Cribl LogStream User Deployment Guide for AWS

Version 1.4, November 2021

Table of Contents

This Quick Start was created by Cribl in collaboration with Amazon Web Services (AWS). [Quick Starts](#) are automated reference deployments that use AWS CloudFormation templates to deploy key technologies on AWS, following AWS best practices.

Overview

Cribl's mission is to unlock the value of all your observability data, regardless of source or destination. Cribl LogStream is a vendor neutral, purpose built observability pipeline that can parse, restructure, and enrich data in flight – ensuring that you get the right data, where you want, in the formats you need. This deployment guide was written to help customers properly deploy Cribl LogStream in their Amazon Web Services environment. This guide will detail the steps to deploy the AWS CloudFormation template following AWS best practices.

What you'll build

Use this Quick Start to automatically setup the following architecture on AWS:
- A VPC configured with public and private subnets according to AWS best practices, to provide you with your own virtual network on AWS.*
- An Amazon EC2 Auto Scaling group with a configurable number of instances.
- An S3 bucket to collect and send data for Cribl LogStream
- An EC2 IAM role granting proper rights to the data available in the AWS account

* The template that deploys the Quick Start into an existing VPC skips the components marked by asterisks and prompts you for your existing VPC configuration.

How to deploy

To add Cribl LogStream to your environment on AWS, follow the instructions in the deployment guide. The deployment process takes about five minutes and includes these steps:

1. If you don't already have an AWS account, sign up at https://aws.amazon.com, and sign into your account.
2. Subscribe to the free offering of Cribl LogStream on the AWS Marketplace
3. Launch the Quick Start by choosing from the following options. Before you create the stack, choose the Region from the top toolbar.
    a. Deploy into a new VPC
    b. Deploy into an existing VPC
4. Test the deployment
5. Post-deployment steps.

Amazon may share user-deployment information with the AWS Partner that collaborated with AWS on the Quick Start.

Cost and licenses

You are responsible for the cost of the AWS services and any third-party licenses used while running this Quick Start reference deployment. There is no additional cost for using the Quick Start.

The AWS CloudFormation templates for this Quick Start include configuration parameters that you can customize. Some of these settings, such as instance type, affect the cost of deployment. For cost estimates, see the pricing pages for each AWS service you use. Prices are subject to change.

> Tip: After you deploy the Quick Start, create AWS Cost and Usage Reports to track costs associated with the Quick Start. These reports deliver billing metrics to an Amazon Simple Storage Service (Amazon S3) bucket in your account. They provide cost estimates based on usage throughout each month and aggregate the data at the end of the month. For more information about the report, see What are AWS Cost and Usage Reports?

Prerequisites and Requirements

Users who want to run Cribl LogStream in AWS should have the following prerequisites available, and meet these requirements:

Time

The deployment itself will take less than 5 minutes for the Single-Instance, and less than 10 minutes for the Distributed Deployment.

Product License

An AWS Marketplace subscription is required for production use of the Cribl LogStream AMI. The default deployment of Cribl LogStream is the Free option.

As an alternative, you can Bring Your Own License (BYOL) – subscribe through AWS Marketplace, and apply an existing LogStream license. This is generally appropriate for deployments with extremely heavy usage, abnormal access patterns, or other concerns. Please contact us directly to order a license at sales@cribl.io.

AWS Account

You must have an AWS account set up. If you don't, we recommend that you visit the following site: https://aws.amazon.com/getting-started/

AWS Identity and Access Management (IAM) Entity

Create an IAM user or role. Your IAM user should have a policy that allows AWS CloudFormation actions. Do not use your root account to deploy the CloudFormation template.

Beyond AWS CloudFormation actions, IAM users who create or delete stacks will require additional permissions that depend on the stack template. This deployment requires permissions to all services listed in the following section. *Reference:* https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/using-iam-template.html

Knowledge Requirements

You must have a working knowledge of the following AWS Services:
- Amazon EC2
- SQS
- S3
- IAM
- Kinesis
- CloudFormation templates

You will not need to request an increase in limits for your AWS Account for this deployment. (More information on proper policy and permissions here: https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/using-iam-template.html)

Note: Individuals possessing the AWS Associate Certificate should have a sufficient depth of knowledge to deploy the resources specified in this guide.

Architecture

This will deploy a single EC2 instance and configure the AWS services and resources listed above. We will be using this deployment for our quick start guide.

