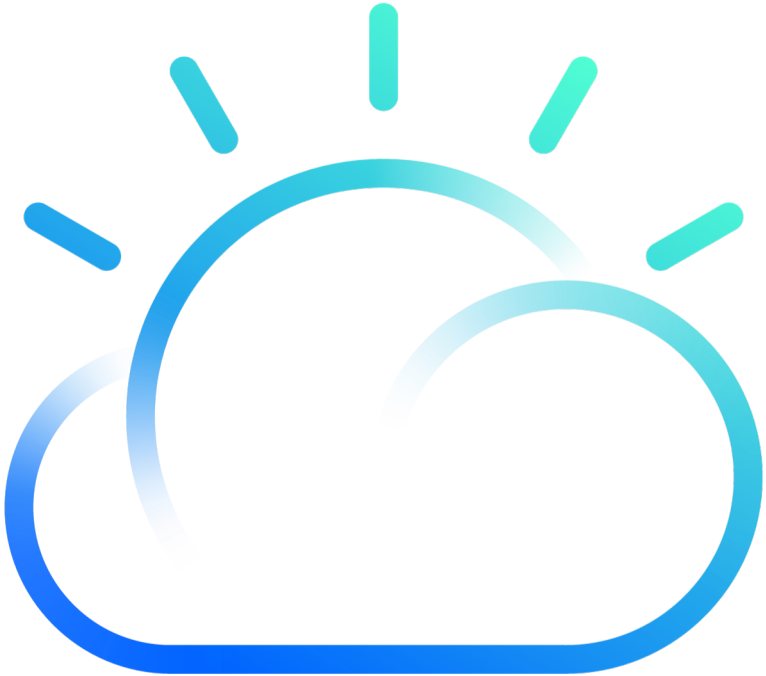
Tips for Troubleshooting IBM Cloud Private



Agenda

Tips for troubleshooting and debugging IBM Cloud Private

Tips for Troubleshooting and Debugging IBM Cloud Private

- Getting a user credential token for the kubectl CLI
- Change the default namespace when using Configure client
- 502 gateway messages when accessing console
- Accessing ICP when the UI doesn’t work
- Default LDAP Group Filter for Active Directory

