Expanse InsightVM Integration v1.2.0

Documentation and User Guide

Expanse's InsightVM integration allows you to automatically import assets detected by Expanse into Rapid7's InsightVM console. This documentation details the requirements for the integration, how to set up and run the integration, how to configure the integration, and how to debug common errors.

Goals and Outcomes

Examples of goals and outcomes for customers using the Expanse InsightVM integration include:

- Keeping an up-to-date inventory in InsightVM of an organization's external facing assets by using data from Expander.
- Importing tags from Expander into InsightVM to allow utilization of Expanse business context within InsightVM.

Integration Requirements

Recommended System Requirements

To run the integration, you will need a system on which you can leave a continuously running Docker container. Check the Docker installation guide for the system requirements to install Docker. In addition, we recommend you have at least 1 GB of RAM on top of what is required for Docker to run the application.

Docker

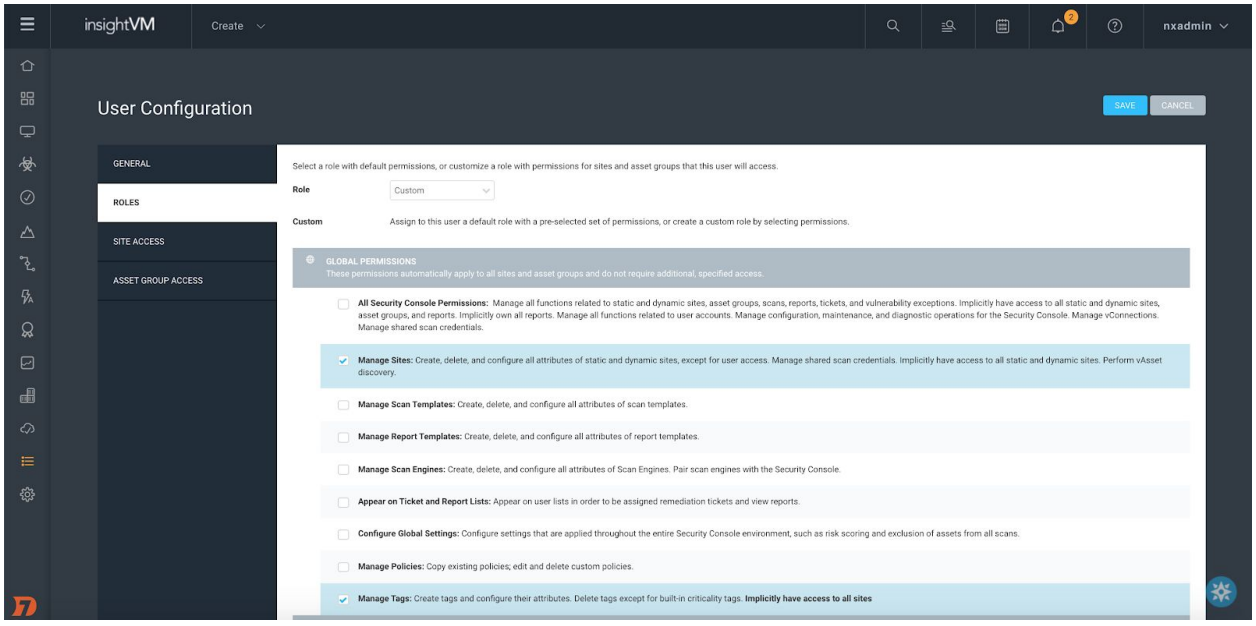
You will need to install Docker to run the integration. Check the Docker installation guide for guidance on the installation process.

