Inside Windows Azure: The Cloud Operating System

Microsoft Cloud Futures Workshop 2011

Mark Russinovich
Technical Fellow
Windows Azure

Agenda

- Introduction to Windows Azure
- Windows Azure Fundamentals
- Fabric Controller Internals
- Deploying a Service
- Updating a Service
- Host OS Upgrades
- Service Health

Windows Azure

- Windows Azure is an OS for the data center
  - Model: Treat the data center as a machine
  - Handles resource management, provisioning, and monitoring
  - Manages application lifecycle
  - Allows developers to concentrate on business logic
- Provides shared pool of compute, disk and network
  - Virtualized storage, compute and network
  - Illusion of boundless resources
- Provides common building blocks for distributed applications
  - Reliable queuing, simple structured storage, SQL storage
  - Application services like access control and connectivity

Windows Azure Application Philosophy: Design for Failure

- Scale out for capacity
- Scale out for redundancy
- Short time outs with retries
- Idempotent operations
- Stateless with durable external storage

Windows Azure Application Characteristics

Automated, Consistent Application Updates	Updates to the application occur in an automated way Updates result in clean components forcing consistency Local storage and OS are left untouched
Automated, Consistent Configuration Changes	Updates to the settings occur in an automated way Updates result in clean settings Local storage and OS are left untouched
Multi-Instance Management	Identical instances are deployed across the service Large scale-out services are guaranteed to be consistent No configuration drift
Scale-out	Application scale-out can occur automatically
High Availability	The application has no downtime, even in the face of hardware failures.
Automated, Consistent OS Servicing	The OS system hosting the application can be updated with the most recent patches in a coordinated and automated way.

Windows Azure Application Characteristics

Windows Azure

	Single Instance Persistent OS	Single Instance Stateless OS	Multi-Instance Stateless OS
Automated, Consistent Application Updates		✓	✓
Automated, Consistent Configuration Changes		✓	✓
Multi-Instance Management			✓
Scale-out			✓
High Availability			✓
Automated, Consistent OS Servicing		✓	✓

Windows Azure Platform Building Blocks

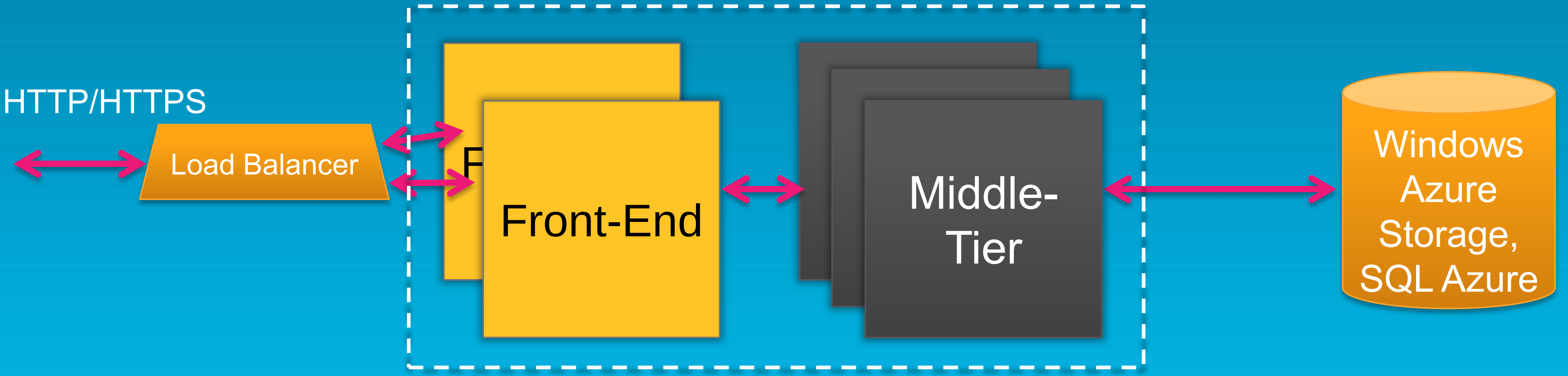
- Windows Azure Compute
- Windows Azure Storage
  - BLOBs
  - Tables
  - Queues
- Windows Azure CDN
- SQL Azure
- AppFabric PaaS Middleware Services
  - AppFabric Caching
  - AppFabric Service Bus
  - AppFabric Access Control Server

Agenda

- Introduction to Windows Azure
- Windows Azure Fundamentals
- Fabric Controller Internals
- Deploying a Service
- Updating a Service
- Host OS Upgrades
- Service Health

Modeling Cloud Applications

- A cloud application is typically made up of different components
  - Front end: e.g. load-balanced stateless web servers
  - Middle worker tier: e.g. order processing, encoding
  - Backend storage: e.g. SQL tables or files
  - Multiple instances of each for scalability and availability



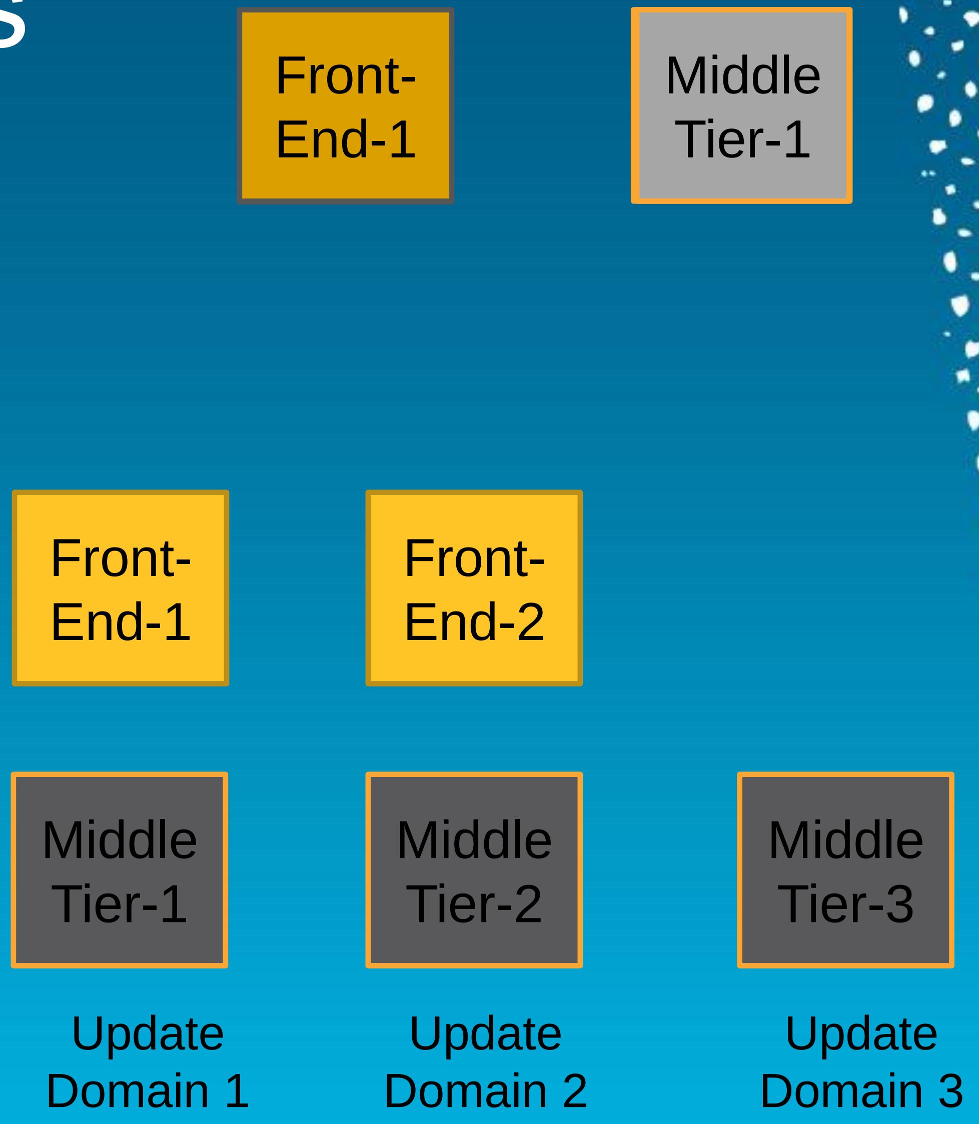
Mark’s Cloud Application

The Windows Azure Service Model

- A Windows Azure application is called a “service”
  - Definition information
  - Configuration information
  - At least one “role”
- Roles are like DLLs in the service “process”
  - Collection of code with an entry point that runs in its own virtual machine
- Windows Azure compute SLA requires two instances of each role
  - 99.95% for connectivity to two instances
  - Achieved with update and fault domains

