Icon

Description automatically generated

Start/Stop VMs during off hours – V2

User Guide



2/26/2021

Prepared by

**Bala**

Table of Contents

[1 Start/Stop V2 Deployment 4](#_Toc65445445)

[1.1 Overview 4](#_Toc65445446)

[1.2 Discovering the solution 4](#_Toc65445447)

[1.3 Pre-requisite 4](#_Toc65445448)

[1.4 Deployment 5](#_Toc65445449)

[1.5 Post-Deployment 6](#_Toc65445450)

[2 Start/Stop V2 Resources 7](#_Toc65445451)

[2.1 Azure Function 7](#_Toc65445452)

[2.2 Logic Apps 7](#_Toc65445453)

[2.3 App Insight 8](#_Toc65445454)

[2.4 Azure Dashboard 8](#_Toc65445455)

[2.5 Storage Account 8](#_Toc65445456)

[3 Scenarios & Options 9](#_Toc65445457)

[3.1 Supported Scenarios 9](#_Toc65445458)

[3.2 Supported Scopes on Scenario 9](#_Toc65445459)

[3.2.1 Subscription Scope 9](#_Toc65445460)

[3.2.2 Resource Group Scope 10](#_Toc65445461)

[3.2.3 VMList Scope 10](#_Toc65445462)

[3.3 Payload Schema 10](#_Toc65445463)

[3.3.1 Scheduled Scenario 11](#_Toc65445464)

[3.3.2 Sequenced Scenario 12](#_Toc65445465)

[3.3.3 AutoStop Scenario 14](#_Toc65445466)

[4 Troubleshooting 17](#_Toc65445467)

[4.1 Azure Dashboard 17](#_Toc65445468)

[4.2 Logic App 17](#_Toc65445469)

[4.3 Storage Table 18](#_Toc65445470)

[4.4 Azure Function 18](#_Toc65445471)

1. Start/Stop V2 Deployment
   1. Overview

The Start/Stop VMs during off-hours V2 feature start or stops Azure VMs. It starts or stops machines on user-defined schedules, provides insights through Azure Application Insights logs, and sends optional emails by using [action groups](https://docs.microsoft.com/en-us/azure/azure-monitor/platform/action-groups). The feature can be enabled on both Azure Resource Manager and classic VMs for most scenarios. This V2 supports all the scenarios & scopes of V1 and additionally it supports 2 new features as below

* Start or Stop VMs across multiple azure subscriptions.
* Auto Update the latest solution without needing to redeploy.
  1. Discovering the solution

We would like to adopt and embrace the GitHub open source culture and to start with that we have hosted the V2 deployment links in GitHub

* [Start/Stop V2 Deployment](https://github.com/microsoft/startstopv2-deployments/blob/main/README.md)
  1. Pre-requisite

Before proceeding with deployment make sure you have the below things

* At least contributor permission at the subscription level is needed.
* The deployment process has been designed using ARM template and the scope targeted at the subscription level.
  1. Deployment

Click [here](https://github.com/microsoft/startstopv2-deployments/blob/main/README.md) to begin the deployment. Please follow the below steps and provide the necessary inputs in the ARM template.

1. Based on your azure cloud environment (Public Vs National) click the appropriate “Deploy to Azure” button. 
2. This will open the custom ARM template deployment.
3. Select your subscription from the dropdown and the region.
4. Provide the value for each parameter. Parameters related to Start/Stop V2 is marked below in red box.
5. Click Review + create button.
6. Click Create button to start the deployment. This will take approximately 3 - 5 minutes to complete.

Graphical user interface, application

Description automatically generated

* 1. Post-Deployment

Once the deployment completes navigate to the resource group (the one you have given during deployment) and you should be able to see the various resources as below. In the next section we will see what the usage of each resource

Graphical user interface, text, application

Description automatically generated

1. Start/Stop V2 Resources
   1. Azure Function

Azure function is the core engine which handles the VMs start and stop execution. To support various scenarios and scopes we have created separate HTTP endpoint function to handle that accordingly. Below is the snapshot of the functions that will be deployed.

Graphical user interface, application

Description automatically generated with medium confidence

For example, **Scheduled** HTTP trigger function will be used to handle scheduled & sequenced scenarios. Similarly **AutoStop** HTTP trigger function will be used to handle the autostop scenario.

The remaining queue based trigger functions are required which is part of the V2 architectural design.

All the timer based triggers are used as to perform the availability test and to monitor the health of the system.

