Using Kubernetes with AIX container technology

User Guide

March 2019

Contents

1 About the software

This software allows the use of AIX nodes as worker nodes in a Kubernetes cluster managed by IBM Cloud Private. It provides a set of services to manage containers and network(s) on AIX nodes and a set of commands to create and manipulate container images on AIX. Containers on AIX are implemented using Workload Partitions (WPARs).

1.1 Overview

Figure 1-1 gives a simplified view of the software architecture.

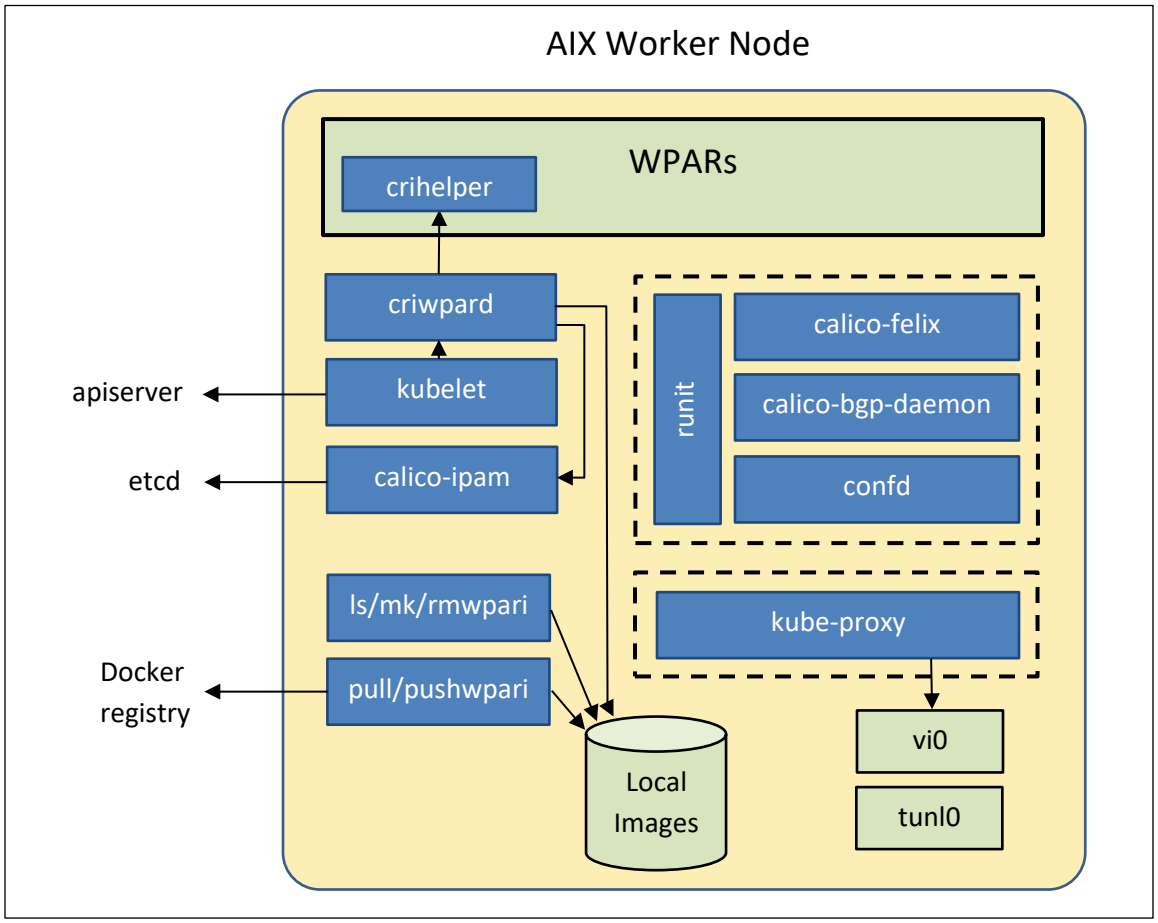

Figure 1-1 Software architecture

The `criwpard` service implements the Container Runtime Interface of Kubernetes. It runs on every AIX worker node. It is the interface between the `kubelet` service (that interacts with the Kubernetes API server running on the master nodes) and the Workload Partitions. `crihelper` is a helper process that runs in every Workload Partition. The `kube-proxy` service manages virtual IP addresses of Kubernetes services. It runs in a pod on every worker node. Calico

services manage IP addresses of Kubernetes pods, Kubernetes network policies and network routes to pods running on other nodes. They run in a pod on every worker node.

