Mandiant

ADD CLOUD ACTIONS

Prerequisites and Important Tips

- When writing a Cloud Action script
  - The main code of the script must be at the top level. Don't use the typical Python idiom `if __name__ == '__main__':` because the code isn't executed as a main module.
  - Spacing matters. If the appropriate lines aren't indented, according to correct Python syntax, the Action cannot run successfully.
- Cloud Actions can include files, but you are not allowed to use malicious files.
- There is a configurable timeout on script execution; the default is 60 seconds and that should be the minimum value for any scripts that you configure. If a Python script takes longer than the timeout to execute, the Action results in an error.
- Similar to Host CLI Actions, Cloud Actions have an approval process associated with them.

- For the installer-based Linux Actor on RHEL7 only, the following repositories must be enabled before you install the Actor (or for upgrades, before upgrading) to be able to run Cloud Actions:

```
subscription-manager repos --enable=rhel-7-server-rpms
subscription-manager repos --enable=rhel-7-server-extras-rpms
subscription-manager repos --enable=rhel-7-server-optional-rpms
```

- This requirement doesn't apply to other supported Linux Actor platforms.

Scripts

- Cloud Actions can contain multiple scripts in a single Action.
- All scripts run on the same Actor, but each can be run with a different set of credentials/cloud profiles.
- Scripts can have special types:
  - **Setup**: You can use the first script within the Action as a Setup Script (for example, for setting up a storage device or an environment with specific permissions within the Cloud).
  - **Cleanup**: The last script within the Action can be used as a Cleanup Script (for example, for tearing down whatever environment has been set up in the Cloud).

- If any script fails, the cleanup script still runs. If there are variables that are needed in the cleanup script, we strongly recommend that the variable be assigned in the setup script, or before any functions are run in a non-setup script. They also need to be outside of any try/except blocks.
  Example:
  - The setup script is successful and creates a needed resource, intended to be deleted in the cleanup script.
  - The main Action script does not run successfully and errors out and continues to the cleanup script
  - If there is a variable inside the main Action script, it is skipped over
  - When the cleanup script is run, there is a high likelihood of a traceback error due to the variable not being assigned.
- Setup/Cleanup Scripts are not required to be used.
- You can collapse then drag the Script sections to order and reorder them, as needed.

