**Backup and Restore Kubernetes cluster using Velero**

Velero (formerly Heptio Ark) gives you tools to back up and restore your Kubernetes cluster resources and persistent volumes. You can run Velero with a public cloud platform or on-premises. Velero lets you:

Take backups of your cluster and restore in case of loss.

Migrate cluster resources to other clusters.

Replicate your production cluster to development and testing clusters.

Velero consists of:

A server that runs on your cluster

A command-line client that runs locally

**How Velero Works?**

Each Velero operation -- on-demand backup, scheduled backup, restore -- is a custom resource, defined with a Kubernetes Custom Resource Definition (CRD) and stored in etcd. Velero also includes controllers that process the custom resources to perform backups, restores, and all related operations.

You can back up or restore all objects in your cluster, or you can filter objects by type, namespace, and/or label.

Velero is ideal for the disaster recovery use case, as well as for snapshotting your application state, prior to performing system operations on your cluster (e.g. upgrades).

The following example setup the velero server and back up and restore a sample application.

For simplicity, the example uses **Minio**, an S3-compatible storage service that runs locally on your cluster.

**Pre-Requisites:**

* Kubernetes cluster >=1.10
* Kubectl
* Dns server installed on server.

Download Velero

1. Download the latest official release's tarball for your client platform, Use the following command to download velero

wget <https://github.com/vmware-tanzu/velero/releases/download/v1.4.0/velero-v1.4.0-linux-amd64.tar.gz>

1. This command downloads velero tar and extract tar using the following command

tar -xvf velero-v1.4.0-linux-amd64.tar.gz

1. cd to velero directory

cd velero-v1.4.0-linux-amd64/

1. Move velero file into /usr/local/bin

mv velero /usr/local/bin

1. Verify velero cli should be installed

velero version

**Output:**

Client:

Version: v1.4.0

Git commit: 5963650c9d64643daaf510ef93ac4a36b6483392

<error getting server version: namespaces "velero" not found

1. Install Minio storage for backups , Run the following docker commands to install Minio storage

docker pull minio/minio

docker run --name minio -p 9000:9000 -v data:/data minio/minio server /data

**output:**

Endpoint: http://172.17.0.2:9000 http://127.0.0.1:9000

Browser Access:

http://172.17.0.2:9000 http://127.0.0.1:9000

Object API (Amazon S3 compatible):

Go: https://docs.min.io/docs/golang-client-quickstart-guide

Java: https://docs.min.io/docs/java-client-quickstart-guide

Python: https://docs.min.io/docs/python-client-quickstart-guide

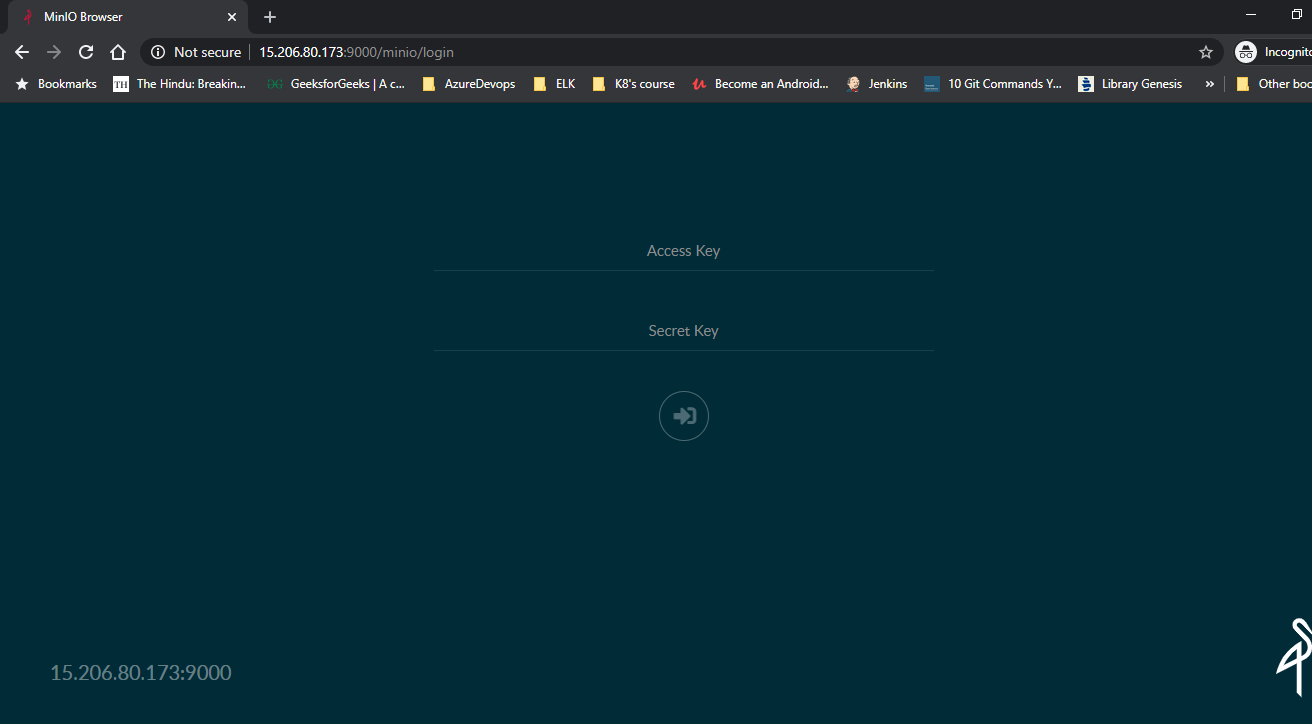
JavaScript: https://docs.min.io/docs/javascript-client-quickstart-guide

