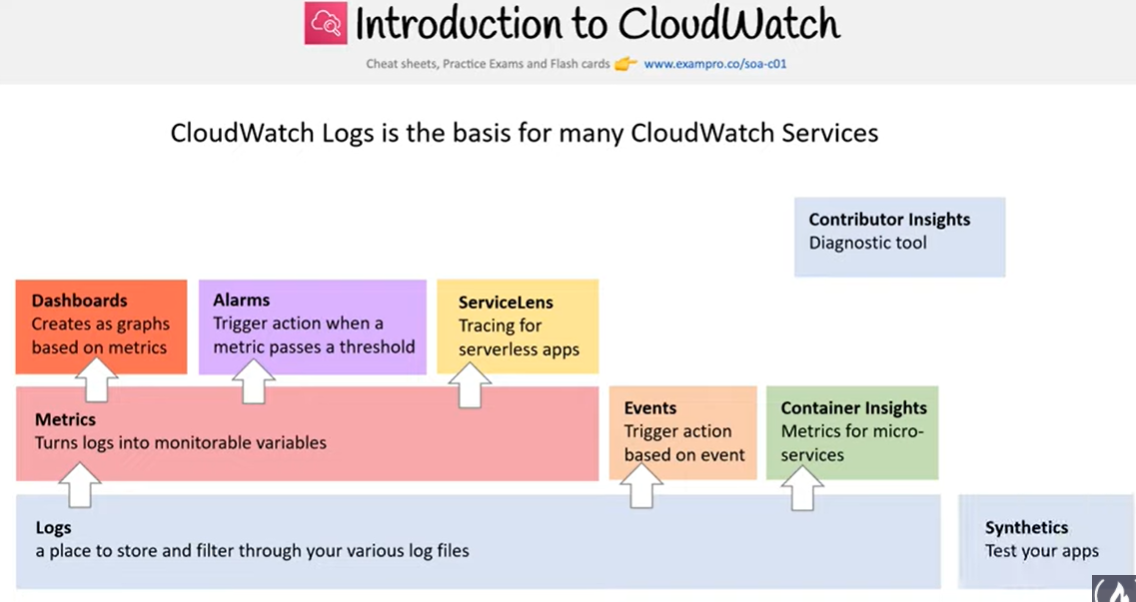
AWS CloudWatch Logs

# 







# AWS CloudWatch Logs

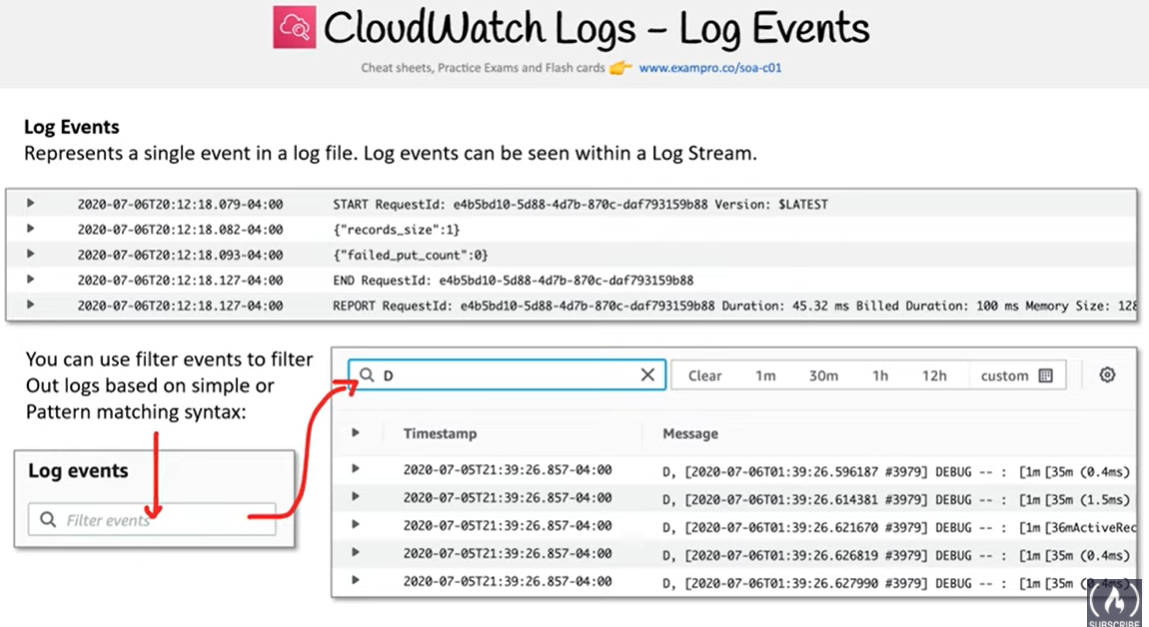
* CloudWatch Logs can be used to monitor, store, and access log files from  EC2 instances, CloudTrail, Route 53, and other sources
* CloudWatch Logs uses the log data for monitoring in an not; so, no code changes are required
* CloudWatch Logs require CloudWatch logs agent to be installed on the EC2 instances and on-premises servers.
* CloudWatch Logs agent makes it easy to quickly send both rotated and non-rotated log data off of a host and into the log service.
* An VPC endpoint can be configured to keep traffic between VPC and CloudWatch Logs from leaving the Amazon network. It doesn’t require an IGW, NAT, VPN connection, or Direct Connect connection
* CloudWatch Logs allows exporting log data from the log groups to an S3 bucket, which can then be used for custom processing and analysis, or to load onto other systems.
* **Log data is encrypted while in transit and while it is at rest**
* **Log data can be encrypted using an AWS KMS or customer master key (CMK).**

