Your fastest way to learn. Why wait?

FIREBRAND

Microsoft Azure Administrator Associate – Pre-reading

R-661-02

Version 1.0

https://firebrand.training

Table of Contents

Introduction

This Value-Add guide will help you prepare for your next course at Firebrand. This pre-reading material will get you ready to succeed in your exams and get Microsoft Certified.

This guide will explain your exam path and all the topics that you will cover during your training and share some content and links you should review.

Being Azure, it is a moving target when it comes to updates. It is hard to near impossible to keep guides like this always updated. Using a mix of content and official links from Microsoft Docs, helps on keeping this the most accurate guide possible.

MAA – Microsoft Azure Administrator Associate

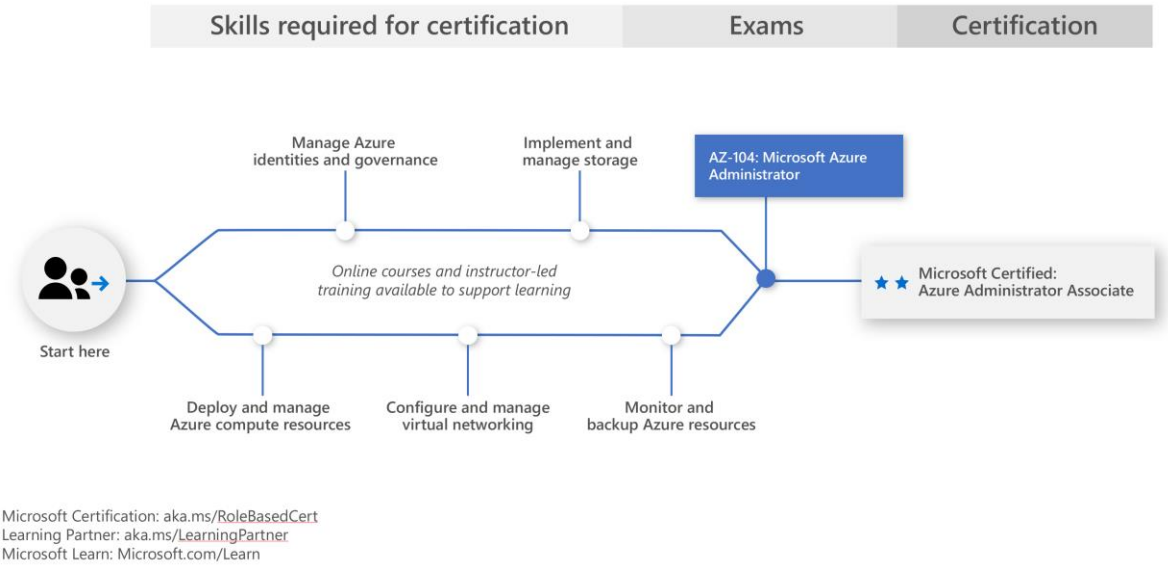
An Azure Administrator implements, manages, and monitors identity, governance, storage, compute, and virtual networks in a cloud environment. The Azure Administrator will provision, size, monitor, and adjust resources as appropriate.

In order to achieve this certification, you need to pass one Microsoft exam:

- Exam AZ-104: Microsoft Azure Administrator - https://docs.microsoft.com/en-us/learn/certifications/exams/az-104

Each exam has a maximum duration of 210 minutes and 40-60 questions in total. The total number of questions depends on the exact exam and can change over time.

Learning path for the Azure Developer Associate certification:



Microsoft Certification: aka.ms/RoleBasedCert
Learning Partner: aka.ms/LearningPartner
Microsoft Learn: Microsoft.com/Learn

Manage Azure identities and governance (15-20%)

Manage Azure AD objects:

- Create users and groups
- Manage user and group properties
- Manage device settings
- Perform bulk user updates
- Manage guest accounts
- Configure Azure AD Join
- Configure self-service password reset

NOT: Azure AD Connect; PIM

Manage role-based access control (RBAC):

- Create a custom role
- Provide access to Azure resources by assigning roles
    - Subscriptions
    - Resource groups
    - Resources (VM, disk, etc.)
- Interpret access assignments
- Manage multiple directories

Manage subscriptions and governance:

- Configure Azure policies
- Configure resource locks
- Apply tags
- Create and manage resource groups
    - Move resources
    - Remove RGs
- Manage subscriptions
- Configure Cost Management
- Configure management groups

Implement and manage storage (10-15%)

Manage storage accounts:

- Configure network access to storage accounts
- Create and configure storage accounts
- Generate shared access signature
- Manage access keys
- Implement Azure storage replication
- Configure Azure AD Authentication for a storage account

Manage data in Azure Storage:

- Export from Azure job
- Import into Azure job
- Install and use Azure Storage Explorer
- Copy data by using AZCopy

Configure Azure files and Azure blob storage:

- Create an Azure file share
- Create and configure Azure File Sync service
- Configure Azure blob storage
- Configure storage tiers for Azure blobs

Deploy and manage Azure compute resources (25-30%)

Configure VMs for high availability and scalability:

- Configure high availability
- Deploy and configure scale sets

Automate deployment and configuration of VMs:

- Modify Azure Resource Manager (ARM) template
- Configure VHD template
- Deploy from template
- Save a deployment as an ARM template
- Automate configuration management by using custom script extensions

Create and configure VMs:

- Configure Azure Disk Encryption
- Move VMs from one resource group to another
- Manage VM sizes
- Add data discs
- Configure networking
- Redeploy VMs

Create and configure containers:

- Create and configure Azure Kubernetes Service (AKS)

- Create and configure Azure Container Instances (ACI)
- NOT: selecting an container solution architecture or product; container registry settings

Create and configure Web Apps:

- Create and configure App Service
- Create and configure App Service Plans
- NOT: Azure Functions; Logic Apps; Event Grid

Configure and manage virtual networking (30-35%)

Implement and manage virtual networking:

- Create and configure VNET peering
- Configure private and public IP addresses, network routes, network interface, subnets, and virtual network

Configure name resolution:

- Configure Azure DNS
- Configure custom DNS settings
- Configure a private or public DNS zone

Secure access to virtual networks:

- Create security rules
- Associate an NSG to a subnet or network interface
- Evaluate effective security rules
- Deploy and configure Azure Firewall
- Deploy and configure Azure Bastion Service
- NOT: Implement Application Security Groups; DDoS

Configure load balancing:

- Configure Application Gateway
- Configure an internal load balancer
- Configure load balancing rules
- Configure a public load balancer
- Troubleshoot load balancing
- NOT: Traffic Manager and FrontDoor and PrivateLink

Monitor and troubleshoot virtual networking:

- Monitor on-premises connectivity
- Use Network Performance Monitor
- Use Network Watcher
- Troubleshoot external networking
- Troubleshoot virtual network connectivity

Integrate an on-premises network with an Azure virtual network:

- Create and configure Azure VPN Gateway
- Create and configure VPNs
- Configure ExpressRoute
- Configure Azure Virtual WAN

Monitor and back up Azure resources (10-15%)

Monitor resources by using Azure Monitor:

- Configure and interpret metrics
    - Analyze metrics across subscriptions
    - Configure Log Analytics
- Implement a Log Analytics workspace
    - Configure diagnostic settings
- Query and analyze logs
    - Create a query
    - Save a query to the dashboard
    - Interpret graphs
- Set up alerts and actions
    - Create and test alerts
    - Create action groups
    - View alerts in Azure Monitor
    - Analyze alerts across subscriptions
- Configure Application Insights
- NOT: Network monitoring

Implement backup and recovery:

- Configure and review backup reports
- Perform backup and restore operations by using Azure Backup Service
- Create a Recovery Services Vault
    - Use soft delete to recover Azure VMs
- Create and configure backup policy
- Perform site-to-site recovery by using Azure Site Recovery
- NOT: SQL or HANA

Microsoft Azure Fundamentals

If you are new to Microsoft Azure is highly recommended to review the information on the following links:

- Azure Fundamentals - https://docs.microsoft.com/en-us/learn/paths/azure-fundamentals/index

If you're already some experience in Microsoft Azure, you can skip it.

