Microsoft

Azure Solutions Architect – AZ303

Introductions

- What’s your name?
- What do you do?
- What do you hope to get out of this class?

Operations

- 9am – 5pm everyday
- Lunch around noon
- Snacks and drinks in the breakroom

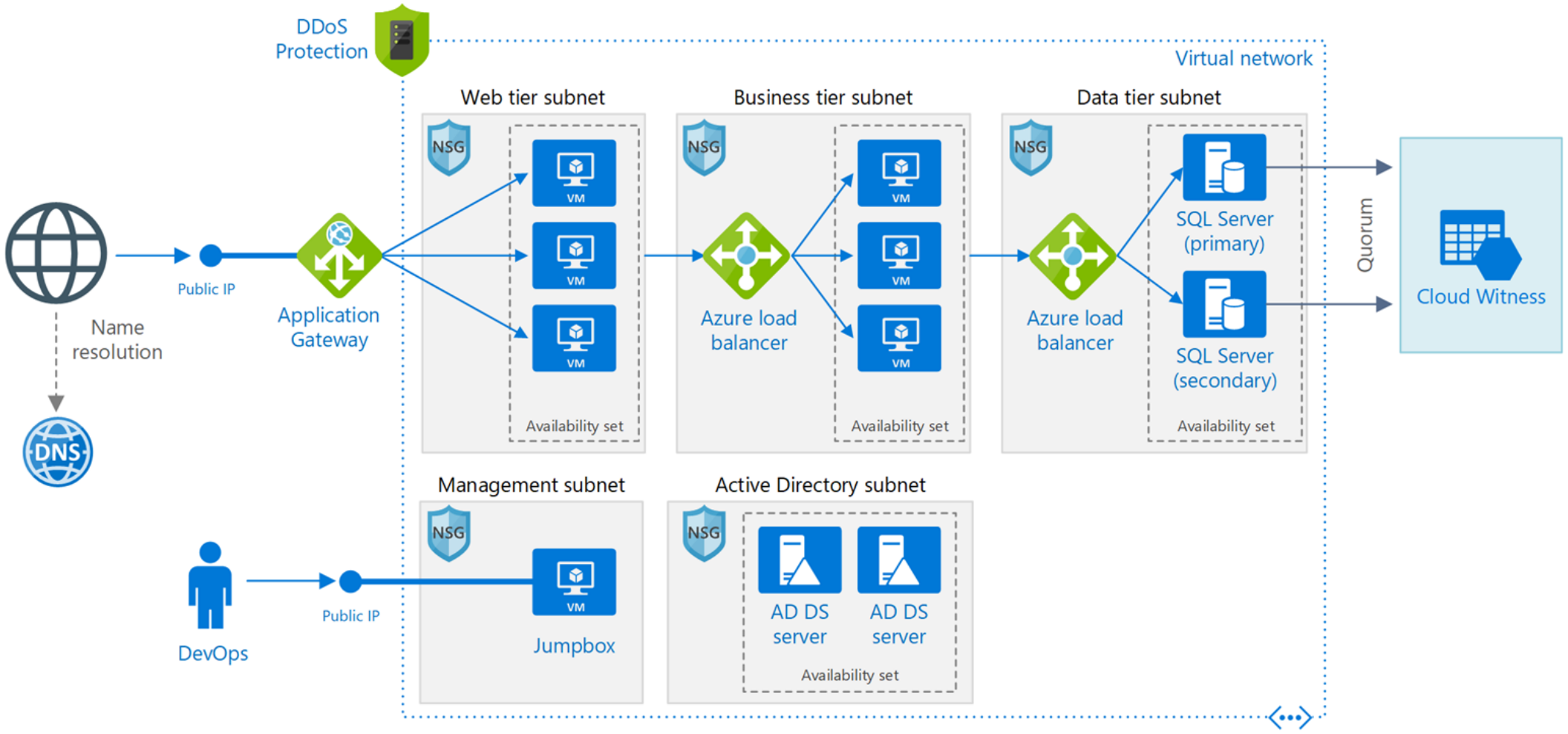
Certification

- Pass AZ-300
- Pass AZ-301
- Become an Azure Certified Solutions Architect

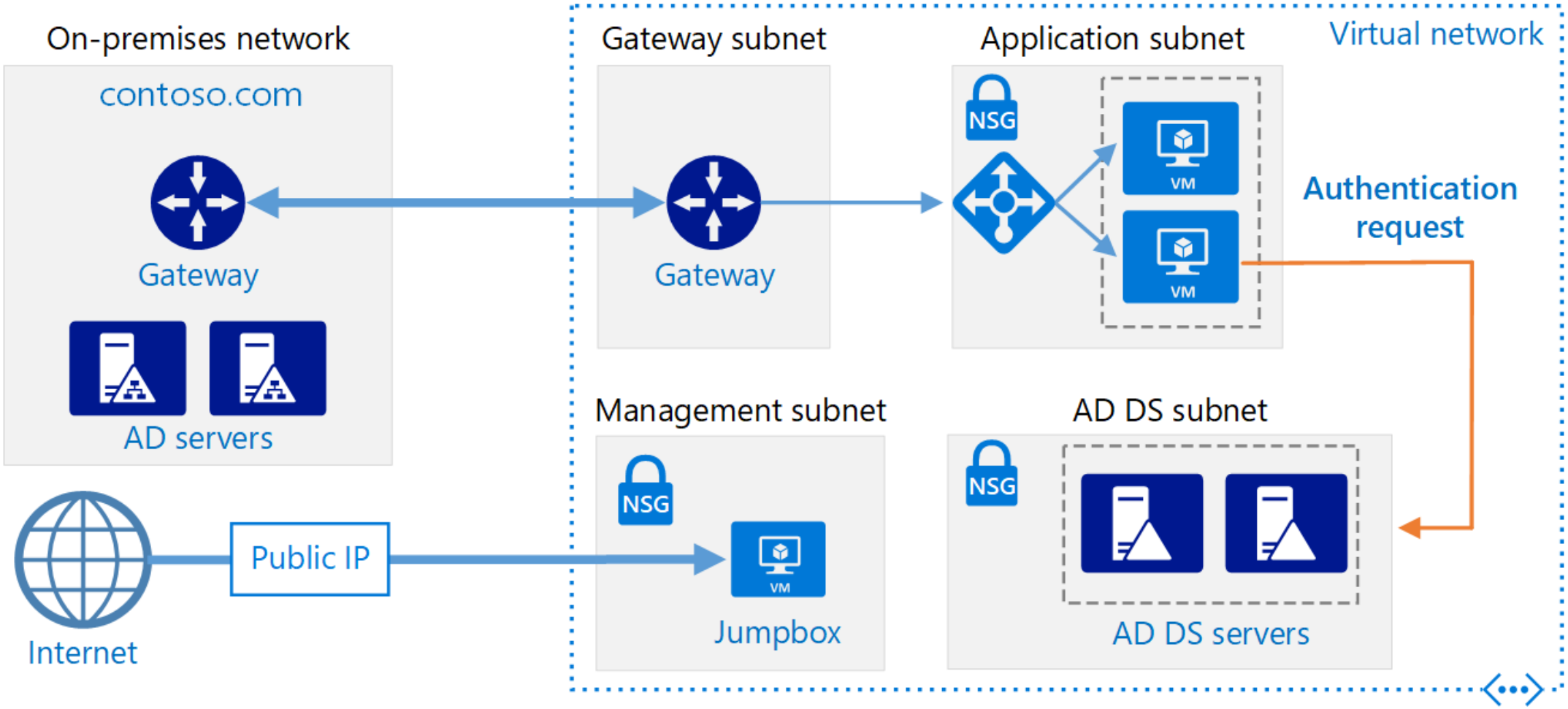
Hands-On Labs

- Use your Azure Pass subscription for labs
- Make sure your subscription is not connected to business related resources
- Lab instructions: https://github.com/mikepfeiffer/azure-arch-masterclass

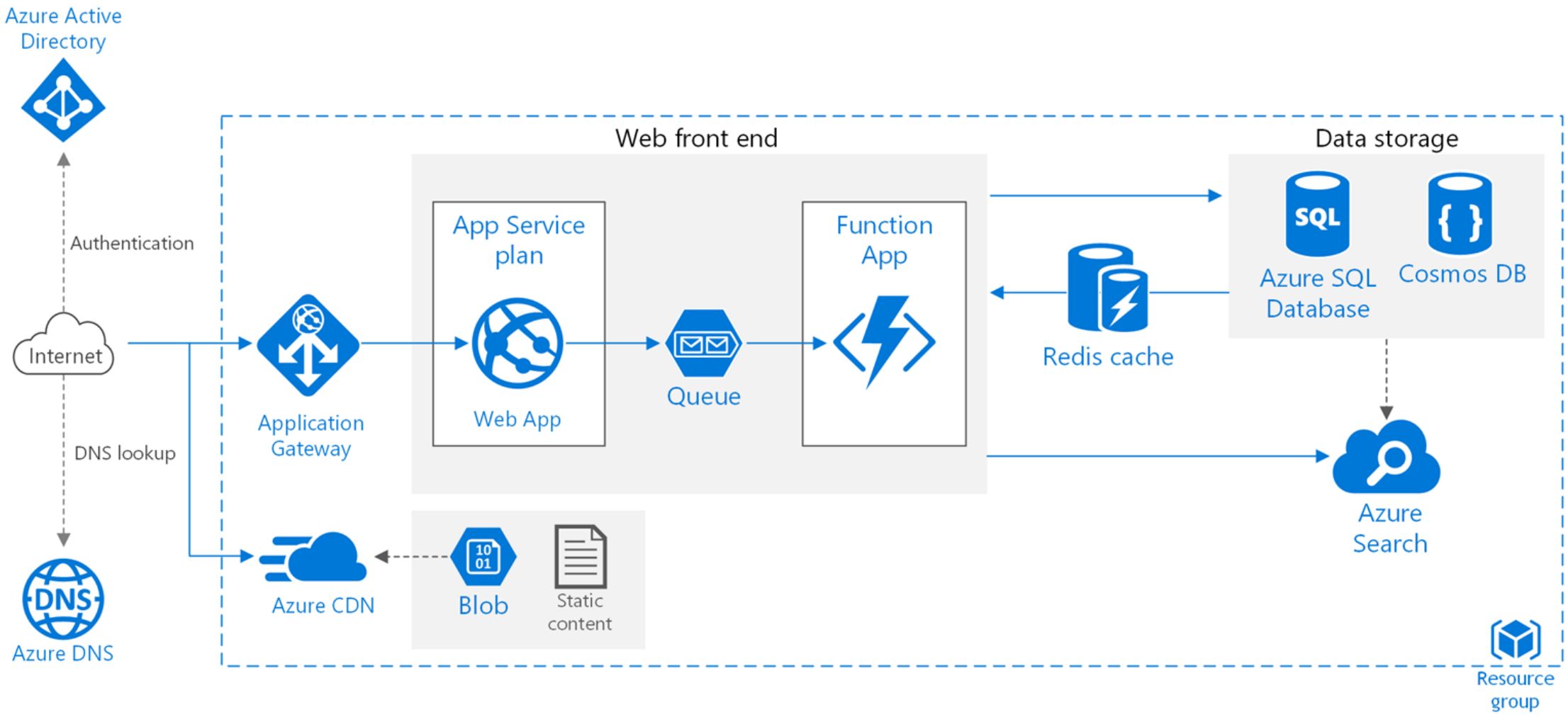
N-tier Application Architecture



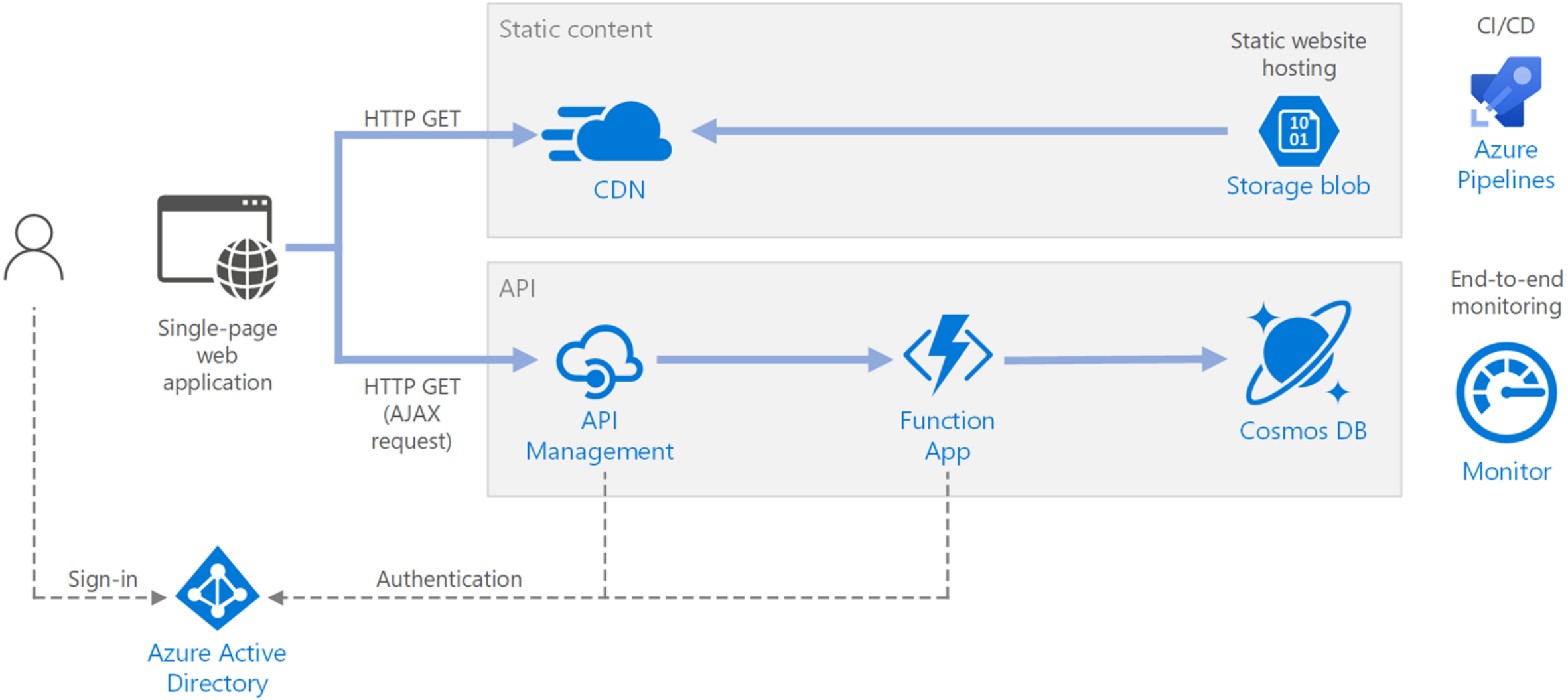
Hybrid Cloud Architecture



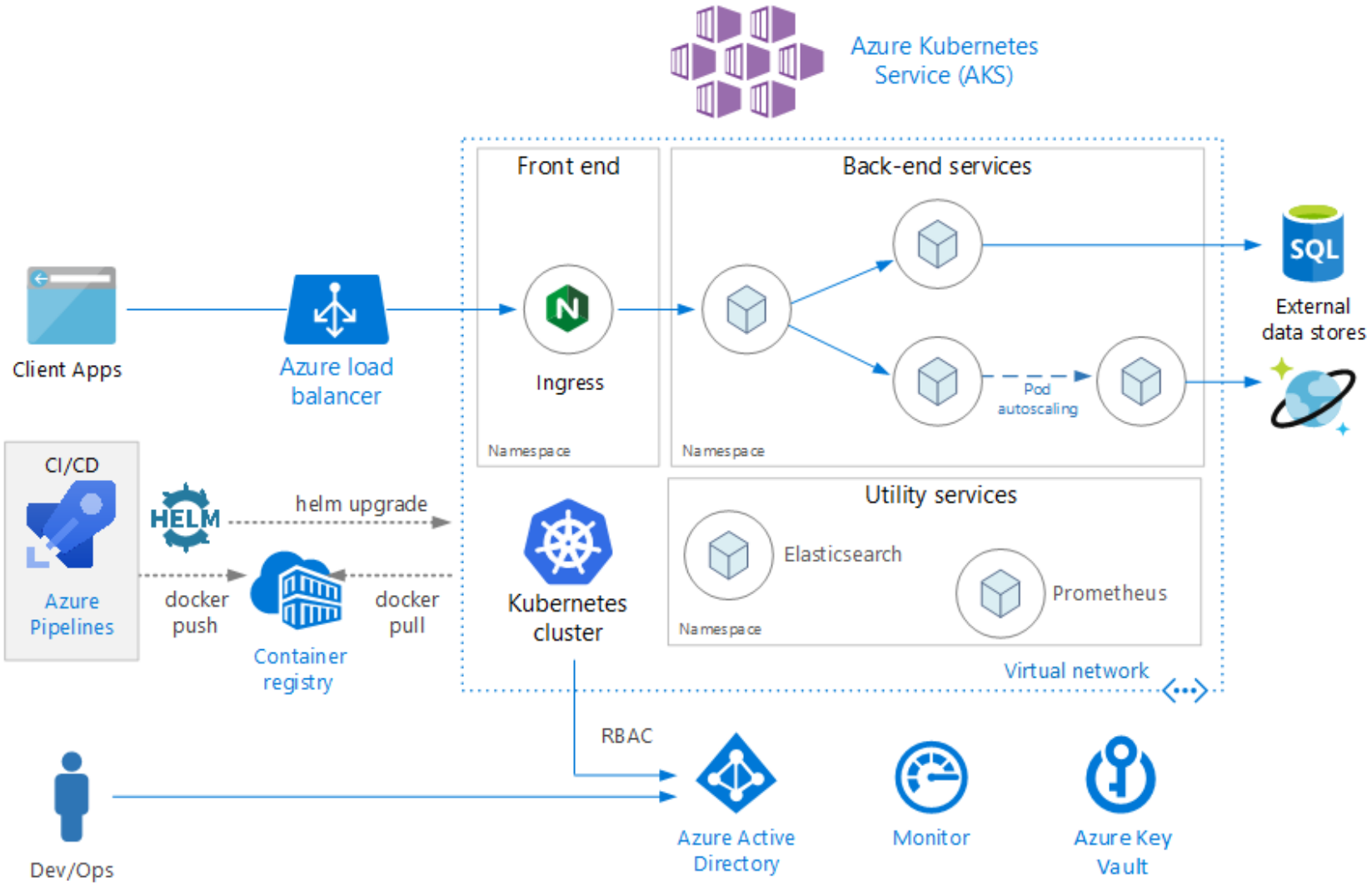
Managed Web App Architecture



Serverless Web App Architecture



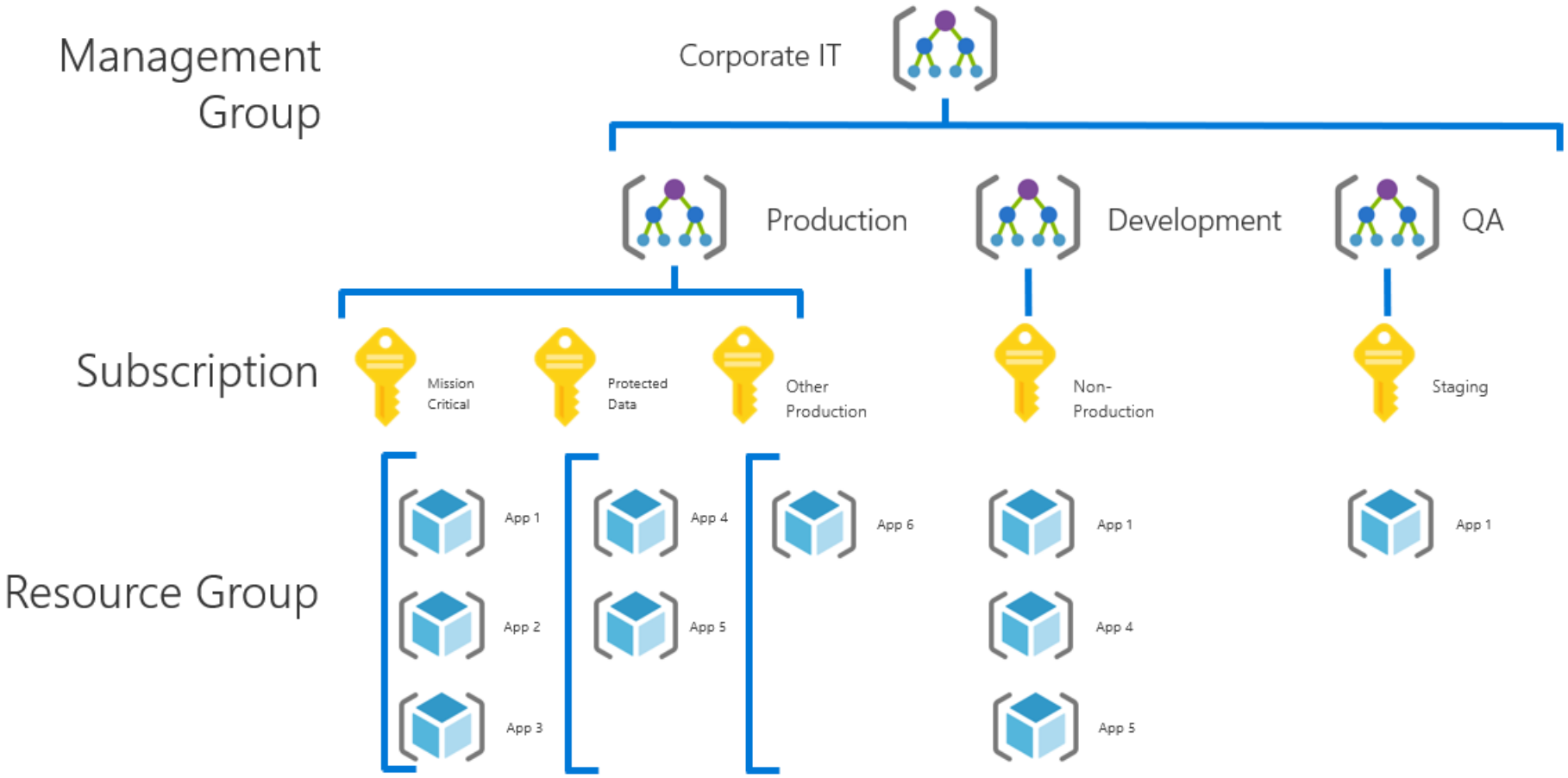
Microservices Architecture



Architecture Best Practices

- Implement the principle of least privilege
- Speed matters; design for automation
- Deploy immutable infrastructure
- Build stateless systems
- Implement loose coupling
- Lead with managed services

Resource Organization



Resource Organization

- Management group for each environment (Prod, Dev, and Test)
- Subscription for each "application categorization"
- Separate resource group for each application
- Consistent nomenclature at each level of this hierarchy

Naming Standards

- Azure Resource Names are Important
- It’s difficult to change names later
- Names must meet requirements specific to their resource type

Organize Resources by Tag

- Allows you to add metadata to Azure resources
- Allows you to query resources using the same tags
- Not all resource types support tags

Azure Policy and RBAC

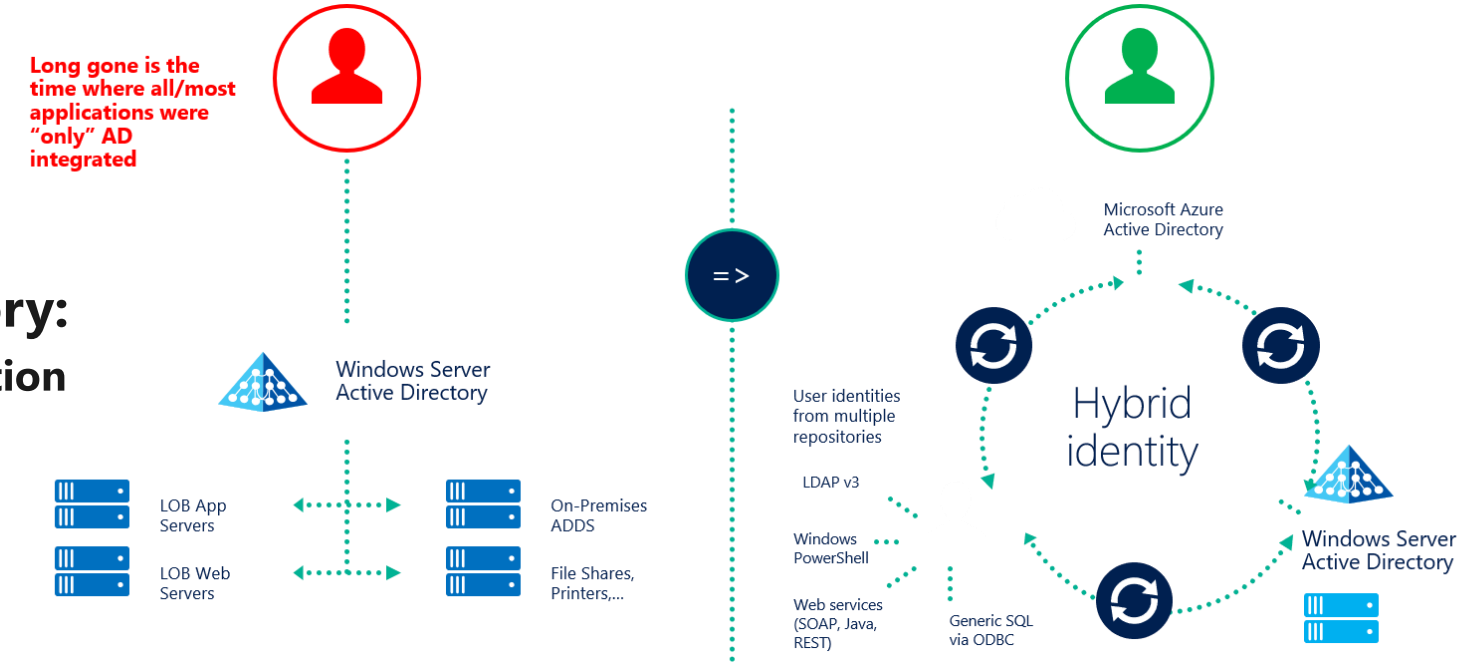
- Policies enforce rules against your resources
- Role-based access control (RBAC) controls user actions
- Can be deployed together via Azure Blueprints

Resource Locks

- Prevent accidental resource changes

Azure Active Directory

- Microsoft's cloud-based directory and identity management service:
  - **Combines directory services, identity governance, and application access management**
  - **Facilitates a wide range of identity and access management features, including:**
    - **Single sign-on for access to SaaS applications**
    - **Multi-factor authentication**
    - **Security monitoring**
    - **Device registration**
    - **Reach auditing**
    - **Alerting**
  - **Integrates with Active Directory:**
    - **SSO with Pass-through Authentication**
    - **SSO with federation (AD FS)**
    - **Password Hash Sync**



Azure Active Directory Benefits

- Reducing costs and enhancing security with self-service capabilities
- Support for iOS, Mac OS X, Android, and Windows devices
- Single sign-on to cloud and on-premises web app
- Extending Active Directory to the cloud
- Protecting cloud and on-premises web applications:
  - **multi-factor authentication**
  - **conditional access policies**
  - **group-based access management**

Active Directory Domain Services (AD DS)

- Windows Server-based identity management and directory service:
  - **Traditional on-premises deployments**
  - **Cloud deployments of IaaS workloads that do not directly support Azure AD**
- Azure AD differs from AD DS in a number of aspects:
  - **Management: Azure AD is a managed service with no infrastructure requirements.**
  - **Communication Protocols: Azure AD uses protocols such as SAML, WS-Federation, and OpenID Connect for authentication and OAuth for authorization.**
  - **Directory lookups: Azure AD Azure AD relies on REST API over HTTP and HTTPS.**
  - **Federation: Azure AD includes built-in federation support.**
  - **Object hierarchy: Azure AD objects exists within a flat namespace. There is no support for custom container-based hierarchy.**

Azure Active Directory Editions

- **Free offers:**
  - **user/group management, single sign-on, self-service password change, basic reporting**
- **Basic offers (in addition to the Free edition features):**
  - **group-based access management, self-service password reset, Azure Active Directory Application Proxy, 99.9% availability SLA**
- **Premium P1 offers (in addition to the Basic edition features):**
  - **access to hybrid application access, self-service identity and access management, security in the cloud**
- **Premium P2 offers (in addition to the Premium P1 features):**
  - **advanced identity protection and privileged identity management**

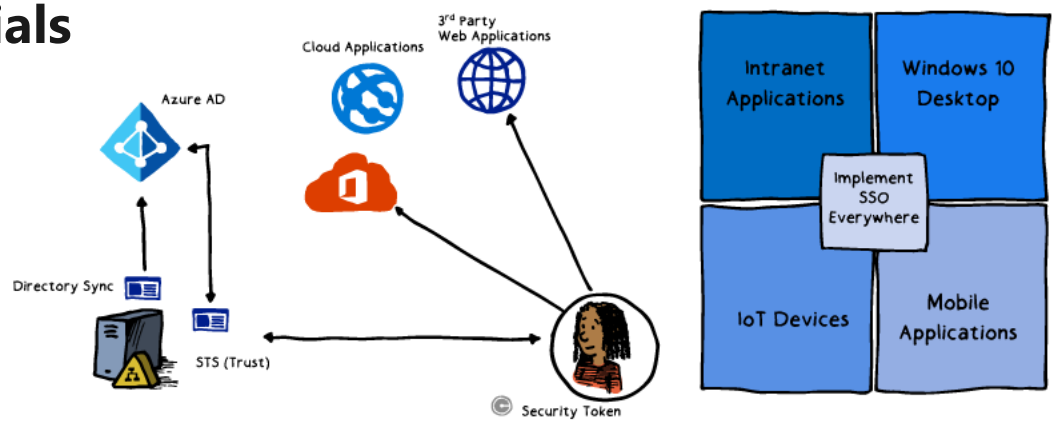
Choosing Between Azure AD and Azure AD DS

- Azure AD DS is a managed AD DS deployment in an Azure VNet:
  - **Integrates with Azure AD**
  - **Delivers core AD DS features:**
    - **Domain join**
    - **Kerberos and NTLM**
    - **LDAP**
    - **Group Policy**
  - **Intended for:**
    - **Azure VMs that rely on AD DS**
    - **Cloud-based environments**

Aspect	Course Content	Azure AD Domain Services
Device controlled by	Azure AD	Azure AD Domain Services managed domain
Representation in the directory	Device objects in the Azure AD directory.	Computer objects in the AAD-DS managed domain.
Authentication	OAuth/OpenID Connect based protocols	Kerberos, NTLM protocols
Management	Mobile Device Management (MDM) software like Intune	Group Policy
Networking	Works over the internet	Requires machines to be on the same virtual network as the managed domain.
Great for ...	End-user mobile or desktop devices	Server virtual machines deployed in Azure

Azure AD B2B & B2C

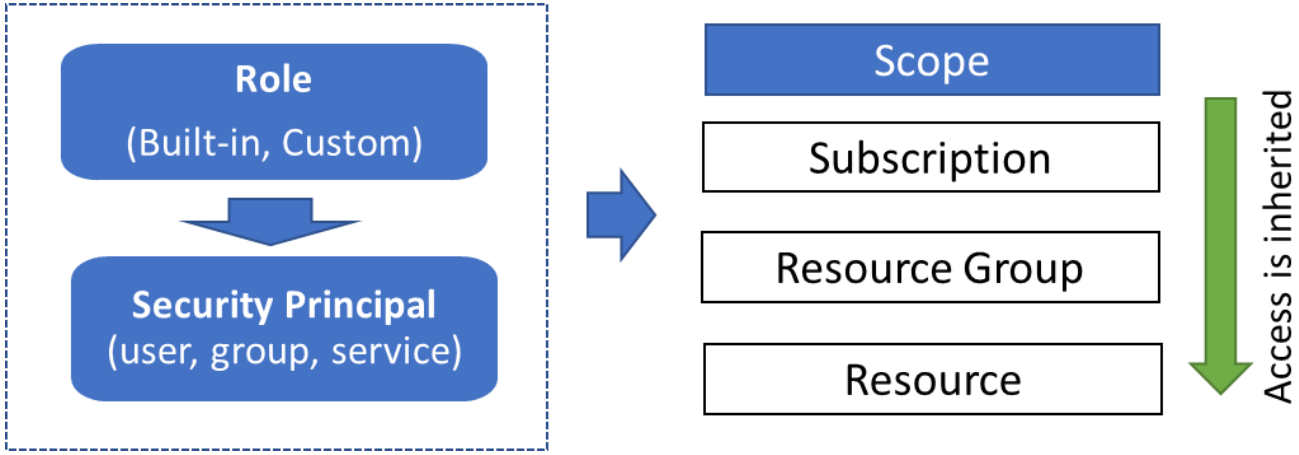
- Azure AD B2B:
  - **Allows granting partner organizations access to local resources and applications**
  - **Allows adding partner organization users to local Azure AD groups**
  - **Supports partner organization user credentials**
  - **Is managed by the host organization:**
    - **Partner lifecycle**
    - **Security policy and compliance**
    - **Branding**

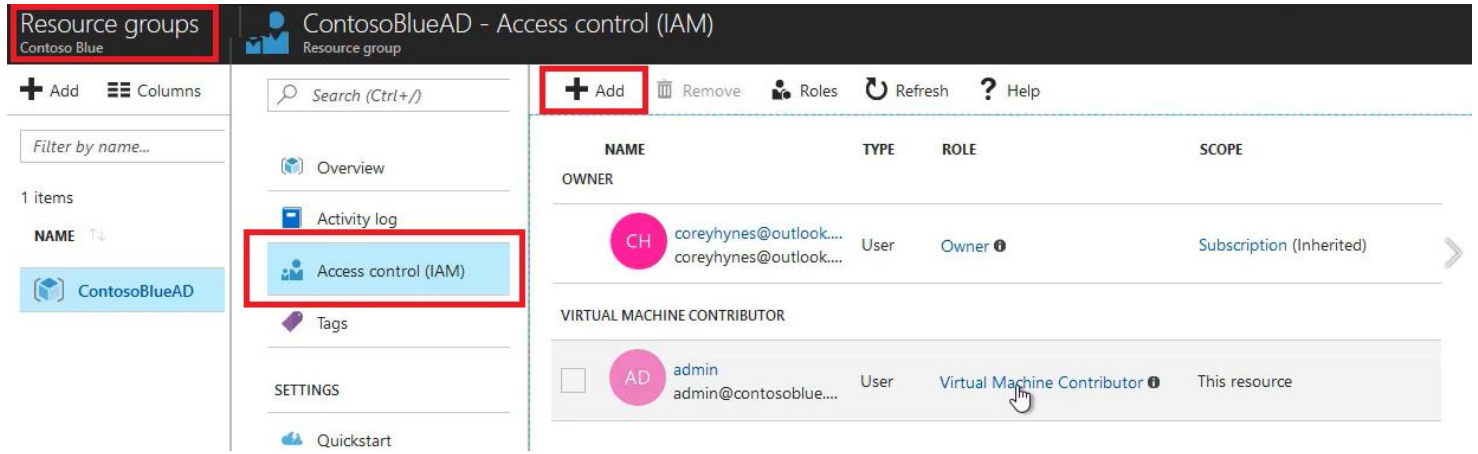


- Azure AD B2C:
  - **Allows granting individual customers access to local resources and applications**
  - **Supports social identities via federation but allows creating new Azure AD accounts**
  - **Requires a separate Azure AD tenant (separately from the host organization's tenant)**
  - **Provides SSO to customer's applications**
  - **Is managed by the host organization with app specific customization**

Role-Based Access Control

- Facilitates granular access to Azure resources:
- Consists of:
  - **Role definition**
  - **Role assignment**
  - **Scope of assignment**
- Can be managed by using:
  - **The Azure portal**
  - **Azure PowerShell**
  - **Azure CLI**
  - **ARM templates**





Built-in Roles

- A role represents a set of permissions to carry out specific actions
- Azure AD provides many built-in roles:
  - **Built-in, not resource specific, including:**
    - **Owner: full access to all resources including the right to delegate access to others.**
    - **Contributor: creating and managing all resources but without the ability to delegate access to others.**
    - **Reader: viewing all resources (except for secrets).**
  - **Built-in, resource specific (e.g. Virtual Machine Contributor)**
- A role is described by a JSON-formatted role definition:
  - **Name, Id, Actions, Not Actions, AssignableScopes**
  - **Can be retrieved by Get-AzureRmRoleDefinition**

```
Get-AzureRmRoleDefinition -Name Owner
Name             : Owner
Id               : 8e3af657-a8ff-443c-a75c-2fe8c4bcb635
IsCustom         : False
Description      : Lets you manage everything, including
                   access to resources.
Actions          : {*}
NotActions       : {}
AssignableScopes : {/}
```

Role Definitions

- Actions and NotActions:
  - **Include or exclude actions associated with the role**
- AssignableScopes:
  - **/subscriptions/[subscription id]**
  - **/subscriptions/[subscription id]/resourceGroups/[resource group name]**
  - **/subscriptions/[subscription id]/resourceGroups/[resource group name]/[resource]**

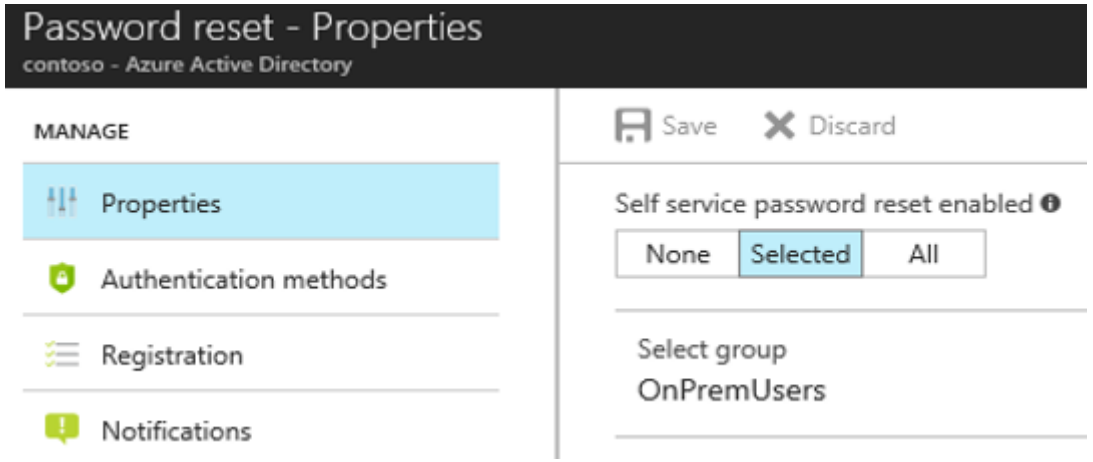
Built-in Role	Action	NotActions
Owner (allow all actions)	*	
Contributor (allow all actions except writing or deleting role assignment)	*	Microsoft.Authorization/*/Delete, Microsoft.Authorization/*/Write, Microsoft.Authorization/elevateAccess/Action
Reader (allow all read actions)	*/read	

Azure PowerShell and CLI

- To automate role management, you can use:
  - **Azure PowerShell:**
    - **New-AzureRmRoleDefinition -InputFile .\sysops.json**
    - **New-AzureRmRoleAssignment -RoleDefinitionName $roleName `**
    - **-SignInName $assigneeName `**
    - **-ResourceGroupName $resourceGroupName**
  - **Azure CLI:**
    - **az role definition create --role-definition "./sysops.json"**
    - **az role assignment create --role $roleName **
      **--assignee $assigneeName **
      --resource-group $resourceGroupName

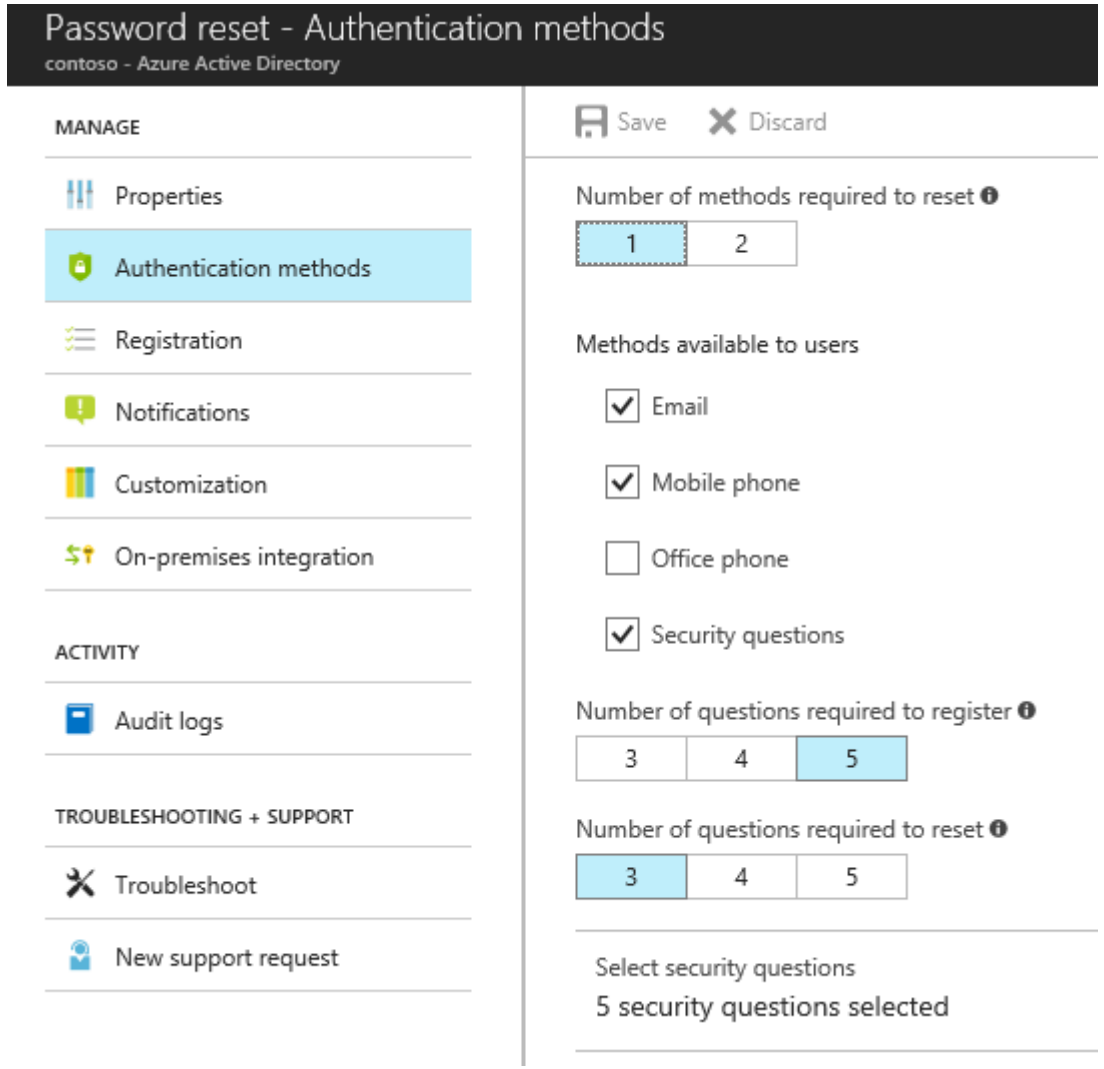
Configuring Self-Service Password Reset

- Available directly from the Azure portal
- Requires designating users in scope:
  - **None**
  - **Selected (limited to specific Azure AD users or groups)**
  - **All**



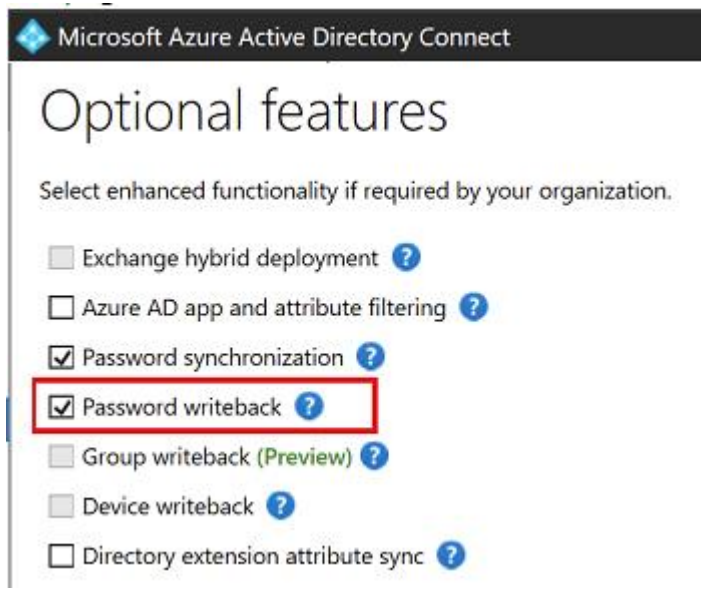
Authentication Methods for Password Reset

- Include (at least one is required):
  - **Email notification**
  - **Text or code sent to the user's mobile device**
  - **Office phone**
  - **A set of security questions**



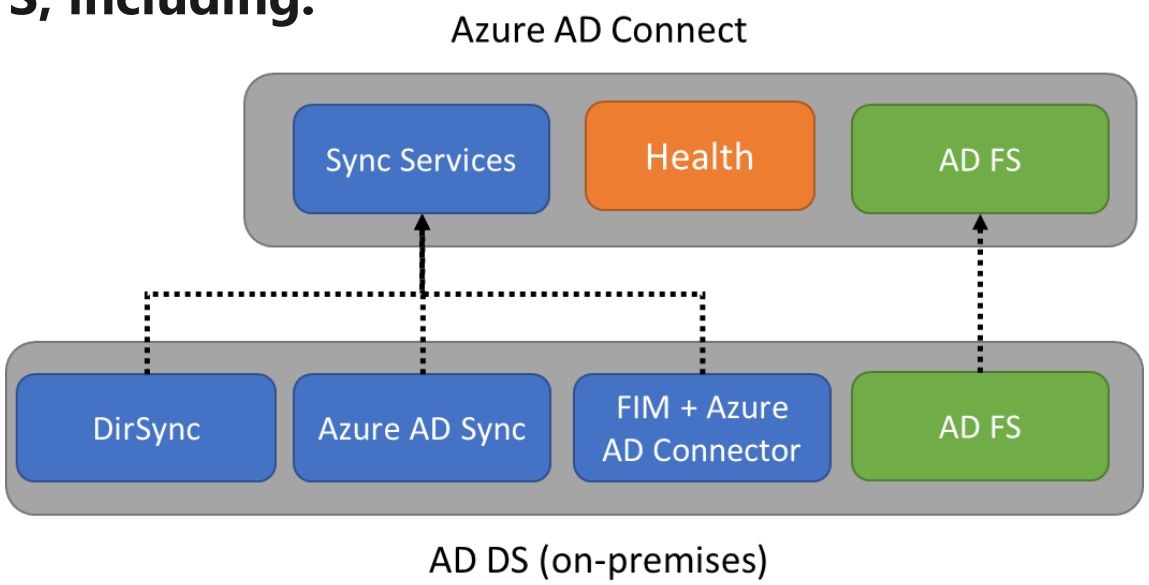
Password Writeback

- Writes Azure AD users' password changes/resets to AD DS
- Eliminates the need for on-premises SSPR solutions
- Requires Azure AD Premium P1 or P2 edition
- Is enabled by using Azure AD Connect:
  - **Start Azure AD Connect Configuration Wizard**
  - **Select Customize synchronization options**
  - **Enable the checkbox on the Optional features page**



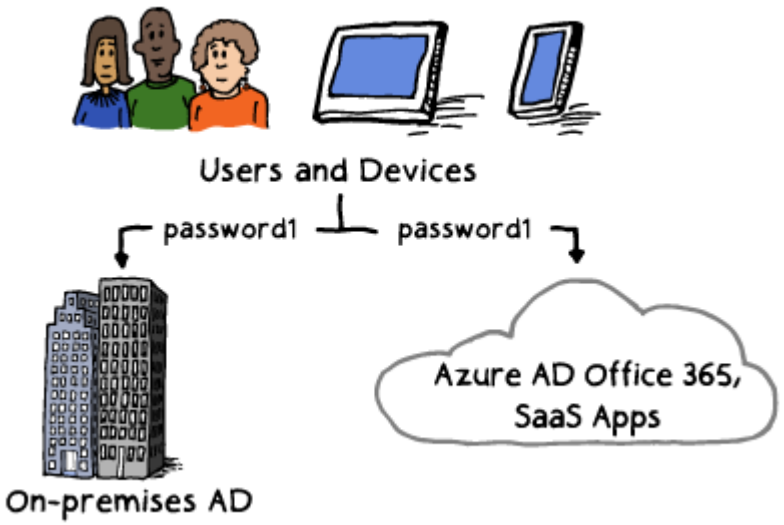
Azure AD Connect

- Integrates AD DS with Azure AD:
  - **Implements a common identity for your users across Azure, Office 365, and SaaS apps**
- Provides support for:
  - **Sync Services: synchronize AD DS objects, such as users and groups.**
  - **Health Monitoring: offers centralized monitoring available from the Azure portal.**
  - **Federation: simplifies configuration of AD FS, including:**
    - **enforcement of AD sign-in policies**
    - **smart card or 3rd party MFA**



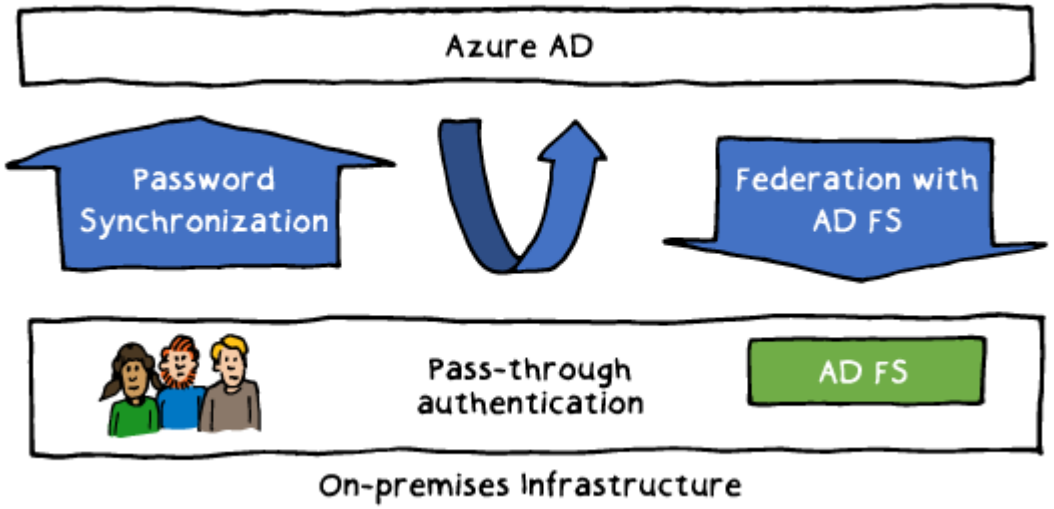
Password Synchronization

- Supported by Azure AD Connect:
  - **Synchronizes user passwords from AD DS to Azure AD**
  - **Allows users to use their AD credentials in order to access:**
    - **Azure resources**
    - **Office 365**
    - **Microsoft Intune**
    - **Dynamics 365**
    - **Azure AD DS**
  - **Improves user productivity**
  - **Reduces helpdesk costs**



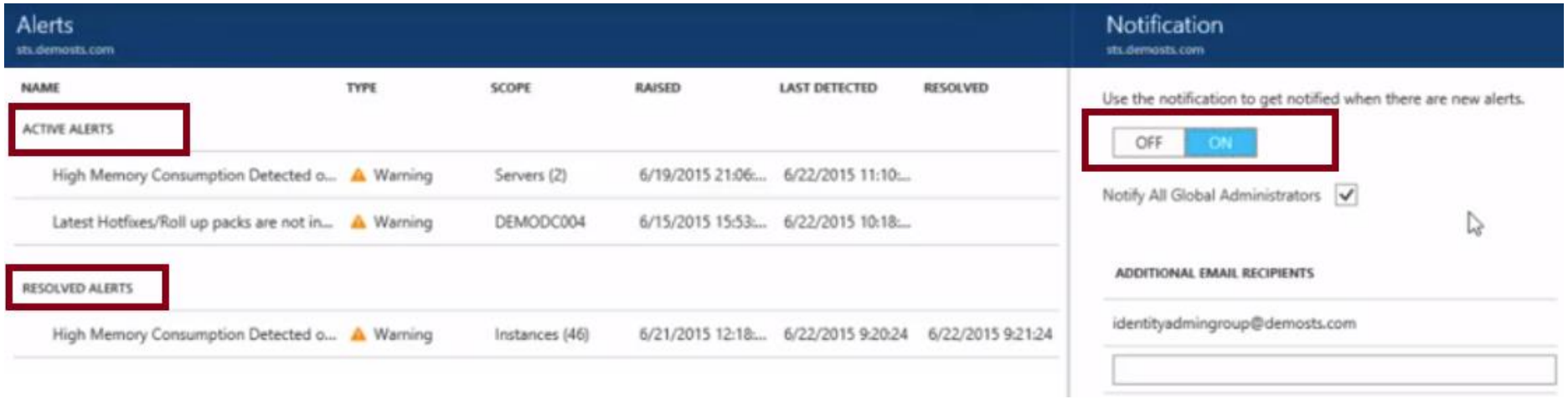
Sign-On Methods

- Available when using Azure AD Connect:
  - **Password Synchronization: synchronizes AD DS user password hashes**
  - **Federation with AD FS: uses federation to redirect authentication requests to AD DS**
  - **Pass-through authentication: redirects authentication requests to AD DS without dependency on a federation service**



Azure AD Connect Health

- Provides monitoring of AD DS and Azure AD integration:
  - **Azure AD Connect and its synchronization engine**
  - **AD DS domain controllers**
  - **AD FS servers**



Device Management

- Azure AD supports device integration by using:
  - **Registering a device in Azure AD:**
    - **Provides the device with an Azure AD identity (a device object) that facilitates device authentication.**
    - **Introduces device object-specific attributes in Azure AD. This facilitates using Azure AD and mobile device management (MDM) solution such as Microsoft Intune to create conditional access rules that allow or block access from devices based on their security and compliance status.**
    - **It supports disabling or enabling the Azure AD identity representing the device.**
  - **Joining a device to Azure AD:**
    - **Constitutes an extension to registering a device**
    - **Provides the benefits of registering a device and, in addition, changes the local state of a device.**
    - **Allows Azure AD users to sign-in to devices using work or school accounts instead of personal accounts.**

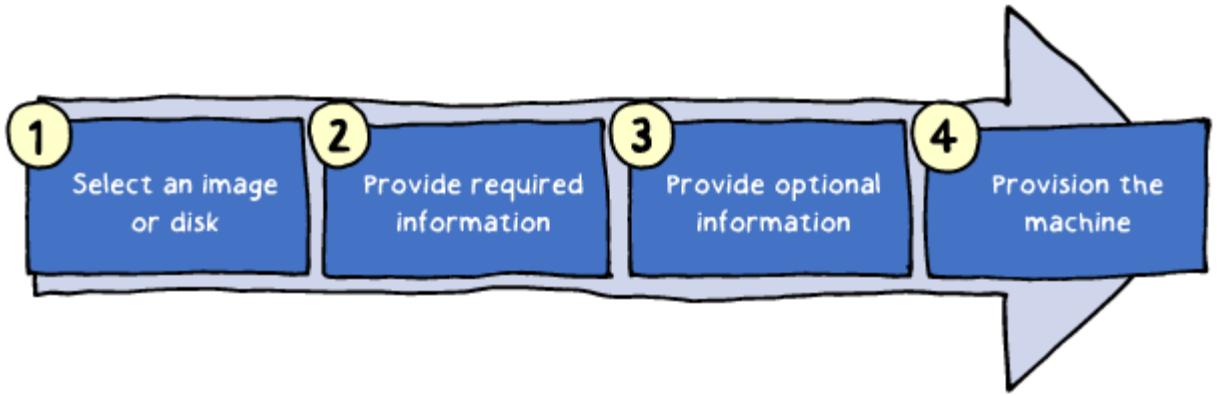
Azure Joined Devices

- Hybrid Azure AD joined devices are joined to:
  - **AD**
  - **Azure AD**
- Hybrid Azure AD join facilitates:
  - **Centralized management of all work-owned devices**
  - **Support for user sign-ins with Active Directory accounts**

	Registered Devices	Azure AD Joined Devices	Hybrid AD Joined Devices
Device Type	Personal	Organization owned	Organization owned
Registration	Manual	Manual	Automatic
Operating System	Windows 10	Windows 10	Windows 7, 8, and 10

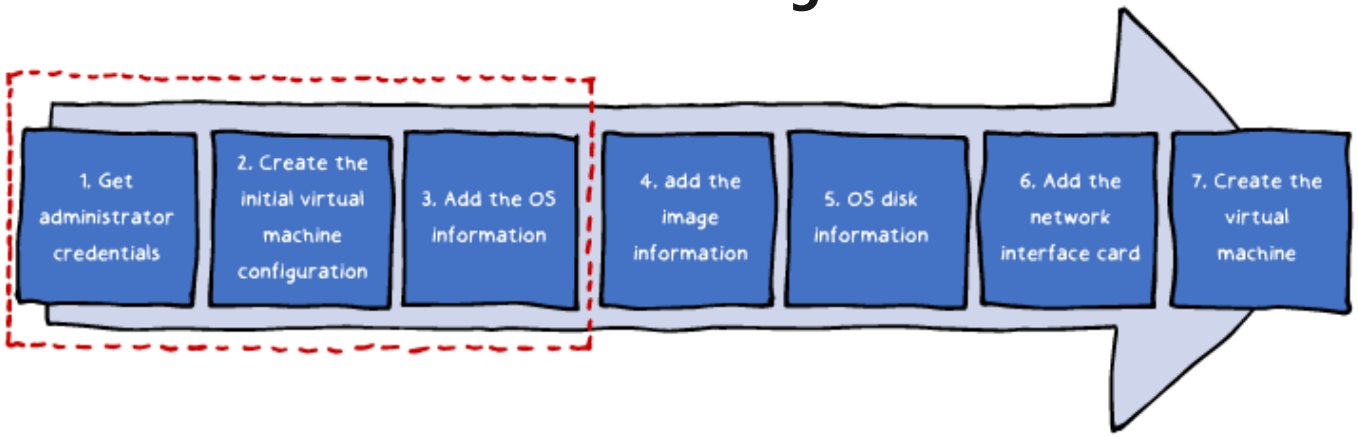
Creating Virtual Machines (Portal)

- High level steps:
  - **1. Select a Marketplace image.**
  - **2. Provide required information, including:**
    - **VM name**
    - **Administrative credentials**
    - **Virtual network and subnet**
    - **Storage type**
  - **3. Provide optional information, including:**
    - **Availability settings**
    - **VM extensions**
  - **4. Provision the machine.**



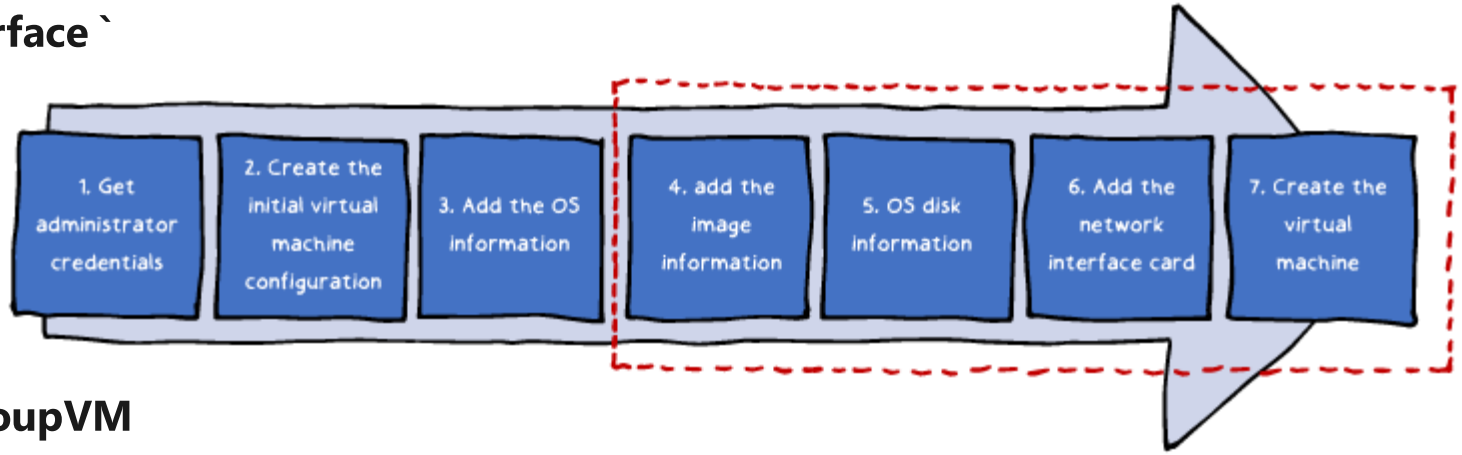
Virtual Machine Example (Part 1)

- A sample provisioning (steps 1-3):
  - **1. Set the credentials of the administrator account:**
    - **$cred = Get-Credential**
  - **2. Create the initial configuration for the virtual machine:**
    - **$vm = New-AzureRmVMConfig -VMName myVM -VMSize Standard_D1**
  - **3. Add the operating system information to the virtual machine configuration:**
    - **$vm = Set-AzureRmVMOperatingSystem `**
    - **-VM $vm `**
    - **-Windows `**
    - **-ComputerName myVM `**
    - **-Credential $cred `**
    - **-ProvisionVMAgent -EnableAutoUpdate**



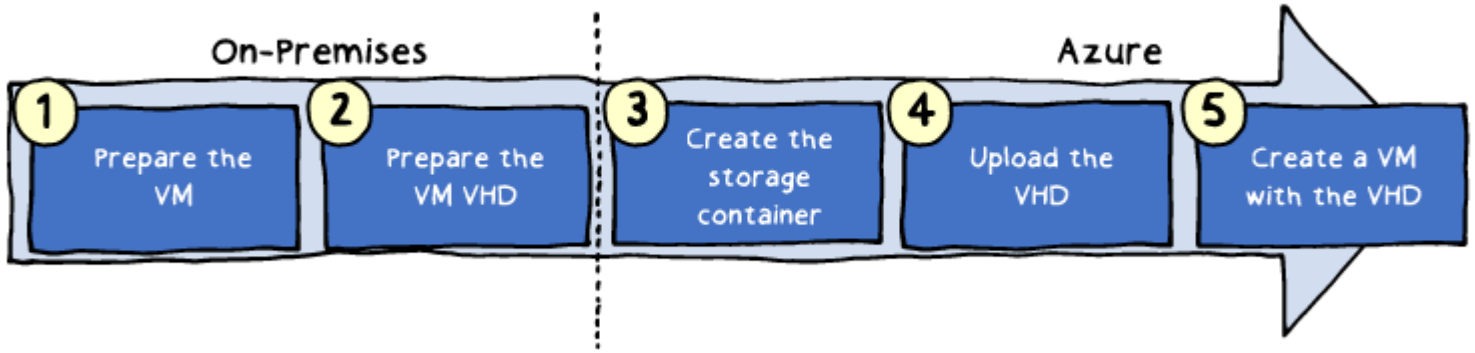
Virtual Machine Example (Part 2)

