

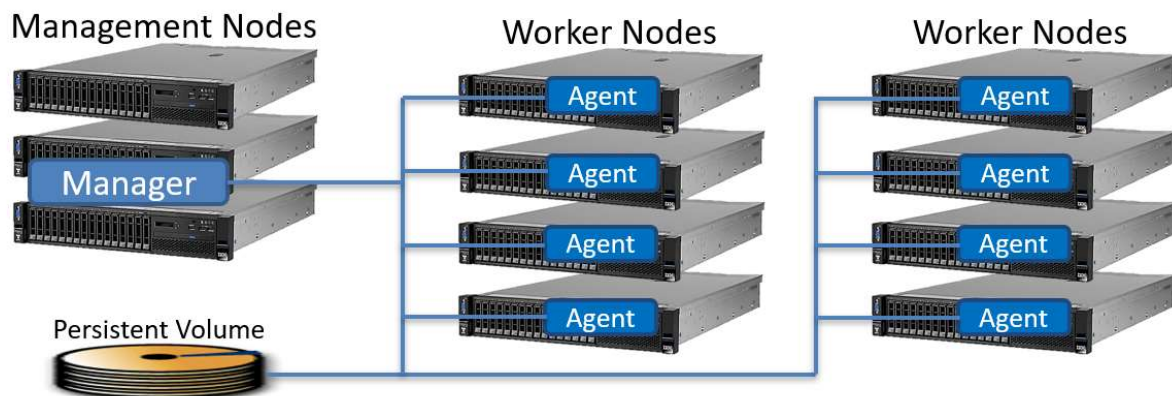
IBM Spectrum Computing Cloud Pak Quickstart Guide

This documents how to install the IBM Spectrum Computing Cloud Pak on IBM Cloud Private 3.1.x. This type of installation adds new features to Kubernetes for running jobs, including the ability to perform the following:

- Schedule complex jobs
- Run parallel jobs
- Prioritize jobs
- Schedule GPU jobs with consideration of CPU/GPU topology
- Share resources equitably among many users

This installation takes an existing IBM Cloud Private cluster and deploys the components needed to provide enhanced scheduling. Existing IBM Spectrum LSF users that want to have better service capabilities can try the alternative installation that adds Kubernetes to a portion of an existing LSF cluster.

