Deploy a website with Azure virtual machines

If your web hosting requirements aren't directly supported by the Azure Web app platform, you can leverage virtual machines to customize and control every aspect of the web server.

Introduction to Azure virtual machines

Network

Virtual networks (VNets) are used in Azure to provide private connectivity between Azure Virtual Machines and other Azure services. VMs and services that are part of the same virtual network can access one another. By default, services outside the virtual network cannot connect to services within the virtual network. You can, however, configure the network to allow access to the external service, including your on-premises servers.

This latter point is why you should spend some time thinking about your network configuration. Network addresses and subnets are not trivial to change once you have them set up, and if you plan to connect your private company network to the Azure services, you will want to make sure you consider the topology before putting any VMs into place.

When you set up a virtual network, you specify the available address spaces, subnets, and security. If the VNet will be connected to other VNets, you must select address ranges that are not overlapping. This is the range of private addresses that the VMs and services in your network can use. You can use unrouteable IP addresses such as 10.0.0.0/8, 172.16.0.0/12, or 192.168.0.0/16, or define your own range. Azure will treat any address range as part of the private VNet IP address space if it is only reachable within the VNet, within interconnected VNets, and from your on-premises location. If someone else is responsible for the internal networks, you should work with that person before selecting your address space to make sure there is no overlap and to let them know what space you want to use, so they don’t try to use the same range of IP addresses.

After deciding the virtual network address space(s), you can create one or more subnets for your virtual network. You do this to break up your network into more manageable sections. For example, you might assign 10.1.0.0 to VMs, 10.2.0.0 to back-end services, and 10.3.0.0 to SQL Server VMs.

By default, there is no security boundary between subnets, so services in each of these subnets can talk to one another. However, you can set up Network Security Groups (NSGs), which allow you to control the traffic flow to and from subnets and to and from VMs. NSGs act as software firewalls, applying custom rules to each inbound or outbound request at the network interface and subnet level. This allows you to fully control every network request coming in or out of the VM.

Plan each VM deployment

Once you have mapped out your communication and network requirements, you can start thinking about the VMs you want to create. A good plan is to select a server and take an inventory:

- What does the server communicate with?
- Which ports are open?
- Which OS is used?
- How much disk space is in use?
- What kind of data does this use? Are there restrictions (legal or otherwise) with not having it on-premises?
- What sort of CPU, memory, and disk I/O load does the server have? Is there burst traffic to account for?

Name the VM

One piece of information people often don't put much thought into is the name of the VM. The VM name is used as the computer name, which is configured as part of the operating system. You can specify a name of up to 15 characters on a Windows VM and 64 characters on a Linux VM.

This name also defines a manageable Azure resource, and it's not trivial to change later. That means you should choose names that are meaningful and consistent, so you can easily identify what the VM does. A good convention is to include the following information in the name:

Element	Example	Notes
Environment	dev, prod, QA	Identifies the environment for the resource
Location	uw (US West), ue (US East)	Identifies the region into which the resource is deployed
Instance	01, 02	For resources that have more than one named instance (web servers, etc.)
Product or Service	service	Identifies the product, application, or service that the resource supports
Role	sql, web, messaging	Identifies the role of the associated resource

What is an Azure resource?

An Azure resource is a manageable item in Azure. Just like a physical computer in your datacenter, VMs have several elements that are needed to do their job:

- The VM itself
- Storage account for the disks
- Virtual network (shared with other VMs and services)
- Network interface to communicate on the network
- Network Security Group(s) to secure the network traffic
- Public Internet address (optional)

Decide the location for the VM

Azure has datacenters all over the world filled with servers and disks. These datacenters are grouped into geographic regions ('West US', 'North Europe', 'Southeast Asia', etc.) to provide redundancy and availability.

When you create and deploy a virtual machine, you must select a region where you want the resources (CPU, storage, etc.) to be allocated. This lets you place your VMs as close as possible to your users to improve performance and to meet any legal, compliance, or tax requirements.

Two other things to think about regarding the location choice. First, the location can limit your available options. Each region has different hardware available and some configurations are not available in all regions. Second, there are price differences between locations. If your workload isn't bound to a specific location, it can be very cost effective to check your required configuration in multiple regions to find the lowest price.

Determine the size of the VM

Once you have the name and location set, you need to decide on the size of your VM. Rather than specify processing power, memory, and storage capacity independently, Azure provides different VM sizes that offer variations of these elements in different sizes. Azure provides a wide range of VM size options allowing you to select the appropriate mix of compute, memory, and storage for what you want to do.

The best way to determine the appropriate VM size is to consider the type of workload your VM needs to run. Based on the workload, you're able to choose from a subset of available VM sizes. Workload options are classified as follows on Azure:

Option	Description
General purpose	General-purpose VMs are designed to have a balanced CPU-to-memory ratio. Ideal for testing and development, small to medium databases, and low to medium traffic web servers.

Compute optimized	Compute optimized VMs are designed to have a high CPU-to-memory ratio. Suitable for medium traffic web servers, network appliances, batch processes, and application servers.
Memory optimized	Memory optimized VMs are designed to have a high memory-to-CPU ratio. Great for relational database servers, medium to large caches, and in-memory analytics.
Storage optimized	Storage optimized VMs are designed to have high disk throughput and IO. Ideal for VMs running databases.
GPU	GPU VMs are specialized virtual machines targeted for heavy graphics rendering and video editing. These VMs are ideal options for model training and inferencing with deep learning.
High performance computes	High performance compute is the fastest and most powerful CPU virtual machines with optional high-throughput network interfaces.

What if my size needs change?

Azure allows you to change the VM size when the existing size no longer meets your needs. You can upgrade or downgrade the VM - as long as your current hardware configuration is allowed in the new size. This provides a fully agile and elastic approach to VM management.

The VM size can be changed while the VM is running, as long as the new size is available in the current hardware cluster the VM is running on. The Azure portal makes this obvious by only showing you available size choices. The command line tools will report an error if you attempt to resize a VM to an unavailable size. Changing a running VM size will automatically reboot the machine to complete the request.

Understanding the pricing model

There are two separate costs the subscription will be charged for every VM: compute and storage. By separating these costs, you scale them independently and only pay for what you need.

Compute costs - Compute expenses are priced on a per-hour basis but billed on a per-minute basis. For example, you are only charged for 55 minutes of usage if the VM is deployed for 55 minutes. You are not charged for compute capacity if you stop and deallocate the VM since this releases the hardware. The hourly price varies based on the VM size and OS you select. The cost for a VM includes the charge for the Windows

operating system. Linux-based instances are cheaper because there is no operating system license charge.

