Public API – ScheduledActions customers

Table of Contents

Troubleshooting guide for Scheduled Actions API

Welcome to ScheduleActions troubleshooting. These article explains how to determine, diagnose, and fix issues that you might encounter when you use ScheduleActions API.

Background information and Context

SLAs for ScheduledActions

- For Start/Stop/Hibernate scenario, ScheduledActions will match the current Azure SLAs.
- Future versions of ScheduledActions will improve both reliability and performance.
- ScheduledActions is currently certified for 5k concurrent operations in less than 13 minutes in a subscription (this time may be more than 13 minutes based on number and type of extensions on a VM) the team is also working on adding support for 20k concurrent operations per subscription.

API functions

- Batch API to schedule operations on resources at scale. The current batch maximum is 100
- Schedule operations (Start/Deallocate/Hibernate) against a batch of VMs within a subscription
- Query status (Scheduled/Activated/Completed) of scheduled operations and their error info (if any).
- Cancel scheduled but not yet activated Operations.
- Current supported resource types are Virtual Machines

API validations

- Resource number validation: Limit of Resources per request is set to **100**.
- Operation conflicting validation: Deadlines of any two **pending** operations on the same Resource must be more than 1 hour apart from each other. E.g., if an operation has a deadline of 9AM, any **pending** operation between 8-10 on the same Resource would create a conflict. Once an operation is completed it is no longer considered in conflict validation
- Deadlines are out of the expected range (not more than 14 days in future and not more than 5 minutes in past).

  .

Limitations

Regional limitations

Public cloud only

Power management operations supported

1. Start
2. Deallocate
3. Hibernate-Resume

API Overview

Below are the operations that can be performed in ScheduledActions

API	Endpoint	Http Method
Schedule Virtual Machine Start at datetime in future	VirtualMachinesSubmitStart endpoint	POST
Schedule Virtual Machine Deallocate at datetime in future	VirtualMachinesSubmitDeallocate endpoint	POST
Schedule Virtual Machine Hibernate at datetime in future	VirtualMachinesSubmitHibernate endpoint	POST
Execute Virtual Machine Start (triggers operation as soon as ScheduledActions receives the request)	VirtualMachinesExecuteStart endpoint	POST
Execute Virtual Machine Deallocate (triggers operation as soon as ScheduledActions receives the request)	VirtualMachinesExecuteDeallocate endpoint	POST
Execute Virtual Machine Hibernate (triggers operation as soon as ScheduledActions receives the request)	VirtualMachinesExecuteHibernate endpoint	POST
Polling end point to read status of operations	VirtualMachinesGetOperationStatus endpoint	POST
Cancel a previously submitted (start/deallocate/hibernate) request	VirtualMachinesCancelOperations endpoint	POST
For additional details on operation errors (like transient errors encountered, additional logs)	VirtualMachinesGetOperationErrors endpoint	POST

Schedule and Execute operation Errors

Using these APIs clients can schedule an operation to be executed at a future datetime or trigger an operation right away, in this section we list error responses that can be returned for start/deallocate/hibernate APIs

Schedule Virtual Machine Starts API

Schedule Virtual Machines Deallocate API

Schedule Virtual Machines Hibernate API

Execute VirtualMachines Start API

Execute VirtualMachines Deallocate API

Execute VirtualMachines Hibernate API

Response codes

- Server response code is 200 means scheduling operation has been successfully registered in the system.
    - In case of error, an HTTP error code of value 400+ will be returned based on the error type (404, 400 etc..). This could mean a validation or throttling error
    - 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**. ScheduledActions is a regional service, so an outage in region A only blocks operations in region A.
    - If Scheduling Operations are blocked a 503 (ScheduledOperationsBlockedException) will be returned with a corresponding Retry-After header.

Exceptions

- BadRequestException
- ServiceUnavailableException
- ScheduledOperationsBlockedException : Unique object result of structure:
    - Code: SchedulingOperationsBlockedException
    - Message: Scheduling Operations are currently being blocked.
    - Retry-After Header
    - HttpStatusCode : 503

Read Operation Status APIs

Get Operation Status API

Description

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

Endpoint Sample

https://management.azure.com/subscriptions/{subscriptionId}/providers/Microsoft.ComputeSchedule/locations/{locationparameter}/virtualMachinesGetOperationStatus?api-version=2024-08-15-preview

Response codes

Server response code is 200 means get operation status operation has been successfully registered in the system.

