KubeCon | CloudNativeCon

North America 2019

KubeCon | CloudNativeCon

North America 2019

SIG-Windows Deep Dive
Day 2 Operations for Windows Containers

Patrick Lang, SIG-Windows Chair [PatrickLang on Slack]
Michael Michael, SIG-Windows Chair [m2 on Slack]

Topics

Where we are today, where we’re going

- Best Practices in App deployments
- Guidelines for Node maintenance
  - Monthly patches
  - OS version upgrades
- Making Windows logs visible with existing tools
- Centralizing Logs
- Node monitoring
- Disaster Recovery

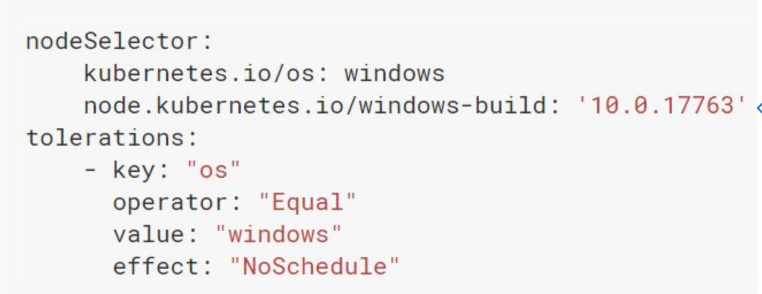
Best Practices - Multi-OS

NodeSelector

- Steer to right OS & version
- Enforce Host\Guest compatibility

Taints

- Prevent accidental deployment to Windows



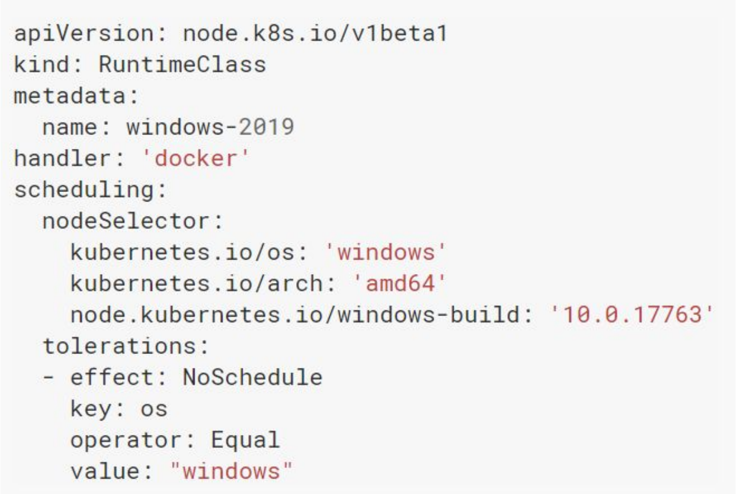
```yaml
nodeSelector:
    kubernetes.io/os: windows
    node.kubernetes.io/windows-build: '10.0.17763'
tolerations:
    - key: "os"
      operator: "Equal"
      value: "windows"
      effect: "NoSchedule"
```

8 lines per pod

New for 1.17

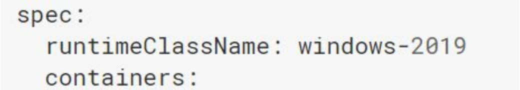
RuntimeClass

- Define once per cluster



```yaml
apiVersion: node.k8s.io/v1beta1
kind: RuntimeClass
metadata:
  name: windows-2019
handler: 'docker'
scheduling:
  nodeSelector:
    kubernetes.io/os: 'windows'
    kubernetes.io/arch: 'amd64'
    node.kubernetes.io/windows-build: '10.0.17763'
  tolerations:
  - effect: NoSchedule
    key: os
    operator: Equal
    value: "windows"
```

1 line per pod



```yaml
spec:
  runtimeClassName: windows-2019
  containers:
```

Best Practices - Resources

CPU

- Shares = no minimum required to start, always relative to load & other containers
- Percentage - PR is ready for review. Be sure to test your app for minimums. Probably at least .1 CPU needed to start up background processes.

Min Memory

- Server Core needs at least 200Mi to start

Memory Considerations

- No pod evictions due to memory pressure
- Processes page to disk → slow performance
- Use `kubelet-reserve` and `system-reserve` to keep 2Gi+ for the node processes
- Always use limits and reserves → honored in scheduler

Testing & Enforcing Best Practices

Testing & Enforcing Best Practices

- Open Policy Agent - tools and Rego language for writing policies
  - https://www.openpolicyagent.org/
- Gatekeeper - admission controller to block deployments failing policies
- Conftest - uses OPA to test Yaml on your own box
  - https://github.com/instrumenta/conftest

Contributions to Gatekeeper rule library welcome!

Node Patching

Monthly Updates

In-Place

1. Cordon node
2. Wait
3. Drain node
4. Run Windows Update, reboot
5. Uncordon node

(async) 4. Rebuild container to update

?

Swap

1. Cordon node
2. Wait
3. Drain node
4. Replace+ReJoin Windows Node
5. Uncordon node

(async) 4. Rebuild container to update

Capacity? - both need at least 1 extra node to preserve uptime
Time to deploy? - adding a node may be faster in the cloud
Time to roll back? - deleting node is faster than uninstalling a patch and rebooting
Canary, Blue/Green, or A/B testing - easier with node swaps

Node OS Upgrade

In-Place

1. Cordon node
2. Wait + Drain
3. Run Windows Update, reboot
4. Uncordon node

Swap

1. Cordon node
2. Wait + Drain
3. Replace Windows Node
4. Uncordon node

Caveat

Cannot in-place upgrade from long-term servicing channel (LTS) to semi-annual channel (SAC) or back

Containers and Node OS Upgrade

1. Either
    a. Ensure NodeSelector/Tolerations are set on all deployments with version
    b. Taint new nodes before uncordoning them
2. Add new Windows version nodes
3. Rebuild app based on newer OS version
4. Update NodeSelector / Tolerations on the deployment

