TD SYNNEX | Services

Migration to AWS

Technical Overview

Document Version History

Version number	Purpose / Change	Author	Date
1.0	First Published Version	Harsh Kumar/Sergio Imperial	23/05/2024
2.0			

Contents

1 Introduction

This document aims to describe the technical steps that will be performed to migrate clients' infrastructures into the Amazon Web Services (AWS) cloud.

The migration process will start with a Migration Design phase. This phase includes initial discovery and assessment of the environment. Firstly we will gather initial information about the client's environment in order to confirm the scope of the migration process and validate that the information provided so far is accurate.

The initial discovery and assessment is done to better understand client's environment, including internal and external dependencies of Virtual Machines, Services, Applications and Databases. Based on the information collected and our evaluation, we will present a detailed Migration design report.

For this discovery & assessment, we will use AWS Application Discovery Service. AWS Application Discovery Service helps in planning the migration to the AWS cloud by collecting usage and configuration data about the customer's on-premises servers. Application Discovery Service is integrated with AWS Migration Hub and AWS Database Migration Service Fleet Advisor. It gives us the capability to view the discovered servers, group them into applications, and then track the migration status of each application from the Migration Hub console in the customer's home Region.

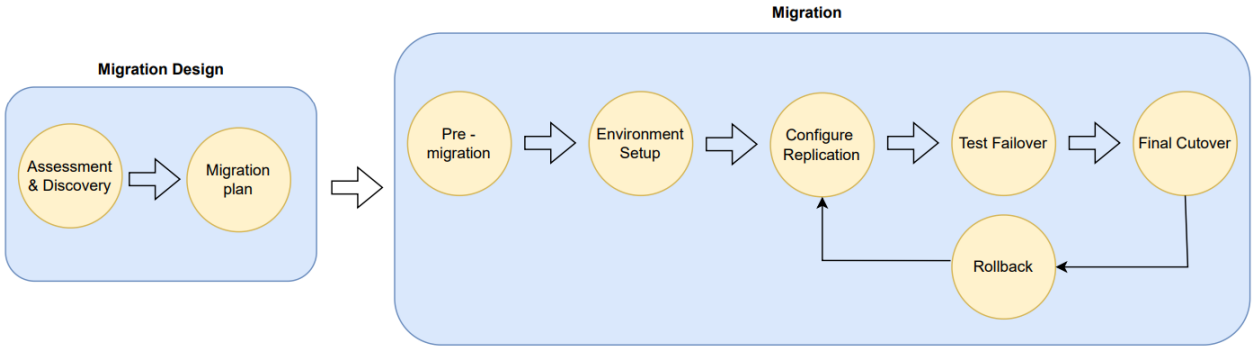
In the end of the Migration Plan phase, TD SYNNEX will deliver to the client a Migration Design document which includes a detailed migration plan. After acceptance of that document by the client, we will move to pre-migration phase, in which we will set up the initial cloud infrastructure, making it ready for the migration to be carried out.

Finally, we will start the actual migration of the environment. Usually, machines will be migrated to AWS using one of the following approaches:

- Re-host (Lift & Shift) migration – Server will be migrated to AWS as-is. For e.g. Application servers or Web servers that the customer would like to migrate as it is to AWS.
- Re-build or Workload Migration – Where we will be spinning up a new Server on AWS and manually transfer the data or roles. Such as DC server where the FSMO roles will be transferred manually to the new AWS server.

The best approach to migrate each of the client's workloads will be defined in Migration Design report.

This document will describe all the migration steps and provide examples of migration of specific use cases with different levels of complexity.



2 Migration Design

2.1 Assessment & Discovery

After the project onboarding and scope confirmation, to help us planning the migration, we will go through a discovery and validation process, gathering hardware and software information, to validate the machines on scope for migration.

