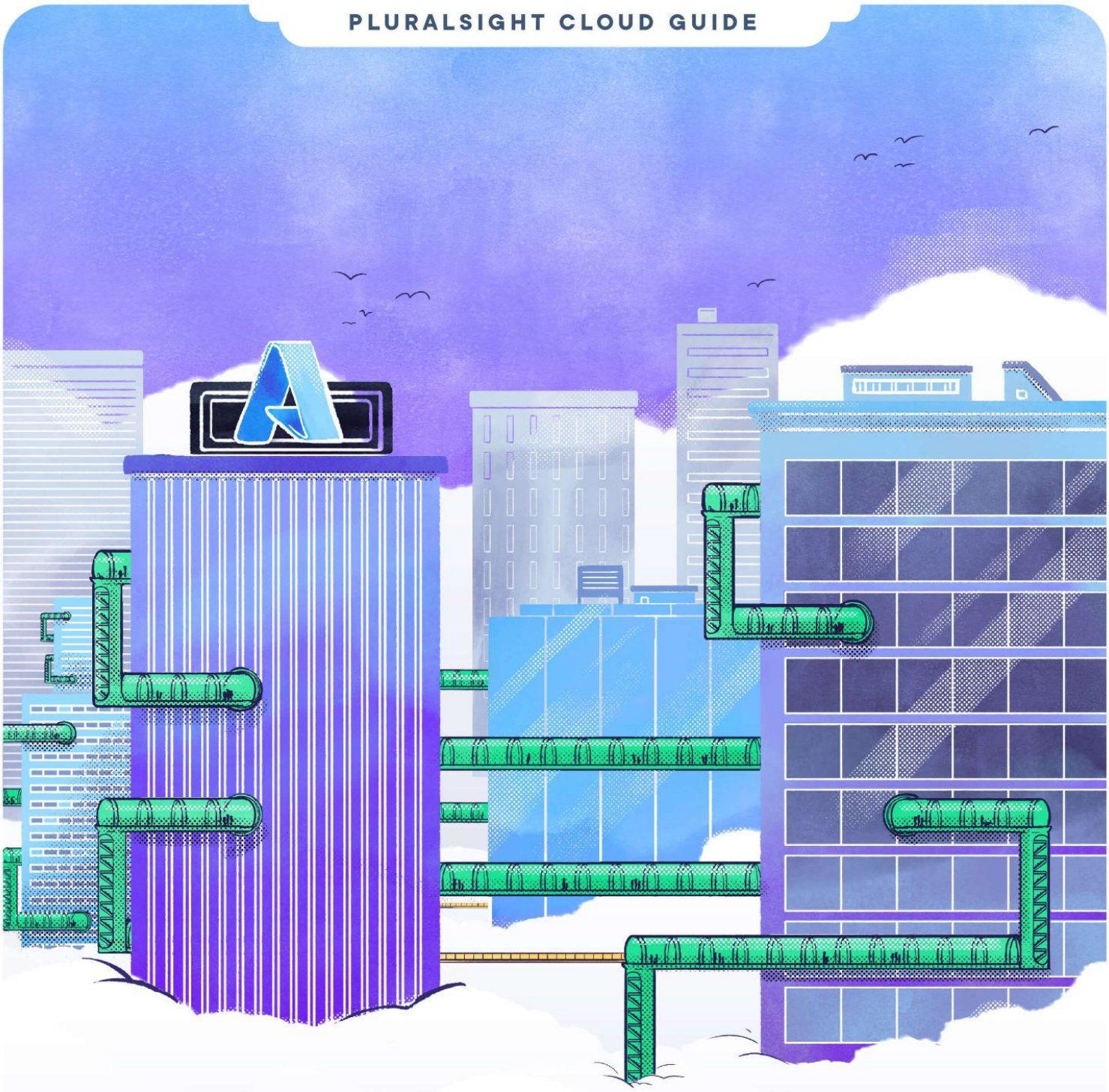
PLURALSIGHT CLOUD GUIDE



Azure

Guidebook

PLURALSIGHT

Azure Guidebook

Last updated: May 2023

Whether you're just diving into the cloud, looking for a refresher, or expanding your knowledge of the different cloud platforms, this guidebook explains the most important terms to help you talk like an Azure local.

