The main scenario

A healthcare facility wants to **collect patient data** about diagnostic codes, vital signs, tests and their results, and sensor data from devices such as pacemakers, blood pressure monitors, and heart rate monitors.

The facility want to **aggregate a snapshot of patient profiles** and cross-reference medical data with billing data. Your solution must incorporate **HIPAA compliance.**

1. How do you decouple protected data from processing and orchestration?
2. How do you automate the tracking of data flow?
3. What logical boundaries can be implemented between protected and general workflows?

Use AWS Artifact to comply BAA. Encryption at rest and in transit, User identification and authentication, auditing of access/use is also important.

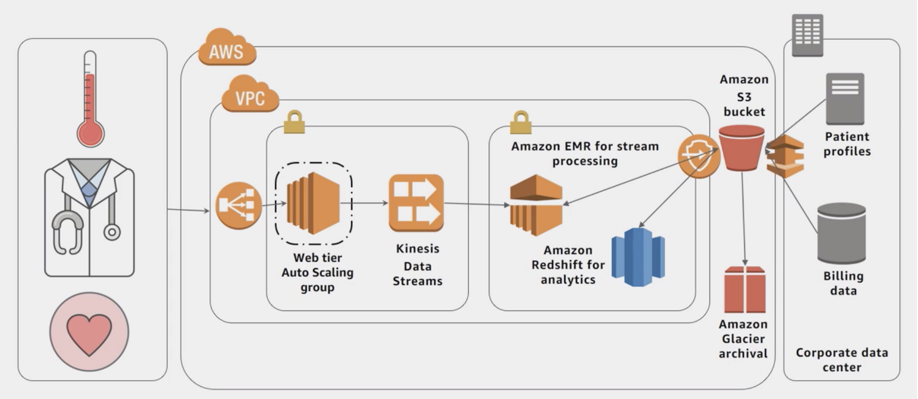


Figure 1: image

This involves moving data between Amazon EMR and relational database in corporate data center. This can be done by Snowball or Amazon Import/Export

Streaming ingestion

First, it uses MQTT topics, creating AWS IoT rules, and IAM permissions in IoT Core.

Then, Kinesis Data Streams enable to process and analyze the streaming data.

- Secured by AWS KMS for data encryption, encrypted endpoints using TLS.
- Integration with KMS, IAM, cloudTrail

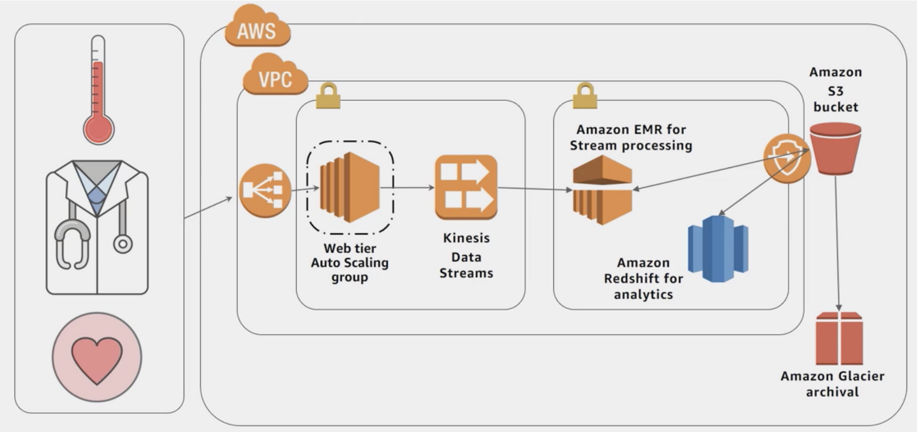

AWS
VPC
Web tier Auto Scaling group
Kinesis Data Streams
Amazon EMR for Stream processing
Amazon Redshift for analytics
Amazon S3 bucket
Amazon Glacier archival

