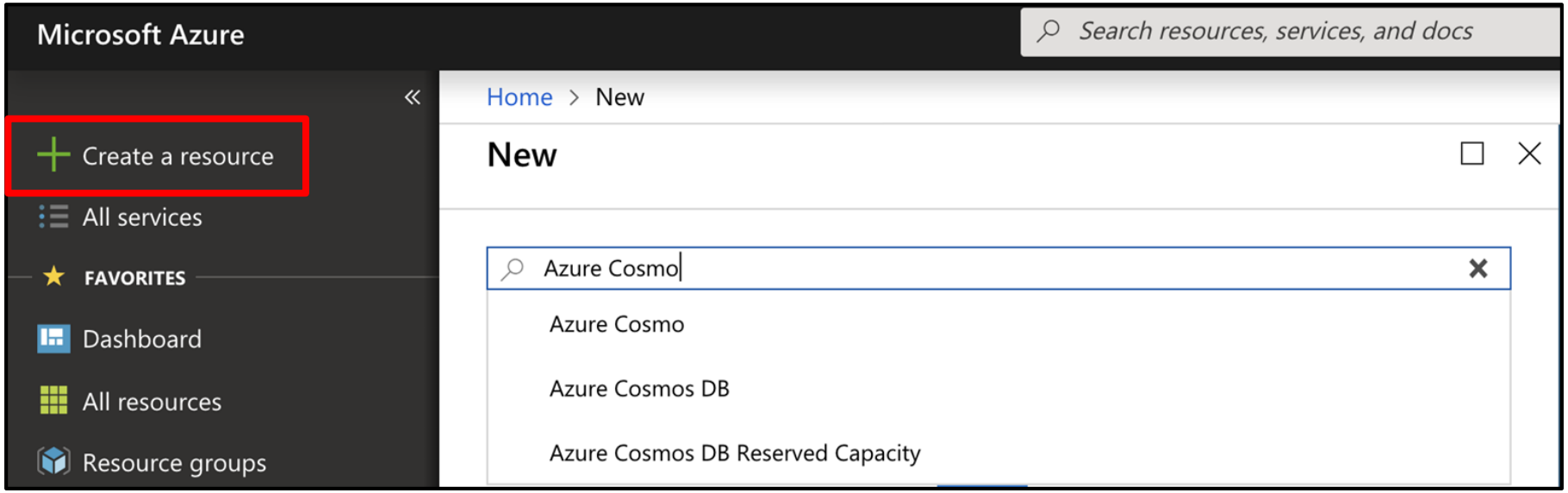
Table of Contents

Create a Cosmos Resource

First, we'll create our database on Azure.

Instructions

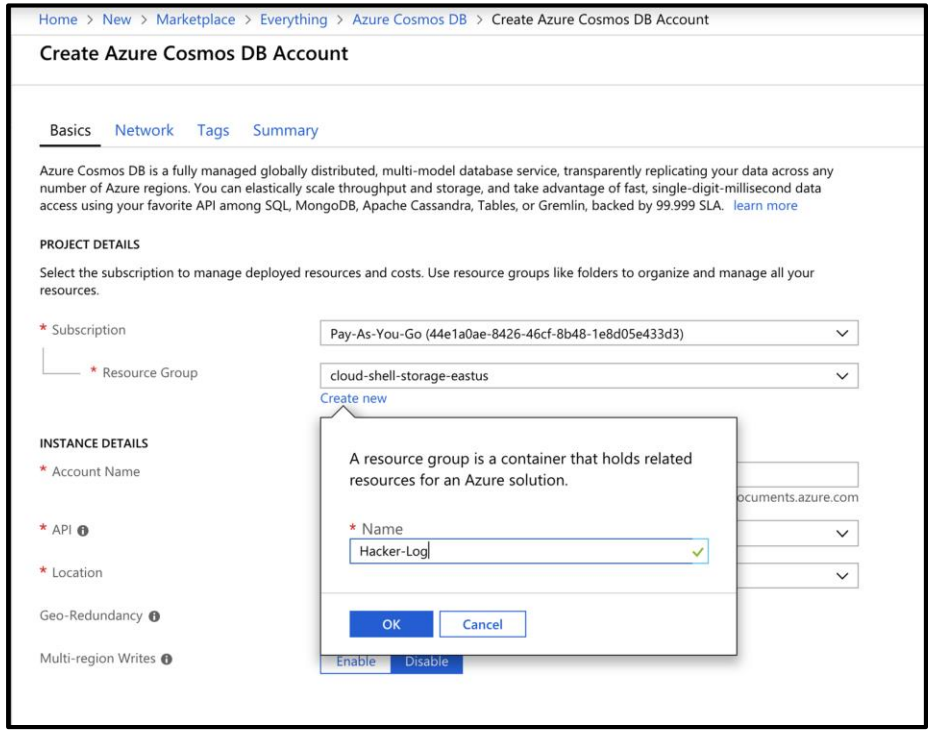
1. Navigate to your Azure dashboard and select **Create a Resource** then search for and select **Azure Cosmos DB**. Click **Create**.



Create a Cosmos Resource

Instructions

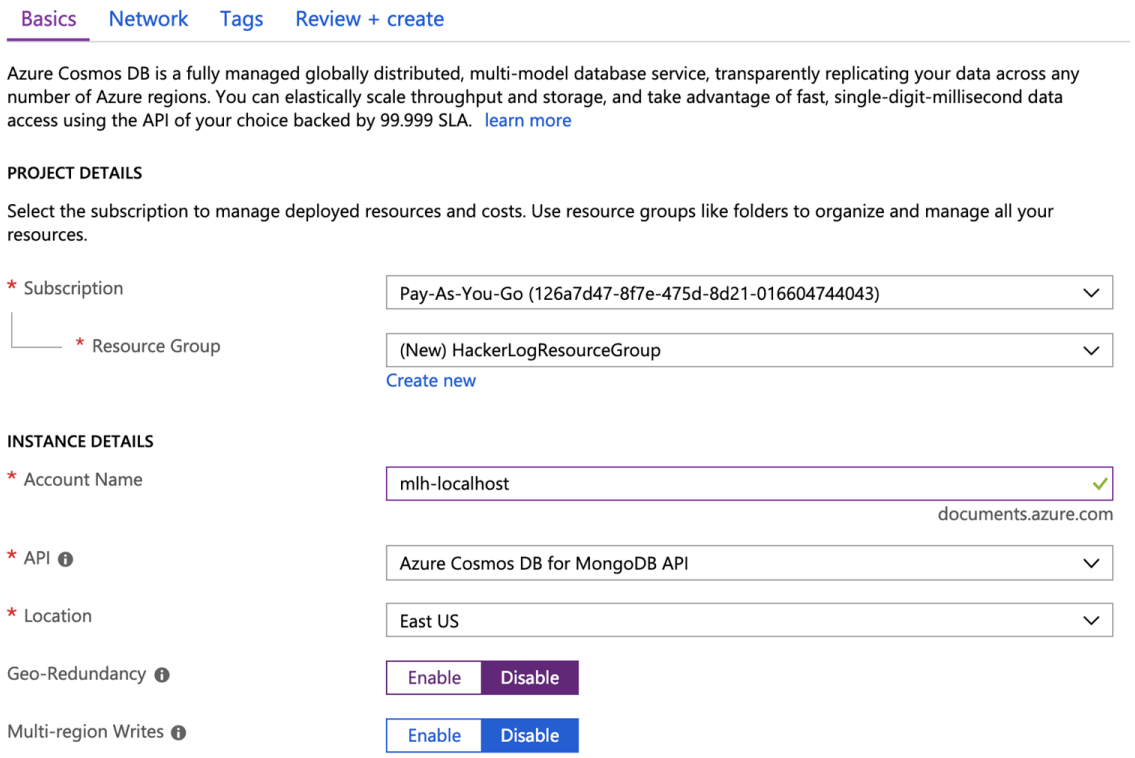
2. Select your subscription.
3. Select **Create new** under Resource Group. Resource Groups are a way of organizing all of your Azure projects.
4. Name your Resource Group. You'll add your app to this group later, too.
5. Select **OK**.



Create a Cosmos Resource

Instructions

6. Create an account name.
7. Select **Azure Cosmos DB for MongoDB** as the API.
8. Select your location.
9. Disable both options.
10. Select **Next: Network**.



