Monitoring & Remediation

What are we building in Lab 2?

- Building off of Lab 1 we are going to implement AWS-centric monitoring and build a monitoring and notification system

- This lab is broken up into 2 sections
  1. Enable Logging and Monitoring
  2. Implement Managed Security Solutions and Alerting

You can build the relevant parts of Lab 1 by running CloudFormation script:
https://s3.amazonaws.com/security-compliance-immersion-day/ImmersionDayCF_Module3.json

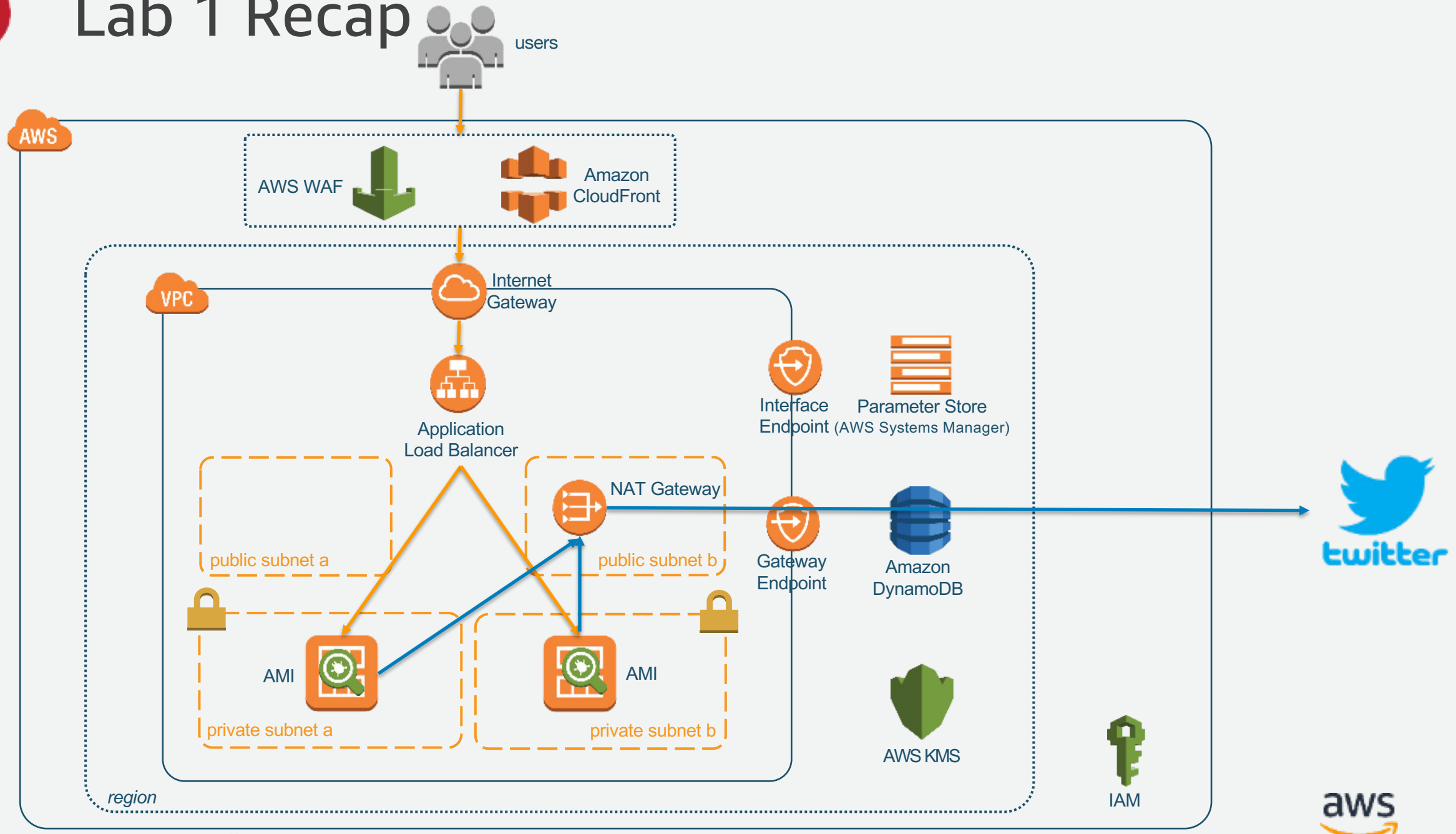
What are we building in Section 1?

