Step-01: Introduction

- Understand about local-exec Provisioner
- The `local-exec` provisioner invokes a local executable after a resource is created.
- This invokes a process on the machine running Terraform, not on the resource.

Step-02: Review local-exec provisioner code

- We will create one provisioner during creation-time. It will output private ip of the instance in to a file named `creation-time.txt`
- We will create one more provisioner during destroy time. It will output destroy time with date in to a file named `destroy-time.txt`
- **c6-linux-virtual-machine.tf**

```
# local-exec provisioner (Creation-Time Provisioner - Triggered during Create Resource)
provisioner "local-exec" {
  command = "echo ${azurerm_linux_virtual_machine.mylinuxvm.public_ip_address} >> creation-time.txt"
  working_dir = "local-exec-output-files/"
  #on_failure = continue
}

# local-exec provisioner - (Destroy-Time Provisioner - Triggered during Destroy Resource)
provisioner "local-exec" {
  when    = destroy
  command = "echo Destroy-time provisioner Instanace Destroyed at `date` >> destroy-time.txt"
  working_dir = "local-exec-output-files/"
}
```

Step-03: Review Terraform manifests & Execute Terraform Commands

```
# Terraform Initialize
terraform init

# Terraform Validate
terraform validate

# Terraform Format
terraform fmt

# Terraform Plan
terraform plan

# Terraform Apply
terraform apply -auto-approve

# Verify
Verify the file in folder "local-exe-output-files/creation-time.txt"
```

Step-04: Clean-Up Resources & local working directory

```
# Terraform Destroy
terraform destroy -auto-approve

# Verify
Verify the file in folder "local-exec-output-files/destroy-time.txt"
```

```
# Delete Terraform files
rm -rf .terraform*
rm -rf terraform.tfstate*
```