

Table of Contents

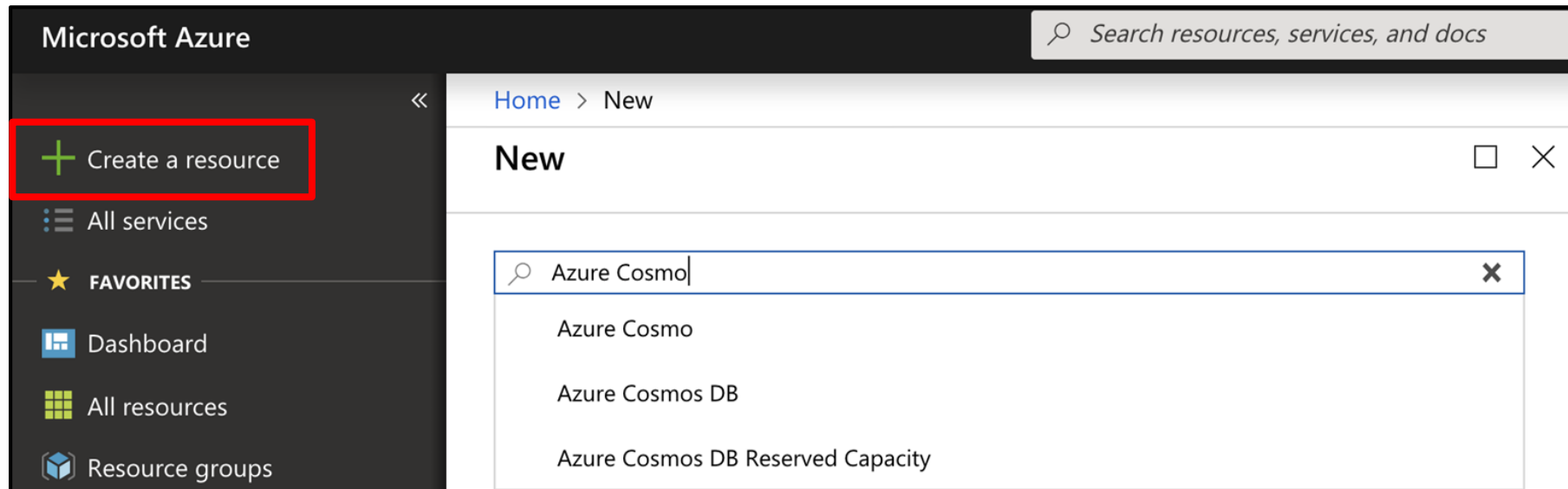
1. Introduction to Web Development
2. Microsoft Azure and Visual Studio
3. Node.js and Express
4.  Deploy Your Web App!
5. Review & Quiz
6. Next Steps

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**.

The screenshot shows the 'Create Azure Cosmos DB Account' page in the Azure portal. The breadcrumb trail at the top is 'Home > New > Marketplace > Everything > Azure Cosmos DB > Create Azure Cosmos DB Account'. The page has tabs for 'Basics', 'Network', 'Tags', and 'Summary', with 'Basics' selected. A descriptive paragraph about Azure Cosmos DB is present. Under the 'PROJECT DETAILS' section, the 'Subscription' is set to 'Pay-As-You-Go (44e1a0ae-8426-46cf-8b48-1e8d05e433d3)'. The 'Resource Group' dropdown is set to 'cloud-shell-storage-eastus', with a 'Create new' link below it. A modal dialog is open, explaining that a resource group is a container for related resources. It has a 'Name' field containing 'Hacker-Log' with a green checkmark, and 'OK' and 'Cancel' buttons. In the background, the 'INSTANCE DETAILS' section is visible, showing 'Account Name' as 'documents.azure.com', 'API' as '...', 'Location' as '...', 'Geo-Redundancy' as '...', and 'Multi-region Writes' as 'Enable'.

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**.

