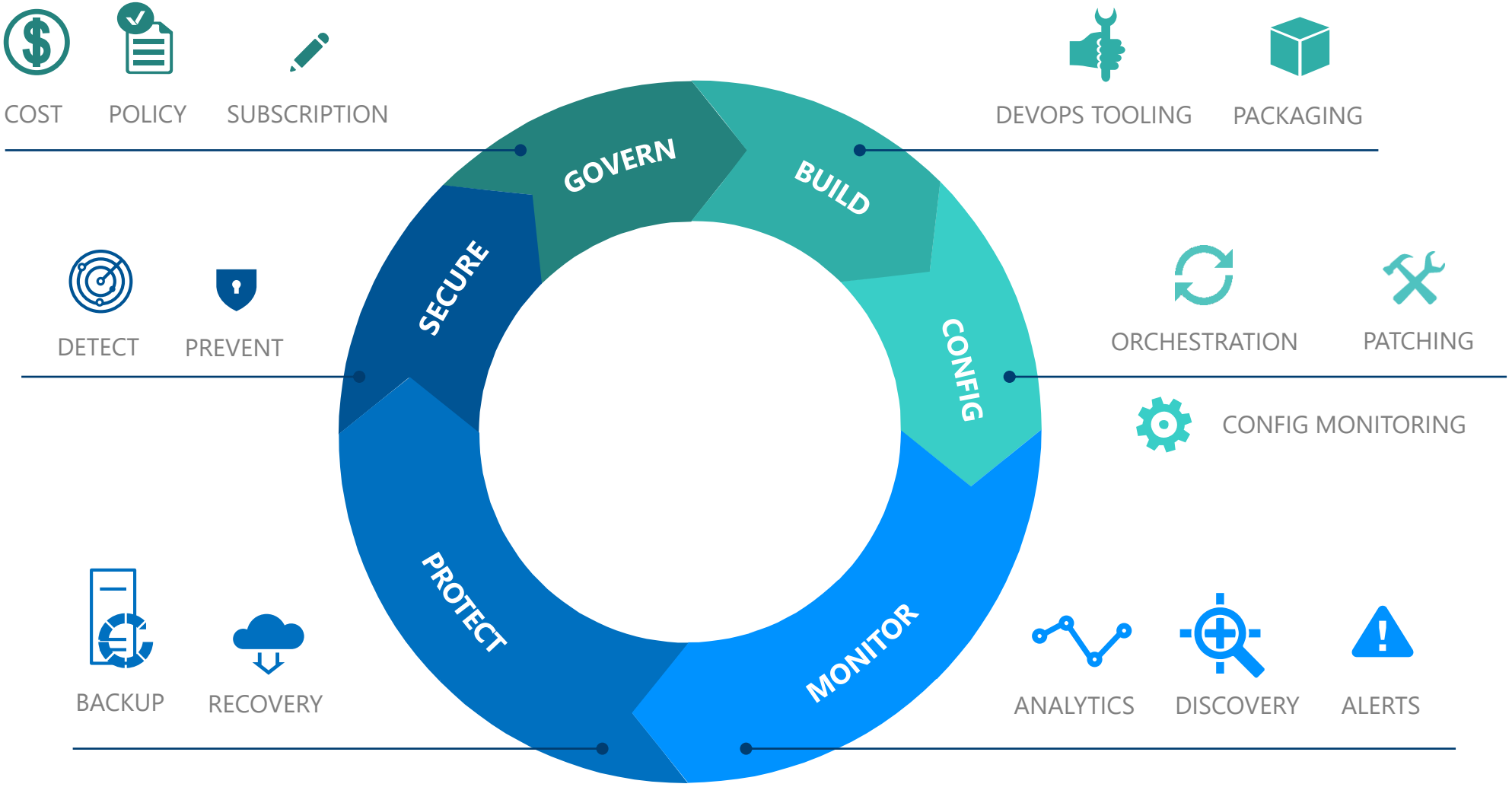
Infrastructure as Code and Modern ITSM

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

Traditional

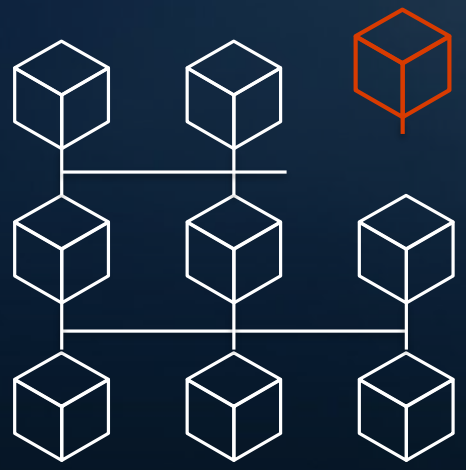
- Mostly manual
- Alert/ticketing driven
- No ideas what state the environment is in
- Constantly „saving the day“

Modern

Competitive Advantage

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

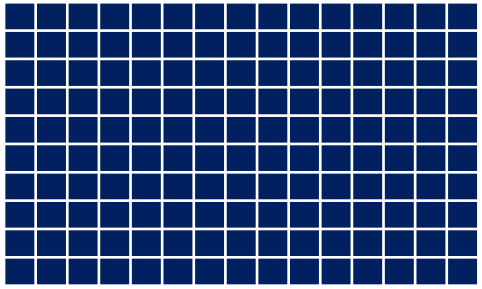
What makes it a competitive advantage?

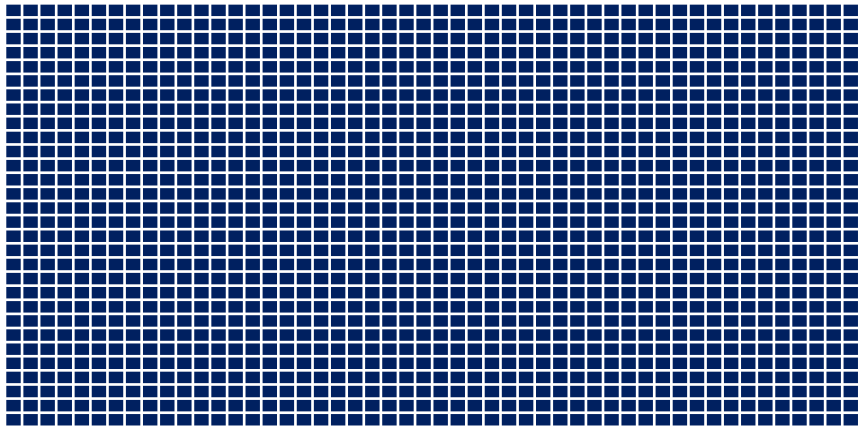


Resolving infrastructure and application issues can be time-consuming and can impact SLA.

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

Making changes manually



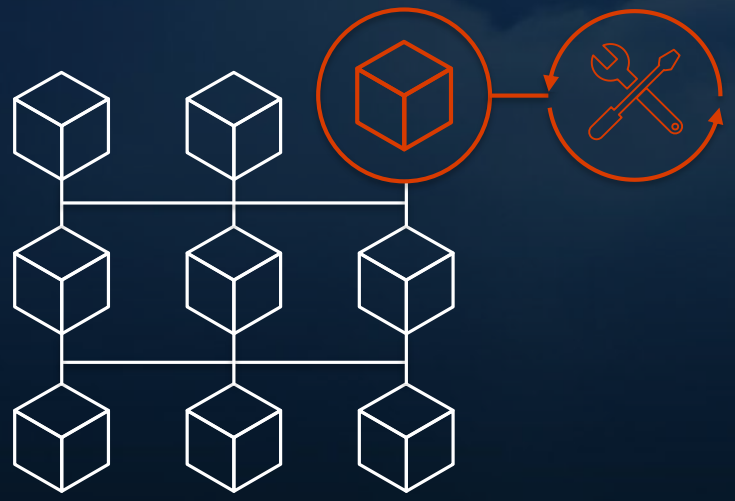


15 Servers: Sort of works

150 Servers: Unconscious deception

1500 Servers: Self destructive

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)

Infrastructure as Code

Overview Azure Resource Manager

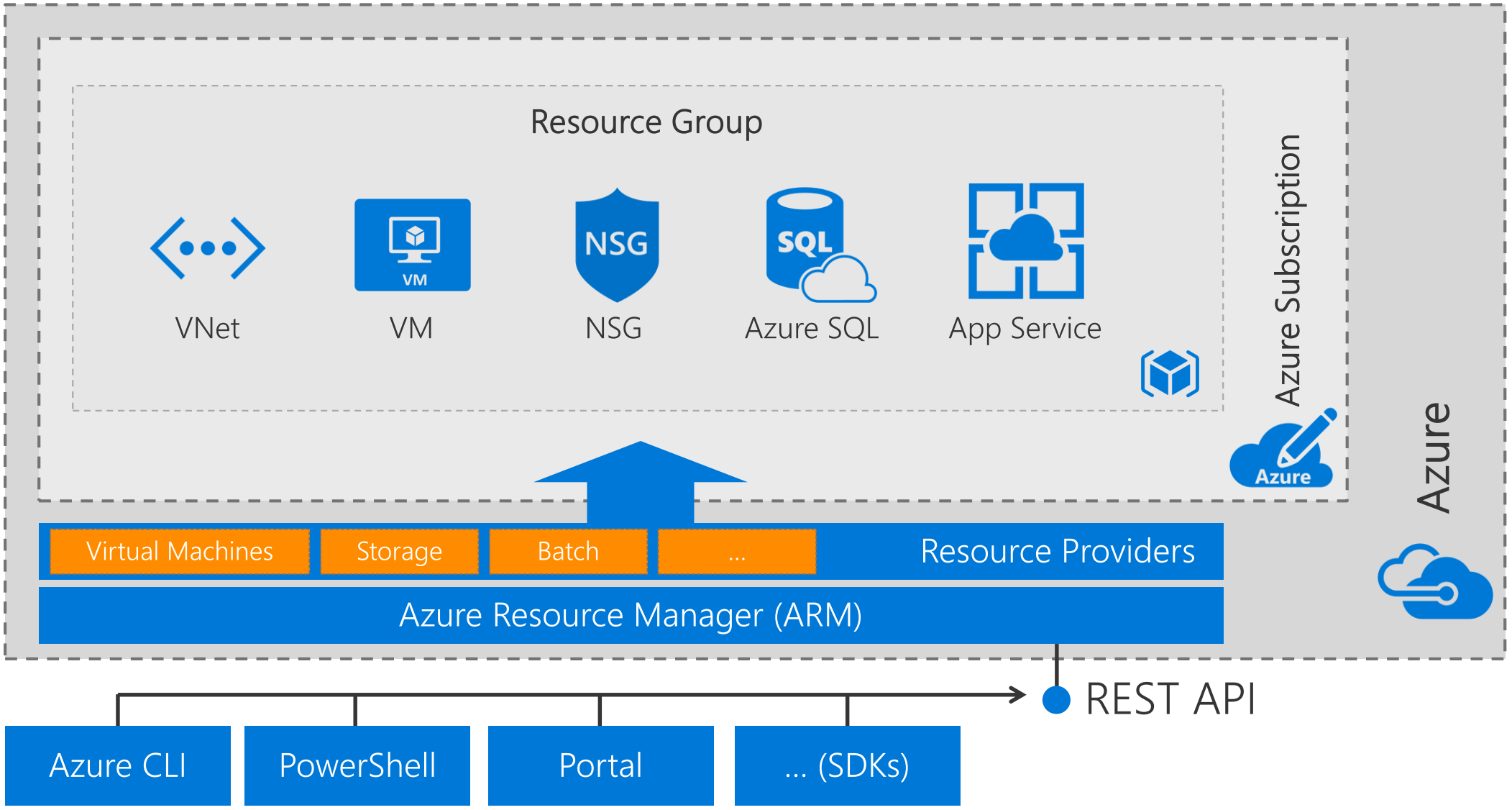
Azure Resource Manager (ARM)

Successor of Azure Service Manager (ASM)
Went GA in 2015
Uses modular approach towards resources

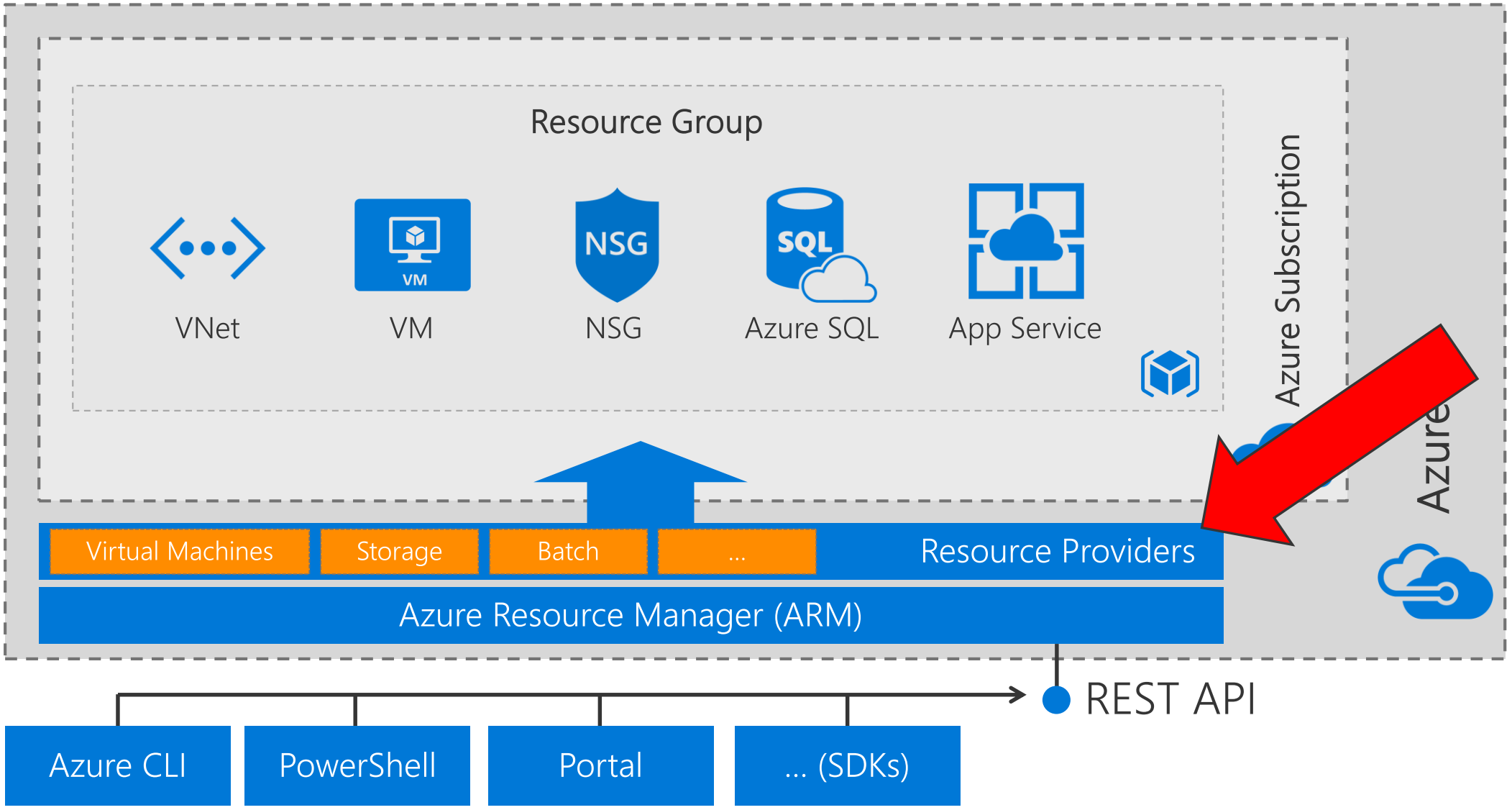
Benefits:

- 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

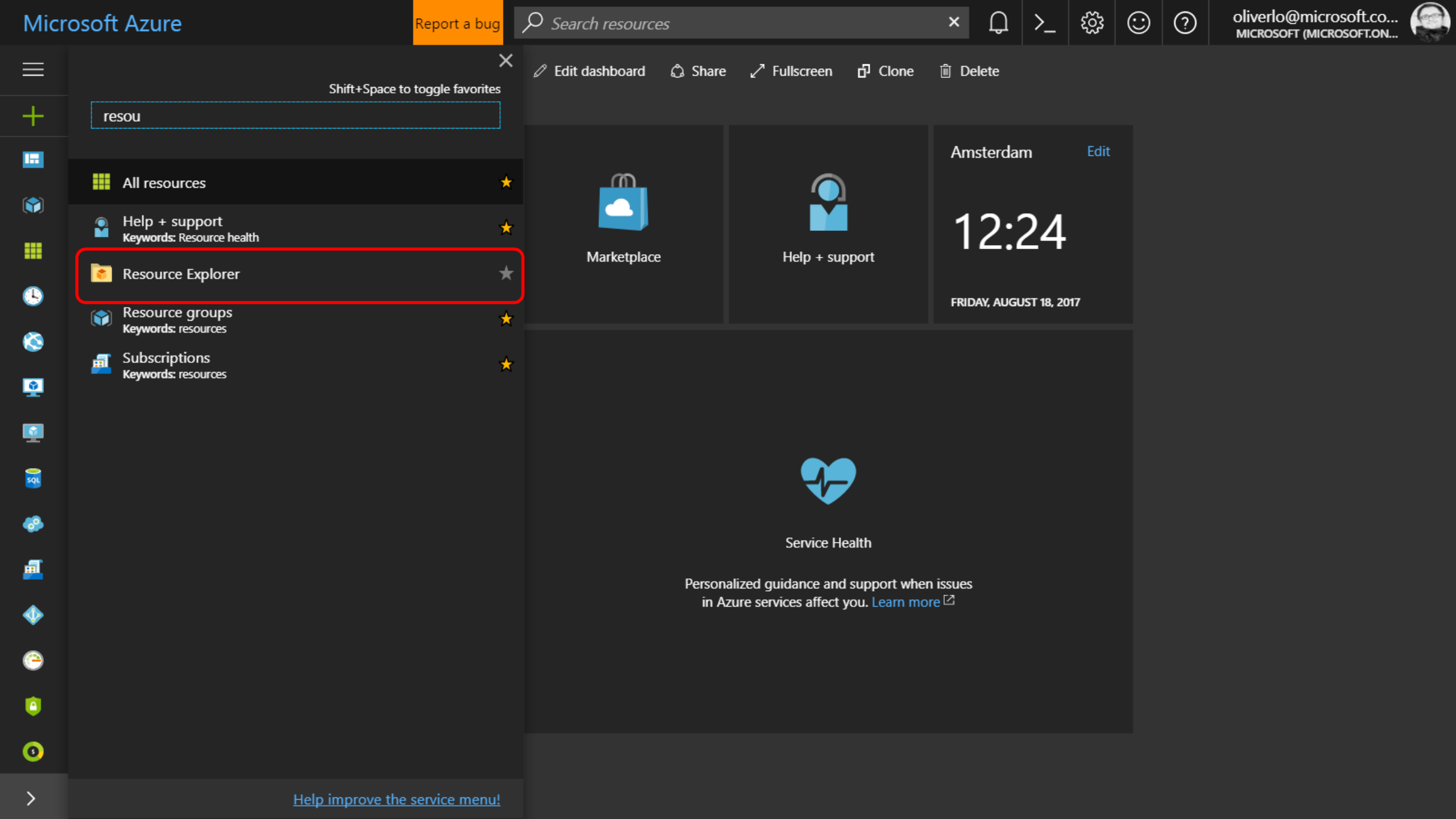
Namespace: Microsoft.Compute

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, …


Microsoft Azure
Report a bug
Search resources
oliverlo@microsoft.co...
MICROSOFT (MICROSOFT.ON...
Edit dashboard
Share
Fullscreen
Clone
Delete
Shift+Space to toggle favorites
resou
All resources
Help + support
Keywords: Resource health
Resource Explorer
Resource groups
Keywords: resources
Subscriptions
Keywords: resources
Help improve the service menu!
Marketplace
Help + support
Amsterdam
Edit
12:24
FRIDAY, AUGUST 18, 2017
Service Health
Personalized guidance and support when issues in Azure services affect you. Learn more

Resource Explorer

microsoft

Search...

- Providers
  - 84codes.CloudAMQP
  - AppDynamics.APM
  - Aspera.Transfers
  - Auth0.Cloud
  - Citrix.Cloud
  - Cloudyn.Analytics
  - Conexlink.MyCloudIT
  - Crypteron.DataSecurity
  - Dynatrace.DynatraceSaaS
  - Dynatrace.Ruxit
  - LiveArena.Broadcast
  - Lombiq.DotNest
  - Mailjet.Email
  - Microsoft.AAD
  - microsoft.aadiam
  - Microsoft.ADHybridHealthService
  - Microsoft.Advisor
  - Microsoft.AnalysisServices
  - Microsoft.ApiManagement
  - Microsoft.AppService
  - Microsoft.Archive
  - Microsoft.Authorization

Providers (Response Time 680ms)

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


```
{
    "value": [
        {
            "namespace": "84codes.CloudAMQP",
            "resourceTypes": [
                {
                    "resourceType": "servers",
                    "locations": [
                        "East US 2",
                        "Central US",
                        "East US",
                        "North Central US",
                        "South Central US",
                        "West US",
                        "North Europe",
                        "West Europe",
                        "East Asia",
                        "Southeast Asia",
                        "Japan East",
                        "Japan West",
                        "Australia East",
                        "Australia Southeast",
                        "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
```

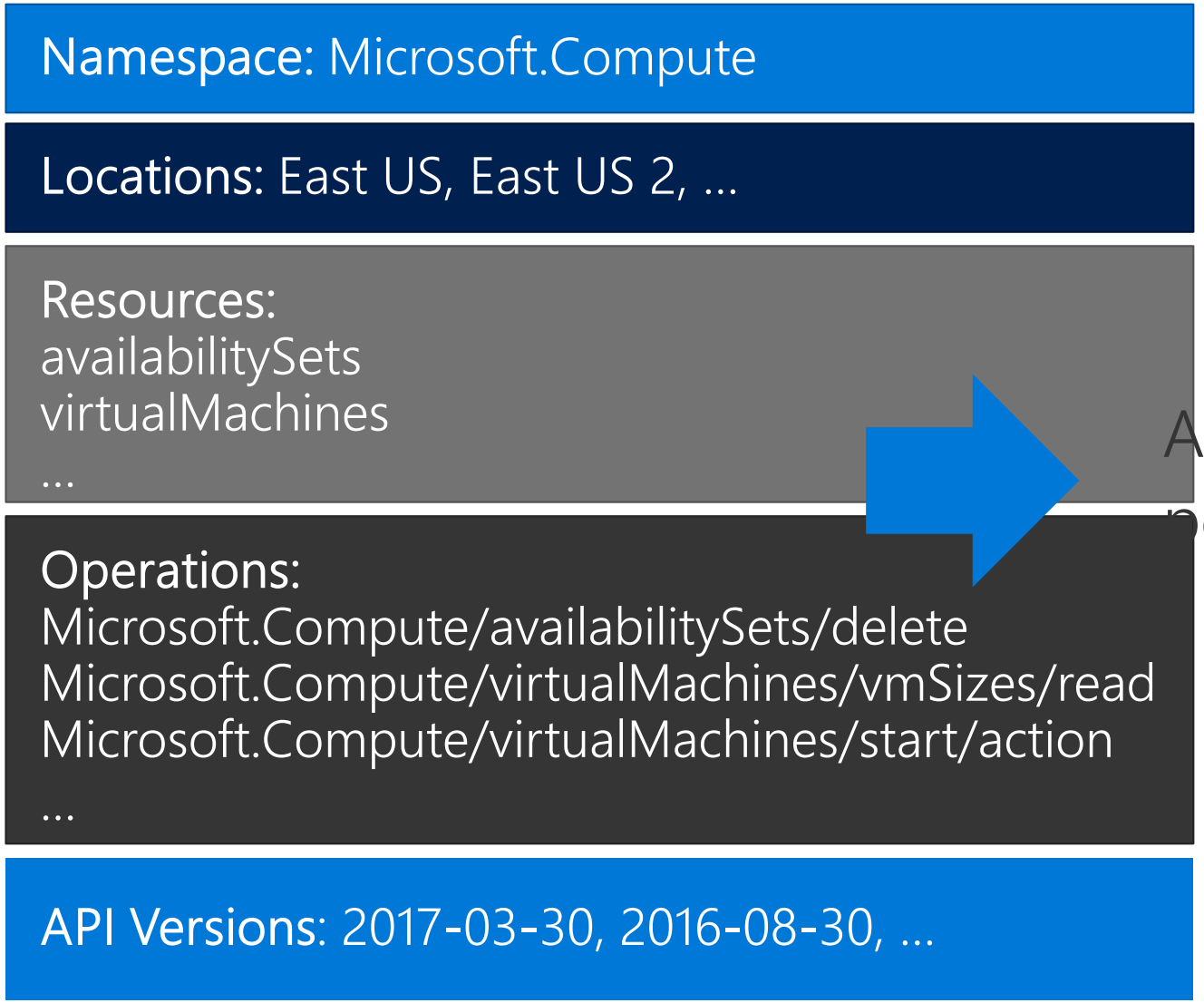
Operations on Resource Types

```
# 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

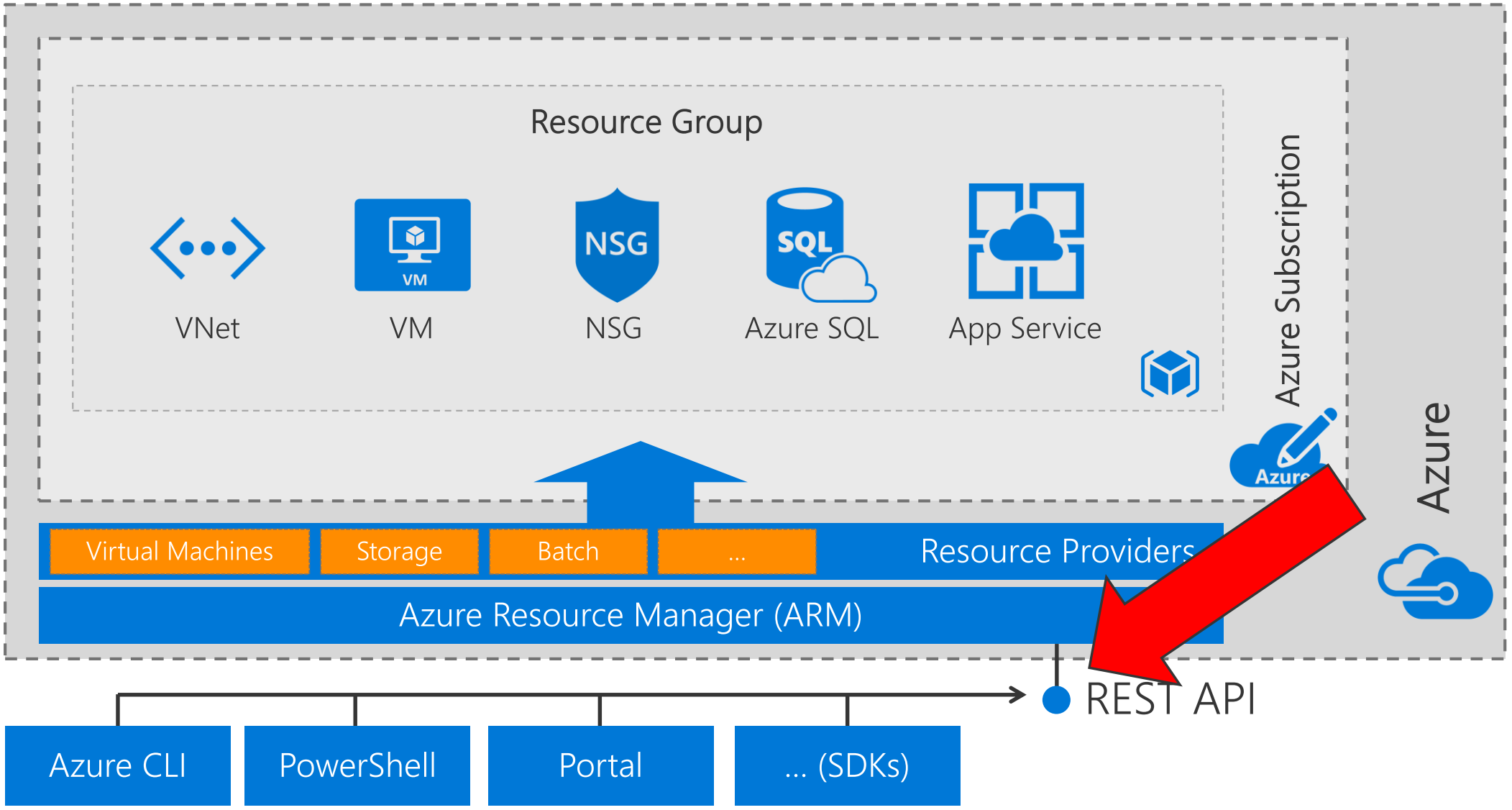
```
# Azure CLI 2.0
az provider register -n "Microsoft.ApiManagement"

# Azure PowerShell
Register-AzureRmResourceProvider -ProviderNamespace "Microsoft.
ApiManagement"
```

Resource Provider - Simplified Structure



An extremely powerful concept!

Azure Resource Manager REST API



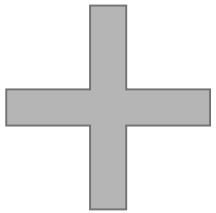
Elements of an Azure REST API Call

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
```