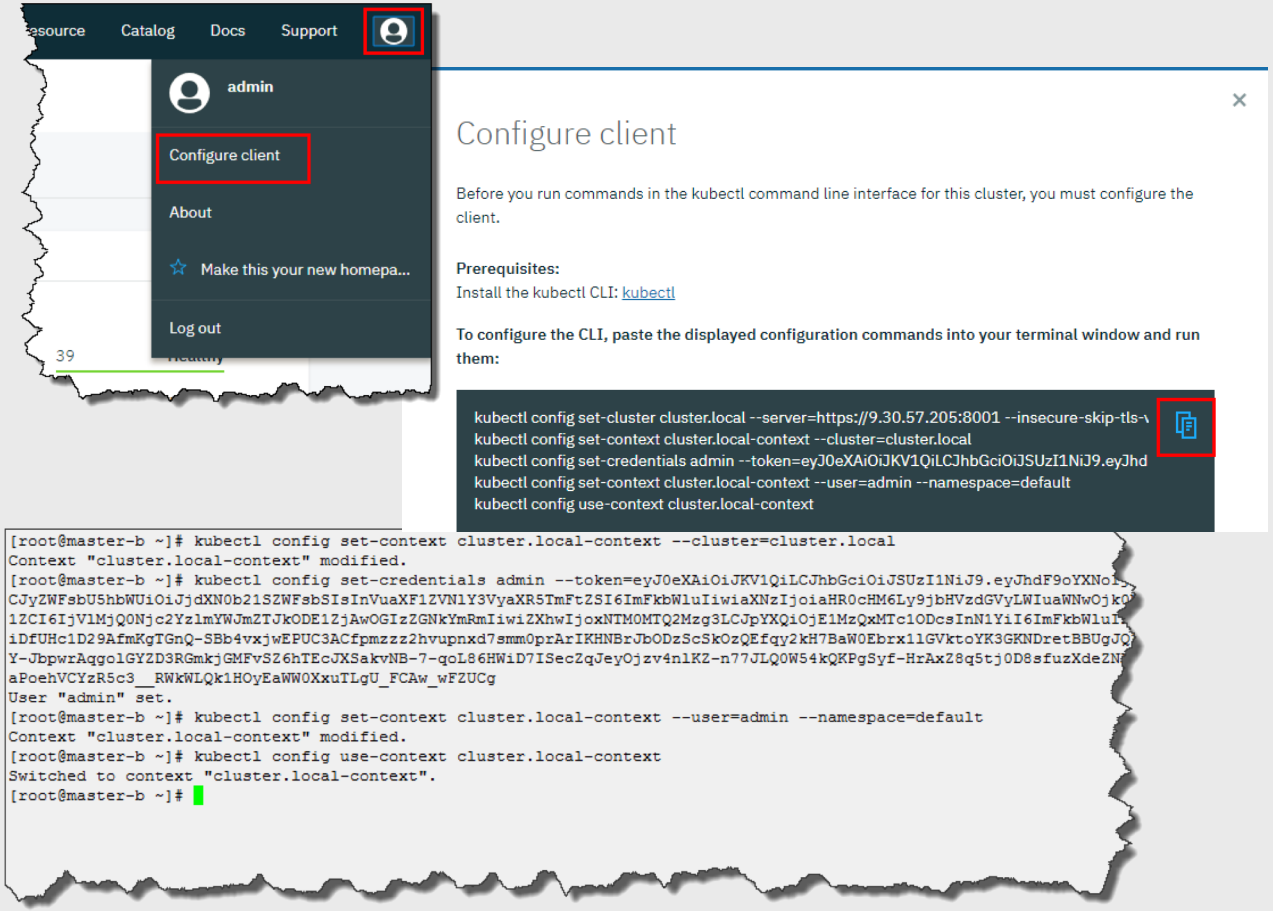
Tips for Troubleshooting IBM Cloud Private

- Do I have to use an IP address to access the console
- Installation problems
- When the Grafana default datasource isn’t configured
- ibmcloud pr – how to update
- Vulnerability Advisor pods fail after running for a while
- ICP Center of Competency (CoC) provided tools

Getting a user credential token for the kubectl CLI

An easy way to get a token to authenticate the kubectl CLI is to use the ICP Console

1. Logon to the console
2. Click the user icon on the console
3. Select “Configure Client”
4. From the pop-up click the copy icon
5. Paste the copied text into your terminal window



