



Apache Airflow

NetApp Solutions

NetApp
July 31, 2024

This PDF was generated from https://docs.netapp.com/us-en/netapp-solutions/ai/aicp_apache_airflow_deployment.html on July 31, 2024. Always check docs.netapp.com for the latest.

Table of Contents

- Apache Airflow 1
- Apache Airflow Deployment 1
- Use the NetApp DataOps Toolkit with Airflow 4

Apache Airflow

Apache Airflow Deployment

This section describes the tasks that you must complete to deploy Airflow in your Kubernetes cluster.



It is possible to deploy Airflow on platforms other than Kubernetes. Deploying Airflow on platforms other than Kubernetes is outside of the scope of this solution.

Prerequisites

Before you perform the deployment exercise that is outlined in this section, we assume that you have already performed the following tasks:

1. You already have a working Kubernetes cluster.
2. You have already installed and configured NetApp Astra Trident in your Kubernetes cluster. For more details on Astra Trident, refer to the [Astra Trident documentation](#).

Install Helm

Airflow is deployed using Helm, a popular package manager for Kubernetes. Before you deploy Airflow, you must install Helm on the deployment jump host. To install Helm on the deployment jump host, follow the [installation instructions](#) in the official Helm documentation.

Set Default Kubernetes StorageClass

Before you deploy Airflow, you must designate a default StorageClass within your Kubernetes cluster. The Airflow deployment process attempts to provision new persistent volumes using the default StorageClass. If no StorageClass is designated as the default StorageClass, then the deployment fails. To designate a default StorageClass within your cluster, follow the instructions outlined in the [Kubeflow Deployment](#) section. If you have already designated a default StorageClass within your cluster, then you can skip this step.

Use Helm to Deploy Airflow

To deploy Airflow in your Kubernetes cluster using Helm, perform the following tasks from the deployment jump host:

1. Deploy Airflow using Helm by following the [deployment instructions](#) for the official Airflow chart on the Artifact Hub. The example commands that follow show the deployment of Airflow using Helm. Modify, add, and/or remove values in the `custom-values.yaml` file as needed depending on your environment and desired configuration.

```
$ cat << EOF > custom-values.yaml
#####
# Airflow - Common Configs
#####
airflow:
  ## the airflow executor type to use
```

```

##
executor: "CeleryExecutor"
## environment variables for the web/scheduler/worker Pods (for
airflow configs)
##
#
#####
# Airflow - WebUI Configs
#####
web:
  ## configs for the Service of the web Pods
  ##
  service:
    type: NodePort
#####
# Airflow - Logs Configs
#####
logs:
  persistence:
    enabled: true
#####
# Airflow - DAGs Configs
#####
dags:
  ## configs for the DAG git repository & sync container
  ##
  gitSync:
    enabled: true
    ## url of the git repository
    ##
    repo: "git@github.com:mboglesby/airflow-dev.git"
    ## the branch/tag/sha1 which we clone
    ##
    branch: master
    revision: HEAD
    ## the name of a pre-created secret containing files for ~/.ssh/
    ##
    ## NOTE:
    ## - this is ONLY RELEVANT for SSH git repos
    ## - the secret commonly includes files: id_rsa, id_rsa.pub,
known_hosts
    ## - known_hosts is NOT NEEDED if `git.sshKeyscan` is true
    ##
    sshSecret: "airflow-ssh-git-secret"
    ## the name of the private key file in your `git.secret`
    ##

```

```

## NOTE:
## - this is ONLY RELEVANT for PRIVATE SSH git repos
##
sshSecretKey: id_rsa
## the git sync interval in seconds
##
syncWait: 60
EOF
$ helm install airflow airflow-stable/airflow -n airflow --version 8.0.8
--values ./custom-values.yaml
...
Congratulations. You have just deployed Apache Airflow!
1. Get the Airflow Service URL by running these commands:
    export NODE_PORT=$(kubectl get --namespace airflow -o
jsonpath="{.spec.ports[0].nodePort}" services airflow-web)
    export NODE_IP=$(kubectl get nodes --namespace airflow -o
jsonpath="{.items[0].status.addresses[0].address}")
    echo http://$NODE_IP:$NODE_PORT/
2. Open Airflow in your web browser