Single-AZ Architecture Diagram (Development and Testing)
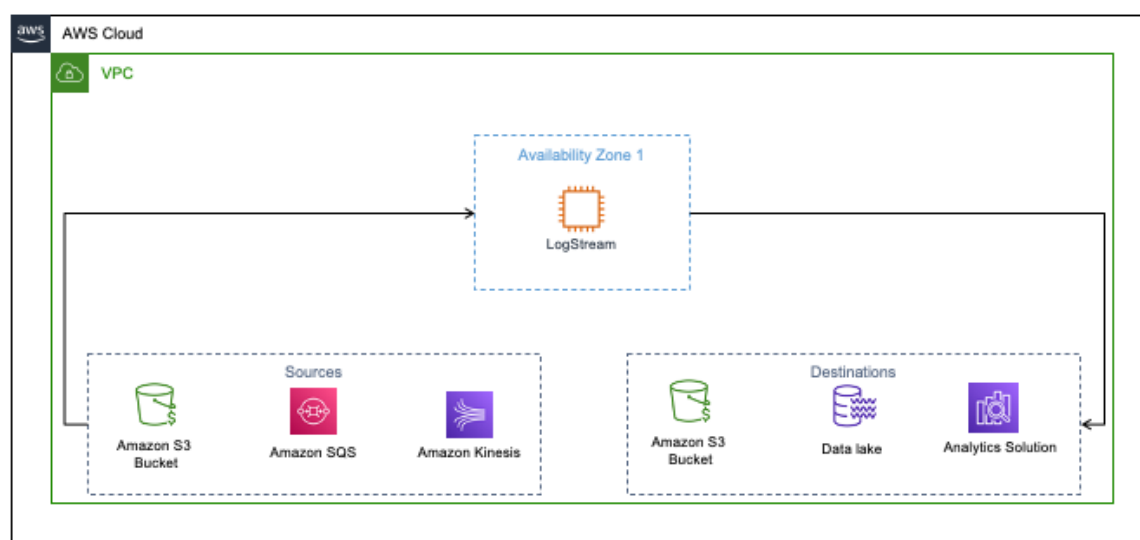
Cribl LogStream Single-Instance is designed for customers who want to validate to make sure our solution will fulfill their requirements. This solution can also be used for testing and development purposes. If you have a small workload, this solution might be sufficient for production; its capacity will be limited by the amount of resources available. Please refer to Cribl's latest sizing and scaling guide: https://docs.cribl.io/docs/scaling .

Deployment

1. Sign into your AWS Account (https://aws.amazon.com/console).
2. Navigate to Cribl LogStream Single Instance and then click on "Continue to Subscribe."
3. Select your deployment method:

VPC Status	ARM64	x86_64
Deploy in an existing VPC	Deploy in an Existing VPC	Deploy in an Existing VPC
Deploy in a new VPC	Deploy in a New VPC	Deploy in a New VPC
Deploy in a new VPC with Logging	Deploy in a new VPC with Logging	Deploy in a new VPC with Logging

4. Deploy the stack in your environment, make sure to check the region as this defaults to **Oregon (us-west-2).**
5. Log into Cribl LogStream with the credential supplied in the "Outputs" tab on your CloudFormation stack.

There are some additional options when deploying Cribl LogStream in your environment.

Tips for Deploying in Your AWS Environment

If this is a new AWS account, you will need to create the following components:

- An SSH key pair to log into your EC2 instance. Please follow this guide for help on creating your key pair.

Security Group CIDR Tips

- Web Access CIDR should be open to a range of IP's that you want to grant access. Or, you can open it to the world if you have no restrictions on public access to the web port (TCP 9000). An example CIDR for open access would be 0.0.0.0/0.
- SSH Access CIDR should be restricted to your own IP address, or to a small range of IP's that you want to grant SSH access to your system. An example

CIDR for your own IP would be <your_ip_here>/32, where your IP Address would replace <your_ip_here>.

- Advanced Settings amild is used only if you are issued this credential by Cribl Support.

Deployment Region Options

By default this CloudFormation template will deploy in **Oregon** (**us-west-2**), but if you want to change the region simply change the region in your CloudFormation drop down and then in the template itself update the **QSS3BucketRegion variable from us-west-2 to your region of choice.**

Use Cases

VPC Flow Logs

This use case is to get VPC Flow logs into your system of record and then convert the log events to metric data. This is based on the excellent blog post by one of the Cribl Co-Founders Dritran Bitincka. There are two ways to collect VPCFlow Logs, depending on cost and latency concerns, you might want to go one way or the other. First, if cost is an issue and you don't mind receiving the data after a few minutes, collecting the logs via S3 is the best approach. However, if you need the logs more quickly, then pushing the data from CloudWatch Logs to an HTTP endpoint would be the recommended approach. There is a higher cost for going down this route which is explained here.

