Abstract

If you are an enterprise who builds an application that processes credit card data, you need to conform to PCI DSS (Payment Card Industry Data Security Standard). Adherence to the standard means that you need to meet control objectives for your network, protect cardholder data, implement strong access controls, manage operations and more. To help customers to quickly standup infrastructure that conform to PCI DSS, we are releasing an Azure Quick Start sample. The template describes a stack that deploys a multi-tiered azure PaaS web application stack. It makes use of many nested templates, and can be customized as desired.

AZURE PAAS - MANAGEMENT & PCI COMPLIANT SOLUTION DEPLOYMENT GUIDE

Contents

[1 High level summary 2](#_Toc475080703)

[2 Pre-Requisites & Permissions required to deploy 2](#_Toc475080704)

[☛ Local machine setup 2](#_Toc475080705)

[☛ Application stack should be configured by 2](#_Toc475080706)

[3 Pre-deployment Steps 2](#_Toc475080707)

[3.1 Manual creation of Azure Automation Account 3](#_Toc475080708)

[3.2 Run Pre-Deployment PowerShell Script 3](#_Toc475080709)

[3.3 Configure AD App 5](#_Toc475080710)

[4 Deployment steps 10](#_Toc475080711)

[4.1 Click Deploy to Azure on GitHub 10](#_Toc475080712)

[4.2 Deployment Timeline 12](#_Toc475080713)

[5 Post Deployment Steps 12](#_Toc475080714)

[5.1 Update 1&1 DNS setting with Application Gateway IP 12](#_Toc475080715)

[5.2 Run Post Deployment PowerShell Script 14](#_Toc475080716)

[5.3 Schedule Runbooks 17](#_Toc475080717)

[5.4 Install OMS Dashboards Views 17](#_Toc475080718)

[5.5 Check and verify OMS solutions are collecting data 20](#_Toc475080719)

[5.6 Enable Azure Security Center 20](#_Toc475080720)

[5.7 Enable Tinfoil Security for web app vulnerability assessment 21](#_Toc475080721)

# High level summary

# Pre-Requisites & Permissions required to deploy

|  |
| --- |
| ☛ **Local machine setup**  * Run the pre-and post deployment scripts with Admin or remote-signed credentials (run as Administrator will work fine) * Install Azure PowerShell. If you don’t know how to install, refer the <https://docs.microsoft.com/en-us/powershell/azureps-cmdlets-docs/> link * Install Azure AD PowerShell. Refer the <https://technet.microsoft.com/en-us/library/dn975125.aspx> link. |
| ☛ **Application stack should be configured by**  * AD Global Admin (if you don’t know what that is, refer the <https://docs.microsoft.com/en-us/azure/active-directory/active-directory-assign-admin-roles#global-administrator> link) * Azure Subscription Role (either of the following roles)   + Service Administrator – refer <https://docs.microsoft.com/en-us/azure/billing-add-change-azure-subscription-administrator>   + Co-administrator - <https://docs.microsoft.com/en-us/azure/billing-add-change-azure-subscription-administrator>   + Subscription Owner - <https://docs.microsoft.com/en-us/azure/billing-add-change-azure-subscription-administrator> |

# Pre-deployment Steps

## Manual creation of Azure Automation Account

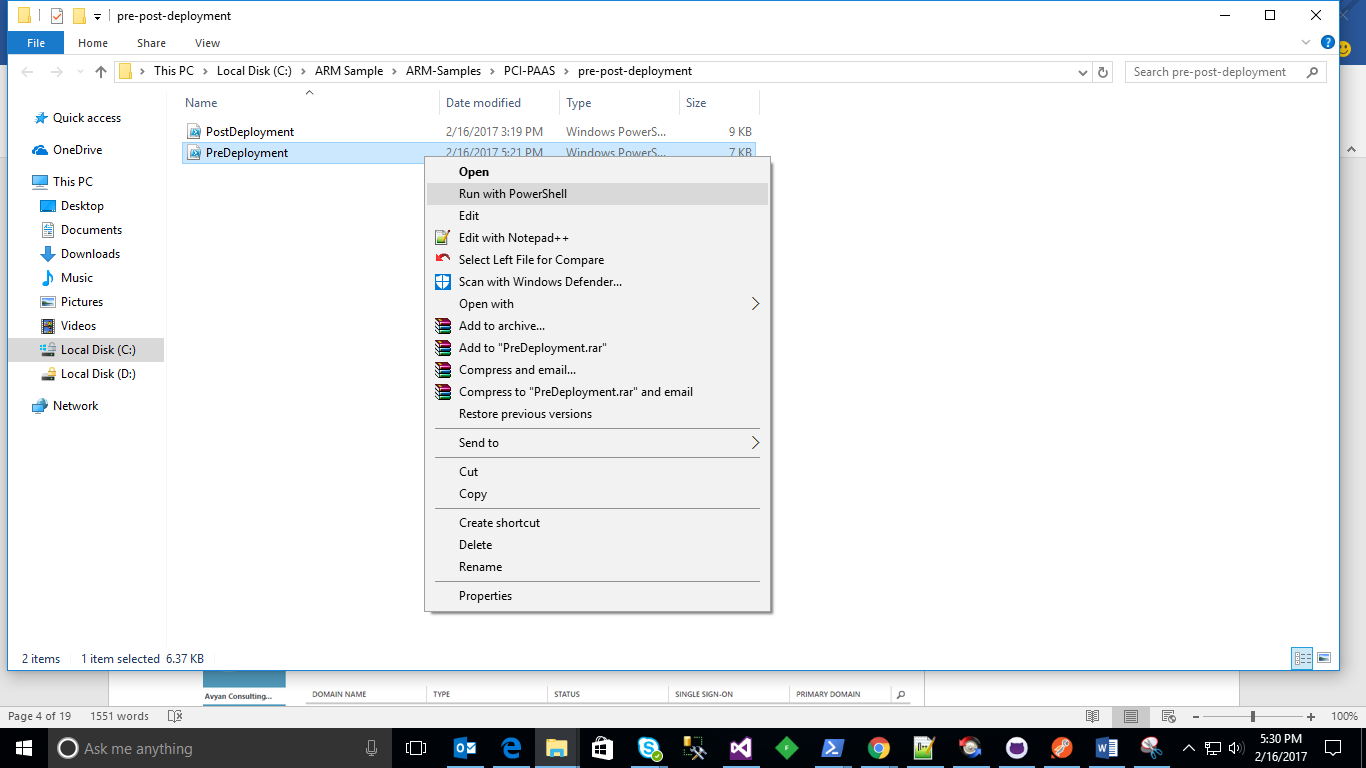
Create an Automation account with **Run As Service principal option**. Unfortunately, ARM templates don't allow for creating AD service principals,hence this step is currently a manual one.

* Refer the blog <https://docs.microsoft.com/en-us/azure/automation/automation-sec-configure-azure-runas-account> for the steps.
* Creation of Service Principal has a propensity to fail randomly. A basic verification whether it was successfully created is **mandatory.**

Note: Save the name of the automation account. You will be using that as a parameter to the ARM template.

