



portworx
backup

Azure immutable backup location

Early Access

Table of Contents


Azure immutable backup location.....	3
Before you begin.....	4
Azure Portal Prerequisites.....	4
Get started.....	5
Install Portworx Backup.....	8
Internet-connected environment.....	8
Air-gapped environment.....	11
Install Stork 24.3.0-ea.....	12
Internet connected environment.....	12
Update Stork deployment.....	14
Air-gapped environment.....	14
Upgrade to 2.7.2 Early Access image.....	15
Internet connected environment.....	15
Air-gapped environment.....	18
Upgrade Stork.....	19
Internet connected environment.....	19
Air-gapped environment.....	20
Configure px-backup-config on Portworx Backup cluster.....	20
Add Azure immutable backup location.....	22
Manage Azure immutable backup location.....	24

Azure immutable backup location

Portworx Backup now supports Azure Blob immutable storage containers as backup locations. This feature aims to extend the current support for S3 object lock to Azure Blob Storage, ensuring data protection through immutability support and implements version-level Write Once Read Many (WORM) policy at both storage-account level and container level.

Immutability support helps to protect data from overwrites and deletes. Starting from Portworx Backup version 2.7.2, you can add your containers (that support immutability) as a backup location through the Portworx Backup web console with a prior [configuration](#). You will need new Azure credentials and configurations to add immutable Azure backup locations.

For more information, refer to [Azure immutable storage concepts](#).

 **Caution: This is an early access feature meant for testing purposes only. Please do not deploy or use in production environments until GA version is available.**



Note: Portworx Backup supports only time-based retention policy and not legal hold immutable policy.

As an early access feature, Azure immutable backup location supports:

1. Time-based retention policies with a minimum retention period of 7 days
2. Version-level WORM policies at both storage account and container levels
3. Locked and unlocked policies
4. Immutability with and without the soft delete option
5. Immutability for backups orchestrated via Portworx (cloudsnaps), CSI, and KDMP/generic backups

6. Azure Blob immutability in 'Global' and 'China' regions
7. Auto-detection of immutability on Azure Blob containers
8. Alerting for Azure immutable backup locations
9. Denial of delete operations on backups stored in Azure immutable backup location until their retention period is over
10. Validation mechanism for immutable Azure backup locations
11. Usage of backup schedule/policy with an immutable Azure backup location



Note: Legal Hold based immutability support, Blob granular and container-level WORM policy is out of scope for this feature.

Before you begin

1. Refer the following topics from Azure documentation:
 - a. [Container-level WORM policies for immutable blob data](#)
 - b. [Version-level WORM policies for immutable blob data](#)
 - c. [Configure immutability policies for containers](#)
 - d. [Configure immutability policies for blob versions](#)

Azure Portal Prerequisites

1. Azure [resource group](#) name
2. At the [Storage Account](#) Level:
 - **Enable versioning for blobs:** enable versioning on the storage account to ensure that every version of an object is preserved. This is essential for

maintaining the history of changes and for the retrieval of previous versions if needed.

- **Version level immutability support:** configure immutability support on the storage account to enforce WORM at the version level. This prevents any version of an object from being altered or deleted within a specified retention period.

3. Set [retention policy](#) either at storage account-level or container-level.



Note:

1. Portworx backup supports both locked and unlocked retention policies and you can set these policies at both storage account and container level through Azure Portal.
2. Retention policies set at container level will have precedence over those set at storage account level.