In this phase we will need the following details to be provided by the client:

- VPN details: VPN or any other method of remote access to client's environment.
- Domain Administrator credentials: Credentials of a domain administrator to be able to access to the servers to migrate. Both the remote access to the client's environment and the domain administrator credentials are essential to allow our team to analyse the environment to migrate in without the need of involvement from the client.
- If the servers are not in domain, then local Admin credentials for the servers will be required.
- SQL Server credentials: In case any SQL Server instance are on-scope for migration.
- VMWare vCenter credentials: In case the virtualization architecture is VMWare vCenter, we will need vCenter credentials with read permissions to run the discovery and migration tools
- Credentials for any relevant application: We will need credentials for any relevant application to migrate as, for instance, Microsoft Dynamics or Microsoft SharePoint.
- On-Premise Network Architecture Diagrams: Any relevant network or application diagrams from the current on-premise state is quite important to help our team in the migration design
- On-Premise Firewall Make & Model: To plan network integration between AWS and the on-premise network, we will require details about the on-premise network device
- Preferred AWS Migration Hub home region.
- Access to Amazon S3 in the home region is required for auto-upgrade to function.
- For discovery, we need an AWS Identity and Access Management (IAM) user in the console with existing AWSApplicationDiscoveryAgentAccess IAM managed policy attached. This policy allows the user to perform necessary agent actions.
- Proposed time to access environment or run any tool: If there is any time constrains in which the client does not want our team to access the on-premise environment, we will need to be notified

To perform the Assessment & discovery process we will use **AWS Application Discovery Service**.

AWS Application Discovery Service helps to plan the migration to the AWS cloud by collecting usage and configuration data about the on-premises servers. Application Discovery Service is integrated with AWS Migration Hub and AWS Database Migration Service Fleet Advisor.

Migration Hub simplifies the migration tracking as it aggregates the migration status information into a single console. It allows to view the discovered servers, group them into applications, and then track the migration status of each application from the Migration Hub console in the home Region. We can also use DMS Fleet Advisor to assess migrations options for database workloads.

Application Discovery Service offers two ways of performing discovery and collecting data about the on-premises servers:

- **Agentless discovery** can be performed by deploying the Application Discovery Service Agentless Collector (Agentless Collector) (OVA file) through the **VMware vCenter**.
    - After Agentless Collector is configured, it identifies virtual machines (VMs) and hosts associated with vCenter.
    - Agentless Collector collects the following static configuration data:
        - Server hostnames
        - IP addresses
        - MAC addresses
        - Disk resource allocations
        - Database engine versions, and database schemas
        - Additionally, it collects the utilization data for each VM and database providing the average and peak utilization for metrics such as CPU, RAM, and Disk I/O.
- **Agent-based discovery** can be performed by deploying the AWS Application Discovery Agent on each of the **VMs and physical servers**. The agent installer is available for Windows and Linux operating systems. It collects static configuration data, detailed time-series system-performance information, inbound and outbound network connections, and processes that are running.

This table provides a features comparison of both methods:

	Agentless Collector	Discovery Agent
Supported server types		
VMware virtual machine	Yes	Yes
Physical server	No	Yes
Deployment		
Per server	No	Yes
Per vCenter	Yes	No
Collected data		
Static server configuration data	Yes	Yes
Database configuration data	Yes	No
VM utilization metrics	Yes	No
Database utilization metrics	Yes	No
Time series performance information	No	Yes (Export only)
Network inbound/outbound connections	No	Yes (Export only)
Running processes	No	Yes (Export only)
Supported OS	Any OS running in VMware vCenter V5.5+	
Supported databases	Oracle, SQL Server, MySQL, and PostgreSQL	None

Our recommendation regarding the method to be used during the discovery & assessment phase is described in the table below:

Source Environment	Preferred Discovery Approach	Secondary Discovery Approach
VMWare	Agentless	Agent Based
Physical	Agent Based	Agent Based
Hyper-V	Agent Based	Agent Based

2.1.1 Prerequisites for Discovery Agent (Agent based assessment)

The following are the prerequisites and the tasks that need to be performed before successfully install the AWS Application Discovery Agent (Discovery Agent).

