# Quote Management System

**Administration and set up documentation**

Contents

[Quote Management System 1](#_Toc5972903)

[Overview 2](#_Toc5972904)

[Provisioned Services 2](#_Toc5972905)

[Original code location 2](#_Toc5972906)

[Provision script. 2](#_Toc5972907)

[Default Solution Settings 3](#_Toc5972908)

[QMS Service Account Registration 4](#_Toc5972909)

[QMS Portal account Registration. 5](#_Toc5972910)

[Application Web Portal settings. 5](#_Toc5972911)

[Adding new Subscription. 8](#_Toc5972912)

[Remove Subscription from monitoring 9](#_Toc5972913)

[Disable QMS 9](#_Toc5972914)

# Overview

The Quote Management System (QMS) is designed as alternative solution for Quote management available only on Subscription level for IaaS in Azure.

QMS consist of PaaS resources internally monitoring deployment activity on the Subscription and setup limitation or deny deployment depends of the provided settings for each Resource Group.

The purpose of the QMS is setting up live monitoring and limitation for the deployment of new IaaS environment in resource group, including VM and Scale set.

# Provisioned Services

Current implementation of the QMS solution contains following Azure services.

* Azure Resource Grid – 1 per subscription
* Azure Storage Account with Table and Queue – 1 per solution
* Azure Function – 1 instance per solution hosted 2 functions.
* Azure Web Application – 1 per solution
* Azure Application Insights – 2 per solution
* Solution Resource Group – 1 per solution
* Custom Policy definition - 2 per subscription.
* Policy Assignment - If exists, 1 per Resource Group and Per subscription.

## Original code location

<https://github.com/true-while/qms-web>

# Provision script.

Provisioning script provided with solution **deploy.sh** contains CLI command to provision solution.

Part 1 - Template deployment

Part 2 – Build Policy Definition

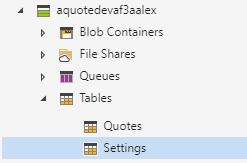
Part 3 – Update application settings

1. Publish Web App from ZIP file.
2. Publish Function from ZIP file.
3. Get Storage Account String
4. Set App settings For Function
5. Generate Event Grid for Quotes
6. Generate Function execution Code
7. Update Function web hock from App settings.

# Default Solution Settings

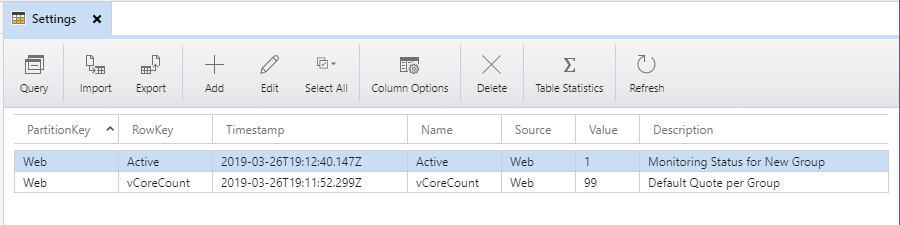
From provisioned storage account the following tables need to be manually created:

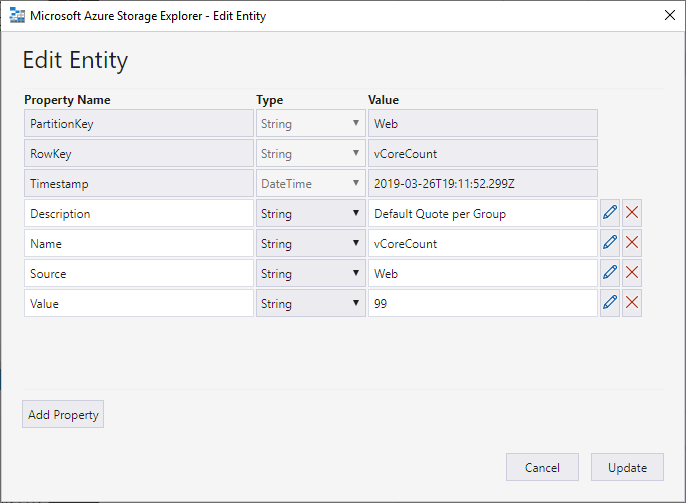
**Quotes** and **Settings**

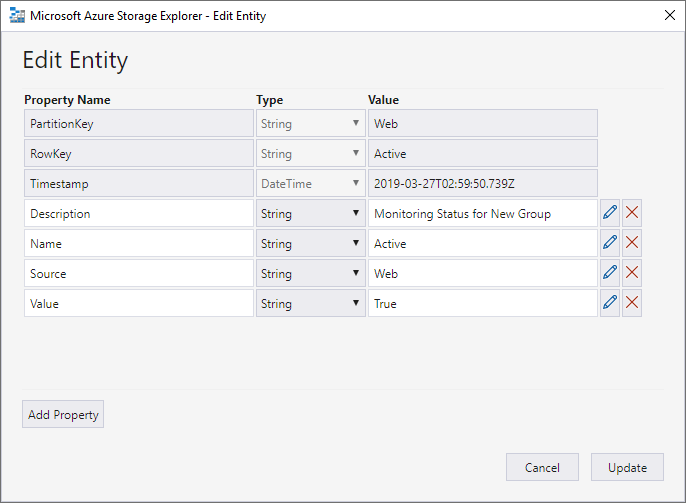


Following default settings need to be provisioned before first start of the application.

