Terraform: Infrastructure as Code

Types of approaches

Types of approaches

- **declarative** (functional): you specify the desired final state of the infrastructure you want to provision and the IaC software handles the rest
- **imperative** (procedural): helps you prepare automation scripts that provision your infrastructure

Tools

Configuration Management Systems Ansible, Chef, Puppet, SaltStack	Infrastructure provisioning Terraform, CloudFormation, Heat
OS Configuration	Infrastructure Automation
Application Installation	VM and Cloud Provisioning
Declarative	Declarative
Limited Infrastructure Automation	Limited OS Configuration Management

Tools

Tool	Approach	Delivery Method
Ansible	Declarative & Imperative	Push (and Pull)*
Chef	Declarative & Imperative	Pull
Puppet	Declarative	Pull
SaltStack	Declarative & Imperative	Push and Pull
Terraform	Declarative	Push

- there is a module ansible-pull

Imperative Approach: create

The imperative focuses on how the infrastructure is to be changed

```
$ aws s3api create-bucket --bucket 7a4a917e-6d15-4995-8e77-addbfaba77c6 --region us-east-2 --create-bucket-configuration LocationConstraint=us-east-2
{
    "Location": "http://7a4a917e-6d15-4995-8e77-addbfaba77c6.s3.amazonaws.com/"
}
```

```
$ aws s3api create-bucket --bucket 7a4a917e-6d15-4995-8e77-addbfaba77c6 --region us-east-2 --create-bucket-configuration LocationConstraint=us-east-2

An error occurred (BucketAlreadyOwnedByYou) when calling the CreateBucket operation: Your previous request to create the named bucket succeeded and you already own it.
$ echo $?
255
```

Imperative Approach: show

```
$ aws s3api list-buckets --region us-east-2
{
    "Buckets": [
        {
            "Name": "7a4a917e-6d15-4995-8e77-addbfaba77c6",
            "CreationDate": "2020-04-30T20:05:56.000Z"
        }
    ],
    "Owner": {
        "ID": "905339dfcf0bb1be6066daadd65c3de1799387cf1d6eeb48581860f51ab59c8d"
    }
}
$ aws s3api list-buckets --region us-east-2 | jq .Buckets[0].Name
"7a4a917e-6d15-4995-8e77-addbfaba77c6"
```

Imperative Approach: delete

```
$ aws s3api delete-bucket --bucket 7a4a917e-6d15-4995-8e77-addbfaba77c6 --region us-east-2
$ aws s3api delete-bucket --bucket 7a4a917e-6d15-4995-8e77-addbfaba77c6 --region us-east-2

An error occurred (NoSuchBucket) when calling the DeleteBucket operation: The specified bucket does not exist
$ echo $?
255
```

Imperative Approach: example

```
output=$($AWS s3api get-bucket-versioning --bucket $dst| jq '(.Status=="Enabled")')
if [[ $output != true ]]
then
        echo "Enabling versioning for $dst"
        $AWS s3api put-bucket-versioning --bucket $dst --versioning-configuration Status=Enabled
        [[ $? -ne 0 ]] && { echo "Can't enable versioning for $dst"; exit 1; }
fi

$AWS s3api get-bucket-encryption --bucket ${dst} &>/dev/null
if [[ $? -ne 0 ]]
then
        if [[ -z ${key_arn} ]]
        then
                cmk_id=$($AWS kms create-key --origin EXTERNAL --region eu-central-1|jq '.KeyMetadata.KeyId'|tr -d \")
                [[ $? -ne 0 ]] && { echo "Can't create key"; exit 1; }
                key_arn="arn:aws:kms:eu-central-1:${id}:key/${cmk_id}"
                $AWS kms get-parameters-for-import --key-id ${cmk_id} \
                        --wrapping-algorithm RSAES_OAEP_SHA_1 \
                        --wrapping-key-spec RSA_2048 --region eu-central-1 >/tmp/get-parameters-for-import
                [[ $? -ne 0 ]] && { echo "Can't download key"; exit 1; }
                openssl enc -d -base64 -A -in PublicKey.b64 -out PublicKey.bin
```

Declarative Approach

The declarative approach focuses on what the eventual target configuration should be:

```
resource "aws_s3_bucket" "main" {
  bucket = "7a4a917e-6d15-4995-8e77-addbfaba77c6"
}
```

Declarative Approach: create

```
$ terraform apply -auto-approve
aws_s3_bucket.main: Creating...
aws_s3_bucket.main: Still creating... [10s elapsed]
aws_s3_bucket.main: Creation complete after 12s [id=7a4a917e-6d15-4995-8e77-addbfaba77c6]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.

Outputs:

arn = arn:aws:s3:::7a4a917e-6d15-4995-8e77-addbfaba77c6
$ terraform apply -auto-approve
aws_s3_bucket.main: Refreshing state... [id=7a4a917e-6d15-4995-8e77-addbfaba77c6]

Apply complete! Resources: 0 added, 0 changed, 0 destroyed.

Outputs:

arn = arn:aws:s3:::7a4a917e-6d15-4995-8e77-addbfaba77c6
```

Declarative Approach: destroy

```
$ terraform destroy -auto-approve
aws_s3_bucket.main: Refreshing state... [id=7a4a917e-6d15-4995-8e77-addbfaba77c6]
aws_s3_bucket.main: Destroying... [id=7a4a917e-6d15-4995-8e77-addbfaba77c6]
aws_s3_bucket.main: Destruction complete after 1s

Destroy complete! Resources: 1 destroyed.
$ terraform destroy -auto-approve


Destroy complete! Resources: 0 destroyed.
```