

# Azure Functions

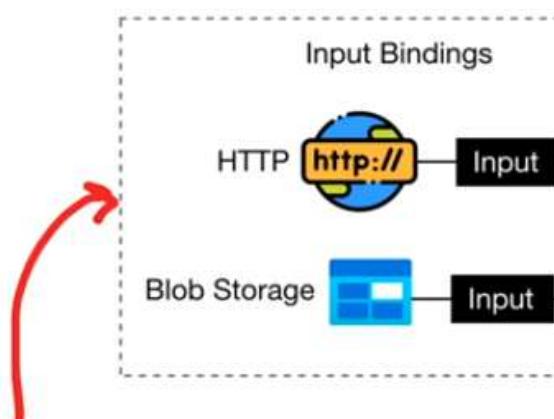
Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/az-204](http://www.exampro.co/az-204)



**Azure Functions** is a Function as Service (FaaS) offering that allows developers to focus on writing code and not worry about maintaining the underlying computing infrastructure.

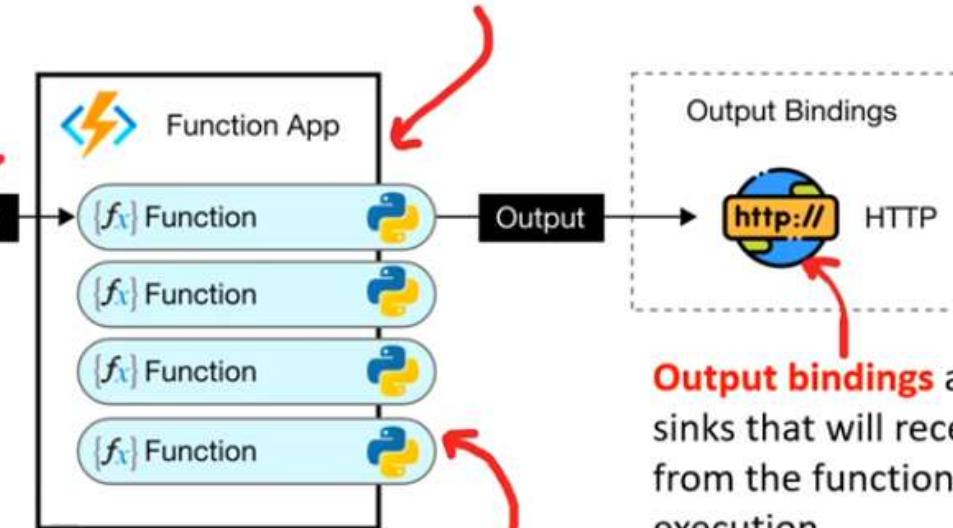
A **trigger** is the chosen event data that will cause function to execute

- You can only have one trigger



**Input Bindings** are one or multiple data sources that will be passed to the function when a trigger occurs.

A **Function App** defines the underlying compute for a collection of functions. A Function App defines the Hosting, Runtime and other global configurations.



**Output bindings** are one more data sinks that will receive outputted data from the function on successful execution

A **function** represents code along with application runtime configuration

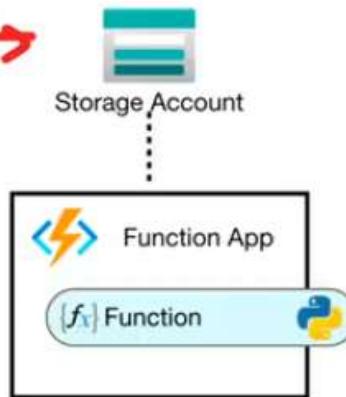
There are four versions of Azure Functions 1.x, 2.x, 3.x and 4.x. We are currently using **4.x**

# Azure Functions – Storage Considerations

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/az-204](http://www.exampro.co/az-204)

Every function app **requires a storage account to operate.** 

If that account is deleted, your functions won't work.



Azure Functions uses the following storage types in the Storage Account



## Blob Storage

Maintain bindings state and function keys



## Azure Files

File share used to store and run your function app code in a Consumption Plan and Premium Plan.

Azure Files is set up by default, but you can create an app without Azure Files under certain conditions.



## Queue Storage

Used by task hubs in Durable Functions.

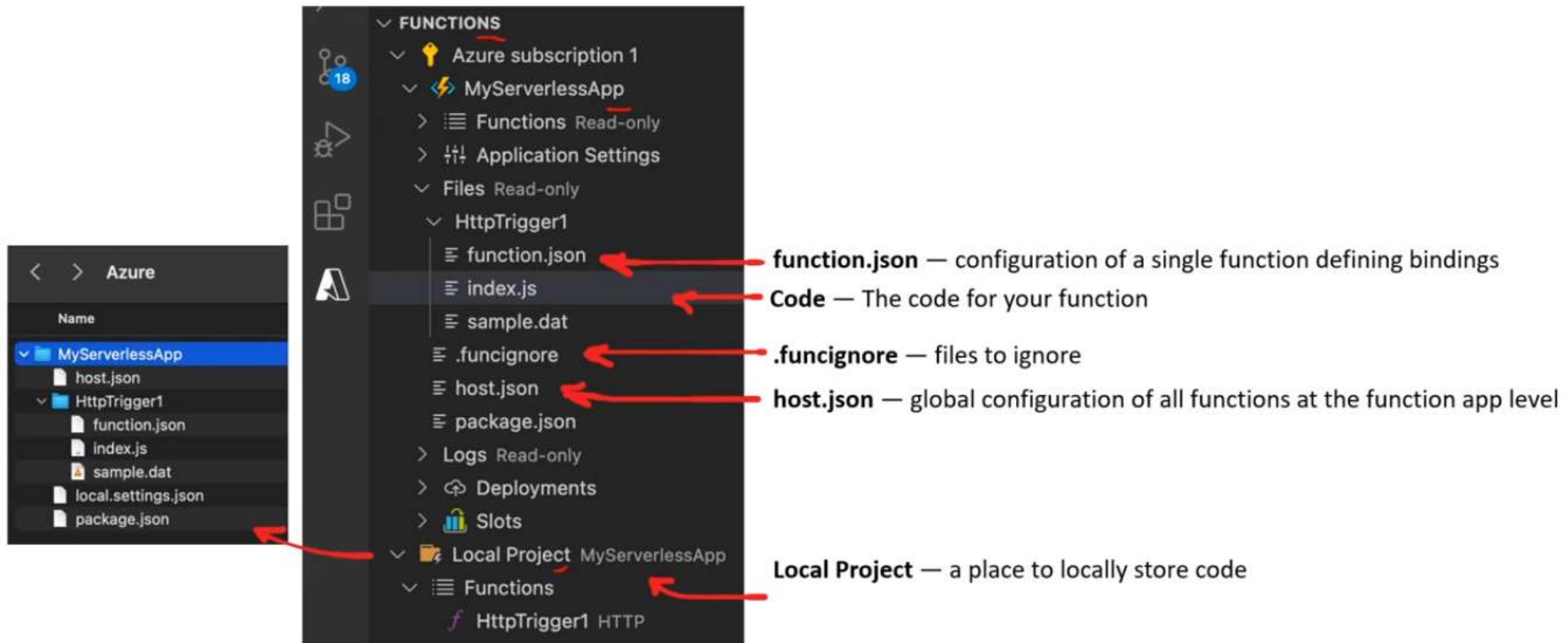


## Table Storage

Used by task hubs in Durable Functions.

# Azure Functions – Anatomy of a Function

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/az-204](http://www.exampro.co/az-204)



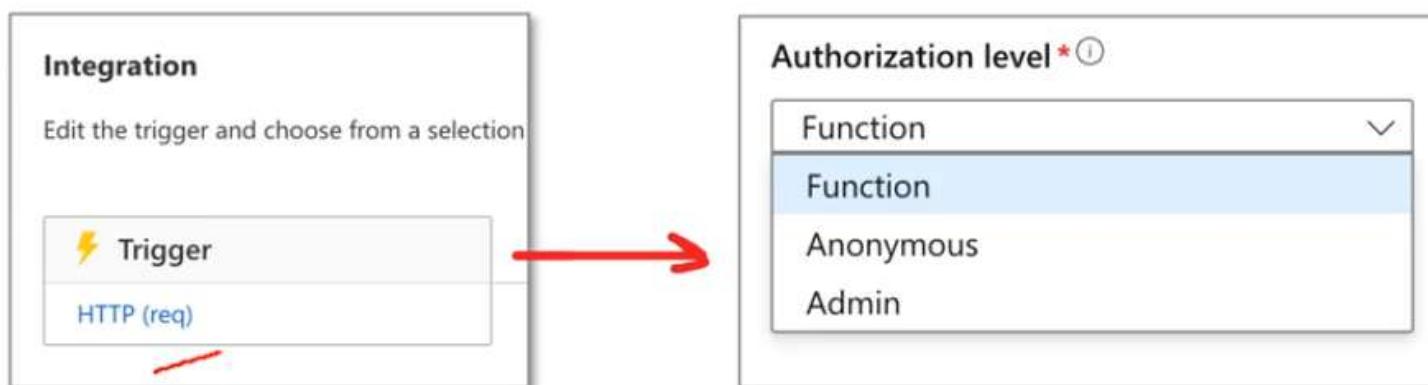
# Azure Functions HTTP – Authorization Levels

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/az-204](http://www.exampro.co/az-204)

**Authorization level** determines what keys, if any, need to be present on the request in order to invoke the function.

The authorization level can be one of the following values:

- anonymous—No API key is required.
- function—A function-specific API key is required (default value).
- admin—The master key is required.



Authorization level can usually be **changed** after creation in the portal, and is set on the trigger

# Azure Functions Debugging

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/az-204](http://www.exampro.co/az-204)

You can **enable streaming** logs for Azure Functions to see near real time logging when an error occurs

There are two ways to view a stream of log files being generated by your function executions

- **Built-in log streaming**
  - the App Service platform lets you view a stream of your application log files.
- **Live Metrics Stream**
  - when your function app is connected to Application Insights, you can view log data and other metrics in near real-time in the Azure portal using Live Metrics Stream



Log streams can be viewed both in the portal and in most local development environments.

The screenshot shows the Azure Functions blade in the Azure portal. On the left, there's a sidebar with icons for Resource Groups, Help and Feedback, Functions, Files, Logs, Deployments, and Slots. The main area shows a list of functions under 'FUNCTIONS'. One function, 'hello', is selected. A context menu is open over this function, with the 'Start Streaming Logs' option highlighted in blue. Other options in the menu include 'Copy Function Url', 'Execute Function Now...', 'Disable Function', 'Stop Streaming Logs', and 'View Properties'. The 'hello' function has three logs listed: 'hello', 'order', and 'queueTrigger'. The 'queueTrigger' log has a timestamp of '17 minutes ago'. At the bottom right, there's a 'PROB' button.

# Azure Functions – Key Concepts

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/az-204](http://www.exampro.co/az-204)

- Azure Functions are **lightweight** and (can be) **serverless**.
- Azure Functions are easier to **write and deploy**.
- Azure Functions are **fast** to execute because there is no large application, startup time, initialization, and other events fired before the code is executed.
- Azure Functions' execution is **triggered** when an **event** is fired.
- Azure Functions do not need any **infrastructure** and have **zero maintenance**.
- Azure Function can be **built, tested, and deployed** in Azure portal using a browser.
- Azure Functions are **easy to upgrade** and doesn't affect other parts of the website.
- Azure Functions use industry standard protocols and can **communicate** with other **APIs, databases, and libraries**
- You only **pay** while your functions are running
- Azure functions automatically scale to meet the demand of traffic
  - Underneath Azure will provision more serverless containers
- Azure Functions can Scale-to-Zero cost.  
Azure Functions has built in monitoring via Azure Monitor
- Azure Functions has built in CI/CD via Azure DevOps
- Azure Functions are event-driven, and will trigger based on event data, and emit event data

# Azure Functions – Use Cases

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/az-204](http://www.exampro.co/az-204)

## Business Use-Cases

- Scheduled Tasks
- Reminders and Notifications
- Lightweight Web API
- Sending background emails
- Running background backup tasks
- Doing backend calculations

## Technical Use-Cases

- sending emails
- starting backup
- order processing
- task scheduling
  - database cleanup
  - sending notifications
  - Messages
  - IoT data processing



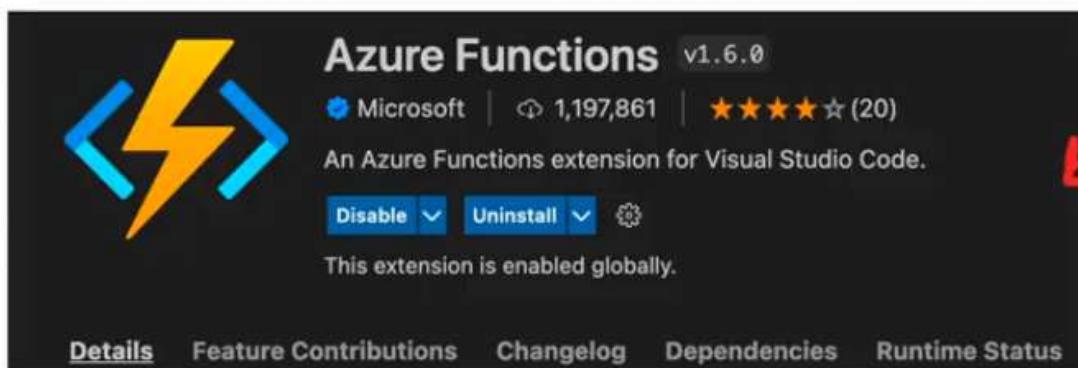
Azure functions are best suited for **smaller** apps have **events** that can work **independently** of other websites.