Hyper-V Isolation will make step 3 optional in the future

Cluster API

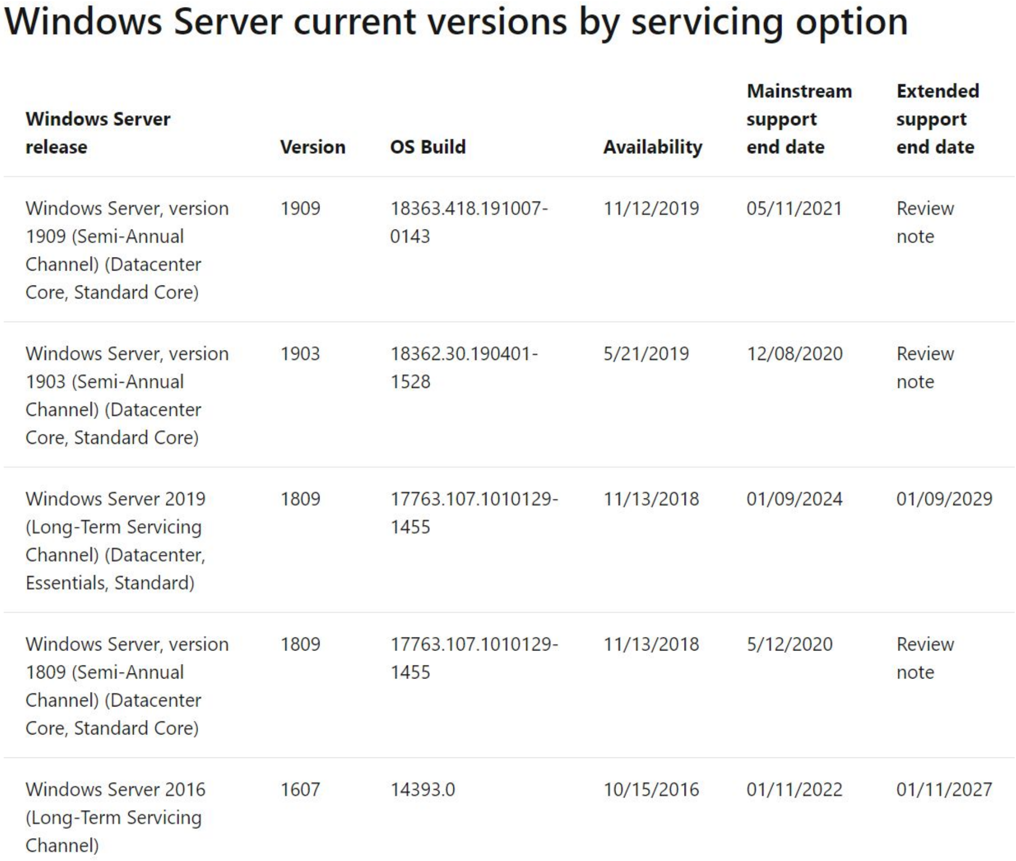
- Declarative, Kubernetes-style API to cluster creation, configuration, and management
- Lifecycle management using Cluster API is one of our top priorities in 2020

Version Support

Currently supporting

- Windows Server 2019
- Windows Server version 1903

SACs supported for 18 months



Windows Server current versions by servicing option

Windows Server release	Version	OS Build	Availability	Mainstream support end date	Extended support end date
Windows Server, version 1909 (Semi-Annual Channel) (Datacenter Core, Standard Core)	1909	18363.418.191007-0143	11/12/2019	05/11/2021	Review note
Windows Server, version 1903 (Semi-Annual Channel) (Datacenter Core, Standard Core)	1903	18362.30.190401-1528	5/21/2019	12/08/2020	Review note
Windows Server 2019 (Long-Term Servicing Channel) (Datacenter, Essentials, Standard)	1809	17763.107.1010129-1455	11/13/2018	01/09/2024	01/09/2029
Windows Server, version 1809 (Semi-Annual Channel) (Datacenter Core, Standard Core)	1809	17763.107.1010129-1455	11/13/2018	5/12/2020	Review note
Windows Server 2016 (Long-Term Servicing Channel)	1607	14393.0	10/15/2016	01/11/2022	01/11/2027

Visit https://bit.ly/347mOUi
for some quick polls

When poll is active, respond at **PollEv.com/michaelmicha980**

Can you use Semi-Annual Channel?

I am a Software Assurance Customer

I am running in a cloud offering SAC

N/A to me

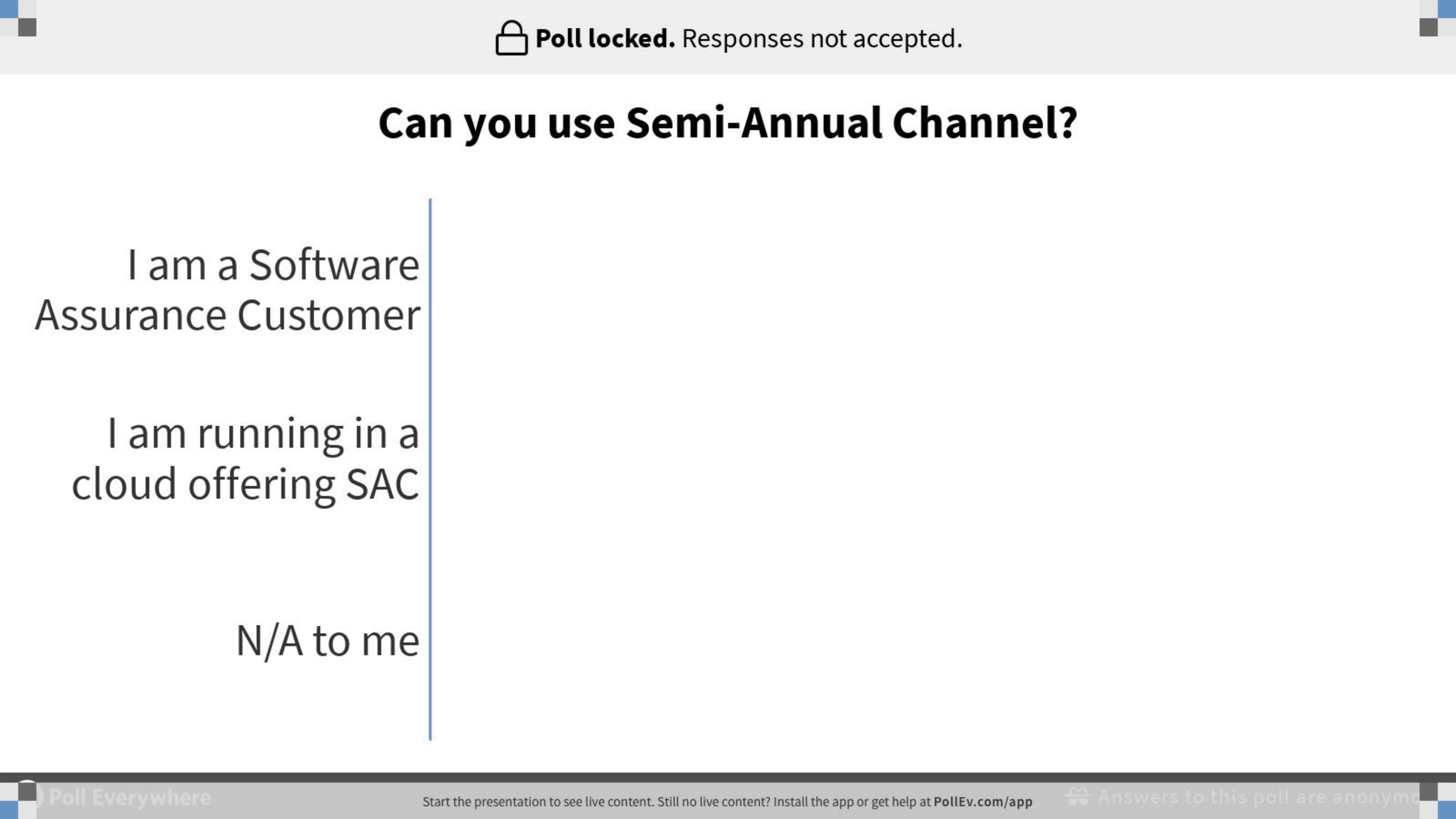
Total Results: 0

Can you use Semi-Annual Channel?

I am a Software Assurance Customer

I am running in a cloud offering SAC

N/A to me



Poll locked. Responses not accepted.

