AZ-305T00A Designing Microsoft Azure Infrastructure Solutions

Design a compute solution

https://learn.microsoft.com/training/modules/design-compute-solution/

Learning Objectives

- Choose a compute service
- Design for Azure virtual machine solutions
- Design for Azure Batch solutions
- Design for Azure App Services solutions
- Design for Azure Container Instances solutions
- Design for Azure Kubernetes Service solutions
- Design for Azure Function solutions
- Design for Azure Logic App solutions
- Case study
- Learning recap

AZ-305: Design Infrastructure Solutions (30-35%)

Design compute solutions

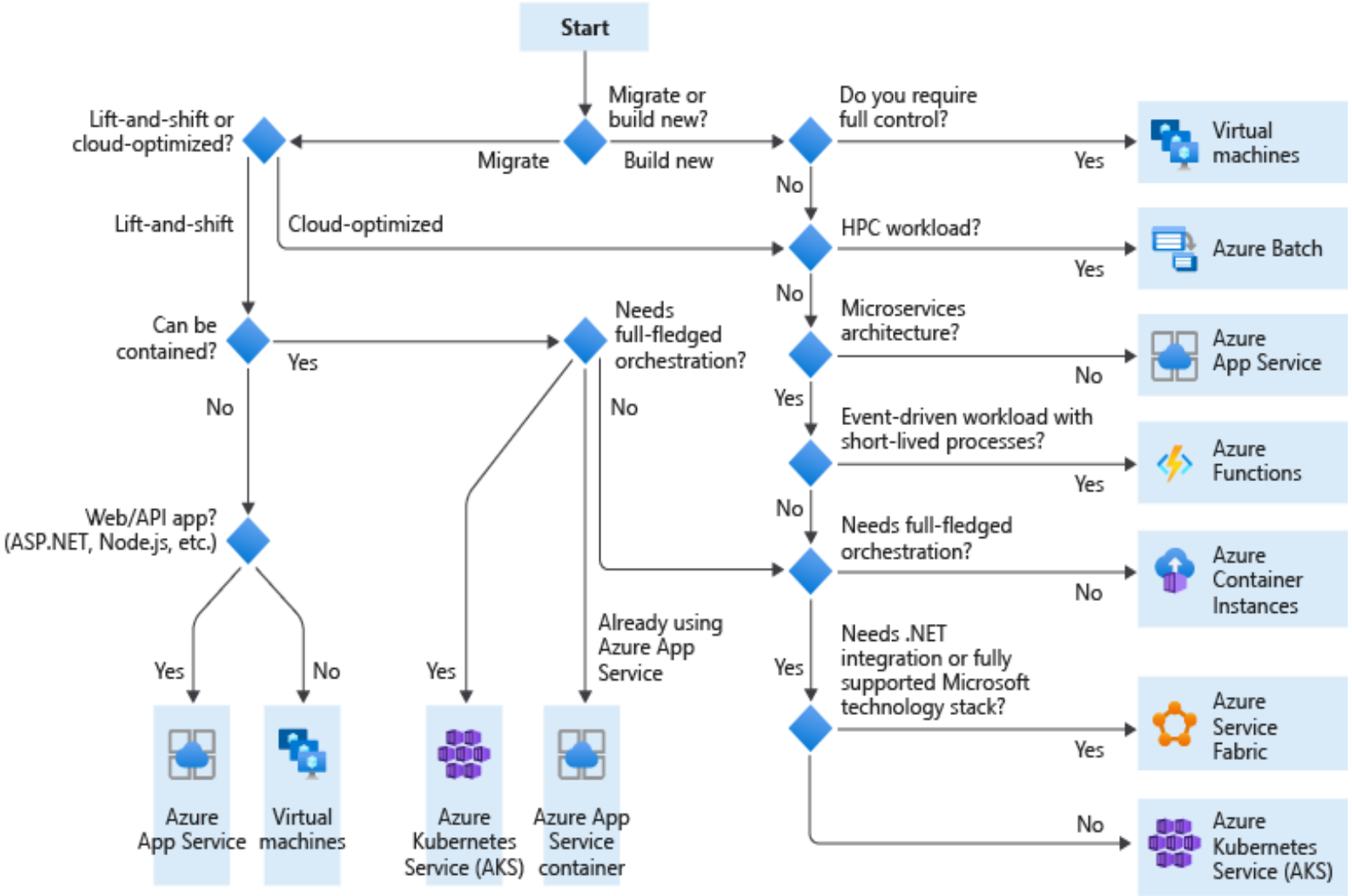
- Specify components of a compute solution based on workload requirements
- Recommend a virtual machine-based solution
- Recommend a container-based solution
- Recommend a serverless-based solution
- Recommend a compute solution for batch processing

Choose a compute solution

Choose a compute service for your application

Azure offers several compute services.

Compute refers to the hosting model for the computing resources that your applications run on.



ⓘ Note

The following diagram has been edited to show only the Azure services described in this module.

Things to know about Azure compute services

Let's take a quick look at the Azure compute services we review in this module.

- **Azure Virtual Machines**: Deploy and manage virtual machines inside an Azure virtual network.
- **Azure Batch**: Apply this managed service to run large-scale parallel and high-performance computing (HPC) applications.
- **Azure App Service**: Host web apps, mobile app backends, RESTful APIs, or automated business processes with this managed service.
- **Azure Functions**: Use this managed service to run code in the cloud, without worrying about the infrastructure.
- **Azure Logic Apps**: Configure this cloud-based *platform* to create and run automated workflows similar to capabilities in Azure Functions.
- **Azure Container Instances**: Run containers in Azure in a fast and simple manner without creating virtual machines or relying on a higher-level service.
- **Azure Kubernetes Service (AKS)**: Run containerized applications with this managed Kubernetes service.

Things to consider when choosing Azure compute services

As you begin to compare Azure compute services to choose your infrastructure solution for Tailwind Traders, there are several implementation points to think about.

- Architecture and infrastructure requirements
- Support for new workload scenarios, like HPC applications
- Required hosting options, including platform, infrastructure, and functions
- Support for migrations, such as cloud-optimized or lift and shift

Workloads and architecture

When you plan for new instances of Azure services and new workloads, consider the following scenarios.

- **Control**: Determine if you require full control over installed software and applications.
- **Workloads**: Consider the workloads you need to support, such as **HPC workloads** or **event-driven workloads**.
- **Architecture**: Think about what architecture best supports your infrastructure, including **microservice**, **full-fledged orchestration**, and **serverless**.

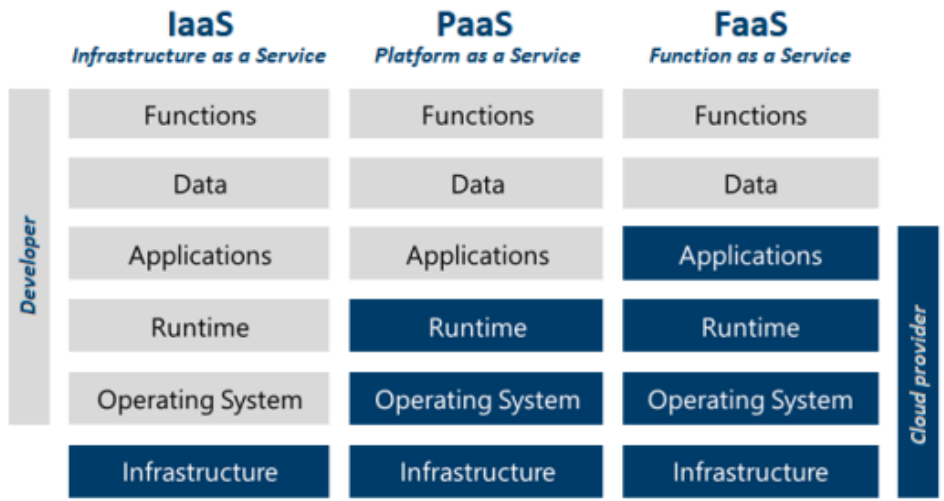
Migrations

An important consideration for your compute service involves analyzing the migration capabilities.

- **Cloud optimized**: To migrate to the cloud and refactor applications to access cloud-native features, consider compute services that are cloud-optimized.
- **Lift and shift**: For lift and shift workload migrations, consider compute services that don't require application redesigns or code changes.
- **Containerized**: In your migration planning, consider whether your compute service needs to support containerized applications, or commercial off the shelf (COTS) apps.

Hosting

The hosting option of your compute solution determines the developer and cloud provider responsibilities. Azure offers three hosting options across the compute services.



- Infrastructure-as-a-Service (IaaS) lets you create individual virtual machines along with the associated networking and storage components. Then you deploy the software and applications you want onto those virtual machines. This model is the closest to a traditional on-premises environment, except that Microsoft manages the infrastructure. You still manage the individual virtual machines. Azure Virtual Machines offers IaaS hosting.

- **Platform-as-a-Service (PaaS)** provides a managed hosting environment, where you can deploy your application without needing to manage virtual machines or networking resources. Azure compute services that offer PaaS hosting include Azure Batch, App Service, Container Instances, and Azure Kubernetes Service.

- **Function-as-a-Service (FaaS)** goes further in removing the need to worry about the hosting environment. In a FaaS model, you deploy your code, and the service automatically runs it. Azure Functions and Logic Apps offer FaaS hosting.

Design for Azure virtual machine solutions

Azure Virtual Machines is the basis of the Azure Infrastructure-as-a-Service (IaaS) model. Virtual Machines can be used for developing, testing, and deploying applications in the cloud, or to extend your datacenter. Virtual Machines offers a fast, scalable, flexible way to add more compute power to your enterprise.