Azure Active Directory

An important part of your application's security is authenticating users before they can use it — but authentication is not an easy thing to implement. You need to store user identities and credentials somewhere, implement password management, create a secure authentication handshake, and so on.

Azure Active Directory (Azure AD) provides all of these things and more out of the box. You store your user identities in Azure AD and have users authenticate against it, redirecting them to your application only after they're authenticated. Azure AD takes care of password management, including resolving common scenarios like forgotten passwords.

Since Azure AD is used by millions of applications every day — including the Azure portal, Outlook.com, and Office 365 — it's able to more readily detect and act on malicious behaviour. For instance, if a user were to sign in to an application from a location in Europe and then one minute later sign in from Australia, Azure AD would flag this as malicious behaviour and ask the user for additional credentials through multifactor authentication.

Azure AD SKUs

Azure AD licenses are built on top of your free existing directory. These extra features can provide self-service password reset, enhanced monitoring, MFA, reporting, device management.

Currently we have the following SKUs available:

- Azure Active Directory Free. Provides user and group management, on-premises directory synchronization, basic reports, self-service password change for cloud users, and single sign-on across Azure, Office 365, and many popular SaaS apps
- Azure Active Directory Premium P1. In addition to the free features, P1 also lets your hybrid users access both on-premises and cloud resources. It also supports advanced administration, such as dynamic groups, self-service group management, Microsoft Identity Manager (an on-premises identity and access management suite) and cloud write-back capabilities, which allow self-service password reset for your on-premises users
- Azure Active Directory Premium P2. In addition to the Free and P1 features, P2 also offers Azure Active Directory Identity Protection to help provide risk-based Conditional Access to your apps and critical company data and Privileged Identity Management to help discover, restrict, and monitor administrators and their access to resources and to provide just-in-time access when needed

- "Pay as you go" feature licenses. You can also get additional feature licenses, such as Azure Active Directory Business-to-Customer (B2C). B2C can help you provide identity and access management solutions for your customer-facing apps. For more information, see Azure Active Directory B2C documentation

Useful Links:

What is Azure Active Directory? - https://docs.microsoft.com/en-us/azure/active-directory/fundamentals/active-directory-whatis

Add or delete users using Azure Active Directory - https://docs.microsoft.com/en-us/azure/active-directory/fundamentals/add-users-azure-active-directory

Creating a new user in Azure AD - https://docs.microsoft.com/en-us/powershell/azure/active-directory/new-user-sample

Azure Active Directory self-service password reset - https://docs.microsoft.com/en-us/azure/active-directory/authentication/howto-sspr-deployment

Azure PowerShell

Azure PowerShell provides a set of cmdlets that use the Azure Resource Manager model for managing your Azure resources. Azure PowerShell uses .NET Standard, making it available for Windows, macOS, and Linux. Azure PowerShell is also available from Azure Cloud Shell allowing you to run PowerShell commands directly from the browser. Azure Cloud Shell runs in a container on Azure, this way you can run against the latest versions of Azure PowerShell modules without the need to have it installed locally.

Exercise: Get started with Azure PowerShell

https://docs.microsoft.com/en-us/powershell/azure/get-started-azureps?view=azps-1.0.0

Exercise: Automate Azure Tasks using scripts with PowerShell

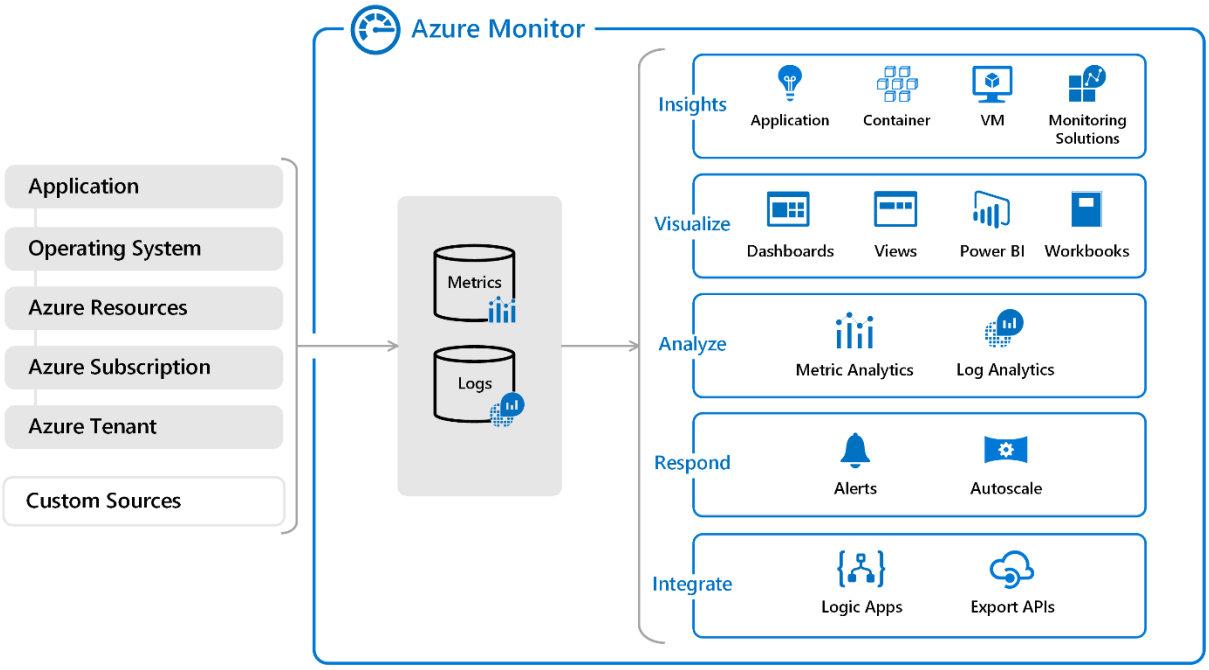
https://docs.microsoft.com/en-us/learn/modules/automate-azure-tasks-with-powershell/

Azure Virtual Machine PowerShell samples:

https://docs.microsoft.com/en-gb/azure/virtual-machines/windows/powershell-samples?toc=%2Fpowershell%2Fazure%2Ftoc.json

Azure Monitor

Azure Monitor enables basic monitoring for Azure services by collecting metrics, activity logs, and diagnostic logs. The metrics collected provide performance statistics for different resources, including the OS associated with a VM.



The activity log will show you when new resources are created or modified. You can view this data with one of the explorers in the Azure portal and send it to Log Analytics for trending and detailed analysis, or you can create alert rules that will proactively notify you of critical issues.

Azure Log Analytics helps you collect and analyze data generated by resources in your cloud and on-premises environments. It provides real-time insights by using integrated search and custom dashboards to analyze millions of records across all your workloads and servers regardless of their physical location.

