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 QuickStart 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](#_Toc473797708)

[2 Pre-deployment Steps 2](#_Toc473797709)

[2.1 Manual creation of Azure AutomationAccount 2](#_Toc473797710)

[3 Deployment steps 2](#_Toc473797711)

[4 Post Deployment Steps 3](#_Toc473797712)

[4.1 Update 1&1 DNS setting with Application Gateway IP 3](#_Toc473797713)

[4.2 Run Post Deployment Powershell Script 5](#_Toc473797714)

[4.3 Schedule Runbooks 5](#_Toc473797715)

[4.4 Configure AD App: 6](#_Toc473797716)

[4.5 SETTING UP THE WEB APPLICATION (BY A SERVICE ADMIN) 9](#_Toc473797717)

[4.6 Install OMS Dashboards VIEWS. 10](#_Toc473797718)

[4.7 Check and verify OMS solutions are collecting data 12](#_Toc473797719)

# High level summary

# Permissions required to deploy

|  |  |
| --- | --- |
| ☛ | **Application stack should be configured by**   * AD Global Admin (if you don’t know what that is refer this link <https://docs.microsoft.com/en-us/azure/active-directory/active-directory-assign-admin-roles#global-administrator> ) * 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 AutomationAccount

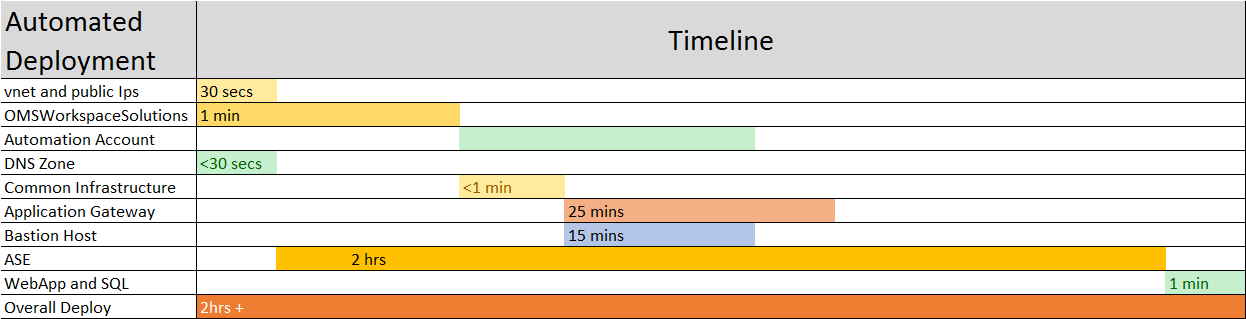
Create an Automation account with **RunAs Service principal**. Unfortunately, ARM templates don't allow for creating AD service principals yet, so this step is currently a manual one.

* Refer the blog https://azure.microsoft.com/en-us/documentation/articles/automationsecconfigure-azure-runas-account/ for the steps.
* Creation of ServicePrincipal has a propensity to fail randomly. A basic verification whether it was successfully created is **mandatory**

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

# Deployment steps

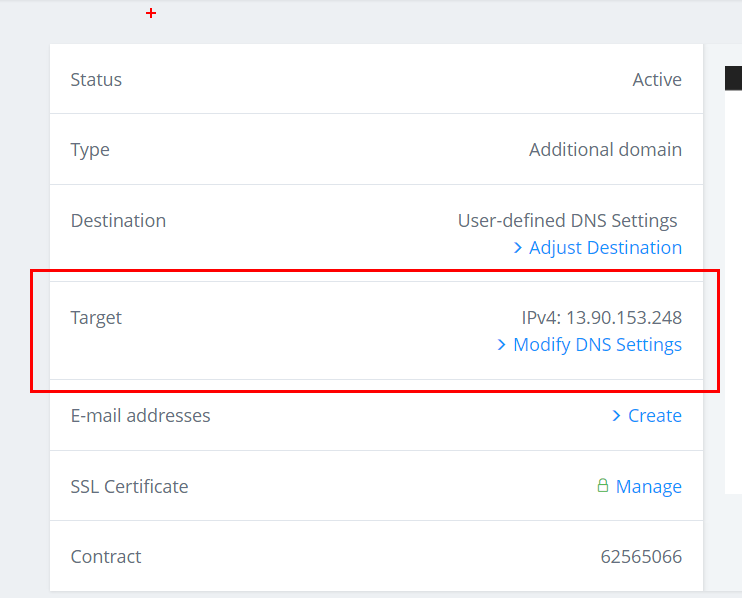
Timeline



# Post Deployment Steps

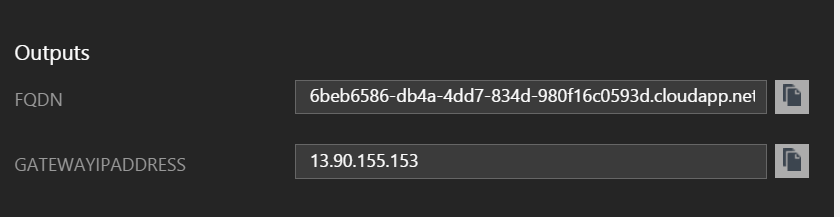
## Update 1&1 DNS setting with Application Gateway IP

