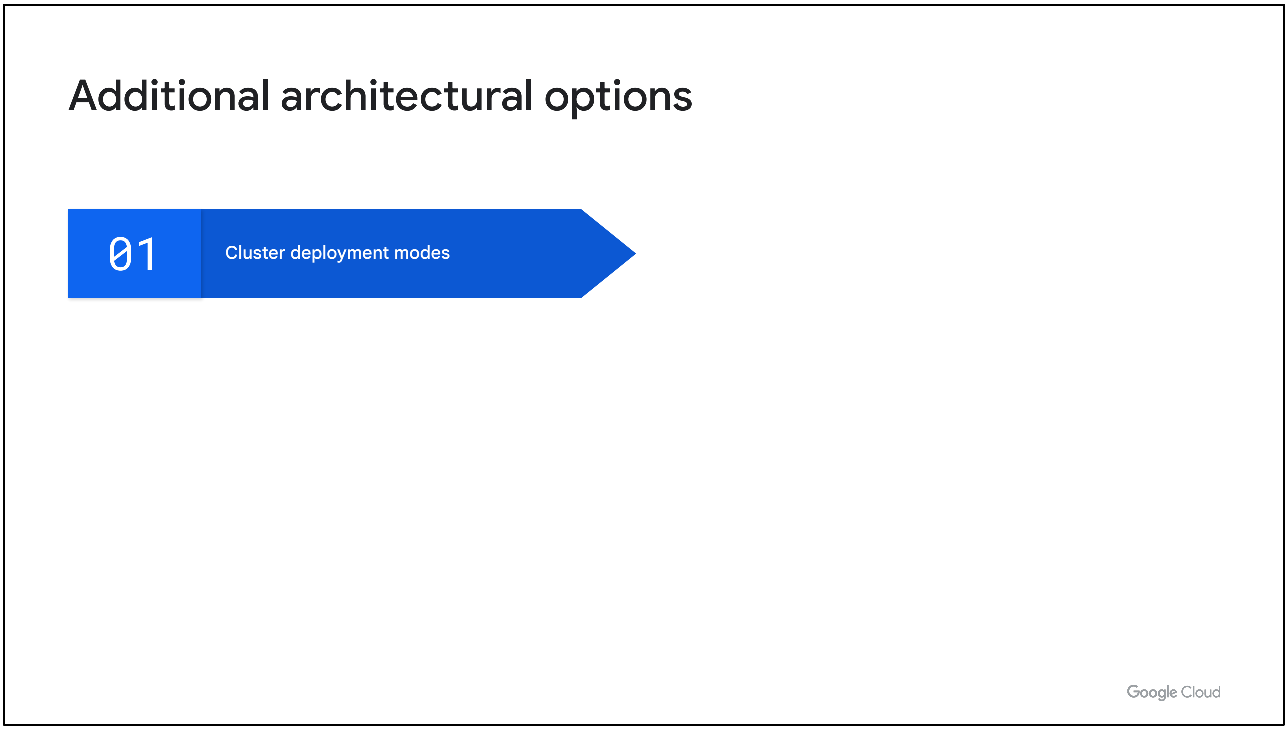
04 Building the admin cluster

05 Troubleshooting the admin bootstrap process

06 Additional architectural options

Google Cloud

For most of this module, we've assumed that you'd be using the most common Anthos on bare metal architecture, using an admin cluster and one or more user clusters, along with bundled load balancing. There are other options though.

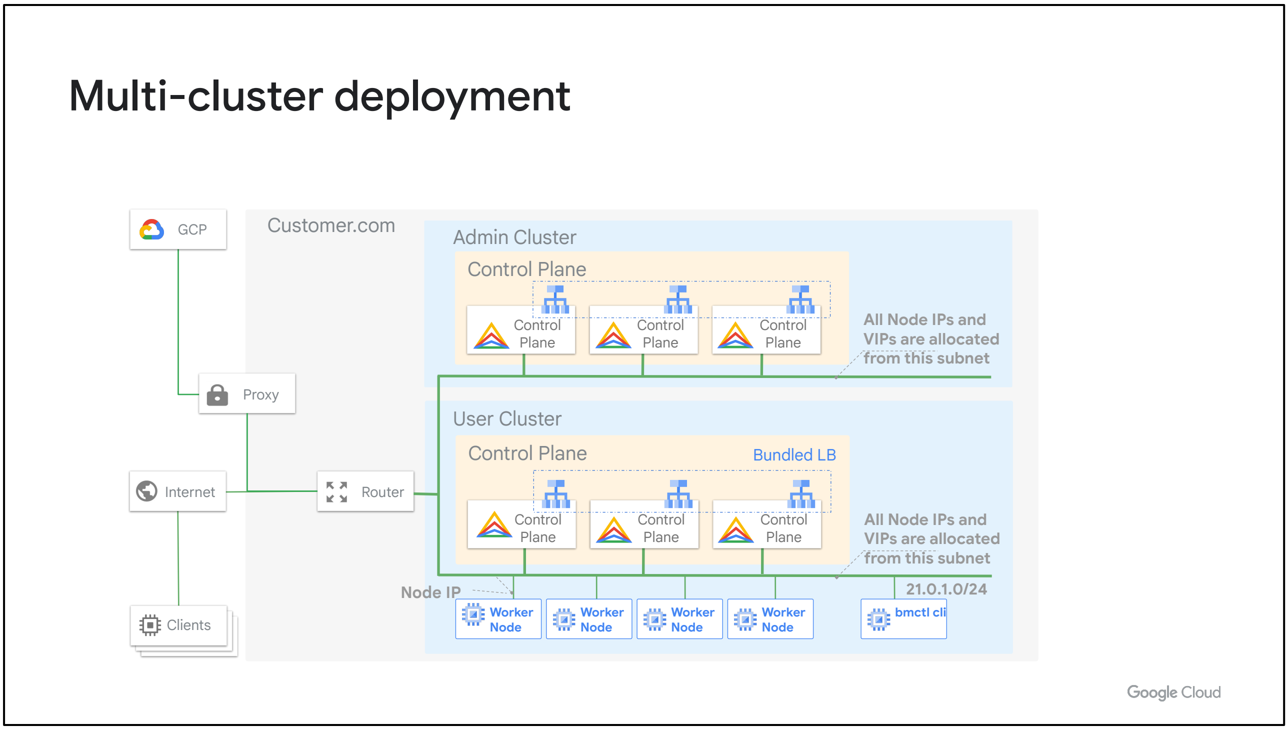


Additional architectural options

01

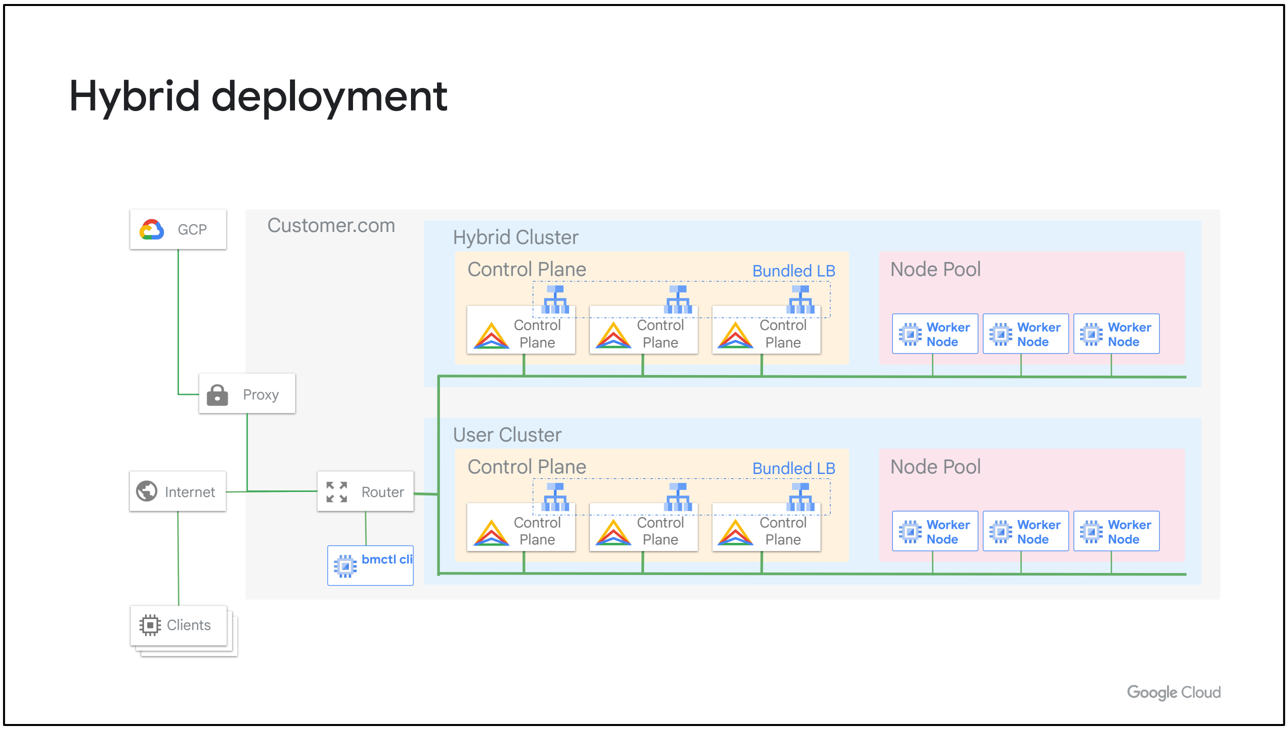
Cluster deployment modes

First, let's explore what Google calls "deployment models".