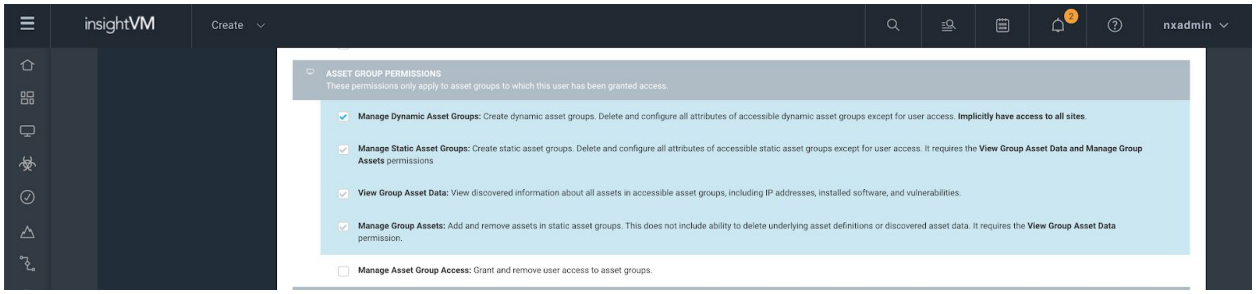
Network Requirements

The system on which the integration is running will need to have a network connection through which it can access both https://expander.expanse.co/ and your InsightVM instance. The integration communicates to both Expander and InsightVM over HTTPS, connecting to port 443 for Expanse and port 3780 for InsightVM.

InsightVM Permissions

In order for the integration to upload asset data to InsightVM, you will need to provide it with InsightVM login credentials that have permission to **manges sites, manage tags,** and **manage dynamic asset groups**. These permissions are shown below in the the User Configuration page of InsightVM.




insightVM
Create
nxadmin
ASSET GROUP PERMISSIONS
These permissions only apply to asset groups to which this user has been granted access.
Manage Dynamic Asset Groups: Create dynamic asset groups. Delete and configure all attributes of accessible dynamic asset groups except for user access. Implicitly have access to all sites.
Manage Static Asset Groups: Create static asset groups. Delete and configure all attributes of accessible static asset groups except for user access. It requires the View Group Asset Data and Manage Group Assets permissions
View Group Asset Data: View discovered information about all assets in accessible asset groups, including IP addresses, installed software, and vulnerabilities.
Manage Group Assets: Add and remove assets in static asset groups. This does not include ability to delete underlying asset definitions or discovered asset data. It requires the View Group Asset Data permission.
Manage Asset Group Access: Grant and remove user access to asset groups.

Setting up and Running the Integration

Setup

1. Ensure that Docker is installed by running the command "`docker version`". If Docker is not installed, you can follow the directions here to install it: https://docs.docker.com/engine/install/

2. Unpack the tarball.

3. Edit the "config.yml" file in the root directory of the application to set your desired configuration settings. Reference the "Configuring Your Expanse InsightVM Integration" section below for more details.

4. Open the "crontab" file in a text editor. On the second to last line, you will see "* * * * *" followed by a series of commands. Edit the "* * * * *" to adjust the run schedule to your desired frequency
   NOTE: Docker uses Greenwich Mean Time by default, so you will need to set your run schedule in GMT. For reference, GMT is 4 hours ahead of EST, and 7 hours ahead of PST.
   Some common examples:

Schedule	Cron Schedule Expression
Every day at midnight GMT (8pm EST / 5pm PST)	0 0 * * *
Every day at midnight EST (4am GMT / 9pm PST)	0 4 * * *
Every day at midnight PST (3am EST / 7am GMT)	0 7 * * *
Every Sunday at midnight PST	0 7 * * 0
Every Friday at midnight PST	0 7 * * 5

   See https://crontab.guru/examples.html for more examples on how to do this. Keep in mind that Docker uses Greenwich Mean Time by default, so you will need to set your run schedule in GMT. For reference, GMT is 4 hours ahead of EST, and 7 hours ahead of PST.

Running

5. Run the command "`docker build -t expanse-insightvm .`" from the root project directory to create the docker image.

6. Run the command "`docker run --name expanse-insightvm expanse-insightvm`" to run the image in a container (Note: you will also need to run the image with environmental variables for any fields you left blank in config.yml). This will begin the cron job, and the Python script will run according to the frequency in crontab.

