Cloud

Caltech | Center for Technology & Management Education

Designing Infrastructure Solutions on Azure

Powerd by simplilearn

Cloud

Design a Compute Solution

Powerd by simplilearn

Caltech Center for Technology & Management Education

Learning Objectives

By the end of this lesson, you will be able to:

- Recommend a solution for Compute Provisioning
- Recommend a solution for App Service
- Determine Appropriate Compute Technologies
- Recommend a solution for Containers
- Recommend a solution for Automating Compute Management

A Day in the Life of an Azure Architect

You are working as an architect in an organization, and you have been asked to design a solution to deploy an application for testing and production. The application should be deployed to different environments and should run without installation dependencies.

Your company is also looking for a solution that can provide the quickest and most straightforward approach to host a container in Azure.

To achieve all of the above, along with some additional features, we would be learning a few concepts in this lesson that will help you find a solution for the above scenario.

Recommend a Solution for Compute Provisioning

Azure Compute Service

Compute refers to the hosting model for the computing resources that the application runs on.

The application may need multiple workloads, and each workload must be evaluated separately.

As a result, it is possible that a complete solution may incorporate two or more compute services.

Types of Compute Strategies

There are two types of compute strategies:

Lift and shift

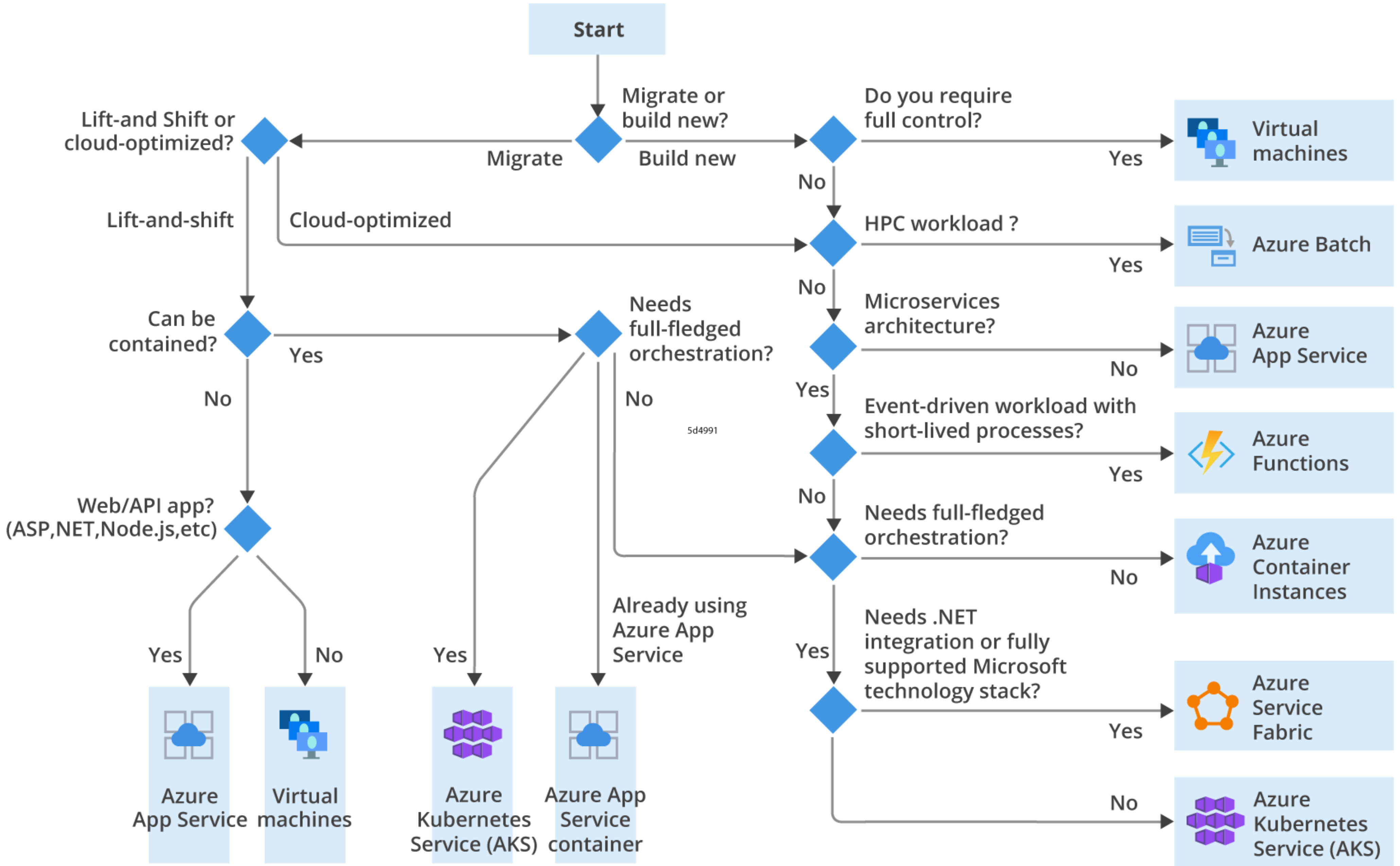
Migrating a service without redesigning the application or making code changes

Cloud optimized

Taking advantage of cloud-native features and capabilities

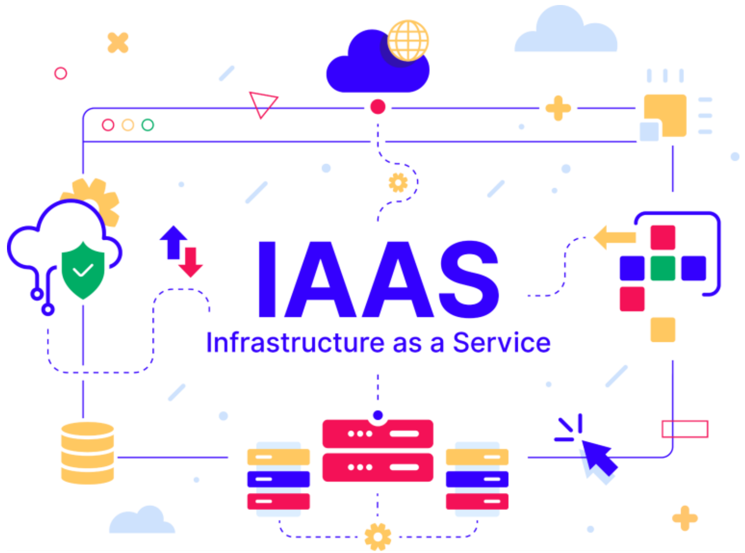
Choose a Candidate Service

This diagram shows how to choose a candidate service:


Start
Migrate or build new?
Migrate
Build new
Lift-and Shift or cloud-optimized?
Lift-and-shift
Cloud-optimized
Can be contained?
Yes
No
Web/API app? (ASP,NET,Node.js,etc)
Yes
No
Azure App Service
Virtual machines
Needs full-fledged orchestration?
Yes
No
Already using Azure App Service
Azure Kubernetes Service (AKS)
Azure App Service container
5d4991
Do you require full control?
Yes
Virtual machines
No
HPC workload ?
Yes
Azure Batch
No
Microservices architecture?
No
Azure App Service
Yes
Event-driven workload with short-lived processes?
Yes
Azure Functions
No
Needs full-fledged orchestration?
No
Azure Container Instances
Yes
Needs .NET integration or fully supported Microsoft technology stack?
Yes
Azure Service Fabric
No
Azure Kubernetes Service (AKS)

Hosting Models

There are two most common hosting models:



- It lets the users provision individual VMs along with the associated networking and storage components.
- It gives the most control, flexibility, and portability.

- It is a complete development and deployment environment in the cloud, with resources.
- It enables the delivery of everything from simple cloud-based apps to sophisticated, cloud-enabled enterprise applications.

Batch

It is an Azure compute management platform that lets users run large-scale parallel batch applications in the cloud.

End-users may now provision a variety of scalable, high-performance resources with ease and at a low cost with the help of Azure Batch.

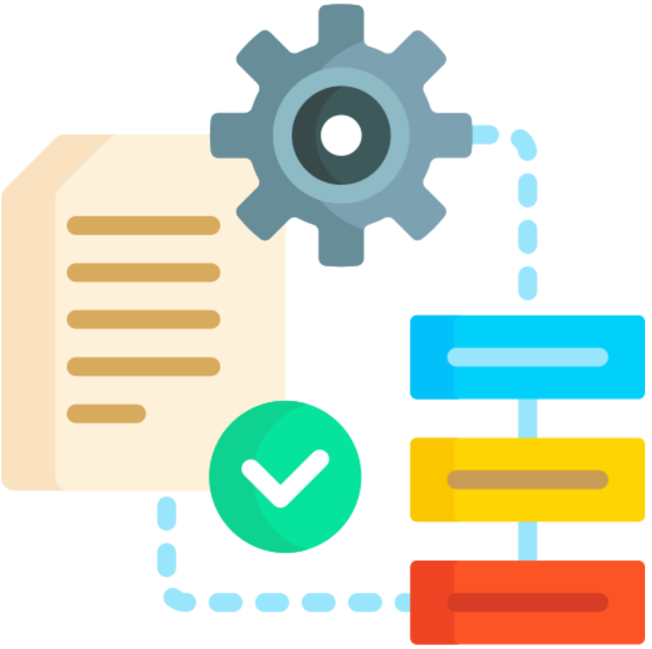
Features of Batch

- It allows users to conduct large-scale parallel workloads at a reasonable cost by utilizing low-priority virtual machines.
- It allows users to fully configure the nodes themselves, given it supports Docker configuration.
- It allows users to create pools and run jobs through an easy-to-use code interface with R.

Applications of Batch

Load testing software application

Software testers can use Azure Batch service to arrange the provisioning of thousands of virtual machine nodes to test their apps for peak load and consumption.



Applications of Batch

Executing high-compute apps and services

The Azure Batch function scheduling service automates the addition of compute clusters for applications and services that require a lot of processing power.

Applications of Batch

Provide on-demand processing

Organizations can use Azure Batch service to provide on-demand and high-end processing capabilities and infrastructure for apps and services.

