



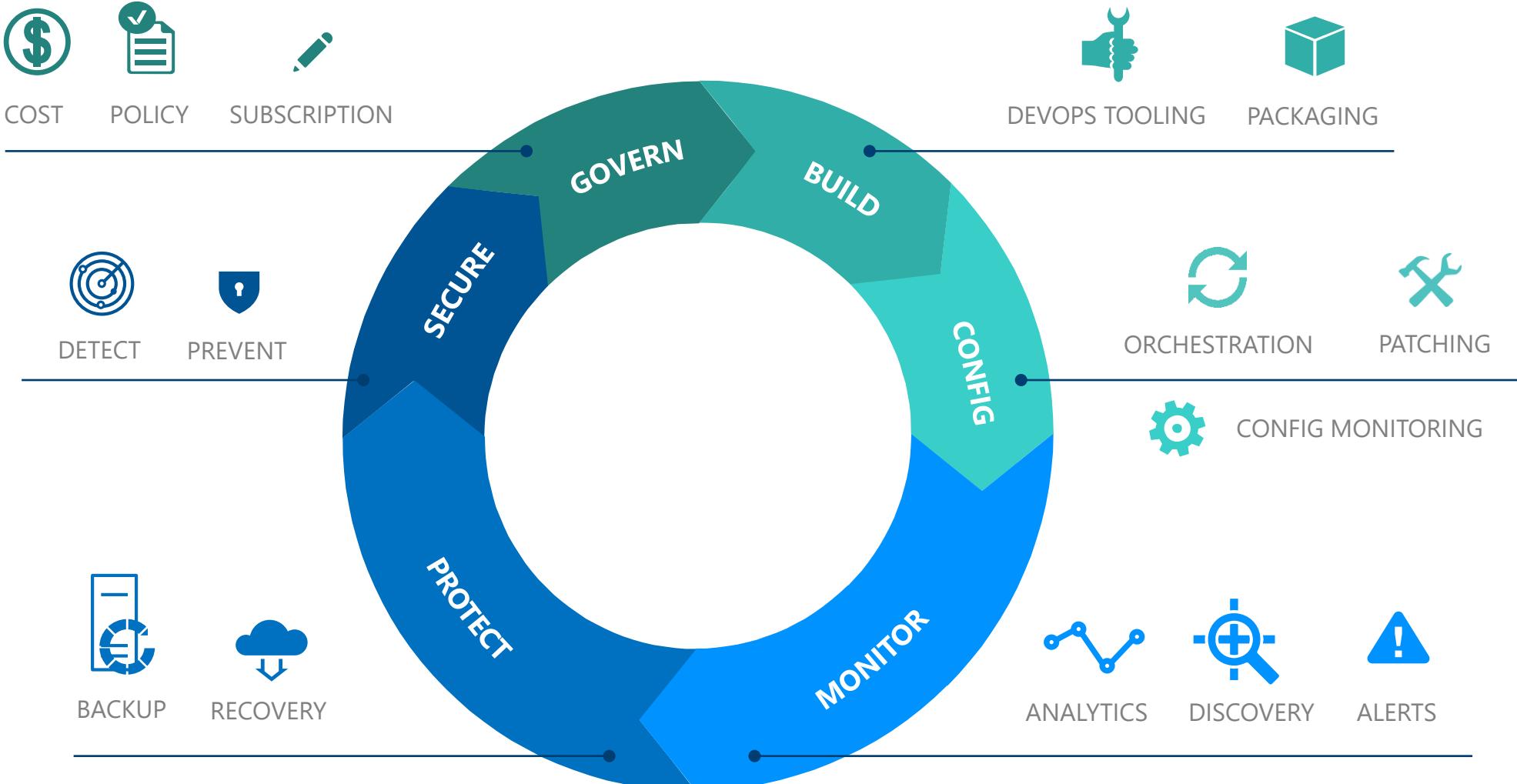
# Infrastructure as Code and Modern ITSM

Simon Schwingel  
Cloud Solution Architect  
Microsoft

# Agenda

- Why modern IT service management?
- Infrastructure as Code
  - Overview Azure Resource Manager
  - Implement ARM templates
  - Design role-based access control
  - Control access

# Operational Consistency



# The customer divide



- Mostly manual
- Alert/ticketing driven
- No ideas what state the environment is in
- Constantly „saving the day“

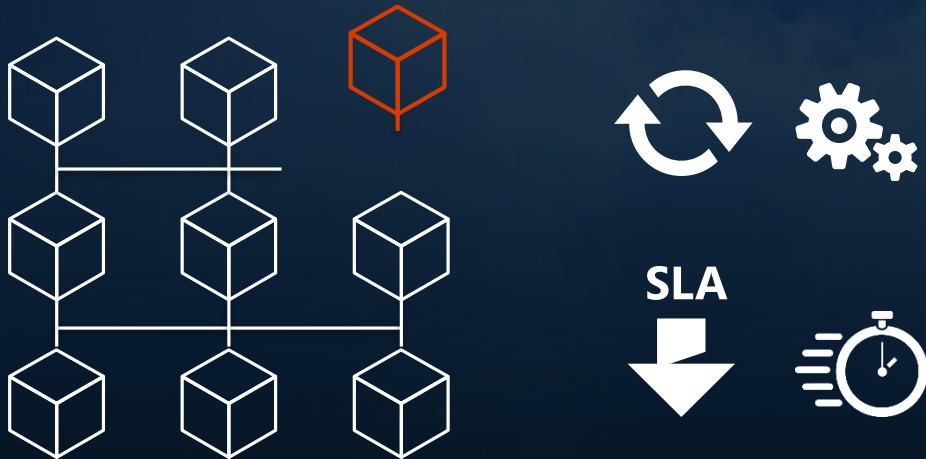
Traditional

Competitive Advantage

Modern

- Mostly automated
- Data driven
- Source Control and Continuous Integration
- Change is normal

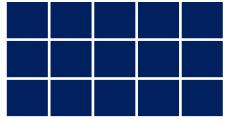
# What makes it a competitive advantage?



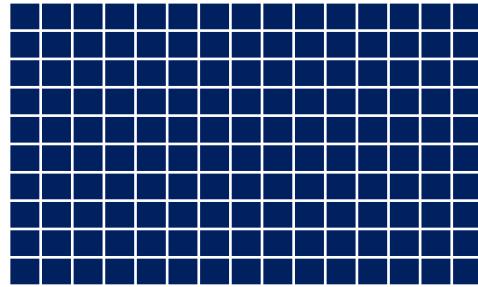
Resolving infrastructure and application issues can be time-consuming and can impact SLA.

- React faster to change
- Increase velocity to market
- Reduce impact of outages
- Improve security posture
- Build employee confidence

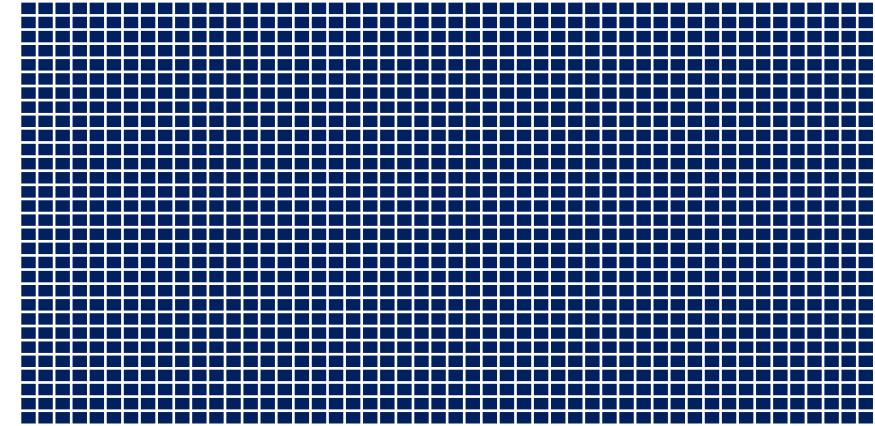
# Making changes manually



15 Servers: Sort of works



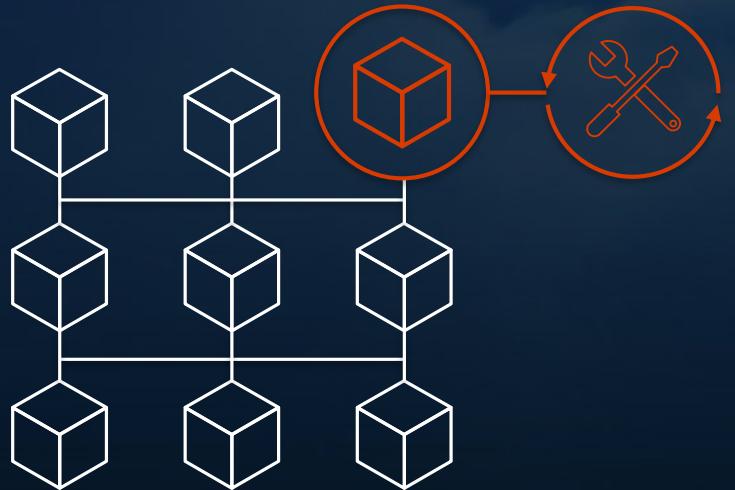
150 Servers: Unconscious  
deception



1500 Servers: Self destructive

The struggle of traditional IT management

# What is modern IT Management



- Everything as code in source control
- Testing as framework for trusted release
- Changes ONLY come from build
- Servers are immutable or incremental with recovery plan
- Read-only access on demand
- Data-driven decisions
- Anything not core value shifts to service
- Contributing to/consuming from community
- Operations is valued contributor with application owners (DevOps)

The background of the slide features a perspective view of a server room filled with tall, dark server racks. A large, fluffy white cloud icon is positioned centrally in the middle ground.

# Infrastructure as Code

## Overview Azure Resource Manager

# Azure Resource Manager (ARM)

Successor of Azure Service Manager (ASM)

