ASTRONOMER

Upgrading to Apache Airflow 2

Airflow Summit
13 July 2021

Kaxil Naik
Airflow Committer and PMC Member
OSS Airflow Team @ Astronomer

Who am I?

- Airflow Committer & PMC Member
- Manager of Airflow Engineering team @ Astronomer
  - Work full-time on Airflow
- Previously worked at DataReply
- Masters in Data Science & Analytics from Royal Holloway, University of London
- Twitter: https://twitter.com/kaxil
- Github: https://github.com/kaxil/
- LinkedIn: https://www.linkedin.com/in/kaxil/

Agenda

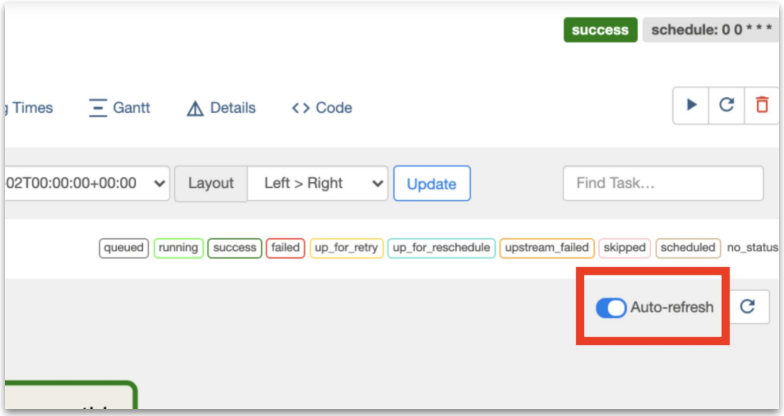
- Why Upgrade?
- Pre-requisites
- `upgrade_check` CLI tool
- Major changes
- Upgrade to 2.x
- Recommendations

http://gph.is/1VBGIPv

Why Upgrade?

Why Upgrade?

- Airflow 1.10.x has reached EOL on **17th June 2021**
- No security patches will be backported
- Airflow 2+ contains
  - tons of performance improvements
  - loads of new features



Upgrade to Python 3

Upgrade to Python 3

- Python 2 reached EOL on **1st January 2020**
- Airflow 2+ requires Python 3.6+
- Officially supported Python versions: 3.6, 3.7 and 3.8
- Python 3.9 will be supported from Airflow 2.1.2

Upgrade to Airflow 1.10.15

Upgrade to Airflow 1.10.15

- Final release in 1.x series
- Many 2.0+ changes backported for cross-compatibility
  - CLI refactor: `airflow trigger_dag` vs `airflow dags trigger`
  - KubernetesExecutor: pod_template_file
  - Configurations (`airflow.cfg`)
- Allows running `upgrade_check` CLI command
- Easier installation of **Backport Providers**

Airflow Upgrade Check Script

About Upgrade Check Script

- Separate Python package (`apache-airflow-upgrade-check`) - PyPI
- Work only with Airflow 1.10.14 and 1.10.15
- Detects deprecated and incompatible changes in:
  - Configuration (`airflow.cfg`)
  - DAG Files
  - Plugins
  - Metadata DB (mainly Airflow Connections)

Install & Run Upgrade Check Script

- Install the latest version (`1.4.0`):
  - `pip install -U apache-airflow-upgrade-check`