Create a Cosmos Resource

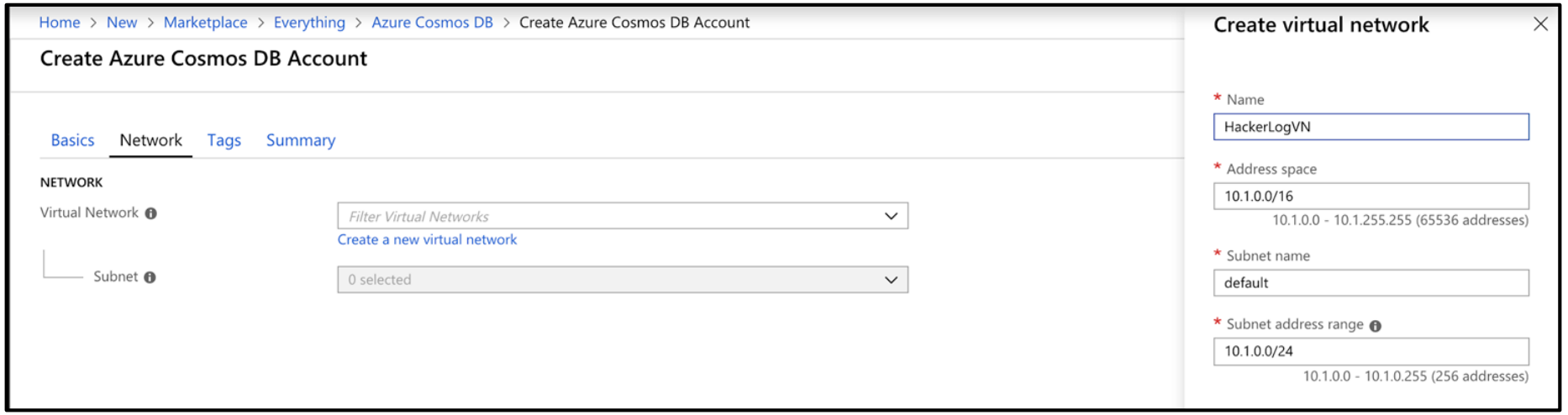
Instructions

11. Click **Create a new virtual network**.

12. Name it HackerLogVN.

13. Select **OK**.

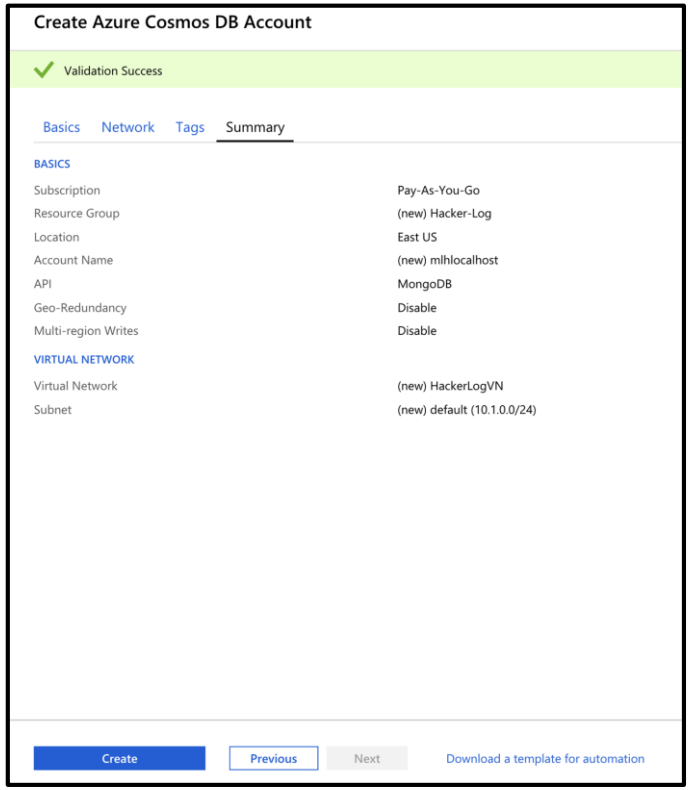
14. Click **Review + create**



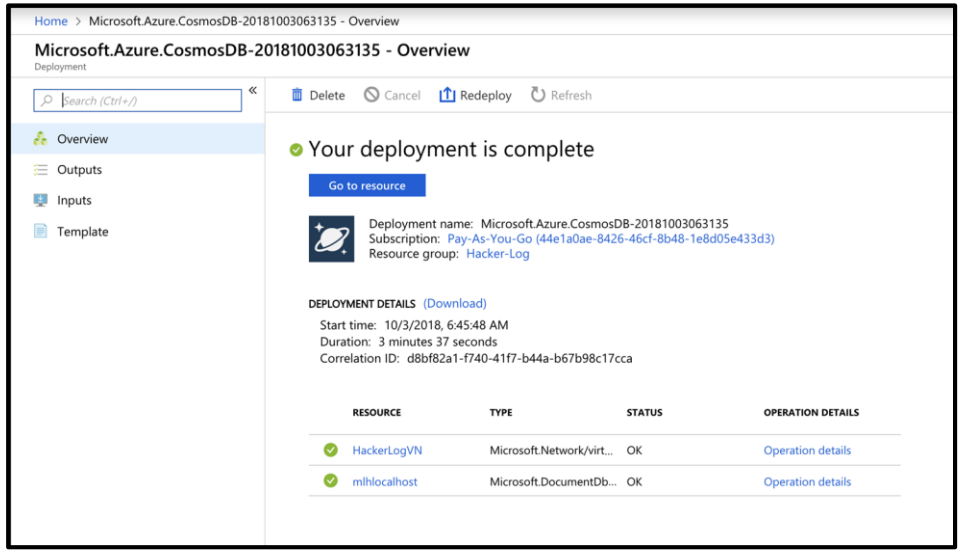
Create a Cosmos Resource

Instructions

15. Click **Create** and wait for your database to deploy!



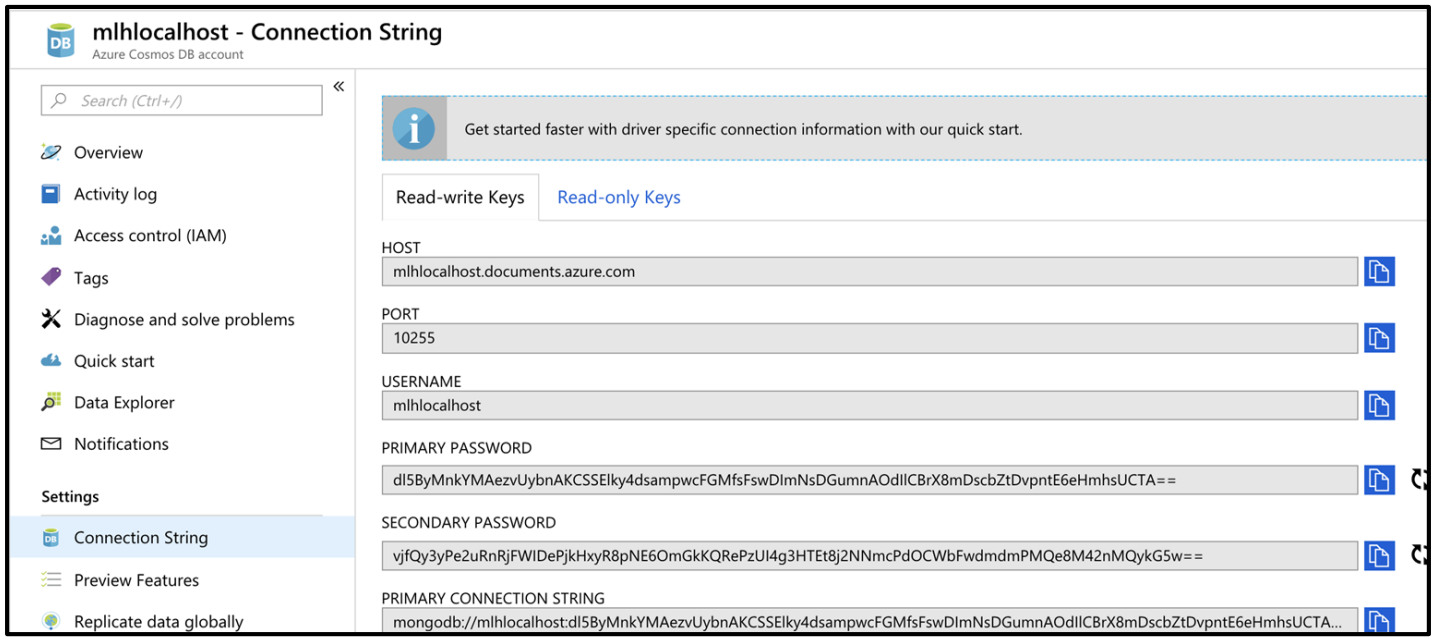
16. It may take up to 10 minutes for your resource to be deployed. When it's done, select **Go to resource**.