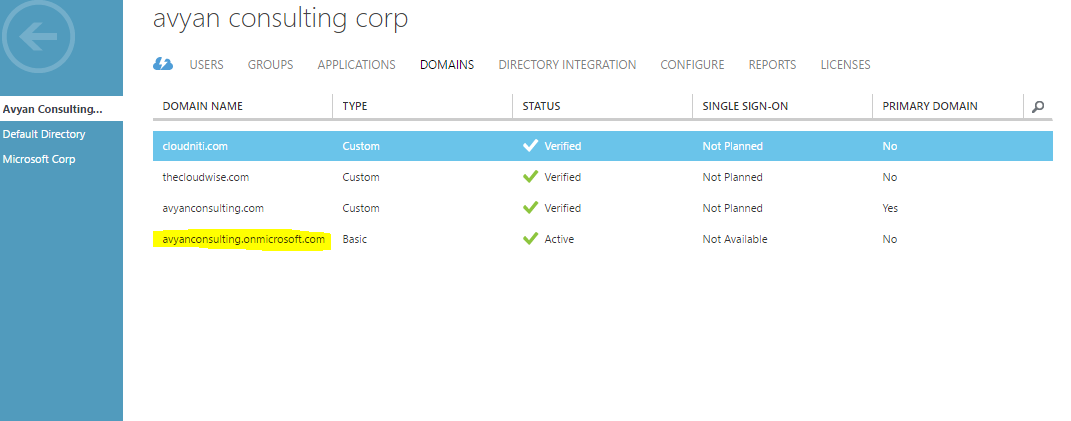
## Run Pre-Deployment PowerShell Script

* Open PowerShell file (PreDeployment.ps1). Right-click and select the “Run with PowerShell” option.

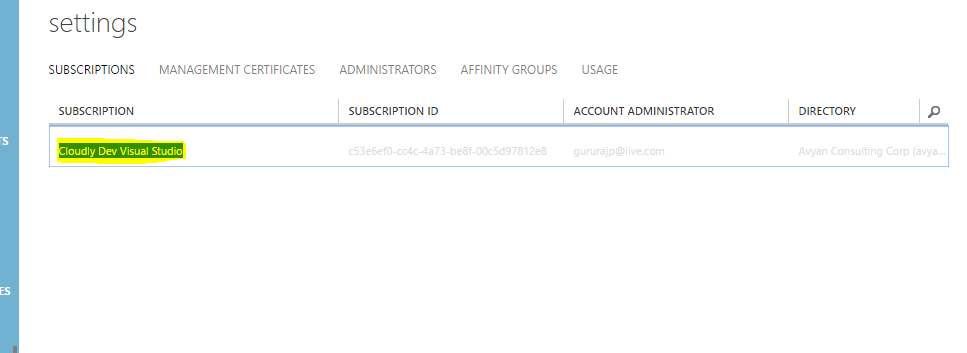


* It will ask for the following mandatory values while run script.

|  |  |  |
| --- | --- | --- |
| Parameter Name | How to get Value? | Comments |
| $azureADDomainName | * Login to <https://manage.windowsazure.com> * Open Domains under Active directory | Screenshots attached below |
| $subscriptionName | * Login to <https://manage.windowsazure.com> * Open Settings in left pane | Screenshots attached below |
| $suffix | * Provide unique website name | This is used to create a unique website name in your organization.  This could be your company name or business unit name  Some examples:   * “MSFT” * “ITCOE” * “ITAppDev” |

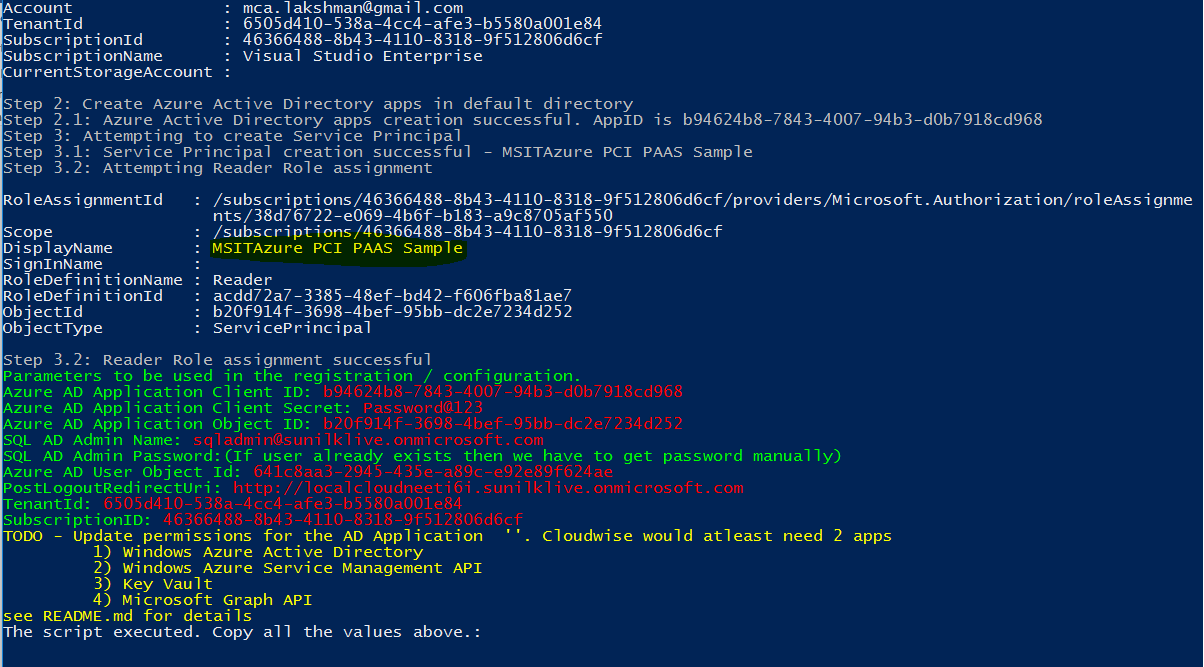


Screenshot: Get Domain Name



Screenshot: Get Subscription Name

* After deployment, copy the following valuesto pass in ARM Deployment.



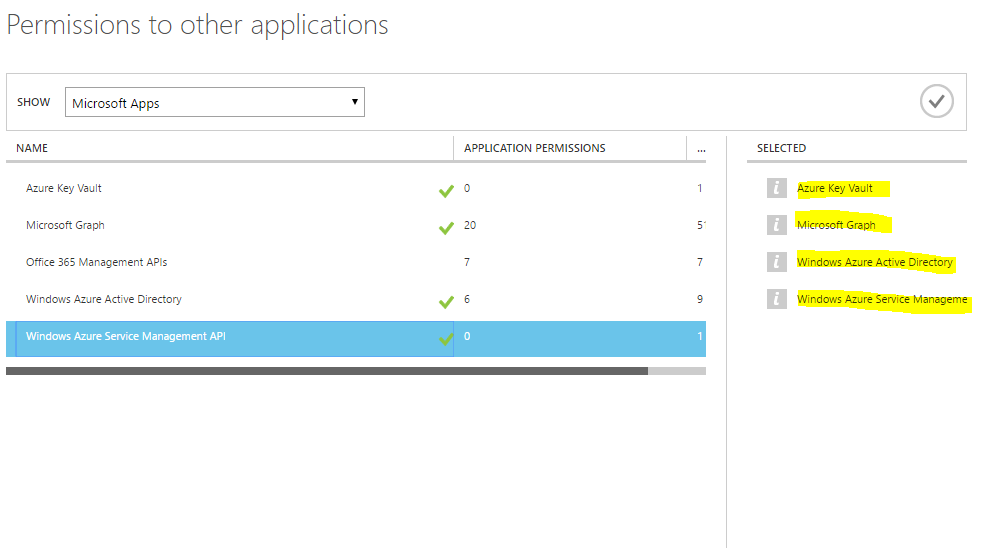
## Configure AD App

* 1. In the Azure portal, search for the Azure Active directory. Open the **Applications** tab.
  2. Open the AD Application that you just created. It should start with the name ($suffix + “Azure PCI PAAS Sample” )

$suffix is whatever you used during pre-deployment script.

**Note**: Highlighted application name in yellow color in above screenshot.

* 1. Click **Configure** to configure the following permissions in the **Required Permissions** tab.
  2. Click **Add Application** at bottom of the page and add the following applications and give permission.



