Terraforming Azure

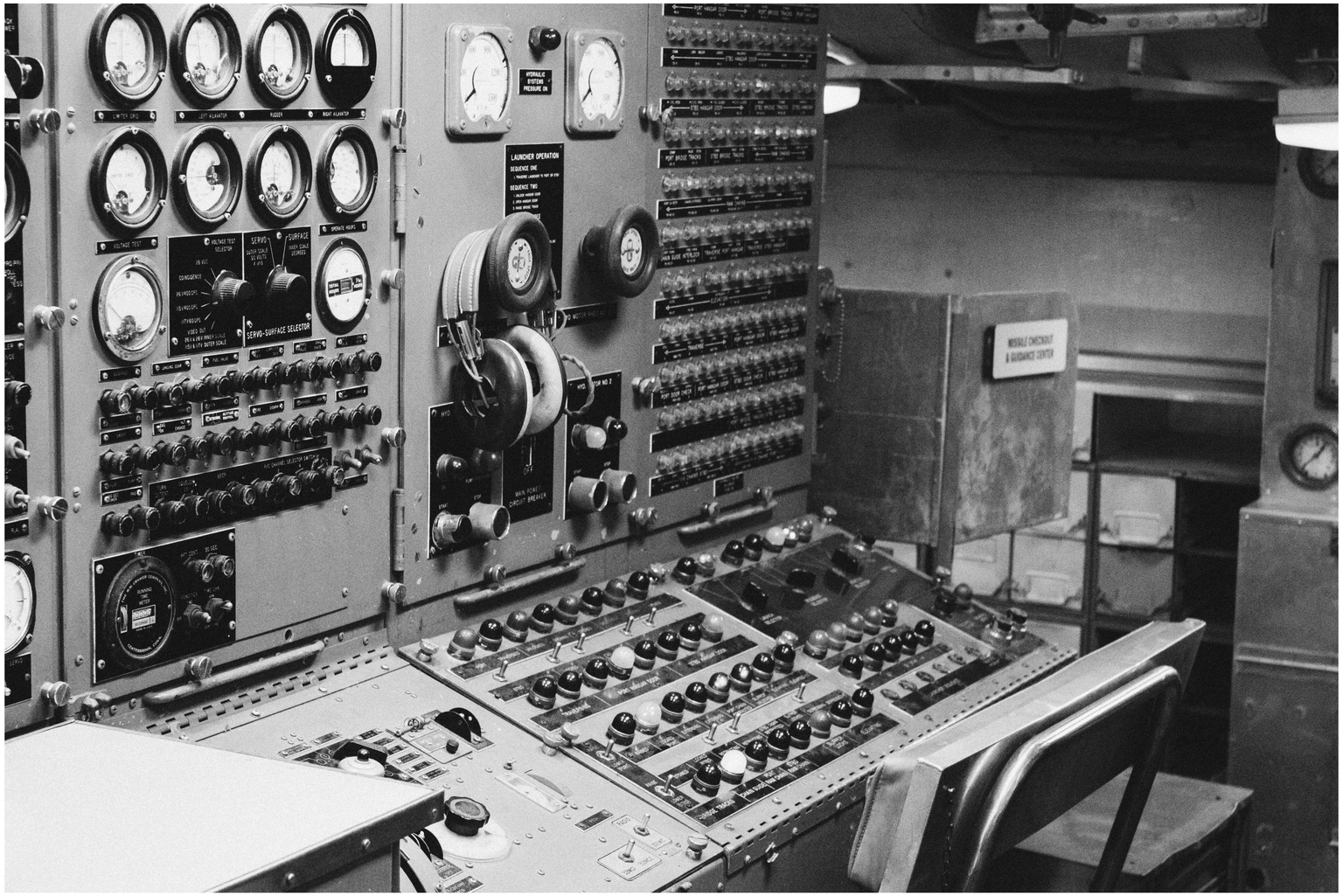
Building Resources via Infrastructure as Code

Beau Peck – Sr. Solutions Engineer
HashiCorp

Provision, secure, connect, and run any infrastructure for any application

What is the industry up to?

The Good Ol’ Days



What is the industry up to?