Script Execution Environment

When you run a Cloud Action, scripts are executed in an Ubuntu 22.04-based container with the following Python environment:

```
$ python3 --version
Python 3.10.12

$  pip3 list
Package                                    Version
------------------------------------------------- -----------
adal                                     1.2.7
aiohappyeyeballs                           2.3.5
aiohttp                                  3.10.3
aiosignal                                 1.3.1
annotated-types                            0.7.0
antlr4-python3-runtime                       4.13.2
anyio                                    4.4.0
applicationinsights                        0.11.10
argcomplete                               3.3.0
async-timeout                              4.0.3
attrs                                   24.2.0
azure-ai-formrecognizer                      3.3.3
azure-ai-textanalytics                      5.3.0
azure-appconfiguration                       1.1.1
azure-batch                              14.2.0
azure-cli                                2.63.0
azure-cli-core                             2.63.0
azure-cli-telemetry                          1.1.0
azure-cognitiveservices-language-spellcheck     2.0.0
azure-cognitiveservices-vision-contentmoderator 1.0.0
azure-cognitiveservices-vision-customvision     3.1.0
azure-common                               1.1.28
azure-core                               1.30.2
azure-cosmos                               3.2.0
azure-data-tables                          12.4.0
azure-datalake-store                        0.0.53
azure-digitaltwins-core                      1.2.0
azure-eventgrid                           4.20.0
azure-eventhub                             5.12.1
azure-eventhub-checkpointstoreblob            1.1.4
azure-eventhub-checkpointstoreblob-aio         1.1.4
azure-graphrbac                            0.60.0
azure-identity                           1.17.1
azure-keyvault-administration                4.4.0b2
azure-keyvault-certificates                 4.7.0
azure-keyvault-keys                        4.9.0b3
azure-keyvault-secrets                      4.7.0
azure-mgmt-advisor                          9.0.0
azure-mgmt-apimanagement                       4.0.0
azure-mgmt-appconfiguration                   3.0.0
azure-mgmt-appcontainers                     2.0.0
azure-mgmt-applicationinsights                1.0.0
azure-mgmt-appplatform                      9.1.0.post2
azure-mgmt-authorization                    4.0.0
azure-mgmt-avs                             8.0.0
azure-mgmt-azurearcdata                       1.0.0
azure-mgmt-azurestackhci                      7.0.0
azure-mgmt-batch                           17.2.0
```

```
azure-mgmt-batch                 17.3.0
azure-mgmt-batchai               7.0.0b1
azure-mgmt-billing               6.0.0
azure-mgmt-botservice            2.0.0
azure-mgmt-cdn                   12.0.0
azure-mgmt-changeanalysis        1.0.0
azure-mgmt-cognitiveservices     13.5.0
azure-mgmt-compute               31.0.0
azure-mgmt-consumption           10.0.0
azure-mgmt-containerinstance     10.1.0
azure-mgmt-containerregistry     10.3.0
azure-mgmt-containerservice      31.0.0
azure-mgmt-core                  1.4.0
azure-mgmt-cosmosdb              9.5.1
azure-mgmt-databoxedge           1.0.0
azure-mgmt-datamigration         10.0.0
azure-mgmt-devtestlabs           4.0.0
azure-mgmt-digitaltwins          6.4.0
azure-mgmt-dns                   8.0.0
azure-mgmt-edgeorder             1.0.0
azure-mgmt-elastic               1.0.0
azure-mgmt-eventgrid             10.2.0b2
azure-mgmt-eventhub              10.1.0
azure-mgmt-extendedlocation      1.0.0b2
azure-mgmt-hdinsight             9.0.0
azure-mgmt-hybridcompute         8.0.0
azure-mgmt-hybridnetwork         2.0.0
azure-mgmt-imagebuilder          1.3.0
azure-mgmt-iotcentral            10.0.0b2
azure-mgmt-iothub                3.0.0
azure-mgmt-iothubprovisioningservices 1.1.0
azure-mgmt-keyvault              10.3.0
azure-mgmt-kusto                 0.3.0
azure-mgmt-loganalytics          13.0.0b4
azure-mgmt-logz                  1.0.0
azure-mgmt-managedservices       1.0.0
azure-mgmt-managementgroups      1.0.0
azure-mgmt-maps                  2.0.0
azure-mgmt-marketplaceordering   1.1.0
azure-mgmt-media                 9.0.0
azure-mgmt-monitor               5.0.1
azure-mgmt-msi                   7.0.0
azure-mgmt-netapp                10.1.0
azure-mgmt-network               26.0.0
azure-mgmt-nspkg                 3.0.2
azure-mgmt-policyinsights        1.1.0b4
azure-mgmt-portal                1.0.0
azure-mgmt-powerbiembedded       3.0.0
azure-mgmt-privatedns            1.0.0
azure-mgmt-rdbms                 10.2.0b17
azure-mgmt-recoveryservices      3.0.0
azure-mgmt-recoveryservicesbackup 9.1.0
azure-mgmt-redhatopenshift       1.4.0
azure-mgmt-redis                 14.3.0
azure-mgmt-redisenterprise       3.0.0
azure-mgmt-reservations          2.3.0
azure-mgmt-resource              23.1.1
azure-mgmt-resourcegraph         8.0.0
```

```
azure-mgmt-resourcegraph                    8.0.0
azure-mgmt-search                           9.1.0
azure-mgmt-security                         6.0.0
azure-mgmt-servermanager                    2.0.0
azure-mgmt-servicebus                       8.2.0
azure-mgmt-servicefabric                    2.1.0
azure-mgmt-servicefabricmanagedclusters     2.0.0b6
azure-mgmt-servicelinker                    1.2.0b2
azure-mgmt-signalr                          2.0.0b1
azure-mgmt-sql                              4.0.0b17
azure-mgmt-sqlvirtualmachine                1.0.0b5
azure-mgmt-storage                          21.2.0
azure-mgmt-storagepool                      1.0.0
azure-mgmt-streamanalytics                  1.0.0
azure-mgmt-support                          7.0.0
azure-mgmt-synapse                          2.1.0b5
azure-mgmt-trafficmanager                   1.0.0
azure-mgmt-web                              7.2.0
azure-mgmt-webpubsub                        1.1.0
azure-monitor-query                         1.2.0
azure-multiapi-storage                      1.2.0
azure-nspkg                                 3.0.2
azure-search-documents                      11.5.1
azure-servicebus                            7.12.2
azure-servicefabric                         8.2.0.0
azure-storage-blob                          12.22.0
azure-storage-common                        1.4.2
azure-storage-file-datalake                 12.16.0
azure-storage-file-share                    12.17.0
azure-storage-queue                         12.11.0
azure-synapse-accesscontrol                 0.5.0
azure-synapse-artifacts                     0.19.0
azure-synapse-managedprivateendpoints       0.4.0
azure-synapse-spark                         0.2.0
bcrypt                                      4.2.0
boto3                                       1.34.162
botocore                                    1.34.162
build                                       0.10.0
cachetools                                  5.4.0
certifi                                     2024.7.4
cffi                                        1.17.0
chardet                                     5.2.0
charset-normalizer                          3.3.2
click                                       8.1.3
colorama                                    0.4.6
cryptography                                43.0.0
decorator                                   5.1.1
Deprecated                                  1.2.14
distro                                      1.9.0
dnspython                                   2.6.1
docstring_parser                            0.16
exceptiongroup                              1.2.2
fabric                                      3.2.2
frozenlist                                  1.4.1
google-api-core                             2.19.1
google-api-python-client                    2.141.0
google-auth                                 2.33.0
google-auth-httplib2                        0.2.0
```

```
google-auth-httplib2                    0.2.0
google-cloud-access-approval            1.13.5
google-cloud-access-context-manager     0.2.0
google-cloud-aiplatform                 1.62.0
google-cloud-api-gateway                1.9.5