Get started

1. **Plan for install or upgrade:** refer the following Portworx Backup prerequisites and ensure that they are met:
 - a. **Install Prerequisites**

Component	Version
Kubernetes	1.29.x
Portworx	3.1.1
Stork	24.3.0-ea

b. Portworx Backup Image repositories

Image	Version
docker.io/portworx/pxcentral-onprem-api	2.7.2-ea
docker.io/portworx/pxcentral-onprem-ui-frontend	2.7.2-ea
docker.io/portworx/pxcentral-onprem-ui-backend	2.7.2-ea
docker.io/portworx/pxcentral-onprem-ui-lhb-backend	2.7.2-ea
docker.io/portworx/pxcentral-onprem-post-setup	2.7.2-ea
docker.io/portworx/px-backup	2.7.2-ea
docker.io/portworx/postgresql	11.19.0-debian-11-r1
docker.io/portworx/keycloak	21.1.2
docker.io/portworx/keycloak-login-theme	2.7.0
docker.io/portworx/busybox	1.35.0
docker.io/portworx/mysql	5.7.44
docker.io/portworx/mongodb	5.0.24-debian-11-r20
docker.io/portworx/kopiaexecutor	1.2.13
docker.io/portworx/nfsexecutor	1.2.13
docker.io/portworx/filesystemctl	1.2.13
docker.io/portworx/prometheus	v2.48.0
docker.io/portworx/alertmanager	v0.26.0
docker.io/portworx/prometheus-operator	v0.70.0
docker.io/portworx/prometheus-config-reloader	v0.70.0
openstorage/stork	24.3.0-ea
openstorage/cmdexecutor	24.3.0-ea

c. [AKS cluster prerequisite](#)

- d. [Plan for installation](#)
2. Install Portworx Backup on Portworx backup cluster, refer the following topics based on your environment to complete the installation:
 - [Install Portworx Backup in internet-connected environment](#)
 - [Install Portworx Backup in air-gapped environment](#)
3. Install Stork
 - [Install Stork 24.3.0-ea in internet connected environment](#)
 - [Install Stork 24.3.0-ea in air-gapped environment](#)
4. [Apply Portworx Backup Licenses](#)
5. Upgrade Portworx Backup
 - [Upgrade to 2.7.2 Early Access image in internet connected environment](#)
 - [Upgrade Portworx Backup in air-gapped environment](#)
6. Upgrade Stork
 - [Upgrade Stork in internet connected environment](#)
 - [Upgrade Stork in air-gapped environment](#)
7. [Configure px-backup-config](#)
8. [Add Azure cloud account](#)
9. [Add Azure immutable backup location](#)
10. [Manage Azure immutable backup location](#)
11. [Add AKS clusters](#)
12. [Create backup rules](#)
13. [Create schedule policies](#)

14. [Create a Backup](#)

15. [Restore a backup](#)

Install Portworx Backup

Internet-connected environment

1. If you are installing Portworx Backup alone without Portworx Enterprise, skip this step. If you want to install Portworx Backup with Portworx Enterprise, you must first [Install Portworx](#), then create the following storage class on your Kubernetes cluster:

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: portworx-sc
provisioner: kubernetes.io/portworx-volume
parameters:
  repl: "3"
```

2. From the CLI, install helm:

```
curl -fsSL -o get_helm.sh
https://raw.githubusercontent.com/helm/helm/main/scripts/get-helm-3
chmod 700 get_helm.sh
./get_helm.sh
```

3. Access [Portworx Central portal](#).
4. From the home page, navigate to **Explore our Products > Backup Services**.

5. Click on **I agree to EULA** and go through the **Portworx Products Terms of Use** carefully.
6. Navigate back to the [Portworx Central portal](#) and click **Start Free Trial**.
7. In the **Spec Details** tab provide the following values:
 - **Backup Version:** make sure you select version
 - **Namespace:** provide the name of the namespace where you want an instance of Portworx Backup to be installed
 - **Install using:** choose Helm 3
 - **Select your environment:** choose On-Premises or Cloud based on your storage environment
 - **StorageClass Name:** name of the StorageClass, refer tooltip for more details
 - **Use your OIDC:** select this checkbox only if your external authorization provider is Auth0 and key in the following fields:
 - Endpoint
 - Client ID
 - Client SecretThese values can be fetched from the **Auth0** web console.
 - **Use existing Prometheus:** select this checkbox if you have to use your existing Prometheus stack to monitor Portworx Backup and enter the values for the following fields:
 - **Prometheus Endpoint:** enter details of the endpoint where your Prometheus is installed
 - **Alertmanager Endpoint:** enter details of the endpoint where your Alertmanager is installed
 - **Prometheus secret name:** enter secret name of your Prometheus stack
 - **Alertmanager secret name:** enter secret name of your Alertmanager

- **Custom email template from PX-Backup:** select to upload Portworx Backup's custom email template to your pre-configured Alertmanager for email notifications
- **Use custom registry:** for air-gapped environments
 - **Custom Image Repository Location:** path of custom image repository
 - **Image Pull Secret(s):** create a secret only if image pulling from an internal repository requires credentials



Note: Create a secret only if image pulling from an internal repository requires credentials.

8. Click **Next** to navigate to the **Finish** tab.
9. From the machine where you run the `helm3` command, download the latest `px-central` package with the following command.