The golden era of datacenters

Over time…things changed

Business took note of technology and it was good…

Built massive complex datacenters

Servers, networks, programs were deployed and routines were set up to maintain…

What is the industry up to?

Then 2007 happened…

What is the industry up to?

Apps are suddenly important because…I need to share what I had for breakfast

The smartphone, paired with the app made information and data easily and quickly consumable at any time from any where

Systems of engagement were born

Business took note of and suddenly things weren't good enough to influence and catch customers' attention, things needed to happen more quickly

But IT processes were slow…

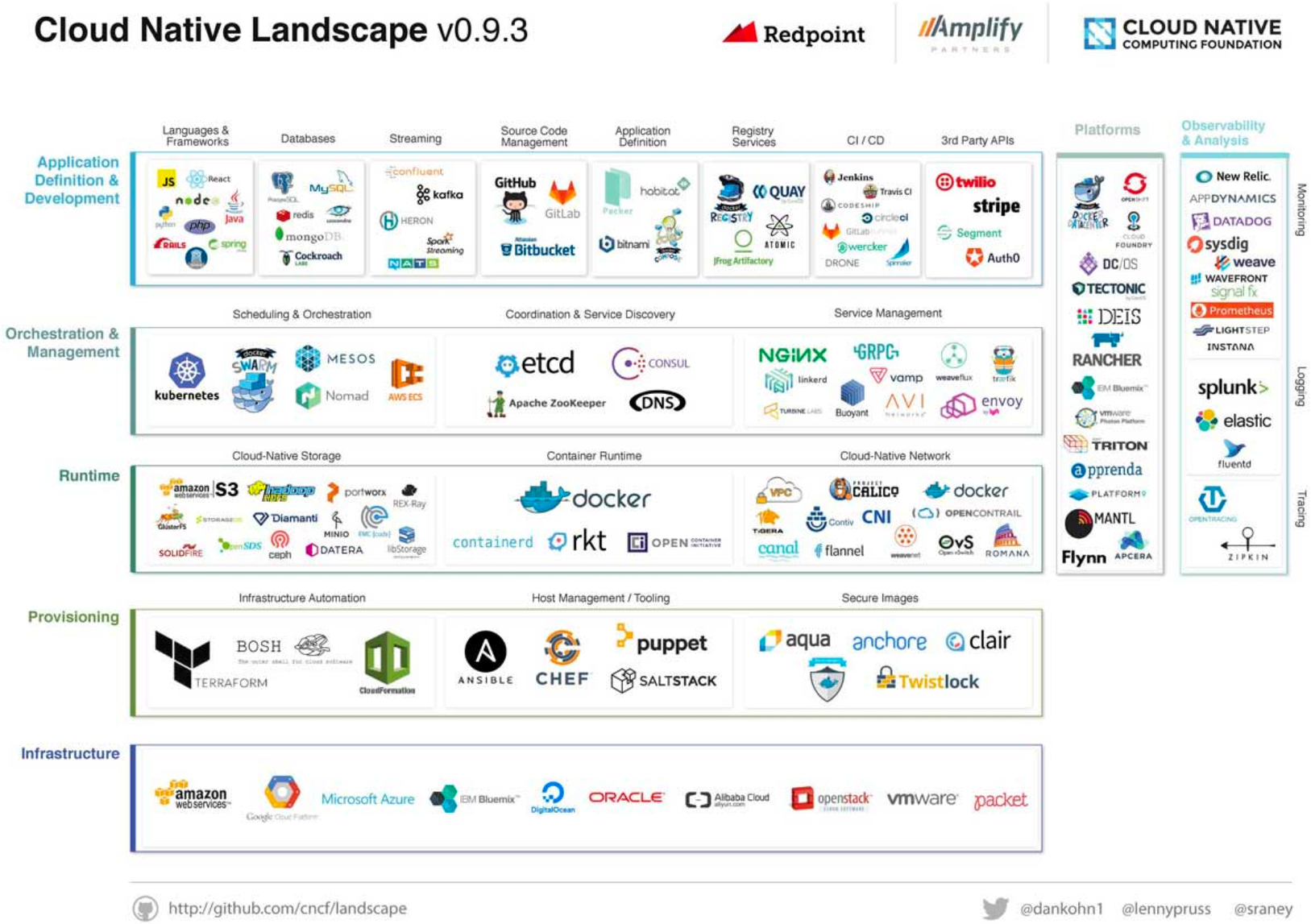
Necessity is the mother of all invention

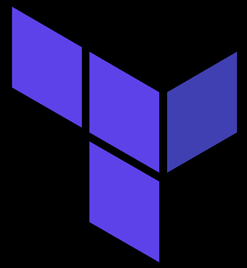
What is the industry up to?