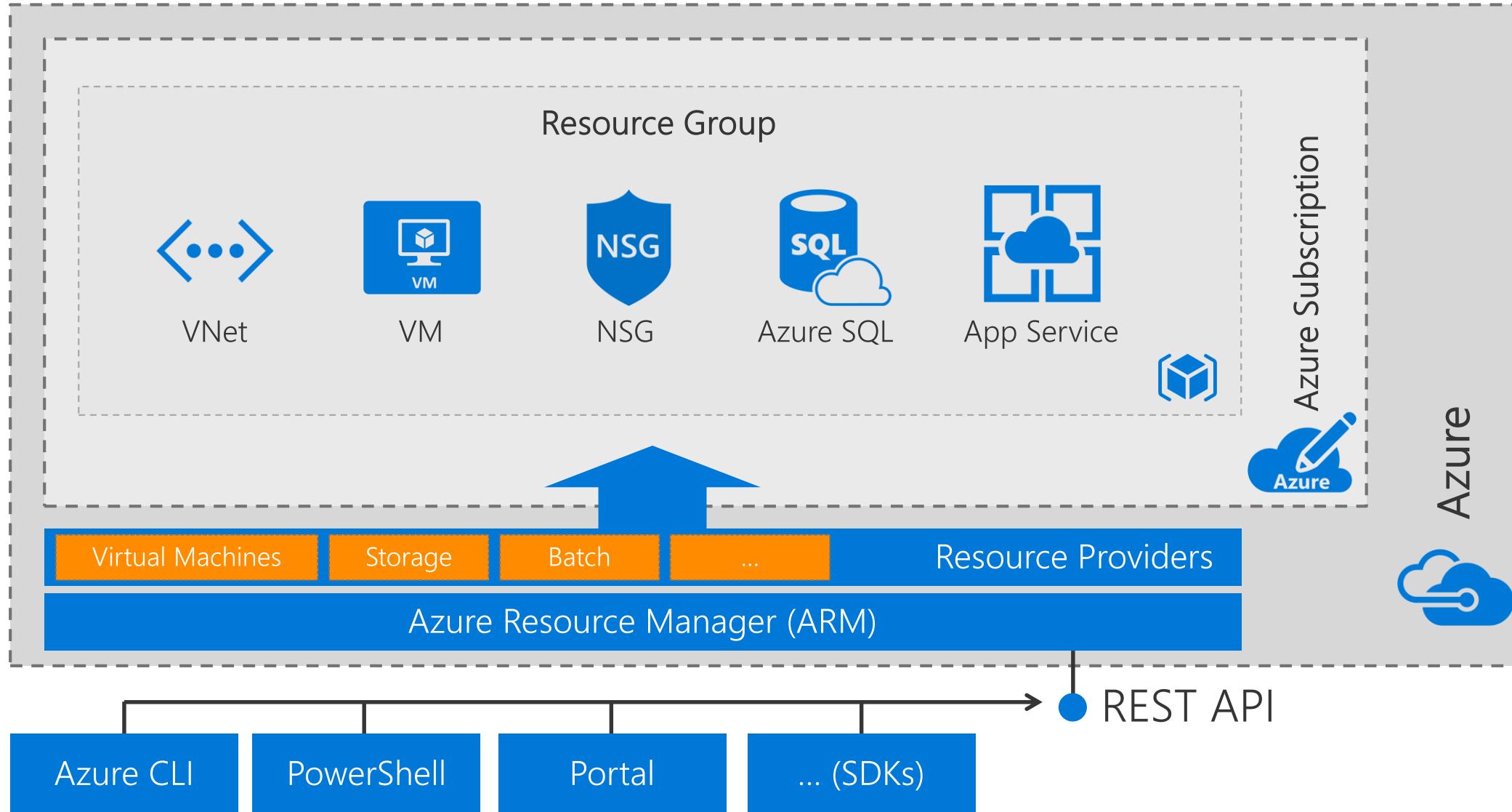
Went GA in 2015

Uses modular approach towards resources

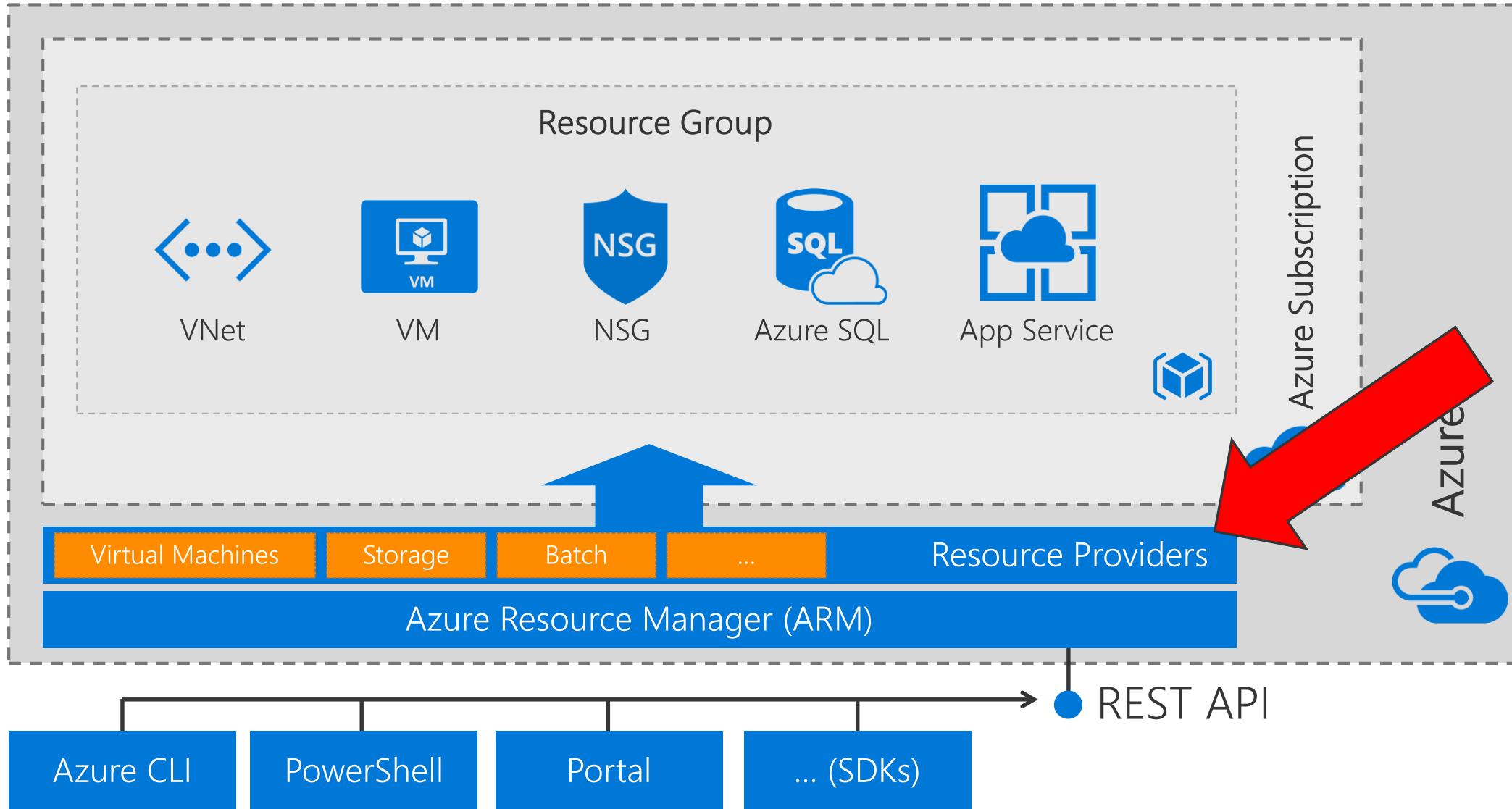
## *Benefits:*

- Features built-in role-based access control (RBAC)
- Eases management by grouping resources
- Supports Template-based deployments

# Azure Resource Manager - Architecture



# Resource Providers



# Resource Provider - Simplified Structure

Namespace: Microsoft.Compute

Locations: East US, East US 2, ...

Resources:  
availabilitySets  
virtualMachines  
...

Operations:  
Microsoft.Compute/availabilitySets/delete  
Microsoft.Compute/virtualMachines/vmSizes/read  
Microsoft.Compute/virtualMachines/start/action  
...

API Versions: 2017-03-30, 2016-08-30, ...



Shift+Space to toggle favorites

resou

[Edit dashboard](#) [Share](#) [Fullscreen](#) [Clone](#) [Delete](#)

Amsterdam Edit

12:24

FRIDAY, AUGUST 18, 2017



Marketplace



Help + support

All resources

Help + support

Keywords: Resource health

Resource Explorer

Resource groups

Keywords: resources

Subscriptions

Keywords: resources



Service Health

Personalized guidance and support when issues  
in Azure services affect you. [Learn more](#)

## Resource Explorer

microsoft



## Providers (Response Time 680ms)

/providers?api-version=2014-04-01-preview

```
1 [ {  
2     "value": [  
3         {  
4             "namespace": "84codes.CloudAMQP",  
5             "resourceTypes": [  
6                 {  
7                     "resourceType": "servers",  
8                     "locations": [  
9                         "East US 2",  
10                        "Central US",  
11                        "East US",  
12                        "North Central US",  
13                        "South Central US",  
14                        "West US",  
15                        "North Europe",  
16                        "West Europe",  
17                        "East Asia",  
18                        "Southeast Asia",  
19                        "Japan East",  
20                        "Japan West",  
21                        "Australia East",  
22                        "Australia Southeast",  
23                        "West US (Partner)"
```

# List Available Resource Providers

```
# Azure CLI 2.0
```

```
az provider list --out table
```

```
# PowerShell
```

```
Get-AzureRmResourceProvider -ListAvailable |  
    Select-Object ProviderNamespace, RegistrationState
```

# Resource Provider List

```
# Azure CLI 2.0  
az provider list --out table
```

Namespace	RegistrationState
Microsoft.Advisor	Registered
Microsoft.AnalysisServices	Registered
Microsoft.ApiManagement	Registered
Microsoft.AppService	Registered
Microsoft.Authorization	Registered
Microsoft.Automation	Registered
Microsoft.AzureActiveDirectory	Registered
[...]	

# Resources Types of a Resource Provider

```
# Azure CLI 2.0
az provider show --namespace Microsoft.Compute \
--query "resourceTypes[*].resourceType" --out table
```

## Result

---

```
-----
availabilitySets
virtualMachines
virtualMachines/extensions
virtualMachineScaleSets
[...]
```

# API Versions of Resource Types

```
# Azure CLI 2.0
az provider show --namespace Microsoft.Compute \
--query "resourceTypes[?resourceType=='virtualMachines'].apiVersions | [0]" \
--out table
```

## Result

---

```
-----
2017-03-30
2016-08-30
2016-04-30-preview
2016-03-30
2015-06-15
2015-05-01-preview
```

# Operations on Resource Types

```
# Azure CLI 2.0
az provider operation show --namespace Microsoft.Compute \
--query "resourceTypes[?name=='virtualMachines'].operations[].name" --out table
```

## Result

---

```
...
Microsoft.Compute/virtualMachines/start/action
Microsoft.Compute/virtualMachines/powerOff/action
Microsoft.Compute/virtualMachines/redeploy/action
Microsoft.Compute/virtualMachines/restart/action
Microsoft.Compute/virtualMachines/deallocate/action
Microsoft.Compute/virtualMachines/generalize/action
...
```

# Registering Resource Providers

```
# Azure CLI 2.0
```

```
az provider register -n "Microsoft.ApiManagement"
```

```
# Azure PowerShell
```

```
Register-AzureRmResourceProvider -ProviderNamespace "Microsoft.ApiManagement"
```

# Resource Provider - Simplified Structure

Namespace: Microsoft.Compute

Locations: East US, East US 2, ...

Resources:  
availabilitySets  
virtualMachines  
...

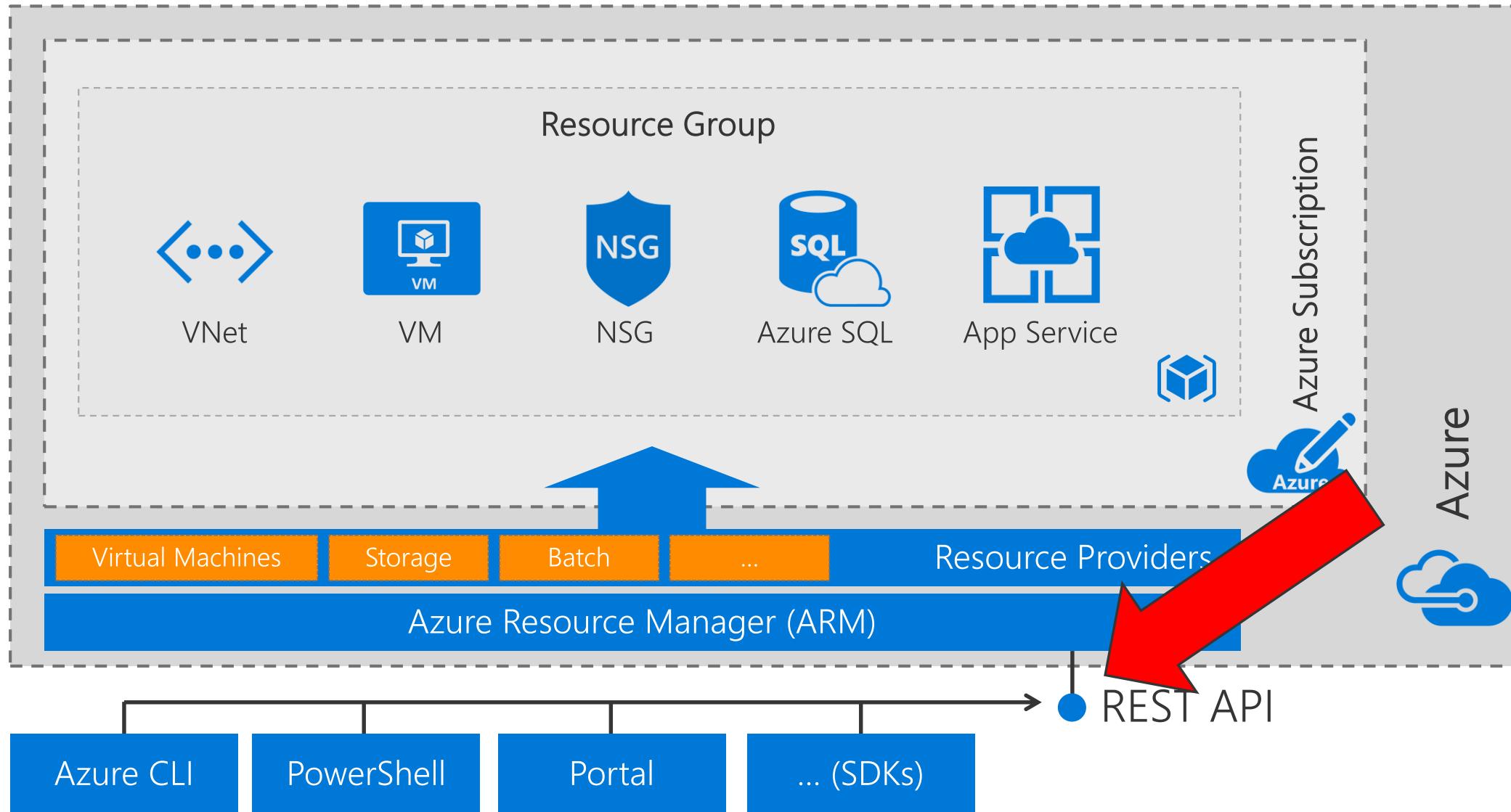
Operations:  
Microsoft.Compute/availabilitySets/delete  
Microsoft.Compute/virtualMachines/vmSizes/read  
Microsoft.Compute/virtualMachines/start/action  
...

