



KubeCon

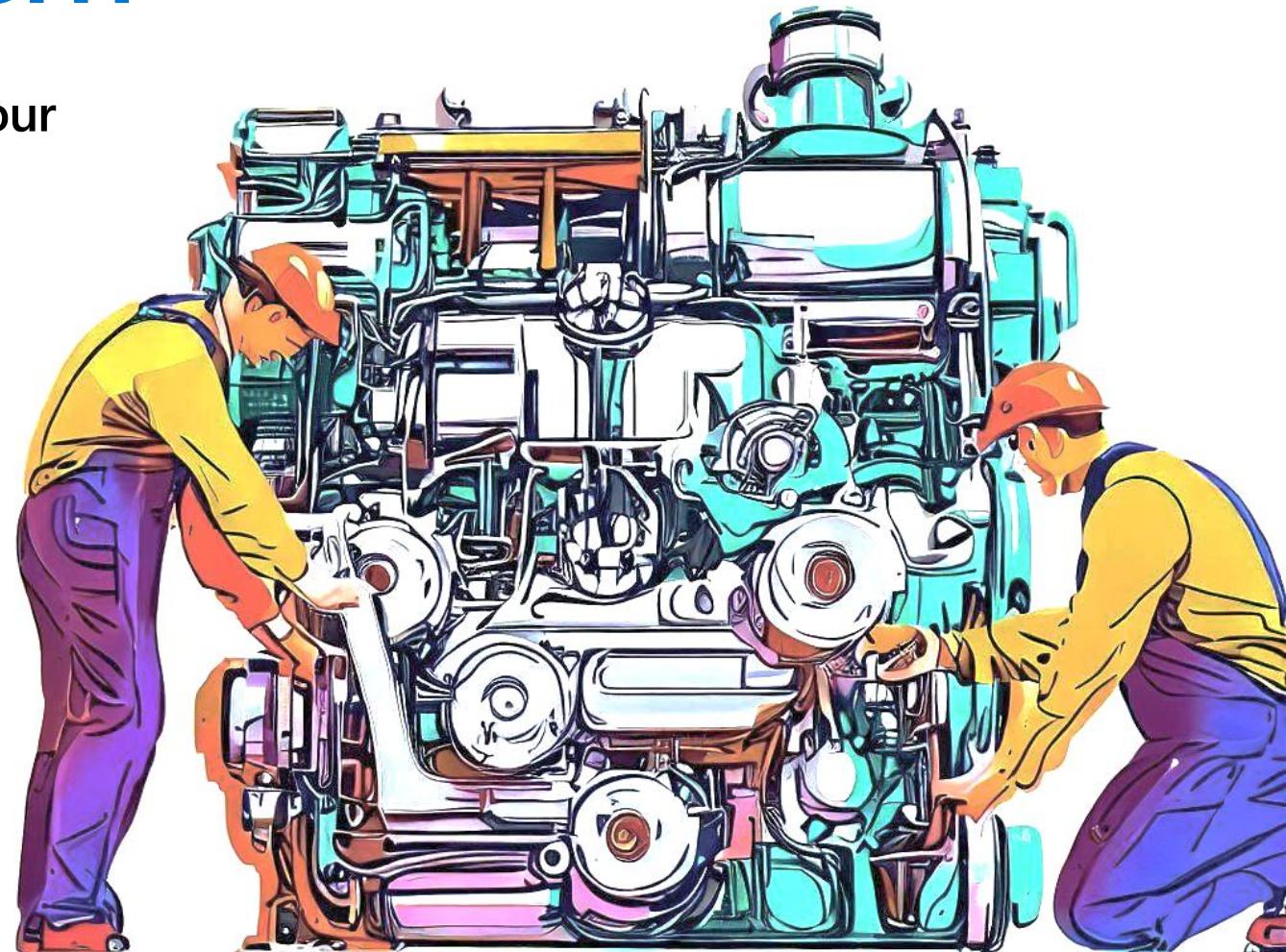


CloudNativeCon

Europe 2023

Making Legacy Modern

How to monitor and fine tune the performance of your Windows clusters





KubeCon



CloudNativeCon

Europe 2023

18-21 April



Brandon Smith

Product Manager

Microsoft

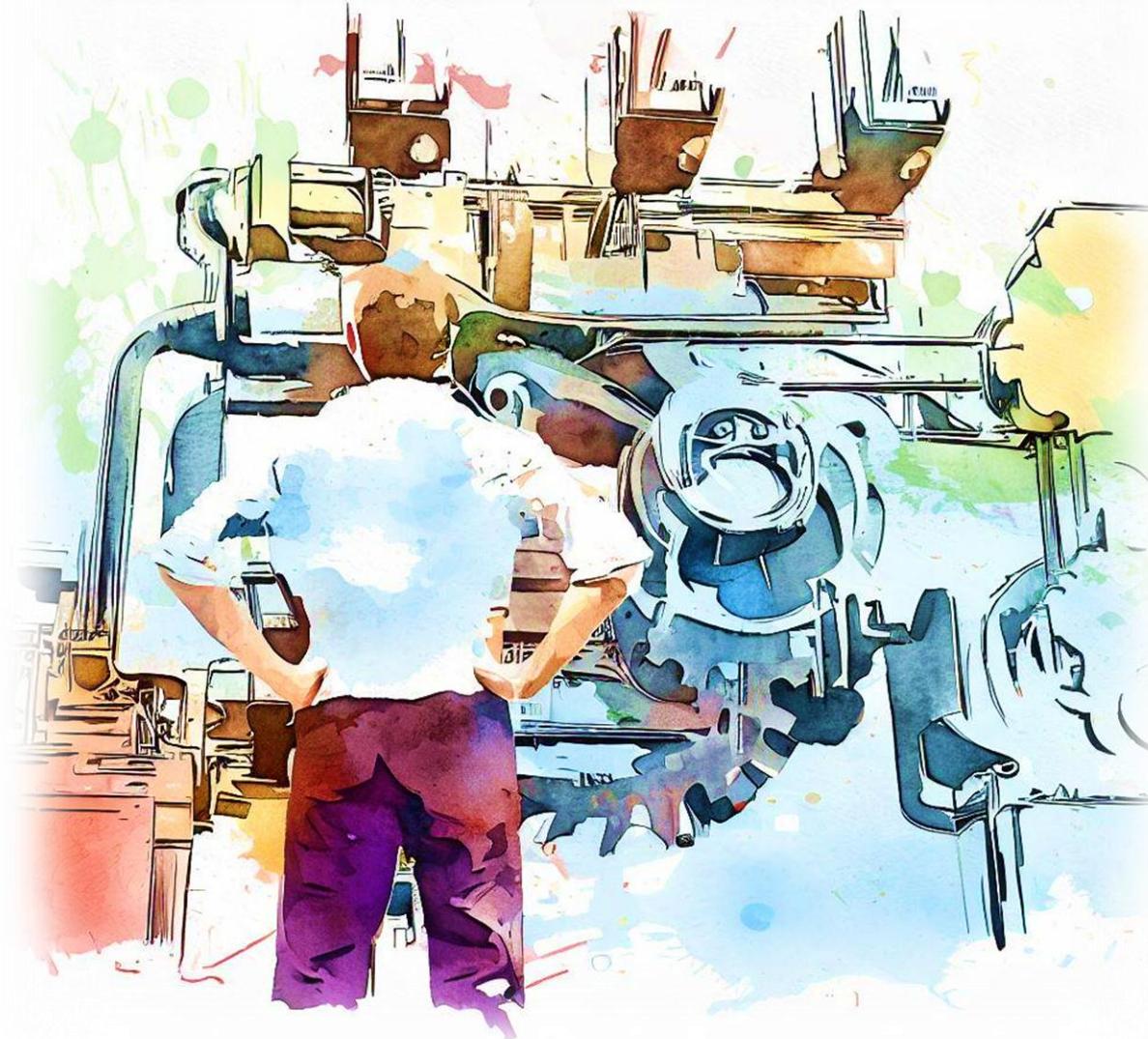


Howard Hao

Principle Engineer

Microsoft

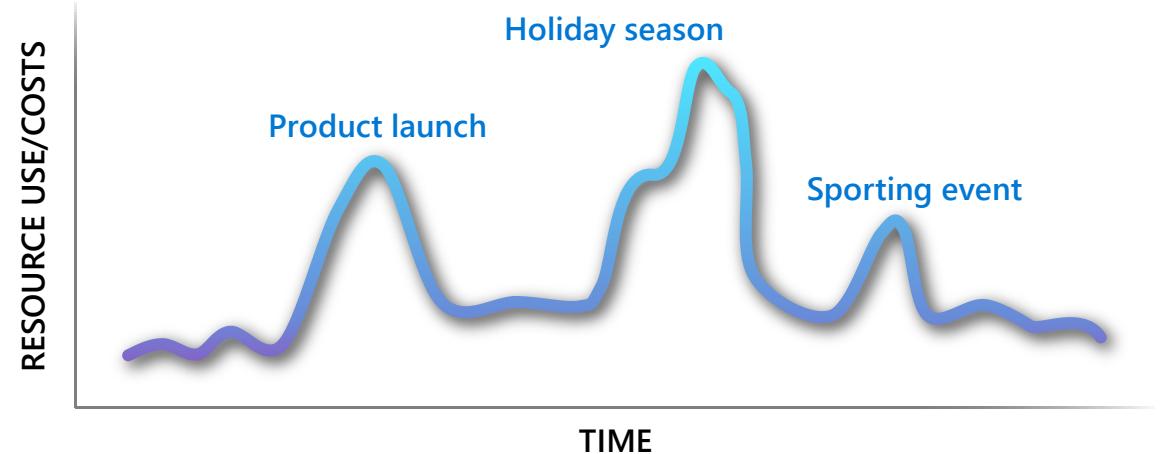
And the entire Windows container team!



"I have a legacy application and I need to find a way to modernize and save costs"

Why migrate?

- Shift CapEx to OpEx, reduce financial risk, and improve agility for macro-economic conditions
- Purpose-built compute, scale sets and rightsized workloads
- Reduce technical debt sooner with simplicity in migrating legacy systems