- AWS Migration Hub home region needs to be set before installing Discovery Agent.
- If there is a 1.x version of the agent installed already, then it must be removed before installing the latest version.
- If the host that the agent is being installed on runs Linux, then verify that the host at least supports the Intel i686 CPU architecture (also known as the P6 micro architecture).
- Verify that the operating system (OS) environment is supported:
    - Linux
        - Amazon Linux 2012.03, 2015.03
        - Amazon Linux 2 (9/25/2018 update and later)
        - Ubuntu 12.04, 14.04, 16.04, 18.04, 20.04
        - Red Hat Enterprise Linux 5.11, 6.10, 7.3, 7.7, 8.1
        - CentOS 5.11, 6.9, 7.3
        - SUSE 11 SP4, 12 SP5
    - Windows
        - Windows Server 2003 R2 SP2
        - Windows Server 2008 R1 SP2, 2008 R2 SP1
        - Windows Server 2012 R1, 2012 R2
        - Windows Server 2016
        - Windows Server 2019
        - Windows Server 2022
- If outbound connections from the customer's network are restricted, then the firewall settings needs to be updated. Agents require access to arsenal over TCP port 443. They don't require any inbound

ports to be open. For example, if the home region is eu-central-1, then we use https://arsenal-discovery.eu-central-1.amazonaws.com:443
- Access to Amazon S3 in the home region is required for auto-upgrade to function.
- For discovery, we need an AWS Identity and Access Management (IAM) user in the console with existing AWSApplicationDiscoveryAgentAccess IAM managed policy attached. This policy allows the user to perform necessary agent actions.
- Time skew from the Network Time Protocol (NTP) servers needs to be checked and corrected if necessary. Incorrect time synchronization causes the agent registration call to fail.

2.1.2 Prerequisites for Agentless Collector

The following are the prerequisites for using Application Discovery Service Agentless Collector (Agentless Collector):

- One or more AWS accounts.
- An AWS account with the AWS Migration Hub home Region set. The Migration Hub data is stored in the home Region for purposes of discovery, planning, and migration tracking.
- An AWS account IAM user that is set up to use the AWS managed policy AWSApplicationDiscoveryAgentlessCollectorAccess. To use the database and analytics data collection module, this IAM user must also use two customer managed IAM policies DMSCollectorPolicy and FleetAdvisorS3Policy. The IAM user must be created in an AWS account with Migration Hub home Region set.
- VMware vCenter Server V5.5, V6, V6.5, 6.7 or 7.0.
    - Note - The Agentless Collector supports all of these versions of VMware, but AWS is currently testing against version 6.7 and 7.0.
- For VMware vCenter Server setup, customer must provide vCenter credentials with Read and View permissions set for the System group.
- Agentless Collector requires outbound access over TCP port 443 to several AWS domains. For a list of these domains, see https://docs.aws.amazon.com/application-discovery/latest/userguide/agentless-collector-gs-prerequisites.html#agentless-collector-gs-prerequisites-firewall
- To use the database and analytics data collection module, Amazon S3 bucket in the AWS Region needs to be created that can be set as Migration Hub home Region. The database and analytics data collection modules stores inventory metadata in this Amazon S3 bucket.
- To set up the Application Discovery Service Agentless Collector (Agentless Collector), we must download and deploy the Agentless Collector Open Virtualization Archive (OVA) file. The Agentless Collector is a virtual appliance that gets installed in the on-premises VMware environment.

If the customer has virtual machines (VMs) that are running in the VMware vCenter environment, we can use the Agentless Collector to collect system information without having to install an agent on each VM. We will have to load this on-premises appliance into vCenter and allow it to discover all of its hosts and VMs.

Agentless Collector captures system performance information and resource utilization for each VM running in the vCenter, regardless of what operating system is in use.

The main Agentless collector limitation is that it cannot “look inside” each of the VMs, and as such, cannot figure out what processes are running on each VM nor what network connections exist. To overcome this limitation, for VMs hosted on VMware, we can use both the Agentless Collector and Discovery Agent to perform discovery simultaneously.

2.2 Migration Plan

After the discovery of the environment, our team will go through a deeper analysis of the client’s environment, understanding internal and external dependencies of Virtual Machines, Services, Applications and Databases. This work will be done in collaboration with the client to better understand their requirements and expectations.