Storage costs - You are charged separately for the storage the VM uses. The status of the VM has no relation to the storage charges that will be incurred; even if the VM is stopped/deallocated and you aren't billed for the running VM, you will be charged for the storage used by the disks.

You're able to choose from two payment options for compute costs.

Option	Description
Pay as you go	With the pay-as-you-go option, you pay for compute capacity by the second, with no long-term commitment or upfront payments. You're able to increase or decrease compute capacity on demand as well as start or stop at any time. Prefer this option if you run applications with short-term or unpredictable workloads that cannot be interrupted. For example, if you are doing a quick test, or developing an app in a VM, this would be the appropriate option.
Reserved Virtual Machine Instances	The Reserved Virtual Machine Instances (RI) option is an advance purchase of a virtual machine for one or three years in a specified region. The commitment is made up front, and in return, you get up to 72% price savings compared to pay-as-you-go pricing. RIs are flexible and can easily be exchanged or returned for an early termination fee. Prefer this option if the VM has to run continuously, or you need budget predictability, and you can commit to using the VM for at least a year.

Storage for the VM

All Azure virtual machines will have at least two virtual hard disks (VHDs). The first disk stores the operating system, and the second is used as temporary storage. You can add additional disks to store application data; the maximum number is determined by the VM size selection (typically two per CPU). It's common to create one or more data disks, particularly since the OS disk tends to be quite small. Also, separating out the data to different VHDs allows you to manage the security, reliability, and performance of the disk independently.

The data for each VHD is held in **Azure Storage** as page blobs, which allows Azure to allocate space only for the storage you use. It's also how your storage cost is measured; you pay for the storage you are consuming.

What is Azure Storage?

Azure Storage is Microsoft's cloud-based data storage solution. It supports almost any type of data and provides security, redundancy, and scalable access to the stored data. A storage account provides access to objects in Azure Storage for a specific subscription. VMs always have one or more storage accounts to hold each attached virtual disk.

Virtual disks can be backed by either **Standard** or **Premium** Storage accounts. Azure Premium Storage leverages solid-state drives (SSDs) to enable high performance and low latency for VMs running I/O-intensive workloads. Use Azure Premium Storage for production workloads, especially those that are sensitive to performance variations or are I/O intensive. For development or testing, Standard storage is fine.

When you create disks, you will have two options for managing the relationship between the storage account and each VHD. You can choose either unmanaged disks or managed disks.

Option	Description
Unmanaged disks	With unmanaged disks, you are responsible for the storage accounts that are used to hold the VHDs that correspond to your VM disks. You pay the storage account rates for the amount of space you use. A single storage account has a fixed-rate limit of 20,000 I/O operations/sec. This means that a storage account is capable of supporting 40 standard virtual hard disks at full utilization. If you need to scale out with more disks, then you'll need more storage accounts, which can get complicated.
Managed disks	Managed disks are the newer and recommended disk storage model. They elegantly solve this complexity by putting the burden of managing the storage accounts onto Azure. You specify the size of the disk, up to 4 TB, and Azure creates and manages both the disk *and* the storage. You don't have to worry about storage account limits, which makes managed disks easier to scale out.

Select an operating system

Azure provides a variety of OS images that you can install into the VM, including several versions of Windows and flavors of Linux. As mentioned earlier, the choice of OS will influence your hourly compute pricing as Azure bundles the cost of the OS license into the price.

If you are looking for more than just base OS images, you can search the Azure Marketplace for more sophisticated install images that include the OS and popular software tools installed for specific scenarios. For example, if you needed a new WordPress site, the standard technology stack would consist of a Linux server, Apache web server, a MySQL database, and PHP. Instead of setting up and configuring each component, you can leverage a Marketplace image and install the entire stack all at once.

Finally, if you can't find a suitable OS image, you can create your disk image with what you need, upload it to Azure storage, and use it to create an Azure VM. Keep in mind that Azure only supports 64-bit operating systems.

Azure Resource Manager

Let's assume you want to create a copy of a VM with the same settings. You could create a VM image, upload it to Azure, and reference it as the basis for your new VM. This process is inefficient and time-consuming. Azure provides you with the option to create a template from which to create an exact copy of a VM.

Typically, your Azure infrastructure will contain many resources, many of them related to one another in some way. For example, the VM we created has the virtual machine itself, storage and network interface. **Azure Resource Manager** makes working with these related resources more efficient. It organizes resources into named **resource groups** that let you deploy, update, or delete all of the resources together. When we created the Ubuntu VM site, we identified the resource group as part of the VM creation, and Resource Manager placed the associated resources into the same group.

Resource Manager also allows you to create *templates*, which can be used to create and deploy specific configurations.

Resource Manager templates are JSON files that define the resources you need to deploy for your solution.

Once you have it working the way you want it, you can take that template and easily re-create multiple versions of your infrastructure, such as staging and production. You can parameterize fields such as the VM name, network name, storage account name, etc., and load the template repeatedly, using different parameters to customize each environment.

You can use automation scripting tools such as the Azure CLI, Azure PowerShell, or even the Azure REST APIs with your favorite programming language to process resource templates, making this a powerful tool for quickly spinning up your infrastructure.

Azure PowerShell

Creating administration scripts is a powerful way to optimize your workflow. You can automate everyday, repetitive tasks, and once a script has been verified, it will run consistently, likely reducing errors. Azure PowerShell is ideal for one-off interactive tasks and/or the automation of repeated tasks.

```
New-AzVm `
    -ResourceGroupName "TestResourceGroup" `
    -Name "test-wp1-eus-vm" `
    -Location "East US" `
    -VirtualNetworkName "test-wp1-eus-network" `
    -SubnetName "default" `
    -SecurityGroupName "test-wp1-eus-nsg" `
    -PublicIpAddressName "test-wp1-eus-pubip" `
    -OpenPorts 80,3389
```

Azure CLI

Another option for scripting and command-line Azure interaction is the Azure CLI.