- Automation of infrastructure and operations



Wow this Windows container thing seems great!

Same app, better scaling, right?

Small & compact – better resource optimized than a VM

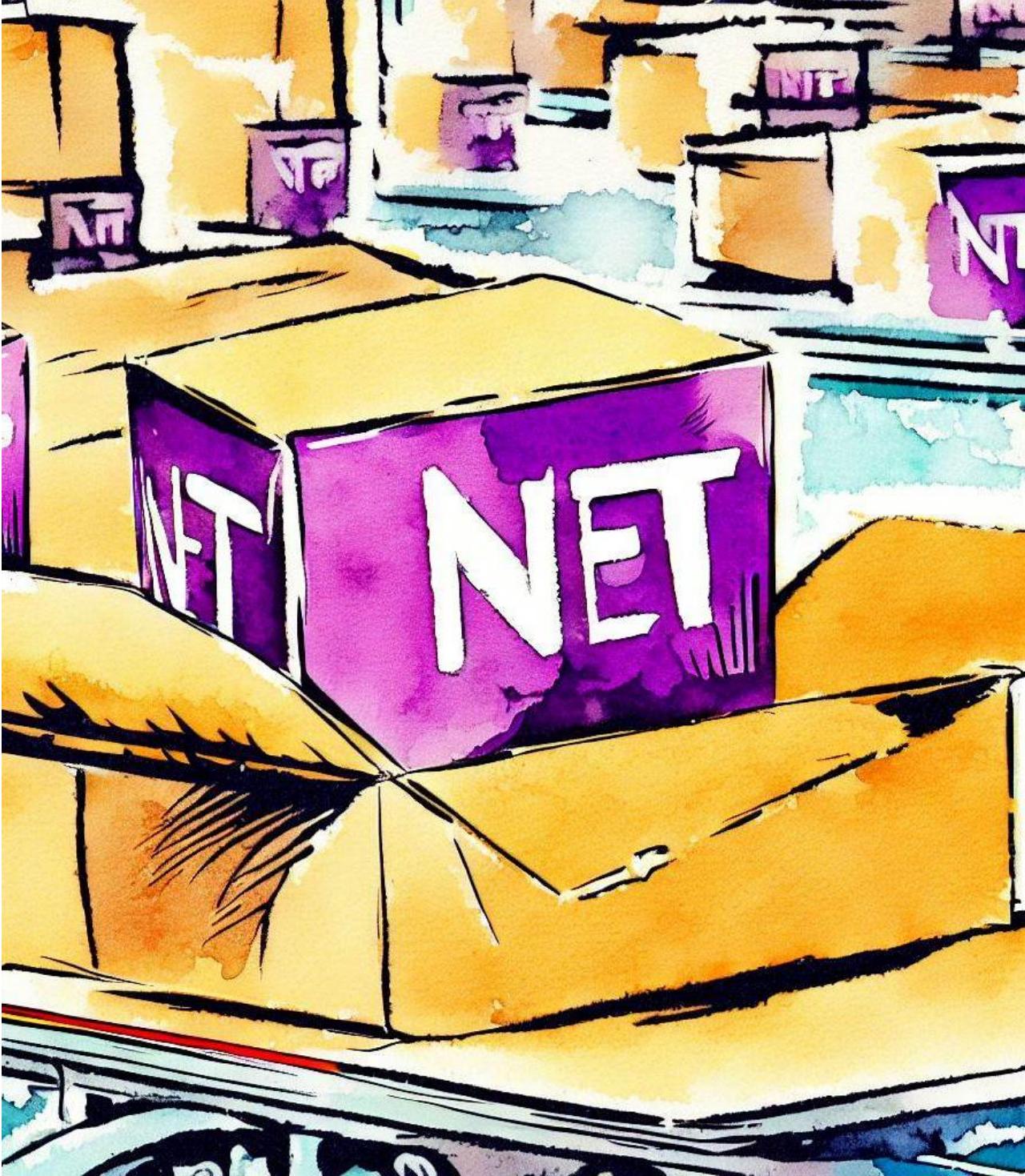
- Kernel is shared between containers, no need to run setup
- Namespace vs. hardware isolation
- Lego-like optimization of Windows server

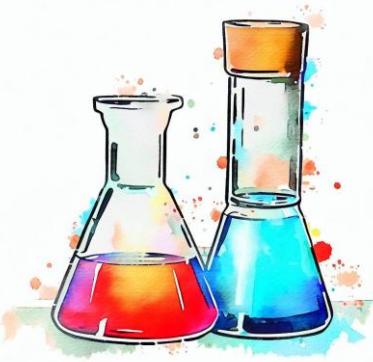
Fast scaling, more adaptive to demand

- OS image is shared across containers on the same VM
- Less overprovisioning, faster start and cleanup times
- Higher density, no manual scaling or management

Predictable deployment and consistent development

- Focus on application over VM / infra management
- More consistent CI/CD pipelines
- Extensive OSS landscape





Testing...