Modify the DNS settings under the Target settings

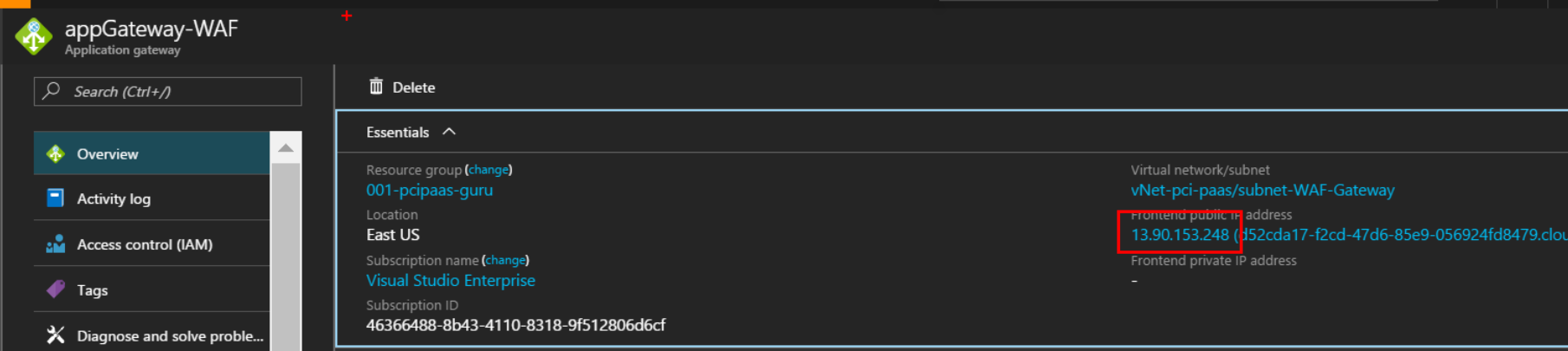


Note the public IP address of App gateway

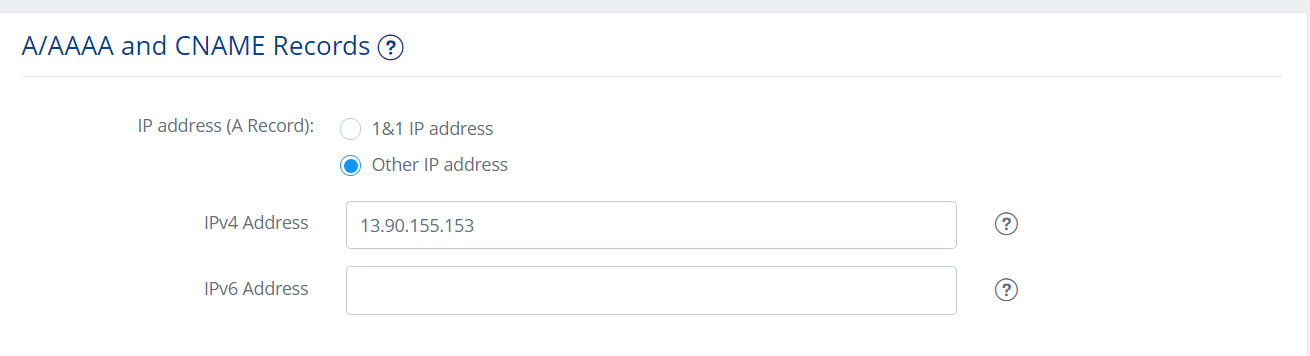
You can find the gateway IP address in the output of the deployment



Or go to the Application gateway object (appGateway-WAF) and checkout the IPaddress

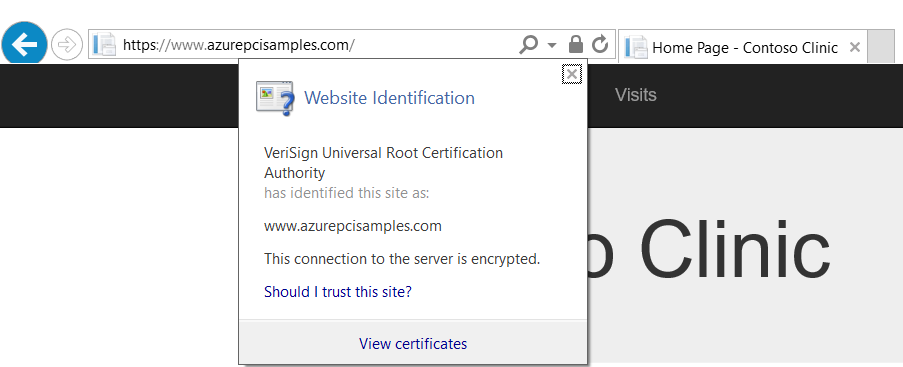


Update the A record IP address to be the App Gateway address



Verification

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



## Run Post Deployment Powershell Script

Post Deployment Powershell script used for 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
* Created AD App Service Principle & Permissions