```
curl -L
https://github.com/portworx/helm/raw/2.7.2-EA/stable/px-central-2.7.2.tgz -o px-central-2.7.2.tgz
```

10. Execute the command under **Step 2** of the web console under **Install using the 'set' command**. To Install Portworx Backup with set command, replace

- a. `portworx/px-central` with `px-central-2.7.2.tgz`
- b. `--version 2.7.1` with `--version 2.7.2`

```
helm install px-central px-central-2.7.2.tgz --namespace
<px-backup-namespace> --create-namespace --version 2.7.2
--set
persistentStorage.enabled=true,persistentStorage.storageClass
Name="<storage-class-name>",pxbackup.enabled=true
```

- c. (Optional) For azure clusters configured with azure proxy, provide `proxy.azureProxyEnabled=true` with `--set` command:

```
helm install px-central px-central-2.7.2.tgz --namespace
<px-backup-namespace> --create-namespace --version 2.7.2
--set
persistentStorage.enabled=true,persistentStorage.storageClass
Name="<storage-class-name>",pxbackup.enabled=true,proxy.azure
ProxyEnabled=true
```



Note: Refrain from using the **Install using the 'values-px-central.yaml'** file option.

11. Click **Finish** to complete the installation.

Air-gapped environment

Before installation you need to pull the docker images listed in [Portworx Backup Image repositories](#). To pull the Docker images listed in and push them to an internal registry:

1. Download the install script for a specific release by specifying a `version` query. For example:

```
curl -o pxcentral-ag-install-backup.sh -L
"https://install.portworx.com/pxcentral-air-gapped?version=2.7.
2-ea&px-backup=true"
```

2. Provide execute permission for the install script:

```
chmod +x pxcentral-ag-install-backup.sh
```

3. Pull the container images using the `pxcentral-ag-install-backup.sh` script:

```
./pxcentral-ag-install-backup.sh pull
```

4. Push the images to a local registry server, accessible by the air-gapped nodes.

Replace <repo> with your registry location.

```
./pxcentral-ag-install-backup.sh push <repo>
```

5. Execute Step 3 to Step 11 from [Install Portworx Backup in internet-connected environment](#).

Install Stork 24.3.0-ea

Internet connected environment

You can install Stork with or without Portworx Enterprise using the following methods:

Deployment method without Portworx Enterprise

To install Stork version 24.3.0-ea on your Kubernetes cluster without installing Portworx Enterprise, run the below commands:

1. Download the Stork deployment spec:

```
curl -fsL -o stork-spec.yaml  
"https://install.portworx.com/pxbackup?comp=stork&storkNonPx=true"
```

2. In the `stork-spec.yaml`, change the Stork version to 24.3.0-ea if the version differs.
3. Apply the `stork-spec.yaml` to install the latest Stork version:

```
kubectl apply -f stork-spec.yaml
```

Deployment Method with Portworx Enterprise

If you have to install Stork 24.3.0-ea along with Portworx Enterprise, you can opt-in for Portworx Operator installation

Stork fresh installation for Portworx Backup through web console

If Stork is not installed as part of Portworx deployment, perform the following steps:

1. From the home page, click **Add cluster**.
2. Choose your Kubernetes platform.
3. Provide cluster name and Kubeconfig details.
4. Click Px-cluster to copy the stork installation command.
5. Run the Stork installation command.



Note: If Stork is installed through the PX-Cluster option from the web console in a namespace other than the namespace where Portworx Enterprise is deployed, perform Step 6 or else go to Step 7.

6. Update the following key-value pairs in stork deployment's (`stork-spec.yaml`) environment variable section, using `kubectl edit` command.

