

Tips for Troubleshooting IBM Cloud Private

IBM



IBM Cloud

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

The screenshot shows the ICP Console interface. A user profile icon in the top right corner is highlighted with a red box. A dropdown menu appears, with the "Configure client" option also highlighted with a red box. Below the menu, there are links for "About", "Make this your new homepage...", and "Log out".

A modal window titled "Configure client" is displayed. It contains instructions: "Before you run commands in the kubectl command line interface for this cluster, you must configure the client." It lists prerequisites: "Install the kubectl CLI: [kubectl](#)". It provides configuration commands to be copied:

```
kubectl config set-cluster cluster.local --server=https://9.30.57.205:8001 --insecure-skip-tls-verify
kubectl config set-context cluster.local-context --cluster=cluster.local
kubectl config set-credentials admin --token=eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJhdF9oYXN0IjoiCjYzWFsbU5hbWUiOijdxKN0b21SZWFsbSIsInVuAXF1ZVN1Y3VyaXR5TmFtZSI6ImFkbWluIiwiXNzIjoiaHR0cHM6Ly9jbHVzdGvYLWIuaWNwOjk0IzCI6IjyVIMjQ0Njc2YzlmYWJmZTjkODE1ZjAwOGIzZGNkYmRmIiw1XhwIjoxNTM0MTQ2Mzg3LCJpXXQiOjE1MzQxMTc1ODcsInNiYiI6ImFkbWluIiDFUhc1D29afmKgTGNq-SBb4vxjwEPUC3ACfpzzz2hvupnxd7smm0prArIKHNBr0bODzScSk0zQEfgykH7baWEbxr11GVkt0YK3GKDretBBUgJOY-JbpwxAqgo1GY2D3RGmkjGMFvSz6hTEcxJSakvNB-7-qoL86HWiD7ISecZqJeyOjzv4n1KZ-n77JLQ0W54kQKPGSyf-HrAxZ8q5tj0D8sfuzXdeZNaaPoehVCYzR5c3__RwKLQk1HOyEaWW0XxuTLgU_FCAw_wFZUcg
User: "admin" set.
```

The terminal window below shows the commands being run:

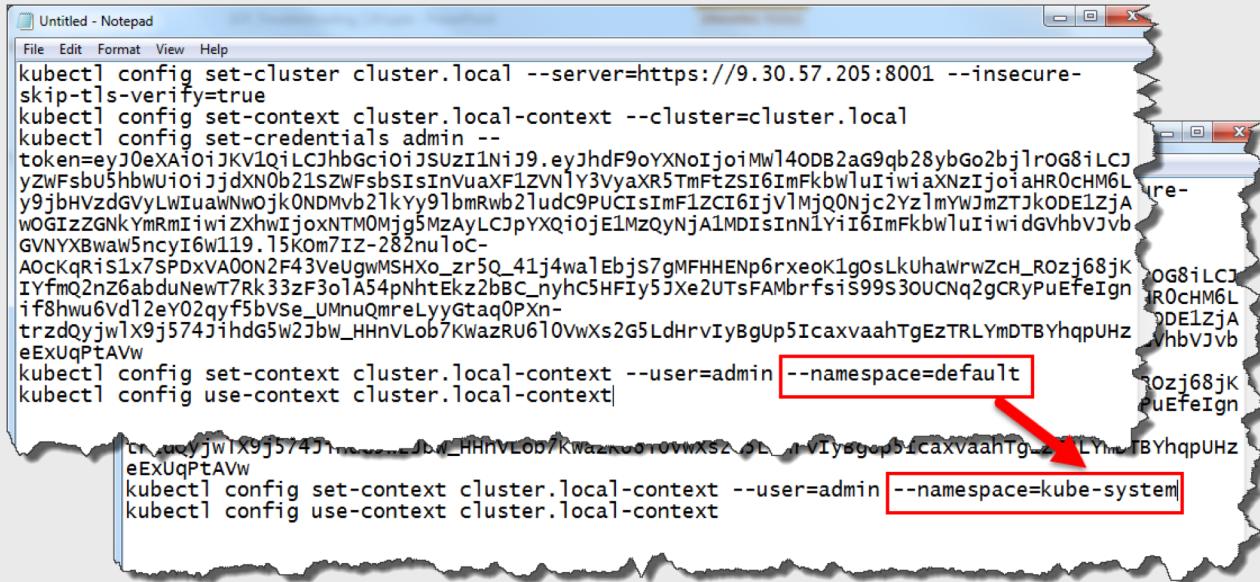
```
[root@master-b ~]# kubectl config set-context cluster.local-context --cluster=cluster.local
Context "cluster.local-context" modified.
[root@master-b ~]# kubectl config set-credentials admin --token=eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJhdF9oYXN0IjoiCjYzWFsbU5hbWUiOijdxKN0b21SZWFsbSIsInVuAXF1ZVN1Y3VyaXR5TmFtZSI6ImFkbWluIiwiXNzIjoiaHR0cHM6Ly9jbHVzdGvYLWIuaWNwOjk0IzCI6IjyVIMjQ0Njc2YzlmYWJmZTjkODE1ZjAwOGIzZGNkYmRmIiw1XhwIjoxNTM0MTQ2Mzg3LCJpXXQiOjE1MzQxMTc1ODcsInNiYiI6ImFkbWluIiDFUhc1D29afmKgTGNq-SBb4vxjwEPUC3ACfpzzz2hvupnxd7smm0prArIKHNBr0bODzScSk0zQEfgykH7baWEbxr11GVkt0YK3GKDretBBUgJOY-JbpwxAqgo1GY2D3RGmkjGMFvSz6hTEcxJSakvNB-7-qoL86HWiD7ISecZqJeyOjzv4n1KZ-n77JLQ0W54kQKPGSyf-HrAxZ8q5tj0D8sfuzXdeZNaaPoehVCYzR5c3__RwKLQk1HOyEaWW0XxuTLgU_FCAw_wFZUcg
User: "admin" set.
[root@master-b ~]# kubectl config set-context cluster.local-context --user=admin --namespace=default
Context "cluster.local-context" modified.
[root@master-b ~]# kubectl config use-context cluster.local-context
Switched to context "cluster.local-context".
[root@master-b ~]#
```

