

# Apache Airflow Deployment

To start the deployment, we need to be logged into the cluster:

# Replace the command with your own one inside the single quotes and run the cell  
# Example OC\_LOGIN\_COMMAND='oc login --token=sha256~3bR5KXgwiUoaQiph2\_kIXCDQnVfm\_HQy3YwU2m-UOrs --server=https://c109-e.us-east.containers.cloud.ibm.com:31656'  
OC\_LOGIN\_COMMAND='oc login --token=sha256~2uCCV-OLqY\_c2BNNn4OVo-PtJQIn0Hpx0oikvSTGEro --server=https://c103-e.us-south.containers.cloud.ibm.com:32536'  
$OC\_LOGIN\_COMMAND

Logged into "https://c103-e.us-south.containers.cloud.ibm.com:32536" as "IAM#angelito@de.ibm.com" using the token provided.  
  
You have access to 67 projects, the list has been suppressed. You can list all projects with 'oc projects'  
  
Using project "default".

In order to identify and separate Airflow from the rest of the cluster, we create a project called airflow

oc new-project airflow

Now using project "airflow" on server "https://c103-e.us-south.containers.cloud.ibm.com:32536".  
  
You can add applications to this project with the 'new-app' command. For example, try:  
  
 oc new-app rails-postgresql-example  
  
to build a new example application in Ruby. Or use kubectl to deploy a simple Kubernetes application:  
  
 kubectl create deployment hello-node --image=k8s.gcr.io/e2e-test-images/agnhost:2.33 -- /agnhost serve-hostname

Now, we get the official helm charts for Airflow:

helm repo add apache-airflow https://airflow.apache.org  
helm repo update

"apache-airflow" already exists with the same configuration, skipping  
Hang tight while we grab the latest from your chart repositories...  
...Successfully got an update from the "apache-airflow" chart repository  
Update Complete. ⎈Happy Helming!⎈

The next commands avoid security errors that cause deployment failures.

oc adm policy add-scc-to-group anyuid system:serviceaccounts:airflow   
oc adm policy add-scc-to-group privileged system:serviceaccounts:airflow

clusterrole.rbac.authorization.k8s.io/system:openshift:scc:anyuid added: "system:serviceaccounts:airflow"  
clusterrole.rbac.authorization.k8s.io/system:openshift:scc:privileged added: "system:serviceaccounts:airflow"

And the actual deployment starts now:

helm upgrade --install airflow apache-airflow/airflow --namespace airflow

If the deployment went well, you will see several pods running:

oc get pods

NAME READY STATUS RESTARTS AGE  
airflow-postgresql-0 1/1 Running 0 7m52s  
airflow-redis-0 1/1 Running 0 7m52s  
airflow-scheduler-d754df798-d2tdp 2/2 Running 0 7m53s  
airflow-statsd-d8c8f886c-hhlwp 1/1 Running 0 7m53s  
airflow-triggerer-7f77657d87-vxhrw 1/1 Running 0 7m53s  
airflow-webserver-6489c6c947-p7jrl 1/1 Running 0 7m53s  
airflow-worker-0 2/2 Running 0 7m52s

Next, you we need to add a route to the custer to access Airflow:

oc get svc  
oc expose svc airflow-webserver

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE  
airflow-postgresql ClusterIP 172.21.64.58 <none> 5432/TCP 8m1s  
airflow-postgresql-hl ClusterIP None <none> 5432/TCP 8m1s  
airflow-redis ClusterIP 172.21.150.136 <none> 6379/TCP 8m1s  
airflow-statsd ClusterIP 172.21.28.183 <none> 9125/UDP,9102/TCP 8m1s  
airflow-webserver ClusterIP 172.21.87.154 <none> 8080/TCP 8m1s  
airflow-worker ClusterIP None <none> 8793/TCP 8m1s  
route.route.openshift.io/airflow-webserver exposed

And, just like we did with databand, we get access url from the OpenShift console:



You can login with the default user: admin and default password: admin. Then you will see the main dashboard of Airflow



Indeed, the main dashboard is empty because the default helm deployment disables the default DAGs. If you really miss them, you can modify an environment and re-deploy Airflow

helm upgrade airflow apache-airflow/airflow -f - << EOF  
extraEnv: |  
 - name: AIRFLOW\_\_CORE\_\_LOAD\_EXAMPLES  
 value: 'True'   
EOF

And now, you will see them:



Next Section: [Airflow integration](./4_airflow_int.ipynb). Previous Section: [Databand deployment](./2_databand_deploy.ipynb)

[Return to main](../README.md)