# Azure Functions – VSCode Extension

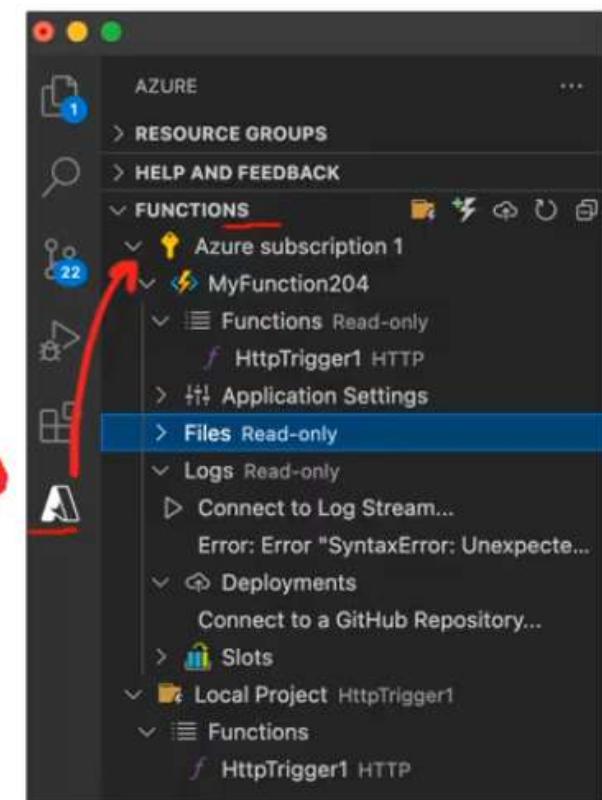
Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/az-204](http://www.exampro.co/az-204)



In order to productively work with Azure Functions you'll need to use  
**Visual Studio Code (VSCode)** and also install the **Azure Functions extension**



Once installed you'll have an Azure icon in the  
**Activity Bar** with a Side Bar drawer called **Functions**



It is possible to use Azure Portal to create an update Windows Hosted  
Azure Functions, but in practice is not an easy experience



Visual Studio can be used for .NET (C#) App functions

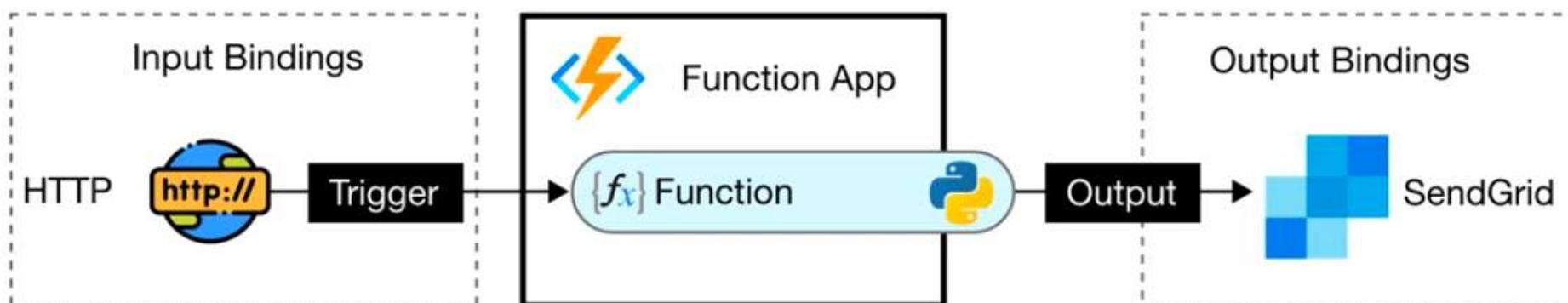
# Azure Functions – Triggers and Bindings Scenarios

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/az-204](http://www.exampro.co/az-204)



## Scenario 2:

Every-time someone signs up to your application you want to trigger a welcome email.



## HTTP

You want to develop an **API** that allows you to send an **email** after a request is received.

You'll use **HTTP** trigger because it's an API that will be triggered based on this request

For the **bindings**, you won't be accessing any data when the function starts so the **In** direction is **None** and you'll use **SendGrid** for the **Out** direction which allows you to send messages via email.

# Azure Functions – Runtimes

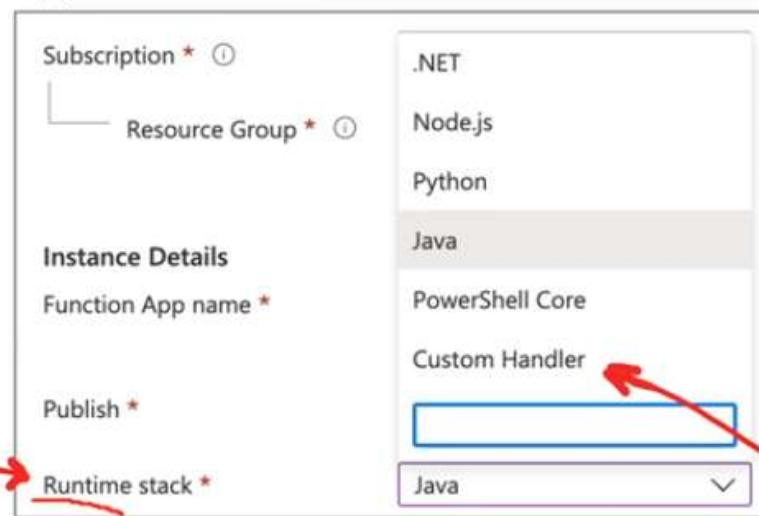
Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/az-204](http://www.exampro.co/az-204)

## What is an Application Runtime?

An application runtime, is compute environment configured with the necessary software and system configuration to run a specific type of application code.

Azure provides multiple application runtimes for popular programming languages (**Runtime stack**):

- .Net (C#)
- Java
- Node.js (JavaScript, TypeScript)
- PowerShell Core
- Python



If you want to use an unsupported or yet-to-be-supported language eg: (Rust, GO, Ruby) you can use a **custom handler**



The runtimes provided by Microsoft, are just Docker containers  
[https://hub.docker.com/\\_/microsoft-azure-functions](https://hub.docker.com/_/microsoft-azure-functions)

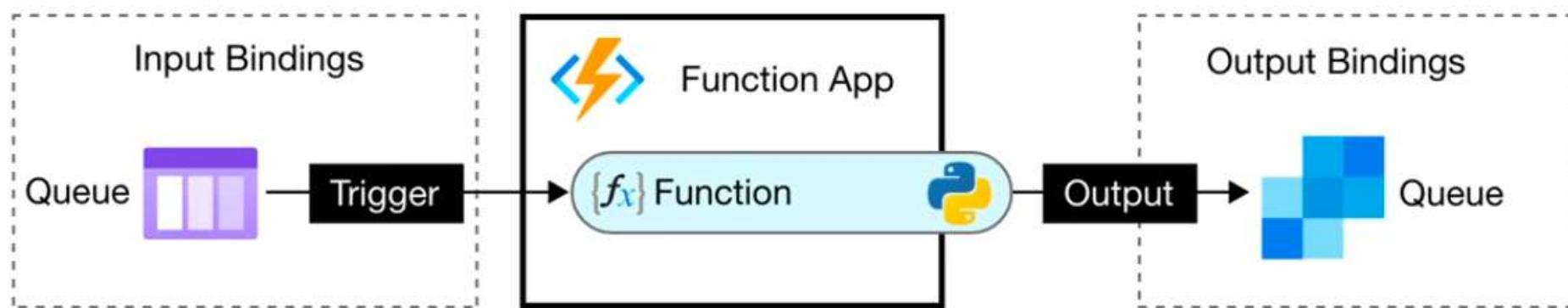
# Azure Functions – Triggers and Bindings Scenarios

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/az-204](http://www.exampro.co/az-204)



## Scenario 3:

Consider a scenario in which you're using a **queue service** and you want a function to process a **storage message** in one queue and enter a **new message** in another.



In this case, you'd want to use a **trigger type queue** and a **binding type queue** for direction **out** because you are **not** accessing any queue data for the **In direction**.

You read data from one queue, process it to create a new message, and then write it to a different queue, which may or may not be connected to another service or function.

# Azure Functions – Windows vs Linux Hosting

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/az-204](http://www.exampro.co/az-204)

When you create an Azure Function App you can  
**choose to host** it on either **Windows or Linux**



Operating system

The Operating System has been recommended for you based on your selection of runtime stack.

Operating System \*

Linux  Windows

- There could be performance differences
- There could be functionality limitations
- There could be feature differences



Generally Microsoft Azure better supports compute running Windows

When using Linux you cannot edit functions via the Azure Portal



 Create  Refresh |  Delete



Editing functions in the Azure portal is not supported for Linux Consumption Function Apps.

# Azure Functions – Binding Expressions

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/az-204](http://www.exampro.co/az-204)

In the *function.json* file and in function parameters and code, you can use **binding expressions** that resolve to values from various sources.

\*Most expressions are identified by wrapping them in **curly braces**. eg. `container/{queueTrigger}`

Types of binding expressions

- App settings
- Trigger file name
- Trigger metadata
- JSON payloads
- New GUID
- Current date and time

# Azure Functions – Windows vs Linux Hosting

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/az-204](http://www.exampro.co/az-204)

To determine the exact OS that is being used, you'll need to go to the Docker Hub repository for the Azure Functions and for your runtime



Windows Images					
Tags	Architecture	Dockerfile	OsVersion	CreatedTime	LastUpdatedTime
3.0-nanoserver-1809	amd64	No Dockerfile	Windows Server 1809	05/29/2020	03/04/2022
Linux Images					
Tags	Architecture	Dockerfile	OsVersion	CreatedTime	LastUpdatedTime
4	amd64	No Dockerfile	Debian 10	06/03/2021	03/03/2022
4-slim	amd64	No Dockerfile	Debian 10	08/26/2021	03/03/2022
4-appservice	amd64	No Dockerfile	Debian 10	06/03/2021	03/03/2022
4-dotnet6-appservice	amd64	No Dockerfile	Debian 10	08/26/2021	03/03/2022
3.0	amd64	No Dockerfile	Debian 10	12/06/2019	03/04/2022
3.0-slim	amd64	No Dockerfile	Debian 10	04/15/2020	03/04/2022
3.0-appservice	amd64	No Dockerfile	Debian 10	12/06/2019	03/04/2022
3.0-dotnet3-appservice	amd64	No Dockerfile	Debian 10	12/06/2019	03/04/2022
3.0-arm32v7	arm32v7	No Dockerfile	Debian 10	04/15/2020	03/04/2022
3.0-bionic-arm32v7	arm32v7	No Dockerfile	Ubuntu 18.04	04/15/2020	03/04/2022

# Azure Functions – Binding Expressions

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/az-204](http://www.exampro.co/az-204)

```
{  
  "bindings": [  
    {  
      "name": "order",  
      "type": "queueTrigger",  
      "direction": "in",  
      "queueName": "%input_queue_name%",  
      "connection": "MY_STORAGE_ACCT_APP_SETTING"  
    }  
  ]  
}
```

## App Settings

When you want to change configuration based on the environment  
Uses percentage signs (%) instead of curly braces

```
{  
  "bindings": [  
    {  
      "name": "image",  
      "type": "blobTrigger",  
      "path": "sample-images/{filename}",  
      "direction": "in",  
      "connection": "MyStorageConnection"  
    },  
    ...  
  ]  
}
```

## Trigger filename

Can be used to change the path of a file name  
Works for both in and out directions

# Azure Functions – Binding Expressions

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/az-204](http://www.exampro.co/az-204)

```
{  
  "bindings": [  
    {  
      "name": "myQueueItem",  
      "type": "queueTrigger",  
      "queueName": "myqueue-items",  
      "connection": "MyStorageConnection",  
    },  
    {  
      "name": "myInputBlob",  
      "type": "blob",  
      "path": "samples-workitems/{queueTrigger}",  
      "direction": "in",  
      "connection": "MyStorageConnection"  
    }  
  ]  
}
```

## Trigger metadata

many triggers provide additional metadata values

These values can be used as input parameters in C# and F# or properties on the context.bindings object in JavaScript.

