AX 2012 ARM Template

Microsoft
Dynamics AX

About

The purpose of this document is to present an in-depth look of the design of the ARM template that can be used to deploy a High Availability environment of the Dynamics AX 2012 R3 on Azure Resource Manager.

Disclaimer

This template is provided as is, without any warranties or liablity. The idea is that partners/customers will use this as a sample template, customize it to their needs and test it thoroughly before using it in any kind of production deployment.

Skills Requirement

Familarity with Azure, ARM Templates, and PowerShell Desired State Configuration is required to successfully deploy AX 2012 R3 enviornments using this template and troubleshoot any deployment issues that may arise.

Solution Overview

Key highlights:

- **No custom image for Dynamics AX VMs**

This template uses VM images already available on Azure and published by Microsoft. With this approach, there is no cost to maintain a custom image with updates and pre-requirements. The pre-requirements are handled by the deployment scripts.

- **Usage of Powershell DSC resources to run machine configuration scripts**

Powershell DSC (Desired State Configuration) is a handy solution to setup Windows machines. There are many existing resources available published by the Powershell team to configure many aspects of Windows and its applications, such as, Active Directory, Network, DNS, Windows Features, Disks, SQL Server and SharePoint.

These resources are always being updated with new versions and its usage is as simple as calling a "method" and passing parameters, avoiding complicated powershell scripts and producing a cleaner code. It is also idempotent by nature, which allows to re-execute the scripts in the same machine without compromising its state. We will discuss more about Powershell DSC later in this document.

- **Azure Active Directory and its consequences**

Instead of a local AD, we will be relying on Azure Active Directory. Unfortunately, AAD is not yet supported on ARM so this template is not able to deploy it. As a consequence, a specific setup is required to connect an ASM/RDFE VNet with an ARM VNet and an additional configuration is also needed to set the DNS name for the RDS connection broker. These configurations are not supported by this project.

When AAD gets available on ARM, all this troublesome configuration can be replaced by some lines in the template.

ARM Template pre-configuration

This ARM Template expects that an **Azure Active Directory** and an **Active Directory Domain Services** are already configured and accessible from an **ARM VNet**. It also expects that the services accounts are already created. They are:

- Application Object Server (AOS) Account
- AX Service Account
- Business Connector Proxy Account
- Dynamics Installation Account (domain administrator)
- Sharepoint Service Account
- SQL Server Service Account

It also expects that the **DNS round robin name** for Remote Desktop Services Connection Broker High Availability is configured in the DNS Server. For instance:

The domain name is “contoso.com”. You have to create the record in the DNS Server for the connection broker round-robin, like:

“rdsfarm.contoso.com” with IP 10.3.0.8 (static private IP for the first connection broker server)
“rdsfarm.contoso.com” with IP 10.3.0.9 (static private IP for the second connection broker server)

Classic and ARM Vnet pre-configuration

In order to allow communication between VMs in the ARM Vnet to AAD/AD DS in the classic Vnet, you can use one of the following two approaches.

- **Virtual network peering**

If your AAD exisits in the same subscription and same azure region, you can connect the Class Vnet where the AAD exists, with ARM Vnet using virtual network peering.

https://blogs.technet.microsoft.com/elliottf/2017/02/28/domain-joining-servers-to-azure-ad-domain-services-office-365-tenant/

- **Site-to-site VPN**

If you AAD exisits in a different subscription, you can use a site-to-site VPN to connect class and ARM Vnet.

https://docs.microsoft.com/en-us/azure/vpn-gateway/vpn-gateway-connect-different-deployment-models-portal

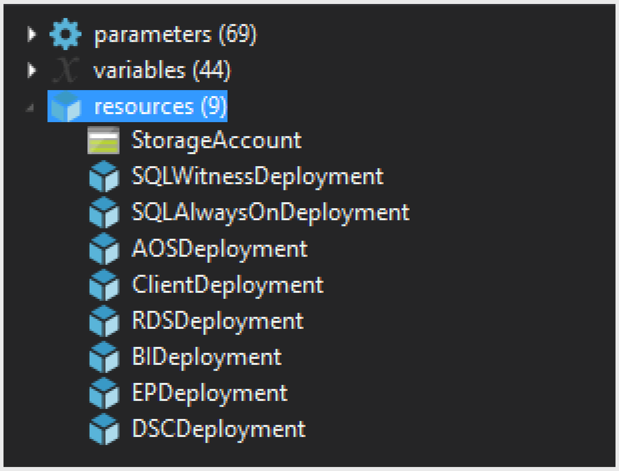
AX2012R3 Components

The AX2012R3 deployment consists in multiple components. Below are listed the components installed from the ARM Template. Any components not listed below are **not installed** by this template (Retail Components, Management Reporter, Data Import Export Framework, etc.).

- Database
- Application Object Server (AOS)
- Business intelligence components
    - Reporting Services extensions
    - Analysis Services configuration
- Client components
    - Client
- Web server components
    - Enterprise Portal (EP)
- Management Utilities

ARM Template Design

The template basically consists in three parts:



parameters (69)
variables (44)
resources (9)
StorageAccount
SQLWitnessDeployment
SQLAlwaysOnDeployment
AOSDeployment
ClientDeployment
RDSDeployment
BIDeployment
EPDeployment
DSCDeployment

1. **Parameters**

   Parameters are values that should vary according to each deployment. For example, the account names and passwords, the number of instances and the size of the VMs are parameters.

2. **Variables**

   Variables are values that don't vary for each deployment and sometimes they are a result of a function from other parameters and variables. Resources references, api versions and URLs for the powershell DSCs configurations are variables.

3. **Resources**

   The resources part is where the actual Azure resources to be deployed are defined. For a better organization of the template file, the main template *HighAvailabilityTemplate.json* has the storage account, other deployment resources, one for each resource group, and an additional one for the DSC extensions. These resources will be explained later in this document, one by one.

