Portworx

svcaccount-px-backup-proc-rvw-doc-proj-01

Updated: 2/19/2024

[Concepts 4](#_Toc159220789)

[Binding 4](#_Toc159220790)

[ClusterRole 4](#_Toc159220791)

[Role 4](#_Toc159220792)

[Procedures Overview 5](#_Toc159220793)

[## overview, Design Goals 5](#_Toc159220794)

[All namespaces – PX-Backup ServiceAccount, ClusterRole, ClusterRoleBinding; ConfigMap Role, RoleBinding; ServiceAccount Secret Token 6](#_Toc159220795)

[sa-px-backup-all.yaml 6](#_Toc159220796)

[sa-px-backup - PX-Backup ServiceAccount 6](#_Toc159220797)

[px-backup-all-clusterrole 6](#_Toc159220798)

[px-backup-all-clusterrolebinding 6](#_Toc159220799)

[ks-configmap-role 6](#_Toc159220800)

[ks-configmap-rolebinding 6](#_Toc159220801)

[sa-px-backup-secret 6](#_Toc159220802)

[config\_sa-px-backup - ; ServiceAccount Kubeconfig 8](#_Toc159220803)

[get service account sa-px-backup token 8](#_Toc159220804)

[this will be inserted in config\_sa-px-backup users.name.token 8](#_Toc159220805)

[get service account sa-px-backup certificate 8](#_Toc159220806)

[config\_sa-px-backup 10](#_Toc159220807)

[Copy existing admin kubeconfig to what will be svc account kubeconfig - cp config\_dr-apps-01 config\_sa-px-backup 12](#_Toc159220808)

[Copy/paste all to/from vanilla text editor with no Word Wrap 12](#_Toc159220809)

[Administrator kubeconfig - /etc/kubernetes/admin.conf – test for cluster access, default permissions 13](#_Toc159220810)

[Overview 13](#_Toc159220811)

[sudo kubeadm certs check-expiration - check expiration on cluster certificates including admin.conf 13](#_Toc159220812)

[Run from control plane node with elevated privileges 13](#_Toc159220813)

[Logs 14](#_Toc159220814)

[Troubleshooting 15](#_Toc159220815)

[Resources 16](#_Toc159220816)

[Adding a Service Account Authentication Token to a Kubeconfig File 16](#_Toc159220817)

[app-with-px-backup – Tom Lapp GitHub 16](#_Toc159220818)

[Certificate Management with kubeadm 16](#_Toc159220819)

[Configure Service Accounts for Pods | Kubernetes 16](#_Toc159220820)

[Grant access to a namespace for non-admin user – PX-Backup 16](#_Toc159220821)

[How To Create Kubernetes Service Account For API Access 16](#_Toc159220822)

[How to regenerate Service Account tokens in Kubernetes | Support | SUSE 16](#_Toc159220823)

[How to Use ClusterRoleBinding With A ServiceAccount in All Namespaces (or a few) 16](#_Toc159220824)

[jwt - Set Expiry of Kubernetes service tokens - Stack Overflow 16](#_Toc159220825)

[Managing Service Accounts 16](#_Toc159220826)

[TokenRequest API 16](#_Toc159220827)

# Concepts

## Binding

Grants permissions to a Role in a namespace or a ClusterRole in all namespaces

## ClusterRole

Sets permissions as a non-namespaced resource

ClusterRole + RoleBinding = particular namespace

ClusterRole + ClusterRoleBinding = all namespaces

## Role

Sets permissions within a particular namespace

Role + RoleBinding = same namespace

# Procedures Overview

## ## overview, Design Goals

Specify ServiceAccount sa-px-backup to protect clusters with PX-Backup

sa-px-backup, px-backup-all-clusterrole, px-backup-all-clusterrolebinding

Cluster-wide access to all namespaces, current, future

Managed, limited lifecycle

Manageable method(s) to end, begin validity

sa-px-backup-default, px-backup-default-clusterrole, px-backup-default-rolebinding

Access to specific namespace only

Managed, limited lifecycle

# All namespaces – PX-Backup ServiceAccount, ClusterRole, ClusterRoleBinding; ConfigMap Role, RoleBinding; ServiceAccount Secret Token