Eg Azure Queue storage trigger supports the following properties

- QueueTrigger
- DequeueCount
- ExpirationTime
- Id
- InsertionTime
- NextVisibleTime
- PopReceipt

# Azure Functions – Templates

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/az-204](http://www.exampro.co/az-204)



Azure provides function **templates** to get you started with common function scenarios:

**HTTP** triggered by an HTTP request and returns HTTP

**Timer** triggered based on a schedule

**Blob Storage** triggered when files **are uploaded/updated** in a Blob storage container.

**Cosmos DB** triggered when processing a new/modified Cosmos DB documents.

**Queue Storage** triggered by **Azure Storage queue messages**.

**Event Grid** triggered by event from Event Grid

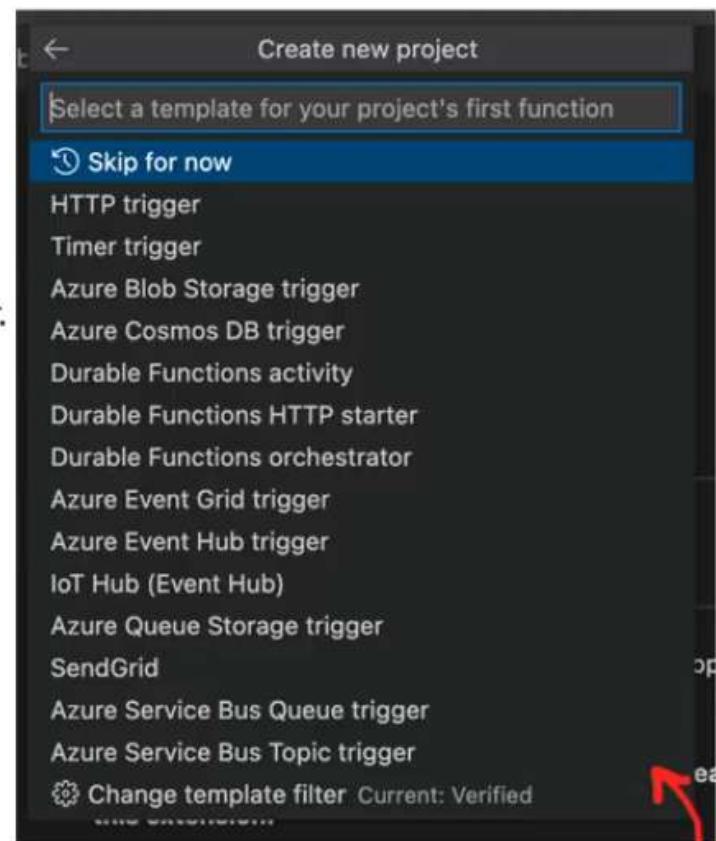
- Many Azure services can trigger a function through Event Grid
- Event Grid is a serverless event bus integrated with Azure services

**Event Hub** – triggered by Event Hub event. (streaming)

**Service Bus Queue** – triggered by a message in a Bus Queue (messaging)

**Service Bus Topics** – triggered by an event from Bus Topic (pub/sub)

**SendGrid** — triggered by an email event in third-party service SendGrid



You choose the function template during project creation **only in Visual Studio Code**

# Azure Functions – Functions Configuration

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/az-204](http://www.exampro.co/az-204)

Every function contains a function configuration file called **function.json**

The function.json file defines the function's trigger, bindings, and other configuration settings.

**type** Name of binding

**direction** Indicates whether the binding is for receiving data into the function or sending data from the function

**name** used for the bound data in the function

- For C#, this is an argument name;
- for JavaScript, it's the key in a key/value list.

```
{  
    "disabled":false,  
    "bindings": [  
        // ... bindings here  
        {  
            "type": "bindingType",  
            "direction": "in",  
            "name": "myParamName",  
            // ... more depending on binding  
        }  
    ]  
}
```

# Azure Functions – Binding Expressions

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/az-204](http://www.exampro.co/az-204)

```
{  
  "bindings": [  
    {  
      "name": "info",  
      "type": "httpTrigger",  
      "direction": "in",  
      "webHookType": "genericJson"  
    },  
    {  
      "name": "blobContents",  
      "type": "blob",  
      "direction": "in",  
      "path": "strings/{BlobName}",  
      "connection": "AzureWebJobsStorage"  
    },  
    {  
      "name": "res",  
      "type": "http",  
      "direction": "out"  
    }  
  ]  
}
```

## JSON Payloads

When a trigger payload is JSON, you can refer to its properties in configuration for other bindings in the same function and in function code

Eg. webhook function that receives a blob name in  
JSON: {"BlobName": "HelloWorld.txt"}

If some of the properties in your JSON payload are objects with properties, you can refer to those directly by using dot (.) notation

# Azure Functions – Hosts Configuration

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/az-204](http://www.exampro.co/az-204)

Every **function app** has a hosts configuration file named **hosts.json**

This configuration file contains global configurations options for all functions within the function app

The host has lot of configuration options:

- aggregator
- applicationInsights
- blobs
- console
- cosmosDb
- customHandler
- durableTask
- eventHub
- extensions
- extensionBundle
- functions
- functionTimeout
- healthMonitor
- http
- logging
- managedDependency
- queues
- retry
- sendGrid
- serviceBus
- singleton
- version
- watchDirectories
- watchFiles

```
{  
    "extensions": {  
        "http": {  
            "routePrefix": "api",  
            "maxOutstandingRequests": 200,  
            "maxConcurrentRequests": 100,  
            "dynamicThrottlesEnabled": true,  
            "hsts": {  
                "isEnabled": true,  
                "maxAge": "10"  
            },  
            "customHeaders": {  
                "X-Content-Type-Options": "nosniff"  
            }  
        }  
    }  
}
```

# Azure Functions – Binding Expressions

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/az-204](http://www.exampro.co/az-204)

```
{  
  "type": "blob",  
  "name": "blobOutput",  
  "direction": "out",  
  "path": "my-output-container/{rand-guid}.txt"  
}
```

## New GUID

Produces a globally unique identifier

eg. 50710cb5-84b9-4d87-9d83-a03d6976a682.

```
{  
  "type": "blob",  
  "name": "blobOutput",  
  "direction": "out",  
  "path": "my-output-container/{DateTime}.txt"  
}
```

## Current Date and Time

Produces the current date and time

Eg. 2018-02-16T17-59-55Z

# Azure Functions – Local Settings File

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/az-204](http://www.exampro.co/az-204)

The Local Settings File stores app settings and settings used by local development tools  
This file is called **local.settings.json** and is expected to be at the root of project folder



Because the local.settings.json may contain secrets, such as connection strings, you should never store it in a remote repository.

## IsEncrypted

When this setting is set to true, all values are encrypted with a local machine key.

```
{  
  "IsEncrypted": false,  
  "Values": {  
    "FUNCTIONS_WORKER_RUNTIME": "<language worker>",  
    "AzureWebJobsStorage": "<connection-string>",  
    "MyBindingConnection": "<binding-connection-string>",  
    "AzureWebJobs.HttpExample.Disabled": "true"  
  },  
  "Host": {  
    "LocalHttpPort": 7071,  
    "CORS": "*",  
    "CORSCredentials": false  
  },  
  "ConnectionStrings": {  
    "SQLConnectionString": "<sqlclient-connection-string>"  
  }  
}
```

## Values

Collection of application settings used when a project is running.

## Host

customize the Functions host process when you run projects locally.

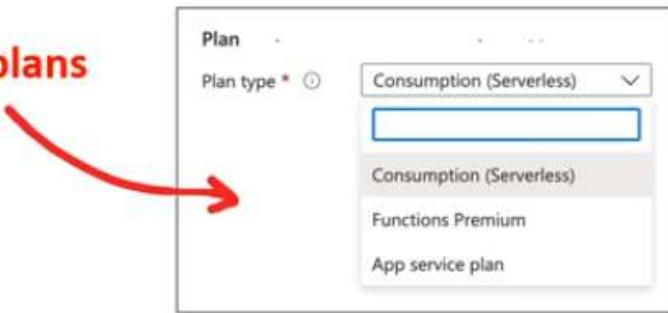
## ConnectionStrings

used only by frameworks that typically get connection strings from the ConnectionStrings

# Azure Functions – Plan Services

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/az-204](http://www.exampro.co/az-204)

Azure Functions has **3 available of plans**



## Consumption Plan (Serverless) **Cold-Starts**

- You only pay for the time your code or application is running.
- Billing is based on the number of executions, the duration of each execution, and the amount of memory used.
- Just pay while you have functions running and scale-out automatically, even through long loading times.



## Premium Plan (Functions Premium) **Pre-Warmed**

- The user has designated a set of pre-warmed cases, which are already online and ready to react instantly.
- Azure provides any additional computing services that are required when your function is running.
- You pay for the constantly pre-warmed instances including any additional instances needed to scale the Azure app in/out.
- Azure Functions host instances are added and removed based on the number of incoming events.



## Dedicated Plan (App service plan) **VM sharing**

- When you use App Service for other apps, your functions will run on the same plan (VMs) at no extra cost.
- You may scale it out manually by adding more VM instances for an App Service plan.
- You may also have autoscale enabled.
- Optimal when you have existing, underutilized VMs, which also operate other instances of the App Service.

# Azure Functions – Custom Handlers

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/az-204](http://www.exampro.co/az-204)

**Custom handlers** are lightweight web servers that receive events from the Functions host.

A custom handler can be written in any language that supports **HTTP primitives**.

Custom handlers are ideal for situations in which you need to:

- You want to use a language not supported by an existing Azure runtime:, eg. Go, Rust or Ruby
- You need a runtime environment configured for specific technology: Deno, Ruby on Rails

```
{  
  "version": "2.0",  
  "customHandler": {  
    "description": {  
      "defaultExecutablePath": "handler.exe"  
    },  
    "enableForwardingHttpRequest": true  
  }  
}
```

Within the functions.json there is a json property called **customHandler** where you define the executable path (the program that will run the function)



With custom handlers, you can use triggers and **input and output** bindings via **extension bundles**.

# Azure Functions – Triggers and Bindings

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/az-204](http://www.exampro.co/az-204)

**Triggers and bindings** lets you avoid hardcoding access to other services and abstracting away boilerplate code keeping your functions lean.

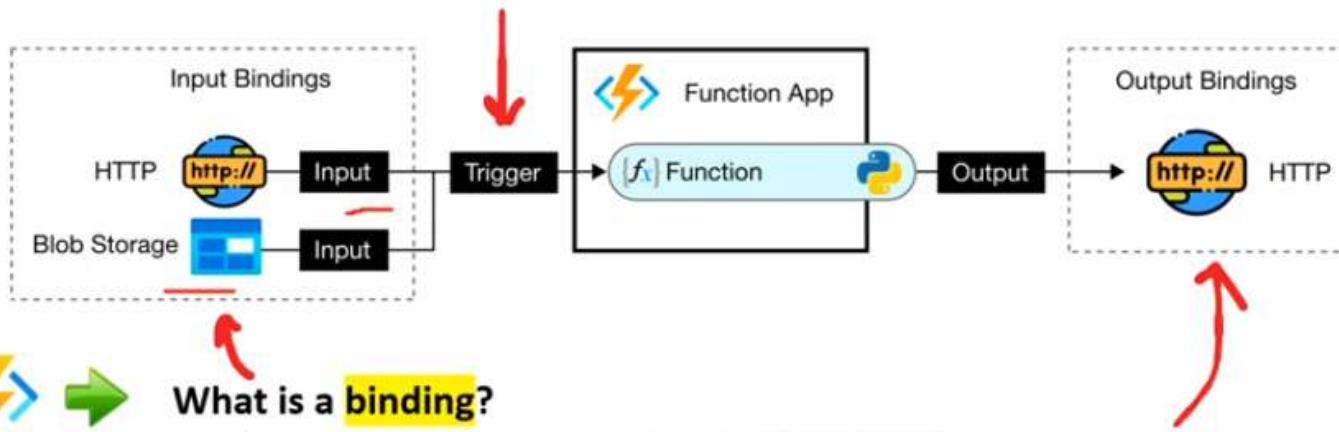


## What is a trigger?

A trigger is a specific type of event which causes the function to **run**.