- A sample provisioning (steps 4-7):
  - **4. Add the image information to the virtual machine configuration:**
    - **$vm = Set-AzureRmVMSourceImage -VM $vm -PublisherName MicrosoftWindowsServer `**
    - **-Offer WindowsServer -Skus 2016-Datacenter -Version latest**
  - **5. Add the operating system disk settings to the virtual machine configuration:**
    - **$vm = Set-AzureRmVMOSDisk -VM $vm -Name myOsDisk -DiskSizeInGB 128 `**
    - **-CreateOption FromImage -Caching ReadWrite**
  - **6. Add the network interface card to the virtual machine configuration:**
    - **$vm = Add-AzureRmVMNetworkInterface `**
    - **-VM $vm -Id $nic.Id**
  - **7. Create the virtual machine:**
    - **New-AzureRmVM `**
    - **-Location EastUS `**
    - **-VM $vm `**
    - **-ResourceGroupName myResourceGroupVM**



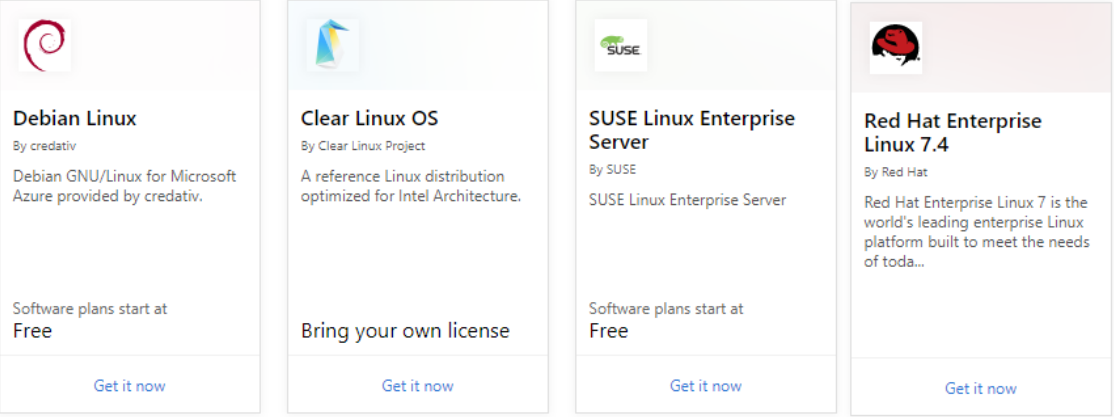
Virtual Machines (Custom Images)

- Images are VHD files containing a generalized operating system:
  - **Can be used to provision Azure VMs**
  - **Can be created by using on-premises VMs:**
    - **On-premises:**
      - **Provision a Hyper-V Generation 1 VM using disks in the VHD format**
      - **Generalize the VM operating system (for Windows OS, run sysprep)**
    - **In Azure:**
      - **Create an Azure Storage account and a container**
      - **Upload the VM VHD files to the container (e.g. by using the Add-AzureRmVhd cmdlet)**



Linux Virtual Machines

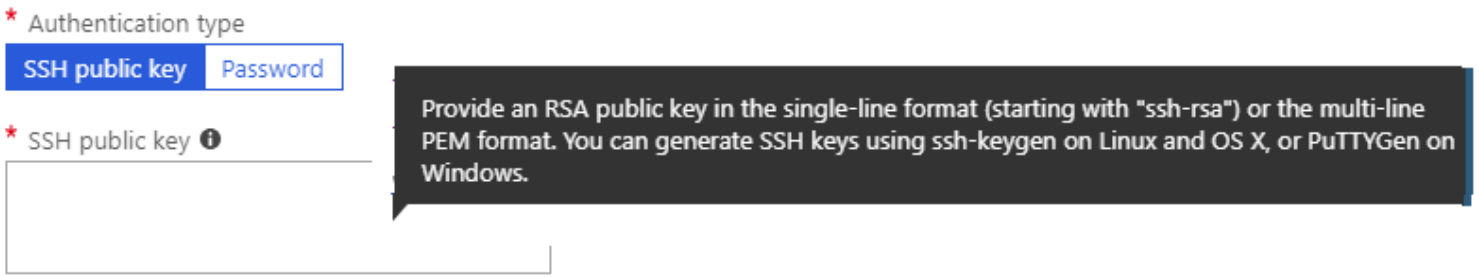
- Azure supports many Linux distributions including:
  - **CentOS by OpenLogic**
  - **Core OS**
  - **Debian**
  - **Oracle Linux**
  - **Red Hat Enterprise Linux**
  - **Ubuntu.**



- Azure Marketplace hosts hundreds of community-built images
  - **Created with Bitnami**
  - **Certified for Azure**
- Linux VMs support the same deployment options as Windows VMs
- Linux VMs support open-source DevOps tools, e.g. Puppet and Chef

Connecting to Linux VMs

- Linux supports two methods of authentication:
  - **Username and SSH-RSA 2048-bit key pair:**
    - **Can be generated by using ssh-keygen (Linux) or PuTTYGen (Windows)**
    - **Private key should be stored securely on the source computer**
    - **Public key resides on the target VM**
  - **Username and password**

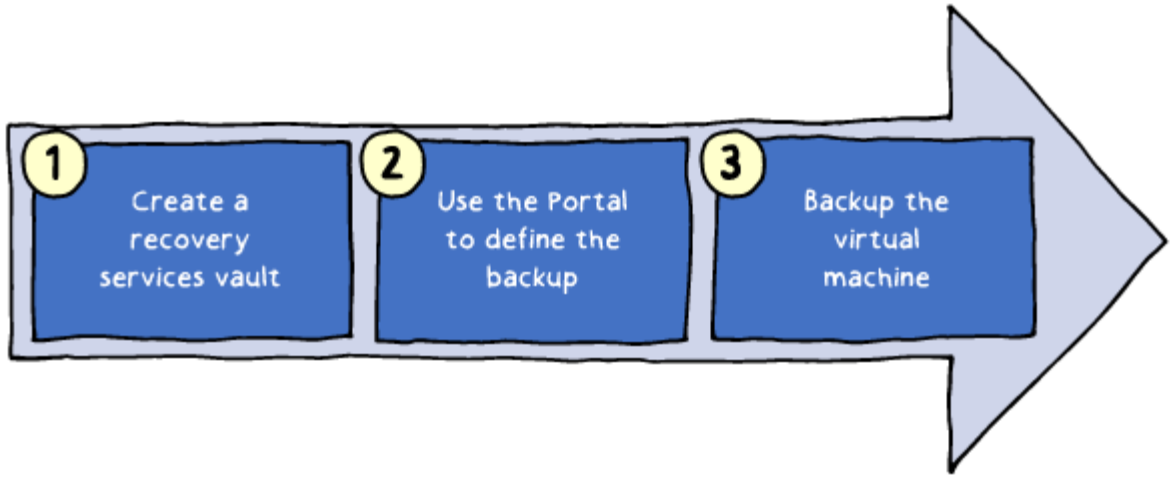


Virtual Machine Backups

- Protection of Azure VMs is available by using:
  - **Azure Backup:**
    - **Supports application-consistent backups of Windows and Linux VMs**
    - **Creates recovery points stored in geo-redundant recovery vaults**
    - **Allows for restoring the whole VM or just specific files**
  - **Azure Site Recovery:**
    - **Protects VMs from outages affecting entire region**
    - **Allows recovery with a single click in matter of minutes**
    - **Replicates to an Azure region of your choice**
  - **Managed Snapshots:**
    - **Read-only full copies of managed disks**
    - **Exist independently of the source disk**
    - **Can be used to create new managed disks for rebuilding a VM**
    - **Accrue charges based on the used portion of the disk**

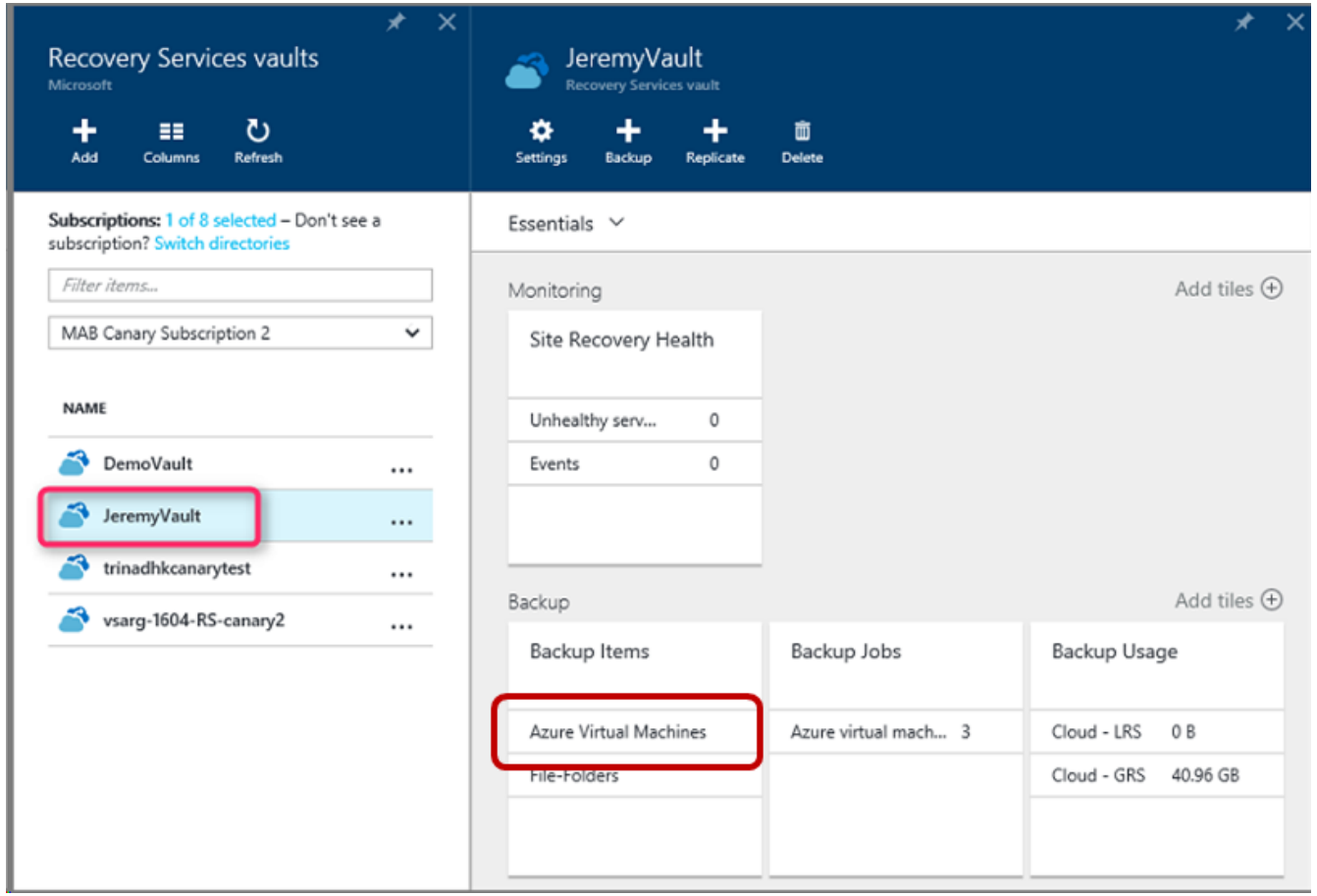
Backup Virtual Machines

- Follows a three-step process:
  - **1. Create a Recovery Services vault with either geo-redundant (default) or locally redundant storage.**
  - **2. Define the backup policy by specifying frequency of recovery points and their retention period.**
  - **3. Initiate backup of Azure VM by installing Backup extension**
    - **Backup VM extension has dependency on the Azure VM Agent:**
      - **Present by default on Marketplace VM images**
      - **Needs to be included in custom VM images**



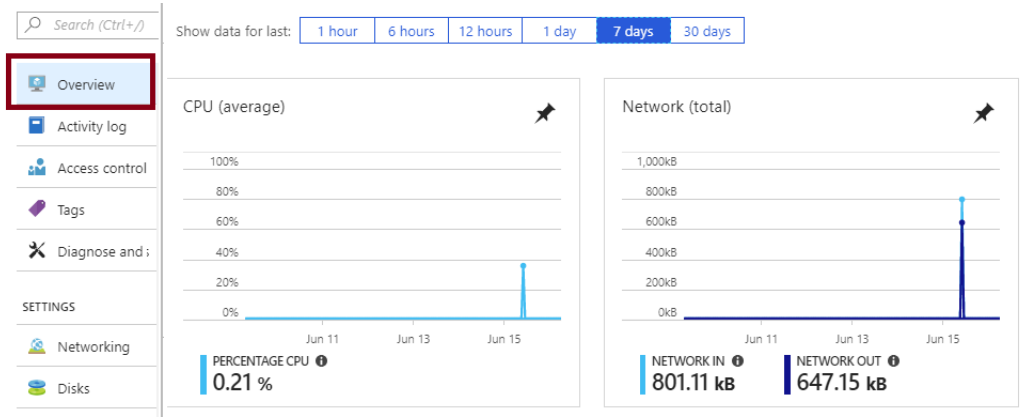
Restore Virtual Machine

- You can trigger the restore operation from the Azure portal:
  - **Azure Backup creates a restore job**
  - **The portal displays restore progress**



Monitoring

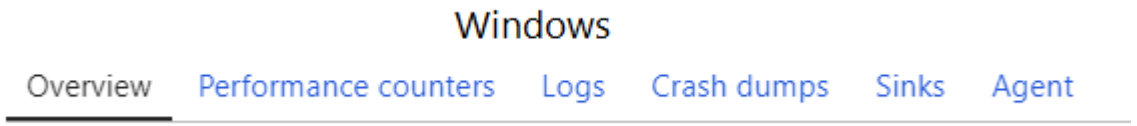
- The Azure VM Overview blade in the Azure portal shows:
  - **CPU**
  - **Network**
  - **Disk bytes**
  - **Disk operations**



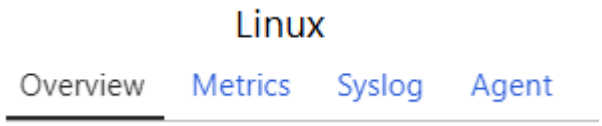
- The Azure VM Monitoring section in the Azure portal shows:
  - **Metrics**
  - **Diagnostic settings**
  - **Advisor recommendations**
  - **Diagram.**

Diagnostic Settings

- On Windows VMs, diagnostics settings include:
  - **Performance counters**
  - **Logs**
  - **Crash dumps**
  - **Agent**
  - **Sinks (deliver diagnostics to other services, such as Application Insights)**

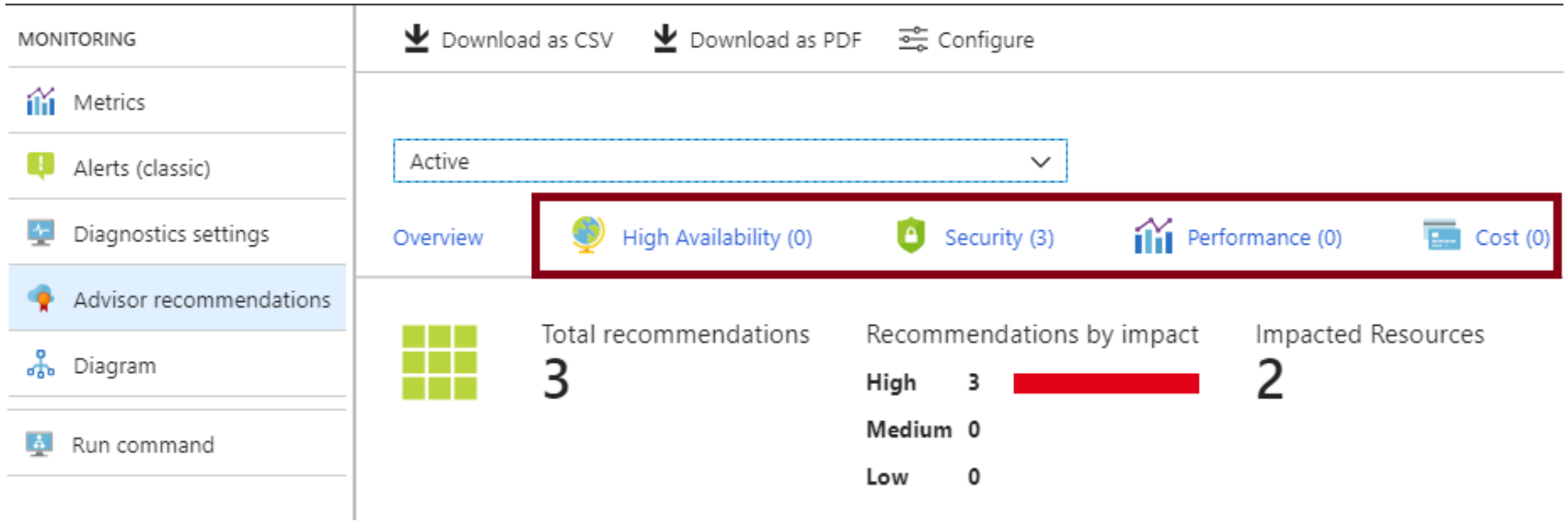


- On Linux VMs, diagnostics settings include:
  - **Metrics**
  - **Syslog**
  - **Agent**



Advisor Recommendations

- Personalized cloud service for optimizing Azure deployments:
  - **Analyzes resource configuration and usage telemetry**
  - **Offers recommendations grouped in four categories:**
    - **High availability: To ensure and improve the continuity of your business-critical applications.**
    - **Security: To detect threats and vulnerabilities that might lead to security breaches.**
    - **Performance: To improve the speed of your applications.**
    - **Cost: To optimize and reduce your overall Azure spending.**



Azure Resource Manager

- Objectives:
  - **Provide consistent deployment and management model in Azure and Azure Stack**
  - **Support grouping of resources**
  - **Facilitate Infrastructure as Code**
  - **Implement resource providers**
  - **Encourage DevOps practices**

Azure Resource Manager Objects

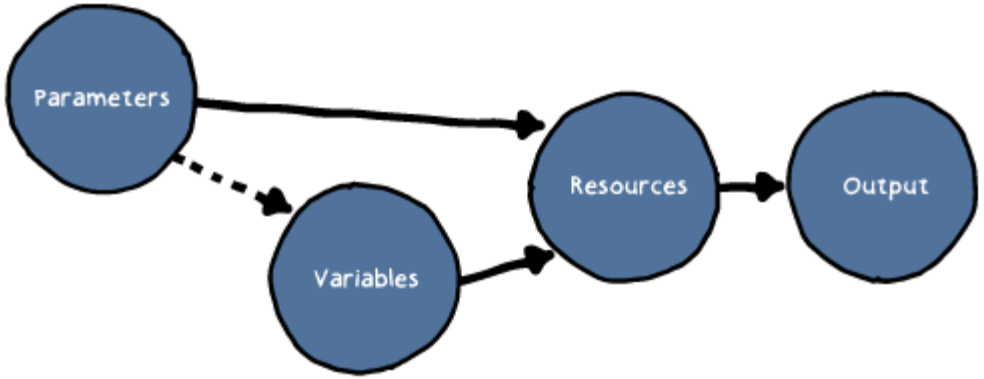
- An Azure Resource Manager resource:
  - **An manageable instance of an Azure service or component of an Azure service (e.g. Azure Web App, an App Service plan, or Azure SQL Database)**
- A resource group:
  - **Logical grouping of Azure Resource Manager resources**
- Resource group templates:
  - **JSON-based representation of Azure Resource Manager resources that can be deployed together into a resource group**

Interacting with Azure Resource Manager

- Azure PowerShell
- Azure CLI
- Client libraries
- Visual Studio and Visual Studio Code
- the Azure portal
- REST API

ARM Templates

- A JSON-based representation of Resource Manager resources:
  - **Allows for deployment via:**
    - **Visual Studio**
    - **Azure PowerShell**
    - **Azure CLI**
    - **the Azure portal**
  - **Facilitates idempotency**
  - **Simplifies orchestration**
  - **Allows to express resource dependencies**
  - **Promotes reuse through parameters and nesting**



JSON

- JavaScript Object Notification (JSON):
  - **A lightweight format for transmitting data objects**
  - **Used in Azure Resource Manager templates**
- Azure Resource Manager template format:
  - **$schema (mandatory)**
  - **contentVersion (mandatory)**
  - **parameters (optional)**
  - **variables (optional)**
  - **functions (optional)**
  - **resources (mandatory)**
  - **outputs (optional)**

```
{
  "$schema": "https://schema.management.azure.com/schemas/2015-01-
01/deploymentTemplate.json#",
  "contentVersion": "1.0.0.0",
  "parameters": {
  },
  "variables": {
  },
  "resources": [
  ],
  "outputs": {
  }
}
```

Role-Based Access Control

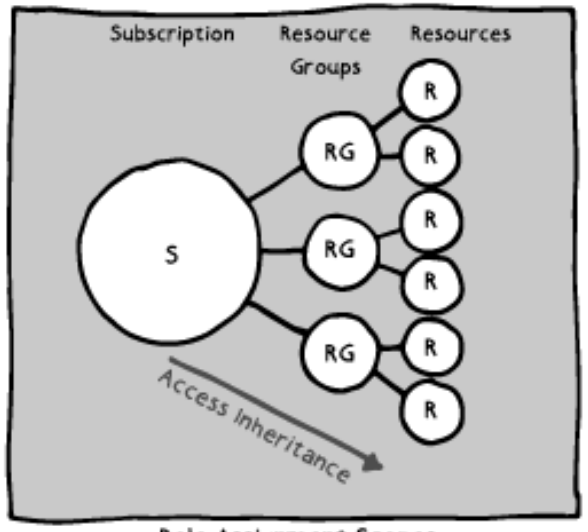
- An administration delegation mechanism that associates:
  - **A role**
  - **An Azure Active Directory identity**
  - **An Azure subscription, resource group, or resource**
- A role represents a set of permissions to carry out specific actions:
  - **Built-in, not resource specific (Owner, Contributor, Reader, User Access Administrator)**
  - **Built-in, resource specific (e.g. Virtual Machine Contributor)**
  - **Custom**
-

Role Assignment

- Role assignment associates a role with identities that exist in the Azure AD tenant associated with the Azure subscription:
  - **Users: include Microsoft and guest accounts that exist in the Azure AD tenant)**
  - **Service principals: represent Azure AD applications and Azure resources)**
  - **Groups: can contain users, service principals, and groups. Group-based assignments are recommended due to lower management overhead (in comparison with direct assignments to users and service principals).**

Resource Scope

- The role assignment can be scoped to:
  - **the subscription**
  - **a resource group**
  - **a resource**
- RBAC-based permissions propagate from a parent to children:
  - **It is possible to grant additional permissions by using child-level assignment**
  - **It is not possible to deny inherited permissions by using child-level assignments**



Role Assignment Scopes

Custom Roles

- Complement built-in roles
- Can be provisioned by using:
  - **Azure PowerShell**
  - **Azure CLI**
  - **REST API**
- To create a custom role (by using Azure PowerShell):
  - **Create a JSON file containing role definition**
  - **Run New-AzureRmRoleDefinition with the InputFile parameter that references the role definition file**

```
{
  "Name": "New Role 1",
  "Id": null,
  "IsCustom": true,
  "Description": "Allows for read access to Azure storage and compute
resources",
  "Actions": [
    "Microsoft.Compute/*/read",
    "Microsoft.Storage/*/read",

  ],
  "NotActions": [
  ],
  "AssignableScopes": [
    "/subscriptions/c489345-9cd4-44c9-99a7-4gh6575315336g"
 ]
}
```

Azure Resource Policies

- Enhances governance and compliance by controlling settings of Azure resources, e.g.:
  - **Resource types**
  - **Location**
  - **Size**
  - **Naming convention**
- To implement policies, you need:
  - **A policy definition**
  - **A policy assignment**

Azure Policies vs RBAC

- RBAC:
  - **Grants Azure AD identities permissions to carry out actions on Azure resources**
  - **Can be scoped to management group, subscription, resource group, or resource**
  - **Default deny, explicit allow**
- Azure policies:
  - **Compares policy conditions to properties of a resource during or after its deployment**
  - **If the comparison is successful, automatically applies the ‘effect’ defined in the policy**
  - **Can be scoped to management group, subscription, resource group, or resource**
  - **Default allow, explicit deny**
- To implement policies, it is required to have two RBAC permissions:
  - **Microsoft.Authorization/policydefinitions/write permission to define a policy**
  - **Microsoft.Authorization/policyassignments/write permission to assign a policy**

Built-in Policies

- Azure offers a library of built-in policy definitions, including:
  - **Allowed locations**
  - **Allowed resource types**
  - **Allowed storage account SKUs**
  - **Allowed virtual machine SKUs**
  - **Apply tag and default value**
  - **Enforce tag and value**
  - **Not allowed resource types**
  - **Require SQL Server version 12.0**
  - **Require storage account encryption**

Policy Definitions

- A policy definition is a JSON-formatted file consisting of:
  - **mode**
  - **parameters**
  - **display name**
  - **description**
  - **policy rule**
  - **logical evaluation**
  - **effect:**
    - **Deny**
    - **Audit**
    - **Append**
    - **AuditIfNotExists**
    - **DeployIfNotExists**

policy not allowed.json

```
1 {
2   "if": {
3     "field": "type",
4     "in": "[parameters('listOfResourceTypesNotAllowed')]"
5   },
6   "then": {
7     "effect": "Deny"
8   }
9 }
```

Policy Assignments

- To take effect, a policy definition must be assigned.
- An assignment consists of:
  - **A policy definition**
  - **Values of parameters of the policy definition**
  - **A scope (a management group, subscription, resource group, or resource)**
  - **Zero or more exclusions (by default, policy assignments at a higher level are inherited by child all resources)**
- You can combine multiple policy definitions into policy initiatives:
  - **Policy initiatives simplify policy management and maintenance**
  - **Assigning policy initiatives follow the same rules as policy assignments**

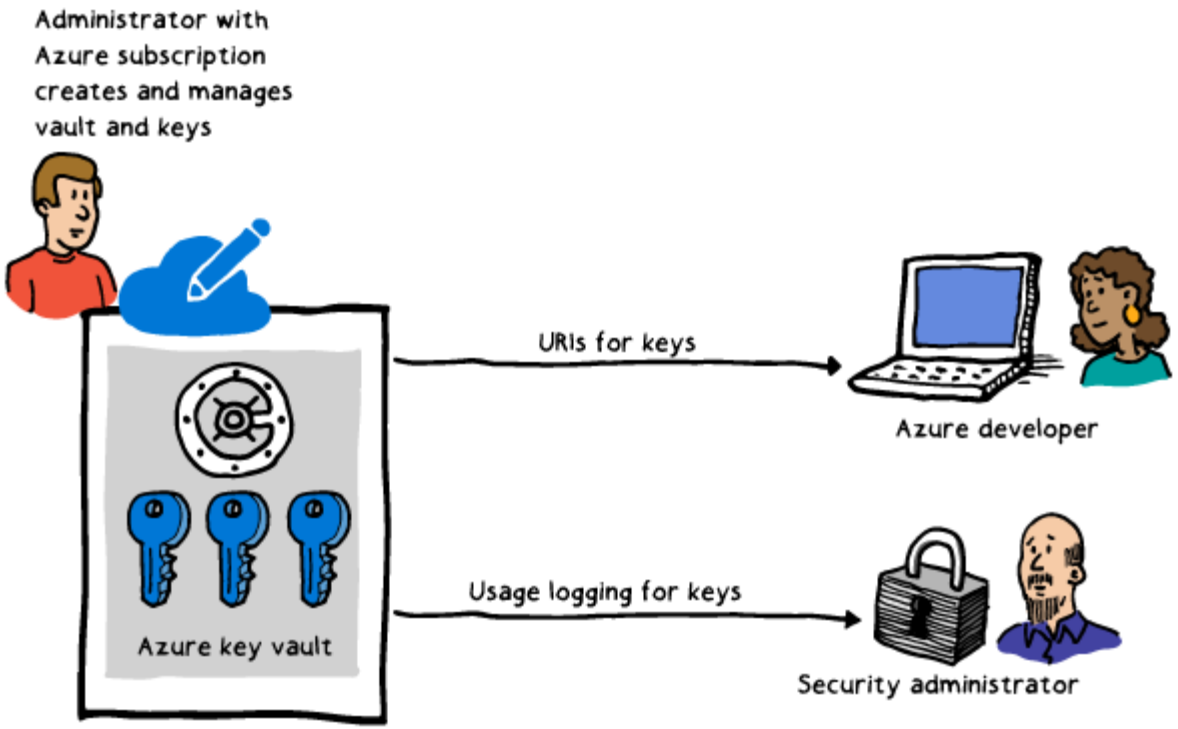
Policies for Naming Conventions

- A policy definition to enforce naming conventions:
  - **Relies on pattern matching and wildcards**
  - **Causes deployment failure of non-compliant resources**
  - **Reports presence of existing non-compliant resources**

```
{
  "if": {
    "not": {
      "field": "name",
      "match": "contoso??????"
    }
  },
  "then": {
    "effect": "deny"
  }
}
```

Azure Key Vault

- An Azure service that uses an HSM to store:
  - **Cryptographic keys**
  - **Certificates**
  - **Secrets**
- Supports role-based separation:
  - **Azure Administrators:**
    - **Manage Key Vault secrets, keys, and certificates**
  - **Developers:**
    - **Reference secrets, keys, and certificates**
  - **Security administrators:**
    - **Monitor usage logs**
    -



Azure Key Vault in ARM Templates

- Protects secrets used during deployment:
  - **Secrets are retrieved automatically during deployment**
  - **An Azure AD identity used to perform deployment does not need permission to access to the secrets directly**
- To implement:
  - **Set the Key Vault's enabledForTemplateDeployment advanced policy to True**
  - **Use RBAC to grant the Azure AD identity used to perform deployment the Microsoft.KeyVault/vaults/deploy/action permission to the Key Vault**
  - **Reference the Key Vault and the secrets in the ARM template**

What is Azure Automation?

- An Automation service integrated with Microsoft Azure for automating the creation, deployment, monitoring and maintenance of Azure resources and resources external to Azure
- Integration with Microsoft Azure remove some complexity of automating in Azure

<u>Azure Automation Capabilities include:</u>

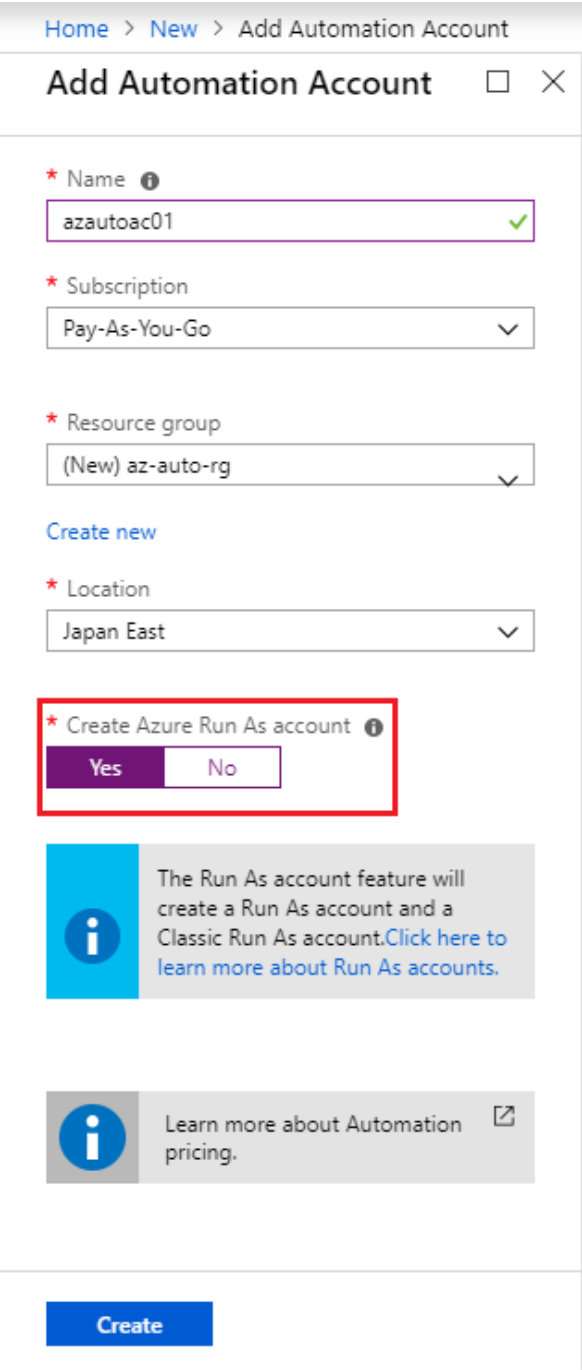
- Manage Shared resources
- State configuration
- Integration with GitHub, Azure DevOps Git/TFVC
- Update management
- Can automate Windows or Linux environments
- Can apply to any system that exposes an API over internet protocols

Automation Accounts

- To use Azure Automation you must create an Azure Automation account
- Automation account acts as a container in which you store, manage and use automation artifacts
- Provides a way to separate your environments or further organize your Automation workflows and resources.
- Requires subscription-owner level access as provides access to all Azure resources via an API
- Need at least 1 but should have multiple for access control

Run As account

- Can create when creating Azure Automation account
- Service principal which allows access to Azure resources when running automation

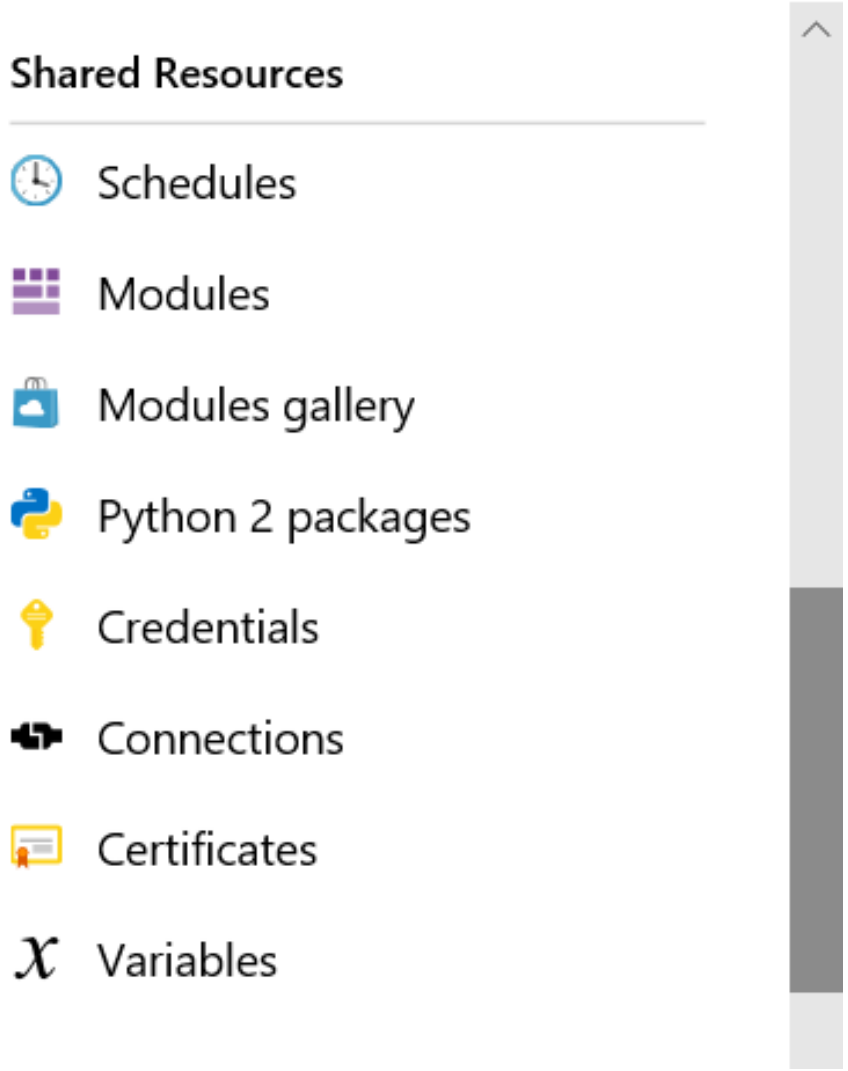


Automation Shared Resources

- Azure Automation contains shared resources that are globally available to be used in, or associated with a runbook

<u>Currently Eight Categories:</u>

- Schedules
- Modules
- Modules gallery
- Python 2 packages.
- Credentials
- Connections
- Certificates
- Variables



What is a Runbook?

- A *runbook* is a set of tasks that perform some automated process in Azure Automation
- Runbooks serve as repositories for your custom scripts and workflows
- Can create your own or import and modify from community via Runbook Gallery

<u>Runbook Types available:</u>

- Graphical runbook
- PowerShell runbooks
- PowerShell Workflow runbooks
- Python runbooks

Automating Processes with Runbooks

- Runbooks in Azure Automation can do anything that PowerShell can do because they are based on Windows PowerShell or Windows PowerShell Workflow and Python2

Example Usage Scenarios:

- Automating resources in your on-premises datacentre
- Start/Stop Azure virtual machines during off-hours solution in Azure Automation
- Auto-stop VMs based on low CPU usage, to optimize VM costs.
- Automate provisioning of a virtual machine in Amazon Web Services (AWS)
- Send your Automation logs to Log Analytics for analysis and have alerts such as email configured to notify you of any failures or other relevant outcomes.

Runbook Gallery

- Azure Automation runbooks are provided to help eliminate the time it takes to build custom solutions
- Already been built by Microsoft and the Microsoft community
- Can be used with or without modification
- Can view or visual runbooks on the gallery as well as see source projects, rating etc

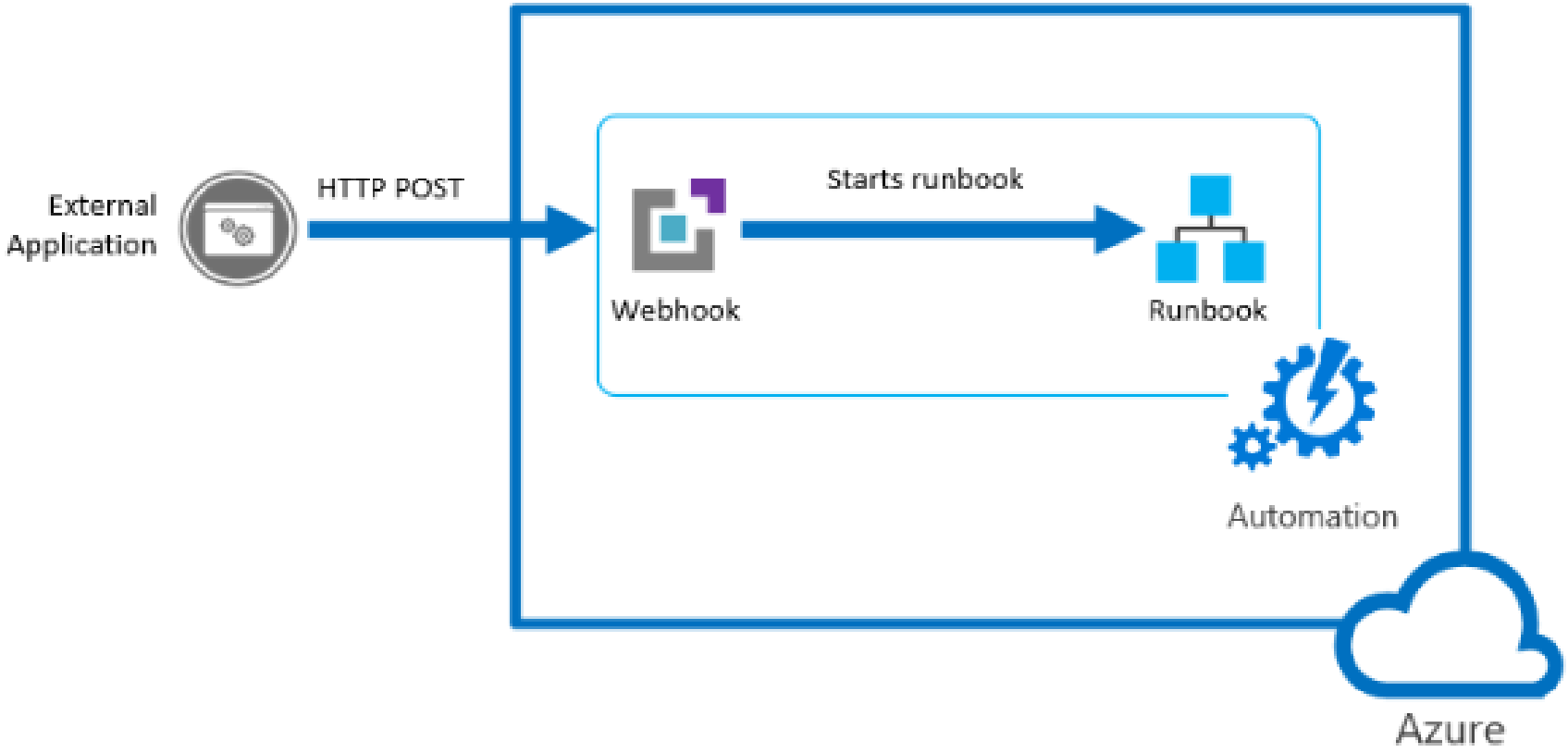
<u>Considerations:</u>

- Python runbooks are also available from the script center gallery. To find them, filter by language and select **Python**
- You cannot use PowerShell to import directly from the Runbook Gallery
- The new Azure PowerShell **Az module** is not supported in Azure Automation. Any scripts downloaded from the PowerShell Gallery with these **Az** cmdlets will not work in Azure Automation

Webhooks

- Automate the process of starting a runbook either by scheduling it, or by using a *webhook*.

- Uses a HTTPS request to start a runbook
- Reduces complexity and allows external services such as Azure DevOps, GitHub, or custom applications to use webhooks



- Webhook Syntax: http://< Webhook Server >/token?=< Token Value >

Source Control Integration

- Azure Automation supports source control integration
- Easier collaboration
- Increased auditing and traceability
- Roll back to earlier versions of your runbooks
- Can push code from Azure Automation to source control or pull your runbooks from source control to Azure Automation
- pull your runbooks from source control to Azure Automation

<u>Azure Automation support the following source Control options:</u>

- GitHub
- Azure DevOps (Git)
- Azure DevOps (TFVC)

Update Management

- A service built-in to Azure Automation
- Can be used to assess and manage operating system updates
- Applicable to Windows or Linux environments in Azure, On-premises, or other clouds providers.
- Onboard machines in multiple Azure subscriptions in the same tenant.

<u>Components used:</u>

- Microsoft Monitoring Agent (MMA) for Windows or Operations Management Suite (OMS) Linux Agent
- PowerShell DSC for Linux
- Automation Hybrid Runbook Worker
- Microsoft Update or Windows Server Update Services (WSUS) for computers running Windows operating systems or yum/apt/zypper package management services available on Linux.

Desired State Configuration (DSC)

- Ensures that an environment is maintained in a state that you specify (*defined state*), to eliminate configuration drift, and no deviation from that *defined state*.

<u>Components:</u>

- **Configurations:** Idempotent declarative PowerShell scripts
- **Resources:** PowerShell scripts compiled into .mof format files to ensure state.
- **Local Configuration Manager (LCM):** Client engine to ensure configuration

<u>Methods of Implementing DSC:</u>

- **Push mode**: A user actively pushed put a configuration to environments
- **Pull mode**: Clients get state from remote pull service automatically. Remote service is provided by a *pull server,* which acts as a central control and manager for the configurations

Azure Automation State configuration (DSC)

- A cloud-based implementation of PowerShell DSC, available as part of Azure Automation.

<u>Characteristics:</u>

- **Built-in pull server:** Built-in pull server in Azure Automation eliminates the need to set up and maintain your own pull server.
- **Management of all DSC artifacts:** Manage all DSC configurations, resources, and target nodes in single instance in the Azure portal or PowerShell
- **Ability to Import Reporting Data directly into Azure Log Analytics:** Can send this data to your Log Analytics workspace

DSC Configuration File

- *Configurations* are Windows PowerShell scripts that define a special type of function.

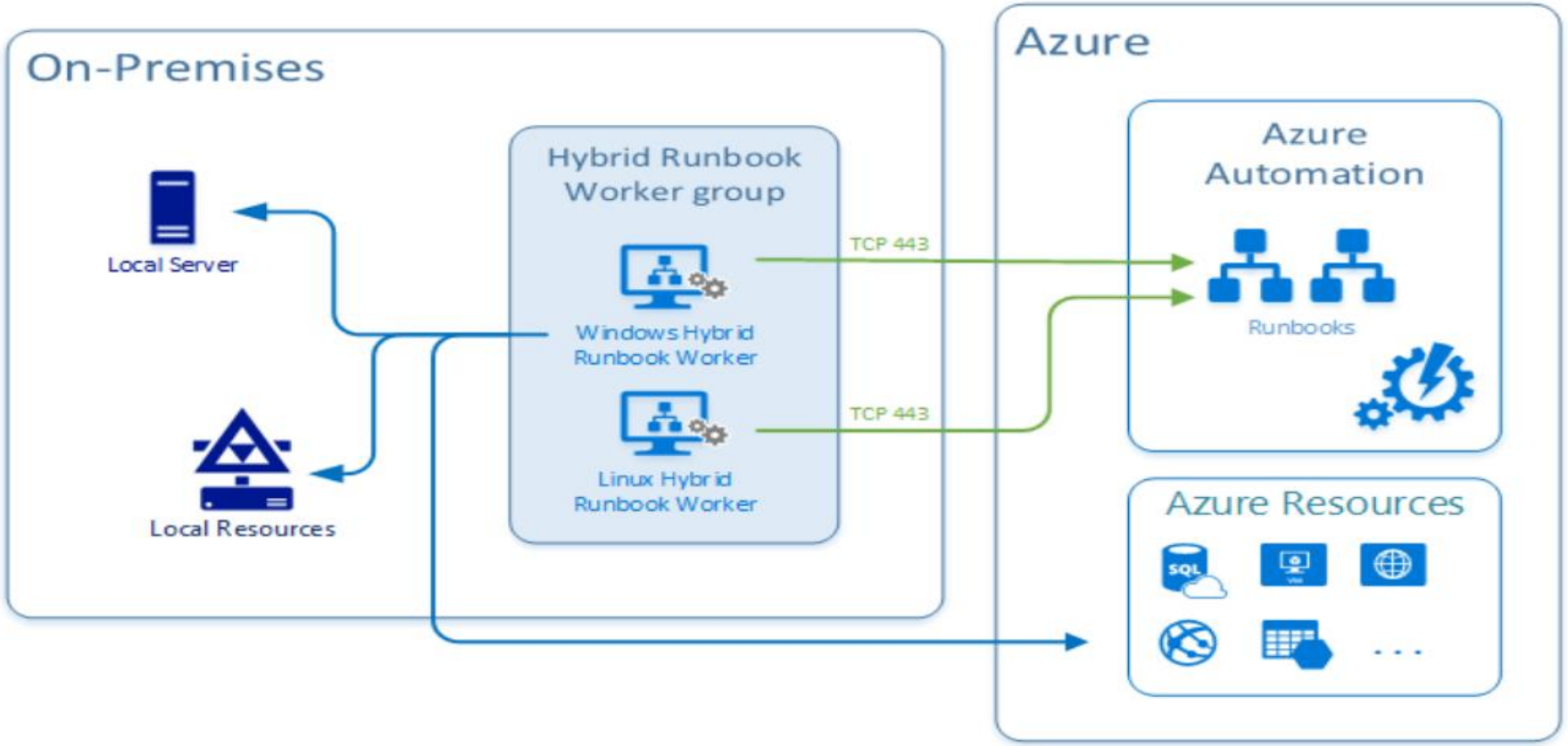
<u>Config File Elements:</u>

- Configuration block: Name of the configuration
- Node block: Define nodes being configured i.e. VMs and servers
- Resource blocks: Defines the actual configuration state for the nodes

```
configuration LabConfig
 {
     Node WebServer
     {
         WindowsFeature IIS
         {
             Ensure = 'Present'
             Name = 'Web-Server'
             IncludeAllSubFeature = $true
         }
     }
}
```

Hybrid Management

- *Hybrid Runbook Worker* feature of Azure Automation allows you to run runbooks that manage local resources in your private datacenter, on machines located in your datacenter.

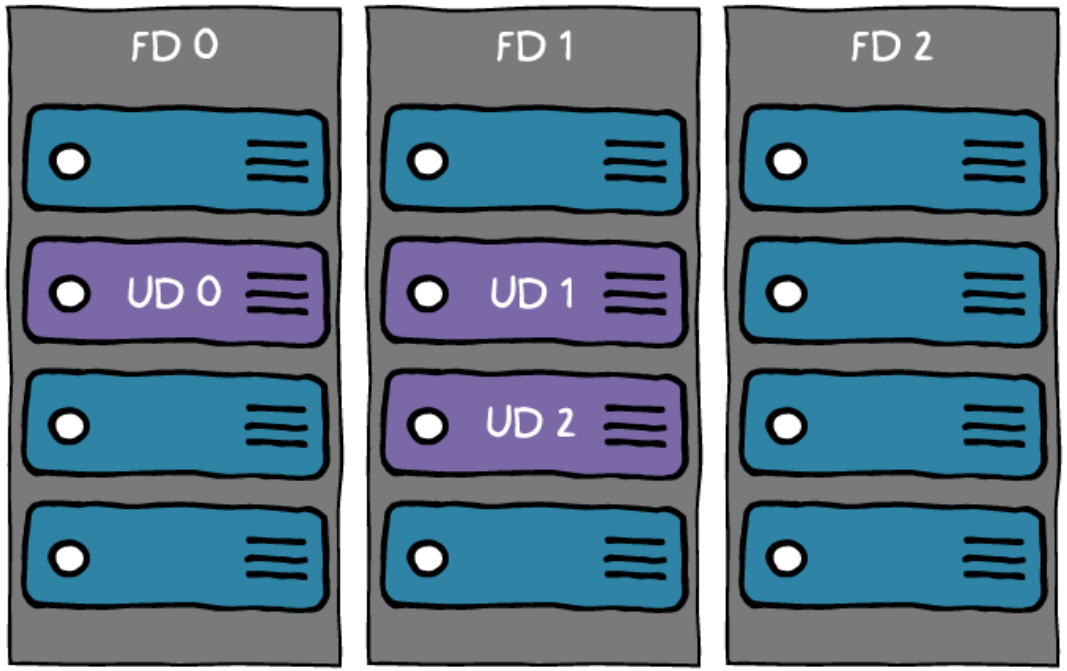


Azure Availability

- Availability of Azure VMs can be impacted by:
  - **An unexpected downtime**
  - **Planned maintenance events**
  - **Emergency maintenance events**
- To optimize availability of Azure VMs:
  - **Deploy two or more identically configured Azure VMs into the same availability set or different availability zones.**
  - **Use managed disks for Azure VMs deployed into availability sets and availability zones.**
  - **Keep track of Azure Scheduled Events to respond proactively to planned downtime.**
  - **Place each tier of multi-tier applications in a separate availability set or zone.**
  - **Use a load balancer in combination with availability sets and availability zones.**
  - **For individual VMs running critical workloads, use exclusively Premium Storage disks.**

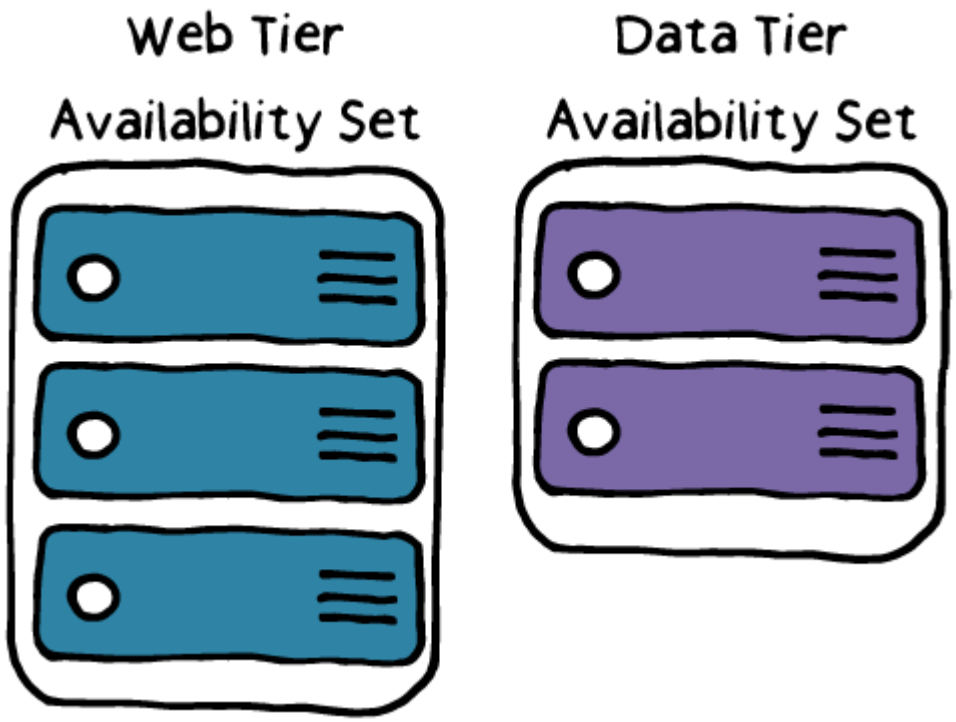
Availability Sets

- Offer 99.95% availability for workloads running on two or more VMs.
- Minimize impact of planned and unplanned downtime.
- Enforce placement of Azure VMs across separate racks in the same datacenter.
- Comprise of:
  - **Fault domains**
  - **Update domains**



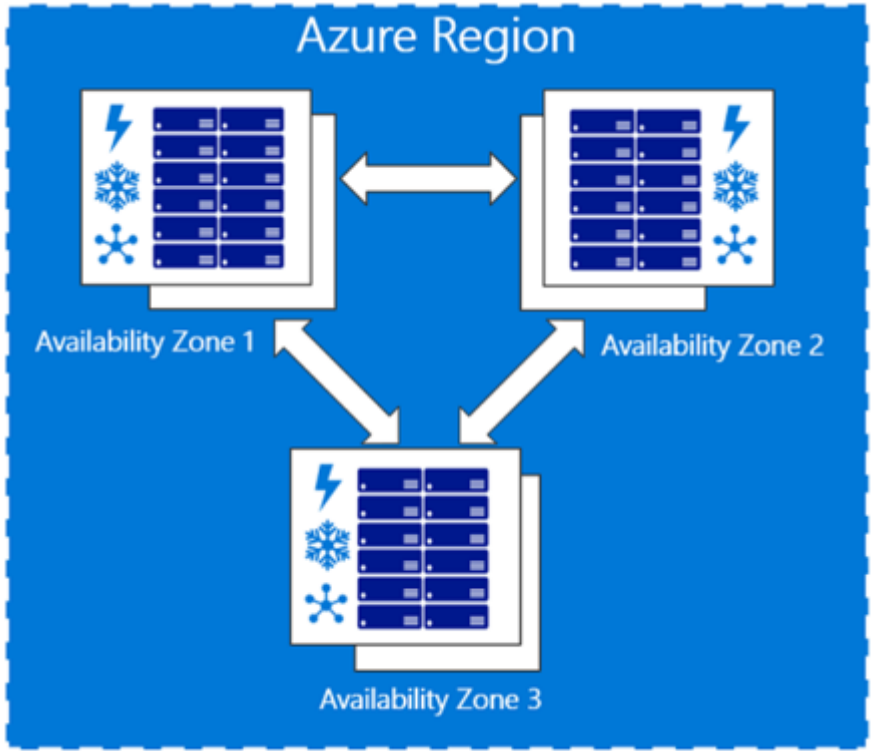
Multiple Availability Sets

- In a multi-tier solution, place each tier in a separate availability set:
  - **The solution remains functional in case of downtime of individual VMs (one per tier)**
  - **For example:**
    - **Web tier availability set**
    - **Data tier availability set**



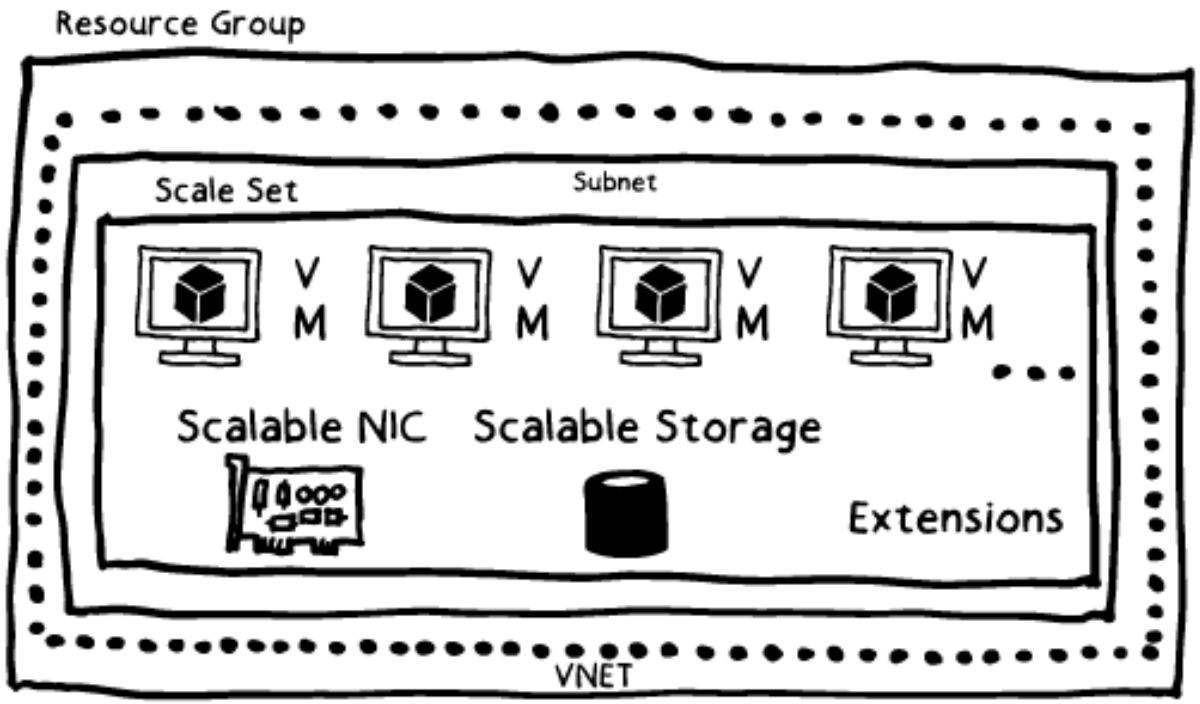
Availability Zones

- Offer 99.99% availability for workloads running on two or more VMs.
- Minimize impact of planned and unplanned downtime
- Enforce placement of Azure VMs across separate datacenters (zones) in the same Azure region.
- Implicitly comprise of:
  - **Fault domains**
  - **Update domains**



Virtual Machines Scale Sets (VMSS)

- Deploy into a subnet of a virtual network
- Provide access to individual VMs via NAT
- Scale for up to 1,000 VMs

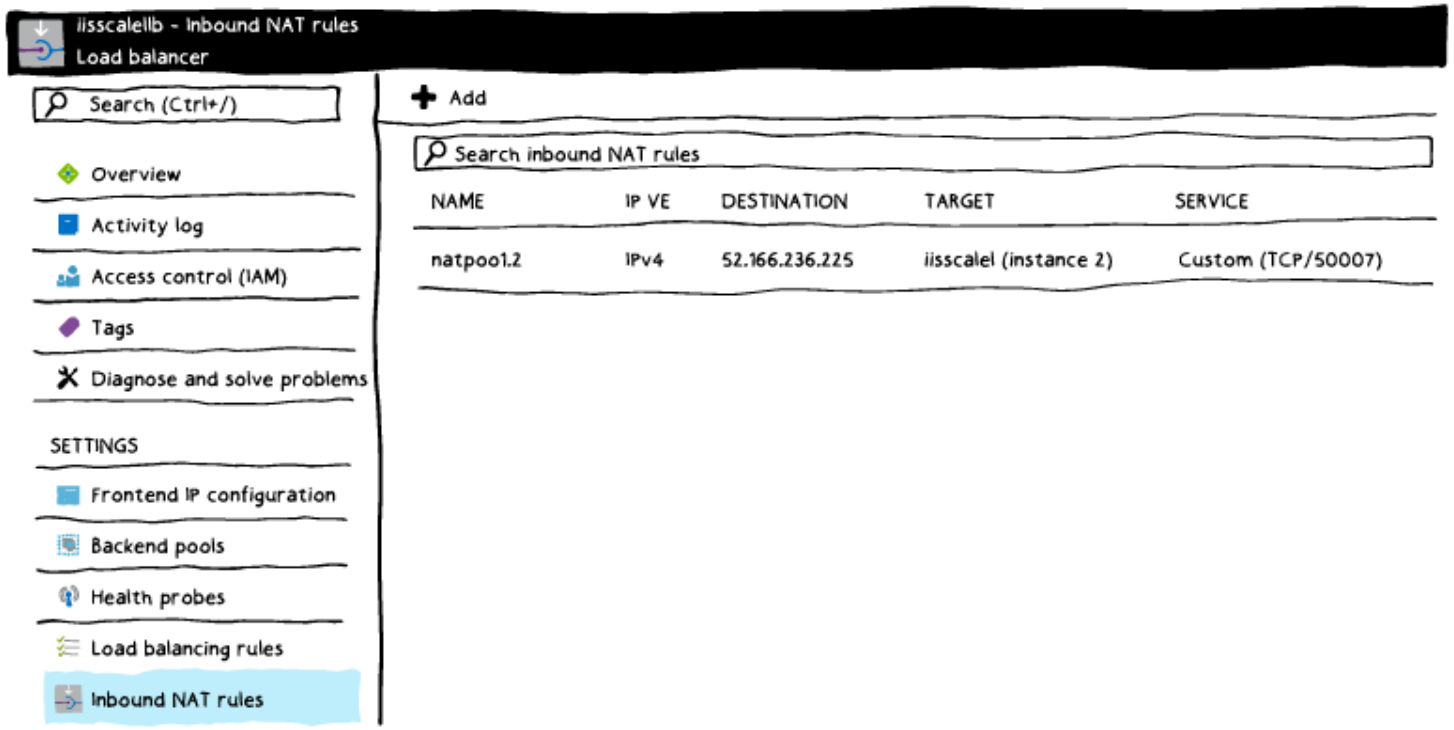


Virtual Machines vs. Virtual Machine Scale Sets

- Azure VM Scale Sets:
  - **Support auto-scaling.**
  - **Offer overprovisioning to accelerate the speed of deployment and scaling out.**
  - **Offer customizable policy controlling upgrade of all VMs in a scale set.**
  - **Support attaching new data disks to all VMs in a scale set (but not individual ones).**
  - **Do not support snapshots or image capture.**
  - **Do not support conversion from unmanaged disks to managed disks.**
  - **Do not support IPv6.**
- Azure VMs:
  - **Support custom configuration, including new or existing data disks.**
  - **Support snapshots and image capture.**
  - **Support conversion of unmanaged disks to managed disks.**
  - **Support IPv6.**

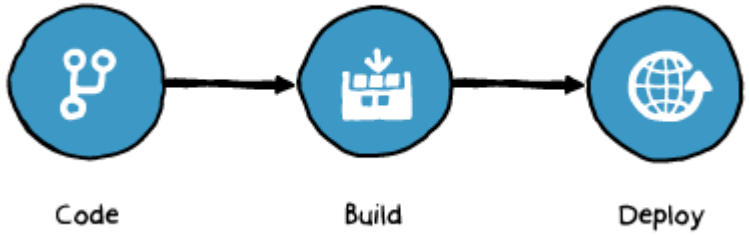
Virtual Machine Scale Set Considerations

- To connect to a VM Scale Set instance VM form Internet:
  - **Use inbound NAT rules of an Internet-facing load balancer in front of the VM Scale Set**
  - **e.g. RDP access via 52.166.236.225:50007**



Continuous Delivery in VMSS

- Use continuous integration/continuous delivery pipelines in Visual Studio Team Services to update an application running in a VMSS:
  - **Code update triggers a new build**
  - **The new built triggers VMSS update:**
    - **By creating and redeploying an updated OS image**
    - **By updating the application (via a VM extension)**
  - **Image based updates offer some benefits:**
    - **Predictability**
    - **Easy roll-back**
    - **Better scaling (no code to install on each VM as it is deployed)**

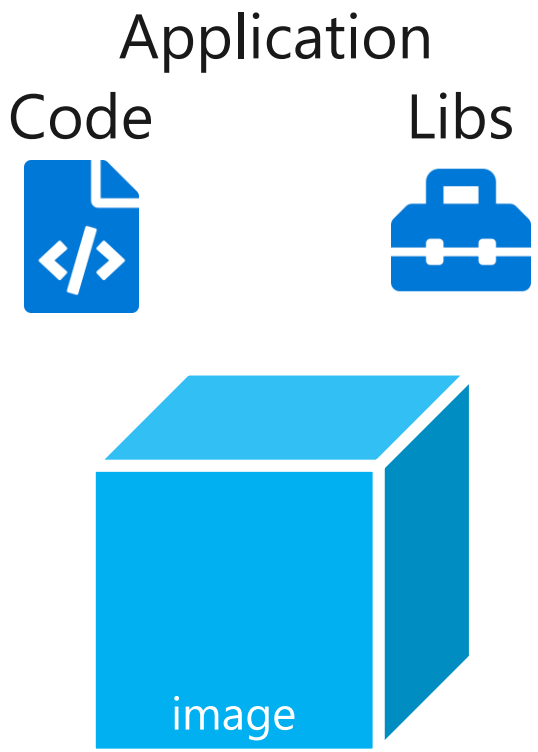


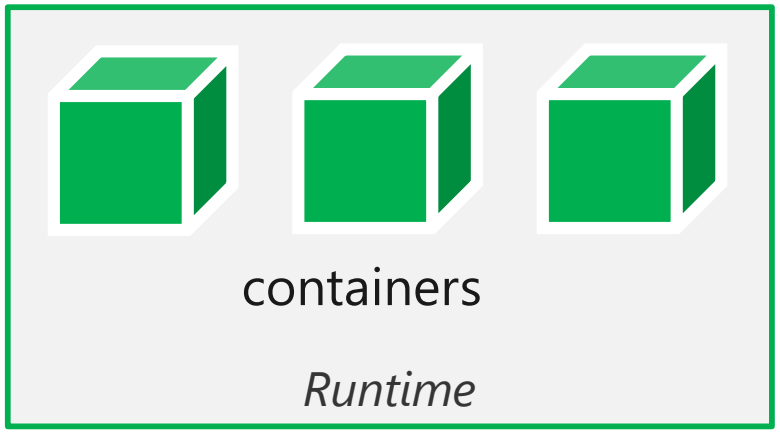
Large VM Scale Sets

- VM Scale Sets with 100+ VMs:
  - **Require the singlePlacementGroup property set to false**
  - **Require the use of managed disks.**
  - **Scale up to 1,000 VMs when based on Azure Marketplace images.**
  - **Scale up to 300 VMs when based on custom images.**
  - **Require the use of Azure Load Balancer Standard SKU for layer-4 load balancing.**
  - **Support Azure Application Gateway for layer-7 load balancing.**
  - **Require sufficiently sized IP subnet (all VMs in the same VMSS are deployed into the same IP subnet)**
  - **Might require an increase in vCPU subscription level quotas.**

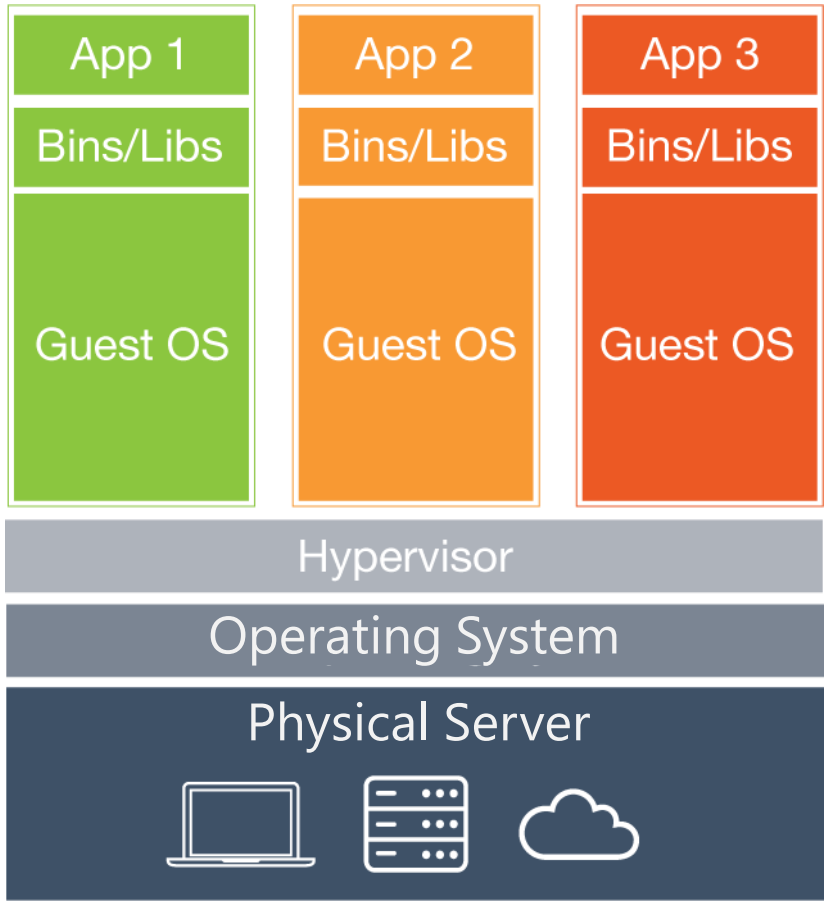
What Are Containers?