Process to run Post Deployment Script

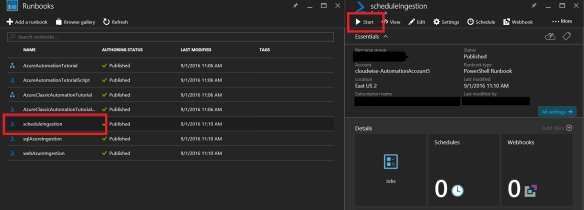
* + - Open Post Deployment Powershell script (Path: ~/ pre-post-deployment/ PostDeployment.ps1)
    - Change the below values in script

|  |
| --- |
| $ResourceGroupName = "01-pci-paas-automation" # Provide Resource Group Name Created through ARM template  $ServerName = "sqlserver-taer4k4qg5zfa" # Provide only Server name not full name  $userId = "testuser" # Provide user id of sql server  $sqlPassword = "PartnerSolutions123" # Provide password of sql server  $cmkName = "CMK1" # Provide Any Name  $cekName = "CEK1" # Provide Any Name  $keyName = "CMK1" # Provide Any Name  $ClientIPAddress = "168.62.48.129" # Provide Client IP address  $ASEOutboundAddress = "13.90.43.202" # Provide ASE Outbound address, we will get it ASE properties in Azure portal  $ADAdministrator = "globaladmin@sunilklive.onmicrosoft.com" # Pass AD Administrator, same we used for ARM Deployment  $subscriptionName = 'Visual Studio Enterprise' # Pass Subscription Name we used to create ARM Deployment  $ArtifactssubscriptionName = 'Cloudly Dev Visual Studio' # Pass Artifacts Subscription Name  $KeyVaultName= 'kv-pcisamples-taer4k4q' # Pass Key Vault Created through ARM template  $cloudwiseAppServiceURL = "" # this is the Unique URL of the Cloudwise App Service deployed by the ARM script. e.g. "http://localcloudniti.sunilklive.onmicrosoft.com"  $suffix = "" #-- Name of the company/deployment. This is used to create a unique website name in your organization. e.g. "MSFT - " |

* + - It will ask Subscription credentials and SQL Credentials to execute complete script

## Schedule Runbooks

* + - Click open the scheduleIngestion runbook and click start to run the runbook. This step will kickstart the data ingestion to the OMS workspace specified.



## Configure AD App:

* 1. In Azure Portal search for Azure Active directory. Open the “App Registrations” 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.

* 1. Configure the following permissions in the “Required Permissions” tab
  2. List of Permissions

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

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

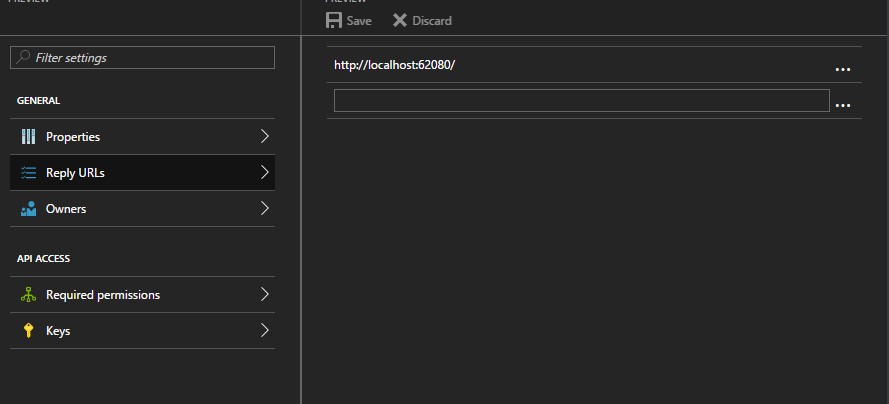
|  |  |
| --- | --- |
|  | Delegated Permissions |

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



|  |
| --- |
| Setting up the web application (by a global admin) |

1. **Ensure**: Reply url of the webapplication is configured in the AD application. It should be something like http://webapp.ase.azurepcisamples.com . or <http://www.azurepcisamples.com>



