elastic

The Elastic Observability guide for AWS

elastic.co →

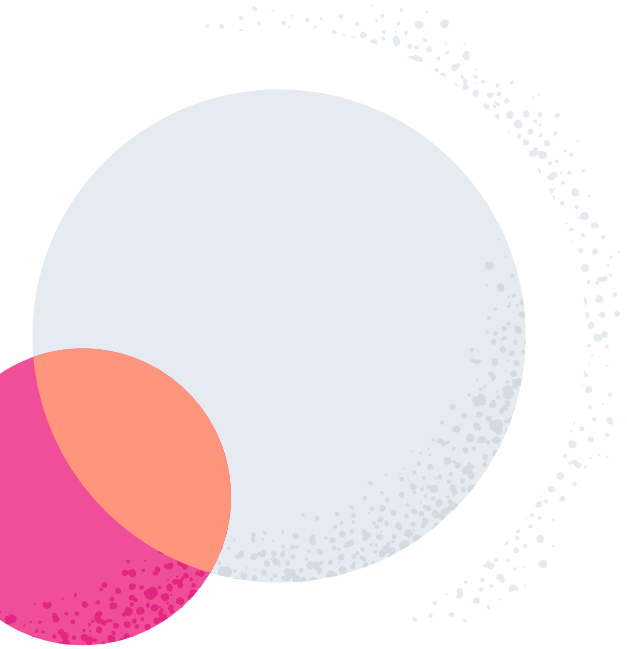
Table of Contents

Introduction

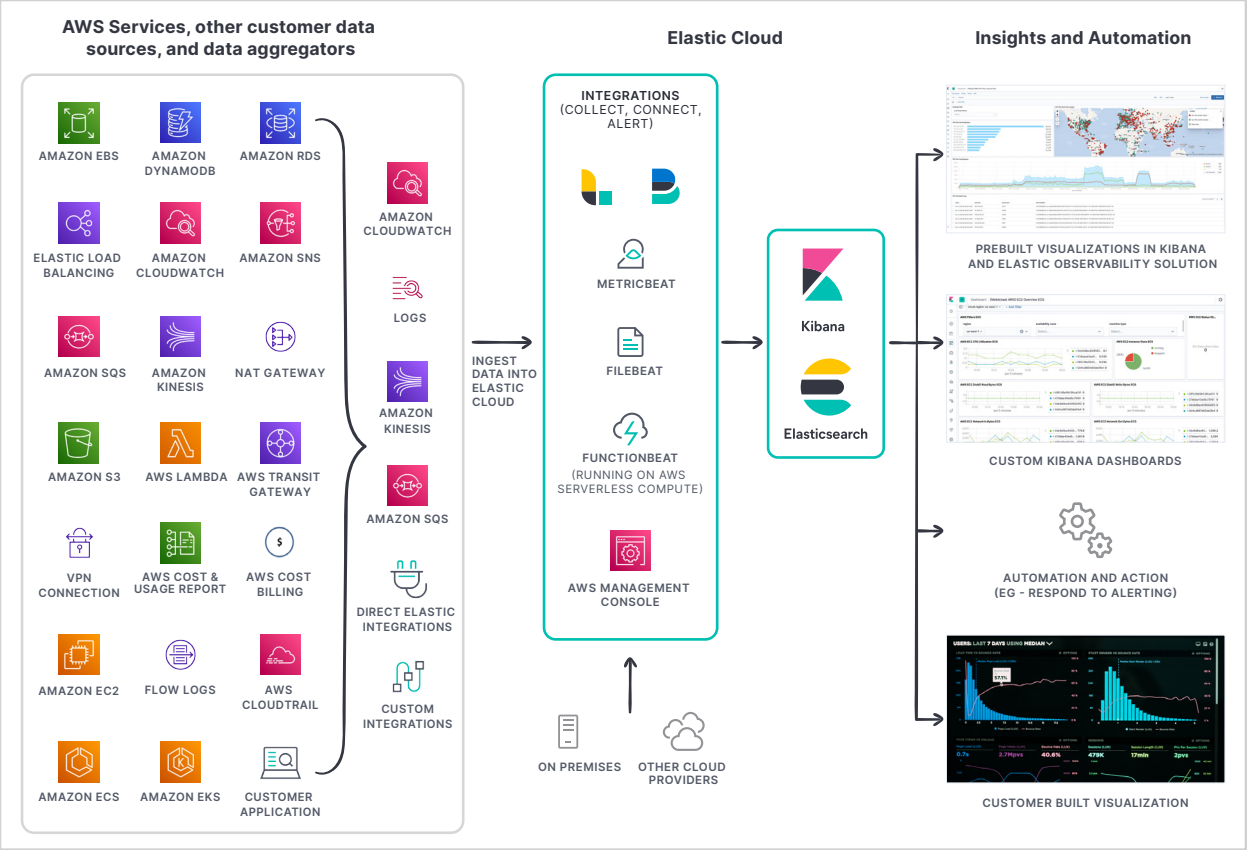
Driving insights and action based on data is critical in order to fully benefit from the agility and flexibility enabled by cloud. With Elastic's observability solution, you can unify visibility across your entire AWS and on premises environments, enabling better understanding of the availability, performance, and overall health of your infrastructure, applications, and business.

AWS gives you a broad range of logs and metrics into their cloud services that allow you to monitor your cloud deployment and make more informed decisions. Elastic Observability integrates with these data sources to bring your data together in a unified manner, enabling you to continuously gain actionable insights into your IT, operations, and business. Easily analyze your data within prebuilt dashboards and tools or build custom visualizations that allow you to react quickly in regards to your business needs.

This guide explains how to best configure Elastic Observability with AWS services so you can more effectively monitor and more quickly react to events as they occur. Please continue reading to learn more about these AWS services, the benefits of using Elastic for monitoring, and best practices that can help maximize the value of your investments in both.



Elastic allows you to do more with your AWS data



Monitor and analyze Amazon CloudWatch Logs with Elastic

Centralize logs from across your infrastructure, applications, and the AWS services that you use, in a single, scalable service with Amazon CloudWatch.

Amazon CloudWatch Logs enable you to quickly and easily:

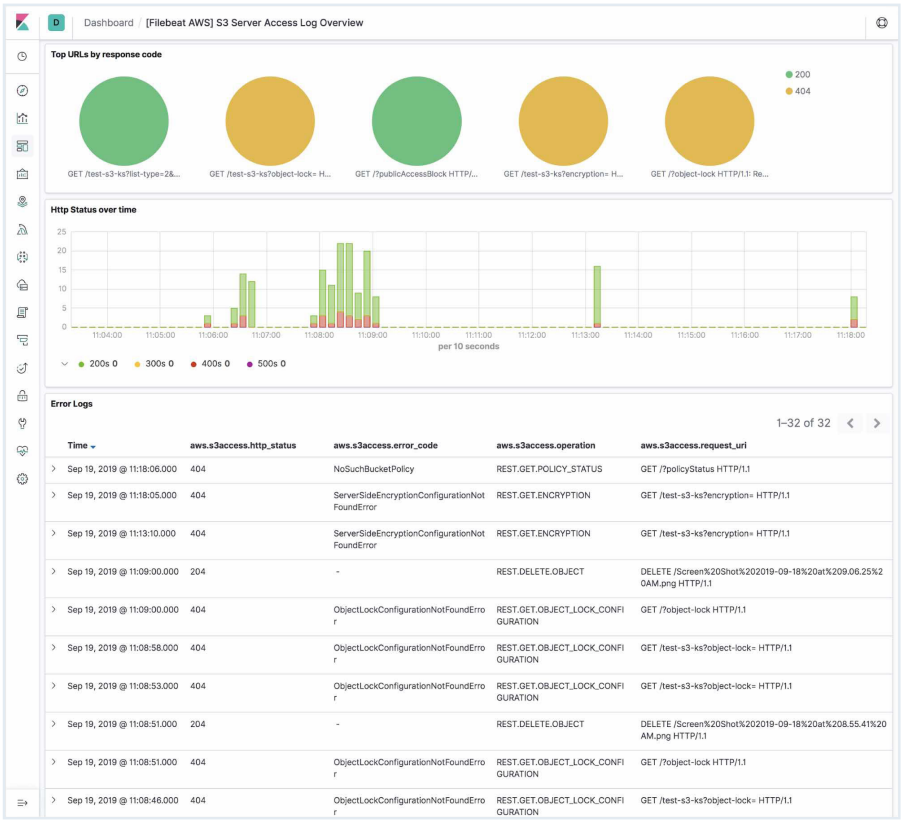
Gather, store, and access log files from disparate sources

Monitor the health and performance of your infrastructure and applications

Observe Amazon CloudWatch Logs directly from different AWS log groups

elastic

How to send Amazon CloudWatch Logs to Elastic:



First, you'll need to gather information about your AWS environment and your Elastic Cloud deployment. Refer to **Appendix A** below for details on those prerequisites. To get started with Amazon CloudWatch Logs follow the steps in **Appendix B** below to get a walk through which includes details on:

1. Setting up an Amazon Simple Storage Service (Amazon S3) bucket and creating an Amazon Simple Queue Service (Amazon SQS) queue
2. Downloading and installing Filebeat
3. Connecting to the Elastic Stack
4. Configuring Filebeat to collect Amazon CloudWatch Logs
5. Enabling and configuring your data collection modules
6. Setting up your pre-configured Kibana dashboards and then starting Filebeat
7. Analyzing Amazon CloudWatch data in Kibana

Analyze Amazon S3 log activity and monitor access with Elastic

Amazon S3 allows you to store data, business applications, and host static websites. With Amazon S3, there are two types of workflows you can implement: the collection of custom logs stored with Amazon S3 and the monitoring of Amazon S3 service access and metrics.

Use Elastic with Amazon S3 to:

Capture details of requests such as, remote IP, requestor, bucket name, and more to get a better understanding of the nature of the traffic against your buckets

Establish baselines, analyze access patterns, and identify trends within Kibana's predefined dashboards

Identify security and compliance issues as well as conduct root cause analysis across your organization

Analyze custom business or application specific logs stored in **Amazon S3**

How to send Amazon S3 logs to Elastic:

First, you'll need to gather information about your AWS environment as well as your Elastic Cloud deployment. Refer to **Appendix A** for more details on those prerequisites. To get started with Amazon S3 logs, follow the steps in **Appendix B** for a walk through which includes details on:

1. Setting up an Amazon S3 bucket and creating an Amazon SQS queue
2. Downloading and installing Filebeat
3. Connecting to the Elastic Stack
4. Enabling and configuring your data collection modules
5. Configuring Filebeat to collect Amazon S3 Logs
6. Setting up your pre-configured Kibana dashboards then start Filebeat
7. Analyzing Amazon S3 log data in Kibana

Stream data into Elasticsearch with Amazon Kinesis

Amazon Kinesis is a fully managed service for delivering real-time, streaming data sources to destinations such as Amazon S3 and Elastic.

With Amazon Kinesis you can:

Stream logs in real-time and analyze them with Elasticsearch and Kibana so you can derive insights quickly and make more informed decisions

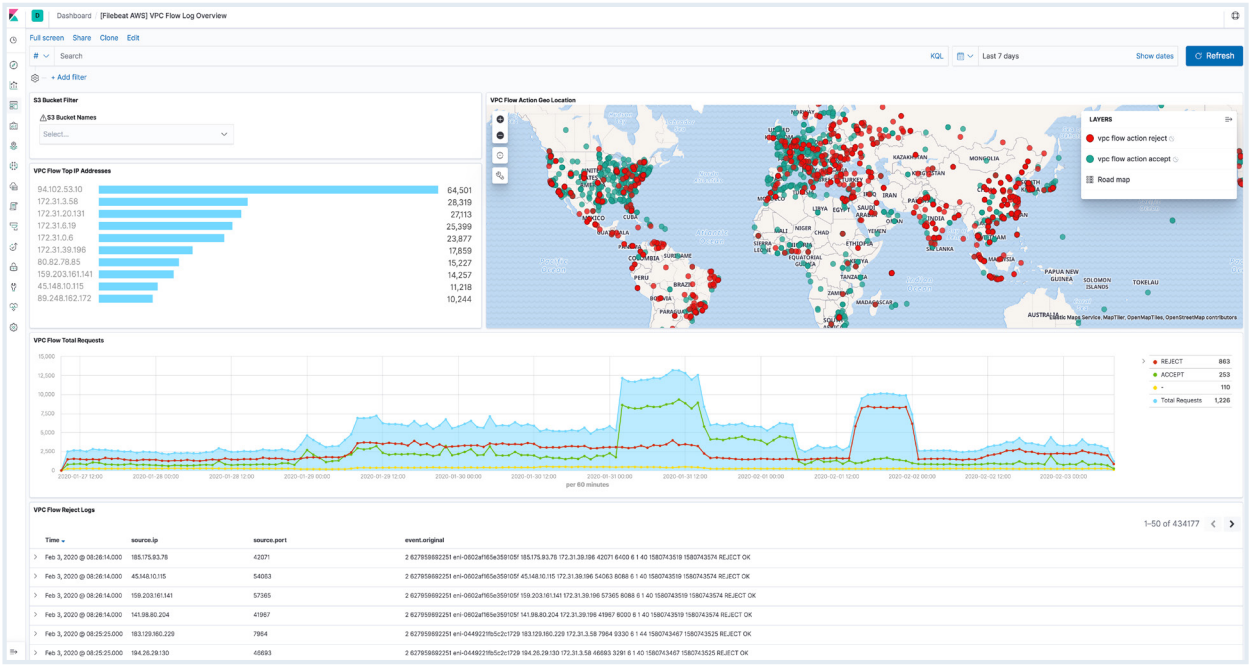
Compress, convert, and encrypt the data in transit to reduce the amount of storage used while increasing security

How to stream data into Elastic using Amazon Kinesis:

You'll need information about your AWS environment as well as your Elastic Cloud deployment before starting. Refer to **Appendix A** for more details on those prerequisites. To get started with Amazon Kinesis follow the steps in **Appendix C** for a walk through which includes details on:

1. Downloading and installing Metricbeat
2. Connecting to the Elastic Stack
3. Configuring Metricbeat to stream data
4. Enabling and configuring your data collection modules
5. Setting up your pre-configured Kibana dashboards and then starting Filebeat
6. Analyzing data in Kibana

Monitor network traffic with Amazon VPC Flow Logs with Elastic



Elastic Observability allows you to quickly search, view, and filter Amazon Virtual Private Cloud (Amazon VPC) Flow Logs to monitor network traffic within your Amazon VPC with Kibana. With this integration, you can analyze the flow log data and compare it with your security group configurations to maintain and improve your cloud security.