google-cloud-apigee-connect             1.9.5
google-cloud-appengine-admin            1.11.5
google-cloud-appengine-logging          1.4.5
google-cloud-artifact-registry          1.11.5
google-cloud-asset                      3.26.3
google-cloud-assured-workloads          1.12.5
google-cloud-audit-log                  0.2.5
google-cloud-automl                     2.13.5
google-cloud-bare-metal-solution        1.7.5
google-cloud-bigquery                   3.25.0
google-cloud-bigquery-connection        1.15.5
google-cloud-bigquery-datatransfer      3.15.5
google-cloud-bigquery-logging           1.4.5
google-cloud-bigquery-reservation       1.13.5
google-cloud-bigquery-storage           2.25.0
google-cloud-bigtable                   2.26.0
google-cloud-billing                    1.13.6
google-cloud-billing-budgets            1.14.5
google-cloud-binary-authorization       1.10.5
google-cloud-build                      3.24.2
google-cloud-channel                    1.18.5
google-cloud-common                     1.3.5
google-cloud-compute                    1.19.2
google-cloud-contact-center-insights    1.17.5
google-cloud-container                  2.50.0
google-cloud-containeranalysis          2.14.5
google-cloud-core                       2.4.1
google-cloud-data-fusion                1.10.5
google-cloud-datacatalog                3.20.1
google-cloud-dataplex                   2.2.2
google-cloud-dataproc                   5.10.2
google-cloud-dataproc-metastore         1.15.5
google-cloud-datastore                  2.20.1
google-cloud-datastream                 1.9.5
google-cloud-debugger-client            1.7.0
google-cloud-deploy                     2.0.1
google-cloud-dialogflow                 2.31.0
google-cloud-dialogflow-cx              1.35.0
google-cloud-dlp                        3.21.0
google-cloud-dms                        1.9.5
google-cloud-documentai                 2.31.0
google-cloud-domains                    1.7.5
google-cloud-essential-contacts         1.7.5
google-cloud-eventarc                   1.11.5
google-cloud-filestore                  1.9.5
google-cloud-firestore                  2.17.2
google-cloud-functions                  1.17.0
google-cloud-game-servers               1.8.3
google-cloud-gke-hub                    1.14.2
google-cloud-iam                        2.15.2
google-cloud-iam-logging                1.3.5
google-cloud-iap                        1.13.5
google-cloud-ids                        1.7.5
```

```
google-cloud-iot                                2.9.2
google-cloud-kms                                2.24.2
google-cloud-language                           2.14.0
google-cloud-logging                            3.11.1
google-cloud-managed-identities                 1.9.5
google-cloud-memcache                           1.9.5
google-cloud-monitoring                         2.22.2
google-cloud-monitoring-dashboards              2.15.3
google-cloud-monitoring-metrics-scopes          1.6.5
google-cloud-ndb                                2.3.1
google-cloud-network-connectivity               2.4.5
google-cloud-network-management                 1.18.0
google-cloud-notebooks                          1.10.5
google-cloud-optimization                       1.8.5
google-cloud-orchestration-airflow              1.13.1
google-cloud-org-policy                         1.11.0
google-cloud-os-config                          1.17.5
google-cloud-os-login                           2.14.6
google-cloud-policy-troubleshooter              1.11.5
google-cloud-private-ca                         1.12.2
google-cloud-pubsub                             2.23.0
google-cloud-pubsublite                         1.11.1
google-cloud-recaptcha-enterprise               1.21.2
google-cloud-recommender                        2.15.5
google-cloud-redis                              2.15.5
google-cloud-resource-manager                   1.12.5
google-cloud-resource-settings                  1.9.6
google-cloud-retail                             1.21.2
google-cloud-scheduler                          2.13.5
google-cloud-secret-manager                     2.20.2
google-cloud-securitycenter                     1.34.0
google-cloud-service-control                    1.12.3
google-cloud-service-directory                  1.11.6
google-cloud-service-management                 1.8.5
google-cloud-service-usage                      1.10.5
google-cloud-shell                              1.9.5
google-cloud-source-context                     1.5.5
google-cloud-spanner                            3.48.0
google-cloud-speech                             2.27.0
google-cloud-storage                            2.18.2
google-cloud-storage-transfer                   1.11.5
google-cloud-talent                             2.13.5
google-cloud-tasks                              2.16.5
google-cloud-texttospeech                       2.16.5
google-cloud-tpu                                1.18.5
google-cloud-trace                              1.13.5
google-cloud-translate                          3.16.0
google-cloud-video-live-stream                  1.8.1
google-cloud-video-transcoder                   1.12.5
google-cloud-videointelligence                  2.13.5
google-cloud-vision                             3.7.4
google-cloud-vm-migration                       1.8.5
google-cloud-vpc-access                         1.10.5
google-cloud-webrisk                            1.14.5
google-cloud-websecurityscanner                 1.14.5
google-cloud-workflows                          1.14.5
google-crc32c                                   1.5.0
```

```
google-resumable-media                  2.7.2
googleapis-common-protos                1.63.2
grafeas                                 1.11.0
grpc-google-iam-v1                      0.13.1
grpc-interceptor                        0.15.4
grpcio                                  1.65.4
grpcio-status                           1.62.3
h11                                     0.14.0
h2                                      4.1.0
hpack                                   4.0.0
httpcore                                1.0.5
httplib2                                0.22.0
httpx                                   0.27.0
humanfriendly                           10.0
hyperframe                              6.0.1
idna                                    3.7
importlib_metadata                      8.0.0
invoke                                  2.2.0
isodate                                 0.6.1
javaproperties                          0.5.2
jmespath                                1.0.1
jsondiff                                2.0.0
knack                                   0.11.0
microsoft-kiota-abstractions            1.3.3
microsoft-kiota-authentication-azure    1.0.0
microsoft-kiota-http                    1.3.3
microsoft-kiota-serialization-form      0.1.0
microsoft-kiota-serialization-json      1.3.0
microsoft-kiota-serialization-multipart 0.1.0
microsoft-kiota-serialization-text      1.0.0
msal                                    1.30.0
msal-extensions                         1.2.0
msgraph-core                            1.1.2
msgraph-sdk                             1.5.4
msrest                                  0.7.1
msrestazure                             0.6.4.post1
multidict                               6.0.5
numpy                                   2.0.1
oauthlib                                3.2.2
opentelemetry-api                       1.26.0
opentelemetry-sdk                       1.26.0
opentelemetry-semantic-conventions      0.47b0
overrides                               7.7.0
packaging                               24.1
paramiko                                3.4.1
pendulum                                3.0.0
pip                                     22.3.1
pip-tools                               6.12.1
pkginfo                                 1.11.1
portalocker                             2.10.1
proto-plus                              1.24.0
protobuf                                4.25.4
psutil                                  6.0.0
pyasn1                                  0.6.0
pyasn1_modules                          0.4.0
pycomposefile                           0.0.31
pycparser                               2.22
```