Candidate Services

These are the different candidate services:

App Service	A managed service for hosting web apps, mobile app back ends, RESTful APIs, or automated business processes
Functions	A managed PaaS service
Batch	A managed Kubernetes service for running containerized applications
Azure Kubernetes Service	A managed service for running large-scale parallel and high-performance computing (HPC) applications

Candidate Services

These are the different candidate services:

Container Instances

The fastest and simplest way to run a container in Azure, without having to provision any virtual machines or adopt a higher-level service

Virtual machines

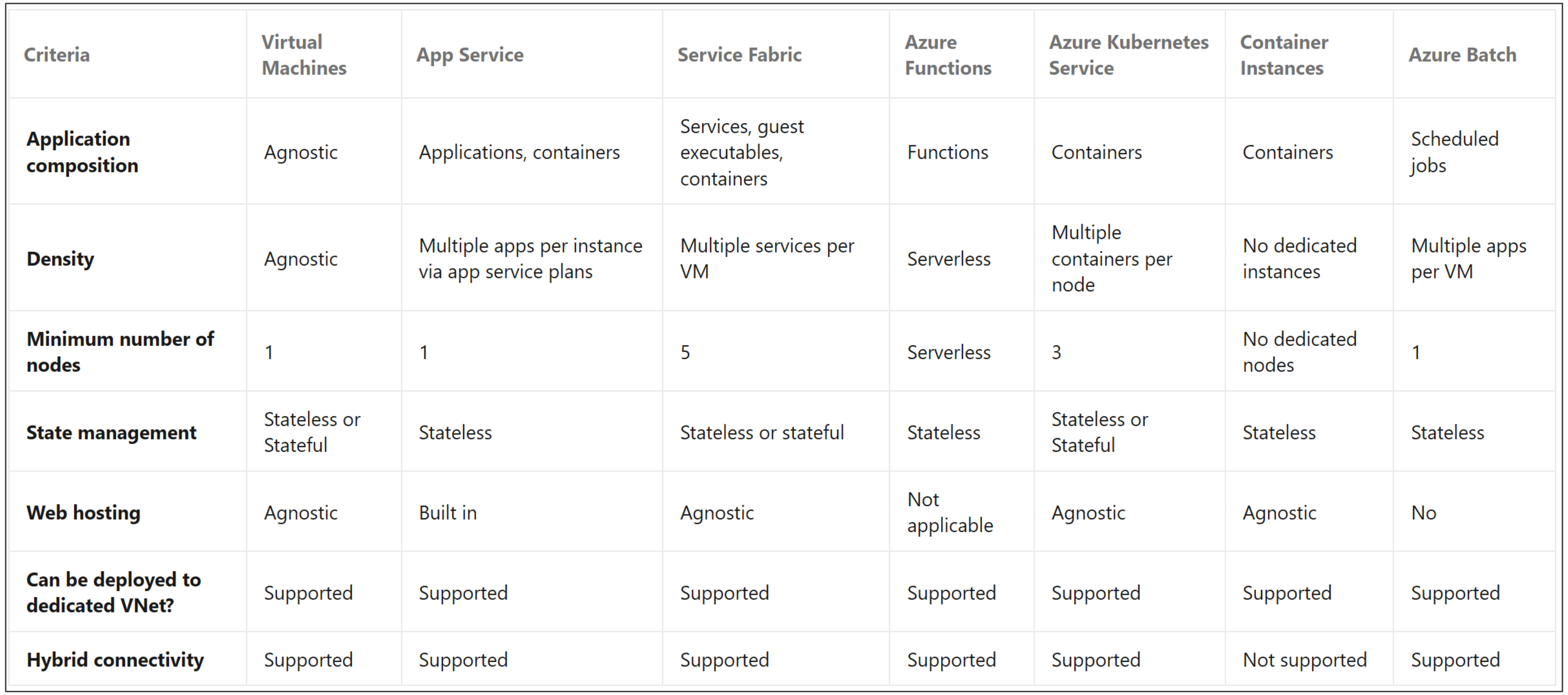
Deployment and management of VMs inside an Azure virtual network

Azure Service Fabric

Packaging, deployment, and management of scalable and reliable microservices and containers

Service Limits and Cost

Every service has service limits applicable.



Criteria	Virtual Machines	App Service	Service Fabric	Azure Functions	Azure Kubernetes Service	Container Instances	Azure Batch
Application composition	Agnostic	Applications, containers	Services, guest executables, containers	Functions	Containers	Containers	Scheduled jobs
Density	Agnostic	Multiple apps per instance via app service plans	Multiple services per VM	Serverless	Multiple containers per node	No dedicated instances	Multiple apps per VM
Minimum number of nodes	1	1	5	Serverless	3	No dedicated nodes	1
State management	Stateless or Stateful	Stateless	Stateless or stateful	Stateless	Stateless or Stateful	Stateless	Stateless
Web hosting	Agnostic	Built in	Agnostic	Not applicable	Agnostic	Agnostic	No
Can be deployed to dedicated VNet?	Supported	Supported	Supported	Supported	Supported	Supported	Supported
Hybrid connectivity	Supported	Supported	Supported	Supported	Supported	Not supported	Supported

It is important to consider the cost, SLA, and regional availability of the service.

Source: https://docs.microsoft.com/

Scalability of Compute Services

Guidance on service to be used for scaling different resources:

Criteria	Virtual Machines	App Service	Service Fabric	Azure Functions	Azure Kubernetes Service	Container Instances	Azure Batch
Autoscaling	Virtual machine scale sets	Built-in service	Virtual machine scale sets	Built-in service	Pod auto-scaling[1], cluster auto-scaling[2]	Not supported	N/A
Load balancer	Azure Load Balancer	Integrated	Azure Load Balancer	Integrated	Azure Load Balancer or Application Gateway	No built-in support	Azure Load Balancer
Scale limit	Platform image: 1000 nodes per scale set; Custom image: 600 nodes per scale set	30 instances, 100 with App Service Environment	100 nodes per scale set	200 instances per Function app	100 nodes per cluster (default limit)	20 container groups per subscription (default limit)	20 core limit (default limit)

Availability of Compute Services

Guidance on service to be used for availing different resources:

Criteria	Virtual Machines	App Service	Service Fabric	Azure Functions	Azure Kubernetes Service	Container Instances	Azure Batch
Multi-region failover	Traffic manager	Traffic manager	Traffic manager, Multi-Region Cluster	Azure Front Door	Traffic manager	Not supported	Not Supported

Assisted Practice

Creating a Virtual Machine

Duration: 10 Min.

Problem Statement:

As an Azure architect, you've been tasked with recommending an Azure computing solution that provides an isolated environment for operating the OS and apps without relying on the underlying host system. Your company will use this solution to deploy an application for testing and production.

Assisted Practice: Guidelines

Steps to create a virtual machine are:

1. Log in to your Azure portal
2. Create a virtual machine on the Azure portal
3. Select Virtual machines
4. Create a new resource group

Unassisted Practice

Manage VM Sizes

Duration: 10 Min.

Problem Statement:

As an Azure Architect, you've been tasked with demonstrating the various VM sizes available in Azure for running your apps and workloads.

Unassisted Practice: Guidelines

Steps to manage VM size are:

1. Log in to your Azure portal
2. Search and select virtual machine
3. Manage VM size

Unassisted Practice

Virtual Machine Extension: PowerShell DSC

Duration: 10 Min.

Problem Statement:

As an Azure architect, you've been requested to recommend a solution for automating and configuring Azure VMs once they've been deployed. For example, we will need the tool if a virtual machine requires software installation, anti-virus protection, or the execution of a script.

Unassisted Practice: Guidelines

Steps to create virtual machine extension: PowerShell DSC are:

1. Log in to your Azure portal
2. Search and select virtual machines
3. Click on Extensions
4. Configure machine extension for a VM

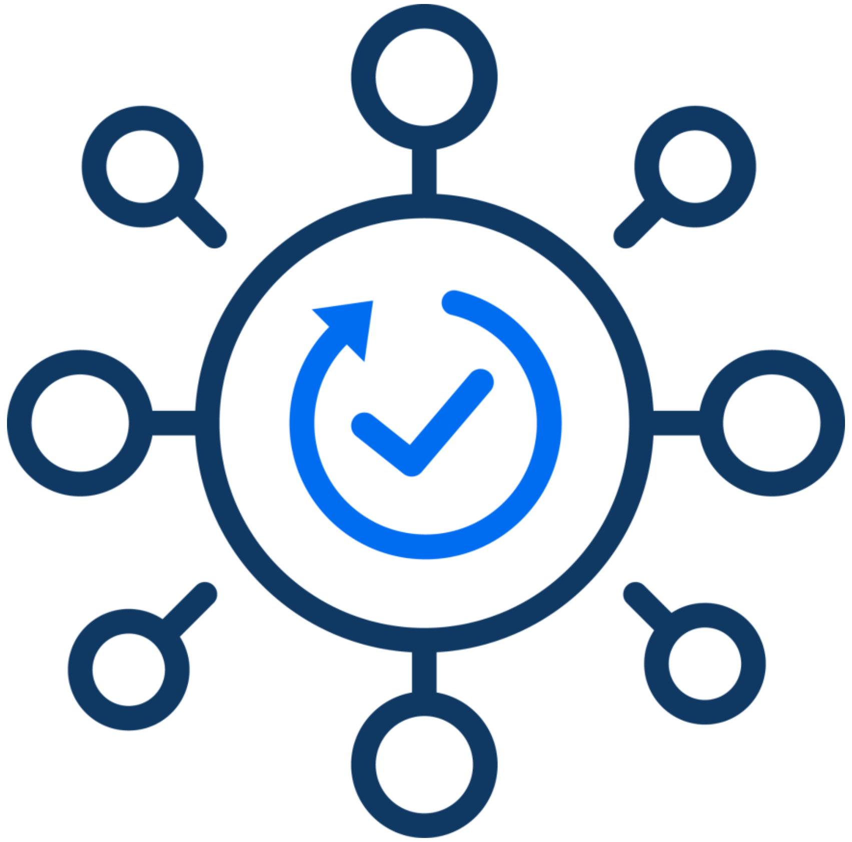
Recommend a Solution for App service

Azure App Service

One-of-a-kind cloud service that allows a user to rapidly build enterprise-ready web and mobile apps for any platform or device and then deploy them on a scalable and reliable cloud infrastructure

Azure App Service High Availability

The Azure App Service is a Platform as a Service offering (PaaS) from Microsoft that enables a user to build, deploy, and scale web applications.



Azure App Service Plan

An App Service plan defines a set of compute resources for web apps.

The App Service plan determines the capabilities and resources available to its web apps.

Azure App Service Plan

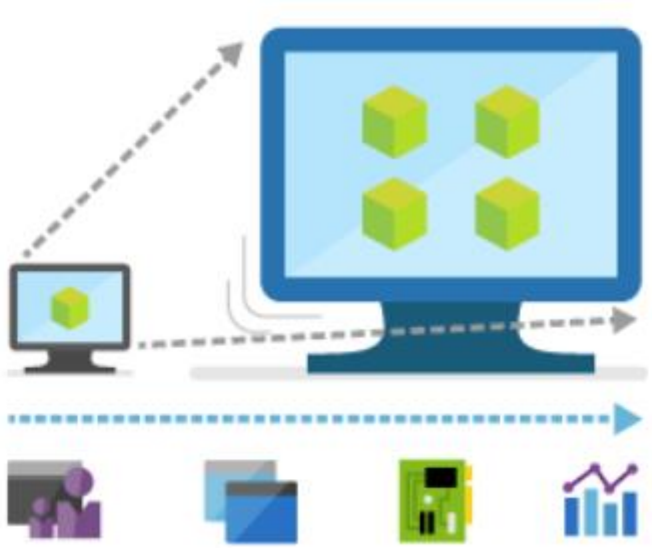
Web apps can share the same App Service plan.

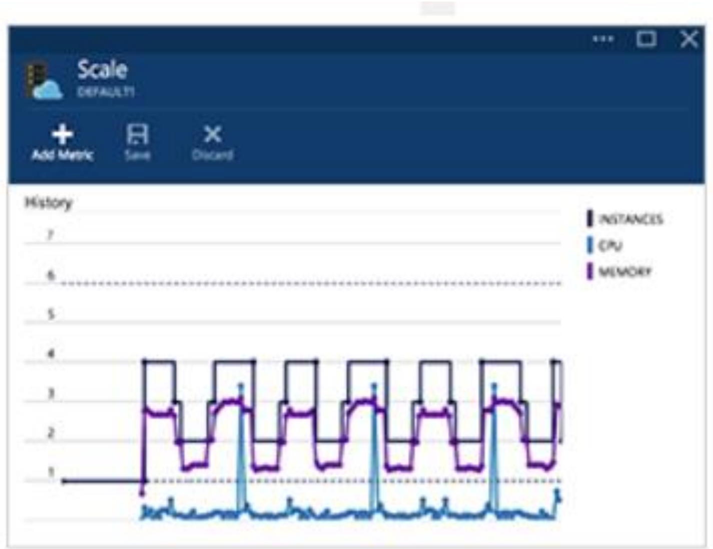
Isolate the app into a new App Service plan when:

- The app is resource-intensive
- A user wants to scale the app independently from other apps
- The app needs resources in a different geographical region

Azure App Service Plans Scalability

The adaptability of the system to the modified amount of workload or traffic





Increasing or decreasing the resources for a user's application

Scale In or Out

The autoscale feature allows the user to scale in and out the number of instances of their app service plan based on the schedules they set or the metrics they define.

