100 XP

Compare Azure Functions hosting options

3 minutes

When you create a function app in Azure, you must choose a hosting plan for your app. There are three basic hosting plans available for Azure Functions: Consumption plan, Premium plan, and App service plan (Dedicated). All hosting plans are generally available (GA) on both Linux and Windows virtual machines.

The hosting plan you choose dictates the following behaviors:

- How your function app is scaled.
- The resources available to each function app instance.
- Support for advanced functionality, such as Azure Virtual Network connectivity.

Following is a summary of the benefits of the three main hosting plans for Functions:

Expand table

Plan	Benefits
Consumption plan	This is the default hosting plan. It scales automatically and you only pay for compute resources when your functions are running. Instances of the Functions host are dynamically added and removed based on the number of incoming events.
Premium plan	Automatically scales based on demand using pre-warmed workers, which run applications with no delay after being idle, runs on more powerful instances, and connects to virtual networks.
Dedicated plan	Run your functions within an App Service plan at regular App Service plan rates. Best for long-running scenarios where Durable Functions can't be used.

There are two other hosting options, which provide the highest amount of control and isolation in which to run your function apps.

Expand table

Hosting option	Details
ASE	App Service Environment (ASE) is an App Service feature that provides a fully isolated and dedicated environment for securely running App Service apps at high scale.
Kubernetes (Direct or Azure Arc)	Kubernetes provides a fully isolated and dedicated environment running on top of the Kubernetes platform.

Hosting plans and scaling

The following table compares the scaling behaviors of the various hosting plans. Maximum instances are given on a per-function app (Consumption) or per-plan (Premium/Dedicated) basis, unless otherwise indicated.

Expand table

Plan	Scale out	Max # instances
Consumption plan	Event driven. Scale out automatically, even during periods of high load. Azure Functions infrastructure scales CPU and memory resources by adding more instances of the Functions host, based on the number of incoming trigger events.	Windows: 200, Linux: 100
Premium plan	Event driven. Scale out automatically, even during periods of high load. Azure Functions infrastructure scales CPU and memory resources by adding more instances of the Functions host, based on the number of events that its functions are triggered on.	Windows: 100, Linux: 20-100
Dedicated plan	Manual/autoscale	10-20

Plan	Scale out	Max # instances
ASE	Manual/autoscale	100
Kubernetes	Event-driven autoscale for Kubernetes clusters using KEDA.	Varies by cluster

Note

The maximum scale out can vary by region and hosting plan. For more information, visit the **Premium plan article** and **App Service plan limits**.

Function app timeout duration

The `functionTimeout` property in the *host.json* project file specifies the timeout duration for functions in a function app. This property applies specifically to function executions. After the trigger starts function execution, the function needs to return/respond within the timeout duration.

The following table shows the default and maximum values (in minutes) for specific plans:

Expand table

Plan	Default	Maximum
Consumption plan	5	10
Premium plan	30	Unlimited
Dedicated plan	30	Unlimited

Storage account requirements

On any plan, a function app requires a general Azure Storage account, which supports Azure Blob, Queue, Files, and Table storage. This is because Functions rely on Azure Storage for operations such as managing triggers and logging function executions, but some storage accounts don't support queues and tables.

The same storage account used by your function app can also be used by your triggers and bindings to store your application data. However, for storage-intensive operations, you should use a separate storage account.

Next unit: Scale Azure Functions

Continue >