

Service Fabric Microservices

- A microservice is whatever you want it to be:
 - ASP.NET, node.js, Java VMs, an arbitrary .exe
- Stateless microservices
 - A microservice that has state where the state is persisted to external storage, such as Azure databases or Azure storage
- Stateful microservices
 - Reliability of state through replication and local persistence
 - Reduces the complexity and number of components in traditional three-tier architecture



Azure Service Fabric

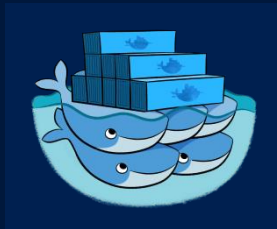
Hyperscale Microservices platform



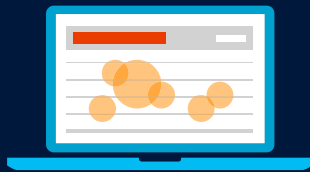
Highly scalable



24 X 7 availability



Windows and Linux
Container
Orchestration



DevOps and
Lifecycle management

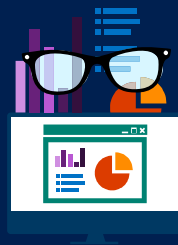
Managed platform



Built-in auto scale
and load balancing



High availability
with auto-patching



Health &
Monitoring



Available on
Private cloud,
Public Cloud &
Hosted Cloud

High productivity development



Simple
Programming models for
.NET, Java



Stateful
microservices



Learn easily using
Party clusters



Simple tooling with
Visual Studio,
Eclipse &
Yeoman

Design web applications

Azure: The Power Of Choice

Compute

Virtual Machines



Container Service



Service Fabric



App Service



More Control

Focus on the App

Customer-managed
(IaaS)

Platform-managed
(PaaS)