- A **lightweight**, stand-alone **bundle** that includes everything needed to run your app
- **Immutable** – build, deploy, replace
- Optimized for **fast deployment** & updating
- Containers have **isolation** – can run alongside existing containers



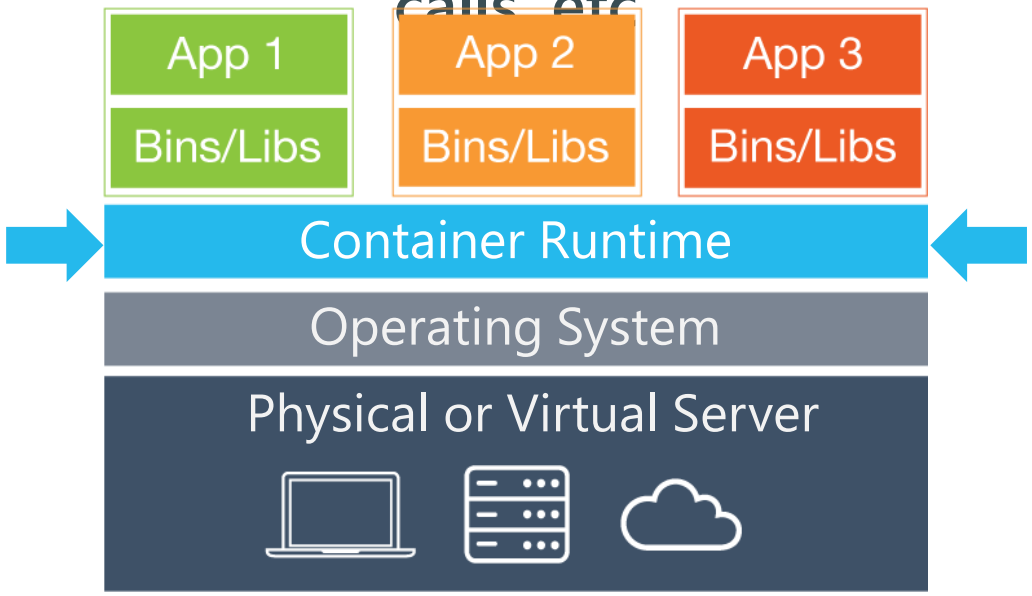


Containers vs Virtual Machines



Virtual Machines

Containers are isolated, but share OS and, where appropriate, binaries/libraries, system calls, etc



Containers

Docker and Containers

- Containers have been around for **many years** (80s)
  - **Developed on top of the Linux Kernel (cgroups)**

- **Docker Inc.** did not invent them
  - **They created open source software to build and manage containers**
  - **First release 2013**

- Docker made containers **easy**

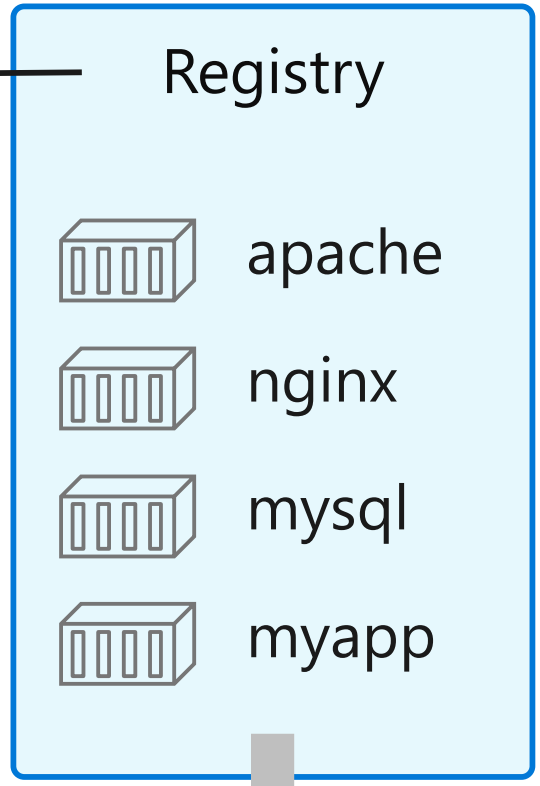
- Docker is the facto standard **container format** and set of **tools**
  - **Docker CLI, Docker Engine, Docker Swarm, Docker Compose, Docker Machine etc. (more later!)**

Docker != Containers

Terminology - Basics

- Container
  - A runtime instance of an image. Represents the execution of a single application, process, or service
- Image
  - A package with all the files, data, dependencies & information needed to create a running container
- Docker Engine
  - Container runtime installed on a host machine (aka ‘Docker host’ or ‘container host’).
    Can be Linux, Windows Server or Desktop OS (e.g. Windows 10). *Note. Other container runtimes exist*
- Registry
  - A service that provides access to stored image repositories, typically a server or cloud service
- Repository
  - A collection of related Docker images, labelled with tags to indicate version or other metadata
- Dockerfile
  - A text file that contains instructions for how to build a Docker image

Stores many static images

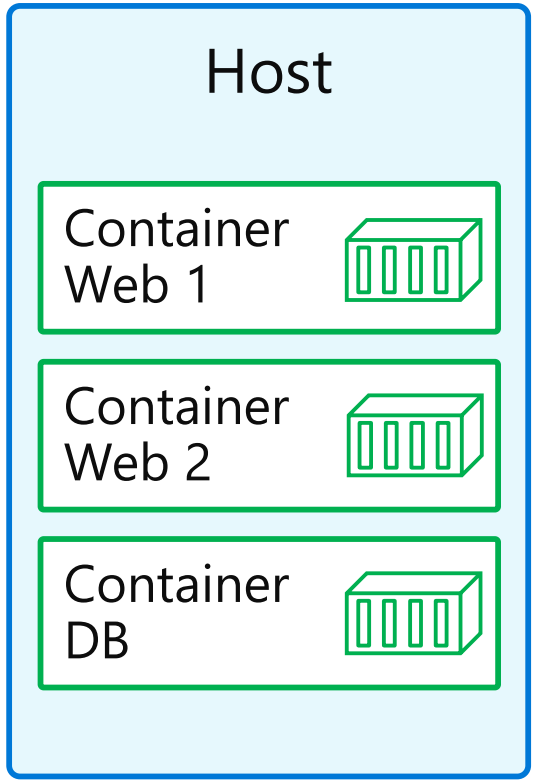


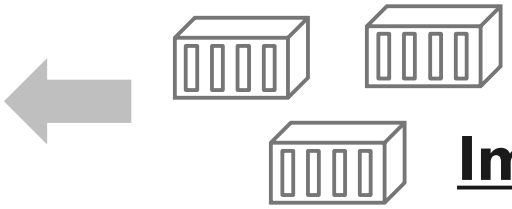
- Default public registry **Docker Hub (hub.docker.com)**
- Private registry e.g. **Azure Container Registry**

Containers

Running instance of image

As app processes

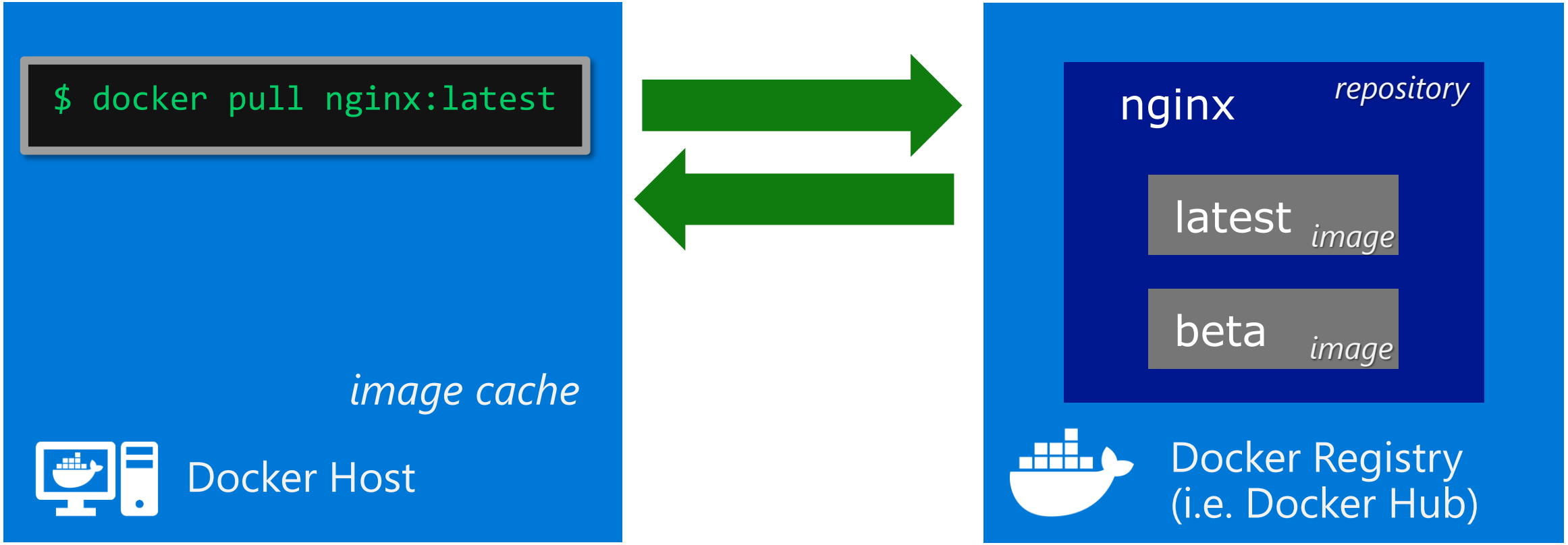




Images

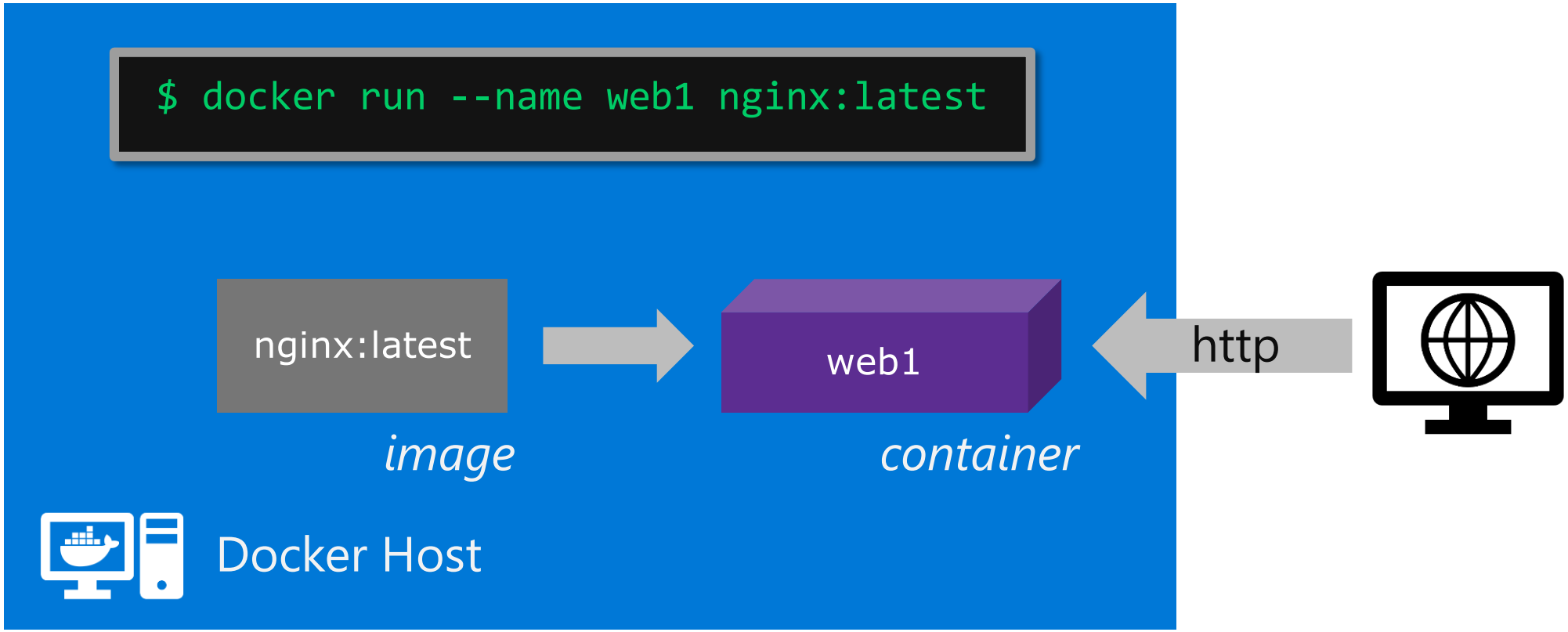
Static, persisted "blobs" of data

Docker - Getting Started



Pulling an image from a registry

Docker - Getting Started



Starting a running container from an image

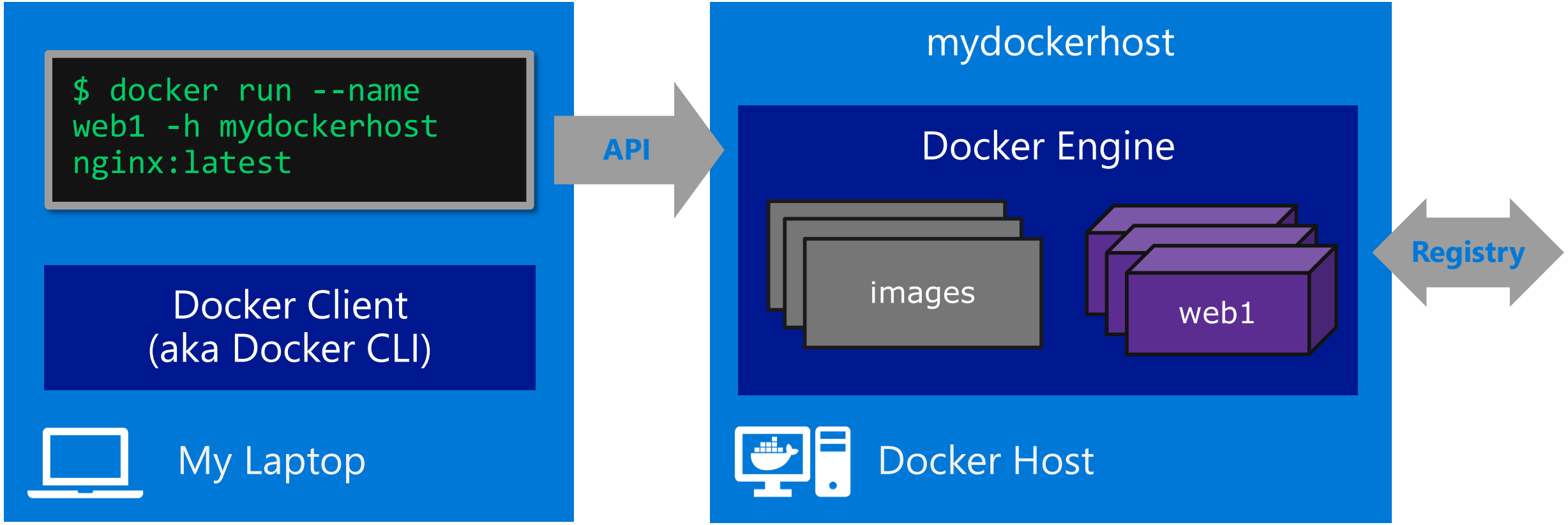
Docker Image Naming

REFERENCE

REGISTRY[:PORT]/REPOSITORY/NAME:TAG

`docker.io/mysql/mysql-server:latest`	Fully qualified
`mysql/mysql-server:latest`	Same as above, as Docker hub is default registry
`mysql/mysql-server`	Same, :latest is implied when no tag supplied
`mysql`	Same, as mysql lives in special “top level” repo
`myreg.azurecr.io/myrepo/myapp:v23`	Azure Container Registry example *Pull myapp v23 from the myrepo repository in myreg registry*
`myreg.azurecr.io/myapp`	Azure Container Registry example, with no repo *Pull myapp latest from the top level of myreg registry*

Docker - Client & Remote Hosts



Docker Client & Engine interaction

Docker CLI - Run Command

Many, many options

```
$ docker run --name mycache redis

$ docker run -it ubuntu

$ docker run -d -p 80 nginx

$ docker run -e MYSQL_ROOT_PASSWORD=r00t mysql

$ docker run -it --rm busybox

$ docker run busybox ls -la
```

`--name` = assign name to container

`-it` = run interactively with TTY

`-d` = detached (in background)

`-p 80` = expose or map ports

`-e` = set environmental variables

`--rm` = remove container when it exits

Pass command/start-up executable to container

Note. Most of the time `docker pull` is unnecessary as `docker run` will automatically pull any image it can not find locally

Docker CLI - Other Common Commands

REFERENCE

docker ps - List Containers

```
$ docker ps
$ docker ps -a
$ docker ps -f status=paused
```

docker images - List Images

```
$ docker images
$ docker images node
$ docker images -f dangling=true
```

docker rm - Remove Containers

```
$ docker rm myoldcontainer
$ docker rm -f mycontainer
$ docker rm 8fd
```

docker rmi - Remove Images

```
$ docker rmi mysql
```

docker kill - Stop Containers

```
$ docker kill mycontainer
```

docker inspect, log & top - Debug

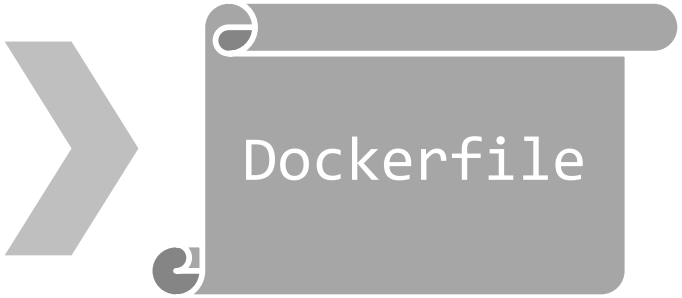
```
$ docker inspect mycontainer
$ docker inspect myimage
$ docker logs mycontainer
$ docker top mycontainer
```

Note. There is a [newer command format introduced in Docker 1.13](), but these older commands are still the most commonly used

Building Images - Basics

- A file called a Dockerfile is used to build images.
  Note. The name of this file is typically just `Dockerfile` (no extension)
- A Dockerfile is simply a set of instructions on how to build the image, much like a script.

- Laying out the filesystem & directory structure the app is expecting
- Copying in binaries and configs
- Installing any pre-req packages, libraries and software
- Running any custom set-up commands
- Setting environmental variables
- Defining the command/executable to start your application



Dockerfile - By Example

Simple Dockerfile for a Node.js application

```
FROM node:v6
WORKDIR /app

COPY "server.js" .
COPY "package.json" .
RUN npm install

EXPOSE 3000
CMD "npm start"
```

← Use v6 of public official node image as starting point

← Set a working directory inside the image

← Copy our application files into image. Note the dot which is our working directory we set above

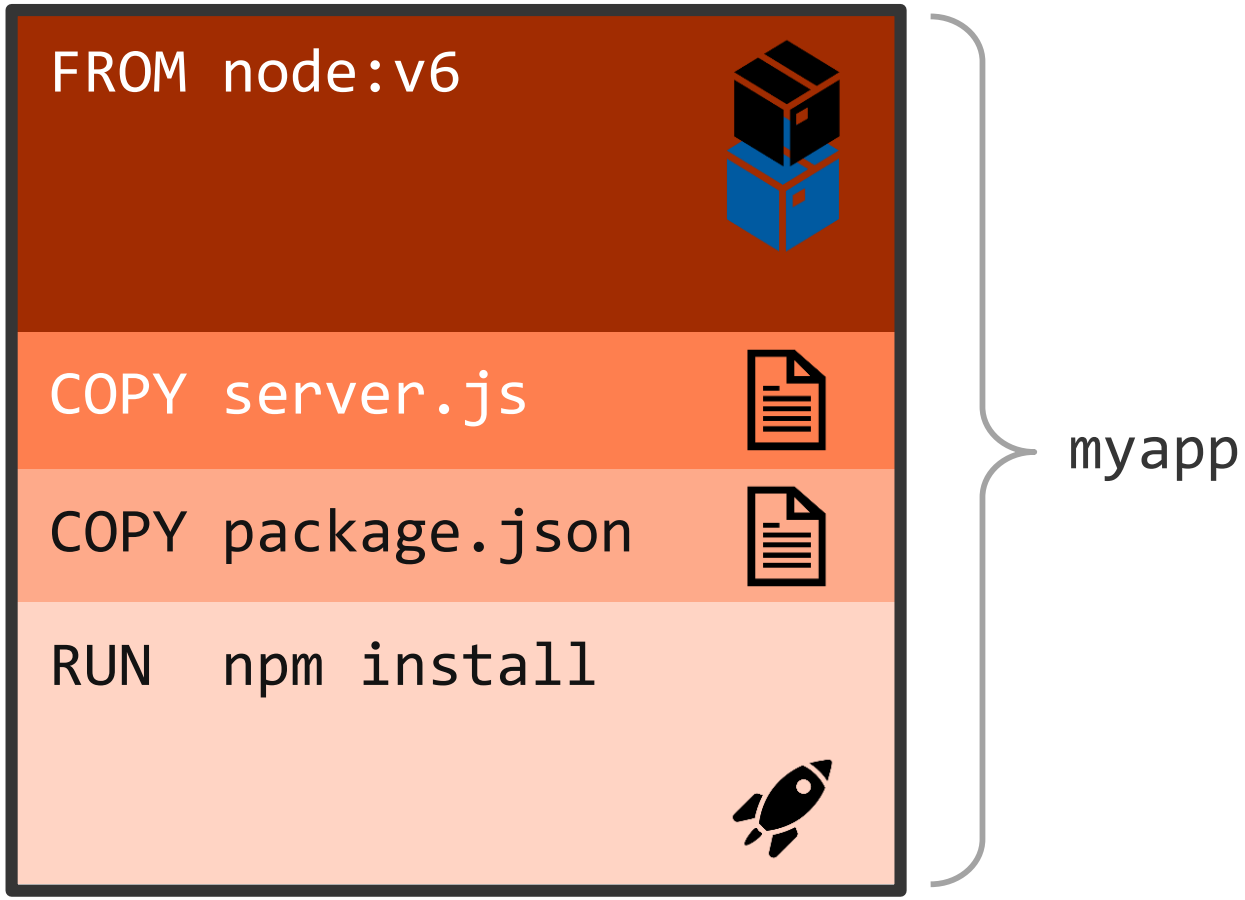
← Run commands inside the image, e.g. install our app

← Tell Docker what ports our app will listen on

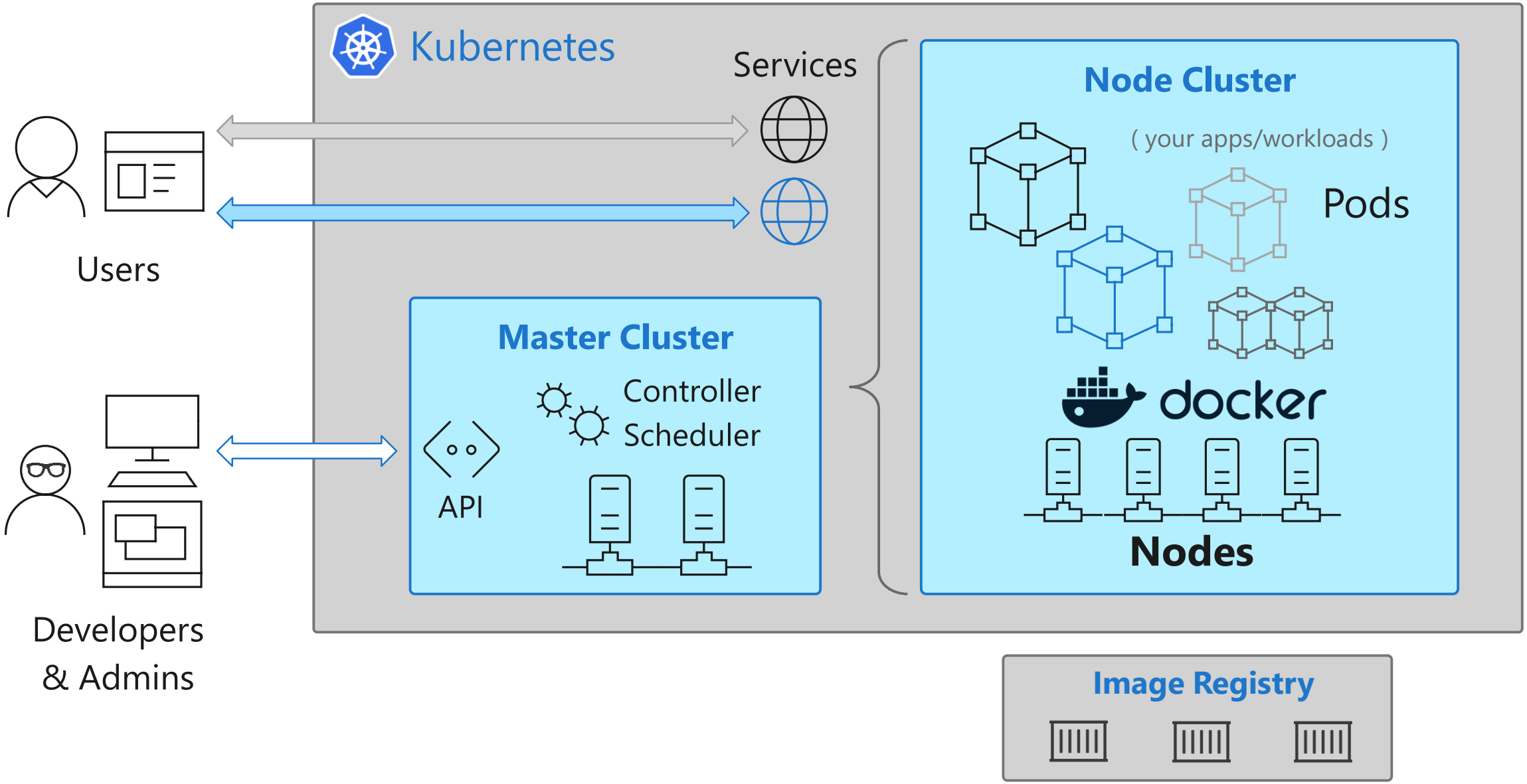
← How to start our app when container is created from img

Image Layers

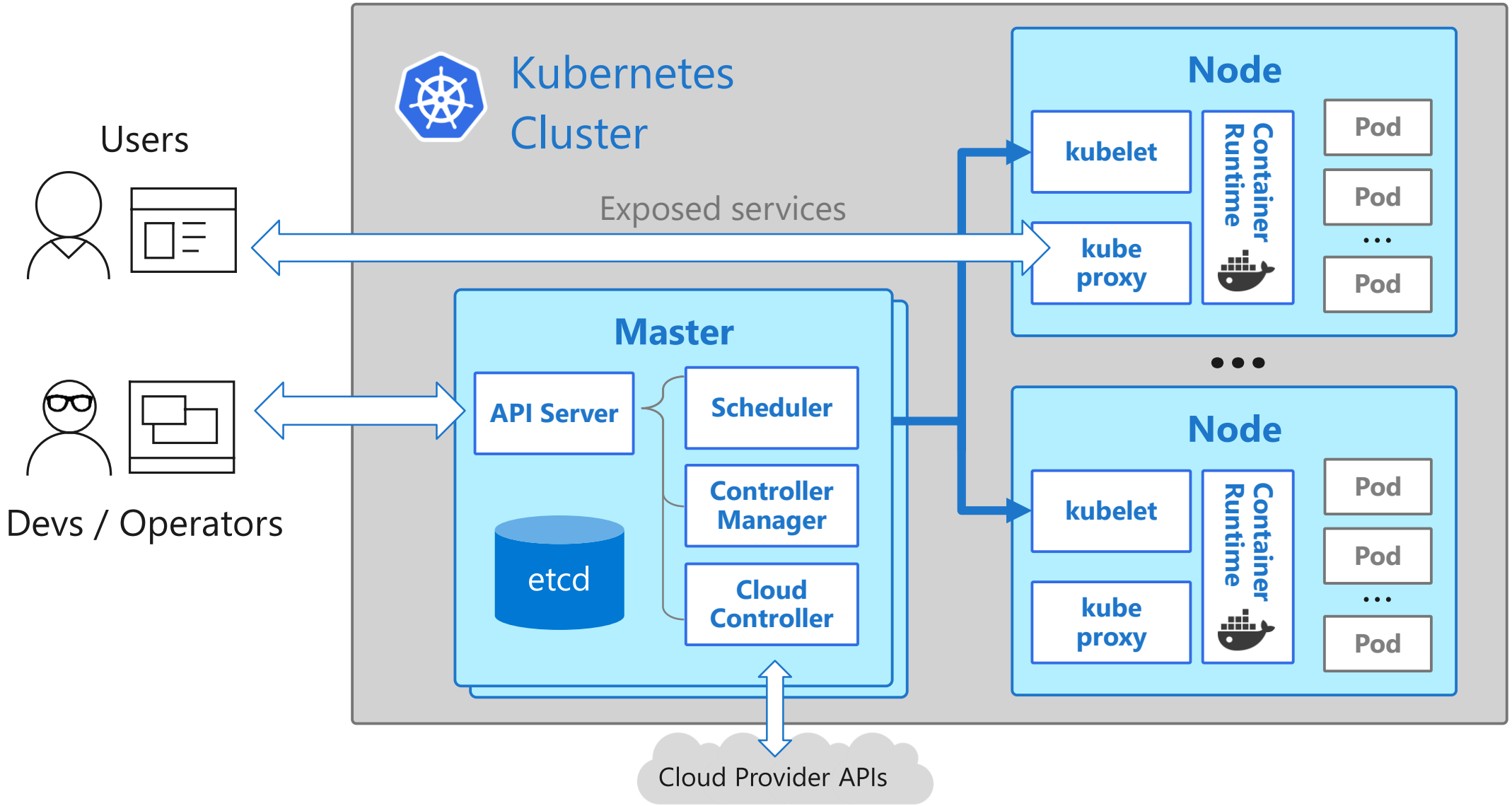
- Images are built of immutable filesystem **layers**
- This lets Docker cache & share layers both on the engine and in the registry
- All images use **FROM** to specify a base image & layer themselves on top of this base
- Each **COPY** and **RUN** instruction in your Dockerfile creates a new layer containing that chunk of the filesystem



Kubernetes – Simplified Architecture



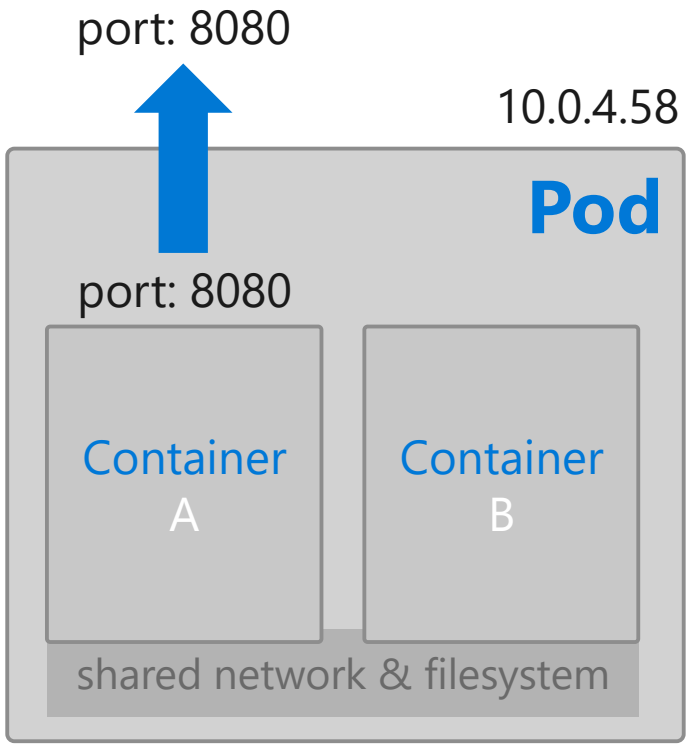
Kubernetes – Internal Architecture



Pods

Fundamental building block of Kubernetes

- *Pods* run one or more containers
- Containers in a pod share network/storage
- *Pods* each have their own IP address
- *Pods* expose one or more ports
- *Pods* are scheduled and run on a *Node*



example Pod

- Pods are the primary way of running your workloads in Kubernetes

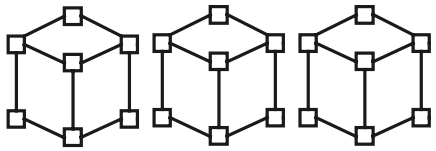
Deployments & Replica Sets

Scale and run pods across multiple nodes

- *Deployments* describe a replicated set of *Pods*
- A *Deployment* represents desired state
  - Rolling updates used to safely roll out changes
- You scale *Deployments* up & down
- *Deployments* typically run stateless workloads
- *Deployments* use *ReplicaSets*

```
Deployment
- name: MyApp
- replicas: 3
```

```
ReplicaSet
- replicas: 3
```



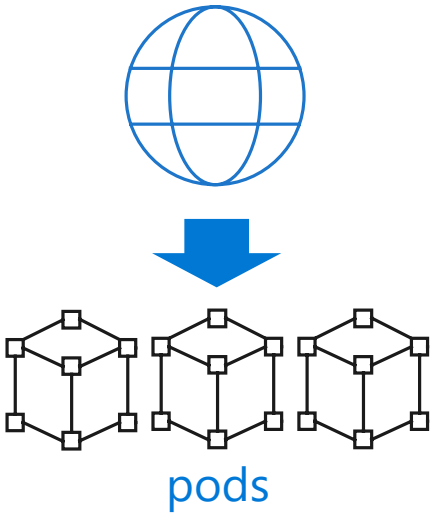
pods

- Deployments let you run & scale stateless workloads in Kubernetes

Services

Network access to Pods

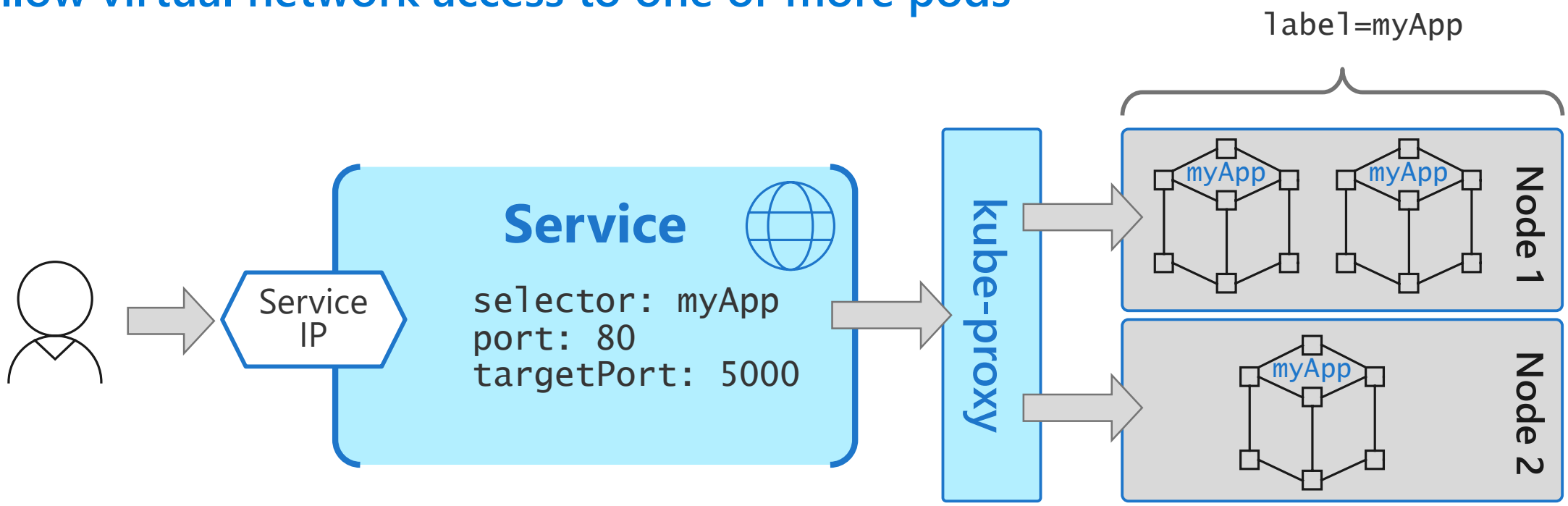
- *Pods* are ephemeral - they can move/die/change without notice. In general, users do not directly access *Pods*
- *Services* are an abstraction which defines a logical set of *Pods* and a policy by which to access them
- *Services* use labels and selectors to map to *Pods*
- *Services* are assigned IP addresses and DNS names



- Services are how you connect to Pods over the network

Services – Illustrated

Allow virtual network access to one or more pods



EXTERNAL

LoadBalancer

Uses cloud provider to present an external load-balanced IP

INTERNAL

ClusterIP

Internal virtual IP, only accessible by other pods/services

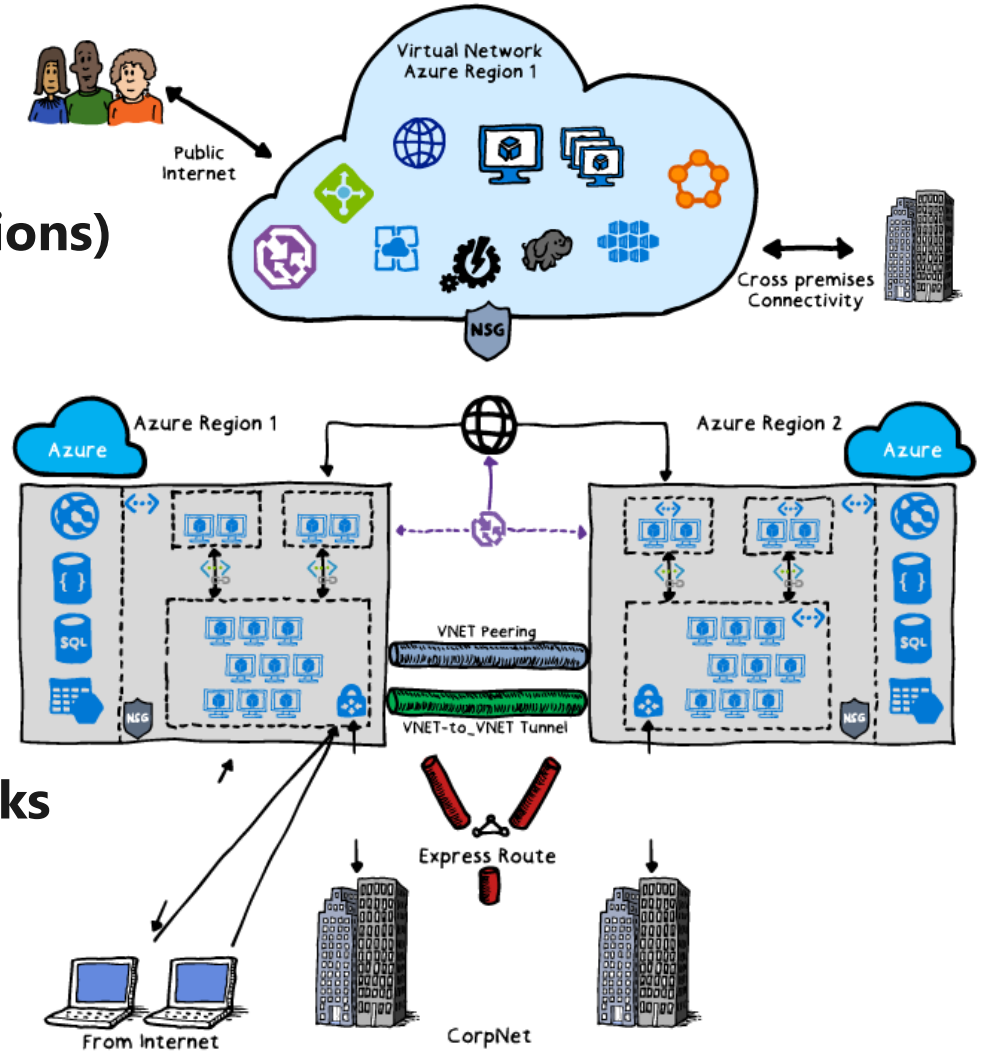
Note. Uses 'round robin' to select pods

Azure Virtual Network (VNET) Architecture

- Virtual networks constitute a core component of Azure networking:
  - **Facilitate connectivity:**
    - **Between Azure VMs in the same Azure region**
    - **Between Azure VMs across Azure regions**
    - **Between on-premises networks and Azure VMs**
  - **Support fundamental networking features:**
    - **IP addressing**
    - **DNS name resolution**
    - **Network traffic filtering**
    - **Routing**

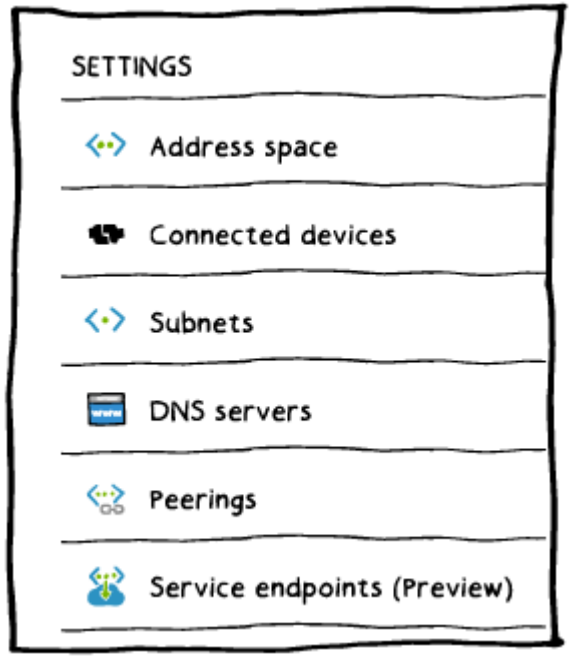
Multi-Region Virtual Network Architecture

- Virtual network communication flows:
  - **Within a virtual network**
  - **Between virtual networks (the same or different regions)**
  - **Hybrid connectivity with on-premises networks**
  - **From a virtual network to Internet (outbound)**
  - **From Internet to a virtual network (inbound)**
- Virtual network connectivity:
  - **VNET Peering: between virtual networks**
  - **ExpressRoute: Hybrid only**
  - **Site-to-Site VPN: hybrid and between virtual networks**
  - **Load Balancers:**
    - **Traffic Manager: global load balancing from Internet**
  - **Network Security Groups: traffic filtering**



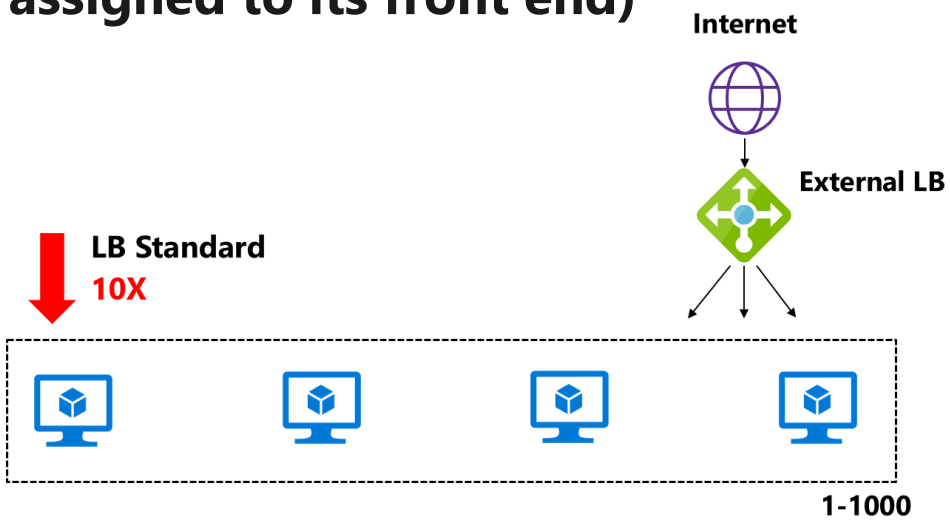
VNETs & Subnets

- **Networking Topology**
  - **Define one or more VNETs in an Azure region and provide a non-overlapping IP address space for each**
  - **Define one or more subnets within a VNET and provide an IP address range for each**
  - **Configure NSGs for each subnet**
  - **Deploy Azure VMs into subnets**
- **Subnet IP addressing:**
  - **Static and dynamic (default) IP address assignments are supported**
  - **5 IP addresses in each subnet are reserved for internal use**
- **Public IP addressing:**
  - **Required for inbound traffic from Internet**
- **Private IP addressing:**
  - **Used for cross-premises and within/cross-VNET communication.**
- **DNS name resolution:**
  - **Configurable on the VNET and individual vNIC level**
  - **Azure provided DNS or custom DNS configuration**



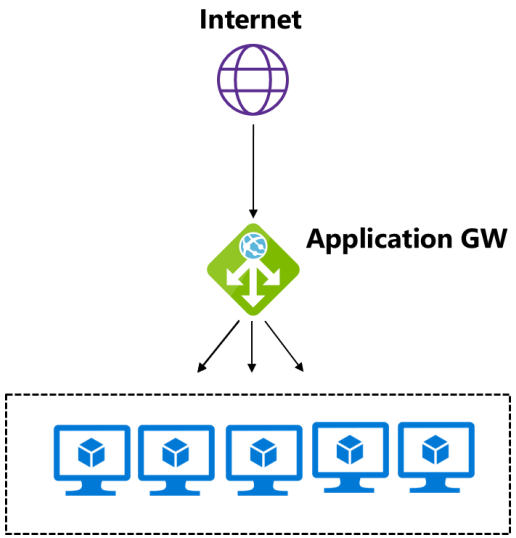
Load Balancing Solutions

- Azure offers a number of Azure load balancing services:
  - **Azure Load Balancer**
  - **Azure Application Gateway**
  - **Azure Marketplace Load Balancing Appliance**
  - **Azure Traffic Manager**
- Azure Load Balancer – managed layer 4 (TCP/UDP) load balancer:
  - **Internal or External (private IP or public IP address assigned to its front end)**
  - **Basic SKU (free service):**
    - **Up to 100 Azure VMs in the same availability set**
    - **NSGs are optional**
  - **Standard SKU (paid service):**
    - **Up to 1,000 Azure VMs with availability zones support**
    - **NSGs are mandatory**
    - **Support for HA ports**



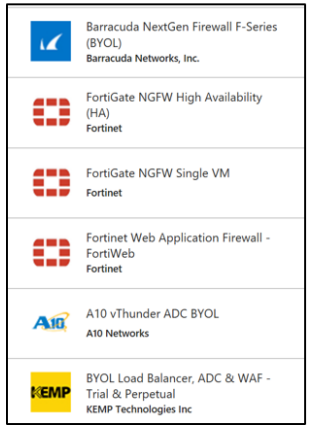
Azure Application Gateway

- Managed layer 7 (HTTP/HTTPS) load balancer:
  - **SSL Offloading**
  - **HTTP to HTTPS redirect**
  - **Cookie Affinity**
  - **URL Based Routing**
  - **Web Application Firewall (WAF) protects against:**
    - **SQL Injection**
    - **Cross-site scripting**
    - **Protocol violations**
    - **Generic attacks**
    - **HTTP rate limiting**
    - **Scanner detection**
    - **Session fixation**
    - **LFI/RFI**



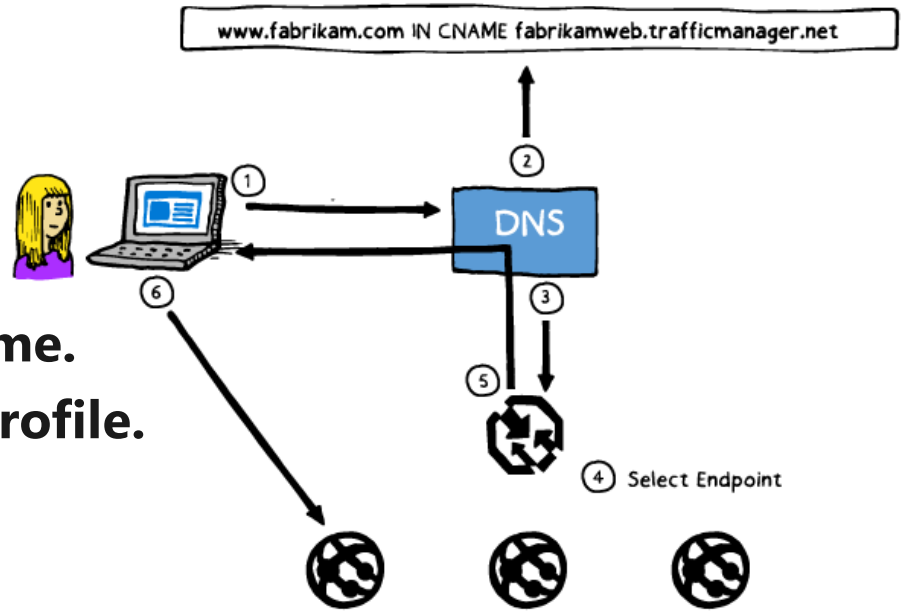
Azure Load Balancing Marketplace Appliances

- Supplement built-in Azure load balancing services:
  - **BYOL: allows customers to reuse their existing licenses when migrating to Azure**
  - **Pay-Per-Use: the usage cost include licensing fees**



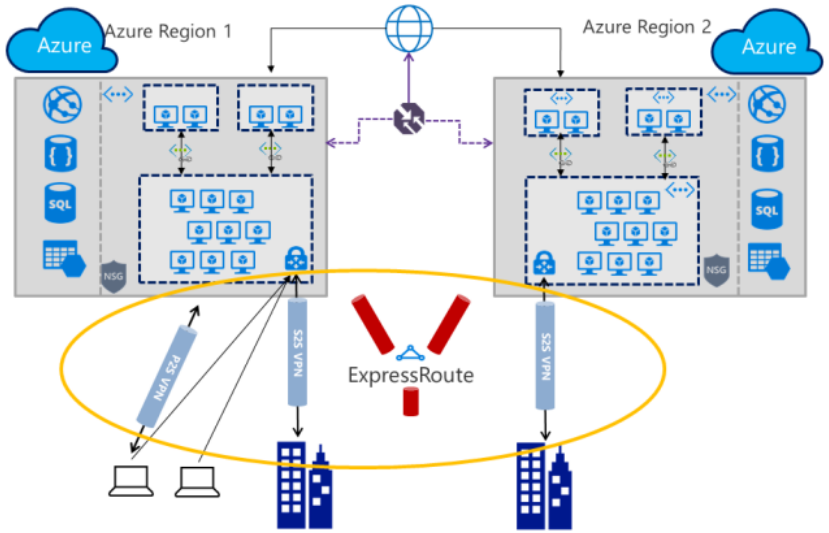
Azure Traffic Manager

- A managed DNS-based global load balancing service
- Supports different traffic distribution algorithms:
  - **Weighted: distributes requests across endpoints according to custom weight values**
  - **Failover: always directs to the primary endpoint but supports failover to a secondary**
  - **Performance: directs requests to the closest endpoint**
  - **Geography: directs requests to the designated region**
- Traffic Manager workflow:
  - **1. A user request to the company DNS name.**
  - **2. The company DNS name to a Traffic Manager DNS name.**
  - **3. The Traffic Manager DNS name to a Traffic Manager profile.**
  - **4. Processing of the Traffic Manager profile rules.**
  - **5. An endpoint DNS domain name to the user.**
  - **6. The user request to the endpoint.**



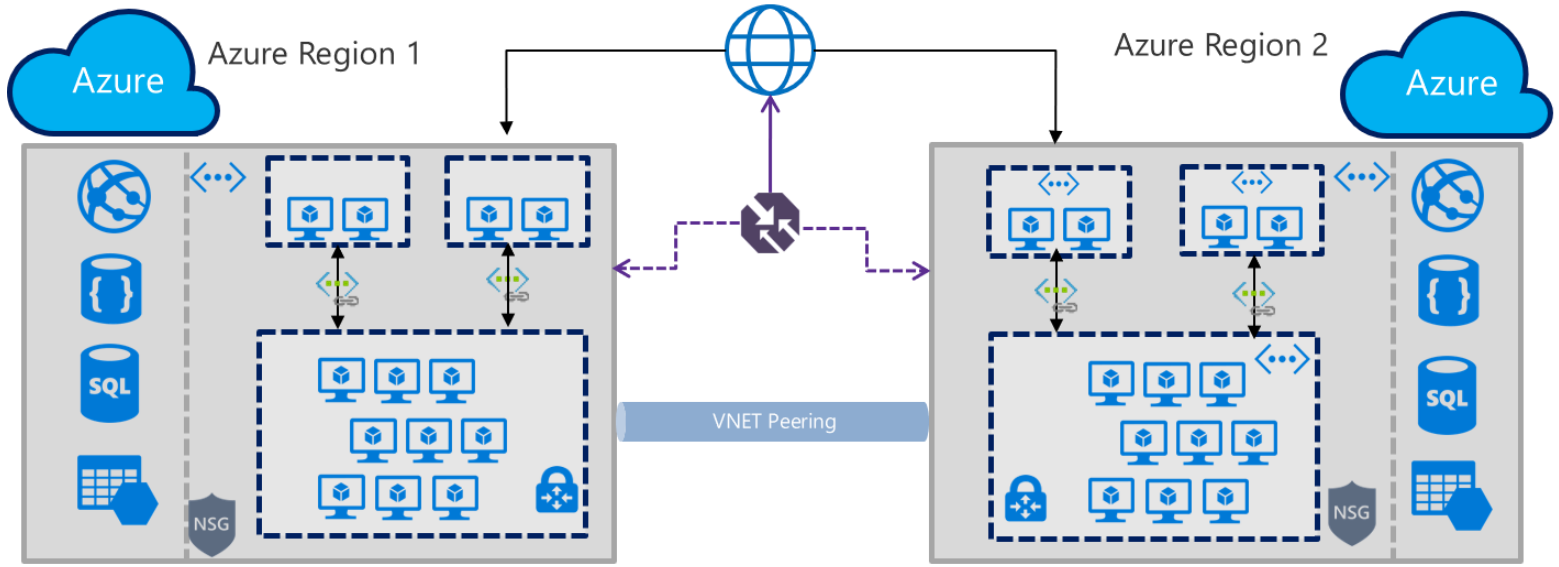
On-Premises to Azure Connectivity

- There are three methods that facilitate hybrid connectivity to Azure:
  - **ExpressRoute:**
    - **Private connectivity between on-premises networks and Azure VNETs**
    - **Private connectivity between on-premises networks and Azure public services**
    - **The lowest latency and the highest throughput**
    - **Superior redundancy via multiple circuits**
    - **Support for ExpressRoute-VPN co-existence**
  - **Site-to-Site VPN:**
    - **VPN connectivity between on-premises networks and Azure VNETs**
    - **Built-in security provided by IPSec tunneling**
    - **Support for active-active configuration via BGP**
  - **Point-to-Site VPN:**
    - **VPN connectivity between individual computers and Azure VNETs**
    - **Support for Window and macOS**
    - **Authentication via Active Directory, RADIUS, or certificates**