* 1. Logic Apps

Logic Apps is mainly used as UI to configure the schedules for the VM take action via calling the function app using payload JSON. By default the solution deployment will create total 5 logic apps for all the scenarios.

* Scheduled scenario : To configure the scheduled start and stop you can use the 2 logic apps ([ststv2\_vms\_Scheduled\_start](https://ms.portal.azure.com/#@microsoft.onmicrosoft.com/resource/subscriptions/374a5ead-138f-464d-a1ab-0d219b8348e4/resourceGroups/ststv2new5/providers/Microsoft.Logic/workflows/ststv2_vms_Scheduled_start) & [ststv2\_vms\_Scheduled\_stop](https://ms.portal.azure.com/#@microsoft.onmicrosoft.com/resource/subscriptions/374a5ead-138f-464d-a1ab-0d219b8348e4/resourceGroups/ststv2new5/providers/Microsoft.Logic/workflows/ststv2_vms_Scheduled_stop))
* Sequenced scenario : To configure the sequenced start and stop you can use the 2 logic apps ([ststv2\_vms\_Sequenced\_start](https://ms.portal.azure.com/#@microsoft.onmicrosoft.com/resource/subscriptions/374a5ead-138f-464d-a1ab-0d219b8348e4/resourceGroups/ststv2new5/providers/Microsoft.Logic/workflows/ststv2_vms_Sequenced_start) & [ststv2\_vms\_Sequenced\_stop](https://ms.portal.azure.com/#@microsoft.onmicrosoft.com/resource/subscriptions/374a5ead-138f-464d-a1ab-0d219b8348e4/resourceGroups/ststv2new5/providers/Microsoft.Logic/workflows/ststv2_vms_Sequenced_stop))
* AutoStop scenario : To configure the autostop functionality you can use [ststv2\_vms\_AutoStop](https://ms.portal.azure.com/#@microsoft.onmicrosoft.com/resource/subscriptions/374a5ead-138f-464d-a1ab-0d219b8348e4/resourceGroups/ststv2new5/providers/Microsoft.Logic/workflows/ststv2_vms_AutoStop) logic app.
  1. App Insight

All the function app execution will gets logged in the app insight which will used it for telemetry dashboard and send email to customer regarding the action performed on the VMs.

* 1. Azure Dashboard

Dashboard helps customer to track and validate the action performed on the VMs.

* 1. Storage Account

Storage account has close connection with Azure function which is integral part of the framework of azure function. Beyond that we have utilized the storage table to store the execution operation metadata for the customer to validate the VMs start and stop action. We have utilized the storage queue which is used by the azure function queue based trigger.

1. Scenarios & Options
   1. Supported Scenarios

Currently start/stop supports 3 scenarios

* Scheduled start/stop
  + This is scheduled based start and stop action that you can perform on VMs.
  + **This scenario will support both ARM/Classic VMs**.
* Sequenced start/stop
  + This is scheduled based start and stop action that you can perform on VMs with predefined sequencing tags on the VMs.
  + Currently it supports 2 tags
    - **sequencestart**
    - **sequencestop**
  + **This scenario will support only ARM VMs since it acts on VM tags**.
* AutoStop
  + This functionality will be mainly used to take only stop action on VMs based on its CPU utilizations.
  + This is also scheduled based take action which create alerts on VMs and based on the condition the alert will get triggered to perform the needed action which is stop functionality.
  + **This scenario will support both ARM/Classic VMs**.
  1. Supported Scopes on Scenario

Currently the start/stop supports 3 scopes for each scenario

* + 1. Subscription Scope
* This option will be used when you need to perform the start and stop action on **all the VMs in the entire subscription**.
* Currently we have extended this functionality to support multi-subscription as well.
* You can also provide the exclude VMs list so that it will ignore those VMs from the action.
* It also supports wildcard options like for example if you provide the VM name as Az\* which means VM name that starts with Az will be ignored.
  + 1. Resource Group Scope
* This option will be used when you need to perform the start and stop action on all the VMs in the **list of resource group names**.
* Currently we have extended this functionality to support multi-subscription as well.
* You can also provide the exclude VMs list so that it will ignore those VMs from the action.
* It also supports wildcard options like for example if you provide the VM name as Az\* which means VM name that starts with Az will be ignored.
  + 1. VMList Scope
* This option will be used when you need to perform the start and stop action on list of VMs that is provided.
* Currently we have extended this functionality to support multi-subscription as well.
* This option doesn’t support the exclude VMs.
* No wildcard is support in the VMList names.
  1. Payload Schema

Payload schema is a JSON format input that will be passed to azure functions via logic app task.

