Experiences with Transformation to Hybrid Cloud: A Case Study for a Large Financial Enterprise

Acknowledgments: Simon Kofkin-Hansen, Long Wang, Christopher Moss, Richard Smith, Theodor Razvan Ionescu (IBM)
Defining the Cloud Models
[The NIST Definition of Cloud Computing, 2011]

- Private cloud
  - cloud infrastructure provisioned for exclusive use by a single organization, comprising multiple consumers
  - owned, managed and operated by the organization, a third party or some combination of them
  - on or off premises

- Public cloud
  - cloud infrastructure provisioned for open use by the general public
  - owned, managed and operated by one or more organizations, a third party or some combination of them
  - on or off premises

- Hybrid cloud
  - composition of two or more cloud infrastructures (usually public and private) that remain unique entities
  - those infrastructures are bound together by technology that enables data and application portability
The Case for Dynamic Hybrid Cloud

Private Cloud & IT

Benefits:
- Fully customizable
- Robust management
- Secure by design

Public Cloud

Benefits:
- Low entry cost
- Pay-per-use
- Highly elastic

Dynamic Hybrid

When done properly, dynamic hybrid cloud can yield
- Best of both worlds
- Better outcomes

- Maximize return on existing IT investments
- Match workloads to best-fit infrastructure
- Hit the right balance of risk to speed
- Meet seasonal capacity without CapEx
- Add new capabilities quickly
The market opportunity for Hybrid Cloud is large

50% of enterprises will have hybrid clouds by 2017

Source: Gartner
A Case Study: Hy-Cloud – A Hybrid Cloud for Finances R Us (FRS)

- Hybrid Cloud for a large financial enterprise
  - Requirements and Challenges
  - Solution Approach
  - Lessons Learned

- Enterprise is Real; Names are Fictional. Let's call:
  - The Enterprise: Finances R Us (FRS)
  - Resulting Hybrid Cloud Infrastructure and Services: Hy-Cloud
Hy-Cloud Requirements

- Deliver an on-premise private cloud without introducing any negative performance, latency, or availability effects to existing environment.
- Utilize the already prevalent virtualization environment (VMWare ESXi) on which FRS had already sunk expenses
- Deliver at least 50% improvement in service delivery SLAs through automation, orchestration, and workflow optimization
- Rapid automated provisioning of VMs and workloads
- Granular chargeback of resources consumed on a per-user, per-group, per-business-unit, and per-cost center basis.
- Integration with external public clouds: Amazon Web Services and Softlayer
- Unified management of all resource environments
- Single self-service portal for servicing all environments based on roles and entitlements
- Integration with FRS’s existing service management tools (like ticketing, service desk, patching, monitoring, configuration management, change management, directory services, authentication services)
Cloud Capability Maturity Model and Levels [IBM CCRA 4.0]
Cloud Capability Maturity Model and Levels [IBM CCRA 4.0]

Hy-Cloud Objective

1. Virtualized
   - Standardized services
   - Simple provisioning automation
   - Service catalog
   - Usage metering

2. Deployed
   - Image management
   - Monitoring
   - Backup and restore
   - Security and patch management

3. Optimized
   - Pattern-based provisioning
   - Service orchestration
   - Hybrid cloud integration

4. Enhanced
   - Cloud storefront
   - Integration with CRM and billing
   - ITIL Managed IaaS
   - Advanced IaaS

5. Monetized
   - Cloud Service Provider adoption pattern
   - Cloud Enabled Data Center adoption pattern
Cloud Adoption Use Case Packages [IBM CCRA]

Each $P_k$ (where $k=0..12$) refers to a set of related use-case packages.
Cloud Adoption Use Case Packages for Hy-Cloud [IBM CCRA]

Each $P_k$ (where $k=0..12$) refers to a set of related use-case packages.
Incremental Approach to Higher Business Value via Cloud Adoption [IBM CCRA 4.0]
Phases in Hy-Cloud Solution Design [IBM CCRA 4.0]

1. UNDERSTAND: the client’s business and needs
2. EXPLORE: options and approach
3. DEVELOP: and agree to client solution
4. IMPLEMENT: client solution
5. CONFIRM: client value and experience

- Understand Client
- Define Client Requirements
- Design Solution
- Detail Design to Define BOM
- Define Roadmap & First Project
Hy-Cloud Overview and Key Characteristics

- Resources and cloud services spanning multiple infrastructures
  - VMWare-based on-premise management platform integrated with Amazon Web Services (AWS) endpoints
  - Orchestration and automation using vRealize Orchestrator and Chef

- Automated provisioning, deployment, and configuration of middleware patterns and operating systems

- Custom design for customer-specific “Day 2” operations automated by cloud service catalog and workflows
  - Customer-specific operations, e.g. “switch an application from Dev to Prod environment”, “start/stop a Oracle instance”, “patch SQL server instance”.
  - Customer-specific roles and responsibilities