From the good ol’ days to the cloudy days…speed is the new currency

What is the industry up to?

New methods and software to meet the speed of business





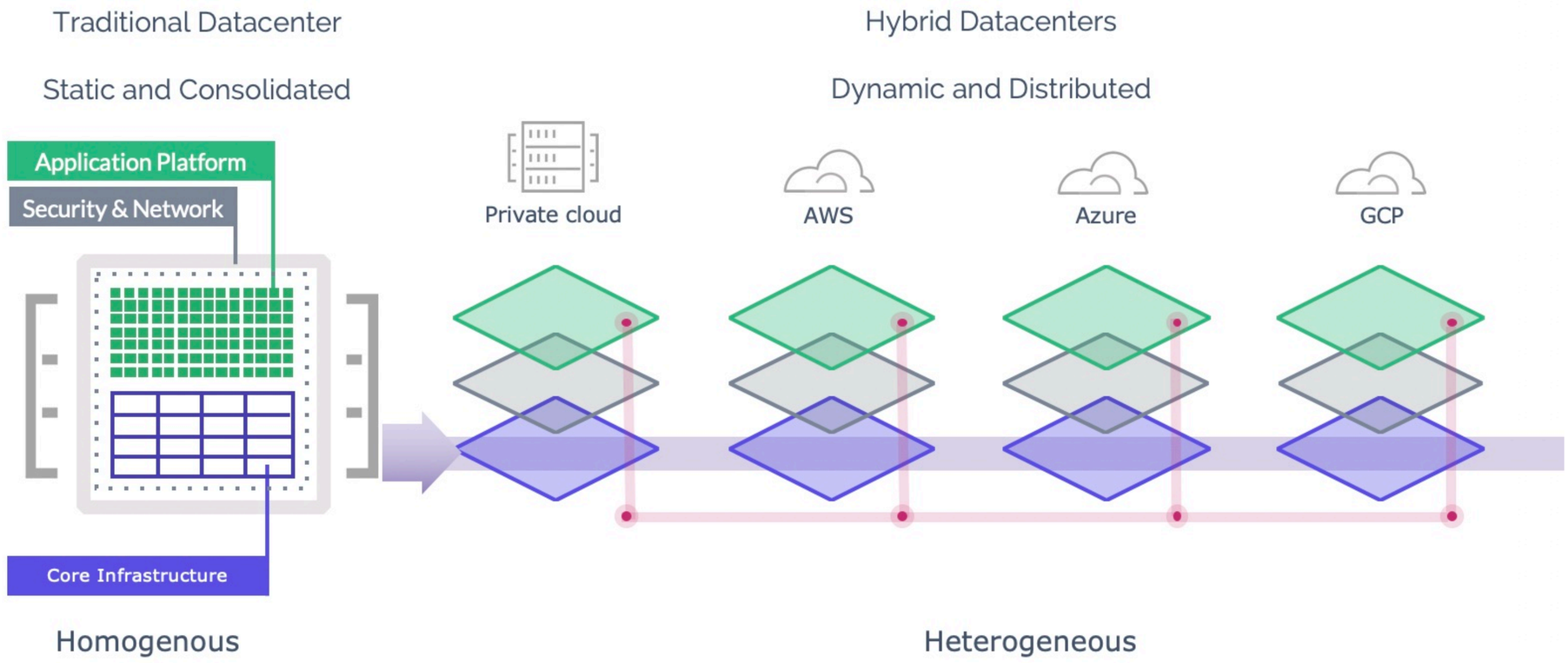
HashiCorp

Terraform

Infrastructure as Code

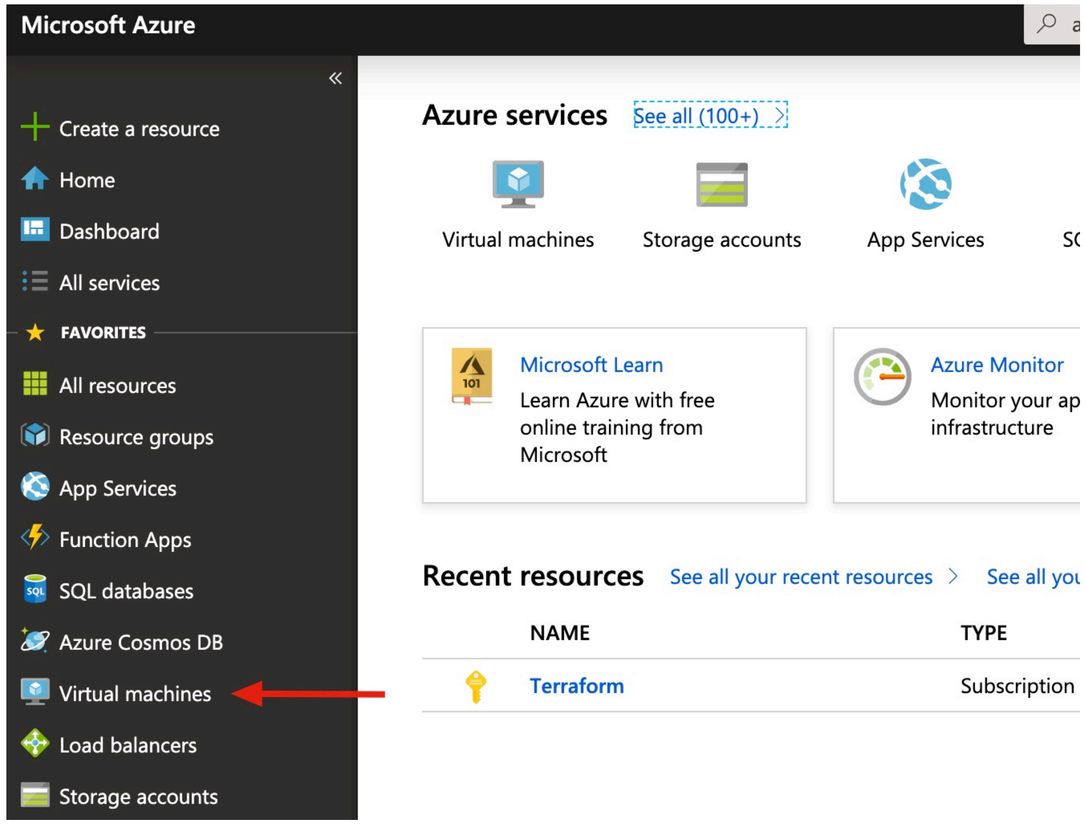
The shift from static to dynamic infrastructure