Tip: Change the default namespace when using Configure client

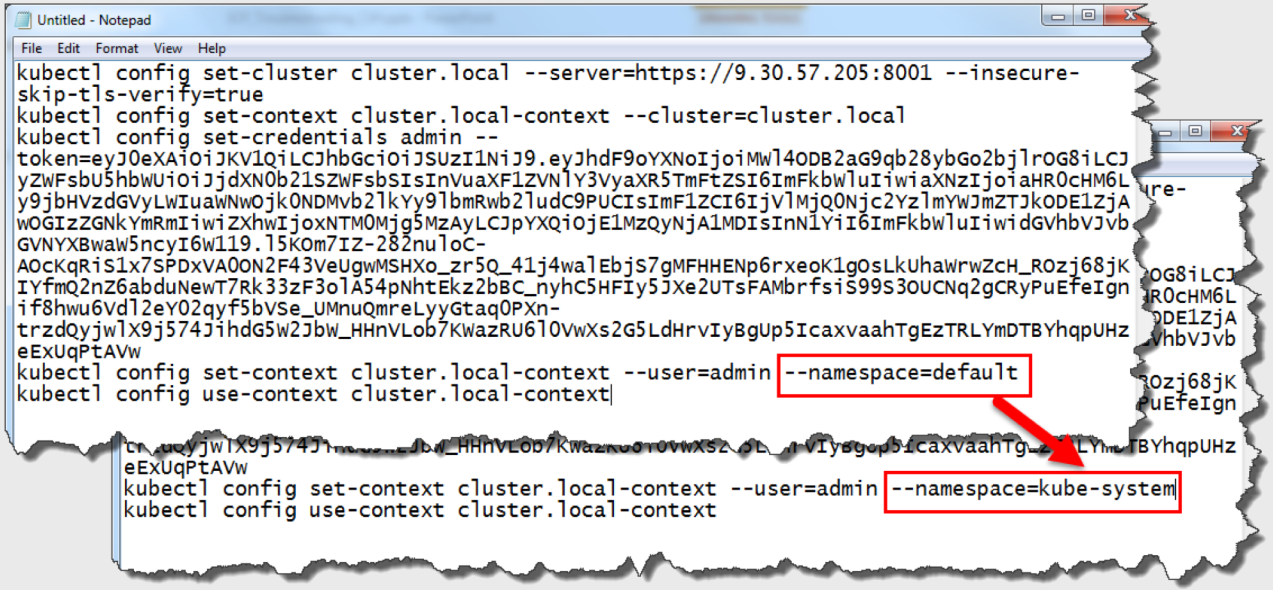
It can be inconvenient to constantly have to type:

-n kube-system

on every kubectl command.

To change default namespace:

1. Clip the config from the console
2. Paste the config into a text editor
3. Change namespace from “default” to “kube-system”
4. Clip the config from the text editor and paste into the master terminal window.



502 gateway messages

Check the “auth” pods to see if any look like they have issues

```
[root@master-b ~]# kubectl -n kube-system get pods | grep auth
auth-apikeys-zfpjn                                        1/1       Running   2          2d
auth-idp-56qrb                                            3/3       Running   6          2d
auth-pap-s4cxw                                            1/1       Running   2          2d
auth-pdp-tsnm2                                            1/1       Running   2          2d
```

If nothing looks amiss a practical next step is to restart all “auth” pods

```
[root@master-b ~]# kubectl -n kube-system get pods | grep auth | awk '{print $1}' | xargs kubectl -n kube-system delete pods
pod "auth-apikeys-zfpjn" deleted
pod "auth-idp-56qrb" deleted
pod "auth-pap-s4cxw" deleted
pod "auth-pdp-tsnm2" deleted
```

Wait until all pods have restarted

```
[root@master-b ~]# kubectl -n kube-system get pods | grep auth
auth-apikeys-pz2ff                                        0/1       ContainerCreating   0          1s
auth-idp-56qrb                                            0/3       Terminating         5          2d
auth-pap-s4cxw                                            1/1       Terminating         2          2d
auth-pdp-tsnm2                                            1/1       Terminating         2          2d
[root@master-b ~]# kubectl -n kube-system get pods | grep auth
auth-apikeys-pz2ff                                        1/1       Running   0          5m
auth-idp-959gz                                            3/3       Running   0          4m
auth-pap-8w6fq                                            1/1       Running   0          5m
auth-pdp-644dt                                            1/1       Running   0          3m
```

Accessing ICP when the UI doesn’t work

Option 1:

Logon to ICP using the using the ibmcloud CLI with the ICP pr plugin

From any node or workstation:

ibmcloud pr login –a https://<master_ip>:8443 --skip-ssl-validation

Note: The node/workstation must have the kubectl CLI and the ibmcloud CLI with the pr plug-in installed.

```
[root@master-b ~]# ibmcloud pr login -a https://cluster-b.icp:8443 --skip-ssl-validation
API endpoint: https://cluster-b.icp:8443

Username> admin

Password>
Authenticating...
OK

Select an account:
1. cluster-b Account (id-cluster-b-account)
Enter a number> 1
Targeted account cluster-b Account (id-cluster-b-account)

Configuring helm and kubectl...
Configuring kubectl: /root/.bluemix/plugins/icp/clusters/cluster-b/kube-config
Property "clusters.cluster-b" unset.
Property "users.cluster-b-user" unset.
Property "contexts.cluster-b-context" unset.
Cluster "cluster-b" set.
User "cluster-b-user" set.
Context "cluster-b-context" created.
Switched to context "cluster-b-context".

Cluster cluster-b configured successfully.

Configuring helm: /root/.helm
Helm configured successfully

OK

[root@master-b ~]# kubectl -n kube-system get pods
NAME                                                              READY     STATUS    RESTA
auth-apikeys-zfpjn                                                1/1       Running   2
auth-idp-56qrb                                                    3/3       Running   6
auth-pap-s4cxw                                                    1/1       Running   2
auth-pdp-tsnm2                                                    1/1       Running   2
```

Accessing ICP when the UI doesn’t work

Option 2: Use kubectl and authenticate using a certificate instead of a token

Authenticate using kubelet’s certificate

```
kubectl --kubeconfig=/var/lib/kubelet/kubelet-config -n kube-system <any kubectl command>
```

Authenticate using kubectl’s certificate

```
kubectl --kubeconfig=/var/lib/kubelet/kubectl-config -n kube-system <any kubectl command>
```

Note: This option only works from a master node with kubectl installed

Accessing ICP when the UI doesn't work

Option 3 – When all else fails: Enable and use the insecure port on the ICP apiserver

Step 1: Enable the insecure port

1. SSH onto a master node
2. Stop the kubelet on the master node:
   > systemctl stop kubelet
3. Edit the file /etc/cfc/pods/master.json
4. In the apiserver section update the line:
   from: "--insecure-port=0",
   to: "--insecure-port=8888",
5. Insert the following line:
   "--insecure-bind-address=127.0.0.1",
6. Save the file
7. Restart kubelet on the master node:
   > systemctl start kubelet (**Note:** Be patient, it may take a couple of minutes to take effect)

Accessing ICP when the UI doesn't work

Option 3 (Con't)

Step 2: Use the apiserver insecure port to execute Kubernetes commands

For example:

kubectl –s 127.0.0.1:8888 –n kube-system get pods

Note: Security for the "insecure" port exists by virtue of:

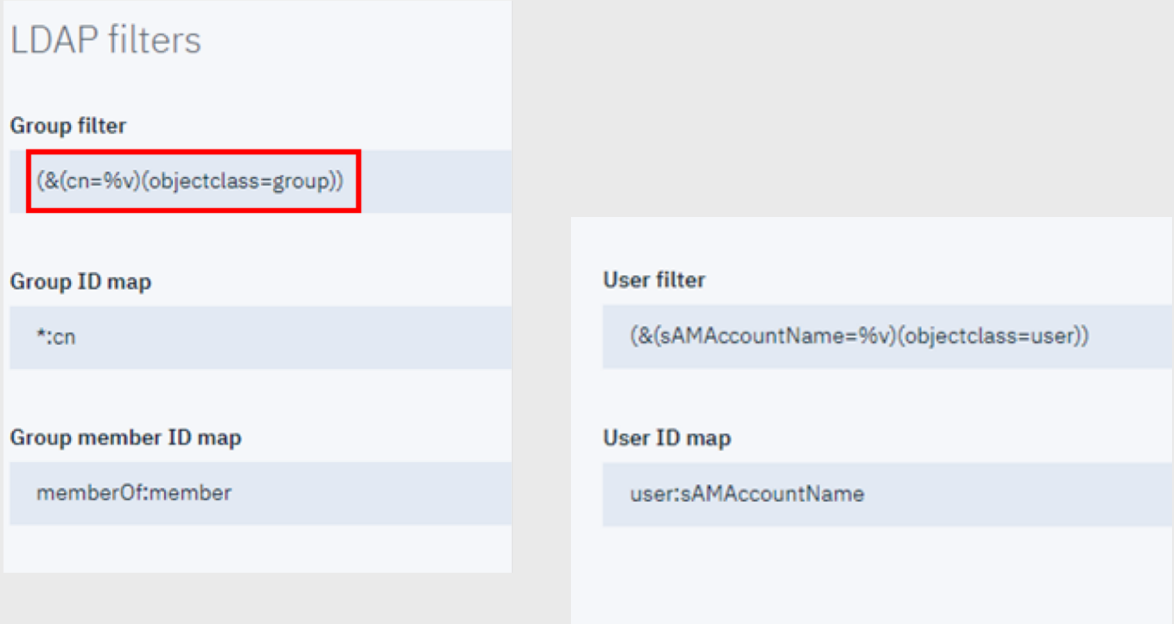
1. Commands can only be executed on the master node using the localhost address
2. You have to have logon credentials to logon to the master node

```
[root@master-b ~]# kubectl -s 127.0.0.1:8888 -n kube-system get pods
NAME                                                              READY
auth-apikeys-pz2ff                                                1/1
auth-idp-959gz                                                    3/3
auth-pap-8w6fq                                                    1/1
auth-pdp-644dt                                                    1/1
calico-kube-controllers-759f7fc556-prk7k                          1/1
calico-node-9rw8x                                                 2/2
calico-node-k7jcf                                                 2/2
calico-node-lgrq5                                                 2/2
calico-node-m9fwk                                                 2/2
calico-node-vh4c6                                                 2/2
catalog-ui-f2krk                                                  1/1
default-backend-7c6d6df9d5-4hdh4                                  1/1
heapster-5649f84695-phmkt                                         2/2
helm-api-978d578f-79dlj                                           2/2
helm-repo-7fd844d7bf-8lvtw                                        1/1
icp-management-ingress-npwmc                                      1/1
icp-mongodb-0                                                     1/1
image-manager-0                                                   2/2
k8s-etcd-172.16.159.245                                           1/1
k8s-master-172.16.159.245                                         3/3
k8s-proxy-172.16.159.245                                          1/1
k8s-proxy-172.16.243.166                                          1/1
k8s-proxy-172.16.243.169                                          1/1
k8s-proxy-172.16.243.204                                          1/1
k8s-proxy-172.16.243.207                                          1/1
kube-dns-2pfvq                                                    3/3
```

Default LDAP Group Filter for Active Directory

When configuring LDAP with MS Active Directory the default Group Filter is incorrect and must be changed to:

(&(cn=%v)(objectclass=group))



Do I have to use an IP address to access the console

Customize the Uniform Resource Locator (URL) that you use to log in to the IBM® Cloud Private cluster management console.

Example of default accepted URI prefixes in ConfigMap: registration-json

```
"trusted_uri_prefixes":[
   "https://172.16.159.245:8443",
   "https://9.30.57.205:8443",
   "https://cluster-b.icp:8443"
   ],
"redirect_uris":[
   "https://172.16.159.245:8443/auth/
```

Supported customization formats

The following customization formats are supported:

- https://<Public IP>:8443/console
- https://<Public IP>:8443/console/
- https://<Private IP>:8443/console/
- https://<Private IP>:custom-port/console/
- https://<host name>:8443/console
- https://<host name>:custom-port/console
- https://localhost:8443/console
- https://localhost:<custom port>/console
- https://<Regex host name>:8443/console
- https://<Regex IP>:8443/console
- https://<Regex host name>:<custom port>/console
- https://<Regex IP>:<custom port>/console
- https://<Regex host name>:<Regex port>/console
- https://<Regex IP>:<Regex Port>/console

Required user type or access level: Cluster administrat

https://www.ibm.com/support/knowledgecenter/SSBS6K_2.1.0.3/user_management/custom_url.html

