Visual Studio LIVE! EXPERT SOLUTIONS FOR .NET DEVELOPERS

ROCK YOUR CODE TOUR · 2017 | CHICAGO

What's New in Azure v2

Eric D. Boyd
Founder & CEO, responsiveX

Level: Intermediate

About Me

responsiveX

eric.boyd@responsiveX.com

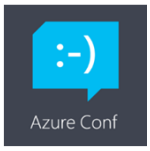
MVP Microsoft® Most Valuable Professional

Microsoft® Regional Director

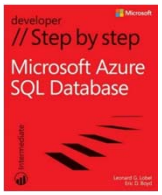
Visual Studio LIVE! EXPERT SOLUTIONS FOR .NET DEVELOPERS

I...

SPEAK AT LOTS OF CONFERENCES



WRITE OCCASIONALLY



TRY TO BE SOCIAL ON TWITTER

@EricDBoyd

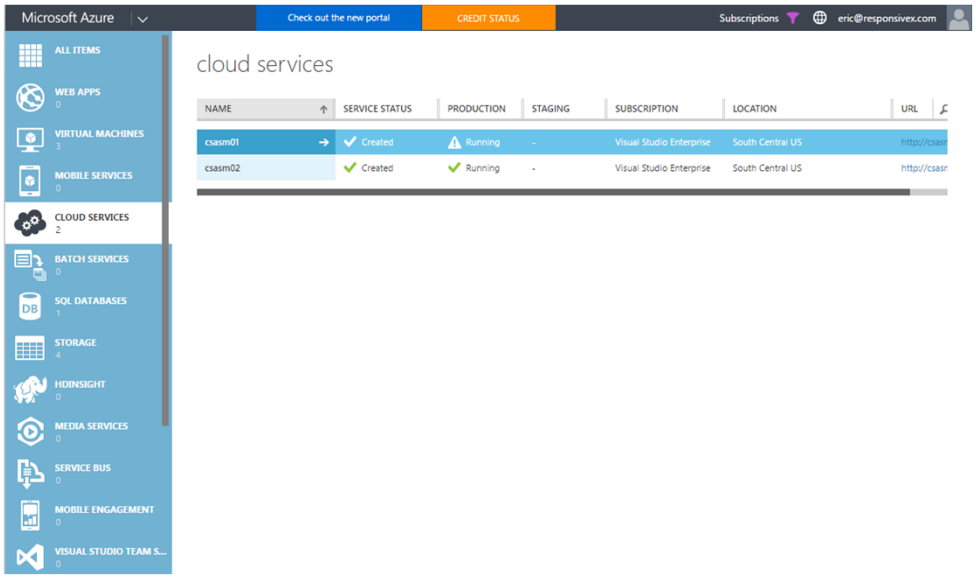
Visual Studio LIVE!
EXPERT SOLUTIONS FOR .NET DEVELOPERS

Agenda

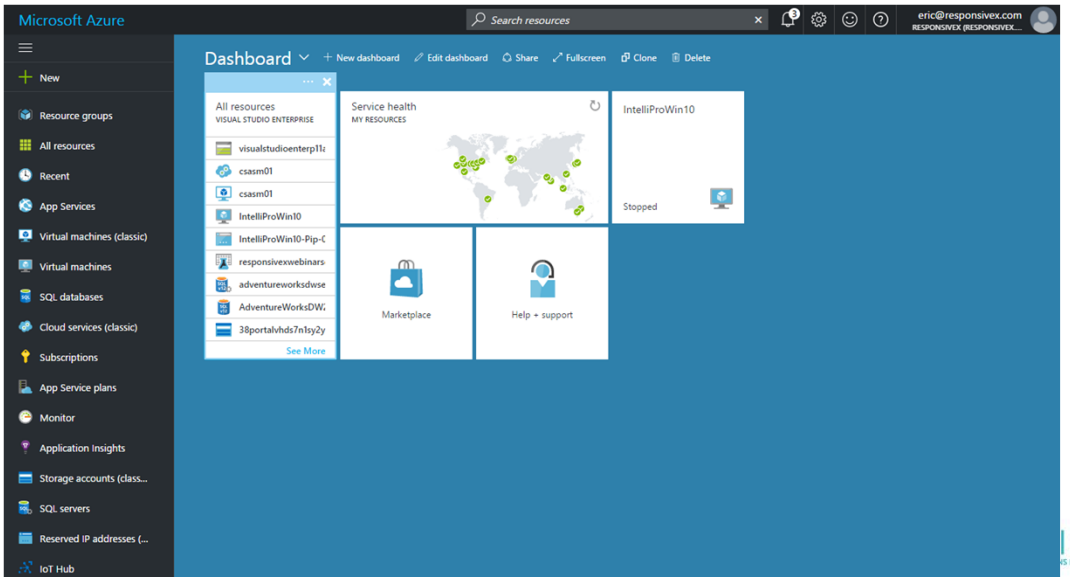
- What is Azure v1? What is Azure v2?
- Take a step back and discuss SDLC/ALM
  - Maturity
  - Challenges
  - DevOps
- Deployment Automation Framework
- Deep Dive Azure v2 and Azure Resource Manager

Visual Studio LIVE!
EXPERT SOLUTIONS FOR .NET DEVELOPERS

What is Azure v1?



What is Azure v2?