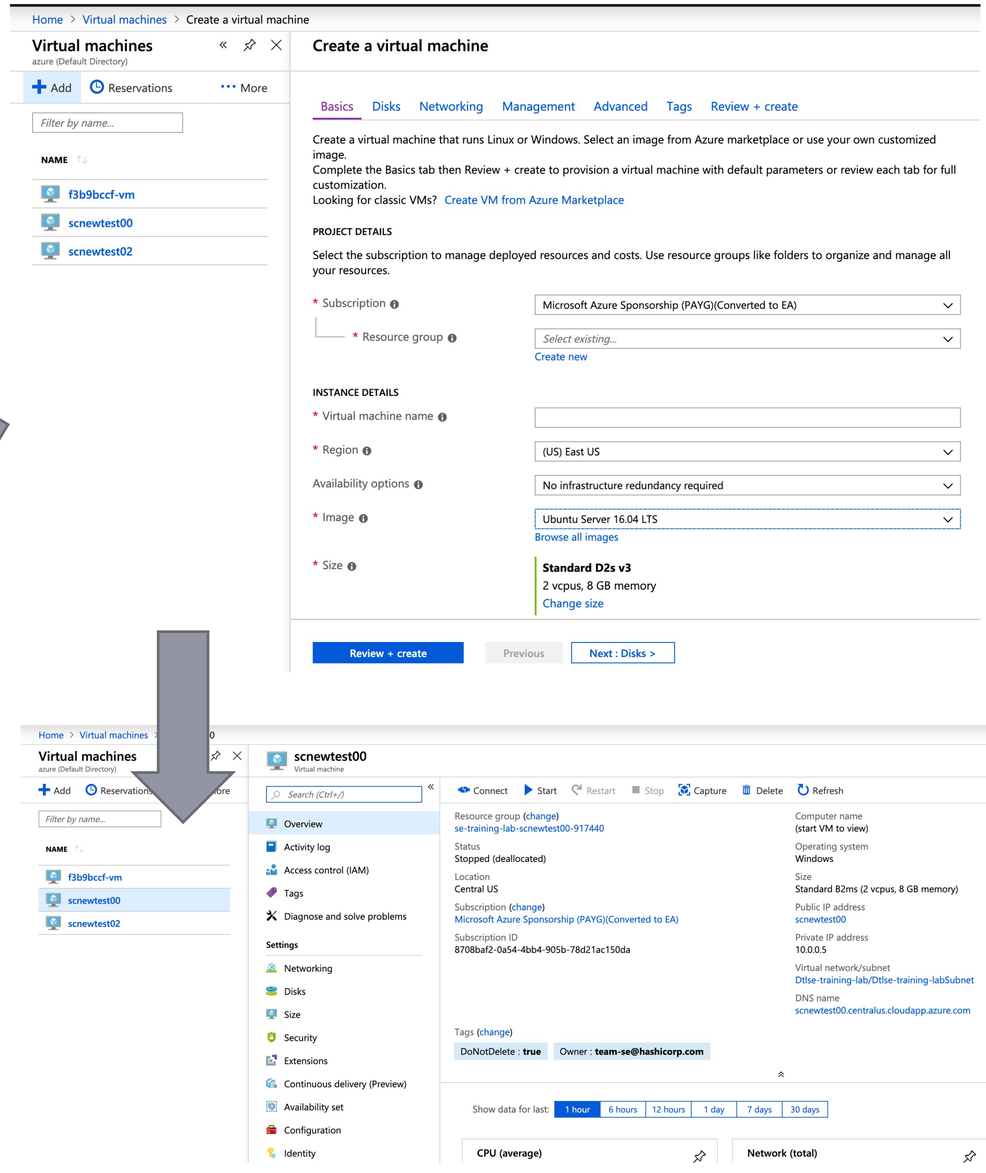
4 planes in the cloud



Typical Deployment Methods

Ol' reliable "Click Ops" – aka use that GUI!





Typical Deployment Methods

We'll knock it up a notch – use the native cloud provisioning tool

Azure / azure-quickstart-templates

Watch 588 | Star 4,996 | Fork 7,243

Code | Issues 555 | Pull requests 24 | Projects 0 | Wiki | Insights

Azure Quickstart Templates https://azure.microsoft.com/en-us/doc...

azure templates arm

22,184 commits | 5 branches | 0 releases | 772 contributors | MIT

Branch: master | New pull request | Create new file | Upload files | Find File | Clone or download

bmoore-msft Merge pull request #5979 from bmoore-msft/cicd ... Latest commit ec48b94 2 days ago

