**Technical Document**

**For**

**ANB Cloud Work streams**

# **Network (Azure Subscription Architecture & Future Service Management)**

Below are few basic information about Windows Azure.

* Cloud infrastructure is shared.
* Cloud infrastructure is built on commodity hardware to achieve best bang-for-buck and it is generally assumed that eventually it will fail.
* A typical cloud application consist of many sub-systems where:
  + Each sub-system is a shared system on its own e.g. Windows Azure Storage.
  + Each sub-system has its limits and thresholds.
* Sometimes individual nodes fail in a datacentre and though very rarely, but sometimes entire datacentre fails.
* You don’t get physical access to the datacentre.
* Understanding latency is very important.

## **Sub work streams tools/technologies**

### **Virtual Machines**

DIP - If you deploy a VM to a virtual network, the VM always receives an internal IP address (DIP) from a pool of internal IP addresses that you specify. VMs communicate within the virtual network by using DIPs. Although Azure assigns the DIP, you can request a static DIP for your virtual machine if you deploy the VM using PowerShell.

VIP - Your VM is also associated with a VIP, although a VIP is never assigned to the VM directly. A VIP is a public IP address that can be assigned to your cloud service. You can, optionally, reserve a VIP for your cloud service.

PIP - Your VM can, optionally, also receive an instance-level public IP address (PIP). The PIP is directly associated with the VM, rather than the cloud service. PIP is currently in Preview.

### **Azure Virtual Network (VNet)**

* Azure Virtual Network easily extends your on-premises network through site-to-site VPN, much the way you’d set up and connect to a remote branch office. You control the network topology, including configuration of DNS and IP address ranges, and manage it just like your on-premises infrastructure
* With Virtual Network, you can build hybrid cloud applications that securely connect to your on-site datacentre—so an Azure web application can access an on-premises SQL Server database or authenticate users against an on-premises Active Directory service.
* Because Virtual Network creates a direct connection between your local development machine and Azure Virtual Machines, you can troubleshoot and debug them using the same tools you would use for on-premises applications.
* Virtual Networks provides an added layer of security. Only virtual machines and services within the same virtual network can identify or connect to each other. Cloud services and virtual machines on a virtual network also avoid the public internet.
* With virtual networks, you can build services that rely on Cloud Services and Virtual Machines. Use Azure web roles for your front end and Virtual Machines for backend databases. Combining Platform-as-a-Service (PaaS) and Infrastructure-as-a-Service (IaaS) within a Virtual Network gives you more flexibility and scalability in building apps.
* If you want to connect to your VMs and cloud services by hostname or SRV records, rather than using the IP address and/or port number, you’ll need name resolution. When you deploy VMs and cloud services to a virtual network you can use Azure-provided name resolution or your own DNS solution, depending on your name resolution requirements.
* The virtual network extends the trust boundary from a single service to the virtual network boundary. You can create several cloud services and virtual machines within a single virtual network and have them communicate with each other without having to go through the internet. You can also setup services that use a common backend database tier or use a shared management service.
* You can join VMs in Azure to your domain running on-premises. You can access and leverage all on-premises investments around monitoring and identity for your services hosted in Azure.
* Virtual machines within a VNet will have a stable private IP address. We assign an IP address from the address range you specify and offer an infinite DHCP lease on it. You can also choose to configure your virtual machine with a specific private IP address from the address range when you create it. This ensures that your virtual machine retains its private IP address even when Stop/Deallocated.
* We can also assign reserved public IP to front end servers. So that it will retain the public IP address even after the reboots.
* A virtual network is limited to a single region. You can create VNet to VNet communication by using REST APIs or Windows PowerShell.
* Up to 128 VPN (Point-2-Site) clients to be able to connect to a virtual network.
* A Maximum of 10 Point-2-site and VNets combined be connect to single virtual network. For example, one Azure virtual network can connect to 6 on-premises sites and 4 virtual networks.

### **Sub Nets Logical**

Subnets are logical grouping of VMs within a Virtual Network. There is no limit on the number of subnets you use within a virtual network. All the subnets must be fully contained in the virtual network address space and should not overlap with one another.

The smallest subnet we support is a /29 (Total 8 IPs and Usable 6 IPs) and the largest is a /8 (Total 8,388,608 IPs and Usable 8,388,606 IPs) (using CIDR subnet definitions). Some IP addresses from each subnet. The first and last IP addresses of the subnets are reserved for protocol conformance. Also additionally reserve a few extra IP addresses for our services.

Subnets can be added to virtual networks at any time as long as the subnet address is not part of another subnet in the virtual network. You can add, remove, expand or shrink a subnet if there are no VMs or services deployed within it by using PowerShell cmdlets or the NETCFG file. You can also add, remove, expand or shrink any prefixes as long as the subnets that contain VMs or services are not affected by the change. You can modify the subnet addresses as long as there are no services or VMs deployed within them by using PowerShell cmdlets or the NETCFG file. You cannot modify or delete a subnet once services or VMs have been deployed to it.

Gateway subnet: gateway service that run to enable cross-premises connectivity. We need 2 IP addresses from your routing domain for us to enable routing between your premises and the cloud. We require you to specify at least a /29 subnet from which we can pick IP addresses for setting up routes. Please note that you must not deploy virtual machines or role instances in the gateway subnet.

Static routing VPNs are also referred to as policy-based VPNs. Policy-based VPNs encrypt and route packets through an interface based on a customer-defined policy. The policy is usually defined as an access list.

Dynamic routing VPNs are also referred to as route-based VPNs. Route-based VPNs depend on a tunnel interface specifically created for forwarding packets. Any packet arriving on the tunnel interface will be forwarded through the VPN connection.

### **Network Security Groups - Isolation**

A network security group (NSG) is a top level object that is associated to your subscription. It can be used to control traffic to one or more virtual machine (VM) instances in your virtual network. An NSG contains access control rules that allow or deny traffic to VM instances. The rules of an NSG can be changed at any time, and changes are applied to all associated instances. A NSG requires a regional VNet. NSGs are not compatible with VNets that are associated with an affinity group.

You can associate an NSG to a VM, or to a subnet within a VNet. When associated with a VM, the NSG applies to all the traffic that is sent and received by the VM instance. When applied to a subnet within your VNet, it applies to all the traffic that is sent and received by ALL the VM instances in the subnet. A VM or subnet can be associated with only 1 NSG, but each NSG can contain up to 200 rules. You can have 100 NSGs per subscription. Endpoint-based ACLs and network security groups are not supported on the same VM instance. You must first remove the Endpoint ACL before associating an NSG.

|  |  |
| --- | --- |
|  |  |

Network security groups are different than endpoint-based ACLs. Endpoint ACLs work only on the public port that is exposed through the Input endpoint. An NSG works on one or more VM instances and controls all the traffic that is inbound and outbound on the VM. A network security group has a Name, is associated to a Region (one of the supported Azure locations), and has a descriptive label. It contains two types of rules, Inbound and Outbound. The Inbound rules are applied on the incoming packets to a VM and the outbound rules are applied to the outgoing packets from the VM. The rules are applied at the server machine where the VM is located. An incoming or outgoing packet has to match an ‘Allow’ rule for it be permitted, if not it will be dropped. Rules are processed in the order of priority. For example, a rule with a lower priority number (e.g. 100) is processed before rules with a higher priority numbers (e.g. 200). Once a match is found, no more rules are processed.

Associating an NSG to a VM - When a NSG is directly associated to a VM, the Network access rules in the NSG are directly applied to all traffic that is destined to the VM. Whenever the NSG is updated for rule changes, the changes are reflected in the traffic handling within minutes. When the NSG is dis-associated from the VM, the state goes back to whatever it was before the NSG, i.e. the system defaults before the introduction if NSG will be used.

Associating an NSG to a Subnet - When a NSG is associated to a subnet, the Network access rules in the NSG are applied to all the VMs in the subnet. Whenever the access rules in the NSG are updated the changes are applied to all Virtual machines in the subnet within minutes.

Associating an NSG to a Subnet and a VM - It is possible that you can associate a NSG to a VM and a different NSG to the subnet where the VM resides. This is supported and in this case the VM gets two layers of protection. On the Inbound traffic the packet goes through the access rules specified in the subnet followed by rules in the VM and in the Outbound case it goes through the rules specified in the VM first before going through the rules specified in the subnet.

### **DNS servers**

We can specify up to 12 DNS servers. We can specify DNS server IP addresses in the virtual network definition. This will be applied as the default DNS server(s) for all virtual machines in the virtual network. You can change the DNS server list for your virtual network at any time. If you change your DNS server list, you will need to restart each of the virtual machines in your virtual network in order for them to pick up the new DNS server.