.NET: https://docs.min.io/docs/dotnet-client-quickstart-guide

Detected default credentials 'minioadmin:minioadmin', please change the credentials immediately using 'MINIO\_ACCESS\_KEY' and 'MINIO\_SECRET\_KEY'

**Access the Minio cloud Storage using browser**

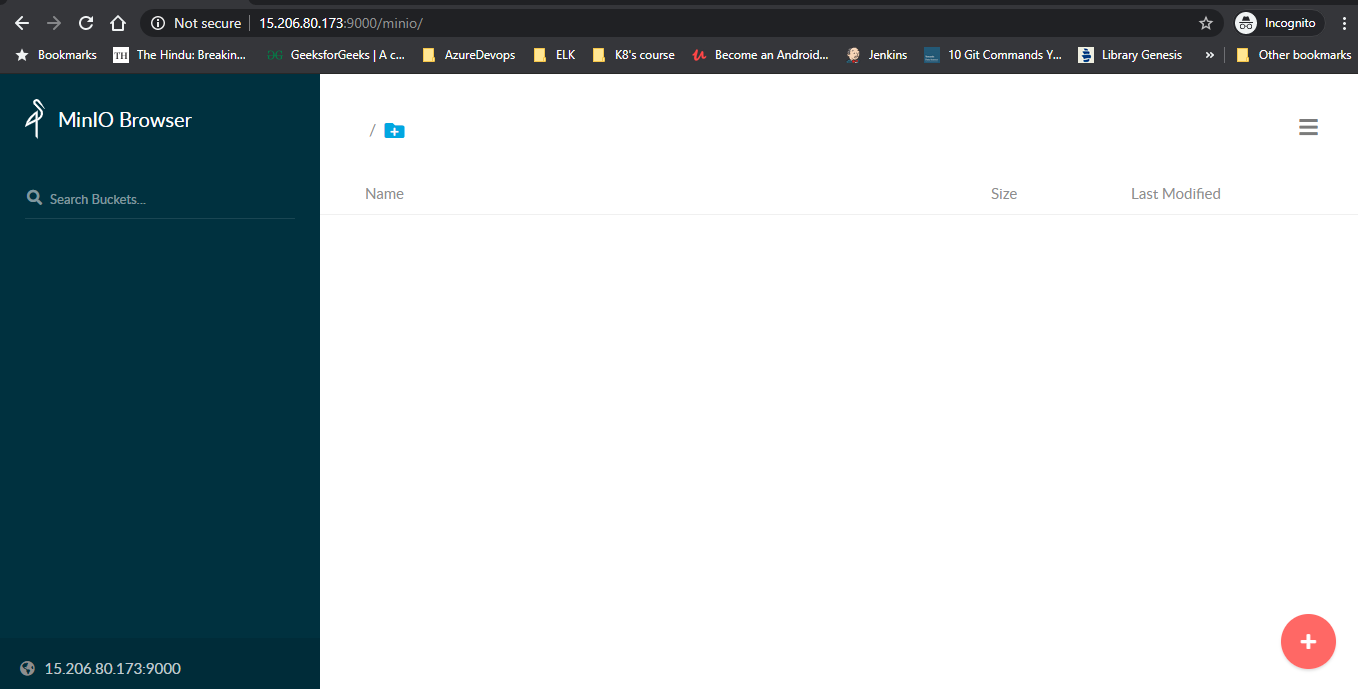