In Basic Tier, users can scale up to 3 instances.

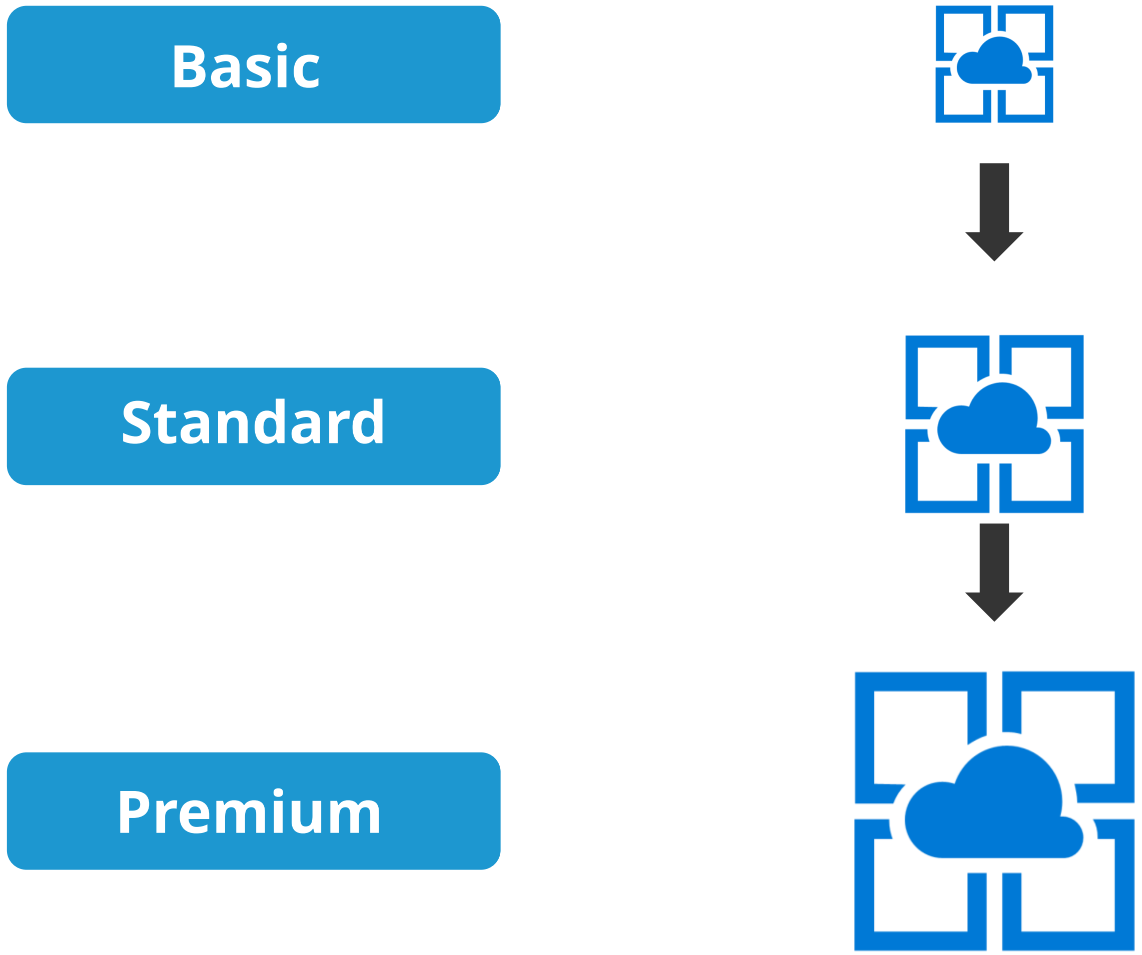
In Isolated (App Service Environment), there can be 100 instances.

In Premium Tier, the maximum is 20 instances.

In Standard Tier, users can scale up to 10 instances.

Scale Up or Down

The following are the different types of tier for scaling up or down:



Basic

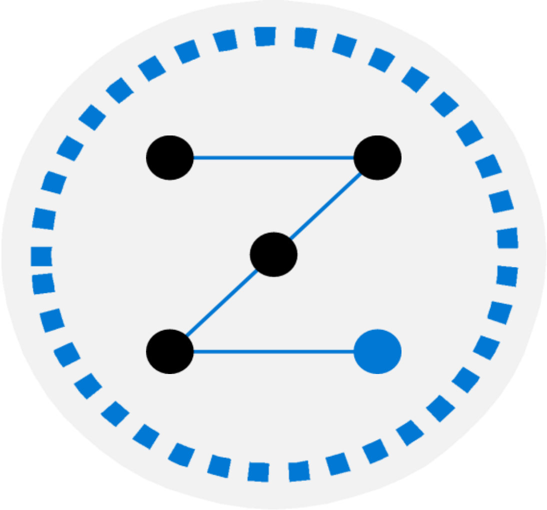
Standard

Premium

Determine Appropriate Compute Technologies

Choosing an Azure Compute Option for Microservices

Below are the popular approaches for a Microservices architecture:



A service orchestrator that manages services running on dedicated nodes (VMs)

A serverless architecture using functions as a service (PaaS)

Service Orchestrators

An orchestrator handles tasks related to deploying and managing a set of services.

Service orchestration

Tasks include:

- Placing services on nodes
- Monitoring the health of services
- Restarting unhealthy services
- Load balancing network traffic across service instances
- Service discovery
- Scaling the number of instances of a service
- Applying configuration updates

Containers with Orchestration

Consider the orchestration services on the Azure Platform:

Azure Kubernetes Service (AKS):

AKS provisions Kubernetes and exposes the Kubernetes API endpoints.

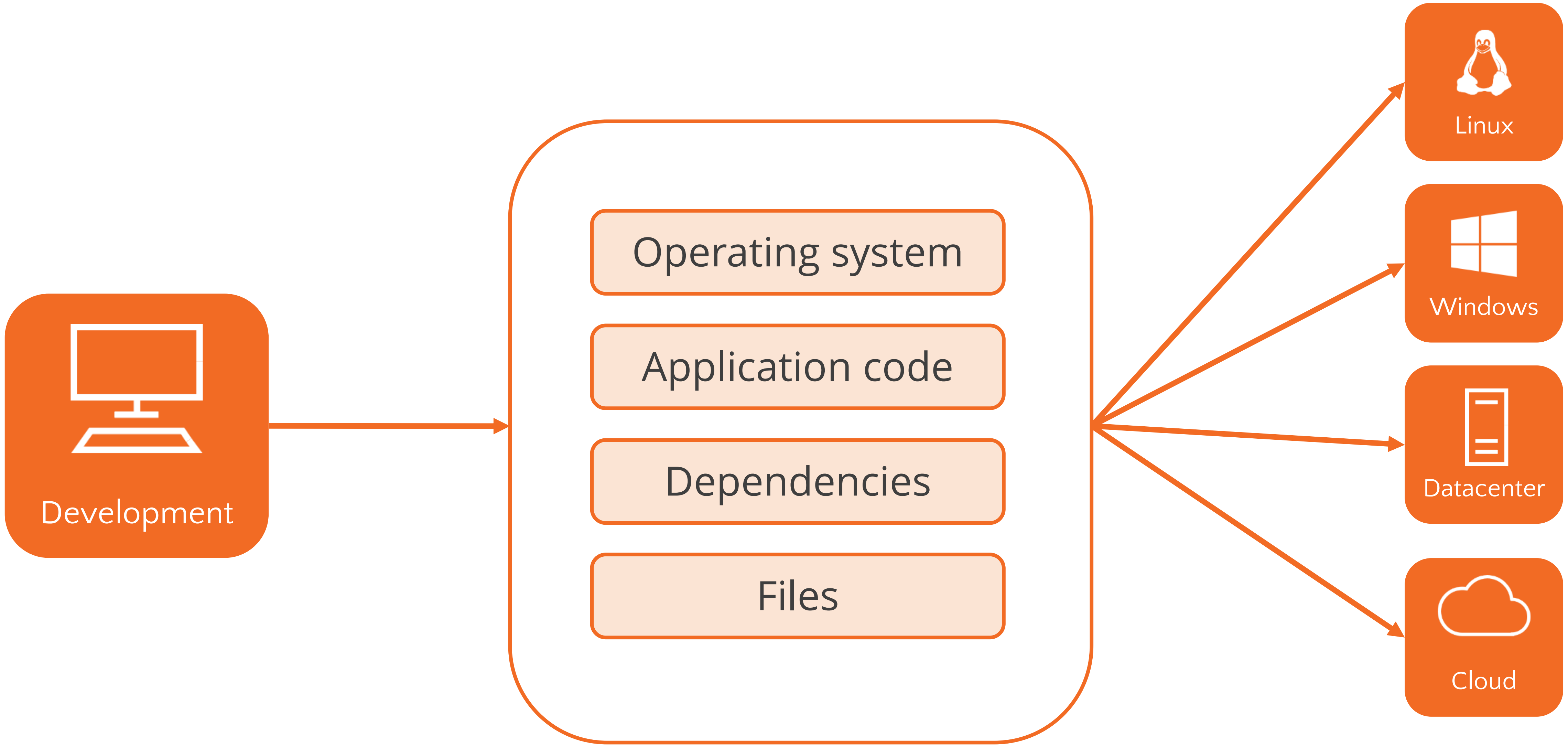
Service Fabric

It is a distributed systems platform for packaging, deploying, and managing microservices.

Note: Docker Enterprise Edition and Mesosphere DC/OS can run in an IaaS environment on Azure.

Containers

A container is a loosely isolated environment that allows us to build and run software packages.



Containers

Containers offer benefits that are specifically relevant to microservices:

Portability

A container image is a standalone package that runs without needing to install libraries or other dependencies.

Agility

Containers can be started and stopped quickly. They spin up new instances to handle more load or to recover from node failures.

Density

Packing multiple containers onto a single node is useful when the application consists of many small services.

Resource isolation

The amount of memory and CPU that is available must be limited to a container to ensure that a runaway process does not exhaust the host resources.

Serverless

With a serverless architecture:

- Users do not manage the VMs or the virtual network infrastructure.
- They deploy code, and the hosting service handles the putting and executing of that code onto a VM.

This approach suits the small granular functions that are coordinated using event-based triggers.

Serverless

Azure Functions is a serverless compute service that supports various function triggers, including:

- HTTP requests
- Service Bus queues
- Event Hubs events

Example: A message being placed onto a queue might trigger a function that reads from the queue and processes the message.

Orchestrator vs. Serverless

These are the differences between orchestrator and serverless:

- Manageability
- Flexibility and control
- Portability
- Application integration
- Cost
- Scalability

- Serverless applications are easy to manage as the platform manages all the compute resources.
- However, with an orchestrator, a user must keep an eye on load balancing, CPU, memory usage, and networking.

Orchestrator vs. Serverless

These are the difference between orchestrator and serverless:

- Manageability
- Flexibility and control
- Portability
- Application integration
- Cost
- Scalability

- In a serverless architecture, users give up some degree of control because management details are abstracted.
- An orchestrator gives the user good control over configuring and managing user services and the cluster.

