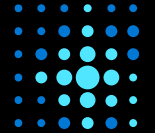


Azure Training Day

Migrating web applications to Azure



Azure VM lift and shift

Part 2 of 5 in the Migrate web apps to Azure series

About us...

Ryan Berry

 Principal Cloud Solution Architect

Matthew Calder (macalde@microsoft.com)

 Sr. Product Marketing Manager – Azure, US Marketing and Operations

For questions or help with this series

MSUSDev@Microsoft.com

For the lab guides and sample code

<https://github.com/MSUSDEV/Migrating-web-apps-to-Azure>



Efficiently Deploying and Migrating Virtual Machines to Azure

Lift & Shift Migration

Infrastructure as Code (IAC)



What is Infrastructure As Code (IAC)

[http://en.Wikipedia.org/
wiki/
Infrastructure_as_
Code](http://en.Wikipedia.org/wiki/Infrastructure_as_Code)

- The process of **managing and provisioning** computer data centers through machine-readable **definition files**.
- The deployment can use **either scripts or declarative definitions**, rather than manual processes.
- While the terminology points to Infrastructure, it **should not be confused with Infrastructure as a Service (IaaS)**, as IAC also allows you to deploy other (cloud) components like Platform as a Service (PaaS)

Infrastructure As Code (IAC) - Values

1. FASTER EXECUTION

- Build once – Deploy many
- Minor changes required to deploy different models
- Deployment of compute resources goes faster than manual provisioning

3. REDUCE COST

- Manual labor is expensive
- Easily scale out your environment
- Portable across different environments (dev/test/staging/different organizations)

2. REDUCE RISK

- Remove Errors and Mistakes
- Remove Security Violations
- Overwrite changes without touching existing deployment state

4. INTEGRATION WITH DEVOPS

- Infrastructure as Code provides an integration with DevOps concepts and processes within an organization
- Allows for end-to-end application landscape provisioning, not just 'infrastructure'

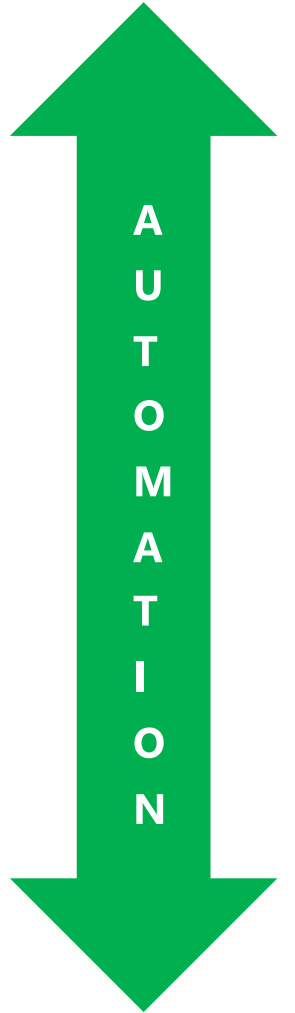
Infrastructure As Code (IAC) - Methods

DECLARATIVE (FUNCTIONAL) "WHAT"

- The final state of the system / environment is defined (declared), in such a way that it defines in "what" state it should be.
- When the process is run, it will configure the system/environment to have the declared state as the final result.

IMPERATIVE (PROCEDURAL) "HOW"

- The automation code used to setup or configure the system / environment is written in such a way, it goes through each configuration step-by-step.
- Automation code built in this way, defines the process of "how" the system is to be configured and what steps need to be taken, in the exact order, to obtain the final result state.



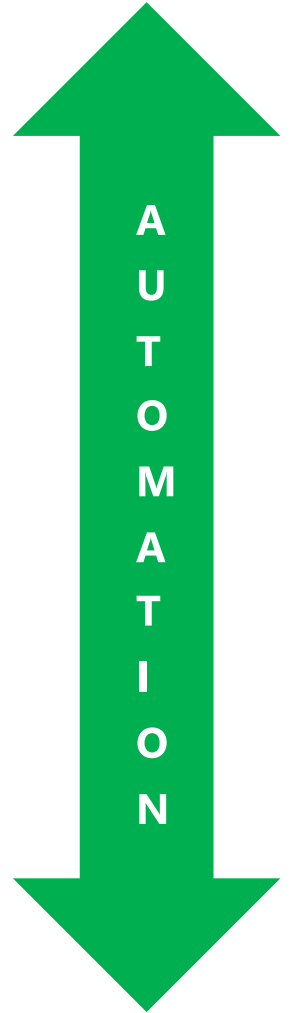
Infrastructure As Code (IAC) – Configuration Options

CONFIGURATION ORCHESTRATION

- Designed and used to automate the deployment of servers and other related (cloud) infrastructure
- Some tools have overlap with configuration management

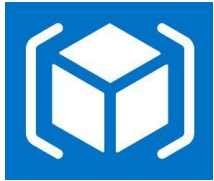
CONFIGURATION MANAGEMENT

- Designed and used to automate the configuration of systems and software on top of the infrastructure which has already been provisioned (out of configuration orchestration)
- Some tools have overlap with configuration Orchestration



Infrastructure as Code – Tools (Azure Popular)

CONFIGURATION ORCHESTRATION



HashiCorp
Terraform



ANSIBLE

CONFIGURATION MANAGEMENT



puppet



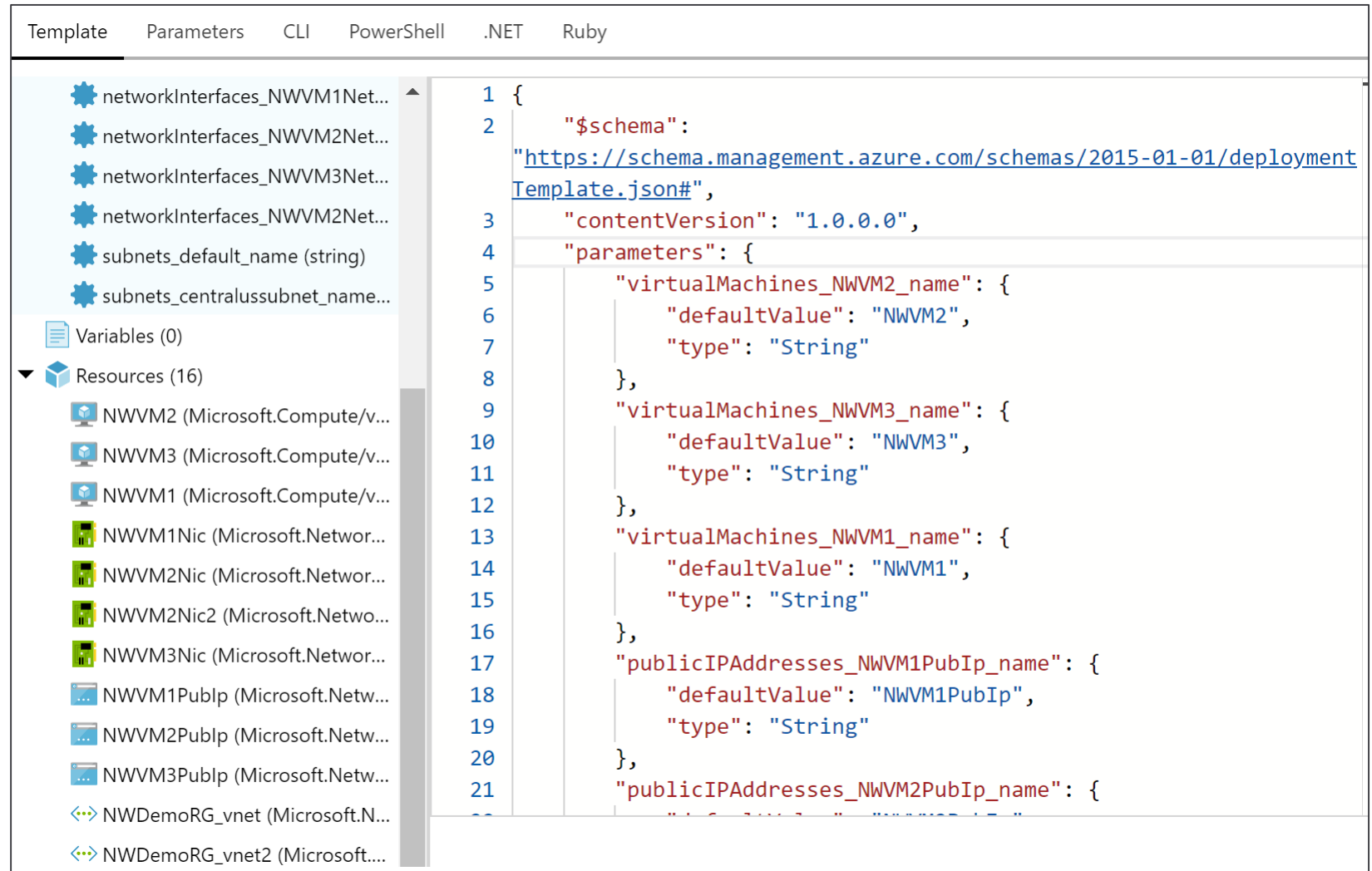
CHEF™



ANSIBLE

Azure Resource Manager Templates

ARM templates are based on JSON syntax



The screenshot displays the Azure Resource Manager Template editor interface. The top navigation bar includes tabs for Template, Parameters, CLI, PowerShell, .NET, and Ruby. The left sidebar shows a tree view of the template structure, including Variables (0) and Resources (16). The main editor area displays the JSON template content, which defines the schema, content version, and parameters for a virtual machine deployment.

```
1 {
2   "$schema":
3     "https://schema.management.azure.com/schemas/2015-01-01/deployment
4     Template.json#",
5   "contentVersion": "1.0.0.0",
6   "parameters": {
7     "virtualMachines_NWVM2_name": {
8       "defaultValue": "NWVM2",
9       "type": "String"
10    },
11    "virtualMachines_NWVM3_name": {
12      "defaultValue": "NWVM3",
13      "type": "String"
14    },
15    "virtualMachines_NWVM1_name": {
16      "defaultValue": "NWVM1",
17      "type": "String"
18    },
19    "publicIPAddresses_NWVM1PubIp_name": {
20      "defaultValue": "NWVM1PubIp",
21      "type": "String"
22    },
23    "publicIPAddresses_NWVM2PubIp_name": {
```

Azure Resource Providers

Resource providers registered for use with your subscription can be [found in the portal \(or via PowerShell, REST API, or CLI\)](#).

Create a resource

All services

FAVORITES

Container instances

Kubernetes services

Container services

Recent

Service Fabric clusters

Automation Accounts

App Services

Logic apps

SQL databases

Cost Management + Bill...

Subscriptions

Network Watcher

SQL servers

Security Center

Monitor

Microsoft Azure Sponsorship - Resource providers

Subscription

Search (Ctrl+/)

Diagnose and solve problems

Security

Events

Billing

Partner information

Settings

Programmatic deployment

Resource groups

Resources

Usage + quotas

Policies

Management certificates

My permissions

Resource providers

Refresh

Microsoft.KeyVault	Registered
Microsoft.Logic	Registered
Microsoft.MachineLearning	Registered
Microsoft.ManagedIdentity	Registered
Microsoft.Maps	Registered
Microsoft.Network	Registered
Microsoft.NotificationHubs	Registered
Microsoft.Operationallnsights	Registered
Microsoft.OperationsManagement	Registered
Microsoft.PolicyInsights	Registered
Microsoft.Portal	Registered
Microsoft.RecoveryServices	Registered
Microsoft.Relay	Registered
Microsoft.ResourceHealth	Registered

Why use ARM Templates?