Create a Cosmos Resource

Instructions

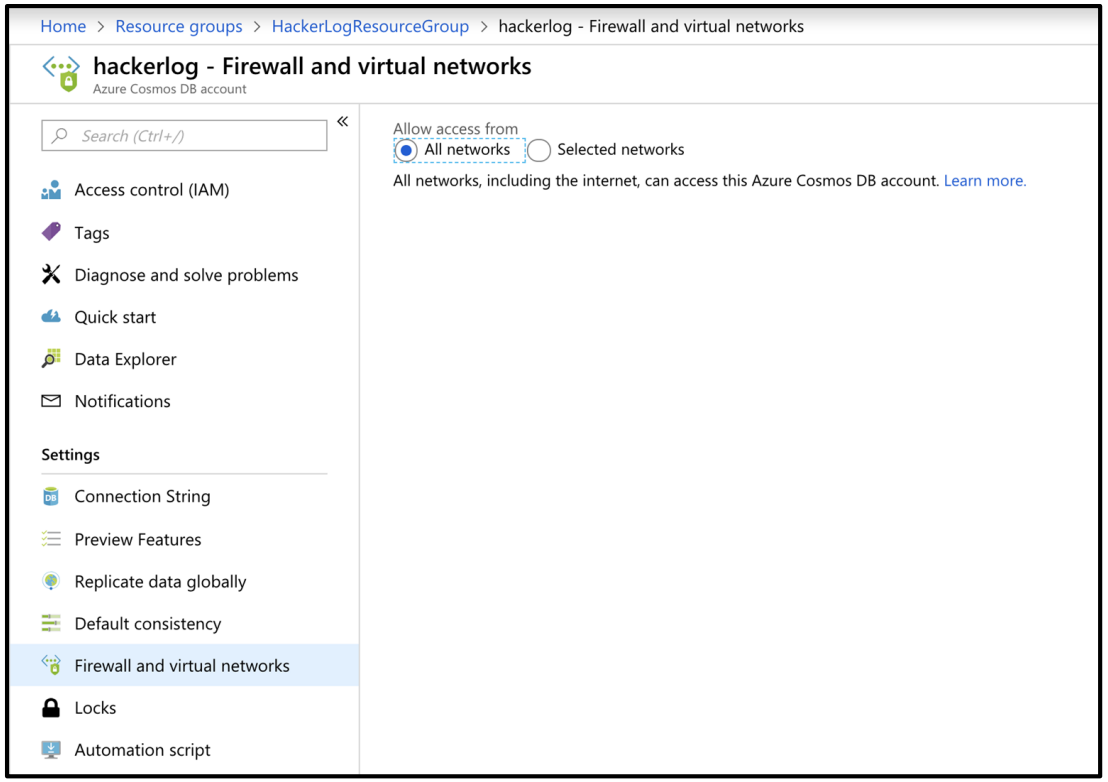
17. Open the **Connection String** menu, under the **Settings** category.
18. Copy **PRIMARY CONNECT STRING**. You'll need it later.



Create a Cosmos Resource

Instructions

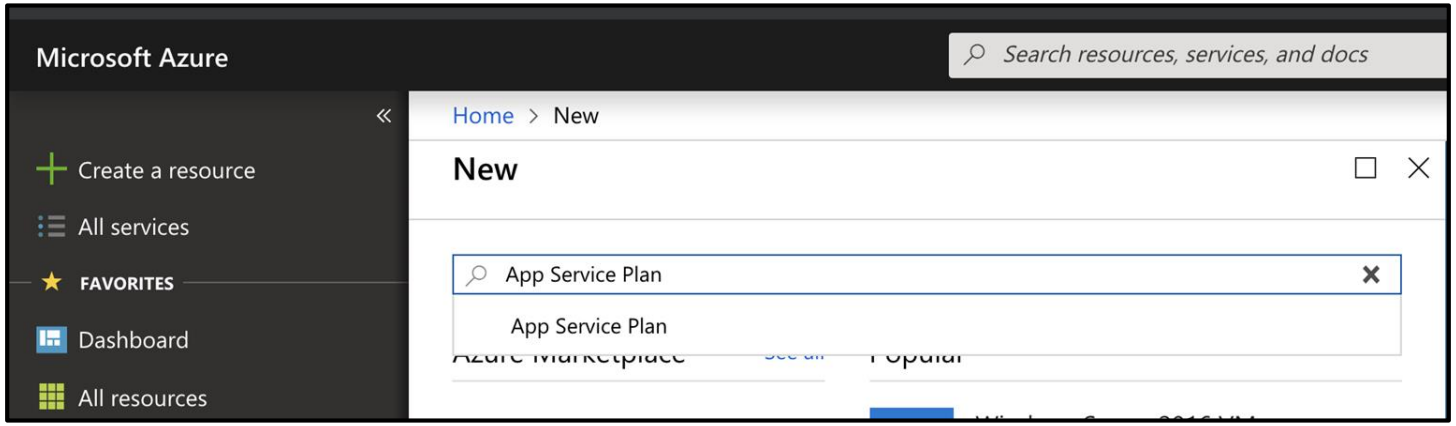
19. Go to the **Firewall and virtual networks** tab under the same **Settings** category.
20. For the purposes of this workshop, select **All networks** under **Allow access from:** (you wouldn't do this in a production app).
21. Save!



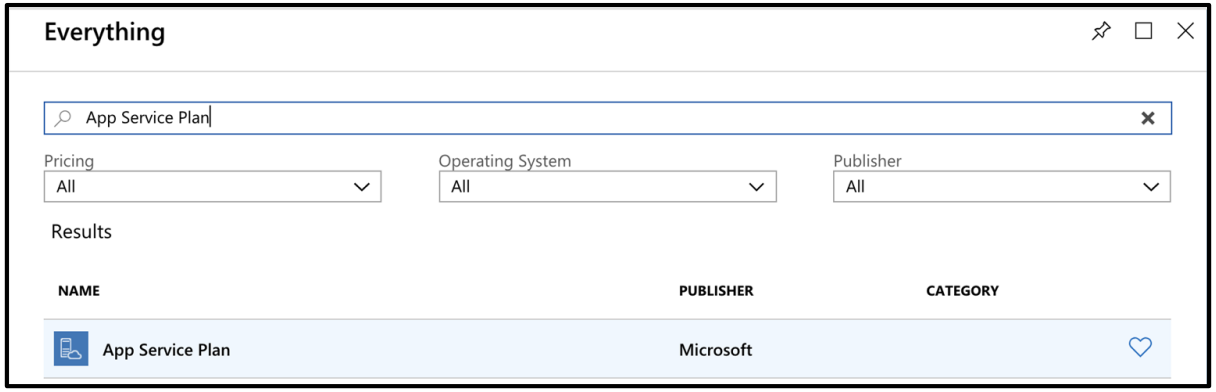
Now that your database is set up, we'll create something called an App Service Plan.

Create an App Service Plan

1. Return to portal.azure.com
2. Select **Create a Resource** then search for "App Service Plan".



3. Select **App Service Plan** then select **Create** in the panel that opens.



Create an App Service Plan

Instructions

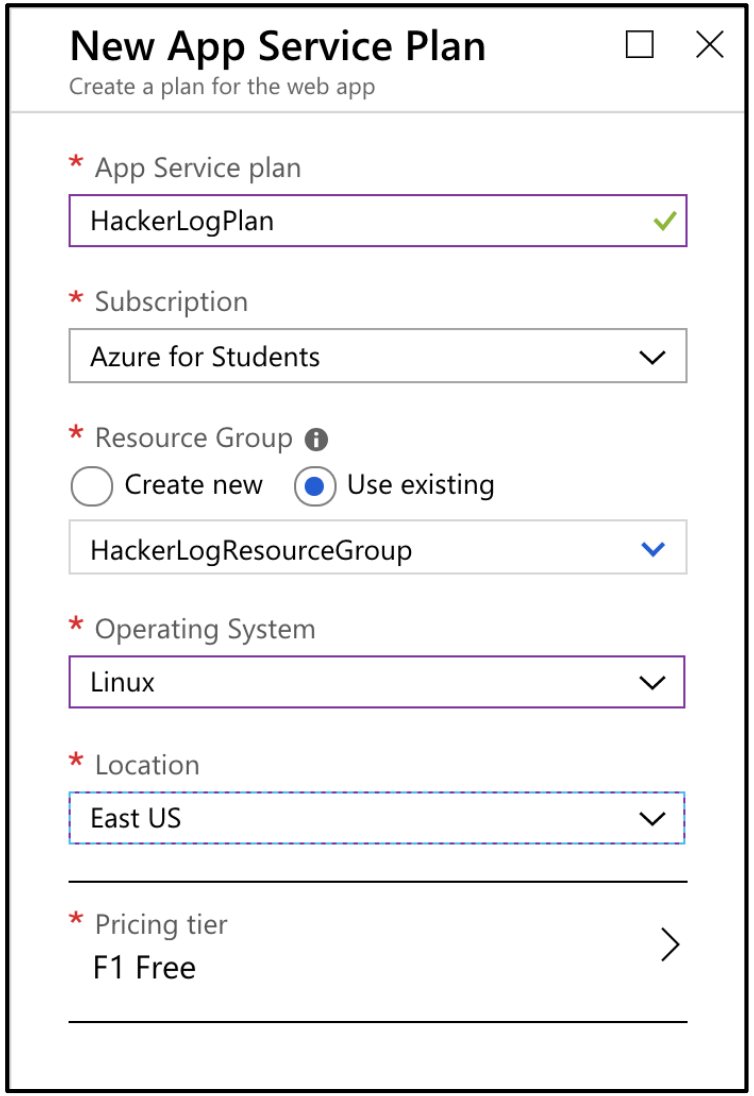
4. In the panel that opens, name your service plan **HackerLogPlan**.

