

# ScheduledActions Manual

*[Last Edit Date: September 2024.]*

## Contents

Background and Context of ScheduledActions.....	3
Basic knowledge before starting.....	5
Prerequisites .....	5
ScheduledActions endpoints Examples .....	5
Instructions with examples .....	5
Overview .....	5
Endpoint examples.....	6
VirtualMachinesSubmitStart endpoint .....	6
VirtualMachineSubmitDeallocate endpoint .....	8
VirtualMachineSubmitHibernate endpoint .....	11
VirtualMachineExecuteStart endpoint .....	13
VirtualMachineExecuteDeallocate endpoint .....	16
VirtualMachineExecuteHibernate endpoint .....	18
VirtualMachinesGetOperationStatus endpoint .....	20
VirtualMachinesCancelOperations endpoint.....	22
Parsing Responses.....	24
Contract type .....	26
Miscellaneous .....	27
Recommended Usage .....	27
Throttler Quotas .....	28
Retry Policy .....	29

## Background and Context of ScheduledActions

### Introduction

Scheduledactions will allow customers schedule one-off operations (Hibernate, Start, Deallocate) for their Azure resources, we currently support Virtual Machines as a resource, however the goal is to support other resource types in the future.

- **SLAs for ComputeSchedule:**
  - For InitiateAt Start/Stop/Hibernate scenario, ScheduledActions will match the current Azure SLAs.
- **ScheduledActions functions:**
  - Schedule operations (SubmitStart/SubmitDeallocate/SubmitHibernate) against a batch of VMs within a subscription and region by a specified deadline.
  - Execute operations (ExecuteStart/ExecuteDeallocate/ExecuteHibernate) against a batch of resources with a deadline of utcNow, deadlinetype of InitiateAt and timezone of UTC by default.
  - Get operation status (Planned/Activated/Completed) of scheduled operations and their error info (if any).
  - Cancel scheduled but not yet executed virtual machine operations, this works like a delete operation.
- **Considerations:**
  - Data related to scheduled operations are purged from our internal stores after 3 days
  - ScheduledActions will use the default retry policy values when either RetryCount or RetryWindowInMinutes are not entered.
    - The default value for RetryWindowInMinutes is 120 minutes
    - The default value for ReryCount is 7
  - ScheduledActions will keep retrying on retrieable errors using the values in RetryPolicy as a retry max
    - The maximum value for RetryWindowInMinutes is 120 minutes while the minimum value is 5 minutes.
    - The maximum value for RetryCount is 7 while the minimum value is 0
  - All virtual machines in one request need to belong to the same subscription id and in the same location the SDK is being called from, that is, if *locationParameter* is eastasia, all virtual machines being operated on must also be in eastasia.
  - Clients agree to a polling interval of 10 seconds per unique request. Polling is allowed for a maximum of 100 operationIds per request for the same operation.
  - For initiateAt requests: ScheduledActions will issue the request to Compute at the requested deadline and will retry when throttling is encountered. In the case of too much concurrency, although ScheduledActions will take care of intermittent errors, clients will experience elevated latency due to throttling from Azure core.
  - The *VirtualMachinesGetOperationStatus* returns a maximum of **100** results per request

### ScheduledActions validations:

These are [the allowed](#) values for operation deadlinetypes in ScheduledActions

- **VM number validation:** Limit of virtual machines per request is set to **100**.
- **Operation conflicting validation:** Deadlines of any two pending operations on the same virtual machine must be more than 1 hour apart from each other. E.g., if an operation on a virtual machine has a deadline of InitiateAt at 9AM, any pending operation between 8-10 on the same virtual machine would create a conflict.
  - **Mitigation:** Cancel the previous operation on the virtual machine using the VirtualMachinesCancelOperations API then schedule a new operation, note if the previous operation has been activated, ie Operation.State is Executing, there is no way to cancel it.
- **Retry window validation:** The retry window in minutes is not allowed to be less than 5 minutes or greater than 120 minutes.
- **Retry Count validation:** The RetryCount is allowed to be between the range of 0 – 7.
- **Timezone Validation:** Scheduledactions will only allow for **UTC** timezone in this first iteration.
- **OperationsRequest Validation:** ScheduledActions will validate that the number of operationids in operations like VirtualMachinesCancelOperations and VirtualMachinesGetOperationStatus are less than or equal to 100.
- **Deadline Validation:** Schedule deadline for submit type operations cannot be more than 14 days in the future or 5 minutes in the past.

### Allowed deadlines

Operation name	Deadline Type	Status
VirtualMachineSubmitDeallocate	InitiateAt	Allowed
VirtualMachineSubmitHibernate	InitiateAt	Allowed
VirtualMachineSubmitStart	InitiateAt	Allowed

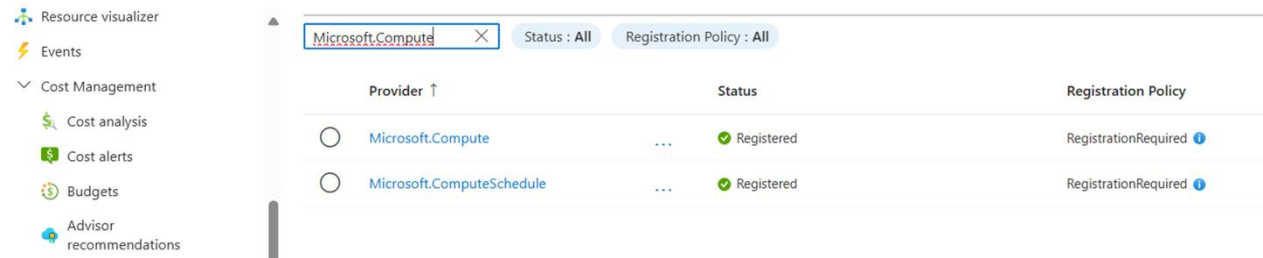
### Exceptions

Exception	Status code	Summary
BadRequestException	400	Invalid or missing parameters, invalid deadlines, parameter over the set limit, etc.
SubscriptionNotFoundException	404	Given subscription was not found
ServiceUnavailableException	503	Scheduling Operations are being blocked, try again retry-after seconds
OperationAlreadyExistsException	409	Attempted operation already exists
VMNotFoundException	404	Given VM was not found.
SchedulingOperationsBlockedException	503	Scheduled operations are being rejected due to an outage in Azure, requests that return this error are not fulfilled.
NonSchedulingOperationsBlockedException	503	Non-Scheduling Operations are being rejected due to an outage in Azure, requests that return this error are not fulfilled.

