This page describes how to manually create and restore backups of GKE on-prem admin and user clusters' etcd key-value stores. This page also provides a script (#backup_script) that you can use to automatically back up your clusters' etcd stores.

You should create backups for recovery from foreseen disasters that might damage etcd data and Secrets. Be sure to store backups in a location that is outside of the cluster and that is not dependent on the cluster's operation. If you want to be safe, consider creating a copy of the backup, too.

While the etcd events Pod that runs in every cluster is not vital to the restoration of a user cluster, you can follow a similar process to back it up. Also note this procedure allows you to back up only the etcd stores; the PersistentVolumes are not backed up as part of this guide, and you should plan for additional backup and restore procedure for those.

- Backing up application-specific data is out of scope for this feature.
- Secrets remain valid until you manually rotate them.
- Workloads scheduled after you create a backup aren't restored with that backup.
- Currently, you aren't able to restore from failed cluster upgrades.

When you run `sudo` commands, you might encounter the following error:

If you do, add the following line to the `/etc/hosts` file:

A user cluster backup contains a snapshot of the user cluster's etcd. A cluster's etcd contains, among other things, all of the Kubernetes objects and any custom objects required to manage cluster state. This snapshot contains the data required to recreate the cluster's components and workloads.

A user cluster's etcd is stored in its control plane node, which you can access using the admin cluster's kubeconfig.

To create a snapshot of etcd, execute the following steps:

1. Shell into the kube-etcd container:

   where:

   - ***[ADMIN_CLUSTER_KUBECONFIG]*** is the admin cluster's kubeconfig file.
   - ***[USER_CLUSTER_NAME]*** is name of the user cluster. Specifically, you're passing in a namespace in the admin cluster that is named after the user cluster.

2. From the shell, use `etcdctl` to a create backup named `snapshot.db` in the local directory:

3. Exit the container:

4. Copy the backup out of the kube-etcd container using `kubectl cp`:

where ***[RELATIVE_DIRECTORY]*** is a path where you want to store your backup.

- Before you restore a backup, be sure to diagnose your cluster (/gke-on-prem/docs/support/diagnose) and resolve existing issues. Restoring a backup to a problematic cluster might recreate or exacerbate issues. Contact the GKE on-prem support team for further assistance on restoring your clusters.
- If you created a HA user cluster (/gke-on-prem/docs/concepts/ha), you should run these steps once per etcd cluster member. You can use the same snapshot when restoring each etcd member. Don't take these steps unless all etcd Pods are crashlooping: this indicates that there is data corruption.

The following instructions explain how to restore a backup in cases where a user cluster's etcd data has become damaged and its etcd Pod is crashlooping. You can recover by deploying a etcd Pod to the existing Pod's volumes and overwriting the damaged data with the backup, assuming that the user cluster's API server is running and can schedule new Pods.

1. Copy the etcd Pod specification below to a file, `restore-etcd.yaml`, after populating the following placeholder values:
    - ***[MEMBER_NUMBER]*** is the numbered Pod that you are restoring.
    - ***[NODE_NAME]*** is the node on which the ***[MEMBER_NUMBER[*** Pod is running.
    - ***[ADMIN_CLUSTER_KUBECONFIG]*** is the admin cluster's kubeconfig file.
    - ***[USER_CLUSTER_NAME]*** is the name of the user cluster.
    - ***[DEFAULT_TOKEN]*** is used for authentication. You can find this value by running the following command:

★ **Note:** The **`image`** value below assumes you are using the gcr.io registry. Change the image name if you are using a private registry.

2. Deploy the Pod:

3. Copy etcd's backup file, `snapshot.db`, to the new Pod. `snapshot.db` lives at the relative directory where you created the backup:

4. Shell into the `restore-etcd` Pod:

5. Run the following command to create a new default.etcd folder containing the backup:

6. Overwrite the damaged etcd data with the backup:

7. Exit the container:

8. Delete the crashing etcd Pod:

9. Verify that the etcd Pod is no longer crashing.
10. Remove `restore-etcd.yaml` and delete the `restore-etcd` Pod:

An admin cluster backup contains the following:

- A snapshot of the admin cluster's etcd.
- Admin control plane's Secrets, which are required for authenticating to the admin and user clusters.

