eMPATHY.CO

Kubernetes Backup and Migration Strategies with Velero

AGENDA

Use Cases Velero

Velero Components

Backup/Restore Data

Generic Workflow

Demo

Disaster Recovery

Data Migration

Conclusions

USE CASES

K8S BACKUP & MIGRATION STRATEGIES WITH VELERO

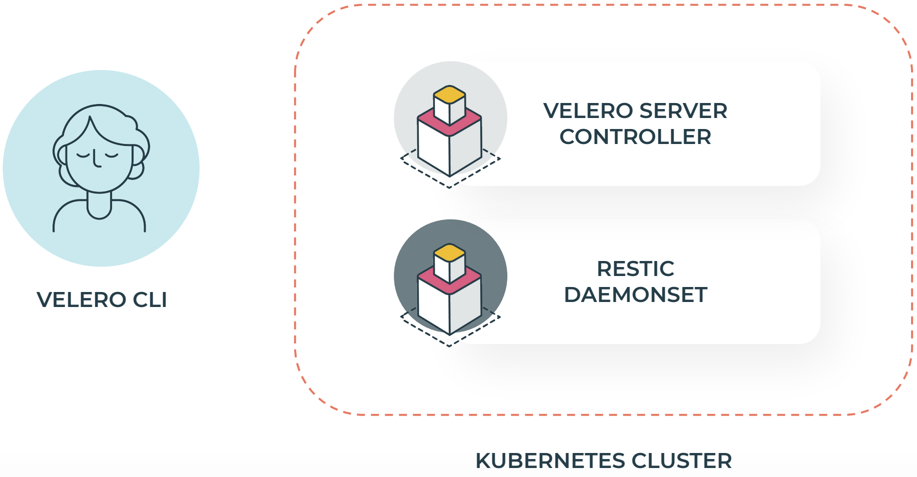
DATA INTEGRITY

DISASTER RECOVERY

DATA MIGRATION

VELERO

COMPONENTS



VELERO CLI

VELERO SERVER CONTROLLER

RESTIC DAEMONSET

KUBERNETES CLUSTER

DATA

ETCD

Kubernetes API Discovery
Backups in cloud object storage

PERSISTENT VOLUMES

Cloud Provider Snapshot APIs
Agnostic File System level backups (Restic)