# Basic knowledge before starting

## Prerequisites

- ARM/AAD is needed to provide service to service authentication.
- Registration of the Microsoft.ComputeSchedule and Microsoft.Compute Resource Providers shown in the image below



## ScheduledActions endpoints Examples

Currently Supported Endpoint Version
2024-06-01-preview ( <b>currently available</b> )
2024-08-15-preview ( <b>available in public preview</b> )

Versioning is query-based which means that API versions are accessed by adding “**?api-version=<version>**” to the end of the ScheduledActions request URL. For example:

`https://management.azure.com/subscriptions/{subscriptionId}/providers/Microsoft.ComputeSchedule/locations/{locationparameter eg australiaeast}/virtualMachinesSubmitStart?api-version=2024-06-01-preview`

## Instructions with examples

### Overview

Below are the operations that can be performed in ScheduledActions

Endpoint	Http Method
<a href="#">VirtualMachinesSubmitStart endpoint</a>	POST
<a href="#">VirtualMachinesSubmitDeallocate endpoint</a>	POST
<a href="#">VirtualMachinesSubmitHibernate endpoint</a>	POST
<a href="#">VirtualMachinesExecuteStart endpoint</a>	POST
<a href="#">VirtualMachinesExecuteDeallocate endpoint</a>	POST
<a href="#">VirtualMachinesExecuteHibernate endpoint</a>	POST

<a href="#">VirtualMachinesGetOperationStatus endpoint</a>	POST
<a href="#">VirtualMachinesCancelOperations endpoint</a>	POST

## Endpoint examples

### VirtualMachinesSubmitStart endpoint

#### Endpoint Sample

<https://management.azure.com/subscriptions/{subscriptionId}/providers/Microsoft.ComputeSchedule/locations/{locationparameter}/virtualMachinesSubmitStart?api-version=2024-06-01-preview>

#### Description

The VirtualMachinesSubmitStart endpoint is used to schedule the start of a batch of resources against a given deadline.

#### VirtualMachinesSubmitStart Request example

```
{
  "schedule": {
    "deadline": "2024-04-24T19:00:00.872Z",
    "timeZone": "UTC",
    "deadlineType": "InitiateAt"
  },
  "resources": {
    "ids": [
      "subscriptions/afe495ca-b99a-4e36-86c8-9e0e41697f1c/resourceGroups/Kronox_CapacityApisDemo_UsWestCentral/providers/Microsoft.Compute/virtualMachines/Kronox-CapacityDemoVm-2"
    ]
  },
  "executionParameters": {
    "retryPolicy": {
      "retryCount": 2,
      "retryWindowInMinutes": 45
    }
  },
  "correlationId": "3fa85f64-5717-4562-b3fc-2c963f66afa6"
}
```

- [ScheduledActions validation](#)
- **subscriptionId**: Client SubscriptionId for the batch of resources.
- **locationparameter**: Enter the known location of the resources.

Request Body:

- **Schedule:** These are the values related to the schedule for the request
  - **Deadline:** Operation deadline in UTC. Deadline of operation is allowed up to 14 days in future.
  - **Timezone:** Timezone for the request, in this iteration, ScheduledActions only supports UTC
  - **DeadlineType:** Operation deadline type. ScheduledActions supports two types of deadlines:
    - **InitiateAt** – Operations with this deadline type are initiated at the specified deadline time (deadlineUtc). This deadline type is valid for all virtual machine operation types.
- **Resources:** These are the resources to schedule a start operation on at the given deadline.
  - **Ids:** This is a list of the full Azure resource name. The number of these resources must be less than or equal to 100 and registered under the customer's subscriptionId.
- **ExecutionParameters:** These are parameters the client can add to support the request, this property is optional and can be omitted.
  - **RetryPolicy:** These are values entered by customers that define customizable constraints for retrievable errors, these values are optional and can be omitted, where they are not entered, ScheduledActions will use default values.
    - **RetryCount:** The retry count is an integer of how many times ScheduledActions should retry retrievable errors, if this value is not explicitly set by the client, the default value of 7 is used.
    - **RetryWindowInMinutes:** The retry window value, in minutes, that ScheduledActions should keep retrying on for retrievable errors, if this value is not explicitly set by the client, the default value of 120 minutes is used.
- **CorrelationId:** Client generated GUID for internal debugging.

#### *VirtualMachinesSubmitStart Response*

```
{
  "description": "Start Resource request",
  "type": "VirtualMachinesSubmitStart",
  "location": "westus",
  "results": [
    {
      "resourceId": "subscriptions/afe495ca-b99a-4e36-86c8-9e0e41697f1c/resourceGroups/Kronox_CapacityApisDemo_UsWestCentral/providers/Microsoft.Compute/virtualMachines/Kronox-CapacityDemoVm-2",
      "errorCode": null,
      "errorDetails": null,
      "operation": {
        "operationId": "9a51d5df-23a1-4aa9-aa42-e4baef9e0f64",
        "resourceId": "subscriptions/afe495ca-b99a-4e36-86c8-9e0e41697f1c/resourceGroups/Kronox_CapacityApisDemo_UsWestCentral/providers/Microsoft.Compute/virtualMachines/Kronox-CapacityDemoVm-2",
        "opType": "Start",
        "subscriptionId": "700935bc-adf2-4176-b9ad-c571731c09fc",
        "deadline": "2023-12-12T19:01:10.872+00:00",
        "deadlineType": "InitiateAt",
        "state": "PendingScheduling",
        "timeZone": "utc",
        "resourceOperationError": null,
        "completedAt": null,
        "retryPolicy": {
          "retryWindowInMinutes": 45,
          "retryCount": 2,
        },
      },
    }
  ]
}
```

- **Response codes:** Server response code is 200 means scheduling operation has been successfully registered in the system.
  - In case of error, a HTTP error code of value 400+ will be returned based on the error type (404, 400 etc..).
  - If Scheduling Operations are being blocked a 503 (ScheduledOperationsBlockedException) will be returned with a corresponding Retry-After header.
  - In the case of an outage in Compute for example, ScheduledActions will hold on to operations sent in by the client but not submit them for processing to Compute, however these operations will be marked by ScheduledActions as being in a **'Blocked'** state. If the deadline for the operations pass before the outage is resolved and the jobs can be submitted to Compute, these operations are then marked as **Failed**.
- The results field shows a result of type **StartOperationResponse** containing a description of the operation, the type of the operation, location from the user and a list containing the results from each resource operation.
- **Exceptions:**
  - BadRequestException
  - SubscriptionNotFoundException
  - ServiceUnavailableException
  - ScheduledOperationsBlockedException : Unique object result of structure:
    - Code: SchedulingOperationsBlockedException
    - Message: Scheduling Operations are currently being blocked.
    - Retry-After Header
    - HttpStatusCode : 503

## VirtualMachineSubmitDeallocate endpoint

### Endpoint Sample

<https://management.azure.com/subscriptions/{subscriptionId}/providers/Microsoft.ComputeSchedule/locations/{locationparameter}/virtualMachinesSubmitDeallocate?api-version=2024-06-01-preview>

### Description

The VirtualMachinesSubmitDeallocate endpoint is used to schedule deallocation for a batch of resources at a given deadline.



### VirtualMachinesSubmitDeallocate Request example

```
{
  "schedule": {
    "deadline": "2023-12-12T19:13:52.067Z",
    "timeZone": "utc",
    "deadlineType": "InitiateAt"
  },
  "resources": {
    "ids": [
      "subscriptions/afe495ca-b99a-4e36-86c8-9e0e41697f1c/resourceGroups/
      Kronox_CapacityApisDemo_UsWestCentral/providers/Microsoft.Compute/virtualMachines/Kronox-
      CapacityDemoVm-3"
    ]
  },
  "executionParameters": {
    "RetryPolicy": {
      "RetryCount": 2,
      "RetryWindowInMinutes": 50
    }
  },
  "correlationId": "3fa85f64-5717-4562-b3fc-2c963f66afa6"
}
```

- [ScheduledActions validation](#)
- **subscriptionId**: Client SubscriptionId for the batch of resources.
- **locationparameter**: Enter the known location of the resources.

