Monitoring Amazon Web Services

Amazon Web Services PowerPack version 115

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.

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 an individual data center located in a specific geographic locale. 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
- AutoScale
- CloudFront
- CloudTrail
- CloudWatch
- Direct Connect
- DynamoDB (DDB)
- ElastiCache
- Elastic Beanstalk
- Elastic Block Store (EBS)
- Elastic Compute Cloud (EC2)
- Elastic Compute Services (ECS)
- Elastic File System (EFS)

- Elastic Load Balancers (ELB)
- Elastic Map Reduce (EMR)
- Glacier
- Lambda
- Lightsail
- OpsWorks
- RedShift
- Relational Data Store (RDS)
- Route53
- Security Groups
- Shield
- Simple Notification Service (SNS)
- Simple Queue Service (SQS)
- Simple Storage Service (S3)
- Storage Gateways (ASG)
- Storage Gateway Volumes
- 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: Classic Dashboards* PowerPack. For more information, see the ***AWS Dashboards*** chapter.

Installing the Amazon Web Services PowerPack

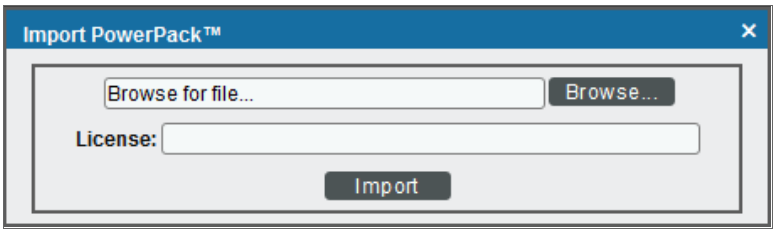
Before completing the steps in this manual, you must import and install the latest version of the *Amazon Web Services* PowerPack.

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 Customer Portal.
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 page 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*.

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.

Chapter 2

Configuration and Discovery

Overview

The following sections describe how to configure and discover Amazon Web Services and component devices for monitoring by SL1 using the *Amazon Web Services* PowerPack:

Configuring Amazon Web Services for Monitoring

To use the AWS Dynamic Applications, you must configure a credential that allows SL1 to connect to the AWS REST API. The *Amazon Web Services* PowerPack includes three credential templates.

To use the credential templates included in the PowerPack, you must download the security credentials for a user associated with your AWS account. The user must meet the following requirements:

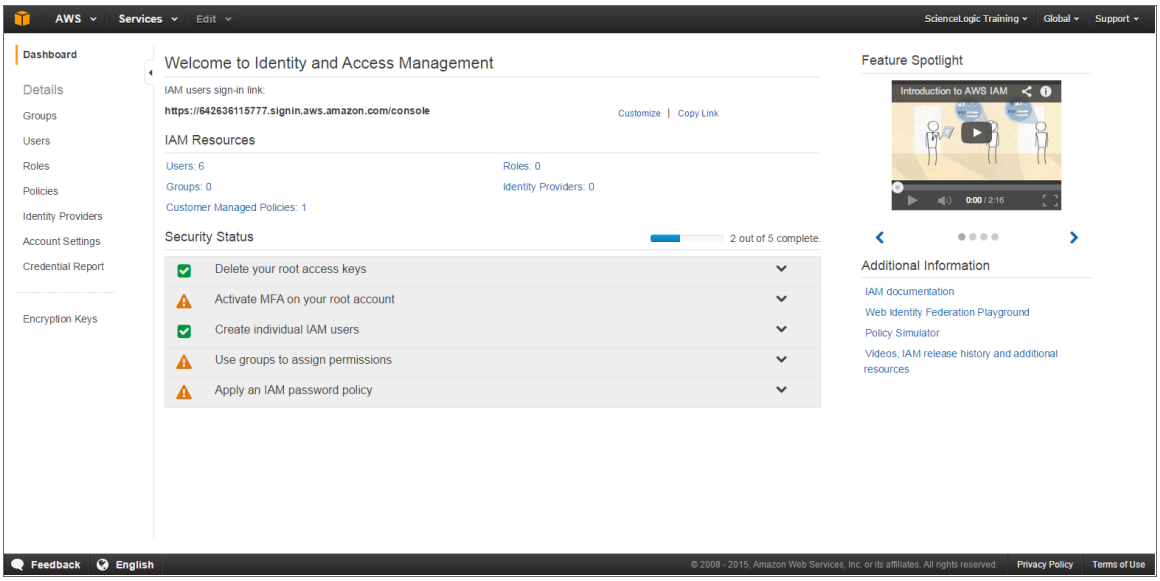
- The Dynamic Applications in the *Amazon Web Services* PowerPack require certain minimum permissions to be set. For more information, see the *Minimum Permissions for Dynamic Applications* section.
- You can use the Dynamic Applications in the *Amazon Web Services* PowerPack to discover and monitor only specific regions and services. To do so, you must create a JSON permissions policy that uses the `NotAction`, `Allow`, and `Deny` policy elements to specify which regions and services you want to monitor or not monitor and select that policy for your AWS user. For more information, see the *Configuring AWS for Region-Specific Monitoring* section.
- To collect billing metrics, the user must have read permission in the us-east-1 zone. For instructions on how to configure your AWS account to report billing metrics, see the *Configuring AWS to Report Billing Metrics* section.
- If you are using multiple users to monitor AWS, each instance of a service must be visible to only one of those users. If an instance is visible to multiple users that are used to monitor AWS in SL1, the device record for that instance will repeatedly switch between the component trees of the accounts that have visibility to that instance.

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

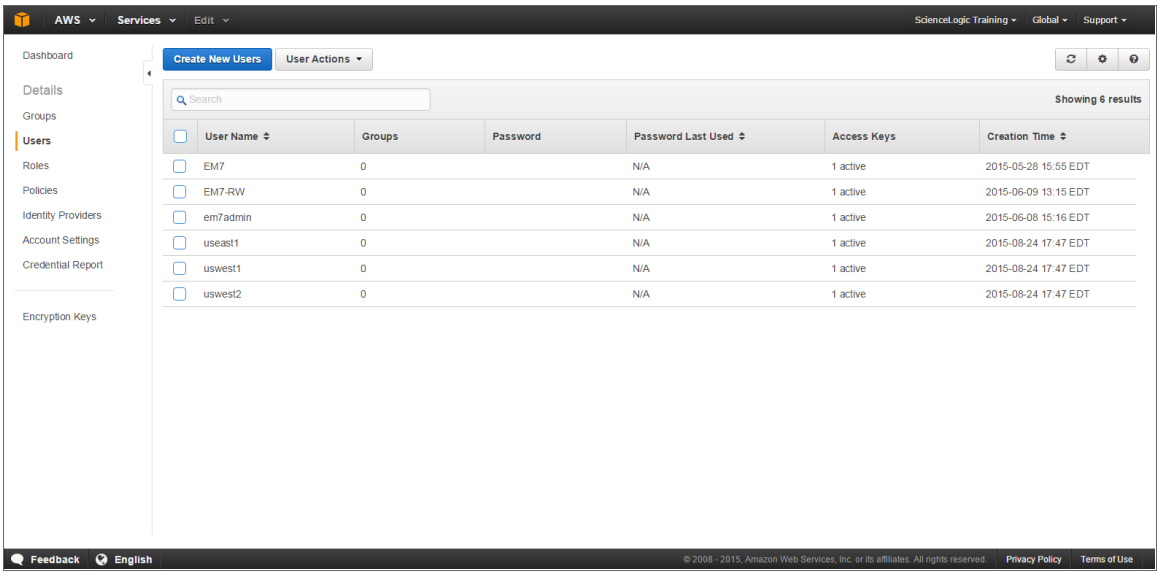
1. Open a browser session and go to aws.amazon.com.

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:

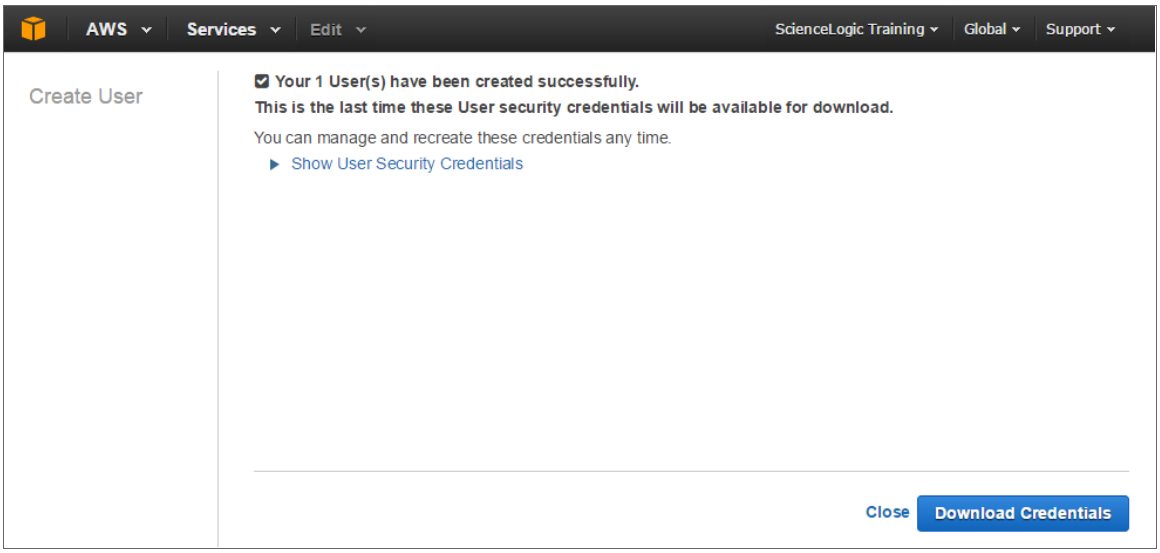


5. To create a user account for SL1, click **[Users]** on the Dashboard menu.

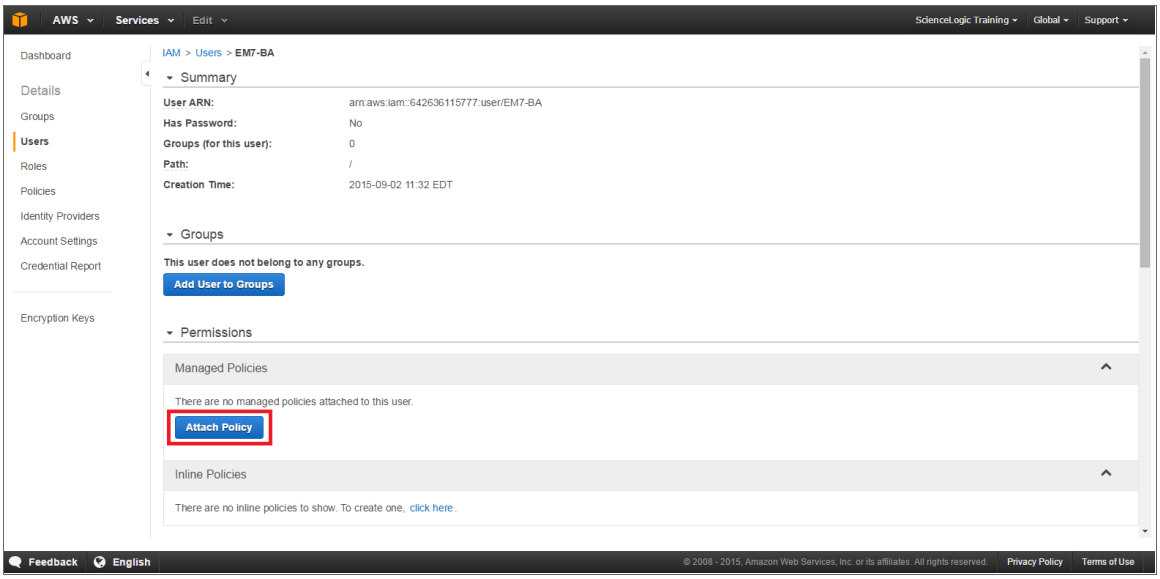


6. Click the **[Create New Users]** button.
7. Enter a username for the new user, e.g. "EM7", and make sure the ***Generate an access key for each user*** checkbox is selected.