To force execution of a single run of the Python script, run "`python bin/main.py`" from within the docker container. This can be achieved from outside the container, for instance, with the command "`docker exec expanse-insightvm "python bin/main.py"`"

To stop the application, simply stop the Docker container. This can be accomplished by running the command "`docker stop expanse-insightvm`".

To view logs, run the command "`docker logs expanse-insightvm`". It may be useful to save the logs to a file, which can be done, for instance, with the command "`docker logs expanse-insightvm >> path/to/log_file`".

For more information about any of these Docker commands, reference the following Docker documentation: https://docs.docker.com/engine/reference/commandline/

Configuring Your Expanse InsightVM Integration

Configuration is handled through the config.yml file, and optionally, through environmental variables. Below is an example of what the config.yml file should look like after being filled out.

```
rapid7:
  # Username used to authenticate requests to InsightVM.
  # Can also be passed in as the environment variable RAPID7_USERNAME
  username: InsightVMUser

  # Password used to authenticate requests to InsightVM.
  # Can also be passed in as the environment variable RAPID7_PASSWORD
  password: InsightVMPassword123!

  # The URL of the InsightVM instance.
  # Can be an IP address and port number (e.g. https://192.168.0.1:3780).
  # Can also be passed in as the environment variable RAPID7_URL
  url: https://192.168.0.1:3780

  # Whether or not to verify the certificate of the InsightVM instance when
  # making HTTPS requests. If your InsightVM instance is using a self-signed
  # certificate, set this value to False.
  # Default is True.
  verify_certificate: True

  # The request timeout in seconds for API requests to the InsightVM console.
  # Default is 10.
  request_timeout: 10


expanse:
  # The refresh token used to get data from Expander.
  # Can also be passed in as the environment variable EXPANDER_BEARER_TOKEN
  api_token: asdfgh3425dfg352tsdfjdfi766efh-CjOrZ0vJWiDF

  # Optional Tag filter for Expanse assets.
  # All assets will be populated if left empty.
  tags:
    - dmz

 # A flag that will cause only responsive assets to be imported into InsightVM.
  responsive_only: False


general:
  # The name of the site to upload assets to.
  # NOTE: If a site by this name does not exist in InsightVM, a new site will
  # be created with this name. If you wish to change the name of the site in
  # InsightVM, be sure to also change this parameter, or else you may get
  # multiple sites in InsightVM.
  # Default is "Expanse Assets".
  site_name: Expanse Assets
```

```
# Maximum duration the integration should be allowed to run in minutes.
# Default is 480 minutes (8 hours)
max_run_duration: 480

# Maximum rate of requests sent to InsightVM in requests per minute.
# Set to 0 for no limit (i.e. make requests as fast as possible).
# It is likely that no rate limiting is required, but if your InsightVM
# instance is having trouble with the request load, setting a value between
# 500 and 1000 may help. A lower value decreases the load, but it will also
# make asset uploads take longer.
# Default is 0 (no rate limiting)
max_requests_per_minute: 0

# Whether to sync IP range assets from Expander with InsightVM
# Default is True
sync_ip_ranges: True

# Whether to sync domain assets from Expander with InsightVM
# Default is True
sync_domains: True

# Logging level (values can be DEBUG, INFO, WARNING, or ERROR)
# All logs of equal or greater severity to the selected level will be shown.
# Below are descriptions of each level in increasing order of severity:
#
# DEBUG:   Detailed info about all HTTP requests and actions taken.
# INFO:    General info giving an overview of the program's progress.
# WARNING:  Info about abnormal events that do not immediately present a
#         problem, but may cause issues later.
# ERROR:   Info about problems that prevent the program from performing a
#         required action.
#
# Default is INFO
logging_level: INFO
```

NOTE: Any time you wish to change the config.yml file, you will need to rebuild and redeploy the Docker container for the changes to take effect. To do this, first stop the container with "`docker stop expanse-insightvm`". Then execute both the "`docker build`" and "`docker run`" commands specified in the "Running" section above.

The config file contains the following fields to be completed.

