Minfy-SREI Migration Use Case

Minfy-SREI Migration Use Case

Document Details

Use Case Name	Minfy Migration - Use Case02
First Draft	15 Jan 2018
Author	Atanu Sarkar
Reviewed By	Pradeep Narayanaswamy

Minfy-SREI Migration Use Case

Contents

1. Scope 4

2. About Customer 4

3. Use Case Description 4

4. Primary & Supporting Actors 5

5. Pre & Post Conditions 5

6. Trigger/Challenge 6

7. Application Architecture on AWS 7

8. Implementation 9

9. Security with Migration 10

10. Benefits 11

Minfy-SREI 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

SREI Infrastructure Finance Limited is a leading infrastructure financing conglomerate in India. SREI, derived from the Hindi word 'Shrey' meaning 'merit' or 'credit', has completed a quarter of a century. In these 25 years, SREI has come a long way after starting perhaps in one of the most difficult and challenging times. In 1989, India had not opened up its economy, which during those times, was in doldrums. There was hardly any quality investment in infrastructure projects. SREI was born then, a child of a dream, with the aim to address two of the most critical problems in our country then – financing and infrastructure

Use Case Description

It has been decided that server and application migration will be conducted using an Automated and Manual Migration process. While Automation will be carried forward using a 3rd Party tool Cloud Endure a team of Infrastructure professionals from Techminfy and Development professionals from SREI will jointly migrate the Applications.

CloudEndure will primarily be used to conduct block level sync for databases and complex and legacy system Applications. For file system integration Techminfy's proprietary application **CloudStorz** will be used.
As per the use case S3 bucket needs to be mapped as drive for the core application. To map the drive on Windows a 3rd party application TNTDrive will be used.

To lift and shift the environment from on-premise to AWS, we have followed below activities,

- Segregate the current infrastructure by Application stacks
- Identify common infrastructure stack shared by various applications, like databases
- Review provisioned and used Core and RAM on existing stack and provision servers on the cloud only as
- per used infrastructure
- Work towards application consolidation to reduce the number of servers (Optional)
- Create infrastructure diagram
- Set up VPN Set up between AWS and current NOC (TechMinfy) for management
- Set Up AWS DirectConnect between AWS and current data center
- Plan AWS Infrastructure plan segregated with VPC and IP Level connectivity
- Plan file system sync activities
- Creation of Infrastructure on AWS
- Configuration of TNT Drive for file sharing between on premise and cloud application
- Configuration of CloudEndure on Database and core application stacks as identified
- Customization and configuration of CloudStorz for File Sync activity and continuous integration
- Start of migration of applications and sync of file system
- Setting up of DNS failover strategy

Minfy-SREI Migration Use Case

Primary & Supporting Actors

- ***Actors from MinfyTech: Atanu***
- ***Actors from Client: Rakesh***

Pre & Post Conditions

Pre-implementation State

At a high level Migration phase will be divided into three phases:
Phase 1: Consisting of core Applications
Phase 2: Consisting of secondary and supporting application stacks
Phase 3: Applications that need to run directly on Amazon AWS
Following applications have been identified to move in each phase:

Application Name Core Stack Type of DR

- Ambit 42 SEFL Windows and Linux Active/Passive
- Ambit 42 SIFL Windows and Linux Active/Passive
- NewGen Windows and RHEL Active/Passive
- Ebiz Linux Active/Passive
- Kastle –SEFL Windows and Linux Active/Passive
- Kastle -SIFL Windows and Linux Active/Passive

All databases will run in an Active/Standby mode with Live replication

Application Name Core Stack Type of DR

- Elink Windows/RHEL Active/Passive
- Informatica IIR Windows/Linux Active/Passive
- SARV Integration Utility / SARV
- Applications
- Windows/Linux Active/Passive
- LOSREPORT Windows Active/Passive
- Variable Pay Linux/RHEL Active/Passive
- SRTS and DTS Windows/Linux Active/Passive
- SMTP – Proxy RHEL Active/Passive
- Radius Server RHEL Active/Passive
- DHCP RHEL Active/Passive
- Oracle Grid Manager RHEL Active/Passive