Can you use Semi-Annual Channel?

I am a Software Assurance Customer

I am running in a cloud offering SAC

N/A to me

Poll Everywhere

Start the presentation to see live content. Still no live content? Install the app or get help at **PollEv.com/app**

Answers to this poll are anonymo

When poll is active, respond at **PollEv.com/michaelmicha980**

Would you upgrade every 6-18 months to get new improvements?

Yes

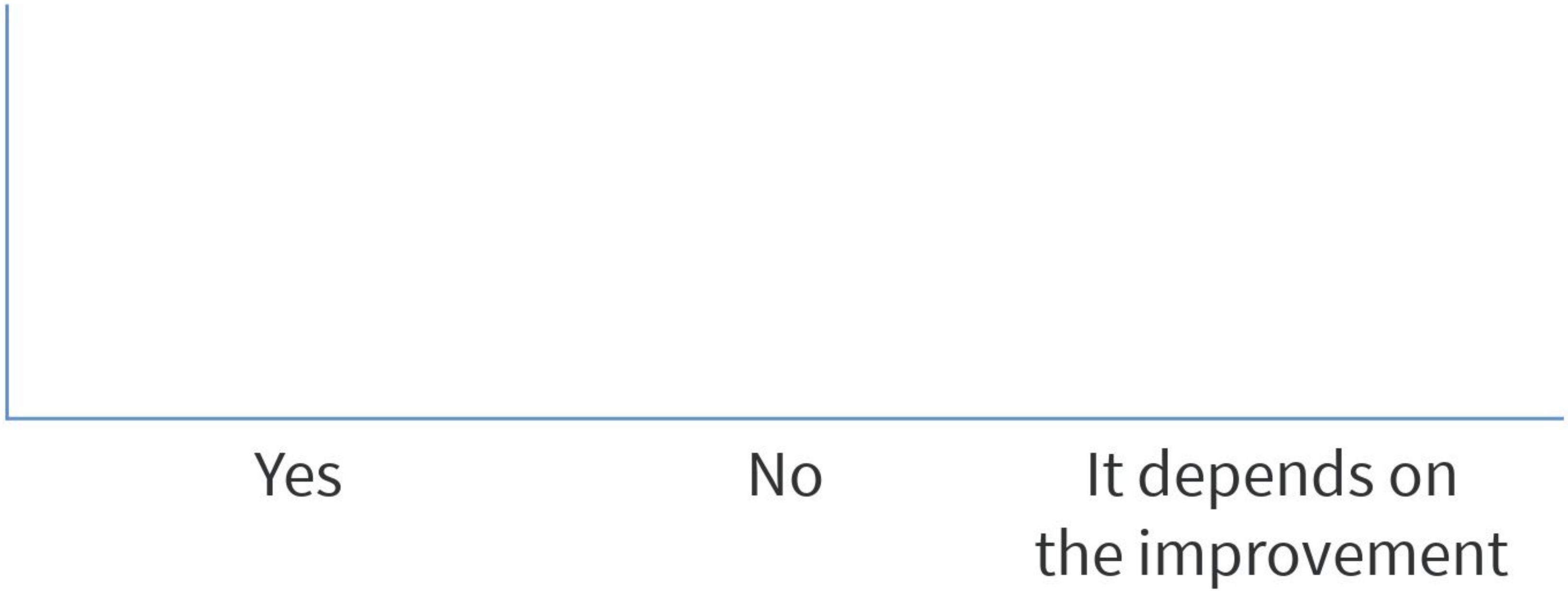
No

It depends on the improvement

Total Results: 0

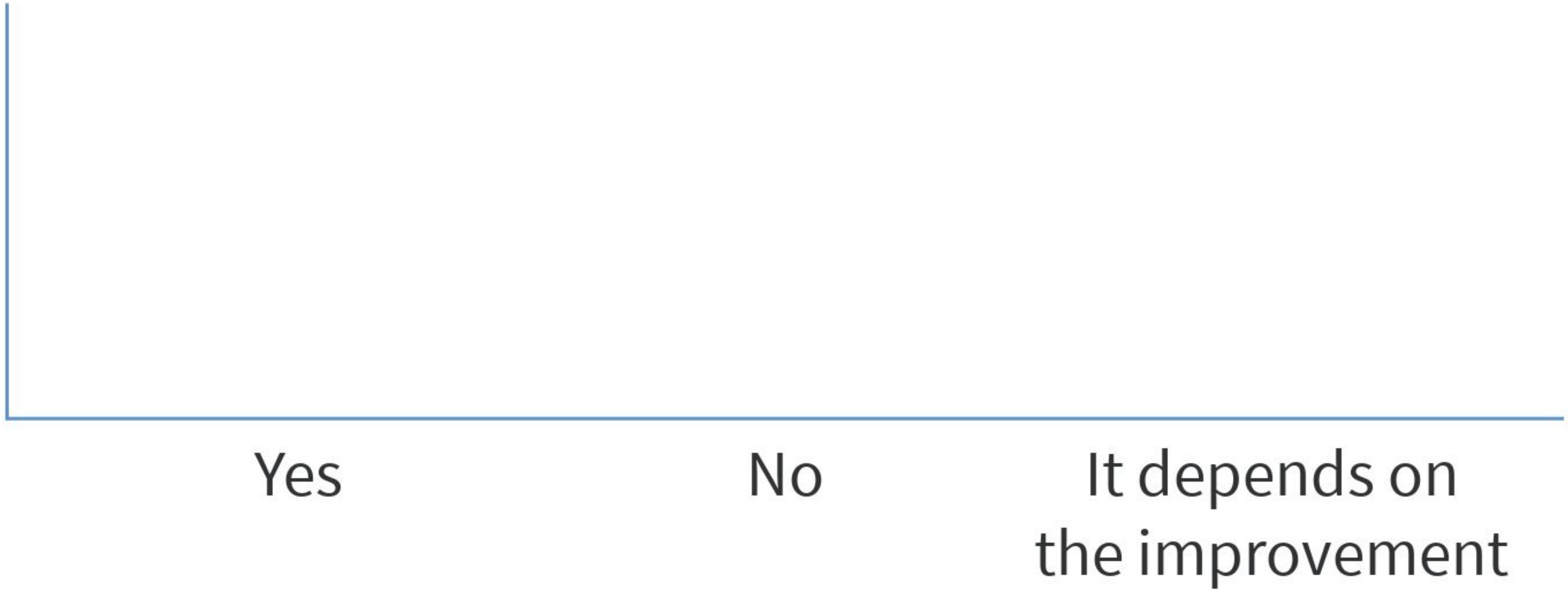
When poll is active, respond at **PollEv.com/michaelmicha980**

Would you upgrade every 6-18 months to get new improvements?

