Logging Services

CloudTrail - logs all **API calls** (SDK, CLI) between **AWS services** (who can we blame)

Who created this bucket?
Who spun up that expensive EC2 instance?
Who launched this SageMaker Notebook?

- Detect developer misconfiguration
- Detect malicious actors
- Automate responses

CloudWatch is a collection of multiple services

- CloudWatch **Logs** A centralized place to store your cloud services log data or application logs.
- CloudWatch **Metrics** Represents a time-ordered set of data points. A variable to monitor
- CloudWatch **Events (EventBridge)** trigger an event based on a condition eg. ever hour take snapshot of server
- CloudWatch **Alarms** triggers notifications based on metrics
- CloudWatch **Dashboard** create visualizations based on metrics

AWS X-Ray is a **distributed tracing system**. You can use it to pinpoint issues with your microservices.
See how data moves from one app to another, how long it took to move, and if it failed to move forward.

AWS CloudTrail

AWS CloudTrail is a service that enables governance, compliance, operational auditing, and risk auditing of your AWS account.

AWS CloudTrail is used to monitor API calls and Actions made on an AWS account.

Easily identify which users and accounts made the call to AWS eg.

- **Where** — Source IP Address
- **When** — EventTime
- **Who** — User, UserAgent
- **What** — Region, Resource, Action


```
{"Records": [{
    "eventVersion": "1.0",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::123456789012:user/Worf",
        "accountId": "123456789012",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Worf"
    },
    "eventTime": "2014-03-24T21:11:59Z",
    "eventSource": "iam.amazonaws.com",
    "eventName": "CreateUser",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "127.0.0.1",
    "userAgent": "aws-cli/1.3.2 Python/2.7.5 Windows/10",
    "requestParameters": {"userName": "LaForge"},
    "responseElements": {"user": {
        "createDate": "Mar 24, 2014 9:11:59 PM",
        "userName": "LaForge",
        "arn": "arn:aws:iam::123456789012:user/LaForge",
        "path": "/",
        "userId": "EXAMPLEUSERID"
    }}
}]}
```

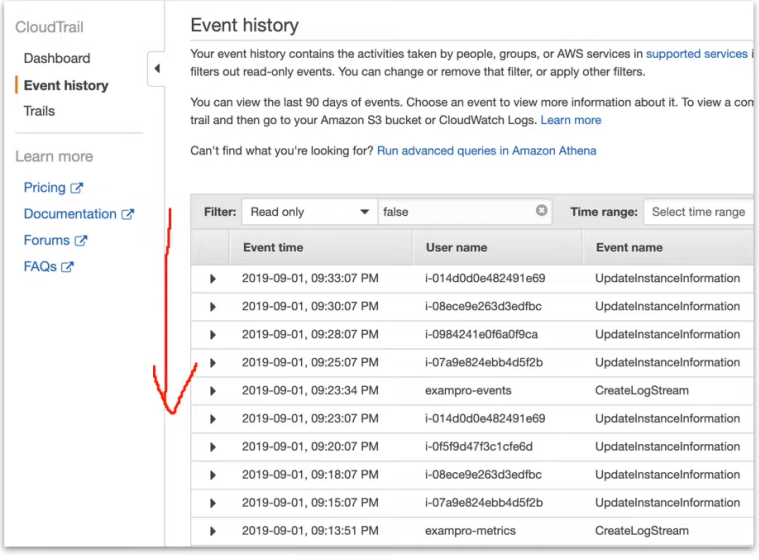
AWS CloudTrail

Cheat sheets, Practice Exams and Flash cards 👉 www.exampro.co/clf-c01

CloudTrail is already logging by default and will collect logs for **last 90 days** via **Event History**

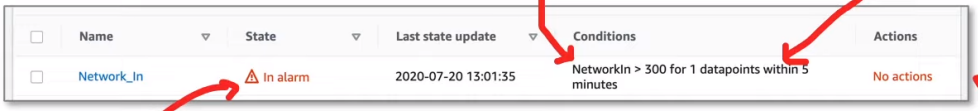
If you need more than 90 days you need to create a **Trail**