Azure App Service

Enterprise-grade apps



Global data
center footprint



Hybrid support



AAD integrated



Secure + compliant

Fully managed platform



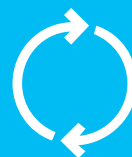
Built-in auto scale
and load balancing



High availability
with auto-patching



Reduced
operations costs



Backup and
recovery

High productivity development



.NET, Java, PHP,
Node, and Python



Staging and
deployment

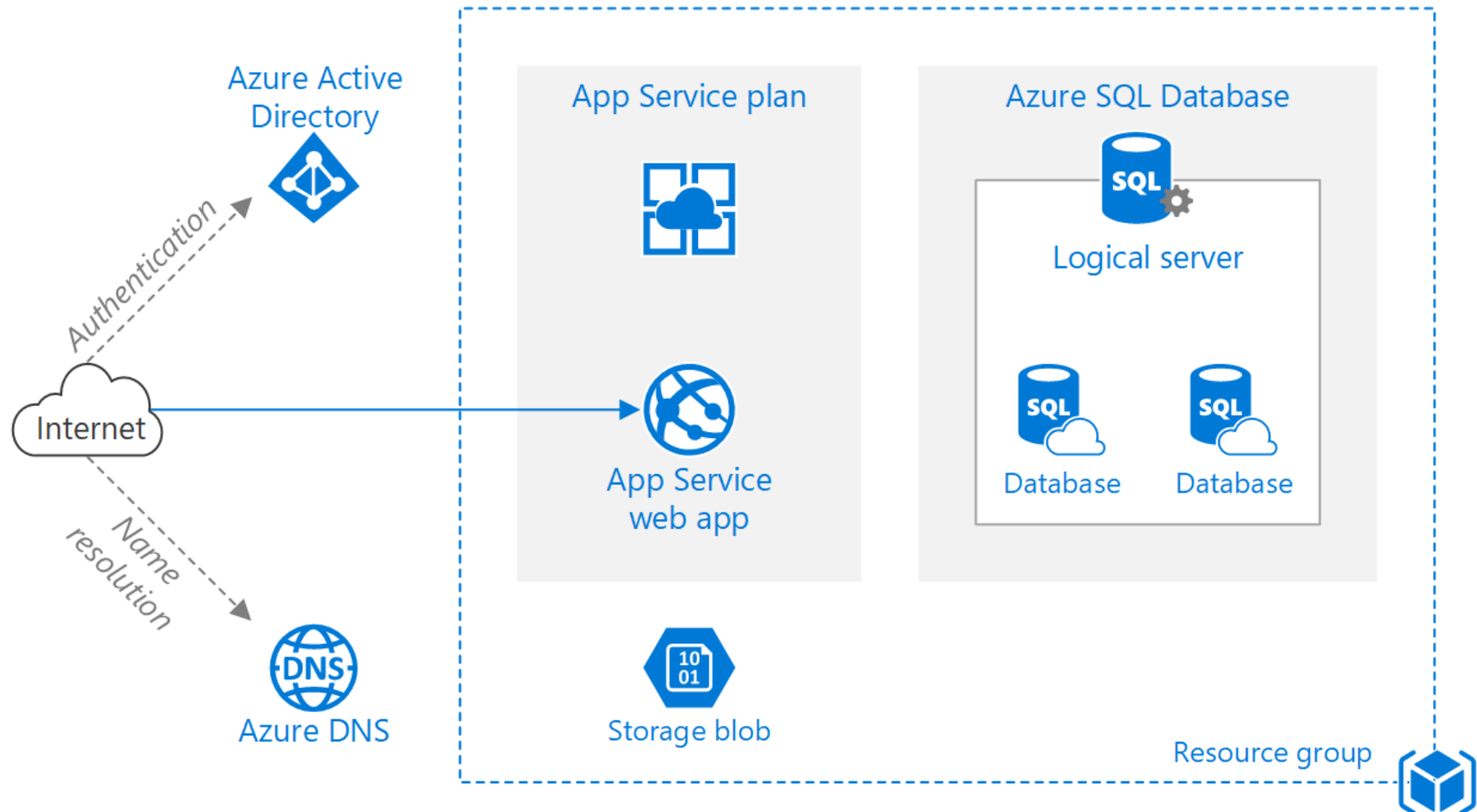


Source code
control
integration

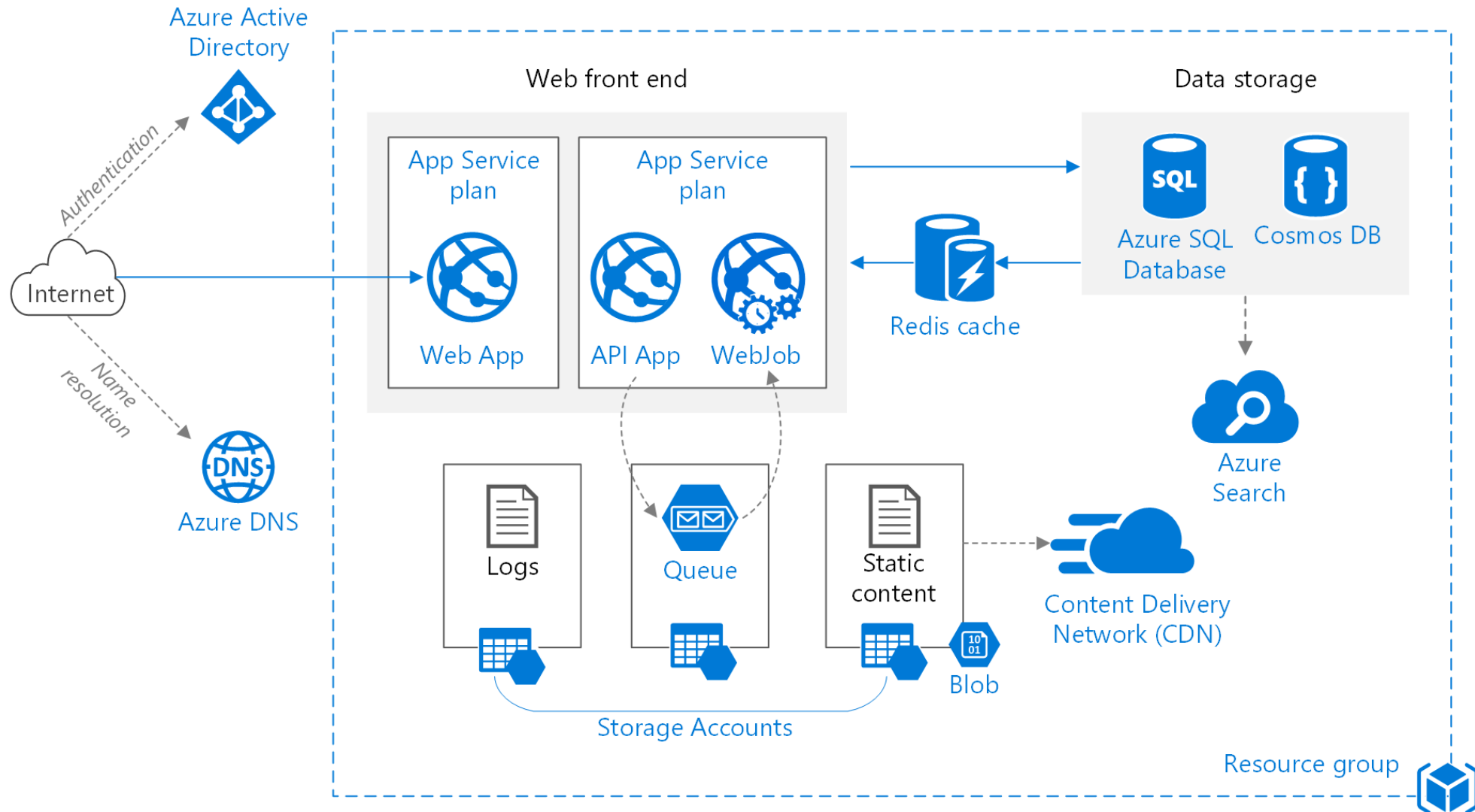


App gallery
marketplace

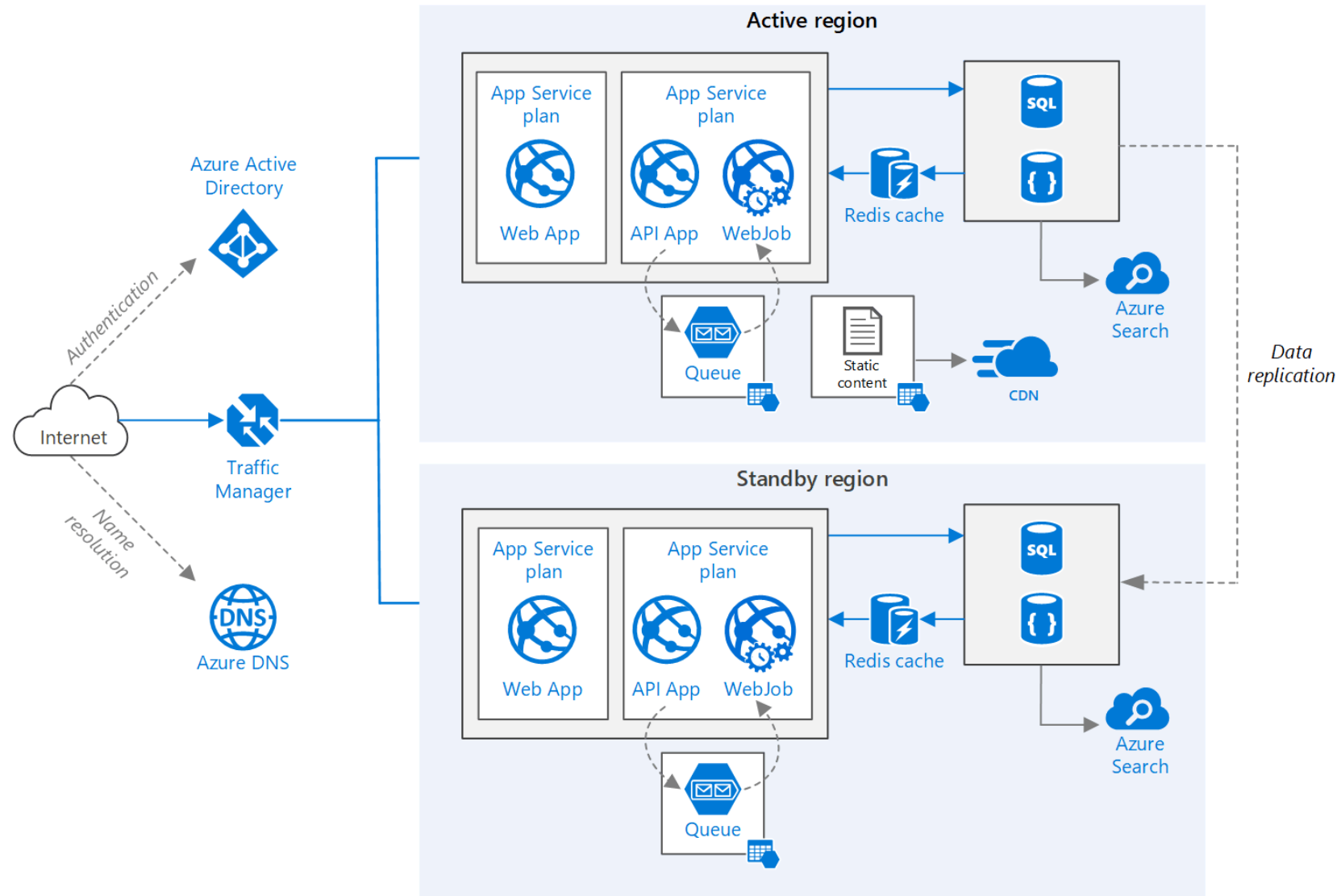
Basic



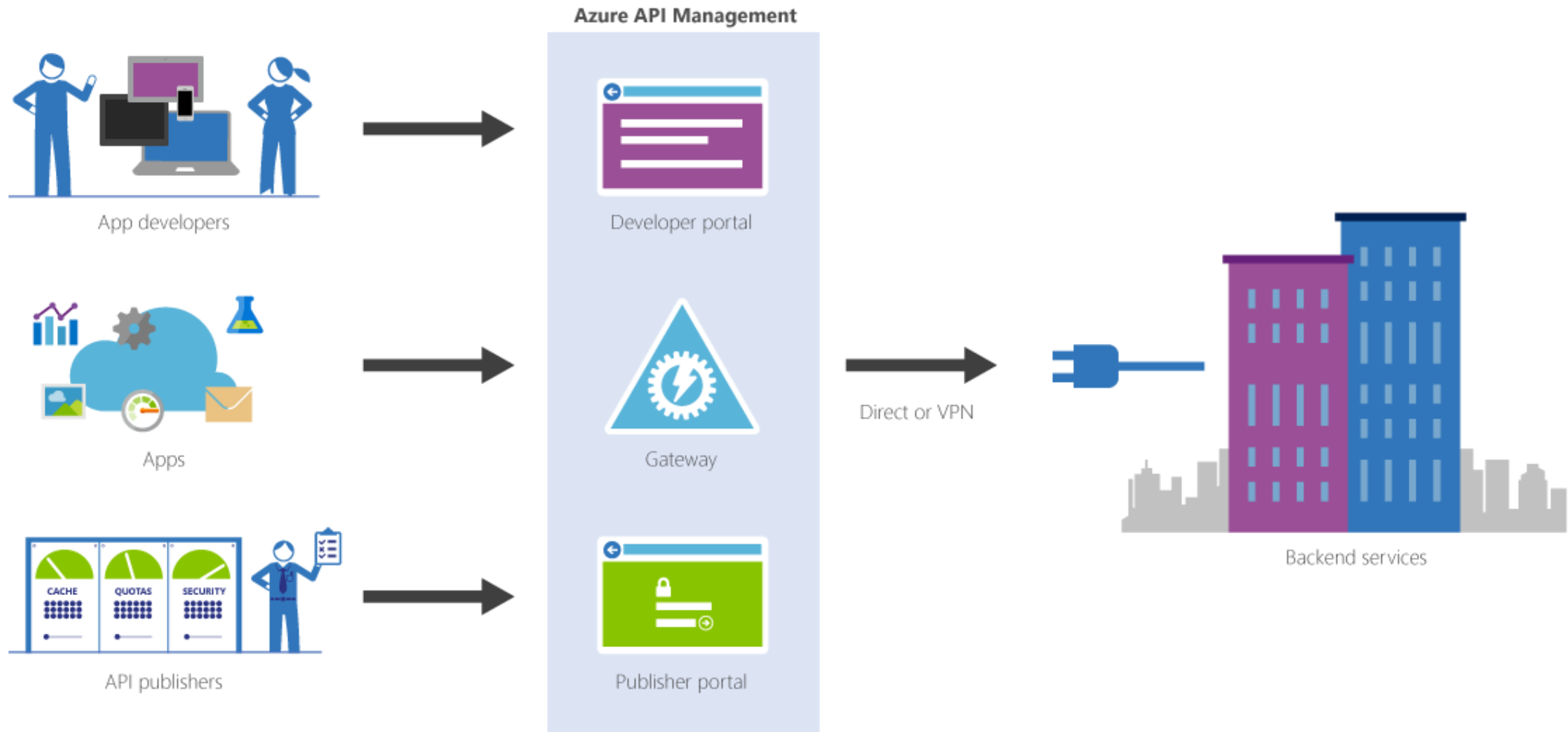
Improved Scalability



Multi-region



Azure API Management



Azure: The Power Of Choice

Compute

Virtual Machines



Container Service



Service Fabric



App Service



Functions



More Control

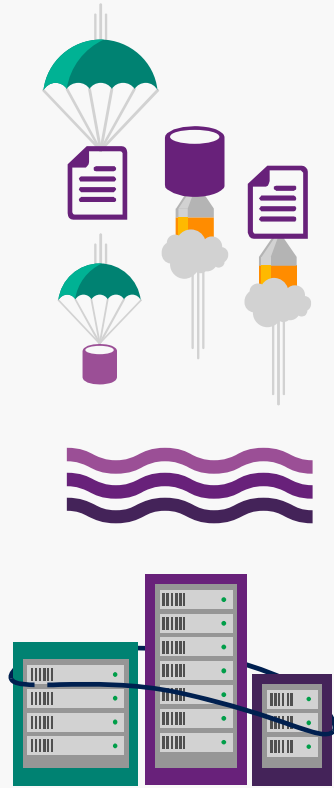
Focus on the App

Customer-managed
(IaaS)

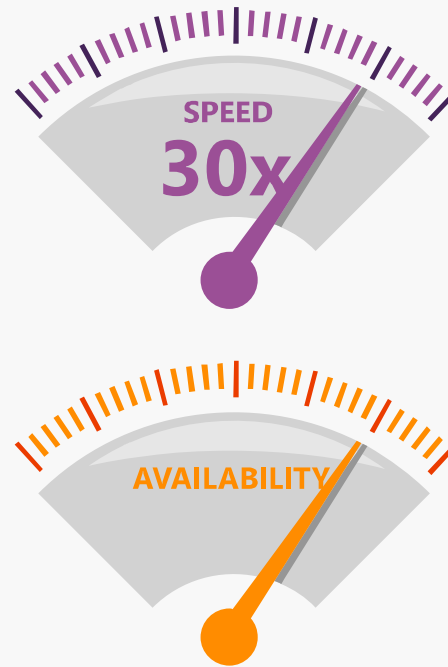
Platform-managed
(PaaS)

Code-only
(serverless)