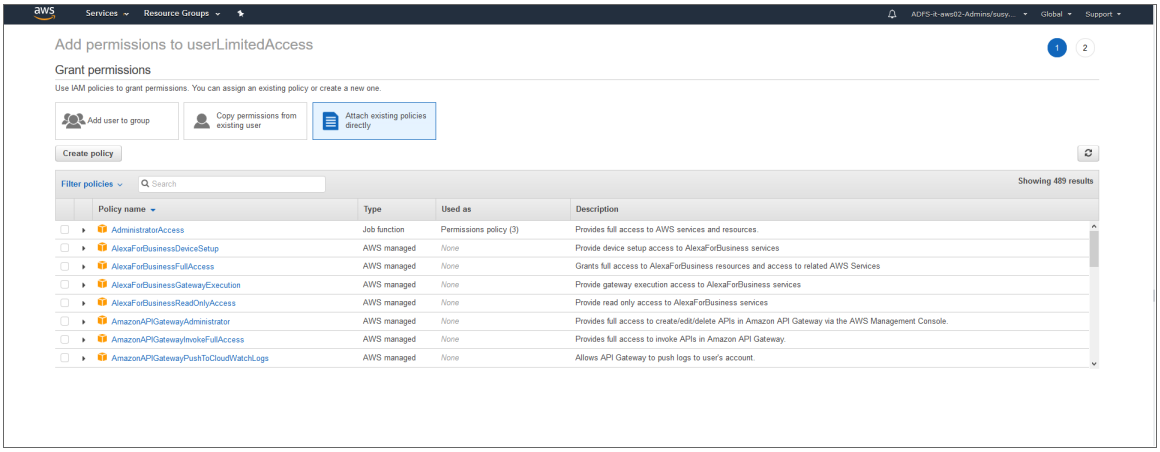
8. Click the **[Create]** button to generate your user account. The **Create User** page appears:



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.

Minimum Permissions for Dynamic Applications

The following table displays the minimum permissions required for Dynamic Applications in the *Amazon Web Services* PowerPack to collect data. These permissions, among others, are in the ReadOnlyAccess AWS Managed policy. ScienceLogic does not recommend using this policy. Instead, ensure that the account used to discover AWS is based on the minimum permission list.

Service	Actions	
API Gateway	Read	GET
CloudFront	List	ListDistributions ListInvalidations ListStreamingDistributions
	Read	GetDistribution GetStreamingDistribution
CloudTrail	List	DescribeTrails
	Read	GetTrailStatus
CloudWatch	List	ListMetrics
	Read	DescribeAlarmHistory DescribeAlarms GetMetricData GetMetricStatistics
Direct Connect	Read	DescribeConnections DescribeTags

Service	Actions	
		DescribeVirtualInterfaces
DynamoDB	List	ListTables
	Read	DescribeTable
EC2	List	DescribeAvailabilityZones DescribeInstances DescribeNetGateways DescribeRegions DescribeRouteTables DescribeSecurityGroups DescribeSubnets DescribeSnapshots DescribeVolumes DescribeVpcPeeringConnections DescribeVpcs DescribeVpnGateways
	Read	DescribeVpnConnections
EC2 Auto Scaling	List	DescribeAutoScalingGroups DescribeAutoScalingInstances DescribeLaunchConfigurations
EFS	List	DescribeFileSystems
Elastic Beanstalk	List	DescribeEnvironments
	Read	DescribeConfigurationSettings DescribeEnvironmentResources
Elastic Container Service	List	ListClusters ListContainerInstances ListServices ListTasks
	Read	DescribeClusters DescribeContainerInstances DescribeServices DescribeTaskDefinition DescribeTasks
ElasticCache	List	DescribeCacheClusters
ELB	List	DescribeLoadBalancers
	Read	DescribeTags
ELB v2	Read	DescribeListeners DescribeLoadBalancers DescribeTags DescribeTargetGroups DescribeTargetHealth
EMR	List	ListClusters ListInstances
Glacier	List	ListTagsForVault

Service	Actions	
		ListVaults
	Read	GetVaultNotifications
IAM	Read	GetUser
Lambda	List	ListAliases ListEventSourceMappings ListFunctions
	Read	GetAccountSettings GetPolicy ListTags
Lightsail	List	GetBundles GetInstances GetRegions
	Read	GetInstanceMetricData
OpsWorks	List	DescribeInstances DescribeStacks
RDS	List	DescribeDBInstances DescribeDBSubnetGroups
	Read	ListTagsForResource
Redshift	List	DescribeClusters
	Read	DescribeLoggingStatus
Route 53	List	GetHostedZone ListHealthChecks ListHostedZones ListResourceRecordSets
S3	List	ListAllMyBuckets ListBucket
	Read	GetBucketLication 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
SNS	List	ListSubscriptions ListTopics
SQS	List	ListQueues
	Read	GetQueueAttributes
Storage	List	ListGateways

Service	Actions	
Gateway		ListVolumes
WAF	List	ListWebACLs
	Read	GetRateBasedRule GetRule GetRuleGroup GetWebACL
WAF Regional	List	ListResourcesForWebACL ListWebACLs
	Read	GetRateBasedRule GetRule GetRuleGroup GetWebACL

Configuring AWS for Region-Specific Monitoring

You can use the Dynamic Applications in the *Amazon Web Services* PowerPack to discover and monitor only the specific regions and services for which your AWS user has IAM policy permissions.

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. You must then attach this permissions policy to the AWS user account you created.

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 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:

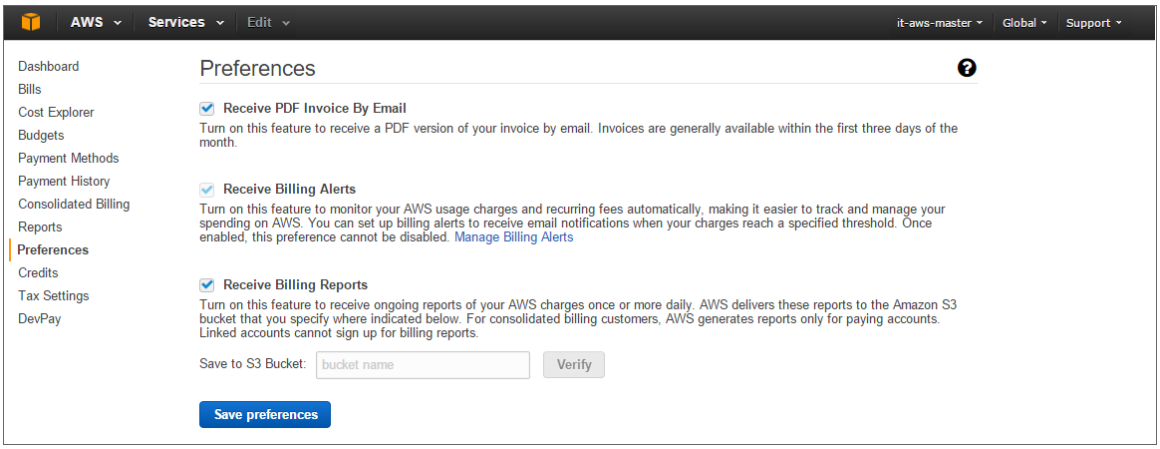
```
No billing metrics can be retrieved. Your AWS account is not configured to export
billing metrics into CloudWatch.
```

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.

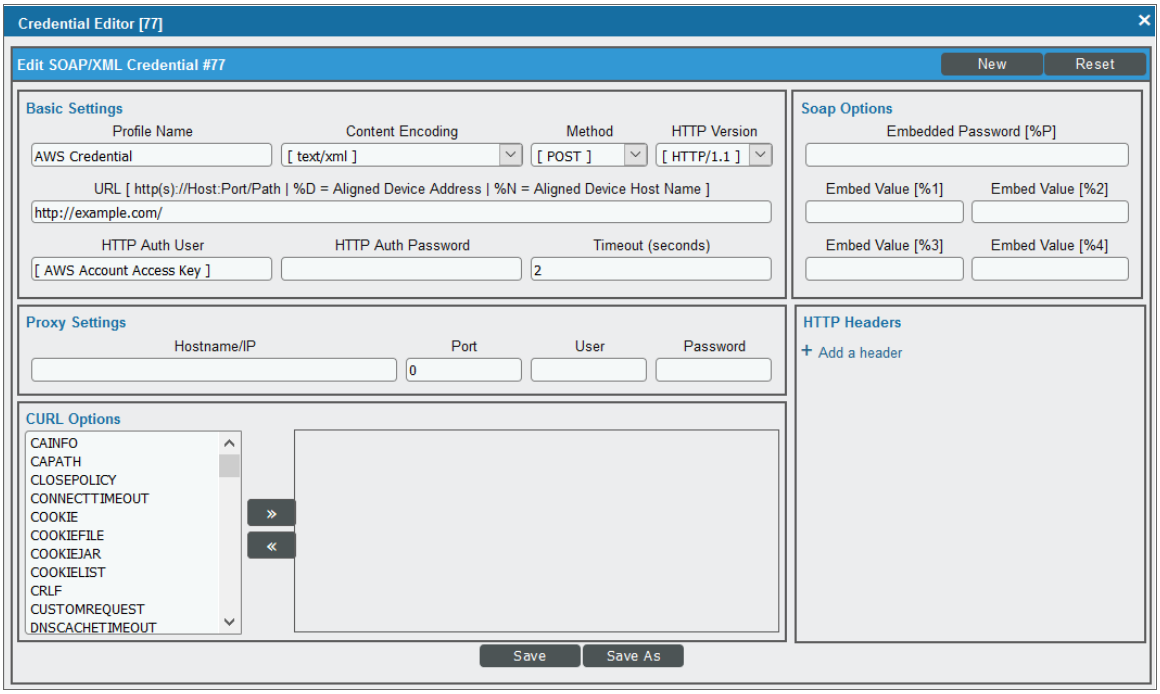
Creating an AWS Credential

To use the Dynamic Applications in the *Amazon Web Services* PowerPack, you must first define an AWS credential in SL1. The PowerPack includes the following sample credentials you can use as templates for creating SOAP/XML credentials for AWS:

- **AWS Credential - Proxy**, for users who connect to AWS through a third-party proxy server
- **AWS Credential - Specific Region**, for users who connect to a GovCloud account or Chinese regions
- **AWS Credential**, for users who do not use a proxy server nor connect to a specific AWS region

To define an AWS credential:

1. Go to the **Credential Management** page (System > Manage > Credentials).
2. Locate the **AWS Credential**, **AWS Credential - Proxy**, or **AWS Credential - Specific Region** 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 ***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.

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

Testing the AWS Credential

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.

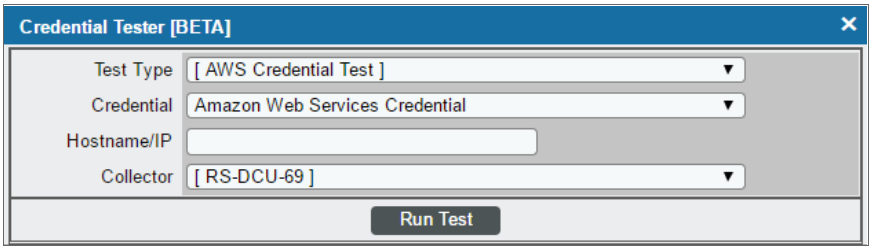
The AWS Credential Test can be used to test a SOAP/XML credential for monitoring AWS using the Dynamic Applications in the *Amazon Web Services* PowerPack. The AWS Credential Test performs the following steps:

- ***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:

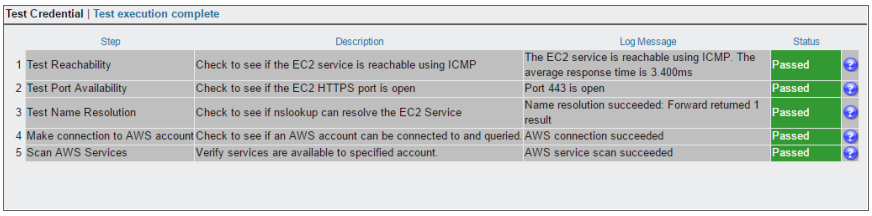
1. Go to the **Credential Test Management** page (System > Customize > Credential Tests).
2. Locate the **AWS Credential Test** and click its lightning bolt icon ( ). The **Credential Tester** modal page appears:



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".

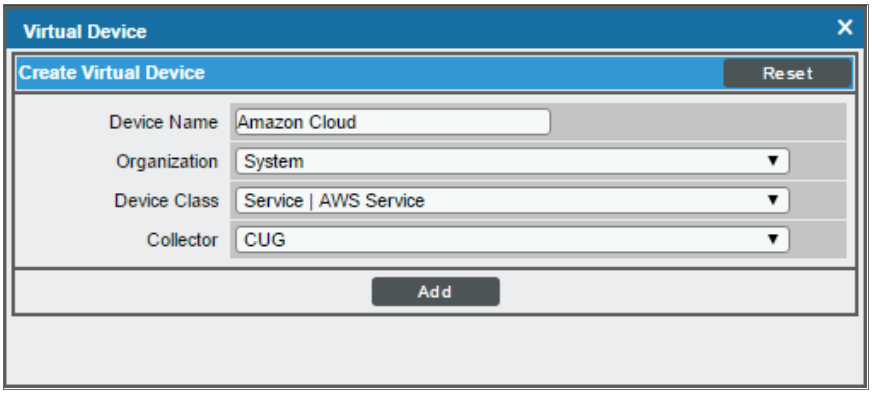
Creating an AWS Virtual Device

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 (Registry > Devices > Device Manager).