Parameters

Each deployment is different and those differences are defined by the parameters. Let's take a look on them one by one:

assetLocation
The root URL where ARM templates from this solution are accessible during the deployment. e.g. https://github.com/Microsoft/AX2012ARMTemplates/raw/master

accAOSServiceAccountName
The account that will run the AOS service. (format: DOMAIN\\username)

accAOSServiceAccountPassword
The password of the account that will run the AOS service.

accBusinessConnectorProxyAccountName
The account used by the Business Connector Proxy. (format: DOMAIN\\username)

accBusinessConnectorProxyAccountPassword
The password of the account used by the Business Connector Proxy.

accDynamicsInstallationAccountName
The account used to run the deployment tasks.

accDynamicsInstallationAccountPassword
The password of the account used to run the deployment tasks.

accLocalAdminAccountName
The local user account. (format: username)

accLocalAdminAccountPassword
The password of the local user account.

accSharepointServiceAccountName
The account that will run the Sharepoint Services. (format: DOMAIN\\username)

accSharepointServiceAccountPassword
The password of the account that will run the Sharepoint Services.

accSqlServerServiceAccountName
The account that will run the SQL Server service. (format: DOMAIN\\username)

accSqlServerServiceAccountPassword
The password of the accouont that will run the SQL Server service.

aosDnsNameForPublicIPPrefix
The prefix for the AOS public IPs' DNS names. It will be concatenated with the instance number. (e.g.: **aos0**.centralus.cloudapp.azure.com)

aosInstanceNamePrefix
The prefix for the AOS instances' names. It will be concatenated with the instance number.

aosLBDnsNameForPublicIP
The AOS load balancer public IP DNS name.

AOSNumberOfInstances
The number of AOS instances.

AOSVMNamePrefix
The prefix for the AOS VMs' names.

AOSVMSize
The size of the AOS VMs according to the supported values by Azure. (e.g.: Standard_D1, Standard_DS3, etc.)

aosWsdlPort
The AOS WSDL Port.

ApplicationSubnetName
The virtual network subnet where the VMs will be created.

biDnsNameForPublicIPPrefix
The prefix for the BI public IPs' DNS names. It will be concatenated with the instance number.

biLBDnsNameForPublicIP
The BI load balancer public IP DNS name.

BINumberOfInstances
The number of BI instances.

BIVMNamePrefix
The prefix for the BI VMs' names.

BIVMSize
The size of the BI VMs according to the supported values by Azure.

ClientDnsNameForPublicIPPrefix
The prefix for the Client public IPs' DNS names. It will be concatenated with the instance number.

ClientNumberOfInstances
The number of Client instances.

ClientVMNamePrefix
The prefix for the Client VMs' names.

ClientVMSize
The size of the Client VMs according to the supported values by Azure.

cumulativeUpdatesAndHotFixes
An array of URLs for the cumulative updates and hotfixes to be installed through the slip-stream process. Those URLs should point to a ZIP file that will be extracted in the Updates folder of the setup. (e.g.: [" **https://yourstorage.blob.core.windows.net/armdeploy/CU12.zip** "])

dbSqlServerDatabase
The name of the Dynamics database.

DiagnosticsStorageType
The type of the diagnostics storage account.

domainToJoin
The FQDN of the domain to join. (e.g.: armproject.onmicrosoft.com)

epDnsNameForPublicIP
The EP public IP's DNS name.

EPVMName
The EP VM's name.

EPVMSize
The size of the EP VM according to the supported values by Azure.

FileAxLicenseUri
The URL for the AX license file.

FileAxSetupPath
The **setup.exe** path relative to the extracted directory. For example, if the extracted path is "C:\Temp" and the setup file is "C:\Temp\CD\setup.exe" than this parameter value should be "CD\\setup.exe"

FileAxSetupUpdatesPath
The update folder used to store the cumulative updates and hot fixes that will be installed through the slip-stream process. Similar to the previous parameter, it is a relative path to the extracted directory. Following the previous example, the updates path would be "CD\\Updates".

FileAxSetupUri
The URL for the AX installation files. It should be a ZIP file.

FileMSChartSetupUri
The URL for the MS Chart Controls Setup file (.exe).

FileReportViewerSetupUri
The URL for the Report Viewer 2012 runtime setup file (.msi).

FileSharePointUpdateSetupUri
The URL for the SharePoint Server 2013 update file (.exe).

FileSqlADOMDSetupUri
The URL for the SQL Server 2008 ADOMD.NET setup file (.msi).

FileSqlAMOSetupUri
The URL for the SQL Server 2008 Analysis Management Objects setup file (.msi).

FileSqlServerNativeClientSetupUri
The URL for the SQL Server Native Client setup file (.msi).

FileSqlSharedManagementObjectsSetupUri
The URL for the SQL Server Shared Management Objects setup file (.msi).

FileSqlSysClrTypesSetupUri
The URL for the SQL Server System CLR Types setup file (.msi).

RDSCertificatePassword
The password for the RDS components certificate.

RDSDnsNameForPublicIPPrefix
The prefix for the RDS public IPs' DNS names. It will be concatenated with the instance number.

RDSFarmName
The name of the RDS servers farm.

rdsLBDnsNameForPublicIP
The RDS load balancer public IP DNS name.

RDSNumberOfInstances
The number of RDS instances.

RDSStaticPrivateIpPrefix
The prefix for the RDS servers static private IPs. It will be concatenated with the RDSStaticPrivateIpValue parameter. (e.g.: **10.2.0.**)

RDSStaticPrivateIpValue
The value for the right-most value in an IPV4 format. The private IPs will be assigned from this value with an increment of 1. For example, if this value is **16**, for 3 RDS servers we will have the following IP addresses: **10.2.0.16**, **10.2.0.17**, **10.2.0.18**.