- Run the upgrade check script
  - `airflow upgrade_check`

Upgrade Check Script - Example Output

```
airflow@843d9d804d71:/opt/airflow$ airflow upgrade_check

================================================================================ STATUS ================================================================================

Check for latest versions of apache-airflow and checker.....SUCCESS
Remove airflow.AirflowMacroPlugin class.....SUCCESS
Ensure users are not using custom metaclasses in custom operators.....SUCCESS
Chain between DAG and operator not allowed.....SUCCESS
Connection.conn_type is not nullable.....SUCCESS
Custom Executors now require full path.....SUCCESS
Check versions of PostgreSQL, MySQL, and SQLite to ease upgrade to Airflow 2.0.....SUCCESS
Hooks that run DB functions must inherit from DBApiHook.....SUCCESS
Fernet is enabled by default.....SUCCESS
GCP service account key deprecation.....SUCCESS
Unify hostname_callable option in core section.....SUCCESS
Changes in import paths of hooks, operators, sensors and others.....SUCCESS
Legacy UI is deprecated by default.....FAIL
Logging configuration has been moved to new section.....SUCCESS
Removal of Mesos Executor.....SUCCESS
No additional argument allowed in BaseOperator.....SUCCESS
Rename max_threads to parsing_processes.....SUCCESS
Users must set a kubernetes.pod_template_file value.....SKIPPED
Ensure Users Properly Import conf from Airflow.....SUCCESS
SendGrid email uses old airflow.contrib module.....SUCCESS
Check Spark JDBC Operator default connection name.....SUCCESS
Changes in import path of remote task handlers.....SUCCESS
Connection.conn_id is not unique.....SUCCESS
Use CustomSQLAInterface instead of SQLAInterface for custom data models.....SUCCESS
Found 2 problems.

============================================================================== RECOMMENDATIONS ==============================================================================

Legacy UI is deprecated by default
----------------------------------
Legacy UI is deprecated. FAB RBAC is enabled by default in order to increase security.

Problems:

  1.  rbac in airflow.cfg must be explicitly set empty as RBAC mechanism is enabled by default.

Users must set a kubernetes.pod_template_file value
---------------------------------------------------
Skipped because this rule applies only to environment using KubernetesExecutor.
```

Rules - Upgrade Check Script

```
airflow@843d9d804d71:/opt/airflow$ airflow upgrade_check --list

Rule Name                                 Description
----------------------------------------  ------------------------------------------------------------------------------
VersionCheckRule                          Check for latest versions of apache-airflow and checker
AirflowMacroPluginRemovedRule             Remove airflow.AirflowMacroPlugin class
BaseOperatorMetaclassRule                 Ensure users are not using custom metaclasses in custom operators
ChainBetweenDAGAndOperatorNotAllowedRule  Chain between DAG and operator not allowed.
ConnTypeIsNotNullableRule                 Connection.conn_type is not nullable
CustomExecutorsRequireFullPathRule        Custom Executors now require full path
DatabaseVersionCheckRule                  Check versions of PostgreSQL, MySQL, and SQLite to ease upgrade to Airflow 2.0
DbApiRule                                 Hooks that run DB functions must inherit from DBApiHook
FernetEnabledRule                         Fernet is enabled by default
GCPServiceAccountKeyRule                  GCP service account key deprecation
HostnameCallable                          Unify hostname_callable option in core section
ImportChangesRule                         Changes in import paths of hooks, operators, sensors and others
LegacyUIDeprecated                        Legacy UI is deprecated by default
LoggingConfigurationRule                  Logging configuration has been moved to new section
MesosExecutorRemovedRule                  Removal of Mesos Executor
NoAdditionalArgsInOperatorsRule           No additional argument allowed in BaseOperator.
ParsingProcessesConfigurationRule         Rename max_threads to parsing_processes
PodTemplateFileRule                       Users must set a kubernetes.pod_template_file value
ProperlyImportConfFromAirflow             Ensure Users Properly Import conf from Airflow
SendGridEmailerMovedRule                  SendGrid email uses old airflow.contrib module
SparkJDBCOperatorConnIdRule               Check Spark JDBC Operator default connection name
TaskHandlersMovedRule                     Changes in import path of remote task handlers
UniqueConnIdRule                          Connection.conn_id is not unique
UseCustomSQLAInterfaceClassRule           Use CustomSQLAInterface instead of SQLAInterface for custom data models.
```

Apply Recommendations - Upgrade Check Script

- Apply recommendations, example enable RBAC UI:
  - `rbac = True` in `[webserver]` section in `airflow.cfg`
- Fix and run until all checks pass
- Ignore certain rules if they are false positives:
  - `airflow upgrade_check --ignore DbApiRule`

DAG File Changes

DAG File Changes - Backport Providers

- In 2.0+ - operators, hooks, sensors are grouped into logical providers
- Most of these providers are “backported” to run in 1.10.x:
    - 66 Backport Providers - link
- **NOTE**: Backport Providers should only be used for 1.10.14 & 1.10.15. Use actual providers for 2.0+.