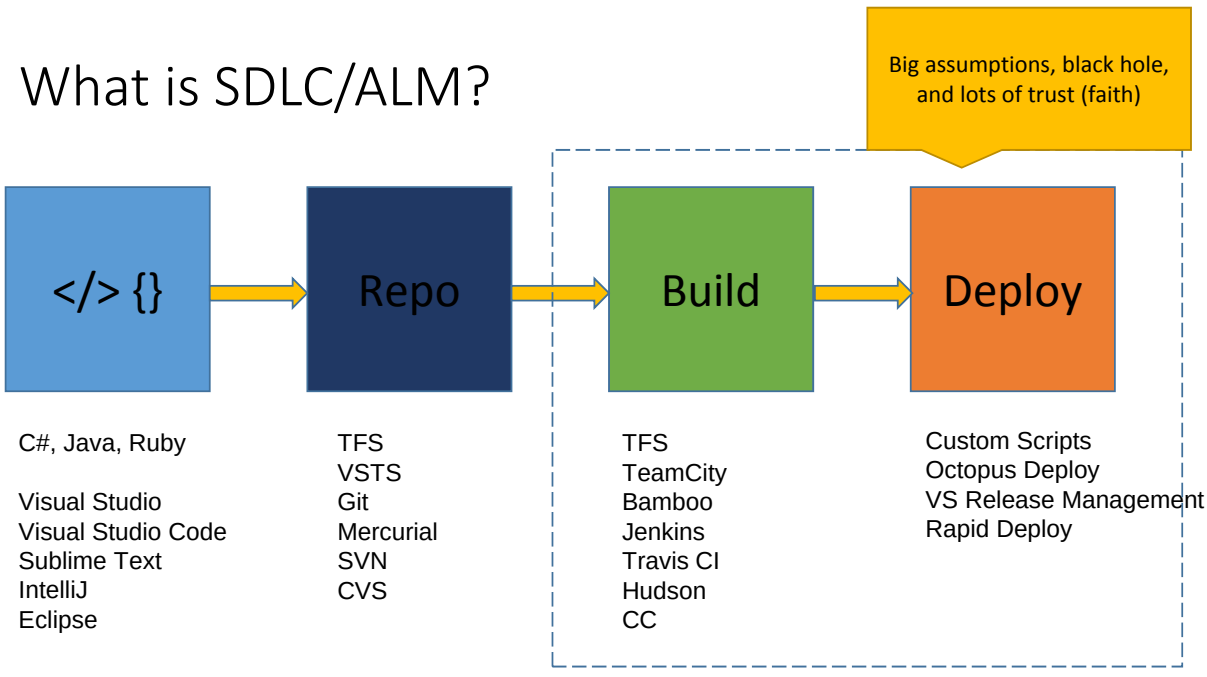


Why should I care?

- Innovation
- Deployment
- Access Control
- Billing

Visual Studio LIVE!
EXPERT SOLUTIONS FOR .NET DEVELOPERS

What is SDLC/ALM?



Why don't we treat infrastructure the same as software?

- It's a one time thing and not worth the investment
- They aren't programmers, they configure servers in GUIs and tools
- The light bulb moment just hasn't occurred

Visual Studio LIVE!
EXPERT SOLUTIONS FOR .NET DEVELOPERS

Infrastructure Automation

Services
Servers
Azure
On-Prem

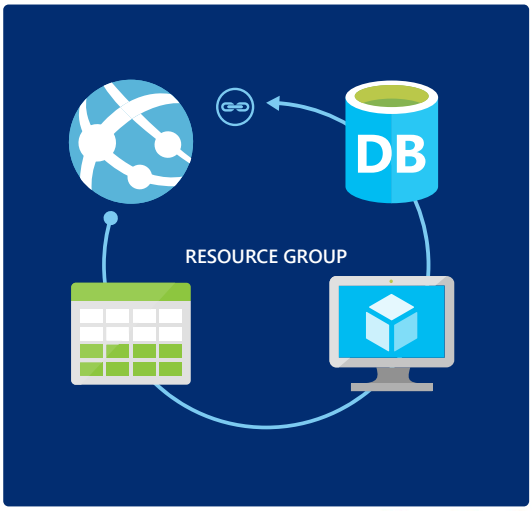
Custom Scripts
PowerShell DSC
Chef
Puppet
Ansible
Salt

Custom Scripts
Octopus Deploy
VS Release Management
Rapid Deploy

Visual Studio LIVE!
EXPERT SOLUTIONS FOR .NET DEVELOPERS

Azure Resource Groups

- → Collection of resources with a similar lifetime
- → Every resource belongs to a resource group
- → Resources have types, defined by resource providers
- → RBAC integration
- → Declarative model driven deployment
- → Consistent



Visual Studio LIVE!
EXPERT SOLUTIONS FOR .NET DEVELOPERS

How Do I Get to Azure v2?