Following are the list ofpPermissions.

|  |  |
| --- | --- |
| **Category** | **Permission** |

|  |  |
| --- | --- |
| **Windows**  **Service**  **Management**  **API** |  |
| **Microsoft.Azure. ActiveDirectory** |  |
| **Azure Key Vault** |  |
| **Microsoft Graph API** | **Application Permissions** |

|  |  |
| --- | --- |
|  | Delegated Permissions |

|  |  |
| --- | --- |
|  |  |

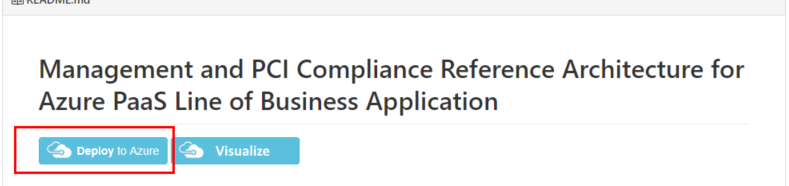


# Deployment steps

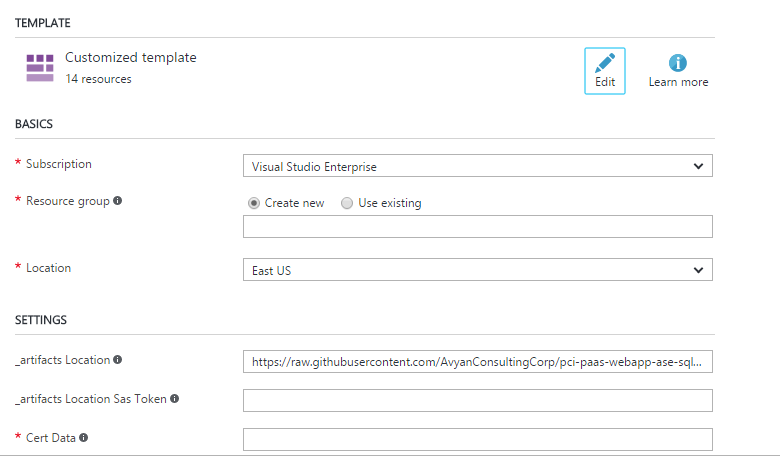
## Click Deploy to Azure on GitHub

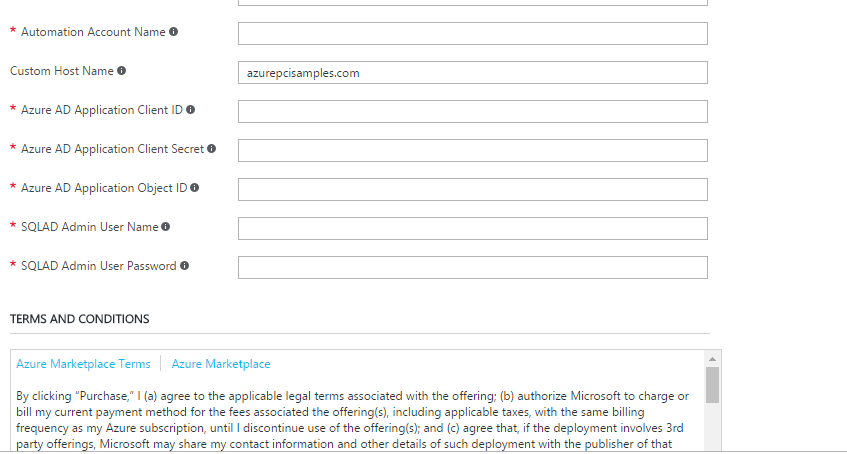
Temporary location:

<https://github.com/AvyanConsultingCorp/pci-paas-webapp-ase-sqldb-appgateway-keyvault-oms>

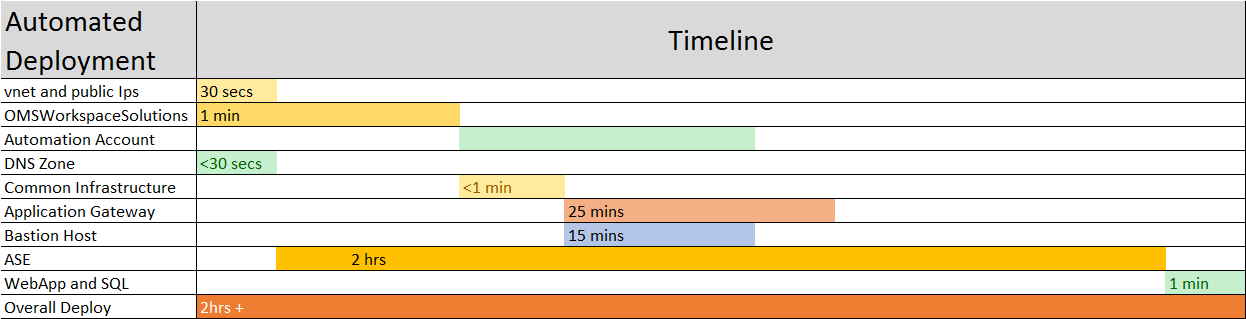


Provide all the mandatory values as mentioned on the following screenshot and click Purchase.





## Deployment Timeline

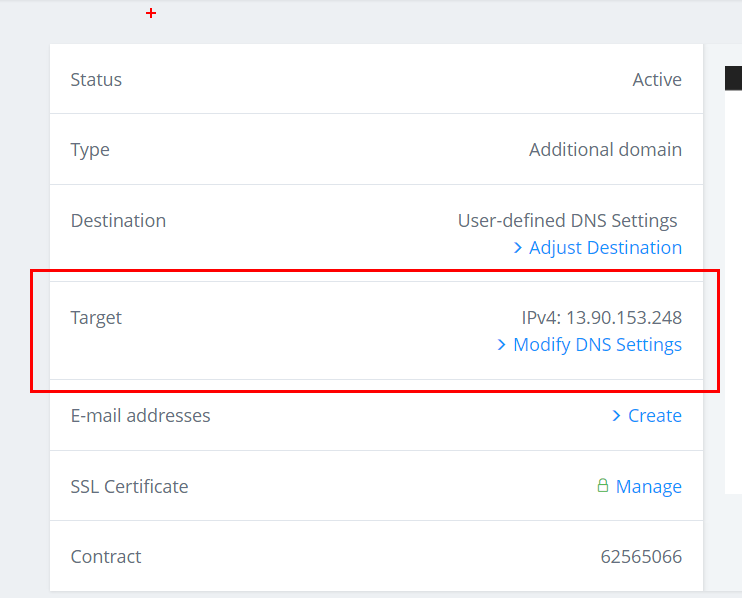


# Post Deployment Steps

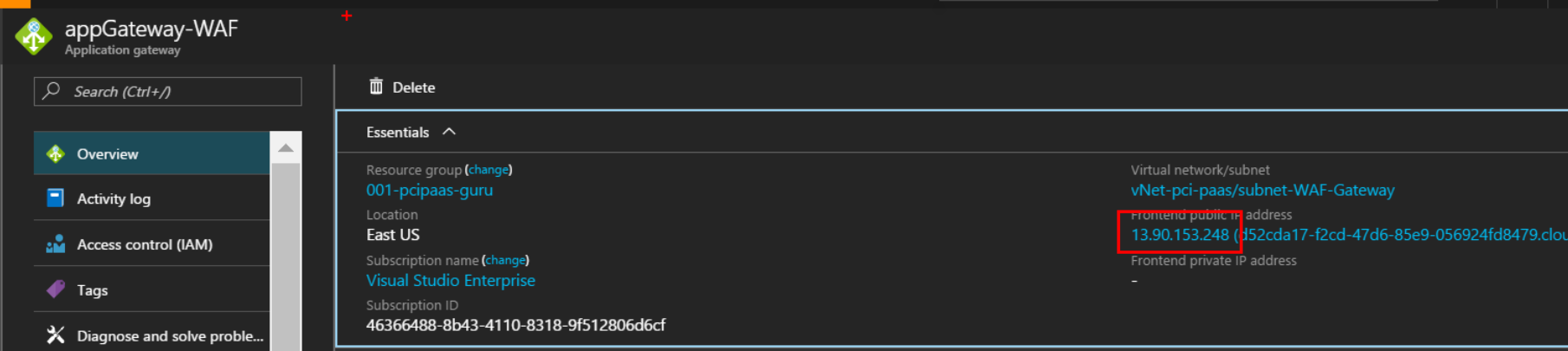
## Update 1&1 DNS setting with Application Gateway IP

We are using 1&1 for managing DNS. It might be different for you.