[Basics](#) [Network](#) [Tags](#) [Review + create](#)

Azure Cosmos DB is a fully managed globally distributed, multi-model database service, transparently replicating your data across any number of Azure regions. You can elastically scale throughput and storage, and take advantage of fast, single-digit-millisecond data access using the API of your choice backed by 99.999 SLA. [learn more](#)

PROJECT DETAILS

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

* Subscription

Pay-As-You-Go (126a7d47-8f7e-475d-8d21-016604744043) ▼

* Resource Group

(New) HackerLogResourceGroup ▼

[Create new](#)

INSTANCE DETAILS

* Account Name

mlh-localhost ✓

[documents.azure.com](#)

* API ⓘ

Azure Cosmos DB for MongoDB API ▼

* Location

East US ▼

Geo-Redundancy ⓘ

Enable Disable

Multi-region Writes ⓘ

Enable Disable

Create a Cosmos Resource

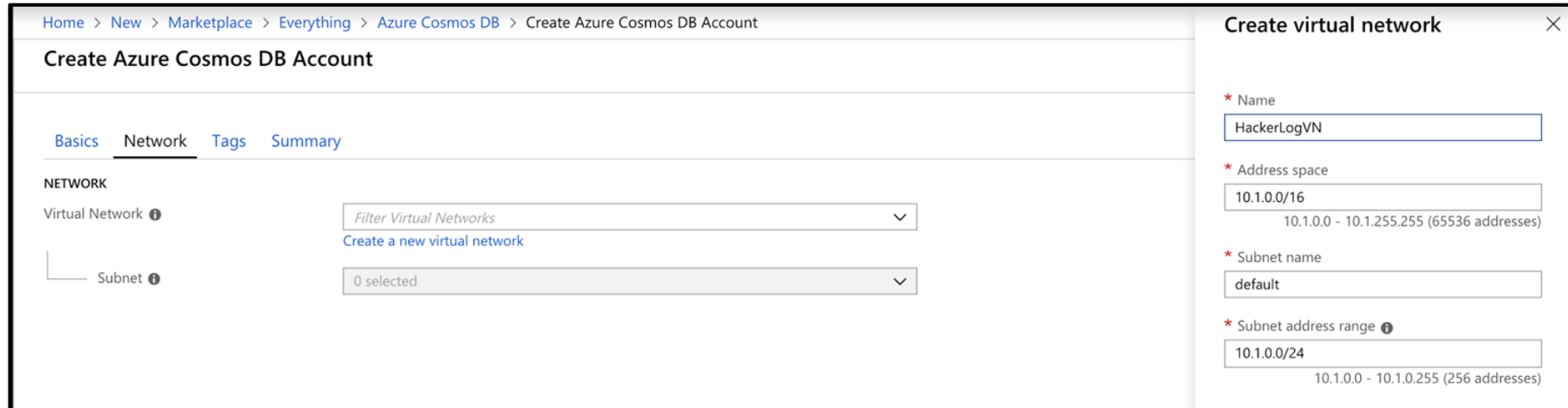
Instructions

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

12. Name it HackerLogVN.

13. Select **OK**.

14. Click **Review + create**

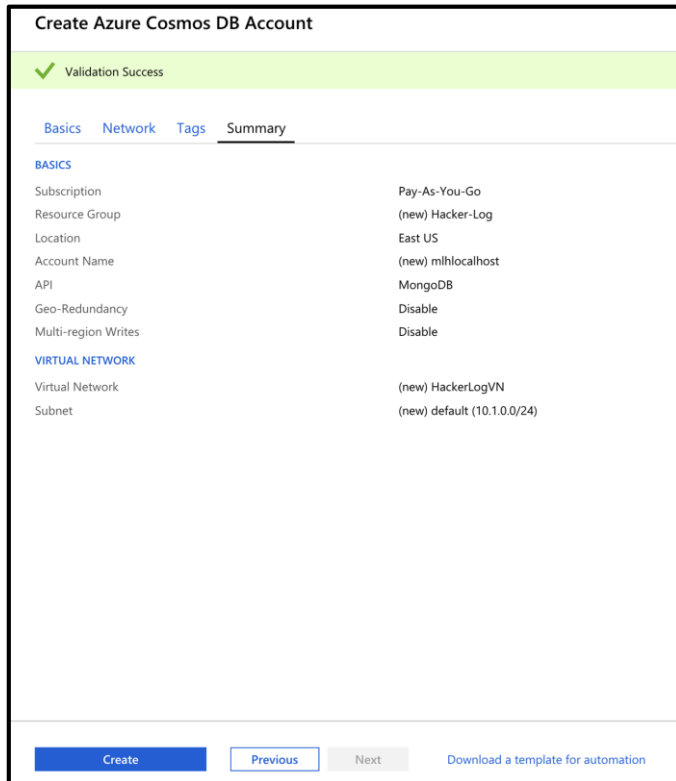


The screenshot shows the Azure portal interface for creating a Cosmos DB account. The breadcrumb trail at the top reads: Home > New > Marketplace > Everything > Azure Cosmos DB > Create Azure Cosmos DB Account. The main heading is 'Create Azure Cosmos DB Account'. Below this, there are four tabs: Basics, Network, Tags, and Summary. The 'Network' tab is currently selected. Under the 'NETWORK' section, there is a 'Virtual Network' field with a dropdown menu showing 'Filter Virtual Networks' and a link to 'Create a new virtual network'. Below this is a 'Subnet' field with a dropdown menu showing '0 selected'. On the right side of the screen, a 'Create virtual network' dialog box is open. It contains the following fields: 'Name' (HackerLogVN), 'Address space' (10.1.0.0/16, with a note '10.1.0.0 - 10.1.255.255 (65536 addresses)'), 'Subnet name' (default), and 'Subnet address range' (10.1.0.0/24, with a note '10.1.0.0 - 10.1.0.255 (256 addresses)').

