Monitoring Amazon Web Services

Amazon Web Services PowerPack version 119

Table of Contents

Chapter 1

Introduction

Overview

This manual describes how to monitor Amazon Web Services (AWS) in SL1 using the *Amazon Web Services* PowerPack. It also describes the reports you can generate and the dashboards you can view after you collect data from AWS, as well as the Run Book Action and Automation policies you can use to automate certain aspects of monitoring AWS.

Use the following menu options to navigate the SL1 user interface:

- To view a pop-out list of menu options, click the menu icon ( ).
- To view a page containing all the menu options, click the Advanced menu icon (•••).

The following sections provide an overview of Amazon Web Services and the *Amazon Web Services* PowerPack:

NOTE: For more information about setting up a SL1 appliance on an Amazon Web Services EC2 instance, see the ***Installation and Initial Configuration*** manual.

NOTE: For more information about setting up an AWS Elasticsearch, Logstash, and Kibana (ELK) stack, see the ***Monitoring AWS ELK Stacks*** manual.

NOTE: ScienceLogic provides this documentation for the convenience of ScienceLogic customers. Some of the configuration information contained herein pertains to third-party vendor software that is subject to change without notice to ScienceLogic. ScienceLogic makes every attempt to maintain accurate technical information and cannot be held responsible for defects or changes in third-party vendor software. There is no written or implied guarantee that information contained herein will work for all third-party variants. See the End User License Agreement (EULA) for more information.

What is AWS?

Amazon Web Services is Amazon's "Infrastructure as a Service" offering. AWS includes multiple products (called ***Services***) including compute, DNS, networking, content delivery, analytics, storage, and database services, among many others.

What is an AWS Region?

An AWS region is a geographical area made up of availability zones located within that region. Each zone may have multiple data centers. Regions have a canonical naming scheme of:

country/continent-direction-number

For example, the 'us-east-1' region is located in the United States, on the east coast, and it is the #1 data center in that region.

AWS regions are also commonly referred to by the city or state in which the data center is located. For example, us-west-2 is commonly referred to as "Oregon", ap-northeast-1 is commonly referred to as "Tokyo", etc.

The Dynamic Applications in the *Amazon Web Services* PowerPack create a "region" component device for each discovered region. The component devices for regions include both the region name and city/state description. For example, the Dynamic Applications might discover a component device called "Oregon: us-west-2". Component devices that represent region-specific AWS services reside under the appropriate "region" component device and appropriate "zone" component device.

NOTE: For more information about AWS regions, see https://docs.amazonaws.cn/en_us/general/latest/gr/rande.html.

What is an AWS Zone?

All instances of an AWS service reside in one or more Zones. A zone is a physical network and power partition (air-gap firewall) within a regional data center. Some AWS instances, like EC2 instances, are in a single zone. Other AWS instances, like an SNS queue, exist in all zones simultaneously.

The AWS naming convention for a zone is:

region[a-z]

For example, zone 'a' for the region 'us-east-1' is named 'us-east-1a'.

When a user deploys a service instance, the user can specify a "zone preference", but the final zone for that service instance is decided by AWS, not the user.

The Dynamic Applications in the *Amazon Web Services* PowerPack create a "zone" component device for each discovered zone.

AWS services with a specific zone affinity reside under the appropriate zone component device. For example, the Dynamic Applications in the PowerPack might discover the zone "us-west-1b" and create a component device called "us-west-1b".

AWS services that are specific to a zone reside under the appropriate "region" component device and appropriate "zone" component device. The Dynamic Applications in the PowerPack create a "multi-zoned" component device for services that are inherently zone agnostic such as the Simple Queue Service (SQS).

Component devices that represent Zones are a named container with no associated performance metrics.

What Does the Amazon Web Services PowerPack Monitor?

To collect data from Amazon Web Services, the ScienceLogic Data Collector or All-In-One Appliance connects via HTTPS to the URLs listed in the following AWS document: http://docs.aws.amazon.com/general/latest/gr/rande.html.

The *Amazon Web Services* PowerPack includes Dynamic Applications that can monitor performance metrics and collect configuration data for the following AWS Services and components:

- API Gateways
- Aurora
- AutoScale
- CloudFront
- CloudTrail
- CloudWatch
- Direct Connect

- DynamoDB (DDB)
- ElastiCache
- Elastic Beanstalk
- Elastic Block Store (EBS)
- Elastic Compute Cloud (EC2)
- Elastic Container Services (ECS)
- Elastic File System (EFS)
- Elastic Kubernetes Service (EKS)
- Elastic Load Balancers (ELB)
- Elastic Map Reduce (EMR)
- Glacier
- IoT
- Key Management Service (KMS)
- Lambda
- Lightsail
- OpsWorks
- RedShift
- Relational Data Store (RDS)
- Route53
- Security Groups
- Shield
- Simple Email Service (SES)
- Simple Notification Service (SNS)
- Simple Queue Service (SQS)
- Simple Storage Service (S3)
- Storage Gateways (ASG)
- Storage Gateway Volumes
- Transit Gateways
- Virtual Private Cloud Service (VPC)
- Virtual Private Networks (VPN)
- Web Application Firewall (WAF)

NOTE: The following services are not monitored for GovCloud accounts:

- API Gateway private integrations
- CloudFront
- Lightsail
- OpsWorks
- Replica Lambda functions
- Shield
- Web Application Firewall

NOTE: Not all AWS services are supported by all AWS regions. For more information about which AWS services are supported by which AWS regions, see https://aws.amazon.com/about-aws/global-infrastructure/regional-product-services.

NOTE: To monitor performance metrics for an AutoScale group, you must activate detailed instance monitoring for that group. For instructions on how to perform this task, see http://docs.aws.amazon.com/AutoScaling/latest/DeveloperGuide/as-instance-monitoring.html.

NOTE: When monitoring EC2-backed ECS clusters, you can optionally use the *Docker* PowerPack to collect container information in addition to what the AWS API provides for the ECS service. For more information, see the section on ***Configuring AWS Integration with Docker***.

NOTE: To monitor Lambda services, you must first configure some of the Dynamic Applications in the *Amazon Web Services* PowerPack prior to discovery. For more information, see the ***Configuring "AWS Lambda Service Discovery"*** and ***Configuring "AWS Lambda Function Qualified Discovery"*** sections.

The Dynamic Applications in the PowerPack also monitor:

- The general health of each AWS service
- Current billing metrics for each service aligned with the account
- Custom, application-specific performance metrics configured on the account
- The state of any AWS Alarms set on metrics in Cloudwatch

In addition to Dynamic Applications, the PowerPack includes the following features:

- Event Policies and corresponding alerts that are triggered when AWS component devices meet certain status criteria

- Device Classes for each of the AWS component devices monitored
- Sample Credentials for discovering AWS component devices
- Reports and dashboards that display information about AWS instances and component devices
- Run Book Action and Automation policies that can automate certain AWS monitoring processes

NOTE: To view Amazon Web Services dashboards, you must first install the *Amazon Web Services: Dashboards* PowerPack. For more information, see the ***AWS Dashboards*** chapter.

Installing the Amazon Web Services PowerPack

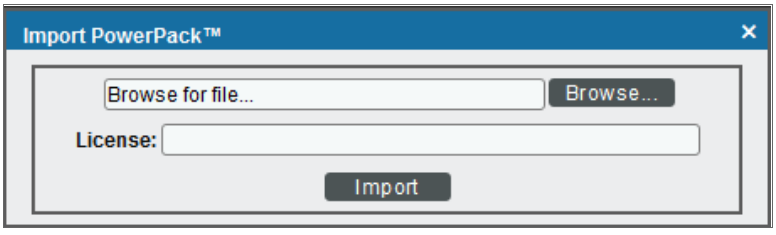
Before completing the steps in this manual, you must import and install the latest version of the *Amazon Web Services* PowerPack.

NOTE: If you are upgrading from an earlier version of the PowerPack, see the Release Notes for the version you are installing for upgrade instructions.

TIP: By default, installing a new version of a PowerPack overwrites all content from a previous version of that PowerPack that has already been installed on the target system. You can use the ***Enable Selective PowerPack Field Protection*** setting in the **Behavior Settings** page (System > Settings > Behavior) to prevent new PowerPacks from overwriting local changes for some commonly customized fields. (For more information, see the ***System Administration*** manual.)

To download and install a PowerPack:

1. Download the PowerPack from the ScienceLogic Support Site.
2. Go to the **PowerPack Manager** page (System > Manage > PowerPacks).
3. In the **PowerPack Manager** page, click the **[Actions]** button, then select *Import PowerPack*.
4. The **Import PowerPack** dialog box appears:



5. Click the **[Browse]** button and navigate to the PowerPack file.
6. When the **PowerPack Installer** modal appears, click the **[Install]** button to install the PowerPack.

NOTE: If you exit the **PowerPack Installer** modal without installing the imported PowerPack, the imported PowerPack will not appear in the **PowerPack Manager** page. However, the imported PowerPack will appear in the **Imported PowerPacks** modal. This page appears when you click the **[Actions]** menu and select *Install PowerPack*.

For information about opportunities and challenges with AWS, watch the video at https://sciencelogic.com/product/resources/whiteboard-aws-opportunities-challenges.

Chapter 2

Controlling What is Discovered by the PowerPack

Overview

The following sections describe the different methods to control what you can discover and monitor with the *Amazon Web Services* PowerPack:

Configuring AWS for Monitoring Regions with AWS Config Enabled

If your accounts have the AWS Config service enabled, then ScienceLogic recommends setting the ***Embed Value [%2]*** field in the **SOAP Options** section of the SOAP/XML credential you will create to "AUTO". The AWS Config service will then be used by SL1 to determine which regions and services are being used and only create the components needed. This will reduce the number of components created and will also reduce the load on the Data Collector.

NOTE: The Dynamic Applications "AWS: Account Resource Count Performance" and "AWS: Region Resource Count Performance" will only show data if the AWS Config service is enabled for those accounts/regions.

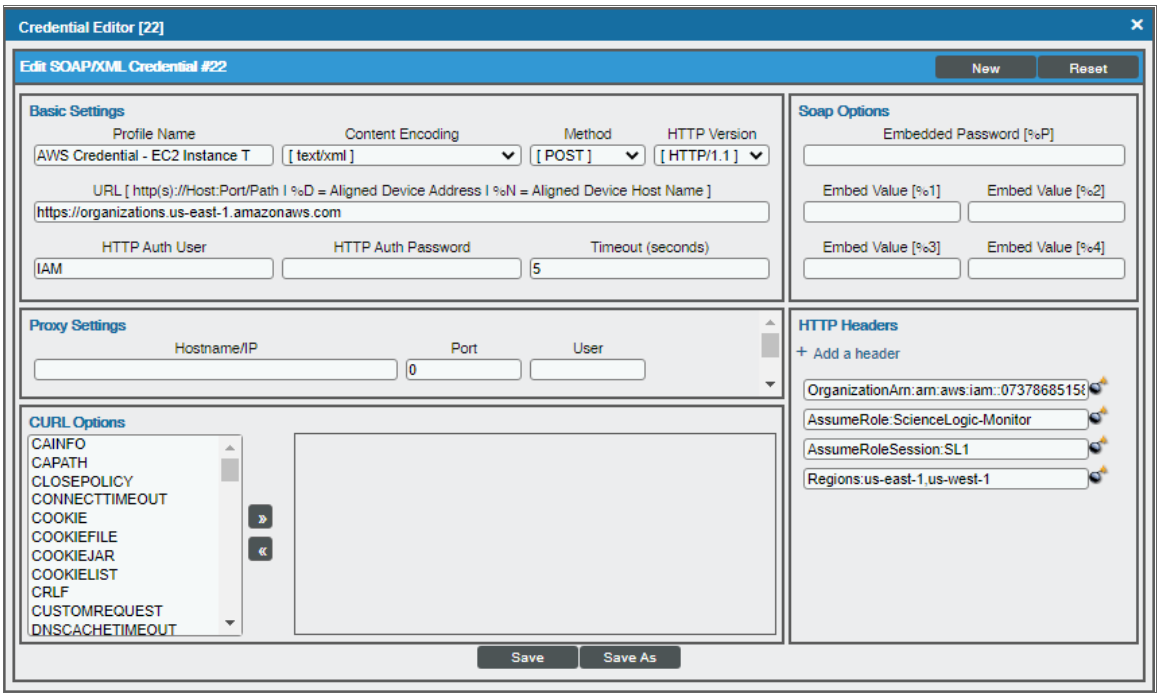
Configuring AWS for Monitoring Regions with AWS CloudWatch

If AWS config is not enabled, then ScienceLogic recommends setting the ***Embed Value [%2]*** in the **SOAP Options** section of the SOAP/XML credential you will create to "FILTER". This will use AWS Cloudwatch to determine which regions are reporting CloudWatch metrics and discover those regions. This will reduce the number of components created and will also reduce the load on the Data Collector.

The Regions Header in the SOAP/XML Credential

The ***Regions*** header is an optional header that can be inserted into the AWS SOAP/XML credential you will create to restrict which regions are discovered. This header supports a comma-separated list of regions that will be discovered and monitored. For example, the credential below shows the header with two specific regions. In this case, only those two regions would be discovered and monitored.

NOTE: The ***Regions*** header must not be included if "FILTER" or "AUTO" are used in the ***Embed Value %2*** field.



Using IAM Permissions to Restrict SL1 Access to Specific Regions and Services

You can use IAM policies in AWS to restrict which regions and services SL1 will monitor. To do this, you can create another IAM policy and apply that along with the SL1 monitoring policy to the applicable user or role(s).

To monitor specific regions and services, you must create a JSON policy in the AWS Management Console that uses the `NotAction`, `Allow`, and `Deny` policy elements to specify the regions and services you want to monitor as well as which regions and services you ***do not*** want to monitor.

NOTE: You must have at least Read-Only JSON policy permissions for the regions you want to monitor. You cannot discover regions for which you do not have policy permissions. At a minimum, you must at least have permissions for the us-east-1 (Virginia) region; without permissions for this region, you cannot discover general AWS services such as CloudFront, Route53, and OpsWorks.

TIP: When discovering resources in specific regions, you should ensure that any Global services or resources you want to monitor have the necessary access permissions.

NOTE: For more information about the `NotAction`, `Allow`, and `Deny` policy elements, see https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_policies_elements_notaction.html.

The following sections provide two examples of region-specific JSON policies.

Example 1: One Region

This JSON Policy will deny any service that is not in the us-east-1 region. As a result, SL1 will discover only components in the us-east-1 region.

NOTE: In addition to the code below, you would need to specify the other resource permissions you want to allow in the policy.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyAllOutsideUSEast1",
      "Effect": "Deny",
      "NotAction": [
        "iam:*",
        "organizations:*",
        "support:*",
        "aws-portal:*",
        "s3:ListAllMyBuckets"
      ],
      "Resource": "*",
      "Condition": {
        "StringNotEquals": {
          "aws:RequestedRegion": "us-east-1"
        }
      }
    }
  ]
}
```

Example 2: Multiple Regions

This JSON Policy will deny any service that is not in the us-east-1, us-west-2, and ap-northeast-1 regions. As a result, SL1 will discover only components in the us-east-1, us-west-2, and ap-northeast-1 regions.

NOTE: In addition to the code below, you would need to specify the other resource permissions you want to allow in the policy.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyAllOutsideUSWest2USEast1APNortheast1",
      "Effect": "Deny",
      "NotAction": [
        "iam:*",
        "organizations:*",
        "support:*",
        "aws-portal:*",
        "s3:ListAllMyBuckets"
      ],
      "Resource": "*",
      "Condition": {
        "StringNotEquals": {
          "aws:RequestedRegion": ["us-east-1", "us-west-2", "ap-northeast-1"]
        }
      }
    }
  ]
}
```

Configuring AWS for Monitoring Regions Using CloudWatch Namespaces

NOTE: These steps will be applied to all discovered AWS accounts on your SL1 system.

Users discovering with CloudWatch metrics can also discover regions where a specific namespace is available by editing the NAMESPACES field in the aws_region_discovery snippet in the "AWS: Region Discovery" Dynamic Application.

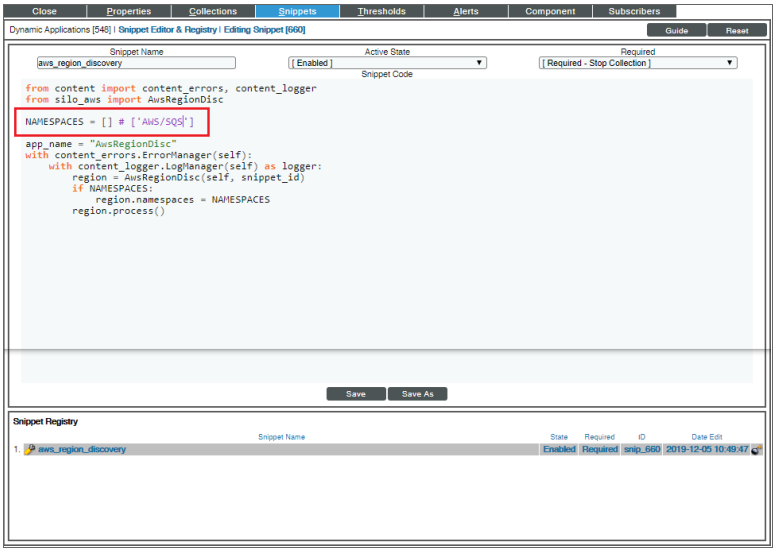
To edit the snippet:

1. Find the "AWS: Region Discovery" Dynamic Application in the **Dynamic Applications Manager** page (System > Manage > Applications) and click its wrench icon ().

2. Click the **[Snippets]** tab and then click the wrench icon () for the aws_regon_discovery snippet.
3. Edit the NAMESPACES field to include the namespace for your region. For example:

```
NAMESPACES = [ 'AWS/SQS' ]
```

4. Click **[Save]**.



Only regions that have services grouped in the specified namespace will be discovered. Global services will also be discovered.

NOTE: For more information about namespaces, see https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/viewing_metrics_with_cloudwatch.html.

Chapter 3

Configuration

Overview

Use the following menu options to navigate the SL1 user interface:

- To view a pop-out list of menu options, click the menu icon ( ).
- To view a page containing all the menu options, click the Advanced menu icon (•••).

The following sections describe several options available for using the *Amazon Web Services* PowerPack to monitor your AWS accounts.

Configuring AWS to Report Billing Metrics

To use the "AWS: Billing Performance Percent" Dynamic Application, your AWS account must meet the following requirements:

- The user account you supplied in the AWS credential must have permission to view the us-east-1 zone.
- Your AWS account must be configured to export billing metrics to the CloudWatch service.

If your AWS account is not configured to export billing metrics to the CloudWatch service, the "AWS: Billing Performance Percent" Dynamic Application will generate the following event:

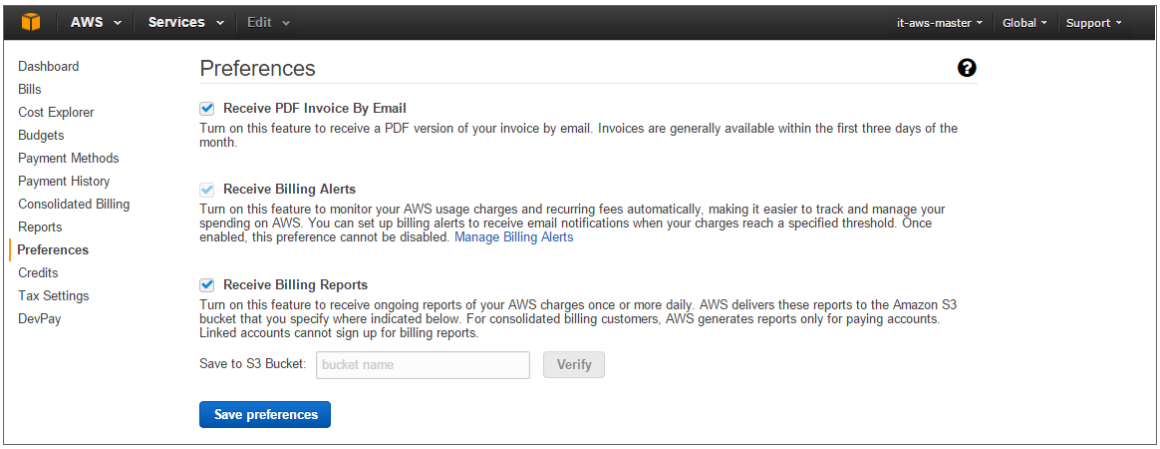
```
No billing metrics can be retrieved. Your AWS account is not configured to export
billing metrics into CloudWatch.
```

To configure your AWS account to export billing metrics to the CloudWatch service, perform the following steps:

1. Open a browser session and go to aws.amazon.com.

2. Click **[My Account]** and then select *Billing & Cost Management*. If you are not currently logged in to the AWS site, you will be prompted to log in:

3. After logging in, the **Billing & Cost Management Dashboard** page appears. In the left navigation bar, click **[Preferences]**. The **Preferences** page appears:



4. Select the ***Receive Billing Alerts*** checkbox.

> **CAUTION:** If you enable this option, this option cannot be disabled.

5. Click the **[Save Preferences]** button.

Filtering EC2 Instances By Tag

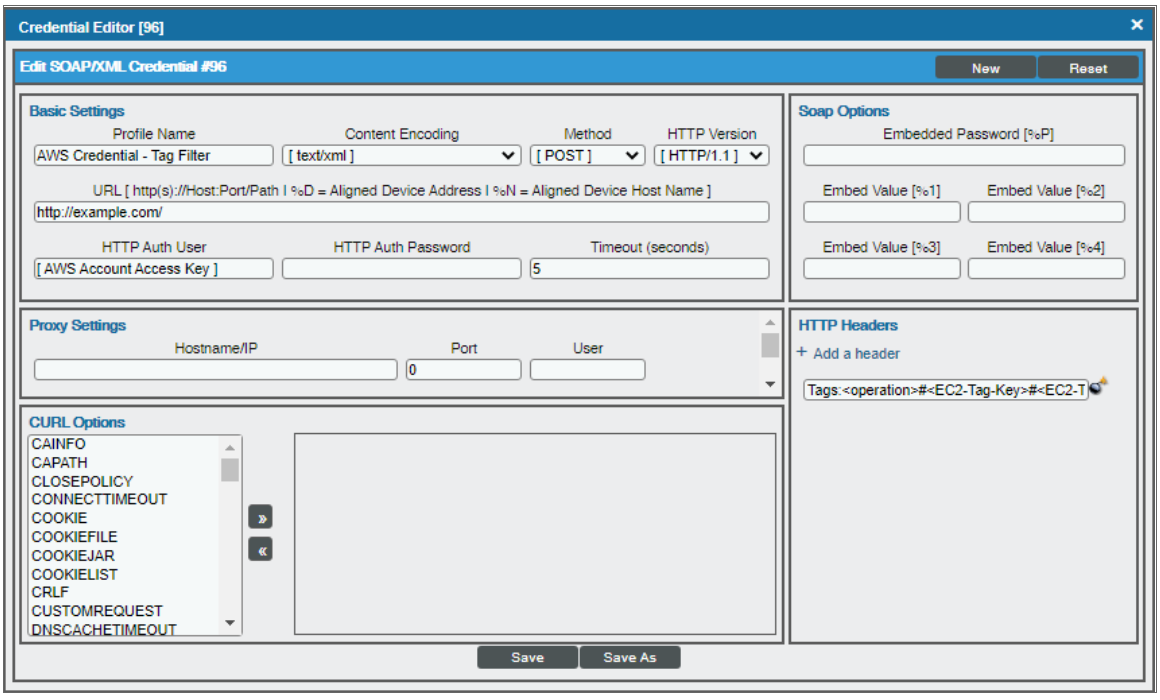
To discover EC2 instances and filter them by tag, you can use the "AWS Credential - Tag Filter" sample credential to enter EC2 tag keys and values.

> **NOTE:** Filtering EC2 instance by tag will apply to all accounts discovered.

> **NOTE:** Any EC2 instances that have already been discovered, but do not match the tag filter, will be set to "Unavailable."

To define an AWS credential to discover EC2 instances and filter them by tag:

1. Go to the **Credential Management** page (System > Manage > Credentials).
2. Locate the **AWS Credential - Tag Filter** sample credential and click its wrench icon (). The **Credential Editor** modal page appears:



3. Enter values in the following fields:

Basic Settings

- ***Profile Name***. Type a new name for your AWS credential.
- ***HTTP Auth User***. Type your AWS access key ID.
- ***HTTP Auth Password***. Type your AWS secret access key.

HTTP Headers

- Edit the HTTP header provided:
  - *Tags:<operation>#<EC2-Tag-Key>#<EC2-Tag-Value>*. Type the tag, followed by its operation, tag key, or tag value. For example, if you want to filter by Tag Name, you would type the following:

    ```
    Tags:equals#Name#Example
    ```

    Valid operations include:

    - equals
    - notEquals
    - contains
    - notContains

    You can chain together multiple filters separating them by a comma. For example:

    ```
    Tags:equals#Name#Example,contains#Owner#Someone
    ```

4. Click the **[Save As]** button, and then click **[OK]**.

Automatic SL1 Organization Creation

This feature is only applicable to the two discovery methods that use the Assume Role and automatically discover multiple accounts.

When multiple accounts are discovered, this feature places each account in its own SL1 organization. This feature requires an optional header in the SOAP/XML credential you will create. When this header is present, it will place each account into a new SL1 organization. When this header is not present, each account will be placed in the SL1 organization selected in the discovery session. The name of the organization can be controlled depending on what is provided in the header as follows:

- ***OrganizationCreation:NAME:ID***. Autocreates an SL1 organization for accounts using AssumeRole. You can enter one of the following options:
  - ***OrganizationCreation:NAME***. The name of the organization will contain the name of the user.
  - ***OrganizationCreation:ID***. The name of the organization will contain the ID of the user.

- ***OrganizationCreation:ID:NAME***. The name of the organization will contain both the ID and name of the user, in that order.
- ***OrganizationCreation:NAME:ID***. The name of the organization will contain both the name and ID of the user, in that order.

Monitoring Consolidated Billing Accounts

Consolidated billing is an option provided by Amazon that allows multiple AWS accounts to be billed under a single account. For more information about consolidated billing, see http://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/consolidated-billing.html.

If a consolidated billing account is monitored by SL1, the billing metrics associated with that account include only the consolidated amounts, per service. If you use consolidated billing and want to collect billing metrics per-account, you must discover each account separately. To monitor only the billing metrics for an AWS account, you can create credentials that include only billing permissions.

ScienceLogic Events and AWS Alarms

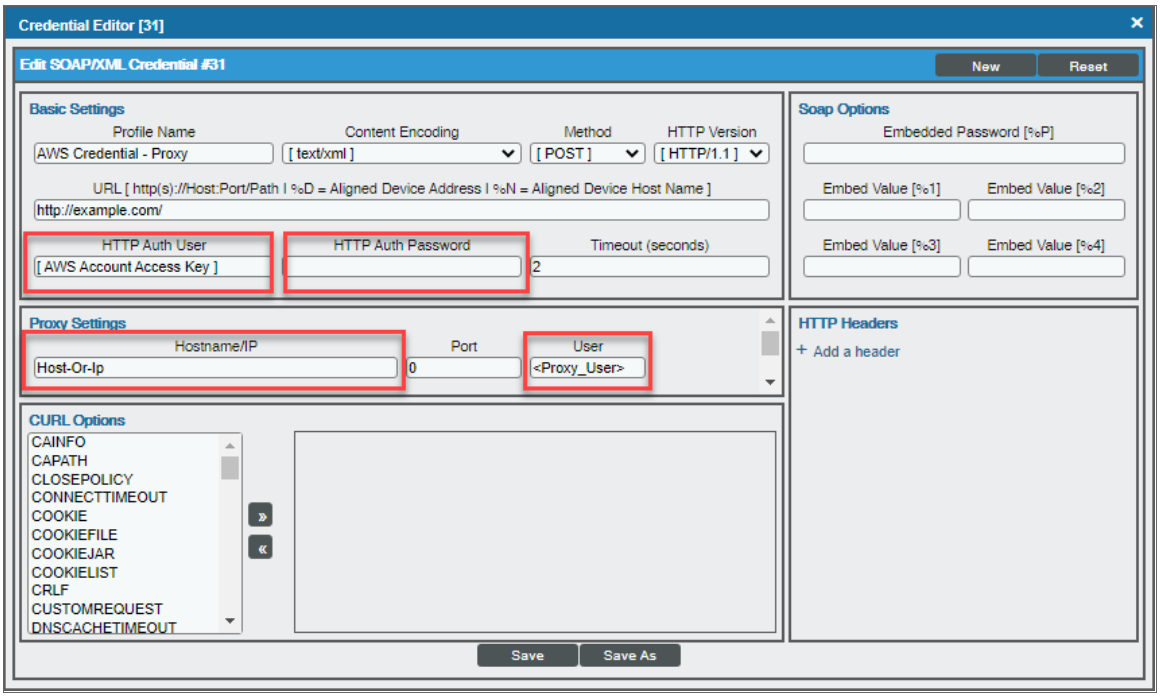
In addition to SL1 collecting metrics for AWS instances, you can configure CloudWatch to send alarm information to SL1 via API. SL1 can then generate an event for each alarm.

For instructions on how configure CloudWatch and SL1 to generate events based on CloudWatch alarms, see the *Configuring Inbound CloudWatch Alarms* section.

Using a Proxy Server

You can use a proxy server with the *Manual Discovery* and the *Automated Discovery Using AssumeRole with a Single IAM Key from the AWS Master Account* discovery methods.

To use a proxy server in both cases, you must fill in the proxy settings in the SOAP/XML credential.



For the *Automated Discovery Using AssumeRole with a Single IAM Key from the AWS Master Account* discovery method, if the proxy does not support ping passthrough you will also need to follow the steps in the ***Automated Discovery Using AssumeRole with a Single IAM Key from the AWS Master Account*** section without ping support.

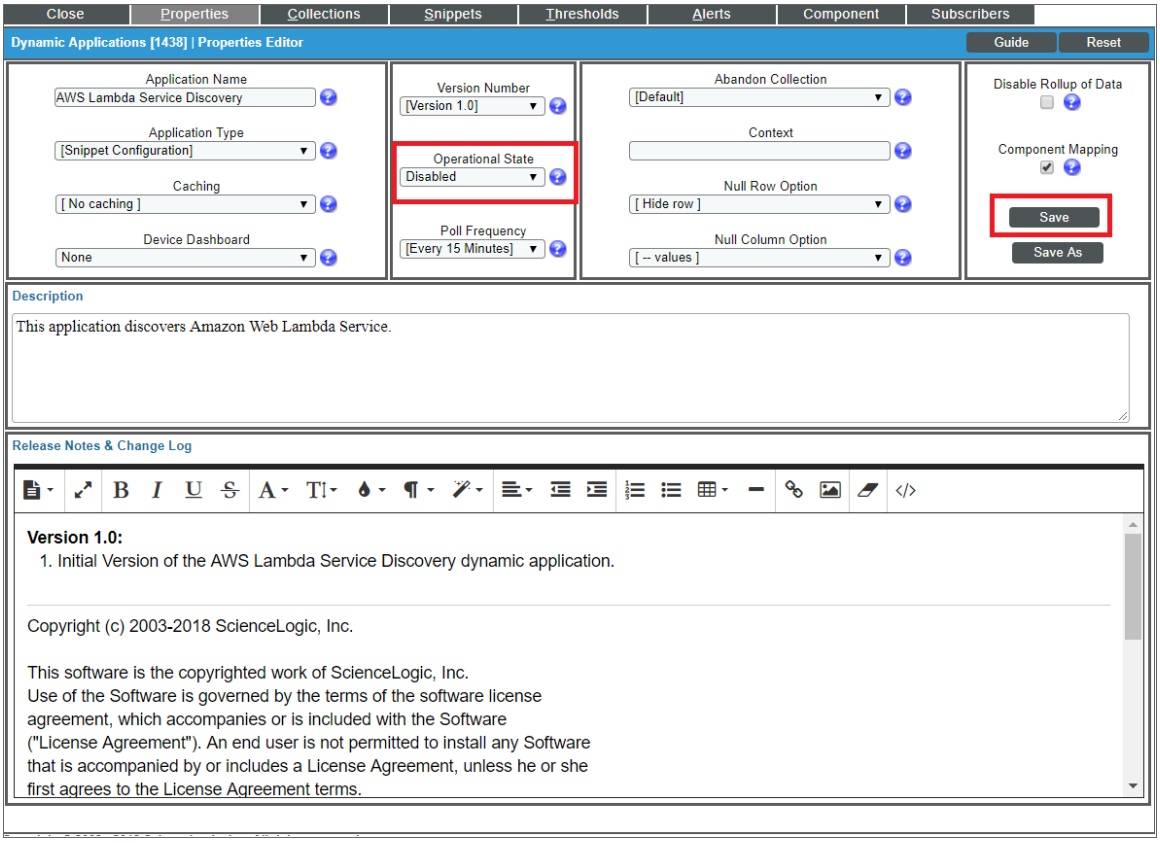
Configuring "AWS: Lambda Service Discovery"

By default, the "AWS: Lambda Service Discovery" Dynamic Application is configured to discover only regular Lambda functions, not replica functions. If you want to discover both regular and replica Lambda functions, then you must configure the "AWS: Lambda Service Discovery" Dynamic Application to do so ***prior*** to discovering your Lambda service.