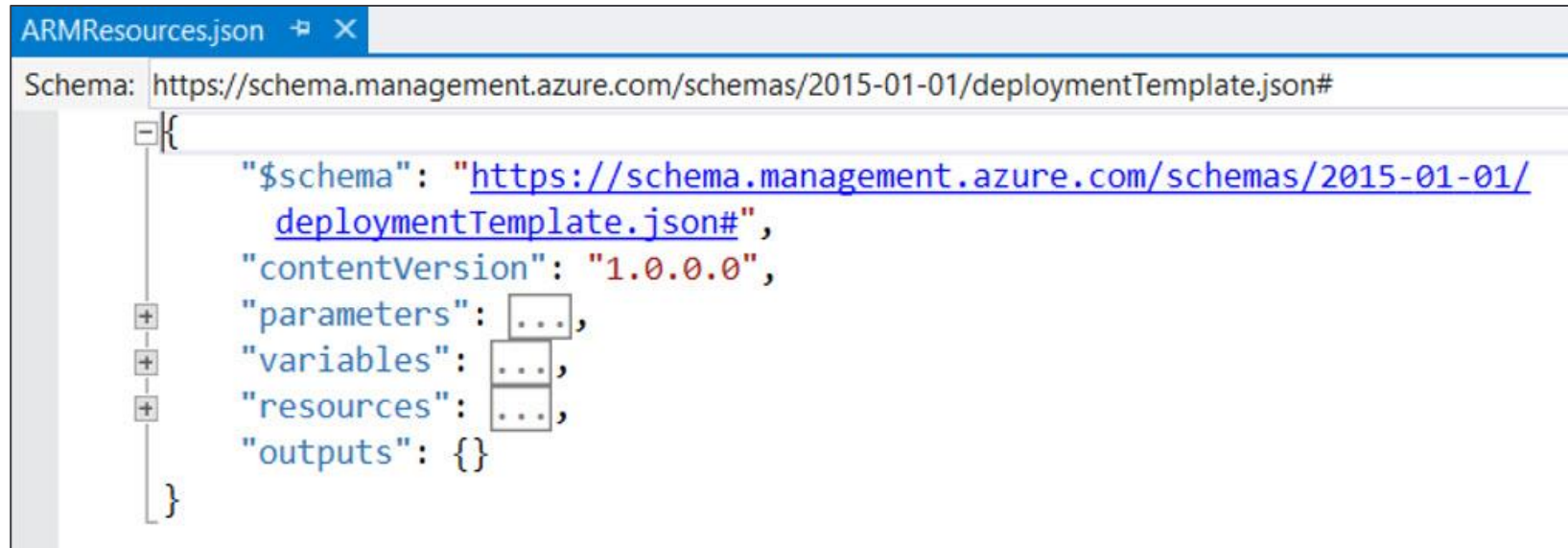
- Include the configuration of Azure resources in source control (“Infrastructure as Code”)
- Repeat the deployment process numerous times
- Automate deployments
- Employ continuous integration techniques
- Utilize DevOps principles and practices
- Repeatedly utilize testing infrastructure then de-provision it when finished

Ways to create an ARM Template (...*What works for Peter...*)

1. From the automation script available from the Azure Portal (which is imperfect – more on that in a moment)
2. An ARM template in Visual Studio / Visual Studio Code
3. [QuickStart Templates](#) from [GitHub](#) (there's a lot to choose from – the templates that start with 101 are less complex)
4. Create from the ground up
5. Start with a combination of 1 and 2 or 3, customizing the way you like it

Basic Template Structure: \$schema

This refers to the JSON schema. Note that the schema specified is different in the parameters file vs. the main ARM deployment file.

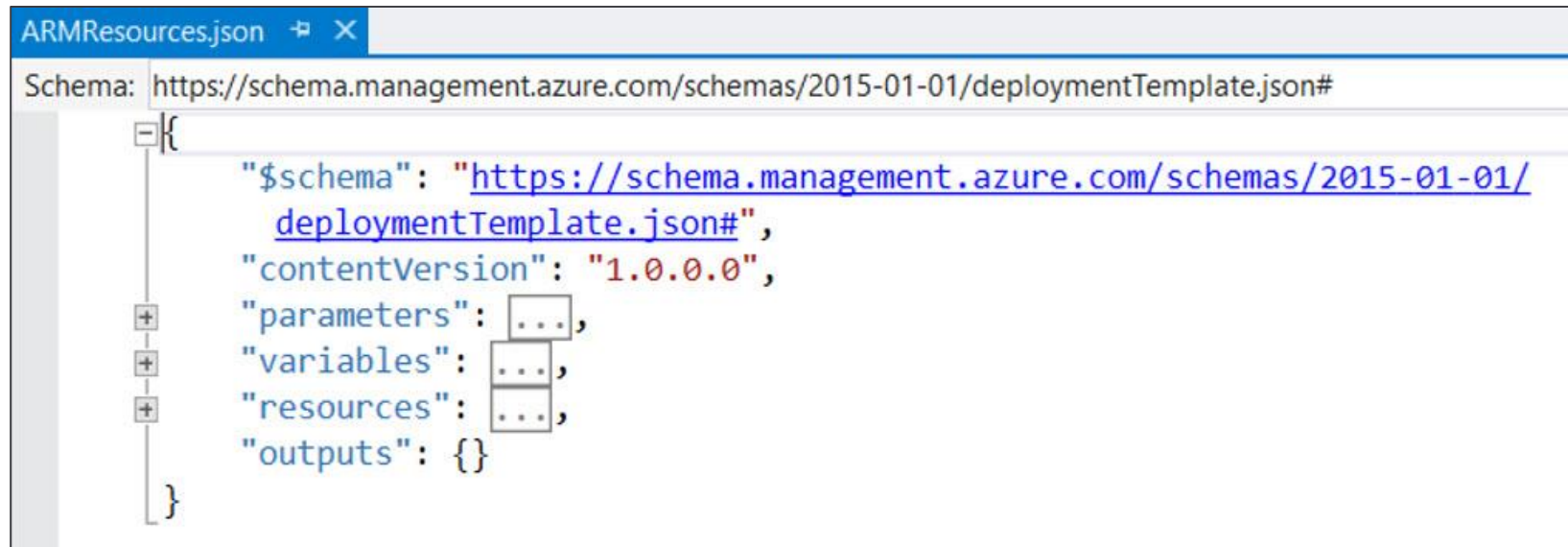


The screenshot shows a code editor window titled 'ARMResources.json'. The 'Schema' field at the top is set to 'https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#'. The main content is a JSON object representing an ARM template structure. The '\$schema' property is highlighted with a blue link. The 'contentVersion' is '1.0.0.0'. The 'parameters', 'variables', and 'resources' properties are each followed by a small box containing '...', indicating they are expandable sections. The 'outputs' property is an empty object {}.

```
{
  "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
  "contentVersion": "1.0.0.0",
  "parameters": ...,
  "variables": ...,
  "resources": ...,
  "outputs": {}
}
```

Basic Template Structure: Contentversion

You can increment this version if you'd like to manage the changes made over time. The default is "1.0.0.0."

A screenshot of a code editor window titled 'ARMResources.json'. The editor shows a JSON schema for an ARM deployment template. The schema URL is 'https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#'. The JSON structure is as follows:

```
{  "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",  "contentVersion": "1.0.0.0",  "parameters": {},  "variables": {},  "resources": {},  "outputs": {}}
```

 The 'contentVersion' field is highlighted in red. The editor has a tree view on the left with expand/collapse icons for each property.

Basic Template Structure: Parameters

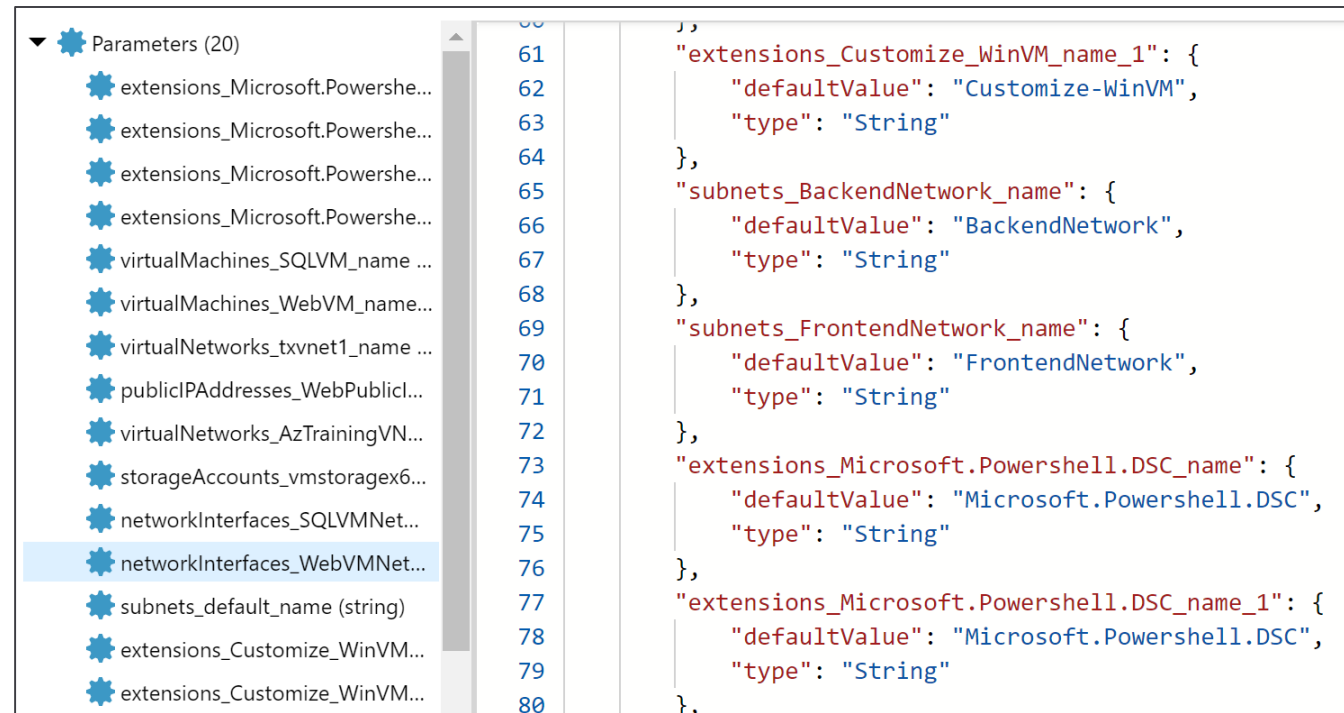
This is where you store the definitions of all parameters used throughout the ARM template

Type

Most common types are string or int;
Password = securestring;

MetaData / Description

Helpful info to remember what it's used for, or any other notes you want to leave for yourself and your team.