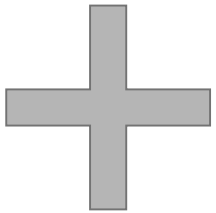


```
{
  "location":"northeurope",
  "tags": {
    "tagname1": "test-tag"
  }
}
```

https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-manager-rest-api

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
```

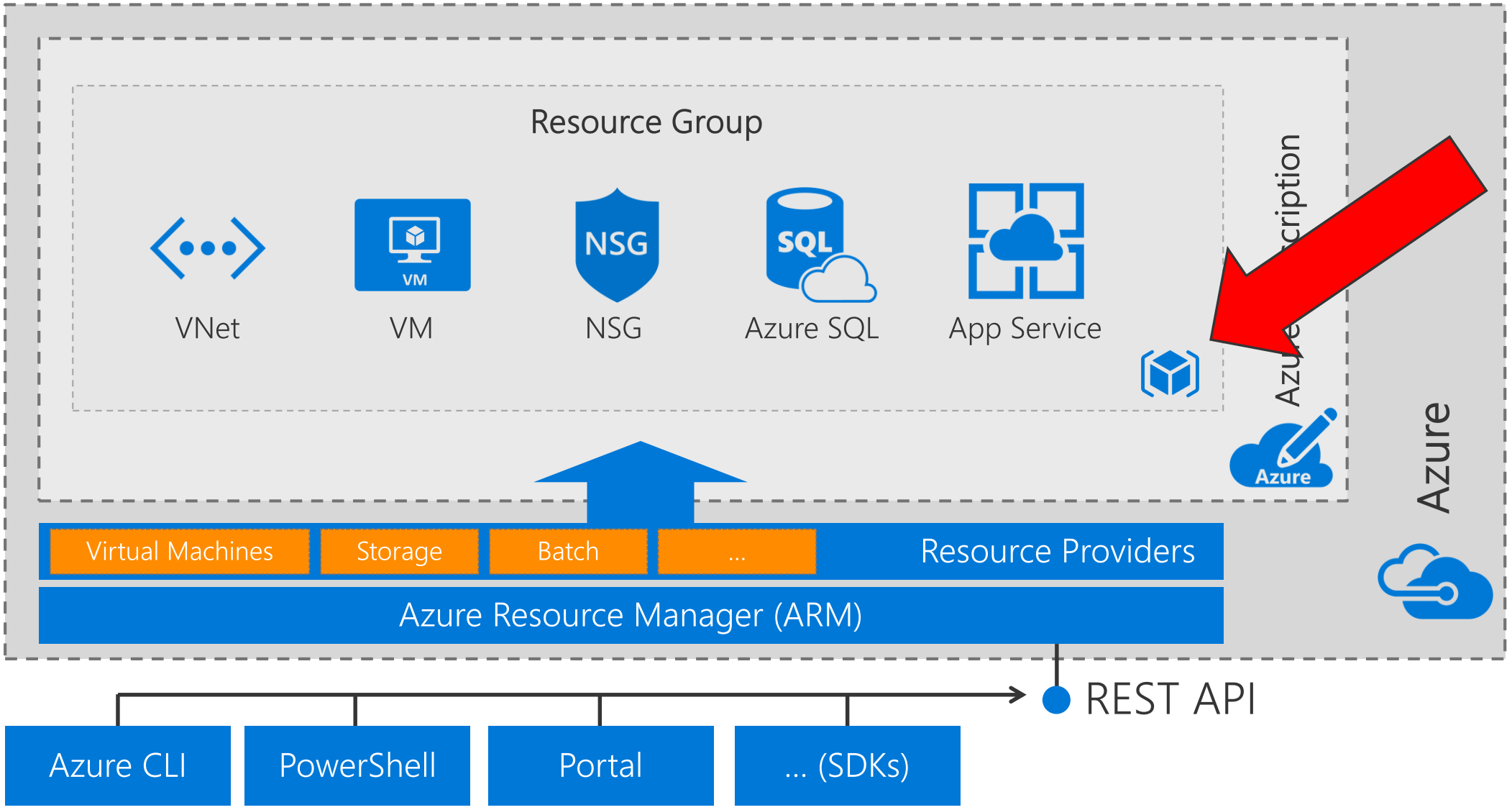


