

Effortlessly Deploying a PI System in Azure

Alex Pedchenko



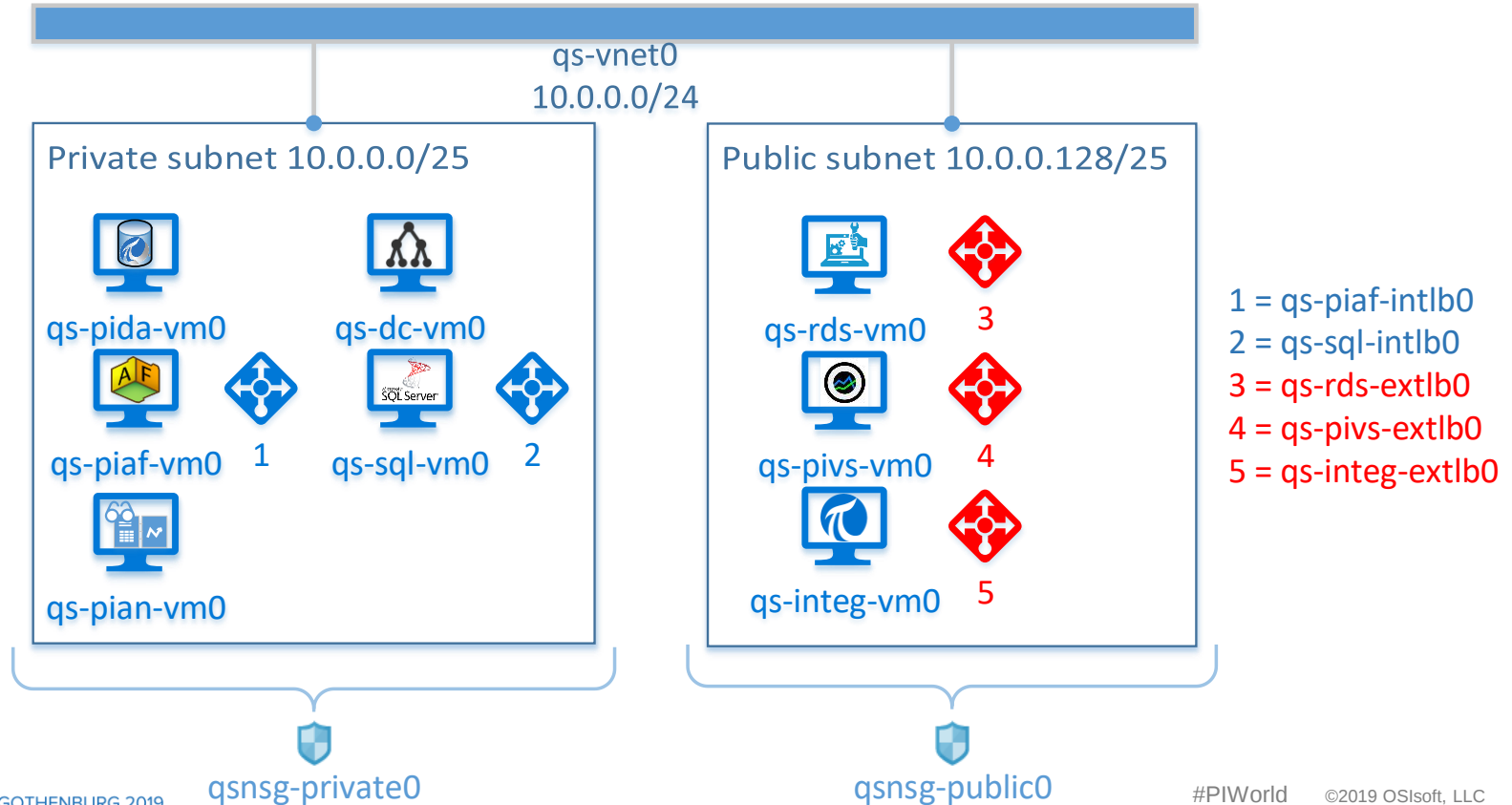
Outline

- Cloud-based PI System
 - PI System Deployment Sample for Microsoft Azure
 - Azure Resource Manager Templates
 - Desired State Configuration
 - PI Asset Framework VM Demo
 - Azure Disk Snapshot Recovery Demo
 - Conclusion and Future Work
 - Questions
- Infrastructure as Code (IaC)