Settings name: “Active” and “vCoreCount”



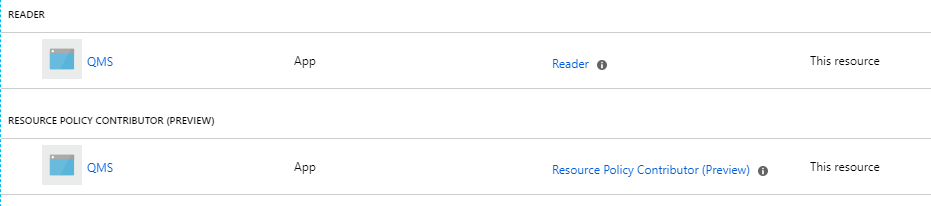




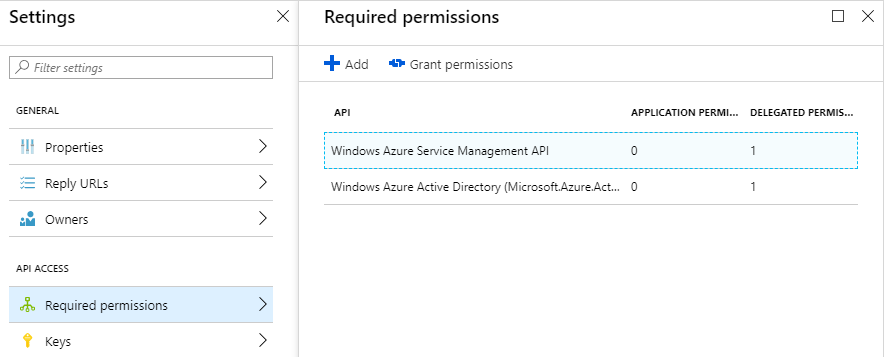
# QMS Service Account Registration

The QMS service account required to be created with following access rights (name QMS might be different):

1. RBAC for subscriptions to be monitored: READER and RESOURCE POLICY CONTRIBUTOR (PREVIEW)



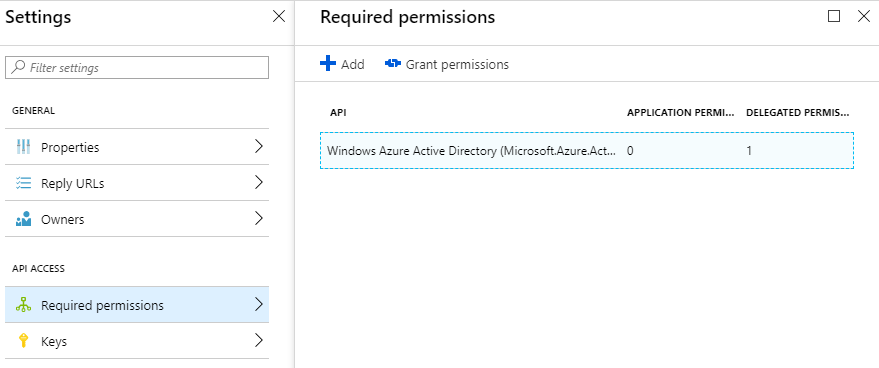
1. Required permission for Azure Services: Windows Azure



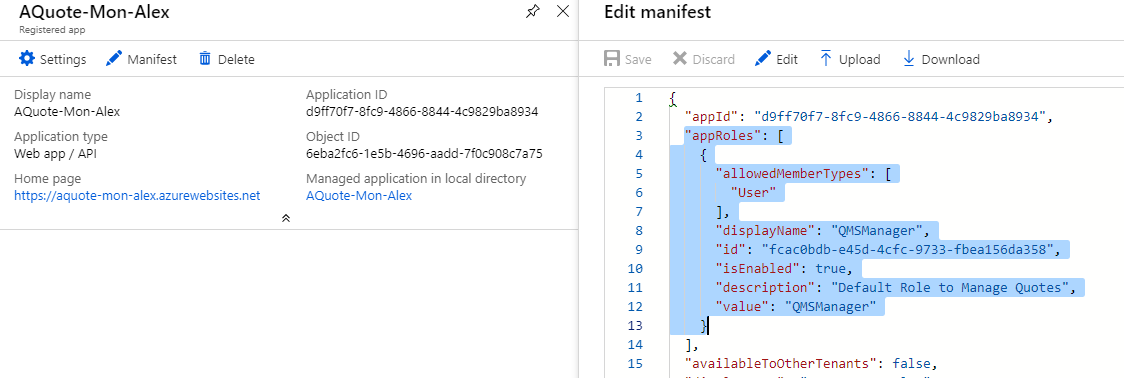
# QMS Portal account Registration.

