# **Velero** **Backup and Recovery for Kubernetes**

## **Introduction to Velero**

Velero is an open-source tool that helps you safely back up, restore, and migrate Kubernetes cluster resources and persistent volumes. It can be used to recover from disasters or to migrate cluster resources between environments.

Key features:

* Backup and restore entire Kubernetes clusters, namespaces, or specific resources.
* Supports multiple cloud platforms (Google Cloud, AWS, Azure, etc.).
* Migrate resources between clusters.
* Schedule automatic backups.

## **Velero Architecture**

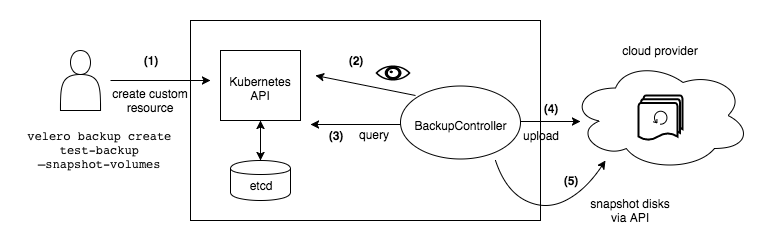
Velero operates with two key components:

1. **Server-side (Velero Server)**: A deployment in your Kubernetes cluster that performs backup and restore operations.
2. **Client-side (Velero CLI)**: A command-line interface to interact with the Velero server.

Both components are required to manage backups, schedule operations, and restore Kubernetes cluster resources.

### **Simple Overview of Velero Backup Workflow**

1. **Create Backup**: When you run velero backup create test-backup, the Velero client asks the Kubernetes API to create a new backup.
2. **Validation**: The BackupController checks the new backup request to ensure it’s valid.
3. **Data Collection**: The BackupController starts gathering data by querying the API for resources that need to be backed up.
4. **Upload Backup**: The BackupController sends the backup file to an object storage service (like AWS S3) for safekeeping.
5. **Snapshots**: By default, Velero takes disk snapshots of any persistent volumes. You can customize this by using extra flags. To disable snapshots, use --snapshot-volumes=false. Run velero backup create --help for more options.



## **Velero Use Cases**

* Disaster recovery: Restore cluster resources in case of failures.
* Cluster migration: Move workloads between different clusters or cloud environments.
* Backup scheduling: Regular backups to ensure data protection.
* Backup verification: Verify backup integrity by periodically restoring.

## **Velero Setup and Installation**

### **1 Prerequisites**

* A Kubernetes cluster (versions supported by Velero).
* Access to a cloud provider (e.g., Google Cloud, AWS, Azure).
* A storage location for backups (object storage such as Google Cloud Storage, S3).

### **2 Installation Steps**

#### **Install Velero CLI**

To install Velero, download the latest release from GitHub, extract it, and move it to your system path.

|  |
| --- |
| wget https://github.com/vmware-tanzu/velero/releases/download/v1.10.1/velero-v1.10.1-linux-amd64.tar.gz  tar -xvf velero-v1.10.1-linux-amd64.tar.gz  sudo mv velero-v1.10.1-linux-amd64/velero /usr/local/bin/ |

To verify the installation:

|  |
| --- |
| velero version |

#### **3. Create a Google Cloud Storage Bucket for Backups (GCP)**

For GCP:

|  |
| --- |
| gsutil mb -l us-central1 gs://<YOUR-BUCKET-NAME> |

Enable versioning for the bucket:

|  |
| --- |
| gsutil versioning set on gs://<YOUR-BUCKET-NAME> |

#### **4. Configure Google Cloud IAM Permissions**

Velero will need a service account with specific permissions to read and write to the Google Cloud Storage bucket.