- Integration with FRS’s existing environment and processes
  - Active Directory, Monitoring, Request/Ticketing, Backup, Configuration Database,...
Patterns (specifically, cloud-computing patterns)
- Logical descriptions of physical and virtual assets that comprise a cloud-computing solution
- Useful in modeling multi-tiered application environments (e.g., SAP environment)
- Useful in modeling complex middleware environments (e.g., cluster environment for developing J2EE apps)

Pattern-based provisioning
- Prepares and supplies cloud-computing patterns within an IT environment and delivers them as a cloud service.
Foundations of Hy-Cloud: IBM Private Modular Cloud (PMC)

- Allows selection and modular assembly of hardware, software, features, and services
  - Reusable modular hardware, software, and services components
  - Start small and scale up quickly if and when needed

- Automated, fast deployment of private cloud using *pattern*-based approach
  - automated provisioning and de-provisioning of OSes (minutes not weeks)
  - automated provisioning and de-provisioning of workloads (minutes not months)
  - 200+ Application patterns (App, Middleware, and Database)

- Fully automated deployment and lifecycle management of cloud services

- Unified management of heterogeneous hybrid environments
  - individualized self-service portal/dashboard
  - automated provisioning
  - automated capacity adjustments
  - automated maintenance
  - “day 2” operations: creating snapshots, de-provisioning machines, reboot, etc.
Cloud Type | Manage-From | Manage-To
---|---|---
On-premise-Hybrid | On-premise | On-premise and AWS
Main Actors in a Typical Hybrid Cloud Scenario [IBM CCRA 4.0]

Cloud Service Provider

Cloud Service Consumer

Cloud Service Creator

- Business Manager
  - Customer Account Manager, Business Office

- Deployment Architect
  - Cloud Service Consultant, Cloud Education Specialist

- Transition Manager

- Service Manager

- Security & Risk Manager

- Operations Manager

- Operator

- Cloud Administrator

- Facilities Manager

- Customer Support & Care

- Offering Manager

- Service Component Developer

- Service Composer

- Consumer Business Manager

- Service Integrator

- Consumer Cloud Administrator
  - Tenant Administrator, Workload Administrator, Data Administrator, Security Administrator, Operator

- Consumer End User

Cloud Administrator specialties:
Storage Admin., Network Admin., OS Admin., Virtualization Admin.
Hy-Cloud: Main Actors [IBM CCRA 4.0]

Cloud Service Creator (IBM)

- Business Manager
  - Customer Account Manager, Business Office
- Deployment Architect
- Transition Manager
  - Cloud Service Consultant, Cloud Education Specialist
- Service Manager
- Security & Risk Manager
- Operations Manager
- Offering Manager

Cloud Service Provider (FRS IT Division)

- Service Component Developer
- Service Composer

Cloud Service Consumer (FRS Business Units)

- Consumer Business Manager
- Service Integrator
- Consumer Cloud Administrator
  - Tenant Administrator, Workload Administrator, Data Administrator, Security Administrator, Operator
- Consumer End User

Cloud Administrator specialties:
  - Storage Admin., Network Admin., OS Admin., Virtualization Admin.

© 2015 IBM Corporation
Hy-Cloud: Simplified Conceptual Architecture

Examples of Cloud Orchestrator and Controller Engine:
- Apache CloudStack
- HP Eucalyptus
- IBM Cloud Orchestrator
- Oracle Enterprise Manager Cloud Control
- VMware vRealize

Examples of Virtual Infr. Manager
- HyperV, vCenter, XenCenter

Endpoint to interface different types of compute resources.

An endpoint could be associated with a cloud account or virtual infrastructure manager.

Possible to have multiple endpoints for the same cloud or virtual infrastructure.

<table>
<thead>
<tr>
<th>Cloud Type</th>
<th>Manage-From</th>
<th>Manage-To</th>
</tr>
</thead>
<tbody>
<tr>
<td>On-premise-Hybrid</td>
<td>On-premise, Vmware based</td>
<td>On-premise, AWS, Softlayer</td>
</tr>
</tbody>
</table>

© 2015 IBM Corporation
Steps in Building Hy-Cloud

1. Using automation scripts, install the cloud management stack.
2. Using IBM's pattern library as a starting point, build a library of PaaS patterns and service catalog.
3. Build Process Automation Workflows that will get invoked when users selects items from the service catalog.
Hy-Cloud Architecture – Service Catalog

- **IaaS Services**
  - base infrastructure provisioning services
  - e.g., Windows VM, Linux VM conforming to customer (i.e., FRS) specified security and deployment standards, integrated into FRS’s service management processes (e.g., backup and monitoring)
  - Day 2 operations, e.g., update lease time, change resources, restart VM