Basic Template Structure: Parameters

This is where you store the definitions of all parameters used throughout the ARM template

Allowed Values

Provides a list to choose from
(e.g. restrict list of VM Sizes)

Default Value

Specifies the default value, but leaving choices (e.g. deploy DS2_v2 as VM size, but one can choose another from the list)

```
90      "type": "Microsoft.Compute/virtualMachines",
91      "name": "[parameters('virtualMachines_SQLVM_name')]",
92      "apiVersion": "2017-12-01",
93      "location": "eastus",
94      "tags": {
95        "displayName": "SQLVM"
96      },
97      "scale": null,
98      "properties": {
99        "hardwareProfile": {
100          "vmSize": "Standard_DS1_v2"
101        },
102        "storageProfile": {
103          "imageReference": {
104            "publisher": "MicrosoftSQLServer",
105            "offer": "SQL2014SP2-WS2012R2",
106            "sku": "Standard",
107            "version": "latest"
108          },
109          "osDisk": {
110            "osType": "Windows",
111            "name": "[concat(parameters('virtualMachines_SQLVM_name'), 'osdisk')]"
```

Basic Template Structure: Variables

The variables section contains references to settings, mostly picked up by the resources section later on, making definitions easier

4 X variables (34)	90	
X customScriptFolder	91	"WebVMImagePublisher": "MicrosoftWindowsServer",
X customScriptUriScriptFi	92	"WebVMImageOffer": "WindowsServer",
X customScriptUri	93	"WebVMOSDiskName": "WebVMOSDisk",
X AzTrainingVNetPrefix	94	"WebVMVmSize": "Standard_DS1_v2",
X AzTrainingVNetSubnet1	95	"WebVMVnetID": "[resourceId('Microsoft.Network/virtualNetworks', 'AzTrainingVNet
X AzTrainingVNetSubnet1	96	"WebVMSubnetRef": "[concat(variables('WebVMVnetID'), '/subnets/', variables('AzT
X AzTrainingVNetSubnet2	97	"WebVMStorageAccountContainerName": "vhds",
X AzTrainingVNetSubnet2	98	"WebVMNicName": "[concat(parameters('WebVMName'), 'NetworkInterface')]",
X vmstorageName	99	"WebPublicIPName": "WebPublicIP",
X WebVMImagePublisher	100	"WebDSCArchiveFolder": "DSC",
X WebVMImageOffer	101	"WebDSCArchiveFileName": "WebDSC.zip",
X WebVMOSDiskName	102	"SQLVMImagePublisher": "MicrosoftSQLServer",
X WebVMVmSize	103	"SQLVMImageOffer": "SQL2014SP2-WS2012R2",
X WebVMVnetID	104	"SQLVMOSDiskName": "SQLVMOSDisk",
X WebVMSubnetRef	105	"SQLVMVmSize": "Standard_DS1_v2",
X WebVMStorageAccount	106	"SQLVMVnetID": "[resourceId('Microsoft.Network/virtualNetworks', 'AzTrainingVNet
X WebVMNicName	107	"SQLVMSubnetRef": "[concat(variables('SQLVMVnetID'), '/subnets/', variables('AzT
X WebPublicIPName	108	"SQLVMStorageAccountContainerName": "vhds",
	109	"SQLVMNicName": "[concat(parameters('SQLVMName'), 'NetworkInterface')]",
		"SQLDISK1": "[concat('http://', variables('vmstorageName'), '.blob.core.windows.net

Basic Template Structure: Resources

The resources section defines each resource to be deployed, with references to parameters and variables as necessary. The elements which are defined vary based on the kind of resource which is being deployed.

```
192 {
193   "name": "[parameters('WebVMName')]",
194   "type": "Microsoft.Compute/virtualMachines",
195   "location": "[resourceGroup().location]",
196   "apiVersion": "2015-06-15",
197   "dependsOn": [
198     "[resourceId('Microsoft.Storage/storageAccounts', variables('vmstorageName'))]",
199     "[resourceId('Microsoft.Network/networkInterfaces', variables('WebVMNicName'))]",
200   ],
201   "tags": {
202     "displayName": "WebVM"
203   },
204   "properties": {
205     "hardwareProfile": {
206       "vmSize": "[variables('WebVMVmSize')]"
207     },
208     "osProfile": {
209       "computerName": "[parameters('WebVMName')]",
210       "adminUsername": "[parameters('WebVMAdminUsername')]",
211       "adminPassword": "[parameters('WebVMAdminPassword')]"
212     },
213     "storageProfile": {
214       "imageReference": {
215         "publisher": "[variables('WebVMImagePublisher')]",
216         "offer": "[variables('WebVMImageOffer')]",
217         "sku": "[parameters('WebVMWindowsOSVersion')]",
218         "version": "latest"
219       }
```

Basic Template Structure: Resources (typical)

Location: You might be tempted to create a parameter for Location. However, a better practice is to inherit the location from the resource group

Type: The Type element for a resource is a combination of Resource Provider (discussed before) plus the Resource Type (ex: Microsoft.sql/servers)

Comments: Helpful info to clarify what the resource is, or what it's being used for.

DependsOn: This helps Azure understand dependencies, so it can deploy resources in parallel, or sequentially, as appropriate

API Version: A version is specified for each resource which is associated with a version of the REST API. The version impacts which elements can be specified for the resource, so the versions are updated on occasion

Azure Automation Script – Resource Group Export

Events

Settings

- Quickstart
- Resource costs
- Deployments
- Policies
- Properties
- Locks
- Automation script**

Monitoring

- Insights (preview)
- Alerts
- Metrics
- Diagnostic settings

Parameters (20)

Variables (0)

Resources (16)

- SQLVM (Microsoft.Compute/virtua...
- WebVM (Microsoft.Compute/virtu...
- SQLVMNic (Microsoft.Network/ne...
- [parameters('networkInterfaces_W...
- WebPublicIP (Microsoft.Network/...
- AzTrainingVNet (Microsoft.Networ...
- [parameters('virtualNetworks_txvn...
- vmstorage (Microsoft.Storage/stor...
- [concat(parameters('virtualMachin...
- Customize-WinVM (Microsoft.Co...
- SQLDSC (Microsoft.Compute/virtu...
- Customize-WinVM (Microsoft.Co...
- WebDSC (Microsoft.Compute/virt...
- [concat(parameters('virtualNetwor...
- [concat(parameters('virtualNetwor...
- [concat(parameters('virtualNetwor...

```
1 {
2   "$schema":
3   "https://schema.management.azure.com/schemas/2015-01-01/deployment
4   Template.json#",
5   "contentVersion": "1.0.0.0",
6   "parameters": {
7     "extensions_Microsoft.Powershell.DSC_modulesUrl": {
8       "defaultValue": null,
9       "type": "SecureString"
10    },
11    "extensions_Microsoft.Powershell.DSC_configurationFunction": {
12      "defaultValue": null,
13      "type": "SecureString"
14    },
15    "extensions_Microsoft.Powershell.DSC_modulesUrl_1": {
16      "defaultValue": null,
17      "type": "SecureString"
18    },
19    "extensions_Microsoft.Powershell.DSC_configurationFunction_1": {
20      "defaultValue": null,
21      "type": "SecureString"
22    }
23  },
24  "resources": [
25    {
26      "name": "[concat(parameters('virtualMachinesNamePrefix'), 'DSCAgent')]",
27      "type": "Microsoft.Compute/virtualMachines/extensions",
28      "apiVersion": "2015-06-01",
29      "location": "[parameters('location')]",
30      "tags": {
31        "displayName": "[concat(parameters('virtualMachinesNamePrefix'), 'DSCAgent')]"
32      },
33      "properties": {
34        "publisher": "Microsoft.Extensions",
35        "type": "DSCAgent",
36        "typeHandlerVersion": "2.0",
37        "autoUpgradeMinorVersion": true,
38        "settings": {
39          "configurationFunction": "[concat(parameters('DSCConfigurationFunction'), 'DSCAgent.ps1')]",
40          "modulesUrl": "[parameters('DSCModulesUrl')]"
41        }
42      }
43    },
44    {
45      "name": "[concat(parameters('virtualMachinesNamePrefix'), 'DSCAgent_1')]",
46      "type": "Microsoft.Compute/virtualMachines/extensions",
47      "apiVersion": "2015-06-01",
48      "location": "[parameters('location')]",
49      "tags": {
50        "displayName": "[concat(parameters('virtualMachinesNamePrefix'), 'DSCAgent_1')]"
51      },
52      "properties": {
53        "publisher": "Microsoft.Extensions",
54        "type": "DSCAgent",
55        "typeHandlerVersion": "2.0",
56        "autoUpgradeMinorVersion": true,
57        "settings": {
58          "configurationFunction": "[concat(parameters('DSCConfigurationFunction'), 'DSCAgent.ps1')]",
59          "modulesUrl": "[parameters('DSCModulesUrl_1')]"
60        }
61      }
62    }
63  ]
64 }
```

Demo

Azure Automation Script

Questions Landing Spot

“...If you want good answers,
ask better questions...”

© Randy Glasbergen

Azure Automation Script – Resource Group Export - Challenges

- The value is part of the parameter name. Ex: if you have a web app named “007FFFWebAppDemo” it will generate a parameter called “sites_007FFFWebAppDemo_name.”
- Not all resources can be scripted out in this manner yet. (You’ll see a message at the top of the template pane when this occurs)
- Default values are overused
- The admin password (when required) is not parameterized
- There are a few inaccurate values that come out, which result in deployment failures when reusing the template as-is

ARM Templates flow to get kickstarted

- Use **Visual Studio 2017/2019** as your « authoring » environment
- Split out **parameter** file in a separate parameters.azuredeploy.json
- **Fine-tune** parameters according your needs
- Matching **naming conventions**
- Add **metadata description** for each parameter
- Add **parameters** for anything that could vary between environments (dev, test, customer A, customer B,...)
- **NEVER** store passwords as cleartext (rather securedstring with prompt or integrate with Azure Key Vault)
- Add **variables** where they make sense
- **Fine-tune** the Resources section
- Use **Resource Group** inherited location
- Use **Tags** (in relation to Azure Policies)

ARM Templates – GitHub Quickstart Templates

<http://www.github.com/Azure/Quickstart-Templates>

Azure / azure-quickstart-templates

Watch 513

Star 3,835

Fork 5,482

<> Code

Issues 425

Pull requests 34

Projects 0

Wiki

Insights

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

azure

templates

arm

18,848 commits

3 branches

0 releases

664 contributors

MIT

Branch: master


New pull request

Create new file









Upload files

Find file

Clone or download

 bmoore-msft Merge pull request #4904 from MadhusudhanRavi/master ...