The QMS Portal app will be responsible for authentication users who will access to the portal and role assignment for the portal.

1. Account required read permission for Azure AD Walmart.



1. Account required update of manifest. Adding role named “QMS Manager”



ID can be generated; new roles can be added as required. The name **must** be “QMSManager” - has hardcoded checks in the Web App code.

"appRoles": [

{

"allowedMemberTypes": [

"User"

],

"displayName": "QMSManager",

"id": "fcac0bdb-e45d-4cfc-9733-fbea156da358",

"isEnabled": true,

"description": "Default Role to Manage Quotes",

"value": "QMSManager"

}

# Application Web Portal settings.

1. Settings for Web app.



Application ID for the [QMS Service Account](#_QMS_Service_Account)

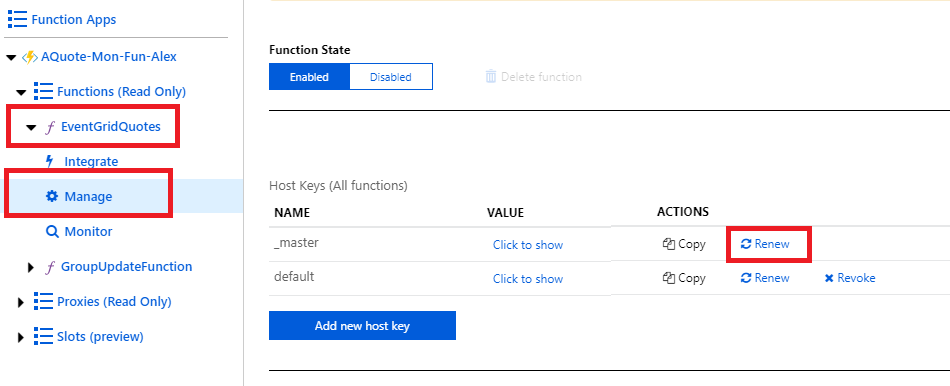
* AppID
* AppKey
* SubscriptionID
* TenantID

Application ID for [QMS Portal](#_QMS_Portal_account) (authentication web app with AAD)

* AuthClientID
* AuthInstance
* AuthTenantID

Webhook to trigger a function for update quotes.

* HttpFunctionWebHook - make sure that function code does not contains “/” – you can regenerate code for function to avoid “/”



Good code:

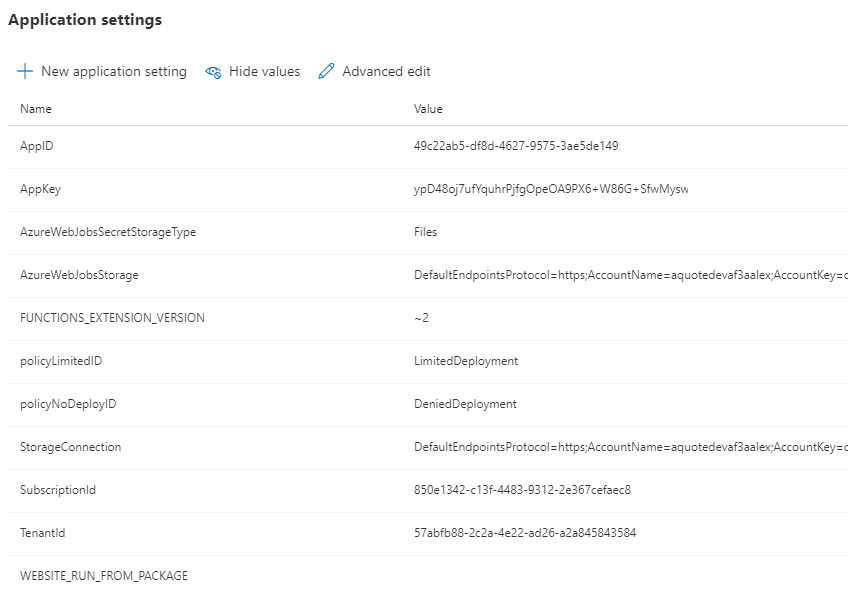


Bad code



Link to the Table Settings and Table Quotes

* StorageConnection

Application Azure Function settings 

Application ID for the [QMS Service Account](#_QMS_Service_Account)