RDSVMNamePrefix
The prefix for the RDS VMs' names.

RDSVMSize
The size of the RDS VMs according to the supported values by Azure.

spFarmPassphrase
The sharepoint farm passphrase.

SQLAlwaysOnAvailabilityGroupListenerName
The name of the SQL Server AlwaysOn availability group listener.

SQLAlwaysOnAvailabilityGroupName
The name of the SQL Server AlwaysOn availability group.

SQLAlwaysOnClusterName
The name of the SQL Server AlwaysOn cluster.

SQLAlwaysOnEndpointName
The name of the SQL Server AlwaysOn endpoint name.

sqlDnsNameForPublicIPPrefix
The prefix for the SQL Server public IPs' DNS names. It will be concatenated with the instance number.

SQLLoadBalancerPrivateIPAddress
The private IP address for the SQL Server load balancer.

SQLServerSubnetName
The virtual network subnet where the SQL Server VMs will be created.

SQLVMDataDiskSize

The size of data disks for the SQL Server VMs.

SQLVMNamePrefix

The prefix for the SQL Server VMs’ names.

SQLVMNumberOfDataDisk

The number of data disks for the SQL Server VMs.

SQLVMSize

The size of the SQL Server VMs according to the supported values by Azure.

sqlWitDnsNameForPublicIP

The file share witness server public IP’s DNS name.

SQLWitnessVMName

The file share witness server VM name.

SQLWitnessVMSize

The size of the file share witness server VM according to the supported values by Azure.

StorageType

The type of the Azure Storage Account according to the supported values by Azure. (e.g.: Premium_LRS). It’s recommended to use premium storage for better performance.

VirtualNetworkName

The name of the virtual network resource that already exists in the Azure subscription. This virtual network should be configured in order to allow the VMs to join the domain.

VirtualNetworkResourceGroup

The name of the resource group of the above parameter virtual network.

Variables

Variables can be used to store values that are result from some operations supported in ARM template or they can be used to contain static values. Below are the variables used in the template:

ApplicationSubnetRef
The reference for the application subnet.

assetLocation
The root URL for the ARM deployment template files.

deploymentApiVersion
The version of the API for the deployment resources.

Diagnostics**StorageName**
The name of the diagnostics storage account.

DSCModuleLocation
The root URL for the Powershell DSC modules.

publicIPAddressType
The type of the public IP addresses.

resourcesApiVersion
The version of the API for other resources.

SQLServerSubnetRef
The reference for the SQL Server subnet.

StorageName
The name of the storage account.

vhdStorageContainerName
The name of the VHDs container.

vnetId
The resource ID of the vritual network.

AOSTemplateURL
The URL for the AOS template. Combined with the assetLocation variable.

AOSVMDSCModuleUrl
The URL for the AOS Powershell DSC Module. Combined with the DSCModuleLocation variable.

AOSVMMachine0DSCConfigurationFunction
The Powershell DSC configuration to be set for AOS VM 0.

AOSVMMachine1NDSCConfigurationFunction
The Powershell DSC configuration to be set for AOS VM 1 to N.

BITemplateURL
The URL for the BI template. Combined with the assetLocation variable.

BIVMDSCConfigurationFunction
The Powershell DSC configuration to be set for BI VMs.

BIVMDSCModuleUrl
The URL for the BI Powershell DSC Module. Combined with the DSCModuleLocation variable.

ClientTemplateURL
The URL for the Client template. Combined with the assetLocation variable.

ClientVMDSCConfigurationFunction
The Powershell DSC configuration to be set for Client VMs.

ClientVMDSCModuleUrl
The URL for the Client Powershell DSC Module. Combined with the DSCModuleLocation variable.

DSCTemplateURL
The URL for the DSC template. Combined with the assetLocation variable.

DiskSelectionTemplateURL
The URL for the Disk Selection template. Combined with the assetLocation variable.

EPTemplateURL
The URL for the EP template. Combined with the assetLocation variable.

EPVMDataDiskSize
The size of the EP VM data disk.

EPVMDSCConfigurationFunction
The Powershell DSC configuration to be set for EP VM.

EPVMDSCModuleUrl

The URL for the EP Powershell DSC Module. Combined with the DSCModuleLocation variable.

RDSTemplateURL

The URL for the RDS template. Combined with the assetLocation variable.

RDSVMDSCModuleUrl

The URL for the RDS Powershell DSC Module. Combined with the DSCModuleLocation variable.

RDSVMMachine0DSCConfigurationFunction

The Powershell DSC configuration to be set for RDS VM 0.

RDSVMMachine1NDSCConfigurationFunction

The Powershell DSC configuration to be set for RDS VM 1 to N.

SQLAlwaysOnTemplateURL

The URL for the SQL Server AlwaysOn template. Combined with the assetLocation variable.

SQLVMCreateClusterDSCConfigurationFunction

The Powershell DSC configuration to be set for SQL VM 1.

SQLVMDSCModuleUrl

The URL for the RDS Powershell DSC Module. Combined with the DSCModuleLocation variable.

SQLVMPrepareDSCConfigurationFunction

The Powershell DSC configuration to be set for SQL VM 0.

SQLWitnessSharePath

The witness file share location

SQLWitnessTemplateURL

The URL for the file share witness template. Combined with the assetLocation variable.

SQLWitnessVMDataDiskSize

The size of the file share witness data disk.

SQLWitnessVMDSCConfigurationFunction

The Powershell DSC configuration to be set for file share witness VM.