API Versions: 2017-03-30, 2016-08-30, ...



An extremely powerful concept!

# Azure Resource Manager REST API



# Elements of an Azure REST API Call

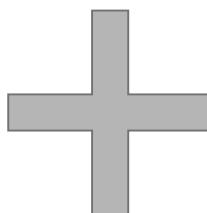
1. Base URL (varies amongst Sovereign Clouds)
2. Authentication Header - OAuth Token (later)
3. Resource (Resource Provider) that should be targeted
4. API Version that should be called

# REST Example: Create a Resource Group



```
PUT https://management.azure.com/subscriptions/{subscriptionId}/  
resourcegroups/{resourceGroupName}?api-version=2017-05-10
```

Authorization: Bearer {TOKEN}  
Content-Type: application/json



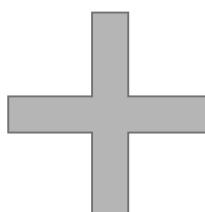
```
{  
  "location": "northeurope",  
  "tags": {  
    "tagname1": "test-tag"  
  }  
}
```

# REST Example: Create a Storage Account



```
PUT https://management.azure.com/subscriptions/{subscriptionId}/  
resourcegroups/{resourceGroupName}/  
providers/Microsoft.Storage/storageAccounts/{storageAccountName}  
?api-version=2016-12-01
```

Authorization: Bearer {TOKEN}  
Content-Type: application/json



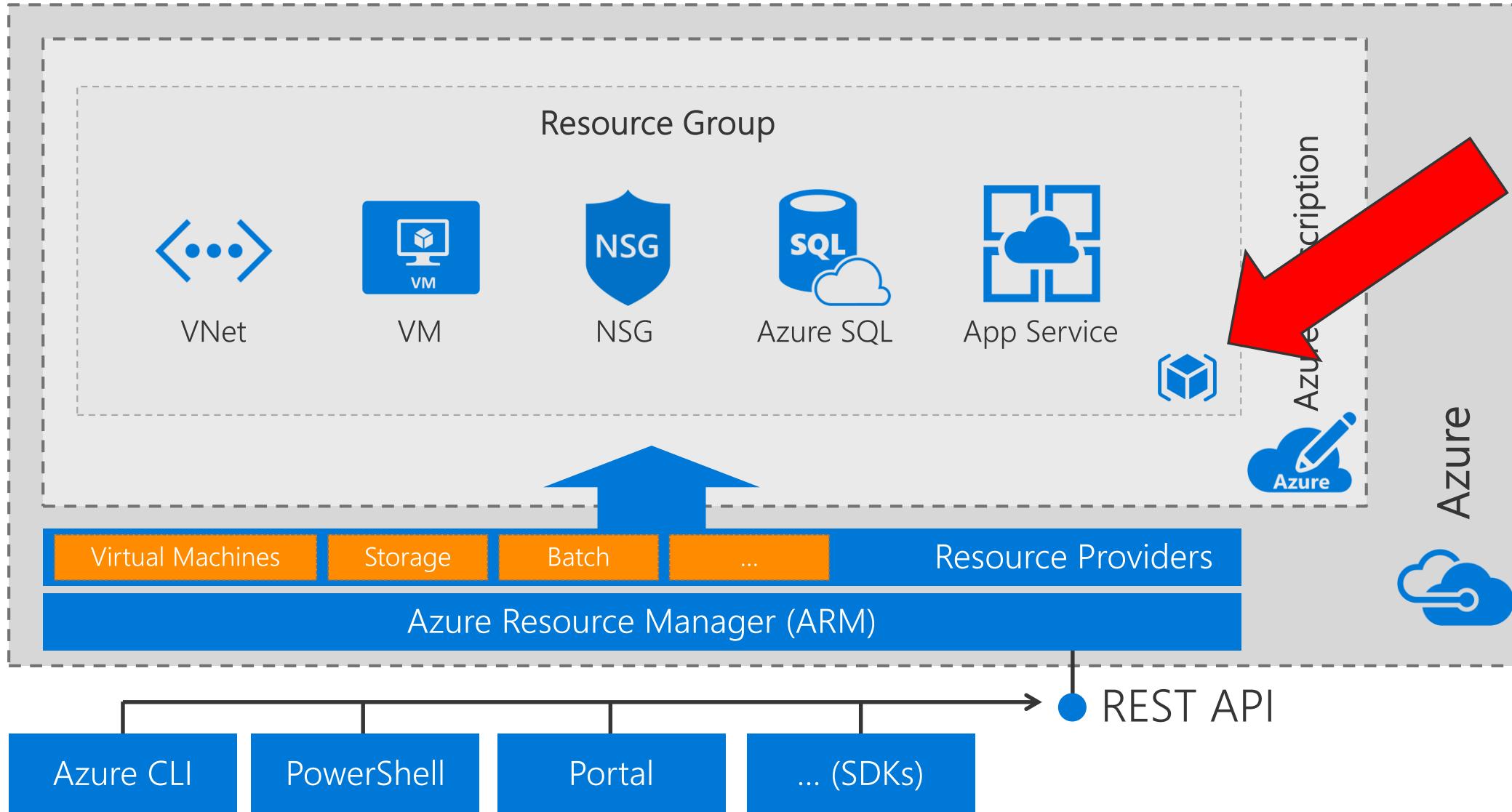
```
{  
  "location": "WestEurope",  
  "tags": {},  
  "sku": {  
    "name": "Standard_LRS"  
  }  
}
```



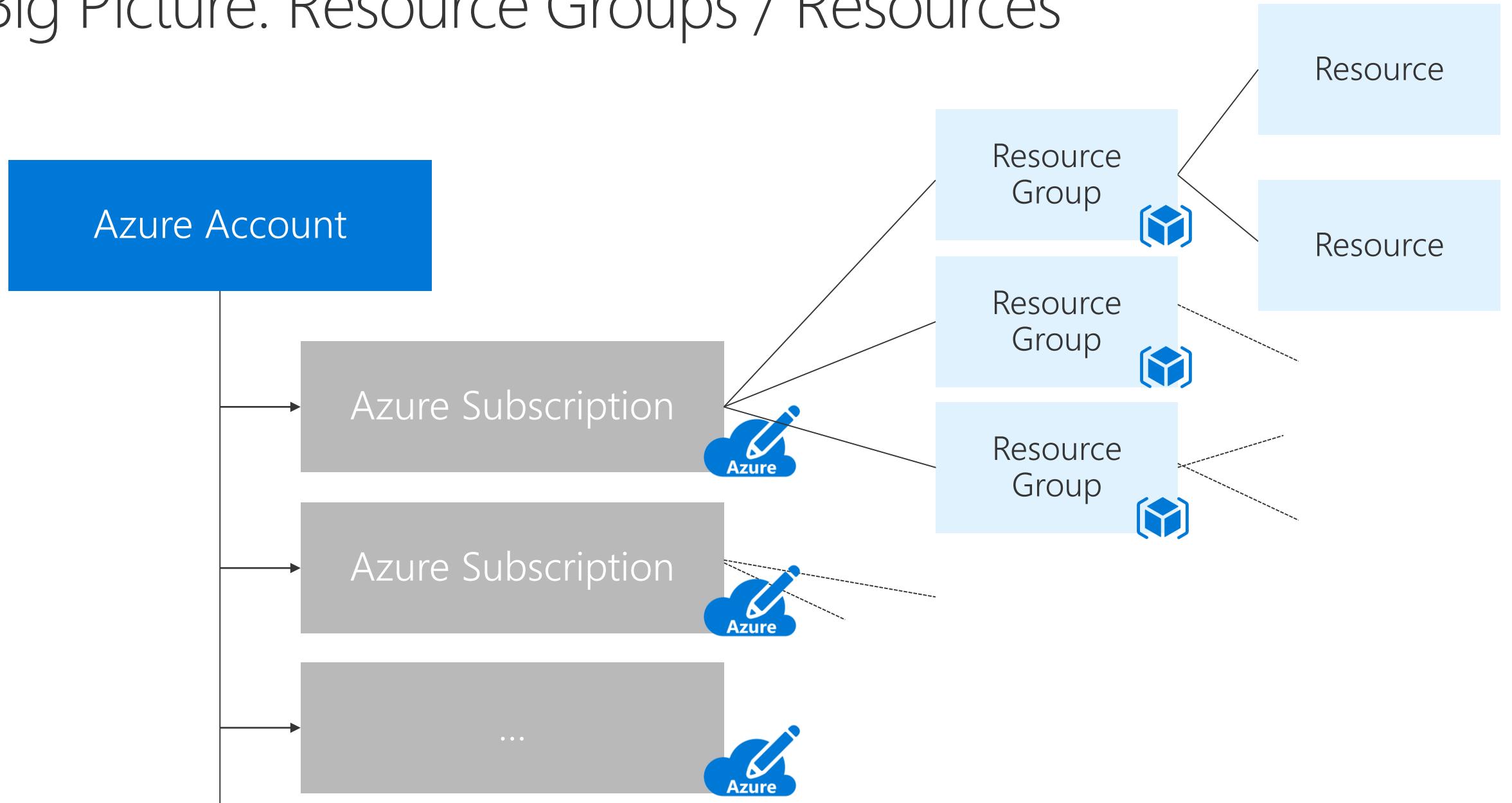
# Infrastructure as Code

## Resource Groups

# Resource Providers – Resource Groups



# Big Picture: Resource Groups / Resources



# Resource Group are Management Units

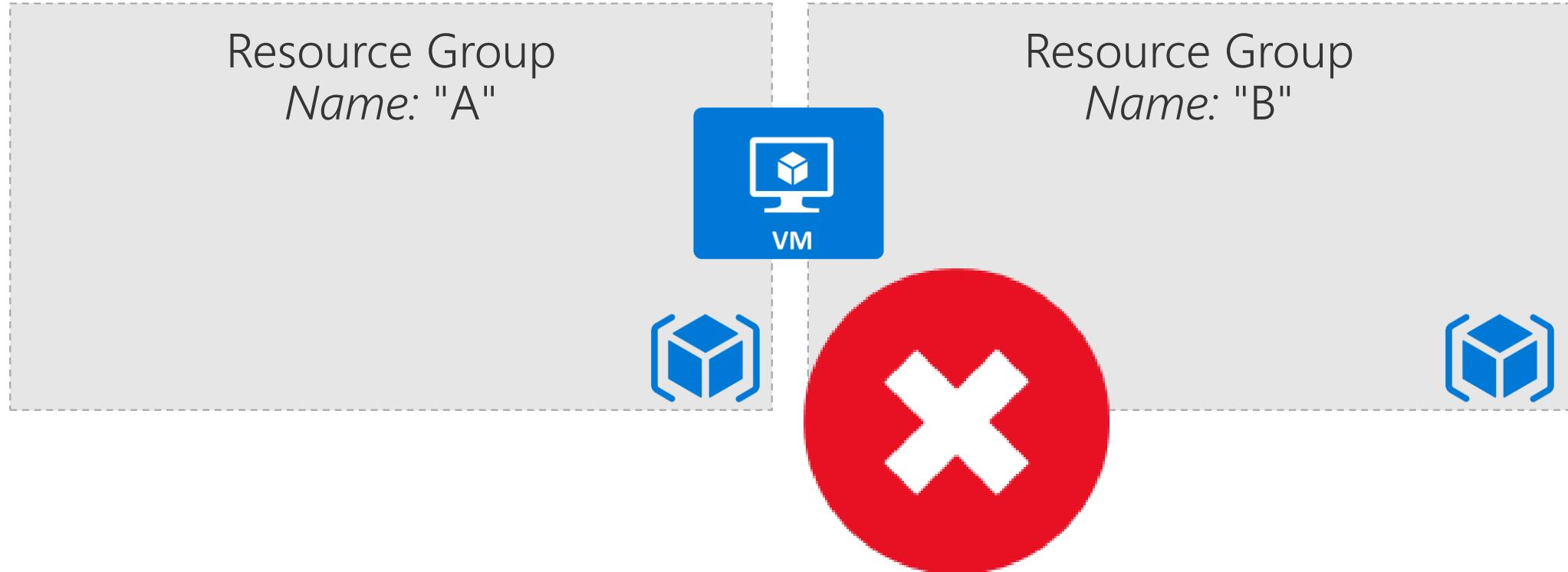
- Organize resources that should share the same life cycle
- Features tags and locks
- Supported as billing filter
- Can be used to scope RBAC