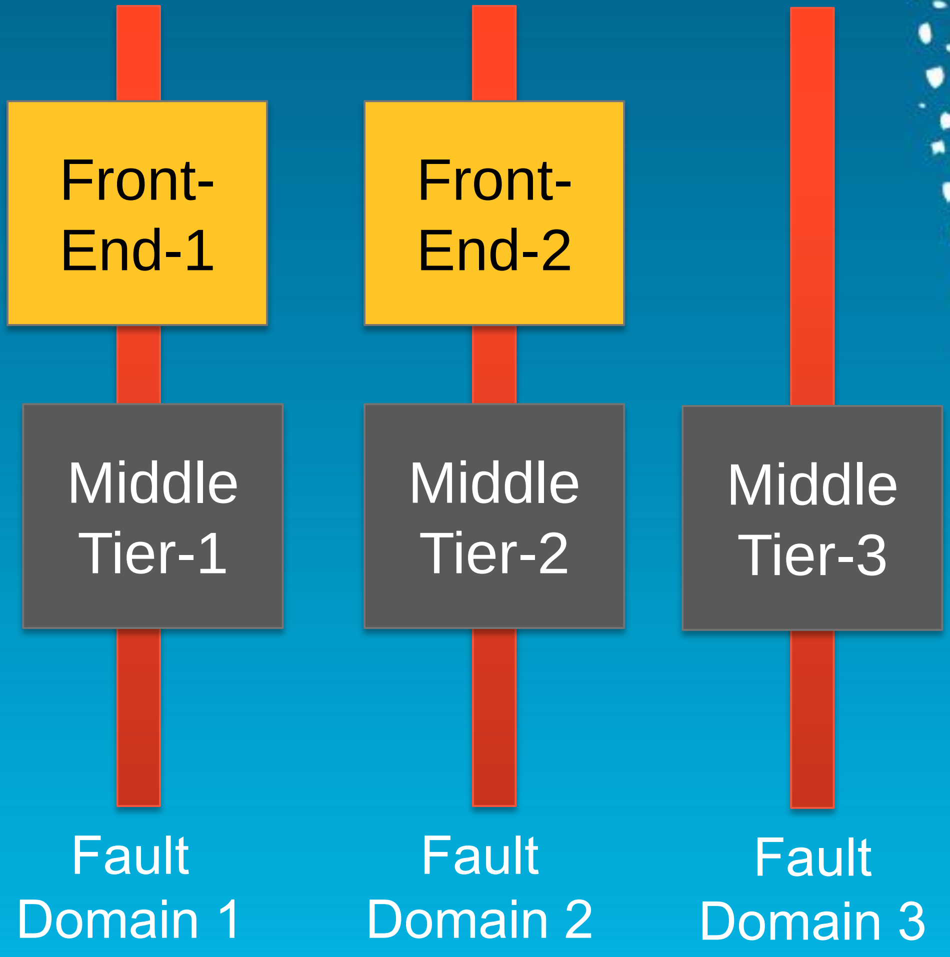
Availability: Update Domains

- Purpose: Ensure service stays up while updating and Windows Azure OS updates
- System considers update domains when upgrading a service
  - 1/Update domains = percent of service that will be offline
  - Default and max is 5, but you can override with upgradeDomainCount service definition property
- The Windows Azure SLA is based on at least two update domains and two role instances in each role



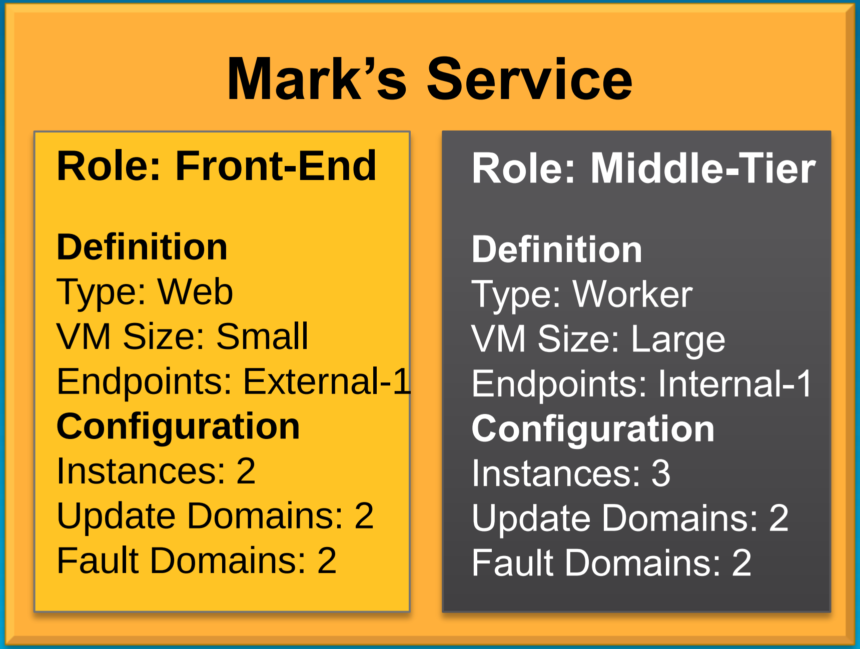
Availability: Fault Domains

- Purpose: Avoid single points of failures
  - Similar concept to update domains
  - But you don't control the updates
- Unit of failure based on data center topology
  - E.g. top-of-rack switch on a rack of machines
- Windows Azure considers fault domains when allocating service roles
  - 2 fault domains per service
  - Will try and spread roles out across more
  - E.g. don't put all roles in same rack



Role Contents

- Definition:
  - Role name
  - Role type
  - VM size (e.g. small, medium, etc.)
  - Network endpoints
- Code:
  - Web/Worker Role: Hosted DLL and other executables
  - VM Role: VHD
- Configuration:
  - Number of instances
  - Number of update and fault domains



Mark's Service

Role: Front-End

Definition
Type: Web
VM Size: Small
Endpoints: External-1
Configuration
Instances: 2
Update Domains: 2
Fault Domains: 2

Role: Middle-Tier

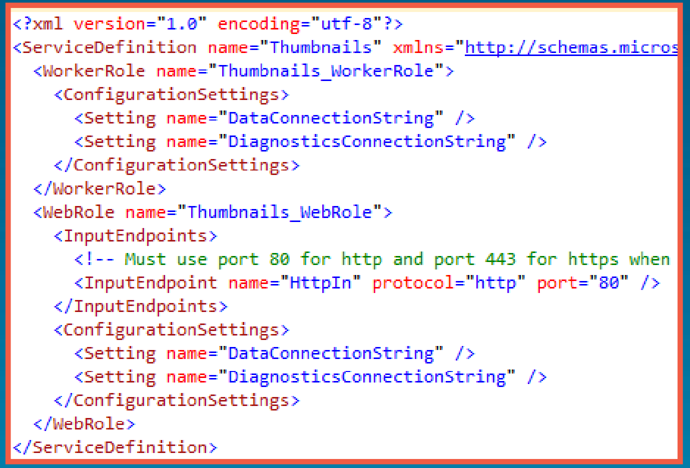
Definition
Type: Worker
VM Size: Large
Endpoints: Internal-1
Configuration
Instances: 3
Update Domains: 2
Fault Domains: 2

Role Types

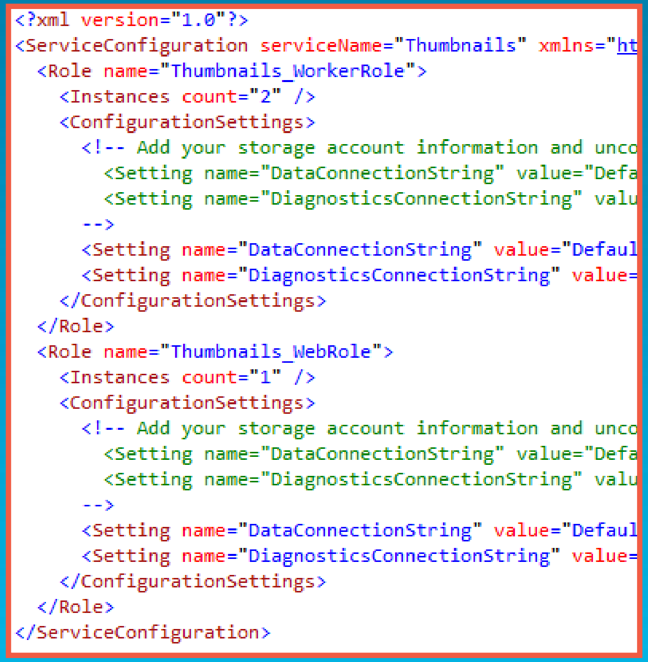
- There are currently three role types:
  - Web Role: IIS7 and ASP.NET in Windows Azure-supplied OS
  - Worker Role: arbitrary code in Windows Azure-supplied OS
  - VM Role: uploaded VHD with customer-supplied OS
- VM Role: is it a VM?
  - No, because it is stateless
  - Good for:
    - Long install (5+ minutes)
    - Manual install/config
    - Fragile install/config