Ingesting Amazon VPC Flow Logs into Elastic enables you to:

- Perform better analysis to make more informed decisions
- Assess security groups rules and uncover security gaps
- Set alarms that alert you when certain traffic types are detected
- Identify latency issues and establish baselines to ensure consistent performance

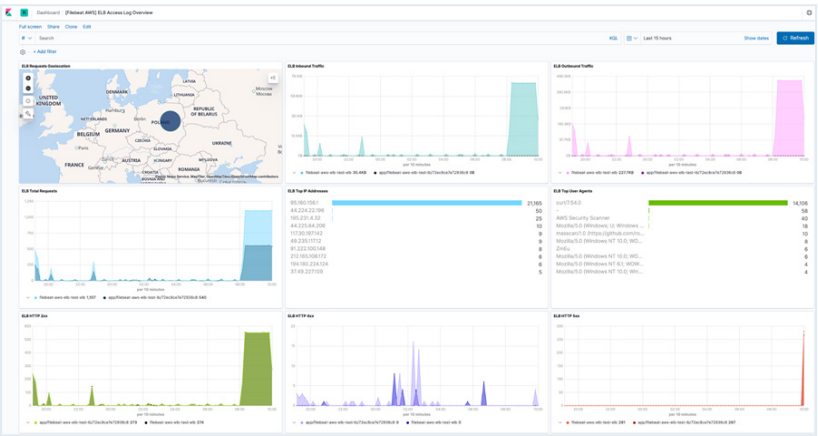
elastic

How to ingest Amazon VPC Logs into Elastic:

Begin by gathering information about your AWS environment as well as your Elastic Cloud deployment. Refer to **Appendix A** for more details on those prerequisites. To get started with Amazon VPC Flow Logs follow the steps in **Appendix B** for a walk through which includes details on:

1. Setting up an Amazon S3 bucket and creating an Amazon SQS queue
2. Downloading and installing Filebeat
3. Connecting to the Elastic Stack
4. Configuring Filebeat to collect Amazon VPC Flow Logs
5. Enabling and configuring your data collection modules
6. Setting up your pre-configured Kibana dashboards and then starting Filebeat
7. Analyzing logs in Kibana

Observe load balancing operations in Elastic with Amazon ELB



The Elastic Load Balancing (ELB) service on AWS allows you to automatically balance network traffic across a set of cloud resources.

When you use centralize ELB logs with Elastic, you're able to:

Observe detailed information about requests sent to the load balancer

Analyze traffic patterns and trouble issues to uncover performance issues

Drilldown into ELB logs to discover server responses, etc.

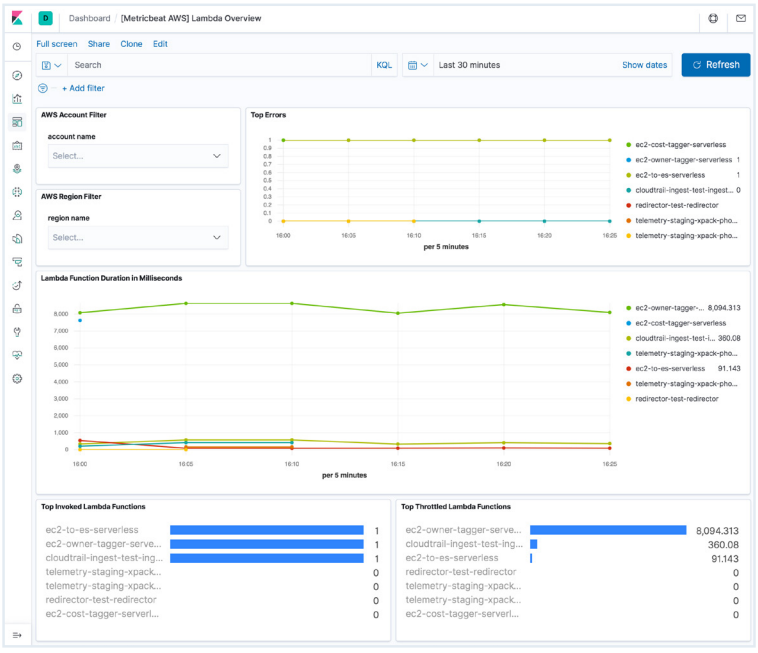
elastic

How to send Elastic Load Balancing data to Elastic:

Prior to getting started, you'll need to gather some information about your AWS environment as well as your Elastic Cloud deployment. Refer to **Appendix A** for more details on those prerequisites. To get started with the ELB on AWS, follow the steps in **Appendix B** for a walk through which includes details on:

1. Setting up an Amazon S3 bucket and creating an Amazon SQS queue
2. Downloading and installing Filebeat
3. Connecting to the Elastic Stack
4. Configuring Filebeat to collect ELB logs on AWS
5. Enabling and configuring your data collection modules
6. Setting up your pre-configured Kibana dashboards and then starting Filebeat
7. Analyzing ELB logs in Kibana

Optimize operational workflows using AWS Lambda in Elastic



With AWS Lambda, you can take advantage of a serverless compute service that allows you to dynamically run code in response to events and optimize operational workflows. Perform any computing tasks, automatically manage your resources with code for any application, and benefit from no administrative tasks required.

When you use AWS Lambda within Elastic you're able to:

Monitor performance from different serverless applications

Process logs and metrics in real time

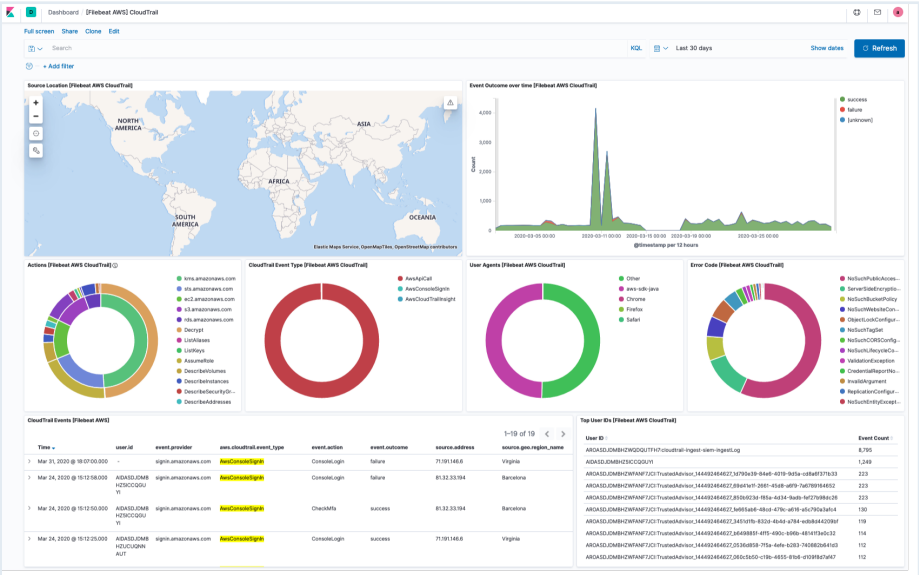
Capture and correlate performance data to Elastic solutions

How to get started with AWS Lambda within Elastic:

First, gather information about your AWS environment as well as your Elastic Cloud deployment. Refer to **Appendix A** for more details on those prerequisites. To get started with AWS Lambda follow the steps in **Appendix D** for a walk through which includes details on:

1. Downloading and installing Functionbeat
2. Connecting to the Elastic Stack
3. Configuring cloud functions
4. Enable and configure data collection modules
5. Setting assets and deploying Functionbeat
6. Building Kibana dashboards for analysis

Ensure governance and compliance standards with AWS CloudTrail in Elastic



elastic

AWS CloudTrail enables governance, compliance, operational auditing, and risk auditing of your AWS account.