- ASM2ARM
  - https://github.com/fullscale180/asm2arm
- MigAz
  - https://github.com/Azure/classic-iaas-resourcemanager-migration/tree/master/migaz
- Azure Site Recovery (ASR)
  - https://docs.microsoft.com/en-gb/azure/site-recovery
- Move-Azure* PowerShell Cmdlets
  - https://docs.microsoft.com/en-us/azure/virtual-machines/virtual-machines-windows-ps-migration-classic-resource-manager

Visual Studio LIVE!
EXPERT SOLUTIONS FOR .NET DEVELOPERS

Azure Resource Manager (ARM)

Consistent management layer

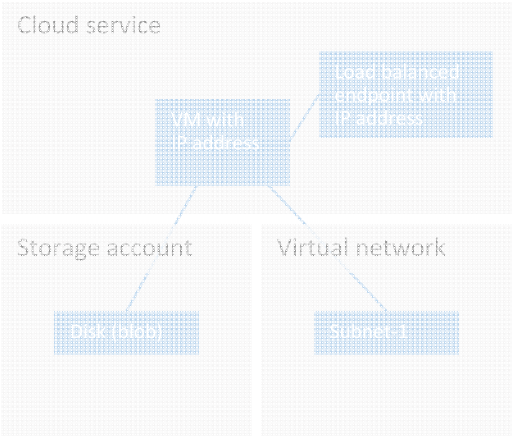
Tools

Curated extensions

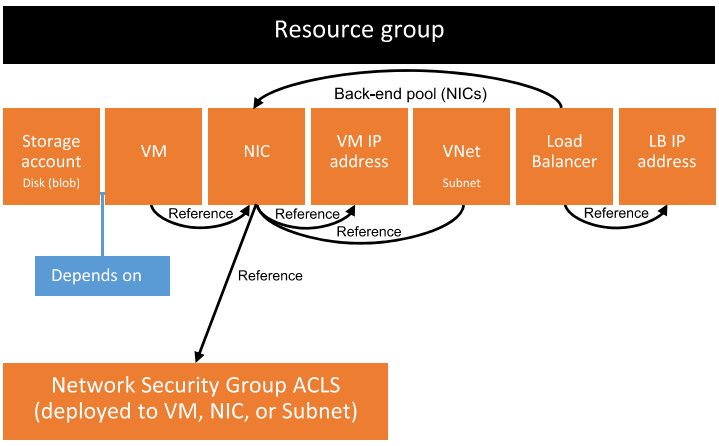
Provider rest points

Resource Manager example

Classic model (v1)



Resource Manager (v2)



imperative or declarative

```
New-AzureRmStorageAccount –Name $acct
New-AzureRmNetworkInterface –Name
New-AzureRmVirtualNetwork
New-AzureRmVM –VM $myVM
```

```
{
 "$schema": "https://../deploymentTemplate.json#",
 "contentVersion": "1.0.0.0",
 "parameters": {},
 "variables": {},
 "resources": [],
 "outputs": {}
}
```

Getting started with Azure templates

Wide range of Quickstart templates

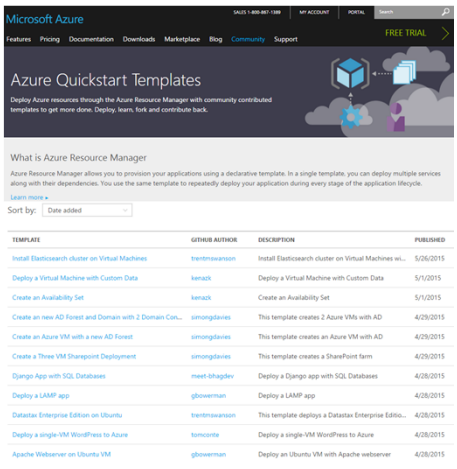
Indexed on Azure.com

GitHub repo

Community and Microsoft contributed

Integration of IaaS with Azure Services

Many examples available @ https://github.com/Azure/azure-quickstart-templates



Demo
Check out ARM Templates

JSON files—simpler than they look

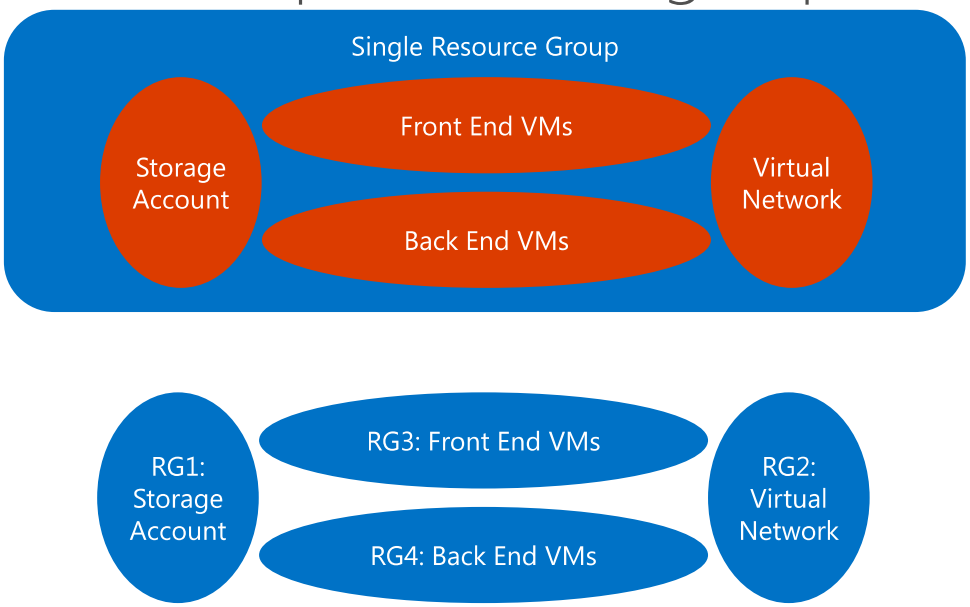
Schema, content version, parameters, variables, resources, and outputs