pydantic	2.8.2
pydantic_core	2.20.1
PyGithub	1.59.1
Pygments	2.18.0
PyJWT	2.9.0
pymemcache	4.0.0
PyNaCl	1.5.0
pyOpenSSL	24.2.1
pyparsing	3.1.2
pyproject_hooks	1.0.0
PySocks	1.7.1
python-dateutil	2.9.0.post0
pytz	2024.1
PyYAML	6.0.2
redis	5.0.8
requests	2.32.3
requests-oauthlib	2.0.0
requests-toolbelt	1.0.0
rsa	4.9
s3transfer	0.10.2
scp	0.13.6
semver	2.13.0
setuptools	65.7.0
shapely	2.0.5
six	1.16.0
sniffio	1.3.1
sqlparse	0.5.1
sshtunnel	0.1.5
std-uritemplate	1.0.5
tabulate	0.9.0
time-machine	2.15.0
tomli	2.0.1
typing_extensions	4.12.2
tzdata	2024.1
uritemplate	4.1.1
urllib3	2.2.2
websocket-client	1.3.3
wheel	0.38.4
wrapt	1.16.0
xmltodict	0.13.0
yarl	1.9.4
zipp	3.20.0

Script Variables

Cloud Actions support the use of Input and Output variables. The Action has specific areas for adding these, but output variables can also be defined directly in the script.

Inputs

All input variables that you want to use in your script must be defined in the Inputs section of the Action.

- Declaring the datatype is not supported; input values are provided to the Python script as strings.
- If input variables are defined, the variables appear as Runtime parameters when you run the Action. The default value is pre-populated but can be changed.

