

## Azure Functions

1.You are developing an application that executes short lived processes on a schedule. You need to recommend a service that keeps costs and admin activities minimal.

What should you recommend?

- A. Azure Web App
- B. Azure Function**
- C. Azure Virtual Machine
- D. Azure Logic Apps

**Correct**

Azure Functions allows you to run small pieces of code (called “functions”) without worrying about application infrastructure. A function is “triggered” by a specific type of event. Supported triggers include responding to changes in data, responding to messages, running on a schedule, or as the result of an HTTP request.

<https://docs.microsoft.com/en-us/azure/azure-functions/functions-overview>

2.You are developing an application using Azure Function App and Azure Blob container. The images uploaded to the Azure Blob container will be processed by Azure Function App. Images must be processed as quickly as possible once they are uploaded. The solution must minimize latency.

You need to configure the Function App. How should you configure?

- A. Use an App Service plan. Configure the Function App to use an Azure Blob Storage input trigger.
- B. Use a Consumption plan. Configure the Function App to use an Azure Blob Storage trigger.
- C. Use a Consumption plan. Configure the Function App to use a Timer trigger.
- D. Use an App Service plan. Configure the Function App to use an Azure Blob Storage trigger.**

**Correct**

The requirement is to minimize the latency. If your function app is on the Consumption plan, there can be up to a 10-minute delay in processing new blobs if a function app has gone idle. To avoid this latency, you should use an App Service plan with Always On enabled.

<https://docs.microsoft.com/en-us/azure/azure-functions/functions-bindings-storage-blob-trigger?tabs=csharp>

You are developing a server less application on Azure. You created a key vault named KV1 to store and read secrets from an Azure Function.

You need to reference KV1 without making any changes to the application source code.

The identity used to connect to KV1 must be deleted if the Azure function is deleted.

The Azure Function must scale based on the number of incoming requests.

Avoid any cold starts.

Azure Function must be able to connect to a virtual network.

You need to implement the Azure Functions application access to the Azure Key Vault.

Which three actions should you perform in sequence?

- Create the Azure Functions app with a consumption plan type
- Create a user-managed identity for the application.

Create an access policy in Azure Key Vault for the application identity

- Create the Azure Functions app with a premium plan type  
Create a user-managed identity for the application.  
Create an access policy in Azure Key Vault for the application identity
- **Create the Azure Functions app with a premium plan type**  
**Create a system-managed identity for the application.**  
**Create an access policy in Azure Key Vault for the application identity**
- Create the Azure Functions app with an app service plan type  
Create a system -managed identity for the application.  
Create an access policy in Azure Key Vault for the application identity

### Correct

The Azure Functions Premium plan (sometimes referred to as Elastic Premium plan) is a hosting option for function apps. The Premium plan provides features like VNet connectivity, no cold start, and premium hardware.

A managed identity from Azure Active Directory (Azure AD) allows your app to easily access other Azure AD-protected resources such as Azure Key Vault. The identity is managed by the Azure platform and does not require you to provision or rotate any secrets.

Your application can be granted two types of identities:

A system-assigned identity is tied to your application and is deleted if your app is deleted. An app can only have one system-assigned identity.

A user-assigned identity is a standalone Azure resource that can be assigned to your app. An app can have multiple user-assigned identities.

<https://docs.microsoft.com/en-us/azure/app-service/overview-managed-identity?tabs=dotnet>  
<https://docs.microsoft.com/en-us/azure/azure-functions/functions-premium-plan>

You develop an Azure Function app that uses HTTP trigger. The Azure Function app process data stored in an Azure blob storage. The Azure Function app is triggered using an output binding on the blob.

The Function app continues to time out after four minutes. The Function app must process the blob data.

You need to implement a solution so that the Function app does not time out and processes the blob data.

You plan to use the Durable Function async pattern to process the blob data.

Did you achieve the requirement?

- **Yes**
- No

### Correct

The maximum amount of time that an HTTP triggered function can take to respond to a request is 230 seconds. This is because of the default idle timeout of Azure Load Balancer. For longer processing times, consider using the Durable Functions async pattern.