Trails are output to S3 and do not have GUI like Event History. To analyze a Trail you'd have to use **Amazon Athena.**



CloudWatch Alarms

A CloudWatch Alarm monitors a **CloudWatch Metric** based on **a defined threshold.**

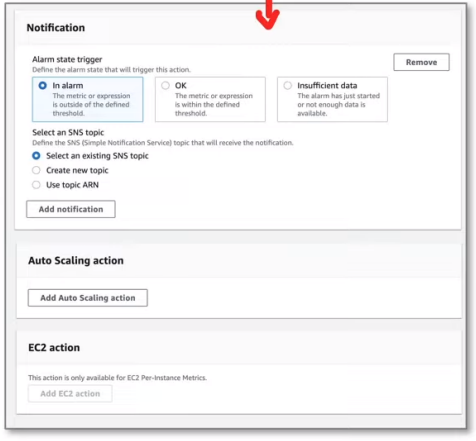


Name	State	Last state update	Conditions	Actions
Network_In	⚠ In alarm	2020-07-20 13:01:35	NetworkIn > 300 for 1 datapoints within 5 minutes	No actions

When alarm breaches (goes outside the defined threshold) than it changes **state**.

When it changes state we can define what **action it should trigger.**

- Notification
- Auto Scaling Group
- EC2 Action



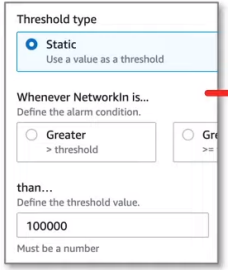
Metric Alarm States

- **OK** The metric or expression is **within** the defined threshold
- **ALARM** The metric or expression is **outside** of the defined threshold
- **INSUFFICIENT_DATA**
  - The alarm has **just started**
  - the metric is **not available**
  - **Not enough data** is available

CloudWatch Alarms — Anatomy of an Alarm

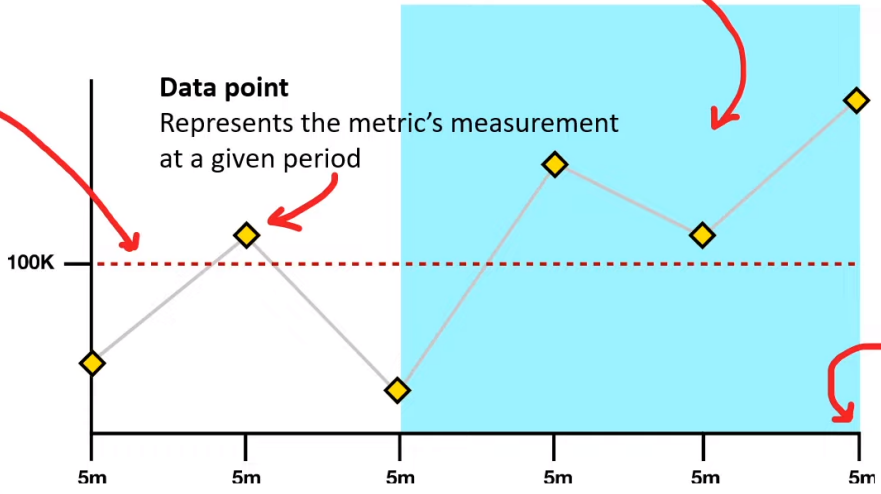
Cheat sheets, Practice Exams and Flash cards 👉 www.exampro.co/clf-c01

Threshold Condition
Defines when a datapoint is breached

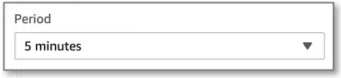


Evaluation Periods
number of previous periods

Data point
Represents the metric's measurement at a given period



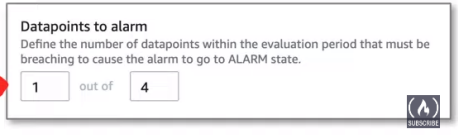
Period
How often it checks to evaluate the Alarm



Metric
The actual data we are measuring

NetworkIn
The volume of incoming network traffic. measured in Bytes. When using 5min monitoring divide by 300 to get Bytes/second