5. Select the Resource Group you made previously.

6. Select **Linux** as your operating system.

7. Select **East US**.

8. Select **Pricing tier**.

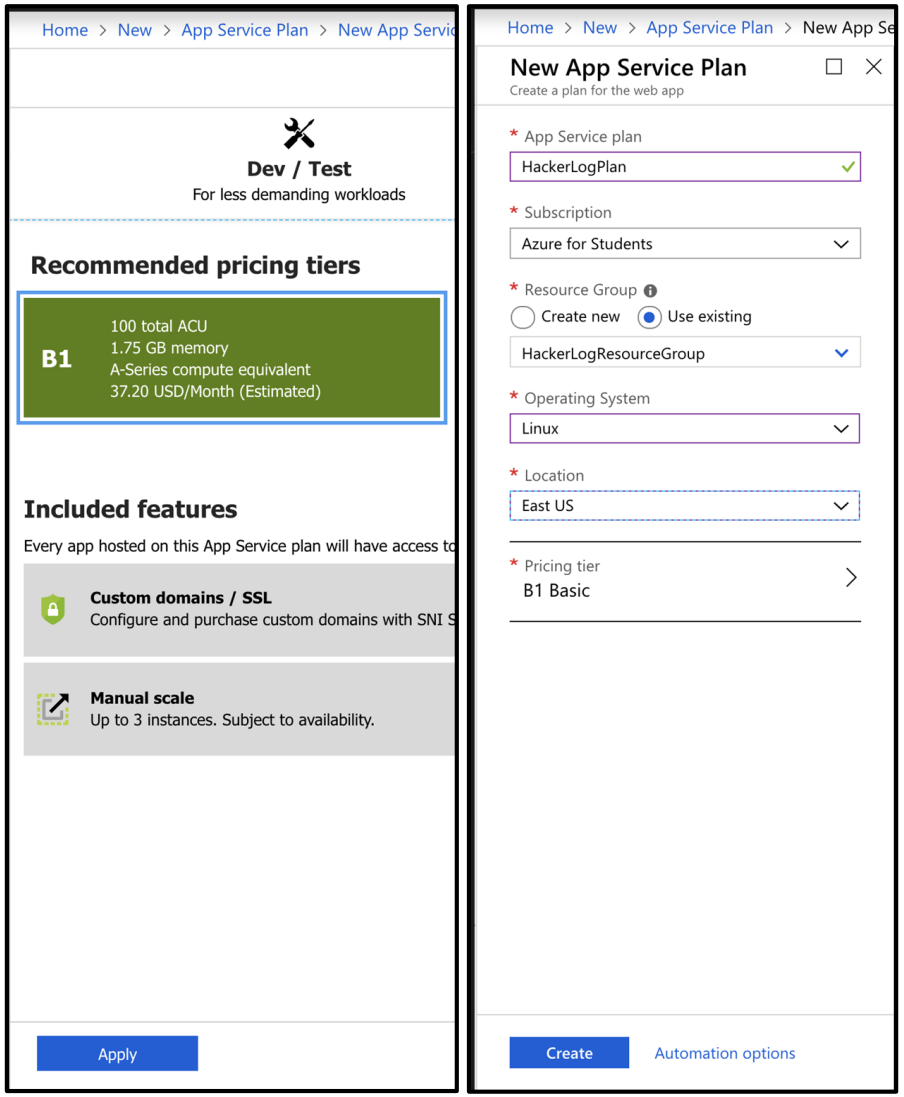


Create an App Service Plan

Instructions

9. Select **Dev/Test**.
10. Select **B1**.
11. Select **Apply**.
12. When the previous screen returns, select **Create**.

    Wait a minute or two for that to finish deploying, then we'll create the resource for the HackerLog Web App!

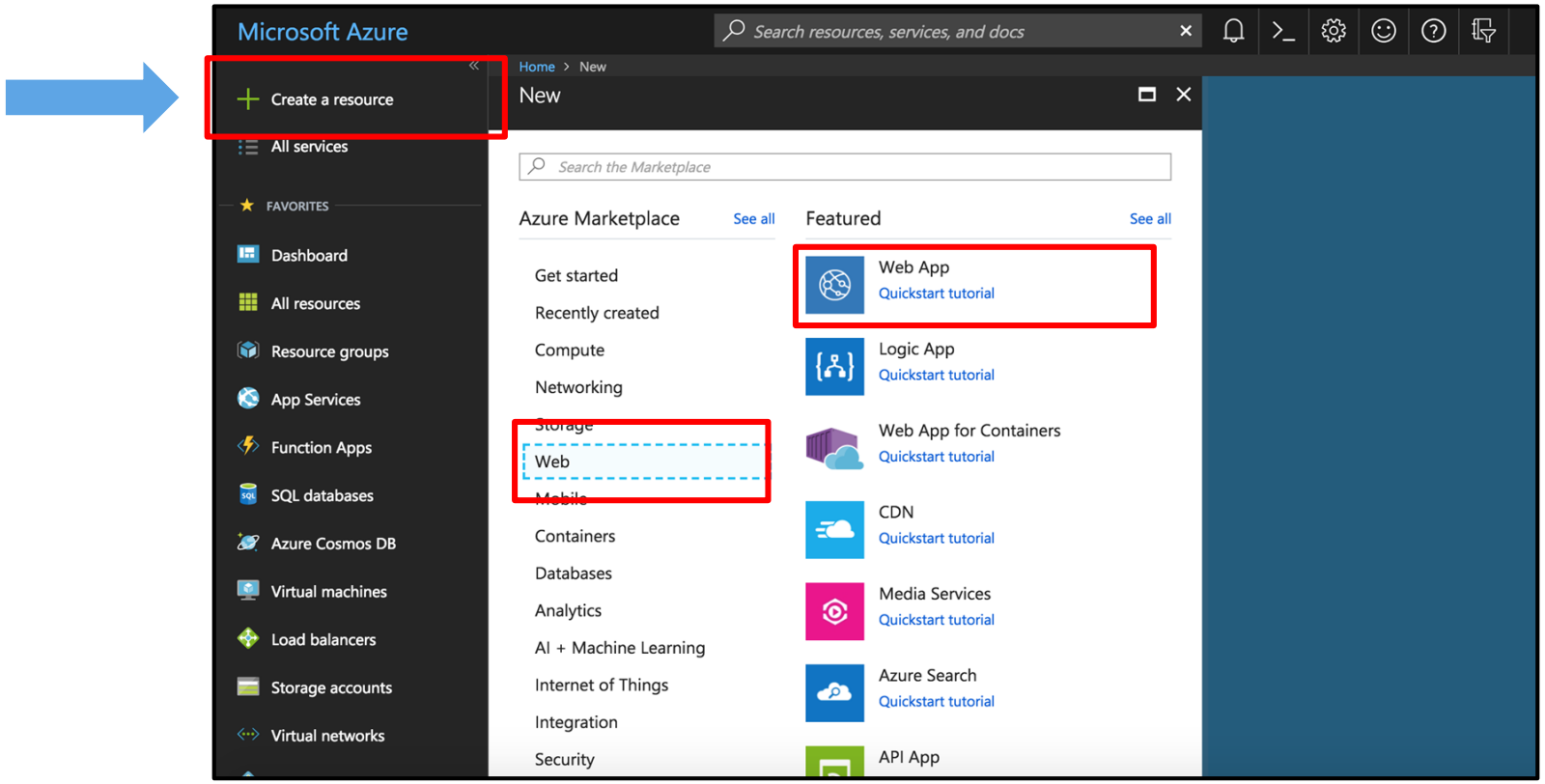


Finally, let's deploy!