Rapid 7 InsightVM Config

username

The username to use when making API calls to InsightVM. Can also be passed in as the environment variable RAPID7_USERNAME.
Remember to make sure that the user provided has the required permissions. See "InsightVM Permissions" under the section "Integration Requirements" above.

password

The password to use when making API calls to InsightVM. Can also be passed in as the environment variable RAPID7_PASSWORD.

url

The URL of the InsightVM instance. Can be an IP address and port number (e.g. https://192.168.0.1:3780). Can also be passed in as the environment variable RAPID7_URL.

verify_certificate

Whether or not the script should perform certificate verification when communicating with the InsightVM console. In some cases where the certificate is self-signed or has a long expiration date you may see communication issues because validation fails. This field is set to True by default.

request_timeout

This is the request timeout for communicating with the InsightVM console. By default this timeout is set to 10 seconds, but if the console is expected to take more than 10 seconds to respond to requests you can increase this value.

Expanse Config

api_token

The refresh token used to get data from Expander. Can also be passed in as the environment variable EXPANDER_BEARER_TOKEN.

tags

You may supply a list of tag names to indicate that IP ranges and domains you wish to import into the InsightVM console. If no tags are supplied, all configured assets will be imported.

responsive_only

This value will dictate whether all assets matching in tag filters are imported, or only the assets which have been found to be responsive. This value is False by default.

General Config

max_run_duration

Maximum duration in minutes that the integration should be allowed to run. Default is 480 minutes (8 hours). Use this setting to ensure that the integration does not run.

max_requests_per_minute

Maximum rate (in requests per minute) at which requests should be made to InsightVM. Set this parameter to 0 for no rate limiting (i.e. make requests as fast as possible). Default is 0.

Use this parameter if you wish to cap the load that this app can place on the InsightVM instance. Although it is likely that no rate limiting is required, if your InsightVM instance is having trouble with the request load, setting a value between 500 and 1000 might be helpful. A lower value decreases the load, but it will also make asset uploads take longer.

sync_ip_ranges

Whether to sync IP range assets from Expander with InsightVM. Should be set to either True or False. Default is True.

sync_domains

Whether to sync domain assets from Expander with InsightVM. Should be set to either True or False. Default is True.

logging_level

The level of detail of logs. Should be either DEBUG, INFO, WARNING, or ERROR. Default is INFO.
All logs of equal or greater severity to the selected level will be shown. Below are descriptions of each level in increasing order of severity.

DEBUG	Detailed info about all HTTP requests and actions taken.
INFO	General info giving an overview of the program's progress.
WARNING	Info about abnormal events that do not immediately present a problem, but may cause issues later.
ERROR	Info about problems that prevent the program from performing a required action.

Environment Variables

To use environment variables for certain config settings, simply make sure that when you run the Docker container, it is running with the correct environment variables. This can be accomplished, for instance, by using the "-e" tag in the "docker run" command.

For example, to set the RAPID7_USERNAME and RAPID7_PASSWORD environment variables to "myUser" and "myPassword" respectively, one could use the following command:

```
docker run -e RAPID7_USERNAME=myUser -e RAPID7_PASSWORD=myPassword --name
expanse-insightvm expanse-insightvm
```

Running the Integration

Once the configuration has been completed and the crontab file has been updated with your desired run interval, you can run the integration by doing the following:

1. Run the command "`docker build -t expanse-insightvm .`" from the root project directory to create the docker image.

2. Run the command "`docker run --name expanse-insightvm expanse-insightvm`" to run the image in a container (Note: you will also need to run the image with environmental variables for any fields you left blank in config.yml). This will begin the cron job, and the Python script will run according to the frequency in crontab.

Once the integration is running, you will not see any activity until the cron job is triggered. Once the job is triggered you should start seeing logs immediately. If this is the first time the integration has run you will see logging about creating a new site in InsightVM. You will then see logs about importing assets into the InsightVM console.

Once the integration has completed a run it will remain active until the cron job is triggered again.

