

Configure Azure App Service plans

1. Introduction

<https://learn.microsoft.com/en-us/training/modules/configure-app-service-plans/1-introduction>

Introduction

Completed

Azure Administrators need to be able to scale a web application. Scaling enables an application to remain responsive during periods of high demand. Scaling also helps to save money by reducing the resources required when demand drops.

Suppose you work for a large chain of hotels. You're responsible for maintaining the hotel website. Customers visit the website to make new reservations and view details for their current bookings. At certain times of the year, the volume of website traffic grows because customers are browsing hotels for vacations during national/regional holidays. At other times, traffic declines. These website usage patterns are predictable.

In this module, you learn to implement Azure App Service plans. You learn how different App Service plans provide different pricing and scaling options. You learn how changing the plan affects performance.

The goal of this module is to ensure you can determine the best App Service plan for your application.

Learning objectives

In this module, you learn how to:

- Select an appropriate Azure App Service plan pricing tier.
- Scale an Azure App Service plan.

Skills measured

The content in the module helps you prepare for [Exam AZ-104: Microsoft Azure Administrator](#).

Prerequisites

- Basic knowledge of scaling and performance concepts.
- Familiarity with the Azure portal so you can configure the correct App Service plan.

2. Implement Azure App Service plans

<https://learn.microsoft.com/en-us/training/modules/configure-app-service-plans/2-implement-azure>

Implement Azure App Service plans

Completed

An App Service plan defines a set of compute resources for a web application to run. The compute resources are analogous to a server farm in conventional web hosting. One or more applications can be configured to run on the same computing resources (or in the same App Service plan).

Things to know about App Service plans

Let's take a closer look at how to implement and use an App Service plan with your virtual machines.

- When you create an App Service plan in a region, a set of compute resources is created for the plan in the specified region. Any applications that you place into the plan run on the compute resources defined by the plan.
- Each App Service plan defines these settings:
 - **Operating system:** Linux or Windows.
 - **Region:** The region for the App Service plan, such as West US, Central India, North Europe, and so on.
 - **Pricing tier:** Determines what App Service features you get and how much you pay for the plan. The pricing tiers available to your App Service plan depend on the operating system selected at creation time.
 - **Number of VM instances:** Determined by your plan.
 - **Size of VM instances:** Defined by CPU, memory, and remote storage.
- You can continue to add new applications to an existing plan as long as the plan has enough resources to handle the increasing load.

Things to consider when using App Service plans

Review the following considerations about using Azure App Service plans to run and scale your applications. Think about what conditions might apply to running and scaling the hotel website.

- **Consider cost savings.** Because you pay for the computing resources that your App Service plan allocates, you can potentially save money by placing multiple applications into the same App Service plan.
- **Consider multiple applications in one plan.** Create a single plan to support multiple applications, to make it easier to configure and maintain shared virtual machine instances. Because the applications share the same virtual machine instances, you need to carefully manage your plan resources and capacity.
- **Consider plan capacity.** Before you add a new application to an existing plan, determine the resource requirements for the new application and identify the remaining capacity of your plan.

Important

Overloading an App Service plan can potentially cause downtime for new and existing applications.

- **Consider application isolation.** Isolate your application into a new App Service plan when:
 - The application is resource-intensive.
 - You want to scale the application independently from the other applications in the existing plan.
 - The application needs resource in a different geographical region.

3. Determine Azure App Service plan pricing

<https://learn.microsoft.com/en-us/training/modules/configure-app-service-plans/3-determine-plan-pricing>

Determine Azure App Service plan pricing

Completed

The pricing tier of an Azure App Service plan determines what App Service features you get and how much you pay for the plan. Pricing tier examples are: Free, Shared, Basic, Standard, Premium, PremiumV2, PremiumV3, Isolated, and IsolatedV2.

How applications run and scale in App Service plans

The Azure App Service plan is the scale unit of App Service applications. Depending on the pricing tier for your Azure App Service plan, your applications run and scale in a different manner. If your plan is configured to run five virtual machine instances, then all applications in the plan run on all five instances. If your plan is configured for autoscaling, then all applications in the plan are scaled out together based on the autoscale settings.