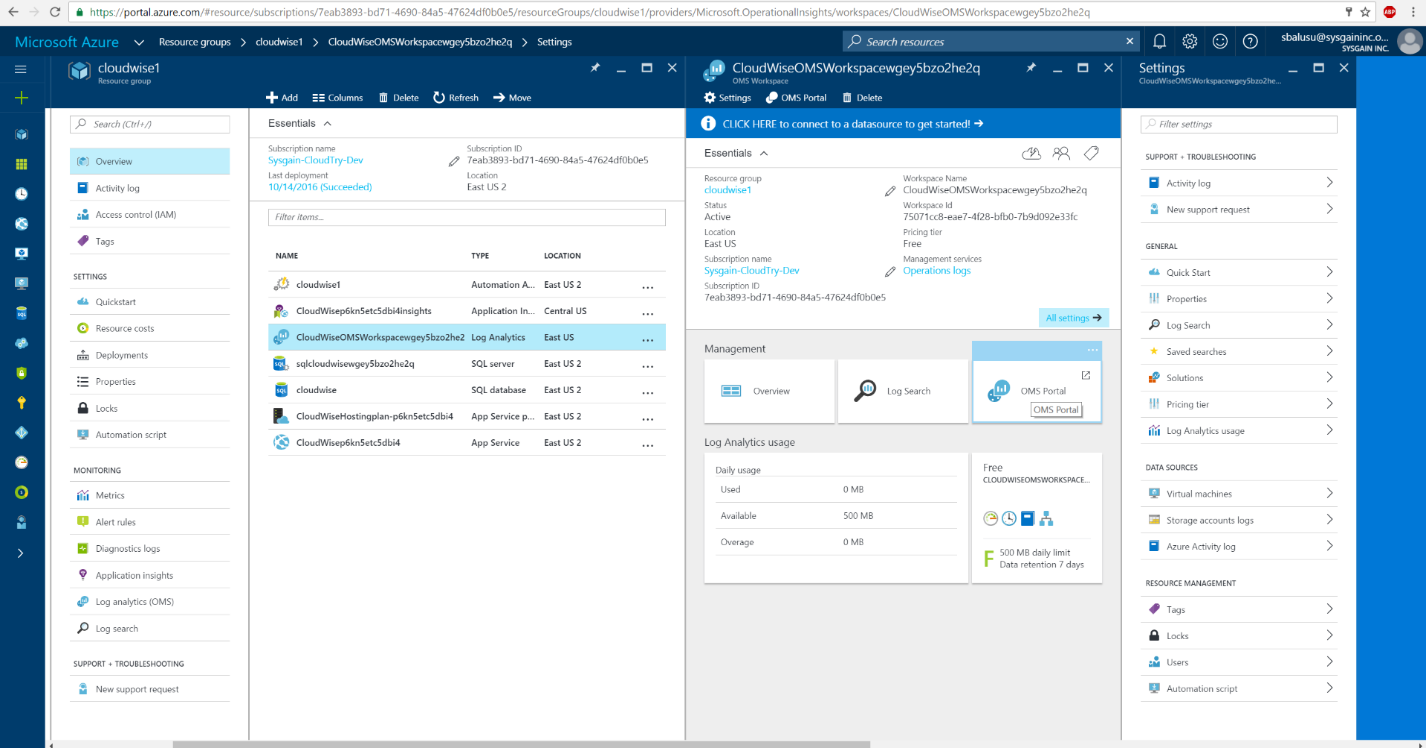
1. Open Portal and then you get an initial screen where you need to put subscription id,Client APplicaion ID and Client secret and then press submit.
2. After submit you will be redirect to AD login page where you need to input your active directory url and then press GO.
3. After this you need to input your LiveID credentials and after successfully login you will be redirect to Rule page where you can see all rules and submit rules.