Installation problems

VALIDATE USING CHECK PARAMETER

```
docker run -e LICENSE=accept --net=host \
-v "$(pwd)":/installer/cluster ibmcom/icp-inception:2.1.0.3-ee ***check***
```

STANDARD INSTALL

```
docker run -e LICENSE=accept --net=host \
-v "$(pwd)":/installer/cluster ibmcom/icp-inception:2.1.0.3-ee ***install***
```

Installation problems

VERBOSE INSTALL

For additional verbose information add the ‘-vvv’ parameter per Knowledge Center link
https://www.ibm.com/support/knowledgecenter/en/SSBS6K_2.1.0.3/troubleshoot/install_hang.html

docker run -e LICENSE=accept --net=host \

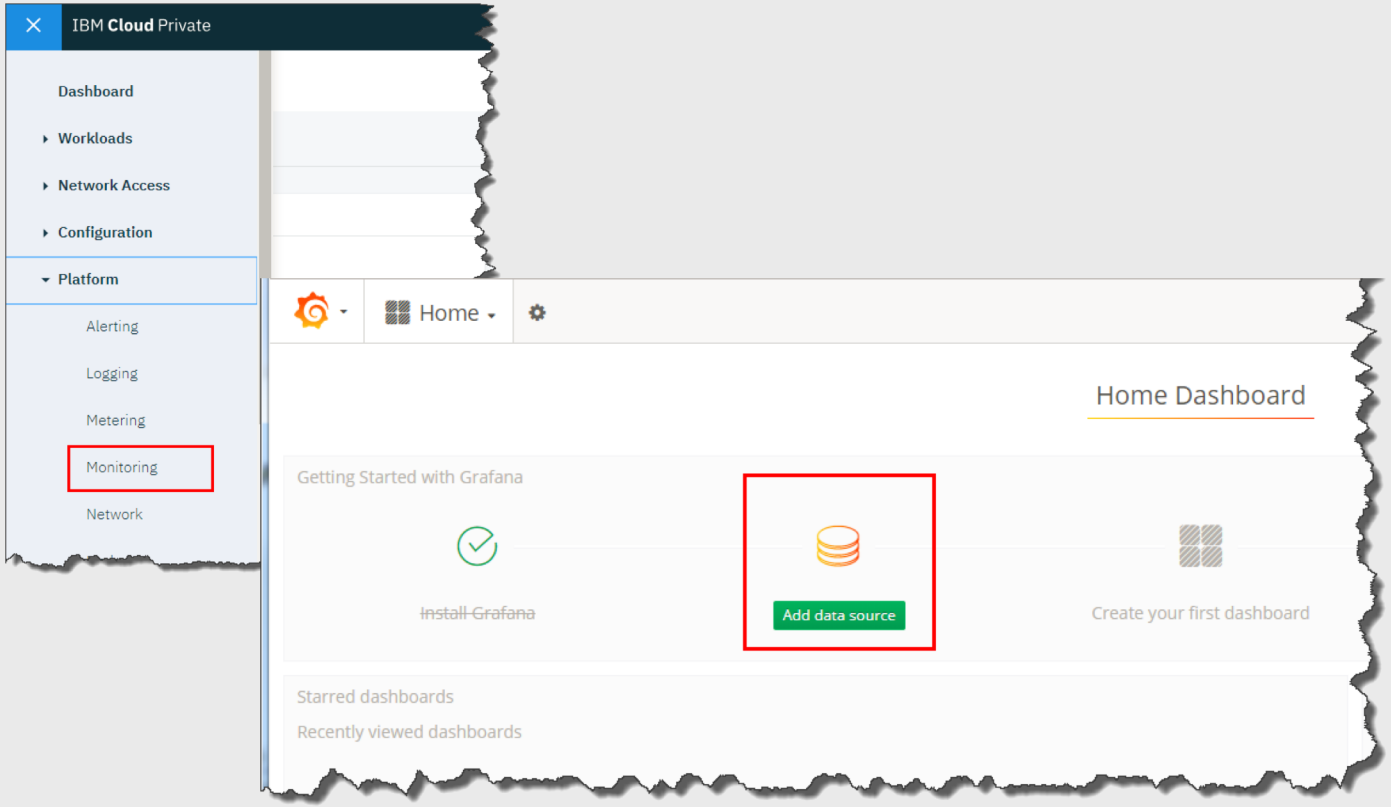
-v "$(pwd)":/installer/cluster ibmcom/icp-inception:2.1.0.3-ee ***install*** **–vvv**