```
kubectl edit deployment stork -n <stork-namespace>
```

```
env:
  - name: PX_NAMESPACE
    value: <portworx-deployed-namespace>
  - name: PX_SERVICE_NAME
    value: portworx-api
  - name: STORK-NAMESPACE
    value: portworx
```

7. Click **Add Cluster**.

Update Stork deployment

Perform the below steps to update Stork installation using Portworx operator option:

1. Edit the stc (Kubernetes resource):

```
kubectl edit stc -n <portworx-deployed-namespace>
```

2. Append the Stork image and version details in Stork section:

```
stork:
args:
  webhook-controller: "true"
  enabled: true
  image: openstorage/stork:24.3.0-ea
```

3. Save and exit.

Air-gapped environment

1. If your application cluster is air-gapped, then you must pull the following images before installing Stork:

Image	Version
openstorage/stork	24.3.0-ea
openstorage/cmdexecutor	24.3.0-ea
openstorage/kopiaexecutor	1.2.13
openstorage/nfsexecutor	1.2.13

2. Push the above images to your internal registry server, accessible by the air-gapped nodes.

3. After pushing the images, follow the instructions in [How to install Stork](#) based on your deployment methods to install your Stork version.

Upgrade to 2.7.2 Early Access image

Internet connected environment

To upgrade from Portworx Backup version to 2.7.2 early access image in internet-connected environments:

1. Access [Portworx Central portal](#).
2. From the home page, navigate to **Backup Services** under **Explore our Products**.
3. Click **I agree to EULA** and go through the **Portworx Products Terms of Use** carefully.
4. Navigate back to the [Portworx Central portal](#) and click **Start Free Trial**.
5. In the Spec Details provide the following values:
 - a. **Backup Version:** select version as **2.7.1**
 - b. **Namespace:** provide the name of the namespace where you want an instance of Portworx Backup to be installed
 - c. **Install using:** choose Helm 3
 - d. **Select your environment:** choose On-Premises or Cloud based on your storage environment
 - e. **StorageClass Name:** name of the StorageClass, refer tooltip for more details
 - f. **Use your OIDC:** select this option only if your external authorization provider is Auth0 and key in the following fields:
 - Endpoint
 - Client ID
 - Client Secret
 - g. These values can be fetched from the Auth0 web console.

- h. **Use existing Prometheus:** select this checkbox if you have to use your existing Prometheus stack to monitor Portworx Backup and enter the values for the following fields:
- **Prometheus Endpoint:** enter details of the endpoint where your Prometheus is installed
 - **Alertmanager Endpoint:** enter details of the endpoint where your Alertmanager is installed
 - **Prometheus secret name:** enter secret name of your Prometheus stack
 - **Alertmanager secret name:** enter secret name of your Alertmanager
 - **Custom email template from PX-Backup:** select to upload Portworx Backup's custom email template to your pre-configured Alertmanager for email notifications
6. **(Optional)** By default, persistent volume size for Prometheus server is 5 GB, if you need more storage, use the below command during the upgrade from previous versions of Portworx Backup to 2.7.2-EA:

```
set persistentStorage.prometheus.storage=8Gi,  
persistentStorage.prometheus.retentionSize=<92% of  
prometheus.storage in MB>7360MB
```

The command above resets the Prometheus server's persistent volume size to 8 GB. You can set the required storage based on your needs.

7. From the machine where you run the helm3 command:
- a. Download the latest px-central package with the following command.

