collabdays | lisbon

October 10th, 2020

Azure ❤ Terraform ❤ Azure

by Peter De Tender - @pdtit – petender@microsoft.com

Our Sponsors

DIAMOND

pointfire Multilingual SharePoint

veeAM

COMMUNITY

ESPC20 ONLINE 14-15 OCTOBER 2020

EUROPEAN COLLABORATION SUMMIT

Microsoft

Office 365 Portugal

sharept

What you will learn

ARM Template Overview

What is Terraform

Terraform Basics

Best of Both Worlds

Terraform in Azure DevOps

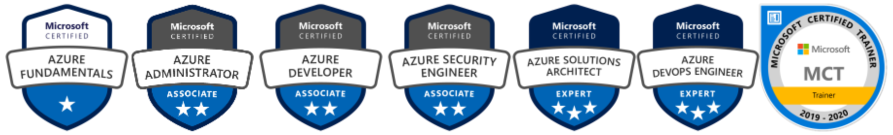
Q&A

Microsoft

Peter De Tender

Azure Technical Trainer @ Microsoft

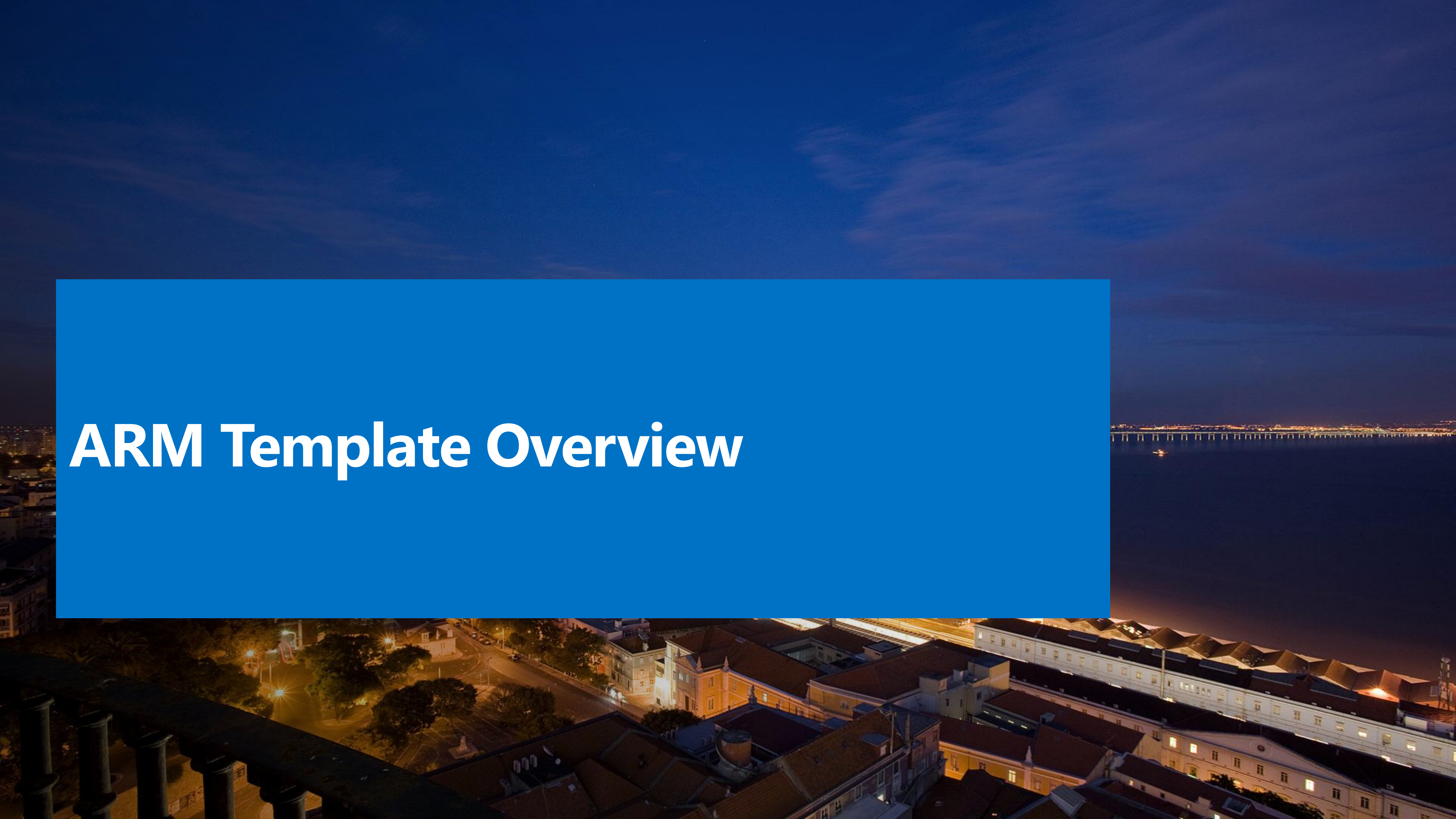
- +22 years in the IT industry
- +10 years MCT, Chairman EMEA IAMCT community
- Last 6 years focus on Azure (Readiness, Architect)
- Former Azure MVP (5y)
- Technical writer, book author (Apress, Packt,…)
- World traveller (for business)


Microsoft Certified Azure Fundamentals
Microsoft Certified Azure Administrator Associate
Microsoft Certified Azure Developer Associate
Microsoft Certified Azure Security Engineer Associate
Microsoft Certified Azure Solutions Architect Expert
Microsoft Certified Azure DevOps Engineer Expert
Microsoft Certified Trainer MCT Trainer 2019 - 2020

"…Bring me an audience, I give them Azure knowledge in return…"