When to select virtual machines

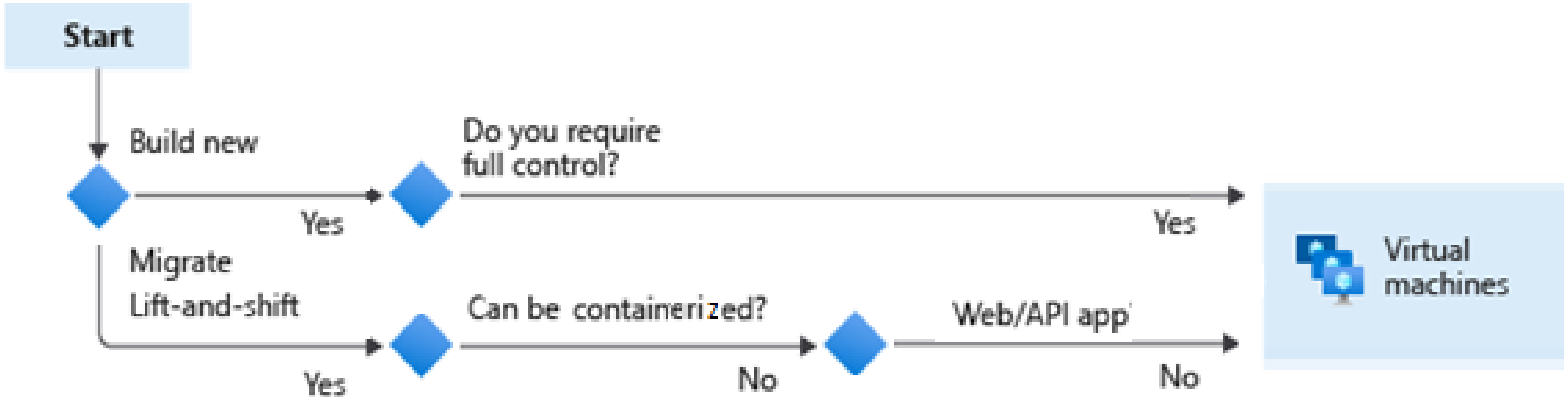
Know when to use virtual machines

- Quickly test and troubleshoot different configurations
- Use specialty hardware such as high-performance computing
- Extend your on-premises datacenter without purchasing additional hardware
- Run legacy apps on modern hardware
- Access third party software that requires local add-ons or plug-ins
- Quickly migrate apps to the cloud
- Fully control the computing environment

Plan the Azure virtual machine deployment

Azure virtual machines scenarios include build new or migrate patterns

- Start with the network
- Name the VM
- Decide the VM location
- Select the VM storage
- Select an VM operating system
- Keep the VM up to date
- Monitor the VM



•**Build new workloads**: Azure Virtual Machines is ideal when you're building new workloads and demand for your applications can fluctuate. It's economical to run your applications on a virtual machine in Azure.

•**Lift and shift migration**: If you're using lift and shift (rehosting) migration to move data and applications from an on-premises location, targeting Azure Virtual Machines in the cloud is an effective strategy.

Determine the virtual machine family

The virtual machine size determines pricing

General purpose

Compute optimized

Memory optimized

Storage optimized

GPU

High performance compute

Classification	Description	Scenarios
General purpose	General-purpose virtual machines are designed to have a balanced CPU-to-memory ratio.	- Testing and development - Small to medium databases - Low to medium traffic web servers
Compute optimized	Compute optimized virtual machines are designed to have a high CPU-to-memory ratio.	- Medium traffic web servers - Network appliances - Batch processes - Application servers
Memory optimized	Memory optimized virtual machines are designed to have a high memory-to-CPU ratio.	- Relational database servers - Medium to large caches - In-memory analytics
Storage optimized	Storage optimized virtual machines are designed to have high disk throughput and I/O.	- Virtual machines running databases
GPU	GPU virtual machines are specialized virtual machines targeted for heavy graphics rendering and video editing.	- Model training and inferencing with deep learning
High performance computes	High performance compute offers the fastest and most powerful CPU virtual machines with optional high-throughput network interfaces.	- Workloads that require fast performance - High traffic networks

https://azure.microsoft.com/pricing/vm-selector/

Virtual machine pricing

A subscription is billed two separate costs for every virtual machine: *compute* and *storage*. By separating these costs, you can 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. If the virtual machine is deployed for 55 minutes, you're charged for only 55 minutes of usage. You're not charged for compute capacity if you stop and deallocate the virtual machine. The hourly price varies based on the virtual machine size and operating system you select.
- **Storage costs**: You're charged separately for the Azure Storage the virtual machine uses. The status of the virtual machine has no relation to the Azure Storage charges that are incurred. You're always charged for any Azure Storage used by the disks.

Azure Storage

Azure Managed Disks handle Azure storage account creation and management in the background for you. You specify the disk size and the performance tier (Standard or Premium). Azure creates and manages the disk. As you add disks or scale the virtual machine up and down, you don't have to worry about the storage being used.

Operating system

Azure provides various operating system images that you can install into the virtual machine, including several versions of Windows and flavors of Linux. Azure bundles the cost of the operating system license into the price.

- If you're looking for more than just base operating system images, you can search Azure Marketplace. There are various install images that include not only the operating system but popular software tools, such as WordPress. The image stack consists of a Linux server, Apache web server, a MySQL database, and PHP. Instead of setting up and configuring each component, you can install an Azure Marketplace image and get the entire stack all at once.
- If you don't find a suitable operating system image, you can create your own disk image. Your disk image can be uploaded to Azure Storage and used to create an Azure virtual machine. Keep in mind that Azure only supports 64-bit operating systems.

When to select virtual machine scale sets

Scale sets are built from virtual machines. With scale sets, the management and automation layers are provided to run and scale your applications.

Scenario	Group of virtual machines	Virtual machine scale sets*
You need to add VM instances for changing workload	Manual process to create, configure, and ensure compliance	Automatically create from central configuration
You need to balance and distribute workloads	Manual process to create and configure Azure load balancer or Application Gateway	Can automatically create and integrate with Azure load balancer or Application Gateway
You need high availability and redundancy	Manually create Availability Set or distribute and track VMs across Availability Zones	Automatic distribution of VM instances across Availability Zone (when zones are selected) and Availability Set
You need to monitor and then scale virtual machines	Manual monitoring and Azure Automation	Autoscale based on host metrics, in-guest metrics, Application Insights, or schedule

* Ensure application supports VMSS

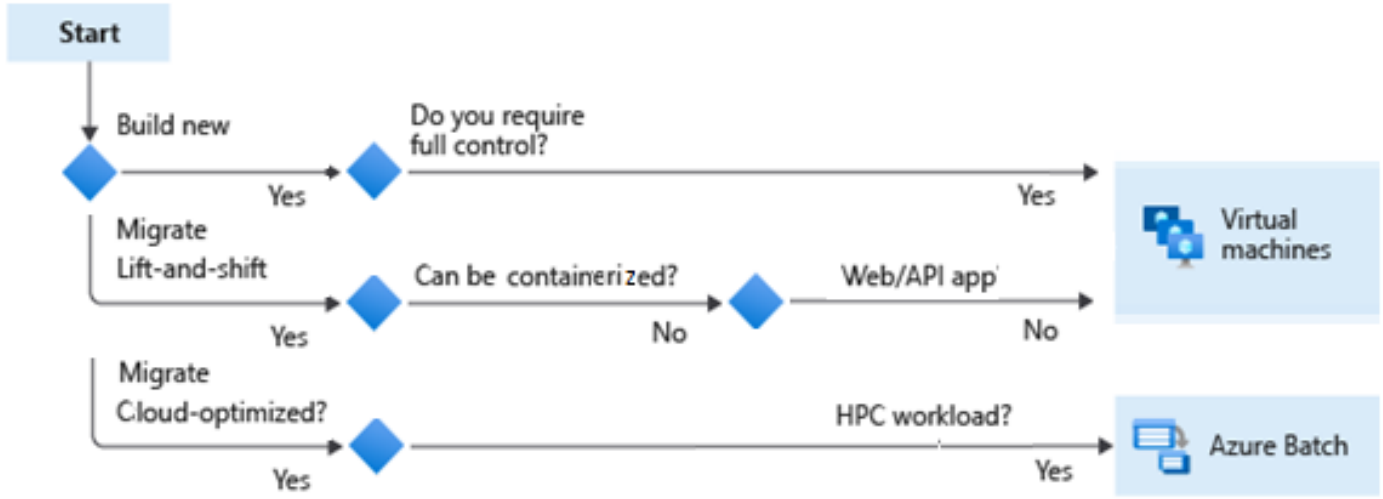
The following table outlines the benefits of scale sets compared to manually managing multiple VM instances.

Design for Azure Batch solutions

Azure Batch is a service that enables you to run large-scale parallel and high-performance computing (HPC) applications efficiently in the cloud. There's no need to manage or configure infrastructure. Just schedule the job, allocate the resources you need, and let Batch take care of the rest

When to use Azure Batch

Azure Batch enables large-scale parallel and HPC batch jobs



•Azure Batch works well with applications that run independently (parallel workloads).
•Azure Batch is effective for applications that need to communicate with each other (tightly coupled workloads). You can use Batch to build a service that runs a Monte Carlo simulation for a financial services company or a service to process images.