```
curl -L  
https://github.com/portworx/helm/raw/2.7.2-EA/stable/px-cent  
ral-2.7.2.tgz -o px-central-2.7.2.tgz
```


- b. Delete the post install hook job:

```
kubectl delete job pxcentral-post-install-hook --namespace  
<namespace>
```

8. You need to upgrade Portworx Backup with default options using set command in the spec gen. Modify the Helm command generated using the **set** command in **Step 2** in **Finish** tab) of the Portworx Backup web console to provide the helm package, instead of providing the repository. To upgrade Portworx Backup with set command:

- a. Replace:

- `helm install` with `helm upgrade`
- `portworx/px-central` with `px-central-2.7.2.tgz`
- `--version 2.7.1` with `--version 2.7.2`

- b. Remove the argument : `--create-namespace`

Set command looks like this after all the modifications:

```
helm upgrade px-central px-central-2.7.2.tgz --namespace  
<px-backup-namespace> --version 2.7.2 --set  
persistentStorage.enabled=true,persistentStorage.storageClass  
Name="<storage-class-name>",pxbackup.enabled=true
```

- c. For azure clusters configured with azure proxy, provide

`proxy.azureProxyEnabled=true` with `--set` command:

```
helm upgrade px-central px-central-2.7.2.tgz --namespace  
<px-backup-namespace> --version 2.7.2 --set  
persistentStorage.enabled=true,persistentStorage.storageClass  
Name="<storage-class-name>",pxbackup.enabled=true,proxy.azu  
reProxyEnabled=true
```

- d. Click **Finish** to execute the command under **Step 2** of the web console under **Install using the 'set' command**.
9. (Optional) Delete the Prometheus operator deployment upgrade to avoid conflicts:

```
kubectl delete deploy prometheus-operator -n <px-backup namespace>
```



Note: Execute this step only if you have configured Prometheus and Grafana following the steps mentioned in this topic [Configure Prometheus and Grafana](#).

Air-gapped environment

For an air-gapped environment before starting with this task ensure that you have pushed the images listed in the topic [Portworx Backup Image repositories](#) to your internal registries. Follow the steps below to upgrade Portworx Backup from the prior versions to 2.7.2-EA:

1. Download the install script for a specific release by specifying a version query.

For example:

```
curl -o pxcentral-ag-install-backup.sh -L  
"https://install.portworx.com/pxcentral-air-gapped?version=2.7.2-ea&px-backup=true"
```

2. Provide execute permission for the install script:

```
chmod +x pxcentral-ag-install-backup.sh
```

3. Pull the container images using the `pxcentral-ag-install-backup.sh` script:

```
./pxcentral-ag-install-backup.sh pull
```

4. Push the images to a local registry server, accessible by the air-gapped nodes.
Replace `<repo>` with your registry location:

```
./pxcentral-ag-install-backup.sh push <repo>
```

5. Execute step 2 to 9 from [Upgrade to 2.7.2 Early Access image in internet connected environment](#).

Upgrade Stork

Internet connected environment

You can upgrade Stork with or without Portworx Enterprise using the following methods:

Upgrade Method with or without Portworx Enterprise

If you have to upgrade Stork 24.3.0-ea along with Portworx Enterprise, you can opt-in for Daemonset upgrade or Portworx Operator upgrade:

Portworx Operator method

Perform the below steps for Stork upgrade using Portworx operator option:

1. Edit the stc (Kubernetes resource):

```
kubectl edit stc <stc-name> -n <portworx-deployed-namespace>
```

2. Change the Stork image and version details in Stork section:

```
stork:
args:
  webhook-controller: "true"
  enabled: true
  image: openstorage/stork:24.3.0-ea
```

3. Save and exit.

Air-gapped environment

1. If your application cluster is air-gapped, then you must pull the following openstorage images before upgrading Stork:

Image	Version
openstorage/stork	24.3.0-ea
openstorage/cmdexecutor	24.3.0-ea
openstorage/kopiaexecutor	1.2.13
openstorage/nfsexecutor	1.2.13

2. You must then push the above images to your internal registry server, accessible by the air-gapped nodes.
3. After pushing the images, follow the instructions in the Upgrade Stork section at the top of this page based on your deployment methods to upgrade your stork version.

Configure px-backup-config on Portworx Backup cluster

Perform the following configuration steps to update px-backup-config ConfigMap on Portworx Backup cluster:

1. SSH to Portworx Backup cluster
2. To view the px-backup-config ConfigMap:

```
kubectl get cm px-backup-config -npx-backup -oyaml
```

Output:

```
apiVersion: v1
data:
  ALL-CLUSTER-UPDATED-TO-BETA1: "true"
  BACKUP_SYNC_LIMIT_CPU: ""
  BACKUP_SYNC_LIMIT_MEMORNFS: ""
  BACKUP_SYNC_REQUEST_CPU: ""
  BACKUP_SYNC_REQUEST_MEMORY: ""
  KDMP_KOPIAEXECUTOR_IMAGE:
    portworx/kopiaexecutor:1.2.12
  KDMP_KOPIAEXECUTOR_IMAGE_SECRET: ""
  OBJECT-LOCK-INCR-BACKUP-COUNT: ""
  OBJECT-LOCK-RETAIN-COUNT: ""
  new-key: new-value
kind: ConfigMap
metadata:
  creationTimestamp: "2024-05-30T10:21:17Z"
  name: px-backup-config
  namespace: px-backup
  resourceVersion: "579866"
  uid: 0bc45af8-c980-46a2-baff-82735b8abe36
```

Where px-backup-namespace is the namespace where you have deployed Portworx Backup.

3. Edit the px-backup-config with the following command:

```
kubectl edit cm px-backup-config <px-backup-namespace>
```

4. Add the following key-value pair in data component:

```
ENABLE_AZURE_IMMUTABILITY_SUPPORT: "true"
```



Note: To view all the ConfigMaps in the Portworx Backup namespace you can use the following command:

```
kubectl get cm <px-backup-namespace>
```

Add Azure immutable backup location

To add an Azure immutable backup location use the Portworx Backup web console. You can add an Azure immutable backup location based out of any geography worldwide or in China.

Prerequisites

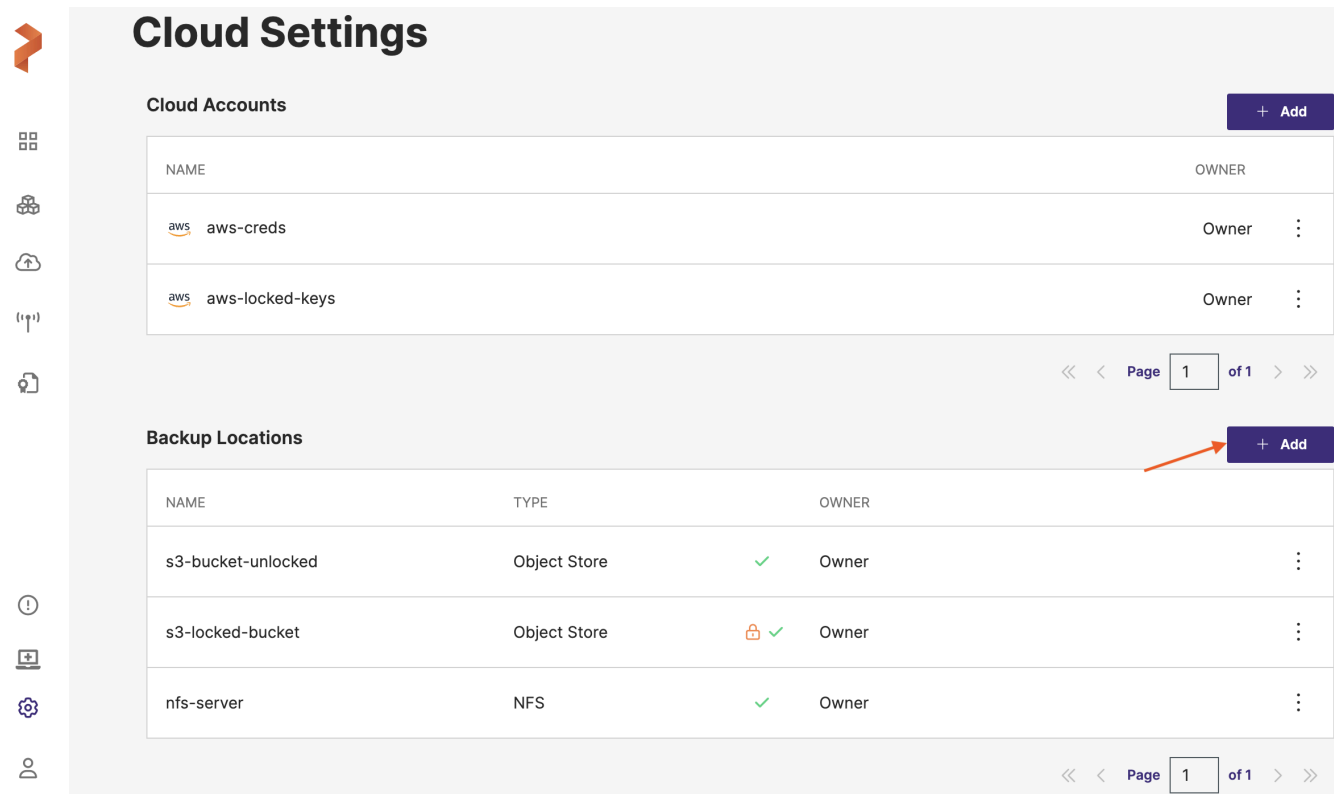
- Make sure a locked or unlocked container exists with versioning enabled and immutable support set

Perform the following steps to add Azure global backup location in Portworx Backup:

1. In the home page, from the left navigation pane, click **Clusters**.
2. At the upper-right corner, click **Settings > Cloud Settings**.

CLUSTER NAME	STATUS	PROTECTED APPS	PROTECTED DATA	K8S VERSION	PERMISSIONS	OWNER
k8s-cluster	✓ Active	1	0 B	v1.28.0	Full access ✓	AA
rke	✓ Active	0	0 B		Full access ✓	AA

3. In the **Backup Locations** section, click **Add**:



Cloud Settings

Cloud Accounts

NAME	OWNER
aws-creds	Owner
aws-locked-keys	Owner

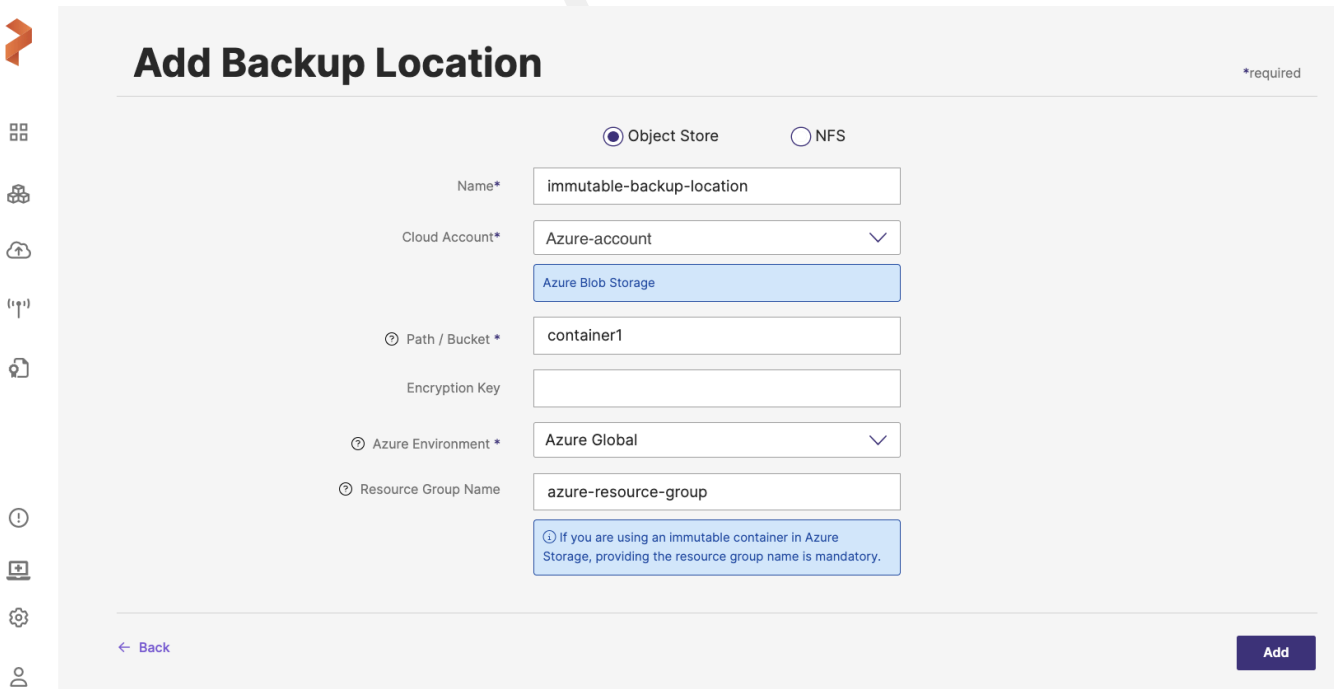
Page 1 of 1

Backup Locations

NAME	TYPE		OWNER
s3-bucket-unlocked	Object Store	✓	Owner
s3-locked-bucket	Object Store	🔒 ✓	Owner
nfs-server	NFS	✓	Owner

Page 1 of 1

4. In the **Add Backup Location** window, Populate the following fields:



Add Backup Location

*required

☒ Object Store ☐ NFS

Name* immutable-backup-location

Cloud Account* Azure-account

Azure Blob Storage

Path / Bucket * container1

Encryption Key

Azure Environment * Azure Global

Resource Group Name azure-resource-group

ⓘ If you are using an immutable container in Azure Storage, providing the resource group name is mandatory.

← Back Add

- **Name:** specify the name for the backup location, Portworx Backup displays this name as backup location name in the web console
- **Cloud Account:** choose the Azure credentials this backup location will use to create backups
- **Path/Bucket:** specify the name of the container this backup location will place backups onto
- **Encryption key (Optional):** enter the optional encryption key to encrypt your backups in-transit.
