Builder Session

I want to focus on three tools and show how to use them:
container level - **CW logs**
Service/cluster level - **CW metrics** (for Fargate, only service level)
AWS API level or auditing - **Cloudtrail**

CW logs ~~~~
the way your container interacts w CW logs is through the AWS logs driver, which is actually something built into the docker engine itself.

the "logConfiguration" parameter in taskdefinition maps to the "docker run –log-driver" option. The behavior is exactly the same.

the "logConfiguration" parameter in ECS taskdefnition takes a number of different log drivers:
log drivers: awslogs (Fargate can only use this), fluentd, gelf, journald, syslog, json-file
E.g. if you would rather aggregate your logs via Fluentd before shipping them off to somewhere else, you can do that.

i. awslogs-group: We need to specify the log group to pass the log to;
ii. awslogs-region: Tell us what region it exists in;
iii. awslogs-streamp-prefix: you can specify a custom prefix for that given taskdef to make it easier to track and filter in CW logs.

Fargate task definitions **only support** the `awslogs` log driver for the log configuration. This configures your Fargate tasks to send log information to Amazon CloudWatch Logs.

CW Metrics ~~~~
In ECS, by default, ECS exposes metrics at two different levels:

1. at the service level: across all the containers in a given service
2. at the cluster level: aggregate metrics of all the instances in a cluster

CloudTrail ~~~~~~
ECS is integrated with AWS CloudTrail.
CloudTrail is a service that captures API calls made by or on behalf of ECS in your AWS account and then deliver those log files to Amazon S3 bucket that you specify.
CloudTrail is super useful for auditing and understanding what's happening at the API level across your account.
Using the info you've collected in CloudTrail you can determine what request was made to a given service, the source IP of the request, who/when made the request.
It can help you track down changes to your infrastructure and when a specific thing was broken.

CloudTrail is enabled on your AWS account when you create the account. When activity occurs in Amazon ECS, that activity is recorded in a CloudTrail event along with other AWS service events in Event history. You can view, search, and download recent events in your AWS account. For more information, see Viewing Events with CloudTrail Event History.

Let's start building:
https://github.com/lemonapple58/loggingandmonitoring

Step 1: Build the fake logging container
“docker build -t fakelogger”

Step 2: Link AWS ECR cred to docker and push to ECR

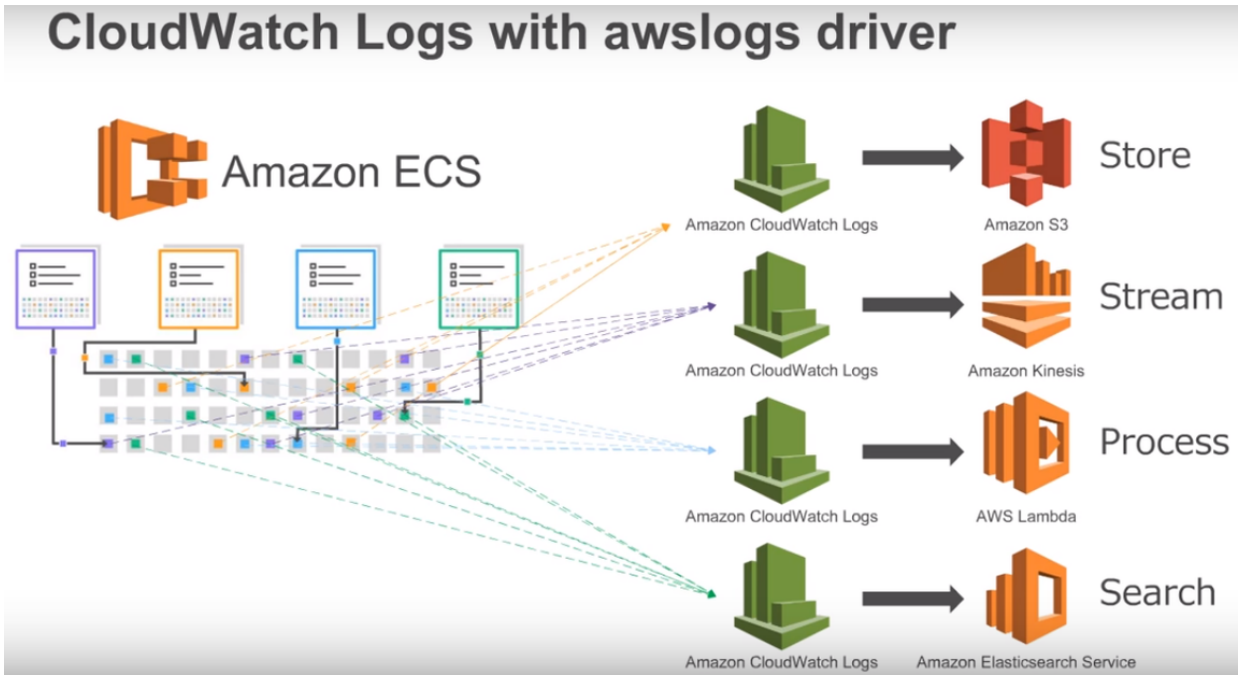
aws ecr get-login --no-include-email”
aws ecr create-repository –repository-name helloworld
docker tag fakelogger {accountid}.dkr.ecr.us-east-1.amazonaws.com/helloworld
docker push {accountid}.dkr.ecr.us-east-1.amazonaws.com/helloworld

Step 3: Create an ECS task definition. Highlight the log configuration section of the task definition. Use the fake logger created above as a part of the task definition.