Passing state—common parameters

Name (string values)	Description
Location	The location where the resources will be deployed from a constrained list of Azure regions
storageAccountNamePrefix	Unique DNS name for the storage account where the VM's disks will be placed
virtualNetworkName	For deployments that create a new virtual network, the name to use for creating that resource. For deployments that use an existing virtual network, the name of the VNet to deploy into
username	User name for the virtual machine(s) and potentially the application(s). More than one user name can be requested from the end user, but at least one must be prompted
password	Password for the virtual machine(s) and potentially the application(s). More than one password can be requested from the end user for different VMs or applications, but at least one must be prompted
tshirtSize	The named scale unit size to provision from a constrained list of offered t-shirt sizes For example, "Small", "Medium", "Large"
enableJumpbox	Parameter that identifies whether to enable a jumpbox for the environment Values: "enabled", "disabled"

Demo
Dev Tools for ARM

Single or multiple resource groups?



Azure Resource Manager templates

ARM templates can:

Simplify deployment

Simplify roll-back

Provide cross-resource configuration and update support

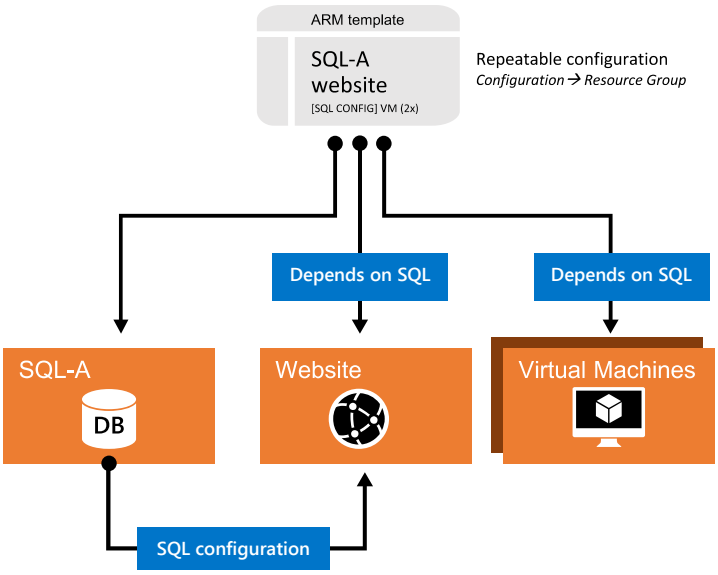
Be used as a learning tool to build to suit

Azure templates are:

Source file, checked-in

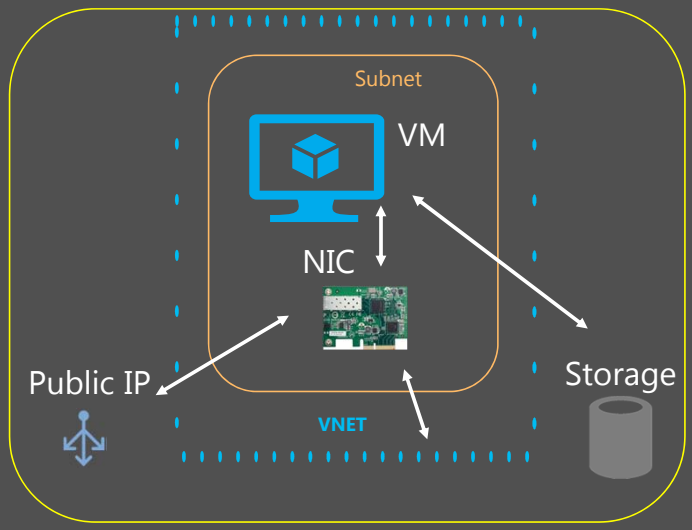
Specifies resources and dependencies (VMs, websites, DBs) and connections (configuration, LB sets)

Configurable parameters for input/output



Azure Resource Manager: Building a Virtual Machine

Resource Group



virtualMachine
- hardwareProfile
- osProfile
- storageProfile
- networkProfile

networkInterface
- privateIPAllocationMethod

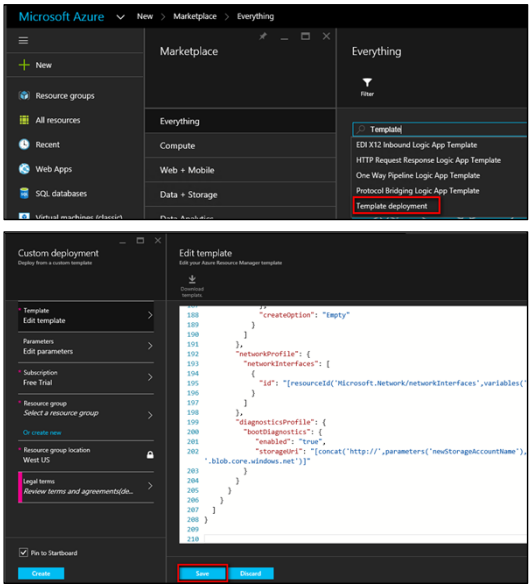
storageAccount
- accountType

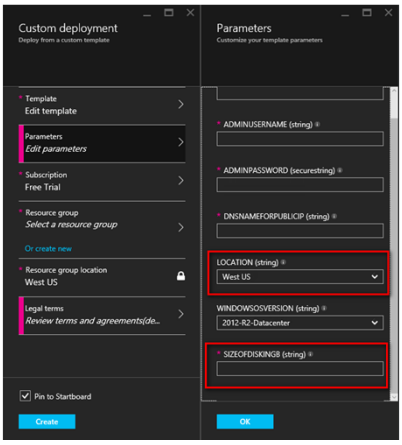
publicIPAddress
- allocationMethod
- domainNameLabel

virtualNetwork
- addressSpace
- Subnet
  - addressPrefix

https://github.com/Azure/azure-quickstart-templates/tree/master/101-simple-windows-

