**AZ-900 Domain Area**

**Weight**

Describe cloud concepts 20-25%

Describe core Azure services 15-20%

Describe core solutions and management tools on Azure 10-15%

Describe general security and network security features 10-15%

Describe identity, governance, privacy, and compliance features 20-25%

Describe Azure cost management and Service Level Agreements 10-15%

### [Azure Fundamentals part 1: Describe core Azure concepts](https://docs.microsoft.com/en-us/learn/paths/az-900-describe-cloud-concepts/)

1. **Azure** is a **cloud** **computing** platform with an ever-expanding set of services to help you build solutions to meet your business goals
2. **Azure services:**
   1. **Simple** **web** **services** for hosting your business presence in the cloud
   2. **Fully** **virtualized** **computers** for you to run your custom software solutions
3. Cloud-based services:
   1. remote storage
   2. database hosting
   3. centralized account management
4. **Azure** also offers **new** **capabilities** like **AI** and **Internet of Things (IoT)**

# What is cloud computing?

1. **Delivery** of computing **services** over the **internet**
2. These services include servers, **storage**, **databases**, **networking**, **software**, **analytics**, and **intelligence**
3. **Cloud** **computing** offers **faster** **innovation**, **flexible** **resources**, and **economies** of **scale**
4. PC is in the cloud, you pay for what you need.
5. **Add** and **remove** **compute** **power** as **needed**. Saves on cost.
6. Always on-line.

# Why is cloud computing typically cheaper to use?

1. **Pay**-**as**-**you**-**go** pricing **model**
2. Helps you:
   1. Lower your operating costs.
   2. Run your infrastructure more efficiently
   3. Scale as your business needs change
3. **Rent** **compute** **power** and **storage** from someone else's datacentre
4. You're billed only for what you use
5. **Instead** of **maintaining** **CPUs** and **storage** in **your** **datacenter**, you **rent** them for the time that you need them
6. **Provider** **maintains** the **underlying** **infrastructure** for you

# Why should I move to the cloud?

1. Helps you move **faster** and **innovate**
2. In our ever-changing digital world, two trends emerge:
   1. **Teams** **deliver** **new** **features** to their users at record speeds
   2. **Users** expect an **increasingly** **rich** and **immersive** **experience** with their devices and with software
3. Software not released in months but rather weeks.
4. **Releases** in **small** batches
5. Even multiple releases in a day?
6. The cloud provides on-demand access to:
   1. Limitless pool of raw **compute**, **storage**, and **networking** components
   2. **Speech** **recognition** and other **cognitive** **services**
   3. **Analytics** **services** that deliver **telemetry** **data**

# What is Azure?

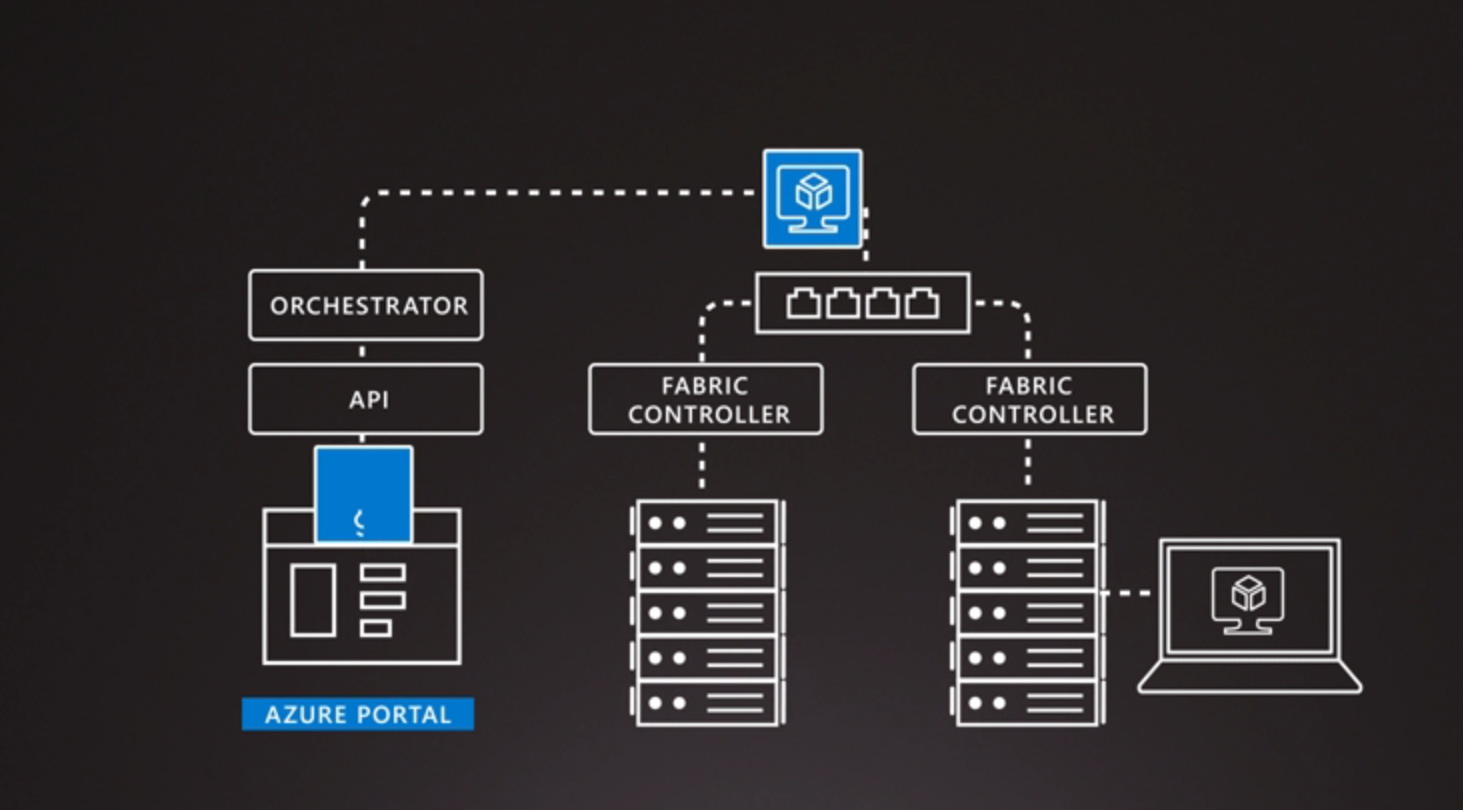
1. **Continually** **expanding** set of **cloud** **services**
2. Help your organization **meet** your **current** and **future** business **challenges**
3. Gives you the **freedom** to **build**, **manage**, and **deploy** applications on a massive global network **using** your **favourite** **tools** and **frameworks**

# What does Azure offer?

1. Benefits that Azure provides, so you can easily invent with purpose:
   1. **Be ready for the future:** Continuous innovation for today and future
   2. **Build on your terms:**  You can build how you want and deploy where you want to
   3. **Operate hybrid seamlessly:** Tools and services for a hybrid cloud solution
   4. **Trust your cloud:** Get security, backed experts, and proactive compliance trusted by many.

