
IBM Spectrum Scale CSI Operator

Release 0.9.1

Dec 05, 2019

Contents

1	Table of Contents	1
1.1	Getting Started	1
1.2	Deployment	3
1.3	Troubleshooting	5
1.4	Developers	5

1.1 Getting Started

1.1.1 Quickstart Guide

The IBM Spectrum Scale CSI Operator runs within a Kubernetes cluster providing a means to deploy and manage the CSI plugin for spectrum scale. For more in depth documentation please refer to the [README](#).

This operator should be used to deploy the CSI plugin.

The configuration process is as follows:

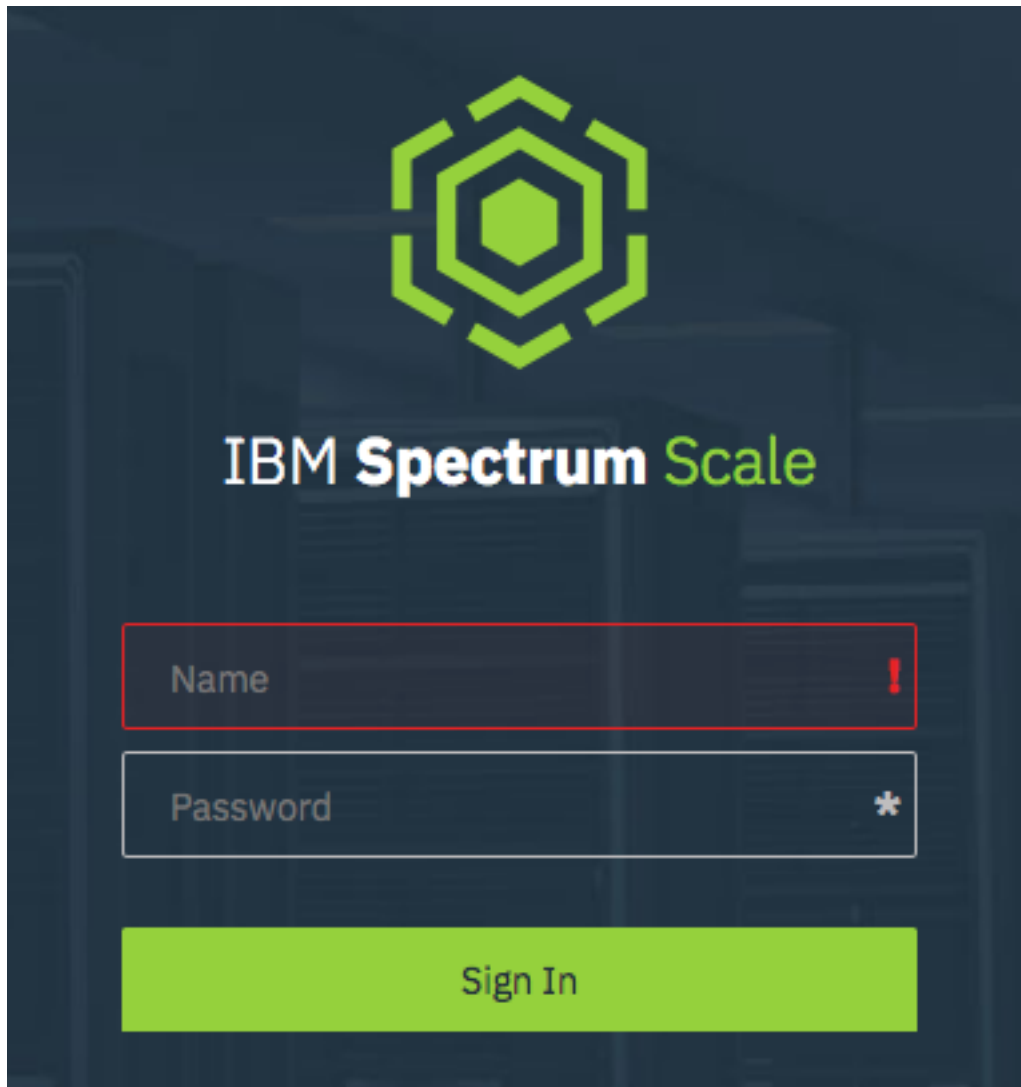
1. *Spectrum Scale GUI Setup*
2. *Custom Resource Configuration*

Spectrum Scale GUI Setup

NOTE: This step only needs to be performed once per GUI.

WARNING: If your daemonset pods (driver pods) do not come up, generally this means you have a secret that has not been defined in the correct namespace.