Orchestrator vs. Serverless

These are the difference between orchestrator and serverless:

- Manageability
- Flexibility and control
- **Portability**
- Application integration
- Cost
- Scalability

- Orchestrators can run on-premises or in multiple public clouds.

Orchestrator vs. Serverless

These are the difference between orchestrator and serverless:

- Manageability
- Flexibility and control
- Portability
- Application integration
- Cost
- Scalability

- Building a complex application using serverless is difficult. Users can opt for Azure Logic Apps to coordinate a set of Azure Functions.

Orchestrator vs Serverless

These are the difference between orchestrator and serverless:

- Manageability
- Flexibility and control
- Portability
- Application integration
- Cost
- Scalability

- With a serverless application, the user pays for the actual compute resources consumed.
- With an orchestrator, the user pays for the VMs running in the cluster.

Orchestrator vs Serverless

These are the difference between orchestrator and serverless:

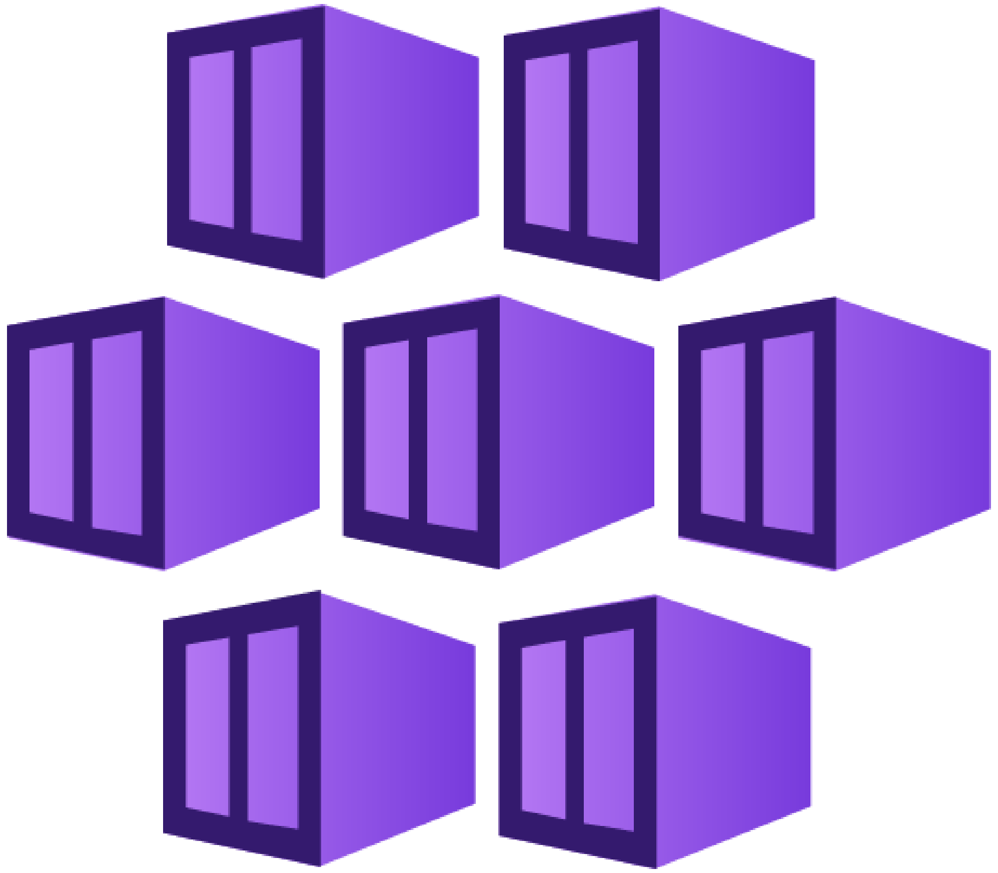
- Manageability
- Flexibility and control
- Portability
- Application integration
- Cost
- Scalability

- An orchestrator scales out by increasing the number of service instances running in the cluster, or, adding additional VMs to the cluster.
- However, serverless like Azure Functions scale automatically.

Recommend a Solution for Containers

Azure Kubernetes Service

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



Azure Kubernetes Service

The Azure Kubernetes Service provides the following features:

Manages container-based applications

- It manages networking and storage requirements alongside
- It focuses on application workloads instead of infrastructure components

Describes the applications declaratively

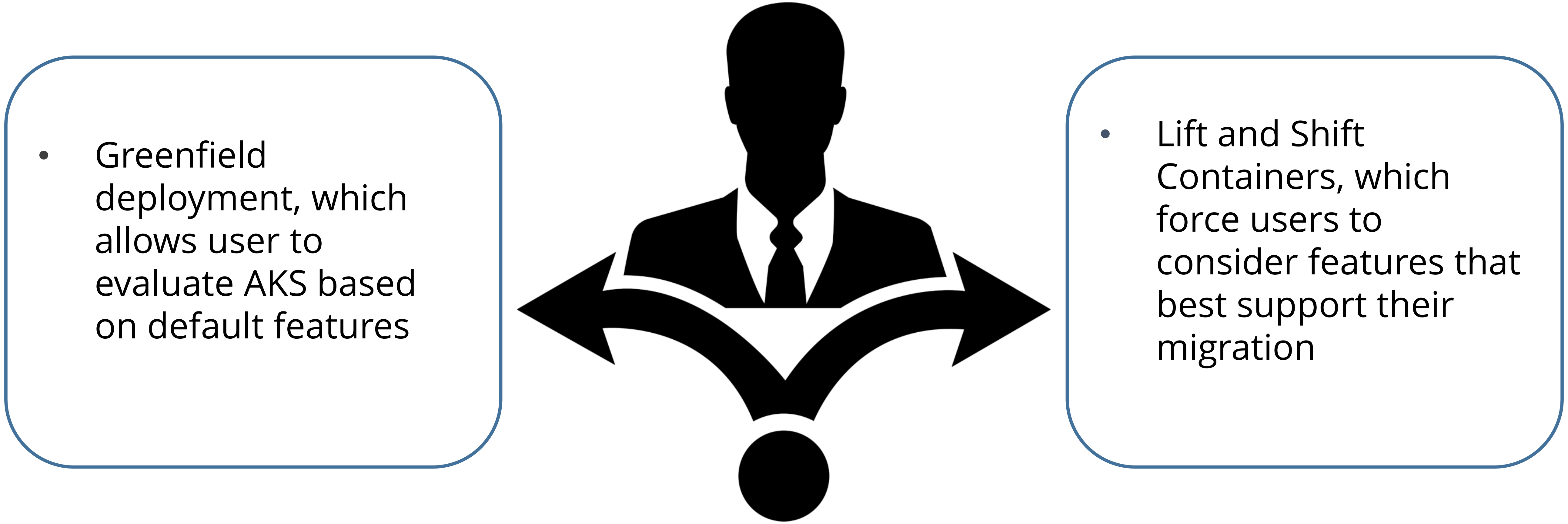
- YAML files can be used to describe application
- Kubernetes handles management and deployment

Makes it easier to orchestrate large solutions using a variety of containers

- Application containers
- Storage containers
- Middleware containers

When to Use Azure Kubernetes Service

Users would approach the decision based on:



- Greenfield deployment, which allows user to evaluate AKS based on default features

- Lift and Shift Containers, which force users to consider features that best support their migration

When to Use Azure Kubernetes Service

These are various features considered while deciding when to use Azure Kubernetes service:

Feature	Considerations and Decisions
Identity and Security	AD Integration
Logging and Monitoring	Azure Monitor for containers and custom monitoring solutions
Auto Scaling	Manual and Auto, Scheduled and Programmatically, Horizontal and Vertical
Cluster Node Upgrade	Azure manages cluster upgrade and software updates
GPU Support	GPU-enabled node pools

When to Use Azure Kubernetes Service

These are various features considered while deciding when to use Azure Kubernetes service:

Feature	Considerations and Decisions
Storage Volumes	Static and dynamic storage volumes
Virtual Network (VNet)	Deployed into existing VNet
HTTP Routing	HTTP application routing support is an add on
Docker Image	Default support for docker file image support
Private Container Registry	ACR (Azure Container Registry) Public and Private repositories

Azure Container Instance

Develop apps quickly without needing to manage virtual machines or learn new tools

- It does not require IaaS provisioning.
- It doesn’t require the adoption of a higher-level service.
- It supports Linux and Windows containers.
- It supports direct mounting of Azure Files Shares.

Azure Container Instance

Azure Container Instance has the following features:

- Run Containers without managing servers.
- Increase agility with containers on demand
- Secure applications with hypervisor isolation

A container can be provisioned with a public IP address and DNS name.

When to Use Azure Container Instances

Feature	Description
Fast startup times	Containers can start in seconds without the need to provision and manage VMs.
Public IP connectivity and DNS name	Containers can be directly exposed to the internet with an IP address and a fully qualified domain name (FQDN).
Hypervisor-level security	Container applications are as isolated in a container as they would be in a VM.
Custom sizes	Container nodes can be scaled dynamically to match actual resource demands for an application.

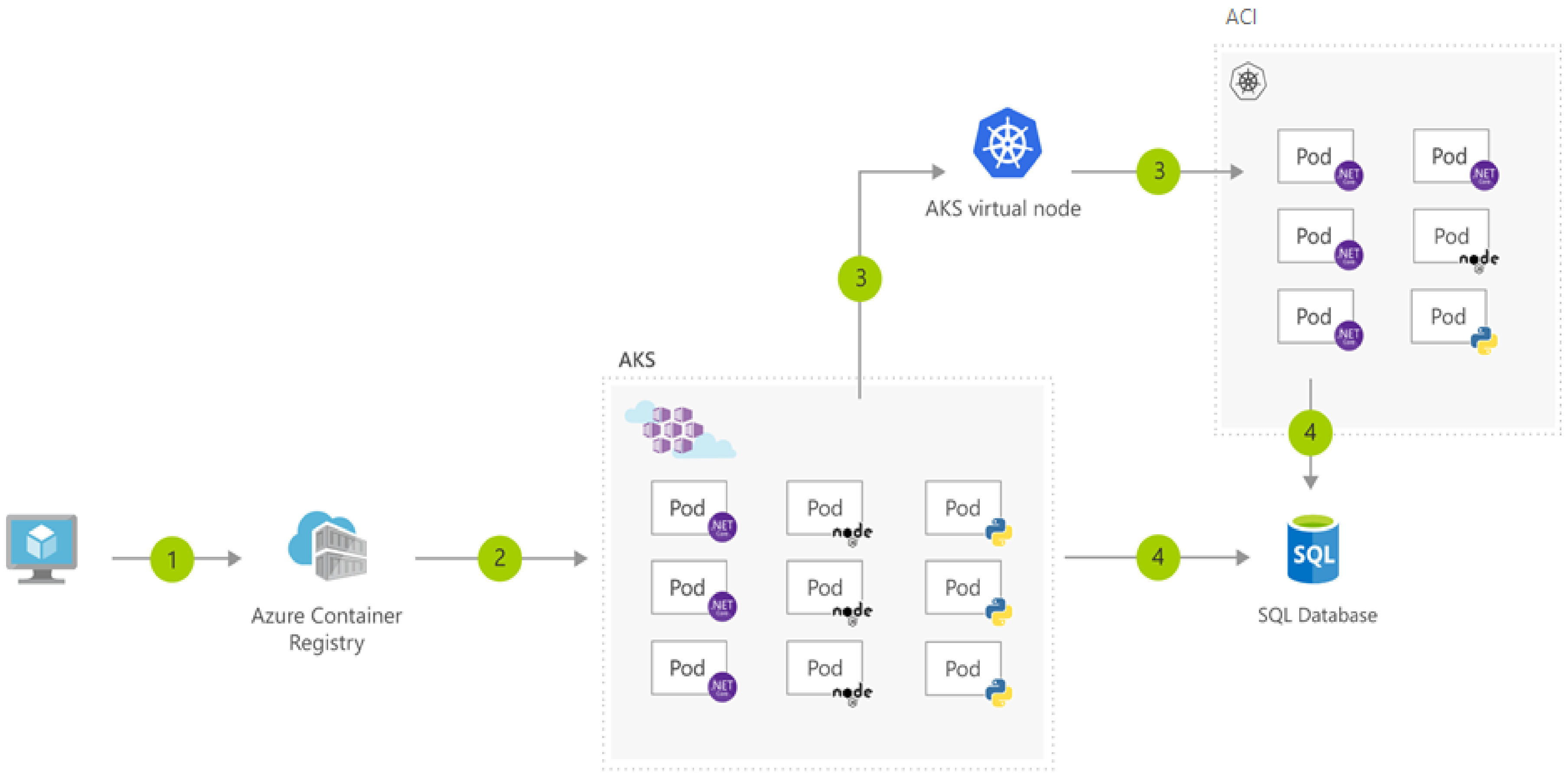
When to Use Azure Container Instances

Feature	Description
Persistent storage	Containers support direct mounting of Azure Files shares.
Linux and Windows containers	The same API is used to schedule both Linux and Windows containers.
Co-scheduled groups	Container Instances supports scheduling of multi-container groups that share host machine resources.
Virtual network deployment	Container Instances can be deployed into an Azure virtual network.