Service Model Files

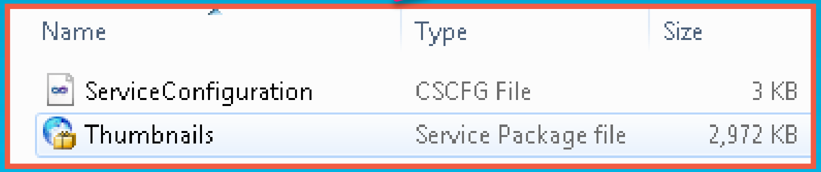
- Service definition is in ServiceDefinition.csdef
- Service configuration is in ServiceConfiguration.cscfg
- CSPack program Zips service binaries and definition into service package file (service.cscfg)



```xml
<?xml version="1.0" encoding="utf-8"?>
<ServiceDefinition name="Thumbnails" xmlns="http://schemas.micros
  <WorkerRole name="Thumbnails_WorkerRole">
    <ConfigurationSettings>
      <Setting name="DataConnectionString" />
      <Setting name="DiagnosticsConnectionString" />
    </ConfigurationSettings>
  </WorkerRole>
  <WebRole name="Thumbnails_WebRole">
    <InputEndpoints>
      <!-- Must use port 80 for http and port 443 for https when
      <InputEndpoint name="HttpIn" protocol="http" port="80" />
    </InputEndpoints>
    <ConfigurationSettings>
      <Setting name="DataConnectionString" />
      <Setting name="DiagnosticsConnectionString" />
    </ConfigurationSettings>
  </WebRole>
</ServiceDefinition>
```



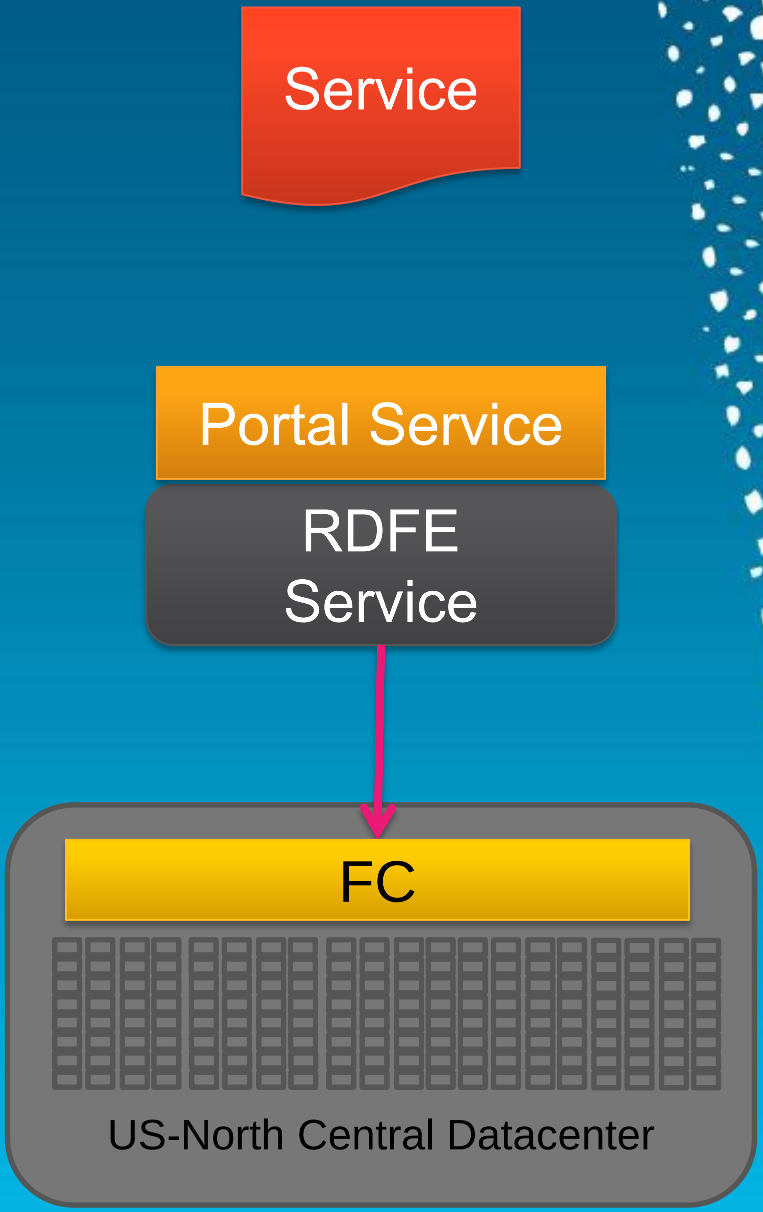
```xml
<?xml version="1.0"?>
<ServiceConfiguration serviceName="Thumbnails" xmlns="ht
  <Role name="Thumbnails_WorkerRole">
    <Instances count="2" />
    <ConfigurationSettings>
      <!-- Add your storage account information and unco
        <Setting name="DataConnectionString" value="Defa
        <Setting name="DiagnosticsConnectionString" valu
      -->
      <Setting name="DataConnectionString" value="Defaul
      <Setting name="DiagnosticsConnectionString" value=
    </ConfigurationSettings>
  </Role>
  <Role name="Thumbnails_WebRole">
    <Instances count="1" />
    <ConfigurationSettings>
      <!-- Add your storage account information and unco
        <Setting name="DataConnectionString" value="Defa
        <Setting name="DiagnosticsConnectionString" valu
      -->
      <Setting name="DataConnectionString" value="Defaul
      <Setting name="DiagnosticsConnectionString" value=
    </ConfigurationSettings>
  </Role>
</ServiceConfiguration>
```



Name	Type	Size
ServiceConfiguration	CSCFG File	3 KB
Thumbnails	Service Package file	2,972 KB

Deploying a Service to the Cloud: The 10,000 foot view

- Service package uploaded to portal
  - Windows Azure Portal Service passes service package to “Red Dog Front End” (RDFE) Azure service
  - RDFE converts service package to native “RD” version
- RDFE sends service to Fabric Controller (FC) based on target region
- FC stores image in repository and deploys and activates service