1.2 Known limitations

This section documents limitations that users should be aware of before installing the software.

1.2.1 IBM Cloud Private worker node only

Running AIX as a proxy or management Kubernetes node is not possible. Running an AIX worker node in a Kubernetes cluster *not* managed by IBM Cloud Private might be possible under some conditions but has not been tested. It would require users to prevent or manage potential compatibility issues between this software, aimed at Kubernetes version 1.12.4, and the actual Kubernetes version installed on master nodes. It would also require setting up a compatible container networking facility (this software assumes Calico). This guide only documents installation of AIX worker nodes in an IBM Cloud Private cluster, as this is the recommended way of using Kubernetes with AIX container technology.

1.2.2 Containers have shared /usr and /opt filesystems

Containers on AIX are implemented using shared Workload Partitions (WPARs). Consequently, they have their `/usr` and `/opt` filesystems mounted from the host in read-only mode. This must be considered when building container images for AIX. Applications running in AIX containers that need to write to a specific `/usr/dirusr` or `/opt/diropt` directory can overlay it using a `namefs` mount, provided that the directory already exists on the host. Be aware that this will mask the global definition of `/usr/dirusr` or `/opt/diropt`.

1.2.3 Container images portability

AIX container images must be deployed on the same AIX level they were created on. If the Kubernetes cluster contains AIX nodes that are at different levels, use Kubernetes labels and node selectors to make sure images are deployed on supported nodes.

1.2.4 One container per pod

Kubernetes pods deployed on AIX cannot have more than one container. The deployment will fail if the pod has more than one container. Consider using separate pods or sharing the same container.

1.2.5 Attribute hostNetwork

Containers with `hostNetwork` attribute set to `true` do not run inside a Workload Partition. Instead, they run inside a process group in a chrooted environment on the host. Consequently, processes running in these containers can see all the processes running on the host, including the ones from other containers. It is highly recommended to use Kubernetes security policies to prevent this kind of container from being deployed by non-administrator users.

1.2.6 Storage and file systems

AIX containers can attach to a limited set of volume types. This includes FC, iSCSI, NFS and GPFS (the latter through `hostPath` mounts only). Only volumes with JFS or JFS2 file systems can be mounted. GlusterFS is not supported.

1.2.7 IBM Cloud Private and IPsec

AIX worker nodes do not support node-to-node data plane network traffic encryption. IPsec must be disabled in the IBM Cloud Private cluster's `config.yaml` file.

1.2.8 IBM Cloud Private VMware NSX-T integration

AIX worker nodes do not support VMware NSX-T 2.0. The `network_type` attribute must be set to `calico` in the IBM Cloud Private cluster's `config.yaml` file.

2 Installation and configuration

2.1 Pre-requisites

This section lists the minimum resources required to run the software.

2.1.1 System

The following systems are required:

- A working IBM Cloud Private cluster (IBM Cloud Private 3.1.2) with one or more Linux master nodes
- One or more POWER8 or above AIX nodes (AIX 7.2 with APAR IJ04559)

Note: At least two AIX nodes are required to support failover of AIX pods.

For information on how to download and install IBM Cloud Private 3.1.2, refer to the Installing IBM® Cloud Private Cloud Native, Enterprise, and Community editions documentation. Please, also refer to the Performance Tuning section on page 21 for information on how to configure the cluster for better performance with AIX nodes.

Important: When installing IBM Cloud Private, *do not* specify the AIX nodes in the `[worker]` section of the `cluster/hosts` file.

2.1.2 Memory

A minimum of 8GB of memory is recommended for each AIX node.

2.1.3 Storage

A minimum storage of 50 GB for `/var` is recommended as shown in Table 2-1.

Note: The /var directory is the default storage location for images and WPARs. To prevent disk space issues, mount the default storage directories on separate paths that have larger disk capacities.