- In case of error, an HTTP error code of value 400+ will be returned based on the error type (404, 400 etc..).The results field shows a list of type ResourceOperation for each operationId associated with a virtual machine.

Exceptions

- BadRequestException.
- ServiceUnavailableException
- Non-ScheduledOperationsBlockedException : Unique object result of structure:
    - Code: NonSchedulingOperationsBlockedException
    - Message: Non-Scheduling Operations are currently being blocked.
    - Retry-After Header
    - HttpStatusCode : 503

Cancel an already Scheduled Operation API

Description

The VirtualMachinesCancelOperations is used to cancel any future operations against the given batch of resources. It only cancels operations that haven't been triggered.

Endpoint Sample

https://management.azure.com/subscriptions/{subscriptionId}/providers/Microsoft.ComputeSchedule/locations/{locationparameter}/virtualMachinesCancelOperations?api-version=2024-08-15-preview

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: This is best effort case, we will try to cancel it but if its already executing 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.

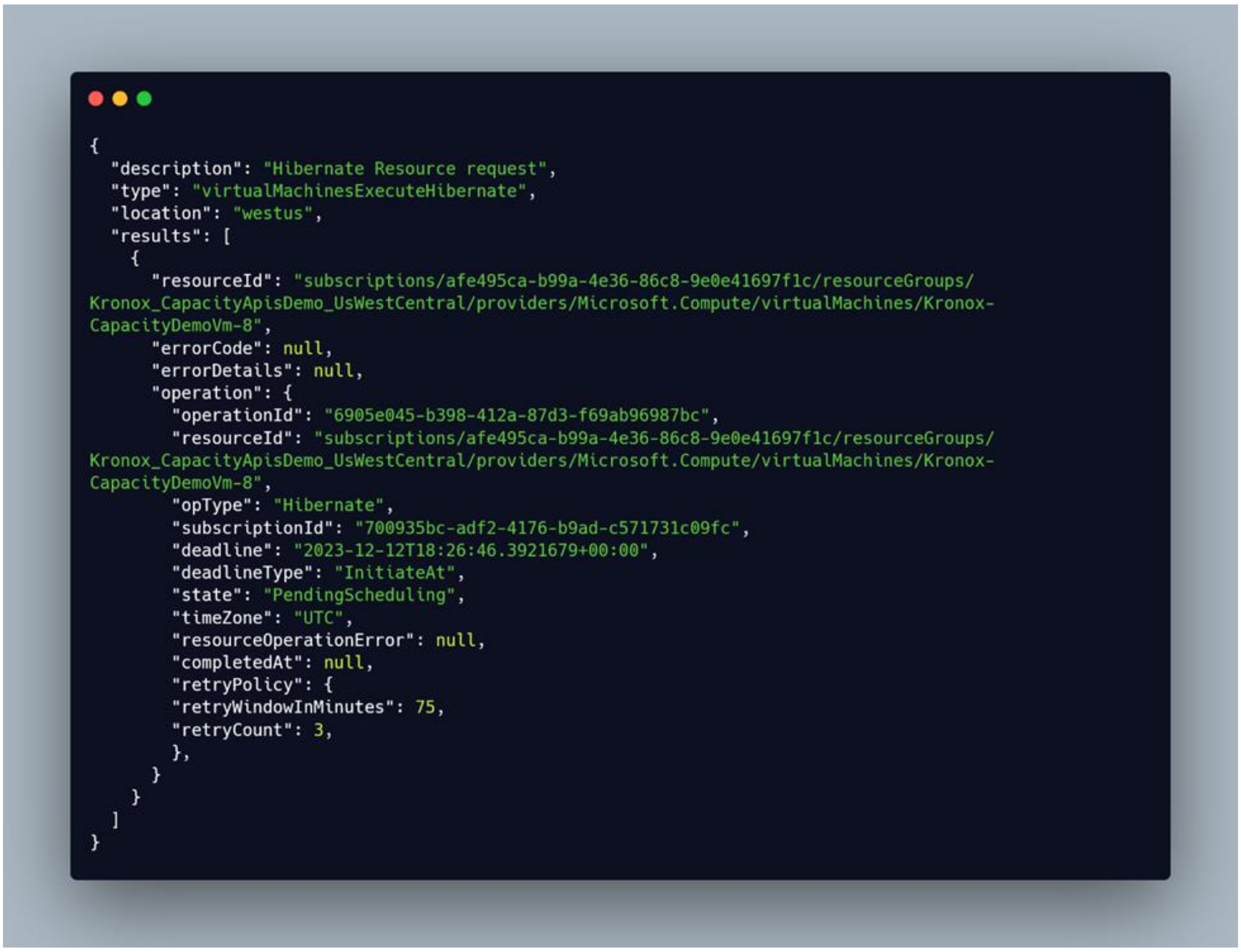
Exceptions

- BadRequestException.
- ServiceUnavailableException
- Non-ScheduledOperationsBlockedException : Unique object result of structure:
    - Code: NonSchedulingOperationsBlockedException
    - Message: Non-Scheduling Operations are currently being blocked.
    - Retry-After Header
    - HttpStatusCode : 503

API response deep dive

How to Parse Get Operation Status Response