#### Request Body:

- **Schedule**: These are the values related to the schedule for the request
  - **Deadline**: Operation deadline in UTC. Deadline of operation is allowed up to 14 days in future.
  - **Timezone**: Timezone for the request, in this iteration, ScheduledActions only supports UTC
  - **DeadlineType**: Should be set to **InitiateAt**
- **Resources**: These are the resources to schedule a deallocate on at the given deadline.
  - **Ids**: This is a list of the full Azure resource name. The number of these resources must be less than or equal to 100 and registered under the customer's subscriptionId.
- **ExecutionParameters**: These are parameters the client can add to support the request, this property is optional and can be omitted.
  - **RetryPolicy**: These are values entered by customers that define customizable constraints for retrievable errors, these values are optional and can be omitted, where they are not entered, ScheduledActions will use default values.
    - **RetryCount**: The retry count is an integer of how many times ScheduledActions should retry on retrievable errors, if this value is not explicitly set by the client, the default value of 7 is used.
    - **RetryWindowInMinutes**: The retry window value, in minutes, that ScheduledActions should keep retrying for retrievable errors, if this value is not explicitly set by the client, the default value of 120 minutes is used.
- **CorrelationId**: Client generated GUID for internal debugging.

## VirtualMachinesSubmitDeallocate Response

```
{
  "description": "Deallocate Resource request",
  "type": "virtualMachinesSubmitDeallocate",
  "location": "westus",
  "results": [
    {
      "resourceId": "subscriptions/afe495ca-b99a-4e36-86c8-9e0e41697f1c/resourceGroups/Kronox_CapacityApisDemo_UsWestCentral/providers/Microsoft.Compute/virtualMachines/Kronox-CapacityDemoVm-3",
      "errorCode": null,
      "errorDetails": null,
      "operation": {
        "operationId": "d204a73d-7c3a-4469-9eba-4fc880b00d47",
        "resourceId": "subscriptions/afe495ca-b99a-4e36-86c8-9e0e41697f1c/resourceGroups/Kronox_CapacityApisDemo_UsWestCentral/providers/Microsoft.Compute/virtualMachines/Kronox-CapacityDemoVm-3",
        "opType": "Deallocate",
        "subscriptionId": "700935bc-adf2-4176-b9ad-c571731c09fc",
        "deadline": "2023-12-12T19:13:52.067+00:00",
        "deadlineType": "InitiateAt",
        "state": "PendingScheduling",
        "timeZone": "utc",
        "resourceOperationError": null,
        "completedAt": null,
        "retryPolicy": {
          "retryWindowInMinutes": 50
        },
        "retryCount": 2,
      },
    }
  ]
}
```

- **Response codes:** Server response code is 200 means scheduling operation has been successfully registered in the system.
  - In case of error, a HTTP error code of value 400+ will be returned based on the error type (404, 400 etc..).
  - If Scheduling Operations are being blocked a 503 (ScheduledOperationsBlockedException) will be returned with a corresponding Retry-After header.
  - In the case of an outage in Compute for example, ScheduledActions will hold on to operations sent in by the client but not submit them for processing to Compute, however these operations will be marked by ScheduledActions as being in a **'Blocked'** state. If the deadline for the operations pass before the outage is resolved and the jobs can be submitted to Compute, these operations are then marked as **Failed**.
- The results field shows a result of type **DeallocateOperationResponse** containing a description of the operation, the type of the operation, location from the user and a list containing the results from each resource operation.
- **Exceptions:**
  - BadRequestException
  - SubscriptionNotFoundException
  - ServiceUnavailableException
  - ScheduledOperationsBlockedException : Unique object result of structure:
    - Code: ScheduledOperationsBlockedException
    - Message: Scheduling Operations are currently being blocked.

- Retry-After Header
- HttpStatusCode : 503

## VirtualMachineSubmitHibernate endpoint

### Endpoint Sample

<https://management.azure.com/subscriptions/{subscriptionId}/providers/Microsoft.ComputeSchedule/locations/{locationparameter}/virtualMachinesSubmitHibernate?api-version=2024-06-01-preview>

### Description:

The VirtualMachinesSubmitHibernate endpoint is used to schedule hibernation of a batch of resources at a given deadline.

### VirtualMachinesSubmitHibernate Request example

```
{
  "schedule": {
    "deadline": "2023-12-12T19:28:07.351Z",
    "timeZone": "utc",
    "deadlineType": "InitiateAt"
  },
  "resources": {
    "ids": [
      "subscriptions/afe495ca-b99a-4e36-86c8-9e0e41697f1c/resourceGroups/Kronox_CapacityApisDemo_UsWestCentral/providers/Microsoft.Compute/virtualMachines/Kronox-CapacityDemoVm-4"
    ]
  },
  "executionParameters": {
    "RetryPolicy": {
      "RetryCount": 2
    }
  },
  "correlationId": "3fa85f64-5717-4562-b3fc-2c963f66afa6"
}
```

- [ScheduledActions validation](#)
- **SubscriptionId**: Client SubscriptionId for the batch of resources.
- **Locationparameter**: Enter the known location of the resources.

### Request Body:

- **Schedule**: These are the values related to the schedule for the request
  - **Deadline**: Operation deadline in UTC. Deadline of operation is allowed up to 14 days in future.
  - **Timezone**: timezone for the request, in this iteration, ScheduledActions only supports UTC
  - **DeadlineType**: Operation deadline type should be set to **InitiateAt**
- **Resources**: These are the resources to schedule a hibernate on at the given deadline.

- **Ids:** This is a list of the full Azure resource name. The number of these resources must be less than or equal to 100 and registered under the customer's subscriptionId.
- **ExecutionParameters:** These are parameters the client can add to support the request, this property is optional and can be omitted.
  - **RetryPolicy:** These are values entered by customers that define customizable constraints for retrievable errors, these values are optional and can be omitted, where they are not entered, ScheduledActions will use default values.
    - **RetryCount:** The retry count is an integer of how many times ScheduledActions should retry on retrievable errors, if this value is not explicitly set by the client, the default value of 7 is used.
    - **RetryWindowInMinutes:** The retry window value, in minutes, that ScheduledActions should keep retrying for retrievable errors, if this value is not explicitly set by the client, the default value of 120 minutes is used.
- **CorrelationId:** Client generated GUID for internal debugging.

#### *VirtualMachinesSubmitHibernate Response*

```
{
  "description": "Hibernate Resource request",
  "type": "virtualMachinesSubmitHibernate",
  "location": "westus",
  "results": [
    {
      "resourceId": "subscriptions/afe495ca-b99a-4e36-86c8-9e0e41697f1c/resourceGroups/Kronox_CapacityApisDemo_UsWestCentral/providers/Microsoft.Compute/virtualMachines/Kronox-CapacityDemoVm-4",
      "errorCode": null,
      "errorDetails": null,
      "operation": {
        "operationId": "18a618f0-d24f-4633-bc45-cbd8229262ee",
        "resourceId": "subscriptions/afe495ca-b99a-4e36-86c8-9e0e41697f1c/resourceGroups/Kronox_CapacityApisDemo_UsWestCentral/providers/Microsoft.Compute/virtualMachines/Kronox-CapacityDemoVm-4",
        "opType": "Hibernate",
        "subscriptionId": "700935bc-adf2-4176-b9ad-c571731c09fc",
        "deadline": "2023-12-12T19:28:07.351+00:00",
        "deadlineType": "InitiateAt",
        "state": "PendingScheduling",
        "timeZone": "utc",
        "resourceOperationError": null,
        "completedAt": null,
        "retryPolicy": {
          "retryWindowInMinutes": 90,
          "retryCount": 2,
        },
      },
    },
  ]
}
```

- **Response codes:** Server response code is 200 means scheduling operation has been successfully registered in the system.
  - In case of error, a HTTP error code of value 400+ will be returned based on the error type (404, 400 etc..).

