


AIP-33 Secrets backend

Status

State	Completed
Discussion Thread	[DISCUSS] AIP-33 secrets backend
Vote Thread	[VOTE] AIP-33 Secrets backend
JIRA	<div> AIRFLOW-5705 - Jira project doesn't exist or you don't have permission to view it.</div>
Pull request	<ul style="list-style-type: none">• https://github.com/apache/airflow/pull/6376• https://github.com/apache/airflow/pull/7742• https://github.com/apache/airflow/pull/7746• https://github.com/apache/airflow/pull/7745• https://github.com/apache/airflow/pull/7741• https://github.com/apache/airflow/pull/7753
Created	<code>\$action.dateFormatter.formatGivenString("yyyy-MM-dd", \$content.getCreationDate())</code>
In Release	1.10.10

Motivation

Currently we can store connections in metastore or env variables.

As an airflow cluster maintainer, I want to be able to store connections in other sources e.g. AWS SSM parameter store. This would give me more flexibility in the ways that I can manage creds.

Considerations

Is there anything special to consider about this AIP? Downsides? Difficulty in implementation or rollout etc?

Originally there was some uncertainty about whether to call it secrets or creds or something else. In discussions on dev list, slack, and the PR, we have coalesced around "secrets".

What change do you propose to make?

- Add a "secrets backend" base class that provides a standard way for retrieving secrets and constructing connection objects.
- Refactor existing `get_connection` functionality in `BaseHook` to be an implementation of this base class
 - we create `EnvironmentVariablesSecretsBackend` and a `MetastoreSecretsBackend`
- As a first alternative secrets backend implementation, add an `AwsSsmSecretsBackend`, which can retrieve creds from SSM Parameter Store.
- Search path is `alternative > metastore > env var` if `alternative secrets backend` is enabled; `env var > metastore` otherwise. Search path cannot be otherwise configured.

What problem does it solve?

Currently there are only two options for sourcing creds — environment variables and metastore database. This change gives users flexibility to easily use other sources.

If your creds are already stored elsewhere, or you would prefer that they are stored elsewhere, then there would be no need to worry about loading them into the metastore, or into env vars. This in itself is convenient. But it opens up some new possibilities.

For example, this would make it possible for a dev team to share one single source for creds instead of having to distribute creds to all developers in a text file. This would make it possible to ensure that all devs are always in sync, using the same creds definitions.

Another use case is for a platform like astronomer cloud, a user might not have access to the airflow CLI. In this case, there's no convenient way to get connections loaded into the metastore. So it would be easier to store them elsewhere.

Another use case is spinning up dev or QA instances. Your dev or QA instances could source creds from a place like SSM parameter store so you don't have to worry about loading them when you create an instance.

Why is it needed?

I wouldn't say this is *necessary*, but more that it is *helpful*, and for reasons discussed above.

Are there any downsides to this change?

I don't think so.

Which users are affected by the change?

This is a refactor that adds an abstraction that is used optionally. It does not change current behavior in any way without active user action.

How are users affected by the change? (e.g. DB upgrade required?)

They are not.

Other considerations?

There are a few implementation details that I had some uncertainty about. For one, I was uncertain about the name, but the community coalesced around "Secrets Backend".

Additionally, initially I wasn't sure about whether the classes should use only class methods, or if instance methods make sense. Ultimately, I decided instance methods made more sense; config params are passed to `__init__`, and the backend is instantiated on initial import.

Proposed amendments

Add a `get_secret` method (deferred)

Jarek has suggested considering extending this to support arbitrary secrets and not just connections. Ultimately we deferred consideration of this. I think we need to think about how we intend for that kind of method to be used. Currently the general convention is to retrieve creds through connection objects. And we have the Variable model for things that are not "secrets".

Remove search path configurability (*accepted*)

Kaxil suggested removing search path configurability. The proposal is as follows:

- only configurability is to specify a single alternative secrets backend (the default is none)
- if specified, the specified alternative secrets backend is moved to front of search path (with env vars and database searched only if none found in the "alternative" backend)

This would simplify the config because you could just have two items: ``backend`` (the alternative backend class) and ``backend_kwargs``, which would be any config info needed by the backend.

In initial PR since multiple backends can be used simultaneously, we needed to provide a way for them to be configured independently, which means more detail has to go into the config.

Change name to secrets from creds (*accepted*)

This was completed

What defines this AIP as "done"?

The PR is merged.