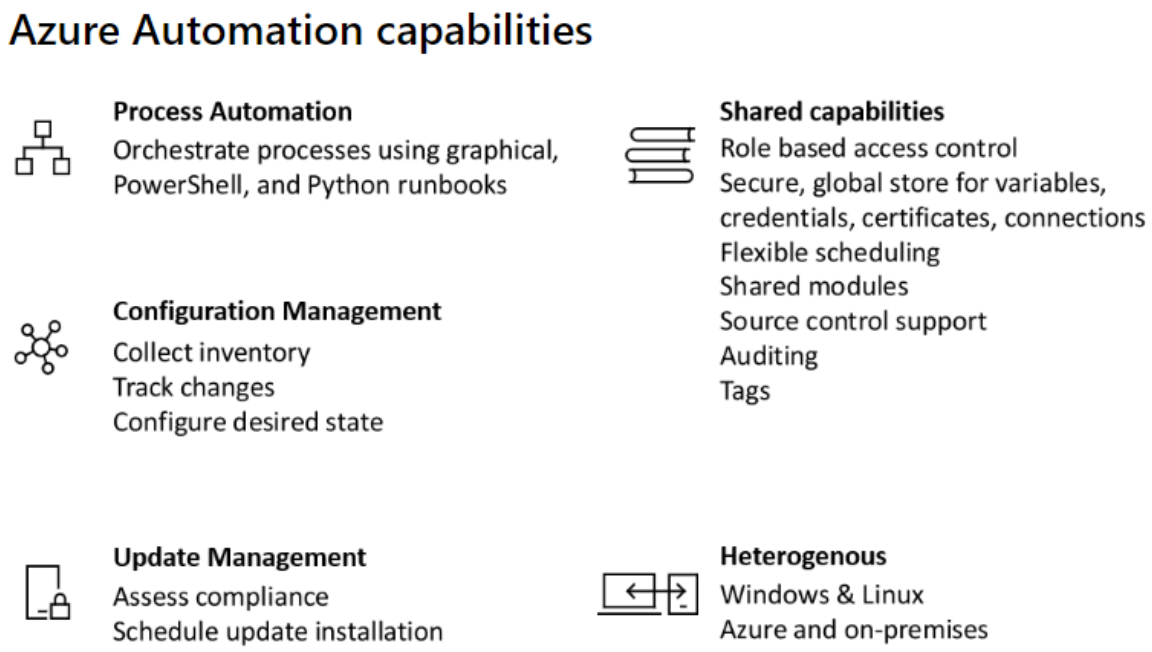
Useful Links:

More information at: https://docs.microsoft.com/en-us/azure/azure-monitor/

Monitoring solutions in Azure Monitor - https://docs.microsoft.com/en-us/azure/azure-monitor/insights/solutions

Azure Automation

Azure Automation delivers a cloud-based automation and configuration service that provides consistent management across your Azure and non-Azure environments. It consists of process automation, update management, and configuration features. Azure Automation provides complete control during deployment, operations, and decommissioning of workloads and resources.



Build/Deploy resources:

- Deploy VMs across a hybrid environment using Runbooks and Azure Resource Manager templates. Integrate into development tools like Jenkins and Azure DevOps

Configure VMs:

- Assess and configure Windows and Linux machines with the desired configuration for the infrastructure and application

Monitor:

- Identify changes on machines that are causing issues and remediate or escalate to management systems

Protect:

- Quarantine VM if security alert is raised. Set in-guest requirements

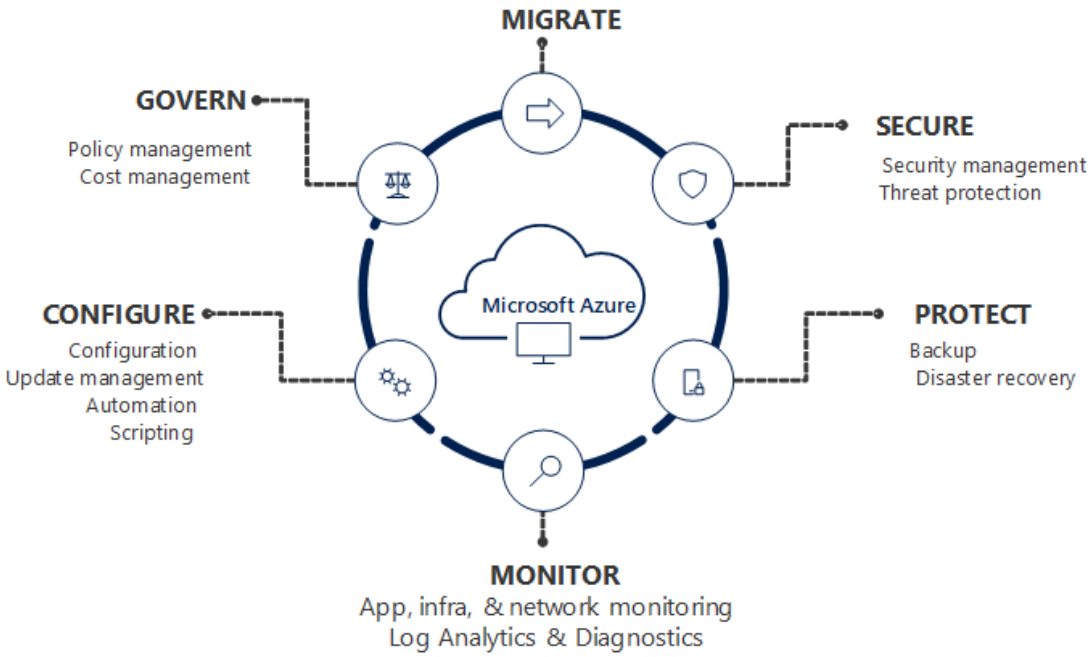
Govern:

- Set up role-based access control for teams. Recover unused resources

Manage Subscriptions and Governance

Azure Management

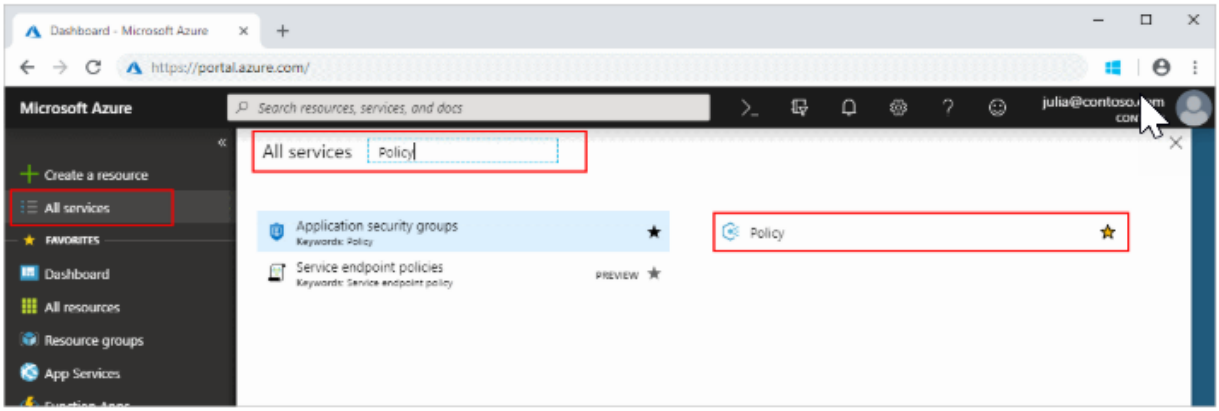
The following diagram illustrates the different areas of management that are required to maintain any application or resource. The different areas can be thought of as a lifecycle. Each area is required in continuous succession over the lifespan of a resource. This resource lifecycle starts with the initial deployment, through continued operation, and finally when retired.



No single Azure service completely fills the requirements of a particular management area. Instead, each is realized by several working toghether. Some service, such as Application Insights, provide targeted monitoring functionality for web applications. Others, like Log Analytics, store management data for other services. This feature allows you to analyze data of different types collected by different services.

