Adding the Acadia Beta CSI driver to your IKS/ROKS cluster

This document describes the steps to add the IBM VPC CSI driver with Acadia support to your IKS/OpenShift cluster.

## Requirements

The CSI drivers is built on the IBM VPC CSI block driver version 5.2.x as a result it has the following Kubernetes / OpenShift requirements

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Version | Supported Kubernetes Range | Supported OpenShift Range |
| vpc-block-csi-driver | 5.2 | >=1.25.0 <1.31.0 | >=4.10.0 <4.16. |

### Access to the CSI driver Acadia enabled container image

The container image needed by the CSI driver needs to be available in your kiubernetes/Openshift cluster. This can be accomplished in 2 ways:

1. Request for your account to be added to the Acadia CopntainerRegistry containing the image
2. Use your own IBM Cloud ContainerRegistry and push the docker image to your CR. See instructions below on loading the image to your CR.

### Remove previous IBM VPC block driver

If you have a managed IKS or ROKS cluster you will need to remove the preinstalled CSI driver. Note that doing so will remove any VPOC block volumes you may have created via PVCs.

See if the addon is installed in your cluster, run

*ibmcloud ks cluster addon ls --cluster <cluster\_bname\_or\_id>*

if the above command shows the ‘***vpc-block-csi-driver***’ addon you will need to remove it by running

*ibmcloud ks cluster addon disable vpc-block-csi-driver --cluster <cluster\_bname\_or\_id>*

## Installing the Driver

### Download the install package

Download the ibm-vpc-block-csi-driver-v5.2.acadia-beta.driver.tgz file from the ‘Acadia Beta Enablement’ folder in Box. Same location as this document.

Unpack the driver files to where you’ll apply to your cluster.

### Loading container image to your ContainerRegistry

If you will be using your own container registry for the csi driver image, you will need to Download the ibm-vpc-block-csi-driver-v5.2.acadia-beta.tgz file from the ‘Acadia Beta Enablement’ folder in Box. Same location as this document. Once download load this into your local docker/podman/etc registry

*docker load -I the ibm-vpc-block-csi-driver-v5.2.acadia-beta.****tgz***

Once loaded you will need to ensure your logged into your CR and then tag and push to your CR

*ibmcloud cr login*

*docker tag <image id> icr.io/<your\_cr>/ibm-vpc-block-csi-driver:amd64-sdp-beta*

*docker push icr.io/<your\_cr>/ibm-vpc-block-csi-driver:amd64-sdp-beta*

### Modify the driver files for your env

If using your own CR, modify the 5.2\_acadia.yaml file and replace

icr.io/acadia-images/ibm-vpc-block-csi-driver:amd64-sdp-beta

with the location of the image you loaded and uploaded to your CR above. There should be 2 references in the file.

Update the slclient\_Gen2.toml to container the resource group id (g2\_resource\_group\_id) and account ApiKey (g2\_api\_key) used when provisioning VPC block volumes. If not using LON (eu-gb\_ update the RIAAS endpoint usrl (g2\_riaas\_endpoint\_url) for the region you’ll be using

### Setup your kubectl env for your cluster

Ensure you are logged into the IBM cli and that your kubectl env is setup to use your cluster

* *ibmcloud ks cluster config --admin --cluster <cluster\_name\_or\_id>*

### Create docker secret

You will need an apikey for the account authorized to pull from the container registry that contains the csi driver image. Either the acadia beta one or your own. Be sure to replace *‘<iam-api-key>’* with your api key

*kubectl create secret docker-registry icr-io-secret --docker-server=icr.io --docker-username=iamapikey --docker-password=-<iam-api-key> --docker-email=iamapikey -n kube-system*

### Install the driver

Run the `deploy-vpc-block-driver.sh` script.

## StorageClasses

As part of the driver install the normal VPC block storage classes are created as well as a set of storageclasses for the new SDP profile. We also provide some examples you may use to create your own SCs if you want to change IOPS or other defaults. The Kubernetes PV/PVC doesn’t allow for specifying IOPS for each PV, this is settable only in the SC. You will need to use your own SCs (based on our ibm-vpc-block-sdp-with-secret example) if you want different IOPS values or use the IBM PVC CSI driver secret technique to provide some config per PVC