- We will implement the monitoring and centralized logging of the application using AWS services.

- In this lab we want to demonstrate:
  1. How to create an ElasticSearch cluster for log evaluation
  2. Setup CloudWatch and CloudTrail to integrate with ElasticSearch
  3. Evaluate logs in ElasticSearch using Kibana

aws

Lab 1 Recap



AWS ElasticSearch

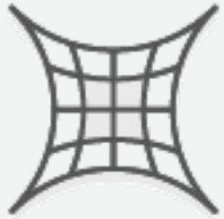
Deploy, secure, operate, and scale Elasticsearch for log analytics, full text search, and application monitoring.

- Analyze un-structured and semi-structured logs
- Capture, pre-process, and load log data using Amazon Kinesis Firehose, Logstash, or Amazon CloudWatch Logs
- Search, explore, and visualize the data using Kibana and the Elasticsearch query DSL

Easy to Use

Supports Open-Source APIs and Tools

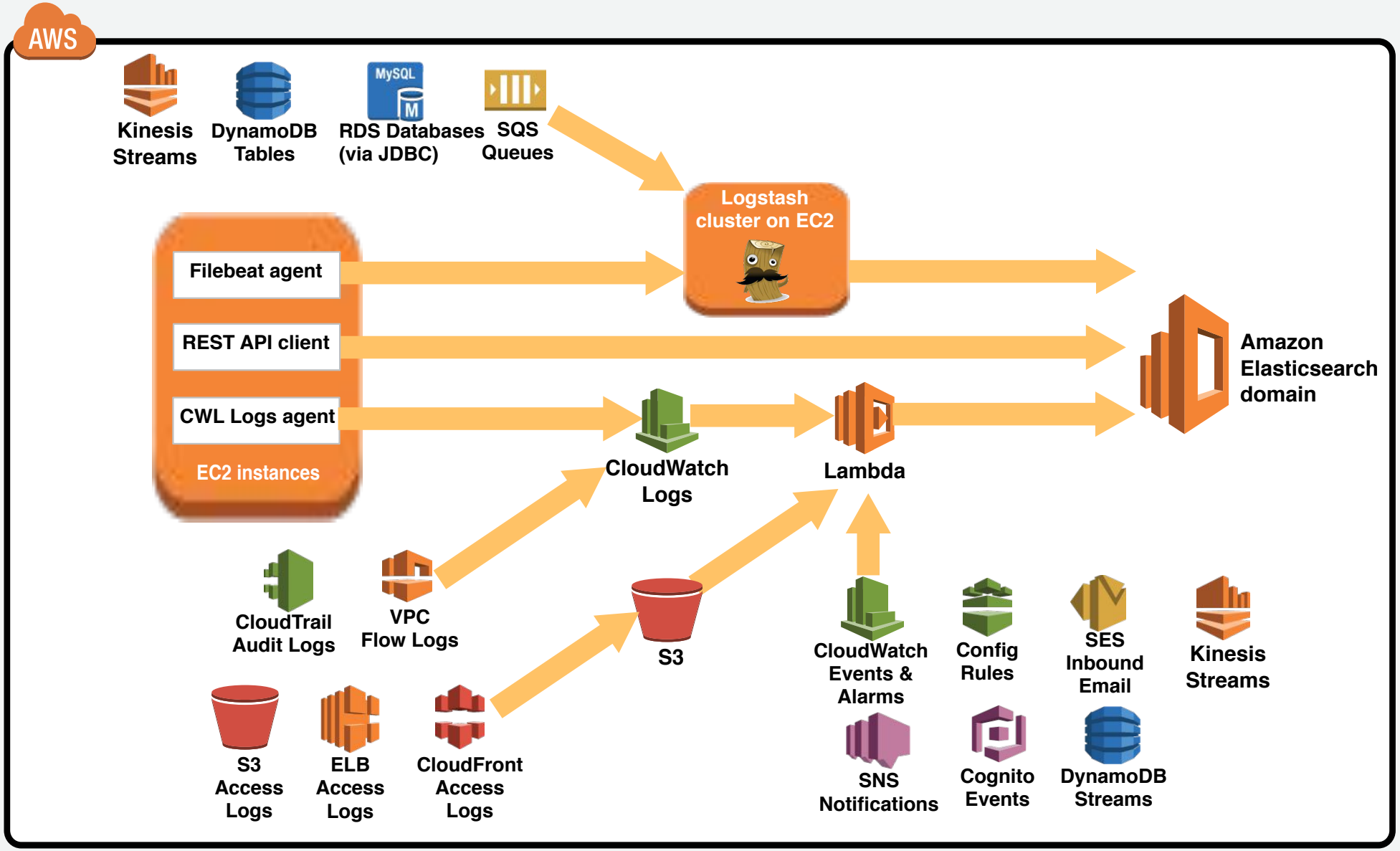
Secure



Easily Scalable

aws

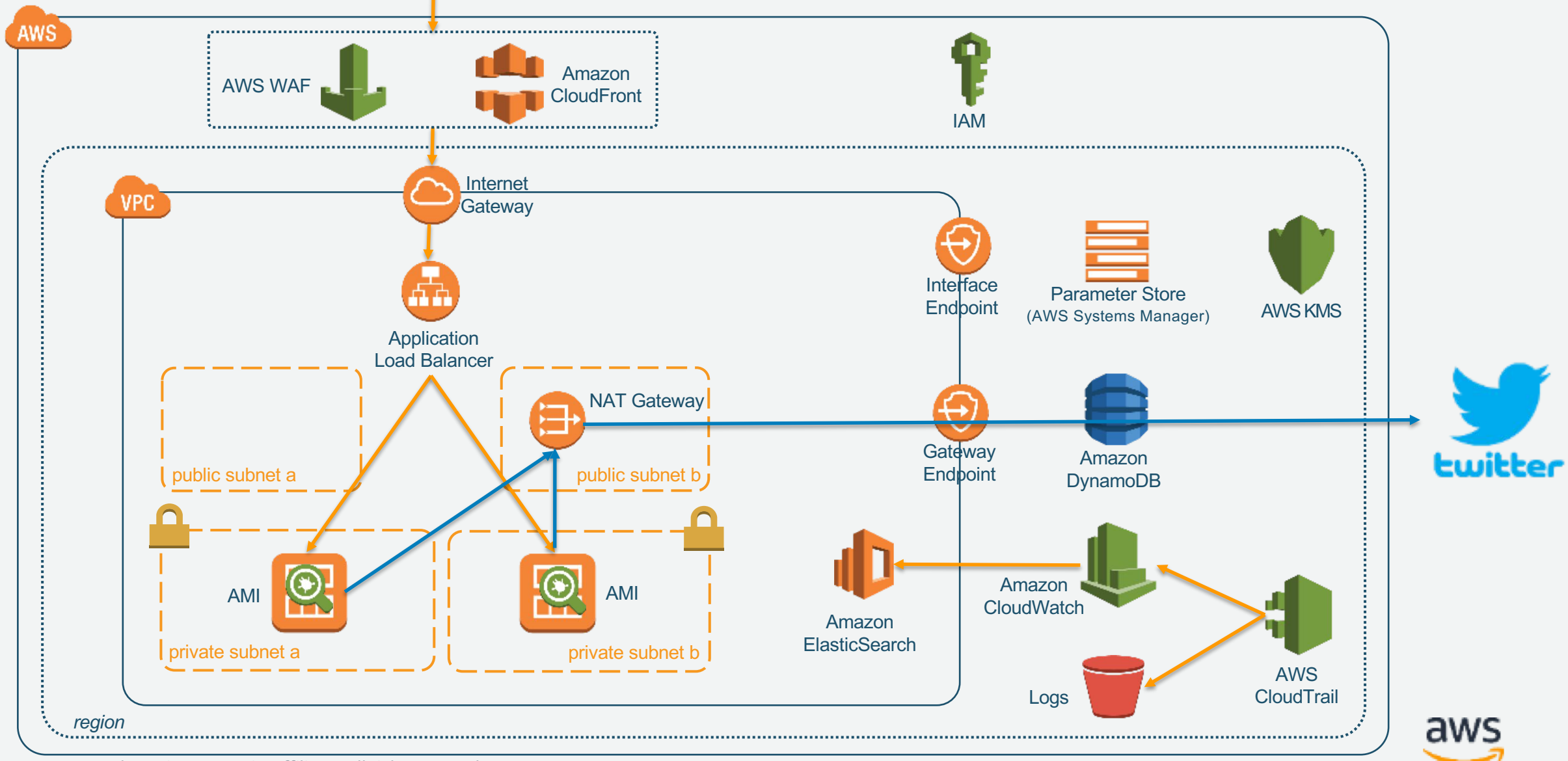
AWS ElasticSearch



Arrow direction indicates general direction of data flow

Section 1

Logging



Step 1 – ElasticSearch setup

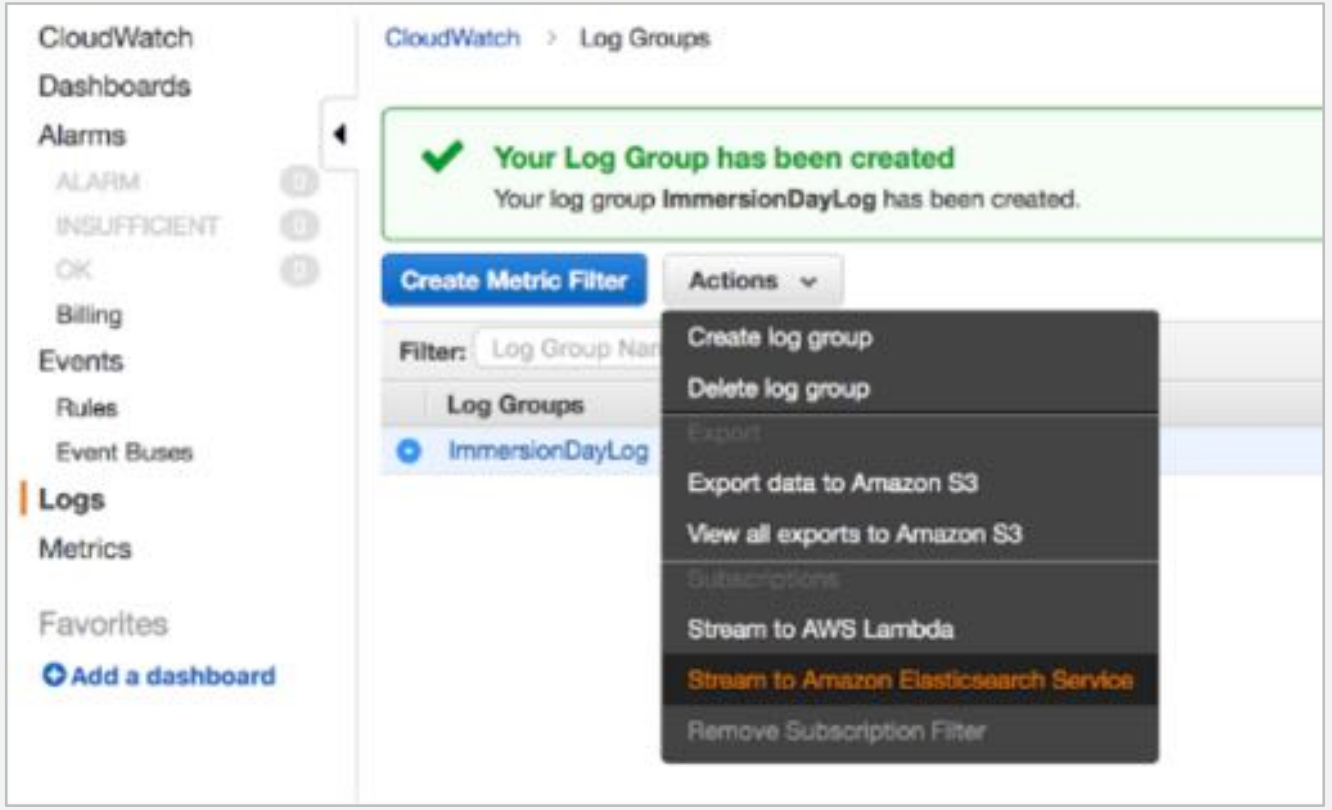
- Create ElasticSearch cluster in a private subnet of our VPC.
- Use the following access policy:

```
{
        "Version": "2012-10-17",
        "Statement": [
        {
                "Effect": "Allow",
                "Principal": { "AWS": "*" },
                "Action": "es:*",
                "Resource": "*"
        }
        ]
}
```

- Note: it takes about 20 mins to create the cluster.

Step 2 – CloudWatch setup

- Create CloudWatch log group (CloudWatch -> Logs -> Create Log Group)
- Send all logs from this group to the previously created ElasticSearch cluster. Log format is CloudTrail and a new IAM role needs to be created for streaming to be successful.