We provide an overview explanation for each term to help you understand the lay of the land. Then we dive into the secrets only the Azure locals know—what to avoid and where to spend the most time. When you want to know more, check out the related courses and hands-on labs.

Ready to explore the world of Azure? Dive right in.

INDEX

Azure Autoscale and Azure Monitor 02

Azure Cosmos DB 03

Azure DNS and Azure Traffic Manager 05

Azure Functions 06

Azure identity and access management (IAM) 07

Azure SQL family 09

Azure Storage: Blob, Disk, and Files 10

Azure Virtual Machines (VMs) 11

Azure Load Balancer 12

Snapshots 14

Azure Virtual Network (VNet) 15

About Pluralsight 16

Azure Autoscale and Azure Monitor

Overview

Azure Autoscale is a feature that modifies application resource allocations up or down based on the demand of the application, which you configure in Azure Monitor. This feature makes your applications more adaptable to changes in requests from users.

Within Azure Monitor, you can look at metrics like CPU percentage, queue length, or rate of data input and output within resources. Azure Monitor tracks these metrics, and Azure Autoscale scales your resources based on set thresholds.

When configuring an autoscaling group, ask four questions:

- How many servers do you want to maintain uptime?
- Do you want to adjust your server count manually?
- Do you want to schedule when to scale up or down?
- Would you like it to be based on conditions with your product performance?

Off the record

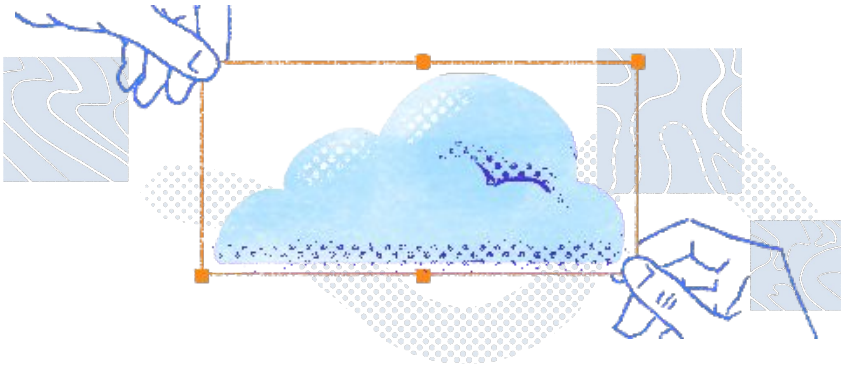
There are two types of scaling you can implement:

Vertical scaling: Scale up or down by changing a resource's capacity. For example, you might increase the processor size of a VM to handle more requests at the same amount of time. This is often referred to as "scaling up."

Horizontal scaling: Add more resources to handle requests as demand increases. This type of scaling is frequently called "scaling out."

Each type of scaling introduces its own set of challenges. If you're scaling vertically, your specific resource may become unavailable, which won't work if you're in the middle of a process. Go for horizontal scaling to introduce load balancing and partitioning.

You also have to decide how you're going to set your autoscaling thresholds. You can set a limit to instances, but that might not be enough. Each rule you add will make your autoscaling processes more complicated and introduce more room for error.



Get hands-on with our Implementing Azure Monitor lab. Explore how Azure Monitor works in a SQL database through the implementation of Azure Monitor and the configuration of alerts, security logs, log analytics queries, and basic metric charts.

Azure Cosmos DB

Overview

Azure Cosmos DB is a system that tries to synchronize data across the globe instead of specific regions. This kind of global synchronization is essential for global applications that need to be highly responsive, have a lot of constantly changing data, and are always available to users. Facebook is a good example. People use it worldwide and are always altering the data.

In practice, although Azure Cosmos DB is replicating data across all regions to ensure a continuous global database, the applications think Azure Cosmos DB is in their region. Azure Cosmos DB also automatically partitions data to optimize performance and storage capacity.