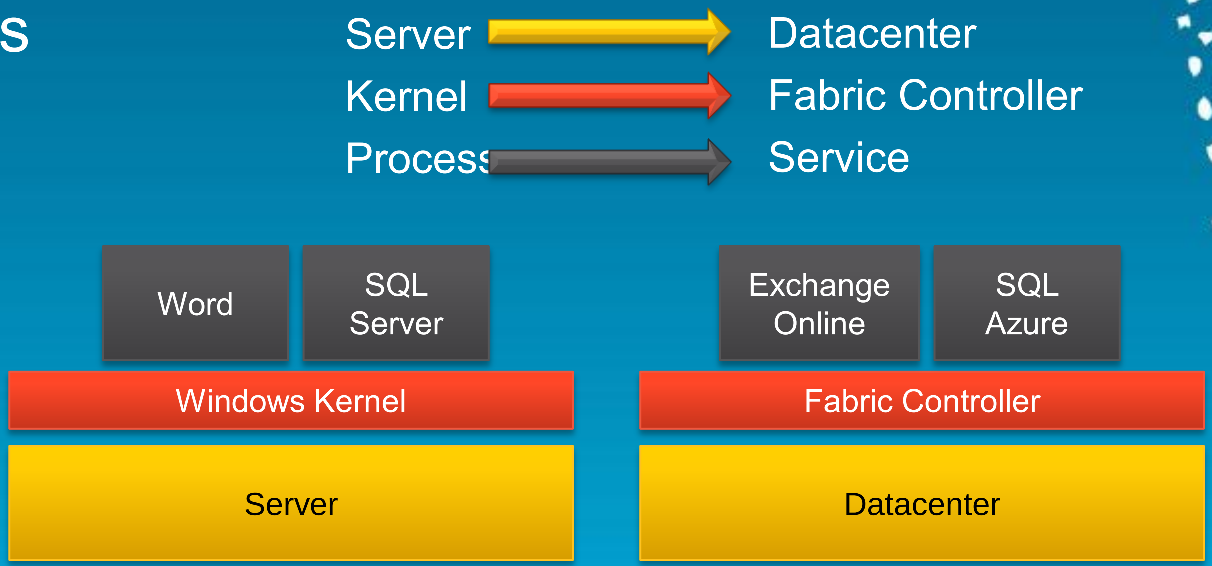


Agenda

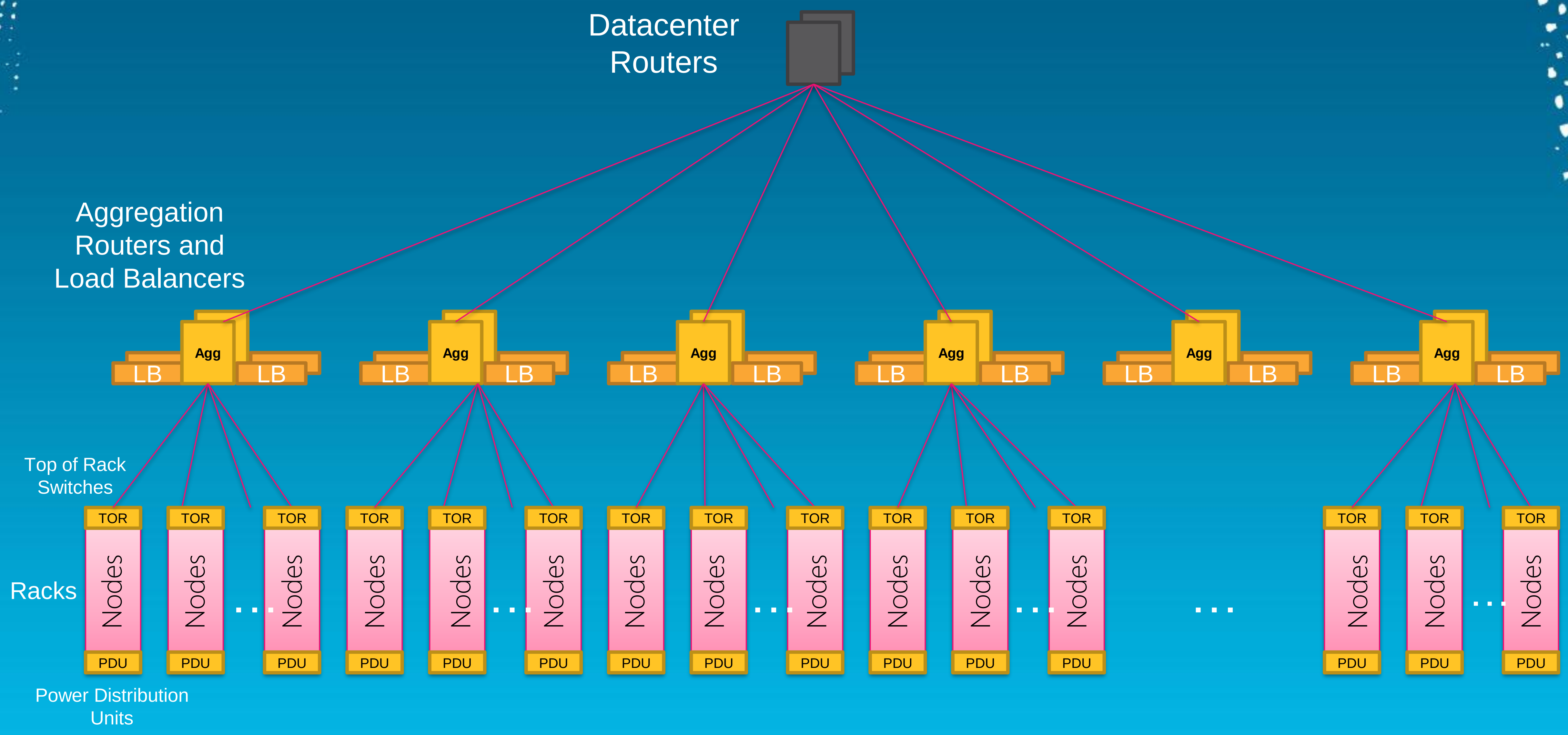
- Introduction to Windows Azure
- Windows Azure Fundamentals
- Fabric Controller Internals
- Deploying a Service
- Updating a Service
- Host OS Upgrades
- Service Health

The Fabric Controller (FC)

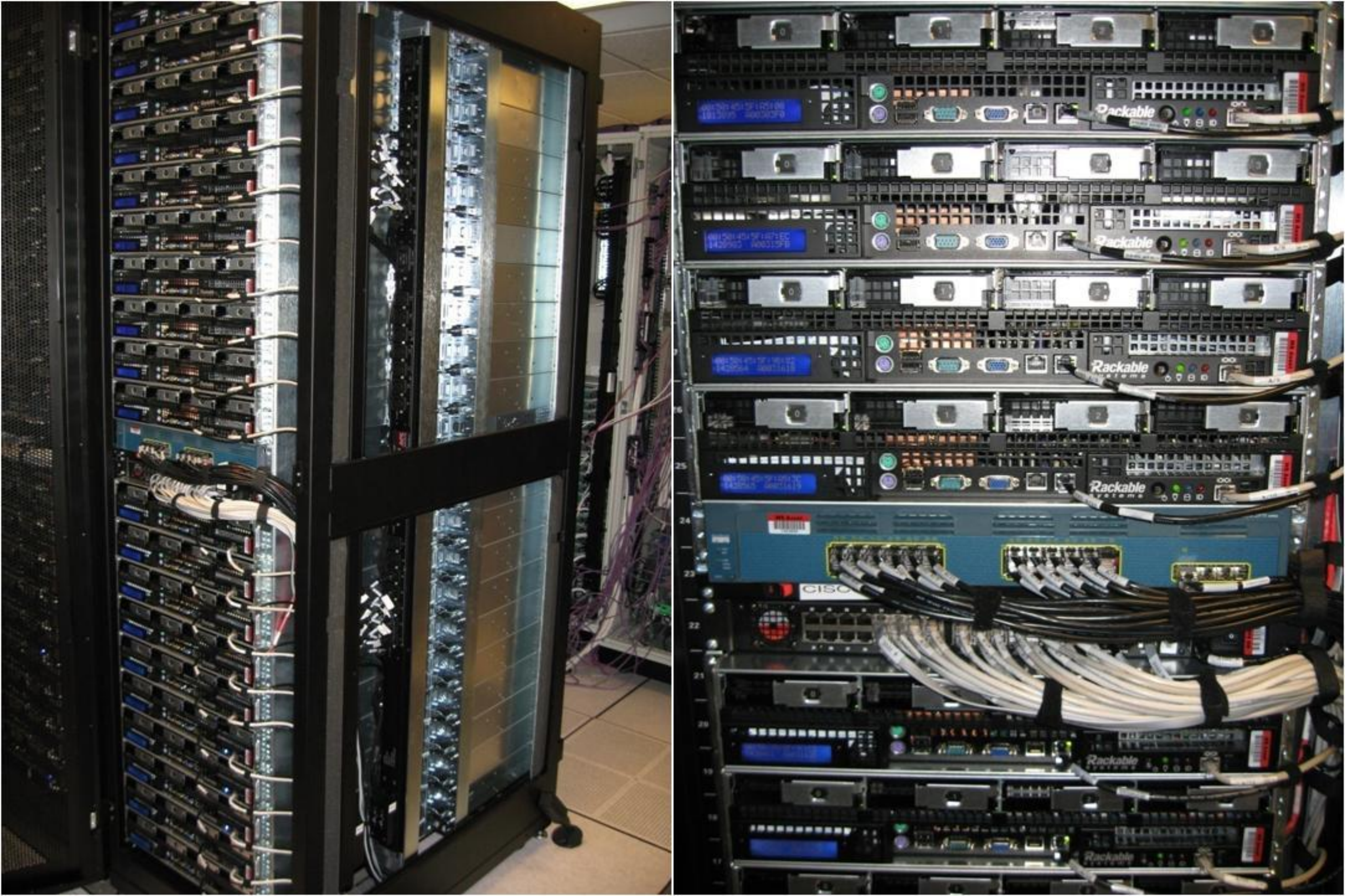
- The "kernel" of the cloud operating system
  - Manages datacenter hardware
  - Manages Windows Azure services
- Four main responsibilities:
  - Datacenter resource allocation
  - Datacenter resource provisioning
  - Service lifecycle management
  - Service health management
- Inputs:
  - Description of the hardware and network resources it will control
  - Service model and binaries for cloud applications