Multi-cluster deployment

The deployment model we've been focused on is the multi-cluster model. This is where you have one or more admin clusters, and one or more user clusters per admin cluster.

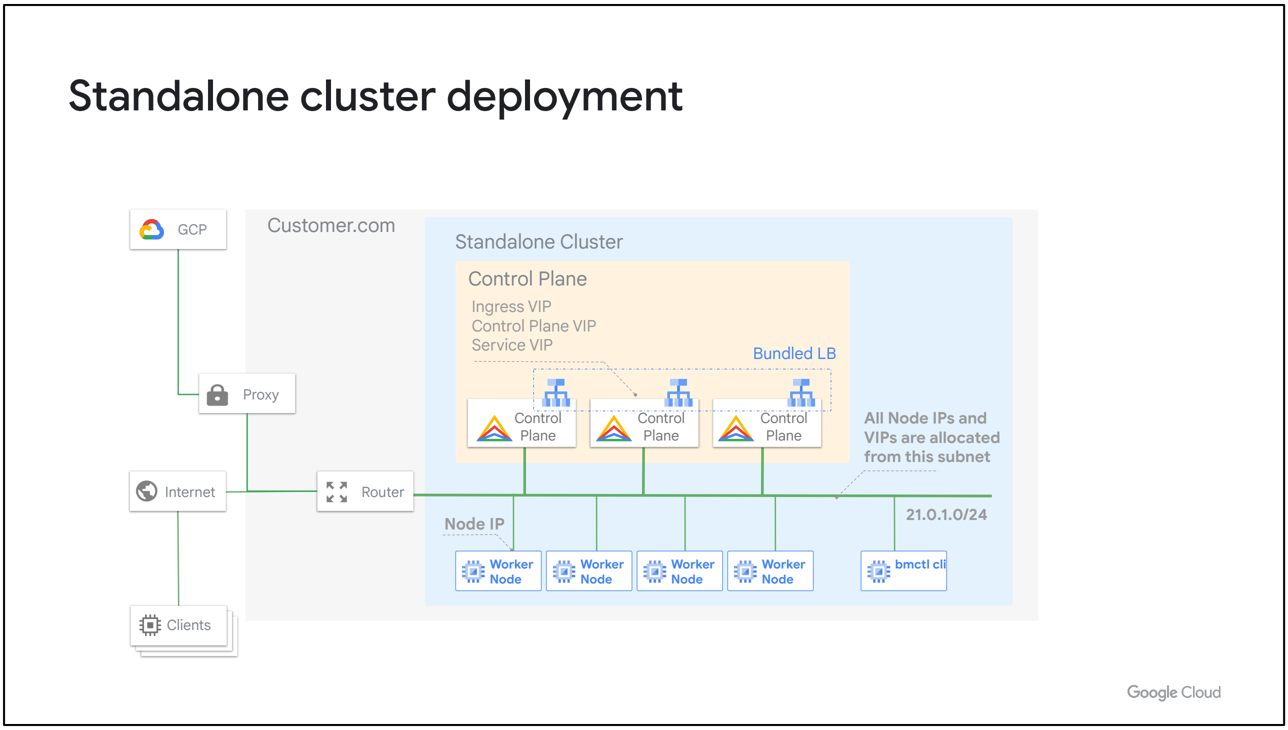


Hybrid deployment

GCP
Proxy
Internet
Router
bmctl cli
Clients
Customer.com
Hybrid Cluster
Control Plane
Bundled LB
Control Plane
Control Plane
Control Plane
Node Pool
Worker Node
Worker Node
Worker Node
User Cluster
Control Plane
Bundled LB
Control Plane
Control Plane
Control Plane
Node Pool
Worker Node
Worker Node
Worker Node

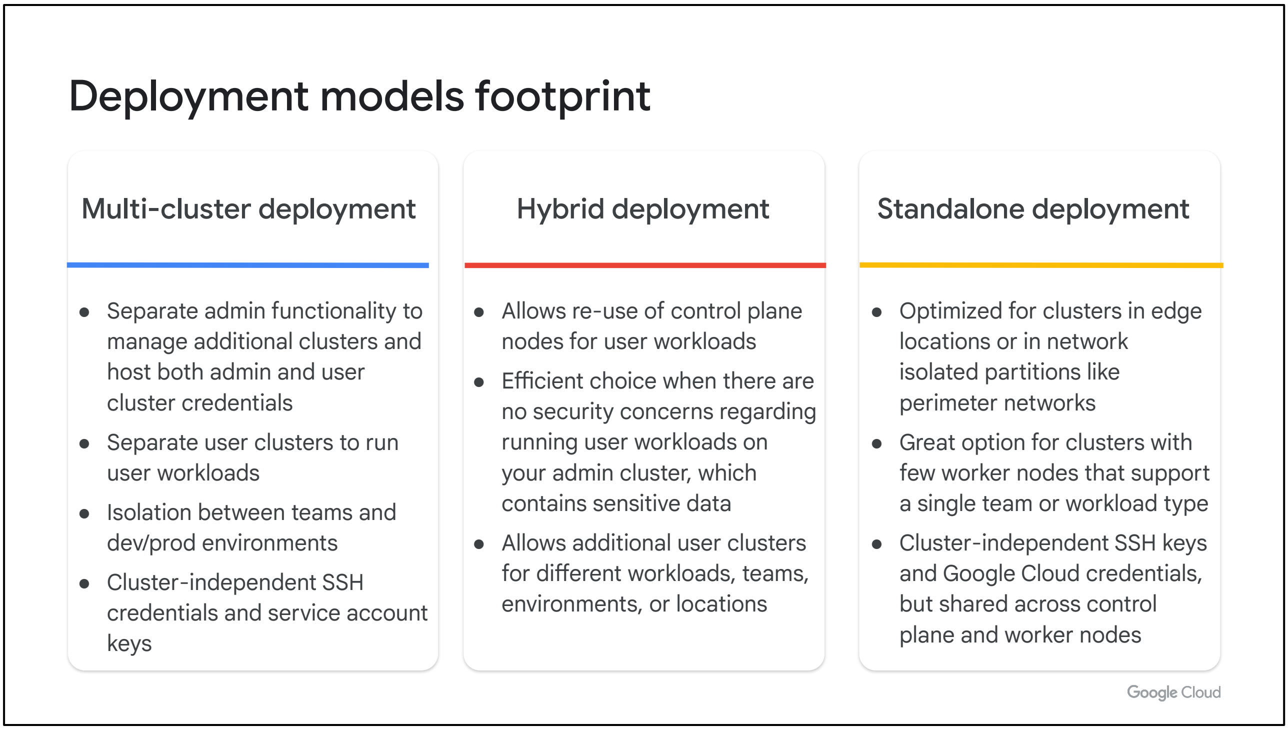
Google Cloud

A hybrid-cluster model is very much like the multi-cluster model, except that your admin clusters also have worker nodes and you deploy applications, above and beyond the Anthos management software, onto the cluster.



Standalone cluster deployment

A standalone deployment means that a cluster is self-contained. It doesn't require a separate admin cluster and can reduce your hardware requirements significantly.