It defines how a function is **invoked** and a function must only have **one** trigger.

Triggers can have associated **data** which is often provided as the payload of the function.



## What is a binding?

Bindings define if your function is **connected** to another service

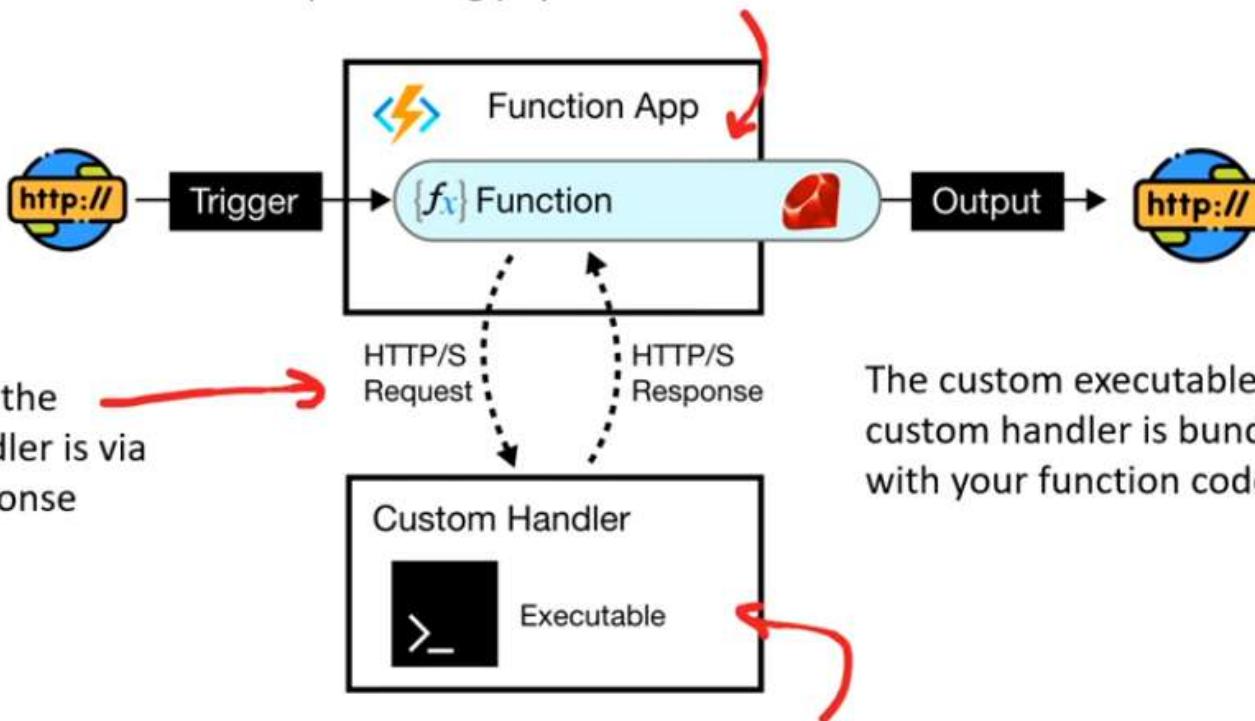
The data from bindings is provided to the function as parameters.

Bindings are optional, and a function can have multiple input and output bindings

# Azure Functions – Custom Handlers

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/az-204](http://www.exampro.co/az-204)

The function will pass along payload to the custom handler



Communication between the function and custom handler is via HTTP/S Request and Response

The custom executable for the custom handler is bundled along with your function code

The custom handler runs on “lightweight web-server”.

- The underlying compute will likely vary based on what service plan is chosen.



# Azure Functions – Supported Bindings

Cheat sheets, Practice Exams and Flash cards [www.exampro.co/az-204](http://www.exampro.co/az-204)



Blob Storage



Azure Cosmos DB



Azure SQL



Dapr



Event Grid



Event Hub



HTTP & webhooks



IoT Hub

Event data emitted from hardware devices



Kafka



Mobile Apps



Notifications Hub



Queue Storage



RabbitMQ



SendGrid



Service Bus



SignalR

Open-source .NET library to send asynchronous notifications to client-side web applications.



Table storage

NoSQL Key/Value Store



Timer

A triggered based on a schedule using a regex expression



Twilio

Cloud Platform for voice and text messaging systems

# Azure Functions – Custom Handlers

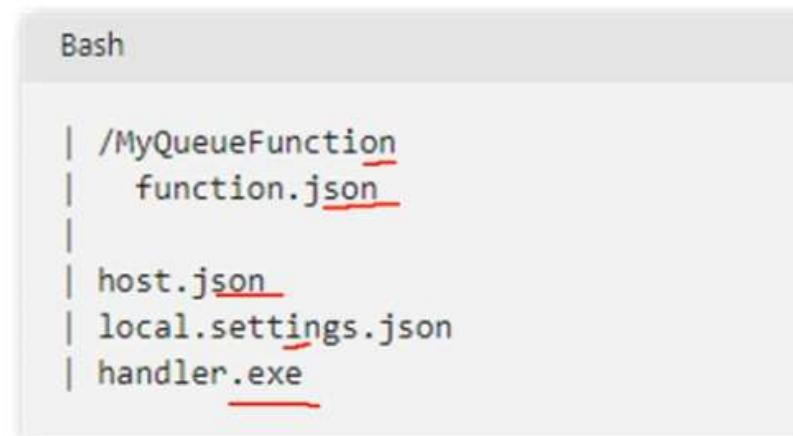
Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/az-204](http://www.exampro.co/az-204)

## Application structure

To implement a custom handler, your application must have the following:

- A ***host.json*** file at the root of your app
- A ***local.settings.json*** file at the root of your app
- A ***function.json*** file for each function (inside a folder that matches the function name)
- A **command, script, or executable**, which runs a web server

The following diagram shows how these files look on the file system for a function named "MyQueueFunction" and a custom handler executable named *handler.exe*.





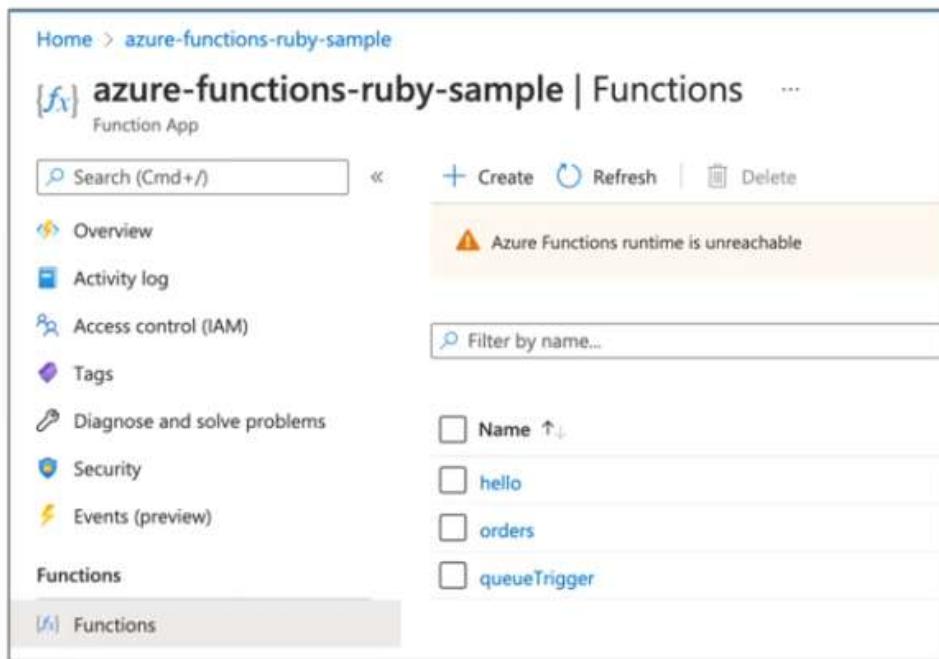
# Azure Functions – Supported Bindings

Cheat sheets, Practice Exams and Flash cards [www.exampro.co/az-204](http://www.exampro.co/az-204)

Type	1.x	2.x and higher <sup>1</sup>	Trigger	Input	Output
Blob storage	✓	✓	✓	✓	✓
Azure Cosmos DB	✓	✓	✓	✓	✓
Azure SQL (preview)		✓	✗	✓	✓
Dapr io <sup>3</sup>		✓	✓	✓	✓
Event Grid	✓	✓	✓		✓
Event Hubs	✓	✓	✓		✓
HTTP & webhooks	✓	✓	✓		✓
IoT Hub	✓	✓	✓		✓
Kafka io <sup>2</sup>		✓	✓		✓
Mobile Apps	✓		✗	✓	✓
Notification Hubs	✓		✗		✓
Queue storage	✓	✓	✓		✓
RabbitMQ <sup>2</sup>		✓	✓		✓
SendGrid	✓	✓			✓
Service Bus	✓	✓	✓		✓
SignalR		✓	✓	✓	✓
Table storage	✓	✓	✗	✓	✓
Timer	✓	✓	✓		
Twilio	✓	✓	✗		

# Azure Functions – Unreachable

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/az-204](http://www.exampro.co/az-204)



Home > azure-functions-ruby-sample

**azure-functions-ruby-sample | Functions** Function App

Search (Cmd+ /) Create Refresh Delete

Azure Functions runtime is unreachable

Filter by name...

Name
hello
orders
queueTrigger

This issue occurs when the Functions runtime can't start. The most common reason for this is that the function app has lost access to its storage account.

## Reasons it might not work

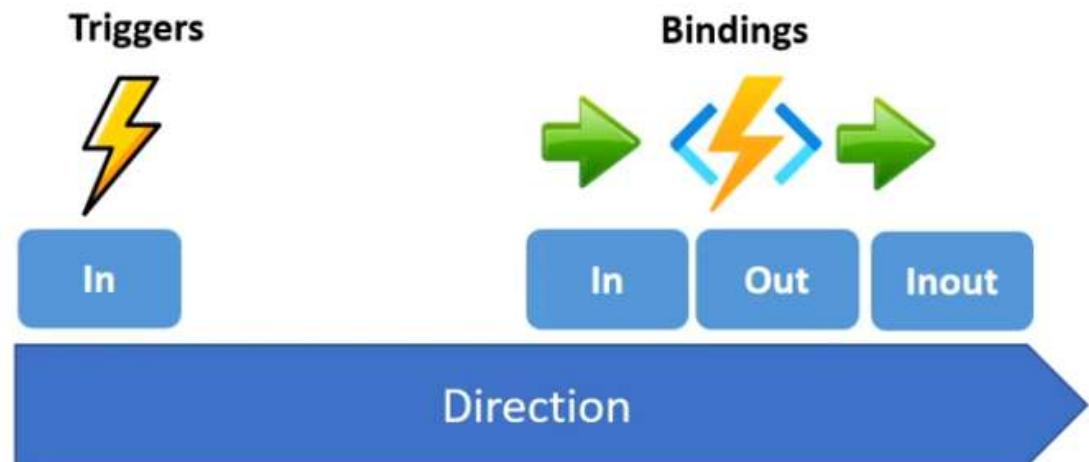
- Storage account was deleted
- Storage account application settings were deleted
- Storage account credentials are invalid
- Storage account is inaccessible
- Daily execution quota is full
- App is behind a firewall

# Azure Functions – Binding Direction

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/az-204](http://www.exampro.co/az-204)

All triggers and bindings have a **direction property** in the **function.json** file:

- The direction of triggers is always **In**
- Input and output bindings use **In** and/or **Out**
- Some bindings support a special direction **Inout**



```
{  
    "scriptFile": "__init__.py",  
    "disabled": false,  
    "bindings": [  
        {  
            "authLevel": "function",  
            "type": "httpTrigger",  
            "direction": "in",  
            "name": "req"  
        },  
        {  
            "type": "http",  
            "direction": "out",  
            "name": "$return"  
        }  
    ]  
}
```

The **Trigger** is defined alongside the Input and Output bindings.