For reading status of your submitted (start/deallocate/hibernate) requests you will be using GetOperationStatus endpoint, and we will describe in this section how to read response to best equip you to understand and troubleshoot issues



```
{
  "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

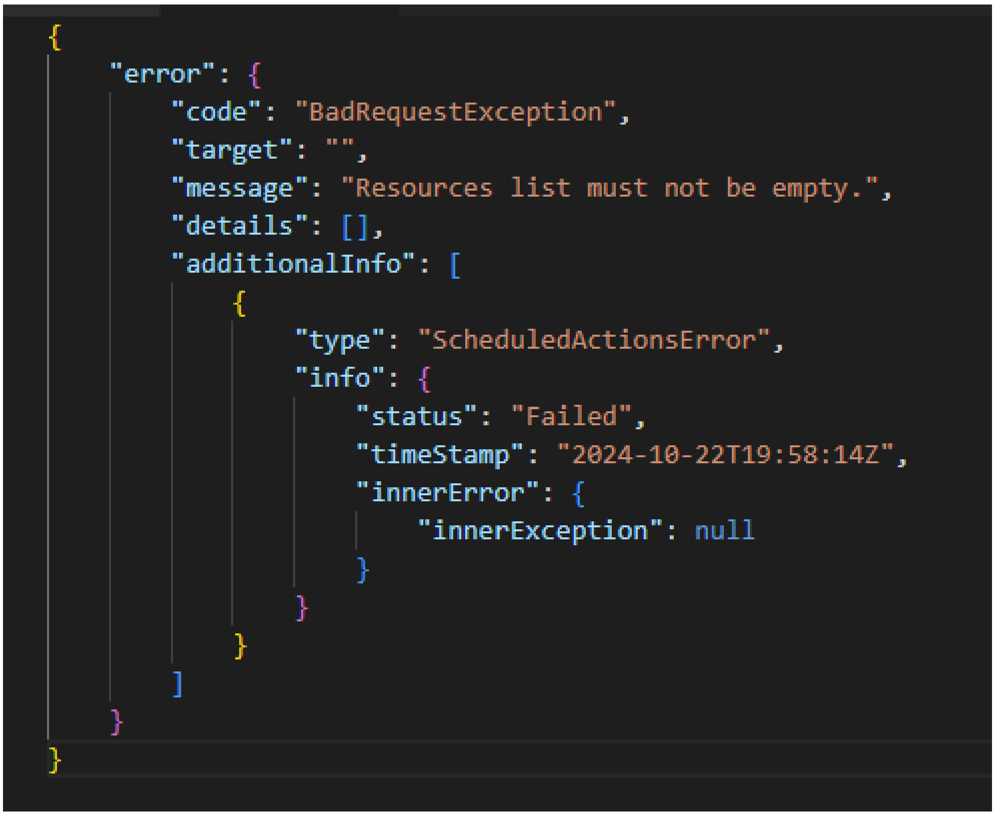
	this is populated, the request operation on the virtual machine was not fulfilled
ErrorDetails	This is the top-level error details explaining the top-level error code
ResourceId	The fully qualified Azure Id of the virtual machine
Operation	The operation object holds more details about the operation performed 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 subscripitonId 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 here
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, AllocationFailure,VMStartTimedOut or other networking or disk related exception. – Common virtual machine error codes in Azure - Virtual Machines | Microsoft Learn
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

Error information in response JSON can be found at 3 levels, these aren't duplicated but depending on the error-type error information will be populated.

1) Top level errors – this error is at request level and in this case operation requests on all VMs are rejected, these errors would fall into following three categories
   a. Validation errors
   b. Throttling errors
   c. Outage

   These errors are at request level, for example deadlines are too far out in future or going over throttling limits, in this case ScheduledActions will reject all the resources in the requests and expect client to send a modified request based on action item below. In this case there won't be an array of results in response.

Sample JSON response for top level error