Datacenter Architecture

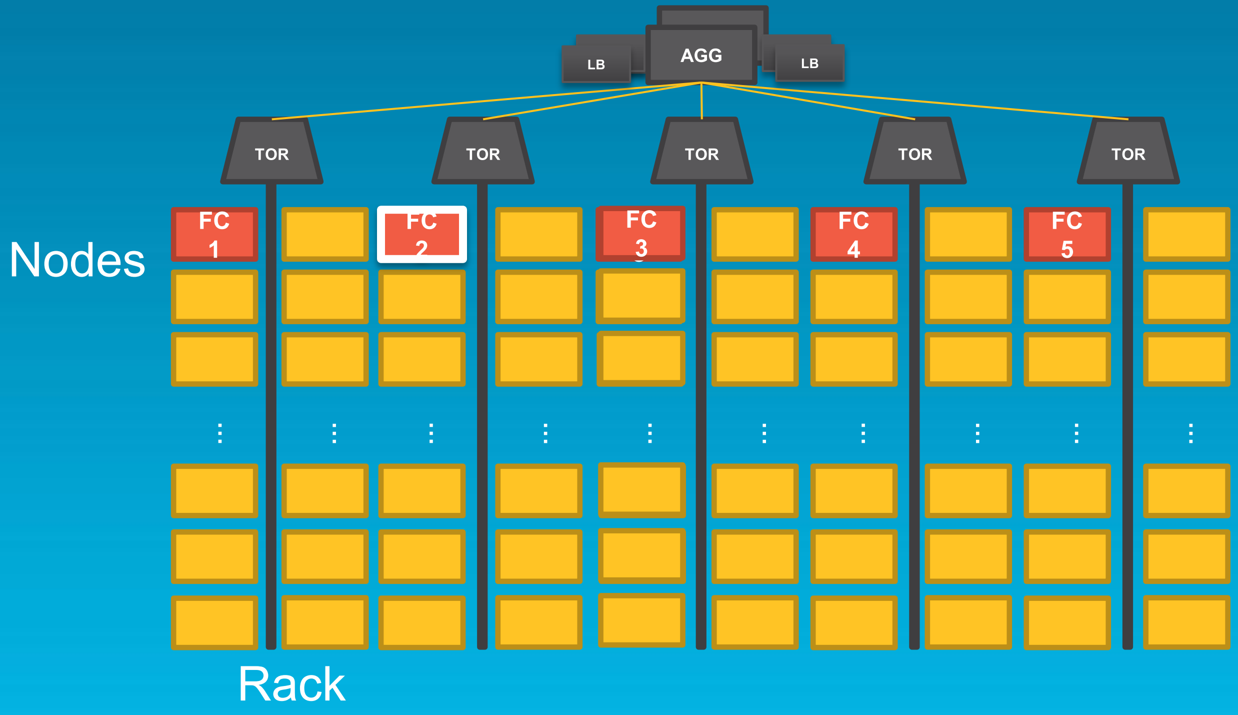


Windows Azure Datacenters



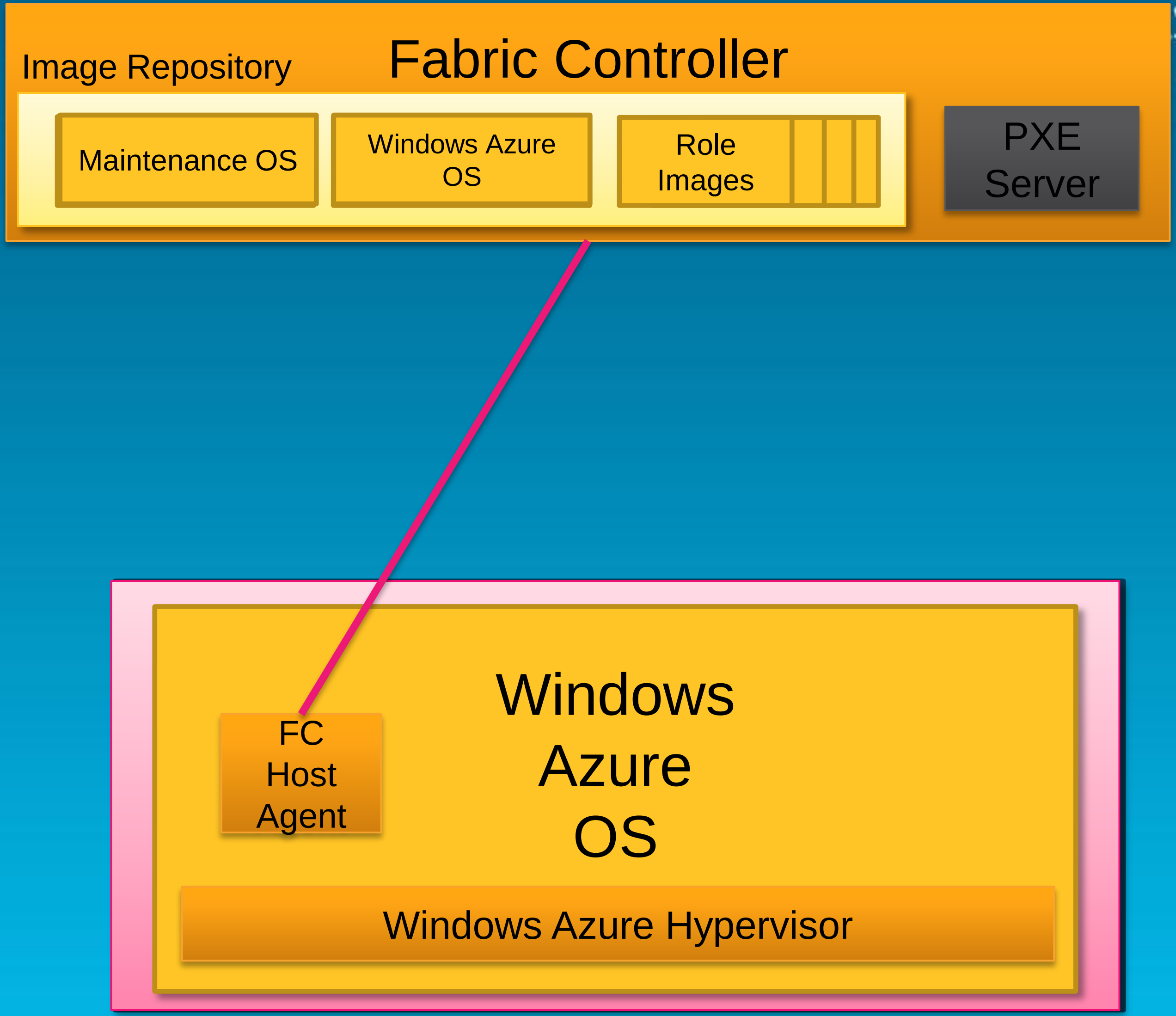
High-Level FC Architecture

- FC is a distributed, stateful application running on nodes (blades) spread across fault domains
  - Installed by “Utility” Fabric Controller
  - One acts as the primary and all others keep view of world in sync
  - Supports rolling upgrade, and services continue to run even if FC fails entirely

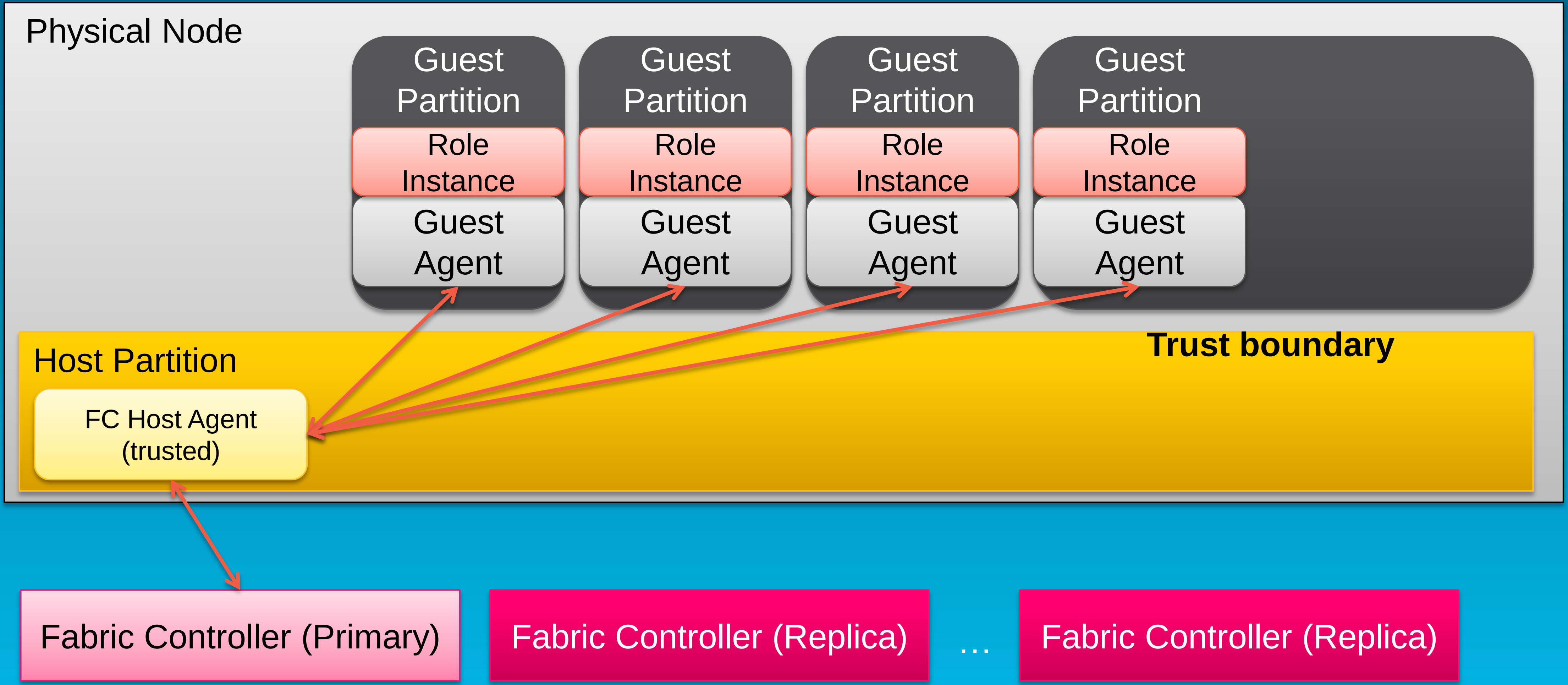


Provisioning a Node

- Power on node
- PXE-boot Maintenance OS
- Agent formats disk and downloads Host OS
- Host OS boots, runs Sysprep /specialize, reboots
- FC connects with the "Host Agent"



Inside a Node



Agenda

- Introduction to Windows Azure
- Windows Azure Fundamentals
- Fabric Controller Internals
- Deploying a Service
- Updating a Service
- Host OS Upgrades
- Service Health