DAG File Changes - Backport Providers

apache-airflow-backport-providers-imap 2021.3.17 — Mar 18, 2021
Backport provider package apache-airflow-backport-providers-imap for Apache Airflow

apache-airflow-backport-providers-snowflake 2021.3.13 — Mar 13, 2021
Backport provider package apache-airflow-backport-providers-snowflake for Apache Airflow

apache-airflow-backport-providers-discord 2021.3.17 — Mar 18, 2021
Backport provider package apache-airflow-backport-providers-discord for Apache Airflow

apache-airflow-backport-providers-oracle 2021.3.17 — Mar 18, 2021
Backport provider package apache-airflow-backport-providers-oracle for Apache Airflow

apache-airflow-backport-providers-jira 2021.3.17 — Mar 18, 2021
Backport provider package apache-airflow-backport-providers-jira for Apache Airflow

apache-airflow-backport-providers-telegram 2021.3.3 — Mar 7, 2021
Backport provider package apache-airflow-backport-providers-telegram for Apache Airflow

DAG File Changes - Backport Providers

- Command to Install:
  - 1.10.15: `pip install apache-airflow-backport-providers-docker`
  - 2.0+: `pip install apache-airflow-providers-docker`
- Most of the paths will continue to work but raise a deprecation warning
- Example import change for `DockerOperator`:
  - Before: `from airflow.operators.docker_operator import DockerOperator`
  - After: `from airflow.providers.docker.operators.docker import DockerOperator`

DAG File Changes - KubernetesPodOperator & Executor

- From Airflow 1.10.12, full Kubernetes API is available for `KubernetesExecutor` and `KubernetesPodOperator`.
- Port, VolumeMount, Volume use K8s API instead of objects in `airflow.kubernetes`
- Details: link

DAG File Changes - KubernetesPodOperator & Executor

Before:

```
from airflow.kubernetes.pod import Port
port = Port('http', 80)
k = KubernetesPodOperator(
    namespace='default',
    image="ubuntu:16.04",
    cmds=["bash", "-cx"],
    arguments=["echo 10"],
    ports=[port],
    task_id="task",
)
```

After:

```
from kubernetes.client import models as k8s
port = k8s.V1ContainerPort(name='http', container_port=80)
k = KubernetesPodOperator(
    namespace='default',
    image="ubuntu:16.04",
    cmds=["bash", "-cx"],
    arguments=["echo 10"],
    ports=[port],
    task_id="task",
)
```

More examples and details in : link

Configuration Changes

Configuration Changes - Compatible

- Renamed (1.10.14)
  - `[scheduler] max_threads` to `[scheduler] parsing_processes`
- Grouped & Moved (2.0.0)
  - Logging configs moved from `[core]` to new section `[logging]`
  - Metrics configs moved from `[scheduler]` to new section `[metrics]`
- Backwards compatible changes
- Remove old configs after rename

Configuration Changes - Breaking - New Webserver

- Default Webserver is changed from Flask-Admin to Flask-AppBuilder
  - `[webserver] rbac = False` to `[webserver] rbac = True`
- New UI contains role-based permissions
- No support for Data Profiling, Ad Hoc Query & Charts in new UI
- Auth is required by default.
  - Support for auth via LDAP, Database (user/pass), Open ID, OAuth

Configuration Changes - Breaking - KubernetesExecutor

Many configurations & sections for `KubernetesExecutor` have been removed & replaced by `pod_template_file`

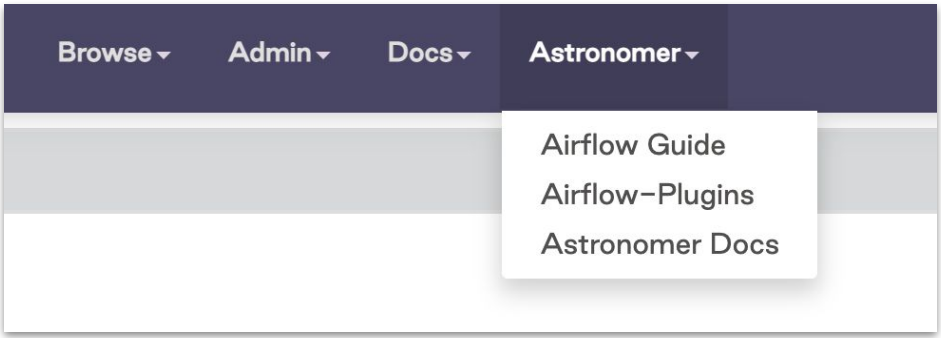
Details: link

```
worker_container_image_pull_policy
airflow_configmap
airflow_local_settings_configmap
dags_in_image
dags_volume_subpath
dags_volume_mount_point
dags_volume_claim
logs_volume_subpath
logs_volume_claim
dags_volume_host
logs_volume_host
env_from_configmap_ref
env_from_secret_ref
git_repo
git_branch
git_sync_depth
git_subpath
git_sync_rev
git_user
git_password
git_sync_root
git_sync_dest
git_dags_folder_mount_point
git_ssh_key_secret_name
git_ssh_known_hosts_configmap_name
git_sync_credentials_secret
git_sync_container_repository
git_sync_container_tag
git_sync_init_container_name
git_sync_run_as_user
worker_service_account_name
image_pull_secrets
gcp_service_account_keys
affinity
tolerations
run_as_user
fs_group
[kubernetes_node_selectors]
[kubernetes_annotations]
[kubernetes_environment_variables]
[kubernetes_secrets]
[kubernetes_labels]
```