#### **Required Mainly for SysOps Associate & DevOps Professional Exam**

## CloudWatch Logs Concepts

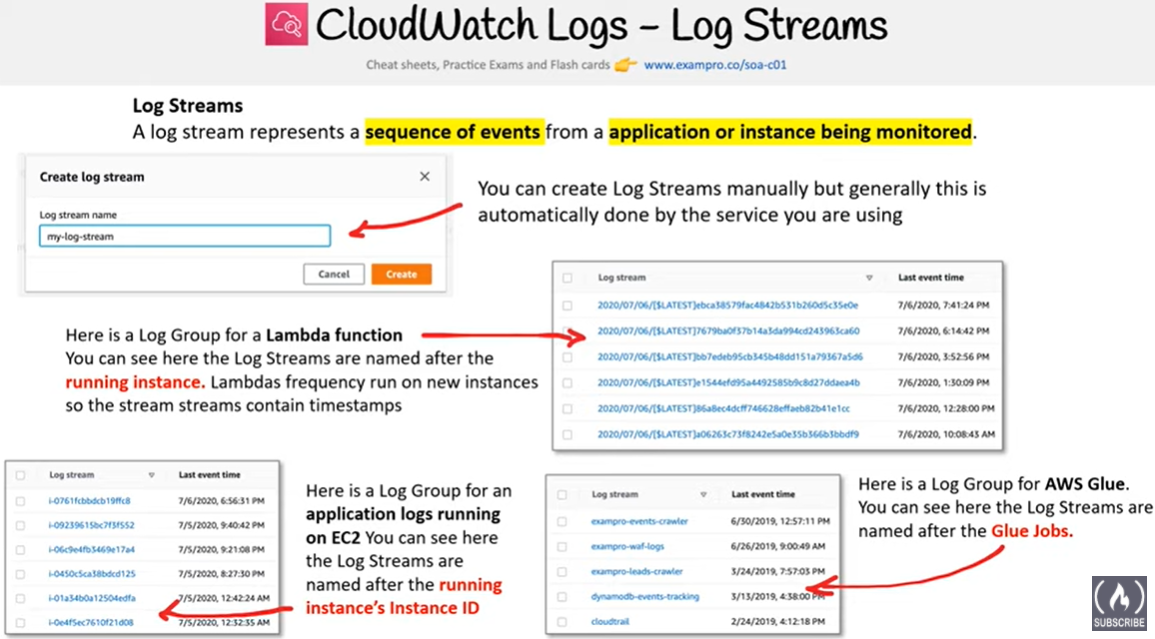
### Log Events

* A log event is a record of some activity recorded by the application or resource being monitored.
* Log event record contains two properties: the timestamp of when the event occurred, and the raw event message