petender@Microsoft.com

@pdtit #007FFFLearning

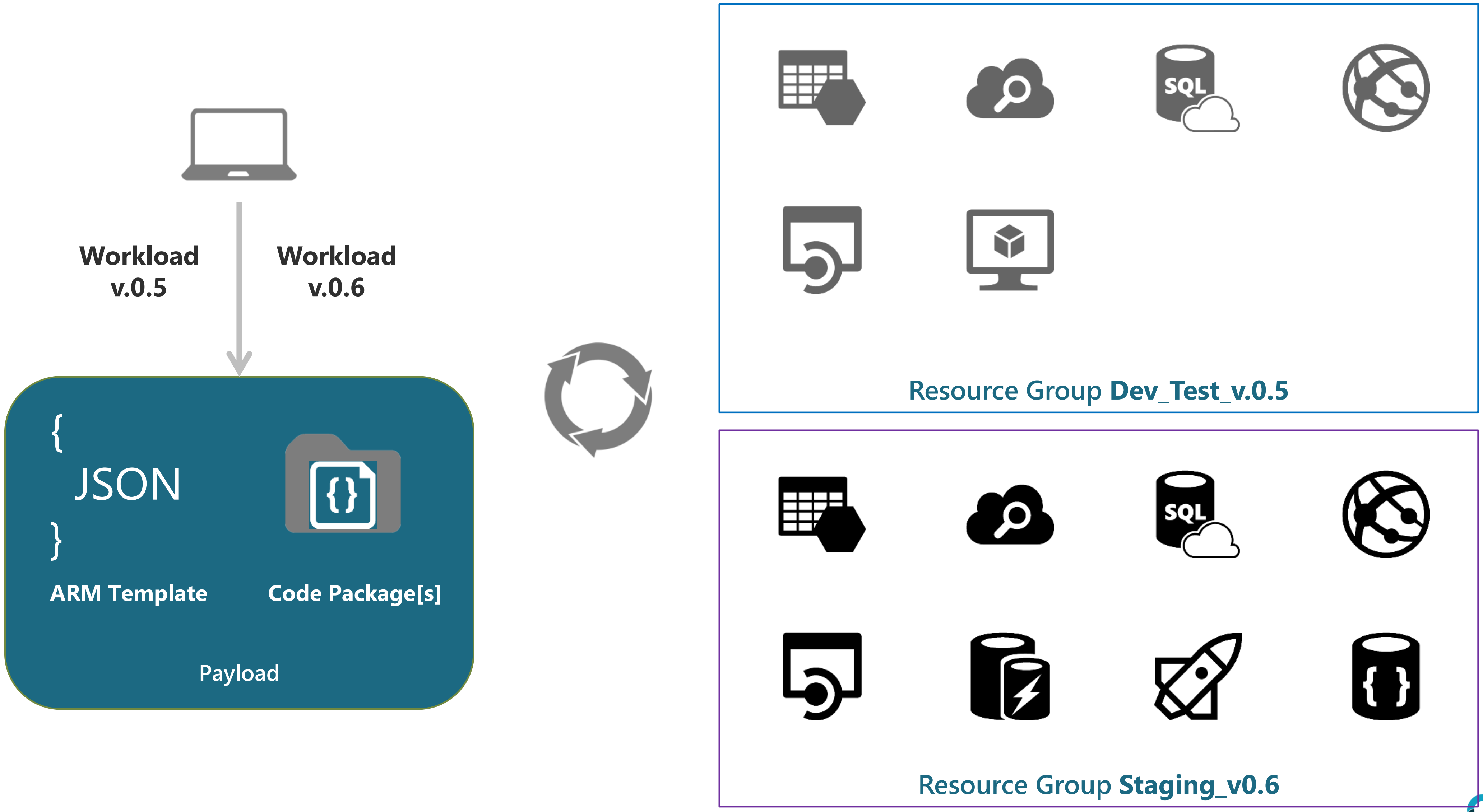


ARM Template Overview

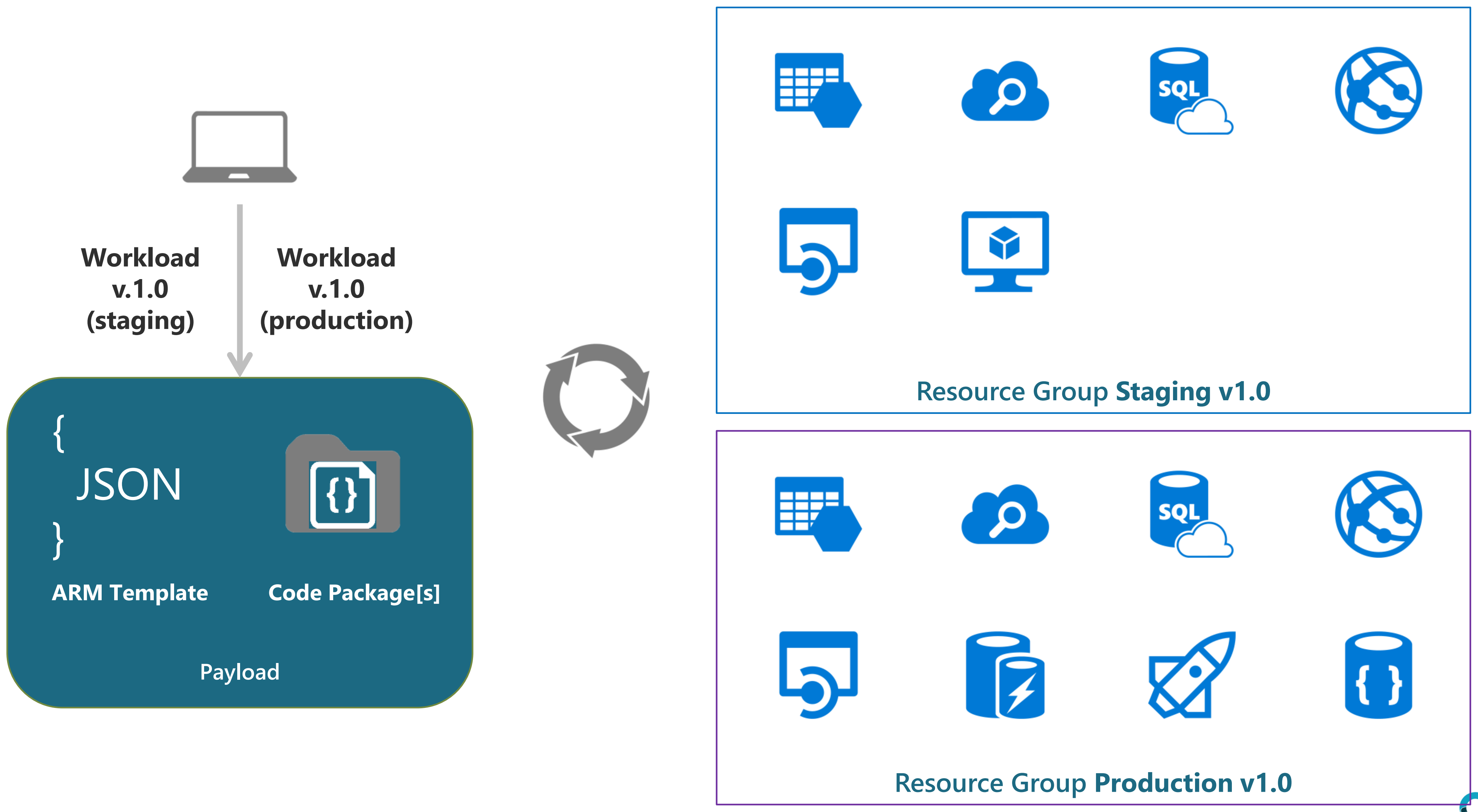
Infrastructure as Code (IAC)

- *"…Management of infrastructure components (storage, networking, servers, containers, app services,…) using a descriptive model…"*
- Idempotence
- Immutable
- Lights-out deployment