- If Scheduling Operations are being blocked a 503 (ScheduledOperationsBlockedException) will be returned with a corresponding Retry-After header.
- In the case of an outage in Compute for example, ScheduledActions will hold on to operations sent in by the client but not submit them for processing to Compute, however these operations will be marked by ScheduledActions as being in a '**Blocked**' state. If the deadline for the operations pass before the outage is resolved and the jobs can be submitted to Compute, these operations are then marked as **Failed**.
- The results field shows a result of type **HibernateOperationResponse** containing a description of the operation, the type of the operation, location from the user and a list containing the results from each resource operation.
- **Exceptions:**
  - BadRequestException
  - SubscriptionNotFoundException
  - ServiceUnavailableException
  - ScheduledOperationsBlockedException : Unique object result of structure:
    - Code: SchedulingOperationsBlockedException
    - Message: Scheduling Operations are currently being blocked.
    - Retry-After Header
    - HttpStatusCode : 503

## VirtualMachineExecuteStart endpoint

### Endpoint Sample

<https://management.azure.com/subscriptions/{subscriptionId}/providers/Microsoft.ComputeSchedule/locations/{locationparameter}/virtualMachinesExecuteStart?api-version=2024-06-01-preview>

### Description

The VirtualMachinesExecuteStart endpoint is used to start a batch of resources, this endpoint allows customers to start resources immediately. For this endpoint, there is no need for the customer to enter schedule information.

### *VirtualMachinesExecuteStart Request example*

```
{
  "resources": {
    "ids": [
      "subscriptions/afe495ca-b99a-4e36-86c8-9e0e41697f1c/resourceGroups/
      Kronox_CapacityApisDemo_UsWestCentral/providers/Microsoft.Compute/virtualMachines/Kronox-
      CapacityDemoVm-6"
    ]
  },
  "executionParameters": {
    "RetryPolicy": {
      "RetryCount": 2,
      "RetryWindowInMinutes": 60
    }
  },
  "correlationId": "3fa85f64-5717-4562-b3fc-2c963f66afa6"
}
```

- [ScheduledActions validation](#)
- **subscriptionId**: Client SubscriptionId for the batch of resources.
- **locationparameter**: Enter the known location of the resources.

#### Request Body:

- **Resources**: These are the resources to run the executeStart operation on.
  - **Ids**: This is a list of the full Azure resource name. The number of these resources must be less than or equal to 100 and registered under the customer's subscriptionId.
- **ExecutionParameters**: These are parameters the client can add to support the request, this property is optional and can be omitted.
  - **RetryPolicy**: These are values entered by customers that define customizable constraints for retrievable errors, these values are optional and can be omitted, where they are not entered, ScheduledActions will use default values.
    - **RetryCount**: The retry count is an integer of how many times ScheduledActions should retry on retrievable errors, if this value is not explicitly set by the client, the default value of 7 is used.
    - **RetryWindowInMinutes**: The retry window value, in minutes, that ScheduledActions should keep retrying for retrievable errors, if this value is not explicitly set by the client, the default value of 120 minutes is used.
- **CorrelationId**: Client generated GUID for internal debugging.

### *VirtualMachinesExecuteStart Response*

```

{
  "description": "Start Resource request",
  "type": "virtualMachinesExecuteStart",
  "location": "westus",
  "results": [
    {
      "resourceId": "subscriptions/afe495ca-b99a-4e36-86c8-9e0e41697f1c/resourceGroups/Kronox_CapacityApisDemo_UsWestCentral/providers/Microsoft.Compute/virtualMachines/Kronox-CapacityDemoVm-6",
      "errorCode": null,
      "errorDetails": null,
      "operation": {
        "operationId": "def4a743-f659-457f-94c8-e1d224d9e352",
        "resourceId": "subscriptions/afe495ca-b99a-4e36-86c8-9e0e41697f1c/resourceGroups/Kronox_CapacityApisDemo_UsWestCentral/providers/Microsoft.Compute/virtualMachines/Kronox-CapacityDemoVm-6",
        "opType": "Start",
        "subscriptionId": "700935bc-adf2-4176-b9ad-c571731c09fc",
        "deadline": "2023-12-12T18:11:08.2829625+00:00",
        "deadlineType": "InitiateAt",
        "state": "PendingScheduling",
        "timeZone": "UTC",
        "resourceOperationError": null,
        "completedAt": null
        "retryPolicy": {
          "retryWindowInMinutes": 60,
          "retryCount": 2,
        },
      },
    }
  ]
}

```

- **Response codes:** Server response code is 200 means execute operation has been successfully registered in the system.
  - In case of error, a HTTP error code of value 400+ will be returned based on the error type (404, 400 etc..).
  - If Scheduling Operations are being blocked a 503 (ScheduledOperationsBlockedException) will be returned with a corresponding Retry-After header.
  - In the case of an outage in Compute for example, ScheduledActions will hold on to operations sent in by the client but not submit them for processing to Compute, however these operations will be marked by ScheduledActions as being in a '**Blocked**' state. If the deadline for the operations pass before the outage is resolved and the jobs can be submitted to Compute, these operations are then marked as **Failed**.
- The results field shows a result of type **StartOperationResponse** containing a description of the operation, the type of the operation, location from the user and a list containing the results from each resource operation.
- **Exceptions:**
  - BadRequestException
  - SubscriptionNotFoundException
  - ServiceUnavailableException
  - ScheduledOperationsBlockedException : Unique object result of structure:
    - Code: SchedulingOperationsBlockedException
    - Message: Scheduling Operations are currently being blocked.
    - Retry-After Header

- HttpStatusCode : 503

## VirtualMachineExecuteDeallocate endpoint

### Endpoint Sample

<https://management.azure.com/subscriptions/{subscriptionId}/providers/Microsoft.ComputeSchedule/locations/{locationparameter}/virtualMachinesExecuteDeallocate?api-version=2024-06-01-preview>

### Description

The VirtualMachinesExecuteDeallocate endpoint is used to deallocate a batch of resources, this endpoint allows customers to deallocate resources immediately. For this endpoint, there is no need for the customer to enter schedule information.

### VirtualMachinesExecuteDeallocate Request example

```
{
  "resources": {
    "ids": [
      "subscriptions/afe495ca-b99a-4e36-86c8-9e0e41697f1c/resourceGroups/
      Kronox_CapacityApisDemo_UsWestCentral/providers/Microsoft.Compute/virtualMachines/Kronox-
      CapacityDemoVm-8"
    ]
  },
  "executionParameters": {
    "RetryPolicy": {
      "RetryCount": 2,
      "RetryWindowInMinutes": 95
    }
  },
  "correlationId": "3fa85f64-5717-4562-b3fc-2c963f66afa6"
}
```

- [ScheduledActions validation](#)
- **subscriptionId**: Client SubscriptionId for the batch of resources.
- **locationparameter**: Enter the known location of the resources.

