

The background features a vibrant, abstract design with a color gradient from dark blue on the left to bright yellow and white on the right. The design consists of overlapping, wavy horizontal bands and a radial pattern of lines emanating from a bright white point on the right side, creating a sense of motion and energy.

CISCO *Live!*

Let's go



The bridge to possible

Cloud Infrastructure Observability using APIs, Terraform and Workflows

Prathima Janakiram, Cloud Developer Advocate
@patenclave

Agenda

- Challenges of Cloud Observability
- Overview of Cisco's Cloud Observability Platform
- Setting up Cloud Observability with API's
- Setting up Cloud Observability with Terraform
- Setting up Cloud Observability with Intersight ICO Workflows
- Lab
- Additional Resources

CISCO *Live!*

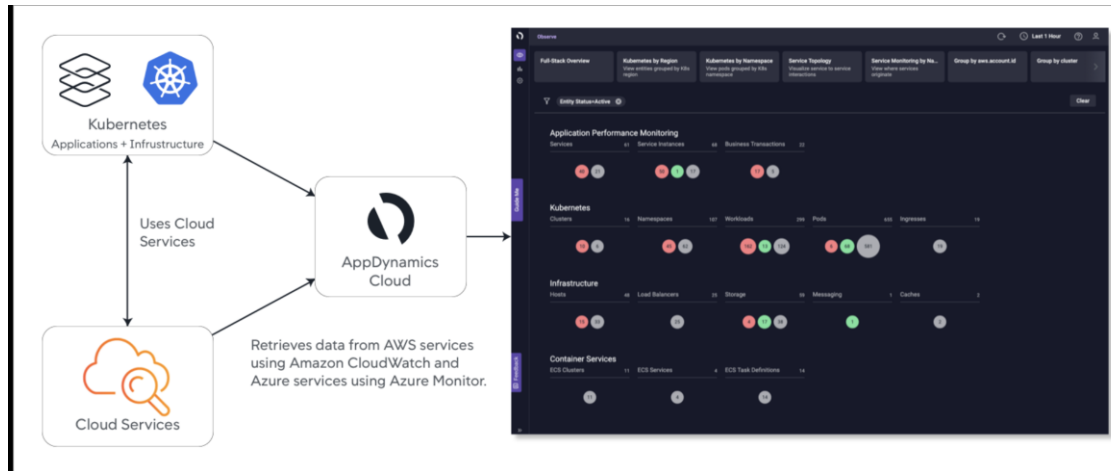


Challenges of Cloud Observability

- When Frost & Sullivan surveyed U.S.-based decision makers, it learned that cloud was top of mind for that population. 92% agreed that the cloud was “the most critical part” of their digital transformation strategy.
- Good Job! You may be one of the 92% and have decided to move your workloads to the cloud.
- But, how do you address obtaining the right kinds and levels of visibility and understanding? This is where **Cloud Observability** comes in.
- At its core, cloud observability means proactively monitoring the availability, performance and security of cloud-based IT infrastructures. This is the basis from which visibility into applications and underlying cloud infrastructure comes, including databases and networking.
- Cloud native observability means measuring and tracking a new, modern breed of applications built using cloud provider application services – such as those from Amazon Web Services (AWS), Microsoft Azure and Google Cloud Platform (GCP) – along with modern container based technologies, especially Kubernetes.
- Cloud native observability needs a platform to provide visibility and understanding of cloud assets – that’s where Cisco Cloud Observability Platform comes in.