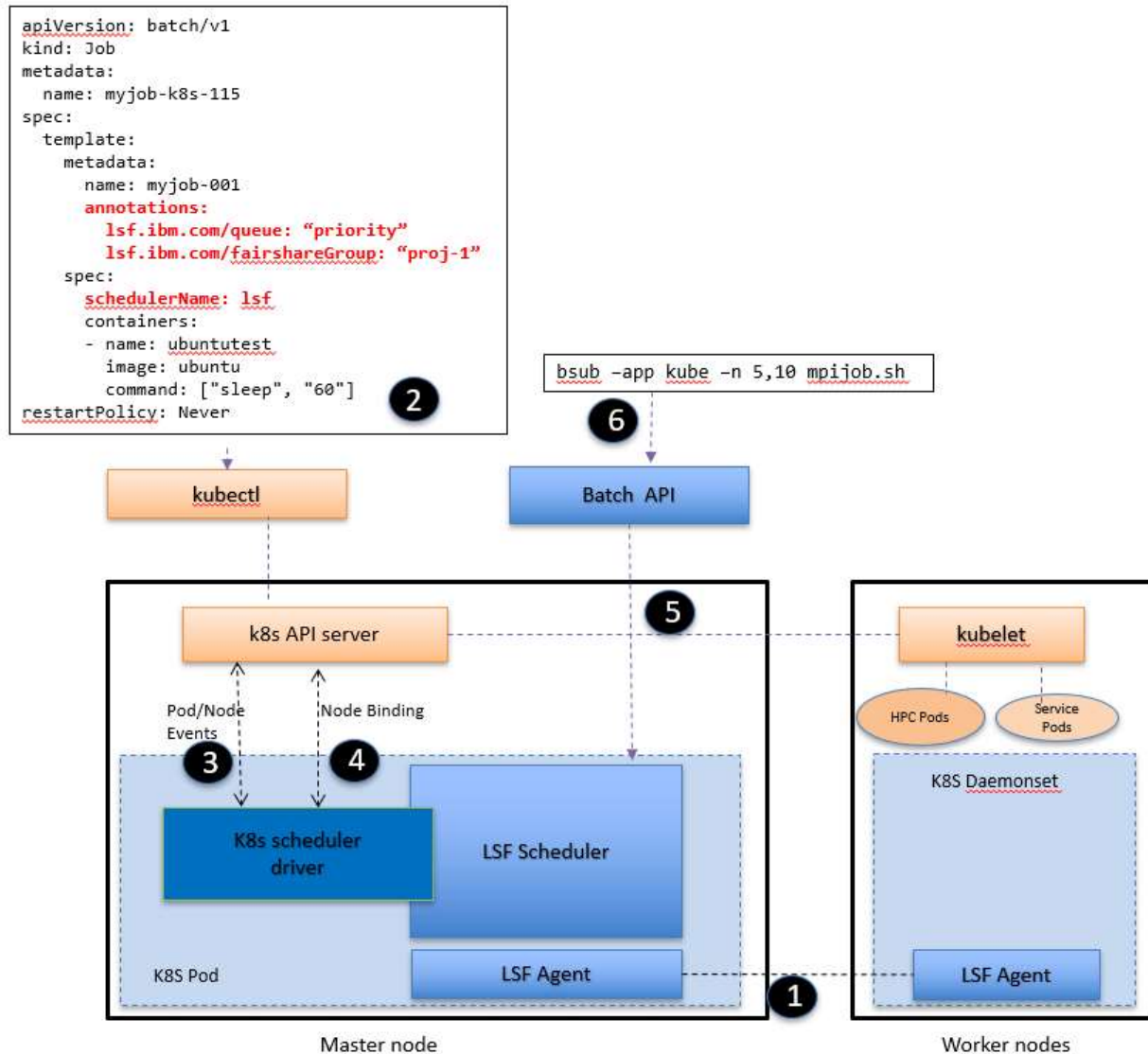
Once the Cloud Pak is deployed you will have a cluster like the following example:



Agents will be deployed on all the worked nodes. These are deployed as a daemonset and gather information for the Manager to use for job execution and data collection. One of the management nodes will run the Manager pod. The Manager pod runs the scheduling and resource management processes needed for the jobs. The persistent volume provides a persistent volume claim that all the pods will use for configuration. The Manager pod will also use this to store job data.

Overview

We have taken the core LSF scheduling technology and integrated into Kubernetes to combine the expressive power of the Kubernetes API with the rich resource sharing and load-balancing technology that is at the heart of LSF. Here is how it works:



1. The LSF Scheduler components are packaged into containers and a Helm chart is provided to deploy into the IBM Cloud Private environment.
2. Users submit workload into K8S API via `kubectl`. To get the LSF Scheduler to be aware of the pod the "schedulerName" field must be set, otherwise the pod will be scheduled by the default scheduler. Scheduler directives can be specified using annotations in the pod.
3. In order to be aware of the status of pods and nodes, the LSF Scheduler uses a driver that listens to Kubernetes API server and translates pod requests into jobs in the LSF Scheduler.
4. Once the LSF Scheduler makes a policy decision on where to schedule the pod, the driver will bind the pod to specific node.

5. The Kubelet will execute and manages pod lifecycle on target nodes in the normal fashion.
6. The LSF Scheduler also supports jobs submitted from the native “bsub” CLI which are mapped to K8S pods and executed by Kubelet as well. In this way it is consistent.