Cloud-based PI System

- NOT OCS
- Self-contained PI environment in Microsoft Azure
- Motivation:
 - Cost/Ease
 - Availability (SLA)
 - Elasticity
 - Accessibility
 - On demand
 - Managed services

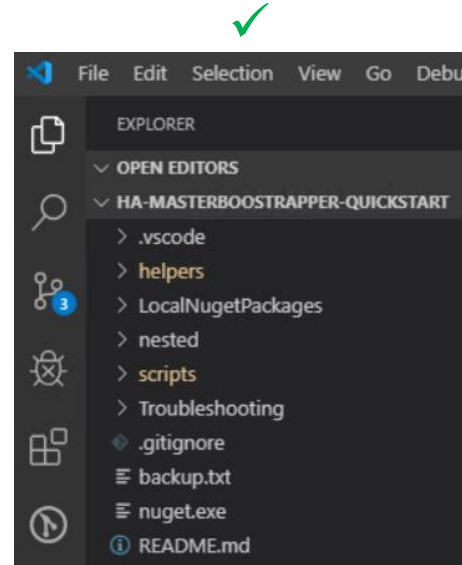
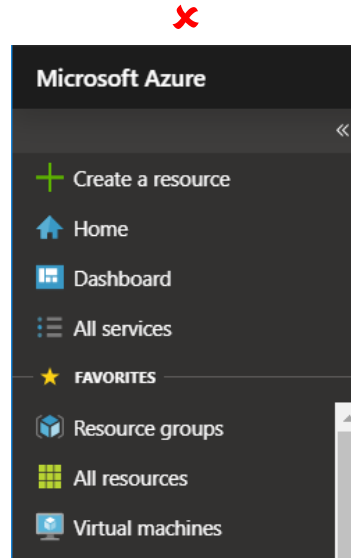
PI System Deployment Sample



Azure Portal vs Infrastructure as Code

- Portal

- + intuitive
- + visual
- + quick to pick up



- IaC

- Learning curve
- **Initial** time investment
- Need to read

- Scalability, repeatability, accuracy
- Speed of environment deployment
- Black box use for fast results

Infrastructure as Code: Core Components

Infrastructure Deployment

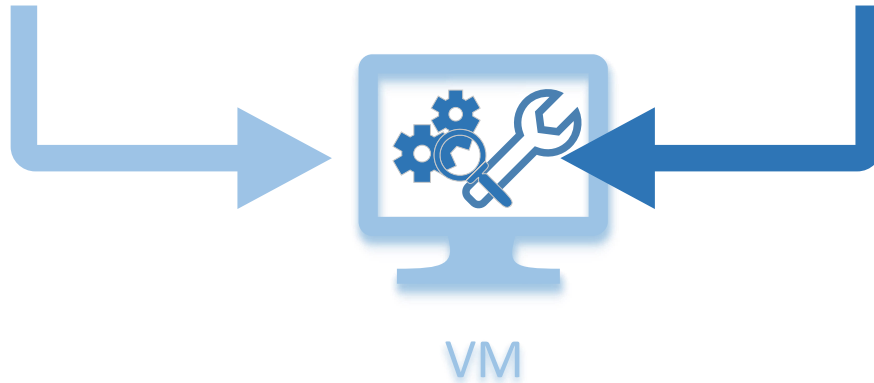


Azure Resource Manager
(ARM) Templates

Internal VM Configuration



PowerShell Desired State
Configuration (DSC)
Scripts



Azure Resource Manager Templates

- Declarative JSON templates
- Idempotence
- Resource dependencies
- Full environment deployment
- Role-Based Access Control