All databases will run in an Active/Standby mode with Live replication
Application Name Core Stack Type of DR

- itrack - Ticket Management Windows HA on AWS
- CTS Windows HA on AWS
- Legatrix Windows HA on AWS
- SAIBA Windows HA on AWS
- Collateral Management Windows HA on AWS
- SREI Café Windows HA on AWS
- BIS Windows HA on AWS
- Axapta Windows HA on AWS
- Customer App Windows HA on AWS

Minfy-SREI Migration Use Case

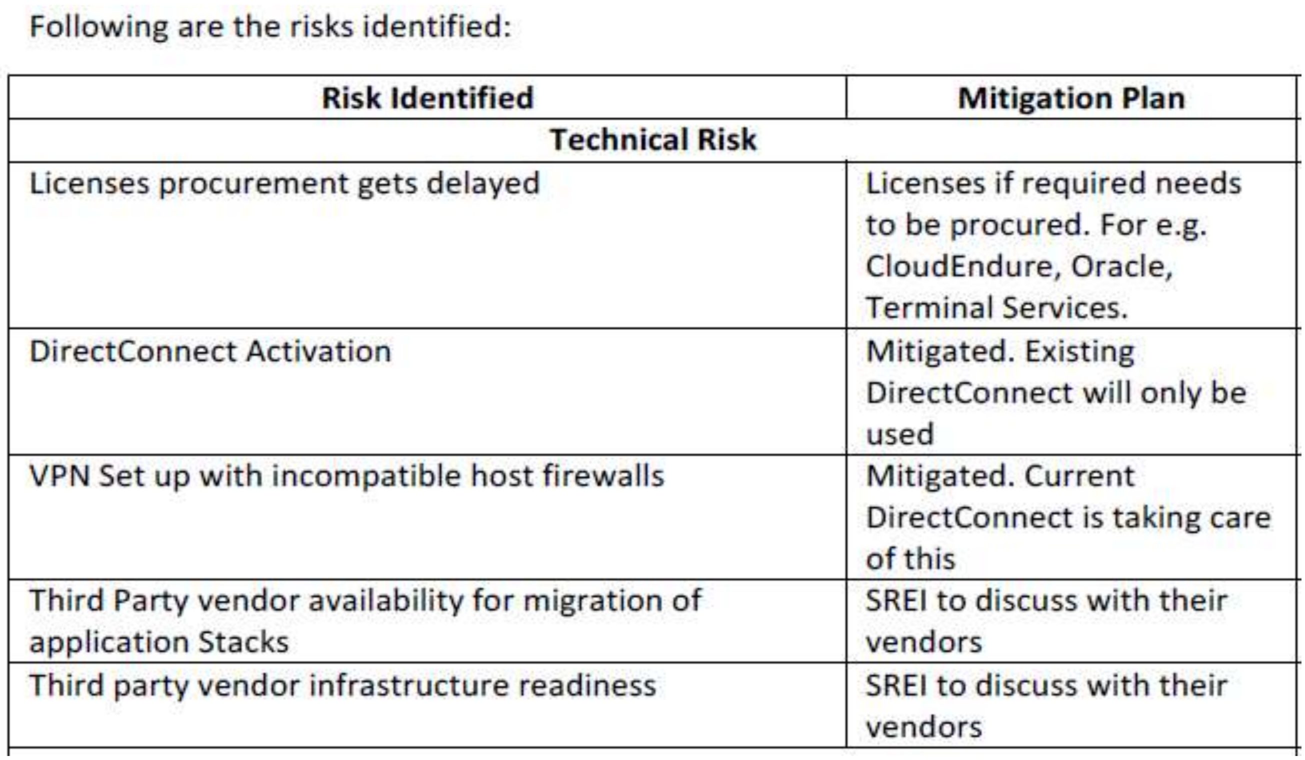
- Password Express Windows HA on AWS
- Inflow Inventory Windows HA on AWS
- Ace Windows HA on AWS
- SVN RHEL HA on AWS
- Kushal Khoj Windows HA on AWS
- Swasth SREI Windows HA on AWS
- Asset Image Windows HA on AWS

It's recommended to consolidate servers running above applications to reduce cost and increase efficiency in maintenance.