- Environment variables `RESOURCE_TAG_KEY` and `RESOURCE_TAG_VALUE` are automatically provided by the Director when running Cloud Actions. The input values for these variables should use the following formats:
  - `RESOURCE_TAG_KEY: msv:job_id`
  - `RESOURCE_TAG_VALUE: org_uuid:job_id`

Input Name	Sample Value or Placeholder	Notes
RESOURCE_TAG_KEY	msv:job_id	This value never changes
RESOURCE_TAG_VALUE	c1587255-67a2-42cf-8dbe-5c8850592b6f:13110	The company's org id followed by the job id
CREATED_BUCKET_NAME	msv-A110-039-qt-script	Bucket name to be created. Should NOT be existing in AWS.
INSTANCE_NAME	MSV_Test_Instance_QT_SCRIPT	Name of the EC2 Instance to create.
TARGET_INSTANCE_ID	i-08b42ba5b21c3c5fc	EXISTING EC2 instance to perform action on.
TARGET_STACK_NAME	StackSet-AWSControlTowerBP-BASELINE-CONFIG-27c86162-55f5-407f-b6c3-13a9bd3f1855	EXISTING CloudFormation stack to perform action on.
TARGET_SNAPSHOT_ID	i08b42ba5b21c3c5fc	EXISTING EC2 Snapshot ID to perform action on.
TARGET_VOLUME_ID	vol-0ed25176d8f4235ff	EXISTING EC2 Volume ID to perform action on.
TARGET_IAM_INSTANCE_PROFILE	some_role	EXISTING instance profile to perform action on.
TARGET_IAM_POLICY_ARN	arn:aws:iam::371003120642:policy/EC2_Policy	EXISTING ARN Policy to perform action on.
TARGET_CLOUDFORMATION_STACK_NAME	StackSet-AWSControlTowerBP-BASELINE-CONFIG-27c86162-55f5-407f-b6c3-13a9bd3f1855	EXISTING Cloudformation Stack to perform action on.
TARGET_IAM_OLD_ROLE	Editor	EXISTING IAM role
TARGET_IAM_USER	Example-user	EXISTING IAM user
TARGET_VPC_TARGET_VPC_SUBNET_ID	Random_Subnet	EXISTING VPC ID
TARGET_S3_BUCKET	Random-bucket	EXISTING S3 bucket
LOG_GROUP_CREATED	MSV-LOG-GROUP	Name of log group to be created, must be unique.
SNAPSHOT_ID	i08b42ba5b21c3c5fc	EXISTING snapshot ID
DELETE_IAM_GROUP_NAME	MSV_Test_TARGET_IAM_GROUP_QT_SCRIPT	Unique IAM group name to be created and then deleted as an action test.

Input Name	Sample Value or Placeholder	Notes
ATTACKER_AWS_ACC_ID	5b17a4b74e11	Valid AWS account ID
MALICIOUS_STACK_NAME	MSV-VRT-TESTSTACK	cloudformation stack name to be created.
TARGET_REGION_NAME	Us-east-1	Valid AWS region name.
TARGET_EC2_SECURITY_GROUP_IDS	Existing-security-group	EXISTING security group ID to perform action on.
TARGET_SECURITY_GROUP	New-security-group	New security group name to be created.
TARGET_IAM_NEW_ROLE	New-iam-role	New IAM role to be created.
CREATE_IAM_GROUP_NAME	MSV_Test_TARGET_IAM_GROUP_QT_SCRIPT	IAM group name to be created and then deleted as an action test.
DRY_RUN	yes/no	Dry run flag: tests for destructive behavior.
TARGET_BUCKET_NAME	NEW_BUCKET_QT_SCRIPT	New S3 bucket to be created.
AMI_IMAGE_ID	ami-0c2b8ca1dad447f8a	AMI template to create EC2 instances.
SNAPSHOT_NAME	MSV_TEST_SNAPSHOT_QT_SCRIPT	Name of the new snapshot to be created.
EXFIL_DOMAIN	example.local	Sample domain exfiltration name.
TARGET_VPC_SUBNET_ID	Rando-subnet	EXISTING subnet ID
TARGET_IAM_ROLE	Editor	EXISTING IAM role
TARGET_EC2_ID	i-08b42ba5b21c3c5fc	EXISTING EC2 instance ID
EC2_KEYPAIR	Test-keypair	EC2 keypair name to create and then delete.
TARGET_GLACIER_VAULT_NAME	MSV-TESTVAULT	EXISTING Vault name
TARGET_IAM_GROUP	Example-group	Existing Glacier vault name to perform actions on.
LOG_STREAM_CREATED	MSV-LOG-STREAM	Name of the log stream to be created. Must be unique.
USER_FILTER	admin1,admin2	Comma-separated list of IAM accounts. Optionally used to filter out non-human accounts.

Outputs

- Output variables can be defined at the Action level or directly in the script.
- Output variables from one script are available in subsequent scripts.
- Output variables must start with the following prefix: **MSV_**.
    - Spaces are not allowed.
    - The variable is case sensitive, so the prefix must be all caps.