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

ARM Template: Parameters

- Values chosen by a user during a deployment
- Options for passing parameters:
 1. Specify in ARM template
 - Default value
 2. Include separate json parameters file
 - DoStuff.json → DoStuff.parameters.json
 3. Pass inline during deployment
 - New-AzureRmDeployment -TemplateFile 'D:\Repo\Dostuff.json' -storageAccountSku 'Premium_LRS' -storageAccountPrefix 'meh'

```
"parameters": {
  "storageAccountSku": {
    "type": "string",
    "defaultValue": "Standard_LRS",
    "allowedValues": ["Standard_LRS", "Standard_ZRS", "Standard_GRS", "Standard_RAGRS", "Premium_LRS"],
    "metadata": {
      "description": "Storage account sku"
    }
  },
  "storageAccountKind": {
    "type": "string",
    "defaultValue": "StorageV2",
    "allowedValues": ["Storage", "StorageV2", "BlobStorage", "FileStorage", "BlockBlobStorage"],
    "metadata": {
      "description": "Storage account type"
    }
  },
  "location": {
    "type": "string",
    "defaultValue": "[resourceGroup().location]",
    "metadata": {
      "description": "Location for resource(s) deployed by ARM template"
    }
  },
  "storageAccountPrefix": {
    "type": "string",
    "defaultValue": "qs",
    "metadata": {
      "description": "Prefix used in automatic name generation of storage account"
    }
  }
},
```

ARM Template: Variables, Resources and Outputs

```
"variables": {  
  "storageAccountName": "[concat(parameters('storageAccountPrefix'), substring(uniqueString(resourceGroup().id), 0, 4))]"  
},  
"resources": [  
  {  
    "name": "[variables('storageAccountName')]",  
    "type": "Microsoft.Storage/storageAccounts",  
    "apiVersion": "2019-04-01",  
    "location": "[parameters('location')]",  
    "kind": "[parameters('storageAccountKind')]",  
    "sku": "[parameters('storageAccountSku')]",  
    "tags": {},  
    "properties": {}  
  }  
],  
"outputs": {  
  "storageAccountId": {  
    "type": "string",  
    "value": "[resourceId('Microsoft.Storage/storageAccounts', variables('storageAccountName'), '2019-04-01')]"  
  }  
}
```

Variables: convenient substitutions for computational results used throughout template

Resources: cloud components to deploy to Azure (e.g., VMs, load balancers, network security groups, etc.)

Outputs: information produced by template for external consumption

Desired State Configuration

- PowerShell platform used for deployment, configuration and management of systems
- Declarative
- Idempotent
- Comprised of three primary components
 - Configuration
 - Resources
 - Local Configuration Manager

DSC: Configuration

- Declarative resources
- Script block keyword
 - Import-DscResource can be used inside; this is not a cmdlet
 - Can use PowerShell code within this block
- Contains node blocks that designate on which machines to apply specified configurations

```
Configuration PIAF {
    Import-DscResource -ModuleName PSDesiredStateConfiguration
    Import-DscResource -ModuleName xActiveDirectory
    param(
        [Parameter(Mandatory)]
        [pscredential]$runAsCredential,
        [Parameter(Mandatory)]
        [pscredential]$svcCredential
    )

    $domainNetBiosName = ((Get-WmiObject -Class Win32_NTDomain -Filter `
        "DnsForestName = '$((Get-WmiObject -Class Win32_ComputerSystem).Domain)'" ).DomainName)
    $PIAFSvcAccountUsername = $svcCredential.UserName
    $runAsAccountUsername = $runAsCredential.UserName
    $domainRunAsCredential = New-Object System.Management.Automation.PSCredential -ArgumentList `
        ("{$domainNetBiosName\$($runAsAccountUsername)", $runAsCredential.Password)

    Node localhost {
        WindowsFeature ADPS {
            Name = 'RSAT-AD-PowerShell'
            Ensure = 'Present'
        }

        xADUser ServiceAccount_PIAF {
            DomainName = $domainNetBiosName
            Username = $PIAFSvcAccountUsername
            CannotChangePassword = $true
            Description = 'PIAF server service account.'
            DomainAdministratorCredential = $domainRunAsCredential
            Enabled = $true
            Ensure = 'Present'
            Password = $svcCredential
            DependsOn = '[WindowsFeature]ADPS'
        }
    }
}
```