Recommendation

For scenarios where users need full container orchestration, including service discovery across multiple containers, automatic scaling, and coordinated application upgrades, use Azure Kubernetes Service (AKS).

Bursting from AKS with ACL



- A user registers a container in Azure Container Registry.
- AKS virtual node, a Virtual Kubelet implementation, provisions pods inside ACI from AKS when traffic spikes.
- AKS and ACI containers write to the shared datastore.

Source: https://docs.microsoft.com/

Assisted Practice

Create a Container Instance

Duration: 10 Min.

Problem Statement:

As an Azure architect, you've been asked to provide your organization with an Azure solution that can provide the quickest and most straightforward approach to host a container in Azure, without the need to manage any virtual machines or adopt a higher-level service.

Assisted Practice: Guidelines

Steps to create a container instance are:

1. Log in to your Azure portal
2. Click on Create a resource
3. Select Containers and click on Container Instances
4. Create a Container Instance

Assisted Practice

Create a Container Registry

Duration: 10 Min.

Problem Statement:

Demonstrate ACR, a private registry for Docker container images hosted on Azure.

Assisted Practice: Guidelines

Steps to create a container registry are:

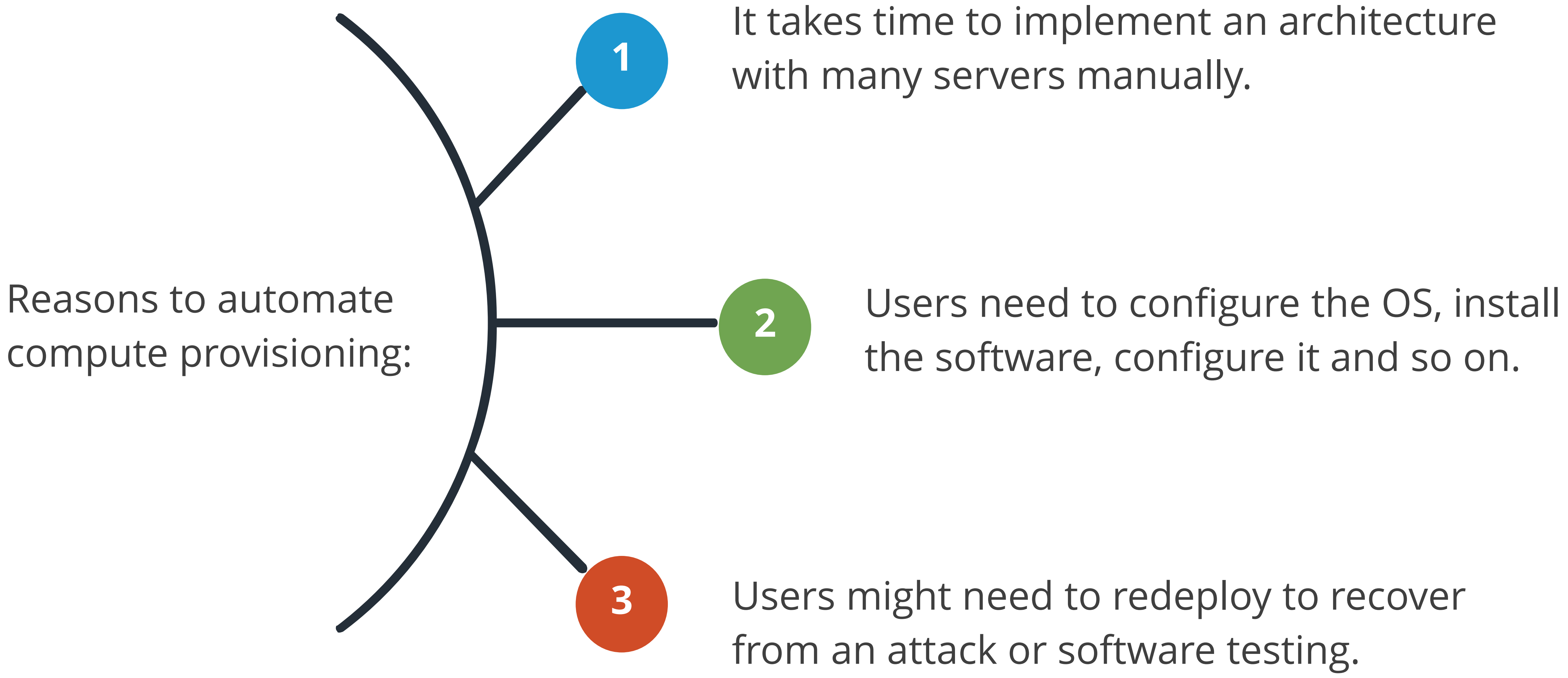
1. Log in to your Azure portal
2. Click on Create a resource
3. Click on Containers, and select Container Registry
4. Create a container registry by providing the required details

Recommend a Solution for Automating Compute Management

Provisioning Solution for Azure Compute Infrastructure

Creating and managing compute resources manually require a lot of time, as they are repetitive tasks. As a result, users may want to automate these processes.

Why Automate Compute Provisioning?



Reasons to automate compute provisioning:

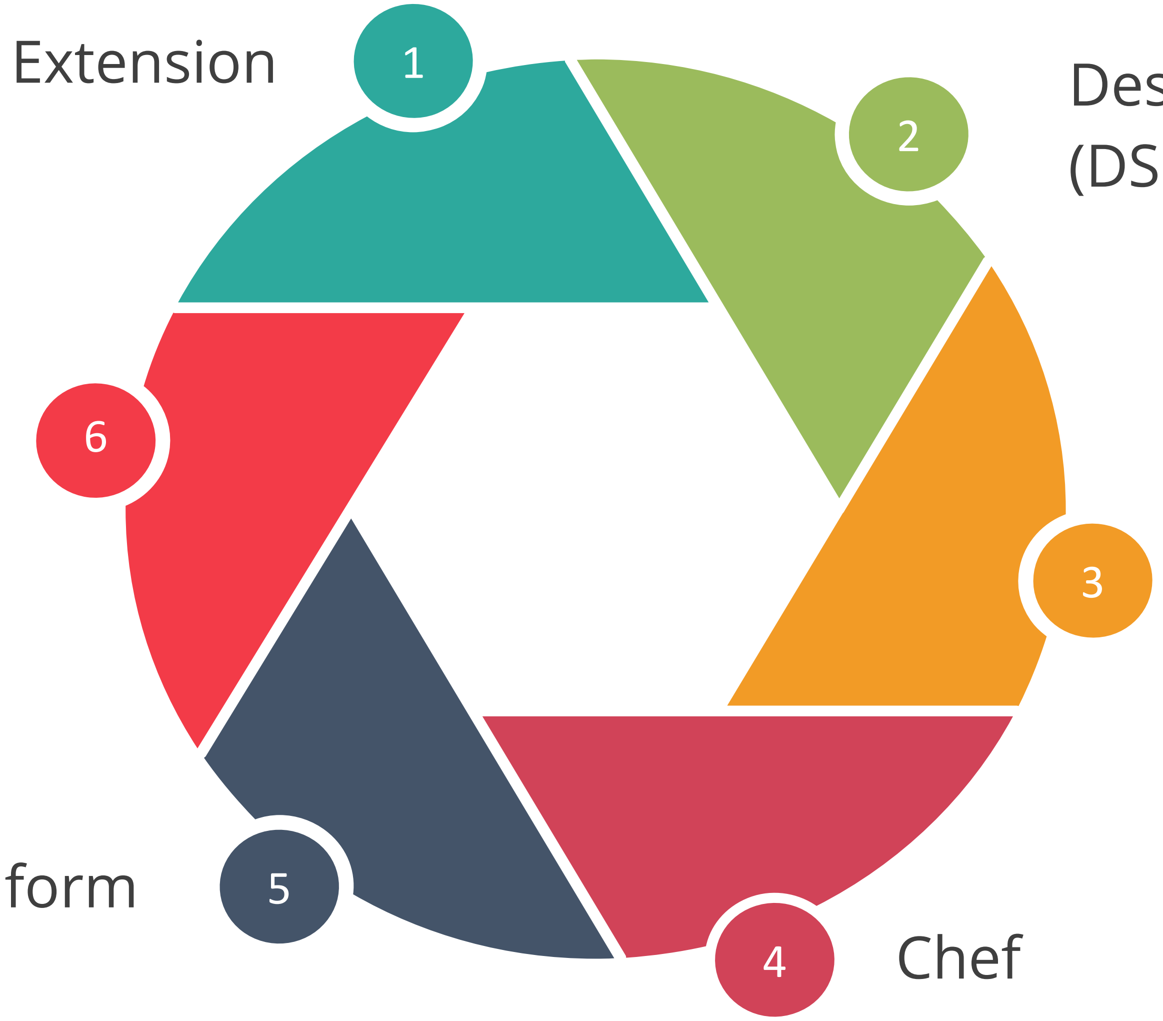
1 It takes time to implement an architecture with many servers manually.

