# A 101 Session into K8s Storage

**Adam Mazouz**

Cloud Solutions Engineer @ Pure Storage

**Geoff Burke**

Senior Cloud Solutions Architect @ Tsunati

# Agenda

01 Why does K8s storage matter?

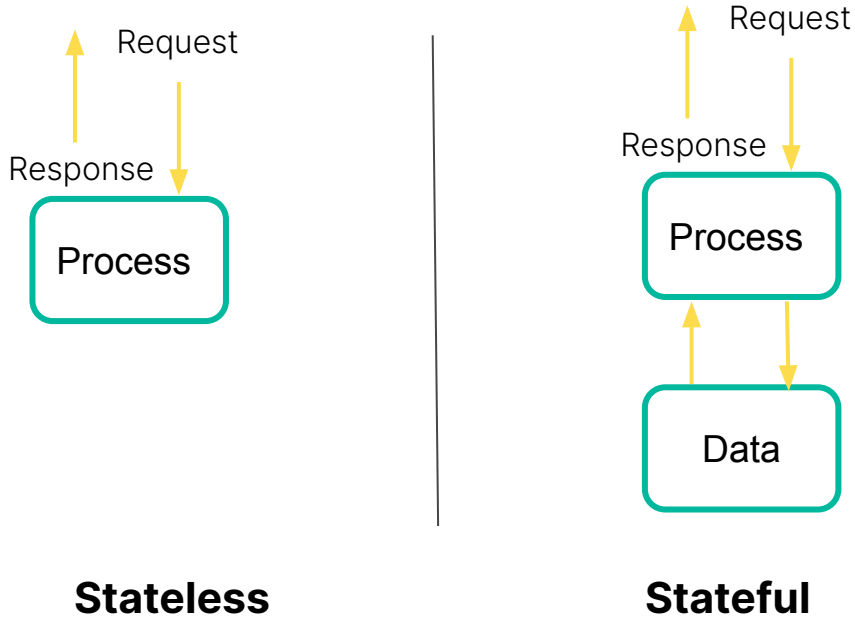
02 Key Terminologies

03 Demo </> Ephemeral Volume

04 Demo </> Persistent Volume

05 Where to Learn and Practice

# Why does K8s Storage Matter?

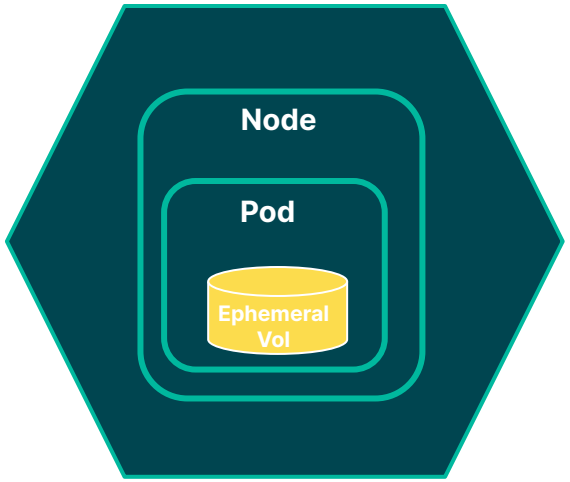



**Stateless** **Stateful**

# Glance at Stateful Applications

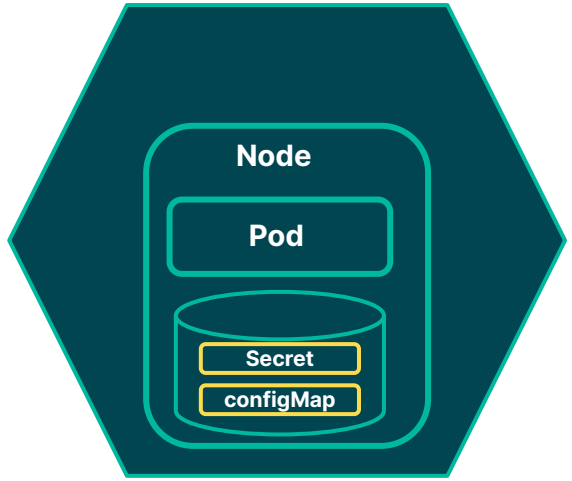
- Running A Stateful Application Outside of Kubernetes (VM, Bare metal)
  - **Pros:** No refactoring or re-architecture for existing stateful applications.
  - **Cons:** Duplicate work by setting up a parallel software workflow outside Kubernetes.
- Running A Stateful Workload as a Cloud Services
  - **Pros:** Easy to setup. Maintained by cloud service, scale up elastically.
  - **Cons:** Comes with cost, lack customization, vendor lock-in, not as great performance and latency properties.
- Running your Stateful Workload Inside Kubernetes
  - **Pros:** Greatest flexibility and operating efficiency in the long term.
  - **Cons:** Most difficult to implement.

