John Rofrano  
Senior Technical Staff Member, Cloud Automation Services, IBM Research  
jir12@nyu.edu or rofrano@us.ibm.com  

Additional Research by: Laura Shwartz, Ph.D., Chris Young  

Migration Techniques  
including Image Migration and Workload Migration
Agenda

- Cloud Migration Quick Review

- Image Migration
  - Workload Complexity
  - Choosing a Migration Approach
  - Image Migration
  - Image Fix-up

- *Tools for Image Migration*
  - Image Migration Demo
  - Live and Near-Live Migrations
  - Converter Software
  - Applications of Image Migration

- Workload Migration
  - Provision, Backup, Restore Motivation
  - Application Complexity
  - Workload Migration Demo
Cloud is...

Cloud Computing

Application
- Monitoring
- Content
- Collaboration
- Communication
- Finance

Platform
- Object Storage
- Identity
- Runtime
- Queue
- Database

Infrastructure
- Compute
- Block Storage
- Network

Laptops
Desksops
Phones
Tablets
What is Migration to Cloud?
Transformation is Migration

- Since the cloud is virtual you cannot “move” a “physical” computer into a cloud
  - But you can move the disk image into the cloud (which can be just as good)
    - …unless you had lots of “junk” on your computer because you might not want that on your new cloud server (note: we call this “your mess for less”)
  - Or you can build a new server in the cloud that looks just like the old server
    - When we build a new server we sometimes call this “workload” migration because we are migrating the “work” that the server is doing into another server in the cloud.

- What if the cloud doesn’t support your server’s OS platform?
  - Now you have to “re-platform” to a new OS which could get tricky
  - This is a lot like building a new server but what about all of the software? Will it still work?
Types of Migration

- **Image Migration**
  - Capture an image of the server and reconstitute in the cloud
    - P2V (Physical-to-Virtual)
    - P2I (Physical-to-Image)

- **Application / Workload Migration**
  - Build a new server to run the workload in the cloud
    - Re-install
    - Re-platform
    - Consolidation
  - Build from a Template
    - IAAS
    - PAAS
  - Build via Specification Cloning
    - Use the physical server specs to build an exact duplicate virtual server
Types of Clouds

- **Unmanaged Clouds**
  - Do-It-Yourself from scratch
  - Possibly bring your own image
  - Anything goes but you fix it when it breaks

- **Managed Cloud**
  - Assistance in on-boarding
  - Only certain image infrastructure is allowed
  - Bring your own image may not be supported
    - Or image compliance must be adhered to
  - Steady state is managed
    - Patch management
    - Upgrade management

- **Clouds can be Public, Private, or Hybrid**
  - Public to all on cloud provider’s infrastructure
  - Private to business on dedicated business infrastructure
  - Private to business on shared cloud provider’s infrastructure
Economics of the Cloud

- Appearance of infinite computing resources on demand
- Elimination of an up-front commitment by Cloud users
- Ability to pay for use of computing resources on a short-term basis as needed
- Not have to run and manage the computing infrastructure
Gain vs. Pain of Transformation

As you saw in Enterprise IT, real world (Lecture 2a)

You saw Analytics and Planning (Lecture 3)

Pain: We want this to be the least!

Gain: We want this to be the most!

E.g., a cloud as in Lecture 2

Source System Cost for steady state

Target System Cost for steady state

Design & Approach

Planning & Implementation

Realization

Improved HW/SW Utilization

Power Consumption

Improved Operational Processes

Steady state cost benefit

Migration Cost

- Improved Operational Processes
- Improved HW/SW Utilization
- Reduced Power Consumption
- Reduced Power Consumption
- Improved HW/SW Utilization
- Improved Operational Processes

As you saw in Enterprise IT, real world (Lecture 2a)

You saw Analytics and Planning (Lecture 3)

You saw Analytics and Planning (Lecture 3)

We start here with Migration (Lecture 4)

E.g., a cloud as in Lecture 2
Transformation to Cloud Architecture

Customer and infrastructure data collection
- (manual and automated)
  - Infrastructure data collection
  - Organizational data (owners ...)
  - Customer goals and plans

Analytics, Design, Planning
- Target options
- Technical compatibility
- Business constraints
- Placement optimization
- Wave planning
- ROI analysis

Procurement and physical setup
- Hardware and software ordering
- Setup of physical infrastructure if new

Migrate, modify
- Core migration methods:
  - Lift-and-shift
  - Copy
  - Provision and backup-restore
  - Reinstall
  - Apply changes from design
  - Start

Test and remediate
- Unit tests
- Comparison tests
- User acceptance tests

Synchronize and cutover
- Educate
- Resynchronize test system with source
- Switch operation
- Registrations etc.