SQLWitnessVMDSCModuleUrl

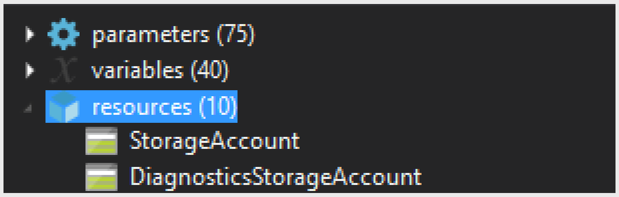
The URL for the file share witness Powershell DSC Module. Combined with the DSCModuleLocation variable.

Resources

In order to achieve a high availability environment, we need to avoid SPOF (single point of failure) that will block our customers operation. Thus, we need to deploy multiple instances for the same component and add extra resources such as load balancers to guarantee maximum availability of them. Let's break down the resources in groups:

Storage Account and Diagnostics Storage Account

An Azure Storage Account is provisioned to hold the VMs disks and another one for the diagnostics data.



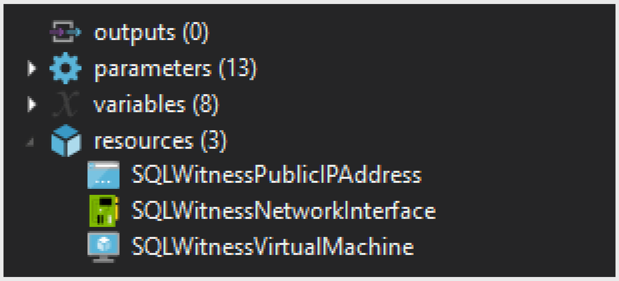
```
▸ parameters (75)
▸ variables (40)
◢ resources (10)
      StorageAccount
      DiagnosticsStorageAccount
```

Database

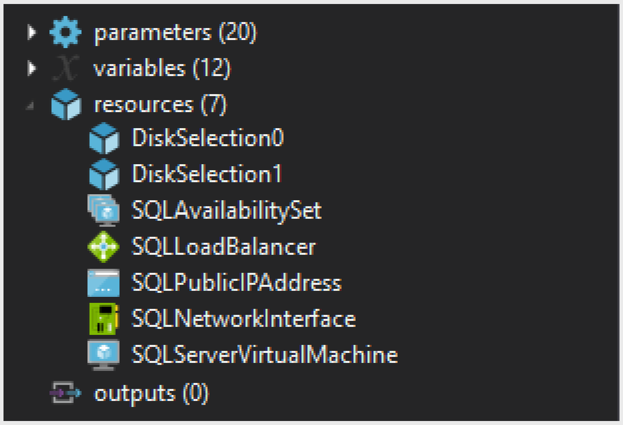
(SQLWitnessTemplate.json and SQLAlwaysOnTemplate.json)

We are using SQL Server AlwaysOn Availability Groups to maximize the availability of the database. The resources required for it are:

- Availability Set (guarantees that VMs will be provisioned in different hardware racks to avoid maintenance events to affect all your VMs at the same time).
- Internal Load Balancer (distributes incoming traffic among healthy instances)
- 2x Windows Server 2012R2 with SQL Server 2014 VMs for Always On Availability Groups
  - 1 OS disk for each VM
  - N data disks of customizable size for each VM
- 2x Public IP for the Database VMs
- 2x Network Interface Card (NIC) for the Database VMs
- Windows Server 2012R2 VM as a *File Share Witness* for Windows Server Cluster.
  - 1 OS disk for the VM
- 1x Public IP for the File Share Witness VM



```
  outputs (0)
▸ parameters (13)
▸ variables (8)
◢ resources (3)
      SQLWitnessPublicIPAddress
      SQLWitnessNetworkInterface
      SQLWitnessVirtualMachine
```



```
▸ parameters (20)
▸ variables (12)
◢ resources (7)
      DiskSelection0
      DiskSelection1
      SQLAvailabilitySet
      SQLLoadBalancer
      SQLPublicIPAddress
      SQLNetworkInterface
      SQLServerVirtualMachine
  outputs (0)
```

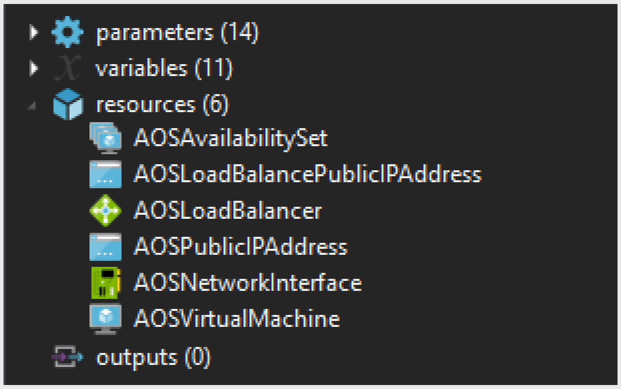
- 1x Network Interface Card (NIC) for the File Share Witness VM

Application Object Server (AOS)

(AOSTemplate.json)

For a high availability topology, we need multiple AOS instances backed up by a load balancer. The resources used are:

- Availability Set
- Load Balancer
- Public IP for Load Balancer
- N x Windows Server 2012R2 VMs
    - 1 OS disk for each VM
- N x Public IP for the AOS VMs
- N x NIC for the AOS VMs



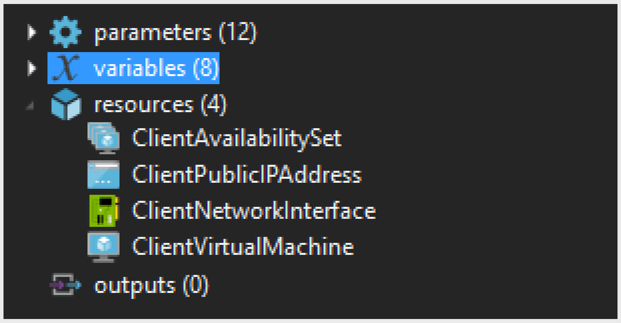
```
▸ parameters (14)
▸ variables (11)
◢ resources (6)
     AOSAvailabilitySet
     AOSLoadBalancePublicIPAddress
     AOSLoadBalancer
     AOSPublicIPAddress
     AOSNetworkInterface
     AOSVirtualMachine
  outputs (0)
```

Client

(ClientTemplate.json)

Same as AOSs, we need multiple VMs with Client installed. These VMs have also the role of **Remote Desktop Session Host (RDSH)** in order to be part of a Remote Desktop Session Deployment and be accessible through Remote Desktop Services.

