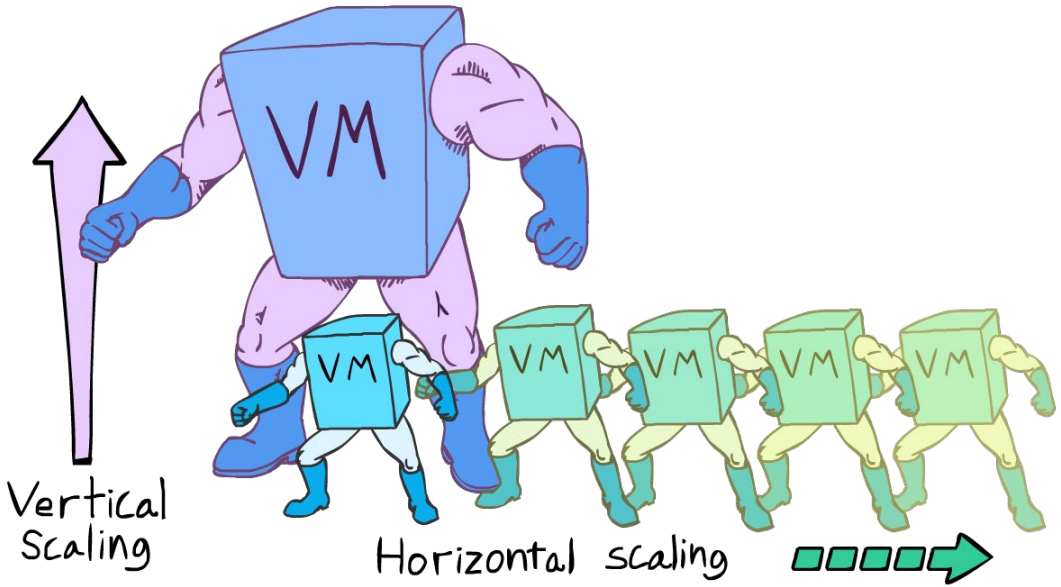
Cyber Defense Organization

Spring 2020 - Terraform

Word Of The Week: Scalability

Word of the Week

Scalability is an attribute that describes the ability of a process, network, software or organization to grow and manage increased demand

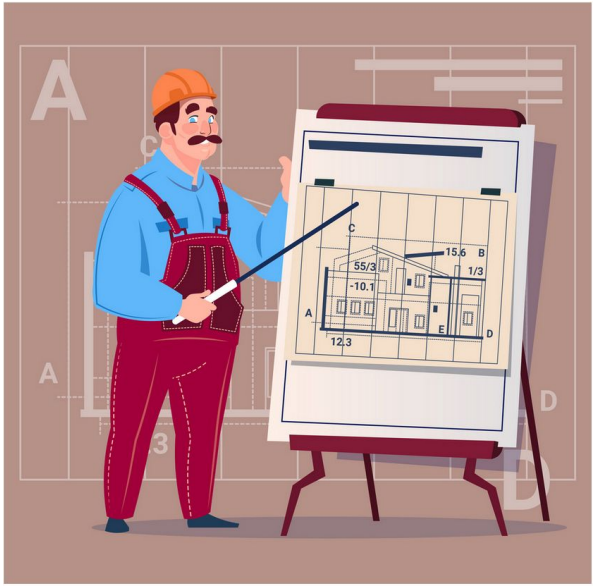


We like this.

Terraform

What is Infrastructure as Code (IAC)

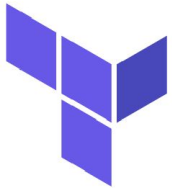
- Process of managing and provisioning computer data centers through machine-readable definition files,
- Does not use manual configuration



Essentially building a blueprint that the virtualization needs to follow.

-Buying a computer, specifying what parts you want, what hard drive you want, what kind of operating system you want. It’s all done in the code

What is Terraform?



HashiCorp
Terraform

- Written in HCML (Hashi Corp Markup Language)
- open-source infrastructure as code software tool
- It enables users to define and provision a datacenter infrastructure using a high-level configuration language known as Hashicorp Configuration Language, or optionally JSON.
- Industry Standard
- Integrates and is compatible with a lot of different software

Lecture about Terraform

Talks to multiple cloud/infrastructure providers

Ensure creation and consistency

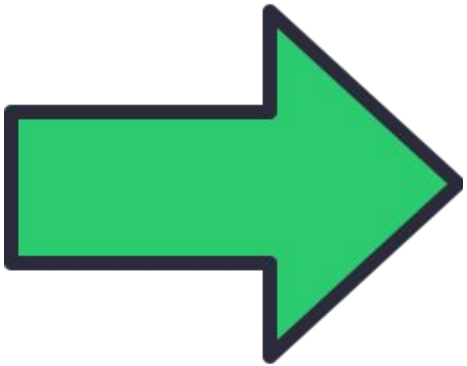
Expressed in an API-Agnostic DSL

What it looks like

Lecture about Terraform

INPUT

- Variables in JSON, HCL, CLI
- Variables are based on source platform, cloud provider etc..
- Modules