Hint: When installing in verbose mode it is hard to know where you are in the installation because information is displayed too quickly on the screen. Open a second terminal session, tail the install log, and grep for “TASK”. This will give you a concise summary of which step you are in during the installation.

When the Grafana default datasource isn’t configured

Sometimes after ICP is installed Grafana starts but the default datasource did not get configured.

You need to configure the default datasource.



When the Grafana default data source isn’t configured

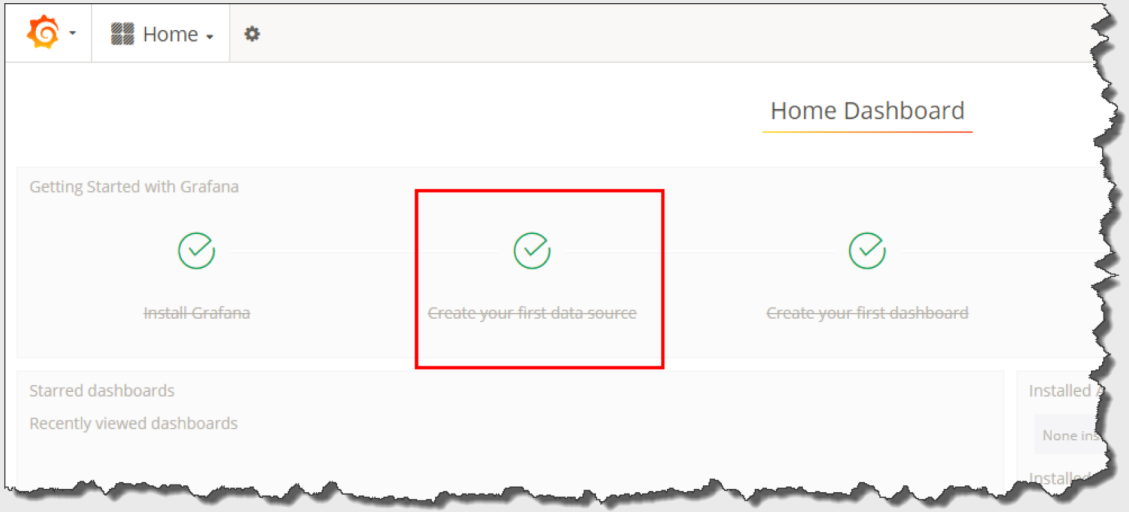
Manually configure a Prometheus data source:

1. On the master/boot node create a yaml file with the yaml from the knowledgecenter
2. Delete the old batch job
3. Execute a new batch job with the yaml file just created

Grafana should now be configured with a Prometheus datasource.

See: Manually configure a Prometheus data source in Grafana
https://www.ibm.com/support/knowledgecenter/en/SSBS6K_2.1.0.3/featured_applications/deploy_monitoring.html

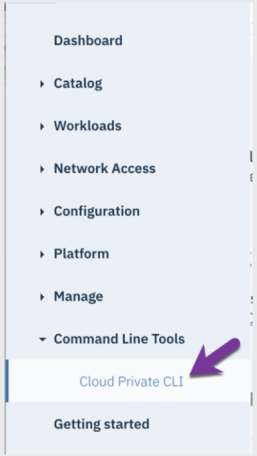
```
[root@master-b ~]# kubectl delete jobs/monitoring-grafana-ds -n kube-system
job.batch "monitoring-grafana-ds" deleted
[root@master-b ~]# kubectl apply -f grafana-datasource.yaml
job.batch "monitoring-grafana-ds" created
[root@master-b ~]#
```

