OneCloud Consulting
The Cloud Services Division of ePlus

Accelerate Cloud Business as a MSP (Managed Service Provider) with ScienceLogic and Cloudhealth Monitoring Tools.

AWS Monitoring/Management and Cost Optimization as MSP

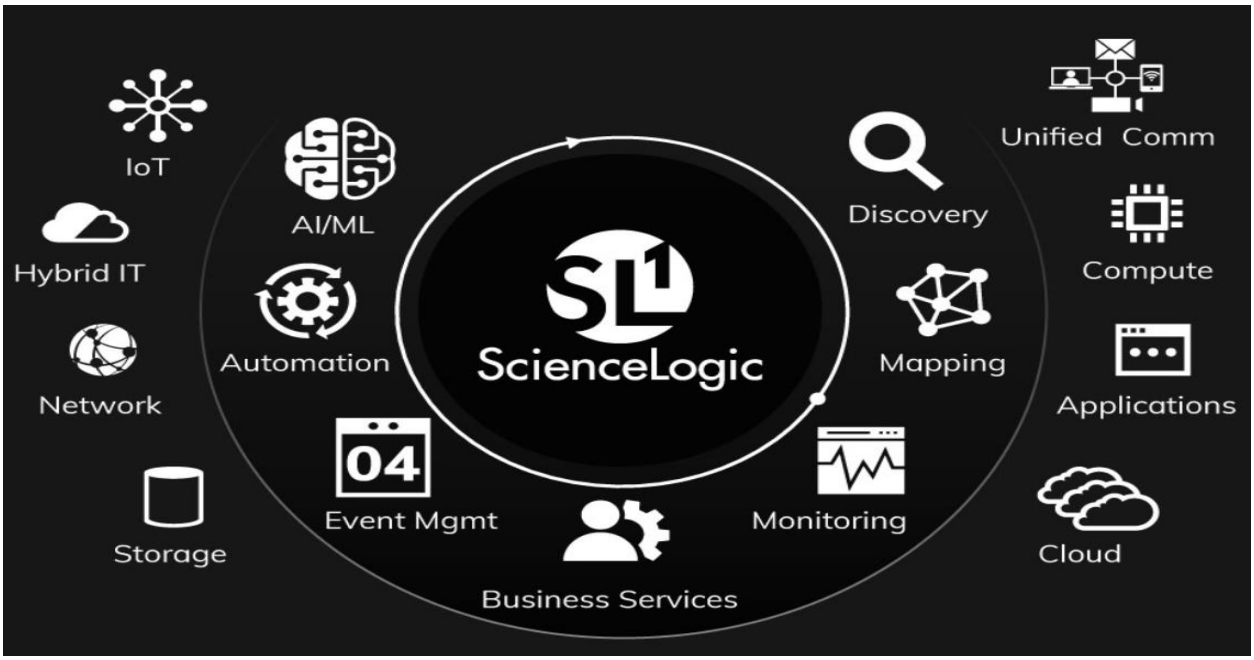
Revision History

Date	Comment	Owner
06/09/2018	Drafted first pass of document	Sahana Jayaram (sahana.jayaramu@eplus.com)

Table of Contents

1. ScienceLogic - Introduction

ScienceLogic is a leader in IT Operations Management, providing modern IT operations with actionable insights to resolve and predict problems faster in a digital, ephemeral world.



SCIENCELOGIC for Amazon Web Services (AWS) Management

https://sciencelogic.com/product/technologies/amazon-web-services

SCIENCELOGIC for Microsoft Azure Cloud Management

https://sciencelogic.com/product/technologies/microsoft/azure

1.1 ScienceLogic Capabilities and Functionalities in monitoring Cloud Platforms

- Gain Deep Visibility into AWS and Azure public Clouds.
- **Use a single platform to monitor everything, everywhere** - Automatically monitor your entire IT universe - on premises and in multiple clouds from a single console
- **Understand AWS, Azure Dependencies** – for entire IT universe in the cloud and on-premises.
- Automatically discover all of AWS and Azure resources and keep track of changes in the cloud environments.
- **Optimize AWS and Azure Investments** - by discovering what you have, what you use, and what it connects to; place workloads optimized for latency, security, availability, and costs.

- **Keep AWS & Azure Cloud resources healthy** with patented discovery, mapping, and pre-configured monitoring policies for AWS services and technologies; monitor additional AWS and Azure services and technologies with ease
- **Optimize investments in AWS & Azure cloud to Boost IT efficiency** by automating IT operational processes for both cloud and on-premises services.
- **Provide Role-Specific Visibility** into all of your AWS and Azure Environments, Public Cloud services and infrastructure, across all regions and zones with built-in, best practice-based dashboards.
- Build your own PowerPacks and custom dashboards with ease.
- **Troubleshoot & Resolve Issues** Quickly - Proactively detect and be alerted on configuration changes and performance issues.

1.2 ScienceLogic - Monitoring Amazon Web Services

The process of setting up a ScienceLogic appliance on an Amazon Web Services EC2 instance. An instance is a virtual server that resides in the AWS cloud.

To get access to the ScienceLogic, login into ScienceLogic customer Portal:

ScienceLogic Customer Portal

https://portal.sciencelogic.com/user/login?destination=portal

This section describes:

- ***How to get the ScienceLogic AMI***
- ***How to define an EC2 Instance from the ScienceLogic AMI***
- ***Assigning an optional Elastic IP Address (EIP)***
- ***Accessing the Appliance Using SSH***
- ***Rebooting Data Collectors and Message Collectors***
- ***Licensing and Configuring the new ScienceLogic Appliance(s)***

To monitor Amazon Web Services (AWS) in the ScienceLogic platform using the Amazon Web Services PowerPack.

https://docs.sciencelogic.com/8-9-0/Content/Web_Vendor_Specific_Monitoring/AWS/aws_title_page_web.htm?TocPath=Section%20IX.%20Vendor-specific%20Monitoring|Monitoring%20Amazon%20Web%20Services|_____0

1.3 ScienceLogic - Configuring Amazon Web Services for Monitoring

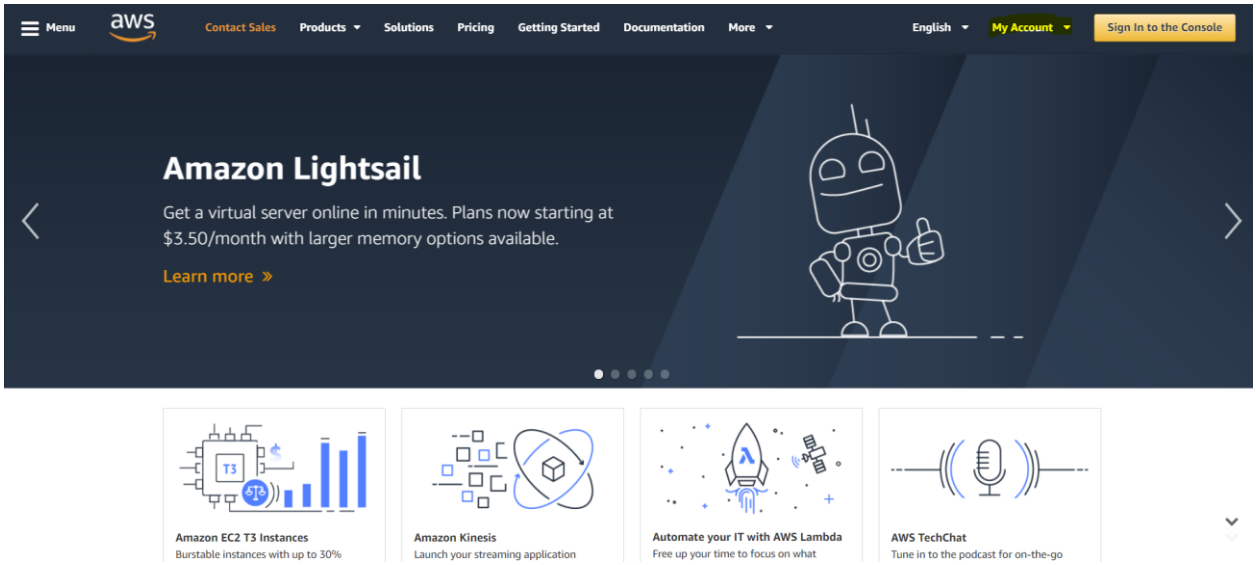
To use the AWS Dynamic Applications, you must configure a credential that allows the **ScienceLogic platform to connect to the AWS REST API**. The ***Amazon Web Services*** **PowerPack** includes three credential templates.

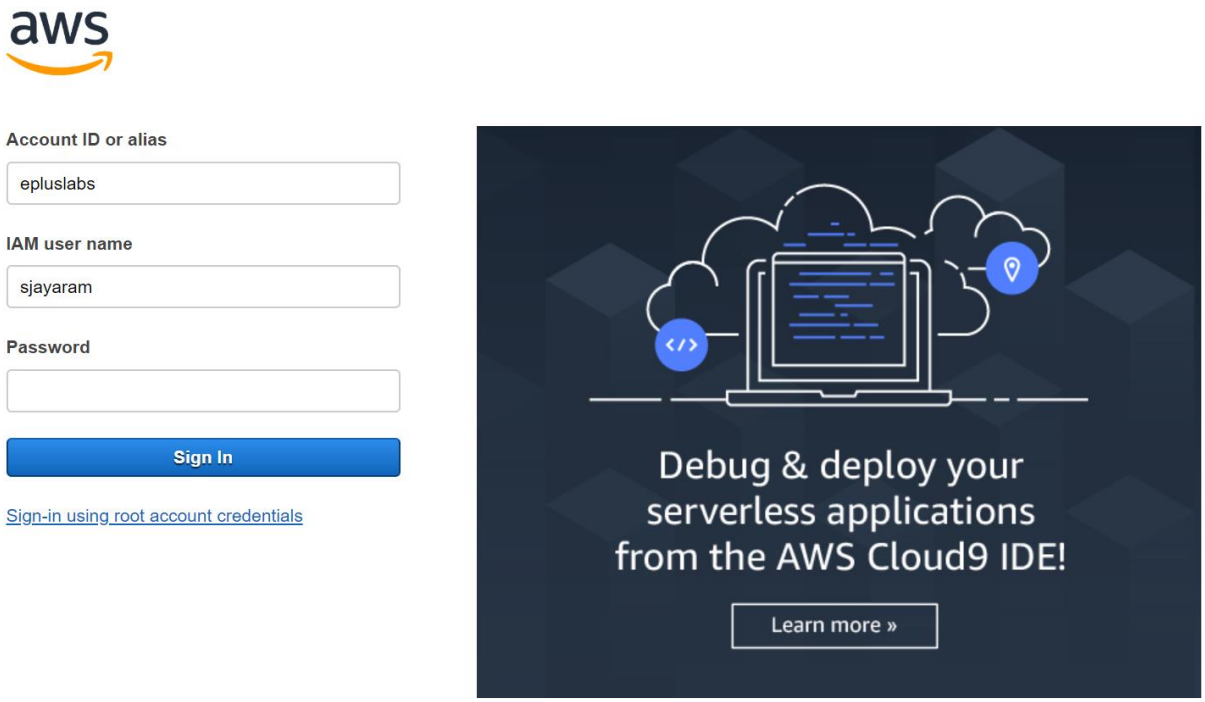
To use the credential templates included in the PowerPack, you must download the security credentials for a user associated with your AWS account. The user must meet the following requirements:

1. The Dynamic Applications in the *Amazon Web Services* PowerPack require, at minimum, the actions that are in the ReadOnlyAccess AWS Managed policy. To set this user policy, see https://console.aws.amazon.com/iam/home?region=us-east-1#/policies/arn:aws:iam::aws:policy/ReadOnlyAccess$serviceLevelSummary?section=policy_versions.
2. 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.
3. 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.
4. 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 the ScienceLogic platform, the device record for that instance will repeatedly switch between the component trees of the accounts that have visibility to that instance.

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

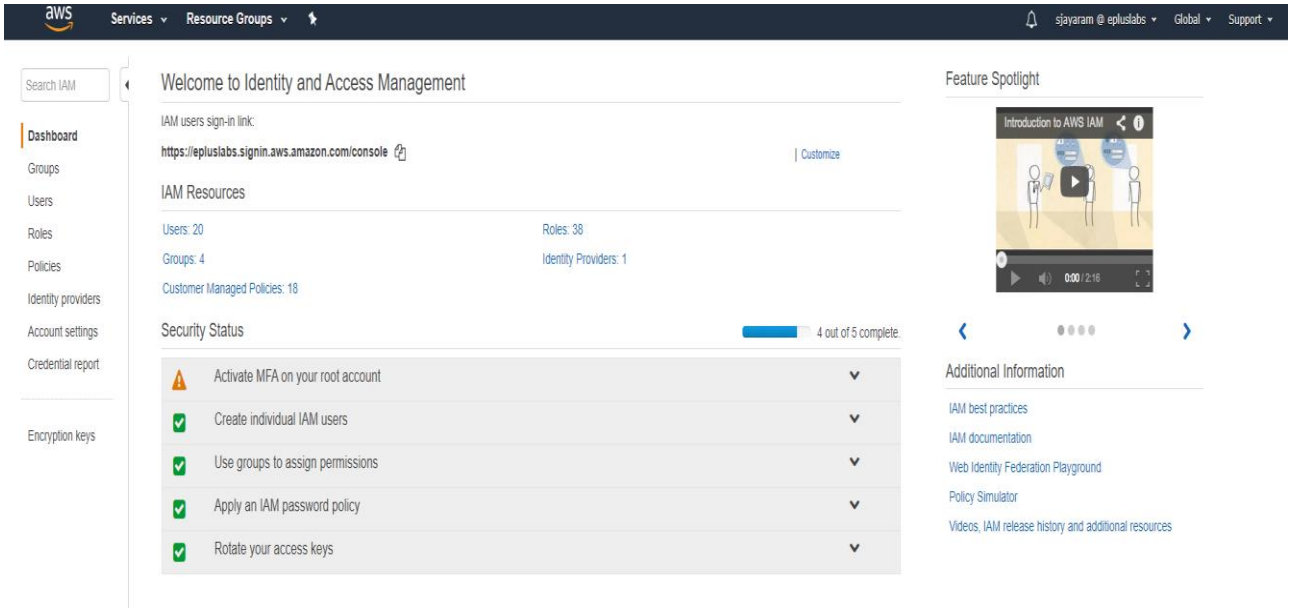
- Open a browser session and go to aws.amazon.com.
- Click **My Account** and then select *AWS Management Console*. If you are not currently logged in to the AWS site, you will be prompted to log in:





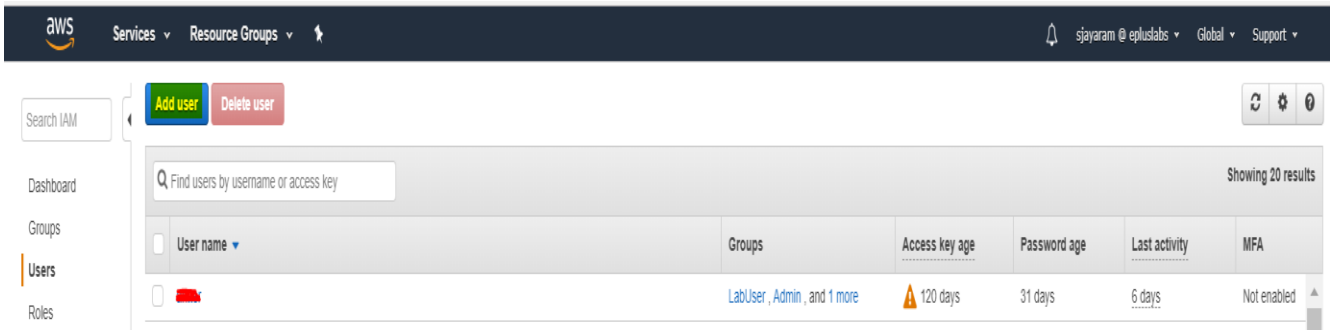
- In the **AWS Management Console**, under the **Services** heading, click **Identity & Access Management** (IAM).

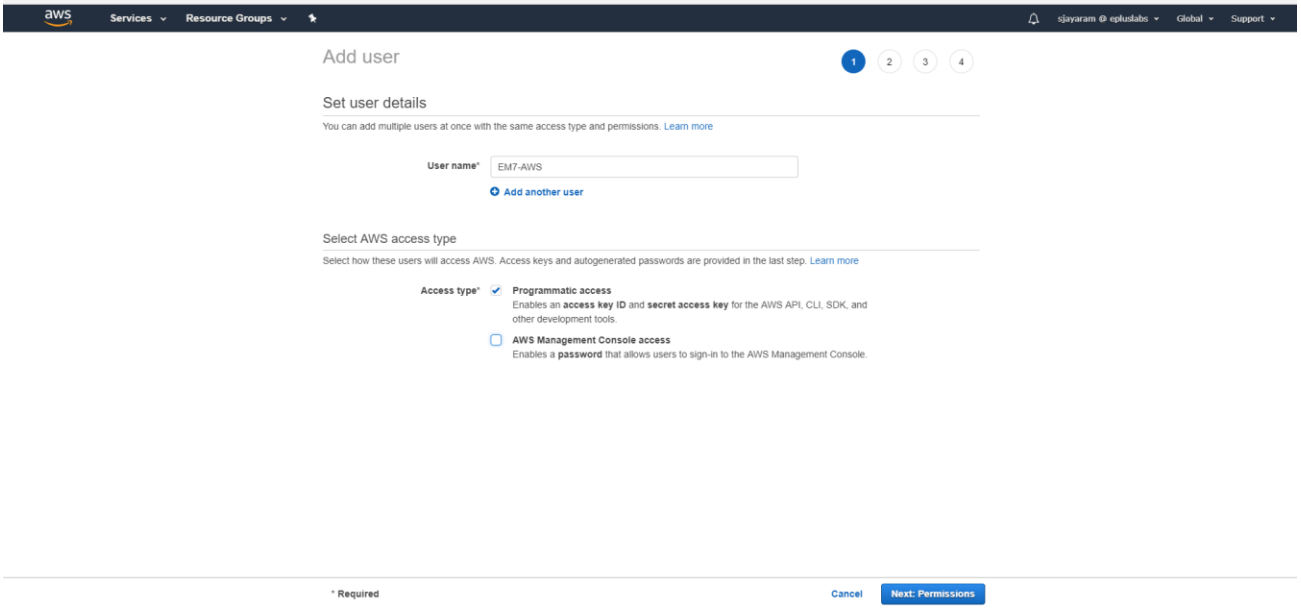
4. After logging in, the **Identity & Access Management Dashboard** page appears:



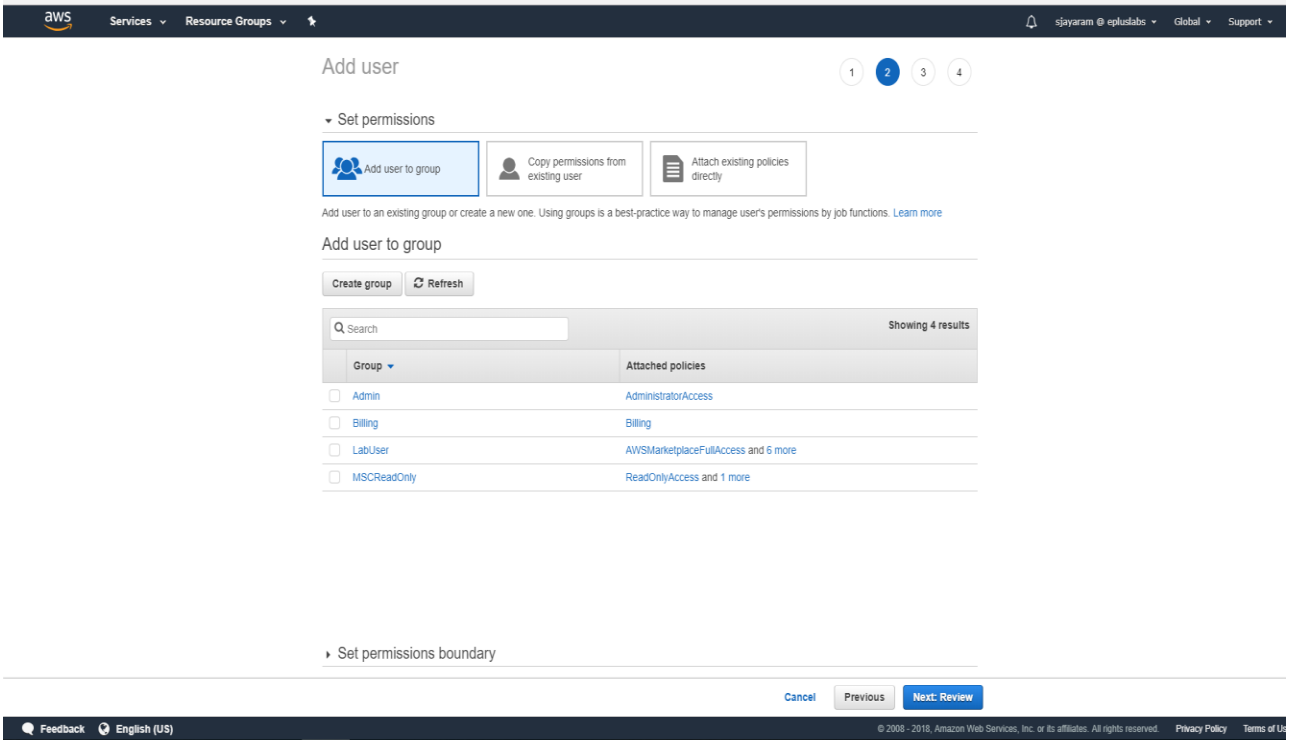
- To create a user account for the ScienceLogic platform, click **Users** on the Dashboard menu.

- Click the **Add User** button.
- Enter a username for the new user, e.g. "EM7-AWS", and make sure the ***Generate an access key for each user*** checkbox is selected.

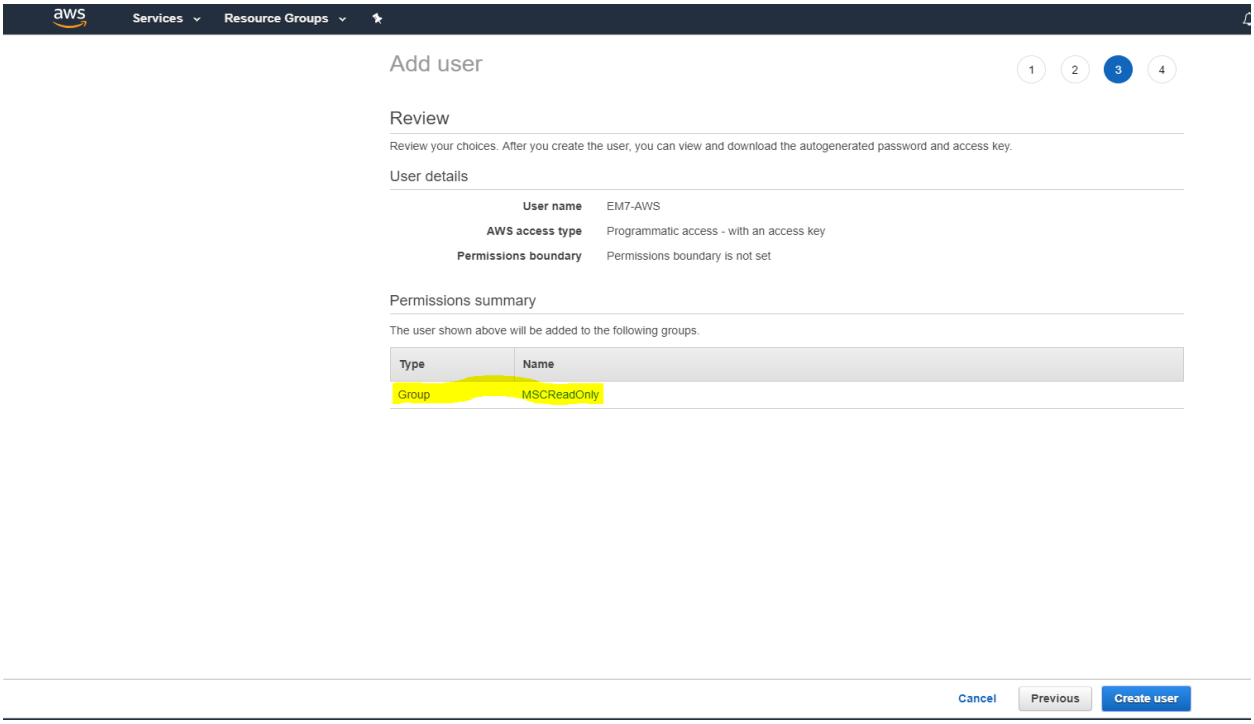




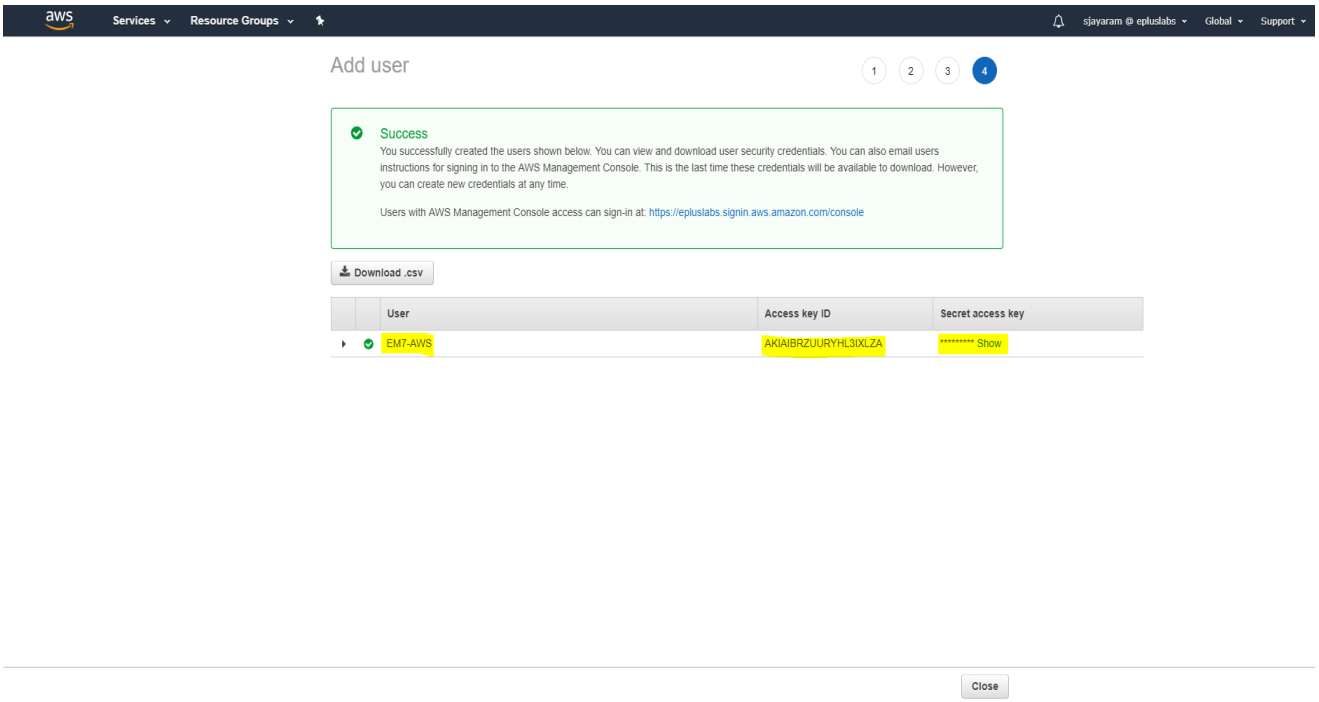
- Click **Next:Permissions** button, to set the ReadOnly permission for the user.



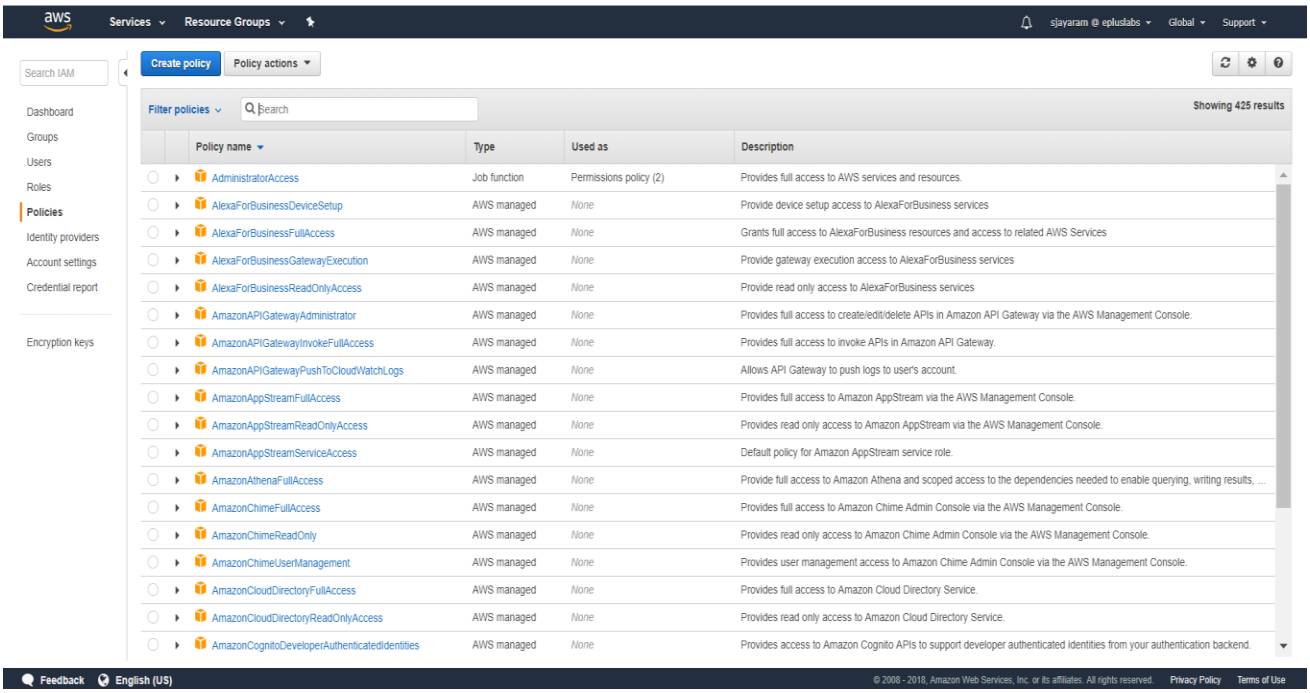
- Click the **Create** button to generate your user account. The **Create User** page appears:



- Click the **Download.csv** button to save your **Access Key ID and Secret Key as a CSV** (comma-separated value) text file, and then click **Close**.