Installation and Evaluation Steps

Perform the following steps to evaluate the IBM Spectrum Computing Cloud Pak:

1. Install the prerequisites
2. Deploy the Cloud Pak
3. Verify the Cloud Pak deployment
4. Deploy the jobs
5. Going Further

The following sections will explain how to complete the steps.

Install the Prerequisites

The following are needed to evaluate the Cloud Pak:

- An installation of IBM Cloud Private 3.1.x with:
 - At least two machines with one dedicated “worker” node.
 - The “cloudctl” command line installed.
 - Optionally the “kubectl” and “helm” command line interfaces.
- Dedicated persistent volume for the deployment to use.

At least two machines are required for the evaluation. During the installation of IBM Cloud Private the `cluster/hosts` file controls the role of the machines in the cluster. A sample `cluster/hosts` file might look like the following:

```
[master]
10.10.10.10

[worker]
10.10.10.20
10.10.10.21
10.10.10.22
```

The `[master]` section defines which machines will run the IBM Cloud Private management functions. One of these machines will run the Manager pod. The `[worker]` section defines the list of machines that will run the services and jobs. The list of workers should not include any machine listed in the `[master]` section, otherwise it will not be available to run jobs.

Once the IBM Cloud Private cluster is installed, the command line interfaces can be installed. For more instructions on how to install these commands, refer to the following:

- Installation of the “cloudctl” cli:
https://www.ibm.com/support/knowledgecenter/en/SSBS6K_3.1.2/manage_cluster/icp_cli.html
- Installation of the “kubectl” cli:
https://www.ibm.com/support/knowledgecenter/en/SSBS6K_3.1.2/manage_cluster/install_kubectl.html
- Installation of the “helm” cli:
https://www.ibm.com/support/knowledgecenter/en/SSBS6K_3.1.2/app_center/create_helm_cli.html

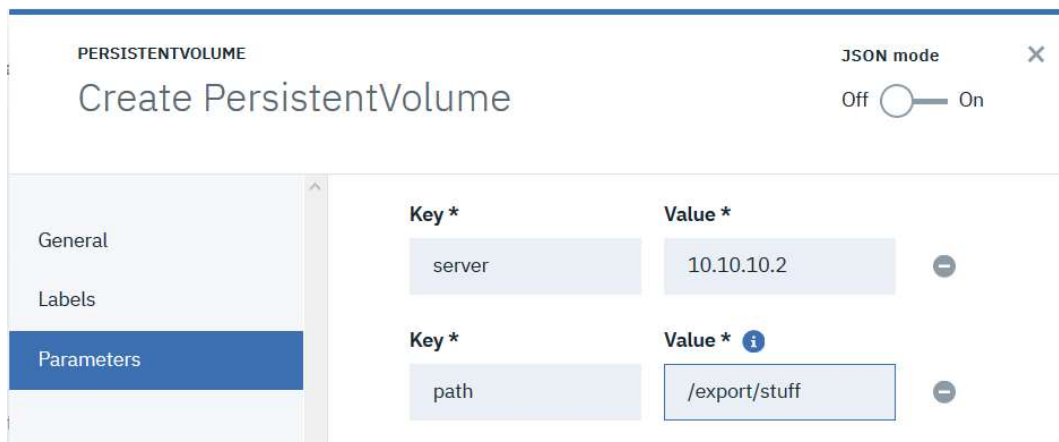
A Persistent Volume (PV) is needed to store the configuration and workload data. This should be in a well-known location as there is a need to back up the data and change the configuration. Create a PV within the GUI by navigating to *Platform -> Storage* and clicking on *Create PersistentVolume*. Set the name and size. Set the Labels as shown follows:



The screenshot shows the 'Create PersistentVolume' window with the 'Labels' tab selected. The window has a title bar 'PERSISTENTVOLUME' and a 'JSON mode' toggle (currently Off). The left sidebar has 'General', 'Labels', and 'Parameters' tabs. The main area shows a table with two columns: 'Label' and 'Value'. There is one row with 'lsfvol' in both fields. Below the table is an 'Add label +' button.

