Minfy-Migration-Use Case

Minfy-Migration-Use Case

Contents

Minfy-Migration-Use Case

Scope

This document provides a detailed use case study on migration using lift and shift of applications from on premise to AWS and re-engineering of AWS infrastructure by following AWS best practices with guidelines for implementation.

About Customer

Customer is one of the leading group of companies in the country which includes multiple healthcare businesses. 20-25 years back there was no comprehensive treatment facilities in the country. It was very difficult to have the diagnosis for a distressed patient from a remote area in Bangladesh. For a minor investigation a patient from a remote area had to stay Dhaka for several days and wander at different places for different tests.

Requirement

- Lab Management ERP to be implemented
- Set up infrastructure on AWS cloud, preferably in the Singapore Region
- All Customer units to securely connect to the Cloud Infrastructure
- Application centralized to the server to run primarily in online mode. Except for the on-premise Lab Hardware Integration Software there would be no offline/online interactions
- Patients should be able to connect through the online portal to view reports
- All Application functions should be accessible only through browser

Solution Implementation

- Servers were launched in the AWS Singapore Region. All access are controlled via Identity and Access Management (IAM) Console of AWS. This also allows enabling Multi Factor Authentication for secure access
- A secure cluster of servers were launched into a private subnet under a Virtual Private Cloud (VPC). VPC is like a cloud in a cloud and is not accessible to the internet world
- Servers were launched in both the availability Zones [2] (Data centers in a Region) which serves the purpose of both high availability and disaster management
- All Servers were launched in a Public or Private Subnet. Subnets are used to virtually segregate the servers into various groups. Each subnet is launched under its own subnet range for additional security. Network Access Security List (NACL) is a stateless firewall that provides both inbound and out bound access restriction features
- Servers were launched in an AutoScale unit. Autoscale is an advanced feature of AWS that guarantees that servers scale horizontally in case there is an increase in load
- Each server (or server group) was launched in a security group. Security groups are stateful firewall where in we can control the inbound IP and Port access
- An Elastic Load Balancer (ELB) was configured that connected to the web application servers in both the regions. Load balancer automatically removes unhealthy instances and requests the Auto Scale unit to provide a new pre-configured server with the application installed to it
- Web Application Server connects to the HIMS and PACS server internally. Both HIMS and PACS server are created in its own AutoScale unit which is mapped by an Internal Load Balancer
- All applications are connected to the Oracle production database as per client requirement
- Oracle SE2 RDS database is configured in a master slave mode for high availability. This ensures that in case of Availability Zone failure there is no impact on the uptime of the application
- Since License is procured by the client, Oracle was provisioned in "Bring your own license mode". License should be purchased as specified by the application vendor separately. Oracle cloud license T&C can be read here: http://www.oracle.com/us/corporate/pricing/cloud-licensing-070579.pdf
- Servers in the VPC did not have public internet connection. For the servers to access the public internet a NAT Gateway was provisioned. NAT behaves as a proxy for the servers inside the VPC
- Since servers are behind a VPC there is no way for the admin/developers to reach them. Therefore a software VPN, OpenVPN was installed on the VPC. Users first login to this secure OpenVPN server to ping the servers inside the VPC. This also allows secure and restricted access to the client data on the server. This is only used by selected Admins and Developers who need to access the infrastructure from outside Customer premises

Minfy-Migration-Use Case

- For all users within Customer premises, a Site-to-Site VPN was configured. This needs a compatible hardware firewall to be present at all locations for connectivity. Users access the application using this secure VPN tunnel only
- Cookie cutter instance has the master image of the core application. This master image is used to launch servers in the AutoScaled units
- Three important access logs are configured,
  - ✓ VPC Flow Logs : To understand the source and destination IPs of the traffic
  - ✓ CloudTrail Logs : To understand all API usage on AWS
  - ✓ Cloudwatch Alarms : Automated Cloudwatch alarms configured at all levels including Server and Infrastructure Monitoring and Traffic Monitoring through WAF
- Test and Stage infrastructure were created in the same region but in separate VPC. These servers can be accessed only via the Open VPN Server. Separate Load Balancer is provisioned for the public access points of Test and Staging servers.
- S3FS is configured to mount an S3 bucket to all the servers. This serves as a common drive to store and retrieve files. Files stored in these mounts are stored directly on AWS S3

PACS Image Management and Processing

PACS images are stored locally at the customer premises and then uploaded to the EC2 server using RSYNC or any other standard mechanism. This process also requires heavy RAM usage thus increasing the server cost. Lack of RAM can also impair uploading files in parallel.

Therefore, it’s recommended to move away from this practice and work on a more robust, fault tolerant and economical storage cum dissemination process as proposed below.

AWS offers a robust image management and processing capabilities. As per this images will get uploaded directly to S3 over https.

Security Best Practices

Document Your AWS Processes and Procedures, then Update Them Regularly

Imagine you have a very specific file structure set up in the cloud, with categorical folders that are protected by different levels of permission. You know that all company sales data should go in a specific folder, but a coworker, though meaning well, *doesn’t know* and decides to transfer sales data to a different, unprotected folder. Chaos ensues.

To avoid this type of confusion, create consistent cloud practices everyone can follow. Document your AWS processes and procedures. Store them in a common space that the organization can access, like a shared drive on the internal network. And update the document every time something changes in your cloud approach to help coworkers, stakeholders, third party vendors, and trading partners remain on the same page.