- Availability Set
- N x Windows Server 2012R2 with Office 365 Professional
    - 1 OS disk for each VM
- N x Public IP for the Client VMs
- N x NIC for Client VMs



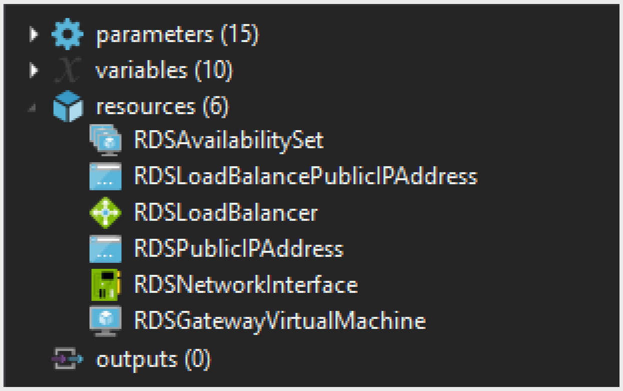
```
▸ parameters (12)
▸ variables (8)
◢ resources (4)
     ClientAvailabilitySet
     ClientPublicIPAddress
     ClientNetworkInterface
     ClientVirtualMachine
  outputs (0)
```

Remote Desktop Services

(RDSTemplate.json)

A High Availability Remote Desktop Services environment requires redundancy for all services part of Remote Desktop Services (Connection Broker, Gateway, WebAccess and Licensing). To achieve this configuration, multiple VMs are deployed for this purpose.

- Availability Set
- Load Balancer
- Public IP for Load Balancer
- N x Windows Server 2012R2 VMs
  - 1 OS disk for each VM
- N x Public IP for the RDS VMs
- N x NIC for the RDS VMs



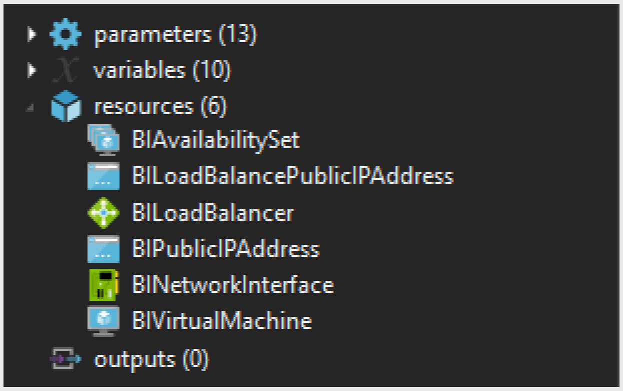
```
parameters (15)
variables (10)
resources (6)
    RDSAvailabilitySet
    RDSLoadBalancePublicIPAddress
    RDSLoadBalancer
    RDSPublicIPAddress
    RDSNetworkInterface
    RDSGatewayVirtualMachine
outputs (0)
```

Business Intelligence Components

(BITemplate.json)

To deploy AX SSRS reports, we need VMs with SQL Server and Reporting Services installed. For High Availability, multiple Report Servers need to be configured.

- Availability Set
- Load Balancer
- Public IP for Load Balancer
- N x Windows Server 2012R2 VMs
  - 1 OS disk for each VM
- N x Public IP for the BI VMs
- N x NIC for the BI VMs



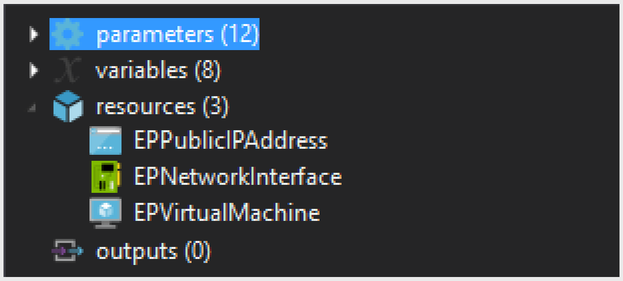
```
parameters (13)
variables (10)
resources (6)
    BIAvailabilitySet
    BILoadBalancePublicIPAddress
    BILoadBalancer
    BIPublicIPAddress
    BINetworkInterface
    BIVirtualMachine
outputs (0)
```

Enterprise Portal

(EPTemplate.json)

Following the existing deployment, Enterprise Portal is a single instance. Supporting a farm of Sharepoint Servers is out of the scope of this project. So, the resources required are:

- 1 x Windows Server 2012 with Sharepoint 2013
  - 1 OS disk for the VM
  - 1 data disk for the VM
- 1 x Public IP for the EP VM
- 1 x NIC for the EP VM



```
▸ parameters (12)
▸ variables (8)
◢ resources (3)
    EPPublicIPAddress
    EPNetworkInterface
    EPVirtualMachine
  outputs (0)
```

Powershell DSC Extensions

All that have been documented until here is related to the infrastructure required to run the High Availability environment for AX. Through Powershell DSC extensions, we can configure the VMs to the desired state, from domain join to ax components installation, all can be achieved through DSC resources. For more information about powershell DSCs, please refer to https://msdn.microsoft.com/en-us/powershell/dsc/overview.

The advantage of the powershell DSC scripts is that it is idempotent. That means that you can run the scripts multiple times and the result will always be the same.

The DSC resources are still being developed and new versions are released often. Most of them can be found at https://github.com/PowerShell. The xDynamicsAX2012R3 resources have been created for this project and it is available in the project files.

