Hands-on Infrastructure Automation with Terraform on AWS

Alexander Savchuk

Managing multiple environments

Section 6

Manipulating state

Useful commands to manipulate Terraform state

- `state` - move resources around
- `import` - import existing infrastructure
- `taint` - force re-creation of a resource

Useful commands to manipulate Terraform state

- **`state`** - move resources around
- `import` - import existing infrastructure
- `taint` - force re-creation of a resource

Useful commands to manipulate Terraform state

- `state` - move resources around
- **`import`** - import existing infrastructure
- `taint` - force re-creation of a resource

Useful commands to manipulate Terraform state

- `state` - move resources around
- `import` - import existing infrastructure
- **`taint`** - force re-creation of a resource

terraform state

```
terraform state list

terraform state mv [options] SOURCE DESTINATION

terraform state pull

terraform state push

terraform state rm

terraform state show
```

terraform import

```
terraform import aws_vpc.main vpc-07906df7b9a2bb9de
```

Step 1: add a resource block

```
resource "aws_vpc" "main" {}
```

Step 2: import the resource

```
terraform import aws_vpc.main vpc-07906df7b9a2bb9de
```

Step 3: align the desired configuration with actual state

```
terraform plan
```

Can only import resources one at a time

terraform taint

```
terraform taint aws_vpc.main
```

Understand state well and know how to manipulate it

Working with Terraform as a team

Next Section