What is “serverless” and proposed Benefits



Abstraction
of servers



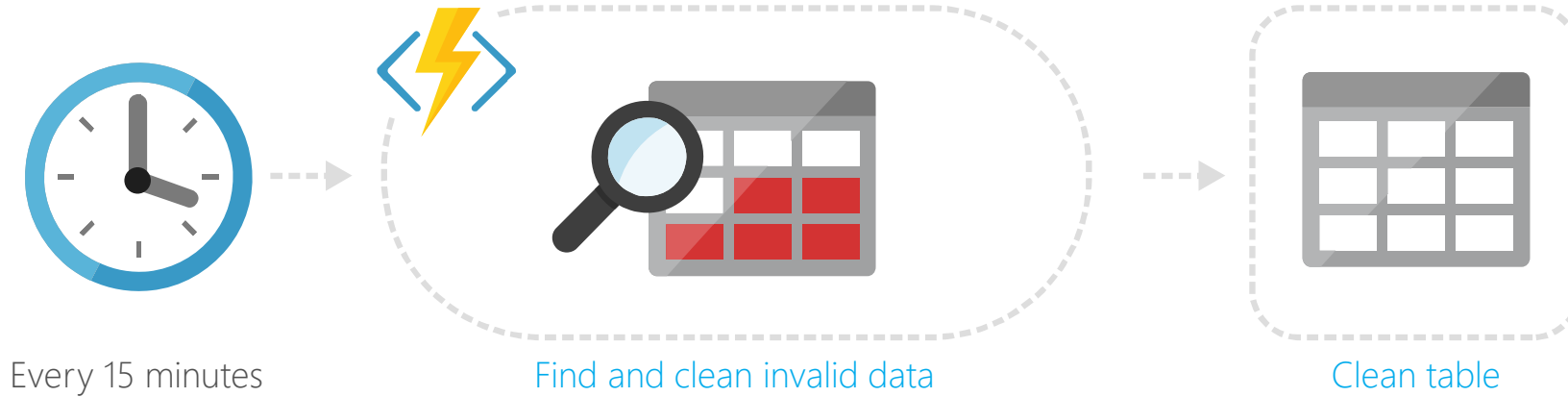
Event-driven
scale



Sub-second
billing

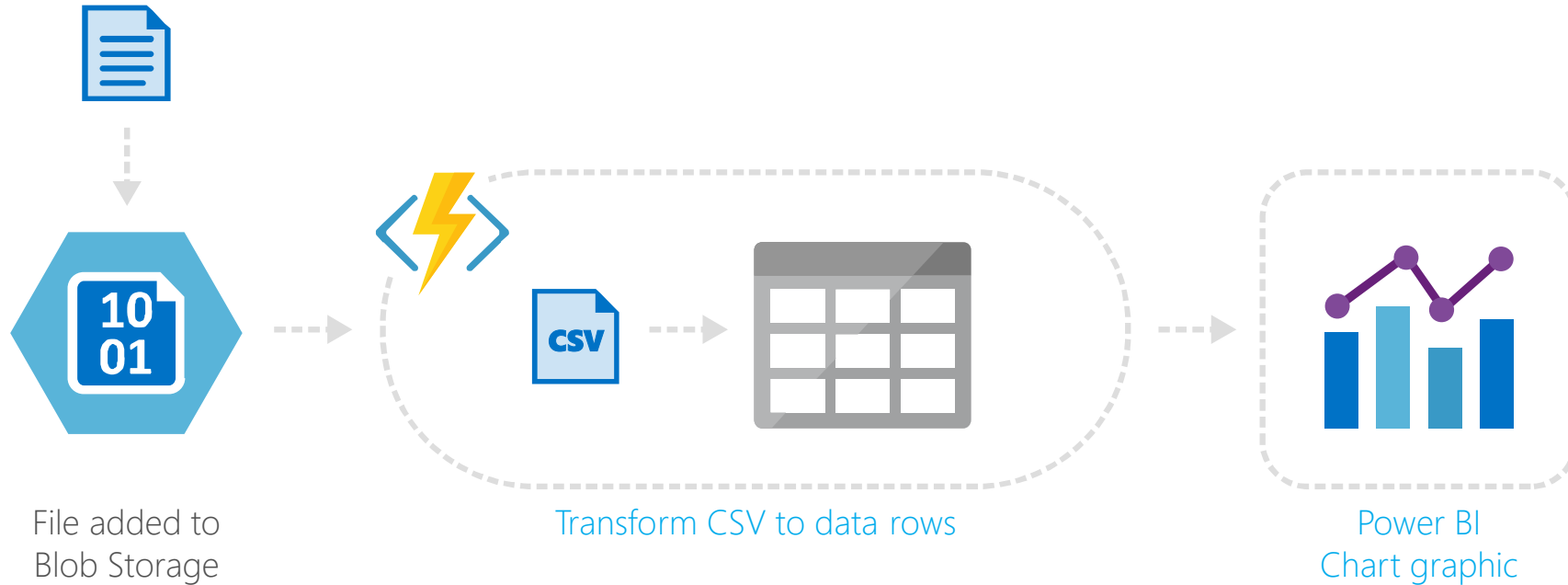
Applications

Example: Timer based processing



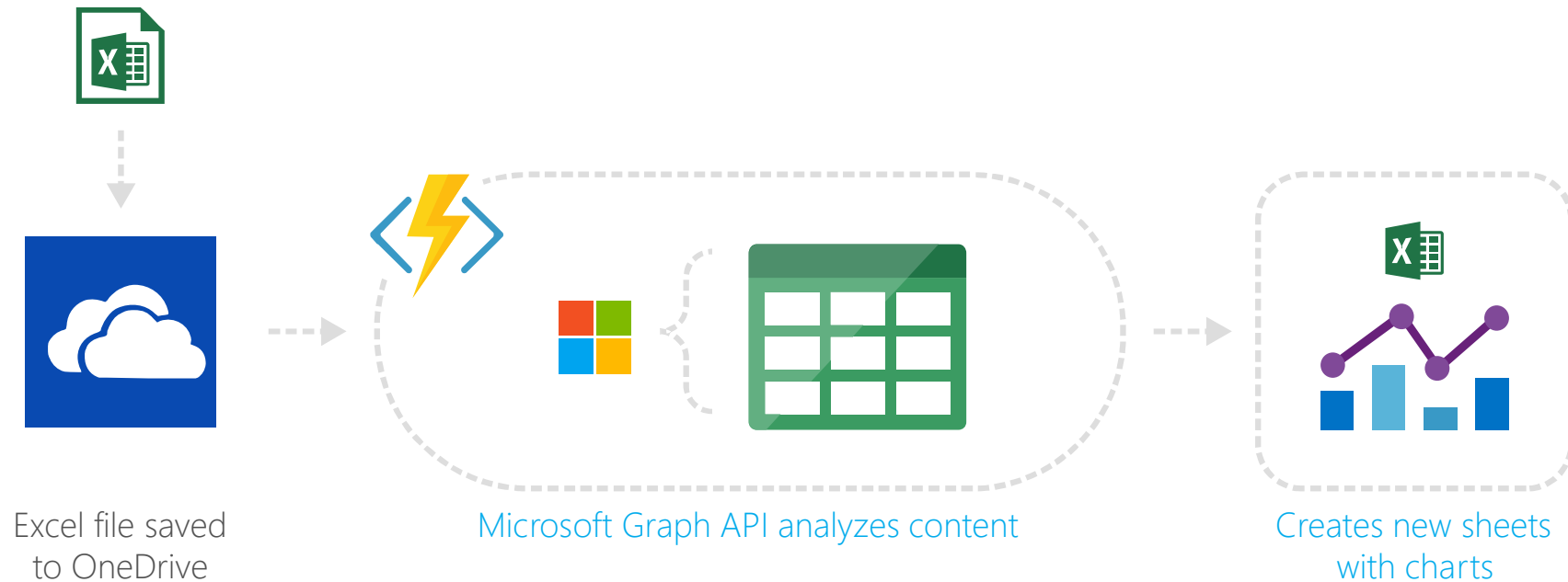
Applications

Example: Azure service event processing



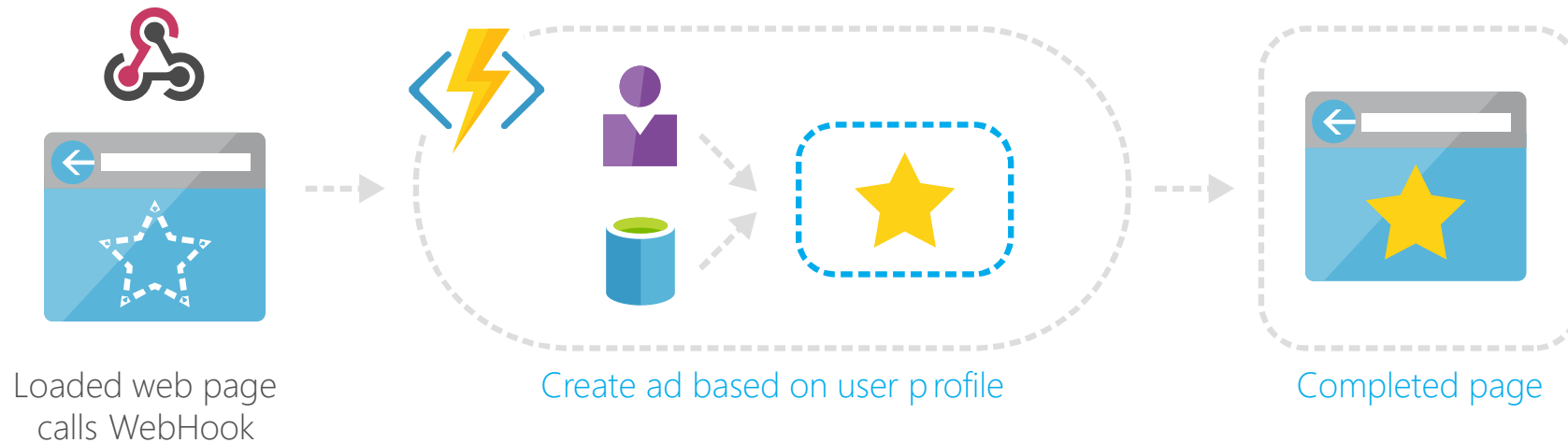
Applications

Example: SaaS event processing



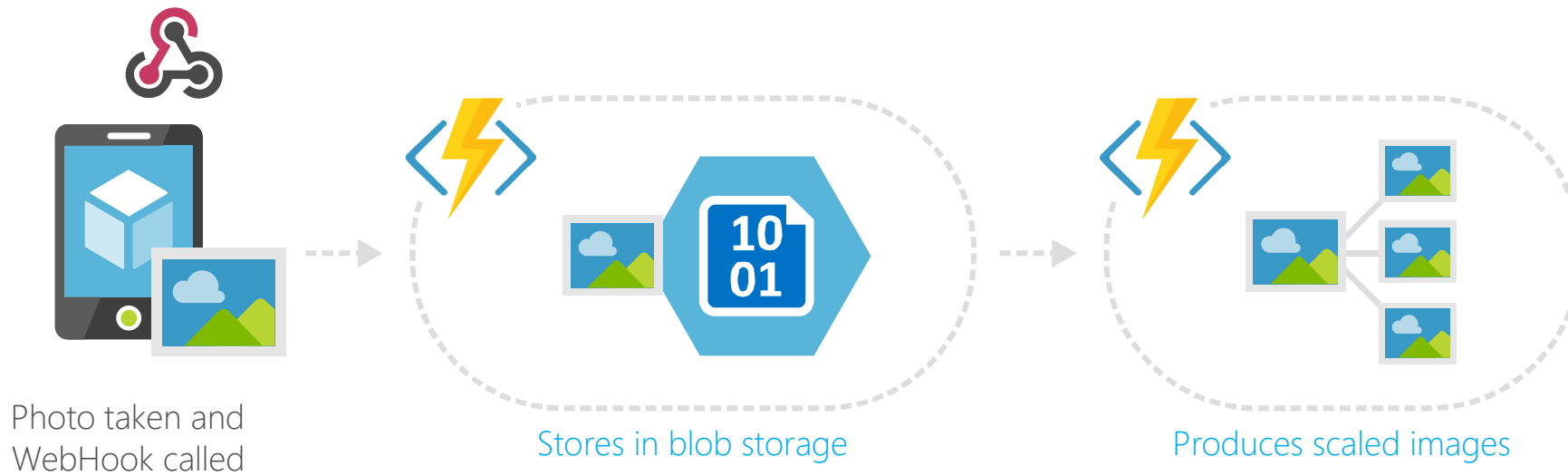
Applications

Example: Serverless Web Applications architectures



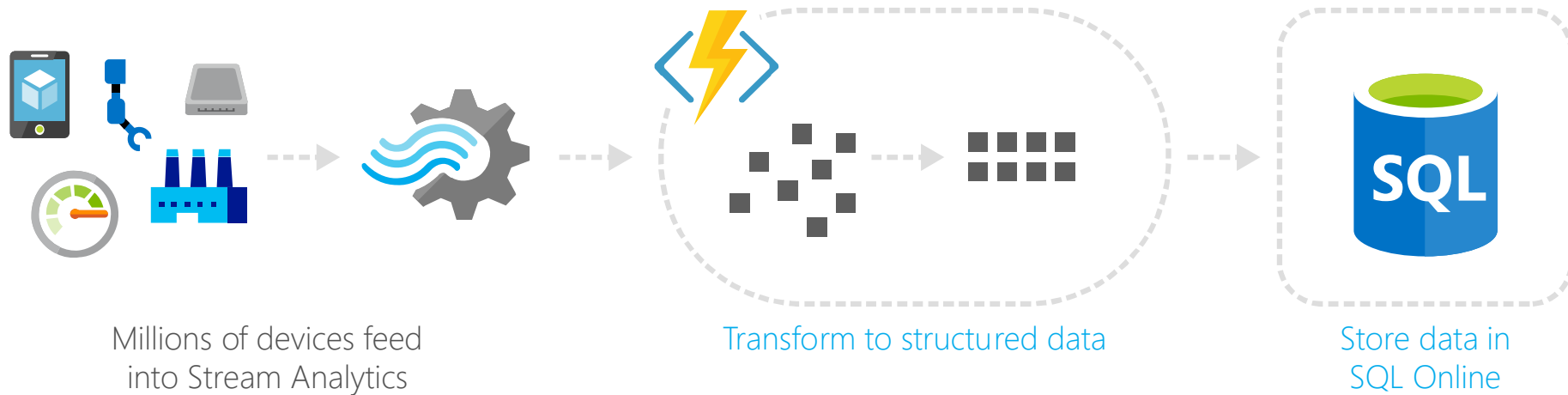
Async background processing

Example: Serverless Mobile back ends



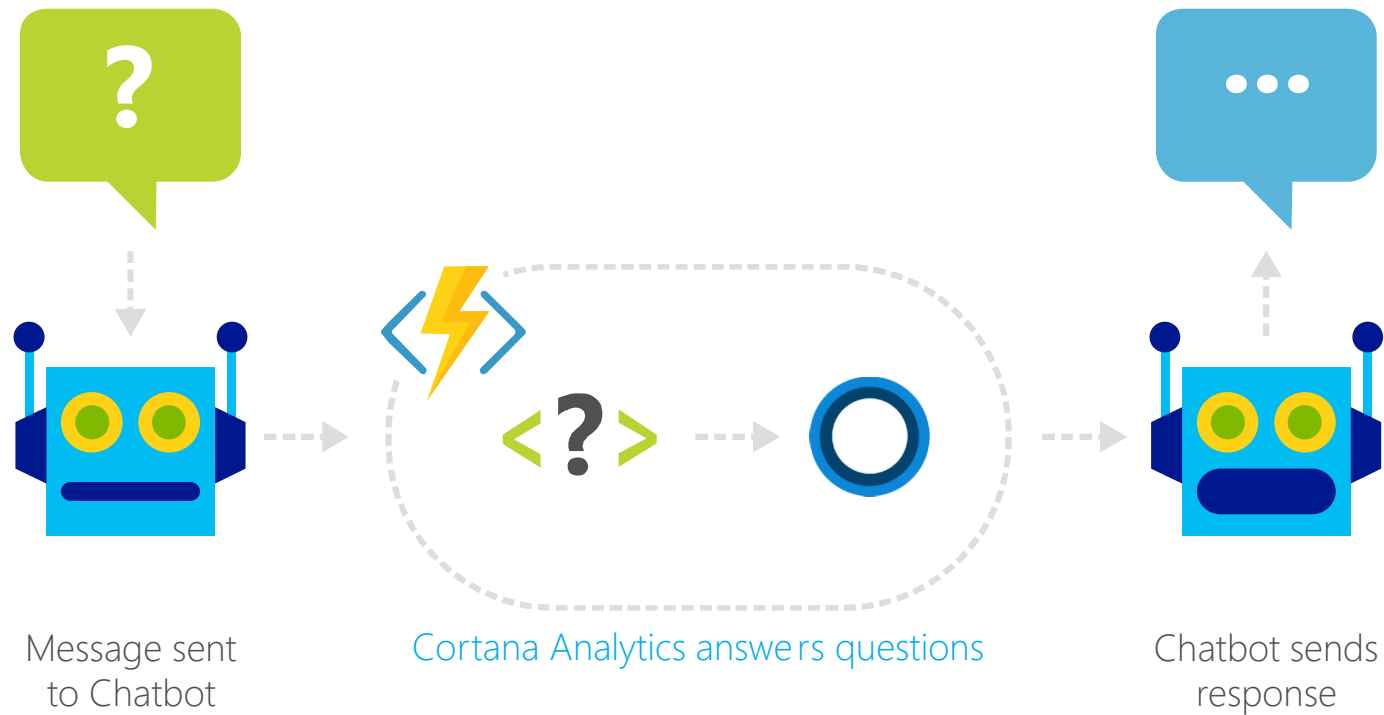
Applications

Example: Real-time stream processing



Applications

Example: Real-time bot messaging





Azure Functions

Serverless



Event-driven
scale



Reduced
Dev Ops

Accelerate development

nodeJS

C#



Develop
your way



Local
development

Bind into services



Azure
Service Bus



Azure
Event Hub



Azure
Storage



Dropbox



Sendgrid



AzureDocDb



OneDrive



Box



Twilio

Azure: The Power Of Choice

Compute

Virtual Machines



Container Service



Service Fabric



App Service



Functions



More Control

Focus on the App

Customer-managed
(IaaS)