- Any variable defined by the script with the **MSV_** prefix are returned to the Director.
- There are three special Validation output variables:
  - ***MSV_BLOCKED***: This variable is required to define the Action's blocked / not blocked conditions. The default value is False and the supported values are True or False.
  - ***MSV_IDENTIFIERS***: This variable can be used for event matching if events can't be auto-discovered. Matching for AWS Request IDs in CloudTrail and GuardDuty events is supported.
    When defining the *MSV_IDENTIFIERS*, enter them as an array of strings. Example 1 defines specific values. Example 2 takes the RequestID from the AWS API response (which, when processed, results in the format used in Example 1).
    - Example 1

      ```
      MSV_IDENTIFIERS = { "cloud_request_ids" => ["ABC123", "DEF456"] }
      ```

    - Example 2

      ```
      MSV_IDENTIFIERS = { "cloud_request_ids": [response['ResponseMetadata'] ['RequestId']] }
      ```

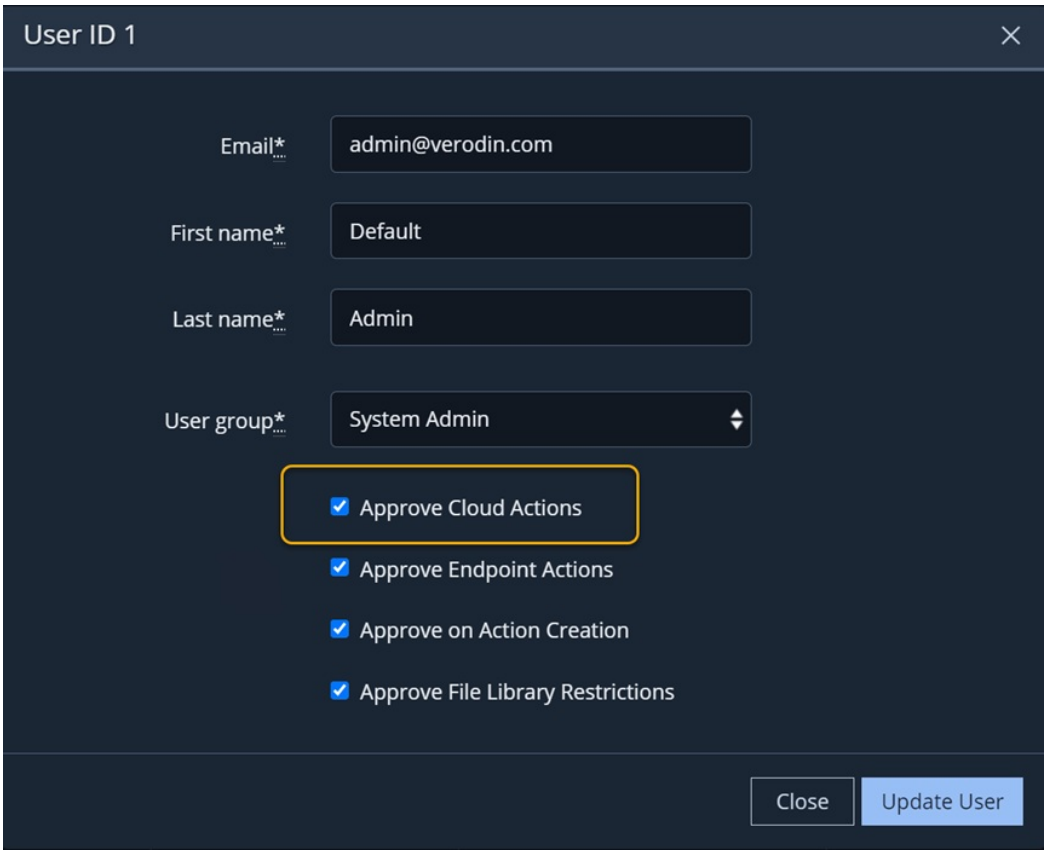
  - ***MSV_ACTION_ERRORS***: This variable is used to specify an error.

    ```
    MSV_ACTION_ERRORS = ["An example of using MSV_ACTION_ERRORS to specify an error"]
    ```

- The data returned in an output variable must be of a JSON-serializable type.
- Output variables other than the preceding three are not currently used.

Verify User Permissions

Before users can add Cloud Actions, they must have the **Approve Cloud Actions** setting enabled in the User dialog.



Approve Cloud Actions selected

- If the Approve Cloud Actions setting is enabled, the user can approve on save.
- If the Approve Cloud Actions setting is not enabled, the user can go through the creation process but cannot approve the Cloud Action on save. Rather, their request goes into an Approval Queue for approval.

Create a Cloud Action

1. Select **Library** > **Actions**.
2. Click **Add Action** and select **Cloud**.

   The create Action form has three sections. The default view is the wizard, with each section on its own page. You can also have the three sections listed vertically. You can switch your view by toggling **Show as Wizard** \ **Show All Steps**.