- 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:
- Under the **Permissions** heading, click the **Attach Policy** button. The **Attach Policy** page appears:
- Select the checkbox for *Read Only Access* or select the policy based on the definition supplied by ScienceLogic.
- Click the **Attach Policy** button.



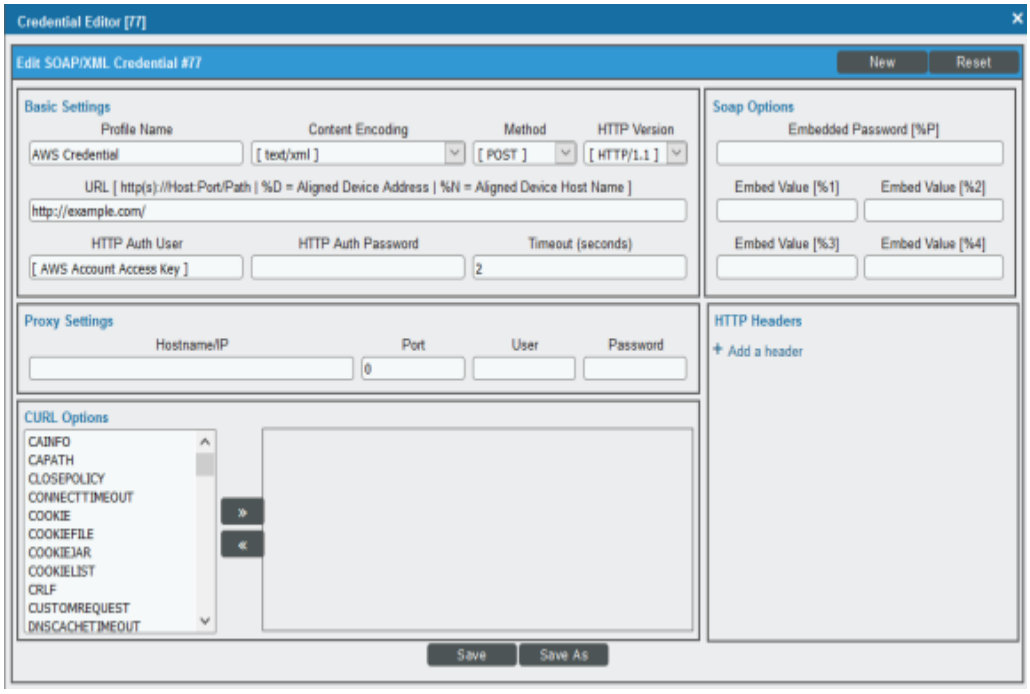
1.4 ScienceLogic - Creating an AWS Credential in ScienceLogic Platform

To use the Dynamic Applications in the *Amazon Web Services* PowerPack, you must first define an AWS credential in the ScienceLogic platform. 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 specific AWS account and region
- **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***.
   - **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.

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

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, the ScienceLogic platform cannot properly discover your AWS services.

1.5 ScienceLogic - Testing the AWS Credential in ScienceLogic Platform

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

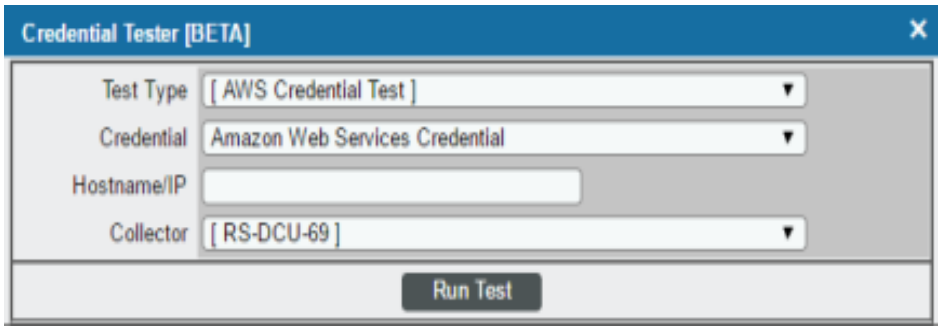
The AWS Credential Test can be used to test a **SOAP/XML** credential for monitoring AWS using the Dynamic Applications in the Amazon Web Services PowerPack. The AWS Credential Test performs the following steps:

- **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 service:** Verifies that the account specified in the credential has access to the ec2, iam, and s3 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".

1.6 ScienceLogic - Configuring AWS for Region-Specific Monitoring

You can 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.

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

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.

1.6.2 IAM JSON Policy Elements

An IAM policy is a JSON document that consists of one or more statements. Each statement is structured as follows:

Policy Syntax

```
{
 "Statement":[{
  "Effect":"effect",
  "Action":"action",
  "Resource":"arn",
  "Condition":{
   "condition":{
    "key":"value"
    }
   }
  }
 ]
}
```

- **Action:** The specific action or actions that will be allowed or denied.

You specify a value using a namespace that identifies a service (iam, ec2 sqs, sns, s3, etc.) followed by the name of the action to allow or deny.

For Examples

- Amazon EC2 action:

"Action": "ec2:StartInstances"

- IAM action

"Action": "iam:ChangePassword"

- **NotAction**: It is an advanced policy element that explicitly matches everything except the specified list of actions.

NotAction with Allow:

You can use the NotAction element in a statement with "Effect": "Allow" to provide access to all of the actions in an AWS service, except for the actions specified in NotAction.

For Example 1: Allow users to access all of the Amazon S3 actions that can be performed on any S3 resource except for deleting a bucket.

Note: This does not allow users to use the **ListAllMyBuckets S3 API** operation, because that action requires the **"*"** resource. This policy also does not allow actions in other services, because other service actions are not applicable to the S3 resources.

```
"Effect": "Allow",
"NotAction": "s3:DeleteBucket",
"Resource": "arn:aws:s3:::*",
```

Example 2: Allows users to access every action in every AWS service except for IAM.

```
"Effect": "Allow",
"NotAction": "iam:*",
"Resource": "*"
```

NotAction with Deny

You can use the NotAction element in a statement with "Effect": "Deny" to deny access to all of the listed resources except for the actions specified in the NotAction element.

Example 1: Conditional example denies access to non-IAM actions if the user is not signed in using MFA. If the user is signed in with MFA, then the "Condition" test fails and the final "Deny" statement has no effect.

```
"Version": "2012-10-17",
"Statement": [{
   "Sid": "DenyAllOutsideEU",
   "Effect": "Deny",
   "NotAction": "iam:*",
   "Resource": "*",
   "Condition": {"BoolIfExists": {"aws:MultiFactorAuthPresent": "false"}}
}]
```

- **NotResource**: It is an advanced policy element that explicitly matches everything except the specified list of resources.

For example, imagine you have a group named **HRPayroll**. Members of HRPayroll should not be allowed to access any **Amazon S3 resources** except the **Payroll folder** in the **HRBucket** bucket. The following policy explicitly denies access to all Amazon S3 resources other than the listed resources.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Deny",
    "Action": "s3:*",
    "NotResource": [
      "arn:aws:s3:::HRBucket/Payroll",
```

```
      "arn:aws:s3:::HRBucket/Payroll/*"
    ]
  }
}
```

1.6.3 Examples of region-specific JSON policies

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

```
{
"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 denies access to any operations outside of the **us-east-1, us-west-2, and ap-northeast-1** regions, except for actions in the listed services.
As a result, the ScienceLogic Platform will discover only components in the us-east-1, us-west-2, and ap-northeast-1 regions.

```
{
"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"]
}
}
}
]
}
```

Example 3: Allows Full EC2 Access Within a Specific Region

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Action": "ec2:*",
         "Resource": "*",
         "Effect": "Allow",
         "Condition": {
            "StringEquals": {
               "ec2:Region": "<REGION>"
            }
         }
      }
   ]
}
```

Example 4: Policy that allows all users Read-only access to a specific group, and allows only specific users access to make changes to the group.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "AllowAllUsersToListAllGroups",
         "Effect": "Allow",
         "Action": "iam:ListGroups",
         "Resource": "arn:aws:iam::*:*"
      },
      {
         "Sid": "AllowAllUsersToViewAndManageThisGroup",
         "Effect": "Allow",
         "Action": [
            "iam:CreateGroup",
            "iam:DeleteGroup",
            "iam:ListGroupPolicies",
```

```
            "iam:UpdateGroup",
            "iam:GetGroup",
            "iam:RemoveUserFromGroup",
            "iam:AddUserToGroup",
            "iam:ListGroupsForUser",
            "iam:AttachGroupPolicy",
            "iam:DetachGroupPolicy",
            "iam:ListAttachedGroupPolicies",
            "iam:GetGroupPolicy",
            "iam:DeleteGroupPolicy",
            "iam:PutGroupPolicy"
        ],
        "Resource": [
            "arn:aws:iam::*:user/*",
            "arn:aws:iam::*:group/AllUsers"
        ]
    },
    {
        "Sid": "LimitGroupManagementAccessToSpecificUsers",
        "Effect": "Deny",
        "Action": [
            "iam:CreateGroup",
            "iam:RemoveUserFromGroup",
            "iam:DeleteGroup",
            "iam:AttachGroupPolicy",
            "iam:UpdateGroup",
            "iam:DetachGroupPolicy",
            "iam:DeleteGroupPolicy",
            "iam:PutGroupPolicy"
        ],
        "Resource": "arn:aws:iam::*:group/AllUsers",
        "Condition": {
            "StringNotEquals": {
                "aws:username": [
                    "specialuser"
                ]
            }
        }
    }
  ]
}
```

Usecase Example: **AWS Windows EC2 instance automation.**

How EM7 can be configured to automatically:

- Create an EC2 instance device when an instance is spun up in AWS
- Discover the instance using PowerShell, based on an AWS tag used to identify the PowerShell credential to be used
- Merge the EC2 device with the PowerShell discovered device and set to the correct Windows device class

- Create a new dynamic device group, again based on a tag from AWS
- Finally, terminate the device in AWS and clean up the environment in EM7

Contains a set of Run-Book Actions, corresponding Automations, and an associated event and dynamic application.