CloudWatch Streaming Metrics

This use case will cover collecting metrics from AWS CloudWatch using their streaming feature via S3 buckets. Follow the steps in the AWS Post to enable your CloudWatch Streaming Metrics.
https://aws.amazon.com/blogs/aws/cloudwatch-metric-streams-send-aws-metrics-to-partners-and-to-your-apps-in-real-time/

CloudTrail logs

This use case is to get CloudTrail logs into your system of record. We will go over some tips to help reduce the volume of these events by removing redundant or `null` values and other events that take up significant space while not giving any value. In this blog post Helping Threat Hunters While Staying Compliant: Categorizing and Scoring AWS CloudTrail Events in Real-Time, we will go into detail on how to help reduce the amount of redundant and high volume, low value data from CloudTrail logs.

Collect and send to an S3 bucket

This use case will show you how to setup Cribl LogStream to send data from any source to an S3 bucket. The CloudFormation template from the AWS Marketplace listing automatically creates an S3 bucket with an IAM policy that allows reading from and writing to the bucket. The AWS side of this deployment has already been configured, we only need to setup a route and pipeline in Cribl LogStream to send the data to the bucket.

Security

In this section, we discuss the Cribl LogStream default configuration deployed according to this guide, AWS general best practices, and options for securing your solution on AWS.

IAM Role
The CloudFormation template creates an S3 bucket, and an IAM EC2 role that allows your instances to read and write to that specific S3 bucket. The role also allows the EC2 instance to read and collect data from Kinesis Streams within the AWS account.

Encrypting Data at Rest

By default, no sensitive data is stored on the LogStream Nodes, so it is not necessary to encrypt the data at rest. If required, then data at rest can be encrypted using LUKS full-disk encryption.

Encrypting Data in Motion

With Cribl LogStream, you can encrypt fields or patterns within events in real time, by using C.Crypto.encrypt() in a Mask function. The Mask function accepts multiple replacement rules, and multiple fields to apply them to.

A Match Regex defines the pattern of content to be replaced. The Replace Expression is a JS expression (or literal) to replace matched content. The C.Crypto.encrypt() method can be used here to generate an encrypted string from a value passed to it. For additional information, please refer our documentation on encryption of data in motion (https://docs.cribl.io/docs/securing-data-encryption).

Amazon Certificate Manager SSL/TLS Certificates

AWS Certificate Manager (ACM) is a service that lets you easily provision, manage, and deploy Secure Sockets Layer/Transport Layer Security (SSL/TLS) certificates for use with AWS services. SSL/TLS certificates provisioned through AWS Certificate Manager are free.

If you don't already have an SSL/TLS certificate for your domain name, Cribl recommends that you request one using ACM. For more information about requesting an SSL/TLS certificate using ACM, please read the AWS Certificate Manager User Guide.

Use ACM to request a certificate or import a certificate into ACM. To use an ACM certificate with CloudFront (optional input parameter), you must request or import the certificate in the US East (N. Virginia) region. To use an ACM certificate with Amazon ELB – Application Load Balancer (optional input parameter), you must request or import the certificate in the region where you create the CloudFormation stack. After you validate ownership of the domain names in your certificate, ACM provisions the certificate. Use the ACM certificate Amazon Resource Name (ARN) as the leader template's optional Amazon CloudFront and/or Public ALB ACM certificate input parameters.

You can secure Cribl LogStream's API and UI access by configuring SSL. To do so, you can use your own private keys and certs, or you can generate a pair with OpenSSL, as shown here:

```
openssl req -nodes -new -x509 -newkey rsa:2048 -keyout myKey.pem -out
myCert.pem -days 420
```

This command will generate both a self-signed cert (certified for 420 days), and an unencrypted, 2048-bit RSA private key.

