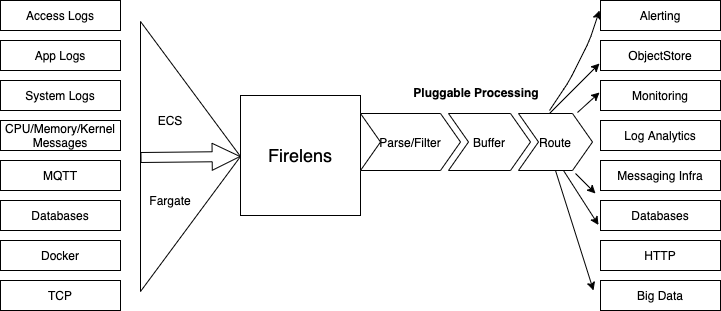
**Firelens demystified**

Firelens is a log routing agent for Amazon ECS containers. Applications on ECS run as docker containers.

One can host their containers on a serverless infrastructure that is managed by ECS by launching your services or tasks using the Fargate launch type. For more control over your infrastructure, one can host containers on a cluster of Amazon Elastic Compute Cloud (Amazon EC2) instances that are managed by using the EC2 launch type. In both cases, AWS manages networking, storage, security, IAM, and other necessary services required to run the containers.

FireLens for Amazon ECS enables the administrator to use task definition attributes to route logs to external log aggregators. [F](https://www.fluentd.org/)irelens unifies the data collection across the ECS cluster. Its pluggable architecture allows adding data sources, parsers, filter/buffering and output plugins.



The biggest advantage with Firelens is that you can connect almost any service endpoint as long as your data sinks can process general-purpose JSON outputs from Firelens router over HTTP, FluentFoward or TCP protocols. The Firelens magic is all about transforming the log output of ECS containers and generating the necessary routing configuration for your destination.

For using Firelens, first, you need to define your log collector or sink, it can be any log aggregation provider like LOGIQ Log Insights.

To use custom log routing with FireLens you must specify the following in your task definition:

* A log router container containing a FireLens configuration. This container must be marked as essential.
* One or more application containers that contain a log configuration specifying the ”**awsfirelens**” log driver.
* A task IAM role ARN containing the permissions needed for the task to route logs to any AWS service that you may require

Below you will find few “**logConfigration**” examples that you can use for your task definition, please note how the “**logDriver**” is set to “**awsfirelens**”, options contain detail about your log sink/destination.

Example 1: "logConfiguration": {

"logDriver":"**awsfirelens**",

"options": {

"Name":"**http**",

"Host": **"**<endpoint>**",**

"URI": "/yourLogProividerURL",

"Port": "443",

"tls": "on",

"tls.verify": "off",

"Format": "json\_lines" }}

Example 2: "logConfiguration": {

"logDriver": "**awsfirelens**",

"options": {

Name": "**forward**"

"Port": "24224",

"Host": "**logiq.example.com**" }}

The ”**awsfirelens**” log driver allows you to specify Fluentd or Fluent Bit output plugin configuration. Your application container logs are routed to a sidecar or independent firelens container inside your cluster that further routes your container log to its destination as defined in your task “**logConfiguration**”. Additionally, you can use the options field of the FireLensConfiguration object in the task definition to serve any advanced use case.

"firelensConfiguration": {

"type": "fluentbit"

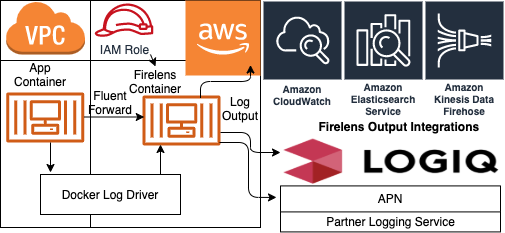
"options": {

"config-file-type": "s3",

"config-file-value": "arn:aws:s3:::mybucket/myFile.conf"

}

}



The diagram above shows how FireLens works. Container logs are sent to the Firelens container using the docker Log Driver.

When the ECS Agent launches a Task that uses FireLens, it constructs a Fluent configuration file:

1. A specification of log source for how to gather the logs from the container
2. A ECS Metadata record transformer
3. Optional User-provided configuration. If you specify your own configuration file, firelens will use the ”**include**” directive to import it in the generated configuration file.
4. Log destinations or sinks derived from the Task Definition

The following snippet shows a configuration for the container definition

{

"containerDefinitions":[

{

"essential":true, "image":"906394416424.dkr.ecr.us-west-2.amazonaws.com/aws-for-fluent-bit:latest",

"name":"log\_router",

"firelensConfiguration":{

"type":"fluentbit",

"options":{

"**enable-ecs-log-metadata**":"true"

}

}

}

]

}

To demonstrate how Firelens works end to end, the below is a task definition example containing a httpd web server and a firelens sidecar container to route logs to multiple destinations including LOGIQ server, CloudWatch and the destinations specified in your fluent conf. In the below example, the Log Router container is configured to send its own logs to the CloudWatch group "firelens-container”. We recommended you send the FireLens logs to CloudWatch to debug Fluent Bit Issues. If you would like to keep this configuration, then a cloudwatch log group named “firelens-container” will be created in us-west-2. Also replace the task execution role if it is named other than the default “executionRoleArn” and populate the account id shown in XXXXXXXXXXXX in the following example:

|  |
| --- |
| { |
|  | "family": "firelens-logiq", |
|  | "taskRoleArn": "arn:aws:iam::**XXXXXXXXXXXX**:role/ecs\_task\_iam\_role", |
|  | "executionRoleArn": "arn:aws:iam::**XXXXXXXXXXXX**:role/ecs\_task\_execution\_role", |
|  | "containerDefinitions": [ |
|  | { |
|  | "essential": true, |
|  | "image": "amazon/aws-for-fluent-bit:latest", |
|  | "name": "log\_router\_logiq", |
|  | "firelensConfiguration": { |
|  | "type": "fluentbit", |
|  | "options": { |
|  | "config-file-type": "s3", |
|  | "config-file-value": "arn:aws:s3:::yourbucket/yourdirectory/extra.conf", |
|  | "enable-ecs-log-metadata": "true" |
|  | } |
|  | }, |
|  | "logConfiguration": { |
|  | "logDriver": "**awslogs**", |
|  | "options": { |
|  | "awslogs-group": "firelens-container" |
|  | "awslogs-region": "us-west-2", |
|  | "awslogs-create-group": "true", |
|  | "awslogs-stream-prefix": "firelens" |
|  | } |
|  | }, |
|  | "memoryReservation": 50 |
|  | }, |
|  | { |
|  | "essential": true, |
|  | "image": "httpd", |
|  | "name": "app", |
|  | "logConfiguration": { |
|  | "logDriver":"**awsfirelens**" |
|  | "options": { |
|  | "Name": "**forward**", |
|  | "Port": "**24224**", |
|  | "Host": "**logiq.example.com**" |
|  | } |
|  | }, |
|  | "memoryReservation": 100 |
|  | } |
|  | ] |
|  | } |