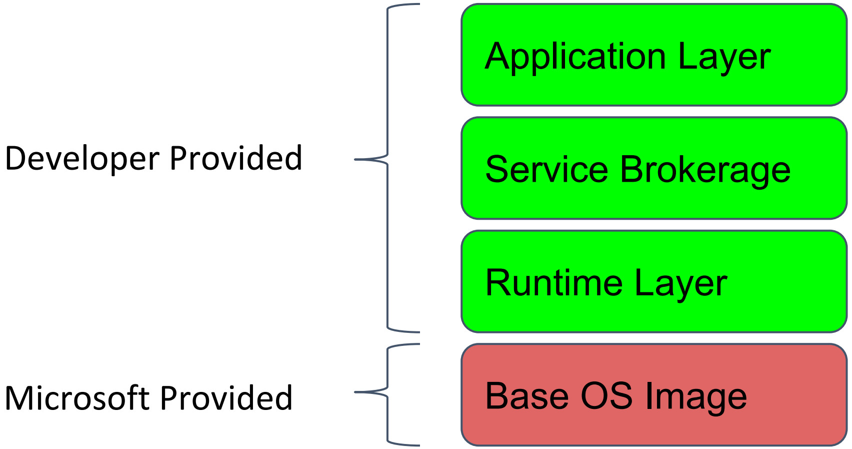


Poll locked. Responses not accepted.

Would you upgrade every 6-18 months to get new improvements?



Anatomy of a Windows Container



Developer Provided

Application Layer

Service Brokerage

Runtime Layer

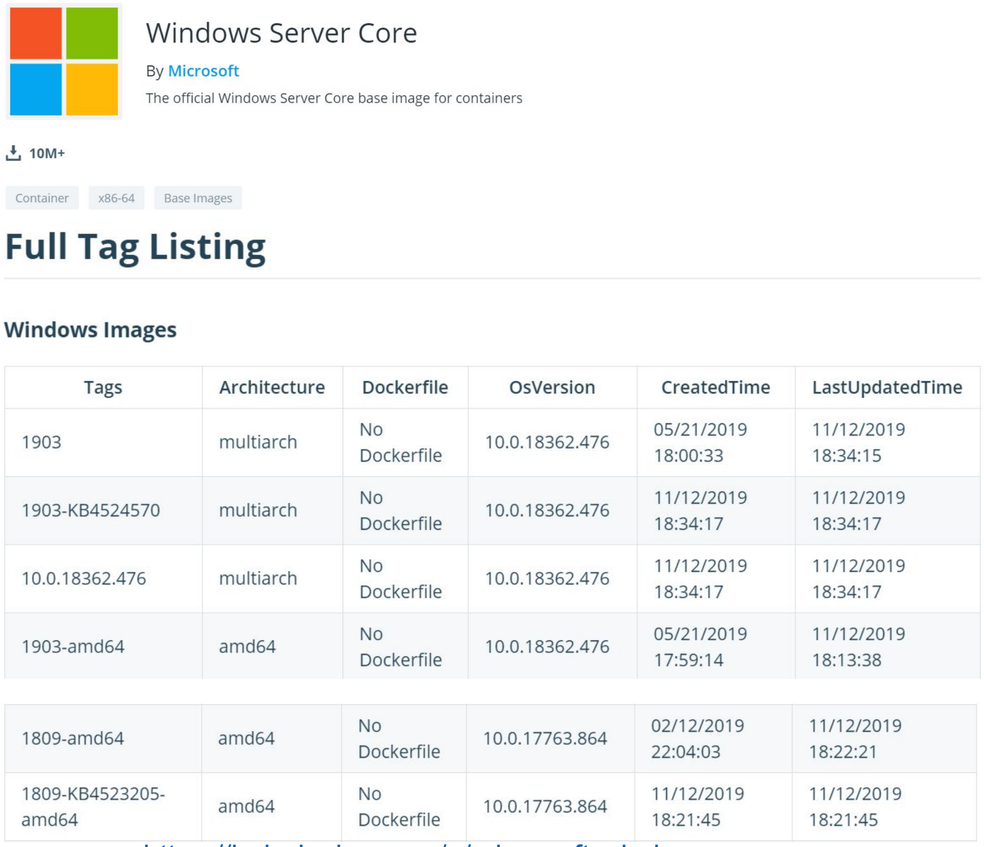
Microsoft Provided

Base OS Image

Use derived images to share application building blocks

Applications - Patching

1. Start from official Microsoft image
2. Find tag for most recent update
3. Update Dockerfile
4. Deploy through your CI / CD system