Latest commit be4bfc2 Jul 6, 2018

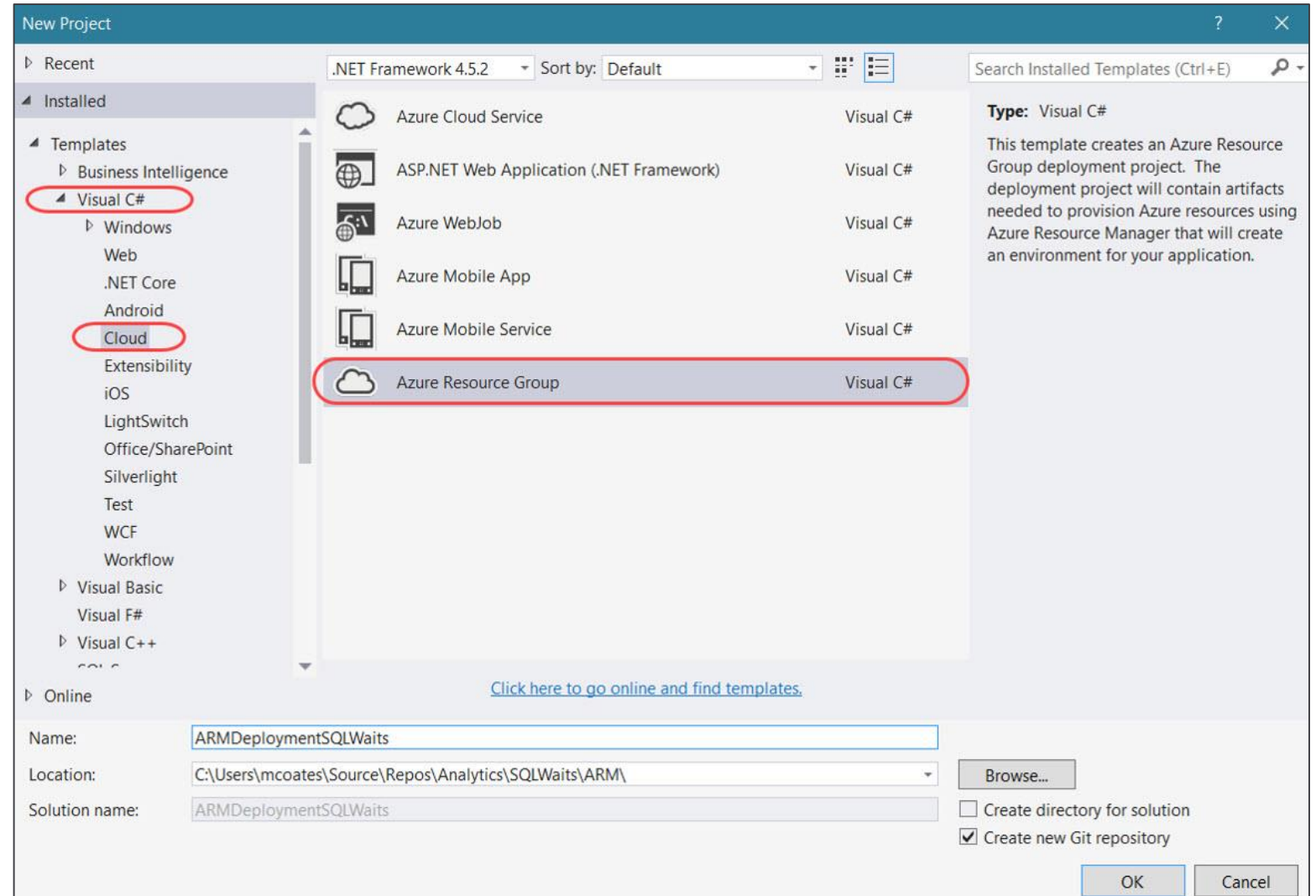
 .github	Update stale.yml	May 22, 2018
 1-CONTRIBUTION-GUIDE	text tweak	Jun 29, 2018
 100-blank-template	added schema propety to metadata.json	Jun 14, 2018
 100-marketplace-sample	added schema propety to metadata.json	Jun 14, 2018
 101-1vm-2nics-2subnets-1vnet	added schema propety to metadata.json	Jun 14, 2018
 101-Telegraf-InfluxDB-Grafana	added schema propety to metadata.json	Jun 14, 2018
 101-aci-dynamicsnav	added schema propety to metadata.json	Jun 14, 2018
 101-aci-linuxcontainer-public-ip	added schema propety to metadata.json	Jun 14, 2018

Demo

Azure QuickStart Templates

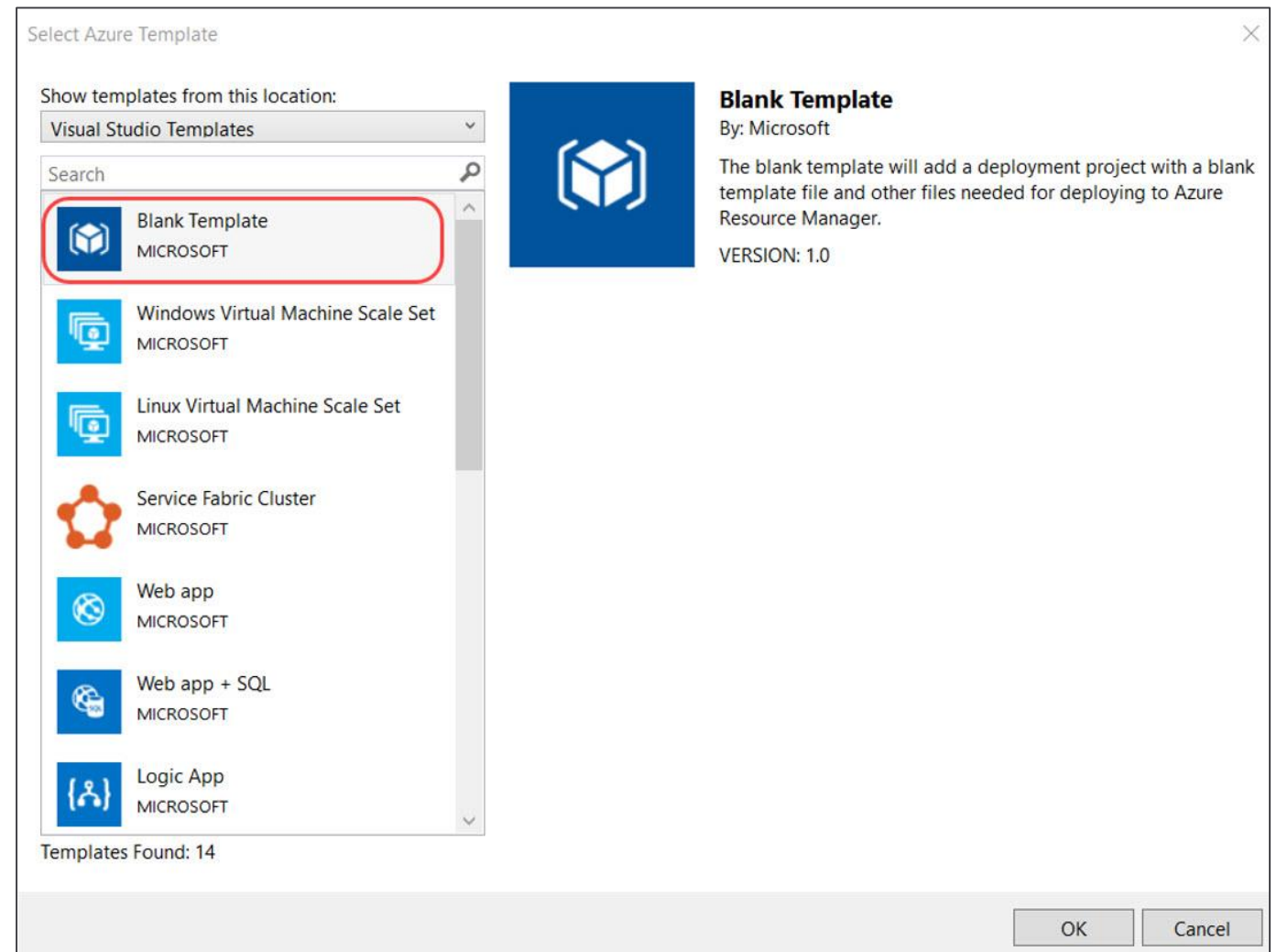
Create an ARM Project in VS2017 / VS2019

If you don't see this option, it means the Azure SDK has not been installed yet



Create an ARM Project in VS2017

**Start from a Blank Template or
use any of the GitHub QuickStart
Templates**

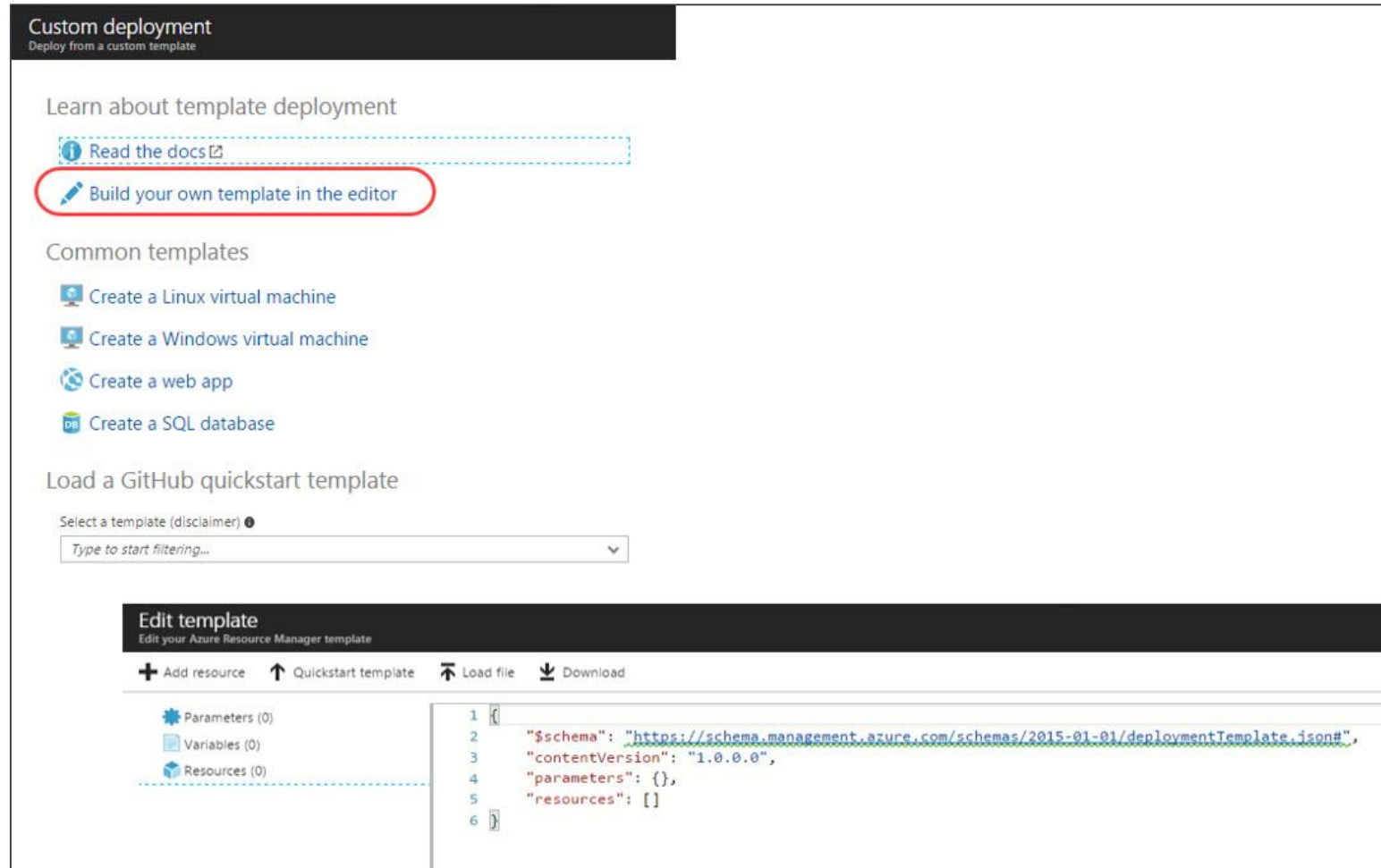


Demo

Authoring ARM Templates using Visual Studio 2019

Running a template deployment from the Azure Portal

Add Resource / Template Deployment



The screenshot displays the 'Custom deployment' interface in the Azure Portal. The top section, titled 'Custom deployment', includes a link to 'Learn about template deployment' and two primary actions: 'Read the docs' and 'Build your own template in the editor'. The latter is highlighted with a red circle. Below this, a 'Common templates' section offers quickstarts for Linux/Windows VMs, web apps, and SQL databases. The 'Load a GitHub quickstart template' section features a search bar. The bottom half of the image shows the 'Edit template' editor, which includes a sidebar for managing parameters, variables, and resources, and a main text area containing a JSON template snippet.

Custom deployment
Deploy from a custom template

Learn about template deployment

[Read the docs](#)

[Build your own template in the editor](#)

Common templates

- Create a Linux virtual machine
- Create a Windows virtual machine
- Create a web app
- Create a SQL database

Load a GitHub quickstart template

Select a template (disclaimer)

Type to start filtering...