<https://docs.microsoft.com/en-us/azure/azure-functions/functions-scale#timeout>

You develop an Azure Function app that uses HTTP trigger. The Azure Function app process data stored in an Azure blob storage. The Azure Function app is triggered using an output binding on the blob.

The Function app continues to time out after four minutes. The Function app must process the blob data.

You need to implement a solution so that the Function app does not time out and processes the blob data.

You plan to pass the HTTP trigger payload into an Azure Service Bus queue to be processed by a queue trigger function and return an immediate HTTP success response.

Did you achieve the requirement?

- **Yes**
- No

**Correct**

Large, long-running functions can cause unexpected timeout issues. Whenever possible, refactor large functions into smaller function sets that work together and return responses fast. For example, a webhook or HTTP trigger function might require an acknowledgment response within a certain time limit; it's common for webhooks to require an immediate response. You can pass the HTTP trigger payload into a queue to be processed by a queue trigger function. This approach lets you defer the actual work and return an immediate response.

<https://docs.microsoft.com/en-us/azure/azure-functions/functions-best-practices>

You develop an Azure Function app that uses HTTP trigger. The Azure Function app process data stored in an Azure blob storage. The Azure Function app is triggered using an output binding on the blob.

The Function app continues to time out after four minutes. The Function app must process the blob data.

You need to implement a solution so that the Function app does not time out and processes the blob data.

You plan to the app to use an App Service hosting plan and enable the Always On setting.

Did you achieve the requirement?

- Yes
- **No**

**Correct**

Instead use Durable functions or use a queue/ event hub to pass the payload and trigger function from queue.

<https://docs.microsoft.com/en-us/azure/azure-functions/functions-best-practices>

You are developing a HTTP-based API to support a web app.

The primary functionality of the API is to allow customers to check the status of their orders.

You need to use Azure Functions to implement the API.

The API must not allow write operations. However, it must provide public read only operations.

You need to recommend a configuration option for authorization level.

What should you recommend?

- **Anonymous**
- Function
- Admin

## Correct

Anonymous allows public read access.

<https://docs.microsoft.com/en-us/azure/app-service/overview-authentication-authorization>

You plan to develop an application using Azure Functions.

You need to use the Azure Function triggers that have built-in retry support.

Which triggers should you use?

- Azure Blob storage
- **Azure Queue storage**
- **Azure Service Bus (queue/topic)**
- Azure Table Storage
- Azure Cosmos DB
- **Event Hubs**

## Incorrect

Here are the Azure Function triggers that have built-in retry support:

- **Azure Queue storage:** This trigger automatically retries delivery of messages to your function if processing fails. You can configure the retry policy with options like the number of retries, delay between retries, and specific exceptions to trigger retries.
- **Azure Service Bus (queue/topic):** Similar to Queue storage, Service Bus queues and topics offer built-in retries for message delivery. You can define the retry strategy with options like fixed delay or exponential backoff.
- **Event Hubs:** This trigger also provides automatic retries for message delivery in case of processing failures. You can configure the retry behavior with properties like maximum delivery attempts and retry interval.

**Triggers that do not have built-in retries:**

- Azure Blob storage
- Azure Table storage
- Azure Cosmos DB

For these triggers, you would need to implement your own retry logic within your function code using libraries like Polly for handling transient errors.

**In summary, the triggers with built-in retry support for Azure Functions are Azure Queue storage, Azure Service Bus (queue/topic), and Event Hubs.**

<https://docs.microsoft.com/en-us/azure/azure-functions/functions-bindings-errors>

You are developing an application that must execute a sequence of activities in a specific order.

You plan to deploy the application in a serverless compute environment.

Which of the below service should you create to deploy the application?

- Azure Functions
- **Durable Functions**
- Azure App Service

## Correct

Durable Functions is an extension of Azure Functions that lets you write stateful functions in a serverless compute environment. The primary use case for Durable Functions is simplifying complex, stateful coordination requirements in serverless applications. In the function chaining pattern, a sequence of functions executes in a specific order. In this pattern, the output of one

function is applied to the input of another function. You can use Durable Functions to implement the function chaining pattern.