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.



```
Untitled - Notepad
File Edit Format View Help
kubectl config set-cluster cluster.local --server=https://9.30.57.205:8001 --insecure-skip-tls-verify=true
kubectl config set-context cluster.local-context --cluster=cluster.local
kubectl config set-credentials admin --
token=eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJhdF9oYXNoIjoimWl40DB2ag9qb28ybGo2bj1rOG8iLCJyZWFBu5hbWuioijdxN0b21S2WFsbSisInVuaXF1ZVN1Y3VyaXR5TmFtZSI6ImFkbwluIiwiiaXNZIjoiaHR0chM6Ly9jbHVzdGVyLWIuawNwojk0NDMvb2lkYy9lbnRw2ludc9PUCIsImFlZCI6IjV1MjQ0Njc2YzlmYWVmZTjkODE1ZjAwOGIZZGNkYmRmIwiZXhwIjoxNTM0Mjg5MzAyLCJpYXQiOjE1MzQyNjA1MDIsInhNIYiI6ImFkbwluIiwidgvhvJvbGVNYXBwaw5nci6W119.15K0m7IZ-282nuloC-A0cKqrIsIx7SPDxVA00N2F43VeUgwMSHXo_zr5Q_41j4walEbjs7gMFHHENp6rxeoK1gOsLkuaharwZCh_R0zj68jkIYfmQ2nZ6abduNewT7Rk33zf3o1A54pNhtEkz2bBC_nyhc5HFiy5Jxe2UTsFAMbrfsiS99s30UCNq2gCrYPuEfeIgndf8hwu6vd12eY02qyf5bvse_UMnuQmreLyyGtaq0Pxnn-trzdQyw1x9j574jihdg5w2jbw_HHnvLob7KwazRU610VwxS2G5LdHrvIyBgUp5IcaxvaahTgEzTRLymDTByhqpUHzeExUqPtAVw
kubectl config set-context cluster.local-context --user=admin --namespace=default
kubectl config use-context cluster.local-context|
```

The line `--namespace=default` is highlighted with a red box and a red arrow points to it from the text above. The line `--namespace=kube-system` is also highlighted with a red box.

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    RESTARTS  
auth-apikeys-zfpjn                   1/1     Running   2  
auth-idp-56qrb                        3/3     Running   6  
auth-pap-s4cxw                         1/1     Running   2  
auth-pdp-tsnnm2                       1/1     Running   2  
calico-kubelet-peer-7-qf7f0556-pv      1/1     Running   1
```

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)