Deployment models footprint

Multi-cluster deployment

- Separate admin functionality to manage additional clusters and host both admin and user cluster credentials
- Separate user clusters to run user workloads
- Isolation between teams and dev/prod environments
- Cluster-independent SSH credentials and service account keys

Hybrid deployment

- Allows re-use of control plane nodes for user workloads
- Efficient choice when there are no security concerns regarding running user workloads on your admin cluster, which contains sensitive data
- Allows additional user clusters for different workloads, teams, environments, or locations

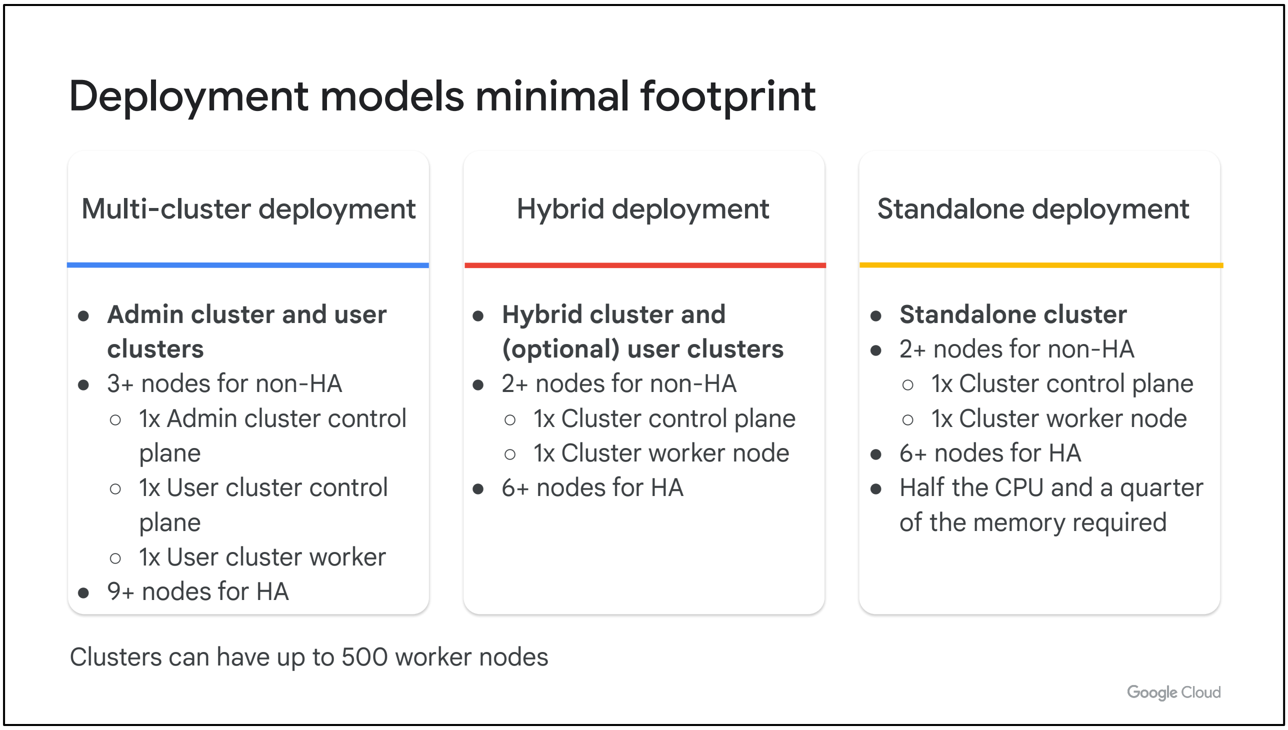
Standalone deployment

- Optimized for clusters in edge locations or in network isolated partitions like perimeter networks
- Great option for clusters with few worker nodes that support a single team or workload type
- Cluster-independent SSH keys and Google Cloud credentials, but shared across control plane and worker nodes

So, multi-cluster deployment separates cluster admin and workload admin. You also use a single set of security credentials to manage across many clusters.

Hybrid deployment allows you to leverage spare capacity on the admin clusters, but means that some workloads may have access to security credentials normally stored on the admin cluster.

Standalone deployments are optimized for edge clusters. Every cluster must be managed separately, and every cluster uses unique credentials, with those credentials being potentially accessible by non-admin workloads.



Deployment models minimal footprint

Multi-cluster deployment

- **Admin cluster and user clusters**
- 3+ nodes for non-HA
  - 1x Admin cluster control plane
  - 1x User cluster control plane
  - 1x User cluster worker
- 9+ nodes for HA

Hybrid deployment

- **Hybrid cluster and (optional) user clusters**
- 2+ nodes for non-HA
  - 1x Cluster control plane
  - 1x Cluster worker node
- 6+ nodes for HA

Standalone deployment

- **Standalone cluster**
- 2+ nodes for non-HA
  - 1x Cluster control plane
  - 1x Cluster worker node
- 6+ nodes for HA
- Half the CPU and a quarter of the memory required

Clusters can have up to 500 worker nodes

Google Cloud

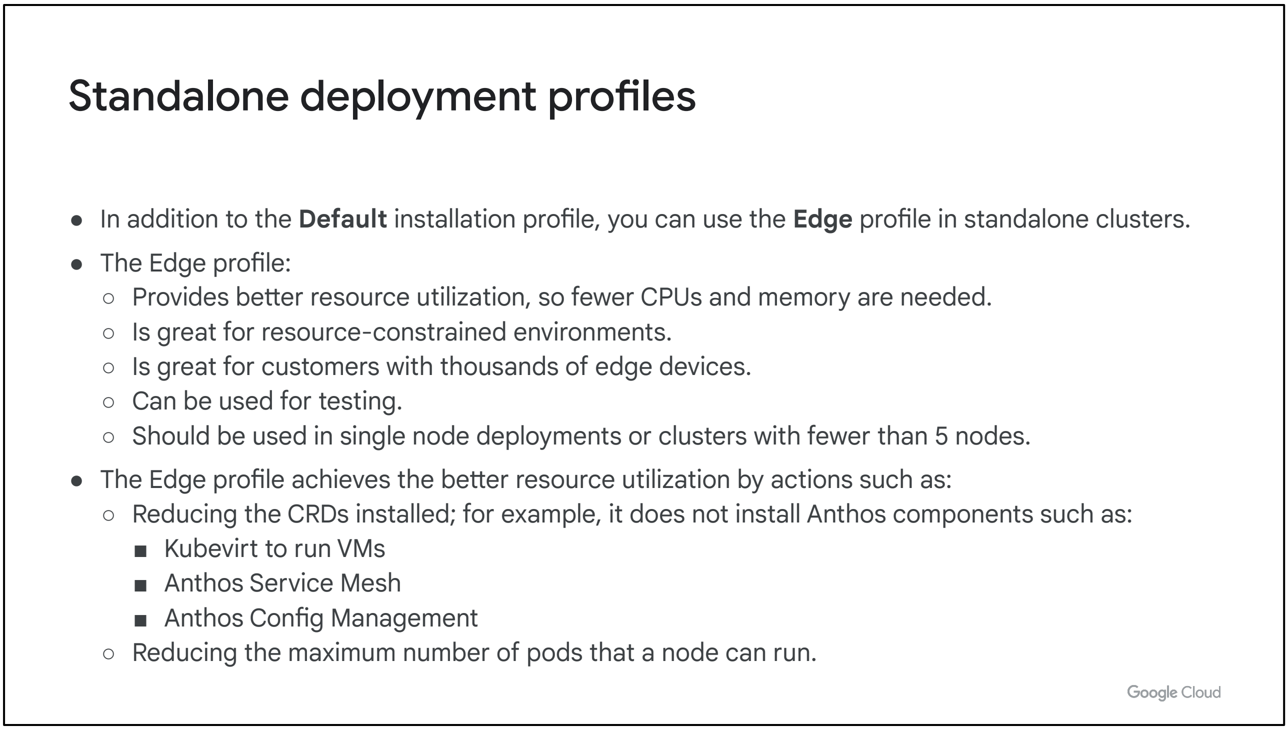
Different deployment models will obviously have different hardware requirements overall.

For multi-cluster deployments, you will typically have at least three servers, one for the admin cluster control plane, one for the user cluster control plane, and one user cluster worker. For HA deployments, you would use at least nine plus servers, three for the admin cluster and six for the user cluster.

For a hybrid deployment, you would have at least two servers for the hybrid cluster, plus at least two for any optional user clusters you create. For an HA configuration, you would be using at least six servers for the hybrid cluster and six for any added user clusters.

For a hybrid deployment, you would have at least two servers for the hybrid cluster, plus at least two for any optional user clusters you create. For an HA configuration, you would be using at least six servers for the hybrid cluster and six for any added user clusters.