In the LogStream UI, you can configure the key and cert via Settings > Encryption Keys and Settings > Certificates. Alternatively, you can edit the local/cribl.yml file's `api` section to directly set the privKeyPath and certPath attributes. For example:

```
cribl.yml
api:
  host: 0.0.0.0
  port: 9000
  disabled : false
  ssl:
    disabled: false
    privKeyPath: /path/to/myKey.pem
    certPath: /path/to/myCert.pem
...
```

Logging/Auditing

Cribl LogStream stores access and error logs to the locally provisioned EBS volume. This data can also be streamed to an S3 bucket or a system of record. If you want to visualize logs or metrics about the stack, please navigate to the LogStream Web UI's Monitoring tab. For more information about in-product monitoring, please refer to:
https://docs.cribl.io/docs/getting-started-guide#monitor-data-throughput

Costs

This guide will create the AWS resources outlined in the guide's [Deployment Assets](#) section. The following assets are required to provide a functional platform for the Single-Instance or Distributed Deployment:

1. Amazon EC2 Instance(s)
2. S3 bucket

Distributed deployment will include a higher number of EC2 instances to increase performance and fault tolerance. Here is a sample calculation with the instances listed for under 400GB/day Single Instance Deployment:
https://calculator.s3.amazonaws.com/index.html#r=IAD&s=EC2&key=files/calc-d446b1a796b6d20061e8390a99a98ce730bb546d&v=ver20201028qG

Here is a sample calculation listed for over 1TB/day in a Distributed Deployment:
https://calculator.s3.amazonaws.com/index.html#r=IAD&s=EC2&key=files/calc-858dc613b35a4d528044d47c9113cb36b0c077f8&v=ver20201028qG

NOTE: Storage and data transfer costs are not included, as these vary depending on configuration. Please consult [AWS Pricing](#) for the latest information.

Below is a table of our license cost mapped to Amazon EC2 instance sizes in all supported regions:

Instance Type	Hourly Enterprise	Annual Enterprise	Hourly Standard	Annual Standard
c5*.large	$2.740	$20,000	$1.315	$9,600
c5*.xlarge	$8.219	$60,000	$3.288	$24,000
c5*.2xlarge	$13.233	$96,600	$6.521	$47,600
c5*.4xlarge	$24.658	$180,00	$11.507	$84,000

Sizing

Cribl's documentation includes a section dedicated to [sizing and scaling out your deployment](#). The **minimum** instance for production work loads would be a C5.2xlarge / C6g.2xlarge, while the **recommended** instance type would be a C5.4xlarge / C6g.4xlarge. Cribl LogStream is generally CPU-bound, and each worker process will consume 1 physical core or 2 vCPUs. That one physical core can handle up to 400GB/day of IN+OUT throughput.

x86_64 Each Node can handle GB / Day				
	Test Environment (>100GB)	100GB - 400GB	401GB - 1.2TB	1.2TB - 2.4TB
EC2	c5*.large	c5*.xl	c5*.2xl	c5*.4xl
EBS	GP2	GP2	GP2	GP2

Allocate 1 physical core (2 vCPUs) for each 400GB/day of IN+OUT throughput. So, to estimate the number of cores needed: Sum your expected input and output volume, then divide by 400GB.

- Example 1: 100GB IN -> 100GB out to each of 3 destinations = 400GB total = 1 physical core.
- Example 2: 3TB IN -> 1TB out = 4TB total = 10 physical cores.
- Example 3: 4 TB IN -> full 4TB to Destination A, plus 2 TB to Destination B = 10TB total = 25 physical cores.

ARM64 Each Node can handle GB / Day				
	Test Environment (>100GB)	100GB - 400GB	401GB - 1.2TB	1.2TB - 2.4TB
EC2	c6*.large	c6*.xl	c6*.2xl	c6*.4xl
EBS	GP2	GP2	GP2	GP2

Here, 1 physical core = 1 vCPU, but overall throughput is ~20% higher than a corresponding Intel or AMD vCPU. So:

Allocate 1 physical core (1 vCPU) for each 240GB/day of IN+OUT throughput. To estimate the number of cores needed: Sum your expected input and output volume, then divide by 240GB.

- Example 1: 100GB IN -> 100GB out to each of 3 destinations = 400GB total = 2 physical cores.
- Example 2: 3TB IN -> 1TB out = 4TB total = 17 physical cores.
- Example 3: 4 TB IN -> full 4TB to Destination A, plus 2 TB to Destination B = 10TB total = 42 physical cores.

Deployment Assets

Deployment Options

The Cribl LogStream CloudFormation template provides two deployment options. The Distributed Deployment option provides a scalable, highly available, redundant architecture that is suitable for production deployments. The Single-Instance deployment option provides a lower-cost alternative that is suitable for development or test workloads.

Deployment Assets (Recommended for Production)

The Cribl LogStream deployment is executed via a CloudFormation template that receives input parameters, and passes them to the appropriate nested templates, which are executed in order based on conditions and dependencies.