Cisco Cloud Observability Platform

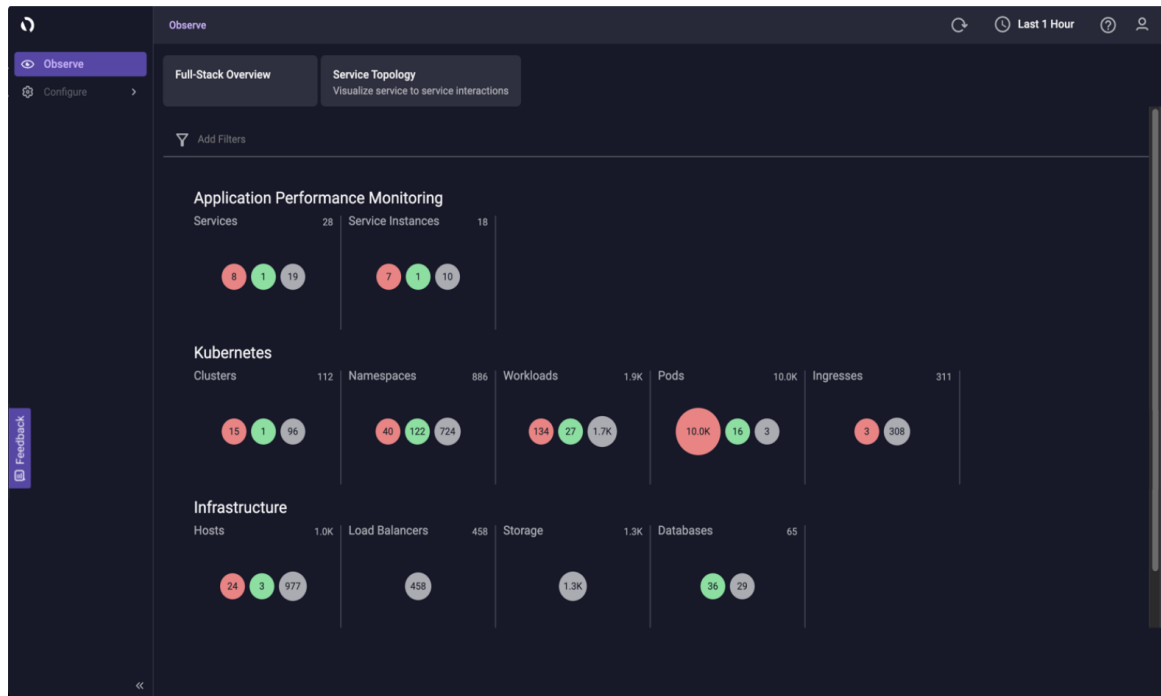
- Cisco Cloud Observability Platform is a software-as-a-service (SaaS) product that offers cloud native and full-stack observability for large, managed Kubernetes deployments on public clouds (Amazon Web Services, Microsoft Azure, and GCP). It provides real-time observability across your entire technology stack: applications, software-defined compute, storage, services, network, and other infrastructure, through the collection and correlation of metrics, events, logs, and traces (MELT).
- Single Observability platform with single-pane-of-glass visibility into their various applications running multi cloud.
- Cisco Cloud Observability built on Cisco Observability Platform is our Application Performance Monitoring solution for cloud native applications. It helps you to monitor cloud-native applications with business context. Ensure performance, prioritize and reduce risk, and optimize workloads.



Observability Cloud Stack

Cisco's Cloud Observability Platform collects and analyzes a broad spectrum of data, including application health and performance, business metrics like conversion rates, user experience mapping, and infrastructure and network telemetry — to resolve issues **before** they impact business KPIs. It comprises three layers of observability:

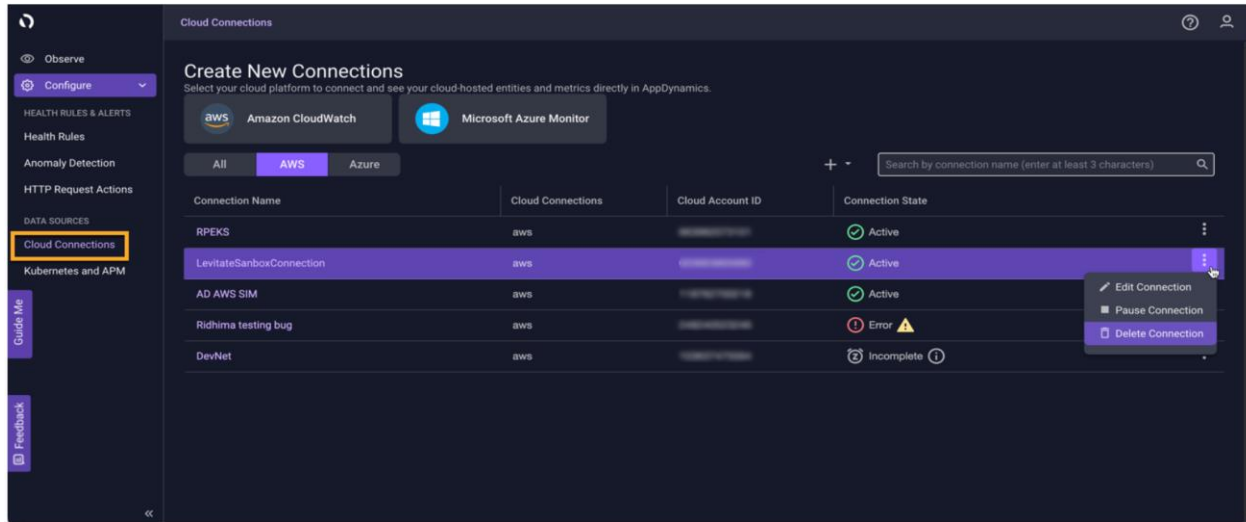
- **APM** (Application Performance Monitoring) represents the services that are being monitored using OpenTelemetry
- **Kubernetes** represents the monitoring of the components in your cluster including namespaces, workloads, pods, and ingresses.
- **Infrastructure** represents the monitoring of the Amazon Web Services (AWS) and Microsoft Azure cloud services that have been integrated through Amazon CloudWatch and Azure Monitor