- Compute-intensive tasks and dynamically adjust resources for your solution without managing infrastructure
- Create and manage jobs in a pool of compute nodes (virtual machines)
- Azure Batch can also install the application that you want to run, and schedule jobs to run on the compute nodes

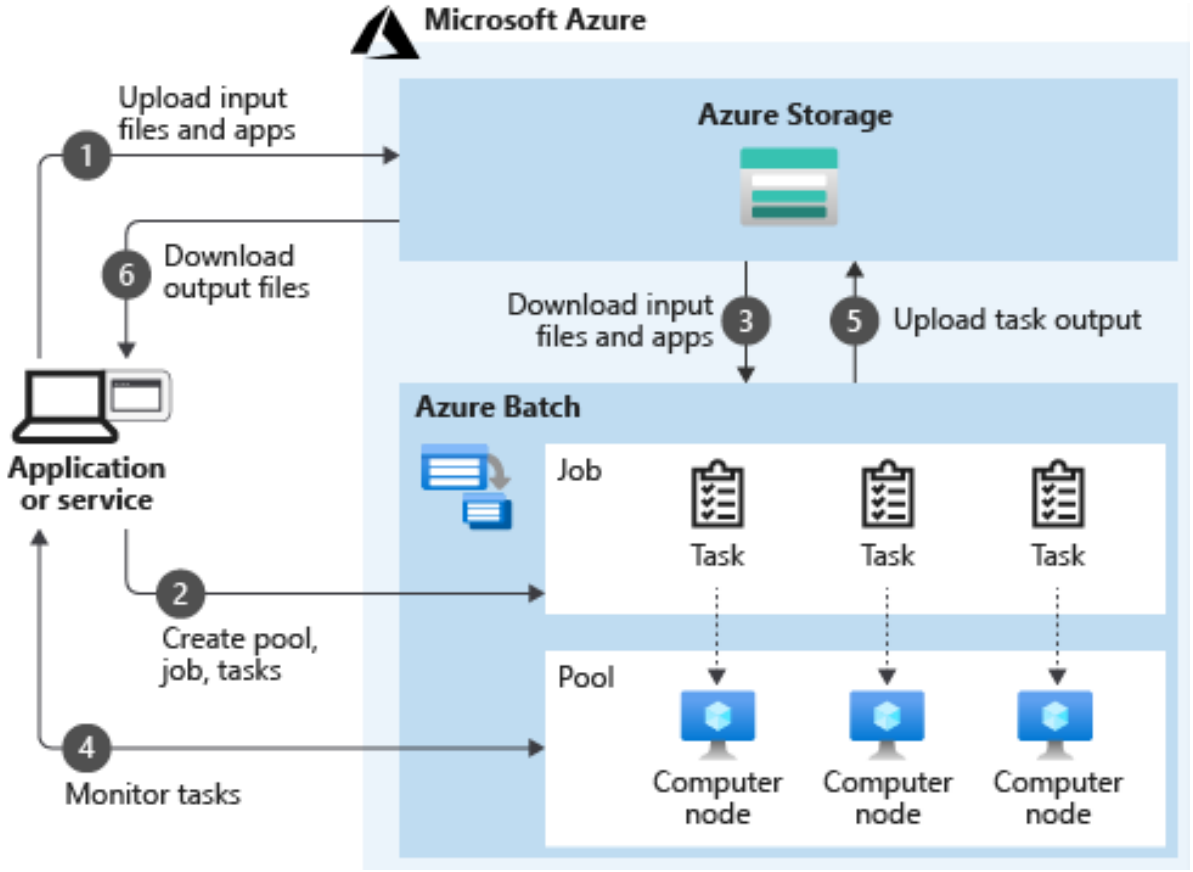
How Azure Batch works

Think of the diagram in two parts:

- **Your service** that uses Azure as the platform
- **Batch as the compute platform behind your service**. Batch uses Azure Storage to fetch applications or data needed to complete a task.

Considerations

- Pools – dynamically allocate jobs, right size the pools
- Nodes – use multiple nodes in the pool, right size the machines
- Jobs – uniquely name the jobs, monitor and log activity



•Azure Batch enables large-scale parallel and high-performance computing (HPC) batch jobs with the ability to scale to tens, hundreds, or thousands of virtual machines. When you're ready to run a job, Azure Batch:

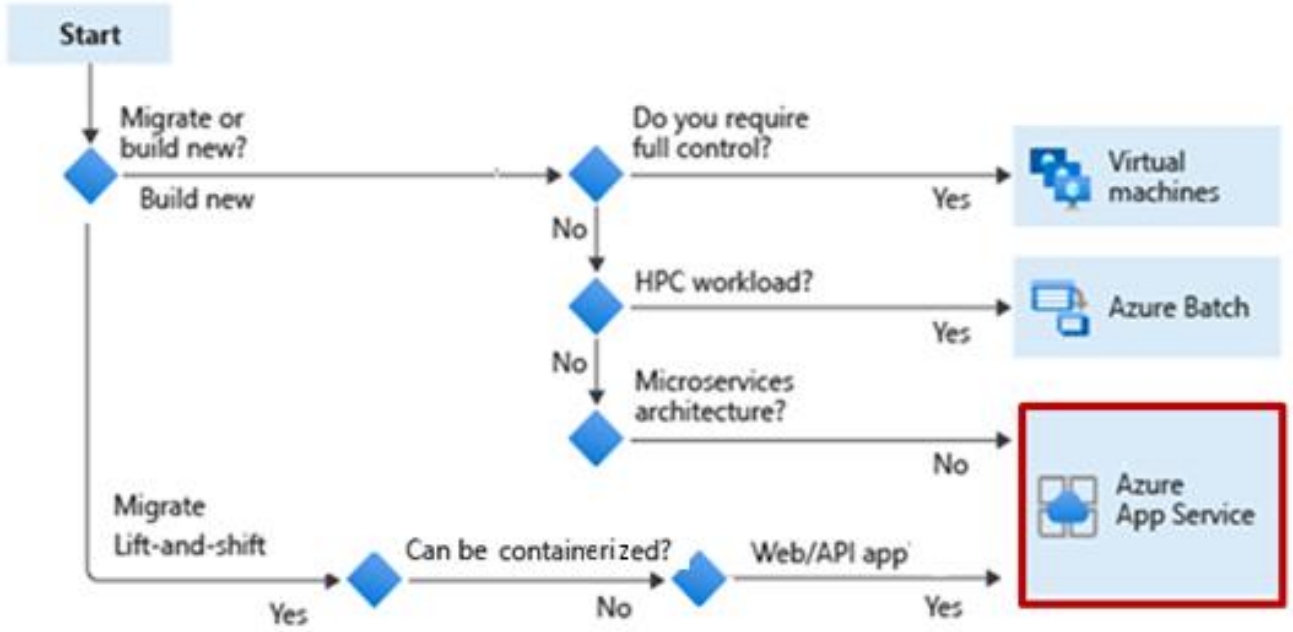
- Starts a pool of compute virtual machines for you.
- Installs applications and staging data.
- Runs jobs with as many tasks as you have.
- Identifies failures, requeues work, and scales down the pool as work completes.

Design for Azure App Services solutions

When to use Azure App Services Web Apps

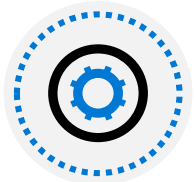
Azure App Service is an HTTP-based service that lets you build and host apps

- Web apps, background jobs, mobile backends, and RESTful APIs.
- Use the programming language of your choice
- Automatic scaling and high availability
- App Service enables automated deployments from GitHub, Azure DevOps, or other source control services



Considerations for App Service Web Apps

Determine the appropriate app service plan – determines cost and scaling



Use App Services deployment slots for continuous deployment

Use the Mobile Apps to build a back end for iOS and Android apps.

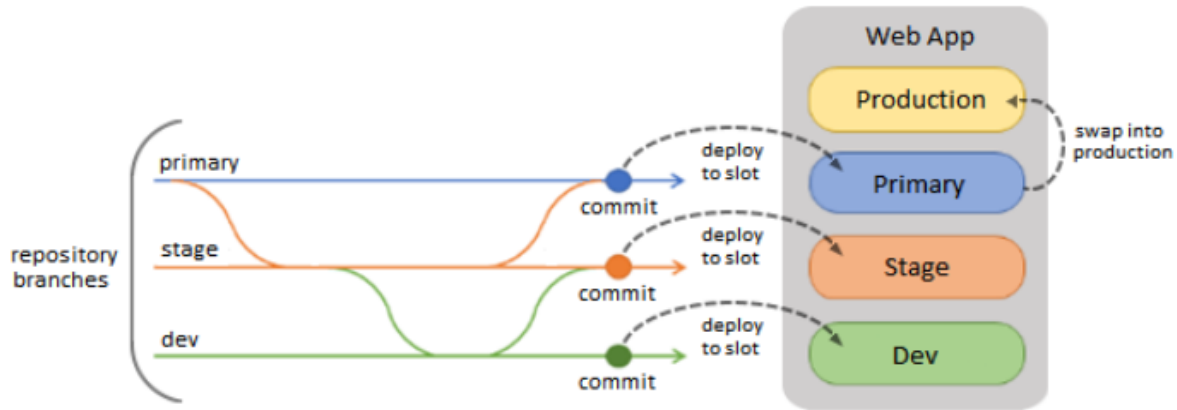
Use built-in authentication and authorization capabilities

Build REST-based web apps with the API

Use WebJobs to run a program or script

Continuous deployment

Azure App Service enables continuous deployment. Azure DevOps provides developer services for support teams to plan work, collaborate on code development, and build and deploy applications. Whenever possible when continuously deploying your code, use deployment slots for a new production build.



When you choose a Standard App Service Plan tier or better, you can deploy your app to a staging environment, validate your changes, and do performance tests. When you're ready, you can swap your staging and production slots. The swap operation triggers the necessary worker instances to match your production scale.

Azure App Service costs

You pay for the Azure compute resources your app uses while it processes requests. The cost is based on the Azure App Service plan you choose. The App Service plan determines how much hardware is devoted to your host. The plan specifies whether you're using dedicated or shared hardware and how much memory is reserved. You can have different app service plans for different apps, and your plan can be scaled up and down at any time.

Things to consider when using Azure App Service

Let's look at some scenarios for using Azure App Service. As you review these options, think about how you can integrate Azure App Service in the Tailwind Traders infrastructure.