```
{  
  "name": "apiserver",  
  "image": "ibmcom/hyperkube:v1.10.0-ee",  
  "imagePullPolicy": "IfNotPresent",  
  "command": [  
    "/hyperkube",  
    "apiserver",  
    "--secure-port=8001",  
    "--bind-address=0.0.0.0",  
    "--advertise-address=172.16.159.245",  
    "--endpoint-reconciler-type=lease",  
    "--insecure-port=8888",  
    "--insecure-bind-address=127.0.0.1",  
    "--etcd-servers=https://172.16.159.245:4001",  
    "--etcd-cafile=/etc/cfc/conf/etcd/ca.pem",  
    "--etcd-cert-file=/etc/cfc/conf/etcd/client.pem",  
    "--etcd-key-file=/etc/cfc/conf/etcd/client.key"  
  ]  
}
```

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   STATUS    RESTARTS   AGE
auth-apikeys-pz2ff                           1/1     Running   0          1d
auth-idp-959gz                             3/3     Running   0          1d
auth-pap-8w6fq                            1/1     Running   0          1d
auth-pdp-644dt                           1/1     Running   0          1d
calico-kube-controllers-759f7fc556-prk7k   1/1     Running   0          1d
calico-node-9rw8x                          2/2     Running   0          1d
calico-node-k7jcf                           2/2     Running   0          1d
calico-node-lgrq5                           2/2     Running   0          1d
calico-node-m9fwk                           2/2     Running   0          1d
calico-node-vh4c6                           2/2     Running   0          1d
catalog-ui-f2krk                           1/1     Running   0          1d
default-backend-7c6d6df9d5-4hdh4           1/1     Running   0          1d
heapster-5649f84695-phmkt                  2/2     Running   0          1d
helm-api-978d578f-79dlj                   2/2     Running   0          1d
helm-repo-7fd844d7bf-81vtw                 1/1     Running   0          1d
icp-management-ingress-npwmc               1/1     Running   0          1d
icp-mongodb-0                             1/1     Running   0          1d
image-manager-0                           2/2     Running   0          1d
k8s-etcd-172.16.159.245                   1/1     Running   0          1d
k8s-master-172.16.159.245                  3/3     Running   0          1d
k8s-proxy-172.16.159.245                  1/1     Running   0          1d
k8s-proxy-172.16.243.166                  1/1     Running   0          1d
k8s-proxy-172.16.243.169                  1/1     Running   0          1d
k8s-proxy-172.16.243.204                  1/1     Running   0          1d
k8s-proxy-172.16.243.207                  1/1     Running   0          1d
kube-dns-2pfvq                           3/3     Running   0          1d
```

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))

The screenshot shows the 'LDAP filters' configuration interface. It consists of two main panels: 'LDAP filters' on the left and 'User filter' on the right.

LDAP filters Panel:

- Group filter:** The value `(&(cn=%v)(objectclass=group))` is displayed, with the entire line highlighted by a red rectangular box.
- Group ID map:** The value `*:cn` is displayed.
- Group member ID map:** The value `memberOf:member` is displayed.

User filter Panel:

- User filter:** The value `(&(sAMAccountName=%v)(objectclass=user))` is displayed.
- User ID map:** The value `user:sAMAccountName` is displayed.