Azure Policy

Azure Policy is a service in Azure that you use to create, assign and, manage policies. These policies enforce different rules and effects over your resources, so those resources stay compliant with your corporate standards and service level agreements. Azure Policy meets this need by evaluating your resources for non-compliance with assigned policies. For example, you can have a policy to allow only a certain SKU size of virtual machines in your environment. Once this policy is implemented, new and existing resources are evaluated for compliance. With the right type of policy, existing resources can be brought into compliance.



Useful Links:

Create a policy assignment to identify non-compliant resources - https://docs.microsoft.com/en-us/azure/governance/policy/assign-policy-portal

Create and manage policies to enforce compliance - https://docs.microsoft.com/en-us/azure/governance/policy/tutorials/create-and-manage

Resource Locks

As an administrator, you may need to lock a subscription, resource group, or resource to prevent other users in your organization from accidentally deleting or modifying critical resources. You can set the lock level to CanNotDelete or ReadOnly. In the portal, the locks are called Delete and Read-only respectively.

- CanNotDelete means authorized users can still read and modify a resource, but they can't delete the resource
- ReadOnly means authorized users can read a resource, but they can't delete or update the resource. Applying this lock is similar to restricting all authorized users to the permissions granted by the Reader role

When you apply a lock at a parent scope, all resources within that scope inherit the same lock. Even resources you add later inherit the lock from the parent. The most restrictive lock in the inheritance takes precedence.

Useful Links:

Lock resources to prevent unexpected changes - https://docs.microsoft.com/en-us/azure/azure-resource-manager/management/lock-resources

Resource Tags

Tags in Azure provide you with a means to apply additional metadata to your resources. This is a powerful feature that you can use as part of your Azure governance implementation and continual management. Tags give you an additional layer of organization to compliment the naming standards, Cost Management and other requirements.

A tag is like a label that can be attached to a Resource Group or individual resources, for example a cost center or business unit. It is mainly thanks to these tags that an Azure billing administrator can get a clear view of what an Azure resource is used for, or at least to which business unit or cost center this resource belongs.

Each resource in Azure can have up to 50 tags. Some examples of common tags are:

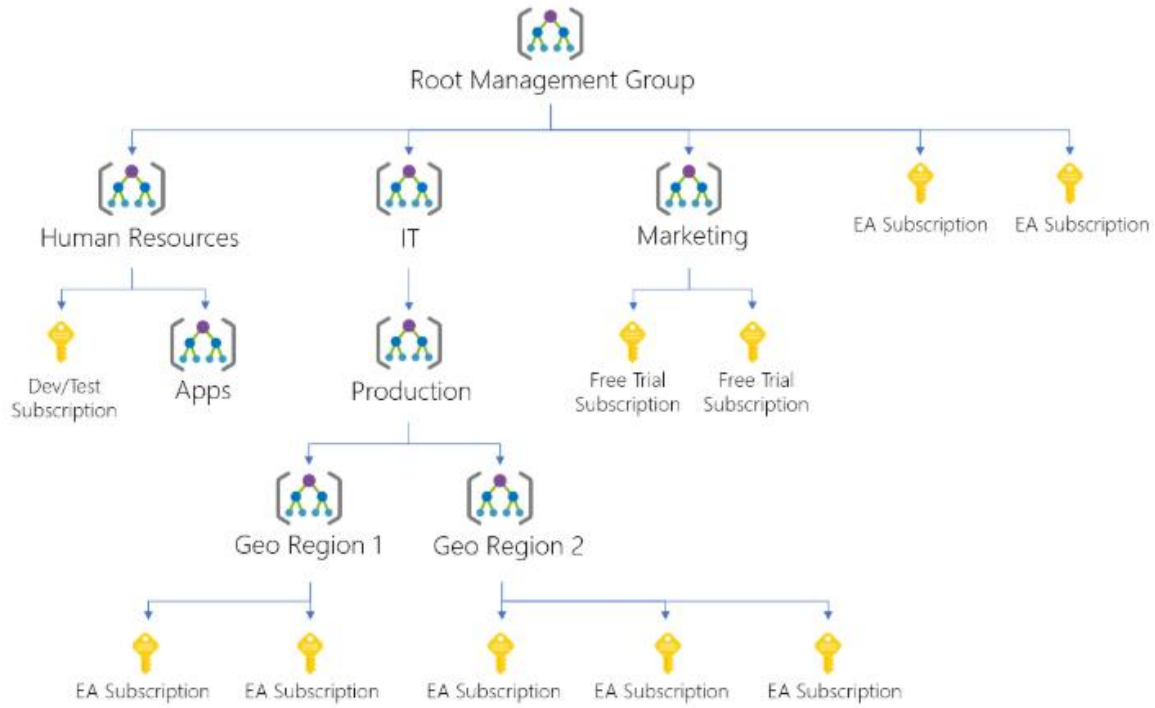
- Environment (Dev, Test, QA, Prod, …)
- Cost Center
- Department

Useful Links:

Use tags to organize your Azure resources and management hierarchy - https://docs.microsoft.com/en-us/azure/azure-resource-manager/management/tag-resources

Management Groups

If your organization has many subscriptions, you may need a way to efficiently manage access, policies, and compliance for those subscriptions. Azure management groups provide a level of scope above subscriptions. You organize subscriptions into containers called "management groups" and apply your governance conditions to the management groups. All subscriptions within a management group automatically inherit the conditions applied to the management group. Management groups give you enterprise-grade management at a large scale no matter what type of subscriptions you might have.



One area where you would use management groups is to provide user access to multi subscriptions. By moving many subscriptions under that management group, you can create one role-based access control (RBAC) assignment on the management group, which will inherit that access to all the subscriptions. One assignment on the management group can enable users to have access to everything they need instead of scripting RBAC over different subscriptions.

Useful Links:

Create management groups for resource organization and management - https://docs.microsoft.com/en-us/azure/governance/management-groups/create

Manage your resources with management groups - https://docs.microsoft.com/en-us/azure/governance/management-groups/manage

Manage Subscriptions

Azure subscriptions have controls available that govern access to the resources within a subscription, govern cost through quotas and tagging, and govern the resources that are allowed in an environment with Azure policy.

You can get an Azure subscription from different ways:

- Free trial
- Pay-as-you-go
- MSDN subscriptions
- Microsoft resellers
- CSP – Cloud Solution Providers
- Microsoft Open licensing
- Enterprise Agreements

Useful Links:

Create an additional Azure subscription - https://docs.microsoft.com/en-us/azure/cost-management-billing/manage/create-subscription

Change your Azure subscription to a different offer - https://docs.microsoft.com/en-us/azure/cost-management-billing/manage/switch-azure-offer

Cost Management

In Azure, you only pay for what you use. As you create and use Azure resources, you are charged for the resources. You use Azure Cost Management and Billing features to conduct billing administrative tasks and manage billing access to costs. You can also use its features to monitor and control Azure spending and to optimize Azure resource use.

Cloudyn is an Azure service that allows you to track cloud usage and expenditures for your Azure resources. Existing Cloudyn features are being integrated directly into the Azure portal wherever possible and is being deprecated by the end of 2020.

Useful Links:

What is Azure Cost Management and Billing? - https://docs.microsoft.com/en-us/azure/cost-management-billing/cost-management-billing-overview

Explore and analyse costs with cost analysis - https://docs.microsoft.com/en-us/azure/cost-management-billing/costs/quick-acm-cost-analysis

Azure Storage

Azure Storage is one of the oldest, most reliable, and most performant services in Azure. Azure Storage offers five types of storage that all benefit from the following shared features:

- Geo-redundancy, which replicates data to different regions so you can recover it if a disaster causes an individual region to fail
- Encryption of data at runtime
- Custom domains

The five Azure Storage types are:

- Blob (Containers):
  - Unstructured Large Page/Block
- Table:
  - Semi-structured flexible schema
- Queue:
  - Queue reliable MSMQ
- File Share:
  - SMB File Share
- Disk:
  - Premium High I/O VM Disks

Azure Blob storage stores large, unstructured data—literally, blobs of data. This data can be video, image, audio, text, or even virtual hard drive (VHD) files for VMs.