Azure Cosmos DB is also accessible through multiple APIs, such as Document DB (for SQL), MongoDB (for NoSQL), Graph API (for Gremlin), and Tables API (for key/value pairs).

Off the record

With all that replication happening across regions, there will be some variation in quality. Azure Cosmos DB calls these variations "consistency levels." They balance performance with predictability and fall into five categories:

- **Strong:** Strong consistency is the most predictable and intuitive. It ensures a guaranteed write operation is committed and visible on a primary Azure Cosmos DB after being committed and confirmed on all replicas.
- **Bounded staleness:** Bounded staleness is the most frequently chosen and allows you to determine how stale data can be used. It decides how far behind a document can be before it needs updating.
- **Session:** This level ensures all read/write operations are consistent within a current user session. For example, in a Facebook user session, Facebook will have data particular to that user.
- **Consistent prefix:** This ensures changes are read in the sequence of the corresponding writes.
- **Eventual:** The loosest consistency, this level commits and writes against a primary immediately. Replicant transactions are synchronously handled and eventually get to replicas.

Azure DNS and Azure Traffic Manager

Overview

Azure DNS contains IP addresses from the domain name, and Azure Traffic Manager picks the right IP from that address list.

DNS (Domain Name System) is the internet's phone book, which converts a name, like pluralsight.com, to an IP address. Azure contains a DNS zone for a domain name, which hosts the IPs for that domain. You build a DNS zone from a unique resource group that gives you a domain, or you can create a private DNS zone (requires setup through the command line). All the records for a domain are in a DNS zone.

Traffic Manager figures out how to intelligently route someone to your application and ensures high availability across different geographic regions. If a region within your solution goes offline, Traffic Manager routes to online regions.

Here's what it looks like:

1. The user loads the solution and does a DNS lookup.
2. Traffic Manager responds with an IP address based on the configuration of the service.
3. The user navigates to the appropriate solution.

Off the record

Azure DNS and Traffic Manager work together to make sure users are able to access your app and have a good experience while doing so. But in classic cloud fashion, there are several routing methods within Traffic Manager:

- **Priority** prioritizes primary and backup endpoints.
- **Weighted** distributes traffic according to weight value (e.g., 20% going to one region and 20% to another).
- **Performance** sends traffic to the closest endpoint, which is good for global solutions.
- **Geographic** routes traffic based on the geographic location of the client.
- **Multivalue** returns multiple endpoints and leaves it up to the client to determine which one to use.
- **Subnet** routes based on the requester's IP address.

Once you've added your custom domain (you must own the domain name), you'll complete your custom domain's validation by uploading a text record from whoever manages your domain name. Or you could configure an Azure Virtual Network (VNet) to enable auto-registration, pointing to a private internal DNS zone. You can use a private DNS for VMs to connect with named domains, such as somethingsomething.internal.

Get hands-on with our Implement an Azure Traffic Manager Environment lab. In 45 minutes, connect two existing app service websites in two regions with a Traffic Manager profile.

Azure Functions

Overview

Azure Functions is Microsoft Azure's serverless computing product. It's the most flexible type of scaling related to workload volumes. The serverless programming model is based on triggers and bindings—focusing on writing code that responds to specific events and returns data the function needs to return. Azure Functions also have a rich end-to-end development experience.

Ultimately, there's a server in Azure you know nothing about that's running your function. You write some code to respond to an event and never see anything that's happening with the server on Microsoft's end.

Azure Functions offer the ability to run single-code pieces in response to events and are billable for that execution time rather than another billing method. You can save quite a bit of money since you're not paying for servers that are up and running but not doing anything.

Off the record

Serverless can feel at odds with traditional cloud computing. Though there are a myriad of benefits, especially when it comes to costs, companies accustomed to managing VMs to handle their applications will have to change the way they think about their structure, including at an architectural level. Here are some architectural considerations essential to understanding functions:

- **Event driven:** Code is executed in response to events on an as-needed basis. Servers aren't sitting around waiting for things to happen.
- **Reactive:** Code is applied to particular events that happen when needed. It's responsive, resilient (e.g., events will be reprocessed if failed), elastic (e.g., as events go up, it scales up), and message driven.
- **Multifactor:** Functions can be deployed via a variety of methods, whether via development pipelines or directly configured within Azure.