PowerPack to automate common tasks when using Amazon Web Services EC2 Windows instances in an auto-scale group. This example was used in the October 2014 ScienceLogic Customer Symposium.

View on the ScienceLogic Customer Portal

1.7 ScienceLogic - Amazon Web Services PowerPack Monitor performance Metrics and collect configuration Data

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	CloudWatch	Elastic Beanstalk	Elastic Map Reduce (EMR)	OpsWorks	Route53	Simple Storage Service (S3)	Virtual Private Networks (VPN)
AutoScale	Direct Connect	Elastic Block Store (EBS)	Glacier		Security Groups	Storage Gateways (ASG)	-
CloudFront	DynamoDB (DDB)	Elastic Compute Cloud (EC2)	Lambda	RedShift	Simple Notification Service (SNS)	Storage Gateway Volumes	-
CloudTrail	ElastiCache	Elastic Load Balancers (ELB)	Lightsail	Relational Data Store (RDS)	Simple Queue Service (SQS)	Virtual Private Cloud Service (VPC)	-

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

- API Gateway private integrations
- CloudFront
- Replica Lambda functions

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
- 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

1.7.2 ScienceLogic - Configuring AWS to Report Billing Metrics

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

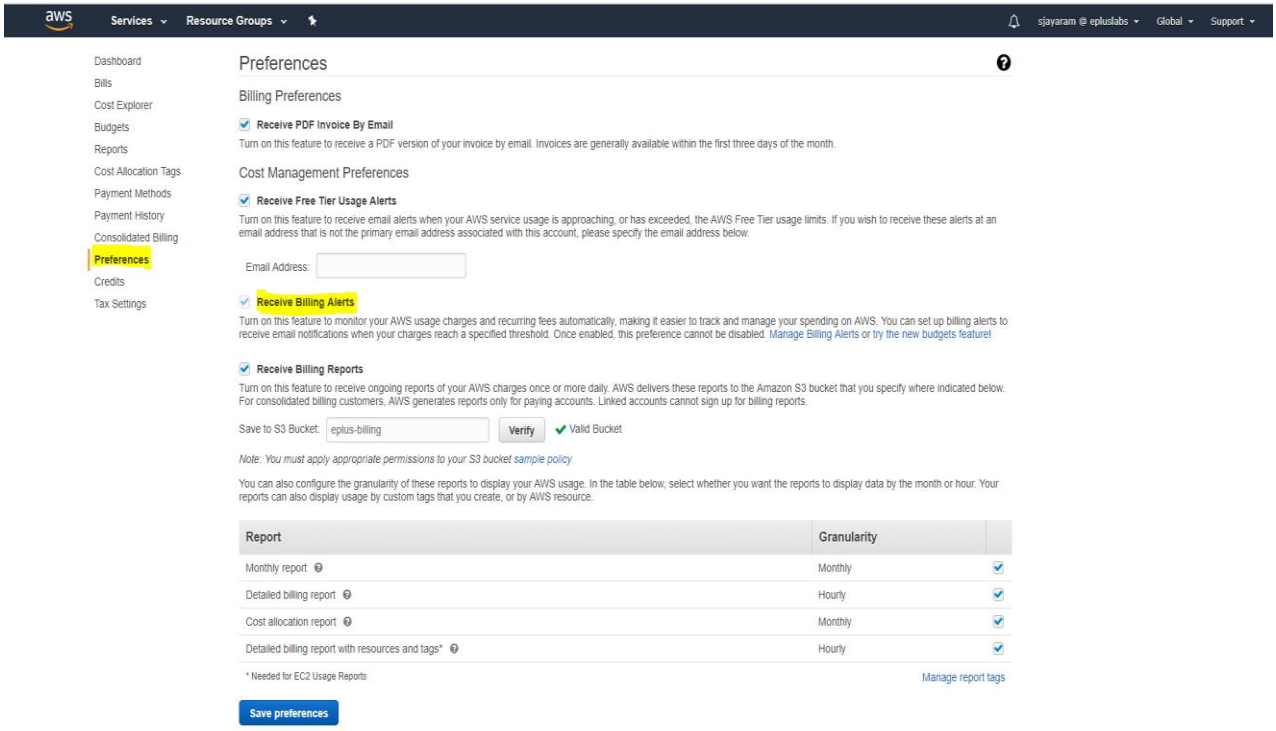
- The user account you supplied in the AWS credential must have permission to view the us-east-1 zone.
- Your AWS account must be configured to export billing metrics to the CloudWatch service.

Note: 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:

- Open a browser and login to AWS Management Console - aws.amazon.com.
- After logging in, the **Billing & Cost Management Dashboard** page appears. In the left navigation bar, click **Preferences**. The **Preferences** page appears:
- Select the ***Receive Billing Alerts*** checkbox.
- Click the **Save Preferences** button.



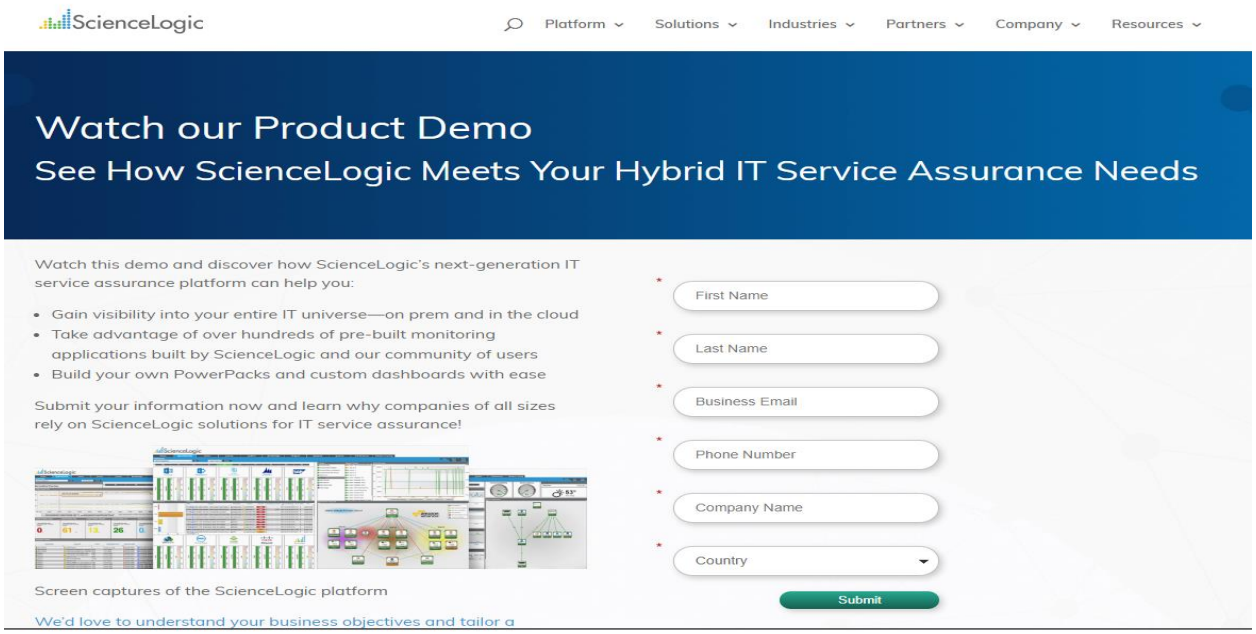
CAUTION: If you enable this option, this option cannot be disabled!

Reference - ScienceLogic Tool Exploration (Trail Version)

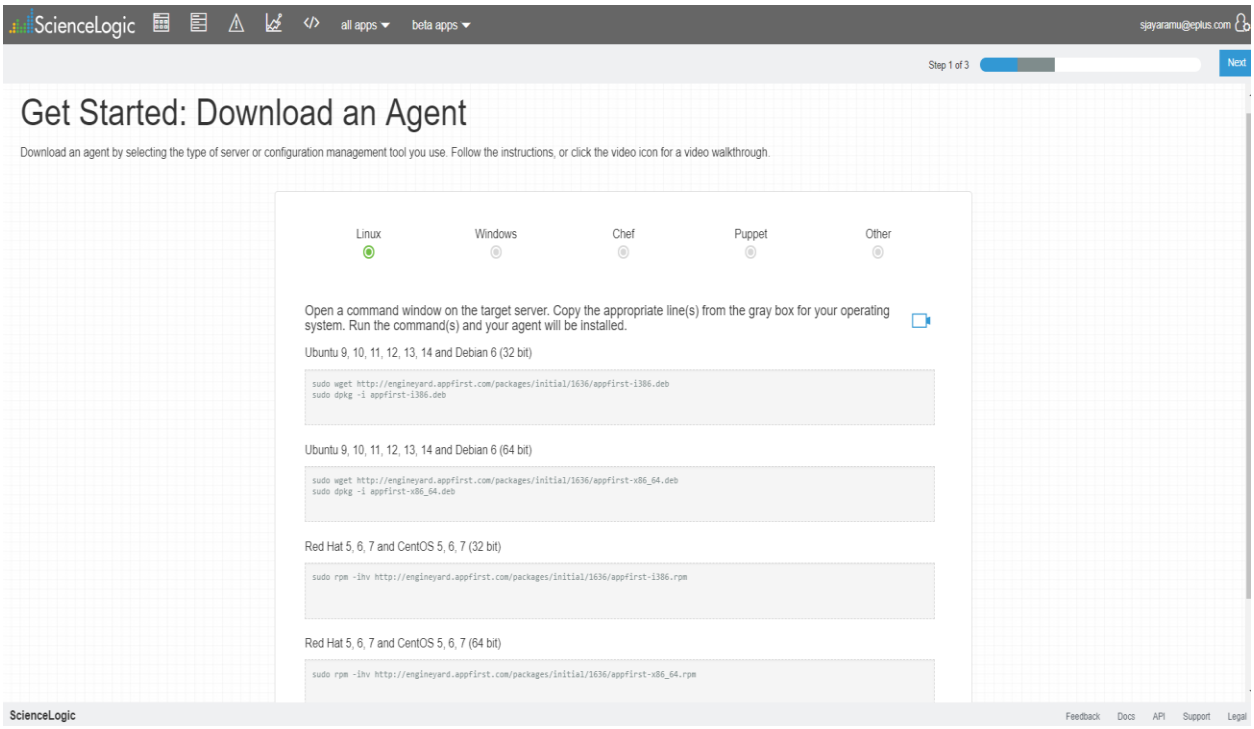
See everything across your entire ecosystem with a 30 day Free Trial. No credit card required.