Post-implementation State

Performance has enhanced post migration by choosing required instance type. Databases moved to RDS for high performance. All websites moved to EC2 Instance and DNS pointed to new AWS server. New licence procured for migrated cPanel.

Trigger/Challenge



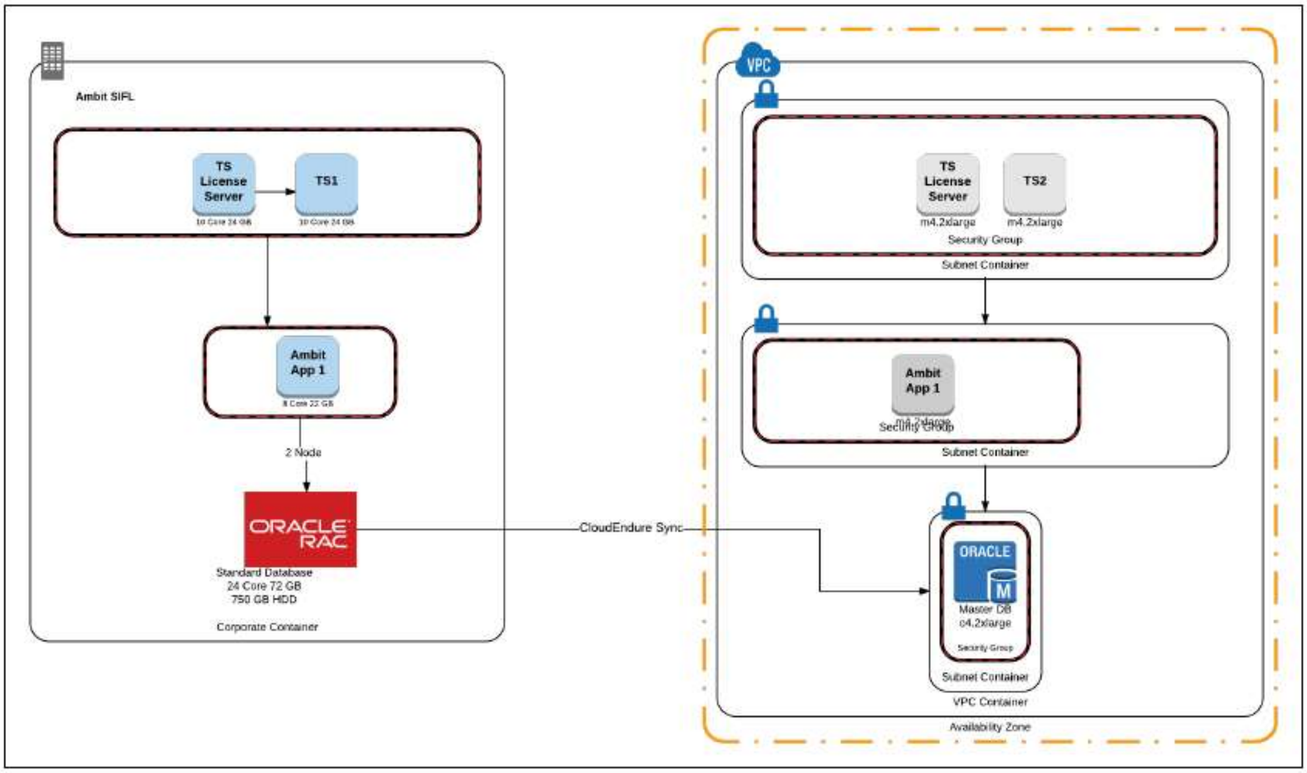
Following are the risks identified:

Risk Identified	Mitigation Plan
Technical Risk	
Licenses procurement gets delayed	Licenses if required needs to be procured. For e.g. CloudEndure, Oracle, Terminal Services.
DirectConnect Activation	Mitigated. Existing DirectConnect will only be used
VPN Set up with incompatible host firewalls	Mitigated. Current DirectConnect is taking care of this
Third Party vendor availability for migration of application Stacks	SREI to discuss with their vendors
Third party vendor infrastructure readiness	SREI to discuss with their vendors

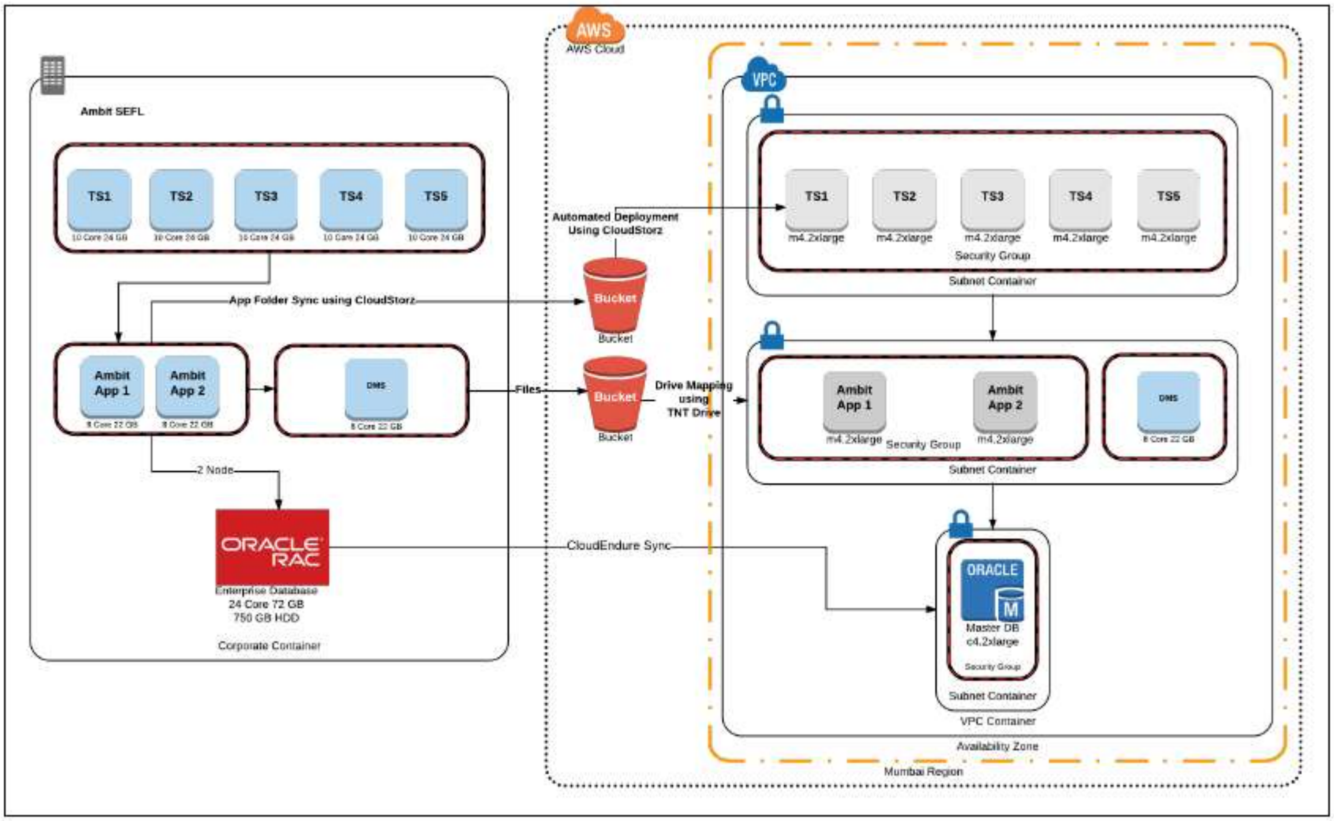
Minfy-SREI Migration Use Case

Application Architecture on AWS

6.1 Ambit SIFL and SEFL

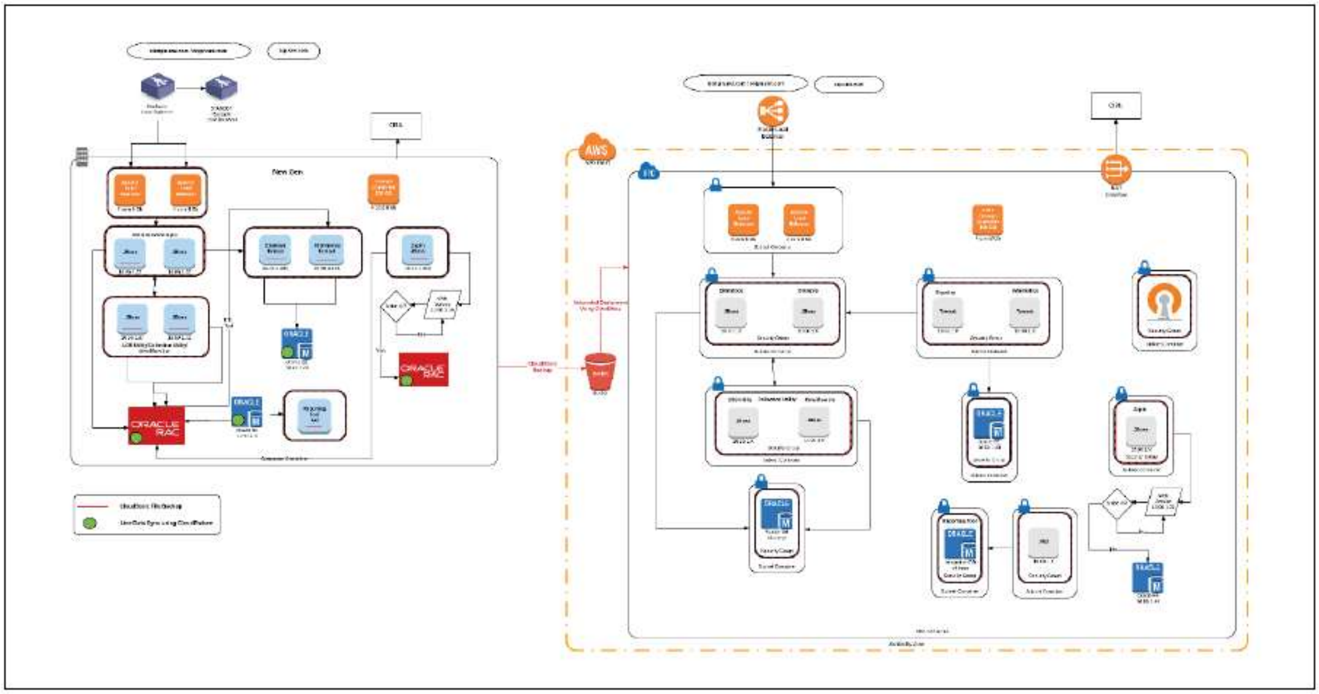


Minfy-SREI Migration Use Case



Minfy-SREI Migration Use Case

New Gen



Implementation

- **Preliminary preparations –** DC end sanity check has been done. During this process entire on premise environment was identified that include application stack, file level architecture and current web services running on DC end.
- **Proposed architecture –** Based on preliminary preparation, we have proposed the architecture based on the AWS environment.
- **Access to AWS Console - IAM User –** After approval of proposed architecture, AWS credential was received will all necessary IAM user and permissions.
- **Provisioning of AWS infrastructure components -** Based on proposed architecture, AWS infrastructure was provisioned.
- **Migrating infrastructure components and websites –** Migration was completed using cPanel migration tool which is suggested by the Customer.
- **Review migrated components –** Once websites are migrated sanity check was done.
- **Cut over to AWS –** After approval of sanity check, final cut over done from on premise to AWS.
- **DNS routing to new AWS server –** DNS routing was done to new AWS Server.