Create a Cosmos Resource

Instructions

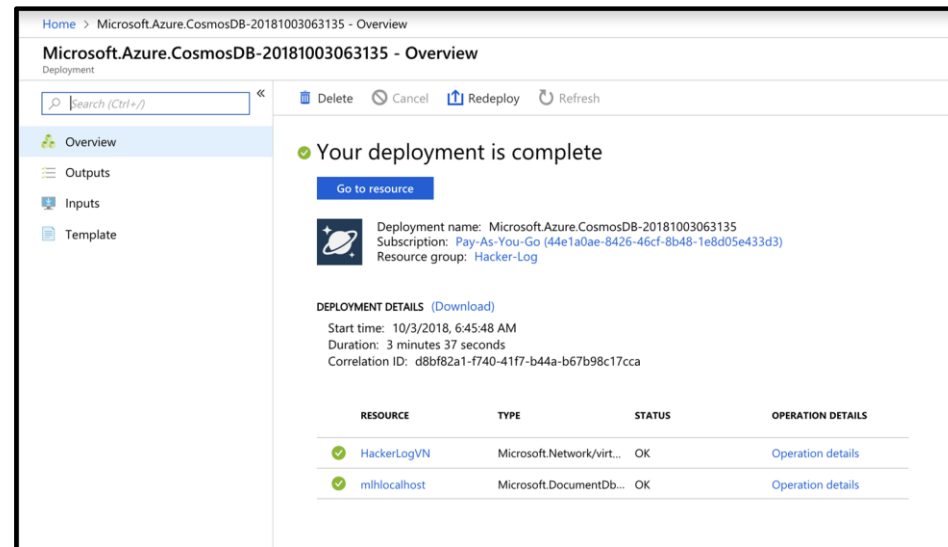
15. Click **Create** and wait for your database to deploy!



The screenshot shows the 'Create Azure Cosmos DB Account' wizard in the Azure portal, specifically the 'Summary' tab. A green banner at the top indicates 'Validation Success'. The 'BASIC' section lists the following configuration: Subscription (Pay-As-You-Go), Resource Group ((new) Hacker-Log), Location (East US), Account Name ((new) mlhlocalhost), API (MongoDB), Geo-Redundancy (Disable), Multi-region Writes (Disable), Virtual Network ((new) HackerLogVN), and Subnet ((new) default (10.1.0.0/24)). The 'VIRTUAL NETWORK' section is also visible. At the bottom, there are buttons for 'Create', 'Previous', 'Next', and a link to 'Download a template for automation'.

Section	Property	Value
BASIC	Subscription	Pay-As-You-Go
	Resource Group	((new) Hacker-Log)
	Location	East US
	Account Name	((new) mlhlocalhost)
	API	MongoDB
	Geo-Redundancy	Disable
VIRTUAL NETWORK	Virtual Network	((new) HackerLogVN)
	Subnet	((new) default (10.1.0.0/24))

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



The screenshot shows the 'Overview' page for the resource 'Microsoft.Azure.CosmosDB-20181003063135'. It displays deployment details and a table of resources. A green checkmark indicates 'Your deployment is complete'. A 'Go to resource' button is available. The deployment details show a start time of 10/3/2018, 6:45:48 AM, a duration of 3 minutes 37 seconds, and a correlation ID. The table below lists the resources deployed as part of the Cosmos DB account.