```
{
  "location": "WestEurope",
  "tags": {},
  "sku": {
    "name": "Standard_LRS"
  }
}
```

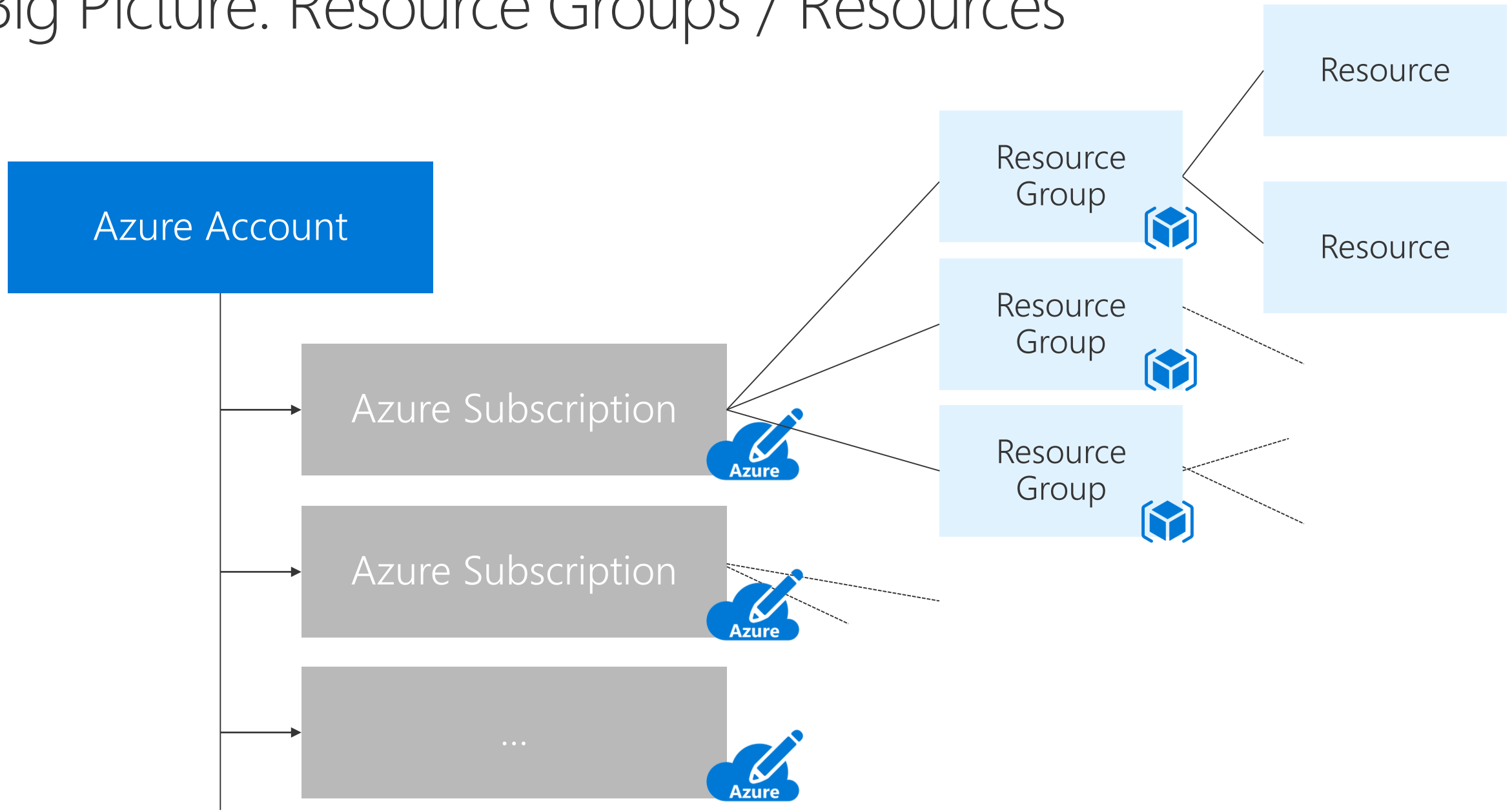
https://docs.microsoft.com/en-us/rest/api/storagerp/StorageAccounts/Create

Infrastructure as Code

Resource Providers – Resource Groups

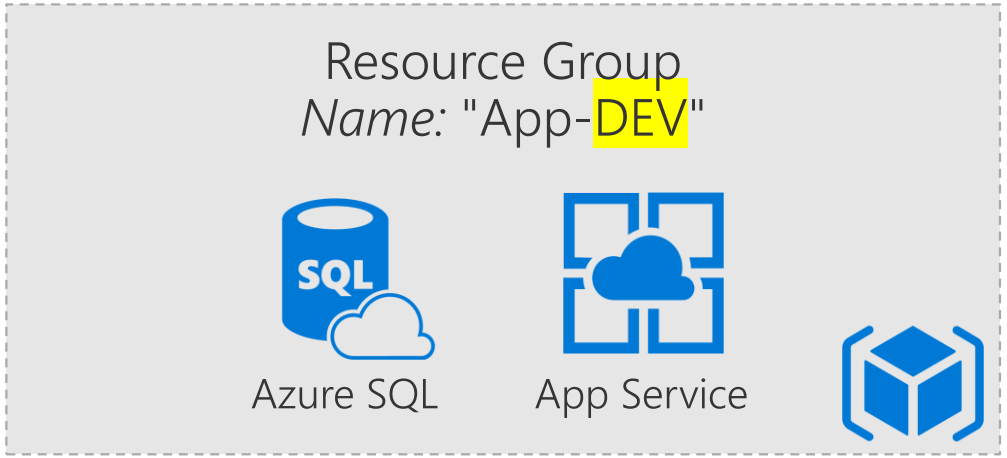


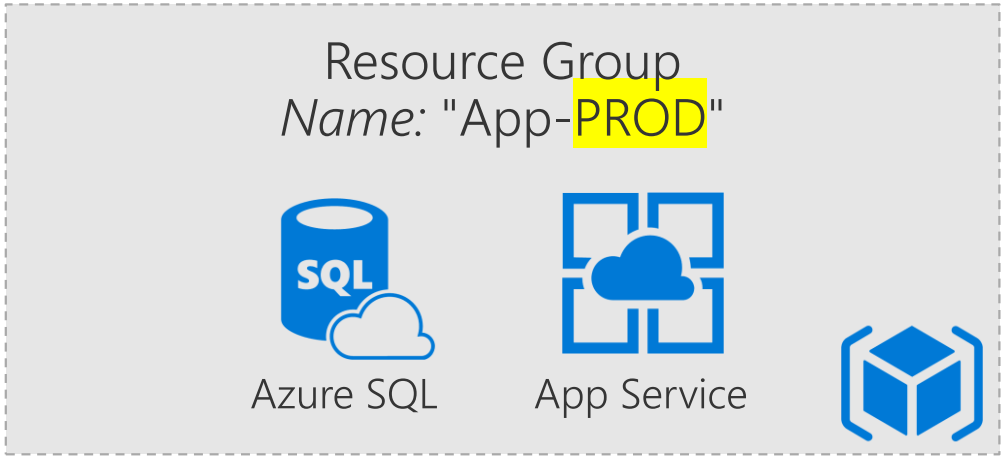
Big Picture: Resource Groups / Resources



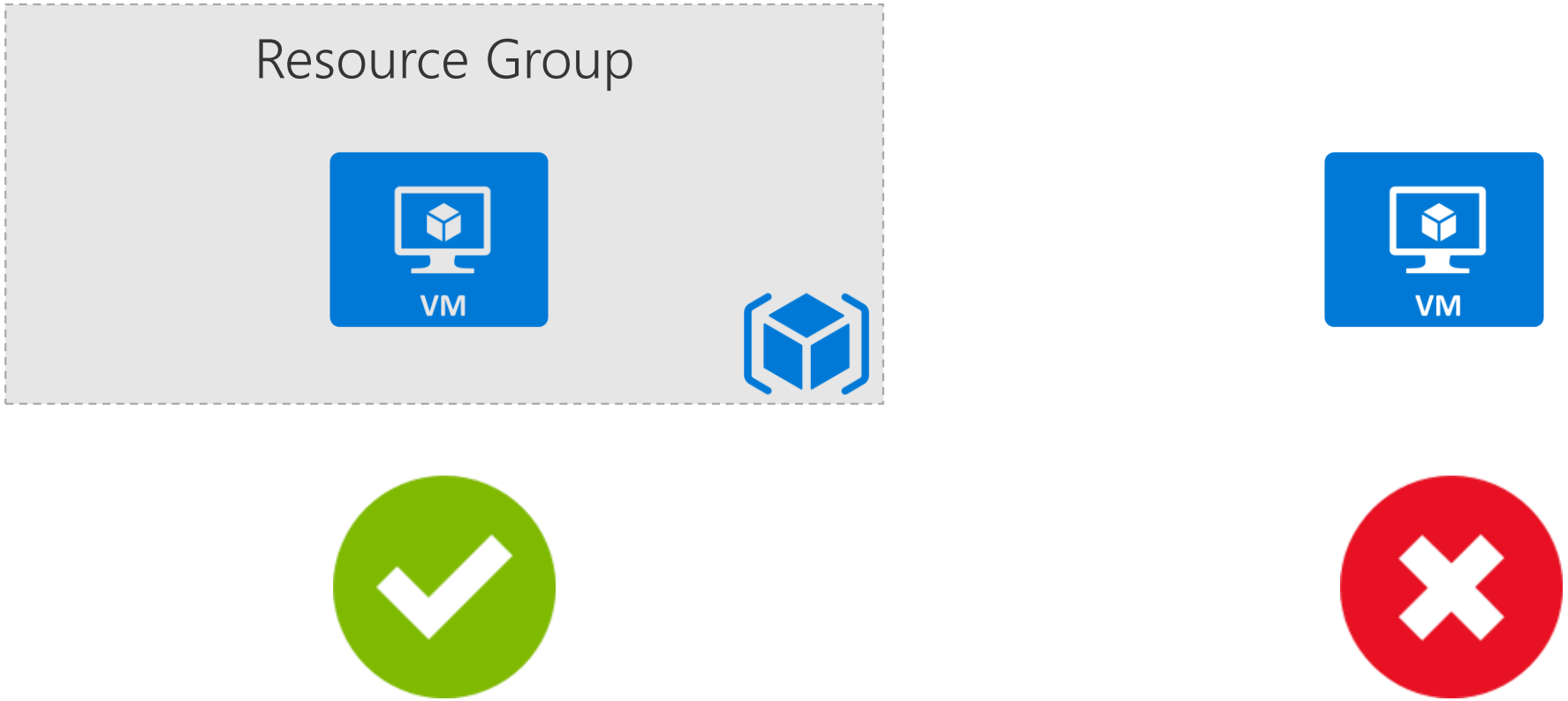
Resource Group are Management Units

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

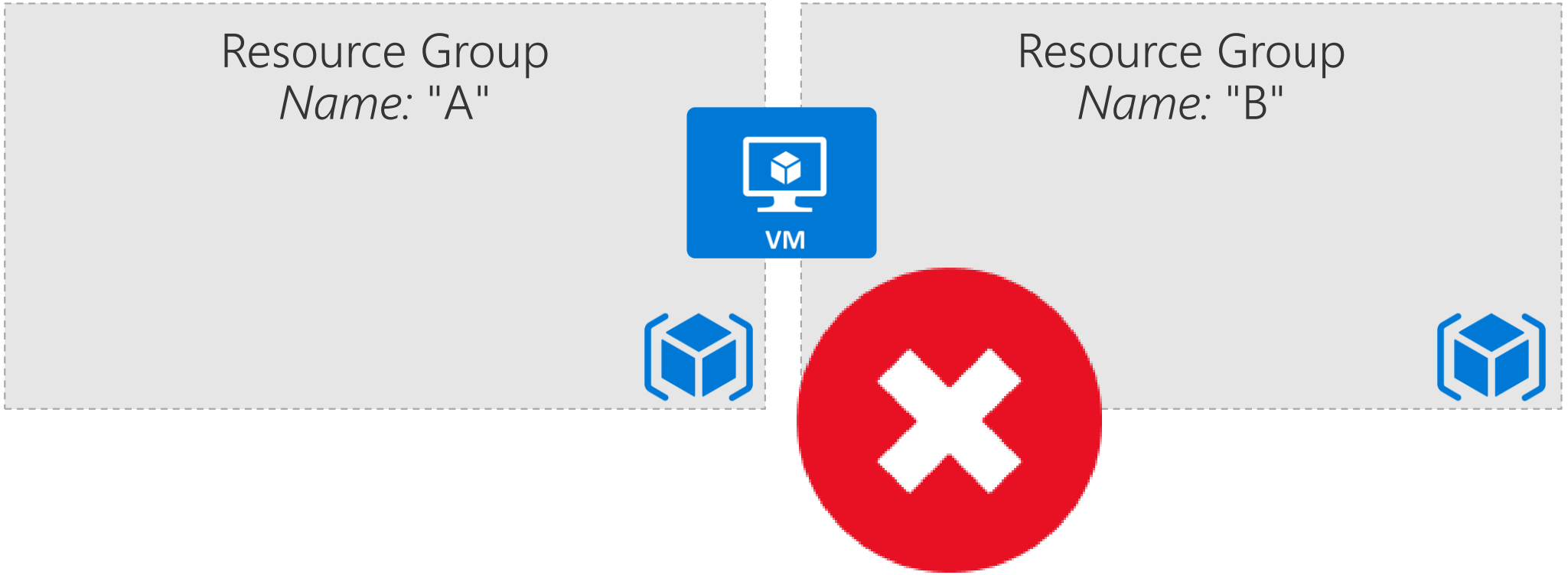




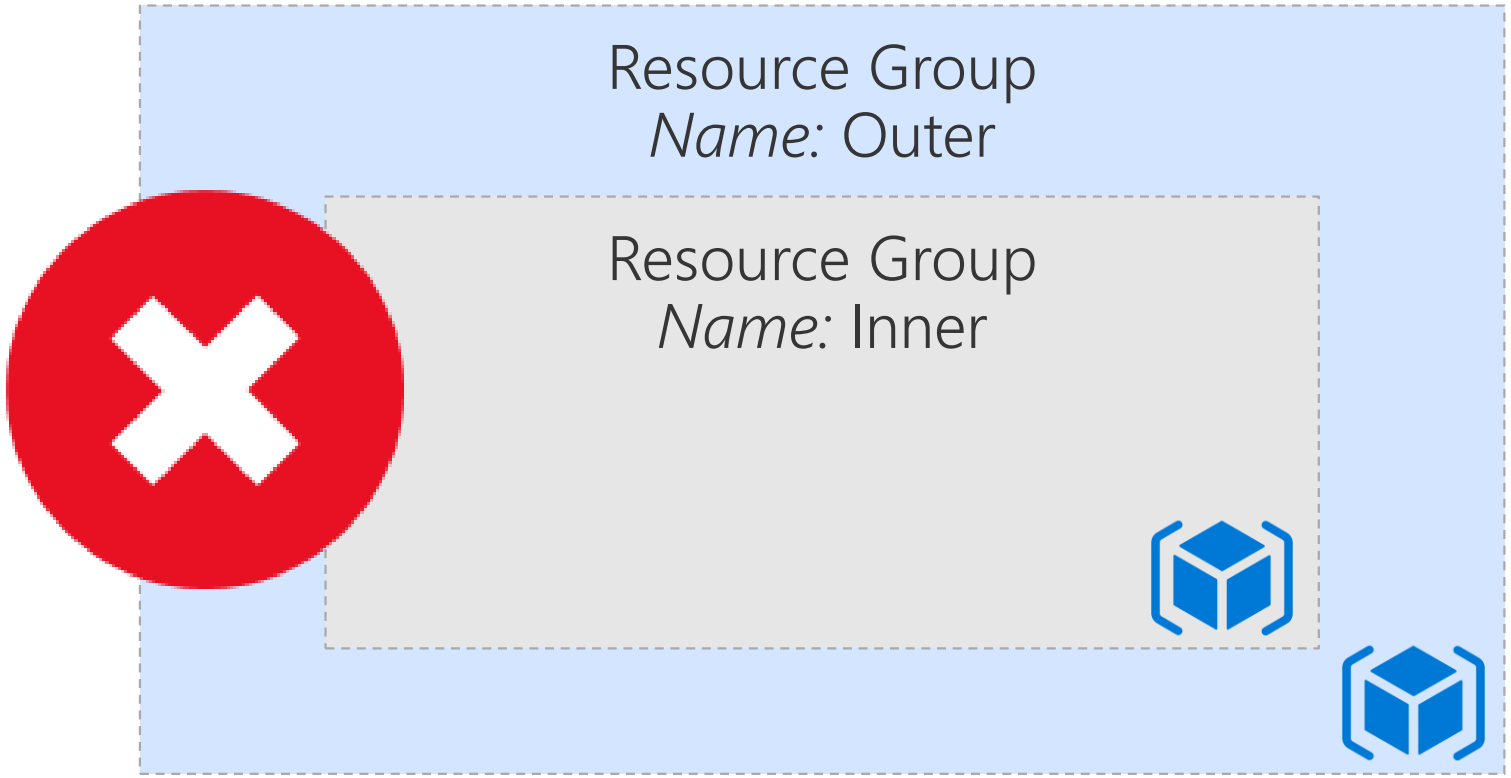
A Resource always belongs to a Resource Group



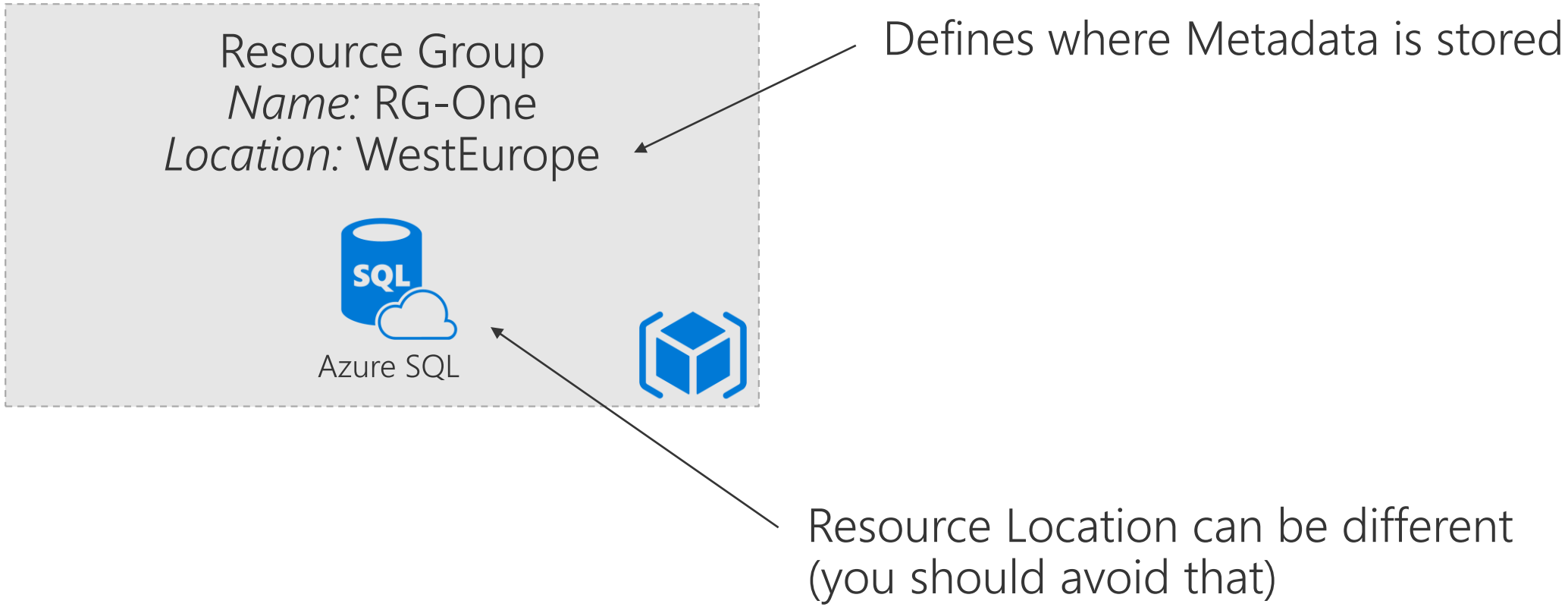
A Resource always belongs to a single Resource Group



Resource Groups cannot be nested



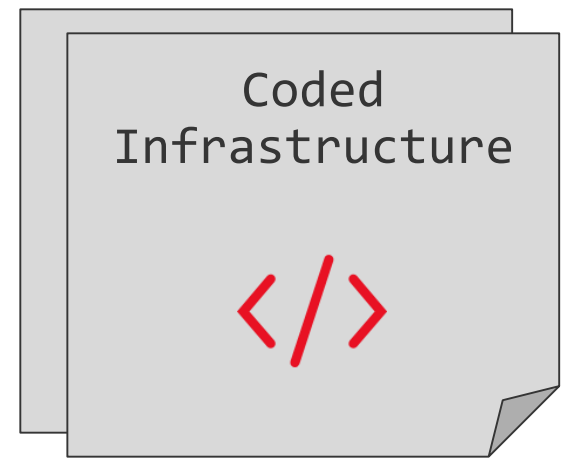
Resource Groups have a Location Property

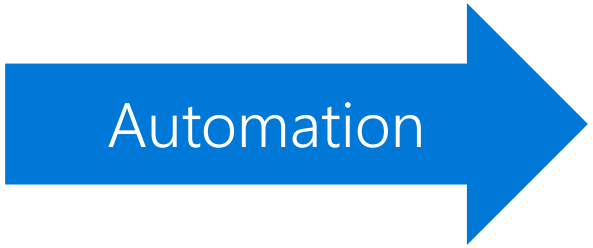


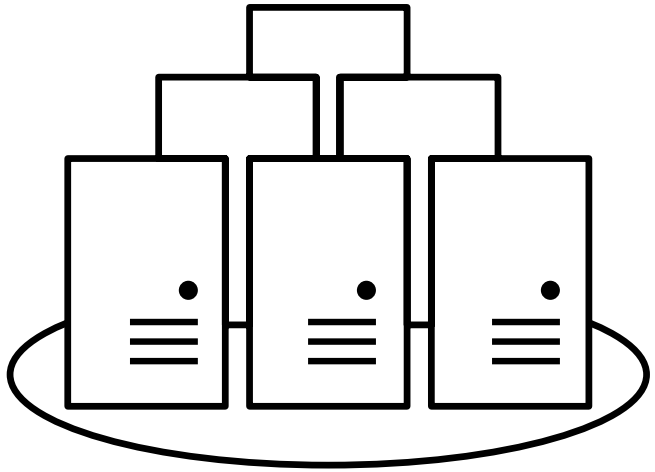
Infrastructure as Code

Implement ARM Templates

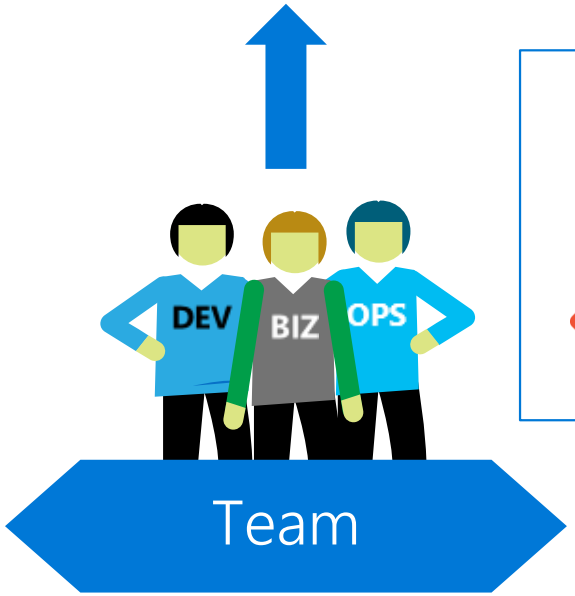
Big Picture: Infrastructure-As-Code







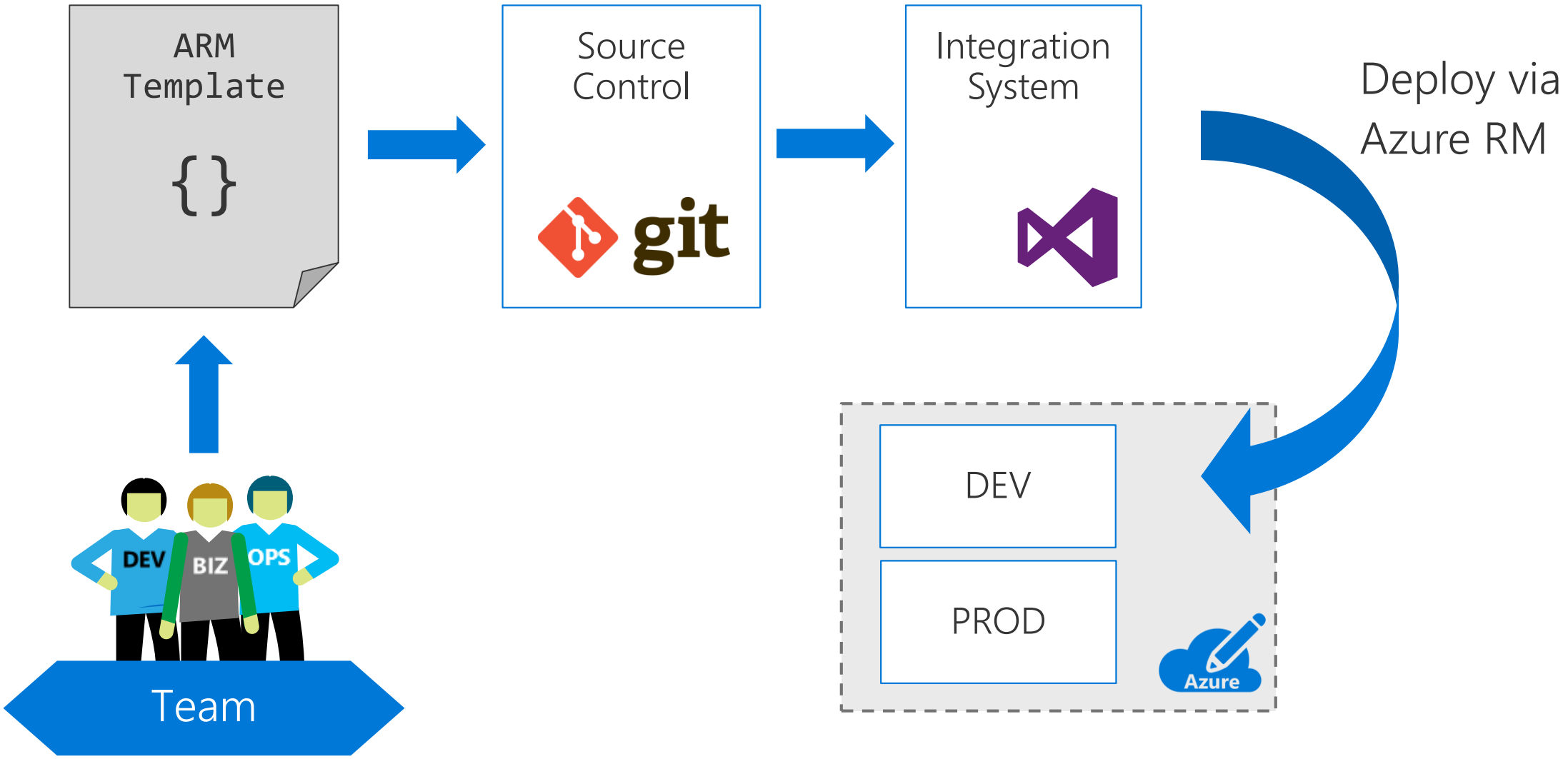
"All is software"



Source
Control

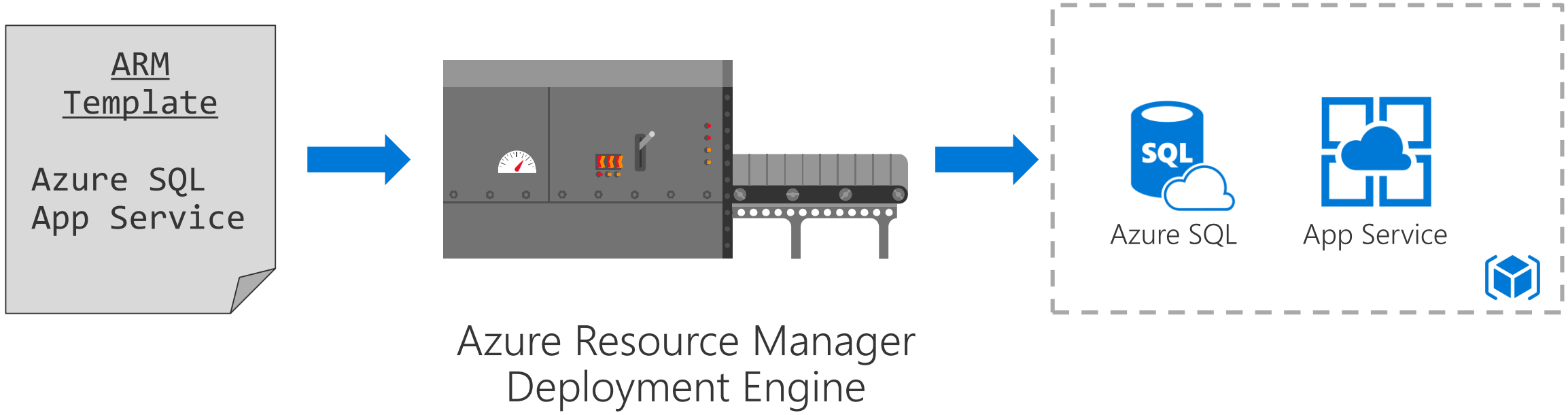
git

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": { }
}
```