**http://<public-ip>:9000**



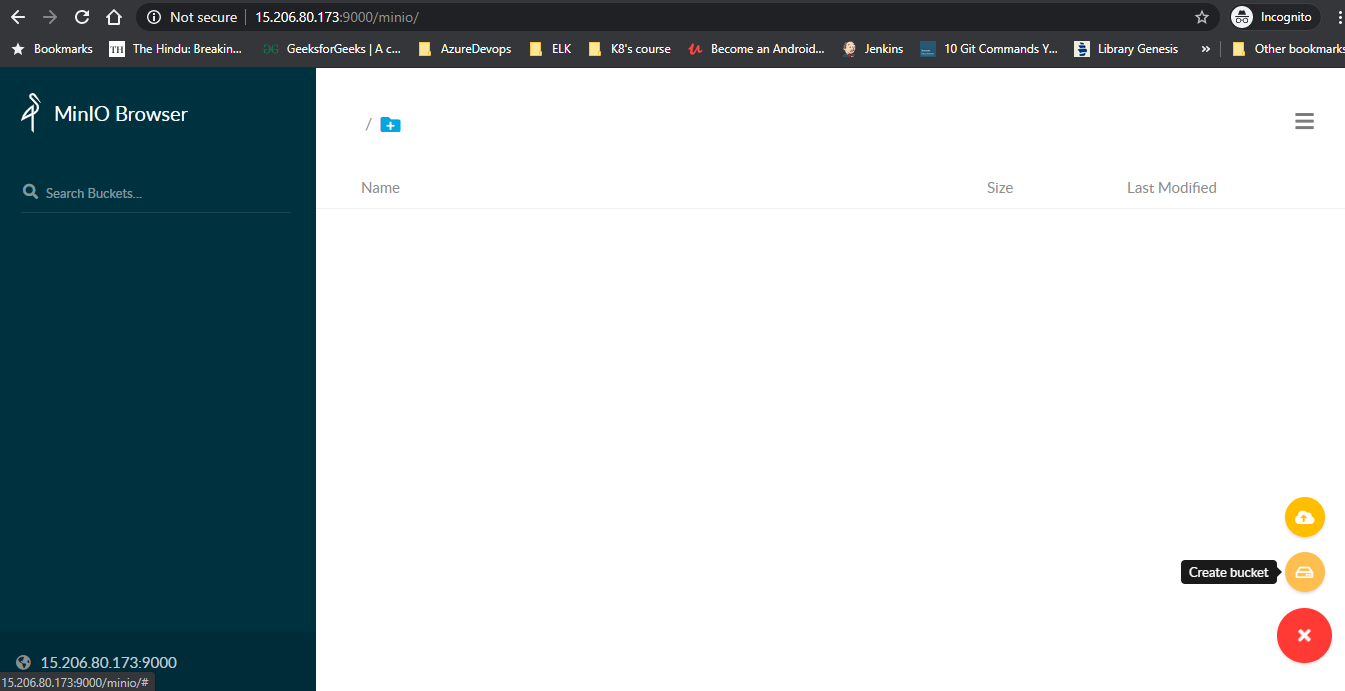
Enter **Access\_Key:** minioadmin

**Secret\_Key:** minioadmin

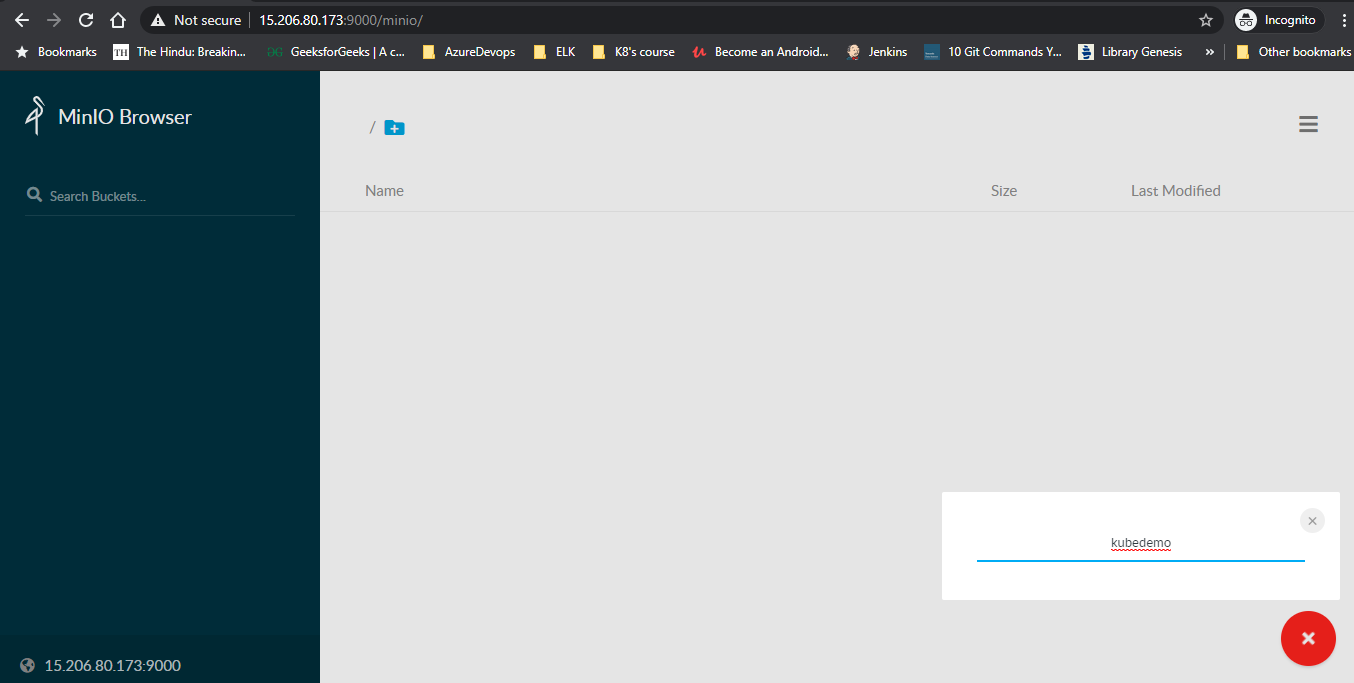
**Once you enter the details you will get the following like this**