Non-customer sources
- External product descriptions
- Benchmarks

Consolidated data repositories
- Customer data
- Historic data
- General technical and financial data
  - Models

Migration Analytics
- Operational Improvement

We'll talk most about this part
Lectures #5, #6: Automatic Migration

- Where ‘the business occurs’ the actual doing of the migration, after all the planning and procurement has concluded.

**Migrate, Modify**

**Core migration methods**
- Copy: Create an exact copy of the source servers, e.g. image copy. Copy this image between source and target directly or via intermediate. Live vs. non-live migrations
- Provision and backup-restore: Provision a new server based on details provided in design starting with OS installation. Introduce software packages in a phased approach. Initiate backup (source) and restore (target) once new system available
- Re-Install and migrate workload: Build a new server in the cloud and move the application to it.

**Apply changes from design (sometimes in middle of migration)**
- Use-case driven modifications often needed.
- Ability for direct operator intervention
- Modification scope varies widely, and is package specific
- For complex scenarios, code change may be necessary

**Start**
- Either manual initiation or scheduled initiation.
- Need for real-time situation awareness
- New ideas: Provisioning to Cloud environments, image analysis.

**Migrate, modify workflow**
- Process constrained by schedule and resource availability per consolidated data repositories
- Requirement for exact data alignment between migration tooling and analysis
- Permit migration engineer overrides and in flight modification (constrained)
End-to-End Cloud Migration Process

Discover
- Collect the client technical, application, and financial data

Analyze
- Select migration candidates
- Create macro design(s), ROI, and plan with client

Map
- Create micro design for selected sources and targets, using discovered data as reference
- Create provisioning request for new or migrated servers

Provision
- Build, provision, and configure the target platform(s) environment and application components:
  - OS
  - Middleware
  - Filesystems
  - Users
  - Hardware
  - Network

Migrate
- Perform migration of source middleware configuration to target environment(s)

Test
- Leverage User transactions to automatically test and compare target servers to source
- Application and script transformation to target OS

Application migrations often involve clean-up and re-architecture
- Migrate existing Servers
- Create net New Servers (Clone, Template, FOAK)
Definitions

- **Application Server** – software framework which provides an environment for applications to run in
- **Scripts** – generally small pieces of code which are used to “glue” together application functions and assist server administrators in performing server maintenance
- **Source** – the existing server in the customer environment
- **Target** – the new server in the cloud environment
- **Image** – a file which contains a complete operating system and possibly middleware
- **Instance** – a deploy image which is currently running
- **Instance Type** – a grouping of virtual hardware resources (CPU, memory, disk)
Definitions

- **WebSphere Application Server**

- **DB2 Database Server**

- **IBM HTTP Web Server**

- **Lotus Domino Mail/Application Server**
Definitions in Action

- Hypervisor
- Virtual Machine
- Operating System
- Middleware
- Middleware
- Script
- Image
- Instance
- Application
Definitions in Action

- App.ear
- AppDB
- WebSphere Application Server
- DB2 Data Server
- Clean.sh
- Red Hat Enterprise Linux
- Virtual Machine
- VMWare ESXi
- Medium
- RHEL001
Case Study for this Course
Case Study Details

- ACME wants to migrate some of their IT Services to the cloud
- They “think” they have about 2000 – 3000 Servers running a variety of:
  - Linux 64-bit, 32-bit
  - AIX
  - Sun Solaris
- Running middleware such as:
  - DB2
  - Oracle
  - MySQL
  - WebSphere Application Server
  - JBOSS
  - Tomcat
  - SAP
- In support of applications like:
  - Company Web Site
  - Human Resources
  - Manufacturing
  - Shipping and Logistics
  - Custom Written Departmental Applications
Capturing an image and moving to target environment.