VNET Peering

- Direct connection between two Azure VNETs:
  - **The same or different Azure regions**
  - **Traffic routed over Microsoft backbone**
  - **Support for connections to classic VNETs**
  - **No built-in encryption**
  - **Automatic routing**



Network Security Groups

- Virtual network-based firewall mechanism:
  - **Can be associated to a subnet of a VNET and to a network adapter of an Azure VM**
  - **There is a limit of one NSG assignment per subnet and one per a network adapter**
  - **NSG blocks all traffic that is not explicitly allowed**
  - **NSG rules determine which traffic should be allowed or blocked:**
    - **Inbound rules**
    - **Outbound rules**
  - **Rules are processed in the order corresponding to their priority**
  - **Processing stops once the match is found**
  - **A single NSG can contains up to 1,000 rules**

Default Network Security Group Rules

- **Have the lowest priority so can be overridden by custom rules but cannot be deleted**
- **Allow outbound connectivity to Internet and block inbound connectivity from Internet**

Inbound

NAME	PRIORITY	SOURCE IP	SOURCE PORT	DESTINATION IP	DESTINATION PORT	PROTOCOL	ACCESS
ALLOW VNET INBOUND	65000	VIRTUAL_NETWORK	*	VIRTUAL_NETWORK	*	*	ALLOW
ALLOW AZURE LOAD BALANCER INBOUND	65001	AZURE_LOADBALANCER	*	*	*	*	ALLOW
DENY ALL INBOUND	65500	*	*	*	*	*	DENY

Outbound

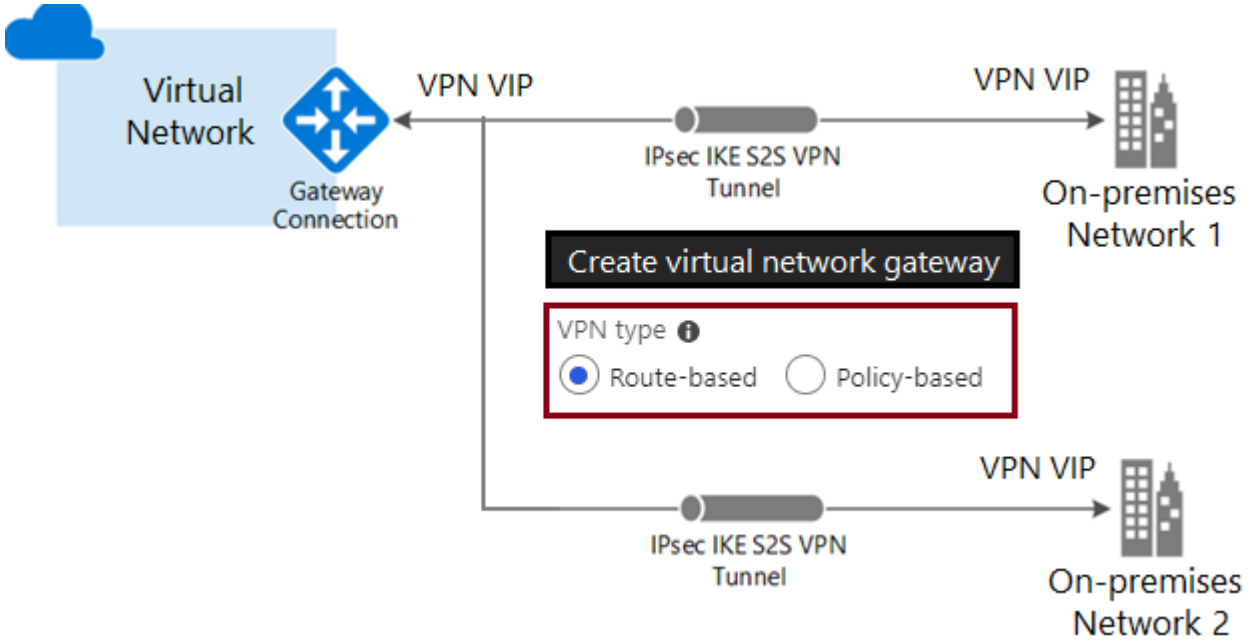
NAME	PRIORITY	SOURCE IP	SOURCE PORT	DESTINATION IP	DESTINATION PORT	PROTOCOL	ACCESS
ALLOW VNET OUTBOUND	65000	VIRTUAL_NETWORK	*	VIRTUAL_NETWORK	*	*	ALLOW
ALLOW INTERNET OUTBOUND	65001	*	*	INTERNET	*	*	ALLOW
DENY ALL OUTBOUND	65500	*	*	*	*	*	DENY

Forced Tunneling and securing access to PaaS services

- Forced Tunneling:
  - **Routes Internet-bound traffic from an Azure VNET via an on-premises network**
  - **Can use Site-to-Site VPN or ExpressRoute**
- Securing Access to PaaS services:
  - **Relies on service endpoints**
  - **Available for:**
    - **Azure Storage**
    - **Azure SQL Database**
    - **Azure Database for PostgreSQL server**
    - **Azure Database for MySQL server**
    - **Azure Cosmos DB**
    - **Azure Key Vault**

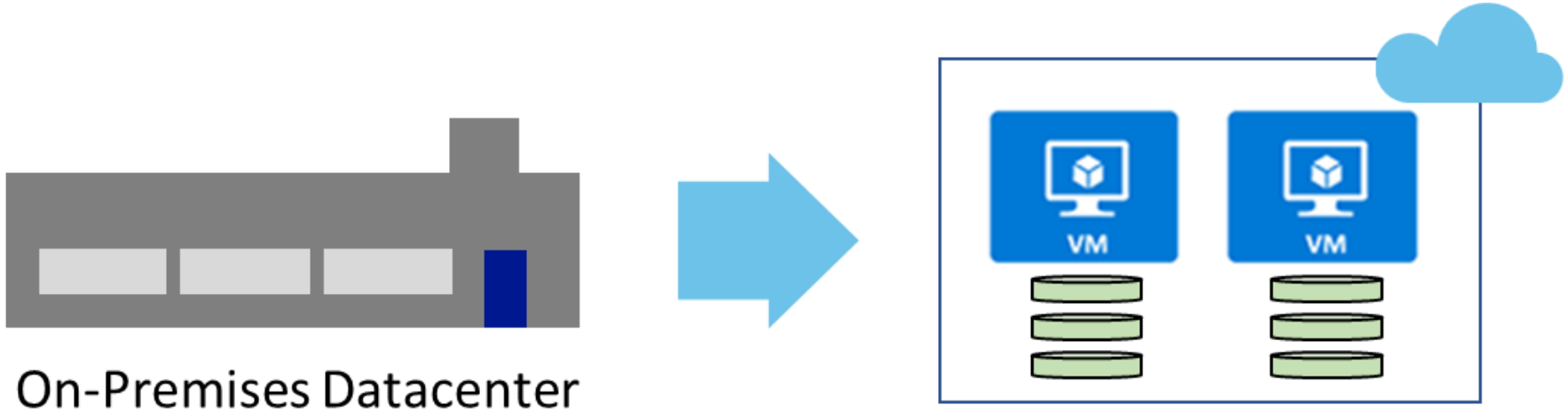
Site-to-Site VPN Connections

- IPSec IKEv1 or IKEv2 VPN tunnels:
  - **Cross-premises**
  - **VNet-to-VNet**
  - **Multi-site:**
    - **Requires a route-based VPN**



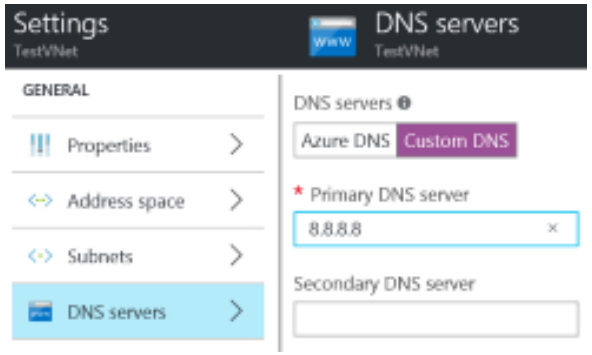
Site-to-Site Scenarios

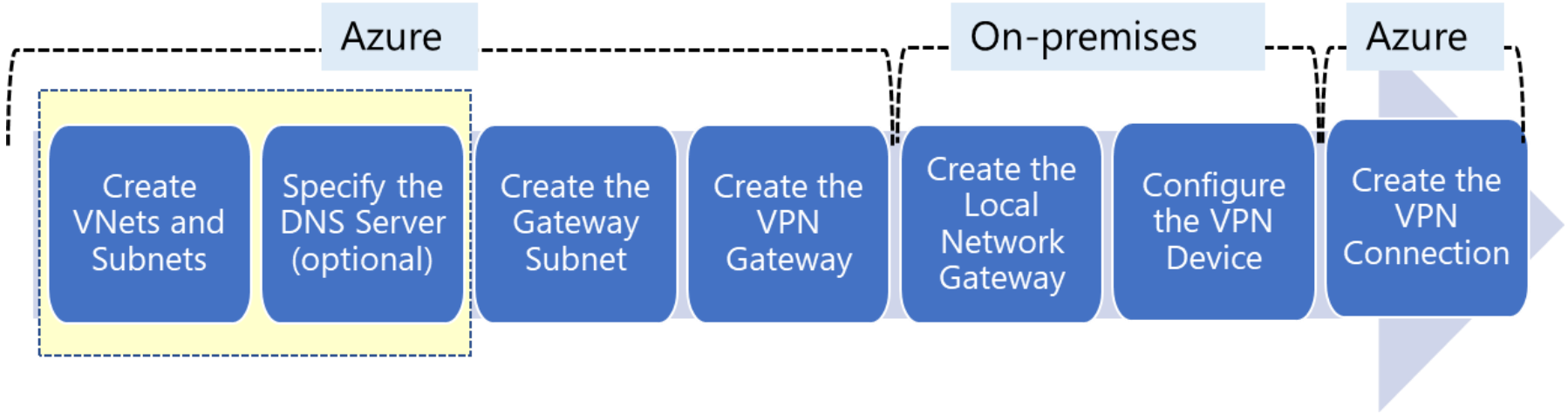
- Capacity On-Demand
- Strategic Migration
- Disaster Recovery



Implementing Site-to-Site VPN

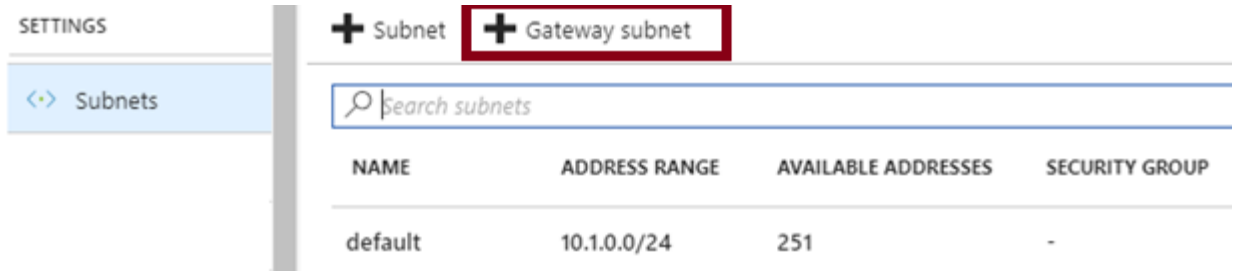
- 1. Create VNets and subnets
- 2. Specify the DNS server (optional)
- 3. Create the Gateway subnet
- 4. Create the VPN Gateway
- 5. Create the Local Network Gateway
- 6. Configure the VPN device
- 7. Create the VPN Connection

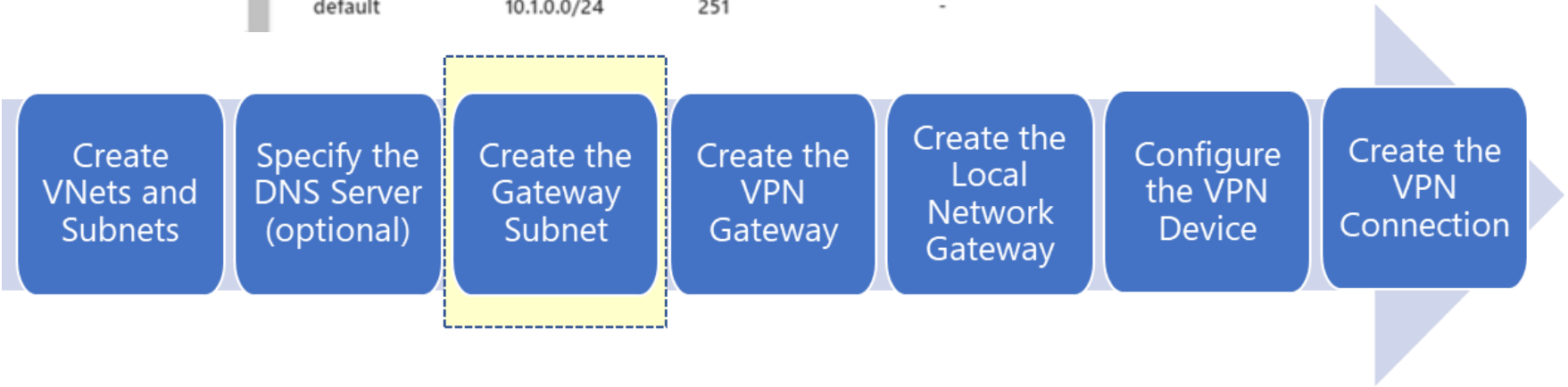




Gateway Subnet

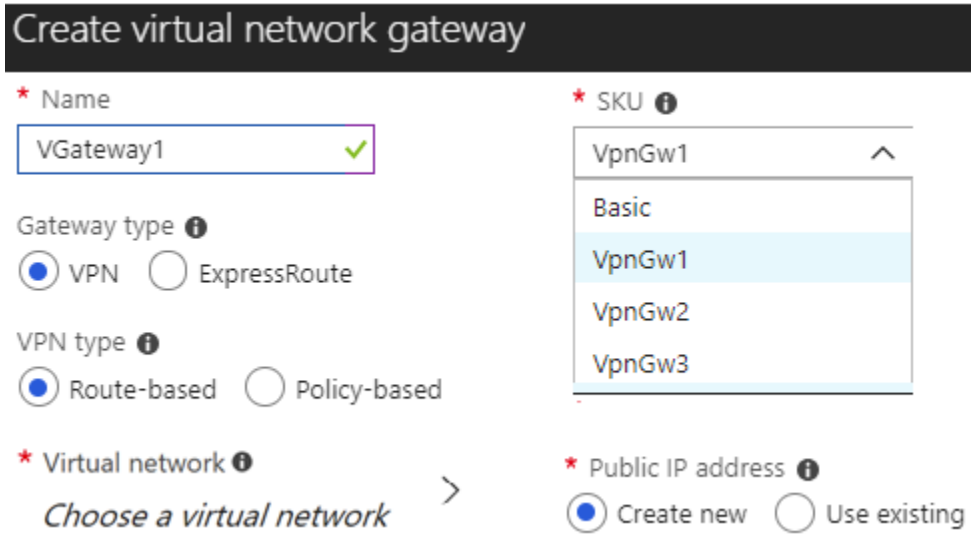
- **Requires an IP range of /28 or larger**
- **Must be named GatewaySubnet**
- **Should not be associated to any Network Security Groups**

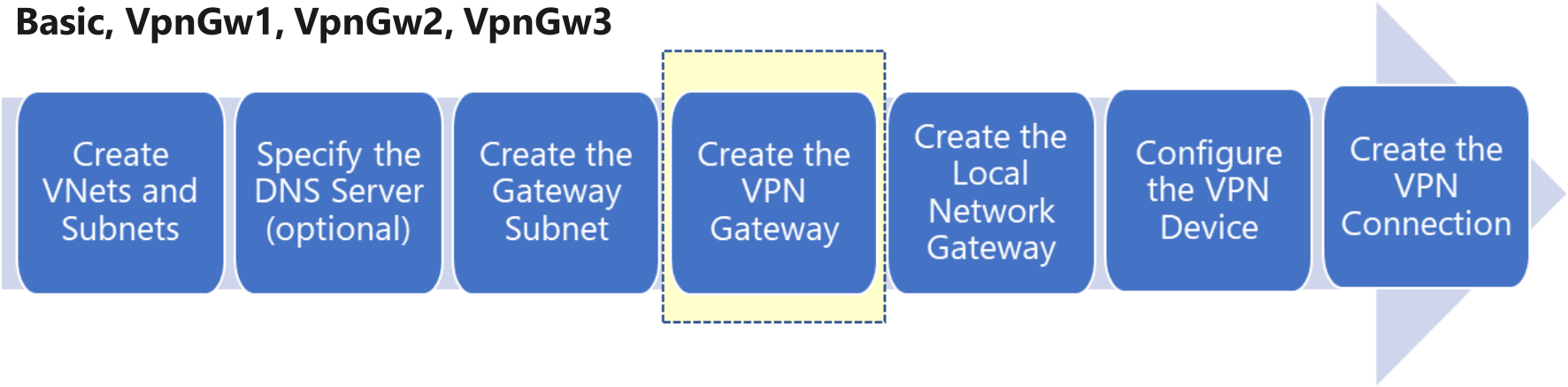




VPN Gateway

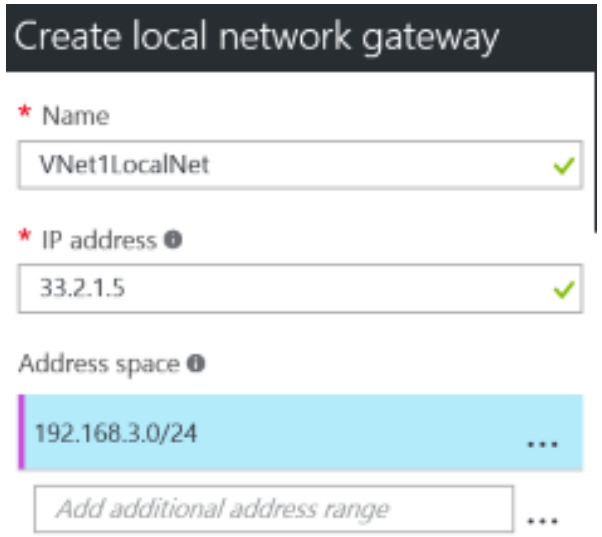
- **Establishes IPSec tunnel:**
- **Handles traffic encryption**
- **Supports multiple connections**
- **One per GatewaySubnet**
- **Supports two VPN types:**
  - **Route-based**
  - **Policy-based**
- **Supports four SKUs**
  - **Basic, VpnGw1, VpnGw2, VpnGw3**

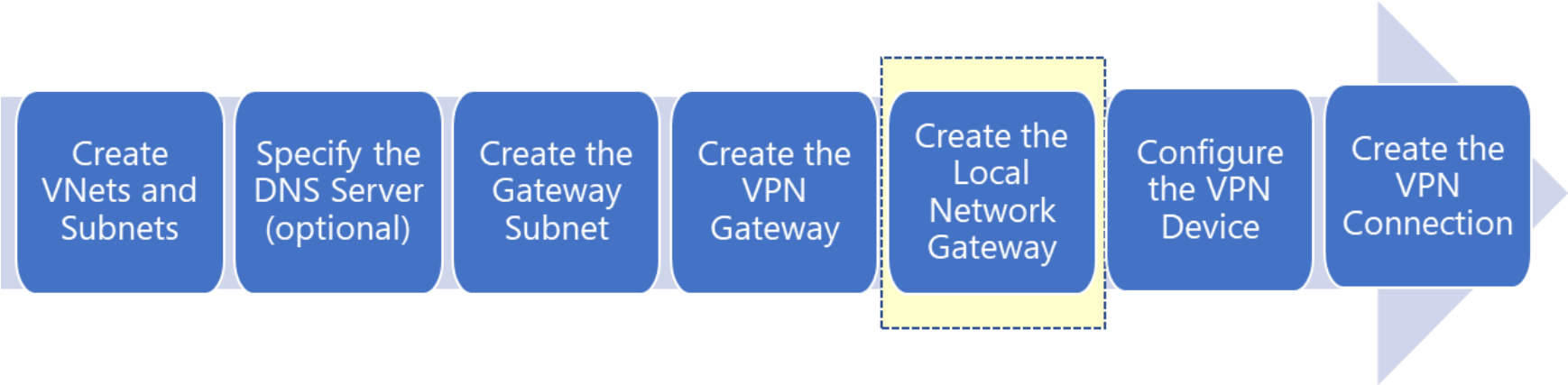




Local Network Gateway

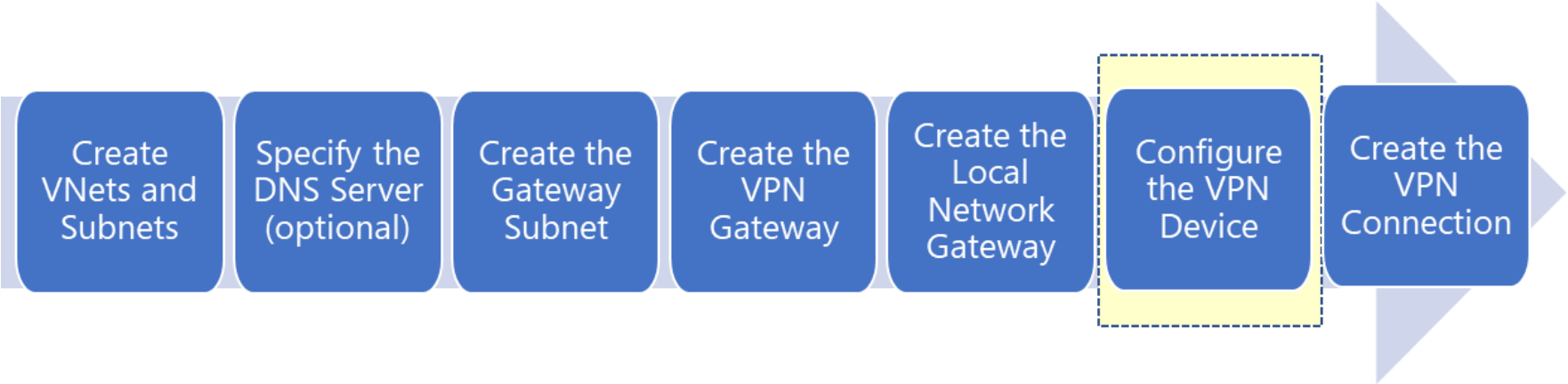
- **In hybrid scenarios, represents on-premises VPN device**
- **To configure, specify:**
  - **Name**
  - **IP address: the public IP address of the VPN device**
  - **Address space: one or more on-premises IP address ranges**





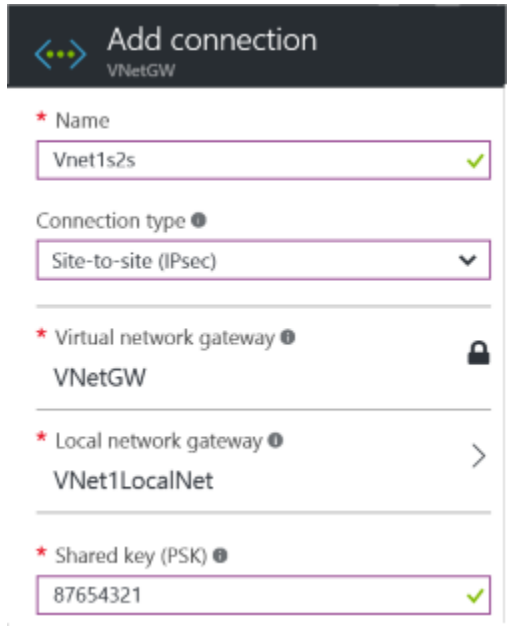
Configure the VPN Device

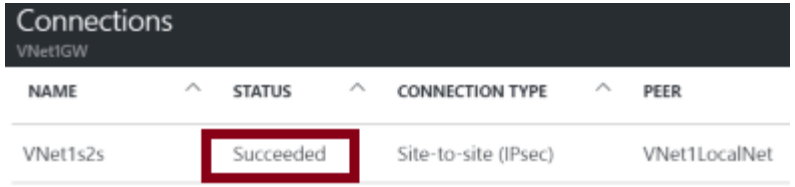
- **Follow instructions provided by Microsoft or device manufacturer:**
  - **Microsoft provides a list of the validated VPN devices**
- **The configuration settings include:**
  - **A shared key**
  - **The public IP address of the VPN gateway**

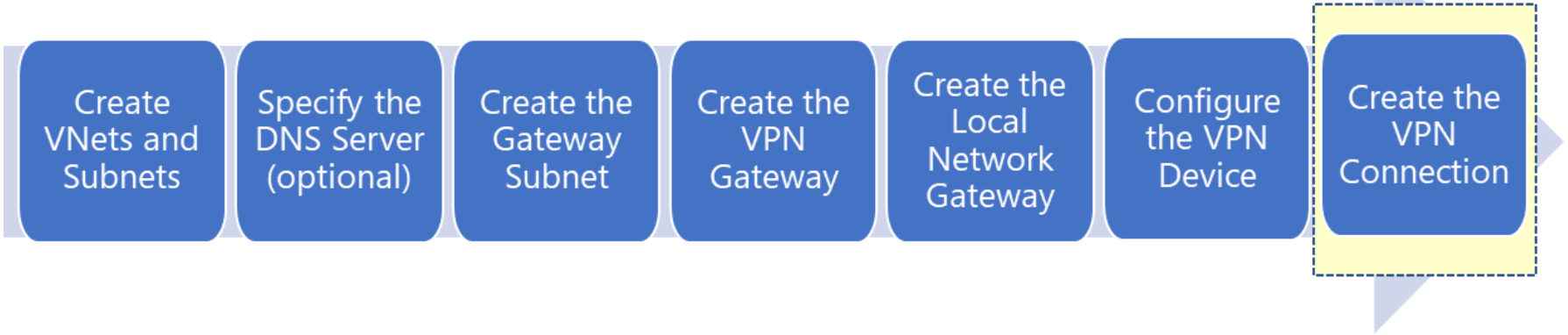


Configure the VPN Connection

- **Establishes IPSec tunnel between:**
  - **Azure VPN gateway**
  - **Local network gateway**
- **Requires:**
  - **Shared key matching the one you used in the previous step**

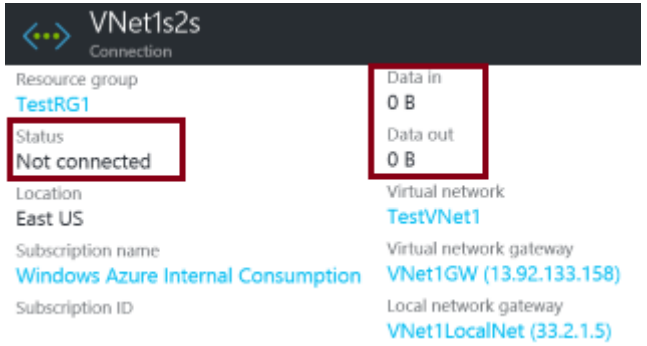






Verify the VPN Connection

- The Azure portal:
  - **Identify the connection status:**
    - **Connected**
    - **Succeeded**
  - **Note the Data in and Data out values**

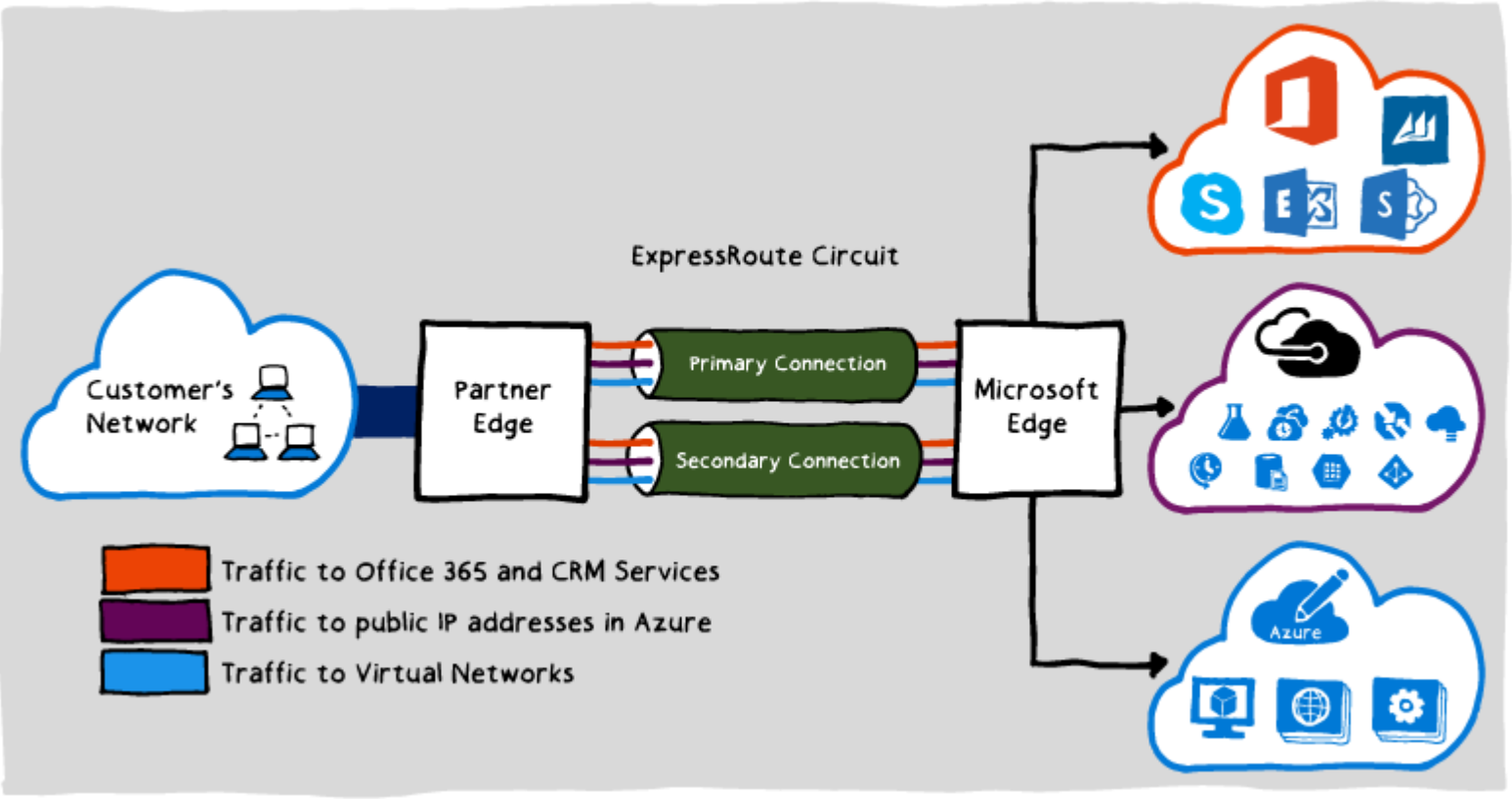


- Azure Powershell:
  - **Run the Get-AzureRmVirtualNetworkGatewayConnection cmdlet**
  - **Examine the values of:**
    - **connectionStatus**
    - **ingressBytesTransferred**
    - **egressBytesTransferred**

```
"connectionStatus": "Connected",
"ingressBytesTransferred": 33509044,
"egressBytesTransferred": 4142431
```

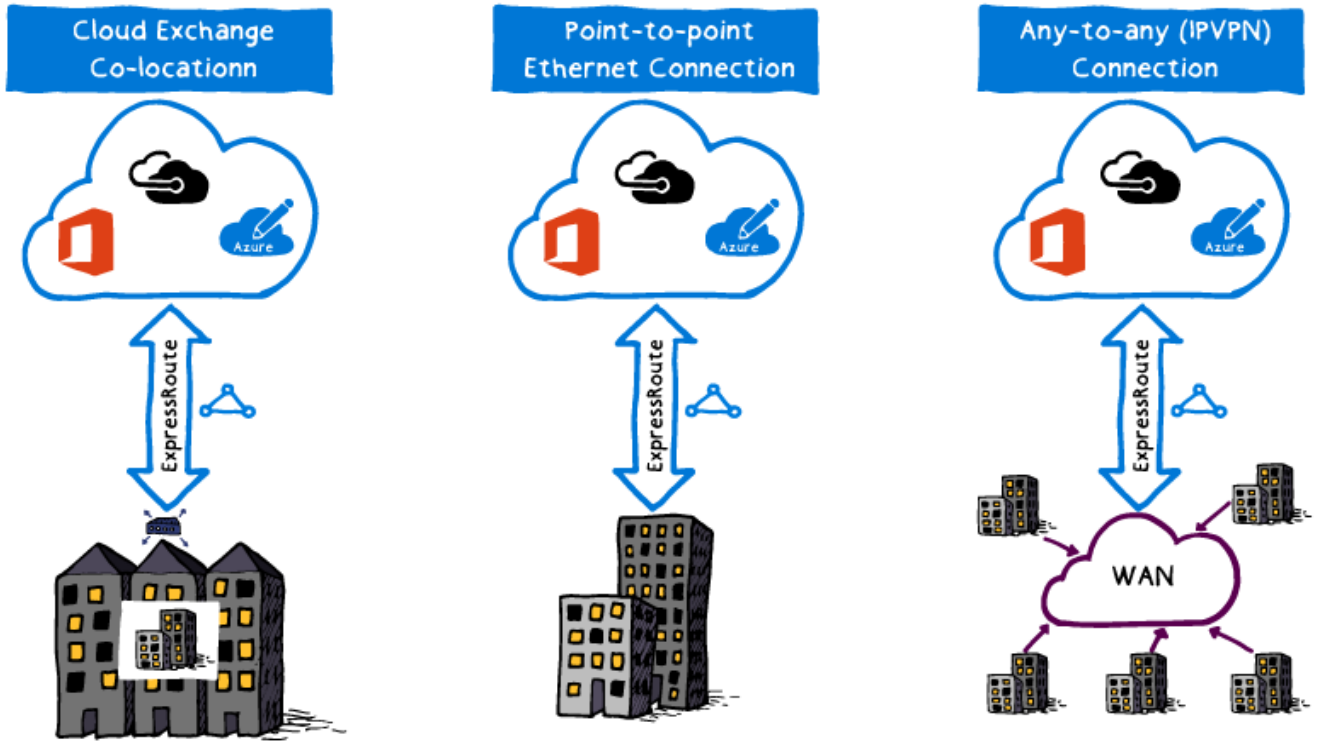
ExpressRoute

- Private connection that extends on-premises network to:
  - **Microsoft Azure**
  - **Office 365**
  - **Dynamics 365**
- Benefits:
  - **Enhanced reliability**
  - **Higher bandwidth**
  - **Lower latency**
  - **Increased security**
- Common scenarios:
  - **Data migration**
  - **Business continuity**



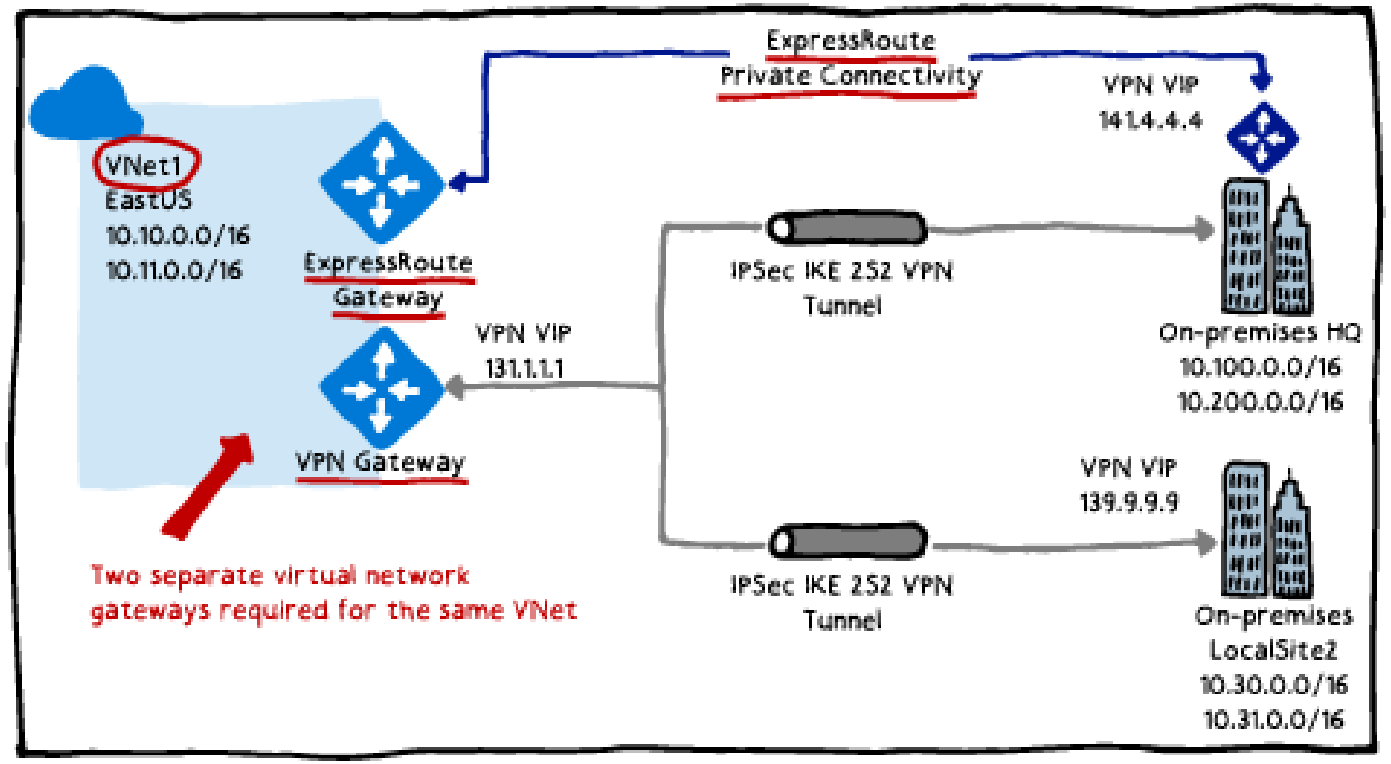
ExpressRoute Connection Options

- **Cloud Exchange Co-location**
- **Point-to-point Ethernet Connection**
- **Any-to-any (IPVPN) Connection**



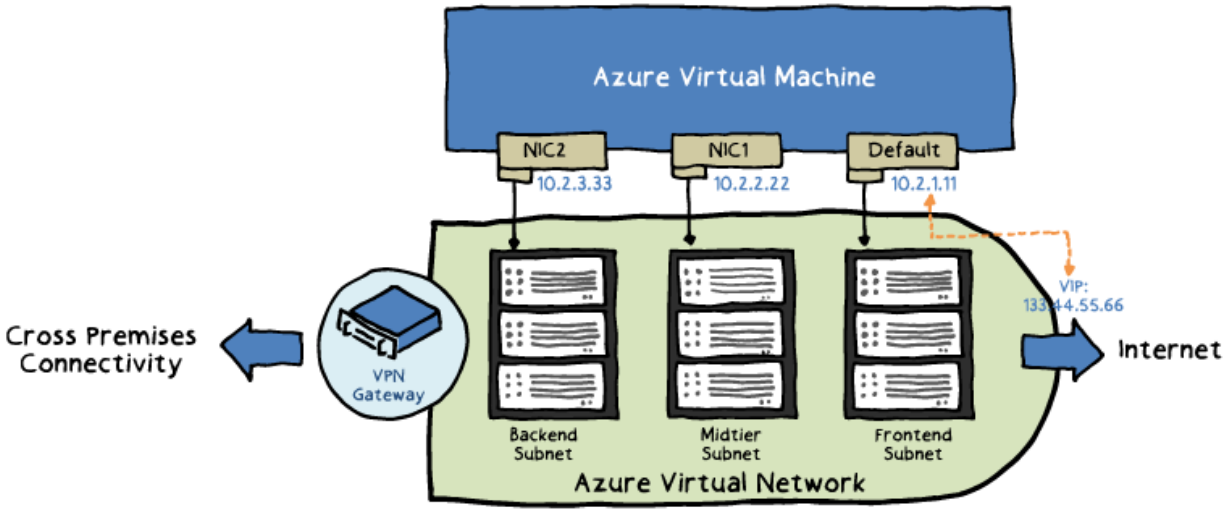
Site-to-Site and ExpressRoute Coexisting Connections

- Configuration:
  - **VPN gateway and ExpressRoute gateway are deployed to the same GatewaySubnet**
- Benefits:
  - **Extended scope of connectivity**
  - **A cost effective failover**



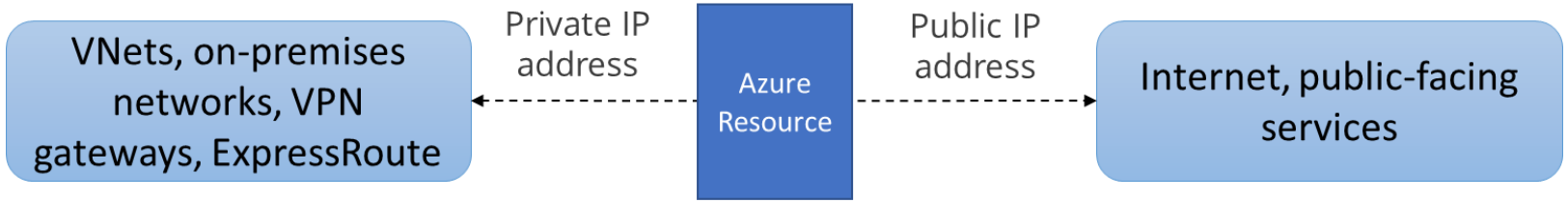
Multiple NICs in Virtual Machines

- A common requirement for network virtual appliances:
  - **Application delivery**
  - **WAN optimization**
- Common scenarios:
  - **isolation of front-end and back-end traffic**
  - **separation of data and control plane**
- Considerations:
  - **The order of NICs inside the OS of a VM is random and can change**
  - **IP addresses and MACs of NICs remain the same for the VM lifetime**
  - **The VM size determines the maximum number of NICs**



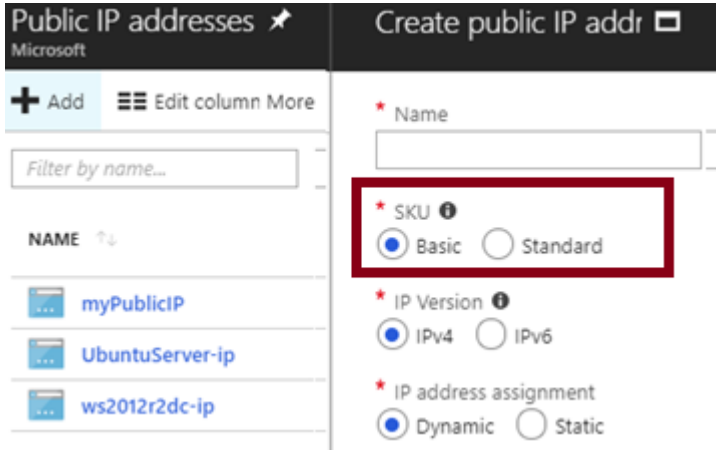
Overview of IP Addressing

- Azure networking supports two types of IP addresses:
  - **Private IP addresses: Used for communication within an Azure virtual network (VNet), and your on-premises network, when you use a VPN gateway or ExpressRoute circuit to extend your network to Azure.**
  - **Public IP addresses: Used for communication with the Internet, including Azure public-facing services.**
- Azure networking supports two types of IP address assignments:
  - **Static: common in scenarios that involve:**
    - **IP address-based security models which require apps or services to have a static IP address.**
    - **Azure VMs serving such roles as Active Directory Domain Controllers and DNS servers.**
    - **Firewall rules that allow or deny traffic using IP address ranges.**
  - **Dynamic**



Public IP Addresses

- Can be assigned to:
  - **VM NICs**
  - **Internet-facing load balancers**
  - **VPN gateways**
  - **Application gateways**
- Support two SKUs:
  - **Basic and Standard**



Public IP addresses	IP address association	Dynamic	Static
Virtual Machine	NIC	Yes	Yes
Load Balancer	Front-end configuration	Yes	Yes
VPN Gateway	Gateway IP configuration	Yes	No
Application Gateway	Front-end configuration	Yes	No

Feature	Basic SKU	Standard SKU
IP assignment	Static or dynamic	Static
Security	Open by default	Are secure by default and closed to inbound traffic
Resources	Network interfaces, VPN Gateways, Application Gateways, and Internet-facing load balancers	Network interfaces or public standard load balancers
Redundancy	Not zone redundant	Zone redundant by default

Private IP Addresses

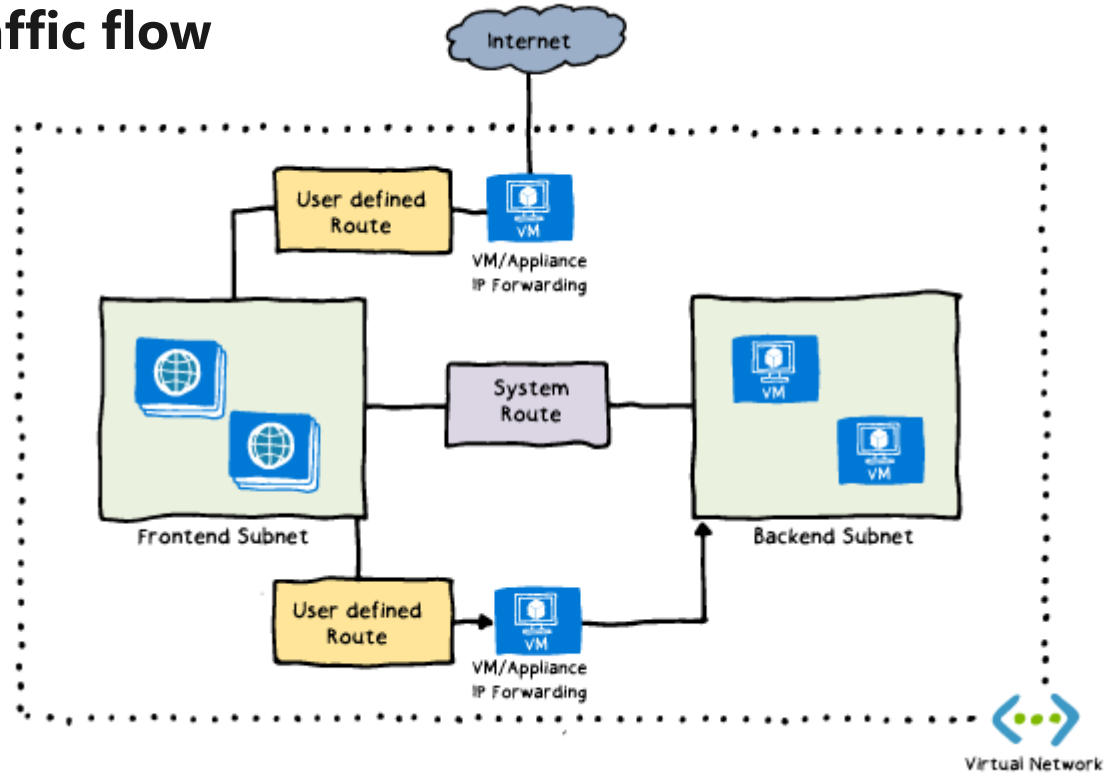
- Can be assigned to:
  - **VM NICs**
  - **Internal load balancers**
  - **Application gateways**

Private IP Addresses	IP address association	Dynamic	Static
Virtual Machine	NIC	Yes	Yes
Internal Load Balancer	Front-end configuration	Yes	Yes
Application Gateway	Front-end configuration	Yes	Yes

- Support two assignment methods:
  - **Static: You select and assign any available IP address in the subnet's address range.**
  - **Dynamic (default): Azure assigns the next available IP address in the subnet's address range.**

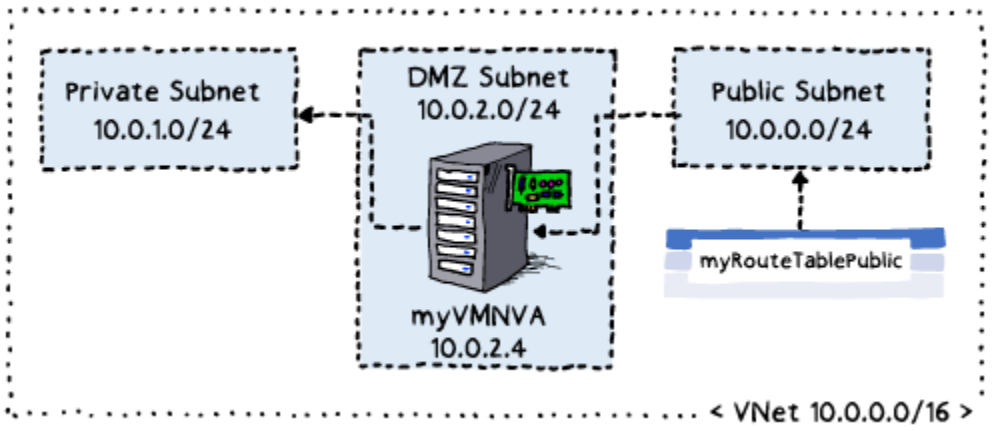
User Defined Routes

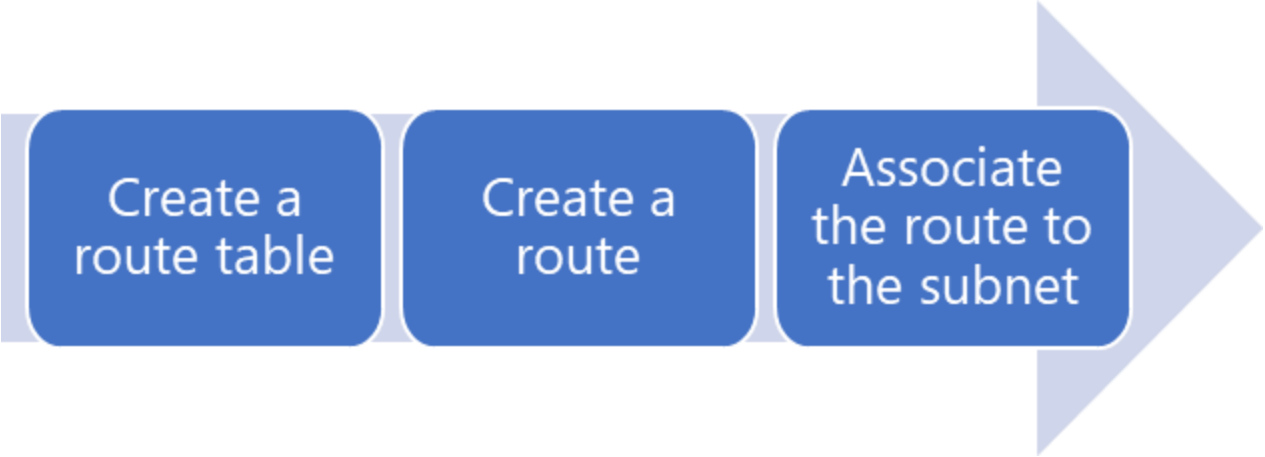
- Alter default system routes:
  - **Direct traffic to a designated virtual appliance**
  - **Facilitate routing, firewalling, or WAN optimization**
  - **Typically specify the next hop of the traffic flow**
- Implementation involves:
  - **Grouping routes into route tables**
  - **Associating route tables to subnets**



Routing Example

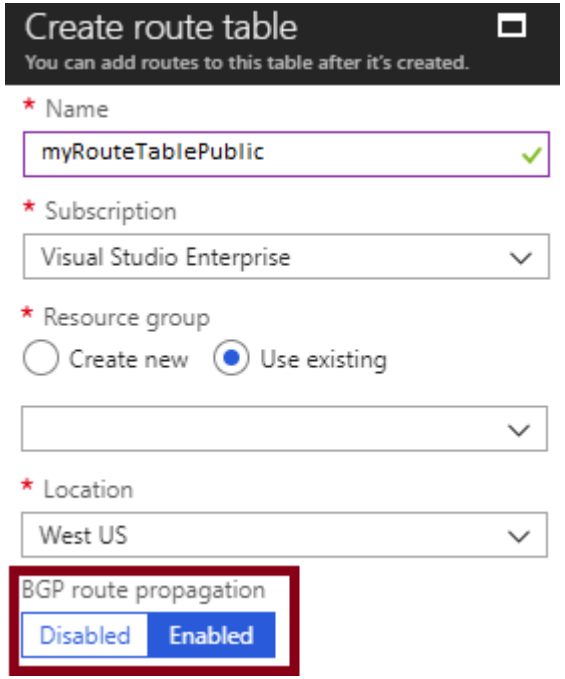
- A sample environment:
  - **Private subnet**
  - **DMZ subnet: hosting an NVA**
  - **Public subnet**
- Requirements:
  - **Traffic from the public subnet must route via NVA**
- Implementation:
  - **Create a route table**
    - **myRouteTablePublic**
  - **Add a route to the table**
    - **with NVA as the next hop**
  - **Associate the table to the Public subnet**





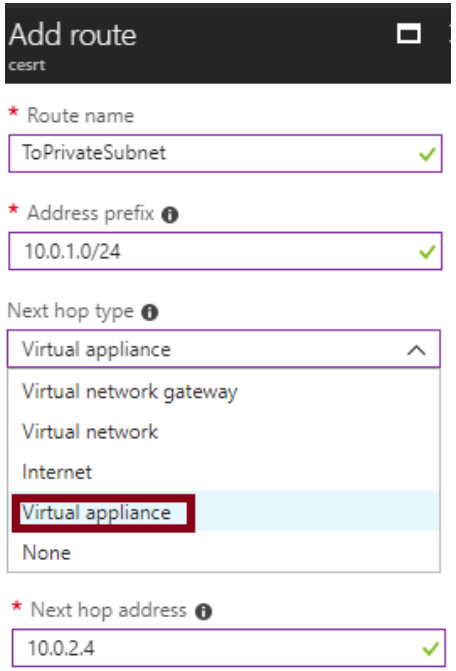
Create Route Table

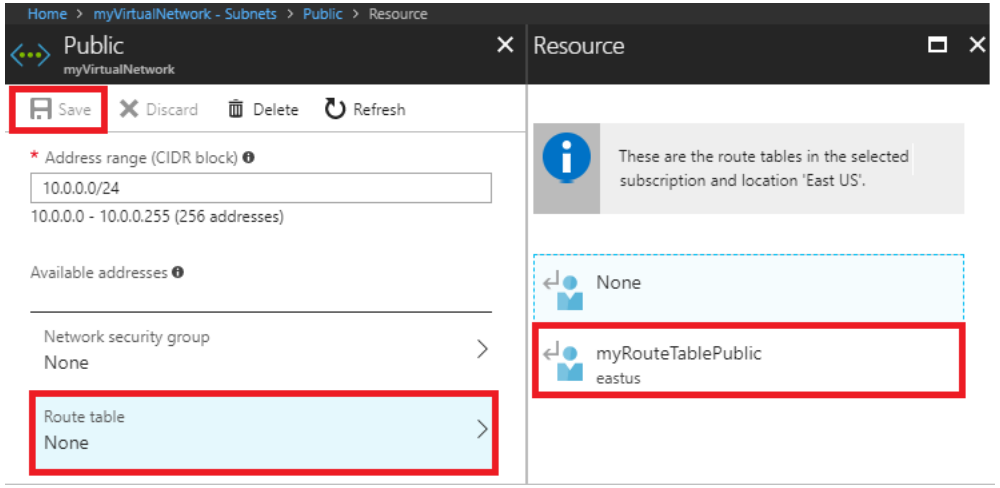
- Azure networking supports route propagation via BGP:
  - **This facilitates exchange routing and reachability information between networks**
  - **It is common in ExpressRoute scenarios**



Create and Associate the Route

- Add route to a route table:
  - **Specify the prefix**
  - **Select the next hop type:**
    - **Virtual network gateway**
    - **Virtual network**
    - **Internet**
    - **Virtual appliance**
    - **None**
  - **Specify the next hop address:**
    - **When using virtual appliance**
- Associate the route table to subnet





Routing Algorithms

- Route selection algorithms:
  - **Longest prefix match**

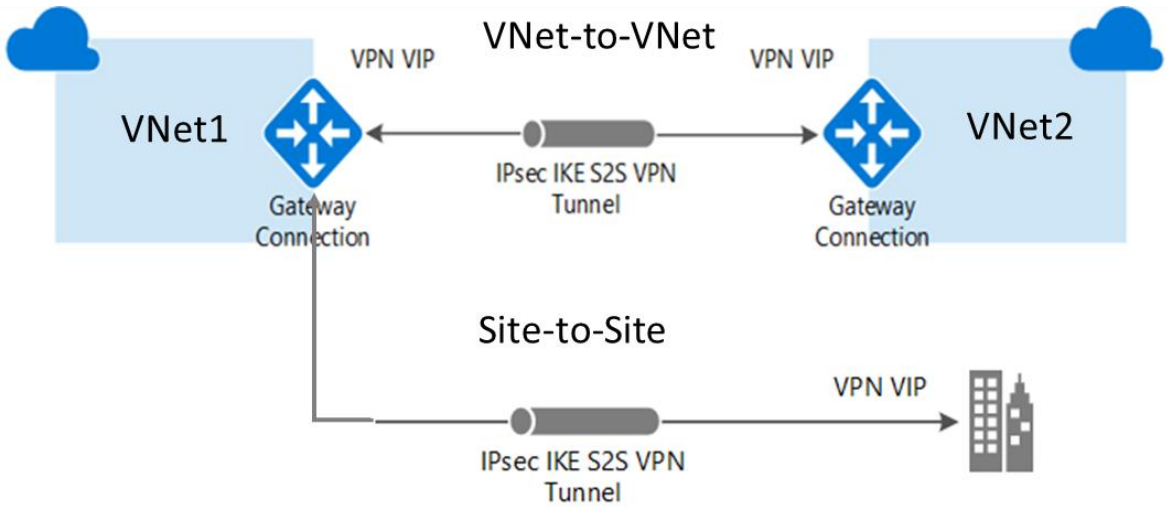
Source	Address prefixes	Next hop type
System	10.0.0.0/24	Internet (selected)
System	10.0.0.0/16	Virtual network gateway

  - **Route priorities:**
    - **1. User-defined route**
    - **2. BGP route**
    - **3. System route**

Source	Address prefixes	Next hop type
User	10.0.0.0/16	Internet (selected)
System	10.0.0.0/16	Virtual network gateway

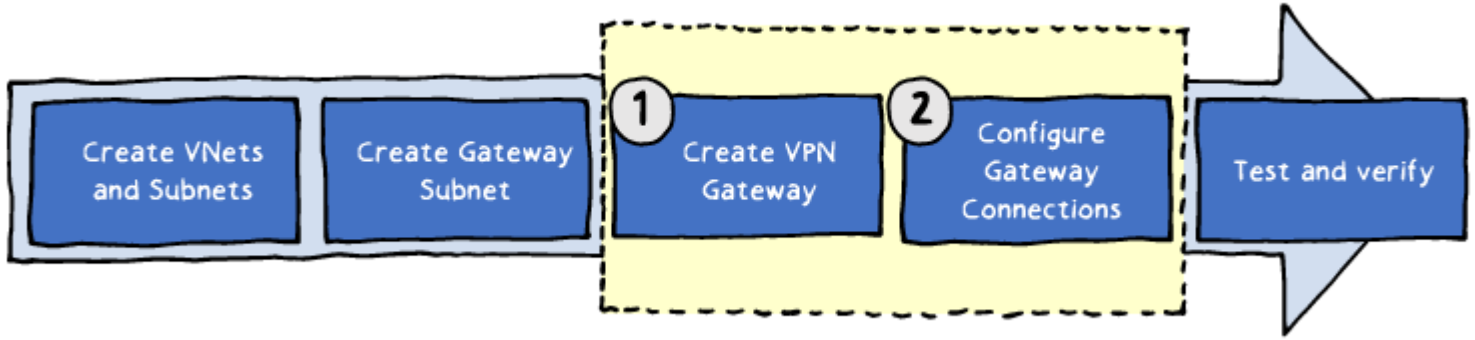
VNet-to-VNet Connections

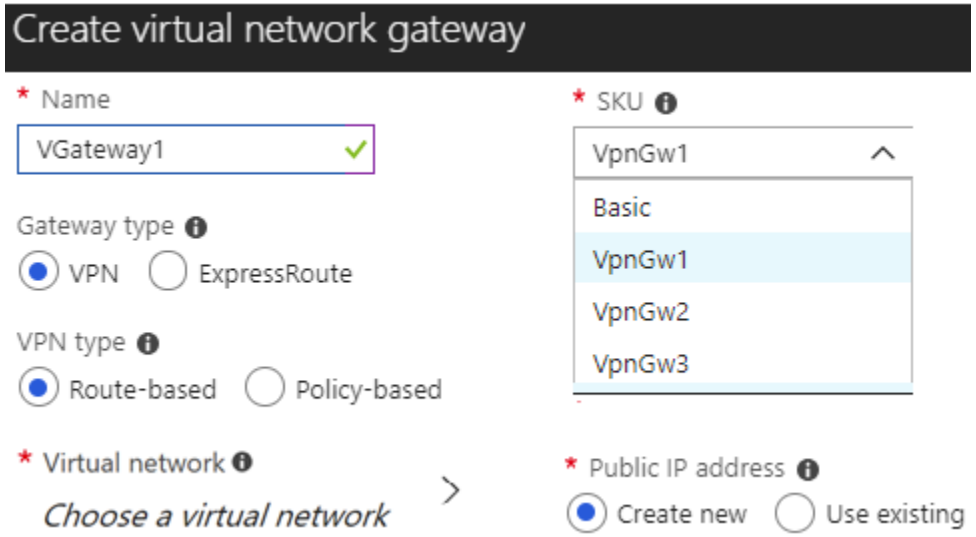
- Rely on VPN connectivity:
  - **Require VPN gateways for each VNet**
  - **Equivalent to Site-to-Site VPN in hybrid scenarios**
- Support connecting:
  - **VNets in the same or different regions.**
  - **VNets in the same or different subscriptions.**
  - **VNets and on-premises networks.**
  - **ARM VNets and classic VNets.**



Implementing VNet-to-VNet VPN

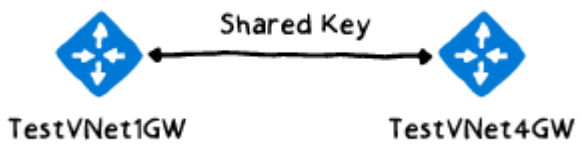
- In each VNet:
  - **Create the Gateway subnet**
  - **Create a VPN gateway:**
    - **Name and Gateway Type**
    - **VPN Type**
    - **SKU**
    - **Virtual Networks**
    - **IP Address**