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.

Understanding the AWS Dynamic Applications

The Dynamic Applications in the *Amazon Web Services* PowerPack are divided in to four types:

- ***Discovery***. These Dynamic Applications poll AWS for new instances of services or changes to existing instances of services.
- ***Configuration***. These Dynamic Applications retrieve configuration information about each service instance and retrieve any changes to that configuration information.
- ***Performance***. These Dynamic Applications poll AWS for performance metrics.
- ***Health***. These Dynamic Applications collect the RSS status update messages from the Amazon Health Status page (http://status.aws.amazon.com/).

Service Discovery Dynamic Applications are responsible for searching the AWS cloud for instances of specific services. Typically, a Service Discovery Dynamic Application will then align Discovery Dynamic Applications for each AWS service it discovers, Performance Dynamic Applications for each discovered service, and Configuration Dynamic Applications for each discovered service.

For example, the Dynamic Application "AWS EC2 Service Discovery" will create a component device for the EC2 service and align the Dynamic Applications "EC2 Service Performance", "EC2 Service Health", and "EC2 Instance Discovery" to that component device.

The Dynamic Application "EC2 Instance Discovery" will create component devices for each EC2 instance and align the Dynamic Applications "EC2 Instance Configuration", "EC2 Instance Performance", and "EBS Discovery" to that component device.

The general Dynamic Application hierarchy is:

- Account Discovery
    - Region Discovery
        - Zone Discovery
            - Service Discovery
                - Service Performance
                - Service Health
                - Instance Discovery
                    - Instance Configuration
                    - Instance Performance

AWS Account Discovery

The Dynamic Application "AWS Account Discovery" is the root Dynamic Application that retrieves the user's account permissions. The "Account" component device uses the full user ID as the device name.

The "AWS Account Discovery" Dynamic Application aligns the "AWS Region Discovery" Dynamic Application to the account component device. This Dynamic Application discovers the AWS Regions that contain services for the user.

The "AWS Account Discovery" Dynamic Application retrieves account-specific performance statistics such as total number of API calls and custom (application-specific) performance metrics.

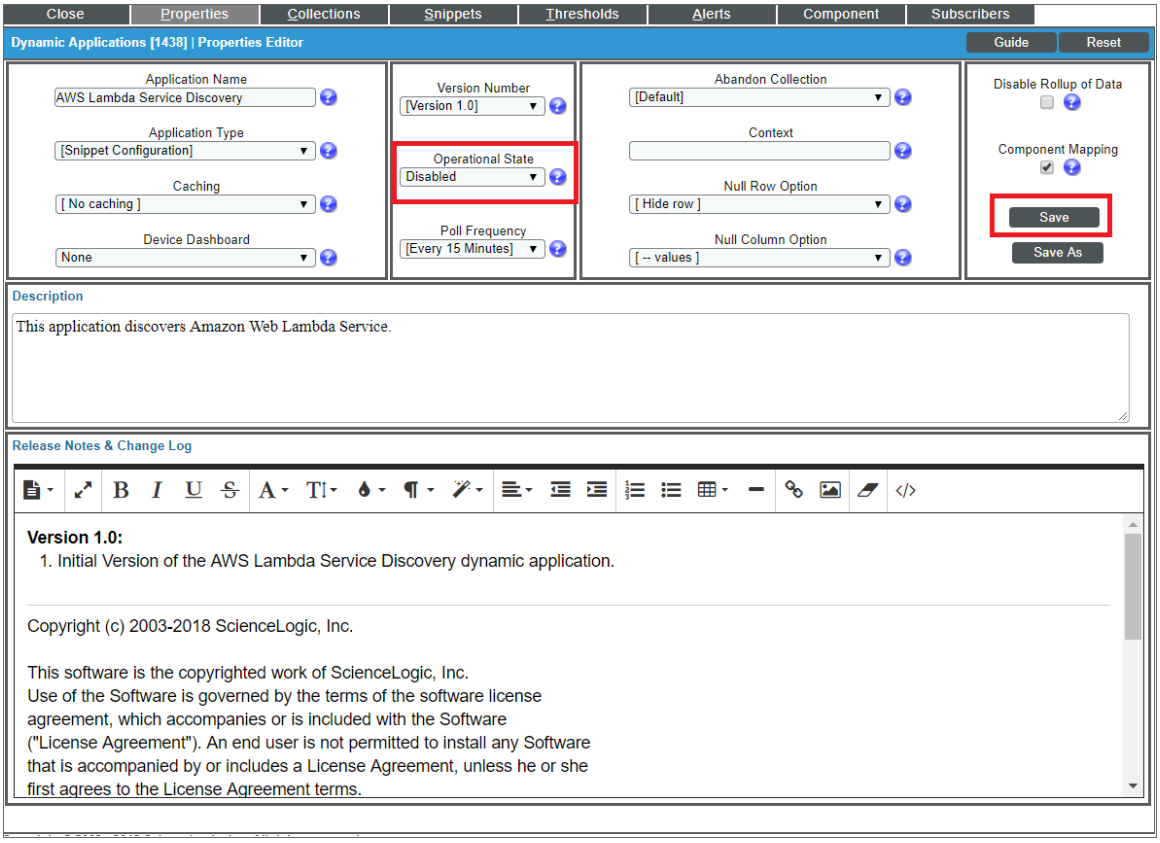
Configuring "AWS Lambda Service Discovery"

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:

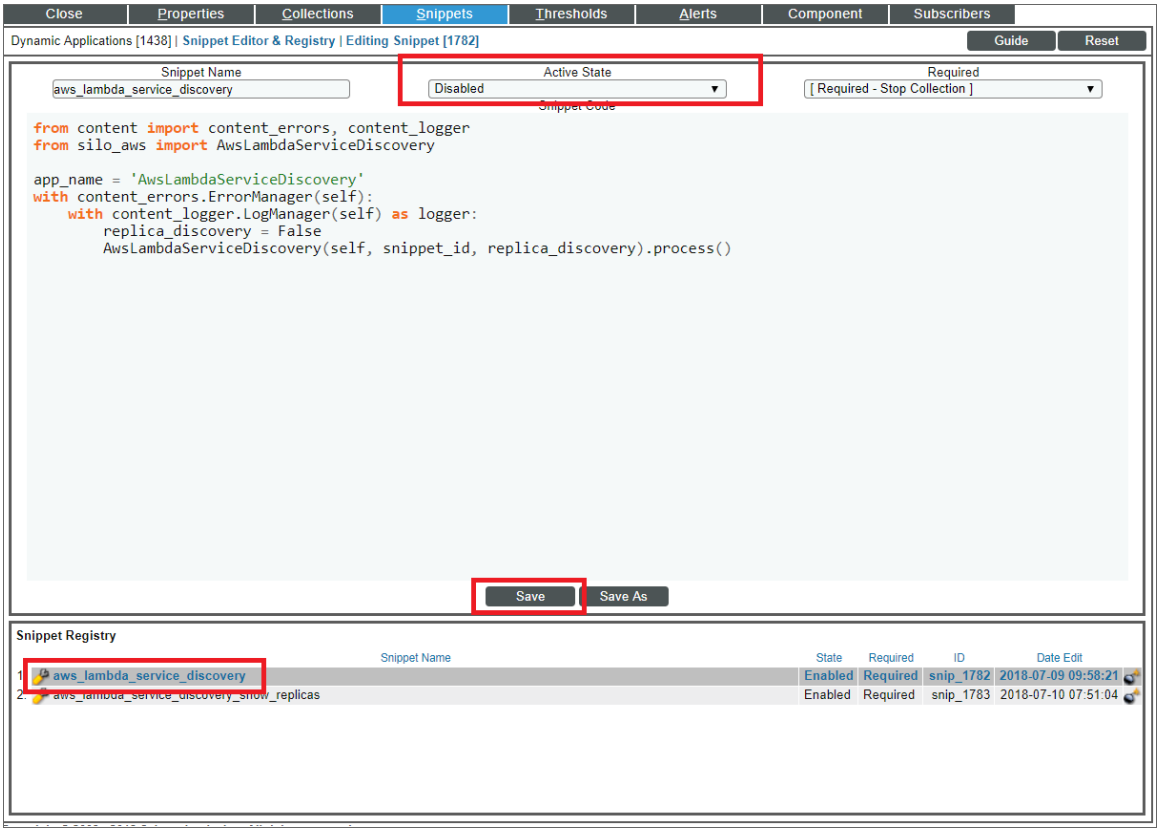
1. Go to the **Dynamic Applications Manager** page (System > Manage > Applications).
2. Locate the "AWS Lambda Service Discovery" Dynamic Application and click its wrench icon (). The **Dynamic Applications Properties Editor** page appears.

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



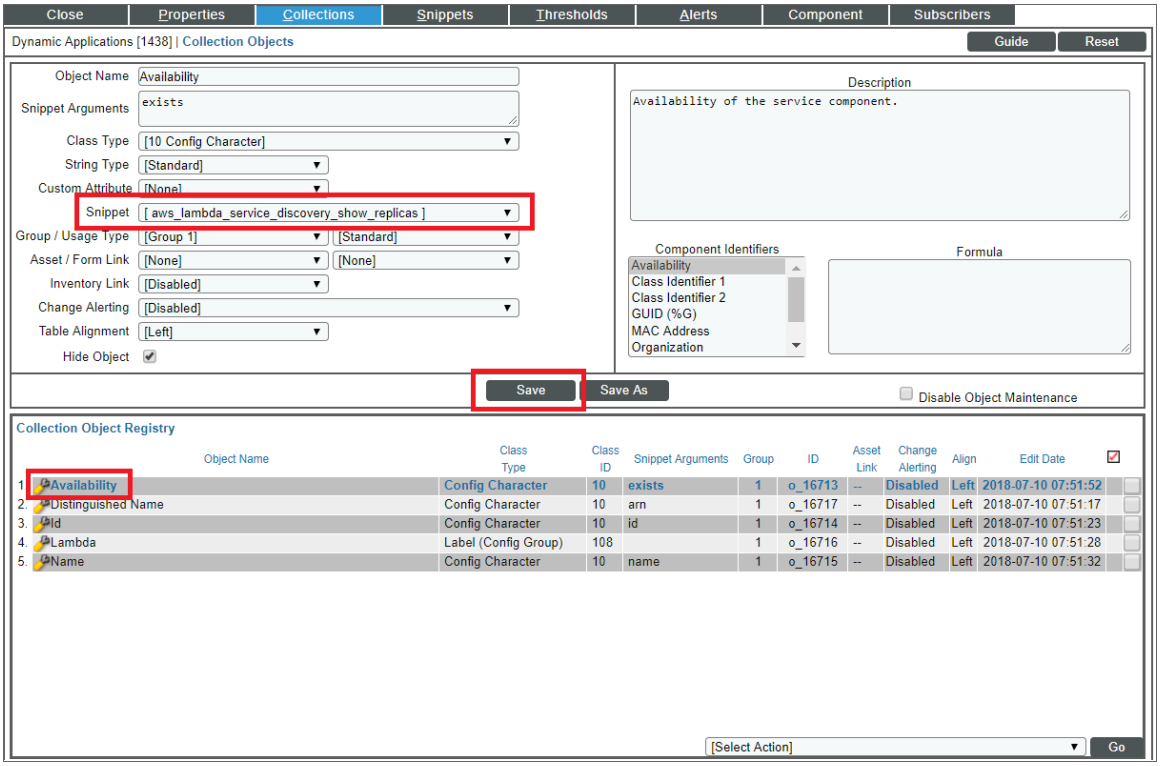
4. Click the **[Snippets]** tab. The **Dynamic Applications Snippet Editor & Registry** page appears.
5. In the **Snippet Registry** pane, click the wrench icon ( ) for the "aws_lambda_service_discovery" snippet.