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/[/g[
```

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 administrator

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

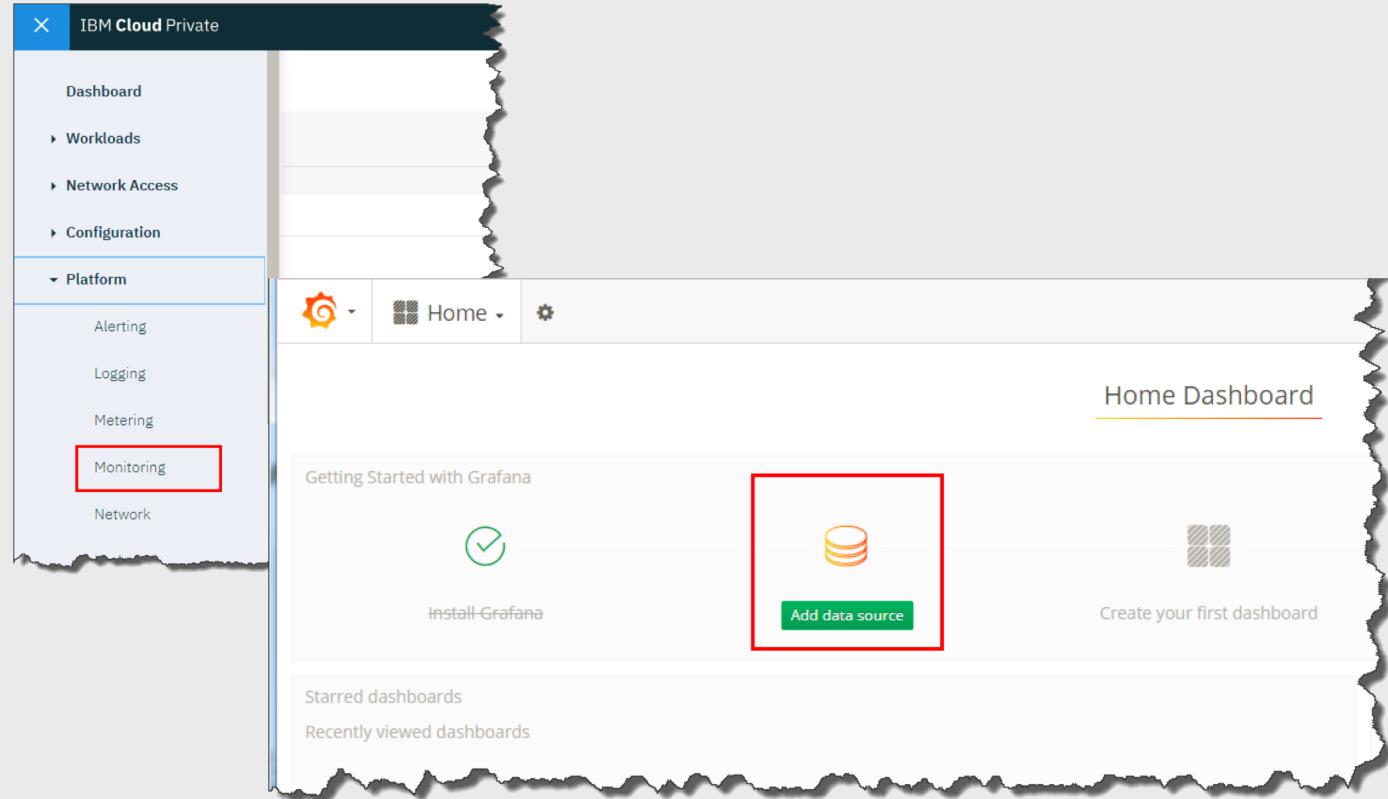
```
docker run -e LICENSE=accept --net=host \
-v "$(pwd)":/installer/cluster ibmcom/icp-inception:2.1.0.3-ee install -vvv
```

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:

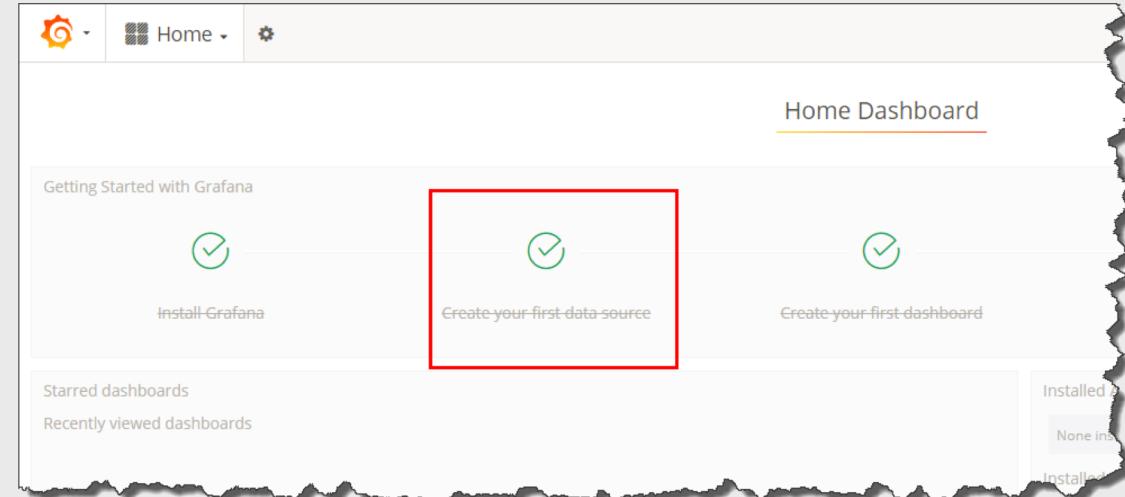
1. On the master/boot node create a yaml file with the yaml from the knowledgecenter
2. Delete the old batch job
3. Execute a new batch job with the yaml file just created

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.

The screenshot shows the 'IBM Cloud Private' website with the 'IBM Cloud Private CLI' page open. On the left, a sidebar menu is displayed with several options: Dashboard, Catalog, Workloads, Network Access, Configuration, Platform, Manage, and Command Line Tools. The 'Command Line Tools' option is expanded, showing 'Cloud Private CLI' and 'Getting started'. A purple arrow points to the 'Cloud Private CLI' link. The main content area on the right provides information about the CLI, including its purpose (to manage applications, containers, infrastructures, services, and other resources), download links for various operating systems, and usage instructions. The download links are arranged in a grid:

- DOWNLOAD FOR Mac OS X
- DOWNLOAD FOR Linux (32-bit)
- DOWNLOAD FOR Linux (64-bit)
- DOWNLOAD FOR Linux (ppc64le)
- DOWNLOAD FOR Windows (32-bit)
- DOWNLOAD FOR Windows (64-bit)

Below the download links, there is a note: "After you install the CLI, you can use it from your command line by typing bx pr [command]. For command details, see [IBM® Cloud Private CLI command reference](#)".

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"}]}}}
```

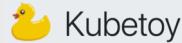
Patch the Kafka StatefulSet:

```
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



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