## sa-px-backup-all.yaml

## sa-px-backup - PX-Backup ServiceAccount

## px-backup-all-clusterrole

## px-backup-all-clusterrolebinding

## ks-configmap-role

## ks-configmap-rolebinding

## sa-px-backup-secret

---

apiVersion: v1

kind: ServiceAccount

metadata:

name: sa-px-backup

namespace: default

---

apiVersion: rbac.authorization.k8s.io/v1

kind: ClusterRole

metadata:

name: px-backup-all-clusterrole

rules:

- apiGroups: ["\*"]

resources: ["namespaces", "clusterrolebindings", "persistentvolumes", "clusterroles", "nodes", "storageclasses"]

verbs: ["get", "list", "watch", "create"]

- apiGroups: ["stork.libopenstorage.org"]

resources: ["schedulepolicies"]

verbs: ["create", "delete", "deletecollection", "get", "list", "patch", "update", "watch"]

- apiGroups: ["stork.libopenstorage.org"] # for controller support

resources: ["\*"]

verbs: ["list", "get", "watch"]

- apiGroups: ["snapshot.storage.k8s.io"]

resources: ["volumesnapshotclasses"]

verbs: ["get", "list"]

- apiGroups: ["", "extensions", "apps"]

resources: ["\*"]

verbs: ["\*"]

- apiGroups: ["batch"]

resources:

- jobs

- cronjobs

verbs: ["\*"]

- apiGroups: ["stork.libopenstorage.org"]

resources: ["\*"]

verbs: ["create", "delete", "deletecollection", "get", "list", "patch", "update", "watch"]

---

apiVersion: rbac.authorization.k8s.io/v1

kind: ClusterRoleBinding

metadata:

name: px-backup-all-clusterrolebinding

subjects:

- kind: ServiceAccount

name: sa-px-backup

namespace: default

roleRef:

apiGroup: rbac.authorization.k8s.io

kind: ClusterRole

name: px-backup-all-clusterrole

---

apiVersion: rbac.authorization.k8s.io/v1

kind: Role

metadata:

name: ks-configmap-role

namespace: kube-system

rules:

- apiGroups: [""]

resources: ["configmaps"]

verbs: ["get", "list"]

---

apiVersion: rbac.authorization.k8s.io/v1

kind: RoleBinding

metadata:

name: ks-configmap-rolebinding

namespace: kube-system

subjects:

- kind: ServiceAccount

name: sa-px-backup

namespace: default

roleRef:

apiGroup: rbac.authorization.k8s.io

kind: Role

name: ks-configmap-role

---

apiVersion: v1

kind: Secret

type: kubernetes.io/service-account-token

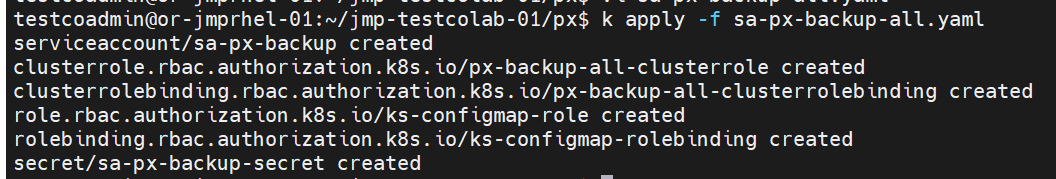
metadata:

name: sa-px-backup-secret

namespace: default

annotations:

kubernetes.io/service-account.name: sa-px-backup



# config\_sa-px-backup - ; ServiceAccount Kubeconfig

## get service account sa-px-backup token

## this will be inserted in config\_sa-px-backup users.name.token

kubectl get secret sa-px-backup-secret -n default -o "jsonpath={.data.token}" | base64 -d