- trigger will have the same as the input type but with “Trigger” appended
  - Eg. blob (input binding) blobTrigger (trigger)

If you use **Inout**, only the **Advanced editor** is available via the **Integrate tab** in the portal.

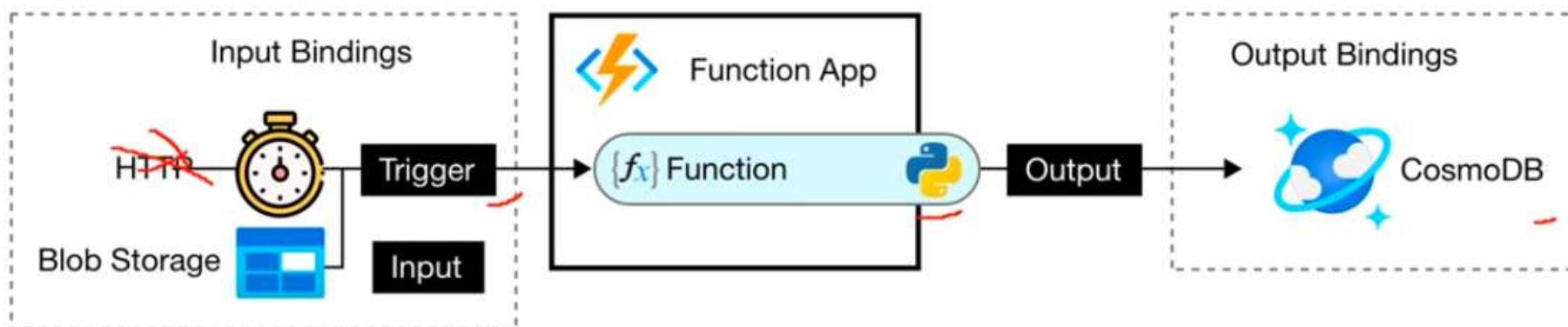
# Azure Functions – Triggers and Bindings Scenarios

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/az-204](http://www.exampro.co/az-204)



## Scenario 1:

Every hour you want to read new log files delivered by your application, and you need to transform the data to be ingested in your No/SQL database that resides in CosmoDB



You'll use the **trigger** type **Timer** because it's a scheduled job that will run at a specific time.

The bindings will be **In** for blob storage and **Out** for cosmos DB because whenever a function runs, it'll be on a **timer**, and when it executes, it'll read data from blob storage, process the data from blob storage, and then **write** some data into **cosmos DB**.

# Azure Functions Core Tools

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/az-204](http://www.exampro.co/az-204)

## Top-Level Commands:

- func **init** — creates a new Functions project in a specific language
- func **logs** — gets logs for functions running in a Kubernetes cluster.
- func **new** — creates a new function in the current project based on a template
- func **run** — Enables you to invoke a function directly, which is similar to running a function using the Test tab in the Azure portal (This is only available in version 1.x only)
- func **start** — Starts the local runtime host and loads the function project in the current folder.
- func **deploy** — replaced with func **kubernetes deploy**

## Command Groups that contain their own set of subcommands

- func **azure** — working with Azure resources, including publishing.
- func **durable** — working with Durable Functions.
- func **extensions** —for installing and managing extensions.
- func **kubernetes** —for working with Kubernetes and Azure Functions.
- func **settings** —for managing environment settings for the local Functions host.
- func **templates** —for listing available function templates.

# Azure Functions – Durable Functions

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/az-204](http://www.exampro.co/az-204)

**Durable Functions** is a serverless compute extension of Azure Functions **that allows you to write stateful functions.**

The extension introduces **two types** of functions:

1. Orchestrator function — define stateful workflows (implicitly representing state via control flow.)
2. Entity function — manage the state of an entity (explicitly representing state)



- They define **workflows** in code. No JSON schemas or designers are needed.
- They can call other functions **synchronously** and **asynchronously**.
  - The output from the called functions can be saved to local variables.
- They **automatically checkpoint** their progress whenever the function awaits.
  - Local state is never lost if the process recycles or the VM reboots.

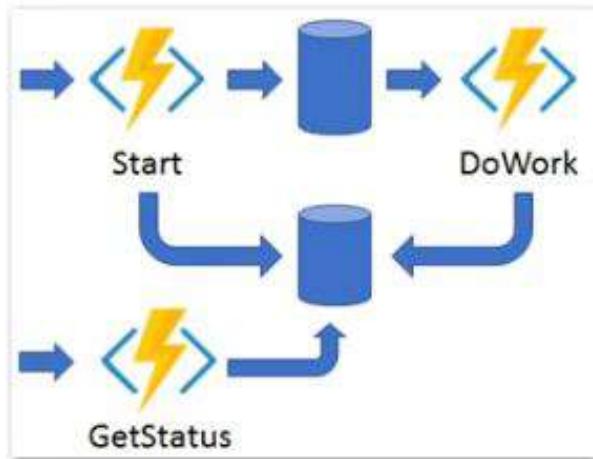
# Patterns of Durable Function

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/az-204](http://www.exampro.co/az-204)

The **Async HTTP API** pattern addresses the problem of **coordinating** the state of long-running operations with **external clients**.

A common way to implement this pattern is to have an **HTTP call trigger** the long-running action, then **redirect** the client to a status **endpoint** that they can **poll** to learn when the operation is complete.

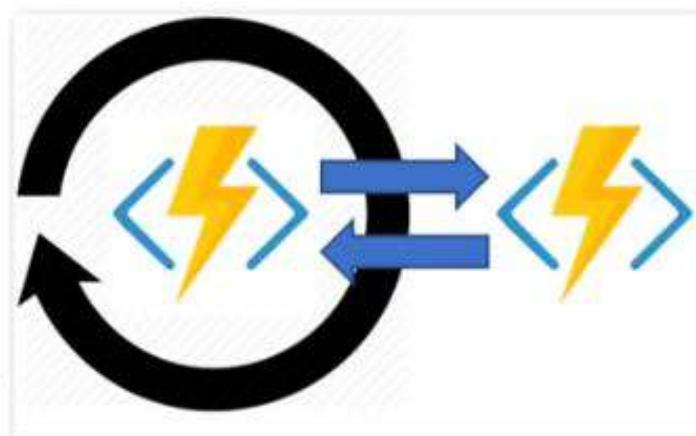
**Durable Functions** provide built-in **APIs** that simplify the code we write for interacting with **long-running function** executions.



The **Monitor** pattern refers to a **flexible recurring process** in a workflow such as polling until certain **conditions** are met.

A simple scenario, such as a periodic cleanup job, can be addressed with a regular **timer-trigger**, but its interval is **fixed**, making managing instance lifetimes difficult.

Durable Functions enables flexible recurrence intervals, task lifetime management, and the ability to create **multiple monitor** processes from a single **orchestration**.



# Patterns of Durable Function

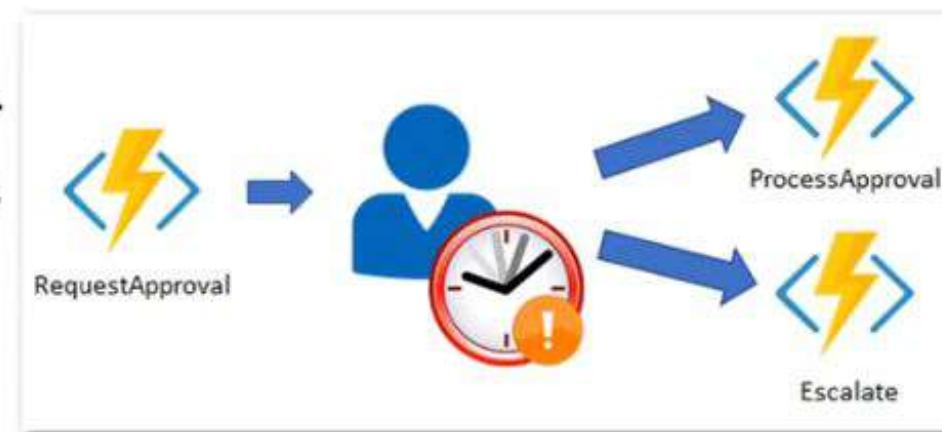
Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/az-204](http://www.exampro.co/az-204)

## Human Interaction

Many automated processes involve some form of **human interaction**.

Humans are not always as available and responsive as cloud services, which makes involving humans in an automated process **tricky**.

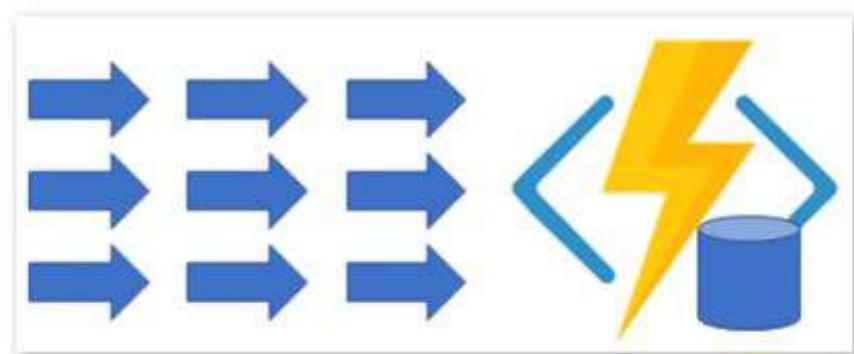
**Automated processes** must allow for this, and they often do so by using **timeouts** and **compensation logic**.



The **Aggregator (stateful entities)** pattern is about aggregating event data over a period of time into a single, addressable **entity**.

The data being aggregated may come from multiple sources, be delivered in batches, or may be scattered over long-periods of time.

The aggregator may need to **act** on event data as it arrives external clients may need to query the aggregated data.



# Azure Functions on Kubernetes

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/az-204](http://www.exampro.co/az-204)

You can deploy any Azure function app to a Kubernetes cluster running **KEDA**



**Kubernetes Event-driven Autoscaling (KEDA)** allows you to setup autoscaling based on events from various cloud-native and third-party services

```
func kubernetes deploy \
--name <name-of-function-deployment> \
--registry <container-registry-username>
```

- Kubernetes-based Functions provides the Functions runtime in a Docker container with event-driven scaling through KEDA.
- KEDA can scale in to 0 instances (when no events are occurring) and out to n instances.
- It does this by exposing custom metrics for the Kubernetes autoscaler (Horizontal Pod Autoscaler).
- Using Functions containers with KEDA makes it possible to replicate serverless function capabilities in any Kubernetes cluster.
- These functions can also be deployed using Azure Kubernetes Services (AKS) virtual nodes feature for serverless infrastructure.



# Introduction to Azure VMs

Cheat sheets, Practice Exams and Flash cards [www.exampro.co/az-204](http://www.exampro.co/az-204)

Azure Virtual Machines (VMs) is a highly configurable server.

Virtualization let you run a server **without having to buy and maintain the physical hardware** that runs it



Virtual Machines still require maintenance such as:

- applying OS system patches
- Installing and configuring packages

## Some things you should know:

- The **size** of the virtual machine is determined by the Image
  - The image defines the combination of vCPUs, Memory and Storage Capacity
- The current limit on a per subscription basis is **20 VMs per region**.
- Azure VMs are billed at an **hourly rate**
- A single instance VMs has an availability of 99.9% (when all storage disks are premium)
- Two instances deployed in Availability Set will give you 99.95% availability
- You can attach multiple Managed Disk to your Azure VMs



# Introduction to Azure VMs

Cheat sheets, Practice Exams and Flash cards [www.exampro.co/az-204](http://www.exampro.co/az-204)

When you **launch** an Azure Virtual Machine other networking components will be either created or associated to your Virtual Machine.



**Network Security Group (NSG)** — attached to the NIC, virtual firewall with rules around ports and protocols



**Network Interface (NIC)** — a device that handle ip protocols and network communication



**Virtual Machine instance** — The actual running server

**Public IP Address** — The address that you will use publicly access your VM



**Virtual Network (VNet)** — The network where your VM will reside

Resource	Type	Status	Operation details
MyNewVirtualMachine	Microsoft.Compute/virtualMachines	OK	<a href="#">Operation details</a>
mynewvirtualmachine839	Microsoft.Network/networkInterfaces	Created	<a href="#">Operation details</a>
MyNewVirtualMachine_group-vnet	Microsoft.Network/virtualNetworks	OK	<a href="#">Operation details</a>
MyNewVirtualMachine-nsg	Microsoft.Network/networkSecurityGroups	OK	<a href="#">Operation details</a>
MyNewVirtualMachine-ip	Microsoft.Network/publicIpAddresses	OK	<a href="#">Operation details</a>



# Azure VMs – Operation Systems

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/az-204](http://www.exampro.co/az-204)

## What is an Operation System (OS)?

The OS is the program that manages all other programs in a computer.

