How to Enable and Use Private Endpoints on your Red Hat OpenShift Cluster in IBM Cloud

JAMES BELTON

GLOBAL IT ARCHITECT FOR IBM CLOUD CUSTOMER SUCCESS

MARCH 2020

# Introduction

By default, Red Hat OpenShift deployments on IBM Cloud come with public endpoints enabled only. This is fine where users want to manage their clusters via the public internet but more of a problem where customers want to manage their environment via VPN or IBM Cloud Direct Link.

In this guide, I’ll show you how to enable private endpoints for your Red Hat OpenShift environment on IBM Cloud, as well as other steps you need to take to get it working.

This document is based on OpenShift version 3.11.x. and is intended to supplement the IBM Cloud documentation. If something does not work, please refer to the official IBM Documentation in the first instance.

# Pre-requisites

This document assumes that you have some working knowledge of Red Hat OpenShift on IBM Cloud and that you have a cluster created. Note these steps only apply if you need to enable and use private end-points in your deployment.

You also need:

* The ibmcloud CLI tools installed – see

curl -sL https://ibm.biz/idt-installer | bash (MacOS users)

or https://cloud.ibm.com/docs/openshift?topic=openshift-openshift-cli

* The OpenShift CLI – see <https://mirror.openshift.com/pub/openshift-v3/clients/>
* A VPN client installed OR access via IBM Cloud Direct Link in place

Details about the IBM Cloud VPN and tools can be found here: <https://cloud.ibm.com/docs/iaas-vpn?topic=iaas-vpn-getting-started>

Note that I am using a MacBook Pro, so will use the MotionPro Plus VPN client. Details on connecting to the IBM Cloud VPN using Windows7 and higher clients can be found here: <https://cloud.ibm.com/docs/iaas-vpn?topic=iaas-vpn-connect-ssl-vpn-windows7>

## Providing Users with VPN access

IBM Cloud account users will only be able to access IBM Cloud via a VPN connection if they have been set up by an account administrator or user with user management privileges. To enable access for a user, see the following page in the IBM Cloud documentation:

<https://cloud.ibm.com/docs/iaas-vpn?topic=iaas-vpn-activate-or-deacivate-ssl-vpn-access-for-a-user>

## Setting / updating user passwords for VPN Access to IBM Cloud

The username and password with which a user accesses IBM Cloud via the VPN will not match their regular IBM Cloud login credentials and must be set up independently by the account administrator or person with privilege to do so. To set and update user passwords, see the following link in the IBM Cloud documentation: <https://cloud.ibm.com/docs/iaas-vpn?topic=iaas-vpn-update-users-ssl-vpn-password>

# Enabling Private Endpoints

To enable private endpoints on your OpenShift cluster:

## Open a terminal / command prompt, log into your Cluster

On your local machine, open a terminal window. You then need to log into your OpenShift cluster from the command line.

To find the login command for your cluster, from the IBM Cloud UI, open the OpenShift web console (press the blue button from your cluster’s pages):

A screenshot of a cell phone

Description automatically generated

And under the username in the top right corner, click *Copy Login Command*.

A screenshot of a cell phone

Description automatically generated

Then, at the command prompt in your open terminal window, paste the login command.

For example:



## Enable the Private Service Endpoint

To enable the private service endpoint, issue the following command to your cluster (replacing <<cluster-name>> with the name of your cluster:

ibmcloud oc cluster feature enable private-service-endpoint --cluster <<cluster-name>>

For example:

ibmcloud oc cluster feature enable private-service-endpoint --cluster james-broad-cluster

You’ll then see output similar to the following:

Enable the **private-service-endpoint** feature for cluster **james-broad-cluster**? **This feature cannot be disabled later.** [y/N]**>**

Respond ‘y’. You’ll then see output similar to this:

**OK**

To apply these changes, refresh the cluster's API server by running 'ibmcloud ks cluster master refresh --cluster james-broad-cluster'. Then reload the workers by running either 'ibmcloud ks worker reload' or 'ibmcloud ks worker update'.

Refresh the API servers for cluster **james-broad-cluster**? [y/N]**>**

Respond ‘y’ again.