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:

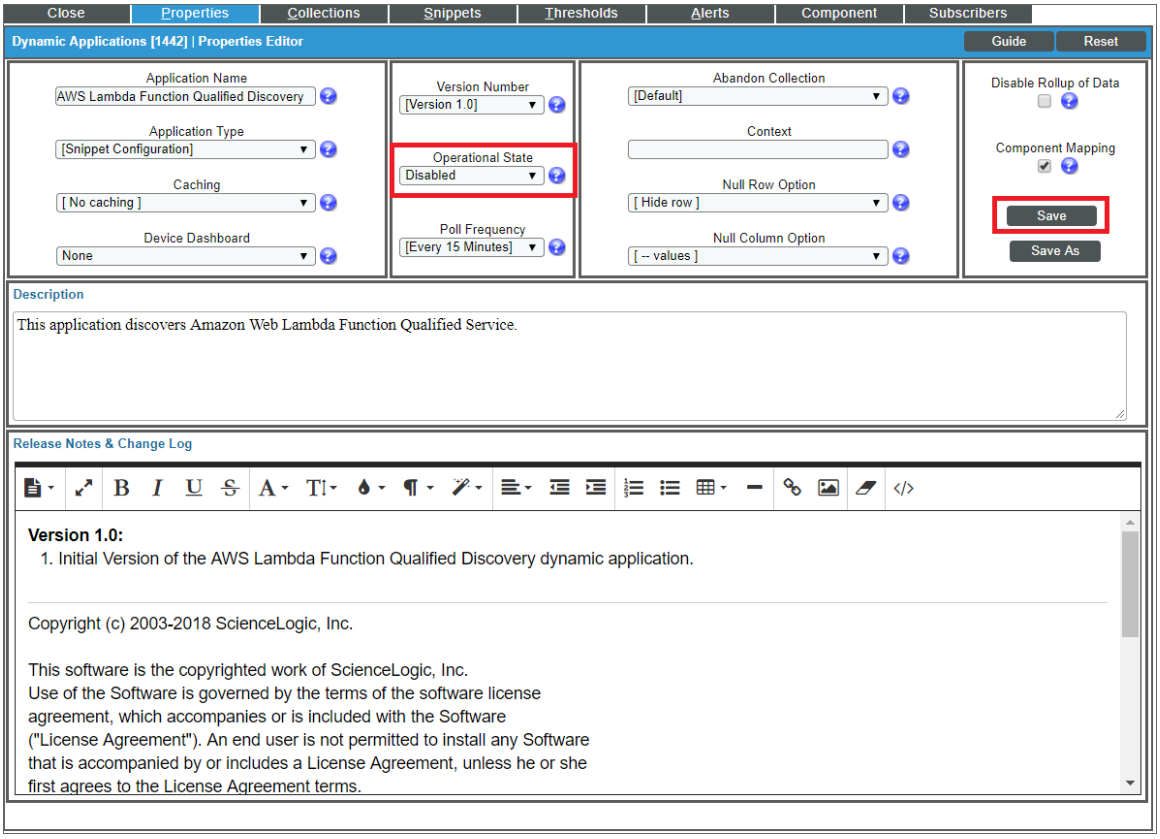
- Discover and model all Lambda version components. If you select this configuration, SL1 collects data for all existing versions of the Lambda function.
- Discover and model only Lambda version components with AWS configurations filtered by a trigger. If you select this configuration, SL1 collects data 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:

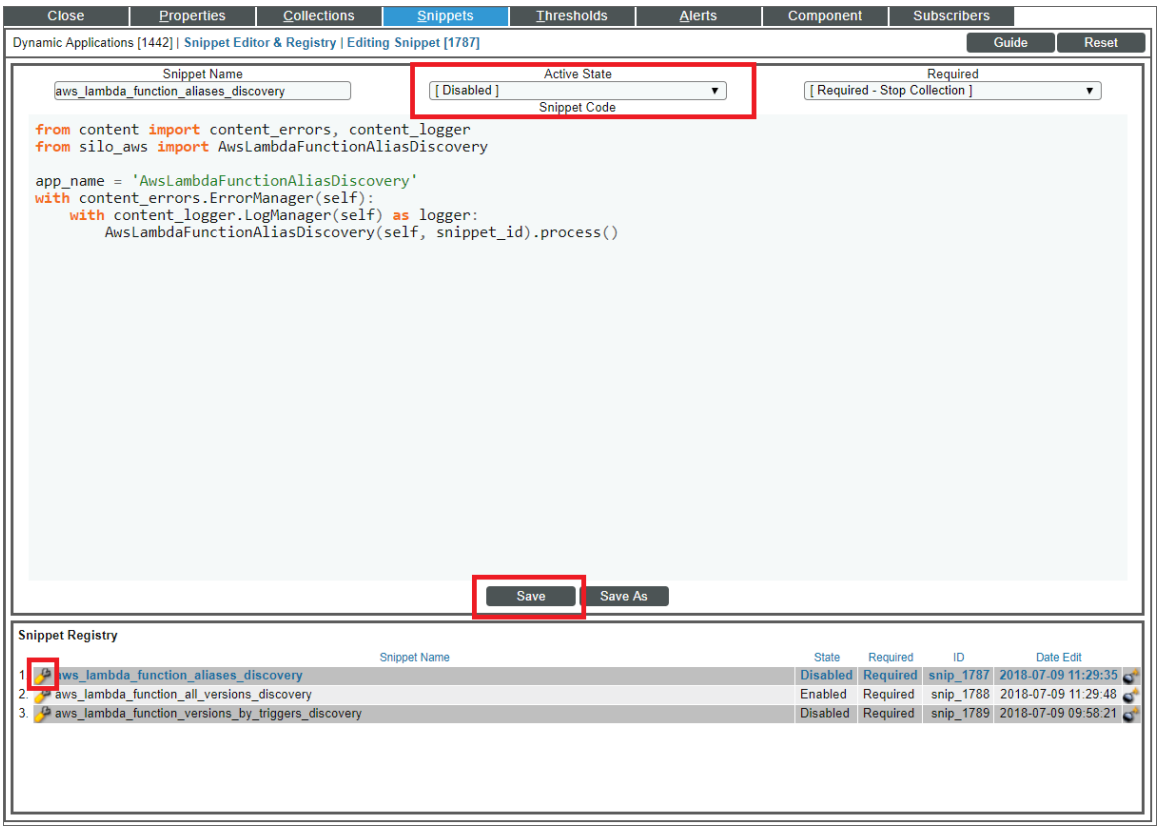
1. Go to the **Dynamic Applications Manager** page (System > Manage > Applications).
2. Locate the "AWS Lambda Function Qualified Discovery" Dynamic Application and click its wrench icon (). The **Dynamic Applications Properties Editor** page appears.

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.

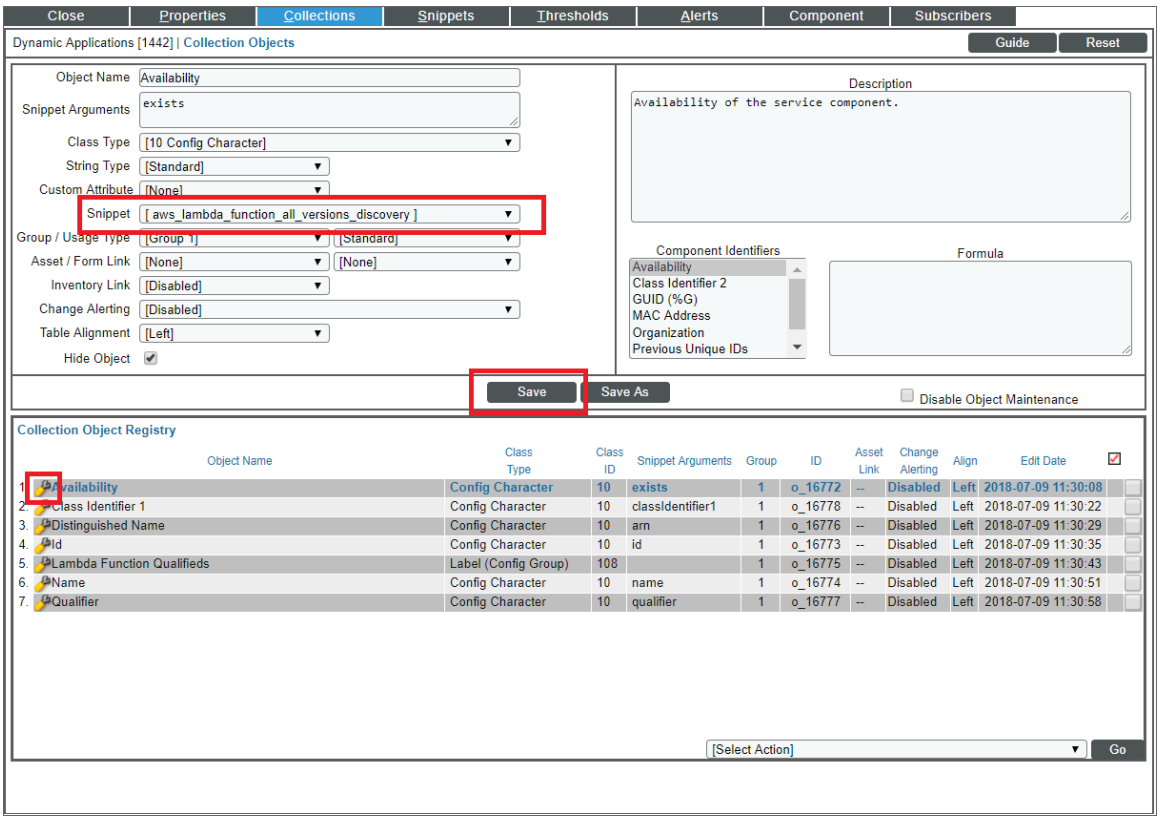
5. One at a time, click the wrench icon ( ) for each of the snippets, select *Enabled* or *Disabled* in the ***Active State*** field, and then click **[Save]** to enable the appropriate snippet and disable the others.



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.

Discovering the AWS Account

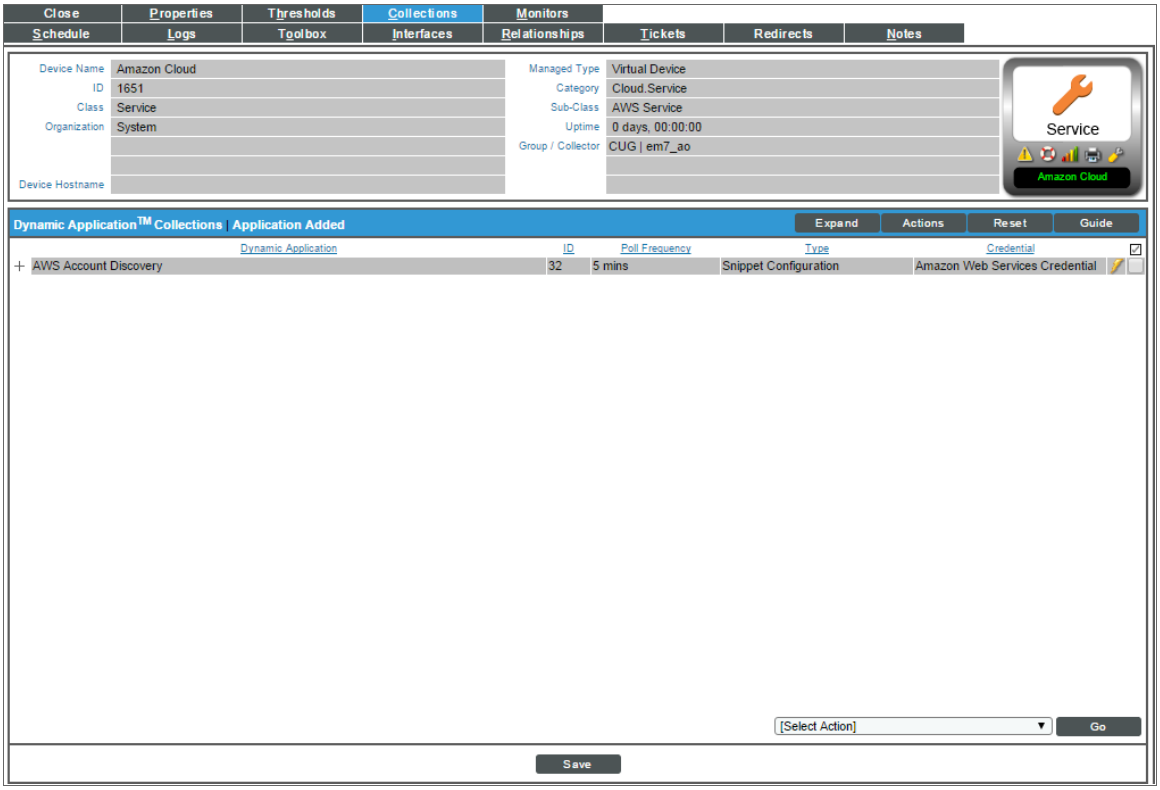
To discover your AWS account, you must manually align the "AWS Account Discovery" Dynamic Application with the AWS virtual device. After you do so, the other Dynamic Applications in the *Amazon Web Services* PowerPack will automatically align to discover and monitor all of the components in your AWS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

