

✓ 100 XP 

Introduction

3 minutes

The deployment slot functionality in App Service is a powerful tool that enables you to preview, manage, test, and deploy your different development environments.

Learning objectives

After completing this module, you'll be able to:

- Describe the benefits of using deployment slots
- Understand how slot swapping operates in App Service
- Perform manual swaps and enable auto swap
- Route traffic manually and automatically

Prerequisites

- Experience using the Azure portal to create and manage App Service web apps

Next unit: Explore staging environments

[Continue >](#)

How are we doing? ☆ ☆ ☆ ☆ ☆

Explore staging environments

3 minutes

When you deploy your web app, web app on Linux, mobile back end, or API app to Azure App Service, you can use a separate deployment slot instead of the default production slot when you're running in the **Standard**, **Premium**, or **Isolated** App Service plan tier. Deployment slots are live apps with their own host names. App content and configurations elements can be swapped between two deployment slots, including the production slot.

Deploying your application to a non-production slot has the following benefits:

- You can validate app changes in a staging deployment slot before swapping it with the production slot.
- Deploying an app to a slot first and swapping it into production makes sure that all instances of the slot are warmed up before being swapped into production. This eliminates downtime when you deploy your app. The traffic redirection is seamless, and no requests are dropped because of swap operations. You can automate this entire workflow by configuring auto swap when pre-swap validation isn't needed.
- After a swap, the slot with previously staged app now has the previous production app. If the changes swapped into the production slot aren't as you expect, you can perform the same swap immediately to get your "last known good site" back.

Each App Service plan tier supports a different number of deployment slots. There's no additional charge for using deployment slots. To find out the number of slots your app's tier supports, visit [App Service limits](#).

To scale your app to a different tier, make sure that the target tier supports the number of slots your app already uses. For example, if your app has more than five slots, you can't scale it down to the **Standard** tier, because the **Standard** tier supports only five deployment slots.

When you create a new slot the new deployment slot has no content, even if you clone the settings from a different slot. You can deploy to the slot from a different repository branch or a different repository.

Next unit: Examine slot swapping

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

Examine slot swapping

3 minutes

When you swap slots (for example, from a staging slot to the production slot), App Service does the following to ensure that the target slot doesn't experience downtime:

1. Apply the following settings from the target slot (for example, the production slot) to all instances of the source slot:

- Slot-specific app settings and connection strings, if applicable.
- Continuous deployment settings, if enabled.
- App Service authentication settings, if enabled.

Any of these cases trigger all instances in the source slot to restart. During **swap with preview**, this marks the end of the first phase. The swap operation is paused, and you can validate that the source slot works correctly with the target slot's settings.

2. Wait for every instance in the source slot to complete its restart. If any instance fails to restart, the swap operation reverts all changes to the source slot and stops the operation.
3. If local cache is enabled, trigger local cache initialization by making an HTTP request to the application root ("/") on each instance of the source slot. Wait until each instance returns any HTTP response. Local cache initialization causes another restart on each instance.
4. If auto swap is enabled with custom warm-up, trigger Application Initiation by making an HTTP request to the application root ("/") on each instance of the source slot.
 - If `applicationInitialization` isn't specified, trigger an HTTP request to the application root of the source slot on each instance.
 - If an instance returns any HTTP response, it's considered to be warmed up.
5. If all instances on the source slot are warmed up successfully, swap the two slots by switching the routing rules for the two slots. After this step, the target slot (for example, the production slot) has the app that's previously warmed up in the source slot.

6. Now that the source slot has the pre-swap app previously in the target slot, perform the same operation by applying all settings and restarting the instances.

At any point of the swap operation, all work of initializing the swapped apps happens on the source slot. The target slot remains online while the source slot is being prepared and warmed up, regardless of where the swap succeeds or fails. To swap a staging slot with the production slot, make sure that the production slot is always the target slot. This way, the swap operation doesn't affect your production app.

When you clone configuration from another deployment slot, the cloned configuration is editable. Some configuration elements follow the content across a swap (not slot specific), whereas other configuration elements stay in the same slot after a swap (slot specific). The following table shows the settings that change when you swap slots.

Settings that are swapped	Settings that aren't swapped
General settings, such as framework version, 32/64-bit, web sockets	Publishing endpoints
App settings (can be configured to stick to a slot)	Custom domain names
Connection strings (can be configured to stick to a slot)	Non-public certificates and TLS/SSL settings
Handler mappings	Scale settings
Public certificates	WebJobs schedulers
WebJobs content	IP restrictions
Hybrid connections *	Always On
Virtual network integration *	Diagnostic log settings
Service endpoints *	Cross-origin resource sharing (CORS)
Azure Content Delivery Network *	

Features marked with an asterisk (*) are planned to be unswapped.