Table 2-1 Storage directories used

Location	Minimum disk space
`/var/lib/criwpard`	> 40 GB of disk space
`/var/lib/kubelet`	> 10 GB of disk space

2.2 Installation

The preferred installation method is to use the yum package manager as it takes care of all package dependencies. However, if the system is not connected to the Internet, RPMs can also be installed by hand.

2.2.1 Installation using yum

Installation using yum requires that the worker nodes have access to the Internet or a private yum repository on the local network. If that is not the case, follow the Installation by hand section. Please refer to the Installing yum on AIX documentation to install the yum package manager on AIX and to the YUM on AIX documentation to configure a local yum repository. Once yum is installed, run the following command as root:

Example 2-1 Installation using yum

```
# yum install icp-worker-3.1.2
```

2.2.2 Installation by hand

First, download the following RPMs from the AIX Toolbox for Linux Applications website:

- ca-certificates-2017.07.17-1.aix6.1.ppc.rpm
- calico-cni-3.3.1-1.aix7.2.ppc.rpm
- calico-node-3.3.1-1.aix7.2.ppc.rpm
- cloudctl-3.1.2.0-1.aix7.2.ppc.rpm
- criwpar-0.2.0-2.aix7.2.ppc.rpm
- icp-worker-3.1.2-1.aix7.2.ppc.rpm
- kubectl-1.12.4-1.aix7.2.ppc.rpm
- kubelet-1.12.4-1.aix7.2.ppc.rpm
- kubernetes-node-img-1.12.4-1.aix7.2.ppc.rpm

On each AIX node, run the following command as root to install the RPMs:

Example 2-2 RPM installation by hand

```
# rpm -ivh ca-certificates-2017.07.17-1.aix6.1.ppc.rpm \
           calico-cni-3.3.1-1.aix7.2.ppc.rpm \
           calico-node-3.3.1-1.aix7.2.ppc.rpm \
           cloudctl-3.1.2.0-1.aix7.2.ppc.rpm \
           criwpar-0.2.0-2.aix7.2.ppc.rpm \
           icp-worker-3.1.2-1.aix7.2.ppc.rpm \
           kubectl-1.12.4-1.aix7.2.ppc.rpm \
           kubelet-1.12.4-1.aix7.2.ppc.rpm \
           kubernetes-node-img-1.12.4-1.aix7.2.ppc.rpm
0513-071 The criwpard Subsystem has been added.
0513-071 The kubelet Subsystem has been added.
```

2.3 Configuration

Binaries are installed under `/opt/freeware/bin/`. It is recommended to add this directory to the PATH environment variable, if not already present:

Example 2-3 Setting PATH

```
# export PATH=$PATH:/opt/freeware/bin
```

2.3.1 Network

Add the IP address and hostname of the IBM Cloud Private master node to the `/etc/hosts` file of each AIX worker node:

Example 2-4 Updating /etc/hosts

```
10.1.0.1    mycluster.icp
```

Copy the `kubelet` and `kube-proxy` certificates and keys from the master node to the AIX worker nodes. On each AIX node, run:

Example 2-5 Copying certificates

```
# scp -r root@mycluster.icp:/etc/cfc/{kubelet,kube-proxy} /etc/cfc/
```

By default, AIX containers are attached to the `en0` interface of the host. To modify the default interface, edit the `/etc/criwpard/criwpard.conf` file:

Example 2-6 /etc/criwpard/criwpard.conf

```
{
  "network": {
    "host_interface": "en0"
  }
}
```

2.3.2 Clock synchronization

The clocks of AIX worker nodes should be synchronized with the rest of the cluster nodes. Clocks can be synchronized using network time protocol (NTP). For information about setting up NTP, refer to the client section of the Configuring NTP on AIX documentation.

2.3.3 Kubectl CLI

The `kubectl` command is installed automatically with the software on AIX. To configure the `kubectl` CLI on AIX nodes, use the `cloudctl login` command:

Example 2-7 Configuring kubectl

```
# cloudctl login -a https://mycluster.icp:8443 -u admin
```

```
Password>
Authenticating...
OK

Targeted account mycluster Account (id-mycluster-account)

Select a namespace:
1. cert-manager
2. Default
3. istio-system
4. kube-public
5. kube-system
6. Platform
7. Services
Enter a number> 2
Targeted namespace default

Configuring kubectl ...
Property "clusters.mycluster" unset.
Property "users.mycluster-user" unset.
Property "contexts.mycluster-context" unset.
Cluster "mycluster" set.
User "mycluster-user" set.
Context "mycluster-context" created.
Switched to context "mycluster-context".
OK

Configuring helm: /var/lib/helm
OK
```

Note: If you are behind a proxy, make sure that the `http_proxy`, `https_proxy` and `no_proxy` environment variables are set correctly such that the master node can be reached using HTTP/HTTPS.