Step 3 – CloudTrail setup

- Create a new CloudTrail trail.
- Configure your destination S3 bucket where the logs will be stored.
- Send all CloudTrail logs to previously created CloudWatch log group.

aws

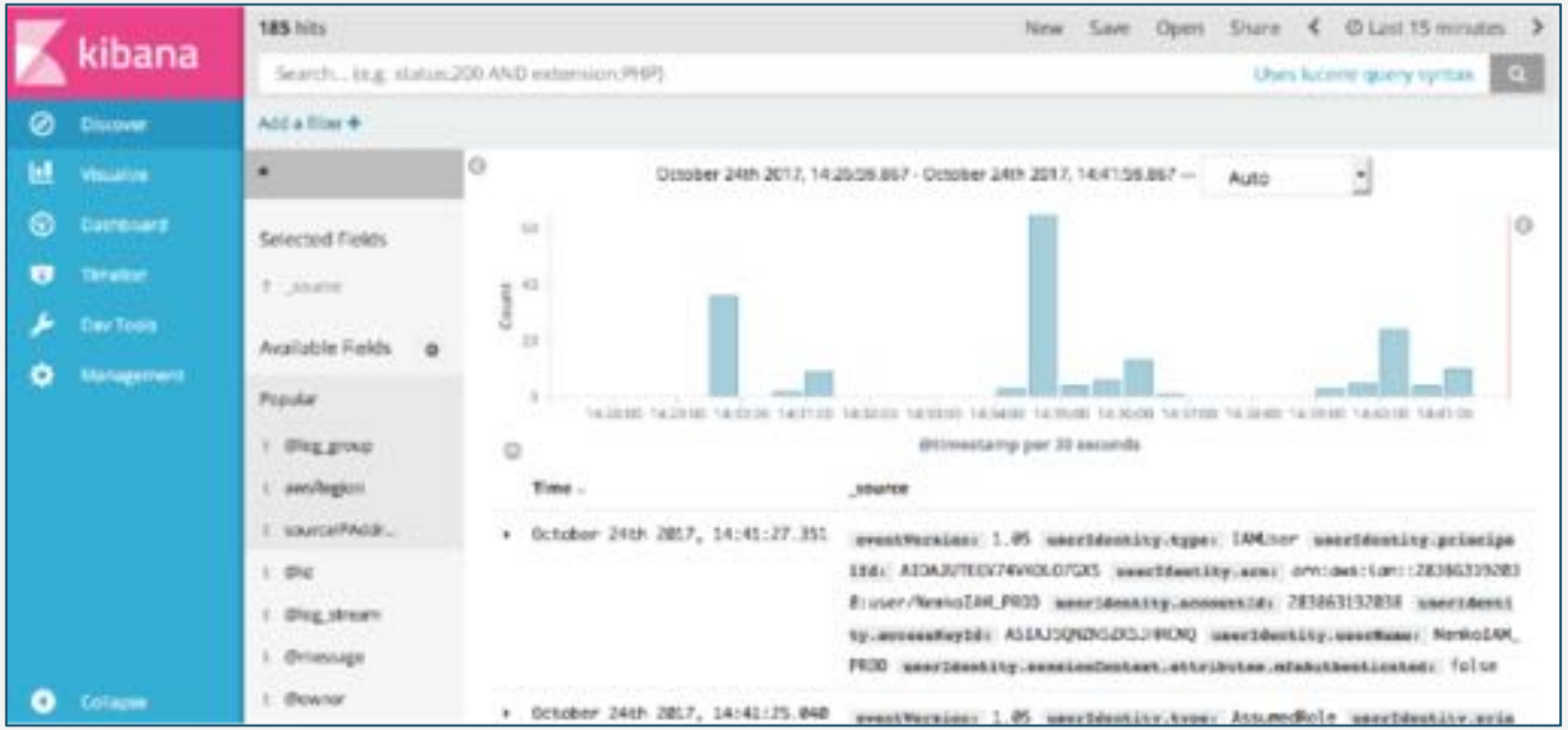
Step 4 – Confirm logs

- Go to the website a few times using the CloudFront URL then wait
- Get the Kibana link from the console
- In Kibana create a query for the last 10 minutes of logs
- Find the logs your traffic generated

aws

What have we achieved?