Complete the following steps before you create an admin cluster backup:

1. Find the admin cluster's external IP address, which is used to SSH in to the admin cluster control plane:

   where ***[ADMIN_CLUSTER_KUBECONFIG]*** is the admin cluster's kubeconfig file.

2. Create an SSH key called `vsphere_tmp` from the admin cluster's private key.

   You can find the private key from the admin clusters Secrets:

   In the command output, you can find the private key in the `vsphere_tmp` field.

   Copy the private key to `vsphere_tmp`:

3. Check that you can shell into the admin control plane using this private key:

4. Exit the container:

You can back up an admin cluster's etcd and its control plane's Secrets.

The following procedure recreates a backed-up admin cluster and all of the user control planes it managed when its etcd snapshot was created.

The user cluster can still run its workloads even if the user control plane is down. However, if a user control plane is d chedule new or updated workloads or handle other cluster failures.

1. Run `scp` to copy `snapshot.db` to the admin control plane:

   where ***[EXTERNAL_IP]*** is the admin control plane's external IP address, which you gathered previously.

2. Shell into the admin control plane:

3. Copy `snapshot.db/` to `/mnt`:

4. Make temporary directory, like `backup`:

★ **Note:** You don't need to do this if there is already a backup folder containing the Secrets.

5. Exit the admin control plane:

6. Copy the certificates to `backup/`:

7. Shell into the admin control plane node:

   where ***[EXTERNAL_IP]*** is the admin control plane's external IP address, which you gathered previously.

8. Run `kubeadm reset`. This stops anything still running in the admin cluster, deletes all etcd data, and deletes Secrets in `/etc/kubernetes/pki/`:

9. Copy the backup Secrets to `/etc/kubernetes/pki/`:

10. Run `etcdctl restore` with Docker:

11. Run `kubeadm init`. This reuses all of the backup Secrets and restarts etcd with the restored snapshot:

12. Exit the admin control plane:

13. Copy the newly generated kubeconfig file out of the admin node:

    where:

    - ***[EXTERNAL_IP]*** is the admin control plane's external IP address.
    - ***[HOME]*** is the home directory on the admin node.

    Now you can use this new kubeconfig file to access restored cluster.

You can use the script given here to automatically back up your clusters. Before you run the script, fill in values for the five variables at the beginning of the script:

- Set `BACKUP_DIR` to the path where you want to store the admin and user cluster backups.
- Set `ADMIN_CLUSTER_KUBECONFIG` to the path of the admin cluster's kubeconfig file
- Set `USER_CLUSTER_NAMESPACE` to the name of your user cluster. The name of your user cluster is a namespace in the admin cluster.
- Set `EXTERNAL_IP` to the VIP that you reserved for the admin control plane service (/gke-on-prem/docs/how-to/requirements#vips).
- Set `SSH_PRIVATE_KEY` to the path of the SSH key you created (/gke-on-prem/docs/how-to/admin-workstation#install) when you set up your admin workstation.
- If you are using a private network, set `JUMP_IP` to your network's jump server's IP address.

For more information, refer to Troubleshooting (/gke-on-prem/docs/resources/troubleshooting).

Use `gkectl diagnose` commands to identify cluster issues and share cluster information with Google. See Diagnosing cluster issues (/gke-on-prem/docs/diagnose).

Even if you don't pass in its debugging flags, you can view `gkectl` logs in the following admin workstation directory:

If a VM fails to start after the admin control plane has started, you can try debugging this by inspecting the Cluster API controllers' logs in the admin cluster:

1. Find the name of the Cluster API controllers Pod in the `kube-system` namespace, where ***[ADMIN_CLUSTER_KUBECONFIG]*** is the path to the admin cluster's kubeconfig file:

2. Open the Pod's logs, where ***[POD_NAME]*** is the name of the Pod. Optionally, use `grep` or a similar tool to search for errors:

- Learn how to diagnose cluster issues (/gke-on-prem/docs/support/diagnose)
- Learn about augur (https://github.com/jpbetz/auger), an open-source tool for restoring individual objects from etcd backups.