When designing the migration and defining a migration plan, we will need to analyse things such as AWS compatibility, workload dependencies, on-premise network and future network designs and migration business impact. The focus and tasks performed in this phase may depend on the size and complexity of client’s on-premise environment, as well as the client’s cloud strategy. Usually, machines can be migrated to AWS using one of the following approaches:

1. Re-Host: consists in moving certain workloads and tasks from on-premises to the cloud as is. The re-host migration is a common option for replicating on-premises applications in the cloud while avoiding costly and time-consuming re-design migration.
2. Re-Build or Workload migration: this consists in moving workload to a newly deployed Virtual Machine in AWS. This may be recommended due to a variety of reasons, such as Operating System upgrade, merging of multiple machines on a single one or just because it is the easiest approach for specific workloads.

It is important to emphasize that, in this phase, collaboration and communication with the client is essential as we will need to know all details about their workloads, services, application and dependencies.

At the end of this phase, we will create an "Migration Design" document containing the following:

- Executive Summary
- Application Discovery
- Server Detailed Discovery
- Network Details
- EC2 Instance Recommendations
- Cost Estimation
- High-level migration Flow

3 Pre-Migration

Following acceptance of the Migration Design document by the client, we will move to pre-migration phase, in which we will set up the initial cloud infrastructure, making it ready for the migration to be carried out. In this phase, the client will need to provide us their cloud subscription details to kick start the migration process:

- Subscription Name
- User Name
- Password

After that, we will start setting up the initial cloud infrastructure needed for the migration. This initial setup of the cloud environment must make it ready for the migration to be carried out. The initial infrastructure consists in:

- **VPC (Virtual Private Cloud):** Is a virtual network dedicated to an AWS account to provision a logically isolated section of the AWS Cloud where AWS resources can be launched. You have complete control over your virtual networking environment, including selection of your own IP address range, creation of subnets, and configuration of route tables and network gateways.
- **Subnet:** Subnets are containers within a VPC that segment off a slice of the CIDR block defined in the VPC. Subnets allow you to give different access rules and place resources in different containers where those rules should apply.
- **Simple Storage Server (S3):** Is an object storage service that offers industry-leading scalability, data availability, security, and performance. Amazon S3 provides easy-to-use management features so you can organize your data and configure finely-tuned access controls to meet your specific business, organizational, and compliance requirements. Given the benefits of Amazon S3 for storage, it is a good practice to use this service to store files and data sets for use with AWS Virtual Machines.
- **Location/Region:** As mentioned before, several factors as pricing, compliance, data residency, service availability, performance and reliability requirements will be considered when choosing which regions and locations AWS resources will be deployed.

4 Migration to AWS

In this section, we will describe the migration processes to Amazon Web Service cloud platform.

We will start the actual migration of the environment according to the migration plan defined on the Migration Design phase. The migration process may vary depending on the migration strategies and plan defined during the Migration Design phase.

In this section of the document, we will provide you high level descriptions and examples of migration processes to AWS platform. We have separated this section into 2 types of migrations:

- Migrations using the re-host approach, in which the machines are migrated as is to the Cloud. For this migration type, we will use AWS Application Migration Service.
- Re-build or workload migration, in which the workload is re-built on a new Virtual Machine in AWS.

In the re-host approach, we will describe below the process common to most migration projects. On re-build, the migration process is defined on each individual case, depending on the type of workload we are migrating and other various factors. As such, in this document, we are just providing some examples of migrations that TD SYNNEX performed previously for its clients.

4.1 Process for Re-Host Migrations utilizing AWS Application Migration Service (MGN)

A Lift-And-Shift migration to AWS will be done using AWS Application Migration Service. MGN replicates source servers into the AWS account. When ready, it automatically converts and launches the servers on AWS.