If your name resolution requirements go beyond the features available from the Azure-provided DNS server, you have the option of using your own DNS server. You may choose to specify a DNS server that is provided by a third-party. An external solution may not support your VMs or role instances. In most cases, an external solution should be avoided except for specific situations where you only need name resolution of external DNS names.

|  |  |
| --- | --- |
| **Scenario** | **Name resolution provided by Azure** |
| Name resolution between role instances located in the same cloud service | Azure-provided name resolution (internal) |
| Name resolution between VMs located in the same cloud service | Azure-provided name resolution (internal) |
| Name resolution between VMs located in the same virtual network | DNS solution of your choice (not Azure-provided)  OR  For resolution using FQDN, you can use Azure-provided name resolution for the first 100 cloud services in the virtual network |
| Name resolution between VMs and role instances located in the same virtual network, but different cloud services | DNS solution of your choice (not Azure-provided)  OR  For resolution using FQDN, you can use Azure-provided name resolution for the first 100 cloud services in the virtual network |
| Name resolution between VMs and role instances located in the same cloud service, not in a virtual network | Not applicable. VMs and role instances cannot be deployed in the same cloud service. |
| Name resolution between role instances located in different cloud services, not in a virtual network | Not applicable. Connectivity between VMs and role instances in different cloud services is not supported outside a virtual network. |
| Cross-premises: Name resolution between role instances or VMs in Azure and on-premises computers | DNS solution of your choice (not Azure-provided) |
| Cross-premises: Name resolution between on-premises computers and role instances or VMs in Azure | DNS solution of your choice (not Azure-provided) |
| Use name resolution to direct traffic between datacentres | Traffic Manager |
| Name resolution between computers on the internet and your public endpoints | Azure-provided name resolution (external) |

### **Multiple subscriptions - Connectivity between subscriptions**

Connecting an Azure virtual network (VNet) to another Azure virtual network is very similar to connecting a virtual network to an on-premises site location. Both connectivity types use a virtual network gateway to provide a secure tunnel using IPsec/IKE. The VNets you connect can be in different subscriptions and different regions. You can even combine VNet to VNet communication with multi-site configurations.

* Cross region geo-redundancy and geo-presence
* You can set up your own geo-replication or synchronization with secure connectivity without going over internet-facing endpoints.
* With Azure Load Balancer and Microsoft or third party clustering technology, you can setup highly available workload with geo-redundancy across multiple Azure regions. One important example is to setup SQL Always On with Availability Groups spreading across multiple Azure regions.
* Regional multi-tier applications with strong isolation boundary
* Within the same region, you can setup multi-tier applications with multiple virtual networks connected together with strong isolation and secure inter-tier communication.
* Cross subscription, inter-organization communication in Azure
* If you have multiple Azure subscriptions, you can now connect workloads from different subscriptions together securely between virtual networks.
* For enterprises or service providers, it is now possible to enable cross organization communication with secure VPN technology within Azure.

**Requirements and Considerations**

* VNet to VNet supports connecting Azure Virtual Networks. It does not support connecting virtual machines or cloud services NOT in a virtual network.
* VNet to VNet requires Azure VPN gateways with dynamic routing VPNs – Azure static routing VPNs are not supported. Connecting multiple Azure virtual networks together does NOT require any on premises VPN gateways, unless cross premises connectivity is required.
* Virtual network connectivity can be used simultaneously with multi-site VPNs, with a maximum of 10 VPN tunnels for a virtual network VPN gateway connecting to ether other virtual networks or on premises sites.
* The address spaces of the virtual networks and on premises local network sites MUST NOT overlap. Overlapping address spaces will cause the creation of virtual networks or uploading netcfg configuration files to fail.
* The virtual networks can be in the same or different subscriptions.
* The virtual networks can be in the same or different Azure regions (locations).
* Redundant tunnels between a pair of virtual networks are not supported.
* A cloud service or a load balancing endpoint CANNOT span across virtual networks even though they are connected together.
* All VPN tunnels of the virtual network, including P2S VPNs, share the available bandwidth on the Azure VPN gateway and the same VPN gateway uptime SLA in Azure.

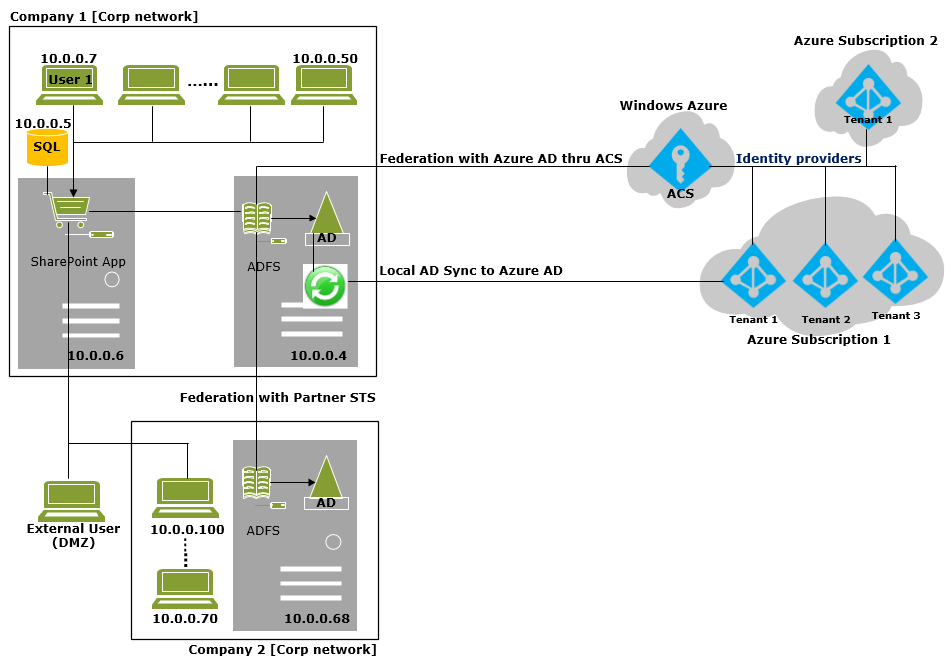
There are 5 sections to plan and configure. Configure each section in the order listed below:

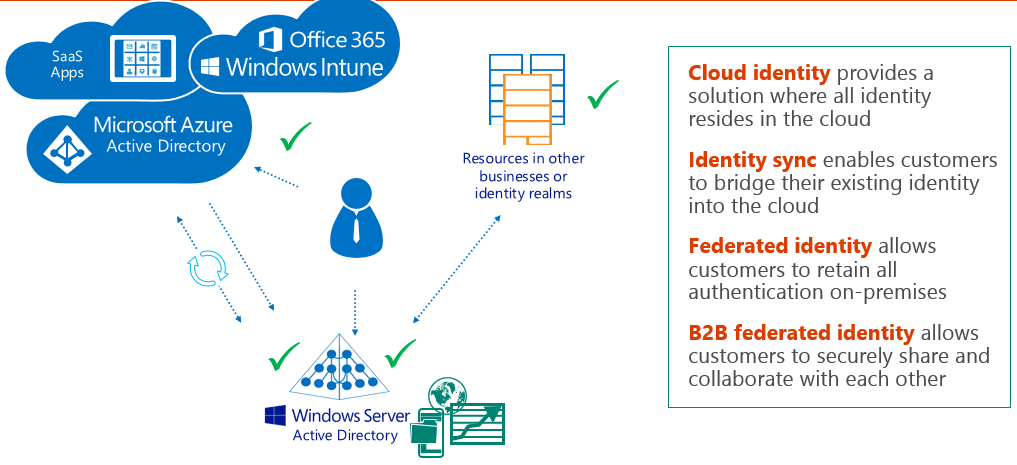
* Plan your IP address ranges
* Create your virtual networks
* Add local networks
* Create the dynamic routing gateways for each VNet.
* Connect the VPN gateways

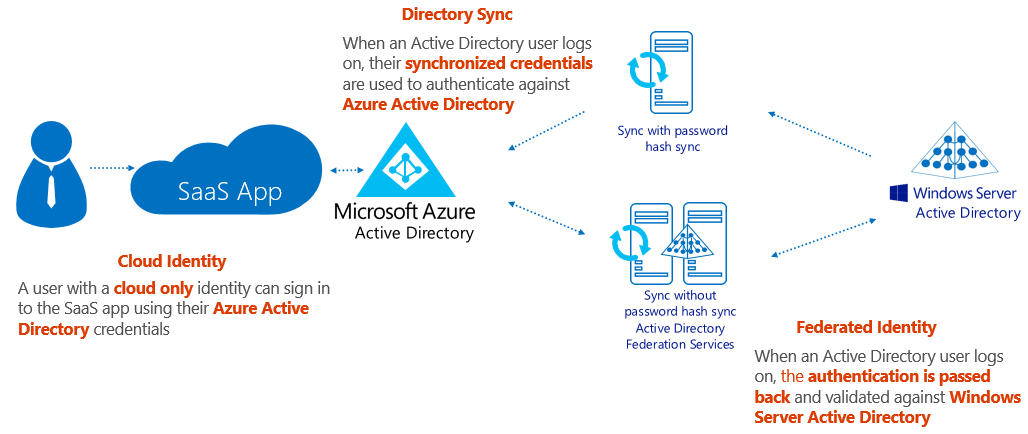
|  |  |
| --- | --- |
|  |  |
|  |  |

### **Azure Single Sign-on - AD**

* Azure Active Directory is a comprehensive identity and access management cloud solution that provides a robust set of capabilities to manage users and groups and help secure access to applications including Microsoft online services like Office 365 and a world of non-Microsoft SaaS applications.
* Enable single sign-on to simplify user access to thousands of cloud applications from Windows, Mac and iOS devices. Users can launch cloud applications from a personalized web-based Access Panel using their company credentials.
* Azure Multi-Factor Authentication prevents unauthorized access to both on-premises and cloud applications by providing an additional level of authentication. Protect your business with security monitoring and alerts and machine learning-based reports that identify inconsistent access patterns to mitigate potential threats.
* Delegate important tasks such as resetting passwords and the creation and management of groups to your employees. Provide Self-service Password Change and Reset and Self-service Group Management with Azure Active Directory Premium.
* Extend Active Directory and any other on-premises directories to Azure Active Directory for single sign on to all cloud-based applications. User attributes can be automatically synchronized to your cloud directory from all kinds of on-premises directories.
* Azure Active Directory Premium can provide the level of scale and reliability required by enterprises. Azure Active Directory is a high available service hosted in globally distributed datacentres. The service handles billions of authentications each day from more than 200 million active users and provides a 99.9 percent SLA.
* Azure Active Directory offers developers an effective way to integrate identity management in their applications. Industry standard protocols such as SAML 2.0, WS-Federation, and OpenID Connect makes sign-in possible on a variety of platforms such as .Net, Java, Node.js, and PHP. The REST-based Graph API enables developers to read and write to the directory from any platform. Through support for OAuth 2.0, developers can build mobile and web applications that integrate with Microsoft and third party web APIs, and build their own secure web APIs. Open source client libraries are available for .Net, Windows Store, iOS and Android with additional libraries under development.







|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

### **Role Based Access**

Every Azure subscription is associated with an Azure Active Directory. Users and services that access resources of the subscription using Azure Management portal or Azure Resource Manager API first need to authenticate with that Azure Active Directory.