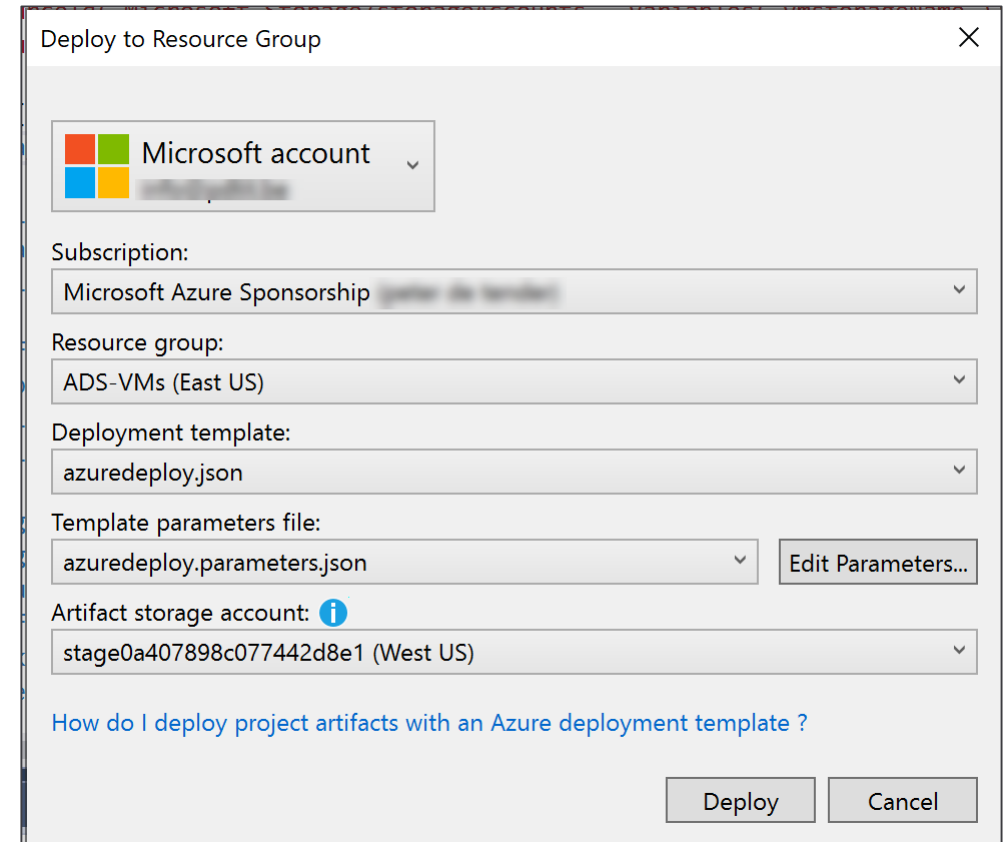
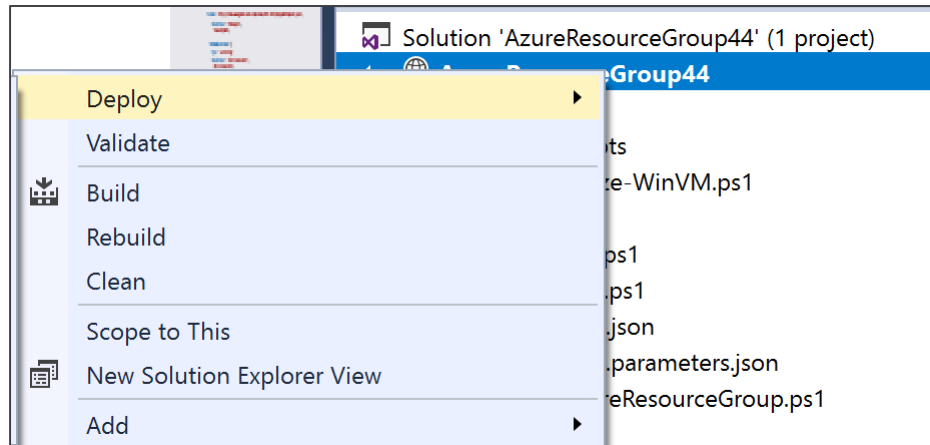
Edit template
Edit your Azure Resource Manager template

+ Add resource ↑ Quickstart template ↶ Load file ↴ Download

Parameters (0)
Variables (0)
Resources (0)

```
1 {  
2   "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",  
3   "contentVersion": "1.0.0.0",  
4   "parameters": {},  
5   "resources": []  
6 }
```


Running a template deployment from VS2017/VS2019



Monitoring ARM template deployment status

Azure Portal

The screenshot shows the Azure Portal interface for the resource group 'SQLSkillsWaitsLibraryRGDev'. The 'Deployments' tab is selected, showing a table with one deployment: 'resources-0606-1840' with a status of 'Succeeded'. The deployment details for 'resources-0606-1840' are shown below the table, including the deployment date (6/6/2017, 2:41:33 PM), status (Succeeded), duration (35 seconds), and resource group (SQLSkillsWaitsLibraryRGDev). The 'Outputs' section shows 'NO DEPLOYMENT OUTPUTS'. The 'Inputs' section shows the environment name as 'Dev', the SQL server administrator user as 'BTeamAdminAcct', the SQL server administrator password as 'BTeamAdminAcct', and the SQL server name as 'sqlserversqlwaitsdev'.

DEPLOYMENT NAME	STATUS	TIMESTAMP	DURATION
resources-0606-1840	Succeeded	6/6/2017, 2:41:33 PM	35 seconds

resources-0606-1840
Deployment

Summary

DEPLOYMENT DATE: 6/6/2017, 2:41:33 PM

STATUS: Succeeded

DURATION: 35 seconds

RESOURCE GROUP: SQLSkillsWaitsLibraryRGDev

RELATED: Events

Outputs

NO DEPLOYMENT OUTPUTS

Inputs

ENVIRONMENTNAME: Dev

SQLSERVERADMINUSER: BTeamAdminAcct

SQLSERVERADMINPW: BTeamAdminAcct

SQLSERVERNAME: sqlserversqlwaitsdev

Visual Studio Output

The screenshot shows the Visual Studio Output window for the resource group 'SQLSkillsWaitsLibraryRGDev'. The output shows the deployment status of various resources, including 'Microsoft.Sql/servers 'sqlserversqlwaitsdev'', 'Microsoft.Sql/servers/firewallRules 'sqlserversqlwaitsdev/AllowAllWindowsAzureIps'', 'Microsoft.Storage/storageAccounts 'diagstrgsqlwaitsdev'', 'Microsoft.Web/serverfarms 'AppServicePlanSQLWaitsDev'', 'Microsoft.Web/sites 'AppSQLWaitsDev'', and 'Microsoft.Sql/servers/databases 'sqlserversqlwaitsdev/sqldatabasesqlwaits''. The deployment status for all resources is 'Succeeded'. The output also shows the deployment name 'resources-0606-1840', the correlation ID '70fef929-867a-4fed-93ea-2a7d613ae1d8', the resource group name 'SQLSkillsWaitsLibraryRGDev', the provisioning state 'Succeeded', the timestamp '6/6/2017 6:41:33 PM', and the mode 'Incremental'.

Output

Show output from: SQLSkillsWaitsLibraryRGDev

14:41:04 - [VERBOSE] 2:41:04 PM - Resource Microsoft.Sql/servers 'sqlserversqlwaitsdev' provisioning status is succeeded

14:41:05 - [VERBOSE] 2:41:05 PM - Checking deployment status in 5 seconds

14:41:10 - [VERBOSE] 2:41:10 PM - Resource Microsoft.Sql/servers/firewallRules 'sqlserversqlwaitsdev/AllowAllWindowsAzureIps' provisioning status is succeeded

14:41:10 - [VERBOSE] 2:41:10 PM - Resource Microsoft.Storage/storageAccounts 'diagstrgsqlwaitsdev' provisioning status is succeeded

14:41:10 - [VERBOSE] 2:41:10 PM - Checking deployment status in 5 seconds

14:41:15 - [VERBOSE] 2:41:15 PM - Checking deployment status in 5 seconds

14:41:20 - [VERBOSE] 2:41:20 PM - Resource Microsoft.Web/serverfarms 'AppServicePlanSQLWaitsDev' provisioning status is succeeded

14:41:21 - [VERBOSE] 2:41:21 PM - Checking deployment status in 5 seconds

14:41:26 - [VERBOSE] 2:41:26 PM - Resource Microsoft.Web/sites 'AppSQLWaitsDev' provisioning status is succeeded

14:41:26 - [VERBOSE] 2:41:26 PM - Checking deployment status in 5 seconds

14:41:31 - [VERBOSE] 2:41:31 PM - Checking deployment status in 5 seconds

14:41:37 - [VERBOSE] 2:41:37 PM - Resource Microsoft.Sql/servers/databases 'sqlserversqlwaitsdev/sqldatabasesqlwaits' provisioning status is succeeded

14:41:37 -

14:41:37 - DeploymentName : resources-0606-1840

14:41:37 - CorrelationId : 70fef929-867a-4fed-93ea-2a7d613ae1d8

14:41:37 - ResourceGroupName : SQLSkillsWaitsLibraryRGDev

14:41:37 - ProvisioningState : Succeeded

14:41:37 - Timestamp : 6/6/2017 6:41:33 PM

14:41:37 - Mode : Incremental

14:41:37 -

14:41:37 -

14:41:37 -

14:41:37 - Successfully deployed template 'deploymenttemplates/resources.json' to resource group 'SQLSkillsWaitsLibraryRGDev'.