The Azure CLI is Microsoft's cross-platform command-line tool for managing Azure resources such as virtual machines and disks from the command line. It's available for macOS, Linux, and Windows, or in the browser using the Cloud Shell. Like Azure PowerShell, the Azure CLI is a powerful way to streamline your administrative workflow. Unlike Azure PowerShell, the Azure CLI does not need PowerShell to function.

```
az vm create \
    --resource-group TestResourceGroup \
    --name test-wp1-eus-vm \
    --image win2016datacenter \
    --admin-username jonc \
    --admin-password aReallyGoodPasswordHere
```

The Azure CLI can be used with other scripting languages, for example, Ruby and Python. Both languages are commonly used on non-Windows-based machines where the developer might not be familiar with PowerShell.

Programmatic (APIs)

Generally speaking, both Azure PowerShell and Azure CLI are good options if you have simple scripts to run and want to stick to command-line tools. When it comes to more complex scenarios, where the creation and management of VMs form part of a larger application with complex logic, another approach is needed.

You can interact with every type of resource in Azure programmatically.

Azure REST API

The Azure REST API provides developers with operations categorized by resource as well as the ability to create and manage VMs. Operations are exposed as URIs with corresponding HTTP methods (GET, PUT, POST, DELETE, and PATCH) and a corresponding response.

The Azure Compute APIs give you programmatic access to virtual machines and their supporting resources. With this API, you have operations to:

- Create and manage availability sets
- Add and manage virtual machine extensions
- Create and manage managed disks, snapshots, and images
- Access the platform images available in Azure
- Retrieve usage information of your resources
- Create and manage virtual machines
- Create and manage virtual machine scale sets

Azure Client SDK

Even though the REST API is platform and language agnostic, most often developers will look toward a higher level of abstraction. The Azure Client SDK encapsulates the Azure REST API, making it much easier for developers to interact with Azure.

```
var azure = Azure
    .Configure()
    .WithLogLevel(HttpLoggingDelegatingHandler.Level.Basic)
    .Authenticate(credentials)
    .WithDefaultSubscription();
// ...
var vmName = "test-wp1-eus-vm";

azure.VirtualMachines.Define(vmName)
    .WithRegion(Region.USEast)
    .WithExistingResourceGroup("TestResourceGroup")
    .WithExistingPrimaryNetworkInterface(networkInterface)
    .WithLatestWindowsImage("MicrosoftWindowsServer", "WindowsServer",
"2012-R2-Datacenter")
    .WithAdminUsername("jonc")
    .WithAdminPassword("aReallyGoodPasswordHere")
    .WithComputerName(vmName)
    .WithSize(VirtualMachineSizeTypes.StandardDS1)
    .Create();
```

The Azure Client SDKs are available for a variety of languages and frameworks, including .NET-based languages such as C#, Java, Node.js, PHP, Python, Ruby, and Go.

Azure VM Extensions

Let's assume you want to configure and install additional software on your virtual machine after the initial deployment. You want this task to use a specific configuration, monitored and executed automatically.

Azure VM extensions are small applications that allow you to configure and automate tasks on Azure VMs after initial deployment. **Azure VM extensions** can be run with the Azure CLI, PowerShell, Azure Resource Manager templates, and the Azure portal.

You bundle extensions with a new VM deployment or run them against an existing system.

Azure Automation Services

Saving time, reducing errors, and increasing efficiency are some of the most significant operational management challenges faced when managing remote infrastructure. If you have a lot of infrastructure services, you might want to consider using higher-level services in Azure to help you operate from a higher level.

Azure Automation allows you to integrate services that allow you to automate frequent, time-consuming, and error-prone management tasks with ease. These services include **process automation, configuration management, and update management.**

- **Process Automation.** Let's assume you have a VM that is monitored for a specific error event. You want to take action and fix the problem as soon as it's reported. Process automation allows you to set up watcher tasks that can respond to events that may occur in your datacenter.
- **Configuration Management.** Perhaps you want to track software updates that become available for the operating system that runs on your VM. There are specific updates you may want to include or exclude. Configuration management allows you to track these updates and take action as required. You use **Microsoft Endpoint Configuration Manager** to manage your company's PC, servers, and mobile devices. You can extend this support to your Azure VMs with Configuration Manager.
- **Update Management.** This is used to manage updates and patches for your VMs. With this service, you're able to assess the status of available updates, schedule installation, and review deployment results to verify updates applied successfully. Update management incorporates services that provide process and configuration management. You enable update management for a VM directly from your **Azure Automation** account. You can also allow update management for a single virtual machine from the virtual machine pane in the portal.

As you can see, Azure provides a variety of tools to create and administer resources so that you can integrate management operations into a process that works for you. Let's examine some of the other Azure services to make sure your infrastructure resources are running smoothly.

What is availability?

Availability is the percentage of time a service is available for use.

Let's assume you have a website and you want your customers to be able to access information at all times. Your expectation is 100% availability concerning website access.

Azure VMs run on physical servers hosted within the Azure Datacenter. As with most physical devices, there's a chance that there could be a failure. If the physical server fails, the virtual machines hosted on that server will also fail. If this happens, Azure will move the

VM to a healthy host server automatically. However, this self-healing migration could take several minutes, during which, the application(s) hosted on that VM will not be available.

The VMs could also be affected by periodic updates initiated by Azure itself. These maintenance events range from software updates to hardware upgrades and are required to improve platform reliability and performance. These events usually are performed without impacting any guest VMs, but sometimes the virtual machines will be rebooted to complete an update or upgrade.

To ensure your services aren't interrupted and avoid a single point of failure, it's recommended to deploy at least two instances of each VM. This feature is called an *availability set*.

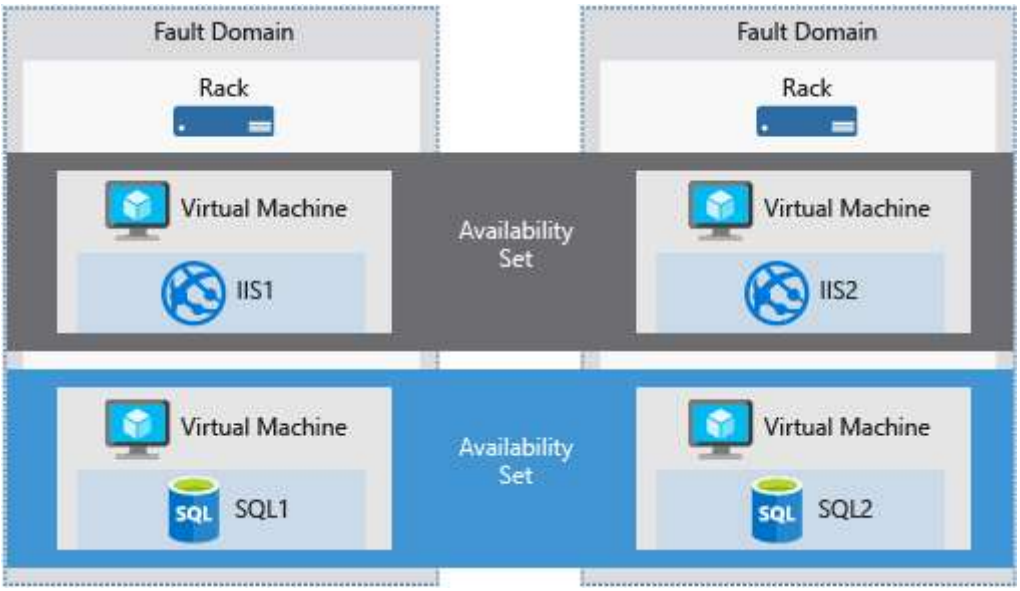
What is an availability set?