Figure 2: image

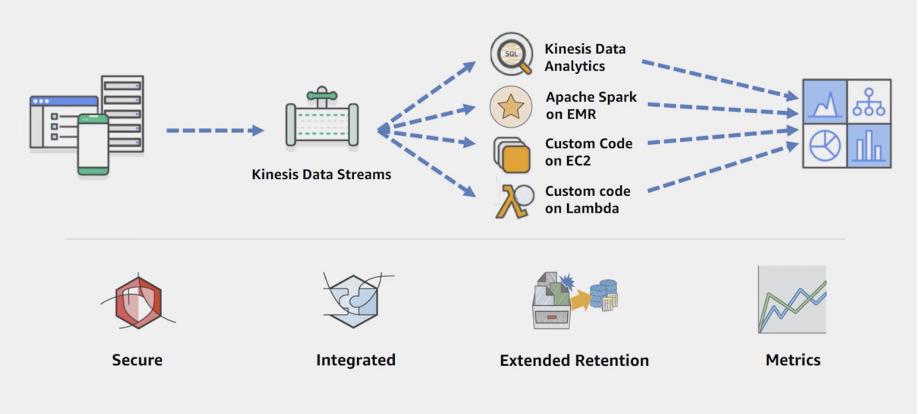


Figure 3: image

- Extended retention (1 day ~ 7days)
- Metrics (stream and shard level), auditing achieved by cloudtrail.

CloudTrail does not currently support the monitoring of data plane operation (CRUD actions)

Kinesis stream automatically encrypts sensitive data as a producer enters into the stream

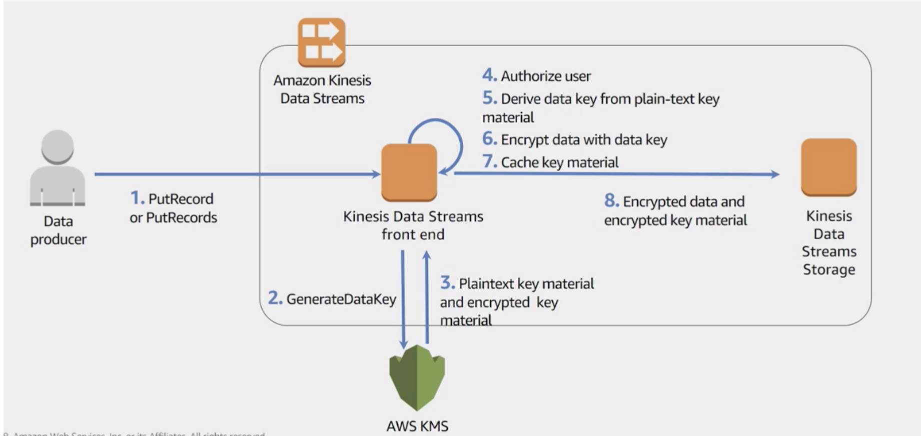

Amazon Kinesis Data Streams
Data producer
1. PutRecord or PutRecords
Kinesis Data Streams front end
4. Authorize user
5. Derive data key from plain-text key material
6. Encrypt data with data key
7. Cache key material
8. Encrypted data and encrypted key material
Kinesis Data Streams Storage
2. GenerateDataKey
3. Plaintext key material and encrypted key material
AWS KMS

Figure 4: image

Kinesis Stream uses AWS KMS master key for encryption. **To read from / write to encrypted stream, producer or consumer application must have the permission to access to the master key**

How about batch ingestion? Having a single node for ingesting large amounts of data is not appropriate. So you may use EMR.

Sqoop is used to ingest data from on-premise data center.

How about securing the EMR cluster?

You can launch the cluster into VPC and connect your data center to VPC through an VPN connection enabling cluster as a remote resource on your internal network.

You can launch your cluster in either private or public subnet. This means you do not need internet connectivity for cluster. But you may need NAT gateway, enabling access to outside of the VPC.

When you launch your cluster, you should configure two Security Groups. One for master node, the other for core & task node (those two types of instances share the same SG!). Optionally, one for the amazon EMR resources used to manage clusters in private subnet.