Tagging

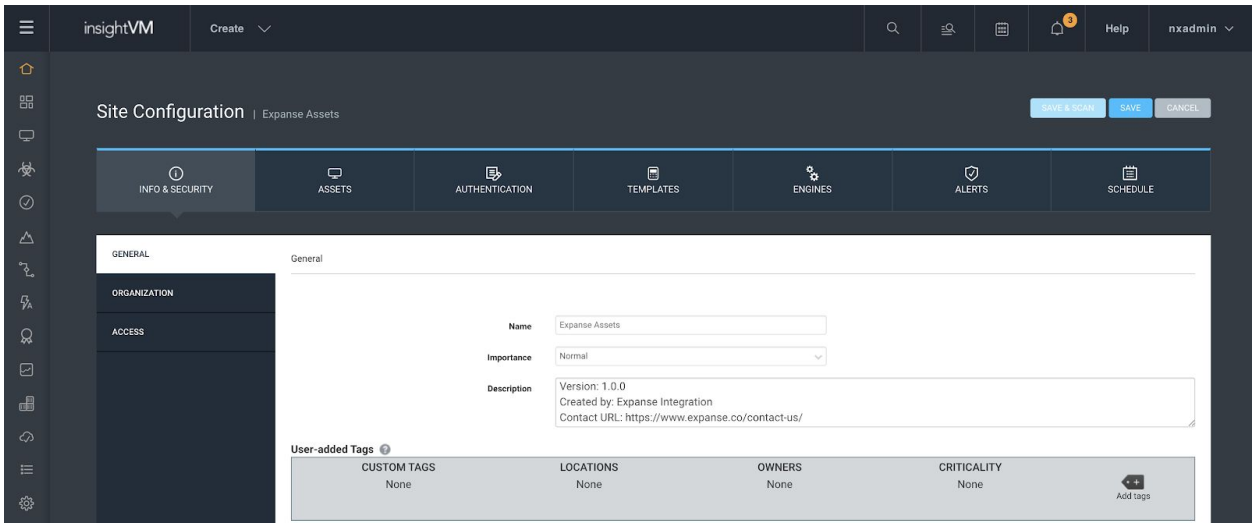
This integration will cause two groups of tags to be created in InsightVM. The first group is known as **Expanse Customer Tags**. These tags are a mirror of the tags that exist in your Expanse instance. All of the Expanse Customer tags begin with "Expanse Custom:" followed by the tag name.

The second group of tags are known as **Expanse Reserved Tags**. These tags do not necessarily correlate to tag assignments within Expanse. See below of an explanation of each reserved tag. Reserved tags begin with "Expanse Reserved:".

Tag Name	**Assignment Criteria**
Expanse Reserved: Current	This tag indicates that the asset was seen during the most recent run of the integration. It is current.
Expanse Reserved: Validated	This tag is assigned by doing a keyword match for "validated". Validated is assigned in Expanse internally by our analysts or Expanse users to imply high confidence in the finding.
Expanse Reserved: Confirmed	This tag is assigned by doing a keyword match for "confirmed". Confirmed is assigned in Expanse internally by our analysts or Expanse users to imply high confidence in the finding.
Expanse Reserved: Removed	This asset previously existed in Expanse but it is no longer appearing in the import results. This could be because the asset was removed from expanse, or because you tag filtering conditions have changed.