The most commonly known operations systems are Windows ,macOS, and Linux



When you launch a Virtual Machine you need to choose an Image which has a specific Operation System.

Microsoft works closely with partners to ensure the images available are updated and optimized for an Azure runtime. Most of these images can be found in the **Azure Marketplace**



- SUSE Linux Enterprise Server
- Red Hat Enterprise Linux
- Ubuntu Server
- Debian
- FreeBSD
- Azure Marketplace - Flatcar Container Linux
- RancherOS
- Bitnami Library for Azure
- Mesosphere DC/OS on Azure
- Docker images
- CloudBees Jenkins Platform



You can **Bring Your Own Linux** by creating a Linux Virtual Hard Disk (VHD)

*(Hyper-V virtual hard disk (VHDX) format isn't supported in Azure, only fixed VHD)*





# Cloud-Init

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/az-204](http://www.exampro.co/az-204)

**Cloud-init** is the industry standard multi-distribution method for cross-platform **cloud instance initialization**. It is supported across all major public cloud providers, provisioning systems for private cloud infrastructure, and bare-metal installations.

## What is Cloud Instance Initialization?

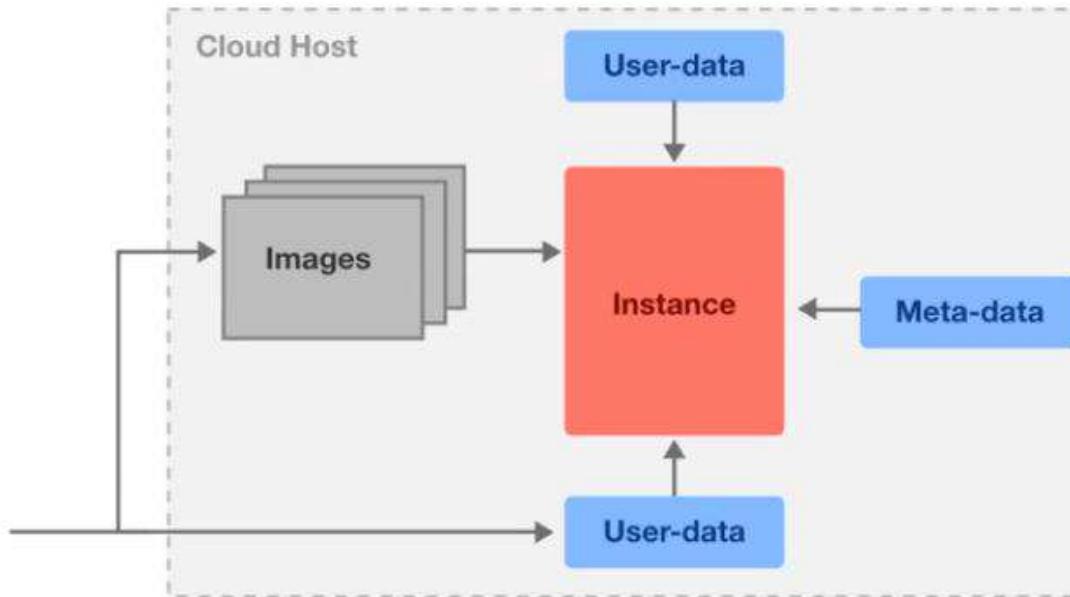
The process of preparing an instance with configuration data for the operation system and runtime environment.

Cloud instances are initialized from a disk image and instance data:

- Meta-data
- **User-data**
- Vendor-data

**User Data** is a script that you want to run when an instance first boots up. eg. Install Apache web-server

Azure Virtual Machines supports for cloud-init across most Linux Distros that support it.





# Azure VMs – Sizes

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/az-204](http://www.exampro.co/az-204)

Azure VMs come in a variety of sizes that are also optimized for specific use cases.

Azure VMs are grouped into:

- **Types** eg. General Purposes, Compute Optimized
- **Sizes** eg . B, Dsv3 (also called Series or SKU Family)

**General Purpose** Balanced CPU-to-Memory ratio. Testing and development, small to medium databases, and low to medium traffic web servers.

SKUs: B, Dsv3, Dv3, Dasv4, Dav4, DSv2, Dv2, Av2, DC, DCv2, Dv4, Dsv4, Ddv4, Ddsv4

**Compute Optimized** High CPU-to-memory ratio. Good for medium traffic web servers, network appliances, batch processes, and app servers.

SKUs: F, Fs, Fsv2

**Memory Optimized** High memory-to-CPU ratio. Great for relational database servers, medium to large caches, and in-memory analytics

SKUs: Esv3, Ev3, Easv4, Eav4, Ev4, Esv4, Edv4, Edsv4, Mv2, M, DSv2, Dv2

**Storage Optimized** High disk throughput and IO ideal for Big Data, SQL, NoSQL databases, data warehousing and large transactional databases.

SKUs: Lsv2

**GPU** Specialized VMs for heavy graphic rendering and video editing, model training and inferencing (ND) with deep learning.

Available with single or multiple GPUs.

SKUs: NC, NCv2, NCv3, NCasT4\_v3 (Preview), ND, NDv2 (Preview), NV, NVv3, NVv4

**High performance compute** Our fastest and most powerful CPU virtual machines with optional high-throughput network interfaces (RDMA).

SKUs: HB, HBv2, HC, H

*There are previous series of Virtual Machines sizes not shown here like Basic A*



## Azure VMs – Sizes

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/az-204](http://www.exampro.co/az-204)

The type of image may limit you to specific VM sizes

Image *	<input type="text" value="Ubuntu Server 18.04 LTS - Gen1"/> 	Size *	<input type="text" value="Standard_B1s - 1 vcpu, 1 GiB memory (CA\$9.72/month)"/> 
<a href="#">Browse all public and private images</a>		<a href="#">Select size</a>	

You can explore sizes and then sort and filter based on a variety of options such as **cost**

Showing 375 VM sizes.		Subscription: Azure subscription 1		Region: East US		Current size: Standard_B1s		Image: Ubuntu Server 18.04 LTS		Learn more about VM sizes	
VM Size ↑↓	Family ↑↓	vCPUs ↑↓	RAM (GiB) ↑↓	Data disks ↑↓	Max IOPS ↑↓	Temp storage (GiB) ↑↓	Premium disk ↑↓	Cost/month ↑↓			
▼ Most used by Azure users ↗											
DS1_v2 ↗	General purpose	1	3.5	4	3200	7		Supported	CA\$68.21		
D2s_v3 ↗	General purpose	2	8	4	3200	16		Supported	CA\$89.70		
B2s ↗	General purpose	2	4	4	1280	8		Supported	CA\$38.87		
B1s ↗	General purpose	1	1	2	320	4		Supported	CA\$9.72		
B2ms ↗	General purpose	2	8	4	1920	16		Supported	CA\$77.74		
B1ms ↗	General purpose	1	2	2	640	4		Supported	CA\$19.34		
B1ls ↗	General purpose	1	0.5	2	160	4		Supported	CA\$4.86		
DS2_v2 ↗	General purpose	2	7	8	6400	14		Supported	CA\$136.42		
B4ms ↗	General purpose	4	16	8	2880	32		Supported	CA\$155.11		
D4s_v3 ↗	General purpose	4	16	8	6400	32		Supported	CA\$179.40		
DS3_v2 ↗	General purpose	8	16	15	12800	28		Supported	CA\$272.70		



# Azure Compute Units

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/az-204](http://www.exampro.co/az-204)

**Azure Compute Unit (ACU)** provides a way of comparing compute (CPU) performance across Azure SKUs.

ACU is currently standardized on a **Small (Standard\_A1)** VM with the value of **100**

All other SKUs then represent approximately how much faster that SKU can run a standard benchmark

SKU Family	ACU / vCPU	vCPU : Core
A1 – A4	100	1:1
D1 - D14	160 - 250	1:1



D1-d14 are **60% to 150%** more performant than the A1-A4



# Monitor VMs via Azure Mobile App

Cheat sheets, Practice Exams and Flash cards [www.exampro.co/az-204](http://www.exampro.co/az-204)

You can install the **Azure Mobile App**, and you can monitor your VMs on the go.

The image shows three screenshots illustrating the Azure mobile app for monitoring VMs:

- Share to mobile screen:** Shows a QR code for "MyWindowsVM". A red arrow points from this screen to the first part of the mobile app interface.
- Mobile app - MyVM screen:** Displays the status of "MyVM" as "Running". It includes an "Activity log" section with 1 info item, a "Metrics" chart for CPU (average), and "States" sections for Power state (Running) and Provisioning state (Succeeded). A red arrow points from the Share to mobile screen to this interface.
- Mobile app - Home screen:** Shows the "Azure services" navigation bar with icons for Virtual machines, Web Apps, SQL databases, and Application Insights. It displays "Latest alerts" (none in the last 24 hours) and "Recent resources". A red arrow points from the MyVM screen to this interface.



# Hyper-V and Generation 1 vs 2

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/az-204](http://www.exampro.co/az-204)



Hyper-V is Microsoft's **hardware virtualization product**.

It lets you **create and run a software version of a computer**, called a *virtual machine*

Each virtual machine acts like a complete computer, running an operating system and programs.



Hyper-V is just like Virtual Box

There are two generations of Hyper-V VMs:

**Generation 1** - support most guest operating systems

**Generation 2** - support most 64-bit versions of Windows and more current versions of Linux and FreeBSD operating systems

Azure has Generation 1 and Generation 2 VMs which are similar **but not exactly the same** as Hyper-V Generations

The most important difference between Azure Gen 1 and Gen 2:

Gen 1

- **BIOS-based** architecture

Gen 2

- **UEFI-based** boot architecture (improved boot and installation times)
- Secure Boot verifies the boot loader is signed by a trusted authority
- Larger boot volume up to 64 TB

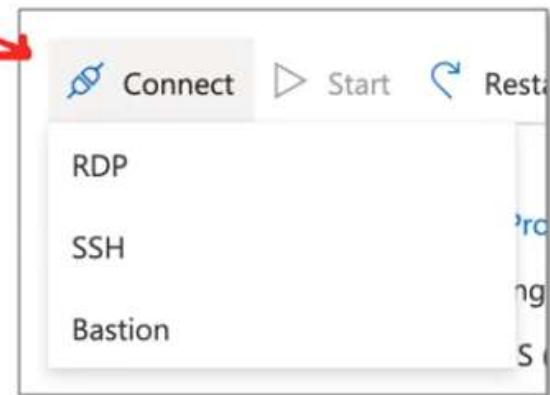
**Hyper-V VMs** are packaged into Virtual Hard Disk formats: VHD or VHDX files



# SSH, RDP and Bastions

Cheat sheets, Practice Exams and Flash cards [www.exampro.co/az-204](http://www.exampro.co/az-204)

There are **3 ways to connect** to your **Virtual Machines**



**Secure Shell (SSH)** is a protocol to establish a secure connection between a client and server.

- This is how you can remotely connect to your Azure VM via terminal
- SSH happens on Port 22 via TCP
- RSA Key Pairs are commonly used to authorize access

**Remote Desktop Protocol (RDP)** is a proprietary protocol developed by Microsoft which provides a user with a graphical interface to connect to another computer over a network connection.

- This is how you can remotely connect to Windows Server via Visual Desktop
- RDP happens on Port 3389 via TCP and UDP

## Bastion

Azure Bastion is a service you deploy that lets you connect to a virtual machine using your browser and the Azure portal. It provides **secure** and seamless RDP/SSH connectivity to your virtual machines directly from the Azure portal over TLS. A bastion is a hardened instance that is monitored. Users connect to this VM which then establishes a connection to the target instance. Sometimes known as jump box since you have one extra security step.



# Secure Shell (SSH)

Cheat sheets, Practice Exams and Flash cards [www.exampro.co/az-204](http://www.exampro.co/az-204)

It is very common to use **SSH key pairs** as a mean to authenticate to your VMs.

**SSH Key Pairs** is when you **generate out** two keys:

- A Private Key
- A Public Key

The private key should remain on your local system and not be shared with others.

The public key is stored on VM.