```

2. Confirm that all Airflow pods are up and running. It may take a few minutes for all pods to start.

```

$ kubectl -n airflow get pod
NAME                                READY   STATUS    RESTARTS   AGE
airflow-flower-b5656d44f-h8qjk      1/1     Running   0           2h
airflow-postgresql-0                1/1     Running   0           2h
airflow-redis-master-0              1/1     Running   0           2h
airflow-scheduler-9d95fcdf9-clf4b  2/2     Running   2           2h
airflow-web-59c94db9c5-z7rg4       1/1     Running   0           2h
airflow-worker-0                    2/2     Running   2           2h

```

3. Obtain the Airflow web service URL by following the instructions that were printed to the console when you deployed Airflow using Helm in step 1.

```

$ export NODE_PORT=$(kubectl get --namespace airflow -o
jsonpath="{.spec.ports[0].nodePort}" services airflow-web)
$ export NODE_IP=$(kubectl get nodes --namespace airflow -o
jsonpath="{.items[0].status.addresses[0].address}")
$ echo http://$NODE_IP:$NODE_PORT/

```

4. Confirm that you can access the Airflow web service.

The screenshot shows the Airflow web interface with the 'DAGs' tab selected. The page title is 'DAGs' and there is a search bar. Below the search bar is a table listing various DAGs. The table has the following columns: DAG, Schedule, Owner, Recent Tasks, Last Run, DAG Runs, and Links. The 'Recent Tasks' column contains a grid of task status icons. The 'DAG Runs' column contains a grid of run status icons. The 'Links' column contains icons for viewing the DAG, its logs, and its history.

DAG	Schedule	Owner	Recent Tasks	Last Run	DAG Runs	Links
ai_training_run	None	NetApp				
create_data_scientist_workspace	None	NetApp				
example_bash_operator	0 0 * * *	Airflow				
example_branch_dop_operator_v3	* * * * *	Airflow				
example_branch_operator	@daily	Airflow				
example_complex	None	airflow				
example_external_task_marker_child	None	airflow				
example_external_task_marker_parent	None	airflow				
example_http_operator	1 day, 0:00:00	Airflow				
example_kubernetes_executor_config	None	Airflow				
example_nested_branch_dag	@daily	airflow				
example_passing_params_via_test_command	* * * * *	airflow				
example_pig_operator	None	Airflow				
example_python_operator	None	Airflow				
example_short_circuit_operator	1 day, 0:00:00	Airflow				
example_skip_dag	1 day, 0:00:00	Airflow				

Use the NetApp DataOps Toolkit with Airflow

The [NetApp DataOps Toolkit for Kubernetes](#) can be used in conjunction with Airflow. Using the NetApp DataOps Toolkit with Airflow enables you to incorporate NetApp data management operations, such as creating snapshots and clones, into automated workflows that are orchestrated by Airflow.

Refer to the [Airflow Examples](#) section within the NetApp DataOps Toolkit GitHub repository for details on using the toolkit with Airflow.

Copyright information

Copyright © 2024 NetApp, Inc. All Rights Reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP “AS IS” AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

LIMITED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (b)(3) of the Rights in Technical Data -Noncommercial Items at DFARS 252.227-7013 (FEB 2014) and FAR 52.227-19 (DEC 2007).

Data contained herein pertains to a commercial product and/or commercial service (as defined in FAR 2.101) and is proprietary to NetApp, Inc. All NetApp technical data and computer software provided under this Agreement is commercial in nature and developed solely at private expense. The U.S. Government has a non-exclusive, non-transferrable, nonsublicensable, worldwide, limited irrevocable license to use the Data only in connection with and in support of the U.S. Government contract under which the Data was delivered. Except as provided herein, the Data may not be used, disclosed, reproduced, modified, performed, or displayed without the prior written approval of NetApp, Inc. United States Government license rights for the Department of Defense are limited to those rights identified in DFARS clause 252.227-7015(b) (FEB 2014).

Trademark information

NETAPP, the NETAPP logo, and the marks listed at <http://www.netapp.com/TM> are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.