

IaC Confluent API Key Rotation Terraform module

This Terraform [module](#) is designed to manage the creation and rotation of Confluent [API Keys](#). The key rotation is triggered based on the number of days since the key's creation, ensuring that keys are regularly updated for enhanced security. You can also configure the module to retain a specific number of API Keys per [Service Account](#), giving you flexibility in how keys are managed.

A Terraform module is essentially a collection of [input](#) and [output](#) variables, resources, and configuration files that encapsulate specific functionality. By defining input variables, you can customize the module's behavior without altering its source code, making it adaptable to various use cases. Output variables provide information that can be used by other modules or configurations. This modular approach not only promotes reusability and composability but also simplifies the sharing of standardized configurations across different Terraform setups, enabling more efficient and consistent infrastructure management.

Table of Contents

- [1.0 Let's get started!](#)
 - [1.1 Visualizing the Terraform Configuration](#)
- [2.0 Resources](#)

1.0 Let's get started!

Important Notice

To ensure seamless operation and avoid potential disruptions when using Terraform with time-based rotation, it's crucial to regularly execute the module within the specified rotation period. Specifically, the execution frequency should match or exceed the configured rotation interval for API key. Failing to adhere to this schedule risks the deletion of multiple key pairs in a single execution cycle. Such an occurrence could potentially remove all active API keys, thereby disrupting any processes that rely on older keys for accessing Snowflake resources. To maintain uninterrupted access and functionality, it is imperative to keep the module execution timely and consistent with the rotation settings.

These are the steps

1. Take care of the cloud environment prerequisites listed below:

You need to have the following cloud accounts:

- [Confluent Account](#)
- [GitHub Account](#) *with OIDC configured for AWS*
- [Terraform Cloud Account](#)

2. Clone the repo:

```
git clone https://github.com/j3-signalroom/iac-snowflake-user-rsa_key_pairs_rotation-tf_module.git
```

3. Update the cloned Terraform module's [main.tf](#) by following these steps:
 - a. Locate the `terraform.cloud` block and replace `signalroom` with your [Terraform Cloud Organization Name](#).
 - b. In the `terraform.cloud.workspaces` block, replace `iac-confluent-api-key-rotation-workspace` with your [Terraform Cloud Organization's Workspaces Name](#).
4. Deploy your Terraform module to GitHub by following these steps:

- a. **Commit your module:** Ensure all changes to your Terraform module are committed to your local Git repository.

- b. **Push to GitHub:** Push your committed changes to your GitHub repository. This makes the module available for use in other projects.

- c. **Add a Module Block:** In the Terraform configuration where you want to use the module, add a module block. Inside this block, include the following:

Source: Reference the GitHub repository URL where your module is stored.

Version: Specify the appropriate branch, tag, or commit hash to ensure you're using the correct version of the module.

- d. **Pass Input Variables:** Within the same module block, pass the required input variables by defining them as key-value pairs:

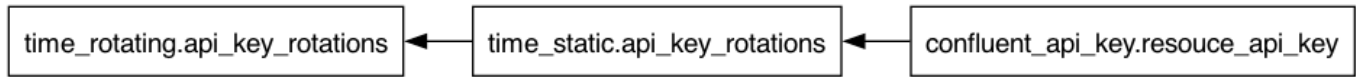
Input Variable	Variable Required	Description
<code>confluent_api_key</code>	Yes	Specifies the Confluent API Key (also referred as Cloud API ID)
<code>confluent_api_secret</code>	Yes	Specifies the Confluent API Secret
<code>day_count</code>	No	[Defaults to 30 days] Specifies how many day(s) should the API Key be rotated for
<code>number_of_api_keys_to_retain</code>	No	[Defaults to 2 API Keys] Specifies the number of API Keys to retain
<code>key_display_name</code>	No	[Defaults to a display name with current date] Specifies the name of the human-readable name for the API Key
<code>owner</code>	Yes	Specifies the API Key Owner. Refer to Confluent API Key Docs for more info
<code>resource</code>	Yes	Specifies the API Key Resource associated with it. Refer to Confluent API Key Docs for more details

- e. **Output Variables:**

Output Variable	Description
<code>active_api_key</code>	Specifies the current active API Key to be used for new logins. Refer to confluent/confluent_api_key for the expected structure
<code>all_api_keys</code>	Specifies all API Keys sorted by creation date. With the current active API Key being the 1st in the collection

1.1 Visualizing the Terraform Configuration

Below is the Terraform visualization of the Terraform configuration. It shows the resources and their dependencies, making the infrastructure setup easier to understand.



To fully view the image, open it in another tab on your browser to zoom in.

When you update the Terraform Configuration, to update the Terraform visualization, use the `terraform graph` command with `Graphviz` to generate a visual representation of the resources and their dependencies. To do this, run the following command:

```
terraform graph | dot -Tpng > .blog/images/terraform-visualization.png
```

2.0 Resources

- [Best Practices for Using API Keys on Confluent Cloud](#)
- [Nikoleta Verbeck shows how to use Terraform to rotate Confluent API Key\(s\)](#)
- [Terraform Resource `time_rotating`](#)
- [Terraform Hidden Gems! Secret Rotation with `time_rotating`](#)