# What can I do with Azure?

1. Provides **more** than **100** **services**
2. Move **existing** **applications** to **virtual** **machines** that run in Azure
3. **Azure** provides **AI** and **machine**-**learning** **services** that can naturally communicate with your users through **vision**, **hearing**, and **speech**
4. **Storage** **solutions** that **dynamically** **grow**
5. Uses a **technology** called **Virtualization** – **Separates** **tight** **coupling** between **hardware** and **operating** system
6. Uses an **abstraction** **layer** called a **Hypervisor**
7. Each server has a hypervisor to run virtual machines
8. **Each** **server** also **runs** a **fabric** **controller**.
9. Also **connected** to an **Orchestrator** – **manages** **everything** that **happens** in **Azure**. Including handling user requests.
10. **User** make **requests** through **Orchestrators** **Web** **Api**. **Can** be **called** from the **Azure** **Portal**.



# What is the Azure portal?

1. A **web**-**based**, **unified** **console** that prov**i**des an **alternative** to **command**-**line** **tools**
2. Can **manage** your **Azure** **subscription** by using a **graphical** **user** **interface**
3. You can:
4. **Build**, **manage**, and **monitor** **simple** **web** **apps** to complex deployments.
5. Create **custom** **dashboards** for an organized view of resources.
6. Configure **accessibility** **options** for an optimal experience.
7. Designed for resiliency and continuous availability
8. Avoids network slowdowns by being close to users

# What is Azure Marketplace?

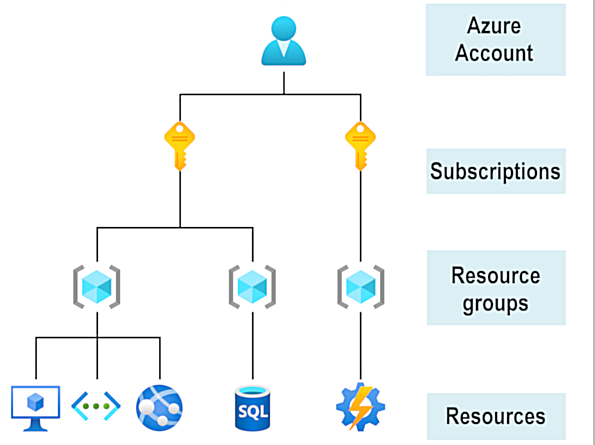
1. **Helps** **connect** users with **Microsoft** **partners**, independent **software** **vendors**, and **start-ups** that are offering **their** **solutions** and **services**
2. **Customers** can **find**, **try**, **purchase**, and provision applications and services from hundreds of leading service providers
3. Provision **end-to-end solutions quickly and reliably**, hosted in your own Azure environment
4. At the time of writing, there are more than 8,000 listings

# Azure Services

1. Azure brings a massive global infrastructure that's always available for you to build your applications on.
2. Services:
   1. **Compute**: Scale on demand. Pay for what you use. Use VM or scale for mobile.
   2. **Networking**: Connection. Using VPN and Load Balancing
   3. **Storage**: Disk, File, Blog or Archival storage. Scale as needed in a secure fashion.
   4. **Mobile**: Build and deploy cross platform and native apps. Send notifications and use Xamarin. Take advantage of cognitive services.
   5. **Databases**: Use a variety of DBs including open source. SQL, Cosmos DB and MySQL
   6. **Web**: Build, develop and scale web apps. Create web apps, api and use Azure Maps geo-spatial contexts.
   7. **IoT**: Analyse data from sensors.
   8. **Big Data**: Open source cluster services to help make decisions
   9. **AI**: Use data to forecast future behaviours. Use machine learning for deployment
   10. **DevOps**: Brings together people, processes and technology by automating software delivery. Create and deliver release pipelines
   11. For more: <https://docs.microsoft.com/en-us/learn/modules/intro-to-azure-fundamentals/tour-of-azure-services>
3. Services:
   1. **Compute**: **Azure** **Virtual** **Machines**, Azure Virtual Machine Scale Sets, **Azure** **Kubernetes** **Service**, Azure Service Fabric, Azure Batch, **Azure** **Container** **Instances**, Az**u**re **Functions**
   2. **Networking**: Azure Virtual Network, **Azure** **Load** **Balancer**, Azure **Application** **Gateway**, Azure VPN Gateway, Azure DNS, Azure Content Delivery Network, Azure DDoS Protection, Azure Traffic Manager, Azure ExpressRoute, Azure Network Watcher, Azure Firewall, Azure Virtual WAN
   3. **Storage**: **Azure** **Blob** storage, **Azure** **File** storage, **Azure** **Queue** storage, Azure **Table** storage. These services all share several common characteristics:
      1. **Durable**
      2. **Secure**
      3. **Scalable**
      4. **Managed**
      5. **Accessible**
   4. **Mobile**: **Back**-**end** **services** for **iOS**, **Android**, and **Windows** apps quickly and easily. Use resources such as **SAP**, **Oracle**, **SQL** **Server**, and **SharePoint**. Other features of this service include:
   5. **Databases**: Azure **Cosmos** DB, Azure **SQL** Database, Azure Database for MySQL, Azure Database for **PostgreSQL**, SQL Server on Azure Virtual Machines, Azure Synapse Analytics, Azure Database Migration Service, Azure Cache for Redis, Azure Database for MariaDB.
   6. **Web**: Azure **App** **Service**, Azure Notification Hubs, Azure API Management, Azure Cognitive Search, Web Apps feature of Azure App Service, Azure SignalR Service
   7. **IoT**: IoT Central, Azure IoT Hub, IoT Edge
   8. **Big Data**: Azure **Synapse** **Analytics**, Azure HDInsight, **Azure** **Databricks**
   9. **AI**: Azure Machine Learning Service, Azure ML Studio
      1. cognitive services:
         1. Vision
         2. Speech
         3. Knowledge mapping
         4. Bing Search
         5. Natural Language processing
   10. **DevOps**: Azure DevOps, Azure DevTest Labs