It works!
(On my machine)



Deploy!

OutOfMemory High CPU



Slow
response
times

Low throughput

Pod
restarts

At the end of the day, software is still a machine

And machines need to be tuned to meet a specific purpose.

So why does this happen?
And how can we solve it?

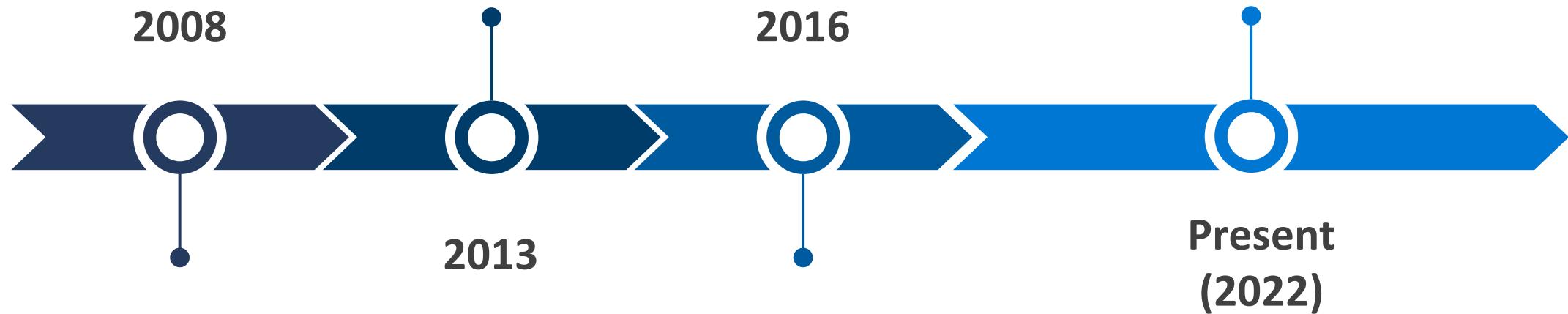


Windows containers...

How they work and why we have them

- Docker is introduced to the world
- Highly efficient image layering model and registries

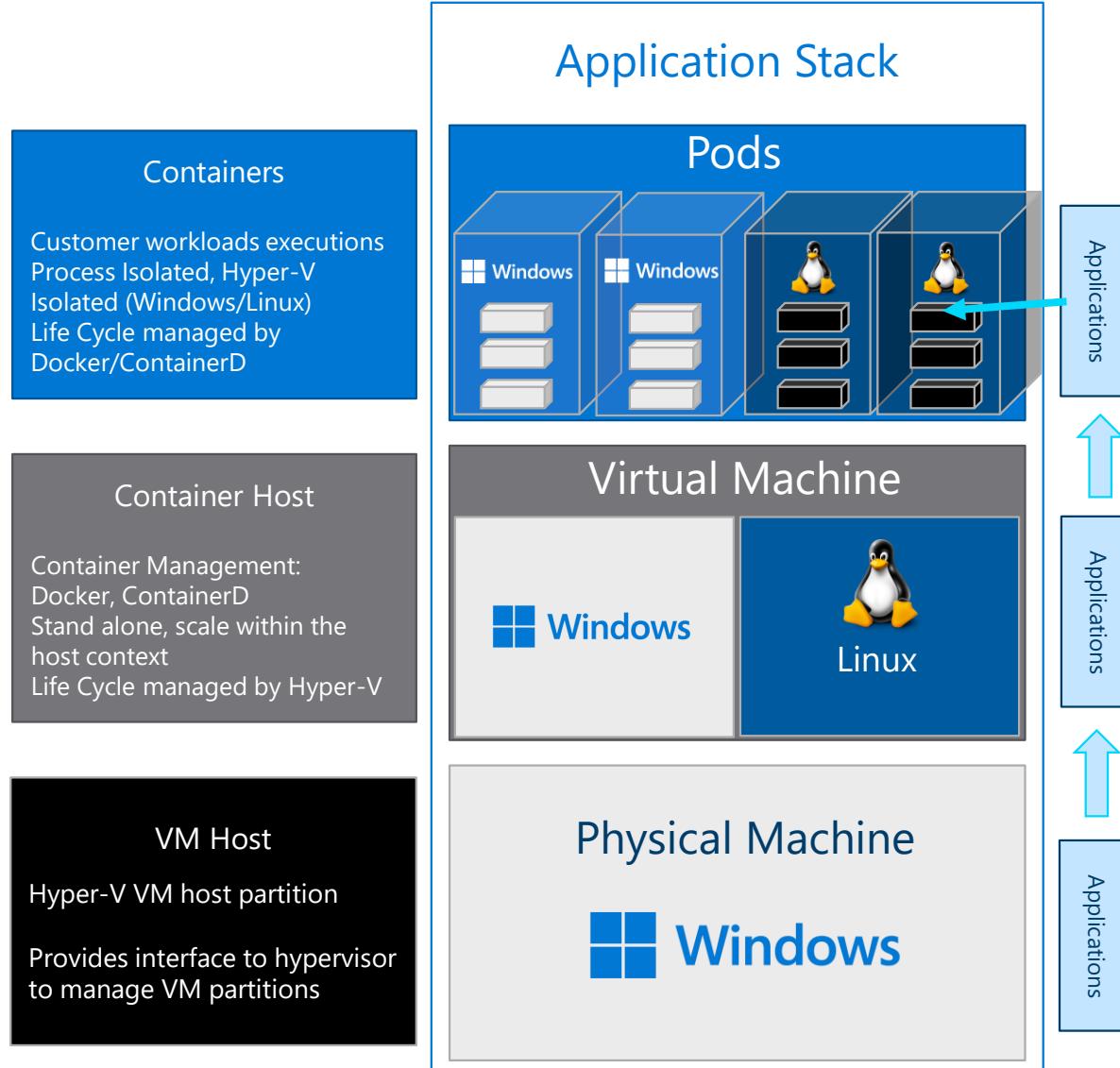
- Kubernetes is the major orchestration method for containers, with support for Windows containers
- Windows containers are available alongside Windows server 2019 and 2022



- LXC, LinuX containers were born.
- First complete implementation of the Linux container manager using cgroups and Linux namespaces.