AWS Cloud Infrastructure Observability

- Observability is an evolution of Monitoring. Monitoring is reactive and tells you when something is not working based on a predefined set of metrics. Observability is proactive and provides visibility across multiple domains using the three pillars – metrics, logs, traces. Monitoring tells you when something is wrong, Observability gives you the root cause of the problem.
- Monitoring is an important part of maintaining the reliability, availability and performance of your compute/storage/network infrastructure. The data you collect should be comprehensive so you can easily troubleshoot a multipoint failure if one were to occur.
- Cisco's Cloud Observability Platform provides end-to-end visibility into the performance of the infrastructure running your applications. The cloud monitoring solution for AWS:
 - Ingests data from Cloud Services automatically. For example, Amazon CloudWatch provides metrics for a better understanding of your resource availability and utilization.
 - Enables you to manage alerts based on infrastructure metrics and monitors the cloud service's health and performance.
- Multi cloud deployments are here to stay. Cisco's Cloud Observability Platform is that single pane of glass that normalizes observability across AWS, Azure, and GCP

Getting started with AWS Cloud Infrastructure Observability



- Set up Data Sources
 - Cloud Connections
- Set up Health Rules & Alerts
 - Health Rules
 - Anomaly Detection
 - HTTP Request Actions

Using Automation Methodologies

Case Study: AWS Infrastructure Observability

We will explore the following automation methodologies in this lab to help you get started with including cloud observability for your AWS deployments. What we do here for AWS can be extended to Azure and GCP using these same methodologies:

- Cisco's Cloud Observability Platform API
- Cisco's Cloud Observability Platform Terraform Provider
- Cisco Intersight Cloud Orchestrator

Cisco's Cloud Observability Platform APIs

Cisco Cloud Observability Platform offers several APIs that let you extend and customize various aspects of the application such as cloud connections, data ingestion, authentication, and so on.

- Service Principals - enable your developed code to securely connect to AppDynamics public APIs on your Cloud Tenant.
- Cloud Connections - connect Cisco Cloud Observability Platform to one more cloud provider accounts.
- Common Ingestion Service - publish OpenTelemetry metric data to the common ingestion pipeline
- Health Rules - enables you to programmatically create, update, and manage one or more health rules for the monitored entities.
- Anomaly Detection - create Anomaly Detection configurations for specific entities and entity types, helps determine whether every service in your application performs within the acceptable performance limits
- Actions - enables you to create, configure, and manage various actions which are predefined, reusable, automated response to an event such as a health violation or an anomaly
- Query Service - allows you to retrieve observation MELT (metrics, events, logs, traces) using AppDynamics domain-specific language called Unified Query Language (UQL).