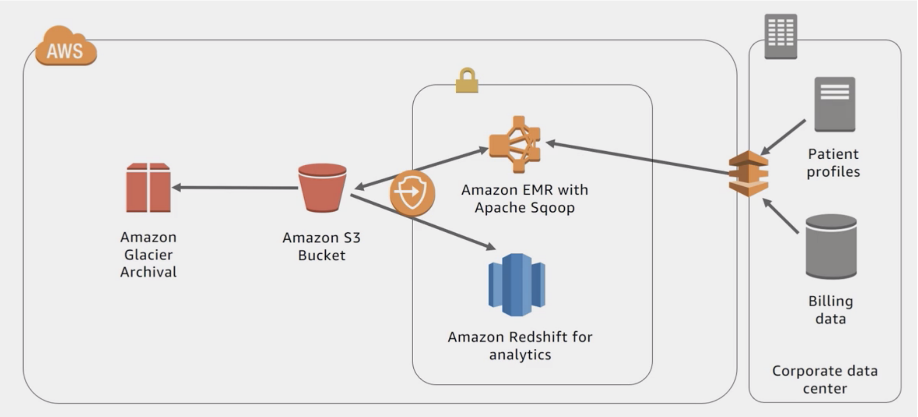
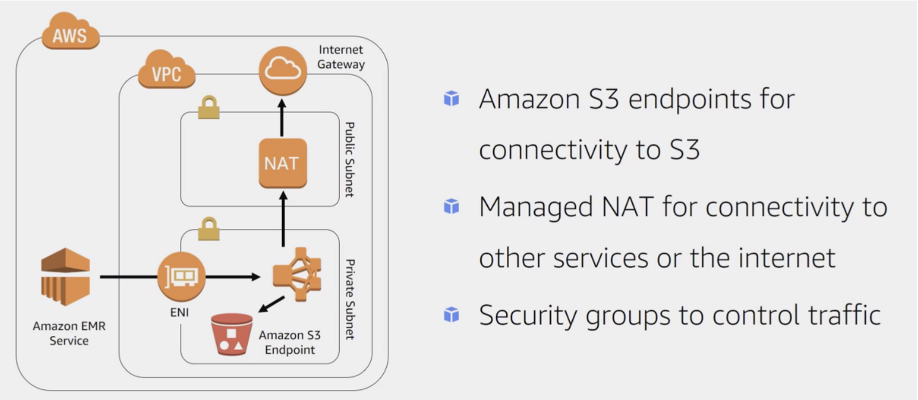


Figure 5: image



- Amazon S3 endpoints for connectivity to S3
- Managed NAT for connectivity to other services or the internet
- Security groups to control traffic

Figure 6: image

Class Discussion 1

If you launch an Amazon EMR cluster using the default security groups, two groups are created for public subnets. What are they?

A: create ElasticMapReduce-master , ElasticMapReduce-slave. The inbound and outbound rules of these SGs ensure that the master and core, task nodes can communicate with each other properly.

Class Discussion 2

If the Amazon EMR cluster is launch in a private subnet using the default security groups, a third security group is created. What is it?

A: ElasticMapReduce-ServiceAccess. It has no inbound rules, but it has outbound rules that allow traffic over HTTPS (port 8443) to the other managed security groups in private subnets. So, if you launch other EMR cluster in the same VPC, using the same security group, instances in this cluster can communicate with any other instances in any other EMR cluster with the same SG in this VPC.

- You can launch the cluster with default security groups using console, CLI, SDK.
- If you use your default security groups, there are no need to change the existing code or to add parameters in your CLI commands.
- When launching a cluster with default security group using console and the default security group doesn't exist, it's automatically created before your cluster is launched. If they do exist, they are automatically assigned.

Amazon EMR and Kerberos!

