Encrypting AWS RDS Instances

⚠ Please note - this is a work in progress

To encrypt an AWS RDS instance

http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Overview.Encryption.html

Q: Can I encrypt data at rest on my Amazon RDS databases?
Amazon RDS supports encryption at rest for all database engines, using keys you manage using AWS Key Management Service (KMS). On a database instance running with Amazon RDS encryption, data stored at rest in the underlying storage is encrypted, as are its automated backups, read replicas, and snapshots. Encryption and decryption are handled transparently. For more information about the use of KMS with Amazon RDS, see the Amazon RDS User's Guide.

You can also add encryption to a previously unencrypted DB instance or DB cluster by creating a DB snapshot and then creating a copy of that snapshot and specifying a KMS encryption key. You can then restore an encrypted DB instance or DB cluster from the encrypted snapshot.
Amazon RDS for Oracle and SQL Server support those engines' Transparent Data Encryption technologies. Transparent Data Encryption in Oracle is integrated with AWS CloudHSM, which allows you to securely generate, store, and manage your cryptographic keys in single-tenant Hardware Security Module (HSM) appliances within the AWS cloud. For more information, see the Amazon RDS User's Guide sections on Oracle and SQL Server.

AWS KMS
You can enable this feature and start to use customer-managed keys for your RDS database instances running MySQL or PostgreSQL with a couple of clicks when you create a new database instance. Turn on Enable Encryption and choose the default (AWS-managed) key or create your own using KMS and select it from the dropdown menu. You can use the ARN of a key from another account to encrypt an RDS DB instance.

If you want full control over a key, then you must create a customer-managed key. You cannot delete, revoke, or rotate default keys provisioned by AWS KMS. You can view audit logs of every action taken with a customer-managed key by using AWS CloudTrail.

CloudHSM
AWS CloudHSM is a service that helps you to meet stringent compliance requirements for cryptographic operations and storage of encryption keys by using single tenant Hardware Security Module (HSM) appliances within the AWS cloud. CloudHSM is now integrated with Amazon RDS for Oracle Database.

Oracle TDE

Customers can also leverage Oracle Transparent Data Encryption (TDE). Oracle TDE is a feature of the Oracle Advanced Security option available in Oracle Enterprise Edition. This feature automatically encrypts data before it is written to storage and automatically decrypts data when the data is read from storage. Customers can also use AWS CloudHSM to store Amazon RDS Oracle TDE keys.

Connections to Amazon RDS for Oracle containing PHI can use transport encryption. This is accomplished using Oracle Native Network Encryption and enabled in Amazon RDS for Oracle option groups.

Regions
All logs, backups, and snapshots are encrypted for an Amazon RDS encrypted instance. A Read Replica of an Amazon RDS encrypted instance is also encrypted using the same key as the master instance when both are in the same region. If the master and Read Replica are in different regions, you encrypt using the encryption key for that region.

NOTE
Once you have created an encrypted DB instance, you cannot change the encryption key for that instance, Therefore, be sure to determine your encryption key requirements before you create your encrypted DB instance.

Oracle SSL
http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Appendix.Oracle.Options.SSL.html

Working Environment

in addition to the AWS RDS instance please keep in mind the following :

- Laws.
- Policy.
- 9 Points.
- Encryption at rest.
- Encryption in transit.
- Dedicated device.
- 2 factor for dedicated device.
- Auditing.
- Logging.
- Limited access.
- Patching.