Windows Server Core

By Microsoft

The official Windows Server Core base image for containers

10M+

Container x86-64 Base Images

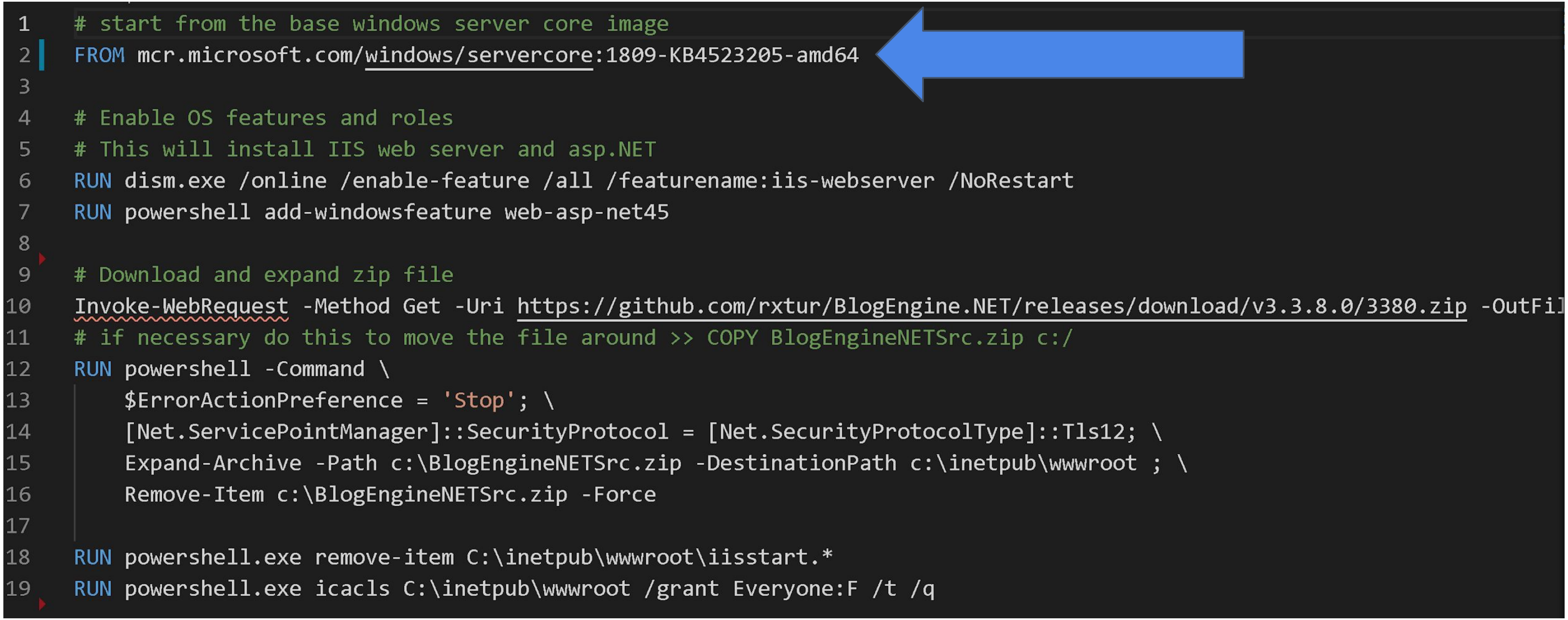
Full Tag Listing

Windows Images

Tags	Architecture	Dockerfile	OsVersion	CreatedTime	LastUpdatedTime
1903	multiarch	No Dockerfile	10.0.18362.476	05/21/2019 18:00:33	11/12/2019 18:34:15
1903-KB4524570	multiarch	No Dockerfile	10.0.18362.476	11/12/2019 18:34:17	11/12/2019 18:34:17
10.0.18362.476	multiarch	No Dockerfile	10.0.18362.476	11/12/2019 18:34:17	11/12/2019 18:34:17
1903-amd64	amd64	No Dockerfile	10.0.18362.476	05/21/2019 17:59:14	11/12/2019 18:13:38
1809-amd64	amd64	No Dockerfile	10.0.17763.864	02/12/2019 22:04:03	11/12/2019 18:22:21
1809-KB4523205-amd64	amd64	No Dockerfile	10.0.17763.864	11/12/2019 18:21:45	11/12/2019 18:21:45

https://hub.docker.com/_/microsoft-windows-servercore

Applications - Patching



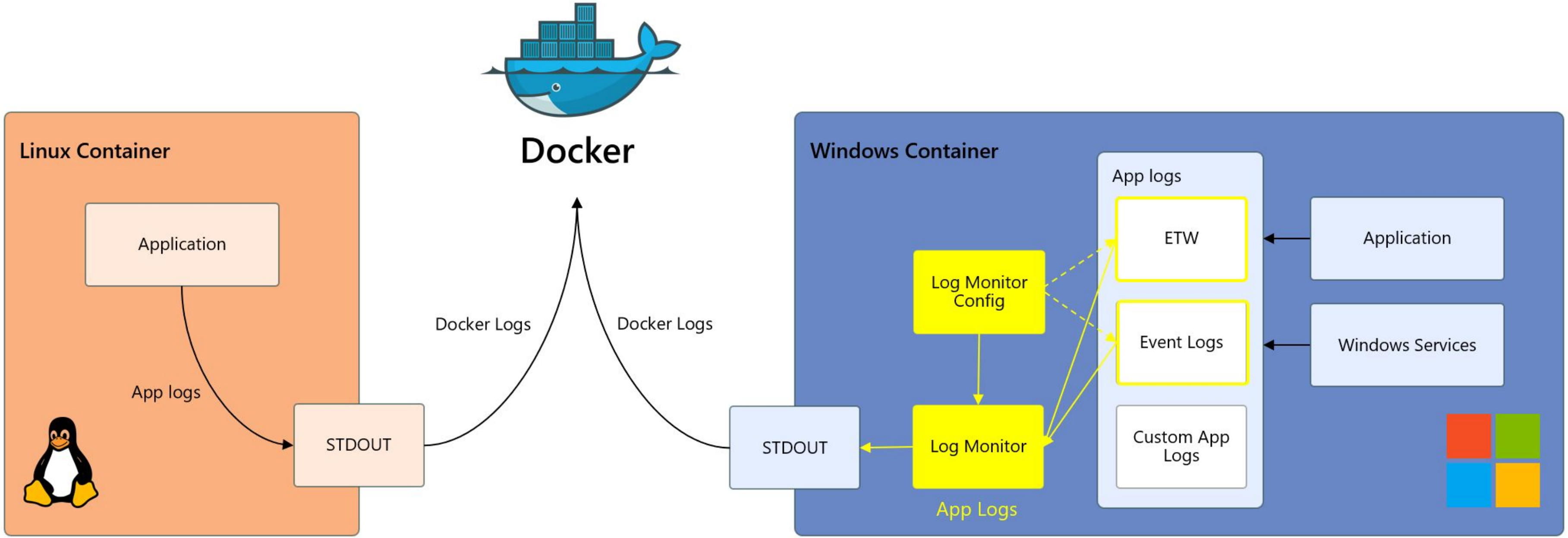
```
# start from the base windows server core image
FROM mcr.microsoft.com/windows/servercore:1809-KB4523205-amd64

# Enable OS features and roles
# This will install IIS web server and asp.NET
RUN dism.exe /online /enable-feature /all /featurename:iis-webserver /NoRestart
RUN powershell add-windowsfeature web-asp-net45

# Download and expand zip file
Invoke-WebRequest -Method Get -Uri https://github.com/rxtur/BlogEngine.NET/releases/download/v3.3.8.0/3380.zip -OutFil
# if necessary do this to move the file around >> COPY BlogEngineNETSrc.zip c:/
RUN powershell -Command \
    $ErrorActionPreference = 'Stop'; \
    [Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12; \
    Expand-Archive -Path c:\BlogEngineNETSrc.zip -DestinationPath c:\inetpub\wwwroot ; \
    Remove-Item c:\BlogEngineNETSrc.zip -Force

RUN powershell.exe remove-item C:\inetpub\wwwroot\iisstart.*
RUN powershell.exe icacls C:\inetpub\wwwroot /grant Everyone:F /t /q
```

Applications - CI/CD Solutions

Survey of CI/CD tools

- AppVeyor
- Azure DevOps
- Azure Container Registry Tasks
- CircleCI
- CodeFresh
- Docker
- … and many others

LogMonitor



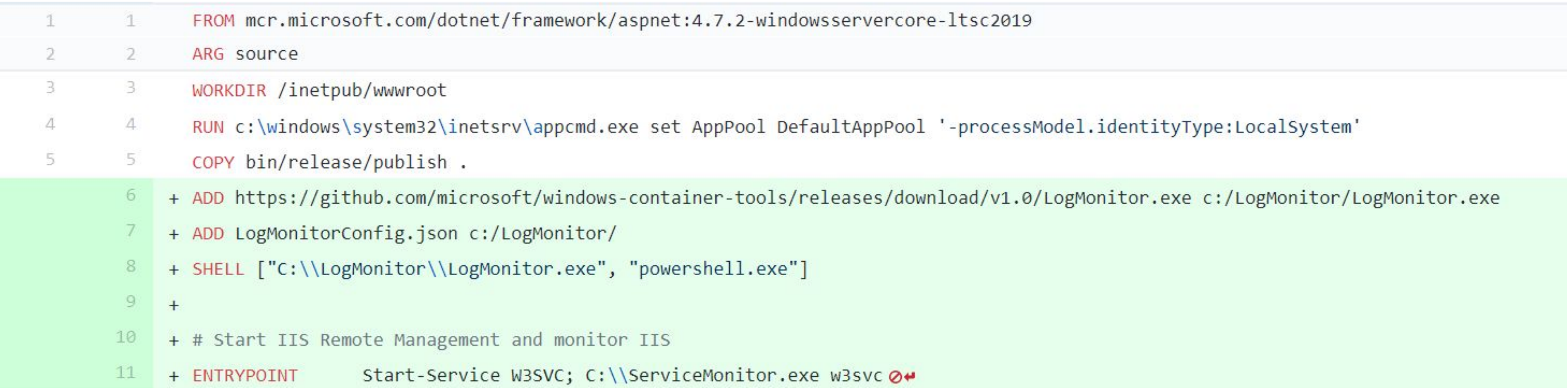
Full Announcement: https://bit.ly/2KE5VZP

LogMonitor

- LogMonitor released on GitHub
  - https://github.com/microsoft/windows-container-tools
- Supports multiple log types
  - Event, ETW Providers, Custom app logs
  - Can tail multiple text files
- Outputs to STDOUT, visible in kubectl logs <pod>
- Simple changes to Dockerfile
  - Modify entrypoint or shell
  - Add a JSON config file listing what to log

Feedback & contributions welcome!

