Luis Quintanilla

Camp 2020

Infrastructure as Code

Luis Quintanilla

@ljquintanilla

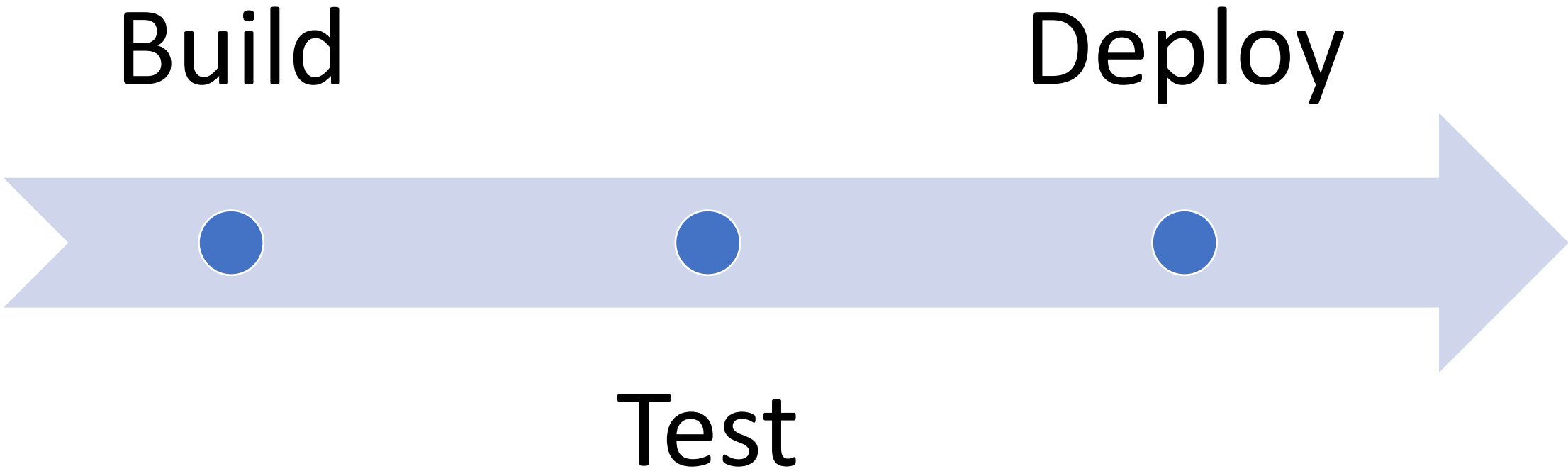
Hello ☺

- Content Developer, ML.NET & Azure Machine Learning
- Email: Luis.Quintanilla@microsoft.com
- Twitter: @ljquintanilla
- LinkedIn: lquintanilla01
- GitHub: luisquintanilla