As during the discovery and assessment phase, migration can be performed using agentless or agent-less based methods. Generally, we will recommend the method to be used during as described in the table below:

Source Environment	Preferred Migration Approach	Secondary Migration Approach
VMWare	Agentless	Agent Based
Physical	Agent Based	Agent Based
Hyper-V	Agent Based	Agent Based

AWS Application Migration Service is supported in the follow AWS regions:
https://docs.aws.amazon.com/mgn/latest/ug/supported-regions.html

AWS Application Migration Service supports the following Operating Systems:
https://docs.aws.amazon.com/mgn/latest/ug/Supported-Operating-Systems.html

4.1.1 Migration flow for Agent based migration

The steps below describe the migration flow of an agent-based migration:

- Installation the AWS MGN Connector in the on-premises environment (AWS Replication Agent on the source servers).
- Ensuring proper connectivity between the on-premises environment and AWS.
- Defining the source servers and their properties in AWS MGN.
- Validation of server connectivity and credentials.
- Configuration of target AWS environment, including VPC, subnets, security groups, etc.
- Ensuring that appropriate resources and capacity are available in AWS for migration.
- Initiate replication of on-premises servers to AWS using AWS MGN.
- Monitoring replication progress and ensure data consistency.
- Waiting until the initial sync is finished.
- Launching of the test instances.
- Performing the acceptance tests on the servers. After the test instance is tested successfully, finalizing the test and delete the test instance.
- Wait for the cutover window.
- Confirm that there is no lag.
- Stop all operational services on the source server.
- Launch a cutover instance.
- Confirm that the cutover instance was launched successfully and then finalize the cutover.
- Archive the source server.

Once the on-premises environment is setup, we will perform all needed configuration to establish the connection between On-premise and AWS, enable the replication of the Virtual Machines and monitor the health of replication. We will also fix any issues during the replication and make sure that replication gets completed successfully.

At this stage there will be no impact on the business, but there could be a little impact on network bandwidth. This can be avoided with proper planning. During Migration Planning, we will discuss and agree the replication timings, for instance off business hours, based on the client's network bandwidth.

Note that some Virtual Machines may be incompatible for replication to AWS and that will be identified on the Migration Design phase. At that time, we will define a specific migration plan for those machines.

Once we have successfully installed the AWS vCenter client, all of the vCenter VMs will be added to AWS MGN in the DISCOVERED state. The DISCOVERED state means that the VM has not been replicated to AWS.

We will start the Replication using the MGN Console. Once started, the AWS MGN Console will indicate that data replication has started.

Once the VM has reached the Ready for testing state under Migration lifecycle, we can continue to launch test and cutover instances and perform all other regular AWS MGN operations on the server.

4.1.2 Agentless migration for vCenter source environments

Agentless snapshot-based replication allows us to replicate source servers on the customer's vCenter environment into AWS without installing the AWS Replication Agent.

In order to use agentless replication, the customer must dedicate at least one VM in the vCenter environment to host the AWS MGN vCenter Client.

The AWS MGN vCenter Client installation process will install services on the client VM which will allow AWS MGN to remotely discover your VMs that are suitable for agentless replication, and to perform data replication between the vCenter environment and AWS though the use of periodic snapshot shipping.

Agentless snapshot based replication is divided into two main operations:

- Discovery:
    - The discovery process involves periodically scanning the vCenter environment to detect source server VMs that are suitable for agentless replication, and adding these VMs to the AWS MGN Console.
    - Once a source server has been added, we can initiate agentless replication on the source VM using the MGN API or Console.
    - The discovery process also collects all of the necessary information from vCenter in order to perform an agentless conversion process once a migration job is launched.
- Replication:
    - The replication process involves continuously starting and monitoring the "snapshot shipping processes" on the source server VM being replicated.
    - The first snapshot shipping process performs an "initial sync" which sends the entire disk contents of the replicating VM into AWS.
    - Following snapshot shipping processes will leverage CBT (Changed Block Tracking) in order to only sync disk changes to the customer's target AWS account.

The following are the main system components of agentless replication:

- AWS MGN vCenter Client – A software bundle that is installed on a dedicated VM in the vCenter environment in order to facilitate agentless replication.
- vCenter Replication Agent – A java agent that is based on the AWS Replication Agent, which replicates a single VM using VDDK and CBT as the data source instead of the AWS MGN driver (that is used by the AWS Replication Agent)
- AWS MGN Service
- AWS MGN Console

The following are the VMware limitations of agentless replication:

- AWS MGN supports VMC on AWS for agentless replication.

