
datacoco-secretsmanager

Release 0.1.0

Mar 18, 2020

Contents

1	Installation	3
2	Quickstart	5
2.1	One Secret	5
2.2	Many Secrets	6
3	Development	9
3.1	Getting Started	9
3.2	Testing	9
3.3	Contributing	9

datacoco-secretsmanager provides basic interaction with the Amazon Web Service (AWS) Secrets Manager service.

CHAPTER 1

Installation

datacoco-secretsmanager requires Python 3.6+

```
python3 -m venv venv
source venv/bin/activate
python -m pip install datacoco_secretsmanager
```


CHAPTER 2

Quickstart

datacoco-secretsmanager utilizes the [boto3](#) library to interact with the AWS Secrets Manager service, requiring AWS credentials configuration. Lookup of credentials by boto3 is documented [here](#).

Based on how you store your AWS credentials, you can use datacoco-secretsmanager in the following ways.

If you have AWS credentials stored in the default `~/.aws/credentials`, instantiate a `SecretsManager` class using:

```
from datacoco_secretsmanager import SecretsManager

sm = SecretsManager()
```

You can also pass in AWS authentication keys directly:

```
from datacoco_secretsmanager import SecretsManager

sm = SecretsManager(
    aws_access_key_id,
    aws_secret_access_key,
    aws_role_arn, # only required if you are using role based access
)
```

Otherwise, if you are running on an Amazon EC2 instance, and credentials are not passed in either way above, you can have boto3 load credentials from the instance metadata service. datacoco-secretsmanager will then assume the same IAM role as you specified when you launched your EC2 instance.

2.1 One Secret

Store a secret in AWS Secrets manager:

AWS Secret name

```
<AWS-secret-name-for-connection>
```

Key	Value
<db-name>	<db-name>
<user>	<user-name>
<host>	<host>
<port>	<port-value>
...	...

To fetch a single secret, use:

```
sm.get_secret(<aws_secret_resource_name>)
```

2.2 Many Secrets

For a project, you may have more than one secret or credentials for more than one system.

You can handle by storing key/value mapping for all required credentials in an AWS secret for the project, then further store credentials in a separate AWS secret for each credential name indicated in a key's value.

For example, storing a single AWS secret to map or provide lookup to all required system/db connections is known as the `cfg_store` name in our module:

AWS Secret name

```
<project-name>/<environment>
```

Note: If using environment, environment variable named `ENVIRONMENT` should be stored and assigned with the same environment name indicated in your AWS secret name.

Additionally, if working in organization with multiple teams using AWS Secrets Manager, you can further denote secrets per team, by using naming convention:

```
<team-name>/<project-name>/<environment>.
```

Store key/values for your `cfg_store` with the following:

Key	Value
<db-connection1-name>	<AWS-secret-name- for -db-connection1>
<db-connection2-name>	<AWS-secret-name- for -db-connection2>

For each Secret value in your `cfg_store`, store the full credentials in an additional AWS Secret, ie:

AWS Secret name

```
<AWS-secret-name-for-db-connection1>
```

Key	Value
<db-name1>	<db-name1>
<user>	<user-name>
<host>	<host>
<port>	<port-value>
...	...

AWS Secret name

<AWS-secret-name-**for**-db-connection2>

Key	Value
<db-name2>	<db-name2>
<user>	<user-name>
<host>	<host>
<port>	<port-value>
...	...

To fetch secrets for a full project/cfg store, use:

```
sm.get_config(
    project_name='your-project-name',
    team_name='your-team-name',      # include only if you want to save as part of
    ↪ your cfg_store name
)
```


3.1 Getting Started

It is recommended to use the steps below to set up a virtual environment for development:

```
python3 -m venv <virtual env name>
source <virtual env name>/bin/activate
pip install -r requirements.txt
```

3.2 Testing

```
pip install -r requirements-dev.txt
```

To run the testing suite, simply run the command: `tox` or `python -m unittest discover tests`

3.3 Contributing

Contributions to `datacoco_secretsmanager` are welcome!

Please reference guidelines to help with setting up your development environment [here](#).