When you centralize AWS CloudTrail logs in Elastic, you can easily:

Visualize your AWS CloudTrail logs as well as account and user activity all within Kibana's pre-built dashboards for faster analysis

Record information about all actions taken to track changes and resolve troubleshooting issues

Secure and monitor your network connections

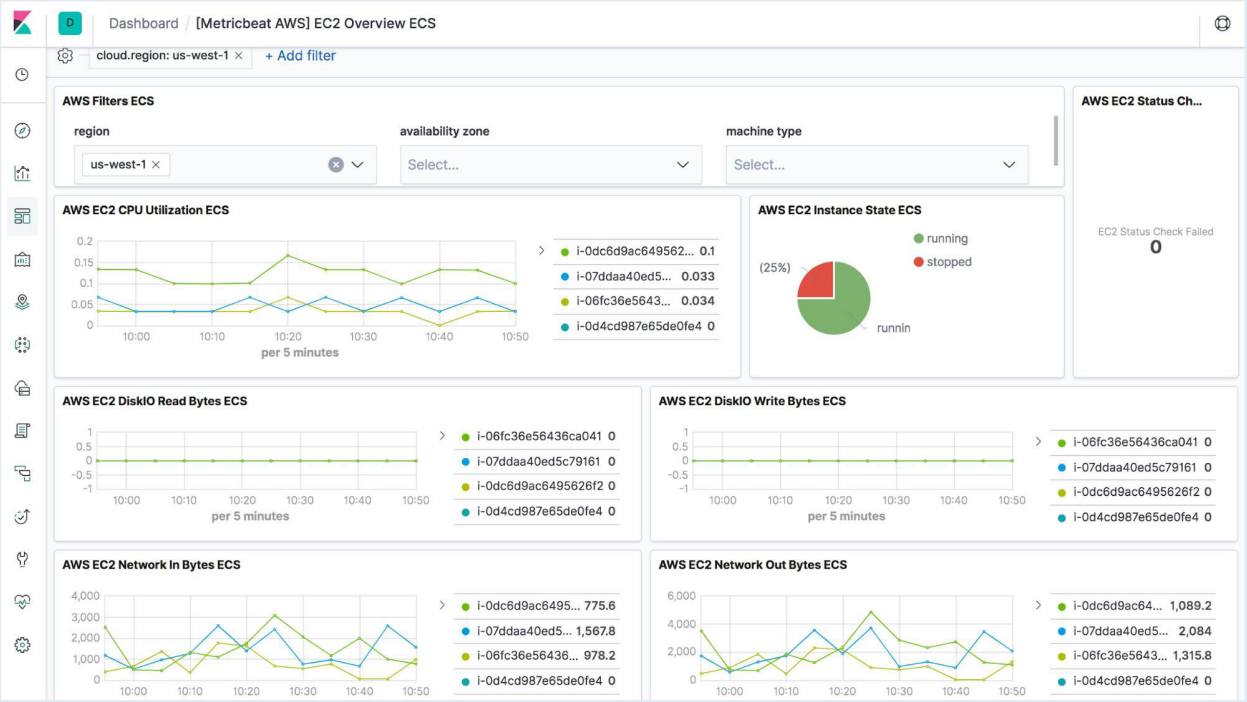
Ensure compliance with regulatory standards and policies

How to ingest AWS CloudTrail data to Elastic:

Before getting started, you'll need to gather some information about your AWS environment as well as your Elastic Cloud deployment. Refer to **Appendix A** for more details on those prerequisites. To get started with AWS CloudTrail follow the steps in **Appendix B** for a walk through which includes details on:

1. Setting up an Amazon S3 bucket and creating an Amazon SQS queue
2. Downloading and installing Filebeat
3. Connecting to the Elastic Stack
4. Configuring Filebeat to collect AWS CloudTrail logs
5. Enabling and configuring your data collection modules
6. Setting up your pre-configured Kibana dashboards then start Filebeat
7. Analyze AWS CloudTrail logs in Kibana

Ingest and unify metrics across your AWS environment to gain comprehensive insights



With Elastic's integrations and pre-built dashboards for AWS, you can collect AWS metrics such as usage, performance, billing, and more to see how every signal correlates — enabling you to make more informed business decisions.

Through continuous monitoring and analysis of your AWS compute, storage, networking, and data metrics, you can react quickly to your evolving business needs:

- Amazon Relational Database Service (Amazon RDS)
- Amazon Elastic Block Store (Amazon EBS)
- Amazon Elastic Compute Cloud (Amazon EC2)
- Amazon VPC Network Address Translation (NAT) Gateway
- Amazon CloudWatch
- Amazon S3
- Amazon DynamoDB
- Amazon Simple Notification Service (SNS)
- Amazon SQS
- AWS Cost & Usage Report
- AWS Billing and Cost Management
- AWS Virtual Private Network (AWS VPN)
- AWS Transit Gateway

elastic

The AWS metrics help you perform comprehensive analysis, allowing you to make more informed decisions that give you the ability to:

Correlate metrics across compute, storage, and data services for unified troubleshooting of issues

Evaluate constraints on capacity, performance, and usage, to make a holistic scaling decisions

Monitor and maintain an optimized cloud deployment with automated analysis and alerting, using a unified data set

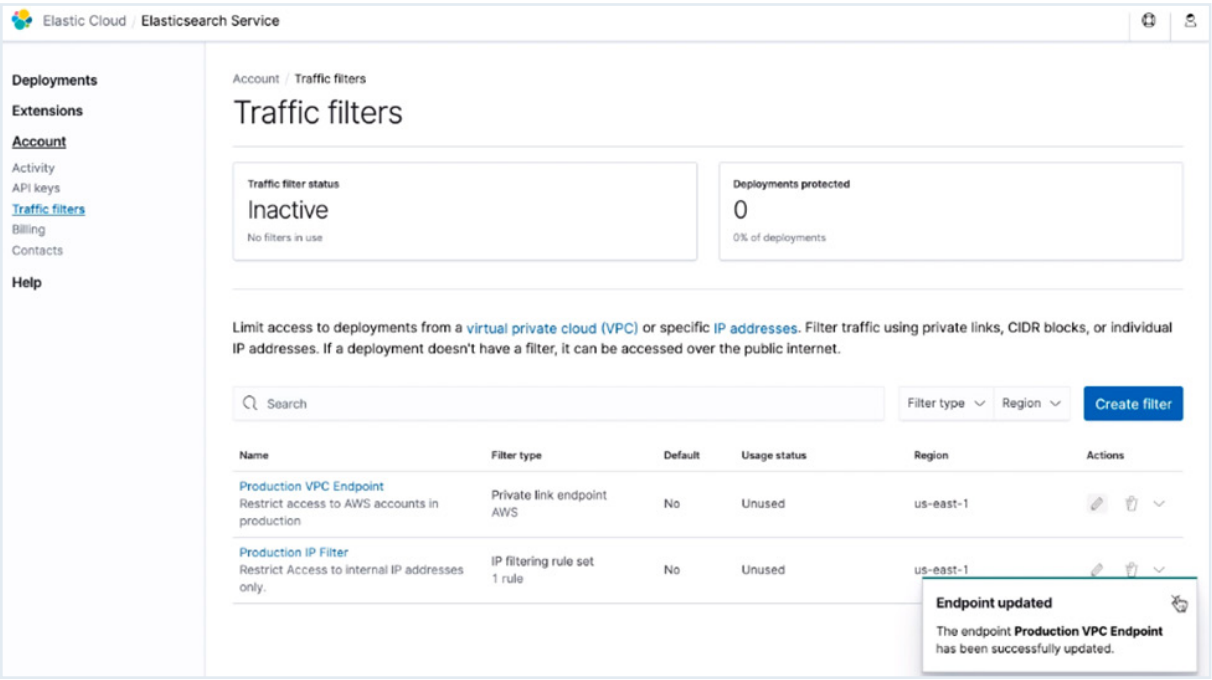
How to get started using AWS metrics and custom dashboards:

You'll need information about your AWS environment as well as your Elastic Cloud deployment before starting. Refer to **Appendix A** for more details on those prerequisites. To get started with creating your dashboard, follow the steps in **Appendix C** for a walk through which includes details on:

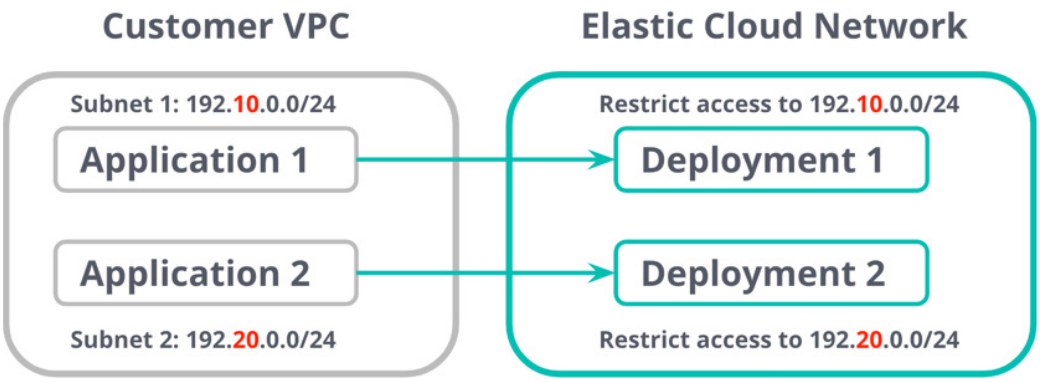
1. Downloading and installing Metricbeat
2. Connecting to the Elastic Stack
3. Configuring Metricbeat to collect metrics
4. Enabling and configuring your data collection modules
5. Setting up your pre-configured Kibana dashboards then start Filebeat
6. Analyzing your metrics in Kibana

To learn how to build a custom dashboard to suit your needs you can check out our **documentation** as well as this quick **video tutorial**.

Get added security and flexibility from Elastic using AWS PrivateLink



AWS PrivateLink provides secure connectivity between your Amazon VPCs, additional AWS resources, and on-premises applications. This makes it easy to secure the network connections between your applications and your Elastic deployment. The traffic between your virtual network and your Elastic deployment travels across the AWS network instead of the public Internet, eliminating data exposure and providing you with added security.



elastic

AWS PrivateLink lets you:

Create endpoints with private IP addresses, so workloads appear to be running within your network

Ensure that all traffic stays within the Amazon network and does not leave at any point

Benefit from simplified network management so you no longer need to maintain complex infrastructure (NAT Gateways, access controls)

Restrict traffic from the customer virtual networks to the endpoint (AWS PrivateLink traffic is unidirectional, as opposed to traffic in Amazon VPC peering, which is bidirectional)

How to get started with AWS PrivateLink:

Check out our **documentation** for step-by-step instructions.

Why Elastic?

Deploy Elastic to bring a set of complementary capabilities to the cloud which help you maximize the value of your AWS investments.

Elastic Observability and its underlying search platform capabilities complement cloud infrastructure innovations

Since its inception, Elastic has delivered a steady flow of search and data analysis innovations and redefined the value of search. Elastic, the creator of Elasticsearch and Kibana, constantly adds new capabilities, security updates, and performance enhancements to these products. Elastic's search innovations at the software application layer complement AWS innovations in the cloud infrastructure layer. When combined, you can rapidly respond to business and operational data, helping you move toward becoming a more agile, data-driven organization.

Choice and flexibility across cloud providers and on-premises

The Elastic search platform has been built to give developers and customers the flexibility to run in their location of choice. Strong investments enable this to build core capabilities into the platform while, at the same time, building deep integrations on the cloud. The Elastic Search Platform also offers a consistent experience in the cloud and on-premises. This hybrid consistency is valuable as you gradually increase your cloud use, a process that can take years at large enterprises.

Consistency across multiple clouds can also make expanding your solution easier if you choose to expand your cloud usage to add best of breed services from across cloud providers. This is particularly valuable for observability and security use cases, where a unified view across locations can help customers speed up troubleshooting and reduce risk.

Ready to use solutions for Enterprise Search, Observability, and Security

Elastic delivers pre-built, ready to use applications out of the box for Enterprise Search use cases — including Workplace Search, App Search, and Site Search; Observability use cases - including logging and Application Performance Monitoring (APM); and Security use cases — including SIEM and endpoint protection.

All the capabilities and external integrations that enable these solution specific applications are built into the Elastic search platform and available to customers who choose to build their own customized applications for their needs. This includes broad integrations to ingest the data required for Observability and Security solutions into AWS.

Community and technical talent

The Elastic search platform is a de facto standard for search powered solutions. The Elasticsearch GitHub community has over 1500 members. In addition, skillsets around Elasticsearch and Kibana are well established in the industry. Elasticsearch also includes readily available integrations for commonly used adjacent applications and data sources. Leveraging Elastic Observability with AWS allows you to use these resources — the talent pool, the integrations, and the collaborative Elasticsearch community — as you grow your search-powered solution.

Connecting with the Elastic Community

Discussion forums

Find advice or lend a helping hand. Ask your most burning questions about all things Elastic and share your wisdom with fellow users on our **discussion forums**, which are also available in your native language.

Slack and local communities

Join our fast-growing **Elastic Slack** to chat with other users and ask for advice in various channels: #elasticsearch, #kubernetes, #kibana-development, or others.

Additionally, many **other online communities** have sprung up all over the world! Join one in your region to share your Elastic story with the local community.

Keep learning

Getting started with the Elastic Stack? Looking for detailed deep dives? Get hands-on with the **Elastic examples repo** and explore curated datasets and step-by-step instructions. Plus, see what's making the rounds in our dev team through our **community newsletter**.

We'd love to hear from you

As technology evolves, so does Elastic. We really value hearing from our community. **Please reach out to us** for help or to share your thoughts about your Elastic experience.