AWS Resources Created:
- Amazon EC2 Instances
- Security Groups
- Auto Scaling group (For the Worker Nodes)

Cloudformation Template Input Parameters

leader Node
- Amazon EC2 Key Name Pair for leader Node
- leader Node EC2 Instance Type

Worker Node
- Amazon EC2 Key Name Pair for Worker Node(s)
- Worker Node EC2 Instance Type
- Worker Count (Number of Worker Nodes to be deployed)

Network
- VPC ID
- Subnet IDs (Minimum 2 Subnet IDs)
- Web Access CIDR Block (to access Web UI)
- SSH Access CIDR Block (to access instances)

Optional Support
- Custom leader AMI ID (Optional)
- Worker AMI ID (Optional)

leader Template

The leader template receives all the input parameters. and passes them to the appropriate nested templates, which are executed in order based on dependencies.

Stack Creation

There is one output URL for the leader template. The **logstreamWebUrl** will take you to your new Cribl LogStream deployment's leader Mode Web UI.
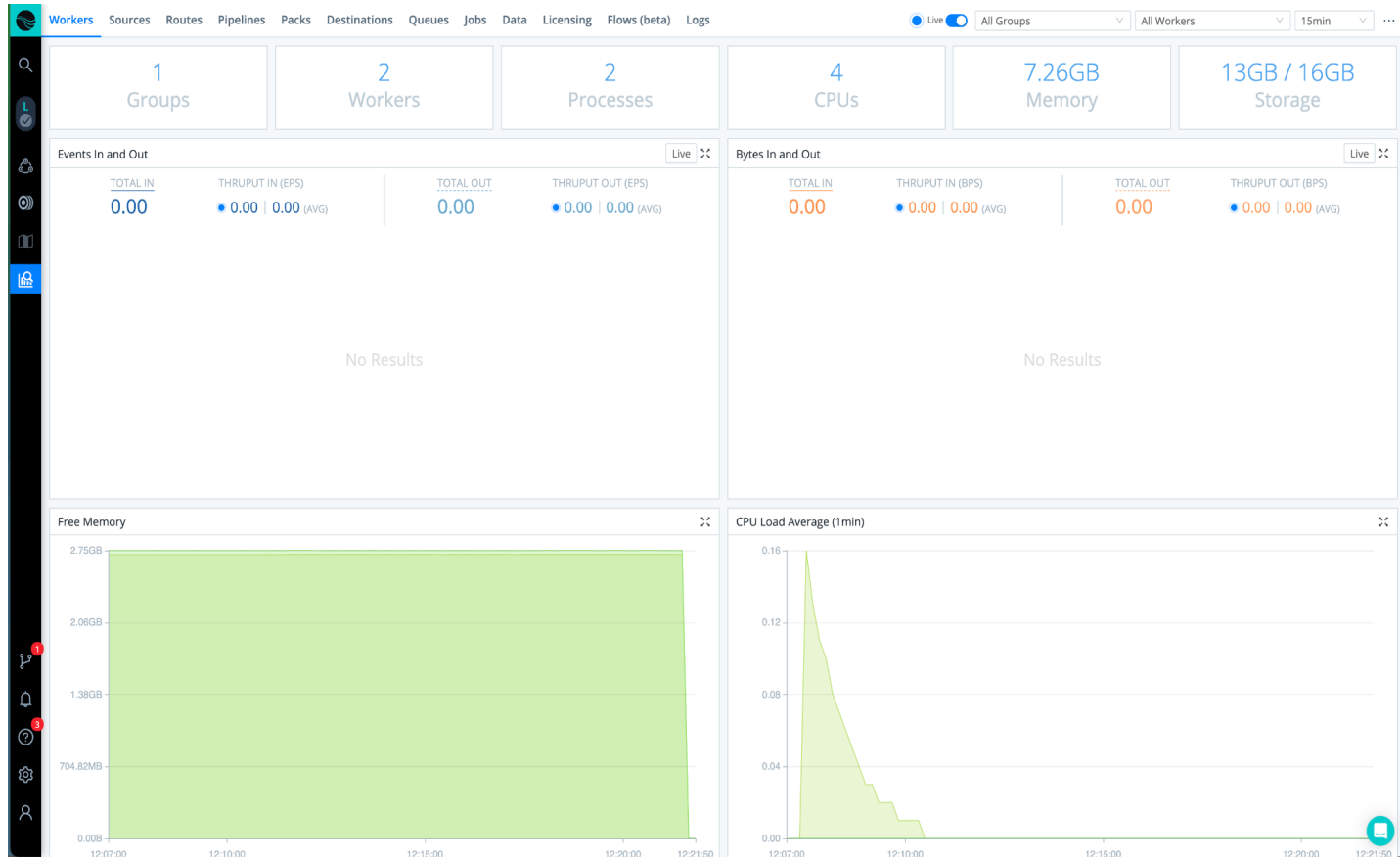
Testing and Deployment

Customers can leverage the built-in Cribl LogStream Monitoring page for both single-instance and distributed deployments. For details, see our documentation: https://docs.cribl.io/docs/monitoring.