IMAGE MIGRATION
## Classifying Workloads

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Migration Strategy</th>
<th>OS upgrade</th>
<th>MW upgrade</th>
<th>Application upgrade</th>
<th>OS/MW Post-configuration</th>
<th>Systems management Post-Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1 (live virtualization and migration)</td>
<td>&quot;classical&quot; p2v, migrate over to target and cut-over instantaneously post-migration</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>S2 (migration of stand-alone apps into test env)</td>
<td>Stand-alone app hosted on single machine on source is migrated into a virtualized test env on target</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>Yes (IP, hostname, filesystem)</td>
<td>Yes</td>
</tr>
<tr>
<td>S3 (distributed apps)</td>
<td>Distributed app hosted on multiple machines on source is migrated into a virtualized env on target</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>Yes (IP, hostname, UID, filesystem, dependencies)</td>
<td>Yes</td>
</tr>
<tr>
<td>S4</td>
<td>Like s3 but MW upgrade. Virtualize into target test env and run MW upgrades post-migration.</td>
<td>No</td>
<td>yes</td>
<td>Yes/no (depending on MW version)</td>
<td>Yes (IP, hostname, filesystem) followed by application specific re-configuration (Data sources, Queues, etc.)</td>
<td>Yes</td>
</tr>
<tr>
<td>S5</td>
<td>Like S2 but requires application upgrade. Migrate and upgrade installation post-migration</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>Yes (IP, hostname, filesystem)</td>
<td>Yes</td>
</tr>
<tr>
<td>S6</td>
<td>Like S1, S2, S3 but with OS/MW version upgrade. Rebuild system on virtualized target environment on new OS/MW version</td>
<td>yes</td>
<td>yes/no</td>
<td>yes/nono</td>
<td>Yes (everything needs to be reconfigured after re-building on target)</td>
<td>Yes</td>
</tr>
</tbody>
</table>
How to Decide Which Migration Type to Use?

- **Map**
- **Provision**
- **Migrate**

Workload/Application Migration

Discover/Analyze

- **Planning**
- **Analysis determines approach**

Fit for Scenario

Image Migration

- **App**
- **OS**
- **Image Fix-up**

Tooling

- **Image Copy**
- **Image Fix-up**

Test

- **user**
Image Migration Overview

- Provides the ability to capture a complete copy of an existing server
  - Software by means of capturing the contents of disks
  - Hardware (optionally) by documenting source details

- Physical to Virtual (P2V) and Physical to Image (P2I) perform identical conversions to image
  - P2V instantiates the image into a target hypervisor or cloud
  - P2I captures the image as file and stores on disk

- Two predominate approaches:
  - Hot/online
    - The image capture can occur on a running system
    - The level of activity on the system is directly related to the time to migrate and the chances of success
  - Cold/offline
    - The image capture occurs on a system which is not running its primary operating system
    - Typically performed using a “boot cd/dvd”, lightweight version of image capture software

- What are the Pros/Cons?
 Capturing an Image

- Take a running server and convert it to an image (file)
  - Also known as Physical to Image (P2I) or Virtual to Image (P2I)

VMware vCenter Converter, PlateSpin Migrate

Prep source (if in scope, shut down services) ➔ Convert physical to virtual ➔ Migrate to target pool to test ➔ Resume services on source (if in scope) ➔ Re-configure/fix-up virtual instance *

Migration (image-granularity)

VMware vCenter

Update infrastructure management tools (i.e., DNS) ➔ Update service management tools ➔ IT health check, compliance check ➔ Sync data (if in scope) ➔ Promote to production pool ➔ Shutdown & decommission source servers

WMW, Fixup Server

Upgrade MW ➔ Fix-up MW/application configuration ➔ Test application

Post-migration transition to steady-state (batch or image-granularity)

*Automated fix-ups: Virtual hardware (storage), OS changes (IP Address, computer name), some MW configurations
Manual fix-ups: Some MW/Application-level configurations
Example of an actual image migration.

IMAGE MIGRATION DEMO
Migration Demo Architecture (i.e., nothing up my sleeve)

Migrate LAMP Stack onto ESXi

VMware ESXi Hypervisor (a.k.a. “the cloud”)

LAMP Stack (“source”)

WinXP (“client”)

VMware Converter (“P2V tool”)

LAMP Stack ’ (“target”)

VMware Fusion Hypervisor

Host Mac OS X
Migrations in real-time and near-real-time.