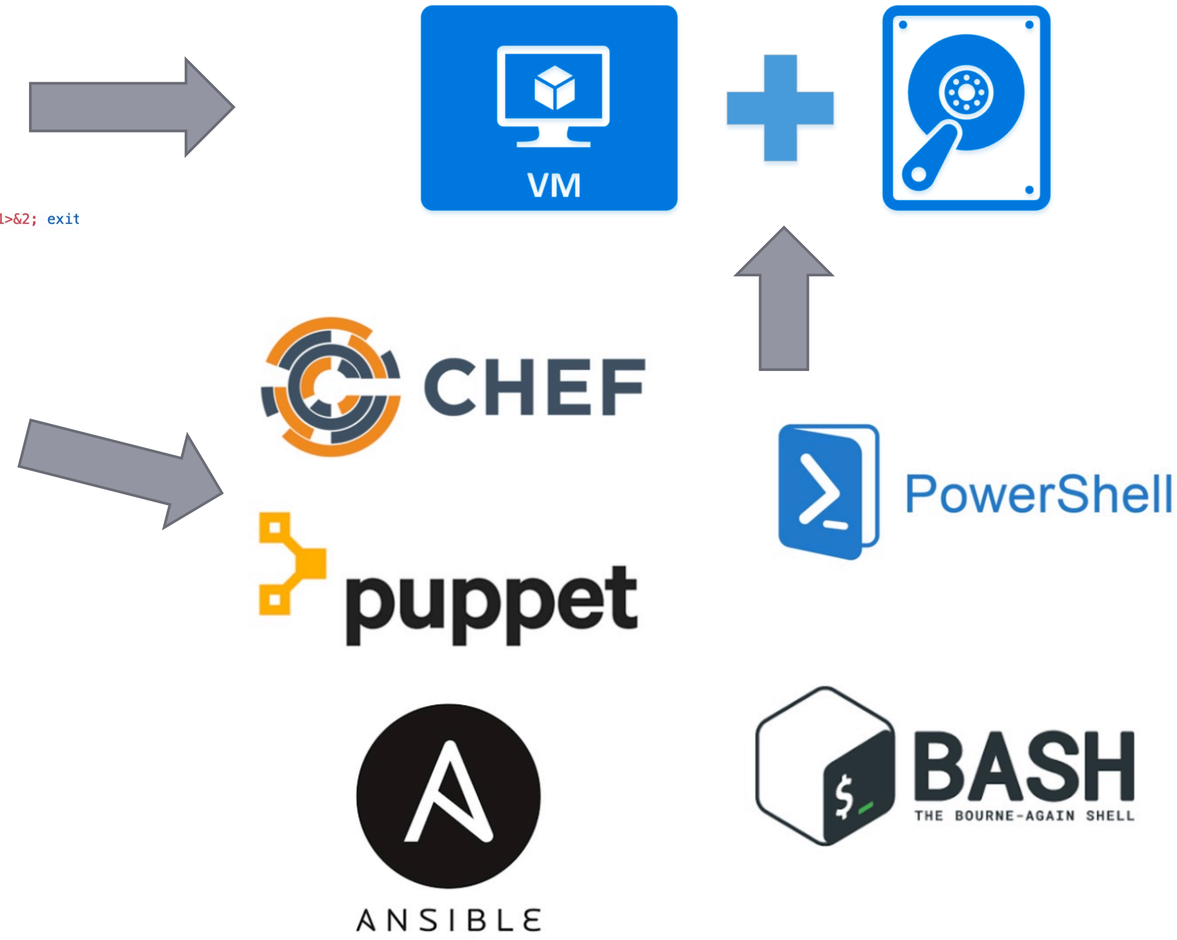
.github	Changed Schema from http to https and 2014 preview to 2015	5 months ago
1-CONTRIBUTION-GUIDE	Update best-practices.md	11 days ago
100-blank-template	Changed Schema from http to https and 2014 preview to 2015	5 months ago
100-marketplace-sample	Changed Schema from http to https and 2014 preview to 2015	5 months ago
101-1vm-2nics-2subnets-1vnet	added type property to metadata.json files	8 months ago
101-AAD-DomainServices	[AUTO-2018-10-05] adding file 101-AAD-DomainServices/azuredeploy.json	3 months ago
101-DDoS-Attack-Prevention	Changed Schema from http to https and 2014 preview to 2015	5 months ago
101-SQL-Injection-Attack-Preventi...	Changed Schema from http to https and 2014 preview to 2015	5 months ago

```
"type": "Microsoft.Compute/virtualMachines",
"apiVersion": "2018-10-01",
"name": "[variables('vmName')]",
"location": "[parameters('location')]",
"dependsOn": [
  "[resourceId('Microsoft.Storage/storageAccounts/', variables('storageAccountName'))]",
  "[resourceId('Microsoft.Network/networkInterfaces/', variables('nicName'))]"
],
"properties": {
  "hardwareProfile": {
    "vmSize": "[variables('vmSize')]"
  },
  "osProfile": {
    "computerName": "[variables('vmName')]",
    "adminUsername": "[parameters('adminUsername')]",
    "adminPassword": "[parameters('adminPasswordOrKey')]",
    "linuxConfiguration": "[if(equals(parameters('authenticationType'), 'password'), json('null'), variables('linuxConfiguration'))]"
  },
  "storageProfile": {
    "imageReference": {
      "publisher": "[variables('imagePublisher')]",
      "offer": "[variables('imageOffer')]",
      "sku": "[parameters('ubuntuOSVersion')]",
      "version": "latest"
    },
    "osDisk": {
      "createOption": "FromImage"
    },
    "dataDisks": [
      {
        "diskSizeGB": 1023,
        "lun": 0,
        "createOption": "Empty"
      }
```

