This page shows how to create an admin cluster and a user cluster.

< Previous (/gke-on-prem/docs/how-to/service-accounts)

SSH into your admin workstation:

where ***[IP_ADDRESS]*** is the IP address of your admin workstation.

Do all of the remaining steps in this topic on your admin workstation.

Log in to Google Cloud using your Google Cloud user account credentials. The user account must hold at least the Viewer Cloud IAM role:

Register `gcloud` as a Docker credential helper (https://docs.docker.com/engine/reference/commandline/login/#credential-helpers). (Read more about this command (/sdk/gcloud/reference/auth/configure-docker)):

To specify the static IP addresses that you want to use for your admin cluster, create a host configuration file named `admin-hostconfig.yaml`. For this exercise, you need to specify five IP addresses to be used by the admin cluster.

The following is an example of a host configuration file with five hosts:

The `ips` field is an array of IP addresses and hostnames. These are the IP addresses and hostnames that GKE on-prem will assign to your admin cluster nodes.

In the host configuration file, you also specify the addresses of the DNS servers, time servers, and default gateway that the admin cluster nodes will use.

To specify the static IP addresses that you want to use for your user cluster, create a host configuration file named `user-hostconfig.yaml`.

The following is an example of a host configuration file with three hosts:

The `ips` field is an array of IP addresses and hostnames. These are the IP addresses and hostnames that GKE on-prem will assign to your user cluster nodes.

Copy the following YAML to a file named `config.yaml`.

Modify `config.yaml` as described in the following sections:

The `vcenter.credentials.address` field holds the IP address or the hostname of your vCenter server.

Before you fill in the `vsphere.credentials.address field`, download and inspect the serving certificate of your vCenter server. Enter the following command to download the certificate and save it to a file named `vcenter.pem`.

where ***[VCENTER_IP]*** is the IP address of your vCenter Server.

Open the certificate file to see the Subject Common Name and the Subject Alternative Name:

The output shows the `Subject` Common Name (CN). This might be an IP address, or it might be a hostname. For example:

The output might also include one or more DNS names under `Subject Alternative Name`:

Choose the `Subject` Common Name or one of the DNS names under `Subject Alternative Name` to use as the value of `vcenter.credentials.address` in your configuration file. For example:

You must choose a value that appears in the certificate. For example, if the IP address does not appear in the certificate, you cannot use it for `vcenter.credentials.address`.

GKE on-prem needs to know your vCenter Server's username and password. To provide this information, set the `username` and `password` values under `vcenter.credentials`. For example:

GKE on-prem needs some information about the structure of your vSphere environment. Set the values under `vcenter` to provide this information. For example:

A vSphere resource pool (https://docs.vmware.com/en/VMware-vSphere/6.5/com.vmware.vsphere.resmgmt.doc/GUID-60077B40-66FF-4625-934A-641703ED7601.html) is a logical grouping of vSphere VMs in your vSphere cluster. If you are using a resource pool other than the default, provide its name to `vcenter.resourcepool`. For example:

If you want GKE on-prem to deploy its nodes to the vSphere cluster's default resource pool, provide an empty string to `vcenter.resourcepool`. For example:

GKE on-prem creates a virtual machine disk (VMDK) to hold the Kubernetes object data for the admin cluster. The installer creates the VMDK for you, but you must provide a name for the VMDK in the `vcenter.datadisk` field. For example:

vSAN datastore: Creating a folder for the VMDK

If you are using a vSAN datastore, you need to put the VMDK in a folder. You must manually create the folder ahead of time. To do so, you could use `govc` to create a folder:

Then set `vcenter.datadisk` to the path of the VMDK, including the folder. For example:

★ **Important:** Do not put a forward slash in front of the folder name.

In version 1.1.1, a known issue (/gke-on-prem/docs/release-notes#vsan-data-disk-uuid-issue) requires that you provide the folder's universally unique identifier (UUID) path, rather than its file path, to `vcenter.datadisk`. Copy this from the output of the above `govc` command.

Then, provide the folder's UUID in the `vcenter.datadisk` field. Do not put a forward slash in front of the UUID. For example:

This issue has been fixed in versions 1.1.2 and later.

When a client, like GKE on-prem, sends a request to vCenter Server, the server must prove its identity to the client by presenting a certificate or a certificate bundle. To verify the certificate or bundle, GKE on-prem must have the root certificate in the chain of trust.

Set `vcenter.cacertpath` to the path of the root certificate. For example:

Your VMware installation has a certificate authority (CA) that issues a certificate to your vCenter server. The root certificate in the chain of trust is a self-signed certificate created by VMware.