# Get started with Azure accounts

1. **You** **need** an **Azure** **subscription**
2. Learners can use learn **sandbox**
3. Azure account, and a subscription
4. **After** you've **created** an **Azure** **subscription**, you can **start** **creating** **Azure** **resources** within each subscription.



# Create an Azure account

1. You can **purchase** **Azure** access from **Microsoft** by signing up on the [Azure website](https://azure.microsoft.com/) or through a **Microsoft** **representative**
2. You can also purchase Azure access through a Microsoft partner
3. **Cloud** **Solution** **Provider** partners offer a range of complete managed-cloud solutions for Azure.

# Free Account

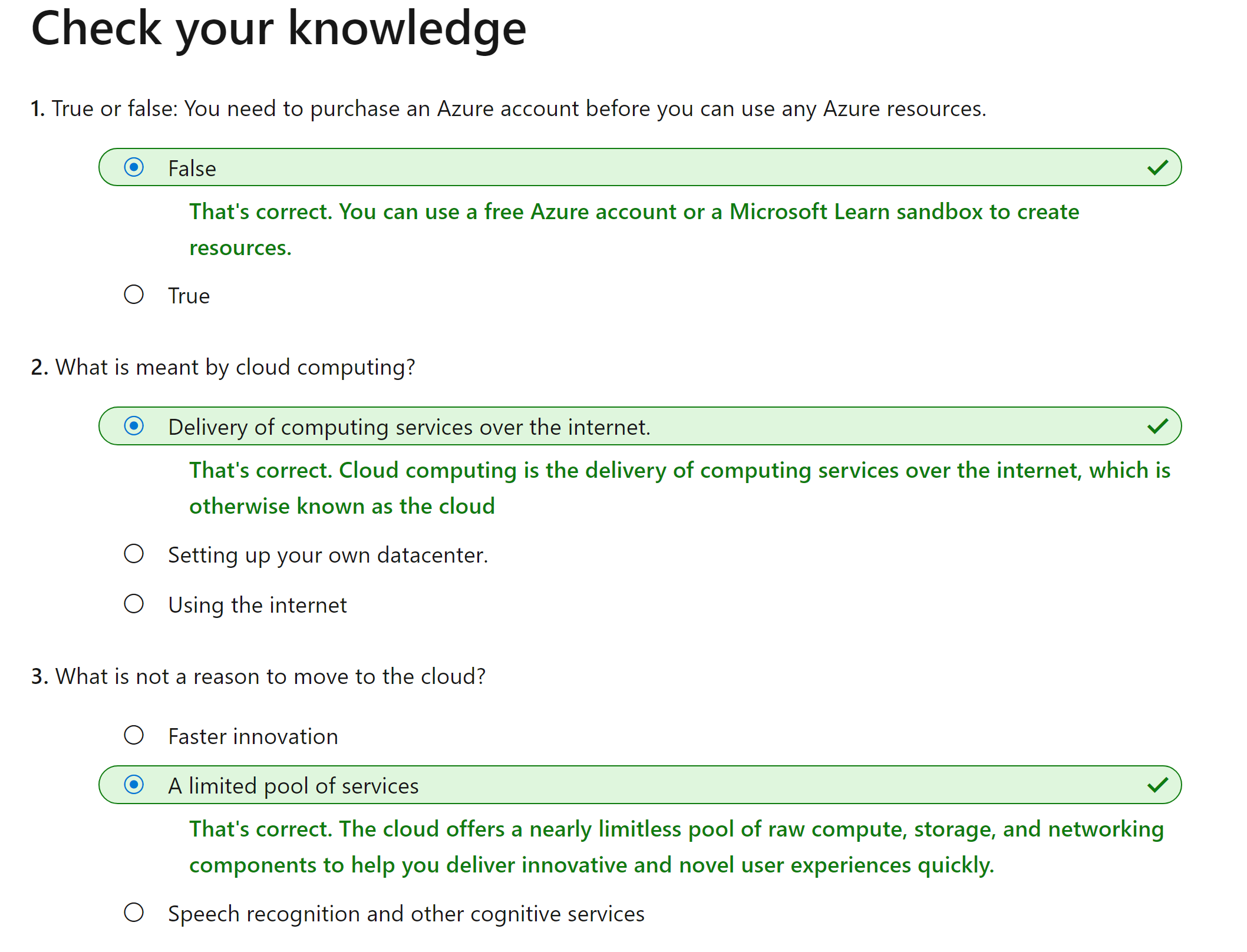
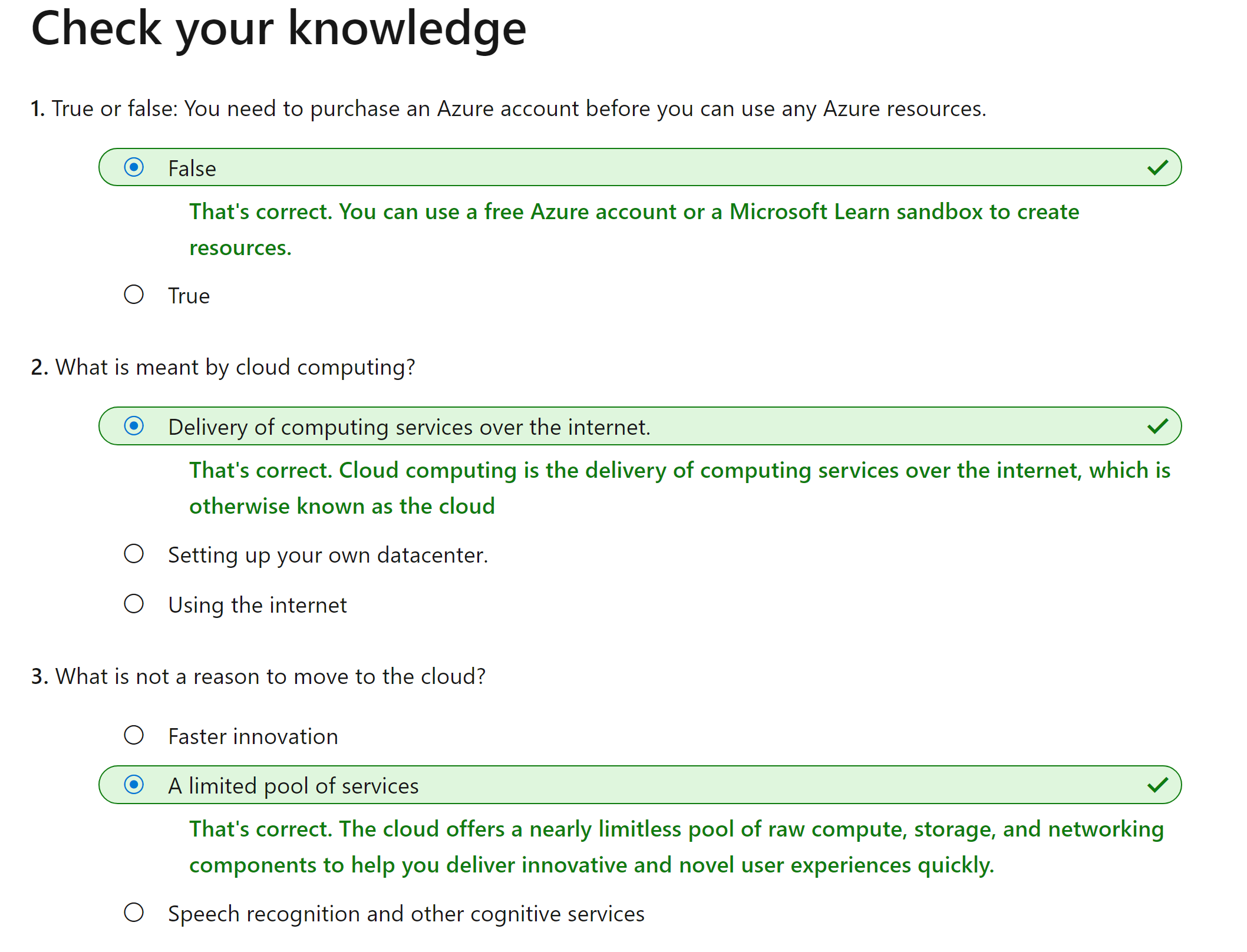
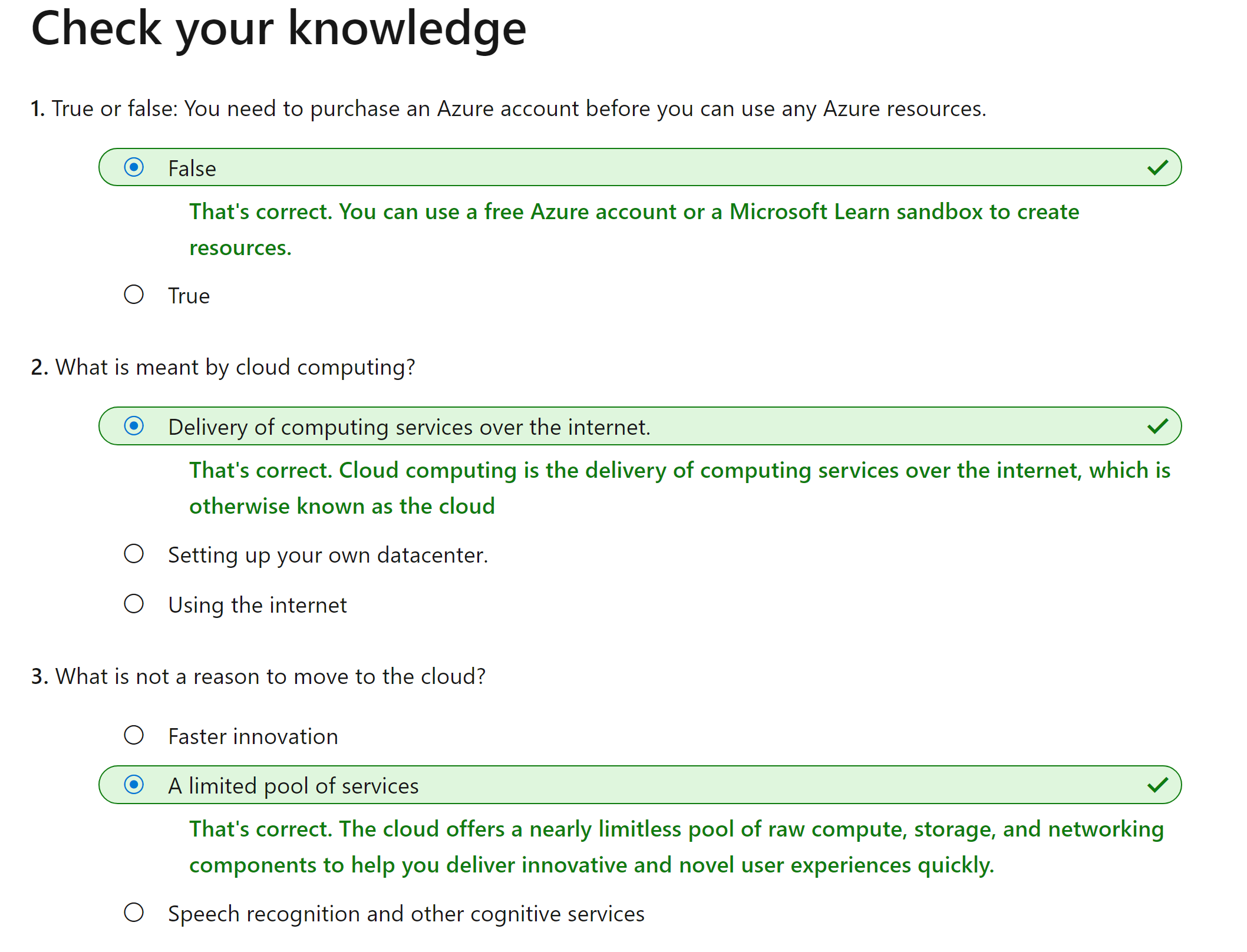
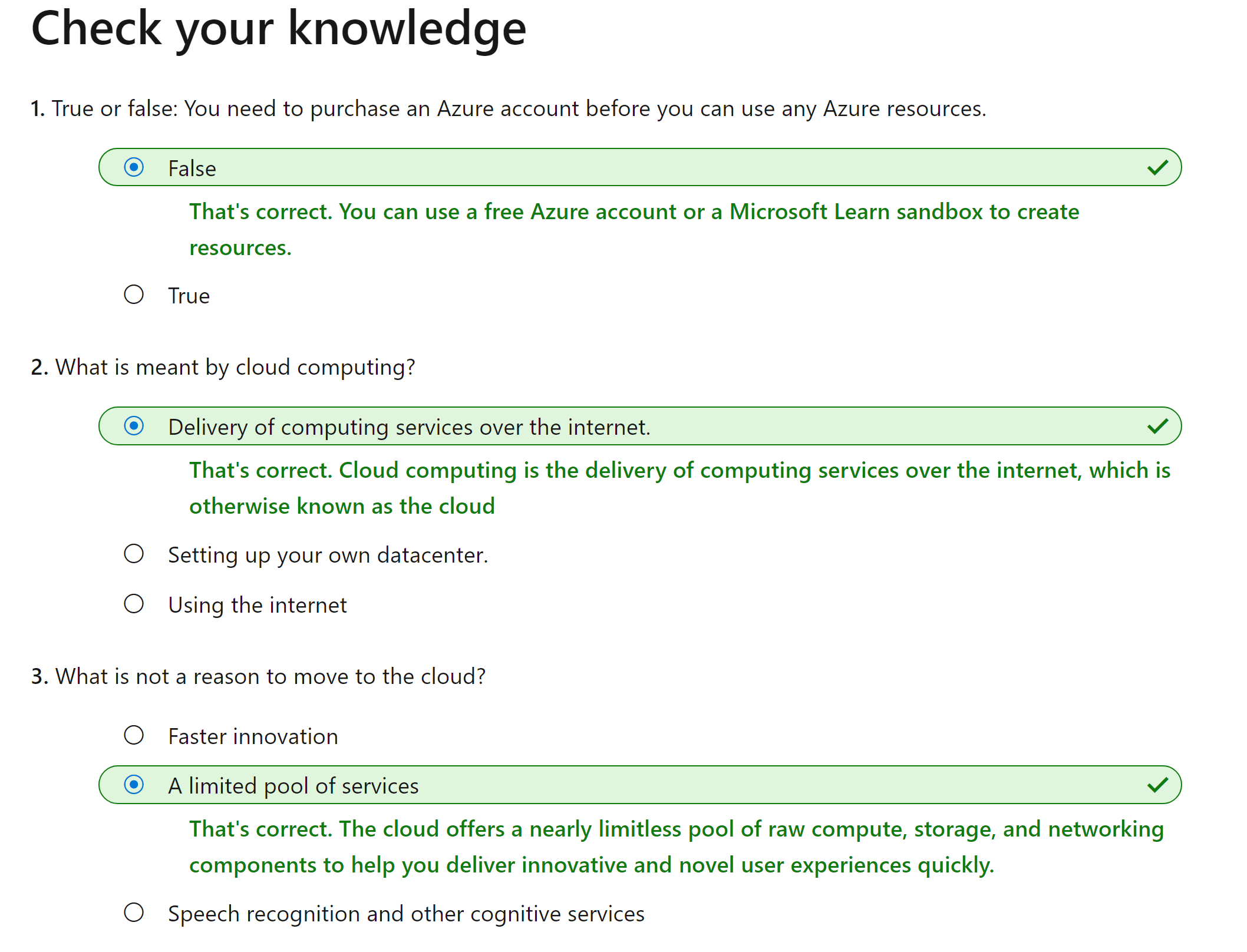
* Free access to certain Azure services for **12** **months**.
* **A credit to use in the first 12 months.**
* **Free** **access** to **certain** **software** **developer** **tools**.