Create a service account:

|  |
| --- |
| gcloud iam service-accounts create velero \  --display-name "Velero service account" |

Assign the necessary permissions:

|  |
| --- |
| gcloud projects add-iam-policy-binding <YOUR-PROJECT-ID> \  --member serviceAccount:velero@<YOUR-PROJECT-ID>.iam.gserviceaccount.com \  --role roles/storage.admin |

Download the service account key:

|  |
| --- |
| gcloud iam service-accounts keys create credentials-velero.json \  --iam-account velero@<YOUR-PROJECT-ID>.iam.gserviceaccount.com |

#### **5. Install Velero Server on Kubernetes**

Install Velero server in your Kubernetes cluster using the velero install command:

|  |
| --- |
| velero install \  --provider gcp \  --plugins velero/velero-plugin-for-gcp:v1.5.0 \  --bucket <YOUR-BUCKET-NAME> \  --secret-file ./credentials-velero.json \  --backup-location-config serviceAccount=velero@<YOUR-PROJECTID> .iam.gserviceaccount.com \  --namespace velero \  --snapshot-location-config region=us-central1 |

This command installs the Velero server and configures it to back up data to the specified GCS bucket.

The --plugins flag is required for specifying the Velero plugins needed for the backup and restore operations. For GCP, you should include the velero/velero-plugin-for-gcp plugin.

## **Performing Backups with Velero**

### **Manual Backup**

To back up a Kubernetes namespace, use the following command:

|  |
| --- |
| velero backup create <backup-name> --include-namespaces <namespace-name> |

Example:

|  |
| --- |
| velero backup create nginx-backup --include-namespaces nginx |

You can check the status of the backup with:

|  |
| --- |
| velero backup describe <backup-name> --details |

### **Scheduled Backups**

Velero allows you to schedule recurring backups. For example, to back up the namespace nginx every day:

|  |
| --- |
| velero create schedule nginx-schedule --schedule="0 1 \* \* \*" --include-namespaces nginx |

This command will back up the namespace nginx daily at 1 AM.

Example:

Schedule a Backup Every 5 Minutes Using Velero

|  |
| --- |
| velero create schedule my-schedule \  --schedule "\*/5 \* \* \* \*" \  --include-namespaces default \  --ttl 15m |

To schedule a backup every 5 minutes using Velero, here's what's happening step by step:

**Command**:

* + velero create schedule my-schedule: This creates a new backup schedule called my-schedule.

**Schedule**:

* + --schedule "\*/5 \* \* \* \*": This means the backup will run every 5 minutes. It's like telling Velero to check and run the backup every 5th minute, continuously.
  + **Summary of \*/5 \* \* \* \*:**

**\*/5 (Minutes)**: Run the task every 5 minutes.

**\* (Hours)**: Every hour of the day.

**\* (Day of the Month)**: Every day of the month.

**\* (Month)**: Every month of the year.

**\* (Day of the Week)**: Every day of the week (Monday to Sunday).

**Namespace**:

* + --include-namespaces default: This specifies that only the default namespace (a specific part of the cluster) will be backed up.

**TTL (Time to Live)**:

* + --ttl 15m: This tells Velero to keep each backup for 15 minutes before deleting it automatically.

## **Restoring Backups**

Velero makes it easy to restore resources from a backup.

|  |
| --- |
| velero restore create --from-backup <backup-name> |

For example:

|  |
| --- |
| velero restore create --from-backup nginx-backup |

### **Restoring Specific Resources**

If you only want to restore specific resources (e.g., deployments, services), you can specify them with --include-resources:

|  |
| --- |
| velero restore create --from-backup nginx-backup --include-resources deployments,services |

## **Velero Plugin System**

Velero supports a plugin architecture that enables you to interact with different cloud providers. Available plugins include:

* **velero/velero-plugin-for-gcp** (for Google Cloud Storage)
* **velero/velero-plugin-for-aws** (for Amazon S3)
* **velero/velero-plugin-for-microsoft-azure** (for Azure Blob Storage)

These plugins are installed using the --plugins flag during the installation proces

## **Migrating Workloads Using Velero**

Velero is often used for cluster migration. By backing up the resources in one cluster and restoring them in another, you can move workloads between clusters or cloud providers.