To configure the "AWS: Lambda Service Discovery" Dynamic Application to discover both regular and replica Lambda functions:

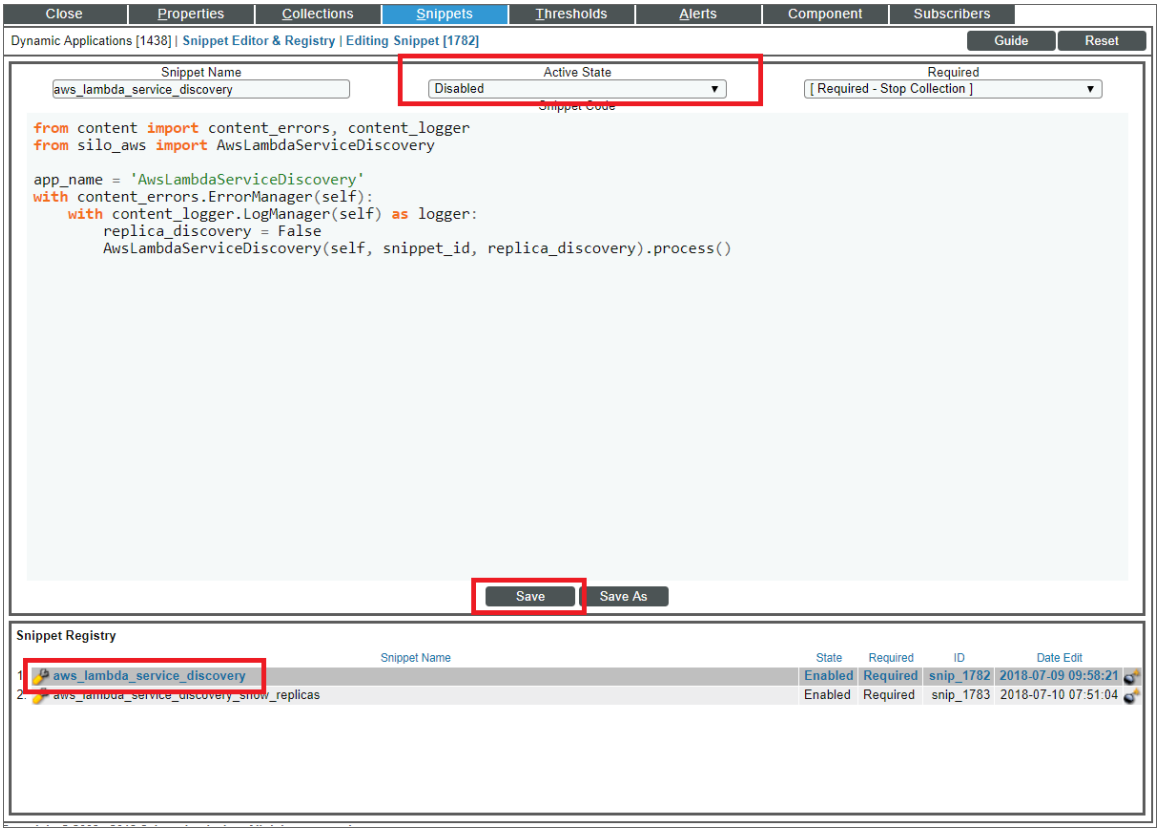
1. Go to the **Dynamic Applications Manager** page (System > Manage > Applications).
2. Locate the "AWS: Lambda Service Discovery" Dynamic Application and click its wrench icon (). The **Dynamic Applications Properties Editor** page appears.

3. In the ***Operational State*** field, select *Disabled*, and then click **[Save]**. This disables the Dynamic Application from collecting data.



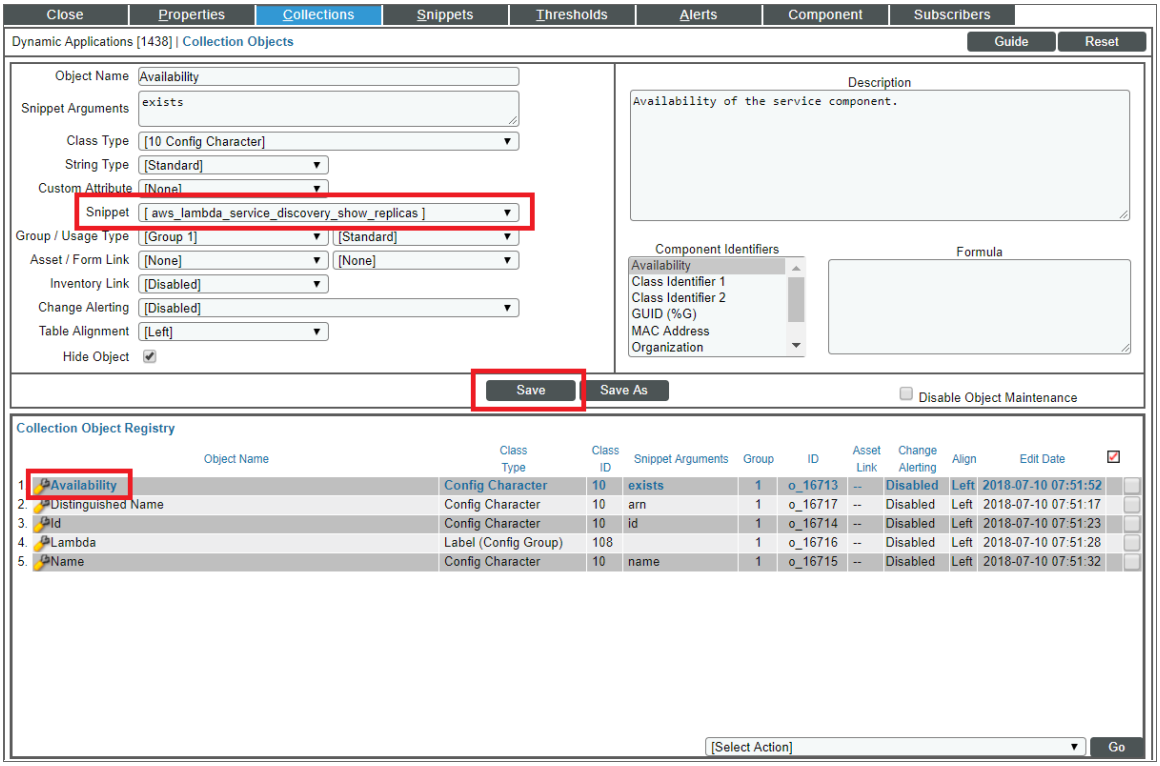
4. Click the **[Snippets]** tab. The **Dynamic Applications Snippet Editor & Registry** page appears.
5. In the **Snippet Registry** pane, click the wrench icon () for the "aws_lambda_service_discovery" snippet.

6. In the ***Active State*** field, select *Disabled*, and then click **[Save]**. This disables the "aws_lambda_service_discovery" snippet.



7. In the **Snippet Registry** pane, click the wrench icon ( ) for the "aws_lambda_service_discovery_show_replicas" snippet.
8. In the ***Active State*** field, select *Enabled*, and then click **[Save]**. This enables the "aws_lambda_service_discovery_show_replicas" snippet.
9. Click the **[Collections]** tab. The **Dynamic Applications | Collections Objects** page appears.

10. Click the wrench icon ( ) for the first Collection Object listed in the **Collection Object Registry** pane, select *aws_lambda_service_discovery_show_replicas* in the ***Snippet*** field for that Collection Object, and then click **[Save]**.



11. Repeat step 10 for all of the remaining Collection Objects listed in the **Collection Object Registry** pane.
12. Click the **[Properties]** tab.
13. In the ***Operational State*** field, select *Enabled*, and then click **[Save]**. This re-enables data collection for the Dynamic Application.

NOTE: If you configure the "AWS: Lambda Service Discovery" Dynamic Application to discover both regular and replica Lambda functions, then when you run discovery, the Dynamic Applications in the *Amazon Web Services* PowerPack will create ***parent/child relationships*** between replica Lambda functions and their corresponding master Lambda functions. In this scenario, the ***Device View and other device component maps*** will display the relationship in this order: Lambda Function Service > Lambda Replica Function > Master Lambda Function. The replica appears as the parent to the master Lambda function because the replica could be in the same or a different region than the master Lambda function.

Configuring "AWS: Lambda Function Qualified Discovery"

By default, the "AWS: Lambda Function Qualified Discovery" Dynamic Application is configured to discover and model all Lambda alias components. An ***alias*** is a qualifier inside an AWS Lambda function that enables the user to control which versions of the Lambda function are executable—for instance, a production version and a test version.

When the "AWS: Lambda Function Qualified Discovery" Dynamic Application is configured to discover alias components, SL1 collects data only for the Lambda function versions specified in the alias.

Depending on your needs, you can optionally configure the Dynamic Application to instead do one of the following:

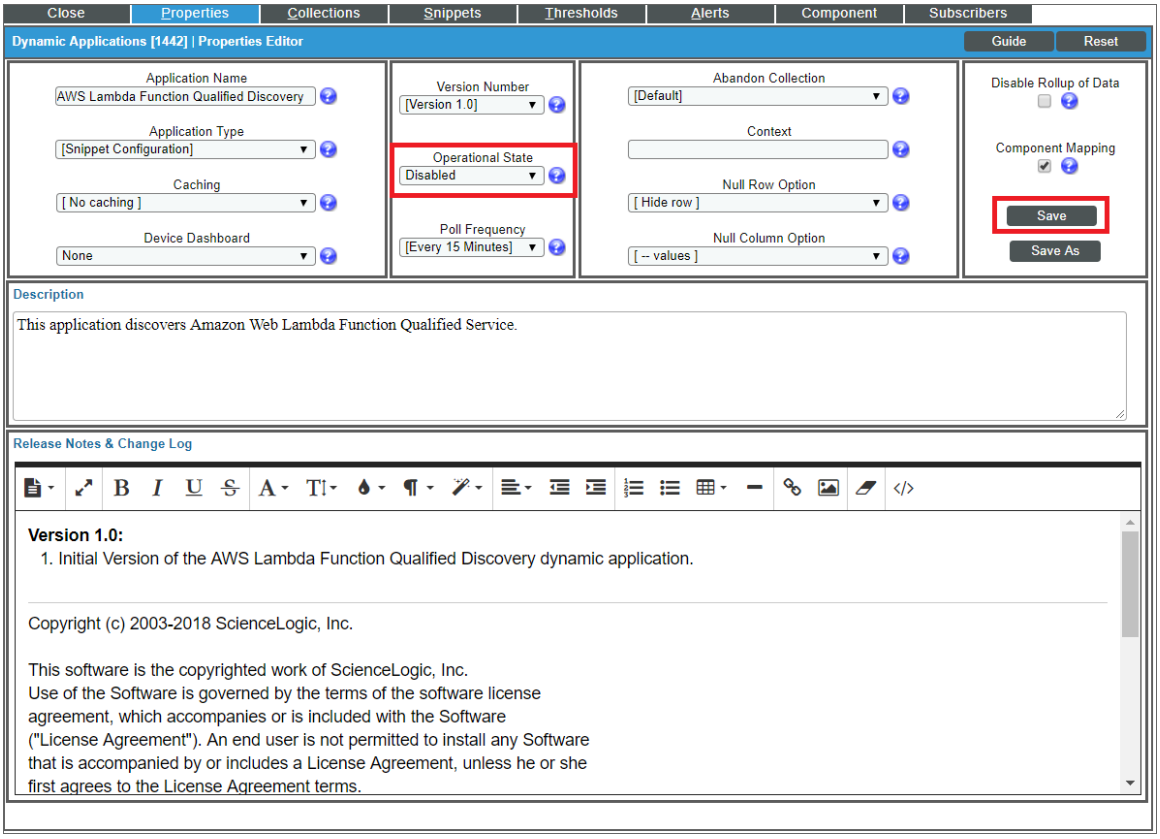
- Discover and model all Lambda version components. If you select this configuration, SL1 collects data for all existing versions of the Lambda function.
- Discover and model only Lambda version components with AWS configurations filtered by a trigger. If you select this configuration, SL1 collects data only for versions of the Lambda function that have triggers or are specified in an alias.

NOTE: If you have *configured the "AWS: Lambda Service Discovery" Dynamic Application* to discover both regular and replica Lambda functions and you want SL1 to *create dynamic component map relationships* between replica Lambda functions and their parent Lambda function versions, you must follow these instructions to configure the "AWS: Lambda Function Qualified Discovery" Dynamic Application to discover and model all Lambda version components.

To configure the "AWS: Lambda Function Qualified Discovery" Dynamic Application:

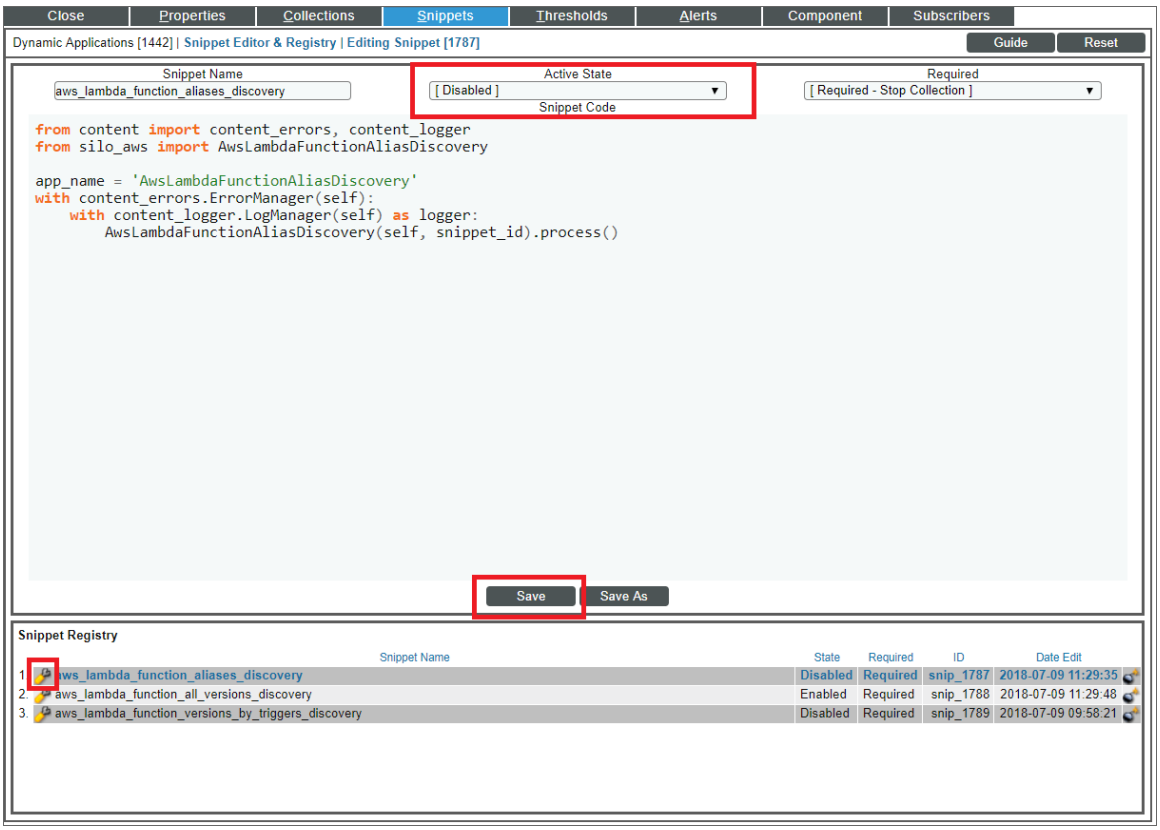
1. Go to the **Dynamic Applications Manager** page (System > Manage > Applications).
2. Locate the "AWS: Lambda Function Qualified Discovery" Dynamic Application and click its wrench icon ( ). The **Dynamic Applications Properties Editor** page appears.

3. In the ***Operational State*** field, select *Disabled*, and then click **[Save]**. This disables the Dynamic Application from collecting data.



4. Click the **[Snippets]** tab. The **Dynamic Applications Snippet Editor & Registry** page appears. The **Snippet Registry** pane includes the following snippets:
   - *aws_lambda_function_aliases_discovery*. When this snippet is enabled, the Dynamic Application discovers all Lambda alias components.
   - *aws_lambda_function_all_versions_discovery*. When this snippet is enabled, the Dynamic Application discovers all Lambda version components.
   - *aws_lambda_function_versions_by_triggers_discovery*. When this snippet is enabled, the Dynamic Application discovers Lambda version components with AWS configurations containing a trigger or those with an alias.

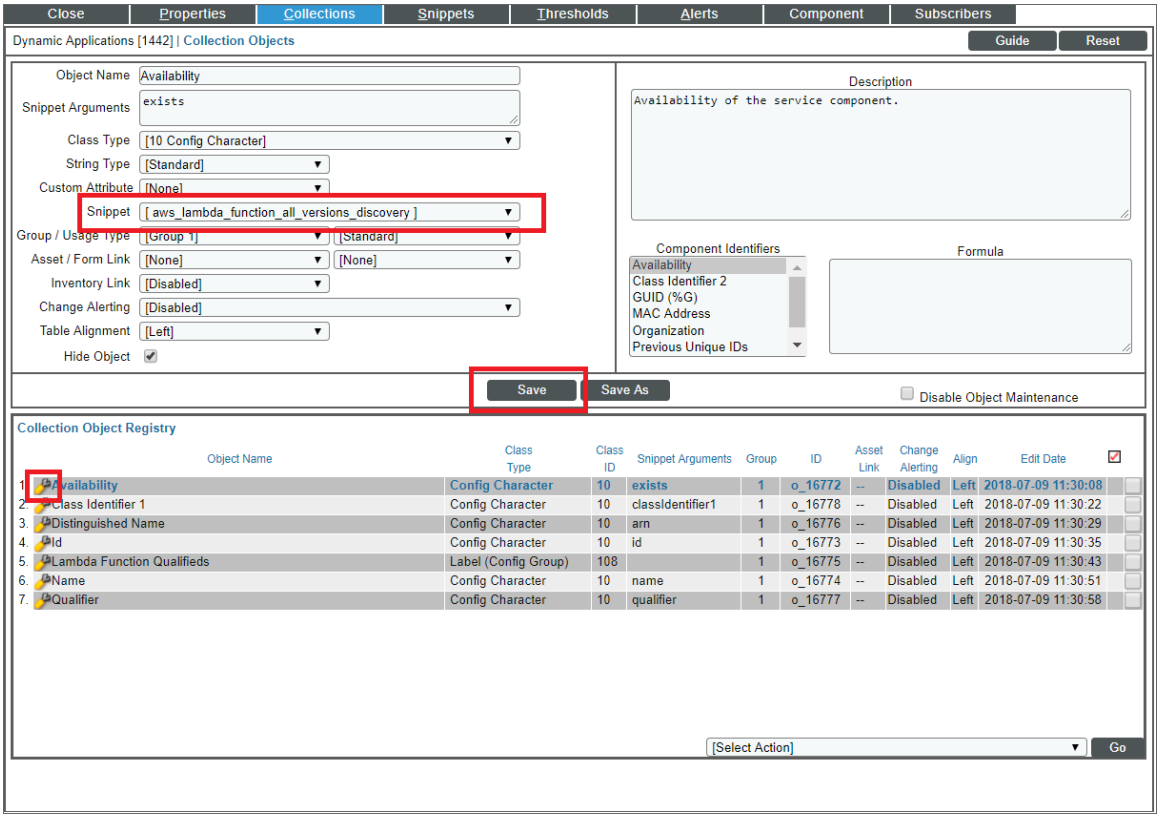
5. One at a time, click the wrench icon ( ) for each of the snippets, select *Enabled* or *Disabled* in the ***Active State*** field, and then click **[Save]** to enable the appropriate snippet and disable the others.



NOTE: You can enable only one of these snippets at a time.

6. Click the **[Collections]** tab. The **Dynamic Applications | Collections Objects** page appears.

7. Click the wrench icon ( ) for the first Collection Object listed in the **Collection Object Registry** pane, select the snippet you enabled in step 5 in the ***Snippet*** field for that Collection Object, and then click **[Save]**.



8. Repeat step 7 for all of the remaining Collection Objects listed in the **Collection Object Registry** pane.
9. Click the **[Properties]** tab.
10. In the ***Operational State*** field, select *Enabled*, and then click **[Save]**. This re-enables data collection for the Dynamic Application. The next time discovery is run, new component devices might be discovered and some previously discovered components might become unavailable, depending on how you configured the Dynamic Application.

NOTE: If you configure the "AWS: Lambda Function Qualified Discovery" Dynamic Application to discover Lambda alias or version components and your AWS service includes an API Gateway that triggers a Lambda Function, then the Dynamic Applications in the *Amazon Web Services* PowerPack will create *a device relationship* between that Lambda Function and its corresponding Lambda alias or version component device.

Configuring AWS Integration with Docker

If you have discovered EC2-backed ECS clusters using the *Amazon Web Services* PowerPack, you can optionally use the *Docker* PowerPack to collect container information in addition to what the AWS API provides for the ECS service.

NOTE: This integration does not work with Fargate-backed ECS clusters.

To configure this integration, cURL version 7.40 or later must be installed on the ECS AMI image. For example, the 2018.03 ECS AMI image is compatible is compatible because it includes cURL 7.43.1.

Additionally, you must install the most recent version of the *Docker* PowerPack on your SL1 System and run a discovery session using an SSH credential that will work on the EC2 host(s). This discovery session will discover the EC2 instances that comprise the ECS cluster and align the Docker host Dynamic Applications with those EC2 instances. Optionally, you can merge the EC2 host with the Docker host if you so choose.

NOTE: For more information about the *Docker* PowerPack, including instructions about creating the SSH credential and running discovery, see the ***Monitoring Docker*** manual.

NOTE: ScienceLogic does not recommend enabling and securing the Docker HTTP API when aligning EC2 instances with Docker hosts. Doing so requires you to complete manual steps on each EC2 host. Furthermore, if you use this method and then merge the EC2 host with the Docker host, data collection will fail for all containers that are children of the merged host.

Configuring AWS Integration with Kubernetes

If you are using the AWS EKS service you can optionally use the *Kubernetes* PowerPack to provide visibility into your Kubernetes worker nodes and their associated workloads.

To use the *Kubernetes* PowerPack with the *Amazon Web Services* PowerPack, you must have the following versions of these PowerPacks installed:

- *Amazon Web Services* version 118 or later
- *Kubernetes* version 104 or later

If you are using AWS EKS but do ***not*** want to use this feature, then it is recommended to disable the "AWS EKS Cluster Virtual Discovery" Dynamic Application. To do this:

1. Go to the **Dynamic Applications Manager** page (System > Manage > Dynamic Applications).
2. Search for "AWS EKS" in the ***Dynamic Application Name*** column.

3. Click on the wrench icon () for the "AWS EKS Cluster Virtual Device Discovery" Dynamic Application and set the ***Operational State*** dropdown to *Disabled*.
4. Click the **[Save]** button.

Using the *Kubernetes* PowerPack is completely automated on SL1. If the proper credentials have been assigned on AWS and the AWS EKS Cluster, then SL1 will automatically discover the Kubernetes worker nodes and the associated workloads. The following additional components will be automatically created:

1. A new DCM tree root device to represent the Kubernetes cluster. This will be a virtual device of the type "Kubernetes Cluster".
2. A child component of the cluster will be created for each worker node in the cluster. This will be a component device of the type "Kubernetes Node".
3. A child component of the cluster will be created that represents the Namespaces. This will be a component device of the type "Kubernetes Namespace Folder".
4. A child component of the Namespace Folder will be created for each Namespace discovered. This will be a component device of the type "Kubernetes Namespace".
5. A child component of the Namespace will be created for each controller discovered as follows:
    - Kubernetes Daemon Set
    - Kubernetes Deployment

    NOTE: At most only a single component is created to represent a controller. If a deployment and replica set exists, SL1 models only the deployment and replica set info as provided by the deployment component.

    - Kubernetes Job
    - Kubernetes Cronjob
    - Kubernetes Replication Controller
    - Kubernetes Replication Set
    - Kubernetes Stateful Set
6. A child component of the cluster will be created for each ingress defined. This will be a component device of the type "Kubernetes: Ingress".

For SL1 to automatically discovery the EKS cluster, you must perform the following steps:

NOTE: When logging into the Kubernetes cluster, ensure that the AWS credentials that `kubectl` is using are already authorized for your cluster. The IAM user that created the cluster has these permissions by default.

1. *Enable the Prometheus Metrics Server*. AWS EKS does not have the metrics server enabled by default. This is highly recommended as it will provide CPU and memory utilization metrics for both the worker nodes as well as the pods.

NOTE: SL1 automatically aggregates the CPU and memory utilization for pods and presents data at the controller level.

2. *Define the cluster role* needed by SL1 so that it can access the necessary APIs. This is done on the EKS Cluster.
3. *Define the ClusterRoleBinding*. This is done on the EKS Cluster.
4. *Map the IAM user or role to the RBAC role and groups* using the aws-auth ConfigMap. This is done on the EKS Cluster.

Enabling the Prometheus Metrics Server

The Prometheus Metrics Server is required to provide CPU and memory utilization for pods and for nodes. The metrics server can be easily installed on Kubernetes clusters with the following:

```
kubectl apply -f https://github.com/kubernetes-sigs/metrics-
server/releases/latest/download/components.yaml
```

To verify that the server is running, execute the command:

```
kubectl get deployment metrics-server -n kube-system
```

The following output will show that the metrics server is running:

```
NAME             READY  UP-TO-DATE  AVAILABLE  AGE
 metrics-server   1/1    1           14h
```

Define the Cluster Role

The cluster role defines the minimum permissions that SL1 needs to monitor the Kubernetes cluster. ClusterRole is used as it provides access to all namespaces. Since SL1 is directly monitoring the Kubernetes cluster via the Kuberneties API, this role's permissions need to be defined on the cluster itself.

To define the cluster role in Kubernetes:

1. Log in to the EKS cluster with the same user or role that created the cluster.
2. Create a new file called `SL1_cluster_role.yaml` and cut and paste the following text into that file:

```
adpiVersion: rbac.authorization.k8s.io/v1
 kind: ClusterRole
 metadata:
   name: eks-readonly-clusterrole
rules:
- apiGroups:
  - ""
  resources:
  - nodes
  - namespaces
```

```
  - pods
  - replicationscontrollers
  - events    {
  - persistentvolumes
  - persistentvolumeclaims
  - componentstatuses
  - services
  verbs:
  - get
  - list
  - watch
-apiGroups:
  - apps
  resources:
  - deployments
  - daemonsets
  - statefulsets
  - replicasets
  verbs:
  - get
  - list
  - watch
- apiGroups:
  - batch
  resources:
  - jobs
  - cronjobs
  verbs:
  - get
  - list
  - watch
- apiGroups:
  - metrics.k8s.io
  resources:
  - nodes
  - pods
  verbs:
  - get
  - list
  - watch
- apiGroups:
  - networking.k8s.io
  resources:
  - ingresses
  verbs:
  - get
  - list
  - watch
- apiGroups:
  - autoscaling
  resources:
  - horizontalpodautoscalers
  verbs:
  - get
  - list
  - watch
```

The above file defines the minimum read-only permissions needed for SL1 to monitor Kubernetes.

3. Once the file is defined, execute the following command to apply the file:

```
kubect1 apply -f cluster_role.yaml
```

Define the ClusterRoleBinding

Once the role is defined, it must be bound to users, groups, or services. This is done by defining a ClusterRoleBinding:

1. Log in to the EKS cluster with the same user or role that created the cluster.
2. Create a new file called `SL1_ClusterRoleBinding.yaml` and cut and paste the following text into that file:

```
apiVersion: rbac.authorization.k8s.io/v1
 kind: ClusterRoleBinding
 metadata:
  name: eks-cluster-role-binding
subjects:
 - kind: User
   name: Sciencelogic-Monitor
   apiGroup: rbac.authorization.k8s.io
roleref:
 kind: ClusterRole
 name: eks-readonly-clusterrole
 apiGroup: rbac.authorization.k8s.io
```

3. Once the file is created, apply the ClusterRoleBinding by executing the following command:

```
kubectl apply -f SL1_ClusterRoleBinding.yaml
```

NOTE: Under subjects, "name: Sciencelogic-Monitor" defines the Kubernetes user and it must match the username field in the config map shown below.

NOTE: Under roleRef, "name: eks-readonly-clusterrole" must match the name defined in the cluster role.

Map the IAM User or Role to the Kubernetes RBAC Role

After defining the ClusterRoleBinding, you must map the AWS credentials that SL1 is using to the username created above in the `SL1_ClusterRoleBinding.yaml` file. To do this, perform the following steps:

1. Enter the `kubectl edit -n kube-system configmap/aws-auth` command. This will bring up the `configmap`. How the `configmap` is updated depends on what type of IAM was used to discover SL1.

NOTE: If the `configmap/aws-auth` does not exist, follow the procedures defined in https://docs.aws.amazon.com/eks/latest/userguide/add-user-role.html

Example 1

If SL1 has discovered your AWS organization using assume role, add the following text to the `mapRoles:` section in the `configmap`:

```
- groups:
   - eks-cluster-role-binding
   rolearn:arn:aws:iam::<Account number that hosts the Kubernetes cluster-
>:role/Sciencelogic-Monitor
   username: Sciencelogic-Monitor
```

NOTE: If `mapRoles` does not exist, then you can add the `mapRoles` section to the `configmap`.

The text should appear in the `configmap` as the highlighted text below:

```
# Please edit the object below. Lines beginning with a '#' will be ignored,
 # and an emty file will abort the edit. If an error occurs while saving, this fiel
will be
 # reopened with the relevant failures
 #
apiVersion: v1
data:
 mapRoles: |
  - groups:
    - system:bootstrappers
    - system:nodes
    rolearn: arn:aws-us-gov:iam::<account number>:role/eksctl-eks-cluster-testfriday-
nod-NodeInstanceRole-6VCMS669U9NA
    username: system:node:{{EC2PrivateDNSName}}
   - groups:
     - eks-cluster-role-binding
     rolearn: arn:aws:iam::<account number>:role/Sciencelogic-Monitor
     username: Sciencelogic-Monitor
kind: ConfigMap
metadata:
 creationTimestamp: "2021-07-30T20:43:55Z"
 name: aws-auth
 namespace: kube-system
 resourceVersion: "173718"
 selfLink: /api/v1/namespaces/kube-system/configmaps/aws-auth
 uid: d1bcdafd-fc40-44e6-96d4-9a079b407d06
```

Example 2

If SL1 has been discovered with a single IAM key for the account, add the following text to the `mapUsers:` section of the `configmap`:

```
- groups:
   - eks-cluster-role-binding
   userarn:arn:aws:iam::<Account number that hosts the Kubernetes cluster>:user/<Name
of the user associated with the IAM key
   username: Sciencelogic-Monitor
```

The text should appear in the `configmap` as the highlighted text below:

```
# Please edit the object below. Lines beginning with a '#' will be ignored,
 # and an emty file will abort the edit. If an error occurs while saving, this fiel
will be
 # reopened with the relevant failures
 #
apiVersion: v1
data:
 mapRoles: |
  - groups:
    - system:bootstrappers
    - system:nodes
    rolearn: arn:aws-us-gov:iam::<account number>:role/eksctl-eks-cluster-testfriday-
nod-NodeInstanceRole-6VCMS669U9NA
    username: system:node:{{EC2PrivateDNSName}}
 mapUsers: |
  - groups:
    - eks-cluster-role-binding
    userarn: arn:aws:iam::<account number>:user/<username>
    username: Sciencelogic-Monitor
kind: ConfigMap
metadata:
 creationTimestamp: "2021-07-30T20:43:55Z"
 name: aws-auth
 namespace: kube-system
 resourceVersion: "173718"
 selfLink: /api/v1/namespaces/kube-system/configmaps/aws-auth
 uid: d1bcdafd-fc40-44e6-96d4-9a079b407d06
```

NOTE: In `userarn: arn:aws:iam::<account number>:user/<username>`, the username is the userarn that SL1 is using to monitor the Kubernetes cluster.

NOTE: Under `mapUsers`, the `username:` is the name used in the ClusterRoleBinding.

Amazon API Throttling Events

By default, SL1 will use the Collector Group aligned with the root AWS virtual device to retrieve data from AWS devices and services.

If SL1 must collect data from a large set of AWS devices and services, SL1 might generate Notify events with a message ending in the text "Retry #1-10 Sleeping: ... seconds". SL1 generates these events when the Amazon API throttles collection in response to a large number of requests to the API. Even though SL1 is generating Notify "Retry" events, SL1 is still collecting data from AWS. This issue commonly occurs when a specific Amazon data center edge is close to capacity.