3. Enter the **Name** and **Description** and then click **Continue**. The **Action Type** defaults to Cloud and can remain unchanged.
4. (Optional) Click **Choose File** to add one or more non-malicious files.
   - Select the **File Upload** tab to upload a new file.
   - Select the **File Library** tab to select a File from the Library. The file library data takes a few moments to load.
5. (Optional) If you added a file, assign a **File Name** to each file that you add to the Cloud Action.

- File dependencies are found in the current working directory of the script, so script authors can reference them without specifying a path - for example, `open('xyz')`, where `xyz` is the name you picked for the file.
- Filenames must be unique per file. For example, if you attach two files with the name `my_file` to the same Action, the Action fails to create and you receive an error saying that the names must be unique. However, you can use the same filename on two separate Actions.

6. (Optional) Enter **Inputs**.

   On Job runs, inputs and their values are sometimes necessary for identifying the correct resources, such as region, resource name, resource ID, and so on. See **Examples of Inputs and Sample Values** for examples.

7. (Optional) Enter **Outputs**.
8. For Scripts, do the following (you can have multiple Scripts in a single Action):
   a. If not already expanded, click **Expand Step** to define Step 1 and complete the following required fields:
      - **Name**: Name for the script
      - **Timeout**: Screen timeout value - enter a minimum value of 60 seconds.
      - **Setup Script**: Check this checkbox if you want this script to be the Setup script.
      - **Script**: Define the script

8. Click **Add Step** to add another script.
   a. If not already expanded, click **Expand Step** to define Step 2 and complete the following fields:
      - **Name**: Name for the script
      - **Timeout**: Screen timeout value - enter a minimum value of 60 seconds.
      - **Cleanup Script**: Check this checkbox if you want this script to be the Cleanup script.
      - **Script**: Define the script

9. Click **Add Step** to add each script.

   You can reorder the scripts by clicking **Collapse** to collapse the scripts. Then, select and hold **Drag to Move Step** and drag the script to move it up or down. As you reorder the scripts, the first script in the list has the Setup script checkbox. The last script has the Cleanup script checkbox.

10. You can delete a script by clicking **Remove Step** by the script you want to remove. You can also Expand or Collapse the entire list of scripts by clicking **Expand All** or **Collapse All** at the top of the Scripts list.
    - Use the root logger (`logging.getLogger()` or just `logging.info()`, `logging.debug()`, etc) to emit messages into the log stream that is returned to the Director. Logs from third party libraries (for example, `boto3`, `requests`) are also collected in the Cloud Action log stream.

      Standard error and standard output (for example, `print()`) is not displayed by the Director.

    - The script should only handle exceptions when it's relevant to determining the blocked status of the Action (or other Action output). An unhandled exception is returned to the Director to provide feedback about the reason the Action failed.

   Click **Need Help ?** to view help around creating Cloud Actions, including sample scripts.

11. When you have defined the scripts, click **Continue**.

> You can log data back to the Director, which can help in both debugging your Python script and provide additional information.

12. (Optional) Enter or Select Tags you want to define, as needed.
13. Define the Dimensions (If there is a common value for a Dimension, it's listed as follows).
    - **Attack Vector**: Cloud API, Cloud Storage, Cloud Workload, or IAM (options found from IaaS)
    - **Attacker Location**:
    - **Behavior Type**:
    - **Covert**: No
    - **OS/Platform**: AWS
    - **Stage of Attack**:
14. Click **Save Action**. A confirmation window displays.
15. Select the Checkbox to acknowledge the message and then choose an option:
    - click **Add to Approval Queue** to require someone else to review before the Action goes live.
    - Click **Approve Now** if you're ready for the Action to go live.

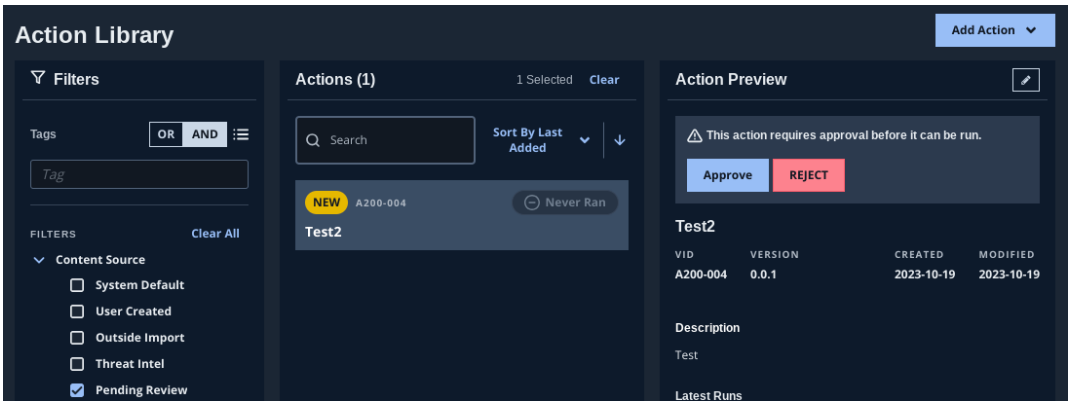
Approve a Cloud Action

The ability to approve Actions is controlled by user permissions. If you cannot Approve Actions and you should, contact your platform administrator.