Distributed Deployment Testing

1. Log into the leader Node's Web UI: http://cribl_leader_node_ip:9000, username **admin**, password *ec2-instance-id.*
2. Click on **Workers** and see that they are all showing a status:

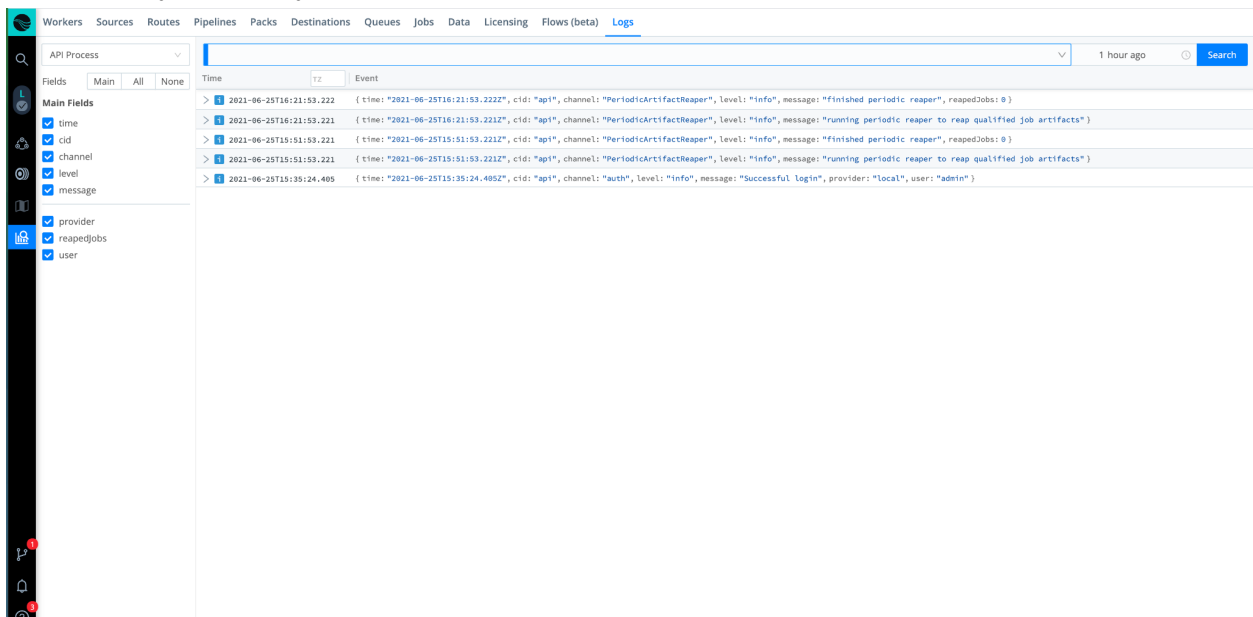3. Click on Monitoring to see the number of Workers and their status:

Single-Instance Deployment Testing

1. Log into the Cribl LogStream instance's Web UI: http://cribl_instance:9000, username **admin,** password *ec2-instance-id*.
2. Click on Monitoring, then make sure all of your Stats are registering events and metrics:
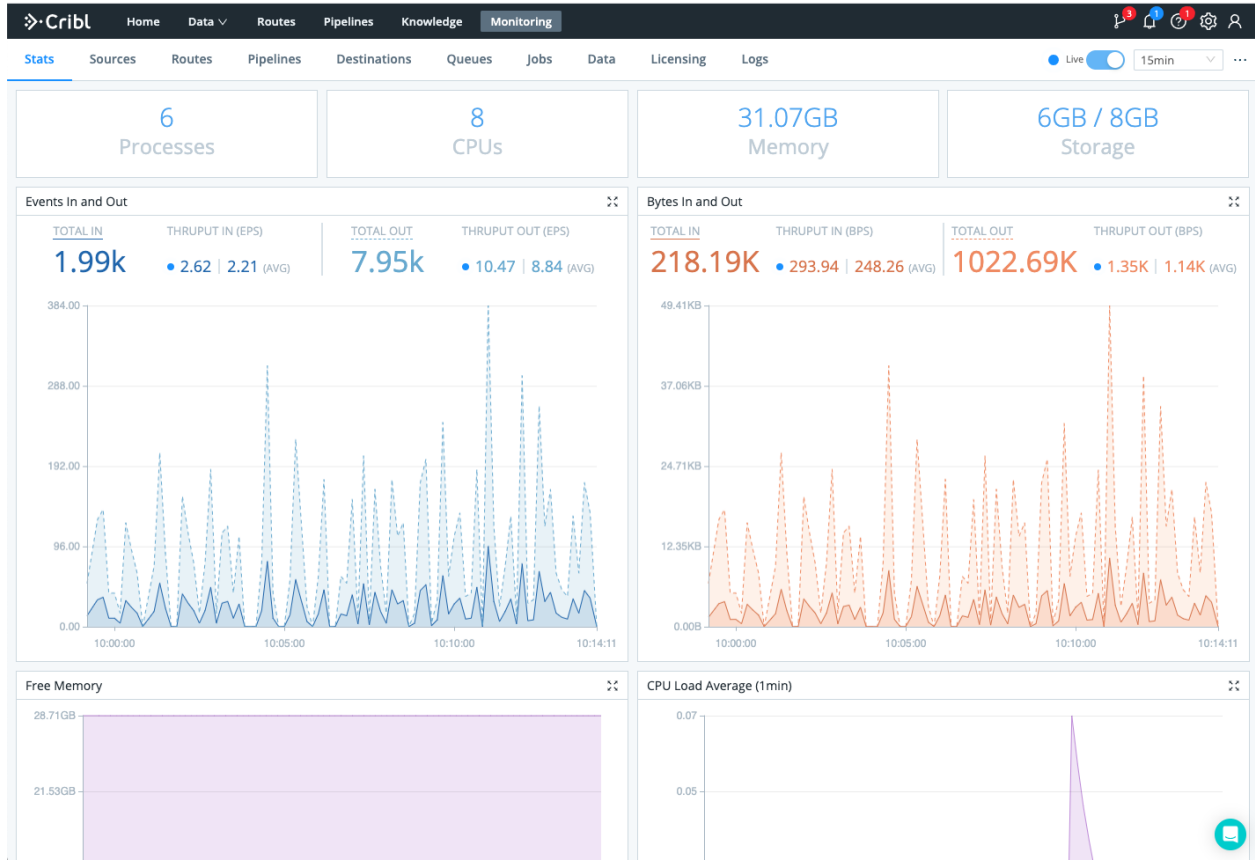
3. On the Monitoring submenu, click Logs to see if there are any potential errors or issues with your deployment.

Backup and Recovery

Backup

Cribl LogStream integrates with Git clients and remote repositories to provide version control of LogStream's configuration. This integration offers backup and rollback for single-instance and distributed deployments.

These options are separate from the Git repo responsible for version control of Worker configurations, located on the leader Node in distributed deployments. We cover all these options and requirements in our documentation: https://docs.cribl.io/docs/version-control.

Instance Failure

leader Nodes

In case a leader Node fails, the Worker Nodes will continue to process data, but no updates will be sent to the Worker Nodes. The Worker Nodes will work independently of the leader Node until a new leader Node comes back online.

Worker Nodes

Workers will periodically (every 10 seconds) send a heartbeat to the leader. This heartbeat includes information about themselves, and a set of current system metrics. The heartbeat payload includes facts – such as hostname, IP address, GUID, tags, environment variables, current software/configuration version, etc. – that the leader tracks with the connection.

The failure of a Worker Node to successfully send two consecutive heartbeat messages to the leader will cause the respective Worker to be removed from the Workers page in the leader's UI until the leader receives a heartbeat message from the affected Worker.

When a Worker Node checks in with the leader:

- The Worker sends a heartbeat to the leader.
- The leader uses the Worker's facts and Mapping Rules to map it to a Worker Group.
- The Worker Node pulls its Group's updated configuration bundle, if necessary.
- Please follow these instructions for Load Balancer and Auto Scaling group settings: https://docs.cribl.io/docs/deploy-distributed#auto-scaling-workers-and-load-balancing-incoming-data

Availability-Zone Failure

Worker Nodes in failed availability-zones will be replaced by the Auto Scaling group to meet the load of data being processed. In < 15 minutes, the Auto Scaling group will re-create the number of instances required to handle the incoming traffic.