2 Users need to configure the OS, install the software, configure it and so on.

3 Users might need to redeploy to recover from an attack or software testing.

Automate Infrastructure Deployment

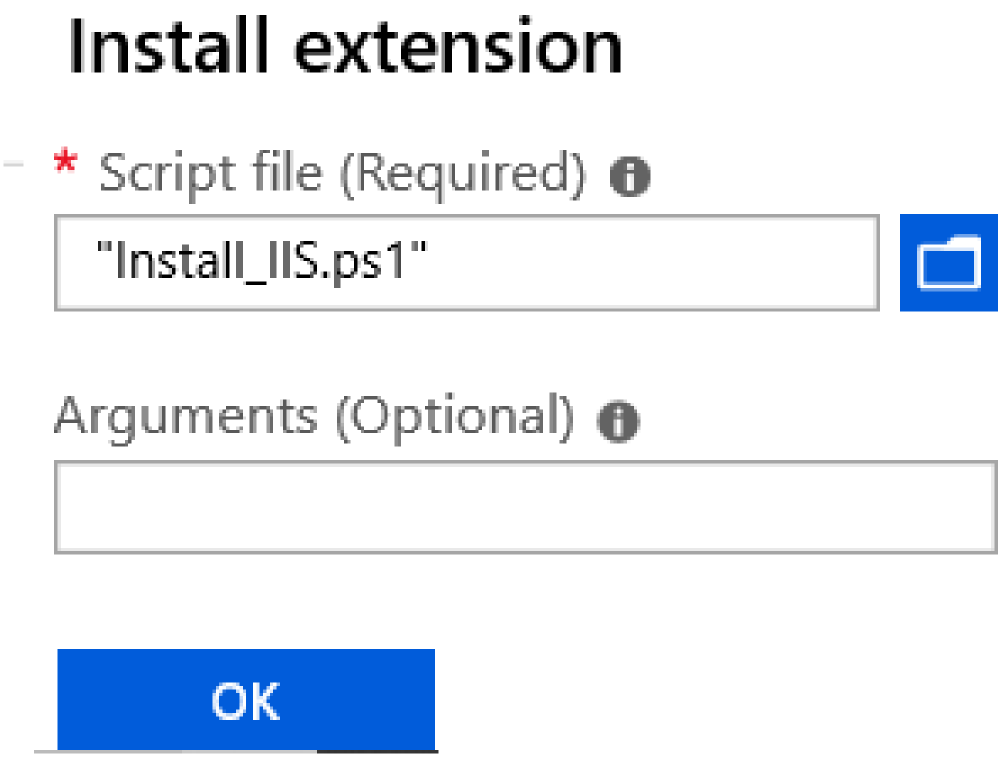
The different services, features, and tools that helps to automate the Infrastructure provisioning are:



1
Custom Script Extension
2
Desired State Configuration (DSC) Extensions
3
Azure Automation State Configuration
4
Chef
5
Terraform
6
Azure Resource Manager Templates

Custom Script Extension

The Custom Script Extension tool offers the following features:



Install extension

* Script file (Required)

"Install_IIS.ps1"

Arguments (Optional)

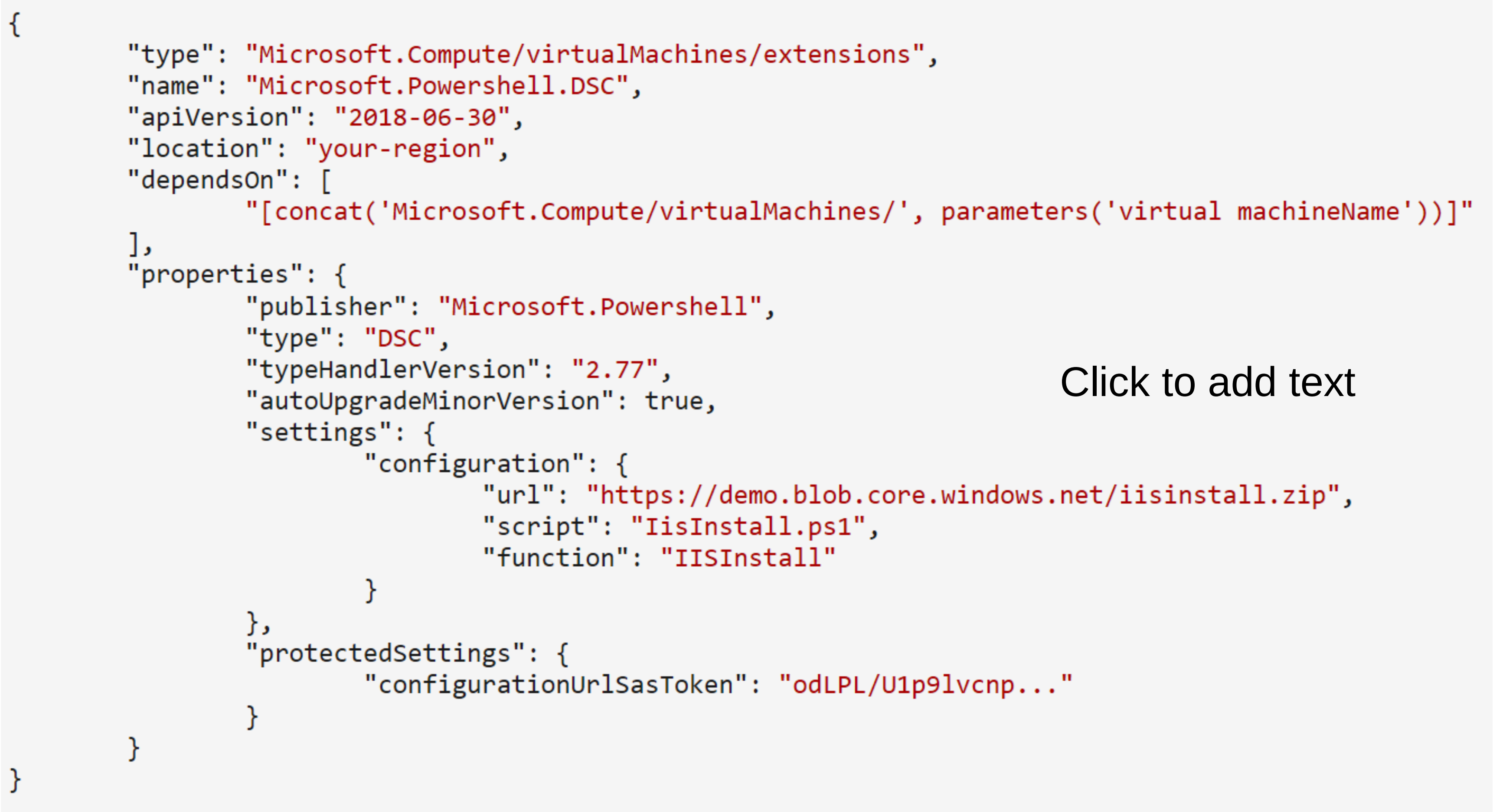
- PowerShell scripts on a file server, GitHub, Azure Storage, or other location can be accessible to a VM.
- An extension looks for a script that should be run on the VM.
- The script is downloaded and executed on the target VM to apply the changes described in the script.
- Users can add a custom script extension to a virtual machine through Azure Resource Manager templates, PowerShell, or the Azure CLI

Desired State Configuration (DSC) Extensions

DSC defines a state for machines instead of writing manual instructions on how to achieve a state for each machine.

- A DSC extension handler can enforce a user's states.
- The configurations for states can be in various locations, such as Azure Blob storage and internal file storage.

Desired State Configuration (DSC) Extensions



```
{
        "type": "Microsoft.Compute/virtualMachines/extensions",
        "name": "Microsoft.Powershell.DSC",
        "apiVersion": "2018-06-30",
        "location": "your-region",
        "dependsOn": [
                "[concat('Microsoft.Compute/virtualMachines/', parameters('virtual machineName'))]"
        ],
        "properties": {
                "publisher": "Microsoft.Powershell",
                "type": "DSC",
                "typeHandlerVersion": "2.77",
                "autoUpgradeMinorVersion": true,
                "settings": {
                        "configuration": {
                                "url": "https://demo.blob.core.windows.net/iisinstall.zip",
                                "script": "IisInstall.ps1",
                                "function": "IISInstall"
                        }
                },
                "protectedSettings": {
                        "configurationUrlSasToken": "odLPL/U1p9lvcnp..."
                }
        }
}
```

Click to add text

- The DSC extension handler grabs the configuration and implements the state on the target VM.

- If reboots are necessary, DSC continues to execute the state configuration after the reboots are completed.

Desired State Configuration

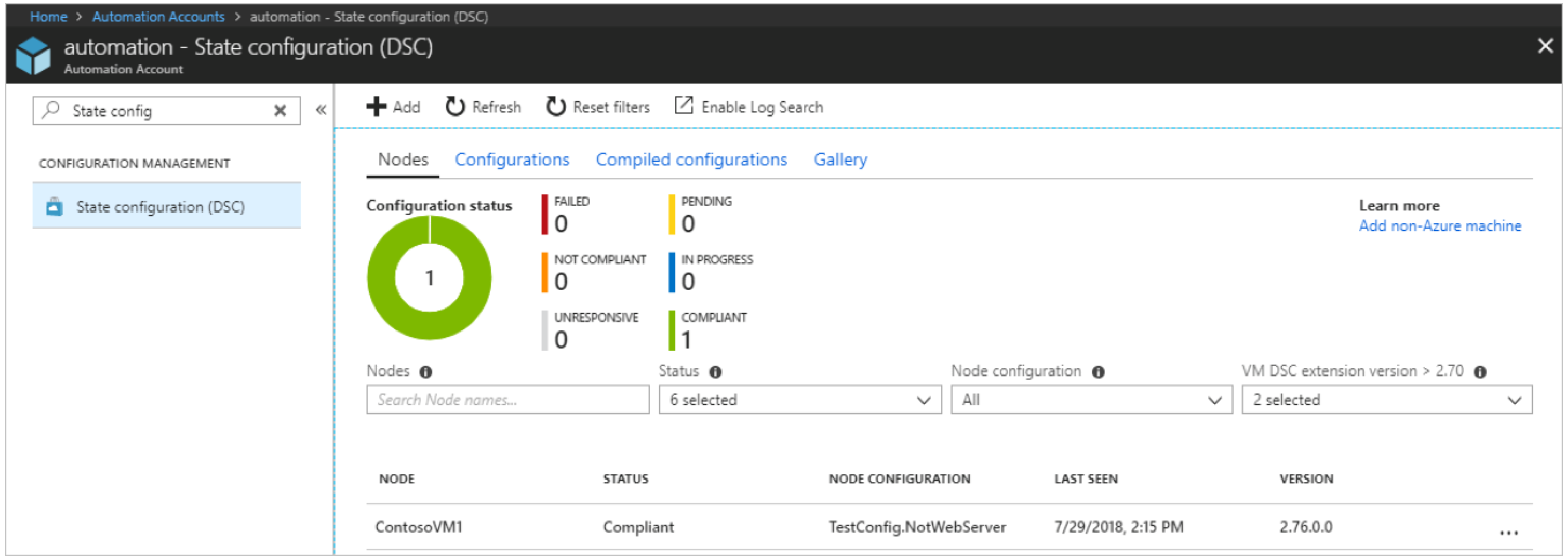
The below mentioned are features of Desired state configuration :

```
configuration IISInstall
{
  Node “localhost”
  {
     WindowsFeature IIS
     {
         Ensure = “Present”
         Name = “Web-Server”
      }
   }
}
```

- Configuration block(s) have a name.
- Node blocks define the computers or VMs that a user configures.
- Resource block(s) configure the resource and its properties.
- There are many built-in configuration resources.

Azure Automation State Configuration

Azure Automation State Configuration service helps in managing and deploying DSC configurations.



Configuration status: 1

FAILED 0
PENDING 0
NOT COMPLIANT 0
IN PROGRESS 0
UNRESPONSIVE 0
COMPLIANT 1

Nodes: Search Node names...
Status: 6 selected
Node configuration: All
VM DSC extension version > 2.70: 2 selected

NODE	STATUS	NODE CONFIGURATION	LAST SEEN	VERSION
ContosoVM1	Compliant	TestConfig.NotWebServer	7/29/2018, 2:15 PM	2.76.0.0

Source: https://docs.microsoft.com/

Azure Automation State Configuration

These are the ways in which Azure automation state configuration manages a variety of machines:

- Azure Virtual Machines
- Azure Virtual Machines (Classic)
- Physical/Virtual Windows machines on-premises or in a cloud other than Azure (including AWS EC2 instances)
- Physical/virtual Linux machines on-premises, in Azure, or in cloud other than Azure

Chef

Chef automates infrastructure deployment, on-premises or in the cloud.

