AWS Hippa Compliance

AWS HIPAA Compliance

Amazon Web Services (AWS) to create HIPAA (Health Insurance Portability and Accountability Act)-compliant applications. HIPAA Privacy and Security Rules for protecting Protected Health Information (PHI).

HIPAA and HITECH impose requirements related to the use and disclosure of PHI, appropriate safeguards to protect PHI, individual rights, and administrative responsibilities.

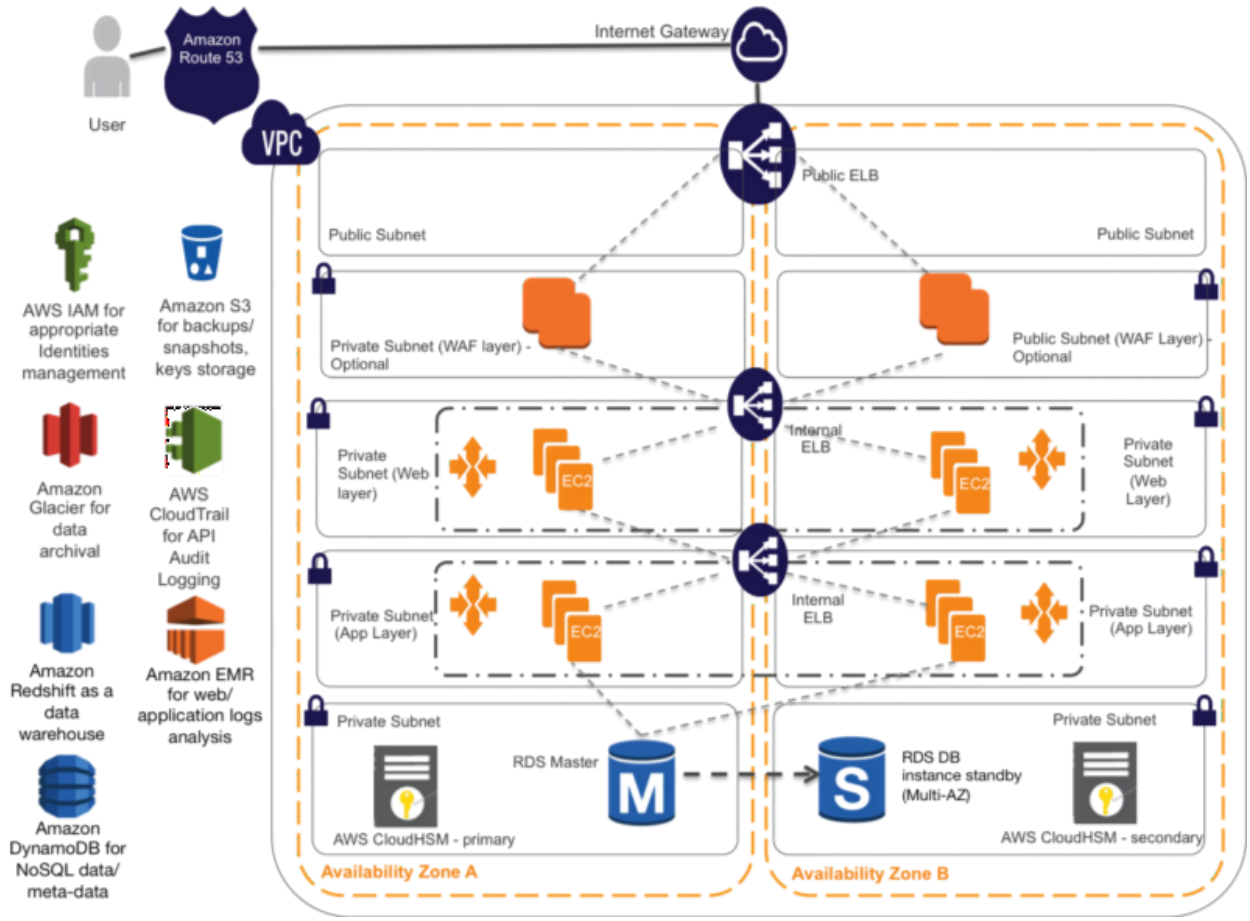
Covered entities and their business associates can use the secure, scalable, low-cost IT provided by Amazon Web Services (AWS) to architect applications in alignment with HIPAA and HITECH compliance requirements. AWS services and data centers have multiple layers of operational and physical security to help ensure the integrity and safety of customer data.
AWS, offers a standardized Business Associate Addendum (BAA) for such customers.
AWS service in an account designated as a HIPAA Account, but they may only process, store and transmit PHI using the HIPAA-eligible services defined in the AWS BAA.

Amazon Web Services which are HIPAA Compliant

- Amazon DynamoDB
- Amazon Elastic Block Store (Amazon EBS)
- Amazon Elastic Compute Cloud (Amazon EC2)
- Elastic Load Balancing
- Amazon Elastic MapReduce (Amazon EMR)
- Amazon Glacier
- Amazon Redshift
- Amazon Relational Database Service (Amazon RDS) for MySQL
- Amazon RDS for Oracle
- Amazon Simple Storage Service (Amazon S3)

HIPAA architectures on AWS

AWS provides multiple services to deploy a highly available, scalable, secure application stack, which can serve a limitless variety of healthcare applications and use cases. In this blog, we will embark on a journey into HIPAA-eligible architectures by scoping the discussion to the following deployment diagram, which can be adopted as a starting point for building a HIPAA-eligible, web-facing application.