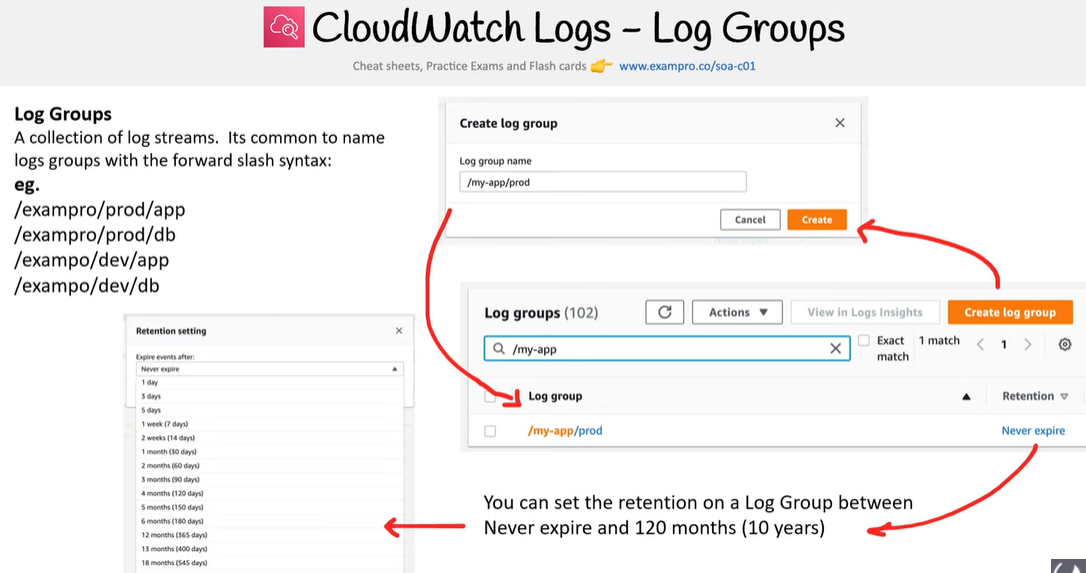
### Log Streams

* A log stream is a sequence of log events that share the same source for e.g. logs events from an Apache access log on a specific host.



### Log Groups

* Log groups define groups of log streams that share the same retention, monitoring, and access control settings for e.g. Apache access logs from each host grouped through log streams into a single log group
* Each log stream has to belong to one log group
* There is no limit on the number of log streams that can belong to one log group.



### Metric Filters

* Metric filters can be used to extract metric observations from ingested events and transform them to data points in a CloudWatch metric.
* Metric filters are assigned to log groups, and all of the filters assigned to a log group are applied to their log streams.

### Retention Settings

* **Retention settings can be used to specify how long log events are kept in CloudWatch Logs**.
* **Expired log events get deleted automatically.**
* Retention settings are assigned to log groups, and the retention assigned to a log group is applied to their log streams.

## CloudWatch Logs Use cases

### Monitor Logs from EC2 Instances in Real-time

* can help monitor applications and systems using log data
* can help track number of errors for e.g. 404, 500, for even specific literal terms “NullReferenceException”, occurring in the applications, which can then be matched to a threshold to send notification

### Monitor AWS CloudTrail Logged Events

* can be used to monitor particular API activity as captured by CloudTrail by creating alarms in CloudWatch and receive notifications

### Archive Log Data

* can help store the log data in highly durable storage, an alternative to S3
* log retention setting can be modified, so that any log events older than this setting are automatically deleted.