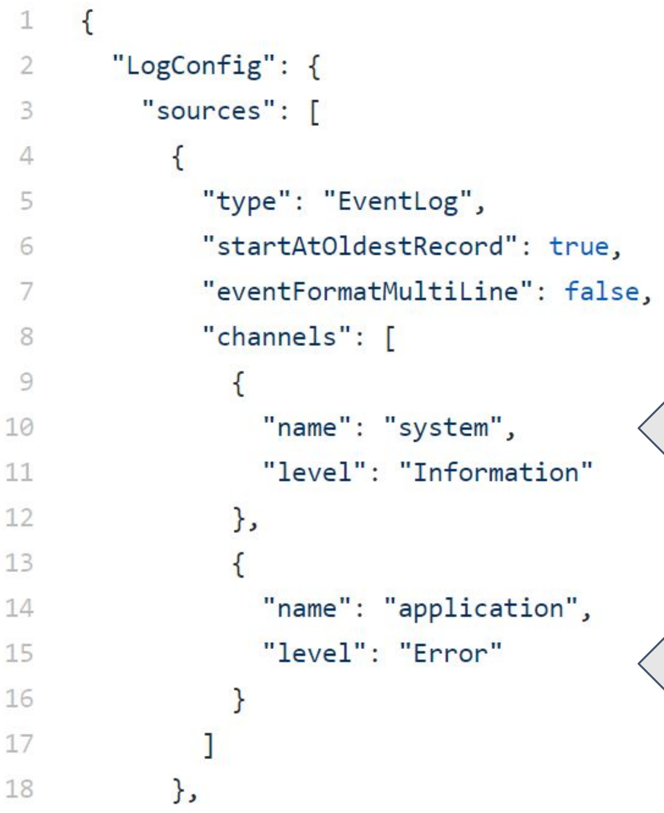
LogMonitor



```
  FROM mcr.microsoft.com/dotnet/framework/aspnet:4.7.2-windowsservercore-ltsc2019
  ARG source
  WORKDIR /inetpub/wwwroot
  RUN c:\windows\system32\inetsrv\appcmd.exe set AppPool DefaultAppPool '-processModel.identityType:LocalSystem'
  COPY bin/release/publish .
+ ADD https://github.com/microsoft/windows-container-tools/releases/download/v1.0/LogMonitor.exe c:/LogMonitor/LogMonitor.exe
+ ADD LogMonitorConfig.json c:/LogMonitor/
+ SHELL ["C:\\LogMonitor\\LogMonitor.exe", "powershell.exe"]
+
+ # Start IIS Remote Management and monitor IIS
+ ENTRYPOINT      Start-Service W3SVC; C:\\ServiceMonitor.exe w3svc
```

Full Code Sample at https://github.com/patricklang/fabrikamfiber

LogMonitor

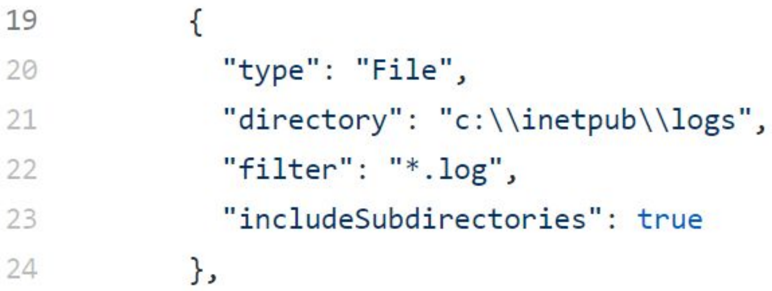


```
{
  "LogConfig": {
    "sources": [
      {
        "type": "EventLog",
        "startAtOldestRecord": true,
        "eventFormatMultiLine": false,
        "channels": [
          {
            "name": "system",
            "level": "Information"
          },
          {
            "name": "application",
            "level": "Error"
          }
        ]
      },
```

Background Services

Crash handlers

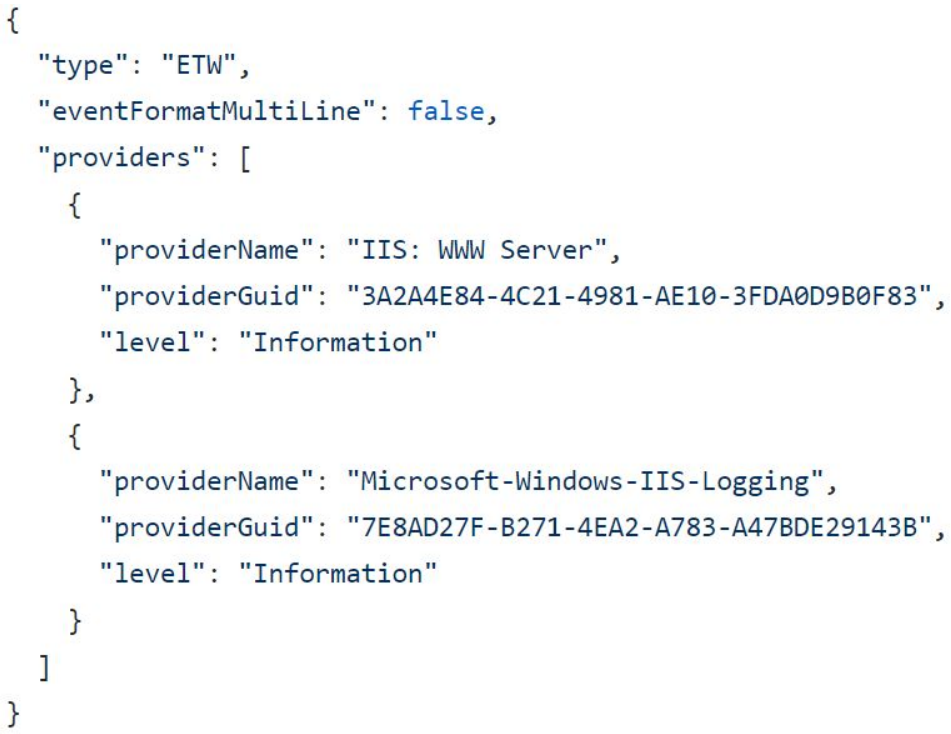
LogMonitor



```
      {
        "type": "File",
        "directory": "c:\\inetpub\\logs",
        "filter": "*.log",
        "includeSubdirectories": true
      },
```