Steps for migration:

1. **Backup resources** from the source cluster.
2. **Install Velero** in the destination cluster.
3. **Restore backups** to the destination cluster.

## **Conclusion**

Velero is a powerful and flexible tool for managing backups, restores, and migrations in Kubernetes. By integrating seamlessly with major cloud providers, it simplifies disaster recovery strategies. Its ease of use, support for scheduling, and multi-cluster support make it an essential tool for managing Kubernetes environments.

**Velero Demo on GKE**

### **Prerequisites**

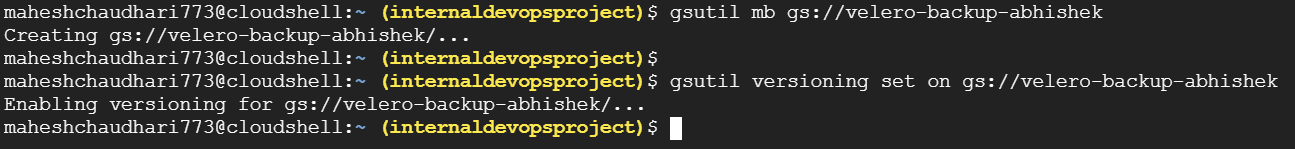
* A Kubernetes cluster (versions supported by Velero).
* Access to a cloud provider (e.g., Google Cloud).
* A storage location for backups (object storage such as Google Cloud Storage, S3).

**Setup**

# Step 1: Create the GCS Bucket and Enable Versioning

gsutil mb gs://velero-backup-abhishek

gsutil versioning set on gs://velero-backup-abhishek



# Step 2: Download and Install Velero CLI

wget https://github.com/vmware-tanzu/velero/releases/download/v1.10.1/velero-v1.10.1-linux-amd64.tar.gz

tar -xvf velero-v1.10.1-linux-amd64.tar.gz

cd velero-v1.10.1-linux-amd64/

sudo mv velero /usr/local/bin/

# Step 3: Install Velero on the GKE Cluster

velero install \

--provider gcp \

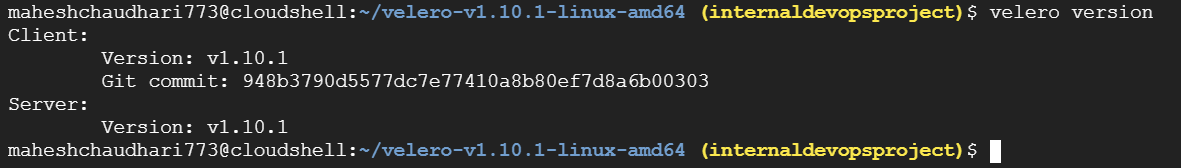
--bucket velero-backup-abhishek \

--backup-location-config [serviceAccount=internaldevopsproject@internaldevopsproject.iam.gserviceaccount.com](mailto:serviceAccount=internaldevopsproject@internaldevopsproject.iam.gserviceaccount.com) \

