KubeCon | CloudNativeCon

North America 2019

KubeCon | CloudNativeCon

North America 2019

SIG-Windows Intro

Michael Michael, SIG-Windows Chair [m2 on Slack]
Deep Debroy, SIG-Windows TL [ddebroy on Slack]

Windows Containers in K8s - Why?

Make Kubernetes truly ubiquitous and continue its lead as the top container orchestration platform, supporting all popular programming frameworks

Operational efficiencies by leveraging existing investments in cloud native tools/solutions

Knowledge/Training on Kubernetes is transferable to Windows

Scalable self-service container platform now available for Windows ecosystem

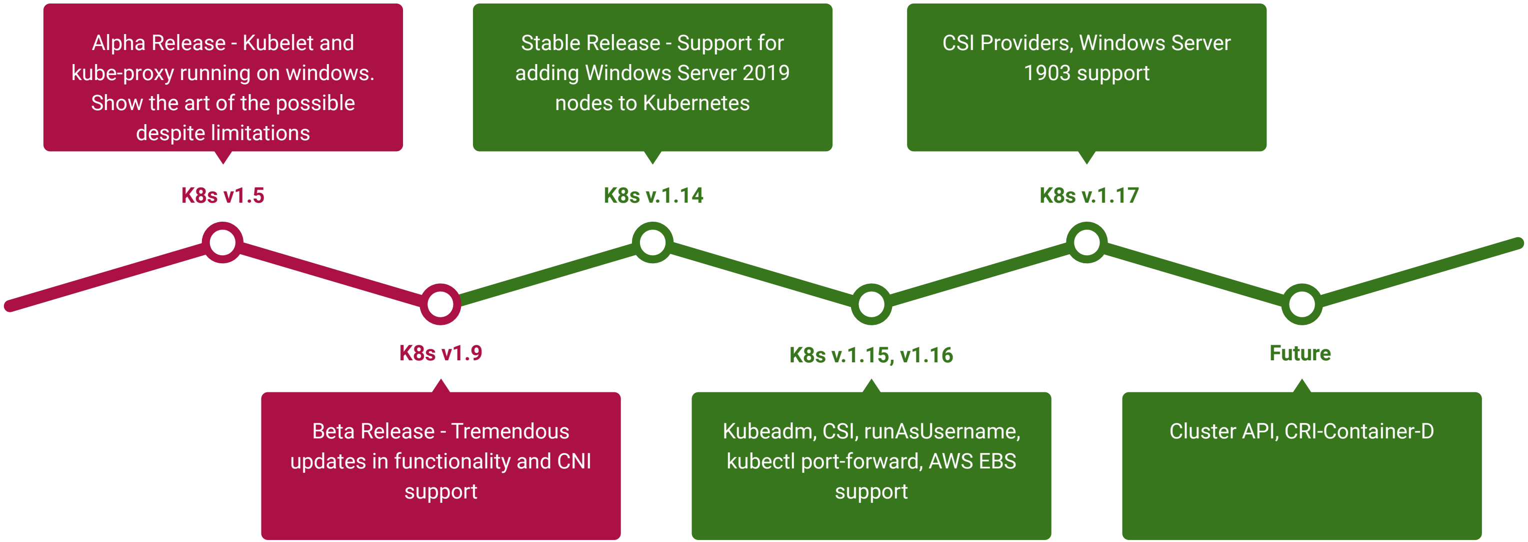
Windows developers can take advantage of cloud native tools to build and deploy distributed applications faster

Retain the benefits of application availability while decreasing costs

•Containerize existing .NET apps to eliminate old HW or underutilized servers

•Streamline migration from end-of-support operating systems

History



K8s v1.5

Alpha Release - Kubelet and kube-proxy running on windows. Show the art of the possible despite limitations

