What critical data should be included in a Kubernetes backup strategy?

- ✅ What Needs to Be Backed Up?

1. **Etcd Database** (the brain of the cluster)
    - Stores **all cluster state**: deployments, secrets, configmaps, etc.
2. **Persistent Volumes (PVs)** if your apps store data
3. **Kubernetes YAML Manifests** (your application specs)

📦 Backup Methods

1. Backup etcd (Control Plane only)

If you have access to the control plane (self-managed clusters):

```
bash
---------
ETCDCTL_API=3 etcdctl \
  --endpoints=https://127.0.0.1:2379 \
  --cacert=/etc/kubernetes/pki/etcd/ca.crt \
  --cert=/etc/kubernetes/pki/etcd/server.crt \
  --key=/etc/kubernetes/pki/etcd/server.key \
  snapshot save /backup/etcd-snapshot.db
```

✅ This creates a snapshot of the cluster state.

💡 What is `etcdctl`?

- `etcdctl` is the **command-line tool** for interacting with an **etcd** database.
- Think of it like the `kubectl` for Kubernetes, but this one is for etcd specifically.

💡 What is `ETCDCTL_API=3`?

- It's an **environment variable** that tells the `etcdctl` command to use **API version 3**.
- etcd has two main APIs (v2 and v3), and v3 is the current and recommended version.

- Without setting this, some commands may fail or behave unexpectedly.

---

2. Use Velero (for cluster + volume backups)

Velero is a powerful, open-source backup tool.

```
bash
--------
# Install Velero (example with AWS backend)
velero install \
  --provider aws \
  --bucket my-backup-bucket \
  --plugins velero/velero-plugin-for-aws:v1.5.0 \
  --backup-location-config region=us-east-1

# Backup all resources
velero backup create full-backup --include-namespaces '*'

# View backups
velero backup get
```

You can even schedule recurring backups!

---

3. Export YAMLs (basic method)

For simple or small clusters, export manifests:

```
bash
--------
kubectl get all --all-namespaces -o yaml > all-resources.yaml
kubectl get configmap,secrets --all-namespaces -o yaml > configs.yaml
```

Later, you can restore:

```
bash
--------
kubectl apply -f all-resources.yaml
```

🔄 Restore Methods

1. Restore etcd snapshot (self-managed)

```
bash
--------
ETCDCTL_API=3 etcdctl snapshot restore /backup/etcd-snapshot.db \
  --data-dir=/var/lib/etcd-from-backup
```

Update your Kubernetes config to point to the new etcd data dir.

---

2. Velero Restore

```
bash
```

```
--------
velero restore create --from-backup full-backup
```

---

3. Re-apply YAML files

```
bash
--------
kubectl apply -f all-resources.yaml
```

This brings back your deployments and services.

🧠 Tips

- **Automate backups** (cron jobs, Velero schedules)
- **Test restores** regularly (especially on staging)
- Use **external storage** (S3, GCS) for backup reliability
- Back up **secrets and configmaps** too!

Amazon EKS (Elastic Kubernetes Service):

🔄 How to Backup and Restore EKS Cluster

Backup:

- Use **Velero** to back up Kubernetes resources and EBS volumes to **Amazon S3**.
- Optionally, use **AWS Backup** for EBS snapshots.

Restore:

- Use **Velero restore** to recreate resources from S3.
- Make sure IAM roles and S3 permissions are properly set up.

Which of the following is the best combination to fully recover a Kubernetes cluster after a disaster?

A) Pod logs, kubelet binaries, ingress rules
B) Etcd backup, Persistent Volumes, and YAML manifests
C) Node IPs, load balancer settings, and Helm charts
D) Docker images, CRDs, and container logs

Correct Answer: B) Etcd backup, Persistent Volumes, and YAML manifests

If the etcd database is lost and no backup is available, what is the most likely consequence?

A) Nodes will reboot automatically
B) Only network policies are lost
C) The cluster will lose all its configuration and state
D) Persistent Volumes will be unaffected

Correct Answer: C) The cluster will lose all its configuration and state