- Microsoft releases Windows containers with Windows server 2016 built on server silos

Not your grandfather's Windows server



How do we tune Windows to meet modern demands?



The three essentials of scaling Windows on Kubernetes



Platform Optimization

- Minimizing system overhead
- Kubernetes control parameters
- Image type, version, and composition
- Effective caching



Application Optimization

- Minimizing overhead and redundancies
- Framework, Service Dependencies, Language dependencies
- Programming for the target platform



Monitoring, Trace Collection, and Analysis

- Ensure a solid feedback loop for optimization
- Experiment and observe with varying configurations (such as resource limits) in large test clusters
- OSS ecosystem support

How we're optimizing the platform for you

Through the contributions of the Windows container teams at Microsoft and the SIG-Windows community, container performance has advanced **drastically** in just **1 year**

Fast Scaling

- A brand-new container-optimized filesystem: **UnionFS + CimFS**, offering a **30%** decrease in container import and start times
- Built-in support for dynamic artifact streaming
- A **40%** reduction in Windows base image sizes, with more on the way!
- Dynamic caching of Kubernetes nodes

Analysis & Monitoring

- Enabling Windows in Prometheus with windows-exporter.
- Windows support in community projects and partnering with popular monitoring & CI vendors.
- Easy trace collection through HostProcess containers
- Intelligent analysis with the performance AutoTuner

Density & Network Performance

- Continued research into maximizing efficient resource sharing between containers and minimizing default container overhead
- 10 Pod deployment time reduced by **25 seconds (~26%)**, 20 pods by **50 seconds (~50%)**.
- Policy operation improved by **~93%**, reduced from **~70 seconds to ~5 seconds**
- Container restart volume **reduced 100x**
- CPU overhead at startup **reduced by 30%**
- **2Gb** memory leak reduction

What can you do?

Tuning & Configuration

- Ensure that you are minimizing potential for bloat and redundancy
- Perform system operations only as often as you need them
- The more you prepare ahead of time, the less overhead you experience

Container Images

1. Pick the smallest image size possible

- Use nanoserver when possible
- Servercore and full server for greater app compat

2. Minimize the number and size of your layers

3. Move to containerd and WS2022

- Huge improvements in perf and reliability with WS2022
- Latest performance enhancements are being developed for the latest OS and containerd runtime

4. Avoid frequent system information queries

- Avoid polling information by launching command line tools

5. Precache the application in a configured state

- Avoid setup at runtime in production (e.g., JIT .NET applications)
- Avoid on-the-fly compilation

6. Utilize advanced Windows features when porting

7. Decouple dependencies in application and language Frameworks

- The GetTimeZoneInformation function was always called by Golang applications.
- Fluent-bit.exe had GDI dependencies even when launched on a core image.

Troubleshooting Windows Containers

Most performance issues (application and platform) need to be investigated via the host

Things to observe:

- **System resource usage:** Which resources are being overutilized and by who?
- **Windows + Application Logs:**
 - Look for Windows events, HCS or HNS crashes, containerd logs
 - Use the Windows Performance Recorder (WPR) to collect performance traces
 - Map to a persistent volume or use the [Windows LogMonitor](#) tool to pipe to stdout.
- **Kubernetes Logs and Prometheus Endpoints**

Example 1: IIS Server Reliability Issues

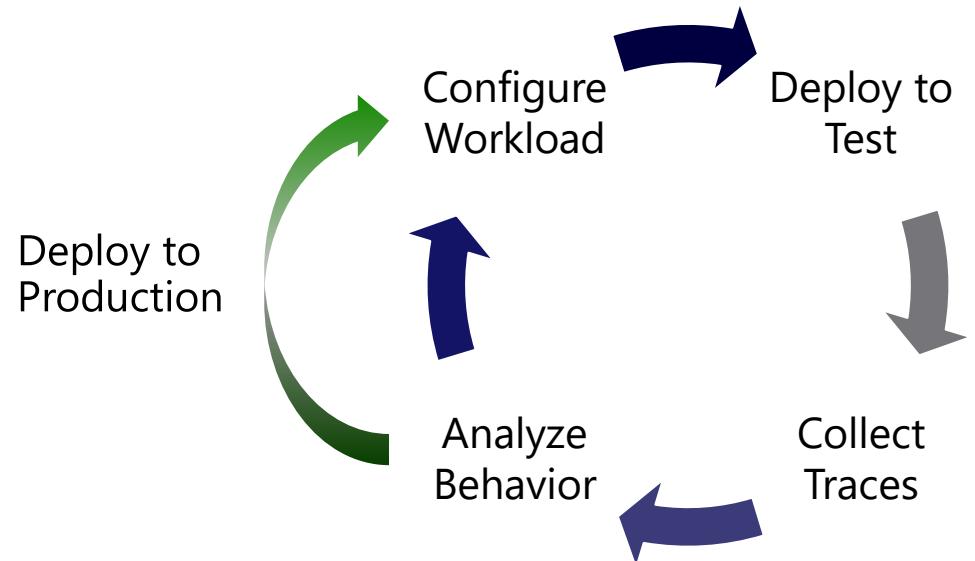
You experience 503 errors when hosted in a container

1. Deploy a HostProcess container to collect IIS logs
2. Push the logs to a mapped volume or endpoint
3. Analyze logs to root-cause issue

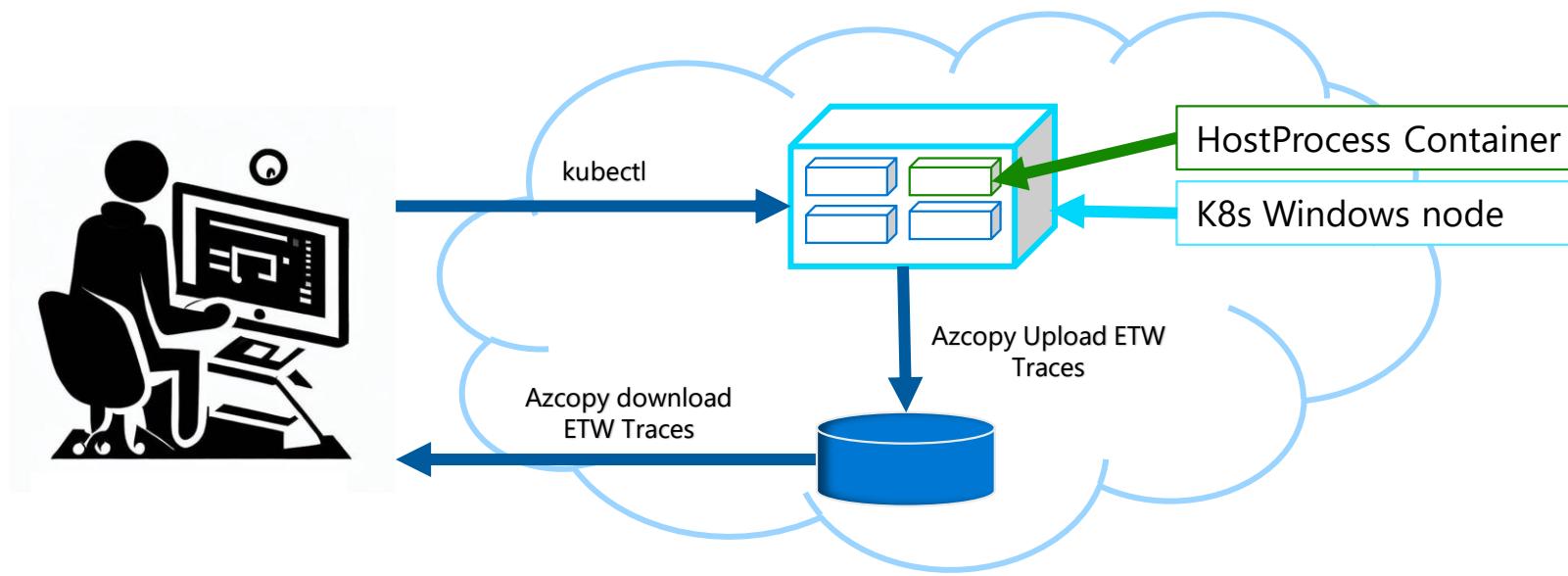
Example 2: IIS Server Performance Issues

