Automating VM Deployment on vSphere

With PowerCLI and Terraform

HashiCorp Terraform

Who Am I?

Stephan McTighe

He | Him

I'm an infrastructure engineer focused on virtualisation and VMware solutions.

VMware vExpert.

https://stephanmctighe.com

me@stephanmctighe.com

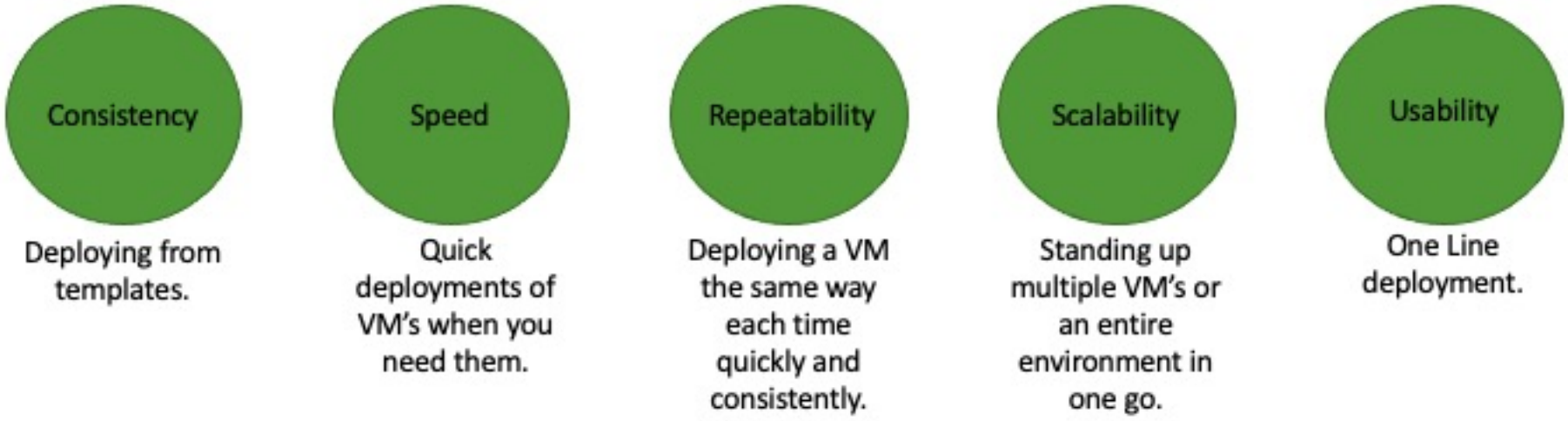
www.linkedin.com/in/stephanmctighe

@vStephanMcTighe

https://github.com/smctighevcp

Automation; Why?

In this day and age, time is money. You don't want to be giving time to repeatable tasks, remediating inconsistencies or human error when its avoidable.

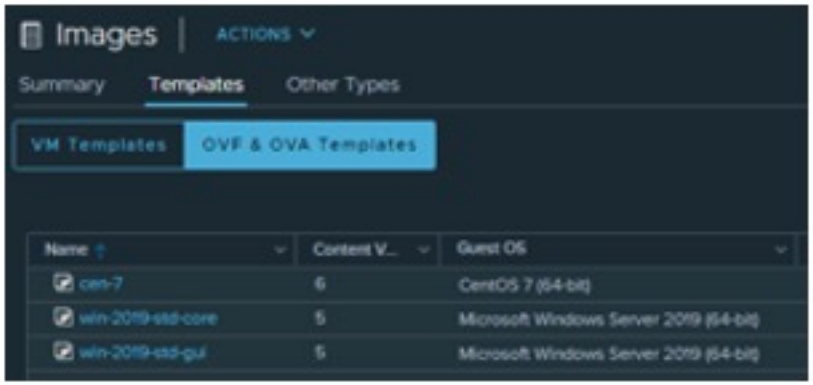


Where to begin?

One of the best ways to ensure your virtual machines are deployed consistently is by starting with a template built to your organisations operational and security standards.

Your template could include:

- Security Baselines
- Standard Software Installations
- Locale settings



Template Management

Packer by HashiCorp, is a great option for deploying templates in a repeatable fashion.

- Easy Installation
- Simple Command Lines
- Minimal Components
- Extensively Customizable

JSON

HCL

PowerCLI

- Readily available
- Common Language
- New Features

```
Class Clusters : System.Management.Automation.IValidateSetValuesGenerator {
    [String[]] GetValidValues() {
        $Cluster = (Get-Cluster).Name
        return [String[]] $cluster
    }
}
```

```
process {
    Write-Host "INFO: Building VM..." -ForegroundColor Green
    New-VM -Name $VMName -OSCustomizationSpec $CustomSpec -Template $Template -resourcePool $Cluster -datastore $Datastore > $null
    $newVM = Get-VM $VMName | Set-VM -MemoryGB $Memory -NumCPU $vCPU -CoresPerSocket $vCPU -Notes "$($_.Notes) Built $(Get-Date)" -Confirm:$false
    $HostPG = $newVM | Get-VMHost | Get-VirtualPortGroup | Where {$_.Name -eq $Network} -WarningAction SilentlyContinue
    $newVM | Get-NetworkAdapter | Set-NetworkAdapter -Portgroup $HostPG -Confirm:$false -WarningAction SilentlyContinue > $null
```

Demo

Deploying Virtual Machines with PowerCLI

PowerCLI

Demo Summary

- Common Language – PowerShell!
- Wide Community Support
- Powerful

- Can become complex
- New Features

Terraform

- Infrastructure as Code
- Good Support Documentation
- HCL – Human and Machine Friendly

```
resource "vsphere_virtual_machine" "tfdemo3" {
  name             = var.computer_name
  resource_pool_id = data.vsphere_compute_cluster.cluster.resource_pool_id
  datastore_id     = data.vsphere_datastore.datastore.id

  num_cpus                 = var.num_cpus
  num_cores_per_socket     = var.num_cores_per_socket
  memory                   = var.memory
  folder                   = var.folder
```

Common Commands

terraform init
terraform plan
terraform apply
terraform destroy

Demo

Deploying Virtual Machines with Terraform

HashiCorp
Terraform

Demo Summary

- Reusable code
- Version Control
- Multi platform/cloud

- Potentially a new concept
- Can destroy as easily as create!

Thank You

Any Questions… ?

https://stephanmctighe.com

me@stephanmctighe.com

www.linkedin.com/in/stephanmctighe

@vStephanMcTighe

https://github.com/smctighevcp