Label	Value
lsfvol	lsfvol

Set the Parameters for your storage. The parameters for an NFS server are shown below. The “server” is the IP address of the NFS server. The “path” should be set to the directory the NFS server is exporting, for example:



The screenshot shows the 'Create PersistentVolume' window with the 'Parameters' tab selected. The window has a title bar 'PERSISTENTVOLUME' and a 'JSON mode' toggle (currently Off). The left sidebar has 'General', 'Labels', and 'Parameters' tabs. The main area shows a table with two columns: 'Key *' and 'Value *'. There are two rows: one with 'server' and '10.10.10.2', and another with 'path' and '/export/stuff'.

Key *	Value *
server	10.10.10.2
path	/export/stuff

Once done click Create. The PersistentVolume list should have the volume, and it should be free.

Deploy the Cloud Pak

Deployment of the Cloud Pak has two steps:

1. Load the Cloud Pak into IBM Cloud Privates local catalog.
2. Deploy the Helm chart.

The “cloudctl” cli is used to import IBM Spectrum Computing Cloud Pak into IBM Cloud Private. Log in to begin the process with:

```
$ cloudctl login -a {URL of the ICP master e.g. https://10.10.10.70:8443} --skip-ssl-validation -u admin
```

It will prompt for a password. This is the same password that was set during installation and used to login to the GUI. It will then prompt for the namespace. The following namespaces have been tested:

- Default
- Kube-system

You can also create a new namespace. Next login to the local repository with:

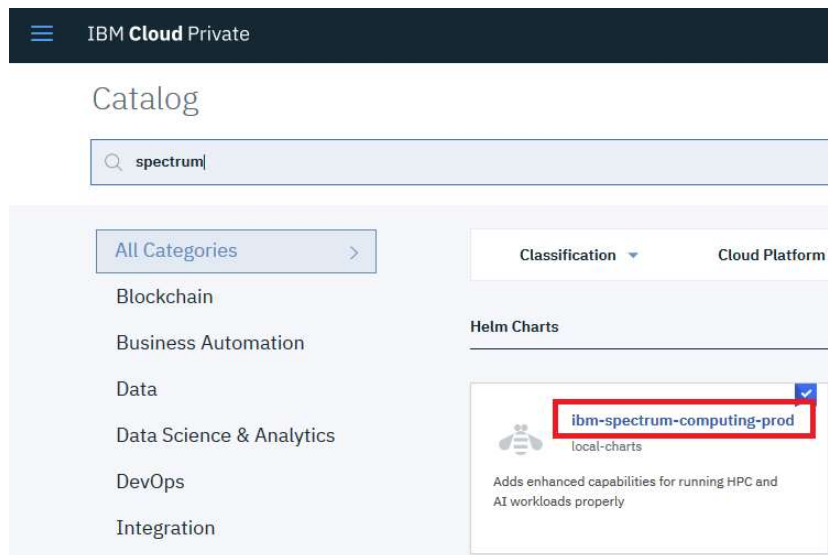
```
$ docker login mycluster.icp:8500
```

Once logged in, import the catalog by running the following command:

```
$ cloudctl catalog load-archive --archive ibm-spectrum-computing-prod.tar
```

After the command finishes it may take a few minutes before the synchronization finishes.

Log in to the IBM Cloud Private GUI to deploy the Cloud Pak. Click on the “Catalog” and in the search bar type “spectrum”. You should see the following:



Select “ibm-spectrum-computing-prod” to continue the deployment. Documentation and sample commands and configuration will be displayed. Click “Configure” to customize the deployment. Do the following:

1. Set the name of the Helm release.
2. Choose the Target namespace.
3. Read through and accept the licenses.

You will see something like the following:

The screenshot shows the IBM Cloud Private interface for configuring the 'ibm-spectrum-computing-prod V 1.0.0' resource. The 'Configuration' tab is active, displaying a form with the following fields:

- Helm release name ***: A text input field containing 'supersched'.
- Target namespace ***: A dropdown menu showing 'default'.
- License ***: A checkbox labeled 'I have read and agreed to the License agreement', which is checked.

