Collecting and reporting AWS billing data

OnCommand Insight

NetApp
October 24, 2024

Table of Contents

Collecting and reporting AWS billing data

The Amazon AWS Cloud Cost data source imports billing data generated by Amazon into Insight as integration data, making it available to the data warehouse for reporting.

There are three parts to making cloud billing data available to Insight:

Verifying your AWS account information.

Configuring the AWS Cloud Cost data source in Insight to collect the data.

Sending the data to Data Warehouse via ETL for use in reports.

Preparing AWS for Insight data collection

Your AWS account must be properly configured to allow Insight to collect cloud cost data.

About this task

The following steps are done through your AWS account. See the Amazon documentation for more information: http://docs.aws.amazon.com. If you are unfamiliar with setting up an AWS cloud account, contact your cloud provider for assistance.

These steps are provided here as a courtesy and are believed correct as of the time of publication. NetApp makes no guarantee of the correctness of these steps. Contact your cloud provider or AWS account holder for information or assistance on configuring your AWS account.

Best practice: Insight recommends that you create a primary IAM user on the same account that owns the S3 bucket where the billing reports are uploaded, and use this user to configure and collect AWS billing data.

To configure your AWS account to allow Insight to collect data, perform the following steps:

Steps

1. Log in to your AWS account as an Identity Access Management (IAM) user. For proper collection, log in to the primary IAM account, as opposed to a group IAM account.
2. Go to **Amazon S3** to create your bucket. Enter a unique bucket name and verify the correct Region.
3. Turn on your Amazon Cost and Usage Report. See https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/billing-reports-gettingstarted-turnonreports.html for information.
    a. Go to the AWS **Billing and Cost Management Dashboard**and choose **Reports**.
    b. Click on **Create report** and enter in the Report Name. For **Time unit**, choose Daily. Check the box to include **Resource IDs**, and click **Next**.
    c. Click on the **Sample Policy** link in the Select delivery options page. Copy the Sample Policy text in the box to the clipboard. Click **Close**.
    d. Go back to the S3 Bucket that was created, click on the **Permissions** tab and select the **Bucket Policy** button.
    e. Paste the text from the Sample Policy, and replace `<bucketname>` with your actual bucket name in each line that contains the following: `"Resource": "arn:aws:s3:: <bucketname>"`. **Save** the

policy.

f. Go back to your Create Report screen, enter in your S3 Bucket and click the **Verify** button. Click **Next**.

g. Verify your information and click **Review and Complete**.

4. You must grant permissions in order for Insight to collect data from AWS. The following link provides details on how to grant permissions to **List All Buckets** (Step 4.1) and set permissions on the objects in the folder (Step 5.2): https://docs.aws.amazon.com/AmazonS3/latest/dev/walkthrough1.html.
5. In the IAM console, go to **Policies** and click **Create policy**.
6. Enter a name in the **Policy Name** field, and click **Create policy** at the bottom.
7. In the IAM console, select your user, then select **Add Inline Policy** at the bottom of the screen.
8. Click on **Choose a service** and select S3.
9. Go to the **JSON** tab. Copy the JSON sample text from step 5.1.2.g of the AWS walkthrough into the JSON box.
10. Replace the *companybucket* and *Development* fields in the JSON with your S3 information.
11. Click **Review Policy** to review your policy settings.

Configuring the AWS Cloud Cost data source

You configure the AWS Cloud Cost data source as you would for any Insight data source.

Before you begin

You must have your Amazon AWS account already set up and prepared for Insight data collection, and have the following pieces of information at hand.

- Report Name
- S3 Bucket Name
- AWS Region where your S3 bucket resides.
- Report path prefix

About this task

Once your AWS account is ready and has the proper permissions set, you are ready to configure OnCommand Insight to collect billing report data.

You will need to add a separate AWS Cloud Cost data source for each billable user/account from which you wish to retrieve billing data.

Steps

1. Log in to OnCommand Insight as an administrator.
2. Click on **Admin** > **Data sources** to open the Insight Data Source page.
3. Click **+Add** to add a new data source. Choose **Amazon** and select **AWS Cloud Cost**.
4. In the **Configuration** section, fill in the *Report name*, *S3 Bucket* name, *S3 Region* (must be the region where your S3 bucket resides), *Report path prefix*, *AWS IAM Access Key ID*, and *AWS IAM Secret Access Key*. If you are unsure of any of these, check with your cloud provider or AWS account holder.