Step4: Create an ECS service and launch instances of the task.

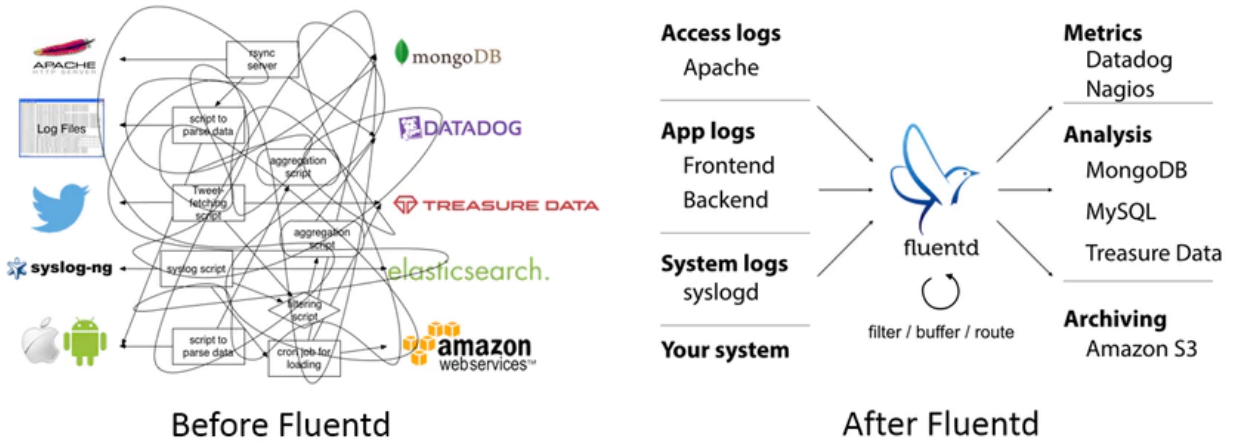
step5: Go to Cloudwatch and see logs and metrics

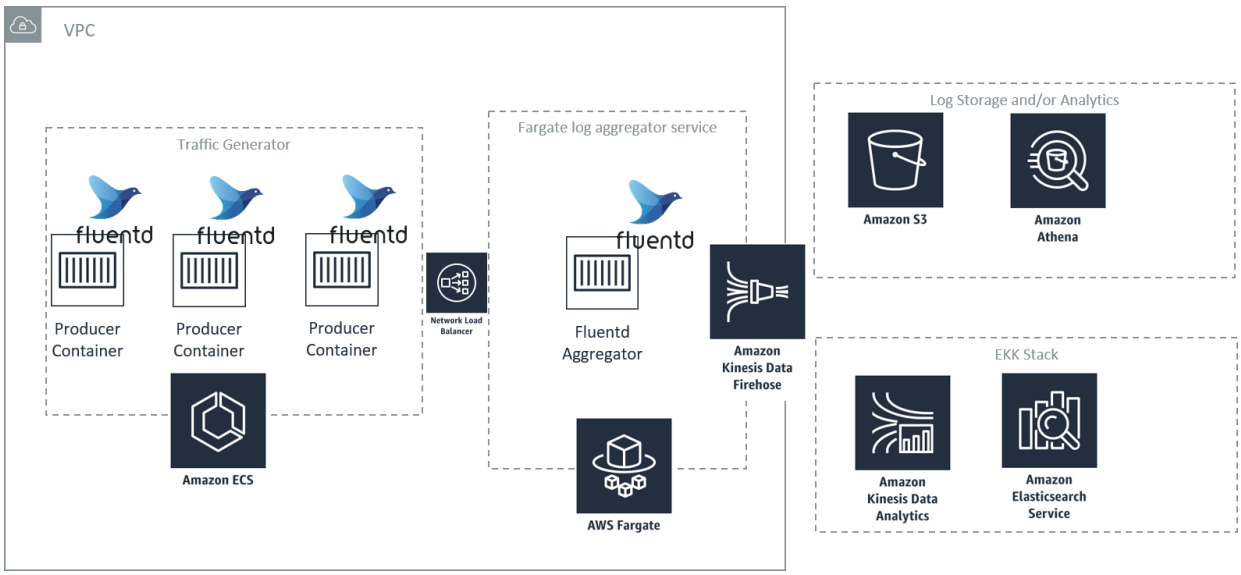
Send CW logs to other places:
You can send CW logs to places where the logs get stored, streamed and processed.



Fluentd
Fluentd is a **data collector**. It lets you unify the data collection and consumption for a better use and understanding of data

You can use Fluentd to build a unified layer that connects your log sources and data backends:





Fluentd. Architecture overview:

- Traffic generator services represent applications that generate log with a variable pattern
- Fluentd log aggregator service collects logs, scales (if needed) and streams logs to Kinesiss Firehose
- Fluentd can be configured to send logs to destinations based on tags/patterns in the stream
- Kinesis Firehose can be set up to stream to destinations such as Splunk, Amazon Elasticsearch and S3

https://www.datadoghq.com/blog/monitor-fluentd-datadog/ TK