eyJhbGciOiJSUzI1NiIsImtpZCI6IkI2cVhuM25IVnZMamlTX0hwOWRKNXR3UmhCSXRxel9uSVNSMjZITTExc28ifQ.eyJpc3MiOiJrdWJlcm5ldGVzL3NlcnZpY2VhY2NvdW50Iiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9uYW1lc3BhY2UiOiJkZWZhdWx0Iiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9zZWNyZXQubmFtZSI6InNhLXB4LWJhY2t1cC1zZWNyZXQiLCJrdWJlcm5ldGVzLmlvL3NlcnZpY2VhY2NvdW50L3NlcnZpY2UtYWNjb3VudC5uYW1lIjoic2EtcHgtYmFja3VwIiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9zZXJ2aWNlLWFjY291bnQudWlkIjoiZmJkYTkzYWYtYzY4OS00OGIzLWI2MzUtMDhlZThlMTJkNzZjIiwic3ViIjoic3lzdGVtOnNlcnZpY2VhY2NvdW50OmRlZmF1bHQ6c2EtcHgtYmFja3VwIn0.JCLKk0NKOBBI9Zs7O5pnutEM8lheOxpvWS\_jBFLuVyUmyGwhMRis0PI5L1iiEk0G58gVgW3sLXLsSwMEwDarFyW68pVXgaRDbzOcS9PxX4UEkJS4u10eDRlRjYhhCiUYM-R95sYSL5GUysRdtRw13HiOJhRiAilKDoKlHZfKh9TdRROBODbCKDGdRNA8aB25AvNNpZ6Tgv7zcZOPXAd-8DJ3XX3HKypULdp09eNsdRvuTT4eqsFvIVgAfx7LFqWTHTDK7ISbXPZdlrp6xVuY7W5DjonW4CuNuVmhezW2ghkNJbN1SZ4eLQiTRORaquPHObs8\_B345LAO26cLVeS7iA

## get service account sa-px-backup certificate

kubectl get secret sa-px-backup-secret -n default -o "jsonpath={.data['ca\.crt']}"



## config\_sa-px-backup

apiVersion: v1

clusters:

- cluster:

certificate-authority-data: 

server: https://10.2.7.131:6443

name: dr-apps-01-sa-px-backup

contexts:

- context:

cluster: dr-apps-01

user: sa-px-backup

name: dr-apps-01-sa-px-backup

kind: Config

preferences: {}

users:

- name: sa-px-backup

user:

client-key-data: 

token: eyJhbGciOiJSUzI1NiIsImtpZCI6IkI2cVhuM25IVnZMamlTX0hwOWRKNXR3UmhCSXRxel9uSVNSMjZITTExc28ifQ.eyJpc3MiOiJrdWJlcm5ldGVzL3NlcnZpY2VhY2NvdW50Iiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9uYW1lc3BhY2UiOiJkZWZhdWx0Iiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9zZWNyZXQubmFtZSI6InNhLXB4LWJhY2t1cC1zZWNyZXQiLCJrdWJlcm5ldGVzLmlvL3NlcnZpY2VhY2NvdW50L3NlcnZpY2UtYWNjb3VudC5uYW1lIjoic2EtcHgtYmFja3VwIiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9zZXJ2aWNlLWFjY291bnQudWlkIjoiZmJkYTkzYWYtYzY4OS00OGIzLWI2MzUtMDhlZThlMTJkNzZjIiwic3ViIjoic3lzdGVtOnNlcnZpY2VhY2NvdW50OmRlZmF1bHQ6c2EtcHgtYmFja3VwIn0.JCLKk0NKOBBI9Zs7O5pnutEM8lheOxpvWS\_jBFLuVyUmyGwhMRis0PI5L1iiEk0G58gVgW3sLXLsSwMEwDarFyW68pVXgaRDbzOcS9PxX4UEkJS4u10eDRlRjYhhCiUYM-R95sYSL5GUysRdtRw13HiOJhRiAilKDoKlHZfKh9TdRROBODbCKDGdRNA8aB25AvNNpZ6Tgv7zcZOPXAd-8DJ3XX3HKypULdp09eNsdRvuTT4eqsFvIVgAfx7LFqWTHTDK7ISbXPZdlrp6xVuY7W5DjonW4CuNuVmhezW2ghkNJbN1SZ4eLQiTRORaquPHObs8\_B345LAO26cLVeS7iA

