Docs » Guides » Import a Resource into Terraform

Import a resource into Terraform

There are some situations you may encounter where it is necessary for Terraform to manage a resource that it did not create itself. In those cases you can define the resource in your Terraform scripts and use the `terraform import` command to "import" the resource into your project's Terraform state. This step only needs to be performed once per account since Terraform will continue to manage the resource after the initial import.

For example, Lambda (and other managed AWS services) will automatically create a log group for your application. You may want to manage this log group yourself in order to modify certain attributes (such as a retention policy) or allow other resources to reference it in your Terraform scripts.

Imagine a Lambda function named `my-test-func` that already has an auto-created log group, named by default `/aws/lambda/my-test-func`. If you wanted to import that resource into your Terraform state, you would first add it to your application's Terraform scripts:

```
resource "aws_cloudwatch_log_group" "logs" {
  name = "/aws/lambda/${aws_lambda_function.lambda.function_name}"
}
```

Then, prior to your next `apply`, you would run the `import` command. The command takes the resource's name as well as an identifier unique to the type of resource being imported.

For a log group, the identifier is simply the log group name so the command we would run in our example is:

```
terraform import aws_cloudwatch_log_group.logs /aws/lambda/my-test-func
```

The suggested way of performing the import is to temporarily update your Octopus Deploy project or Gitlab CI builds to run the import command in a separate step before running your apply. This may require adding additional variables to handle cases where the resource identifier differs between accounts.

An example Octopus Deploy step for our example above could be a script like this:

```
terraform init
terraform import aws_cloudwatch_log_group.logs $(get_octopusvariable "LogGroup")
```

After the change is successfully released to each environment any added steps/variables can be safely removed.