Infrastructure as Code (IAC)



Infrastructure as Code (IAC)



Demo

ARM Template Deployment

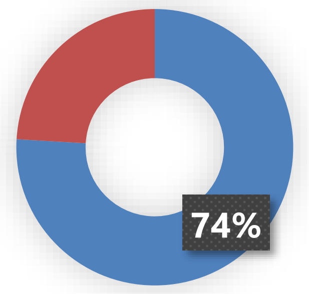
Challenges of ARM Templates

- **JSON Format** is fine for computers, but not for humans
- Complex in **structure**
- **Authoring process** is hard (and frustrating…)
- **Doesn't** allow for **comments**
- Deployment itself is used to validate **deployment state**
- No "**Undo**" or **Delete** feature
- Azure **native**

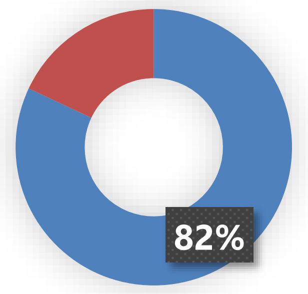
Challenges of ARM Templates

- ✓ Regulation
- ✓ Data Governance
- ✓ Customization
- ✓ Compliance
- ✓ (Legacy) Apps

74% runs Hybrid



82% multi-cloud



ARM Templates are not helping those organizations…

(and the other challenges don’t help either…)

What is Terraform

What is Terraform?

- **"... A product allowing for provisioning infrastructure and application resources across private cloud, public cloud and external services, using a common workflow..."**
- **Multi-cloud**
- **Hashicorp Configuration Language (HCL)**
- **Infrastructure as Code + more**

HashiCorp
Terraform

- Terraform
- Vault
- Consul
- Nomad
- Vagrant
- Packer

Why Terraform?

- HCL feels like a natural / human language
- Faster Deployments
- State is key
- Destroy

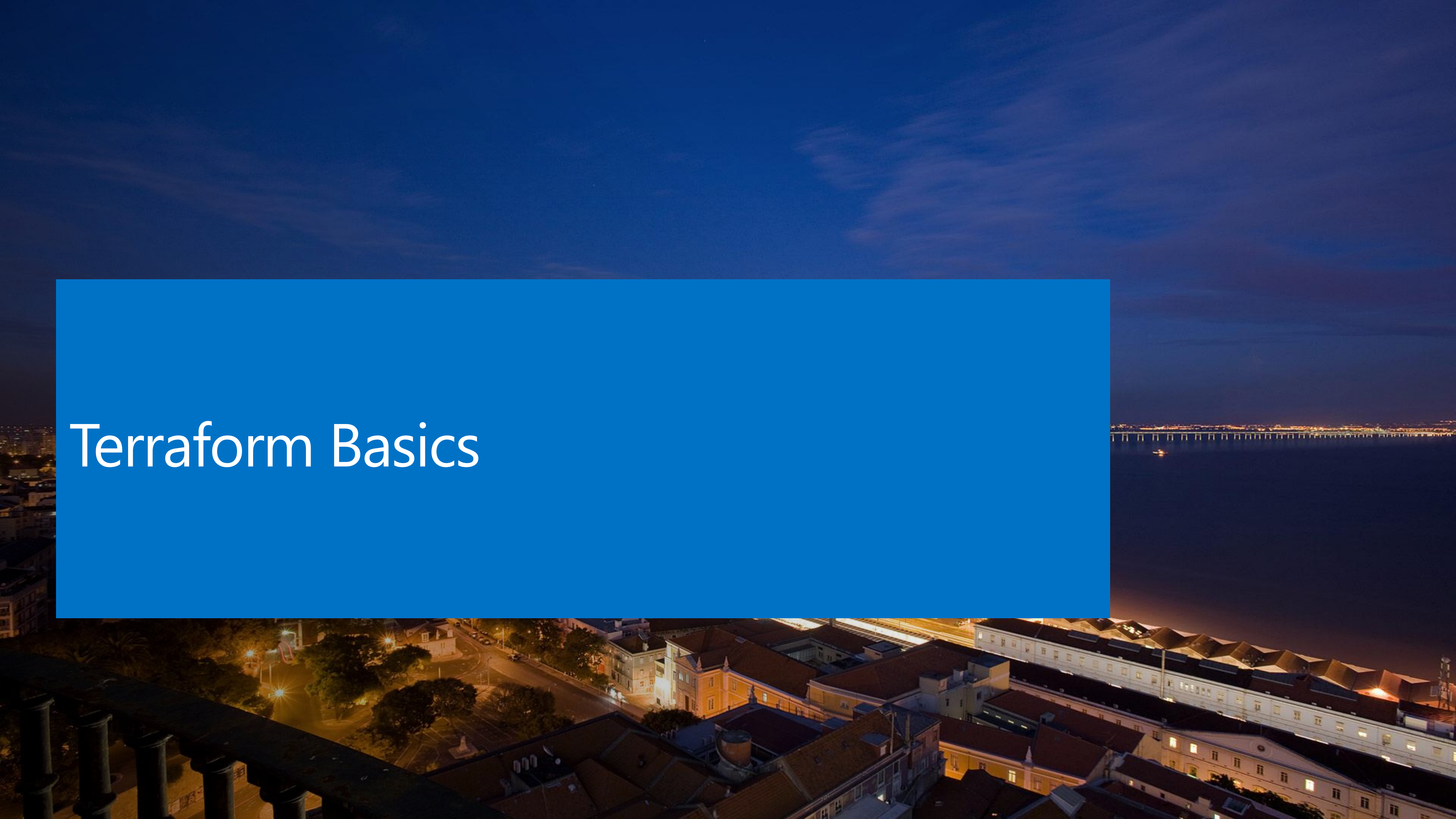
ARM Template Syntax <> HCL Syntax

```
    "$schema": "https://schema.management.azure.com/..json#",
    "contentVersion": "1.0.0.0",
    "parameters": {},
    "variables": {},
    "resources": [{
            "type": "Microsoft.Resources/resourceGroups",
            "apiVersion": "2018-05-01",
            "location": "eastus",
            "name": "demo-storage",
            "properties": {}
        },
        {
            "type": "Microsoft.Storage/storageAccounts",
            "name": "demo-storage",
            "apiVersion": "2018-02-01",
            "location": "eastus",
            "sku": {
                "name": "Standard_LRS"
            },
            "kind": "Storage",
            "properties": {}
        }
    ]
```

```
resource "azurerm_resource_group" "testrg" {
    name = "resourceGroupName"
    location = "westus"
}

resource "azurerm_storage_account" "testsa" {
    name = "storageaccountname"
    resource_group_name = "testrg"
    location = "westus"
    account_tier = "Standard"
    account_replication_type = "GRS"
}
```