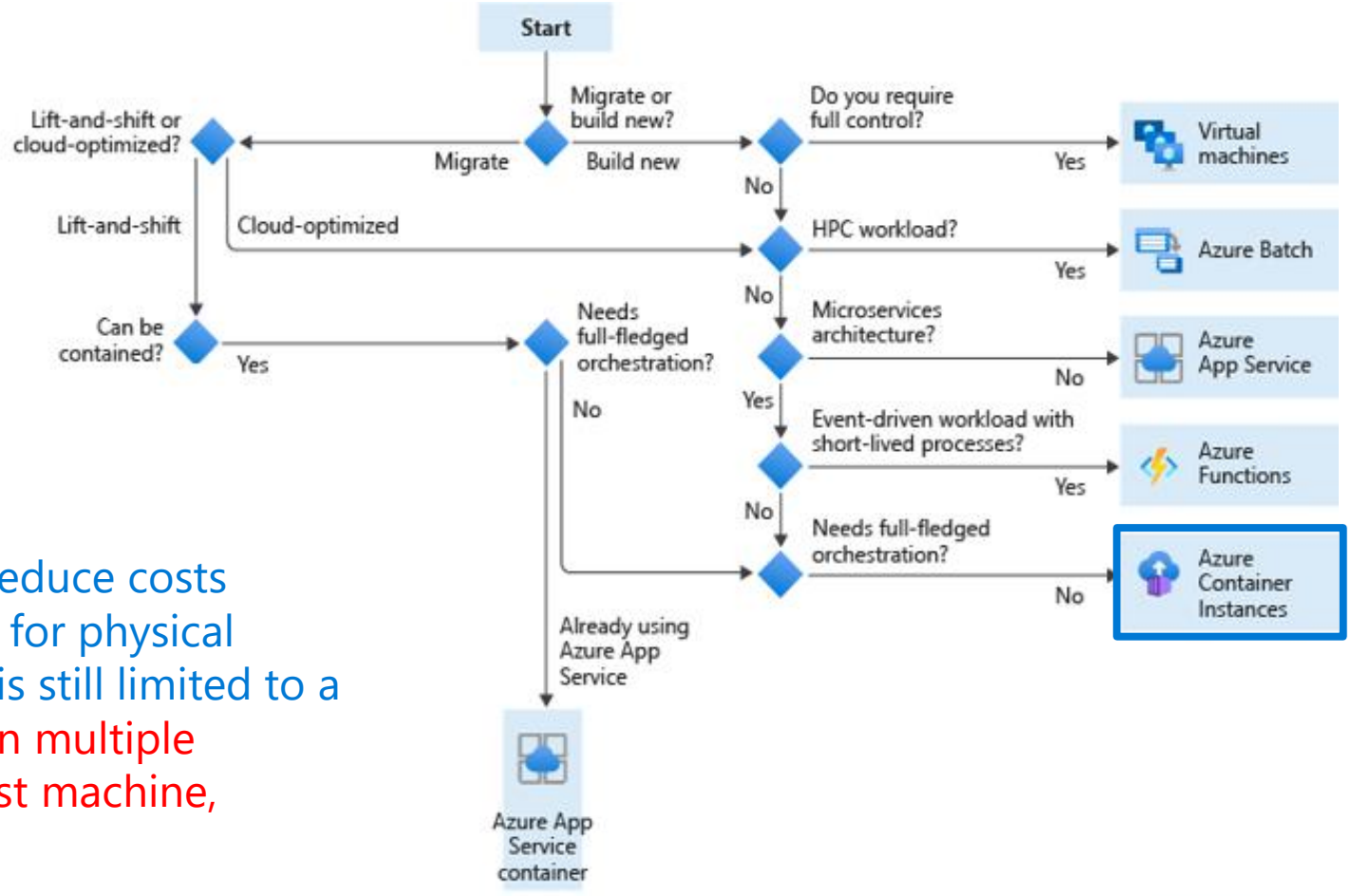
- **Consider web apps.** Create web apps with App Service by using ASP.NET, ASP.NET Core, Java, Ruby, Node.js, PHP, or Python. You can choose either Windows or Linux as the host operating system.
- **Consider API apps.** Build API apps similar to REST-based web APIs with your choice of language and framework. Azure App Service offers full Swagger support, and the ability to package and publish your API in Azure Marketplace. The apps can be consumed from any HTTP or HTTPS client.
- **Consider WebJobs.** Use the App Service WebJobs feature to run a program or script. Program examples include Java, PHP, Python, or Node.js. Script examples include cmd, bat, PowerShell, or Bash. WebJobs can be scheduled or run by a trigger. WebJobs are often used to run background tasks as part of your application logic.
- **Consider Mobile apps.** Exercise the Mobile Apps feature of Azure App Service to quickly build a backend for iOS and Android apps. On the mobile app side, App Service provides SDK support for native iOS and Android, Xamarin, and React native apps. With just a few steps in the Azure portal, you can:
  - Store mobile app data in a cloud-based SQL database.
  - Authenticate customers against common social providers, such as MSA, Google, Twitter, and Facebook.
  - Send push notifications.
  - Execute custom back-end logic in C# or Node.js.
- **Consider continuous deployment.** Choose the Standard App Service Plan tier or better to enable continuous deployment of your code. Deploy your app to a staging slot and validate your app with test runs. When the app is ready for release, swap your staging and production slots. The swap operation warms up the necessary worker instances to match your production scale, which eliminates downtime.
- **Consider authentication and authorization.** Take advantage of the built-in authentication capabilities in Azure App Service. You don't need any language, SDK, security expertise, or even any code to use the functionality in your web app or API. You can integrate with multiple sign-in providers, such as Microsoft Entra ID, Facebook, Google, and Twitter. Azure Functions offers the same built-in authentication features that are available in App Service.
- **Consider multiple plans to reduce costs.** Configure different Azure App Service plans for different apps. Scale your plan up and down at any time. Start testing your web app in a Free App Service plan and pay nothing. When you want to add your custom DNS name to the web app, just scale your plan up to the Shared tier.

Design for Azure Container Instances solutions

What is Azure Container Instance

Azure Container Instances offers a fast and simple way to run a container in Azure, without having to manage any virtual machines and without having to adopt a higher-level service.

- Ensure the integrity of images throughout the lifecycle
- Monitor container resource activity
- Consider container groups



Virtual machines are an excellent way to reduce costs versus the investments that are necessary for physical hardware. However, each virtual machine is still limited to a single operating system. If you want to run multiple instances of an application on a single host machine, containers are an excellent choice.

When to use Azure Container Instances

Azure Container Instances are a fast and simple way to run a container on Azure.

Pros:

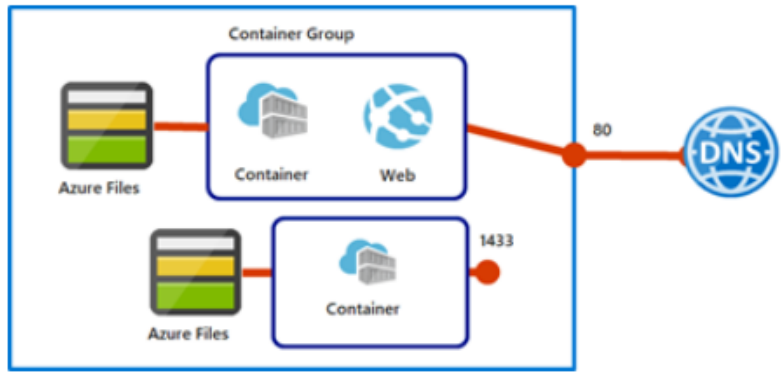
- Fast and Easy
- Used for testing and development.
- Used for short-lived processes.
- Can be used for AKS overflow

Cons:

- Doesn’t scale
- Not designed for microservices

Container groups

The top-level resource in Azure Container Instances is the *container group*. A container group is a collection of containers that get scheduled on the same host machine. The containers in a container group share a lifecycle, resources, local network, and storage volumes.



Multi-container groups are useful when you want to divide a single functional task into several container images. These images can then be delivered by different teams and have separate resource requirements. Some example scenarios include:

- A container serving a web application and a container pulling the latest content from source control.
- An application container and a logging container. The logging container collects the logs and metrics output by the main application and writes them to long-term storage.
- An application container and a monitoring container. The monitoring container periodically makes a request to the application to ensure it's running and responding correctly, and raises alerts as needed.
- A front-end container and a back-end container. The frontend might serve a web application with the backend running a service to retrieve data.

Things to consider when using Azure Container Instances

When you work with Azure Container Instances, there are several recommended security practices.

- **Use a private registry**. Containers are built from images that are stored in one or more repositories. These repositories can belong to a public registry or to a private registry. An example of a private registry is the Docker Trusted Registry, which can be installed on-premises or in a virtual private cloud. Another example is Azure Container Registry that can be used to build, store, and manage container images and artifacts.

- **Ensure image integrity throughout the lifecycle**. Part of managing security throughout the container lifecycle is to ensure the integrity of the container images. Images with vulnerabilities, even minor, shouldn't be allowed to run in a production environment. Keep the number of production images small to ensure they can be managed effectively.

- **Monitor container resource activity**. Monitor your resource activity, like files, network, and other resources that your containers access. Monitoring resource activity and consumption are useful both for performance monitoring and as a security measure.

Compare containers to virtual machines

App Services can also run containers

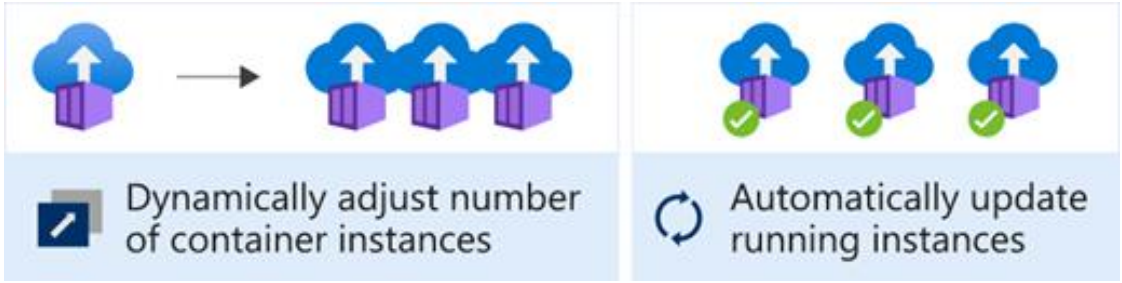
Feature	Containers	Virtual Machines
Isolation	Lightweight isolation from the host and other container.	Complete isolation, strong security boundary, from the host and other VMs.
Operating system	Runs the user mode portion of an operating system	Runs a complete operating system
Deployment	Deploy using Docker	Deploy in the cloud or on-premises – portal, PowerShell, CLI, templates, automation ..
Persistent storage	Azure disks or file share	Virtual hard disk (VHD)or a file share
Fault tolerance	If a cluster node fails, any containers running on it are rapidly recreated by the orchestrator on another cluster node.	VMs can fail over to another server in a cluster, with the VM's operating system restarting on the new server.