IMPORTANT: use the resources available inside the Powershell DSC module zip files available in the project. **Some of them have been modified from the original ones** or is using a specific version. In case you want to use the latest version of the resources, make sure that the parameters match and that the deploy works properly.

Let’s break down all the configurations for each VM:

File Share Witness VM

(DatabaseConfiguration.ps1\\SQLWitnessMachine)

For the configuration of the file share machine, we have to:

- Join domain (xComputer)
- Join machine admin group (Group)
- Wait for data disk to be available (xWaitforDisk)
- Assign disk a drive letter (cDiskNoRestart)
- Enable “Remote Active Directory Administration with Windows PowerShell” feature (WindowsFeature)
- Create file share directory (File)

- Set the file share directory as shared folder (xSmbShare)

SQL Server AlwaysOn First VM

(DatabaseConfiguration.ps1\\SQLAlwaysOnPrepare)

For the configuration of the SQL Server AlwaysOn machine, we have to:

- Wait for data disk to be available (xWaitforDisk)
- Assign disk a drive letter (cDiskNoRestart)
- Wait for log disk to be available (xWaitforDisk)
- Assign disk a drive letter (cDiskNorestart)
- Enable “Failover Clustering” feature (WindowsFeature)
- Enable “Failover Cluster Module for Windows Powershell” feature (WindowsFeature)
- Enable “Remote Active Directory Administration with Windows PowerShell” feature (WindowsFeature)
- Join domain and join admin group (CommonSetup)
- Enable firewall for database engine port 1433 (xFirewall)
- Enable firewall for database mirroring port 5022 (xFirewall)
- Enable firewall for availability group listener port 59999 (xFirewall)
- Add domain admin as SQL Server sysadmin (xSqlLogin)
- Add SQL Server service account as SQL Server sysadmin (xSqlLogin)
- Configure SQL Server with AlwaysOn (xSqlServer)

SQL Server AlwaysOn Second VM

(DatabaseConfiguration.ps1\\SQLAlwaysOnCreateCluster)

For the configuration of the second SQL Server AlwaysOn server, we have to:

- Enable CredSSP authentication as a server (xCredSSP)
- Enable CredSSP authentication a a client (xCredSSP)
- Wait for data disk to be available (xWaitforDisk)
- Assign disk a drive letter (cDiskNoRestart)
- Wait for log disk to be available (xWaitforDisk)
- Assign disk a drive letter (cDiskNoRestart)
- Enable “Failover Clustering” feature (WindowsFeature)
- Enable “Failover Cluster Module for Windows Powershell” feature (WindowsFeature)
- Enable “Remote Active Directory Administration with Windows PowerShell” feature (WindowsFeature)
- Join domain and join admin group (CommonSetup)

- Create windows server cluster (xCluster)
- Check for file share witness location (xWaitForFileShareWitness)
- Configure cluster quorum with witness file share (xClusterQuorum)
- Disable storage clustering (Script)
- Increase cluster timeout (Script)
- Enable firewall for database engine port 1433 (xFirewall)
- Enable firewall for database mirroring port 5022 (xFirewall)
- Enable firewall for availability group listener port 59999 (xFirewall)
- Add domain admin as SQL Server sysadmin (xSqlLogin)
- Add SQL Server service account as SQL Server sysadmin (xSqlLogin)
- Configure SQL Server with AlwaysOn on primary replica (xSqlServer)
- Configure SQL Server AlwaysOn endpoint on primary replica (xSqlEndpoint)
- Configure SQL Server with Always On on secondary replica (xSqlServer)
- Configure SQL Server AlwaysOn endpoint on secondary replica (xSqlEndpoint)
- Configure SQL Server AlwaysOn availability group (xSqlAvailabilityGroup)
- Configure SQL Server AlwaysOn availability group listener (xSqlAvailabilityGroupListener)
- Download AX setup file and extract ZIP (DownloadAXSetup)
- Download AX CU and hotfixes files and extract ZIPs (PrepareUpdatesAndHotFixes)
- Install Dynamics AX Database (xDynamicsDatabase)
- Set Dynamics database in availability group (xSqlNewAGDatabase)
- Test database in availability group (xSqlTestAGDatabase)

AOS First VM

(AOSConfiguration.ps1\\AOSMachine0)

For the configuration of the first AOS server, we have to:

- Join domain and join admin group (CommonSetup)
- Enable ".NET Framework 3.5" feature (WindowsFeature)
- Download AX setup file and extract ZIP (DownloadAXSetup)
- Download AX CU and hotfixes files and extract ZIPs (PrepareUpdatesAndHotFixes)
- Download and install MS Chart Controls (DownloadAndInstallMSChart)
- Download and install SQL Server System CLR Types (DownloadAndInstallSQLSysClrTypes)
- Download and install SQL Server Analysis Management Objects (DownloadAndInstallSQLAMO)
- Download and install SQL Server ADOMD.NET (DownloadAndInstallSQLADOMD)
- Download and install Report Viewer (DownloadAndInstallReportViewer)
- Download AX license file (DownloadAxLicenseFile)
- Install Dynamics AX AOS (xDynamicsAOS)
- Install Dynamics AX Client (xDynamicsClient)
- Initialize AX with license file (xDynamicsInitialization)

The license file needs to be applied to the system otherwise subsequent AOS instances won't start. To apply the license file, the client and its pre-requirements need to be installed.