1. Ensure the Spectrum Scale GUI is running by pointing your browser to the IP hosting the GUI:



If you do not see a login follow on screen instructions, or review the [GUI Documentation](#)

2. Create a CsiAdmin group account on in the GUI (currently requires a CLI call):

```
export USERNAME="SomeUser"
export PASSWORD="SomePassword"
/usr/lpp/mmfs/gui/cli/mkuser ${USERNAME} -p ${PASSWORD} -g CsiAdmin
```

3. Create a Kubernetes secret for the CsiAdmin user:

```
export USERNAME_B64=$(echo $USERNAME | base64)
export PASSWORD_B64=$(echo $PASSWORD | base64)
export OPERATOR_NAMESPACE="ibm-spectrum-scale-csi-driver" # Set this to the
↪namespace you deploy the operator in.

cat << EOF > /tmp/csisecret.yaml
apiVersion: v1
data:
```

(continues on next page)

(continued from previous page)

```

password: ${PASSWORD_B64}
username: ${USERNAME_B64}
kind: Secret
type: Opaque
metadata:
  name: csisecret      # This should be in your CSIScaleOperator definition
  namespace: ${OPERATOR_NAMESPACE}
  labels:
    app.kubernetes.io/name: ibm-spectrum-scale-csi-operator # Used by the operator to
    ↪ detect changes, set on load of CR change if secret matches name in CR and namespace.
EOF

kubectl create -f /tmp/csisecret.yaml
rm -f /tmp/csisecret.yaml

```

Custom Resource Configuration

The bundled Custom Resource example represents the minimum settings needed to run the operator. If your environment needs more advanced settings (e.g. remote clusters, node mapping, etc.) please refer to the sample [Custom Resource](#).

1.2 Deployment

1.2.1 Manual

Note: For OpenShift environments, replace `kubectl` with `oc`

The following `.yaml` files need to be applied to your cluster

namespace.yaml This configuration file creates the `ibm-spectrum-scale-csi-driver` namespace

ibm-spectrum-scale-csi-operator.yaml This is an auto-generated combined configuration file that starts the operator.

ibm-spectrum-scale-csi-operator-cr.yaml This is a custom resource file (CR) that the admin must modify to match their Spectrum Scale install, which loads the `csi-driver` plugin.

1. Download the `.yaml` files from the code repository

```

curl -O https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-csi-operator/master/
    ↪ stable/ibm-spectrum-scale-csi-operator-bundle/operators/ibm-spectrum-scale-csi-
    ↪ operator/deploy/namespace.yaml
curl -O https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-csi-operator/master/
    ↪ generated/installer/ibm-spectrum-scale-csi-operator.yaml
curl -O https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-csi-operator/master/
    ↪ stable/ibm-spectrum-scale-csi-operator-bundle/operators/ibm-spectrum-scale-csi-
    ↪ operator/deploy/crds/ibm-spectrum-scale-csi-operator-cr.yaml

```

2. Apply the namespace and operator configuration files.

```

kubectl apply -f namespace.yaml
kubectl apply -f ibm-spectrum-scale-csi-operator.yaml

```

3. Create and apply the secret for the Spectrum Scale GUI.

Create a file `secret.json` with the following, replacing the `name` | `username` | `password` fields.

```
{
  "apiVersion": "v1",
  "kind": "List",
  "items":
  [{
    "kind": "Secret",
    "apiVersion": "v1",
    "metadata": {
      "name": "{{ gui_secret_name }}",
      "label": {
        "app.kubernetes.io/name": "ibm-spectrum-scale-csi-operator"
      }
    },
    "data": {
      "username": "{{ gui_user | b64encode }}",
      "password": "{{ gui_pass | b64encode }}"
    }
  ]
}
```

Then apply with the following command:

```
kubectl apply -f secret.json
```

4. Edit and apply the `ibm-spectrum-scale-csi-operator-cr.yaml` file to start the csi-driver plugin.

```
# Modify this file to match your environment properties
kubectl apply -f ibm-spectrum-scale-csi-operator-cr.yaml
```

1.2.2 Operator Lifecycle Manager (OLM)

Note: For OpenShift environments, replace `kubectl` with `oc`

1. Install OLM:

```
curl -sL https://github.com/operator-framework/operator-lifecycle-manager/releases/
↳download/0.11.0/install.sh | bash -s 0.11.0
```

2. Download the CSI Operator `.yaml` and apply

```
curl https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-csi-operator/master/
↳stable/ibm-spectrum-scale-csi-operator-bundle/operators/ibm-spectrum-scale-csi-
↳operator/deploy/olm-scripts/operator-source.yaml > operator-source.yaml

kubectl apply -f operator-source.yaml
```


1.3 Troubleshooting

1.3.1 operator-sdk

Removing Stuck Operator [operator-sdk/issue/2094]

In cases where deleting the operators Custom Resource fails, the following can be executed:

```
# This may need to be customized in OLM environments:
NAMESPACE=ibm-spectrum-scale-csi-driver
kubectl get csiscaleoperators -n ${NAMESPACE} -o json | jq '.spec = {"finalizers":[]}'
↪ ' >temp.json
curl -k -H "Content-Type: application/json" -X PUT --data-binary @temp.json 127.0.0.
↪ 1:8001/api/v1/namespaces/${NAMESPACE}/finalize
rm -f temp.json
```

Typically this happens when deleting the Custom Resource Definition before removing all of the Custom Resources.