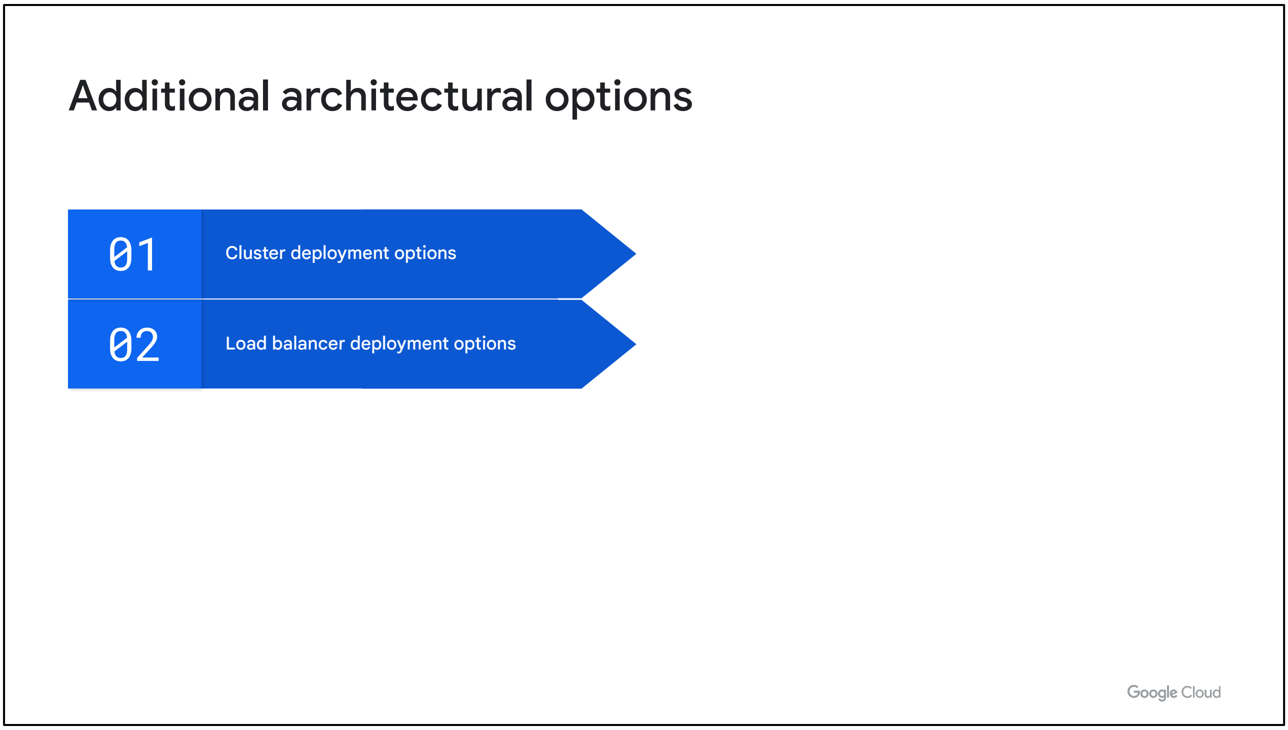
For a standalone deployment, you would have two servers for a non-HA cluster and six servers for an HA cluster. However, each node will require significantly less CPU and memory.



Standalone deployment profiles

- In addition to the **Default** installation profile, you can use the **Edge** profile in standalone clusters.
- The Edge profile:
  - Provides better resource utilization, so fewer CPUs and memory are needed.
  - Is great for resource-constrained environments.
  - Is great for customers with thousands of edge devices.
  - Can be used for testing.
  - Should be used in single node deployments or clusters with fewer than 5 nodes.
- The Edge profile achieves the better resource utilization by actions such as:
  - Reducing the CRDs installed; for example, it does not install Anthos components such as:
    - Kubevirt to run VMs
    - Anthos Service Mesh
    - Anthos Config Management
  - Reducing the maximum number of pods that a node can run.

When creating a standalone, you can choose between a Default and an Edge profile. The Edge profile will install fewer features, foregoing installation of Anthos Service Mesh, Anthos Config Management, and others, in order to reduce the hardware demands on your servers. Google recommends this profile be used for small standalone clusters, typically with fewer than five nodes.

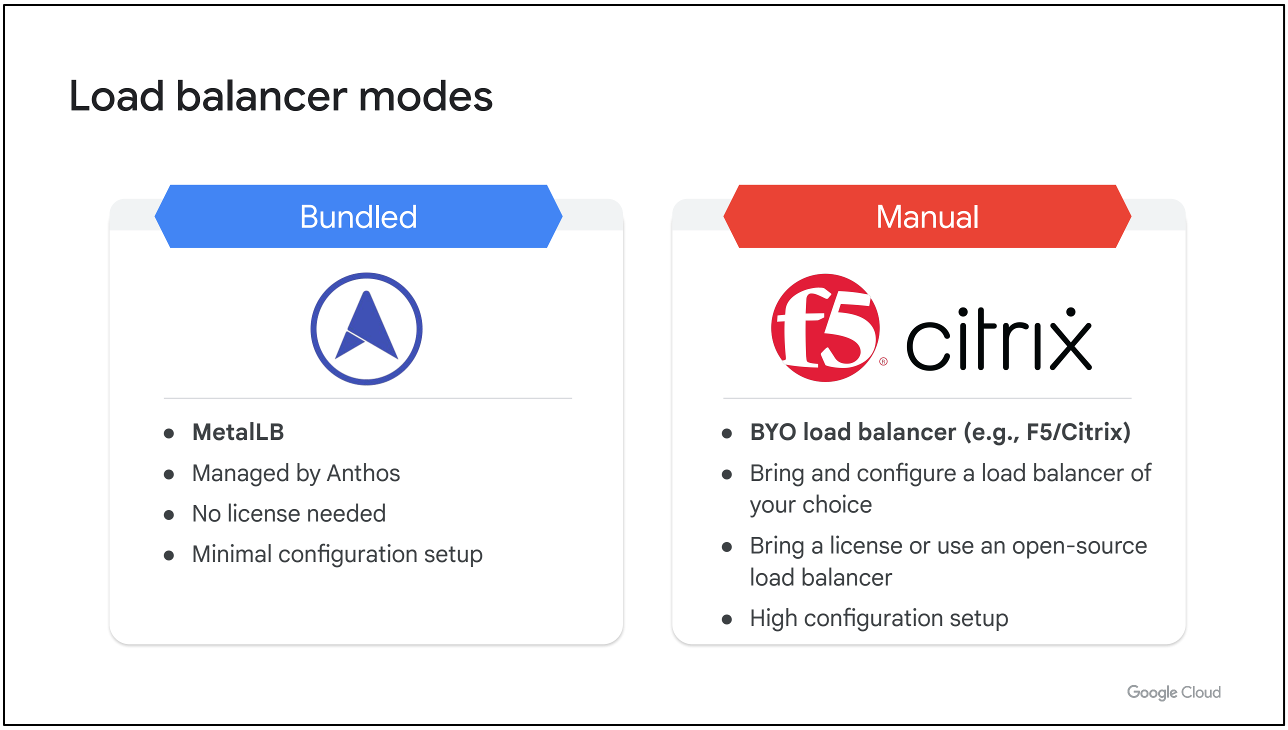


Additional architectural options

01 Cluster deployment options

02 Load balancer deployment options

In addition to choosing your cluster deployment model, you can choose how your Anthos bare metal clusters do load balancing.



Load balancer modes

Bundled

- **MetalLB**
- Managed by Anthos
- No license needed
- Minimal configuration setup