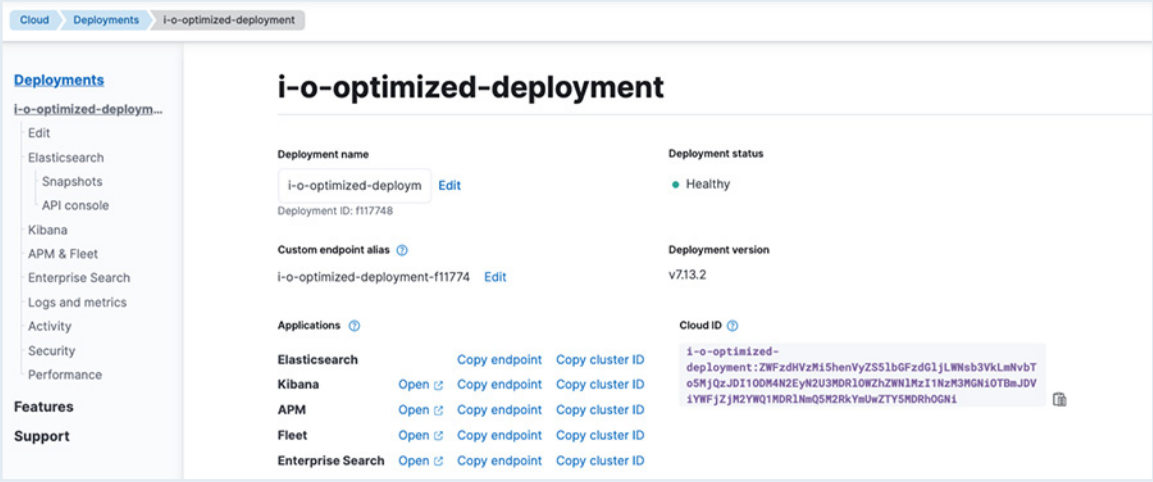
Appendix A – Prerequisites to getting started

Follow the instructions below to obtain the following information prior to getting started with ingesting your AWS data:

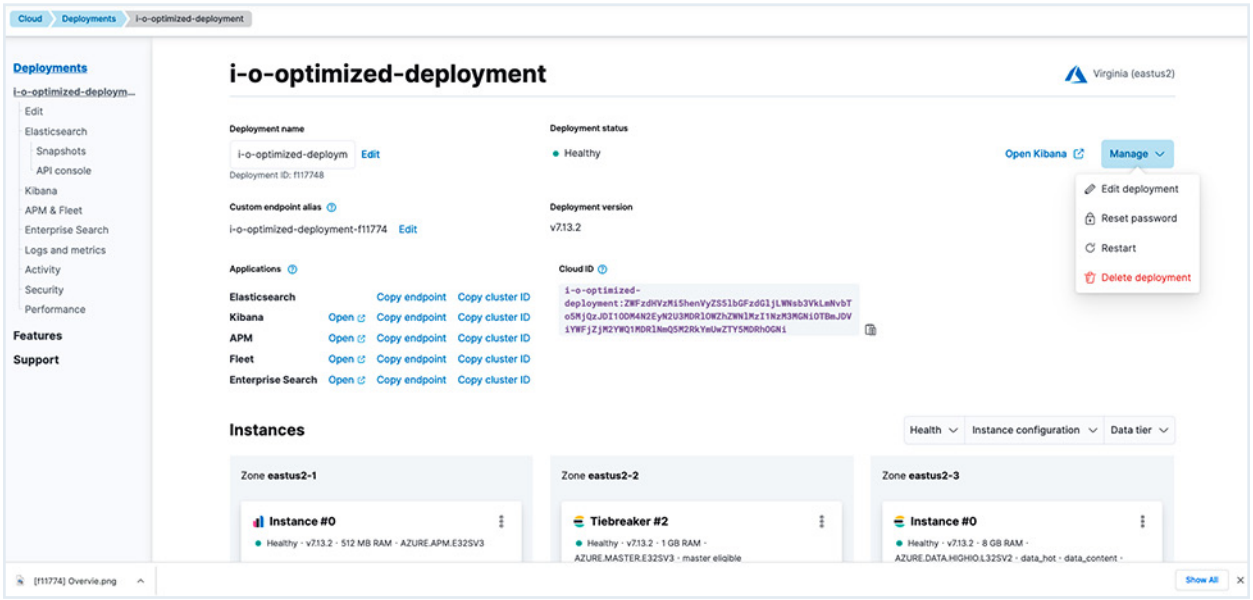
- Locate cloud ID
- Get login credentials
- Create AWS access key ID and access key

Locate cloud ID

You can find your cloud ID by navigating to cloud.elastic.co and selecting the relevant deployment.



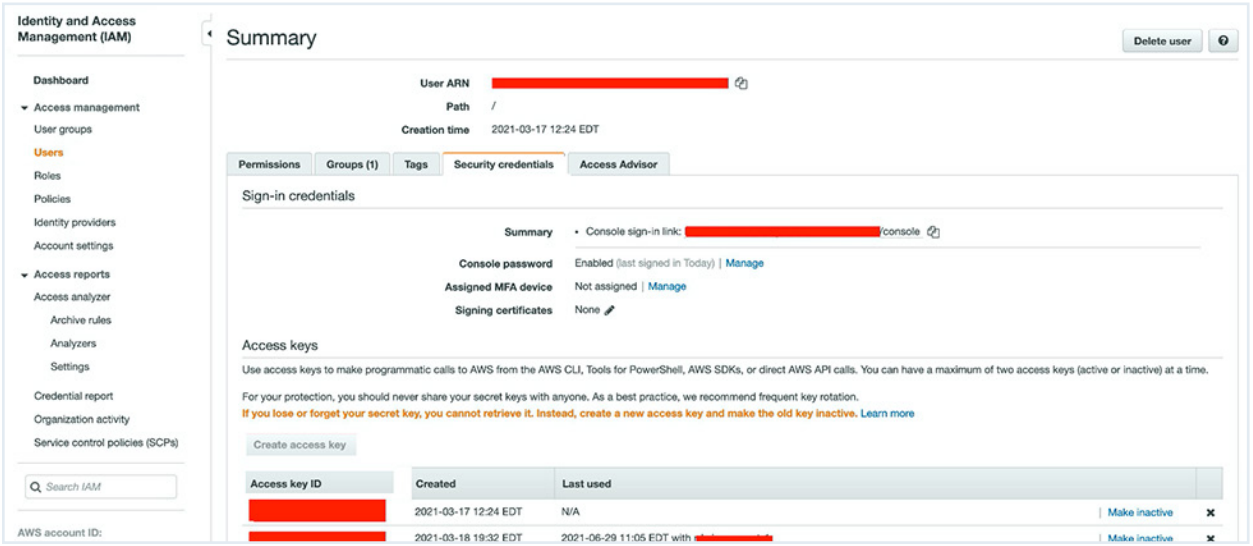
Get login credentials



When sending data to Elasticsearch you can use the default `Elastic` user and the password you were given when you created the cluster, or you can set up dedicated users and roles, with the least required privileges to accomplish the tasks. In this example, we'll be using the `Elastic` user and password provided.

If you didn't download or forgot the password, you can navigate to cloud.elastic.co and reset it by clicking selecting Manage.