You notice a degradation in disk I/O alongside CPU thrashing and container restarts

1. Deploy a HostProcess container to run WPR and capture an ETL trace file
`wpr -start containerplatformtrace.wprp -start cpu -start fileio -start diskio`
2. Push the trace file to a mapped volume or endpoint
3. Analyze traces, attempt to repro in test, tune configuration and redeploy



Collecting Traces with Host Process Containers (demo)



The screenshot shows two terminal windows. The left window is titled "Administrator: Windows Pow" and displays the command "hostname" followed by the output "demohost". The right window is titled "Administrator: Command Pr" and displays the command "ver" followed by the output "Microsoft Windows [Version 10.0.22621.1413]".

WSL+K8s+azcopy
Collect trace with K8s (collecttrace.sh, 147 lines)
Azure CLI : K8s cluster(createcluster.sh, 43 lines)
Storage blob(createstorage.sh, 23 lines)

Visual Studio Code to edit the bash scripts
Launch WPA.exe to investigate collected traces

File Edit Selection View Go Run ... hostsysprocess.yaml - demo - Visual Studio Code [Administrator] ⚡ ⚡ ⚡ ⚡

EXPLORER ...

OPEN EDITORS

- \$ deletestorage.sh
- \$ createstorageformat.sh
- ! myscenario.yaml
- X ! hostsysprocess.yaml 1**
- \$ createcluster.sh
- \$ collecttraces.sh
- \$ createclusterchatgpt.sh
- \$ createclusterchatgpt.s...
- \$ downloadinstallazcop...
- \$ createstorage.sh

DEMO

- createstorage.sh
- downloadazcopy-v10-linux
- downloadazcopy-v10-linux.1
- downloadazcopy-v10-wind...
- downloadazcopy-v10-wind...
- \$ downloadinstallazcopy.sh
- ! hostsysprocess.yaml 1**
- ! myscenario.yaml

```

! hostsysprocess.yaml 1 × $ createcluster.sh $ collecttraces.sh $ createclusterchatgpt.sh ▶

! hostsysprocess.yaml > YAML > {} spec > {} nodeSelector
    io.k8s.api.core.v1.Pod (v1@pod.json)
1  apiVersion: v1
2  kind: Pod
3  metadata:
4    labels:
5      pod: hostsysprocess
6      name: hostsysprocess
7  spec:
8    securityContext:
9      windowsOptions:
10        hostProcess: true
11        runAsUserName: "NT AUTHORITY\SYSTEM"
12    hostNetwork: true
13    containers:
14      - name: test
15        image: mcr.microsoft.com/oss/kubernetes/pause:3.9-windows-ltsc2022-amd64
16        imagePullPolicy: IfNotPresent
17        command:
18          - pause.exe
19    nodeSelector:
20      vm: "0"

```



EXPLORER

...

orange.sh

\$ createstorageformat.sh

! myscenario.yaml

! hostsysprocess.y



▼ OPEN EDITORS

\$ createstorageformat.sh

! myscenario.yaml

! hostsysprocess.yaml 1

\$ createcluster.sh

X \$ collecttraces.sh

\$ createclusterchatgpt.sh

\$ createclusterchatgpt.s...

\$ downloadinstallazcop...

\$ createstorage.sh

▼ DEMO

> Images

> results

\$ collecttraces.sh

\$ createcluster.sh

\$ createstorage.sh

\$ downloadinstallazcopy.sh

! hostsysprocess.yaml 1

! myscenario.yaml

\$ collecttraces.sh

127 DownloadingTraces()

128 {

129 #download trace to my local volume

130 azcopy cp "\${full_url}" ./results/\$tracefile

131 }

132

133

134 # Set the default account

135 az account set --subscription "My Subscription"

136 #step 1

137 CreateHostProcessContainer

138 #step 2

139 StartCollectingTraces

140 #step 3

141 ScenarioRepro

142 #step 4

143 StopCollectingTraces

144 #step 5

145 UploadingTraces

146 #step 6

147 DownloadingTraces

EXPLORER ... orange.sh \$ createstorageformat.sh ! myscenario.yaml ! hostsysprocess.yaml 1 \$ createcluster.sh \$ collecttraces.sh

OPEN EDITORS

- \$ createstorageformat.sh
- ! myscenario.yaml
- ! hostsysprocess.yaml 1
- \$ createcluster.sh
- X \$ collecttraces.sh
- \$ createclusterchatgpt.sh
- \$ createclusterchatgpt.s...
- \$ downloadinstallazcop...
- \$ createstorage.sh

DEMO

- > Images
- > results
- \$ collecttraces.sh
- \$ createcluster.sh
- \$ createstorage.sh
- \$ downloadinstallazcopy.sh
- ! hostsysprocess.yaml 1
- ! myscenario.yaml

\$ collecttraces.sh

```
1
2 #!/bin/bash
3
4 # This shell script demonstrates how to use a host process container to collect
5 # ETW events for troubleshooting reliability and performance issues in Windows containers.
6 #
7 # Key concepts and tools used:
8 #   1. `kubectl` to deploy pods and Label nodes
9 #   2. Host process containers
10 #   3. ETW trace event collection with wpr.exe
11 #   4. `azcopy.exe` to upload and download collected trace files
12 #   5. Interactions between WSL and Windows desktop
13 #   6. WPA to examine the collected etl file.
14
15 # By pass sudo command password
16 # sudo nano /etc/sudoers
17 # Insert the following line to sudoers
18 # <username> ALL=(ALL) NOPASSWD:ALL
19
20 # ===== Variables=====
21 # storage account for uploading collected trace files
22 rgName="kubedemo"
23 location="northeurope"
24 accountName="kubecondemostorage"
25 containerName="kebecondemocontainer"
26 sasExpiryDate=$(date -u -d '1 day' '+%Y-%m-%dT%H:%MZ')
27
28 # Define the global variable full_url as azcopy.exe target
29 declare -g full_url=""
30
31 #The trace file name
32 tracefile="mysenariotrace.etl"
33
34 #node name defined in the create cluster.sh
35 nodename=wnode
36
37 hostprocesspod=hostsysprocess
38
39 #set up alias as shortcuts
40 k=kubectl
41
42 CreateHostProcessContainer()
```