There are many ways to execute a function: on-time intervals, HTTP requests, when something is uploaded in Blob storage, a message from an Azure Storage queue, an Azure Cosmos DB document change, or an event hub receiving a new event.

Dive into serverless functions with our Serverless Computing with Azure Functions course. This deep-dive course includes tools for creating, running, and operating Azure Functions.

Azure identity and access management (IAM)

Overview

Let's say you built a team that manages an Azure application. You govern the access the team has to the services associated with their application but not other services that might contain sensitive information.

Identity and access management (IAM) is core to any cloud usage, allowing you to manage services, resources, and applications. It gives developers the ability to push updates to production and auditors access to inspect your work. All cloud providers have some form of IAM. It's the foundation for providing security in the cloud. And you can set up IAM to include a variety of ways to log in, including a Windows account through Active Directory (AD).

With IAM, you enable access to your applications at a granular level. Each user with access to your accounts has a unique username and password combination, often with additional security measures. You might also have application accounts with their own access or secret keys for developer use. And remember, it's best practice to include multi-factor authentication (MFA) for every user.

Off the record

IAM is the application of Azure's role-based access control (Azure RBAC), where there are both Azure roles and Azure AD roles. Azure AD roles are used in cases like creating or editing users, assigning admin roles, resetting passwords, managing user licenses, and managing domains. Azure roles offer fine-grained access management to Azure resources:

- **Owners** have full access to all resources and the ability to delegate access to others.
- **Contributors** create and manage all types of Azure resources and create new tenants in Azure AD but can't grant access to others.
- **Readers** view resources.
- **User access managers** define access to Azure resources.

A lot can go wrong when implementing IAM—life is much easier if you implement robust IAM with the right policies to keep it clean. You run into problems when it creeps out of control and is held together by a thousand Confluence pages (our version of duct tape).

Managing and securing access is another key issue. Developers might lose access to their keys or have them stolen. MFA on root accounts and customized password rotations protect against this to some extent. But if you're a little too cavalier when using identity federation, a breached account somewhere else could lead to a breach to your Azure console. You can provide temporary access, but if you forget to disable it, you could end up increasing your surface area for breaches yet again.

Dive into offerings available in Azure for building, administrating, and working with identity and access management in our Identity and Access Management for Azure course.

Azure SQL family

Overview

A relational database service (RDS) is a managed database that controls everything—it abstracts the entire process of running and maintaining a database. You engage it only when you need to read or write data. CPU, memory, storage, and IOPS are split so you can scale them independently, bringing each one up or down. All of Microsoft's Azure SQL databases are relational databases outside of Azure Cosmos DB, which is NoSQL. Azure offers three options in its SQL family:

- **Azure SQL Database** is a core product that covers most bases and includes serverless compute.
- **Azure SQL Managed Instance** represents a fully managed SQL server instance hosted in the Azure cloud. Traditionally, service instances are infrastructure as an instance, but Microsoft uses this for a platform as a service (PaaS). This is based on the programming model of an on-premises SQL server and uses Microsoft support. It has high availability built in and allows you to back up and restore to Azure blob storage quickly.
- **SQL Server on Azure Virtual Machines** migrates workloads to Azure while maintaining SQL server compatibility and operating-system-level access. This can be helpful if you're in the middle of migration but still need to access it as if it were on-premises.

Microsoft also offers a service called Hyperscale, which can autoscale up to 100TB of storage. This works well for companies that want to migrate to the cloud or are limited by the max database size restrictions. But companies with smaller databases that require high performance and scaling options might also find it useful.

RDS tools try to lower the overall cost of ownership and strip away routine tasks, such as provisioning, backup, recovery, and other core requirements for any system, so you can focus on everything else that pertains to your business.