If you do not want to use the VMWare CA, which is the default, you can configure VMware to use a different certificate authority (https://docs.vmware.com/en/VMware-vSphere/6.7/com.vmware.psc.doc/GUID-3D0DE463-D0EC-442E-B524-64759D063E25.html).

If your vCenter server uses a certificate issued by the default VMware CA, there are several ways you can get the root certificate:

- where ***[SERVER_ADDRESS]*** is the address of your vCenter server.
- In a browser, enter the address of your vCenter server. In the gray box at the right, click **Download trusted root CA certificates**.
- Enter this command to get the serving certificate:

  In the output, find a URL like this: https://[SERVER_ADDRESS]/afd/vecs/ca. Enter the URL in a browser. This downloads the root certificate.

The downloaded file is named `download.zip`.

Install unzip command and unzip the file:

If the unzip command doesn't work the first time, enter the command again.

Find the certificate file in `certs/lin`.

If your network is behind a proxy server, set `proxy.url` to the address of your proxy server.

For `proxy.noproxy`, provide a list of IP addresses, IP address ranges, hostnames, and domain names. When GKE on-prem sends a request to one of these addresses, hosts, or domains, it will send the request directly. It will not send the request to the proxy server. For example:

Because you are using static IP addresses, you must have a host configuration file as described in Configuring static IPs (#configuring_static_ips_admin). Provide the path to your host configuration file in the `admincluster.ipblockfilepath` field. For example:

GKE on-prem needs to know the IP address or hostname, username, and password of your F5 BIG-IP load balancer. Set the values under `admincluster.bigip` to provide this information. For example:

Previously, you created a BIG-IP partition (/solutions/partners/installing-f5-big-ip-adc-for-gke-on-prem#additional_configuration) for your admin cluster. Set `admincluster.bigip.partition` to the name of your partition. For example:

Set the value of `admincluster.vips.controlplanevip` to the IP address that you have chosen to configure on the load balancer (/gke-on-prem/docs/how-to/load-balance-basic#setting_aside_virtual_ip_addressesalancer) for the Kubernetes API server of the admin cluster. Set the value of `ingressvip` to the IP address you have chosen to configure on the load balancer for the admin cluster's ingress service. For example:

The admin cluster must have a range of IP addresses to use for Services and a range of IP addresses to use for Pods. These ranges are specified by the `admincluster.serviceiprange` and `admincluster.podiprange` fields. These fields are populated when you run `gkectl create-config`. If you like, you can change the populated values to values of your choice.

The Service and Pod ranges must not overlap. Also, the Service and Pod ranges must not overlap with IP addresses that are used for nodes in any cluster.

Example:

Previously, you created a BIG-IP partition (/solutions/partners/installing-f5-big-ip-adc-for-gke-on-prem#additional_configuration) for your user cluster. Set `usercluster.bigip.partition` to the name of your partition. For example:

Set the value of `usercluster.vips.controlplanevip` to the IP address that you have chosen to configure on the load balancer (/gke-on-prem/docs/how-to/load-balance-basic#setting_aside_virtual_ip_addresses) for the Kubernetes API server of the user cluster. Set the value of `ingressvip` to the IP address you have chosen to configure on the load balancer for the user cluster's ingress service. For example:

The user cluster must have a range of IP addresses to use for Services and a range of IP addresses to use for Pods. These ranges are specified by the `usercluster.serviceiprange` and `usercluster.podiprange` fields. These fields are populated when you run `gkectl create-config`. If you prefer, you can change the populated values to values of your choice.

The Service and Pod ranges must not overlap. Also, the Service and Pod ranges must not overlap with IP addresses that are used for nodes in any cluster.

Example:

As of version 1.1.0-gke.6, GKE on-prem automatically creates VMware Distributed Resource Scheduler (https://www.vmware.com/products/vsphere/drs-dpm.html) (DRS) anti-affinity rules for your user cluster's nodes, causing them to be spread across at least three physical hosts in your datacenter. As of version 1.1.0-gke.6, this feature is automatically enabled for new clusters and existing clusters.

This feature requires that your vSphere environment meets the following conditions:

- VMware DRS is enabled. VMware DRS requires vSphere Enterprise Plus license edition. To learn how to enable DRS, see Enabling VMware DRS in a cluster (https://kb.vmware.com/s/article/1034280).
- The vSphere user account provided in the `vcenter` field (#vcentercredentials) has the `Host.Inventory.EditCluster` permission.
- There are at least three physical hosts available.