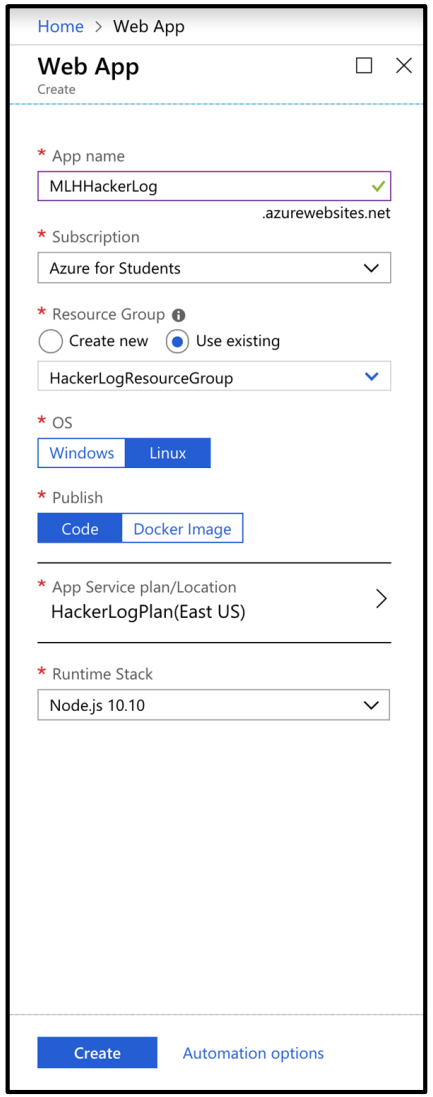
Deploy Your App to Microsoft Azure

Instructions

1. Navigate to your Azure dashboard and select **Create a resource** then **Web** then **Web App**.



Deploy Your App to Microsoft Azure



Instructions

2. In the panel that opens, name your app something unique.

3. Select the Resource Group you made previously.

4. Select **Linux** as your operating system.

5. Select **Node.js 10.10** as your Runtime Stack.

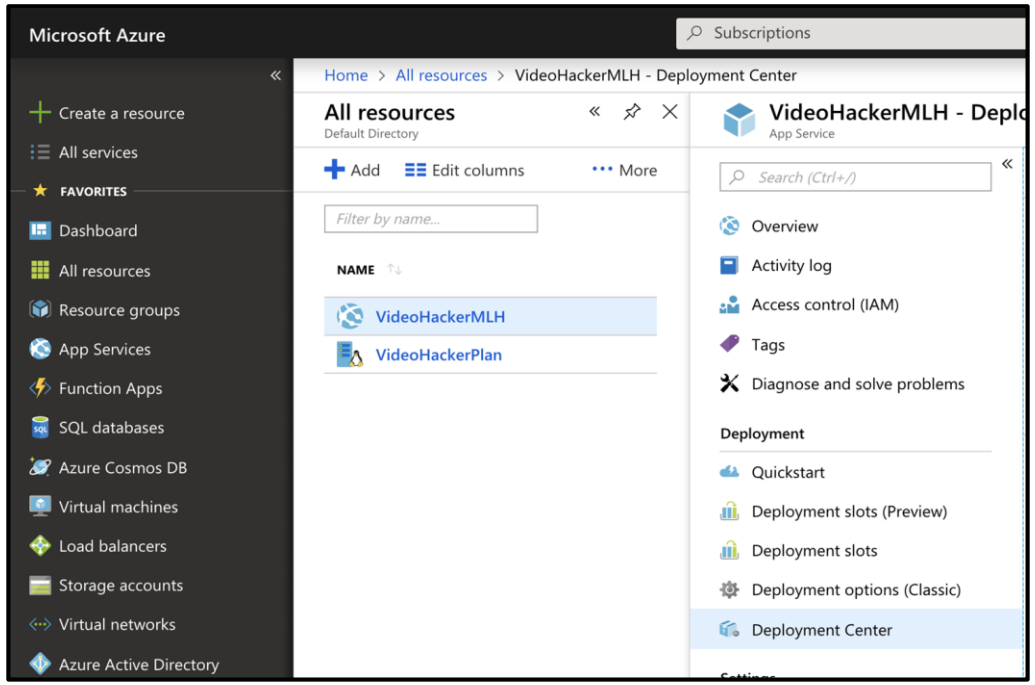
6. Click **Create** at the bottom of the screen.

Wait a moment for the app to deploy, then follow the next steps.

Deploy Your App to Microsoft Azure

Instructions

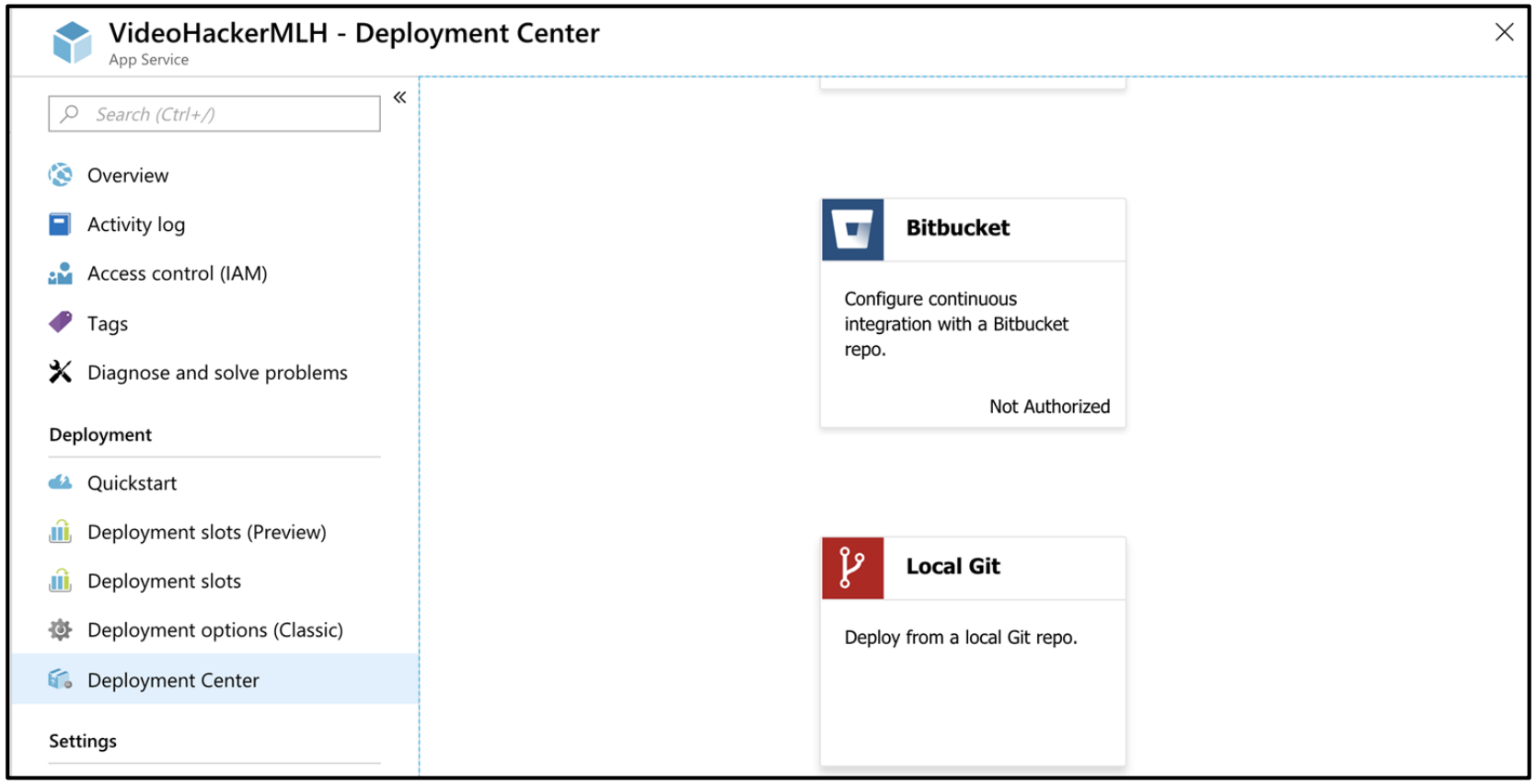
1. On the left side of your Azure portal, select **All resources**.
2. Click the App Service you just made.
3. Select **Deployment Center**.