### Log Route 53 DNS Queries

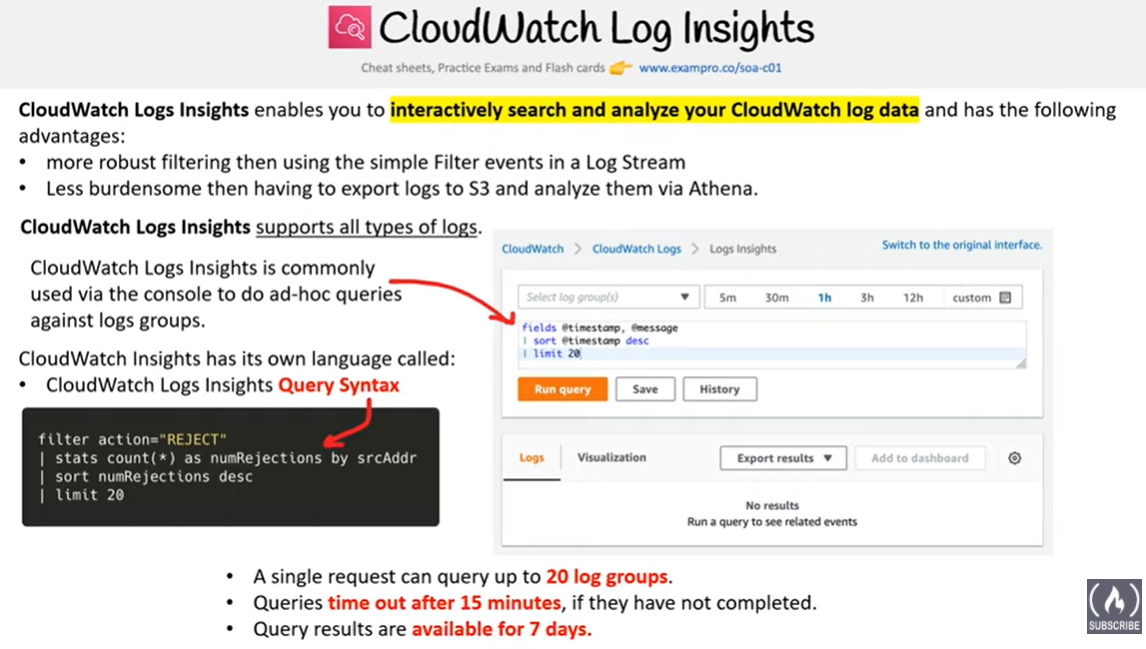
* can help log information about the DNS queries that Route 53 receives.

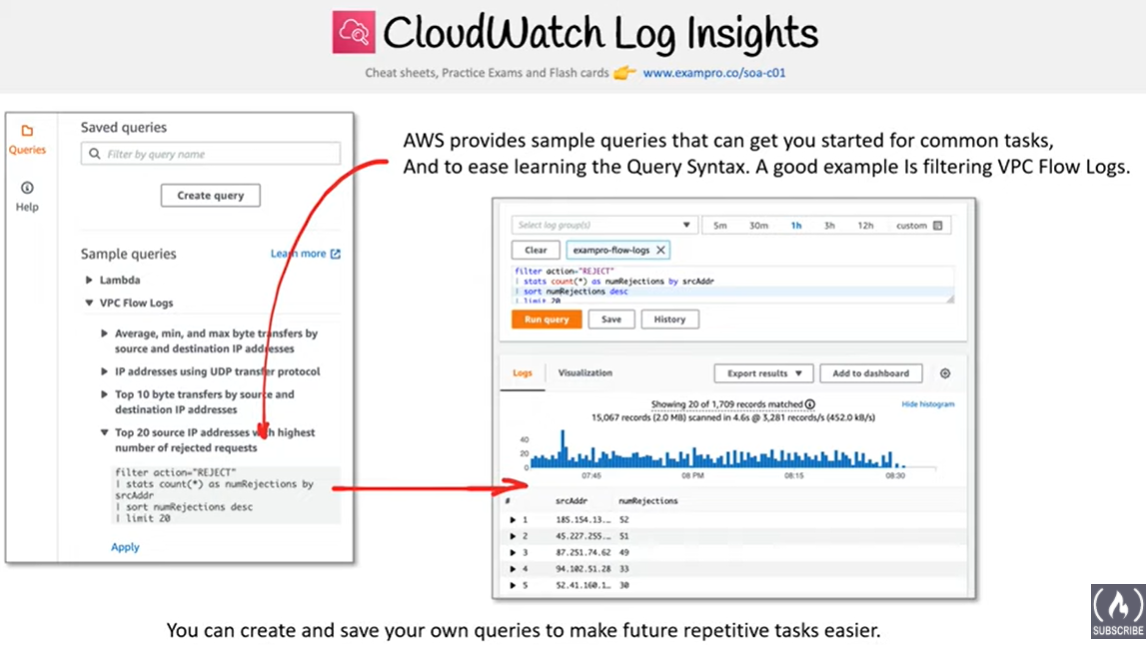
**Real-time Processing of Log Data with Subscriptions**