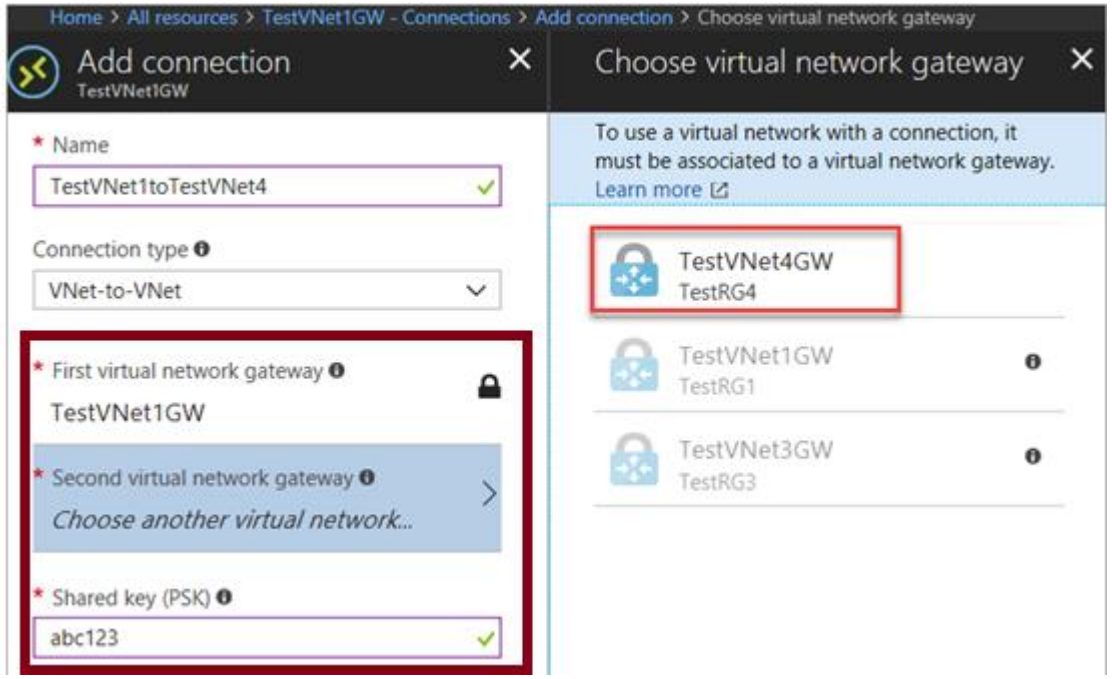




Configuring Gateway Connections

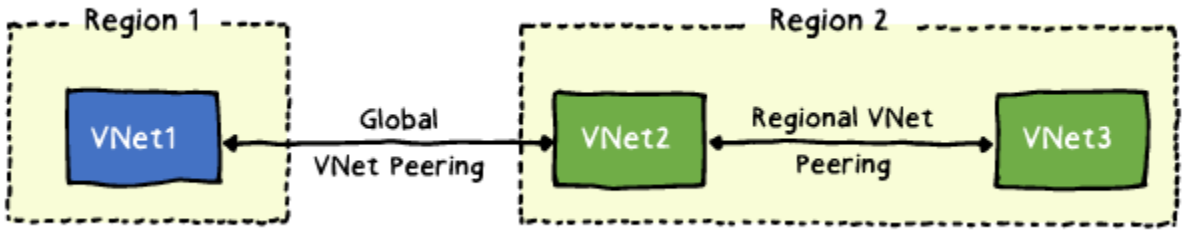
- The next step after provisioning of VPN gateways is completed
- Initial authentication is based on a shared key
- Implementation via:
  - **Azure PowerShell:**
    - **For VNets in the same or different subscriptions**
  - **The Azure portal:**
    - **For VNets in the same subscription**





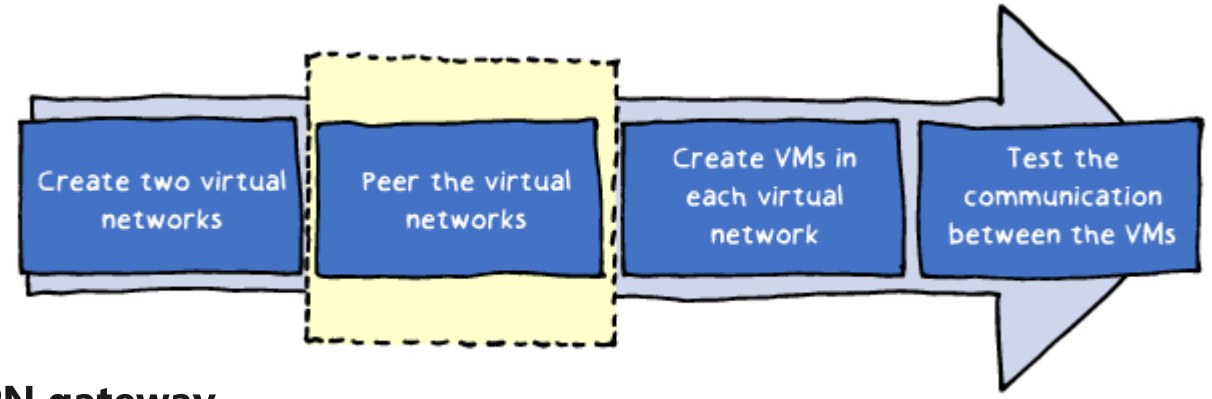
Overview of VNet Peering

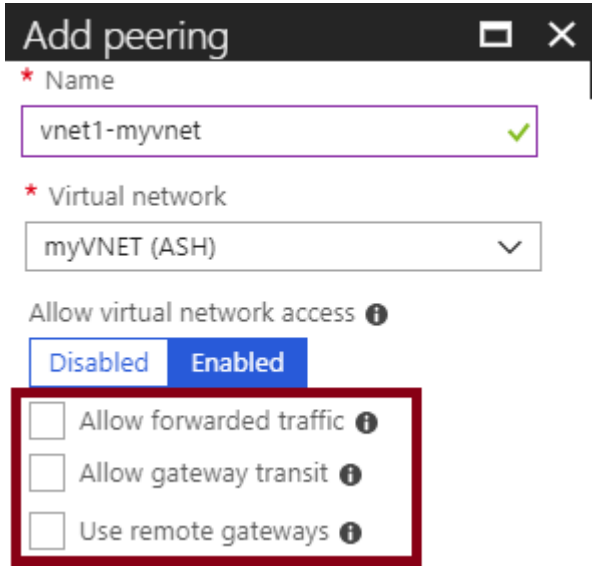
- Seamlessly connect virtual networks:
  - **Regional VNet peering: VNets in the same Azure region**
  - **Global VNet peering: VNets in different Azure regions**
- Offers a range of benefits:
  - **Security**
  - **Performance**
  - **Simplicity**
  - **Efficient**



Regional VNet Peering

- Configuration is required for each virtual network:
  - **Allow virtual network access:**
    - **Must be set to Enabled**
  - **Allow forwarded traffic:**
    - **Allows traffic routing via peered network**
  - **Allow gateway transit:**
    - **Allows the peered network to route to the local VPN gateway**
  - **Use remote gateways:**
    - **Allows the use of the VPN gateway on the peered network**



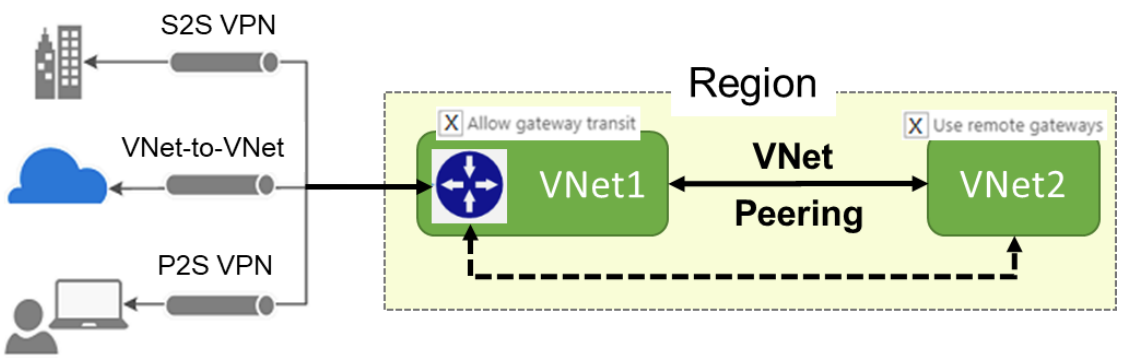


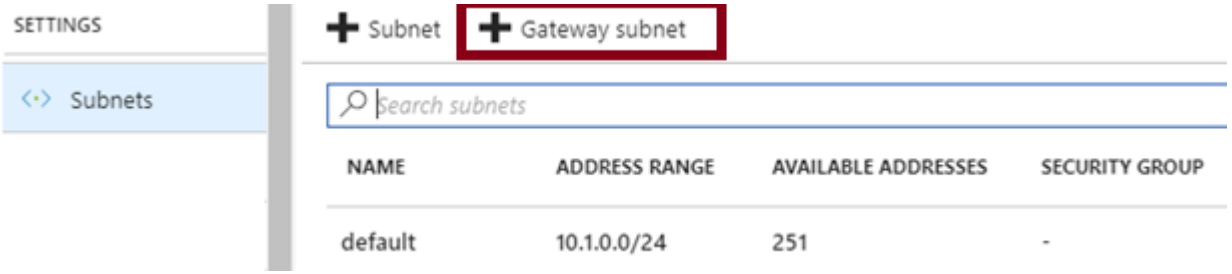
Global VNet Peering

- Provides peering across Azure regions:
  - **When implementing, peering status changes from Initiated to Connected**
- Requirements and constraints:
  - **No support for remote gateways and gateway transit**
  - **No peering between public clouds and national clouds**
  - **No access to front-end IP addresses of Azure internal load balancers in peered networks**
  - **No support for transitivity (for both regional and global peering)**

Gateway Transit

- Provides routing by using a VPN Gateway in a peered VNet to:
  - **an on-premises network (via site-to-site VPN)**
  - **another virtual network (via VNet-to-Vnet)**
  - **a remote VPN client (via point-to-site VPN)**
- Facilitates hub-spoke topology
- Requires Gateway subnet:
  - **hosting the Gateway only (no VMs)**



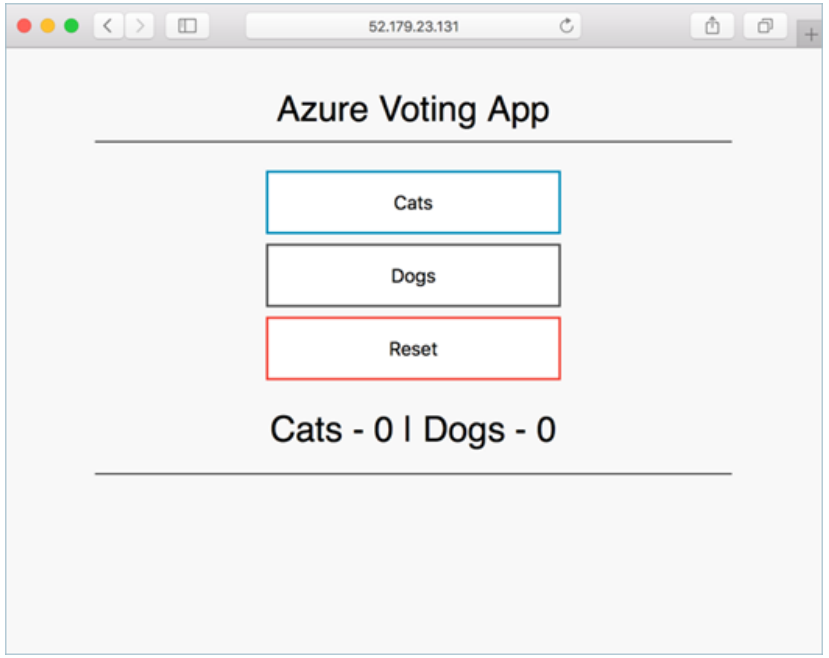


About Azure Kubernetes Service

- Managed Kubernetes cluster hosted in Azure with a range of benefits:
  - **Flexible deployment options**
  - **Identity and security management**
  - **Integrated logging and monitoring**
  - **Cluster node scaling**
  - **Cluster node upgrades**
  - **HTTP application routing**
  - **GPU enabled nodes**
  - **Development tooling integration**
  - **Virtual network integration**
  - **Private container registry**

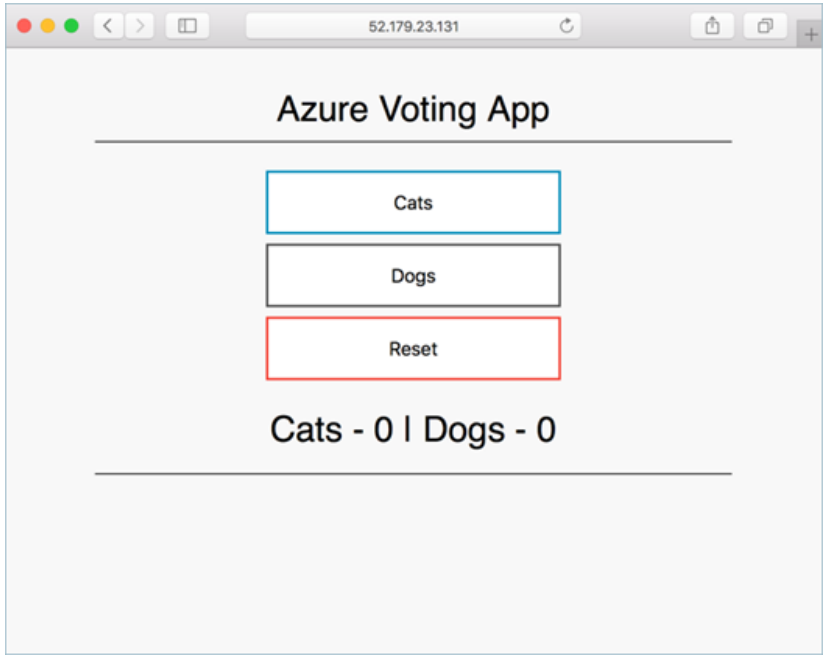
Deploy an AKS cluster using Azure CLI

- Implementation steps:
  - **1. Create a resource group:**
    - **az group create --name myAKSCluster --location eastus**
  - **2. Create an AKS cluster:**
    - **az aks create --resource-group myAKSCluster --name myAKSCluster --node-count 1 --enable-addons monitoring --generate-ssh-keys**
  - **3. Connect to the AKS cluster:**
    - **az aks get-credentials --resource-group myAKSCluster \ --name myAKSCluster**
  - **4. Create a Kubernetes manifest file (azure-vote.yaml)**
  - **5. Run the containerized application:**
    - **kubectl apply -f azure-vote.yaml**
  - **6. Monitor the progress of the deployment:**
    - **kubectl get service azure-vote-front –watch**
  - **7. Monitor health and logs (from the Azure portal)**



Deploy an AKS cluster using Azure Portal

- Implementation steps:
  - **1. Create a resource > Kubernetes Service:**
    - **Basics: Project details, Cluster details, Scale**
    - **Authentication: A new or existing service principal and RBAC settings**
    - **Networking: Http application routing and Network configuration (Basic or Advanced)**
    - **Monitoring: A new or existing Log Analytics workspace.**
  - **2. Connect to the cluster:**
    - **az aks get-credentials --resource-group myAKSCluster **
      --name myAKSCluster
  - **3. Create a Kubernetes manifest file (azure-vote.yaml)**
  - **4. Run the containerized application:**
    - **kubectl apply -f azure-vote.yaml**
  - **5. Monitor health and logs (from the Azure portal)**



Azure Container Registry overview

- Key concepts:
  - **Registry**
  - **Repository**
  - **Image**
  - **Container**
- ACR is a managed container registry based on Docker Registry 2.0:
  - **Integrates with Azure Container Registry Build (for building container images)**
  - **Is available in three service tiers: Basic, Standard, and Premium**
  - **Offers:**
    - **Webhook integration**
    - **Azure AD authentication**
    - **Delete functionality**
    - **Geo-replication (with the Premium tier)**

```
docker push <acrLoginServer>/aci-helloworld:v1
```

Deploy an image to ACR using Azure CLI

- Implementation steps:
  - **1. Create a resource group:**
    - **az group create --name myResourceGroup --location eastus**
  - **2. Create a container registry:**
    - **az acr create --resource-group myResourceGroup --name myContainerRegistry007 --sku Basic**
  - **3. Log in to ACR:**
    - **az acr login --name <acrName>**
  - **4. Push a new image to ACR (or pull and tag an existing image first):**
    - **docker pull microsoft/aci-helloworld**
    - **docker tag microsoft/aci-helloworld <acrLoginServer>/aci-helloworld:v1**
    - **docker push <acrLoginServer>/aci-helloworld:v1**
  - **5. Deploy a container by using an image in ACR:**
    - **az container create --resource-group myResourceGroup --name acr-quickstart --image <acrLoginServer>/aci-helloworld:v1 --cpu 1 --memory 1 --registry-username <acrName> --registry-password <acrPassword> --dns-name-label aci-demo --ports 80**

Azure Container Instances Overview

- A managed container hosting option offering a range of benefits:
  - **Fast startup times**
  - **Public IP connectivity and DNS name**
  - **Hypervisor-level security**
  - **Custom sizes**
  - **Persistent storage**
  - **Linux and Windows containers**
  - **Co-scheduled groups**

Implement an application using Virtual Kubelet

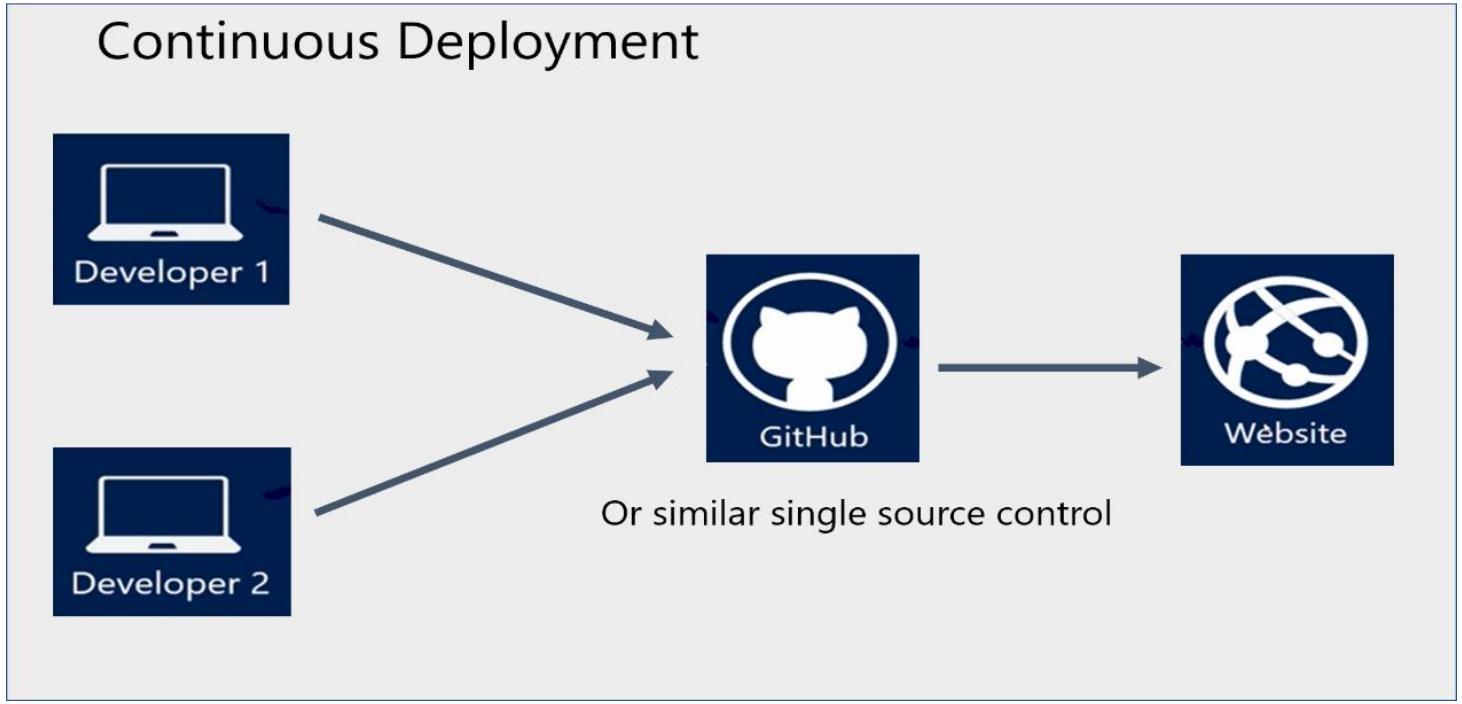
- Virtual Kubelet provider is an experimental open source project:
  - **Allows scheduling AKS-managed containers on Azure Container Instances**
  - **Combines functional benefits of AKS with low pricing of ACI**
  - **Supports both Windows and Linux containers**
- To install the Virtual Kubelet provider:
  - **Ensure that the prerequisites are satisfied:**
    - **An existing AKS cluster**
    - **Azure CLI version 2.0.33 and Helm installed**
    - **A service account and role binding for use with Tiller (for RBAC-enabled AKS clusters)**
  - **Run the az aks install-connector command:**
    - **az aks install-connector --resource-group myAKSCluster --name myAKSCluster --connector-name virtual-kubelet --os-type Both**
  - **Once the installation completes, you can deploy Windows and Linux containers to ACI:**
    - **In the .yaml file, replace kubernetes.io/hostname with the name of the Linux/Windows Virtual Kubelet node**

Web App Features

- Service included in the App Service offering, which provides:
  - **Multiple languages and frameworks**
  - **DevOps optimization**
  - **Global scale with high availability**
  - **Connections to SaaS platforms and on-premises data**
  - **Security and compliance**
  - **Application templates**
  - **Visual Studio integration**
  - **API and mobile features**
  - **Serverless code**

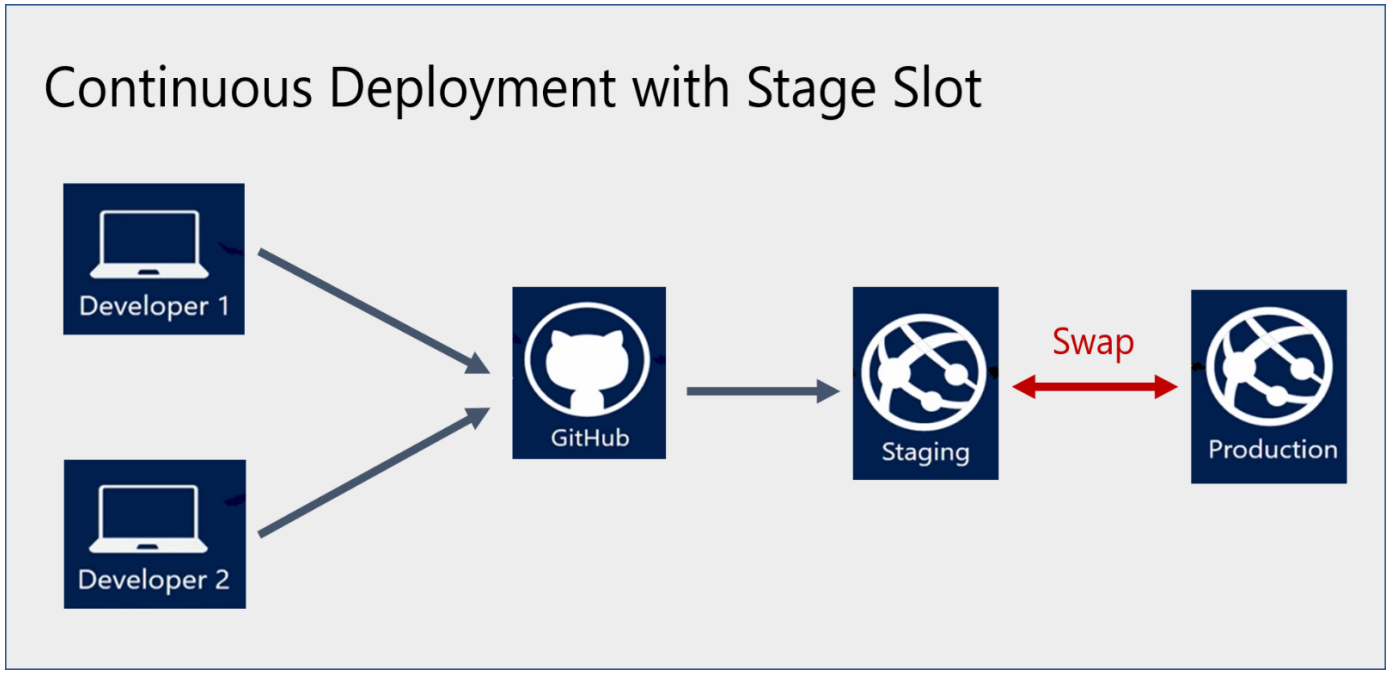
What is Continuous Development?

- Automatic publishing of updates to source control code:
  - **Visual Studio Team Services**
  - **Bitbucket**
  - **GitHub**
  - **Git**
  - **and many others…**
  -



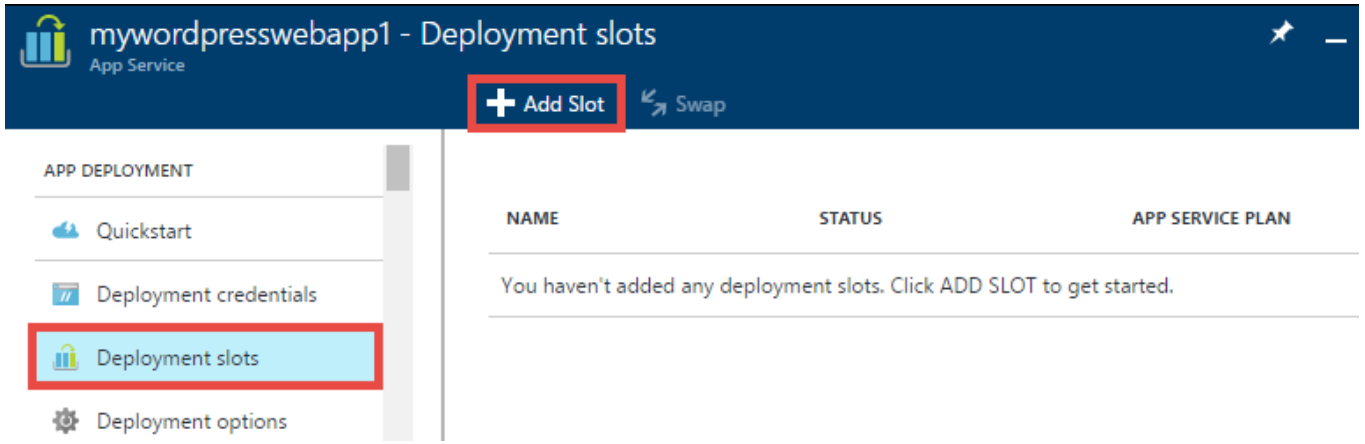
Staging Environments in App Service

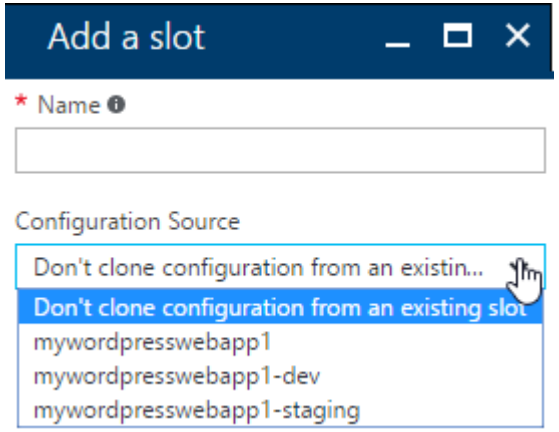
- A separate deployment slot of App Service services:
  - **Supported by web apps, mobile apps, and API apps**
  - **Requires a higher pricing tier:**
    - **Standard**
    - **Premium**
    - **Isolated**
  - **Offers a number of benefits:**
    - **Additional validation of app changes**
    - **Slot warming before swap**
    - **Support for auto-swap**
    - **Immediate rollback**



Add a Deployment Slot

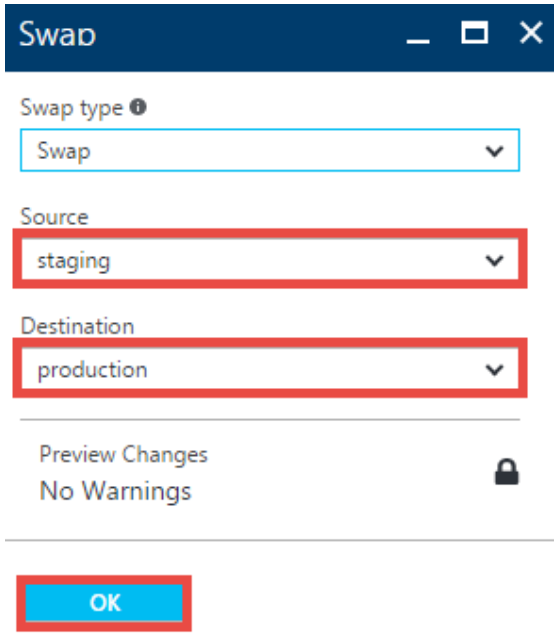
- The operation can be performed directly from the Azure portal
- Slot cloning:
  - **For the first slot:**
    - **you can clone the production slot**
  - **For subsequent slots:**
    - **you can clone any existing slots**





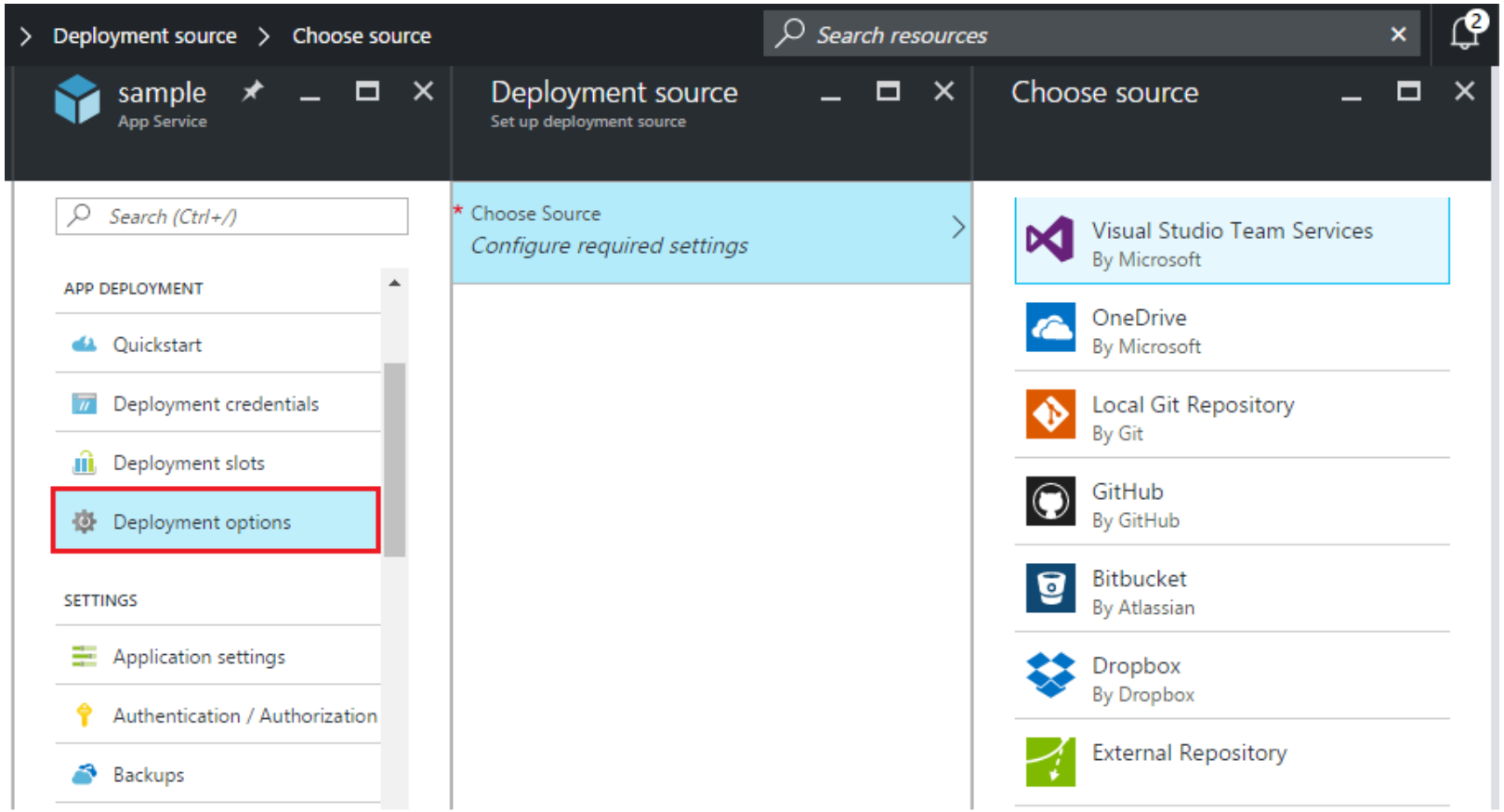
Swap Deployment Slots

- There are three types of slot swapping settings:
  - **Settings that are not swapped:**
    - **publishing endpoints, custom domain names, SSL certificates and bindings, scale settings, WebJob schedulers.**
  - **Settings that are swapped:**
    - **general settings, handler mappings, monitoring, diagnostics, and WebJobs.**
  - **Settings that are configurable (either swapped or not swapped):**
    - **App settings and connection strings.**



Enabling Continuous Deployment

- Configurable from the Azure portal:
  - **VSTS**
  - **OneDrive**
  - **Local Git**
  - **GitHub**
  - **Bitbucket**
  - **Dropbox**
  - **External Repository**

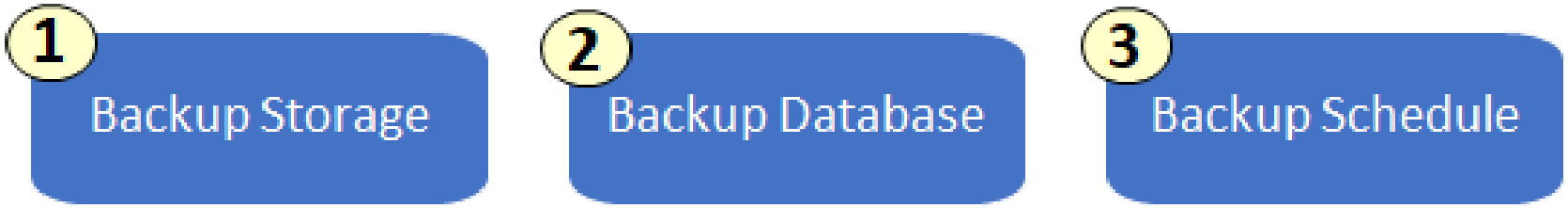


Web App Templates

- Azure Resource Manager templates for:
  - **Deploying a Windows-based web app:**
  - **Configuring a Windows-based web app:**
  - **Deploying and configuring a Linux-based web app**
  - **Deploying web apps with connected resources (e.g. an Azure SQL Database)**
  - **Creating App Service Environment v2**

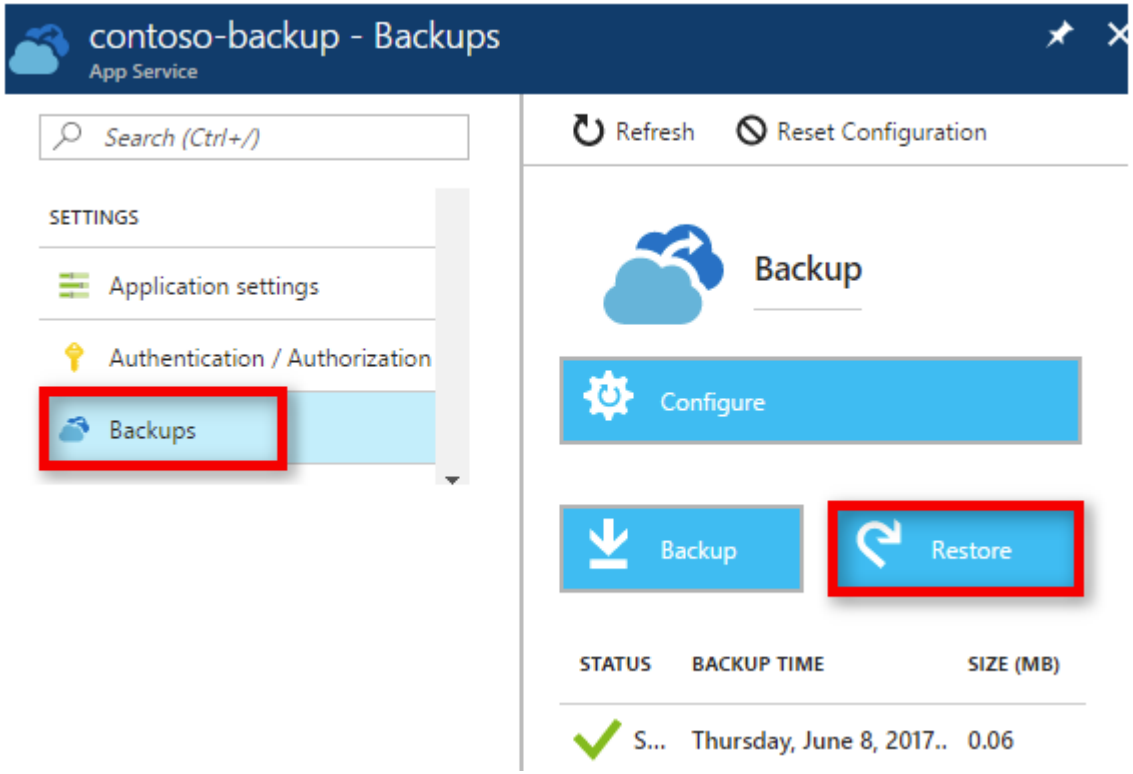
Backup Your App

- To configure Azure web app backups, provide:
  - **1. Backup storage which designates backup destination:**
    - **Azure Storage account and container**
  - **2. Backup database (in scenarios where the web app uses a back-end database):**
    - **The database must be referenced in the connection strings of the web app**
    - **The backup supports Azure SQL Database, Azure Database for MySQL, Azure Database for PostgreSQL, and MySQL in-app**
  - **3. Backup schedule (in order to automate backups)**



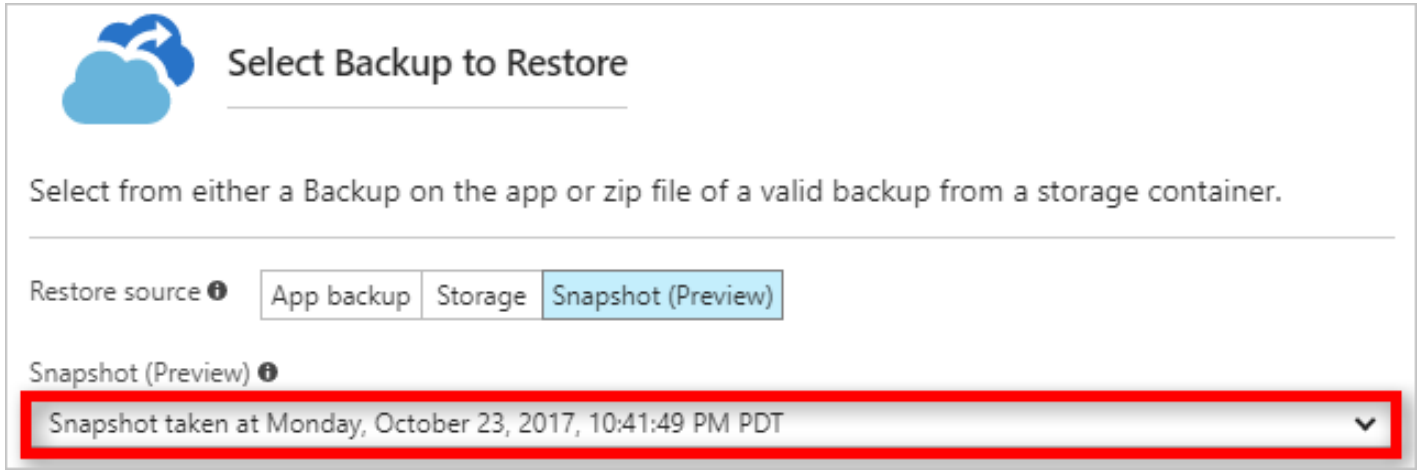
Restore a Backup

- Available from the Azure portal
- Included in higher pricing tiers:
  - **Standard**
  - **Premium**
- Requires designating:
  - **Restore source:**
    - **Any backup of the current web app**
    - **Any web app backup in Azure Storage**
  - **Restore destination:**
    - **An existing web app**
    - **A new web app**



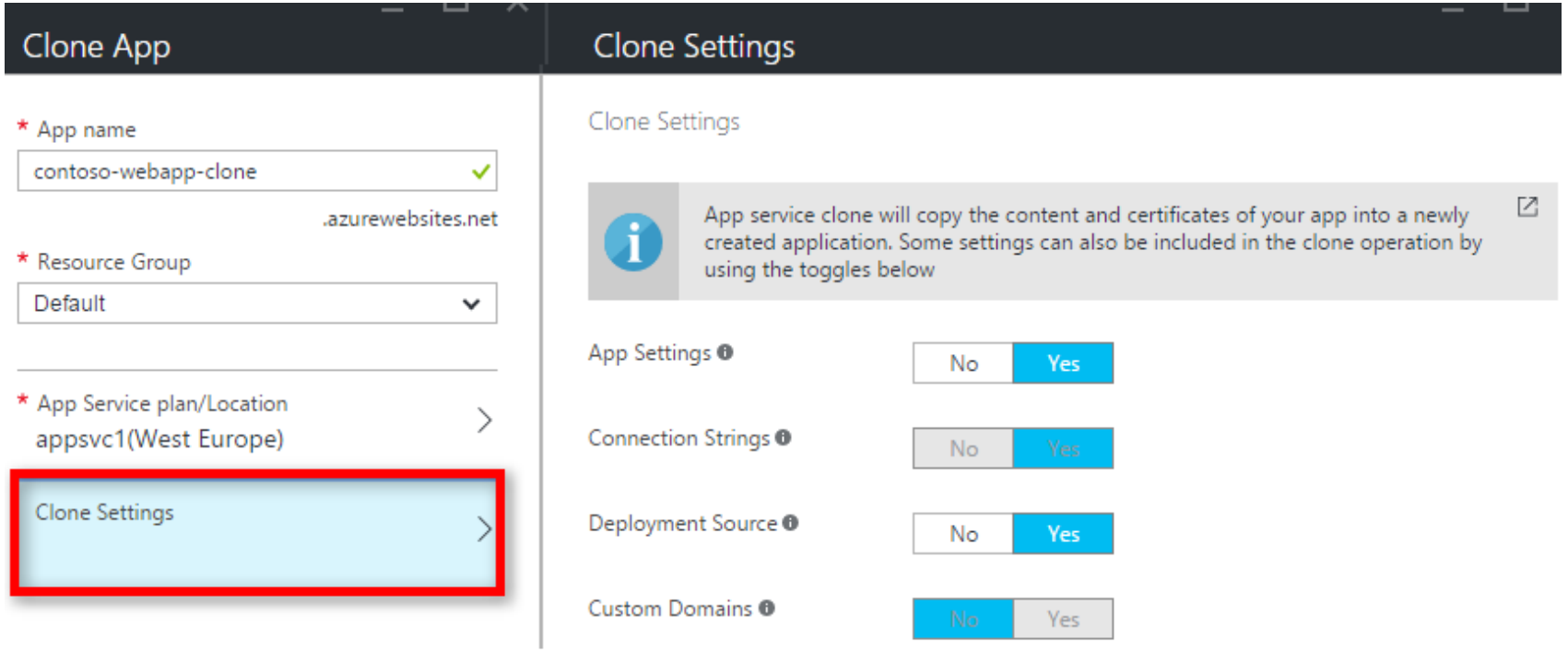
Restore a Snapshot

- Automatic, incremental shadow copies of a web app:
  - **Requires the Premium pricing tier**
  - **Is a subject to a number of constraints:**
    - **You can only restore to the same app or to a slot belonging to that app.**
    - **App Service stops the target app or target slot during the restore.**
    - **App Service keeps three months' worth of snapshots for platform data recovery purposes.**
    - **You can only restore snapshots from the last 30 days.**
  - **Offers several benefits:**
    - **No file copy errors due to file locks.**
    - **No storage size limitation.**
    - **No configuration required**



Cloning an App

- Creates a replica of a web app in the same or a different region:
  - **Requires the Premium pricing tier**
  - **Allows selecting settings to clone, including:**
    - **Connection strings**
    - **Deployment source**
    - **Custom domains**
  - **Does not include:**
    - **Auto scale settings**
    - **Backup schedule**
    - **VNET settings**
    - **Auth settings**
    - **Kudu Extensions**
    - **TiP rules**
    - **Database content**
    - **App Insights**



App Service Security Levels

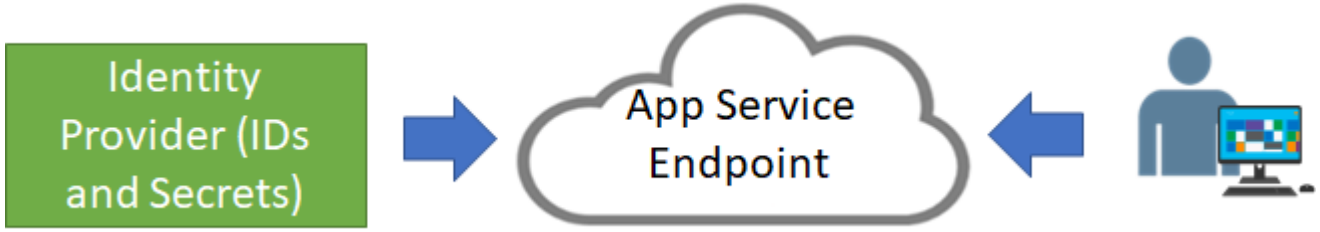
- Infrastructure and platform security:
  - **Controlled by the Azure platform**
  - **Includes compute, storage, network, web frameworks, integration features, etc.**
  - **Involves continuous security and compliance checks:**
    - **App Service apps are isolated from both the Internet and from the other customers' Azure resources.**
    - **Communication of secrets between App Service apps and other Azure resources does not cross network boundaries and is always encrypted.**
    - **Communication between App Service apps and external resources, such as PowerShell, Azure CLI, Azure SDKs, REST APIs, and hybrid connections, is properly encrypted.**
    - **24-hour threat management protects App Service resources from malware, distributed denial-of-service (DDoS), man-in-the-middle (MITM), and other threats.**
- Application security:
  - **Controlled by customers**
  - **Includes Azure AD integration, certificate management, secure communication, etc.**

Application Security

- Customers are responsible for:
  - **Developing, deploying, and managing App Service code and content in a secure way**
  - **Protecting apps from cyber threats such as:**
    - **SQL Injection**
    - **Session hijacking**
    - **Cross-site-scripting**
    - **Application level Man-In-the-Middle (MITM) attacks**

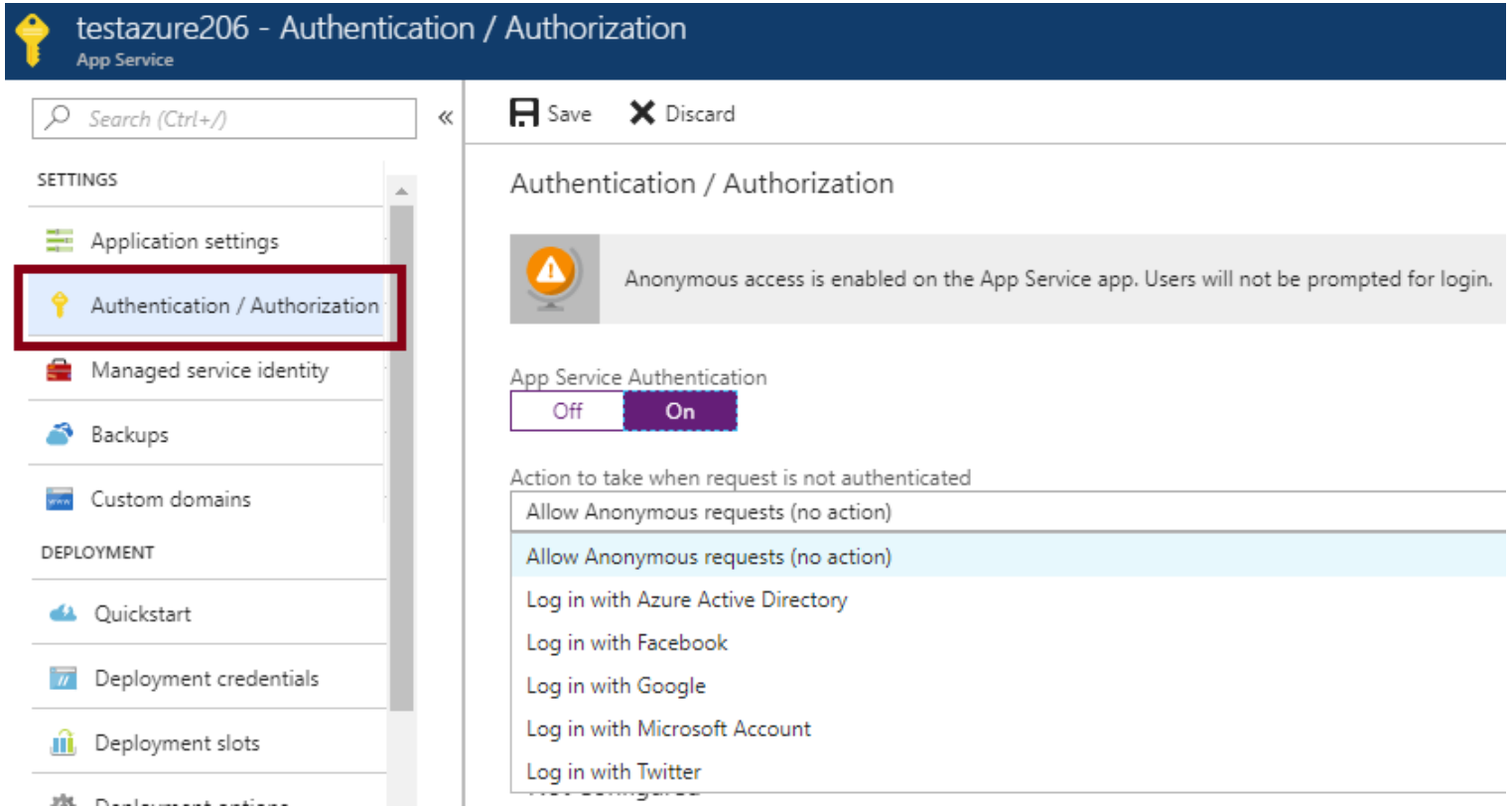
App Service Authentication

- Relies on a trust relationship with an identity provider (IdP):
  - **You select an identity provider for your App Service app**
  - **The identity provider handles authentication and generates authentication token**
  - **The App Service app validates token assertions and authorizes access to its resources**
  - **Cookies, JWTs, and bearer tokens validate access for the duration of a session**
- Directs authentication requests to an authentication endpoint
- Includes built-in supports for the most popular identity providers



Authentication Providers

- Configurable directly from the Azure portal:
  - **Relies on the Authentication/Authorization feature**
  - **Uses federated identity**
  - **Offers built-in support for:**
    - **Azure AD**
    - **Facebook**
    - **Google**
    - **Microsoft Account**
    - **Twitter**

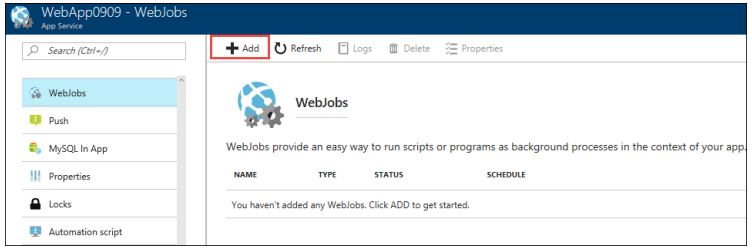


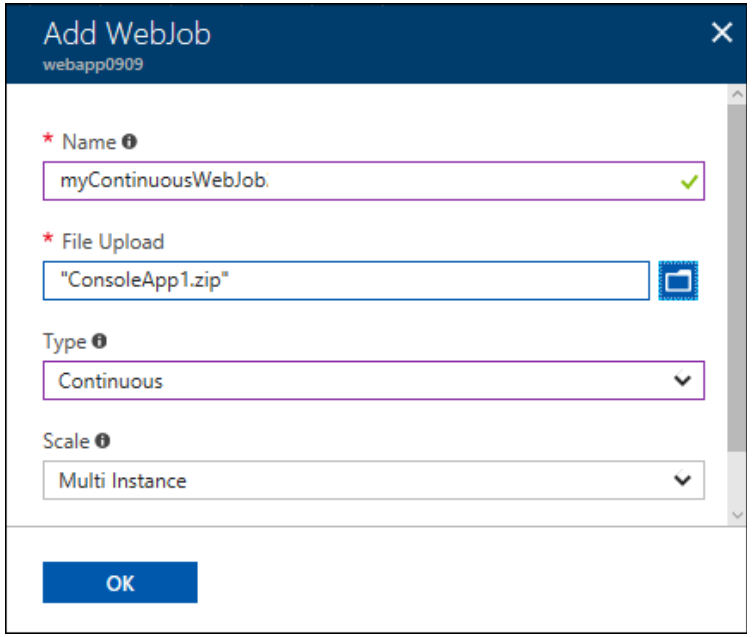
Overview of WebJobs

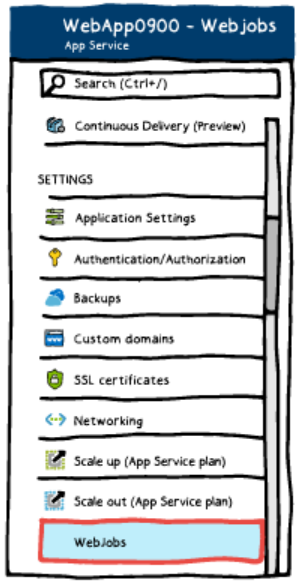
- A feature of Azure App Service for execution of background tasks:
  - **Includes support for executables or scripts (.cmd, .bat, .exe, .ps1, .sh, .php, .py, .js, .jar)**
  - **Tasks execute in the same context as the App Service app**
  - **Azure WebJobs SDK provides programmability**
  - **Can run:**
    - **Continuously:**
      - **Start immediately when the WebJob is created and run in an endless loop**
      - **Run on all instances of the web app (you can restrict it to individual instances)**
      - **Support remote debugging**
    - **On trigger:**
      - **Start on-demand or on-schedule**
      - **Run on a single instance selected by the Azure platform**
      - **Do not support remote debugging**
  - **Times out after 20 minutes of inactivity, unless you enable the Always On feature;**
    - **Always On requires Basic, Standard, or Premium pricing tier**

Creating a continuous Webjob

- Implementation steps (the Azure portal):
  - **1. Navigate to the App Service page of the web app, API app, or mobile app.**
  - **2. On the WebJobs blade, click Add.**
  - **3. Configure Add WebJob settings.**



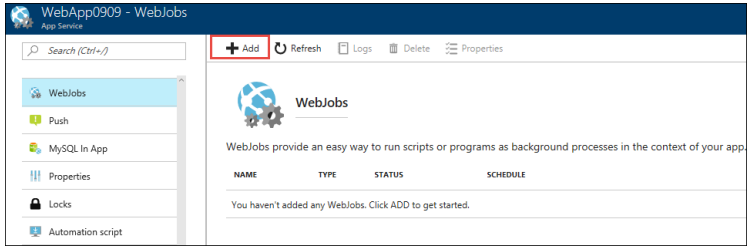


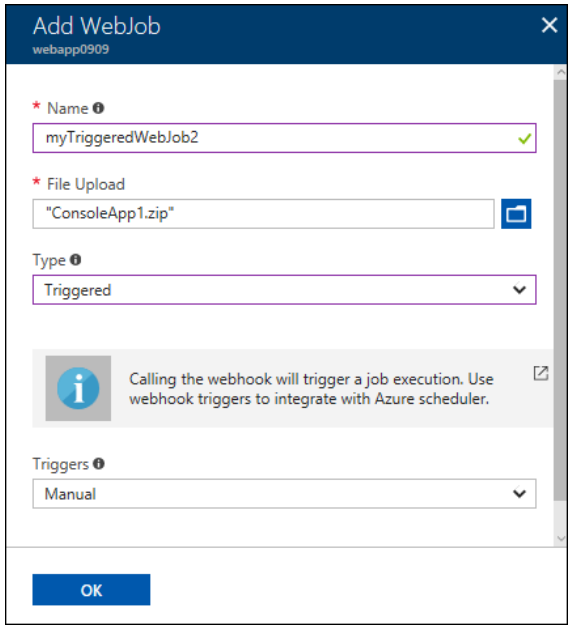


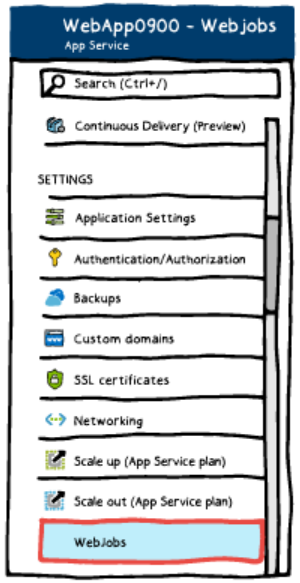
Setting	Sample Value	Description
Name	myContinuousWebJob	A name that is unique within an App Service app. Must start with a letter or a number and cannot contain special characters other than "-" and "_".
File Upload	ConsoleApp.zip	A *.zip* file that contains your executable or script file as well as any supporting files needed to run the program or script. The supported executable or script file types are listed in *Overview of Webjobs* section of this course.
Type	Continuous	Choose the type of of WebJob you want to create, this example uses *continuous*
Scale	Multi instance	Available only for Continuous WebJobs. Determines whether the program or script runs on all instances or just one instance. The option to run on multiple instances doesn't apply to the Free or Shared pricing tiers.

Creating a triggered Webjob

- Implementation steps (the Azure portal):
  - **1. Navigate to the App Service page of the web app, API app, or mobile app.**
  - **2. On the WebJobs blade, click Add.**
  - **3. Configure Add WebJob settings.**



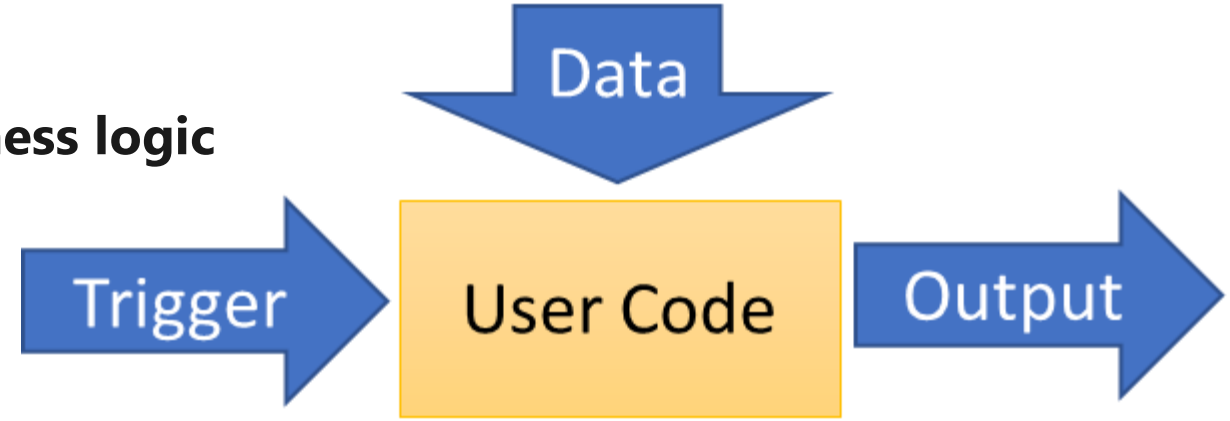




Setting	Sample Value	Description
Name	myContinuousWebJob	A name that is unique within an App Service app. Must start with a letter or a number and cannot contain special characters other than "-" and "_".
File Upload	ConsoleApp.zip	A *.zip* file that contains your executable or script file as well as any supporting files needed to run the program or script. The supported executable or script file types are listed in *Overview of Webjobs* section of this course.
Type	Triggered	Choose the type of of WebJob you want to create, this example uses *Triggered*.
Triggers	Manual	

Serverless Computing

- Abstracts servers, infrastructure, and operating systems
- Reacts to events and triggers in near-real time
- Offers a range of benefits:
  - **Eliminates management overhead**
  - **Allows developers to focus on business logic**
  - **Implements flexible scaling**
  - **Reduces costs**



Infrastructure spun-up, scaled, and spun-down when no longer needed

Serverless Applications

- Compute:
  - **Azure Functions**
- Cloud Messaging:
  - **Event Grid**
  - **Service Bus**
- Workflow Orchestration:
  - **Logic Apps**

Comparing Serverless Options

- **Azure Functions vs Logic Apps:**

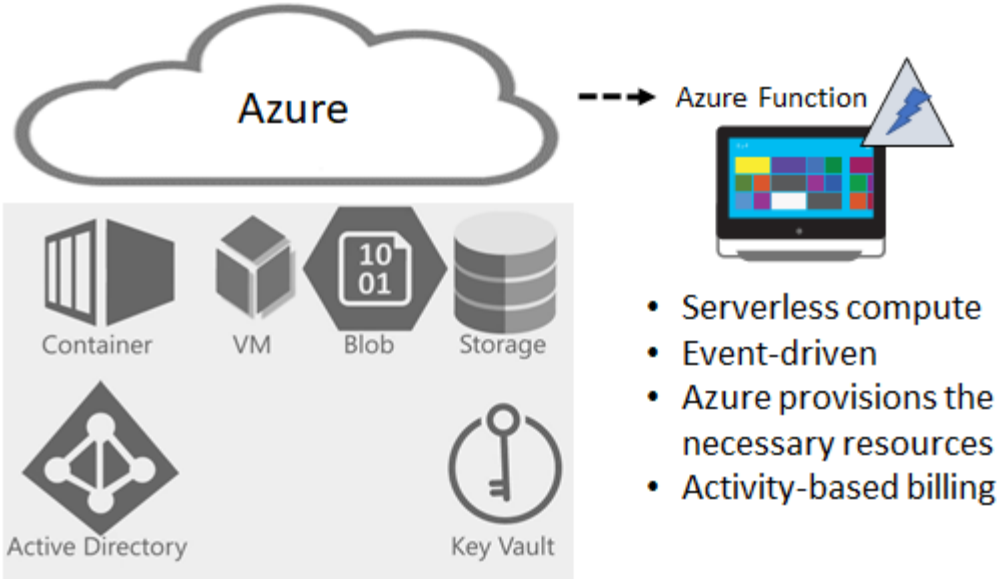
	Durable Functions	Logic Apps
Development	Code-first (imperative)	Designer-first (declarative)
Connectivity	About a dozen built-in binding types, write code for custom bindings	Large collection of connectors, Enterprise Integration Pack for B2B scenarios, build custom connectors
Actions	Each activity is an Azure function; write code for activity functions	Large collection of ready-made actions
Monitoring	Azure Application Insights	Azure portal, Operations Management Suite, Log Analytics
Management	REST API, Visual Studio	Azure portal, REST API, PowerShell, Visual Studio
Execution context	Can run locally or in the cloud.	Runs only in the cloud.