Check the Storage Configuration in the Parameter Section. When using an NFS persistent volume make sure to set the GID of the persistent volume. If the NFS server is exporting “/export/stuff”, check the group for /export/stuff for example,

```
# ls -la /export/stuff
total 8
drwxr-xr-x 2 mblack lsfadmin 4096 Mar 25 14:55 .
```

Here the group is “lsfadmin”. Make sure the filesystem group read, write and execute bit is set by running:

```
# chmod 775 /export/stuff
```

Determine the GID of the filesystem group, in this case “lsfadmin”, by running:

```
# getent group |grep lsfadmin
lsfadmin:x:495:
```

Here, the GID of the “lsfadmin” group is 495. Use that value for the:

- The fsGroup for the PVC
- The supplementalGroups for the PVC

For example:

Storage Configuration

Provide the configuration values for the storage

The fsGroup for the PVC

495

The supplementalGroups for the PVC

495

PV Label

lsfvol

PV Label Value

lsfvol

NOTE: You NFS server will have a different path and group. Use your values.

Click on “Install”. The pods will be deployed on the worker machines and one of the manager machines.

View the helm release to see what is being deployed.

Verify the Cloud Pak Deployment

Use the following procedures to verify that the Cloud Pak is functioning correctly. Log in with the “cloudctl” cli, for example:

```
$ cloudctl login -a {URL of the ICP master e.g. https://10.10.10.10:8443} --skip-ssl-validation -u admin
```

List the helm deployments with the following command:

```
$ helm list --tls |grep spectrum-computing
supersched                      1                      Mon Mar 25 15:07:56 2019
DEPLOYED                        ibm-spectrum-computing-prod-1.0.0  default
```

This shows that the chart is deployed and that it is called: “supersched”. Now check the daemonset by running the following command:

```
$ kubectl get daemonsets --namespace {Namespace used to deploy chart}
NAME                                DESIRED  CURRENT  READY  UP-TO-DATE  AVAILABLE  NODE
SELECTOR                            AGE
ibm-scheduler-agent                4        4        3      4           3          node-
role.kubernetes.io/worker=true    23m
```

The desired number of pods is 4, but only 3 are running. List the pods to see which one is not working:

```
# kubectl get pods
NAME                                READY  STATUS             RESTARTS  AGE
ibm-scheduler-64c9b6d84f-7kzmq    1/1    Running            0         23m
ibm-scheduler-agent-4q68g         1/1    Running            0         23m
ibm-scheduler-agent-ffbhd         0/1    ContainerCreating  0         23m
ibm-scheduler-agent-qgwx5         1/1    Running            0         23m
```

```
ibm-scheduler-agent-vg8rg      1/1      Running      0      23m
```

Check the logs of the container that is not in Running state with the following command:

```
$ kubectl logs ibm-scheduler-agent-ffbhd
```

Or get more information about the pod with the following command:

```
$ kubectl describe pod ibm-scheduler-agent-ffbhd
```

Connect to the `ibm-scheduler` pod for the following tests with the following command:

```
$ kubectl exec -ti ibm-scheduler-64c9b6d84f-7kzmq bash
```

```
LSF POD [root@master /]#
```

NOTE: The prompt changed to indicate you are operating in a pod.