The pricing tiers are grouped into three categories:

- **Shared compute:**
 - Free and Shared, the two base tiers, run an app on the same Azure VM as other App Service apps, including apps of other customers.
 - These tiers allocate CPU quotas to each app that runs on the shared resources, and the resources can't scale out.
 - These tiers are intended to be used only for development and testing purposes.
- **Dedicated compute:**
 - The Basic, Standard, Premium, PremiumV2, and PremiumV3 tiers run apps on dedicated Azure VMs.
 - Only apps in the same App Service plan have the same compute resources. The higher the tier, the more VM instances that are available to you for scale-out.
- **Isolated:**
 - The Isolated and IsolatedV2 tiers run dedicated Azure VMs on dedicated Azure virtual networks.
 - This tier provides network isolation on top of compute isolation to your apps.
 - This tier provides the maximum scale-out capabilities.

Here's a sample of different [plan details](#).

Feature	Free F1	Basic B1	Standard S1	Premium P1V3	Isolated V2
Usage	Development, Testing	Development, Testing	Production workloads	Enhanced scale, performance	Network-isolated workloads
Staging slots	N/A	N/A	5	20	20
Auto scale	N/A	Manual	Rules	Rules, Elastic	Rules
Scale instances	N/A	3	10	30	200
Daily backups	N/A	N/A	10	50	50

Free and Shared

The Free and Shared service plans are base tiers that run on the same Azure virtual machines as other applications. Some applications might belong to other customers. These tiers are intended to be

used for development and testing purposes only. No SLA is provided for the Free and Shared service plans. Free and Shared plans are metered on a per application basis.

Basic

The Basic service plan is designed for applications that have lower traffic requirements, and don't need advanced auto scale and traffic management features. Pricing is based on the size and number of instances you run. Built-in network load-balancing support automatically distributes traffic across instances. The Basic service plan with Linux runtime environments supports Web App for Containers.

Standard

The Standard service plan is designed for running production workloads. Pricing is based on the size and number of instances you run. Built-in network load-balancing support automatically distributes traffic across instances. The Standard plan includes auto scale that can automatically adjust the number of virtual machine instances running to match your traffic needs. The Standard service plan with Linux runtime environments supports Web App for Containers.

Premium

The Premium service plan is designed for production apps that need higher performance and scale. PremiumV3 is the current Premium tier, offering Dav4 and Ddv4-series virtual machines and SSD storage. PremiumV3 supports standard compute SKUs and memory-optimized SKUs for high-memory workloads. PremiumV3 supports both rule-based autoscaling and automatic scaling. PremiumV3 is recommended for new deployments.

Isolated

The Isolated service plan supports mission-critical workloads needing network isolation. IsolatedV2 is the preferred tier offering newer hardware, up to 200 instances, private environments, and enhanced security. IsolatedV2 is recommended for new workloads due to better performance and simpler pricing.