* + 1. Scheduled Scenario

Below is the payload schema for scheduled start/stop

* **Subscription Level**

{

"Action": "start",

"EnableClassic": false,

"RequestScopes": {

"ExcludedVMLists": [],

"Subscriptions": [

"/subscriptions/12345678-1234-5678-1234-123456781234/"

]

}

}

* **Resource Group Level**

{

"Action": "start",

"EnableClassic": false,

"RequestScopes": {

"ExcludedVMLists": [],

"ResourceGroups": [

"/subscriptions/12345678-1234-5678-1234-123456781234/resourceGroups/rg1/",

"/subscriptions/12345678-1234-5678-1234-123456781234/resourceGroups/rg2/"

]

}

}

* **VMList Level**

{

"Action": "start",

"EnableClassic": true,

"RequestScopes": {

"ExcludedVMLists": [],

"VMLists": [

"/subscriptions/12345678-1234-5678-1234-123456781234/resourceGroups/rg1/providers/Microsoft.Compute/virtualMachines/vm1",

"/subscriptions/12345678-1234-5678-1234-123456781234/resourceGroups/rg2/providers/Microsoft.ClassicCompute/virtualMachines/vm2",

"/subscriptions/12345678-1234-5678-1234-123456781234/resourceGroups/rg3/providers/Microsoft.Compute/virtualMachines/vm3"

]

}

}

* + 1. Sequenced Scenario

Below is the payload schema for sequenced start/stop

* **Subscription Level**

{

"Action": "stop",

"RequestScopes":{

"Subscriptions":[

"/subscriptions/12345678-138f-1234-1243-1234567891234/",

"/subscriptions/11112222-3333-4444-5555-565656565656/"

],

"ExcludedVMLists":[

"/subscriptions/12345678-138f-1234-1243-1234567891234/resourceGroups/vmrg1/providers/Microsoft.Compute/virtualMachines/vm1"

]

},

"**Sequenced**": **true**

}

* **Resource Group Level**

{

"Action": "start",

"RequestScopes":{

"ResourceGroups":[

"/subscriptions/12345678-1111-2222-3333-1234567891234/resourceGroups/vmrg1/",

"/subscriptions/12345678-1111-2222-3333-1234567891234/resourceGroups/vmrg3/"

],

"ExcludedVMLists": [

"/subscriptions/12345678-1111-2222-3333-1234567891234/resourceGroups/vmrg1/providers/Microsoft.Compute/virtualMachines/vm1"

]

},

"**Sequenced**": **true**

}

* **VMList Level**

{

"Action": "stop",

"RequestScopes":{

"VMLists":[

"/subscriptions/12345678-1111-2222-3333-1234567891234/resourceGroups/vmrg1/providers/Microsoft.Compute/virtualMachines/vm1",

"/subscriptions/12345678-1111-2222-3333-1234567891234/resourceGroups/vmrg1/providers/Microsoft.Compute /virtualMachines/vm2"

],

"ExcludedVMLists":[]

},

"Sequenced": true

}

* + 1. AutoStop Scenario

Below is the payload schema for sequenced start/stop

* **Subscription Level**

{

"Action": "stop",

"EnableClassic": false,

"AutoStop\_MetricName": "Percentage CPU",

"AutoStop\_Condition": "LessThan",

"AutoStop\_Description": "Alert to stop the VM if the CPU % exceed the threshold",

"AutoStop\_Frequency": "00:05:00",

"AutoStop\_Severity": "2",

"AutoStop\_Threshold": "5",

"AutoStop\_TimeAggregationOperator": "Average",

"AutoStop\_TimeWindow": "06:00:00",

"RequestScopes":{

"Subscriptions":[

"/subscriptions/12345678-1111-2222-3333-1234567891234/",

"/subscriptions/12345678-2222-4444-5555-1234567891234/"

],

"ExcludedVMLists":[]

}

}

* **Resource Group Level**

{

"Action": "stop",

"AutoStop\_Condition": "LessThan",

"AutoStop\_Description": "Alert to stop the VM if the CPU % exceed the threshold",

"AutoStop\_Frequency": "00:05:00",

"AutoStop\_MetricName": "Percentage CPU",

"AutoStop\_Severity": "2",

"AutoStop\_Threshold": "5",

"AutoStop\_TimeAggregationOperator": "Average",

"AutoStop\_TimeWindow": "06:00:00",

"EnableClassic": true,

"RequestScopes": {

"ExcludedVMLists": [],

"ResourceGroups": [

"/subscriptions/12345678-1111-2222-3333-1234567891234/resourceGroups/vmrg1/",

"/subscriptions/12345678-1111-2222-3333-1234567891234/resourceGroupsvmrg2/",

"/subscriptions/12345678-2222-4444-5555-1234567891234/resourceGroups/VMHostingRG/"