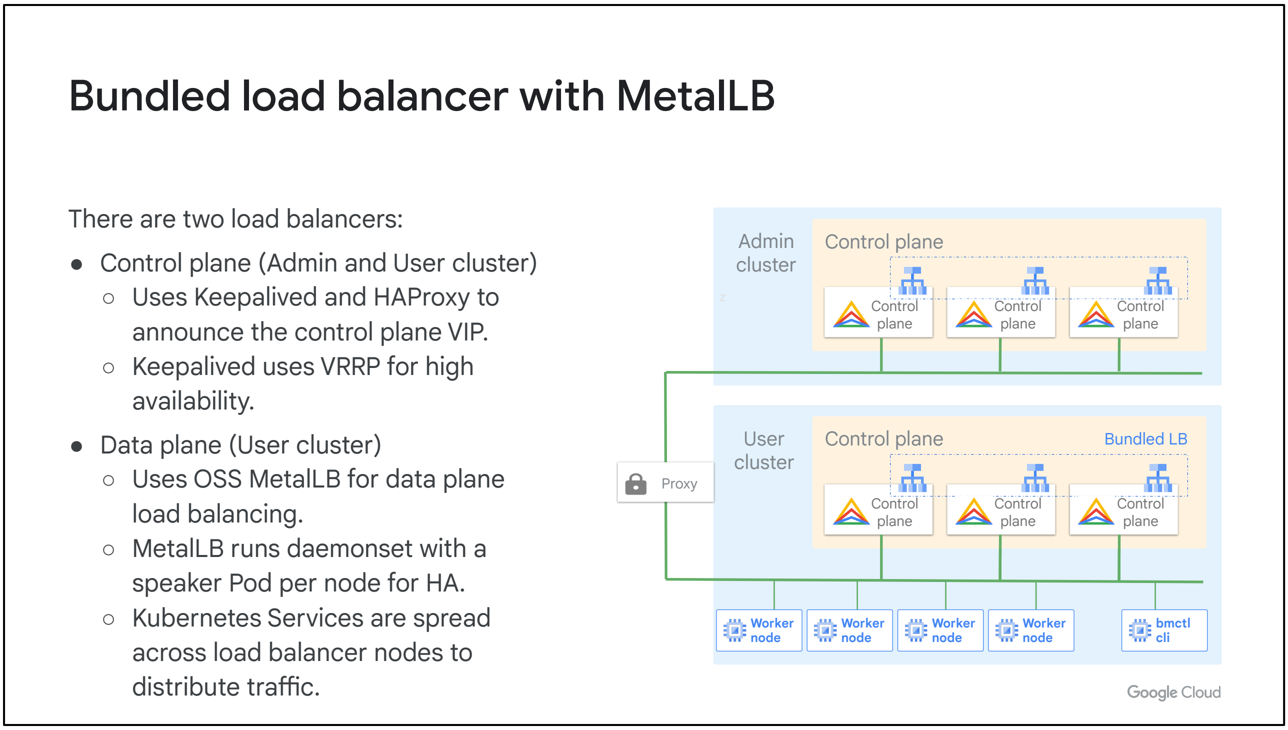
Manual

- **BYO load balancer (e.g., F5/Citrix)**
- Bring and configure a load balancer of your choice
- Bring a license or use an open-source load balancer
- High configuration setup

Google Cloud

Bundled load balancing is the default, and it uses a MetalLB as the data plane load balancing software. This solution doesn't require any additional paid licenses, and Anthos can automatically configure the load balancer to expose newly deployed LoadBalancer services on your cluster.

Alternatively, if a company already has licensed load balancers it would like to leverage, you can configure clusters to not use bundled load balancing. This requires more extensive setup of the load balancing solution during cluster creation, and Anthos doesn't automatically configure VIPs on the load balancer when new LoadBalancer services are deployed.

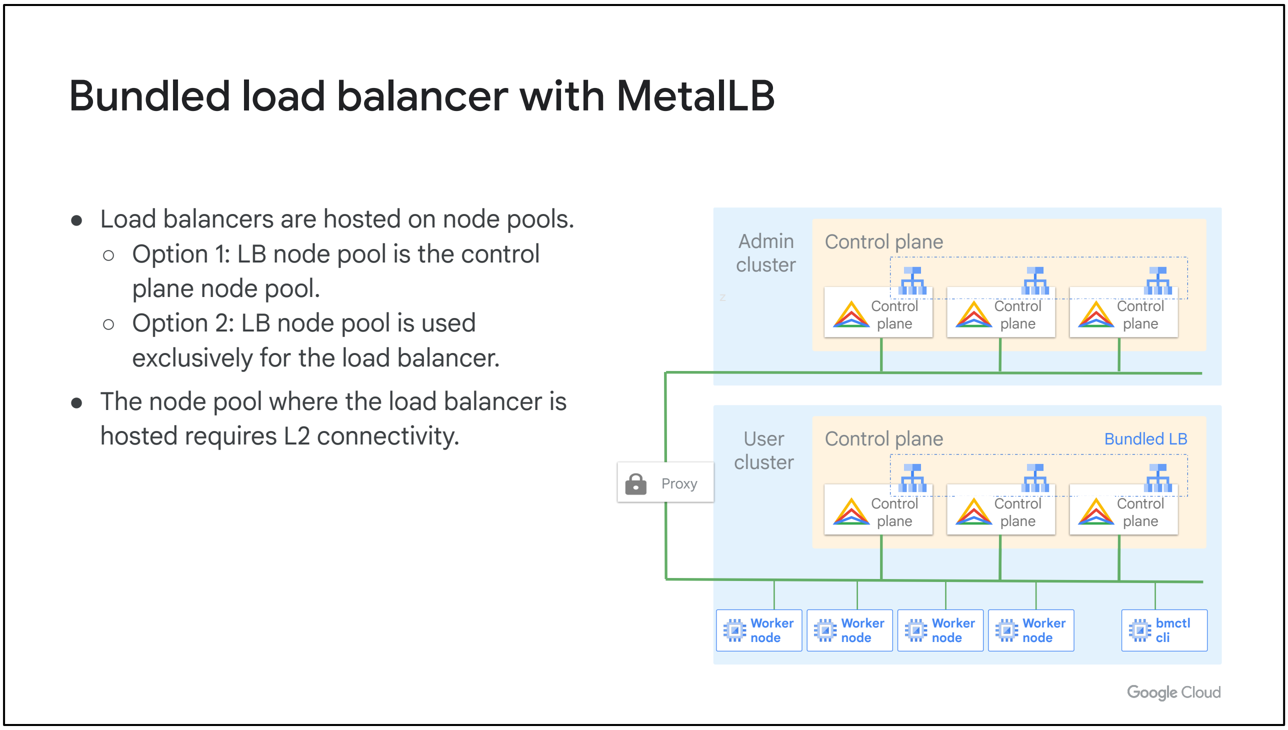


Bundled load balancer with MetalLB

There are two load balancers:

- Control plane (Admin and User cluster)
  - Uses Keepalived and HAProxy to announce the control plane VIP.
  - Keepalived uses VRRP for high availability.
- Data plane (User cluster)
  - Uses OSS MetalLB for data plane load balancing.
  - MetalLB runs daemonset with a speaker Pod per node for HA.
  - Kubernetes Services are spread across load balancer nodes to distribute traffic.

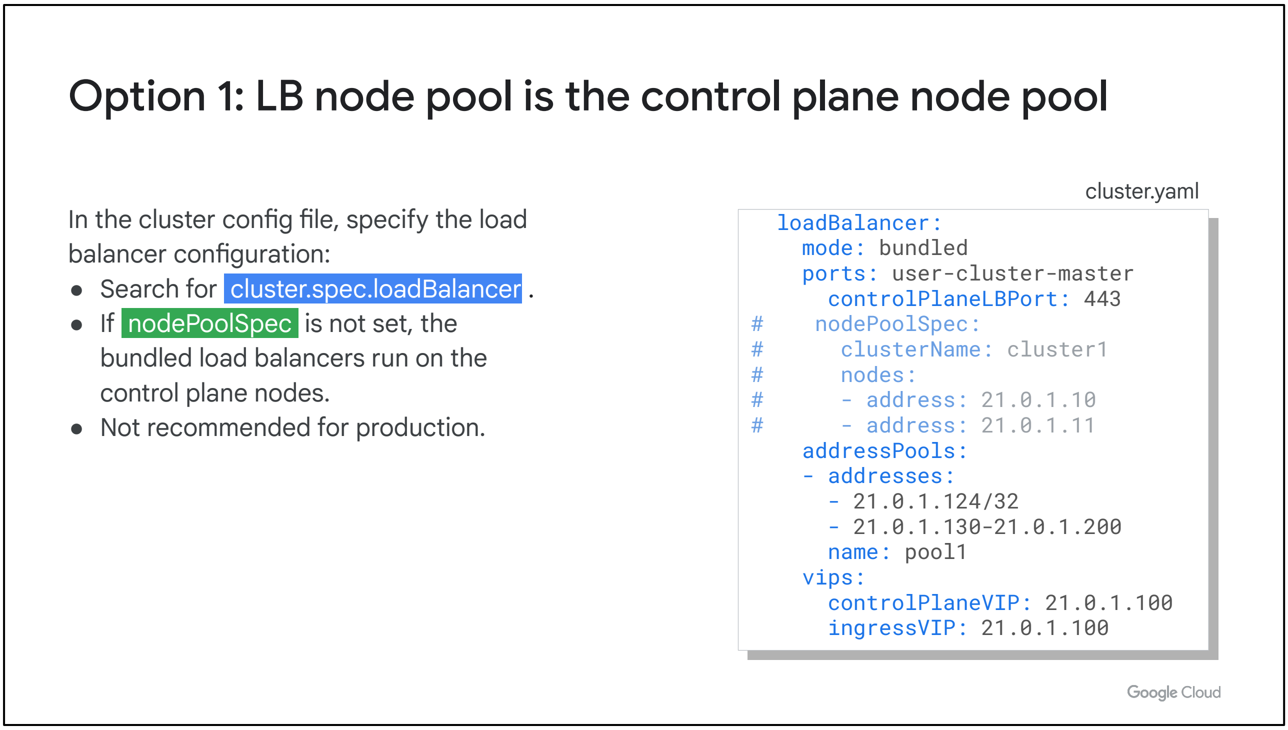
Let's drill down a bit into the bundled load balancing solution. The control plane is accessed via a load balancer that uses Keepalived and HAProxy. This software is installed and configured automatically by the bmctl utility, and has a high-availability option. The data plane load balancing is handled by MetalLB, which is also automatically installed and configured.