### Request Body:

- **Resources**: These are the resources to run the executeDeallocate operation on.
  - **Ids**: This is a list of the full Azure resource name. The number of these resources must be less than or equal to 100 and registered under the customer's subscriptionId.
- **ExecutionParameters**: These are parameters the client can add to support the request, this property is optional and can be omitted.
  - **RetryPolicy**: These are values entered by customers that define customizable constraints for retrievable errors, these values are optional and can be omitted, where they are not entered, ScheduledActions will use default values.



- **RetryCount:** The retry count is an integer of how many times ScheduledActions should retry on retrieable errors, if this value is not explicitly set by the client, the default value of 7 is used.
- **RetryWindowInMinutes:** The retry window value, in minutes, that ScheduledActions should keep retrying for retrieable errors, if this value is not explicitly set by the client, the default value of 120 minutes is used.
- **CorrelationId:** Client generated GUID for internal debugging.

#### *VirtualMachinesExecuteDeallocate Response*

```
{
  "description": "Deallocate Resource request",
  "type": "virtualMachinesExecuteDeallocate",
  "location": "westus",
  "results": [
    {
      "resourceId": "subscriptions/afe495ca-b99a-4e36-86c8-9e0e41697f1c/resourceGroups/Kronox_CapacityApisDemo_UsWestCentral/providers/Microsoft.Compute/virtualMachines/Kronox-CapacityDemoVm-7",
      "errorCode": null,
      "errorDetails": null,
      "operation": {
        "operationId": "6561174a-b867-4a89-a6c3-c0d1fc8b7e50",
        "resourceId": "subscriptions/afe495ca-b99a-4e36-86c8-9e0e41697f1c/resourceGroups/Kronox_CapacityApisDemo_UsWestCentral/providers/Microsoft.Compute/virtualMachines/Kronox-CapacityDemoVm-7",
        "opType": "Deallocate",
        "subscriptionId": "700935bc-adf2-4176-b9ad-c571731c09fc",
        "deadline": "2023-12-12T18:17:37.7323681+00:00",
        "deadlineType": "InitiateAt",
        "state": "PendingScheduling",
        "timeZone": "UTC",
        "resourceOperationError": null,
        "completedAt": null,
        "retryPolicy": {
          "retryWindowInMinutes": 95,
          "retryCount": 2,
        },
      },
    }
  ]
}
```

- **Response codes:** Server response code is 200 means execute operation has been successfully registered in the system.
  - In case of error, a HTTP error code of value 400+ will be returned based on the error type (404, 400 etc..).
  - If Scheduling Operations are being blocked a 503 (ScheduledOperationsBlockedException) will be returned with a corresponding Retry-After header.
  - In the case of an outage in Compute for example, ScheduledActions will hold on to operations sent in by the client but not submit them for processing to Compute, however these operations will be marked by ScheduledActions as being in a '**Blocked**' state. If the deadline for the operations pass before the outage is resolved and the jobs can be submitted to Compute, these operations are then marked as **Failed**.

- The results field shows a result of type **DeallocateOperationResponse** containing a description of the operation, the type of the operation, location from the user and a list containing the results from each resource operation.
- **Exceptions:**
  - BadRequestException
  - SubscriptionNotFoundException
  - ServiceUnavailableException
  - ScheduledOperationsBlockedException : Unique object result of structure:
    - Code: SchedulingOperationsBlockedException
    - Message: Scheduling Operations are currently being blocked.
    - Retry-After Header
    - HttpStatusCode : 503

## VirtualMachineExecuteHibernate endpoint

### Endpoint Sample

<https://management.azure.com/subscriptions/{subscriptionId}/providers/Microsoft.ComputeSchedule/locations/{locationparameter}/virtualMachinesExecuteHibernate?api-version=2024-06-01-preview>

### Description:

The VirtualMachinesExecuteHibernate endpoint is used to hibernate a batch of resources, this endpoint allows customers to hibernate resources immediately. For this endpoint, there is no need for the customer to enter schedule information.

### VirtualMachinesExecuteHibernate Request example

```
{
  "resources": {
    "ids": [
      "subscriptions/afe495ca-b99a-4e36-86c8-9e0e41697f1c/resourceGroups/Kronox_CapacityApisDemo_UsWestCentral/providers/Microsoft.Compute/virtualMachines/Kronox-CapacityDemoVm-7"
    ]
  },
  "executionParameters": {
    "RetryPolicy": {
      "RetryCount": 2,
      "RetryWindowInMinutes": 75
    }
  },
  "correlationId": "3fa85f64-5717-4562-b3fc-2c963f66afa6"
}
```

- [ScheduledActions validation](#)
- **subscriptionId**: Client SubscriptionId for the batch of resources.
- **locationparameter**: Enter the known location of the resources.

### Request Body:

- **Resources:** These are the resources to run the executeHibernate operation on.
  - **Ids:** This is a list of the full Azure resource name. The number of these resources must be less than or equal to 100 and registered under the customer's subscriptionId.
- **ExecutionParameters:** These are parameters the client can add to support the request, this property is optional and can be omitted.
  - **RetryPolicy:** These are values entered by customers that define customizable constraints for retrievable errors, these values are optional and can be omitted, where they are not entered, ScheduledActions will use default values.
    - **RetryCount:** The retry count is an integer of how many times ScheduledActions should retry on retrievable errors, if this value is not explicitly set by the client, the default value of 7 is used.
    - **RetryWindowInMinutes:** The retry window value, in minutes, that ScheduledActions should keep retrying for retrievable errors, if this value is not explicitly set by the client, the default value of 120 minutes is used.
- **CorrelationId:** Client generated GUID for internal debugging.

### *VirtualMachinesExecuteHibernate Response*

```
{
  "description": "Hibernate Resource request",
  "type": "virtualMachinesExecuteHibernate",
  "location": "westus",
  "results": [
    {
      "resourceId": "subscriptions/afe495ca-b99a-4e36-86c8-9e0e41697f1c/resourceGroups/Kronox_CapacityApisDemo_UsWestCentral/providers/Microsoft.Compute/virtualMachines/Kronox-CapacityDemoVm-8",
      "errorCode": null,
      "errorDetails": null,
      "operation": {
        "operationId": "6905e045-b398-412a-87d3-f69ab96987bc",
        "resourceId": "subscriptions/afe495ca-b99a-4e36-86c8-9e0e41697f1c/resourceGroups/Kronox_CapacityApisDemo_UsWestCentral/providers/Microsoft.Compute/virtualMachines/Kronox-CapacityDemoVm-8",
        "opType": "Hibernate",
        "subscriptionId": "700935bc-adf2-4176-b9ad-c571731c09fc",
        "deadline": "2023-12-12T18:26:46.3921679+00:00",
        "deadlineType": "InitiateAt",
        "state": "PendingScheduling",
        "timeZone": "UTC",
        "resourceOperationError": null,
        "completedAt": null,
        "retryPolicy": {
          "retryWindowInMinutes": 75,
          "retryCount": 3,
        },
      },
    }
  ]
}
```

- **Response codes:** Server response code is 200 means execute operation has been successfully registered in the system.