Recall that if you have a vSpphere Standard license (/gke-on-prem/docs/how-to/vsphere-requirements-basic#license-edition-and-version-requirements), you cannot enable VMware DRS.

If you do not have DRS enabled, or if you do not have at least three hosts to which vSphere VMs can be scheduled, add `usercluster.antiaffinitygroups.enabled: false` to your configuration file. For example:

The `gkeconnect` specification holds information that GKE on-prem needs to set up management of your on-prem clusters from Google Cloud Console.

Set `gkeconnect.projectid` to the project ID of the Google Cloud project where you want to manage your on-prem clusters.

Set the value of `gkeconnect.registerserviceaccountkeypath` to the path of the JSON key file for your register service account (/gke-on-prem/docs/how-to/service-accounts#register_service_account). Set the value of `gkeconnect.agentserviceaccountkeypath` to the path of the JSON key file for your connect service account (/gke-on-prem/docs/how-to/service-accounts#connect_service_account).

Example:

In versions prior to 1.1.0-gke.6, the Connect Agent used the proxy server specified in the **`gkeconnect.proxy`** field. g with version 1.1.0-gke.6, the Connect Agent uses the proxy server specified in the global **`proxy`** field. For more ation, see the release notes (/gke-on-prem/docs/release-notes#january_3_2020).

The **`stackdriver.proxyconfigsecretname`** field is removed as of version 1.1.0-gke.6. Don't include it in your uration file.

The `stackdriver` specification holds information that GKE on-prem needs to store log entries generated by your on-prem clusters.

Set `stackdriver.projectid` to the project ID of the Google Cloud project that you want to associate with Stackdriver. Connect exports cluster logs from to Stackdriver by way of this project.

Set `stackdriver.clusterlocation` to a Google Cloud region where you want to store logs. It is a good idea to choose a region that is near your on-prem data center.

Set `stackdriver.proxyconfigsecretname` to a Kubernetes Secret that you define in the `kube-system` namespace. This Secret should have a single value defining `https_proxy_url`. The default Secret

`stackdriver-proxy-config` is immutable and simply serves as an example.

Set `stackdriver.enablevpc` to `true` if you have your cluster's network controlled by a VPC (/vpc/). This ensures that all telemetry flows through Google's restricted IP addresses.

Set `stackdriver.serviceaccountkeypath` to the path of the JSON key file for your Stackdriver service account (/gke-on-prem/docs/how-to/service-accounts#stackdriver_service_account).

Example:

Set the value of `gcrkeypath` to the path of the JSON key file for your whitelisted service account. For example: Note: To learn more about this command, see Running preflight checks (/gke-on-prem/docs/how-to/preflight-checks).

After you've modified the configuration file, run `gkectl check-config` to verify that the file is valid and can be used for installation:

If the command returns any `FAILURE` messages, fix the issues and validate the file again. Use `--fast` flag to skips checks that create test VMs, which depends on the node OS image being uploaded from the admin workstation to vCenter by `gkectl prepare`.

Run `gkectl prepare` to initialize your vSphere environment:

The `gkectl prepare` command imports the node OS image to vSphere and marks it as a VM template.

If you're using internal DNS names, or your workstation cannot access Google public DNS addresses (8.8.8.8 or 8.8.4
ust specify your internal nameserver in **`/etc/resolv.conf`**
s://help.ubuntu.com/lts/serverguide/network-configuration.html.en#name-resolution).

After uploading the node OS image by running `gkectl prepare`, run `gkectl check-config` without the `--fast` flag, so that additional checks that create test VMs can be executed.

If the command returns any `FAILURE` messages, fix the issues and validate the file again.

To learn more about this command, see Running preflight checks (/gke-on-prem/docs/how-to/preflight-checks).

Create the admin cluster and the user cluster:

The `gkectl create cluster` command creates a file named `kubeconfig` in the current directory. The GKE on-prem documentation uses the placeholder ***[ADMIN_CLUSTER_KUBECONFIG]*** to refer to this file.

To verify that the admin cluster was created, enter the following command:

The output shows the admin cluster nodes.

The `gkectl create cluster` command creates a file named `init-user-cluster-kubeconfig` in the current directory. The GKE on-prem documentation uses the placeholder ***[USER_CLUSTER_KUBECONFIG]*** to refer to this file.

To verify that the user cluster was created, enter the following command:

The output shows the user cluster nodes. For example:

< Previous (/gke-on-prem/docs/how-to/service-accounts) Next > (/gke-on-prem/docs/how-to/deploy-first-app)