DSC: Resources

```
[ClassVersion("1.0.0.0"), FriendlyName("WindowsFeature")]
```

```
class MSFT_WindowsFeature : OMI_BaseResource
```

```
{
```

- **[Key, Description("The name of the role or feature to install or uninstall.")] String Name;**

```
[Write, Description("Specifies whether the role or feature should be installed or  
uninstalled. To install the feature, set this property to Present. To uninstall the feature, set  
the property to Absent."), ValueMap{ Present, "Absent"}, Values{ Present, "Absent"}]
```

```
String Ensure;
```

```
[Write, Description("Specifies whether the subfeatures of the main feature should also  
be installed.")] Boolean IncludeAllSubFeature;
```

- **[Write, Description("The path to the log file to log this operation.") String LogPath;**

```
[Write, Description("A credential, if needed, to install or uninstall the role or feature."),  
EmbeddedInstance("MSFT_Credential")] String Credential;
```

```
[Read, Description("The display name of the retrieved role or feature.")] String  
DisplayName;  
};
```

DSC: Resources

```
WindowsFeature ADPS {
  Name      = 'RSAT-AD-Tools'
  Ensure    = 'Present'
}

xADUser ServiceAccount_P {
  DomainName
  UserName
  CannotChangePassword
  Description
  DomainAdministratorCredential
  Enabled
  Ensure      = 'Present'
  Password   = $svcCredential
  DependsOn  = '[WindowsFeature]ADPS'
}
```

PS C:\SingleInstanceAzureQuickstart\OSIsoft-Azure-QuickStart\helpers> **Get-DscResource -Syntax WindowsFeature**

- Get-DscResource -Syntax <ResourceName>

- Get-TargetResource: reporting; not used by LCM
- Test-TargetResource: resource in desired state?
- Set-TargetResource: What to do if Test returns \$false