Cisco's Cloud Observability Platform Terraform Provider

- Terraform Provider details can be found here:

<https://github.com/cisco-open/terraform-provider-appd>

- Current focus on cloud connections
- Extensions leveraging API's will be demonstrated in this lab

Cisco Intersight Cloud Orchestrator

| Display Name | Description | Default Ve... | Executions | Last Execution ... | Last Execution Time | Failed/Ter... | Validation... | Last Update | |
|----------------------------|-------------|---------------|------------|--------------------|---------------------|-----------------------|---------------|-------------|-----------------------|
| CloudObservabilityDeleteWF | | | 1 | 5 | Success | Nov 19, 2023 12:31 PM | Disabled | Valid | Nov 14, 2023 2:44 PM |
| CloudObservabilityWF | | | 1 | 10 | Success | Nov 19, 2023 12:27 PM | Disabled | Valid | Oct 18, 2023 11:02 AM |
| CloudObsWorkflow | | | 1 | 7 | Success | Oct 18, 2023 9:15 AM | Disabled | Valid | Oct 18, 2023 11:00 AM |

- With Intersight Cloud Orchestrator you can streamline automation with a user-friendly designer that enables you to build and run sophisticated workflows without coding. We will exercise a simple workflow to setup everything that you did with API and Terraform above.
- For more details on Intersight Cloud Orchestrator, check out:

https://us-east1.intersight.com/help/saas/orchestration/overview_orch

Lab case Study

- We will use a simple example to walk through automation for observability with API, Terraform and ICO
- Use API's/Terraform/ICO to set up Observability for EC2 instances in Cisco's Cloud Observability Platform:
 - Setup Cloud Connection to AWS Account
 - Setup Health Rules, Actions & Triggers
 - Setup Anomaly Detection, Actions, Triggers

Observe AWS Entities

Load Balancers

Storage

Databases

Hosts

ECS

Serverless

Observe AWS Infrastructure Entities

After you [Configure AWS Cloud Connections](#), you can monitor the performance of the infrastructures in your cloud platform.

The AWS cloud infrastructure services you can monitor are:

AWS Load Balancers

AWS Application, Classic, and Network Load Balancers

→

AWS Storage

Amazon Elastic Block Storage

Amazon Elastic File System

→

AWS Databases

Amazon Relational Database Service

→

AWS Hosts

AppDynamics Hosts for AWS

Amazon Elastic Compute Cloud

→

Amazon Elastic Container Services

Amazon ECS Clusters

Amazon ECS Services

Amazon ECS Task Definitions

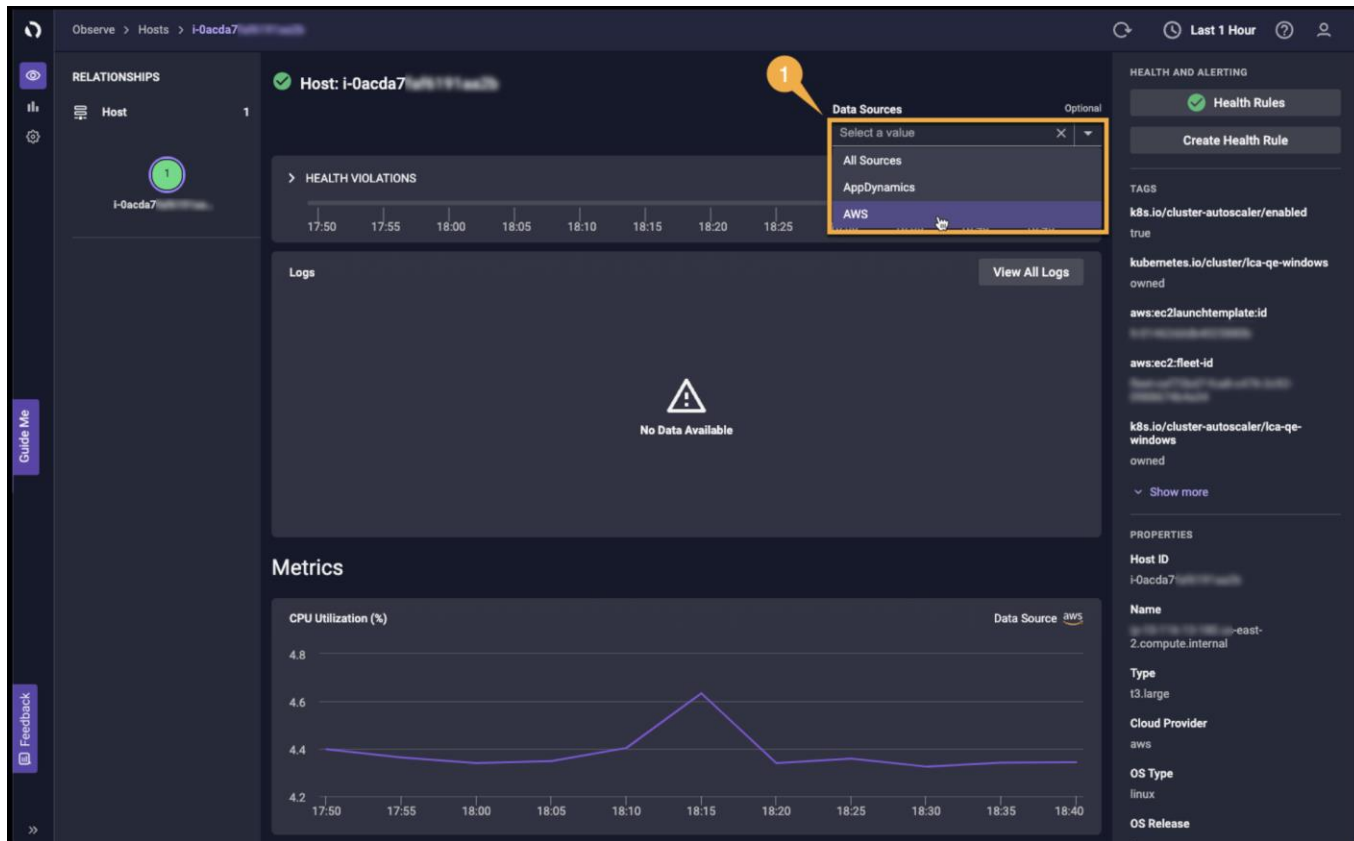
→

AWS Serverless

AWS Lambda

→

Observe AWS Hosts



Observe AWS Load Balancers

The screenshot displays the Cisco Observability console interface for monitoring AWS Load Balancers. The top navigation bar includes a search icon, the breadcrumb "Observe > Load Balancers", and a "Last 1 Week" time filter. The left sidebar contains navigation links for "Guide Me" and "Feedback".

The main content area is titled "Load Balancers (441)". It features two buttons: "Overview" (labeled "See All Load Balancers") and "View Unhealthy Load Balancers" (labeled "Filter list to show Unhealthy Load Balancers"). Below these are filter tabs for "All", "AWS" (which is selected), and "Azure".

A "Filter View" section includes an "Add filters" input field and an "Apply" button. Below this is a table listing the load balancers. The table has the following columns: Health, Name, DNS Name, Type, Target Response ..., Unhealthy Host C..., and Target TLS Negot....

| Health | Name | DNS Name | Type | Target Response ... | Unhealthy Host C... | Target TLS Negot... |
|--------|------|----------|---------|---------------------|---------------------|---------------------|
| ? | ... | ... | network | - | - | - |
| ? | ... | ... | network | - | - | - |
| ? | ... | ... | network | - | - | - |
| ? | ... | ... | network | - | - | - |
| ? | ... | ... | network | - | - | - |
| ? | ... | ... | network | - | - | - |

On the right side of the console, there is a "HEALTH AND ALERTING" section with buttons for "Health Rules" and "Create Health Rule".

On the left sidebar, there are summary cards for "Containers" (4.1K), "Hosts" (304), and "Load Balancers" (441). Each card displays three colored circles (red, green, grey) with numerical values inside them.

Observe AWS Storage

The screenshot displays the Cisco Observability console for monitoring AWS Storage. The interface is divided into several sections:

- Left Sidebar:** Contains navigation options for "Hosts" (94) and "Storages" (107). The "Storages" section is currently selected, showing a summary of 3 unhealthy and 104 healthy storage entities.
- Main Content Area:**
 - Storages (107):** Overview section with buttons for "Overview" (See All Storage Entities) and "View Unhealthy Storage Entities" (Filter list to show Unhealthy Storage Entities).
 - Filters:** Tabs for "All", "AWS" (selected), and "Azure".
 - Filter View:** A section for adding filters.
 - Table:** A table displaying storage metrics for AWS. The table has columns: Health, Disk ID, Read Ops/Sec, Write Ops/Sec, Throughput (%), Queue Length, Total Write Ti..., and Total Read Ti....
- Right Sidebar:** Contains "HEALTH AND ALERTING" options, including "Health Rules" and "Create Health Rule".