Demo

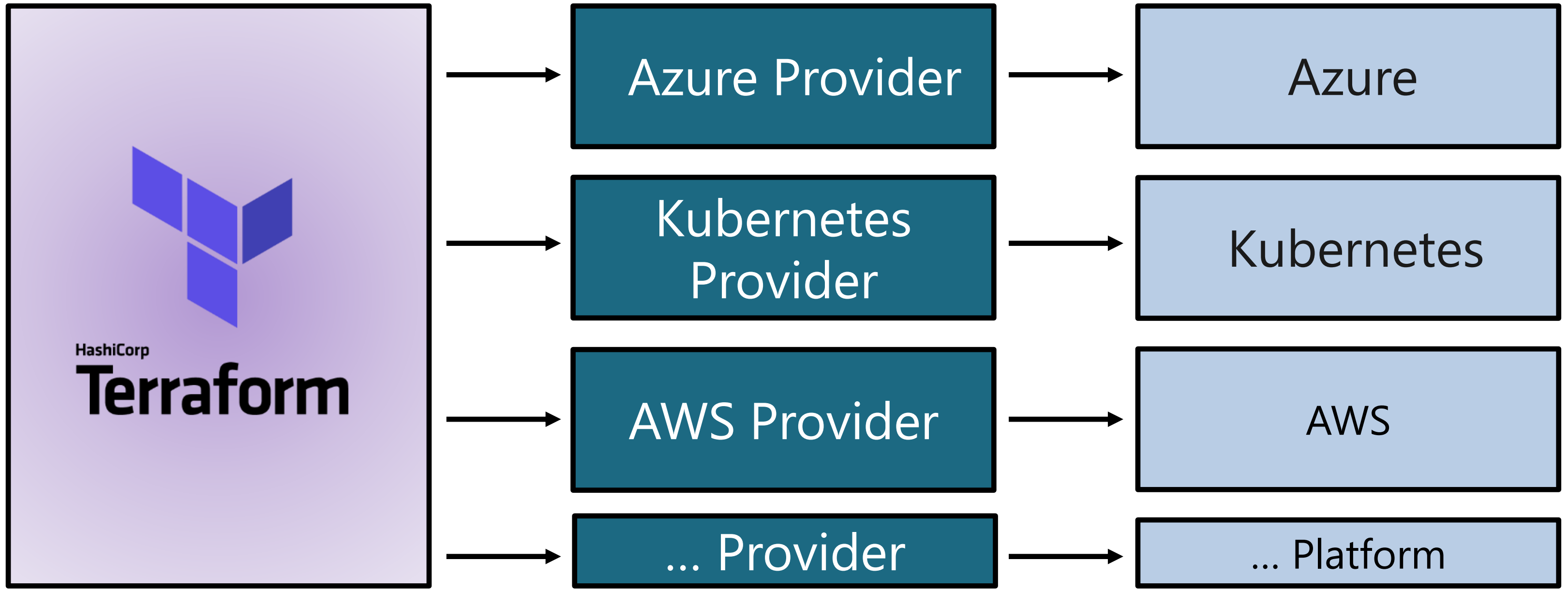
HCL Template Syntax



Terraform Basics

Terraform Providers

- Extensions allow for resources deployment
- Manage cloud / endpoint specific API interactions
- Available for major cloud and other platforms



Terraform Template for Azure < > AWS

```
# Configure the AWS Provider

  provider "aws" {

    region = "us-east-1"

  }


  Variable "vpc_id" {}

  data "aws_vpc" "selected" {
    id = "${var.vpc_id}"
  }
  resource "aws_subnet" "example" {
    vpc_id = "${data.aws_vpc.selected.id}"
    availability_zone = "us-west-2a"
    cidr_block =
"${cidrsubnet(data.aws_vpc.selected.cidr_block,
4, 1)}“

 }
```

1

2

3

4

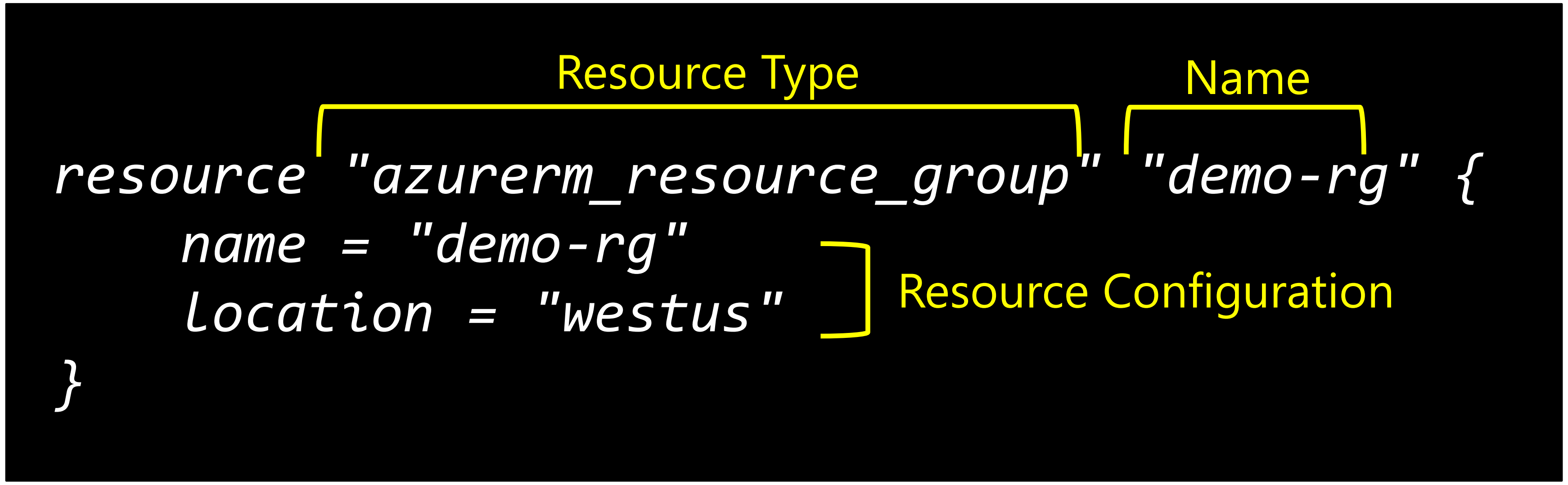
```
# Configure the Azure Provider

  provider "azurerm" {

  version = "=1.22.0"

 }


Variable “vnet_id” {}

data "azurerm_virtual_network" "test" {
  name = "production"
  resource_group_name = "networking"
}

resource "virtual_network_id" {
    vnet = "${data.azurerm_virtual_network.test.id}"
    subnet = “${data.azurerm_virtual_network.subnet.id}
}
```