Minfy-SREI Migration Use Case

Security with Migration

Tighten CloudTrail security configurations

- Enable CloudTrail across all geographic regions and AWS services to prevent activity monitoring gaps.
- Turn on CloudTrail log file validation so that any changes made to the log file itself after it has been delivered to the S3 bucket is trackable to ensure log file integrity.
- Enable access logging for CloudTrail S3 bucket so that you can track access requests and identify potentially unauthorized or unwarranted access attempts.
- Turn on multifactor authentication (MFA) to delete CloudTrail S3 buckets, and encrypt all CloudTrail log files in flight and at rest.

Followed Identity and Access Management (IAM) best practices: -

- When creating IAM policies, ensure that they're attached to groups or roles rather than individual users to minimize the risk of an individual user getting excessive and unnecessary permissions or privileges by accident.
- Provision access to a resource using IAM roles instead of providing an individual set of credentials for access to ensure that misplaced or compromised credentials don't lead to unauthorized access to the resource.
- Ensure IAM users are given minimal access privileges to AWS resources that still allows them to fulfil their job responsibilities.
- As a last line of defence against a compromised account, ensure all IAM users have multifactor authentication activated for their individual accounts, and limit the number of IAM users with administrative privileges.
- Rotate IAM access keys regularly and standardize on a selected number of days for password expiration to ensure that data cannot be accessed with a potential lost or stolen key.
- Enforce a strong password policy requiring minimum of 14 characters containing at least one number, one upper case letter, and one symbol. Apply a password reset policy that prevents users from using a password they may have used in their last 24 password resets.

Follow security best practices when using AWS database and data storage services

- Ensure that no S3 Buckets are publicly readable/writeable unless required by the business.
- Encrypt data stored in EBS as an added layer of security.
- Encrypt Amazon RDS as an added layer of security.
- Restrict access to RDS instances to decrease the risk of malicious activities such as brute force attacks, SQL injections, or DoS attacks.

Custom applications security best practices: -

- Inventory and categorize all existing custom applications deployed in AWS
- Involve IT security teams throughout the application development lifecycle
- Grant the fewest privileges possible for application users
- Enforce a single set of data loss prevention policies

Minfy-SREI Migration Use Case

Benefits

Cost Savings: A Cloud Hosted Desktop provides you with scalable computing power, while minimizing IT requirements and physical data storage, providing you with significant savings.

Security: Perhaps the weakest link in the initial days of cloud adoption was security concerns. But today, more people have begun to realize these concerns are misguided. Cloud IT service providers actually provide higher levels of security and data integrity. Why? Because they make huge investments in the resources and technology, along with a skilled team of IT experts and engineers smaller businesses just can't afford to do on their own.

Connectivity & Accessibility: Keep users connected no matter where they work with anytime, anywhere access. Users may access files anytime, anywhere, using any device. That means no more risk of files being stored on any computer.

Reduced Risk of Data Loss: Even more security for users by backing-up data offsite – decreasing the potential for hackers, viruses, ransomware, and other cybersecurity problems. Let's repeat that again, more security. (Read: 3 Tips to Prevent Data Loss).
Faster Deployment: Cloud-based services can be deployed within just an hour or a few days rather than the weeks, months or years it can take to strategically plan, buy, build and implement an internal IT infrastructure with internal personnel.

Increased Collaboration: Cloud computing enables employees situated in various locations to collaborate easily. By providing simultaneous syncing, working and sharing documents and records in real time, cloud computing helps increase the collaboration and efficiency of employees.

Improved Efficiency: After migrating to the cloud, you no longer need to worry about power requirements, space considerations, expensive computer hardware, or software updates. You get to keep your entire company focused on generating revenue and relationships, not on IT.