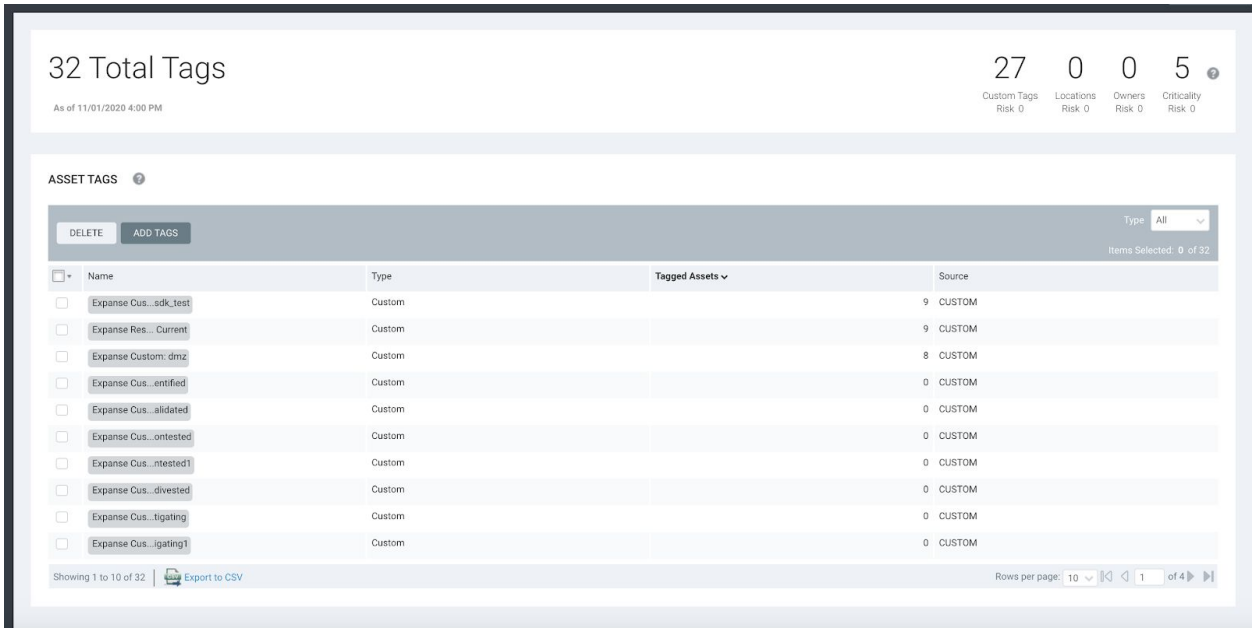
Expected Results

A new site will be added to the console if an existing site is not configured.

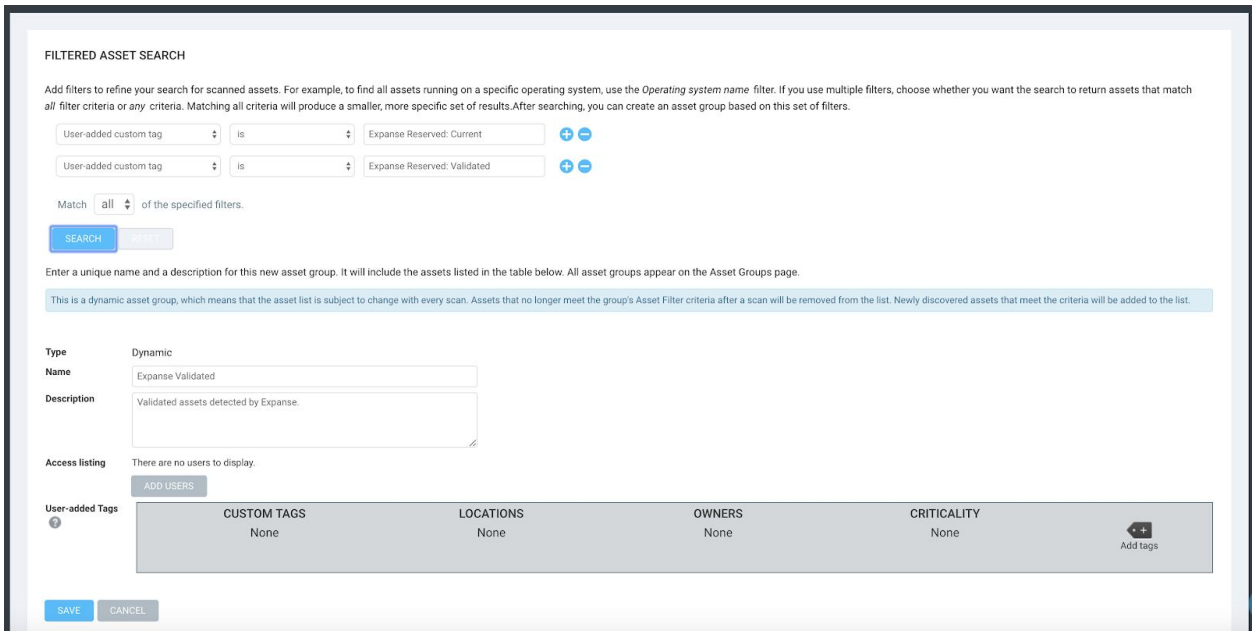


Assets will be imported into the site.