Changes to Plugins

Changes to Plugins

- Changes to custom Views and custom Menus for the RBAC UI
  - `admin_views` -> `appbuilder_views`
  - `menu_links` -> `appbuilder_menu_items`



Changes to Plugins

Before

```python
from airflow.plugins_manager import AirflowPlugin

from flask_admin import BaseView, expose
from flask_admin.base import MenuLink


class TestView(BaseView):
    @expose('/')
    def test(self):
        # in this example, put your test_plugin/test.html template at airflow/plugins/templates/test_plugin/test.html
        return self.render("test_plugin/test.html", content="Hello galaxy!")

v = TestView(category="Test Plugin", name="Test View")

ml = MenuLink(
    category='Test Plugin',
    name='Test Menu Link',
    url='https://airflow.apache.org/'
)

class AirflowTestPlugin(AirflowPlugin):
    admin_views = [v]
    menu_links = [ml]
```

After

```python
from airflow.plugins_manager import AirflowPlugin
from flask_appbuilder import expose, BaseView as AppBuilderBaseView

class TestAppBuilderBaseView(AppBuilderBaseView):
    default_view = "test"

    @expose("/")
    def test(self):
        return self.render_template("test_plugin/test.html", content="Hello galaxy!")

v_appbuilder_view = TestAppBuilderBaseView()
v_appbuilder_package = {"name": "Test View",
                        "category": "Test Plugin",
                        "view": v_appbuilder_view}

# Creating a flask appbuilder Menu Item
appbuilder_mitem = {"name": "Google",
                    "category": "Search",
                    "category_icon": "fa-th",
                    "href": "https://www.google.com"}


# Defining the plugin class
class AirflowTestPlugin(AirflowPlugin):
    name = "test_plugin"
    appbuilder_views = [v_appbuilder_package]
    appbuilder_menu_items = [appbuilder_mitem]
```

Changes to Plugins

- Adding Operators, Hooks and Sensors via plugins is no longer supported
- Use normal python modules. Check Modules Management for details
- Move files with custom operators, hooks or sensors to dirs in `PYTHONPATH`
- Import changes:
  - Before: `from airflow.operators.custom_mod import MyOperator`
  - After: `from custom_mod import MyOperator`

Changes to Automation Scripts

Changes to Automation Scripts - CLI

- Update CLI commands
- Full list: link
- Works with 1.10.14+

Old command	New command	Group
airflow worker	airflow celery worker	celery
airflow flower	airflow celery flower	celery
airflow trigger_dag	airflow dags trigger	dags
airflow delete_dag	airflow dags delete	dags
airflow show_dag	airflow dags show	dags
airflow list_dags	airflow dags list	dags
airflow dag_status	airflow dags status	dags
airflow backfill	airflow dags backfill	dags
airflow list_dag_runs	airflow dags list-runs	dags
airflow pause	airflow dags pause	dags
airflow unpause	airflow dags unpause	dags
airflow next_execution	airflow dags next-execution	dags
airflow test	airflow tasks test	tasks
airflow clear	airflow tasks clear	tasks
airflow list_tasks	airflow tasks list	tasks
airflow task_failed_deps	airflow tasks failed-deps	tasks

Changes to Automation Scripts - API

- Experimental API deprecated (but not yet removed)
- Use new Stable REST API after upgrading to 2.0+
- Migration Guide: link