--secret-file /home/maheshchaudhari773/internaldevopsproject-a2b357447f36.json \

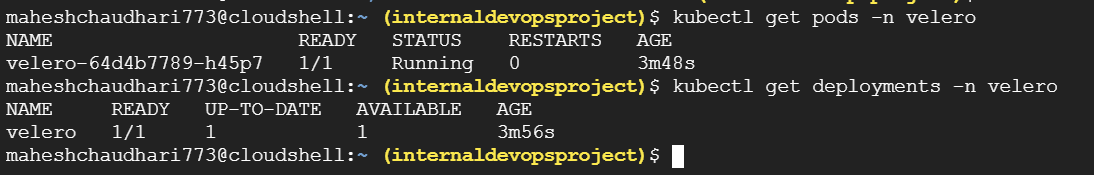
--namespace velero \

--plugins velero/velero-plugin-for-gcp



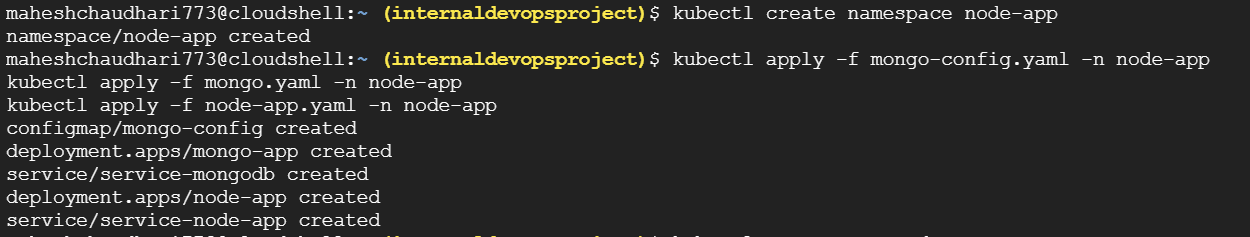
# Step 4: Verify Velero Installation

kubectl get pods -n velero  
kubectl get deployments -n velero



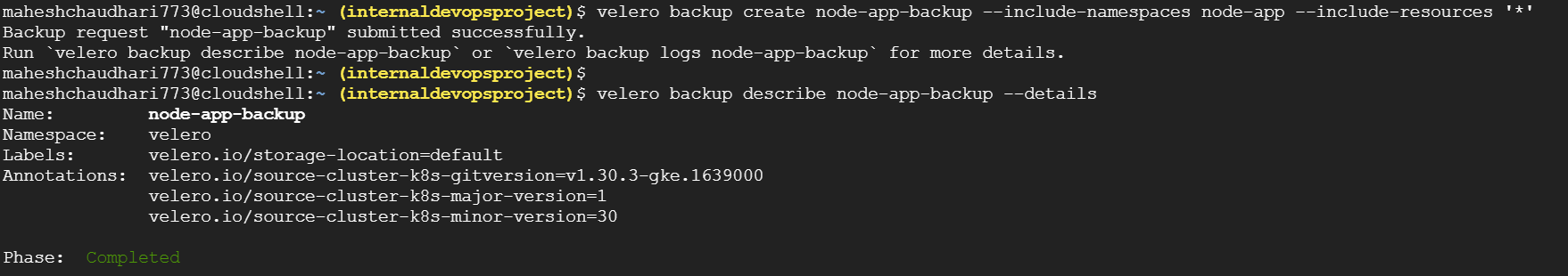
# Step 5: Create a namespace for the application

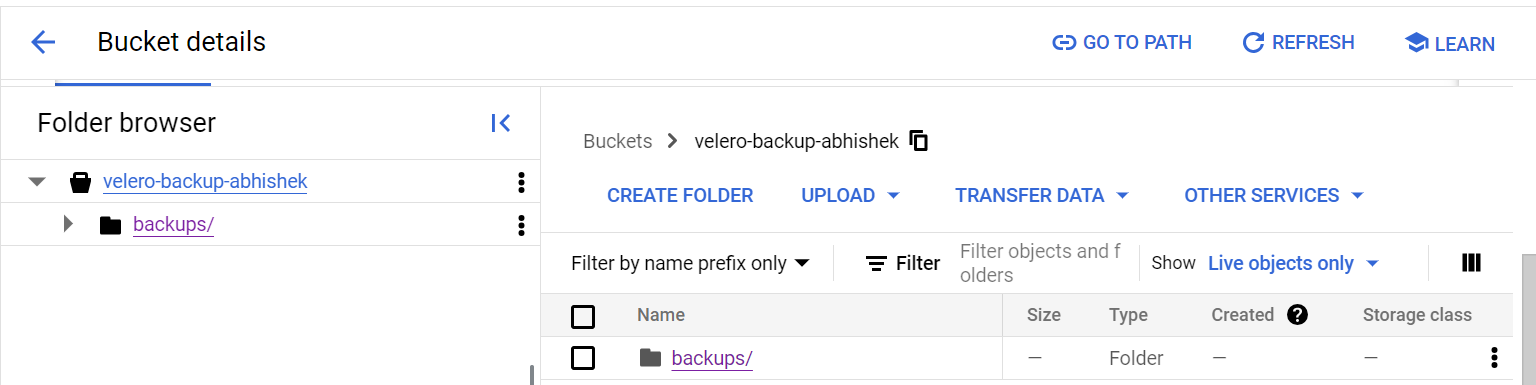
kubectl create namespace node-app  
kubectl apply -f mongo-config.yaml -n node-app  
kubectl apply -f mongo.yaml -n node-app  
kubectl apply -f node-app.yaml -n node-app