Table Data:

| Health | Disk ID | Read Ops/Sec | Write Ops/Sec | Throughput (%) | Queue Length | Total Write Ti... | Total Read Ti... |
|--------|---------|--------------|---------------|----------------|--------------|-------------------|------------------|
| ? | vol-... | 0 | 0 | - | 0 | 0.42 | 0 |
| ? | vol-... | 0.09 | 0.09 | - | 0 | 0.01 | 0 |
| ? | vol-... | - | - | - | - | - | - |
| ? | vol-... | 0 | 0 | - | 0 | 0 | 0 |
| ? | vol-... | - | - | - | - | - | - |

Observe AWS Databases

The screenshot displays the Cisco Observability console interface for monitoring database instances. The top navigation bar shows 'Observe > Database Instances' and a 'Last 1 Week' filter. The left sidebar contains navigation icons for 'RELATIONSHIPS', 'Database Instances' (31), and a 'Guide Me' button. The main content area is titled 'Database Instances (31)' and includes tabs for 'All', 'AWS', and 'Azure'. Below the tabs is a 'Filter View' section with a 'Loading...' indicator and an 'Apply' button. A table lists the database instances with columns for 'Health', 'Name', and 'Type'. The table shows various instances, including 'vishaka-aurora-instance-instance-1' (aurora-mysql), 'cce2esim-cluster-i1' (aurora-mysql), and 'cloudkart-aws-dev-orderdb' (mysql). The right sidebar contains a 'HEALTH AND ALERTING' section with buttons for 'Health Rules' and 'Create Health Rule'.

Observe > Database Instances

RELATIONSHIPS

Database Instances 31

Database Instances (31)

All AWS Azure

Filter View

Loading...

Apply

| Health | Name | Type |
|--------|--|--------------|
| ? | vishaka-aurora-instance-instance-1 | aurora-mysql |
| ? | vishaka-aurora-instance-instance-1-us-west-2b | aurora-mysql |
| ? | cce2esim-cluster-i1 | aurora-mysql |
| ? | cce2esim-cluster-i0 | aurora-mysql |
| ? | /subscriptions/f0319a3f-4404-48f3-8a1e-06b0d38b0dcb/resourc... | postgres |
| ! | cce2esim-mysql-db | mysql |
| ! | cloudkart-aws-dev-orderdb | mysql |
| ! | cloudkart-aws-ga-orderdb | mysql |
| ? | vishaka-mysql | mysql |

HEALTH AND ALERTING

Health Rules

Create Health Rule

Observe AWS ECS Clusters

Observe > ECS Clusters

RELACTIONSIPS

ECS Clusters 11

11

Guide Me

Feedback

ECS Clusters (11)

Filter View

Add filters

Apply

| Health | Name | Service Count | Task Count | CPU Utilized (%) | Memory Utilize... | Network Trans... | Network Recel... |
|--------|--------------------|---------------|------------|------------------|-------------------|------------------|------------------|
| ? | cce2esim-clust... | 1 | 2 | 0.21 | 20.71 | 400.23 | 26.74 |
| ? | edmunds-dock... | - | - | - | - | - | - |
| ? | david-mirza-clu... | - | - | - | - | - | - |
| ? | shivtest-cluster | 1 | 2 | 0.79 | 33.74 | 1.076K | 30.17 |
| ? | cw | - | - | - | - | - | - |
| ? | my-test-cluster | - | - | - | - | - | - |
| ? | sim-test | - | - | - | - | - | - |
| ? | CloudWatch | 6 | 5 | 0.01 | 4.84 | 10.44 | 15.53 |
| ? | erden-containe... | 4 | 7 | 0.01 | 4.24 | 0.06 | 0 |
| ? | default | - | - | - | - | - | - |
| ? | ecs-test-cloud-... | - | - | - | - | - | - |

HEALTH AND ALERTING

Health Rules

Create Health Rule

Observe AWS Lambdas

Observe / Lambdas

RELATIONSHIPS

No Data Available

Lambdas (49)

Filter View

```
isActive = true && attributes(status) in [Unknown 1, null]
```

Apply

Group View (up to 2 levels of tags)