- **PaaS Services**
  - Provision and deploy middleware patterns (i.e., OS + middleware/application stack) into specified environments
  - E.g., MS SQL pattern conforming to customer (i.e., FRS) specified security and deployment standards, integrated into FRS’s service management processes (e.g., backup and monitoring)
  - Day 2 operations, e.g., create database, change logfile settings, create JDBC connections

- **Support Services**
  - Open ticket for specified infrastructure tasks (e.g., firewall update)
# Hy-Cloud Pattern Examples

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Software Components</th>
<th>Features / Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle</td>
<td>Oracle</td>
<td>Single Server Oracle or multiple server Oracle RAC installation.</td>
</tr>
<tr>
<td>SQL Server</td>
<td>SQL Server</td>
<td>Single Server or multiple server clustered installation.</td>
</tr>
<tr>
<td>Hybrid Linux Web</td>
<td>Apache, Tomcat</td>
<td>Single Server installation of Apache and Tomcat.</td>
</tr>
<tr>
<td>Multi-Tier Windows Web</td>
<td>IIS, .NET + IIS, HA Proxy</td>
<td>Multiple tier installation of IIS and .NET. The SQL Server installation in this pattern may be optional.</td>
</tr>
</tbody>
</table>
Lessons Learned in:
Integration of Multiple Cloud Platforms

- Different cloud platforms (VMware, AWS) have different rules or capabilities
  - VM naming, agents for various capabilities, logon mechanisms, availability of endpoint plugins, etc.

- Different cloud platforms require coordination of different types of workflows. Workflow interactions covering multiple platforms may be particularly challenging.

- Credentials may be handled quite differently in different cloud platforms
  - Customer Active Directory, Quest Authentication Service, VMware user accounts, AWS accounts, vRealize business groups, AWS security groups, …
Lessons Learned in:
Automation of Middleware-Pattern Deployment and Configuration

- Automation of “last mile” is demanded
  – If VMs are provisioned in minutes but software deployment requires weeks, there is not much value in cloud provisioning
  – Hy-Cloud supports automated deployment of software combinations via Chef-based Rapid Deployment Service

- For the service catalog to be useful, it is important to identify common configurations of middleware patterns in customer applications
  – Applications created by hundreds of developers in the customer for different environments (Dev, Staging, Prod, …)
  – We designed an analytics engine to extract and analyze common configurations, and used those to guide pattern creation and customization

- PaaS automation means automated deployment, configuration and inter-connection of a number of middleware
Lessons Learned in:
Customer-specific Requirements and Automation Design

- Gathering of customer-specific requirements is time-intensive and may involve interactions with many parts of the customer’s organization
  - Interaction with multiple customer business units over many days may be needed to collect requirements covering
    - Deployment processes for a set of common operating systems and middleware
    - Clarification of roles, responsibilities, approval policies and processes
    - Determination of important post-provisioning operations (Day-2 operations)
    - Security requirements

- Customer-specific requirements will affect design of service catalog, service workflows, and self-service portal
  - Cloud service catalog access needs to be based on customer-specified roles and responsibilities
  - Service workflows need to cover Day-2 operations
    - E.g. “switch a QA customer application to Prod”, “start/stop a Oracle instance”, “patch a system”, “add JVM memory for a Weblogic instance”, “build a deployment from spec”, “enable backup for VMs”, etc.
  - Self-service portal needs to:
    - Support custom service catalog and Day-2 operations for multiple cloud platforms
    - Allow checking status of resources, components, and operations in multiple cloud platforms, through custom workflows
    - Accommodate customer’s special portal requirements (portal customization)
Lessons Learned in: Interactions with Existing Management Tools

- Many automated processes may require interacting with existing management tools in customer environment
  - Provisioning, post-provision operations
  - DNS, AD, ticketing, CMDB, backup, anti-virus, patching, monitoring, …
- Integration requires thorough understanding of existing tools and processes
- Integration requires buy-in from customer’s IT operations team
Hy-Cloud Summary

- Multiple cloud platforms (VMWare, AWS, Softlayer)
- Support for both on-premise and off-premise
- Emphasis on automation of “last mile” by using orchestration and pattern standardization
- Custom design for customer-specific entitlements, portal and post-provisioning operations
- Integration with customer’s existing environment, processes, and management tools
https://www.youtube.com/watch?v=8Nhued2B-cM

Epcot SCO Demo

https://www.youtube.com/watch?v=vkPU4Kd-yVU

IBM Private Modular Cloud Overview

DEMO
References

- The NIST Definition of Cloud Computing

- IBM Cloud Computing Reference Architecture (CCRA) 4.0

- IBM Private Modular Cloud

- Cloud Design Patterns
  – Microsoft Cloud Design Patterns
  – www.cloudpatterns.org