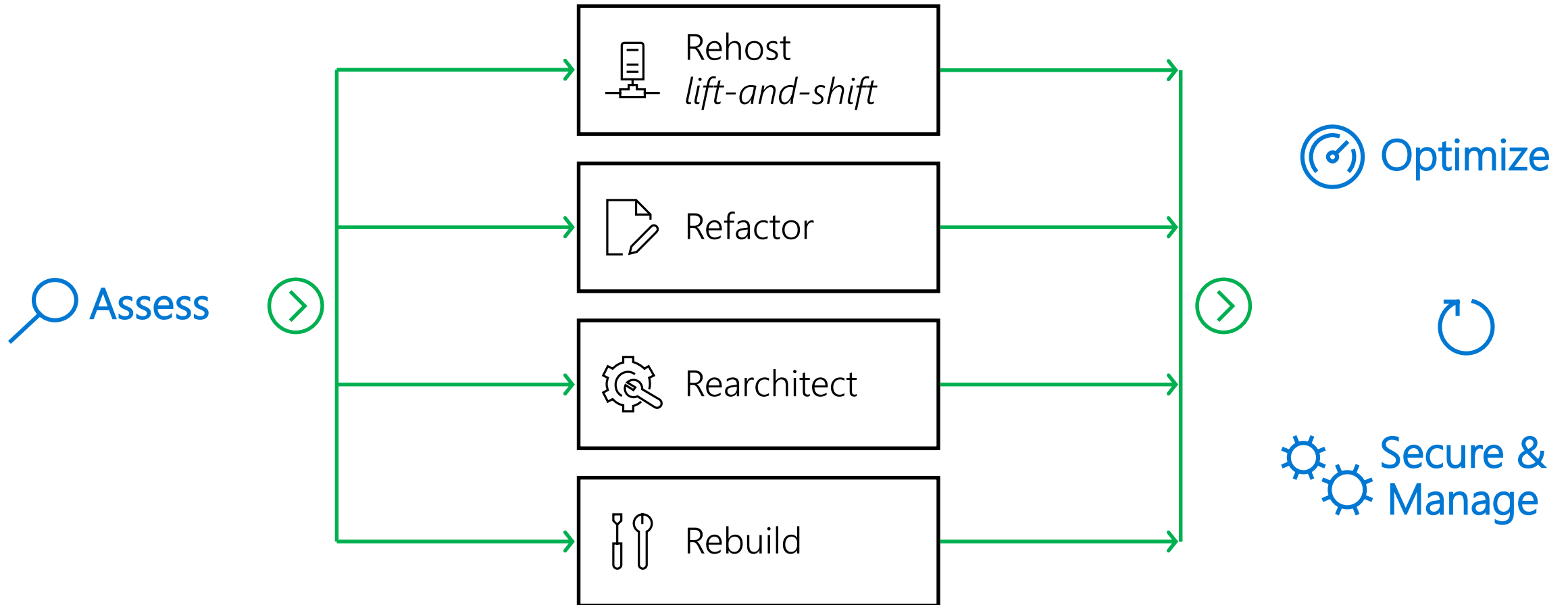
Demo

Monitoring ARM Template Deployment

VM Lift & Shift Migration using Azure Migrate

Azure Migration Experience: Unified and Extensible

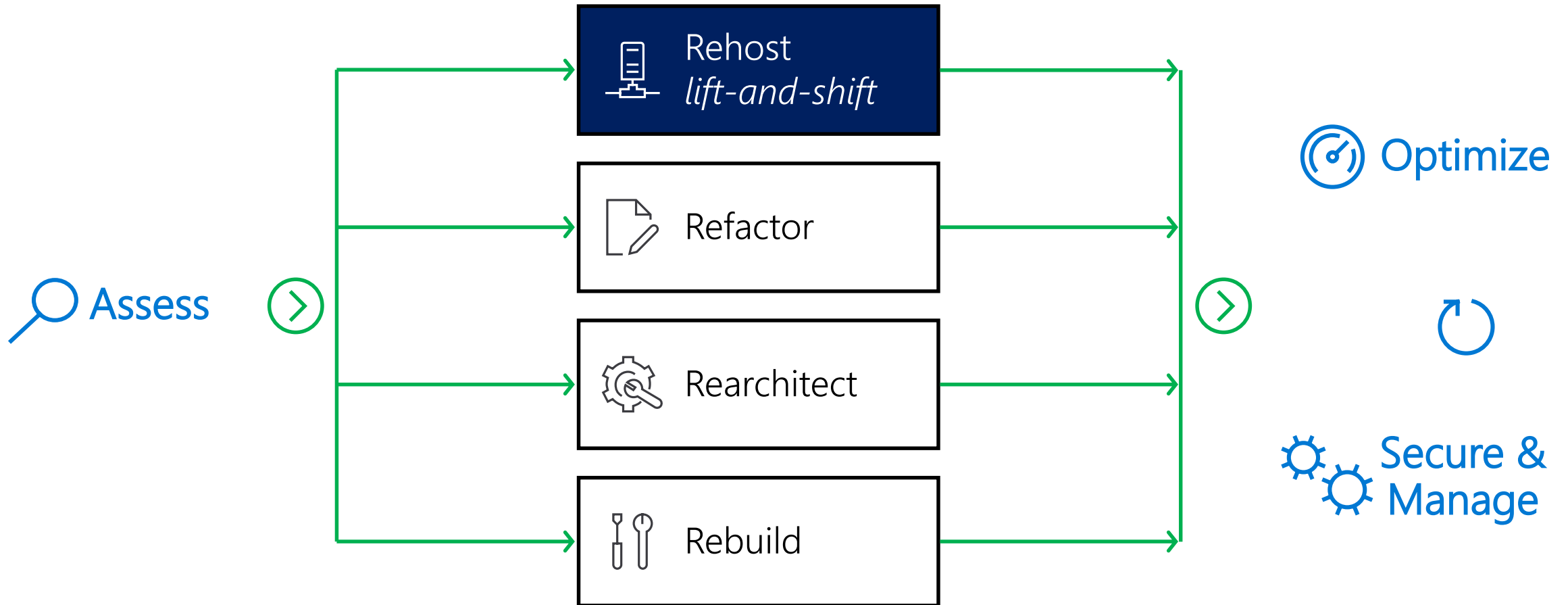
➤ Migrate*



*These migration strategies are adopted from Gartner research. Gartner also calls out a 5th strategy called "Replace" with SaaS

Azure Migration Experience: Unified and Extensible

➤ Migrate*



*These migration strategies are adopted from Gartner research. Gartner also calls out a 5th strategy called "Replace" with SaaS

Workloads Migrations Matrix

Migration Type	Rehost (lift and shift)	Refactor	Revise	Rebuild	Replace
Complexity	Moderate to Complex	Moderate to Complex	Moderate to Complex	Complex	Moderate
Example Workloads	RDS, Web Applications, Database	RDS, Web Applications	Legacy Applications, Databases	Legacy Applications	Legacy Applications
Tooling Capabilities	Extensive (First and 3 rd party)	Several (First and 3 rd party)	Minimal (First and 3 rd party)	Extensive (First and 3 rd party)	NA
Benefits	Proven, Quick and Container Support	Semi-Optimized and Container Support	Optimized, Container Support and PaaS enabled	Fully Optimized, Containerized and PaaS enabled	State of the Art, SaaS, HA, BCDR enabled.
DevOps / Automation	Moderate (Automation)	Moderate (Automation, Continuous Delivery)	Moderate (Automation, CI/CD)	Maximum (Automation, CI/CD)	NA

Migration strategies: Rehost application (i.e., lift & shift)



Rehost
(lift and shift)



Refactor



Rearchitect



Rebuild

What is it?

Redeploy an existing application to a cloud platform without modifying its code. The application is migrated “as is”, which provides baseline cloud benefits without the risk or costs of making code changes.

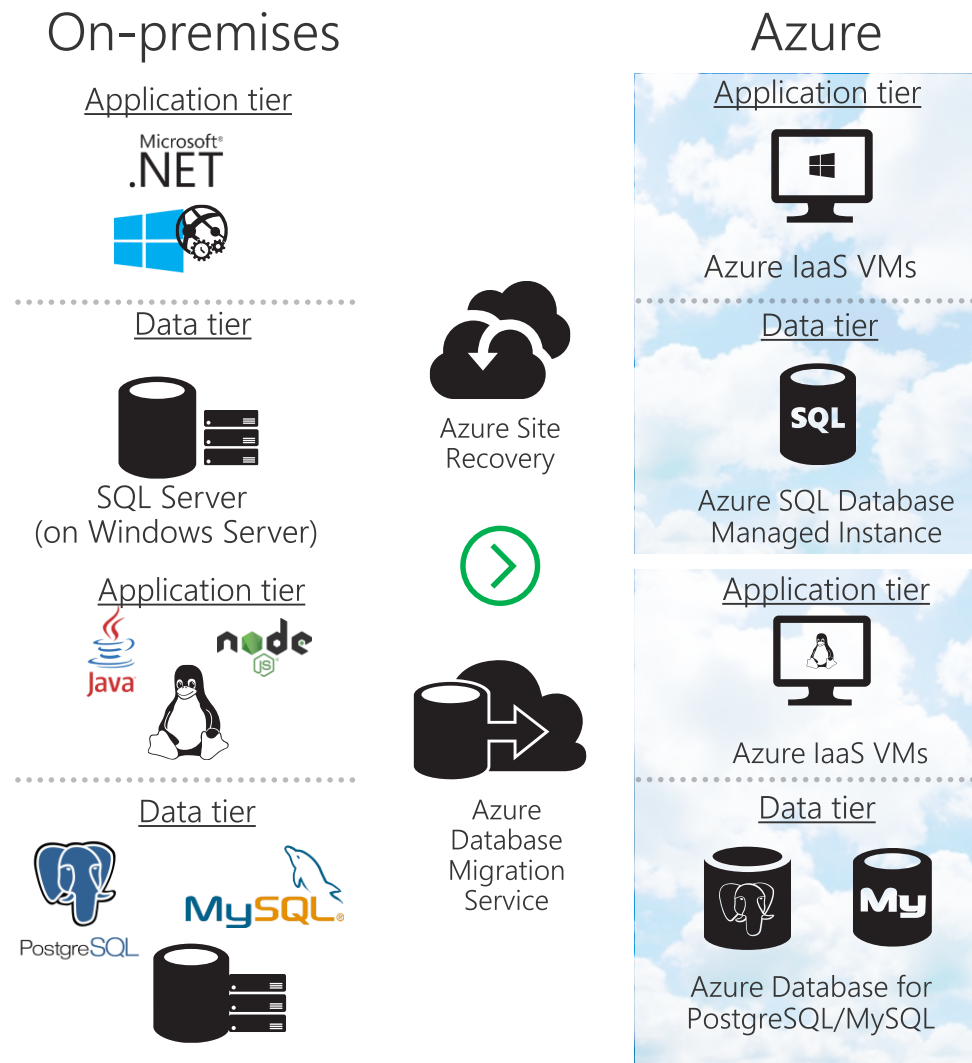
When to use

- Need to quickly move applications from on-premises to the cloud (e.g., datacenter contract expiry)
- When the application is needed, but evolving its capabilities isn't a business priority
- For applications which are architected to leverage Azure IaaS scalability
- Specific application or database requirements which can only be met by Azure VMs

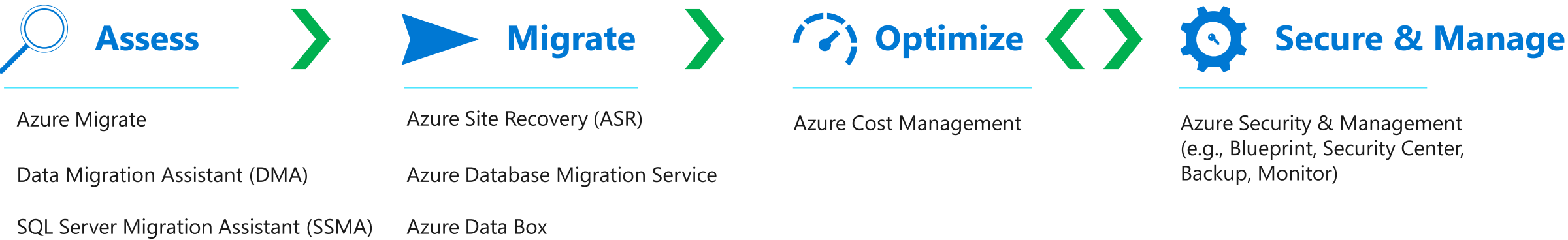
Example

Move a line of business application to Azure VMs

Sample illustration



Choice of tools for every stage and every requirement



Partners



Choosing between 1st and 3rd party assessment tools



Use Azure Migrate when customer needs...

- ✓ A Microsoft Azure branded tool which is free
- ✓ Assessment of VMware environments¹
- ✓ Assessments of brand new Azure platform functionality

Use ISV tools when customer needs...

