
◆ Configure Deployment Slots for Azure App Service

1. What Are Deployment Slots?

- A **deployment slot** is a live App Service environment with its own **hostname, configuration, and content**.
- Common slots:
 - **Production (default)**
 - **Staging**
 - (Optional) Dev, Test, Pre-Prod, etc.
- Benefits:
 - Zero-downtime deployments.
 - Validate new code in staging before swapping to production.
 - Rollback quickly if issues occur.

2. Prerequisites

- App Service Plan in **Standard, Premium, or Isolated** tier.
(Free/Shared/Basic tiers ❌ do not support slots).
 - An existing **App Service** deployed.
-

3. Create a Deployment Slot

1. In [Azure Portal](#) → Go to your **App Service**.
2. In the left menu, click **Deployment slots** → **+ Add Slot**.
3. Enter:
 - **Name:** e.g., `staging`.
 - **Configuration Source:**
 - Select **Clone settings from Production** (copies app settings, connection strings, etc.).
 - Or choose **Don't clone** to start fresh.
4. Click **Add** → Wait for slot creation.

👉 Each slot gets a unique URL like:

- Production: `https://myapp.azurewebsites.net`
 - Staging: `https://myapp-staging.azurewebsites.net`
-

4. Deploy Code to Staging Slot

- You can deploy your app to the **staging slot** instead of production using:
 - **Azure DevOps Pipelines / GitHub Actions** → target the staging slot.
 - **Deployment Center** in Azure Portal → select the slot.
 - **Manual publish** (FTP/ZIP/Visual Studio).

👉 Test the new version in staging before impacting production.

5. Swap Slots (Staging ↔ Production)

1. In App Service → **Deployment slots**.
2. Click **Swap**.
3. Select:
 - **Source slot:** staging.
 - **Target slot:** production.
4. (Optional) **Preview Swap**: View configuration changes before applying.
5. Click **Swap**.

👉 This performs a **zero-downtime deployment**:

- New code goes to production.
 - Old production code moves to staging (rollback-ready).
-

6. Manage Slot Settings

- In **Configuration** → **Application Settings**, you can mark specific settings as **Slot Settings**.
- Example:
 - **Connection Strings** → keep different DB for staging vs. production.
 - API keys, secrets, or feature flags.

👉 This ensures settings do **not swap** when you swap slots.

7. Remove or Scale Slots

- Slots share the **App Service Plan resources**.
 - You can add multiple slots (limits depend on pricing tier).
 - If no longer needed, go to **Deployment slots** → **Delete slot**.
-

8. Best Practices

- Always test in a **staging slot** before swapping.
 - Use **slot-specific settings** for sensitive configs.
 - Consider **warm-up settings** (preload app before swap for faster startup).
 - Automate swaps in **CI/CD pipelines**.
-

Summary:

1. Create a new slot (e.g., staging).
 2. Deploy new code to staging.
 3. Test in staging URL.
 4. Swap staging → production for zero downtime.
 5. Use slot settings for environment-specific configs.
-