Generating the infrastructure on AWS through Terraform

Git commit with doc: df6705d

HTML version

1. Introduction

The following Steps will allow you to create an infrastructure on AWS and access it. There is no need to manually install Terraform. It is executed via Docker using the `terraform.sh` script.

To follow these Steps, ensure you have correctly configured a `./.env` file in terraform's directory.

> 💡 If there is a file `../docker/.env` the `terraform.sh` script will use it.

2. Steps

2.1. Step 1 → Initialize the project

```
$ cd terraform
$ ./terraform.sh init
```

2.2. Step 2 → Create the infrastructure

```
$ ./terraform.sh apply
```

2.3. Step 3 → Accessing the EC2 instance that run the apps in AWS

```
$ ./apps/ssh.sh
```

Inside the EC2 instance, you can control the Docker images in a similar way you do locally:

```
$ cd snowplow-demo

$ ./docker/stats.sh # <- show the statistics for the docker containers
$ ./docker/down.sh # <- stop the docker containers
$ ./docker/up.sh # <- start the docker containers
```

3. Other commands

3.1. Check versions

```
$ ./terraform.sh --version
Terraform v1.10.0
on linux_amd64
+ provider registry.terraform.io/hashicorp/aws v5.79.0
+ provider registry.terraform.io/hashicorp/local v2.5.2
+ provider registry.terraform.io/hashicorp/tls v4.0.6
```

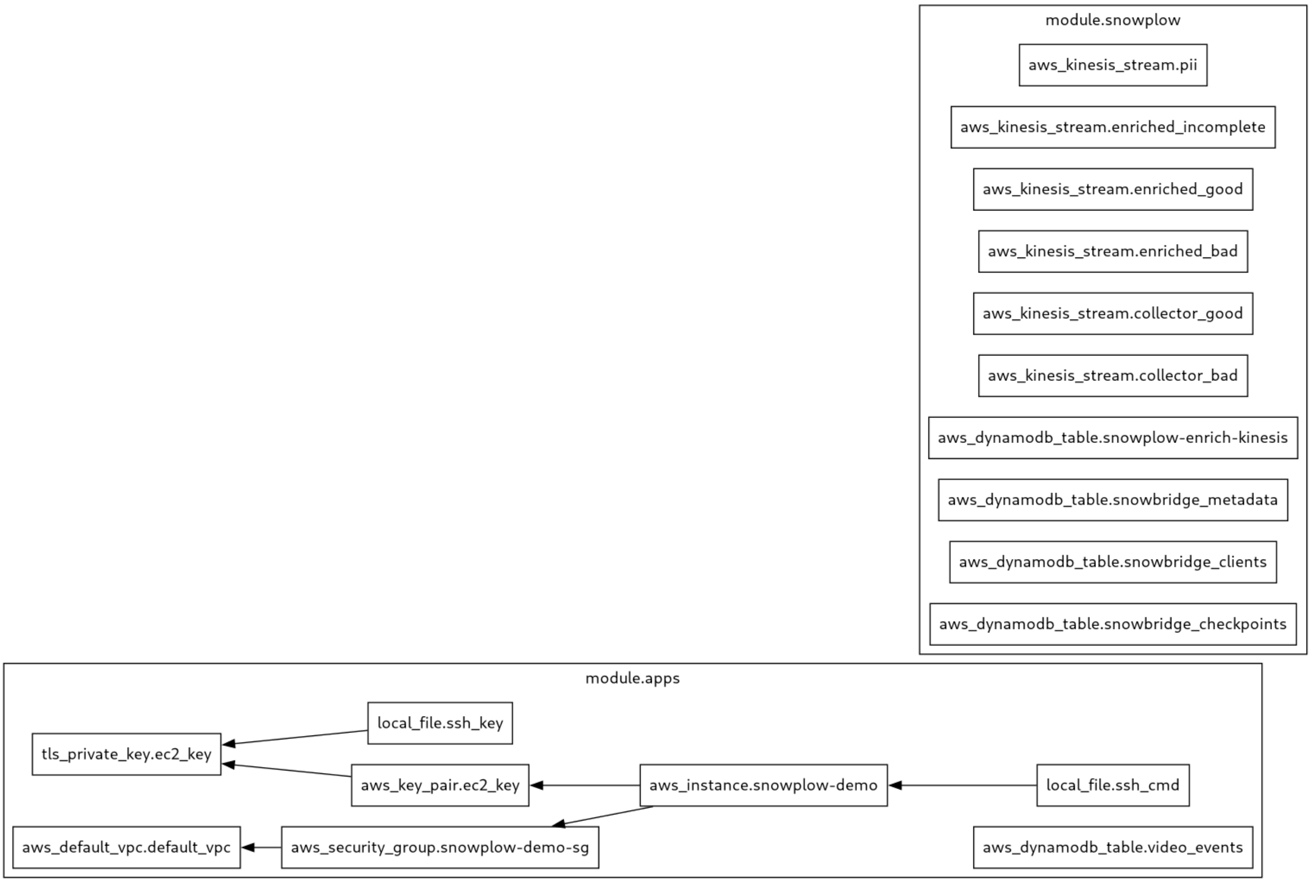
3.2. Check the Terraform plan

```
$ ./terraform.sh plan
```

3.3. Generate a PNG image for the Terraform modules in this project

```
$ ./terraform.sh png
```

Current PNG image of the available modules:


module.snowplow
aws_kinesis_stream.pii
aws_kinesis_stream.enriched_incomplete
aws_kinesis_stream.enriched_good
aws_kinesis_stream.enriched_bad
aws_kinesis_stream.collector_good
aws_kinesis_stream.collector_bad
aws_dynamodb_table.snowplow-enrich-kinesis
aws_dynamodb_table.snowbridge_metadata
aws_dynamodb_table.snowbridge_clients
aws_dynamodb_table.snowbridge_checkpoints
module.apps
local_file.ssh_key
tls_private_key.ec2_key
aws_key_pair.ec2_key
aws_instance.snowplow-demo
local_file.ssh_cmd
aws_default_vpc.default_vpc
aws_security_group.snowplow-demo-sg
aws_dynamodb_table.video_events

3.4. Destroy the infrastructure

```
$ ./terraform.sh destroy
```