- Amazon EMR 5.10.0 and later supports Kerberos
- Services and users that authenticate are called *principals*
  - principal is a elementary entity that can be granted with *tickets*
  - There are three kinds of it: Users, Services, Hosts
- Principals exist in Kerberos *realm*.
- The Key Distribution Center (KDC) manages authentication (provides the means to authenticate)
- The KDC issues tickets.
- Principals can be from other Realms. This requires Cross-Realm Trust. This is usually used for authenticating users from Active Domain.
- Quote from official document: Note that a Kerberos Server manages one Realm only, a Realm can be managed by more than one Kerberos server :

this is mandatory to avoid a single point of failure, if a Kerberos server halts for any reason.

- Amazon EMR configures Kerberos for the applications, the components, and subsystems so that they are authenticated with each other.
- You can use cross-realm trust to authenticate a user from different KDC, or you can manually add that user to the cluster dedicated KDC.
- You can then configure Hadoop user directories so that they can use Kerberos credentials to run jobs, and connect to the cluster.

In EMR storage permissions,

- EMRFS storage-based permissions
- Consistent View (optional view) (provides consistency checking, read-after-write for new PUT request) (it uses additional DynamoDB table for storing metadata and track consistency in S3)
- Data encryption
- Each security configuration is stored in EMR, rather than cluster configuration, so you can easily re-use it when launching another cluster

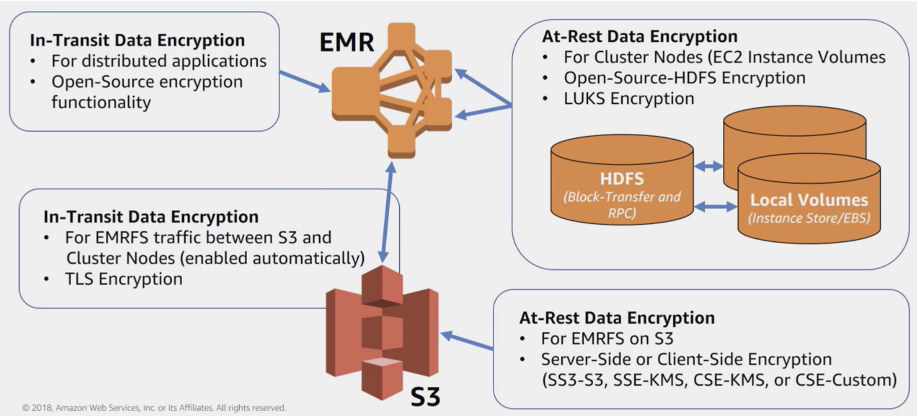

In-Transit Data Encryption
• For distributed applications
• Open-Source encryption functionality
EMR
At-Rest Data Encryption
• For Cluster Nodes (EC2 Instance Volumes
• Open-Source-HDFS Encryption
• LUKS Encryption
HDFS
(Block-Transfer and RPC)
Local Volumes
(Instance Store/EBS)
In-Transit Data Encryption
• For EMRFS traffic between S3 and Cluster Nodes (enabled automatically)
• TLS Encryption
At-Rest Data Encryption
• For EMRFS on S3
• Server-Side or Client-Side Encryption (SS3-S3, SSE-KMS, CSE-KMS, or CSE-Custom)
S3
© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Figure 7: image

SSE-C is not available for encryption at rest in S3.