File Edit Selection View Go Run Terminal Help

collecttraces.sh - demo - Visual Studio Code [Admin]

```

$ collecttraces.sh
40 R=KubeCon
41
42 CreateHostProcessContainer()
43 {
44     #get all the pods on the current cluster before host process container
45     $k get pods -A
46
47     #Create host process container in the default name space
48     $k label nodes aks"$nodename" 000000 vm=0
49     $k delete pod $hostprocesspod
50     $k apply -f ./hostprocesspod.yaml --wait
51     $k wait --for=condition=Ready --all --timeout -1s pod
52
53     #get all the pods on the current cluster after host process container
54     $k get pods -A
55 }
56
57 StartCollectingTraces()
58 {
59     #Create d:\perf folder to collect traces
60     $k exec $hostprocesspod -- cmd.exe md d:\\perf
61
62     #Start collecting traces with etw providers
63     $k exec $hostprocesspod -- wpr.exe \
64         -start "c:\\Program Files\\Containerd\\containerplatform.wprp" \
65         -start cpu \
66         -start diskio \
67         -start fileio \
68         -compress \
69         -skipPdbGen \
70         -recordtempo d:\\\\perf
71 }
72
73 ScenarioRepro()
74 {
75     $k apply -f myscenario.yaml
76 }
77
78 StopCollectingTraces()
79 {
80     #Stop collecting traces
81     $k exec $hostprocesspod -- wpr.exe \
82         -stop d:\\perf\\$tracefile
83 }

```



```

$ collecttraces.sh
84
85 GetStorageShareURL()
86 {
87     accountkey=$(az storage account keys list \
88         --resource-group $rgName \
89         --account-name $accountName \
90         --query "[0].value" -o tsv)
91
92     # Generate a SAS token for the container
93     sastoken=$(az storage container generate-sas \
94         --account-name ${accountName} \
95         --account-key ${accountkey} \
96         --name ${containerName} \
97         --permissions "rwdl" \
98         --expiry ${sasExpiryDate} \
99         --https-only \
100        --output tsv)
101
102    # Combine the URL, share name, and file path to get the full URL path
103    echo "storage_account_url: ${storage_account_url}"
104    echo "containerName: ${containerName}"
105    echo "tracefile: ${tracefile}"
106    echo "sas_token: ${sastoken}"
107
108    full_url="https://${accountName}.blob.core.windows.net/${containerName}/${tracefile}?${sastoken}"
109
110    #full_url="${storage_account_url}/${containerName}/${tracefile}?${sastoken}"
111    echo $full_url
112 }
113
114 UploadingTraces()
115 {
116     #copy azcopy.exe to the target node with host process container
117     $k cp ./azcopy.exe $hostprocesspod:d:/perf/azcopy.exe
118     GetStorageShareURL
119
120     #copy the trace to the storage share
121     $k exec $hostprocesspod -- d:\\perf\\azcopy.exe cp d:\\perf\\$tracefile "${full_url}"
122
123     #delete d:\\perf folder to collect traces
124     $k exec $hostprocesspod -- cmd.exe delete /Y d:\\\\perf
125 }
126
127 DownloadingTraces()
128 {
129     #download trace to my Local volume
130     azcopy cp "${full_url}" ./results/$tracefile
131 }

```

Microsoft Azure



```
Administrator: Windows Powe X demo@demohc: /mnt/d/dem + v
demo@demohost:/mnt/d/demo$ bash -x collecttraces.sh
+ rgName=kubedemo
+ location=northeurope
+ accountName=kubecondemostorage
+ containerName=kebecondemoccontainer
++ date -u -d '1 day' +%Y-%m-%dT%H:%M%Z
+ sasExpiryDate=2023-04-11T03:43Z
+ declare -g full_url=
+ tracefile=mysenariotrace.etl
+ nodename=wnode
+ hostprocesspod=hostsysprocess
+ k=kubectl
+ az account set --subscription 'My Subscription'
+ CreateHostProcessContainer
+ kubectl get pods -A
NAMESPACE      NAME          READY   STATUS    RESTARTS   AGE
kube-system    azure-ip-masq-agent-pjgcv   1/1     Running   0          50m
kube-system    azure-npm-9jxxn       1/1     Running   0          50m
kube-system    azure-npm-win-lzkqm      1/1     Running   0          44m
kube-system    cloud-node-manager-fg9zx   1/1     Running   0          50m
kube-system    cloud-node-manager-windows-d6zsx  1/1     Running   0          45m
kube-system    coredns-77f75ff65d-lcwtc    1/1     Running   0          50m
kube-system    coredns-77f75ff65d-xdnqw    1/1     Running   0          50m
kube-system    coredns-autoscaler-577678f588-whv6t  1/1     Running   0          50m
kube-system    csi-azuredisk-node-65vx9     3/3     Running   0          50m
kube-system    csi-azuredisk-node-win-k5fgd    3/3     Running   0          45m
kube-system    csi-azurefile-node-q9ql5      3/3     Running   0          50m
kube-system    csi-azurefile-node-win-jwmmq    3/3     Running   0          45m
kube-system    konnectivity-agent-6c4c5f8ff9-5tmpc  1/1     Running   0          13m
kube-system    konnectivity-agent-6c4c5f8ff9-nhbff    1/1     Running   0          13m
kube-system    kube-proxy-8r8qv      1/1     Running   0          50m
kube-system    metrics-server-56c547ff47-cs7k2    2/2     Running   0          44m
kube-system    metrics-server-56c547ff47-h4wak    2/2     Running   0          44m
+ kubectl label nodes akswnode000000 vm=0
node/akswnode000000 not labeled
```

```
+ kubectl label nodes akswnode000000 vm=0
node/akswnode000000 not labeled
+ kubectl apply -f ./hostsysprocess.yaml --wait
pod/hostsysprocess created
+ kubectl wait --for=condition=Ready --all --timeout -1s pod
pod/hostsysprocess condition met
+ kubectl get pods -A
```