<https://docs.microsoft.com/en-us/azure/azure-functions/durable/durable-functions-overview?tabs=csharp>

You plan to develop an Azure function app to connect to Azure Event Grid.

The resources for the function app must be allocated dynamically. The costs must be based on the number of executions.

How should you configure when you create the function app?

- the Windows operating system and the App Service plan hosting plan
- the Docker container and an App Service plan that uses the B1 pricing tier
- **the Windows operating system and the Consumption plan hosting plan**
- the Docker container and an App Service plan that uses the S1 pricing tier

**Correct**

When you're using the Consumption plan, instances of the Azure Functions host are dynamically added and removed based on the number of incoming events. This serverless plan scales automatically, and you're charged for compute resources only when your functions are running. On a Consumption plan, a function execution times out after a configurable period of time.

<https://docs.microsoft.com/en-us/azure/azure-functions/functions-scale>

You develop a serverless application using several Azure Functions.

You need to change configuration settings in the host.json file.

Which tool should you use?

- **Visual Studio**
- Azure portal
- Azure PowerShell
- Azure CLI

**Correct**

The Functions editor built into the Azure portal lets you update your code and your function.json file directly inline. This is recommended only for small changes or proofs of concept – best practice is to use a local development tool like VS Code.

<https://docs.microsoft.com/en-us/azure/azure-functions/functions-reference#fileupdate>

You are developing an application managed photos. Users upload photos to an API which then stores the photos in an Azure blob storage of type General-purpose V2.

You need to implement a module that processes photos and produce a thumbnail version of the photo. The process must start in less than one minute when photos were uploaded.

To achieve requirement, you create photo processing login in an Azure Function triggered from the blob upload.

Does the solution meet the goal?

- **Yes**
- No

**Correct**

The Blob storage trigger starts a function when a new or updated blob is detected. The blob contents are provided as input to the function.

The Azure Blob storage trigger requires a general-purpose storage account. Storage V2 accounts with hierarchal namespaces are also supported.

<https://docs.microsoft.com/en-us/azure/azure-functions/functions-bindings-storage-blob-trigger?tabs=csharp>

You are developing an application managed photos. Users upload photos to an API which then stores the photos in an Azure blob storage of type General-purpose V2.

You need to implement a module that processes photos and produce a thumbnail version of the photo. The process must start in less than one minute when photos were uploaded.

To achieve requirement, you convert the Azure Storage account to a BlockBlobStorage storage account.

Does the solution meet the goal?

- Yes
- **No**

**Correct**

Instead you can use an Azure Function to create thumbnails.

The Blob storage trigger starts a function when a new or updated blob is detected. The blob contents are provided as input to the function.

The Azure Blob storage trigger requires a general-purpose storage account. Storage V2 accounts with hierarchal namespaces are also supported.

<https://docs.microsoft.com/en-us/azure/azure-functions/functions-bindings-storage-blob-trigger?tabs=csharp>

You are developing an application managed photos. Users upload photos to an API which then stores the photos in an Azure blob storage of type General-purpose V2.

You need to implement a module that processes photos and produce a thumbnail version of the photo. The process must start in less than one minute when photos were uploaded.

To achieve requirement, you trigger the photo processing from Blob storage events

Does the solution meet the goal?

- **Yes**
- No

**Correct**

Azure Storage events allow applications to react to events, such as the creation and deletion of blobs. It does so without the need for complicated code or expensive and inefficient polling services. The best part is you only pay for what you use.

Blob storage events are pushed using Azure Event Grid to subscribers such as Azure Functions, Azure Logic Apps, or even to your own http listener. Event Grid provides reliable event delivery to your applications through rich retry policies and dead-lettering.

<https://docs.microsoft.com/en-us/azure/storage/blobs/storage-blob-event-overview>

You have the source code for an application in GitHub repository under master branch. You need to create an Azure Function and configure it to deploy code from a GitHub repository. Which of the below Azure CLI command should you use?