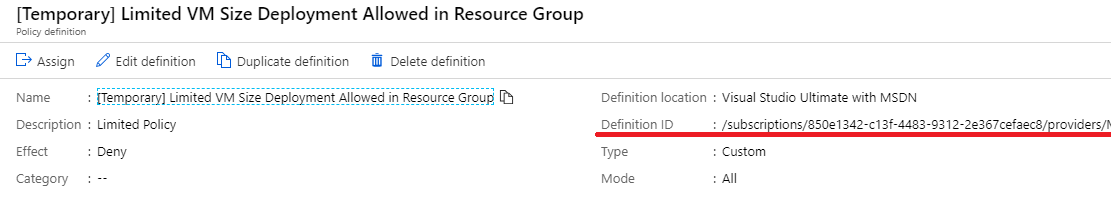
* AppID
* AppKey
* SubscriptionID
* TenantID

Default Value – do not need to be changed.

* AzureWebJobsSecretStorageType
* FUNCTIONS\_EXTENSION\_VERSION

Link to the Table Settings and Table Quotes

* StorageConnection
* AzureWebJobsStorage

Policy ID need to be copied from created policy. Policy name must be the same for all monitored subscription. 

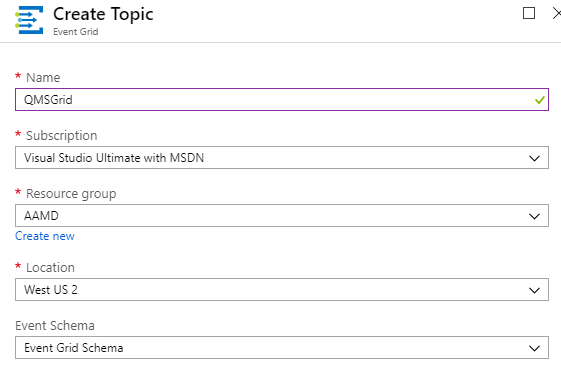
* policyLimitedID
* policyNoDeployID

# Adding new Subscription.

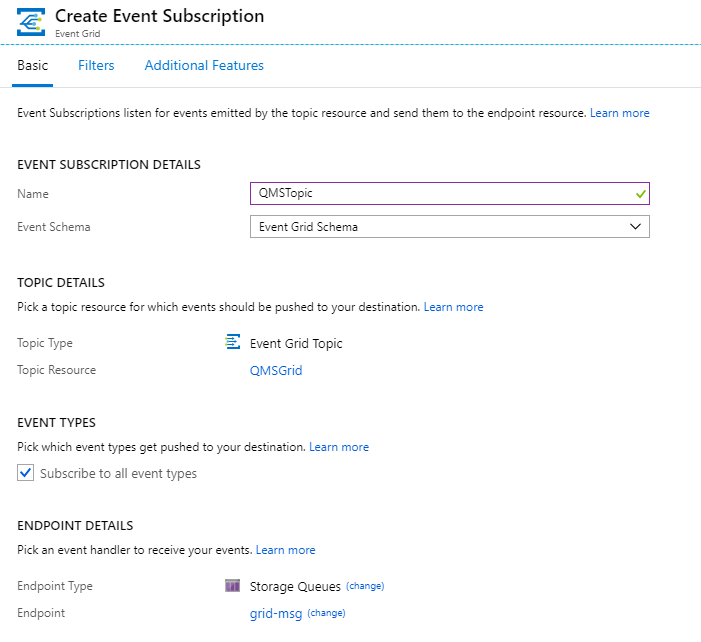
Assignment a new subscription for monitoring can be done manual or by script **per\_sub.sh**

Manually:

1. Create Policy definition with name [provided](#_Application_settings.) in function settings
2. Assign QMS service account to the subscription for [roles Read and Policy Contributor](#_QMS_Service_Account).
3. Create new Event Grid subscription for existed Queue. (Queue need to be manually (or deployment script) created for storage account provided in the settings.)
4. New event subscription need to be created.



1. New Topic subscription need to be created. Instead of all events



# Remove Subscription from monitoring

1. Delete Event Grid Topic for subscription
2. Delete policy assignments for resource group in subscription
3. Delete Policy Definition for Limited and Denied deployment

# Disable QMS

1. Delete Event Grid Topic set up [previously created per subscription](#_Adding_new_Subscription.).
2. Disable or delete policy assignment and definition [previously created per subscription](#_Adding_new_Subscription.).
3. Disable Function.
4. Disable Web App.
5. Remove [QMS Service App](#_QMS_Service_Account) and [QMS Portal App](#_QMS_Portal_account)
6. Delete storage accounts and QMS Resource group.