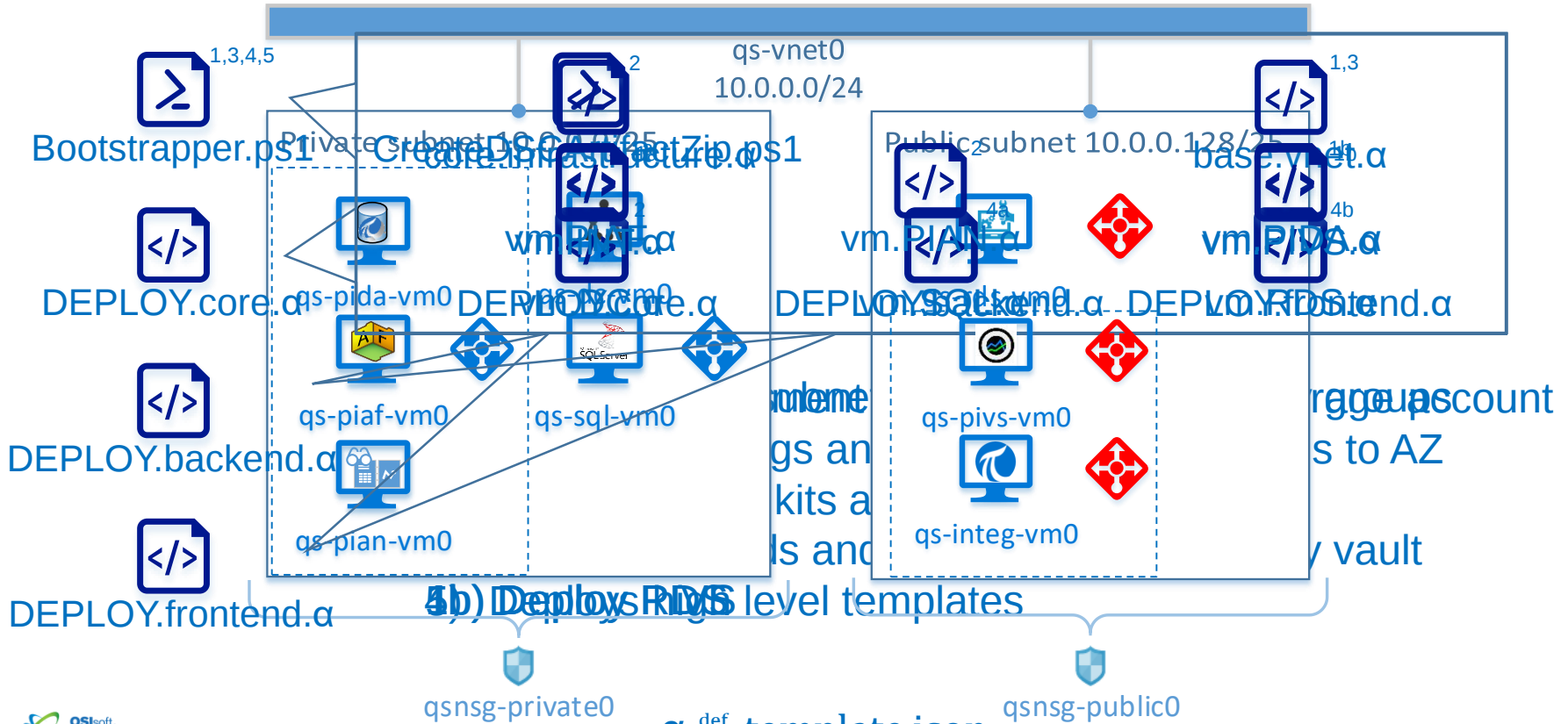
DSC: Local Configuration Manager

- Execution engine which acts between configuration and resources
- When a DSC configuration is invoked, a <NodeName>.mof file is generated; LCM accepts .mof files as input

```
[DSCLocalConfigurationManager()]\nconfiguration LCMConfig\n{\n    Node localhost\n    {\n        Settings\n        {\n            RefreshMode = 'Push'\n        }\n    }\n}
```

- LCM settings can be modified using...
- ApplyOnly, **ApplyAndMonitor** or ApplyAndAutoCorrect
- Must complete initial configuration
- Can restart node if necessary

PI System Deployment Sample



DEMO

Deploying a PI Asset Framework VM Into
a Pre-existing Environment

Demo Takeaway: Results

The screenshot displays the PI System Explorer (Administrator) interface. The main window shows the 'Demo-PI-Points' table with columns for Name, Value, and Time Stamp. The table contains several rows of data, including 'BA:ACTIVE.1' (Inactive), 'BA:CONC.1' (0 °C), 'BA:LEVEL.1' (0), 'BA:PHASE.1' (Phase1), 'BA:TEMP.1' (0 °C), 'CDEP158' (0), 'CDM158' (Manual), 'CDT158' (51.585 °C), 'SINUSOID' (43.142), and 'SINUSOIDU' (43.142). A red box highlights the status bar at the bottom, which reads: 'PI Analysis Service: qs-pian-vm0.qs.int, Status updated at 8/26/2019 11:53:43 PM'. To the right of the table, a function library is visible, showing a list of functions including 'Abs(number x)' with a description: 'Return the absolute value of an integer or real number. Example: Abs(-2.2) [Returns 2.2]'.