- In case of error, a HTTP error code of value 400+ will be returned based on the error type (404, 400 etc..).
- If Scheduling Operations are being blocked a 503 (ScheduledOperationsBlockedException) will be returned with a corresponding Retry-After header.
- In the case of an outage in Compute for example, ScheduledActions will hold on to operations sent in by the client but not submit them for processing to Compute, however these operations will be marked by ScheduledActions as being in a '**Blocked**' state. If the deadline for the operations pass before the outage is resolved and the jobs can be submitted to Compute, these operations are then marked as **Failed**.
- The results field shows a result of type **HibernateOperationResponse** containing a description of the operation, the type of the operation, location from the user and a list containing the results from each resource operation.
- **Exceptions:**
  - BadRequestException
  - SubscriptionNotFoundException
  - ServiceUnavailableException
  - ScheduledOperationsBlockedException : Unique object result of structure:
    - Code: SchedulingOperationsBlockedException
    - Message: Scheduling Operations are currently being blocked.
    - Retry-After Header
    - HttpStatusCode : 503

## VirtualMachinesGetOperationStatus endpoint

### Endpoint Sample

<https://management.azure.com/subscriptions/{subscriptionId}/providers/Microsoft.ComputeSchedule/locations/{locationparameter}/virtualMachinesGetOperationStatus?api-version=2024-06-01-preview>

### Description

The VirtualMachinesGetOperationStatus endpoint is used to query operations status on virtual machines by their operationIds.

### VirtualMachinesGetOperationStatus Request example

```
{
  "operationIds": [
    "d204a73d-7c3a-4469-9eba-4fc880b00d47", "9a51d5df-23a1-4aa9-aa42-e4baef9e0f64"
  ],
  "correlationId": "3fa85f64-5717-4562-b3fc-2c963f66afa6"
}
```

- [ScheduledActions validation](#)
- **subscriptionId**: Client SubscriptionId for the batch of resources.
- **locationparameter**: Enter the known location of the resources.

#### Request Body:

- **operationIds**: A list of operationIds for the resources started/hibernated/deallocated. The number of operationIds per request must be less than or equal to 100.
- **CorrelationId**: Client generated GUID for internal debugging.

#### *VirtualMachinesGetOperationStatus Response*

```
{
  "results": [
    {
      "resourceId": "subscriptions/afe495ca-b99a-4e36-86c8-9e0e41697f1c/resourceGroups/Kronox_CapacityApisDemo_UsWestCentral/providers/Microsoft.Compute/virtualMachines/Kronox-CapacityDemoVm-3",
      "errorCode": null,
      "errorDetails": null,
      "operation": {
        "operationId": "d204a73d-7c3a-4469-9eba-4fc880b00d47",
        "resourceId": "subscriptions/afe495ca-b99a-4e36-86c8-9e0e41697f1c/resourceGroups/Kronox_CapacityApisDemo_UsWestCentral/providers/Microsoft.Compute/virtualMachines/Kronox-CapacityDemoVm-3",
        "opType": "Deallocate",
        "subscriptionId": "700935bc-adf2-4176-b9ad-c571731c09fc",
        "deadline": "2023-12-12T19:13:52.067+00:00",
        "deadlineType": "InitiateAt",
        "state": "PendingScheduling",
        "timeZone": "",
        "resourceOperationError": null,
        "completedAt": null,
        "retryPolicy": {
          "retryWindowInMinutes": 50,
          "retryCount": 7,
        },
      },
    },
    {
      "resourceId": "subscriptions/afe495ca-b99a-4e36-86c8-9e0e41697f1c/resourceGroups/Kronox_CapacityApisDemo_UsWestCentral/providers/Microsoft.Compute/virtualMachines/Kronox-CapacityDemoVm-2",
      "errorCode": null,
      "errorDetails": null,
      "operation": {
        "operationId": "9a51d5df-23a1-4aa9-aa42-e4baef9e0f64",
        "resourceId": "subscriptions/afe495ca-b99a-4e36-86c8-9e0e41697f1c/resourceGroups/Kronox_CapacityApisDemo_UsWestCentral/providers/Microsoft.Compute/virtualMachines/Kronox-CapacityDemoVm-2",
        "opType": "Start",
        "subscriptionId": "700935bc-adf2-4176-b9ad-c571731c09fc",
        "deadline": "2023-12-12T19:01:10.872+00:00",
        "deadlineType": "InitiateAt",
        "state": "PendingScheduling",
        "timeZone": "",
        "resourceOperationError": null,
        "completedAt": null,
        "retryPolicy": {
          "retryWindowInMinutes": 45,
          "retryCount": 7,
        },
      },
    },
  ]
}
```

- **Response codes**: Server response code is 200 means get operation status operation has been successfully registered in the system.
  - In case of error, a HTTP error code of value 400+ will be returned based on the error type (404, 400 etc..).

- If Scheduling Operations are being blocked a 503 (ScheduledOperationsBlockedException) will be returned with a corresponding Retry-After header.
- There are specific scenarios that operation can run into when it is sent to our dependency service, in which case error object would be populated with the relevant error code.
  - AllocationFailure
  - FeatureNotEnabledForSubscriptionException
- The results field shows a list of type ResourceOperation for each operationId associated with a virtual machine.
- **Exceptions:**
  - BadRequestException.
  - SubscriptionNotFoundException.
  - ServiceUnavailableException
  - Non-ScheduledOperationsBlockedException : Unique object result of structure:
    - Code: NonSchedulingOperationsBlockedException
    - Message: Non-Scheduling Operations are currently being blocked.
    - Retry-After Header
    - HttpStatusCode : 503

## VirtualMachinesCancelOperations endpoint

### Endpoint Sample

<https://management.azure.com/subscriptions/{subscriptionId}/providers/Microsoft.ComputeSchedule/locations/{locationparameter}/virtualMachinesCancelOperations?api-version=2024-06-01-preview>

### Description

The VirtualMachinesCancelOperations is used to delete any future operations against the given batch of resources. It only cancels operations that have not been scheduled in Azure.

### VirtualMachinesCancelOperations Request example

```
{
  "operationIds": [
    "d204a73d-7c3a-4469-9eba-4fc880b00d47", "9a51d5df-23a1-4aa9-aa42-e4baef9e0f64"
  ],
  "correlationId": "3fa85f64-5717-4562-b3fc-2c963f66afa6"
}
```

- [ScheduledActions validation](#)
- **subscriptionId:** Client SubscriptionId for the batch of resources.
- **locationparameter:** Enter the known location of the resources.

### Request Body:

- **operationIds:** A list of operationIds for resource operations the user wants to cancel. The number of operationIds per request must be less than or equal to 100.
- **CorrelationId:** Client generated GUID for debugging.