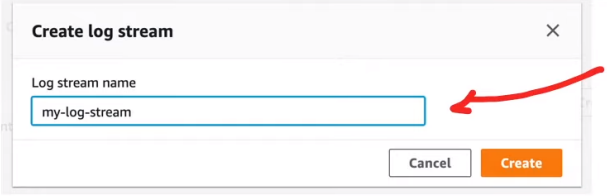
Datapoints to alarm
1 data point is breached in an evaluation period going back 4 periods.
This is what triggers the alarm



CloudWatch Logs – Log Streams

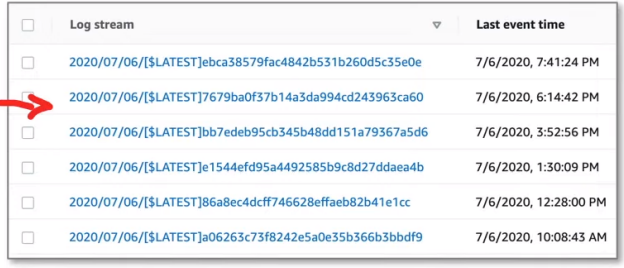
Log Streams

A log stream represents a **sequence of events** from a **application or instance being monitored**.

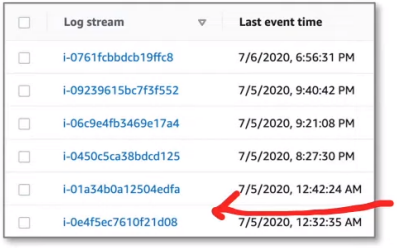


You can create Log Streams manually but generally this is automatically done by the service you are using

Here is a Log Group for a **Lambda function**
You can see here the Log Streams are named after the **running instance.** Lambdas frequency run on new instances so the stream streams contain timestamps

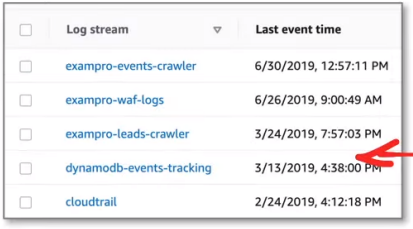


Log stream	Last event time
2020/07/06/[$LATEST]ebca38579fac4842b531b260d5c35e0e	7/6/2020, 7:41:24 PM
2020/07/06/[$LATEST]7679ba0f37b14a3da994cd243963ca60	7/6/2020, 6:14:42 PM
2020/07/06/[$LATEST]bb7edeb95cb345b48dd151a79367a5d6	7/6/2020, 3:52:56 PM
2020/07/06/[$LATEST]e1544efd95a4492585b9c8d27ddaea4b	7/6/2020, 1:30:09 PM
2020/07/06/[$LATEST]86a8ec4dcff746628effaeb82b41e1cc	7/6/2020, 12:28:00 PM
2020/07/06/[$LATEST]a06263c73f8242e5a0e35b366b3bbdf9	7/6/2020, 10:08:43 AM



Log stream	Last event time
i-0761fcbbdcb19ffc8	7/6/2020, 6:56:31 PM
i-09239615bc7f3f552	7/5/2020, 9:40:42 PM
i-06c9e4fb3469e17a4	7/5/2020, 9:21:08 PM
i-0450c5ca38bdcd125	7/5/2020, 8:27:30 PM
i-01a34b0a12504edfa	7/5/2020, 12:42:24 AM
i-0e4f5ec7610f21d08	7/5/2020, 12:32:35 AM

Here is a Log Group for an **application logs running on EC2** You can see here the Log Streams are named after the **running instance's Instance ID**



Log stream	Last event time
exampro-events-crawler	6/30/2019, 12:57:11 PM
exampro-waf-logs	6/26/2019, 9:00:49 AM
exampro-leads-crawler	3/24/2019, 7:57:03 PM
dynamodb-events-tracking	3/13/2019, 4:38:00 PM
cloudtrail	2/24/2019, 4:12:18 PM

Here is a Log Group for **AWS Glue**.
You can see here the Log Streams are named after the **Glue Jobs.**