Design for Azure Kubernetes Service solutions

Considerations for Azure Kubernetes Services

Azure Kubernetes Service (AKS) orchestrates your containerized applications

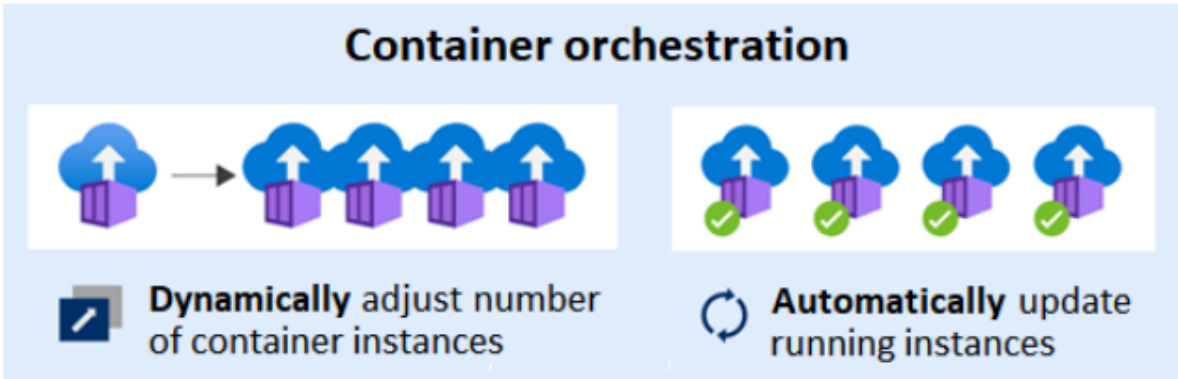
- Enterprise scalability
- Automatic cluster node and pod scaling
- Granular network control
- Cluster node upgrades
- Storage volume support
- Ingress with HTTP application routing support
- Private container registry support



Design for Azure Kubernetes Service solutions

4 minutes

Kubernetes is a portable, extensible open-source platform for automating deployment, scaling, and the management of containerized workloads. This orchestration platform provides the same ease of use and flexibility as with Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) offerings. Kubernetes provides both *container management* and *container orchestration*.

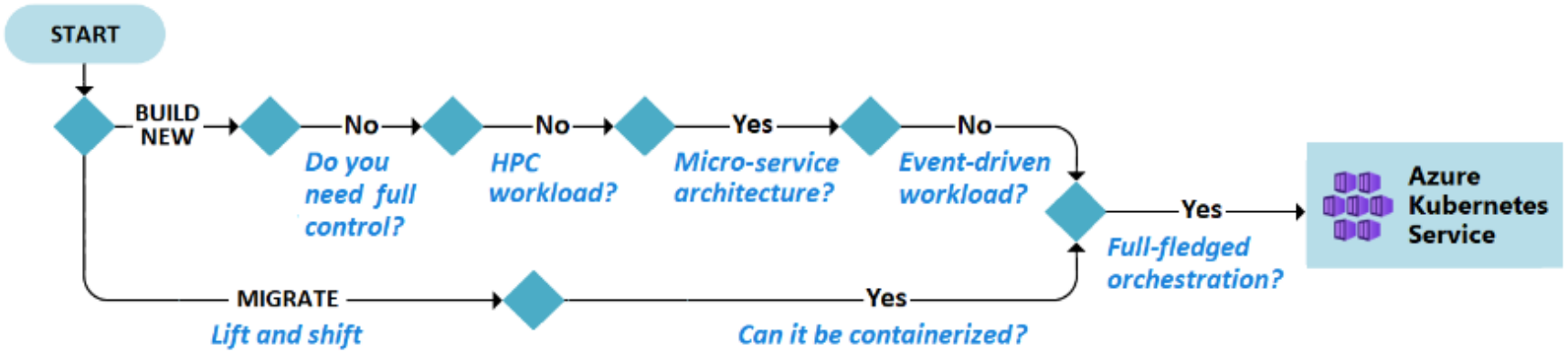


Container management is the process of organizing, adding, removing, or updating a significant number of containers. Most of these tasks are manual and error prone. Container orchestration is a system that automatically deploys and manages containerized applications. The orchestrator can dynamically increase or decrease the deployed instances of the managed application. The orchestrator can also ensure all deployed container instances get updated if a new version of a service is released.

Azure Kubernetes Service (AKS) manages your hosted Kubernetes environment and makes it simple to deploy and manage containerized applications in Azure.

Things to know about Azure Kubernetes Service

The Azure Kubernetes Service environment is enabled with many features, such as automated updates, self-healing, and easy scaling. Review the following characteristics that make AKS an appealing compute option to build new workloads and support lift and shift migrations.



- The Kubernetes cluster is managed by Azure and is free. You manage the agent nodes in the cluster and only pay for the virtual machines on which your nodes run.
- When you create the cluster, you can use Azure Resource Manager (ARM) templates to automate cluster creation. With ARM templates, you specify features like as advanced networking, Microsoft Entra integration, and monitoring.
- AKS gives you the benefits of open-source Kubernetes. You don't have the complexity or operational overhead of running your own custom Kubernetes cluster.

Things to consider when using Azure Kubernetes Service

There are several factors to consider when deciding whether Azure Kubernetes Service is the right compute solution for your infrastructure. A good approach is to plan your strategy from two points of view. Consider the features from the approach of a *green field* new project, and also from the perspective of a *lift-and-shift* migration. The following features are configurable when you create a new cluster and also after you deploy.

Expand table

Feature	Consideration	Solution
Identity and security management	*Do you already use existing Azure resources and make use of Microsoft Entra ID?*	You can configure an Azure Kubernetes Service cluster to integrate with Microsoft Entra ID and reuse existing identities and group membership.
Integrated logging and monitoring	*Are you using Azure Monitor?*	Azure Monitor provides performance visibility of the cluster.
Automatic cluster node and pod scaling	*Do you need to scale up or down a large containerization environment?*	AKS supports two auto cluster scaling options. The *horizontal pod autoscaler* watches the resource demand of pods and increases pods to meet demand. The *cluster autoscaler* component watches for pods that can't be scheduled because of node constraints. It automatically scales cluster nodes to deploy scheduled pods.
Cluster node upgrades	*Do you want to reduce the number of cluster management tasks?*	AKS manages Kubernetes software upgrades and the process of cordoning off nodes and draining them.
Storage volume support	*Does your application require persisted storage?*	AKS supports both static and dynamic storage volumes. Pods can attach and reattach to these storage volumes as they're created or rescheduled on different nodes.
Virtual network support	*Do you need pod-to-pod network communication or access to on-premises networks from your AKS cluster?*	An AKS cluster can be deployed into an existing virtual network with ease.
Ingress with HTTP application routing support	*Do you need to make your deployed applications publicly available?*	The HTTP application routing add-on makes it easy to access AKS cluster deployed applications.
Docker image support	*Do you already use Docker images for your containers?*	By default, AKS supports the Docker file image format.
Private container registry	*Do you need a private container registry?*	AKS integrates with Azure Container Registry (ACR). You aren't limited to ACR though, you can use other container repositories, public, or private.

https://customers.microsoft.com/story/784791-mercedes-benz-r-and-d-creates-container-driven-cars-powered-by-microsoft-azure

Design a highly available container solution

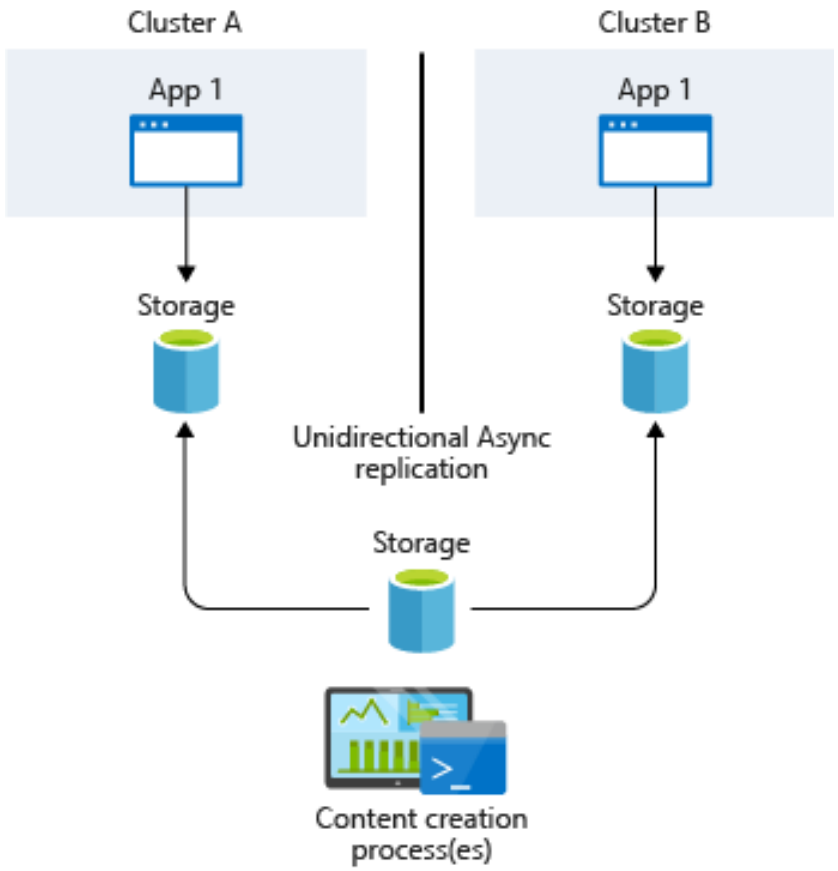
AKS provides high availability using multiple nodes in a virtual machine scale set.