- We can now see all activity in our account that is recorded by CloudTrail.
- That activity is stored in S3 and streamed via CloudWatch to ElasticSearch.
- Logs in ElasticSearch can be examined by using Kibana (for Index Pattern enter just star *)



What are we building in Section 2?

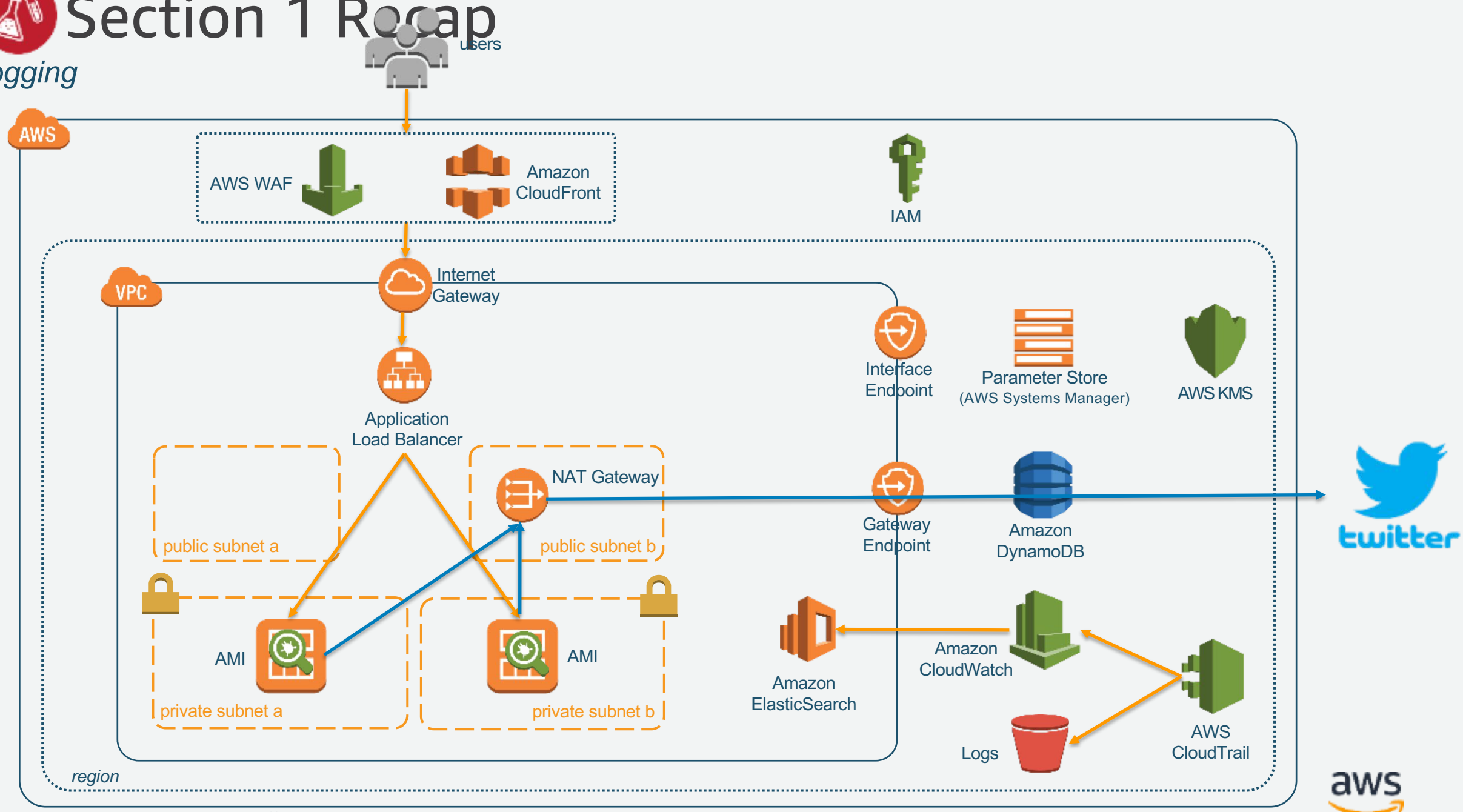
- We will use native AWS Security services to provide active monitoring of our resources.