For more details on this check the following [operator-sdk/issue/2094](#).

1.4 Developers

1.4.1 Clone and Build

Clone

Warning: This repository needs to be accessible in your GOPATH. The examples use the `root` user and `GOPATH=/root/go`

Warning: Due to current constraints in golang, relative paths are not supported. You **must** clone this repository under your GOPATH. If not, the `operator-sdk` build operation may fail.

```
1 # Set up some helpful variables
2 export GOPATH="/root/go"
3 export IBM_DIR="${GOPATH}/src/github.com/IBM"
4
5 # Ensure the dir is present then clone.
6 mkdir -p ${IBM_DIR}
7 cd ${IBM_DIR}
8 git clone https://github.com/IBM/ibm-spectrum-scale-csi-operator.git
```

Build

Environment

To assist in proper configuration of the build environment, a playbook is provided:

```
ansible-playbook $GOPATH/src/github.com/IBM/ibm-spectrum-scale-csi-operator/ansible/
↳ dev-env-playbook.yaml
```

Create the the Image

Navigate to the operator directory and use `operator-sdk` to build the container image.

```
# IBM_DIR is defined in the previous step
export OPERATOR_DIR="$IBM_DIR/ibm-spectrum-scale-csi-operator"
cd ${OPERATOR_DIR}/stable/ibm-spectrum-scale-csi-operator-bundle/operators/ibm-
↳ spectrum-scale-csi-operator

export GO111MODULE="on"
operator-sdk build csi-scale-operator
```

Note: This requires `docker`

1.4.2 Deployment

Container Repository

In order to use the container image that you just built in the previous step, the image needs to be pushed to some container repository.

- **Quay.io (recommended)**

Follow this tutorial to configure [quay.io](#).

Create a repository called `ibm-spectrum-scale-csi-operator`.

- **Docker**

Deploying your own Docker registry is an [involved process](#) and outside of the scope of this document.

The documentation will assume that the `quay.io` path is being used.

Pushing the image

Once you have a repository ready:

```
# Authenticate to quay.io
docker login <credentials> quay.io

# Tag the build
docker tag csi-scale-operator quay.io/<your-user>/ibm-spectrum-scale-csi-operator:v0.
↳ 9.1

# push the image
docker push quay.io/<your-user>/ibm-spectrum-scale-csi-operator:v0.9.1

# Update your deployment to point at your image.
hack/change_deploy_image.py -i quay.io/<your-user>/ibm-spectrum-scale-csi-
↳ operator:v0.9.1
```

Installing Operator

Note: For OpenShift environments, replace `kubectl` with `oc`

Run the following to deploy the operator manually:

```
cd ${OPERATOR_DIR}/stable/ibm-spectrum-scale-csi-operator-bundle/operators/ibm-
  ↳ spectrum-scale-csi-operator

kubectl apply -f deploy/namespace.yaml
kubectl apply -f deploy/operator.yaml
kubectl apply -f deploy/role.yaml
kubectl apply -f deploy/role_binding.yaml
kubectl apply -f deploy/service_account.yaml
kubectl apply -f deploy/crds/ibm-spectrum-scale-csi-operator-crd.yaml
```

Starting the CSI Driver

Note: Before starting the plugin, add any GUI secrets to the appropriate namespace.

A Custom Resource (CR) file is provided `deploy/crds/ibm-spectrum-scale-csi-operator-cr.yaml`. Modify this file to match the properties in your environment.

To start:

```
kubectl apply -f deploy/crds/ibm-spectrum-scale-csi-operator-cr.yaml
```

To stop:

```
kubectl delete -f deploy/crds/ibm-spectrum-scale-csi-operator-cr.yaml
```

Removing the CSI Operator

To remove the operator:

```
# The following removes the csi-driver
kubectl delete -f deploy/crds/ibm-spectrum-scale-csi-operator-cr.yaml

# The following removes the csi-operator
kubectl delete -f deploy/operator.yaml
kubectl delete -f deploy/role.yaml
kubectl delete -f deploy/role_binding.yaml
kubectl delete -f deploy/service_account.yaml
kubectl delete -f deploy/crds/ibm-spectrum-scale-csi-operator-crd.yaml

# The following removes the namespace
kubectl delete -f deploy/namespace.yaml
```

This will completely destroy the operator and all associated resources.