Task to be done: Select an App Service plan

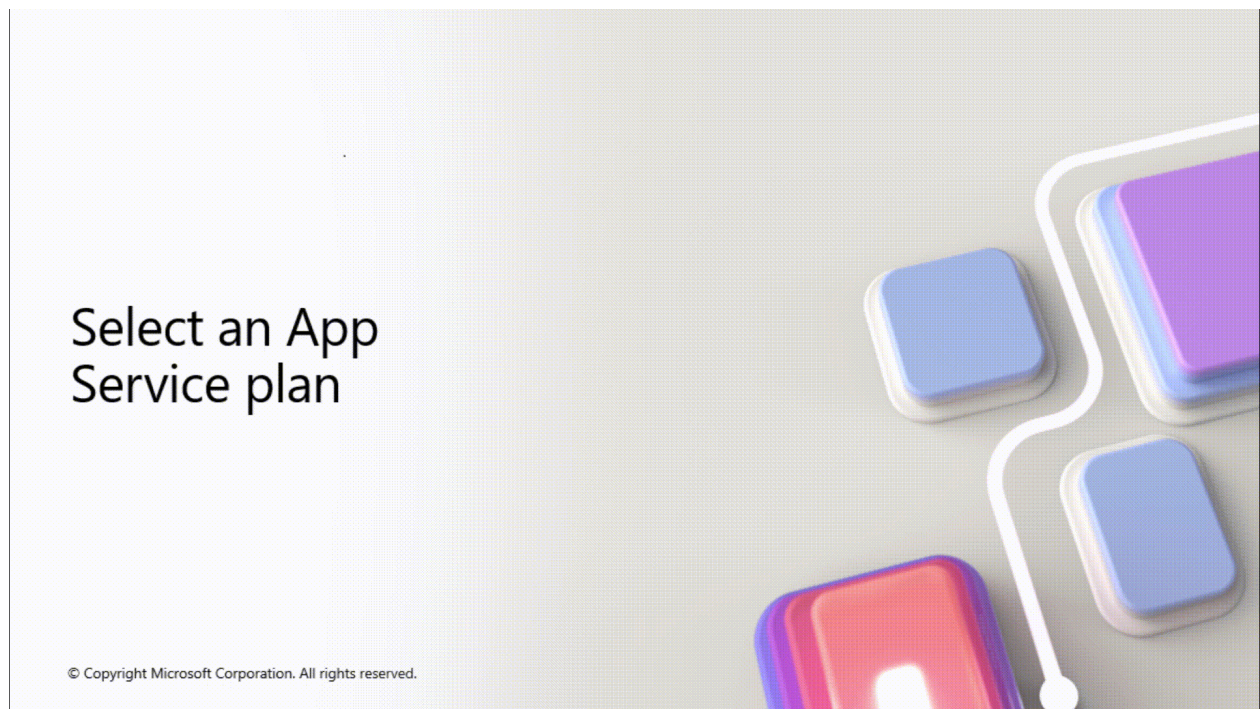
You can view the available App Service plans in the Azure portal. You can make your choice based on hardware or feature requirements. Hardware considerations include CPU, memory, and scaling instances. Feature considerations include backups, staging slots, and zone redundancy.

Tip

When selecting a service plan, consider both hardware and feature requirements.

1. In the Azure portal search for and select **App Service plans**.
2. **Create** a new App Service plan.

3. Select **Explore pricing plans** to view the available plans.



4. Scale up and scale out Azure App Service

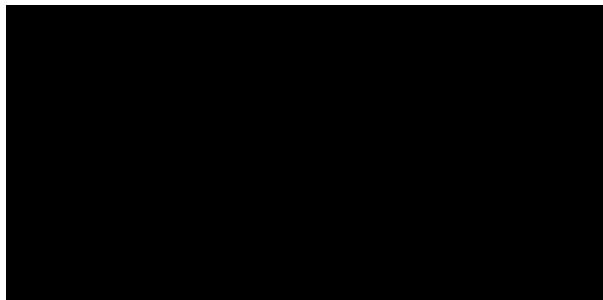
<https://learn.microsoft.com/en-us/training/modules/configure-app-service-plans/4-scale-up-scale-out>

Scale up and scale out Azure App Service

Completed

There are two methods for scaling your Azure App Service plan and applications: *scale up* and *scale out*. You can scale your applications manually or automatically, which is referred to as *autoscale*.

Watch the following video about how to implement automatic scaling for your Azure App Service plan and applications.



Things to know about Azure App Service scaling

Let's examine the details of scaling for your Azure App Service plan and App Service applications.