# A Resource always belongs to a Resource Group



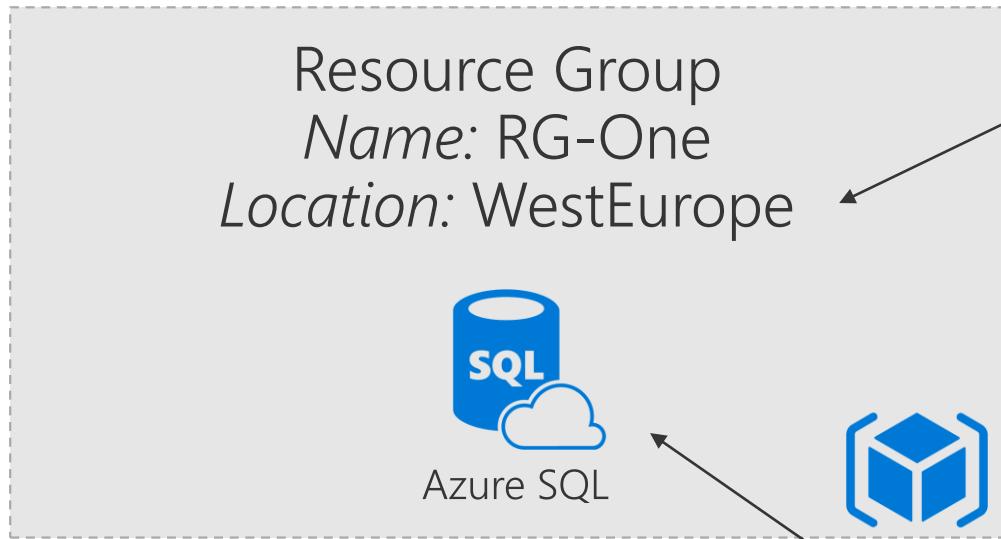
# A Resource always belongs to a single Resource Group



# Resource Groups cannot be nested



# Resource Groups have a Location Property



Defines where Metadata is stored

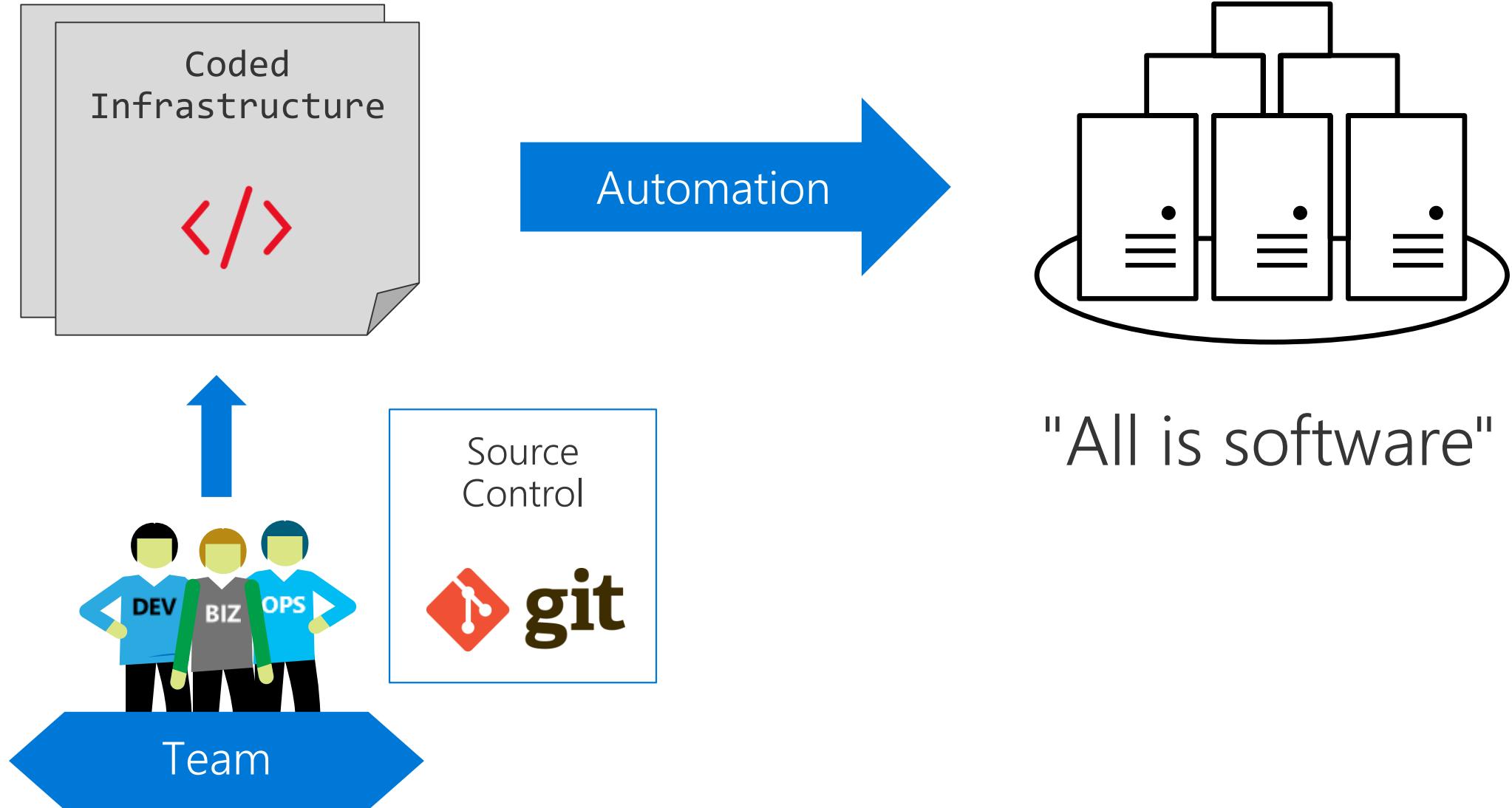
Resource Location can be different  
(you should avoid that)

The background of the slide features a perspective view of a server room. Rows of server racks are visible on both sides, receding into the distance. A large, fluffy white cloud graphic is positioned in the center of the room, partially obscuring the racks. The overall color palette is dark, with blue and grey tones.

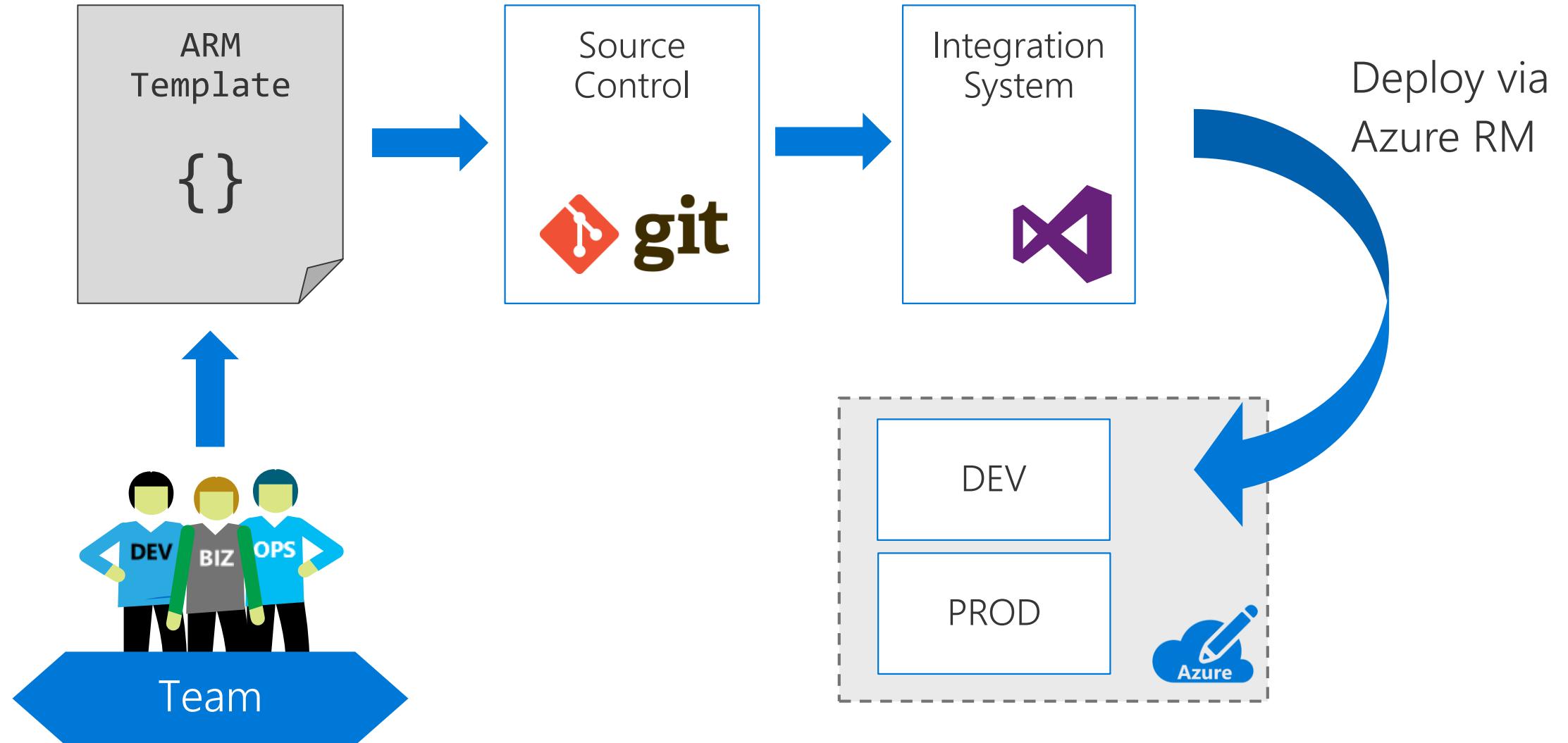
# Infrastructure as Code

Implement ARM Templates

# Big Picture: Infrastructure-As-Code

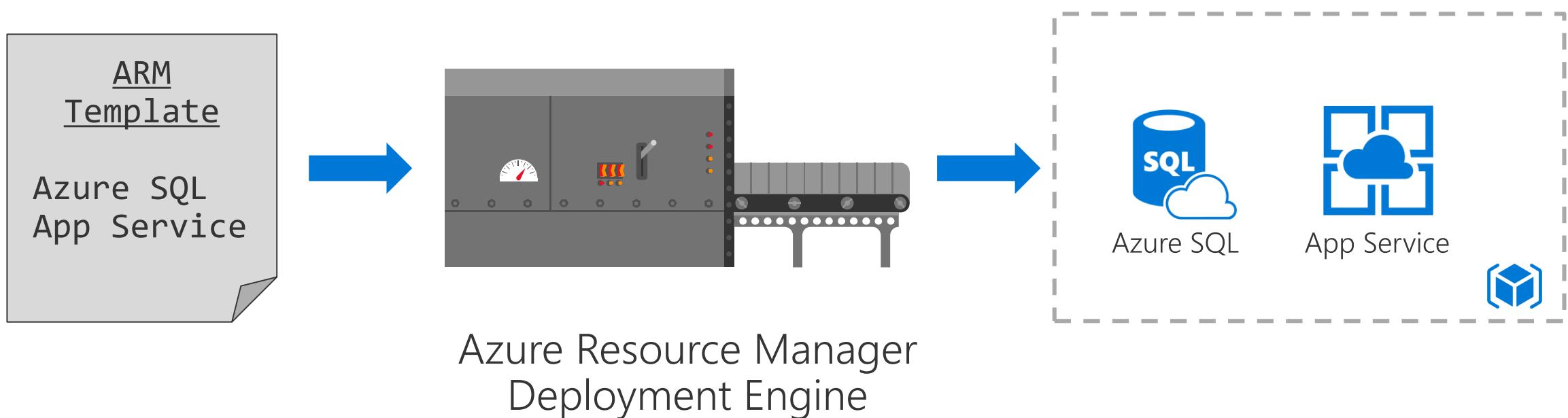


# ARM Templates - Infrastructure as Code



# ARM Templates

Declarative JSON-based template deployments in Azure



# Imperative

```
$ni = New-AzureRmNetworkInterface [...]  
$pip = New-AzureRmPublicIPAddress [...]  
$conf = New-AzureRmVMConfig -Pip $pip [...]  
[...]  
New-AzureRmVirtualMachine -Config $conf
```

"I tell you exactly what you should do."