- **Azure Functions and WebJobs:**

	Functions	WebJobs with WebJobs SDK
Serverless app model with automatic scaling	Yes	No
Develop and test in browser	Yes	No
Pay-per-use pricing	Yes	No
Integration with Logic Apps	Yes	No

Overview of Azure Functions

- Serverless compute service:
  - **Eliminates the need to explicitly provision or manage infrastructure**
  - **Enables running code on-demand or in response to a trigger**
  - **Incurs charges only when active**

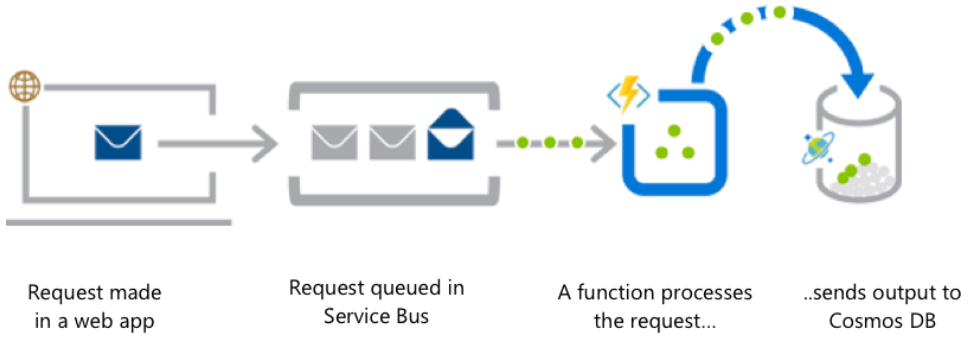


Features of Azure Functions

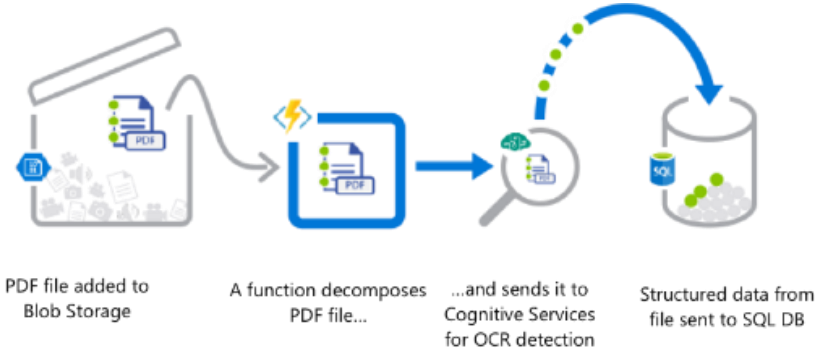
- **Support for a range of programming languages:**
  - **C#, F#, Node.js, Python, PHP, batch, bash, or any executable.**
- **Pay-per-use pricing model**
- **Support for custom dependencies:**
  - **NuGet and NPM-based libraries.**
- **Integration with the most popular OAuth providers:**
  - **Azure AD, Facebook, Google, Twitter, and Microsoft Account.**
- **Integration with other Azure services and SaaS apps.**
- **Flexible development:**
  - **Directly from the Azure portal**
  - **Via continuous integration through GitHub, VSTS, and other supported development tools.**
- **Open-source: available on GitHub.**
- **Ease of code reuse:**
  - **Developers can reuse their functions in multiple applications.**

Azure Functions (Examples)

- **Web app backend**



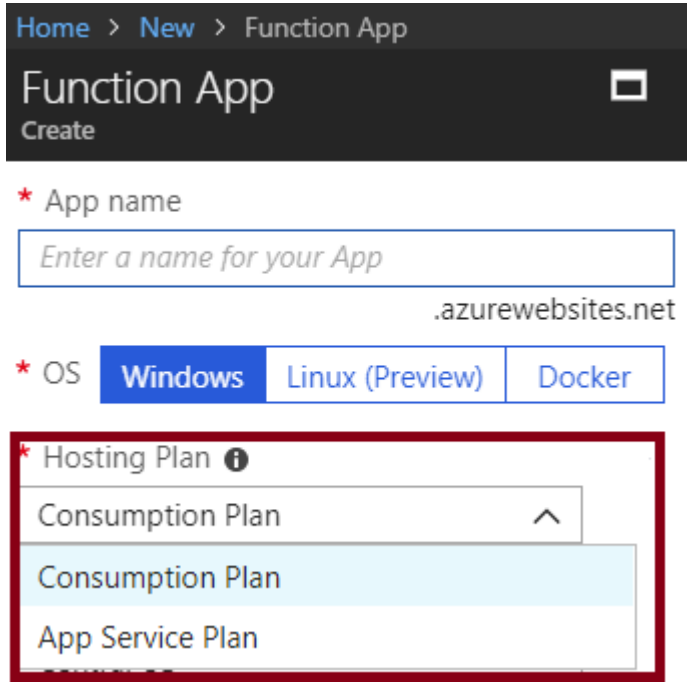
- **Real-time file processing**



- **Automation of scheduled tasks**

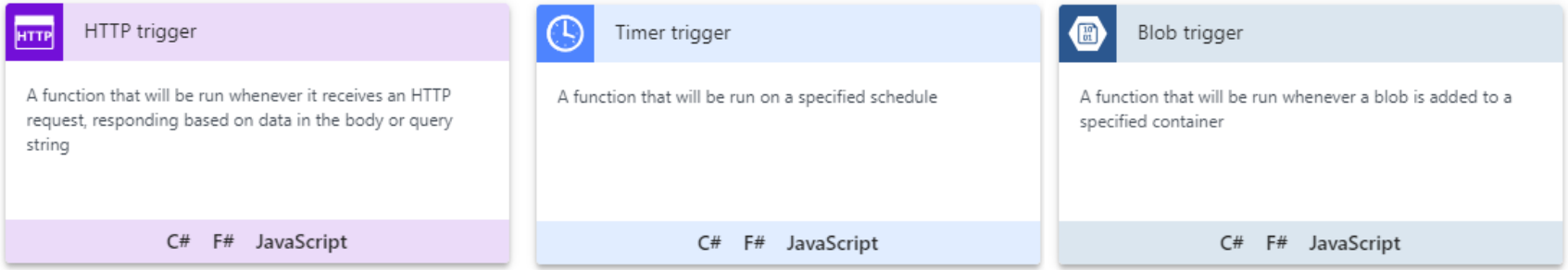
Function Service Plans

- Hosting plans:
  - **Consumption Plan:**
    - **Azure provides all the necessary compute resources on-demand**
    - **Automatic scaling of CPU and memory**
    - **Billing is based on:**
      - **number of executions**
      - **execution time**
      - **memory used.**
  - **App Service Plan:**
    - **The same resource and billing model as for web, mobile, and API apps**
    - **Allows for leveraging underutilized App Service deployments**
    - **Required for Linux-based function apps**



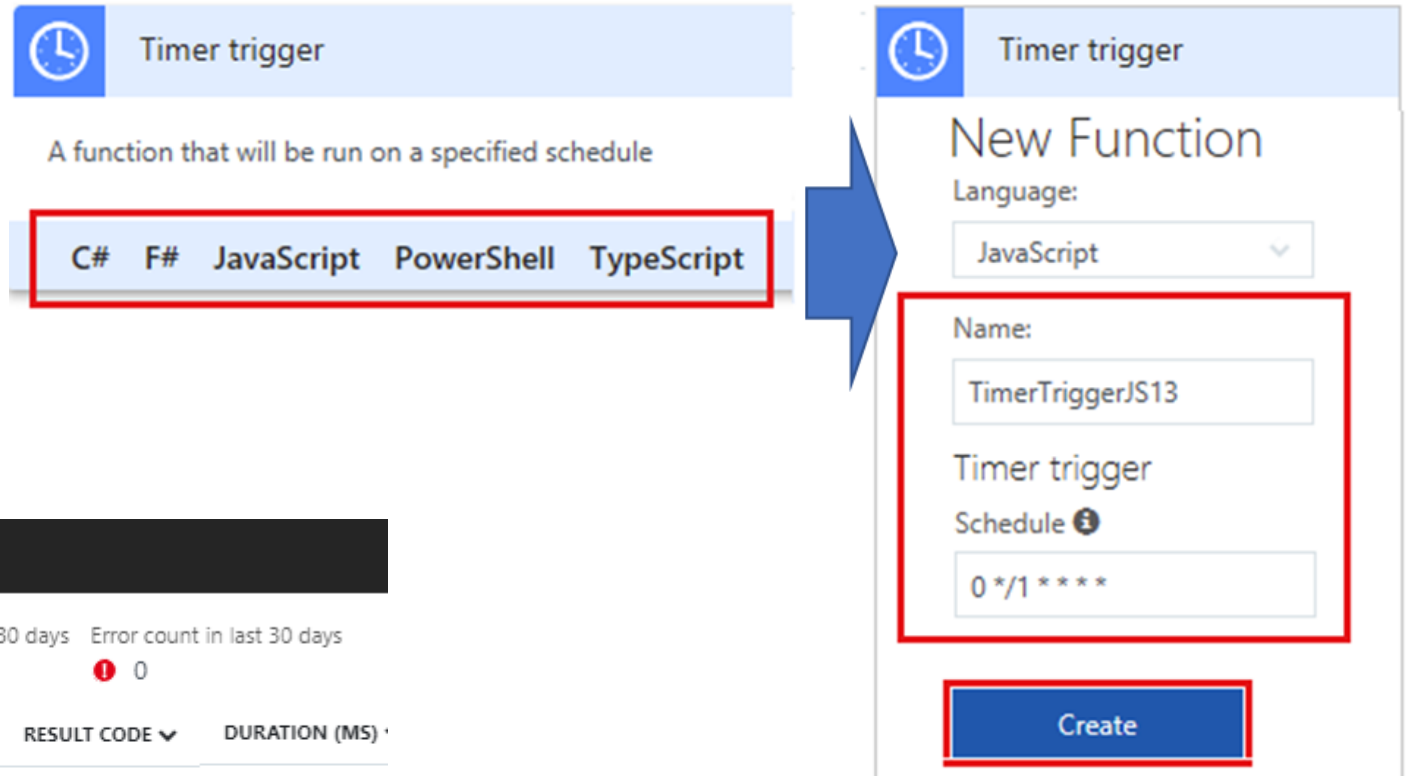
Function Templates

- Simplify creating functions for a specific trigger type:
  - **HTTP**
  - **Timer**
  - **Blob**
  - **Event Hub**
  - **GitHub**
  - **Webhook**
  - **Queue**

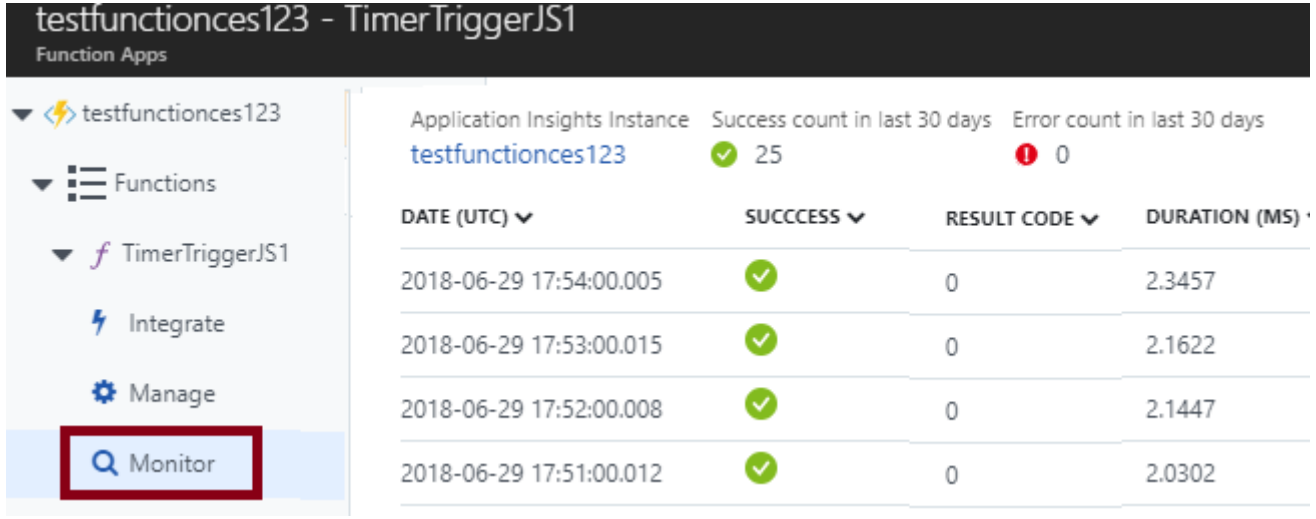


Implementing Functions

- In-portal authoring

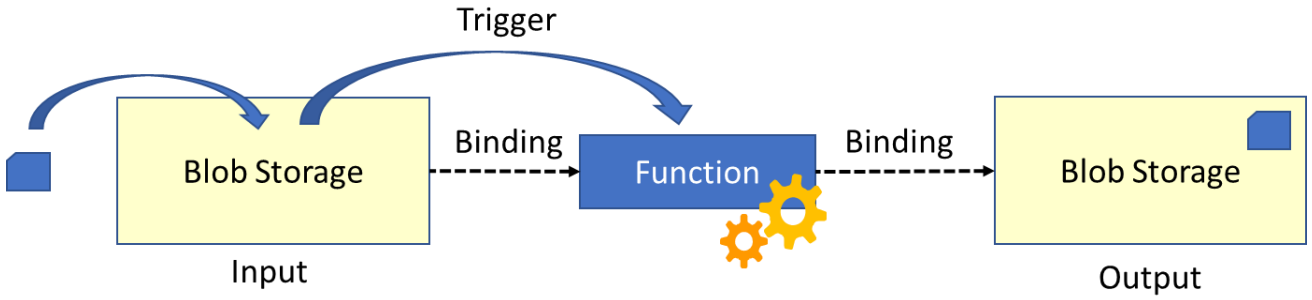


- In-portal monitoring

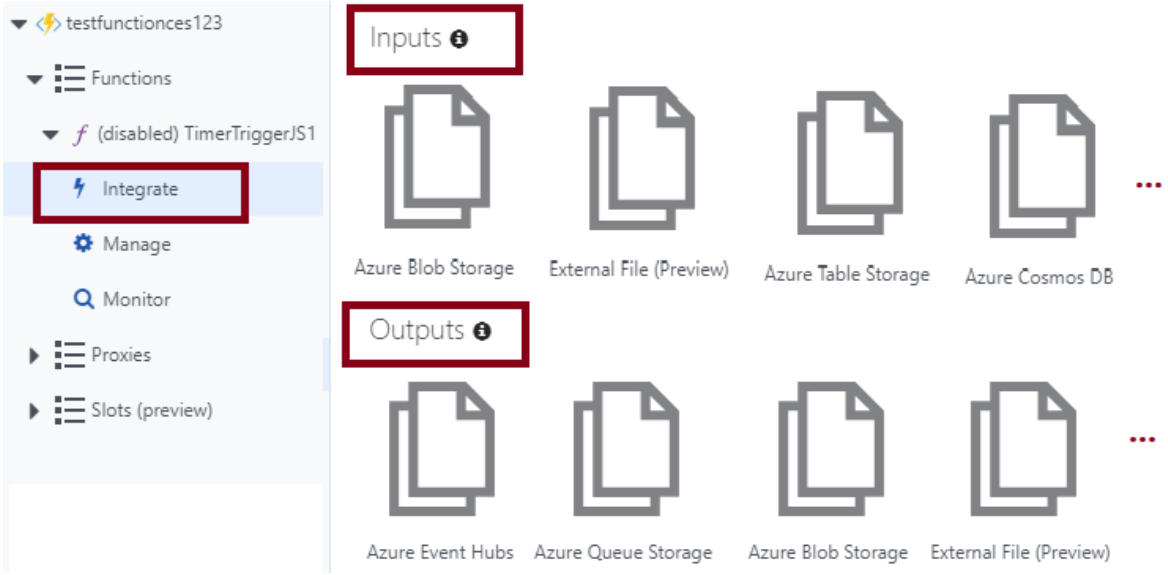


Bindings

- Link functions to other services as:
  - **Inputs**
  - **Outputs**

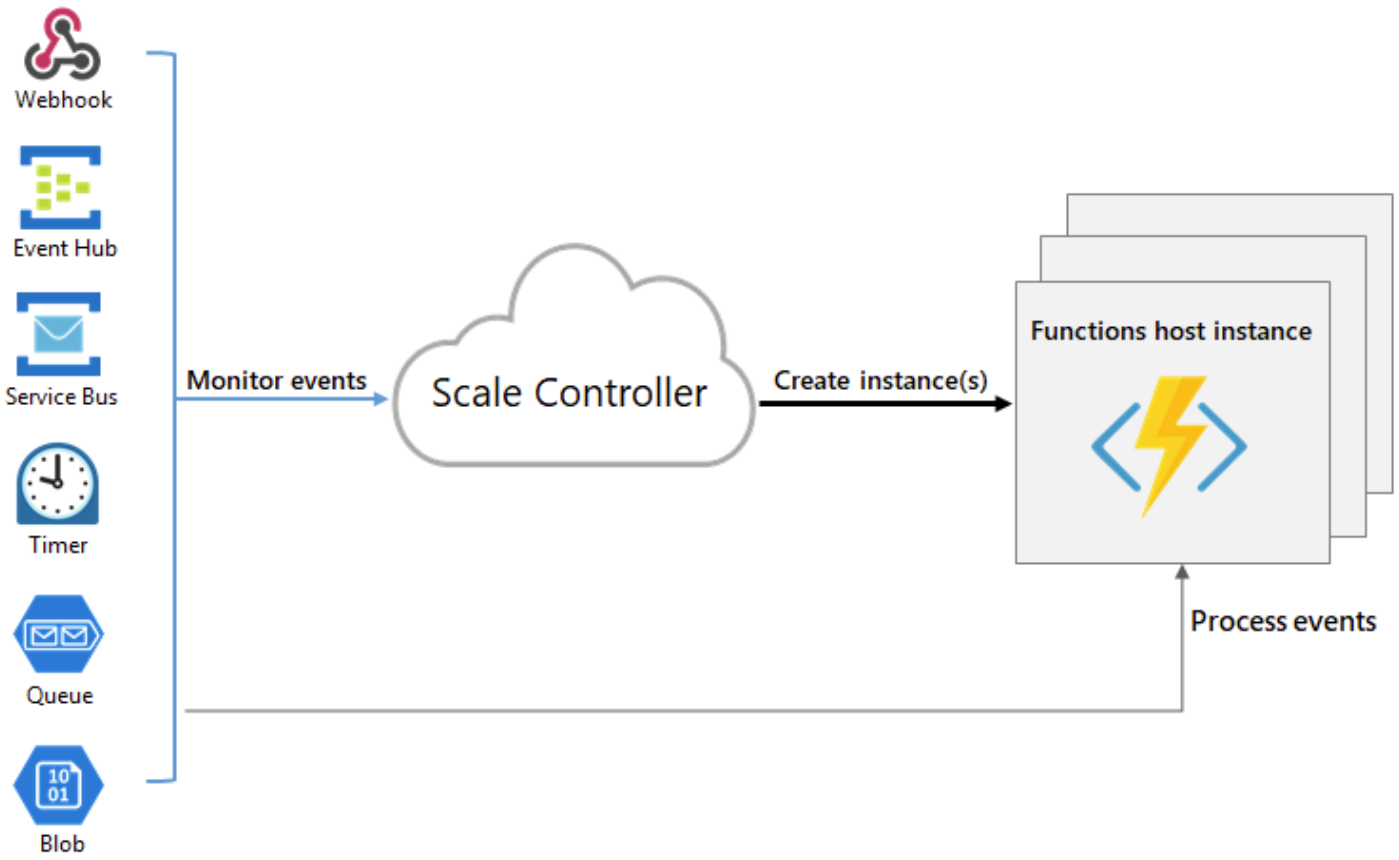


- Configurable in the Azure portal:
  - **Accessible via the Integrate link**



Function Scaling

- Relies on the platform-managed Scale Controller
- Scale Controller:
  - **Monitors the rate of events**
  - **Controls horizontal scaling**
  - **Relies on heuristics:**
    - **E.g. for Queue Storage trigger:**
      - **The queue length**
      - **The age of the oldest message**

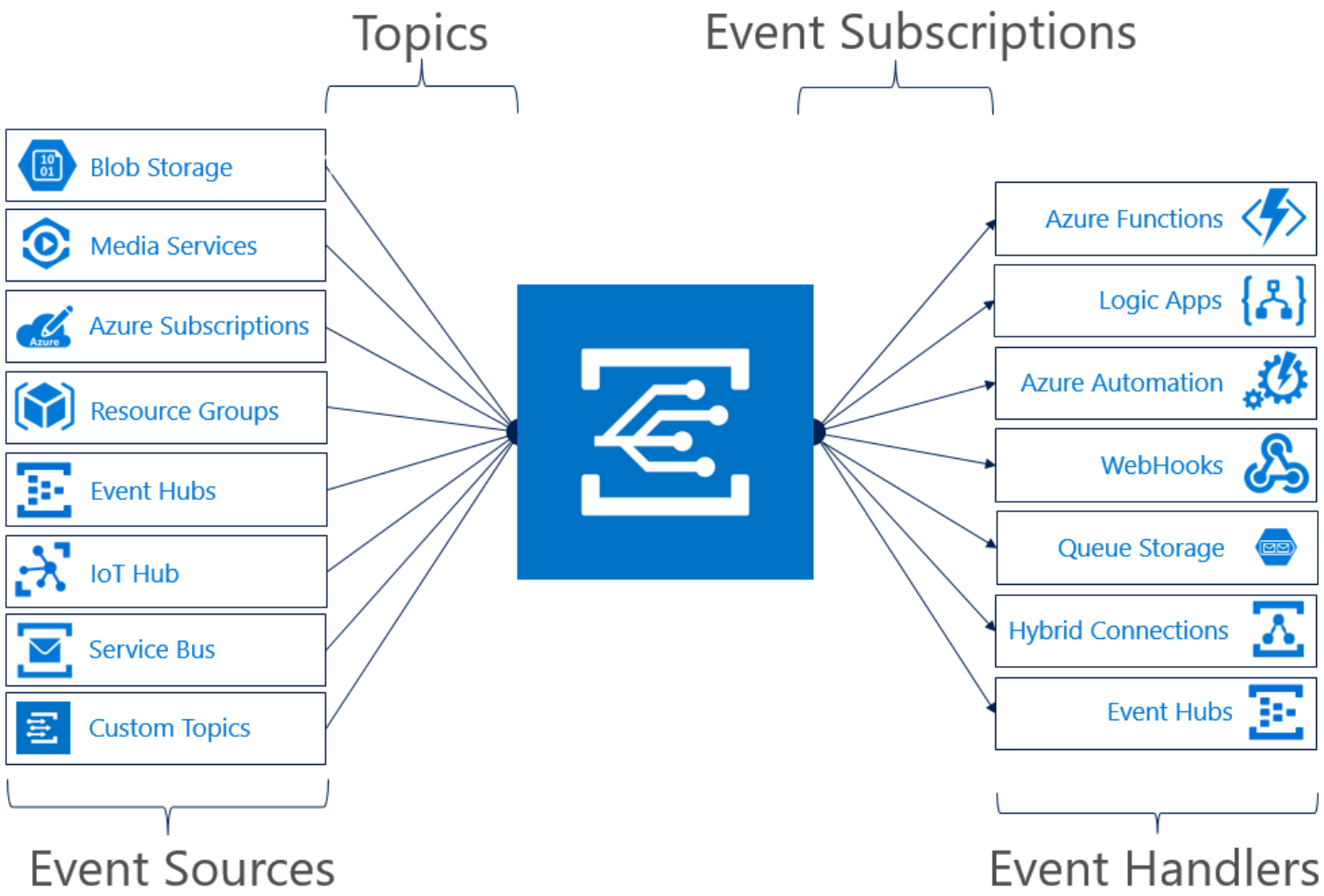


Overview of Event Grid

- Managed event routing service with a wide range of benefits:
  - **Simplifies event delivery:**
    - **Connects multiple event sources and destinations, including virtually all Azure services and custom sources**
    - **Provides fully managed event delivery, intelligent filtering, and the ability to send events to multiple recipients at once**
    - **Eliminates polling and the associated cost and latency**
    - **Decouples event publishers from subscribers by using a pub/sub model and HTTP-based event delivery,**
    - **Simplifies building scalable serverless applications, microservices, and distributed systems.**
  - **Facilitates developing reliable cloud applications:**
    - **Supports massive auto-scaling with near-real-time notifications for changes**
    - **Implements reactive programming leveraging guaranteed event delivery**
  - **Promotes focus on product innovation**

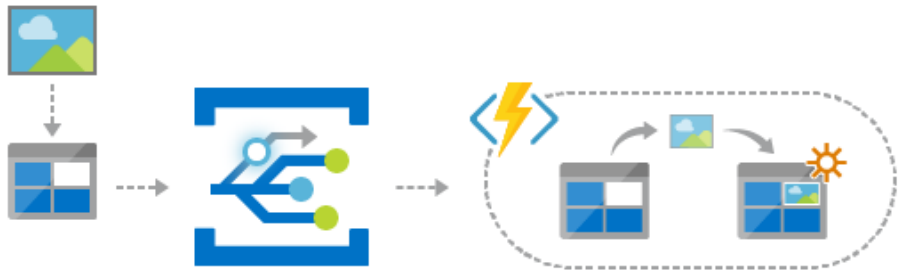
Event Grid Concepts

- Event Sources
- Topics
- Event Subscriptions
- Event Handlers

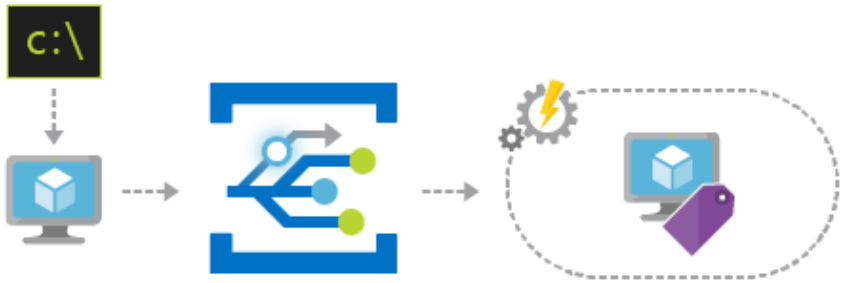


Event Grid Examples

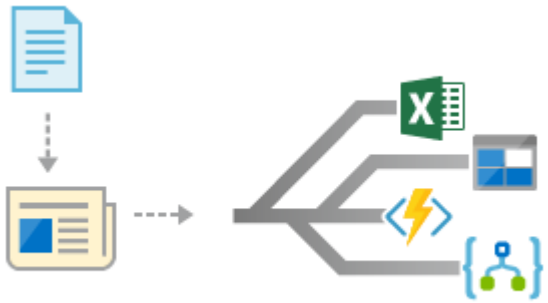
- Serverless application architecture



- Ops automation

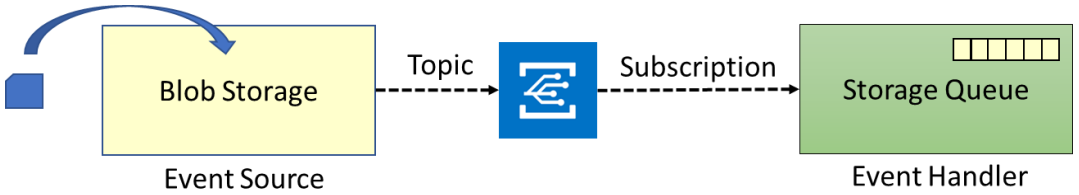


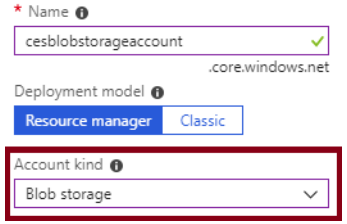
- Application integration

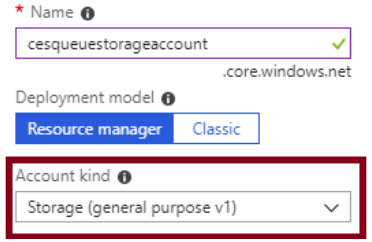


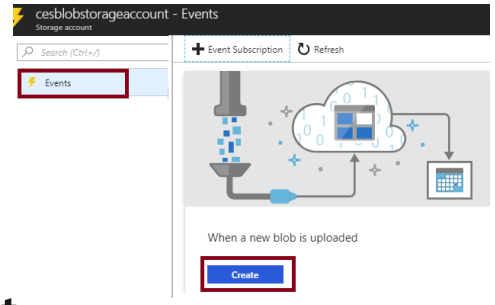
Implementing Event Grid (Part 1)

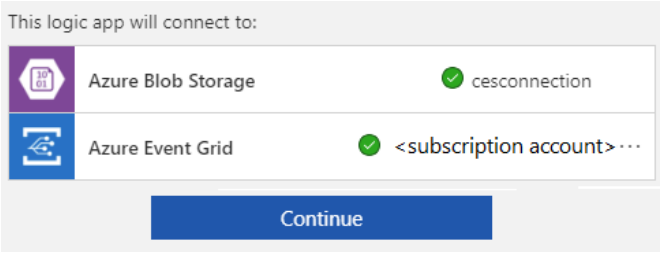
- Implementation components:
  - **Event source: Azure Blob Storage**
  - **Event Handler: Azure Queue Storage**
- Implementation steps (part 1):
  - **1. Create an Azure Storage account (Blob storage)**
  - **2. Create an Azure Storage account (general purpose v1))**
  - **3. Create an Azure Storage queue**
  - **4. Create the event source:**
    - **Use the When a new blob is uploaded template**
  - **5. Configure the event source:**
    - **When prompted, provide the Blob storage account**





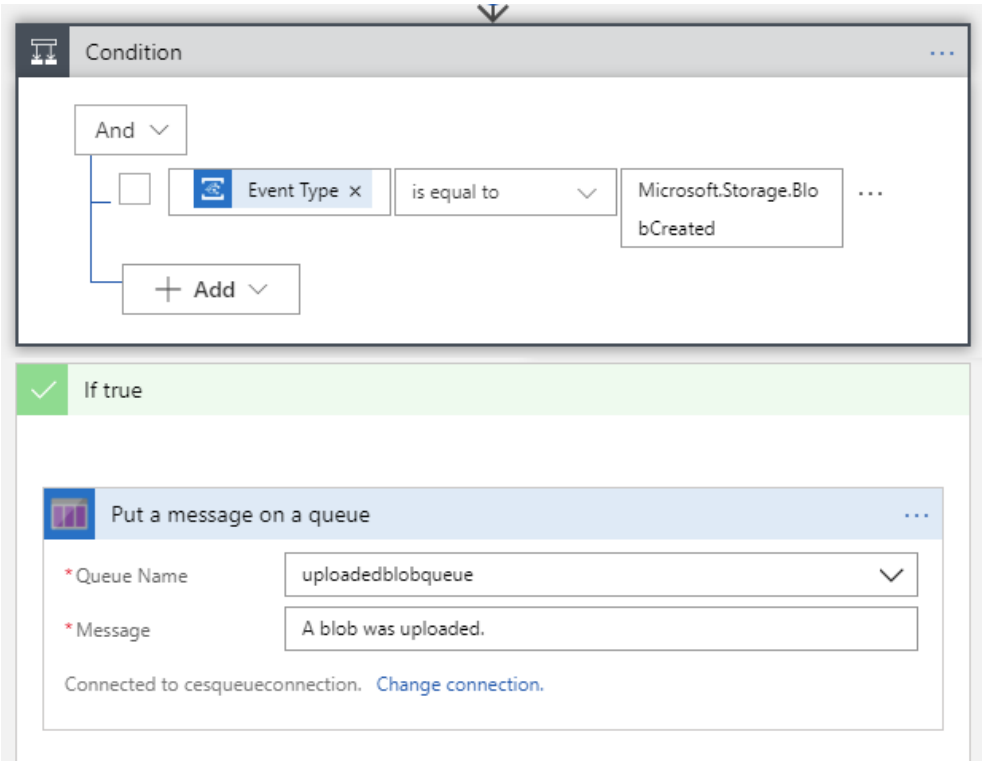






Implementing Event Grid (Part 2)

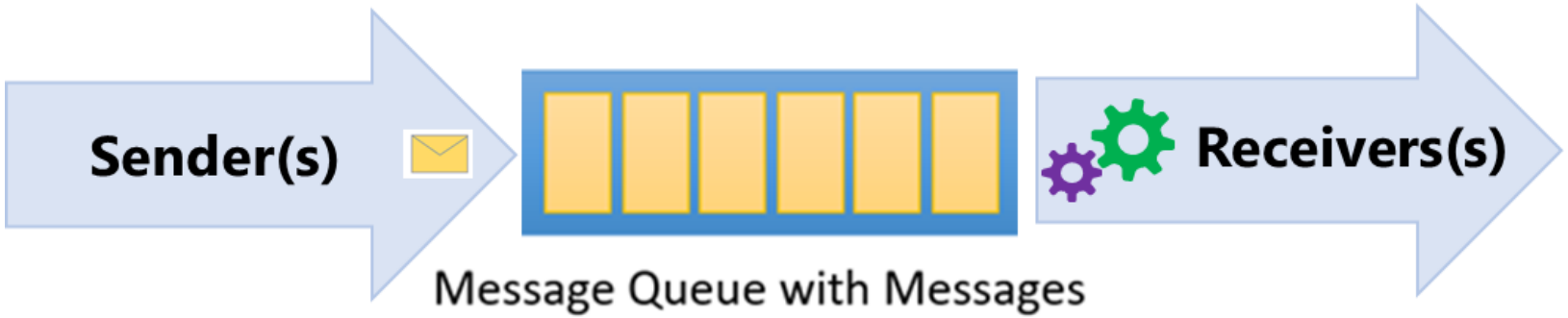
- Implementation steps (part 2):
  - **6. Edit the Logic App:**
    - **One condition: True whenever a blob is created**
    - **One action: Add a message to the queue.**
  - **7. Save changes and start the Logic App**
  - **8. Create an storage container and upload a blob**
  - **9. View messages in the Storage queue**
  - **10. To avoid future charges, delete all resources**



ID	MESSAGE TEXT	INSERTION TIME	EXPIRATION TIME	DEQUEUE COUNT
5d9fe07a-efaa-47f1-a587...	A blob was uploaded.	Thu, 05 Jul 2018 23:55:36 GMT	Thu, 12 Jul 2018 23:55:36 GMT	0
8baedddd-44b2-4dc2-a1...	A blob was uploaded.	Thu, 05 Jul 2018 23:55:54 GMT	Thu, 12 Jul 2018 23:55:54 GMT	0

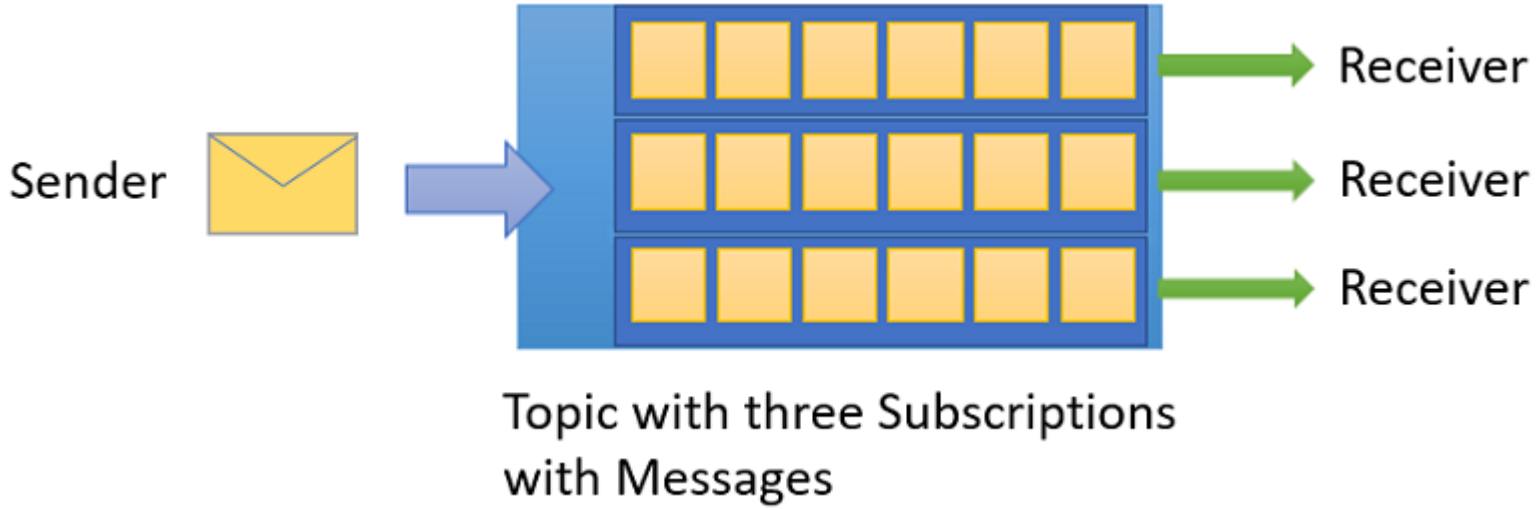
Queues

- Azure Service Bus Queues provide:
  - **Asynchronous, brokered messaging**
  - **Temporal decoupling of message senders and receivers**
  - **Structured message processing**
  - **Publish/subscribe capabilities**
  - **First In, First Out (FIFO) message delivery**



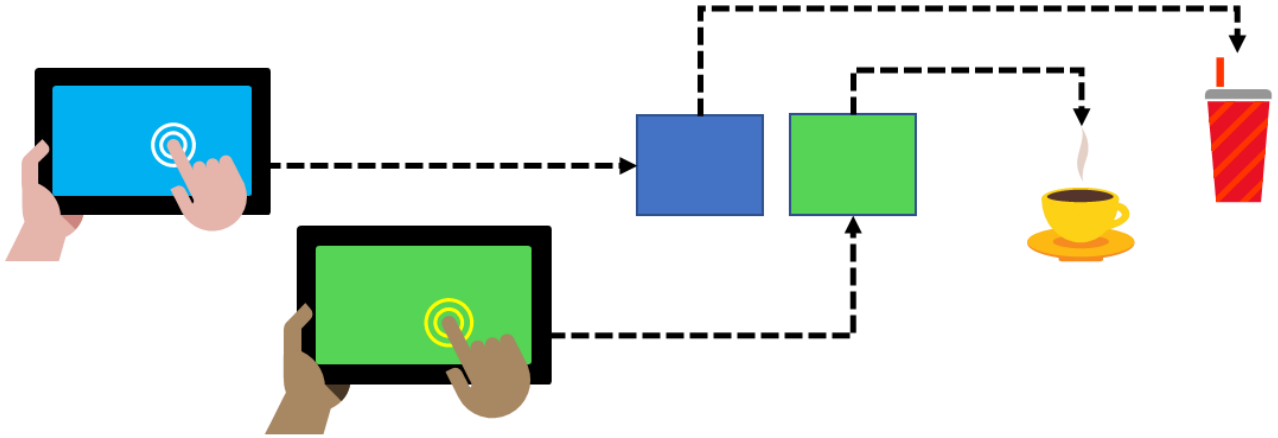
Topics and Subscriptions

- One-to-many communication in a publish/subscribe pattern
- Useful for scaling large number of recipients
- Each published message is available to topic-registered subscriptions
- Subscriptions use filters to designate messages to receive



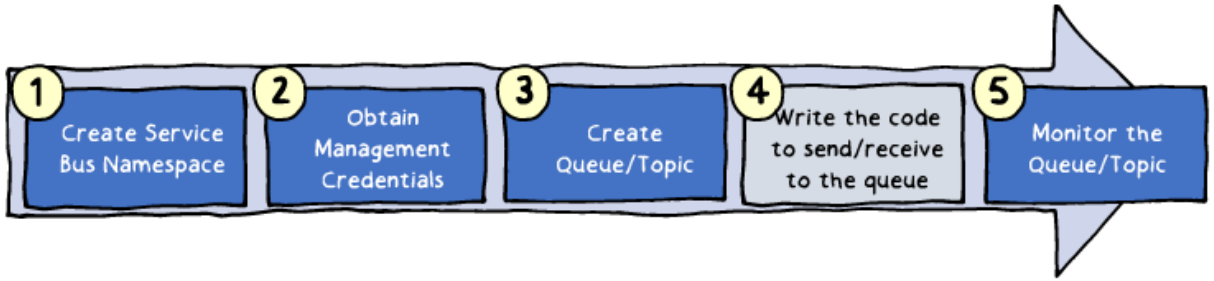
Service Bus Features

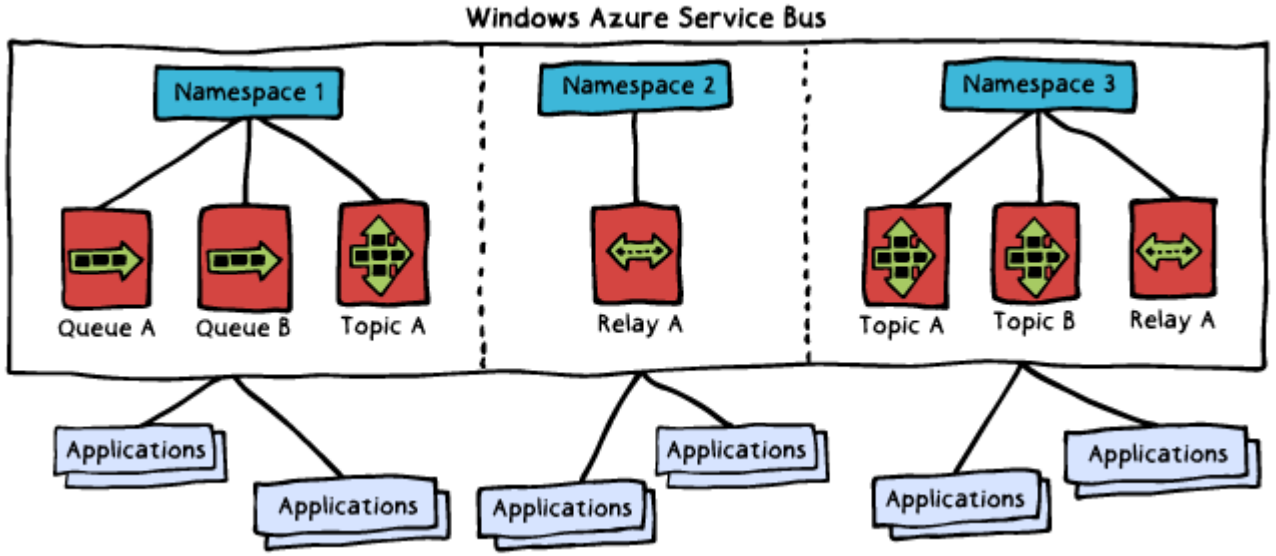
- Load Leveling
- Loose Coupling
- Load Balancing



Implementing Service Bus

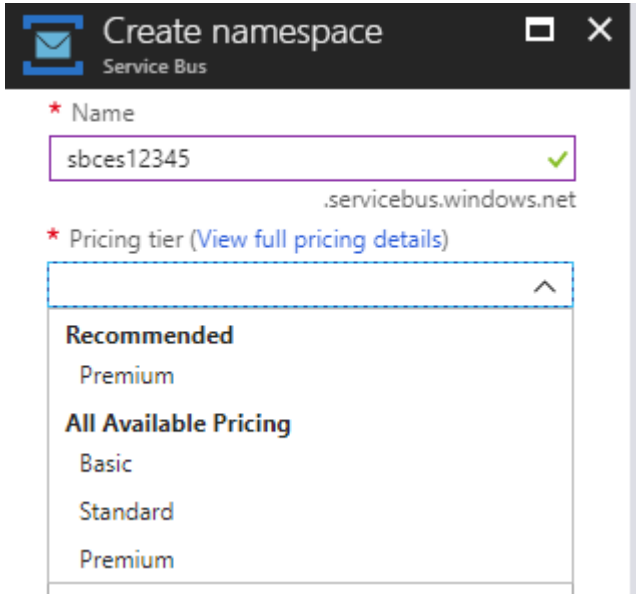
- Implementation steps:
  - **1. Creating a Service Bus namespace**
  - **2. Obtaining management credentials**
  - **3. Creating queues and topics**
  - **4. Writing code to send/receive messages**
  - **5. Managing and monitoring queues and topics**

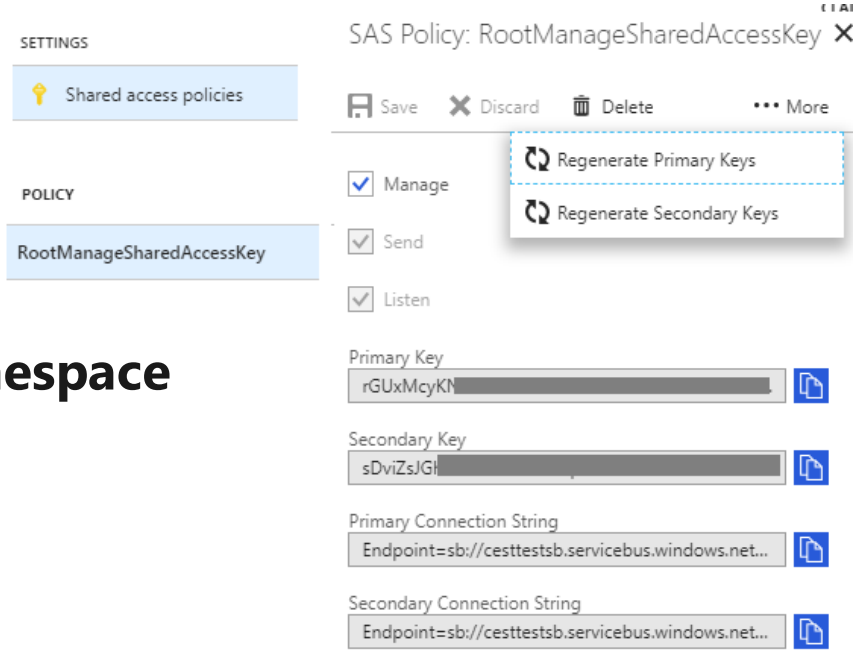




Creating the Namespace

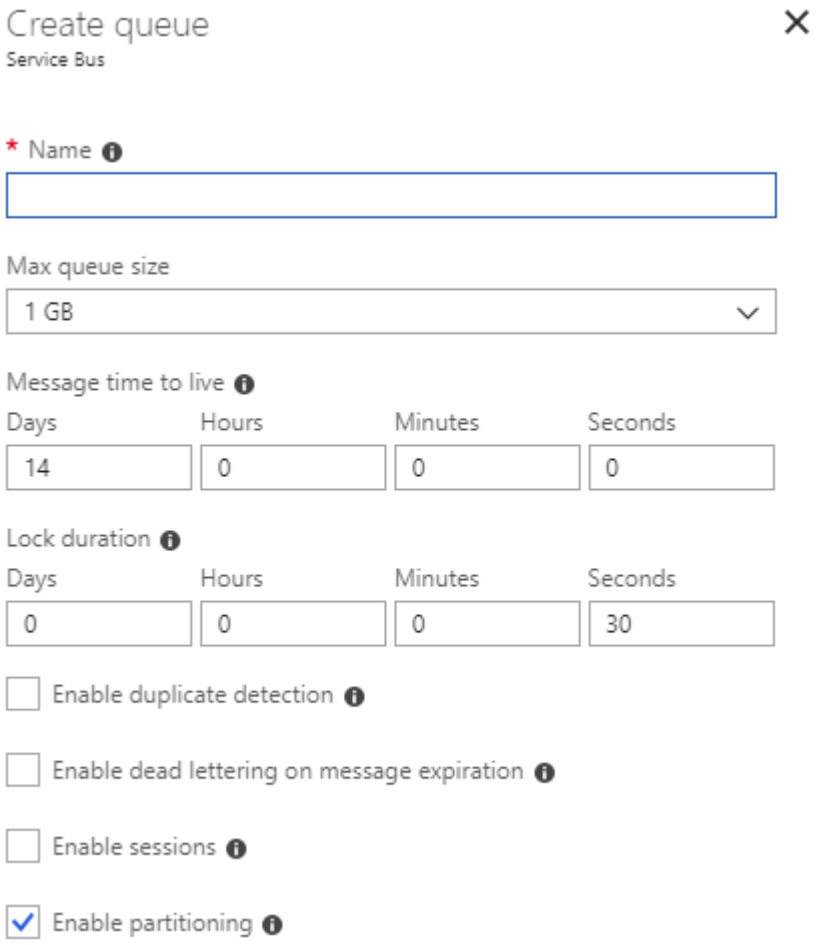
- Available directly from the Azure portal:
  - **Name: globally identifies the namespace**
  - **Pricing tier: determines capacity and capabilities**
    - **Basic**
    - **Standard**
    - **Premium**
- Generates a SAS policy:
  - **Includes:**
    - **Primary key**
    - **Secondary key**
  - **Grants full control to the namespace**





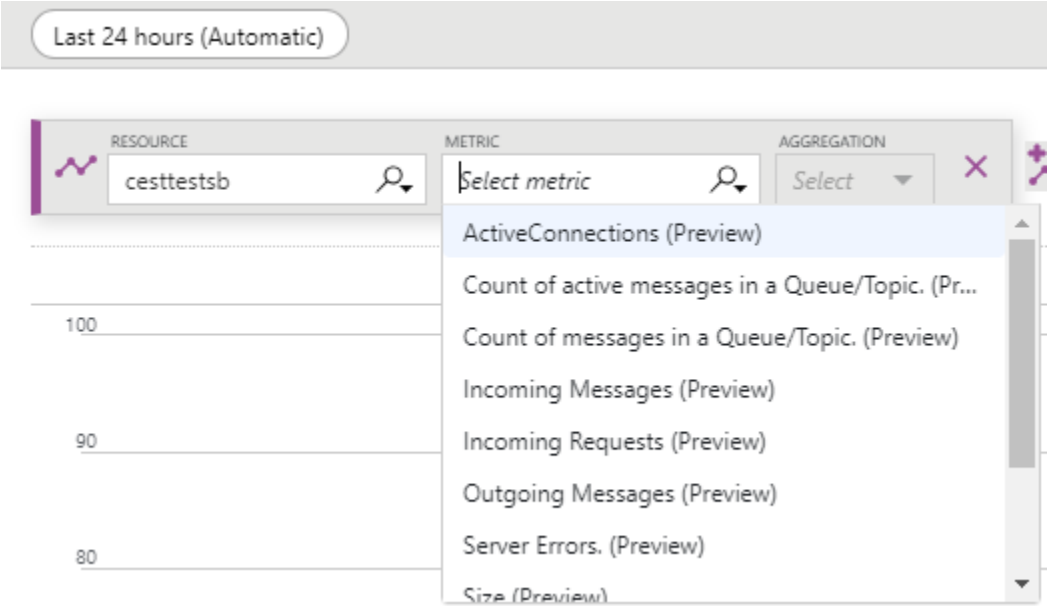
Create a Queue

- Available directly from the Azure portal:
  - **Name: identifies the queue within the namespace**
  - **Max queue size**
  - **Message time to live**
  - **Lock duration**
  - **Enable duplicate detection**
  - **Enable dead lettering**
  - **Enable sessions**
  - **Enable partitioning**



Monitoring Service Bus

- Available directly from the Azure portal:
  - **A rich set of metrics:**
    - **Namespace level**
    - **Queue/topic/message level**
  - **Diagnostic logs:**
    - **Capture all queue activities**



Comparing Service Bus and Storage Queues

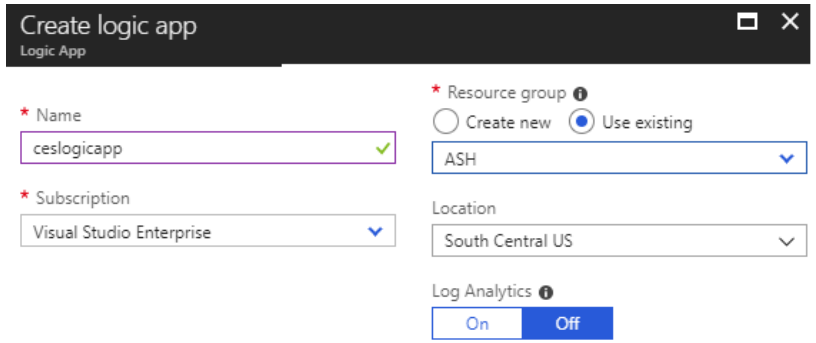
Comparison Criteria	Storage Queues	Service Bus Queues
Ordering guarantee	No	Yes – FIFO
Delivery guarantee	At-Least-Once	At-Least-Once At-Most-Once
Lease/lock level	Message level	Queue level
Batch receive	Yes	Yes
Batch send	No	Yes
Scheduled delivery	Yes	Yes
Automatic dead lettering	No	Yes
Message auto-forwarding	No	Yes
Message groups	No	Yes
Duplicate detection	No	Yes

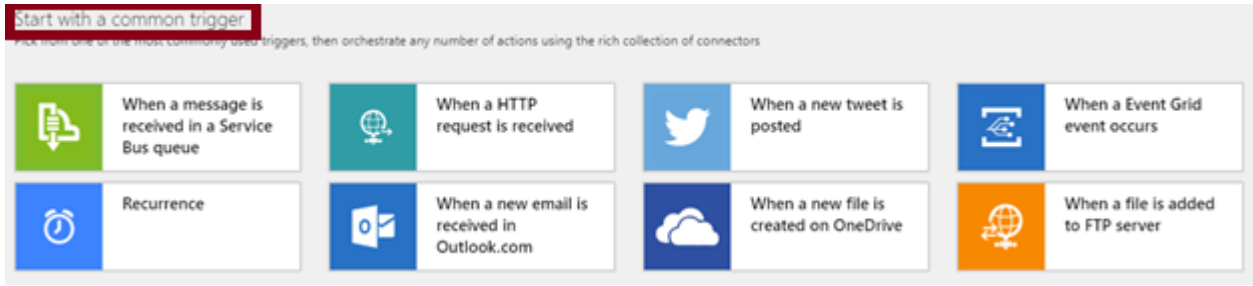
Logic Apps

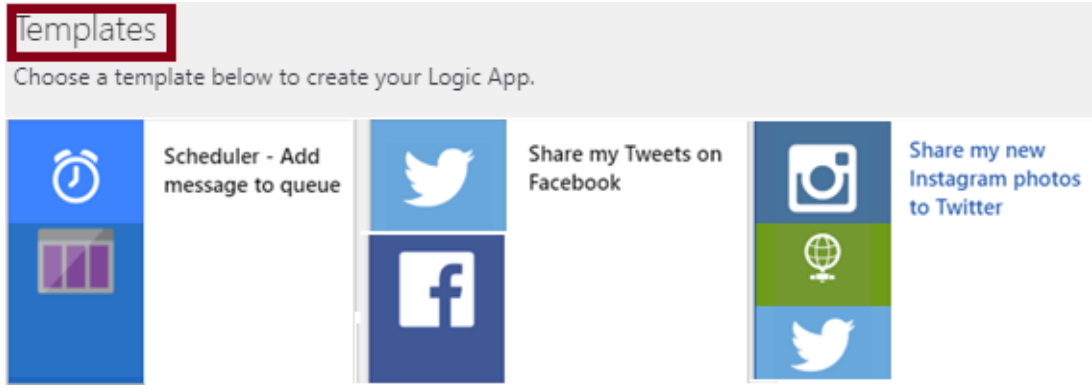
- Implement scalable integrations and workflows
- Include a visual designer to model workflows
- Offer built-in connectors for cloud and on-premises services and apps
- Provide a range of benefits, including:
  - **Getting started quickly from templates.**
  - **Saving time by designing complex processes using easy to understand design tools.**
  - **Implementing patterns and workflows seamlessly, that would otherwise be difficult to implement in code.**
  - **Customizing your logic app with your own custom APIs, code, and actions.**
  - **Connecting and synchronizing disparate systems across on-premises and the cloud.**

Implementing Logic Apps

- From the Azure portal:
  - **To create a logic app, provide:**
    - **Name**
    - **Subscription**
    - **Resource Group**
    - **Location**
  - **To configure a logic app:**
    - **Use the Logic Apps Designer to:**
      - **Add triggers**
      - **Add connectors**
      - **Leverage templates**

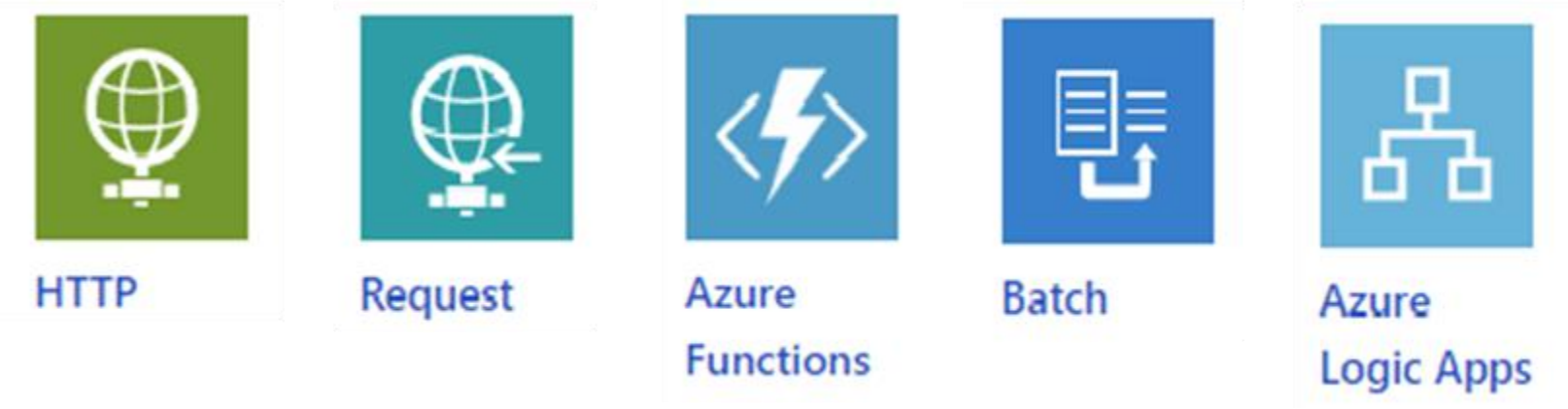




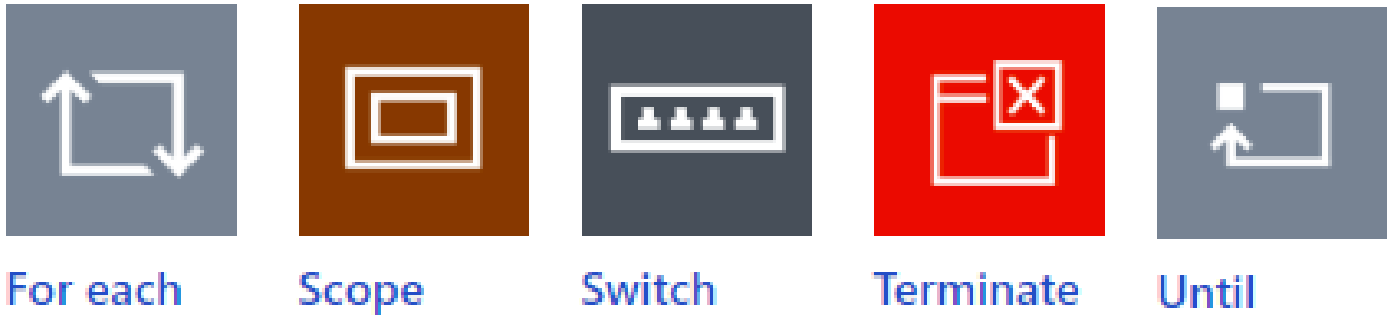


Built-in Triggers and Actions

- **Triggers:**
  - **Recurrence**
  - **HTTP**
  - **Request**
  - **Azure Functions**
  - **Batch**
  - **Logic Apps**

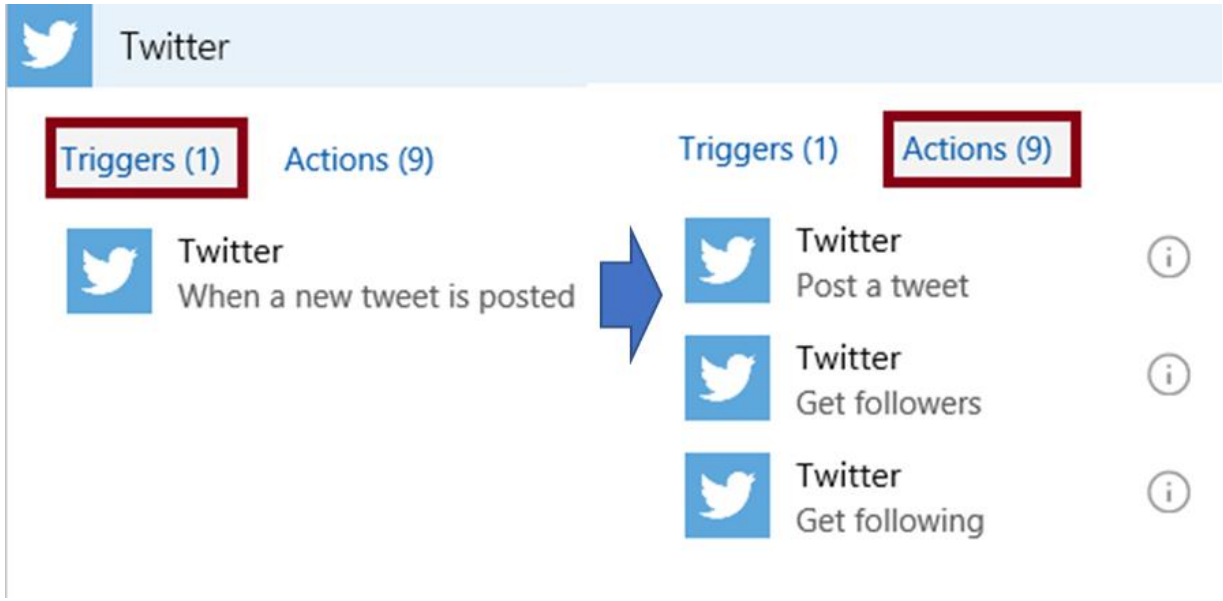


- **Actions:**
  - **Condition**
  - **For each**
  - **Scope**
  - **Switch**
  - **Terminate**
  - **Until**