Platform-managed
(PaaS)

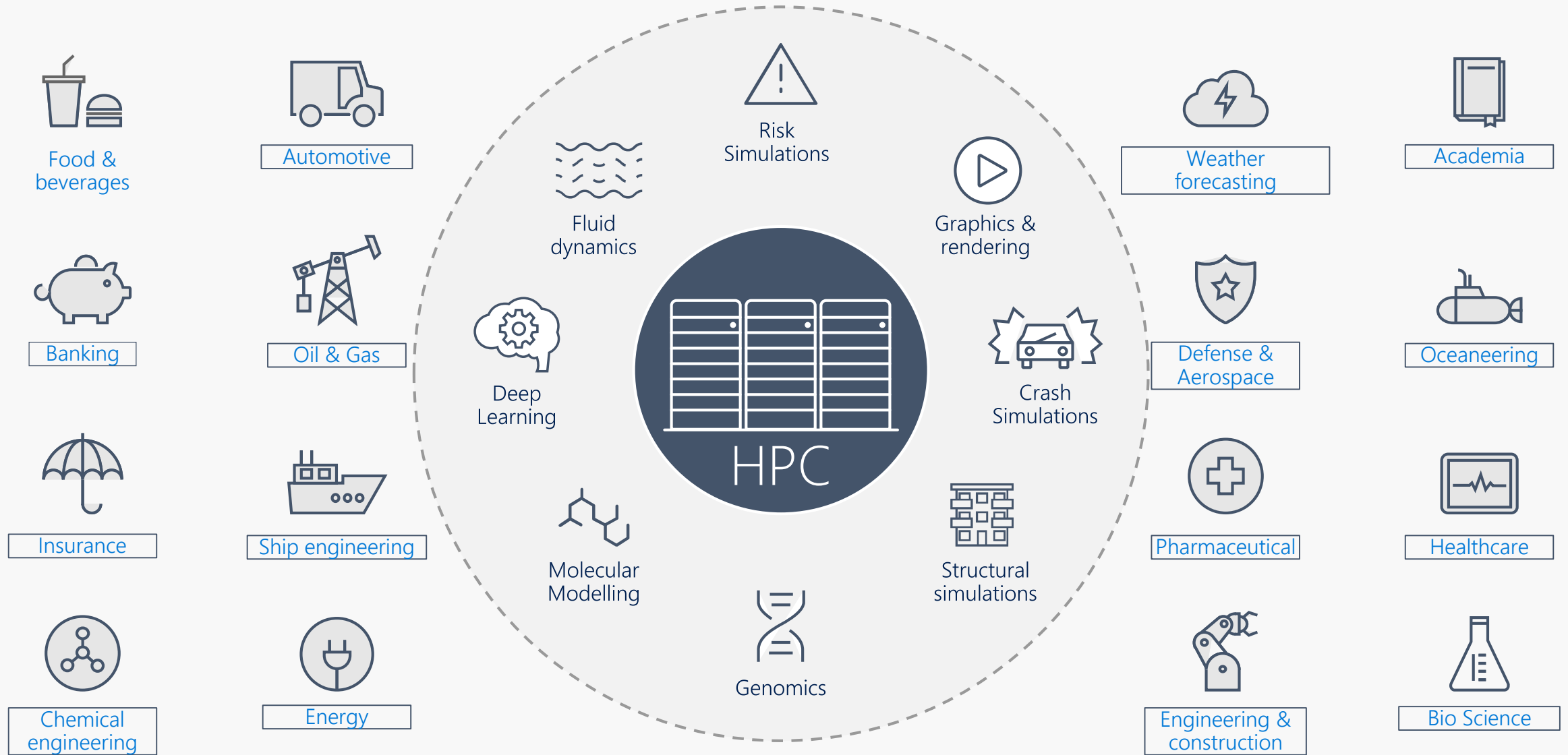
Code-only
(serverless)

Summarizing the options

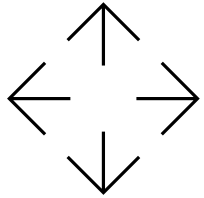
Service	Best used for...
VMs	Lift-and-shift for a set of “pets”
VM Scale Sets	Scaling and managing a set of identical VMs
Batch	Highly parallelized computation
Container Service	Deploying and managing a set of arbitrary Linux containers
Container Instances	Running individual containers with low overhead and no VM management
Service Fabric	Building microservice-based applications on Windows using .NET
App Service	Building standard web and mobile apps with limited management responsibilities
Functions	Building small, event-driven software with granular auto-scale

Create compute-intensive application

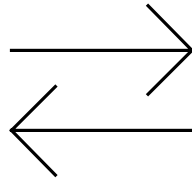
WHAT IS HIGH PERFORMANCE COMPUTING?



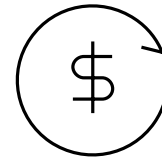
Cloud value for big compute



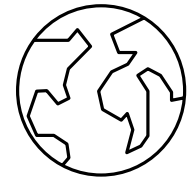
Scale



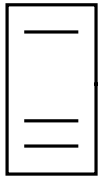
Elasticity



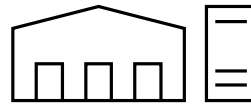
Pay for use



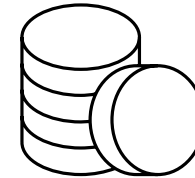
Reach & locality



Hardware selection

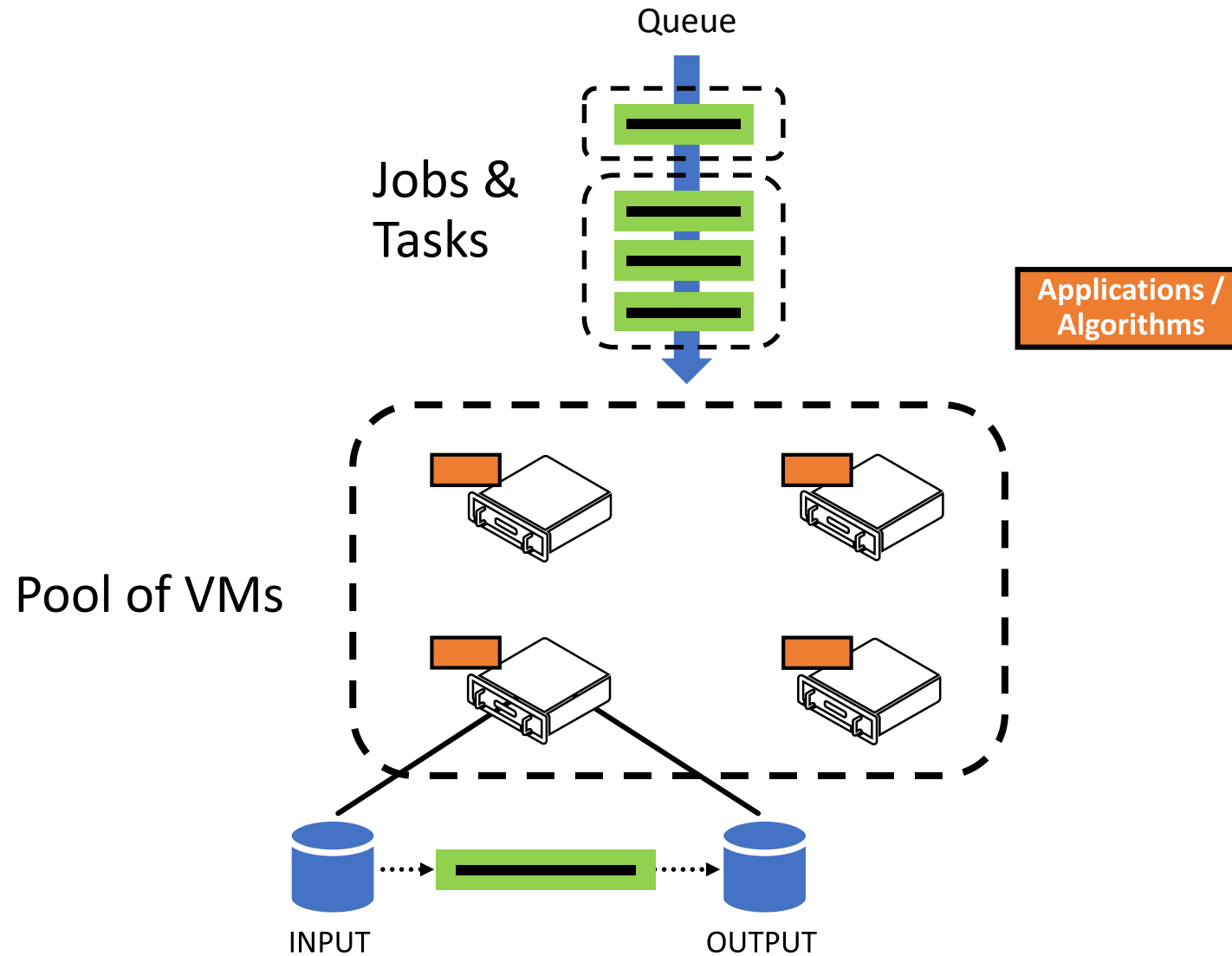


No infrastructure



Known costs

Azure Batch Concepts

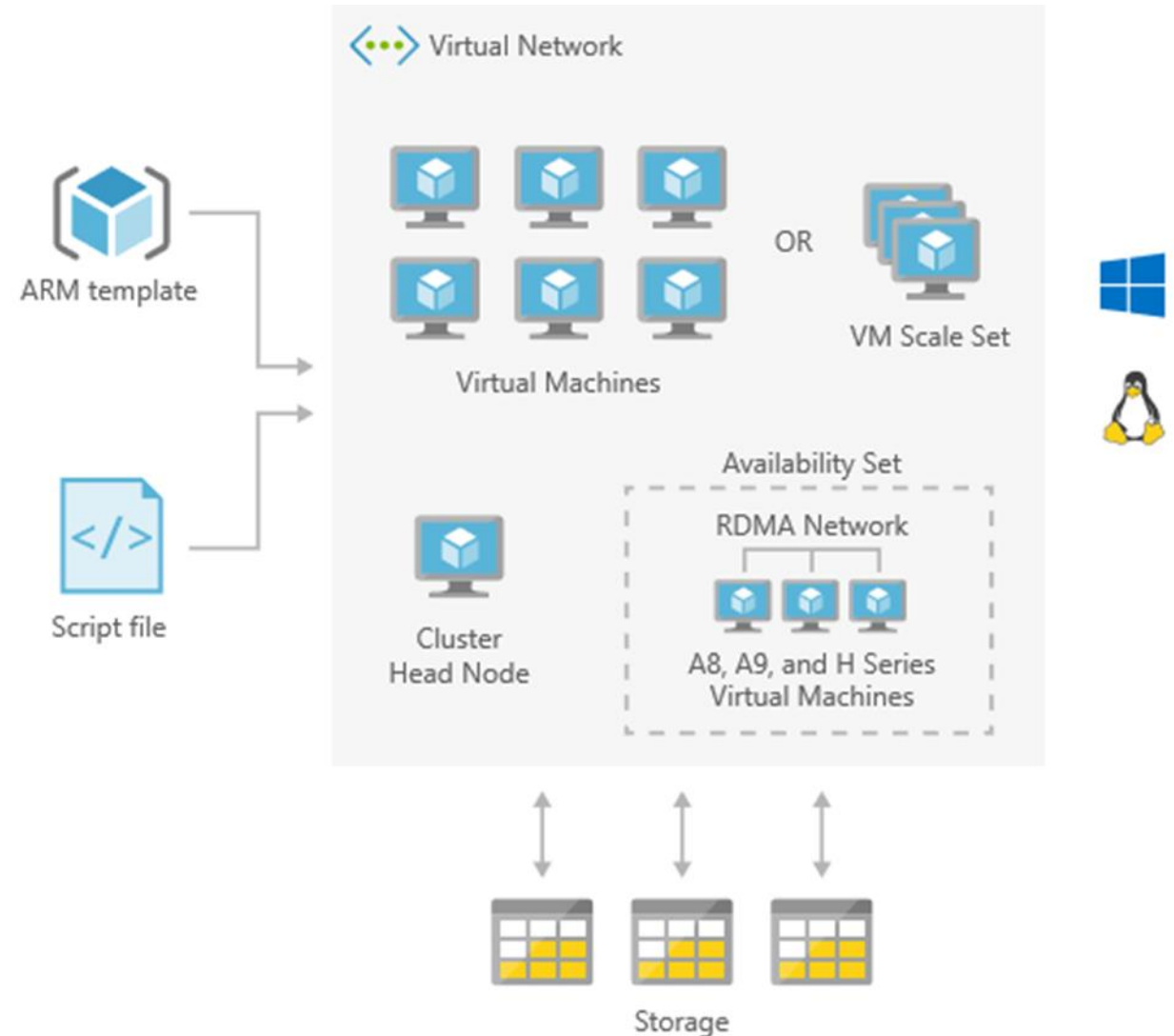


HPC cluster deployed in the cloud

Solution architecture

High performance computing (HPC) applications can scale to thousands of compute cores, extend on-premises big compute, or run as a 100% cloud native solution. This HPC solution including the head node, compute nodes, and storage nodes, runs in Azure with no hardware infrastructure to maintain.

