

What's New in Kubernetes 1.14



CLOUD NATIVE
COMPUTING FOUNDATION

Presenters



Claire Laurence

1.14 Enhancements Lead



Michael Michael

SIG-Windows



Patrick Lang

SIG-Windows



Michelle Au

SIG-Storage



Kaitlyn Barnard

1.14 Communications



Agenda

1.14 Enhancements Overview

Persistent Local Volumes

Production-level support for Windows Nodes

Q&A



1.14 Enhancements

Overview

- 31 total enhancements tracked in 1.14
 - 10 Stable Enhancements
 - 12 Graduating to Beta
 - 9 Introduced Alpha features



Highlights

Updated Plugin Mechanism for Kubectl

- Allows users to extend the functionality of kubectl by creating and publishing their own custom kubectl subcommands,



Integrate Kustomize into Kubectl

- Introduces a declarative workflows for users to access kubectl commands



Pod Readiness Gates

- This enhancement help users know when their Pod is ready to serve traffic
- Adds additional indicators of PodStatus
 - PodStatus will now take into account container readiness and external feedback



Pid Limiting

- Allows users to control pod-to-pod isolation and node-to-pode isolation,
- Prevents pid exhaustion
- Ensures that Pids have a limited impact to other workloads



Add Pod Priority and Preemption

- Enables users to set priority of Pods where higher priority pods will be scheduled ahead of lower priority pods.
- Also enables low priority pods to be removed when a cluster is out of resources



Harden the default RBAC discovery clusterbindings

- Removes the discovery from the set of APIs which allow unauthenticated access by default
- Removed concerns around API extensions that are exposed by the default discovery bindings
- Improves the default privacy and security posture for new clusters



Persistent Local Volumes

Michelle Au, SIG-Storage

Overview

Direct-attached disk as a PersistentVolume

Scheduler intelligence

- Always schedule pods that use local PVs to the correct node
- Bind PVCs to local PVs on nodes that the pod can run on
 - Respects any scheduling constraint: Node selectors/affinity, pod (anti-)affinity, taints, resource requests, etc.

Targeted use cases: high-performance distributed databases (StatefulSets, operators)

-



Not a general purpose storage solution!

Data unavailability/loss when disk or node fails

- Application needs replication and backups
- Recovery is manual or needs an operator
- Node upgrades need to be carefully coordinated

Pod is harder to schedule (only one node choice)

- Use pod priority and preemption to evict lower priority pods

Autoscaling and dynamic provisioning not supported (yet)

- Capacity planning and allocation per application



Adoption

Production use in on-premise and cloud environments

Stressed to ~1000 node clusters, ~10k local PVs

Distributed, scale-out databases



Kubernetes User API

Same PersistentVolumeClaim API! Just specify the appropriate StorageClass.

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: example-local-claim
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 100Gi
storageClassName: local-storage
```



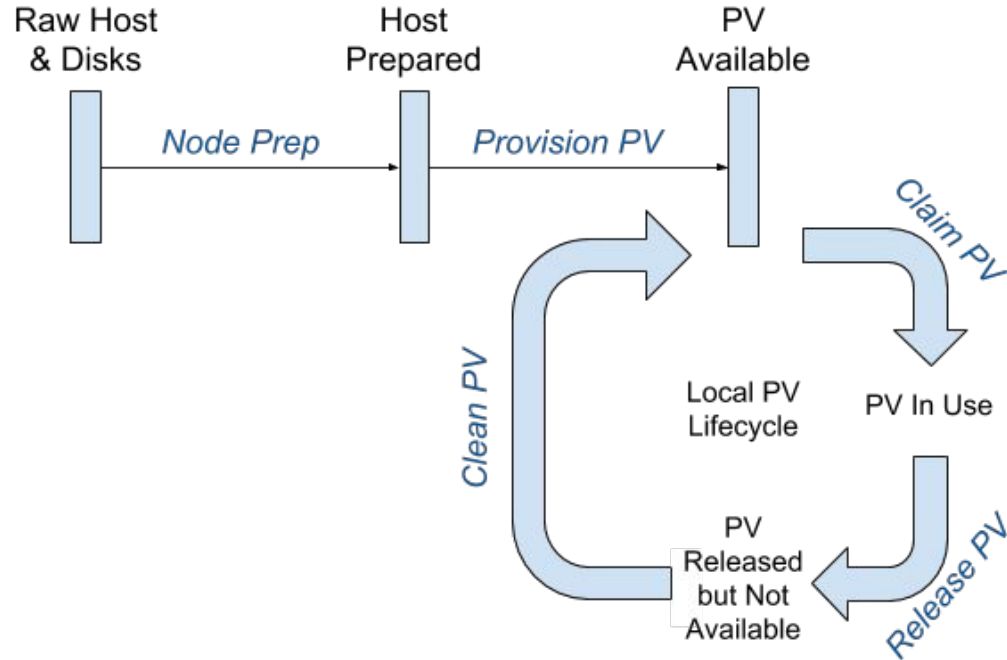
Kubernetes Admin API

Enable intelligent scheduling in the StorageClass

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: local-storage
provisioner: kubernetes.io/no-provisioner
volumeBindingMode: WaitForFirstConsumer
reclaimPolicy: Delete
```