- Chef server can handle 10,000 nodes (machines) at a time.
- Chef hosted, and it runs as a service.
- Chef server is used to manage recipes.
- Recipes are commands to run configurations.

Chef's Knife tool is used to deploy VMs and simultaneously apply recipes to VMs.

Terraform

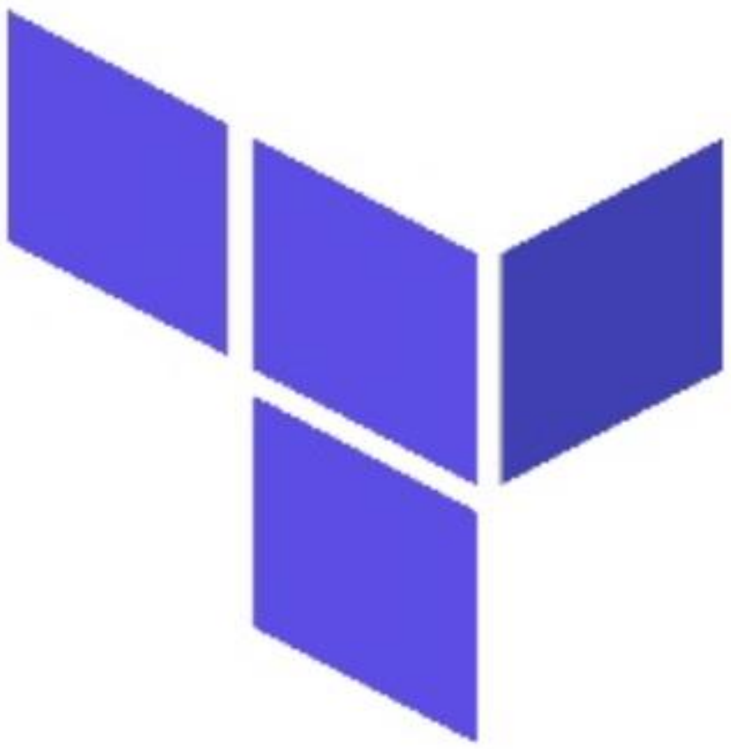
Terraform is an open-source infrastructure-as-code software tool.

Terraform Tool

- Executes Terraform scripts.
- Uses Hashicorp Configuration Language (HCL) or Json.
- Provides easy-to-read script templates, defining resources to create.
- Can be applied to different cloud service providers (Azure, AWS).
- Requires Terraform installation (VM/CloudShell).

Automate infrastructure management

Terraform's template-based configuration file allows you to define, provision, and configure Azure resources.



Benefits

- Reduces human errors when deploying and managing errors
- Deploys the same template multiple times to create identical development, test, and production environment
- Reduces the cost of development and test environments

Source: https://docs.microsoft.com/

Azure Resource Manager (ARM) Templates

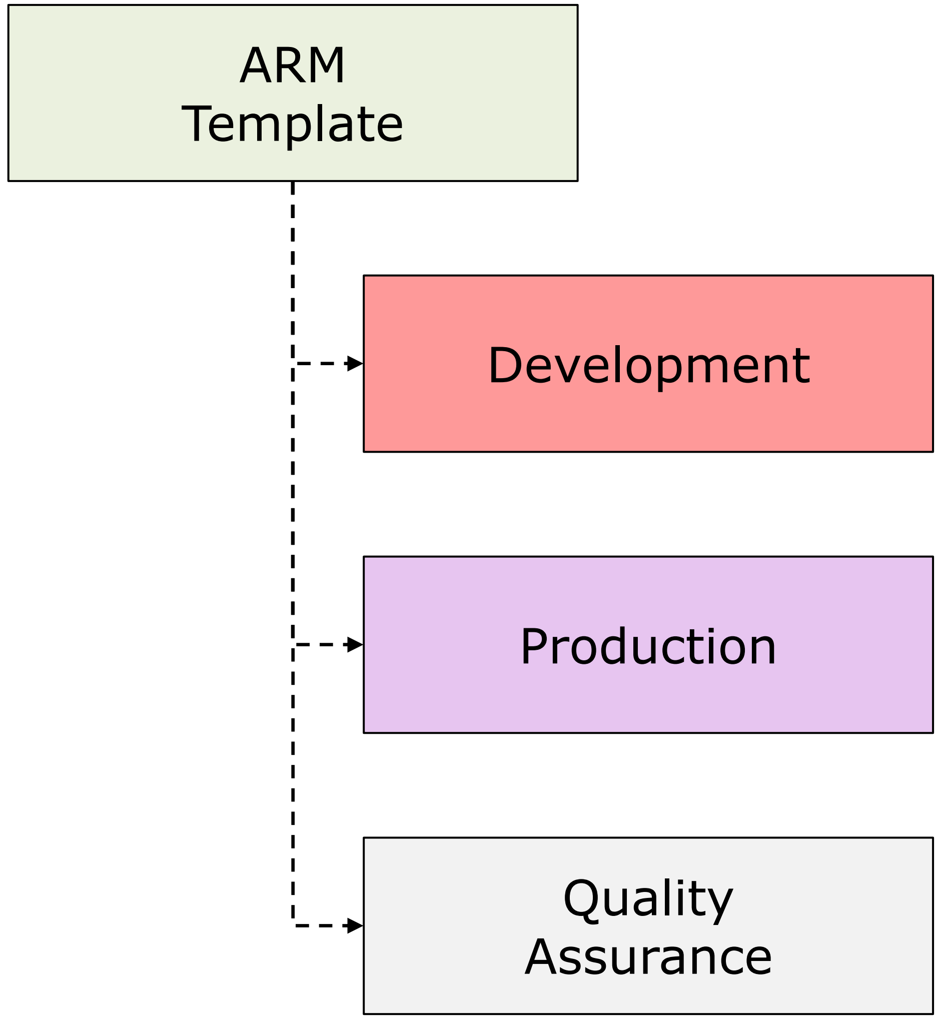
ARM templates are JSON files used to define the Azure resources for deployment.

```
"$schema" :
"http://schema.management.azure.com/schemas/2015-01-
01/deployment Template.json#",
"contentVersion": "1.0.0.0",
"parameters": {
"location": {
"type": "string"
},
"storageAccountName": {
"type": "string"
},
"accountType": {
"type": "string"
},
"kind": {
"type": "string"
},
"accessTier": {
"type": "string"
},
"minimumTlsVersion": {
"type": "string"
},
"supportsHttpsTrafficOnly": {
"type": "bool"
},
"allowBlobPublicAccess": {
"type": "bool"
},
```

- It enables embedding extensions into VMs.
- It helps in deployment using Azure PowerShell, the Azure CLI, or the Azure portal.
- It allows testing for ARM templates before deployment.

Template Advantages

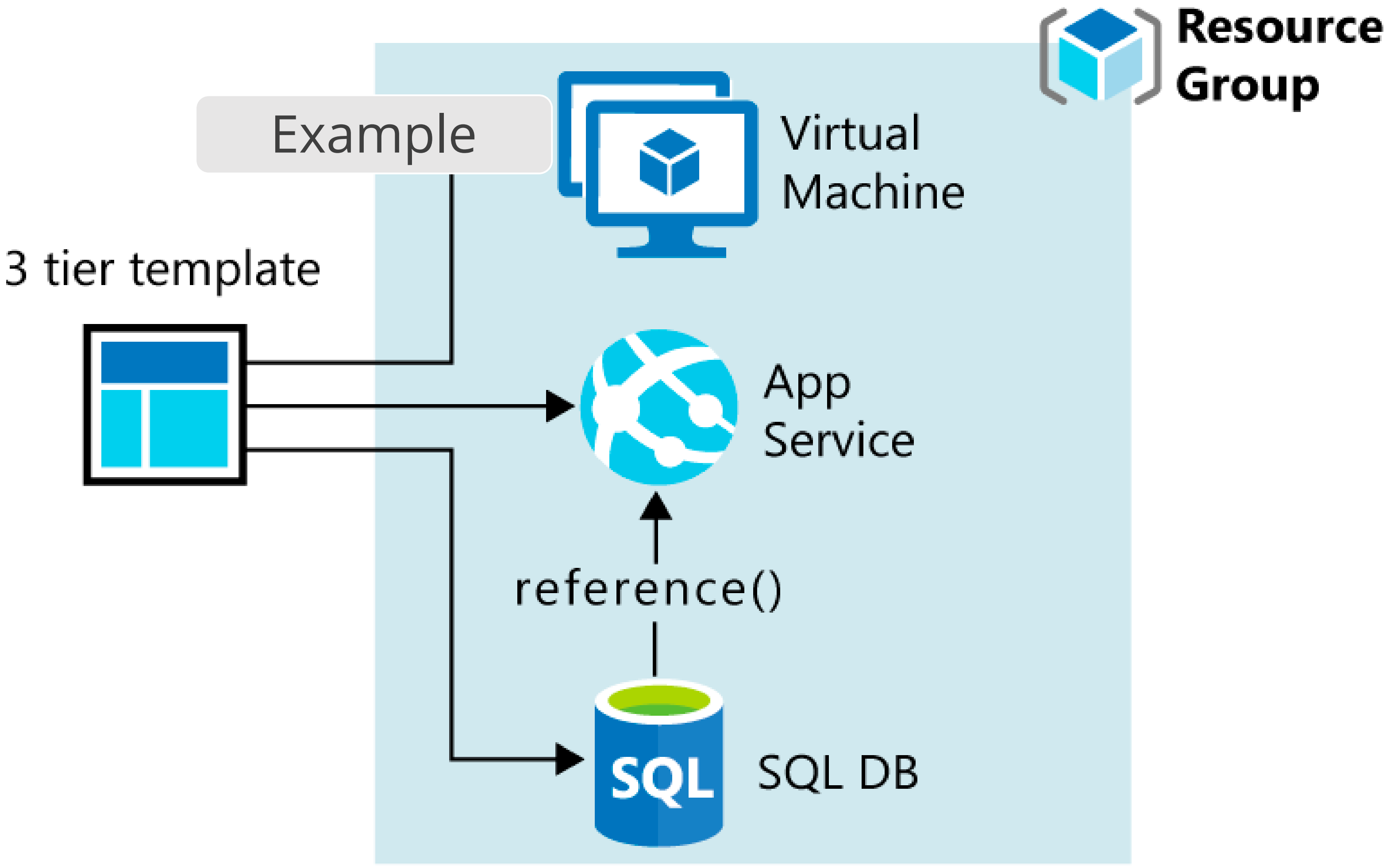
Using Resource Manager templates, one can make deployments faster and more repeatable.



- Improves consistency.
- Express complex deployments.
- Reduce manual, error prone tasks.
- Express requirements through code.
- Promotes reuse.
- Is modular and can be linked.
- Simplifies orchestration

Template Design

Users can define templates and resource groups depending on how they want to manage the solution.