5. Click the checkbox to verify your understanding that AWS will bill you for API requests and data transfers made by the Insight data source.
6. In **Advanced Configuration**, enter the HTTP connection and socket timeout. The default is 300 seconds.
7. Click **Save**.

Processing AWS Cloud Cost data in Insight

Insight collects data from your AWS billing report once a month for the previous month, and reflects the finalized cloud cost for that month.

After you set up your AWS Cloud Cost data source(s), if you already had billing reports generated to S3, you will get up to three months of past data immediately after the first data source poll.

Insight collects AWS “final” data once a month. This collection occurs a few days after the close of the previous month, allowing AWS time to finalize the actual data.

AWS billing data is sent to Insight’s Data Warehouse for use in reporting.

Keep in mind that each data source must be configured for a single billable account/user.

Reporting on Cloud Cost data in Insight

Cloud cost monthly data collected in Insight is sent to the data warehouse and is available in the Cloud Cost datamart for use in reports.

Before you begin

You must have data sources configured to collect cloud cost data from AWS. Each billable user/account must have a separate data source.

Allow Insight at least 36 hours to begin collecting data.

Allow ETL to run at least once after that time, to send the data to the data warehouse.

About this task

After your data has been collected and sent to the data warehouse, you can view it in any of several pre-configured reports, or create custom reports. Insight stores the data in its own Cloud Cost datamart.

To view your cloud cost data in one of the pre-configured reports:

Steps

1. Open Insight Reporting by one of these methods:
   - Click on the Reporting Portal icon  in the Insight server web UI or in the Data Warehouse UI.
   - Launch Reporting directly by entering the following URL: `https://<dwh_server_name\>:9300/p2pd/servlet/dispatch` or `https://<dwh_server_name\>:9300/bi` `(7.3.3 and later)`
2. Once you are logged in to Reporting, click on **Public Folders** and select **Cloud Cost**.
3. You can view your AWS billing data in the available reports located in the **Cloud Cost** folder, or create your

own custom report using the **Cloud Cost datamart** available from the **Packages** folder.

Elevate role

You must elevate your ServiceNow role to security_admin before you can integrate with insight.

Steps

1. Log into your ServiceNow instance with administrator permissions.
2. Under the **System Administrator** drop-down, choose **Elevate Roles** and elevate your role to security_admin. Click OK.

Install update set

As part of the integration between ServiceNow and OnCommand Insight you must install an Update Set, which loads pre-configured data into ServiceNow in order to provide the connector with specific fields and tables for extracting and loading data.

Steps

1. Navigate to the remote update sets table in ServiceNow by searching for “Retrieved update sets”.
2. Click on **Import Update Set from XML**.
3. The update set is in the Python connector .zip file previously downloaded to your local drive (in our example, the `C:\OCI2SNOW` folder) in the `\update_sets` sub-folder. Click on **Choose File** and select the .xml file in this folder. Click **Upload**.
4. Once the Update Set is loaded, open it and click on **Preview Update Set**.

   If errors are detected, you must correct them before you can commit the Update Set.

5. If there are no errors, click **Commit Update Set**.

   Once the Update Set has been committed it will show on the **System Update Sets** > **Update Sources** page.

ServiceNow integration - Set up user

You must set up a ServiceNow user for Insight to connect with and synchronize data.

About this task

Steps

1. Create a services account in ServiceNow. Login to ServiceNow and navigate to **system security** > **users and groups** > **users**. Click on **New**.
2. Enter a user name. In this example, we will use “OCI2SNOW” as our integration user. Enter a password for this user.

In this How-to we use a services account user named “OCI2SNOW”`" across the documentation. You may use a different services account, but be sure it is consistent across your environment.

3. Right-click on the menu bar and click **Save**. This will allow you to stay on this user in order to add roles.
4. Click **Edit** and add the following roles to this user:
   - asset
   - import_transformer
   - rest_service
5. Click **Save**.
6. This same user must be added to OnCommand Insight. Log in to Insight as a user with Administrator permissions.
7. Navigate to **Admin** > **Setup** and click on the **Users** tab.
8. Click the **Actions** button and select **Add user**.
9. For name, enter “OCI2SNOW”. If you used a different user name above, enter that name here. Enter the same password you used for the ServiceNow user above. You may leave the email field blank.
10. Assign this user the **User** role. Click **Save**.

Install Python and libraries

Python can be installed on the Insight server or on a standalone host or VM.