The underlying theme to this architecture is **encryption everywhere**

HIPAA ON AWS PROCESS

1) Obtain a Business Associate Agreement with AWS

Once you have determined that storing, processing, or transmitting protected health information (PHI) is absolutely necessary, before moving any of this data to AWS infrastructure you must contact AWS and make sure you have all the necessary contracts and a Business Associate Agreement (BAA) in place. These contracts will serve to clarify and limit, as appropriate, the permissible uses and disclosures of protected health information.

2) Authentication and Authorization

The authentication and authorization mechanisms you define for your HIPAA-eligible system must be documented as part of a System Security Plan (SSP) with all roles and responsibilities documented in detail along with a configuration control process that specifies initiation, approval, change, and acceptance processes for all change requests. Although the details of defining these processes won't be discussed here, the AWS Identity and Access Management (AWS IAM) service does offer the granular policies required for achieving the necessary controls under HIPAA and HITECH.

enable multi-factor authentication (MFA) on your AWS root account and lock away the access keys

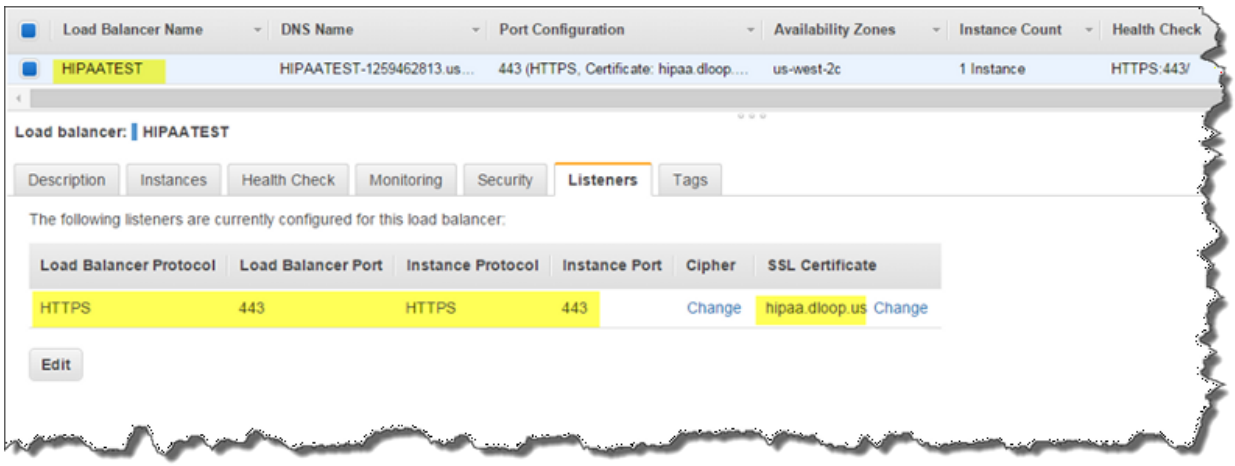
IAM account that has significant privileges in your AWS account

3) Web and Application Layers

DNS resolution is relatively straightforward and can be achieved using Amazon Route 53. Just be sure not to use any PHI in the URLs.

Amazon Elastic Load Balancer Configuration

The primary entity that receives the request from Amazon Route 53 is an Internet-facing Elastic Load Balancer. There are multiple ways in which an ELB load balancer can be configured, as explained here. To protect the confidential PHI data, you must enable secure communication options only, like HTTPS-based or TCP/SSL-based end-to-end communication. Although you can use TCP/SSL pass-through mode on the ELB load balancer for your web tier requests, using this option limits the use of some of the HTTP/HTTPS specific features like sticky sessions and X-Forward-For headers. For this reason, many startups prefer to make use of HTTPS-based communication on ELB, as shown in the following screenshot.



As shown in the configuration, there's a single listener configured that accepts HTTPS requests on port 443 and sends requests to back-end instances using HTTPS on port 443. Because HTTPS is used for the front-end connection, you must create the certificate as per your publicly accessible domain name, get the certificate signed by a CA (for an internal load balancer you can use a self-signed certificate as well), and then upload the certificate using AWS IAM, which manages your SSL certificates, as explained in the ELB documentation. This certificate is then utilized to decrypt the HTTPS-based encrypted requests that are received by the ELB load balancer.

