Pipeline Resources

At the end of this chapter you will be able to :

- Understand what is a pipeline resource ?
- Create a pipeline resource

Prerequisite

The following checks ensure that each chapter exercises are done with the right environment settings.

Kubernetes **OpenShift**

- Set your local docker to use minikube docker daemon

```
eval $(minikube docker-env) &&\
minikube profile tektontutorial
```

- Kubernetes should be v1.18+

```
kubectl version --short
```

The output should be like

```
Client Version: v1.18.6
Server Version: v1.18.3
```

Ensure Tekton piplines is deployed and the API is available for use

```
kubectl api-resources --api-group='tekton.dev'
```

The command show an output like:

```
NAME                SHORTNAMES   APIGROUP     NAMESPACED   KIND
clustertasks                     tekton.dev   false        ClusterTask
conditions                       tekton.dev   true         Condition
pipelineresources                tekton.dev   true         PipelineResource
pipelineruns        pr,prs       tekton.dev   true         PipelineRun
pipelines                        tekton.dev   true         Pipeline
taskruns            tr,trs       tekton.dev   true         TaskRun
tasks                            tekton.dev   true         Task
```

Create a pipeline resource

If you are not in tutorial chapter folder, then navigate to the folder:

```
cd $TUTORIAL_HOME/resources
```

The following snippet shows what a Tekton `PipelineResource` YAML looks like:

build-resources.yaml

```
apiVersion: tekton.dev/v1alpha1
kind: PipelineResource
metadata:
  name: git-source
spec:
  type: git
  params:
    - name: url
      value: https://github.com/redhat-scholars/tekton-tutorial-greeter
    - name: revision
      value: staging
---
apiVersion: tekton.dev/v1alpha1
kind: PipelineResource
metadata:
  name: tekton-tutorial-greeter-image
spec:
  type: image
  params:
    - name: url
      # use internal registry
      value: example.com/rhdevelopers/tekton-tutorial-greeter
      # if you are on OpenShift uncomment the line below
      #value: "image-registry.openshift-image-registry.svc:5000/tektontutorial
```

Each pipeline resource has:

- **name**: the name using which it will be referred in other places
- **type**: the type of the pipeline resource, in this example we have two types
  - *git* - this type of resource refers to a GitHub repository
  - *image* - this type of resource is linux container image
- **params**: each type can have one or more parameters that will be used to configure the underlying type. In the above example for the `git-source` pipeline resource, the parameters `url` and `revision` are used to identify the GitHub repository url and revision of the sources respectively.

More details on other types of pipeline resource types is available here.

Deploy a pipeline resource

The pipeline resource could be created using the command:

```
kubectl apply -n tektontutorial -f build-resources.yaml
```

```
pipelineresource.tekton.dev/git-source created
pipelineresource.tekton.dev/tekton-tutorial-greeter-image created
```

See what you have deployed

We will use the Tekton cli to inspect the created resources

```
tkn res ls
```

The above command should list two resources as shown below:

```
NAME                            TYPE    DETAILS
git-source                      git     url: https://github.com/redhat-scholar
tekton-tutorial-greeter-image   image   url: example.com/rhdevelopers/tekton-t
```

All Tekton API resources/objects has the **describe** option that gives more details of respective Tekton API object.

e.g. To describe the PipelineResource that we just created, run:

```
tkn res describe git-source
```

```
Name:                    git-source
Namespace:               tektontutorial
PipelineResource Type:   git

Params
NAME       VALUE
url        https://github.com/redhat-scholars/tekton-tutorial
revision   master

Secret Params
No secret params
```

TIP

Use the command **help** via `tkn res --help`

Prev
Setup

Next
Tasks