When planning to implement AKS clusters across multiple region deployments, consider the following:

- AKS region availability
- Azure paired regions
- Service availability

Two ways to synchronize storage.

- Infrastructure-based asynchronous replication
- Application-based asynchronous replication



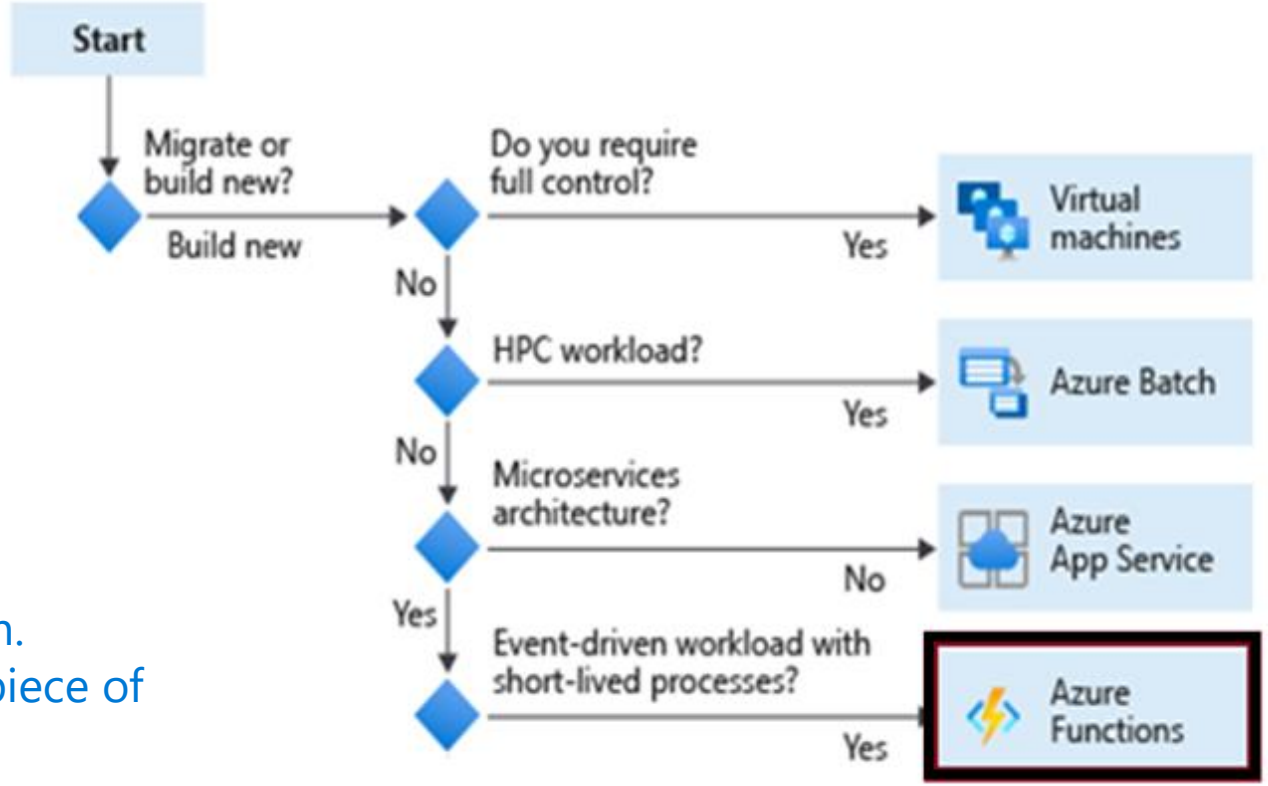
Design for Azure Functions

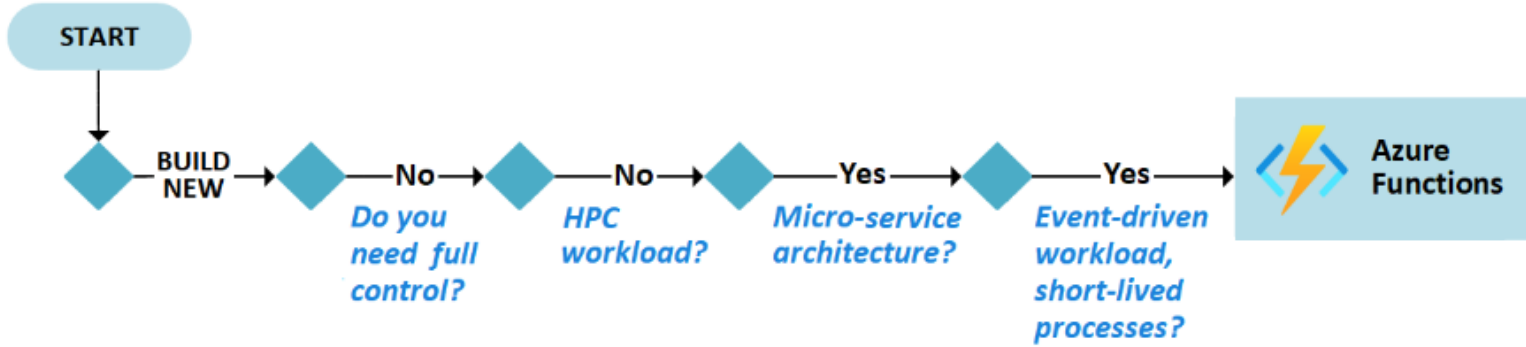
When to use Azure Functions

Azure Functions is a serverless application platform for compute-on-demand

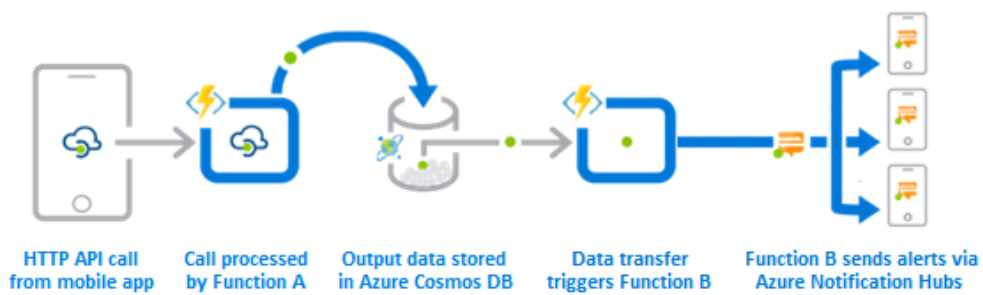
- Implement your system's logic into readily available blocks of code
- Supports a microservice design
- Promotes code reuse
- Scales easily
- Event-driven

Azure Functions is a serverless application platform. Functions are used when you want to run a small piece of code in the cloud, without worrying about the infrastructure.





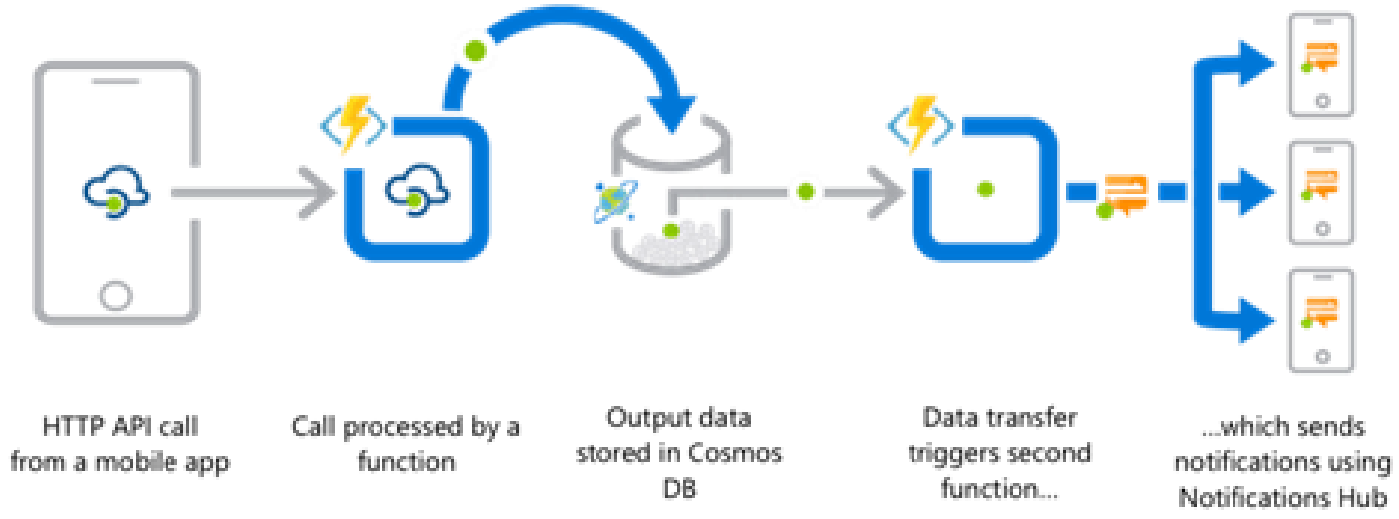
- Azure Functions provides intrinsic scalability. You're charged only for the resources you use.
- With Azure Functions, you can write your function code in the language of your choice.
- Azure Functions supports *compute on demand* in two significant ways:
  - Azure Functions lets you implement your system's logic into readily available blocks of code. These code blocks (functions) can run anytime you need to respond to critical events.
  - As requests increase, Azure Functions meets the demand with as many resources and function instances as necessary. As requests complete, any extra resources and application instances drop off automatically.
- Azure Functions is an ideal solution for handling specific definable actions triggered by an event. A function can process an API call and store the processed data in Azure Cosmos DB. After the data transfer happens, another function can trigger a notification.



Considerations for Azure Functions

Azure Functions are best when handling specific definable actions triggered by an event.

- Avoid long running functions
- Know when to use durable functions
- Organize functions for performance and scaling
- Write defensive functions



Things to consider when using Azure Functions

Let's look at some best practices for using Azure Functions. As you consider these suggestions, think about the advantages to using Azure Functions in the Tailwind Traders infrastructure.