# What is the Learn sandbox?

1. **Learn** **exercises** use a technology called the sandbox,
2. Learn automatically cleans up the temporary resources for you after you've completed the module

# Check You Knowledge

https://docs.microsoft.com/en-us/learn/modules/intro-to-azure-fundamentals/knowledge-check

Bottom of Form

# What are public, private, and hybrid clouds?

### **Public cloud:** Services are offered over the public internet and available to anyone who wants to purchase them

* **No** **capital** **expenditures** to **scale** **up**.
* **Applications** can be **quickly** **provisioned** and **deprovisioned**.
* **Organizations** pay only for what they **use**.

### **Private cloud:** A private cloud consists of computing resources used exclusively by users from one business or organization

* **Hardware** must be **purchased** for start-up and maintenance.
* **Organizations** have **complete** **control** over **resources** and **security**.
* **Organizations** are **responsible** for **hardware** **maintenance** and updates.

### **Hybrid cloud:**  Combines a public cloud and a private cloud

* **Provides** the **most** **flexibility**.
* **Organizations** **determine** where to run **their** **applications**.
* **Organizations** **control** **security**, **compliance**, or **legal** **requirements**.

# What are some cloud computing advantages?

1. **High availability:** Depending on the service-level agreement (SLA)
2. **Scalability**
   1. Scale vertically - increase compute capacity by adding RAM or CPUs
   2. Scaling horizontally -  adding VMs to the configuration
3. **Elasticity**
   1. *Auto scaling*, so your apps always have the resources they need.
4. **Agility:** Deploy and configure cloud-based resources quickly
5. **Geo-distribution**
6. **Disaster recovery: Backup services, disaster recovery**

# Capital expenses vs. operating expenses

1. **Capital Expenditure (CapEx)** - **up-front spending** of **money** on **physical** **infrastructure**, and then **deducting** that up-front expense **over** **time**. The up-front cost from CapEx has a value that reduces over time.
2. **Operational Expenditure (OpEx): spending** **money** on **services** or **products** **now**, and being **billed** for them **now**. There is **no** **up**-**front** **cost**, as you pay for a service or product as you use it.