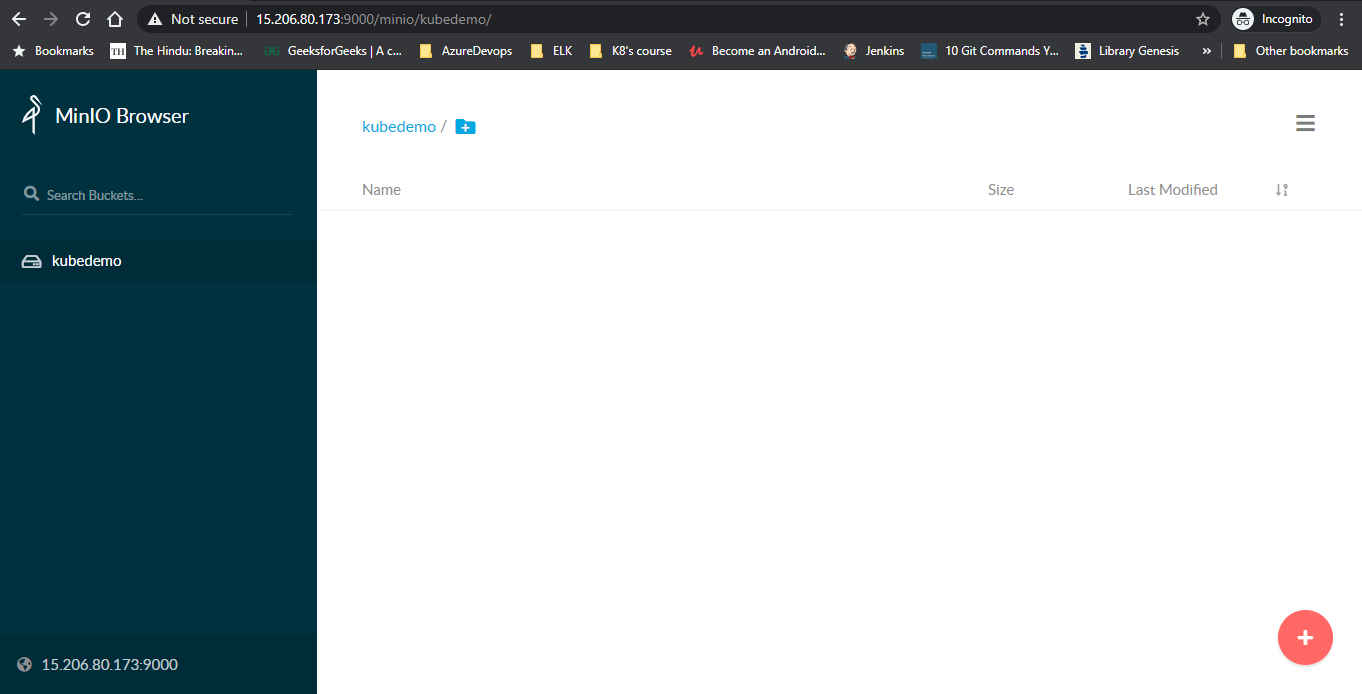
Click **+** and choose create a bucket



* Provide the name of the bucket e.g. Kubedemo and Press Enter



* Bucket should be created as follows



**Create Credentials for Velero Initiliazation**

cat <<EOF>> minio.credentials

[default]

aws\_access\_key\_id=minioadmin

aws\_secret\_access\_key=minioadmin

EOF

**Install Velero in the Kubernetes cluster**

Use the following command

velero install \

--provider aws \

--plugins velero/velero-plugin-for-aws:v1.0.0 \

--bucket kubedemo \

--secret-file ./minio.credentials \

--backup-location-config region=minio,s3ForcePathStyle=true,s3Url=http://15.206.80.173:9000

**Output:**

CustomResourceDefinition/backups.velero.io: attempting to create resource

CustomResourceDefinition/backups.velero.io: already exists, proceeding

CustomResourceDefinition/backups.velero.io: created

CustomResourceDefinition/backupstoragelocations.velero.io: attempting to create resource

CustomResourceDefinition/backupstoragelocations.velero.io: already exists, proceeding

CustomResourceDefinition/backupstoragelocations.velero.io: created

CustomResourceDefinition/deletebackuprequests.velero.io: attempting to create resource

CustomResourceDefinition/deletebackuprequests.velero.io: already exists, proceeding

CustomResourceDefinition/deletebackuprequests.velero.io: created

CustomResourceDefinition/downloadrequests.velero.io: attempting to create resource

CustomResourceDefinition/downloadrequests.velero.io: already exists, proceeding

CustomResourceDefinition/downloadrequests.velero.io: created