85-90% of the HCL syntax is identical

collabdays | lisbon

Basic Resource Creation

Resource Type	= Required Provider
Name	= Internal name you want
Configuration	= Deployment Parameters and details



```
resource "azurerm_resource_group" "demo-rg" {
    name = "demo-rg"
    location = "westus"
}
```

Basic Terraform Commands

INIT	PLAN	APPLY	DESTROY
Initialize the working folder	Pre-flight validation	Actual Deploy	Removes Resources

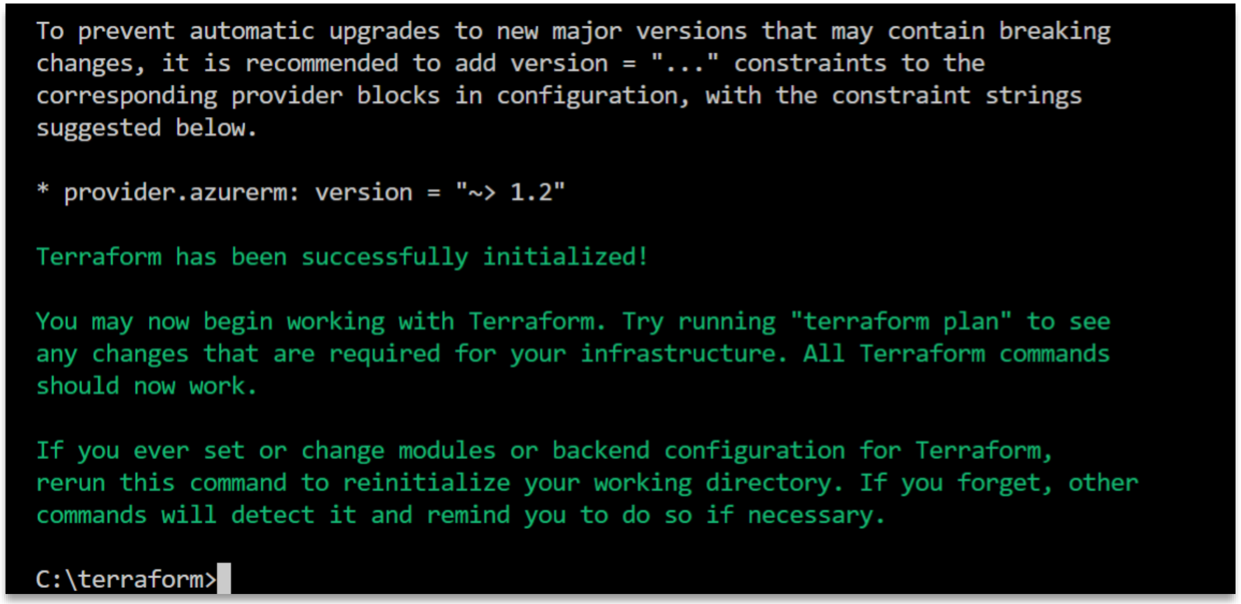
Yes, that's all it takes…

… 4 commands to manage your cloud environments

Terraform “init”

- Initializes a working directory, containing configuration files

- This is the first command to run after creating/updating config



```
To prevent automatic upgrades to new major versions that may contain breaking
changes, it is recommended to add version = "..." constraints to the
corresponding provider blocks in configuration, with the constraint strings
suggested below.

* provider.azurerm: version = "~> 1.2"

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.

C:\terraform>
```

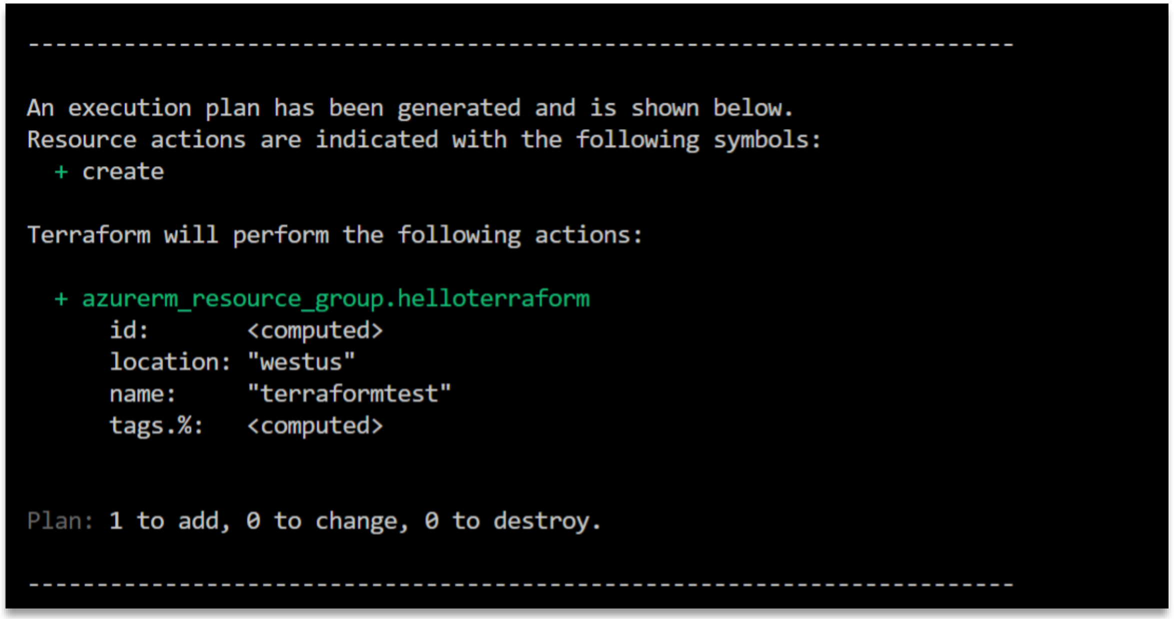
Demo

Terraform Init

Terraform "plan"

- Creates an execution plan of a deployment, validating and confirming the configuration, before running the actual deployment

- Allows "–out" parameter to store the file for later usage (apply)