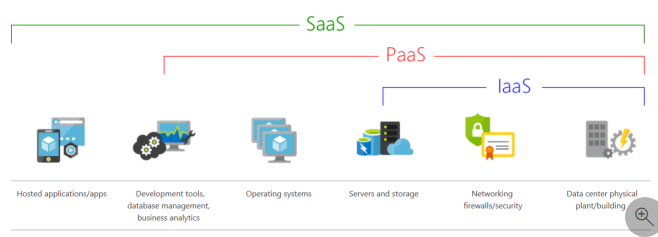
**Cloud** **services**, on the other hand, are categorized as an **OpEx**, because of their **consumption** **model**.  As a result, **OpEx** has a di**r**ect **impact** on **net** **profit**, **taxable** **income**, and the associated **expenses** on the balance sheet.

# Cloud computing is a consumption-based model

1. **Consumption**-**based** **model**, which means that end users only **pay** for the **resources** that they **use**
2. A consumption-based model has many benefits:
   1. **No** upfront **costs**.
   2. **No** need to **purchase** and **manage** costly **infrastructure**
   3. **Pay** for **additional** res**o**urces when they are **needed**
   4. **Ability** to **stop** **paying** for resources

# What are cloud service models?

1. These models define the different levels of shared responsibility that a cloud provider and cloud tenant are responsible for.
2. **IaaS - Cloud provider** will **keep** the **hardware** **up**-**to**-**date**. **Operating** **system** **maintenance** and **network** **configuration** is up to **you**. E.g Azure Virtual Machines.
3. **PaaS - Managed** **hosting** **environment**. **Cloud** **provider** **manages** the **virtual** **machines** and **networking** **resources**. **Tenant** **deploys** their **applications** into the **managed** **hosting** **environment**. E.g Azure App Services – **Developers** can **upload** **applications** **without** **worrying** about **physical** **hardware**.
4. **SaaS - Cloud** **provider** **manages** all **aspects** of the **application** **environment**. Such as virtual machines, networking resources, data storage, and applications. Tenant only needs to provide their data to the application managed by the cloud provider. For example **Microsoft Office 365**



# IaaS

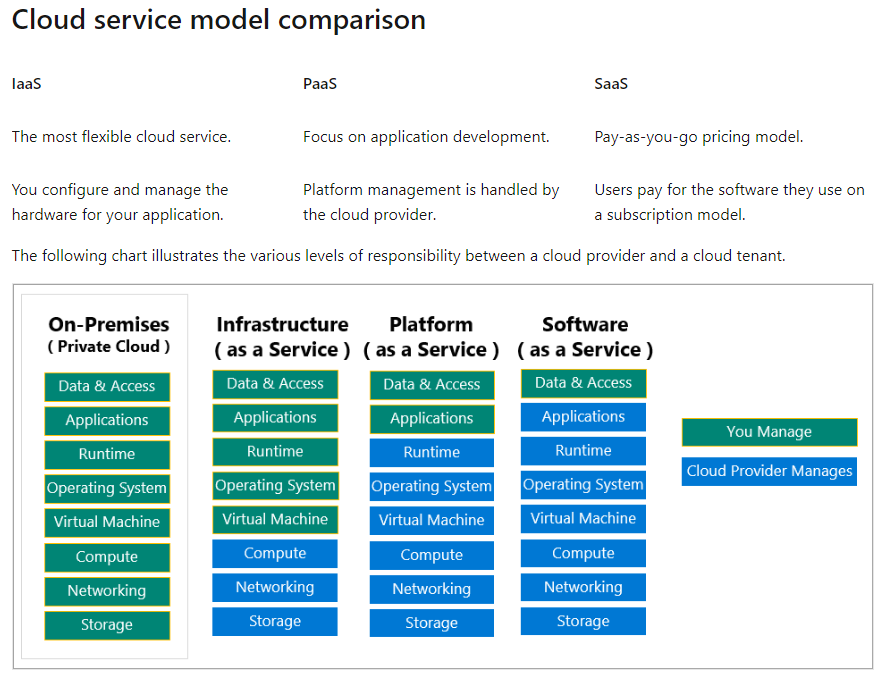
1. The **most** **flexible** category of cloud services
2. Gives you **complete** **control** over the hardware that runs your application. Instead of buying hardware, with IaaS, you rent it.
3. **Advantages: No CapEx, Agility, Management, Consumption-based model, Skills, Cloud benefits, Flexibility.**

# PaaS

1. Same benefits and considerations as **IaaS**
2. **Advantages: No CapEx, Agility, Consumption-based model, Skills, Cloud benefits, Productivity**
3. **Disadvantage:** **Platform limitations - Limitations** to a cloud platform that **might** **affect** how an **application** runs

# SaaS

1. Software that's centrally hosted and managed for you and your users or customers
2. Licensed through a monthly or annual subscription
3. SaaS provides the same benefits as IaaS
4. **Advantages: No CapEx, Agility, Pay-as-you-go pricing model, Skills, Flexibility**
5. **Flexibility: Software limitations -** Affect how users work, you don't have direct control of features. When you're evaluating which SaaS platform is best suited for a workload, be sure to consider any business needs and software limitations.



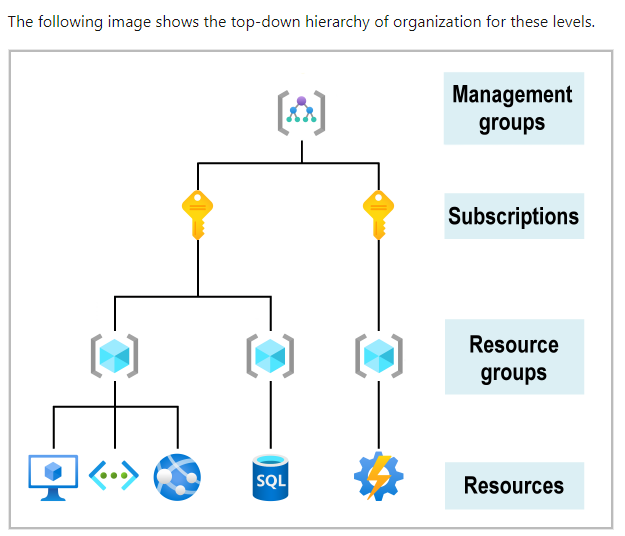
# What is serverless computing?