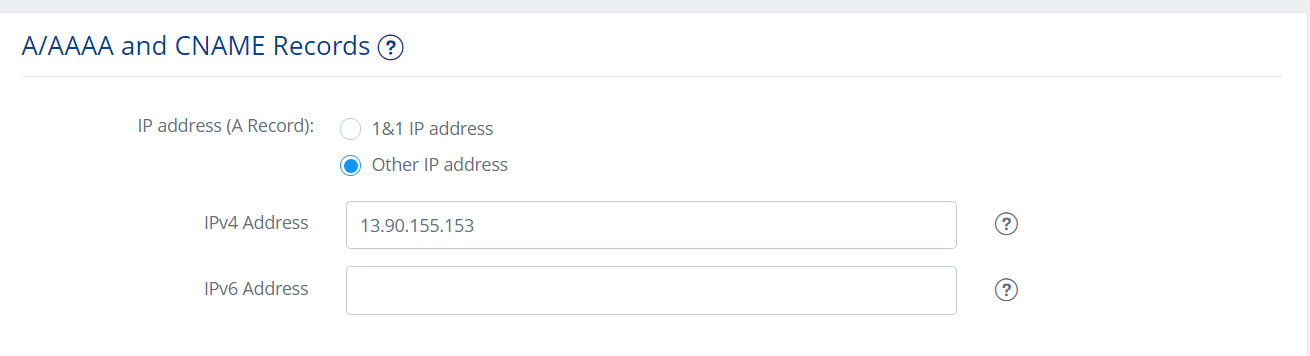
1. Modify the DNS settings under the **Target** settings.



1. Note the public IP address of App gateway. Go to the Application gateway object (appGateway-WAF) and checkout the IP Address.

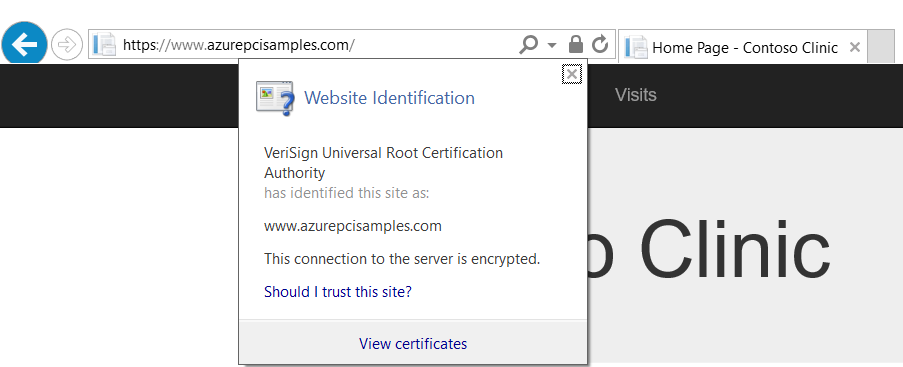


1. Update the A record IP address to be the App Gateway address.



***Verification***

Site is working with <https://www.azurepcisamples.com>



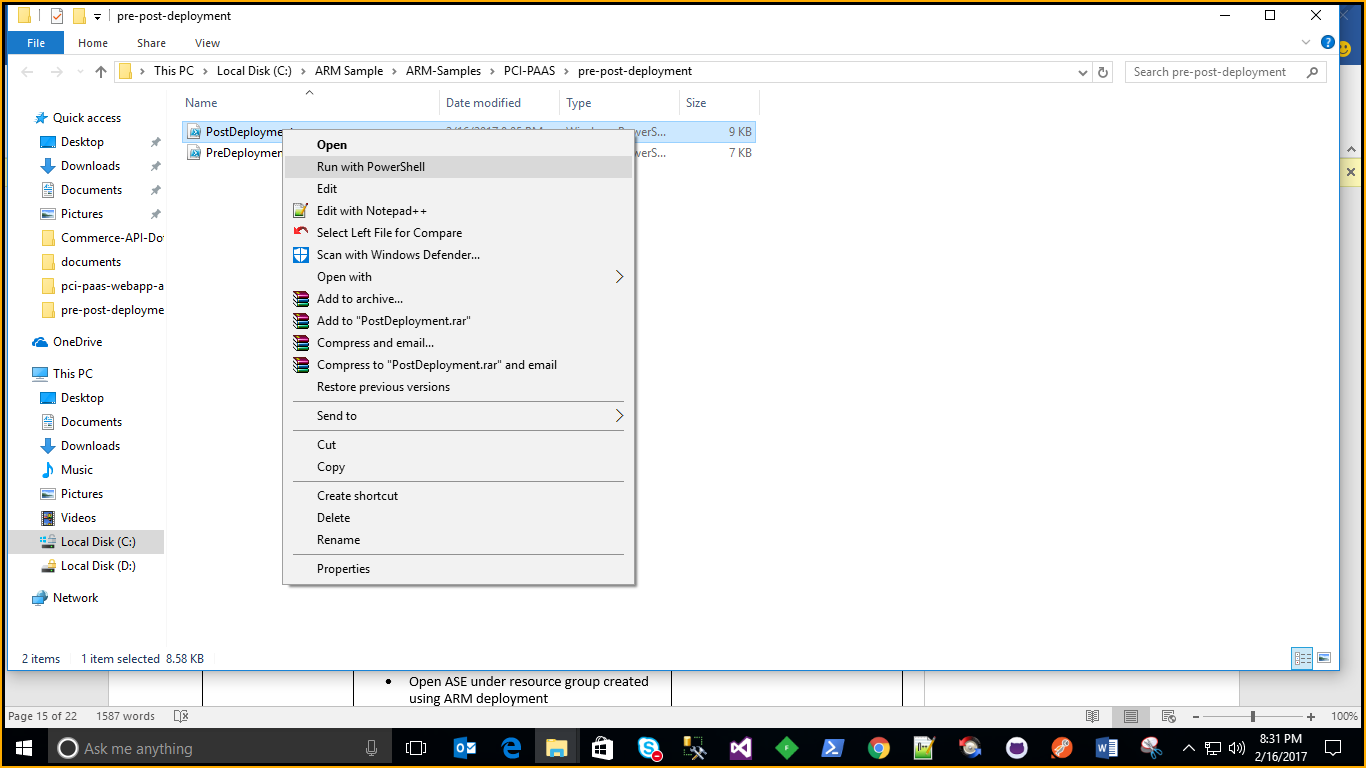
## Run Post Deployment PowerShell Script

Post Deployment PowerShell script used for the following configuration

* Set Firewall rules for ASE Outbound and Client IP Address(To run scripts)
* Restore bacpac file into ContosoClinicDB DB
* Configure Dynamic Data Masking in Patients table
* Encrypt Columns using Key vault
* Set AD Authentication Admin
* Enable OMS Diagnostics