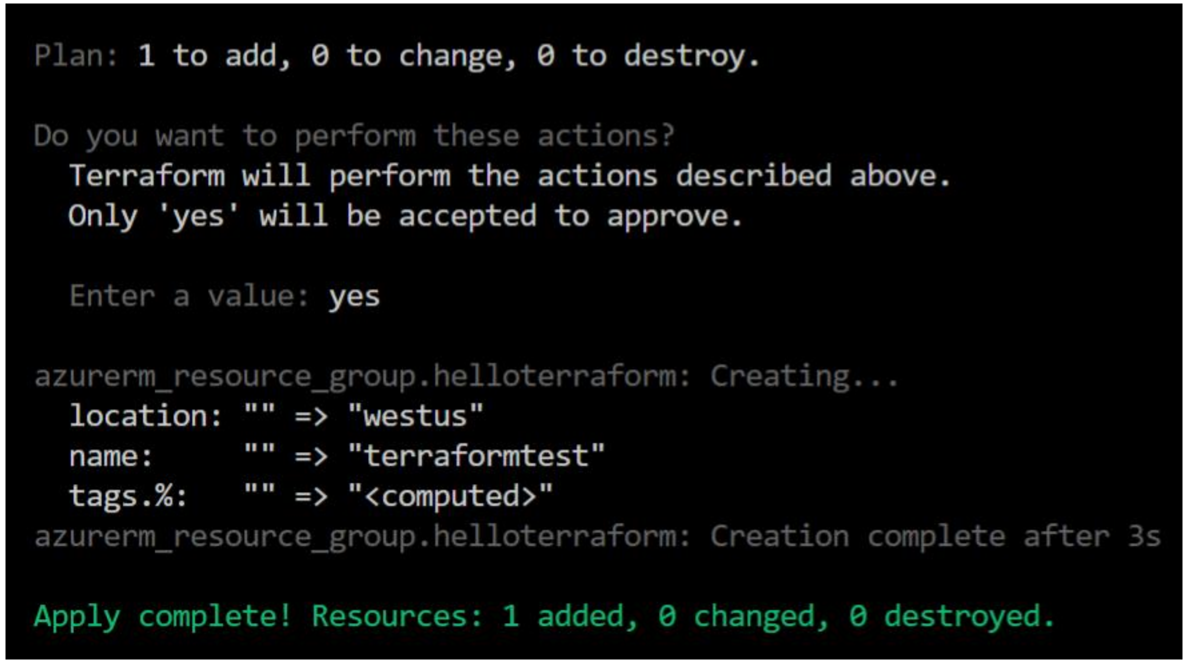
```
------------------------------------------------------------------------

An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  + azurerm_resource_group.helloterraform
      id:        <computed>
      location:  "westus"
      name:      "terraformtest"
      tags.%:    <computed>


Plan: 1 to add, 0 to change, 0 to destroy.

------------------------------------------------------------------------
```

Demo

Terraform Plan

Terraform "apply"

- Runs the actual execution of the deployment/configuration
- Guarantees applying the changes until the desired configuration is reached, based on the settings in the plan phase

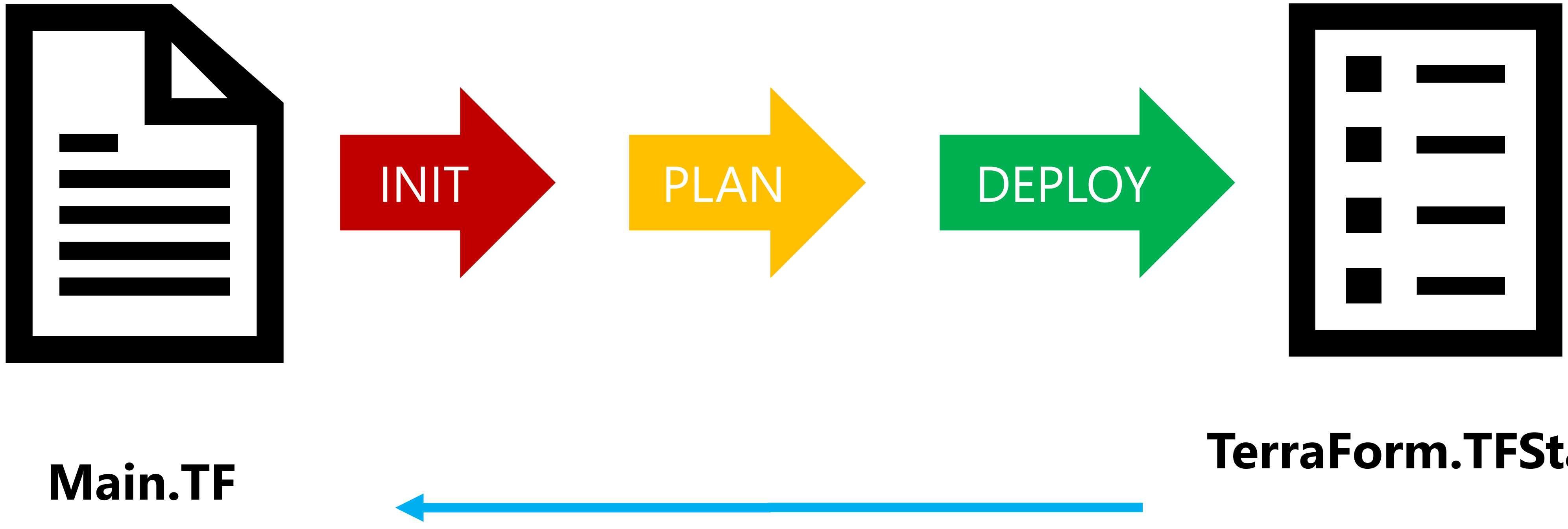


```
Plan: 1 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

azurerm_resource_group.helloterraform: Creating...
  location: "" => "westus"
  name:     "" => "terraformtest"
  tags.%:   "" => "<computed>"
azurerm_resource_group.helloterraform: Creation complete after 3s

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
```