Link- https://sciencelogic.com/watch-product-demo

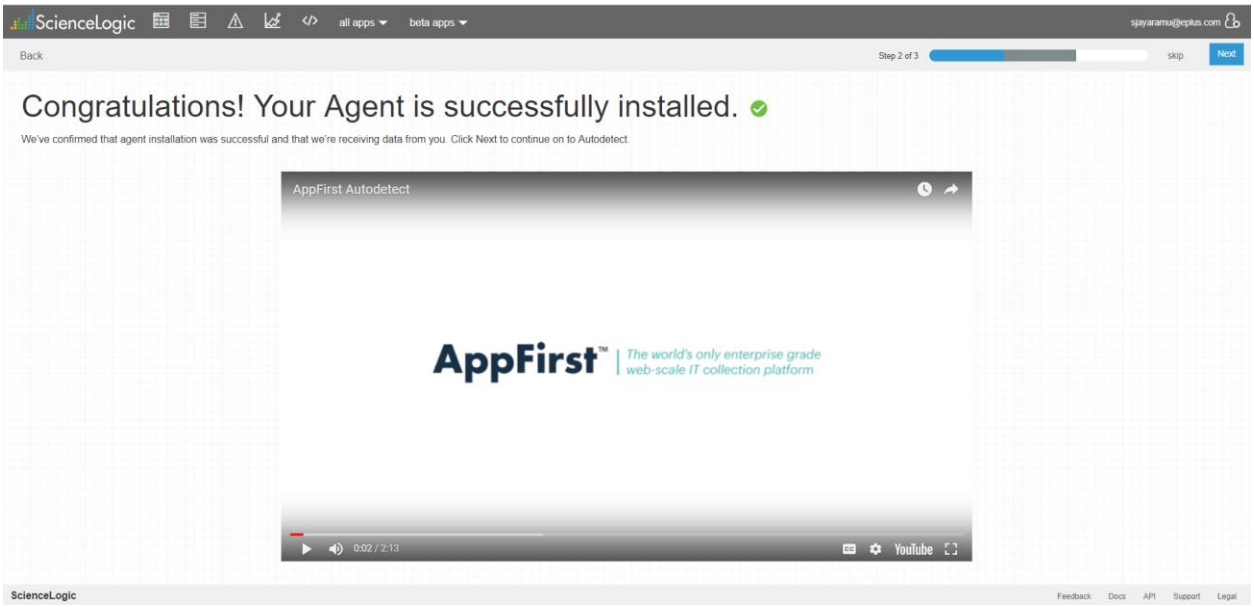
Register to ScienceLogic free trail version:



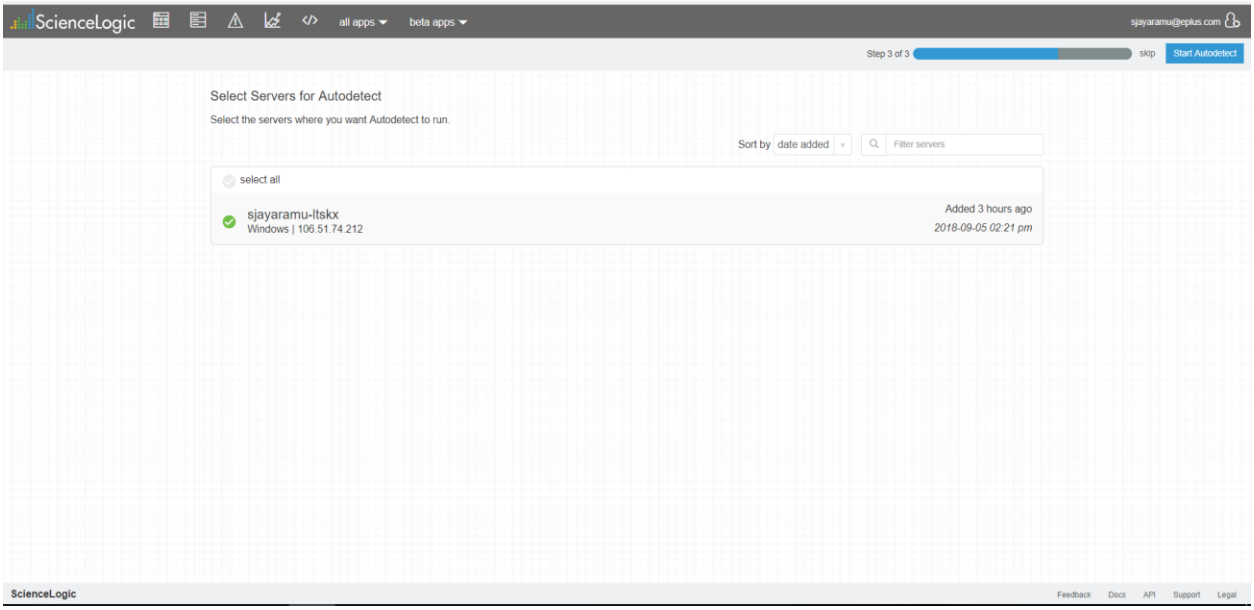
ScienceLogic - Download an Agent



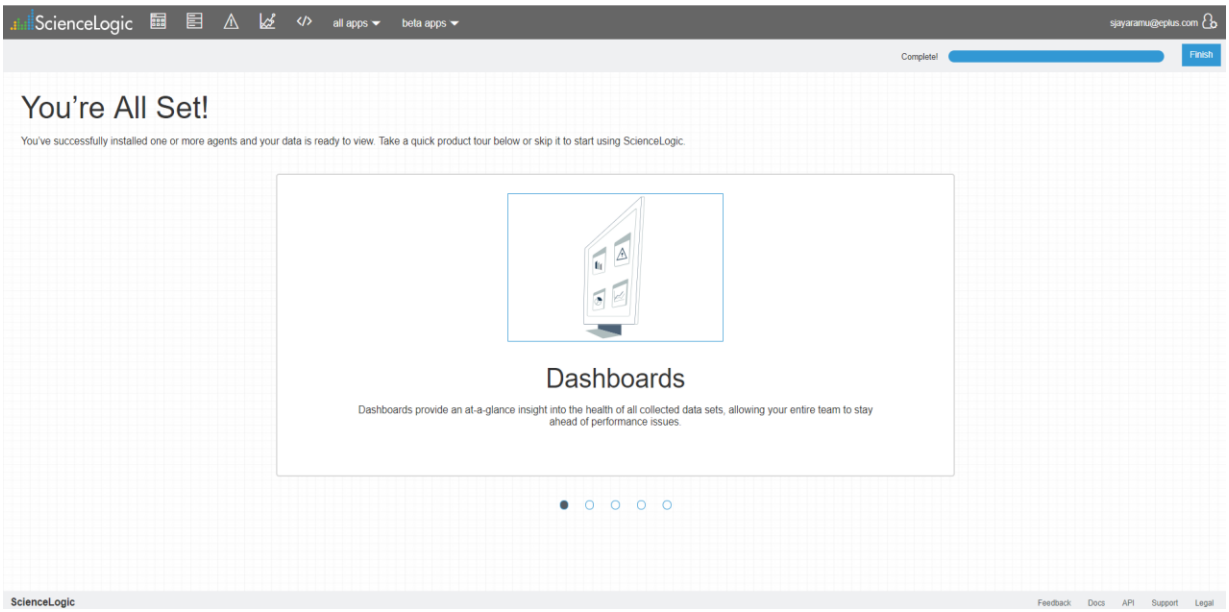
ScienceLogic – Agent installed successfully (Installed on windows)



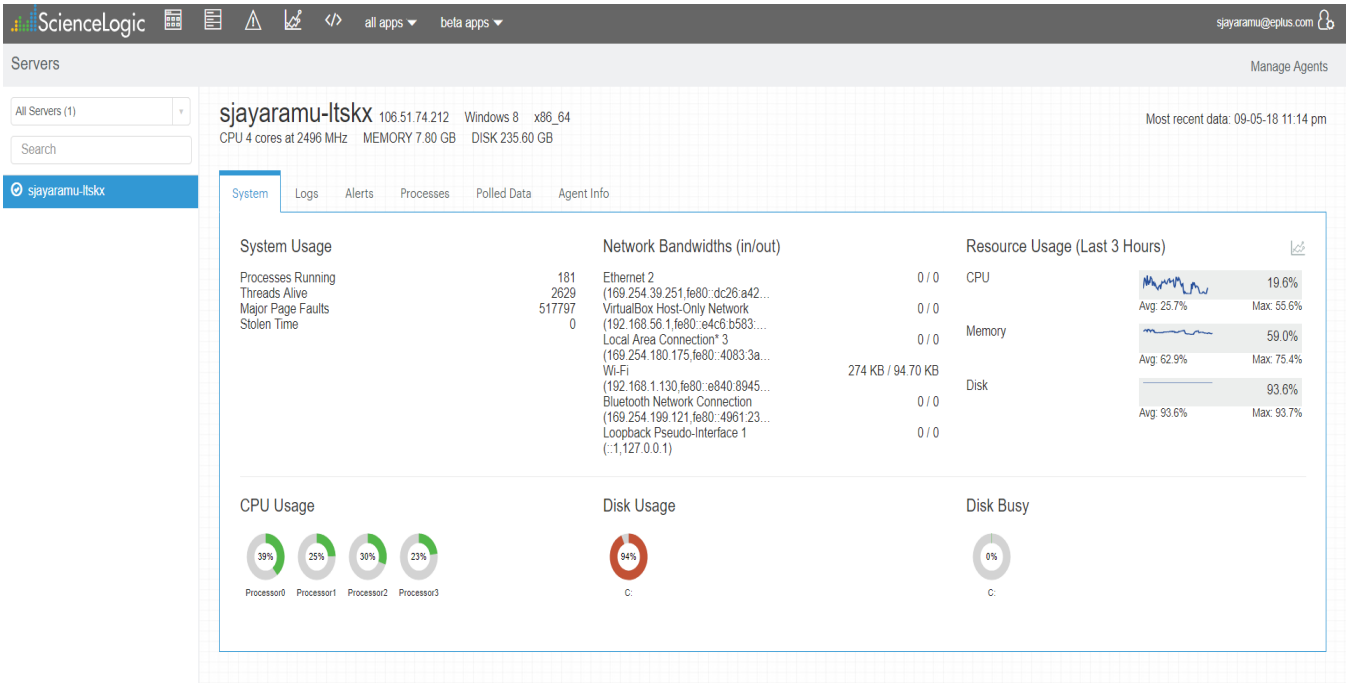
ScienceLogic – Select Server to AutoDetect (window Server – sjayaramu-ltskx)