K8s v1.9

Beta Release - Tremendous updates in functionality and CNI support

K8s v.1.14

Stable Release - Support for adding Windows Server 2019 nodes to Kubernetes

K8s v.1.15, v1.16

Kubeadm, CSI, runAsUsername, kubectl port-forward, AWS EBS support

K8s v.1.17

CSI Providers, Windows Server 1903 support

Future

Cluster API, CRI-Container-D

Things to Consider

- ❑ Read the documentation!
- ❑ Where the container runs
  - ❑ Need a Windows Server node = Use NodeSelectors and Taints/Tolerations
- ❑ Resource Consumption
  - ❑ Need higher limits (300Mb min) - need Windows background services per container
- ❑ Kernel/User compatibility
  - ❑ Windows kernel major version should match (for now) – use versioned tags, not latest!
  - ❑ Build on Windows Server 2019 = must run on Windows Server 2019
  - ❑ Hyper-V isolation coming soon

Recent Features

- ❑ Enable users to leverage Windows identity options in containers
  - ❑ gmsaCredentialSpecName, gmsaCredentialSpec - for Group Managed Service Accounts in beta
  - ❑ runAsUserName in beta with 1.17
- ❑ Alpha support for kubeadm join
  - ❑ Maintain scripts to install prerequisites and CNIs
  - ❑ Add a Windows node to a cluster
- ❑ Alpha support for CSI
  - ❑ Leverage persistent storage options for Windows containers
  - ❑ Use host OS proxy to bypass privileged container limitations

K8s 1.17 GMSA: Credential Spec YAMLs

```yaml
apiVersion: windows.k8s.io/v1alpha1
kind: GMSACredentialSpec
metadata:
  name: gmsa-webapp-1  #used for reference
credspec:
  ActiveDirectoryConfig:
    GroupManagedServiceAccounts:
    - Name: WebApp1   #GMSA account Username
      Scope: CONTOSO  #NETBIOS Domain Name
  CmsPlugins:
  - ActiveDirectory
  DomainJoinConfig:
    DnsName: contoso.com  #DNS Domain Name
    DnsTreeName: contoso.com #DNS Domain Name Root
    Guid: 244818ae-87ac-4fcd-92ec-e79e5252348a  #GUID
    MachineAccountName: WebApp1 #GMSA account Username
    NetBiosName: CONTOSO  #NETBIOS Domain Name
    Sid: S-1-5-21-2126449477-2524075714-3094792973 #GMSA SID
```

```yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: webapp1-role
rules:
- apiGroups: ["windows.k8s.io"]
  resources: ["gmsacredentialspecs"]
  verbs: ["use"]
  resourceNames: ["gmsa-webapp-1"]
```

```yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name:default-svc-account-read-gmsa
  namespace: default
subjects:
- kind: ServiceAccount
  name: default
  namespace: default
roleRef:
  kind: ClusterRole
  name: webapp1-role
  apiGroup: rbac.authorization.k8s.io
```