There are three types of blobs:

- Page:
  - Optimized for random read and write operations and are perfect for storing a VHD
- Append:
  - Optimized for append operations, such as storing operation logs that can't be updated or deleted
- Block:
  - Optimized for efficiently uploading large amounts of data. These are perfect for storing large video files that don't change often

Useful Links:

Store data in Azure - https://docs.microsoft.com/en-us/learn/paths/store-data-in-azure/

Azure storage account overview https://docs.microsoft.com/en-us/azure/storage/common/storage-account-overview

Create a storage account https://docs.microsoft.com/en-us/azure/storage/common/storage-quickstart-create-account?tabs=azure-portal

Azure Storage Service Encryption for data at rest https://docs.microsoft.com/en-us/azure/storage/common/storage-service-encryption

Azure Storage security overview https://docs.microsoft.com/en-us/azure/security/security-storage-overview

Azure Storage security guide https://docs.microsoft.com/en-us/azure/storage/common/storage-security-guide#using-client-side-encryption-to-secure-data-that-you-send-to-storage

Azure Table Storage

Azure Table storage is an inexpensive, extremely fast NoSQL key-value store you can use to store data in flexible tables. A table can contain one row describing an order and another row describing customer information. You don't need to define a data schema, making Table storage very flexible.

Table storage - https://docs.microsoft.com/en-us/azure/cosmos-db/table-storage-overview

Azure Virtual Machines

When using the IaaS model, we have greater control deploying compute infrastructure to support our workloads. We usually choose IaaS instead of PaaS when we need the extra control or when we need to interact and host legacy workloads. Using IaaS, you deploy your own virtual machines where you will deploy all the needed components for solution – Load Balancers, Firewalls, Virtual Network, and Network Security.

Azure provides all the hardware to run Virtual Machines but you're still responsible for all the configuration, manage administrative access, upgrades and patching.

Provision VMs

Deploying a Virtual Machine in Azure is a very basic process. You can achieve this using:

- Azure Portal - https://docs.microsoft.com/en-us/azure/virtual-machines/windows/quick-create-portal
- Azure PowerShell - https://docs.microsoft.com/en-us/azure/virtual-machines/windows/quick-create-powershell
- Azure CLI - https://docs.microsoft.com/en-us/azure/virtual-machines/windows/quick-create-cli
- ARM Templates

Azure supports Windows and Linux operating systems. When creating the Virtual Machine, we choose the OS image from the Azure Marketplace. If you can't find an image that suits you, you can always create your own custom images. Check if your image OS is supported on Azure.

Windows is supported from Windows Server 2008 up to 2019.

For Linux support refer to this table:

Distribution	Version	Drivers	Agent
CentOS	CentOS 6.3+, 7.0+, 8.0+	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+, 9, 10	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+, 8.0+	In kernel	Package: In repo under "WALinuxAgent" Source code: GitHub
SUSE Linux Enterprise	SLES/SLES for SAP 11 SP4 12 SP1+ 15	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 42.2+	In kernel	Package: In Cloud:Tools repo under "python-azure-agent" Source code: GitHub
Ubuntu	Ubuntu 12.04+ [1]	In kernel	Package: In repo under "walinuxagent" Source code: GitHub

After you choose the Operating System there are other important options to select, for example:

- Subscription
- Resource Group
- Name
- Region
- Size
- Administration Account
- Inbound port rules
- Network Configuration
- Disks
- Management Option

Create a virtual machine

Basics | Disks | Networking | Management | Advanced | Tags | Review + create

Create a virtual machine that runs Linux or Windows. Select an image from Azure marketplace or use your own customized image.
Complete the Basics tab then Review + create to provision a virtual machine with default parameters or review each tab for full customization.

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * : Azure - Training

Resource group * :

Create new

Instance details

Virtual machine name * :

Region * : (US) Central US

Availability options : No infrastructure redundancy required

Image * : Ubuntu Server 18.04 LTS

Browse all public and private images

Azure Spot instance : Yes / No

Size * :

Standard D2s v3
2 vcpus, 8 GiB memory (€67.72/month)
Change size

Administrator account

Authentication type : SSH public key / Password

Username * :

SSH public key * :

Learn more about creating and using SSH keys in Azure

Inbound port rules

Select which virtual machine network ports are accessible from the public internet. You can specify more limited or granular network access on the Networking tab.

Public inbound ports * : None / Allow selected ports

Select inbound ports * : SSH (22)

This will allow all IP addresses to access your virtual machine. This is only recommended for testing. Use the Advanced controls in the Networking tab to create rules to limit inbound traffic to known IP addresses.

Quick start: Create a Windows virtual machine in the Azure portal - https://docs.microsoft.com/en-us/azure/virtual-machines/windows/quick-create-portal

Tutorial: Create and Manage Windows VMs with Azure PowerShell - https://docs.microsoft.com/en-us/azure/virtual-machines/windows/tutorial-manage-vm

ARM templates

One of the biggest advantages of using Azure IaaS is the level of automation that you can achieve when deploying new services, resources, or infrastructure. One of the main reasons you can do this is because Microsoft provides you the Azure Resource Manager (ARM), which is the deployment and management service in Azure. The ARM service oversees creating, updating, and deleting the different kind of services you can deploy in your subscription.

All actions offered by the ARM service are exposed through the same API. This means that no matter which mechanism you use—portal, PowerShell, Azure CLI, Rest API, or client SDKs—you will get a consistent behaviour and result when interacting with ARM.