**Process to run Post Deployment Script**

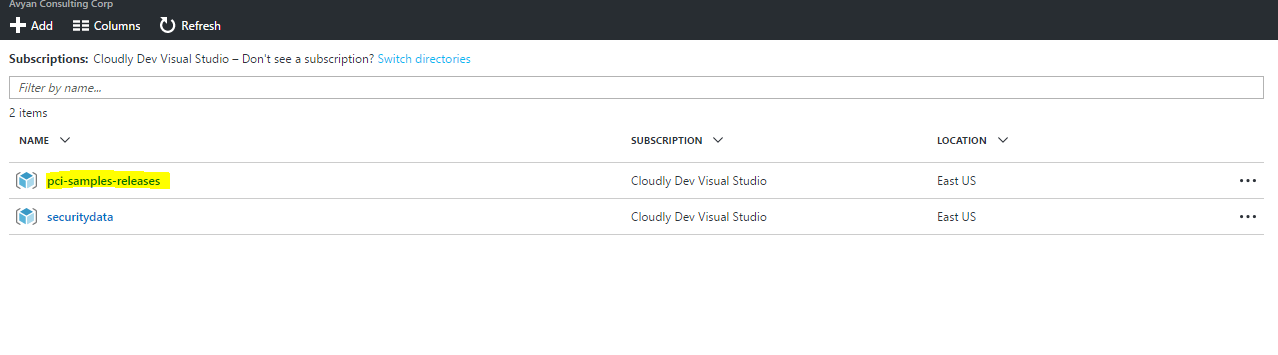
* Open Post Deployment PowerShell script (Path: ~/ pre-post-deployment/ PostDeployment.ps1) and Right-click and select **Run with PowerShell**.



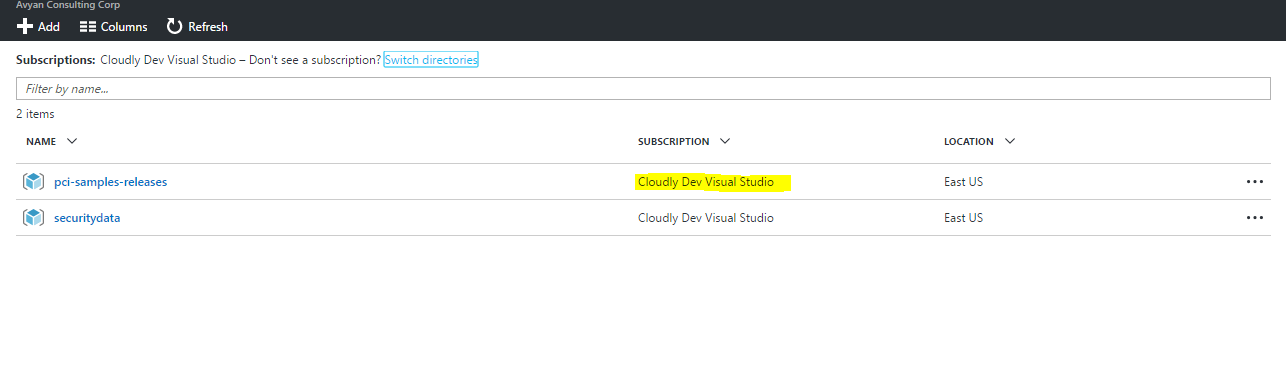
* Provide below mandatory values while run script. It will ask Subscription credentials to execute complete script

|  |  |  |
| --- | --- | --- |
| **Parameter Name** | **How to get Value?** | **Comments** |
| $ResourceGroupName | * Login to <https://portal.azure.com> * Open Resource groups and copy resource group created using ARM deployment | Screenshots attached below. Provide Resource Group Name Created through ARM template |
| $SQLServerName | * Login to <https://portal.azure.com> * Copy Sql Server name under resource group created using ARM deployment | Provide Sql Server name (not required full name) Created through ARM template |
| $sqlPassword | * Provide SQL Password passed while creating ARM template | This value passed as input parameter in ARM Deployment |
| $ClientIPAddress | * Open windows cmd prompt and type ipconfig | Your system IP address |
| $ASEOutboundAddress | * Login to <https://portal.azure.com> * Open ASE under resource group created using ARM deployment * Open Properties and copy Outbound IP address |  |
| $SQLADAdministrator | * Provide SQL AD Administrator name passed while creating ARM template | This value passed as input parameter in ARM Deployment |
| $subscriptionName | * Login to <https://portal.azure.com> * Open Resource groups and copy subscription name of resource group created using ARM deployment | Screenshots attached below |
| $KeyVaultName | * Login to <https://portal.azure.com> * Copy Key Vault name under resource group created using ARM deployment | Provide Key Vault Name Created through ARM template |

(for to)



Screenshot: Get Resource Name



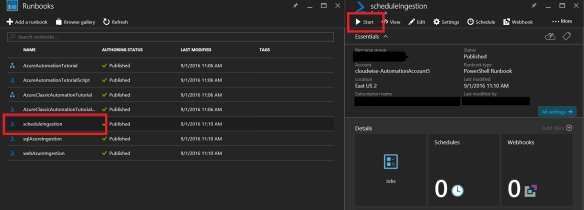
Screenshot: Get Subscription Name

Here are some sample values we got from one of our deployment runs



## Schedule Runbooks

Click open the **Schedule Ingestion** runbook and click start to run the runbook. This step will kick start the data ingestion to the OMS workspace specified.