DATA

FLEXIBILITY

Schedule backups
Filtering by multiple fields

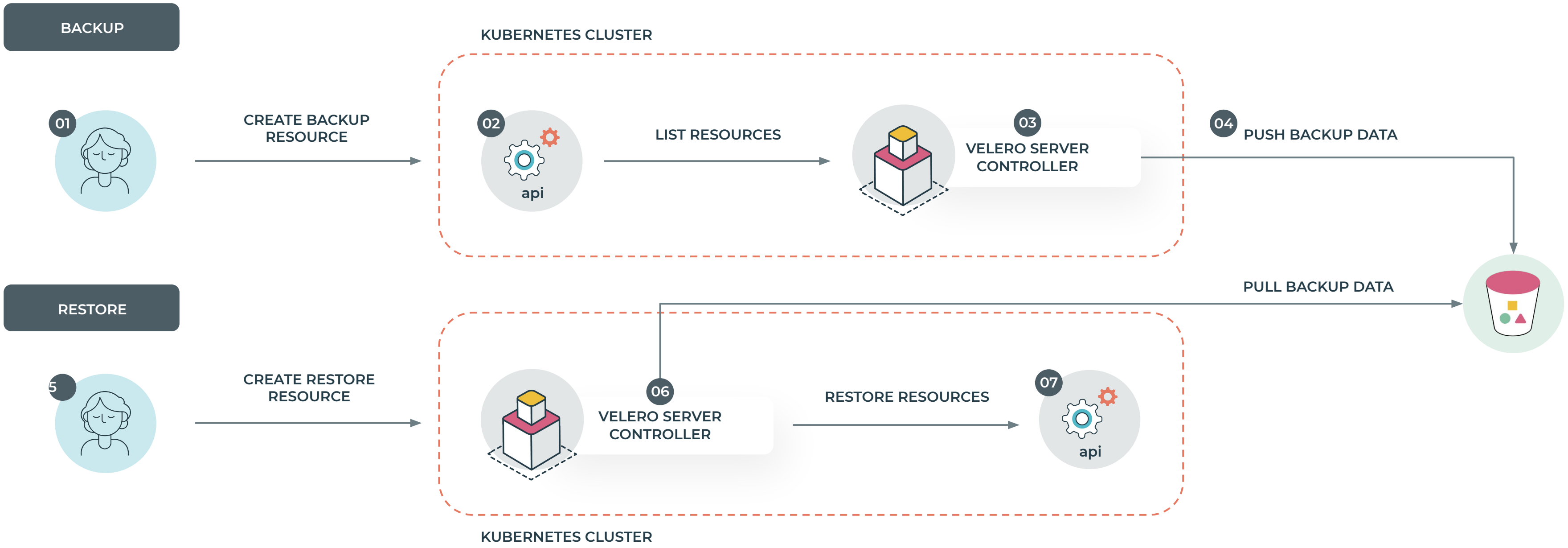
EXTENSIBILITY

Plugins
Hooks

empathy.co

VELERO WORKFLOW

GENERIC DIAGRAM



01 Using the Velero CLI, the user creates a Backup custom resource using the Kubernetes API. The instruction accept filtering by namespace, resource or labelling, for instance `velero backup create cluster --include-namespace ghost --wait`.

02 The Velero Controller notice about the new Backup resource and validated the command.

03 The Velero Controller starts the backup process. It queries the Kubernetes API to backup the resources.

04 The Velero Controller push the backup file to a cloud object storage. Besides, it creates snapshots using the cloud Snapshot API of any existing PV. The backup process finished after upload the file to the cloud object storage.

05 Using the Velero CLI, the user creates a Restore custom restore using the Kubernetes API. The instruction accept filtering by namespace, resource or labelling, for instance to restore from the sample above `velero restore create --from-backup cluster --include-namespace ghost`.

06 The Velero Controller notices about the new Restore resource, validates the command and start pulling the data from the cloud object storage.

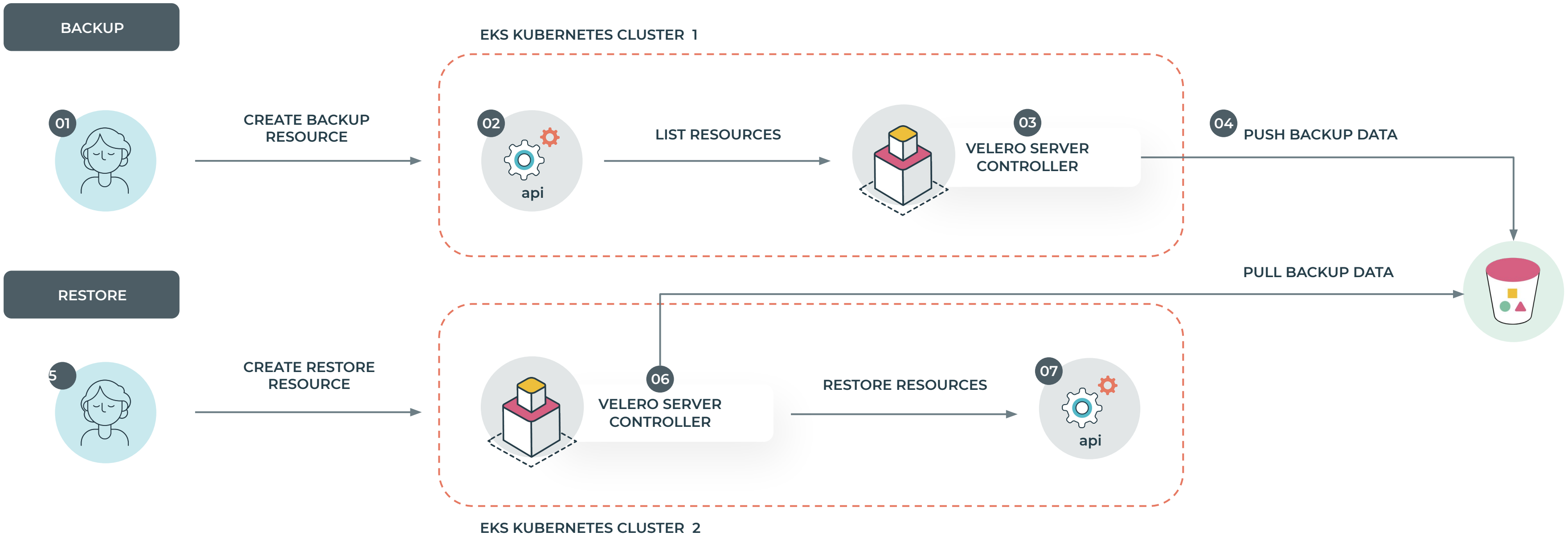
07 The Velero Controller restores the resources querying the Kubernetes API to restore the resources. The restore process finished after deploying the pull data to the Kubernetes API.