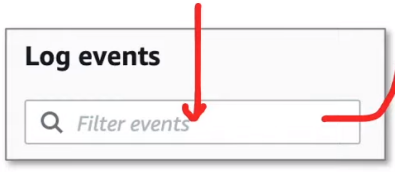
CloudWatch Logs – Log Events

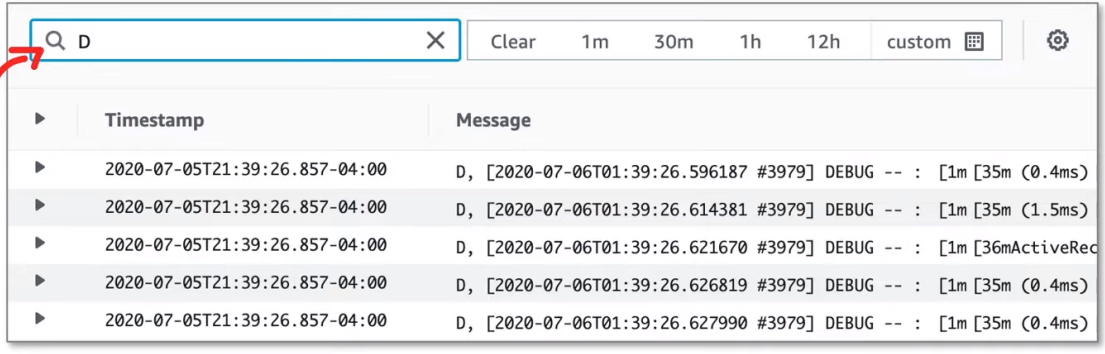
Log Events

Represents a single event in a log file. Log events can be seen within a Log Stream.

▶	2020-07-06T20:12:18.079-04:00	START RequestId: e4b5bd10-5d88-4d7b-870c-daf793159b88 Version: $LATEST
▶	2020-07-06T20:12:18.082-04:00	{"records_size":1}
▶	2020-07-06T20:12:18.093-04:00	{"failed_put_count":0}
▶	2020-07-06T20:12:18.127-04:00	END RequestId: e4b5bd10-5d88-4d7b-870c-daf793159b88
▶	2020-07-06T20:12:18.127-04:00	REPORT RequestId: e4b5bd10-5d88-4d7b-870c-daf793159b88 Duration: 45.32 ms Billed Duration: 100 ms Memory Size: 128

You can use filter events to filter Out logs based on simple or Pattern matching syntax:





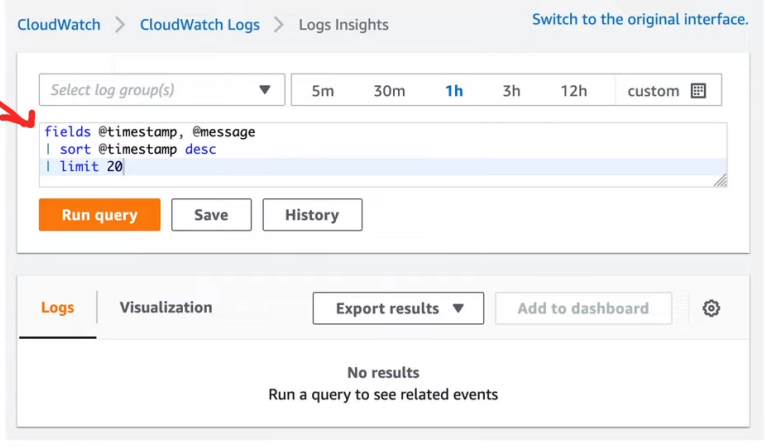
CloudWatch Logs — Log Insights

CloudWatch Logs Insights enables you to **interactively search and analyze your CloudWatch log data** and has the following advantages:

- more robust filtering then using the simple Filter events in a Log Stream
- Less burdensome then having to export logs to S3 and analyze them via Athena.

CloudWatch Logs Insights supports all types of logs.

CloudWatch Logs Insights is commonly used via the console to do ad-hoc queries against logs groups.