**When you go to SSH** you provide your private key and its matched against the public key to authenticate you.

```
~> ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/data/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/data/.ssh/id_rsa.
Your public key has been saved in /home/data/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:Up6KjbnEV4Hgfo75YM393QdQsK3Z0aTNBz0DoirrW+c data@yar
The key's randomart image is:
+---[RSA 2048]---+
|   .     .oo..|
|   . .   .o.X.|
|   . o.  ..+ B|
|   . o.o  .+ ..|
|   ..o.S  o..|
|   . %o=      .|
|   @.B...    .|
|   o.=. o.  . .|
|   .oo E. . . .|
+---[SHA256]---+
```

```
ssh -i ~/.ssh/id_rsa.pub azureuser@10.111.12.123
```

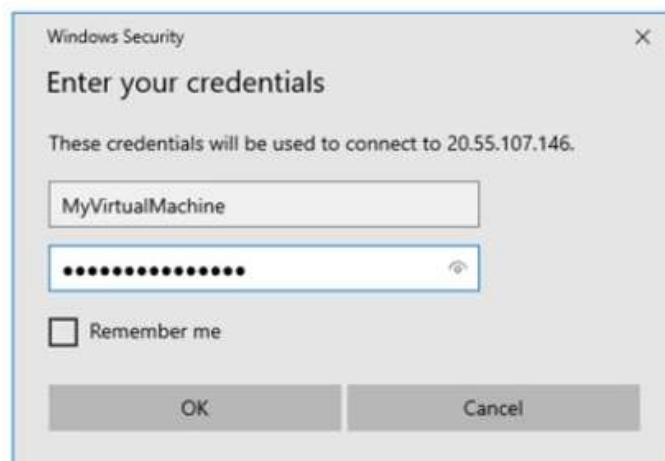


# Remote Desktop Protocol (RDP)

Cheat sheets, Practice Exams and Flash cards [www.exampro.co/az-204](http://www.exampro.co/az-204)

In order to RDP into your Windows Server you'll need to **download the RDP file.**

The Remote Desktop Client is already installed in Windows 10.  
If you are macOS you can download the Microsoft Remote Desktop  
From the Apple Store.



RDP    SSH    BASTION

Connect with RDP

IP address \*

Public IP address (20.55.107.146)

Port number \*

3389

**Download RDP File**



MyWindowsVM.rdp

Once you open the RDP file you will use the  
Username and Password during the creation of your  
VM in the Azure Portal



# Azure Bastion

Cheat sheets, Practice Exams and Flash cards [www.exampro.co/az-204](http://www.exampro.co/az-204)

Azure Bastion is an **intermediate hardened instance** you can use to connect to your target server via SSH or RDP  
It will provision a web-based RDP client or SSH Terminal

**Some devices cannot run an RDP Client** such as Google Chromebook  
and so Azure Bastion is one of the only ways to allow you to do that

When you create an Azure Bastion  
You need to add a Subnet to your VNet  
called **AzureBastionSubnet** with at least a  
size of /27 (32 addresses)

The screenshot shows the Azure portal interface for creating a new virtual network. In the 'Address space' section, a subnet range of '10.2.0.0/27' is selected, highlighted with a red arrow from the text above. In the 'Subnets' section, a new subnet named 'AzureBastionSubnet' is being added, also highlighted with a red arrow. Both the selected address range and the new subnet entry have green checkmarks indicating they are valid.

Address space	Subnets
<input type="checkbox"/> Address range	<input type="checkbox"/> Subnet name
<input type="checkbox"/> 10.1.0.0/16	<input type="checkbox"/> default
<input type="checkbox"/> 10.2.0.0/27	<input type="checkbox"/> AzureBastionSubnet



# Azure Bastion

Cheat sheets, Practice Exams and Flash cards [www.exampro.co/az-204](http://www.exampro.co/az-204)

If you have a Windows Server which requires RDP, and have a Bastion in the same VNet  
You just enter in your Username and Password as you normally would

The image displays two windows side-by-side. On the left is a web-based connection interface titled 'Connect using Azure Bastion'. It shows a summary: 'Using Bastion: MyBastion, Provisioning State: Succeeded'. Below this, it asks for 'Please enter username and password to your virtual machine to connect using Bastion.' A checked checkbox 'Open in new window' is present. Two input fields are shown: 'Username \*' containing 'MyWindowsVM' and 'Password \*' containing a redacted password. A blue 'Connect' button is at the bottom. A red curved arrow points from the 'Connect' button to the right window. The right window is a standard RDP session titled 'MyVM'. It features a large blue circular profile picture placeholder with a white user icon. Below it is the text 'MyWindowsVM' and 'Welcome'.



# Azure Bastion

Cheat sheets, Practice Exams and Flash cards ↗ [www.exampro.co/az-204](http://www.exampro.co/az-204)

If you have a Linux server you can SSH with the Bastion.

You can use SSH Private Key or Password that you set when you created your VM

**Connect using Azure Bastion**

Azure Bastion Service enables you to securely and seamlessly RDP exposing a public IP on the VM, directly from the Azure portal, without software. [Learn more about Azure Bastion](#).

Open in new window

Username \*  

Authentication Type \*   
 Password  SSH Private Key  SSH Private Key from Local File

Local File \*  

Advanced

**Connect**

```
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 5.4.0-1031-azure x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Thu Nov 12 21:27:05 UTC 2020

System load: 0.18          Processes:           116
Usage of /: 4.5% of 28.90GB Users logged in: 0
Memory usage: 20%          IP address for eth0: 10.1.0.6
Swap usage: 0%

0 packages can be updated.
0 updates are security updates.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.
»
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

MyWindowsVM@MyVM:~$
```



# Windows vs Linux Servers

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/az-204](http://www.exampro.co/az-204)

You can launch both **Windows** and **Linux** services on Azure VMs



## Windows

- You need a Windows License (or your Windows account with by unactivated)
- You can bring your own license via Hybrid License
- You set a user name and password
- You have to use a much larger instances to run Windows at least a B2
- It's a full desktop environments



## Linux

- Most versions of Linux require no type of license.
- You set either a username and password or create an ssh-key pair
- You can utilize smaller VM sizes because you're not running a full desktop experience
- Unix and Linux based system traditionally are terminal based environments



# Update Management

Cheat sheets, Practice Exams and Flash cards [www.exampro.co/az-204](http://www.exampro.co/az-204)

**Update Management** allows you to **manage and install operating system updates and patches** for both **Windows** and **Linux** virtual machines that are deployed in Azure, on-premises, or in other cloud providers

When you launch an Azure VM you can go to **Operations** and turn on **Guest + Host Updates**. This will install the Microsoft Monitoring Agent (MMA) that will be used to monitor your instances

Operations

- Bastion
- Auto-shutdown
- Backup
- Disaster recovery
- Guest + host updates

**Guest OS updates**  
Update management  
Update management in Azure Automation machines. You can quickly assess the for your servers. [Learn more](#)

[Go to Update management](#)



**Azure Automations** is the underlying service that is installed the agent.

- Update Management will perform a scan for update compliance
- A compliance scan is by default, **performed every 12 hours** on a **Windows** and **every 3 hours** on a **Linux**
- It can take between **30 minutes and 6 hours** for the dashboard to display updated data from managed computers.



In Azure Automation, you can enable the Update Management, Change Tracking and Inventory, and Start/Stop VMs during off-hours features for your servers and virtual machines. These features have a dependency on a **Log Analytics** workspace, and therefore require linking the workspace with an Automation account.

# Virtual Machines *CheatSheet*

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/az-204](http://www.exampro.co/az-204)



Azure Virtual Machines (VMs) allows you to create Linux and Windows virtual machines

The size of the virtual machine is determined by the Image

- The image defines the combination of vCPUs, Memory and Storage Capacity

The current limit on a per subscription basis is **20 VMs per region**.

Azure VMs are billed at an **hourly rate**

A single instance VMs has an availability of 99.9% (when all storage disks are premium)

Two instances deployed in Availability Set will give you 99.95% availability

You can attach multiple Managed Disk to your Azure VMs

When you **launch** an Azure Virtual Machine other networking components will be either created or associated to your Virtual Machine.

- Network Security Group (NSG), Network Interface (NIC), Public IP Address, VNet

You can **Bring Your Own Linux** by creating a Linux Virtual Hard Disk (VHD)

Azure VMs come in a variety of sizes that are also optimized for specific use cases.

- **General Purpose , Compute Optimized, Memory Optimized, Storage Optimized, GPU, High performance compute**

Azure Compute Unit (ACU) provides a way of comparing compute (CPU) performance across Azure SKUs.

ACU is currently standardized on a **Small (Standard\_A1)** VM with the value of **100**

All other SKUs then represent approximately how much faster that SKU can run a standard benchmark

You can install the **Azure Mobile App**, and you can monitor your VMs on the go.

Hyper-V is Microsoft's **hardware virtualization product**.

- It lets you **create and run a software version of a computer**, called a *virtual machine*

There are two generations of Hyper-V VMs:

**Generation 1** - support most guest operating systems

**Generation 2** - support most 64-bit versions of Windows and more current versions of Linux and FreeBSD operating systems

Hyper-V VMs are packaged into Virtual Hard Disk formats: VHD or VHDX files

# Virtual Machines *CheatSheet*



Cheat sheets, Practice Exams and Flash cards [www.exampro.co/az-204](http://www.exampro.co/az-204)

There are **3 ways to connect** to your **Virtual Machines**

**Sure Shell (SSH)** to connect via a terminal or SSH client eg. PuTTY

- SSH happens on **Port 22** via TCP
- RSA Key Pairs are commonly used to authorize access

**Remote Desktop Protocol (RDP)** a graphical interface to connect to another computer over a network connection

- This is how you can remotely connect to Windows Server via Virtual Desktop
- RDP happens on **Port 3389** via TCP and UDP

**Azure Bastion** a service you deploy that lets you connect to a virtual machine using your browser and the Azure portal

- Supports both SSH or RDP, useful when you only have browser like a chromebook, or do not have permission to configure or install software

**Update Management** allows you to **manage and install operating system updates and patches** for both **Windows** and **Linux** virtual machines that are deployed in Azure, on-premises, or in other cloud providers

- Update Management will perform a scan for update compliance
- A compliance scan is by default, **performed every 12 hours** on a **Windows** and **every 3 hours** on a **Linux**
- It can take between **30 minutes** and **6 hours** for the dashboard to display updated data from managed computers.



# ARM Templates

Cheat sheets, Practice Exams and Flash cards [www.exampro.co/az-204](http://www.exampro.co/az-204)

## What is Infrastructure As Code? (IaC)

the process of **managing and provisioning** computer data centers (eg, Azure) through machine-readable **definition files** (eg. JSON files) rather than physical hardware configuration or interactive configuration tools.

You write a script that will setup cloud services for you.

IaCs can either be:

- **Declarative** — You defined exactly what you want, and you get exactly that
- **Imperative** — You define what you generally want, and the service will guess what you want

**ARM templates** are **JSON files that define azure resources** you want to provision and azure services you want to configure.