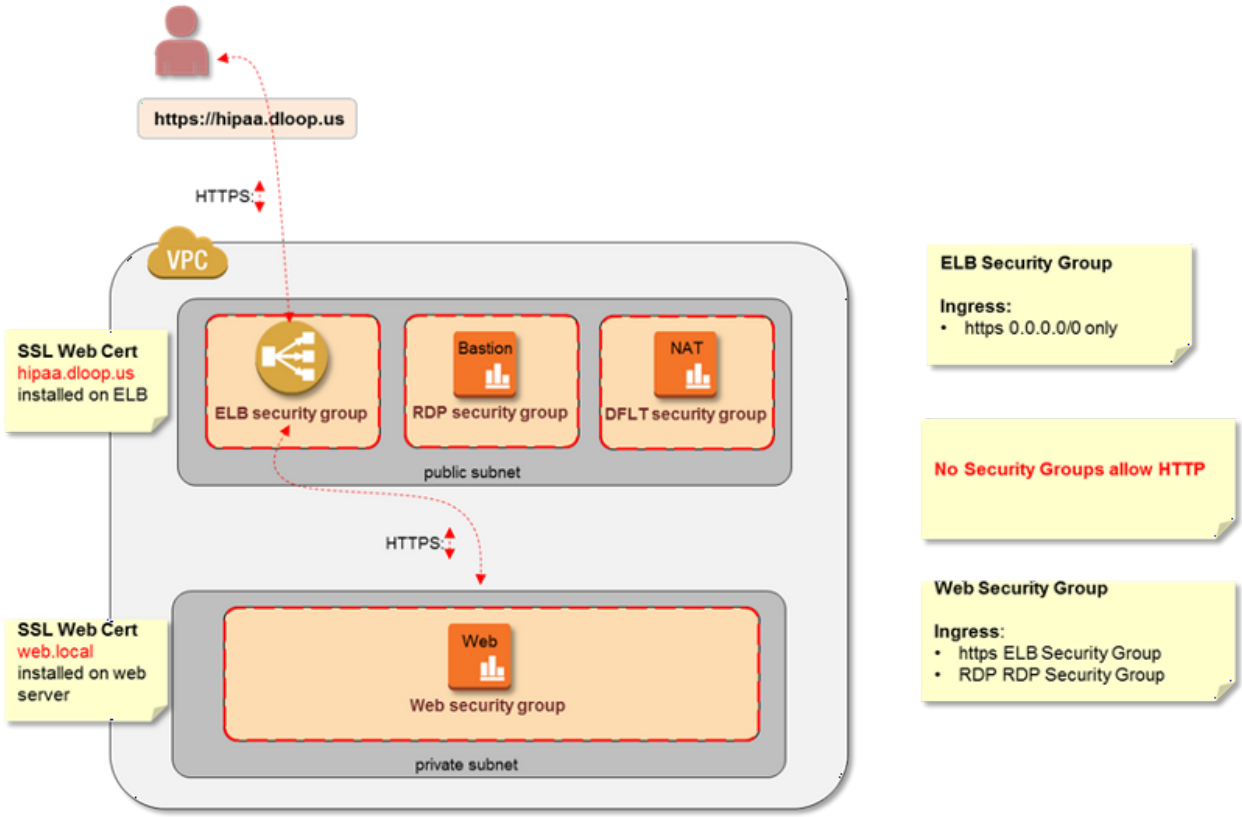
To route the requests from the ELB load balancer to the back-end instances, you must use back-end server authentication so that the communication is encrypted throughout. You can enable this by creating a public key policy that uses a public key for authentication. You use this public key policy to create a back-end server authentication policy. Finally, you enable the back-end server authentication by setting the back-end server authentication policy with the back-end server port, which in this case would be 443 for an HTTPS protocol. For an example of how to set this up easily using OpenSSL, check out the ELB documentation and Apache Tomcat's documentation on certificates.

WAF/IDS/IPS Layer

Many of our customers make use of an extra layer of security (like web application firewalls and intrusion detection/prevention solutions) in front of their web layer to avoid any potential malicious attacks to their sensitive applications. There are multiple options available in the AWS Marketplace to provision tools like WAF/IDS/IPS, etc. So you could start from there instead of setting it up from scratch on an EC2 instance.

Web Layer

The next layer is the web tier, which could be auto-scaled for high availability and placed behind an internal ELB load balancer with only a HTTPS listener configured. To further secure the access to web servers, you should open up your web server instance's' security group to accept requests only from the designated load balancer, as shown in the following diagram.



App Layer

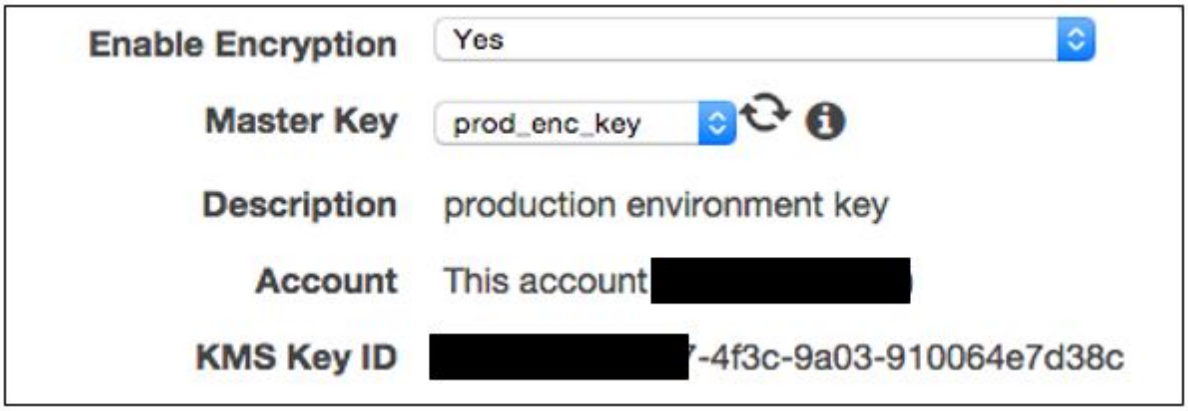
Encryption of traffic between the web layer and app layer will look similar to the setup in the preceding diagram. Again, there will be an internal ELB load balancer with HTTPS listener configured. On the application servers, SSL certificates are set up to keep the communication channel encrypted end-to-end.