If SL1 generates the Minor event "Collection missed on <device> on 5 minute poll", this indicates that SL1 was unable to retrieve that specific datum from the Amazon cloud during the most recent five-minute polling cycle. If you frequently see the "Collection missed" event across your cloud, you must contact Amazon support to whitelist the IP address of your Data Collector. This will prevent further throttling from occurring.

Support for AWS China Regions

Currently, the only method of discovery for AWS China Regions is the *Manual Discovery* method. In this case, the ***Embed Value %1*** field in the *SOAP/XML credential* must contain the specific Chinese region to be monitored.

Support for AWS GovCloud Regions

AWS GovCloud Regions can be discovered using all discovery methods as defined below:

- For an individual account using the *Manual Discovery* method, type the name of the AWS GovCloud region in the ***Embed Value %1*** field in the *SOAP/XML credential*.
- For those using one of the discovery methods with AssumeRole, enter one of the following URLs in the ***URL*** field of the *SOAP/XML credential* to specify the specific government region:
    - https://organizations.us-gov-west-1.amazonaws.com
    - https://organizations.us-gov-east-1.amazonaws.com

NOTE: All examples shown are for commercial AWS accounts. When AWS Gov is being monitored, the JSON data that refers to ARN will need to be modified from "aws" to "aws-us-gov". For example: `Resource": "arn:aws:iam::<account number>:role/Sciencelogic-Monitor` would need to be `Resource": "arn:aws:iam-us-gov::<account number>:role/Sciencelogic-Monitor`

Migrating from Using an IAM Key Per Account to Using AssumeRole

SL1 supports the ability to migrate accounts that were originally discovered using an IAM key per account to start using AssumeRole. To upgrade, perform the following steps:

1. Run steps 1-4 in the *Automated Discovery when the Data Collector Runs as an EC2 Instance* section

   or

   Run steps 1-3 of the *Automated Discovery Using AssumeRole with a Single IAM Key from the AWS Master Account* section.
2. Next you must disable collection for every account that is being migrated. To do this, go to the **Device Components** page (Devices > Device Components, or Registry > Devices > Device Components in the SL1 classic user interface) and enter "AWS | Service" in the ***Device Class | Sub-class*** column.
3. Select the checkbox for each account being migrated.
4. In the ***Select Action*** menu, select *_Disabled* under **Change Collection State**. Click **[Go]**.

WARNING: Failing to disable collection for accounts that will be migrated will result in a loss of data.

If you have a large number of accounts that will be migrated, it is recommended to start with a single account to ensure that all settings migrate correctly. To limit the accounts that are migrated, put only a single account in the ec2-collector policy so SL1 will assume the role for that single account. Once you have ensured that the one account has been migrated successfully, you can add the other accounts back into the ec2-collector policy.

NOTE: If you had previously changed the account name in SL1, the upgrade process will overwrite any changes made to the name of the account component.

NOTE: If you had previously placed your accounts into different SL1 organizations, these organization names will be preserved upon upgrading. However, if you add the headers for SL1 organizations to your SOAP/XML credential, then the SL1 organizations will be set according to the SOAP header.

Once discovery has completed successfully, the Dynamic Applications aligned to the root device and the component device will display the new credential used in the discovery process, while the Dynamic Applications for the child device(s) of the account device will still display the old credential. While you cannot delete the credentials, you can remove the IAM keys from those credentials as they are no longer being used by the Dynamic Applications.

Chapter 4

Minimum Permissions

Overview

The following sections describe the minimum permissions that must be set before you can run discovery with the *Amazon Web Services* PowerPack:

Minimum Permissions Needed to Monitor Your AWS Accounts

The following table displays the minimum permissions required for Dynamic Applications in the *Amazon Web Services* PowerPack to collect data.

Service	Actions	
API Gateway	Read	GET
CloudFront	List	ListDistributions ListInvalidations ListStreamingDistributions
	Read	GetDistribution GetStreamingDistribution
CloudTrail	List	DescribeTrails
	Read	GetTrailStatus

Service	Actions	
CloudWatch	List	ListMetrics
	Read	DescribeAlarmHistory DescribeAlarms GetMetricData GetMetricStatistics
Direct Connect	Read	DescribeConnections DescribeTags DescribeVirtualInterfaces
DynamoDB	List	ListTables
	Read	DescribeTable
EC2	List	DescribeAvailabilityZones DescribeImages DescribeInstances DescribeNatGateways DescribeRegions DescribeRouteTables DescribeSecurityGroups DescribeSubnets DescribeSnapshots DescribeTransitGatewayRouteTables DescribeTransitGateways DescribeTransitGatewayAttachments DescribeVolumes DescribeVpcPeeringConnections DescribeVpcs DescribeVpnGateways
	Read	DescribeVpnConnections
EC2 Auto Scaling	List	DescribeAutoScalingGroups DescribeAutoScalingInstances DescribeLaunchConfigurations
EFS	List	DescribeFileSystems
Elastic Beanstalk	List	DescribeEnvironments
	Read	DescribeConfigurationSettings DescribeEnvironmentResources DescribeEnvironmentHealth DescribeInstancesHealth
Elastic Container Services (ECS)	List	ListClusters ListContainerInstances ListServices ListTasks
	Read	DescribeClusters DescribeContainerInstances DescribeServices DescribeTaskDefinition

Service	Actions	
		DescribeTasks
ElasticCache	List	DescribeCacheClusters
Elastic Kubernetes Service (EKS)	List	ListClusters
	Read	DescribeClusters
ELB	List	DescribeLoadBalancers
	Read	DescribeTags
ELB v2	Read	DescribeListeners DescribeLoadBalancers DescribeTags DescribeTargetGroups DescribeTargetHealth
EMR	List	ListClusters
	Read	ListInstances
Glacier	List	ListTagsForVault ListVaults
	Read	GetVaultNotifications
IAM	Read	GetUser GetAccountAuthorizationDetails
IoT	List	ListThings ListTagsForResource
	Read	DescribeThing
Key Management Service (KMS)	List	ListKeys ListAliases
	Read	DescribeKey ListResourceTags
Lambda	List	ListFunctions ListAliases ListEventSourceMappings
	Read	ListTags
Lightsail	List	GetBundles GetRegions
	Read	GetInstanceMetricData GetInstances
OpsWorks	List	DescribeInstances DescribeStacks
RDS	List	DescribeDBClusters DescribeDBInstances DescribeDBSubnetGroups
	Read	ListTagsForResource

Service	Actions	
Redshift	List	DescribeClusters
	Read	DescribeLoggingStatus
Route 53	List	GetHostedZone ListHealthChecks ListHostedZones ListResourceRecordSets
S3	List	ListAllMyBuckets ListBucket
	Read	GetBucketLocation GetBucketLogging GetBucketTagging GetBucketWebsite GetObject (Restrict access to specific resources of Elastic Beanstalk. For instance, Bucket name: elasticbeanstalk-*, Any Object name.)
Shield	List	ListAttacks ListProtections
	Read	DescribeEmergencyContactSettings GetSubscriptionState
Simple Email Service (SES)	List	ListIdentities
Simple Notification Service (SES)	List	ListTopics ListSubscriptions
SQS	List	ListQueues
	Read	GetQueueAttributes
Storage Gateway	List	ListGateways ListVolumes
STS	Read	GetCallerIdentity
WAF	List	ListWebACLs
	Read	GetRateBasedRule GetRule GetRuleGroup GetWebACL
WAF Regional	List	ListResourcesForWebACL ListWebACLs
	Read	GetRateBasedRule GetRule GetRuleGroup GetWebACL

To create the Minimum Permission policy:

1. Go to the AWS console and select **IAM > Policies > Create Policy**. Select **JSON** and cut and paste the following JSON document:

```
{
   "Statement": [
       {
               "Action": [
                       "apigateway:GET",
                       "autoscaling:DescribeAutoScalingGroups",
                       "autoscaling:DescribeAutoScalingInstances",
                       "autoscaling:DescribeLaunchConfigurations",
                       "cloudfront:GetDistribution",
                       "cloudfront:ListDistributions",
                       "cloudfront:ListInvalidations",
                       "cloudfront:ListStreamingDistributions",
                       "cloudtrail:DescribeTrails",
                       "cloudtrail:GetTrailStatus",
                       "cloudwatch:DescribeAlarmHistory",
                       "cloudwatch:DescribeAlarms",
                       "cloudwatch:GetMetricData",
                       "cloudwatch:GetMetricStatistics",
                       "cloudwatch:ListMetrics",
                       "config:GetDiscoveredResourceCounts",
                       "directconnect:DescribeConnections",
                       "directconnect:DescribeTags",
                       "directconnect:DescribeVirtualInterfaces",
                       "dynamodb:DescribeTable",
                       "dynamodb:ListTables",
                       "ec2:DescribeAvailabilityZones",
                       "ec2:DescribeImages",
                       "ec2:DescribeInstances",
                       "ec2:DescribeNatGateways",
                       "ec2:DescribeRegions",
                       "ec2:DescribeRouteTables",
                       "ec2:DescribeSecurityGroups",
                       "ec2:DescribeSnapshots",
                       "ec2:DescribeSubnets",
                       "ec2:DescribeTransitGatewayAttachments",
                       "ec2:DescribeTransitGatewayRouteTables",
                       "ec2:DescribeTransitGateways",
                       "ec2:DescribeVolumes",
                       "ec2:DescribeVpcPeeringConnections",
                       "ec2:DescribeVpcs",
                       "ec2:DescribeVpnConnections",
                       "ec2:DescribeVpnGateways",
                       "ecs:DescribeClusters",
                       "ecs:DescribeContainerInstances",
                       "ecs:DescribeServices",
                       "ecs:DescribeTaskDefinition",
                       "ecs:DescribeTasks",
                       "ecs:ListClusters",
                       "ecs:ListContainerInstances",
                       "ecs:ListServices",
                       "ecs:ListTasks",
                       "eks:DescribeCluster",
                       "eks:ListClusters",
```

```
"elasticache:DescribeCacheClusters",
"elasticbeanstalk:DescribeConfigurationSettings",
"elasticbeanstalk:DescribeEnvironmentResources",
"elasticbeanstalk:DescribeEnvironments",
"elasticbeanstalk:DescribeEnvironmentHealth",
"elasticbeanstalk:DescribeInstancesHealth",
"elasticfilesystem:DescribeFileSystems",
"elasticloadbalancing:DescribeListeners",
"elasticloadbalancing:DescribeLoadBalancers",
"elasticloadbalancing:DescribeTags",
"elasticloadbalancing:DescribeTargetGroups",
"elasticloadbalancing:DescribeTargetHealth",
"elasticmapreduce:ListClusters",
"elasticmapreduce:ListInstances",
"glacier:GetVaultNotifications",
"glacier:ListTagsForVault",
"glacier:ListVaults",
"iam:GetAccountAuthorizationDetails",
"iam:GetUser",
"iot:DescribeThing",
"iot:ListTagsForResource",
"iot:ListThings",
"kms:DescribeKey",
"kms:ListAliases",
"kms:ListKeys",
"kms:ListResourceTags",
"lambda:GetAccountSettings",
"lambda:ListAliases",
"lambda:ListEventSourceMappings",
"lambda:ListFunctions",
"lambda:ListTags",
"lightsail:GetBundles",
"lightsail:GetInstanceMetricData",
"lightsail:GetInstances",
"lightsail:GetRegions",
"opsworks:DescribeInstances",
"opsworks:DescribeStacks",
"rds:DescribeDBClusters",
"rds:DescribeDBInstances",
"rds:DescribeDBSubnetGroups",
"rds:ListTagsForResource",
"redshift:DescribeClusters",
"redshift:DescribeLoggingStatus",
"route53:GetHostedZone",
"route53:ListHealthChecks",
"route53:ListHostedZones",
"route53:ListResourceRecordSets",
"s3:GetBucketLocation",
"s3:GetBucketLogging",
"s3:GetBucketTagging",
"s3:GetBucketWebsite",
"s3:GetObject",
"s3:ListAllMyBuckets",
"s3:ListBucket",
"ses:ListIdentities",
"shield:DescribeEmergencyContactSettings",
```

```
                        "shield:GetSubscriptionState",
                        "shield:ListAttacks",
                        "shield:ListProtections",
                        "sns:ListSubscriptions",
                        "sns:ListTopics",
                        "sqs:GetQueueAttributes",
                        "sqs:ListQueues",
                        "ssm:GetParameters",
                        "storagegateway:ListGateways",
                        "storagegateway:ListVolumes",
                        "sts:GetCallerIdentity",
                        "tag:Get*",
                        "waf-regional:GetRateBasedRule",
                        "waf-regional:GetRule",
                        "waf-regional:GetRuleGroup",
                        "waf-regional:GetWebACL",
                        "waf-regional:ListResourcesForWebACL",
                        "waf-regional:ListWebACLs",
                        "waf:GetRateBasedRule",
                        "waf:GetRule",
                        "waf:GetRuleGroup",
                        "waf:GetWebACL",
                        "waf:ListWebACLs"
                ],
                "Effect": "Allow",
                "Resource": "*",
                "Sid": "VisualEditor0"
            }
    ],
    "Version": "2012-10-17"
}
```

2. Click **[Next: Tags]**. If applicable, enter your Tags.
3. Click **[Next: Review]**. Name the policy "SL1MinimumPermissions" and click **[Create Policy]**.

This policy needs to be available in each account that is to be monitored and will be referenced in the following sections.

Chapter 5

AWS Discovery

Overview

The following sections describe the different methods of discovery that can be used with the *Amazon Web Services* PowerPack:

Discovering Amazon Web Services

SL1 currently supports the following methods to discover your AWS accounts:

- *Manual Discovery*. Requires the creation of a virtual device, manual alignment of Dynamic Applications, and an IAM key. This process needs to be repeated for each AWS account.
- *Automated Discovery using Assume Role with single IAM key from Master Account*. Provides an automated mechanism to discover all your AWS accounts within an organization using a single IAM key. This is the recommended method of discovery when your Data Collector is not an EC2 instance.
- *Automated Discovery when the Data Collector runs as an EC2 instance*. Provides a fully automated mechanism to discover all your AWS accounts when your Data Collectors are running as EC2 instances. SL1 does not need any AWS credentials in this case. This is the recommended approach when your Data Collectors are EC2 instances.
- *AWS Guided Discovery*. Uses a guided workflow in SL1. This method is recommended when you want to use a separate IAM key for each AWS account. The guided workflow provides a more user-friendly version of the manual process.

Before determining your method of discovery, it is recommended to define the minimum permissions policy in AWS. This policy defines the minimum permissions needed to monitor all AWS services and is needed regardless of which of the above methods is used.

You can discover a maximum of 10 accounts with the following requirements on the Data Collector:

- 8 cores
- 32 GB of RAM
- 100 GB of HDD

Manual Discovery

Manual discovery is used to discover a single AWS account at a time and requires an IAM key for the account.

NOTE: Using one of the Assume Role methods of discovery is recommended.

The process consists of the following steps:

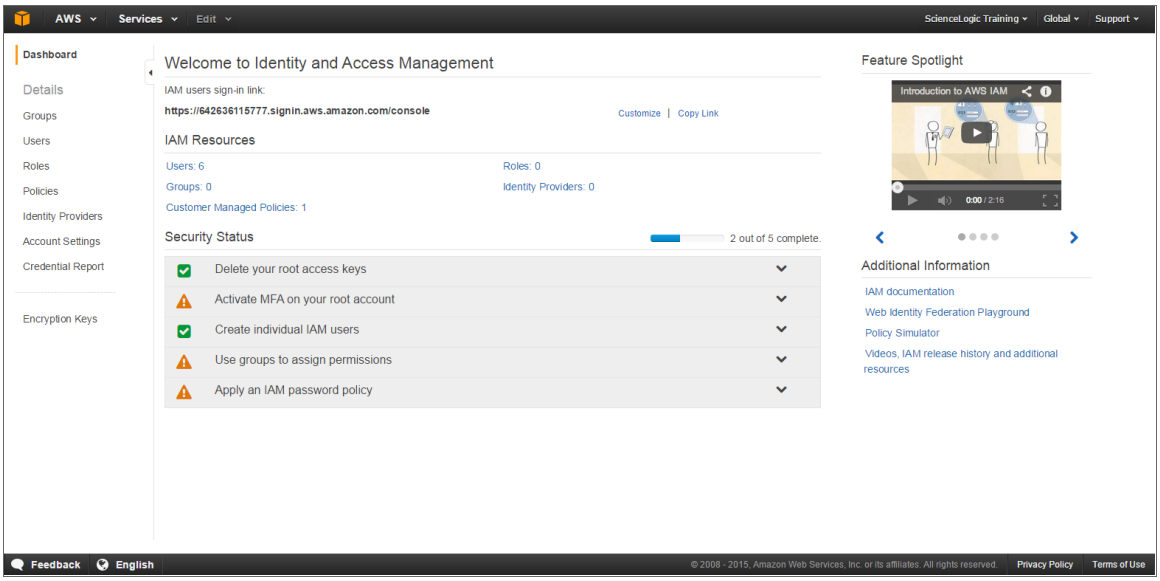
1. *Configure a user in the AWS Account*
2. *Configure the SL1 Credential*
3. *Create a Virtual Device*
4. *Align the Discovery Dynamic Application*

Configuring a User in AWS

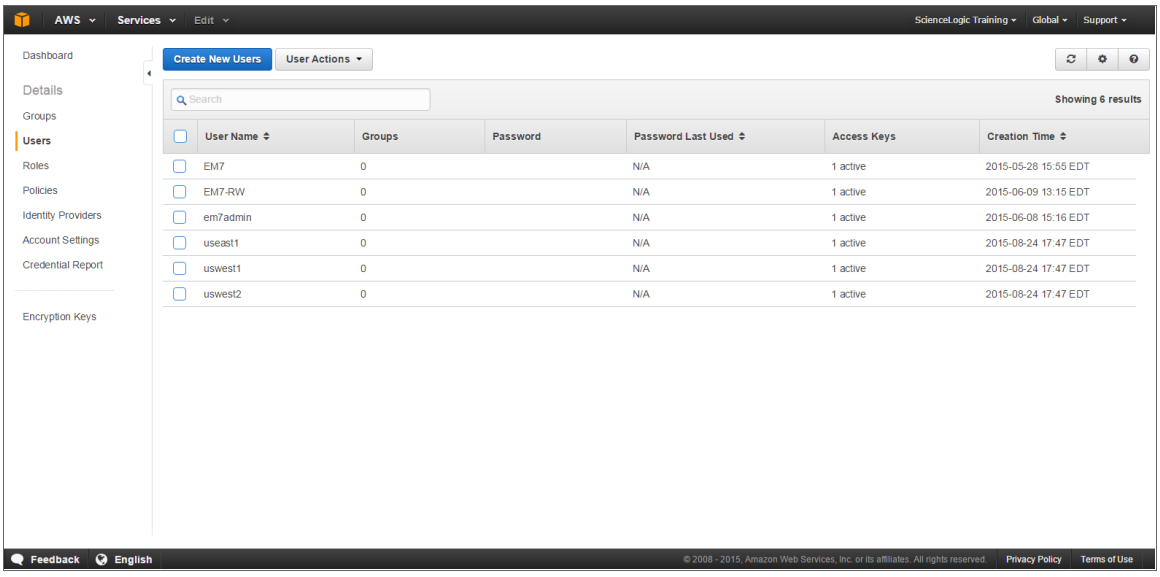
To create a read-only user account in AWS, perform the following steps:

1. Open a browser session and go to aws.amazon.com.
2. Click **[My Account]** and then select *AWS Management Console*. If you are not currently logged in to the AWS site, you will be prompted to log in:

3. In the **AWS Management Console**, under the **Security & Identity** heading, click **[Identity & Access Management]**.
4. After logging in, the **Identity & Access Management Dashboard** page appears:

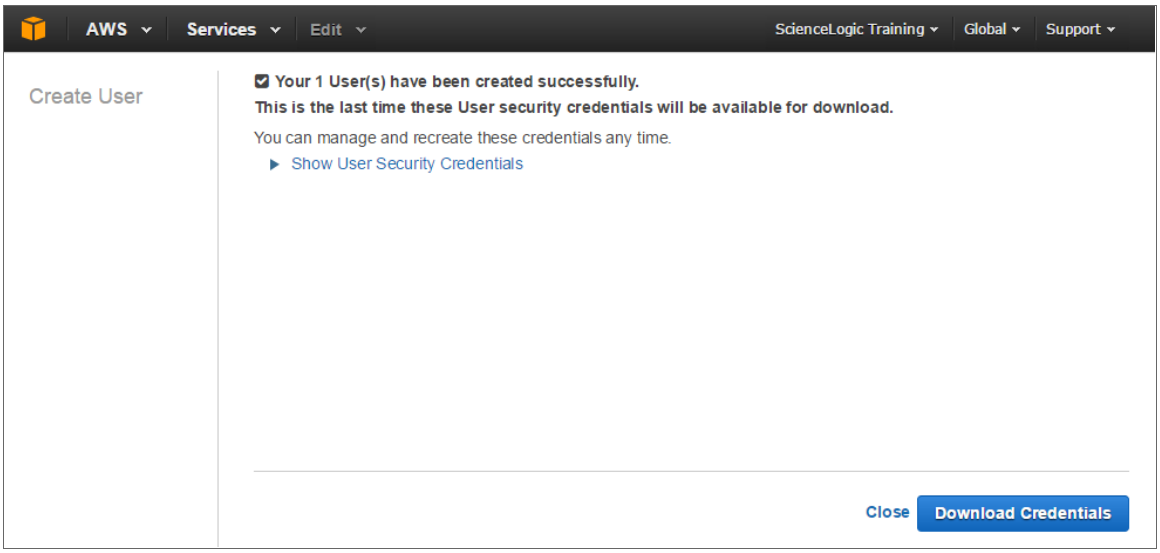


5. To create a user account for SL1, click **[Users]** on the Dashboard menu.

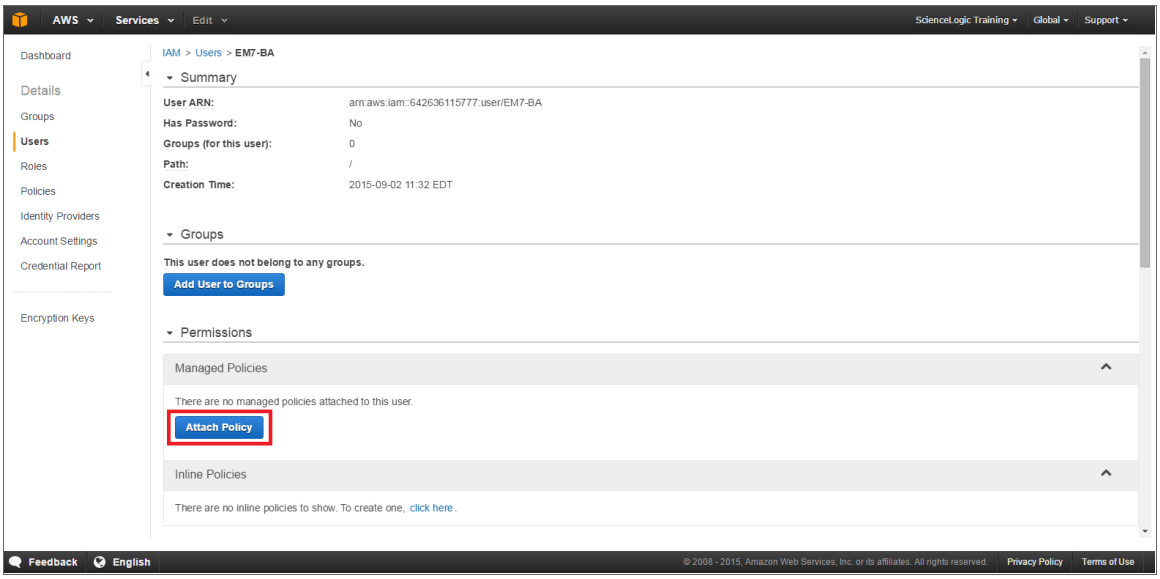


6. Click the **[Create New Users]** button.
7. Enter a username for the new user, e.g. "SL1", and make sure the ***Generate an access key for each user*** checkbox is selected.

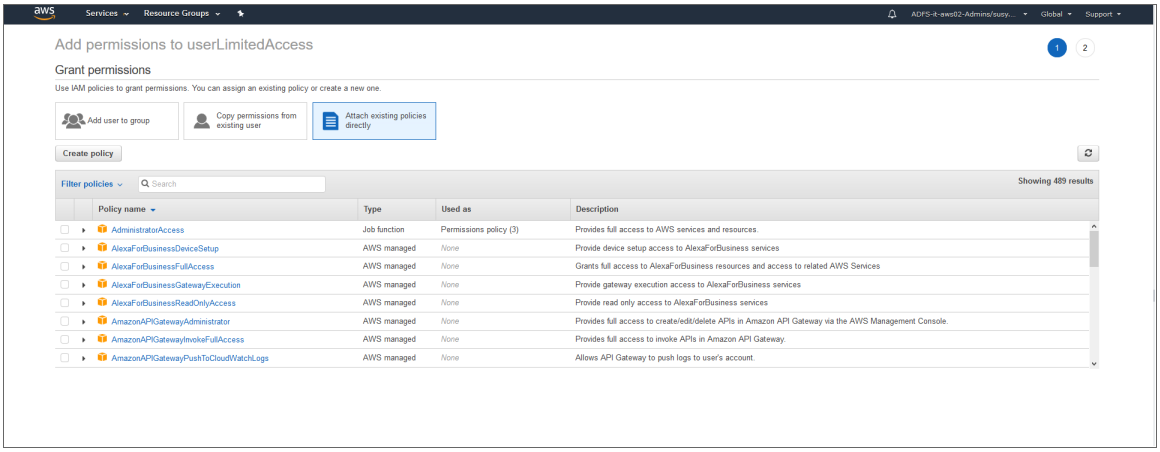
8. Click the **[Create]** button to generate your user account. The **Create User** page appears:



9. Click the **[Download Credentials]** button to save your Access Key ID and Secret Key as a CSV (comma-separated value) text file, and then click **[Close]**.
10. After creating a user, you must assign it a set of permissions policies. Click the username of the user account you created. The user's account information appears:



11. Under the **Permissions** heading, click the **[Attach existing policies directly]** button. The **Add permissions** page appears:



12. Select the checkbox for your policy based on the definition of the minimum required permissions described in the *Minimum Permissions for Dynamic Applications* section.
13. Click the **[Attach Policy]** button.

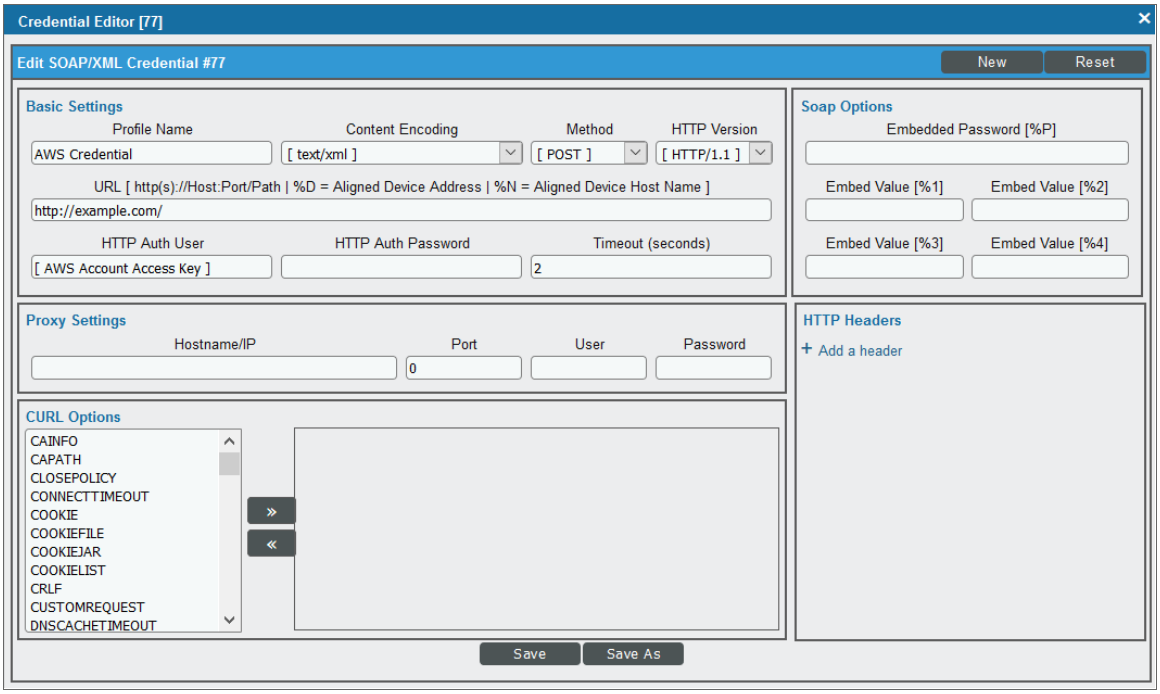
Creating the SOAP/XML Credential for AWS

To discover AWS using the manual discovery method, you must first define an AWS credential in SL1.

To define an AWS credential:

1. Go to the **Credential Management** page (System > Manage > Credentials).

2. Locate the **AWS Manual Discovery** sample credential and click its wrench icon ( ). The **Credential Editor** modal page appears:



3. Enter values in the following fields:

Basic Settings

- ***Profile Name***. Type a new name for your AWS credential.
- ***URL***. Enter a valid URL. This field is not used for this discovery method but must be populated with a valid URL for discovery to complete.
- ***HTTP Auth User***. Type your ***Access Key ID***.
- ***HTTP Auth Password***. Type your ***Secret Access Key***. The characters appear as asterisks to protect your password privacy.

Proxy Settings

NOTE: The ***Proxy Settings*** fields are required only if you are discovering AWS services through a proxy server. Otherwise, leave these fields blank.

- ***Hostname/IP***. Type the host name or IP address of the proxy server.
- ***Port***. Type the port on the proxy server to which you will connect.
- ***User***. Type the username used to access the proxy server.
- ***Password***. Type the password used to access the proxy server.

CAUTION: If you are creating a credential from the **AWS Credential - Proxy** example and the proxy server does not require a username and password, then the ***User*** and ***Password*** fields must both be blank. In that scenario, if you leave the "<Proxy_User>" text in the ***User*** field, SL1 cannot properly discover your AWS services.

SOAP Options

- ***Embed Value [%1]***. Do one of the following:
  - To monitor a GovCloud account, type "us-gov-west-1" or "us-gov-east-1".
  - To monitor the Beijing region, type "cn-north-1".
  - To monitor the Ningxia region, type "cn-northwest-1".

  Otherwise, leave this field blank.

NOTE: If you are monitoring both the Beijing and Ningxia regions, you must create a unique credential for each region.

- ***Embed Value [%2]***:
  - If you are using the AWS Config service and want to discover only regions that have that service enabled, type "[AUTO]" in this field. After discovery, only regions that have AWS Config enabled will be displayed in the dynamic component map tree. Global resources will also be discovered.
  - If you are using not using the AWS Config service, type "[FILTER]" in this field so it will discover only regions that are reporting CloudWatch metrics. This will reduce the number of regions being monitored and the load on the Data Collector.

CAUTION: If you are performing discovery using [AUTO] or [FILTER] in the ***Embed Value [%2]*** field, the status of regions that don't meet these requirements will change to *Unavailable* and vanish if enabled.

NOTE: If you are performing discovery based on the AWS Config service and do not have any regions with the AWS Config service enabled, the *Amazon Web Services* PowerPack will discover all regions that have resources.

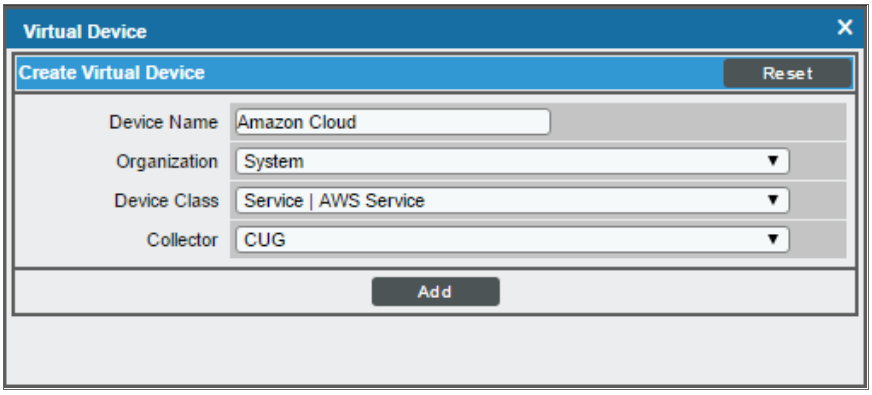
4. Click the **[Save As]** button, and then click **[OK]**.

Creating an AWS Virtual Device for Discovery in the SL1 Classic User Interface

Because the Amazon Web Service does not have a specific IP address, you cannot discover an AWS device using discovery. Instead, you must create a ***virtual device*** that represents the Amazon Web Service. A virtual device is a user-defined container that represents a device or service that cannot be discovered by SL1. You can use the virtual device to store information gathered by policies or Dynamic Applications.