Note

To make settings swappable, add the app setting `WEBSITE_OVERRIDE_PRESERVE_DEFAULT_STICKY_SLOT_SETTINGS` in every slot of the app and set its value to `0` or `false`. These settings are either all swappable or not at all. You can't make just some settings swappable and not the others. Managed identities are never swapped and are not affected by this override app setting.

To configure an app setting or connection string to stick to a specific slot (not swapped), go to the Configuration page for that slot. Add or edit a setting, and then select **Deployment slot setting**. Selecting this check box tells App Service that the setting is not swappable.

Next unit: Swap deployment slots

[Continue >](#)

How are we doing? ☆ ☆ ☆ ☆ ☆

Swap deployment slots

3 minutes

You can swap deployment slots on your app's Deployment slots page and the Overview page. Before you swap an app from a deployment slot into production, make sure that production is your target slot and that all settings in the source slot are configured exactly as you want to have them in production.

Manually swapping deployment slots

To swap deployment slots:

1. Go to your app's **Deployment slots** page and select **Swap**. The **Swap** dialog box shows settings in the selected source and target slots that will be changed.
2. Select the desired **Source** and **Target** slots. Usually, the target is the production slot. Also, select the **Source Changes** and **Target Changes** tabs and verify that the configuration changes are expected. When you're finished, you can swap the slots immediately by selecting **Swap**.

To see how your target slot would run with the new settings before the swap actually happens, don't select **Swap**, but follow the instructions in *Swap with preview* below.

3. When you're finished, close the dialog box by selecting **Close**.

Swap with preview (multi-phase swap)

Before you swap into production as the target slot, validate that the app runs with the swapped settings. The source slot is also warmed up before the swap completion, which is desirable for mission-critical applications.

When you perform a swap with preview, App Service performs the same swap operation but pauses after the first step. You can then verify the result on the staging slot before completing the swap.

If you cancel the swap, App Service reapplies configuration elements to the source slot.

To swap with preview:

1. Follow the steps above in Swap deployment slots but select **Perform swap with preview**.
The dialog box shows you how the configuration in the source slot changes in phase 1, and how the source and target slot change in phase 2.
2. When you're ready to start the swap, select **Start Swap**.

When phase 1 finishes, you're notified in the dialog box. Preview the swap in the source slot by going to `https://<app_name>-<source-slot-name>.azurewebsites.net`.

3. When you're ready to complete the pending swap, select **Complete Swap** in **Swap action** and select **Complete Swap**.

To cancel a pending swap, select **Cancel Swap** instead.

4. When you're finished, close the dialog box by selecting **Close**.

Configure auto swap

Auto swap streamlines Azure DevOps scenarios where you want to deploy your app continuously with zero cold starts and zero downtime for customers of the app. When auto swap is enabled from a slot into production, every time you push your code changes to that slot, App Service automatically swaps the app into production after it's warmed up in the source slot.

ⓘ Note

Auto swap isn't currently supported in web apps on Linux.

To configure auto swap:

1. Go to your app's resource page and select the deployment slot you want to configure to auto swap. The setting is on the **Configuration > General settings** page.
2. Set **Auto swap enabled** to **On**. Then select the desired target slot for Auto swap deployment slot, and select **Save** on the command bar.
3. Execute a code push to the source slot. Auto swap happens after a short time, and the update is reflected at your target slot's URL.

Specify custom warm-up

Some apps might require custom warm-up actions before the swap. The `applicationInitialization` configuration element in `web.config` lets you specify custom initialization actions. The swap operation waits for this custom warm-up to finish before swapping with the target slot. Here's a sample `web.config` fragment.

 Copy

```
<system.webServer>
  <applicationInitialization>
    <add initializationPage="/" hostName="[app hostname]" />
    <add initializationPage="/Home/About" hostName="[app hostname]" />
  </applicationInitialization>
</system.webServer>
```

For more information on customizing the `applicationInitialization` element, see [Most common deployment slot swap failures and how to fix them](#).