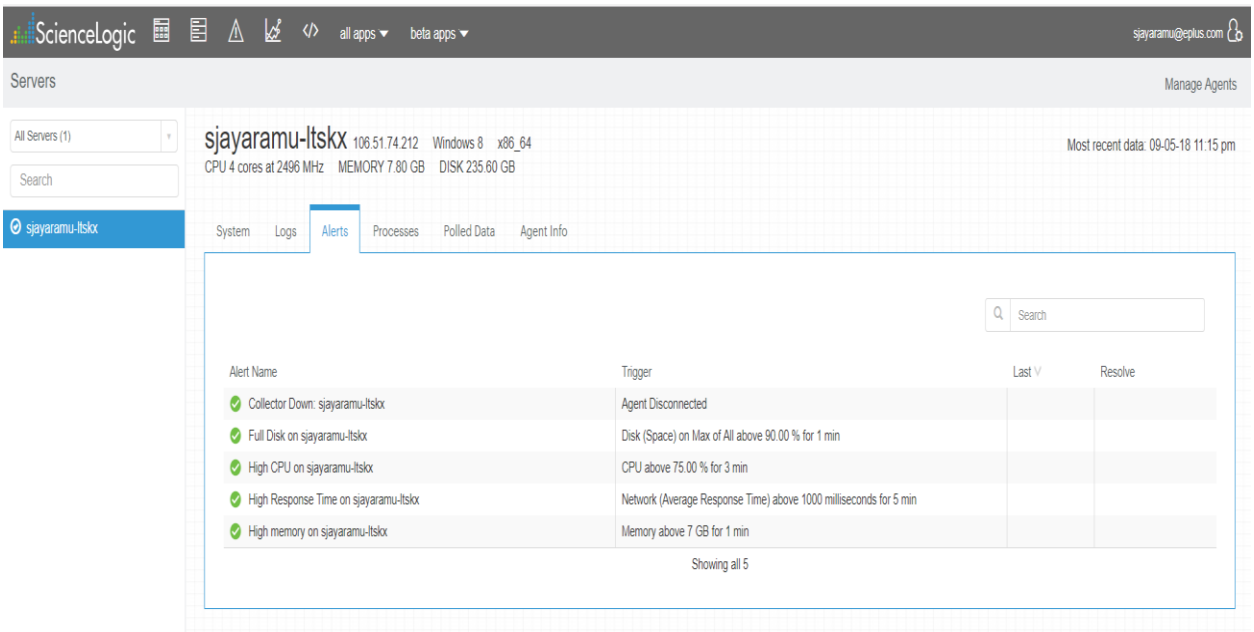
ScienceLogic – All Set with Dashboard!



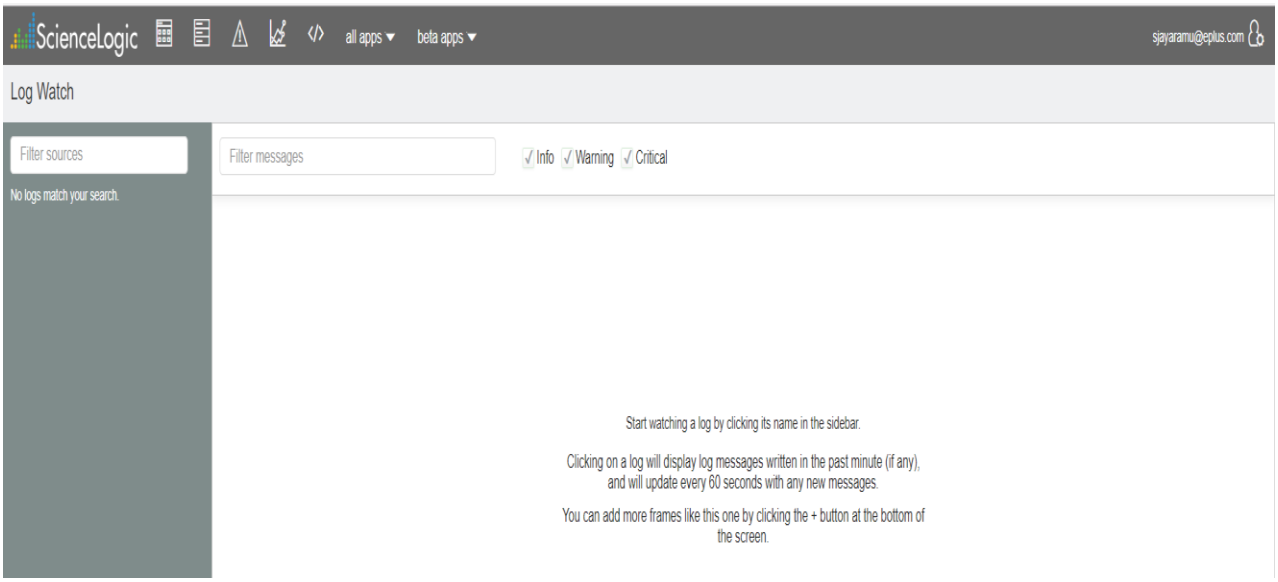
ScienceLogic – Under Servers, can view the system usage (CPU, Memory, Disk, Network)



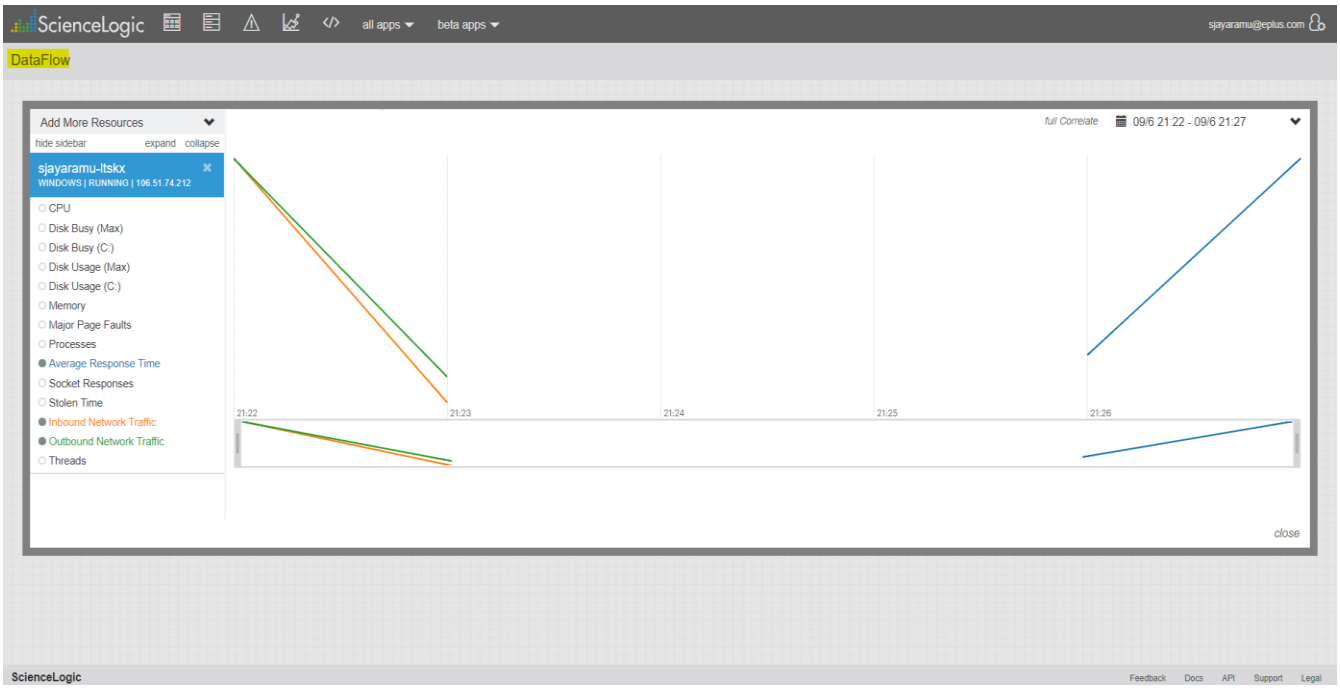
ScienceLogic – Under Servers, can view the Alerts.



ScienceLogic – Log watch.