## Install OMS Dashboards Views.

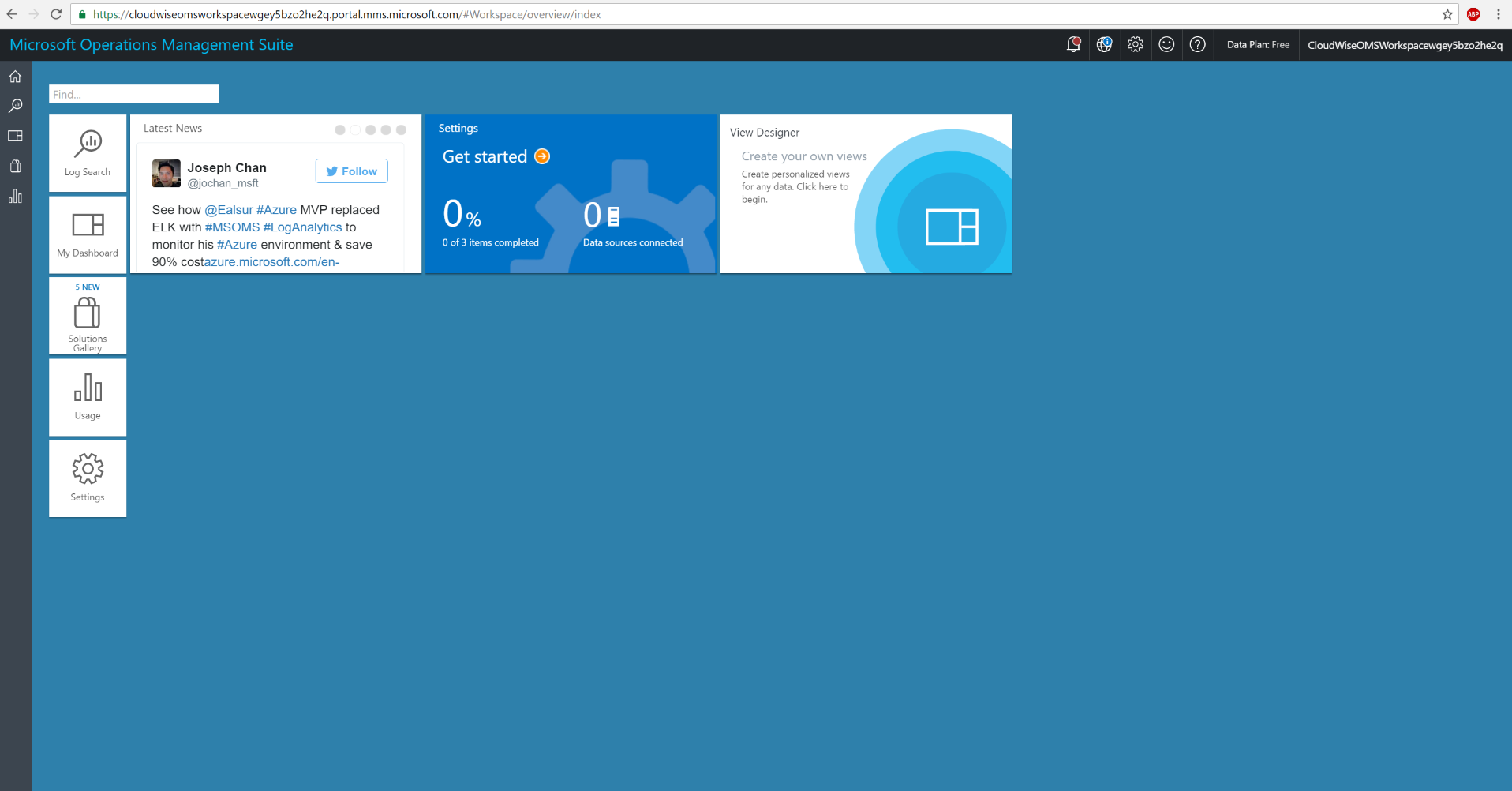
This is currently a manual process as ARM json deploys do not yet support creation of OMS views.

(By a ServiceAdmin/Contributor role)