Steps

1. On your VM or host, download Python 3.6 or later.
2. Choose custom installation and choose the following options. These are either necessary for proper connector script operation or are highly recommended.
   - Install launcher for all users
   - Add Python to the PATH
   - Install pip (which allows Python to install other packages)
   - Install tk/tcl and IDLE
   - Install the Python test suite
   - Install py launcher for all users
   - Associate files with Python
   - Create shortcuts for installed applications
   - Add python to environment variables
   - Precompile standard library
3. After Python is installed, install the “requests” and “psnow” Python libraries. Run the following command:
   `python -m pip install requests pysnow`

   NOTE: This command might fail when you are operating in a proxy environment. To work around this issue, you need to manually download each one of the Python Libraries and run the install requests one by one and in the correct order.

The command will install several files.

4. Verify the Python libraries are installed correctly. Start Python using one of the following methods:
   - Open a cmd prompt and type `python`
   - On Windows, open **Start** and choose **Python** > **python-<version>.exe**
5. At the Python prompt, type `modules`

   Python will ask you to wait a moment while it gathers a list of modules, which it will then display.

Setup Python middleware

Now that Python and the necessary libraries are installed, you can configure the middleware connector to communicate with OnCommand Insight and ServiceNow.

Steps

1. On the host or VM where you downloaded the connector software, open a cmd window as administrator and change to the `\OCI2SNOW\` folder.
2. You must initialize the script to generate an empty **config.ini** file. Run the following command: `oci_snow_sync.pyz init`
3. Open the **config.ini**file in a text editor and make the following changes in the [OCI] section:
   - Set url to <code><a href="https://<name.domain&gt" class="bare">https://<name.domain&gt;</code> or <code><a href="https://<ip" class="bare">https://<ip address></code> for the Insight instance.
   - Set **user** and **password** to the Insight user created, for example, OCI2SNOW.
   - Set **include_off_vms** to **false**
4. In the [SNOW] section, make the following changes:
   - Set **Instance** to the FQDN or ip address for your ServiceNow instance
   - Set **User** and **Password** to the ServiceNow service account user, for example, the OCI2SNOW.
   - Under **Field for the OCI URL**, set the **url** field to “u_oci_url”. This field is created as part of the connector OCI update set. You can change this in the customer environment, but if you do so, you need to modify it here and in ServiceNow. Best practice is to leave this field as is.
   - Set the **filter_status** field to “Installed, In Stock”. If you have a status that is different, you must set that status here in order to get all the records to match with Insight records prior to upload of new records. In most cases this field should remain unchanged.
   - Set **stale_status**to “Retired”.
5. The [Proxy] section is only required if you use a proxy server. If you need to use this section, ensure the following settings:
   - ;https = <a href="http://<host>:<port&gt" class="bare">http://<host>:<port&gt;
   - ;http = <a href="http://<host>:<port&gt" class="bare">http://<host>:<port&gt;
   - ;include_oci = True
   - ;include_snow = True
6. Edit the [Log] section only if you need deeper debug information.

7. To test the connector, open a cmd prompt as administrator and change to the \OCI2SNOW folder. Run the following command: `oci_snow_sync.pyz test`

   Details can be seen in the `logs\` folder.

Syncing the connector

Once ServiceNow, Insight and the connector are properly configured, you can synchronize the connector.

Steps

1. Open a cmd prompt and change to the \OCI2SNOW folder.
2. Run the following command twice. The first sync updates the items, the second sync updates the relationships: `oci_snow_sync.pyz sync`
3. Verify that the Storage Server table in your ServiceNow instance is populated. Open a storage server and verify that resources related to that storage are listed.

Scheduling synchronization to occur daily

You can use the Windows task scheduler to automatically sync the ServiceNow connector.

About this task

Automatic synchronization ensures that Insight data is regularly moved to ServiceNow. You can use any method for scheduling. The following steps use the Windows task scheduler to accomplish automatic syncing.

Steps

1. On the Windows screen, click **Start** and enter **run** > **task scheduler**.
2. Click **Create Basic Task…**
3. Enter a meaningful name, such as “OCI2SNOW Connector Sync”. Enter a description of the task. Click **Next**.
4. Select to run the task **Daily**. Click **Next**.
5. Choose a time of day to run the task. Click **Next**.
6. For Action, select **Start a program**. Click **Next**.
7. In the **Program/script** field, enter `C:\OCI2SNOW\oci_snow_sync_pyz`. In the **Arguments** field, enter `sync`. In the **Start in** field, enter `C:\OCI2SNOW`. Click **next**.
8. Review the Summary details, and click **Finish**.

   The synchronization is now scheduled to run daily.

Copyright information

Trademark information