Both the app and web layers should also be in private subnets with auto-scaling enabled to ensure a highly responsive and stable healthcare application.

4) Database Layer

The easiest way to get started with database encryption is to make use of Amazon RDS (MySQL or Oracle engine). To protect your sensitive PHI data, you should consider the following best practices for Amazon RDS:

- You should have access to the database enabled only from the application tier (using appropriate security group/NACL rules).
- Any data that has the potential to contain PHI should always be encrypted by enabling the encryption option for your Amazon RDS DB instance, as shown in the following screenshot. Data that is encrypted at rest includes the underlying storage for a DB instance, its automated backups, read replicas, and snapshots.



- For encryption of data in-transit, MySQL provides a mechanism to communicate with the DB instance over an SSL channel, as described here. Likewise, for Oracle RDS you can configure Oracle Native Network Encryption to encrypt the data as it moves to and from a DB instance.
- For encryption of data at rest, you could also make use of Oracle's Transparent Data Encryption (TDE) by setting the appropriate parameter in the Options Group associated with the RDS instance. With this, you can enable both TDE tablespace encryption (encrypts entire application tables) and TDE column encryption (encrypts individual data elements that contain sensitive data) to protect your PHI data. You could also store the Amazon RDS Oracle TDE Keys by leveraging AWS CloudHSM, a service that provides dedicated Hardware Security Module (HSM) appliances within the AWS cloud. More details on this integration are available here.

  For additional discussion on Amazon RDS encryption mechanisms, please refer back to the whitepaper.

5) Backup/Restore

To protect your patient data, you should be vigilant about your backup and restore processes. Most AWS services have mechanisms in place to perform backup so that you can revert to a last known stable state if any changes need to be backed out. For example, features like EC2 AMI creation or snapshotting (as in the Amazon EBS, Amazon RDS, and Amazon Redshift services) should be able to meet the majority of backup requirements.

You can also make use of third-party backup tools, which integrate with Amazon S3 and Amazon Glacier to manage secure, scalable, and durable copies of your data. When using Amazon S3, you have multiple ways to encrypt your data at rest and can leverage both client-side encryption and server-side encryption mechanisms. Details on these options are available in the Amazon S3 documentation.

PHI in S3 buckets should always be encrypted. You can also enforce the server-side encryption (SSE) option on any of the buckets by adding the following condition to your Amazon S3 bucket policy:

```
"Condition": {
"StringEquals": {
"s3:x-amz-server-side-encryption":"AES256"
},
"Bool": {
"aws:SecureTransport": "true"
}
},
```

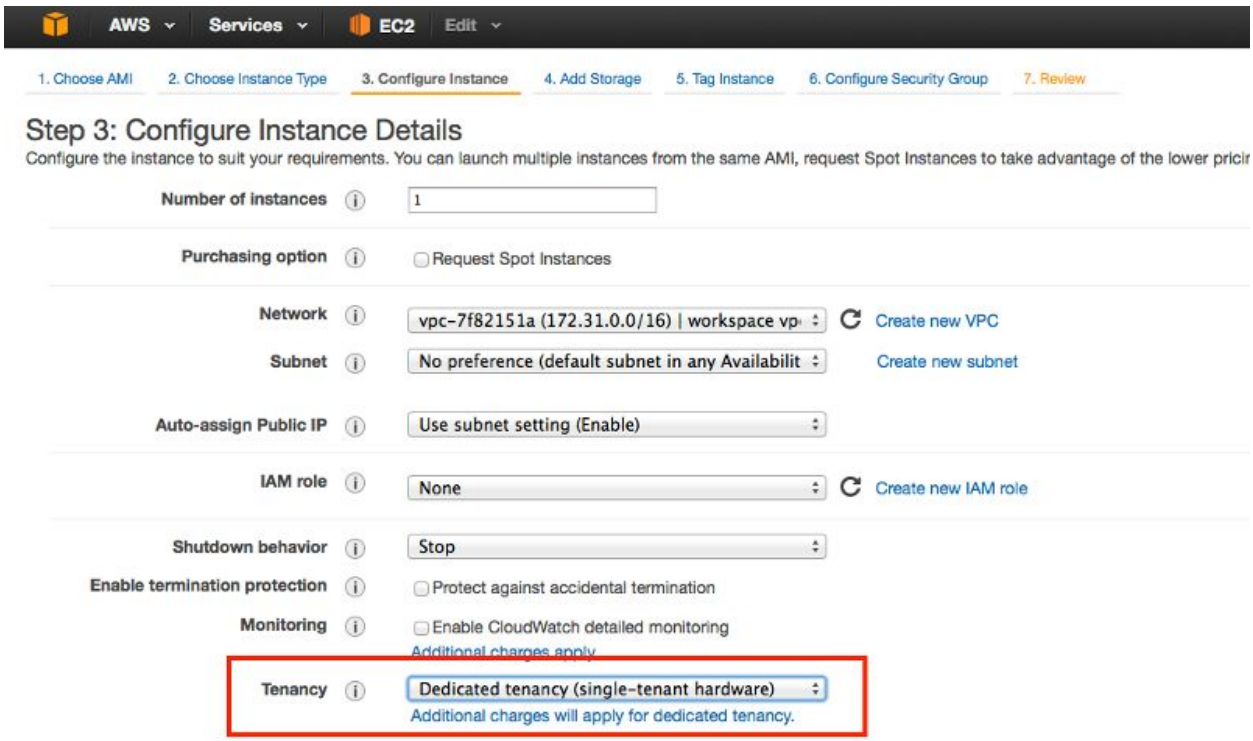
For security of data in transit, you should always use Secure Sockets Layer (SSL) enabled endpoints for all the services, including Amazon S3 for backups. If you are enabling backup of your data from the EC2 instances in a VPC to Amazon S3, then you could also make use of VPC endpoints for Amazon S3. This feature creates a private connection between your private VPC and Amazon S3 without requiring access over the Internet or a NAT/proxy device.