Deploy Your App to Microsoft Azure

Instructions

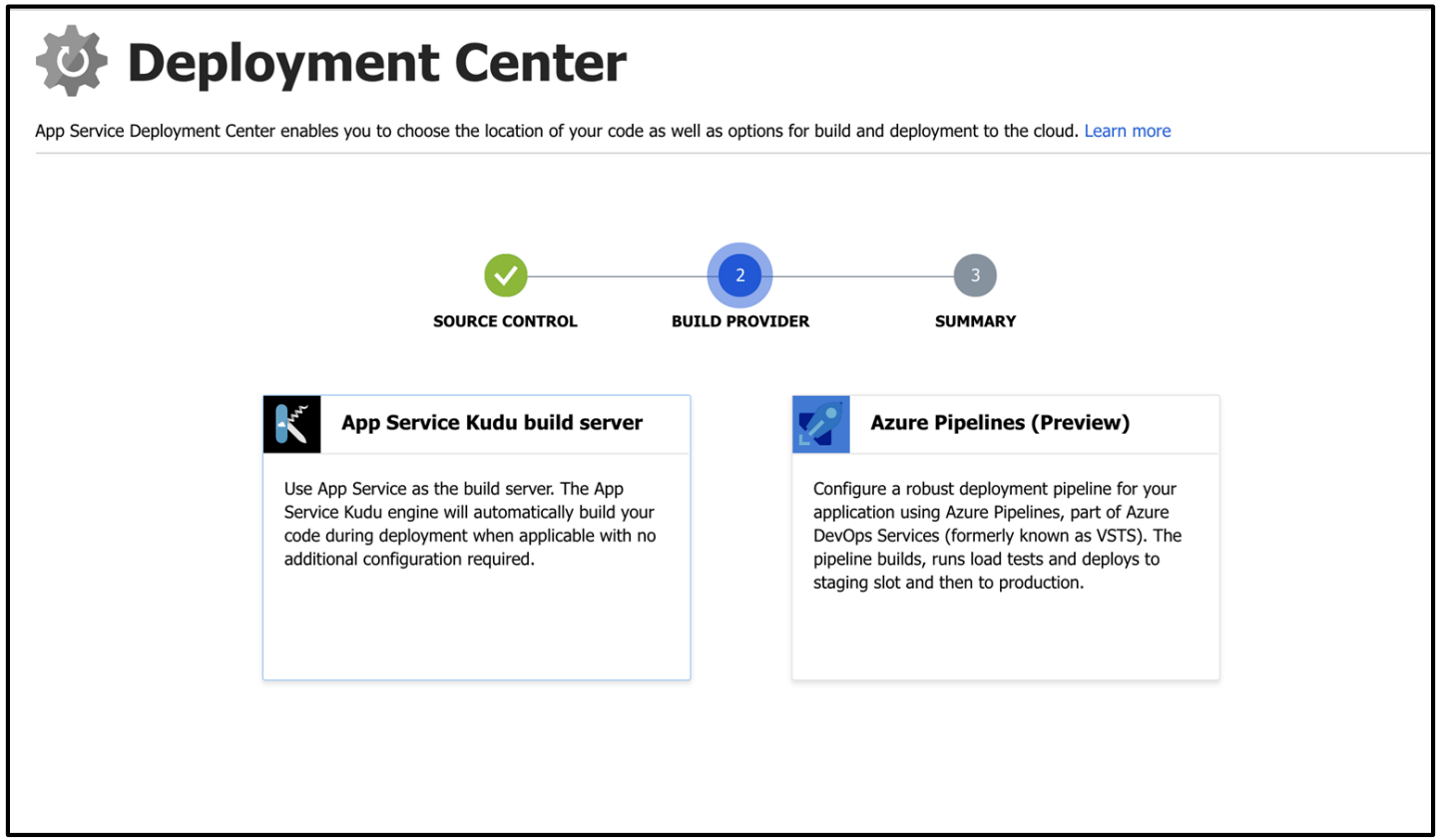
4. Scroll to select **Local Git**. Select **Continue**.



Deploy Your App to Microsoft Azure

Instructions

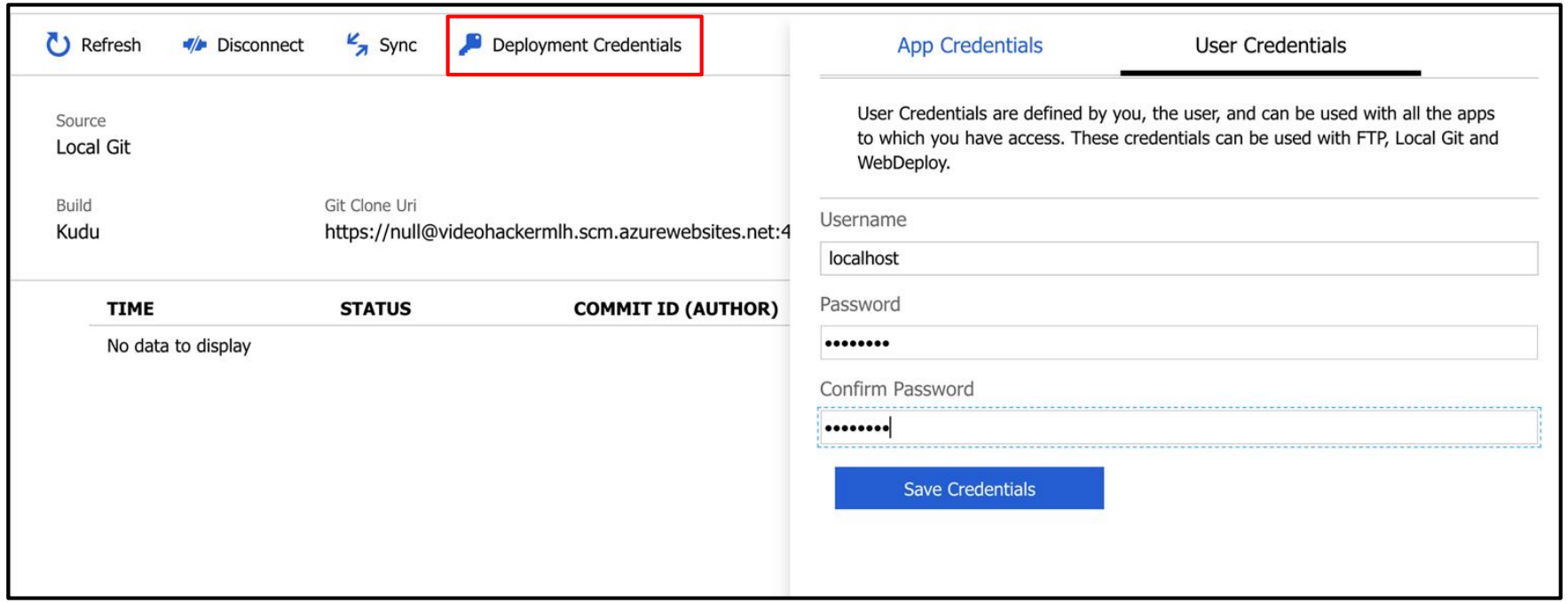
5. Select App Service Kudu build server then select **Continue** then **Finish**.



Deploy Your App to Microsoft Azure

Instructions

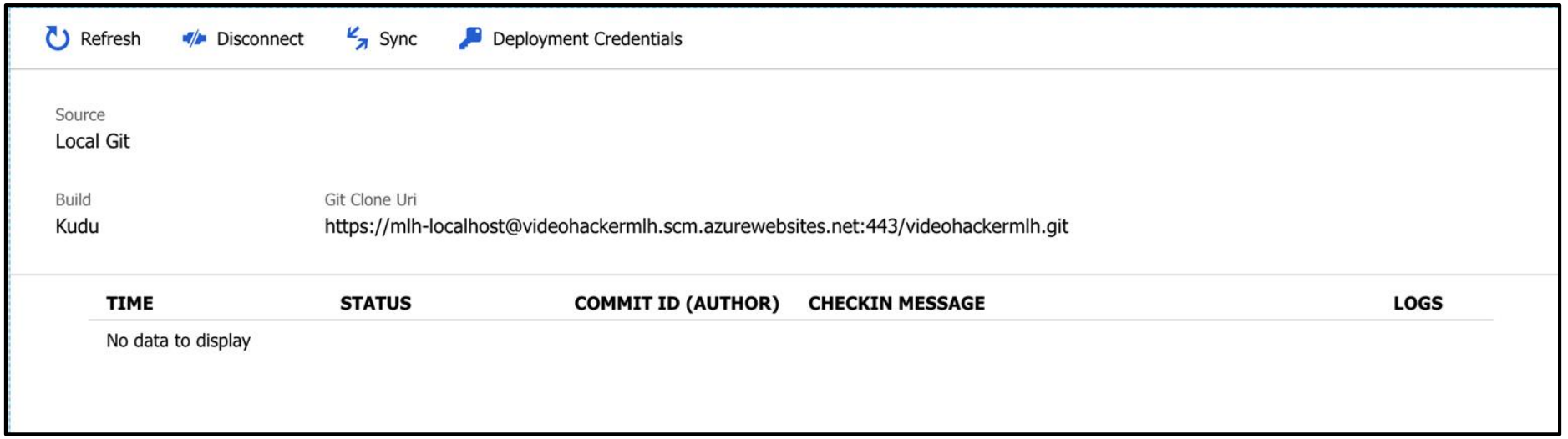
6. Select **Deployment Credentials** then **User Credentials**.
7. Create a Username and Password.