To create a virtual device that represents your Amazon service:

1. Go to the **Device Manager** page (Devices > Device Manager or Registry > Devices > Device Manager in the SL1 classic user interface).
2. Click the **[Actions]** button, then select *Create Virtual Device*. The **Virtual Device** modal page appears:



3. Enter values in the following fields:
    - ***Device Name***. Enter a name for the device. For example, you could enter "Amazon Cloud" in this field.
    - ***Organization***. Select the organization for this device. The organization the device is associated with limits the users that will be able to view and edit the device.
    - ***Device Class***. Select *Service | AWS Service*.
    - ***Collector***. Select the collector group that will monitor the device.
4. Click the **[Add]** button to create the virtual device.

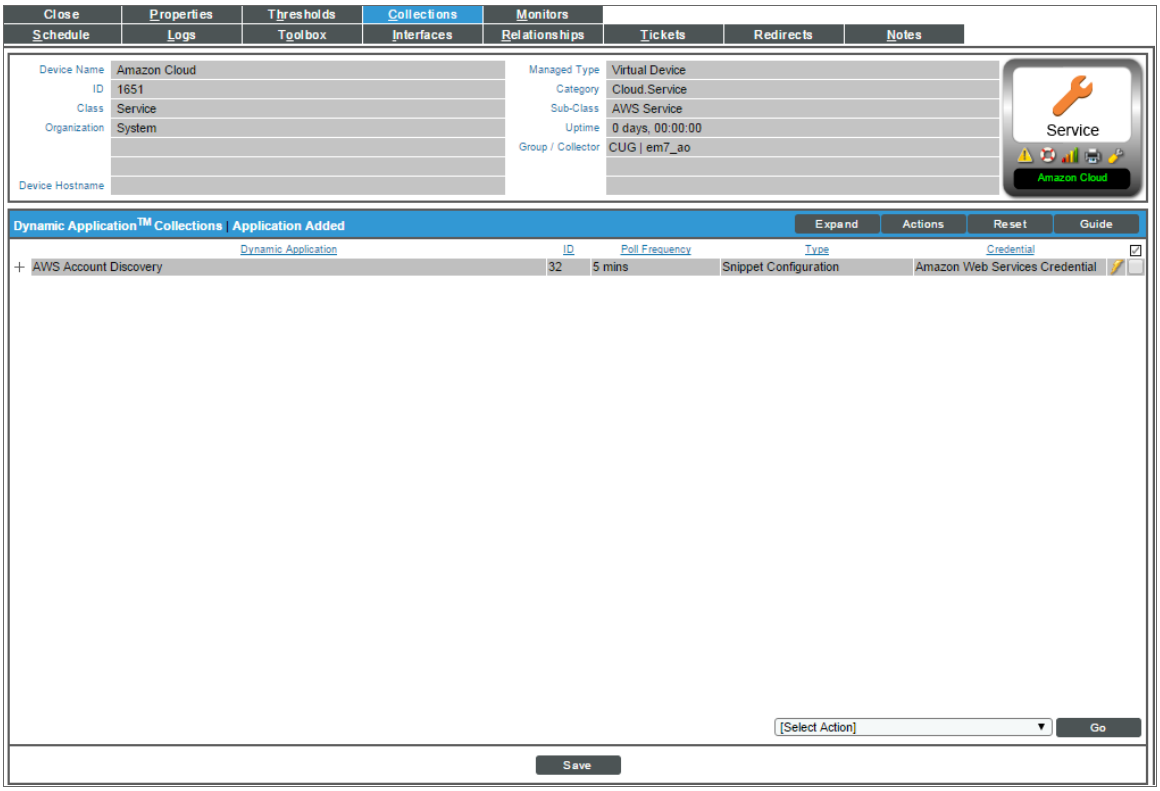
Aligning the Discovery Dynamic Application in the SL1 Classic User Interface

To discover your AWS account, you must manually align the "AWS: Account Discovery" Dynamic Application with the AWS virtual device. After you do so, the other Dynamic Applications in the *Amazon Web Services* PowerPack will automatically align to discover and monitor all of the components in your AWS account.

TIP: If your AWS account includes API Gateways or Lambda services to be monitored and you want SL1 to put those component devices in a "vanished" state if the platform cannot retrieve data about them for a specified period of time, ScienceLogic recommends setting the ***Component Vanish Timeout Mins.*** field to at least 120 minutes. For more information, see the chapter on "Vanishing and Purging Devices" in the ***Device Management*** manual.

To align the "AWS: Account Discovery" Dynamic Application to your virtual device:

1. Go to the **Device Manager** page (Devices > Device Manager or Registry > Devices > Device Manager in the SL1 classic user interface).
2. Click the wrench icon () for your virtual device.
3. In the **Device Administration** panel, click the **[Collections]** tab. The **Dynamic Application Collections** page appears:
4. Click the **[Actions]** button, and then select *Add Dynamic Application* from the menu.
5. In the **Dynamic Application Alignment** modal page, select *AWS: Account Discovery* in the ***Dynamic Applications*** field.
6. In the ***Credentials*** field, select the *credential you created for your AWS service*.
7. Click the **[Save]** button to align the Dynamic Application.



Automated Discovery Using AssumeRole with a Single IAM Key from the AWS Master Account

Automated discovery using AssumeRole with an IAM key is the recommended approach to monitor your AWS accounts when your Data Collectors are ***not*** acting as EC2 instances. In this method of discovery, your organization will be discovered first and then the accounts within the organization will be created automatically.

This method of discovery has the following benefits:

- Only a single IAM key needs to be managed on SL1, instead of an IAM key for every AWS account.
- The IAM key is only used to get the information about the organization, and all the actual monitoring is done via temporary tokens, which is the recommended approach by AWS.

This method can also be used in the following scenarios:

- When a proxy server is between the Data Collector and the AWS cloud
- When Ping is not available
- In the Government cloud

NOTE: All examples shown are for commercial AWS accounts. When AWS Gov is being monitored, the JSON data that refers to ARN will need to be modified from "aws" to "aws-us-gov". For example: `Resource": "arn:aws:iam::<account number>:role/Sciencelogic-Monitor` would need to be `Resource": "arn:aws:iam-us-gov::<account number>:role/Sciencelogic-Monitor`

To use this method of discovery, perform the following steps:

1. *Configure a user in the master billing account*
2. *Create a role in each account*
3. *Configuring the SL1 credential*
4. *Create and run the discovery session*

NOTE: If Ping is blocked, then you must follow the steps in the *Manually Create the Organization and Align the Dynamic Applications* section.

Configure a User in the Master Billing Account

The first step in this discovery method is to create a policy that defines the permissions needed by SL1. To do this, copy the policy below into an editor:

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "VisualEditor0",
         "Effect": "Allow",
         "Action": [
            "organizations:ListAccounts",
            "organizations:DescribeOrganization",
            "organizations:DescribeAccount"
         ],
         "Resource": "*"
      },
      {
         "Sid": "VisualEditor1",
         "Effect": "Allow",
         "Action": "sts:AssumeRole",
         "Resource": "arn:aws:iam::<account number>:role/Sciencelogic-Monitor"
      }
   ]
}
```

For each account that needs to be monitored, duplicate the `"Resource": "arn:aws:iam::<Account Number>:role/Sciencelogic-Monitor"` line and set the `<Account Number>` to the correct account number.

After editing the policy, perform the following steps in the AWS console:

1. Go to **IAM > Policies > Create Policy**. Select the **JSON** tab and copy the edited JSON text into the AWS console.
2. Click **Next: Tags** and then click **Next: Review**.
3. Type a name for the policy (for example, "SL1MasterBillingPermissions") and then select **[Create Policy]**.
4. To create a user in the master billing account, go to **IAM > Users > Add User**.
5. Type the user's name and select the option for **Programmatic Access**. Click **[Next: Permissions]**.
6. Select ***Attach existing policies directly*** and select the checkbox for the policy you created.
7. Select **Next: Tags > Next: Review > Create User**.

NOTE: The Access Key and Secret Key need to be saved as these will be needed when configuring the SL1 credential.

Create a Role in Each Account

In every AWS account that is to be monitored, a role with the ***same name*** needs to be created. The default name is "ScienceLogic-Monitor". To create the role, perform the following steps for each account that is to be monitored:

1. In the AWS console, go to **IAM > Roles** and select **Create Role**.
2. Select **Another AWS Account** and enter the account ID of the Master Billing Account. Select **Next: Permissions**.
3. Select the policy that was created in the *Minimum Permissions Needed to Monitor Your AWS Accounts* section.
4. Select **Next: Tags** and then **Next: Review**.
5. Enter "ScienceLogic-Monitor" in the ***Role name*** field and then select **[Create role]**.
6. Repeat these steps for each AWS account that you want to monitor.

Next you will need to edit the trust relationship of the role to restrict the principle to the user you created. To do this:

1. In the AWS console, go to **IAM > Roles** and select the "ScienceLogic-Monitor" role.
2. Select the **Trust Relationships** tab and click **[Edit trust relationship]**.
3. Edit the JSON to look like the following:

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Principal": [
            "AWS": "arn:aws:iam::<Master Billing Account>:user/<Master Billing Account
User>"
      },

      {
         "Action": "sts:AssumeRole",
         "Condition": {}
      }
   ]
}
```

NOTE: The ARN above is the ARN of the user that was created in the previous steps.

4. Once you have updated the policy, click **[Update Trust Policy]**.

Configure the SL1 Credential

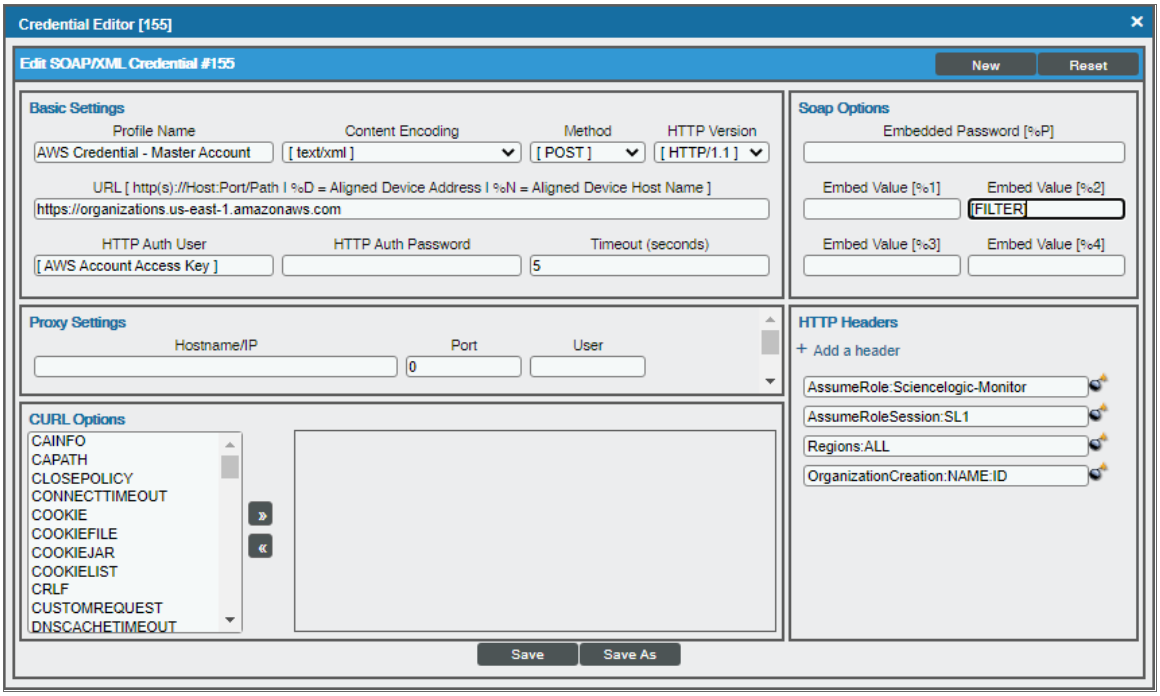
You can use your master organization account to automatically discover all AWS accounts, instead of having to enter a key for each account. This process will also create a separate DCM tree for each account.

NOTE: Ensure that you use the "AWS Credential - Master Account" credential, as this credential is valid for AssumeRole and has the correct headers for AssumeRole discovery. Do not use the classic "AWS Credential" to discover an AssumeRole pingable device, as it will not work.

NOTE: Discovery of China accounts does not support alignment using AssumeRole. For those accounts customers must continue to use manual alignment of Dynamic Applications.

To define the credential:

1. Go to the **Credential Management** page (System > Manage > Credentials).
2. Locate the **AWS Credential - Master Account** sample credential that you need and click its wrench icon ( ). The **Credential Editor** modal page appears:



3. Enter values in the following fields:

Basic Settings

- ***Profile Name***. Type a new name for your AWS credential.
- ***URL***. Type `https://organizations.us-east-1.amazonaws.com` in the field. If your administrator has configured a different region, you can change it or use the default region. To discover Gov accounts using AssumeRole, type `https://organizations.us-gov-west-1.amazonaws.com`
- ***HTTP Auth User***. Type the AWS access key ID of the user you created in the master account.
- ***HTTP Auth Password***. Type the AWS secret access key of the user created in the master account.

SOAP Options

- ***Embed Value [%2]***:
  - If you are using the AWS Config service and want to discover only regions that have that service enabled, type "[AUTO]" in this field. After discovery, only regions that have AWS Config enabled will be displayed in the dynamic component map tree. Global resources will also be discovered.
  - If you are using not using the AWS Config service, type "[FILTER]" in this field so it will discover only regions that are reporting CloudWatch metrics. This will reduce the number of regions being monitored and the load on the Data Collector.

HTTP Headers

- Click **+ *Add a header*** to add a header field. You can enter the following options:
  - *AssumeRole*. Type the AWS Role you created in each account. The default name is "ScienceLogic-Monitor".
  - *AssumeRoleSession*. Optional. The default value is "AssumeRoleSession:SL1".
  - *Regions*. The regions entered in this field will be discovered. For example, entering "Regions:ap-southeast-2, us-east-2" will discover two regions. If left blank, all regions will be discovered. The default value is "Regions:ALL".
  - *OrganizationCreation:NAME:ID*. Autocreates an SL1 organization for accounts using AssumeRole. You can enter one of the following options:
    - **OrganizationCreation:NAME**. The name of the organization will contain the name of the user.
    - **OrganizationCreation:ID**. The name of the organization will contain the ID of the user.
    - **OrganizationCreation:ID:NAME**. The name of the organization will contain both the ID and name of the user, in that order.
    - **OrganizationCreation:NAME:ID**. The name of the organization will contain both the name and ID of the user, in that order.

NOTE: The existing organization will be changed by this setting only if it is the default (System) organization. If this header is not included, then **all** the discovered accounts will be placed into the organization selected in the discovery session.

4. Click the **[Save As]** button, and then click **[OK]**.

NOTE: If the "AWS: Account Creation" Dynamic Application is reporting that it is unable to use your AssumeRole, double-check your trust relationships on your configured roles.

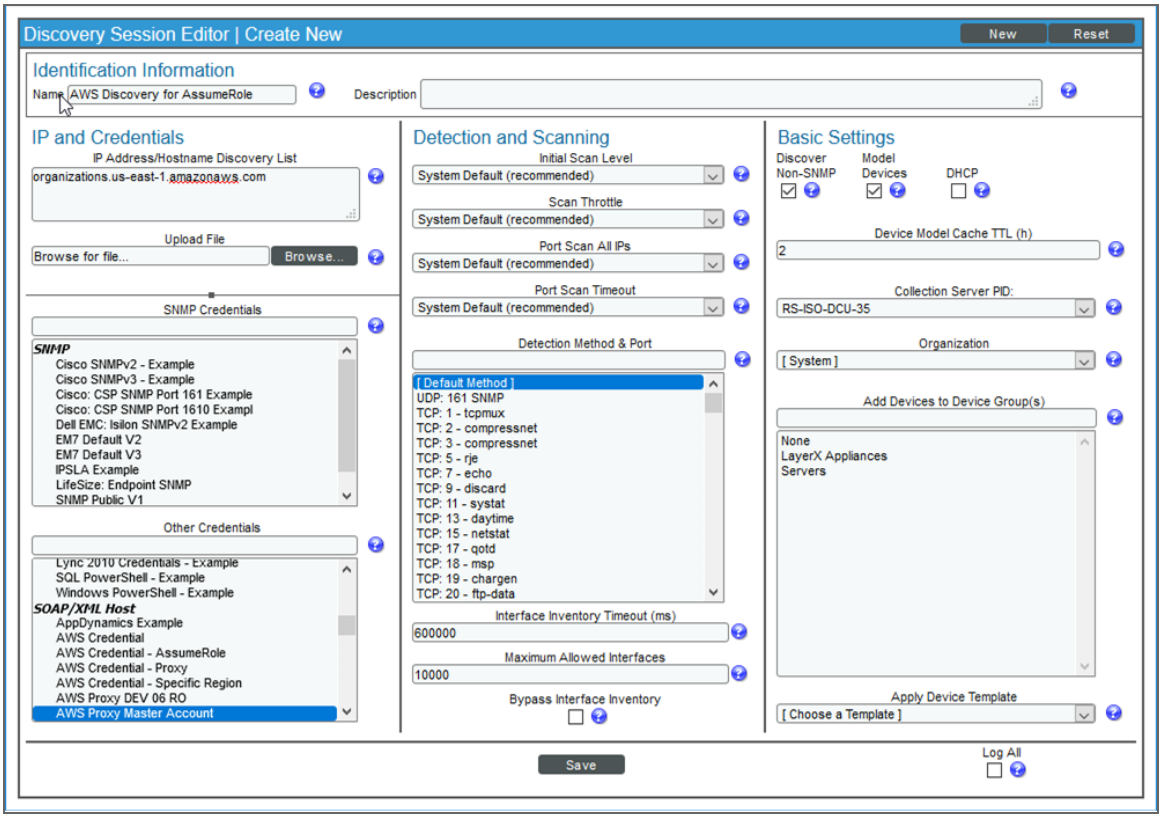
Create and Run the Discovery Session

To discover AWS Accounts in an AWS Organization using AssumeRole, perform the following steps:

NOTE: If Ping is not supported between the Data Collector and AWS, you can skip this section and go to the *Manually Create the Organization and Align Dynamic Applications* section.

1. Go to the **Discovery Control Panel** page (System > Manage > Classic Discovery).

2. Click the **[Create]** button. The **Discovery Session Editor** page appears:



3. Supply values in the following fields:

   - ***IP Address Discovery List***. Type the URL of your AWS master billing account.
   - ***Other Credentials***. Select the credential you created.
   - ***Discover Non-SNMP***. Select this checkbox.
   - ***Model Devices***. Select this checkbox.

4. Optionally, supply values in the other fields in this page. For a description of the fields in this page, see the ***Discovery & Credentials*** manual.
5. Click the **[Save]** button.
6. The **Discovery Control Panel** page will refresh. Click the lightning bolt icon ( ) for the discovery session you just created.
7. In the pop-up window that appears, click the **[OK]** button. The page displays the progress of the discovery session.

NOTE: If you discontinue monitoring on any devices that are using the Assume Role authentication method, ScienceLogic recommends the best practice of first disabling the devices, deleting the devices from the DCM tree, and then cleaning up any AWS permissions in IAM. This will avoid any unnecessary alerts.

Manually Creating the Organization and Aligning Dynamic Applications

NOTE: The following steps are needed only if ping is **not** supported between the Data Collector and AWS.

To create a virtual device to create the organization:

1. Go to the **Device Manager** page (Devices > Device Manager or Registry > Devices > Device Manager in the SL1 classic user interface).
2. Click the **[Actions]** button, then select *Create Virtual Device*. The **Virtual Device** modal page appears:
3. Enter values in the following fields:
    - ***Device Name***. Enter a name for the device. For example, you could enter "Amazon Organization" in this field.
    - ***Organization***. Select the organization for this device. The organization the device is associated with limits the users that will be able to view and edit the device.
    - ***Device Class***. Select *AWS | Organization*.
    - ***Collector***. Select the collector group that will monitor the device.
4. Click the **[Add]** button to create the virtual device.

Next, you must manually align the "AWS: Account Creation" Dynamic Application with the AWS virtual device. After you do so, the other Dynamic Applications in the *Amazon Web Services* PowerPack will automatically align to discover and monitor all of the components in your AWS account.

To align the "AWS: Account Creation" Dynamic Application to your virtual device:

1. Go to the **Device Manager** page (Devices > Device Manager or Registry > Devices > Device Manager in the SL1 classic user interface).
2. Click the wrench icon () for your virtual device.
3. In the **Device Administration** panel, click the **[Collections]** tab. The **Dynamic Application Collections** page appears:
4. Click the **[Actions]** button, and then select *Add Dynamic Application* from the menu.
5. In the **Dynamic Application Alignment** modal page, select *AWS: Account Creation* in the ***Dynamic***

Applications field.

6. In the ***Credentials*** field, select the credential you created for your AWS service.
7. Click the **[Save]** button to align the Dynamic Application.

Automated Discovery when the Data Collector Runs as an EC2 Instance

This method of discovery is recommended for monitoring your AWS accounts within an organization when your Data Collectors are EC2 instances. In this case, a standard SL1 discovery process is created, and this mechanism will first discover your organization and then create all the accounts within the organization.

This method of discovery has the following benefits:

- No AWS credentials are needed in SL1

NOTE: All examples shown are for commercial AWS accounts. When AWS Gov is being monitored, the JSON data that refers to ARN will need to be modified from "aws" to "aws-us-gov". For example: `Resource": "arn:aws:iam::<account number>:role/Sciencelogic-Monitor` would need to be `Resource": "arn:aws:iam-us-gov::<account number>:role/Sciencelogic-Monitor`

To use this method of discovery, perform the following steps:

1. *Create an AWS role in the master billing account*
2. *Create an AWS role in account that the collector is in*
3. *Create an AWS role in each account that is to be monitored*
4. *Create an SL1 credential*
5. *Create and run the discovery session*

Create a Role in the Master Billing Account

The role you will create in the master billing account is assumed from the account that the EC2 instance is in. This role will enable SL1 to temporarily log in to the master billing account and discover other accounts.

Before creating the role, you must first create a policy that defines the permissions needed by SL1. To do this, copy the policy from below into an editor:

```
{ "Version": "2012-10-17",
   "Statement":
      {"Sid": "VisualEditor0",
         "Effect": "Allow",
         "Action": [
            "organizations:ListAccounts",
            "organizations:DescribeOrganization",
```

```
            "organizations:DescribeAccount"
          ]
          "Resource": "*"
      },
}
```

Next, perform the following steps:

1. Log in to the Master Billing Account via the AWS console and select **IAM > Policies > Create Policy**.
2. Select the **JSON** tab and paste the JSON text you copied above into the AWS console.
3. Click **Next: Tags** and then click **Next: Review**.
4. Type a name for the policy (for example, "SL1MasterBillingPermissions") in the ***Name*** field and then click **Create Policy**.

To create the role:

1. Go to **IAM > Roles > Create Role**.
2. Under ***Select type of trusted entity***, select **Another AWS account**.
3. Type the account number of the account that contains the EC2 instance running on the collector in the ***Account ID*** field, and then click **Next: Permissions**.
4. Select the checkbox for the policy you created above.
5. Click **Next: Tags** and then click **Next: Review**.
6. Type the role name from the example above (SL1MasterAccountRole) in the ***Role name*** field, then click **Create role**.

The trust policy is set up by the console automatically as follows:

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Principal": {
          "AWS": "arn:aws:iam::581618222958:root"
         },
         "Action": "sts:AssumeRole",
         "Condition":{}

        }
      ]
    }
```

7. In the console, edit the trust relationship and replace `:root` with `:role/ec2-collector`.

NOTE: "ec2-collector" is the name of the role that will be created in the account that the EC2 collector is in. This policy allows only the "ec2-colletor" role to assume this role in the master billing account. If you use another name for the role, then this trust relationship must use that name instead of "ec2-collector".

Create an AWS Role in the Account your Data Collector is In

The role you create in the account your Data Collector is in will be assigned to the EC2 instances that house those Data Collectors. This role enables the SL1 Data Collector to assume a role in the master billing account, which is then used to discover the organization and retrieve the accounts associated with that organization. Once the accounts have been discovered, this role allows SL1 to assume the monitor role in each of the accounts.

First you will need to create a policy in the accounts that the Data Collectors are in. To create this policy, first cut and paste the following JSON text into an editor:

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "VisualEditor0",
         "Effect": "Allow",
         "Action": "sts:AssumeRole",
         "Resource": [
                    "arn:aws:iam::<master billing account ID>:role/SL1MasterAccountRole",
                    "arn:aws:iam::<monitored account 1>:role/ScienceLogic-Monitor",
                    "arn:aws:iam::<monitored account 2>:role/ScienceLogic-Monitor",
                    "arn:aws:iam::<monitored account 3>:role/ScienceLogic-Monitor"
         ]
      }
   ]
}
```

Replace the "**master billing account**" with your master billing account number.

For each account to be monitored, ensure that there is a line under Resource that matches the account ID. The example above shows three accounts to be monitored.

NOTE: If the master billing account is to be monitored, it will also need a line in the Resource list.

If you did not use the example "SL1MasterAccountRole" name, replace it with the name of your role.

Next, perform the following steps:

1. Log in to the AWS console and select **IAM > Policies > Create Policy**.
2. Select the **JSON** tab and copy the JSON text you edited above into the AWS console.
3. Click **Next: Tags** and then click **Next: Review**.
4. Type a name for the policy (for example, "EC2CollectorPolicy") in the ***Name*** field and then click **Create Policy**.

To create the role:

NOTE: If you already have a role assigned to the Data Collector that houses the EC2 instance, then you can add the policy you just created to that existing role. Otherwise, follow the steps below to create the role.

1. Go to **IAM > Roles > Create Role**.
2. Under ***Select type of trusted entity***, select **AWS service**.
3. Under ***Choose a use case***, select **EC2**.
4. Click **Next: Permissions** and select the policy you created above.
5. Click **Next: Tags** and then click **Next: Review**.
6. Type the name from our example (ec2-collector) in the ***Role name*** field, then click **Create role**.

Next, you need to assign this instance profile to the EC2 instances that are Data Collectors. To do this:

1. Go to the AWS console and click **EC2 > Instances**.
2. Select the checkbox for each instance that is a Data Collector.
3. Click **Actions > Security > Modify IAM Role**.
4. In the drop-down field, select the role that you just created and then click **[Save]**.

Create a Role in Each Account

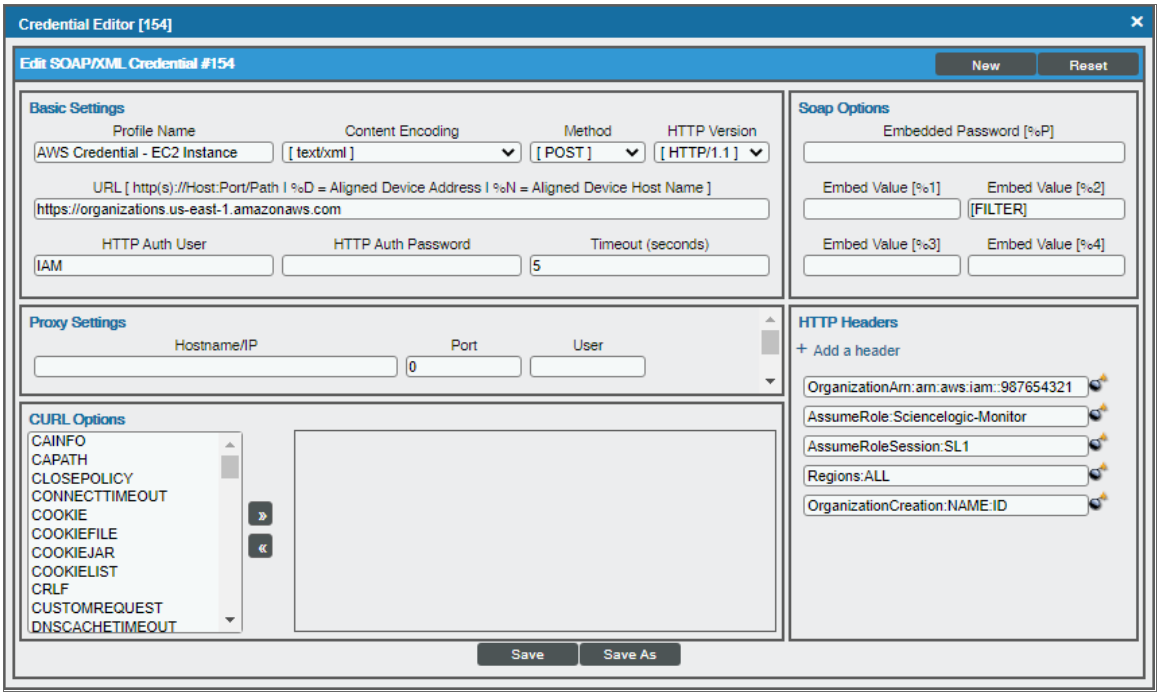
In every account that is to be monitored, a role with the ***same name*** needs to be created. The default name is ScienceLogic-Monitor. The following steps must be performed for each account that is to be monitored:

1. In the AWS console for the account and go to **IAM > Roles > Create Role**.
2. Under ***Select type of trusted entity***, select **Another AWS account**.
3. Type the account number that houses the EC2 collectors in the ***Account ID*** field, and then click **Next: Permissions**.
4. Select the checkbox for the policy you created in the *Minimum Permissions Needed to Monitor Your AWS Accounts* section (called "SL1MinimumPermissions").
5. Click **Next: Tags** and then click **Next: Review**.
6. Type ScienceLogic-Monitor in the ***Role name*** field, then click **Create role**.
7. Click on the role that was just created and select the **Trust Relationships** tab.
8. Click the **[Edit trust relationship]** button.
9. In the **Policy Document** editor, change the Principle from `"AWS": "arn:aws:iam::<ec2 collector account>:root` to `"AWS": "arn:aws:iam::<collector account>:role/ec2-collector"` (where `ec2-collector` is the name of the role created on the account housing the EC2 collector). Then click the **[Update Trust Policy]** button.
10. Repeat these steps for each account that is to be monitored.

Configuring the Credential to Discover AWS on an EC2 Collector

To define an AWS credential to discover AWS on an EC2 collector:

1. Go to the **Credential Management** page (System > Manage > Credentials).
2. Locate the **AWS Credential - EC2 Instance** sample credential that you need and click its wrench icon ( ). The **Credential Editor** modal page appears:



3. Enter values in the following fields:

Basic Settings

- ***Profile Name***. Type a new name for your AWS credential.
- ***URL***. Type `https://organizations.us-east-1.amazonaws.com` in the field. If your administrator has configured a different region, you can change it or use the default region. **To discover Gov accounts** using AssumeRole, type `https://organizations.us-gov-west-1.amazonaws.com`.
- ***HTTP Auth User***. Leave the default value "IAM" in the field.

SOAP Options

- ***Embed Value [%2]***:

- If you are using the AWS Config service and want to discover only regions that have that service enabled, type "[AUTO]" in this field. After discovery, only regions that have AWS Config enabled will be displayed in the dynamic component map tree. Global resources will also be discovered.
- If you are using not using the AWS Config service, type "[FILTER]" in this field so it will discover only regions that are reporting CloudWatch metrics. This will reduce the number of regions being monitored and the load on the Data Collector.

HTTP Headers

- Click **+ *Add a header*** to add a header field. You can enter the following options:
  - *OrganizationArn*. Defines the ARN for the AssumeRole. This is the ARN of the role created in the master billing account. In the ***example above*** it was called "SL1MasterAccountRole". For example, `OrganizationArn:arn:aws:iam::<Master Billing Account>:role/SL1MasterAccountRole`
  - *AssumeRole*. Type the AWS Role you created in each account. The default name is "ScienceLogic-Monitor".
  - *AssumeRoleSession*. Optional. The default value is "AssumeRoleSession:SL1".
  - *OrganizationCreation:NAME:ID*. Autocreates an SL1 organization for accounts using AssumeRole. You can enter one of the following options:
    - **OrganizationCreation:NAME**. The name of the organization will contain the name of the user.
    - **OrganizationCreation:ID**. The name of the organization will contain the ID of the user.
    - **OrganizationCreation:ID:NAME**. The name of the organization will contain both the ID and name of the user, in that order.
    - **OrganizationCreation:NAME:ID**. The name of the organization will contain both the name and ID of the user, in that order.

NOTE: The existing organization will be changed by this setting only if it is the default (System) organization.