Off the record

Microsoft's SQL family represents a step function for companies looking to migrate to the cloud. Cloud-first companies might jump straight to a managed instance. Companies with traditional on-premises databases may already use Microsoft SQL but are starting their transition. As an example, one might look a bit like this:

On-premises Microsoft SQL
↓
Microsoft SQL in private clouds
↓
SQL server in Azure VMs
↓
Azure SQL database managed instances

You can pick from many options for the Azure SQL database, including MariaDB, MySQL, PostgreSQL, and Azure Cosmos DB.

RDS tools are not a magic bullet for getting everything else off your plate. You still have quite a bit to consider, like the type of database engine to use, how backups are handled, and how monitoring is managed. And when there's a lot to configure, especially when you're planning for a set-it and-forget-it tool, a great deal can go wrong.

Learn to leverage SQL in Azure, including selecting, updating, and deleting information and creating and altering database objects in our SQL Deep Dive course.

Azure Storage: Blob, Disk, and Files

Overview

Azure Blob Storage
A blob (binary large object) can be a video file or an image file that fits into a container you've created within Azure Storage. Each blob has a unique address, making it easy to retrieve items from containers. You could have a browser fetch images from a container, stream video or audio, or store any other kind of data.

Azure Blob Storage uses Azure Active Directory (Azure AD) for authentication and role-based access control for authorization. It's also suitable for producing static websites or creating a managed disk from a storage blob (e.g., if you're migrating from an on-premises solution). Azure Storage supports three types of blobs:

- **Block blobs:** blocks of data for storing text or binary files
- **Append blobs:** data like block blobs but optimized for append operations
- **Page blobs:** 512-byte pages up to 8TB ideal for storing a virtual hard drive or serving as a disk for a virtual machine (VM) on Azure

Azure Disk Storage
Azure also looks after the physical managed disk attached to your VMs and guarantees uptime and backup. You can also easily upgrade the disk size or type. There are multiple disk types, each with varying costs:

- **Hard disk drive (HDD):** old-school hard drives that are cheaper but slower than SSDs
- **Standard solid-state drive (SSD):** solid-state drives that have lower latency than an old-school drive
- **Premium SSD:** high-performance, low-latency storage for critical workloads
- **Ultra disk:** high-throughput, low-latency storage for the most I/O demanding workloads

Azure Files
Finally, Azure offers Azure Files for managing file shares, which extends on-premises file share configurations into Azure VMs.

Off the record

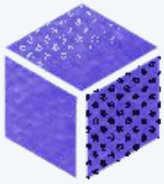
Microsoft offers so many options that you can outline a custom configuration based on how much storage you need and how often you need to access it instead of going for a one-size-fits-all approach. You're also able to save money if you know your exact storage needs and how frequently you'll use your files.

Azure Blob Storage offers three tiers depending on how often you need to access your data, each with a different cost structure:

- **Hot tier** for frequently accessed data
- **Cool tier** for lower storage costs and higher access times—meant for data that remains in the cool tier for at least 30 days
- **Archive tier** for the lowest cost and highest access time

Identify the different Azure Storage services, including Azure Blog Storage, Azure Files, Azure Disks, and more in our Azure Storage Deep Dive course.

AWS Amazon Elastic Compute Cloud (Amazon EC2) | **Google** Compute Engine



Azure Virtual Machines (VMs)

Overview

Virtual machines (VMs) are simulations of physical computers. They're servers but behave like computers, removing the need to manage the hardware. VMs are the workhorses for Azure, containing disk storage, processors, and operating systems, and the most expensive part of Azure deployment. The operating systems in Azure are Linux or Windows.

Azure VMs are the resources doing the actual work, like going through a process or serving a website for someone. It's where you store your data in memory when you're operating on it.

Off the record

Because a VM behaves like an actual computer, you have to treat it as a real computer . . . that you can't see. You have to secure and maintain your instances, manage the correct configurations, and build a set of VMs to achieve what you need with as little cost as possible.

Given that you can scale up the number of VMs you need with various configurations, you have to understand how each piece works and be able to track how many VMs you have running. Some VMs will be running at a constant pace. Others you need only in certain situations.