https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-group-authoring-templates

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

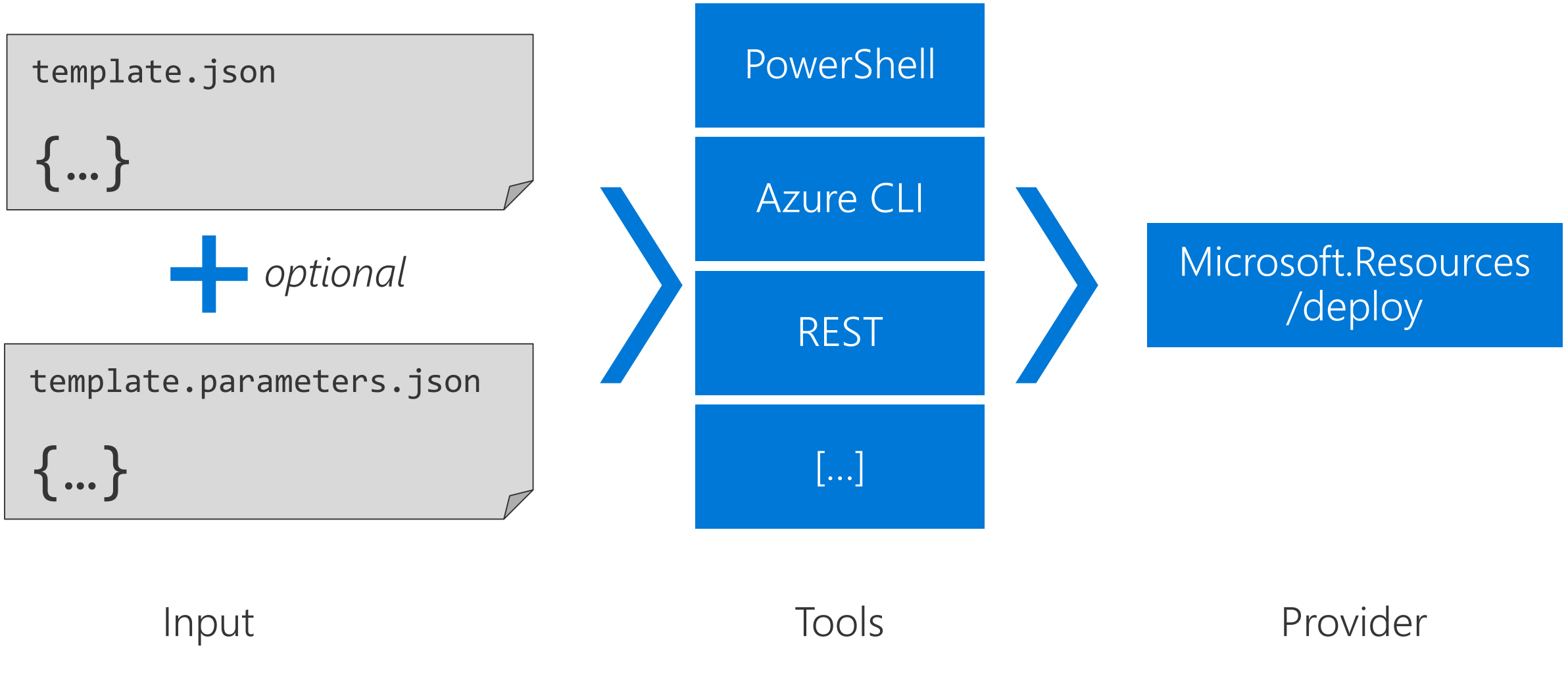
```
concat()          # Concatenate two or more strings
resourceId()      # Get resource ID by name
uniqueString()    # Generate a 13-char hash from input

reference()       # Returns object with resource runtime state

resourceGroup() # Get resource group ID of deployment
subscription()  # Get subscription group ID of deployment
```

https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-group-template-functions

Advanced ARM Templates

Copy Loops (Link)
Conditionals (Link)
Linked Templates (Link)

Deploying ARM Templates



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
```

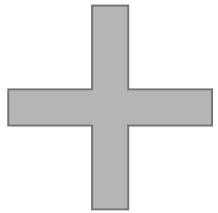
Deploying with PowerShell

```
# 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

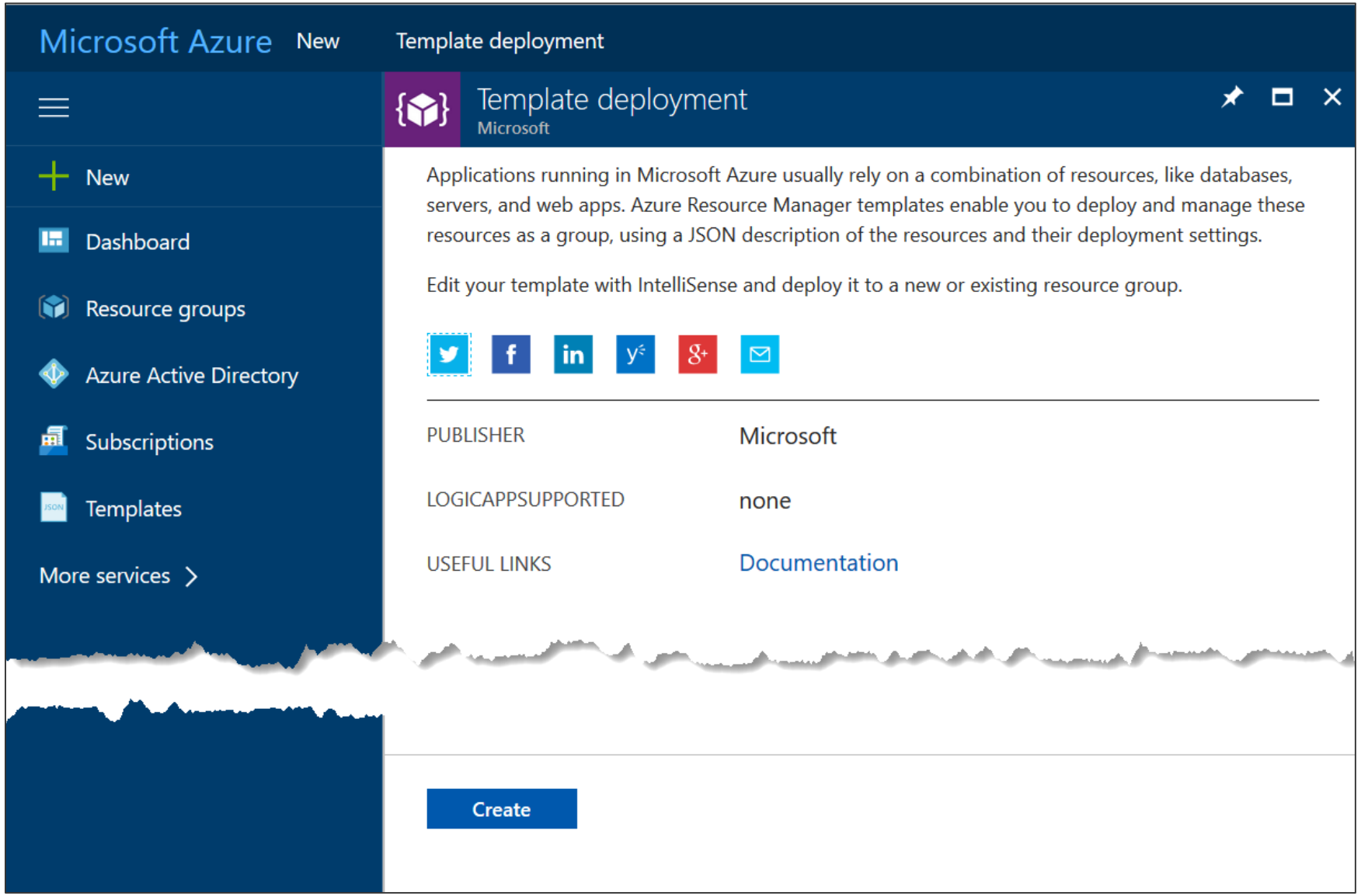
```
# Prerequisite: OAuth Authorization (Bearer Token)

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: {...}
  }
}
```

