Environment Variable - Add Environment Variables to an ECS Service

Overview

Amazon Elastic Container Service (ECS) is a highly scalable, high-performance container management service that supports Docker containers and allows you to easily run applications on a managed cluster of Amazon EC2 instances. Amazon ECS provides a service scheduler (for long-running tasks and applications), the ability to run tasks manually (for batch jobs or single run tasks), with Amazon ECS placing tasks on your cluster for you. In this article, we discuss how to add an environment variable to an ECS service.

Content

Environment variables in ECS Service component:

The ECS service component gives the ability to create 1 or multiple Secrets Manager secrets and have their values automatically injected into the service container at startup. This is done by way of environment variables defined in the service's ECS Task Definition and pointing to their respective Secrets Manager secret.

Variables are used to make this work:

- `secrets`: Map which contains `secret_name` in AWS Secrets Manager and `secret_json` which is the value containing key/value pair that will become the name and value of the environment variables injected into the container at startup.

Each item in the `secrets` map will become a Secrets Manager secret whose name is provided by the map `key` and whose value will be provided by the `value` of map. The map `value` is a JSON-encoded string that will contain a map of all values for that secret.

At runtime, the container will be injected with environment variables named after the `name` field and whose value will be the content of the `value` (or similarly the JSON-encoded version of the corresponding Secrets Manager secret value).

Example

```
locals {
  secret_this_value = {
     this_value = "something"
  }

  secret_that_value = {
     that_value = "something else"
  }
}

module "service" {
  source = "git::https://...../terraform-aws-fargate-service"

  ...

  secrets = {
      "SECRET_THIS": "${jsonencode(local.secret_this_value)}"
      "SECRET_THAT": "${jsonencode(local.secret_that_value)}"
  }
}
```

The container will then receive the following environment variables:

```
SECRET_THIS="{\"this_value\":\"something\"}"
SECRET_THAT="{\"that_value\":\"something else\"}"
```

In NodeJS, the code could simply do something like this to retrieve the content of them:

```
let this_value, that_value;
try{
this_value = JSON.parse(process.env['SECRET_THIS']);
that_value = JSON.parse(process.env['SECRET_THAT']);
} catch(e){
...
}
```

References

Fargate Service Component: https://gitlab.com/mc-components/terraform-aws-fargate-service