

Hands-on Infrastructure Automation with Terraform on AWS

Alexander Savchuk

Managing multiple environments

Section 6

Storing environment
configuration

Keep environments
segregated

Share common
configuration and
avoid code
duplication

Current workspace interpolation

```
${terraform.workspace}
```

Local value configuration

```
locals {  
    name = "${terraform.workspace}-workspace"  
}
```

Some configuration blocks do
not support interpolation

Partial backend configuration

```
terraform {  
  backend "s3" {}  
}
```

```
terraform init -backend-config=backend.tfvars
```


Partial backend configuration

- Can be set interactively, using command line key-value pairs, or a configuration file
- You can provide multiple configuration files, in which case the values are merged

Named variable files

Files matching `terraform.tfvars` or `*.auto.tfvars` are loaded automatically.

Can also load variables from the named files:

```
terraform apply -var-file=prod.tfvars
```

Reduces duplicate
code

Requires complex
command
invocations

Summary

Workspaces:

- Great for switching between multiple instances of a single configuration within its single backend
- Can work for isolated environments if you use partial backend configuration

Summary

Workspaces:

- Great for switching between multiple instances of a single configuration within its single backend
- Can work for isolated environments if you use partial backend configuration

Subfolders

- More options to control backend and module configuration
- Lead to either code duplication or having to use wrappers

Manipulating state

Next video