# Spark Operator

## Created resources

### Deployments

Spark Operator is running its own Deployments called ‘spark-operator-xxxx’. One is spark operator controller and second is spark operator webhook (if using webhook).

### Pods

There are Pods running for each Deployment.

### Services

Spark Operator is running its own Services called ‘spark-operator-xxxx’.

### Service Accounts

Spark Operator’s Pods are using Service Accounts which need to have proper permissions. We can check what are those Accounts by checking configurations of Pods:

Kubectl get pod <pod-name> -n <namespace> -o yaml

## Webhooks

Spark Operator uses webhooks to talk to a Kubernetes API server in order to modify resources. They are using certificates for security. Certificates files are mounted into a webhook Pod.

We can check how certificates are mounted it by checking webhook Pod specification:

* kubectl get pod -n <namespace> <webhook-pod-name> -o yaml

Webhooks allows us to provide some additional fields for Driver and Executors Pods specification, which are not supported by the SparkApplication. For example adding tolerations.

We can also use PodTemplateFile to add additional specification fields.

## Taints and tolerations

In order to run Spark Operator on a master node (control plane) we need to set up proper tolerations for spark-operator-controller and spark-operator-webhook (if webhook is enabled) deployments.

We can run Spark Operator normally on worker nodes, without setting up tolerations.

We need to patch and restart deployments for both spark-operator-controller and spark-operator-webhook:

A screen shot of a computer

AI-generated content may be incorrect.

# SparkApplication

## Created resources

After deploying SparkApplication resource it creates:

* SparkApplication resource
* Spark Driver Pod
* Spark Executor Pods

## VolumeMounts

We can use volume mounts in the SparkApplication in order to access data from a volume in the Driver and Executors Pods.

## Taints and tolerations

In order to deploy sparkApplication on a master node (control plane) we need to set up proper tolerations for SparkApplication.

We can deploy SparkApplication normally on worker nodes.

We can specify tolerations for Driver and Executors Pods in the Spark Application manifest:

A screenshot of a computer code

AI-generated content may be incorrect.

Adding tolerations requires using webhook.

## imagePullSecrets

We can use the imagePullSecrets field in the Spark Application manifest in order to access secrets which contains values for authentication to container registry from which we want to pull a Docker image used in the Spark Application.

For example we can create a secret for container registry like that:

A screen shot of a computer code

AI-generated content may be incorrect.

And use it in Spark Application:

A black screen with white text

AI-generated content may be incorrect.

# Pod Template File

If we can’t add some fields in the SparkApplication manifest, then we can use Pod Template File.

We create a separate YAML manifest, which is our Pod Template File, and then we specify in Spark configuration in the Spark Application that we want to use it.

**Issues:**

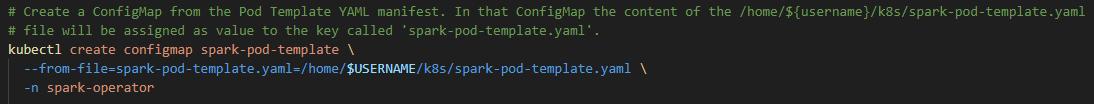
I don’t know how to use it, I was trying different options:

Create a Pod Template File:

A screen shot of a computer

AI-generated content may be incorrect.

Add the content of that Pod Template File to the ConfigMap value:



Mount Pod Template File from configMap to the driver and executor pods, and specify in spark configuration that we want to use thos Pod Template Files:

A screenshot of a computer

AI-generated content may be incorrect.

I was also trying to patch spark operator controller deployment to mount that volume with Pod Template File to it:

A screenshot of a computer program

AI-generated content may be incorrect.

I was always getting error that the Pod Template File can’t be found.

Someone on a forum said that they have mounted Pod Template File to Spark Operator but I don’t know how.