6) EC2 and EBS requirements

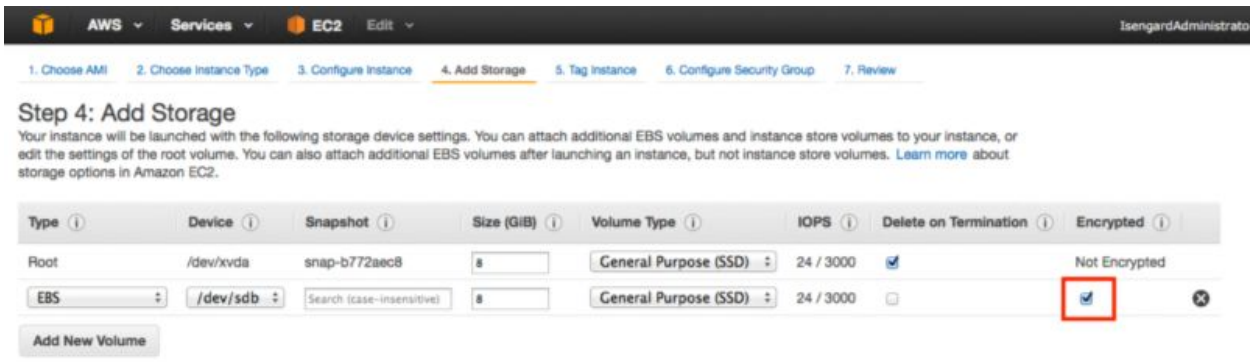
Amazon EC2 is a scalable, user-configurable compute service that supports multiple methods for encrypting data at rest, ranging from application-level or field-level encryption of PHI as it is processed, to transparent data-encryption features of commercial databases, to the use of third-party tools. For a more complete discussion of the options, see the whitepaper.

In the next example, we show you a simple approach to architecting HIPAA-eligible web servers.

First, you must be sure that your EC2 instance is running on hardware that is dedicated to a single customer by using a dedicated instance. You can do this by setting the tenancy attribute to "dedicated" on either the Amazon VPC that the instance is launched in, the Auto-Scaling Launch Configuration, or on the instance itself, as shown in the following screenshot.



Because Amazon Elastic Block Store (Amazon EBS) storage encryption is consistent with HIPAA guidance at the time of this blog writing, the easiest way to fulfill the at-rest encryption requirement is to choose an EC2 instance type that supports Amazon EBS encryption, and then add the encrypted EBS volume to your instance. (See the EBS link for a list of instance types.)



You should keep all of your sensitive PHI data on the encrypted EBS volumes, and be sure never to place PHI on the unencrypted root volume.
You might want to take some additional precautions to ensure that the unencrypted volume does not get used for PHI. For example, you can consider a partner solution from the AWS Marketplace, which offers full-drive encryption to help you feel more at ease. This will help to ensure that if there ever is a program (such as a TCP core dump) that uses the root drive as temporary storage or scratch space without your knowledge, it will be encrypted. Other startups have developed their own techniques for securing the root volume by using Logical Volume Management (LVM) to repartition the volume into encrypted segments and to make other portions read-only.

7) Key Management

At every turn in this architecture, we have mentioned encryption. Ensuring end-to-end encryption of our PHI is an essential component of keeping our data secure. Encryption in flight protects you from eavesdroppers, and encryption at rest defends against hackers of the physical devices. However, at some point we do need to open this ciphertext PHI in order to use it in our application. This is where key management becomes a “key” piece of the implementation (pun intended).

AWS does not place limitations on how you choose to store or manage your keys. Essentially, there are four general approaches to key management on AWS:

1. Do it yourself
2. Partner solutions
3. AWS CloudHSM
4. AWS KMS

A full discussion (or even a good starting discussion) on key management far exceeds what we can provide in a single blog entry, so we will just provide some general advice about key management as it relates to HIPAA.

The first piece of advice is that you should strongly consider the built-in AWS option. All of the checkbox encryption methods — such as Amazon S3 server-side encryption, Amazon EBS encrypted volumes, Amazon Redshift encryption, and Amazon RDS encryption make it very easy to keep your PHI encrypted and you should explore these options to see if these tools meet your BAA requirements and HHS guidance. These methods automate or abstract many of the tasks necessary for good key maintenance such as multifactor encryption and regular key rotation. AWS handles the heavy lifting and ensures that your encryption methods are using one of the strongest block ciphers available.

If you need to create a separation of duties between staff that maintain the keys vs. developers who work with the keys, or if you would simply like additional control of your keys and want to be able to easily create, control, rotate and use your encryption keys then you should look at using the Amazon Key Management Service (KMS). This service is still integrated with AWS SDKs and other AWS services like AWS CloudTrail, which can help provide auditable logs to help meet your HIPAA compliance requirements.