```
{
    "error": {
        "code": "BadRequestException",
        "target": "",
        "message": "Resources list must not be empty.",
        "details": [],
        "additionalInfo": [
            {
                "type": "ScheduledActionsError",
                "info": {
                    "status": "Failed",
                    "timeStamp": "2024-10-22T19:58:14Z",
                    "innerError": {
                        "innerException": null
                    }
                }
            }
        ]
    }
}
```

ResultCode	**Error Message**	**Http StatusCode**	**Action**
Validation			
BadRequestException	The request deadline is too far out in future. Please limit it to within 14 days	400	Send a request within 14 days from now
	The request deadline is too far in past. Please limit it to within 5 minutes.	400	Consider using Execute API to immediately trigger an operation
	Resources list must not be empty.	400	Ensure the VMs value of the request is populated with at least 1 VM
	Too many VMs. Requests are allowed to have up to 100 VMs.	400	Ensure the VMs value of the request is populated with no more than a 100 VMs
	Invalid DeadlineType: Unknown	400	Supply a valid deadline type of InitiateAt
	Initiate At operations cannot be completed with Optimization preferences	400	Remove optimization preference from request
	Scheduled Actions support UTC timezones only.	400	Resubmit request with UTC
	Retry window should be within range	400	retry policy

	Retry count should be within range	400	retry policy
Outage			
SchedulingOperationsBlockedException	Scheduling Operations are currently being blocked. This happens in case of an Azure outage in a region	503	Wait for the Retry-After period to elapse before attempting to perform operation again. Keep in mind it could fail again if the outage is still on
NonSchedulingOperationsBlockedException	Non-Scheduling Operations are currently being blocked.	503	Wait for the Retry-After period to elapse before attempting to perform operation again. Keep in mind it could fail again if the outage is still on
Throttling			
TooManyRequestsException	Request limit has been reached	429	Wait for the Retry-After period to elapse before attempting to perform operation again.

2) *Resource specific errors (bottom level errors)*

These error codes are per resource, after request passes basic validation (deadline, number of resources, max and min limits for Retry policy etc.) platform will begin second phase of validation where each resource in the batch will be checked. In this case errorCode and errorDetails field in the results array will be populated for each resource that fail validation. These failures are per resource and independent of each other for e.g. a client sends a request with 10 VMs and there is another operation already scheduled on 2 out of these 10 VMs then ScheduledActions will proceed with 8 VMs and return “OperationConflict” for those 2 VMs.

In this case no operation will be created for resources that failed.

a. ResourceOperationError – this is bottom most level of errors, when request is valid, all the resources in the request also pass validation, now the platform will start executing these operations and send them to Azure Compute RP, if an operation can’t be completed by the Compute stack because of reasons like low capacity or networking or extension related failure down the stack, in this case ScheduledAction will mark the operation as failed and it will populate ResourceOperationErrorCode and ResourceOperationErrorDetails in ResourceOperationError JSON object. these errors are beyond the scope of ScheduleActions platform, for their mitigation we will leverage existing Azure document on common error

Vm Operation - Error Code	Error Message	Action
ScheduleActions Error codes		