 Cloudamize

 TSO Logic

 turbonomic

Assessment of heterogeneous environments (physical, VMware and Hyper-V)



Agentless dependency visualization



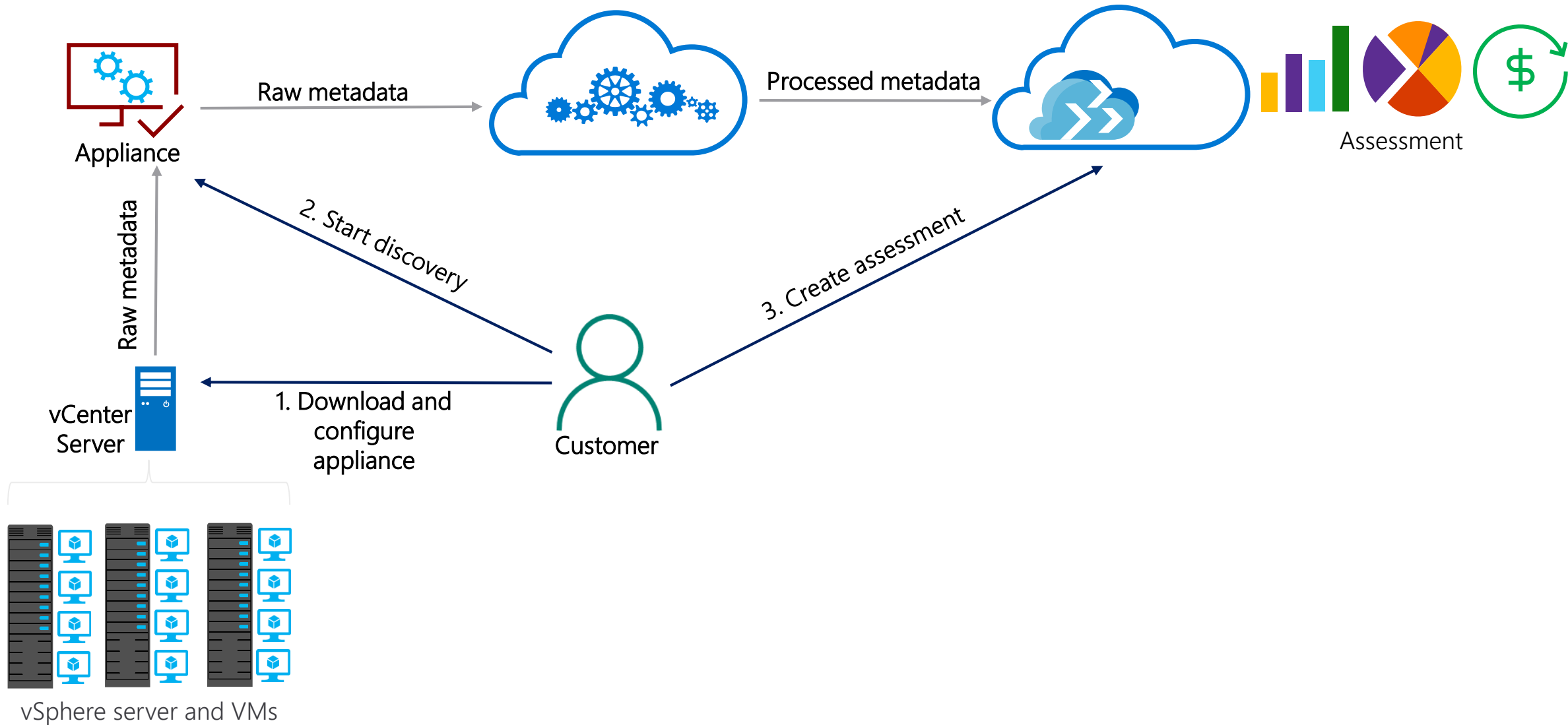
Cloud economics cost modelling (including storage/networking costs)



¹ Support for Hyper-V and physical servers is coming soon

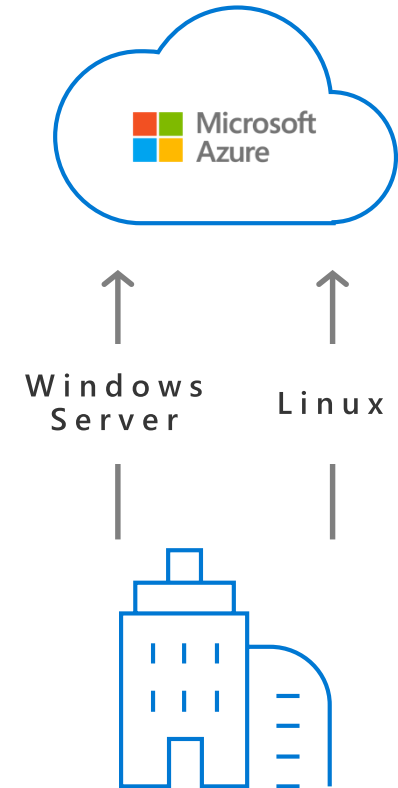
Tools listed are recommendations, but not an exhaustive list. See azure.com/migration/partners for full list.

Azure Migrate – How it works




Azure Migrate v2

- **Unified migration platform:** Use a single portal to start, run, and track your migration journey to Azure.
- **Range of tools:** Azure Migrate provides native tools, and integrates with other Azure services, as well as with ISV tools. Select the right assessment and migration tools, based on your organizational requirements.
- **Azure Migrate Server Assessment:** Use the Server Assessment tool to assess on-premises VMware VMs and Hyper-V VMs, for migration to Azure.
- **Azure Migrate Server Migration:** Use the Server Migration tool to migrate on-premises VMware VMs, Hyper-V VMs, cloud VMs, and physical servers to Azure.
- **Azure Migrate Database Assessment:** Assess on-premises databases for migration to Azure.
- **Azure Migrate Database Migration:** Migrate on-premises databases to Azure.



Replicate VMs

Azure **Migrate project** > **Servers**,
Azure Migrate: Server Migration, click
Replicate

 Azure Migrate - Servers
Microsoft

Overview

Servers

Databases

Data Box

Manage

Discovered items


Support + troubleshooting

New support request

Refresh

Last refreshed at: 7/8/2019, 11:08:58 AM (Click on "Refresh" to update the page)





Assessment tools


 **Azure Migrate: Server Assessment**

Discover

Assess


Overview

 Discovered servers	442
 Groups	2
 Assessments	2
 Notifications	0

 **Next step:** Start migrating your servers or optionally you can refine your application grouping with dependency analysis

Add more assessment tools? [Click here.](#)

Migration tools


 **Azure Migrate: Server Migration**

Discover

Replicate

Migrate

Overview

 Discovered servers	442
--	-----

Demo

VM Lift & Shift migration using Azure Migrate

Select the source

Replicate, > Source settings > Are your machines virtualized?, select Yes, with VMware vSphere.

Replicate

Source settings

Virtual machines

Target settings

Compute

Disks

Review + Start replication

The first step in migrating servers is to replicate them. Once replication completes, you can move the servers to Azure.

* Are your machines virtualized?

Yes, with VMware vSphere

* On-premises appliance

<appliance-name>

Replicate

Source settings

Virtual machines

Target settings

Compute

Disks

Review + Start replication

Select the virtual machines to be migrated.

* Import migration settings from an assessment?