If you need additional controls beyond what is provided by AWS, you should be sure that you have proper security experts who can ensure the safe management of your encryption keys. Remember, a lost key could render your entire dataset useless, and AWS Support will not have any way to help a problematic situation.

For more on encryption and key Management in AWS, check out this video from last year's re:Invent, and read the Securing Data at Rest with Encryption whitepaper.

8) Logging and Monitoring

Logging and monitoring of system access will play a starring role in your HIPAA-eligible architecture. The goal is to put auditing in place to allow security analysts to examine detailed activity logs or reports to see who had access, IP address entry, what data was accessed, etc. The data should be tracked, logged, and stored in a central location for extended periods of time in case of an audit.

At the AWS account level, be sure to launch AWS CloudTrail and immediately start recording all AWS API calls. You should also launch AWS Config, which will provide you with an AWS resource inventory, configuration history, and configuration change notifications.

You will also need to monitor and maintain the logs of your AWS resources for keeping a record of system access to PHI as well as running analytics that could serve as part of your HIPAA Security Risk Assessment. One way to do this is with AWS CloudWatch, a monitoring service that you can use to collect server logs from your EC2 instances as well as logs from the Amazon RDS DB instance, Amazon EBS volumes, and the ELB elastic load balancer. You can even develop custom metrics to obtain the necessary log information from your own applications.

CloudWatch has other useful features:

- View graphs and statistics on the console
- Set up alarms to automatically notify you of abnormal system behavior
- Capture network traffic in a single repository through the integration of CloudWatch with VPC Flow Logs

With all these logging mechanisms, you want to be sure that no PHI is actually stored in the logs. This usually requires some special attention. For example, sometimes you might need to encrypt PHI in your custom metric before sending to AWS CloudTrail. You also should be aware of everything that is coming into the logs. For example, the combination of session user and IP address coming from the ELB logs is

considered PHI in some situations, so you should catch these special circumstances to be sure PHI is fully scrubbed from the logs.
Finally, Amazon S3 is a fantastic repository for all these logs. However, take extra precautions to lock down the permissions for log access of these highly sensitive data sets. You might want to consider some more stringent access requirements such as requiring multi-factor authentication to read the logs, turning on versioning to retain any logs that get deleted, or even setting up cross-region replication to keep a second copy of the logs in an entirely different AWS account.

AWS Environment

AWS environment to meet your HIPAA Compliance needs, but we will also provide ongoing managed services 24/7 to help ensure that your AWS environment remains HIPAA Compliant. Our HIPAA Compliance Support Plan for AWS includes a comprehensive suite of security and support features designed to specifically address the HIPAA and HITECH standards, including the necessary levels of encryption within AWS.

Encryption and Protection of PHI in AWS

implement encryption, customers may evaluate and take advantage of the encryption features native to the HIPAA-eligible services or they can satisfy the encryption requirements through other means consistent with the Guidance. The following sections provide high-level details about using available encryption features in each of the HIPAA-eligible services and other patterns for encrypting PHI. A final section describes how AWS KMS can be used to encrypt the keys used for encryption of PHI on AWS.

Amazon EC2

Amazon EC2 is a scalable, user-configurable compute service that supports multiple methods for encrypting data at rest.
For example, customers might select to perform application- or field-level encryption of PHI as it is processed within an application or database platform hosted in an Amazon EC2 instance.
Approaches range from encrypting data using standard libraries in an application framework such as Java or .NET; leveraging Transparent Data Encryption features in Microsoft SQL or Oracle; or by integrating other third-party and software as a service (SaaS)-based solutions into their applications. Customers can choose to integrate their applications running in Amazon EC2 with AWS KMS SDKs, simplifying the process of key management and storage. Customers can also implement encryption of data at rest using file-level or full disk encryption (FDE) by utilizing third-party software from AWS Marketplace Partners or native file system encryption tools (such as dm-crypt, LUKS, etc.).

Network Control
Network traffic containing PHI must encrypt data in transit. For traffic between external sources (such as the Internet or a traditional IT environment) and Amazon EC2, customers should use industry-standard transport encryption

mechanisms such as TLS or IPsec virtual private networks (VPNs), consistent with the Guidance. Internal to an Amazon Virtual Private Cloud (VPC) for data traveling between Amazon EC2 instances, network traffic containing PHI must also be encrypted; most applications support TLS or other protocols providing in-transit encryption that can be configured to be consistent with the Guidance. For

applications and protocols that do not support encryption, sessions transmitting PHI can be sent through encrypted tunnels using IPsec or similar implementations between instances.

Amazon EC2 instances that customers use to process, store, or transmit PHI are run on Dedicated Instances, which are instances that run in an Amazon VPC on hardware dedicated to a single customer. Dedicated Instances are physically isolated at the host hardware level from instances that are not Dedicated Instances and from instances that belong to other AWS accounts. For more information on Dedicated Instances, see http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/dedicated-instance.html.