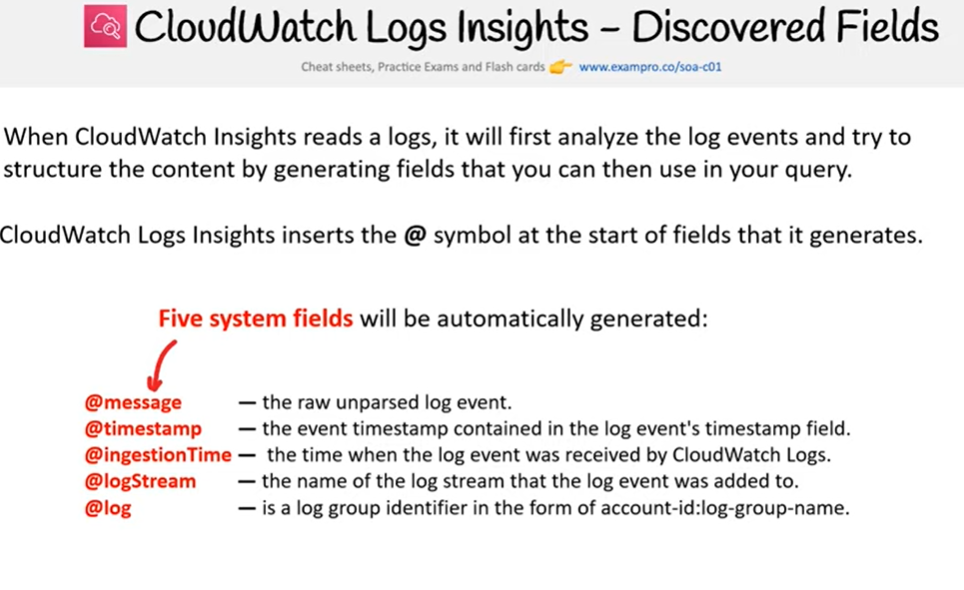
* Subscriptions can help get access to real-time feed of logs events from CloudWatch logs and have it delivered to other services such as Kinesis stream, Kinesis Data Firehose stream, or AWS Lambda for custom processing, analysis, or loading to other systems
* A subscription filter defines the filter pattern to use for filtering which log events get delivered to the AWS resource, as well as information about where to send matching log events to.
* CloudWatch Logs log group can also be configured to stream data Elasticsearch Service cluster in near real-time

