**Lab Guide: Creating and Managing an Azure Web App with Deployment Slots (GUI)**

**Overview**

This lab provides detailed steps to create and deploy a Java-based web app on Azure using the portal and Visual Studio Code, focusing on GUI-based workflows. You'll also learn how to create and manage deployment slots.

**Prerequisites**

1. **Azure Account:** Ensure you have access to an Azure subscription.
2. **Visual Studio Code Installed:** Ensure VS Code is installed along with the Azure Services extension.
3. **Maven Installed:** Check for Maven on your local machine using mvn -v.
4. **Java 17 Installed:** Verify using java -version.

**Step 1: Create a Web App on Azure**

1. **Navigate to the Azure Portal** and log in.
2. In the **Search bar**, type App Services and select the **App Services** resource.
3. Click **+ Create**.
4. Fill in the required fields:
   * **Subscription:** Select your subscription.
   * **Resource Group:** Create a new resource group or use an existing one.
   * **Instance Details:**
     + Name: Provide a unique name for your web app.
     + Runtime Stack: Select **Java 17**.
     + Operating System: **Linux**.
     + Region: Choose a region (e.g., East US 2).
   * **Pricing Plan:**
     + Click on **Change Size** and choose the **Basic B1** tier.
5. Click **Review + Create** and then **Create**.
6. Wait for the deployment to complete and click **Go to Resource** to access your web app.

**Step 2: Set Up Your Local Development Environment**

1. **Clone Your Repository**:
   * On your local machine, create a folder (e.g., Desktop/webapp-project).
   * Open **Visual Studio Code** and use the Source Control tab to clone your repository.
2. **Verify Tools in Terminal**:
   * Open the terminal in VS Code and run the following:
   * mvn -v

java -version

Ensure Maven and Java 17 are correctly installed.

1. **Build the Project**:
   * Run:

mvn clean package

* + Ensure the .jar or .war file is created in the target directory.

**Step 3: Publish the Artifact to Azure**

1. In VS Code, install the **Azure App Service Extension**.
2. **Sign in to Azure** within the extension.
3. **Deploy Artifact**:
   * Right-click your web app in the Azure Resources Explorer and select **Deploy to Web App**.
   * Navigate to your target folder, select the .jar or .war file, and follow the prompts to publish it.

**Step 4: Create a Deployment Slot**

1. Navigate to your web app in the Azure Portal.
2. In the left-hand menu, select **Deployment slots**.
3. Click **+ Add Slot**.
4. Provide a slot name (e.g., otherversion) and clone the settings from the **Production** slot.
5. Click **Add** and wait for the slot to be created.
6. Note the slot URLs:
   * **Production Slot:** https://<webapp-name>.azurewebsites.net
   * **Other Version Slot:** https://<webapp-name>-<slot-name>.azurewebsites.net

**Step 5: Update and Deploy to the Slot**

1. **Modify the Code**:
   * Open your project in VS Code and make changes. For example, update the background color to **purple** for the deployment slot version.
   * Rebuild the project:

mvn clean package

1. **Deploy to Deployment Slot**:
   * In the Azure Resources Explorer in VS Code, navigate to your app's deployment slots.
   * Right-click the otherversion slot and select **Deploy to Slot**.
   * Select the updated artifact and deploy it to the slot.

**Step 6: Verify Your Deployment**

1. Access the web app using the respective URLs:
   * **Production Slot:** https://<webapp-name>.azurewebsites.net
   * **Other Version Slot:** https://<webapp-name>-otherversion.azurewebsites.net
2. Confirm that the changes (e.g., purple background) are visible in the slot.

**Cleaning Up Resources**

1. Navigate to the Azure Portal.
2. Delete the resource group associated with your app to remove all resources:
   * Go to **Resource Groups**.
   * Select your resource group and click **Delete Resource Group**.
   * Confirm the deletion by typing the resource group name.

This concludes the lab guide for creating, managing, and deploying an Azure Web App with deployment slots using GUI-based methods.