Changes to Automation Scripts - API

Purpose	Experimental REST API Endpoint	Stable REST API Endpoint
Create a DAGRuns(POST)	/api/experimental/dags/<DAG_ID>/dag_runs	/api/v1/dags/{dag_id}/dagRuns
List DAGRuns(GET)	/api/experimental/dags/<DAG_ID>/dag_runs	/api/v1/dags/{dag_id}/dagRuns
Check Health status(GET)	/api/experimental/test	/api/v1/health
Task information(GET)	/api/experimental/dags/<DAG_ID>/tasks/<TASK_ID>	/api/v1//dags/{dag_id}/tasks/{task_id}
TaskInstance public variable(GET)	/api/experimental/dags/<DAG_ID>/dag_runs/<string:execution_date>/tasks/<TASK_ID>	/api/v1/dags/{dag_id}/dagRuns/{dag_run_id}/taskInstances/{task_id}
Pause DAG(PATCH)	/api/experimental/dags/<DAG_ID>/paused/<string:paused>	/api/v1/dags/{dag_id}
Information of paused DAG(GET)	/api/experimental/dags/<DAG_ID>/paused	/api/v1/dags/{dag_id}
Latest DAG Runs(GET)	/api/experimental/latest_runs	/api/v1/dags/{dag_id}/dagRuns
Get all pools(GET)	/api/experimental/pools	/api/v1/pools
Create a pool(POST)	/api/experimental/pools	/api/v1/pools
Delete a pool(DELETE)	/api/experimental/pools/<string:name>	/api/v1/pools/{pool_name}
DAG Lineage(GET)	/api/experimental/lineage/<DAG_ID>/<string:execution_date>/	/api/v1/dags/{dag_id}/dagRuns/{dag_run_id}/taskInstances/{task_id}/xcomEntries

Changes to Automation Scripts - Installing “Extras”

- From Airflow 2.0 onwards “extras” are used for
  - Installing optional core dependencies (ldap, rabbitmq, statsd, virtualenv, etc)
  - Installing Providers (amazon, google, spark, hashicorp, etc)
  - Pre-installed Providers: ftp, http*, imap, sqlite
- Latest released provider versions are installed if installing via extra
  - e.g. `pip install -U apache-airflow[google]` currently installs `apache-airflow-providers-google==4.0.0`
- List of available extras: link

Changes to “Extras”

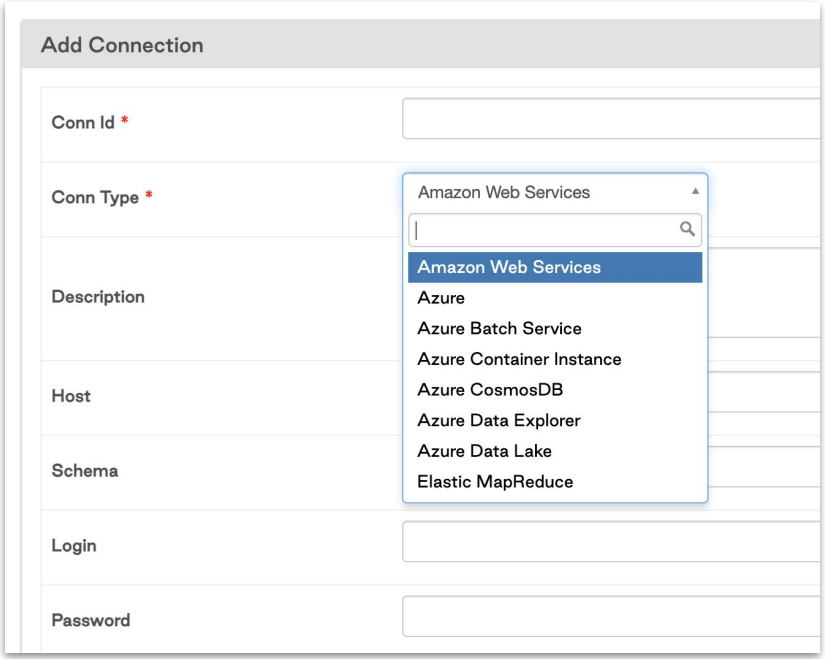
Deprecated extra	Extra to be used instead
atlas	apache.atlas
aws	amazon
azure	microsoft.azure
cassandra	apache.cassandra
crypto	
druid	apache.druid
gcp	google
gcp_api	google
hdfs	apache.hdfs
hive	apache.hive
kubernetes	cncf.kubernetes
mssql	microsoft.mssql
pinot	apache.pinot
qds	qubole
s3	amazon
spark	apache.spark
webhdfs	apache.webhdfs
winrm	microsoft.winrm