Try some LSF commands to see the cluster state for example:

```
LSF POD [root@master /]# lsid
IBM Spectrum LSF Standard Edition 10.1.0.0, Mar 25 2019
Copyright International Business Machines Corp. 1992, 2016.
US Government Users Restricted Rights - Use, duplication or disclosure
restricted by GSA ADP Schedule Contract with IBM Corp.
```

```
My cluster name is myCluster
```

```
My master name is master
```

This tells you that the manager processes are running. Next check the workers state by running the following command:

```
LSF POD [root@master /]# lshosts
HOST_NAME      type      model  cpuf  ncpus  maxmem  maxswp  server  RESOURCES
master         X86_64    Intel_EM  60.0   20  127.8G   3.9G    Yes (mg)
host88b1       X86_64    PC6000  116.1   12   31.2G   3.9G    Yes ()
host88b2       X86_64    PC6000  116.1   10   31.2G   3.9G    Yes ()
host88b3       X86_64    PC6000  116.1   10   31.2G   3.9G    Yes ()
```

and

```
# bhosts
HOST_NAME      STATUS      JL/U    MAX  NJOBS    RUN  SSUSP  USUSP  RSV
host88b1       unreach     -       12    0        0    0      0      0
host88b2       closed      -       10    0        0    0      0      0
host88b3       ok          -       10    0        0    0      0      0
master         ok          -       20    0        0    0      0      0
```

The machines STATUS should be ok. If the worker machine state is ok, they are ready to accept jobs. If the cluster just started or you reconfigured the cluster, it may take a minute for all machines to be ok.

Deploy Jobs

Once installed the Cloud Pak enables new options for jobs run through Kubernetes. These options are needed for running High Performance Computing (HPC) applications. They extend the pod specification with annotations to define which scheduling and placement policies to use. To enable the enhanced features the pod specification must have the schedulerName set to "lsf" for example:

```
spec.template.spec.schedulerName: lsf
```

The new features for kubernetes jobs that this provides includes the following:

- Job Priority
- Application Profiles
- Fair sharing of resources
- Parallel Jobs
- GPU Management

The following table shows the new annotations along with the LSF equivalent.

Pod Spec Field	Description	LSF Job Submission Option
*.metadata.name	A name to assign to the job	Job Name (-J)
++.lsf.ibm.com/minconcurrent	The number of pods to run as part of the parallel job. Set parallelism to the same value.	Number of machines (-n)
++.lsf.ibm.com/project	A project name to assign to job	Project Name (-P)
++.lsf.ibm.com/application	An application profile to use	Application Profile (-app)
++.lsf.ibm.com/gpu	The GPU requirements for the job	GPU requirement (-gpu)
++.lsf.ibm.com/queue	The name of the job queue to run the job in	Queue (-q)
++.lsf.ibm.com/jobGroup	A job group to assign to job	Job Group (-g)
++.lsf.ibm.com/fairshareGroup	The fairshare group to assign the job to	Fairshare Group (-G)
++.lsf.ibm.com/user	The user to run the application as, and for accounting	Job submission user
++.lsf.ibm.com/serviceClass	The service class to apply to the job	Service class (-sla)
++.lsf.ibm.com/reservation	The resources to reserve prior to running the job	Advanced Reservation (-U)
*.spec.containers[].resources.requests.memory	The amount of memory to reserve for the job	Memory Reservation (-R "rusage[mem=...]")
*.spec.schedulerName	Set to "lsf"	N/A

NOTE:

- * - The pod specification files should be prefaced with *spec.template*.
- ++ - The pod specification files should be prefaced with *spec.template.metadata.annotations*.

For information on the annotations and their meanings refer to the following:

https://www.ibm.com/support/knowledgecenter/SSWRJV_10.1.0/lsf_welcome/lsf_kc_cluster_ops.html

These capabilities are accessed by modifying the pod specifications for jobs. Below is a minimal example:

```
apiVersion: batch/v1
kind: Job
metadata:
  name: myjob-001
spec:
  template:
    metadata:
      name: myjob-001
    spec:
      schedulerName: lsf          # This directs scheduling to the LSF Scheduler
      containers:
      - name: ubuntuutest
        image: ubuntu
        command: ["sleep", "60"]
        resources:
          requests:
            memory: 5Gi
        restartPolicy: Never
```

This example enables Kubernetes to use **lsf** as the job scheduler. The LSF job scheduler can then apply its policies to choose when and where the job will run.