Managed Connectors

- 200+ built-in connectors, including:
  - **Managed API connectors:**
    - **Azure Blob Storage, Office 365, Dynamics, Power BI, OneDrive, Salesforce, and SharePoint Online.**
  - **On-premises connectors:**
    - **SQL Server, SharePoint Server, Oracle DB, Twitter, Salesforce, Facebook, and file shares.**
  - **Integration account connectors:**
    - **Require a paid-for integration account**
    - **Transform and validate XML**
    - **Encode and decode flat files**
    - **Process B2B messages**
  - **Enterprise connectors:**
    - **Incur extra cost**
    - **Provide access to enterprise systems:**
      - **Support SAP and IBM MQ.**

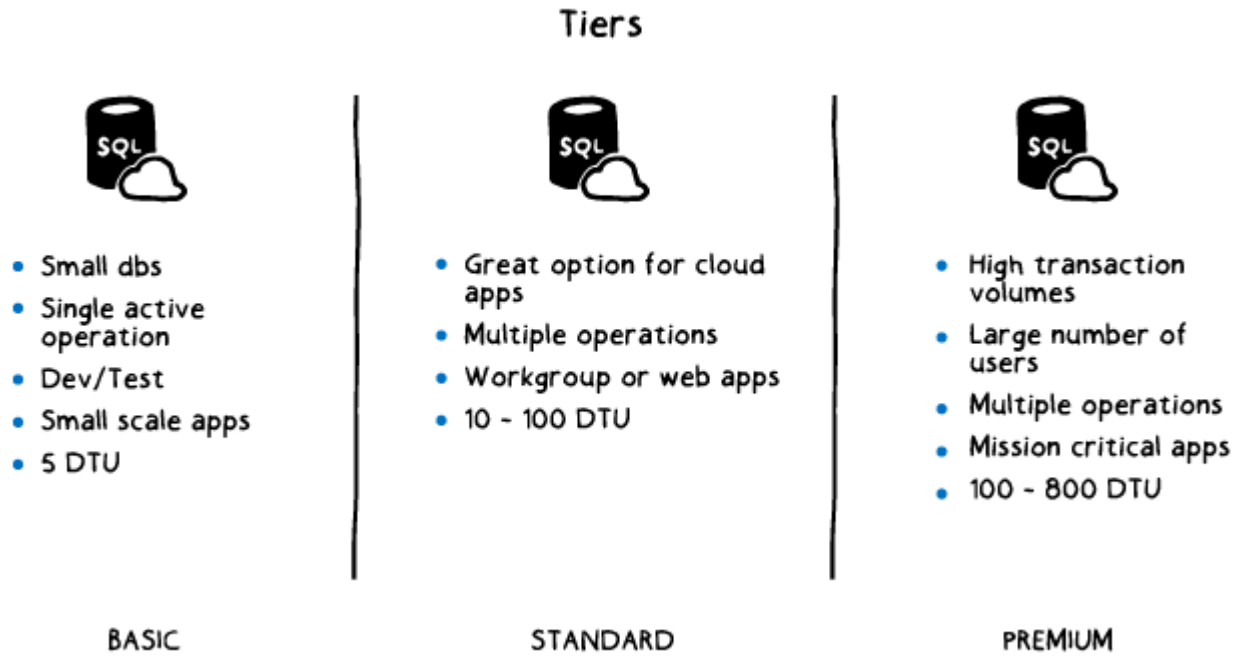


Azure SQL Database

- A relational Database-as-a-Service offering that provides:
  - **Predictable performance: measured in Database Throughput Units (DTUs)**
  - **High compatibility: supporting existing SQL client applications via a Tabular Data Stream (TDS) endpoint**
  - **Simplified management: including SQL Server-specific and Azure tools**

Azure SQL Database Tiers

- Pricing tiers:
  - **Basic: small databases with single concurrent user**
  - **Standard: Medium size databases that must support multiple concurrent connections**
  - **Premium: Large databases that must support large number of concurrent connections and operations**
- Performance levels:
  - **Categories within service tiers**
  - **Provide more granular scaling**
  - **Determine:**
    - **Maximum DTUs**
    - **Maximum database size**



Azure SQL Database Elastic Scale

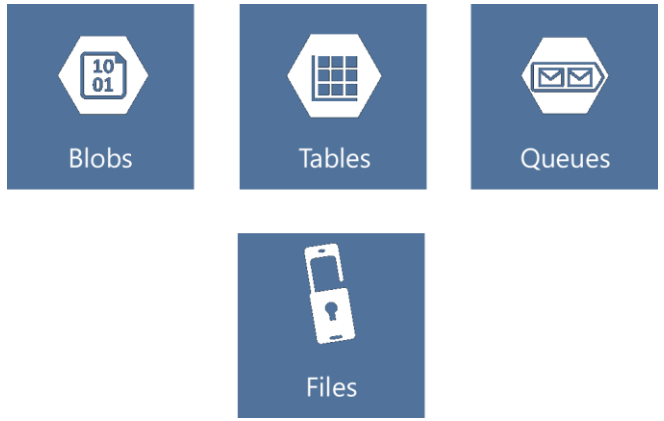
- Facilitates horizontal scaling and includes:
  - **Elastic Scale library for client applications to configure and access shards**
  - **Elastic Scale server-side component that implements sharding**
- Relies on the shard-based scaling strategy:
  - **Uses a shard key to determine partitioning of the database**
  - **Automatically directs transactions to the appropriate shard**
  - **Coordinates data movement between shards to split or merge data ranges**

Third-Party Databases in Azure

- Managed database options:
  - **Offer:**
    - **Built-in high availability with no additional cost.**
    - **Predictable performance, using pay-as-you-go pricing.**
    - **Automatic scaling.**
    - **Encryption at-rest and in-transit.**
    - **Automatic backups and point-in-time-restore for up to 35 days.**
    - **Enterprise-grade security and compliance.**
  - **Include:**
    - **Azure Database for MySQL**
    - **Azure Database for PostgreSQL**
- Non-managed database options:
  - **Windows and Linux Azure VMs hosting MySQL installations**
  - **ClearDB offering managed MySQL instance**

Azure Storage

- Massively scalable
- Uses partitioning system that facilitates load-balanced data access
- Offers different types of storage services:
  - **Blobs**
  - **Tables**
  - **Queues**
  - **Files**



- Accessible via:
  - **REST API**
  - **Client libraries, including .NET, Java/Android, Node.js, PHP, Ruby, and Python**

Replication

- Content of each storage account is always replicated:
  - **Ensures durability and high availability**
- Scope of replication is configurable:
  - **Locally redundant storage (LRS):**
    - **Consists of 3 copies of the same content in the same availability zone in one region**
  - **Zone-redundant storage (ZRS):**
    - **Consists of 3 copies of the same content across multiple zones in one region**
  - **Geo-redundant storage (GRS):**
    - **Consists of 6 copies of the same content in two regions, with 3 copies per region**
    - **Facilitates failover to a remote region in case of the primary region failure**
  - **Read access geo-redundant storage (RA-GRS):**
    - **Consists of 6 copies of the same content in two regions, with 3 copies per region**
    - **Facilitates failover to a remote region in case of the primary region failure**
    - **Provides read access to the remote region**

Azure Storage Tables

- A NoSQL data store:
  - **Provides storage for massive amounts of structured, non-relational data**
  - **Is suitable for datasets that do not require joins, foreign keys, or stored procedures**
  - **Offers access via OData protocol and LINQ queries with WCF Data Service .NET libraries**
  - **Includes a clustered index for data queries**
  - **Includes the following components:**
    - **An Azure Storage account**
    - **A table:**
      - **a collection of entities with no enforced schema**
    - **Entities:**
      - **sets of properties similar to database rows**
    - **Properties:**
      - **Name-value pairs**
      - **Up to 252 custom properties per entity**
      - **3 system properties: PartitionKey, RowKey, and Timestamp**

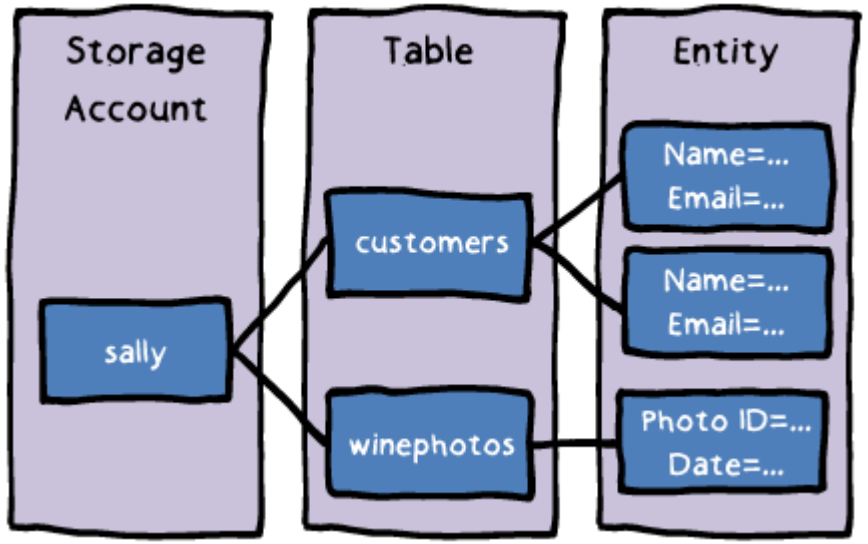
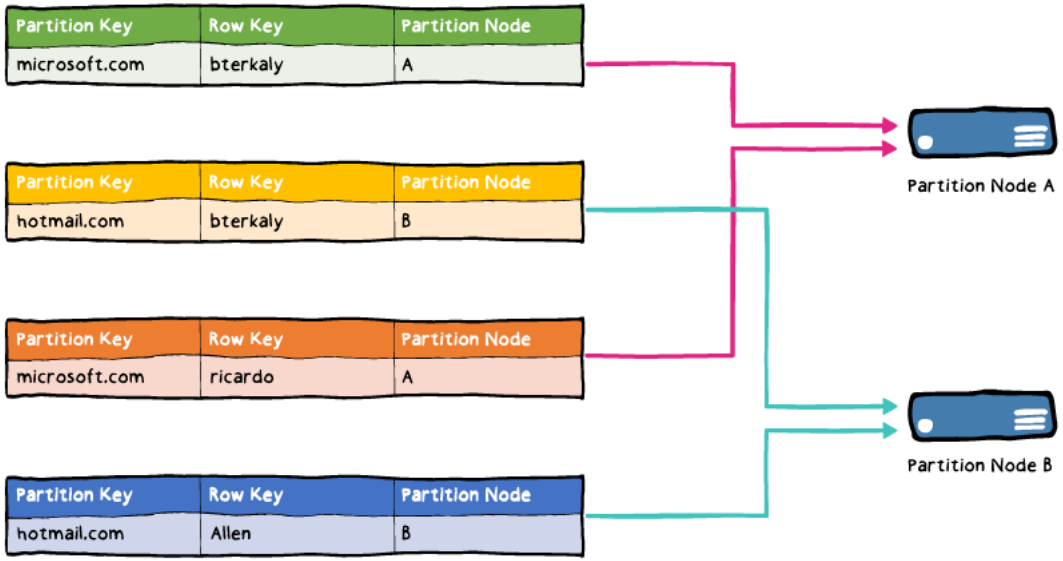


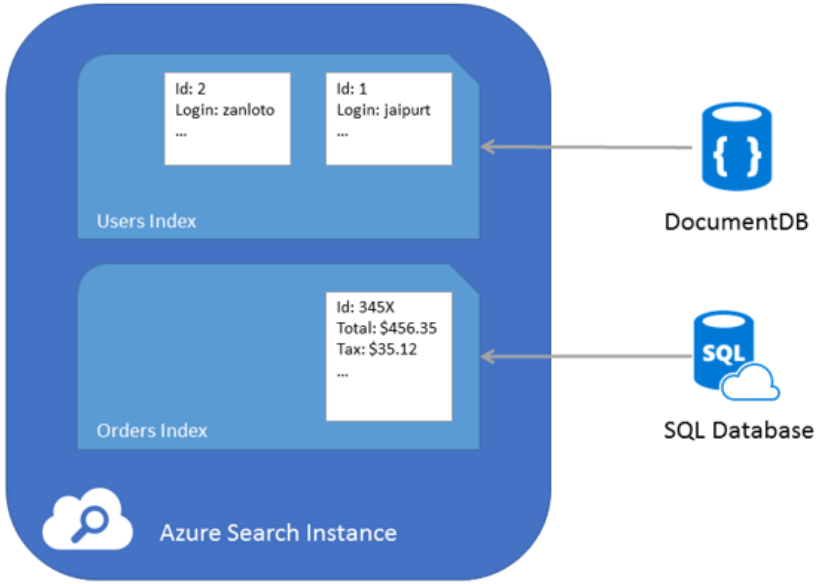
Table Partitioning

- Azure Storage Tables partitions:
  - **Represent collections of entities with the same PartitionKey values**
  - **Determine partitioning of the underlying storage**
  - **A single partition has the scalability target of 500 entities per second**
  - **To scale horizontally, it is necessary to configure multiple partitions**
  - **The choice of PartitionKey is critical:**
    - **It affects horizontal scalability**
    - **It affect indexing behavior:**
      - **The primary key combinesPartitionKey and RowKey**
      - **The clustered index is based on the primary key**



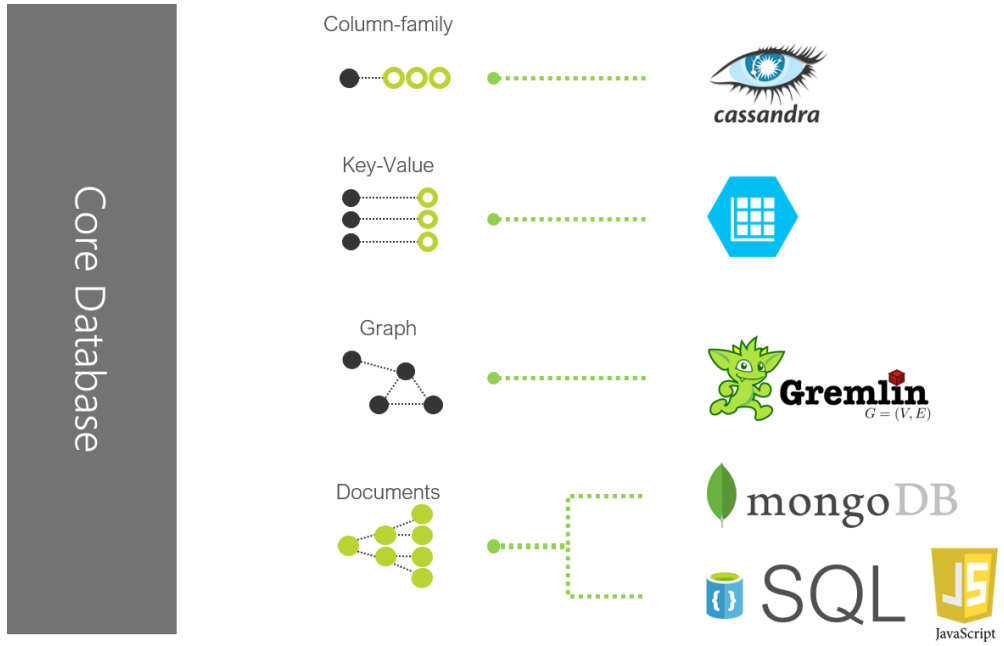
Azure Search

- Managed search service:
  - **Allows developers to build search applications by using .NET SDK or REST API**
  - **Based on automatically generated index with support for custom schemas**
  - **Provides full-text search over custom content**
  - **Offers advanced search capabilities:**
    - **Type-ahead query suggestions**
    - **Hit-highlighting**
    - **Faceted navigation**
    - **Built-in native language support**
  - **Available in two pricing tiers:**
    - **Free: offers limited, shared resources**
    - **Standard: implements autoscaling**
  - **Includes indexing of Azure services:**
    - **Azure SQL Database**
    - **Cosmos DB**



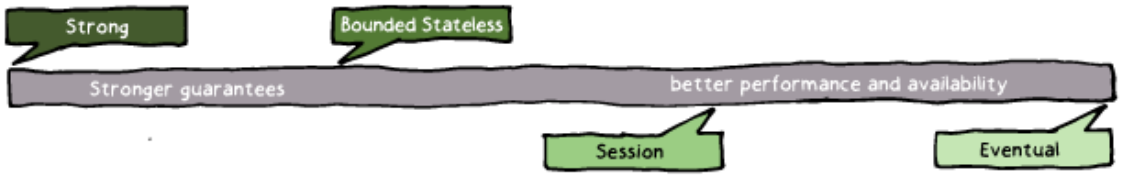
Azure Cosmos DB APIs

- Azure Cosmos DB is a globally distributed, multi-model database:
  - **Accessible via different API (depending on the choice of the model):**
    - **DocumentDB (SQL) API**
    - **MongoDB API**
    - **Graph (Gremlin) API**
    - **Tables (Key/Value) API**
  - **Automatically partitioned:**
    - **based on a custom partition key**
    - **with each partition offering:**
      - **Fixed performance**
      - **Fixed storage capacity**



Consistency Levels

- **Strong:**
  - **Guarantees that a write operation is only committed (and visible) on the primary after it has been committed and confirmed by ALL replicas.**
- **Bounded staleness:**
  - **Allows you to configure how stale documents can be within replicas. Staleness refers to the quantity of time (or version count) a replica document can be behind the primary document.**
- **Session:**
  - **Guarantees that all read and write operations are consistent within a user session.**
- **Consistent prefix:**
  - **Ensures that changes are read in the order that matches the sequence of the corresponding writes.**
- **Eventual:**
  - **Offers the loosest consistency and commits any write operation against the primary immediately. Replica transactions are asynchronously handle and will eventually (over time) be consistent with the primary.**



Choosing a Consistency Strategy

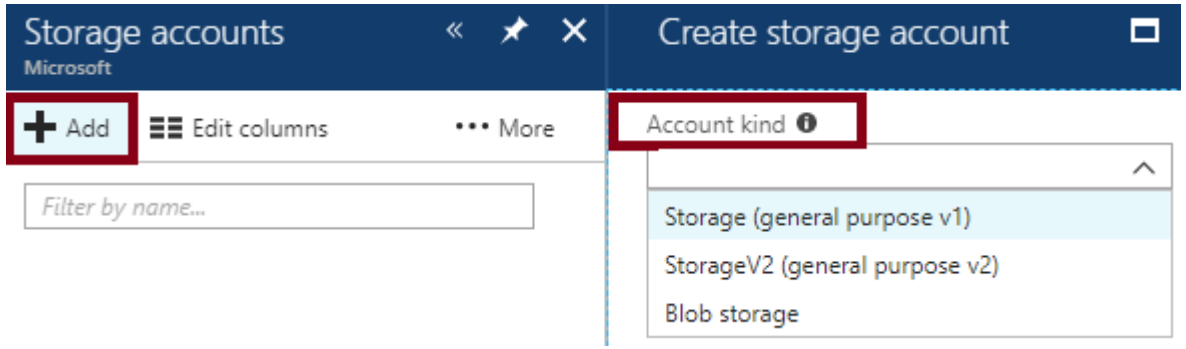
- Two primary considerations:
  - **a stronger consistency level will ensure that versions of documents in replicas do not lag behind the primary:**
    - **Is recommended for applications that require all replicas to exactly match the primary at any point in time**
    - **Affects negatively the speed of write operations, which must wait for every replica to confirm that the operation has been committed**
  - **A weaker consistency level will ensure that your database operates at peak efficiency:**
    - **Is recommended for applications that require the best performance**
    - **It introduces the possibility that read operations against a replica return stale data**

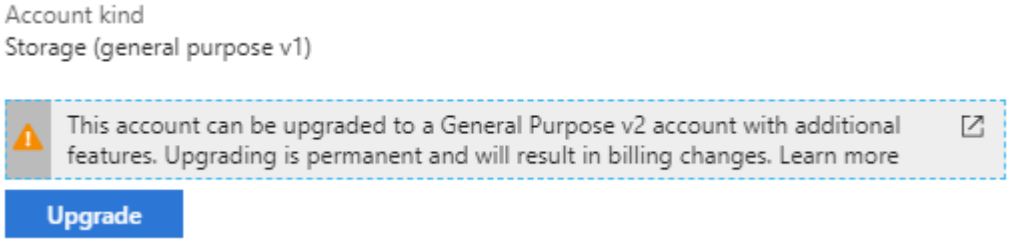
Azure Storage

- Azure Storage serves as:
  - **Storage for Azure VMs:**
    - **operating system and data disks**
    - **file storage**
  - **Store for unstructured data:**
    - **Blobs**
    - **Queues**
    - **Data Lake Store**
  - **Store for structured data:**
    - **Tables**
    - **Cosmos DB**
    - **Azure SQL Database**

Azure Storage Accounts

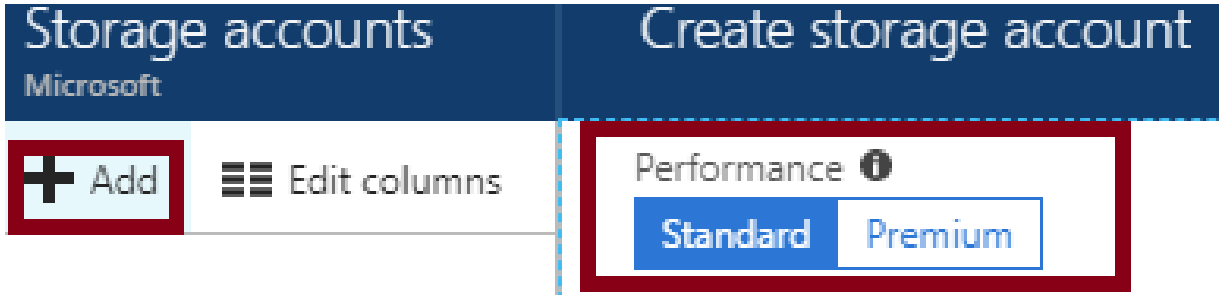
- Storage (general purpose v1):
  - **Can contain blobs (including Azure VM disks), tables, queues, files**
  - **Supports performance tiers: Standard and Premium**
- Blob storage:
  - **Can contain blobs only**
  - **Supports access tiers: hot, cool, archive**
- Storage V2 (general purpose v2):
  - **Can contain blobs (including Azure VM disks), tables, queues, files**
  - **Supports performance tiers: Standard and Premium**
  - **Supports access tiers: hot, cool, archive**
  - **Support upgrade from:**
    - **Blob storage**
    - **Storage (general purpose v1)**





Standard and Premium Storage Accounts

- Performance tiers:
  - **Standard:**
    - **Best for workloads that do not require high performance I/O**
    - **Offer the lowest cost per GB**
    - **Backed by magnetic drives (HDD)**
  - **Premium (available exclusively for provisioning Azure VM disks):**
    - **Best for I/O-intensive workloads, such as databases.**
    - **Offer consistent low-latency, high throughput/IOPS**
    - **Backed by solid state drives (SSD)**



Storage Account Endpoints

- Blob service: http://*mystorageaccount*.blob.core.windows.net
- Table service: http://*mystorageaccount*.table.core.windows.net
- Queue service: http://*mystorageaccount*.queue.core.windows.net
- File service: http://*mystorageaccount*.file.core.windows.net

Configuring Custom Domain Names

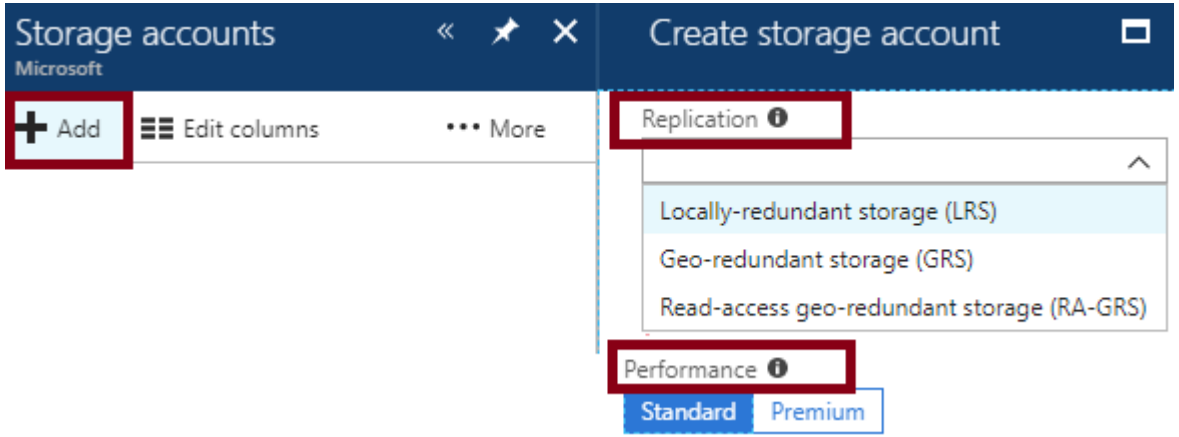
- Direct CNAME mapping:
  - **Create a CNAME record that points to the Azure storage account [storage account].blob.core.windows.net.**
  - **This method can result in a brief downtime during registration of the custom domain with Azure.**
- Intermediary mapping with asverify:
  - **Step 1. Prepend asverify to your own CNAME record. This allows Azure to recognize your custom domain.**
  - **Step 2. Modify the DNS record for the domain.**
  - **This methods eliminates downtime.**

Pricing and Billing

- Storage costs depend on:
  - **The amount of data**
  - **Performance level**
  - **Access tier (per-gigabyte cost decreases as the tier gets cooler)**
- Storage costs include:
  - **Data access costs: increase as the tier gets cooler.**
  - **Transaction costs: increases as the tier gets cooler.**
  - **Geo-Replication data transfer costs (per GB): for GRS and RA-GRS storage accounts.**
  - **Outbound data transfer costs (per GB): transferred out of an Azure region**
  - **Changing the storage tier:**
    - **Changing the account storage tier from cool to hot incurs a charge equal to reading all the data existing in the storage account.**
    - **changing the account storage tier from hot to cool incurs a charge equal to writing all the data into the cool tier (GPv2 accounts only).**

Replication Options

- Locally-redundant storage (LRS):
  - **The only replication option when using Premium performance tier**
- Zone-redundant storage (LRS):
- Geo-redundant storage (GRS)
- Read-access geo-redundant storage (RA-RGS)



Locally Redundant Storage

- Copies:
  - **3 in the same Azure facility**
- Strategy:
  - **Data replicated synchronously across 3 replicas in the same Azure facility**
- Protection:
  - **Localized hardware failures**
- Additional considerations:
  - **LRS is an economical option if your data can be easily reconstructed**
  - **LRS is the only available option if you use the Premium performance tier**
  - **When using LRS, you should plan for an alternative recovery strategy**

Geo-Redundant Storage

- Copies:
  - **Total of 6, with 3 per Azure region.**
- Strategy:
  - **Data replicated synchronously across 3 replicas in the same Azure facility**
  - **Data replicated synchronously within each region and asynchronously across regions.**
- Protection:
  - **Localized hardware failures and region-wide disasters**
- Additional considerations:
  - **With GRS, data in the secondary region becomes available for reads and writes (via the primary endpoint) only after Microsoft initiates a failover from the primary region.**
  - **With RA-GRS, data in the secondary region is always available for reads (via the secondary endpoint). It becomes available for writes (via the primary endpoint) only after Microsoft initiates a failover from the primary region.**

Zone Redundant Storage

- Copies:
  - **3 across different Azure facilities in separate zones in the same region.**
- Strategy:
  - **Data replicated synchronously across 3 replicas in separate zones in the same region.**
- Protection:
  - **Localized hardware failures and failures of individual facilities**
- Additional considerations:
  - **ZRS is not yet available in all regions.**
  - **ZRS may not protect your data against a regional disaster where multiple zones are permanently affected.**
  - **ZRS does not support Azure VM disk files**

Replication Option Comparison

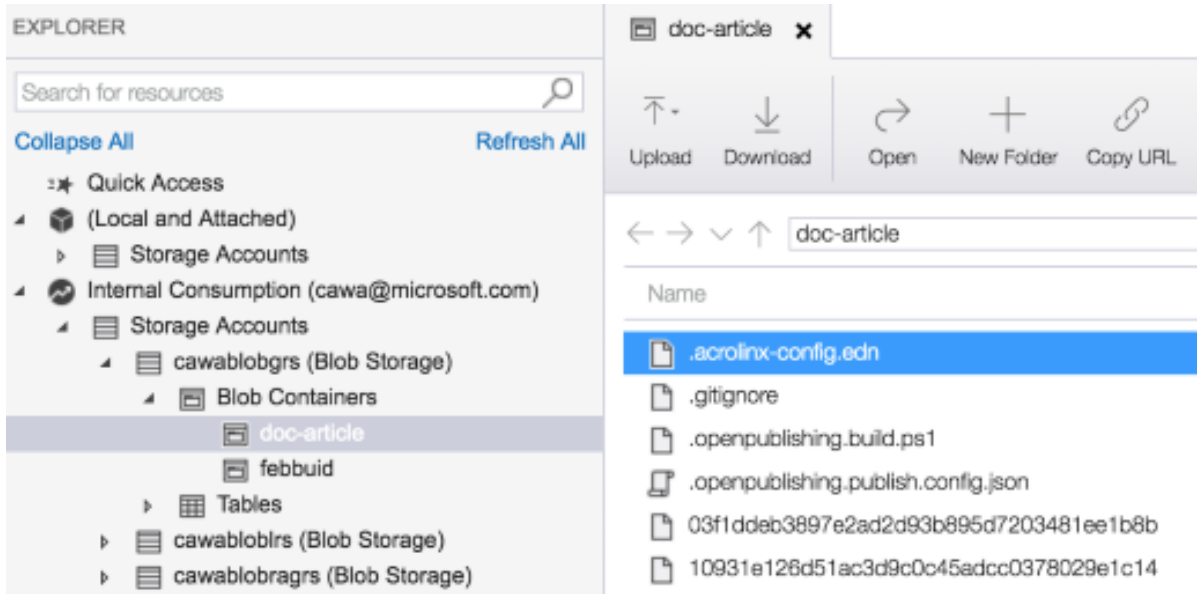
Replication Option	LRS	ZRS	GRS	RA-GRS
Node unavailability within a data center	Yes	Yes	Yes	Yes
An entire data center (zonal or non-zonal) becomes unavailable	No	Yes	Yes	Yes
A region-wide outage	No	No	Yes	Yes
Read access to your data (in a remote, geo-replicated region) in the event of region-wide unavailability	No	No	No	Yes
Available in storage account types	GPv1, GPv2, Blob	Standard,GPv2	GPv1, GPv2, Blob	GPv1, GPv2, Blob

Storage Account PowerShell Tasks

Task	Example
Check to see if a storage account name is available.	**Get-AzureRmStorageAccountNameAvailability** -Name 'mystorageaccount'
Create a storage account.	**New-AzureRmStorageAccount** -ResourceGroupName MyResourceGroup -AccountName mystorageaccount -Location westus -SkuName Standard_GRS
Retrieve a specific storage account or all the storage accounts in a resource group or subscription.	**Get-AzureRmStorageAccount** -ResourceGroupName "RG01" -AccountName "mystorageaccount"
Modify storage account properties, such as type.	**Set-AzureRmStorageAccount** -ResourceGroupName "MyResourceGroup" -AccountName "mystorageaccount" -Type "Standard_RAGRS"

Azure Storage Explorer

- An open-source Microsoft app for managing Azure storage, including:
  - **Viewing and editing Blob, Queue, Table, File, Cosmos DB storage and Data Lake Storage.**
  - **Creating, deleting, viewing, and editing storage objects.**
  - **Generating shared access signature (SAS) keys.**
  - **Accessing multiple subscriptions.**
- Available for:
  - **Windows**
  - **macOS**
  - **Linux**



Storage Explorer Functionality

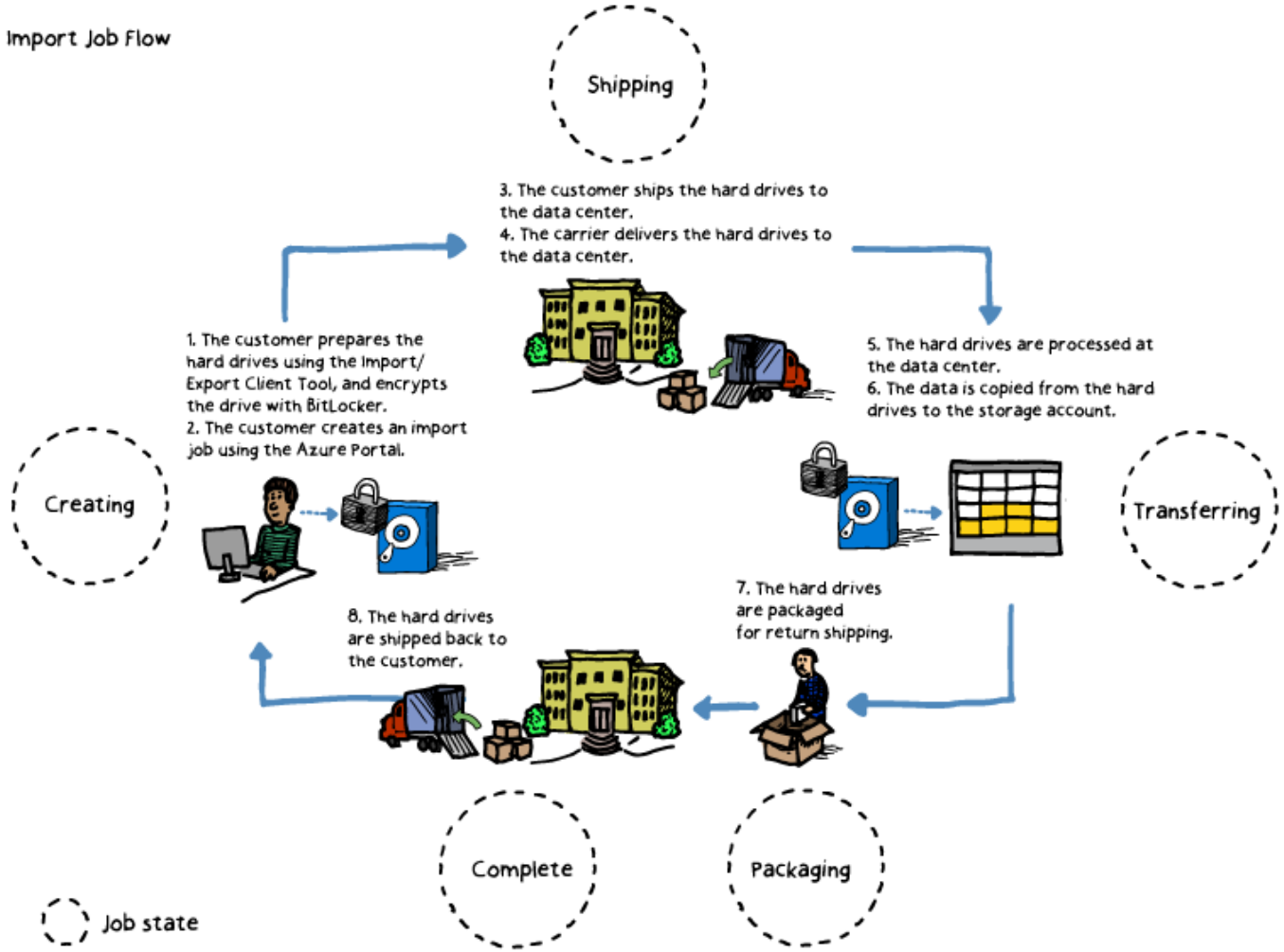
- **Connect to an Azure subscription:**
  - **Manage storage resources that belong to your Azure subscription.**
  - **Manage storage resources that belong to another Azure subscription**
  - **Manage storage resources in national Azure clouds**
- **Work with local development storage:**
  - **Manage local storage by using the Azure Storage Emulator.**
- **Connect to storage accounts by using the storage account's name and key.**
- **Connect to storage resources by using a shared access signature (SAS).**
- **Connect to a specific storage service (blob container, queue, or table) by using a SAS.**
- **Connect to an Azure Cosmos DB account by using a connection string.**

Import and Export Service

- Transfers very large amounts of data between on-premises and Azure:
  - **Import. Securely transfers data to Azure Blob storage (block and page blobs) and Azure Files. You ship hard drives containing your data to an Azure data center.**
  - **Export. Securely transfers data from Azure Blob storage (block, page, and append blobs) to your location. You ship empty hard drives to an Azure data center.**
- Scenarios:
  - **Migrating data to the cloud**
  - **Content distribution**
  - **Backup and data recovery**

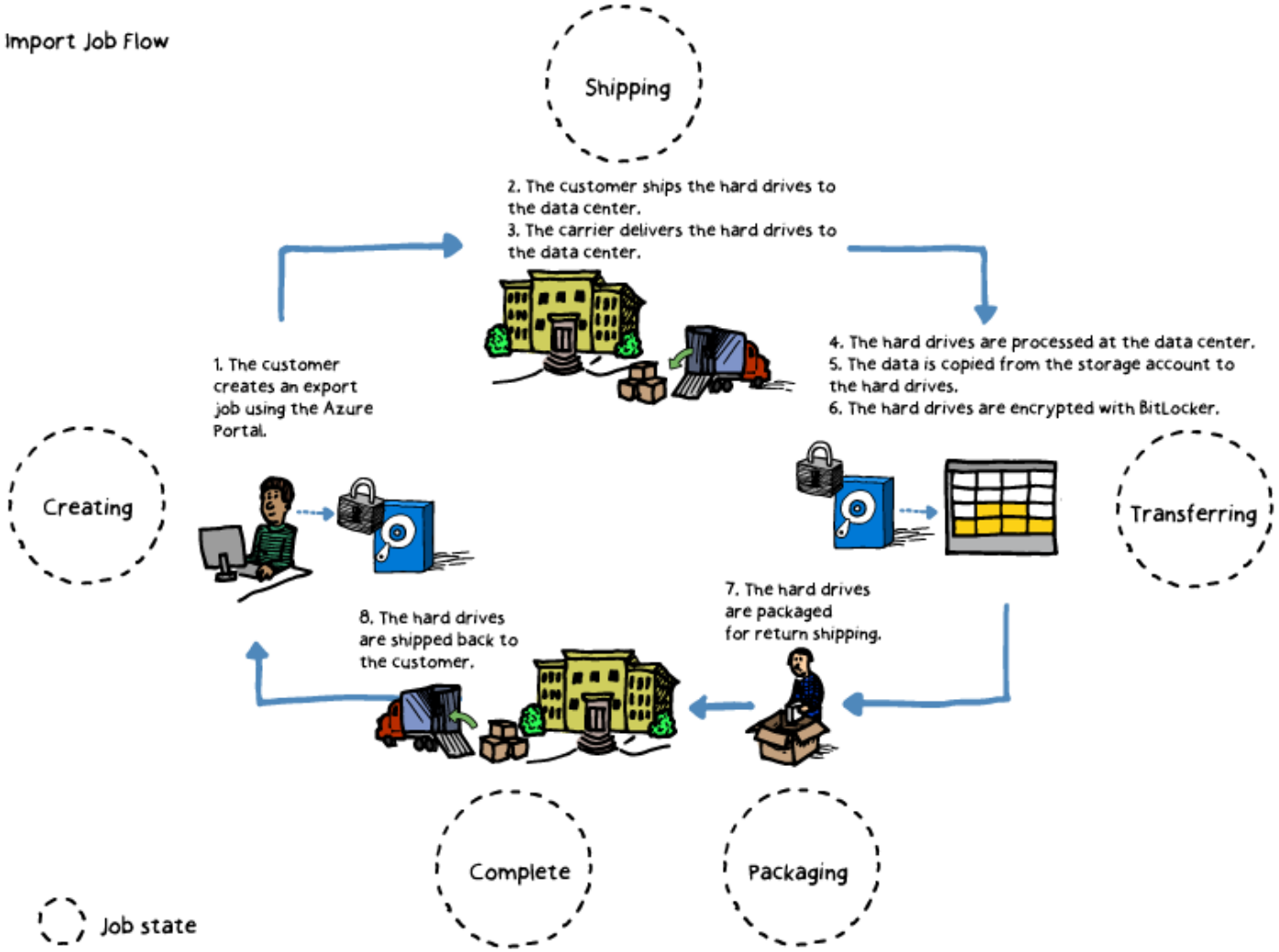
Import Jobs

- **Identify:**
  - **data to be imported**
  - **the number of drives**
  - **destination blob or file**
- **Run WAImportExport Tool:**
  - **To copy data to hard disk drives**
  - **To encrypt data with BitLocker**
- **Create an import job:**
  - **Via Azure portal (upload drive journal)**
  - **Via Import/Export REST API**
- **Provide shipping info**
- **Ship the hard disk drives**
- **Update tracking number**
- **Microsoft handles:**
  - **Job processing**
  - **Shipment of drives back to you**



Export Jobs

- **Identify:**
  - **data to be exported**
  - **the number of drives**
  - **the source blobs**
- **Create an export job:**
  - **via the Azure portal**
  - **via Import/Export REST API**
- **Provide shipping info**
- **Ship empty hard drives**
- **Update delivery tracking info**
- **Microsoft handles:**
  - **job processing**
  - **shipment of drives back to you**



AzCopy

- a command-line utility for automating Azure Storage tasks:
  - **Copy to/from Microsoft Azure Blob, File, and Table storage**
  - **Copy between a file system and a storage account, or between storage accounts**

Administrator: Windows PowerShell

```
C:\Program Files (x86)\Microsoft SDKs\Azure\AzCopy>azcopy /? | more
------------------------------------------------------------------------------
AzCopy 5.0.0 Copyright (c) 2015 Microsoft Corp. All Rights Reserved.
------------------------------------------------------------------------------

AzCopy </Source:> </Dest:> [/SourceKey:] [/DestKey:] [/SourceSAS:] [/DestSAS:]
       [/V:] [/Z:] [/@:] [/Y] [/NC:] [/SourceType:] [/DestType:] [/S]
       [/Pattern:] [/CheckMD5] [/L] [/MT] [/XN] [/XO] [/A] [/IA] [/XA]
       [/SyncCopy] [/SetContentType] [/BlobType:] [/Delimiter:] [/Snapshot]
       [/PKRS:] [/SplitSize:] [/EntityOperation:] [/Manifest:]
       [/PayloadFormat:]
```

Administrator: Windows PowerShell

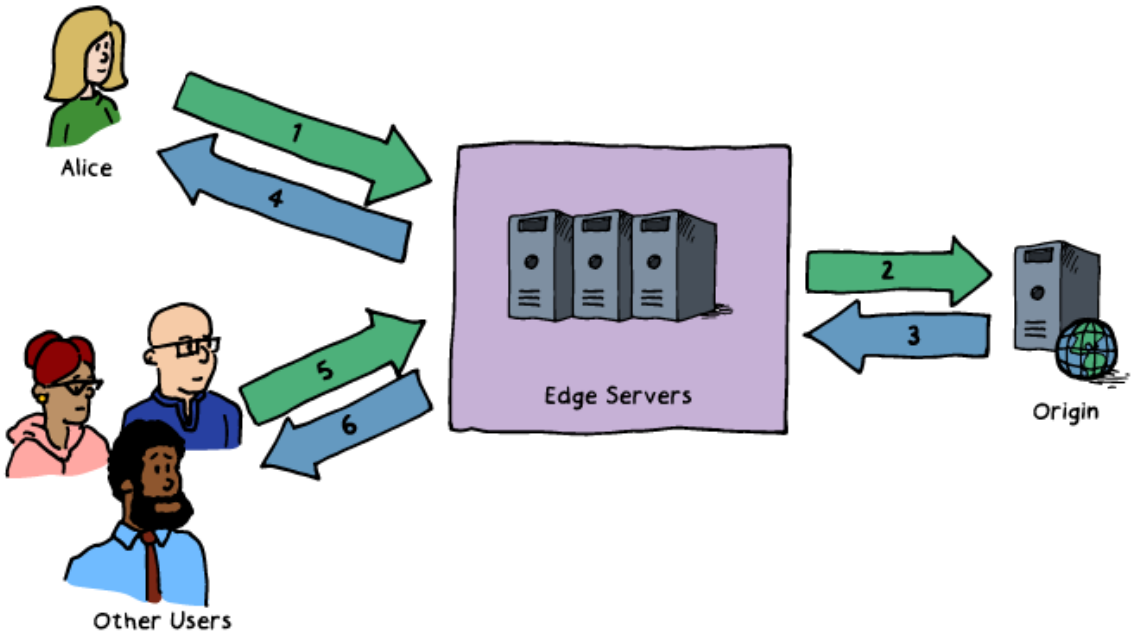
```
#1 - Download a blob from Blob storage to the file system, for example,
download 'https://myaccount.blob.core.windows.net/mycontainer/abc.txt'
to 'D:\test\'
    AzCopy /Source:https://myaccount.blob.core.windows.net/mycontainer/
    /Dest:D:\test\ /SourceKey:key /Pattern:"abc.txt"

#2 - Copy a blob within a storage count
    AzCopy /Source:https://myaccount.blob.core.windows.net/mycontainer1/
    /Dest:https://myaccount.blob.core.windows.net/mycontainer2/
    /SourceKey:key /DestKey:key /Pattern:"abc.txt"

#3 - Upload files and subfolders in a directory to a container, recursively
    AzCopy /Source:D:\test\
    /Dest:https://myaccount.blob.core.windows.net/mycontainer/
    /DestKey:key /S
```

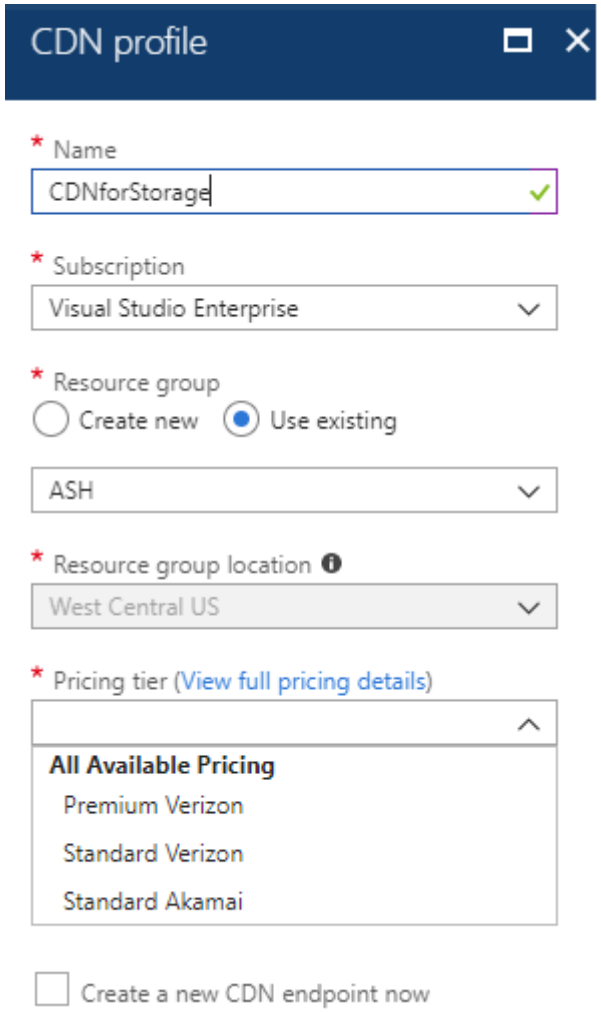
How CDN Works

- Azure content caching service:
  - **Designed to expedite delivery of audio, video, images, etc. hosted by Azure services**
  - **Utilizes the following data flow:**
    - **1. Alice requests a file using a URL pointing to a CDN location. DNS routes the request to the optimal POP**
    - **2. If edge servers do not have the file cached, they requests it from the origin.**
    - **3. The origin returns the file, including its TTL.**
    - **4. The server caches the file and returns it to Alice. The file remains cached until the TTL expires.**
    - **5. Subsequent requests are served from the cache.**



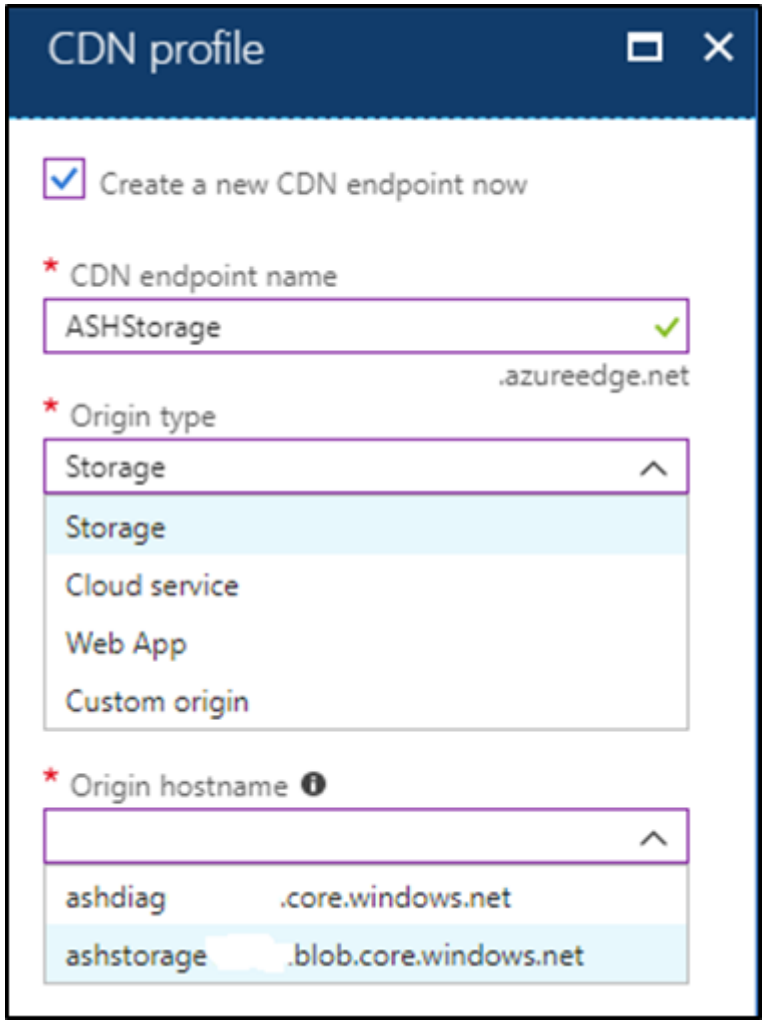
CDN Profile

- A collection of CDN endpoints with the same:
  - **Pricing tier and provider:**
    - **Premium Verizon**
    - **Standard Verizon**
    - **Standard Akamai**
- Pricing based on:
  - **GBs of outbound data transfers**
  - **Zone:**
    - **Zone 1—North America, Europe, Middle East and Africa**
    - **Zone 2—Asia Pacific (including Japan)**
    - **Zone 3—South America**
    - **Zone 4—Australia**
    - **Zone 5—India**



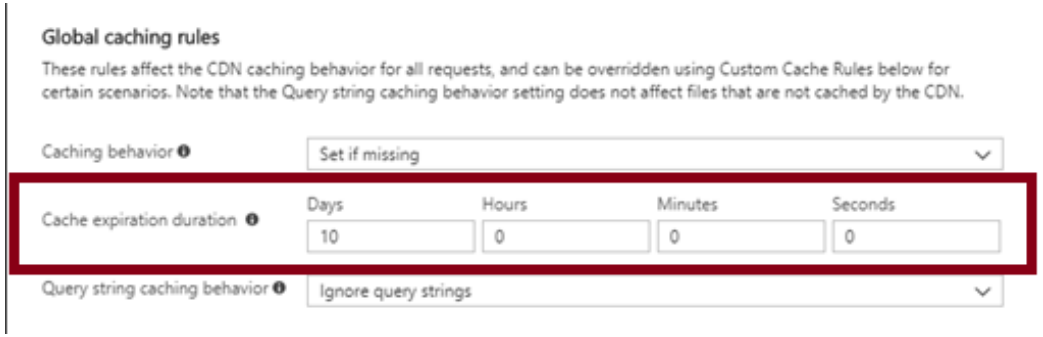
CDN Endpoints

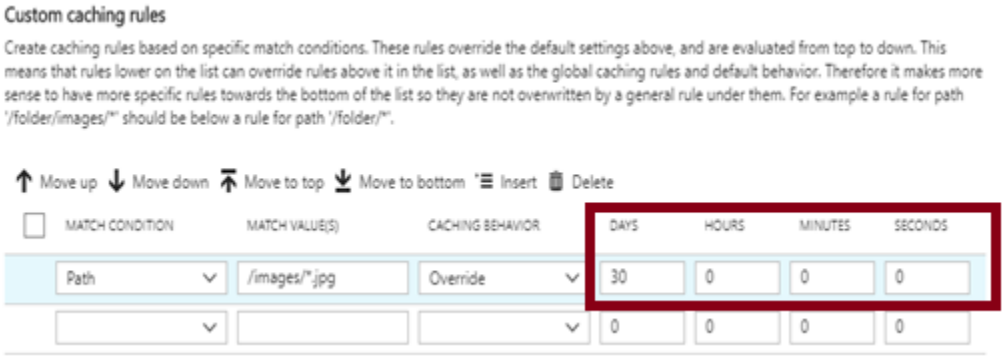
- Represent configuration of content origin:
  - **CDN endpoint name**
  - **Origin type:**
    - **Storage**
    - **Cloud Service**
    - **Web App**
    - **Custom origin**
  - **Origin hostname**
  - **CDN features that affect content delivery, such as:**
    - **Compression**
    - **Query string**
    - **Geo filtering**
    - **Custom domains**



CDN Time-to-Live

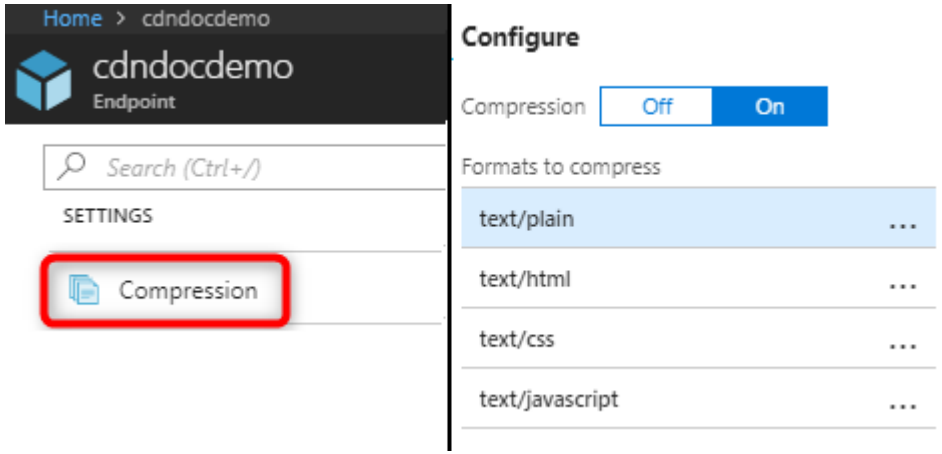
- CDN TTL can be:
  - **Included in cache-directive headers in the HTTP response from the content origin**
  - **Determined by Cache Expiration Duration, configurable by using:**
    - **Global caching rules: affect all requests to the endpoint**
    - **Custom caching rules: match paths and file extensions**
  - **Custom caching rules override global ones**





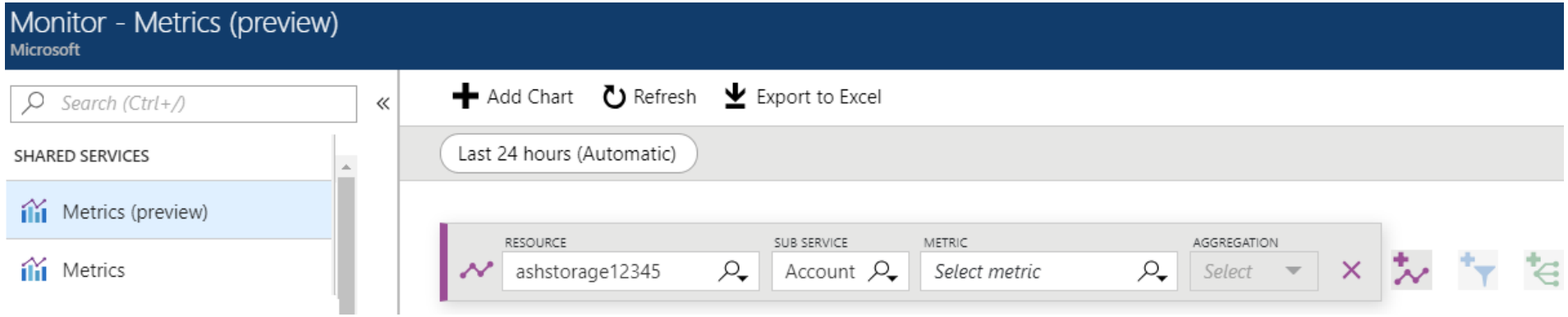
CDN Compression

- Improves file transfer speed and increases page-load performance:
- Supports:
  - **Enabling compression on the origin. In this case, the CDN passes along the compressed files and delivers them to clients that request them.**
  - **Enabling compression directly on the CDN edge servers. In this case, the CDN compresses the files and serves them to end users.**
  - **Enabling compression based on the MIME types list, allowing selection of the content formats to compress.**



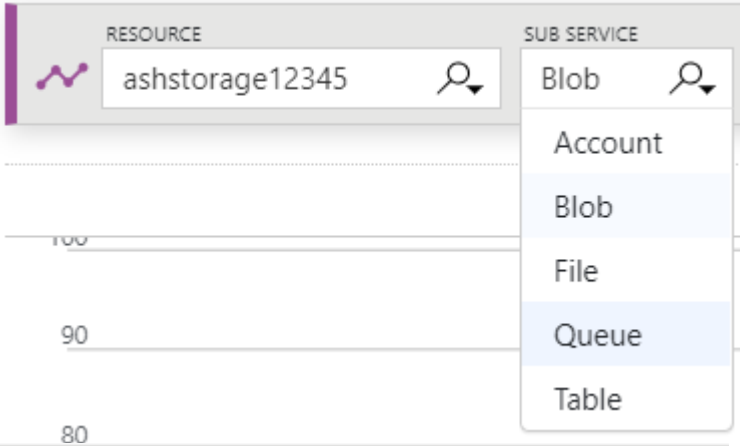
Monitor Metrics

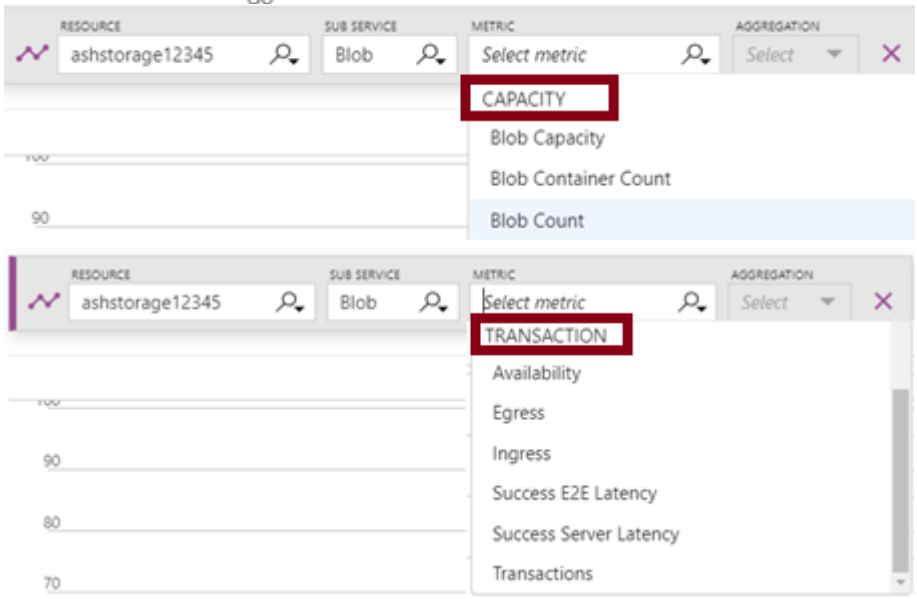
- Azure Monitor provides unified interface for monitoring Azure services
  - **Includes Azure Storage monitoring**
  - **Offers access via the Azure portal and Monitor API (REST and .NET)**
  - **Integrates with Log Analytics and Event Hubs**
  - **Stores metrics for 30 days and allows long term retention in Azure Storage**
  - **Supports consistent metrics filtering (resource, subservice) and aggregation criteria.**



Capacity and Transaction Metrics

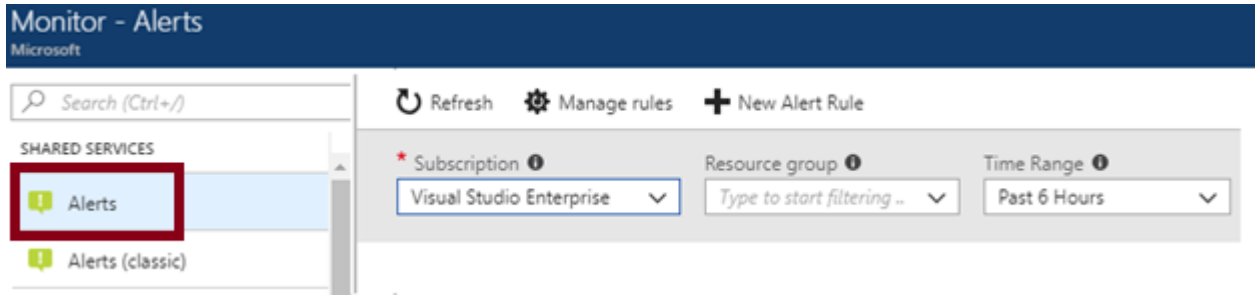
- Available from the Azure portal:
  - **Select a Storage Account resource**
  - **Select a sub-service: Account, Blob, File, Queue, or Table**
  - **The sub-service selection determines Capacity metrics:**
    - **For example, for Blob sub-service, Capacity metrics include:**
      - **Blob Capacity**
      - **Blob Container Count**
      - **Blob Count**
  - **Transaction metrics are available at:**
    - **account level**
    - **service level:**
      - **Blob storage**
      - **Table storage**
      - **Queue storage**
      - **Azure Files**





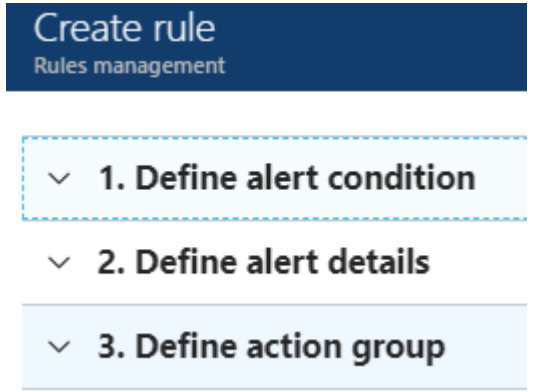
Azure Monitor Alerts

- New experience – comparing with Alerts (Classic):
  - **Better notification system**
  - **A unified authoring experience**
  - **Access to Log Analytics alerts from the Azure portal**
  - **Separation of Fired Alerts and Alert Rules**
  - **Better workflow**



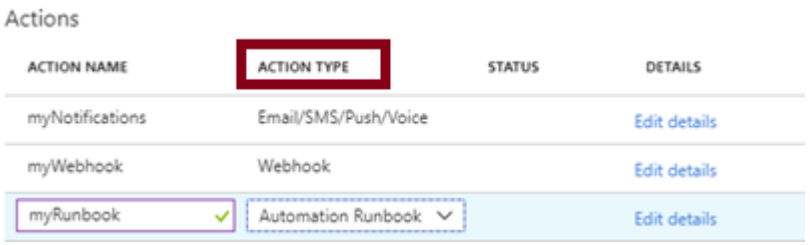
Alert Rules

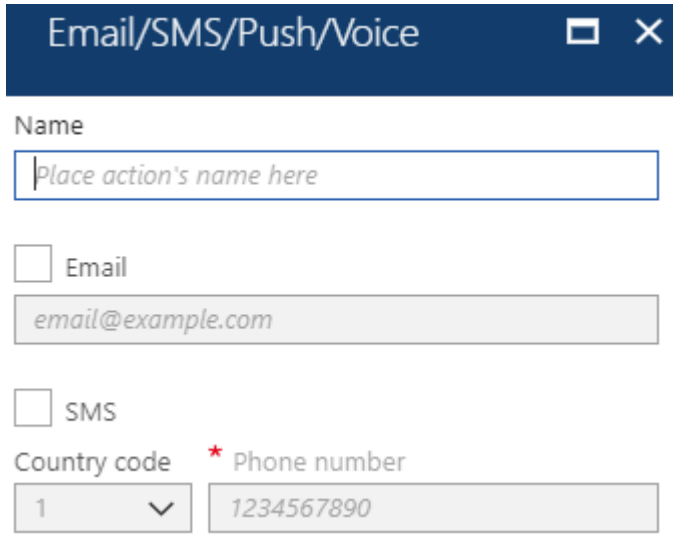
- Part of the new experience:
  - **alerts authoring is consistent regardless of the monitoring service or signal type**
  - **all alerts fired and related details are available on a single page**
- Authoring is a three-step process:
  - **1. Define alert condition, including:**
    - **Target selection. For example, a storage account.**
    - **Alert criteria. For example, Used Capacity.**
    - **Alert logic. For example, over a six-hour period whenever the Used Capacity is over 1000000 bytes.**
  - **2. Define alert details, including:**
    - **Alert rule name, description, and severity. There are five severity levels, Severity 0 to Severity 4.**
  - **3. Define an action group to notify your team via email and text messages or automate actions using webhooks and runbooks.**



Action Groups

- Action groups facilitate configuring actions in response to an alert:
  - **Email/SMS/Push/Voice**
  - **Logic App**
  - **Webhook**
  - **Automation Runbook**
  - **IT Service Management**





Signal Types and Metrics

- Signals are emitted by target resources and include:
  - **Metrics**
  - **Activity log**
  - **Application Insights**
  - **Logs**
- The latest metrics alerts offer:
  - **Improved latency**
  - **Support for multi-dimensional metrics**
  - **More control over metrics conditions**
  - **Combined monitoring of multiple metrics**
  - **Metrics from Logs (limited public preview)**

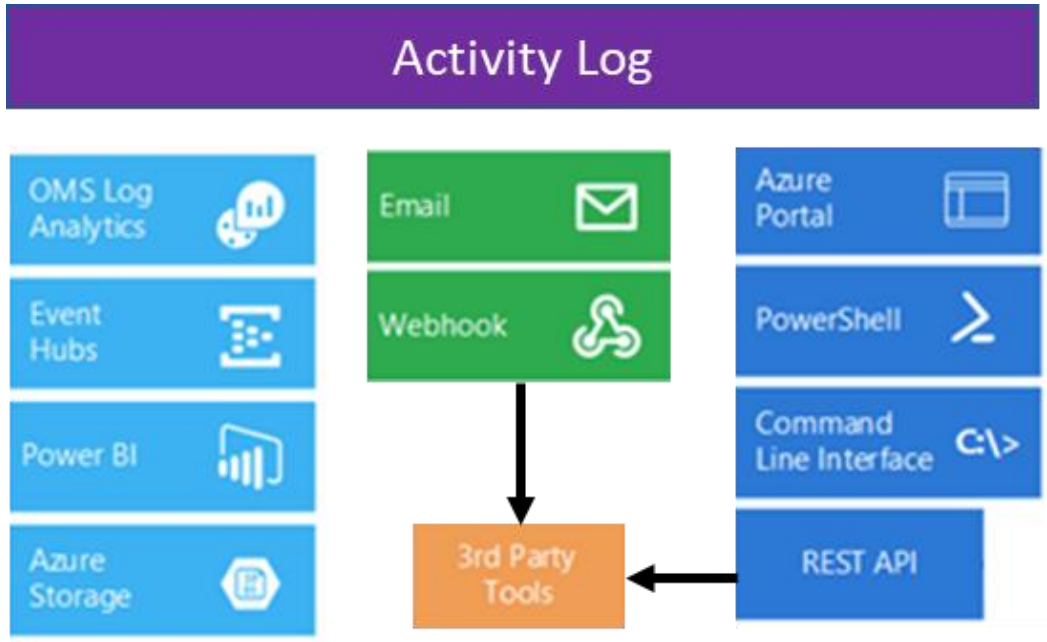
Configure signal logic

Define your alert criteria by choosing a signal below and defining your alert condition on the next screen.