- The scale up method increases the amount of CPU, memory, and disk space. Scaling up gives you extra features like dedicated virtual machines, custom domains and certificates, staging slots, autoscaling, and more. You scale up by changing the pricing tier of the Azure App Service plan where your application is placed.
- The scale-out method increases the number of virtual machine instances that run your application. You can scale out to the maximum number of instances for your pricing tier. Take advantage of App Service Environments in the Isolated tier to further increase your scale-out count to 100 instances. The scale instance count can be configured manually or automatically (autoscale).
- With autoscale, you can automatically increase the scale instance count for the scale-out method. Autoscale is based on predefined rules and schedules.
- Your App Service plan can be scaled up and down at any time by changing the pricing tier of the plan.

Things to consider when using Azure App Service scaling

Review the following benefits of implementing scaling for your App Service plan and applications. Think about the scaling advantages for your hotel website.

- **Consider manually adjusting plan tiers.** Start your plan at a lower pricing tier and scale up as needed to acquire more App Service features. Scale down when features are no longer needed, and control your overall costs.

Consider a scenario where you start testing your web app by using the Azure App Service Free tier, where you pay nothing to use the service. After a while, you decide to add a custom DNS name to your web app, so you scale your plan up to the Shared tier. Next, you discover you need to create an SSL binding, so you scale your plan up to the Basic tier. Later, you determine a need for staging environments, so you scale up to the Standard tier. When you need more cores, memory, or storage, you can scale up to a bigger virtual machine size in the same tier.

The same scaling process works in reverse. If you decide you no longer need capabilities or features of a higher tier, scale your plan down to a lower tier and save money.

- **Consider autoscale to support users and reduce costs.** Keep serving your users when your application is experiencing high throughput. Implement autoscale to control how many features and support are offered at a given time based on your preference settings and rule conditions. Autoscale helps you save money when the load on your application decreases by automatically reducing your subscribed features.
- **Consider no redeployment.** When you change your scale settings, you don't need to change your code or redeploy your applications. Changing your plan scale settings takes only seconds

to apply. Your changes affect all applications in your App Service plan.

- **Consider scaling for other Azure services.** If your App Service application depends on other Azure services, such as Azure SQL Database or Azure Storage, you can scale these resources separately. The App Service Plan doesn't manage these resources.

5. Configure Azure App Service autoscale

<https://learn.microsoft.com/en-us/training/modules/configure-app-service-plans/5-plan-scaling>

Configure Azure App Service autoscale

Completed

The autoscale process allows you to have the right amount of resources running to handle the load on your application. You can add resources to support increases in load and save money by removing idle resources.

Things to know about autoscale

Let's take a closer look at how to use autoscale for your Azure App Service plan and applications.

- To use autoscale, you specify the minimum, and maximum number of instances to run by using a set of rules and conditions.
- When your application runs under autoscale conditions, the number of virtual machine instances are automatically adjusted based on your rules. When rule conditions are met, one or more autoscale actions are triggered.
- An autoscale setting is used by the autoscale engine to determine whether to scale out or in. Autoscale settings are grouped into profiles.
- Autoscale rules include a trigger and a scale action (in or out). The trigger can be metric-based or time-based.

- **Metric-based** rules measure application load and add or remove virtual machines based on the load, such as "do this action when CPU usage is above 50%." Example metrics include CPU time, Average response time, and Requests.
- **Time-based** rules (or, schedule-based) allow you to scale when you see time patterns in your load and want to scale before a possible load increase or decrease occurs. An example is "trigger a webhook every 8:00 AM on Saturday in a given time zone."
- The autoscale engine uses notification settings.

A notification setting defines what notifications should occur when an autoscale event occurs based on satisfying the criteria of an autoscale setting profile. Autoscale can notify one or more email addresses or make calls to one or more webhooks.

Things to consider when configuring autoscale

There are several considerations to keep in mind when you configure autoscale for your Azure App Service plan and applications.

- **Minimum instance count.** Set a minimum instance count to make sure your application is always running even when there's no load.
- **Maximum instance count.** Set a maximum instance count to limit your total possible hourly cost.
- **Adequate scale margin.** Make sure your maximum and minimum instance count values are different, and set an adequate margin between the two values. You can automatically scale between the minimum and maximum by using rules you create.
- **Scale rule combinations.** Always use a scale-out and scale-in rule combination that performs an increase and decrease. If you don't set a scale-out rule, your application might fail, or performance might degrade under increased loads. If you don't set a scale-in rule, you can experience unnecessary and extensive costs when the load decreases.

