



portworx  
backup

# **Concurrent Deletion of Backups**


**2.8.0**

# Table of Contents

Overview.....	3
About concurrent deletion.....	3
Workflow.....	5
Benefits.....	6
Use Cases.....	6
Get started.....	7
Install Portworx Backup.....	9
Internet-connected environment.....	9
Air-gapped environment.....	12
Install Stork 24.3.3-ea.....	13
Internet connected environment.....	13
Update Stork deployment.....	15
Air-gapped environment.....	16
Upgrade to 2.8.0 Early Access image.....	16
Internet connected environment.....	16
Air-gapped environment.....	19
Upgrade Stork.....	20
Internet connected environment.....	20
Air-gapped environment.....	21
Post Installation.....	22

# Overview

Concurrent or parallel deletion process in Portworx Backup handles the deletion of backups and their associated volume content efficiently and concurrently. This feature allows users to delete backups while ensuring dependencies, such as incremental backups, are managed correctly.

 **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.**

## About concurrent deletion

1. You can delete multiple backups in parallel, improving the efficiency of the deletion process.
2. By default, 5 backups are selected for deletion at a time, with each backup deletion process spawning multiple threads for the deletion of associated volumes.
3. **Thread configuration:**

Users can adjust the number of concurrent deletions or volume threads by editing two parameters in the Portworx Backup deployment specification. The number of backups being processed concurrently and the threads assigned for each backup can be adjusted with the following parameters in the `px-backup` deployment spec:

```
- name: BACKUP_DELETE_WORKER
  value: "5"
- name: VOL_DELETE_WORKER
  value: "5"
```

- `BACKUP_DELETE_WORKER`: defines the maximum number of backups to be processed concurrently and the default value is 5
- `VOL_DELETE_WORKER`: defines the maximum number of threads assigned per backup to handle volume deletion (default value is 5).

#### 4. Deletion states:

- **Deleting state:**

- Once a user initiates a deletion action (either through the UI or API), the backup enters the **Deleting** state.
- This state signals that the deletion process has been requested and that the backup is now eligible for further processing by the delete threads.
- The **Deleting** state indicates that the backup is in the queue for deletion

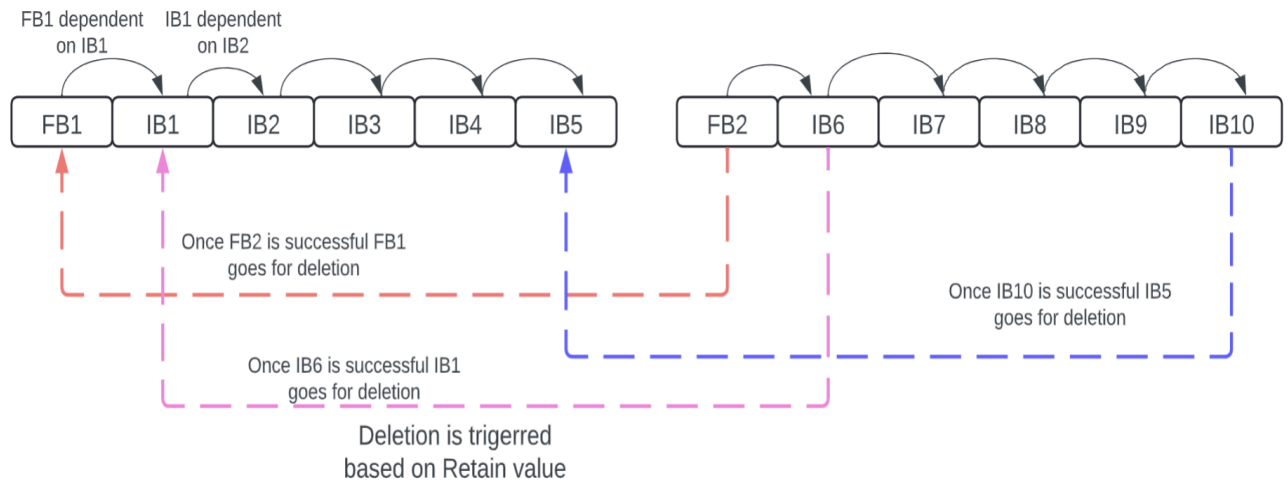
- **Delete Pending state:**

- After a delete thread picks up a backup for deletion, if any volumes associated with the backup belong to Portworx (PX), the system performs a check for dependent incremental backups.
- If such dependencies are found (e.g., incremental backups that rely on the backup being deleted), these dependent backups are moved to the **Delete Pending** state.
- The **Delete Pending** state indicates that there are other backups that must be deleted before the primary backup can be deleted.
- This ensures the system maintains the integrity of incremental backup chains, preventing data loss from accidental or premature deletions.

#### 5. Dependency Management:

- The system automatically scans for dependent backups, particularly in the case of incremental backups, ensuring that backup chains remain intact.
- Incremental backups that rely on previously deleted full or incremental backups are flagged, and their deletion is placed in a pending state until their dependencies are resolved.

For example, consider an incremental backup with `incr = 5` (5 incremental backups are created between each full backup) and `retain = 6`, where 6 backups (1 full + 5 incremental) are retained before deletion is triggered.



After FB2 is successfully created, FB1 is marked for deletion, and after IB6 is successful, IB1 can be deleted. After IB10 is successful, IB5 is also marked for deletion, with the strategy to retain the most recent backups and deleting the older ones.

## Workflow

### 1. User-Initiated Deletion:

- A user requests the deletion of a backup from the web console
- The backup transitions into the **Deleting** state.