HTTP logs

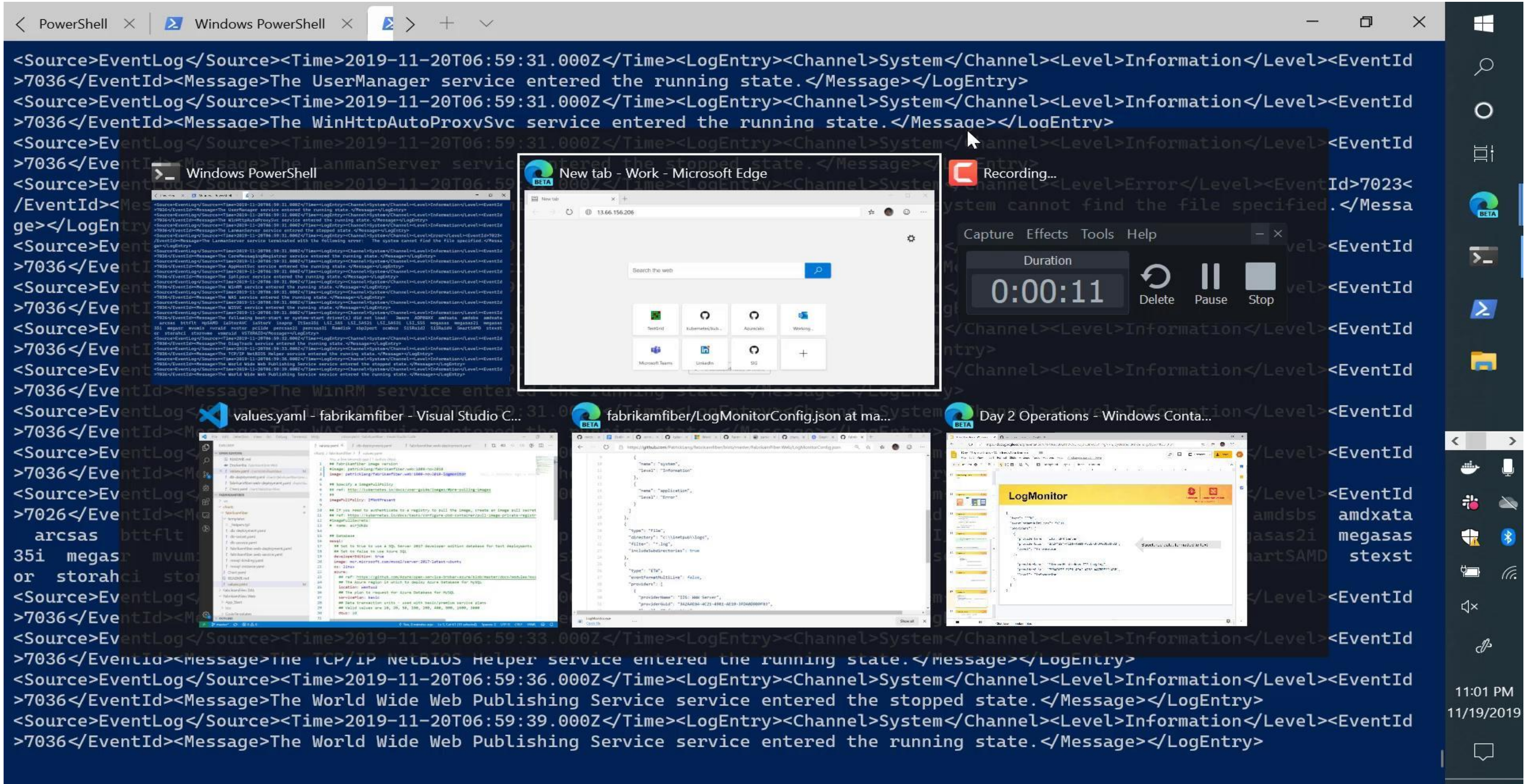
LogMonitor



```
{
  "type": "ETW",
  "eventFormatMultiLine": false,
  "providers": [
    {
      "providerName": "IIS: WWW Server",
      "providerGuid": "3A2A4E84-4C21-4981-AE10-3FDA0D9B0F83",
      "level": "Information"
    },
    {
      "providerName": "Microsoft-Windows-IIS-Logging",
      "providerGuid": "7E8AD27F-B271-4EA2-A783-A47BDE29143B",
      "level": "Information"
    }
  ]
}
```

Structured data, formatted to text

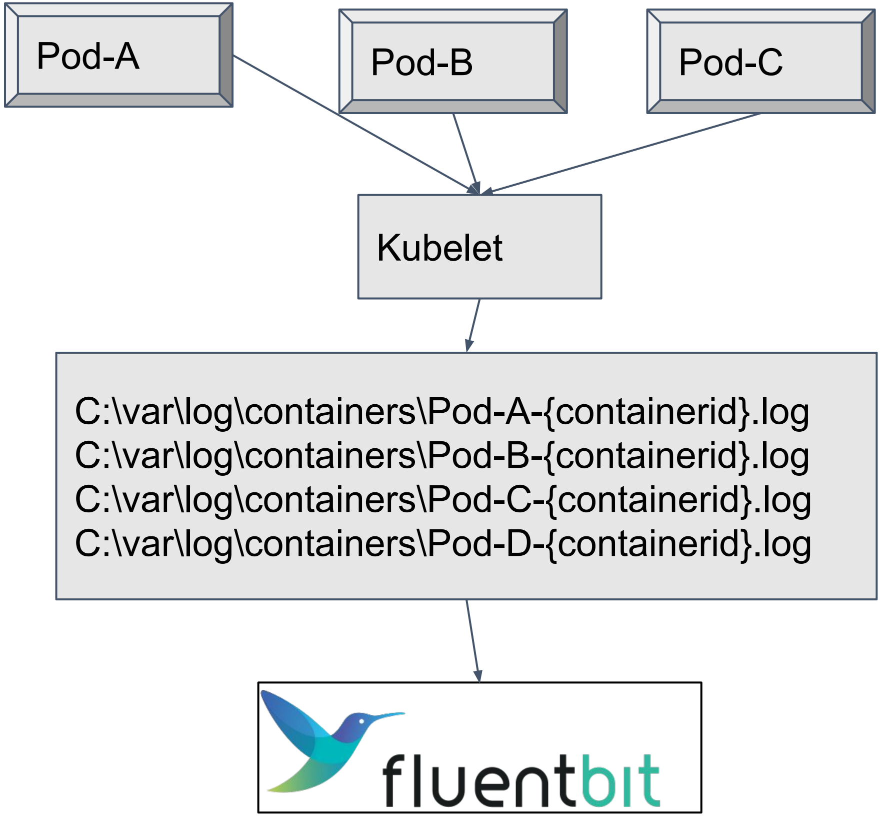
Log Monitor Demo



```
<Source>EventLog</Source><Time>2019-11-20T06:59:31.000Z</Time><LogEntry><Channel>System</Channel><Level>Information</Level><EventId
>7036</EventId><Message>The UserManager service entered the running state.</Message></LogEntry>
<Source>EventLog</Source><Time>2019-11-20T06:59:31.000Z</Time><LogEntry><Channel>System</Channel><Level>Information</Level><EventId
>7036</EventId><Message>The WinHttpAutoProxySvc service entered the running state.</Message></LogEntry>
<Source>EventLog</Source><Time>2019-11-20T06:59:31.000Z</Time><LogEntry><Channel>System</Channel><Level>Information</Level><EventId
>7036</EventId><Message>The LanmanServer service entered the stopped state.</Message></LogEntry>
<Source>EventLog</Source><Time>2019-11-20T06:59                                     Channel><Level>Error</Level><EventId>7023<
/EventId><Mes                                                             ystem cannot find the file specified.</Messa
ge></LogEntry>
<Source>Ev                                                                                                      vel><EventId
>7036</Eve
<Source>Ev                                                                                                      vel><EventId
>7036</Eve
<Source>Ev                                                            </Channel><Level>Information</Level><EventId
>7036</Eve                                                            ntry>
<Source>Ev                                                            </Channel><Level>Information</Level><EventId
>7036</EventId><Message>The WinRM service entere                                        >
<Source>EventLog                   31.0                                            ion</Level><EventId
>7036</EventId><M                 WAS service entered the
<Source>EventLog<                                                                          /Level><EventId
>7036</EventId><M
<Source>EventLog<                                                                          /Level><EventId
>7026</EventId><M                                                                  amdsbs  amdxata
  arcsas  bttflt                                                                   gasas2i  megasas
35i  megasr  mvumi                                                                 martSAMD  stexst
or  storahci  stor
<Source>EventLog<                                                                          /Level><EventId
>7036</EventId><M
<Source>EventLog</Source><Time>2019-11-20T06:59:33.000Z</Time><LogEntry><Channel>System</Channel><Level>Information</Level><EventId
>7036</EventId><Message>The TCP/IP NetBIOS Helper service entered the running state.</Message></LogEntry>
<Source>EventLog</Source><Time>2019-11-20T06:59:36.000Z</Time><LogEntry><Channel>System</Channel><Level>Information</Level><EventId
>7036</EventId><Message>The World Wide Web Publishing Service service entered the stopped state.</Message></LogEntry>
<Source>EventLog</Source><Time>2019-11-20T06:59:39.000Z</Time><LogEntry><Channel>System</Channel><Level>Information</Level><EventId
>7036</EventId><Message>The World Wide Web Publishing Service service entered the running state.</Message></LogEntry>
```