SIGNAL NAME	SIGNAL TYPE	MONITOR SERVICE
Used capacity	Metric	Platform
Transactions	Metric	Platform
All Administrative operations	Activity Log	Administrative
List Storage Account Keys (storageAccounts)	Activity Log	Administrative
Regenerate Storage Account Keys (storageAcc...	Activity Log	Administrative
Delete Storage Account (storageAccounts)	Activity Log	Administrative

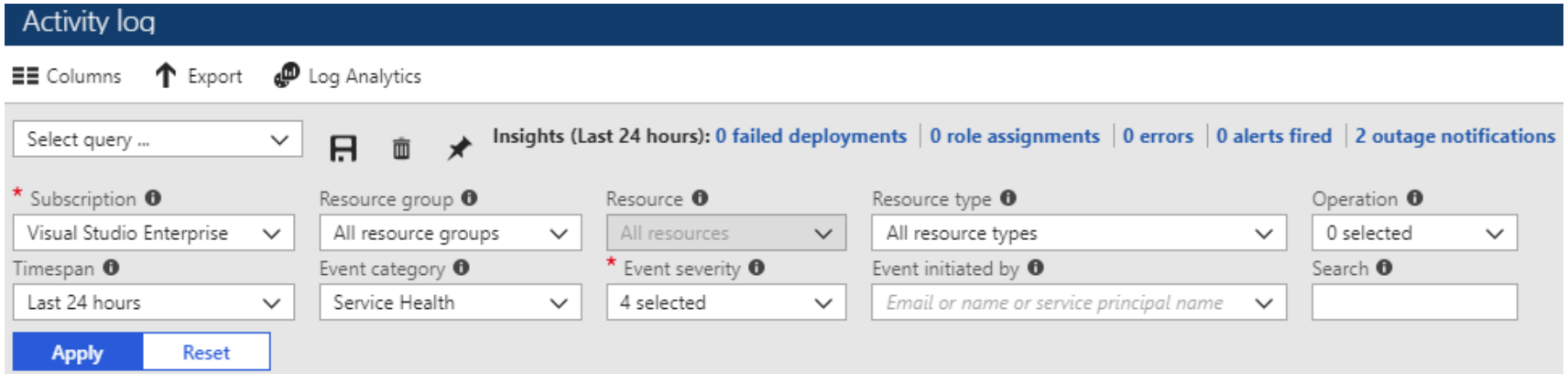
Activity Log

- A subscription-level log that captures:
  - **Azure Resource Manager operational data (control plane)**
  - **Service Health events**
- Includes audit trail of all write/change operations:
  - **Helps determine "what, who, and when"**
  - **Provides the status of each operation**
- Its content can be:
  - **Streamed to Event Hub.**
  - **Analyzed with Power BI.**
  - **Viewed in the Azure portal.**
  - **Forwarded to Log Analytics.**
  - **Archived in an Azure Storage account.**
  - **Processed with PowerShell, CLI, and REST API.**



Query the Activity Log

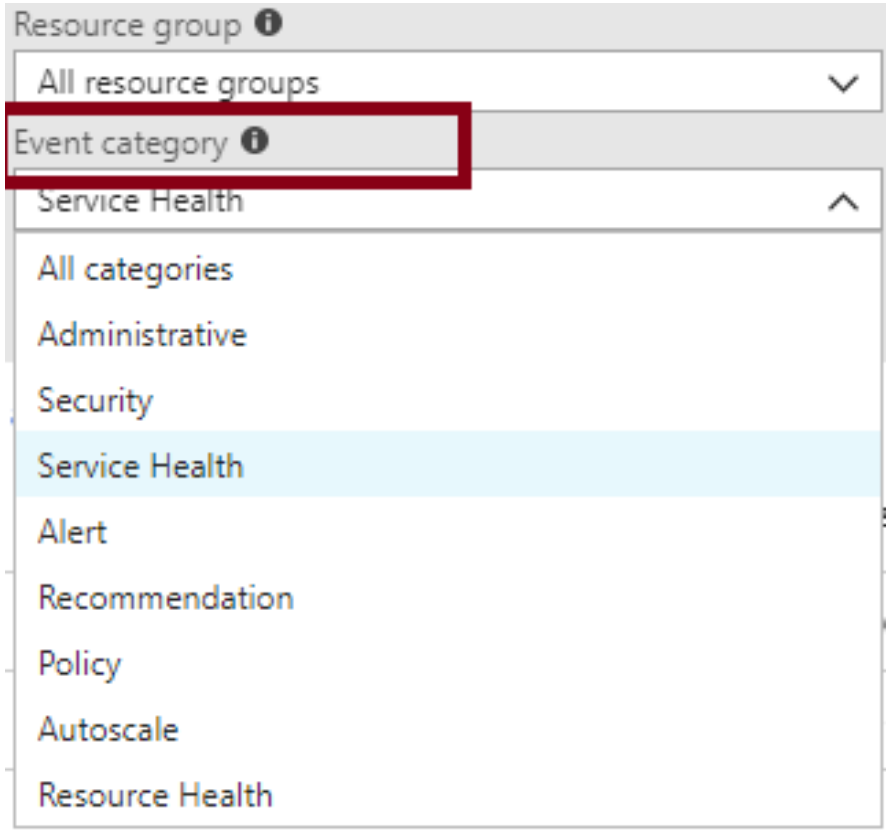
- Available directly from the Azure portal, with filtering based on:
  - **Subscription**
  - **Resource group**
  - **Resource (name)**
  - **Resource type**
  - **Operation (type)**
  - **Timespan**
  - **Event category**
  - **Event severity**
  - **Event initiated by**



- Filter selection can be saved
- Queries can be pinned to the Azure portal dashboard

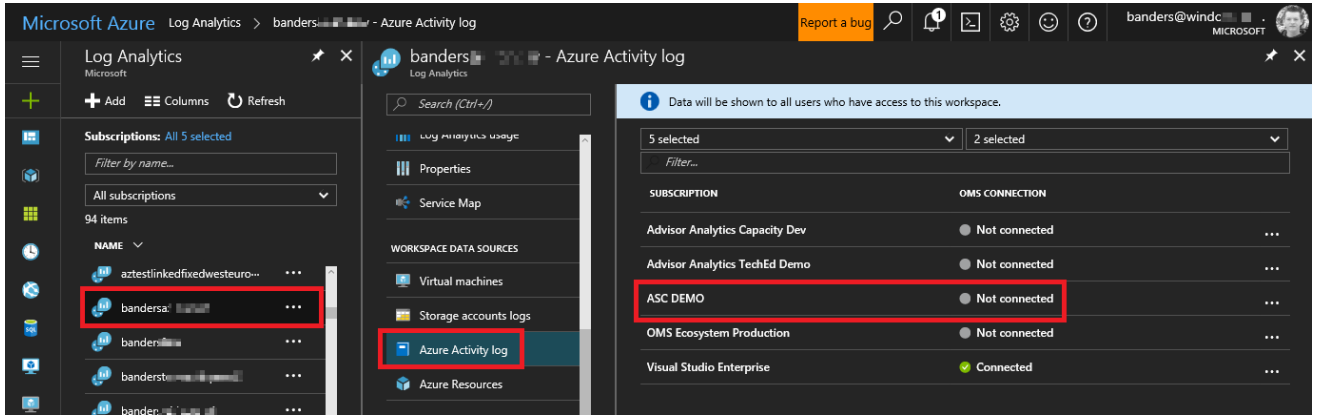
Event Categories

- Administrative
- Security
- Service Health
- Alert
- Recommendation
- Policy
- Autoscale
- Resource Health



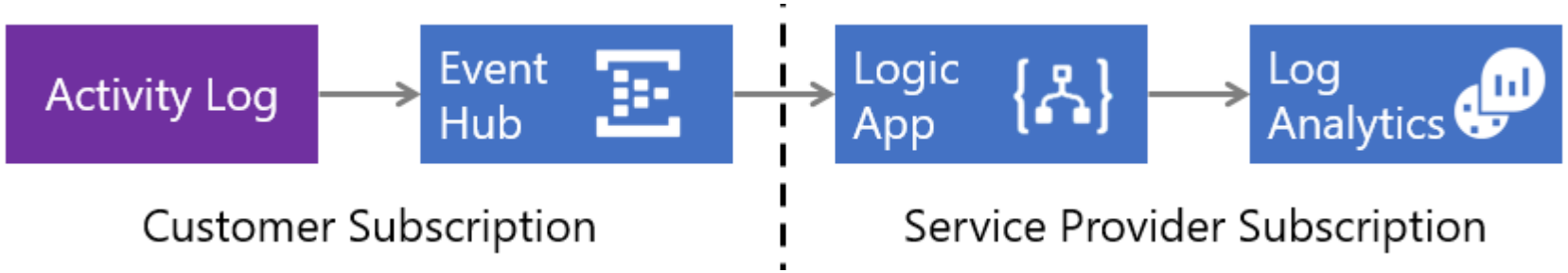
Activity Log and Log Analytics

- Activity log can be added to Azure Log Analytics workspace:
  - **Can be in the same or a different subscription (if it has the same Azure AD tenant)**
  - **Provides the ability to:**
    - **Correlate user activities, auto-scale operations, authorization changes, and service health to other logs or metrics from your environment.**
    - **Correlate activity logs with other Azure services and application data.**
    - **Analyze and search activity logs from multiple Azure subscriptions.**
    - **Keep activity logs for longer than the default of 90 days.**
    - **View operational activities aggregated by status.**
    - **Analyze activity logs with pre-defined views.**
    - **Identify outage or service health issues.**
    - **Report on authorization changes.**
    - **View trends of activities.**



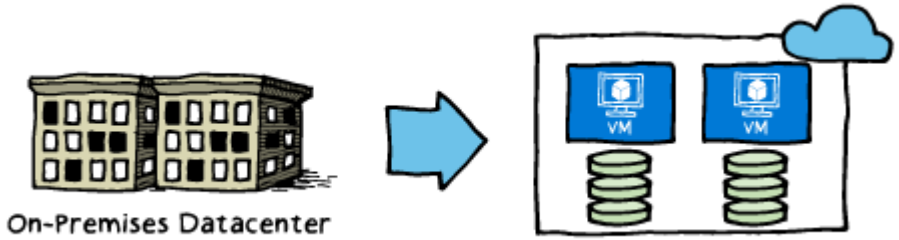
Collect Across Subscriptions

- You can collect Azure Activity Logs into a Log Analytics workspace using the Azure Log Analytics Data Collector connector for Logic Apps
  - **This facilitates using a workspace in a subscription that has a different Azure AD tenant**
  - **It offers a number of advantages, including:**
    - **Low latency, since the Azure Activity Log is streamed into the Event Hub**
    - **Serverless services, with no infrastructure components to deploy**
    - **Minimal custom code**



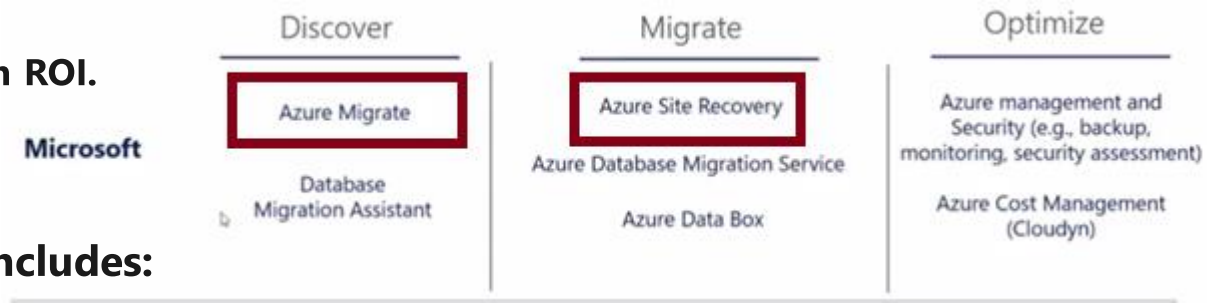
Migration Goals

- Technology-focused and business-focused, including:
  - **Addressing the hardware obsolescence cycle**
  - **Moving away from the ‘pre-purchase capacity’ model**
  - **Lack of IT agility**
  - **Desire to re-focus on core competencies**
  - **Expense of maintaining a global presence**
  - **Enable disaster-recovery scenarios**



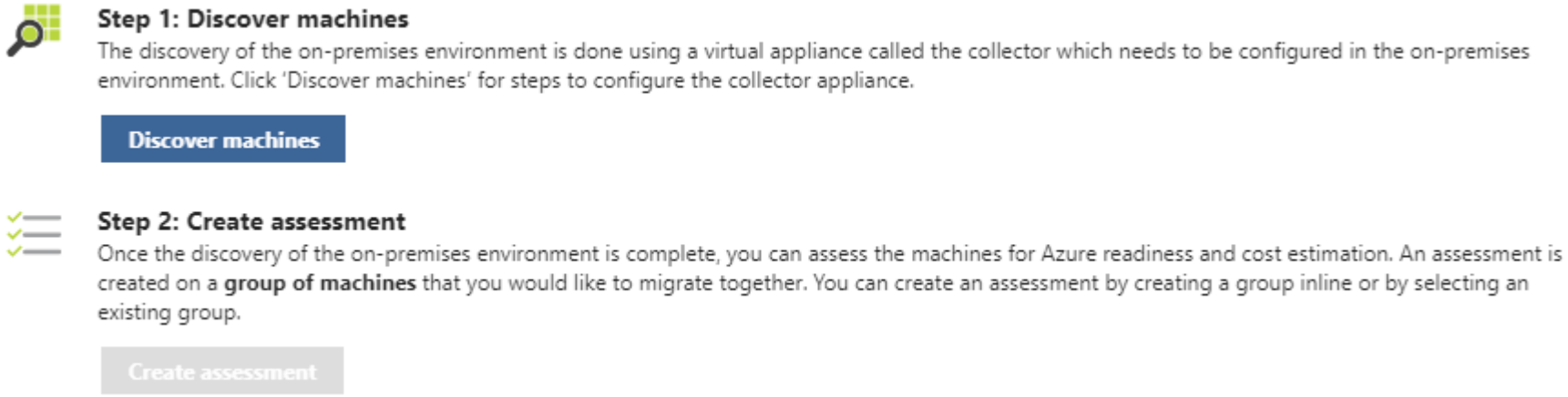
Migration Phases

- When planning for migration to Azure, consider the following phases:
  - **Discover: gain better visibility of on-premises workloads and assess the optimal resource level to run them in Microsoft Azure.**
    - **Azure Migrate is the primary tool for this, and includes:**
      - **Automated server, app, and database discovery.**
      - **Intelligent workload right-sizing and costing for maximum ROI.**
      - **Workload configuration analyses and recommendations.**
  - **Migrate: move selected workloads to Azure.**
    - **Azure Site Recovery is the primary tool for this and includes:**
      - **Lifting and shifting of servers, apps, databases, and data.**
      - **Containerization of existing applications and infrastructure**
      - **Modernization options for apps and databases.**
  - **Optimize: fine tune your Azure-based workloads and maximize your ROI.**
    - **There are many Microsoft partners to help you with backup, monitoring, security assessments, and cost management.**



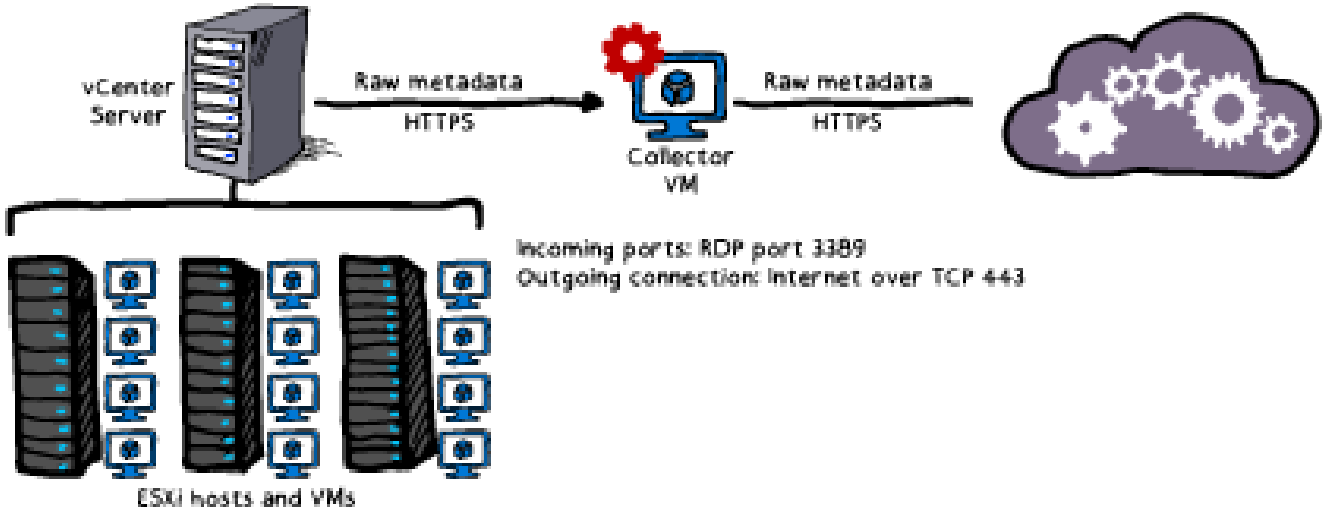
Azure Migration Service

- Provides assessment of on-premises workloads for migration to Azure:
  - **migration suitability of on-premises machines**
  - **performance-based sizing**
  - **cost estimates for running your on-premises machines on Azure VMs**
- Follows two steps:
  - **Discover machines**
  - **Create Assessments**



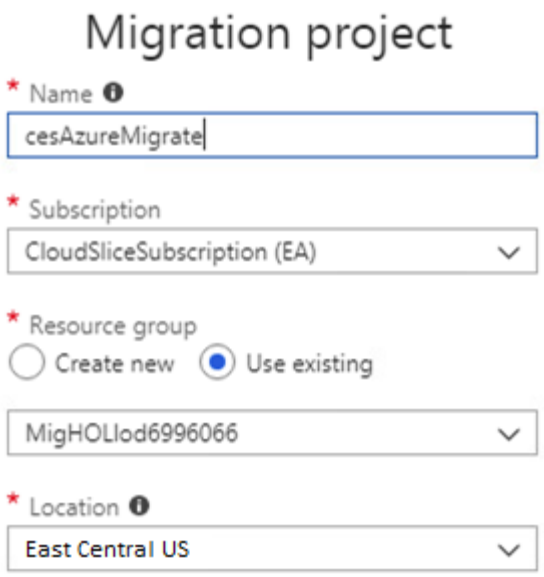
Using Azure Migrate: A Look ahead

- Architecture:
  - **Closely integrated with the components of an VMware vCenter environment**
- Process
  - **Create a project**
  - **Discover the machines**
  - **Collect the information**
  - **Assess the project**



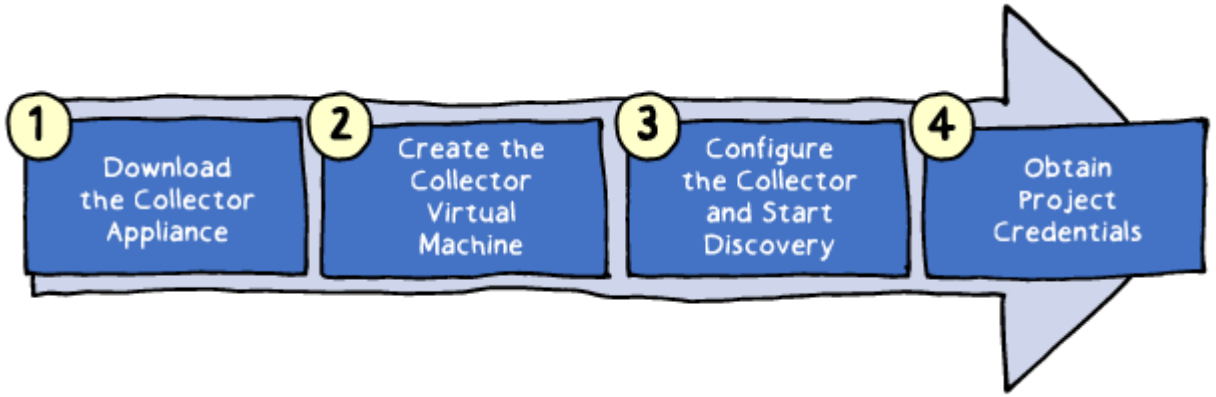
Creating a Project

- The initial step of the Azure Migrate
- A project contains metadata representing on-premises environment:
  - **Name**
  - **Subscription**
  - **Resource Group**
  - **Location**
  - **Discovery records of VMware VMs**
- Project limits:
  - **Up to 1,500 discovered VMs per project**
  - **Up to 1,500 assessed VMs per project**
  - **Up to 20 projects per subscription**



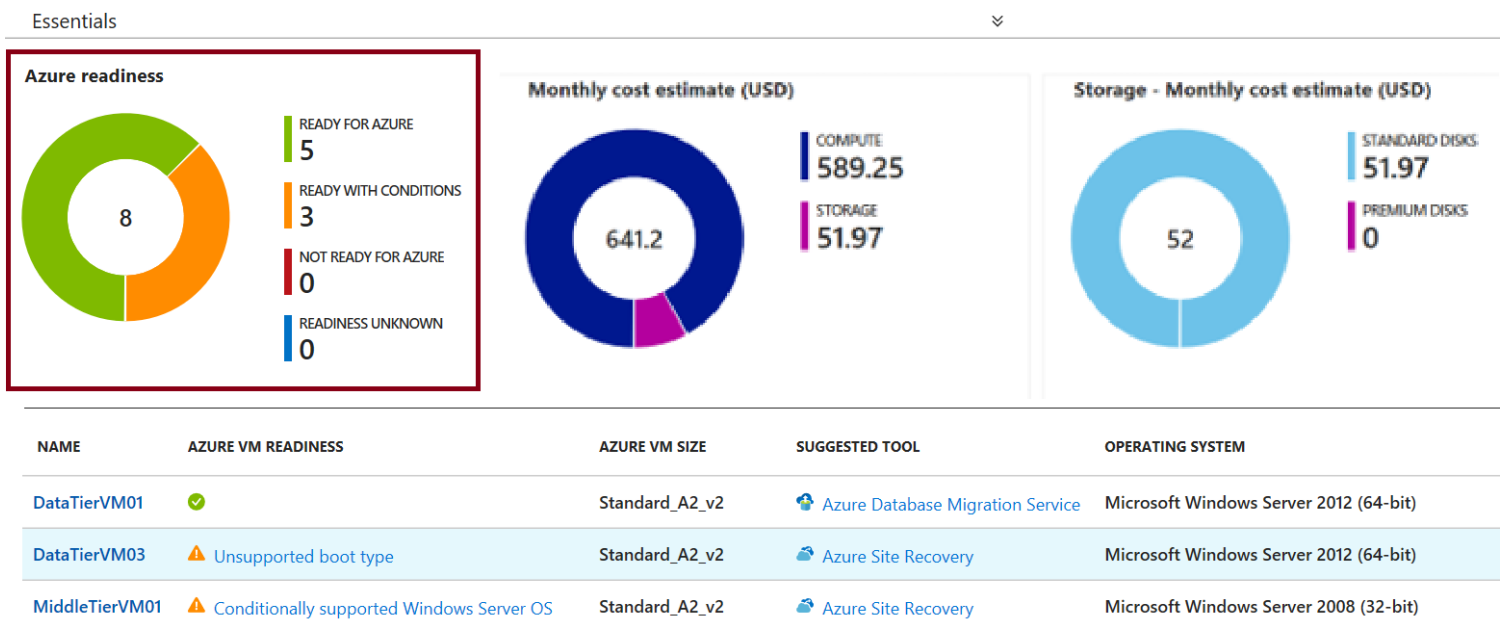
Creating a Collector

- A virtual appliance which handles discovery
- Implementation consists of four main steps:
  - **Downloading the Collector appliance (an Open Virtualization Appliance (.ova) file downloadable from the Azure Migrate project in the Azure portal)**
  - **Creating the Collector virtual machine by importing the .ova file on the vCenter server**
  - **Configuring the Collector to initiate discovery**
  - **Assigning to the Collector project credentials including Azure Migrate project ID and key**



Assessing Readiness

- Assessment is based on the readiness status of discovered VMs:
  - **Ready for Azure (green): along with the recommended Azure VM size**
  - **Ready with conditions (Orange) and Not ready for Azure (Red): including readiness issues and remediation steps.**
  - **Readiness unknown (Blue)**

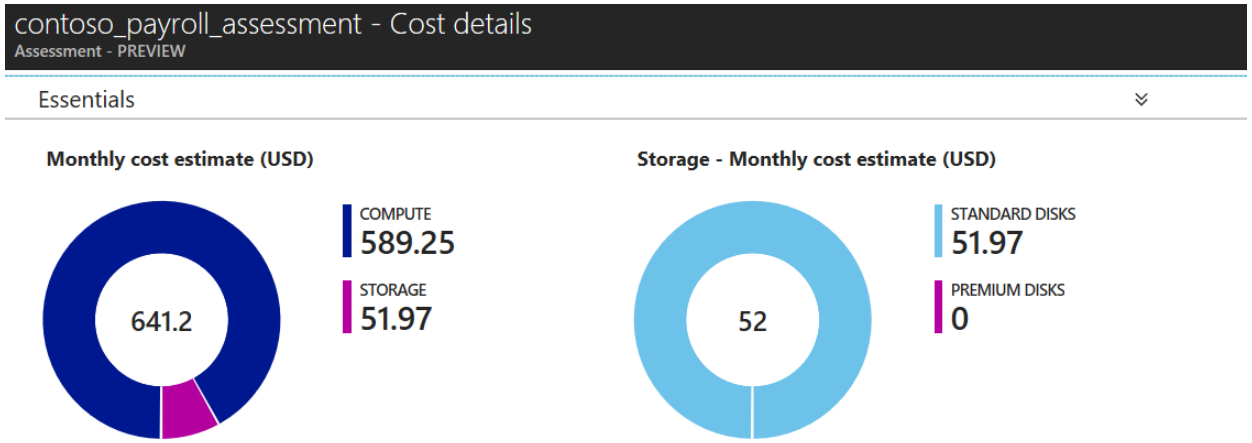


Assessing VM Sizing

- Azure Migrate VM assessment offers two types of sizing :
  - **Performance-based sizing (default) which takes into account:**
    - **Storage: maps the size and performance of VM disks to Azure VM disks**
    - **Network: identifies Azure VM sizes that offer matching number and performance of network adapters**
    - **Compute: determines CPU and memory requirements based on performance history of discovered VMs**
  - **On-premises sizing:**
    - **Matches the size of on-premises VM to an equivalent Azure VM**
    - **Does not take into account performance history of discovered VMs**

Estimating Cost

- Provides the total costs of Azure VMs:
  - **Compute: aggregated monthly cost, which takes into account:**
    - **OS type**
    - **Software Assurance**
    - **Reserved Instances**
    - **VM uptime**
    - **VM location**
    - **currency settings**
  - **Storage: aggregated monthly cost.**
    - **No offer specific settings**



NAME	AZURE VM SIZE	AZURE STORAGE	MONTHLY COMPUTE COST ESTIMATE (USD)	MONTHLY STORAGE COST ESTIMATE (USD)
DataTierVM01	Standard_A2_v2	80 GB (80 GB Standard, 0 GB Premium)	64.73	5.89
DataTierVM03	Standard_A2_v2	80 GB (80 GB Standard, 0 GB Premium)	64.73	5.89
MiddleTierVM01	Standard_A2_v2	80 GB (80 GB Standard, 0 GB Premium)	64.73	5.89

Customizing the Assessment

- Customization settings include:
  - **Performance history duration**
  - **Target location**
  - **Pricing tier**
  - **Storage type**
  - **Comfort factor**
  - **Currency**
  - **Discounts**
  - **VM uptime**

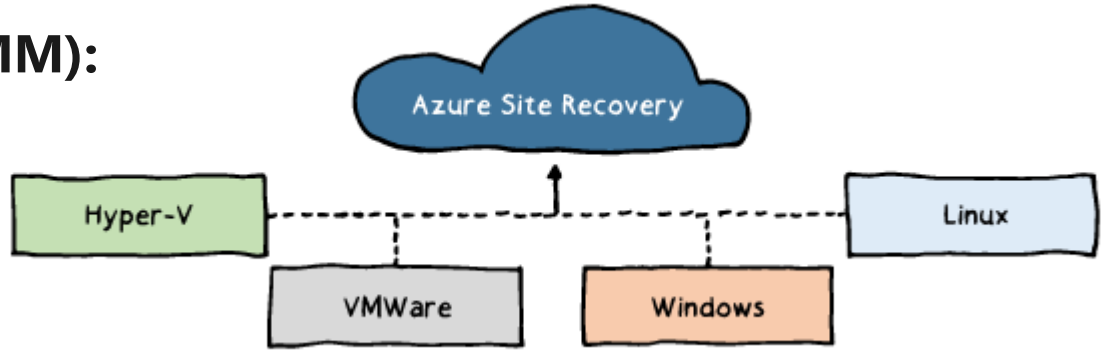
Setting	Details	Default
Target location	The Azure location to which you want to migrate. Azure Migrate currently supports 30 regions.	West US 2 is the default location.
Pricing tier	You can specify the pricing tier (Basic/Standard) for the target Azure VMs.	By default the Standard tier is used.
Storage type	You can specify the type of disks you want to allocate in Azure.	The default value is Premium managed disks.
Comfort factor	Azure Migrate considers a buffer (comfort factor) during assessment. This buffer is applied on top of machine utilization data for VMs.	Default setting is 1.3x.

Troubleshooting Azure Migrate

- The most common issues include:
  - **Migration project creation failed**
  - **No performance data**
  - **Collector is not able to connect to the internet**
  - **Date and time synchronization error**
  - **Error UnableToConnectToServer**

ASR Scenarios

- Hyper-V VM replication:
  - **Hyper-V with Virtual Machine Manager (VMM):**
    - **Replication to Azure or a secondary datacenter.**
  - **Hyper-V without VMM:**
    - **Replication to Azure only.**
- VMware VM replication:
  - **Replication to a secondary site running VMware or to Azure.**
- Physical Windows and Linux machines:
  - **Replication to a secondary site running VMware or to Azure.**

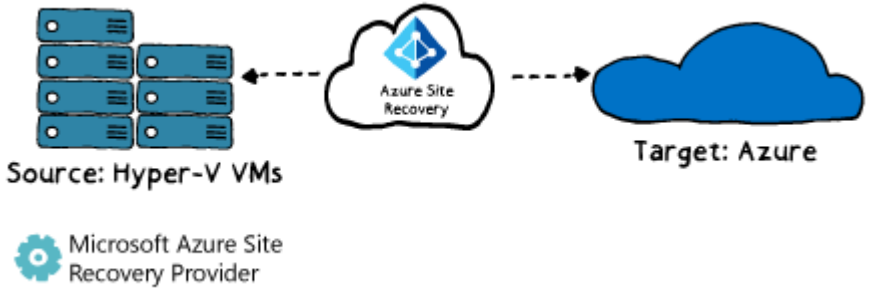


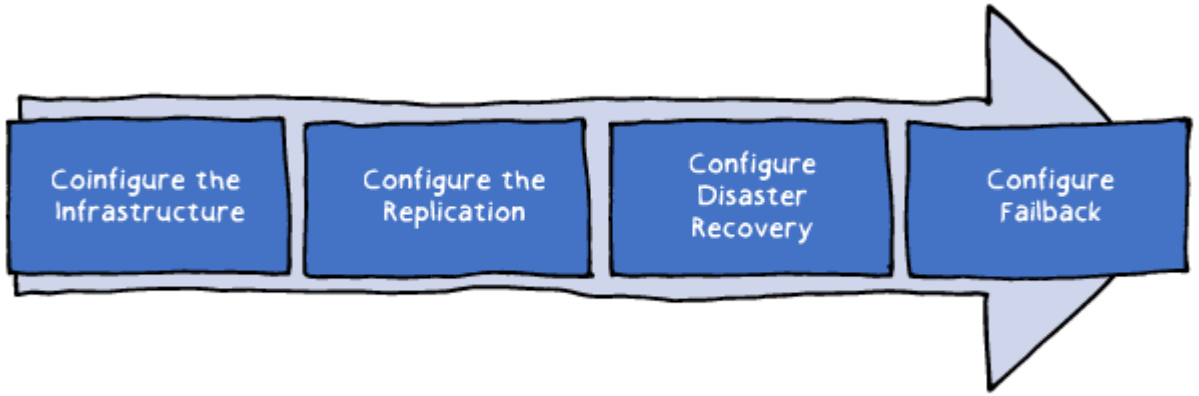
ASR Features

- Eliminating the need for disaster recovery sites
- Reducing infrastructure costs
- Automatically replicating to Azure
- Safeguarding against outages of complex workloads
- Extending or boosting capacity
- Continuous health monitoring

Using ASR: A Look Ahead

- Architecture:
  - **The primary and a secondary site**
  - **On-premises or in Azure**
- Process:
  - **Configure the Infrastructure**
  - **Configure the Replication**
  - **Configure Disaster Recovery**
  - **Configure Failback**



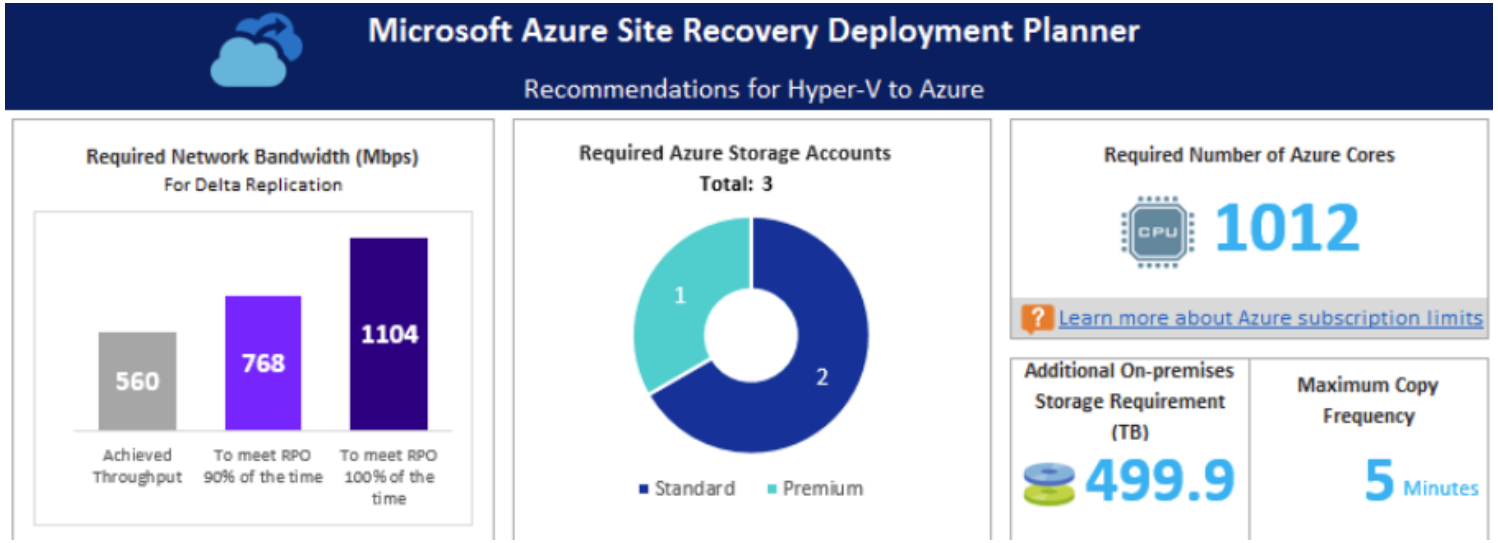


Configuring Azure

- Delegate permissions within the target Azure subscription:
  - **Virtual Machine Contributor built-in role**
  - **Site Recovery Contributor built-in role**
- Create an Azure Storage account:
  - **General Purpose**
  - **Read-access geo-redundant**
  - **Secure transfer: Disabled**
- Create an Azure virtual network:
  - **Ensure sufficient bandwidth for intersite replication**

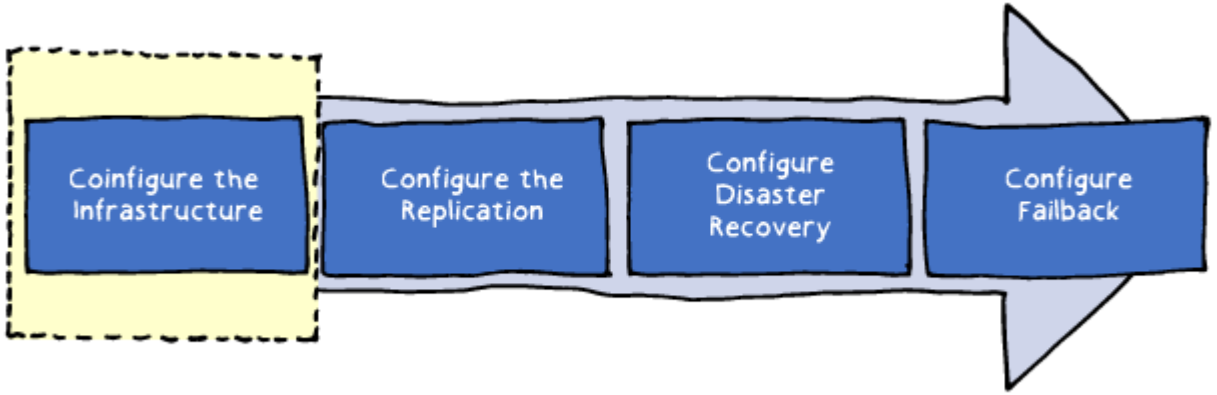
ASR Deployment Planner

- A command line tool for profiling Hyper-V VMs to estimate:
  - **Intersite network bandwidth requirements**
  - **Azure Storage requirements**
- Generates a report including:
  - **On-premises summary**
  - **Recommendations**
  - **VM storage placement**
  - **Compatible VMs**
  - **Incompatible VMs**
  - **Storage requirement**
  - **Initial Replication batching**
  - **Cost estimate**



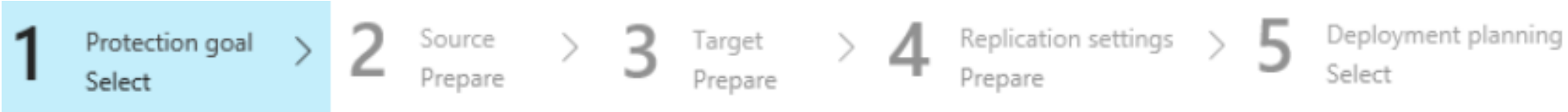
Configure the Infrastructure: Requirements

- **Verify Hyper-V operating system requirements:**
  - **Windows Server 2016 (including server core installation)**
  - **Windows Server 2012 R2 with latest updates.**
  - **Mixing hosts running Windows Server 2016 and 2012 R2 isn't supported.**
- **Verify Hyper-V host and guest storage requirements:**
  - **Hyper-V host storage can include SMB 3.0, SAN (iSCSI), and Multi-path (MPIO).**
  - **No support for shared cluster disks and encrypted disks.**
- **Verify internet access:**
  - **Hyper-V hosts should have direct access to the internet without using a proxy.**
- **Prepare Windows VMs for access following a failover (optional)**



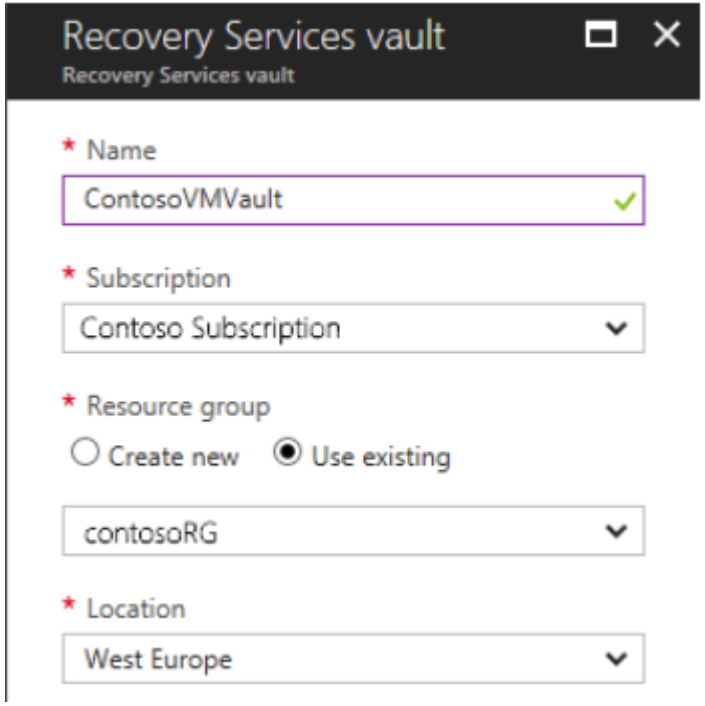
Configure the Infrastructure: Process

- The Azure portal driven sequence:
  - **1. Protection goal (e.g. replicating Hyper-V machines to Azure)**
  - **2. Source environment (e.g. Hyper-V site)**
  - **3. Target environment (e.g. a resource group in an Azure subscription)**
  - **4. Replication settings**
  - **5. Deployment planning**



Recovery Services Vault

- Hosts ASR metadata and replicated VM disks
- Can be created directly from the Azure portal
- Serves as the primary configuration interface



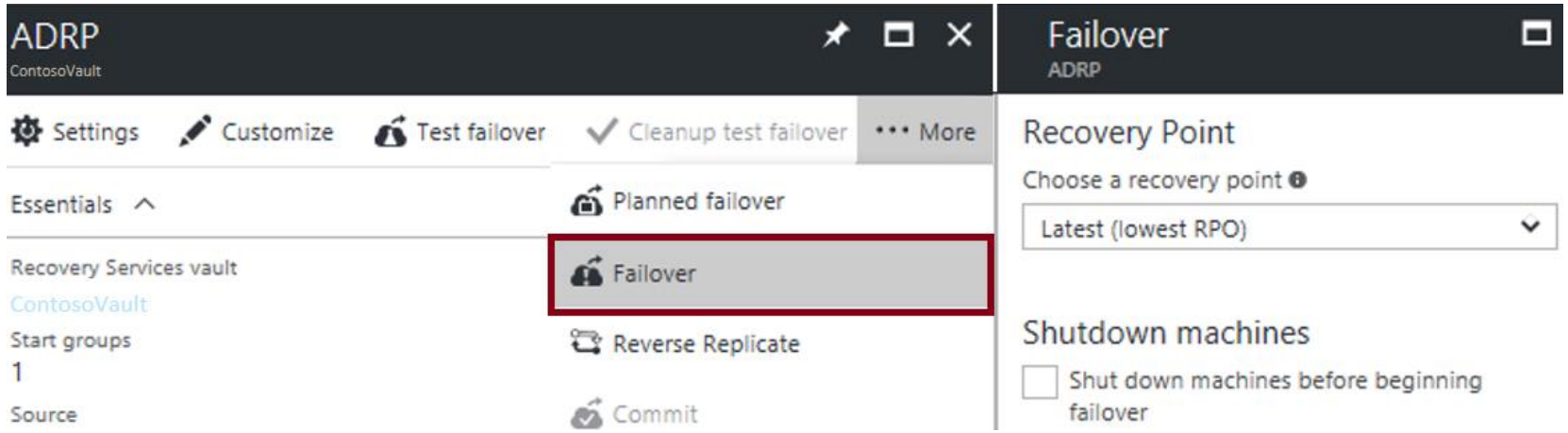
Customize the Recovery Plan

- Select a replication goal
- Set up the source and target environment
- Set up a replication policy
- Enable replication
- Run a test migration to make sure everything's working as expected
- Run a one-time failover to Azure

Planned and Unplanned Failovers

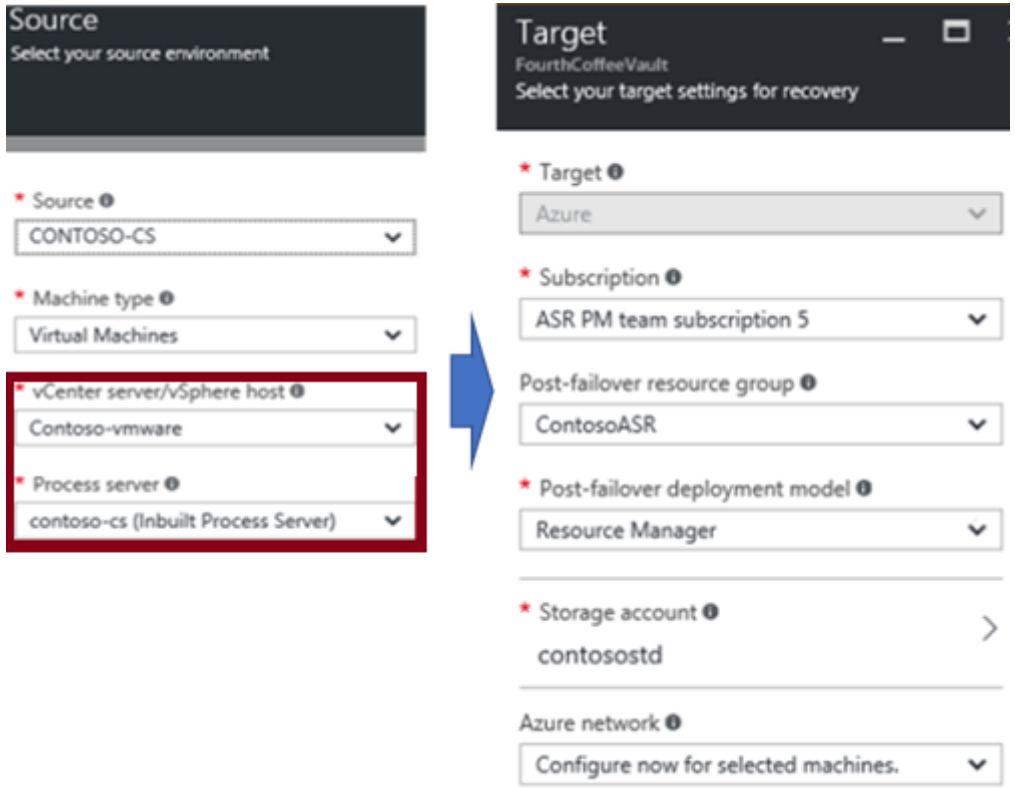
- ASR facilitates two types of failover:
  - **Planned: guaranteed no data loss**
  - **Unplanned: possible data loss**
- ASR failovers:
  - **are not automatic**
  - **can be initiated from the Azure portal**

Scenario	Requirement
Planned failover due to an upcoming datacenter downtime	Zero data loss for the application when a planned activity is performed
Failover due to an unplanned datacenter downtime (natural or IT disaster)	Minimal data loss for the application



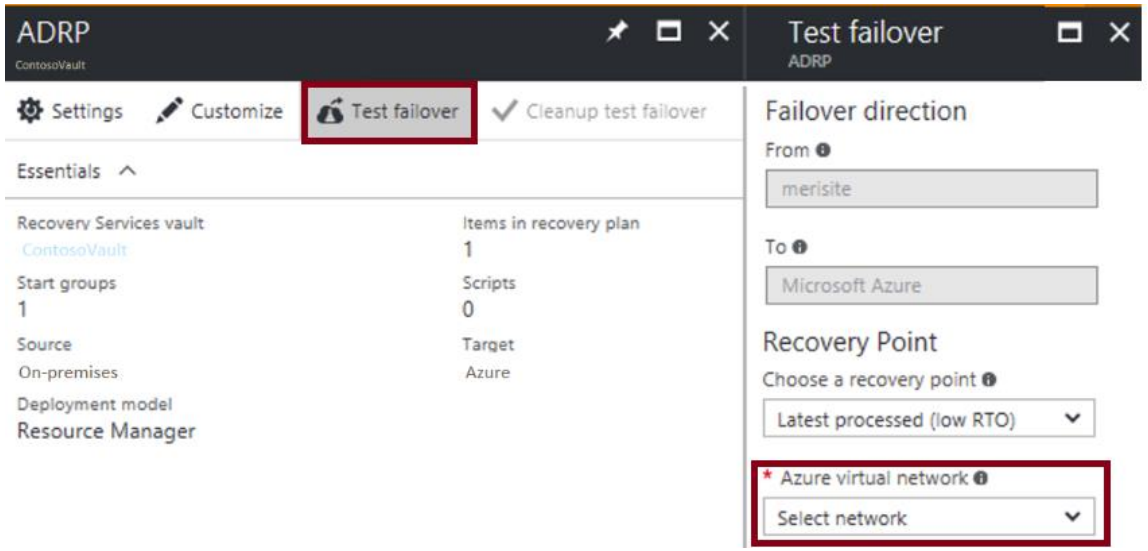
Enable Replication

- Start by configuring:
  - **Replication source**
  - **Replication target**



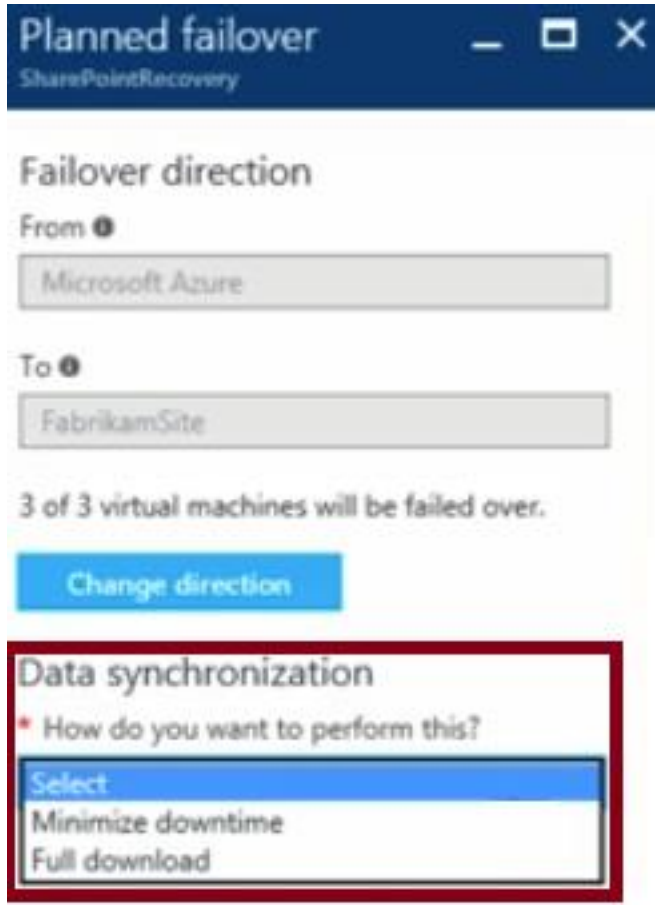
Test Failover

- No-impact validation of replication and disaster recovery strategy:
  - **Does not affect ongoing replication and the production environment**
  - **Supports single or multiple VMs**
  - **Does not require downtime**
  - **Does not introduce data loss**
- Best practices:
  - **Use failover groups**
  - **Failover to an isolated network**
  - **Integrate with Azure Automation**



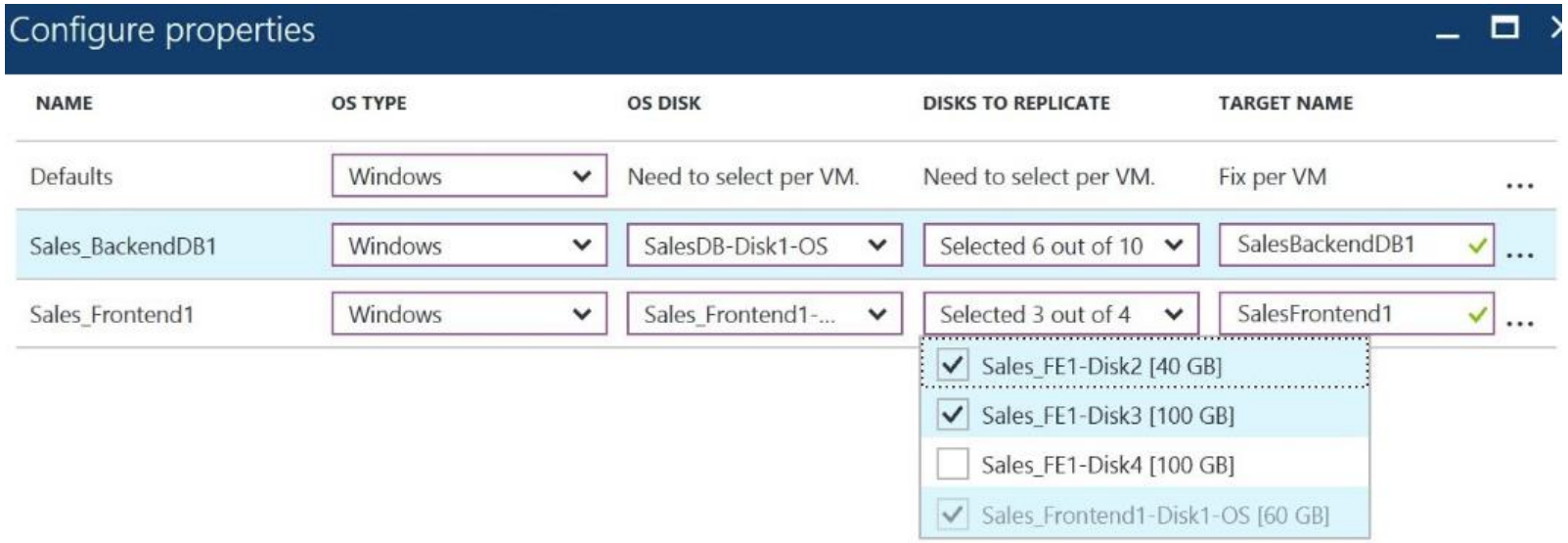
Failback

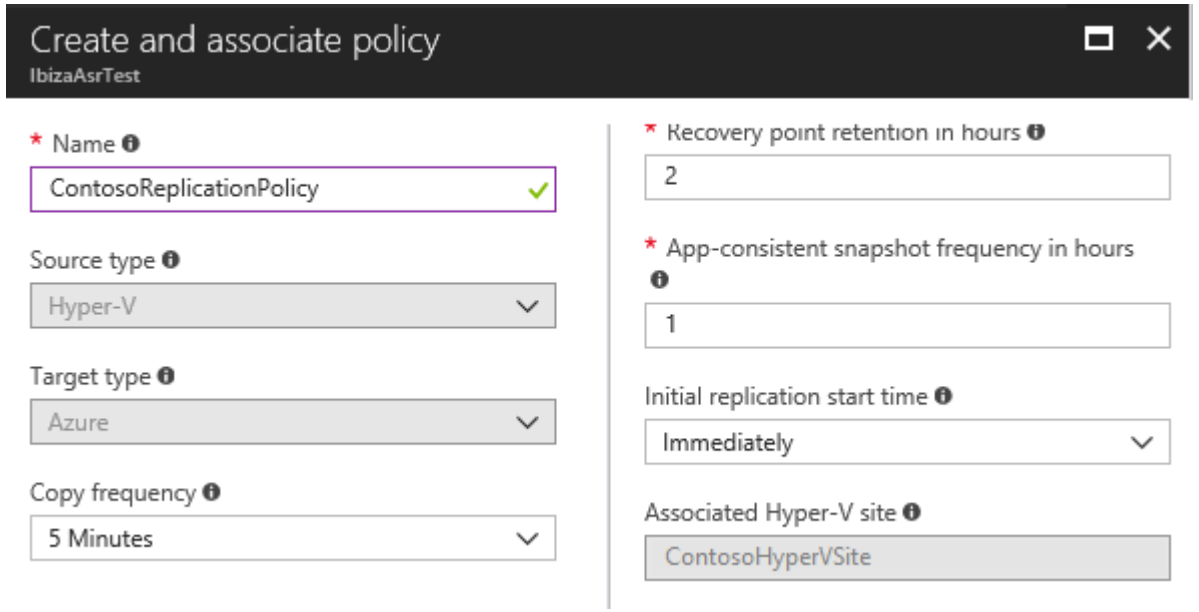
- A planned failover back to the primary site
- Offers two data synchronization methods:
  - **Full download:**
    - **Faster**
    - **Requires VMs to be shut down**
  - **Minimize downtime:**
    - **Slower**
    - **Eliminates the need to shut down VMs**



Virtual Machine Replication

- To configure:
  - **Select source computers**
  - **For each, select the OS type, OS disk, and data disks to replicate**
  - **Optionally, exclude disks (e.g. disks hosting a paging file)**
  - **For each, specify the name of a target VM**



Replication Policy

- To configure, specify the following parameters:
  - **Copy frequency:**
    - **30 seconds**
    - **5 minutes**
    - **15 minutes (to be retired)**
  - **Recovery point retention**
  - **App-consistent snapshot**
  - **Initial replication start time**