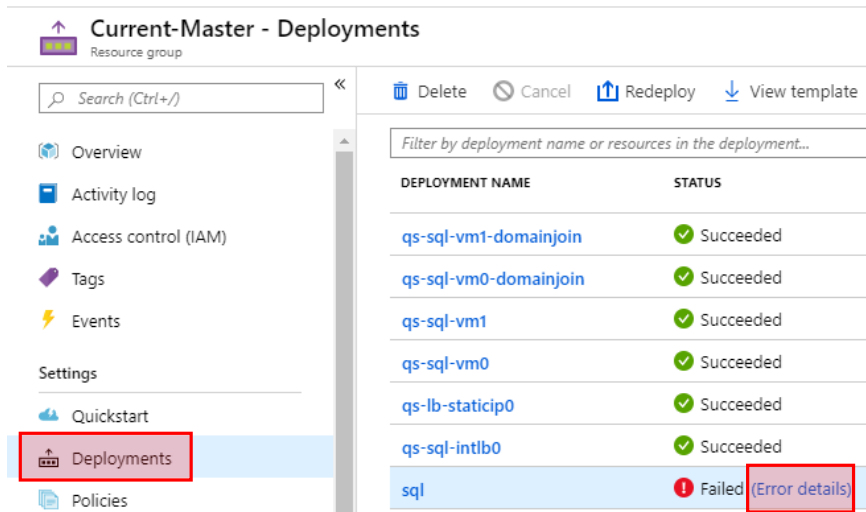
Name	Value	Time Stamp
BA:ACTIVE.1	Inactive	8/26/2019 11:44:14 PM
BA:CONC.1	0 °C	8/26/2019 11:44:14 PM
BA:LEVEL.1	0	8/26/2019 11:44:14 PM
BA:PHASE.1	Phase1	8/26/2019 11:44:14 PM
BA:TEMP.1	0 °C	8/26/2019 11:44:14 PM
CDEP158	0	8/26/2019 11:44:14 PM
CDM158	Manual	8/26/2019 11:44:14 PM
CDT158	51.585 °C	8/26/2019 11:44:14 PM
SINUSOID	43.142	8/26/2019 11:44:14 PM
SINUSOIDU	43.142	8/26/2019 11:44:14 PM

PI Analysis Service: qs-pian-vm0.qs.int, Status updated at 8/26/2019 11:53:43 PM

Connected to the PI Analysis Service.

- Deploying PI AF into pre-existing environment
- Using IaC: ARM and DSC in action
- Using Azure portal to track deployments

Demo Takeaway: ARM Error Logging



Current-Master - Deployments
Resource group

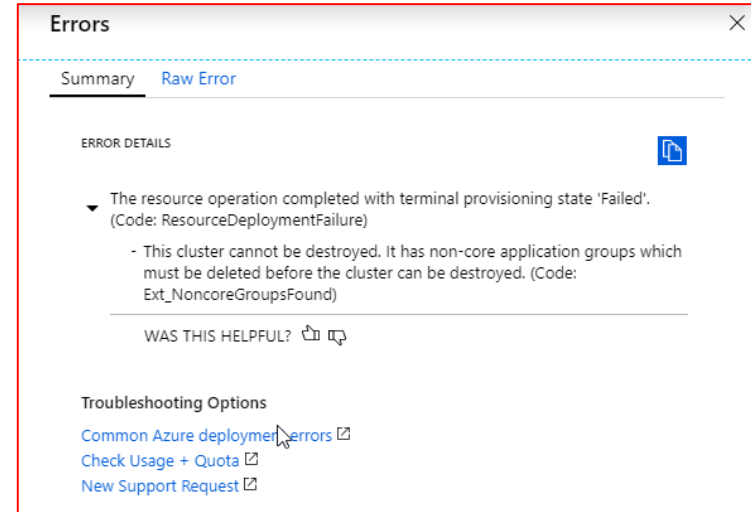
Search (Ctrl+ /)

Overview
Activity log
Access control (IAM)
Tags
Events
Settings
Quickstart
Deployments
Policies

Delete Cancel Redeploy View template

Filter by deployment name or resources in the deployment...