# Kubernetes Volumes and Storage Types



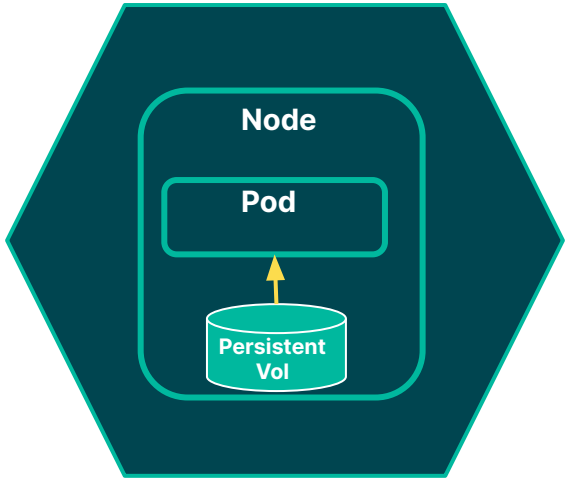

## Ephemeral Volumes

Ephemeral volumes will be destroyed when a pod is destroyed.




## Projected Volumes

Projected volume maps several existing volume sources into the same directory.

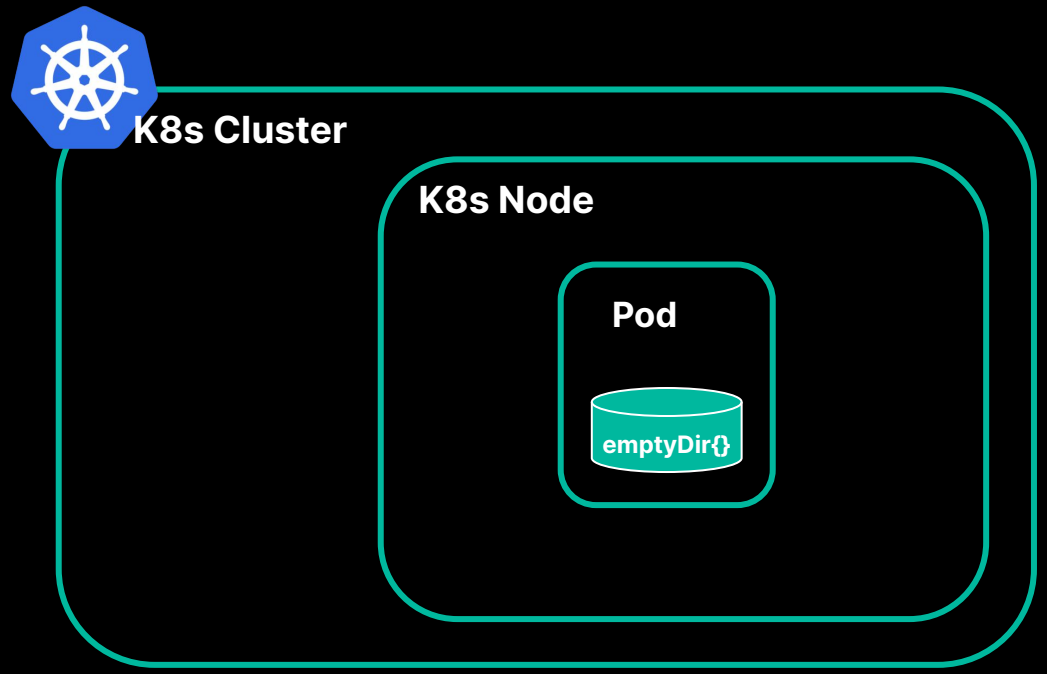



## Persistent Volumes

Persistent volumes exist beyond the lifetime of a pod.