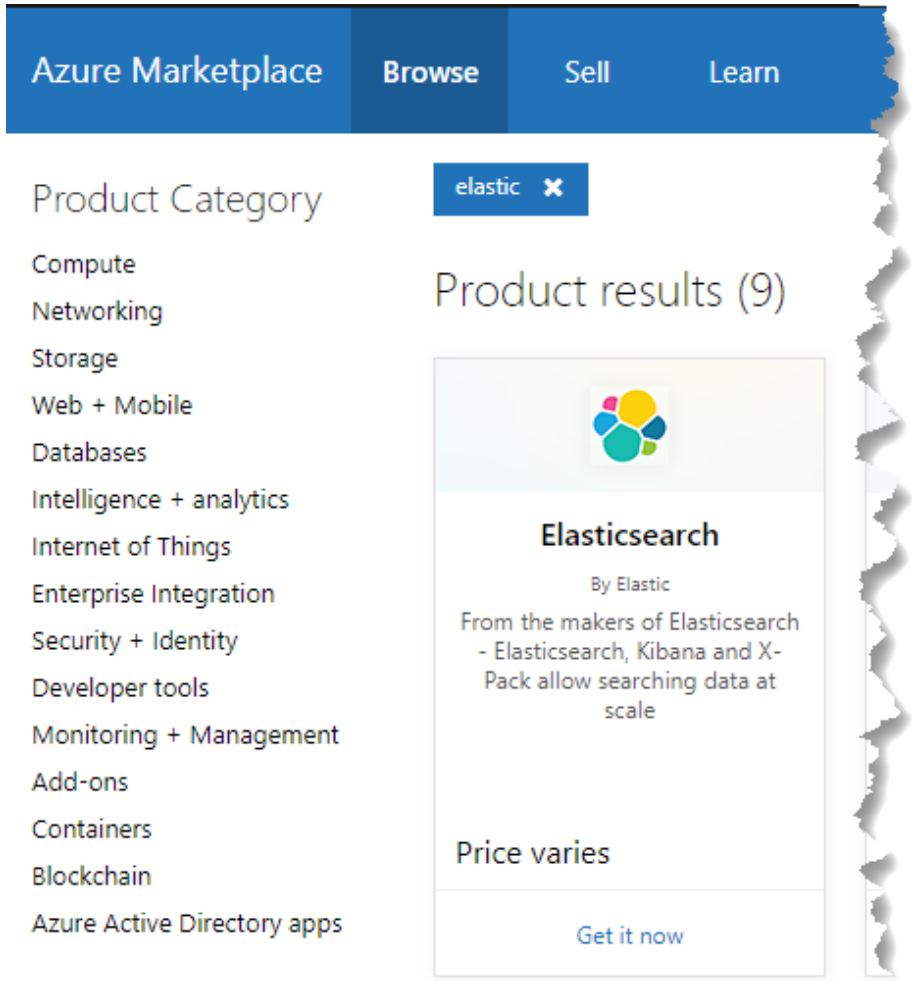
Deploy from Portal

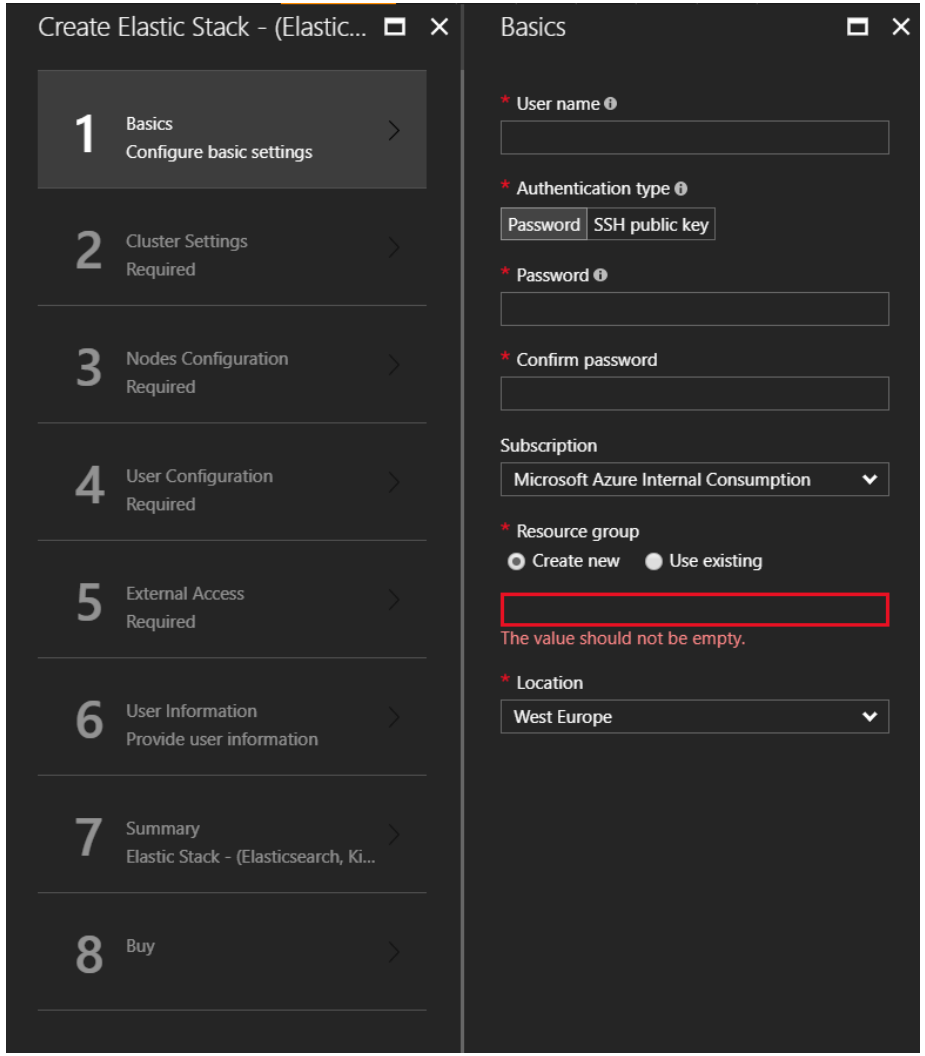


Demo:
Deploying an ARM Template

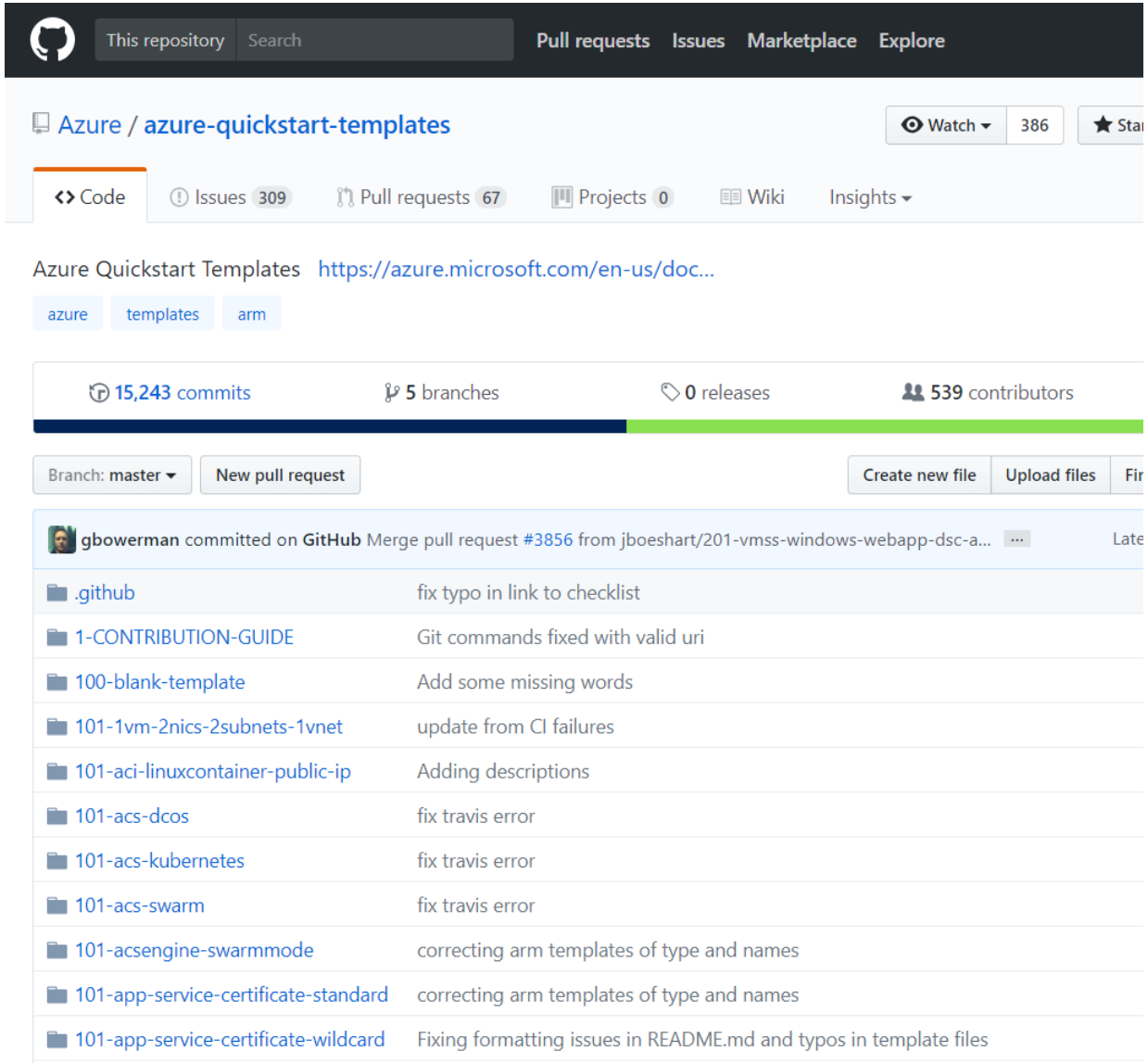
Simon Schwingel

Azure Marketplace





ARM Quickstart Templates



Quickstart Templates Gallery:

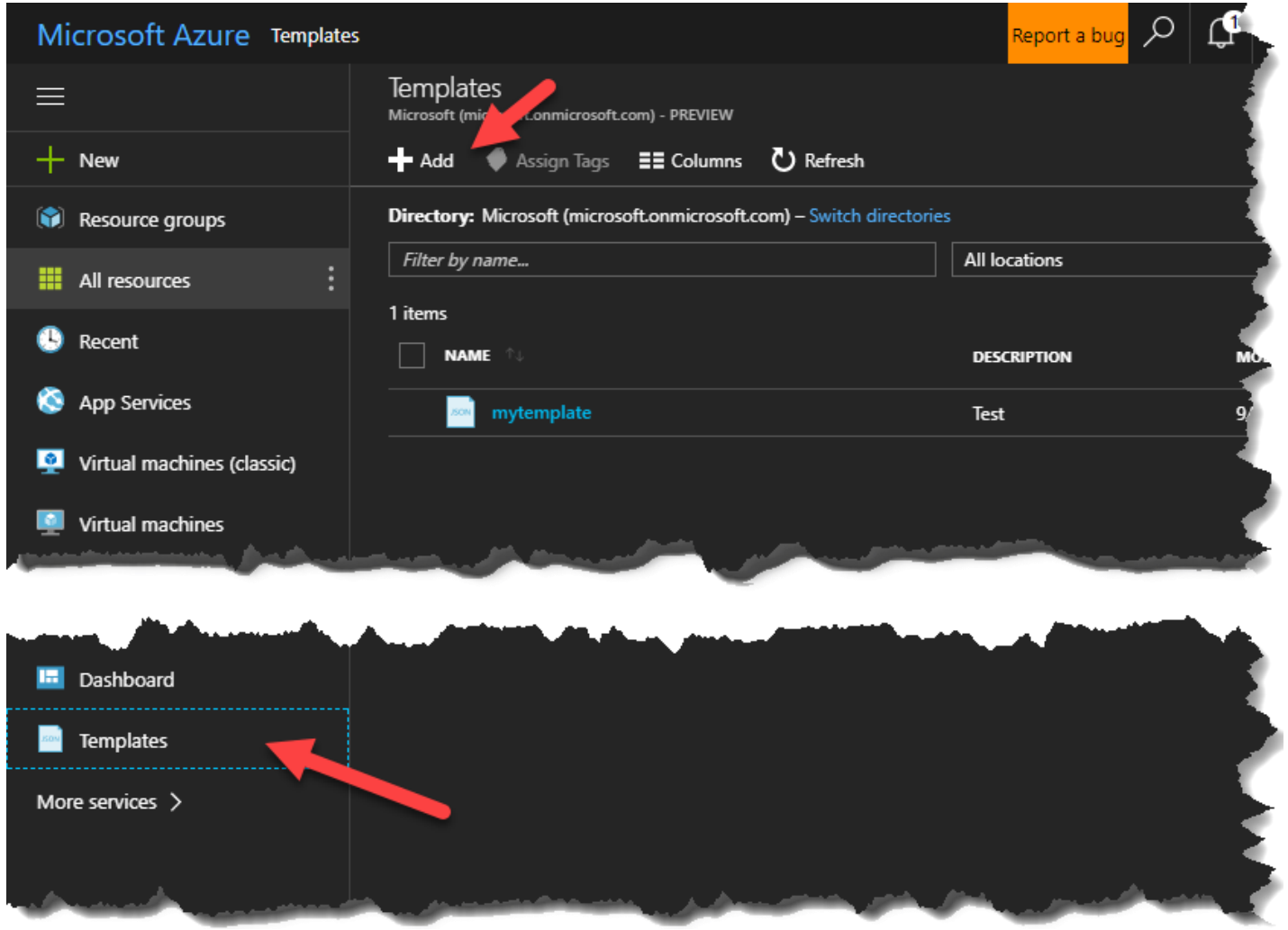
https://azure.microsoft.com/en-us/resources/templates/

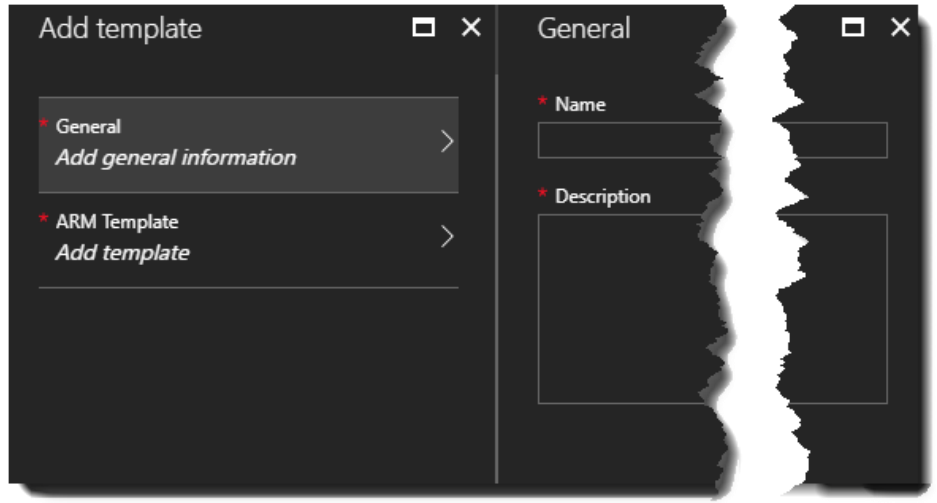
Quickstart Templates on GitHub

https://github.com/Azure/azure-quickstart-templates

>580 samples

Managing Private ARM Template in the Portal



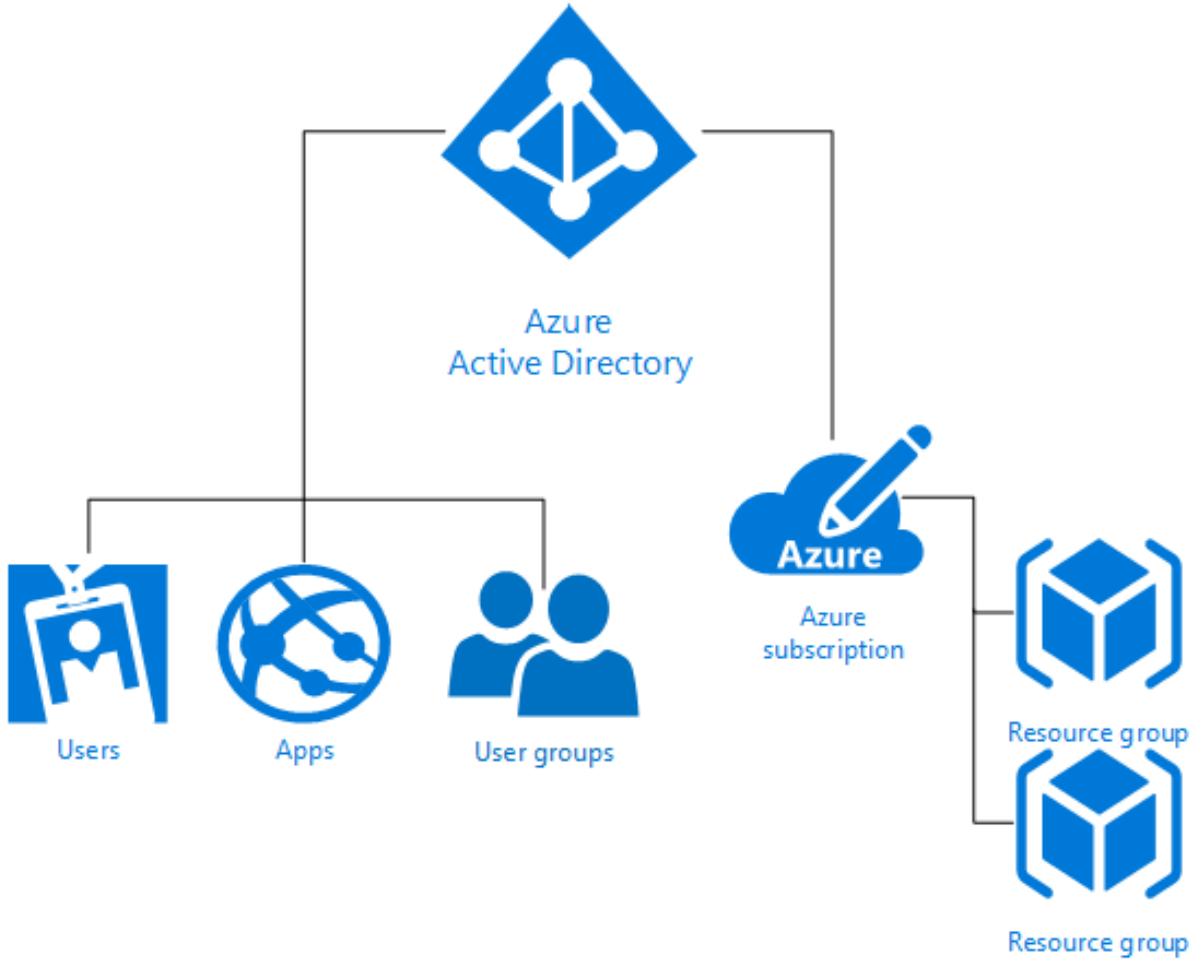


https://docs.microsoft.com/en-us/azure/marketplace-consumer/mytemplates-getstarted