Use AWS CloudTrail to Track Your AWS Usage

Understanding what actions users take in the cloud is an important step toward keeping your data secure and in the hands of those you trust. Use an AWS service like Amazon CloudTrail to anticipate and 0auditing of your AWS account.

AWS CloudTrail can do the following tasks, and more:

- Create API call history logs
- Record when objects or data are created, read, or modified
- Calculate and give you risk reports on your cloud storage account
- Determine who makes changes to your cloud storage infrastructure
- Track who logs in to your accounts (including successful and failed login attempts**)**

Complete Risk Assessments Often

Even though the cloud is run by Amazon Web Services, both AWS and your organization are responsible for making sure nothing falls through the cracks. This includes maintaining “adequate governance over the entire IT control environment regardless of how IT is deployed" and having "an understanding of required compliance objectives and requirements," among other things.

AWS completes and publishes risk assessments for their services, and you should do the same for the data you've stored in the cloud. Each time you give a new key player (including third party vendors and trading partners) access to your AWS cloud storage, walk through the following steps:

- Review the risks you currently know about and ensure they're still being addressed
- Identify and add new risk scenarios to your list. Plan for how to tackle them
- Identify the key players who have access to AWS and ensure they're following standard security hygiene
- Assess your AWS account. Make sure your settings, policies, and security are still relevant
- Consider the steps you should take next to manage your data and prevent future risk

Remember, risk assessment is an ongoing process that allows you to find and address security concerns in your infrastructure. Since storing data in the cloud takes away some of your control over sensitive company information by not being on-premises, it's vital you complete assessments often to keep on top of potential security gaps and vulnerabilities.

Follow Standard Security Hygiene for Host and Guest Systems
Practicing standard security hygiene is one of the easiest ways to keep your data protected. These habits should become second nature, just like washing your hands or brushing your teeth, and benefit you immensely without requiring much time or resources.

Enable multi-factor authentication for all accounts
Amazon Web Service's MFA requires a user to provide two pieces of information to prove they're authentic. The first piece is knowledge (something you know, your login credentials), the second is possession (something you have, an authentication code sent to an AWS MFA enabled device). Just enable multi-factor authentication for your AWS accounts to get an immediate boost in security.

Remove privileges from defunct accounts
When an employee, trading partner, or third party vendor leaves the relationship, clean out their account and delete any privileges they were given. This removes the temptation for a renegade player—or a hacker guessing at passwords and emails—to return at a later date and compromise sensitive company information.

Disable password-only access for guests
Even guest accounts should use multi-factor authentication wherever possible, even if they have limited authorities and privileges.

Manage and Review AWS Accounts, Users, Groups, and Roles
Every so often, we recommend you review your AWS accounts, users, groups, and roles to gain a proper overview of the privileges and permissions they have. Are any of these stagnant or similar to other setups? Consider combining them. Are any of them no longer necessary? Limit the clutter. The less overlap there is, the better.
Administrators of Amazon Web Services accounts should pay special attention to the permissions listed for their S3 buckets. Several different types of access can be given to users, including list, upload, delete, view, and edit. A bucket can also be set to viewable for AWS account holders or anonymous users, which may cause high risk depending on the files in the bucket, so make sure to review your S3 buckets and permissions to avoid potential security pitfalls.

The bottom line? Provide your accounts, users, groups, and roles with the least amount of privileges they need to function. If someone needs temporary access, it's better to add them in as they're required and remove them right after to avoid information falling into the wrong hands.

Protect Your Access and Encryption Keys
If you're using AWS to store your data in the cloud, you're bound to have access keys and encryption keys. Access keys help AWS verify your identity against your login attempt and give you *access* to the resources you've been given. Users with different access keys may not be able to see the same things you do, so it's imperative you keep your keys safe.

Similarly, encryption keys are used to encrypt and decrypt data. Since they unlock sensitive information, keep them separate from your data. This best practice is especially important for companies who need to comply with regulations like HIPAA, FISMA, and PCI DSS. "Essentially, the compliance requirements all say the same thing," writes Luke Probasco for Pantheon, "encryption keys should never reside in the same environment or server as the encrypted data. This is a technical way of saying, don't leave your key under the doormat a hacker walks in over."

Here are just a few ways to keep your access and encryption keys safe:

- Periodically delete any unused keys
- Use temporary access keys instead of permanent ones wherever possible. This way, if an attacker compromises an account or discovers a user's credentials, their access time-sensitive
- Watch the encryption key life cycle and make sure new ones are properly saved and secured
- Create procedures for worst case scenarios in the event a key is lost or tampered with

An easy way to protect your keys is to use AWS Key Management Services, the service Amazon offers that "makes it easy for you to create and control the encryption keys used to encrypt your data." AWS KMS even integrates with AWS CloudTrail, Amazon's log auditing service, so you can view logs of your key usage.

Secure Your Data at Rest and in Transit

When moving data between your network and the cloud, *always* encrypt your files and protect your communication using SFTP, FTPS, or SCP. Furthermore, keep them encrypted even when they're at rest, sitting in an AWS S3 bucket or on a server. You can choose to encrypt single files or entire folders depending on your needs.

A secure cloud file transfer solution can encrypt your files both ways using modern encryption methods. Good cloud file transfer or managed file transfer (MFT) software help you stay up-to-date as encryption standards change over time, while also making sure your data transfers are easy to manage and audit.

Some solutions, like GoAnywhere MFT, are available in the AWS Cloud Marketplace, making it quick and easy to implement these practices in your environment.