2.3.4 Setup

A script is provided with the software to install Kubernetes daemonsets required for proper operation of AIX nodes into the IBM Cloud Private cluster. Before running the script, make sure that `kubectl` is configured with `admin` privileges. The script automatically determines the settings to use based on the pre-configured daemonsets for Linux nodes.

Example 2-8 Initial setup

```
# /opt/freeware/bin/k8s-setup.sh
```

```
This script installs Kubernetes daemonsets for AIX nodes in the IBM Cloud Private
cluster.
kubectl CLI must be configured with "admin" privileges on the node before running
this script.
You can run 'cloudctl login -a https://<mycluster.icp>:8443 -u admin' for this
purpose.

Continue [y/N]? y

Checking kubectl... done

Local network interface from /etc/criwpard/criwpard.conf: en0
Local IP address determined from network interface: 10.1.0.2
Found IP address of Kubernetes master node: 10.1.0.1
Found cluster CIDR: 10.1.0.0/16
Found IP-in-IP setting: Always

Creating /etc/cfc/pods/kube-proxy.json...done
Install calico-node-aix daemonset [y/N]? y
Removing calico-node-aix daemonset... done
Patching calico-node daemonset...done
Installing calico-node-aix daemonset... daemonset.extensions/calico-node-aix created
done
```

2.4 Startup

To start Kubernetes services on the AIX node, run:

Example 2-9 Starting Kubernetes services by hand

```
# startsrc -g kube-system
```

To configure Kubernetes services to automatically start after boot, run:

Example 2-10 Starting Kubernetes services during boot

```
# cp /opt/freeware/share/kubernetes/Skube /etc/rc.d/rc2.d/
```

Once Kubernetes services are started, the AIX node should be visible from the Kubernetes cluster and have a status of Ready:

Example 2-11 Listing Kubernetes cluster nodes

```
# kubectl get nodes
NAME            STATUS   ROLES                          AGE   VERSION
10.1.0.1        Ready    etcd,management,master,proxy   3d       v1.12.4+icp
10.1.0.2        Ready    worker                         1d       v1.12.4+aix
```

2.5 Uninstallation

If the software was installed with yum, simply run the following command to remove the software and its dependencies:

Example 2-12 Uninstalling with yum

```
# yum remove icp-worker
```

If the software was installed with the rpm command, run:

Example 2-13 Uninstalling with rpm

```
# rpm -e calico-cni calico-node cloudctl criwpar icp-worker kubectl kubelet \
         kubernetes-node-img
```

3 Container Images

Container images on AIX are `savewpar` backup files stored in a Docker™-compatible format. They are made of a single layer. Given that they are Docker™-compatible, they can be pushed and pulled from any registry implementing the Docker™ Registry HTTP API V2 specification, including the registry available in IBM Cloud Private.

3.1 Commands

The software comes with the following commands to manipulate container images on AIX:

- `mkwpari` – Create AIX container images
- `lswpari` – List local images
- `rmwpari` – Remove local images or images stored in a Docker™ registry
- `pushwpari` – Push local images to a Docker™ registry
- `pullwpari` – Pull images from a Docker™ registry

Table 3-1 shows the Docker™ commands that are used to manipulate images and their equivalent commands on AIX:

Table 3-1 Docker-equivalent commands on AIX

Docker™ command	**AIX command**
`docker images`	`lswpari`
`docker inspect <image>`	`lswpari <image>`
`docker rmi <image>`	`rmwpari <image>`
`docker push <image>`	`pushwpari <image> <remote>`
`docker pull <image>`	`pullwpari <image>`
`docker tag <source> <target>`	`mkwpari -T <source> <target>`

`docker build -t <image> \`
`  --build-arg <arg=value> \`
`  -f <Dockerfile> .`
	`mkwpari -a <arg=value> \`
	`  -f <Dockerfile> <image>`

`docker login <server>`	Use `-u <user>` and `-p <password>` flags in `pushwpari` and `pullwpari`.