EMR and Apache Hive

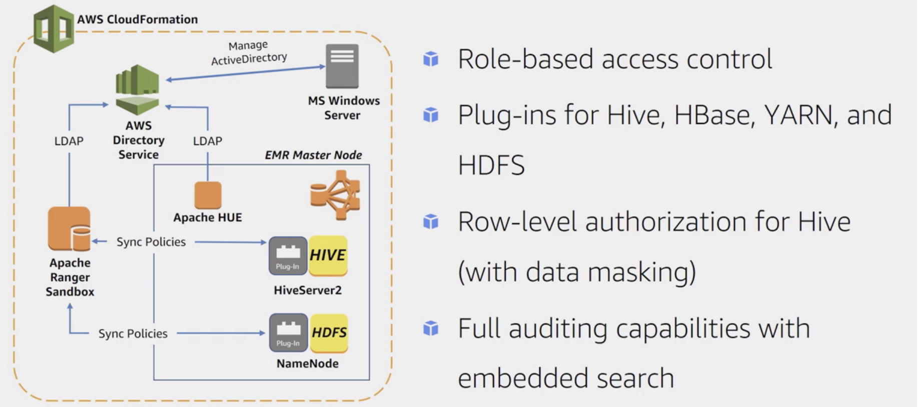
Four features of Hive that are **specific to Amazon EMR**:

- Load table partitions automatically from Amazon S3

- Specify an off-instance metadata store (in default, metadata sits in master node, and ceases to exist when the cluster terminates)
- Write data directly to Amazon S3 (no temporary files are created. produces SIGNIFICANT performance boost. BUT, from hive's perspective, S3 and HDFS behaves differently. So, you cannot read or write within the same statement to the same table located in S3 (you may create temporary table in the cluster's local HDFS))
- Access resources located in Amazon S3

Apache ranger

It is a role-based access control framework in order to enable, monitor, manage the data security across the Hadoop data platform.



- Role-based access control
- Plug-ins for Hive, HBase, YARN, and HDFS
- Row-level authorization for Hive (with data masking)
- Full auditing capabilities with embedded search

Figure 8: image

EMR Log Auditing

CloudTrail audits EMR API calls, KMS API calls, S3 API calls, . . .

Inside of the EMR itself, many types of logs are written to the master node. (e.g. steps, bootstrap action, instance state logs.) (by default, cluster created by console automatically archive logs to S3, but for cluster created by API or CLI, you must configure the S3 archiving function manually)

Apache Hadoop writes jobs, tasks, task attempt.

RedShift Security

- IAM roles to access data on Amazon S3

- IAM SSO authentication
- SSL to secure data in transit
- Encryption to secure data at rest
- No direct access to compute nodes
- Support for Amazon VPC
- User audit logging and AWS CloudTrail integration

Also you can use your Redshift security to limit data access using *views*

RedShift Networking - Enhanced VPC routing

- Enforces all the traffic by COPY and UNLOAD command route only using VPC (otherwise, goes through internet)
- Query traffics flows only through customer VPC
- Strict data traffic management
- Amazon S3 endpoint to access Amazon S3
- Locked down security groups
- SSL certificate for each Amazon Redshift cluster
- Also you can monitor it by viewing VPC flow logs
- And every feature provided by VPC is also usable (NACL, IGW, SG, Endpoint . . .)

Redshift IAM authorization

- IAM support for data LOAD / UNLOAD
- IAM roles for LOAD / UNLOAD operations
- Cluster can have access to specific S3 buckets
- Simplify credentials management
- Access to AWS KMS for encryption

Redshift Encryption

It is an unmodified attribute! It must be enabled during cluster launch process. To make unencrypted cluster to encrypted one, UNLOAD it, and COPY that into your wanted setting.

In transit

- Amazon Redshift API calls are made using HTTPS
- SSL certificate for each Amazon Redshift cluster required ()
- Best practice is to make unique SSL certificate for each cluster.

At rest

- Enable cluster encryption
- Encrypted via:
  - AWS KMS
  - Hardware security module
- Supports server-side encryption using SSE-KMS and SSE-S3

Sample Question

A data engineer is building a high frequency transactional processing store on Amazon EMR and is concerned about the Amazon S3 eventual consistency model. But they are receiving multiple S3ServiceException responses. The exception response says that the specified key doesn't exist! How would you debug it?

A: This could be due to eventual consistency in Amazon S3. Use the Amazon EMRFS consistent view feature.

Sample Question 2

Just remember 1. gzip, snappy doesn't support file splitting, 2. avro is all about data serialization. And **bzip2 supports file splitting (partitioning)**