This solution is built on the Azure managed services: [Virtual Machine Scale Sets](#), [Virtual Network](#) and [Storage](#). These services run in a high-availability environment, patched and supported, allowing you to focus on your solution instead of the environment they run in.

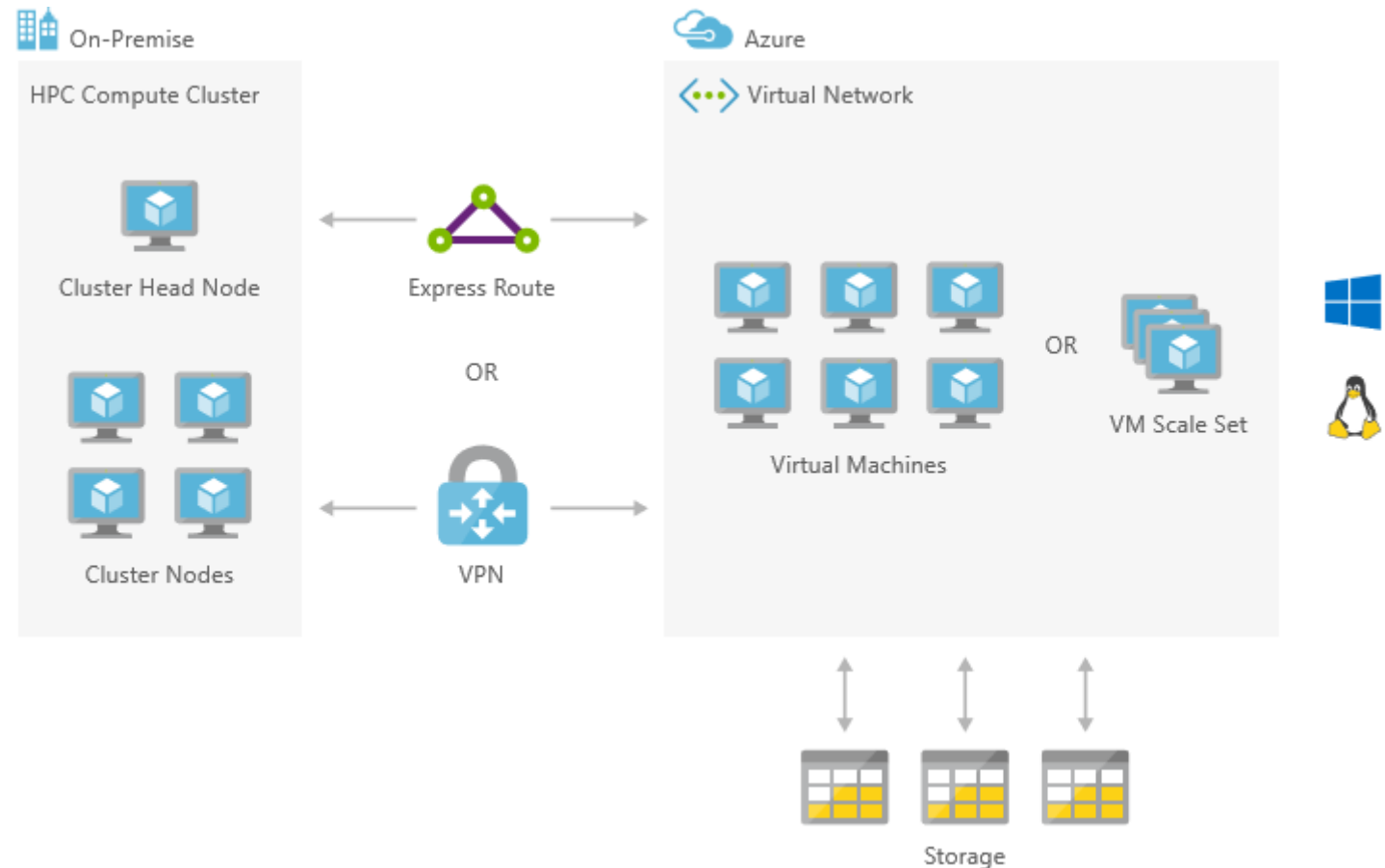


On-premises HPC implementation bursting to Azure

Solution architecture

High performance computing (HPC) applications can scale to thousands of compute cores, extend on-premises big compute, or run as a 100% cloud native solution. This HPC solution can extend its computational capacity by leveraging the compute-intensive instances of Virtual Machines running in Azure and accessed via Express Route or VPN.

This solution is built on the Azure managed services: [Virtual Machines](#), [Virtual Network](#), [VPN Gateway](#), [ExpressRoute](#) and [Storage](#). These services run in a high-availability environment, patched and supported, allowing you to focus on your solution instead of the environment they run in.



Appendix

VMs – Supported Operating Systems

- Linux
 - CentOS
 - CoreOS
 - Debian
 - Oracle
 - Red Hat
 - SUSE
 - openSUSE
 - Ubuntu
- Windows Server
 - Windows 2003*
<https://support.microsoft.com/en-us/help/3206074>
 - **Windows 2008R2+**

Linux distributions and versions supported on Azure

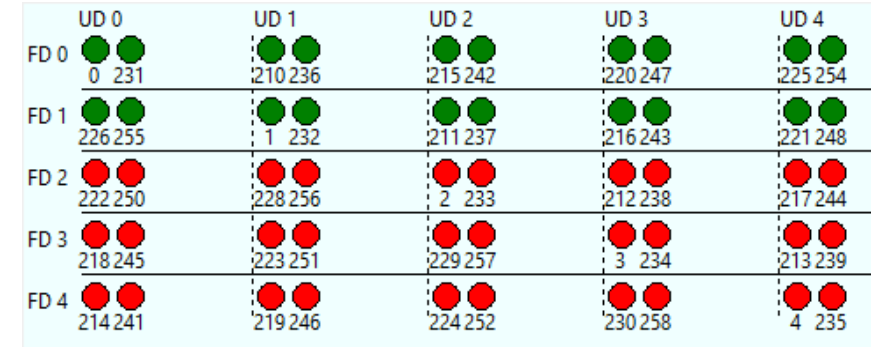
Distribution	Version	Drivers	Agent
CentOS	CentOS 6.3+, 7.0+	CentOS 6.3: LIS download CentOS 6.4+: In kernel	Package: In repo under "WALinuxAgent" Source code: GitHub
CoreOS	494.4.0+	In kernel	Source code: GitHub
Debian	Debian 7.9+, 8.2+	In kernel	Package: In repo under "waagent" Source code: GitHub
Oracle Linux	6.4+, 7.0+	In kernel	Package: In repo under "WALinuxAgent" Source code: GitHub
Red Hat Enterprise Linux	RHEL 6.7+, 7.1+	In kernel	Package: In repo under "WALinuxAgent" Source code: GitHub
SUSE Linux Enterprise	SLES/SLES for SAP 11 SP4 12 SP1+	In kernel	Package: for 11 in Cloud:Tools repo for 12 included in "Public Cloud" Module under "python-azure-agent" Source code: GitHub
openSUSE	openSUSE Leap 42.1+	In kernel	Package: In Cloud:Tools repo under "python-azure-agent" Source code: GitHub
Ubuntu	Ubuntu 12.04, 14.04, 16.04, 16.10	In kernel	Package: In repo under "walinuxagent" Source code: GitHub

VM Scaling

- Resize VMs (Scale up or down)
 - Offline operation
 - Just a reboot if in a cluster that supports new size
 - Requires a deallocation if new size is not supported in the cluster (a little longer)
- Scale In or Out
 - You can create new instances of the VM based on platform images (max 1000) or your own custom images (max 100)
 - Using VM Scale Sets
 - Using Azure Automation
 - Scheduled
 - Azure Autoscale
 - System Metrics (CPU, RAM, etc)
 - Custom Metrics

Virtual Machine Scale Sets

- Deploy and manage VMs as a set
- An Azure Compute resource
Microsoft.Compute/virtualMachineScaleSets
- Scale - Deploy identical resources
 - Easily
 - Rapidly
 - At scale.. 10s – 10,000s of cores
- Low cost - Dynamically increase/decrease compute power to optimize costs
- Elasticity - Automatically scale to changing demand



Search virtual machine instances		
NAME	STATUS	LATEST MODEL
bigcore_0	✓ VM running	Yes
bigcore_1	✓ VM running	Yes
bigcore_2	⊖ VM stopped	⚠ No
bigcore_3	⊖ VM stopped	⚠ No
bigcore_4	⊖ VM stopped	⚠ No
bigcore_210	✓ VM running	Yes
bigcore_211	✓ VM running	Yes

VM Disks

- Disks can be resized up to 4TB
- Managed Vs Unmanaged.
- Encryption
- Disk Caching
 - None – for Write only and Write heavy (SQL Logs)
 - ReadOnly – for Read only or read-intensive disks (SQL Data files)
 - ReadWrite – for Operating System disks

VM Disks

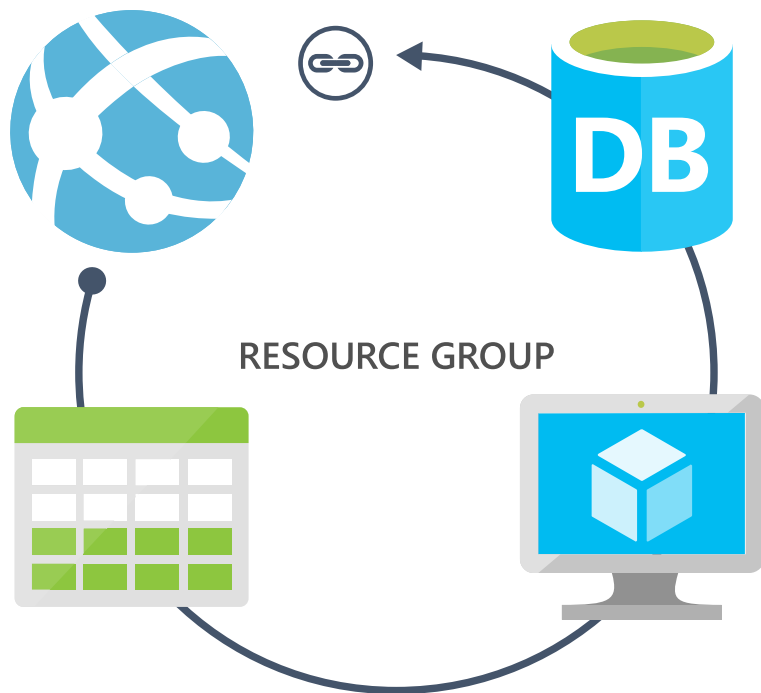
- **Images are representations of your VM that you can use to create more instances of the VM**
 - You can capture an Image of a VM and it will include all the disks
 - Snapshots only capture one disk at a time.
- **Disk Snapshots let you capture current state copies of your disks.**
 - Managed Disks: Take Snapshots of the disk
 - Unmanaged Disks: Take Snapshots of the underlying VHD blob

Exam Tip - Increase size of existing Disks

You can easily increase the size of existing disks. For example, you might want to increase the size of a 30-GB disk to 128 GB, or even to 4 TB. Or, you might want to convert your P20 disk to a P30 disk because you need more capacity or more IOPS and throughput.