# Declarative

```
{  
    "type": "virtualMachine",  
    "size": "Standard_DS3"  
    "networkInterface": "..."  
    [...]  
}
```

"I describe to you what I would like to have."

# Working with ARM Templates

- Azure Portal
- Visual Studio
- Visual Studio Code
- ... basically any text editor

# ARM Template Structure

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

ARM Template schemas in detail:

<https://github.com/Azure/azure-resource-manager-schemas>

# ARM Template Parameters

```
"parameters": {  
    "blob_storage_name_prefix": {  
        "type": "string",  
        "defaultValue": "foobar",  
        "minLength": 1,  
        "maxLength": 11,  
        "metadata": {  
            "description": "Blob Storage account name prefix."  
        }  
    },  
    [...]  
}
```

# ARM Template Parameters (cont.)

```
"parameters": {  
    "blob_storage_sku": {  
        "type": "string",  
        "defaultValue": "Standard_LRS",  
        "allowedValues": [  
            "Standard_LRS",  
            "Standard_GRS"  
        ],  
        "metadata": {  
            "description": "The storage account's SKU. Only LRS and GRS are allowed."  
        }  
    },  
    [...]  
}
```

# ARM Template Variables

```
"variables": {  
    "blob_storage_name":  
        "[concat(  
            parameters('blob_storage_name_prefix'),  
            uniqueString('resourceGroup().id')  
        )]"  
},
```

# ARM Template Resources

```
"resources": [  
    {  
        "name": "[variables('blob_storage_name')]",  
        "type": "Microsoft.Storage/storageAccounts",  
        "apiVersion": "2016-01-01",  
        "sku": {  
            "name": "[parameters('blob_storage_sku')]"  
        },  
        "kind": "BlobStorage",  
        "location": "[parameters('location')]"  
    }  
],
```

# ARM Template Outputs

```
"outputs": {  
    "blob_storage_account_name": {  
        "type": "string",  
        "value": "[variables('blob_storage_name')]"  
    },  
    "blob_storage_account_url": {  
        "type": "string",  
        "value": "[reference(  
            resourceId('Microsoft.Storage/storageAccounts',  
            variables('blob_storage_name'))).primaryEndpoints.blob]"  
    }  
}
```

# ARM Template Outputs (cont.)

// Result

```
[...]
  "outputs": {
    "blob_storage_account_name": {
      "type": "String",
      "value": "contosogd3f7mnjwpuyu"
    },
    "blob_storage_account_url": {
      "type": "String",
      "value": "https://contosogd3f7mnjwpuyu.blob.core.windows.net/"
    }
  },
  [...]
```

# Most Common ARM Template Functions

```
concat()          # Concatenate two or more strings  
resourceId()     # Get resource ID by name  
uniqueString()   # Generate a 13-char hash from input  
  
reference()      # Returns object with resource runtime state  
  
resourceGroup()  # Get resource group ID of deployment  
subscription()   # Get subscription group ID of deployment
```

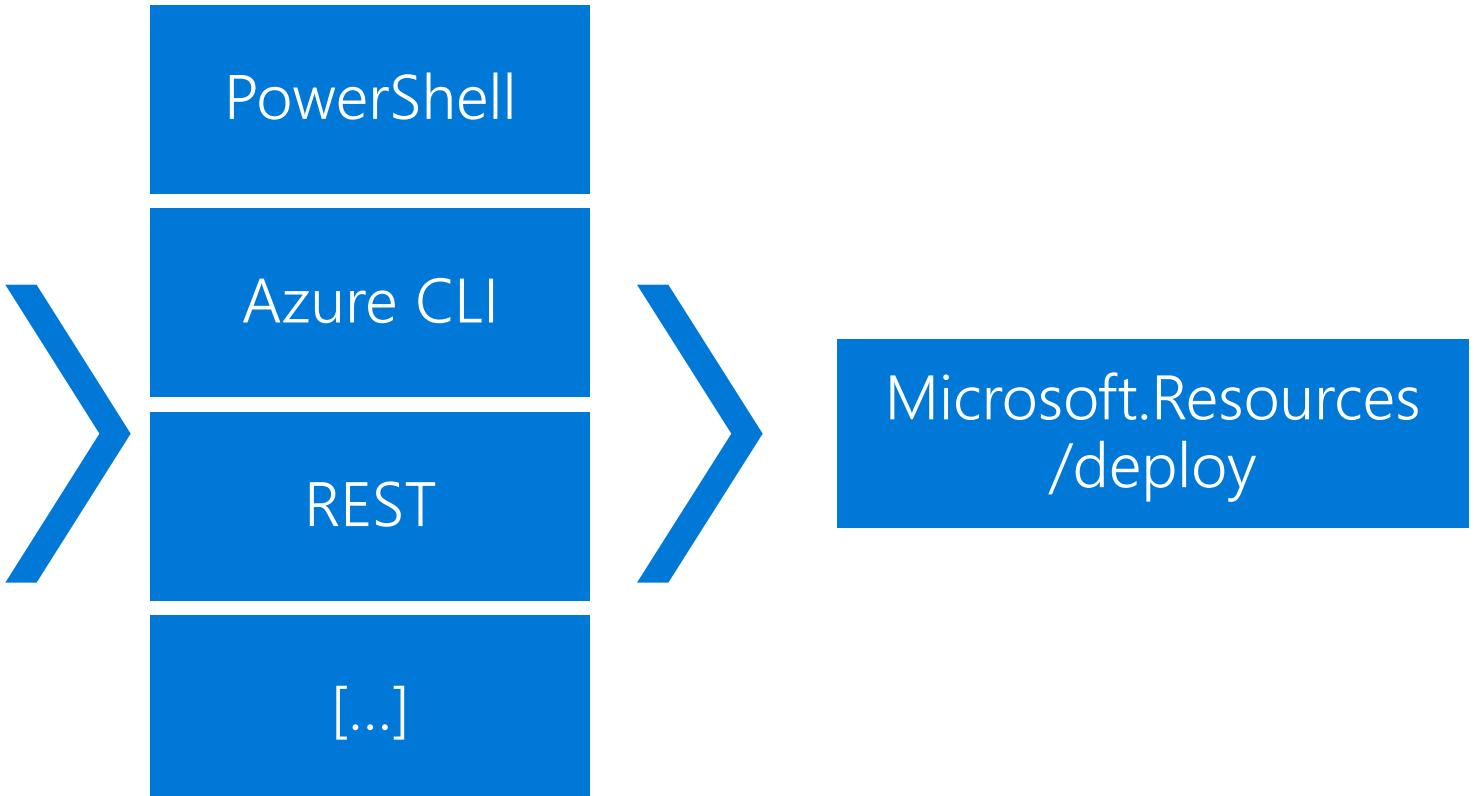
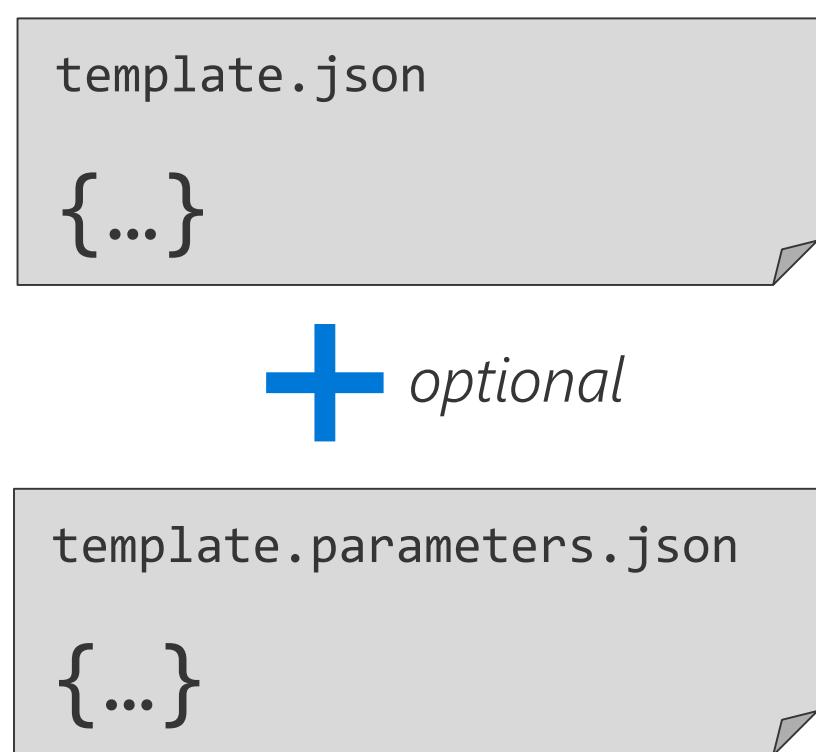
# Advanced ARM Templates

Copy Loops ([Link](#))

Conditionals ([Link](#))

Linked Templates ([Link](#))

# Deploying ARM Templates



Input

Tools

Provider

# Deploying with Azure CLI 2.0

```
# Prerequisite: An existing resource group
```

```
az group create -n "storage_deploy_test_rg" -l "WestEurope"
```

```
# Actual Deployment
```

```
az group deployment create --resource-group "storage_deploy_test_rg" \
--template-file storage.json
--name MyDeployment
--parameters blob_storage_name_prefix=contoso location=WestEurope
```

# Deploying with PowerShell

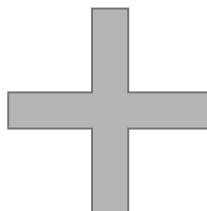
```
# Prerequisite: An existing resource group  
New-AzureRmResourceGroup -Name storage_deploy_test_rg -Location WestEurope  
  
# Actual Deployment  
$params = @{blob_storage_name_prefix="contoso";location="WestEurope"}  
New-AzureRmResourceGroupDeployment \  
    -ResourceGroupName "storage_deploy_test_rg" \  
    -TemplateFile .\storage.json \  
    -Name MyDeployment  
    -TemplateParameterObject $params
```

# Deploying with REST

# Prerequisite: OAuth Authorization (Bearer Token)

```
PUT https://management.azure.com/subscriptions/{subscriptionId}/  
    resourcegroups/{resourceGroupId}/  
    providers/microsoft.resources/  
    deployments/{deploymentName}?api-version=2016-09-01
```

Authorization: Bearer {TOKEN}  
Content-Type: application/json



```
{  
  properties: {  
    parameters: {...}  
    template: {...}  
  }  
}
```

# Deploy from Portal

The screenshot shows the Microsoft Azure portal interface. On the left, there's a sidebar with various service icons: New, Dashboard, Resource groups, Azure Active Directory, Subscriptions, Templates, and More services >. The main area is titled "Template deployment" and features a Microsoft logo icon. It contains descriptive text about Azure Resource Manager templates, a "Create" button at the bottom, and social sharing icons for Twitter, Facebook, LinkedIn, YouTube, Google+, and Email.

Microsoft Azure   New   Template deployment

Template deployment  
Microsoft

Applications running in Microsoft Azure usually rely on a combination of resources, like databases, servers, and web apps. Azure Resource Manager templates enable you to deploy and manage these resources as a group, using a JSON description of the resources and their deployment settings.

Edit your template with IntelliSense and deploy it to a new or existing resource group.

PUBLISHER Microsoft

LOGICAPPSSUPPORTED none

USEFUL LINKS [Documentation](#)

Create

# Demo: Deploying an ARM Template

Simon Schwingel

# Azure Marketplace

The screenshot shows the Azure Marketplace interface. At the top, there's a blue header bar with the 'Azure Marketplace' logo, 'Browse', 'Sell', and 'Learn' buttons. Below the header, a sidebar on the left lists various product categories: Compute, Networking, Storage, Web + Mobile, Databases, Intelligence + analytics, Internet of Things, Enterprise Integration, Security + Identity, Developer tools, Monitoring + Management, Add-ons, Containers, Blockchain, and Azure Active Directory apps. A search bar at the top right contains the text 'elastic' with a clear button. The main area displays 'Product results (9)' for 'Elasticsearch'. An item card for 'Elasticsearch' by 'Elastic' is shown, featuring a small icon, the product name, the developer name, a brief description ('From the makers of Elasticsearch - Elasticsearch, Kibana and X-Pack allow searching data at scale'), a note that the price varies, and a 'Get it now' button.