### Searching and Filtering

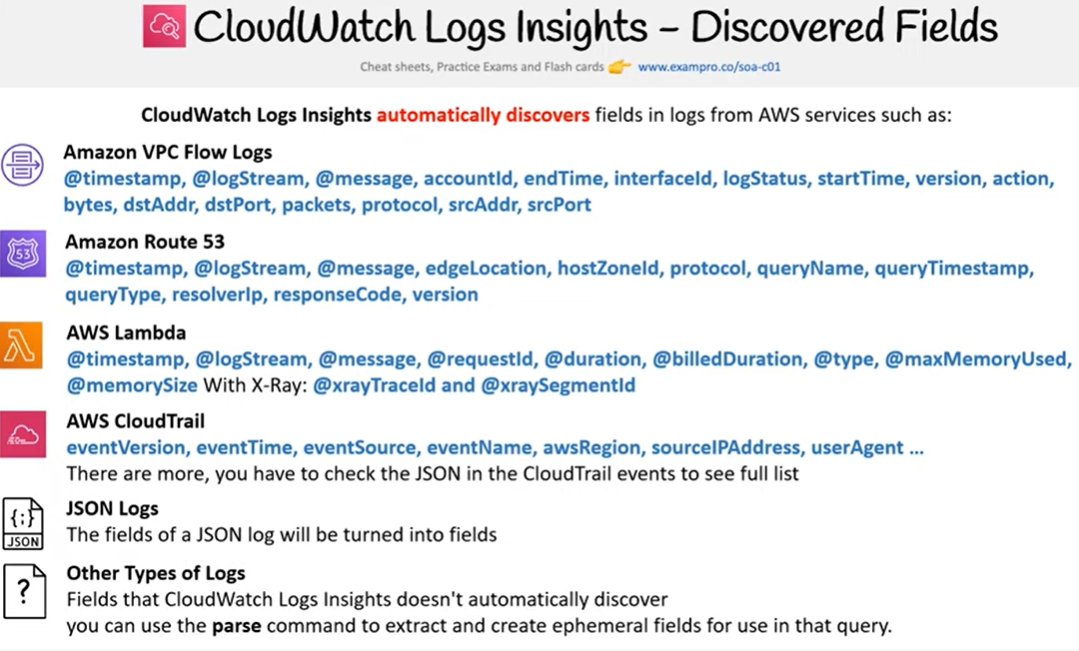
* **CloudWatch Logs allows searching and filtering the log data by creating one or more metric filters**.
* **Metric filters define the terms and patterns to look for in log data as it is sent to CloudWatch Logs.**
* **CloudWatch Logs uses these metric filters to turn log data into numerical CloudWatch metrics that can be put as graph or set an alarm on.**