Deployment Details:

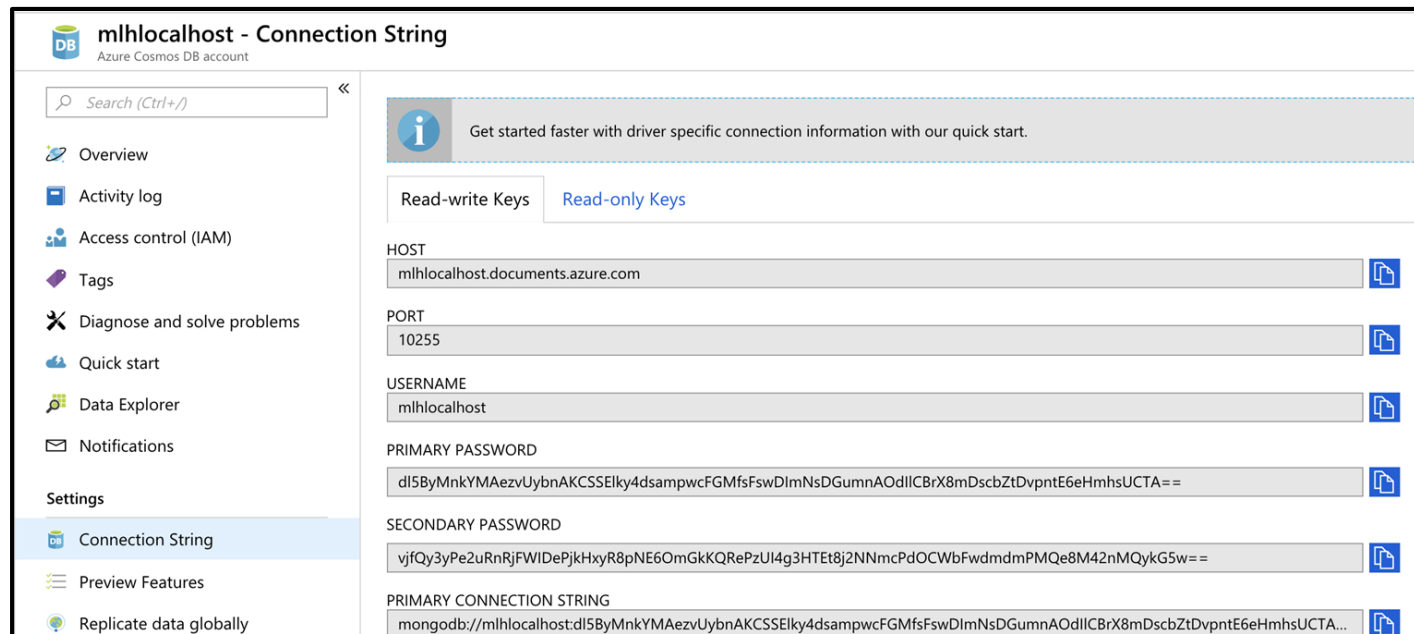
- Deployment name: Microsoft.Azure.CosmosDB-20181003063135
- Subscription: Pay-As-You-Go (44e1a0ae-8426-46cf-8b48-1e8d05e433d3)
- Resource group: Hacker-Log
- Start time: 10/3/2018, 6:45:48 AM
- Duration: 3 minutes 37 seconds
- Correlation ID: d8bf82a1-f740-41f7-b44a-b67b98c17cca

RESOURCE	TYPE	STATUS	OPERATION DETAILS
HackerLogVN	Microsoft.Network/virt...	OK	Operation details
mlhlocalhost	Microsoft.DocumentDb...	OK	Operation details