VMs sit within regional availability zones, so you need to determine where you want to drop your servers. With VMs, your users' physical distance matters because data exchange between regions within a billing zone is free. It costs more when going between different billing zones.

There are subscriptions based on credits for services. There's also a pay-as-you-go model, which can be the most expensive option since you're paying a premium for flexibility. Azure also has reserved instances, which can help you save money if you have a predictable set of needs.

Create and connect to an Azure virtual machine in our Deploying Your First Azure Virtual Machine lab.

Azure Load Balancer

Overview

Azure Load Balancer does exactly what it sounds like: balance the load of traffic across multiple different resources. Load balancing lets you scale up your infrastructure and services by balancing inbound and outbound traffic. For high availability, you want duplicate resources that serve the same purpose. A load balancer helps distribute that load across those duplicated resources.

Load balancers instantly reconfigure themselves when you scale your resources up and down. Health probes ensure the back-end pool of VMs is healthy and can receive traffic. If a health probe validates a server in your VNet, your load balancer can route traffic to that VM.

Off the record

Microsoft Azure has two additional types of load balancers: internal and external load balancers with basic or standard SKUs within the two.

The standard SKU supports up to 1,000 instances, while the basic SKU supports up to 100 instances in the back-end pool. Standard SKUs use a mixture of VMs, VM Scale Sets, and availability sets. You can also use HTTPS for health probes, while basic SKUs can use only HTTP and TCP. You can also choose the availability zone when using a standard SKU in some regions.

Internal load balancers direct traffic only to resources in a VNet or use a VPN to access Azure infrastructure. You might use an internal load balancer to balance traffic across VMs in the same VNet or from on-premises hardware to VMs on the same VNet.

Snapshots

Overview

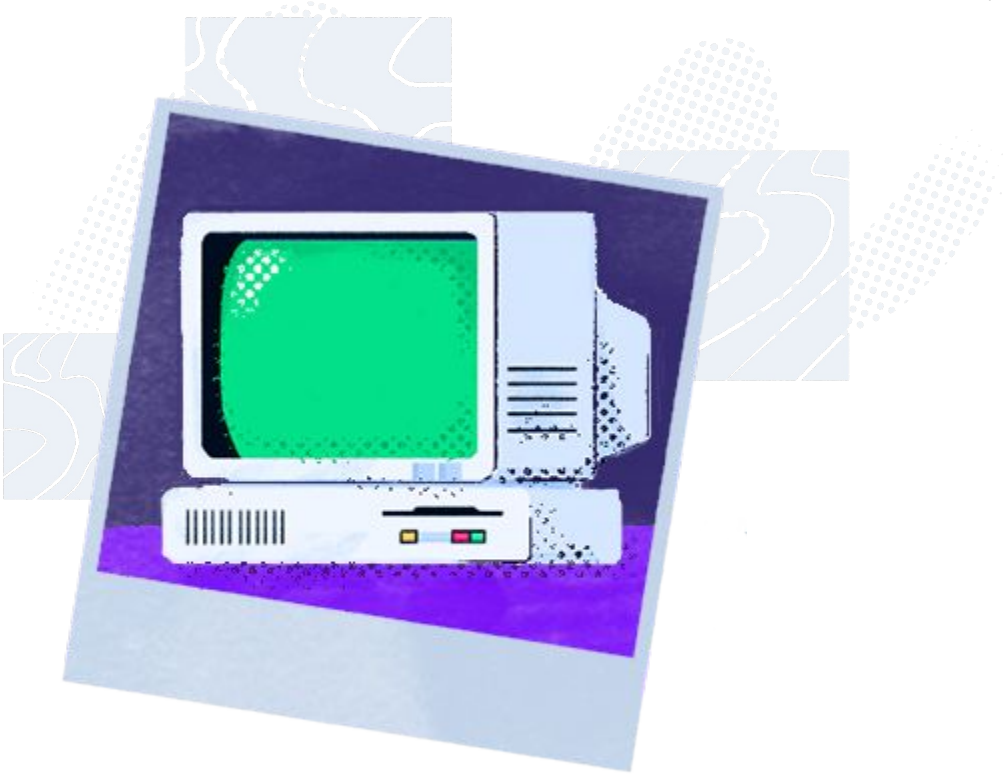
Snapshots are a point-in-time copy of a managed disk. Choose the account type, such as HDD or SSD, for whatever managed disk you want to save with a snapshot. These snapshots become backups for your managed disks.