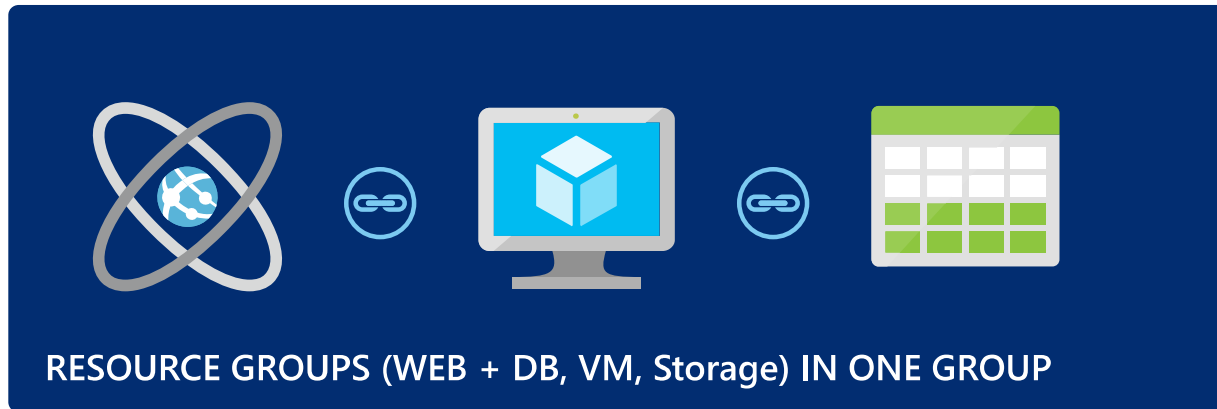
ARM Templates

Azure Resource Manager (ARM)



- Enable application management within Azure
- Resource groups are containers that can contain multiple IaaS + PaaS resources
- Support lifecycle management with integrated Role Based Access Control (RBAC)
- Templatize application deployment and configuration
- Supports DevOps

Azure Resource Manager



OR



Describe

WHERE

Resource Inventory

WHAT

Component Relationships

HOW

Tags + links + groups

Provision

WHERE

Across Regions

WHAT

Across Resources

HOW

In service and in guest

Control

WHO

Access control

WHAT

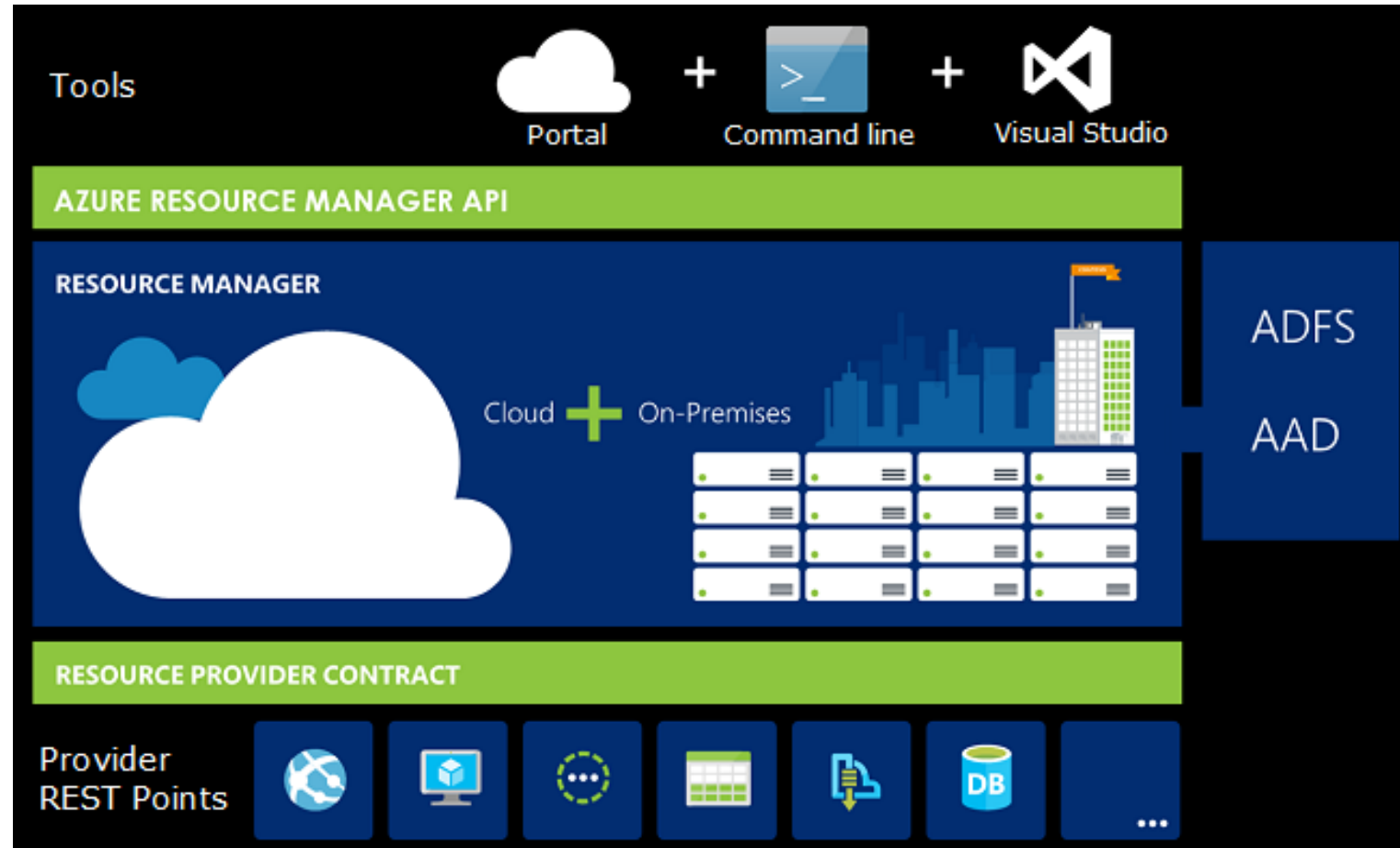
Changes

HOW

RBAC

What is ARM?

- Resource
- Resource Group
- Resource Provider
- Resource Manager Template



ARM Templates - Structure

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

Element name	Required	Description
\$schema	Yes	Location of the JSON schema file that describes the version of the template language. Use the URL shown in the preceding example.
contentVersion	Yes	Version of the template (such as 1.0.0.0). You can provide any value for this element. When deploying resources using the template, this value can be used to make sure that the right template is being used.
parameters	No	Values that are provided when deployment is executed to customize resource deployment.
variables	No	Values that are used as JSON fragments in the template to simplify template language expressions.
resources	Yes	Resource types that are deployed or updated in a resource group.
outputs	No	Values that are returned after deployment.

Template Deployment Mode

- Complete: Resource Manager **deletes** resources that exist in the resource group but are not specified in the template.
- **Incremental**: Resource Manager **leaves unchanged** resources that exist in the resource group but are not specified in the template.

- **Existing Resource Group**

contains:

- Resource A
- Resource B
- Resource C

- **Template defines:**

- Resource A
- Resource B
- Resource D

When deployed in incremental mode, Resource C is deleted. The resource group contains:

Resource A
Resource B
Resource C
Resource D

When deployed in complete mode, Resource C is deleted.
The resource group contains:

Resource A
Resource B
Resource D

To use complete mode, use the Mode parameter

ARM Templates - Parameters

Element name	Required	Description
parameterName	Yes	Name of the parameter. Must be a valid JavaScript identifier.
type	Yes	Type of the parameter value. See the list of allowed types after this table.
defaultValue	No	Default value for the parameter, if no value is provided for the parameter.
allowedValues	No	Array of allowed values for the parameter to make sure that the right value is provided.
minValue	No	The minimum value for int type parameters, this value is inclusive.
maxValue	No	The maximum value for int type parameters, this value is inclusive.
minLength	No	The minimum length for string, secureString, and array type parameters, this value is inclusive.
maxLength	No	The maximum length for string, secureString, and array type parameters, this value is inclusive.
description	No	Description of the parameter that is displayed to users through the portal.

- Allowed Types

- string
- secureString
- int
- bool
- object
- secureObject
- array

```
"parameters": {
  "<parameter-name>" : {
    "type" : "<type-of-parameter-value>",
    "defaultValue": "<default-value-of-parameter>",
    "allowedValues": [ "<array-of-allowed-values>" ],
    "minValue": <minimum-value-for-int>,
    "maxValue": <maximum-value-for-int>,
    "minLength": <minimum-length-for-string-or-array>,
    "maxLength": <maximum-length-for-string-or-array-parameters>,
    "metadata": {
      "description": "<description-of-the parameter>"
    }
  }
}
```

Re-use ARM Templates

By passing in parameters to your ARM Templates, you can re-use them in many different scenarios. You can deploy full solutions with t-shirt sizing (Small, Medium, Large) based on parameters you pass in: Size of VM, # of VMs, etc.

Be aware of what values in the ARM template need to be unique when re-using templates: <https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-manager-template-best-practices#resource-names>

<https://docs.microsoft.com/en-us/azure/azure-resource-manager/best-practices-resource-manager-design-templates>

ARM Templates –Variables

```
"variables": {  
    "<variable-name>": "<variable-value>",  
    "<variable-name>": {  
        <variable-complex-type-value>  
    }  
},  
"resources": [
```

- Use Variables for
 - Use variables for values that you need to use more than once in a template.
 - Include variables for resource names that must be unique.
 - Use variables for extra manipulation of parameters
- You can group variables into complex objects. Use the **variable.subentry** format to reference a value from a complex object.

```
"variables": {  
    "storage": {  
        "name": "[concat(uniqueString(resourceGroup().id),'storage')]",  
        "type": "Standard_LRS"  
    }  
},  
"resources": [
```

ARM Templates – Functions and Expressions

- Templates are JSON
- **Functions and Expressions extend the JSON capabilities**
- They let you use some coding notions inside your templates to create or evaluate dynamic values
- Functions you see often:
 - concat – concatenates multiple values
 - copyIndex – returns the index of an iteration loop
 - resourceGroup – access properties of this resource group, like the location.
 - resourceId – get the unique identifier for a resource.
- All Functions: <https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-group-template-functions>

```
"variables": {  
  "location": "[resourceGroup().location]",  
  "usernameAndPassword": "[concat(parameters('username'), ':', parameters('password'))]",  
  "authorizationHeader": "[concat('Basic ', base64(variables('usernameAndPassword')))]"  
}
```