DEMO

eMPATHY.CO

VELERO WORKFLOW

DISASTER RECOVERY



BACKUP

EKS KUBERNETES CLUSTER 1

01

CREATE BACKUP RESOURCE

02 api

LIST RESOURCES

03 VELERO SERVER CONTROLLER

04 PUSH BACKUP DATA

PULL BACKUP DATA

RESTORE

5

CREATE RESTORE RESOURCE

06 VELERO SERVER CONTROLLER

RESTORE RESOURCES

07 api

EKS KUBERNETES CLUSTER 2

01 Using the Velero CLI, the user creates a Backup custom resource on EKS Cluster 1 using the Kubernetes API. The instruction accept filtering by namespace, resource or labelling, for instance `velero backup create cluster1 --include-namespace ghost --wait`.

02 The Velero Controller notice about the new Backup resource and validated the command.

03 The Velero Controller starts the backup process. It queries the Kubernetes API to backup the resources.

04 The Velero Controller push the backup file to a cloud object storage. Besides, it creates snapshots using the cloud Snapshot API of any existing PV. The backup process finished after upload the file to the cloud object storage.

05 Using the Velero CLI, the user creates a Restore custom restore on EKS Cluster 2 using the Kubernetes API. The instruction accept filtering by namespace, resource or labelling, for instance to restore from the sample above `velero restore create --from-backup cluster1 --include-namespace ghost`.

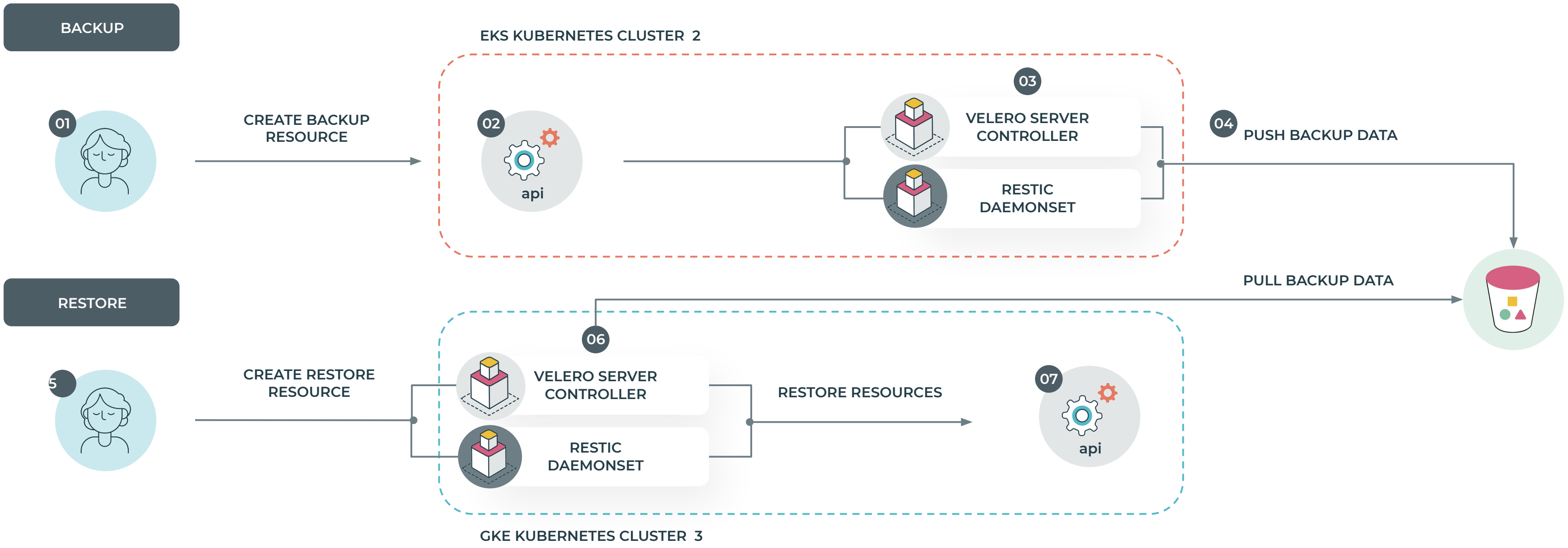
06 The Velero Controller notices about the new Restore resource, validates the command and start pulling the data from the cloud object storage.

