Velero Kubernetes backup

START HERE

`velero` **`install --provider`** *`<p>`* **`--bucket`** *`<b>`* **`--plugins`** *`<img>`*
Install server

`velero` **`backup logs`** *`<name>`*
Read backup logs

`velero` **`backup create`** *`<name>`*
Back up the cluster

`velero` **`backup-location get`**
Is storage Available?

`velero` **`backup get`**
List backups

`velero` **`backup describe`** *`<name>`* **`--details`**
Inspect a backup

`velero` **`restore create --from-backup`** *`<name>`*
Restore it

`velero` **`schedule create`** *`<name>`* **`--schedule="0 3 * * *"`**
Back up daily

velero CLI creates CRs → `Backup CR` → **Velero server** deployment in velero ns → `upload` → **Object store** BackupStorageLocation → `snapshot` → **Volumes** VolumeSnapshotLocation

CONCEPTS

Velero stores backups as object-store tarballs; the CLI just creates Backup, Restore and Schedule objects the in-cluster server acts on.

Backups are namespaced to the velero namespace; use -n <ns> if you installed elsewhere.

A restore never overwrites existing objects by default; they are skipped with a warning.

INSTALL

`velero` **`install --provider`** *`<provider>`* **`--bucket`** *`<bucket>`* **`--secret-file`** *`<file>`* **`--plugins`** *`<image>`*
Install server components
The --plugins image is required, e.g. the AWS plugin; add --backup-location-config region=<r> for S3-style stores.

`velero` **`install --no-secret`**
Install using workload identity

`velero` **`install --use-node-agent`**
Enable file system backup

`velero` **`install --default-volumes-to-fs-backup`**
Use fs backup for all volumes

INSTALL (cont.)

`velero` **`install --dry-run -o yaml`**
Render install manifests only

`velero` **`uninstall`**
Remove what install created

`velero` **`version`**
Show client and server version

BACKUPS

`velero` **`backup create`** *`<name>`*
Back up all namespaces

`velero` **`backup create`** *`<name>`* **`--include-namespaces`** *`<ns>`*
Back up chosen namespaces

`velero` **`backup create`** *`<name>`* **`--exclude-namespaces`** *`<ns>`*
Skip namespaces

`velero` **`backup create`** *`<name>`* **`--selector`** *`<k>=<v>`*
Back up by label

`velero` **`backup create`** *`<name>`* **`--ttl`** *`<duration>`*
Set retention (default 720h)

`velero` **`backup create`** *`<name>`* **`--wait`**
Block until finished

`velero` **`backup create`** *`<name>`* **`--snapshot-volumes=false`**
Skip volume snapshots

BACKUPS (cont.)

`velero` **`backup create`** *`<name>`* **`--default-volumes-to-fs-backup`**
Copy volume data with fs backup

`velero` **`backup create --from-schedule`** *`<schedule>`*
Run a schedule's backup now

`velero` **`backup create`** *`<name>`* **`--storage-location`** *`<bsl>`*
Pick a storage location

`velero` **`backup get`**
List backups and status

`velero` **`backup describe`** *`<name>`* **`--details`**
Show contents and volume results

`velero` **`backup logs`** *`<name>`*
Show backup log

`velero` **`backup delete`** *`<name>`*
Delete backup and its data
kubectl delete backup only removes the CR, not the stored data.

RESTORES

`velero` **`restore create --from-backup`** *`<name>`*
Restore a backup

RESTORES (cont.)

`velero` **`restore create --from-schedule`** *`<schedule>`*
Restore latest scheduled backup

`velero` **`restore create`** *`<name>`* **`--from-backup`** *`<backup>`*
Restore with a chosen name

`velero` **`restore create --from-backup`** *`<name>`* **`--include-namespaces`** *`<ns>`*
Restore only some namespaces

`velero` **`restore create --from-backup`** *`<name>`* **`--include-resources`** *`<res>`*
Restore only some kinds

`velero` **`restore create --from-backup`** *`<name>`* **`--namespace-mappings`** *`<old>:<new>`*
Restore into another namespace

`velero` **`restore create --from-backup`** *`<name>`* **`--existing-resource-policy update`**
Update objects that already exist

`velero` **`restore get`**
List restores

`velero` **`restore describe`** *`<name>`* **`--details`**
Show warnings and errors

RESTORES (cont.)

`velero` **`restore logs`** *`<name>`*
Show restore log

SCHEDULES

`velero` **`schedule create`** *`<name>`* **`--schedule="`***`<cron>`***`"`**
Create a recurring backup
Standard 5-field cron; prefix CRON_TZ=<zone> for a time zone.

`velero` **`schedule create`** *`<name>`* **`--schedule="@every 6h" --ttl`** *`<duration>`*
Schedule with retention

`velero` **`schedule create`** *`<name>`* **`--schedule="`***`<cron>`***`" --include-namespaces`** *`<ns>`*
Schedule a subset

`velero` **`schedule get`**
List schedules

`velero` **`schedule describe`** *`<name>`*
Show schedule and last run

STORAGE LOCATIONS

`velero` **`backup-location get`**
List locations and phase

`velero` **`backup-location create`** *`<name>`* **`--provider`** *`<p>`* **`--bucket`** *`<b>`*
Add a storage location

STORAGE LOCATIONS (cont.)

`velero` **`backup-location set`** *`<name>`* **`--default`**
Make it the default

HOOKS AND VOLUMES

`pre.hook.backup.velero.io/command`
Command run before backing up a pod
Not run in a shell; invoke /bin/sh -c explicitly. Set the command as a JSON array.

`backup.velero.io/backup-volumes=`*`<vol1>`***`,`***`<vol2>`**
Opt a pod's volumes into fs backup

`velero.io/exclude-from-backup=true`
Label that keeps an object out of backups

Annotate with: kubectl -n <ns> annotate pod/<pod> <annotation>=<value>

TROUBLESHOOTING

`kubectl logs deployment/velero -n velero`
Read server logs