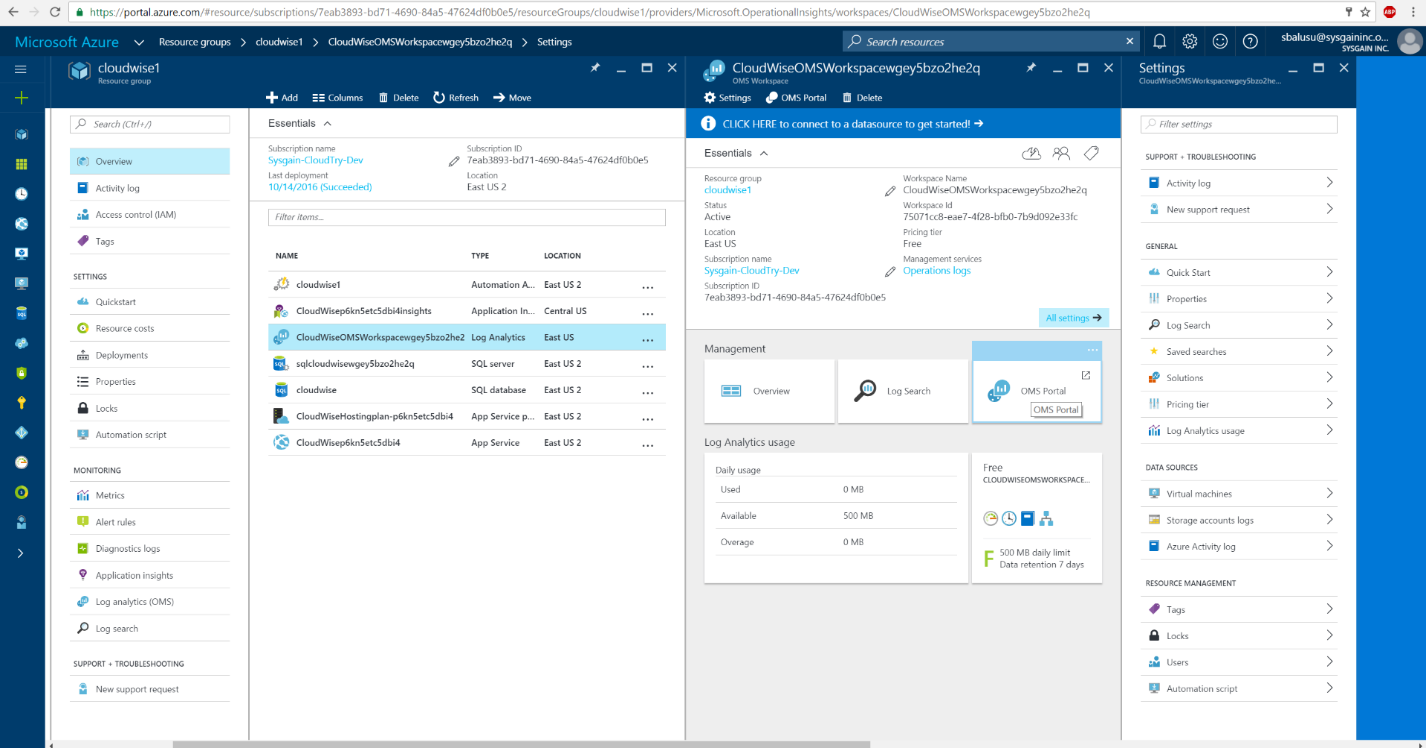


## Install OMS Dashboards Views

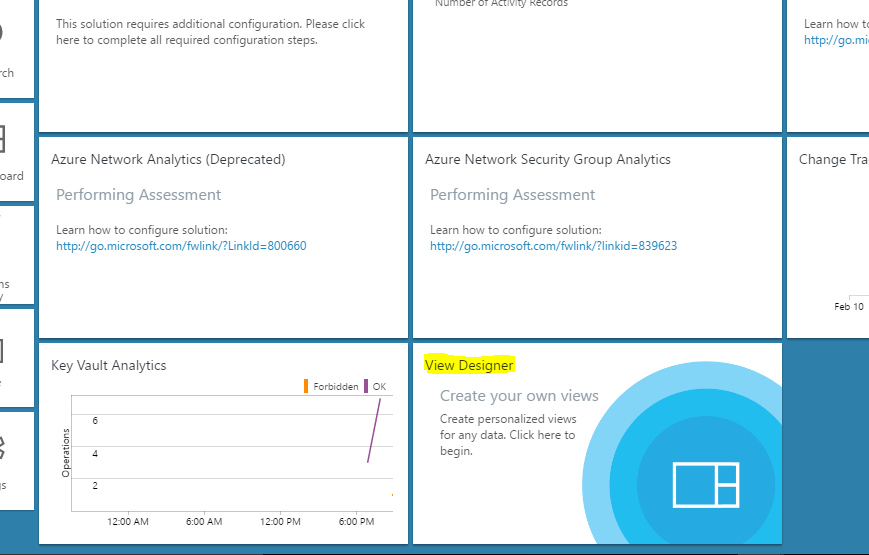
This is currently a manual process as ARM JSON deploys do not yet support creation of OMS views.

(By a Service Admin/Contributor role)

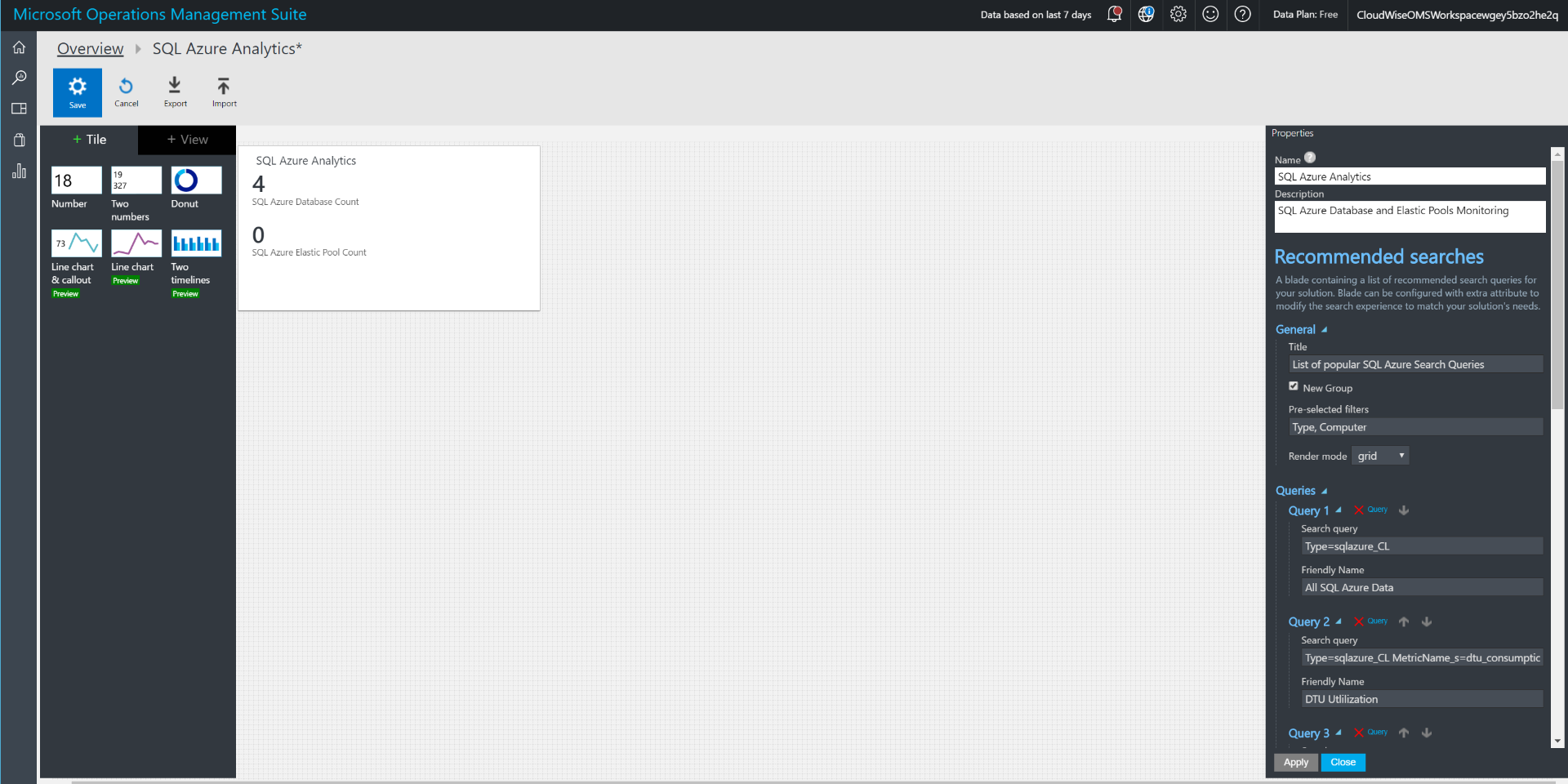
1. Open the resource group and click the **OMS Portal** link. This will open the OMS portal in a different window.