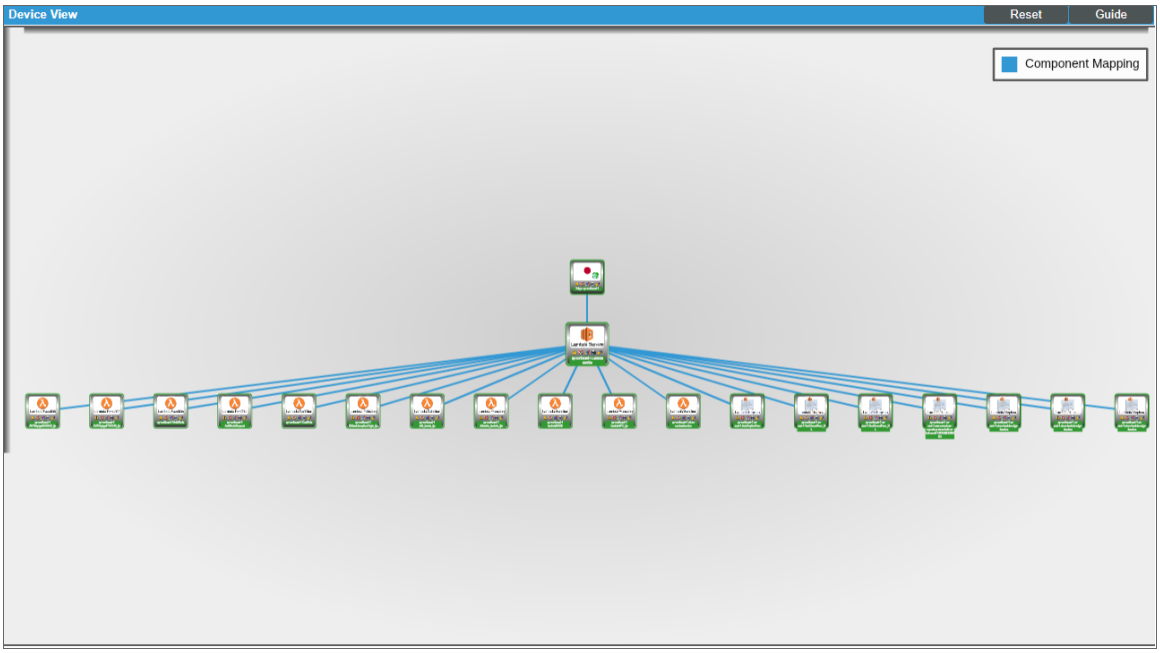


Viewing AWS Component Devices

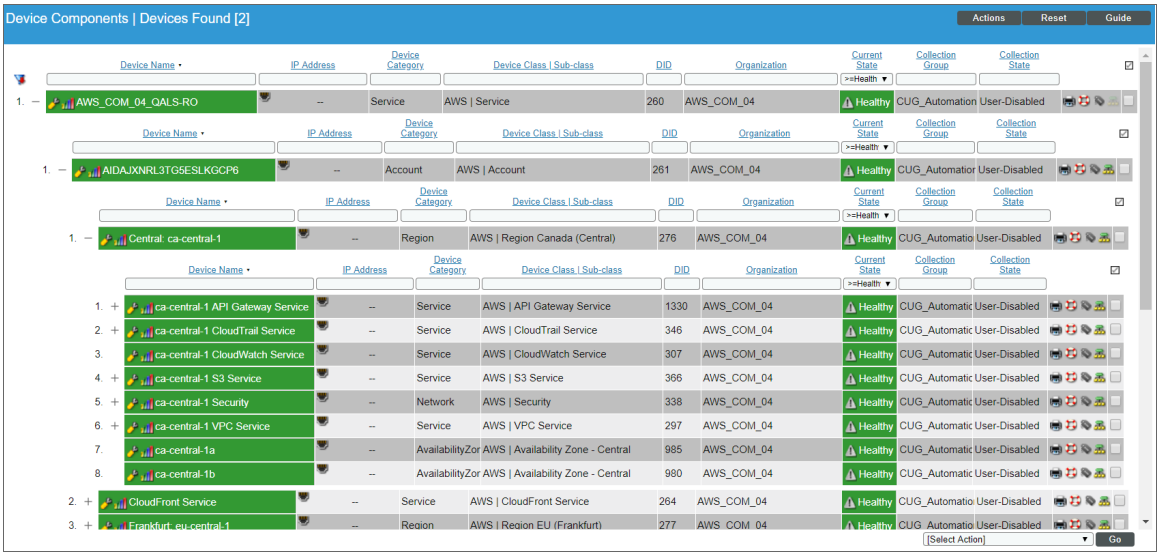
When SL1 performs collection for the AWS virtual device, SL1 will create component devices that represent each element in your AWS infrastructure and align other Dynamic Applications to those component devices. Some of the Dynamic Applications aligned to the component devices will also be used to create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

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

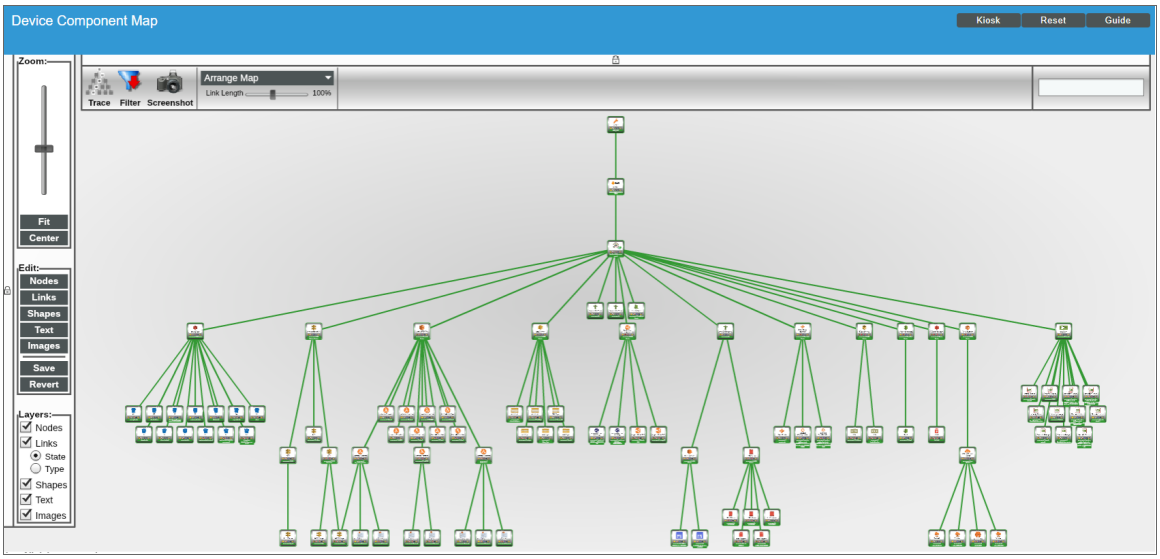
- The **Device View** page displays a map of a particular device and all of the devices with which it has parent-child relationships. Double-clicking any of the devices listed reloads the page to make the selected device the primary device:



- The **Device Components** page (Registry > 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 (Views > 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 Views > Device Maps > Components, and select the map from the list in the left NavBar. To learn more about the **Component Map** page, see the ***Views*** 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 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 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* in this manual 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.

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.

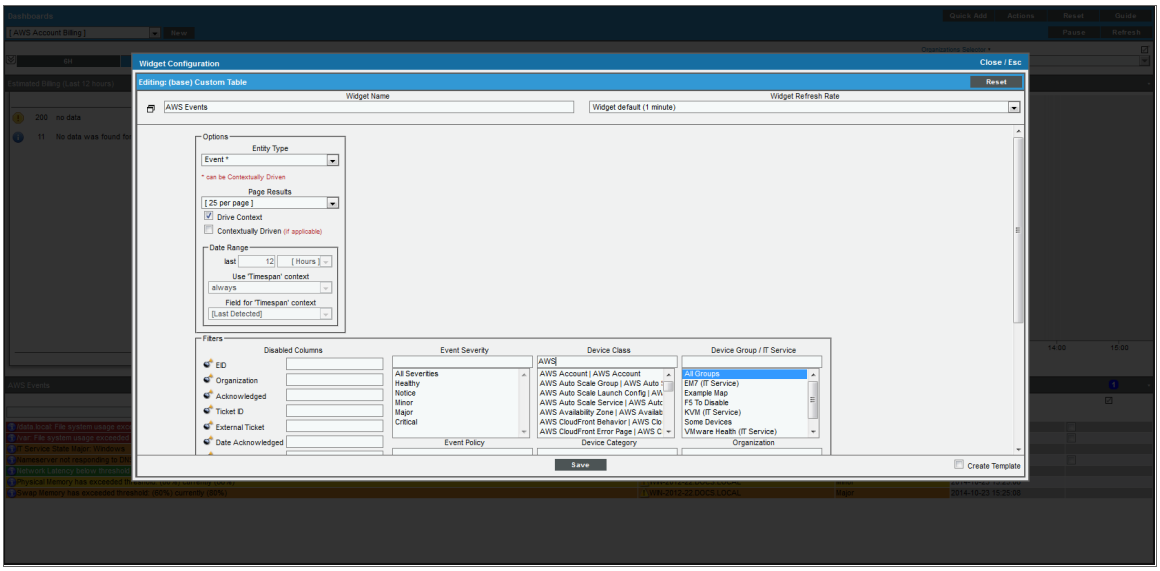
NOTE: ScienceLogic does not recommend enabling and securing the Docker HTTP API when aligning EC2 instances with Docker hosts. Doing so requires you to complete manual steps on each EC2 host. Furthermore, if you use this method and then merge the EC2 host with the Docker host, data collection will fail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

Configuring 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 > 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.

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.

Chapter 3

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:

Event Policy Name	Description	Event Source	Severity
AWS: CloudWatchAlarm_Action_Failed	An Amazon CloudWatch alarm action has failed.	API	Major
AWS: CloudWatchAlarm_Action_InProgress	An Amazon CloudWatch alarm action is in progress.	API	Notice
AWS: CloudWatchAlarm_Action_Succeeded	An Amazon CloudWatch alarm action has succeeded.	API	Notice
AWS: CloudWatchAlarm_ConfigurationUpdate	A ConfigurationUpdate alarm type is received.	API	Notice
AWS: CloudWatchAlarm_StateUpdate_Alarm	A CloudWatch alarm transitions to an "Alarm" state.	API	Major
AWS: CloudWatchAlarm_StateUpdate_InsufficientData	A CloudWatch alarm transitions to an "Insufficient Data" state.	API	Notice
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'
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.
Redshift	'ClusterIdentifier' **NOTE:** Alarms for this service will be associated with the component device for the AWS account.
Route53	'HealthCheckId'

AWS Service	Dimension
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.

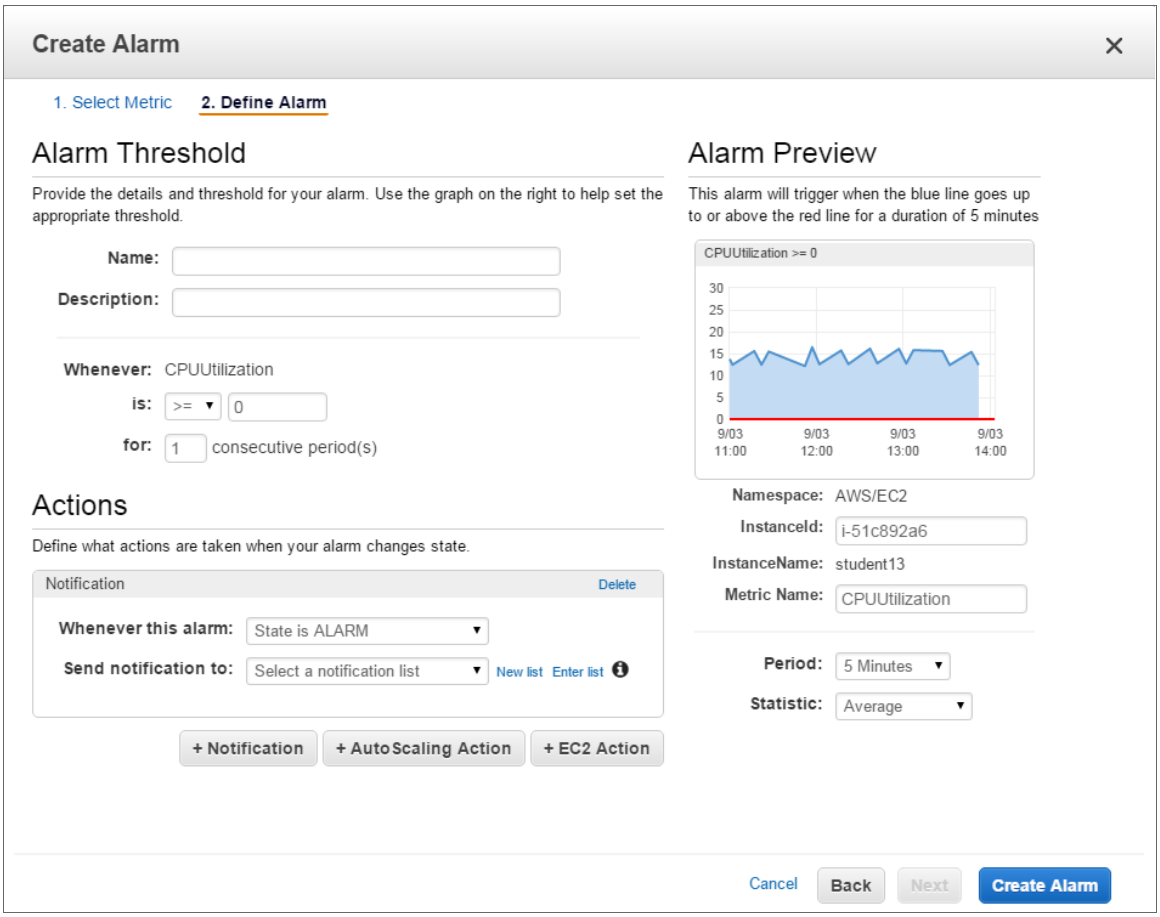
Configuring CloudWatch to Send Alarms for a Metric