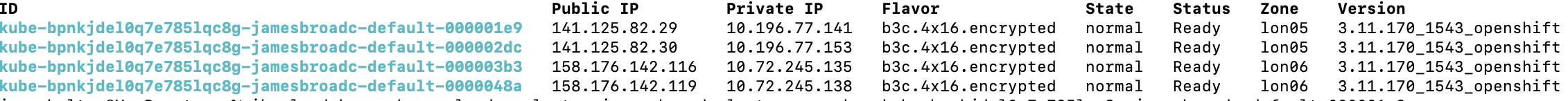
You’ll then see a message like to following:

Reload the workers by running either 'ibmcloud ks worker reload' or 'ibmcloud ks worker update'.

This means that you need to reload your worker nodes. For this you’ll need the ID’s of your worker nodes. You can find them in the UI but since we’re already in a terminal session, just issue the command:

ibmcloud ks worker ls --cluster <<cluster-name>>

You’ll get output similar to this:



Each of the nodes (in the case above, 4 of them) will need to be reloaded in turn. You can do them all at once but if you have running workloads, you may want to stagger the reloads so that you workloads don’t go off-line (as the nodes are reloaded, the cluster will restart workloads on nodes which are still running).

The reload command is as follows:

ibmcloud ks worker reload --cluster james-broad-cluster –worker <<worker-id>>

For example:

ibmcloud ks worker reload --cluster james-broad-cluster --worker kube-bpnkjdel0q7e785lqc8g-jamesbroadc-default-000001e9

When each of the workers have been reloaded, your cluster will have a private end point. You can view it using the command:

ibmcloud oc cluster get --cluster <<cluster-name>> --show-resources

This will give you output similar to the following:

A screenshot of a cell phone

Description automatically generated

You now have your private endpoint, but you still need to go through a couple or so more steps before you can connect to it.

# Accessing the Cluster through the Private Endpoint

This section of the document shows you how to connect to the cluster via the private endpoint. This is a shortened version of the documentation that you can find here: <https://cloud.ibm.com/docs/openshift?topic=openshift-access_cluster#access_private_se>

The documentation states that:

*The OpenShift master is accessible through the private service endpoint if authorized cluster users are in your IBM Cloud private network or are connected to the private network through a*[*VPN connection*](https://cloud.ibm.com/docs/iaas-vpn?topic=iaas-vpn-getting-started)*or*[*IBM Cloud Direct Link*](https://cloud.ibm.com/docs/direct-link?topic=direct-link-get-started-with-ibm-cloud-direct-link)*. However, communication with the Kubernetes master over the private service endpoint must go through the 166.X.X.X IP address range, which is not routable from a VPN connection or through IBM Cloud Direct Link. You can expose the private service endpoint of the master for your cluster users by using a private network load balancer (NLB). The private NLB exposes the private service endpoint of the master as an internal 10.X.X.X IP address range that users can access with the VPN or IBM Cloud Direct Link connection. If you enable only the private service endpoint, you can use the Kubernetes dashboard or temporarily enable the public service endpoint to create the private NLB.*

These are the steps that follow, which create the load balancer and enable log-in.

## Create a private Network Load Balancer

First of all, make sure that you are still logged into your OpenShift cluster (see how to do this in the steps above) and get details of the private endpoint by running the command

ibmcloud oc cluster get --cluster <cluster\_name >

You’ll see output like this:

A screenshot of a cell phone

Description automatically generated

In the example above, <https://c100.private.eu-gb.containers.cloud.ibm.com:31436> is the full address of the end-point, 31436 is the *private\_service\_endpoint\_port* and c100.private.eu-gb-containers.cloud.ibm.com is the *route\_address.*

Next, create a file on your file system called oc-api-via-nlb.yaml

On a Linux or MacOS system, this can be created using your favourite text editor (I use *vi*) and on Windows, you can use Textpad or an equivalent.

Add the following to the file:

apiVersion: v1

kind: Service

metadata:

name: oc-api-via-nlb

annotations:

service.kubernetes.io/ibm-load-balancer-cloud-provider-ip-type: private

namespace: default

spec:

type: LoadBalancer

ports:

- protocol: TCP

port: <private\_service\_endpoint\_port>

targetPort: <private\_service\_endpoint\_port>

---

kind: Endpoints

apiVersion: v1

metadata:

name: oc-api-via-nlb

namespace: default

subsets:

- addresses:

- ip: 172.20.0.1

ports:

- port: 2040

Replace the <private\_service\_endpoint\_port> entries with the values from your private endpoint. In the case of my cluster, I would replace these with the value 31436.