]

}

}

* **VMList Level**

{

"Action": "stop",

"AutoStop\_Condition": "LessThan",

"AutoStop\_Description": "Alert to stop the VM if the CPU % exceed the threshold",

"AutoStop\_Frequency": "00:05:00",

"AutoStop\_MetricName": "Percentage CPU",

"AutoStop\_Severity": "2",

"AutoStop\_Threshold": "5",

"AutoStop\_TimeAggregationOperator": "Average",

"AutoStop\_TimeWindow": "06:00:00",

"EnableClassic": true,

"RequestScopes": {

"ExcludedVMLists": [],

"VMLists": [

"/subscriptions/12345678-1111-2222-3333-1234567891234/resourceGroups/rg3/providers/Microsoft.ClassicCompute/virtualMachines/Clasyvm11",

"/subscriptions/12345678-1111-2222-3333-1234567891234/resourceGroups/vmrg1/providers/Microsoft.Compute/virtualMachines/vm1"

]

}

}

1. Troubleshooting

Below section covers how to troubleshoot when you want to validate and identify the root cause.

* 1. Azure Dashboard
* Dashboard is quick and easy way to verify the status of each operation that’s performed on your VMs.
* Especially the table “recently attempted actions on VMs” will show all the recent operations that executed on VMs.
* Please make sure that there is some latency (approx. 5 mins) to show up the data in this report as it directly maps the application insight resource.
  1. Logic App
* Navigate to the respective logic app i.e. based on the scenario which you have configured and enabled.
* For example if you’re using Scheduled scenario then you have 2 logic apps ([ststv2\_vms\_Scheduled\_start](https://ms.portal.azure.com/#@microsoft.onmicrosoft.com/resource/subscriptions/374a5ead-138f-464d-a1ab-0d219b8348e4/resourceGroups/ststv2heartbeat1/providers/Microsoft.Logic/workflows/ststv2_vms_Scheduled_start), [ststv2\_vms\_Scheduled\_stop](https://ms.portal.azure.com/#@microsoft.onmicrosoft.com/resource/subscriptions/374a5ead-138f-464d-a1ab-0d219b8348e4/resourceGroups/ststv2heartbeat1/providers/Microsoft.Logic/workflows/ststv2_vms_Scheduled_stop)) assigned to perform start and stop operation respectively.
  + Open the “[ststv2\_vms\_Scheduled\_start](https://ms.portal.azure.com/#@microsoft.onmicrosoft.com/resource/subscriptions/374a5ead-138f-464d-a1ab-0d219b8348e4/resourceGroups/ststv2heartbeat1/providers/Microsoft.Logic/workflows/ststv2_vms_Scheduled_start)” logic app
  + Click the last executed task under the Overview section.
  + Open function task

Diagram

Description automatically generated

* + Check the Body for the output message which can be either the successful or failure error.

Graphical user interface, text, application

Description automatically generated

* + Click on “Show raw outputs” section to see the detailed message
    - 
  1. Storage Table
* Navigate to the storage account and click on the “Storage Explorer (preview)” from the left side menu 
* Click the “TABLES” Table

  Description automatically generated with medium confidence
* Open the “requeststoretable”
* This table will have all the operation that performed on the VMs and filter it by Timestamp.
  1. Azure Function
* Navigate to the Azure function and click on “Functions” Graphical user interface, application

  Description automatically generatedfrom the left side menu.
* You will see several functions associated for scenario and below is the example for Scheduled & Sequenced since same function handle both the scenario and varies by payload schema.
  1. Execution flow for Scheduled scenario as below
     1. Scheduled HTTP 🡪 VirtualMachineRequestOrchestrator Queue 🡪 VirtualMachineRequestExecutor Queue
     2. From the logic app the Scheduled HTTP function will be invoked with Payload schema.
     3. Once the Scheduled HTTP function receive the request it will send the information to the Orchestrator queue function which intern create several queues for each VM to perform action.
  2. Click on the Scheduled HTTP function Graphical user interface, application

     Description automatically generated
  3. Click on “Monitor” from the left side menu

Graphical user interface, application

Description automatically generated

* 1. Click on the latest execution trace to see the invocation details and the message section for the detailed logging.
  2. 
  3. Graphical user interface

     Description automatically generated with medium confidence
  4. Repeat the same steps for each functions mentioned in A1.