PSO2PX USER GUIDE - v1.0.0-BETA VERSION

PSO2PX Tool User Guide

“The PSO to Portworx Volume Migration Automation Tool”

Contents

Introduction

The *pso2px* tool is a containerized, python automation tool. *pso2px* is designed to help PSO customers migrate PSO volumes to Portworx. It converts PSO's Kubernetes storage objects, including StorageClasses, PVs and PVCs, into Portworx. This allows the Portworx driver to take over the driver control to continue volume CURD operations and data IOs. *pso2px* requires Portworx version 2.11 or greater, which includes *FlashArray DirectAccess (FADA)* and *FlashBlade DirectAccess (FBDA)* features.

pso2px supports *pure-block* and *pure-file* type volumes migration. It can migrate any typical Kubernetes application types, such as:

- pod
- daemonset
- deployment
- statefulset

pso2px only converts Kubernetes objects, and does not copy data during migration. Usually, migrating one k8s namespace only requires one or two minutes. Before starting, *pso2px performs* a preflight check to ensure that your PSO and Porworx setups are qualified for migration.

Disclaimer

The current pso2px release version (v1.0.0-beta) is intended for testing purposes only. Pure Storage does not recommend you attempt the migration in a production environment.

Prerequisites

- PSO 6.x.x installed
- Portworx 2.11.0-ea-migration-1 installed
- Applications actively using PSO PVC's to be scaled down manually.

Limitation and Scope

- Application downtime: The migration tool migrates the PSO driver to the Portworx driver (control plane), and results in application downtime due to Kubenetes constraints. Prior to migrating a namespace, you must scale down **all** applications in that specific namespace, run the migration tool, and scale up the application.
- pso2px does not support Snapshot object migration.
- PSO 5 migration is not supported. We plan to support it in the v1.1.0 version.
- pso2px cannot migrate any volume sizes greater than 40TiB. We plan to support it in the v1.1.0 version.

Tool Installation

1. Save the following yaml spec to pso2px.yaml file.

   wget https://raw.githubusercontent.com/purestorage/pso-csi/master/pso2px/v1.0.0-beta/pso2px.yaml

```yaml
# PSO to PX Tool Spec
apiVersion: v1
kind: ServiceAccount
metadata:
  name: pso2px-tool
---
kind: ClusterRole
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: pso2px-tool
rules:
  - apiGroups: ["*"]
    resources: ["*"]
    verbs: ["*"]
---
kind: ClusterRoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: pso2px-tool
subjects:
- kind: ServiceAccount
  name: pso2px-tool
  namespace: default
roleRef:
  kind: ClusterRole
  name: pso2px-tool
  apiGroup: rbac.authorization.k8s.io
---
apiVersion: v1
kind: Pod
metadata:
  name: pso2px
  labels:
    app: pso2px
spec:
  containers:
```

```
 - name: pso2px
   image: portworx/pso2px:v1.0.0-beta
   imagePullPolicy: Always
   volumeMounts:
   - mountPath: /csi.sock
     name: csi-volume
   - mountPath: /usr/local/pso2px/k8s_backups
     name: backup-path
volumes:
  - name: csi-volume
    hostPath:
      path: /var/lib/kubelet/plugins/pxd.portworx.com/csi.sock
  - name: backup-path
    hostPath:
      path: /usr/local/pso2px/k8s_backups
      type: DirectoryOrCreate
serviceAccount: pso2px-tool
serviceAccountName: pso2px-tool
```

2. Apply the yaml file to create a running pod of the tool:

   `kubectl apply -f pso2px.yaml`

3. check if the pod is up and running:

   `kubectl describe pod/pso2px`

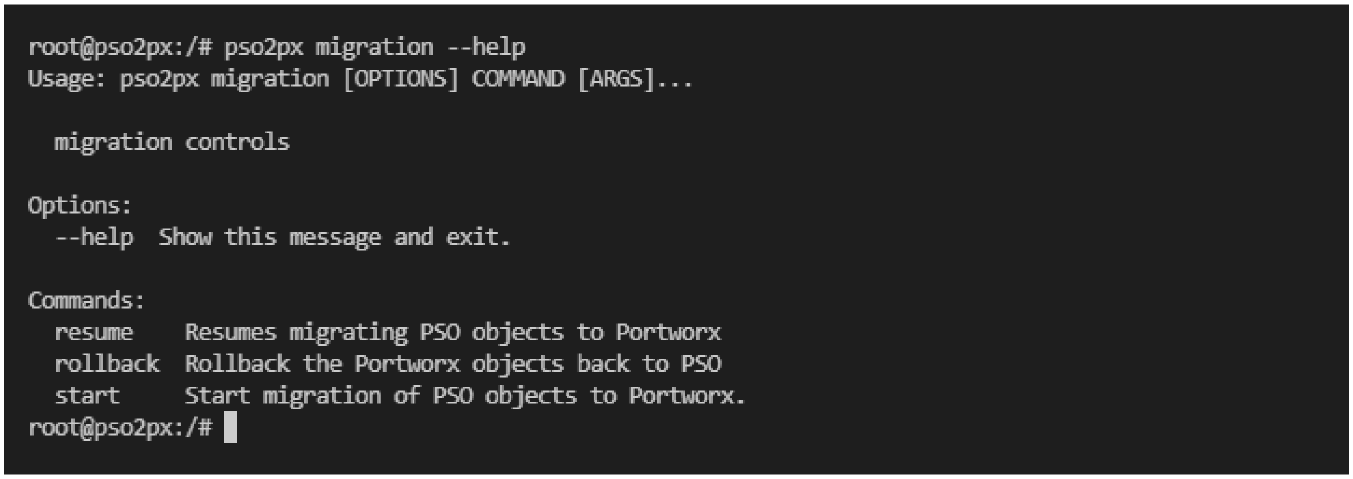
How to run the pso2px tool

From your terminal, run the following command to login to the container's shell environment:

`kubectl exec -it pso2px -- bash`

Usage - migration

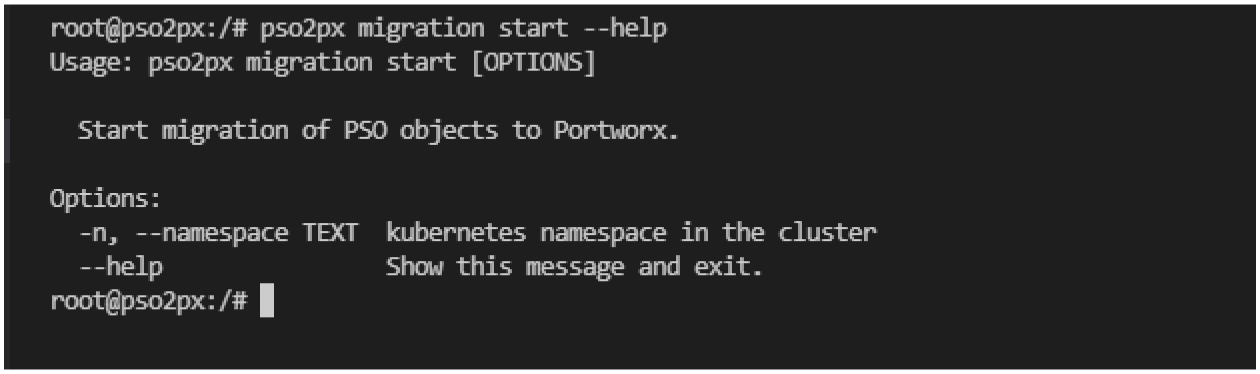
The pso2px tool's migration function has 3 major sub-commands: `start`, `rollback`, and `resume`.



```
root@pso2px:/# pso2px migration --help
Usage: pso2px migration [OPTIONS] COMMAND [ARGS]...

  migration controls

Options:
  --help  Show this message and exit.

Commands:
  resume    Resumes migrating PSO objects to Portworx
  rollback  Rollback the Portworx objects back to PSO
  start     Start migration of PSO objects to Portworx.
root@pso2px:/#
```

Note: The beta release only supports the "start" sub-command. The "resume" and "rollback" sub-commands will be supported in the next version of the tool.