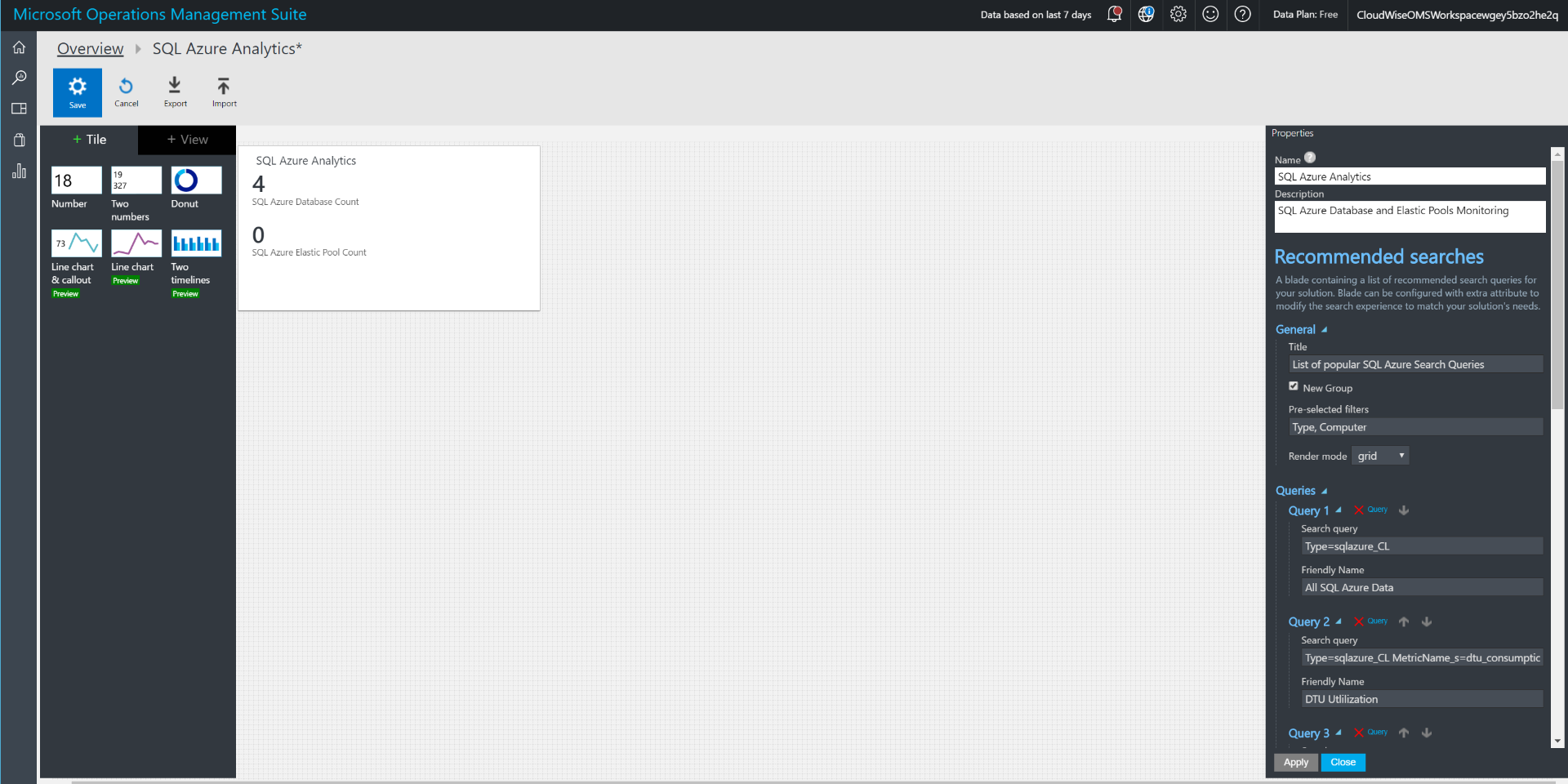
1. Open the resource group and click on the OMS Portal link. This will open the OMS portal in a different window