Azure role-based access control allows you to grant appropriate access to Azure AD users, groups, and services, by assigning roles to them on a subscription or resource group or individual resource level. The assigned role defines the level of access that the users, groups, or services have on the Azure resource.



# **Built-in Roles**

At the first preview, Azure role-based access control comes with three built-in roles that can be assigned to users, groups, and services.

* **Owner**: has full control over Azure resources. Owner can perform all management operations on a resource including access management.
* **Contributor**: can perform all management operations except access management. So, a contributor can’t grant access to others.
* **Reader**: can only view resources. Reader can’t view secrets associated with a resource.

You can’t modify the definition of built-in roles. In an upcoming release of Azure RBAC, you will be able to define custom roles by composing a set of actions from a list of available ones that can be performed on Azure resources.

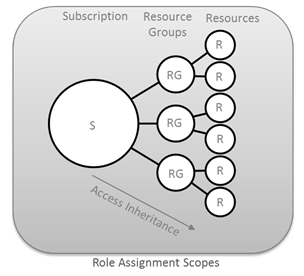
# **Actions and non-actions**

The actions property of the role definition specifies the allowed actions on Azure resources. Action strings can use wildcard characters. The not actions property of the role definition specifies the actions that must be excluded from the allowed actions.

|  |  |  |
| --- | --- | --- |
| **Built-in Role** | **Actions** | **Not Actions** |
| Owner (allow all actions) | \* | NA |
| Contributor (allow all actions except writing or deleting role assignments) | \* | Microsoft.Authorization/ \* /Write, Microsoft.Authorization/ \* /Delete |
| Reader (allow all read actions) | \*/Read | NA |

# **Resource scope**

Access does not need to be granted to the entire subscription. Roles can also be assigned for resource groups as well as for individual resources. In Azure RBAC, a resource inherits role assignments from its parent resources. So if a user, group, or service is granted access to only a resource group within a subscription, they will be able to access only that resource group and resources within it, and not the other resources groups within the subscription. As another example, a security group can be added to the Reader role for a resource group, but be added to the Contributor role for a database within that resource group.



# **Authorization for management versus data operations**

Role-based access control is supported only for management operations of the Azure resources in Azure Preview portal and Azure Resource Manager APIs. Not all data level operations for Azure resources can be authorized via RBAC. For instance, create/read/update/delete of Storage Accounts can be controlled via RBAC, but create/read/update/delete of blobs or tables within the Storage Account cannot yet be controlled via RBAC. Similarly, create/read/update/delete of a SQL DB can be controlled via RBAC but create/read/update/delete of SQL tables within the DB cannot yet be controlled via RBAC.

### **Best practices**

**Protection against Hardware Issues**

* Deploy multiple instances of your application.
* Scale out instead of scale up or in other words favour horizontal scaling over vertical scaling. It is generally recommended that you go with smaller sized Virtual Machines (VM) instead of few larger sized VMs unless you have a specific need for larger sized VMs.
* Don’t rely on VM’s local storage as it is transient and not fail-safe. Use persistent storage like Windows Azure Blob Storage instead.
* Build decoupled applications to safeguard your application against hardware failures.

**Cloud Services Development**

* It is important to understand what web role and worker role are and what benefit they offer. Choose wisely to distribute functionality between a web role and worker role.
* Decouple your application logic between web role and worker role.
* Build stateless applications. For state management, it is recommended that you make use of distributed cache.
* Identify static assets in your application (e.g. images, CSS, and JavaScript files) and use blob storage for that instead of including them with your application package file.
* Make proper use of service configuration / app.config / web.config files. While you can dynamically change the values in a service configuration file without redeploying, the same is not true with app.config or web.config file.
* To achieve best value for money, ensure that your application is making proper use of all VM instances in which it is deployed.

**General Recommendations**

* Blob/Table/SQL Database – Understand what they can do for you. For example, one might be tempted to save images in a SQL database whereas blob storage is the most ideal place for it. Likewise one could consider Table storage over SQL database if transaction/relational features are not required.
* It is important to understand that these are shared resources with limits and thresholds which are not in your control i.e. you don’t get to set these limits and thresholds.
* It is important to understand the scalability targets of each of the storage component and design your application to stay within those scalability targets.
* Be prepared that you’ll encounter “transient errors” and have your application handle (and recover from) these transient errors.
* It is recommended that your application uses retry logic to recover from these transient errors.
* You can use TOPAZ or Storage Client Library’s built-in retry mechanism to handle transient errors. If you don’t know, TOPAZ is Microsoft’s Transient Fault Handling Application Block which is part of Enterprise Library 5.0 for Windows Azure. You can read more about TOPAZ here: http://entlib.codeplex.com/wikipage?title=EntLib5Azure.
* For best performance, co-locate your application and storage. With storage accounts, the cloud service should be in the same affinity group while with WASD, the cloud service should be in the same datacentre for best performance.
* From disaster recovery point of view, please enable geo-replication on your storage accounts.

**Windows Azure SQL Database (WASD)**

* It is important to understand that it’s a shared resource. So expect your requests to get throttled or timed out.
* It is important to understand that WASD != On Premise SQL Server. You may have to make some changes in your data access layer.
* It is important to understand that you don’t get access to data/log files. You will have to rely on alternate mechanisms like “Copy Database” or “BACPAC” functionality for backup purposes.
* Prepare your application to handle transient errors with WASD. Use TOPAZ for implementing retry logic in your application.
* Co-locate your application and SQL Database in same data centre for best performance.

**Windows Azure Storage (Blobs, Tables & Queues)**

* It is important to understand that it’s a shared resource. So expect your requests to get throttled or timed out.
* Understand the scalability targets of Storage components and design your applications accordingly.
* Prepare your application to handle transient errors with WASD. Use TOPAZ or Storage Client library’s Retry Policies for implementing retry logic in your application.
* Co-locate your application and storage account in same affinity group (best option) or same data centre (next best option) for best performance.
* Table Storage does not support relationships so you may need to de-normalize the data.
* Table Storage does not support secondary indexes so pay special attention to querying data as it may result in full table scan. Always ensure that you’re using PartitionKey or PartitionKey/RowKey in your query for best performance.
* Table Storage has limited transaction support. For full transaction support, consider using Windows Azure SQL Database.
* With Table Storage, pay very special attention to “PartitionKey” as this is how data in a table is organized and managed.

**Managing Latency**

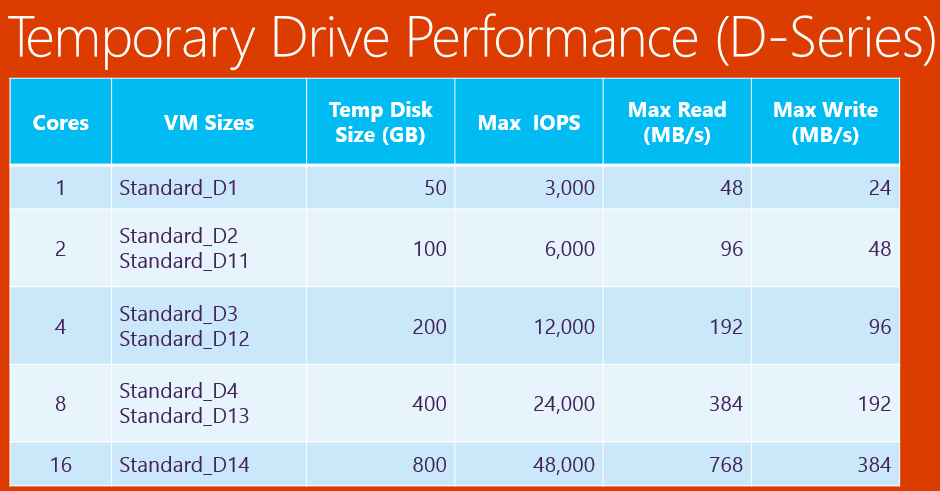
* Co-locate your application and data stores. For best performance, co-locate your cloud services and storage accounts in the same affinity group and co-locate your cloud services and SQL database in the same data centre.
* Make appropriate use of Windows Azure CDN.
* Load balance your application using Windows Azure Traffic Manager when deploying a single application in different data centres.