Usage - migration start (sub-command)



```
root@pso2px:/# pso2px migration start --help
Usage: pso2px migration start [OPTIONS]

  Start migration of PSO objects to Portworx.

Options:
  -n, --namespace TEXT  kubernetes namespace in the cluster
  --help                Show this message and exit.
root@pso2px:/#
```

pso2px can migrate PSO objects either in a particular Kubernetes namespace, or all the available namespaces.

To perform migration all namespaces, enter the following command:

`pso2px migration start`

Pure Storage does not recommend migrating all namespaces at once. Migrating namespace-by-namespace can limit the impact of your service downtime only within one namespace. The services in other namespaces will still be up and running while migrating the current namespace.

To perform migration on a particular namespace, you must pass on the namespace as an option to the migration command:

```
pso2px migration start --namespace <k8s-namespace>
```

Note: Please make sure that you scale down all applications in the namespace before you start the migration. Tool has the preflight check to check it for you. The preflight check will stop if it detects your application's replica is greater than 0.

Usage - migration resume (sub-command)

The "`migration resume`" sub-command is not supported in the beta version yet.

Usage - migration rollback (sub-command)

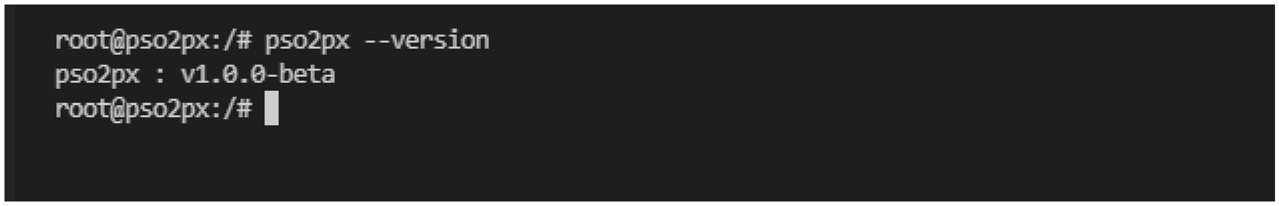
The "`migration rollback`" sub-command is not supported in the beta version yet.

Usage - version

Once you are in the shell, you can now launch the pso2px migration tool.

To see which the version of pso2px you're running, enter the following command:

```
pso2px --version
```

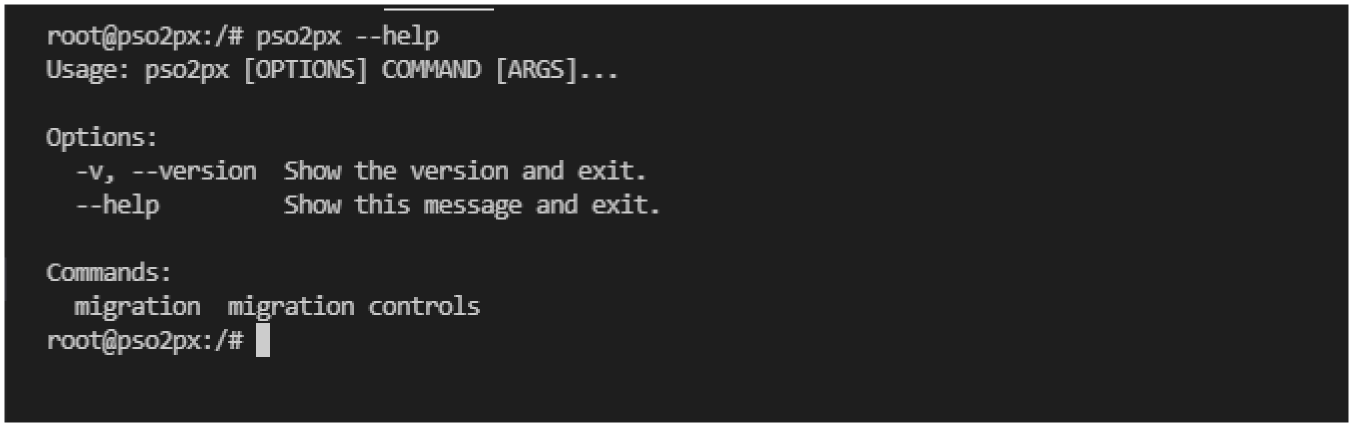


```
root@pso2px:/# pso2px --version
pso2px : v1.0.0-beta
root@pso2px:/#
```