Typical Deployment Methods

Ol’ Zeke is a master scripter – we can blend multiple tools

```
#!/bin/bash
set -euo pipefail
IFS=$'\n\t'

# -e: immediately exit if any command has a non-zero exit status
# -o: prevents errors in a pipeline from being masked
# IFS new value is less likely to cause confusing bugs when looping arrays or arguments (e.g. $@)

usage() { echo "Usage: $0 -i <subscriptionId> -g <resourceGroupName> -n <deploymentName> -l <resourceGroupLocation>" 1>&2; exit

declare subscriptionId=""
declare resourceGroupName=""
declare deploymentName=""
declare resourceGroupLocation=""

# Initialize parameters specified from command line
while getopts ":i:g:n:l:" arg; do
        case "${arg}" in
                i)
                        subscriptionId=${OPTARG}
                        ;;
                g)
                        resourceGroupName=${OPTARG}
                        ;;
                n)
                        deploymentName=${OPTARG}
                        ;;
                l)
                        resourceGroupLocation=${OPTARG}
                        ;;
                esac
done
shift $((OPTIND-1))

#Prompt for parameters is some required parameters are missing
if [[ -z "$subscriptionId" ]]; then
        echo "Subscription Id:"
        read subscriptionId
```



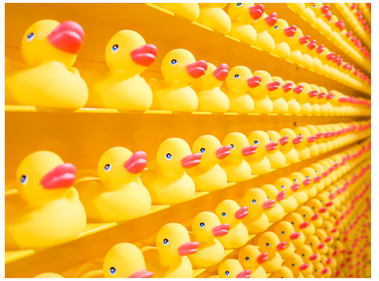
Infrastructure As Code

What is it?

“The process of creating, deploying and managing your infrastructure resources (datacenters) via machine-readable definition files”

- A means to declaratively define a workflow to create your infrastructure
- Gives the ability to build new infrastructure as well as modify existing infrastructure in an automated fashion
- Utilize software development techniques and application code workflows applied to your infrastructure (eg. code review, CI/CD, Git, Azure DevOps)
- Immutable / indempotent

vs.



- Reusable modules – create repeatable standards and remove duplication of effort

Infrastructure As Code

What is it?

Easy to think of it more as **executable documentation**

Benefits

- Infrastructure is documented
- Policies and security requirements are enforced
- You enable cross team collaboration
- Remove the back and forth / work-stop-work-stop churn

Terraform Deployment Method

Abstraction and automation through codification – read / share / collaborate

- Uses a DSL known as HCL (HashiCorp Configuration Language)

- Workflows, not technology – use a consistent workflow to provision infrastructure while preserving uniqueness of each provider – work by declarative state: **init** / **plan** / **apply** / **destroy**

- Operate on state – know what's there and should be there / don't deviate

- Support for over 160 providers – 80 HashiCorp supported providers / 80+ community providers – write your own provider

```
resource "azurerm_resource_group" "vaultworkshop" {
name = "${var.prefix}-vault-workshop"
location = "${var.location}"
}

resource "azurerm_virtual_network" "vnet" {
name = "${var.prefix}-vnet"
location = "${azurerm_resource_group.vaultworkshop.location}"
address_space = ["${var.address_space}"]
resource_group_name = "${azurerm_resource_group.vaultworkshop.name}"
}

resource "azurerm_subnet" "subnet" {
name = "${var.prefix}-subnet"
virtual_network_name = "${azurerm_virtual_network.vnet.name}"
resource_group_name = "${azurerm_resource_group.vaultworkshop.name}"
address_prefix = "${var.subnet_prefix}"
```

How Terraform forms terra

in other words…to the code