**Availability Set Guidance**

* VMs in Availability Set Must Be in Same Cloud Service
* Availability Set: 5 Update Domains, 2 Fault Domains
  + Update Domain – Host Maintenance
  + Fault Domain – Isolation from component failure in rack unit
* Maximum of 50 VMs in a Availability Set
* Avoid Availability Sets with Single VM
  + This eliminates notification for host maintenance operations

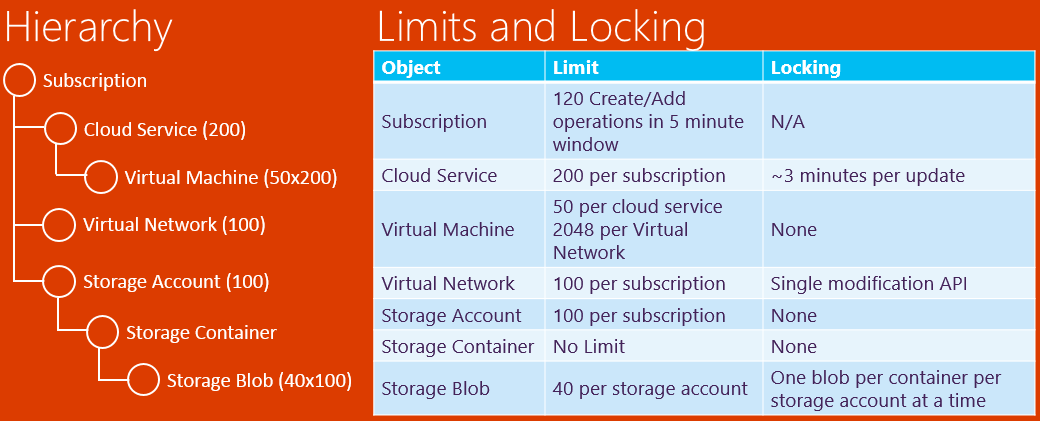
**Temporary Drive Guidance**

* Never Place Critical Unreplicated Data on Temp Drive!!
* Use for SQL TempDB and Buffer Pool Extension on D-Series and G-Series VM Sizes Only (SSD Temp Disks)
* Use Scheduled Tasks to Configure Temporary Disk
* Test Scheduled Tasks via Resize VM Operation



**Large Deployment Guidance**

* Deploy VMs in Blocks
  + Block size: Minimum of Total VMs / 5 -or- 100
  + Deploy one VM per cloud service; use common availability set in each cloud service
  + Wait 5 minutes between starting each block of VMs to avoid throttling
* Limit use of Get\* APIs During Deployment
  + Limits possibility of throttling
  + Back-off if any 503-ServiceUnavailable or 429-TooManyRequests errors are received
* Use Virtual Network for Connectivity
* Custom Image Management
  + If using a custom OS image then place up to 40 OS disks in a storage account



**Azure Scale Unit (Azure Compute Cluster)**

* Definition
* A compute unit capable of supporting a defined range of VM sizes
* Each Cloud Service is bound to a single Scale Unit
* Each Affinity Group with one or more VM is bound to a single Scale Unit
* Impact of Resize
* VMs can only be resized to a size supported on Scale Unit where the VM is deployed
* Current VM Sizes Supported on Scale Units
* Scale Unit 1: A0 – A4
* Scale Unit 2: A0 – A7
* Scale Unit 3: A8/A9 Only
* Scale Unit 4: A0 – A7 and D1 – D14
* Scale Unit 5: G1 – G5 (Coming Soon)

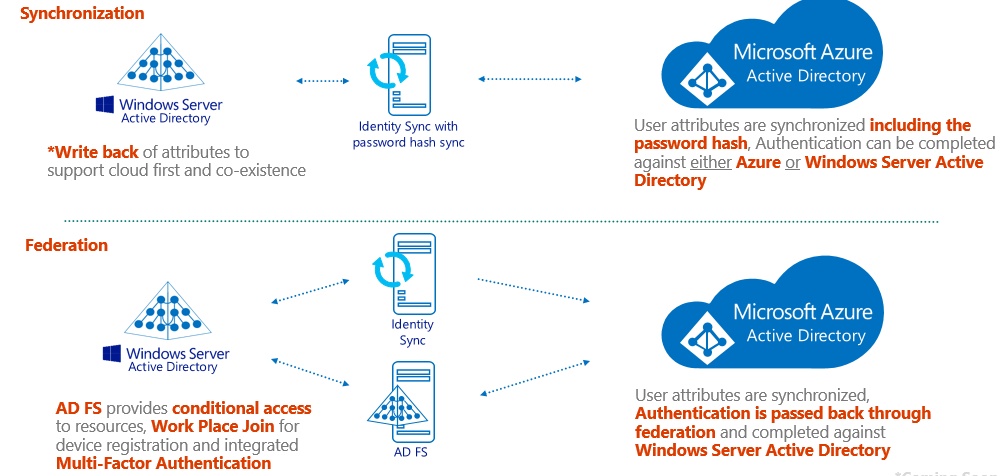
## **Automation Opportunities**

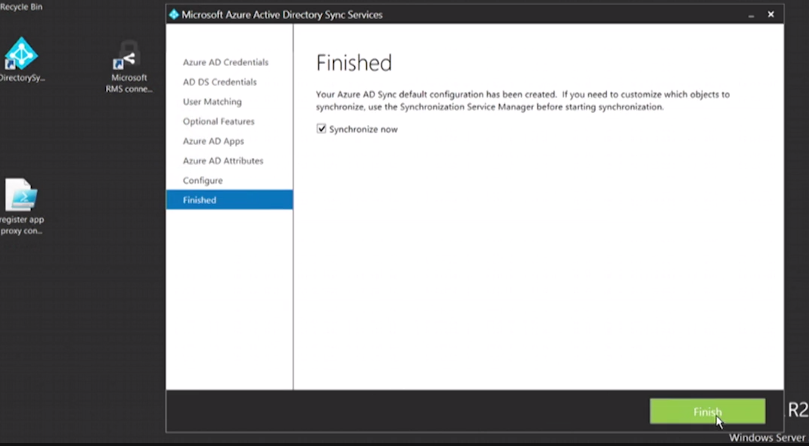
### **DirSync**

DirSync and Active Directory Federation Services are two options to connect both. DirSync can now be used as a backup for ADFS. DirSync is a software tool used to synchronize objects located in an on-premises, single forest Active Directory to Azure Active Directory. Azure Active Directory is the Microsoft multi-tenant cloud version of Active Directory used for identity management for services like Office 365. DirSync is basically an implementation of Forefront Identity Manage but with limited features. For example it is not able to sync objects of multiple on-premises AD forests nor is it able to handle multiple Exchange organizations.

To support these scenarios enterprises are at the moment required to use Forefront Identity Manager. However, configuring FIM can be challenging and can take considerable time. The new tool named Azure Active Directory Sync Services or AADSync. AADSync significantly simplifies the configuration and makes it more predictive.

Microsoft Azure Active Directory Sync Services (AADSync) is used to onboard an on-prem environment to Windows Azure Active Directory and Office 365 and continue to synchronize changes. It is used for more advanced scenarios where DirSync does not provide support, for example multiple on-prem AD forests. At the moment AADSync does not support multiple Azure subscriptions. AADSync will also be able to synchronize Exchange Global Address Lists. Support for PowerShell is also available, it has about 58 commands.





### **Setup VNET**

Power Shell/Run book scripts will be used for automation.

### **Setup Subnets**

Power Shell/Run book scripts will be used for automation.

### **Network Security Groups**

Power Shell/Run book scripts will be used for automation.

## **Processes**

Process Diagrams to be done.

# **Network (Hybrid connectivity)**

## **Sub work streams tools/technologies**

**Express Route considerations**

Azure ExpressRoute enables you to create private connections between Azure datacenters and infrastructure that’s on your premises or in a colocation environment. ExpressRoute connections do not go over the public Internet, and offer more reliability, faster speeds, lower latencies and higher security than typical connections over the Internet. In some cases, using ExpressRoute connections to transfer data between on-premises and Azure can also yield significant cost benefits. With ExpressRoute, you can establish connections to Azure at an ExpressRoute location (Exchange Provider facility) or directly connect to Azure from your existing WAN network (such as a MPLS VPN) provided by a network service provider.

**Storage, Backup and Recovery**

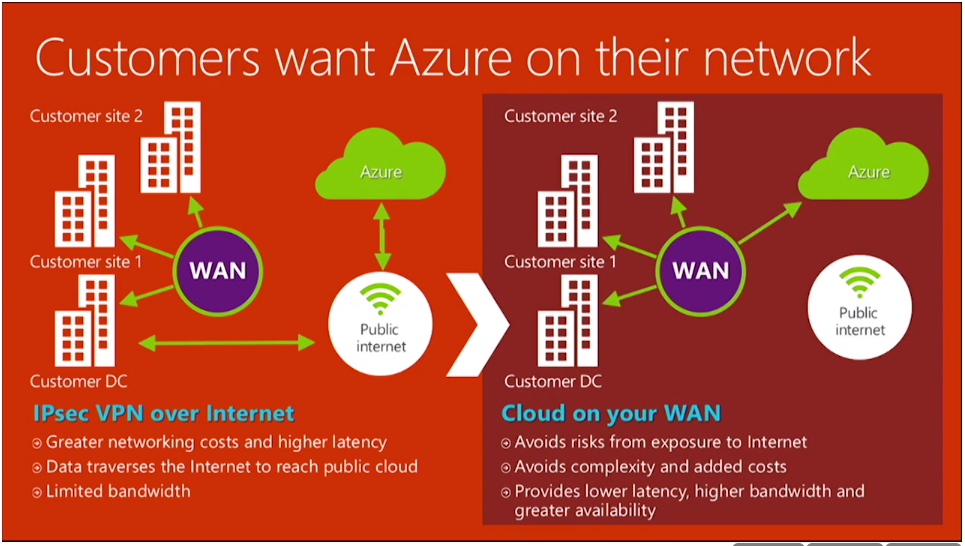
ExpressRoute gives you a fast and reliable connection to Azure making it suitable for scenarios like periodic data migration, replication for business continuity, disaster recovery and other high availability strategies. It can also be a cost-effective option for transferring large amounts of data such as datasets for high performance computing applications or moving large VMs between your dev/test environment in Azure and on-premises production environment.