- AWS MGN partially supports vMotion, Storage vMotion, and other features based on virtual machine migration (such as DRS and Storage DRS) subject to the following limitations:
- Migrating a virtual machine to a new ESXi host or datastore after one replication run ends, and before the next replication run begins, is supported as long as the vCenter account has sufficient permissions on the destination ESXi host, datastores, and datacenter, and on the virtual machine itself at the new location.
- Migrating a virtual machine to a new ESXi host, datastore, and/or datacenter while a replication run is active – that is, while a virtual machine upload is in progress – is not supported. Cross vCenter vMotion is not supported for use with AWS MGN.
- AWS does not provide support for migrating VMware Virtual Volumes.
- AWS MGN does not support replicating VMware VMs that have snapshots.

4.1.3 AWS MGN vCenter Client requirements

- The customer must dedicate at least one VM in the vCenter environment to host the AWS MGN vCenter Client
- AWS MGN vCenter Client must be installed on a VM that has that has outbound and inbound network connectivity to the AWS Application Migration Service API endpoints and outbound and inbound network connectivity to the vCenter endpoint. Customers who want to use PrivateLink can use VPN or DirectConnect to connect to AWS.

- The AWS MGN vCenter Client currently only supports VirtualDiskFlatVer2BackingInfo VMDK on CBT.
- The AWS MGN vCenter Client requires the following vCenter user permissions for agentless deployment. It is a best practice to create a dedicated role with these permissions and a dedicated user group with which the role will be associated. Every new user created for the AWS MGN vCenter Client will need to be a member of that group in order to obtain the required permissions. The vCenter predefined role: " Consolidated Backup user (sample) " provides most of these permissions. If that role is used, the following additional permission must be provided: Toggle disk change tracking.
    - Change configuration
        - Acquire disk lease
        - Toggle disk change tracking
    - Provisioning
        - Allow read-only disk access
        - Allow virtual machine download
    - Snapshot management
        - Create snapshot
        - Remove snapshot
- The VM on which the AWS MGN vCenter Client is installed should meet the following RAM, CPU, and memory requirements:
    - Minimal requirements (these requirements will allow the replication of up to 5 servers in parallel) – 2 GiB RAM, 1 core, 10 GiB of free disk space
    - Optional performance requirements (these requirements will allow the replication of the maximum number of 50 servers in parallel) – 16 GiB RAM, 8 cores, 10 GiB of free disk space
- VMs that are being replicated into AWS should have at least 2 GiB of free disk space.
- The VM on which the AWS MGN vCenter Client is installed should not allow any incoming (ingress) traffic.
- The VM on which the AWS MGN vCenter Client is installed should only allow outgoing traffic as following:
    - Egress TCP on the port on which the vCenter API is ran.
    - Egress TCP on port 443 for communication with the MGN API.
    - Egress TCP on port 1500 – for the replication server.
- Patching of guest OS running AWS vCenter client should be handled by the customer as part of shared responsibility.
- IAM credentials used by the vCenter Client should be rotated on a regular schedule. IAM credentials can be regenerated by reinstalling the AWS Replication Agent.
- The VM that hosts the AWS MGN vCenter Client should only be used for client hosting and should not be used for any other purposes.

- The AWS MGN vCenter Client should be located in an isolated and dedicated network and considered a sensitive segment.

4.1.4 vCenter environment requirements for agentless migration

- AWS Application Migration Service (AWS MGN) supports VM hardware version 7 and higher with CBT activated. Customer must ensure that they upgrade any VMs they have to hardware version 7 or higher. Ensure that CBT support is activated in the vSphere deployment. MGN activates CBT on replicating VMs. Customer can deactivate CBT after cutover.
- The VM being replicated into AWS MGN must not contain any existing VMware snapshots.
- Once added to AWS MGN, snapshot-based replication will create snapshots on the replicated VM, which may result in slower disk performance.
- VMs with independent disks, Raw Device Mappings (RDM), or direct-attach disks (iSCSI, NBD) are not supported for replication into AWS MGN.
- The VM being replicated into AWS MGN can be either stopped or running. Changing the VM state during data replication will not affect data replication and will cause no data corruption.

