Backup and Restore EKS Kubernetes Using Velero.

Velero is an open-source tool that helps automate the backup and restore of Kubernetes clusters, including any application and its data.
Velero lets you backup your entire cluster or namespace(s) or filter objects by using labels. Velero helps with migrating your on-prem Kubernetes workloads to the cloud, cluster upgrades, and disaster recovery.

Velero enables the following use cases:

- Disaster recovery — backup of the cluster and restore in case of a disaster.
- Application migration — migrate an application along with its data from one cluster to another.
- Application cloning — replicating production environments for testing and debugging.

In this module, you will learn the below things:

1.how to backup and restore an EKS cluster using Velero.

2. how to backup and restore data on 2 different EKS Cluster.
3. how to backup and restore the data at a particular time(Scheduled backup and restore).

- **Prerequisites:**
  1. AWS CLI needs to be configured in the machine where you execute Velero commands.
  2. Kubectl needs to be configured with the EKS cluster where you need to take the backup.

1. CREATE S3 BUCKET AND IAM USER FOR VELERO

1. Create the S3 bucket using AWS console.
    go to AWS -> S3-> create bucket

2. Create an IAM user.
go to AWS -> IAM Console -> add user

3. Add the below permission to the user.
replace ${BUCKET} with S3 bucket name which we created for velero.

4. Add user credentials to the server.
using below comment

2. INSTALL VELERO Client

Install Velero binary
wget
https://github.com/vmware-tanzu/velero/releases/download/v1.3.2/velero-v1.3.2-linux-amd64.tar.gz

Extract the tarball:
tar -xvf velero-v1.3.2-linux-amd64.tar.gz -C /tmp

Move the extracted velero binary to /usr/local/bin
sudo mv /tmp/velero-v1.3.2-linux-amd64/velero /usr/local/bin

Verify installation
velero version

if you got the velero not found message then set a below path variable for velero.
 export PATH=$PATH:/usr/local/bin

3. Install Velero on EKS

```
velero install \
    --provider aws \
    --plugins velero/velero-plugin-for-aws:v1.0.1 \
```

```
    --bucket <bucketname>\
    --backup-location-config region=<region> \
    --snapshot-location-config region=<region> \
    --secret-file /root/.aws/credentials
```

replace your bucket name, region, and credentials path in the command.

```
velero install \
    --provider aws \
    --plugins velero/velero-plugin-for-aws:v1.0.1 \
    --bucket velero-backup-ahmed \
    --backup-location-config region=us-east-1 \
    --snapshot-location-config region=us-east-1 \
    --secret-file /root/.aws/credentials
```

Inspect the resources created
kubectl get all -n velero

4. DEPLOY TEST APPLICATION

Create namespace and deploy the application

kubectl create namespace <namespacename>
kubectl create namespace ahmed

Deploy 2 sample applications in the ahmed namespace.

kubectl create deployment web --image=gcr.io/google-samples/hello-app:1.0 -n ahmed
kubectl create deployment nginx --image=nginx -n ahmed

Verify deployment
kubectl get deployments -n ahmed

5. BACKUP AND RESTORE

- Let's back up the ahmed namespace using velero backup:
velero backup create <backupname> --include-namespaces <namespacename>
velero backup create test1 --include-namespaces ahmed

- Check the status of backup
velero backup describe <backupname>

- Check-in S3 bucket :
backup is stored in the S3 bucket.

- Let’s delete the ‘ahmed’ namespace to simulate a disaster
kubectl delete namespace ahmed

Restore ahmed namespace

- restore:

Run the velero restore command from the backup created. It may take a couple of minutes to restore the namespace.

velero restore create --from-backup <backupname>
velero restore create --from-backup test1

Verify if deployments, replica sets, services, and pods are restored.
You will see, ahmed namespace is newly created and restored using velero, and all old deployment, pods, svc also restored back.

Here are some useful commands for velero :

- **Backup:**

Create a backup every 6 hours with the @every notation
velero schedule create <SCHEDULE_NAME> --schedule="@every 6h"

Create a daily backup of the namespace
velero schedule create <SCHEDULE_NAME> --schedule="@every 24h"
--include-namespaces <namspacename>

Create a weekly backup, each living for 90 days (2160 hours)
velero schedule create <SCHEDULE_NAME> --schedule="@every 168h" --ttl 2160h0m0s
##default TTL time is 720h

Create a backup including the test and default namespaces
velero backup create backup --include-namespaces test,default

Create a backup excluding the kube-system and default namespaces
velero backup create backup --exclude-namespaces kube-system,default

To backup entire cluster
velero backup create <BACKUPNAME>

#To backup namespace in a cluster
velero backup create <BACKUPNAME> --include-namespaces <NAMESPACENAME>

- **Restore:**

#Manual Restore
velero restore create --from-backup <backupname>

#Scheduled Backup
velero restore create <RESTORE_NAME> --from-schedule <SCHEDULE_NAME>

Create a restore including the test and default namespaces
velero restore create --from-backup backup --include-namespaces nginx,default

Create a restore excluding the kube-system and default namespaces
velero restore create --from-backup backup --exclude-namespaces kube-system,default

#Retrieve restore logs
velero restore logs <RESTORE_NAME>