AOS Other VMs

(AOSConfiguration.ps1\\AOSMachine1N)

For the configuration of the other AOS servers, we have to:

- Join domain and join admin group (CommonSetup)
- Enable ".NET Framework 3.5" feature (WindowsFeature)
- Download AX setup file and extract ZIP (DownloadAXSetup)
- Download AX CU and hotfixes files and extract ZIPs (PrepareUpdatesAndHotFixes)
- Download and install MS Chart Controls (DownloadAndInstallMSChart)
- Download and install SQL Server Analysis Management Objects (DownloadAndInstallSQLAMO)
- Download and install SQL Server ADOMD.NET (DownloadAndInstallSQLADOMD)
- Install Dynamics AX AOS (xDynamicsAOS)

Client VMs

(ClientConfiguration.ps1\\ClientMachine)

For the configuration of the Client VMs, we have to:

- Join domain and join admin group (CommonSetup)
- Enable ".NET Framework 3.5" feature (WindowsFeature)
- Enable "Remote Desktop Session Host Server" feature (WindowsFeature)
- Download AX setup file and extract ZIP (DownloadAXSetup)
- Download AX CU and hotfixes files and extract ZIPs (PrepareUpdatesAndHotFixes)
- Download and install MS Chart Controls (DownloadAndInstallMSChart)
- Download and install SQL Server System CLR Types (DownloadAndInstallSQLSysClrTypes)
- Download and install SQL Server Analysis Management Objects (DownloadAndInstallSQLAMO)
- Download and install SQL Server ADOMD.NET (DownloadAndInstallSQLADOMD)
- Download and install Report Viewer (DownloadAndInstallReportViewer)
- Install Dynamics AX Client (xDynamicsClient)

BI VMs

(BIConfiguration.ps1\\BIMachine)

For the configuration of the BI VMs, we have to:

- Join domain and join admin group (CommonSetup)
- Add domain admin as SQL Server admin (xSqlServerSysAdmin)
- Add NT SERVICE\ReportServer as SQL Server admin (xSqlServerSysAdmin)
- Enable CredSSP authentication as a server (xCredSSP)
- Enable CredSSP authentication a a client (xCredSSP)
- Enable firewall for reporting services (xSqlServerFirewall)
- Configure reporting services (xSqlServerRSConfig)
- Download AX setup file and extract ZIP (DownloadAXSetup)
- Download AX CU and hotfixes files and extract ZIPs (PrepareUpdatesAndHotFixes)
- Install Dynamics AX BI Components and deploy reports (xDynamicsReportingAndAnalysis)

RDS First VM

(RDSConfiguration.ps1\\RDSMachine0)

For the configuration of the RDS VM 0, we have to:

- Join domain and join admin group (CommonSetup)
- Enable Remote Desktop Services features (WindowsFeatures)
- Download and install SQL Server Native Client (DownloadAndInstallSqlServerNativeClient)
- Download and install SQL Server System CLR Types (DownloadAndInstallSQLSysClrTypes)
- Download and install SQL Server Shared Management Objects (DownloadAndInstallSQLSharedManagementObjects)
- Add machine as SQL Server sysadmin and dbcreator (xSqlServerLogin)
- Create Remote Desktop Session Deployment (xRDSessionDeployment)
- Configure Remote Desktop Services Web Access (xRDWebAccessConfiguration)
- Configure Remote Desktop Services Connection Broker High Availability (xRDConnectionBrokerHighAvailability)
- Configure Remote Desktop Services License (ConfigureRDSLicense)
- Configure Remote Desktop Services Gateway (ConfigureRDSGateway)
- Create Remote Desktop Services Session Collection (xRDSessionCollection)
- Publish Dynamics AX Remote App (xRDRemoteApp)
- Configure Remote Desktop Services Certificate (xRDCertificate)

RDS Other VMs

(RDSConfiguration.ps1\\RDSMachine1N)

For the configuration of the other RDS VMs, we have to:

- Join domain and join admin group (CommonSetup)
- Enable Remote Desktop Services features (WindowsFeatures)
- Download and install SQL Server Native Client (DownloadAndInstallSqlServerNativeClient)
- Download and install SQL Server System CLR Types (DownloadAndInstallSQLSysClrTypes)
- Download and install SQL Server Shared Management Objects (DownloadAndInstallSQLSharedManagementObjects)
- Add machine as SQL Server sysadmin and dbcreator (xSqlServerLogin)
- Add machine as part of the Connection Broker (xRDServer)
- Add machine as part of the Web Access (xRDServer)
- Configure Remote Desktop Services Web Access (xRDWebAccessConfiguration)
- Configure Remote Desktop Services Connection Broker High Availability (xRDConnectionBrokerHighAvailability)
- Configure Remote Desktop Services License (ConfigureRDSLicense)
- Configure Remote Desktop Services Gateway (ConfigureRDSGateway)