### *VirtualMachinesCancelOperations Responseexample*

```
{
  "results": [
    {
      "resourceId": "subscriptions/afe495ca-b99a-4e36-86c8-9e0e41697f1c/resourceGroups/Kronox_CapacityApisDemo_UsWestCentral/providers/Microsoft.Compute/virtualMachines/Kronox-CapacityDemoVm-9",
      "errorCode": null,
      "errorDetails": null,
      "operation": {
        "operationId": "b5368034-df16-4ac1-8360-ba7d271a34de",
        "resourceId": "subscriptions/afe495ca-b99a-4e36-86c8-9e0e41697f1c/resourceGroups/Kronox_CapacityApisDemo_UsWestCentral/providers/Microsoft.Compute/virtualMachines/Kronox-CapacityDemoVm-9",
        "opType": "Hibernate",
        "subscriptionId": "700935bc-adf2-4176-b9ad-c571731c09fc",
        "deadline": "2023-12-13T22:25:31.719+00:00",
        "deadlineType": "InitiateAt",
        "state": "OperationCancelled",
        "timeZone": "",
        "resourceOperationError": {
          "errorCode": "OperationCancelled",
          "errorDetails": "Operation b5368034-df16-4ac1-8360-ba7d271a34de was cancelled by user"
        },
        "completedAt": null,
        "retryPolicy": {
          "retryWindowInMinutes": 70,
          "retryCount": 5,
        },
      },
    },
    {
      "resourceId": "subscriptions/afe495ca-b99a-4e36-86c8-9e0e41697f1c/resourceGroups/Kronox_CapacityApisDemo_UsWestCentral/providers/Microsoft.Compute/virtualMachines/Kronox-CapacityDemoVm-10",
      "errorCode": null,
      "errorDetails": null,
      "operation": {
        "operationId": "e528fddd-d9a3-41c7-8802-97cf7acea0b6",
        "resourceId": "subscriptions/afe495ca-b99a-4e36-86c8-9e0e41697f1c/resourceGroups/Kronox_CapacityApisDemo_UsWestCentral/providers/Microsoft.Compute/virtualMachines/Kronox-CapacityDemoVm-10",
        "opType": "Start",
        "subscriptionId": "700935bc-adf2-4176-b9ad-c571731c09fc",
        "deadline": "2023-12-13T22:29:14.839+00:00",
        "deadlineType": "InitiateAt",
        "state": "OperationCancelled",
        "timeZone": "",
        "resourceOperationError": {
          "errorCode": "OperationCancelled",
          "errorDetails": "Operation e528fddd-d9a3-41c7-8802-97cf7acea0b6 was cancelled by user"
        },
        "completedAt": null,
        "retryPolicy": {
          "retryWindowInMinutes": 50,
          "retryCount": 7,
        },
      },
    },
  ]
}
```

- **Response codes:** Server response code is 200 means cancelling operation has been successfully registered in the system.
  - In case of error, a HTTP error code of value 400+ will be returned based on the error type (404, 400 etc..).
  - If Scheduling Operations are being blocked a 503 (ScheduledOperationsBlockedException) will be returned with a corresponding Retry-After header.

- If Non-Scheduling Operations are being blocked a 503 will be returned with a corresponding Retry-After header.
- Note: If an operation is in an activated state, this will be a best effort case, we will try to cancel it but if its already executed by downstream Compute service, then cancellation won't work.
- The results field shows a list of type ResourceOperation for each operationId associated with a virtual machine.
- **Exceptions:**
  - BadRequestException.
  - SubscriptionNotFoundException.
  - ServiceUnavailableException
  - Non-ScheduledOperationsBlockedException : Unique object result of structure:
    - Code: NonSchedulingOperationsBlockedException
    - Message: Non-Scheduling Operations are currently being blocked.
    - Retry-After Header
    - HttpStatusCode : 503

## Parsing Responses

The following section explains the results from using Scheduledactions.

```
{
  "description": "Hibernate Resource request",
  "type": "virtualMachinesExecuteHibernate",
  "location": "westus",
  "results": [
    {
      "resourceId": "subscriptions/afe495ca-b99a-4e36-86c8-9e0e41697f1c/resourceGroups/Kronox_CapacityApisDemo_UsWestCentral/providers/Microsoft.Compute/virtualMachines/Kronox-CapacityDemoVm-8",
      "errorCode": null,
      "errorDetails": null,
      "operation": {
        "operationId": "6905e045-b398-412a-87d3-f69ab96987bc",
        "resourceId": "subscriptions/afe495ca-b99a-4e36-86c8-9e0e41697f1c/resourceGroups/Kronox_CapacityApisDemo_UsWestCentral/providers/Microsoft.Compute/virtualMachines/Kronox-CapacityDemoVm-8",
        "opType": "Hibernate",
        "subscriptionId": "700935bc-adf2-4176-b9ad-c571731c09fc",
        "deadline": "2023-12-12T18:26:46.3921679+00:00",
        "deadlineType": "InitiateAt",
        "state": "PendingScheduling",
        "timeZone": "UTC",
        "resourceOperationError": null,
        "completedAt": null,
        "retryPolicy": {
          "retryWindowInMinutes": 75,
          "retryCount": 3,
        },
      },
    }
  ]
}
```



Item	Explanation
Description	The type of operation that was performed on the virtual machines, is Hibernate, Start, Deallocate
Type	The Scheduledactions resource type
Location	The location of the virtual machines which is also the location the scheduledactions URL was called from
Results	The result object is a list containing the results of the start/hibernate/deallocate request on the batch of virtual machines, each item in the results list is a resourceOperations.
ResourceId	The fully qualified Azure Id of the virtual machine
ErrorCode	This is the top-level error code that will be populated if there is an issue with the schedule request eg validation errors. When this is populated, the request operation on the virtual machine was not fulfilled
ErrorDetails	This is the top-level error details explaining the top-level error code
ResourceId	The fully qualified Azure Id of the virtual machine
Operation	The operation object holds more details about the operation performed on the virtual machine
OperationId	Unique GUID which is used to track the status of the operation using the VirtualMachinesGetOperationStatus endpoint.
OpType	The type of operation performed on the virtual machine, it could either be Start/Deallocate/Hibernate
SubscriptionId	The subscripitionId associated with the virtual machine
Deadline	The deadline for the requested operation in UTC
DeadlineType	This defines the deadline type of the operation, InitiateAt
State	This describes the current state of the operation performed on the virtual machine. See more details on state definitions <a href="#">here</a>
ResourceOperationError	This object holds errors related to the unique operation on a virtual machine. A null value means there was no error on the operation and the start/deallocate/hibernate operation was processed. The rows below explain what populated fields for this object means
ResourceOperationError.ErrorCode	This is the operation-level error code that will be populated when there is an issue with the operation on the virtual machine, eg OperationConflict exception. When this is populated, the request operation on the virtual machine was attempted
ResourceOperationError.ErrorDetails	This is the operation -level error details explaining the operation -level error code

RetryPolicy	This object holds the values used in the request for the retryCount and RetryWindowInMinutes
-------------	--

## Contract type Description

Endpoint	Contract
VirtualMachinesSubmitStart VirtualMachinesExecuteStart	StartResourceOperationResponse
VirtualMachinesSubmitDeallocate VirtualMachinesExecuteDeallocate	DeallocateResourceOperationResponse
VirtualMachinesSubmitHibernate VirtualMachinesExecuteHibernate	HibernateResourceOperationResponse
VirtualMachinesGetOperationStatus	GetOperationStatusRequest GetOperationStatusResponse
VirtualMachinesCancelOperations	CancelOperationsRequest CancelOperationsResponse