Deploying custom JSON files



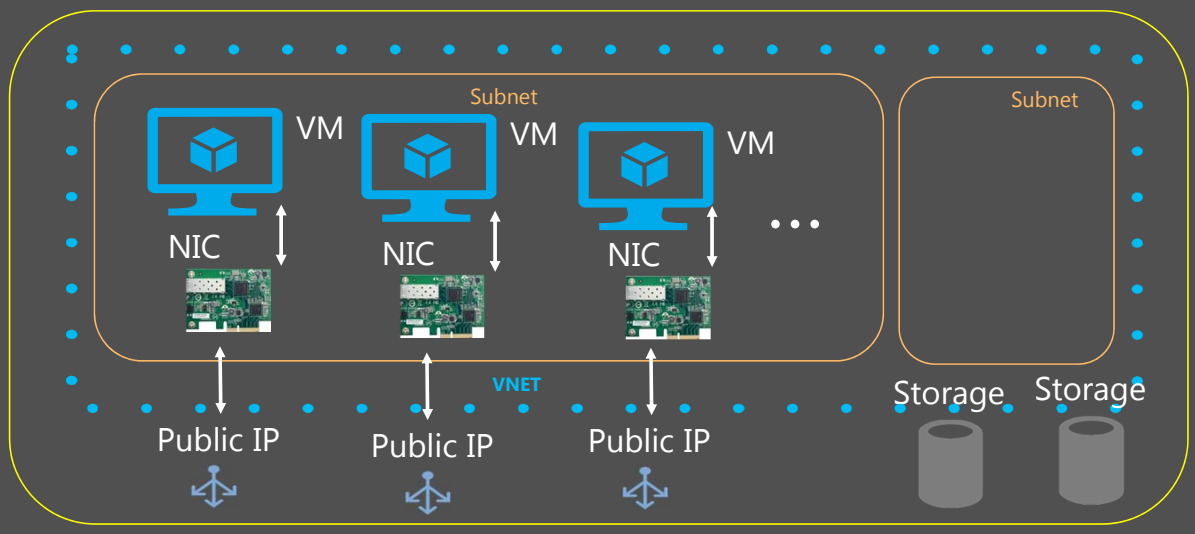



```
New-AzureResourceGroupDeployment -DeploymentName
"Simple-VM" -ResourceGroupName
RG-AZITCAMP -TemplateFile "C:\GitHub\Templates\101-
simple-windows-vm\azuredeploy.json
```

Demo
Deploy It

Go Deeper with ARM

Resource Loops: Declare multiple resources



https://github.com/Azure/azure-quickstart-templates/blob/master/resource-loop-vms-vnet/azuredeploy.json

Resource Loops: copy() / copyIndex()

For each resource, use copy to set the number

1

```
{
  "apiVersion": "2015-05-01-preview",
  "type": "Microsoft.Network/networkInterfaces",
  "name": "[concat('nic', copyindex())]",
  "location": "West US",
  "copy": {
      "name": "nicLoop",
      "count": "10"
  },
  ...
```

Use copyIndex to create the names.

2

```
{
    "apiVersion": "2015-05-01-preview",
    "type": "Microsoft.Compute/virtualMachines",
    "name": "[concat('myvm', copyIndex())]",
    "location": "West US",
    "copy": {
       "name": "virtualMachineLoop",
       "count": "10"
    },
    "dependsOn": [
        "[concat('Microsoft.Network/networkInterfaces/', 'nic', copyindex())]",
        "[concat('Microsoft.Storage/storageAccounts/', parameters('newStorageAccountName'))]"
    ],
```

Template language expressions

- base64encode('stringtoencode')
- concat('string','to','encode')
- copyIndex(offset)
- listKeys(storageAccountResourceId, apiVersion)
- padLeft(stringToPad,targetLength,paddingCharacter)
- parameters('parameterName')
- providers(namespace, resourceType)
- reference(resourceId,apiVersion)
- resourceGroup()
- resourceId('namespace/resourceType', 'resourceName')
- subscription()
- variables('variables')

Passing state in and out of templates

Multiple types of state

- Parameters
- Static variables
- Dynamic variables

Templates accept parameters

Templates return variables as output values

Simple or complex object types are supported

Passing state—complex objects

- Easier to pass a number of related values with a single variable
- Object.Property approach provides additional context when reading the template