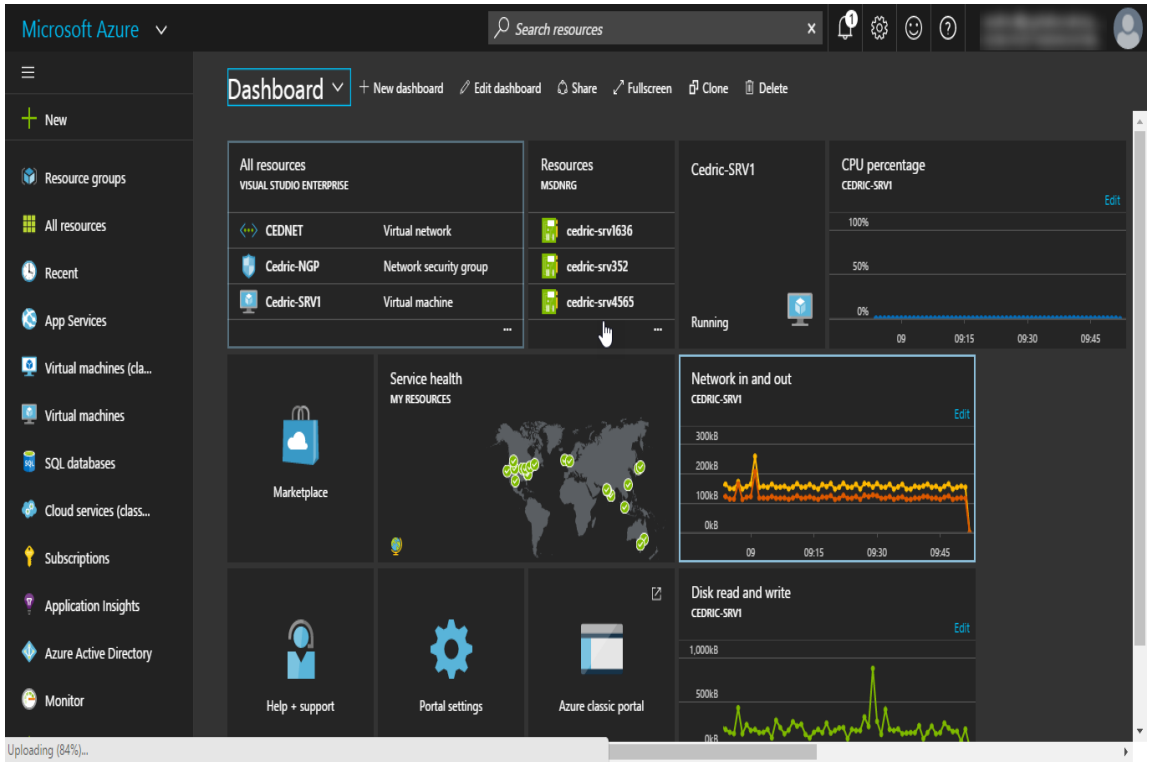
Agenda

- Deployment Process
- Azure Portal
- Infrastructure as Code
- ARM Templates
- Terraform
- Pulumi

Deployment Process



Azure Portal



- Pros
  - Web UI
  - Wizards / Informational Tooltips
  - All options are accessible
  - Discoverable
- Cons
  - Information overload
  - Manual
  - Not reproducible
  - No versioning

Infrastructure As Code

- Management and Provisioning of compute resources through configuration / definition files.

ARM Templates

```
{
    "$schema": "http://schema.management.azure.com/schemas/2014-04-01-preview/deploymentTemplate.jsc
    "contentVersion": "1.0.0.0",
    "parameters": {
        "name": {
            "type": "String"
        },
        "location": {
            "type": "String"
        },
        "apiType": {
            "type": "String"
        },
        "sku": {
            "type": "String"
        }
    },
    "resources": [
        {
            "type": "Microsoft.CognitiveServices/accounts",
            "apiVersion": "2017-04-18",
            "name": "[parameters('name')]",
            "location": "[parameters('location')]",
            "sku": {
                "name": "[parameters('sku')]"
            },
            "kind": "[parameters('apiType')]",
            "properties": {
                "customSubDomainName": "[parameters('name')]"
            }
        }
    ]
}
```

- Pros
  - Somewhat familiar syntax (JSON)
  - Multiple ways to deploy (REST, CLI, PowerShell)
  - Can be automated
  - Versioned
- Cons
  - No UI
  - Not validated
  - Not discoverable
  - Manual(ish)

Demo
ARM Template

Terraform

```
provider "azurerm" {
  version="~> 1.27"
}

resource "azurerm_resource_group" "iaac-terraform-example" {
    name = "iaac-rg"
    location = "East US"
}

resource "azurerm_cognitive_account" "iaac-cv-terraform" {
  name="iaac-cv-terraform"
  resource_group_name=azurerm_resource_group.iaac-terraform-example.name
  location=azurerm_resource_group.iaac-terraform-example.location
  kind="ComputerVision"
  sku_name="S1"
}

output "computer_vision_endpoint" {
    value="${azurerm_cognitive_account.iaac-cv-terraform.endpoint}"
}
```

- Pros
  - Versioned
  - Validated
  - Multiple Providers
  - Versioned
  - Code-Like
  - Modularized
- Cons
  - Domain Specific Language
  - Custom tooling
  - Not discoverable

Demo
Terraform

Pulumi

```python
import pulumi
from pulumi_azure import core, cognitive

# Create an Azure resource (Cognitive Services Account)
cvaccount = cognitive.Account('iaac-cv-pulumi',
                              # The location for the resource will be derived automatically from the resource group.
                              resource_group_name='iaac-rg',
                              kind='ComputerVision',
                              sku_name='S1')

# Export the resource endpoint for the cognitive services resource
pulumi.export('endpoint', cvaccount.endpoint)
```

- Pros
  - Code-First (.NET, TypeScript,Python)
  - Validated
  - Modularized
  - Leverage developer's existing skills
  - Versioned
  - Cross-Cloud + Kubernetes
- Cons
  - Custom tooling
  - Some scaffolding required (though automated by CLI)

Demo
Pulumi

Takeaways

- Infrastructure as Code (IaC) is a loaded term and it’s a spectrum
- DO: Use the provisioning method that best meets your needs
- DO: Use when you have to:
  - Provision multiple (often related) resources
  - Continuously provisioning resources
- DON’T: Use these methods for one-off scenarios

Please fill out the survey,
so we can grow.

www.aka.ms/campfeedback

Questions

Resources

- https://github.com/luisquintanilla/Presentations/tree/master/IaC042020
- https://docs.microsoft.com/en-us/azure/azure-resource-manager/templates/
- https://www.pulumi.com/docs/
- https://www.terraform.io/docs/index.html
- https://www.luisquintanilla.me/2019/01/05/automate-machine-learning-service-provisioning-azure-terraform/