Demo

Terraform Apply

Terraform State file



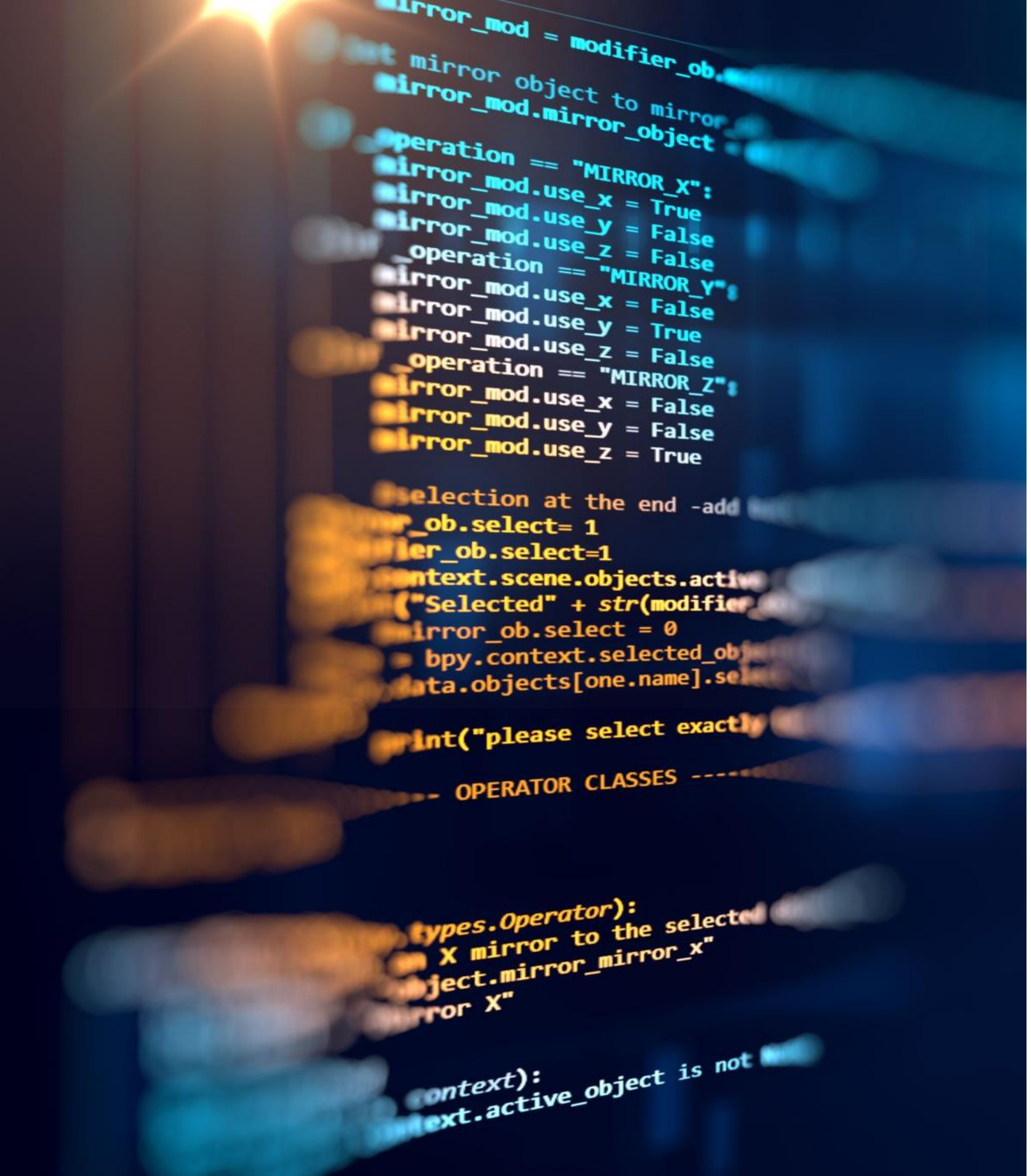
Main.TF

= your definition

TerraForm.TFState

= the result state

Best of Both Worlds



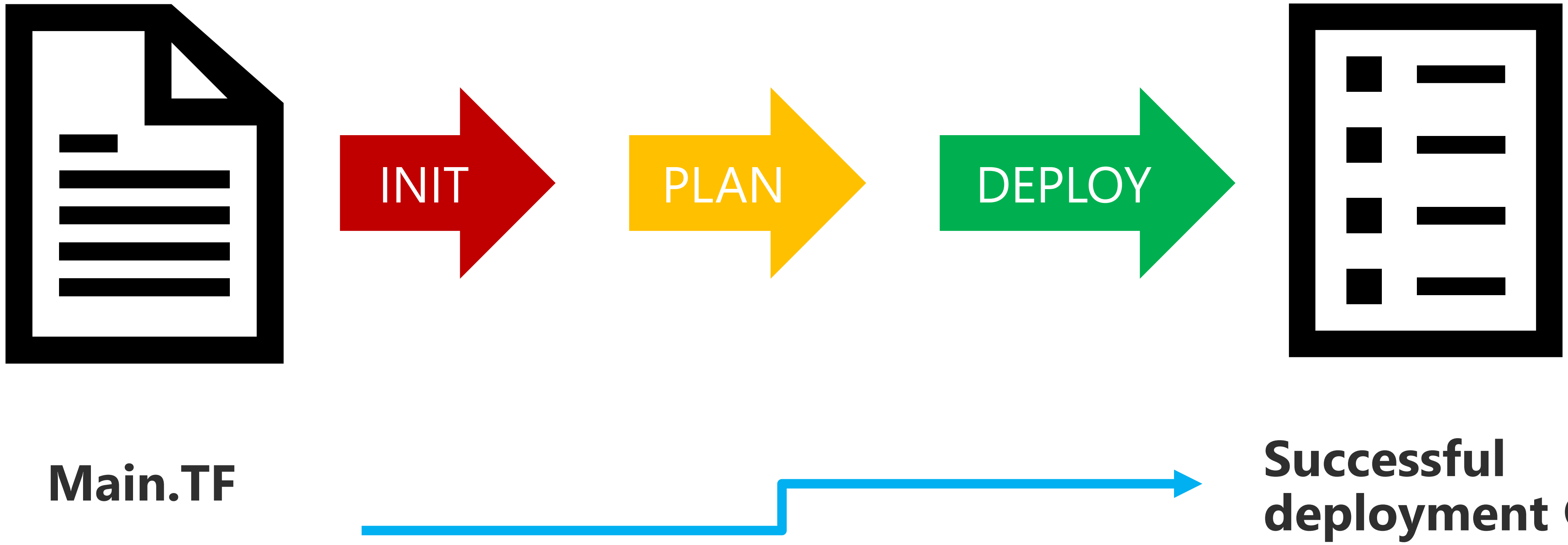
Best of Both Worlds

Terraform executable is integrated with **Cloud Shell**

Terraform can recognize an **ARM template** as part of a Terraform template, in combination with Terraform code

Terraform integrates with **Azure DevOps** CI/CD Pipelines

Best of Both Worlds



Main.TF

Successful deployment ☺

= TerraForm snippets
+ ARM Template snippets

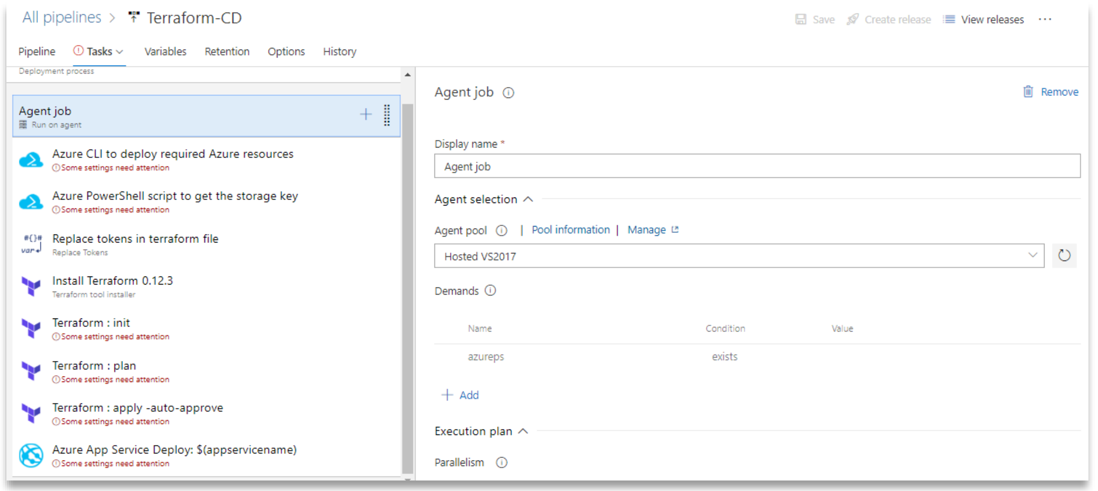
Demo

Terraform Best of Both Worlds

Terraform in Azure DevOps

Terraform in Azure DevOps

- Same 3 Terraform steps, but now automated
- Terraform “state” file preferably stored in Azure Storage to allow sharing across teams/pipelines