NAMESPACE	NAME	READY	STATUS	RESTARTS	AGE
default	hostsysprocess	1/1	Running	0	13s
kube-system	azure-ip-masq-agent-pjgcv	1/1	Running	0	51m
kube-system	azure-npm-9jxxn	1/1	Running	0	51m
kube-system	azure-npm-win-lzkqm	1/1	Running	0	45m
kube-system	cloud-node-manager-fg9zx	1/1	Running	0	51m
kube-system	cloud-node-manager-windows-d6zsx	1/1	Running	0	45m
kube-system	coredns-77f75ff65d-lcwtc	1/1	Running	0	50m
kube-system	coredns-77f75ff65d-xdnqw	1/1	Running	0	51m
kube-system	coredns-autoscaler-577678f588-whv6t	1/1	Running	0	51m
kube-system	csi-azuredisk-node-65vx9	3/3	Running	0	51m
kube-system	csi-azuredisk-node-win-k5fgd	3/3	Running	0	45m
kube-system	csi-azurefile-node-q9ql5	3/3	Running	0	51m
kube-system	csi-azurefile-node-jwmmq	3/3	Running	0	45m
kube-system	konnectivity-agent-6c4c5f8ff9-5tmpc	1/1	Running	0	14m
kube-system	konnectivity-agent-6c4c5f8ff9-nhbff	1/1	Running	0	14m
kube-system	kube-proxy-8r8qv	1/1	Running	0	51m
kube-system	metrics-server-56c547ff47-cs7k2	2/2	Running	0	44m
kube-system	metrics-server-56c547ff47-h4wqk	2/2	Running	0	44m

```
+ StartCollectingTraces
+ kubectl exec hostsysprocess -- cmd.exe md 'd:\perf'
```

```
Microsoft Windows [Version 10.0.20348.1607]
(c) Microsoft Corporation. All rights reserved.
```

```
C:\C\7451cb0f34de20a169d7d36b8e7ce5c5659335dd089a0d7ceb36fd47aff7e811>+ kubectl exec hostsysprocess -- wpr.exe -start 'c:\Program Files\Containerd\containerplatform.wpr' -start cpu -start diskio -start fileio -compress -skipPdbGen -recordtempo 'd:\perf'
+ ScenarioRepro
+ kubectl apply -f myscenario.yaml
deployment.apps/basic-deployment unchanged
+ StopCollectingTraces
+ kubectl exec hostsysprocess -- wpr.exe -stop 'd:\perf\mysenariotrace.etl'
Press Ctrl+C to cancel the stop operation.
The trace was successfully saved.
+ UploadingTraces
+ kubectl cp ./azcopy.exe hostsysprocess:d:/perf/azcopy.exe
+ GetStorageShareURL
++ az storage account keys list --resource-group kubedemo --account-name kubecondemostorage --query '[0].value' -o tsv
+ accountkey=HQsGUueTDhbiMrM8NTE1anYBw4fugxBpsN0sGyNTupzmCGzy+UTKuQqLBiBYE3746bALe2KcnN55+ASTCLYuCA==
++ az storage container generate-sas --account-name kubecondemostorage --account-key HQsGUueTDhbiMrM8NTE1anYBw4fugxBpsN0sGyNTupzmCGzy+UTKuQqLBiBYE3746bALe2KcnN55+ASTCLYuCA== --name kebecondemocontainer --permissions rwdl --expiry 2023-04-11T03:43Z --https-only --output tsv
+ sastoken='se=2023-04-11T03%3A43Z&sp=rwdl&spr=https&sv=2021-06-08&sr=c&sig=VphHT2AY0WypQhQ%2Bc1yTHixKyWJTc8glcVZSGs9jzc%3D'

+ kubectl exec hostsysprocess -- 'd:\perf\azcopy.exe' cp 'd:\perf\mysenariotrace.etl' 'https://kubecondemostorage.blob.core.windows.net/kebecondemocontainer/mysenariotrace.etl?se=2023-04-11T03%3A43Z&sp=rwdl&spr=https&sv=2021-06-08&sr=c&sig=VphHT2AY0WypQhQ%2Bc1yTHixKyWJTc8glcVZSGs9jzc%3D'
INFO: Scanning...
INFO: Any empty folders will not be processed, because source and/or destination doesn't have full folder support

Job da856f0a-bea0-9640-5786-84223801f4cc has started
Log file is located at: C:\Windows\system32\config\systemprofile\.azcopy\da856f0a-bea0-9640-5786-84223801f4cc.log

100.0 %, 1 Done, 0 Failed, 0 Pending, 0 Skipped, 1 Total, 2-sec Throughput (Mb/s): 133.5847

Job da856f0a-bea0-9640-5786-84223801f4cc summary
Elapsed Time (Minutes): 0.0335
Number of File Transfers: 1
Number of Folder Property Transfers: 0
Number of Symlink Transfers: 0
Total Number of Transfers: 1
Number of File Transfers Completed: 1
Number of Folder Transfers Completed: 0
Number of File Transfers Failed: 0
Number of Folder Transfers Failed: 0
Number of File Transfers Skipped: 0
Number of Folder Transfers Skipped: 0
TotalBytesTransferred: 33554432
Final Job Status: Completed
```

```
+ DownloadingTraces
+ azcopy cp 'https://kubecondemostorage.blob.core.windows.net/kebecondemocontainer/mysenariotrace.etl?se=2023-04-11T03%3A43Z&sp=rwdl&spr=https&sv=2021-06-08&sr=c&sig=VphHT2AY0WypQhQ%2Bc1yTHixKyWJTc8glcVZSGs9jzc%3D' ./results/mysenariotrace.etl
INFO: Scanning...
INFO: Any empty folders will not be processed, because source and/or destination doesn't have full folder support

Job 45b503fe-651c-6544-6971-007b1377aa75 has started
Log file is located at: /home/hhao/.azcopy/45b503fe-651c-6544-6971-007b1377aa75.log

100.0 %, 1 Done, 0 Failed, 0 Pending, 0 Skipped, 1 Total,

Job 45b503fe-651c-6544-6971-007b1377aa75 summary
Elapsed Time (Minutes): 0.3002
Number of File Transfers: 1
Number of Folder Property Transfers: 0
Number of Symlink Transfers: 0
Total Number of Transfers: 1
Number of File Transfers Completed: 1
Number of Folder Transfers Completed: 0
Number of File Transfers Failed: 0
Number of Folder Transfers Failed: 0
Number of File Transfers Skipped: 0
Number of Folder Transfers Skipped: 0
TotalBytesTransferred: 33554432
Final Job Status: Completed

demo@demohost:/mnt/d/demo$
```

D:\demo>dir

```
Volume in drive D is New Volume
Volume Serial Number is C4E8-750F

Directory of D:\demo

04/09/2023  08:19 PM    <DIR>          .
04/09/2023  08:19 PM    31,232,960 azcopy.exe
03/31/2023  11:12 AM    <DIR>          azcopy_linux_amd64_10.18.0
03/31/2023  11:12 AM    <DIR>          azcopy_windows_amd64_10.18.0
04/09/2023  08:43 PM    3,889 collecttraces.sh
04/09/2023  07:15 PM    1,234 createcluster.sh
04/09/2023  04:20 PM    724 createteststorage.sh
03/31/2023  11:17 AM    14,504,516 downloadazcopy-v10-linux
03/31/2023  11:17 AM    14,799,837 downloadazcopy-v10-windows
04/09/2023  08:16 PM    362 downloadinstallazcopy.sh
04/09/2023  06:50 PM    423 hostsysprocess.yaml
04/09/2023  08:53 PM    <DIR>          Images
04/09/2023  06:50 PM    634 myscenario.yaml
04/09/2023  08:47 PM    <DIR>          results
               9 File(s)   60,544,579 bytes
               5 Dir(s)  1,431,142,563,840 bytes free
```