Bundled load balancer with MetalLB

- Load balancers are hosted on node pools.
  - Option 1: LB node pool is the control plane node pool.
  - Option 2: LB node pool is used exclusively for the load balancer.
- The node pool where the load balancer is hosted requires L2 connectivity.

As noted earlier, you can run the load balancing software either on the control plane servers, or on a separate load balancing node pool. All the load balancing servers must operate in the same L2 domain.



Option 1: LB node pool is the control plane node pool

In the cluster config file, specify the load balancer configuration:

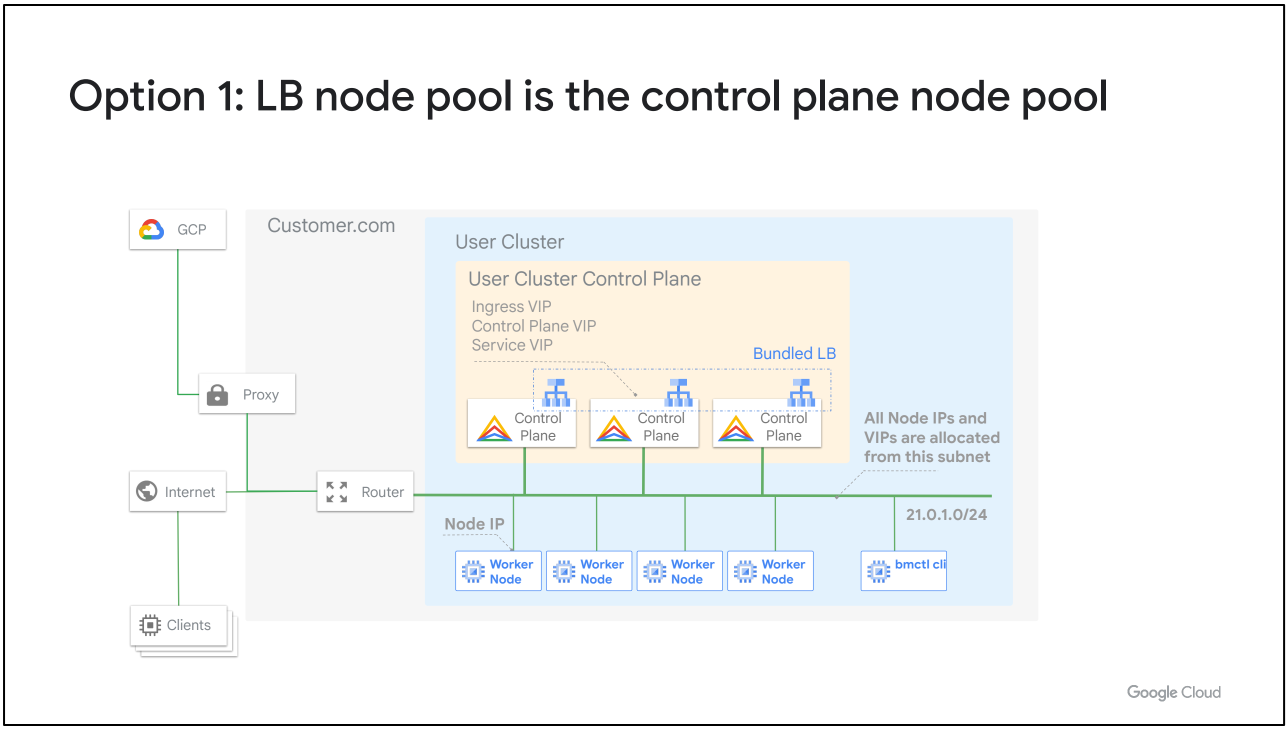
- Search for cluster.spec.loadBalancer .
- If nodePoolSpec is not set, the bundled load balancers run on the control plane nodes.
- Not recommended for production.

cluster.yaml

```
 loadBalancer:
  mode: bundled
  ports: user-cluster-master
   controlPlaneLBPort: 443
#  nodePoolSpec:
#   clusterName: cluster1
#   nodes:
#   - address: 21.0.1.10
#   - address: 21.0.1.11
  addressPools:
  - addresses:
   - 21.0.1.124/32
   - 21.0.1.130-21.0.1.200
   name: pool1
  vips:
   controlPlaneVIP: 21.0.1.100
   ingressVIP: 21.0.1.100
```

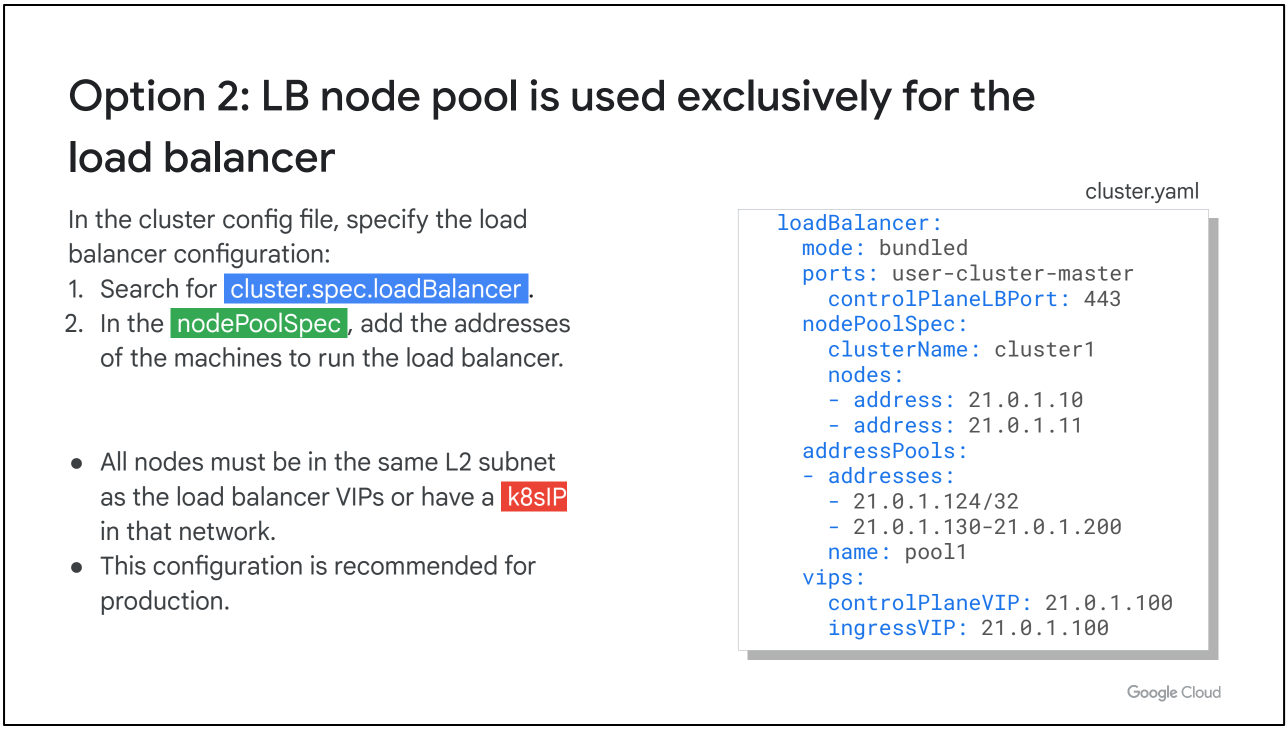
Google Cloud

To run the load balancer on the control plane nodes, leave the nodePoolSpec area of the loadBalancer section commented out. Google recommends this configuration only for non-production clusters.



Option 1: LB node pool is the control plane node pool

Here, you can see the cluster is running the MetalLB software on the control plane nodes.



Option 2: LB node pool is used exclusively for the load balancer

In the cluster config file, specify the load balancer configuration:

1. Search for cluster.spec.loadBalancer.
2. In the nodePoolSpec, add the addresses of the machines to run the load balancer.