- **ErrorCodes:**
  - VmNotFound: Id field is populated with the invalid VM ID.
  - OperationNotFound: OperationId field is populated with the operation ID that doesn't exist for subscription.
  - OperationConflict: OperationId field is populated with the operation ID of an already scheduled operation.
- Resource operation details:
  - OpType:
    - Unknown
    - Start
    - Deallocate
    - Hibernate.
  - subscriptionId: The subscription id.
  - resourceId: The full name of the Azure resource ie the ArmID.
  - deadlineUtc: The deadline for the operation in UTC.
  - deadlineType:
    - Unknow
    - InitiateAt
  - State:
    - Unknown: Not Allowed
    - PendingScheduling: Operation has been sent to ScheduledActions, but not yet scheduled
    - Scheduled: Operation has been scheduled for execution in Scheduledactions

- PendingExecution: Operation has been scheduled and is waiting to be picked up by execution engine
- Executing: Operation is currently being executed
- Succeeded: Operation has succeeded, and virtual machine is in desired state
- Failed: Operation has failed, and virtual machine is not in desired state
- Cancelled: Operation has been cancelled
- Blocked: Operation is blocked due to any number of reasons from outage to error surge
- Error:
  - errorCode: Error code from dependency service (operation processing failure at ScheduledActions or partners).
  - errorDetails: Error message from dependency service (operation processing failure at ScheduledActions or partners).
- CompletedAt: The DateTime in UTC that the operation is completed at.
- ResourceOperationError: Error at the virtual machine operation level eg OperationConflict

## Miscellaneous

### Recommended Usage

Below are some best practices to follow when using Scheduledactions:

1. Leverage the batching feature as much as possible to minimize throttling at ARM level for submitting operations and checking the operations statuses. Max 100 per request
2. To further prevent throttling, when repeatedly polling for operation status on virtual machines, we recommend not polling on operations in a terminal state, that is not polling operations where Operation.State is Succeeded, Failed or Cancelled.
  - a. **Example:** If 100 operations, sent in a single request are being polled on, and the initial poll result returns 39 in a terminal state, the next polling request should be for the 61 that are not yet in a terminal state
3. For each request, we recommend parsing the results as follows:
  - a. Check the top-level error code for request level errors such as validation errors or errors from partner services, if this is not null, an error has occurred at the request level and the schedule request was not processed
  - b. If the top-level error is null, proceed to check Operation.State
    - i. If Operation.State is Successful, then the Start/Deallocate/Hibernate operation has succeeded, and the virtual machine is in the desired state. In this case the ResourceOperationError will be null
    - ii. If Operation.State is Failed, then the operation on the virtual machine has failed and the ResourceOperationError will be populated with the reason for the failure
    - iii. If Operation.State is Cancelled, then the operation on the virtual machine was cancelled and the ResourceOperationError will be populated with the cancellation details

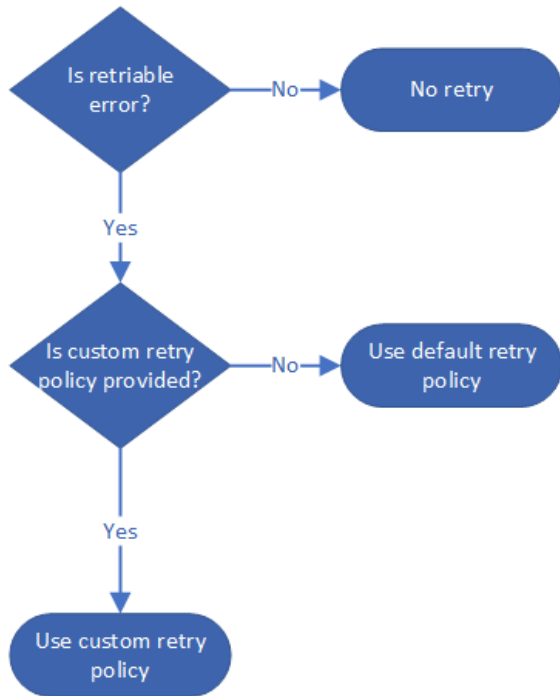
## Throttler Quotas

Below are the quotas for operations in ScheduledActions. We recommend using the batching feature that Scheduledactions provides, that is, sending operations in batches to avoid ARM throttling.

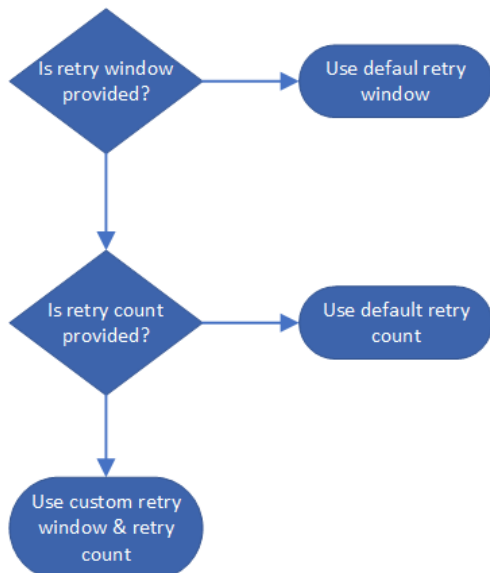
Operation	TimeWindow	RequestLimit (updated 17/1)
<div>Schedule Operations:<ul style="list-style-type: none"><li>VirtualMachinesSubmitDeallocate</li><li>VirtualMachinesSubmitStart</li><li>VirtualMachinesSubmitHibernate</li><li>VirtualMachinesExecuteHibernate</li><li>VirtualMachinesExecuteStart</li><li>VirtualMachinesExecuteDeallocate</li></ul> (NB: VirtualMachinesSubmitDeallocate, VirtualMachinesSubmitStart, VirtualMachinesSubmitHibernate, VirtualMachinesExecuteStart, VirtualMachinesExecuteDeallocate, VirtualMachinesExecuteHibernate have a combined quota)</div>	60 seconds	10,000 vmId/operation Ids
VirtualMachinesCancelOperations	60 seconds	10,000 vmId/operation Ids
VirtualMachinesGetOperationStatus	60 seconds	35,000 vmId/operation Ids

## Retry Policy

RetryPolicy is one of the fields in the ExecutionParameters, customers can pass RetryPolicy to control the retry mechanism of the operation. Keep in mind however, that the retry policy is a best effort retry. ScheduledActions will internally identify if the error is retryable, if yes, the custom retry policy will be used for retrying. At any point, if a non-retryable error is received, the retry mechanism will stop to conserve resources.



For the retry policy, the user can pass either retry count or retry window or both. If the retry window or the retry count is not provided, the default value will be used.



When does the retry stop?

After the retry window and retry count is set (whether by default or custom), our service will stop retrying when either one of the two conditions is met.

**For example:** If the retry window is 30min, and retry count is 3. We will stop retrying when the retry duration exceeds 30min or the retry count reaches 3. Refer to the table below on the results on each condition.

Custom retry count	Custom retry window in minutes	Final retry policy	Stop retry condition
3	30	3 / 30	Either retry attempts reach 3 or retry duration exceed 30 min
N/A	30	7(default) / 30	Either retry attempts reach 7 or retry duration exceed 30 min
3	N/A	3 / 90(default)	Either retry attempts reach 3 or retry duration exceed 120min
N/A	N/A	7 / 90	Either retry attempts reach 7 or retry duration exceed 120min