Deploy Your App to Microsoft Azure

Instructions

8. You will know the previous step was successful if the Uri has `<your-new-username>@<rest of website>`.

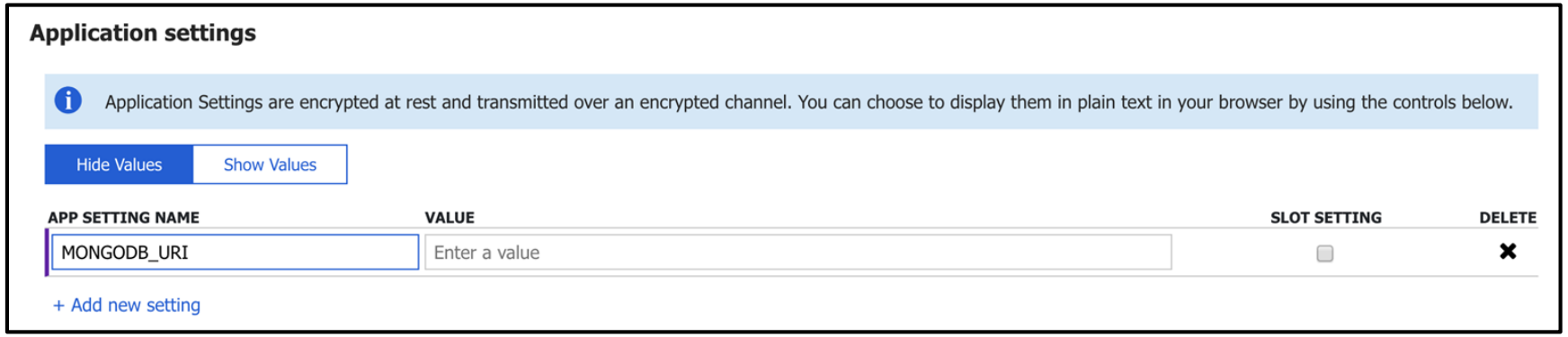


Deploy Your App to Microsoft Azure

Instructions

9. Select **Application settings**.
10. Click **+ Add new setting**.
11. The name of the new setting is "MONGODB_URI"
12. The value of the new setting is the PRIMARY CONNECTION STRING you copied when you made the CosmosDB resource. This is how your app and the DB can talk to each other. It will look like this:

```
mongodb://<username>:<pimary_password>@<hostname>:<port>/?ssl=true&replicaSet=globaldb
```



Deploy Your App to Microsoft Azure

Instructions

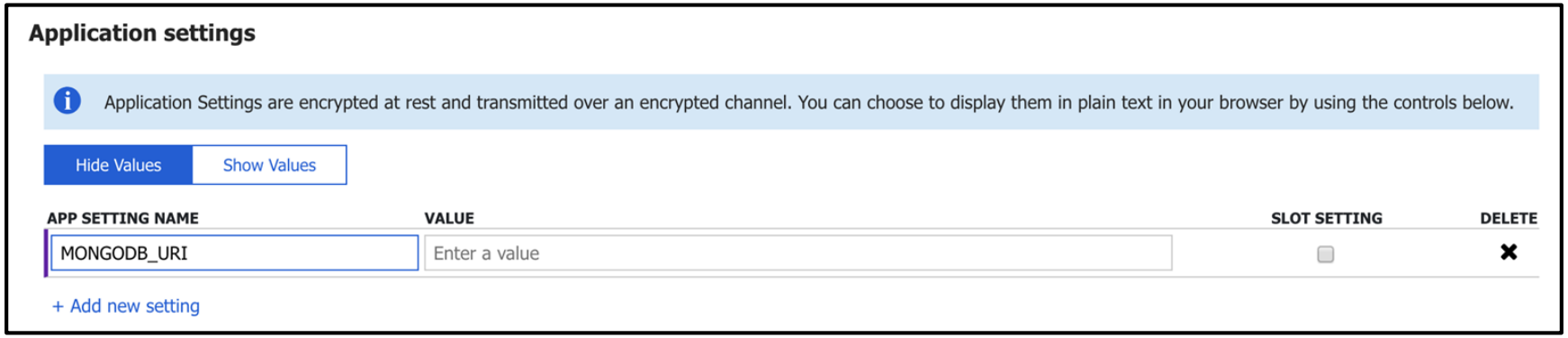
13. This part is a little tricky, so ready very carefully. Towards the end of your connection string, after the **`/`** and before the **`?`**, include the word **`admin`**.

Before:

```
mongodb://<username>:<pimary_password>@<hostname>:<port>/?ssl=true&replicaSet=globaldb
```

After:

```
mongodb://<username>:<pimary_password>@<hostname>:<port>/admin?ssl=true&replicaSet=globaldb
```



You just:

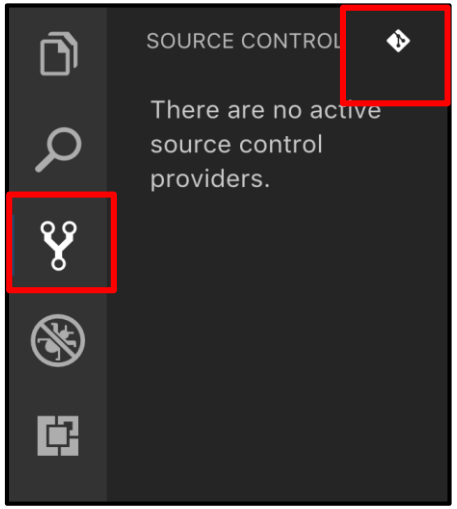
1. **Created a database on Microsoft Azure.**
2. **Created an App Service Plan.**
3. **Created a Node app environment to deploy your app to.**

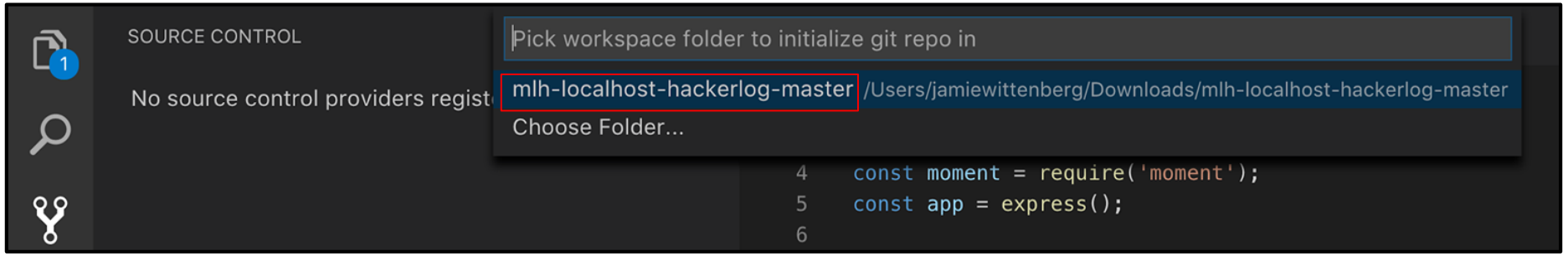
Head back to Visual Studio Code for the rest!

Deploy Your App to Microsoft Azure

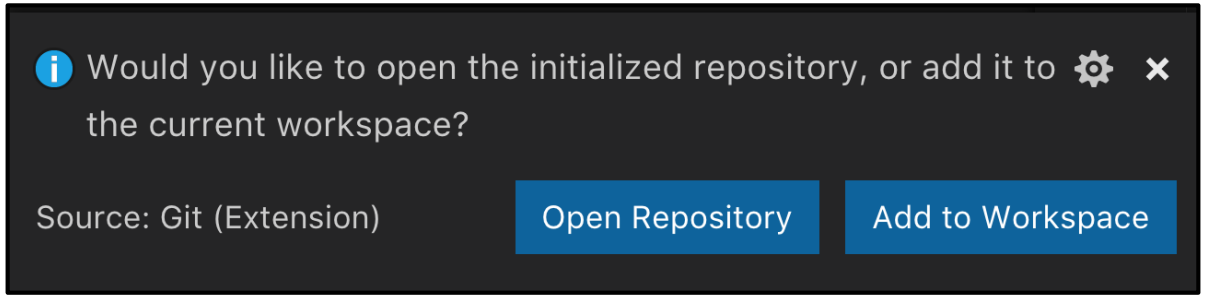
1. In Visual Studio Code, click the source control symbol.
2. Then, click the symbol next to `SOURCE CONTROL`.
3. Be sure that the folder is **`mlh-localhost-hackerlog-master`** and click it.