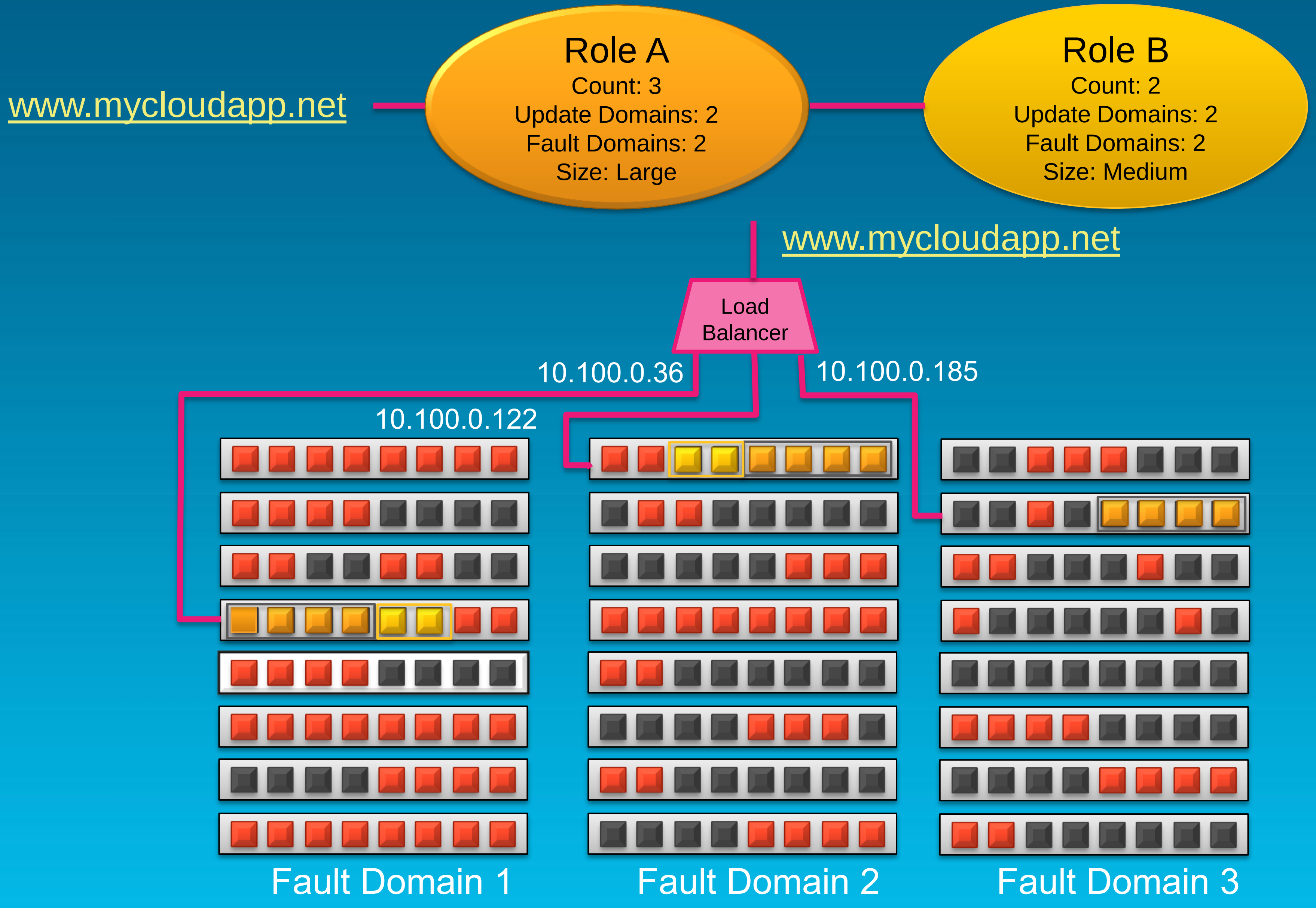
Service Deployment Steps

- Process service model files
  - Determine resource requirements
  - Create role images
- Allocate compute and network resources
- Prepare nodes
  - Place role images on nodes
  - Create virtual machines
  - Start virtual machines and roles
- Configure networking
  - Dynamic IP addresses (DIPs) assigned to blades
  - Virtual IP addresses (VIPs) + ports allocated and mapped to sets of DIPs
  - Programs load balancers to allow traffic

Service Resource Allocation

- Goal: allocate service components to available resources while satisfying all hard constraints
  - HW requirements: CPU, Memory, Storage, Net
  - Fault domains
- Secondary goal: Satisfy soft constraints
  - Prefer allocations which will simplify servicing the host OS/hypervisor: pick nodes that already have instances from the same update domain
  - Optimize network proximity: pack nodes
- Service allocation produces the goal state for the resources assigned to the service components
  - Node and VM configuration (OS, hosting environment)
  - Images and configuration files to deploy
  - Processes to start
- Service allocation also allocates network resources such as LB and VIPs

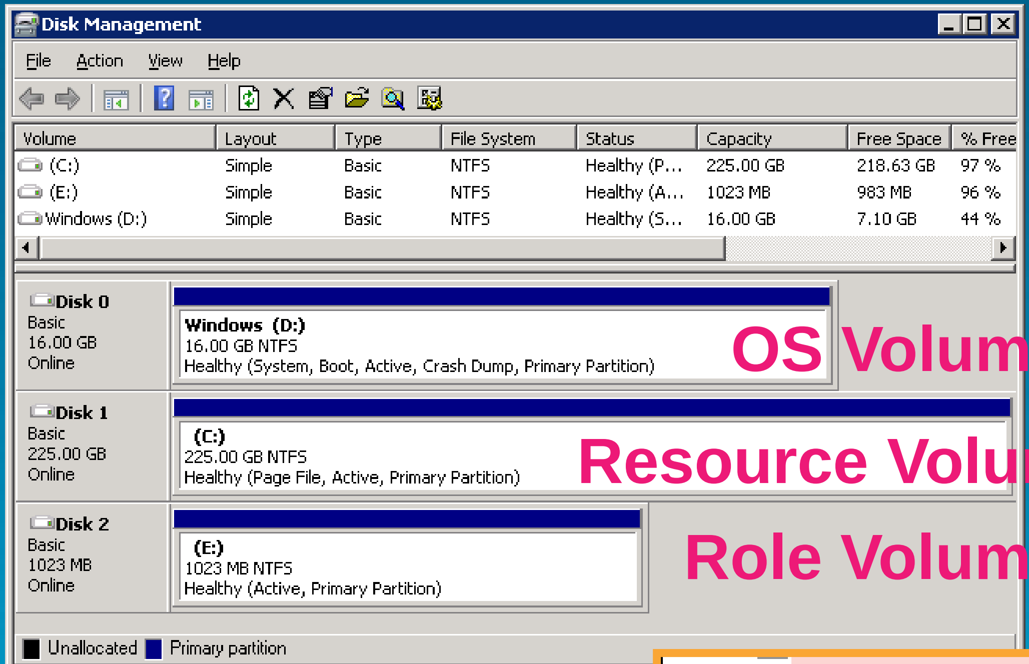
Example Service Allocation



Provisioning a Role Instance

- FC pushes role files and configuration information to target node host agent
- Host agent creates three VHDs:
    - Differencing VHD for OS image (D:\)
        - Host agent injects FC guest agent into VHD for Web/Worker roles
    - Resource VHD for temporary files (C:\)
    - Role VHD for role files (first available drive letter e.g. E:\, F:\)
- Host agent creates VM, attaches VHDs, and starts VM
- Guest agent starts role host, which calls role entry point
    - Starts health heartbeat to and gets commands from host agent
- Load balancer only routes to external endpoint when it responds to simple HTTP GET (LB probe)

Inside a Role VM



Volume	Layout	Type	File System	Status	Capacity	Free Space	% Free
(C:)	Simple	Basic	NTFS	Healthy (P...	225.00 GB	218.63 GB	97 %
(E:)	Simple	Basic	NTFS	Healthy (A...	1023 MB	983 MB	96 %
Windows (D:)	Simple	Basic	NTFS	Healthy (S...	16.00 GB	7.10 GB	44 %

Disk 0
Basic
16.00 GB
Online

Windows (D:)
16.00 GB NTFS
Healthy (System, Boot, Active, Crash Dump, Primary Partition)

Disk 1
Basic
225.00 GB
Online

(C:)
225.00 GB NTFS
Healthy (Page File, Active, Primary Partition)