You can also customize the warm-up behavior with one or both of the following app settings:

- `WEBSITE_SWAP_WARMUP_PING_PATH`: The path to ping to warm up your site. Add this app setting by specifying a custom path that begins with a slash as the value. An example is `/statuscheck`. The default value is `/`.
- `WEBSITE_SWAP_WARMUP_PING_STATUSES`: Valid HTTP response codes for the warm-up operation. Add this app setting with a comma-separated list of HTTP codes. An example is `200,202`. If the returned status code isn't in the list, the warmup and swap operations are stopped. By default, all response codes are valid.

Roll back and monitor a swap

If any errors occur in the target slot (for example, the production slot) after a slot swap, restore the slots to their pre-swap states by swapping the same two slots immediately.

If the swap operation takes a long time to complete, you can get information on the swap operation in the activity log.

On your app's resource page in the portal, in the left pane, select **Activity log**.

A swap operation appears in the log query as `Swap Web App Slots`. You can expand it and select one of the suboperations or errors to see the details.

Next unit: Route traffic in App Service

[Continue >](#)

How are we doing? ☆ ☆ ☆ ☆ ☆

Route traffic in App Service

3 minutes

By default, all client requests to the app's production URL (`http://<app_name>.azurewebsites.net`) are routed to the production slot. You can route a portion of the traffic to another slot. This feature is useful if you need user feedback for a new update, but you're not ready to release it to production.

Route production traffic automatically

To route production traffic automatically:

1. Go to your app's resource page and select **Deployment slots**.
2. In the **Traffic** % column of the slot you want to route to, specify a percentage (between 0 and 100) to represent the amount of total traffic you want to route. Select **Save**.

After the setting is saved, the specified percentage of clients is randomly routed to the non-production slot.

After a client is automatically routed to a specific slot, it's "pinned" to that slot for the life of that client session. On the client browser, you can see which slot your session is pinned to by looking at the `x-ms-routing-name` cookie in your HTTP headers. A request that's routed to the "staging" slot has the cookie `x-ms-routing-name=staging`. A request that's routed to the production slot has the cookie `x-ms-routing-name=self`.

Route production traffic manually

In addition to automatic traffic routing, App Service can route requests to a specific slot. This is useful when you want your users to be able to opt in to or opt out of your beta app. To route production traffic manually, you use the `x-ms-routing-name` query parameter.

To let users opt out of your beta app, for example, you can put this link on your webpage:

HTML

 Copy

```
<a href="<webappname>.azurewebsites.net/?x-ms-routing-name=self">Go back to produc-  
tion app</a>
```

The string `x-ms-routing-name=self` specifies the production slot. After the client browser accesses the link, it's redirected to the production slot. Every subsequent request has the `x-ms-routing-name=self` cookie that pins the session to the production slot.

To let users opt in to your beta app, set the same query parameter to the name of the non-production slot. Here's an example:

HTML

 Copy

```
<webappname>.azurewebsites.net/?x-ms-routing-name=staging
```

By default, new slots are given a routing rule of 0%, a default value is displayed in grey. When you explicitly set this value to 0% it is displayed in black, your users can access the staging slot manually by using the `x-ms-routing-name` query parameter. But they won't be routed to the slot automatically because the routing percentage is set to 0. This is an advanced scenario where you can "hide" your staging slot from the public while allowing internal teams to test changes on the slot.

Next unit: Knowledge check

[Continue >](#)

How are we doing? ☆ ☆ ☆ ☆ ☆

Knowledge check

3 minutes

Check your knowledge

1. By default, all client requests to the app's production URL (`http://<app_name>.azurewebsites.net`) are routed to the production slot. One can automatically route a portion of the traffic to another slot. What is the default routing rule applied to new deployment slots?

☒ 0%

✓ That's correct. By default, new slots are given a routing rule of 0%.

☐ 10%

☐ 20%

2. Some configuration elements follow the content across a swap (not slot specific), whereas other configuration elements stay in the same slot after a swap (slot specific). Which of the settings below are swapped?

☐ Publishing endpoints

☒ WebJobs content

✓ That's correct. WebJobs content are swapped.

☐ WebJobs schedulers

Next unit: Summary

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

[< Previous](#)

Unit 7 of 7

✓ 100 XP



Summary

3 minutes

In this module, you learned how to:

- Describe the benefits of using deployment slots
- Understand how slot swapping operates in App Service
- Perform manual swaps and enable auto swap
- Route traffic manually and automatically

Module complete:

[Review your Learning Path history >](#)[Explore other paths](#)

How are we doing? ☆ ☆ ☆ ☆ ☆