An **availability set** is a logical feature used to ensure that a group of related VMs are deployed so that they aren't all subject to a single point of failure and not all upgraded at the same time during a host operating system upgrade in the datacenter. VMs placed in an availability set should perform an identical set of functionalities and have the same software installed.

When you place VMs into an availability set, Azure guarantees to spread them across **Fault Domains** and **Update Domains**.

What is a fault domain?

A fault domain is a logical group of hardware in Azure that shares a common power source and network switch. You can think of it as a rack within an on-premises datacenter. The first two VMs in an availability set will be provisioned into two different racks so that if the network or the power failed in a rack, only one VM would be affected. Fault domains are also defined for managed disks attached to VMs.



What is an update domain?

An update domain is a logical group of hardware that can undergo maintenance or be rebooted at the same time. Azure will automatically place availability sets into update domains to minimize the impact when the Azure platform introduces host operating system changes. Azure then processes each update domain one at a time.

Availability sets are a powerful feature to ensure the services running in your VMs are always available to your customers. However, they aren't foolproof. What if something happens to the data or the software running on the VM itself? For that, we'll need to look at other disaster recovery and backup techniques.

Failover across locations

You can also replicate your infrastructure across sites to handle regional failover. **Azure Site Recovery** replicates workloads from a primary site to a secondary location. If an outage happens at your primary site, you can fail over to a secondary location. This failover allows users to continue to access your applications without interruption. You can then fail back to the primary location once it's up and running again. Azure Site Recovery is about replication of virtual or physical machines; it keeps your workloads available in an outage.

While there are many attractive technical features to Site Recovery, there are at least two significant business advantages:

1. Site Recovery enables the use of Azure as a destination for recovery, thus eliminating the cost and complexity of maintaining a secondary physical datacenter.
2. Site Recovery makes it incredibly simple to test failovers for recovery drills without impacting production environments. This makes it easy to test your planned or unplanned failovers. After all, you don’t have a good disaster recovery plan if you’ve never tried to failover.

The recovery plans you create with Site Recovery can be as simple or as complex as your scenario requires. They can include custom PowerShell scripts, Azure Automation runbooks, or manual intervention steps. You can leverage the recovery plans to replicate workloads to Azure, easily enabling new opportunities for migration, temporary bursts during surge periods, or development and testing of new applications.

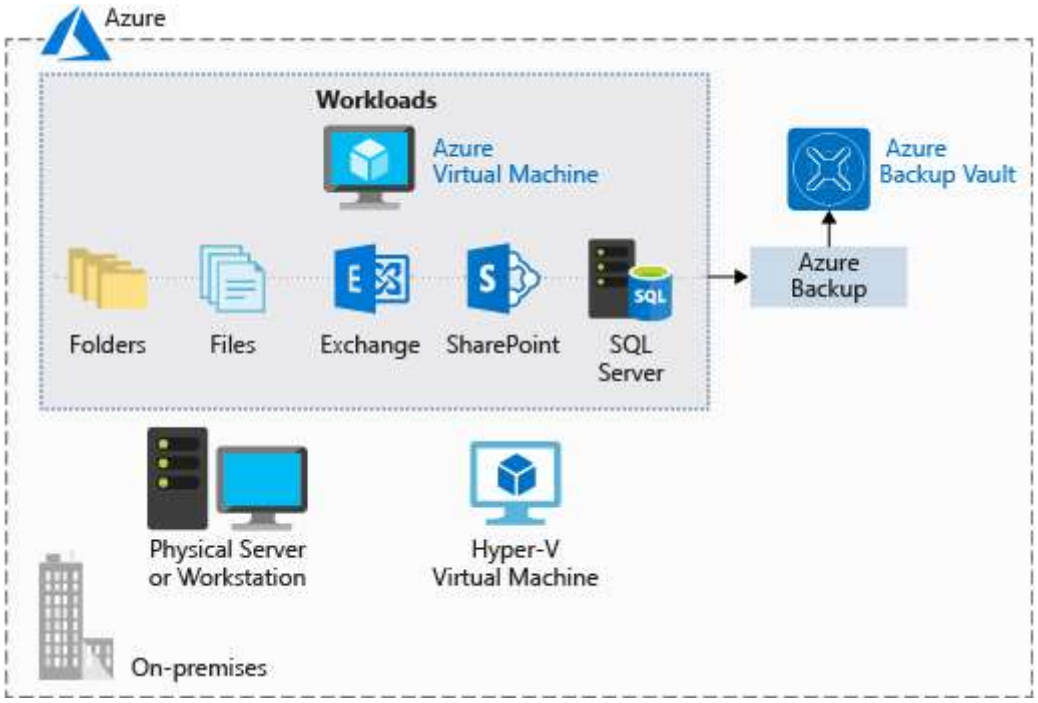
Azure Site Recovery works with Azure resources, or Hyper-V, VMware, and physical servers in your on-premises infrastructure and can be a key part of your organization’s business continuity and disaster recovery (BCDR) strategy by orchestrating the replication, failover, and recovery of workloads and applications if the primary location fails.

Back up your virtual machines

Azure Backup is a *backup as a service* offering that protects physical or virtual machines no matter where they reside: on-premises or in the cloud.