Create AWS access key ID and access key



The AWS access key ID and access key are used to sign programmatic requests you make to AWS. To obtain these simply:

- Log in to AWS Identity and Access Management and open the IAM console at **https://console.aws.amazon.com/iam/**
- Select Users in the left navigation pane.
- Choose the user then select the Security credentials tab
- Under the Access Keys section, click Create access key and to view the access key pair select Show, then copy and save them to configure Filebeat and Metricbeat.

Appendix B – Filebeat Configuration

Below you will find a walk through for how to install Filebeat and enable AWS modules. The flow is as follows:

1. Set up an Amazon S3 Bucket and create and Amazon SQS queue
2. Download and install Filebeat
3. Connect to the Elastic Stack
   - This is where you'll need your cloud ID and cloud password for your Elastic deployment
4. Enable and configure your Filebeat module
5. Configure Filebeat to collect your AWS logs
   - This is where you'll need your AWS module code as well as your AWS access key ID and access key
6. Set up your pre-configured Kibana dashboards then start Filebeat
7. View and analyze data in Kibana

Step 1: Set up an Amazon S3 Bucket and create an Amazon SQS queue

To avoid significant lagging with polling all log files from each Amazon S3 bucket, Filebeat combines notification and polling together by using Amazon SQS for Amazon S3 notification when a new Amazon S3 object is created. Refer to this **Configuring S3 event notifications using Amazon SQS** to learn how to set up your Amazon S3 bucket and Amazon SQS queue.

Step 2: Download and install Filebeat

Download and install Filebeat. Use the commands that work for your system.

- For this example we will be using Linux commands. To find the latest version navigate to the **Filebeat documentation** then select Quick start: installation and configuration. Here you'll also find commands for other operating systems.

```
curl -L -O
https://artifacts.elastic.co/downloads/beats/filebeat/
filebeat-7.13.3-linux-x86_64.tar.gz
tar xzvf filebeat-7.13.3-linux-x86_64.tar.gz
```

Step 3: Connect to the Elastic Stack

Connections to Elasticsearch and Kibana are required to set up Filebeat. You will need to modify the configuration file, which is the filebeat.yml file.

This is where you will use the cloud ID and password you obtained. Specify the **cloud.id** of your Elasticsearch Service, and set **cloud.auth** (username:password) to a user who is authorized to set up Filebeat. For example:

```
cloud.id:
"staging:dxMtZWFzdC0xLmF3cy5mb3VuZC5pbyRjZWM2ZjI2MWE3NGJmMjRjZTMzYmI4ODExY
jg0Mjk0ZiRjNmMyY2E2ZDA0MjI0OWFmMGNjN2Q3YTllOTYyNTc0Mw=="
cloud.auth: "elastic:<elastic-password>"
```

For additional security, you can leverage the **Filebeat keystore** to obfuscate the credentials (username, password, cloud.id, etc.), and create dedicated users and roles with the least required permissions for the task. For this example, the default username and password you received when you created your deployment will be used. Additionally, you're using the default superuser as an example. For production, you'd want to set up users and roles with the **least privileges necessary** for the task.

Be sure to create a custom role to use for the deployed function. For example:

```
role: arn:aws:iam::123456789012:role/MyFunction
```

Make sure the custom role has the permissions required to run the function. For more information, see **IAM permissions required for deployment**.

Step 4: Enable and configure data collection modules

To enable the aws module navigate to the Filebeat directory and input the following command:

```
./filebeat modules enable aws
```

Step 5: Configure Filebeat to collect your AWS logs

Navigate to the AWS module configurations under modules.d directory in the aws.yml file. If the code for the integration you want is missing, you can locate the code in **Appendix E**.

You will also need your AWS credentials that you received from Appendix A to add to the aws.yml file at the top:

- access_key_id: "YOUR AWS ACCESS KEY ID"
- secret_access_key: "YOUR AWS ACCESS KEY"

If you prefer to use another method of authentication, refer to **AWS credential options** for more details.

Please refer to the following example below to add your AWS access key ID and access key:

```
module: aws
var.access_key_id: "XyzW4VIA6DCIEKDUNB"
var.secret_access_key: "p4873PxKFRB/enxV98PExUtQkEU82Coafo1w6"
```

Please refer to the following example below to add your IAM role:

```
module: aws
#AWS IAM Role to assume
var.role_arn: arniam::123456789012:role/test-mb
```

Note that you can also use the **Filebeat keystore** to obfuscate your AWS access key id and access key.

Step 6: Set up your pre-configured Kibana dashboard and start Filebeat

Filebeat comes with predefined assets for parsing, indexing, and visualizing your data. To load these assets:

- Make sure the user specified in filebeat.yml is **authorized to set up Filebeat** if you're not using the 'elastic' user (default user)
- From the installation directory, run:

```
./filebeat setup -e
```

Before starting Filebeat, modify the user credentials in filebeat.yml and specify a user who is authorized to publish events.

To start Filebeat, use the following commands:

```
sudo chown root filebeat.yml
sudo chown root modules.d/aws.yml
sudo ./filebeat -e -c filebeat.yml &
```

Step 7: View and analyze data in Kibana

Filebeat comes with pre-built Kibana dashboards and a dedicated Logs application for visualizing, searching and filtering log data, as well as easy to configure anomaly detection. You loaded the dashboards earlier when you ran the setup command.

To launch Kibana:

- **Log in** to you Elastic Cloud account
- Navigate to the Kibana endpoint in your deployment to view and analyze your data

Appendix C – Metricbeat Configuration

Below you will find a walk through for how to install Metricbeat and enable AWS modules. The flow is as follows:

1. Download and install Metricbeat.
2. Connect to the Elastic Stack
   - This is where you'll need your cloud ID and cloud password for your Elastic deployment
3. Enable and configuring data collection modules
4. Configure Filebeat to collect AWS metrics
   - This is where you'll need your AWS module code as well as your AWS access key ID and access key
5. Set up your pre-configured Kibana dashboards then start Metricbeat
6. View and analyze data in Kibana

Step 1: Download and install Metricbeat

Download and install Metricbeat. Use the commands that work for your system.

For this example, we will be using Linux commands. To find the latest version navigate to the **Metricbeat documentation** then select Quick start: installation and configuration. Here you'll also find commands for other operating systems.

```
curl -L -O https://artifacts.elastic.co/downloads/beats/
metricbeat/metricbeat-7.13.4-linux-x86_64.tar.gz
tar xzvf metricbeat-7.13.4-linux-x86_64.tar.gz
```

Step 2: Connect to the Elastic Stack

When you configure Metricbeat, you have to edit the configuration file, metricbeat.yml.

This is where you will use the cloud ID and password you obtained. Specify the **cloud.id** of your Elasticsearch Service, and set **cloud.auth** (username:password) to a user who is authorized to set up Metricbeat. For example:

```
cloud.id:
"staging:dxMtZWFzdC0xLmF3cy5mb3VuZC5pbyRjZWM2ZjI2MWE3NGJmMjRjZTMzYmI4ODExY
jg0Mjk0ZiRjNmMyY2E2ZDA0MjI0OWFmMGNjN2Q3YTllOTYyNTc0Mw=="
cloud.auth: "elastic:<elastic-password>"
```

For additional security, you can leverage the **Metricbeat keystore** to obfuscate the credentials (username, password, cloud.id, etc.), and create dedicated users and roles with the least required permissions for the task. For this example, the default username and password you received when you created your deployment will be used. Additionally, you're using the default superuser as an example. For production, you'd want to set up users and roles with the **least privileges necessary** for the task.

