JupiterOne

CASE STUDY

# How Databricks Achieves Greater Asset Visibility with JupiterOne

databricks

As the leader in Unified Data Analytics, Databricks helps organizations make all their data ready for analytics, empower data science and data-driven decisions across the organization, and rapidly adopt machine learning to outpace the competition. By providing data teams with the ability to process massive amounts of data in the Cloud and power AI with that data, Databricks helps organizations innovate faster and tackle challenges like treating chronic disease through faster drug discovery, improving energy efficiency, and protecting financial markets.

# How Databricks Achieves Greater Asset Visibility with JupiterOne

The first requirement for any tool Adam Youngberg, Kishore Fernando, and the security engineering team at Databricks adds must enable the company to maintain the highest level of security.

Security is at the core of Databricks' mission. Adam and Kishore were brought on board at Databricks by Caleb Sima, former VP of Security, to find or build a solution that enables greater visibility and discoverability across the company's cloud assets, as well as the owners of those assets. "Centralization and visibility can enable greater security and speedier remediation," noted Adam.

Most asset management and configuration monitoring tools fall short on visibility, openness, and flexibility. Not only that, most lean towards legacy, on-prem businesses with cloud as an afterthought. They wanted something cloud-first.

## Getting Up and Running

Traditionally, asset management and configuration monitoring tools are time-intensive and taxing to deploy, especially virtual appliances. Being 100% SaaS, JupiterOne prioritizes deployment speed and time to first value for already busy security teams. As a result, getting your resources ingested and mapped takes only minutes, whether your environment is complex or relatively simple.

"One thing we were pleasantly surprised by at the outset

## Challenges

- Needs a cloud-first cyber asset management solution that offers visibility, openness, and flexibility to scale
- Reactive alerting and vulnerability management in managing Amazon S3 buckets

## Results

- Within one hour of deploying JupiterOne, ingested and confirmed that data from our AWS resources
- Continuous and automated management of AWS and other cloud assets
- Improved overall company security posture by validating user completion of relevant KnowBe4 awareness training

## Key Integrations

aws AWS

Qualys

KnowBe4 Knowbe4

"JupiterOne and its asset monitoring capability has become the foundation and the core to almost everything in our security operations. It is our one source of truth."

**Caleb Sima,** former VP of Security at DataBricks

was how painless the integration process was," Adam noted. "Within an hour and a half on that first day, we had ingested and confirmed that the data coming into JupiterOne from our AWS resources looked right. All of the managed integrations are painless."

## JupiterOne for S3 Bucket Security

One of the early objectives for Databricks was ensuring Amazon S3 bucket security. "When you think about security and vulnerability, you have to know about your assets. Leveraging Amazon GuardDuty for alerts is helpful, but it's a reactive approach with limited input."

For example, Adam could dive in to see which buckets appear exposed because they are publicly accessible. But there are situations where a bucket is not, itself, publicly accessible but could still be exposed. For example, If a resource is only accessible via Cloudfront, but Cloudfront is public, the assets should be considered public and exposed.

"We needed the ability to quickly and reliably know what S3 buckets existed, who owned them, and whether they were publicly accessible themselves or via another service. Not only that, we wanted to move from being reactive to proactive in our vulnerability management. The only way to do that was to look at the relationships between the resources in our AWS environment."

To be proactive, Adam needed deeper details about their cloud asset states and the relationships between multiple services. This information would help the team determine if an asset's state is conducive to problems, making it easy to confirm the alerts you are getting.

## Incident Response

Because of the transient nature of many of the cloud assets used in Databricks products, it is important to respond to reported vulnerabilities with data. Adam and the team useJupiterOne as a critical tool during their incident response and triage process. It is often the starting place for triage and follow-up answers.

> **JupiterOne has become critical for S3 bucket security because it provides a really good line of sight into assets to get ahead of vulnerability management.**

"If someone reports that an IP address owned by Databricks has security concerns, we can establish a timeline for when we owned the IP address and whether the identified problems are ours. Most times, the problem is for an address that was ours but now belongs to someone else, and the security issue is not ours. Here the vulnerability belongs to one of their assets, and querying the graph makes it easy to assess what needs to be changed and who is responsible for changing it. This insight boosts confidence in our incident response process," said Adam.

## Beyond S3 Bucket Security

Adam enjoys using the relationship queries in JupiterOne for reviewing access controls of both their cloud and non-cloud digital assets.

## JupiterOne Streamlines & Centralizes Reporting

Identifies stale resources & improves cloud hygiene, to reduce noise

Centralized graphDB gives clear visibility & simple analysis of all cloud assets

Automated data collection & relationship mapping provides a complete picture

Reports are automatically generated, removing the need for manual creation

"With most tools, it is pretty easy to see who has admin access. But when there are a lot of accounts, the process is tedious."

With JupiterOne, Adam and Kishore can review admin policy details across all tools and resources in a single UI. They can also run a query that displays accounts with root access that can assume admin privileges of accounts for greater access visibility and security.

"For both cases, we configured a JupiterOne query to gain visibility. That query is saved as a rule, and we are quickly alerted when changes occur."

Every JupiterOne query we add answers a critical question for us and becomes its own use case.

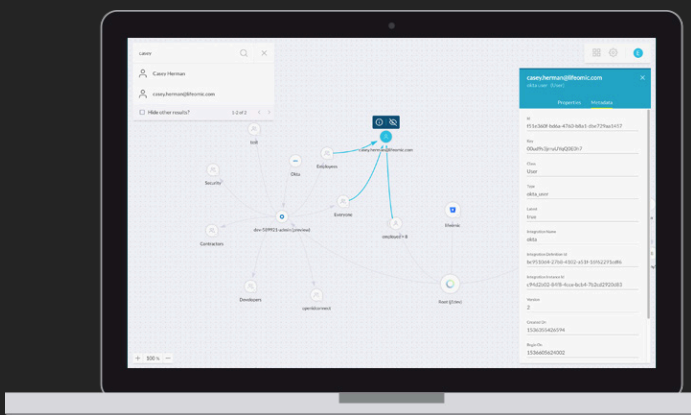
## Building on JupiterOne's Openness with AWS

JupiterOne is an open platform. As a result, security engineering teams don't need to wait for added integrations if they want more centralized information. Instead, they can leverage JupiterOne's API to ingest the data. Not only that, JupiterOne's graph data model will automatically map relationships on known entities.

Adam and Databricks built a Slack integration using the JupiterOne API, a lambda, and Amazon CloudWatch for scheduling. The integration runs every 30 minutes and notifies the security team when there are changes to admin users, which should be rare, and if the organization ownership changes. This integration helps ensure that those who have high access aren't making unexpected changes through the wrong channels.

"The nice thing when creating our integration was that JupiterOne's Mapping was smart enough to determine the email address added in our Slack integration matched those added by existing integrations, mapped relationships properly and automatically, with no additional configuration from me."





Because of the graph model, the Databricks security team can further validate that those with Slack admin access have also completed relevant KnowBe4 awareness training.

## Always Speedy Support

Databricks leverages the JupiterOne community slack channel to collaborate with the JupiterOne engineering team.

"Whether I want help configuring a JupiterOne query, have an idea for simplifying a process in JupiterOne, or come across an issue, the team at JupiterOne responds and delivers. Almost routinely, items in the release notes come from requests we had put in – often within the same release, sometimes even the same day! So it's obvious how much impact customers have on the product."