Azure Backup can be used for a wide range of data backup scenarios, such as the following:

- Files and folders on Windows OS machines (physical or virtual, local or cloud)
- Application-aware snapshots (Volume Shadow Copy Service)
- Popular Microsoft server workloads such as Microsoft SQL Server, Microsoft SharePoint, and Microsoft Exchange
- Native support for Azure Virtual Machines, both Windows, and Linux
- Linux and Windows 10 client machines



Advantages of using Azure Backup

Traditional backup solutions don't always take full advantage of the underlying Azure platform. The result is a solution that tends to be expensive or inefficient. The solution either offers too much or too little storage, does not offer the correct types of storage, or has cumbersome and long-winded administrative tasks. Azure Backup was designed to work in tandem with other Azure services and provides several distinct benefits.

- **Automatic storage management.** Azure Backup automatically allocates and manages backup storage and uses a pay-as-you-use model. You only pay for what you use.
- **Unlimited scaling.** Azure Backup uses the power and scalability of Azure to deliver high availability.
- **Multiple storage options.** Azure Backup offers locally redundant storage where all copies of the data exist within the same region and geo-redundant storage where your data is replicated to a secondary region.
- **Unlimited data transfer.** Azure Backup does not limit the amount of inbound or outbound data you transfer. Azure Backup also does not charge for the data that is transferred.

- **Data encryption.** Data encryption allows for secure transmission and storage of your data in Azure.
- **Application-consistent backup.** An application-consistent backup means that a recovery point has all required data to restore the backup copy. Azure Backup provides application-consistent backups.
- **Long-term retention.** Azure doesn't limit the length of time you keep the backup data.

Using Azure Backup

Azure Backup utilizes several components that you download and deploy to each computer you want to back up. The component that you deploy depends on what you want to protect.

- Azure Backup agent
- System Center Data Protection Manager
- Azure Backup Server
- Azure Backup VM extension

Azure Backup uses a Recovery Services vault for storing the backup data. A vault is backed by Azure Storage blobs, making it a very efficient and economical long-term storage medium. With the vault in place, you can select the machines to back up and define a backup policy (when snapshots are taken and for how long they're stored).

Summary

1. Suppose you want to run a network appliance on a virtual machine. Which workload option should you choose?
Compute optimized
Compute optimized virtual machines are designed to have a high CPU-to-memory ratio. Suitable for medium traffic web servers, network appliances, batch processes, and application servers.

2. True or false: Resource Manager templates are JSON files?
True
Resource Manager templates are JSON files that define the resources you need to deploy for your solution. The template can then be used to easily re-create multiple versions of your infrastructure, such as staging and production.

Create a Linux virtual machine in Azure

Introduction to Azure Virtual Machines

Azure Virtual Machines is an on-demand, scalable cloud-computing resource. They include processors, memory, storage, and networking resources. You can start and stop virtual machines at will and manage them from the Azure portal or with the Azure CLI. You can also use a remote Secure Shell (SSH) to connect directly to the running VM and execute commands as if you were on a local computer.

Creating Linux-based VMs in Azure is easy. Microsoft has partnered with prominent Linux vendors to ensure their distributions are optimized for the Azure platform. You can create virtual machines from prebuilt images for a variety of popular Linux distributions, such as SUSE, Red Hat, and Ubuntu, or build your own Linux distribution to run in the cloud.

Resources used in a Linux VM

When creating a Linux VM in Azure, you also create resources to host the VM. These resources work together to virtualize a computer and run the Linux operating system. These must exist (and be selected during VM creation), or they will be created with the VM:

- A virtual machine that provides CPU and memory resources
- An Azure Storage account to hold the virtual hard disks
- Virtual disks to hold the OS, applications, and data
- A virtual network (VNet) to connect the VM to other Azure services or your on-premises hardware
- A network interface to communicate with the VNet
- An optional public IP address so you can access the VM

Like other Azure services, you'll need a **Resource Group** to contain the VM (and optionally group these resources for administration). When you create a new VM, you can either use an existing resource group or create a new one.

VM sizes are grouped into categories, starting with the B-series for basic testing and running up to the H-series for massive computing tasks. It is possible to change the size of a VM after it's been created, but the VM must be stopped first. So, it's best to size it appropriately from the start if possible.

What are you doing?	**Consider these sizes**
General use computing/web: Testing and development, small to medium databases, or low to medium traffic web servers.	B, Dsv3, Dv3, DSv2, Dv2
Heavy computational tasks: Medium traffic web servers, network appliances, batch processes, and application servers.	Fsv2, Fs, F
Large memory usage: Relational database servers, medium to large caches, and in-memory analytics.	Esv3, Ev3, M, GS, G, DSv2, Dv2

Data storage and processing: Big data, SQL, and NoSQL databases that need high disk throughput and I/O.	Ls
Heavy graphics rendering or video editing, as well as model training and inferencing (ND) with deep learning.	NV, NC, NCv2, NCv3, ND
High-performance computing (HPC): Your workloads need the fastest and most powerful CPU virtual machines with optional high-throughput network interfaces.	

Choosing storage options

Options include a traditional platter-based hard disk drive (HDD) or a more modern solid-state drive (SSD). Just like the hardware you purchase, SSD storage costs more but provides better performance.

There are two levels of SSD storage available: standard and premium. Choose Standard SSD disks if you have normal workloads but want better performance. Choose Premium SSD disks if you have I/O intensive workloads or mission-critical systems that need to process data very quickly.

Mapping storage to disks

Azure uses virtual hard disks (VHDs) to represent physical disks for the VM. VHDs replicate the logical format and data of a disk drive but are stored as page blobs in an Azure Storage account. You can choose on a per disk basis what type of storage it should use (SSD or HDD). This allows you to control the performance of each disk, likely based on the I/O you plan to perform on it.

By default, two virtual hard disks (VHDs) will be created for your Linux VM:

1. The **operating system disk**: This is your primary drive, and it has a maximum capacity of 2048 GB. It will be labeled as /dev/sda by default.
2. A t**emporary disk**: This provides temporary storage for the OS or any apps. On Linux virtual machines, the disk is /dev/sdb and is formatted and mounted to /mnt by the Azure Linux Agent. It is sized based on the VM size and is used to store the swap file.

You can store data on the primary drive along with the OS, but a better approach is to create dedicated data disks. You can create and attach additional disks to the VM. Each disk can hold up to 32 767 gibibytes (GiB) of data, with the maximum amount of storage determined by the VM size you select. One GiB is approximately 1.074 GB.

An interesting capability is to create a VHD image from a real disk. This allows you to easily migrate existing information from an on-premises computer to the cloud.

Unmanaged vs. managed disks

The final storage choice you'll make is whether to use **unmanaged** or **managed** disks.