D:\demo>cd results

D:\demo\results>dir

```
Volume in drive D is New Volume
Volume Serial Number is C4E8-750F

Directory of D:\demo\results

04/09/2023  08:47 PM    <DIR>          .
04/09/2023  08:19 PM    <DIR>          ..
04/09/2023  08:47 PM    33,554,432 mysenariotrace.etl
               1 File(s)   33,554,432 bytes
               2 Dir(s)  1,431,142,563,840 bytes free
```

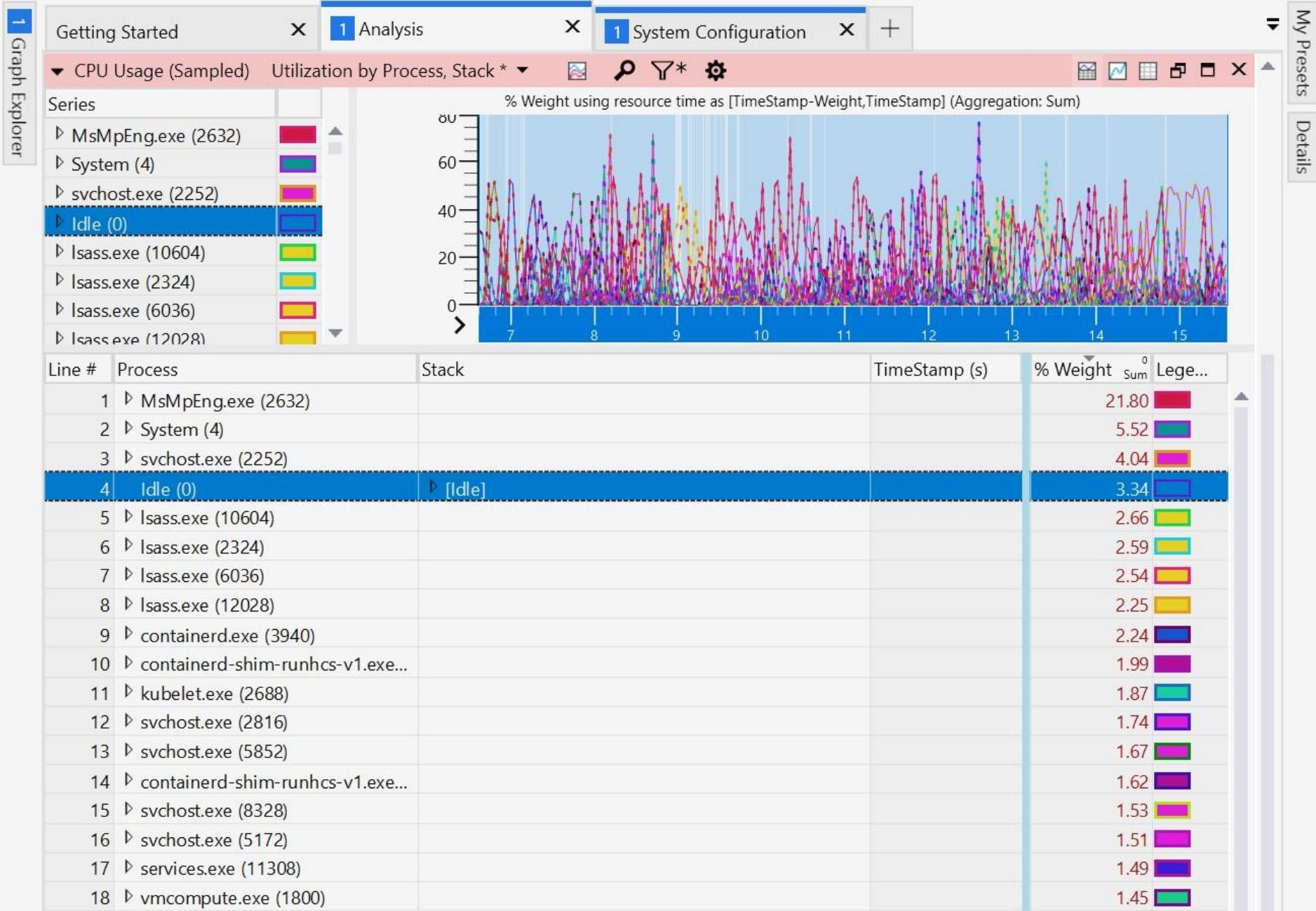
D:\demo\results>

D:\ demo\results\mysenariotrace.etl - Windows Performance Analyzer (Internal)

File Trace Profiles Window Help

1 Graph Explorer Getting Started Analysis 1 System Configuration +

Display	Line #	Configuration	Value
Flight Id	1	OS Version	10.0
General	2	Build	20348
IRQ	3	System Manufacturer	Microsoft Corporation
Network	4	System Product Name	Virtual Machine
Optical	5	BIOS Date	12/07/2018
PnP	6	BIOS Version	090008
Power Settings	7	Computer Name	akswnode000000
Services	8	Domain Name	
Storage	9	Product Name	Windows Server 2022 Datacenter
Trace Statistics	10	Build Lab	20348.1607.amd64fre.fe_release_svc_prod2.220707-1832
Traces	11	UAP Info	2814751100634695
	12	Device Family Id	9
	13	Device Form Id	0
	14	Processor Name	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz
	15	Number of Processors	2
	16	Processor Speed	2594 MHz
	17	Processor Nominal Speed	
	18	Processor 1 Nominal Speed	2594 MHz
	19	Processor 2 Nominal Speed	2594 MHz
	20	Processor Relative Performance	1:100 2:100
	21	Processor Efficiency Class	1:0 2:0
	22	Hyper-Threading Enabled Processors	0x0000000000000003
	23	Memory Size	8192 MB
	24	Page Size	4096 Bytes
	25	Allocation Granularity	65536 Bytes
	26	Supported Power States	S5
	27	Boot Drive	Disk 0 - Drive C - NTFS
	28	VBS Enabled	True
	29	HVCI Enabled	False
	30	HyperVisor Enabled	True
	31	Boot Flags	0
	32	Firmware Type	BIOS
	33	SecureBoot Capable	False
	34	SecureBoot Enabled	False



Let's work together!

Helping you helps us make Windows better

1. Making Windows performance analysis easily accessible

- More public guidance, learning paths, and tooling
- Guidance on what to look for in performance traces
- An auto-analysis and tuning service for your Windows workloads

2. Making it easier to contact Microsoft experts

- Get involved with SIG-Windows! We are involved extensively and you're welcome to bring issues and concerns to the weekly community meetings.
- Submit (sanitized) traces, logs, bugs, or feature requests to our Github repo! We triage this regularly and try to answer as many questions as we can.
- The AKS roadmap also will often include Windows features
- The Kubernetes slack channels!
- Chat with us here at KubeCon ☺

3. Share your stories!

- It's always helpful for us to hear how people are using our technology and lets us iterate and improve to empower all our users!

Kubernetes.slack.com

@Brandon Smith (MS)

<https://github.com/microsoft/Windows-Containers>

@brasmith-ms

<https://github.com/microsoft/windows-container-tools>

<https://github.com/Azure/AKS>



KubeCon



CloudNativeCon

— Europe 2023 —

Questions?