Here is an example of an ARM template to deploy a simple Windows Virtual Machine:

```
{
  "$schema": "https://schema.management.azure.com/schemas/2019-04-
01/deploymentTemplate.json#",
  "contentVersion": "1.0.0.0",
  "parameters": {
    "adminUsername": {
      "type": "string",
      "metadata": {
        "description": "Username for the Virtual Machine."
      }
    },
    "adminPassword": {
      "type": "securestring",
      "metadata": {
        "description": "Password for the Virtual Machine."
      }
    },
    "dnsLabelPrefix": {
      "type": "string",
      "metadata": {
        "description": "Unique DNS Name for the Public IP used to access the Virtual Machine."
      }
```

```
  },
  "windowsOSVersion": {
   "type": "string",
   "defaultValue": "2016-Datacenter",
   "allowedValues": [
    "2008-R2-SP1",
    "2012-Datacenter",
    "2012-R2-Datacenter",
    "2016-Nano-Server",
    "2016-Datacenter-with-Containers",
    "2016-Datacenter",
    "2019-Datacenter"
   ],
   "metadata": {
    "description": "The Windows version for the VM. This will pick a fully patched image of this given Windows version."
   }
  },
  "vmSize": {
   "type": "string",
   "defaultValue": "Standard_A2_v2",
   "metadata": {
    "description": "Size of the virtual machine."
   }
  },
  "location": {
   "type": "string",
   "defaultValue": "[resourceGroup().location]",
   "metadata": {
    "description": "Location for all resources."
   }
  }
 },
 "variables": {
  "storageAccountName": "[concat(uniquestring(resourceGroup().id), 'sawinvm')]",
  "nicName": "myVMNic",
  "addressPrefix": "10.0.0.0/16",
  "subnetName": "Subnet",
  "subnetPrefix": "10.0.0.0/24",
  "publicIPAddressName": "myPublicIP",
  "vmName": "SimpleWinVM",
  "virtualNetworkName": "MyVNET",
  "subnetRef": "[resourceId('Microsoft.Network/virtualNetworks/subnets', variables('virtualNetworkName'), variables('subnetName'))]",
  "networkSecurityGroupName": "default-NSG"
```

```
},
"resources": [
 {
  "type": "Microsoft.Storage/storageAccounts",
  "apiVersion": "2018-11-01",
  "name": "[variables('storageAccountName')]",
  "location": "[parameters('location')]",
  "sku": {
   "name": "Standard_LRS"
  },
  "kind": "Storage",
  "properties": {}
 },
 {
  "type": "Microsoft.Network/publicIPAddresses",
  "apiVersion": "2018-11-01",
  "name": "[variables('publicIPAddressName')]",
  "location": "[parameters('location')]",
  "properties": {
   "publicIPAllocationMethod": "Dynamic",
   "dnsSettings": {
    "domainNameLabel": "[parameters('dnsLabelPrefix')]"
   }
  }
 },
 {
  "comments": "Default Network Security Group for template",
  "type": "Microsoft.Network/networkSecurityGroups",
  "apiVersion": "2019-08-01",
  "name": "[variables('networkSecurityGroupName')]",
  "location": "[parameters('location')]",
  "properties": {
   "securityRules": [
    {
     "name": "default-allow-3389",
     "properties": {
      "priority": 1000,
      "access": "Allow",
      "direction": "Inbound",
      "destinationPortRange": "3389",
      "protocol": "Tcp",
      "sourcePortRange": "*",
      "sourceAddressPrefix": "*",
      "destinationAddressPrefix": "*"
```

```
        }
       }
      ]
     }
    },
    {
     "type": "Microsoft.Network/virtualNetworks",
     "apiVersion": "2018-11-01",
     "name": "[variables('virtualNetworkName')]",
     "location": "[parameters('location')]",
     "dependsOn": [
      "[resourceId('Microsoft.Network/networkSecurityGroups',
variables('networkSecurityGroupName'))]"
     ],
     "properties": {
      "addressSpace": {
       "addressPrefixes": [
        "[variables('addressPrefix')]"
       ]
      },
      "subnets": [
       {
        "name": "[variables('subnetName')]",
        "properties": {
         "addressPrefix": "[variables('subnetPrefix')]",
         "networkSecurityGroup": {
          "id": "[resourceId('Microsoft.Network/networkSecurityGroups',
variables('networkSecurityGroupName'))]"
         }
        }
       }
      ]
     }
    },
    {
     "type": "Microsoft.Network/networkInterfaces",
     "apiVersion": "2018-11-01",
     "name": "[variables('nicName')]",
     "location": "[parameters('location')]",
     "dependsOn": [
      "[resourceId('Microsoft.Network/publicIPAddresses/', variables('publicIPAddressName'))]",
      "[resourceId('Microsoft.Network/virtualNetworks/', variables('virtualNetworkName'))]"
     ],
     "properties": {
      "ipConfigurations": [
```

```
      {
       "name": "ipconfig1",
       "properties": {
        "privateIPAllocationMethod": "Dynamic",
        "publicIPAddress": {
         "id":
"[resourceId('Microsoft.Network/publicIPAddresses',variables('publicIPAddressName'))]"
        },
        "subnet": {
         "id": "[variables('subnetRef')]"
        }
       }
      }
     ]
    }
   },
   {
    "type": "Microsoft.Compute/virtualMachines",
    "apiVersion": "2018-10-01",
    "name": "[variables('vmName')]",
    "location": "[parameters('location')]",
    "dependsOn": [
     "[resourceId('Microsoft.Storage/storageAccounts/', variables('storageAccountName'))]",
     "[resourceId('Microsoft.Network/networkInterfaces/', variables('nicName'))]"
    ],
    "properties": {
     "hardwareProfile": {
      "vmSize": "[parameters('vmSize')]"
     },
     "osProfile": {
      "computerName": "[variables('vmName')]",
      "adminUsername": "[parameters('adminUsername')]",
      "adminPassword": "[parameters('adminPassword')]"
     },
     "storageProfile": {
      "imageReference": {
       "publisher": "MicrosoftWindowsServer",
       "offer": "WindowsServer",
       "sku": "[parameters('windowsOSVersion')]",
       "version": "latest"
      },
      "osDisk": {
       "createOption": "FromImage"
      },
```

```
      "dataDisks": [
       {
        "diskSizeGB": 1023,
        "lun": 0,
        "createOption": "Empty"
       }
      ]
     },
     "networkProfile": {
      "networkInterfaces": [
       {
        "id": "[resourceId('Microsoft.Network/networkInterfaces',variables('nicName'))]"
       }
      ]
     },
     "diagnosticsProfile": {
      "bootDiagnostics": {
       "enabled": true,
       "storageUri": "[reference(resourceId('Microsoft.Storage/storageAccounts/',
variables('storageAccountName'))).primaryEndpoints.blob]"
      }
     }
    }
   }
 ],
 "outputs": {
  "hostname": {
   "type": "string",
   "value": "[reference(variables('publicIPAddressName')).dnsSettings.fqdn]"
  }
 }
}
```

A good starting point when building ARM templates is the GitHub repo for ARM template quick start. You can find several templates in this git repo. Reference: https://github.com/Azure/azure-quickstart-templates.

To understand how to build ARM template check the reference documentation on docs: https://docs.microsoft.com/en-us/azure/azure-resource-manager/templates/template-syntax

Extend Azure Resource Manager template functionality - https://docs.microsoft.com/en-us/azure/architecture/building-blocks/extending-templates

Azure Resource Manager templates overview - https://docs.microsoft.com/en-us/azure/azure-resource-manager/templates/overview

Tutorial: Create and deploy your first Azure Resource Manager template - https://docs.microsoft.com/en-us/azure/azure-resource-manager/templates/template-tutorial-create-first-template