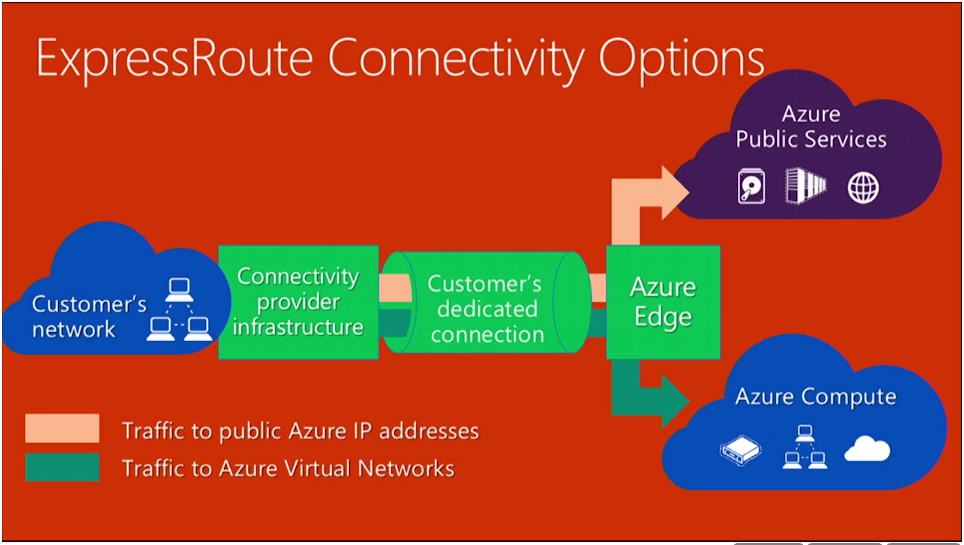
**Extend your datacenter**

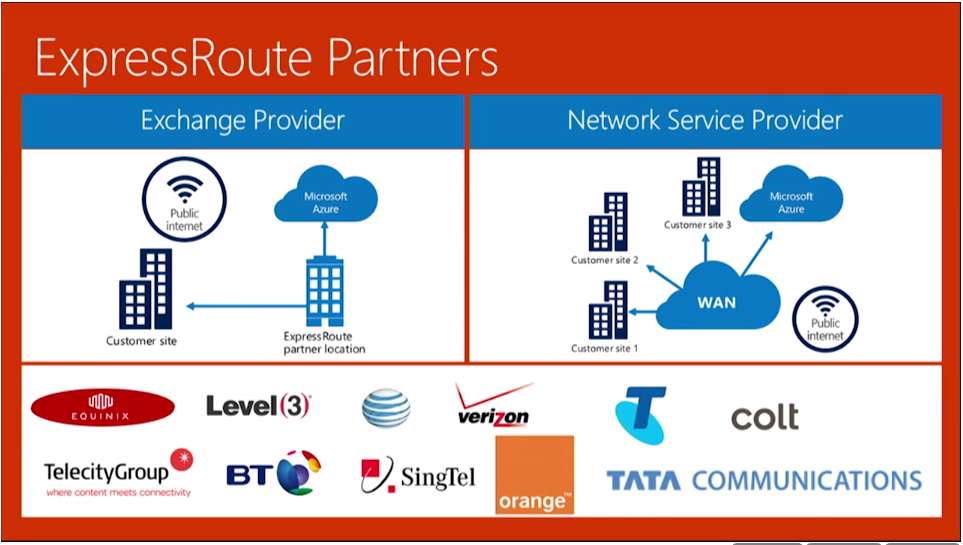
ExpressRoute allows you to securely add compute and storage capacity to your existing datacenter. With high throughput and fast latencies, Azure will feel like a natural extension to your datacenter so you enjoy the scale and economics of the public cloud without having to compromise on network performance.

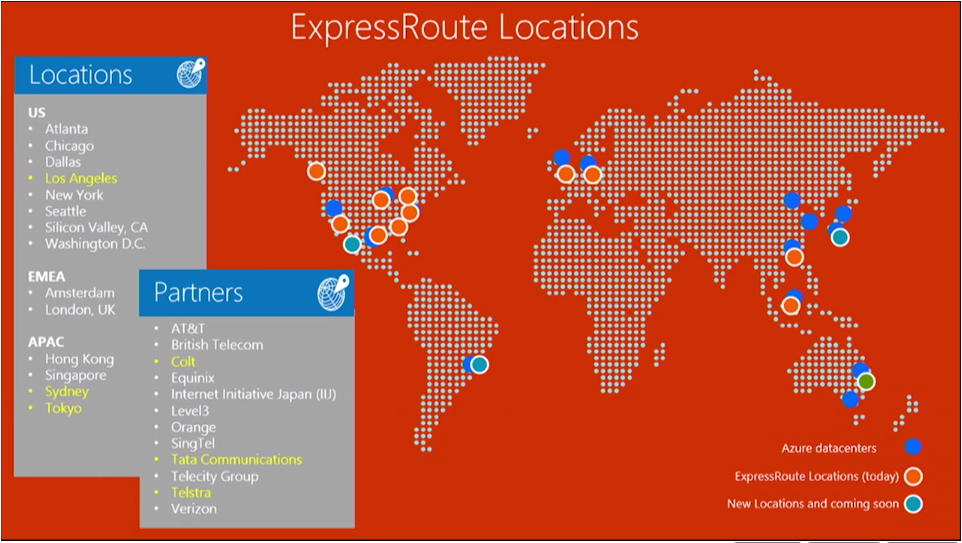
**Build hybrid applications**

With predictable, reliable and high-throughput connections offered by ExpressRoute, you can build applications that span on-premises infrastructure and Azure without compromising security or performance. For example, you can have your corporate intranet application running in Azure that authenticates users using an on-premises Active Directory, and serve all corporate users without traffic ever routing through the public Internet.









**MPLS Latency**

Windows Azure ExpressRoute enables enterprises extend their private networks to Windows Azure using fast, low-latency network connections. ExpressRoute enables you to establish dedicated, private, high-throughput network connectivity between your on-premises IT infrastructure and Windows Azure. ExpressRoute leverages the network services offered by telecommunications partners like Level 3, AT&T, and Equinix to extend your existing networks to the Microsoft's Windows Azure infrastructure without having to route your private network traffic over the public Internet.

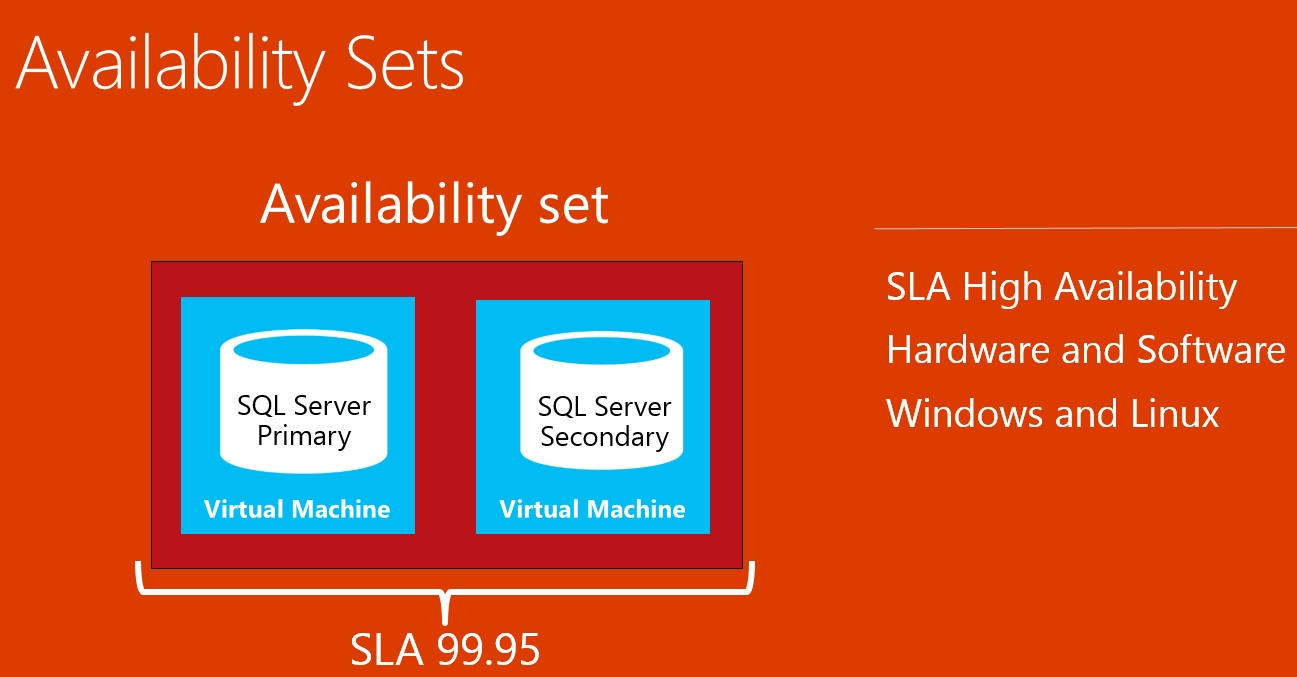
To take advantage of Windows Azure ExpressRoute, you can aggregate your network traffic over Equinox datacenters or you can add Azure services to your AT&T MPLS (multiprotocol label switching) VPN. Level 3 allows you to choose either an aggregated connection or the use of an MPLS VPN. Windows Azure's web management interfaces CREATE A VIRTUAL NETWORK site-to-site configuration options. ExpressRoute enables virtual machine (VM) and storage on Windows Azure to be used like it was a part of your organization's internal resources