LIVE AND NEAR-LIVE MIGRATION
Cloud to Cloud Migration

- **Problem**
  - How do we perform long distance migrations?
    - Could be cloud-to-cloud or data center to data center

- **Solution 1 (Near Live)**
  - Use VMware Storage VMotion (SVMotion) to move instances to a local NFS Server
  - Sync local NFS Server with remote NFS Server (across data centers/clouds)
  - Shutdown source server ("near live")
  - Rsync the NFS Servers
  - Use VMotion to register instance with new hypervisor and start it
  - Use VMware Storage VMotion to move the instance into new hypervisor

- **Solution 2 (Live)**
  - Have a common NFS Server
  - Use SVMotion to move instance from hypervisor to NFS Server
  - Use VMotion to assign instance to new hypervisor
  - Use SVMotion to move instance from NFS Server to new hypervisor
Near-Live Migration
Live Migration
Why the Network Filesystem?

- VMotion requires that the source and target hosts be within the same vCenter and that the source and target hosts have a shared datastore on which the source VM resides.
  - In many cases, this is not possible. Thus we cannot do a simple VMotion.

- We are not able to use SVMotion to copy the contents of the local datastore from the source side to the target local datastore, because SVMotion requires that the source and target of an SVMotion be within the same vCenter.

- eMigrate solves this problem by providing an NFS datastore on both sides, mounted on the respective hosts, so that an SVMotion can be accomplished between the local datastore and the NFS datastore while having the VM active.
Other applications of image migration based technology.

APPLICATIONS OF IMAGE MIGRATION
Disaster Recovery Cloud

- Traditional IT Lifecycle

On-Going Operations

Continuity Service Provider

Inventory of Machines

On-Going Operations

Continuity Service Provider

Inventory of IT Services

Invocation

Customer

Declare Disaster

Admin

Install services

Invocation

Customer

Declare Disaster
Disaster Recovery Cloud

- Traditional IT Lifecycle
  - On-Going Operations
    - Continuity service provider manages redundant computer systems
    - Pre-configuration of computer systems up to the Operating System level
  - Invocation
    - Customer gets pre-configured systems and installs Middleware and services on top of it

- Recovery Cloud
  - On-Going Operations
    - “Service Replication” process creates a “Service Replica” that is stored on VM images in the cloud
    - Configuration and data of the “Service Replica” are kept in sync with the original service
  - Invocation
    - Customer starts up the latest VM images and can use the “Service Replica”
Disaster Recovery Process

- Backup process
Disaster Recovery Process

- Image based backup process
Disaster Recovery Process

- Recovery process

Create Latest Backup Instance

Done 😊
Comparison of converter software.

CONVERTER SOFTWARE
Sample Products

- VMWare vCenter Converter

- Novell PlateSpin Migrate

- Vision Double Take Move

- zCon zConverter

- Consider individual motivations from a business perspective
  - What usage model do they impose?
  - High labour? Pay per use? Limited functionality?
VMWare vCenter Converter

- Sample sources for converter software:
  - Physical to Virtual
  - Virtual to Virtual
  - Third Party Images to Virtual

- Sample destinations:
  - Bare metal hypervisor
  - OS based hypervisor

Source: http://www.vmware.com/products/converter/
Uploading an Image to EC2

- Clouds differ in terms of their acceptance of images
  - Unmanaged cloud – upload anything you like
  - Managed cloud – image must meet strict requirements

- Example with Amazon’s EC2
  - Extension for VMWare vCenter Converter
    - Allows EC2 to be seen as a target for migrations
  - EC2 API tools command

```
ec2-import-instance DISK_IMAGE_FILENAME -t INSTANCETYPE -f FORMAT -a ARCHITECTURE-SYSTEM -b S3_BUCKET_NAME -o OWNER -w SECRETKEY
```

```
ec2-import-volume DISK_IMAGE_FILENAME -f FORMAT -s SIZE-IN-GB -z AVAILABILITY_ZONE -b S3_BUCKET_NAME -o OWNER -w SECRETKEY
```
# Comparison Charts

<table>
<thead>
<tr>
<th>Features</th>
<th>VMware</th>
<th>Rapid</th>
<th>Platespin</th>
<th>Double Take</th>
<th>Vizioncore</th>
</tr>
</thead>
<tbody>
<tr>
<td>Management</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Centralized Management Console</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hot/Cold Cloning on local datacenter</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hot/Cold Cloning on remote datacenter</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Automatic discovery of candidates in Management Console</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>On the fly hardware customization</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Agentless</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scheduled tasks</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Conversions type supported</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Converters in Action