Create a Cosmos Resource

Instructions

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



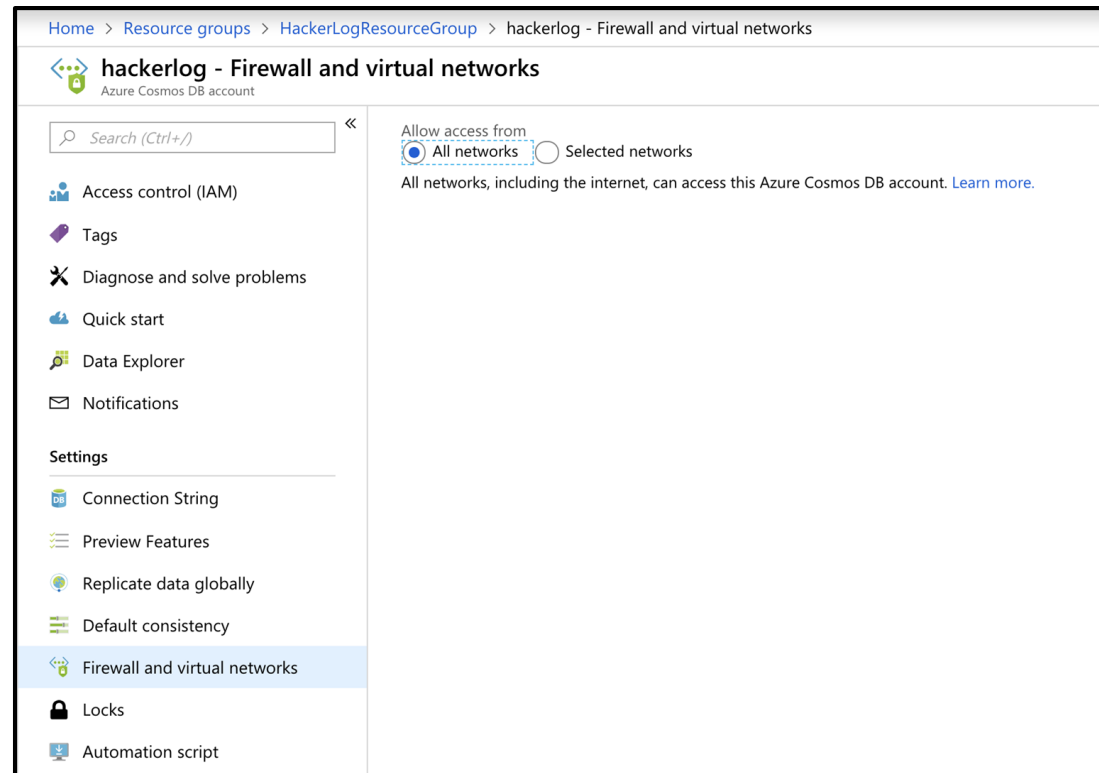
Create a Cosmos Resource

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).

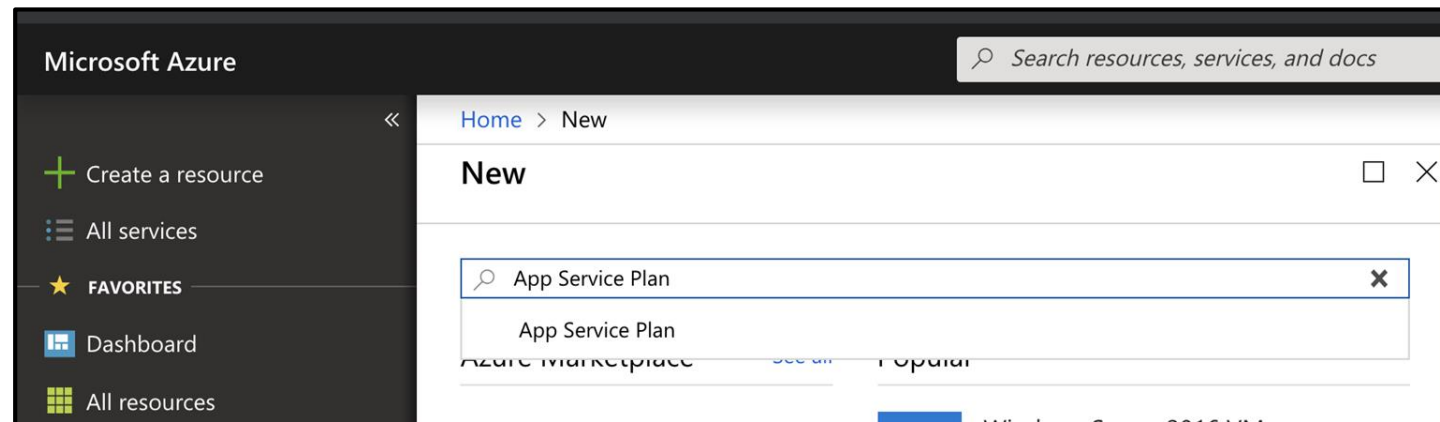
21. Save!