Infrastructure as Code

Design Access Control

Basics of Access Management in Azure



Core Built-in Roles

Owner

- Full Access
- Delegate to others

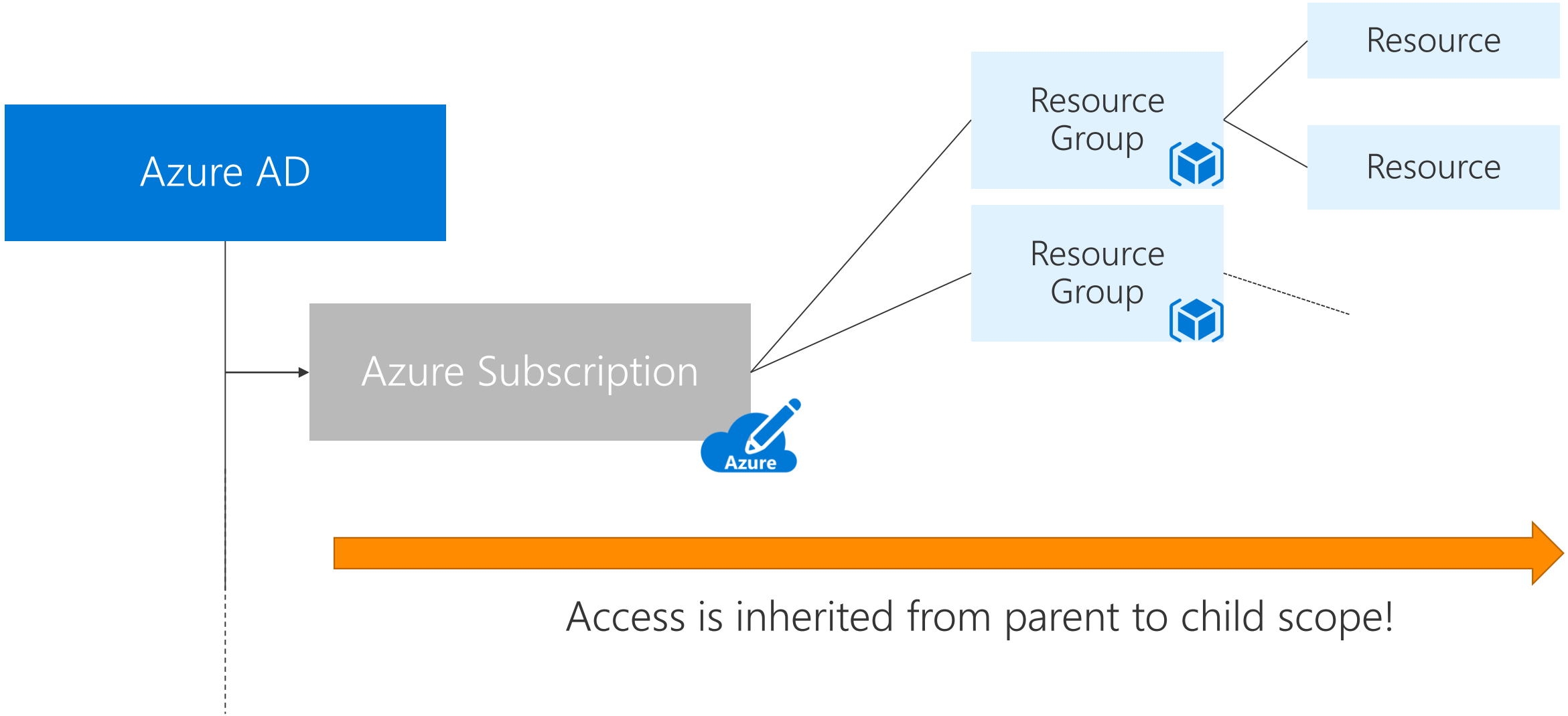
Contributor

- 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

Actions

Microsoft.Compute/locations/*
Microsoft.Compute/virtualMachines/*
…

Not-Actions

Scope

/ (globally available, built-in Role)

Role:
Virtual Machine Contributor

Basic Anatomy of a Role - Contributor

Actions

*

Not-Actions

Microsoft.Authorization/*/Delete
Microsoft.Authorization/*/Write
Microsoft.Authorization/elevateAccess/Action

Scope

/ (globally available, built-in Role)

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
[...]
```

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

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

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

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

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/
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%)

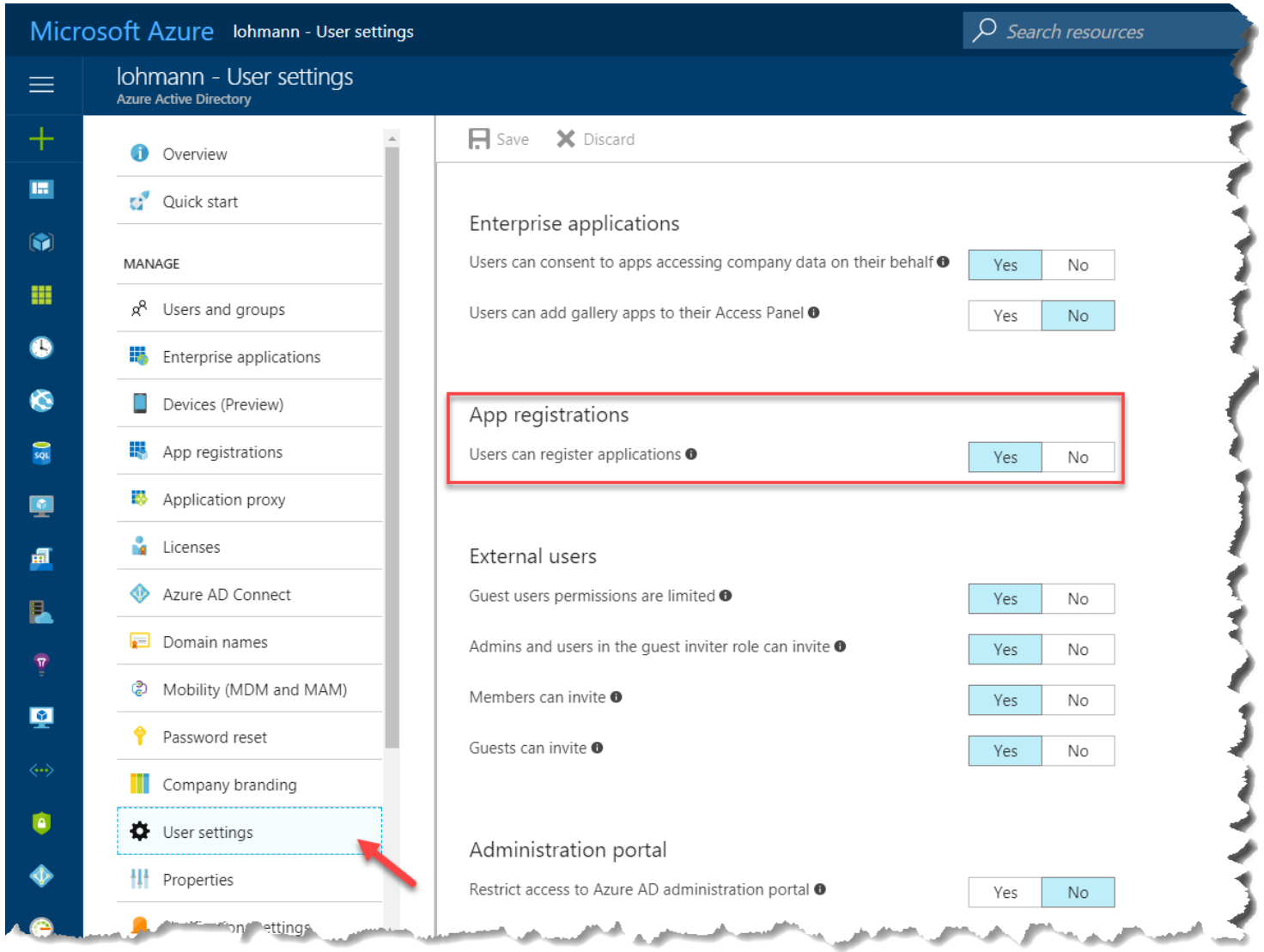
Control Access	
Use portal to create an Azure Active Directory application and service principal that can access resources	https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-group-create-service-principal-portal
Use Resource Manager authentication API to access subscriptions	https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-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

Create Service Principles - Options

- Portal (Link)
- CLI 1.0 (Link)
- CLI 2.0 (Link)
- PowerShell (Link)

Required Permissions in Azure AD



"App registrations" is required for Service Principle creation.

Creating a Service Principle with CLI 2.0

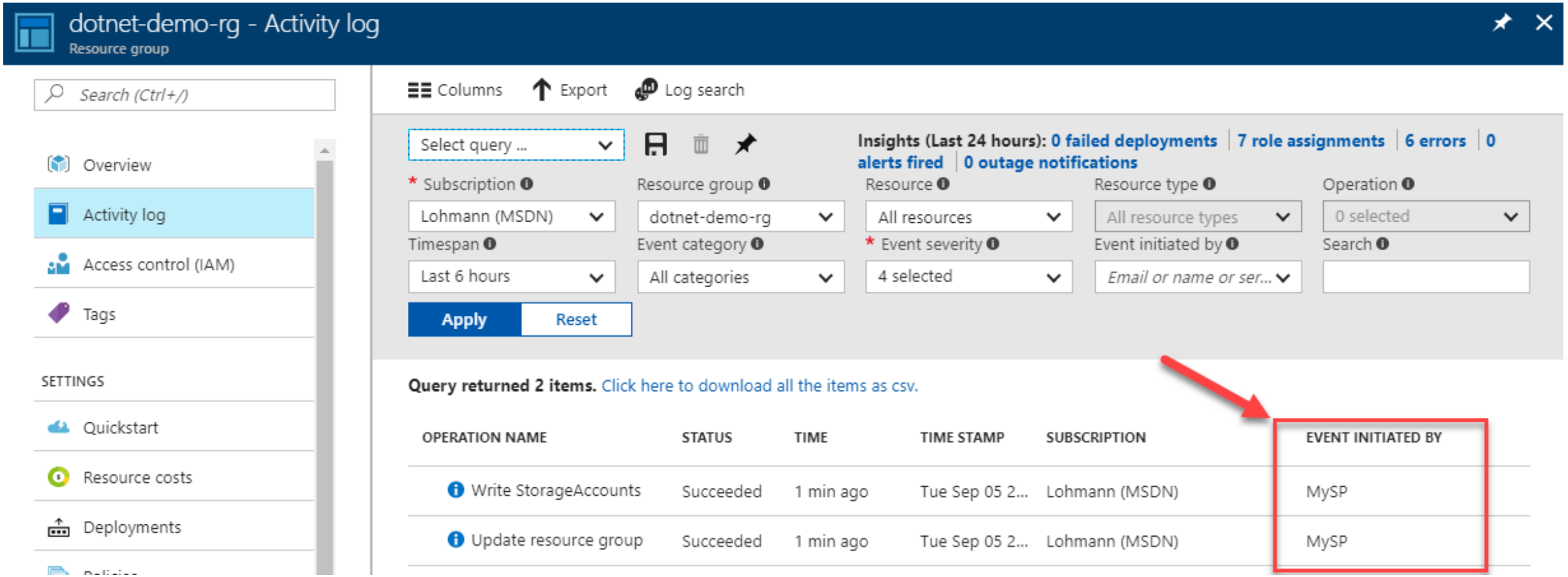
```
# List all existing App registrations
az ad app list

# 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-...-...-...-..."
}
```

https://docs.microsoft.com/en-us/cli/azure/create-an-azure-service-principal-azure-cli?toc=%2Fazure%2Fazure-resource-manager%2Ftoc.json&view=azure-cli-latest

Usage Sample: Custom Code with .NET

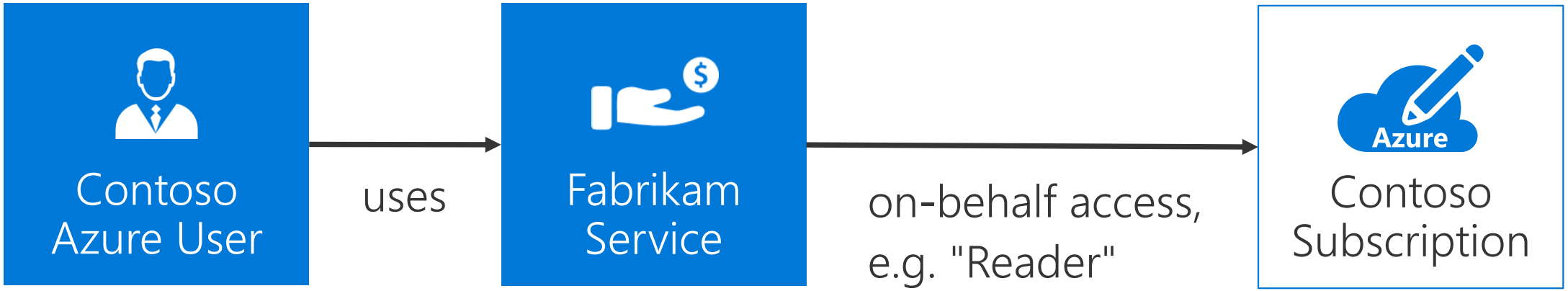
```
var tenantId = "a2583895-...";
var subscriptionId = "397015c1-...";

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.)



