

Recap of Terraform Basics with Examples

Deploy a Single EC2 Machine

- `terraform init`
- When you run `terraform init`, any plugins required, such as the AWS Terraform provider, are automatically downloaded and saved locally to a `.terraform` directory.
- `terraform apply`

Hands-on:

Let's use terraform to create an EC2 Machine using AWS Provider. Refer to 01-Getting-Started/01-Single-EC2-Server

- Key takeaways:
 - Look at the generated plan

Hands-on:

Let's add the name of the EC2 Machine

- Key takeaways:
 - Refer to the generated plan. Notice that Terraform already knows that the EC2 machine exists, and it just needs to update a tag.
 - That's the advantage of Declarative Language.
 - You just tell Terraform the state you want. It would automatically identify what's needed to be done to get the desired state.

Hands-on:

Let's destroy resources with auto approve switch

- Key takeaways:
 - Refer to the generated plan.

```
terraform destroy -auto-approve
```

Learn about Input and Output Vars

Hands-on:

Let's use terraform to create an EC2 Machine using AWS Provider. Refer to 01-Getting-Started/02-Single-EC2-Server-With-Vars

- Key takeaways:
 - Look at how variables.tf defines all possible vars along with their default values.
 - Look at outputs.tf file which defines the output vars
 - Note that terraform prompts to provide the var name

Various ways by which one can provide input:

- You can also use following command

```
terraform apply -var ec2_machine_ami=ami-0cb0e70f44e1a4bb5
```

- Or, you can also create an Environment variable to provide input vars:

```
export TF_VAR_ec2_machine_ami=ami-0cb0e70f44e1a4bb5  
terraform apply
```

- Or, you can add tfvars file to the working directory:

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

- Notice the output var printed at the end of the command execution
- To get the output vars you can also run following:

```
terraform output  
terraform output <output_var_name>  
terraform output public_ip
```

- provide human-readable output from a state
- inspect the current state as Terraform sees it
- Let's look at the *dependency graph* generated by Terraform

```
terraform graph
```

- <http://www.webgraphviz.com/>

More Commands: <https://www.terraform.io/docs/commands/index.html>