1. Select **Library > Actions**.
2. Expand the **Content Source** Filter and select **Pending Review**.
3. Select the Action.
4. Approve or Reject the Action.
    a. Click **Approve** if it's a non-destructive behavior.

    > This button does not appear if there is a malicious file attached.

    b. Click **Reject** if there is an issue with the Action.



Approve or Reject Action

Sample Cloud Actions

AWS Action

This Action checks to see if it can confirm a specific bucket name exists and provides info to the Director on how to identify events.

- Inputs required: search_bucket
- Cloud Profile required? Yes

```
import logging
import os
import boto3

client = boto3.client('s3')
response = client.list_buckets()
search_bucket = ""
MSV_buckets = []

for b in response['Buckets']:
  MSV_buckets.append(b['Name'])

# define block condition (optional)
# if the bucket is not found, mark it as blocked
if search_bucket not in MSV_buckets:
  logging.info("{} not found, marking action as blocked".format(search_bucket))
  MSV_BLOCKED = True

# define how to identify events
MSV_IDENTIFIERS = [response['ResponseMetadata']['RequestId']]
```

Azure Action

This Action demonstrates an attacker listing all Azure Blob containers inside the environment. Blob containers usually contain sensitive information such as logs containing operational status or personally identifiable information.

Successful execution of this Action by a malicious actor would result in a list of all Blob containers in the environment.

- Inputs required: yes; this action requires input for `TARGET_STORAGE_ACCOUNT` (the name of an existing storage account).
- Cloud Profile required? Yes

```
import logging
import re

import azure.core.exceptions
from azure.identity import DefaultAzureCredential
from azure.storage.blob import BlobServiceClient

Blocked_Errors = [r'Code:\s*AuthorizationPermissionMismatch', r'Code:\s*AuthorizationFailure']


def list_azure_blobs(storage_account):
    blob_client = BlobServiceClient(
                account_url=f'https://{storage_account}.blob.core.windows.net',
                credential=DefaultAzureCredential())
    container_poller = blob_client.list_containers()
    container_list = [c.name for c in container_poller]

    for container in container_list:
        logging.info("Found container %s", container)

    return container_list

try:
    MSV_OUTPUT = list_azure_blobs(TARGET_STORAGE_ACCOUNT)
    MSV_BLOCKED = False
    MSV_REASON = "Blobs were listed successfully. This Action was not blocked."
except azure.core.exceptions.HttpResponseError as e:
    logging.error(str(e))
    if any(re.search(rex, str(e)) for rex in Blocked_Errors):
        MSV_BLOCKED = True
        MSV_REASON = "Blobs were not able to be listed. This Action was blocked."
    else:
        raise
```

Google Cloud Action

This Action demonstrates deleting a storage bucket within a given Google Cloud project ID. Buckets are the basic containers that hold data in Google Cloud. Buckets can be used to organize and control access to data. Execution of this behavior by a malicious actor could result in loss of data, or defense evasion by deleting malicious code.

- Inputs required: no, but optionally, `BUCKET_NAME` can be populated to suit the environment.
- Cloud Profile required? Optionally for the Setup Script, Delete Bucket, and Cleanup Script

```
import os
import logging
import random
from google.cloud import storage

MSV_CLEANUP = False

def create_bucket(msv_new_bucket_name):
    storage_client = storage.Client()
    bucket = storage_client.create_bucket(msv_new_bucket_name)
    labels = bucket.labels
    labels[os.getenv('RESOURCE_TAG_KEY').replace(':', '_')] = os.getenv('RESOURCE_TAG_VALUE').replace(':', '_')
    bucket.labels = labels
    bucket.patch()

    output = (f'Bucket {msv_new_bucket_name} created')
    storage_client.get_bucket(msv_new_bucket_name)
    return output


try:
    MSV_NEW_BUCKET_NAME = f'{BUCKET_NAME}' + str(random.randint(0, 99))
    create_bucket(MSV_NEW_BUCKET_NAME)
    MSV_CLEANUP = True
    logging.debug('Setup was successful')
except Exception as e:
    MSV_CLEANUP = False
    logging.error(str(e))
    logging.error('Setup failed.')
    raise
```