With unmanaged disks, you are responsible for the storage accounts that are used to hold the VHDs that correspond to your VM disks. You pay the storage account rates for the amount of space you use. A single storage account has a fixed rate limit of 20,000 I/O operations/sec. This means that a single storage account is capable of supporting 40 standard virtual hard disks at full throttle. If you need to scale out, then you need more than one storage account, which can get complicated.

Managed disks are the newer and recommended disk storage model. They elegantly solve this complexity by putting the burden of managing the storage accounts onto Azure. You specify the disk type (Premium or Standard) and the size of the disk, and Azure creates and manages both the disk and the storage it uses. You don't have to worry about storage account limits, which makes them easier to scale out. They also offer several other benefits:

- **Increased reliability**: Azure ensures that VHDs associated with high-reliability VMs will be placed in different parts of Azure Storage to provide similar levels of resilience.
- **Better security**: Managed disks are real managed resources in the resource group. This means they can use role-based access control to restrict who can work with the VHD data.
- **Snapshot support**: Snapshots can be used to create a read-only copy of a VHD. We recommend that you shut down the VM to clear out any processes that are in progress. Creating the snapshot only takes a few seconds. Once it's done, you can power on the VM and use the snapshot to create a duplicate VM to troubleshoot a production issue or roll back the VM to the point in time that the snapshot was taken.
- **Backup support**: Managed disks can be automatically backed up to different regions for disaster recovery with Azure Backup without affecting the service of the VM.

Network communication

Virtual machines communicate with external resources using a virtual network (VNet). The VNet represents a private network in a single region that your resources communicate on. A virtual network is just like the networks you manage on-premises. You can divide them up with subnets to isolate resources, connect them to other networks (including your on-premises networks), and apply traffic rules to govern inbound and outbound connections.

What is SSH?

Secure Shell (SSH) is an encrypted connection protocol that allows secure sign-ins over unsecured connections. SSH allows you to connect to a terminal shell from a remote location using a network connection.

There are two approaches we can use to authenticate an SSH connection: **username and password**, or an **SSH key pair**.

Although SSH provides an encrypted connection, using passwords with SSH connections leaves the VM vulnerable to brute-force attacks of passwords. A more secure and preferred method of connecting to a Linux VM with SSH is a public-private key pair, also known as SSH keys.

With an SSH key pair, you can sign in to Linux-based Azure virtual machines without a password. This is a more secure approach if you only plan to sign in to the VM from a few computers. If you need to be able to access the Linux VM from a variety of locations, a username and password combination might be a better approach. There are two parts to an SSH key pair: a public key and a private key.

- The **public key** is placed on your Linux VM or any other service that you wish to use with public-key cryptography. This can be shared with anyone.
- The **private key** is what you present to verify your identity to your Linux VM when you make an SSH connection. Consider this confidential information and protect this like you would a password or any other private data.

You can use the same single public-private key pair to access multiple Azure VMs and services.

Private key passphrase

You can provide a passphrase while generating your private key. This is a password you must enter when you use the key. This passphrase is used to access the private SSH key file and is not the user account password.

When you add a passphrase to your SSH key, it encrypts the private key using 128-bit AES so that the private key is useless without the passphrase to decrypt it.

We strongly recommended that you add a passphrase. If an attacker stole your private key and that key did not have a passphrase, they would be able to use that private key to log in to any servers that have the corresponding public key. If a passphrase protects a private key, it cannot be used by that attacker. This provides an additional layer of security for your infrastructure on Azure.

Azure VM IP addresses

As we saw a moment ago, Azure VMs communicate on a virtual network. They can also have an optional public IP address assigned to them. With a public IP, we can interact with the VM over the Internet. Alternatively, we can set up a virtual private network (VPN) that connects our on-premises network to Azure - letting us securely connect to the VM without exposing a public IP. This approach is covered in another module and is fully documented if you are interested in exploring that option.

Public IP addresses in Azure are dynamically allocated by default. That means the IP address can change over time - for VMs the IP address assignment happens when the VM

is restarted. You can pay more to assign static addresses, if you want to connect directly to an IP address and need to ensure that the IP address will not change.

Opening ports in Azure VMs

By default, new VMs are locked down.

Apps can make outgoing requests, but the only inbound traffic allowed is from the virtual network (e.g., other resources on the same local network) and from Azure Load Balancer (probe checks).

There are two steps to adjusting the configuration to support different protocols on the network. When you create a new VM, you have an opportunity to open a few common ports (RDP, HTTP, HTTPS, and SSH). However, if you require other changes to the firewall, you will need to adjust them manually.

The process for this involves two steps:

1. Create a network security group.
2. Create an inbound rule allowing traffic on the ports you need.

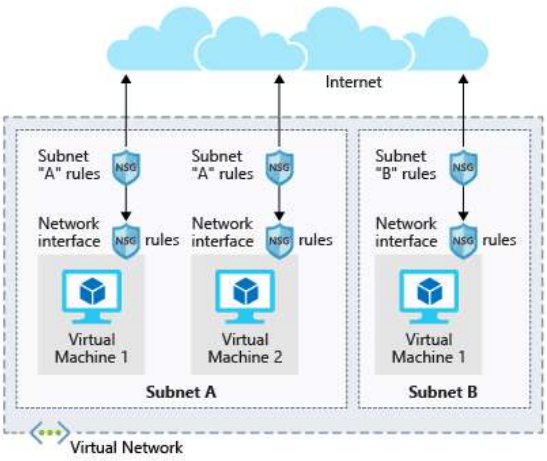
What is a network security group?

Virtual networks (VNets) are the foundation of the Azure networking model and provide isolation and protection. Network security groups (NSGs) are the primary tool you use to enforce and control network traffic rules at the networking level. NSGs are an optional security layer that provides a software firewall by filtering inbound and outbound traffic on the VNet.

Security groups can be associated to a network interface (for per host rules), a subnet in the virtual network (to apply to multiple resources), or both levels.

Security group rules

NSGs use rules to allow or deny traffic moving through the network. Each rule identifies the source and destination address (or range), protocol, port (or range), direction (inbound or outbound), a numeric priority, and whether to allow or deny the traffic that matches the rule.



Each security group has a set of default security rules to apply the default network rules described above. These default rules cannot be modified but can be overridden.

How Azure uses network rules

For inbound traffic, Azure processes the security group associated to the subnet and then the security group applied to the network interface. Outbound traffic is handled in the opposite order (the network interface first, followed by the subnet).