# Step 6: Create Backup of Node.js Application in the 'node-app' Namespace

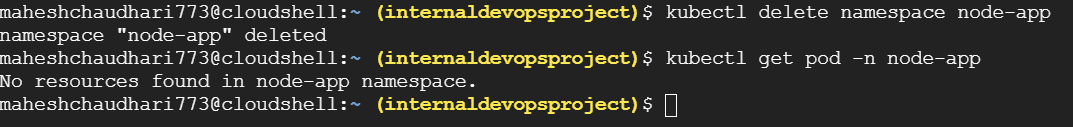
velero backup create node-app-backup --include-namespaces node-app   
--include-resources '\*'  
velero backup describe node-app-backup --details





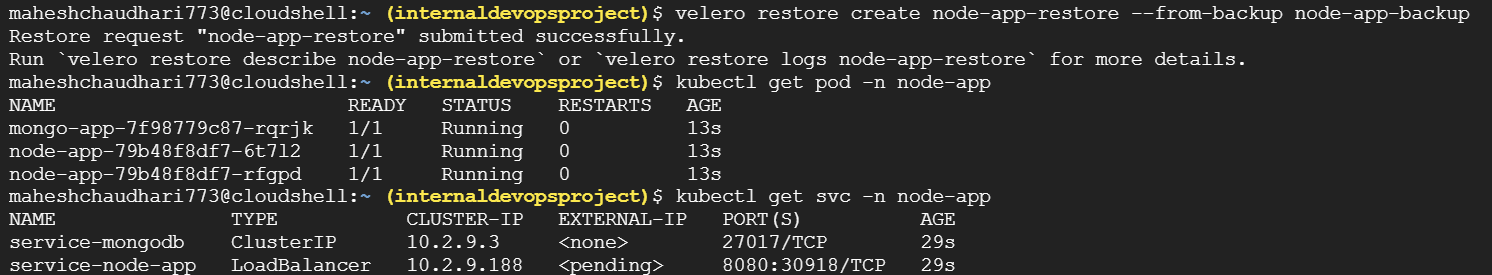
# Step 7: Delete the node-app Namespace (to test restoration)

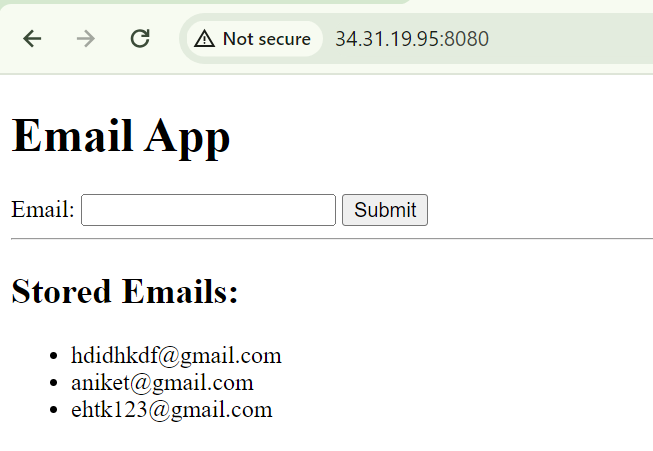
kubectl delete namespace node-app



# Step 8: Restore from Backup

velero restore create node-app-restore --from-backup node-app-backup





# Step 12: Schedule Backups Every 5 Minutes

velero create schedule my-schedule \

--schedule "\*/5 \* \* \* \*" \

--include-namespaces node-app \

--ttl 15m