4. Click the **[Save As]** button, then click **[OK]**.

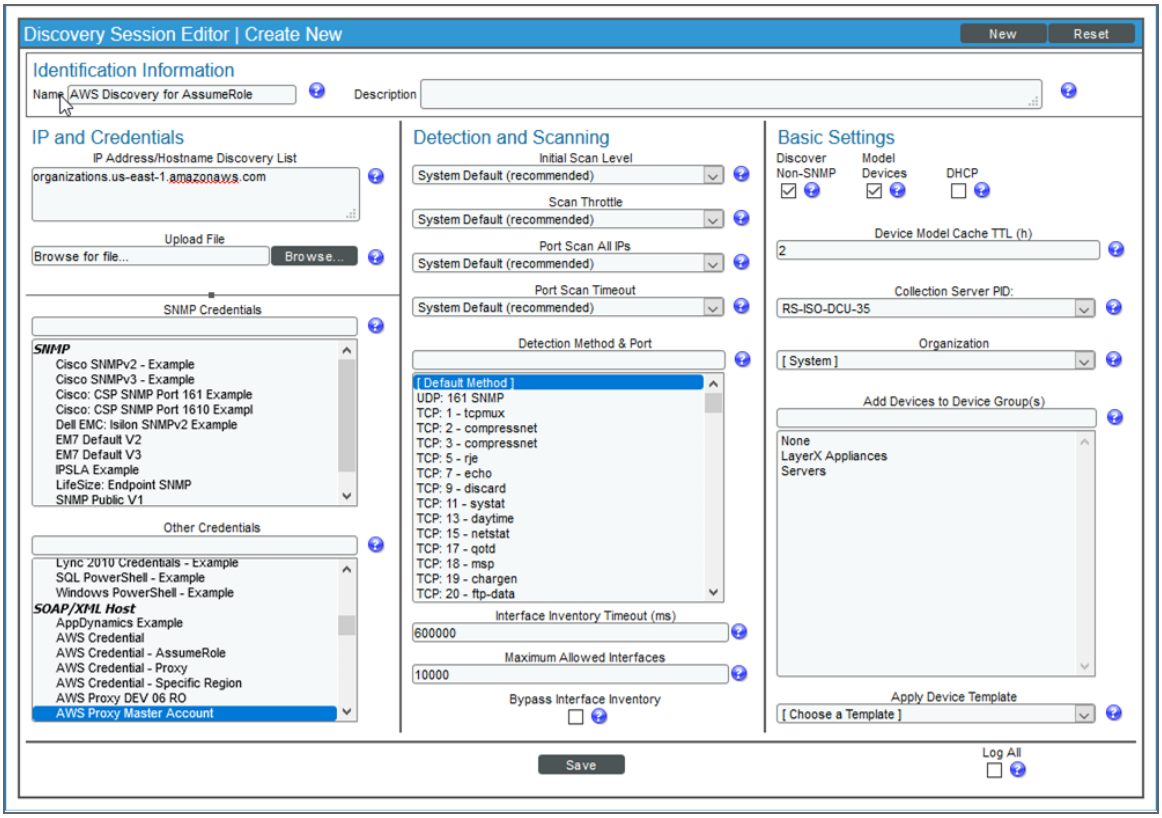
Create and Run the Discovery Session

To discover AWS Accounts in an AWS Organization using AssumeRole, perform the following steps:

NOTE: If you are upgrading the PowerPack and had previously discovered accounts within an organization separately and now want to use a different discovery method, you must first disable the "AWS: Account Discovery" Dynamic Application in each account that is being upgraded.

1. Go to the **Discovery Control Panel** page (System > Manage > Classic Discovery).

2. Click the **[Create]** button. The **Discovery Session Editor** page appears:



3. Supply values in the following fields:

   - ***IP Address Discovery List***. Type the URL of your AWS master billing account.
   - ***Other Credentials***. Select the credential you created.
   - ***Discover Non-SNMP***. Select this checkbox.
   - ***Model Devices***. Select this checkbox.

4. Optionally, supply values in the other fields in this page. For a description of the fields in this page, see the ***Discovery & Credentials*** manual.
5. Click the **[Save]** button.
6. The **Discovery Control Panel** page will refresh. Click the lightning bolt icon ( ) for the discovery session you just created.
7. In the pop-up window that appears, click the **[OK]** button. The page displays the progress of the discovery session.

NOTE: If you discontinue monitoring on any devices that are using the Assume Role authentication method, ScienceLogic recommends the best practice of first disabling the devices, deleting the devices from the DCM tree, and then cleaning up any AWS permissions in IAM. This will avoid any unnecessary alerts.

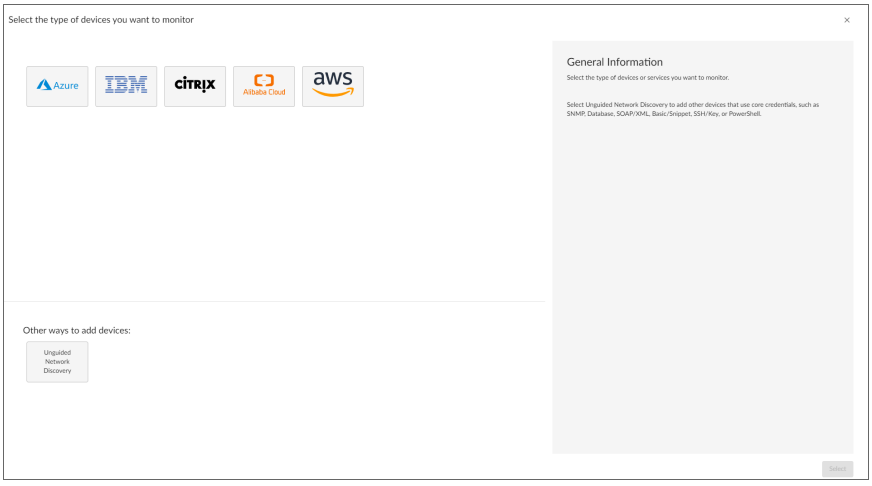
AWS Guided Discovery

You can use the Universal Discovery Framework process in SL1 that guides you through a variety of existing discovery types in addition to traditional SNMP discovery. This process, which is also called "guided discovery", lets you pick a discovery type based on the type of devices you want to monitor. The Universal Discovery workflow includes a button for Amazon Web Services.

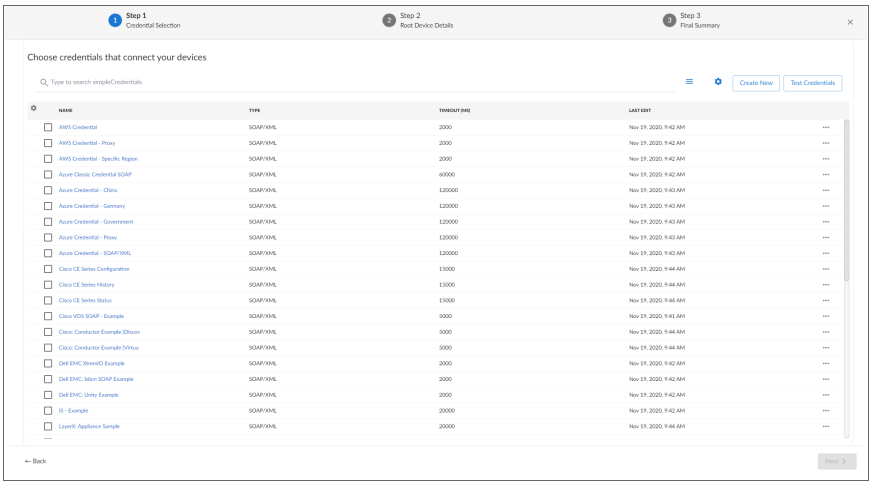
NOTE: If you want to discover one of the third-party products that are available as an option when using the Universal Discovery workflow, you must have the corresponding PowerPack installed on your SL1 system to ensure that the appropriate Dynamic Applications, Device Classes, and other elements can be utilized for discovery. For example, if you want to discover an Amazon Web Services account, you must have the *Amazon Web Services* PowerPack installed.

To run a guided or Universal Discovery:

1. On the **Devices** page ( ) or the **Discovery Sessions** page (Devices > Discovery Sessions), click the **[Add Devices]** button. The **Select** page appears.

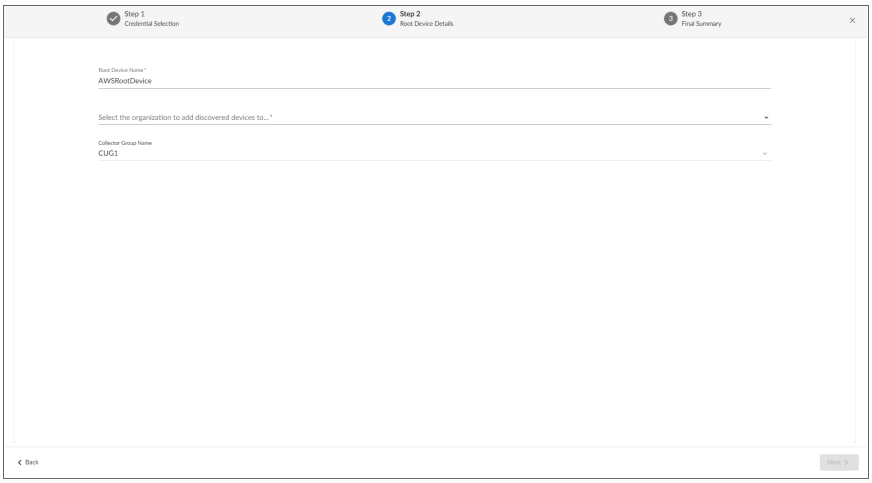


2. Select the **Amazon Web Services** button. Additional information about the requirements for device discovery appears in the **General Information** pane to the right.
3. Click **[Select]**. The **Credential Selection** page appears.



NOTE: During the guided discovery process, you cannot click **[Next]** until the required fields are filled on the page, nor can you skip to future steps. However, you can revisit previous steps that you have already completed.

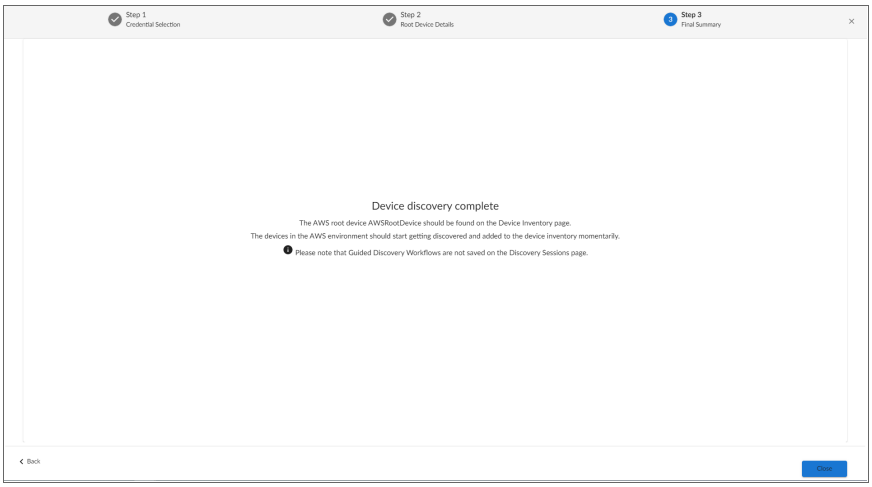
4. On the **Credential Selection** page of the guided discovery process, select the AWS credential that you configured, and then click **[Next]**. The **Root Device Details** page appears.



5. Complete the following fields:

    - ***Root Device Name***. Type the name of the root device for the Amazon Web Services root device you want to monitor.
    - ***Select the organization to add discovered devices to***. Select the name of the organization to which you want to add the discovered device.
    - ***Collector Group Name***. Select an existing collector group to communicate with the discovered device. This field is required.

6. Click **[Next]**. SL1 creates the AWS root device with the appropriate Device Class assigned to it and aligns the relevant Dynamic Applications. The **Final Summary** page appears.



8. Click **[Close]**.

NOTE: The results of a guided discovery do not display on the **Discovery Sessions** page (Devices > Discovery Sessions).

Chapter 6

The AWS Credential Test and Viewing Component Devices

Overview

The following sections describe how to use the AWS credential test, understanding AWS Dynamic Applications, and how to view AWS component devices:

Testing the AWS Credential

NOTE: The Credential Test is for use with the *Manual Discovery* method only.

SL1 includes a Credential Test for Amazon Web Services. Credential Tests define a series of steps that SL1 can execute on demand to validate whether a credential works as expected.

The AWS Credential Test can be used to test a SOAP/XML credential for monitoring AWS using the Dynamic Applications in the *Amazon Web Services* PowerPack. The AWS Credential Test performs the following steps:

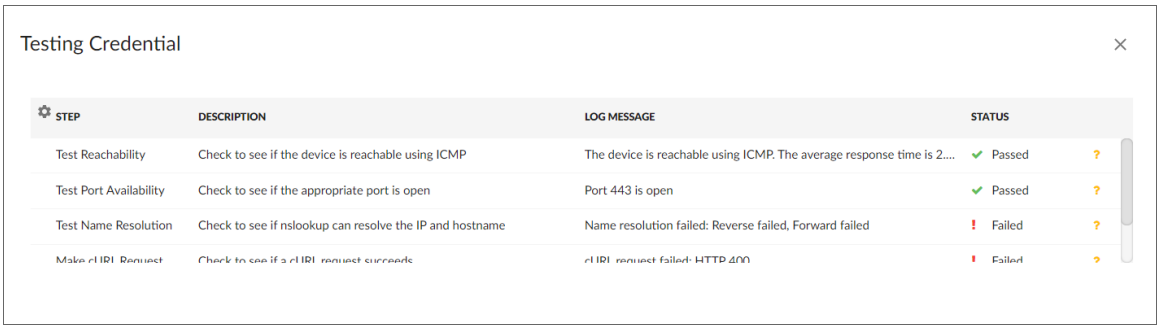
- ***Test Reachability***. Performs an ICMP ping request to the URL for the EC2 service in the region specified in the credential. If a region is not specified in the credential, the us-east-1 region is used.

- ***Test Port Availability***. Performs an NMAP request to TCP port 443 on the URL for the EC2 service in the region specified in the credential. If a region is not specified in the credential, the us-east-1 region is used.
- ***Test Name Resolution***. Performs an nslookup request on the URL for the EC2 service in the region specified in the credential. If a region is not specified in the credential, the us-east-1 region is used.
- ***Make connection to AWS account***. Attempts to connect to the AWS service using the account specified in the credential.
- ***Scan AWS services***. Verifies that the account specified in the credential has access to the services.

NOTE: The AWS Credential Test does not support the testing of credentials that connect to AWS through a proxy server.

To test the AWS credential:

1. Go to the **Credentials** page (Manage > Credentials).
2. Locate the credential you wish to test, select the **Actions** button (···) next to it and click *Test*.
3. The **Credential Test Form** modal page appears. Fill out the following fields on this page:
   - ***Credential***. This field is read-only and displays the name of the credential you selected.
   - ***Select Credential Test***. Select **AWS Credential Test**.
   - ***Collector***. Select the All-In-One Appliance or Data Collector that will run the test.
   - ***IP or Hostname to Test***. Enter a valid IP address.
4. Click the **[Run Test]** button to run the credential test. The **Testing Credential** window appears:



The **Testing Credential** window displays a log entry for each step in the credential test. The steps performed are different for each credential test. The log entry for each step includes the following information:

- ***Step***. The name of the step.
- ***Description***. A description of the action performed during the step.
- ***Log Message***. The result of the step for this execution of the credential test.

- ***Status***. Whether the result of this step indicates the credential and/or the network environment is configured correctly (Passed) or incorrectly (Failed).
- ***Step Tip***. Mouse over the question mark icon (?) to display the tip text. The tip text recommends what to do to change the credential and/or the network environment if the step has a status of "Failed".

Testing the AWS Credential in the SL1 Classic User Interface

> **NOTE:** The Credential Test is for use with the *Manual Discovery* method only.

SL1 includes a Credential Test for Amazon Web Services. Credential Tests define a series of steps that SL1 can execute on demand to validate whether a credential works as expected.

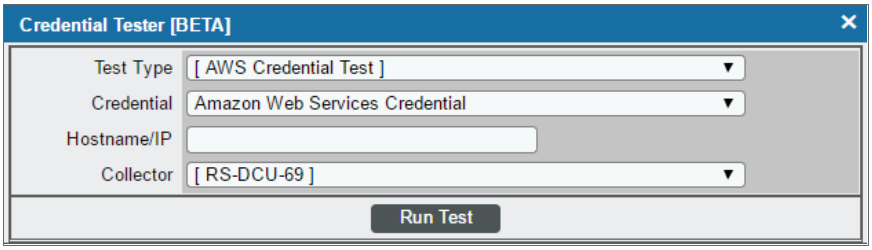
The AWS Credential Test can be used to test a SOAP/XML credential for monitoring AWS using the Dynamic Applications in the *Amazon Web Services* PowerPack. The AWS Credential Test performs the following steps:

- ***Test Reachability***. Performs an ICMP ping request to the URL for the EC2 service in the region specified in the credential. If a region is not specified in the credential, the us-east-1 region is used.
- ***Test Port Availability***. Performs an NMAP request to TCP port 443 on the URL for the EC2 service in the region specified in the credential. If a region is not specified in the credential, the us-east-1 region is used.
- ***Test Name Resolution***. Performs an nslookup request on the URL for the EC2 service in the region specified in the credential. If a region is not specified in the credential, the us-east-1 region is used.
- ***Make connection to AWS account***. Attempts to connect to the AWS service using the account specified in the credential.
- ***Scan AWS services***. Verifies that the account specified in the credential has access to the services.

> **NOTE:** The AWS Credential Test does not support the testing of credentials that connect to AWS through a proxy server.

To test the AWS credential:

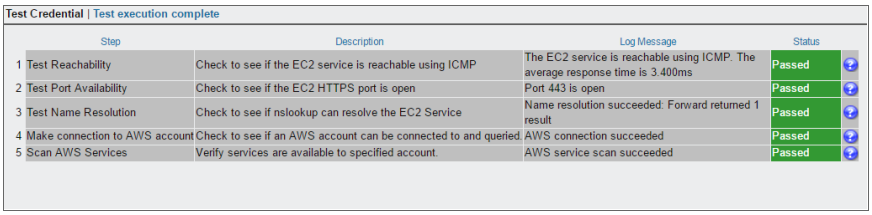
1. Go to the **Credential Test Management** page (System > Customize > Credential Tests).
2. Locate the **AWS Credential Test** and click its lightning bolt icon ( ). The **Credential Tester** modal page appears:



3. Supply values in the following fields:

    - ***Test Type***. This field is pre-populated with the credential test you selected.
    - ***Credential***. Select the credential to test. This drop-down list includes only credentials that you have access to that can be tested using the selected credential test.
    - ***Hostname/IP***. Leave this field blank.
    - ***Collector***. Select the All-In-One Appliance or Data Collector that will run the test.

4. Click the **[Run Test]** button to run the credential test. The **Test Credential** window appears:



The **Test Credential** window displays a log entry for each step in the credential test. The steps performed are different for each credential test. The log entry for each step includes the following information:

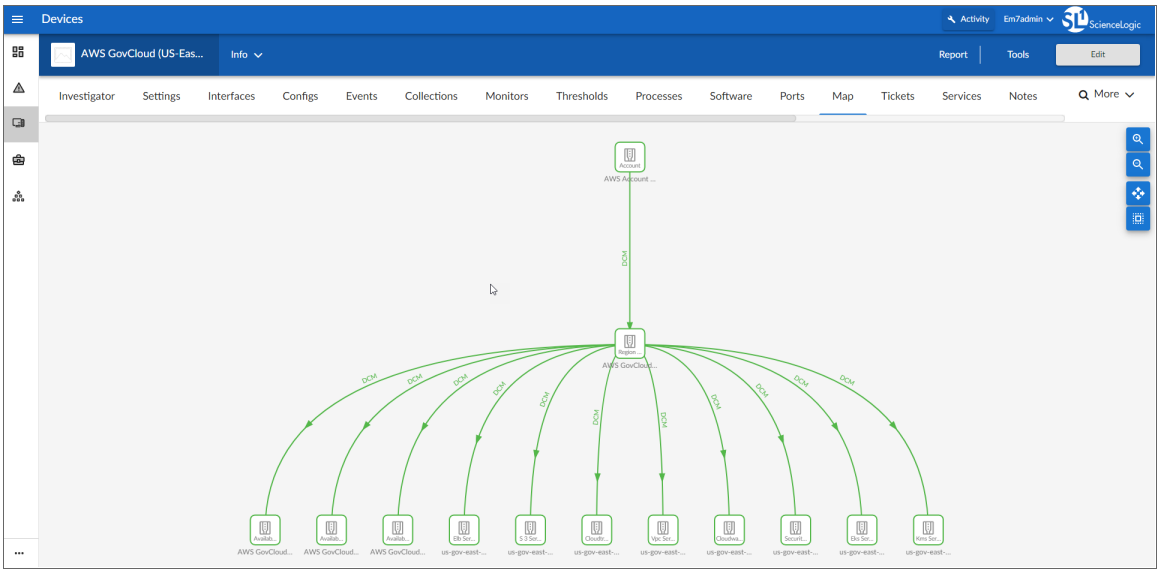
- ***Step***. The name of the step.
- ***Description***. A description of the action performed during the step.
- ***Log Message***. The result of the step for this credential test.
- ***Status***. Whether the result of this step indicates the credential or the network environment is configured correctly (Passed) or incorrectly (Failed).
- ***Step Tip***. Mouse over the question mark icon ( ) to display the tip text. The tip text recommends what to do to change the credential or the network environment if the step has a status of "Failed".

Viewing AWS Component Devices

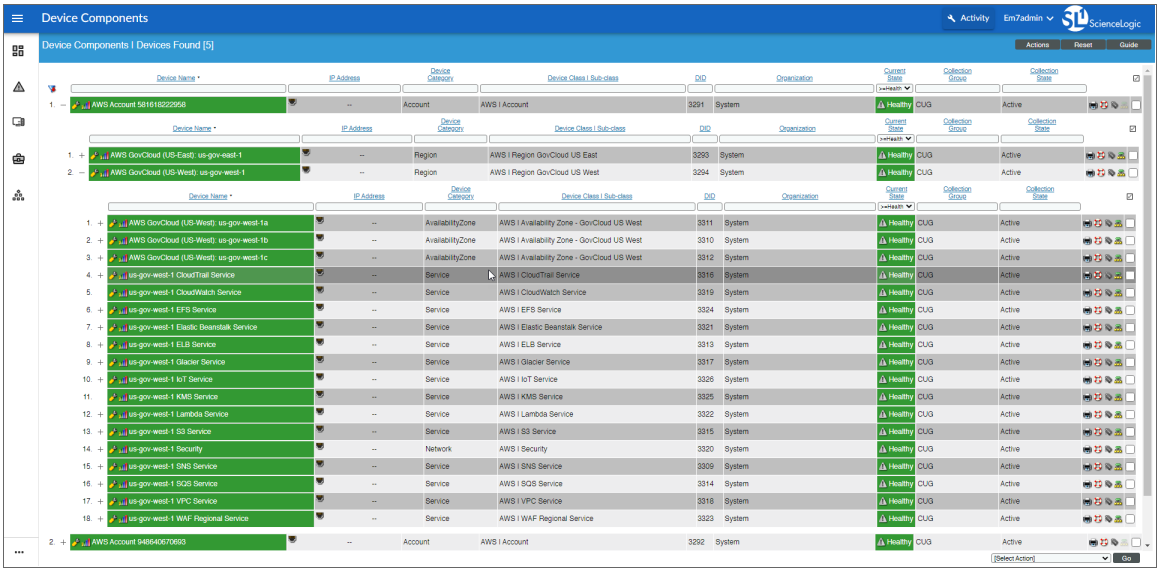
When SL1 performs collection for the AWS virtual device, SL1 will create component devices that represent each element in your AWS infrastructure and align other Dynamic Applications to those component devices. Some of the Dynamic Applications aligned to the component devices will also be used to create additional component devices. All component devices appear in the **Devices** page.

In addition to the **Devices** page, you can view the AWS service and all associated component devices in the following places in the user interface:

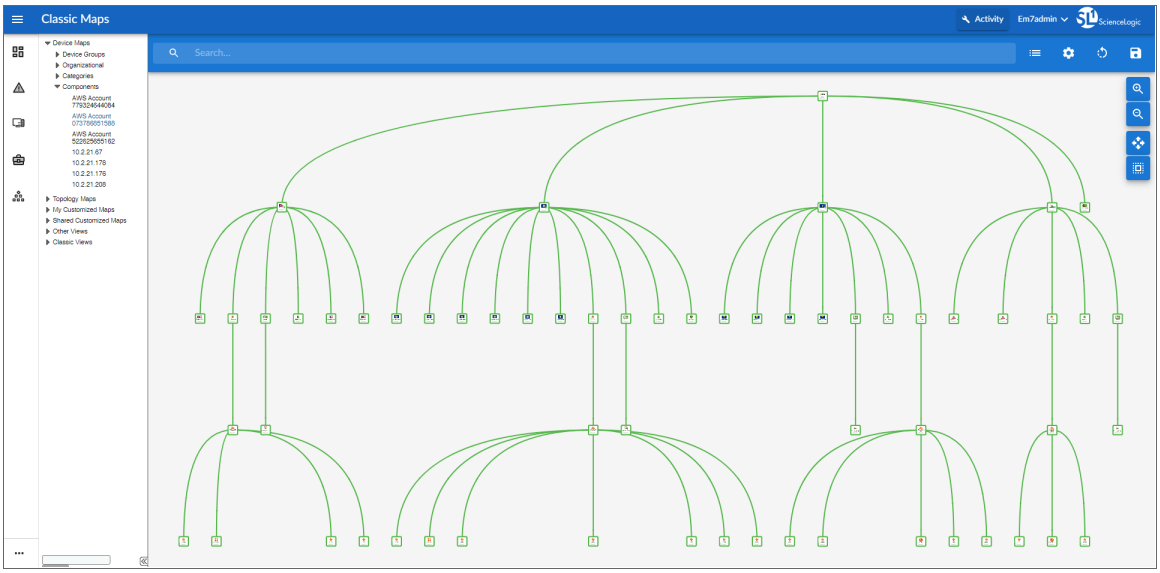
- The **Device Investigator** Map page (click **Map** in the **Device Investigator** page) displays a map of a particular device and all of the devices with which it has parent-child relationships. Double-clicking any of the listed devices reloads the page to make the selected device the primary device



- The **Device Components** page (Devices > Device Components) displays a list of all root devices and component devices discovered by SL1 in an indented view, so you can easily view the hierarchy and relationships between child devices, parent devices, and root devices. To view the component devices associated with an AWS service, find the AWS virtual device and click its plus icon (+).



- The **Component Map** page (Classic Maps > Device Maps > Components) allows you to view devices by root node and view the relationships between root nodes, parent components, and child components in a map. This makes it easy to visualize and manage root nodes and their components. SL1 automatically updates the **Component Map** as new component devices are discovered. SL1 also updates each map with the latest status and event information. To view the map for an AWS service, go to Classic Maps > Device Maps > Components, and select the map from the list in the left NavBar. To learn more about the **Component Map** page, see the ***Maps*** manual.



Relationships Between Component Devices

In addition to the parent/child relationships between component devices, relationships are automatically created by the Dynamic Applications in the *Amazon Web Services* PowerPack between the following component devices:

- AWS API Gateway Services and AWS Network Load Balancers
- AWS API Instances and AWS Lambda Functions
- AWS Application ELBs and AWS Availability Zones
- AWS Application ELBs and AWS Route 53-Hosted Zones
- AWS Application ELBs and AWS Security Groups
- AWS Application ELBs and AWS Target Groups
- AWS Application ELBs and AWS VPC Instances
- AWS Auto Scale Groups and AWS Auto Scale Launch Configurations
- AWS Direct Connect Virtual Instances and AWS Virtual Private Gateways
- AWS ECS Instances and AWS EC2 Instances
- AWS ECS Services and AWS Classic Load Balancers
- AWS ECS Services and AWS Security Groups
- AWS ECS Services and AWS Target Groups
- AWS ECS Services and AWS VPC Instances
- AWS ECS Services and AWS VPC Subnets
- AWS EC2 Instances and AWS Auto Scale Groups
- AWS EC2 Instances and AWS EBS Volumes
- AWS EC2 Instances and AWS Elastic Beanstalk Applications

- AWS EC2 Instances and AWS ELB Instances
- AWS EC2 Instances and AWS EMR Instances
- AWS EC2 Instances and AWS OpsWorks Instances
- AWS EC2 Instances and AWS Security Groups
- AWS EC2 Instances and AWS Target Groups
- AWS EC2 Instances and AWS VPC Instances
- AWS EC2 Instances and AWS VPC Subnets
- AWS EC2 Instances and the Cisco Cloud Center application
- AWS Lambda Functions and AWS Security Groups
- AWS Lambda Functions and AWS Simple Notification Services (SNS)
- AWS Lambda Functions and AWS Simple Queue Services (SQS)
- AWS Lambda Functions and AWS VPC Instances
- AWS Lambda Functions and AWS VPC Subnets
- AWS Lambda Function Qualified Services and AWS Security Groups
- AWS Lambda Function Qualified Services and AWS VPC Instances
- AWS Lambda Function Qualified Services and AWS VPC Subnets
- AWS Lambda Function Replicas and their parent AWS Lambda Function Versions
- AWS Network ELBs and AWS Availability Zones
- AWS Network ELBs and AWS Route 53-Hosted Zones
- AWS Network ELBs and AWS Target Groups
- AWS Network ELBs and AWS VPC Instances
- AWS Organizations and AWS Accounts
- AWS RDS Aurora Clusters and AWS RDS DB Instances
- AWS Redshift Instances and AWS Security Groups
- AWS Redshift Instances and AWS VPC Instances
- AWS Route Tables and AWS Virtual Private Gateways
- AWS Route Tables and AWS VPC Subnets
- AWS S3 Instances and AWS CloudTrail Instances
- AWS Security Groups and AWS VPC Instances
- AWS SNS Instances and AWS CloudTrail Instances
- AWS SNS Instances and AWS Glacier Instances
- AWS Transit Gateways and AWS VPC Instances
- AWS VPC Instances and AWS ELB Instances
- AWS VPC Instances and AWS Target Groups
- AWS VPC Instances and other intra-account AWS VPC Instances

Vanishing Component Devices

If SL1 cannot retrieve information about a component device for the amount of time specified in the ***Component Vanish Timeout*** field (in either the **Global Threshold Settings** page, the **Device Thresholds** page for the component device, or the **Device Thresholds** page for a device higher in the component tree), SL1 sets the device to "vanished".

When a device is set to "vanished", SL1 stops trying to collect data about the component device. The vanished device will not appear in reports or views. The vanished device will appear only in the **Vanished Device Manager** page. When a device is set to "vanished", all children of that device are also set to "vanished".

> **NOTE:** This section describes the standard device vanishing behavior that ***does not*** use the "AWS: Vanish Terminated EC2 Instances" Run Book Action and Automation policies. If you use the "AWS: Vanish Terminated EC2 Instances" Run Book Action and Automation policies, see the chapter on **"AWS Run Book Actions and Automations"** for more information about device vanishing.

Most AWS component devices operate using the standard SL1 vanishing logic: If the device is terminated in AWS, it then becomes unavailable in SL1. If the device is unavailable for the amount of time specified in the ***Component Vanish Timeout*** field, then that device is vanished.

However, two AWS component device types operate using slightly different logic:

- ***EC2***. EC2 instances that are deleted in AWS still appear in the AWS portal for one to two hours in a *terminated* state. If SL1 polls that device and receives a response from AWS that the EC2 is terminated, SL1 will classify the device as unavailable. If the ***Component Vanish Timeout*** setting has been enabled, then SL1 will vanish this device automatically. If, however, the EC2 instance has merely been *stopped* rather than terminated, SL1 will not vanish the device, even if the ***Component Vanish Timeout*** setting has been enabled.
- ***RDS***. RDS instances that have a status of *stopped* or *stopping* in AWS will be classified as unavailable in SL1. If the ***Component Vanish Timeout*** setting has been enabled, then SL1 will vanish this device automatically.

ScienceLogic recommends setting the ***Component Vanish Timeout*** to *120 minutes* when monitoring AWS accounts.

For more information about vanishing devices, see the chapter on "Vanishing & Purging Devices" in the ***Device Management*** manual.

Chapter 7

Configuring Inbound CloudWatch Alarms

Overview

The following sections describe the CloudWatch alarm Event Policies that are included in the *Amazon Web Services* PowerPack and information about configuring CloudWatch and SL1 to generate events based on CloudWatch alarms:

CloudWatch Alarm Event Policies

Amazon CloudWatch is a service that allows you to monitor your AWS resources and applications in near real-time. You can use CloudWatch to collect and track metrics, and use CloudWatch alarms to send notifications or automatically trigger changes to the resources being monitored based on rules that you define.

In addition to SL1 collecting metrics for AWS instances, you can configure CloudWatch to send alarm information to SL1 via API. SL1 can then generate an event for each alarm.

The *Amazon Web Services* PowerPack includes an "AWS :CloudWatch Alarms Performance" Dynamic Application. This Dynamic Application monitors CloudWatch alarms and associates the alarms with the appropriate AWS component devices, if applicable. If an appropriate component device does not exist in SL1 or cannot be determined, the alarm is instead associated with the component device for the AWS account.

CAUTION: The performance data collected by the "AWS: CloudWatch Alarms Performance" Dynamic Application is metadata intended to give general insight into the alarm activity the Dynamic Application is processing. This metadata can help identify overall trends, but users should be cautioned that the data presented can be imprecise in certain scenarios, such as when the Dynamic Application is being run in debug mode while data is still being collected.