Examples

```
"networkSettings": {
  "vnetName": "[parameters('virtualNetworkName')]",
  "addressPrefix": "10.0.0.0/16",
  "subnets": {
   "dmz": {
    "name": "dmz",
    "prefix": "10.0.0.0/24",
    "vnet": "[parameters('virtualNetworkName')]"
   },
   "data": {
    "name": "data",
    "prefix": "10.0.1.0/24",
    "vnet": "[parameters('virtualNetworkName')]"
   }
  }
}
```

```
"osSettings": {
  "imageReference": {
   "publisher": "Canonical",
   "offer": "UbuntuServer",
   "sku": "14.04.2-LTS",
   "version": "latest"
  }
```

```
"availabilitySetSettings": {
  "name": "pgsqlAvailabilitySet",
  "fdCount": 3,
  "udCount": 5
 }
```

```
"tshirtSizeSmall": {
  "vmSize": "Standard_A1",
  "diskSize": 1023,
  "vmTemplate": "[concat(variables('templateBaseUrl'),
'database-2disk-resources.json')]",
  "vmCount": 2,
  "storage": {
   "name":
"[parameters('storageAccountNamePrefix')]",
   "count": 1,
   "pool": "db",
   "map": [0,0],
   "jumpbox": 0
  }
 },
```

Passing state—output variables

A template can return values to its caller via the outputs section

```
"outputs": {
"masterip": {
"value":
"[reference(concat(variables('nicName'),0)).ipConfigurations[0].properties.privateIPAddress]",
"type":"string"
}}
```

These values can then be used by the caller

```
"masterIpAddress": {
"value":
"[reference('master-node').outputs.masterip.value]"
 } }
```

Control flow

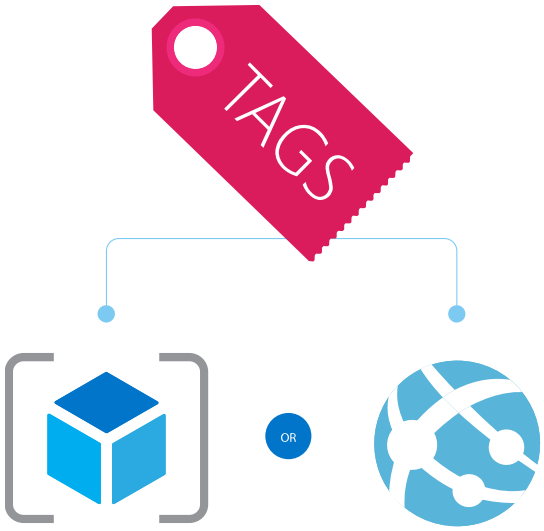
No control flow logic built into ARM template language

An approach with parameters, variables, and linked templates

- Use provides parameter value that provides context, e.g., tshirtSize parameter is passed in as a value of 'small'
- Using concat and a pre-defined variable, a new variable value is created which points to the specific , e.g., 'tshirtSize-small.json'
- Template linking is incorporated into the template and uses this new value to identify which template to deploy.
- Common examples are "tshirt sizes" and optional features for a deployment, e.g., "enableJumpbox"

Other ARM Benefits

Resource Tags



- Name-value pairs assigned to resources or groups
- Subscription-wide taxonomy
- Each resource can have up to 15 tags

Tagging Tips

- Notes: Simple note for VM
- Creator: track the "owner" of a VM
- Department/Cost center: who pays
- Environment: production vs. pre-production vs. test

Access Control: RBAC

What is RBAC

- allows secure access with granular permissions to resources
- assignable to users, groups or service principals
- built-in roles make it easy to get started

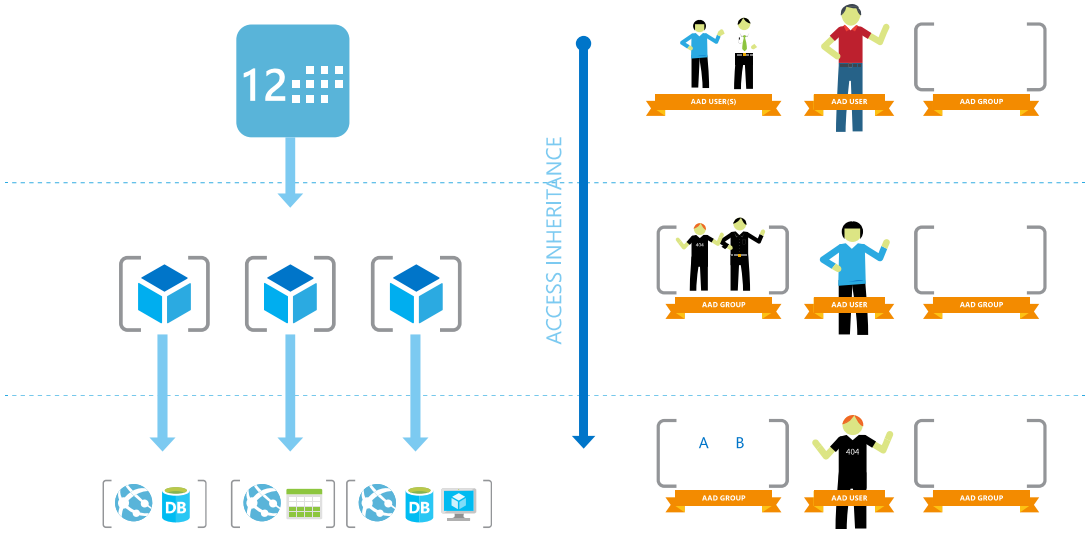
Role Definitions

- describes the set of permissions (e.g. read actions)
- can be used in multiple assignments

Role Assignments

- associate role definitions with an identity (e.g. user/group) at a scope (e.g. resource group)
- always inherited – subscription assignments apply to all resources