# Demo-1 ...

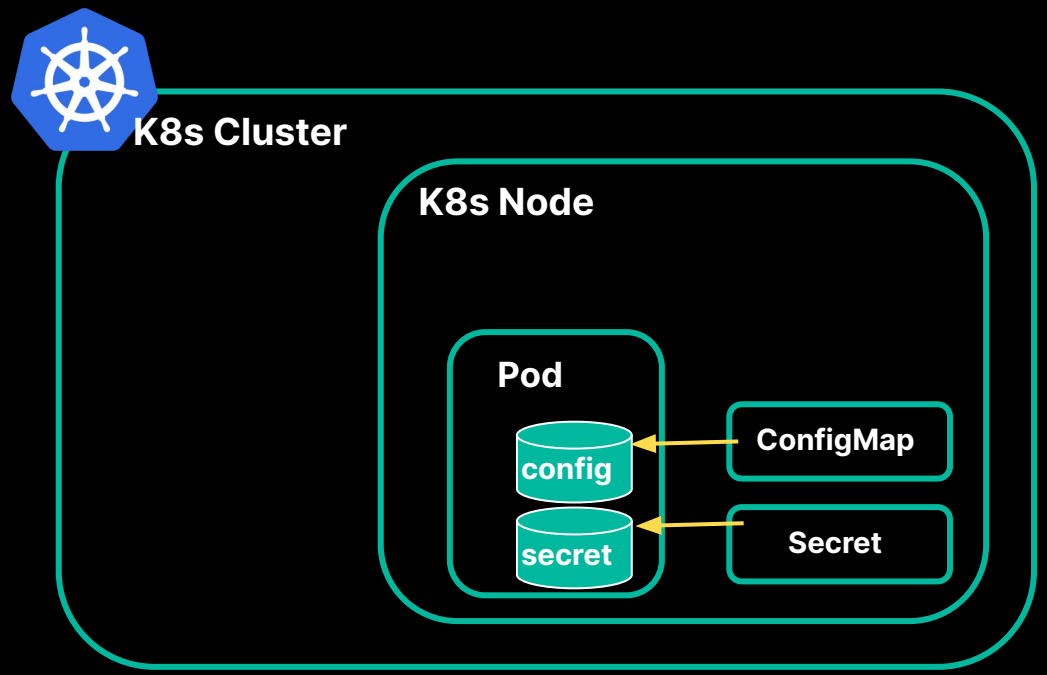
1. Create Pod
2. Restrat the Pod and Validate the Data Exist




Kubernetes.io
PV Example

# Demo-2 ...

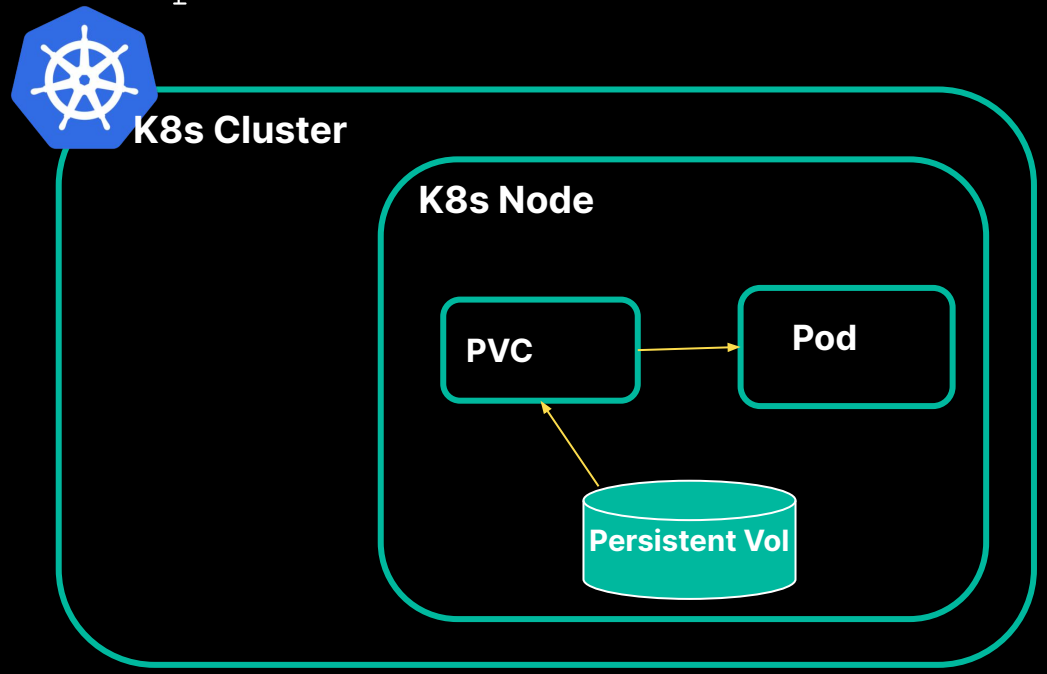
1. Create Secret, and ConfigMap
2. Create Pod.
3. Validate projected config objects after Pod restarted/replaced




Kubernetes.io
PV Example

# Demo-3 ...

1. Create PersistentVolume
2. Create PersistentVolumeClaim
3. Create Pod
4. Validate data persist after Pod restarted/replaced




Kubernetes.io
PV Example