- **Consider long running functions**. Avoid large, long-running functions that can cause unexpected timeout issues. Whenever possible, refactor large functions into smaller function sets that work together and return responses faster. The default timeout is 300 seconds for Consumption Plan functions, and 30 minutes for any other plan.

- **Consider durable functions**. Overcome timeout issues in your configuration with durable functions and smaller function sets. Durable functions let you write stateful functions. Behind the scenes, the function manages the application state, checkpoints, and restarts. An example application pattern for durable functions is *function chaining*. Function chaining executes a sequence of functions in a specific order. The output of one function is applied to the input of another function.

- **Consider performance and scaling**. Plan how to group functions with different load profiles. Consider a scenario where you have two functions. One function processes many thousands of queued messages and has low memory requirements. The other function is called only occasionally but has high memory requirements. In this scenario, you might want to deploy separate function applications, where each function has its own set of resources. Separate resources means you can independently scale the functions.

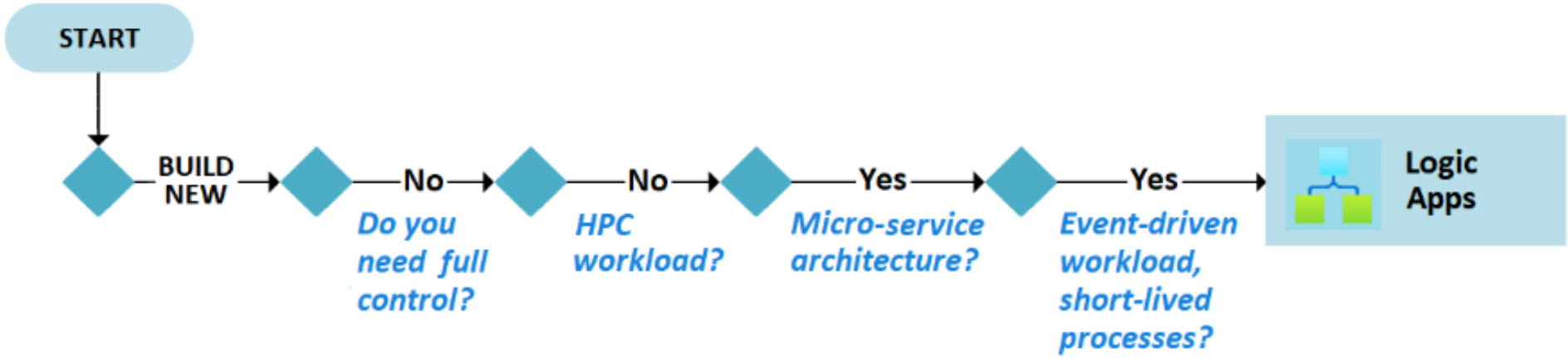
- **Consider defensive functions**. Design your functions to handle exceptions. Downstream services, network outages, or memory limits can cause a function to fail. Write your functions so they can continue if a failure occurs.

- **Consider not sharing storage accounts**. Maximize performance by using a separate storage account for each function application. When you create a function app, associate it with a unique storage account. Using a unique storage account is important if your function generates a high volume of storage transactions.

Design for Azure Logic App solutions

Things to know about Azure Logic Apps

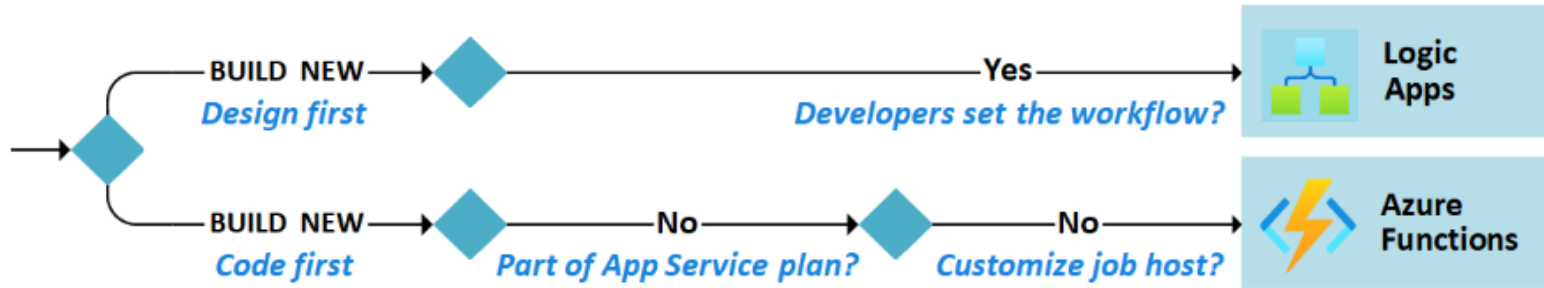
Let's review some characteristics of Azure Logic Apps and scenarios for using the compute solution to build new workloads.



- Azure Logic Apps is a component of Azure Integration Services. Logic Apps simplifies the way you connect legacy, modern, and cutting-edge systems across cloud, on-premises, and hybrid environments.
- With Logic Apps, you can schedule and send email notifications by using Office 365 when a specific event happens, such as a new file uploaded.
- Use Logic Apps to route and process customer orders across on-premises systems and cloud services.
- Implement Logic Apps to move uploaded files from an SFTP or FTP server to Azure Storage.
- Monitor tweets and analyze sentiment with Logic Apps, and create alerts or tasks for items that need review.

Compare Azure Logic Apps and Azure Functions

Azure Logic Apps is similar to Azure Functions as a compute service, but there are basic differences. Azure Functions is a **code-first** technology that uses durable functions. Azure Logic Apps is a **design-first** technology. Review the following flowchart and table to compare the two solutions.



Expand table

Compare	Azure Functions	Azure Logic Apps
Development	Code-first	Design-first
Method	Write code and use the durable functions extension	Create orchestrations with a GUI or by editing configuration files
Connectivity	- Large selection of built-in binding types - Write code for custom bindings	- Large collection of connectors - Enterprise Integration Pack for B2B scenarios - Build custom connectors
Monitoring	Azure Application Insights	Azure portal, Azure Monitor Logs (Log Analytics)

Things to consider when using Azure Logic Apps

There are several points to consider when deciding whether Azure Logic Apps is the ideal compute solution for your infrastructure. Review the following considerations, and think about how Azure Logic Apps can enhance the compute strategy for Tailwind Traders.

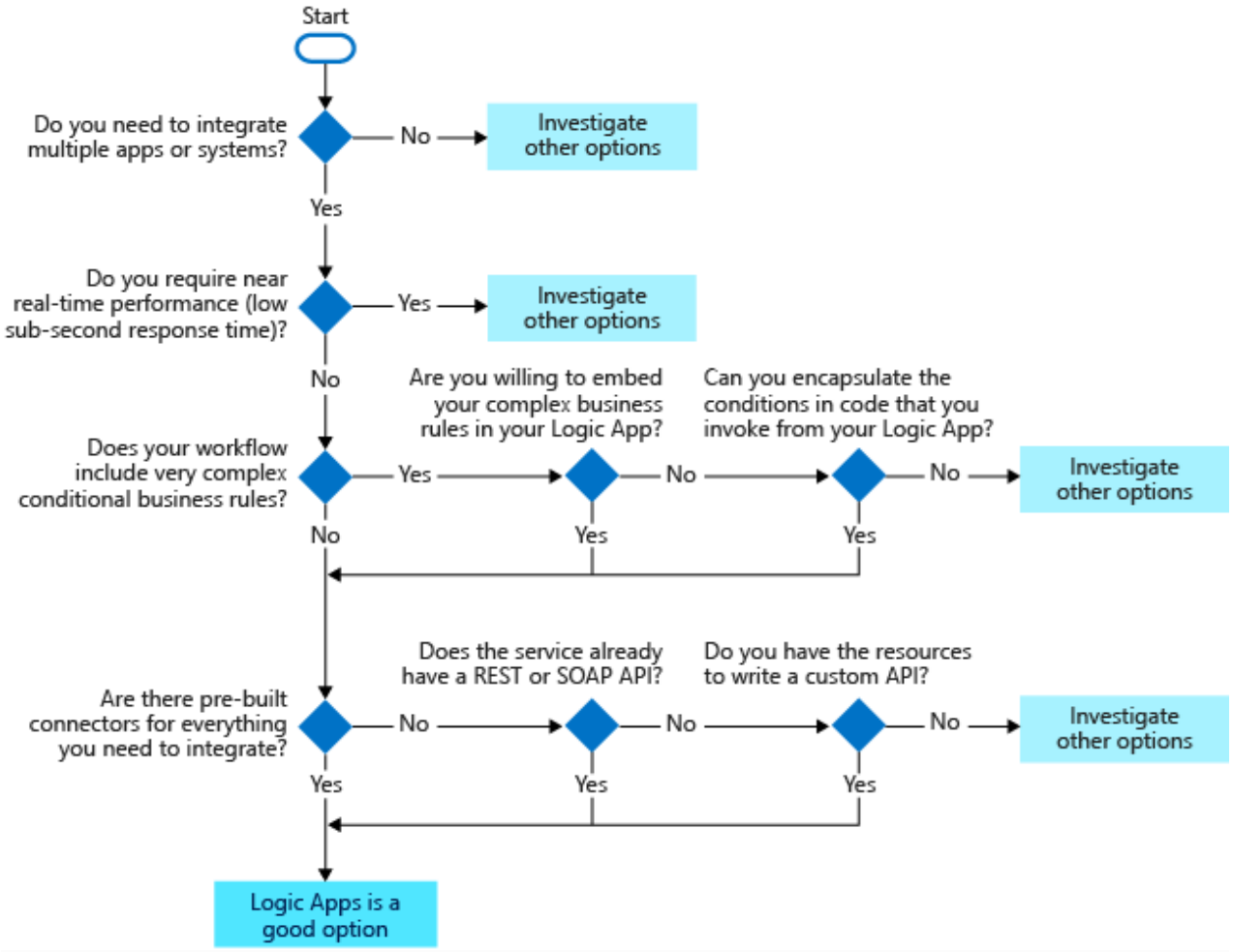
- **Consider integration**. Use Logic Apps to provide the critical infrastructure component of integration with services. Logic Apps is a good option when you need to get multiple applications and systems to work together. If you're building an app with no external connections, Logic Apps is probably not the best option.
- **Consider performance**. Scale your apps automatically with the Logic Apps execution engine. Logic Apps can process large datasets in parallel to let you achieve high throughput. However, fast activation time isn't always guaranteed, nor enforcement of real-time constraints on execution time.
- **Consider conditional expressions**. Build highly complex and deeply nested conditionals into your Logic Apps. Logic Apps provides control constructs like Boolean expressions, switch statements, and loops so your apps can make decisions based on your data.
- **Consider connectors**. Investigate whether pre-built connectors are available for all the services you need to access. You might need to create custom connectors. If a service has an existing REST or SOAP API, you can make the custom connector in a few hours without writing any code. Otherwise, you need to create the API first before making the connector.
- **Consider mixing compute solutions**. Take advantage of diverse features by mixing and matching services when you build an orchestration. You can call functions from Logic Apps, and call logic apps from an Azure function. Build each orchestration based on the service capabilities or your personal preference.
- **Consider other options**. Know when *not* to use Azure Logic Apps. There are cases where Logic Apps might not be the best option. Logic Apps isn't an ideal solution for real-time requirements, complex business rules, or if you're using non-standard services.