Demo:
Service Principles

Resource Manager API Authentication

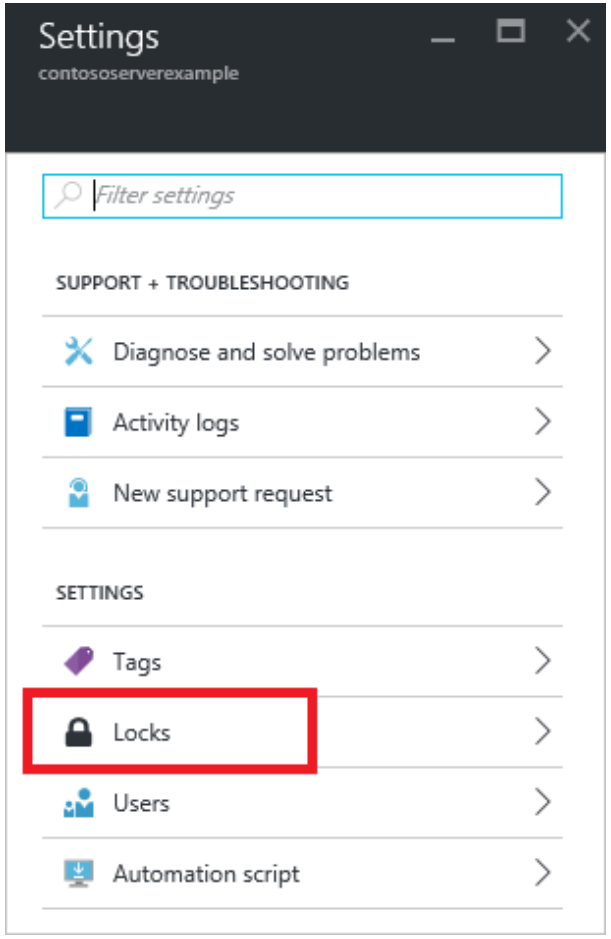


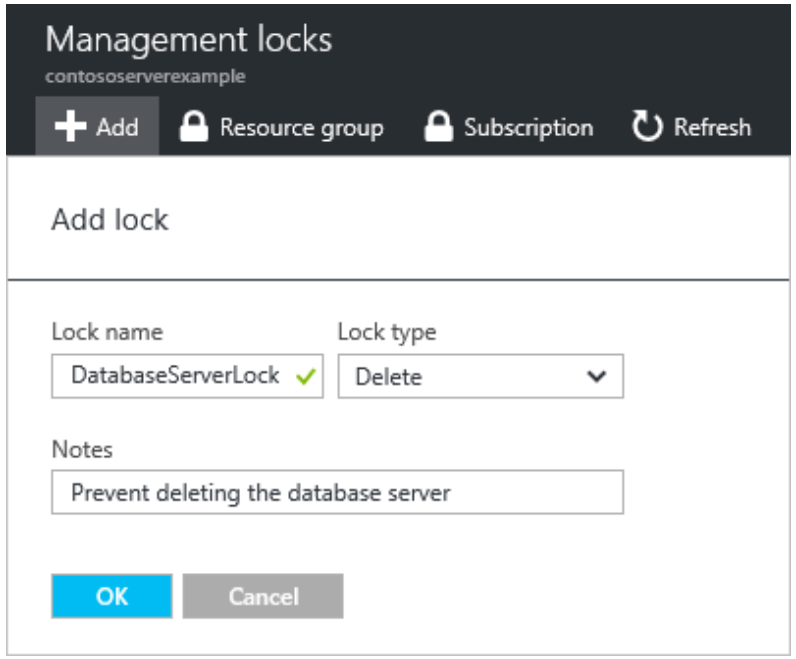
Scenario: Offer custom Azure Services

https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-manager-api-authentication

Resource Locks

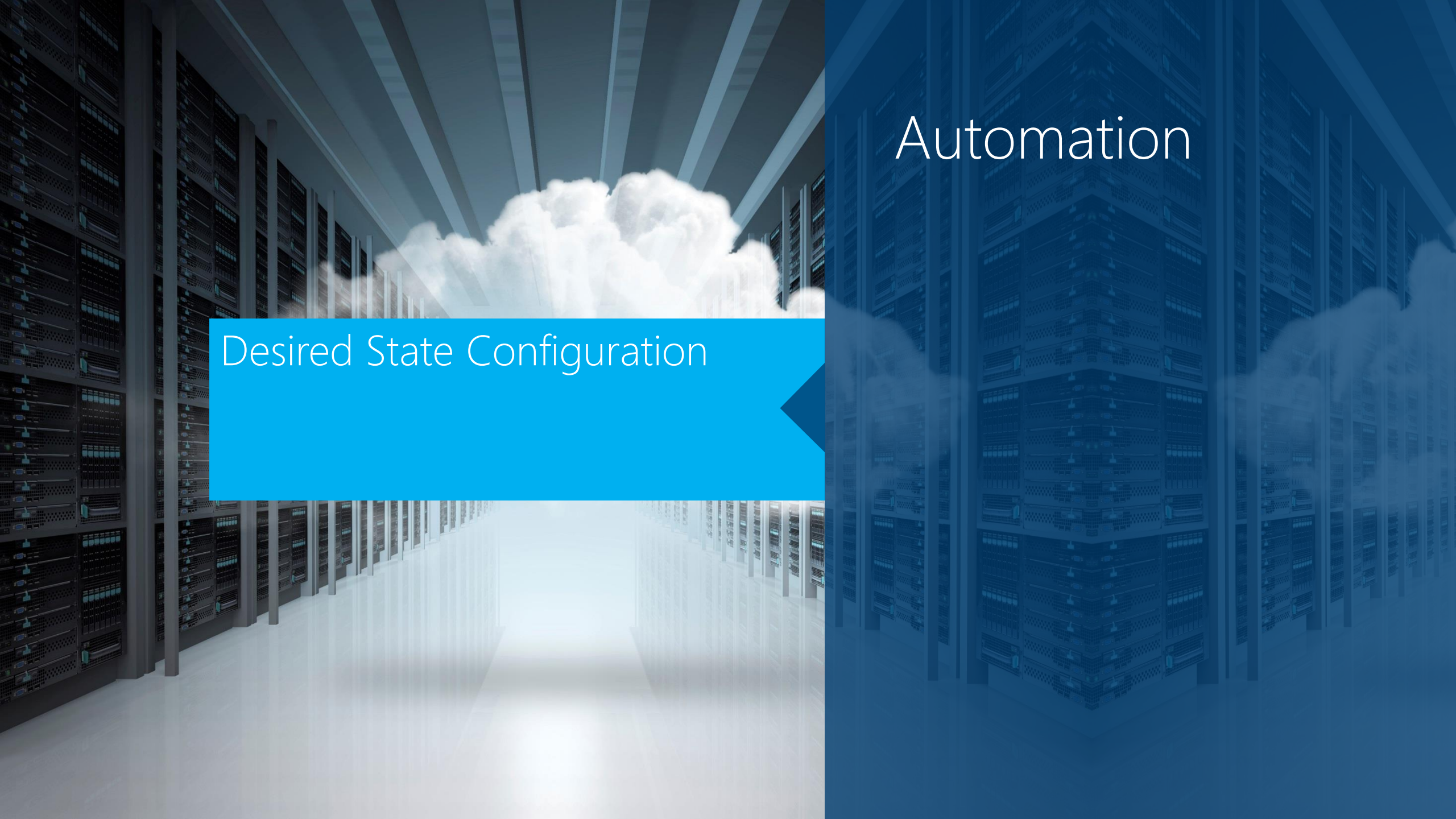
Protect resources against accidental deletion or changes





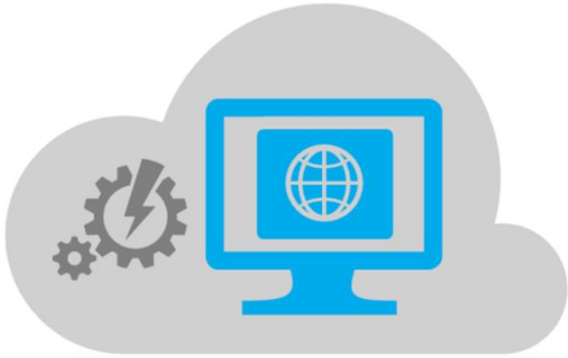
https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-group-lock-resources

Demo:
Resource Locks


Automation
Desired State Configuration

Desired State Configuration (DSC)

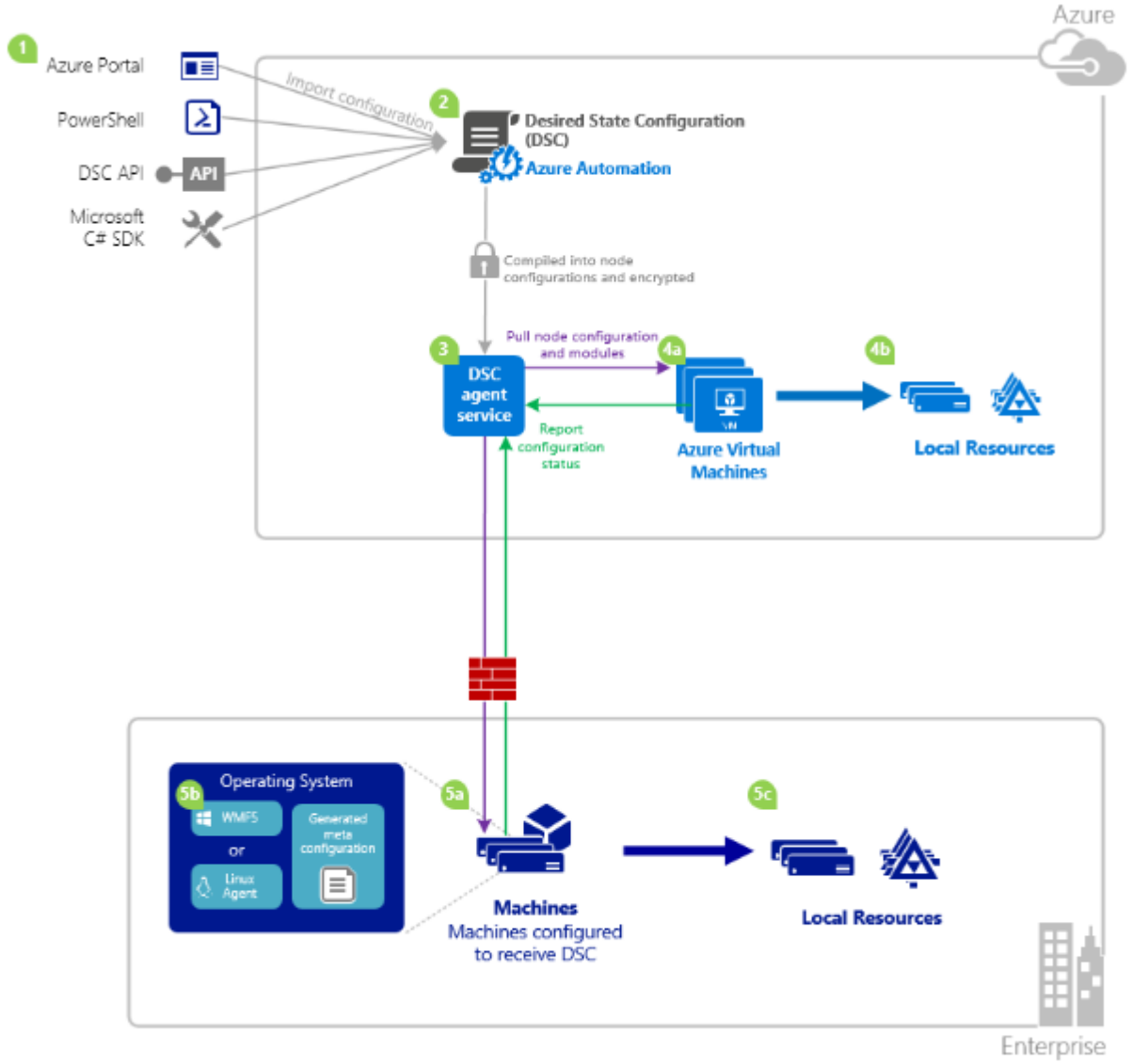
- Azure Automation Desired State Configuration (DSC) allows you to consistently deploy, monitor, and automatically update the desired state of all your IT resources, at scale from the cloud.
- Built on PowerShell DSC and can align machine configuration with a specific state across physical and virtual 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



1. An actor imports a DSC configuration to Azure Automation
2. An actor compiles the DSC configuration into node configuration and it's encrypted by Azure Automation
3. The node configuration is placed in the DSC agent service and is then pulled by cloud or on-premises machines
- 4a. Cloud – 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
- 4b. Azure VMs conform to the desired state
- 5a. On-premises – node configuration is pulled on to local machine configured to receive DSC.
- 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
- 5c. Local machines conform to the desired state

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.

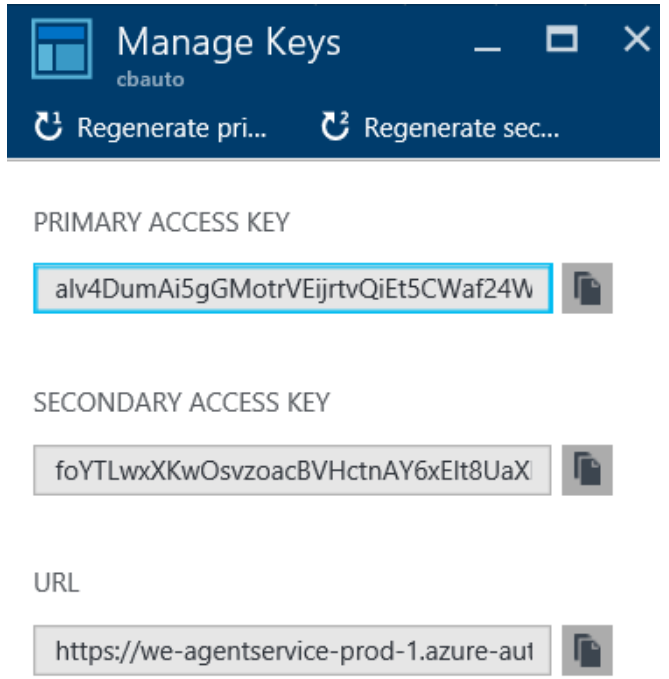
DSC Metaconfiguration & Secure Registration

- 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.



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