CustomResourceDefinition/podvolumebackups.velero.io: attempting to create resource

CustomResourceDefinition/podvolumebackups.velero.io: already exists, proceeding

CustomResourceDefinition/podvolumebackups.velero.io: created

CustomResourceDefinition/podvolumerestores.velero.io: attempting to create resource

CustomResourceDefinition/podvolumerestores.velero.io: already exists, proceeding

CustomResourceDefinition/podvolumerestores.velero.io: created

CustomResourceDefinition/resticrepositories.velero.io: attempting to create resource

CustomResourceDefinition/resticrepositories.velero.io: already exists, proceeding

CustomResourceDefinition/resticrepositories.velero.io: created

CustomResourceDefinition/restores.velero.io: attempting to create resource

CustomResourceDefinition/restores.velero.io: already exists, proceeding

CustomResourceDefinition/restores.velero.io: created

CustomResourceDefinition/schedules.velero.io: attempting to create resource

CustomResourceDefinition/schedules.velero.io: already exists, proceeding

CustomResourceDefinition/schedules.velero.io: created

CustomResourceDefinition/serverstatusrequests.velero.io: attempting to create resource

CustomResourceDefinition/serverstatusrequests.velero.io: already exists, proceeding

CustomResourceDefinition/serverstatusrequests.velero.io: created

CustomResourceDefinition/volumesnapshotlocations.velero.io: attempting to create resource

CustomResourceDefinition/volumesnapshotlocations.velero.io: already exists, proceeding

CustomResourceDefinition/volumesnapshotlocations.velero.io: created

Waiting for resources to be ready in cluster...