extra	install command
async	`pip install 'apache-airflow[async]'`
celery	`pip install 'apache-airflow[celery]'`
cgroups	`pip install 'apache-airflow[cgroups]'`
cncf.kubernetes	`pip install 'apache-airflow[cncf.kubernetes]'`
dask	`pip install 'apache-airflow[dask]'`
deprecated_api	`pip install 'apache-airflow[deprecated_api]'`
github_enterprise	`pip install 'apache-airflow[github_enterprise]'`
google_auth	`pip install 'apache-airflow[google_auth]'`
kerberos	`pip install 'apache-airflow[kerberos]'`
ldap	`pip install 'apache-airflow[ldap]'`
leveldb	`pip install 'apache-airflow[leveldb]'`
password	`pip install 'apache-airflow[password]'`
rabbitmq	`pip install 'apache-airflow[rabbitmq]'`
sentry	`pip install 'apache-airflow[sentry]'`
statsd	`pip install 'apache-airflow[statsd]'`
virtualenv	`pip install 'apache-airflow[virtualenv]'`

Changes to Connections

Changes to Connections - Breaking Change

- Duplicate Connection IDs are not allowed from Airflow 2.0+
- Connection Types are only visible for installed providers



Prune old data in Metadata DB

Prune old data in Metadata DB

- Backup Metadata DB before Airflow version upgrade or pruning
- 19 Database Migrations between 1.10.15 and 2.0.0
- Prune TaskInstance, DagRuns, XComs, Log, TaskReschedule etc tables
- Maintenance DAGs from Clairvoyant

Upgrade to Airflow 2

Upgrade to Airflow 2+

- Pause all the DAGs & make sure no tasks are running
- BackUp Metadata DB, airflow.cfg and Environment Variables
- Stop all the components: Webserver, Scheduler and Workers
- Remove all backport-providers:

```
pip freeze | grep apache-airflow-backport | xargs pip uninstall -y
```

Upgrade to Airflow 2+

- Upgrade to new Airflow version (using constraints file):

```
AIRFLOW_VERSION=2.1.1
PYTHON_VERSION="$(python --version | cut -d " " -f 2 | cut -d "." -f 1-2)"
CONSTRAINT_URL="https://raw.githubusercontent.com/apache/airflow/constraints-${AIRFLOW_VERSION}/constraints-${PYTHON_VERSION}.txt"
pip install --upgrade "apache-airflow[postgres,google]==${AIRFLOW_VERSION}" --constraint "${CONSTRAINT_URL}"
```

  - Install core “extras” like statsd if you were using it previously
  - Install all the providers via extras or directly that are used in DAGs (after testing them !)

    ```
    pip install apache-airflow-providers-google==4.0.0
    ```

  - Providers FAQ: link

Upgrade to Airflow 2+

- Make sure all breaking changes are taken care of:
  - Changes in DAG Files
  - Configuration changes (remove deprecated configs, pod_template_file, etc)
  - Verify Airflow Connections (duplicates are removed, providers are installed)
  - Automation scripts like Terraform if migrating to Stable API
  - Quick glance over UPDATING.md & Updating Guide to verify

Upgrade to Airflow 2+

- Upgrade the Metadata DB
  - `airflow db upgrade`
  - Can take up to 10-15 mins if there are 100s of DAGs and DB hasn’t been cleaned
- Start all the Airflow Components

Recommendations

Recommendations

- Use Postgres
- Test upgrade in a dev environment first
- Only add configs to airflow.cfg that you want to override
- Always upgrade to latest patch release: we now follow strict SemVer
- Use constraints file for installation

```
AIRFLOW_VERSION=2.1.1
PYTHON_VERSION="$(python --version | cut -d " " -f 2 | cut -d "." -f 1-2)"
CONSTRAINT_URL="https://raw.githubusercontent.com/apache/airflow/constraints-${AIRFLOW_VERSION}/constraints-${PYTHON_VERSION}.txt"
pip install "apache-airflow[async,postgres,google]==${AIRFLOW_VERSION}" --constraint "${CONSTRAINT_URL}"
```

Links / References

Links

- Airflow
  - Repo: https://github.com/apache/airflow
  - Website: https://airflow.apache.org/
  - Blog: https://airflow.apache.org/blog/
  - Documentation: https://airflow.apache.org/docs/
  - Slack: https://s.apache.org/airflow-slack
  - Twitter: https://twitter.com/apacheairflow
- Contact Me:
  - Twitter: https://twitter.com/kaxil
  - Github: https://github.com/kaxil/
  - LinkedIn: https://www.linkedin.com/in/kaxil/

Thank You!