Workspaces

- Each Terraform configuration has an associated backend that defines how operations are executed and where persistent data such as the Terraform state are stored.
- The persistent data stored in the backend belongs to a workspace.
- So far, without knowing, we were working with *default* workspace.
- Certain backends support multiple *named workspaces*. The configuration still has only one backend, but multiple distinct instances of that configuration to be deployed without configuring a new backend or changing authentication credentials.
- Multiple workspaces are currently supported by AWS S3 backend.
- In the 0.9 line of Terraform releases, this concept was known as "environment". It was renamed in 0.10 based on feedback about confusion caused by the overloading of the word "environment" both within Terraform itself and within organizations that use Terraform.

Using Workspaces

- Workspaces are managed with the `terraform workspace` set of commands.

Hands-on:

> Let's use terraform workspaces Refer to 06-Workspaces/01-Using-Workspaces First, we'll create resources in default workspace

```
terraform apply
```

> Now we'll create new workspace and create the same resource

```
terraform workspace show

terraform workspace new foo

terraform apply
```

- Key takeaways:
  - terraform show doesn't show any existing resources that exists on default (or any other workspaces)
  - Refer to terraform.tfstate.d directory
  - This tells us that workspace is technically equivalent to renaming your state file
  - The "current workspace" name is stored only locally in the ignored .terraform directory. This allows multiple team members to work on different workspaces concurrently.

Hands-on:

> Let's cleanup

© Copyright Wednesday, Aug 12, 2020, 3:44 PM by Kapil Malhotra

```
terraform workspace select foo

terraform destroy –auto-approve

terraform workspace select default

terraform destroy –auto-approve
```

When to use multiple workspaces

- A common use for multiple workspaces is to create a parallel, distinct copy of a set of infrastructure in order to test a set of changes before modifying the main production infrastructure.
- For example, a sysadmin working on a complex set of infrastructure changes might create a new temporary workspace in order to freely experiment with changes without affecting the default workspace.
- Non-default workspaces are often related to feature branches in version control.
- The default workspace might correspond to the "master" or "trunk" branch, which describes the intended state of production infrastructure.
- Named workspaces are not a suitable isolation mechanism for Dev/Test/Stage/Prod environments. Because, organisations commonly want to create a strong separation between these different environments, which means different AWS accounts and hence different credentials.

© Copyright Wednesday, Aug 12, 2020, 3:44 PM by Kapil Malhotra