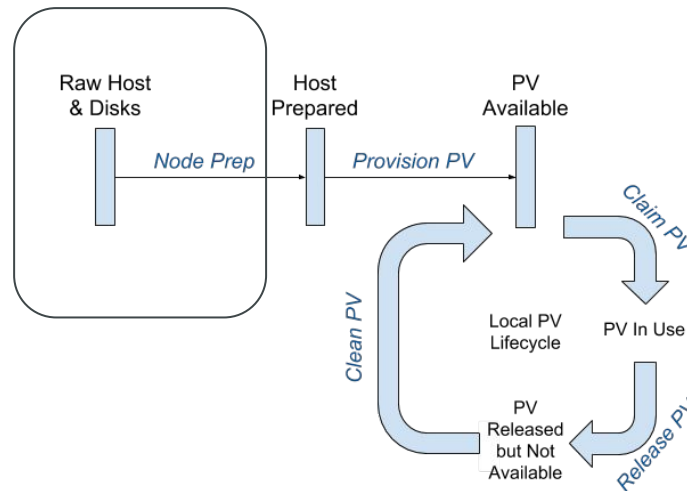
Local PV Management



Local PV Management

Node preparation

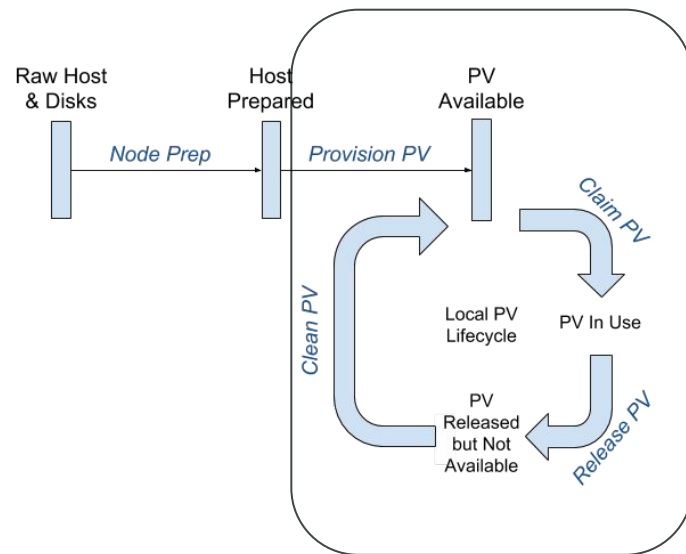
- Environment/cluster-specific, some providers do this automatically
- Partition, format, mount disks as needed
- Link/mount disks for static provisioner



Local PV Management

Local volume **static** provisioner

- DaemonSet that manages the PV lifecycle for local disks
- Admin configures directory per StorageClass and link the disks there
- Controller creates PVs, cleans up disk when PV released, and creates a new PV



Future

Join us at sig-storage if you want to contribute!

Common node preparation tasks - such as RAID

Node/disk failure recovery controller

Dynamic provisioning with LVM



More Information

[1.14 GA Blog Post](#)

[Official documentation](#)

[Local volume static provisioner](#)

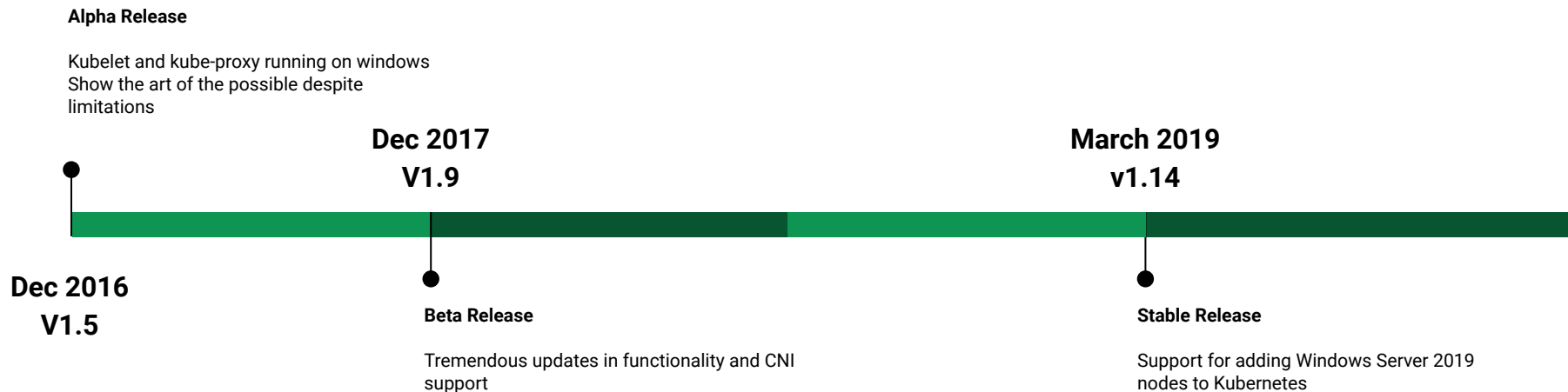


Production-level support for Windows Nodes

Michael Michael, SIG-Windows

Patrick Lang, SIG-Windows

History



Windows containers in Kubernetes

Compute

- Windows Server 2019 Support for containers and nodes
- Pods, service types, workload controllers, and metrics/quotas

Network/Storage

- Multiple CNIs supporting NodePort, ClusterIP, LoadBalancer, ExternalName
- FlexVolume with SMB and iSCSI
- Azure and GCE Disks

Security

- Group Managed Service Accounts



Networking

Azure-CNI

Flannel host-gateway and VXLAN

OVN-Kubernetes

win-bridge, win-overlay



What's next for Windows

→ Listen to user feedback and prioritize

1. Kubeadm support
2. CRI-containerd support, including RuntimeClass for Hyper-V isolation
3. Continue investing in Group Managed Service Account support
4. More CNIs and Storage plugins



Windows Demo

What's coming next?

- Already 3 weeks into 1.15
 - Enhancements freeze is April 30th
- Targeted GA is June 18th



Questions?

Thank You