DR

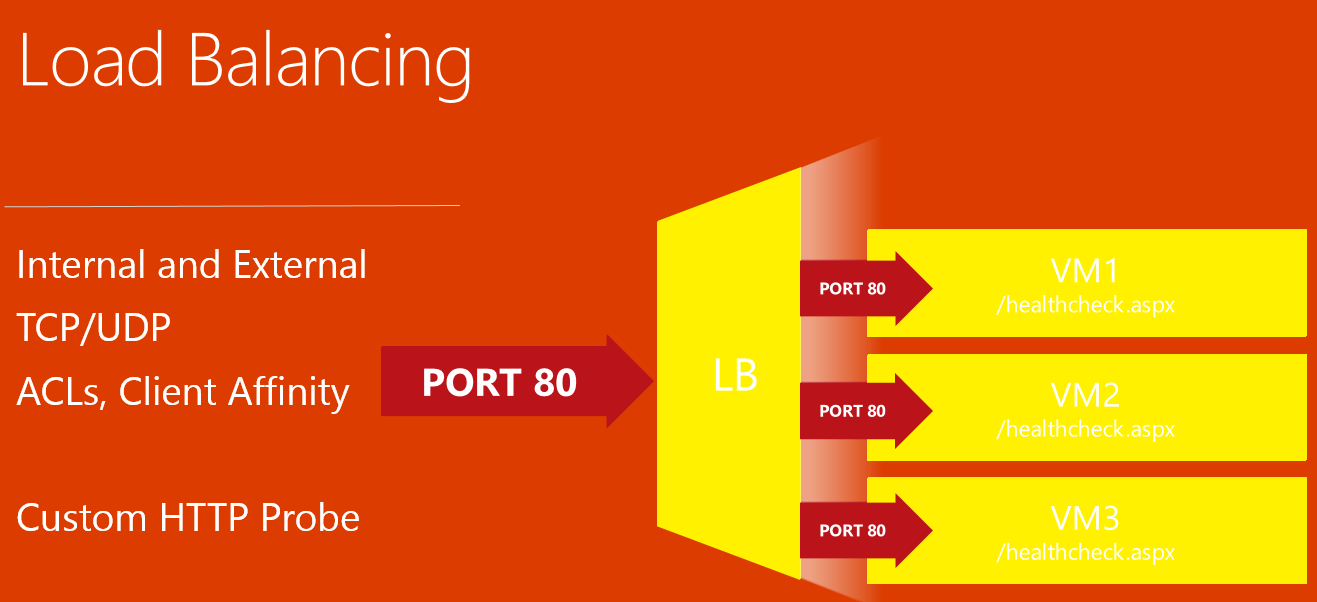
To be done.

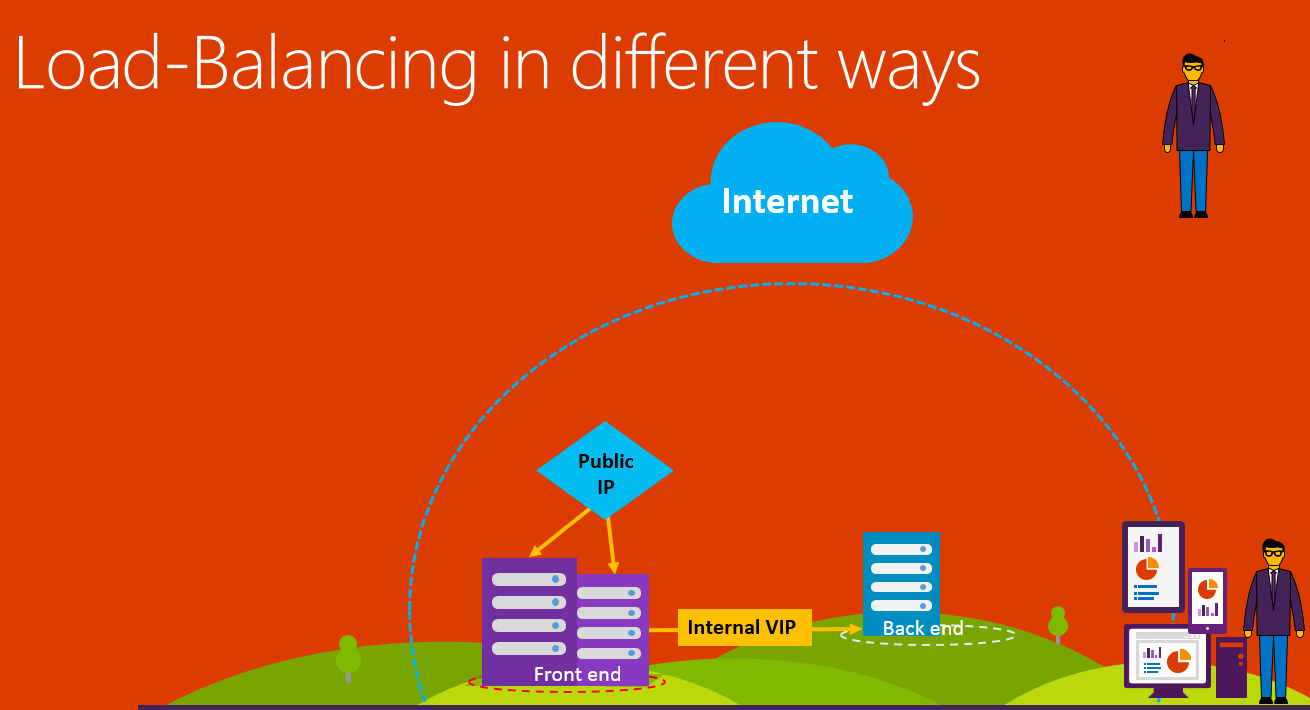
HA

Discussed below.









## **Automation Opportunities**

Yet to be identified

## **Processes**

Process Diagrams to be done.

# **Application Architecture & App Security**

## **Sub work streams tools/technologies**

**High Availability**

Latency

To be done.

Resiliency - Web Role, Azure Redis Cache, SQL Database

To be done.

## **Automation Opportunities**

Can we automate HA, Resiliency?

## **Processes**

Process Diagrams to be done.

# **Application Architecture on PaaS**

## **Sub work streams tools/technologies**

Continuous Integration

To be done.

BDD (User journey and acceptance)

To be done.

Unit Testing and Integration

To be done.

## **Automation Opportunities**

Kickoff tests upon check-in

## **Processes**

Process Diagrams to be done.

# **ANB Customer Application - Code Gap analysis to enable PaaS / Iaas**

## **Sub work streams tools/technologies**

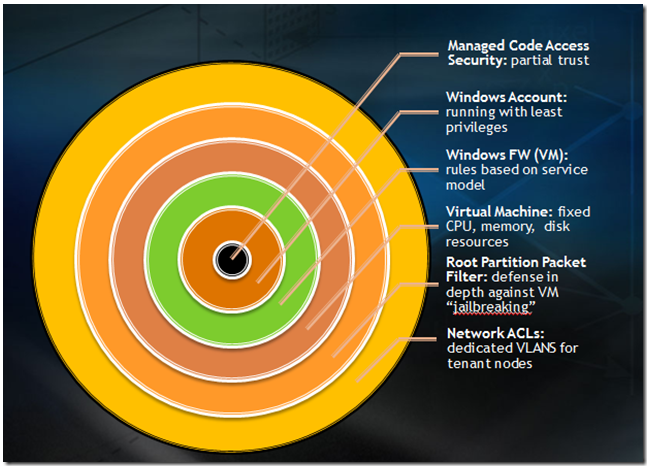
Code Refactoring

To be done.

Security checks for the cloud

To be done.

Defense in depth



Encryption

To be done.

Fail Fast

To be done.

Web Vulnerability scanning

To be done.

PCI analysis

To be done.

Static Analysis

To be done.

## **Automation Opportunities**

Yet to be identified

## **Processes**

Process Diagrams to be done.

# **ANB Performance Engineering**

## **Sub work streams tools/technologies**

VS Load Testing

To be done.

App Insight

To be done.

## **Automation Opportunities**

Automation Web Test / Nightly Builds

Generate reports

Machine learning (performance analysis)

## **Processes**

Process Diagrams to be done.

# **Release & Change Management**

## **Sub work streams tools/technologies**

Release Management

The faster your software is deployed, the quicker you can get feedback. With release management in Visual Studio you can configure, approve and deploy your applications for any environment. Create automated deployment orchestrations for each environment no matter how complex the configuration. Delivering your software more frequently and easily to an environment allows your testers to get to work validating your system and keeps your stakeholders involved in giving feedback.