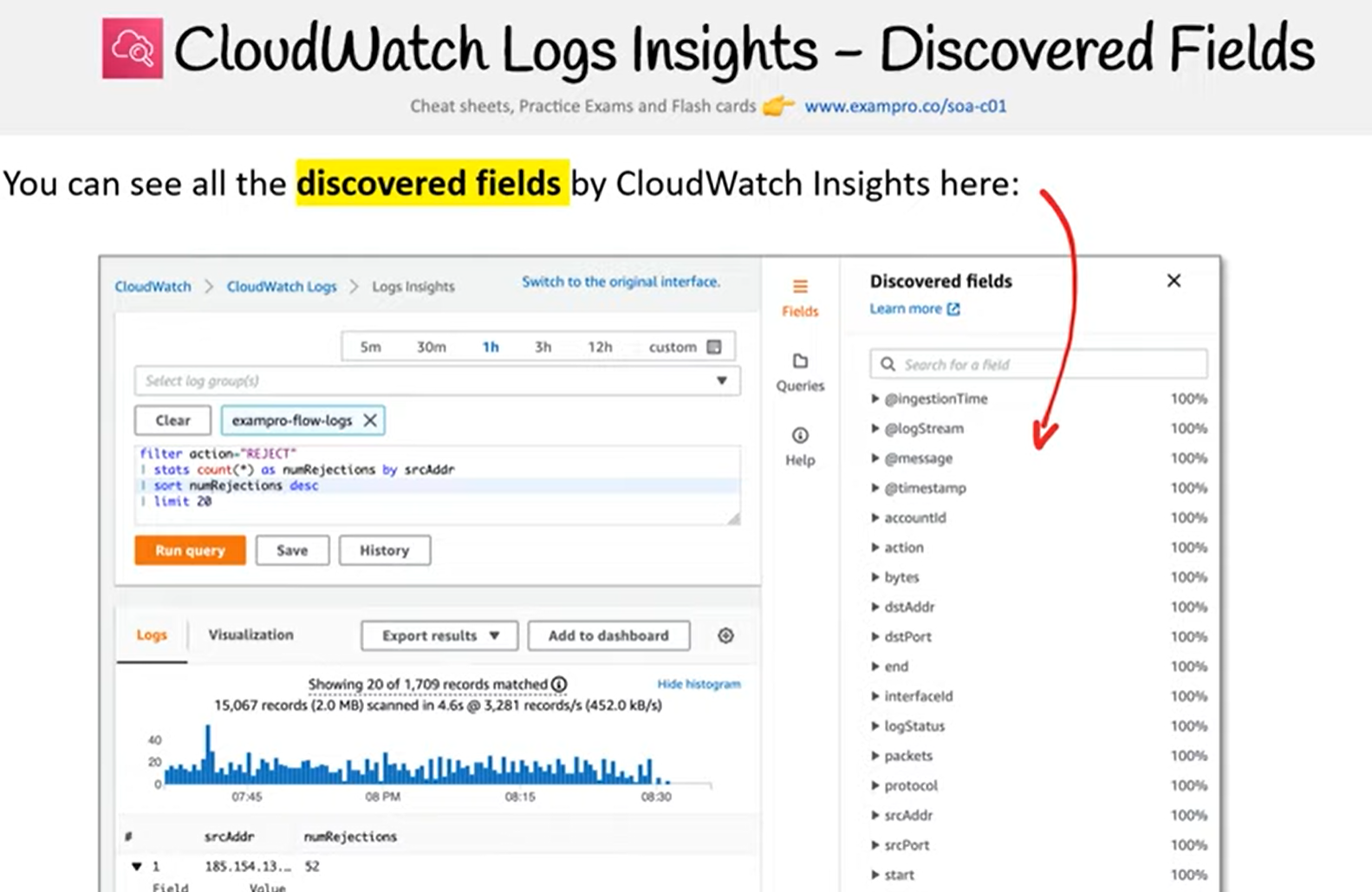






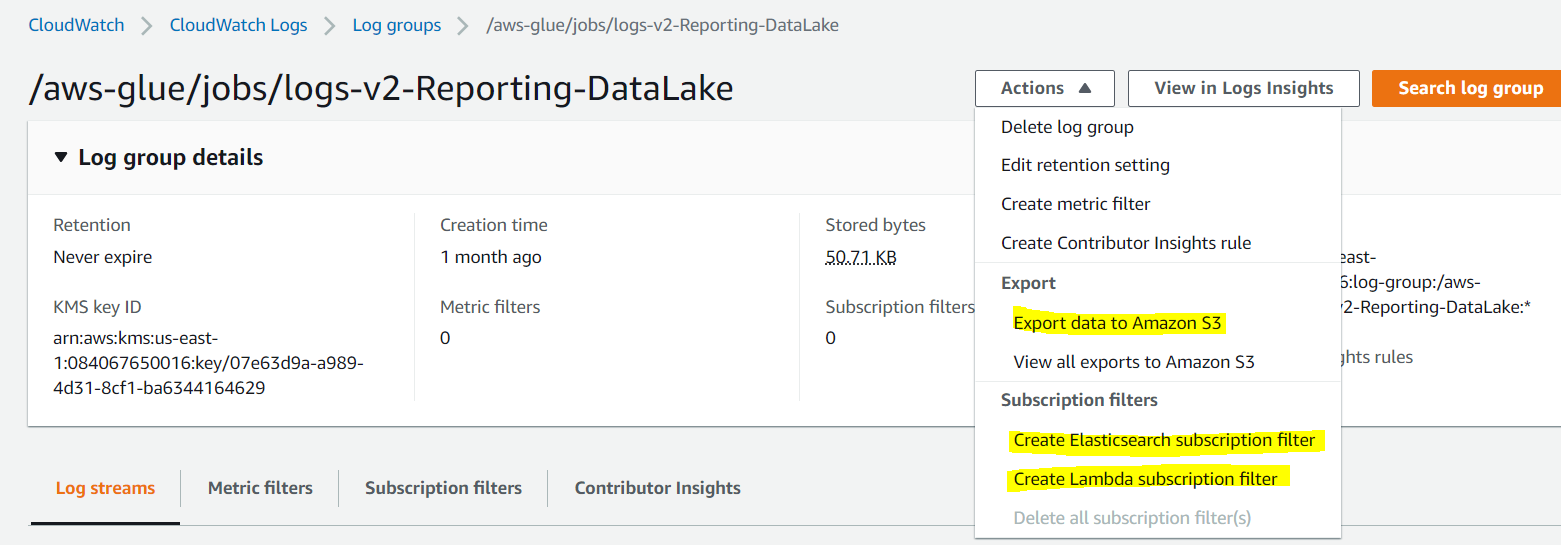






## AWS Certification Exam Practice Questions

1. Once we have our logs in CloudWatch, we can do a number of things such as: Choose 3. Choose the 3 correct answers:**[CDOP]**
   1. **Send the log data to AWS Lambda for custom processing or to load into other systems**
   2. **Stream the log data to Amazon Kinesis**
   3. **Stream the log data into Amazon Elasticsearch in near real-time with CloudWatch Logs subscriptions.**
   4. Record API calls for your AWS account and delivers log files containing API calls to your Amazon S3 bucket



1. You have decided to set the threshold for errors on your application to a certain number and once that threshold is reached you need to alert the Senior DevOps engineer. What is the best way to do this? Choose 3. Choose the 3 correct answers: **[CDOP]**
   1. **Set the threshold your application can tolerate in a CloudWatch Logs group and link a CloudWatch alarm on that threshold.**
   2. **Use CloudWatch Logs agent to send log data from the app to CloudWatch Logs from Amazon EC2 instances**
   3. Pipe data from EC2 to the application logs using AWS Data Pipeline and CloudWatch
   4. **Once a CloudWatch alarm is triggered, use SNS to notify the Senior DevOps Engineer.**
2. You are hired as the new head of operations for a SaaS company. Your CTO has asked you to make debugging any part of your entire operation simpler and as fast as possible. She complains that she has no idea what is going on in the complex, service-oriented architecture, because the developers just log to disk, and it’s very hard to find errors in logs on so many services. How can you best meet this requirement and satisfy your CTO? **[CDOP]**
   1. Copy all log files into AWS S3 using a cron job on each instance. Use an S3 Notification Configuration on the <code>PutBucket</code> event and publish events to AWS Lambda. Use the Lambda to analyze logs as soon as they come in and flag issues. (is not fast in search and introduces delay)