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

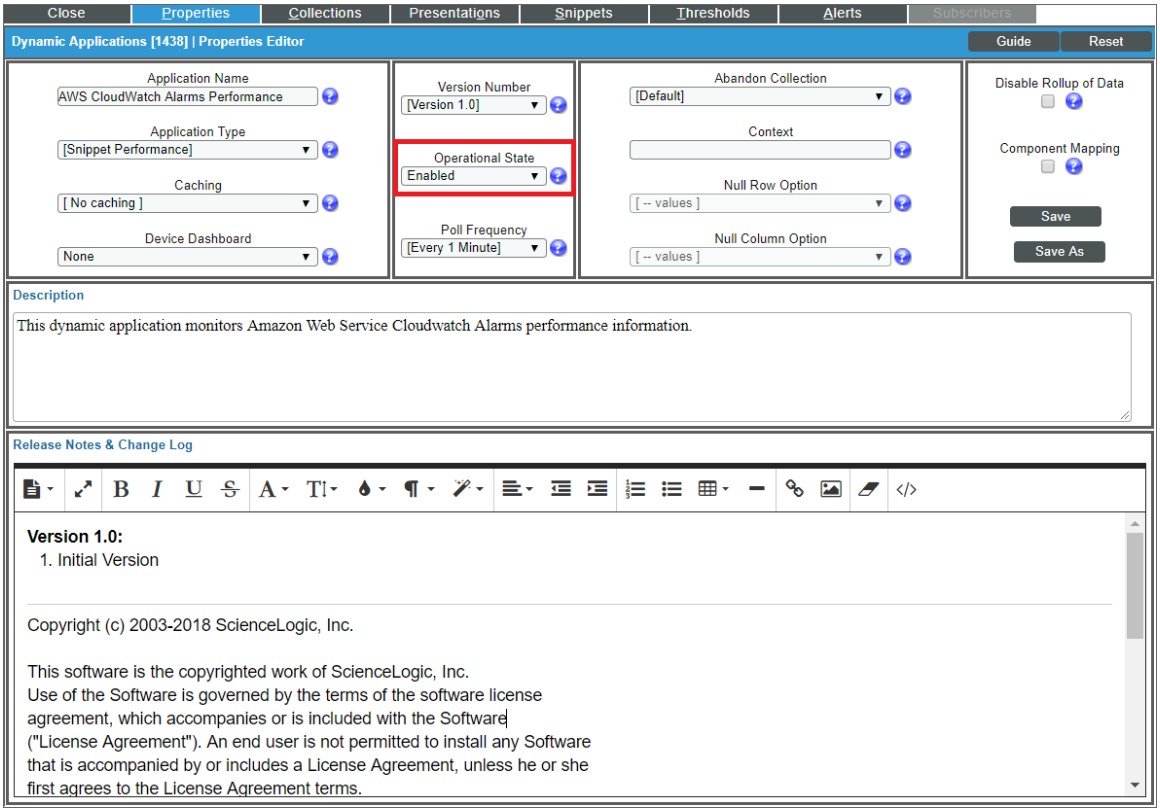
The *Amazon Web Services* PowerPack includes an "AWS CloudWatch Alarms Performance" Dynamic Application that monitors CloudWatch alarms and associates the alarms with the appropriate AWS component devices, if applicable. This Dynamic Application must be enabled if you want SL1 to generate CloudWatch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



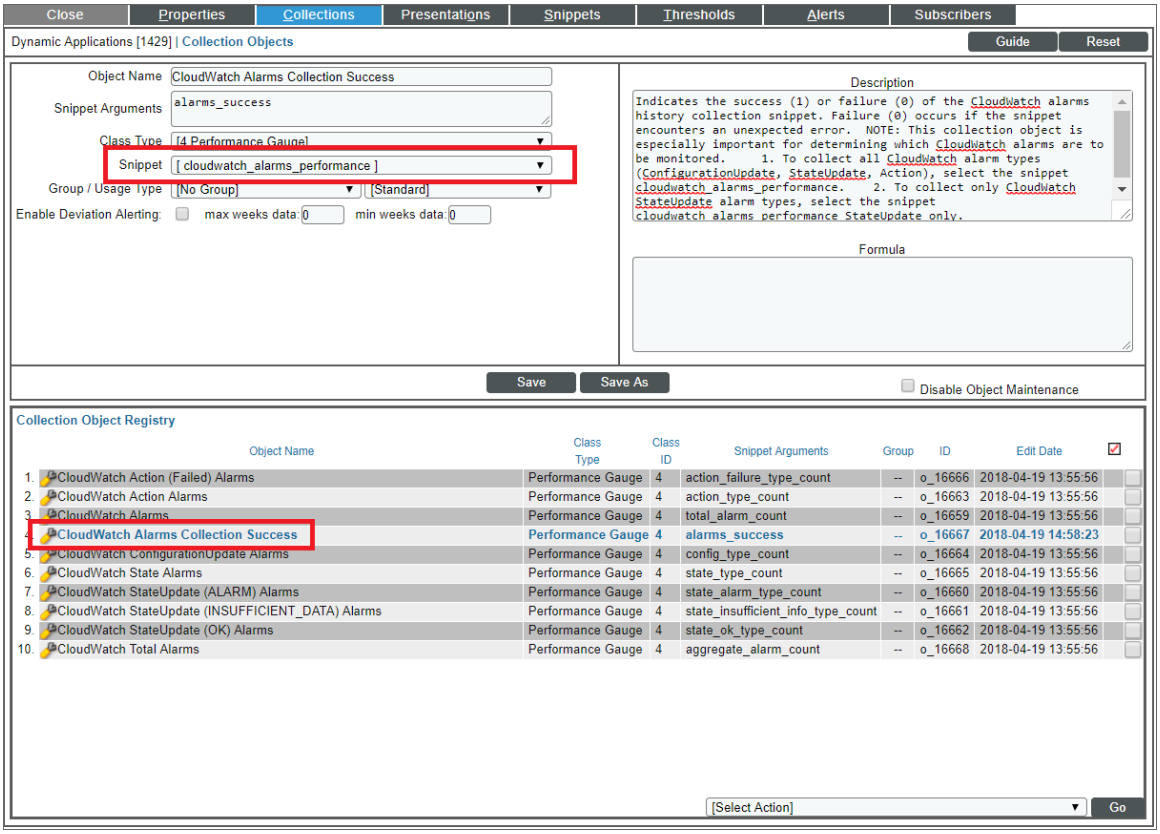
3. In the ***Operational State*** field, select *Enabled*.
4. Click **[Save]**.

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 *cloudwatch_alarms_performance*.

> **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]**.

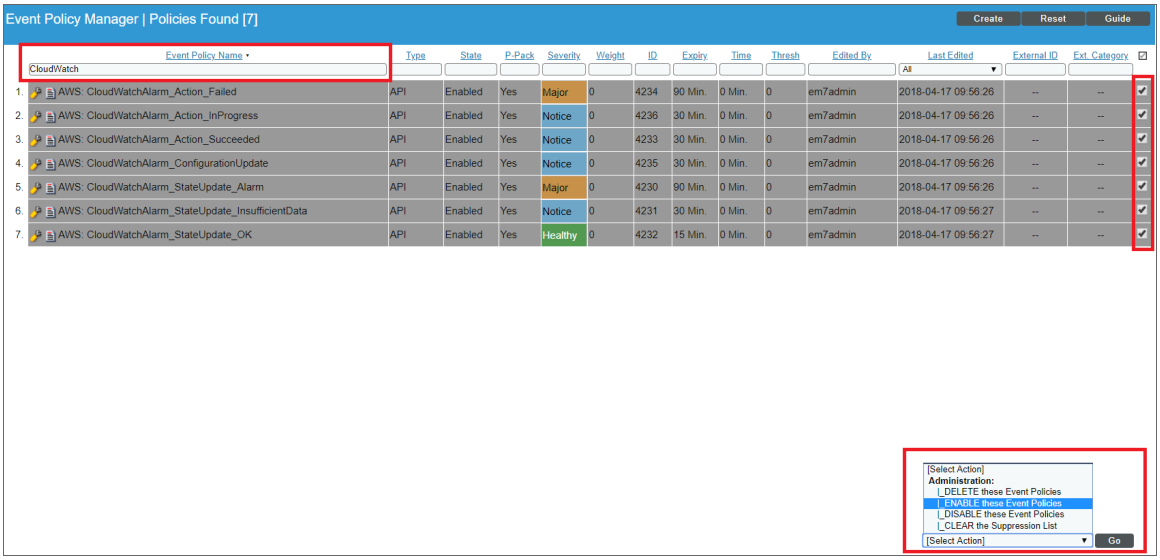
Enabling CloudWatch Alarm Events in SL1

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 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 4

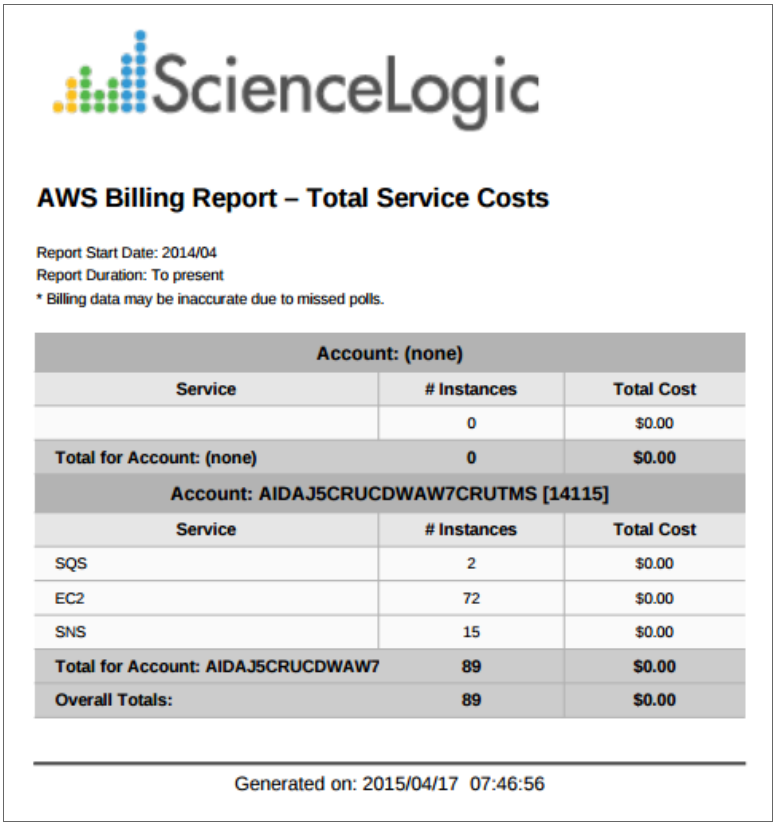
Reports

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.



