

# Installing Airflow Locally

## Install Script

```
# Airflow needs a home. `~/airflow` is the default, but you can put it
# somewhere else if you prefer (optional)
export AIRFLOW_HOME=~/airflow

# Install Airflow using the constraints file
AIRFLOW_VERSION=2.5.3
PYTHON_VERSION="$(python --version | cut -d " " -f 2 | cut -d "." -f 1-2)"
# For example: 3.7
CONSTRAINT_URL="https://raw.githubusercontent.com/apache/airflow/constraints-
${AIRFLOW_VERSION}/constraints-${PYTHON_VERSION}.txt"
# For example: https://raw.githubusercontent.com/apache/airflow/constraints-2.5.3/constraints-
3.7.txt
pip install "apache-airflow==${AIRFLOW_VERSION}" --constraint "${CONSTRAINT_URL}"

# The Standalone command will initialise the database, make a user,
# and start all components for you.
airflow db init

# Visit localhost:8080 in the browser and use the admin account details
# shown on the terminal to login.
# Enable the example_bash_operator DAG in the home page
```

## Config File

There is a file named `airflow.cfg` which contains configuration for your airflow instance including the full path to the DAGs folder and also SQLAlchemy connection credentials.

## Google Cloud Compatibility

Install airflow with `google` optional module: `pip install apache-airflow[google]`.

Credentials can be side-loaded in via an environment variable

```
export AIRFLOW_CONN_GOOGLE_CLOUD_DEFAULT='{ "conn_type":  
"google-cloud-platform", "key_path": "/secrets/key.json", "scope":  
"https://www.googleapis.com/auth/cloud-platform", "project": "airflow",  
"num_retries": 5}'
```

# Google OIDC auth

It's possible to authenticate airflow against your google org using SAML/OpenID flow:

<https://airflow.apache.org/docs/apache-airflow-providers-google/stable/api-auth-backend/google-openid.html>

---

Revision #2

Created 9 July 2023 15:42:18 by James

Updated 9 July 2023 15:44:00 by James