### 2. Concurrent Backup Deletion:

- A maximum of 5 backups are selected for deletion, with each backup processing up to 5 volume threads concurrently.
- The thread configuration can be modified as needed in the `px-backup` deployment spec.

### 3. **Dependency Check:**

- a. Once the deletion threads begin processing, the system checks if the volumes belong to Portworx.
- b. If there are dependent incremental backups, these backups are moved to the **Delete Pending** state.

### 4. **Final Deletion:**

- a. Once the dependencies are resolved and no incremental backups are relying on the backup , the deletion process proceeds, and the backup is deleted.

## **Benefits**

- **Increased Efficiency:** Parallel deletion of multiple backups and their volumes reduces overall deletion time.
- **Customizable Performance:** Thread configurations allow users to tune the deletion process to suit their environment and resource availability.

## **Use Case**

- **Large Scale Deletion:** In environments where large numbers of backups need to be deleted quickly, this enhancement allows for concurrent processing, saving time.



**Note:** Ensure that they have properly configured the `BACKUP_DELETE_WORKER` and `VOL_DELETE_WORKER` values based on your cluster's performance and resources.

## 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	1.31 and below
Portworx	3.2.0
Stork	24.3.3-ea

- b. **Portworx Backup Image repositories**

Image	Version
<code>docker.io/portworx/pxcentral-onprem-api</code>	2.8.0-ea
<code>docker.io/portworx/pxcentral-onprem-ui-frontend</code>	2.8.0-ea
<code>docker.io/portworx/pxcentral-onprem-ui-backend</code>	2.8.0-ea
<code>docker.io/portworx/pxcentral-onprem-ui-lhbackend</code>	2.8.0-ea
<code>docker.io/portworx/pxcentral-onprem-post-setup</code>	2.8.0-ea
<code>docker.io/portworx/pxcentral-onprem-pre-upgrade</code>	2.8.0-ea
<code>docker.io/portworx/px-backup</code>	2.8.0-ea

<code>docker.io/portworx/postgresql</code>	11.19.0-debian-11-r1
<code>docker.io/portworx/keycloak</code>	25.0.2
<code>docker.io/portworx/keycloak-login-theme</code>	2.8.0
<code>docker.io/portworx/busybox</code>	1.35.0
<code>docker.io/portworx/mysql</code>	8.0.37
<code>docker.io/portworx/mongodb</code>	7.0.14-debian-11-r0
<code>docker.io/portworx/mongodb</code>	6.0.13-debian-11-r21
<code>docker.io/portworx/kopiaexecutor</code>	1.2.15
<code>docker.io/portworx/nfsexecutor</code>	1.2.15
<code>docker.io/portworx/filesystemctl</code>	1.2.15
<code>docker.io/portworx/prometheus</code>	v2.53.0
<code>docker.io/portworx/alertmanager</code>	v0.26.0
<code>docker.io/portworx/prometheus-operator</code>	v0.75.0
<code>docker.io/portworx/prometheus-config-reloader</code>	v0.75.0
<code>openstorage/stork</code>	24.3.3-ea



## 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 as **2.7.2**
  - **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 Secret

These 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/2.8.0-EA/stable/px-central-2.8.0.tgz -o px-central-2.8.0.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-2.8.0.tgz`
- b. `--version 2.7.2` with `--version 2.8.0`

```
helm install px-central px-central-2.8.0.tgz --namespace
<px-backup-namespace> --create-namespace --version 2.8.0
```

```
--set  
persistentStorage.enabled=true,persistentStorage.storageClass  
sName="<storage-class-name>",pxbackup.enabled=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=2.8.  
0-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 24.3.3-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 24.3.3-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 24.3.3-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 24.3.3-ea along with Portworx Enterprise, you can opt-in for Portworx Operator installation

### Stork 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:24.3.3-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	24.3.3-ea
openstorage/cmdexecutor	24.3.3-ea
openstorage/kopiaexecutor	1.2.15
openstorage/nfsexecutor	1.2.15

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 2.8.0 Early Access image

### Internet connected environment

To upgrade from Portworx Backup version to 2.8.0 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:
- Backup Version:** select version as **2.7.2**
  - 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 option only if your external authorization provider is Auth0 and key in the following fields:
    - Endpoint
    - Client ID
    - Client Secret
  - These 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

6. 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/2.8.0-EA/stable/px-cent
ral-2.8.0.tgz -o px-central-2.8.0.tgz
```

- b. Delete the post install hook job:

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

7. 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-2.8.0.tgz`
- `--version 2.7.2` with `--version 2.8.0`

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

- c. Add helm timeout `--timeout 120m0s`

Set command looks like this after all the modifications:

```
helm upgrade px-central px-central-2.8.0.tgz --namespace  
<px-backup-namespace> --version 2.8.0 --set  
persistentStorage.enabled=true,persistentStorage.storageClass  
Name="<storage-class-name>",pxbackup.enabled=true --timeout  
120m0s
```

- d. Click **Finish** to execute the command under **Step 2** of the web console under **Install using the 'set' command**.
8. (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 2.8.0-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=2.8.  
0-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 2.8.0 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 24.3.3-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:24.3.3-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	24.3.3-ea
openstorage/cmdexecutor	24.3.3-ea
openstorage/kopiaexecutor	1.2.15
openstorage/nfsexecutor	1.2.15

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 based on your deployment methods to upgrade your stork version.

## Post Install/Upgrade

Refer the following modules for post install/upgrade tasks:

1. [Configure Portworx Backup](#)
2. [Operate Portworx Backup](#)

Early Access



portworx<sup>®</sup>

by Pure Storage<sup>®</sup>