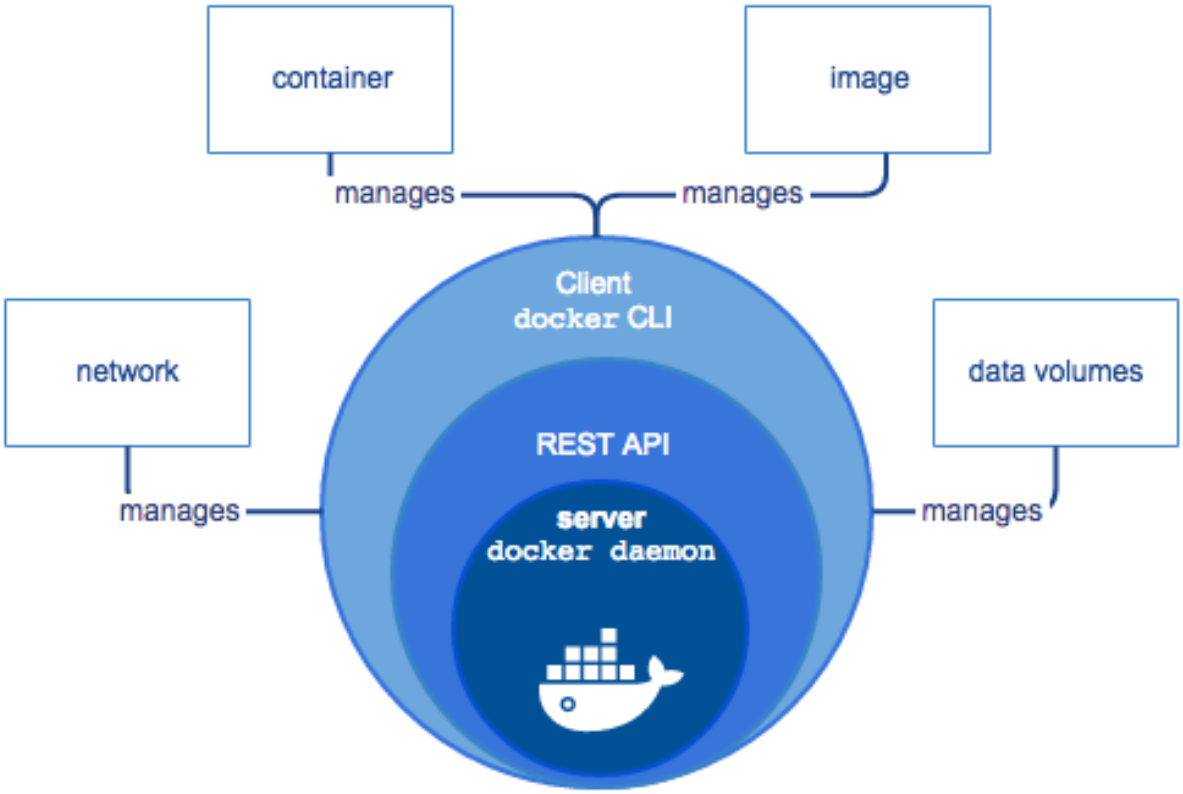
Compute Containers

While much more lightweight, containers are similar to VMs, and you can start and stop them in a few seconds. Containers also offer tremendous portability, which makes them ideal for developing an app locally on your machine and then hosting it in the cloud, in test, and later in production.

You can even run containers on-premises or in other clouds—the environment that you use on your development machine travels with your container, so your app always runs in the same ecosystem.

Introduction to Docker

Docker is an open platform for developing, shipping, and running applications. Docker enables you to separate your applications from your infrastructure so you can deliver software quickly. With Docker, you can manage your infrastructure in the same ways you manage your applications. By taking advantage of Docker's methodologies for shipping, testing, and deploying code quickly, you can significantly reduce the delay between writing code and running it in production.



Commands for Containers

Docker provide commands that let the user manage any containers that are on the given machine:

Command	Functionality
docker ps	Shows all running containers
docker ps -a	Shows all containers
docker stop <container name>	Stops running container with the given name
docker stop $(docker ps -a -q)	Stops all running containers
docker rm $(docker ps -a)	Deletes all containers

Commands for Images

Likewise, we can manage the images on our server using similar commands:

Command	Functionality
docker pull <image name>	Pulls a specified Docker image from DockerHub
docker run <image name>	Starts a container based on the image that was specified (pulls it automatically if not available locally but on DockerHub)
docker run --name <image name>	Starts a container based on the specified image and assigns it the specified name
docker run -d <image name>	Starts a container based on the specified image as a background process (it is not logged to current console)
docker images	Lists all intermediate images
docker images <image name>	Lists all images with the given name
docker image prune	Deletes all images that are not being used by a container

For more information about Docker refer to:
https://docs.docker.com/engine/docker-overview/

Azure Container Registry (ACR)

Once you've created a container image to run your application in, you can store that container in Azure Container Registry (ACR). This is a highly available and secure storage service, specifically built to store container images. This is great for storing your private Docker images.

You can also use ACR for your existing container development and deployment pipelines. Use ACR Build to build container images in Azure. You can either build on demand or fully automate builds with source code commit and base image update build triggers.

More information at: https://docs.microsoft.com/en-us/azure/container-registry/

Push you first image to ACR - https://docs.microsoft.com/en-us/azure/container-registry/container-registry-get-started-docker-cli

Azure Container Instances (ACI)

You can host your container using Azure Container Instances (ACI). ACI provides fast, isolated compute to meet traffic that comes in spikes, without the need to manage servers.

For example, Azure Kubernetes Service (AKS) can use the Virtual Kubelet to provision pods inside ACI that start in seconds. This enables AKS to run with just enough capacity for an average workload. As you run out of capacity in your ACS cluster, you can scale out additional pods in ACI without any additional servers to manage. The ACI service is billed per second, per virtual CPU, per gigabyte, or by memory.

More information at: https://docs.microsoft.com/en-us/azure/container-instances/

Microsoft Learn: Run Docker containers with Azure Container Instances - https://docs.microsoft.com/en-us/learn/modules/run-docker-with-azure-container-instances/

Azure App Service Web Apps

Azure App Service is one of the key services in Azure that you can use to host your applications. Each of these services brings unique capabilities to the table, but they all share some common features:

Scaling:

- Azure App Service runs on App Service plans, which are abstractions from virtual machines (VMs). One or more VMs run your Azure App Service, but since Azure takes care of them, it's not necessary for you to know which ones. You can, however, scale the resources that run your Azure App Service
- You can either choose a higher pricing tier (ranging from free to premium) or increase the number of application instances that are running. It's even possible to have even have Azure App Service automatically scale the number of instances for you, based on a schedule or metrics like CPU, memory, or HTTP queue length

Deployment slots:

- After deploying a new version of your application to a deployment slot, you can test whether it works as expected and then move it into your production slot
- You can even use Azure's Testing in Production feature to route a percentage of traffic from your production app to a deployment slot. For example, if you shunt 10 percent of your users to the new version of your app in the deployment slot, you can see whether the new features are functioning as expected and whether users are using them
- When you're satisfied with how the new version of your app is performing in the deployment slot, you can carry out a "swap," which exchanges the app in the deployment slot with that in your production slot. You can also swap from a development slot to a staging slot, and then to the production slot. Before doing this, the swap operation verifies that the new version of your website is warmed up and ready to go. When this has been confirmed, the swap operation switches the slots, and your users now see the new version of the app—with no downtime. You can also swap back and revert the deployment of the new version
- You use deployment slots within environments, such as development, test, or production. You don't use deployment slots as environments, because they all reside in the same App Service plan. Those should be separated for security, scaling, billing, and performance. You can swap deployment slots manually through the Azure command-line interface (CLI) and through the Azure Management API. This allows tools like Azure DevOps to perform swap operations during a release

Continuous Deployment:

- To publish your application to App Service, you can use services such as Jenkins, Octopus Deploy, and more. You also can use the Continuous Deployment (CD) feature from Azure DevOps in App Service. This makes it possible for you to create a build-test-release pipeline right in App Service

The process does the following:

1. Retrieves the latest source code from the repository that you indicate
2. Builds the code according to a template that you pick (ASP.NET, Node.js, and so on)
3. Deploys the app in a staging environment and load-tests it
4. Deploys the app to production after approval (you can indicate whether you want to use a deployment slot)

Connect to on-premises resources:

- You can connect external resources like data stores to your App Services. These resources don't need to be located in Azure; they can be anywhere, such as on-premises or in your own datacenter. Depending on your requirements, you can connect to services on-premises through many mechanisms, such as Azure Hybrid Connections, Azure Virtual Networks, and Azure ExpressRoute

Custom domains and Azure App Service certificates:

- When you spin up an app in Azure App Service, it exposes a URL—for example, https://myazurewebsite.azurewebsites.net. Most likely, you will want to use your own custom domain, which you do by mapping that domain name to App Services. Here's how to do that. Additionally, you can ensure that your application is served over HTTPS by using a Secure Sockets Layer (SSL) certificate. Just bring your own certificate or buy one directly from the Azure portal. When you buy an SSL certificate from the Azure portal, you buy an Azure App Service certificate. You can configure this to be used by your custom domain bindings

Automatic OS and development framework patching:

- Because you're using a fully managed platform, you don't manage your own infrastructure at all and benefit from automatic operating system (OS) and framework patching

More information at: https://docs.microsoft.com/en-us/azure/app-service/

Quickstart: Create an ASP.NET Core web app in Azure - https://docs.microsoft.com/en-us/azure/app-service/app-service-web-get-started-dotnet

Deploy Web App - https://docs.microsoft.com/en-us/azure/devops/pipelines/targets/webapp?view=azure-devops&tabs=yaml

Custom Domains - https://docs.microsoft.com/en-us/azure/app-service/manage-custom-dns-buy-domain

Scaling App Service - https://docs.microsoft.com/en-us/azure/app-service/manage-scale-up

Configure App Service - https://docs.microsoft.com/en-us/azure/app-service/configure-common

Virtual Networks

When you start to setup a new branch office, you will make sure that cabling, switches and routers are in place. All this before on-board servers and users. Azure is no different. You need to setup a Virtual Network before bring IaaS workloads like Virtual Machines, Load Balancer.