ARM Templates - Dependencies

- **Resources may require a dependency chain; e.g. You need a VNET before you can deploy a VM.**
- Use “dependsOn” property to explicitly build the dependencies between resources. Resources won't get created until their dependencies are created.
- Use Child resources. Only certain resources can have child resources.
- Use the “reference” function to create an implicit relationship. Resources that reference another resource are created after the referenced resource.

```
{
  "type": "Microsoft.Compute/virtualMachineScaleSets",
  "name": "[variables('namingInfix')]",
  "location": "[variables('location')]",
  "apiVersion": "2016-03-30",
  "tags": {
    "displayName": "VMScaleSet"
  },
  "dependsOn": [
    "[variables('loadBalancerName')]",
    "[variables('virtualNetworkName')]",
    "storageLoop",
  ],
  ...
}
```

```
JSON Copy

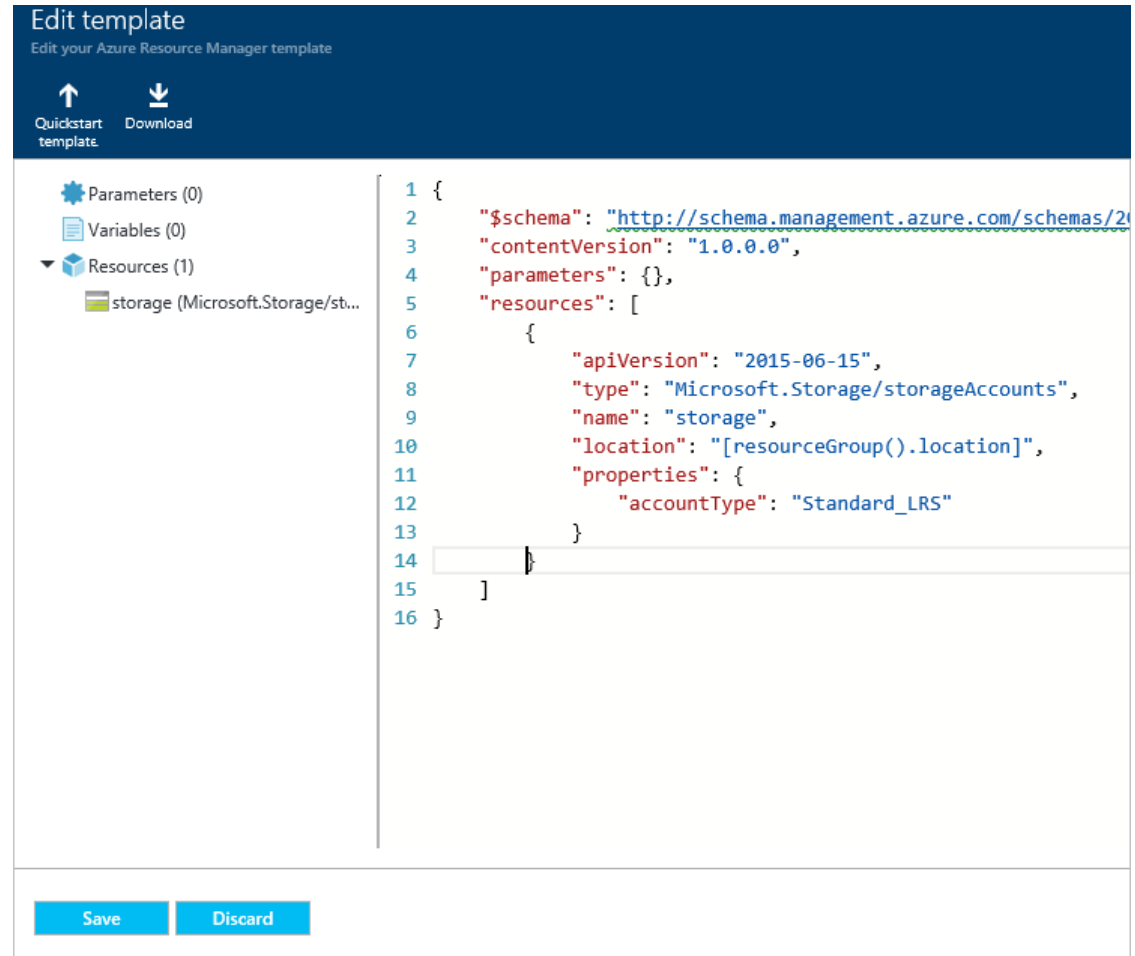
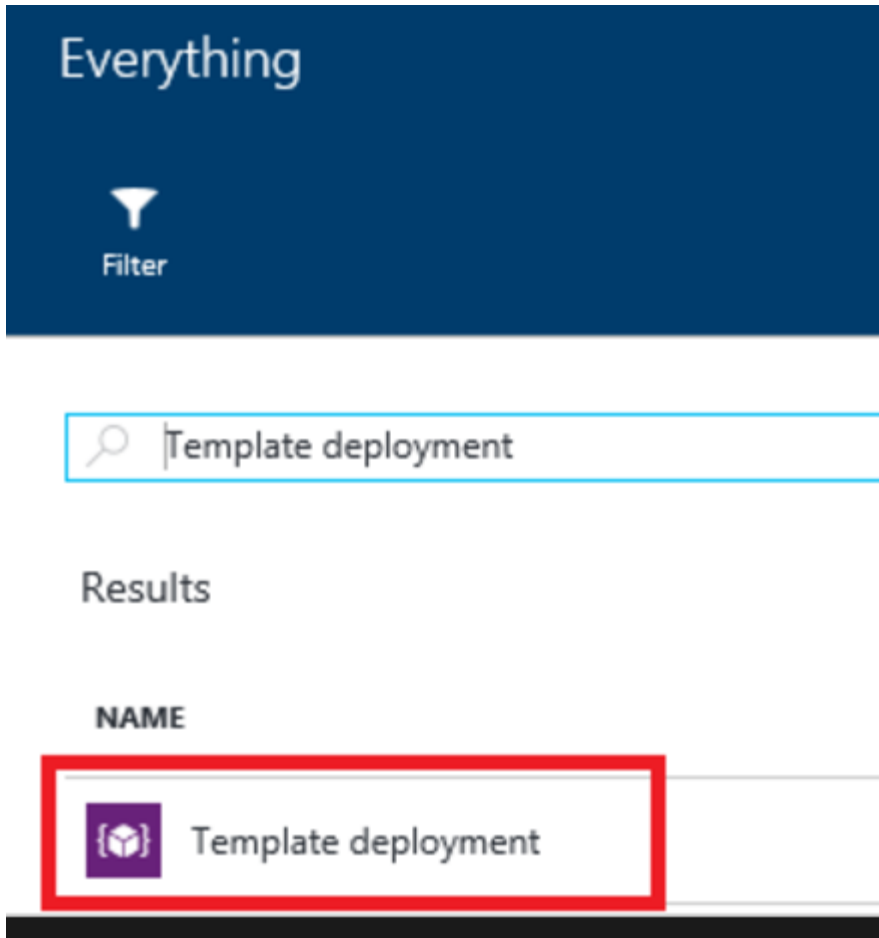
{
  "name": "[variables('endpointName')]",
  "type": "endpoints",
  "location": "[resourceGroup().location]",
  "apiVersion": "2016-04-02",
  "dependsOn": [
    "[variables('profileName')]"
  ],
  "properties": {
    "originHostHeader": "[reference(variables('webAppName')).hostNames[0]]",
    ...
  }
}
```

ARM Templates Advanced Topics

- Creating and Updating Resources in one ARM Template based deployment: <https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-manager-update>
- Share state between linked templates: <https://docs.microsoft.com/en-us/azure/azure-resource-manager/best-practices-resource-manager-state>
- Patterns for deploying resources: <https://docs.microsoft.com/en-us/azure/azure-resource-manager/best-practices-resource-manager-design-templates>

Deploy Templates


Deploy Templates – Portal, Custom Template




Deploy Templates – Portal, Quickstart

Edit template

Edit your Azure Resource Manager template


Quickstart
template


Download

Load a quickstart template

Select a template (disclaimer) ⓘ

101-vm-simple-linux

This template takes a minimum amount of parameters and deploys a Linux VM, using the latest patched version.

Author: coreysa
Last updated: 2015-12-21
[Learn more](#)

OK

Cancel

Deploy Templates – Portal, from Account

The screenshot illustrates the steps to deploy a template in the Azure Portal. The left sidebar shows the navigation menu with 'Browse >' highlighted. The main area displays the 'Templates' section, which includes a search bar and a table of templates. The 'myvmtemplate' is highlighted, and a detail view shows the 'Deploy' button highlighted.

Navigation Menu:

- Subscriptions
- App Service plans
- Storage accounts
- Browse >**

Templates List:

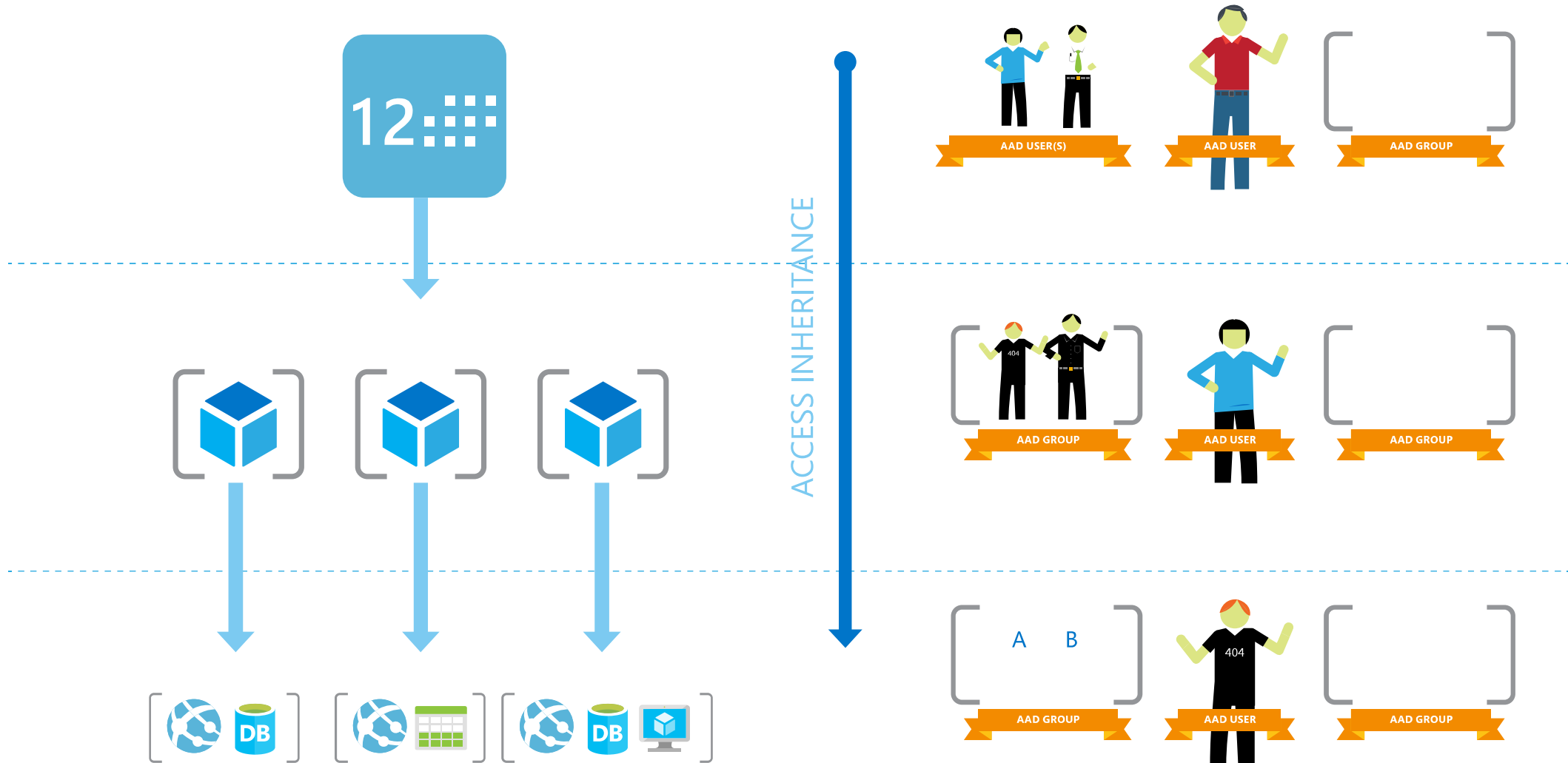
NAME	DESCRIPTION	MODIFIED	SHARED WITH
myvmtemplate	An example template	4/8/2016	Only me

Template Detail View:

myvmtemplate
Template - PREVIEW

Deploy Edit Delete Share

ARM Role Based Access Control



Deploy Templates – Portal

[Create resource group](#)

[Deploy resources from Marketplace](#)

[Deploy resources from custom template](#)

[Deploy resources from a template saved to your account](#)

[Next Steps](#)

Deploy Templates - Powershell

- Deploy a Template from a local json file

```
Login-AzureRmAccount
```

```
New-AzureRmResourceGroup -Name ExampleResourceGroup -Location "South Central US"
New-AzureRmResourceGroupDeployment -Name ExampleDeployment -ResourceGroupName ExampleResourceGroup `
  -TemplateFile c:\MyTemplates\storage.json -storageAccountType Standard_GRS
```

Deploy a Template from an external source:

```
New-AzureRmResourceGroupDeployment -Name ExampleDeployment -ResourceGroupName ExampleResourceGroup `
  -TemplateUri https://raw.githubusercontent.com/Azure/azure-quickstart-templates/master/101-storage-account-create/azuredeploy.json `
  -storageAccountType Standard_GRS
```

- Pass in a local parameter file:

```
New-AzureRmResourceGroupDeployment -Name ExampleDeployment -ResourceGroupName ExampleResourceGroup `
  -TemplateFile c:\MyTemplates\storage.json `
  -TemplateParameterFile c:\MyTemplates\storage.parameters.json
```

- Pass in an external parameter file

```
New-AzureRmResourceGroupDeployment -Name ExampleDeployment -ResourceGroupName ExampleResourceGroup `
  -TemplateUri https://raw.githubusercontent.com/Azure/azure-quickstart-templates/master/101-storage-account-create/azuredeploy.json `
  -TemplateParameterUri https://raw.githubusercontent.com/Azure/azure-quickstart-templates/master/101-storage-account-create/azuredeploy
```

Deploy Templates - PowerShell

- You can pass in parameters inline and with a local parameter file.
- You CANNOT pass in parameters inline and with an external parameter file.
- You can TEST a deployment before running it.

```
Test-AzureRmResourceGroupDeployment -ResourceGroupName testgroup `
  -TemplateFile c:\MyTemplates\storage.json -storageAccountType badSku
```

```
Code      : InvalidTemplate
```

```
Message : Deployment template validation failed: 'The provided value 'badSku' for the template parameter 'storageAccountType' at line '15' and column '24' is not valid. The parameter value is not part of the allowed value(s): 'Standard_LRS,Standard_ZRS,Standard_GRS,Standard_RAGRS,Premium_LRS'.'. '
```

```
Details :
```

- **Template Deployments have 2 modes: INCREMENTAL or COMPLETE**
 - Both modes deploy resources defined in Template.
 - Complete Deployment DELETES resources in group not defined in Template
 - Incremental does not DELETE resources not defined in Template
 - Use “-Mode” argument to set Mode.