The rules are evaluated in priority order, starting with the **lowest priority** rule. Deny rules always **stop** the evaluation. For example, if a network interface rule blocks an outbound request, any rules applied to the subnet will not be checked. For traffic to be allowed through the security group, it must pass through all applied groups.

The last rule is always a **Deny All** rule. This is a default rule added to every security group for both inbound and outbound traffic with a priority of 65500. That means to have traffic pass through the security group, you must have an allow rule, or the final default rule will block it.

Summary

1. True or false: for security reasons, you must use an image from the official Azure Marketplace when creating a new virtual machine.
False
Azure lets you configure your virtual machines to meet your needs. This includes support for using your own VM images.

2. What is the effect of the default network security settings for a new virtual machine?
Outbound request are allowed. Inbound traffic is only allowed from within the virtual network.
Outbound requests are considered low risk, so they are allowed by default. Inbound traffic from within the virtual network is allowed. By placing a VM in a virtual network, the VM owner is implicitly opting-in to communication among the resources in the virtual network.

3. Suppose you have several Linux virtual machines hosted in Azure. You will administer these VMs remotely over SSH from three dedicated machines in your corporate headquarters. Which of the following authentication methods would typically be considered best-practice for this situation?
Private key with passphrase
Private key access with a passphrase is the most secure option. Even if an attacker acquires your private key, they will be unable to use it without the passphrase.

Create a Windows virtual machine in Azure

Azure VMs are an on-demand scalable cloud computing resource. They're similar to virtual machines that are hosted in Windows Hyper-V. They include processor, memory, storage, and networking resources. You can start and stop virtual machines at will, just like with Hyper-V, and manage them from the Azure portal or with the Azure CLI. You can also use a Remote Desktop Protocol (RDP) client to connect directly to the Windows desktop user interface (UI) and use the VM as if you were signed in to a local Windows computer.

When creating a Windows VM, you also create resources to host the VM. These resources work together to virtualize a computer and run the Windows operating system. These must either exist (and be selected during VM creation), or they will be created with the VM.

- A virtual machine that provides CPU and memory resources.
- An Azure Storage account to hold the virtual hard disks.
- Virtual disks to hold the OS, applications, and data.
- Virtual network (VNet) to connect the VM to other Azure services or your own on-premises hardware.
- A network interface to communicate with the VNet.
- A public IP address so you can access the VM. This is optional.

What are you doing?	**Consider these sizes**
General use computing / web Testing and development, small to medium databases, or low to medium traffic web servers.	B, Dsv3, Dv3, DSv2, Dv2
Heavy computational tasks Medium traffic web servers, network appliances, batch processes, and application servers.	Fsv2, Fs, F
Large memory usage Relational database servers, medium to large caches, and in-memory analytics.	Esv3, Ev3, M, GS, G, DSv2, Dv2
Data storage and processing Big Data, SQL, and NoSQL databases, which need high disk throughput and IO.	Ls
Heavy graphics rendering or video editing, as well as model training and inferencing (ND) with deep learning.	NV, NC, NCv2, NCv3, ND

High-performance computing (HPC) If you need the fastest and most powerful CPU virtual machines with optional high-throughput network interfaces.	H

Mapping storage to disks

Azure uses virtual hard disks (VHDs) to represent physical disks for the VM. VHDs replicate the logical format and data of a disk drive but are stored as page blobs in an Azure Storage account. You can choose on a per-disk basis what type of storage it should use (SSD or HDD). This allows you to control the performance of each disk, likely based on the I/O you plan to perform on it.

By default, two virtual hard disks (VHDs) will be created for your Windows VM:

1. The **Operating System disk**. This is your primary or C: drive and has a maximum capacity of 2048 GB.
2. A **Temporary disk.** This provides temporary storage for the OS or any apps. It is configured as the D: drive by default and is sized based on the VM size, making it an ideal location for the Windows paging file.

Unmanaged vs. Managed disks

With **unmanaged disks**, you are responsible for the storage accounts that are used to hold the VHDs that correspond to your VM disks. You pay the storage account rates for the amount of space you use. A single storage account has a fixed rate limit of 20,000 I/O operations/sec. This means that a single storage account is capable of supporting 40 standard virtual hard disks at full throttle. If you need to scale out, then you need more than one storage account, which can get complicated.

Managed disks are the newer and recommended disk storage model. They elegantly solve this complexity by putting the burden of managing the storage accounts onto Azure. You specify the disk type (Premium or Standard) and the size of the disk and Azure creates and manages both the disk and the storage it uses. You don't have to worry about storage account limits, which makes them easier to scale out. They also offer several other benefits:

- **Increased reliability**: Azure ensures that VHDs associated with high-reliability VMs will be placed in different parts of Azure storage to provide similar levels of resilience.
- **Better security**: Managed disks are truly managed resources in the resource group. This means they can use role-based access control to restrict who can work with the VHD data.
- **Snapshot support**: Snapshots can be used to create a read-only copy of a VHD. You have to shut down the owning VM but creating the snapshot only takes a few seconds. Once it's done, you can power on the VM and use the snapshot to create a duplicate VM to troubleshoot a production issue or rollback the VM to the point in time that the snapshot was taken.
- **Backup support**: Managed disks can be automatically backed up to different regions for disaster recovery with Azure Backup all without affecting the service of the VM.

Network communication

Virtual machines communicate with external resources using a virtual network (VNet). The VNet represents a private network in a single region that your resources communicate on. A virtual network is just like the networks you manage on-premises. You can divide them up with subnets to isolate resources, connect them to other networks (including your on-premises networks), and apply traffic rules to govern inbound and outbound connections.

What is the Remote Desktop Protocol?

Remote Desktop (RDP) provides remote connectivity to the UI of Windows-based computers. RDP enables you to sign in to a remote physical or virtual Windows computer and control that computer as if you were seated at the console. An RDP connection enables you to carry out the vast majority of operations that you can do from the console of a physical computer, with the exception of some power and hardware-related functions.

An RDP connection requires an RDP client. Microsoft provides RDP clients for the following operating systems:

- Windows (built-in)
- macOS
- iOS
- Android

Connecting to a VM in Azure using RDP is a simple process. In the Azure portal, you go to the properties of your VM, click Connect. This will show you the IP addresses assigned to the VM and give you the option to download a **preconfigured.rdp** file that Windows then opens in the RDP client. You can choose to connect over the public IP address of the VM in the RDP file. Instead, if you're connecting over VPN or ExpressRoute, you can select the internal IP address. You can also select the port number for the connection.