Conflict	operation {op.Id} on {op.VmId} is in conflict with an existing Op	Cancel existing operation using Cancel API on this VM before scheduling a new request, or send this scheduling request after current operation is complete
OperationTimedOutException	Operation {opId} timed out within scheduledActions. The operation may still complete in CRP.	
OperationExpiredDuringOutage	Operation: 81d0e24c-568b-4397-a370-d907e53a8d74 was cancelled because it expired during an outage.	Send a new request for the VM, once the outage is over
Azure control plane Error codes *any operation error that is outside 4 mentioned above (Conflict,OperationTimedOut,FeatureNotEnabled,OperationExipredDuringOutage) are originating from compute stack, it could come from network, disk, extensions, capacity and various other sources, these errors are outside the scope of scheduledActions troubleshooting guide and we should follow existing Azure virtual machines troubleshooting guide for addressing those.*		
Examples OperationNotAllowed, AllocationFailed, VMStartTimeOut etc.		Common virtual machine error codes in Azure - Virtual Machines | Microsoft Learn

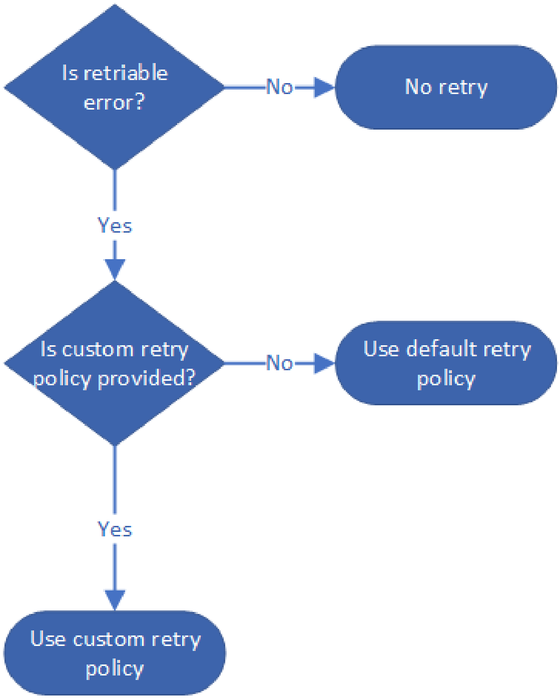
Throttling limits

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.

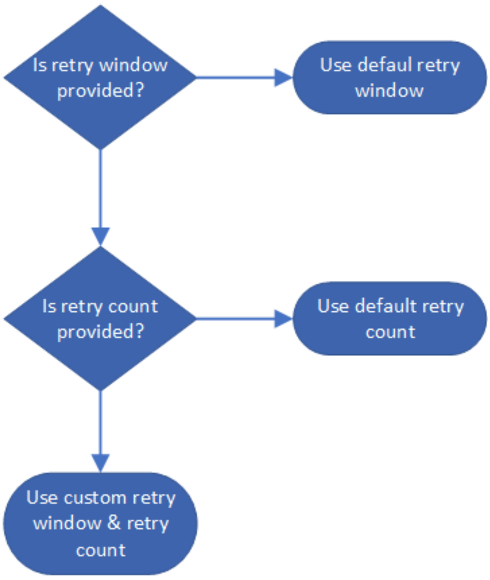
Operation	TimeWindow	RequestLimit (updated 17/1)
Schedule Operations: • Schedule VirtualMachines Deallocate • Schedule VirtualMachines Start • Schedule VirtualMachines Hibernate • Execute VirtualMachines Hibernate • Execute VirtualMachines Start • Execute VirtualMachines Deallocate (**NB**: Schedule VirtualMachines Deallocate Schedule VirtualMachines Start Schedule VirtualMachines Hibernate Execute VirtualMachines Hibernate Execute VirtualMachines Start Execute VirtualMachines Deallocate have a combined quota)	60 seconds	10,000 Resource Id/operation Ids
Cancel Scheduled Operation API	60 seconds	10,000 Resource Id/operation Ids
Get Operation Status Get Operation Error Details	60 seconds	35,000 Resource Id/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 retriable, if yes, the custom retry policy will be used for retrying. At any point, if a non-retriable 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 90min
N/A	N/A	7 / 90	Either retry attempts reach 7 or retry duration exceed 90min

Scheduled Actions key definitions and contracts

Blocking

In the event of a service outage, either within scheduledActions platform or external service that scheduledActions is dependent on, scheduledActions will accrue operation requests, that will compound pressure on an already lagging platform and create a cascading effect of failures, not allowing service to recover. In order to shield platform such cascading failures, scheduledActions has built a protective mechanism that will used to stop accepting any new requests while we detect a regression. This blocking is manually enabled by DRI after we have confirmed that this could be an outage that will take time to resolve. When a customer tries to execute a start/stop or hibernate operation while this outage is enabled, clients will see a 503-http response error along with retry-after header. This retry-after header is our best estimate when we expect regression to be mitigates and service can return to BAU.

ScheduledOperations blocking – start/stop/hibernate and cancel operations request will be blocked with 503 errors, while clients can still poll for operation status by using GetOperation status and Get error details endpoints.

Non-SchedulingOperation blocking – During this type of blocking, read operations are blocked, clients will get 503 when they poll for operation status by using GetOperation status and Get error details endpoints

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:

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

CompletedAt: The DateTime in UTC that the operation is completed at.