EP VM

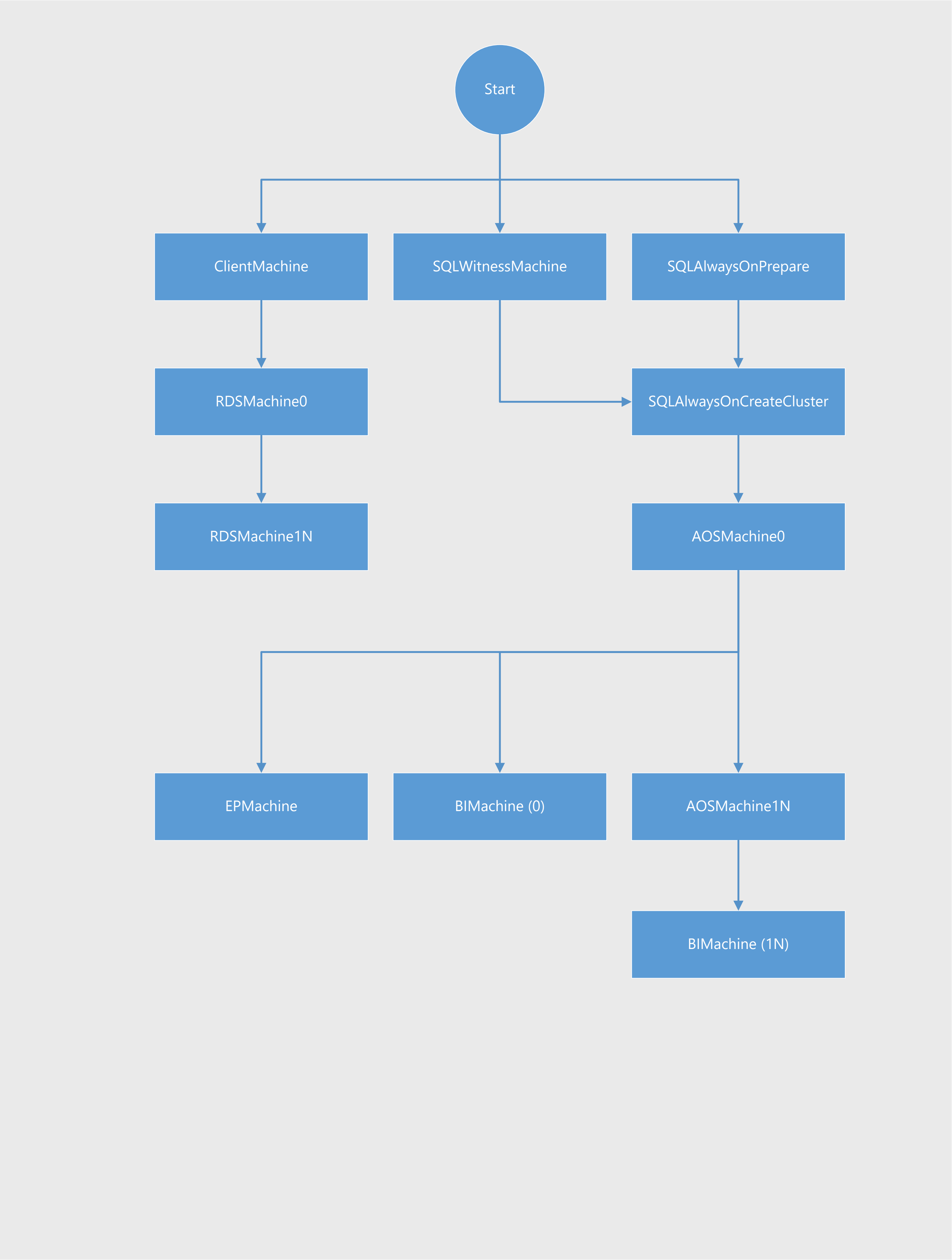
(EPConfiguration.ps1\\EPMachine)

For the configuration of the EP VM, we have to:

- Join domain and join admin group (CommonSetup)
- Download and install sharepoint update (DownloadAndInstallSharePointUpdate)
- Download and install MS Chart Controls (DownloadAndInstallMSChart)
- Download and install SQL Server System CLR Types (DownloadAndInstallSQLSysClrTypes)
- Download and install SQL Server Analysis Management Objects (DownloadAndInstallSQLAMO)
- Download and install Report Viewer (DownloadAndInstallReportViewer)
- Create SharePoint Farm (xSPCreateFarm)
- Enable Distributed Cache (xSPDistributedCacheService)
- Create SharePoint Web Application (xSPWebApplication)
- Create SharePoint Site (xSPSite)
- Configure claims to windows token service (xSPServiceInstance)
- Restart IIS (Script)
- Download AX setup file and extract ZIP (DownloadAXSetup)
- Download AX CU and hotfixes files and extract ZIPs (PrepareUpdatesAndHotFixes)
- Install Dynamics AX Enterprise Portal (xDynamicsEnterprisePortal)

Scripts Dependencies

To achieve a successful deployment of the AX environment, it is important to follow a specific order of the execution of the scripts. Changing such order will fail the deployment. The order is assured by the **dependendsOn** property in the Powershell DSC extension resources. The diagram below shows the deployment dependencies:


Start
ClientMachine
SQLWitnessMachine
SQLAlwaysOnPrepare
RDSMachine0
SQLAlwaysOnCreateCluster
RDSMachine1N
AOSMachine0
EPMachine
BIMachine (0)
AOSMachine1N
BIMachine (1N)

Template Execution

To execute the template and deploy an environment -

- In your azure subscription, create a blob storage that would be accessible during the template execution.
- Copy all files required by various parameters such as FileAxLicenseUri, FileAxSetupUri to the blob.
- Take the highavailabilitytemplate.parameter.json file in this solution and change the parameters accordingly for your deployment.
- Use the following PowerShell commands to execute the template.

```
Login-AzureRmAccount

Select-AzureRmSubscription -SubscriptionName YOUR-SUBSCRIPTION-NAME

New-AzureRmResourceGroup -Name AX2012R3ARMRG -Location EastUS -Verbose -Force -
ErrorAction Stop

New-AzureRmResourceGroupDeployment -Name AX2012R3ARMDeployment -ResourceGroupName
AX2012R3ARMRG -TemplateUri https://<YOUR RG>.blob.core.windows.net/<YOUR
BLOB>/HighAvailabilityTemplate.json -TemplateParameterFile
c:\armtest\HighAvailabilityTemplate.parameters.json -Force -Verbose
```

Troubleshooting Tips

As the ARM template and Desired Configuration Scripts orchestrate deployment of several components, you may ecounter an error during the execution. If you run the execution with the -Verbose option, you will be able to see which step of the deployment failed.

As the templates are idempotent by nature, you can re-execute the deployment, and it will try to redploy the components that had failed during the previous deployment.

Each VM deployed would have a folder located at **C:\Packages\Plugins\Microsoft.Powershell.DSC\2.26.0.0\Status** in wich you can find more logs about the DSC script execution.

In the event viewer, logs are located under **Applications and Services Logs > Microsoft > Windows > Descired Sate Configuration > Operational**