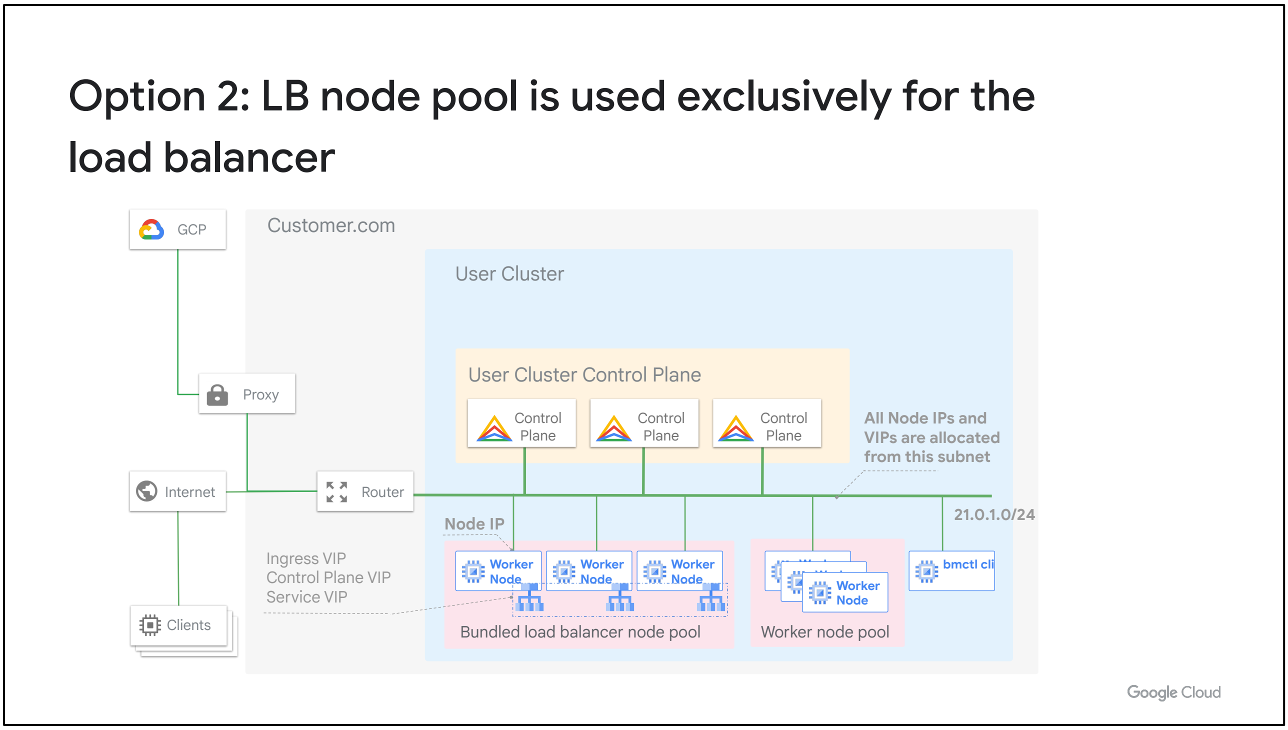
- All nodes must be in the same L2 subnet as the load balancer VIPs or have a k8sIP in that network.
- This configuration is recommended for production.

cluster.yaml

```yaml
loadBalancer:
  mode: bundled
  ports: user-cluster-master
    controlPlaneLBPort: 443
  nodePoolSpec:
    clusterName: cluster1
    nodes:
    - address: 21.0.1.10
    - address: 21.0.1.11
  addressPools:
  - addresses:
    - 21.0.1.124/32
    - 21.0.1.130-21.0.1.200
    name: pool1
  vips:
    controlPlaneVIP: 21.0.1.100
    ingressVIP: 21.0.1.100
```

Google Cloud

To use a separate load balancing node pool, simply populate the nodePoolSpec section of the config file as shown in the example. The IP addresses for the nodes must be in the same subnet as the VIPs that will be advertised, or must have a K8sIP in that subnet. This is the recommended configuration for production clusters.



Option 2: LB node pool is used exclusively for the load balancer

GCP

Proxy

Internet

Router

Clients

Customer.com

User Cluster

User Cluster Control Plane

Control Plane

Control Plane

Control Plane

All Node IPs and VIPs are allocated from this subnet

21.0.1.0/24

Node IP

Ingress VIP
Control Plane VIP
Service VIP

Worker Node

Worker Node

Worker Node

Bundled load balancer node pool

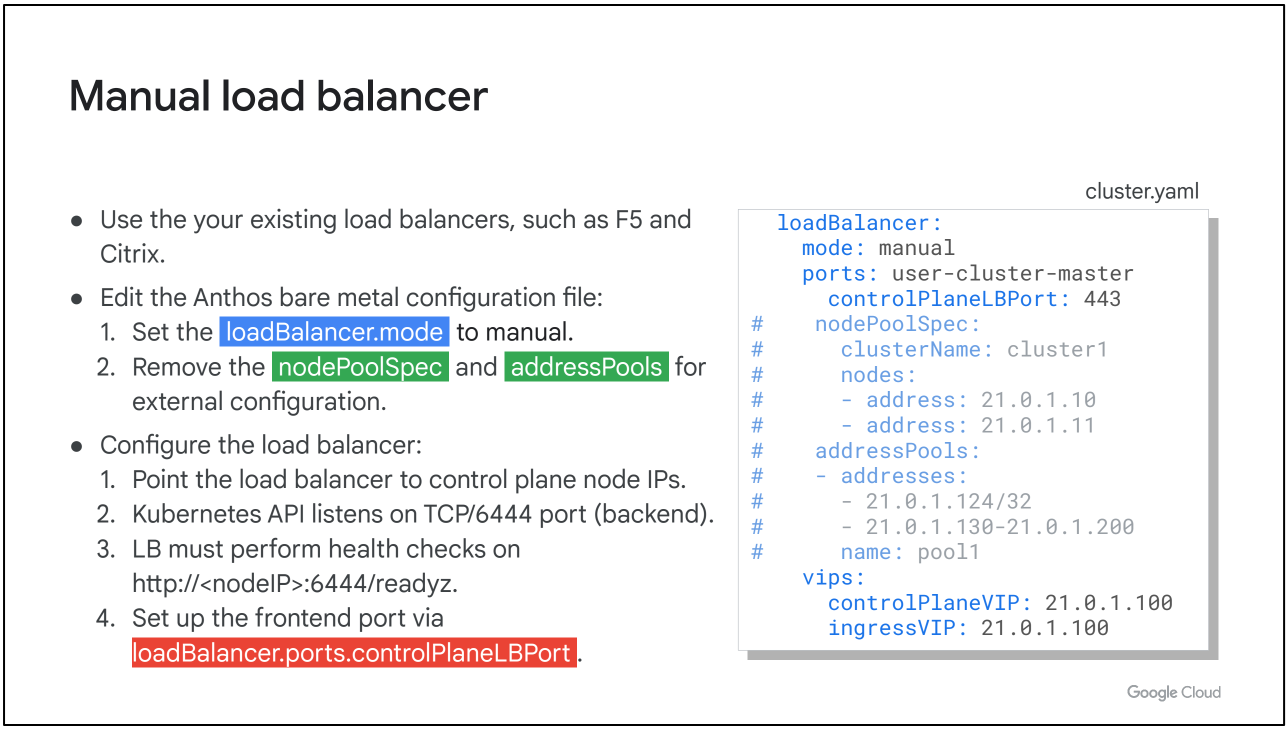
Worker Node

Worker node pool

bmctl cli

Google Cloud

Here, you can see the load balancing software is installed on a separate node pool.



Manual load balancer

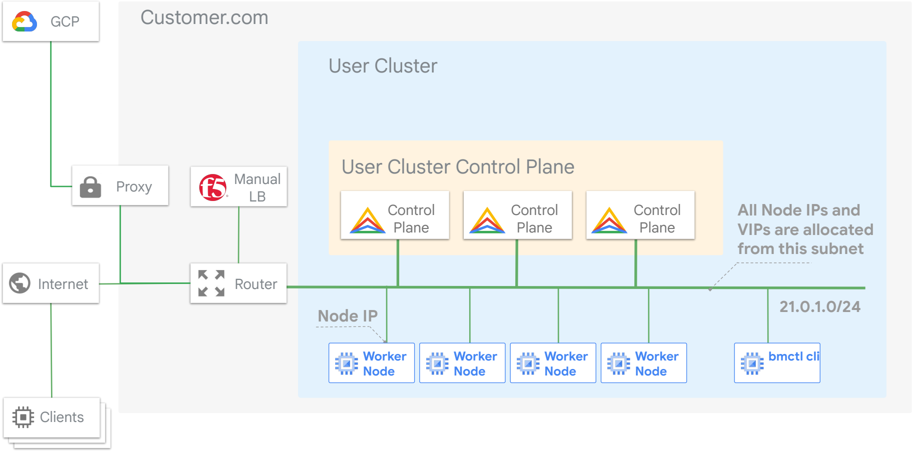
- Use the your existing load balancers, such as F5 and Citrix.
- Edit the Anthos bare metal configuration file:
  1. Set the loadBalancer.mode to manual.
  2. Remove the nodePoolSpec and addressPools for external configuration.
- Configure the load balancer:
  1. Point the load balancer to control plane node IPs.
  2. Kubernetes API listens on TCP/6444 port (backend).
  3. LB must perform health checks on http://<nodeIP>:6444/readyz.
  4. Set up the frontend port via loadBalancer.ports.controlPlaneLBPort.

cluster.yaml

```
  loadBalancer:
    mode: manual
    ports: user-cluster-master
      controlPlaneLBPort: 443
#    nodePoolSpec:
#      clusterName: cluster1
#      nodes:
#      - address: 21.0.1.10
#      - address: 21.0.1.11
#    addressPools:
#    - addresses:
#      - 21.0.1.124/32
#      - 21.0.1.130-21.0.1.200
#      name: pool1
    vips:
      controlPlaneVIP: 21.0.1.100
      ingressVIP: 21.0.1.100
```

To configure your clusters to use manual load balancing, edit the config file, changing the mode to manual and removing the nodePoolSpec and addressPools sections. You will then need to configure your load balancer with a virtual IP for the control plane and have it forward to the IP addresses of the control plane nodes on port 6444. Health checks will need to be configured to point to /readyz.