The screenshot shows a configuration wizard titled 'Create Elastic Stack - (Elastic...)' with a 'Basics' tab selected. The wizard consists of eight numbered steps:

- 1 Basics: Configure basic settings
- 2 Cluster Settings: Required
- 3 Nodes Configuration: Required
- 4 User Configuration: Required
- 5 External Access: Required
- 6 User Information: Provide user information
- 7 Summary: Elastic Stack - (Elasticsearch, Ki...
- 8 Buy

On the right side of the wizard, there are several input fields and dropdown menus:

- 'User name' (required)
- 'Authentication type': Password or SSH public key
- 'Password' (required)
- 'Confirm password'
- 'Subscription': Microsoft Azure Internal Consumption
- 'Resource group': Create new (radio button selected) or Use existing
- 'Location': West Europe

A red box highlights the 'Resource group' field, which has an error message: 'The value should not be empty.'

# ARM Quickstart Templates

Azure / azure-quickstart-templates

Code Issues Pull requests Projects Wiki Insights

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

azur... templates arm

15,243 commits 5 branches 0 releases 539 contributors

Branch: master New pull request Create new file Upload files

gbowerman committed on GitHub Merge pull request #3856 from jboeshart/201-vmss-windows-webapp-dsc-a... Late

.github fix typo in link to checklist

1-CONTRIBUTION-GUIDE Git commands fixed with valid uri

100-blank-template Add some missing words

101-1vm-2nics-2subnets-1vnet update from CI failures

101-aci-linuxcontainer-public-ip Adding descriptions

101-acs-dcos fix travis error

101-acs-kubernetes fix travis error

101-acs-swarm fix travis error

101-acsengine-swarmmode correcting arm templates of type and names

101-app-service-certificate-standard correcting arm templates of type and names

101-app-service-certificate-wildcard Fixing formatting issues in README.md and typos in template files

Quickstart Templates Gallery:  
<https://azure.microsoft.com/en-us/resources/templates/>

Quickstart Templates on GitHub  
<https://github.com/Azure/azure-quickstart-templates>

>580 samples

# Managing Private ARM Template in the Portal

The screenshot shows the Microsoft Azure Templates portal. On the left, there's a sidebar with options like 'New', 'Resource groups', 'All resources', 'Recent', 'App Services', 'Virtual machines (classic)', and 'Virtual machines'. The main area is titled 'Templates' and shows a preview of 'mytemplate'. A red arrow points to the '+ Add' button at the top left of the template list.

The screenshot shows the Microsoft Azure portal's navigation bar. The 'Templates' option is highlighted with a red arrow, indicating it's the active service.

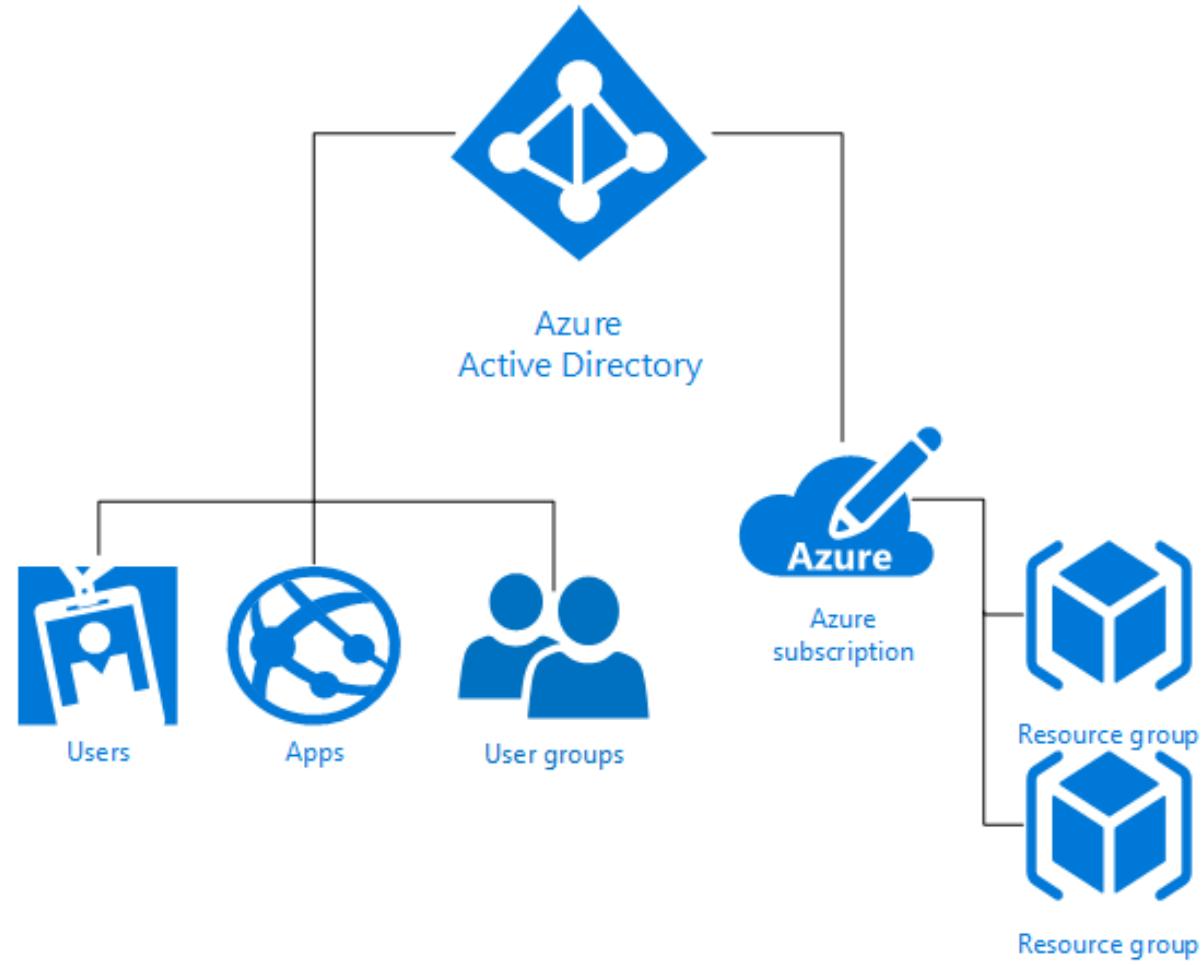
The screenshot shows the 'Add template' dialog box. It has two main sections: 'General' and 'ARM Template'. The 'General' section contains fields for 'Name' and 'Description'. The 'ARM Template' section contains a 'Add template' button. Both sections have a 'Add general information' and 'Add template' link respectively.



Infrastructure as  
Code

Design Access Control

# Basics of Access Management in Azure



# Core Built-in Roles

## Owner



- Full Access
- Delegate to others

## Contributor



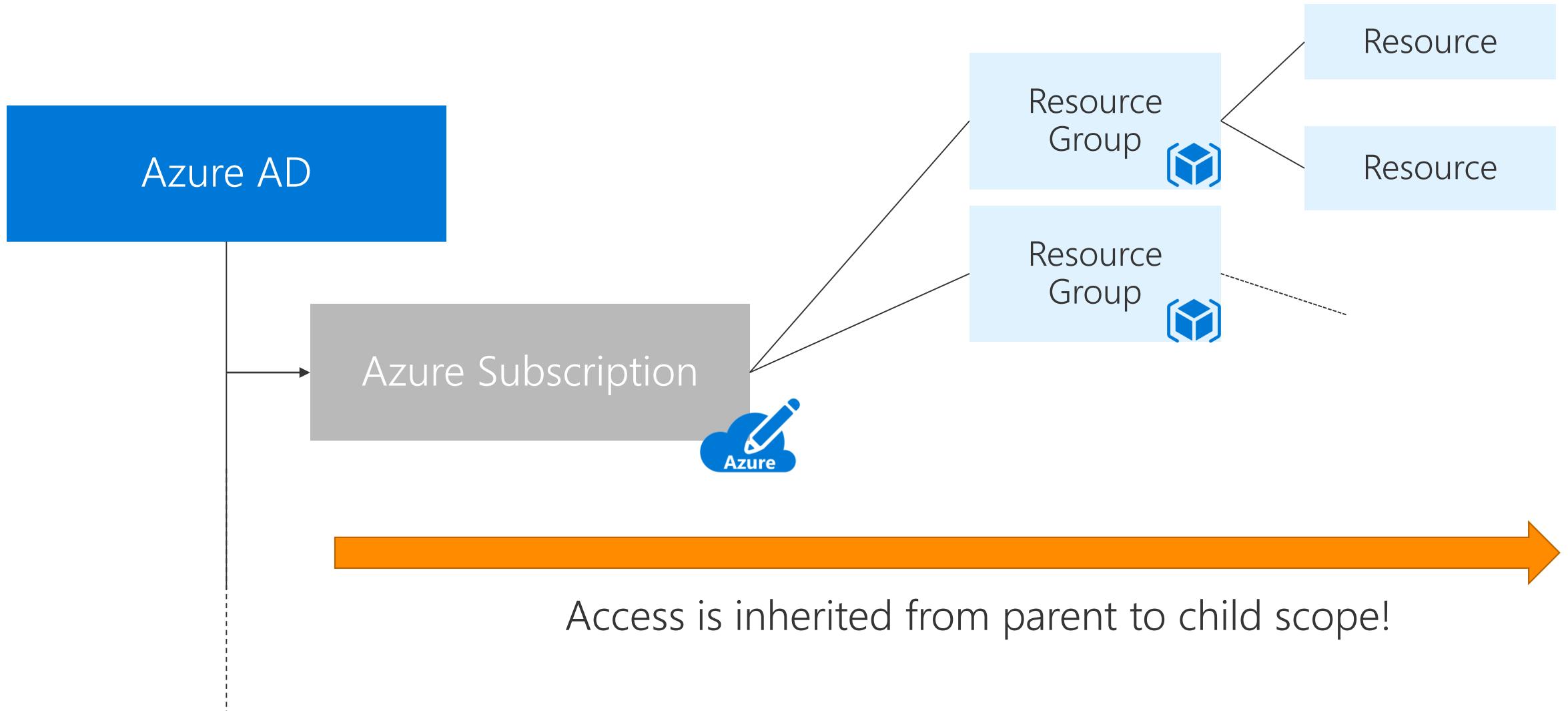
- Full Access
- Cannot delegate to others

## Reader



- Read Only

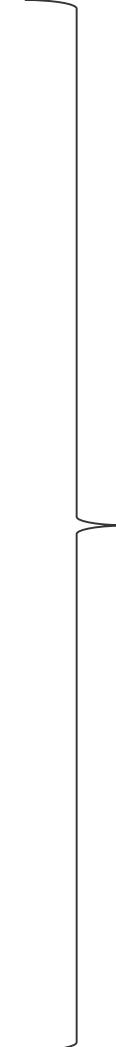
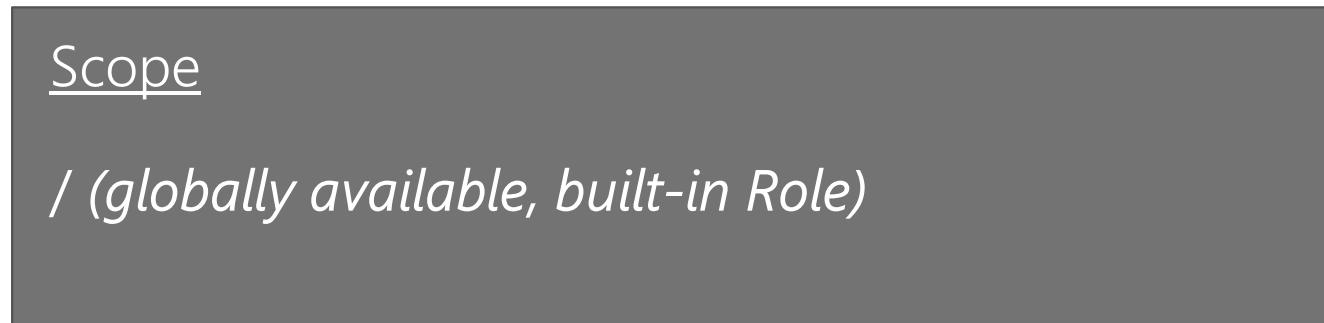
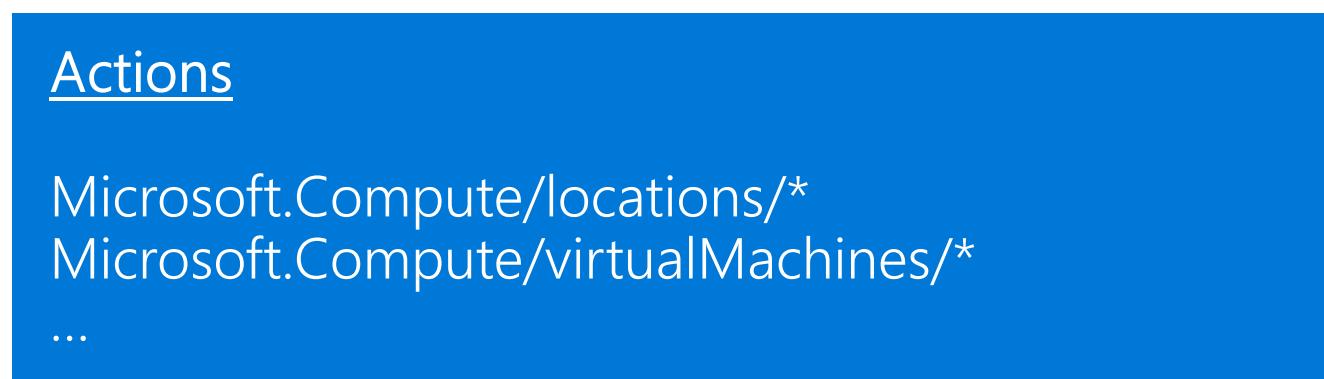
# Resource Hierarchy and Access Inheritance