3.2 Pre-installed images

The software comes with a set of pre-defined images that it needs for normal operations. These images should not be removed.

Example 3-1 Pre-defined images

```
# lswpari
NAME                              CREATED        SIZE
ibmcom/calico-cni-aix:v3.3.1      2 weeks ago    88.5MB
ibmcom/calico-node-aix:v3.3.1     2 weeks ago    237.2MB
ibmcom/hyperkube-aix:v1.12.4      2 weeks ago    105.0MB
```

3.3 Creating images

On AIX, the command used to create container images is `mkwpari`. This command can be used to create images from scratch or based on an existing AIX container image with a Dockerfile-compatible syntax. It can also be used to create container images from `savewpar` backup files.

3.3.1 Using Dockerfiles

Please refer to the Dockerfile reference document for information about the Dockerfile syntax and the list of supported Dockerfile commands.

Note: `mkwpari` only supports the current working directory as the build context. `.dockerignore` files are not supported.

To create an image from a Dockerfile, use the `mkwpari` command with the `-f` parameter. A dash character can be specified as the filename to read from standard input instead of a file. Example 3-2 creates a container image named `testimage:1.0` that, when deployed, runs the `vmstat` command with a default interval of 1 second. The interval can be changed at container deployment time through the `VMSTAT_INTERVAL` environment variable of the container.

Example 3-2 Creating an image from scratch

```
# mkwpari -f - testimage:1.0 << "EOF"
FROM scratch
ENV PATH /usr/bin:/etc:/usr/sbin:/sbin
ENV VMSTAT_INTERVAL 1
CMD vmstat ${VMSTAT_INTERVAL}
EOF
```

A new version of the image that adds an extra `VMSTAT_COUNT` parameter and that changes the entry-point command can then be created from the initial one, as shown in Example 3-3.

Example 3-3 Creating an image from an existing one

```
# mkwpari -f - testimage:1.1 << "EOF"
FROM testimage:1.0
ENV VMSTAT_COUNT 100
CMD vmstat ${VMSTAT_INTERVAL} ${VMSTAT_COUNT}
EOF
```

Note: Images created by `mkwpari` from Dockerfiles have their `/usr` and `/opt` filesystems mounted read-only from the host. Consequently, Dockerfile ADD and COPY commands cannot be used to copy files to `/usr` or `/opt`.

Build arguments can be passed to `mkwpari` with the `-a` parameter, as shown in Example 3-4.

Example 3-4 Passing arguments to the Dockerfile

```
# mkwpari -a default_count=100 -f – testimage:1.1 << "EOF"
FROM testimage:1.0
ARG default_count
ENV VMSTAT_COUNT ${default_count}
CMD vmstat ${VMSTAT_INTERVAL} ${VMSTAT_COUNT}
EOF
```

Creating an image from scratch can take several minutes. It is recommended to do this process only once and then to create other images based on this initial image. When creating a container from scratch, many daemons and network services are enabled by default (the same services that are enabled by default in a shared Workload Partition). If you do not want to run any service in your container, you can add the following commands to your Dockerfile:

Example 3-5 Dockerfile removing default services

```
FROM scratch
# Disable most services
RUN rmitab rctcpip \
    && rmitab rcnfs \
    && rmitab cron \
    && rmitab syslogc \
    && rmitab qdaemon \
    && rmitab writesrv
# Disable sshd
RUN installp -u -Or openssh.base.server
```

3.3.2 Using existing savewpar backup files

AIX container images can also be created from existing Workload Partitions. Beware that only files from the Workload Partition are preserved. Network settings, devices etc… are not preserved.

Example 3-6 creates a container image named `testimage:1.0` from Workload Partition `mywpar`. When deployed, the image runs the `vmstat` command with a default interval of 1 second. The interval can be changed at container deployment time through the `VMSTAT_INTERVAL` environment variable of the container:

Example 3-6 Creating an image from a backup file

```
# savewpar -f /tmp/mywpar.bff mywpar
# mkwpari -e PATH=/usr/bin:/etc:/usr/sbin:/sbin \
    -e VMSTAT_INTERVAL=1 -c vmstat -c '${VMSTAT_INTERVAL}' \
    /tmp/mywpar.bff testimage:1.0
# rm /tmp/mywpar.bff
```

Note: Deploying images created from WPARs with private `/usr` and `/opt` filesystems can take several minutes. Deployment time is proportional to the size of the image.