You can also create a new managed disk based on a snapshot of another managed disk. You might use that snapshot for troubleshooting or as a master snapshot for creating new VMs.

Off the record

There are two kinds of snapshots: full snapshots and incremental snapshots. The former is exactly what it sounds like: a full, point-in-time backup of your disk. The downside? You'll quickly rack up storage costs.

Incremental snapshots are also backups, but not full backups. Each snapshot isn't a complete backup of your storage or disk. Instead, it stores the incremental changes from your last snapshot, and the previous snapshot captures the changes from the snapshot before it. You can save money with incremental snapshots because you're backing up incremental changes instead of creating a new full backup.



Get hands-on with our Create and Restore File Share Snapshots in Azure lab. Become a snapshot guru in 30 minutes after you take a snapshot of a file share and restore it to your Windows machine.

Azure Virtual Network (VNet)

Overview

Often called a VNet, virtual private networks enable Azure VMs to communicate with each other, on-premises networks, and on the internet. Just like a VM, it's yours to use, but the physical hardware is removed. You can define multiple subnets within each VNet. This segregates your network, allocating an IP space for resources such as SQL databases or VMs.

Each VNet comes with an address space that you can divide across multiple subnets. Perhaps more importantly, you can isolate subnets from the internet within your VNets.

VNets, and the Azure VMs within them, behave just like the Azure VMs within Microsoft's Azure cloud. You can scale VNets or add VNets as you need them and isolate resources within them through subnets. You can also peer VNets to allow routing between the two, while the Azure public cloud provides high availability.

The upside? Companies starting their migration to the cloud have an option to keep some of their resources walled off from the internet. While they might not need a VNet during the migration, it helps when trying to internally herd cats (namely your compliance team). VNets allow you to isolate information that should be separate from the internet, like regulated data.

Off the record

A whole lot is going on, with many points of failure—especially if you're trying to keep data isolated from the internet. The last thing you want to do is unintentionally increase your attack surface. If you're starting your cloud migration, you'll want to make sure you have a networking expert on hand to implement the custom routing and networking security group (NSG) rules to allow or block traffic coming in and out of your network.

In addition, subnets within your VNet are restricted to one availability zone rather than spanning multiple availability zones. You'll have to configure your systems to ensure you don't run into a scenario where one subnet needs to be talking to another in a different availability zone.

Get hands-on with our Configuring an Azure VNet-to-VNet VPN Gateway lab. Connect one VNet to another in an Azure resource group, then test connectivity between virtual machines located in each VNet.

About Pluralsight

Pluralsight helps organizations around the globe advance their technology workforce. Because the hardest part of building a business isn't building software and technology. It's building up the people who grow your business. That's why everyone from CIOs to developers trust Pluralsight—the only partner that helps leaders build better teams and better products, all at the same time.

Our software and solutions are purpose-built to address your top challenges and outcomes:

- Onboard new engineers faster
- Build products faster and improve the developer experience
- Develop internal cloud talent and enable cloud transformation
- Improve retention and cut hiring costs
- Improve cycle times and reduce burnout for remote teams
- Develop teams that deliver on key tech initiatives
- Increase delivery speed and overcome Agile roadblocks
- Hire job-ready, diverse talent
- Build fluency and collaboration organization-wide

Our cloud transformation solutions help you create the cloud talent you need, when you need it, to deliver on your biggest, boldest vision. Pluralsight Skills delivers expert-authored courses in the latest cloud technologies, paired with unlimited access to hands-on labs, sandboxes, and certification prep. Upskilling your teams with Skills equips your teams to execute on strategic cloud investments that ultimately drive innovation, automation, and efficiency.

PLURALSIGHT