Be sure to create a custom role to use for the deployed function. For example:

```
role: arn:aws:iam::123456789012:role/MyFunction
```

Make sure the custom role has the permissions required to run the function. For more information, see **IAM permissions required for deployment**.

Step 3: Enable and configure data collection modules

When you configure Metricbeat, you need to specify which modules to run. Metricbeat uses modules to collect metrics. To enable the aws config in the modules.d directory, input the following command:

```
./metricbeat modules enable aws
```

Step 4: Configure Metricbeat to collect your AWS metrics

Navigate to the AWS module configurations under modules.d directory in the aws.yml file. If the code for the integration you want is missing, you can locate the code in **Appendix E**.

You will also need your AWS credentials to add to the aws.yml file at the top:

- access_key_id: "YOUR AWS ACCESS KEY ID"
- secret_access_key: "YOUR AWS ACCESS KEY"

If you prefer to use another method of authentication, refer to **AWS credential options** for more details.

Please refer to the following example below to add your AWS access key ID and access key:

```
module: aws
access_key_id: "XyzW4VIA6DCIEKDUNB"
secret_access_key: "p4873PxKFRB/enxV98PExUtQkEU82Coafo1w6"
```

Please refer to the following example below to add your IAM role:

```
module: aws
#AWS IAM Role to assume
role_arn: arniam::123456789012:role/test-mb
```

Note that you can also use the **Metricbeat keystore** to obfuscate your AWS access key id and access key.

Step 5: Set up your preconfigured Kibana dashboards and start Metricbeat

Metricbeat comes packaged with example Kibana dashboards, visualizations, and searches for visualizing AWS metric data in Kibana, and easy to configure alerting and anomaly detection.

- Make sure the user specified in metricbeat.yml is **authorized to set up Metricbeat** if you're not using the 'elastic' user (default user)
- From the installation directory, run:

  ```
  ./metricbeat setup -e
  ```

To start Metricbeat, use the following commands:

```
sudo chown root metricbeat.yml
sudo chown root modules.d/aws.yml
sudo ./metricbeat -e -c metricbeat.yml &
```

Step 6: View and analyze data in Kibana

Metricbeat comes with pre-built Kibana dashboards and a dedicated application for visualizing metric data. You loaded the dashboards earlier when you ran the setup command.

To launch Kibana:

- **Log in** to you Elastic Cloud account
- Navigate to the Kibana endpoint in your deployment

Appendix D – Functionbeat Configuration

Below you will find a walk-through for how to install Functionbeat and enable AWS modules. The flow is as follows:

1. Download and installing Functionbeat
2. Connect to the Elastic Stack
   - This is where you'll need your cloud ID and cloud password for your Elastic deployment
3. Configure cloud functions
   - This is where you'll need your AWS module code as well as your AWS access key ID and access key
4. Set up assets and deploy Functionbeat
5. Build out your Kibana dashboards for analysis

Step 1: Download and install Functionbeat

Download and install Metricbeat. Use the commands that work for your system.

- For this example, we will be using Linux commands. See **documentation** for commands for other operating systems.

```
curl -L -O https://artifacts.elastic.co/downloads/beats/
functionbeat/functionbeat-7.13.4-linux-x86_64.tar.gz
tar xzvf functionbeat-7.13.4-linux-x86_64.tar.gz
```

Step 2: Connect to the Elastic Stack

Connections to Elasticsearch and Kibana are required to use Filebeat. You will need to modify the configuration file, functionbeat.yml.

This is where you will use the cloud ID and password you obtained. Specify the **cloud.id** of your Elasticsearch Service, and set **cloud.auth** (password) to a user who is authorized to set up Functionbeat. For example:

```
cloud.id:
"staging:dxMtZWFzdC0xLmF3cy5mb3VuZC5pbyRjZWM2ZjI2MWE3NGJmMjRjZTMzYmI4ODExY
jg0Mjk0ZiRjNmMyY2E2ZDA0MjI0OWFmMGNjN2Q3YTllOTYyNTc0Mw=="
cloud.auth: "functionbeat_setup:YOUR_PASSWORD"
```

Be sure to create a custom role to use for the deployed function. For example:

```
role: arn:aws:iam::123456789012:role/MyFunction
```

Make sure the custom role has the permissions required to run the function. For more information, see **IAM permissions required for deployment**.

Step 3: Configure your cloud functions

Before deploying Functionbeat to AWS, you need to specify details about the cloud functions you plan to deploy, including the function names and types and the triggers that will cause the function to execute.

In functionbeat.yml, configure the functions that you want to deploy. The configuration settings vary depending on the type of function and cloud provider you're using. If the code for the integration you want is missing, you can locate the code in **Appendix E**. This section provides an example configuration.

```
functionbeat.provider.aws.endpoint: "s3.amazonaws.com"
functionbeat.provider.aws.deploy_bucket: "functionbeat-deploy"
functionbeat.provider.aws.functions:
  - name: cloudwatch
    enabled: true
    type: cloudwatch_logs
    description: "lambda function for cloudwatch logs"
    triggers:
      - log_group_name: /aws/lambda/my-lambda-function
```

You will also need your AWS credentials. Configure your AWS credentials at the top of the functionbeat.yml file:

- access_key_id: "YOUR AWS ACCESS KEY ID"
- secret_access_key: "YOUR AWS ACCESS KEY"

If you prefer to use another method of authentication, refer to **AWS credential options** for more details.

Please refer to the following example below:

```
module: cloudwatch
enabled: true
access_key_id: "XyzW4VIA6DCIEKDUNB"
secret_access_key: "p4873PxKFRB/enxV98PExUtQkEU82Coafo1w6"
```

Step 4: Set up assets and deploy Functionbeat

Functionbeat comes with predefined assets for parsing, indexing, and visualizing your data. To load these assets:

Make sure the user specified in functionbeat.yml is **authorized to set up Functionbeat**. From the installation directory, run:

```
./functionbeat setup -e
```

To deploy the cloud functions, use the following commands:

```
./functionbeat -v -e -d "*" deploy cloudwatch
```

Now the function is deployed to AWS and ready to send log events to the configured output.

Step 5: Build out your Kibana dashboards for analysis

Now you can build out your dashboards in Kibana. To learn how to view and explore your data, see the **Kibana User Guide**. To launch Kibana:

- **Log in** to your Elastic Cloud account
- Navigate to the Kibana endpoint in your deployment.

Appendix E – Additional resources

For advanced AWS configurations see these documents for:

- **Filebeat**
- **Metricbeat**
- **Functionbeat**

elastic

Search. Observe. Protect.

Elastic makes data usable in real time and at scale for enterprise search, observability, and security. Elastic solutions are built on a single free and open technology stack that can be deployed anywhere to instantly find actionable insights from any type of data — from finding documents, to monitoring infrastructure, to hunting for threats. Thousands of organizations worldwide, including Cisco, Goldman Sachs, Microsoft, The Mayo Clinic, NASA, The New York Times, Wikipedia, and Verizon, use Elastic to power mission-critical systems. Founded in 2012, Elastic is publicly traded on the NYSE under the symbol ESTC. Learn more at elastic.co.

AMERICAS HQ
800 West El Camino Real, Suite 350, Mountain View, California 94040
General +1 650 458 2620, Sales +1 650 458 2625

info@elastic.co