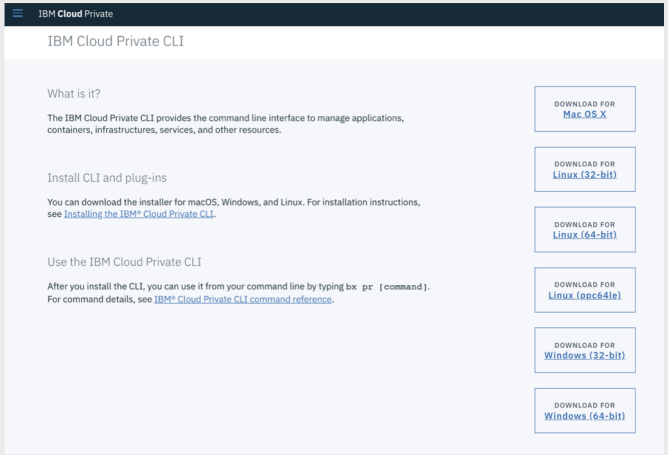


ibmcloud pr – how to update

Use of the ***ibmcloud plugin update*** does not currently update the IBM Cloud Private plugin.

A newer version must be downloaded from the link provided in the ICP Command Line Tools menu option. Once downloaded the new version must be install using the ***ibmcloud plugin install <file>*** command.





Vulnerability Advisor pods fail after running for a while

After installation the Vulnerability Advisor runs successfully for a while but slowly some pods start to fail with CrashLoopBackOff

Using the kubectl patch commands in the referenced document:

1. Patch the Elasticsearch StatefulSet
2. Patch the Kafka StatefulSet
3. Delete all failing VA pods

https://www.ibm.com/support/knowledgecenter/en/SSBS6K_2.1.0.3/troubleshoot/va_pod_fail.html

Patch the Elasticsearch StatefulSet:

kubectl patch statefulset/vulnerability-advisor-elasticsearch-data -n kube-system -p '{"spec":{"template":{"spec":{"containers":[{"env":[{"name":"ES_JAVA_OPTS","value":"-Xms1024m -Xmx1024m"}],"name":"es-data"}]}}}}‘

Patch the Kafka StatefulSet:

kubectl patch statefulset/vulnerability-advisor-kafka -n kube-system -p '{"spec":{"template":{"spec":{"containers":[{"command":["bash","-c","KAFKA_BROKER_ID=$((${HOSTNAME##*-}+1001)) KAFKA_RESERVED_BROKER_MAX_ID=$((${HOSTNAME##*-}+1001)) /usr/bin/start-kafka.sh"],"env":[{"name":"KAFKA_DELETE_TOPIC_ENABLE","value":"true"}],"name":"kafka-server"}]}}}}‘

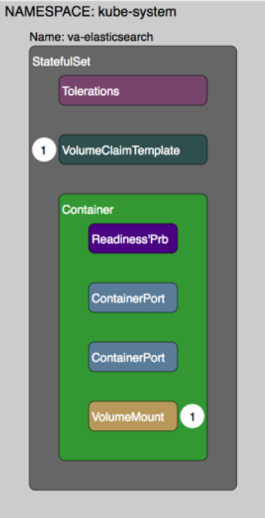
Find all failing Vulnerability Advisor pods and delete them:

> kubectl –n kube-system get pods | grep vulnerability-advisor | grep –v Running

IBM Cloud Private Center of Competency (CoC) provided tools

Kubetoy

An easy to use and easy to deploy app to help test and understand Kubernetes
https://github.com/IBM-ICP-CoC/KubeToy



Visual parsed Kubernetes (VpK) definitions
https://github.com/IBM-ICP-CoC/VpK

Questions?

Thank you