Add Group

Apply

| Health | Name | Concurrent Executions | Duration | Errors | Invocations |
|--------|--------------------|-----------------------|----------|--------|-------------|
| ? | MeasureTimes | - | - | - | - |
| ? | erden-s3-destin... | - | - | - | - |
| ? | myFirstJsLamb... | - | - | - | - |
| ? | VishakaEfsLam... | - | - | - | - |
| ? | erden-s3-destin... | - | - | - | - |
| ? | sim-k8s-vgw-p... | 1 | 21.11K | 0 | 36 |
| ? | BugBashErrorL... | - | - | - | - |
| ? | snapshot-analy... | - | - | - | - |
| ? | MyFunction | - | - | - | - |
| ? | VishakaEfsLam... | - | - | - | - |
| ? | aws-controllow... | - | - | - | - |
| ? | aws-controllow... | - | - | - | - |
| ? | CloudkartNotifi... | - | - | - | - |
| ? | aws-controllow... | - | - | - | - |

HEALTH AND ALERTING

Health Rules

Create Health Rule

Feedback

Additional resources

Cisco Cloud Observability Platform Developer API's: <https://developer.cisco.com/docs/appdynamics/#!/introduction>

Cisco Cloud Observability:

<https://docs.appdynamics.com/appd-cloud/en/about-appdynamics-cloud>

[https://www.appdynamics.com/product/cisco-cloud-](https://www.appdynamics.com/product/cisco-cloud-observability#xd_co_f=YmUyZmEwMGEtOGEwZi00MzRkLWFkYTEtNGQ3YjA2YmY3Yjk4~)

[observability#xd_co_f=YmUyZmEwMGEtOGEwZi00MzRkLWFkYTEtNGQ3YjA2YmY3Yjk4~](https://www.appdynamics.com/product/cisco-cloud-observability#xd_co_f=YmUyZmEwMGEtOGEwZi00MzRkLWFkYTEtNGQ3YjA2YmY3Yjk4~)

Useful Blogs to get started:

<https://blogs.cisco.com/developer/appdynamicscloudapis01>

<https://blogs.cisco.com/news/cisco-further-delivers-on-full-stack-observability-ecosystem-vision>

<https://www.appdynamics.com/resources/ebook/how-to-solve-your-biggest-cloud-problems>

<https://blogs.cisco.com/applications/delivering-application-performance-to-maximize-business-kpis>

Observability & Monitoring:

<https://www.cisco.com/c/en/us/solutions/full-stack-observability/what-is-observability.html>

<https://blogs.cisco.com/developer/observabilityvsmonitoringdevops01>

Learning lab link:

<https://developer.cisco.com/learning/labs/appd-fso-clapj2/introduction/>

<https://developer.cisco.com/learning/labs/appd-fso-clapj/introduction/>

Sandbox link: <https://dcloud2-rtp.cisco.com/>

Postman collection: <https://www.postman.com/ciscodevnet/workspace/cisco-fso-platform/overview>

Homework!

Reserve the DevNet sandbox:

Sandbox: <https://dcloud2-rtp.cisco.com/content/instantdemo/appdynamics-observability-in-aws?returnPathTitleKey=content-view>

Exercise the learning lab but with your own AWS account:

Learning Lab:

<https://developer.cisco.com/learning/labs/appd-fso-clapj2/introduction/>

<https://developer.cisco.com/learning/labs/appd-fso-clapj/introduction/>

In the section where it lists the AWS account, put in your own AWS account details and make the cloud connection to your own cloud assets.

Related Cisco Live DevNet Sessions

Thursday, Dec 7

12:30 PM - 1:15 PM AEDT

DevNet Workshop 2 Infrastructure Observability for AWS/Azure/GCP with CNAO using APIs, Terraform and Workflows [DEVWKS-2178]

Friday, Dec 8

12:00 PM - 12:45 PM AEDT

DevNet Workshop 2 Infrastructure Observability for AWS/Azure/GCP with CNAO using APIs, Terraform and Workflows [DEVWKS-2178]

Friday, Dec 8

10:15 AM - 11:00 AM AEDT

DevNet Workshop 1 - Leveraging the Cisco FSO Platform: Integrating with Cisco Intersight for Enhanced Observability [DEVWKS-2795]



The bridge to possible

Thank you

CISCO *Live!*

The background of the slide is a vibrant, abstract graphic. It features a large, stylized cloud shape on the left side, composed of overlapping, semi-transparent bands of color in shades of red, orange, yellow, and green. To the right of the cloud, a bright, multi-colored sunburst or starburst pattern radiates outwards, with colors transitioning from yellow and orange in the center to blue and green towards the edges. The overall effect is energetic and colorful.

cisco *Live!*

Let's go