K8s 1.17 Windows Security Context

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: webapp
spec:
  securityContext:
    windowsOptions:
      runAsUserName: "NT AUTHORITY\\NETWORK SERVICE"
      gmsaCredentialSpecName: gmsa-webapp-1
  containers:
    - name: webapp
      image: org/iis:webserver-core-ltsc2019
      securityContext:
        windowsOptions:
          runAsUserName: "ContainerAdministrator"
    - name: logger
      ...
```

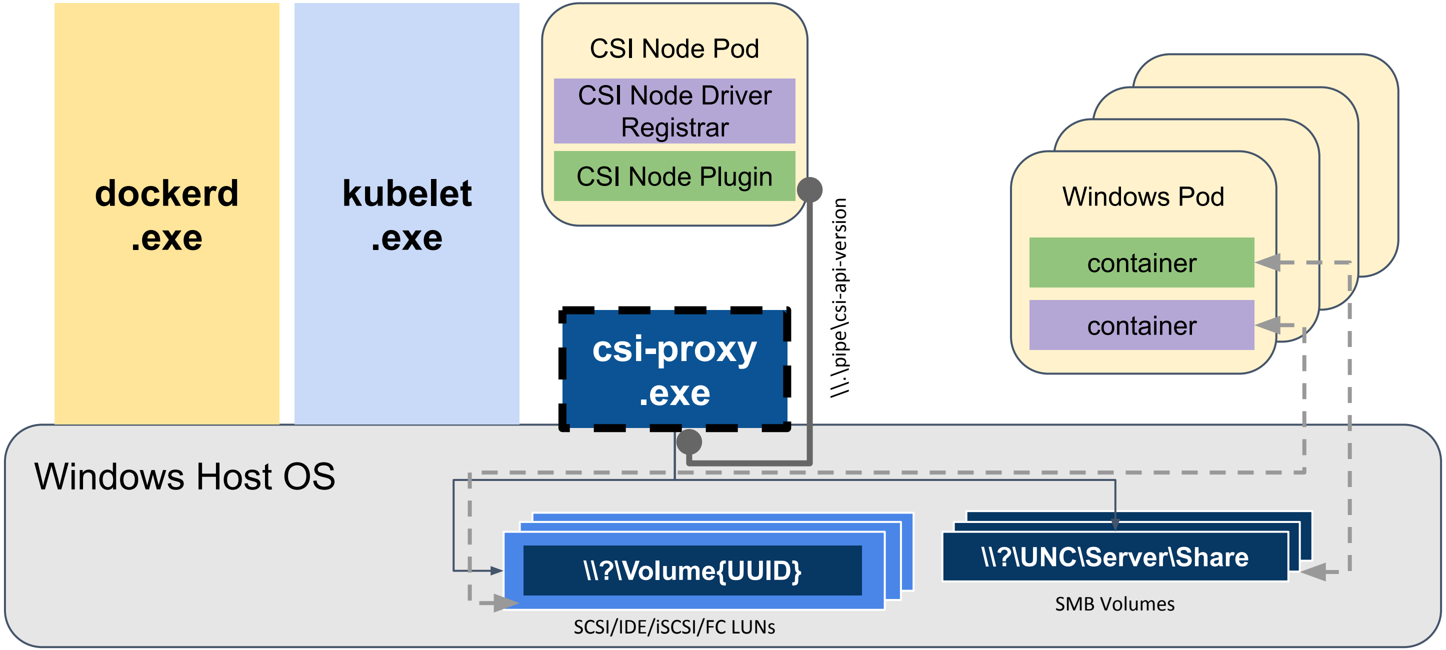
1. Default pod-wide windowsOptions

2. Option to override windowsOptions for each container

3. gmsaCredentialSpec field populated based on gmsaCredentialSpec by a mutating webhook

4. Use postStart lifecycle hook to restart netlogon until nltest returns positive response for GMSA

K8s 1.17 CSI Node Plugin Support



K8s 1.17 CSI Node Plugin Support

Bulk of the work is in the CSI Proxy component [https://github.com/kubernetes-csi/csi-proxy]

- API versioning support (based on model for Kubernetes code generators) complete.
- Versioned API groups to support Disk/Volume/SMB/iSCSI operations in progress.

Enhancements in kubelet and CSI node driver registrar

- Completed in v1.16

Prototyping and testing of experimental versions of CSI Proxy with:

- GCE PD CSI Driver
- AzureDisk CSI Driver

Plans for upcoming cycles

- ❑ Alpha CRI-ContainerD support (sig-node collaboration)
  - ❑ RuntimeClass for Hyper-V isolation
- ❑ Continued kubeadm investments (sig-cluster-lifecycle collaboration)
  - ❑ Cluster API support for Windows worker nodes for CAP-A and CAP-V
- ❑ Promote CSI work to beta (sig-storage collaboration)
- ❑ Promote gMSA to stable (sig-node/sig-api/sig-auth collaboration)

How you can contribute

Join our weekly meetings at 12.30pm Eastern every Tuesday

View recorded community meetings

Find bugs you can fix in our project board

Help us write additional documentation and user stories

Where to find us

https://groups.google.com/forum/#!forum/kubernetes-sig-windows
https://discuss.kubernetes.io/c/general-discussions/windows

#sig-windows
@patricklang
@m2
@ddebroy
@bmo

https://www.youtube.com/playlist?list=PL69nYSiGNLP2OH9InCcNkWNu2bl-gmIU4

https://github.com/kubernetes/community/tree/master/sig-windows

https://zoom.us/j/297282383
Every Tuesday 12.30pm EST

https://kubernetes.io/docs/setup/production-environment/windows