07 The Velero Controller restores the resources querying the Kubernetes API to restore the resources. The restore process finished after deploying the pull data to the Kubernetes API. Data from cluster1 has been recovered on cluster2.

eMPATHY.CO

VELERO WORKFLOW

DATA MIGRATION



BACKUP

EKS KUBERNETES CLUSTER 2

01

CREATE BACKUP RESOURCE

02

api

03

VELERO SERVER CONTROLLER

RESTIC DAEMONSET

04

PUSH BACKUP DATA

PULL BACKUP DATA

RESTORE

5

CREATE RESTORE RESOURCE

06

VELERO SERVER CONTROLLER

RESTIC DAEMONSET

RESTORE RESOURCES

07

api

GKE KUBERNETES CLUSTER 3

01 Using the Velero CLI, the user creates a Backup custom resource on EKS Cluster2 using the Kubernetes API. The instruction accept filtering by namespace, resource or labelling, for instance `velero backup create cluster2 --include-namespace ghost --wait`.

02 The Velero Controller notice about the new Backup resource and validated the command.

03 The Velero Controller starts the backup process. It queries the Kubernetes API to backup the resources. Velero ensures a restic repository exists for the pod's namespace. Velero creates a PodVolumeBackup custom resource per volume listed in the pod annotation. The main Velero process waits for the PodVolumeBackup resources to complete or fail.

04 The Velero Controller push the backup file to a cloud object storage. This file would be used for restores.

05 Using the Velero CLI, the user creates a Restore custom restore on GKE cluster3 using the Kubernetes API. The instruction accept filtering by namespace, resource or labelling, for instance to restore from the sample above `velero restore create --from-backup cluster2 --include-namespace ghost`.

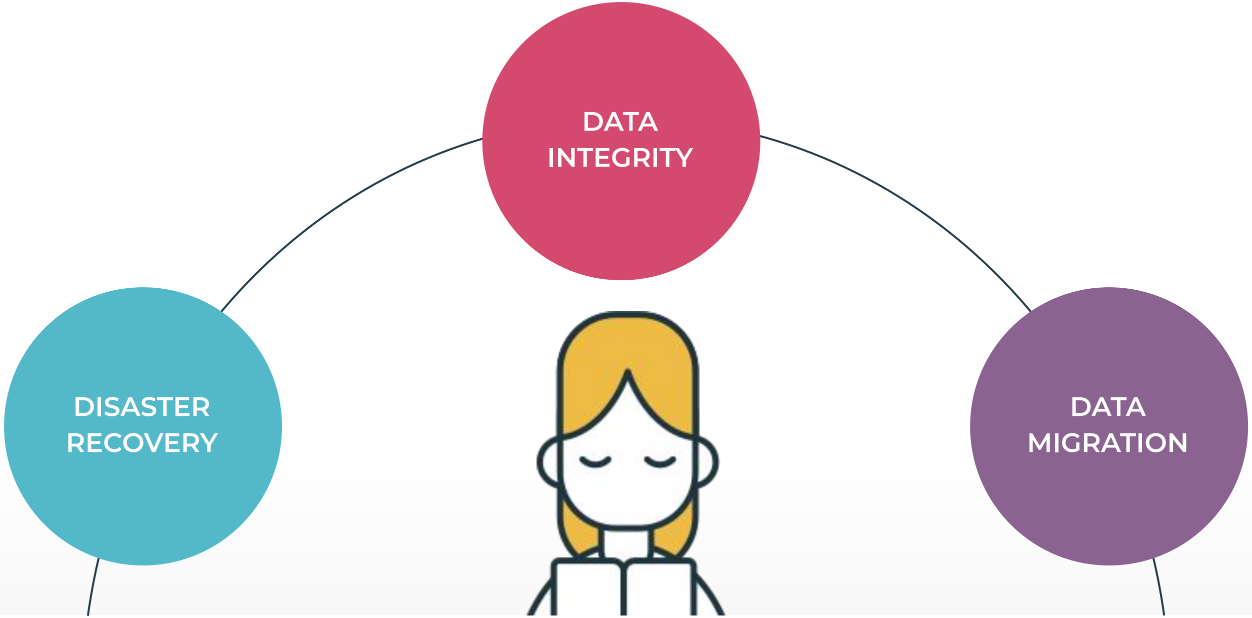
06 The Velero Controller notices about the new Restore resource, validates the command and start pulling the data from the cloud object storage. Velero creates a PodVolumeRestore custom resource for each volume to be restored in the pod.

07 The Velero Controller waits for each PodVolumeRestore resource to complete or fail. It restores the resources querying the Kubernetes API to restore the resources. The restore process finished after deploying the pull data to the Kubernetes API.

SHOWTIME

CONCLUSIONS

K8S BACKUP & MIGRATION STRATEGIES WITH VELERO



Get more info in

ramiroalvfer.dev

github.com/empathyco/platform-velero-showcases