If you're using a static public IP address for the VM, you can save the **.rdp** file to your desktop. If you're using dynamic IP addressing, the .rdp file only remains valid while the VM is running. If you stop and restart the VM, you must download another .rdp file.

When you connect, you'll typically receive two warnings. These are:

- **Publisher warning** - caused by the .rdp file not being publicly signed.
- **Certificate warning** - caused by the machine certificate not being trusted.

In test environments, these warnings can be ignored. In production environments, the .rdp file can be signed using **RDPSIGN.EXE** and the machine certificate placed in the client's **Trusted Root Certification Authorities** store.

Opening ports in Azure VMs

By default, new VMs are locked down.

Apps can make outgoing requests, but the only inbound traffic allowed is from the virtual network (e.g. other resources on the same local network), and from Azure's Load Balancer (probe checks).

There are two steps to adjusting the configuration to support FTP. When you create a new VM you have an opportunity to open a few common ports (RDP, HTTP, HTTPS, and SSH). However, if you require other changes to the firewall, you will need to do them yourself.

The process for this involves two steps:

1. Create a Network Security Group.
2. Create an inbound rule allowing traffic on port 20 and 21 for active FTP support.

Virtual networks (VNets) are the foundation of the Azure networking model and provide isolation and protection. Network Security Groups (NSGs) are the main tool you use to enforce and control network traffic rules at the networking level. NSGs are an optional security layer that provides a software firewall by filtering inbound and outbound traffic on the VNet.

Security groups can be associated to a network interface (for per-host rules), a subnet in the virtual network (to apply to multiple resources), or both levels.

NSGs use rules to allow or deny traffic moving through the network. Each rule identifies the source and destination address (or range), protocol, port (or range), direction (inbound or outbound), a numeric priority, and whether to allow or deny the traffic that matches the rule. The following illustration shows NSG rules applied at the subnet and network interface levels.

Summary

1. When creating a Windows virtual machine in Azure, which port would you open using the INBOUND PORT RULES in order to allow remote-desktop access?
RDP (3389)
The Remote Desktop Protocol (RDP) uses port 3389 by default so this port is the standard port you would open if you wanted to use an RDP client to administer your Windows virtual machines.
2. Suppose you have an application running on a Windows virtual machine in Azure. What is the best-practice guidance on where the app should store data files?
An attached data disk
Dedicated data disks are generally considered the best place to store application data files. They can be larger than OS disks and you can optimize them for the cost and performance characteristics appropriate for your data.
3. What is the final rule that is applied in every Network Security Group?
Deny All
This is a safe choice. It will block all traffic that you don't specifically allow.

Build and run a web application with the MEAN stack on an Azure Linux virtual machine

MEAN is a development stack for building and hosting web applications. MEAN is an acronym for its component parts: MongoDB, Express, AngularJS, and Node.js.

The main reason you might consider MEAN is if you're familiar with JavaScript. Here are some other reasons you might want to choose MEAN, or choose a different development stack for your next web application.

Many applications require a database. Here you'll install MongoDB, the "M" in the MEAN stack. It's a popular NoSQL database solution that's free and open source. A NoSQL database doesn't require data to be structured in a pre-defined way as it would with a relational database like SQL Server or MySQL.

MongoDB stores its data in JSON-like documents that don't require rigid data structures. You interact with MongoDB using queries and commands sent using JavaScript Object Notation, or JSON.

The N in the MEAN acronym. Like MongoDB, Node.js is open source.

Node.js will act as the server-side host for your web application and handle inbound HTTP traffic. Node.js also provides you with a way to communicate with your MongoDB installation, which you'll see later.

You can get Node.js in two ways:

- **Long Term Support (LTS)** - LTS is generally considered to be more stable and is recommended for most users and for production environments.
- **Current** - Current is for those who want to experiment with the latest features. Because it can introduce breaking changes between releases, it's not recommended for production environments.

So far, you've installed MongoDB and Node.js on your VM. What about Express, the E in the MEAN acronym?

Express is a web server framework that's built for Node.js that simplifies the process for building web applications.

The main purpose of Express is to handle request routing. Routing refers to how the application responds to a request to a specific endpoint. An endpoint is made up of a path, or URI, and a request method, such as GET or POST. For example, you might respond to a GET request to the /book endpoint by providing the list of all books in the database. You might respond to a POST request to the same endpoint by adding an entry to the database based on fields the user entered into a web form.

In the web application you'll build shortly, you'll use Express to route HTTP requests and to return web content to your user. Express can also help your web applications work with HTTP cookies and process query strings.

Express is a Node.js package. You use the npm utility, which comes with Node.js, to install and manage Node.js packages. Later in this part, you'll create a file named package.json to define Express and other dependencies and then run the npm install command to install these dependencies.

Like Express, you haven't yet installed AngularJS, the A in the MEAN acronym.

AngularJS makes web applications easier to write and test because it enables you to better separate the appearance of your web page, your HTML code, from how your web page behaves. If you're familiar with the model–view–controller (MVC) pattern or the concept of data binding, AngularJS will be familiar to you.

AngularJS is what's called a front-end JavaScript framework, which means it needs to only be available on the client that accesses the application. In other words, AngularJS runs in your user's web browser, not on your web server. And because AngularJS is JavaScript, you can use it to easily fetch data from your web server to show on the page.

You don't really install AngularJS. Instead, you add a reference to the JavaScript file in your HTML page, just as you do with other JavaScript libraries. There are several ways to include AngularJS in your web pages. Here you'll load AngularJS from a content delivery network, or CDN. A CDN is a way to distribute images, video, and other content geographically to improve download speeds. You would typically add this code to the <head> section of your HTML page.

Summary

1. Which is a good reason to pick MEAN for your development stack?
Everyone on your development team is an expert with JavaScript.
The MEAN stack greatly relies on JavaScript. JavaScript appears in both the client side and the server side of your web app.

2. What's a NoSQL database?
It's a database that doesn't require data to be structured in a pre-defined way.
MongoDB, for example, stores its data in JSON-like documents that don't require the rigid data structures required by MySQL or other relational databases.

3. What's the role of AngularJS in the MEAN stack?
AngularJS implements data binding for HTML and JavaScript. For example, you can use *AngularJS to retrieve items from a database and display those items in the UI.*
Data binding helps keep your HTML clean and minimal. AngularJS displays updates immediately to the user when the underlying data changes.