- **Azure Environment:** this drop-down lists two regions; Azure Global and Azure China and by default takes Azure Global as the value for Azure environment. Choose Azure Global or Azure China (based on the geographical location of your backup target) to add your Azure backup location.
- **Resource group name:** name of the Azure resource group and this field is mandatory to add an immutable Azure backup location

5. Click **Add**.

Portworx Backup validates the data you have provided in the above fields, adds the backup location if the data is accurate and then displays the added Azure immutable backup location in **Home** page > **Clusters** > **Settings** > **Cloud Settings** > **Backup Locations** page.

If data provided is inaccurate, the web console displays an error message stating it failed to add the backup location.

Manage Azure immutable backup location

After you add an Azure immutable backup location, you can perform the following actions on the added backup location:

Backup Locations			+ Add	
NAME	TYPE	OWNER		
s3-backup-location	Object Store	Owner	View Json	⋮
azure-blob-location	Object Store	Owner	Remove	⋮
			Edit	⋮
			Validate	⋮
nfs-backup-location	NFS	Owner	User Access	⋮

- **View Json:** Provides metadata, detailed information on backup location and the email ID of the backup location owner
- **Remove:** deletes the backup location from the system and the display
- **Edit:** Allows you to change the cloud account associated with the backup location
- **Validate:** validates the backup location after it is added
- **User Access:** allows you to change the backup location accessibility to public (all the users who use Portworx Backup web console) or to a single user or group



Note: Backups stored in immutable container(s) or containers associated with immutable storage accounts cannot be deleted if their retention period is still active. You can delete the backups only after the container(s) retention period expires.



portworx[®]

by Pure Storage[®]