- VMware vCenter Converter
  - http://www.youtube.com/watch?v=IXJT0MTR7FM

- NetApp Virtualization
  - http://www.youtube.com/watch?v=kexvPM08qNo

- NetApp Rapid Cloning
  - http://www.youtube.com/watch?v=vqidYMcrOvI

- Platespin Migrate
  - http://www.youtube.com/watch?v=G4ErJ1Ro_Xs

- Double-Take Virtual Recovery Assistant
  - http://www.youtube.com/watch?v=icAdovkhIHY
Build a server in the cloud to run the workload

WORKLOAD MIGRATION
## Classifying Workloads

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Migration Strategy</th>
<th>OS Upgrade</th>
<th>MW Upgrade</th>
<th>Application Upgrade</th>
<th>OS/MW Post-configuration</th>
<th>Systems management Post-Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1 (live virtualization and migration)</td>
<td>“classical” p2v, migrate over to target and cut-over instantaneously post-migration</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>S2 (migration of stand-alone apps into test env)</td>
<td>Stand-alone app hosted on single machine on source is migrated into a virtualized test env on target</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>Yes (IP, hostname, filesystem)</td>
<td>Yes</td>
</tr>
<tr>
<td>S3 (distributed apps)</td>
<td>Distributed app hosted on multiple machines on source is migrated into a virtualized env on target</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>Yes (IP, hostname, UID, filesystem, dependencies)</td>
<td>Yes</td>
</tr>
<tr>
<td>S4</td>
<td>Like s3 but MW upgrade. Virtualize into target test env and run MW upgrades post-migration.</td>
<td>No</td>
<td>yes</td>
<td>Yes/no (depending on MW version)</td>
<td>Yes (IP, hostname, filesystem) followed by application specific re-configuration (Data sources, Queues, etc.)</td>
<td>Yes</td>
</tr>
<tr>
<td>S5</td>
<td>Like S2 but requires application upgrade. Migrate and upgrade installation post-migration</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>Yes (IP, hostname, filesystem)</td>
<td>Yes</td>
</tr>
<tr>
<td>S6</td>
<td>Like S1, S2, S3 but with OS/MW version upgrade. Rebuild system on virtualized target environment on new OS/MW version</td>
<td>yes</td>
<td>yes/no</td>
<td>yes/nono</td>
<td>Yes (everything needs to be reconfigured after re-building on target)</td>
<td>Yes</td>
</tr>
</tbody>
</table>
How to Decide Which Migration Type to Use?

**Discover/Analyze**

- Analysis determines approach

**Map**

- System z
- System p

**Provision**

- Cloud

**Migrate**

- Workload/Application Migration
- System z
- System p

**Test**

- User

**Fit for Scenario**

- Tooling
- Image Migration
- Image Copy
- Image Fix-up
- App OS
- App OS

**Image Fix-up**
### Application Configuration Examples

<table>
<thead>
<tr>
<th>Complexity</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>Single Server</td>
</tr>
<tr>
<td></td>
<td>• Web server (IHS)</td>
</tr>
<tr>
<td></td>
<td>• No external dependencies</td>
</tr>
<tr>
<td>Medium</td>
<td>Few Distributed Servers</td>
</tr>
<tr>
<td></td>
<td>• Two security zones</td>
</tr>
<tr>
<td></td>
<td>• Some external dependencies</td>
</tr>
<tr>
<td></td>
<td>• Well known network flows</td>
</tr>
<tr>
<td>High</td>
<td>Many Distributed Servers</td>
</tr>
<tr>
<td></td>
<td>• Several security zones</td>
</tr>
<tr>
<td></td>
<td>• Variations in middleware versions, editions</td>
</tr>
<tr>
<td></td>
<td>• Custom compiled code</td>
</tr>
</tbody>
</table>

![Diagram for Low Complexity](image1)

![Diagram for Medium Complexity](image2)

![Diagram for High Complexity](image3)
Technical Challenges

- Customized Environments
  - High degrees of customization allow for little speed-up/learning as the migrations progress
  - Tooling unable to effectively repeat one-of processes