The *Amazon Web Services* PowerPack also includes several pre-defined event policies for CloudWatch alarms:

Alarm Type	Alarm State	Event Policy Name	Description	Event Source	Severity
Action	Failed	AWS: CloudWatchAlarm_ Action_Failed	An Amazon CloudWatch alarm action has failed.	API	Major
Action	InProgress	AWS: CloudWatchAlarm_ Action_InProgress	An Amazon CloudWatch alarm action is in progress.	API	Notice
Action	Succeeded	AWS: CloudWatchAlarm_ Action_Succeeded	An Amazon CloudWatch alarm action has succeeded.	API	Notice
Configuration Update	Configuration Update	AWS: CloudWatchAlarm_ ConfigurationUpdate	A ConfigurationUpdate alarm type is received.	API	Notice
Status Update	Alarm	AWS: CloudWatchAlarm_ StateUpdate_Alarm	A CloudWatch alarm transitions to an "Alarm" state.	API	Major
Status Update	Insufficient Data	AWS: CloudWatchAlarm_ StateUpdate_ InsufficientData	A CloudWatch alarm transitions to an "Insufficient Data" state.	API	Notice
Status Update	OK	AWS: CloudWatchAlarm_ StateUpdate_OK	A CloudWatch alarm transitions to an "OK" state.	API	Healthy

These events are aligned to AWS Account component devices in the following way:

- If the CloudWatch alarm is configured on a device that is discovered in SL1, then the event in SL1 will be aligned with the component device for that instance.
- If the CloudWatch alarm is configured on a device that is either not discovered or not supported by CloudWatch, or if SL1 cannot determine a correct component device, then that alarm will be aligned to the Account component device.

The "AWS: CloudWatch Alarms Performance" Dynamic Application and related Event Policies are disabled by default. If you want SL1 to monitor CloudWatch alarms and generate events about them, you must enable the Dynamic Application and Event Policies. You must also configure the Dynamic Application to specify which types of alarms you want to monitor.

For more information about enabling and configuring the "AWS: CloudWatch Alarms Performance" Dynamic Application, see the *Configuring the "AWS: CloudWatch Alarms Performance" Dynamic Application* section. For more information about enabling the CloudWatch alarms Event Policies, see the *Enabling CloudWatch Alarm Events in the ScienceLogic Platform* section.

NOTE: Because the AWS services make new data points available at varying time intervals, there might be a difference in the data points collected by SL1 when compared to data presented in CloudWatch at a given time. The difference between SL1 and CloudWatch is typically less than 1%.

NOTE: If an event expires and the CloudWatch alarm in AWS is still in an "Alarm" state, SL1 will not generate any additional CloudWatch events unless that CloudWatch alarm changes states in AWS.

Creating Custom CloudWatch Metrics

A CloudWatch alarm watches a single metric and performs one or more actions based on the value of the metric relative to a threshold over a number of time periods. A CloudWatch metric consists of the following elements:

- A ***namespace***, such as *AWS/EC2*
- A ***metric name***, such as *CPUUtilization*
- A ***value***, such as *42*
- A ***dimension*** that identifies a particular resource instance, such as {'*Name*': '*InstanceId*', '*Value*': '*i-0a6a989bb8d57b074*'}

NOTE: For a complete list of supported CloudWatch Metrics and Dimensions, see https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/CW_Support_For_AWS.html.

The *Amazon Web Services* PowerPack uses the metric dimensions identified in an alarm to associate the alarm message to a particular ScienceLogic component device. The following table lists the services that are currently supported and the dimensions used to associate an alarm to a component device:

AWS Service	Dimension
API Gateway	'ApiName' | 'ApiName | Stage' **NOTE:** ScienceLogic recommends that you create API Gateways with unique names within the same region.
ApplicationELB	'LoadBalancer' | 'TargetGroup'
CloudFront	'DistributionId'
Direct Connect	'ConnectionID'
DynamoDB	'TableName'
EBS	'VolumeId'
ECS	'ClusterName' | 'ServiceName'
EC2	'InstanceId' | 'AutoScalingGroupName'
EKS Cluster	'ClusterName'
ElasticBeanstalk	'EnvironmentName'
ElastiCache	'CacheClusterId' **NOTE:** Alarms for this service will be associated with the component device for the AWS account.
ElasticMapReduce	'JobFlowId'
ELB	'LoadBalancerName'
Glacier	'VaultId' **NOTE:** This service is not supported by CloudWatch. You must define a custom metric and publish the metric to the CloudWatch service using an agent toolkit or the AWS command-line interface.
Lambda	'FunctionName', 'Resource', 'Version', 'Alias', 'Executed Version' **NOTE:** Alarms "across all functions" for this service will be associated with the component device for the AWS account. Alarms "by function name" will be aligned to a specific Lambda function.
NetworkELB	'LoadBalancer' | 'TargetGroup'
OpsWorks	'StackId' | 'InstanceId'
RDS	'DBInstanceIdentifier' **NOTE:** Alarms for this service will be associated with the component device for the AWS account.

AWS Service	Dimension
Redshift	'ClusterIdentifier' **NOTE:** Alarms for this service will be associated with the component device for the AWS account.
Route53	'HealthCheckId'
Shield	'ShieldService' **NOTE:** CloudWatch alarms are available only for Shield Advanced Services.
SNS	'TopicName'
SQS	'QueueName'
StorageGateway	'GatewayId' | 'VolumeId'
S3	'BucketName'
WAF	'WebACLId'

AWS enables users to create custom metrics for these services and then publish those metrics to CloudWatch using the AWS command-line interface (CLI) or an application programming interface (API). The Dynamic Applications in the *Amazon Web Services* PowerPack can then collect data for these custom AWS metrics (which are not in the "AWS" cloud namespace).

NOTE: For the *Amazon Web Services* PowerPack to collect data for these custom metrics, you must enable certain Dynamic Applications that are disabled by default. For more information, see the ***Enabling Custom Metrics Collection in the ScienceLogic Platform*** section.

When creating a custom metric, it is important that the metric is correctly formed. For SL1 to align a custom metric to a particular ScienceLogic component device, the following must be true:

- The metric namespace must include the service being tracked.

  For example, *MyVendorName/EC2* would be a valid namespace that the *Amazon Web Services* PowerPack could use to identify the EC2 service for a tracked metric.

- The dimension must include one or more of the dimensions listed in the preceding table. The dimension enables SL1 to identify which device to associate with the alarm.

  For example, if the dimension included {'*Name*': '*InstanceId*', '*Value*': '*i-0a6a989bb8d57b074*'}, this would identify the EC2 component. Other dimensions are permitted, but 'InstanceId' is necessary to locate the EC2 instance.

  If the component device was an AutoScaleGroup component that is also under the EC2 service, then the dimension might look like this: {'*Name*': '*AutoScalingGroupName*', '*Value*': '*Y1Z55ZJ390UP*'}.

NOTE: If the CloudWatch event cannot align to a particular ScienceLogic component device, it will instead align to the component device for the AWS account.

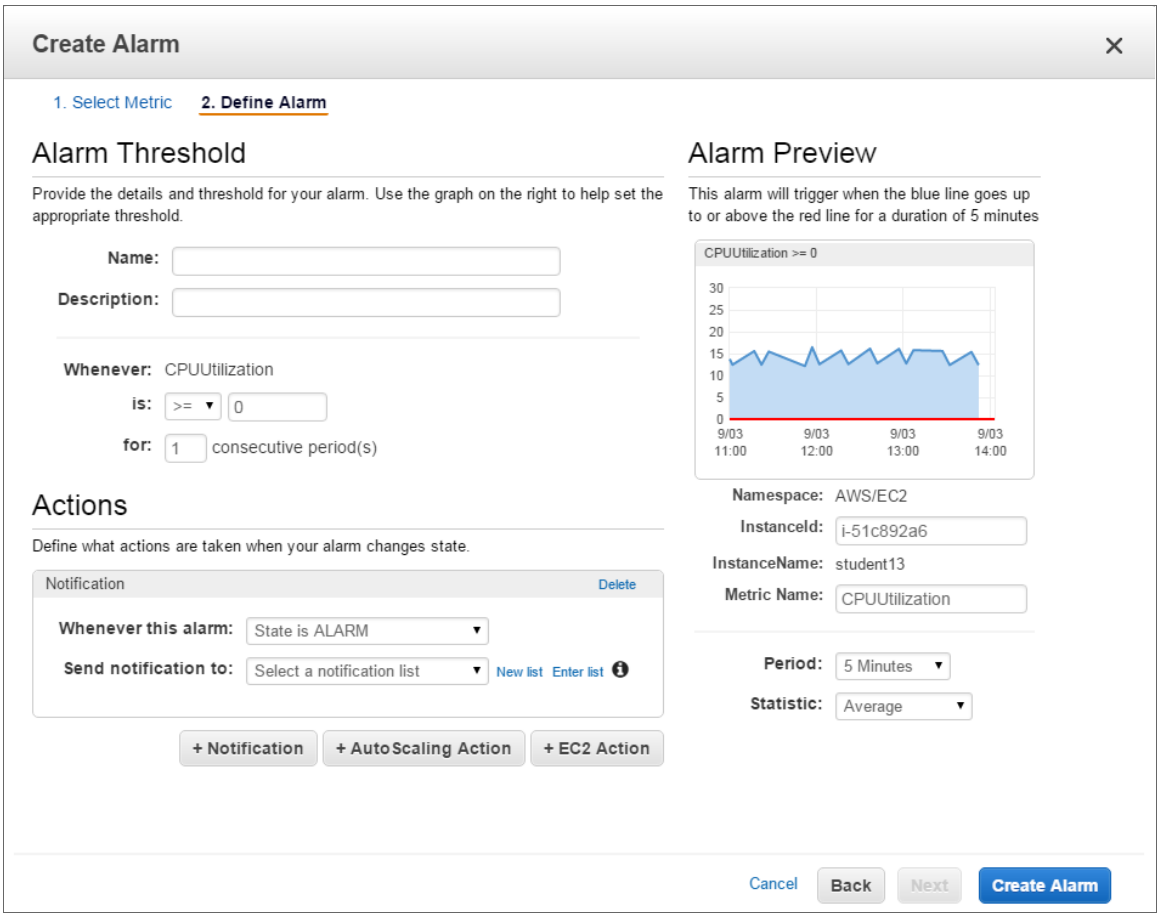
Configuring CloudWatch to Send Alarms for a Metric

To configure CloudWatch to send alarms to SL1 for a metric, perform the following steps:

1. Open a browser session and go to aws.amazon.com.
2. Click **[My Account]** and then select *AWS Management Console*. If you are not currently logged in to the AWS site, you will be prompted to log in:

3. In the **AWS Management Console**, under the **Management Tools** heading, click **[CloudWatch]**.
4. Click the **[Browse Metrics]** button.
5. Select the metric for which you want CloudWatch to send alarms.
6. Select the instances for which you want CloudWatch to send alarms for this metric.

7. Click the **[Create Alarm]** button. The **Create Alarm** page is displayed:



8. Specify a Name and Description for the alarm.
9. If you have previously configured an alarm for SL1, select the notification list for SL1 in the ***Send notification to*** field. Otherwise, select the **[New list]** link to the right of the ***Send notification to*** field and supply values in the following fields:

   - ***Send notification to***. Enter a name for the new notification list. If you add additional alarms, you can select the name you enter in this field instead of re-entering the email address.
   - ***Email list***. Enter the email address to which you want CloudWatch notifications sent.

10. Supply values in the other fields in this page as desired.
11. Click the **[Create Alarm]** button.
12. Log in to the email account you configured to receive email from the email alias.
13. Open the confirmation email from Amazon and click the **[Confirm subscription]** link.

Enabling Custom Metrics Collection in SL1

AWS enables users to publish their own custom metrics to CloudWatch using the AWS command-line interface (CLI) or an application programming interface (API). The *Amazon Web Services* PowerPack includes Dynamic Applications that collect data for custom AWS metrics (which are not in the "AWS" cloud namespace). However, these Dynamic Applications are disabled by default and must be enabled for use.

To enable these Dynamic Applications:

1. Go to the **Dynamic Applications Manager** page (System > Manage > Applications).
2. Click the wrench icon () for the "AWS: Custom Metrics" Dynamic Application. The **Dynamic Applications Properties Editor** page appears.
3. In the ***Operational State*** field, select *Enabled*.
4. Click the **[Save]** button.
5. Repeat steps 1 - 4 for the "AWS: Custom Metrics Cache" Dynamic Application.

Configuring the "AWS: CloudWatch Alarms Performance" Dynamic Application

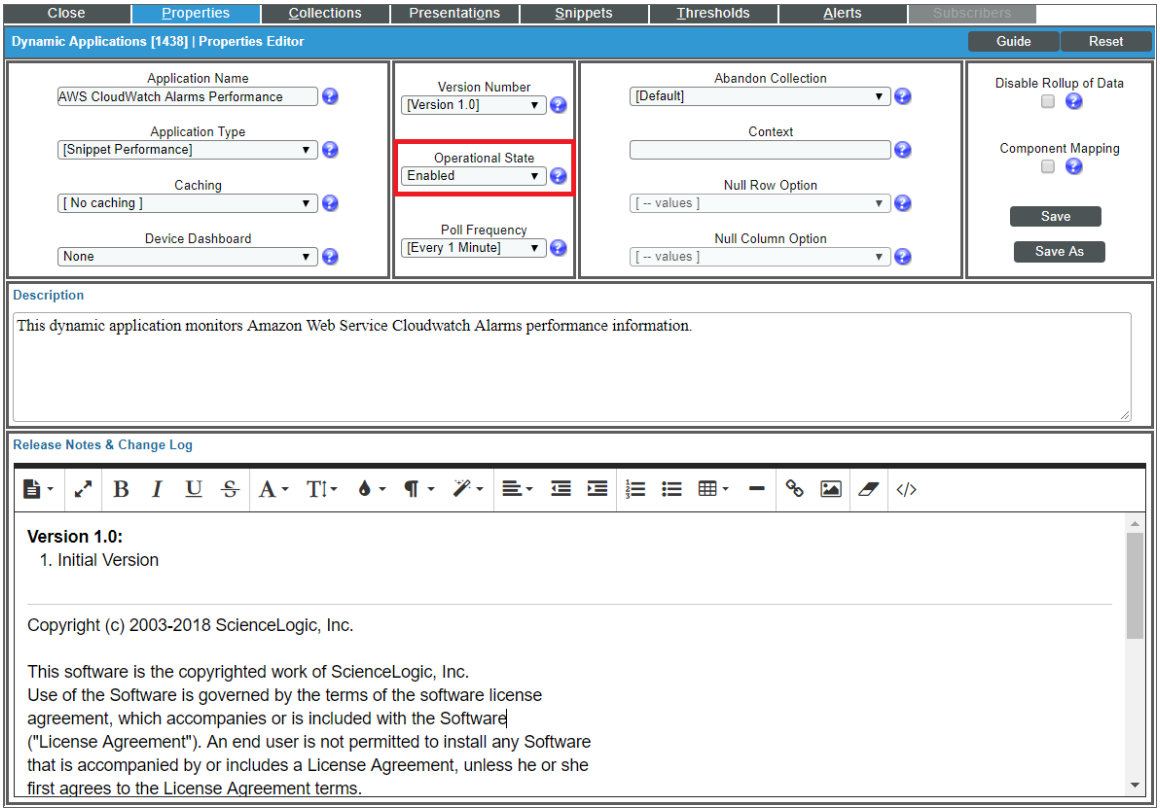
The *Amazon Web Services* PowerPack includes an "AWS: CloudWatch Alarms Performance" Dynamic Application that monitors CloudWatch alarms and associates the alarms with the appropriate AWS component devices, if applicable. This Dynamic Application must be enabled if you want SL1 to generate CloudWatch alarm events.

NOTE: If an appropriate component device does not exist in SL1 or cannot be determined, the alarm is instead associated with the "Account" component device.

To enable the "AWS: CloudWatch Alarms Performance" Dynamic Application:

1. Go to the **Dynamic Applications Manager** page (System > Manage > Applications).

2. Locate the "AWS: CloudWatch Alarms Performance" Dynamic Application and then click its wrench icon ( ). The **Dynamic Applications Properties Editor** page appears.



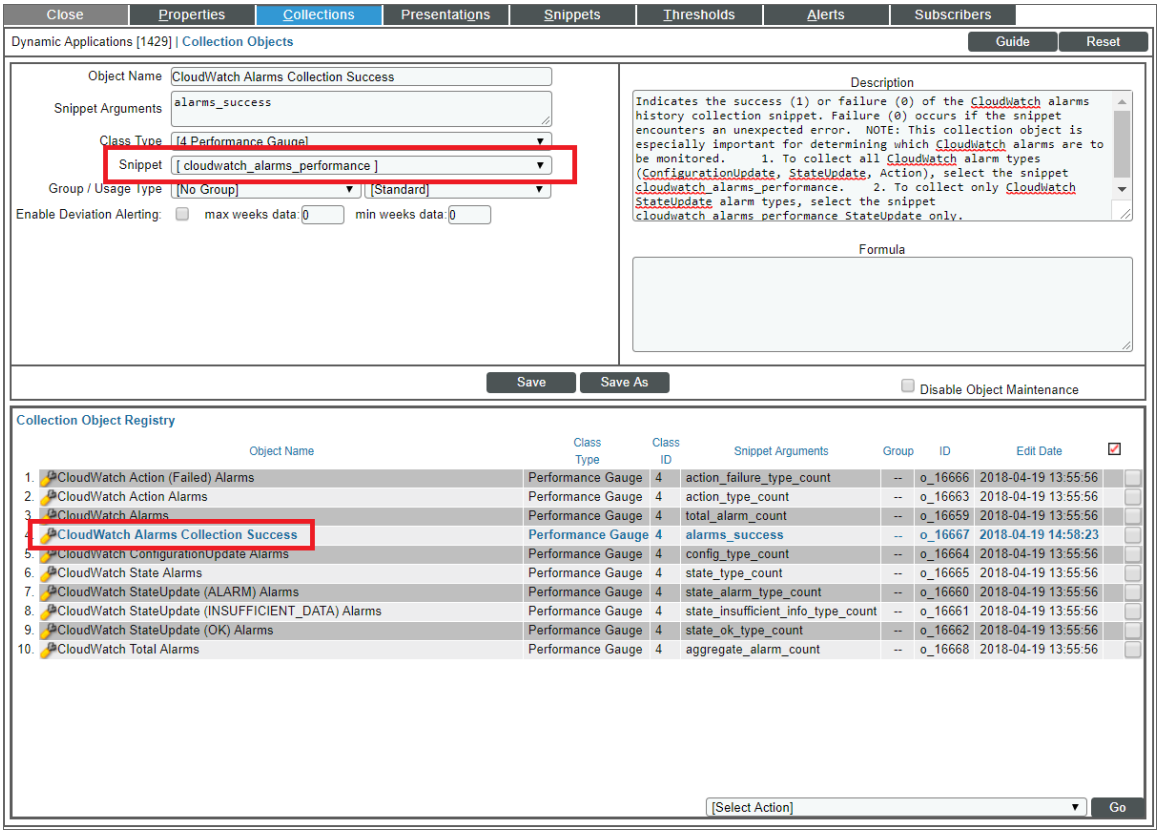
3. In the ***Operational State*** field, select *Enabled*.
4. Click **[Save]**.

By default, the "AWS: CloudWatch Alarms Performance" Dynamic Application monitors only the "StateUpdate" type of CloudWatch alarms. If you want the Dynamic Application to also monitor "Action" and "ConfigurationUpdate" alarm types, you must configure the Dynamic Application to do so.

To configure the "AWS: CloudWatch Alarms Performance" Dynamic Application to monitor all CloudWatch alarm types:

1. Go to the **Dynamic Applications Manager** page (System > Manage > Applications).
2. Locate the "AWS: CloudWatch Alarms Performance" Dynamic Application and then click its wrench icon ( ). The **Dynamic Applications Properties Editor** page appears.
3. Click the **[Collections]** tab. The **Collection Objects** page appears.

4. On the **Collection Objects** page, locate the "CloudWatch Alarms Collection Success" collection object and then click its wrench icon ( ).



5. In the ***Snippet*** field, select one of the following options:

   - *cloudwatch_alarms_performance*. This option is selected by default. This snippet triggers notifications if any alarm configuration is modified.
   - *cloudwatch_alarms_performance_StateUpdate_only*. This snippet will only trigger events for State Update alarms.
   - *cloudwatch_alarms_statistics*. This snippet will trigger events for all CloudWatch alarm types (Action, Configuration Update, and State Update).

NOTE: If you want to revert back to monitoring only the "StateUpdate" CloudWatch alarms, then select *cloudwatch_alarms_performance_StateUpdate_only* in the ***Snippet*** field.

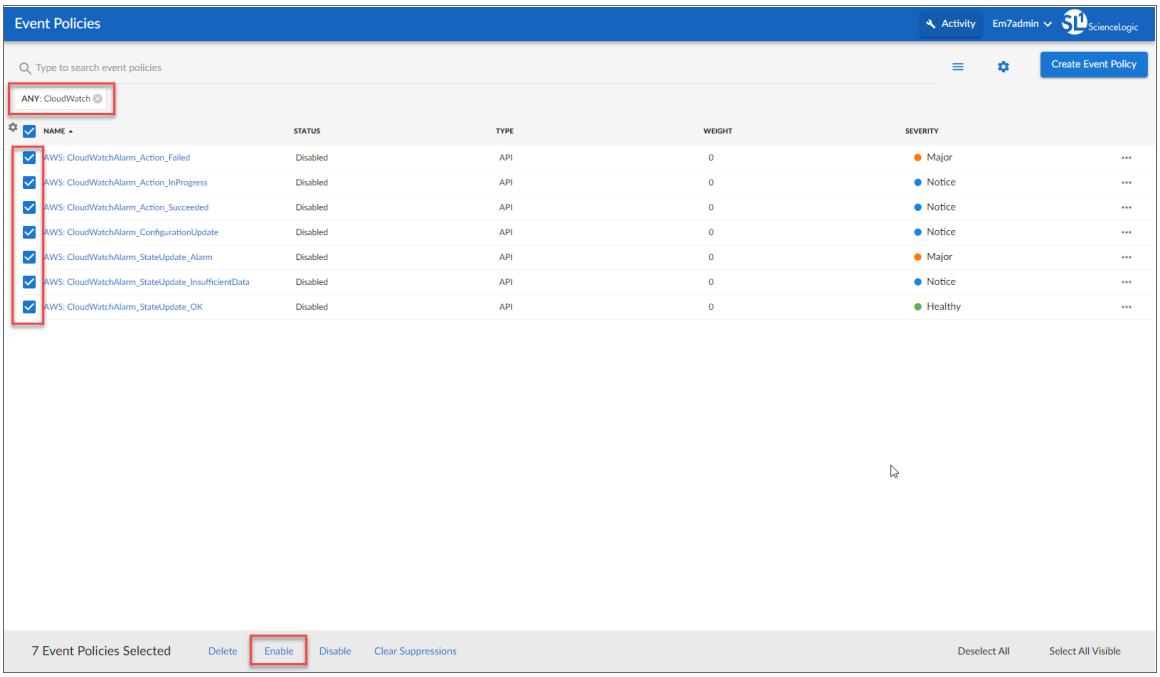
6. Click **[Save]**. This Dynamic Application will be executed every 1 minute.

Enabling CloudWatch Alarm Events in SL1

The *Amazon Web Services* PowerPack also includes several pre-defined event policies for CloudWatch alarms. These Event Policies must be enabled if you want SL1 to generate CloudWatch alarm events.

To enable the CloudWatch alarms Event Policies:

1. Go to the **Event Policies** page (Events > Event Policies).
2. Perform a search for "CloudWatch".

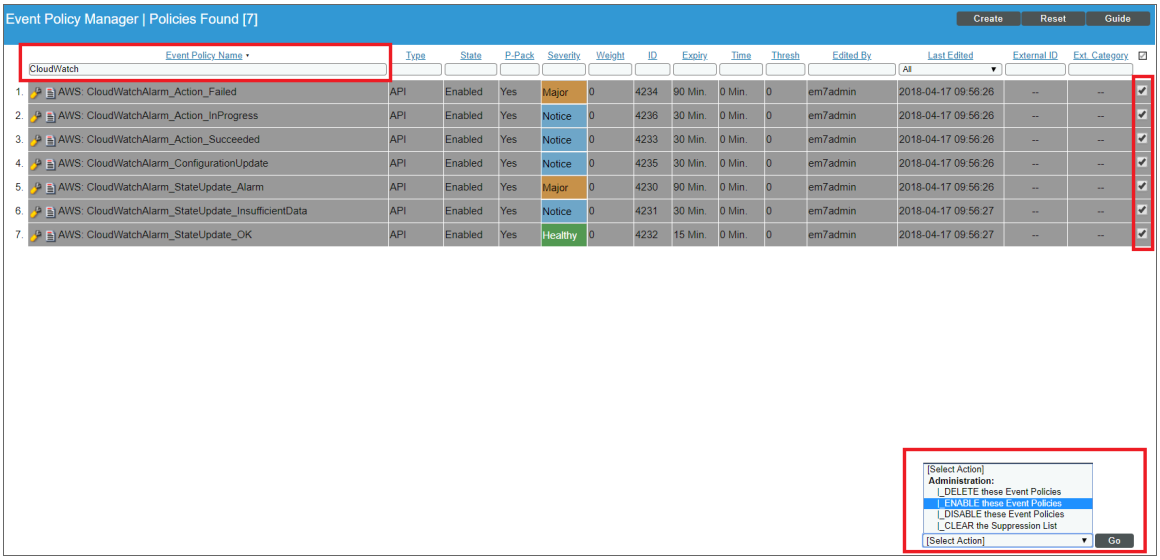


3. Select the check boxes for the events you want to enable.
4. Select **Enable** at the bottom of the screen.

To enable the CloudWatch alarms Event Policies in the SL1 classic user interface:

1. Go to the **Event Policy Manager** page (Registry > Events > Event Manager).

2. In the ***Event Policy Name*** filter-while-you-type field, type "CloudWatch".



3. Select the check boxes for the events you want to enable.
4. In the ***Select Action*** drop-down field, select *ENABLE these Event Policies*.
5. Click **[Go]**.

Preserving CloudWatch Alarm Event Changes

If you have modified CloudWatch alarm event policies that are included in the *Amazon Web Services* PowerPack, those changes will be overwritten when the PowerPack is updated in your system. If you have modified event policies that are included in the PowerPack, you can:

- Re-implement those changes after each update of the *Amazon Web Services* PowerPack.
- Remove the content from the PowerPack on your system. When the *Amazon Web Services* PowerPack is updated in your system, updated versions of this content will not be installed on your system and your local changes will be preserved.

To remove event policies from the *Amazon Web Services* PowerPack on your system:

1. Go to the **PowerPack Manager** page (System > Manage > PowerPacks).
2. Click the wrench icon ( ) for the *Amazon Web Services* PowerPack. The **Editing PowerPack** page appears.
3. In the left NavBar of the **Editing PowerPack** page, click **[Event Policies]**. The **Embedded Event Policies** and **Available Event Policies** panes appear.
4. In the upper pane, click the bomb icon ( ) for each event policy that you want to remove from the *Amazon Web Services* PowerPack on your system.

Chapter 8

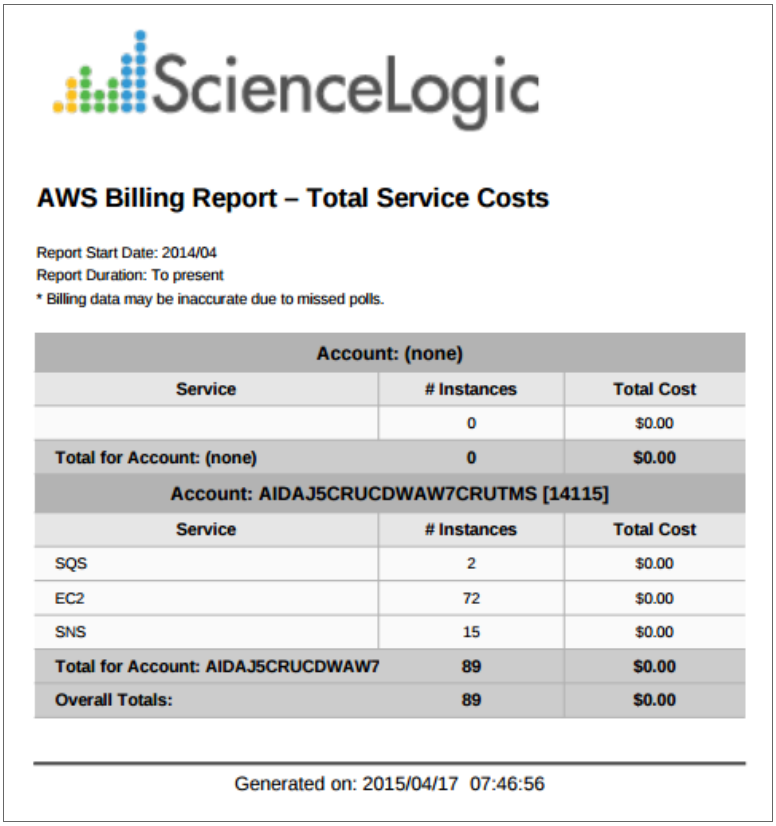
Reports

Overview

The following sections describe the reports that are included in the *Amazon Web Services* PowerPack:

AWS Billing Report

This report displays service costs for Amazon Web Services. The report includes Total, Monthly, Quarterly, and Annual costs.



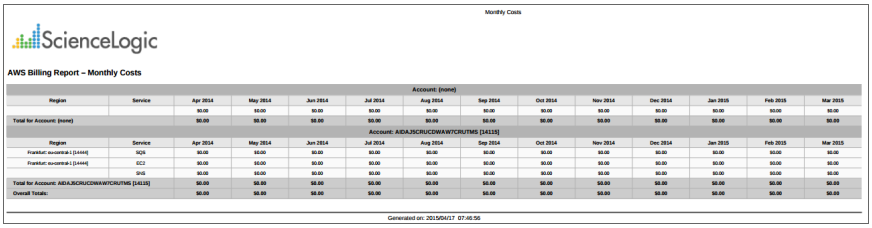
ScienceLogic

AWS Billing Report – Total Service Costs

Report Start Date: 2014/04
Report Duration: To present
* Billing data may be inaccurate due to missed polls.

