Set up Terraform Cloud for local use

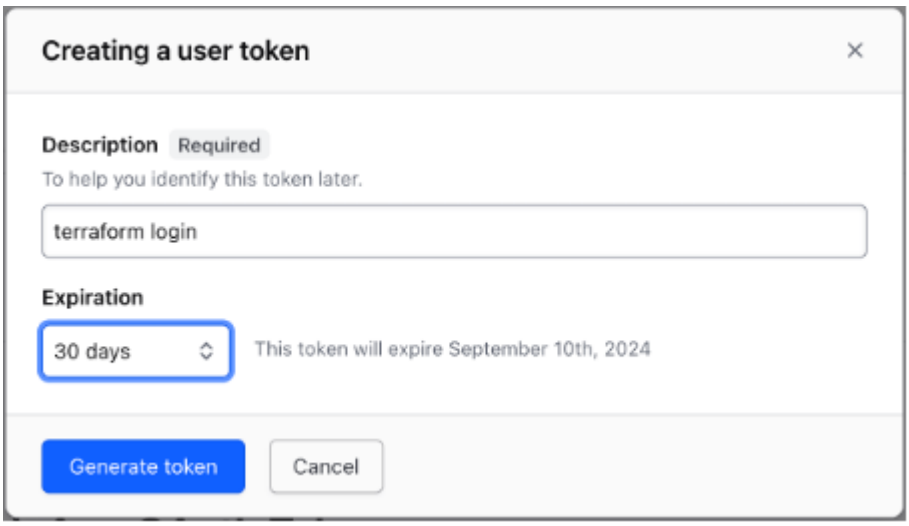
Table of Contents

1.0 Get Terraform Cloud API token

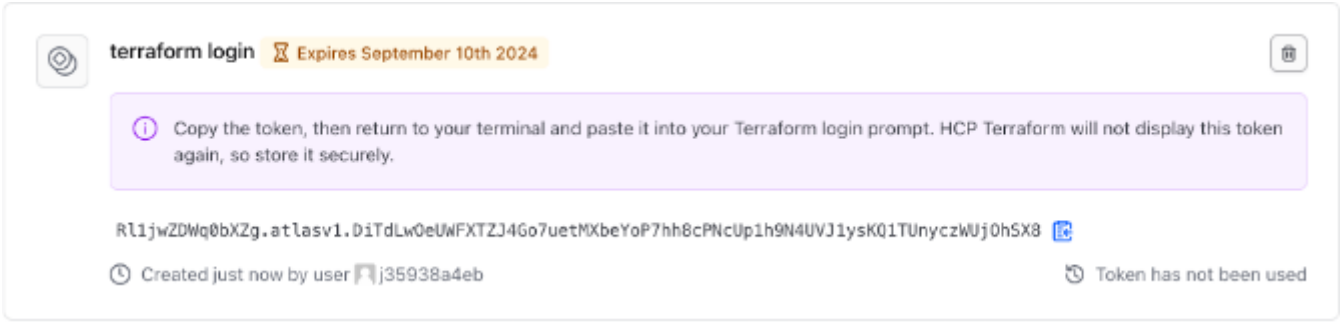
In order to authenticate with HCP Terraform, run the `terraform login` CLI command. Enter `yes` to the prompt to confirm that you want to authenticate.

```
(base) jeffreyjonathanjennings@J3s-MacBook-Pro apache_flink-kickstarter % terraform login
Terraform will request an API token for app.terraform.io using your browser.

If login is successful, Terraform will store the token in plain text in
the following file for use by subsequent commands:
    /Users/jeffreyjonathanjennings/.terraform.d/credentials.tfrc.json

Do you want to proceed?
  Only 'yes' will be accepted to confirm.

  Enter a value: yes


---------------------------------------------------------------------------

Terraform must now open a web browser to the tokens page for app.terraform.io.

If a browser does not open this automatically, open the following URL to proceed:
    https://app.terraform.io/app/settings/tokens?source=terraform-login


---------------------------------------------------------------------------

Generate a token using your browser, and copy-paste it into this prompt.
```

Generate token A browser window will automatically open to the HCP Terraform login screen. Enter a token name in the web UI, or leave the default name, `terraform login`:



Click Create API token to generate the authentication token:



Save a copy of the token in a secure location. It provides access to your HCP Terraform organization. Terraform will also store your token locally at the file path specified in the command output (see the picture above).

Add the token to the CLI prompt When the Terraform CLI prompts you, paste the user token exactly once into your terminal. Terraform will hide the token for security when you paste it into your terminal. Press `Enter` to complete the authentication process:

1.1 Set up your Confluent Cloud environment

Then in the main.tf replace `<TERRAFORM CLOUD ORGANIZATION NAME>` in the `terraform.cloud` block with your Terraform Cloud Organization Name and `<TERRAFORM CLOUD ORGANIZATION's WORKSPACE NAME>` in the `terraform.cloud.workspaces` block with your Terraform Cloud Organization's Workspaces Name.

> **Confluent Cloud API**

> Confluent Cloud requires API keys to manage access and authentication to different parts of the service. An API key consists of a key and a secret. You can create and manage API keys by using the Confluent Cloud CLI. Learn more about Confluent Cloud API Key access here. Using the Confluent CLI, execute the follow command to generate the Cloud API Key:
>
> ```
> confluent api-key create --resource "cloud"
> ```
>
> Then, copy-and-paste the API Key and API Secret values to the respective, `<CONFLUENT_CLOUD_API_KEY>` and `<CONFLUENT_CLOUD_API_SECRET>` property values in the `terraform.tfvars` file.

The configuration leverages the IaC Confluent Cloud Resource API Key Rotation Terraform module to handle the creation and rotation of each of the Confluent Cloud Resource API Key for each of the Confluent Cloud Resources:

- Schema Registry Cluster
- Kafka Cluster
- Kafka Topics
- Flink Compute Pool

Along with the Schema Registry Cluster REST endpoint, Kafka Cluster's Bootstrap URI, Flink Compute Pool ID, and Principal ID, these are stored in AWS Secrets Manager (click here for a description of the secrets).