- Information
  - Inadequate tools available to discover all source information, transform with migration policies and execute provisioning the new server

- Scale and Complexity
  - Applications with a large number of undocumented characteristics, many different technologies and implementations
Business Challenges

- **Cost**
  - Must define the migration such that a quick pay-back period can be realized through reduced capital expenditures, operational expenditures

- **Risk**
  - Must take into account lost revenue/increased cost associated with even a short application outage
  - Other factors at play which can complicate matters

- **Organizational**
  - Handoffs – partitioning of teams who must contribute to the migration
  - Cycle-time – long delays for initial discovery to be performed due to security, compliance risk
  - Qualification – challenging to identify the right parties which must sign-off on a migration (technical and business staff)
  - Skills – may not be readily available for the new cloud, operating system or middleware
  - Management systems – may not be appropriate for a fast-paced, iterative and collaborative process
How Can Tools Assist?

- Recipes and Recommendations for Target Servers
- Validation of both Source Data and Target Server Requests
- Automatic Provisioning of Full Target Server Stack
- Migrate current operational MW configuration from Source to Target
  - Supports MW Version Upgrades
- Utilize previous requests as template for additional new servers
Source to target attribute mapping.

MAP SOURCE TO TARGET
Map – Design a Migration

- A request centric tool which helps users to view source server configurations and map them to a new target server
- Collects key technical information to create the “container” which will hold the application
- Each section is influenced (or dictated) by the source server configuration:
  - **Hardware Architecture** – Intel x86, PowerPC, Mainframe
  - **Operating System** - Windows, Linux, AIX
  - **Middleware** - database, application server, mail server
  - **Filesystems** – size, mount, user, group, permissions
  - **Users/Groups** – owner, group, shell, home directory
  - **Hardware** – CPU, memory, swap
  - **Cloud** – hostname, network configurations
- Pair with administrative details related to the overall migration
Map – Migration Policies

- Many different areas to consider
- How does one know which items should be migrated, and which are no longer required?
- The action of moving a configuration from source to target is always made within a policy (implicit or explicit)
  - Like for like – migration of all configurations regardless of purpose
  - Justify everything – must prove a configuration is required before moving it over
  - Owners choose – distribute inclusion/exclusion to each component owner
- Policy is generally driven by underlying motivation for migration
  - Technology refresh?
  - Licensing savings?
  - Clean-up?
- The optimization criteria will differ for each policy
Map – Source Influences the Target

- Source on left-hand-side, target on right-hand-side
  - Select individual items to migrate (i.e. move user “bob”)
  - Manually adjust / modify based on outside factors
    - Migrate middleware to latest stable
    - Migrate middleware to latest and greatest
    - Migrate middleware to best match

- Backing configuration provides user with elements to include in the migration
  - Supported middleware products
  - Standard operating system images
  - Available clouds

- Collaborate with others to define the proper configurations
  - Look to subject matter experts to configure specific areas (i.e. database administrator to provide revised storage requirements)
Map – Operating System and Middleware

- Select the **operating system** which will be used for the target application deployment
  - Must be supported by the destination cloud
  - Previous analysis determined the proper version to move to considering:
    - Software compatibilities
    - Performance, security, reliability considerations
    - Organizational aspects (cost, expertise)
  - Operating systems available for selection include default configurations for filesystem space and users

- Select the **middleware** products required to support the application
  - If a specific middleware is not available then must manually flag the request for human intervention
  - Optionally include middleware which will augment the server (i.e. include new standard tools which should be used on the cloud)
Map – Users, Groups and Filesystems

- There are generally a large number of users and groups on systems
  - Some have accumulated over time and can safely be removed
  - Some represent human uses who must login to the server
  - Some represent application ids which are mandatory for the application to run
  - Groups are tied to specific security policies which is defined by the operating system (i.e. Windows ACLs, Unix sudo).

<table>
<thead>
<tr>
<th>Group</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Administrators</td>
<td>Add users, install/remove applications</td>
</tr>
<tr>
<td>DBAdmins</td>
<td>Start/stop database, backups</td>
</tr>
</tbody>
</table>

- Filesystems define the amount of space that will be made available to the application
  - Typically related to a user and group
  - Can be very detailed in terms of allocation to volume groups, logical volumes and the backing storage type (local, network attached storage, storage area network)
Demonstration of a workload migration tool

MAP DEMO