   2. Begin using CloudWatch Logs on every service. Stream all Log Groups into S3 objects. Use AWS EMR cluster jobs to perform adhoc MapReduce analysis and write new queries when needed. (is not fast in search and introduces delay)
   3. Copy all log files into AWS S3 using a cron job on each instance. Use an S3 Notification Configuration on the <code>PutBucket</code> event and publish events to AWS Kinesis. Use Apache Spark on AWS EMR to perform at-scale stream processing queries on the log chunks and flag issues. (is not fast in search and introduces delay)
   4. **Begin using CloudWatch Logs on every service. Stream all Log Groups into an AWS Elasticsearch Service Domain running Kibana 4 and perform log analysis on a search cluster.**(ELK – Elasticsearch, Kibana stack is designed specifically for real-time, ad-hoc log analysis and aggregation)
3. You use Amazon CloudWatch as your primary monitoring system for your web application. After a recent software deployment, your users are getting Intermittent 500 Internal Server Errors when using the web application. You want to create a CloudWatch alarm, and notify an on-call engineer when these occur. How can you accomplish this using AWS services? (Choose three.) [CDOP]
   1. Deploy your web application as an AWS Elastic Beanstalk application. Use the default Elastic Beanstalk CloudWatch metrics to capture 500 Internal Server Errors. Set a CloudWatch alarm on that metric.
   2. **Install a CloudWatch Logs Agent on your servers to stream web application logs to CloudWatch.**
   3. Use Amazon Simple Email Service to notify an on-call engineer when a CloudWatch alarm is triggered.
   4. **Create a CloudWatch Logs group and define metric filters that capture 500 Internal Server Errors. Set a CloudWatch alarm on that metric.**
   5. **Use Amazon Simple Notification Service to notify an on-call engineer when a CloudWatch alarm is triggered.**
   6. Use AWS Data Pipeline to stream web application logs from your servers to CloudWatch.