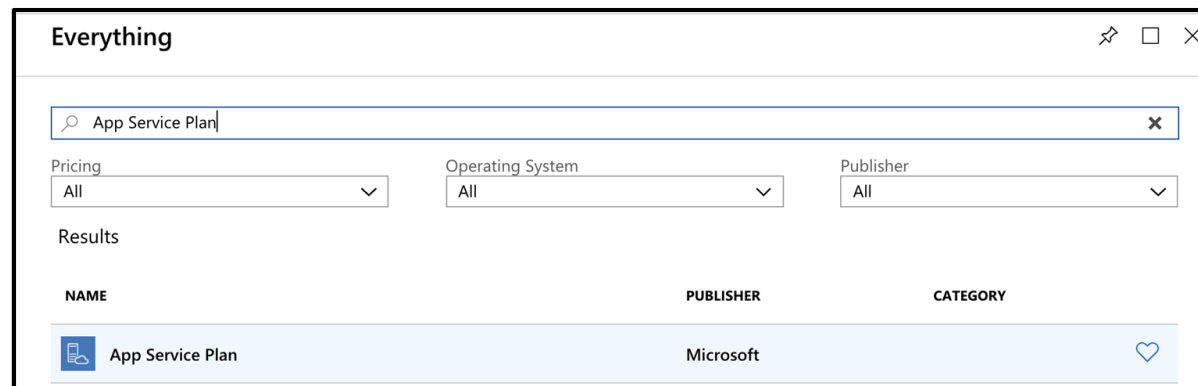
**Now that your database is set up, we'll
create something called an App Service
Plan.**

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**.

New App Service Plan

Create a plan for the web app

* App Service plan

HackerLogPlan ✓

* Subscription

Azure for Students ▼

* Resource Group ⓘ

☐ Create new ☒ Use existing

HackerLogResourceGroup ▼

* Operating System

Linux ▼

* Location

East US ▼

* Pricing tier

F1 Free >

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!

The image displays two side-by-side screenshots of the Azure portal's 'New App Service Plan' configuration page. The left screenshot shows the 'Dev / Test' pricing tier selection screen, where the 'B1' tier is highlighted in a green box. The right screenshot shows the 'New App Service Plan' configuration page with various settings filled out: App Service plan is 'HackerLogPlan', Subscription is 'Azure for Students', Resource Group is 'HackerLogResourceGroup', Operating System is 'Linux', Location is 'East US', and Pricing tier is 'B1 Basic'. The 'Create' button is visible at the bottom of the right screenshot.

Home > New > App Service Plan > New App Service Plan

Dev / Test
For less demanding workloads

Recommended pricing tiers

B1
100 total ACU
1.75 GB memory
A-Series compute equivalent
37.20 USD/Month (Estimated)

Included features
Every app hosted on this App Service plan will have access to the following features:

- Custom domains / SSL**
Configure and purchase custom domains with SNI support.
- Manual scale**
Up to 3 instances. Subject to availability.

New App Service Plan
Create a plan for the web app

- * App Service plan
HackerLogPlan ✓
- * Subscription
Azure for Students
- * Resource Group ⓘ
☐ Create new ☒ Use existing
HackerLogResourceGroup
- * Operating System
Linux
- * Location
East US
- * Pricing tier
B1 Basic >