With ARM templates you can:

- **ARM templates** are declarative. (you get exactly what you define)
- stand up, tear down or share entire architectures in minutes
- Reduce configuration mistakes
- Know exactly what you have defined for a stack to establish an architecture baseline for compliance



# ARM Templates

Cheat sheets, Practice Exams and Flash cards [www.exampro.co/az-204](http://www.exampro.co/az-204)

With ARM templates you can:

- **ARM templates** are declarative. (you get exactly what you define)
- stand up, tear down or share entire architectures in minutes
- Reduce configuration mistakes
- Establish an architecture baseline for compliance
- **Modularity** Break up your architecture in multiple files and reuse them
- **Extensibility** Add PowerShell and Bash scripts to your templates
- **Testing** You can use the ARM template tool kit (arm-ttk)
- **Preview Changes** Before you create infrastructure via template, see what it will create
- **Built-In Validation** Will only deploy your template if it passes
- **Tracked Deployments** Keep track of changes to architecture over time
- **Policy as Code** Apply Azure policies to ensure you remain compliant
- **Microsoft Blueprints** (establishes relationship between resource and the template)
- **CI/CD integration**
- **Exportable Code** (exporting the current state of a resource groups and resources)
- **Authoring Tools** Visual Studio Code has advanced features for authoring ARM templates



# ARM Template – Skeleton

Cheat sheets, Practice Exams and Flash cards [www.exampro.co/az-204](http://www.exampro.co/az-204)

## Skeleton

The general structure of an ARM template

**\$schema** describes the properties that are available within a template

**contentVersion** the version of the template.

You can provide any value for this element

**apiProfile** Use this value to avoid having to specify API versions for each resource in the template

**parameters** values you can pass along to your template

**variables** you transform parameters or resource properties using function expressions

**functions** User-defined functions available within the template

**resources** the azure resources you'll want to deploy or update

```
{  
  "$schema":  
    "https://schema.management.azure.com/schemas/2019-  
    04-01/deploymentTemplate.json#",  
  "contentVersion": "1.0.0",  
  "apiProfile": "",  
  "parameters": { },  
  "variables": { },  
  "functions": [ ],  
  "resources": [ ],  
  "outputs": { }  
}
```

**outputs** values that are returned after deployment



# ARM Template – Resources

Cheat sheets, Practice Exams and Flash cards [www.exampro.co/az-204](http://www.exampro.co/az-204)

## Resource

An Azure Resource you want to provision

### **type**

Type of the resource

Follows the format of {ResourceProvider}/ResourceType

### **apiVersion**

Version of the REST API to use for the resource

Each resource provider published its own API versions,

### **name**

Name of the resource

### **location**

Most resources have a location property

The region where the resource will be deployed

### **Other Properties**

Other properties we can use to configure the resource

Will vary per resource type

```
{  
    "$schema":  
        "https://schema.management.azure.com/schemas/2019-  
        04-01/deploymentTemplate.json#",  
    "contentVersion": "1.0.0.0",  
    "resources": [  
        {  
            "type": "Microsoft.Storage/storageAccounts",  
            "apiVersion": "2019-04-01",  
            "name": "{provide-unique-name}",  
            "location": "eastus",  
            "sku": {  
                "name": "Standard_LRS"  
            },  
            "kind": "StorageV2",  
            "properties": {  
                "supportsHttpsTrafficOnly": true  
            }  
        }  
    ]  
}
```



# ARM Template – Parameters

Cheat sheets, Practice Exams and Flash cards [www.exampro.co/az-204](http://www.exampro.co/az-204)

## Parameters

Allows you to pass variables to your ARM template

**type** the expected data type of the inputted value

- **string, securestring, int, bool, object, secureObject, and array.**

**defaultValue** if not value is provided it will be set to this value

**allowedValues** an array of allowed values

**minValue** the minimal possible value

**maxLength** the maximum possible value

**minLength** the maximum length of characters or array

**maxLength** the maximum length of characters or array

**description** the description that will be displayed to the in the Azure Portal

**Setting a parameter**

**Accessing a parameter**

```
{  
    "$schema":  
        "https://schema.management.azure.com/schemas/2019-04-01/deploymentTemplate.json#",  
    "contentVersion": "1.0.0.0",  
    "parameters": {  
        "storageName": {  
            "type": "string",  
            "minLength": 5,  
            "maxLength": 20  
        }  
    },  
    "resources": [  
        {  
            "type": "Microsoft.Storage/storageAccounts",  
            "apiVersion": "2019-04-01",  
            "name": "[parameters('storageName')]",  
            "location": "eastus",  
            "sku": {  
                "name": "Standard_LRS"  
            },  
            "kind": "StorageV2",  
            "properties": {  
                "supportsHttpsTrafficOnly": true  
            }  
        }  
    ]  
}
```



# ARM Template – Functions

Cheat sheets, Practice Exams and Flash cards [www.exampro.co/az-204](http://www.exampro.co/az-204)

## Functions

Allows you to apply transformations to your ARM variables

- Template Functions — built-in functions
- Used-Defined Functions — custom functions you create

Functions are called using **parentheses eg. ()**:

```
{ "condition": "[equals(parameters('newOrExisting'), 'new')]" }
```



## Template Functions

- **Array:** array, concat, contains, createArray, empty, first, intersection, last, length, min, max, range, skip, take, union
- **Comparison:** coalesce, equals, less, lessOrquals, greater, greaterOrEqual
- **Date:** dateAdd, utcNow
- **Deployment:** deployment, environment, **parameters**, **variables**
- **Logical:** and, or, if, not, or
- **Numeric:** add, copyIndex, div, float, int, min, max, mod, mul, sub
- **Object:** contains, empty, intersection, json, length, union
- **Resource:** extensionResourceId, ListAccountSas, listKeys, listSecrets, list\*, picZones, providers, reference, resourceGroup, resourceId, subscription, subscriptionResourceId, tenantResourceId
- **String:** base64, base64ToJson, base64ToString, concat, contains, dataUri, DataUriToString, empty, endsWith, first, format, guid, indexOf, last, lastIndexof, length, newGuid, padLeft, replace, skip, split, startsWith, string, substring, take, toLower, toUpper, trim, uniqueString, uri, uriComponent, uriComponentToString



# ARM Template – Variables

Cheat sheets, Practice Exams and Flash cards [www.exampro.co/az-204](http://www.exampro.co/az-204)

## Variables

Template variables are used to simplify your arm templates.

You transform parameters and resource properties using functions  
and then assign them into a reusable variable



```
"variables": {  
    "storageName": "[concat(toLower(parameters('storageNamePrefix')),uniqueString(resourceGroup().id))]"  
},
```

To call a variable you use the **variable()** function



```
"resources": [  
    {  
        "type": "Microsoft.Storage/storageAccounts",  
        "name": "[variables('storageName')]",  
        ...  
    }  
]
```



# ARM Template – Variables

Cheat sheets, Practice Exams and Flash cards [www.exampro.co/az-204](http://www.exampro.co/az-204)

## Nested Variables

You can use json object to have nested variables to scope your variables for multiple use cases.

Scoping/Nesting variables **based on environment**

```
"variables": {  
    "environmentSettings": {  
        "test": {  
            "instanceSize": "Small",  
            "instanceCount": 1  
        },  
        "prod": {  
            "instanceSize": "Large",  
            "instanceCount": 4  
        }  
    }  
},
```

Using params to **choose the environment**

```
"parameters": {  
    "environmentName": {  
        "type": "string",  
        "allowedValues": [  
            "test",  
            "prod"  
        ]  
    },  
},
```

**Referencing nested variables** eg. `Variables()[].property`

```
"[variables('environmentSettings')[parameters('environmentName')].instanceSize]"
```



# ARM Template – Outputs

Cheat sheets, Practice Exams and Flash cards [www.exampro.co/az-204](http://www.exampro.co/az-204)

## Outputs

Returns values from deployed resources, so you can use them programmatically

You specific **the type and value** under outputs

```
"outputs": {  
    "resourceID": {  
        "type": "string",  
        "value": "[resourceId('Microsoft.Network/publicIPAddresses', parameters('publicIPAddresses_name'))]"  
    }  
}
```

You can use the Azure API via **CLI**, PowerShell  
or SDK to fetch outputs

```
az deployment group show \  
  -g <resource-group-name> \  
  -n <deployment-name> \  
  --query properties.outputs.resourceID.value
```

# ARM Templates *CheatSheet*

Exam Pro

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/az-204](http://www.exampro.co/az-204)

**Infrastructure As Code (IaC)** is the process of **managing and provisioning** computer data centers (eg, Azure) through machine-readable **definition files** (eg. JSON files) rather than physical hardware configuration or interactive configuration tools.

IaCs can either be:

- **Declarative** — You defined exactly what you want, and you get exactly that
- **Imperative** — You define what you generally want, and the service will guess what you want

ARM templates are **JSON files that define azure resources** you want to provision and azure services you want to configure.

ARM templates are declarative. (you get exactly what you define)

An ARM template is made of the following JSON structure:

- **\$schema** describes the properties that are available within a template
- **contentVersion** the version of the template. You can provide any value for this element
- **apiProfile** Use this value to avoid having to specify API versions for each resource in the template
- **parameters** values you can pass along to your template
- **variables** you transform parameters or resource properties using function expressions
- **functions** User-defined functions available within the template
- **resources** the azure resources you'll want to deploy or update
  - **type** Type of the resource
  - **apiVersion** Version of the REST API to use for the resource, Each resource provider published its own API versions
  - **name** Name of the resource
  - **Location** Most resources have a location property, The region where the resource will be deployed
  - **Other Properties** Other properties we can use to configure the resource. Will vary per resource type
- **outputs** values that are returned after deployment

# Azure Container Instances (ACI)



**package, deploy, and manage** cloud applications using **containers**  
Fully Managed Docker as a Service



# Introduction to ACI

Cheat sheets, Practice Exams and Flash cards [www.exampro.co/az-204](http://www.exampro.co/az-204)

Azure Container Instances (ACIs) allow you to **launch containers** without the need to worry about configuring or managing the underlying virtual machine

Azure Container Instances is designed for isolate containers:

- simple applications
- task automation
- build jobs

- Containers can be provisioned **within seconds** where VMs can take several minutes
- Containers are **billed per second** where VMs are billed per hour (greater savings)
- Containers have **granular and custom sizing of vCPUs, Memory and GPUs** where VMs sizes are predetermined
- ACI can deploy both **Windows** and **Linux** containers
- You can **persist storage with Azure Files** for your ACI containers
- ACIs are accessed via a fully qualified domain name (FQDN) eg *customlabel.azureregion.azurecontainer.io.*

Azure provides Quickstart images to start launching example applications but you can also **source** containers from:

- Azure Container Registry
- Docker Hub
- Privately Hosted Container Registry

Image source \*

Quickstart images  
 Azure Container Registry  
 Docker Hub or other registry





# Introduction to ACI

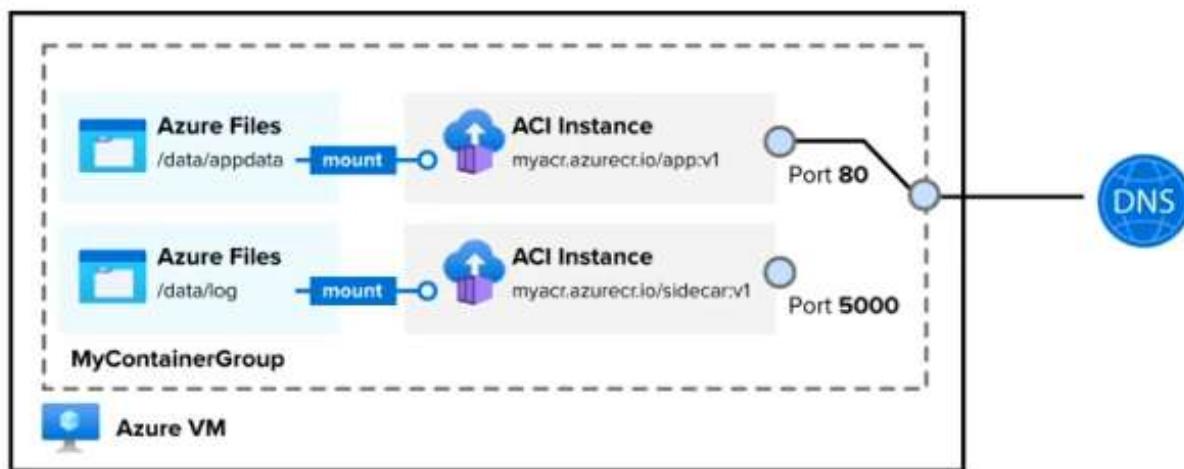
Cheat sheets, Practice Exams and Flash cards [www.exampro.co/az-204](http://www.exampro.co/az-204)

**Container Groups** are collection of containers that get scheduled on the same host machine.

The containers in a container group share:

- lifecycle
- Resources
- local network
- storage volumes

*Container Groups are similar to a Kubernetes pod*



*Multi-container groups currently support only Linux containers.*

There are two ways to deploy a multi-container group:

- **Resource Manager Template (ARM template)** — when you need to deploy additional Azure service resources
- **YAML File** — when your deployment includes only container instances.

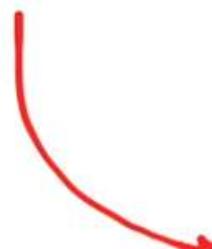


# Container Restart Policies

Cheat sheets, Practice Exams and Flash cards [www.exampro.co/az-204](http://www.exampro.co/az-204)

A container restart policy specifies what a container should do when their process has completed. Azure Container Instances has 3 restart-policy options:

- **Always** (default) Containers are **always restarted**. Suited for long running tasks eg. **web-servers**
- **Never** Containers **run one time only**. Suited for one off tasks. eg. **background jobs**
- **OnFailure** Containers that encounter an error



Basics Networking Advanced Tags Review + create

Configure additional container properties and variables.

Restart policy ⓘ

On failure ^

On failure

Always

Never