# Demo: Basic RBAC in the Portal

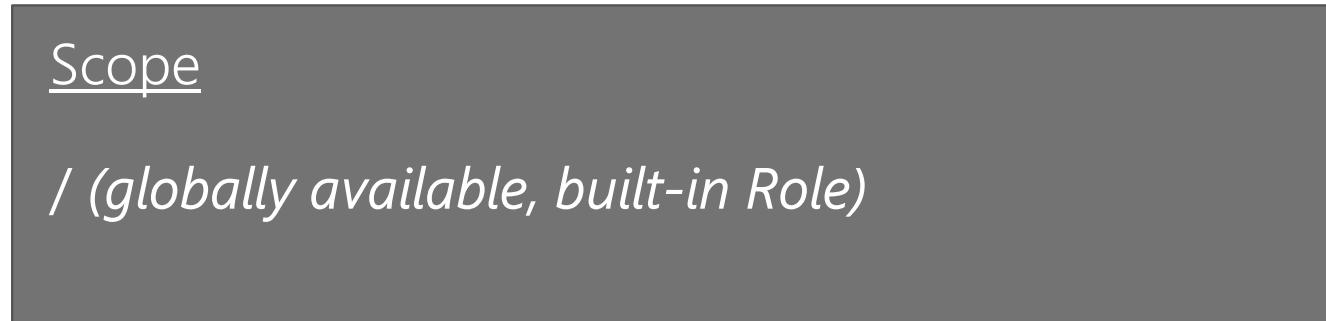
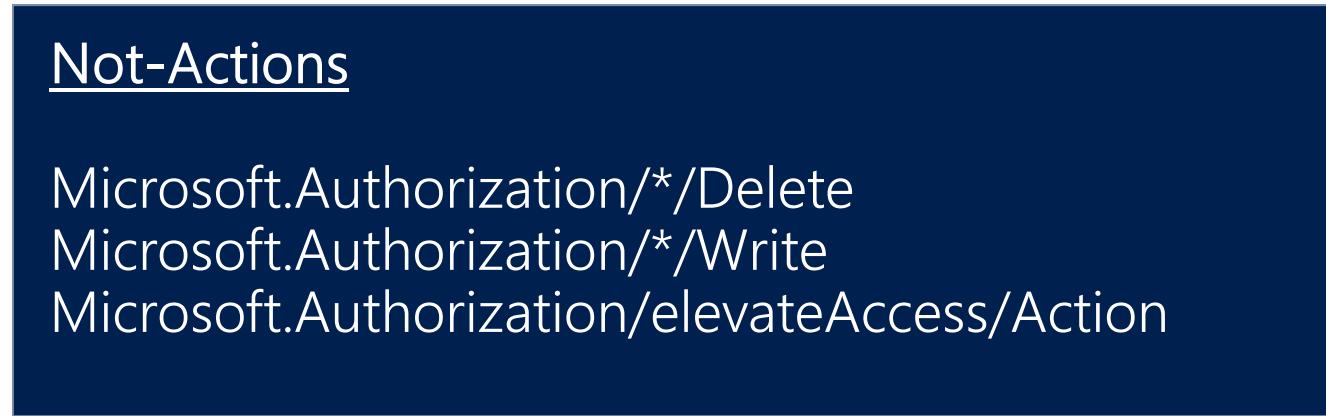
Simon Schwingel

# Basic Anatomy of a Role - VM Contributor



Role:  
Virtual Machine Contributor

# Basic Anatomy of a Role - Contributor



Role:  
Contributor

# Translating RBAC Operations to HTTP Verbs

Write enables you to perform PUT, POST, PATCH, and DELETE operations.

Read enables you to perform GET operations.

# Inspecting Built-In Roles

```
# PowerShell
```

```
Get-AzureRmRoleDefinition | Select-Object -Property Name
```

```
# Result
```

```
Name
```

```
----
```

```
Azure Service Deploy Release Management Contributor
```

```
GenevaWarmPathResourceContributor
```

```
Monitor permissions
```

```
Office DevOps
```

```
OMS Workspace Administrator
```

```
API Management Service Contributor
```

```
API Management Service Operator Role
```

```
[...]
```

# Inspecting Actions of Built-in Roles

```
# PowerShell
```

```
(Get-AzureRmRoleDefinition "Virtual Machine Contributor").Actions
```

```
# Result
```

```
Microsoft.Authorization/*/read
Microsoft.Compute/availabilitySets/*
Microsoft.Compute/locations/*
Microsoft.Compute/virtualMachines/*
Microsoft.Compute/virtualMachineScaleSets/*
Microsoft.DevTestLab/schedules/*
Microsoft.Insights/alertRules/*
Microsoft.Network/applicationGateways/backendAddressPools/join/action
Microsoft.Network/loadBalancers/backendAddressPools/join/action
Microsoft.Network/loadBalancers/inboundNatPools/join/action
[...]
```

# Inspecting Not-Actions of Built-in Roles

```
# PowerShell
```

```
(Get-AzureRmRoleDefinition "Contributor").NotActions
```

```
# Result
```

```
Microsoft.Authorization/*/Delete
```

```
Microsoft.Authorization/*/Write
```

```
Microsoft.Authorization/elevateAccess/Action
```

# Custom Roles

Consists also of Actions and NotActions

Have to be applied to Scopes (Subscription, RG, R)

Creation via JSON or scripting

## Advice

Use when built-in roles do not fit well

If possible prefer built-in rules and avoid over-engineering

# Custom Role - Virtual Machine Operator

## Actions

Microsoft.Compute/\*/read  
Microsoft.Compute/virtualMachines/start/action  
Microsoft.Compute/virtualMachines/restart/action  
...

## Not-Actions

## Scope

/subscriptions/39...-b...-5...-21...fc

Role:  
Virtual Machine Operator



# References

# Design and deploy ARM templates (10-15%)

## Implement ARM templates

Authoring Azure Resource Manager templates	<a href="https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-group-authoring-templates">https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-group-authoring-templates</a>
Deploy resources with Resource Manager templates and Azure portal	<a href="https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-group-template-deploy-portal">https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-group-template-deploy-portal</a>
Deploy resources with Resource Manager templates and Azure PowerShell	<a href="https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-group-template-deploy">https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-group-template-deploy</a>
Deploy resources with Resource Manager templates and Azure CLI	<a href="https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-group-template-deploy-cli">https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-group-template-deploy-cli</a>
Deploy resources with Resource Manager templates and Resource Manager REST API	<a href="https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-group-template-deploy-rest">https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-group-template-deploy-rest</a>
Create your first Azure Resource Manager template	<a href="https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-manager-create-first-template">https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-manager-create-first-template</a>

# Design and deploy ARM templates (10-15%)

## Design role-based access control (RBAC)

Get started with Role-Based Access Control in the Azure portal

<https://docs.microsoft.com/en-us/azure/active-directory/role-based-access-control-what-is>

Built-in roles for Azure role-based access control

<https://docs.microsoft.com/en-us/azure/active-directory/role-based-access-built-in-roles>

Use Role-Based Access Control to manage access to your Azure subscription resources

<https://docs.microsoft.com/en-us/azure/active-directory/role-based-access-control-configure>

Create custom roles for Azure Role-Based Access Control

<https://docs.microsoft.com/en-us/azure/active-directory/role-based-access-control-custom-roles>

Role Based Access Control and Azure Subscription GUID

[https://blogs.msdn.microsoft.com/uk\\_faculty\\_connection/2017/05/30/role-based-access-control-and-azure-subscription-guid/](https://blogs.msdn.microsoft.com/uk_faculty_connection/2017/05/30/role-based-access-control-and-azure-subscription-guid/)

Intro on role-based access control

<https://docs.microsoft.com/en-us/azure/active-directory/role-based-access-control-create-custom-roles-for-internal-external-users>

# Design and deploy ARM templates (10-15%)

## Control Access

Use portal to create an Azure Active Directory application and service principal that can access resources <https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-group-create-service-principal-portal>

Use Resource Manager authentication API to access subscriptions <https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-manager-api-authentication>

Lock resources to prevent unexpected changes <https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-group-lock-resources>



Infrastructure as  
Code

Control Access

# Create Service Principles - Options

- Portal ([Link](#))
- CLI 1.0 ([Link](#))
- CLI 2.0 ([Link](#))
- PowerShell ([Link](#))

# Required Permissions in Azure AD

Microsoft Azure lohmann - User settings

Search resources

lohmann - User settings

Azure Active Directory

☰

+ Overview

Quick start

MANAGE

Users and groups

Enterprise applications

Devices (Preview)

App registrations

Application proxy

Licenses

Azure AD Connect

Domain names

Mobility (MDM and MAM)

Password reset

Company branding

User settings

Properties

Save Discard

Enterprise applications

Users can consent to apps accessing company data on their behalf  Yes  No

Users can add gallery apps to their Access Panel  Yes  No

App registrations

Users can register applications  Yes  No

External users

Guest users permissions are limited  Yes  No

Admins and users in the guest inviter role can invite  Yes  No

Members can invite  Yes  No

Guests can invite  Yes  No

Administration portal

Restrict access to Azure AD administration portal  Yes  No

The screenshot shows the 'User settings' page for a user named 'lohmann'. On the left, a sidebar lists various management options like 'Users and groups', 'Enterprise applications', and 'App registrations'. The 'User settings' option is highlighted with a dashed blue border and has a red arrow pointing to it from the bottom-left. The main content area is titled 'Enterprise applications' and contains several configuration items. One item, 'Users can register applications', is highlighted with a red box. Below this, there's a section for 'External users' and another for the 'Administration portal', each with its own set of configuration options. At the top right, there are 'Save' and 'Discard' buttons.

"App registrations" is required for Service Principle creation.

# Creating a Service Principle with CLI 2.0

```
# List all existing App registrations
```

```
az ad app list
```

```
# List all existing App registrations
```

```
az ad sp create-for-rbac -n "MySP"
```

```
# Result
```

```
{  
    "appId": "5916c94f-...-...-...-...",  
    "displayName": "MySP",  
    "name": "http://MySP",  
    "password": "71643222-...-...-...-...",  
    "tenant": "a2583895-...-...-...-..."  
}
```

# Usage Sample: Custom Code with .NET

```
var tenantId = "a2583895-...";  
var subscriptionId = "397015c1-...";  
  
var sp = new ServicePrincipalLoginInformation()  
{  
    ClientId = "5916c94f-...",  
    ClientSecret = "71643222-..."  
};  
  
var azureCredentials = new AzureCredentials(sp, tenantId, AzureEnvironment.AzureGlobalCloud);  
var azure = Azure.Authenticate(azureCredentials).WithSubscription(subscriptionId);  
  
var storageAccountName = "dotnetdemo2328svhs3";  
azure.StorageAccounts.Define(storageAccountName)  
    .WithRegion("WestEurope")  
    .WithNewResourceGroup("dotnet-demo-rg")  
    .WithSku(StorageModels.SkuName.StandardLRS)  
    .Create();
```

# Usage Sample: Custom Code with .NET (cont.)

dotnet-demo-rg - Activity log X

Resource group

Search (Ctrl+ /)

Columns Export Log search

Select query ... ✖️ ✖️ ✖️

\* Subscription ⓘ Resource group ⓘ  
Lohmann (MSDN) dotnet-demo-rg

Timespan ⓘ Event category ⓘ  
Last 6 hours All categories

Resource ⓘ Resource type ⓘ Operation ⓘ  
All resources All resource types 0 selected

\* Event severity ⓘ Event initiated by ⓘ Search ⓘ  
4 selected Email or name or ser... Search

Insights (Last 24 hours): 0 failed deployments | 7 role assignments | 6 errors | 0 alerts fired | 0 outage notifications

Apply Reset

Query returned 2 items. Click here to download all the items as csv.