## Copy existing admin kubeconfig to what will be svc account kubeconfig - cp config\_dr-apps-01 config\_sa-px-backup



## Copy/paste all to/from vanilla text editor with no Word Wrap

Changes to config\_sa-px-backup

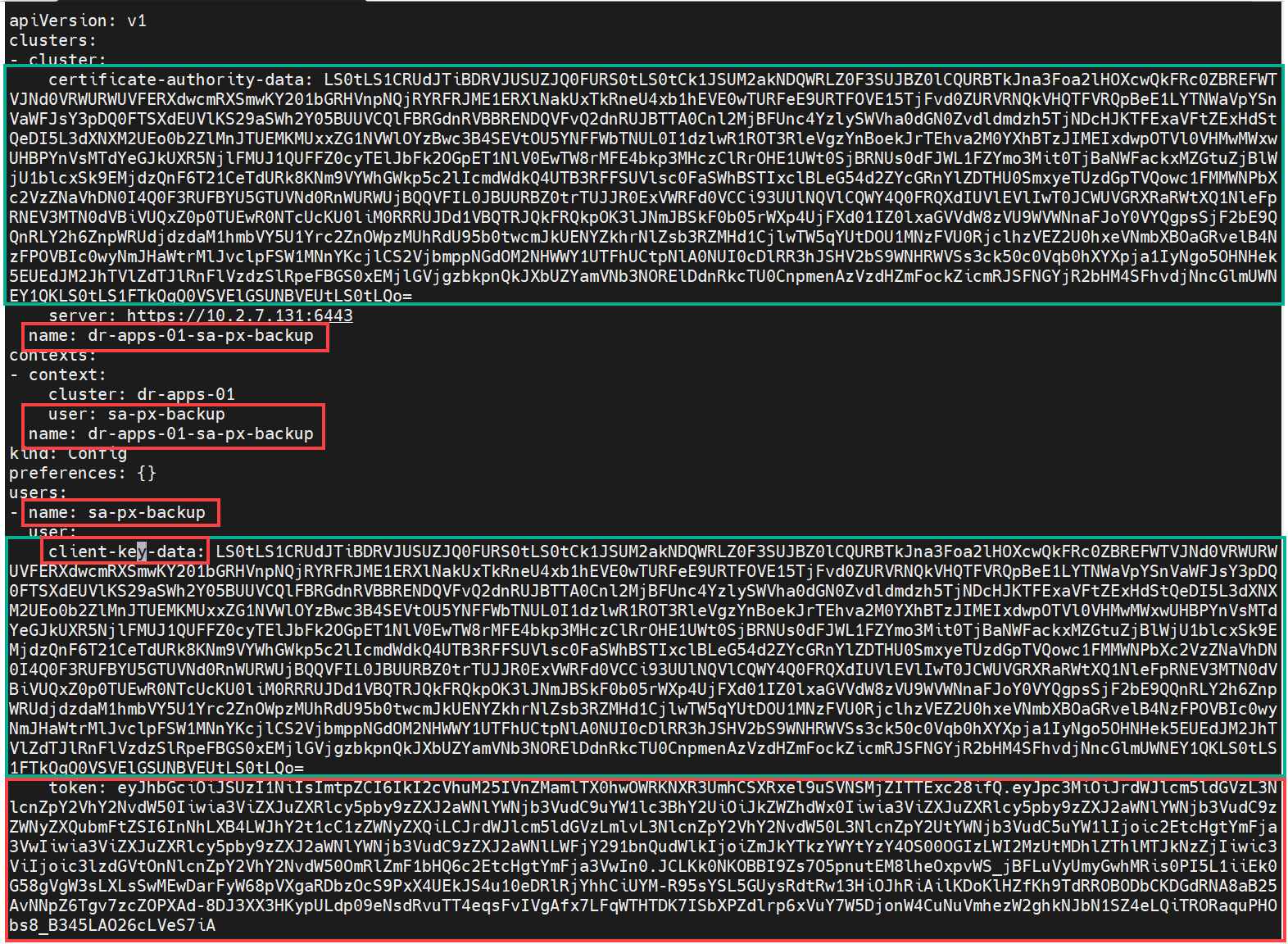
Clusters.cluster.certificate-authority-data – same as admin kubeconfig