3.4 Tagging images

To create an alternative name for an existing image, use `mkwpari` with the `-T` flag:

Example 3-7 Tagging an image

```
# mkwpari -T testimage:1.1 testimage:latest
```

Note that the new image is just an alias, so it does not consume additional disk space.

3.5 Pushing images

Once an image is created, it can be made available to all the nodes in the Kubernetes cluster by pushing it to a Docker™ registry using the `pushwpari` command.

Important: AIX container images created with `mkwpari` contain licensed materials that are property of IBM. They cannot be freely redistributed and must not be shared on public Docker™ registries. They are subject to the same restrictions as `savewpar` backup files.

For Docker™ registries that require authentication using TLS certificates, the certificates must be copied to `/etc/criwpard/certs/<server:port>/`. For example, to copy the certificates from the IBM Cloud Private master node to the AIX node, run:

Example 3-8 Copying certificates

```
# mkdir -p /etc/criwpard/certs/mycluster.icp:8500
# scp root@mycluster.icp:/etc/docker/certs.d/mycluster.icp:8500/ca.crt
/etc/criwpard/certs/mycluster.icp:8500/
```

The image can now be pushed with:

Example 3-9 Pushing an image

```
# pushwpari -u admin testimage:1.1 mycluster.icp:8500/default/testimage:1.1
Password:
Uploading configuration file
sha256:eb9a50e6f340d006d321cac16d523afa5f0324655ef108b1233f1e0b8fcbf4d9
Uploading layer
sha256:84a4ed9f683679b3bfa39ae9f71474253b36c3a1fdd8a4a1373d6b243f717455
Uploading manifest
Done
```

The image should be visible from the IBM Cloud Private Dashboard in **Container Images:**

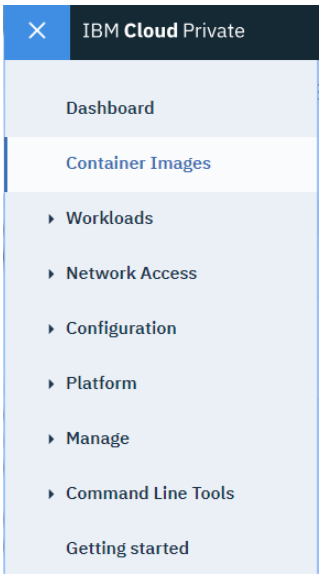

Figure 3-1 IBM Cloud Private: Main menu

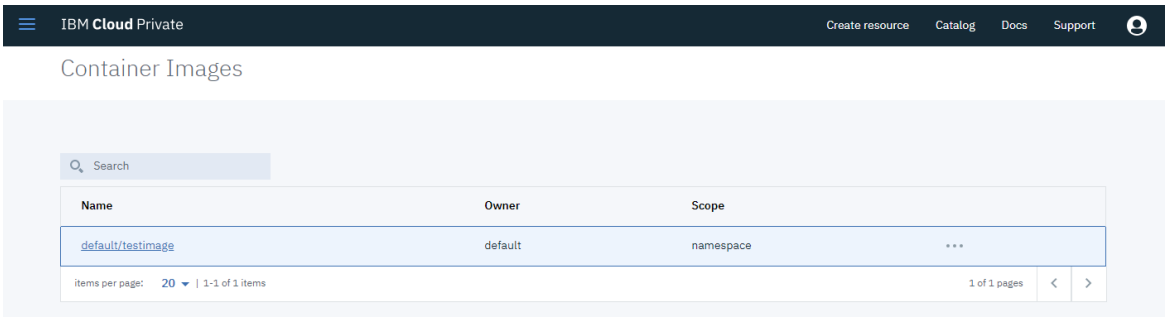

Figure 3-2 IBM Cloud Private: Container Images

The image can be removed from the registry directly from the IBM Cloud Private Dashboard interface or by using the `rmwpari` command with the `-r` (remote) parameter as shown in Example 3-10.