When to use Azure Logic Apps

https://customers.microsoft.com/en-us/story/1567549472346870706-eon-esmartsystems-azure-en how Azure Logic Apps distributes data from drones inspecting power lines.

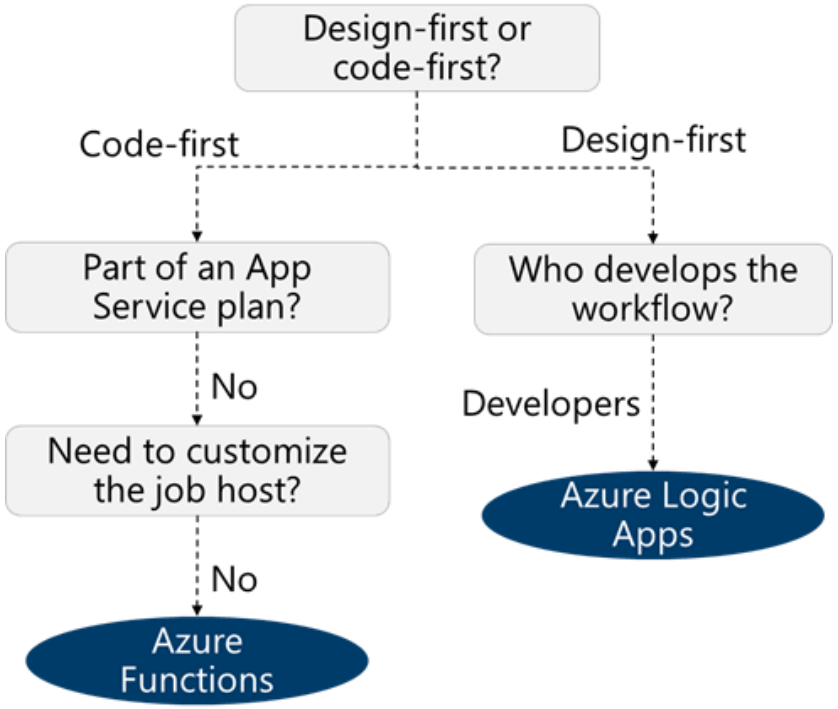
Azure Logic Apps is a cloud-based platform creating and running workflows.

- Send email notifications using Office 365 when a specific event happens
- Route and process customer orders across on-premises systems and cloud services.
- Move uploaded files from an FTP server to Azure Storage.
- Monitor tweets, analyze the sentiment, and create alerts or tasks for items that need review.



How are Logic Apps different from Functions?

Code-first vs designer-first



Comparison	Logic Apps	Durable Functions
Development	Designer-first	Code-first
Method	Create orchestrations by using a GUI or editing configuration files	Write code and use the durable functions extension
Connectivity	Large collection of connectors , B2B pack, custom connectors	Large selection of built-in binding types, write code for custom bindings
Monitoring	Azure portal, Azure Monitor logs	Azure Application Insights

Case studies and review

Review - Product and usage case (matching)

- Virtual machines
- App Service
- Azure Container Instances
- Azure Kubernetes Service
- Azure Functions
- Azure Logic Apps

Usage case?

Move your independent docker containers to a PaaS offering

Schedule tasks and messages

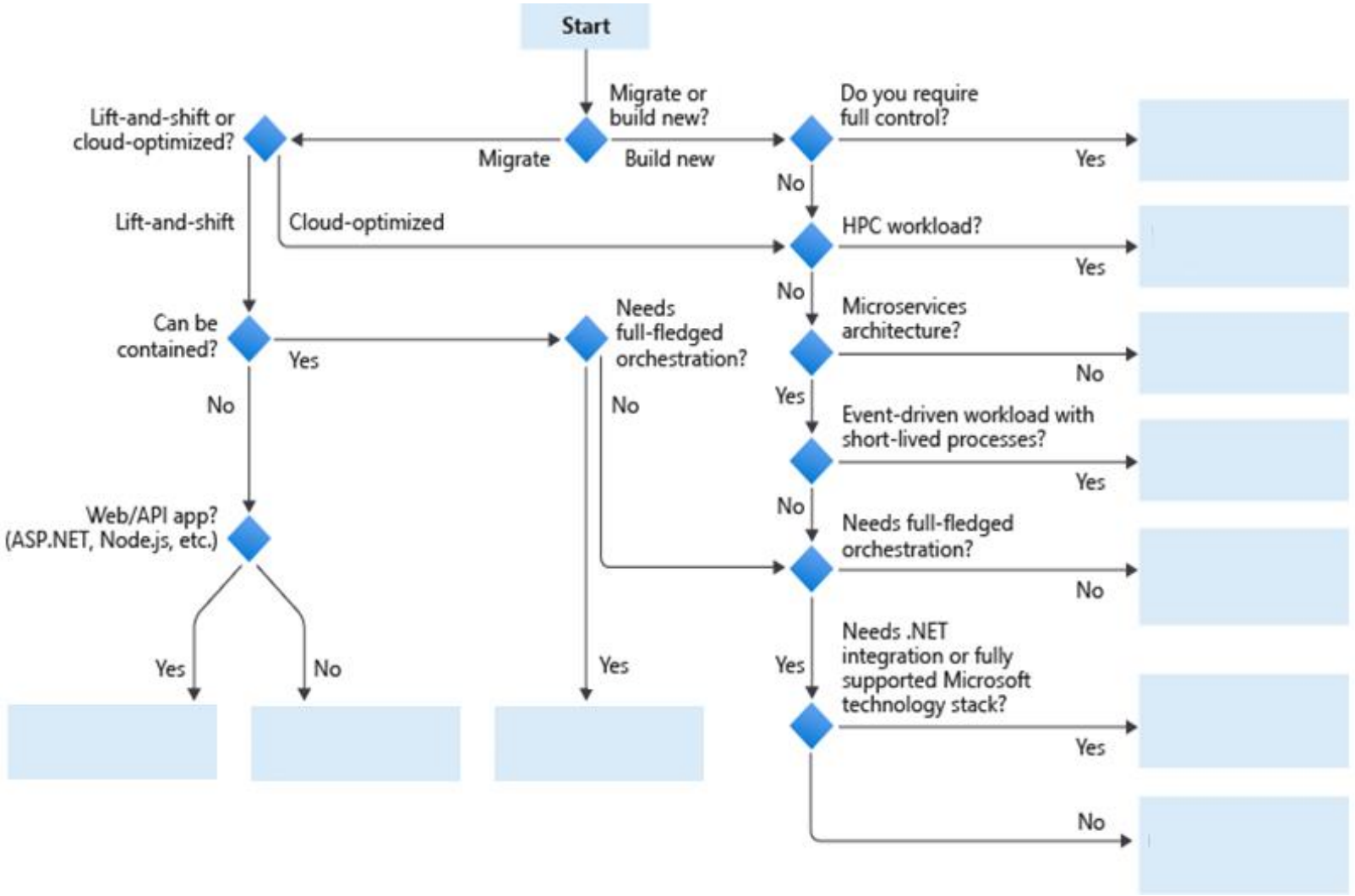
Run mobile apps

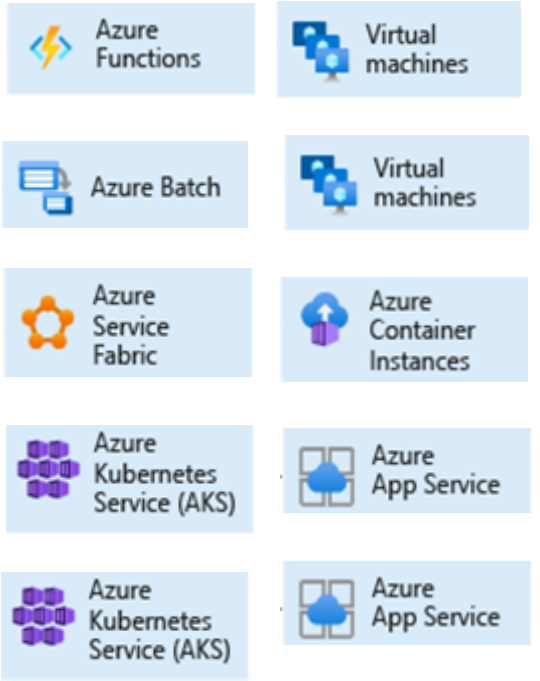
Manage container orchestration

Provide full operating system access

Trigger a workflow based on an event

Review – Choose a compute service (drag and drop)





Case study – Compute solutions

- Which Azure compute service would you recommend for the front-end tier? Address both the workload hosting and the web application.
- Which Azure compute service would you recommend for the middle tier application? Diagram what you decide and explain your solution.