The Virtual Network of VNet in Azure provides the foundation for the Azure networking infrastructure. Virtual machines are connected to virtual networks. This connection provides inbound and outbound connectivity, to other virtual machines, to on-premises networks, and to the Internet. Azure provides many networking features which will be familiar to those already experienced in networking, such as the ability to control which network flows are permitted and to control network routing. This allows Azure deployments to implement familiar network architectures, such as network segmentation between layers of an N-tier application.

A virtual network (VNet) is an Azure resource. When creating a VNet, the most important setting to choose is the IP range (or ranges) the VNet will use.

IP ranges are defined using Classless Inter-Domain Routing (CIDR) notation. For example, the range 10.5.0.0/16 represents all IP ranges starting with 10.5 (the /16 indicates that the first 16 bits of the IP address given are fixed, while the remaining bits are variable across the IP range being defined). Each virtual network can use either a single IP range, or multiple disjoint IP ranges.

Your VNet IP ranges will typically be taken from the private address ranges defined in RFC 1918. These IP ranges are:

- 10.0.0.0 - 10.255.255.255 (10.0.0.0/8)
- 172.16.0.0 - 172.31.255.255 (172.16.0.0/12)
- 192.168.0.0 - 192.168.255.255 (192.168.0.0/16)

Subnets:

- Subnets are used to divide the VNet IP space. Different subnets can have different network security and routing rules, enabling applications and application tiers to be isolated, and network flows between them controlled. For example, consider a typical 3-tier application architecture comprising a web tier, an application tier and a database tier. By implementing each tier as a separate subnet, you can control precisely which network flows are permitted between tiers and from the Internet
- The name of a subnet must be unique within that VNet. You cannot change the subnet name after is has been created
- Each subnet must also define a single network range (in CIDR format). This range must be contained within the IP ranges defined by the VNet. Only IP addresses from within the subnets can be assigned to virtual machines and

other resources. Subnets do not have to span the entire VNet address space—they can be a subset, leaving unused space for future expansion

Useful Links:

What is Azure Virtual Network? - https://docs.microsoft.com/en-us/azure/virtual-network/virtual-networks-overview

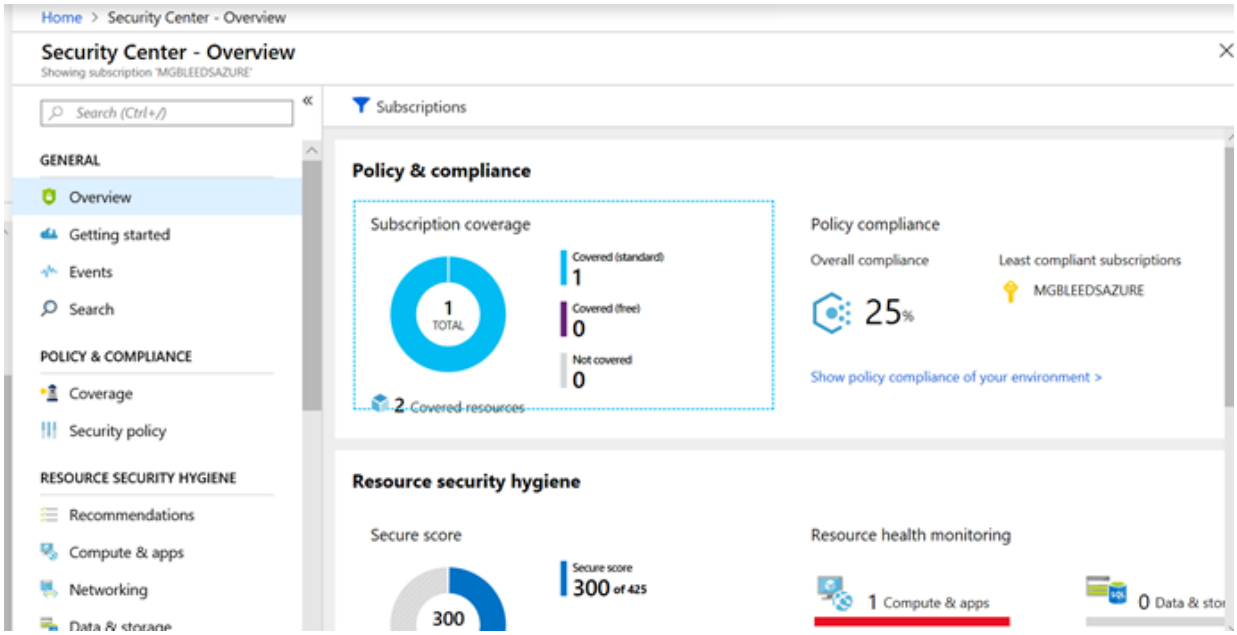
Create a virtual network using the Azure portal - https://docs.microsoft.com/en-us/azure/virtual-network/quick-create-portal

Application Gateway configuration overview - https://docs.microsoft.com/en-us/azure/application-gateway/configuration-overview

Azure Security

Azure Security Centre is a unified infrastructure security management system that strengthens the security posture of your data centres and provides advanced threat protection across your hybrid workloads in the cloud - whether they're in Azure or not - as well as on premises.

Security Centre Details



Azure Backup:

- Azure Backup is the Azure-based service you can use to back up (or protect) and restore your data in the Microsoft cloud. Azure Backup replaces your existing on-premises or off-site backup solution with a cloud-based solution that is reliable, secure, and cost-competitive. Azure Backup offers multiple components that you download and deploy on the appropriate computer, server, or in the cloud. The component, or agent, that you deploy depends on what you want to protect
- All Azure Backup components (no matter whether you're protecting data on-premises or in the cloud) can be used to back up data to a Recovery Services vault in Azure

Azure Backup Overview

Azure Site Recovery:

- As an organization you need to adopt a business continuity and disaster recovery (BCDR) strategy that keeps your data safe, and your apps and workloads up and running, when planned and unplanned outages occur
- Azure Recovery Services contribute to your BCDR strategy

Site Recovery service:

- Site Recovery helps ensure business continuity by keeping business apps and workloads running during outages. Site Recovery replicates workloads running on physical and virtual machines (VMs) from a primary site to a secondary location. When an outage occurs at your primary site, you fail over to secondary location, and access apps from there. After the primary location is running again, you can fail back to it

Backup service:

- The Azure Backup service keeps your data safe and recoverable by backing it up to Azure
- Site Recovery can manage replication for:
    - Azure VMs replicating between Azure regions
    - On-premises VMs, Azure Stack VMs and physical servers

Azure Migrate

The Azure Migrate service assesses on-premises workloads for migration to Azure. The service assesses the migration suitability of on-premises machines, performs performance-based sizing, and provides cost estimations for running on-premises machines in Azure.

If you're contemplating lift-and-shift migrations, or are in the early assessment stages of migration, this service is for you. After the assessment, you can use services such as Azure Site Recovery and Azure Database Migration Service, to migrate the machines to Azure.

Azure Migrate Overview