Customers can launch Amazon EC2 Dedicated Instances in several ways:

- Set the tenancy attribute of an Amazon VPC to "dedicated" so that all instances launched into the Amazon VPC will run as Dedicated Instances
- Set the placement tenancy attribute of an Auto-Scaling Launch Configuration for instances launched into an Amazon VPC
- Set the tenancy attribute of an instance launched into an Amazon VPC

Amazon Virtual Private Cloud offers a set of network security features well-aligned to architecting for HIPAA compliance. Features such as stateless network access control lists and dynamic reassignment of instances into stateful security groups afford flexibility in protecting the instances from unauthorized network access. Amazon VPC also allows customers to extend their own network address space into AWS, as well as providing a number of ways to connect their data centers to AWS. VPC Flow Logs provide an audit trail of accepted and rejected connections to instances processing, transmitting or storing PHI. For more information on Amazon VPC, see http://aws.amazon.com/vpc/.

Amazon Elastic Block Store

Amazon EBS encryption at rest is consistent with the Guidance that is in effect at the time of publication of this whitepaper. Because the Guidance might be updated, customers should continue to evaluate and determine whether Amazon EBS encryption satisfies their compliance and regulatory requirements. With Amazon EBS encryption, a unique volume encryption key is generated for each EBS volume; customers have the flexibility to choose which master key from the AWS Key Management Service is used to encrypt each volume key. For more information

Amazon Redshift

Amazon Redshift provides database encryption for its clusters to help protect data at rest. When customers enable encryption for a cluster, Amazon Redshift encrypts all data, including backups, by using hardware-accelerated Advanced Encryption Standard (AES)-256 symmetric keys. Amazon Redshift uses a four-tier, key-based architecture for encryption. These keys consist of data encryption keys, a database key, a cluster key, and a master key. The *cluster key* encrypts the database key for the Amazon Redshift cluster. Customers can use either AWS KMS or an AWS CloudHSM (Hardware Security Module) to manage the cluster key. Amazon Redshift encryption at rest is consistent with the Guidance that is in effect at the time of publication of this whitepaper. Because the Guidance might be updated, customers should continue to evaluate and determine whether Amazon Redshift encryption satisfies their compliance and regulatory requirements

Amazon S3

Customers have several options for encryption of data at rest when using Amazon S3, including both server-side and client-side encryption and several methods of managing keys

Amazon Glacier

Amazon Glacier automatically encrypts data at rest using AES 256-bit symmetric keys and supports secure transfer of customer data over secure protocols.
Connections to Amazon Glacier containing PHI must use endpoints that accept encrypted transport (HTTPS). For a list of regional endpoints

Amazon RDS for MySQL

Amazon RDS for MySQL allows customers to encrypt MySQL databases using keys that customers manage through AWS KMS. On a database instance running with Amazon RDS encryption, data stored at rest in the underlying storage is encrypted consistent with the Guidance in effect at the time of publication of this whitepaper, as are automated backups, read replicas, and snapshots. Because the Guidance might be updated, customers should continue to evaluate and determine whether Amazon RDS for MySQL encryption satisfies their compliance and regulatory requirements. For more information on encryption at rest using Amazon RDS

Amazon RDS for Oracle

Customers have several options for encrypting PHI at rest using Amazon RDS for Oracle.
Customers can encrypt Oracle databases using keys that customers manage through AWS KMS. On a database instance running with Amazon RDS encryption, data stored at rest in the underlying storage is encrypted consistent with the Guidance in effect at the time of publication of this whitepaper, as are automated backups, read replicas, and snapshots. Because the Guidance might be updated, customers should continue to evaluate and determine whether Amazon RDS for Oracle encryption satisfies their compliance and regulatory requirements. For more information on encryption at-rest using Amazon RDS, see

Elastic Load Balancing

Customers may use Elastic Load Balancing to terminate and process sessions containing PHI. Customers may choose either the Classic Load balancer or the Application Load Balancer. Because all network traffic containing PHI must be encrypted in transit end-to-end, customers have the flexibility to implement two different architectures:

Customers can terminate HTTPS, HTTP/2 over TLS (for Application) , or SSL/TLS on Elastic Load Balancing by creating a load balancer that uses an encrypted protocol for connections. This feature enables traffic encryption between the customer's load balancer and the clients that initiate HTTPS , HTTP/2 over TLS, or SSL/TLS sessions, and for connections between the load balancer and customer back-end instances. Sessions containing PHI must encrypt both front-end and back-end listeners for transport encryption. Customers should evaluate their certificates and session negotiation policies and maintain them consistent to the Guidance. For more information

Amazon EMR

Amazon EMR deploys and manages a cluster of Amazon EC2 instances into a customer's account. All Amazon EC2 instances that process, store, or transmit PHI must be Dedicated Instances. In order to meet this requirement, EMR clusters must be created in a VPC with tenancy attribute of "dedicated." This ensures that all cluster nodes (instances) launched into the VPC will run as Dedicated Instances.

Amazon DynamoDB

Connections to Amazon DynamoDB containing PHI must use endpoints that accept encrypted transport (HTTPS). For a list of regional endpoints, see
http://docs.aws.amazon.com/general/latest/gr/rande.html#ddb_region .

PHI stored in Amazon DynamoDB must be encrypted at-rest consistent with the Guidance. Amazon DynamoDB customers can use the application development framework of their choice to encrypt PHI in applications before storing the data in Amazon DynamoDB. Alternatively, a client-side library for encrypting content is available from the AWS Labs GitHub repository. Customers may evaluate this implementation for consistency with the Guidance. For more information, see https://github.com/awslabs/aws-dynamodb-encryption-java. Careful consideration should be taken when selecting primary keys and when creating indexes such that unsecured PHI is not required for queries and scans in Amazon DynamoDB.