Usage - help

To view the pso2px usage options, enter the following command:

```
pso2px --help
```



```
root@pso2px:/# pso2px --help
Usage: pso2px [OPTIONS] COMMAND [ARGS]...

Options:
  -v, --version  Show the version and exit.
  --help         Show this message and exit.

Commands:
  migration  migration controls
root@pso2px:/# 
```

Logging

pso2px prints the progress of the tool to the console, and doesn't redirect anything to a log file. To redirect the output to both the console and to a log file, you can pipe the pso2px output to the tee command:

```
pso2px migration start -n <namespace> 2>&1 | tee pso2px.log
```

Migration Workflow

We explain the details of the migration automation process in the tool for people to understand what they would expect during and after migration. The migration has four stages:

1. Preflight Check
2. Application Status Check
3. Backup PSO Objects
4. Migrate PSO Objects to Portworx

Preflight Check:

During the preflight check stage, the tool checks for the prerequisites in the cluster. Your cluster should be running PSO 6.0+ and it should be installed with Portworx 2.11+. The Portworx installation should be the operator based installation. If any of these prerequisites are not satisfied, the tool exits giving the appropriate reason.

Application Status Check:

The pso2px tool expects the application pods, which are actively using the PSO PVC's to be scaled down manually. As the underlying storage driver of the StorageClass object is getting changed during the migration, the applications which are actively using those PVC's needs to be scaled down. This is how Kubernetes expects during the storage driver migration.

If the migration is initiated for a particular namespace, then the application pods (Single pod applications, Deployments, StatefulSets, DaemonSets, Jobs, CronJobs) actively using the PSO PVC's in the same namespace, needs to be scaled down.

If the applications are not scaled down, then the pso2px tool, when run, will present a summary of the applications which are actively using the PSO PVCs.

Backup PSO Objects:

After ensuring that no application pods are actively using the PVC's in the namespace where the migration is initiated, the tool will backup the existing PSO Storage Class, PVCs and PVs. These backups will serve as an option to recreate the PSO objects if the migration runs into any issues.

Migrate PSO Objects to Portworx:

After successful backup of existing PSO objects, the tool starts the migration for the given namespace. When the migration starts,

- Deletes the PSO StorageClass
- Creates the Portworx StorageClass using the PSO StorageClass, as its source. Except for some StorageClass attributes like, the provisioner, backend, other attributes are retained for the Portworx StorageClass.
- Before deleting the PVC, the tool checks the volume reclaim policy of the PVC. If the policy is set to Delete, then the tool changes it to "Retain". The idea here is not to delete the backend volume during migration. If the policy is already set to Retain, it is retained as is.
- After updating the volume reclaim policy, the PSO PVC is deleted.
- Followed by that, the PSO PV is also deleted.
- Portworx PV is created using the PSO PV as its source. Except for some PV attributes like spec→csi→driver and spec→csi→volumeAttributes→backend, other attributes are retained for Portworx PV.
- Portworx PVC is created using the PSO PVC as its source.

This process is repeated for all the PSO PVC & PV in the given namespace. When no namespace is given, this process gets repeated for all the PSO PVC & PV across all the namespaces, one by one.

Best Practices

- Migrate namespace by namespace. Doing so, the applications running on other namespaces can continue to be online while migration is in progress in the chosen namespace

- When migrating applications by specifying a namespace, try to have all the applications using the same Pure storage class