Disk 2
Basic
1023 MB
Online

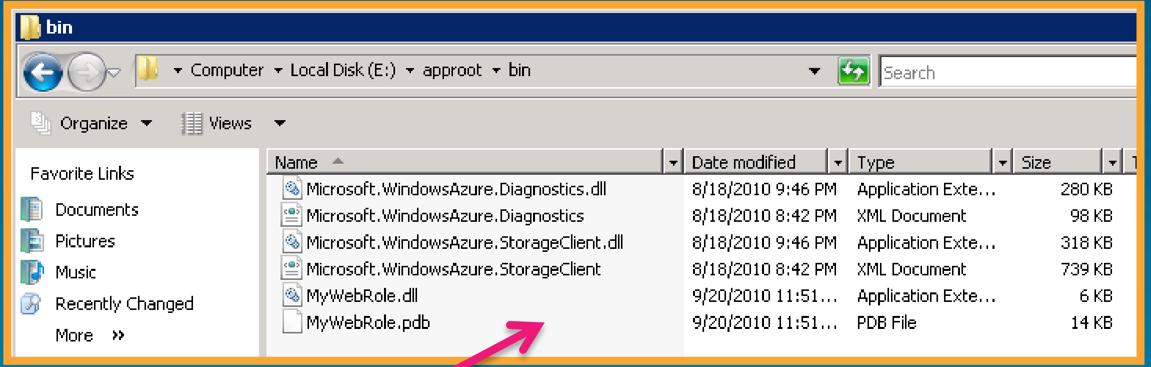
(E:)
1023 MB NTFS
Healthy (Active, Primary Partition)

■ Unallocated ■ Primary partition

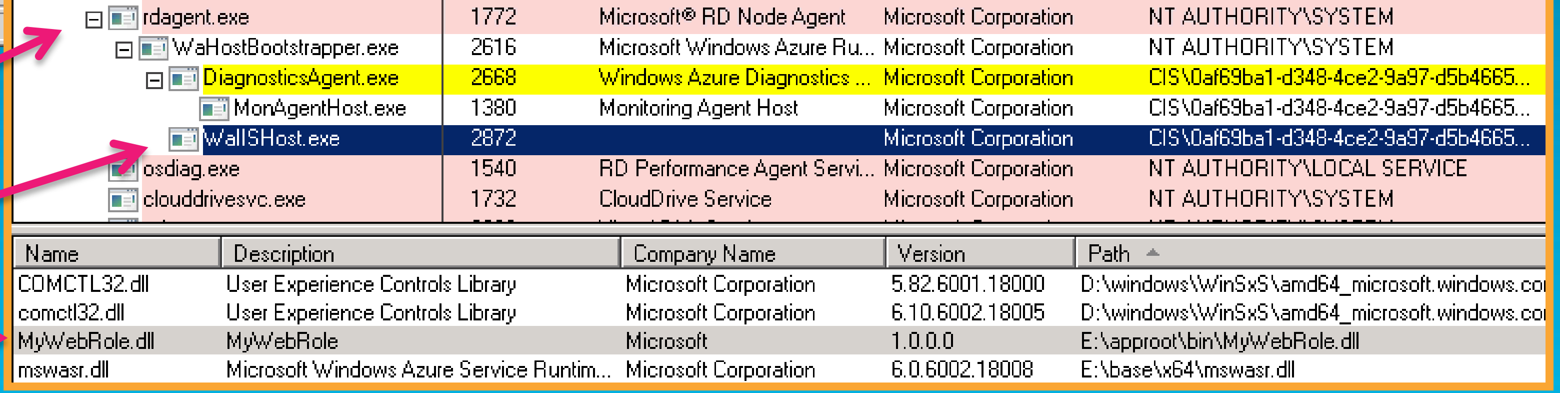
OS Volume

Resource Volume

Role Volume



Name	Date modified	Type	Size
Microsoft.WindowsAzure.Diagnostics.dll	8/18/2010 9:46 PM	Application Exte...	280 KB
Microsoft.WindowsAzure.Diagnostics	8/18/2010 8:42 PM	XML Document	98 KB
Microsoft.WindowsAzure.StorageClient.dll	8/18/2010 9:46 PM	Application Exte...	318 KB
Microsoft.WindowsAzure.StorageClient	8/18/2010 8:42 PM	XML Document	739 KB
MyWebRole.dll	9/20/2010 11:51...	Application Exte...	6 KB
MyWebRole.pdb	9/20/2010 11:51...	PDB File	14 KB



rdagent.exe	1772	Microsoft® RD Node Agent	Microsoft Corporation	NT AUTHORITY\SYSTEM
WaHostBootstrapper.exe	2616	Microsoft Windows Azure Ru...	Microsoft Corporation	NT AUTHORITY\SYSTEM
DiagnosticsAgent.exe	2668	Windows Azure Diagnostics ...	Microsoft Corporation	CIS\0af69ba1-d348-4ce2-9a97-d5b4665...
MonAgentHost.exe	1380	Monitoring Agent Host	Microsoft Corporation	CIS\0af69ba1-d348-4ce2-9a97-d5b4665...
WaIISHost.exe	2872		Microsoft Corporation	CIS\0af69ba1-d348-4ce2-9a97-d5b4665...
osdiag.exe	1540	RD Performance Agent Servi...	Microsoft Corporation	NT AUTHORITY\LOCAL SERVICE
clouddrivesvc.exe	1732	CloudDrive Service	Microsoft Corporation	NT AUTHORITY\SYSTEM

Name	Description	Company Name	Version	Path
COMCTL32.dll	User Experience Controls Library	Microsoft Corporation	5.82.6001.18000	D:\windows\WinSxS\amd64_microsoft.windows.co
comctl32.dll	User Experience Controls Library	Microsoft Corporation	6.10.6002.18005	D:\windows\WinSxS\amd64_microsoft.windows.co
MyWebRole.dll	MyWebRole	Microsoft	1.0.0.0	E:\approot\bin\MyWebRole.dll
mswasr.dll	Microsoft Windows Azure Service Runtim...	Microsoft Corporation	6.0.6002.18008	E:\base\x64\mswasr.dll

Guest Agent

Role Host

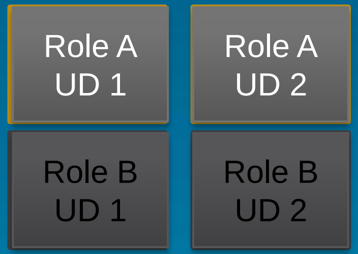
Role Entry Point

Agenda

- Introduction to Windows Azure
- Windows Azure Fundamentals
- Fabric Controller Internals
- Deploying a Service
- Updating a Service
- Host OS Upgrades
- Service Health

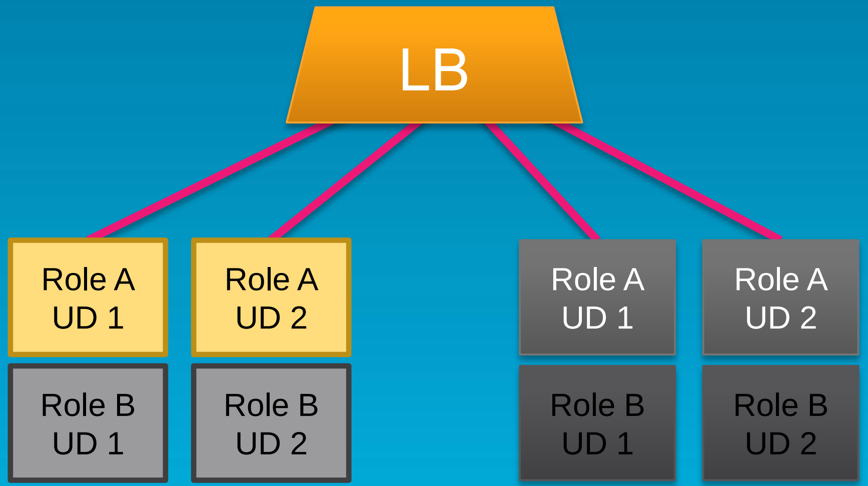
Update Types

- There are two update types:
  - In-place
  - VIP swap
- In-place update:
  - Role instances upgraded one update domain at a time
  - Two modes: automatic and manual
- VIP swap update:
  - New version of service deployed, external VIP/DIP mapping swapped with old



Role A UD 1
Role A UD 2
Role B UD 1
Role B UD 2

In-Place Update



LB
Role A UD 1
Role A UD 2
Role B UD 1
Role B UD 2
Role A UD 1
Role A UD 2
Role B UD 1
Role B UD 2

VIP Swap Update

In-Place Update Detail

- FC deploys updated role files and configuration to all nodes in parallel
- Prepares new role instances:
  - FC host agent creates new role VHD
  - Attaches and mounts new role VHD
- Stops old role instance:
  - FC instructs guest agent to stop role instance
  - Dismounts and detaches old role VHD
- Starts new role instances:
  - Calls new role code entry point
  - Considers role instance update successful when role code reports “ready”
- Note that resource volume is preserved updates of role instance