1. Like **PaaS**, **serverless** **computing** enables developers to **build** **applications** **faster** by eliminating the need for them to **manage** **infrastructure**
2. **Cloud** **service** **provider** **automatically** provisions, scales, and **manages** the **infrastructure** required to run the **code**
3. **Serverless** **architectures** are **highly** **scalable** and event-driven, **only** **using** **resources** when a specific function or **trigger** occurs.
4. "**Serverless**" **name** comes from the fact that the tasks associated with infrastructure provisioning and management are **invisible** to the **developer**.
5. This **approach** enables **developers** to **increase** **focus** on the **business** **logic**, and deliver more value
6. Helps teams **increase** their **productivity** and bring products to **market** **faster**

# Knowledge Check

https://docs.microsoft.com/en-us/learn/modules/fundamental-azure-concepts/knowledge-check

# Overview of Azure subscriptions, management groups, and resources



1. **Resources:** Are instances of services like  virtual machines, storage, or SQL databases
2. **Resource groups: Logical** **container** into which Azure resources like web apps, databases, and storage accounts are deployed and managed
3. **Subscriptions: Groups** **together** **user** **accounts** and the resources. For each subscription, there are limits or quotas on the amount of resources that you can create and use.
4. **Management groups:** Helps you **manage** **access**, **policy**, and compliance for multiple **subscriptions**.  All subscriptions in a management group automatically inherit the conditions applied to the management group.

# Azure regions, availability zones, and region pairs

1. **Resources** are **created** in **regions**, which are different geographical locations around the globe that contain Azure datacenters
2. When you use a service or create a resource such as a SQL database or virtual machine (VM), you're **using** **physical** **equipment** in one or more of these **locations**

# Azure regions

1. A **region** is a **geographical** **area** on the **planet** with multiple datacentres that are nearby and networked together with a low-latency network
2. When you **deploy** a **resource** in **Azure**, you'll often need to choose the **region**
3. Some services or VM features are only available in certain regions, such as **VM** **sizes**.
4. **Global** **Azure** **services** that don't require you to select a particular region: **Azure** **Active** **Directory**, **Azure** **Traffic** **Manager**, and **Azure** **DNS**

# Why are regions important?

1. **Global** regions provide better **scalability** and **redundancy**
2. They also **preserve** **data** **residency** for your services

# Special Azure regions

1. **US DoD Central, US Gov Virginia, US Gov Iowa and more:**
2. **China East, China North, and more:** partnership between Microsoft and 21Vianet, whereby Microsoft doesn't directly maintain the datacenters

# Azure availability zones

1. **Ensure** your **services** and **data** are **redundant** so you can protect your information in case of **failure**
2. **Azure** can help make your app **highly** **available** through **availability** **zones**

# What is an availability zone?

1. **Availability** **zones** are **physically** **separate** **datacentres** within an Azure region
2. **Availability** **zone** is made up of one or more **datacenters** **equipped** with **independent** **power**, **cooling**, and **networking**
3. An **availability** **zone** is set up to be an **isolation** **boundary**
4. If **one** **zone** **goes** **down**, the **other** continues **working**

# Use availability zones in your apps

1. Use **availability** **zones** to run **mission**-**critical** **applications** and build high-availability into your application
2. Co-locating your compute, storage, networking, and data resources within a zone and replicating in other zones
3. **Cost** to **duplicating** your **services** and **transferring** to zones.
4. Azure services that support availability zones fall into two categories:
   1. **Zonal services**
   2. **Zone-redundant services**

# Azure region pairs

1. **An** **outage** **big** enough to **affect** even **two** **datacenters**. That's why Azure also creates **region** **pairs**.

# What is a region pair?

1. **Each** **Azure** **region** is always **paired** with **another** **region** **within** the **same** **geography** (such as US, Europe, or Asia) at least 300 miles away
2. Approach **allows** for the **replication** of **resources**
3. Reduce the likelihood of interruptions because of events such as natural disasters, civil unrest, power outages, or physical network outages
4. **Additional advantages of region pairs: Restored** as **quickly** as **possible** for applications hosted in that region pair, **Updates** are **rolled** **out** **one** **region** at a **time** to **minimize** **downtime**, **Data** **continues** to **reside** within the **same** **geography** as its pair for **law**-**enforcement** **jurisdiction** purposes

# Azure resources and Azure Resource Manager

1. After you've created a subscription you're ready to start creating resources and storing them in resource groups
2. It's important to define those terms:
   1. **Resource:** Virtual machines (VMs), storage accounts, web apps, databases, and virtual networks are examples
   2. **Resource group:** A container that holds related resources for an Azure solution

# Azure resource groups

1. A **resource** **group** is a **logical** **container** for resources deployed on Azure
2. **Resources** are **anything** you **create** in an **Azure** subscription like VMs, Azure Application Gateway instances, and Azure Cosmos DB instances
3. **Resource** can **only** be a **member** of a **single** **resource** **group**.
4. **Many** **resources** can be **moved** between resource **groups**
5. Resource groups **can't** be **nested**
6. **Before** any **resource** can be **provisioned**, **you** **need** a **resource** **group** for it to be placed in

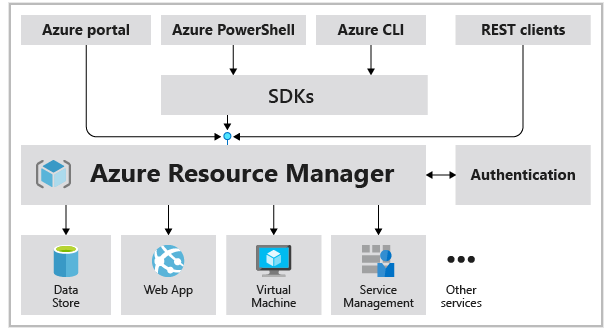
# Life cycle

1. If you **delete** a **resource** **group**, **all** **resources** contained within it are also **deleted**

# Authorization

1. **Resource** **groups** are **also** a scope for **applying** role-based access control (**RBAC**) **permission**