Tags will be created and applied to Expanse assets.



A Dynamic Asset Group will be created and resolve to all Expanse assets that are both “Current” and “Validated”.



Troubleshooting

In this section, we will cover what some of the most common log messages mean, as well as what the most common error messages are and how to fix them.
As mentioned in the “Setting up and Running the Integration” section, logs can be viewed with the command “`docker logs expanse-insightvm`”.

“Script is already running. Exiting...”

This message is printed when either the user or the cron scheduler attempts to run the script while another instance of the script is already running. When this happens, the new instance of the script is stopped, while the old instance is allowed to continue.

“Maximum runtime of ### minutes exceeded. Ending run.”

This message indicates that the script has exceeded the allotted time specified by the “max_run_duration” parameter in the config file, and so the script has ended. This message is **to be expected** in the first few runs of the integration after installation, since it is most likely that the script will run out of time before uploading all assets from Expander. However, after the first few runs, the script should be finished with the initial asset ingest, and should no longer display this message on a regular basis.
If your runs continue to exit with this message beyond the first 3-4 runs, you may want to either increase the “max_run_duration” parameter in the config file, or if you have set a non-zero value for “max_requests_per_minute”, you may try either increasing the value or setting it to zero.

“An unrecoverable error occurred. Aborting run...”

This message indicates that something failed which required the run to end prematurely. This message should be accompanied by a more detailed message and stack trace explaining the circumstances of the error.

“Missing config value `xxxxx.xxxxx`”

This message indicates that the specified config value was not found. If this happens, check the “config.yml” file to make sure that the required config value is there. If you had intended to pass in the config value in as an environment variable, check to make sure that name of variable you passed in is correct.

“Could not initialize Expanse API connection.”

This indicates that the application encountered an error trying to connect the Expanse API. There should be another message accompanying this one which gives more details about the error. Most likely, you will see one of the following.

“API returned an error while refreshing JWT: Could not get an id token based on the given refresh token”
This most likely means that the Expanse refresh token you provided was incorrect or mistyped. Check that you have provided the correct token.
“Request returned an exception: HTTPSConnectionPool(host='expander.expanse.co', port=443): ...”
This means that a connection could not be established to the Expanse API. Make sure that your system is connected to the internet, and that it can reach “https://expander.expanse.co.”

“Rapid7 InsightVM responded with an error (code ###):”

This message indicates that an API call to InsightVM returned an error. The error code and error message are given along with this notification. Some common error codes and messages are given below.

“(code 401): The supplied credentials are invalid”
This means the username and password you supplied were not valid. Check that the correct credentials were provided.
“(code 404): The resource does not exist or access is prohibited”

This means that either the resource being accessed does not exist, or that the user does not have permissions to access it. Most likely, it is the latter. Check that the username you provided to the integration has all the required permissions.

"(code 400): Cannot import asset data from 'mm/dd/yy 12:00 AM', the existing asset data from 'mm/dd/yy 12:00 AM' is newer."

This message means that the integration tried to upload an asset that already exists in InsightVM with more recent data. The integration is designed to avoid this issue by only uploading assets which are not already in InsightVM. If this error appears, it is possibly due to new asset data being uploaded while the integration was running. Ensure that you did not accidentally start two Docker containers running the integration at the same time.

"Could not find the validated tag named "validated" in Expander."

This is a warning message that indicates that the script could not find a tag named "validated" in Expander. This is important because the integration uses the "validated" tag in Expander to assign the "Expanse Reserved: Validated" tag in InsightVM. The script will still run in this case, but it will not assign the "Expanse Reserved: Validated" tag to any assets.