During a restart, to minimize ingestion disruption and to increase availability of network ports, Worker Processes on a Worker Node are restarted in a rolling fashion. Specifically, 20% of running processes – with a minimum of one process – are restarted at a time. A Worker Process must come up and report as started before the next one is restarted. This rolling restart continues until all Processes have restarted. If a Worker Process fails to restart, the nodes' configurations will be rolled back.

Region Failure

The architecture outlined in this guide does not natively support multi-region operations. Therefore, you must explicitly enable cross-region replication, and must back up your configurations to a git repo that is available in multiple regions.

Cribl LogStream can be quickly deployed in a new region, and the configurations can be bootstrapped from the git repository, but it will take time for the data streams to cut over, due to DNS / networking updates.

The time to restore varies with resource availability and network cutover, but a <4-hour recovery time objective (RTO) and <24-hour recovery point objective (RPO) are generally possible. To restore operations in another region:

- Leverage the standard AWS CloudFormation templates to re-create the architecture on-demand.
- Restore the configurations, using the git repository with all the configuration commits.
- Update DNS / output settings and point data collectors to the new stack.

General Failure Considerations

Make sure to back up LogStream configurations to a git repository, to make recovery and scaling out easier. Please follow the guidance for distributed and single-instance deployments in our documentation: https://docs.cribl.io/docs/version-control

Routine Maintenance

Cribl LogStream leverages git to push updates to the leader Node. The upgrade process is initially the same for single-instance and distributed deployments, although distributed deployments require extra steps to upgrade the Worker Nodes. For detailed information, please refer to the documentation: https://docs.cribl.io/docs/upgrading

Cribl recommends following the AWS best practices for ongoing tasks, including:

- Access key rotation
- Service limit evaluations
- Certificate renewals

Standalone/Single-Instance

This path requires upgrading only the single/standalone node:

1. Stop Cribl LogStream.
2. Uncompress the new version on top of the old one.
   On some Linux systems, tar might complain with: "cribl/bin/cribl: Cannot open: File exists." In this case, please remove the cribl/bin/cribl directory if it's empty, and untar again. If you have custom functions in cribl/bin/cribl, please move them under $CRIBL_HOME/local/cribl/functions/ before untarring again.
3. Restart LogStream.

Distributed Deployment

For a distributed deployment, the order of upgrade is: Upgrade first the leader Node, then upgrade the Worker Nodes, then commit and deploy the changes on the leader.

Upgrade the leader Node
1. Commit and deploy your desired last version. (This will be your most recent checkpoint.)
   ○ Optionally, git push to your configured remote repo.
2. Stop Cribl LogStream.
   ○ Optional, but recommended: Back up the entire $CRIBL_HOME directory.
   ○ Optional: Check that the Worker Nodes are still functioning as expected. In absence of the leader Node, they should continue to work with their last deployed configurations.
3. Uncompress the new LogStream version on top of the old one.
4. Restart LogStream and log back in.

5. Wait for all the Worker Nodes to report to the leader, and ensure that they are correctly reporting the last committed configuration version.

Emergency Maintenance

To help diagnose LogStream problems, you can share a diagnostic bundle with Cribl Support. The bundle contains a snapshot of configuration files and logs at the time the bundle was created, and gives troubleshooters insights into how LogStream was configured and operating at that time.For more information on creating a diagnostic bundle, please refer to our documentation: https://docs.cribl.io/docs/diagnosing

Support

Troubleshooting

- I cannot "Create stack" in CloudFormation.
  - Please check that you have the appropriate permissions to "Create Stack". Contact your AWS account admin for permissions, or AWS Support if you continue to have issues.

- Cribl LogStream is throwing licensing errors.
  - Please check that you have provided a valid license.

Clean Up

- Follow the AWS CloudFormation Delete documentation to delete the resources deployed by this document.

- Manually delete any additional resources that you manually created to integrate or assist with the deployment.

Contact Us

Support at the SLAs listed below is included with all AWS Marketplace offerings at no additional cost. To contact Cribl support, please email support@cribl.io, or visit https://cribl.io/support/ and sign up for our community Slack channel.

Please refer to our Support page for up-to-date information on our SLAs for Enterprise and Standard customers. Community users on our Free offering can use our community Slack channel for support.

Appendix

Github repo: https://github.com/criblio/aws-quickstart-cribl-logstream