4.1.5 Test Fail Over by Launching test instances

After we have added all of the source servers and configured their launch settings, we will be ready to launch a test instance. It is crucial to test the migration of the source servers to AWS prior to initiating a cutover in order to verify that the source servers function properly within the AWS environment.

AWS recommends that as a best practice, we should perform a test at least two weeks before we plan to migrate the source servers. This time frame allows us and the customer to identify potential problems and solve them, before the actual cutover takes place. After launching test instances, use either SSH (Linux) or RDP (Windows) to connect to the instance to ensure that everything is working correctly.

After the replication is completed, we will perform an initial test fail over to check servers' health and test applications' consistency. After an initial test done by our team, the environment's details, such as Virtual Machines details, credentials and new public IPs, if any, will be provided to the client so he can also validate that all servers and applications are working as expected and all data is in place. This testing can be done during business hours and with no downtime since new URLs for the environment will be provided. If the client finds any issue, we will fix those in this phase.

4.1.6 Final Cut Over Testing

Once this is completed, a final cutover testing will be performed by shutting down the on-premises servers and updating DNS records as required with new addresses on AWS.

If during this testing, issues that cannot be resolved within cut over time are found, the client can call out for rollback. Rollback steps should make sure that on-premises environment would work as previously. Rollback steps for AWS are:

- Clean up cloud resources which are created as part of Final Cut Over Test
- Power on on-premise Virtual Machines and servers
- Modify internal DNS records back to On-premise with internal IP Addresses
- Modify integration settings with other systems, if any
- Revert all the configuration changes, such as Application and Database configurations, if any
- Configure external DNS back to on-premises environment, if any
- Test on-premises environment

4.2 Process for Re-Build Migrations

As mentioned before, is quite challenging to describe this type of migrations in a generic way since a specific migration plan needs to be designed to ensure that applications and services provided by this machine are properly migrated to the cloud. Since it is difficult to describe a generic migration approach for these scenarios, in this section we will provide some examples of this type of migrations.

4.2.1 Active Directory Migration

Client Scenario

- Client has a single Active Directory domain in a single forest with two Domain Controller servers – a primary and an additional one.

Migration Recommendation

- **Active directory in AWS VM:** Active directory is configured on an AWS Virtual Machine.

  Recommendation would be to setup a new IaaS Virtual Machine in AWS and promote it as an additional domain controller. Then, replicate Active Directory and transfer the FSMO roles. Finally, decommission on-premises Active Directory servers.

  Lift & Shift migration of Active Directory machines are not recommended since it may cause conflicts in the domain and the creation of a new VM on AWS is an easier approach.

Migration Steps

- Setup new Virtual Machine in AWS.
- Install Active Directory Domain Service role and promote it as an additional domain controller.
- Replicate the on-premise Active Directory to the additional domain controller in AWS.
- Transfer the FSMO roles to the additional domain controller in AWS and make it primary.
- Spin up an additional Virtual Machine and configure it as an additional domain controller (if required).
- Decommission On-premise Active Directory servers. Decommissioning will be customer's responsibility.

4.2.2 File Servers Migration

Client Scenario

- The client has file servers. Distributed File System (DFS) has been configured on AD Domain Controllers so some application servers can reach files located in one of file servers. One of the file servers is Microsoft Windows Server 2003, Standard Edition, which is not recommended to be migrated by AWS lift & shift method.

Migration Recommendation

After capturing performance key indicators from the file servers and discussed this in collaborative meetings with the client, the file server can be migrated to AWS manually.

Migration Steps

- Create a new Virtual Machine in AWS and configure File Server role
- Use RoboCopy tool to migrate file AWS from current domain on-premise to target domain in AWS.
- Configure DFS role on Domain Controllers in AWS to point files to the File Server Virtual Machine
- Decommission the on-premise servers. Decommissioning will be customer's responsibility.