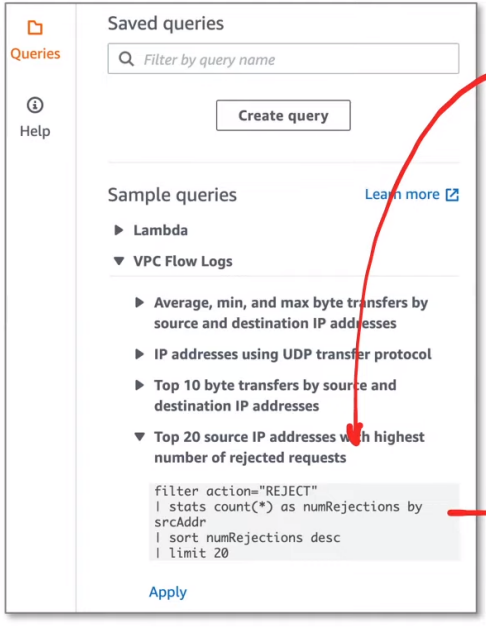
CloudWatch Insights has its own language called:

- CloudWatch Logs Insights **Query Syntax**

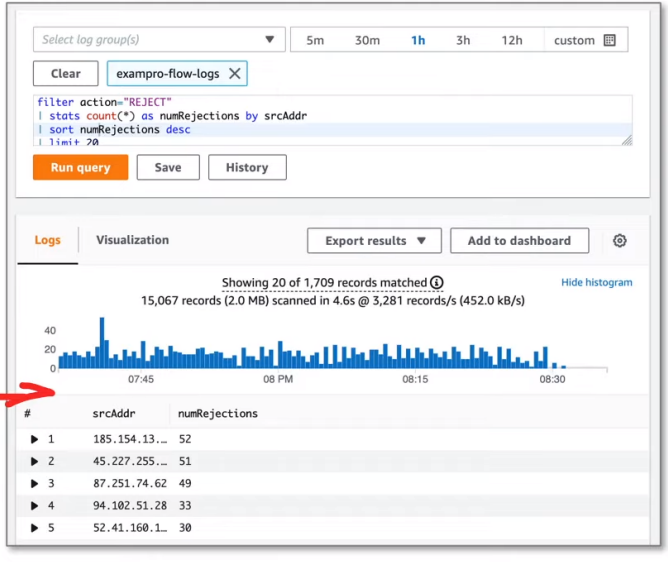
```
filter action="REJECT"
| stats count(*) as numRejections by srcAddr
| sort numRejections desc
| limit 20
```

- A single request can query up to **20 log groups**.
- Queries **time out after 15 minutes**, if they have not completed.
- Query results are **available for 7 days.**

CloudWatch Logs – Log Insights



AWS provides sample queries that can get you started for common tasks,
And to ease learning the Query Syntax. A good example Is filtering VPC Flow Logs.

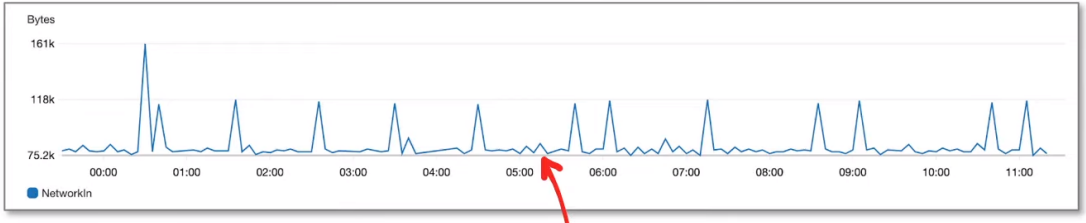


You can create and save your own queries to make future repetitive tasks easier.

CloudWatch Metrics

A CloudWatch Metric represents a **time-ordered set of data points**
Its a **variable** that is **monitored over time**.

CloudWatch comes with many **predefined** metrics that are generally name spaced by AWS Service.



EC2 Per-Instance Metrics

- CPUUtilization
- DiskReadOps
- DiskWriteOps
- DiskReadBytes
- DiskWriteBytes
- **NetworkIn**
- NetworkOut
- NetworkPacketsIn
- NetworkPacketsOut