Additional parameters can be added to the pod yaml file to control the job. The example below shows how to use the additional annotations:

```
apiVersion: batch/v1
kind: Job
metadata:
  name: myjob-001
spec:
  template:
    metadata:
      name: myjob-001
      # The following annotations provide additional scheduling
      # information to better place the pods on the worker nodes
      # NOTE: Some annotations require additional LSF configuration
      annotations:
        lsf.ibm.com/project: "big-project-1000"
        lsf.ibm.com/queue: "normal"
        lsf.ibm.com/jobGroup: "/my-group"
        lsf.ibm.com/fairshareGroup: "bestshare"
    spec:
      # This directs scheduling to the LSF Scheduler
      schedulerName: lsf
      containers:
      - name: ubuntuutest
        image: ubuntu
        command: ["sleep", "60"]
        restartPolicy: Never
```

In the previous example, the annotations provide the LSF scheduler more information about the job and how it should be run.

Users that submit a job through Kubernetes typically are trusted to run services and workloads as other users. For example, the pod specifications allow the pod to run as other users:

```
apiVersion: batch/v1
kind: Job
metadata:
  name: myjob-uid1003-0002
spec:
  template:
    metadata:
      name: myjob-uid1003-0002
    spec:
      schedulerName: lsf
      containers:
      - name: ubuntutest
        image: ubuntu
        command: ["id"]
        restartPolicy: Never
        securityContext:
          runAsUser: 1003
          fsGroup: 100
          runAsGroup: 1001
```

In the previous example, the pod would run as UID 1003 and produce the following output:

```
uid=1003(billy) gid=0(root) groups=0(root),1001(users)
```

Note the GID and groups. Care should be taken to limit who can create pods. Alternatively LSF applications can be used to allow the administrator to predefine pod specification file.

Going Further

Additional examples and help are available from:

<https://github.com/IBMSpectrumComputing/lsf-kubernetes>

Bugs may be logged to the Github site. Do not post any confidential information. Newer versions of this document may also be found there.

Change the Scheduler Configuration

The scheduler configuration files are stored in the persistent volume. The configuration can be changed by editing these files either from within the pod or directly from the persistent volume.

The persistent volume claim has the following structure:

```
{PV mount point} / lsf / conf
                  / work
```

Within the running containers the conf and work directories are symbolically linked to the following file paths:

```
/opt/ibm/lsfsuite/lsf/conf
/opt/ibm/lsfsuite/lsf/work
```

The conf directory has the following important files:

```
conf/cshrc.lsf
conf/profile.lsf
conf/hosts
conf/lsf.conf
conf/lsf.cluster.myCluster
conf/lsf.shared
conf/lsf.task
conf/lsbatch/myCluster/configdir/lsb.users
conf/lsbatch/myCluster/configdir/lsb.nqsmaps
conf/lsbatch/myCluster/configdir/lsb.reasons
conf/lsbatch/myCluster/configdir/lsb.hosts
conf/lsbatch/myCluster/configdir/lsb.serviceclasses
conf/lsbatch/myCluster/configdir/lsb.resources
conf/lsbatch/myCluster/configdir/lsb.modules
conf/lsbatch/myCluster/configdir/lsb.threshold
conf/lsbatch/myCluster/configdir/lsb.applications
conf/lsbatch/myCluster/configdir/lsb.globalpolicies
conf/lsbatch/myCluster/configdir/lsb.params
conf/lsbatch/myCluster/configdir/lsb.queues
```

Details of the configuration files is in IBM Spectrum Computing documentation:

https://www.ibm.com/support/knowledgecenter/SSWRJV_10.1.0/lsf_welcome/lsf_kc_cluster_ops.html

Additional helper scripts are:

- `conf/add-users-groups.sh`
 - Helper scripts to import users and groups into the scheduler pods
- `conf/trigger-reconfig.sh`
 - Triggers reconfiguration of the cluster after any of the above files have been changed.