Using AWS KMS for Encryption of PHI

Master keys in AWS KMS can be used to encrypt/decrypt data encryption keys used to encrypt PHI in customer applications or in AWS services that are integrated with AWS KMS. AWS KMS can be used in conjunction with a HIPAA account, but PHI may only be processed, stored, or transmitted in HIPAA-eligible services. KMS does not need to be a HIPAA-eligible service so long as it is used to generate and manage keys for applications running in other HIPAA-eligible services. For example, an application processing PHI in Amazon EC2 could use the GenerateDataKey API call to generate data encryption keys for encrypting

and decrypting PHI in the application. The data encryption keys would be protected by customer master keys stored in AWS KMS, creating a highly auditable key hierarchy as API calls to AWS KMS are logged in AWS CloudTrail.

Auditing, Back-Ups, and Disaster Recovery

HIPAA's Security Rule also requires in-depth auditing capabilities, data back-up procedures, and disaster recovery mechanisms. The services in AWS contain many features that help customers address these requirements.

In designing an information system that is consistent with HIPAA and HITECH requirements, customers should put auditing capabilities in place to allow security analysts to examine detailed activity logs or reports to see who had access, IP address entry, what data was accessed, etc. This data

should be tracked, logged, and stored in a central location for extended periods of time, in case of an audit. Using Amazon EC2, customers can run activity log files and audits down to the packet layer on their virtual servers, just as they do on traditional hardware. They also can track any IP traffic that reaches their virtual server instance. A customer's administrators can back up the log files into Amazon S3 for long-term reliable storage.

Under HIPAA, covered entities must have a contingency plan to protect data in case of an emergency and must create and maintain retrievable exact copies of electronic PHI. To implement a data back-up plan on AWS, Amazon EBS offers persistent storage for Amazon EC2 virtual server instances. These volumes can be exposed as standard block devices, and they offer off-instance storage that persists independently from the life of an instance. To align with HIPAA guidelines, customers can create point-in-time snapshots of Amazon EBS volumes that automatically are stored in Amazon S3 and are replicated across multiple Availability Zones, which are distinct locations engineered to be insulated from failures in other Availability Zones. These snapshots can be accessed at any time and can protect data for long-term durability. Amazon S3 also provides a highly available solution for data storage and automated back-ups. By simply loading a file or image into Amazon S3, multiple redundant copies are automatically created and stored in separate data centers.

accessed at any time, from anywhere (based on permissions), and are stored until intentionally deleted.

Disaster recovery, the process of protecting an organization's data and IT infrastructure in times of disaster, is typically one of the more expensive HIPAA requirements to comply with. This involves maintaining highly available systems, keeping both the data and system replicated off-site, and enabling continuous access to both. AWS inherently offers a variety of disaster recovery mechanisms.

With Amazon EC2, administrators can start server instances very quickly and can use an Elastic IP address (a static IP address for the cloud computing environment) for graceful failover from one machine to another. Amazon EC2 also offers Availability Zones. Administrators can launch Amazon EC2 instances in multiple Availability Zones to create geographically diverse, fault tolerant systems that are highly resilient in the event of network failures, natural disasters, and most other probable sources of downtime. Using Amazon S3, a customer's data is replicated and automatically stored in separate data centers to provide reliable data storage designed to provide 99.99% availability.

For more information on disaster recovery, see the AWS Disaster Recovery whitepaper available at http://aws.amazon.com/disaster-recovery/.

Reduces Time. Reduces Cost. Reduces Risk

Aws allows benefit of customers to reduce their cost to become HIPAA Compliant in AWS, and it significantly reduces the time required as well by avoiding costly delays and mistakes. We understand what AWS components are not supported in a HIPAA environment, which ones are supported, and how to implement them to meet the HIPAA standards. Connectria's staff will also assist you in getting a Business Associate Agreement (BAA) signed with Amazon, and we will enter into a BAA directly with each of our customers as well.
Connectria's security controls and processes go far beyond AWS, and extend throughout our entire company to all of our employees. Each of our staff members are required to take and pass HIPAA Compliance certification, and we undergo an annual HIPAA HITECH Assessment by a qualified 3rd party assessor to ensure that Connectria and our employees continue to meet the HIPAA HITECH standards. Connectria provides access to our HIPAA Compliance Team at no additional cost in order to assist our customers with achieving their HIPAA Compliance objectives.

Many of our customers include Independent Software Vendors (ISVs) who serve the healthcare market and require HIPAA compliance. Some also wish to move their applications to a hosted Software as a Service (SaaS) model. Whether you are a Covered Entity, a Business Associate, or a technology provider to the healthcare market, Connectria can help you implement and manage a HIPAA Compliant environment in AWS.

Hipaa Compliant Website

- Information that is being transported must ALWAYS be encrypted.
- PHI is backed up and is recoverable.
- Using unique access controls the information is only accessible by Authorized personnel.
- The information is not tampered with or altered.
- Information can be permanently disposed of when no longer needed.
- Information located on a server that is secured by HIPAA security rule requirements and/or a web server company who you have a HIPAA Business Associate Agreement with.

https://nuancedmedia.com/hipaa-compliant-website-design/