Account: (none)		
Service	**# Instances**	**Total Cost**
	0	$0.00
Total for Account: (none)	**0**	**$0.00**
Account: AIDAJ5CRUCDWAW7CRUTMS [14115]		
Service	**# Instances**	**Total Cost**
SQS	2	$0.00
EC2	72	$0.00
SNS	15	$0.00
Total for Account: AIDAJ5CRUCDWAW7	**89**	**$0.00**
Overall Totals:	**89**	**$0.00**

Generated on: 2015/04/17 07:46:56



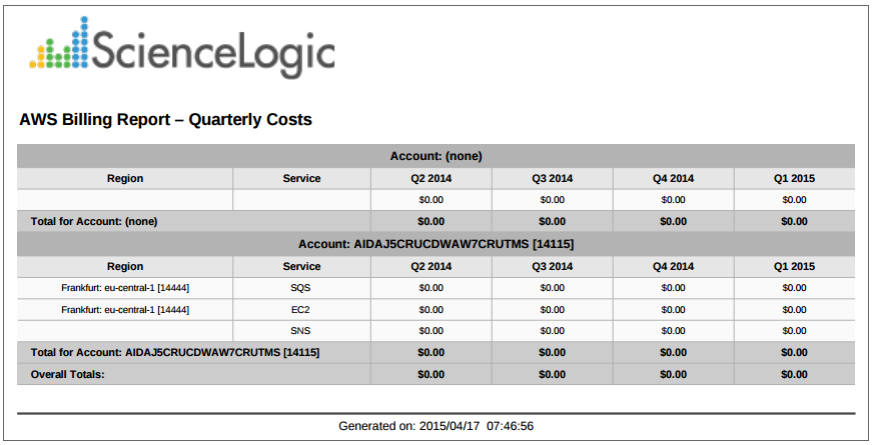
Monthly Costs

ScienceLogic

AWS Billing Report – Monthly Costs

Account: (none)														
Region	Service	Apr 2014	May 2014	Jun 2014	Jul 2014	Aug 2014	Sep 2014	Oct 2014	Nov 2014	Dec 2014	Jan 2015	Feb 2015	Mar 2015
		$0.00	$0.00	$0.00	$0.00	$0.00	$0.00	$0.00	$0.00	$0.00	$0.00	$0.00	$0.00
Total for Account: (none)		$0.00	$0.00	$0.00	$0.00	$0.00	$0.00	$0.00	$0.00	$0.00	$0.00	$0.00	$0.00
Account: AIDAJ5CRUCDWAW7CRUTMS [14115]														
Region	Service	Apr 2014	May 2014	Jun 2014	Jul 2014	Aug 2014	Sep 2014	Oct 2014	Nov 2014	Dec 2014	Jan 2015	Feb 2015	Mar 2015
Frankfurt: eucentral1 [14444]	SQS	$0.00	$0.00	$0.00	$0.00	$0.00	$0.00	$0.00	$0.00	$0.00	$0.00	$0.00	$0.00
Frankfurt: eucentral1 [14444]	EC2	$0.00	$0.00	$0.00	$0.00	$0.00	$0.00	$0.00	$0.00	$0.00	$0.00	$0.00	$0.00
	SNS	$0.00	$0.00	$0.00	$0.00	$0.00	$0.00	$0.00	$0.00	$0.00	$0.00	$0.00	$0.00
Total for Account: AIDAJ5CRUCDWAW7CRUTMS [14115]		$0.00	$0.00	$0.00	$0.00	$0.00	$0.00	$0.00	$0.00	$0.00	$0.00	$0.00	$0.00
Overall Totals:		$0.00	$0.00	$0.00	$0.00	$0.00	$0.00	$0.00	$0.00	$0.00	$0.00	$0.00	$0.00

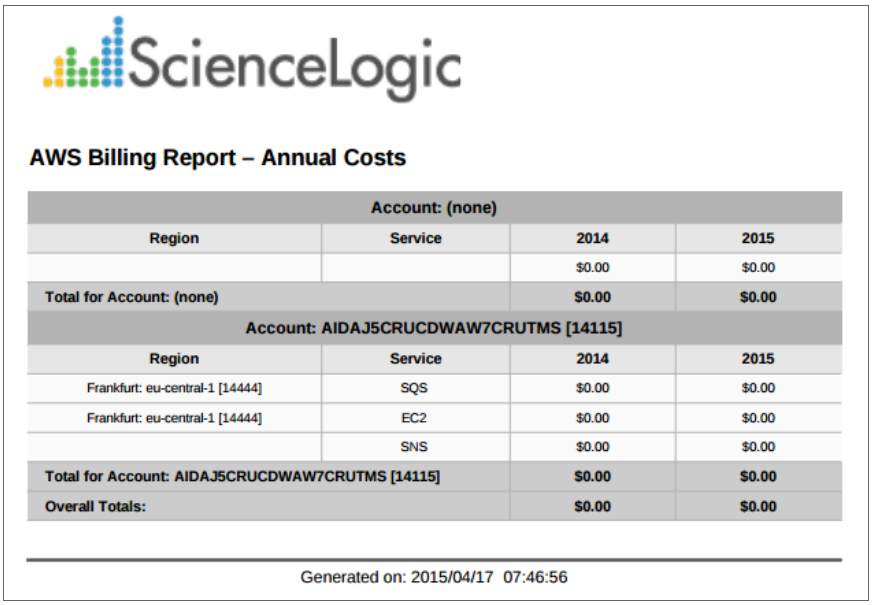
Generated on: 2015/04/17 07:46:56



AWS Billing Report – Quarterly Costs

Account: (none)					
Region	**Service**	**Q2 2014**	**Q3 2014**	**Q4 2014**	**Q1 2015**
		$0.00	$0.00	$0.00	$0.00
Total for Account: (none)		**$0.00**	**$0.00**	**$0.00**	**$0.00**
Account: AIDAJ5CRUCDWAW7CRUTMS [14115]					
Region	**Service**	**Q2 2014**	**Q3 2014**	**Q4 2014**	**Q1 2015**
Frankfurt: eu-central-1 [14444]	SQS	$0.00	$0.00	$0.00	$0.00
Frankfurt: eu-central-1 [14444]	EC2	$0.00	$0.00	$0.00	$0.00
	SNS	$0.00	$0.00	$0.00	$0.00
Total for Account: AIDAJ5CRUCDWAW7CRUTMS [14115]		**$0.00**	**$0.00**	**$0.00**	**$0.00**
Overall Totals:		**$0.00**	**$0.00**	**$0.00**	**$0.00**

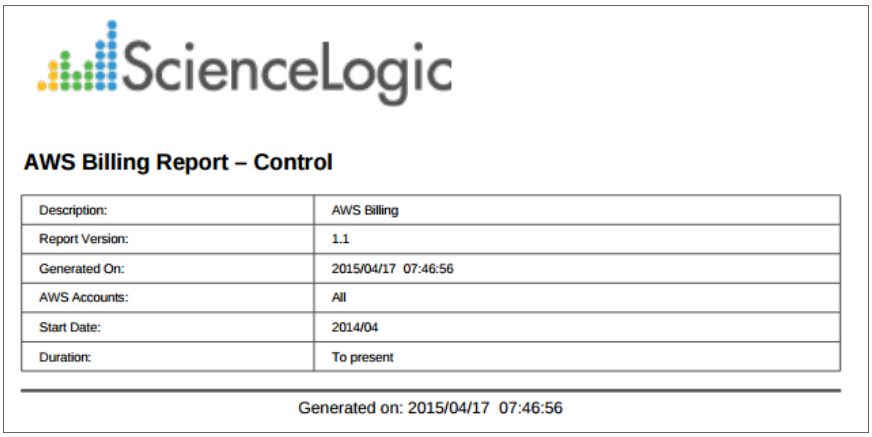
Generated on: 2015/04/17 07:46:56



AWS Billing Report – Annual Costs

Account: (none)			
Region	**Service**	**2014**	**2015**
		$0.00	$0.00
Total for Account: (none)		**$0.00**	**$0.00**
Account: AIDAJ5CRUCDWAW7CRUTMS [14115]			
Region	**Service**	**2014**	**2015**
Frankfurt: eu-central-1 [14444]	SQS	$0.00	$0.00
Frankfurt: eu-central-1 [14444]	EC2	$0.00	$0.00
	SNS	$0.00	$0.00
Total for Account: AIDAJ5CRUCDWAW7CRUTMS [14115]		**$0.00**	**$0.00**
Overall Totals:		**$0.00**	**$0.00**

Generated on: 2015/04/17 07:46:56



AWS Billing Report – Control

Description:	AWS Billing
Report Version:	1.1
Generated On:	2015/04/17 07:46:56
AWS Accounts:	All
Start Date:	2014/04
Duration:	To present

Generated on: 2015/04/17 07:46:56

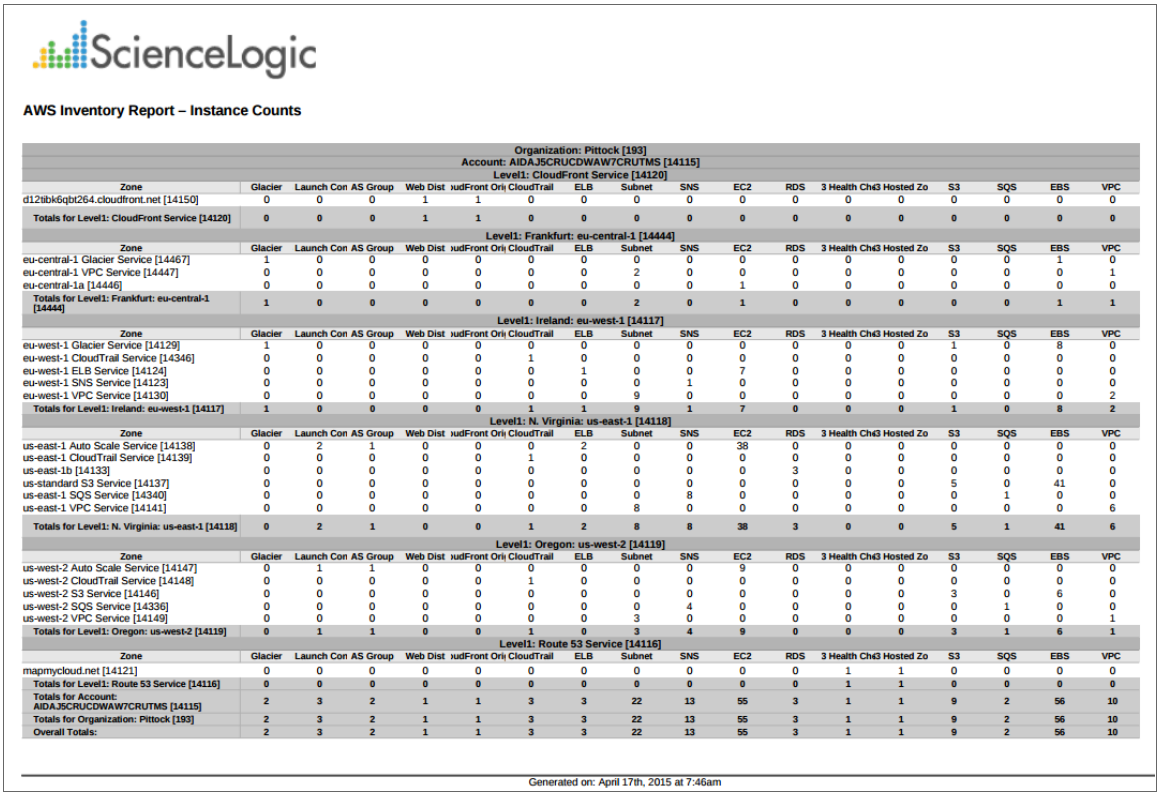
The following input options are available when generating the report (Reports > Run Report > Cloud > AWS Billing):

- ***AWS Accounts***. Select the AWS Account(s) for which you want to generate the report. The *All Accounts* checkbox is selected by default. De-selecting this checkbox allows you to select one or more specific accounts for which to generate a report.
- ***Report Span***. Select a span from one to 36 months for the report, or specify a specific starting date for the report.

This description covers the latest version of this report as shipped by ScienceLogic. This report might have been modified on your SL1 system.

AWS Inventory Report

This report displays an inventory of AWS instance counts. The report includes the number of each kind of instance in every zone associated with the chosen accounts. It also includes a count of each EC2 instance size in each zone.

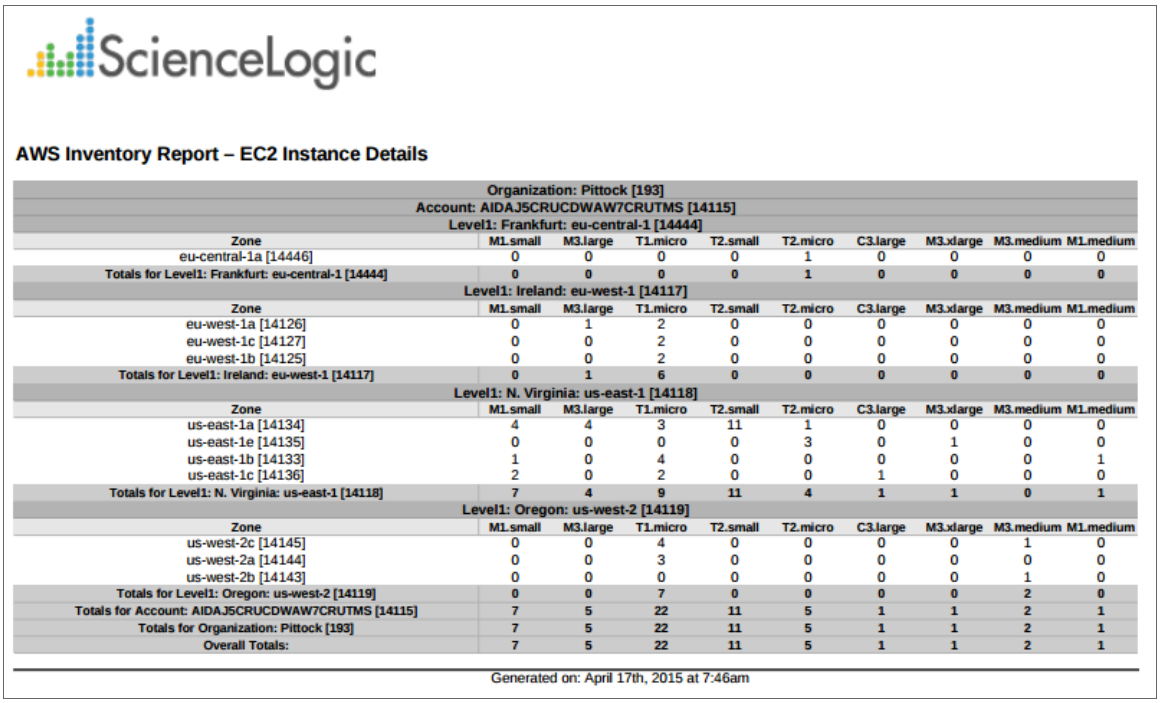


ScienceLogic

AWS Inventory Report – Instance Counts

Zone	Glacier	Launch Con	AS Group	Web Dist	udFront Ori	CloudTrail	ELB	Subnet	SNS	EC2	RDS	3 Health Ch	3 Hosted Zo	S3	SQS	EBS	VPC
Organization: Pittock [193]																	
Account: AIDAJ5CRUCDWAW7CRUTMS [14115]																	
Level1: CloudFront Service [14120]																	
d12tibk6qbt264.cloudfront.net [14150]	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
Totals for Level1: CloudFront Service [14120]	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
Level1: Frankfurt: eu-central-1 [14444]																	
Zone	**Glacier**	**Launch Con**	**AS Group**	**Web Dist**	**udFront Ori**	**CloudTrail**	**ELB**	**Subnet**	**SNS**	**EC2**	**RDS**	**3 Health Ch**	**3 Hosted Zo**	**S3**	**SQS**	**EBS**	**VPC**
eu-central-1 Glacier Service [14467]	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
eu-central-1 VPC Service [14447]	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	1
eu-central-1a [14446]	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
Totals for Level1: Frankfurt: eu-central-1 [14444]	1	0	0	0	0	0	0	2	0	1	0	0	0	0	0	1	1
Level1: Ireland: eu-west-1 [14117]																	
Zone	**Glacier**	**Launch Con**	**AS Group**	**Web Dist**	**udFront Ori**	**CloudTrail**	**ELB**	**Subnet**	**SNS**	**EC2**	**RDS**	**3 Health Ch**	**3 Hosted Zo**	**S3**	**SQS**	**EBS**	**VPC**
eu-west-1 Glacier Service [14129]	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	8	0
eu-west-1 CloudTrail Service [14346]	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
eu-west-1 ELB Service [14124]	0	0	0	0	0	0	1	0	0	7	0	0	0	0	0	0	0
eu-west-1 SNS Service [14123]	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
eu-west-1 VPC Service [14130]	0	0	0	0	0	0	0	9	0	0	0	0	0	0	0	0	2
Totals for Level1: Ireland: eu-west-1 [14117]	1	0	0	0	0	1	1	9	1	7	0	0	0	1	0	8	2
Level1: N. Virginia: us-east-1 [14118]																	
Zone	**Glacier**	**Launch Con**	**AS Group**	**Web Dist**	**udFront Ori**	**CloudTrail**	**ELB**	**Subnet**	**SNS**	**EC2**	**RDS**	**3 Health Ch**	**3 Hosted Zo**	**S3**	**SQS**	**EBS**	**VPC**
us-east-1 Auto Scale Service [14138]	0	2	1	0	0	0	2	0	0	38	0	0	0	0	0	0	0
us-east-1 CloudTrail Service [14139]	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
us-east-1b [14133]	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0
us-standard S3 Service [14137]	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	41	0
us-east-1 SQS Service [14340]	0	0	0	0	0	0	0	0	8	0	0	0	0	0	1	0	0
us-east-1 VPC Service [14141]	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0	6
Totals for Level1: N. Virginia: us-east-1 [14118]	0	2	1	0	0	1	2	8	8	38	3	0	0	5	1	41	6
Level1: Oregon: us-west-2 [14119]																	
Zone	**Glacier**	**Launch Con**	**AS Group**	**Web Dist**	**udFront Ori**	**CloudTrail**	**ELB**	**Subnet**	**SNS**	**EC2**	**RDS**	**3 Health Ch**	**3 Hosted Zo**	**S3**	**SQS**	**EBS**	**VPC**
us-west-2 Auto Scale Service [14147]	0	1	1	0	0	0	0	0	0	9	0	0	0	0	0	0	0
us-west-2 CloudTrail Service [14148]	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
us-west-2 S3 Service [14146]	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	6	0
us-west-2 SQS Service [14336]	0	0	0	0	0	0	0	0	4	0	0	0	0	0	1	0	0
us-west-2 VPC Service [14149]	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	1
Totals for Level1: Oregon: us-west-2 [14119]	0	1	1	0	0	1	0	3	4	9	0	0	0	3	1	6	1
Level1: Route 53 Service [14116]																	
Zone	**Glacier**	**Launch Con**	**AS Group**	**Web Dist**	**udFront Ori**	**CloudTrail**	**ELB**	**Subnet**	**SNS**	**EC2**	**RDS**	**3 Health Ch**	**3 Hosted Zo**	**S3**	**SQS**	**EBS**	**VPC**
mapmycloud.net [14121]	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
Totals for Level1: Route 53 Service [14116]	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
Totals for Account: AIDAJ5CRUCDWAW7CRUTMS [14115]	2	3	2	1	1	3	3	22	13	55	3	1	1	9	2	56	10
Totals for Organization: Pittock [193]	2	3	2	1	1	3	3	22	13	55	3	1	1	9	2	56	10
Overall Totals:	2	3	2	1	1	3	3	22	13	55	3	1	1	9	2	56	10

Generated on: April 17th, 2015 at 7:46am



AWS Inventory Report – EC2 Instance Details

Zone	M1.small	M3.large	T1.micro	T2.small	T2.micro	C3.large	M3.xlarge	M3.medium	M1.medium
Organization: Pittock [193]									
Account: AIDAJ5CRUCDWAW7CRUTMS [14115]									
Level1: Frankfurt: eu-central-1 [14444]									
eu-central-1a [14446]	0	0	0	0	1	0	0	0	0
Totals for Level1: Frankfurt: eu-central-1 [14444]	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**
Level1: Ireland: eu-west-1 [14117]									
eu-west-1a [14126]	0	1	2	0	0	0	0	0	0
eu-west-1c [14127]	0	0	2	0	0	0	0	0	0
eu-west-1b [14125]	0	0	2	0	0	0	0	0	0
Totals for Level1: Ireland: eu-west-1 [14117]	**0**	**1**	**6**	**0**	**0**	**0**	**0**	**0**	**0**
Level1: N. Virginia: us-east-1 [14118]									
us-east-1a [14134]	4	4	3	11	1	0	0	0	0
us-east-1e [14135]	0	0	0	0	3	0	1	0	0
us-east-1b [14133]	1	0	4	0	0	0	0	0	1
us-east-1c [14136]	2	0	2	0	0	1	0	0	0
Totals for Level1: N. Virginia: us-east-1 [14118]	**7**	**4**	**9**	**11**	**4**	**1**	**1**	**0**	**1**
Level1: Oregon: us-west-2 [14119]									
us-west-2c [14145]	0	0	4	0	0	0	0	1	0
us-west-2a [14144]	0	0	3	0	0	0	0	0	0
us-west-2b [14143]	0	0	0	0	0	0	0	1	0
Totals for Level1: Oregon: us-west-2 [14119]	**0**	**0**	**7**	**0**	**0**	**0**	**0**	**2**	**0**
Totals for Account: AIDAJ5CRUCDWAW7CRUTMS [14115]	**7**	**5**	**22**	**11**	**5**	**1**	**1**	**2**	**1**
Totals for Organization: Pittock [193]	**7**	**5**	**22**	**11**	**5**	**1**	**1**	**2**	**1**
Overall Totals:	**7**	**5**	**22**	**11**	**5**	**1**	**1**	**2**	**1**

Generated on: April 17th, 2015 at 7:46am

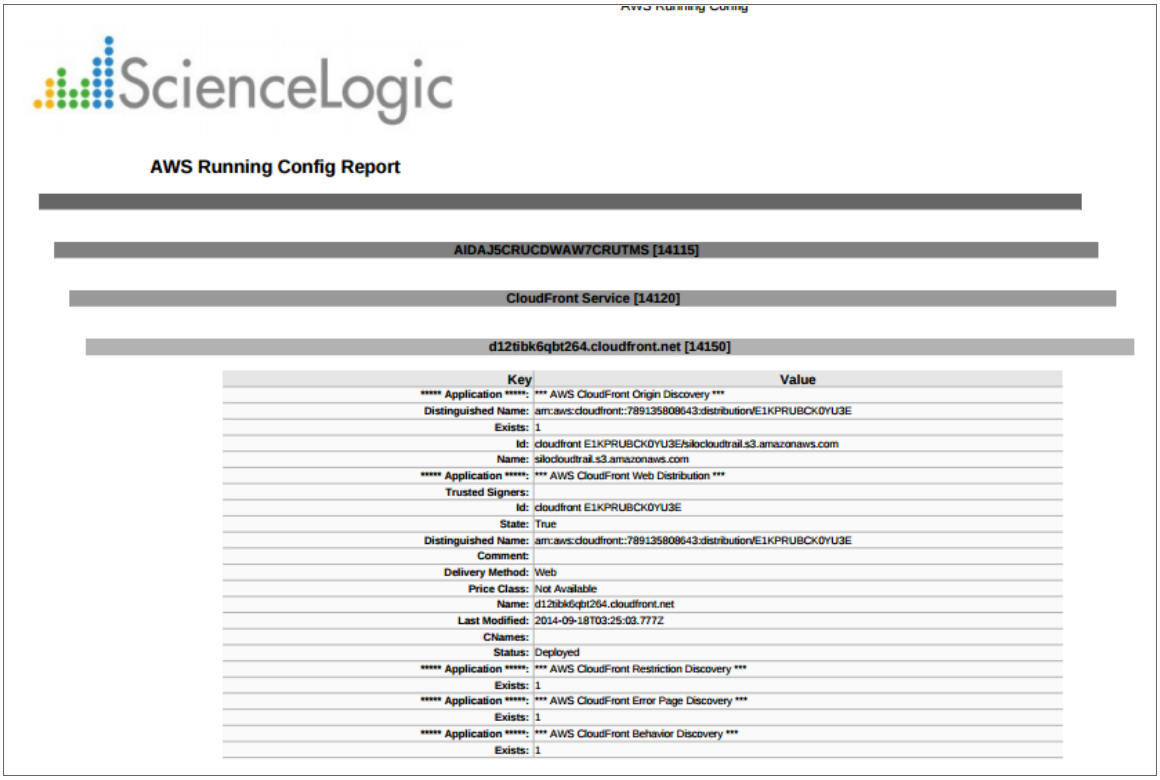
The following input options are available when generating the report (Reports > Run Report > Others > AWS Inventory):

- ***Organizations***. Select the organization for which you want to generate the report. The *All Organizations* checkbox is selected by default. De-selecting this checkbox allows you to select one or more specific organizations for which to generate a report.
- ***AWS Accounts***. Select the AWS Account(s) for which you want to generate the report. The *All Accounts* checkbox is selected by default. De-selecting this checkbox allows you to select one or more specific accounts for which to generate a report.
- ***Filter on EC2 Instance Config Data***. Select the EC2 instances that will be included in the report based on the configuration data reported for each EC2 instance:
  - Choose up to four configuration parameters for EC2 instances.
  - For each selected configuration parameter, enter a value to match against and select how that value should be matched.
  - In the ***Comparison Operator*** field, select whether an EC2 instance must match all configuration parameters (*and*) or only one configuration parameter (*or*) to be included on the report.
- ***Report Options***. Select the *Include Terminated Instances* checkbox to include all terminated instances.

This description covers the latest version of this report as shipped by ScienceLogic. This report might have been modified on your SL1 system.

AWS Running Config Report

This report displays the running config of all AWS instances for one to all organizations across a number of AWS billing accounts.



AWS Running Config Report

AIDAJ5CRUCDWAW7CRUTMS [14115]

CloudFront Service [14120]

d12tibk6qbt264.cloudfront.net [14150]

Key	Value
***** Application *****:	*** AWS CloudFront Origin Discovery ***
Distinguished Name:	arn:aws:cloudfront::789135808643:distribution/E1KPRUBCK0YU3E
Exists:	1
Id:	cloudfront E1KPRUBCK0YU3E/silocloudtrail.s3.amazonaws.com
Name:	silocloudtrail.s3.amazonaws.com
***** Application *****:	*** AWS CloudFront Web Distribution ***
Trusted Signers:	
Id:	cloudfront E1KPRUBCK0YU3E
State:	True
Distinguished Name:	arn:aws:cloudfront::789135808643:distribution/E1KPRUBCK0YU3E
Comment:	
Delivery Method:	Web
Price Class:	Not Available
Name:	d12tibk6qbt264.cloudfront.net
Last Modified:	2014-09-18T03:25:03.777Z
CNames:	
Status:	Deployed
***** Application *****:	*** AWS CloudFront Restriction Discovery ***
Exists:	1
***** Application *****:	*** AWS CloudFront Error Page Discovery ***
Exists:	1
***** Application *****:	*** AWS CloudFront Behavior Discovery ***
Exists:	1

The following input options are available when generating the report (Reports > Run Report > Others > AWS Running Config):

- ***Organizations***. Select one, multiple, or all organizations to include in the report.
    - *All Organizations*. This checkbox is selected by default. De-selecting this checkbox allows you to select one or more specific organizations for the report.
    - *Organizations*. If you unchecked the ***All Organizations*** checkbox, select one or more organizations to include in the report.
- ***AWS Accounts***. Select one, multiple, or all AWS Accounts to include in the report.
    - *All Accounts*. This checkbox is selected by default. De-selecting this checkbox allows you to select one or more specific AWS accounts for the report.
    - *Accounts*. If you unchecked the ***All Accounts*** checkbox, select one or more AWS Accounts to include in the report.

- ***Filter on EC2 Instance Config Data***. Select the EC2 instances that will be included on the report based on the configuration data reported for each EC2 instance:
    - Choose up to four configuration parameters for EC2 instances.
    - For each selected configuration parameter, enter a value to match against and select how that value should be matched.
    - In the ***Comparison Operator*** field, select whether an EC2 instance must match all configuration parameters (*and*) or only one configuration parameter (*or*) to be included in the report.
- ***Report Options***. Select the *Include Terminated Instances* checkbox to include all terminated instances.

This description covers the latest version of this report as shipped by ScienceLogic. This report might have been modified on your SL1 system.

Chapter 9

Dashboards

Overview

The following sections describe how to install the *Amazon Web Services: Dashboards* PowerPack and a description of each dashboard that is included in the PowerPack:

Installing the Amazon Web Services: Dashboards PowerPack

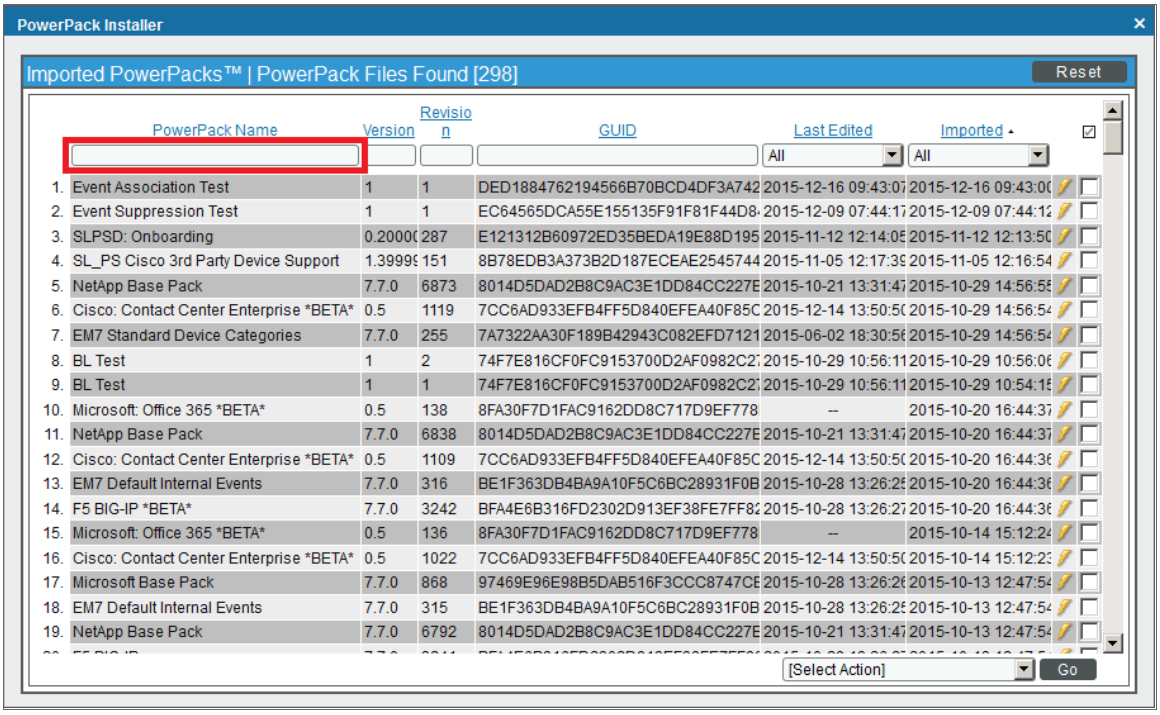
To view the Amazon Web Services dashboards in SL1, you must first install the *Amazon Web Services: Dashboards* PowerPack.

> **NOTE:** The AWS dashboards have a default ***Access Control*** setting of "Private", which means they can be viewed only by an administrator. For more information about dashboard access settings, see the ***Dashboards*** manual.

To install the PowerPack:

1. Go to the **PowerPack Manager** page (System > Manage > PowerPacks).
2. Click the **[Actions]** button, then select *Install PowerPack*. The **Imported PowerPacks** modal page appears.

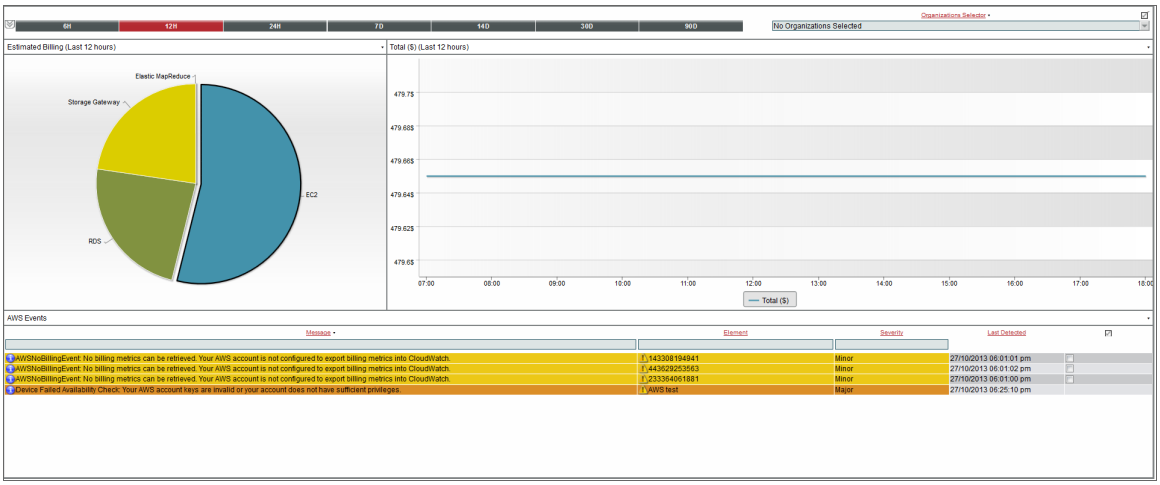
3. Use the search filter in the ***PowerPack Name*** column heading to locate the PowerPack you want to install. To do so, enter text to match, including special characters, and the **Imported PowerPacks** modal page displays only PowerPacks that have a matching name.



4. Click the lightning-bolt icon ( ) for the PowerPack that you want to install.
5. The **Install PowerPack** modal page appears. To install the PowerPack, click **[Install]**.
6. The PowerPack now appears in the **PowerPack Manager** page. The contents of the PowerPack are automatically installed in your SL1 System.

AWS Account Billing Dashboard

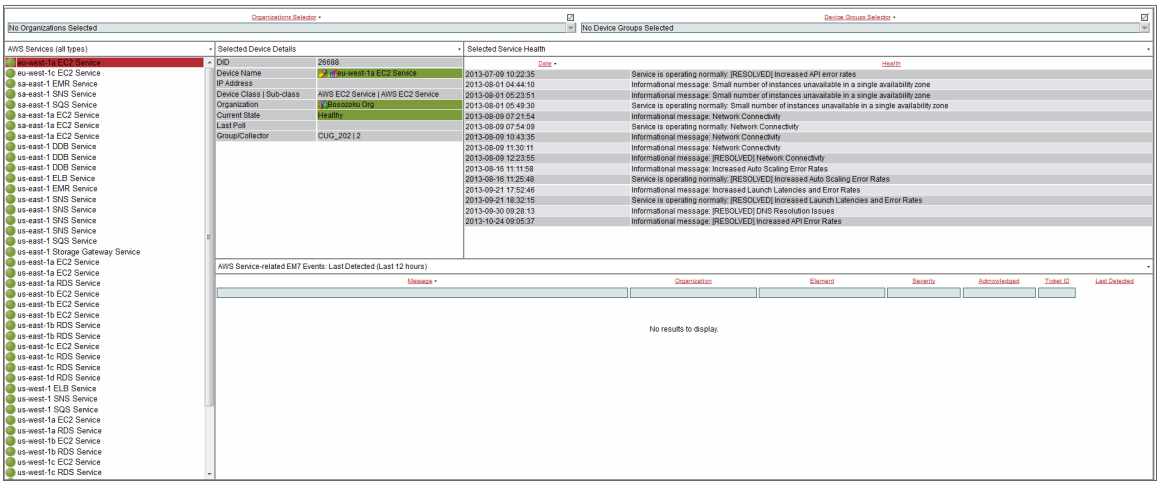
The AWS Account Billing Dashboard displays:



- A pie chart that shows the estimated billing amount for each service over the selected time period.
- A performance graph that shows the estimated billing amount for the selected service, over time. To select a service, click on the pie-chart segment for that service.
- A table that shows the currently active AWS events.
- A time span selector that controls the amount of data shown in the pie chart and the performance graph.
- An organization selector that limits the data in the pie chart and performance graph to include only instances associated with the selected organizations.