All pipelines > Terraform-CD

Agent job
Run on agent

Azure CLI to deploy required Azure resources
Some settings need attention

Azure PowerShell script to get the storage key
Some settings need attention

Replace tokens in terraform file
Replace Tokens

Install Terraform 0.12.3
Terraform tool installer

Terraform : init
Some settings need attention

Terraform : plan
Some settings need attention

Terraform : apply -auto-approve
Some settings need attention

Azure App Service Deploy: $(appservicename)
Some settings need attention

Agent job

Display name *: Agent job

Agent selection

Agent pool | Pool information | Manage: Hosted VS2017

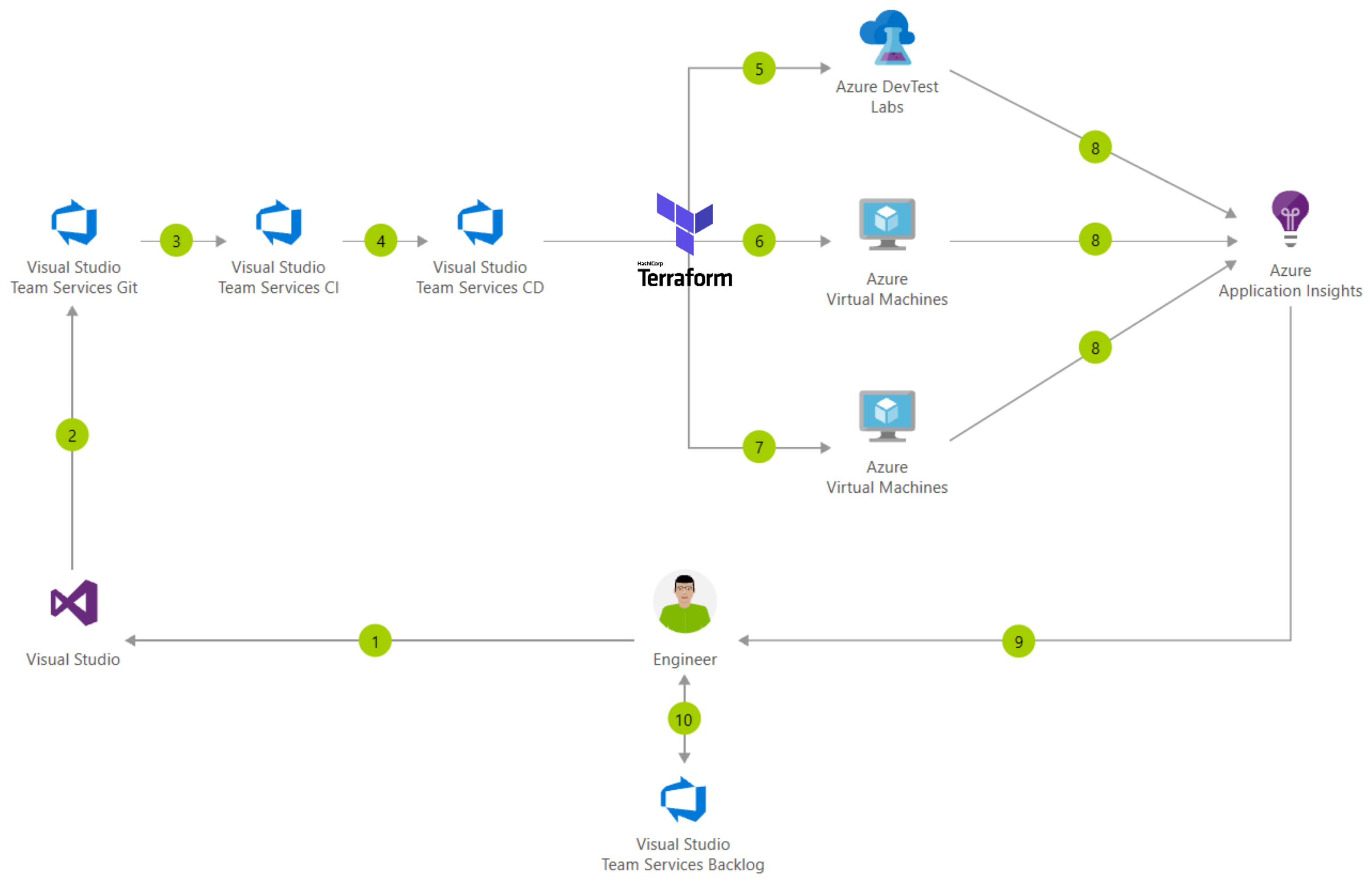
Demands

Name	Condition	Value
azureps	exists	

Execution plan

Parallelism

Terraform in Azure DevOps



Azure DevTest Labs
Visual Studio Team Services Git
Visual Studio Team Services CI
Visual Studio Team Services CD
HashiCorp Terraform
Azure Virtual Machines
Azure Application Insights
Azure Virtual Machines
Visual Studio
Engineer
Visual Studio Team Services Backlog
1
2
3
4
5
6
7
8
8
8
9
10

1. Change application source code
2. Commit Application Code and Azure Resource Manager (ARM) Template
3. Continuous integration triggers application build and unit tests
4. Continuous deployment trigger orchestrates deployment of application artifacts with environment specific parameters
5. Deployment to QA environment
6. Deployment to staging environment
7. Deployment to production environment
8. Application Insights collects and analyses health, performance and usage data
9. Review health, performance and usage information
10. Update backlog item

DEMO

Terraform Azure DevOps CI/CD Pipelines

Q & A

What you learned

ARM Template Overview

What is Terraform

Terraform Basics

Best of Both Worlds

Terraform in Azure DevOps

Q&A

Microsoft

"...Bring me an audience, I give them Azure knowledge in return..."

petender@Microsoft.com

@pdtit #007FFFLearning

Our Sponsors

DIAMOND

pointfire
Multilingual SharePoint

veeAM

COMMUNITY

ESPC20 ONLINE 14-15 OCTOBER 2020

EUROPEAN COLLABORATION SUMMIT

Microsoft

Office 365 Portugal

sharept

collabdays | lisbon