DataFlow:



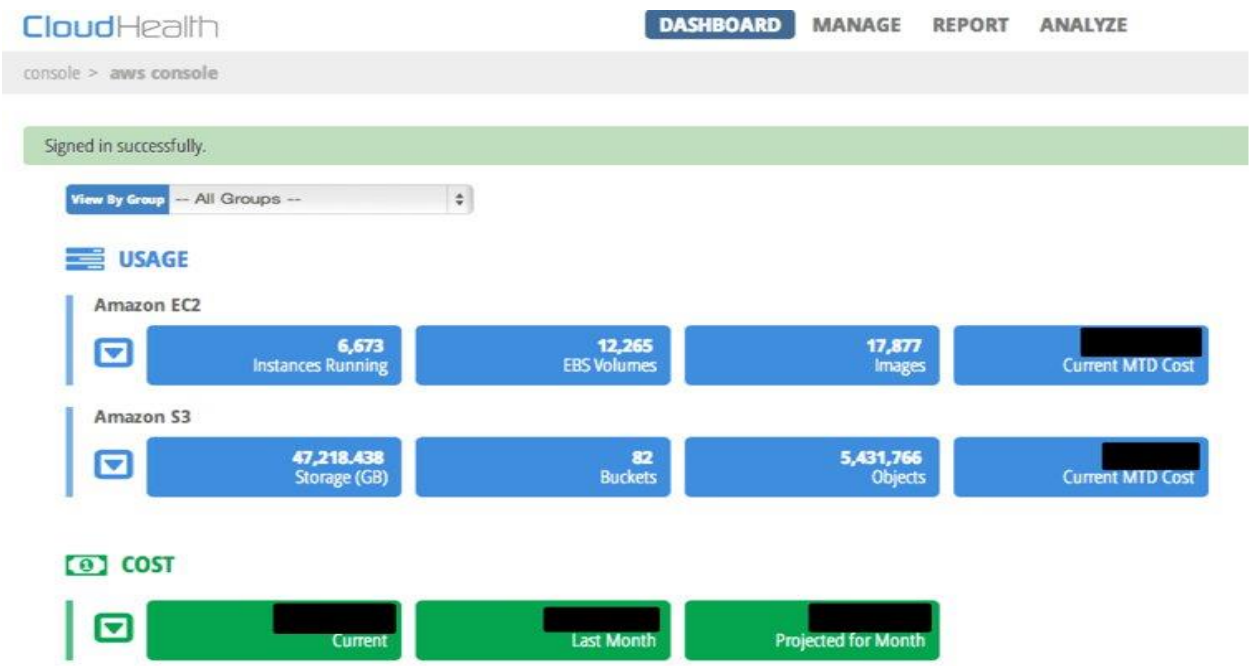
2. AMAZON WEB SERVICES AND CLOUDHEALTH

Efficiently scale your aws cloud by providing visibility into cost, configuration, usage, performance, and security, cloudhealth gives you a single pane of glass from which you can manage your aws cloud.

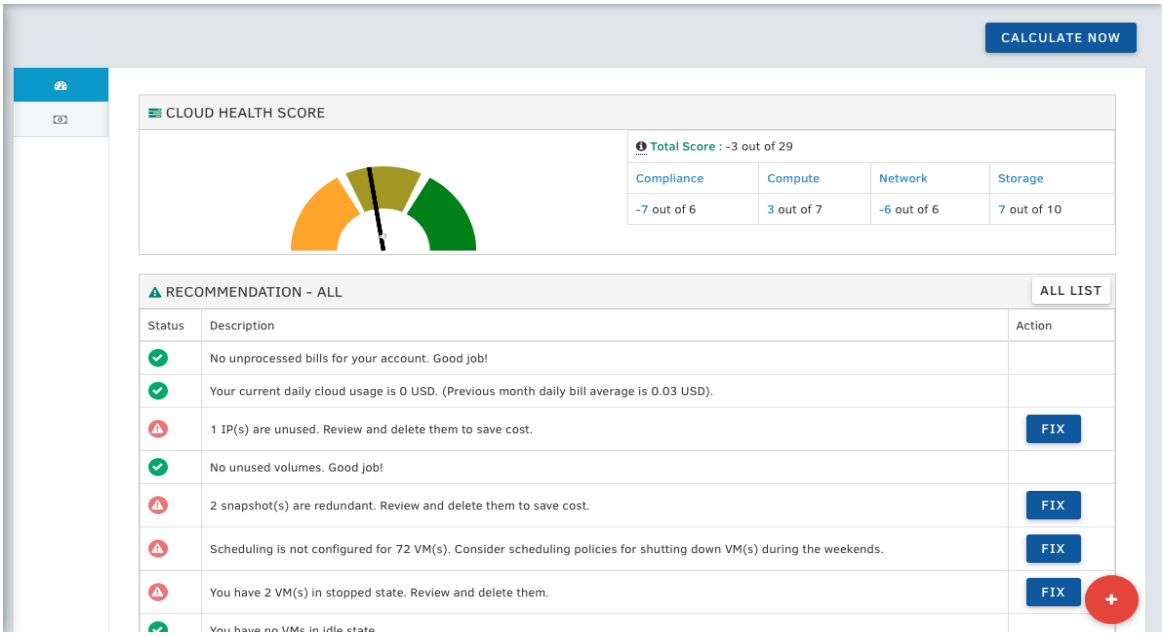
2.1 CLOUDHEALTH - Capabilities and Functionalities in monitoring Cloud Platforms

- Cost management with visibility into allocation, reallocation, budgeting and amortization to drive accountability
- Resource-based grouping for reporting, trending and Reserved Instances (RI) management to evaluate your AWS environment aligned to your business
- RI management with automated modifications, purchases, and utilization tracking
- Automated actions and policies for security, cost, performance, and configuration management
- EC2 Instance and EBS Volume Rightsizing and recommendations, based on CPU, memory, disk, and network data collected from agents and partners
- Security policies to continuously monitor your AWS environment for potential vulnerabilities, based on AWS and Center for Internet Security (CIS) best practices.

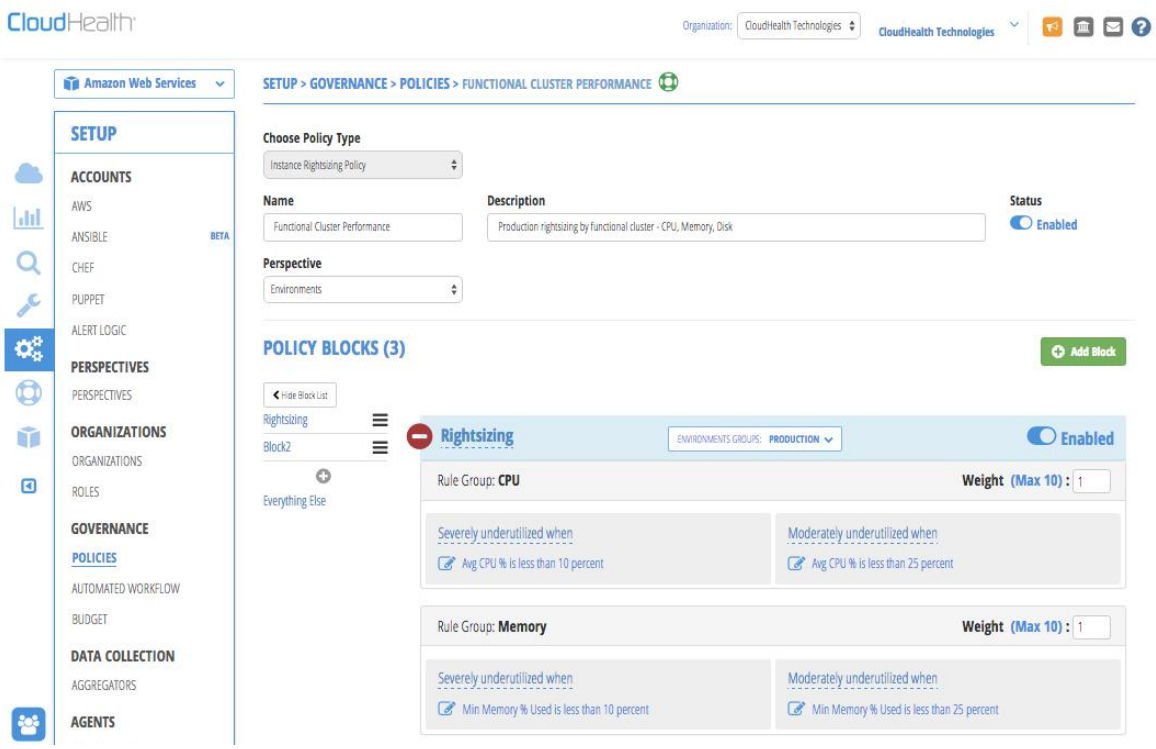
2.3 Cloudhealth Dashboard View:



AWS Health Score (Example)



Cloudhealth Policies:



Pricing Information

Below are the total costs for these different subscription durations. Additional taxes may apply.

CloudHealth Cloud Service Management	
Units	**12 MONTHS**
CHT100KAWSSpend	$37500
CHT250KAWSSpend	$77625
CHT500KAWSSpend	$138000
CHT750KAWSSpend	$18112

3 Ways To Reduce Cloud Spend in AWS:

- Use Reserved Instances and keep them optimized
- Continuously rightsize infrastructure
- Eliminate zombies instances

Reference Links: https://www.cloudhealthtech.com/blog/3-ways-reduce-cloud-spend-aws

10 Best Practices for Reducing Spend in AWS:

- Terminate Zombie Assets
- Rightsize EC2 Instances & EBS Volumes
- Upgrade instances to the latest generation
- Delete Disassociated Elastic IP Addresses

Link: http://go.cloudhealthtech.com/rs/933-ZUR-080/images/eBook_10%20Best%20Practices%20for%20Reducing%20Spend%20in%20AWS.pdf

Four Phases of Cloud Optimization:

Webinar: http://go.cloudhealthtech.com/thanks-wc-recording-4-phases-cloud-optimization.html?aliId=25992722

The Ultimate Guide to Amazon EC2 Reserved Instances (PDF)

http://go.cloudhealthtech.com/rs/933-ZUR-080/images/The%20Ultimate%20Guide%20to%20AWS%20EC2%20Reserved%20Instances.pdf

RESOURCE UTILIZATION (Track and Manage Resource Utilization in the Cloud)

Link: https://www.cloudhealthtech.com/solutions/increase-cloud-resource-utilization

https://www.cloudhealthtech.com/blog/3-rs-lowering-aws-costs

https://www.cloudhealthtech.com/aws-cost-optimization

Gains Visibility and cost under Control with CloudHealth
https://www.cloudhealthtech.com/sites/default/files/case-study-ogangi.pdf