PowerShell
Windows PowerShell
Windows PowerShell
New tab - Work - Microsoft Edge
Recording...
Capture Effects Tools Help
Duration
0:00:11
Delete
Pause
Stop
values.yaml - fabrikamfiber - Visual Studio C...
fabrikamfiber/LogMonitorConfig.json at ma...
Day 2 Operations - Windows Conta...
LogMonitor
11:01 PM
11/19/2019

Log Aggregation



```
C:\var\log\containers\Pod-A-{containerid}.log
C:\var\log\containers\Pod-B-{containerid}.log
C:\var\log\containers\Pod-C-{containerid}.log
C:\var\log\containers\Pod-D-{containerid}.log
```

State of Fluent Bit

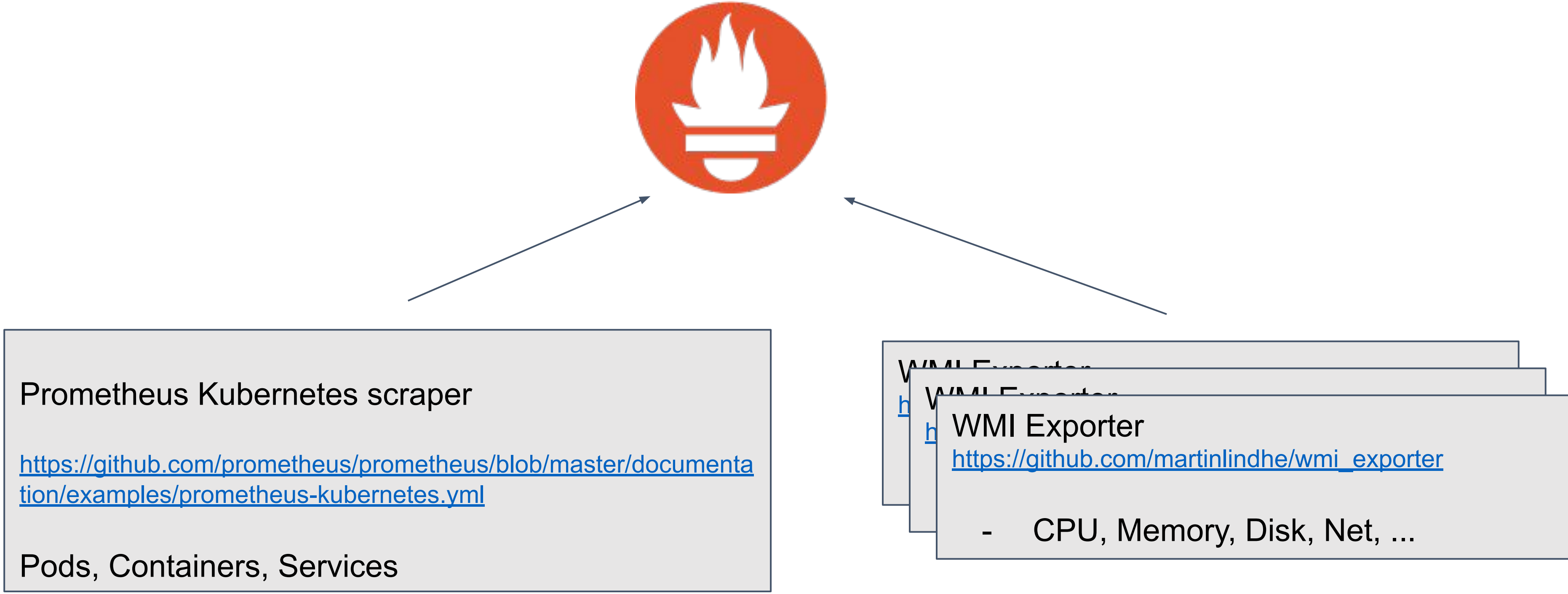
Builds available on Windows - beta in 1.1

Progress tracked on GitHub: https://github.com/fluent/fluent-bit/issues/960

Current state

- Need fixes to wildcard handling on Windows
- People actively looking at how to run as a daemonset, mounting c:\var\log\containers using HostPath

Node Metrics



Prometheus Kubernetes scraper

https://github.com/prometheus/prometheus/blob/master/documentation/examples/prometheus-kubernetes.yml

Pods, Containers, Services

WMI Exporter
https://github.com/martinlindhe/wmi_exporter

- CPU, Memory, Disk, Net, ...

More details at
https://github.com/kubernetes-monitoring/kubernetes-mixin/#dashboards-for-windows-nodes

Compliance

- Integrate it with your CI/CD
- Host images in a private registry
- Scan images for vulnerabilities and compliance
  - Aqua Security
  - Twistlock / Palo Alto Networks
  - Anchore Enterprise (under investigation)

Disaster Recovery

- Existing Kubernetes DR practices apply
- Back up your K8s state and PVs
- Velero Community is working on supporting Windows
- You can do multi-cloud DR by leveraging HA DNS/IngressController/PV/Datastore
    - Tradeoff between availability and consistency (CAP Theorem)

How you can help

Share your story

- Docs
- Blogs
- SIG-Windows meetings - demos welcome, working or not

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