- **Metric statistics.** Carefully choose the appropriate statistic for your diagnostic metrics, including Average, Minimum, Maximum, and Total.
- **Default instance count.** Always select a safe default instance count. The default instance count is important because autoscale scales your service to the count you specify when metrics aren't available.
- **Notifications.** Always configure autoscale notifications. It's important to maintain awareness of how your application is performing as the load changes.

Things to consider when configuring automatic scaling

In addition to rule-based autoscale, Azure App Service offers Automatic scaling (also called Elastic scaling) for PremiumV2 and PremiumV3 tiers. This is a separate scaling feature that works differently from the autoscale rules.

- **HTTP traffic-based.** Automatic scaling responds directly to incoming HTTP requests without requiring you to configure scaling rules.
- **Platform-managed.** Azure automatically manages the scaling decisions based on traffic patterns, eliminating the need for rule configuration.
- **Always-ready instances.** Maintains warmed instances to handle traffic spikes immediately.
- **Tier availability.** Available only on PremiumV2 and PremiumV3 tiers.

Choose between Autoscale and Automatic scaling

- **Use rule-based Autoscale.** You need custom scaling logic, want to scale based on multiple metrics, or need schedule-based scaling.
- **Use Automatic scaling.** You want less management, can't predict load patterns, or need fast response to traffic changes without rule configuration.

6. Module assessment

<https://learn.microsoft.com/en-us/training/modules/configure-app-service-plans/6-knowledge-check>

Module assessment

Completed

7. Summary and resources

<https://learn.microsoft.com/en-us/training/modules/configure-app-service-plans/7-summary-resources>

Summary and resources

Completed

In this module, you learned about Azure App Service plans and how they're used to define the compute resources for running applications in Azure App Service. These plans can be configured with a specific region, number of virtual machine instances, and size of virtual machine instances. The pricing tier of the App Service plan determines the features and cost. Pricing tiers include Free and Shared plans for development and testing purposes. Pricing tiers also include Isolated plans for mission-critical workloads.

You learned about scaling in Azure App Service. Scale up involves increasing the CPU, memory, and disk space by changing the pricing tier. Scale out increases the number of virtual machine instances running the application. Autoscaling allows you to automatically adjust the number of resources based on the load on your application. Autoscale can be configured with metric-based or time-based rules.

The main takeaways from this module are:

- Azure App Service plans are used to define the compute resources for running web applications in Azure App Service.
- The pricing tier of the App Service plan determines the features and cost, with options ranging from Free and Shared plans to Isolated plans.
- Scaling in Azure App Service can be done through scale up (changing the pricing tier) or scale out (increasing the number of virtual machine instances).
- Autoscaling allows for automatic adjustment of resources based on application load, with metric-based and time-based rules.

Learn more with Copilot

Copilot can assist you in configuring Azure infrastructure solutions. Copilot can compare, recommend, explain, and research products and services where you need more information. Open a Microsoft Edge browser and choose Copilot (top right) or navigate to copilot.microsoft.com. Take a few minutes to try these prompts and extend your learning with Copilot.

- In Microsoft Azure, what are app service pricing plans? Provide examples of when to use each plan.

- In Microsoft Azure, what does scale in and scale out mean? How do I determine when to scale an application?

Learn more with documentation

- [Azure App Service plans](#). This article provides an overview of App Service plans.
- [Manage an App Service plan in Azure](#). This guide shows how to create and manage an App Service plan.
- [Scale up an app in Azure App Service](#). This article shows you how to scale your app in Azure App Service.

Learn more with self-paced training

- [Scale apps in Azure App Service](#). Learn how autoscale operates in App Service. Learn to identify autoscale factors, enable autoscale, and create autoscale conditions.