**Automate multi-stage deployments**

* Visually create the configuration paths that describe the promotion path through your environments.
* Automatically trigger releases upon completion of builds.
* One click to deploy to all environments of the pipeline.
* Deploy to Windows or Linux servers using PowerShell, Desired State Configuration, Chef, or Visual Studio Deployment Agent.
* Run automated validation tests in each stage

**Automate approval workflows**

* Define pre-deployment or post deployment approvals for multiple approvers or teams.
* Enable approvers to schedule each promoted release at a time that suits them, providing the versatility of automation while still maintaining control.
* Get approval notifications and provide approval using a web page instead of the Release Management client.

**Deploy the same way to all stages**

* Use the same deployment artifacts for every environment, including deploying to production.
* Test the same deployment sequence multiple times before going to production.
* Centrally manage configurations that vary from one environment to another in the release pipeline.

**Retain full traceability**

* Retain full traceability of everything that happens to a release with release logs and approval tracking.
* Understand the user stories or bug fixes that went into releases.
* Audit release trails anytime to understand all aspects of releases.

**Manage and control release security**

* Avoid intrusions or accidental deployments by properly managing security of pipeline.
* Assign roles and responsibilities to the right groups for releases, definitions and release artefacts.

**Deploy to on-premises and Azure with same ease**

* Deploy to on-premises or Azure environments.
* Test on Azure and release to production on-premises.
* Import and manage your Azure environments from within Release Management.
* Manage passwords and configurations for on-premises and Azure environments centrally.

**Extend Release Management with customization**

* Use custom Desired State Configuration and PowerShell scripts for deployment.
* Add custom tools and actions.
* Release components from TFS and Visual Studio online builds, or use REST APIs to release components from your own build systems.

**Fail elegantly using Rollbacks**

* Manage deployment failures gracefully using rollbacks.
* Always leave an environment in a stable state, even in cases of unknown failures.
* Easily diagnose failures using logs collected from all servers.

Change Management

To be done

## **Automation Opportunities**

Yet to be identified

## **Processes**

Process Diagrams to be done.

# **Life cycle Maintenance**

## **Sub work streams tools/technologies**

Azure Guest OS family

To be done.

Rolling upgrades

To be done.

Upgrade domains

To be done.

Prod / Staging - Web Roles

To be done.

Chef for Deployment

To be done.

PowerShell Desired State Configuration

To be done.

Production Parallel

To be done.

## **Automation Opportunities**

Yet to be identified

## **Processes**

Process Diagrams to be done.

# **Monitoring & Alerting Guidance**

## **Sub work streams tools/technologies**

Monitoring and alerting

To be done.

Azure Infrastructure Alerts

To be done.

Azure Application Alerts

To be done.

## **Automation Opportunities**

Setup alerts

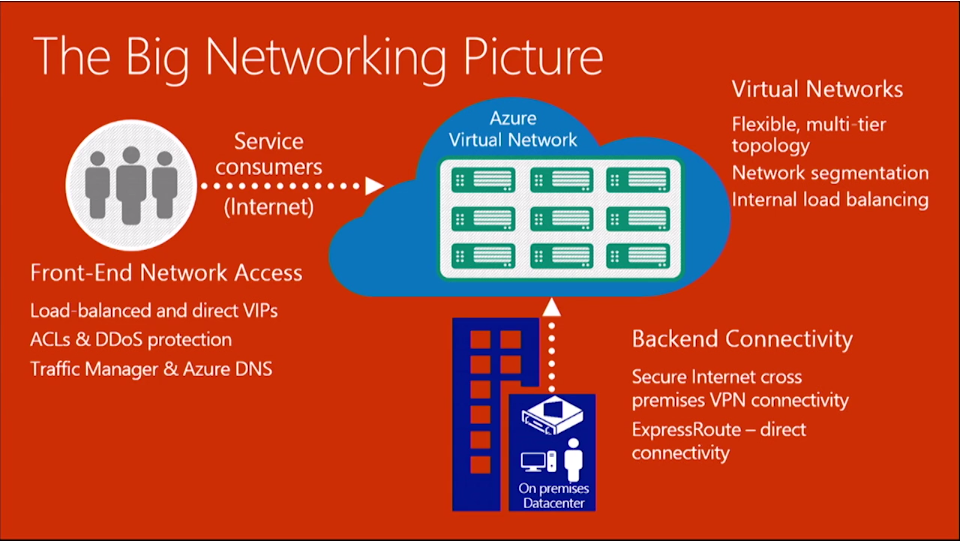
Change alert thresholds

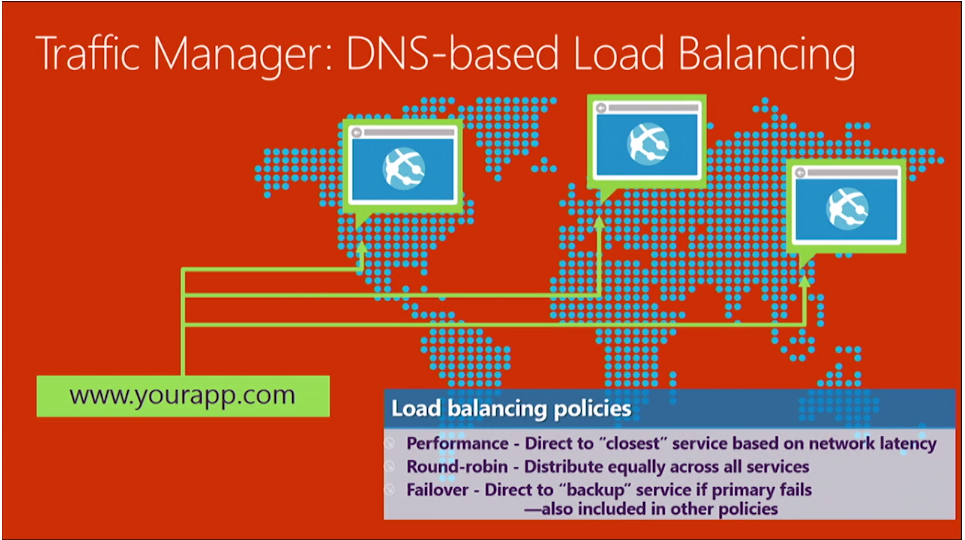
## **Processes**

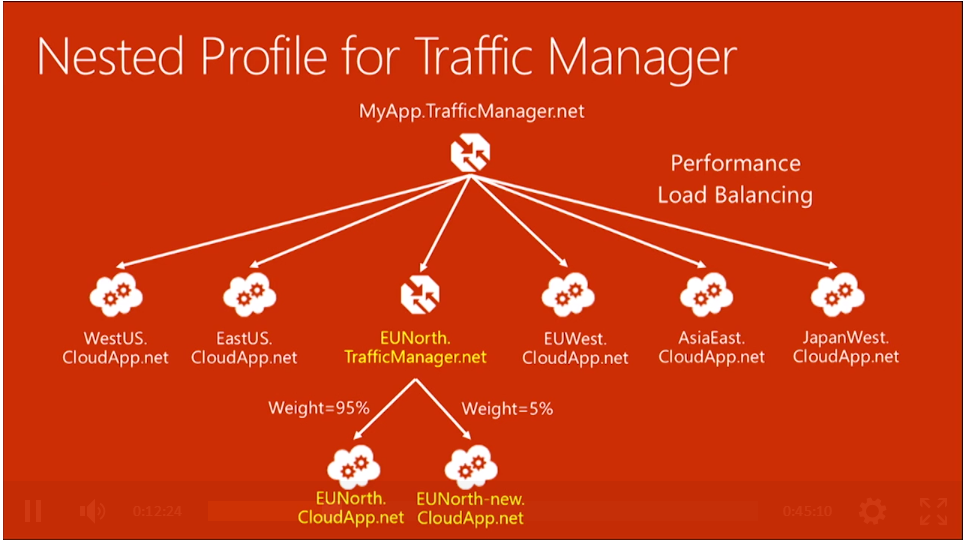
Process Diagrams to be done.

# **Glossary**

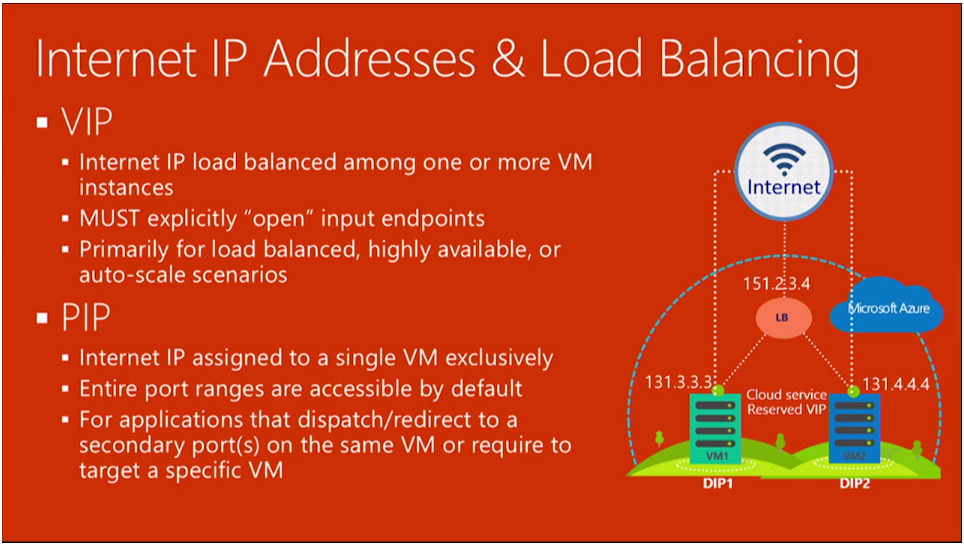
|  |  |
| --- | --- |
| **Term** | **Description** |
| **BDD** | Behavioural Driven Development |
| **TDD** | Test Driven Development |
| **PCI** | Payment Card Industry |
| **NSG** | Network Security Group |
| **ACL** | Access Control List |
| **DMZ** | Demilitarized Zone |
| **CIDR** | Classless InterDomain Routing |
| **MPLS** | Multiprotocol Label Switching |
|  |  |