- In this lab we want to demonstrate:
  1. How to enable and configure GuardDuty
  2. How to enable and configure Inspector
  3. How to enable and configure Macie
  4. How to configure email alerts based on specific findings

aws

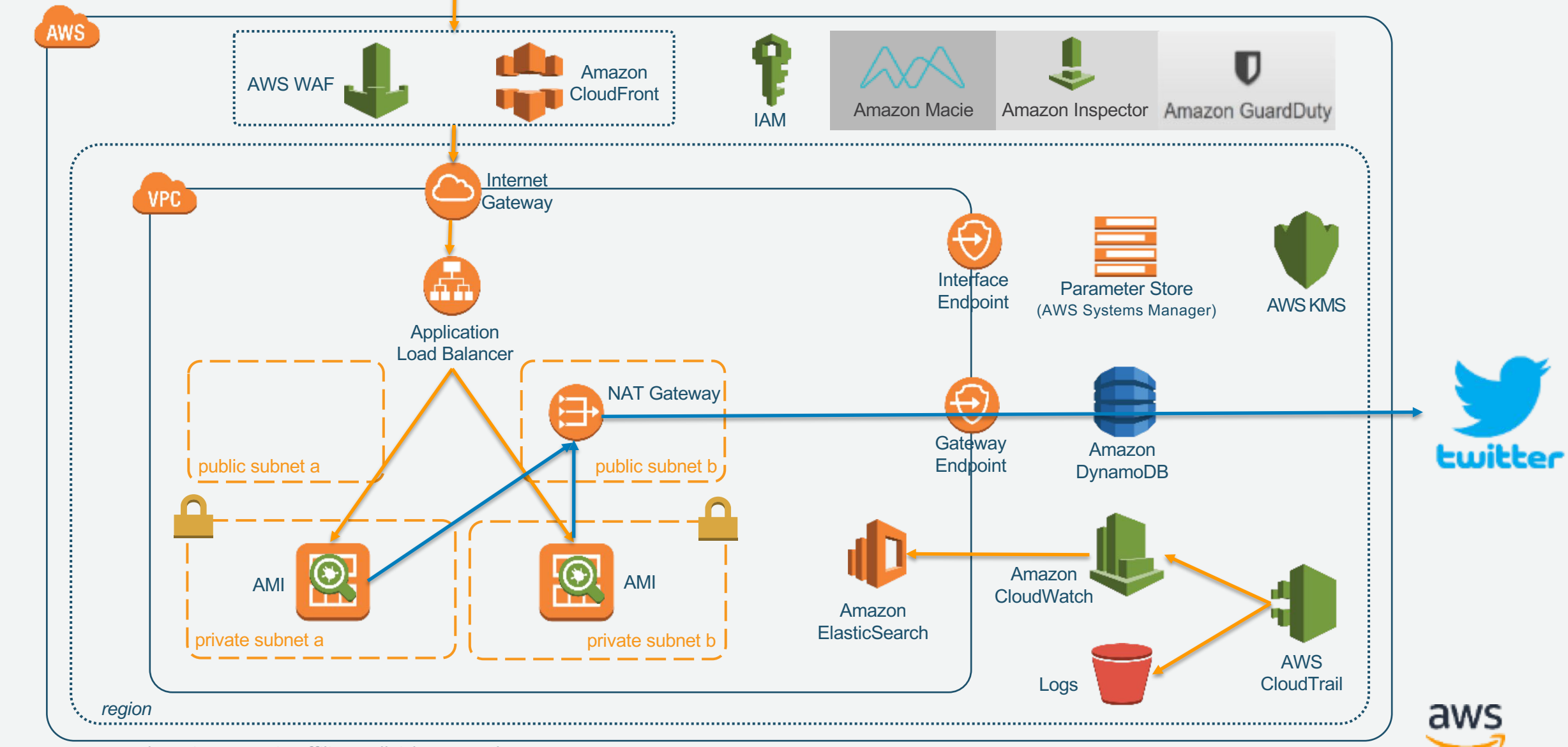
Section 1 Recap

Logging



Section 5

Managed Security Services

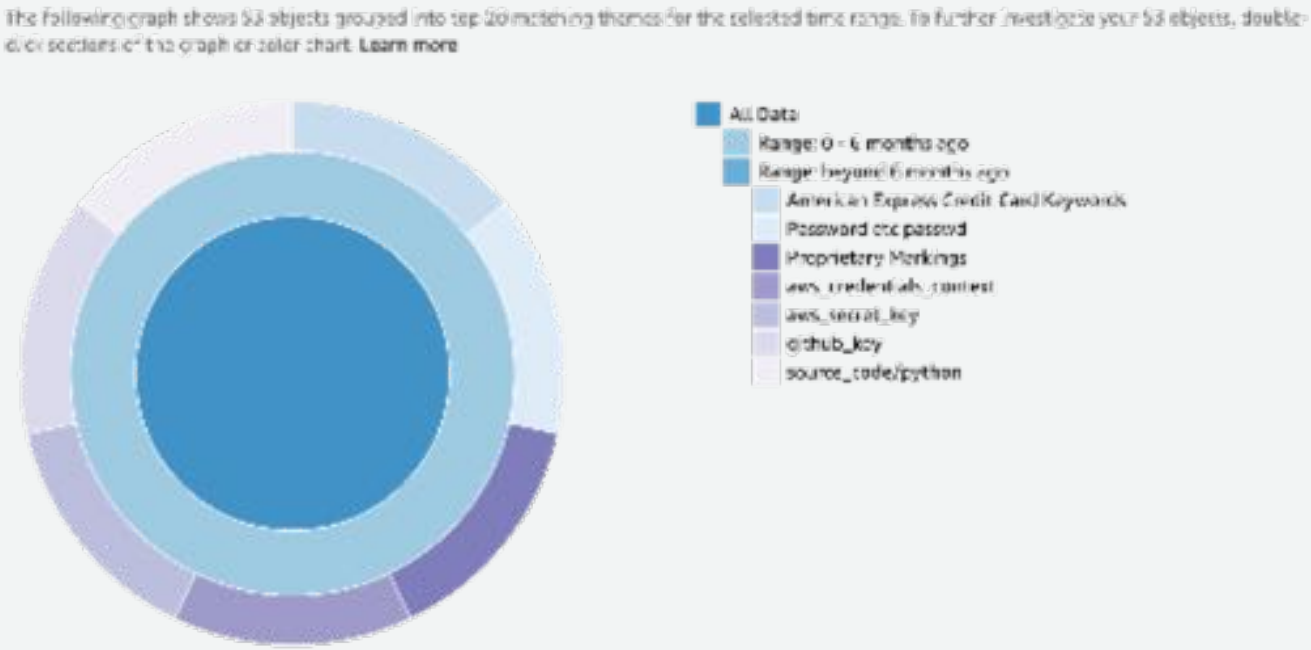


Step 1 – GuardDuty activation

- Enable GuardDuty for your account
- The results won’t be shown immediately, it takes some time for GuardDuty to run.
- GuardDuty finding can be something like this:

Step 2 - Amazon Macie activation

- Enable Macie for your account
- Point Macie ONLY at the S3 buckets created for this lab
- The results won't be shown immediately, it takes some time for Macie to run.
- Macie finding can be something like this:



Step 3 – Create CloudWatch email notifications

- Go to the **Simple Notification Service**
- Create a **Topic**
- Create an **email subscription** using your email address
- Open the **CloudWatch console**
- Create a **Rule for GuardDuty** that uses your newly created **SNS Topic** as a **Target**
- Create a 2nd Rule for **Macie** using the same SNS Topic

Step 4 – Amazon Inspector activation

- Amazon Inspector is using an agent that is already packaged with our AMI.
- Create Assessment Targets – those will be our EC2 instances in private subnets.
- The assessments targets are identified by tags. For Key enter “instance” and for Value enter “immersionday” as these are the tags we used for our EC2 instances.
- Create Assessment Template where you can select those two EC2 instances and select rules to run for a certain duration period.

Amazon Inspector findings

- After running “Assessment Runs” for our template, we should get first results within an hour.

- The results might have findings such as:

Rules package	Common Vulnerabilities and Exposures-1.1
AWS agent ID	i-0b3c824618cf3271b
Finding	Instance i-0b3c824618cf3271b is vulnerable to CVE-2018-1000021
Severity	High
Description	GIT version 2.15.1 and earlier contains a Input Validation Error vulnerability in Client that can result in problems including messing up terminal configuration to RCE. This attack appear to be exploitable via The user must interact with a malicious git server, (or have their traffic modified in a MITM attack).
Recommendation	Use your Operating System's update feature to update package git-1:2.7.4-0ubuntu1.3. For more information see https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2018-1000021