Role Based Access Control



Granular Scopes

/subscriptions/{id}/resourceGroups/{name}/providers/.../virtualmachines/{vmname}

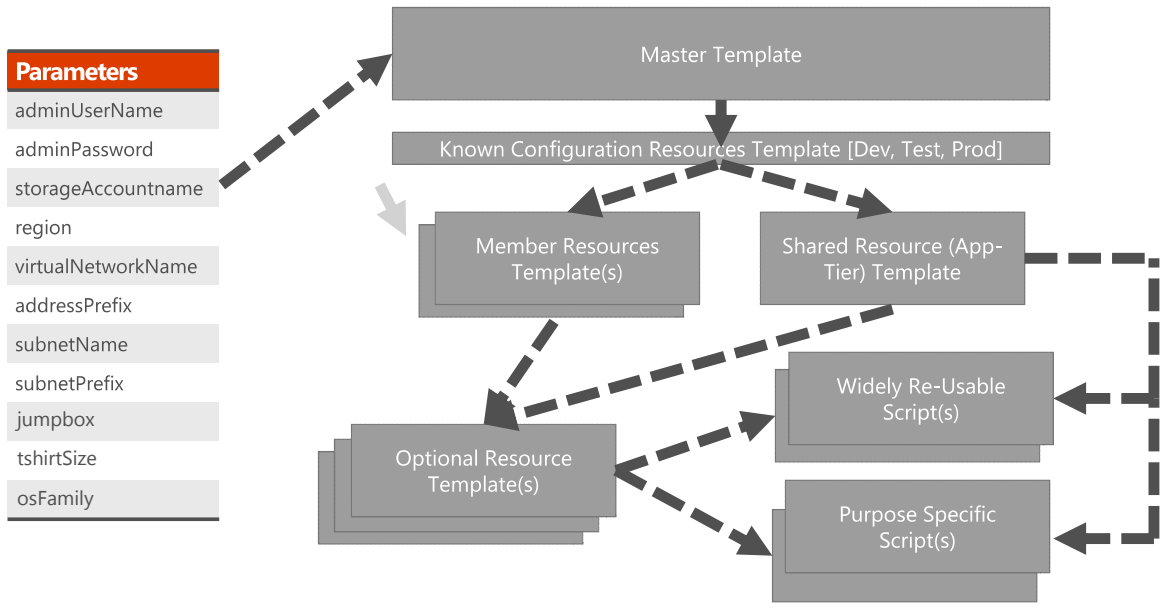
subscription level – grants permissions for all resources in the sub

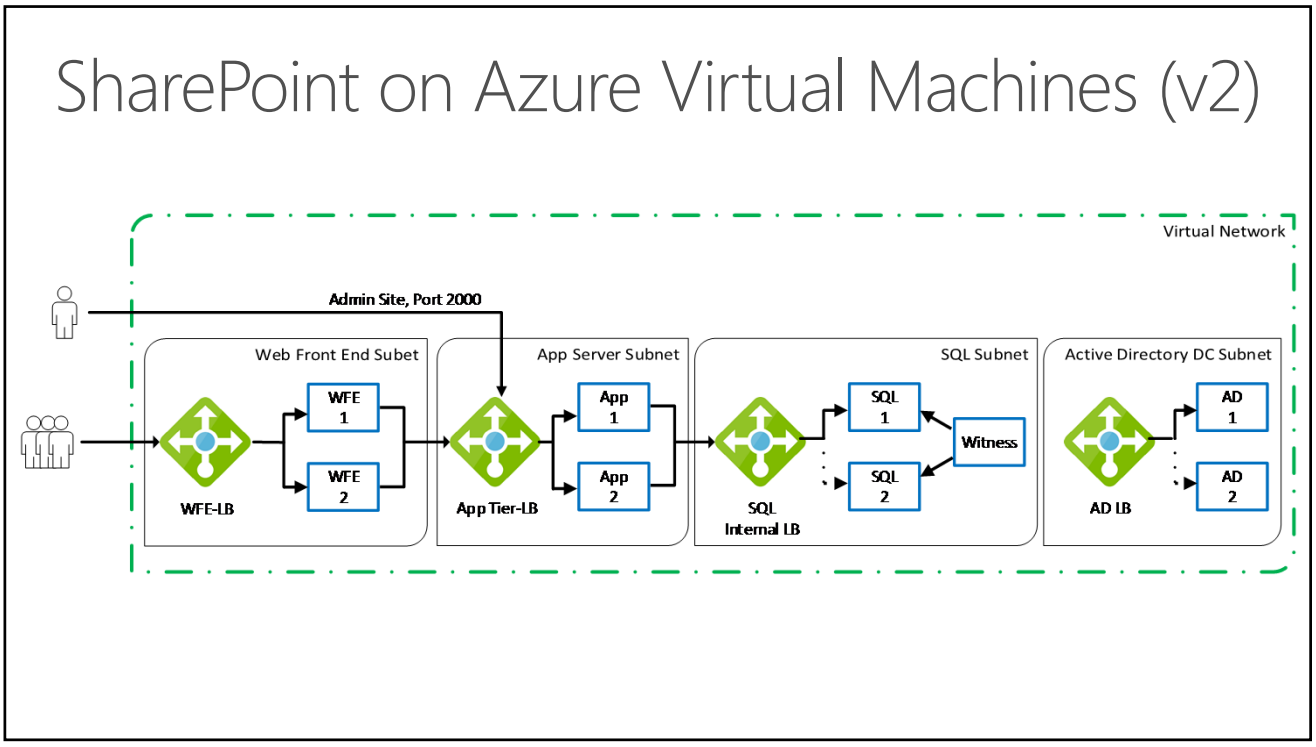
resource group level – grants permissions for all resources in the group