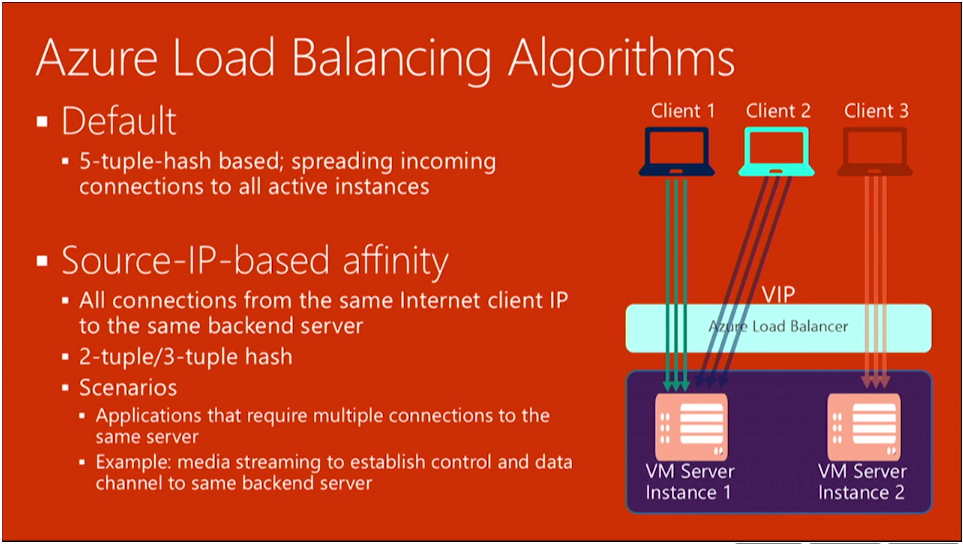


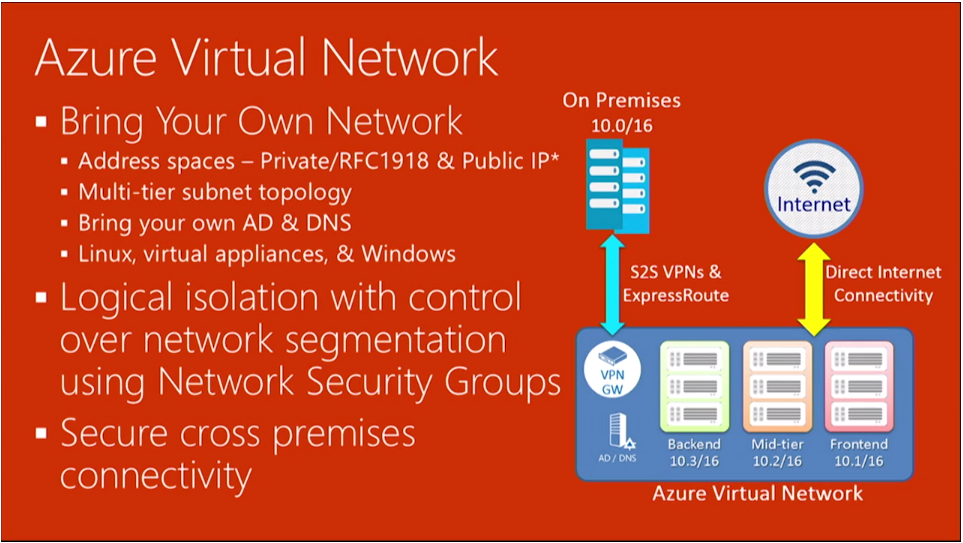




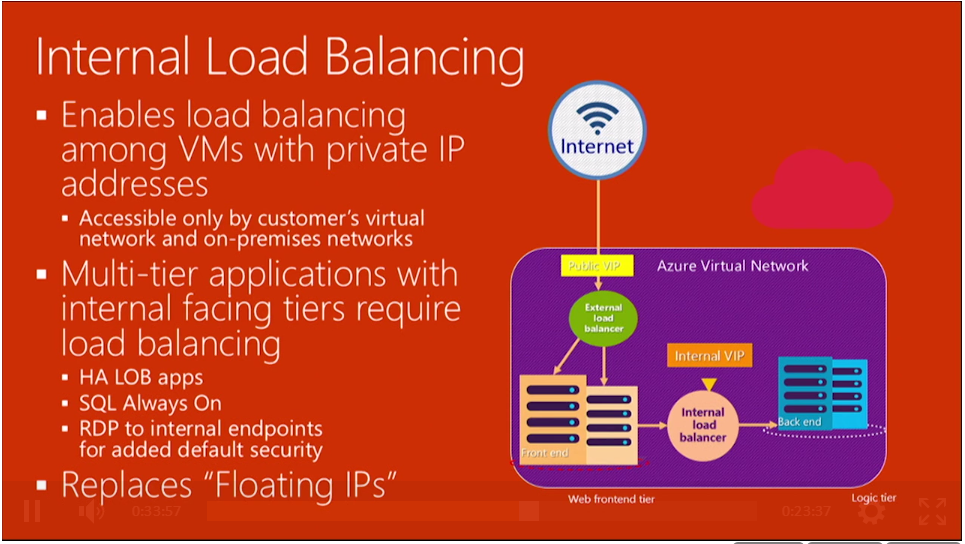
Power Shell; API

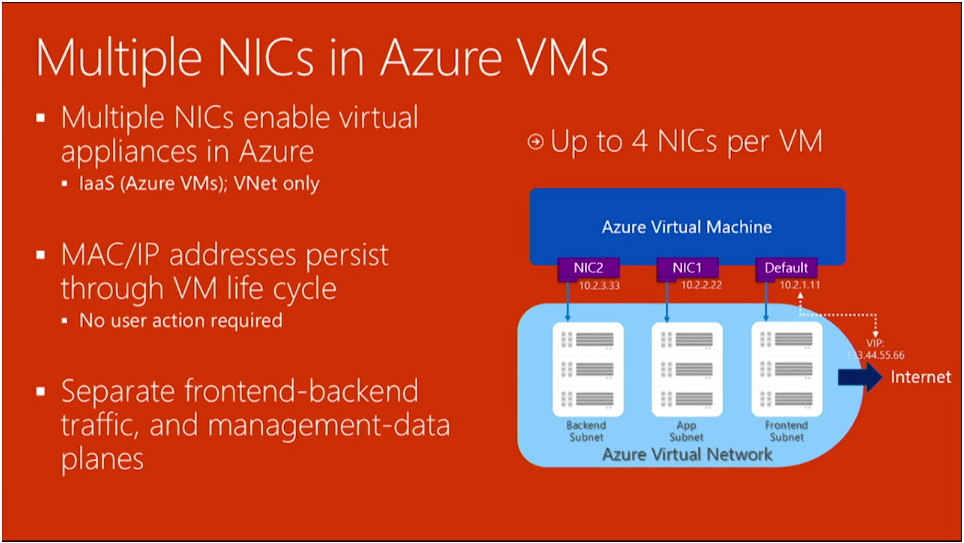


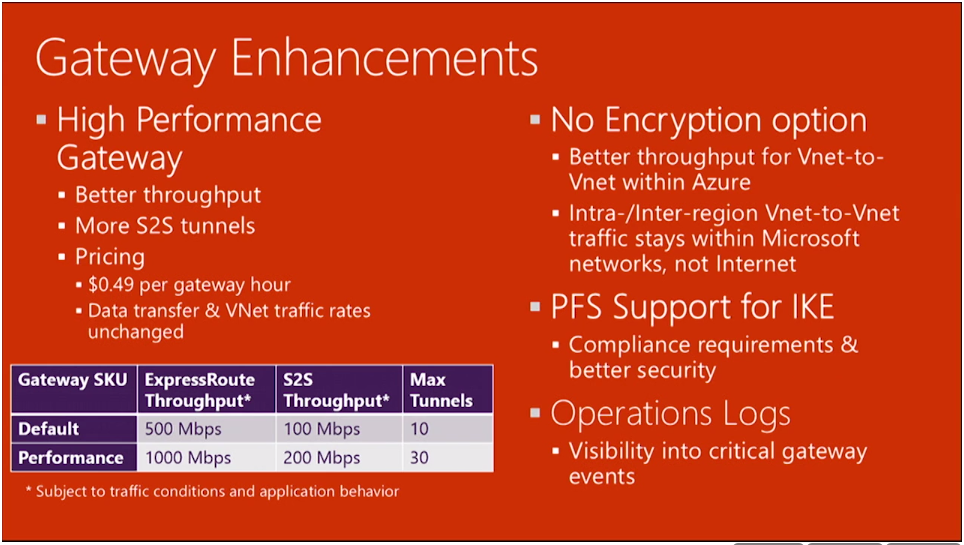




**Only using Power Shell and API**







Only thru Power Shell

**Virtual Machines:**

32 data disks