OPERATION NAME	STATUS	TIME	TIME STAMP	SUBSCRIPTION	EVENT INITIATED BY
Write StorageAccounts	Succeeded	1 min ago	Tue Sep 05 2...	Lohmann (MSDN)	MySP
Update resource group	Succeeded	1 min ago	Tue Sep 05 2...	Lohmann (MSDN)	MySP

A red arrow points from the text "Event initiated by" in the search filters to the "EVENT INITIATED BY" column header in the table results. A red box highlights the entire "EVENT INITIATED BY" column.

Demo:  
Service Principles

# Resource Manager API Authentication



Scenario: Offer custom Azure Services

# Resource Locks

Protect resources against accidental deletion or changes

The image shows two screenshots of the Azure portal. The left screenshot is a 'Settings' page for a resource group named 'contososerverexample'. It includes sections for 'SUPPORT + TROUBLESHOOTING' (Diagnose and solve problems, Activity logs, New support request) and 'SETTINGS' (Tags, Locks, Users, Automation script). The 'Locks' option is highlighted with a red box. The right screenshot is a 'Management locks' dialog box for the same resource group. It has tabs for '+ Add', 'Resource group', 'Subscription', and 'Refresh'. It shows an 'Add lock' section with a lock name 'DatabaseServerLock' (status green checkmark), a lock type dropdown set to 'Delete', and a notes field containing 'Prevent deleting the database server'. There are 'OK' and 'Cancel' buttons at the bottom.

Demo:  
Resource Locks



Automation

Desired State Configuration

# Desired State Configuration (DSC)

- Azure Automation Desired State Configuration (DSC) allows you to consistently deploy, monitor, and automatically update the desired state of all your IT resources, at scale from the cloud.
- Built on PowerShell DSC and can align machine configuration with a specific state across physical and virtual machines, using Windows or Linux, in the cloud or on-premises
- Builds on top of PowerShell DSC to provide an easier configuration management experience.
- Automation DSC includes:
  - Author and manage PowerShell DSC configurations
  - Import DSC Resources
  - Generate DSC Node configurations (MOF documents)
- DSC configuration files are stored on an Azure Automation DSC Pull server so that target nodes can download and apply them.

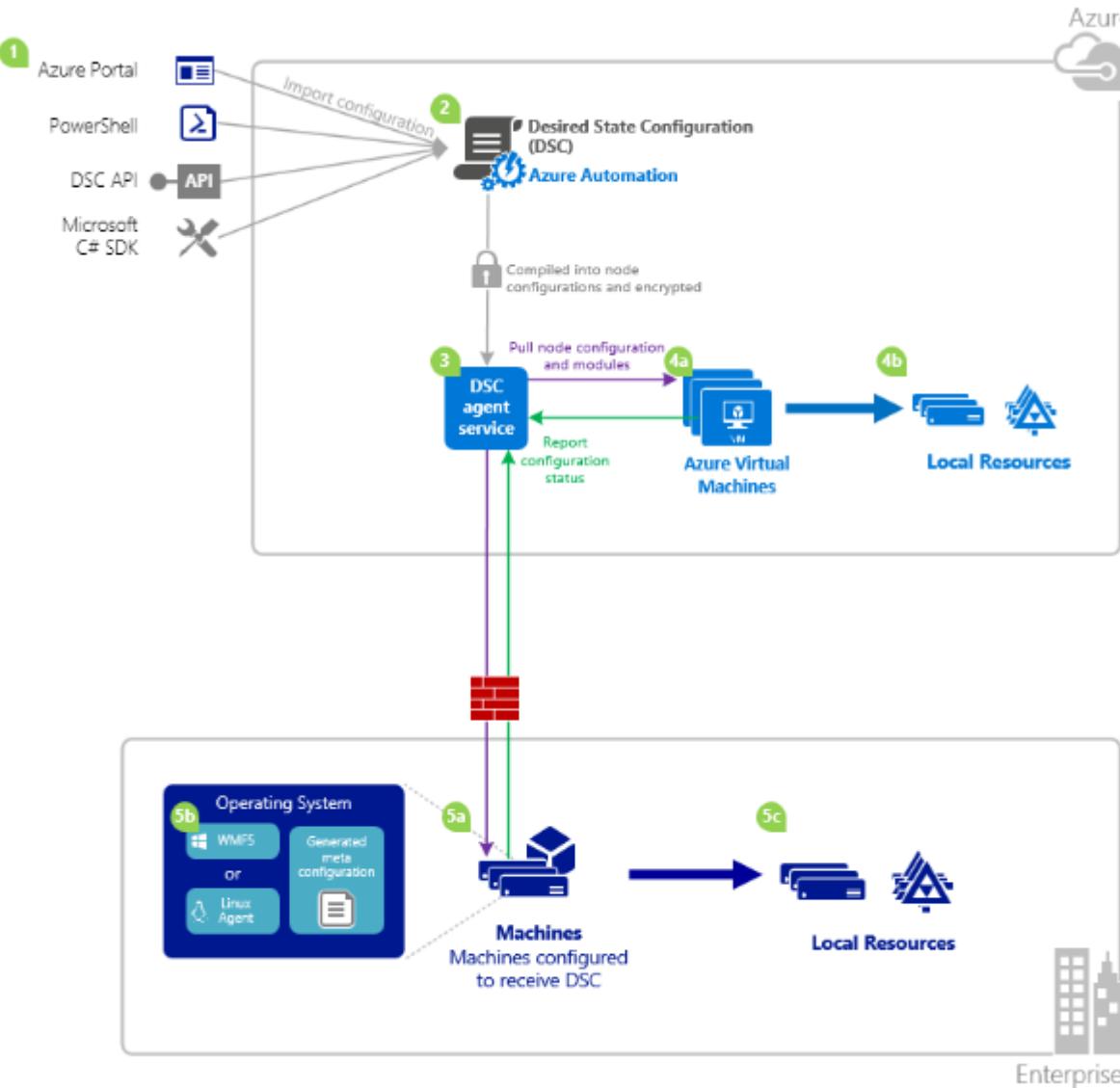


# Automation DSC Terms

- Configuration – Introduced in PowerShell DSC and allows you to define the desired state of your environment using PowerShell syntax.
- Node Configuration – Is a file that is produced when a DSC configuration is compiled, this is typically the configuration document that nodes will apply.
- Node – Any machine that has its configuration managed by DSC.
- Resource – A PowerShell module that is used to define a DSC configuration. They are seen as the building blocks of DSC configuration.
- Compilation Job – An instance of compilation of a configuration to create a node configuration.
  - Similar to Azure Automation Runbook jobs, but they do not perform any task, only compile configurations
  - Automatically stored on an Azure DSC pull server
  - Overwrites previous versions of node configurations



# Automation DSC Process



- An actor imports a DSC configuration to Azure Automation (1).
- An actor compiles the DSC configuration into node configuration and it's encrypted by Azure Automation (2).
- The node configuration is placed in the DSC agent service and is then pulled by cloud or on-premises machines (3).
- Cloud – 4a:** Node configurations or modules are pulled by the DSC on-boarded Azure VM's and the VM's report on their configuration status and compliance back to the service (4a).
- On-premises – 4b:** Azure VMs conform to the desired state (4b).
- On-premises – 5a:** Node configuration is pulled on to local machine configured to receive DSC (5a).
- On-premises – 5b:** On-premises machines must have either Windows Management Framework 5 installed if it's a Windows node or the Linux Agent installed for a Linux node. Those machines act on local resources as instructed (5b).
- Local Resources – 5c:** Local machines conform to the desired state (5c).

# Onboarding Nodes for Management

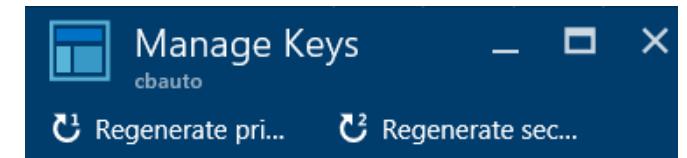
- Azure Automation DSC allows onboarding of the following machines:
  - Azure Classic VM's
  - Azure Resource Manager VM's
  - Amazon Web Services VM's
  - Physical or virtual Windows machines on-premises, or in a cloud other than Azure or AWS
  - Physical or virtual Linux machines on-premises, in Azure, or in a cloud other than Azure
- Azure classic VM's can be onboarded via the new Azure portal or PowerShell.
- Azure Resource Manager VM's can be onboarded via the new Azure portal, ARM templates or PowerShell.
- Amazon Web Services VM's can be onboarded using the AWS DSC Toolkit.
- Windows machines on-premises or in a cloud other than Azure or AWS must have WMF 5.0 installed and have the PowerShell DSC metaconfiguration applied.
- Linux machines on-premises, in Azure or in another cloud must have the latest DSC Linux agent installed and have the PowerShell DSC metaconfiguration applied.

# DSC Node Registration Parameters

- **Registration key** – Specifies which Automation account Access Key to use for the DSC node to authenticate with.
- **Node Configuration Name** – Specifies the name of the DSC configuration file to be used.
- **Refresh Frequency** – Specifies how often the DSC node will contact the Pull server and download the latest node configuration.
- **Configuration Mode Frequency** – Specifies how often the downloaded node configuration will be applied.
- **Configuration Mode** – Specifies the mode of configuration e.g. ApplyAndMonitor, ApplyOnly.
- **Allow Module Override** – Specifies whether or not newer modules downloaded from the Pull server are allowed to overwrite older modules on the DSC node.
- **Reboot Node if Needed** – Specifies whether or not to reboot following a configuration update.
- **Action after Reboot** – Specifies actions to take following a reboot e.g. ContinueConfiguration or StopConfiguration.

# DSC Metaconfiguration & Secure Registration

- DSC Metaconfiguration is a script that consists of the DSC engine settings that will be used to connect a node to a DSC pull server and keep it updated.
- DSC metaconfigurations for Azure Automation DSC can be generated using either a PowerShell DSC configuration, or the Azure Automation PowerShell cmdlets.
- Must be applied to on-premises or non Azure virtual machines in order to onboard the server to Automation DSC.
- Secure Registration is a registration protocol, allows a DSC node to authenticate to a PowerShell DSC V2 Pull server (including Azure Automation DSC).
  - The node registers to the DSC Pull server at a Registration URL, and authenticates using a Registration key specified in the DSC Metaconfiguration.
  - A certificate is generated and used for future communication between the node and the DSC Pull server.



PRIMARY ACCESS KEY

alv4DumAi5gGMotrVEijrtvQiEt5CWaf24W

SECONDARY ACCESS KEY

foYTLwxXKwOsvzoacBVHctnAY6xElt8UaX

URL

<https://we-agentservice-prod-1.azure-aut>

# Compiling Configuration

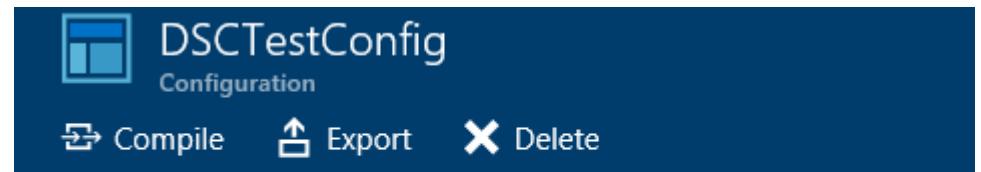
- DSC configuration can be compiled in two ways with Azure Automation, the Azure portal or with Windows PowerShell.

## Azure portal

- Simplest method with interactive user interface
- Form to provide simple parameter values
- Easily track job state
- Access authenticated with Azure logon

## Windows PowerShell

- Call from command line with PowerShell cmdlets
- Can be included in automated solution with multiple steps
- Provide simple and complex parameter values
- Track job state
- Client required to support PowerShell cmdlets
- Pass ConfigurationData
- Compile configurations that use credentials



Essentials ^