DEPLOYMENT NAME	STATUS
qs-sql-vm1-domainjoin	✓ Succeeded
qs-sql-vm0-domainjoin	✓ Succeeded
qs-sql-vm1	✓ Succeeded
qs-sql-vm0	✓ Succeeded
qs-lb-staticip0	✓ Succeeded
qs-sql-intlb0	✓ Succeeded
sql	❌ Failed (Error details)



Errors

Summary Raw Error

ERROR DETAILS

▼ The resource operation completed with terminal provisioning state 'Failed'. (Code: ResourceDeploymentFailure)

- This cluster cannot be destroyed. It has non-core application groups which must be deleted before the cluster can be destroyed. (Code: Ext_NoncoreGroupsFound)

WAS THIS HELPFUL? 👍 🗨

Troubleshooting Options

[Common Azure deployment errors](#) ⓘ

[Check Usage + Quota](#) ⓘ

[New Support Request](#) ⓘ

- Error details in “Deployments” blade of failed resource’s resource group
- Can also see in terminal when using “-Verbose” flag

Demo Takeaway: DSC Error Logging

- Located in JSON files found in:
“C:\Windows\System32\Configuration\ConfigurationStatus”
- Recall LCM default: ApplyAndMonitor at ~15 min

```
LCM: [ Start Resource ] [[WindowsFeature]ADPS]  "},
LCM: [ Start Test ]    [[WindowsFeature]ADPS]  "},
                                [[WindowsFeature]ADPS] The operation 'Get-WindowsFeature' started: RSAT-AD-PowerShell"},
                                [[WindowsFeature]ADPS] The operation 'Get-WindowsFeature' succeeded: RSAT-AD-PowerShell"},
LCM: [ Start Resource ] [[WindowsFeature]ADPS]  "},
LCM: [ Start Test ]    [[WindowsFeature]ADPS]  "},
                                [[WindowsFeature]ADPS] The operation 'Get-WindowsFeature' started: RSAT-AD-PowerShell"},
                                [[WindowsFeature]ADPS] The operation 'Get-WindowsFeature' succeeded: RSAT-AD-PowerShell"},
LCM: [ End Test ]      [[WindowsFeature]ADPS] False in 0.9740 seconds."},
LCM: [ End Resource ]  [[WindowsFeature]ADPS]  "},
                                [[WindowsFeature]ADPS] Installation succeeded."},
                                [[WindowsFeature]ADPS] Successfully installed the feature RSAT-AD-PowerShell."},
LCM: [ End Set ]       [[WindowsFeature]ADPS]  in 61.3350 seconds."},
LCM: [ End Resource ]  [[WindowsFeature]ADPS]  "},
```

DEMO

VM Recovery Using Azure
Disk Snapshots

Demo Takeaway: Results

pida	✓ Succeeded	8/27/2019, 12:21:06 AM	20 minutes 1 second
☐	NAME ↑↓	TYPE ↑↓	
☐	 dataSnap-qs-pida-vm0-datadisk0binaries	Snapshot	
☐	 dataSnap-qs-pida-vm0-datadisk1archives	Snapshot	
☐	 dataSnap-qs-pida-vm0-datadisk2queues	Snapshot	
☐	 dataSnap-qs-pida-vm0-datadisk3backups	Snapshot	
☐	 osSnap-qs-pida-vm0_OsDisk_1_35c3fb4e6dda...	Snapshot	
pida	✓ Succeeded	8/23/2019, 2:14:52 PM	6 minutes 9 seconds

- Restoration of a VM from a previously saved state
- Much, much faster than from scratch deployment
- Cons: large/not portable, restoration has to be to exact same environment

Conclusions and Future Direction

- Comparative ease, speed and low cost of deploying the PI System into a cloud platform
- Power of Infrastructure as Code in deployment and disaster recovery scenarios
- Advantages of leveraging ARM and DSC
 - Declarative
 - Idempotent
 - Tried and tested
- Future direction:
 - HA PI environment
 - Authoring DSC resources from PI validation scripts

About the Authors



- DevOps Engineer
 - Alexander Pedchenko
 - OSIssoft
 - apedchenko@osisoft.com
-
- Work would not be possible without...
 - Steven Rea
 - Valentin Ivanov
 - Charlston Bulacan



Questions?

Please wait for
the **microphone**

State your
name & company



Please remember to...

Complete Survey!

Navigate to this session in
mobile agenda for survey

