Hands-on Infrastructure Automation with Terraform on AWS

Alexander Savchuk

Working with Terraform as a team

Section 7

Running Terraform in automation for CI/CD

Terraform development workflow

1. Make changes and test locally
2. Commit , push to the upstream repository, and open a pull request
3. Run a plan for the changes on CI/CD
4. Post the plan output on the pull request for review
5. Approve pull request
6. Run apply and notify about the results

1. Make changes and test locally

```
terraform workspace new <feature-branch>
terraform plan
terraform apply
terraform destroy
```

2. Open pull request

3. Run a plan on CI/CD

3. Run a plan on CI/CD (pre-requisites)

Install Terraform or use Docker to run.

3. Run a plan on CI/CD (setup)

```
export TF_IN_AUTOMATION=”yes-please”
```

3. Run a plan on CI/CD (setup contd)

```
export TF_IN_AUTOMATION=”yes-please”
rm -rf .terraform
```

3. Run a plan on CI/CD (setup contd)

```
export TF_IN_AUTOMATION=”yes-please”
rm -rf .terraform

[export TF_PLUGIN_CACHE_DIR="$HOME/.terraform.d/plugin-cache"]
terraform init -input=false
-plugin-dir=/usr/lib/custom-terraform-plugins
[-backend-config=backend.tfvars]
```

3. Run a plan on CI/CD (plan)

```
terraform plan -input=false -out=tfplan [-var-file=test.tfvars]
```

4. Post the plan output on pull request

5. Approve pull request

6. Run apply

```
terraform init -input=false
terraform apply -input=false -auto-approve tfplan
```

Wrap-up and next steps

Next video