Namespace/velero: attempting to create resource

Namespace/velero: created

ClusterRoleBinding/velero: attempting to create resource

ClusterRoleBinding/velero: already exists, proceeding

ClusterRoleBinding/velero: created

ServiceAccount/velero: attempting to create resource

ServiceAccount/velero: created

Secret/cloud-credentials: attempting to create resource

Secret/cloud-credentials: created

BackupStorageLocation/default: attempting to create resource

BackupStorageLocation/default: created

VolumeSnapshotLocation/default: attempting to create resource

VolumeSnapshotLocation/default: created

Deployment/velero: attempting to create resource

Deployment/velero: created

Velero is installed! ⛵ Use 'kubectl logs deployment/velero -n velero' to view the status.

* Deploy the sample nginx application

cd /home/ubuntu/velero-v1.4.0-linux-amd64/examples/nginx-app

* Deploy the base.yaml file using the following command

Kubectl apply -f base.yaml

Output:

namespace/nginx-example created

deployment.apps/nginx-deployment created

service/my-nginx created

**Backup:**

1. Create a backup for any object that matches the app=nginx label selector

Use the following commad

velero backup create nginx-backup --selector app=nginx

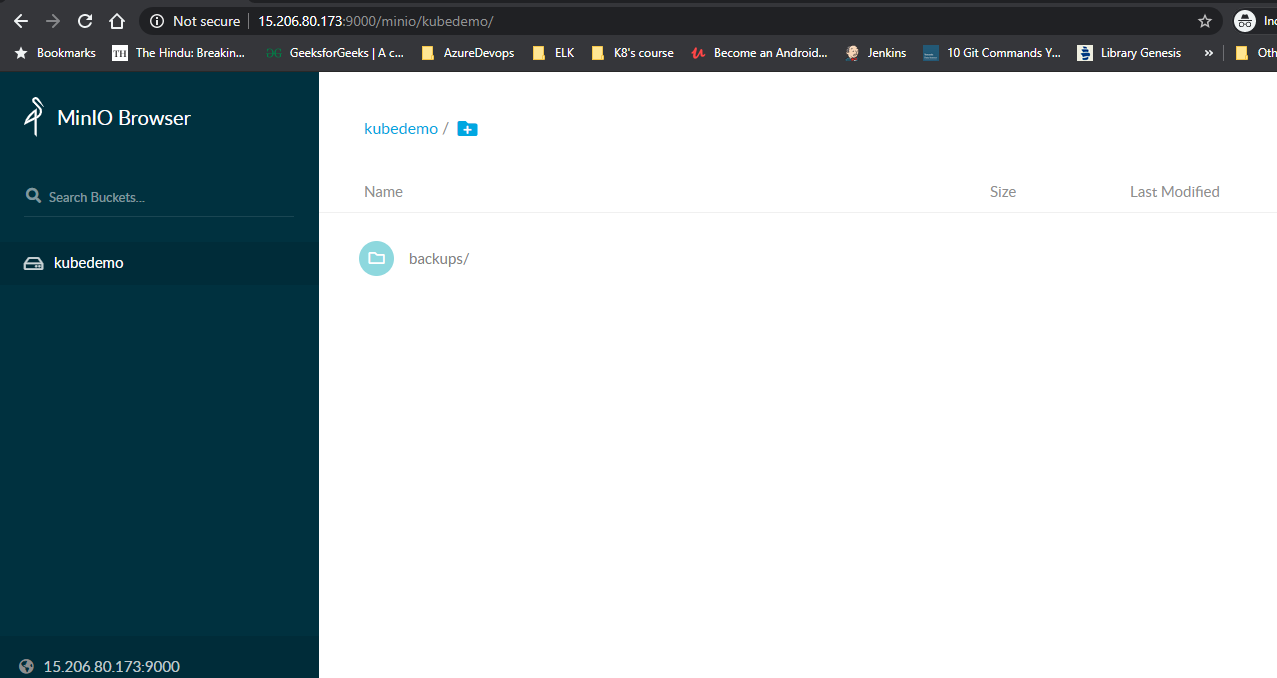
**Output:**

Backup request "nginx-backup" submitted successfully.