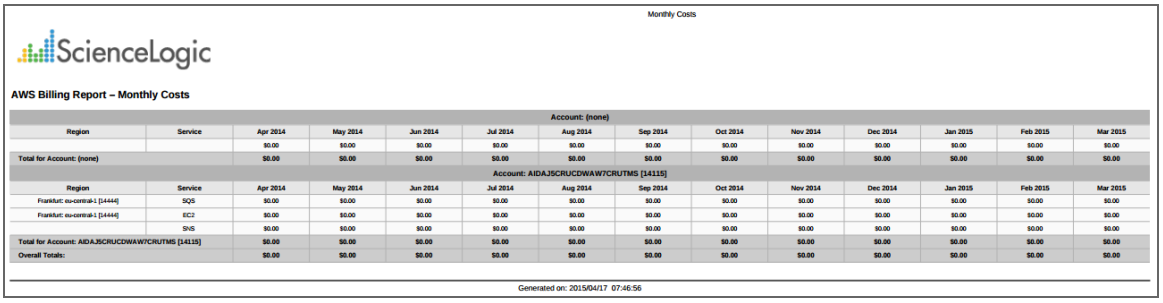
ScienceLogic

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



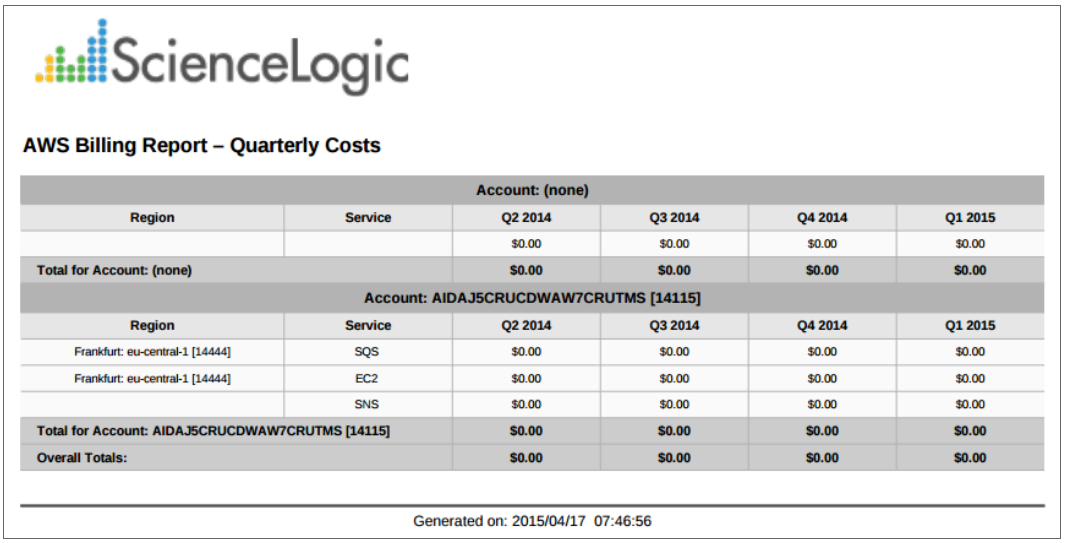
Monthly Costs

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: eu-central-1 [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: eu-central-1 [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

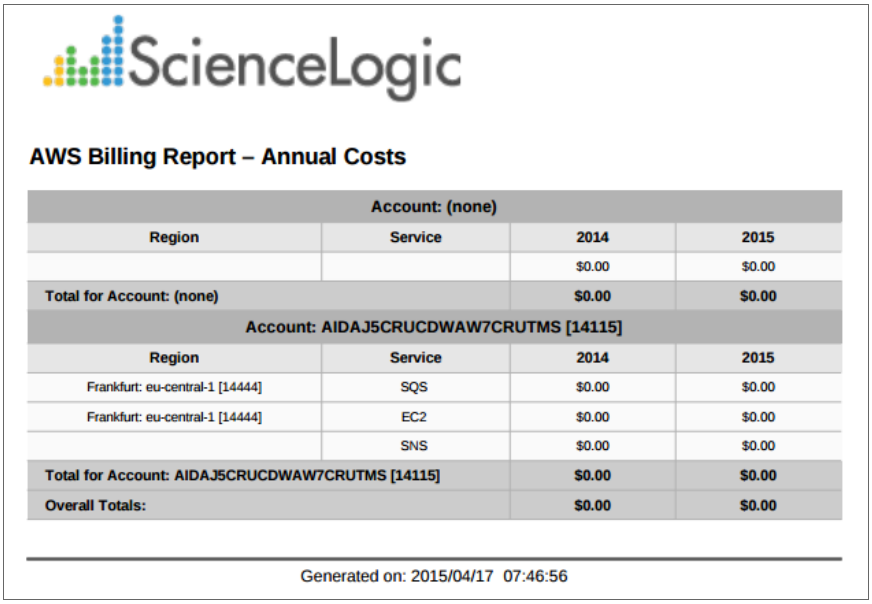
Generated on: 2015/04/17 07:46:56



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**

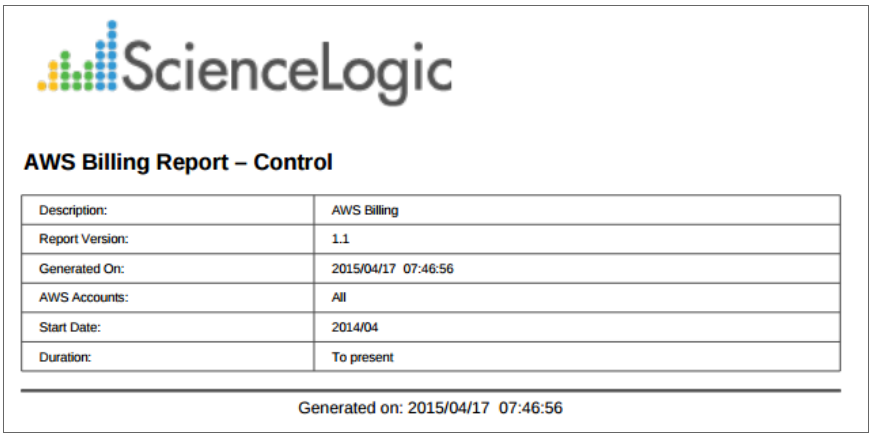
Generated on: 2015/04/17 07:46:56



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

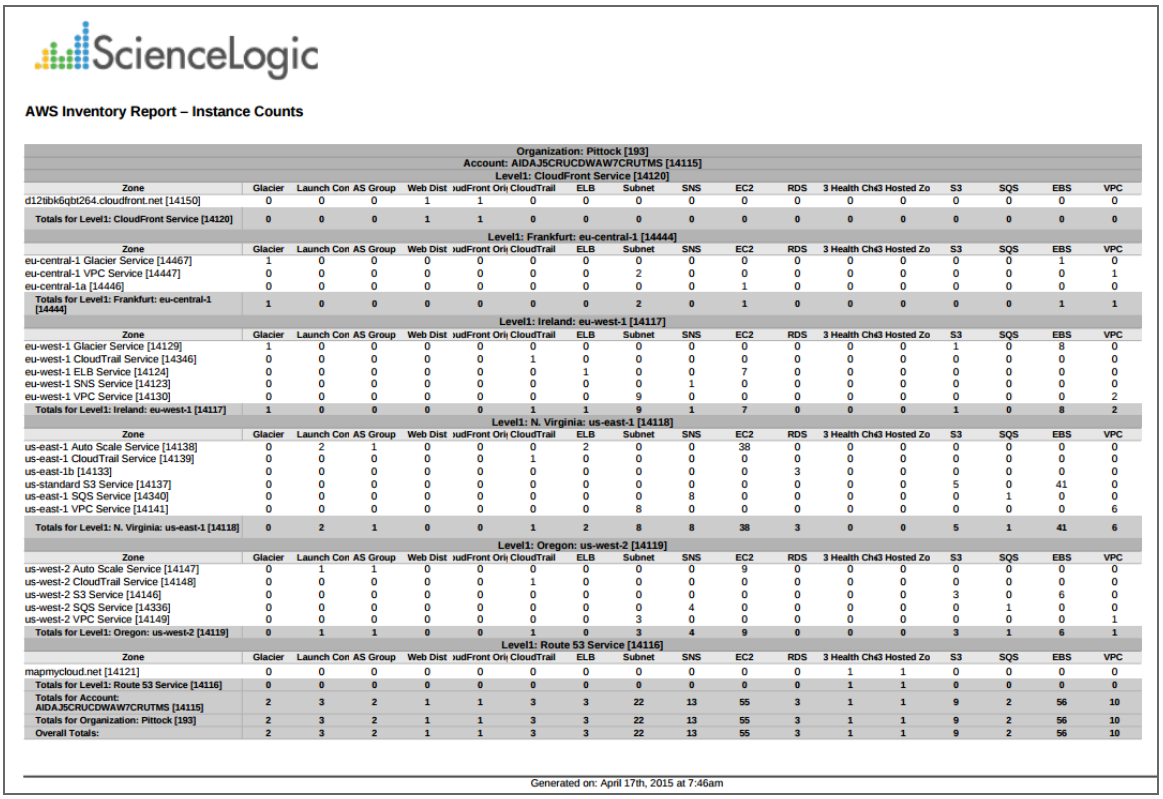
The following input options are available when generating the report:

- ***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.

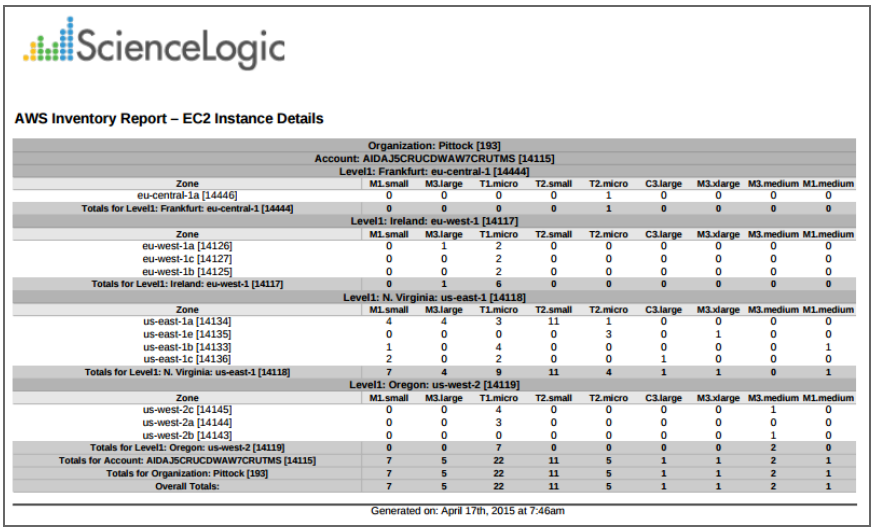


ScienceLogic

AWS Inventory Report – Instance Counts

Organization: Pittock [193]																	
Account: AIDAJ5CRUCDWAW7CRUTMS [14115]																	
Level1: CloudFront Service [14120]																	
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
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



ScienceLogic

AWS Inventory Report – EC2 Instance Details

Organization: Pittock [193]									
Account: AIDAJ5CRUCDWAW7CRUTMS [14115]									
Level1: Frankfurt: eu-central-1 [14444]									
Zone	M1.small	M3.large	T1.micro	T2.small	T2.micro	C3.large	M3.xlarge	M3.medium	M1.medium
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]									
Zone	M1.small	M3.large	T1.micro	T2.small	T2.micro	C3.large	M3.xlarge	M3.medium	M1.medium
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]									
Zone	M1.small	M3.large	T1.micro	T2.small	T2.micro	C3.large	M3.xlarge	M3.medium	M1.medium
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]									
Zone	M1.small	M3.large	T1.micro	T2.small	T2.micro	C3.large	M3.xlarge	M3.medium	M1.medium
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:

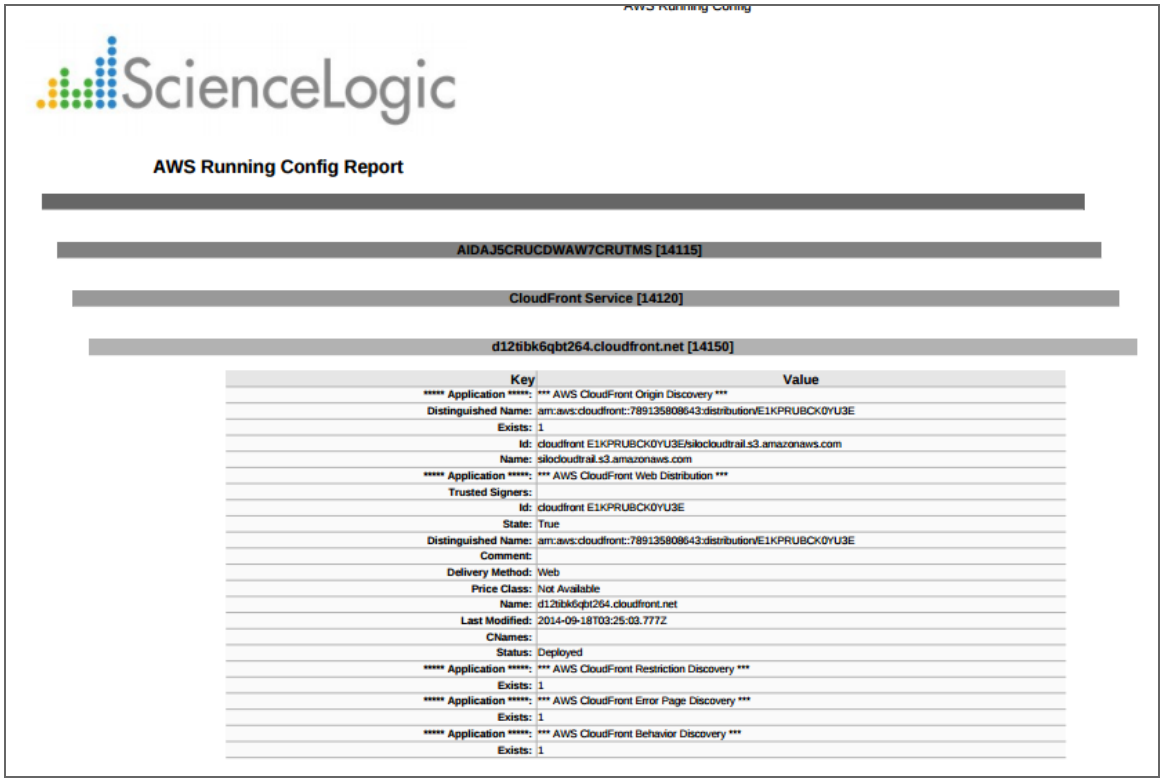
- ***Organizations***. Select the organization for which you want to generate the report. The *All Organizations* checkbox is selected by default. De-selecting this checkbox allows you to select one or more specific organizations for which to generate a report.