Apply

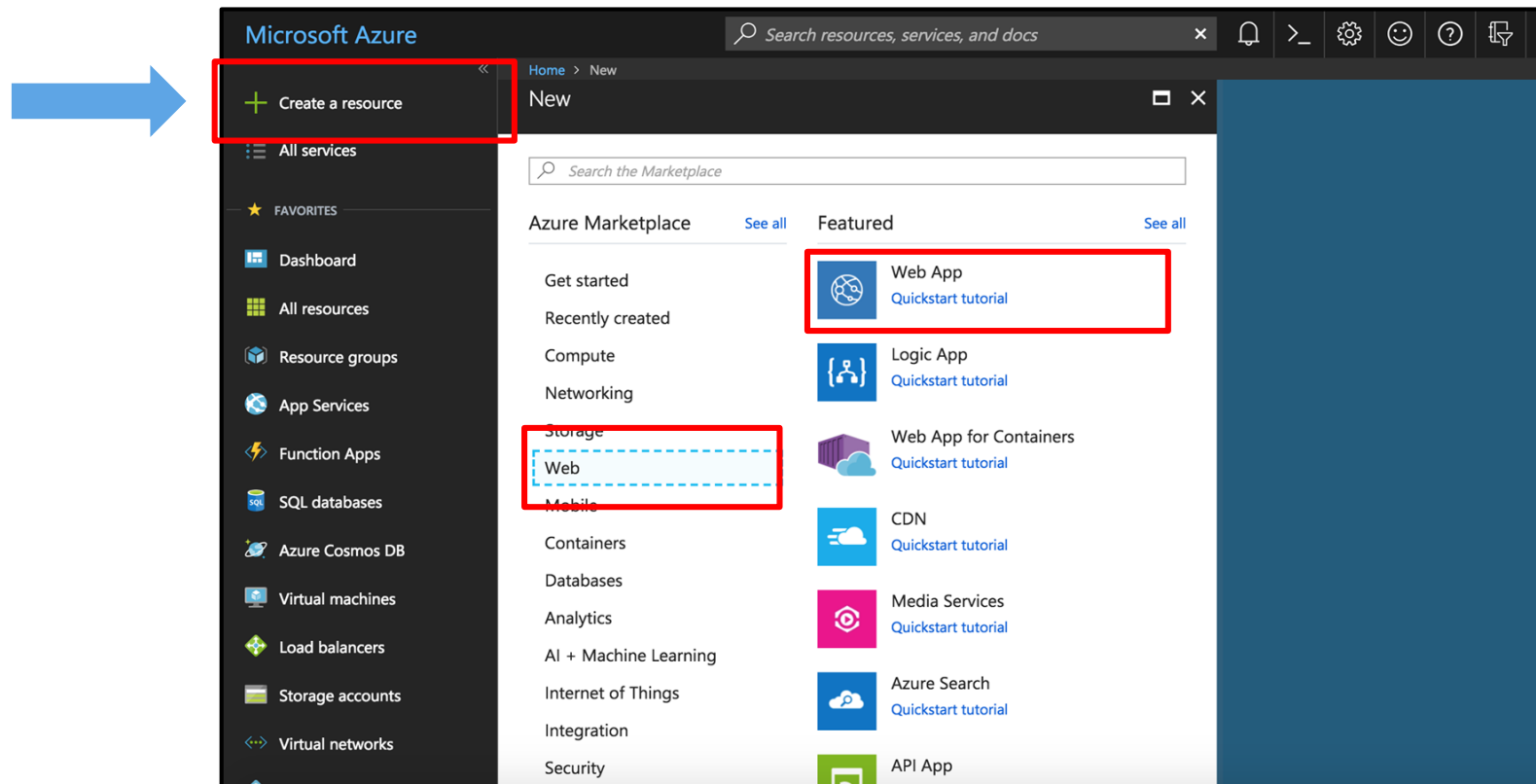
Create Automation options

Finally, let's deploy!