AWS Health Status Dashboard

The AWS Health Status Dashboard displays:



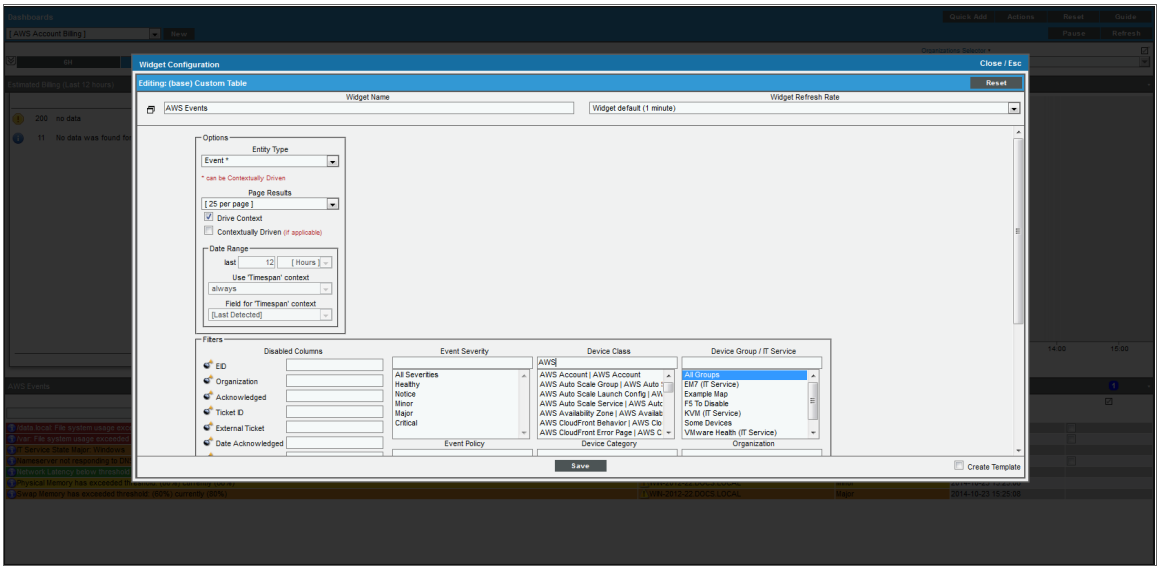
- A traffic light widget that displays a list of AWS services. To populate the other widgets in this dashboard, select a service.

- A tear-sheet widget that displays information and links for the selected service.
- A service health widget, that displays log messages about the health of the service.
- A table that displays currently active events for the service.
- An organization selector and a device group selector that control which services are shown in the traffic light widget.

Configuring the AWS Dashboards

The AWS Account Billing and AWS Health Status dashboards must have their (base) Custom Table widgets manually configured to filter only AWS service-specific events. To do this:

1. Go to Dashboards > Classic Dashboards and select AWS Account Billing, or in the SL1 classic user interface go to Dashboards and select AWS Account Billing.
2. Click the down-arrow in the upper-right of the AWS Events widget, and then select *Configure* from the **Options** menu. The **Widget Configuration** modal page appears.
3. In the ***Device Class*** filter, enter "AWS" to show only AWS device classes:



4. Control-click on the following items in the ***Device Class*** field:

   - AWS DDB Service
   - AWS EC2 Service
   - AWS ELB Service
   - AWS EMR Service
   - AWS RDS Service
   - AWS SNS Service

- AWS SQS Service
- AWS Storage Gateway Service

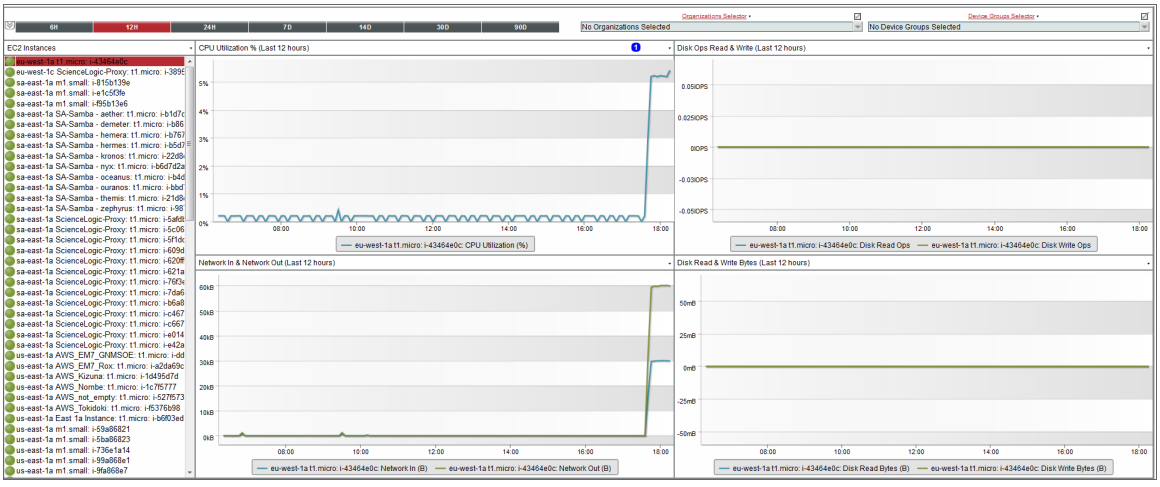
5. Click the **[Save]** button.
6. Repeat steps 1 - 5 for the AWS Health Status dashboard.

AWS Service Instance Performance Dashboards

The *Amazon Web Services: Dashboards* PowerPack includes a dashboard for each service type. Each dashboard displays performance metrics for instances of an AWS service. The following dashboards are included:

- AWS Application ELB Performance
- AWS Classic ELB Performance
- AWS DDB Performance
- AWS EBS Performance
- AWS EC2 Performance
- AWS EMR Performance
- AWS Network ELB Performance
- AWS RDS Performance
- AWS SQS Performance
- AWS Storage Gateway Performance

Each performance dashboard includes:



- A traffic light widget that shows the status of all instances for the service.
- Four performance graphs that show applicable metrics when you select an instance from the traffic light widget.

- A time span selector that controls the amount of data shown in the performance graphs.
- An organization selector and device group selector that control which instances are shown in the traffic light widget.

Chapter 10

Run Book Actions and Automations

Overview

The following sections describe the Run Book Action and Automation policies that are included in the *Amazon Web Services* PowerPack and how to use them:

About the Run Book Actions and Automations

The *Amazon Web Services* PowerPack includes Run Book Action and Automation policies that can be used to:

- Automatically disable EC2 and EBS devices based on EC2 tags collected from AWS
- Automatically create and start a discovery session for the public or private IP address of an EC2 instance after a component and physical device are merged
- Automatically move an EC2 instance to a vanished state if the EC2 instance is in a terminating or terminated state
- Align AWS region device classes with the correct AWS Region

The following table describes the automation policies and what they do:

Policy Name	Result
AWS: Account Creation	SL1 creates a virtual device for an AWS account.
AWS: Disable EBS Instances by EC2 Tag	If a component device belongs to the AWS EBS Volumes device group and has an EC2 tag, SL1 disables the device.
AWS: Disable EC2 and EBS Instances by EC2 Tag	If a component device belongs to either the AWS EBS Volumes or AWS EC2 Instances device group and has an EC2 tag, SL1 disables the device.
AWS: Disable or Discover EC2 Instances	SL1 automatically discovers EC2 instances by public or private IP address. Additionally, if a component device belongs to the AWS EC2 Instances device group and has an EC2 tag, SL1 disables the device.
AWS: Discover EC2 Instances	SL1 automatically discovers EC2 instances by public or private IP address.
AWS: EKS Cluster Creation	SL1 automatically discovers EKS Clusters when an AWS EKS Cluster is configured.
AWS: Merge with EC2	If SL1 determines that the IP address of a physical device matches a custom attribute added to an EC2 Instance component device, SL1 merges the devices.
AWS: Organization Creation	SL1 creates a virtual device for an AWS organization.
AWS: RDS DB Instance Device Class Alignment	SL1 aligns the correct RDS device class the RDS Instance.
AWS: Region Device Class Alignment	If a Region is aligned to an incorrect Region device class, SL1 will align the Region to the correct device class.

Policy Name	Result
AWS: Vanish Terminated EC2 Instances	If a device belongs to the AWS EC2 Instances device group and is in a terminated or terminating state, SL1 un-merges the EC2 Instance and physical device (if applicable), clears the device's associated events, and then moves the device to a vanished state.

NOTE: The automation policies in the *Amazon Web Services* PowerPack are disabled by default. To use these automations, you must enable the automation policies and optionally modify the parameters in the automation actions.

NOTE: To use the automation policies in the *Amazon Web Services* PowerPack, the AWS EBS Volumes and AWS EC2 Instances device groups must already be created and populated.

Disabling EC2 and EBS Instances by EC2 Tag

The automation described in this section disables EC2 and EBS devices based on EC2 tags. This can be set up in the "AWS: Disable Instance by Tag" Run Book Automation, so if an EBS or EC2 instance has the tag(s) you specify, SL1 will disable the device.

The automation for disabling EC2 and EBS instances includes two automation actions that are executed in the following order:

- ***AWS: Get EC2 Instance Configuration***. This action requests information from the AWS API about the EC2 instance that triggered the automation action or the EC2 instance associated with the EBS instance that triggered the automation action. Information about the EC2 instance associated with an EBS instance is returned only if one EC2 instance is associated with the EBS instance.
- ***AWS: Disable Instance By Tag***. This action compares the information collected by the ***AWS: Get EC2 Instance Configuration*** automation action with a pre-defined list of key/value pairs. If an AWS tag matches a key/value pair, the triggering device is disabled.

The *Amazon Web Services* PowerPack includes three automation policies that trigger these actions:

- ***AWS: Disable EC2 and EBS Instances by EC2 Tag***. If enabled, this automation policy can trigger for any device with which the "AWS: EC2 Instance Configuration" or the "AWS: EBS Instance Configuration" Dynamic Applications are aligned (the members of the AWS EC2 Instances and AWS EBS Volumes device groups). The automation policy triggers when the "Component Device Record Created" event is active on the matching devices, immediately after the devices are discovered in the system. Enable this automation policy if you want to disable EC2 and EBS instances by EC2 tag, but do not want to enable automated discovery of EC2 instances by public or private IP address.

- ***AWS: Disable or Discover EC2 Instances***. If enabled, this automation policy can trigger for any device with which the "AWS: EC2 Instance Configuration" Dynamic Application is aligned (the members of the AWS EC2 Instances). The automation policy triggers when the "Component Device Record Created" event is active on the matching devices, immediately after the devices are discovered in the system. Enable this automation policy if you want to disable EC2 instances by EC2 tag *and* want to enable automated discovery of EC2 instances by public or private IP address. This automation policy is configured to run both processes in the correct order for EC2 instances. If you enable this automation policy and want to automatically disable associated EBS instances, you must also enable the ***AWS: Disable EBS Instances by EC2 Tag*** automation policy.
- ***AWS: Disable EBS Instances by EC2 Tag***. If enabled, this automation policy can trigger for any device with which the "AWS: EC2 Instance Configuration" Dynamic Application is aligned (the members of the AWS EC2 Instances). The automation policy triggers when the "Component Device Record Created" event is active on the matching devices, immediately after the devices are discovered in the system. Enable this automation policy if you want to disable EC2 instances by EC2 tag, want to enable automated discovery of EC2 instances by public or private IP address, and want to disable EBS instances by EC2 tag.

To use this automation, you must:

- *Modify the parameters of the automation actions (optional)*
- *Enable the Component Device Record Created event policy*
- *Enable the automation policies*
- *Configure your system to preserve these changes*

Modifying the Parameters of the Automation Actions

The snippet for the ***AWS: Disable Instance by Tag*** automation action includes the pre-defined list of key/value pairs with which the tags collected from the AWS API are compared. You must modify this list to include the key/value pairs that you want to use to disable EC2 instances.

To modify the parameters for the ***AWS: Disable Instance by Tag*** automation action:

1. Go to the **Action Policy Manager** page (Registry > Run Book > Actions).
2. Click the wrench icon () for the ***AWS: Disable Instance By Tag*** automation action.
3. In the ***Snippet Code*** field, locate and edit the following line:

   ```
   DISABLE_TAGS = [('ExampleKey','ExampleValue')]
   ```

   The line must be in the following format, with each key and each value inside single-quotes and each key/value pair comma-separated inside parentheses, with commas separating each key/value pair.

   ```
   DISABLE_TAGS = [('Key','Value'),('Key','Value'),...,('Key','Value')]
   ```

   For example, suppose you want to disable an EC2 instance where the "Environment" key is either "dev" or "test" or the "Owner" key is "Sales". You would update the line so it looks like this:

   ```
   DISABLE_TAGS = [('Environment','dev'),('Environment','test'),('Owner','Sales')]
   ```
4. Click the **[Save]** button.

Enabling the Component Device Record Created Event Policy

To enable the "Component Device Record Created" event policy:

1. Go to the **Event Policies** page (Events > Event Policies).
2. Click the Actions menu ( ) for the "Component Device Record Created" event policy and select *Edit*.
3. In the **Event Policy Editor** page, click on the ***Enable Event Policy*** toggle to enable the event policy.
4. Click **[Save]**.

To enable the "Component Device Record Created" event policy in the SL1 classic user interface:

1. Go to the **Event Policy Manager** page (Registry > Events > Event Manager).
2. Click the wrench icon ( ) for the "Component Device Record Created" event policy.
3. In the ***Operational State*** field, select *Enabled*.
4. Click **[Save]**.

To prevent this change from being overwritten when the PowerPacks installed on the system are updated, you can enable the ***Selective PowerPack Field Protection*** option. To enable this option:

1. Go to the **Behavior Settings** page (System > Settings > Behavior).
2. Check the ***Enable Selective PowerPack Field Protection*** checkbox.
3. Click **[Save]**.

Enabling the Automation Policies

To enable one or more automation policies in the *Amazon Web Services* PowerPack:

1. Go to the **Automation Policy Manager** page (Registry > Run Book > Automation).
2. Click the wrench icon ( ) for the automation policy you want to enable.
3. In the ***Policy State*** field, select *Enabled*.
4. Click **[Save]**.

Preserving Automation Changes

If you have modified automation actions and policies that are included in the *Amazon Web Services* PowerPack, those changes will be overwritten when the PowerPack is updated in your system. If you have modified automation actions and policies that are included in the PowerPack, you can:

- Re-implement those changes after each update of the *Amazon Web Services* PowerPack.
- Remove the content from the PowerPack on your system before you update it. When the *Amazon Web Services* PowerPack is updated in your system, updated versions of this content will not be installed on your system and your local changes will be preserved.

To remove automation actions or automation policies content from the *Amazon Web Services* PowerPack on your system:

1. Go to the **PowerPack Manager** page (System > Manage > PowerPacks).
2. Click the wrench icon () for the *Amazon Web Services* PowerPack. The **Editing PowerPack** page appears.
3. In the left NavBar of the **Editing PowerPack** page, select the type of content you want to remove:
   - To remove an automation action, click **Run Book Actions**. The **Embedded Run Book Actions** and **Available Run Book Actions** panes appear.
   - To remove an automation policy, click **Run Book Policies**. The **Embedded Run Book Policies** and **Available Run Book Policies** panes appear.
4. In the upper pane, click the bomb icon () for each automation action or automation policy that you want to remove from the *Amazon Web Services* PowerPack on your system.

Discovering EC2 Instances by Public or Private IP Address

The automation in this section automatically creates and starts a discovery session for the public or private IP address of an EC2 instance after a component and physical device are merged. If SL1 determines that the IP address of a physical device matches a custom attribute added to an EC2 instance component device, SL1 merges the devices.

The automation for discovering EC2 instances by public or private IP addresses includes three automation actions that are executed in the following order:

- ***AWS: Get EC2 Instance Configuration***. This action requests information from the AWS API about the EC2 instance that triggered the automation action.
- ***AWS: Discover from EC2 IP***. This action uses the IP address and port information in the response from the AWS API to create and run a discovery session. This action also adds a custom attribute to the EC2 component device record that can be used to match a newly discovered device to the EC2 instance.
- ***AWS: Merge Physical with Component***. This action matches the IP address of a physical device with the custom attribute added to EC2 component devices by the ***AWS: Discover from EC2 IP*** automation action. If a match is found, the matching EC2 component device is merged with the physical device.

The *Amazon Web Services* PowerPack includes three automation policies that trigger these actions:

- ***AWS: Discover EC2 Instances***. If enabled, this automation policy can trigger for any device with which the "AWS: EC2 Instance Configuration" Dynamic Application is aligned (the members of the AWS EC2 Instances). The automation policy triggers when the "Component Device Record Created" event is active on the matching devices, immediately after the devices are discovered in the system. Enable this automation policy if you want to enable automated discovery of EC2 instances by public or private IP address but do not want disable EC2 and EBS instances by EC2 tag.

- ***AWS: Disable or Discover EC2 Instances***. If enabled, this automation policy can trigger for any device with which the "AWS: EC2 Instance Configuration" Dynamic Application is aligned (the members of the AWS EC2 Instances). The automation policy triggers when the "Component Device Record Created" event is active on the matching devices, immediately after the devices are discovered in the system. Enable this automation policy if you want to disable EC2 instances by EC2 tag **and** want to enable automated discovery of EC2 instances by public or private IP address. This automation policy is configured to run both in the correct order for EC2 instances.
- ***AWS: Merge with EC2***. If enabled, this automation policy can trigger for any device. The automation policy triggers when the "Device Record Created" event is active on the matching devices, immediately after the devices are discovered in the system. Enable this automation policy if you want to enable automated discovery of EC2 instances by public or private IP address.

To use this automation, you must:

- *Modify the parameters of the automation actions (optional)*
- *Enable the Component Device Record Created event policy*
- *Enable the Device Record Created event policy*
- *Enable the automation policies*
- *Configure your system to preserve these changes*

Modifying the Parameters of the Automation Actions

The snippet for the ***AWS: Discover from EC2 IP*** automation action includes parameters that define how the automation action creates discovery sessions. You can edit the following lines in the ***Snippet Code*** field of the ***AWS: Discover from EC2 IP*** automation action to change these parameters:

- `EC2_IP_ATTRIBUTE = 'PrivateIpAddress'`

  The attribute returned by the AWS API for EC2 instances that contains the IP address to use in the discovery session. By default, the private IP address is used. To use the public IP address of the EC2 instance, change this line to:

  ```
  EC2_IP_ATTRIBUTE = 'PublicIpAddress'
  ```

- `EXTRA_SCAN_PORTS = ["21","22","23","25","80","443","5985","5986"]`

  The list of TCP ports used in the discovery session includes any TCP ports that are specified explicitly in the security group associated with the EC2 instance, plus any TCP ports included in the EXTRA_SCAN_PORTS parameter. You can add or remove ports from this default list. For example, if you wanted to remove TCP port 21 from this list and add TCP port 53, you would change this line to:

  ```
  EXTRA_SCAN_PORTS = ["22","23","25","53","80","443","5985","5986"]
  ```

  NOTE: The EXTRA_SCAN_PORTS parameter must be populated if there are no rules for specific ports in the security group associated with the EC2 instance.

- `AUTO_INCLUDE_CREDS = True`

If the AUTO_INCLUDE_CREDS parameter is "True", the automation will automatically add credentials to the discovery session. A credential will be added automatically if it meets one of the following requirements:

- The credential is an SNMP credential, the Security Group associated with the EC2 instance includes a rule that allows access to UDP port 161, and the credential is explicitly aligned within the organization of the EC2 instance.
- The credential is an SNMP credential, the Security Group associated with the EC2 instance includes a rule that allows access to UDP port 161, the credential is associated with all organizations in the system, and the INCLUDE_ALL_ORG_CREDS parameter is "True".
- The credential is not an SNMP credential or an LDAP/AD credential, the TCP port used by the credential is included in the list of TCP ports for the discovery session (the credential is specified explicitly in the security group associated with the EC2 instance or is included in the EXTRA_SCAN_PORTS parameter), and the credential is explicitly aligned with in the organization of the EC2 instance.
- The credential is not an SNMP credential or an LDAP/AD credential, the TCP port used by the credential is included in the list of TCP ports for the discovery session (the credential is specified explicitly in the security group associated with the EC2 instance or is included in the EXTRA_SCAN_PORTS parameter), and the INCLUDE_ALL_ORG_CREDS parameter is "True".

To disable the automatic alignment of credentials to the discovery session, change this line to:

```
AUTO_INCLUDE_CREDS = False
```

- `INCLUDE_ALL_ORG_CREDS = True`

  If INCLUDE_ALL_ORG_CREDS is "True" and the AUTO_INCLUDE_CREDS parameter is "True", credentials that are aligned with all organizations (credentials that do not have an explicit organization alignment) are automatically included in the discovery session when that credential meets the other requirements for being automatically included in the discovery session.

- `EXTRA_CREDS = ""`

  In addition to the credentials that are automatically included in the discovery sessions based on open ports, you can optionally specify a string of comma-separated credential IDs for credentials that will be included in every discovery session created by this automation. For example, if you wanted to include credentials with IDs 10 and 13 in every discovery session created by this automation, you would change this line to:

  ```
  EXTRA_CREDS = "10,13"
  ```

- `DISCOVER_NON_SNMP = "1"`

  If DISCOVER_NON_SNMP is set to "1", discovery sessions created by this automation will be configured to discover non-SNMP devices. If you want the discovery sessions created by this automation to discover only SNMP devices, change this line to:

  ```
  DISCOVER_NON_SNMP = "0"
  ```

- `TEMPLATE_NAME = ""`

If you specify a device template name in the TEMPLATE_NAME parameter, that device template will be automatically aligned with all discovery sessions created by this automation. For example, if you wanted to align a device template called "Standard Device Template" to every discovery session created by this automation, you would change this line to:

```
TEMPLATE_NAME = "Standard Device Template"
```

To modify the parameters for the ***AWS: Discover from EC2 IP*** automation action, perform the following steps:

1. Go to the **Action Policy Manager** page (Registry > Run Book > Actions).
2. Click the wrench icon () for the ***AWS: Discover from EC2 IP*** automation action.
3. In the ***Snippet Code*** field, locate and edit the line(s) for the parameter(s) you want to change:
4. Click the **[Save]** button.

If you modified the EC2_IP_ATTRIBUTE parameter in the ***AWS: Discover from EC2 IP*** automation action, you must perform the following steps to update the ***AWS: Merge Physical with Component*** automation action:

To modify the parameters for the ***AWS: Discover from EC2 IP*** automation action, perform the following steps:

1. Go to the **Action Policy Manager** page (Registry > Run Book > Actions).
2. Click the wrench icon () for the ***AWS: Discover from EC2 IP*** automation action.
3. In the ***Snippet Code*** field, locate and edit the following line:

   ```
   IP_ATTRIBUTE = 'c-EC2_PrivateIpAddress'
   ```

   If you changed the EC2_IP_ATTRIBUTE parameter in the ***AWS: Discover from EC2 IP*** automation action to 'PublicIpAddress', change this line to:

   ```
   IP_ATTRIBUTE = 'c-EC2_PublicIpAddress'
   ```
4. Click the **[Save]** button.

Enabling the Component Device Record Created Event Policy

To enable the "Component Device Record Created" event policy:

1. Go to the **Event Policies** page (Events > Event Policies).
2. Click the Actions menu (•••) for the "Component Device Record Created" event policy and select *Edit*.
3. In the **Event Policy Editor** page, click on the ***Enable Event Policy*** toggle to enable the event policy.
4. Click **[Save]**.

To enable the "Component Device Record Created" event policy in the SL1 classic user interface:

1. Go to the **Event Policy Manager** page (Registry > Events > Event Manager).
2. Click the wrench icon () for the "Component Device Record Created" event policy.

3. In the ***Operational State*** field, select *Enabled*.
4. Click **[Save]**.

To prevent this change from being overwritten when the PowerPacks installed on the system are updated, you can enable the ***Selective PowerPack Field Protection*** option. To enable this option:

1. Go to the **Behavior Settings** page (System > Settings > Behavior).
2. Check the ***Enable Selective PowerPack Field Protection*** checkbox.
3. Click **[Save]**.

Enabling the Device Record Created Event Policy

To enable the "Device Record Created" event policy:

1. Go to the **Event Policies** page (Events > Event Policies).
2. Click the Actions menu ( ) for the "Device Record Created" event policy and select *Edit*.
3. In the **Event Policy Editor** page, click on the ***Enable Event Policy*** toggle to enable the event policy.
4. Click **[Save]**.

To enable the "Device Record Created" event policy in the SL1 classic user interface:

1. Go to the **Event Policy Manager** page (Registry > Events > Event Manager).
2. Click the wrench icon ( ) for the "Device Record Created" event policy.
3. In the ***Operational State*** field, select *Enabled*.
4. Click **[Save]**.

To prevent this change from being overwritten when the PowerPacks installed on the system are updated, you can enable the ***Selective PowerPack Field Protection*** option. To enable this option:

1. Go to the **Behavior Settings** page (System > Settings > Behavior).
2. Check the ***Enable Selective PowerPack Field Protection*** checkbox.
3. Click **[Save]**.

Enabling the Automation Policies

To enable one or more automation policies in the *Amazon Web Services* PowerPack:

1. Go to the **Automation Policy Manager** page (Registry > Run Book > Automation).
2. Click the wrench icon ( ) for the automation policy you want to enable.
3. In the ***Policy State*** field, select *Enabled*.
4. Click **[Save]**.

Preserving Automation Changes

If you have modified automation actions and policies that are included in the *Amazon Web Services* PowerPack, those changes will be overwritten when the PowerPack is updated in your system. If you have modified automation actions and policies that are included in the PowerPack, you can:

- Re-implement those changes after each update of the *Amazon Web Services* PowerPack.
- Remove the content from the PowerPack on your system before you update it. When the *Amazon Web Services* PowerPack is updated in your system, updated versions of this content will not be installed on your system and your local changes will be preserved.

To remove automation actions or automation policies content from the *Amazon Web Services* PowerPack on your system:

1. Go to the **PowerPack Manager** page (System > Manage > PowerPacks).
2. Click the wrench icon () for the *Amazon Web Services* PowerPack. The **Editing PowerPack** page appears.
3. In the left NavBar of the **Editing PowerPack** page, select the type of content you want to remove:
    - To remove an automation action, click **Run Book Actions**. The **Embedded Run Book Actions** and **Available Run Book Actions** panes appear.
    - To remove an automation policy, click **Run Book Policies**. The **Embedded Run Book Policies** and **Available Run Book Policies** panes appear.
4. In the upper pane, click the bomb icon () for each automation action or automation policy that you want to remove from the *Amazon Web Services* PowerPack on your system.

Aligning AWS Regions to the AWS Region Device Class

The automation for aligning an AWS Region to the correct AWS Region device class includes one automation action:

- ***AWS: Region Device Class Alignment***. This action updates the AWS device class to the correct AWS Region.

> **NOTE:** Device classes for AWS Regions are updated in the second cycle of the "AWS: Region Device Class Discovery" Dynamic Application. Regions will be updated after 24 hours.

The *Amazon Web Services* PowerPack includes an automation policy that triggers this action:

- ***AWS: Region Device Class Alignment***. If enabled, this automation policy can trigger for any device with which the "AWS: Region Device Class Discovery" Dynamic Application is aligned. The automation policy triggers when the "AWS: Device Class Change" event is active on the matching devices, and the automation policy will repeat every 10 minutes until that event is no longer active.

Vanishing Terminated or Terminating EC2 Instances

The automation in this section automatically moves an EC2 instance to a vanished state if the EC2 instance is in a terminating or terminated state. SL1 unmerges the EC2 instance and physical device, clearing the associated events, and moves the devices to a vanished state.

The automation for vanishing terminated EC2 instances includes one automation action:

- ***AWS: Vanish Terminated EC2 Instances***. If an EC2 instance has been terminated in Amazon, its corresponding device in SL1 becomes unavailable. This action then requests information from the AWS API about the EC2 instance that triggered the automation action. If the response from the AWS API indicates that the EC2 instance that triggered the automation action is in a terminated or terminating state, the action performs the following steps:
    - If the automation triggers for a physical device that is merged with an EC2 instance, the devices are un-merged.
    - If the automation triggers for a physical device that is merged with an EC2 instance, after being un-merged the physical device is moved to a virtual collector group.
    - If the automation triggers for a physical device that is merged with an EC2 instance, after being unmerged, all events associated with the physical device are cleared.
    - All events associated with the component device are cleared.
    - The component device is vanished.

NOTE: If an EC2 instance is stopped in AWS rather than terminated, then the "AWS Vanish Terminated EC2 Instances" action is not triggered.

The *Amazon Web Services* PowerPack includes an automation policy that triggers this action:

- ***AWS: Vanish Terminated EC2 Instances***. If enabled, this automation policy can trigger for any device with which the "AWS: EC2 Instance Configuration" Dynamic Application is aligned (the members of the AWS EC2 Instances). The automation policy triggers when the "Availability Check Failed" event is active on the matching devices, and the automation policy will repeat every 10 minutes until that event is no longer active.

To use this automation, you must:

- *Enable the AWS: Vanish Terminated EC2 Instances automation policy*
- *Configure your system to preserve this change*

Enabling the Automation Policies

To enable one or more automation policies in the *Amazon Web Services* PowerPack:

1. Go to the **Automation Policy Manager** page (Registry > Run Book > Automation).
2. Click the wrench icon () for the automation policy you want to enable.
3. In the ***Policy State*** field, select *Enabled*.
4. Click **[Save]**.

Preserving Automation Changes

If you have modified automation actions and policies that are included in the *Amazon Web Services* PowerPack, those changes will be overwritten when the PowerPack is updated in your system. If you have modified automation actions and policies that are included in the PowerPack, you can:

- Re-implement those changes after each update of the *Amazon Web Services* PowerPack.
- Remove the content from the PowerPack on your system before you update it. When the *Amazon Web Services* PowerPack is updated in your system, updated versions of this content will not be installed on your system and your local changes will be preserved.

To remove automation actions or automation policies content from the *Amazon Web Services* PowerPack on your system:

1. Go to the **PowerPack Manager** page (System > Manage > PowerPacks).
2. Click the wrench icon () for the *Amazon Web Services* PowerPack. The **Editing PowerPack** page appears.
3. In the left NavBar of the **Editing PowerPack** page, select the type of content you want to remove:
    - To remove an automation action, click **Run Book Actions**. The **Embedded Run Book Actions** and **Available Run Book Actions** panes appear.
    - To remove an automation policy, click **Run Book Policies**. The **Embedded Run Book Policies** and **Available Run Book Policies** panes appear.
4. In the upper pane, click the bomb icon () for each automation action or automation policy that you want to remove from the *Amazon Web Services* PowerPack on your system.

Copyrights and Trademarks

ScienceLogic, the ScienceLogic logo, and EM7 are trademarks of ScienceLogic, Inc. in the United States, other countries, or both.

Below is a list of trademarks and service marks that should be credited to ScienceLogic, Inc. The ® and ™ symbols reflect the trademark registration status in the U.S. Patent and Trademark Office and may not be appropriate for materials to be distributed outside the United States.

- ScienceLogic™
- EM7™ and em7™
- Simplify IT™
- Dynamic Application™
- Relational Infrastructure Management™

The absence of a product or service name, slogan or logo from this list does not constitute a waiver of ScienceLogic's trademark or other intellectual property rights concerning that name, slogan, or logo.

Please note that laws concerning use of trademarks or product names vary by country. Always consult a local attorney for additional guidance.

Other

If any provision of this agreement shall be unlawful, void, or for any reason unenforceable, then that provision shall be deemed severable from this agreement and shall not affect the validity and enforceability of any remaining provisions. This is the entire agreement between the parties relating to the matters contained herein.

In the U.S. and other jurisdictions, trademark owners have a duty to police the use of their marks. Therefore, if you become aware of any improper use of ScienceLogic Trademarks, including infringement or counterfeiting by third parties, report them to Science Logic's legal department immediately. Report as much detail as possible about the misuse, including the name of the party, contact information, and copies or photographs of the potential misuse to: legal@sciencelogic.com

800-SCI-LOGIC (1-800-724-5644)

International: +1-703-354-1010