OUTPUT

- Specific Text
- IP Addresses
- Errors (Very Helpful)
- Success and Status

Lecture about Terraform

```
provider "aws" {
    region = "us-east-2"
    }

    resource "aws_instance" "example" {
        ami           =   "ami-0d5d9d301c853a04a"
        instance_type = "t2.micro"
        key_name = "ilovezeal"
        vpc_security_group_ids = [aws_security_group.instance.id]

        user_data = <<-EOF
                #! /bin/bash
                    sudo apt-get update
                    sudo apt-get install -y openjdk-8-jdk
                    sudo wget -q -O - https://pkg.jenkins.io/debian/jenkins.io.key | sudo apt-key add -
                    sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'
                    sudo apt update
                    sudo apt install -y jenkins

        EOF

        tags = {
            Name = "terraform-example"

        }
    }
```

```
# Configure the Docker provider
provider "docker" {
  host = "tcp://127.0.0.1:2376/"
}

# Create a container
resource "docker_container" "foo" {
  image = "${docker_image.ubuntu.latest}"
  name  = "foo"
}

resource "docker_image" "ubuntu" {
  name = "ubuntu:latest"
}
```

File/Folder Structure

```
my-terraform-files
|
└──my-terraform-module
|   |   main.tf
|   |   variables.tf
|   |   outputs.tf
|
└──my-other-terraform-module
|   |   main.tf
|   |   variables.tf
|   |   outputs.tf
```

File Tree

main.tf

```
provider "aws" {
 region = "eu-west-1"
 version = "~> 1.19"
 access_key = "${var.aws_access_key}"
 secret_key = "${var.aws_secret_key}"
}
```

outputs.tf

```
output "instance_ip_addr" {
  value = aws_instance.server.private_ip
}
```

variables.tf

```
variable "instance_type" {
 description = "Type of EC2 instance to use"
 default = "t2.small"
}

variable "instance_types" {
 type    = "map"
 default = {
   "dev" = "t2.small"
   "prod" = "t3.large"
 }
}

variable "autoscale_min" {
 description= "The minimum number of EC2 instances"
 default = 2
}
```

Commands

Lecture about Terraform

Command	Description
Terraform init	The terraform init command is used to initialize a working directory containing Terraform configuration files. This is the first command that should be run after writing a new Terraform configuration or cloning an existing one from version control.
Terraform plan	Creates a plan of changes required. Does nothing to the infra.
Terraform apply	The terraform apply command is used to apply the changes required to reach the desired state of the configuration, or the pre-determined set of actions generated by a terraform plan execution plan.
Terraform destroy	Deletes all the resources

Demo

Watch me Do

- Build Instance
- Build Security Group (Security!)
- Terraform init
- Terraform Plan
- Terraform Apply
- Terraform Destroy
- Show

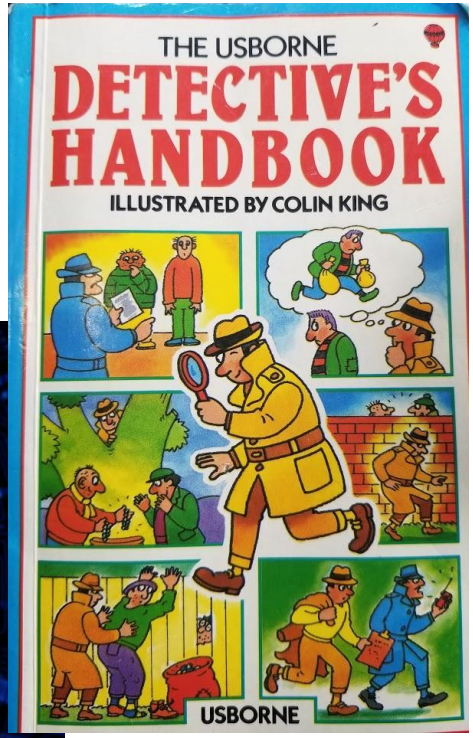
Interactive Part

- Infrastructure as Code in Docker using Katacoda

- Building an Nginx Web Server in Docker via Terraform

https://www.katacoda.com/courses/terraform/deploy-nginx

Next Week: Windows Forensics



Add us on Social Media!

Twitter: **@ualbanyCDO**

Instagram: **ualbany_cdo**

Website: **uacyber.org**

Myinvolvement: **Cyber Defense Org**

We have a discord!