- ***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.



The following input options are available when generating the report:

- ***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 5

Dashboards

Overview

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

Installing the Amazon Web Services: Classic Dashboards PowerPack

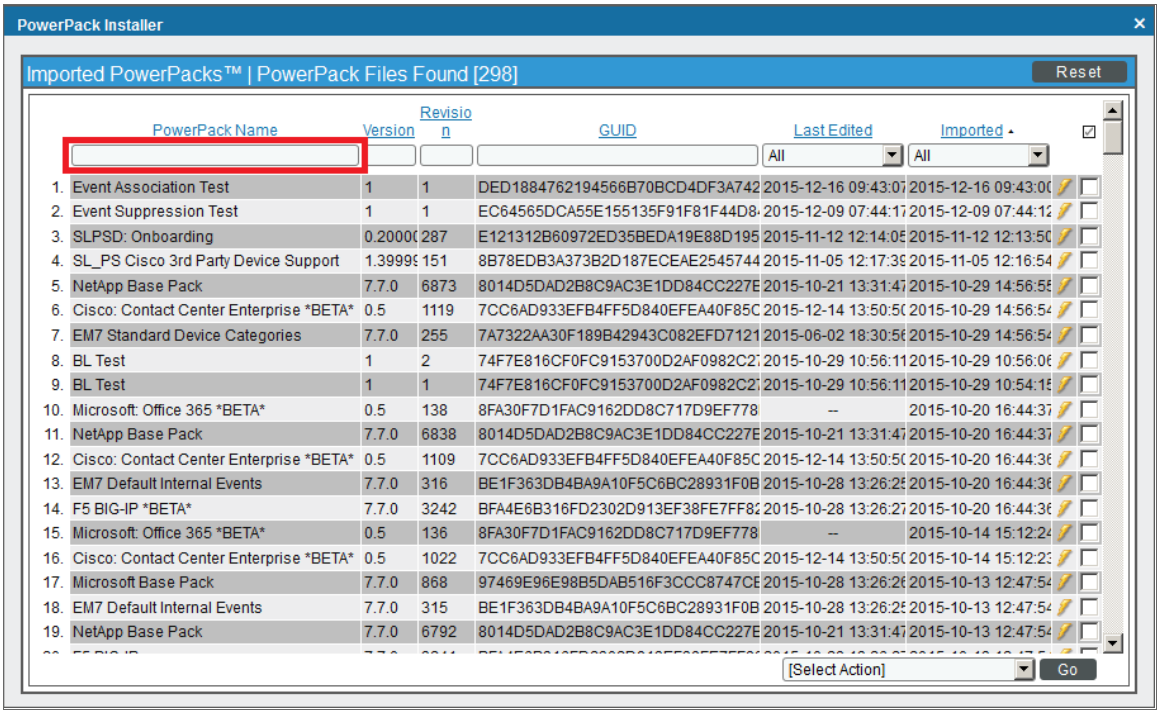
To view the Amazon Web Services dashboards in SL1, you must first install the *Amazon Web Services: Classic Dashboards* PowerPack.

> **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.

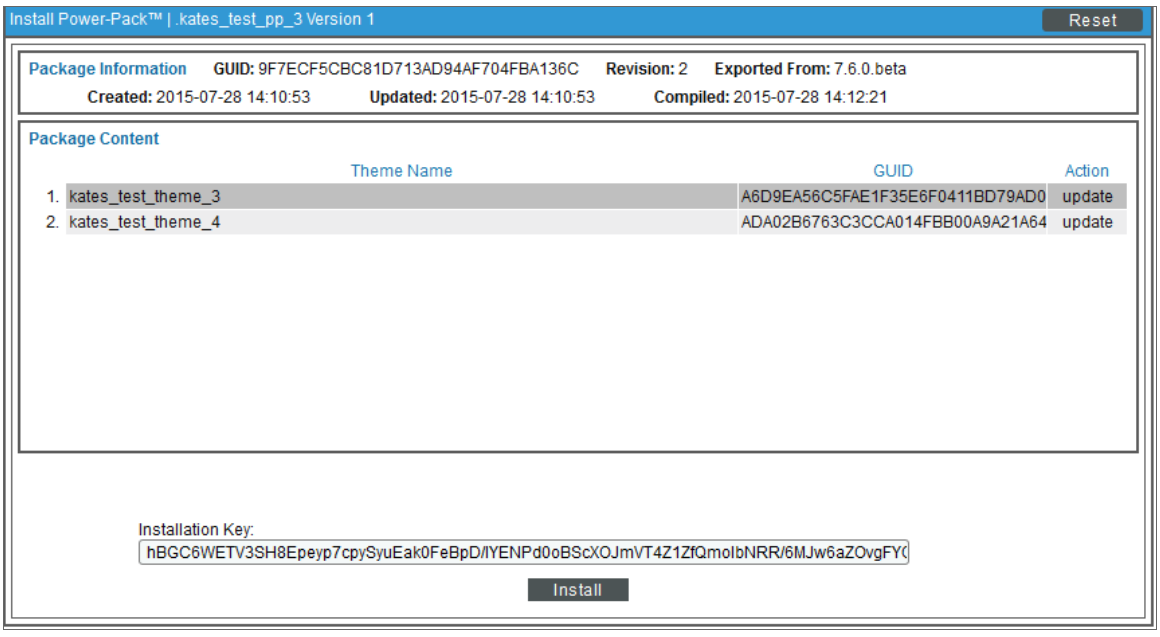
To install the PowerPack:

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.



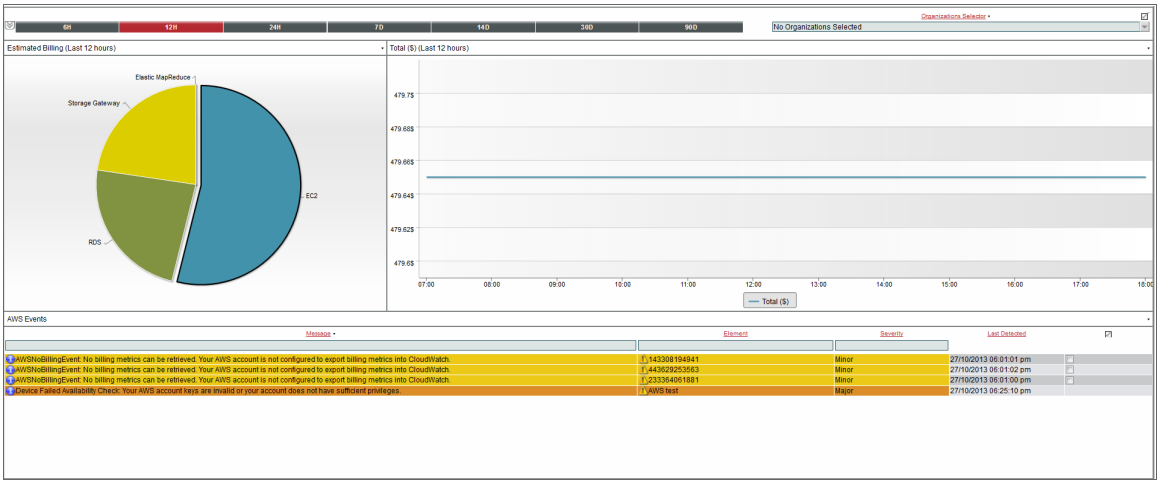
4. Click the lightning-bolt icon ( ) for the PowerPack that you want to install.
5. The **Install PowerPack** modal page appears. To install the PowerPack, click **[Install]**.



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

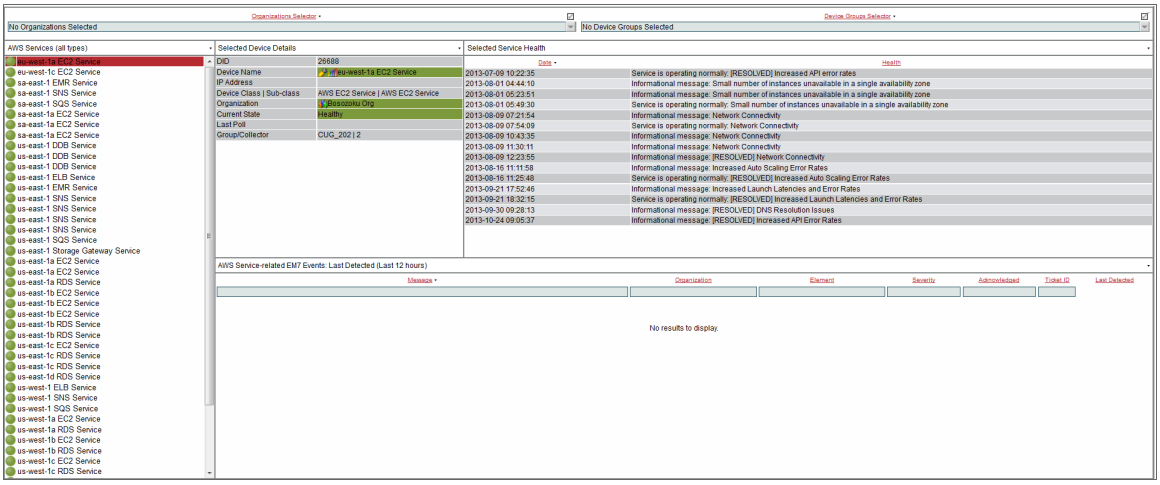
The AWS Account Billing Dashboard displays:



- 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.

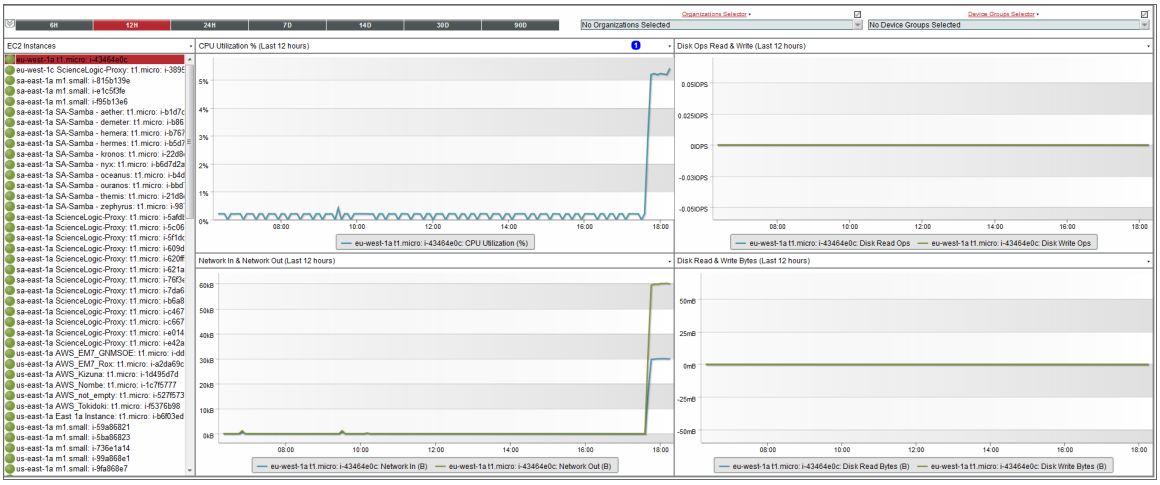
AWS Service Instance Performance Dashboards

The *Amazon Web Services: Classic 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 6

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

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

Policy Name	Result
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: 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: 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 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 EBS Instances by EC2 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 EBS Instances by EC2 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 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. 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 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 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 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. 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.

Vanishing Terminated or Terminating EC2 Instances

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" 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. 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.

© 2003 - 2019, ScienceLogic, Inc.

All rights reserved.

LIMITATION OF LIABILITY AND GENERAL DISCLAIMER

ALL INFORMATION AVAILABLE IN THIS GUIDE IS PROVIDED "AS IS," WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED. SCIENCELOGIC™ AND ITS SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT.

Although ScienceLogic™ has attempted to provide accurate information on this Site, information on this Site may contain inadvertent technical inaccuracies or typographical errors, and ScienceLogic™ assumes no responsibility for the accuracy of the information. Information may be changed or updated without notice. ScienceLogic™ may also make improvements and / or changes in the products or services described in this Site at any time without notice.

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