Clusters.name append with sa-px-backup

Users.name append with sa-px-backup

Contexts.context.user = users.name

Users.user.**client.key-data** – same as admin kubeconfig



# Administrator kubeconfig - /etc/kubernetes/admin.conf – test for cluster access, default permissions

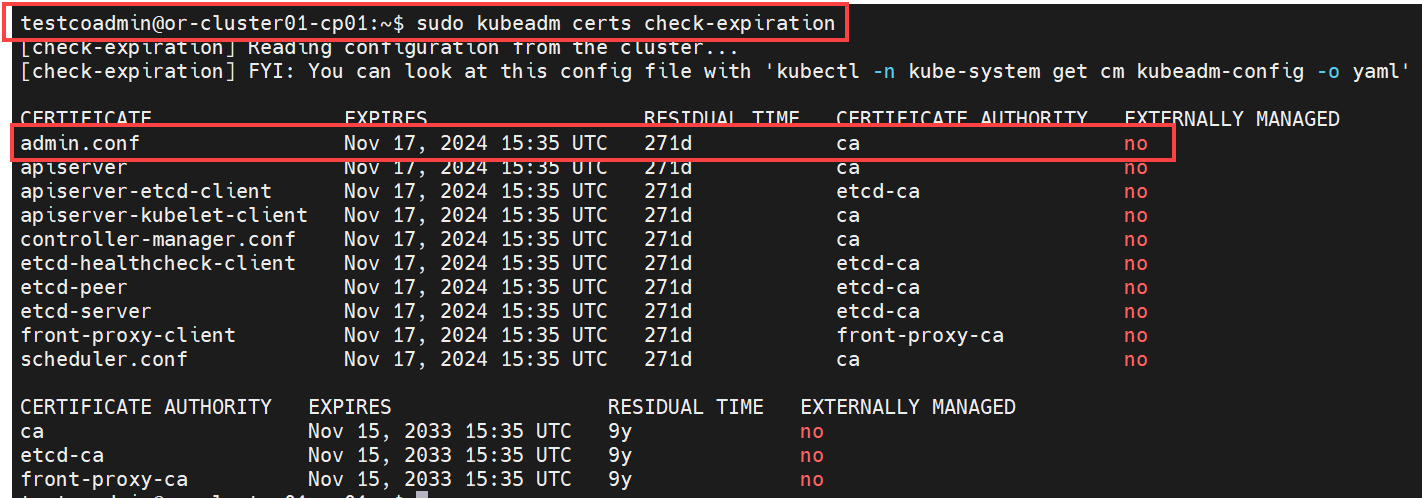
## Overview

* The default administrator kubeconfig for the cluster (etc/kubernetes/admin.conf) may be used for basic troubleshooting for cluster access confirmation etc.
* It is not a best practice to use this kubeconfig in production

Instead use the procedures in this document to create and use a least privileges kubeconfig

## sudo kubeadm certs check-expiration - check expiration on cluster certificates including admin.conf

## Run from control plane node with elevated privileges



# Logs

# Troubleshooting

# Resources

## Adding a Service Account Authentication Token to a Kubeconfig File

https://docs.oracle.com/en-us/iaas/Content/ContEng/Tasks/contengaddingserviceaccttoken.htm

## app-with-px-backup – Tom Lapp GitHub

https://github.com/tmlapp/app-with-px-backup

## Certificate Management with kubeadm

https://kubernetes.io/docs/tasks/administer-cluster/kubeadm/kubeadm-certs/

## Configure Service Accounts for Pods | Kubernetes

https://kubernetes.io/docs/tasks/configure-pod-container/configure-service-account/#service-account-token-volume-projection

## Grant access to a namespace for non-admin user – PX-Backup

<https://docs.portworx.com/portworx-backup-on-prem/configure/namespace-access.html>

## How To Create Kubernetes Service Account For API Access

https://devopscube.com/kubernetes-api-access-service-account/

## How to regenerate Service Account tokens in Kubernetes | Support | SUSE

https://www.suse.com/support/kb/doc/?id=000020019

## How to Use ClusterRoleBinding With A ServiceAccount in All Namespaces (or a few)

<https://travis.media/clusterrolebinding-with-service-account-all-namespaces/>

## jwt - Set Expiry of Kubernetes service tokens - Stack Overflow

https://stackoverflow.com/questions/62791028/set-expiry-of-kubernetes-service-tokens

## Managing Service Accounts

https://kubernetes.io/docs/reference/access-authn-authz/service-accounts-admin/

## TokenRequest API

https://kubernetes.io/docs/reference/access-authn-authz/service-accounts-admin/#tokenrequest-api