Manually provisioned load balancer



GCP

Customer.com

User Cluster

User Cluster Control Plane

Control Plane

Control Plane

Control Plane

All Node IPs and VIPs are allocated from this subnet

Proxy

Manual LB

Internet

Router

21.0.1.0/24

Node IP

Worker Node

Worker Node

Worker Node

Worker Node

bmctl cli

Clients

Google Cloud

Here, you can see no load balancing software on the servers in your cluster, rather F5 is running on a separate VM.

Questions and answers

1. What type of Anthos on bare metal cluster does not exist?

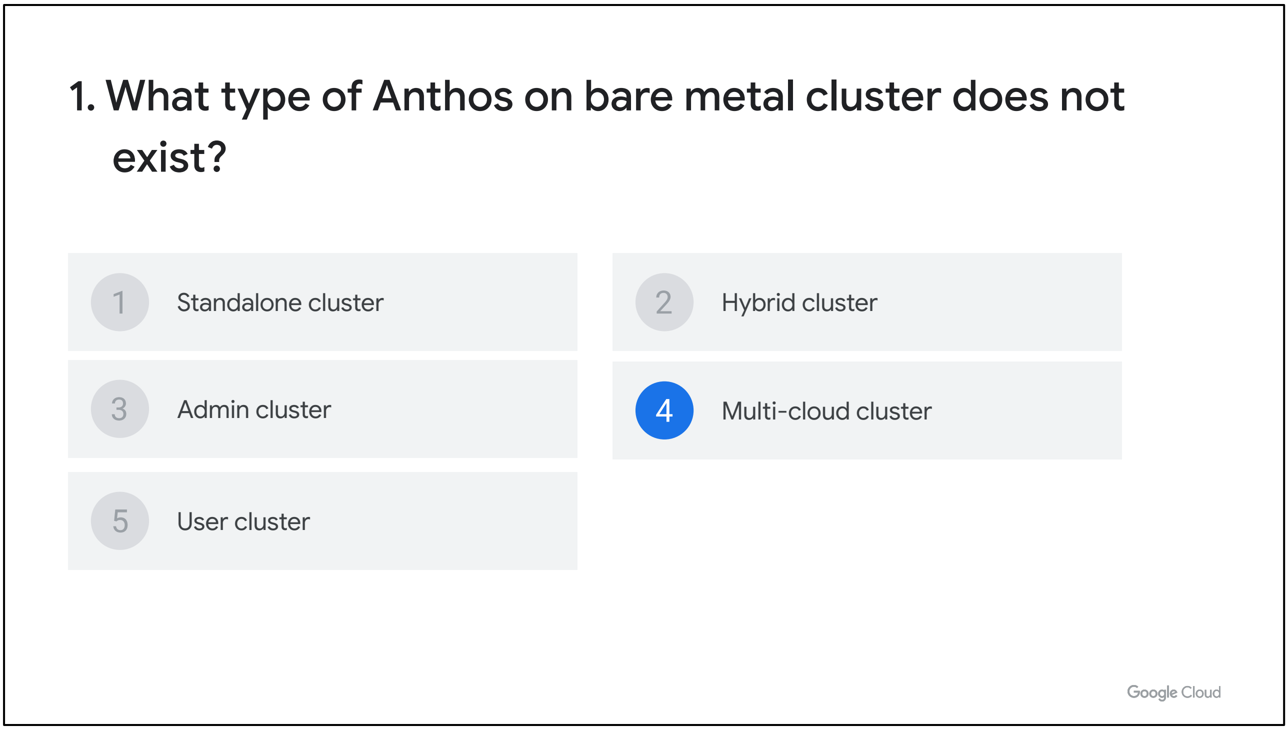
1 Standalone cluster

2 Hybrid cluster

3 Admin cluster

4 Multi-cloud cluster

5 User cluster



1. What type of Anthos on bare metal cluster does not exist?

1 Standalone cluster

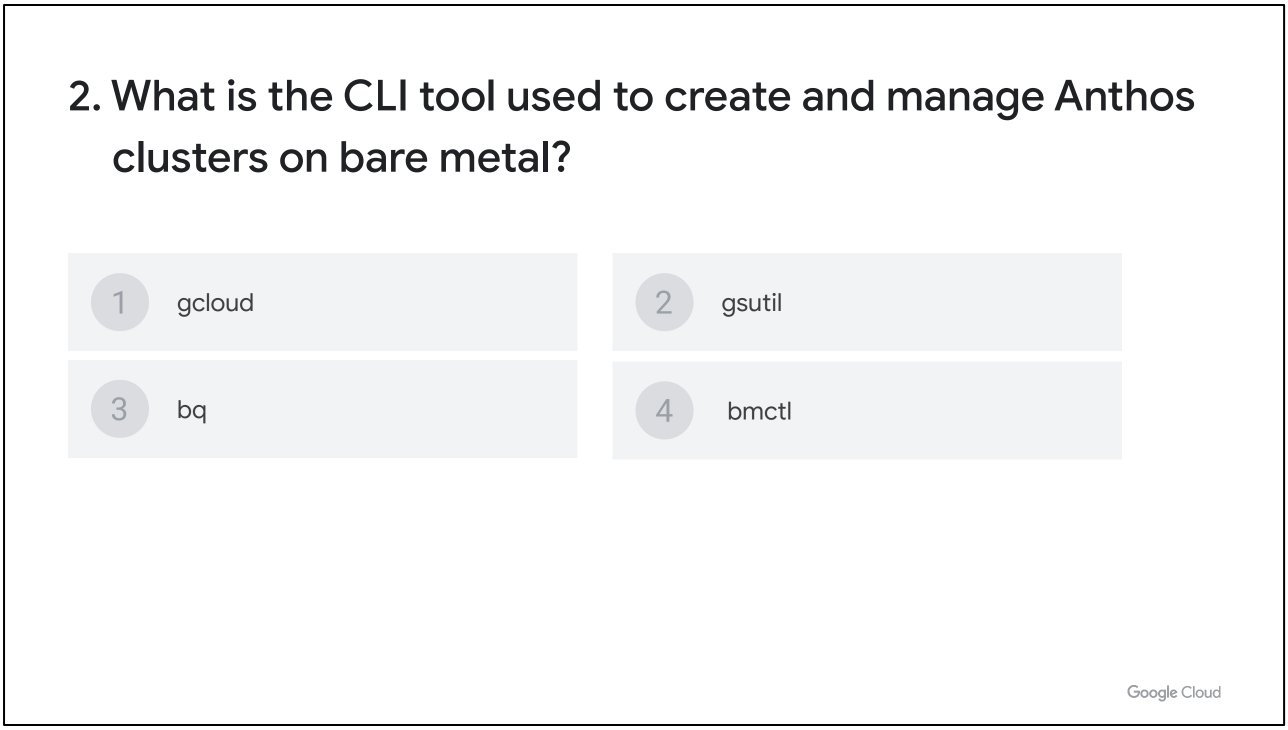
2 Hybrid cluster

3 Admin cluster

[✓ 4 Multi-cloud cluster]

5 User cluster

Google Cloud



2. What is the CLI tool used to create and manage Anthos clusters on bare metal?

1. gcloud
2. gsutil
3. bq
4. bmctl

2. What is the CLI tool used to create and manage Anthos clusters on bare metal?

1 gcloud	2 gsutil
3 bq	[✓ 4 bmctl]

3. Why does the Anthos on bare metal installer create a kind cluster?

1 To reuse the software used to create and configure other clusters.

2 Only to perform pre-flight checks on created clusters.

3 The installer actually uses Minikube.

4 To install the Kubernetes software in the nodes after systemd performs the pre-flight checks.

3. Why does the Anthos on bare metal installer create a kind cluster?

1. To reuse the software used to create and configure other clusters.
2. Only to perform pre-flight checks on created clusters.
3. The installer actually uses Minikube.
4. To install the Kubernetes software in the nodes after systemd performs the pre-flight checks.

Google Cloud