Save the file.

Now, you need to apply the configuration from the new yaml file to your cluster with this command:

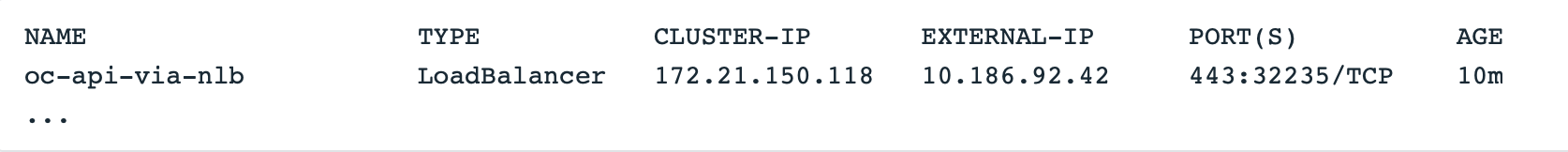
oc apply -f oc-api-via-nlb.yaml

If this command errors, then you may not be logged into your cluster. Check and try again.

This will create the private network load balancer (NLB). Let’s check it’s there. Run the command:

oc get svc -o wide

You’ll see something like this:



This denotes that the IP address for the private service endpoint of the master is 10.186.92.42.

Next, use this to provide access to the enpoint via the hosts file on authorised user’s client machines. On a Linux or MacOS client, edit the /etc/hosts file, on a Windows client, edit the file C:\Windows\System32\drivers\etc\hosts

In both cases, enter a line similar to the following:

10.186.92.42 c100.private.eu-gb-containers.cloud.ibm.com

Where c100.private.eu-gb-containers.cloud.ibm.com is the *route\_address* obtained earlier from running the command

ibmcloud oc cluster get --cluster <cluster\_name >

Save the hosts file.

## Create an API Key

You need to create an API key so that you can log into the service. Do this by runnining the following command from your terminal window:

ibmcloud iam api-key-create <key-name> --file <file-name>

Replace the <key-name> and <file-name> values with meaningful ones. For example:

ibmcloud iam api-key-create my-cluster-key --file my-apikey

This will write the key information to a file (in this case) called my-apikey. You can omit the –file <file-name> part of the command if you wish but be aware that you need to save the key details somewhere as you cannot retrieve them again, only create a new key.

If you look inside the file, you will see some JSON lines, one of which looks like this:

"apikey": "8-NrXXXFhXXXsy2XXXPYsVXXXJ0VmXXXihCXXXQMXXXs"

This line contains your API Key ‘password’.

## Connect to a VPN or Direct Link

To be able to connect to the private endpoint, you need to be connected to a private connection. This can either be a VPN tunnel, or an IBM Cloud Direct Link connection. Make sure that you have connectivity to the datacentre or zones in which your client resides. Check the pre-requisites section about using a VPN client. The following link provides the available VPN endpoints that the VPN client can attach to: <https://cloud.ibm.com/docs/iaas-vpn?topic=iaas-vpn-available-vpn-endpoints>

My cluster is in the London region, so my VPN client is attached to vpn.lon.softlayer.com.

## Log into the OpenShift Cluster

You can now log into the OpenShift cluster on the private endpoint using the following command in the terminal window:

oc login -u apikey -p <API\_key> --server=<private\_service\_endpoint>

Replace <API\_key> with the ‘password’ obtained above and the <private\_service\_endpoint> value with the full endpoint URL, for example:

oc login -u apikey -p 8-NrXXXFhXXXsy2XXXPYsVXXXJ0VmXXXihCXXXQMXXXs --server=https://c100.private.eu-gb.containers.cloud.ibm.com:31436

Your session should then connect, and you will see output similar to:

You have access to the following projects and can switch between them with 'oc project <projectname>':

\* default

ibm-cert-store

ibm-system

james-mongo-proj

kube-proxy-and-dns

kube-public

kube-service-catalog

kube-system

openshift

openshift-ansible-service-broker

openshift-console

openshift-infra

openshift-monitoring

openshift-node

openshift-template-service-broker

openshift-web-console

Using project "default".

Enter oc version and you will then see some details about your cluster, including the address of the endpoint you are attached to, which should be the private one. For example:

A close up of a sign

Description automatically generated

The private endpoint is now configured and can be used to manage the cluster.