No, I'll specify the migration settings manually

* Virtual machines

Search to filter machines

< Previous

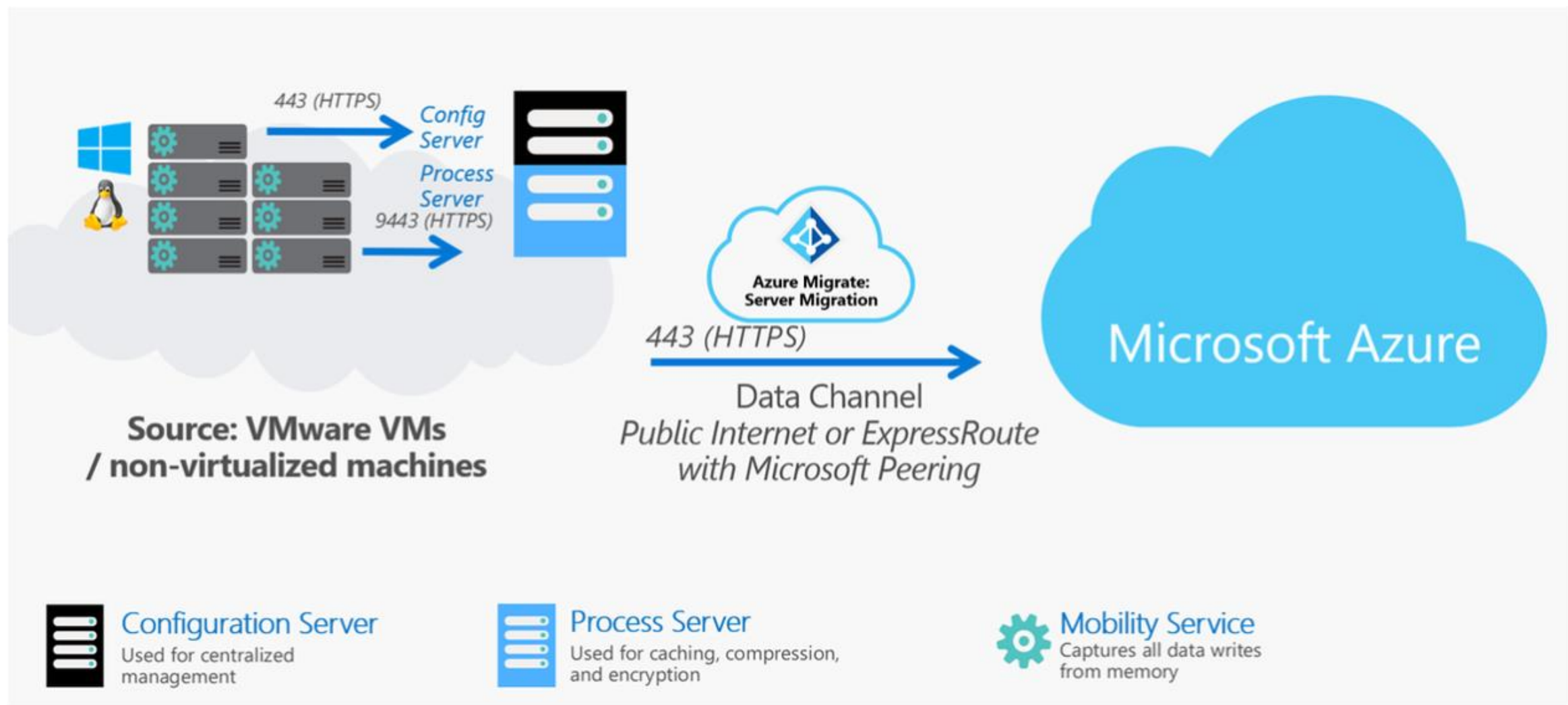
Page 1

Next >

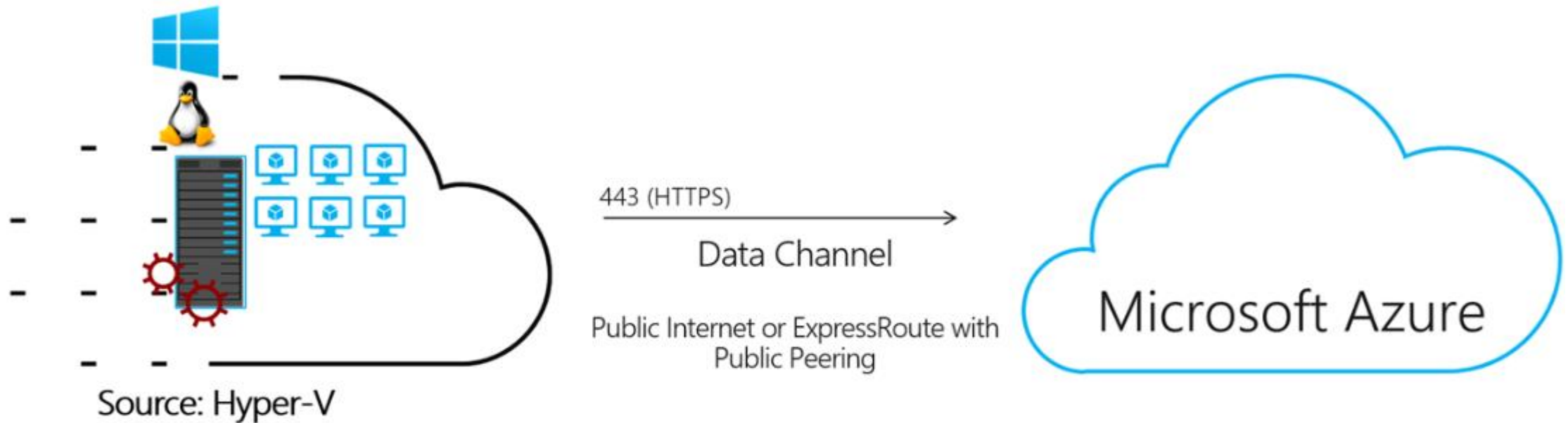
	NAME	IP ADDRESS	OPERATING SYSTEM	BOOT TYPE
<input checked="" type="checkbox"/>	ContosoVMwareMigr...	2404:f801:4800:25:c95f:5fd3:7347:4f91,1...	Microsoft Windows Server Threshold (64...	bios
<input checked="" type="checkbox"/>	ContosoCSASR	2404:f801:4800:25:29f9:2ebd:1ee0:eeb4,...	Microsoft Windows Server 2012 (64-bit)	bios
<input checked="" type="checkbox"/>	Contoso-FrontTier3		Microsoft Windows Server Threshold (64...	bios
<input checked="" type="checkbox"/>	ContosoWeb1	2404:f801:4800:25:9091:9912:5f46:9108,...	Microsoft Windows Server 2008 (32-bit)	bios
<input checked="" type="checkbox"/>	Contoso-Configuratio...		Microsoft Windows Server 2012 (64-bit)	bios
<input checked="" type="checkbox"/>	Contoso-AzureMigrat...		Microsoft Windows Server 2012 (64-bit)	bios
<input checked="" type="checkbox"/>	ContosoWeb3	10.150.13.218,2404:f801:4800:25:250:56f...	CentOS 4/5/6/7 (64-bit)	bios
<input checked="" type="checkbox"/>	ContosoAppSrv2	2404:f801:4800:25:5de5:e919:3448:be33,...	Microsoft Windows Server 2012 (64-bit)	bios
<input checked="" type="checkbox"/>	ContosoWeb2	10.150.13.201,2404:f801:4800:25:250:56f...	CentOS 4/5/6/7 (64-bit)	bios

Selected items : 9

Agent based Server Migration Architecture (VMware, AWS, Physical)



Agent based Server Migration Architecture (Hyper-V)



Azure Site Recovery for Azure IaaS VMs Lift & Shift Migration



Ensure
compliance

1

No-impact DR testing

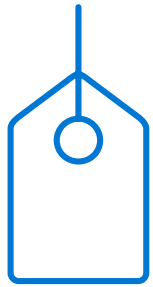
2

Meet your RPO and RTO SLAs

3

Centralized monitoring and alerting

Azure Site Recovery for Azure IaaS VMs Lift & Shift Migration



Simple

1

One-click replication

2

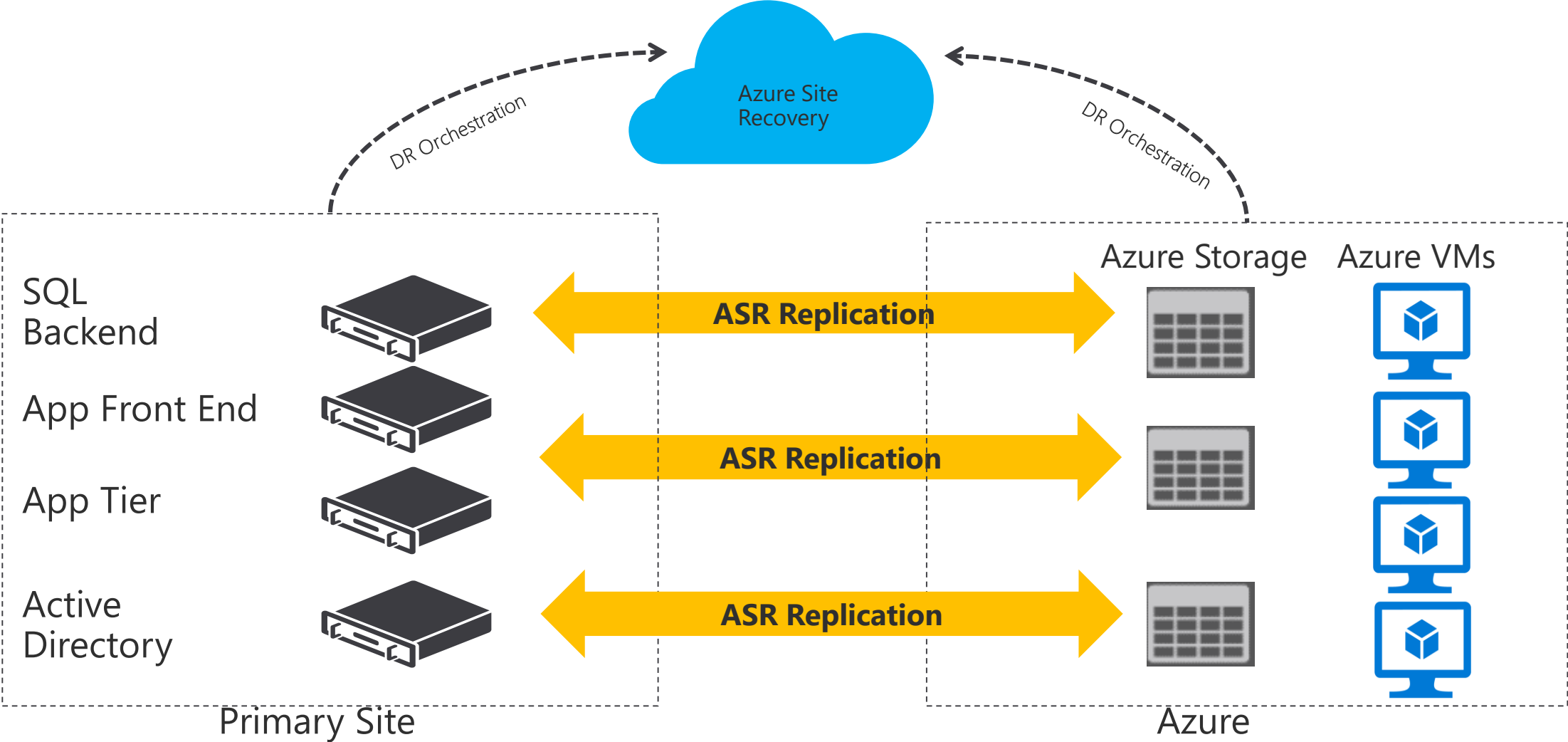
One-click VM Migration

Azure Site Recovery for Azure IaaS VMs Lift & Shift Migration

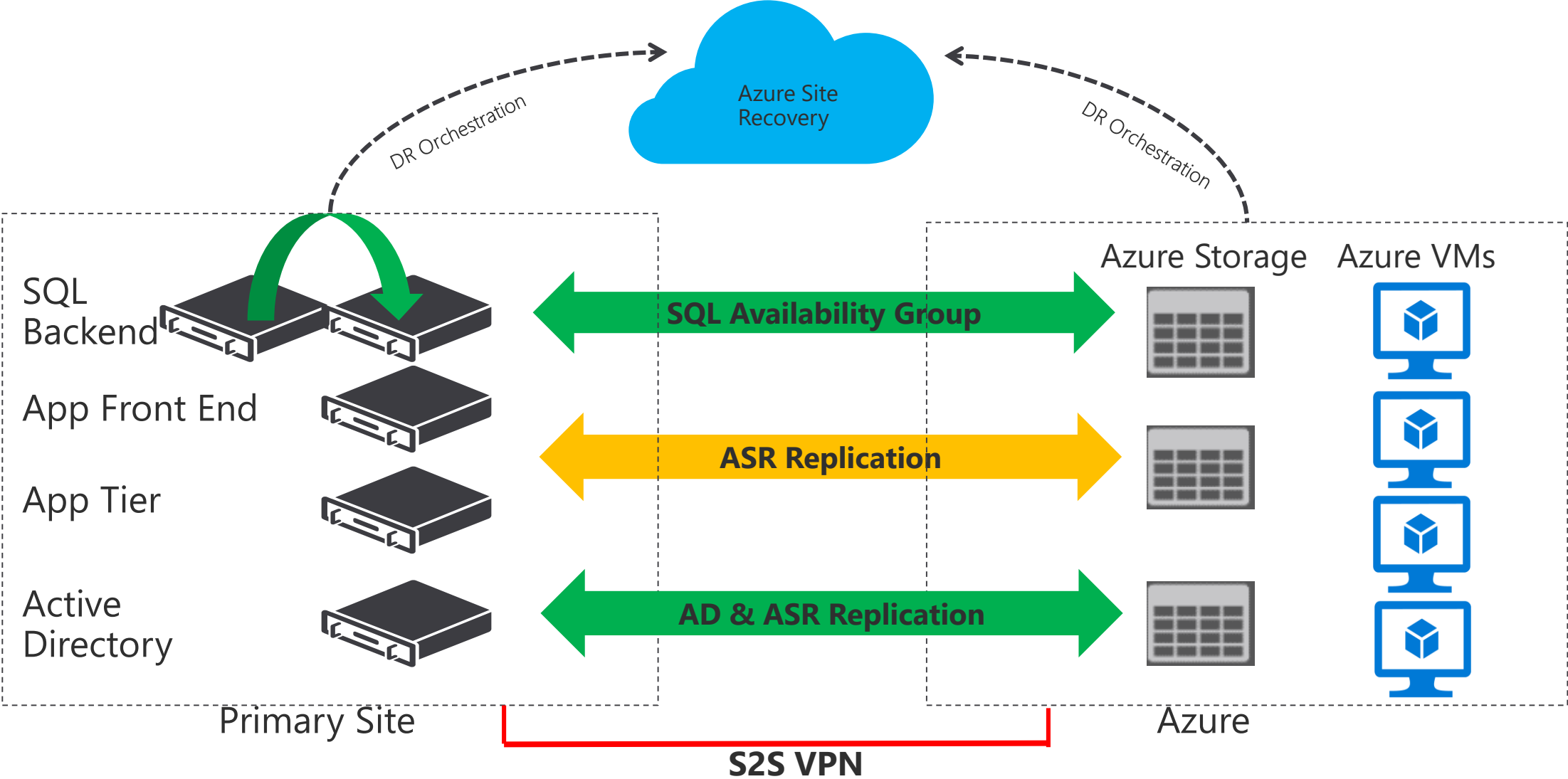


- **Zero** application **data loss** during migration
- **Near-zero** application **downtime** during migration
- **Broad coverage** for hypervisors, applications, operating systems, and Azure features
- **No-impact** application **testing** in Azure

Azure Site Recovery for Azure IaaS VMs Lift & Shift Migration



Azure Site Recovery for Azure IaaS VMs Lift & Shift Migration



Demo

Use Azure Site Recovery for Lift & Shift VM Migration

Section Take-Aways

1. Infrastructure as Code (IAC) allows you to deploy Azure resources in an automated, repeatable way
2. Azure Migrate (v2) is recommended for agent-less migration – of VM Lift & Shift scenarios
3. Azure Site Recovery (ASR) performs the actual VM replication process, and can be used out of Azure Migrate, or as a stand-alone service for Lift & Shift migrations

Questions Landing Spot

“...If you want good answers,
ask better questions...”

© Randy Glasbergen



Azure

Next Module...

(SQL) Database Migration



Thank You