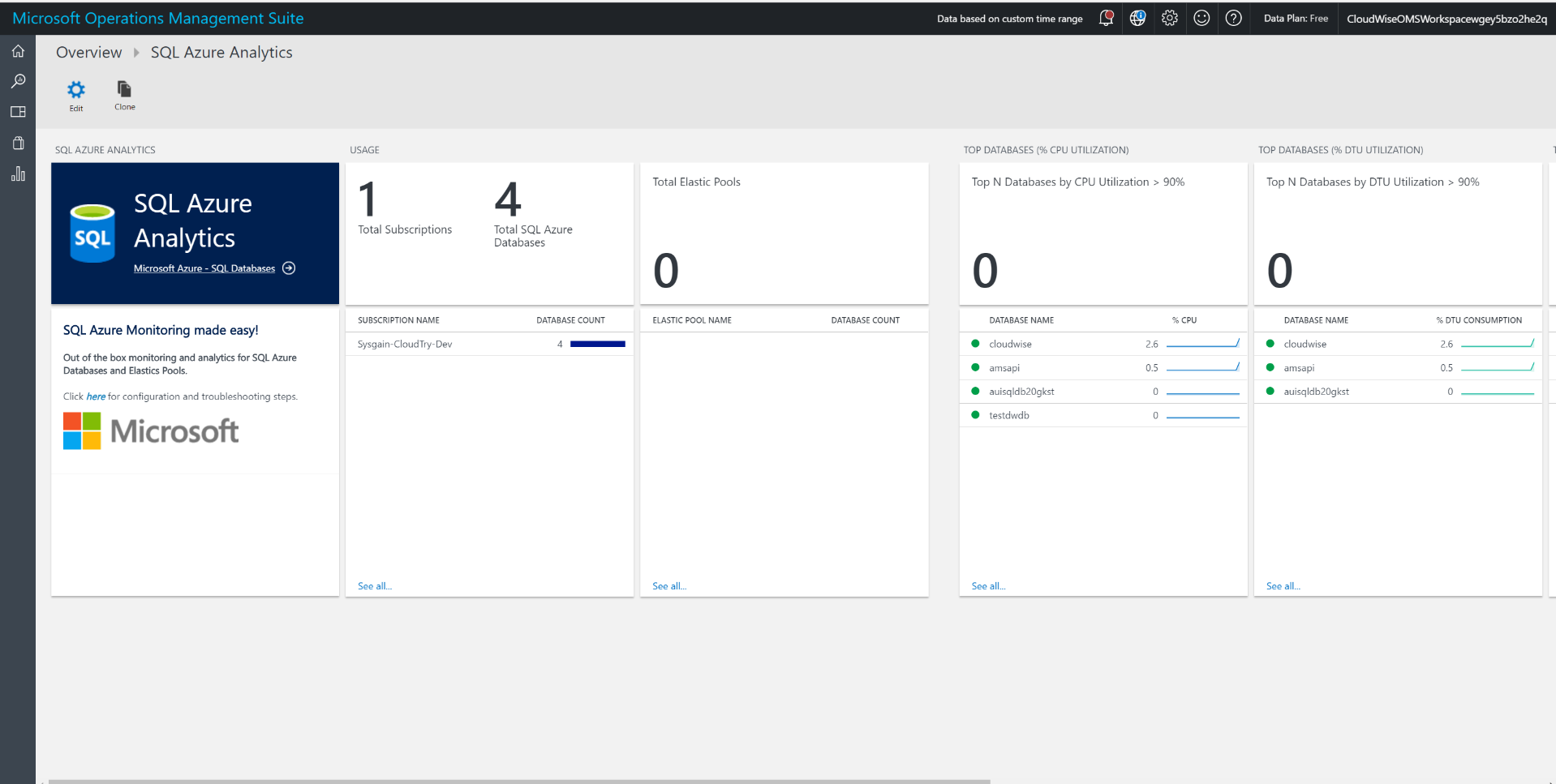


1. Click the **View Designer**. You may have to scroll down.

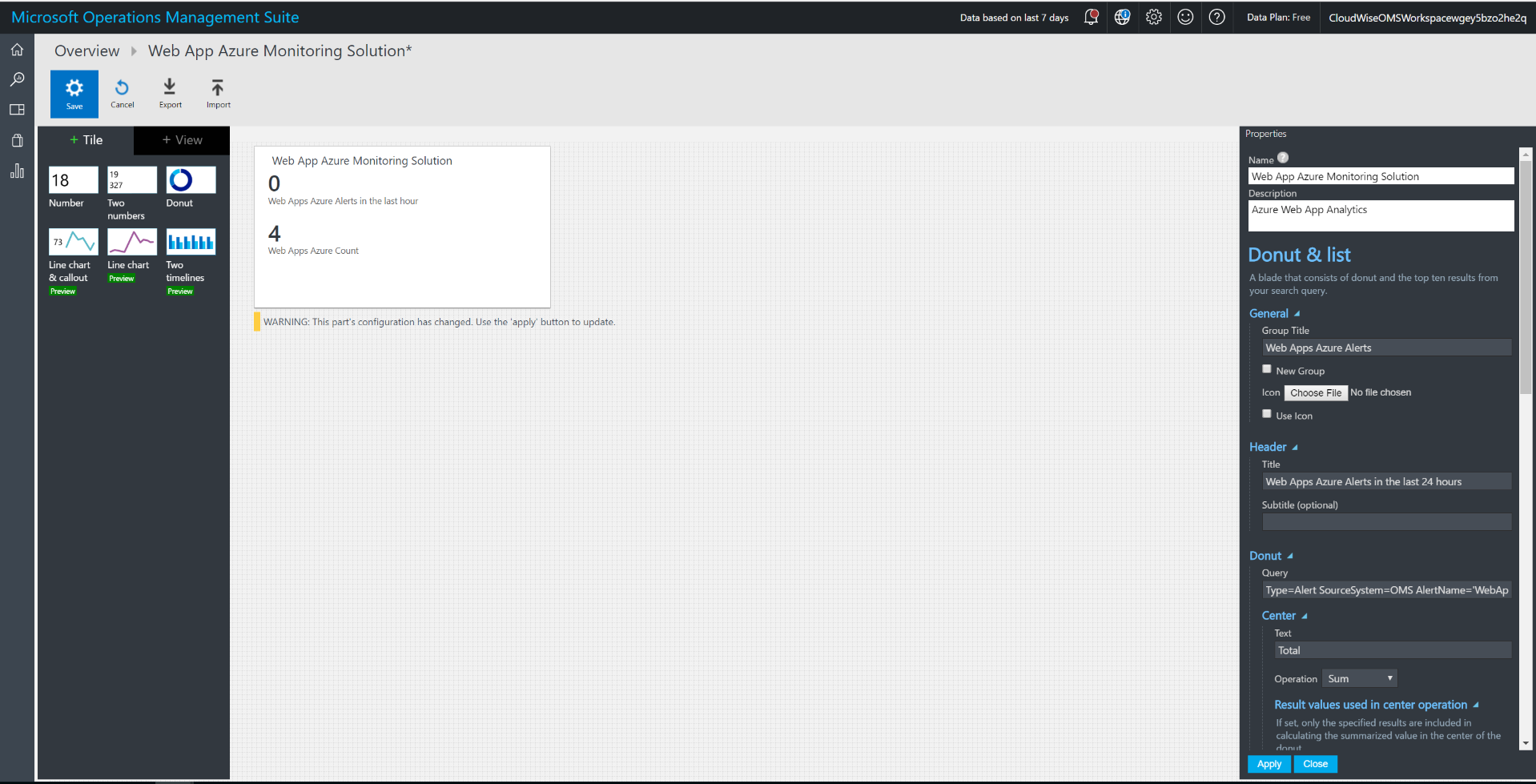


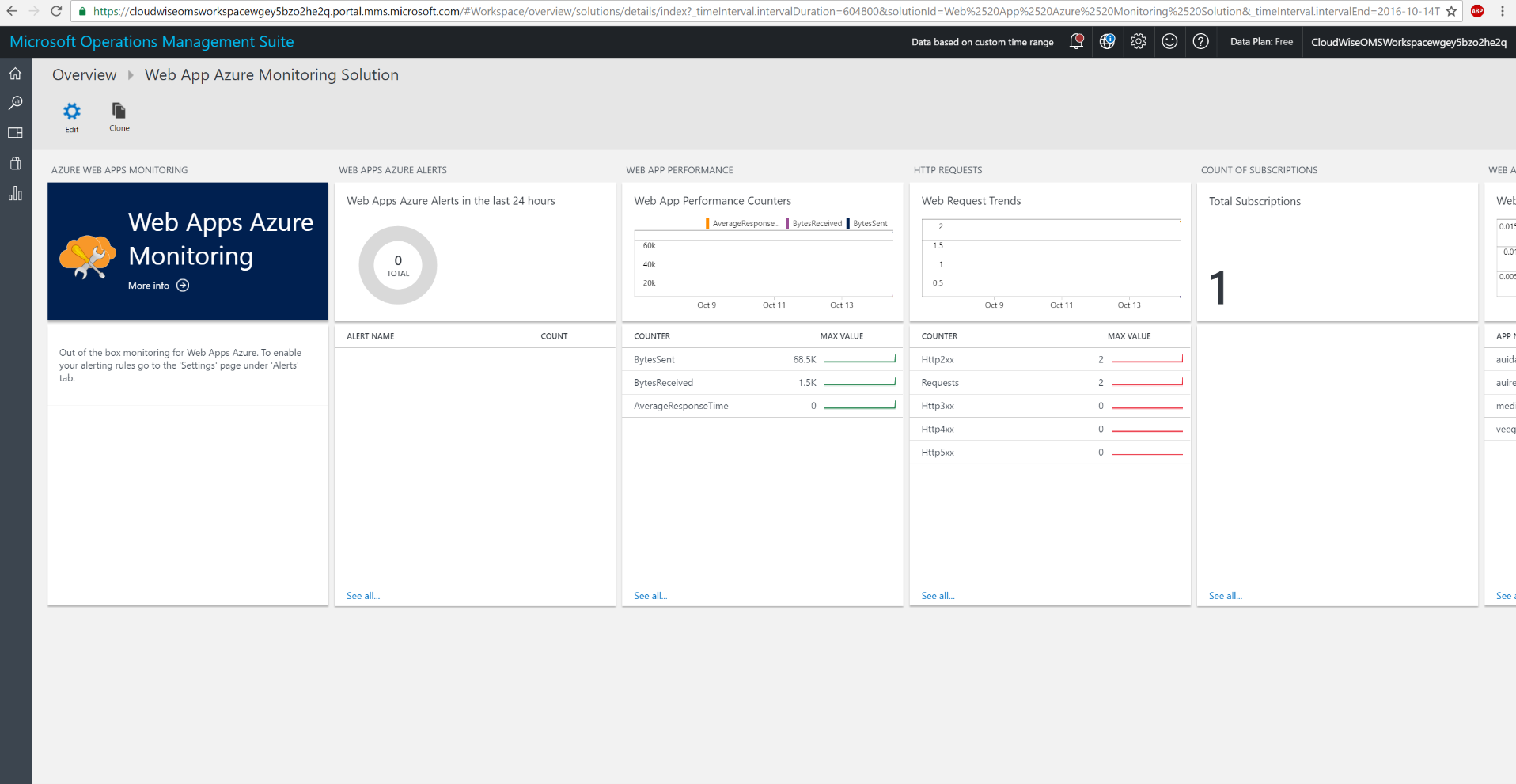
1. Import **SQL DB** view by clicking the **Import** button and browsing to the file (omsDashboards\OMSSQLDBAzureMonitoringSolution.omsview).





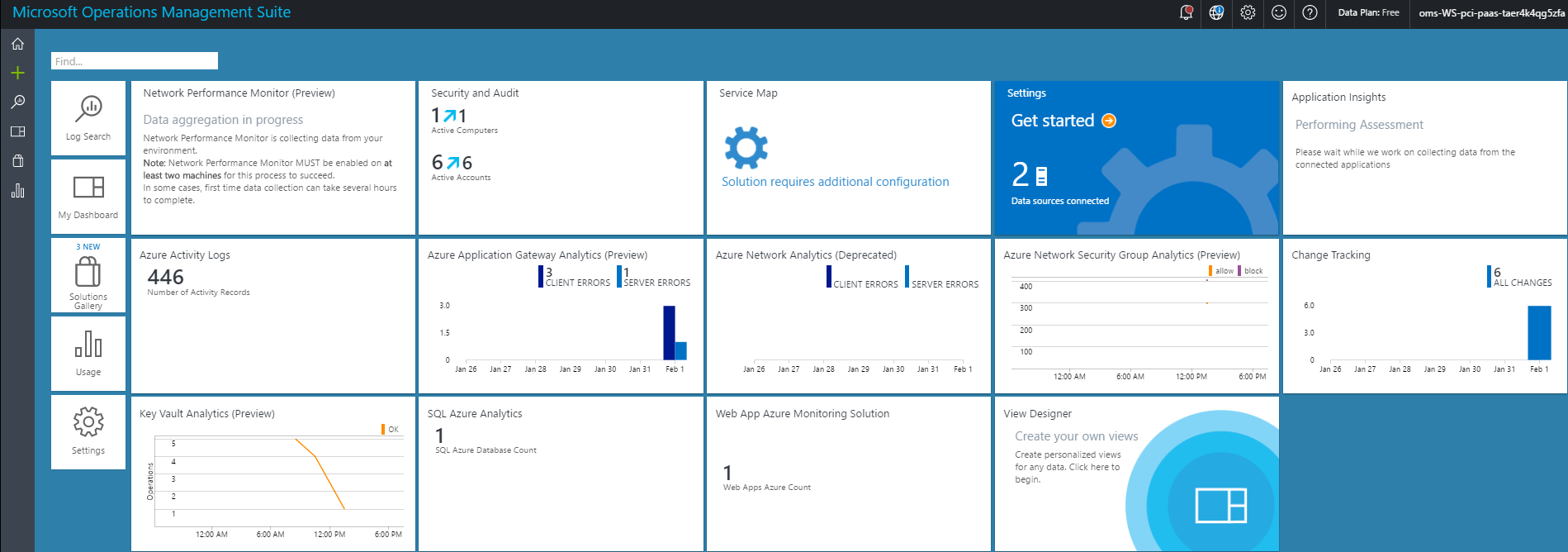
1. Repeat the same step for the **Web App Monitoring** dashboard. Import view omsDashboards\ OMSWebAppAzureMonitoringSolution.omsview.





## Check and verify OMS solutions are collecting data

Login to the OMS workspace installed in the resource group. Then, navigate to the portal and you’ll see something like this. Data is being collected and Solution is being monitored and operationally ready for you to test out.



## Enable Azure Security Center

Follow the instructions from here to enable data collections from Azure Security Center <https://docs.microsoft.com/en-us/azure/security-center/security-center-get-started>

To Do

1. Enable Data collection at the Subscription level.
2. Ensure all rules/policies are enabled (they are enabled by default)
3. Run recommendations
4. View the recommendation results. You will see that the solutions pass most of the rules:
   * WAF enabled for public IP address
   * VM disk and data disk encrypted
   * All Azure Storages accounts are encrypted
   * SQL TDE, Auditing enabled
   * And many more…

☛ **Note:**

Currently OMS Monitoring agent is automatically installed along with the Bastion Host deployment. We have (on purpose) not installed the security center VM agent, as the ASC team is moving towards using OMS agent for their purposes. Once they completely move to OMS agent, this solution will automatically work well with ASC.

## Enable Tinfoil Security for web app vulnerability assessment

Refer the following links to understand the Tinfoil security. It’s an optional component to take in to production.

**Note**: This step involves credit card information, hence, we have not automated it.

<https://azure.microsoft.com/en-us/blog/web-vulnerability-scanning-for-azure-app-service-powered-by-tinfoil-security/>

<https://www.tinfoilsecurity.com/azure>