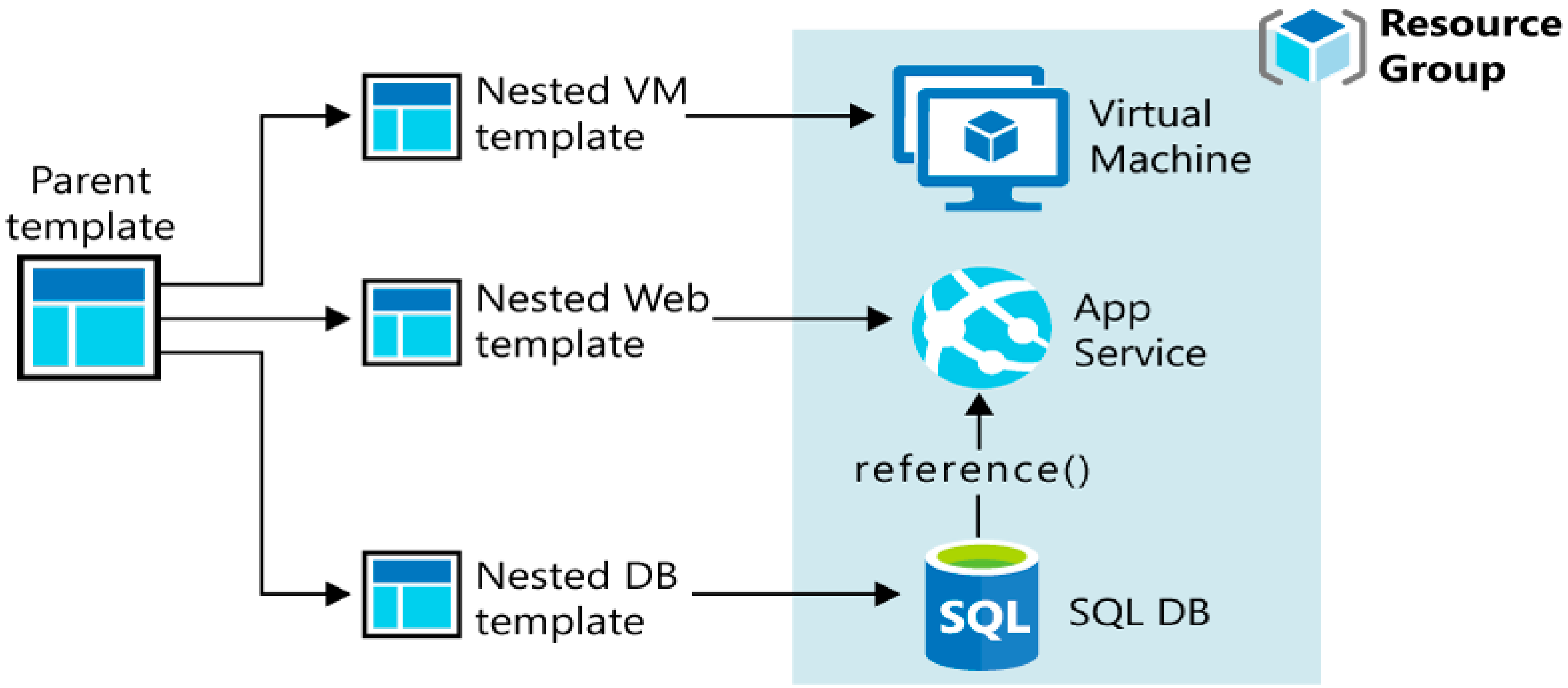


Users can deploy their three-tier application through a single template to a single resource group.

Source: https://docs.microsoft.com/

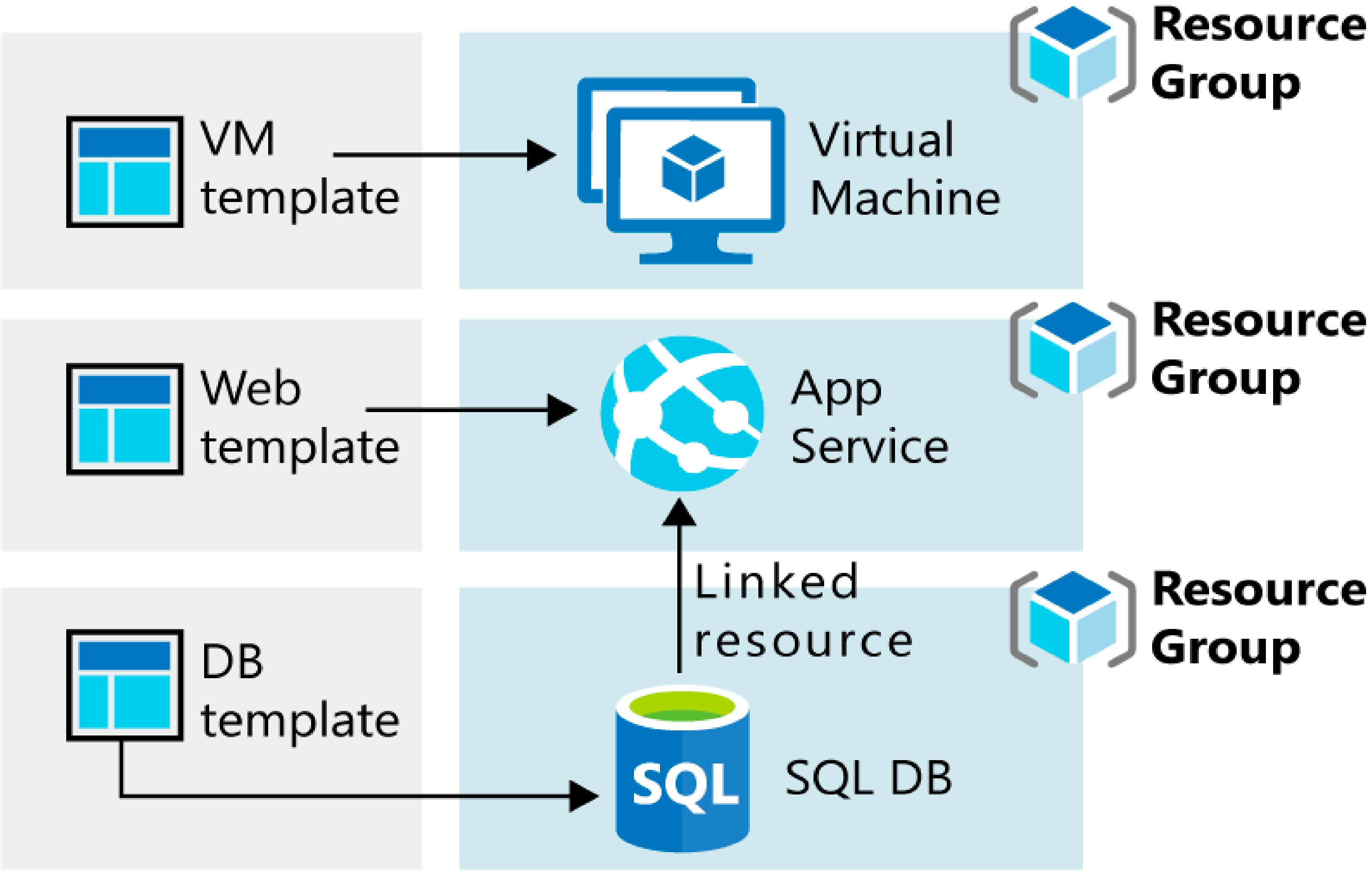
Template Design

Users can divide their deployment requirements into a set of targeted, purpose-specific templates.



Template Design

If user tiers have separate life cycles, users can deploy three tiers to separate resource groups.



Key Takeaways

- AKS is a managed service for running large-scale parallel and high-performance computing (HPC) applications.

- Azure Functions is a serverless compute service that supports various function triggers.

- A container can be provisioned with a public IP address and DNS name.

- The DSC extension handler grabs the configuration and implements the state on the target VM.

- ARM Template enables embedding extensions into VMs.

Implement Web App

Duration: 25 min.

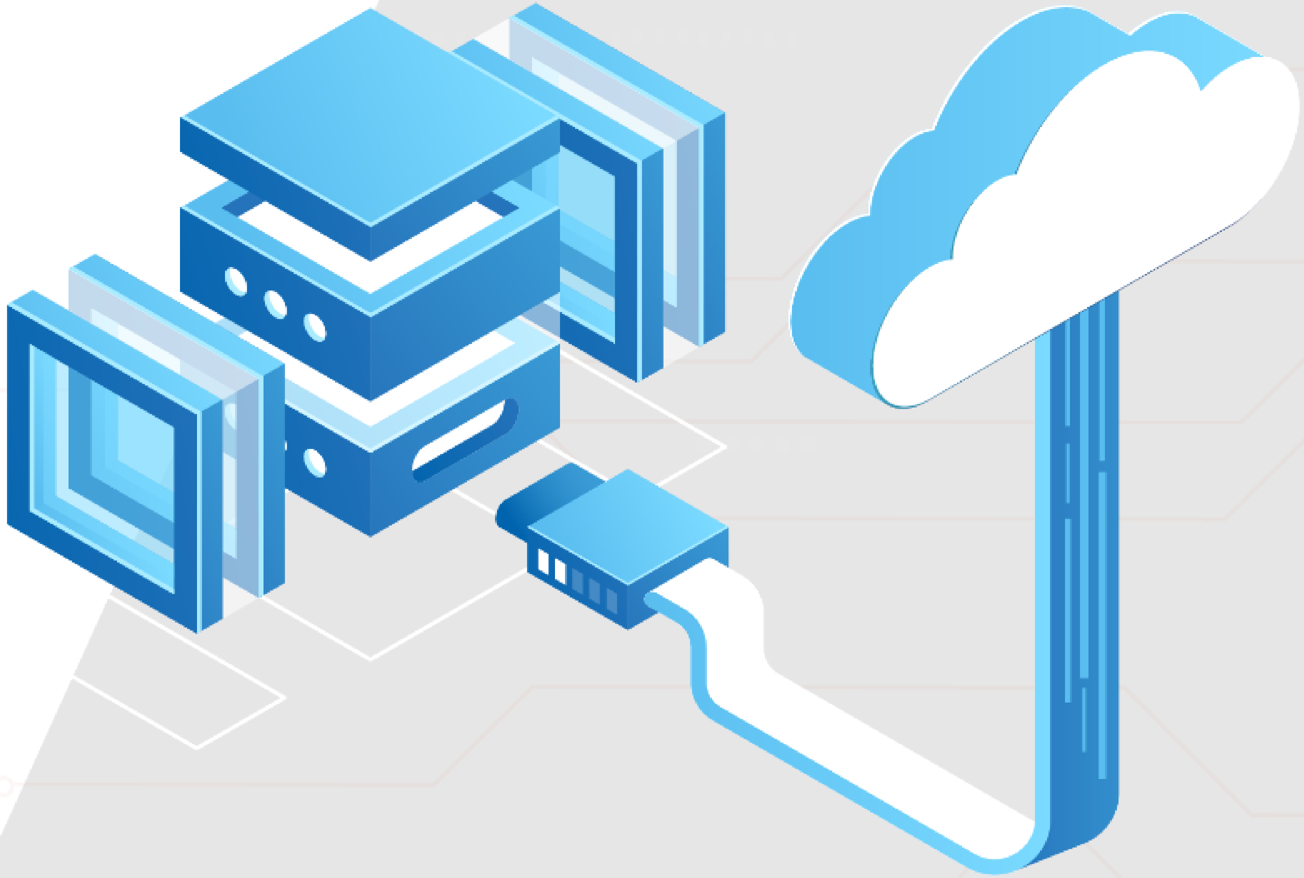
Project Agenda: To evaluate the use of Azure Web apps for hosting a web site

Objective: You have been given a project to evaluate the use of Azure Web apps for hosting Edu-sim's web sites, hosted currently in the company's on-premises data centers. The web sites are running on Windows servers using PHP runtime stack. You also need to determine how you can implement DevOps practices by leveraging Azure web apps deployment slots.

Perform the following:

1. Create an Azure web app
2. Create a staging deployment slot
3. Configure web app deployment settings
4. Deploy code to the staging deployment slot

LESSON-END PROJECT



Thank you