Agenda

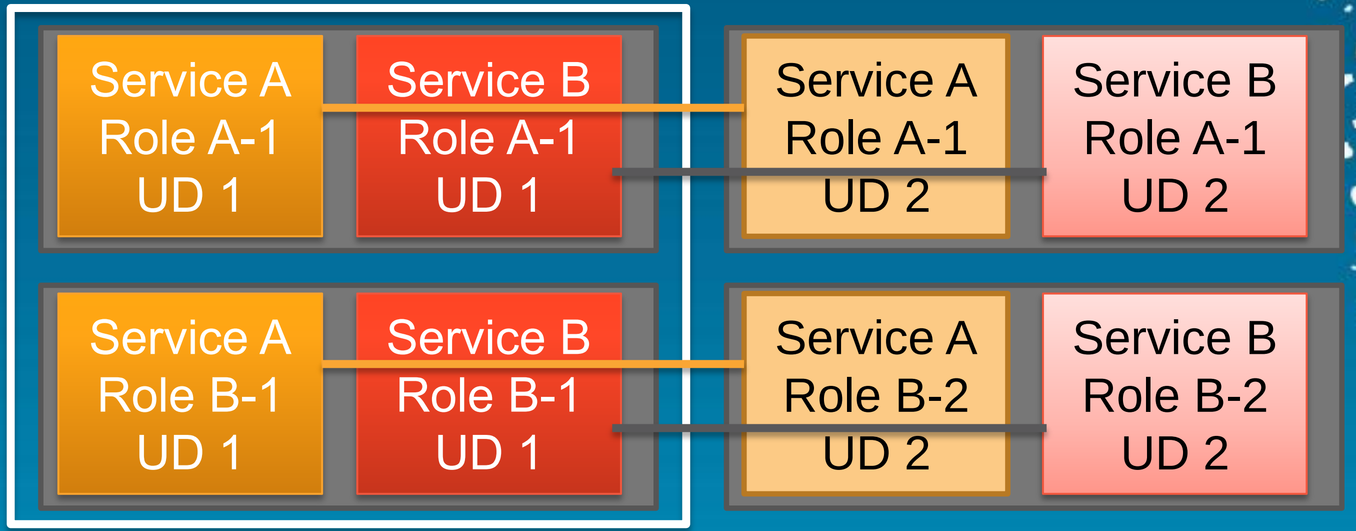
- Introduction to Windows Azure
- Windows Azure Fundamentals
- Fabric Controller Internals
- Deploying a Service
- Updating a Service
- Host OS Upgrades
- Service Health

Updating the Host OS

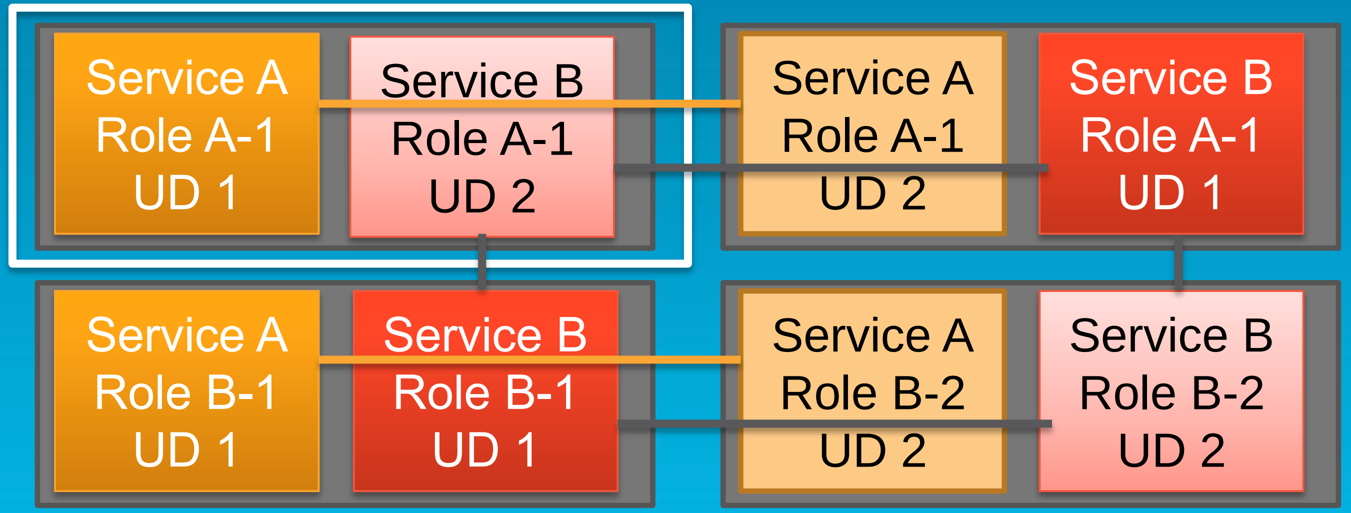
- Initiated by the Windows Azure team
    - Typically no more than once per month
- Goal: update all machines as quickly as possible
- Constraint: must not violate service SLA
    - Service needs at least two update domains and role instances for SLA
    - Can’t allow more than one update domain of any service to be offline at a time
- Note: your role instance keeps the same VM and VHDs, preserving cached data in the resource volume
- Essentially a graph coloring problem
    - Edges exist between vertices (nodes) if the two nodes host instances of the same service role in different update domains
    - Nodes that don’t have edges between them can update in parallel

Example Allocations

- Both allocations are valid from the services point of view
  - Allocation 1 allows for 2 nodes rebooting simultaneously
  - Allocation 2 allows only one node to be down at any time
- Host OS upgrade rollout is 2x faster with allocation 1



Allocation 1



Allocation 2

Agenda

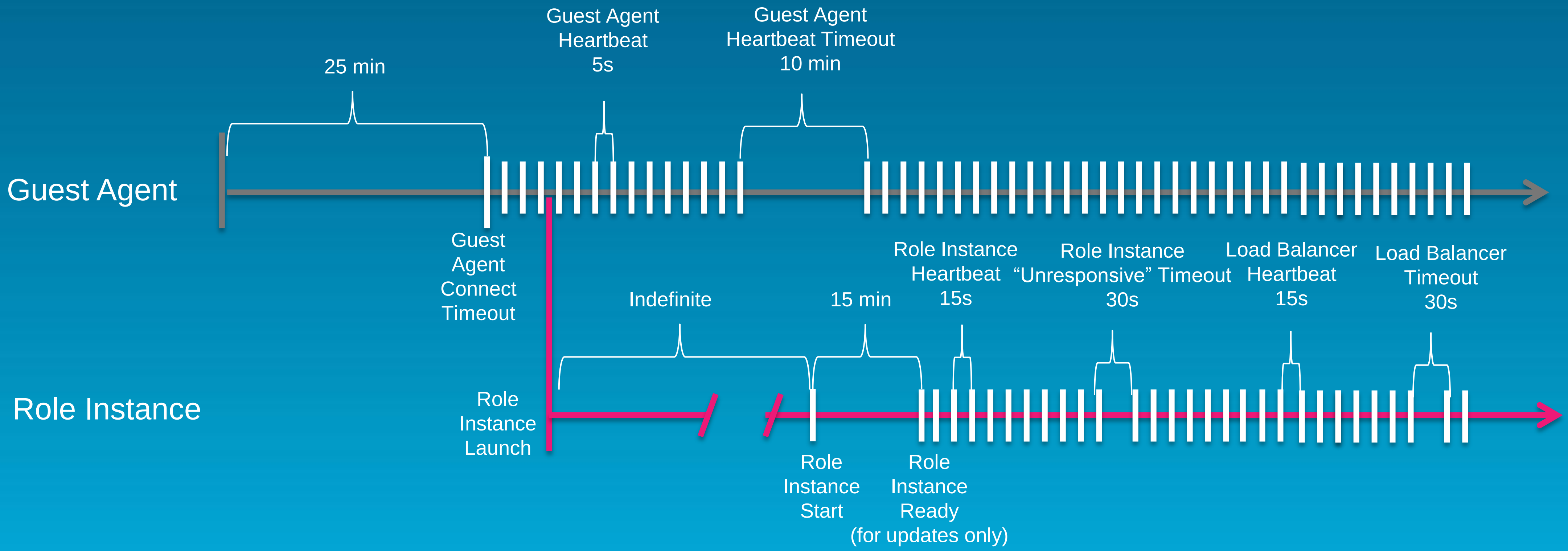
- Introduction to Windows Azure
- Windows Azure Fundamentals
- Fabric Controller Internals
- Deploying a Service
- Updating a Service
- Host OS Upgrades
- Service Health

Node and Role Health Maintenance

- FC maintains service availability by monitoring the software and hardware health
  - Based primarily on heartbeats
  - Automatically “heals” affected roles

Problem	How Detected	Fabric Response
Role instance crashes	FC guest agent monitors role termination	FC restarts role
Guest VM or agent crashes	FC host agent notices missing guest agent heartbeats	FC restarts VM and hosted role
Host OS or agent crashes	FC notices missing host agent heartbeat	Tries to recover node FC reallocates roles to other nodes
Detected node hardware issue	Host agent informs FC	FC migrates roles to other nodes Marks node “out for repair”

Guest Agent and Role Instance Heartbeats and Timeouts



Moving a Role Instance (Service Healing)

- Moving a role instance is similar to a service update
- On source node:
  - Role instances stopped
  - VMs stopped
  - Node reprovisioned
- On destination node:
  - Same steps as initial role instance deployment
- Warning: Resource VHD is not moved

Conclusion

- Platform as a Service is all about reducing management and operations overhead
- The Windows Azure Fabric Controller is the foundation for Windows Azure's PaaS
  - Provisions machines
  - Deploys services
  - Configures hardware for services
  - Monitors service and hardware health
- The Fabric Controller continues to evolve