Example 3-10 Removing a remote image

```
# rmwpari -r -u admin mycluster.icp:8500/default/testimage:1.1
Password:
```

3.6 Pulling images

There is usually no need to pull an image by hand. Images are pulled automatically if not already present locally on the host when containers are deployed. It is still possible to pull images by hand using the `pullwpari` command as shown in Example 3-11.

Example 3-11 Pulling an image

```
# pullwpari -u admin mycluster.icp:8500/default/testimage:1.1
Password:
```

3.7 Deploying images

Once an image is created and pushed to a Docker™ registry, it can be deployed the normal way using Kubernetes. Please refer to the Pod Overview document to learn how to deploy pods with Kubernetes.

Note: To prevent the undesirable scheduling of Linux pods on AIX nodes, AIX nodes have a `NoSchedule` taint by default. A toleration must be added to pods that can be run on AIX.

In addition to the toleration, a node selector should be added to prevent the pod from being scheduled to non-AIX nodes. Example 3-12 shows a YAML specification of a pod to be run on AIX nodes only.

Example 3-12 testpod.yaml

```
apiVersion: v1
kind: Pod
metadata:
  name: testpod
spec:
  containers:
  - name: vmstat
    image: mycluster.icp:8500/default/testimage:1.1
    env:
    - name: VMSTAT_INTERVAL
      value: "2"
  restartPolicy: Never
  tolerations:
  - key: "ibm.com/aix"                          ← AIX toleration
    operator: "Exists"
    effect: "NoSchedule"
  nodeSelector:
    beta.kubernetes.io/os: aix                  ← AIX node selector
```

To deploy this pod, run:

Example 3-13 Deploying a Kubernetes pod

```
# kubectl create -f testpod.yaml
pod "testpod" created
```

To view the logs produced by the pod, run:

Example 3-14 Displaying logs

```
# kubectl logs -f testpod
```

```
k8s_vmstat_1 configuration: @lcpu=16 @mem=8192MB @ent=0.40

kthr    memory              page              faults              cpu
----- ----------- ------------------------ ------------ -----------------------
 r  b   avm   fre  re  pi  po  fr   sr  cy  in   sy  cs us sy id wa    pc    ec
 0  0 1194010 536054   0   0   0    0    0   0   7 134645 583 27 12 61  0  0.27  67.3
 0  0 1194014 536050   0   0   0    0    0   0   3  631 335  1  1 98  0  0.02   5.5
 0  0 1194015 536049   0   0   0    0    0   0   6  755 331  1  2 98  0  0.02   5.8
```

To delete the pod, simply run:

Example 3-15 Deleting a Kubernetes pod

```
# kubectl delete pod testpod
pod "testpod" deleted
```

4 Volumes

4.1 Volume types

AIX supports the following volume types:

- FC
- iSCSI
- NFS
- GPFS (through `hostPath` mounts)

Logical volume types (`configMap`, `downwardAPI`, `emptyDir`, `hostPath` and `secret`) are also supported. Please refer to the Volumes section in the Kubernetes documentation to learn how to setup persistent volumes with Kubernetes.

Note: Only volumes using JFS or JFS2 file systems can be mounted.

4.2 IBM Spectrum Scale (GPFS)

Dynamic storage provisioning with IBM Spectrum Scale is not supported. Please refer to the Using IBM Spectrum Scale™ for storage in your IBM® Cloud Private cluster documentation to learn how to use GPFS through `hostPath` mounts.

4.3 Kubernetes Raw Block Volumes

Kubernetes raw block volumes (alpha feature) can be created from FC volumes only. The `BlockVolume` feature gate must be enabled on the apiserver, controller-manager and kubelet as shown in Example 4-1. Please refer to Customizing the cluster with the config.yaml file documentation to enable this feature.

Example 4-1 Enabling BlockVolume feature in IBM Cloud Private's config.yaml file

```
## Kubernetes Settings
kubelet_extra_args: ["--feature-gates=BlockVolume=true"]
kube_apiserver_extra_args: ["--feature-gates=BlockVolume=true"]
kube_controller_manager_extra_args: ["--feature-gates=BlockVolume=true"]
```

Note: The `devicePath` attribute is ignored on AIX and the disk appears as a normal `hdiskX` device inside the Workload Partition implementing the container.