- `az functionapp create --name $functionAppName \`  
`--storage-account $storageName \`  
`--consumption-plan-location $region \`  
`--resource-group myResourceGroup \`  
`--deployment-source-url $gitrepo \`  
`--deployment-source-branch root \`  
`--functions-version 2`
- **`az functionapp create --name $functionAppName \`**  
**`--storage-account $storageName \`**  
**`--consumption-plan-location $region \`**  
**`--resource-group myResourceGroup \`**  
**`--deployment-source-url $gitrepo \`**  
**`--deployment-source-branch master \`**  
**`--functions-version 2`**
- `az functionapp create --name $functionAppName \`  
`--storage-account $storageName \`  
`--consumption-plan-location $region \`  
`--resource-group myResourceGroup \`  
`--deployment-source $gitrepo \`  
`--deployment-source-branch master \`  
`--functions-version 2`

**Correct**

# Create a function app with source files deployed from the specified GitHub repo.

```
az functionapp create \
--name $functionAppName \
--storage-account $storageName \
--consumption-plan-location $region \
--resource-group myResourceGroup \
--deployment-source-url $gitrepo \
--deployment-source-branch master \
--functions-version 2
```

<https://docs.microsoft.com/en-us/azure/azure-functions/scripts/functions-cli-create-function-app-github-continuous>

You are developing an Azure Function which process the messages in a queue. The outcome of the process must be stored in an Azure Cosmos DB.

Select the following code snippet that will suffice the requirement.

- `[FunctionName("QueueToDocDB")]`  
`public static void Run(`

```
[CosmosDBTrigger("myqueue-items", Connection = "AzureWebJobsStorage")] string
myQueueItem,
[QueueTrigger("ToDoList", "Items", Id = "id", ConnectionStringSetting = "myCosmosDB")] out
dynamic document)
{
...
}
```

- [FunctionName("QueueToDocDB")]  
public static void Run(  
[QueueTrigger("myqueue-items", Connection = "AzureWebJobsStorage")] string  
myQueueItem,  
[CosmosDB("ToDoList", "Items", Id = "id", ConnectionStringSetting = "myCosmosDB")] in  
dynamic document)  
{  
...  
}
- **[FunctionName("QueueToDocDB")]**  
**public static void Run(**  
**[QueueTrigger("myqueue-items", Connection = "AzureWebJobsStorage")] string**  
**myQueueItem,**  
**[CosmosDB("ToDoList", "Items", Id = "id", ConnectionStringSetting =**  
**"myCosmosDB")] out dynamic document)**  
**{**  
**...**  
**}**

## Correct

The QueueTrigger lets you trigger the Function when a message arrives in the queue.

The Azure Cosmos DB output binding lets you write a new document to an Azure Cosmos DB database using the SQL API.

The direction must be set to out.

<https://docs.microsoft.com/en-us/azure/azure-functions/functions-bindings-cosmosdb-v2-output?tabs=csharp#attributes-and-annotations>

You plan to develop an Azure Function app to write the functions by using the Rust language. Which Azure Function app features should you use?

- Runtime
- Hosting Plan
- Extension bundle
- **Custom handler**
- Trigger

## Correct



Custom handlers are lightweight web servers that receive events from the Functions host. Any language that supports HTTP primitives can implement a custom handler.

Custom handlers are best suited for situations where you want to:

Implement a function app in a language that's not currently offered out-of-the box, such as Go or Rust.

Implement a function app in a runtime that's not currently featured by default, such as Deno.

<https://docs.microsoft.com/en-us/azure/azure-functions/functions-custom-handlers>

You plan to develop an Azure Function app that declaratively connect to an Azure Blob Storage account.

Which Azure Function app features should you use?

- Runtime
- Hosting Plan
- **Extension bundle**
- Custom handler
- Trigger

**Incorrect**

Extension bundles is a way to add a compatible set of binding extensions to your function app. You enable extension bundles in the app's host.json file. Standard triggers along with input and output bindings are available by referencing extension bundles in your host.json file.

<https://docs.microsoft.com/en-us/azure/azure-functions/functions-bindings-register>