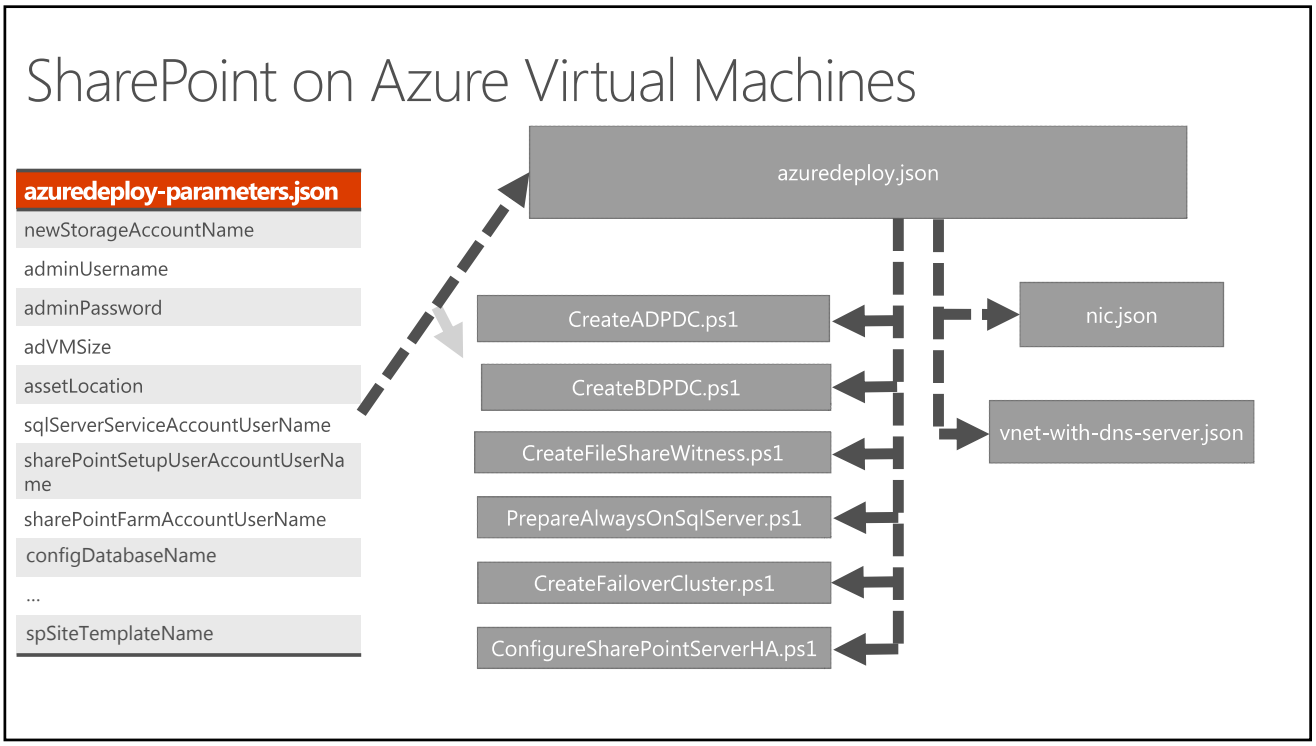
resource level – grants permissions to the specific resource

Advanced Deployments

Architecting Complex Applications using Templates




SharePoint on Azure Virtual Machines (v2)
Virtual Network
Admin Site, Port 2000
Web Front End Subet
App Server Subnet
SQL Subnet
Active Directory DC Subnet
WFE 1
WFE 2
WFE-LB
App 1
App 2
App Tier-LB
SQL 1
SQL 2
Witness
SQL Internal LB
AD 1
AD 2
AD LB


SharePoint on Azure Virtual Machines
azuredeploy-parameters.json
newStorageAccountName
adminUsername
adminPassword
adVMSize
assetLocation
sqlServerServiceAccountUserName
sharePointSetupUserAccountUserName
sharePointFarmAccountUserName
configDatabaseName
...
spSiteTemplateName
azuredeploy.json
CreateADPDC.ps1
CreateBDPDC.ps1
CreateFileShareWitness.ps1
PrepareAlwaysOnSqlServer.ps1
CreateFailoverCluster.ps1
ConfigureSharePointServerHA.ps1
nic.json
vnet-with-dns-server.json

Resources

Azure Resource Manager Schemas
https://github.com/Azure/azure-resource-manager-schemas/

VS Code Tools
https://marketplace.visualstudio.com/items?itemName=msazurermtools.azurerm-vscode-tools

Azure Rest API Explorer
https://resources.azure.com

Template Visualizer
http://armviz.io/

ARM Examples
https://github.com/rjmax/ArmExamples

Visual Studio LIVE!
EXPERT SOLUTIONS FOR .NET DEVELOPERS

Questions

eric.boyd@responsiveX.com

responsiveX

MVP Microsoft® Most Valuable Professional

Microsoft® Regional Director

Visual Studio LIVE!
EXPERT SOLUTIONS FOR .NET DEVELOPERS