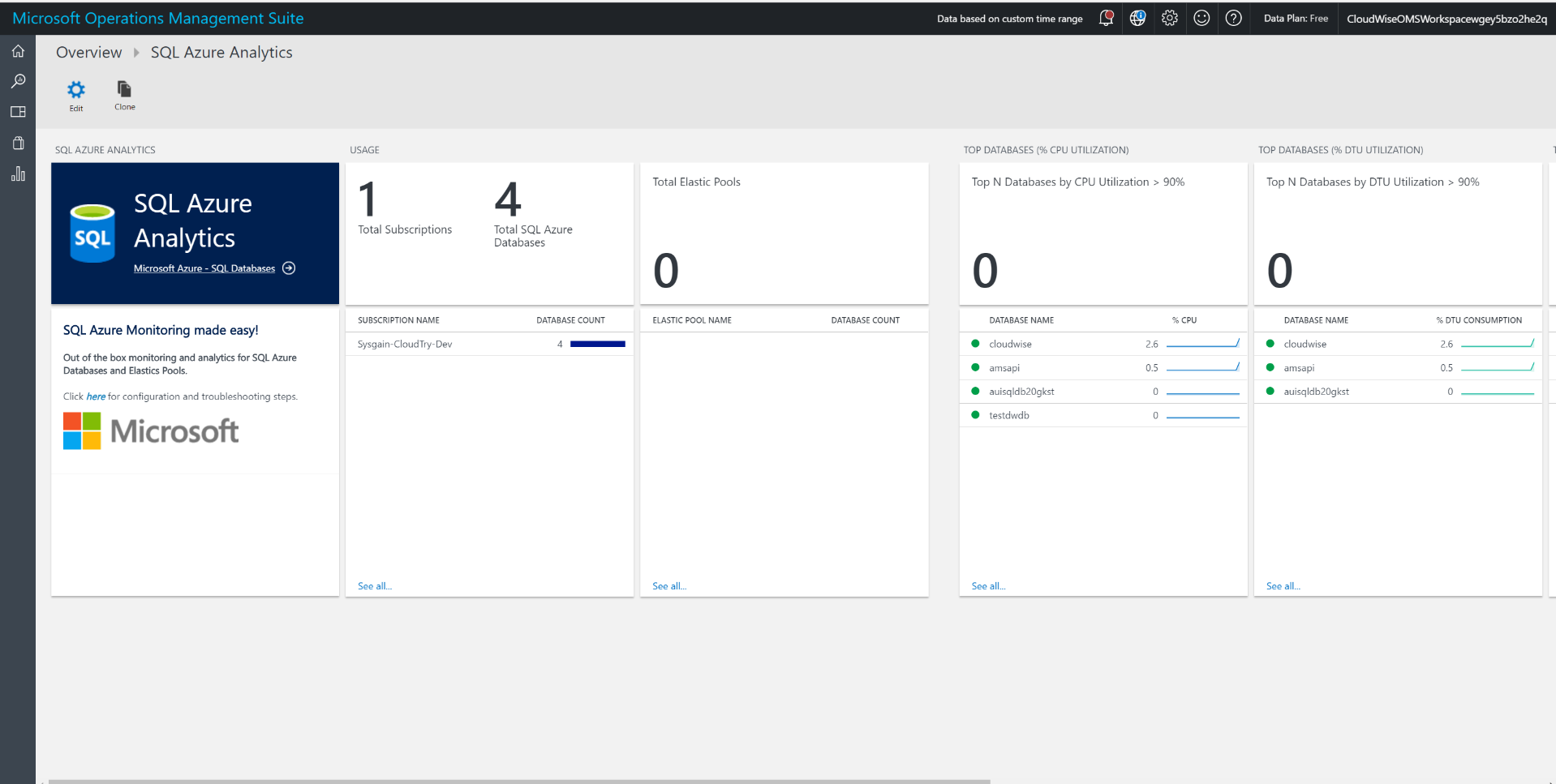


1. Click on the View Designer

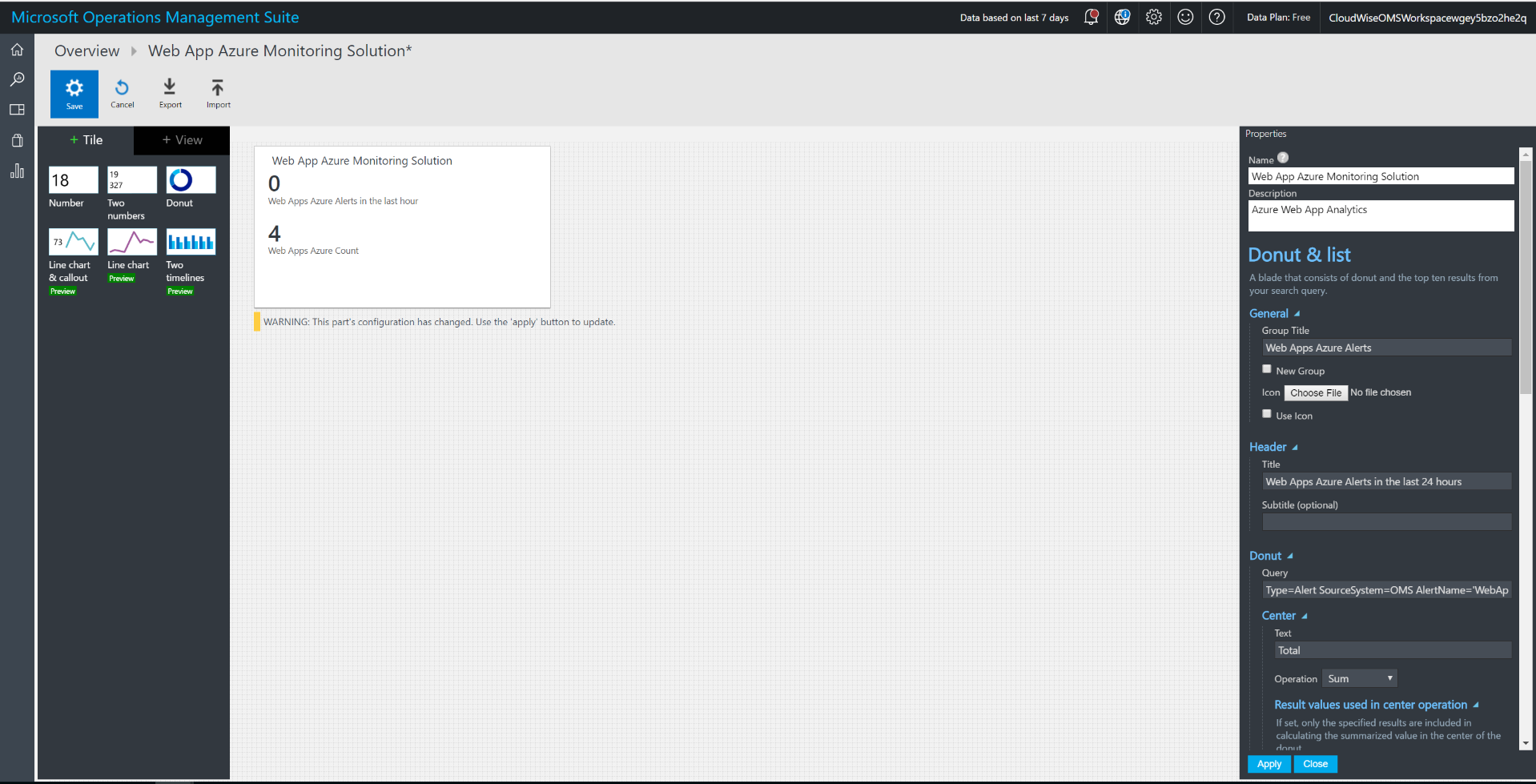


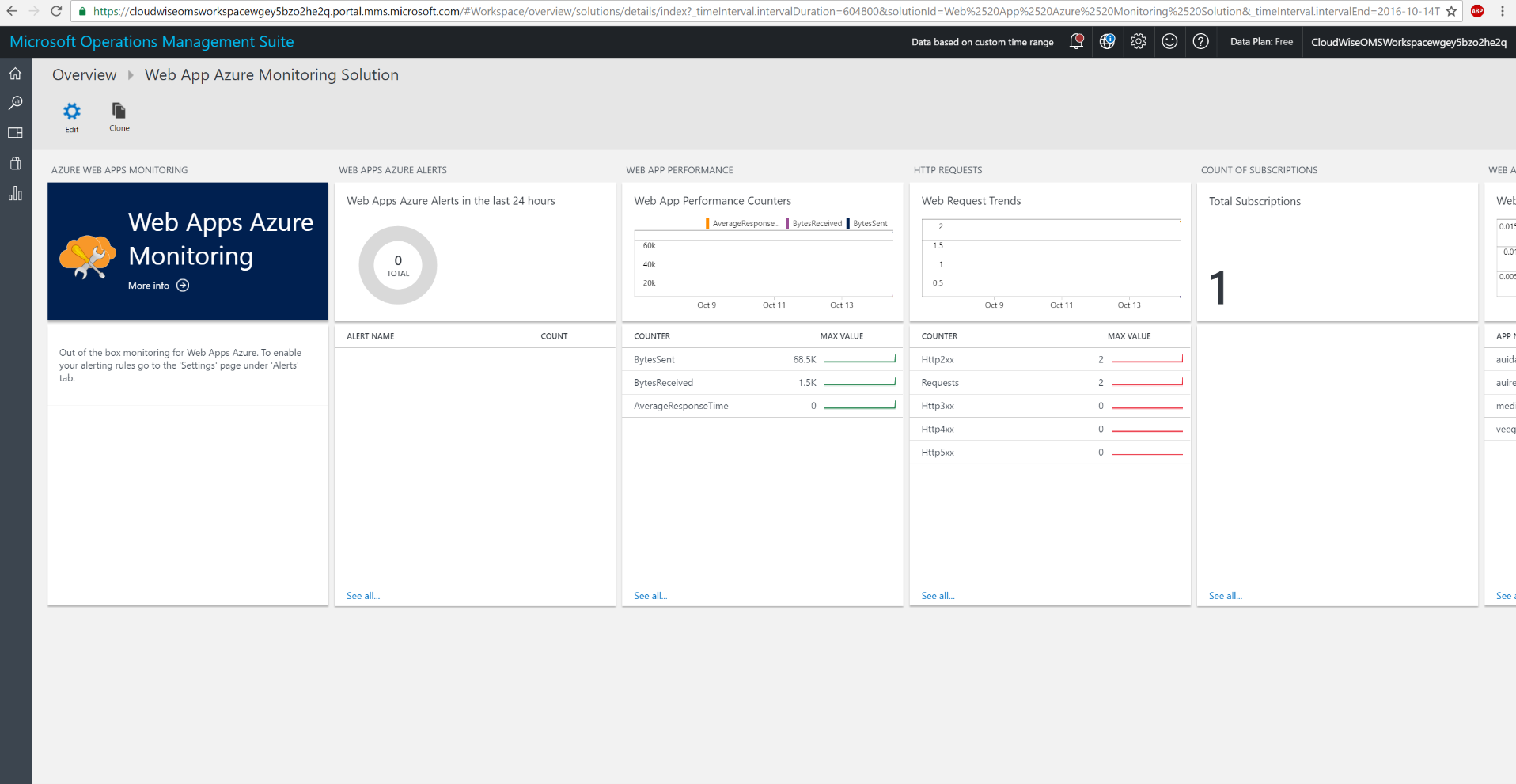
1. Import SQL DB view by clicking on the Import button and browsing to the file (OMSAzureDashboards\OMSSQLDBAzureMonitoringSolution.omsview)





1. Repeat the same step for the Web App Monitoring datboard. Import view OMSAzureDashboards\ OMSWebAppAzureMonitoringSolution.omsview





## Check and verify OMS solutions are collecting data