   Note: Make sure the folder is NOT your Desktop or entire Documents folder.

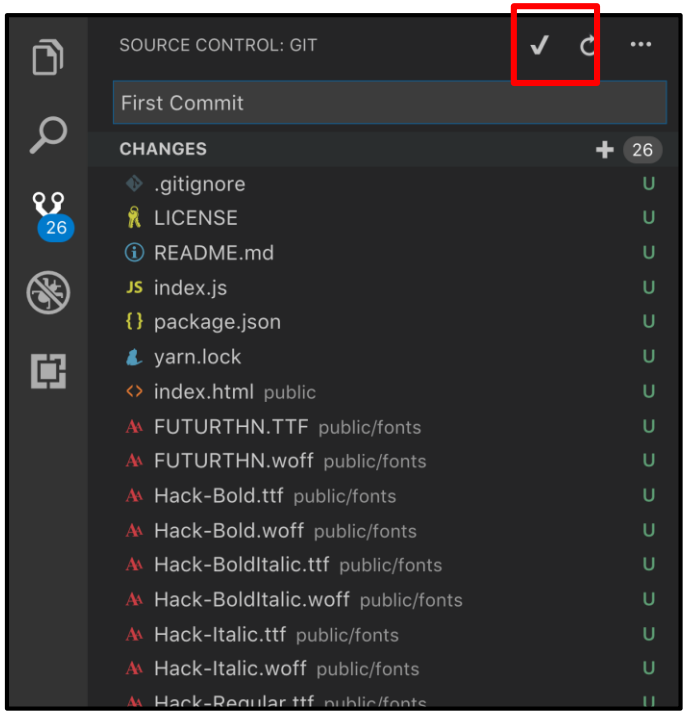




Deploy Your App to Microsoft Azure

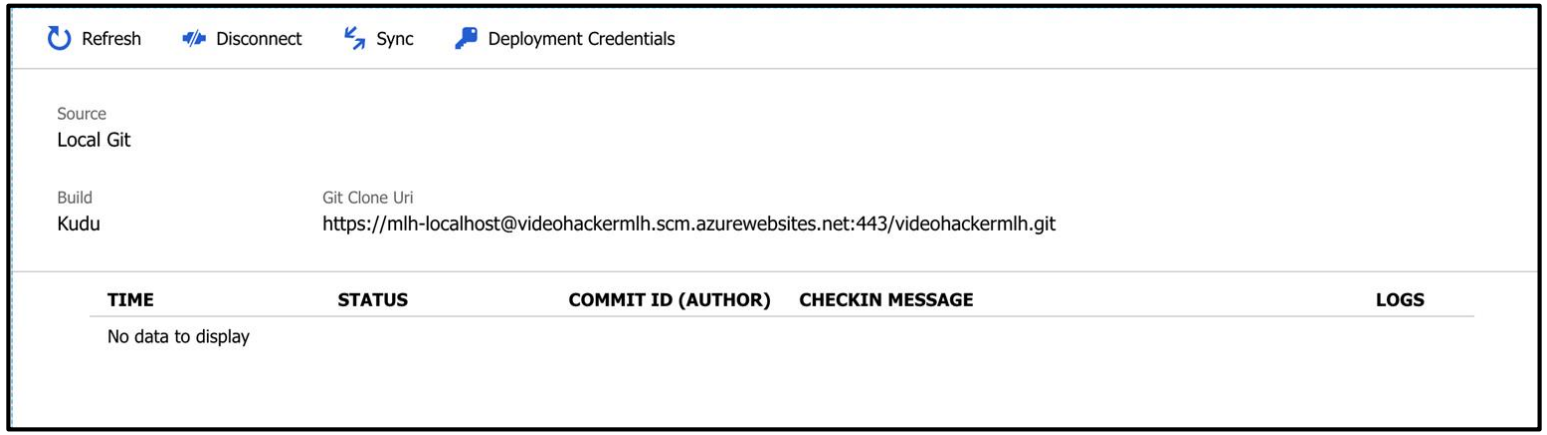


4. Select **Add to Workspace** if prompted.
5. Type "First Commit" above CHANGES.
6. Click the checkmark.
7. Click **Yes** if you get a modal asking about staging changes.

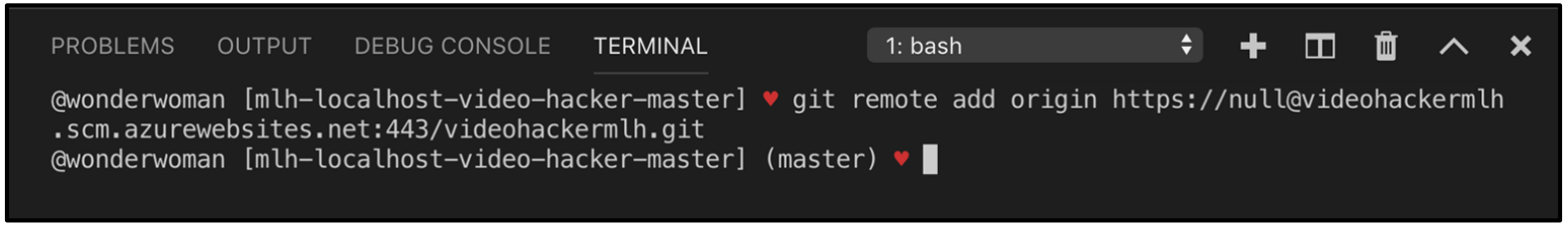


Deploy Your App to Microsoft Azure

8. Return to your Azure portal. In your Deployment Center tab, copy the Git clone Uri.

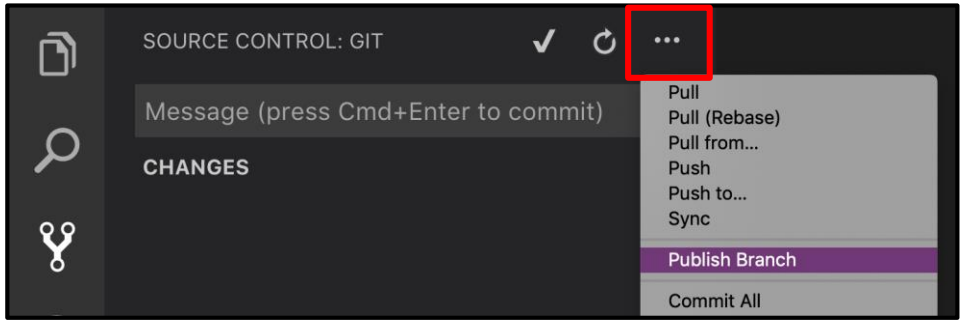


8. In your terminal, type **`git remote add origin`** and paste what you just copied. This is how you tell Git where to send your code.

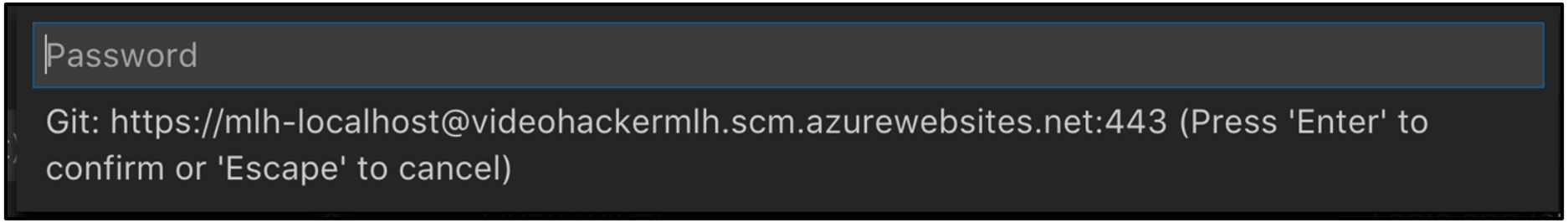


Deploy Your App to Microsoft Azure

10. Click the ellipses **(. . .)** near SOURCE CONTROL.
11. Select **Publish Branch**.



10. Enter the Password you created when prompted.



Deploy Your App to Microsoft Azure

13. Back in Azure Deployment Centre, you can see that your deploy is in progress!

Deploy Your App to Microsoft Azure

14. When the build has finished, select Overview.
15. Click the URL for your web app, and try it live!