Deploy resources with Resource Manager templates and Azure PowerShell

[Deploy a template from your local machine](#)

[Deploy a template from an external source](#)

[Parameter files](#)

[Test a template deployment](#)

[Incremental and complete deployments](#)

[Sample template](#)

[Next steps](#)

Regional Availability

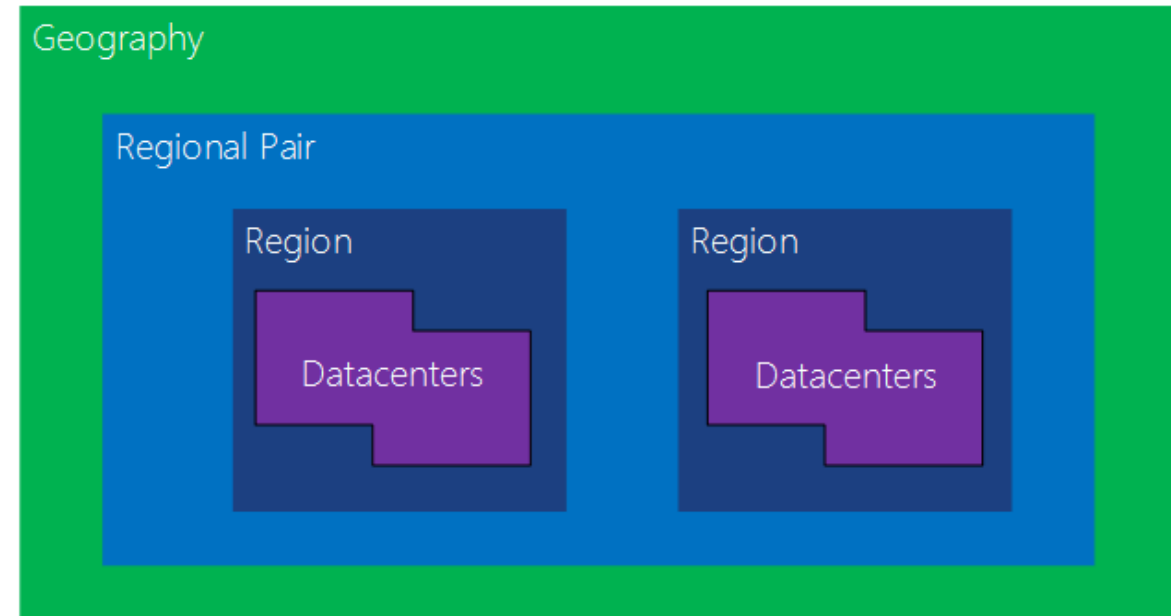
Regions

- An Azure Region is a geographic region that contains a collection of data centers.
- It's more than one building.
- Over 40 Azure regions
- Special / Sovereign Regions



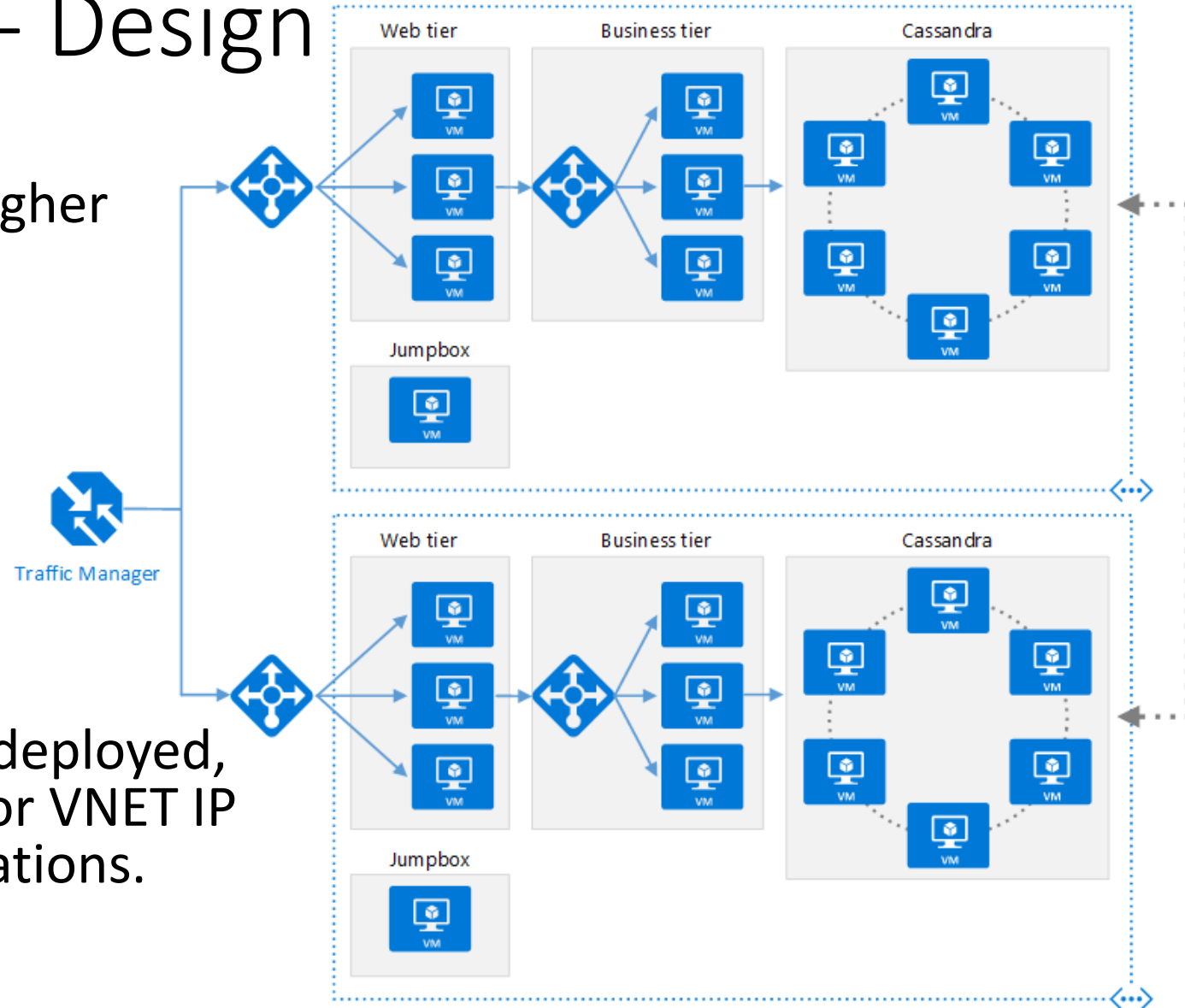
Regional Availability - Regions

- Region Pairs
 - In same geography (such as US, Europe or Asia)
 - Replicated resources are replicated across pairs
 - In broad geographic outage, one region in the pair is prioritized
 - Data resides in the same geography as its pair (except Brazil South)
 - Azure Storage GRS and RA-GRS replicates data from one region to its pair.



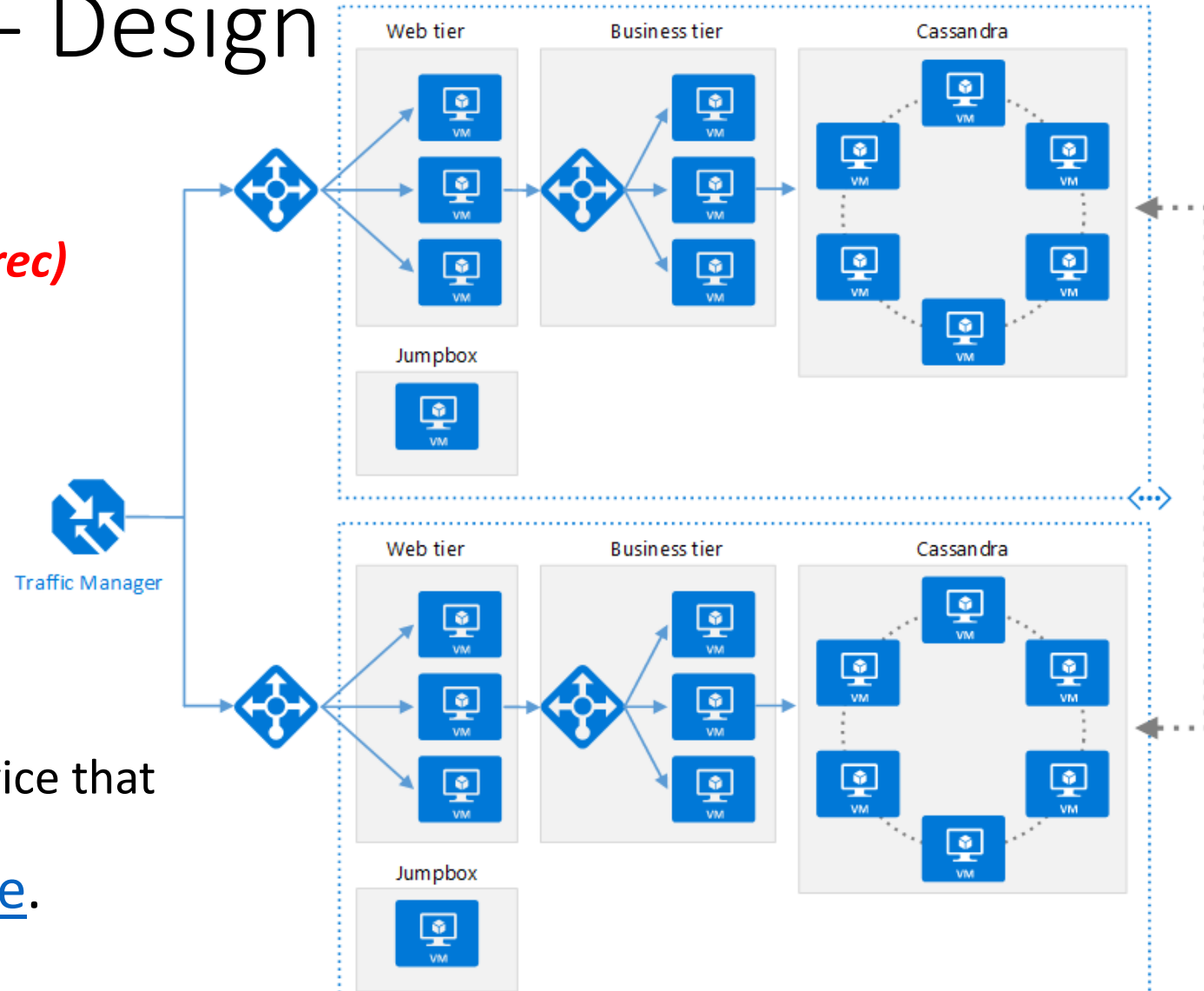
Regional Availability - Design

- Primary/Secondary Region for Higher Availability
- **Traffic Manager** routes requests
- Put Primary, Secondary and Traffic Manager in separate regions
- Make sure each region's VNET IP range does not overlap
- The same ARM template can be deployed, with minor parameter changes for VNET IP address space, and Resource locations.



Regional Availability - Design

- Configurations
 - **Active/passive with hot standby (rec)**
 - Active/passive with cold standby
 - Active/active
- Use Regional Pairing
 - In broad outage, one region from pair is prioritized
 - Data residency issues mitigated
- Traffic Manager
 - routing = priority,
 - health probe = write a custom service that reports accurately
- Read Reference Architecture [Here](https://docs.microsoft.com/en-us/azure/architecture/reference-architectures/virtual-machines-linux/multi-region-application).



High Availability

- Availability within a region vs. Regional Availability
- Azure Storage makes 3 copies of data within a region (including VM disks)
- Use Availability Sets
- Design loosely coupled service with asynchronous communication. Use Azure Storage queues or Azure Service Bus for queue-centric workflow.
- Fault Detection and Retry Logic within your application
- Circuit Breaker Pattern
- Immutable Infrastructure, Infrastructure as Code
- <https://docs.microsoft.com/en-us/azure/architecture/resiliency/high-availability-azure-applications>
- <https://docs.microsoft.com/en-us/azure/architecture/resiliency/index#designing-for-resiliency>