5 Helm Charts

The `helm` command can optionally be installed on AIX. If the yum package manager is installed on the system, run:

Example 5-1 Installing helm using yum

```
# yum install helm-2.9.1
```

Otherwise, download the helm-2.9.1-1.aix7.2.ppc.rpm RPM from the AIX Toolbox for Linux Applications website and run:

Example 5-2 Installing helm by hand

```
# rpm -i helm-2.9.1-1.aix7.2.ppc.rpm
```

Please, refer to Setting up the Helm CLI to learn how use Helm in an IBM Cloud Private environment.

6 Performance Tuning

This section gives some hints to improve performance when using IBM Cloud Private with AIX worker nodes.

6.1 IP-in-IP tunneling

IBM Cloud Private uses Calico to manage network communications in the Kubernetes cluster. By default, IBM Cloud Private is configured to use IP-in-IP tunnels for pod to pod traffic across nodes. This can limit network throughput considerably because of the reduced MTU. Therefore, if all the nodes in the IBM Cloud Private cluster are in the same subnet, network throughput can be improved by disabling the use of IP-in-IP tunnels. Refer to the documentation of the `calico_ipip_enabled` setting in Customizing the cluster with the config.yaml file.

After changing the `calico_ipip_enabled` setting in IBM Cloud Private, it is required to reinstall the Calico daemonsets for AIX. This can be done from a single AIX node as shown in Example 6-1.

Example 6-1 Updating IBM Cloud Private daemonsets for AIX nodes

```
# /opt/freeware/bin/k8s-setup.sh
```

6.2 Kubernetes services

Kube-proxy on AIX runs in userspace mode. This can limit the network throughput of outgoing connections to Kubernetes services. If network throughput is important, try to use the IP address of the pod you want to communicate with and avoid using IP addresses of network services such that the traffic is not routed through the userspace proxy.

7 Troubleshooting

7.1 Logging

Both `kubelet` and `criwpard` services use syslog with the `local0` facility for logging. Example 7-1 shows a syslog configuration that uses log rotation with compression of older logs.

Example 7-1 /etc/syslog.conf

```
local0.debug    /var/log/kubernetes.log rotate size 1m files 8 compress
```

For this change to take effect, restart syslog as shown in Example 7-2.

Example 7-2 Restarting syslogd

```
# touch /var/log/kubernetes.log
# refresh -s syslogd
0513-095 The request for subsystem refresh was completed successfully.
```

Container logs can also be inspected using the `kubectl logs` command.

7.2 Cleanup

If `criwpard` terminates abruptly, pods and Workload Partitions that are left running can be removed with `criwpard -c` (cleanup) as shown in Example 7-3.

Example 7-3 Cleaning up pods and WPARs

```
# stopsrc -s criwpard
# criwpard -c
```

Note that Workload Partitions managed by `criwpard` are prefixed by "`k8s_`".

7.3 Calicoctl

The Calico command line tool, `calicoctl`, is available on AIX. If the yum package manager is installed on the system, run:

Example 7-4 Installing calicoctl using yum

```
# yum install calicoctl-3.3.1
```

Otherwise, download the calicoctl-3.1.3-1.aix7.2.ppc.rpm RPM from the AIX Toolbox for Linux Applications website and run:

Example 7-5 Installing calicoctl by hand

```
# rpm -i calicoctl-3.3.1-1.aix7.2.ppc.rpm
```

Please refer to Setting up the Calico CLI to setup the Calico command line for use with IBM Cloud Private.

Online resources

These websites are also relevant as further information sources:

- *IBM® Cloud Private v3.1.2 documentation*
  https://www.ibm.com/support/knowledgecenter/SSBS6K_3.1.2/kc_welcome_containers.html
- *Kubernetes Documentation*
  https://kubernetes.io/docs/home/
- *Docker Documentation*
  https://docs.docker.com/
  https://docs.docker.com/engine/reference/builder/
- *AIX Toolbox for Linux Applications*
  https://www.ibm.com/developerworks/aix/library/aix-toolbox/
  https://public.dhe.ibm.com/aix/freeSoftware/aixtoolbox/ezinstall/ppc/README-yum