Log Files

The log files can provide useful information for troubleshooting problems. The log files to look at are in the Manager pod. To access the logs, use the following procedure:

1. Get the name of the Manager pod:

```
$ kubectl get pods --namespace {Namespace used to deploy chart}
```

Look for the pod named “ibm-scheduler-{Some UUID}”, and not the “ibm-scheduler-agent-*” pods

2. Connect to the manager pod:

```
$ kubectl exec -ti {Pod name from above} bash
```

3. Go to the log directory and look at logs:

```
LSF POD [root@master /]# cd /opt/ibm/lsfsuite/lsf/log
```

```
LSF POD [root@master /]# more batch-driver.log
```

An example of a configuration error might look like the following:

```
LSF POD [root@master log]# more batch-driver.log
Log file created at: 2019/03/27 14:38:40
Running on machine: malconv02
Binary: Built with gc go1.11.2 for linux/amd64
Log line format: [IWEF]mmdd hh:mm:ss.uuuuuu threadid file:line] msg
E0327 14:38:40.125268      376 jobhandler.go:308] Unable to submit the job
to lsf. cmd<bsub -q normal -g my-group -P big-project-1000 -G bestshare
-J default/myjob-001-6hljx -ext kube[default/myjob-001-6hljx] sleep
1000000> error<Bad or empty job group name. Job not submitted.>
```

To correct this problem, and ones like it, modify the job pod specification and make sure LSF is configured with the right fair share groups, job groups, users etc.

Adding Users and Groups

Some of the workload policies need to be aware of the user running the job. For these policies to function, it is necessary to provide the usernames, UIDs, and GIDs typically found in `/etc/passwd` and `/etc/group` files. The **add-users-groups.sh** sample script is provided to import the information from the host machine’s operating system. To import the users and groups, go a Linux machine that has both users and groups configured (such as LDAP/NIS), and has access to the persistent volume, then run the following commands:

```
$ cd {Location of PV mount}
$ cd lsf/conf
$ add-users-groups.sh
```

This script will call **getent** to gather all the users and group information and generate two files:

- `passwd.append`
- `group.append`

The containers, upon detecting these files will import the passwd and group information. Once this is done the fairshare policies for users can be configured.

Reconfiguring the Cluster

When the configuration files are changed it is necessary to reconfigure the cluster. This can be done within the cluster by running the appropriate commands, or alternatively by using the **trigger-reconfig.sh** helper script. Inside the container the script is located here:

```
/opt/ibm/lfsuite/lsf/conf/trigger-reconfig.sh
```

Within the persistent volume it is located here:

```
{PV mount point} / lsf / conf / trigger-reconfig.sh
```

Multi-NIC Hosts

Hosts with multiple configured NICs can cause problems for the deployment. They may inadvertently create the wrong configuration for the cluster. A host may prefer to send the traffic over a NIC that other hosts may not be able to route to or may be NAT'ed. The **conf/hosts** file provides a way to control which NIC the scheduler will prefer. The order in the file controls the order it will try to use. An incorrect entry might look like the following:

```
192.168.1.10      master  master.br1
10.10.10.10       master  master.eno1
10.10.10.20       worker001 worker001.eth0
10.10.10.21       worker002 worker002.eth0
```

The correct configuration should be like the following:

```
10.10.10.10       master  master.eno1
192.168.1.10      master  master.br1
10.10.10.20       worker001 worker001.eth0
10.10.10.21       worker002 worker002.eth0
```

With this configuration all scheduler traffic will stay on the 10.0.0.x network. The actual IPs used will be detected automatically.

Copyright and trademark information

© Copyright IBM Corporation 2019

U.S. Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

IBM®, the IBM logo and ibm.com® are trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.