# Azure Resource Manager

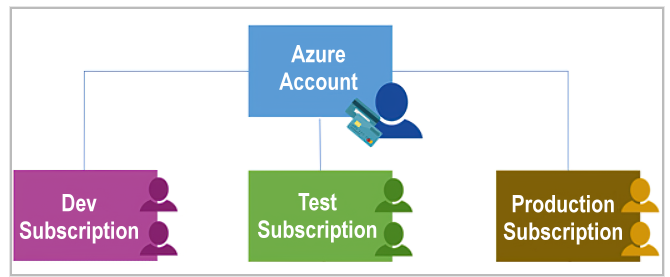
1. Azure Resource Manager is the deployment and management service
2. It **provides** a **management** **layer** that **enables** you to **create**, **update**, and delete **resources**
3. Features like access control, locks, and tags to secure and organize your resources after deployment
4. **When** a **user** **sends** a **request** from any of the **Azure** **tools**, APIs, or SDKs, Resource **Manager** **receives** the **request**. It **authenticates** and **authorizes** the request
5. Resource Manager sends the request to the Azure service, which takes the requested action
6. **Because** all **requests** are **handled** through the **same** **API**, you see consistent results

# The benefits of using Resource Manager

1. **Manage** your **infrastructure** through declarative **templates** rather than scripts. Resource Manager **Template** is a **JSON** **file**.
2. **Deploy**, **manage**, and **monitor** all the **resources** for your solution as a group, rather than handling these resources individually.
3. **Redeploy** your **solution** throughout the development life cycle and have confidence your resources are deployed in a consistent state.
4. **Define** the **dependencies** between resources
5. **Apply** **access** **control** to all services
6. Apply **tags** to resources to logically organize
7. **Clarify** your organization's **billing** by viewing **costs** for a group of resources

# Azure subscriptions

1. **Using** **Azure** **requires** an Azure **subscription**
2. **Subscription** **provides** you with **authenticated** and **authorized** access to **Azure** **products** and **services**
3. Allows you to provision resources
4. **Azure** **subscription** is a **logical** **unit** of **Azure** **services** that links to an Azure account which is an identity in Azure Active Directory (Azure AD)



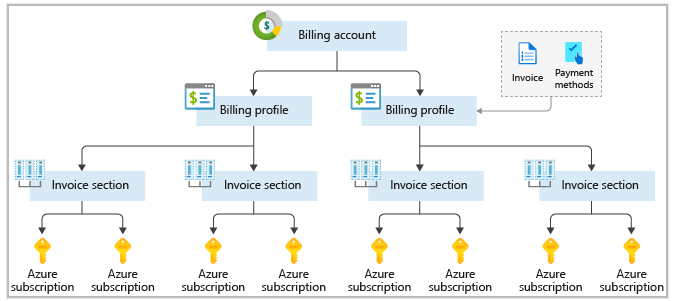
1. An **account** can have **one** **subscription** or **multiple** **subscriptions** that have different billing models
2. There are two types of subscription boundaries that you can use:
   1. **Billing boundary**: This subscription type determines how an Azure account is billed
   2. **Access control boundary**: You have **different** **departments** to which you apply distinct Azure subscription policies

# Create additional Azure subscriptions

1. You might choose to create additional subscriptions to separate:
   1. **Environments:** Set up separate environments for development and testing, security, or to isolate data for compliance reasons
   2. **Organizational structures:** create subscriptions to reflect different organizational structures. For example lower cost resources
   3. **Billing:** You might want to also create additional subscriptions for billing purposes.
   4. **Subscription limits:**  Subscriptions are bound to some hard limitations

# Customize billing to meet your needs

1. If you have **multiple** **subscriptions**, you can **organize** them into **invoice** **sections**
2. You might need a single invoice for your organization but want to organize charges by department, team, or project
3. You can set up **multiple** **invoices** within the **same** **billing** **account**
4. Each billing profile has its own monthly invoice

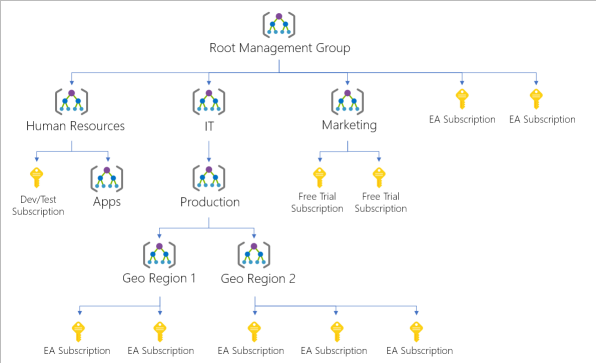


# Azure management groups

1. **Azure** **management** **groups** provide a level of scope above subscriptions
2. You organize **subscriptions** into **containers** called **management** **groups** and apply your governance conditions to the management groups
3. All **subscriptions** within a management group automatically inherit the conditions applied to the management group
4. All **subscriptions** within **must** **trust** the **same** **Azure** **AD** **tenant**.

# Hierarchy of management groups and subscriptions

1. The following diagram shows an example of creating a hierarchy for governance by using management groups:



1. Important facts about management groups:
   1. **10,000** **management** **groups** can be **supported** in a **single** **directory**
   2. A **management** **group** **tree** can support up to **six** levels of depth
   3. Each **management** **group** and **subscription** can support only **one** **parent**
   4. Each **management** **group** can have **many** **children**.
   5. All **subscriptions** and **management** **groups** are within a **single** **hierarchy** in each directory.

# AZURE EXERCISE

# What is App Service?

1. **HTTP**-**based** **service** that enables you to **build** and **host** many types of **web**-**based** **solutions** without managing infrastructure
2. For example, you can host web apps, mobile back ends, and RESTful APIs in several supported programming languages.
3. .**NET**, .**NET** **Core**, **Java**, **Ruby**, **Node**.**js**, **PHP**, or **Python** can run in and scale with ease on both Windows-based and Linux-based environments

# Knowledge Check

https://docs.microsoft.com/en-us/learn/modules/azure-architecture-fundamentals/knowledge-check