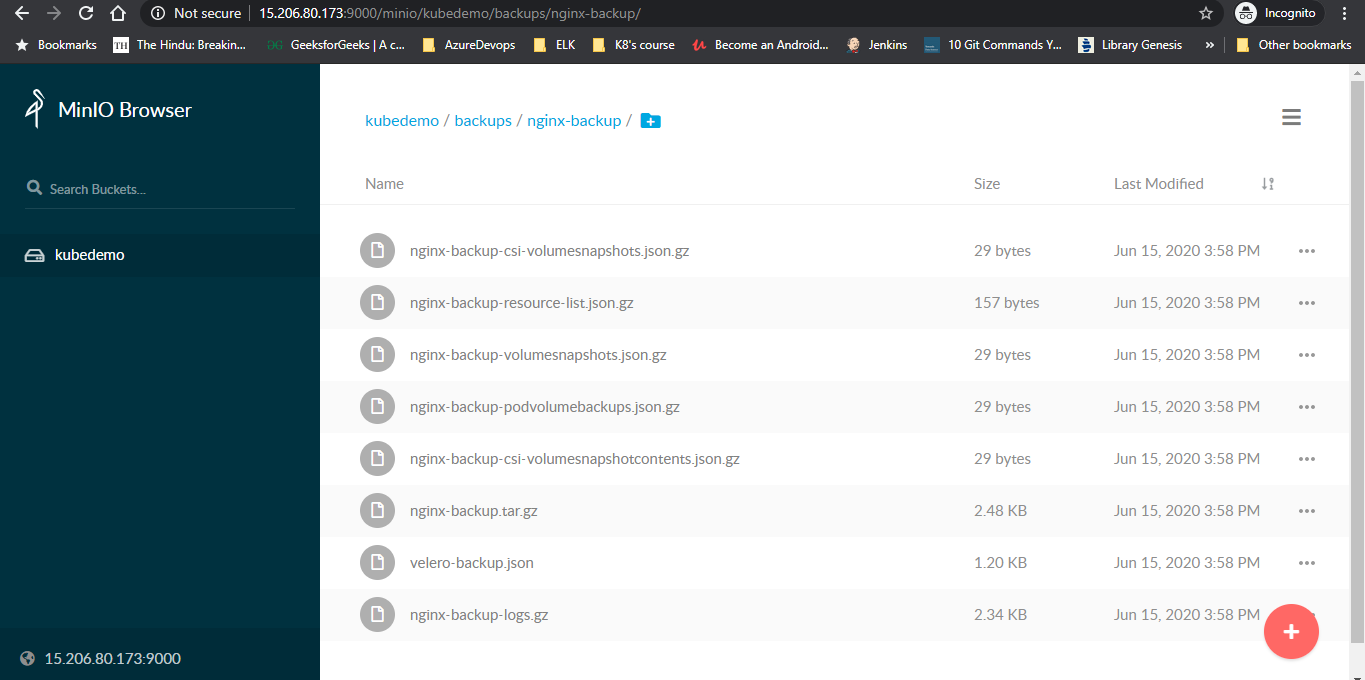
Run `velero backup describe nginx-backup` or `velero backup logs nginx-backup` for more details.

* Now navigate to the Minio browser verify the backup should appear on storage bucket.

Click on Backups



* Click on Nginx-backup name



* **Now delete a namespace nginx -example**

kubectl delete namespace nginx-example

**Output:**

namespace "nginx-example" deleted

**Restore:**

* velero restore create --from-backup nginx-backup

**Output:**

Restore request "nginx-backup-20200615104423" submitted successfully.

Run `velero restore describe nginx-backup-20200615104423` or `velero restore logs nginx-backup-20200615104423` for more details.

Run

**velero restore get**

**Output:**

NAME BACKUP STATUS WARNINGS ERRORS CREATED SELECTOR

nginx-backup-20200615104423 nginx-backup Completed 0 0 2020-06-15 10:44:23 +0000 UTC <none>

* You should see a namespace is restored with a latest time stamp

kubectl get ns

**Output:**

NAME STATUS AGE

default Active 55d

hrms-dileep Active 25d

kube-node-lease Active 55d

kube-public Active 55d

kube-system Active 55d

nginx-example Active 38s

rs2-dev-ns Active 14d

velero Active 16m

* Navigate to Minio browser and verify the restore folder