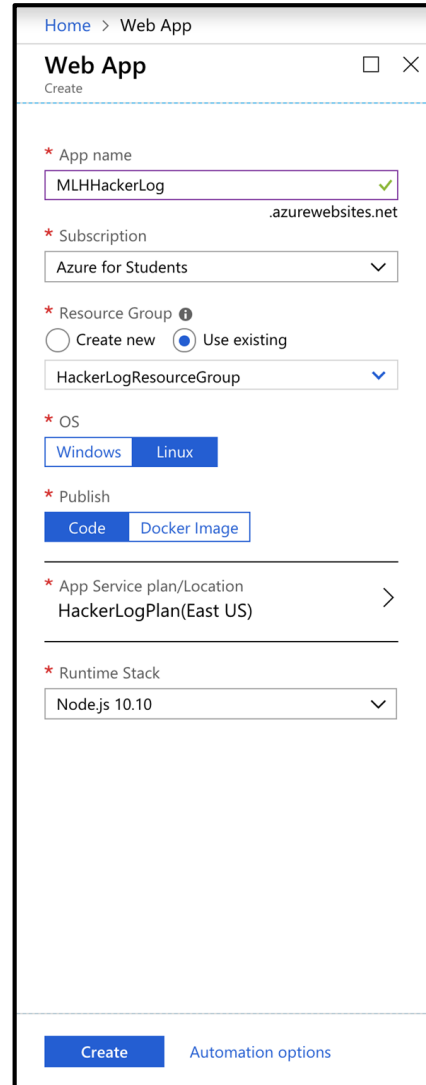
Deploy Your App to Microsoft Azure

Instructions

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



Deploy Your App to Microsoft Azure



The screenshot shows the 'Web App' creation wizard in the Azure portal. The breadcrumb navigation at the top reads 'Home > Web App'. The title 'Web App' is followed by a 'Create' button and a close icon. The form contains several sections, each marked with a red asterisk:

- App name:** A text input field containing 'MLHHackerLog' with a green checkmark icon to its right. Below the field, the text '.azurewebsites.net' is visible.
- Subscription:** A dropdown menu showing 'Azure for Students'.
- Resource Group:** Radio buttons for 'Create new' and 'Use existing' (selected). Below, a dropdown menu shows 'HackerLogResourceGroup'.
- OS:** Two buttons, 'Windows' and 'Linux' (selected).
- Publish:** Two buttons, 'Code' and 'Docker Image'.
- App Service plan/Location:** A dropdown menu showing 'HackerLogPlan(East US)' with a right-pointing chevron.
- Runtime Stack:** A dropdown menu showing 'Node.js 10.10'.

At the bottom of the form, there is a blue 'Create' button and a link for 'Automation options'.

Instructions

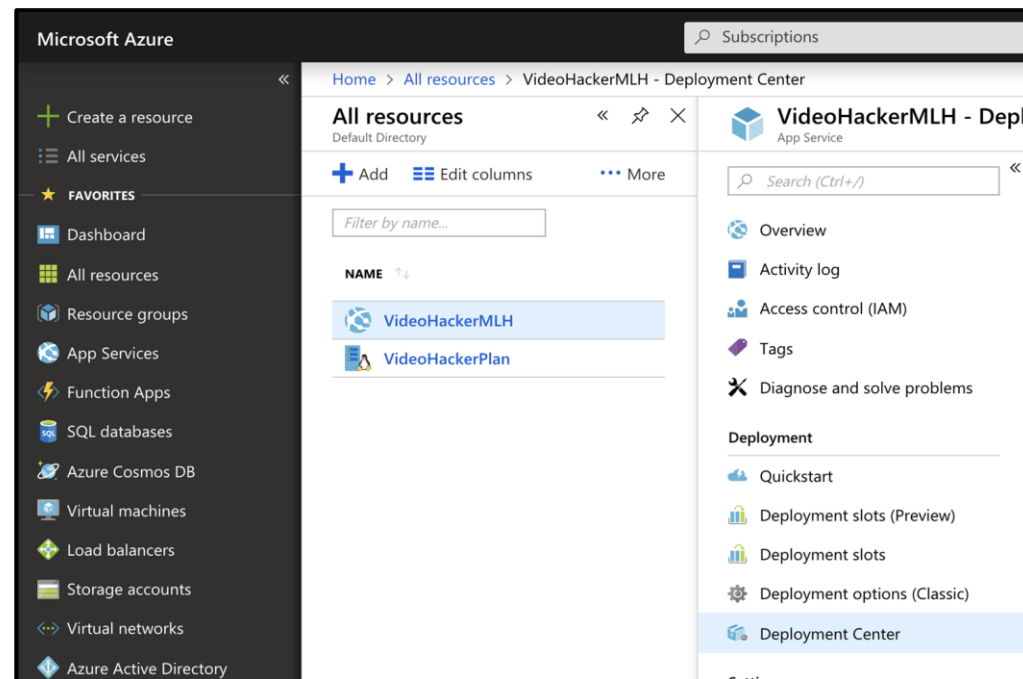
2. In the panel that opens, name your app something unique.
3. Select the Resource Group you made previously.
4. Select **Linux** as your operating system.
5. Select **Node.js 10.10** as your Runtime Stack.
6. Click **Create** at the bottom of the screen.

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

Deploy Your App to Microsoft Azure

Instructions

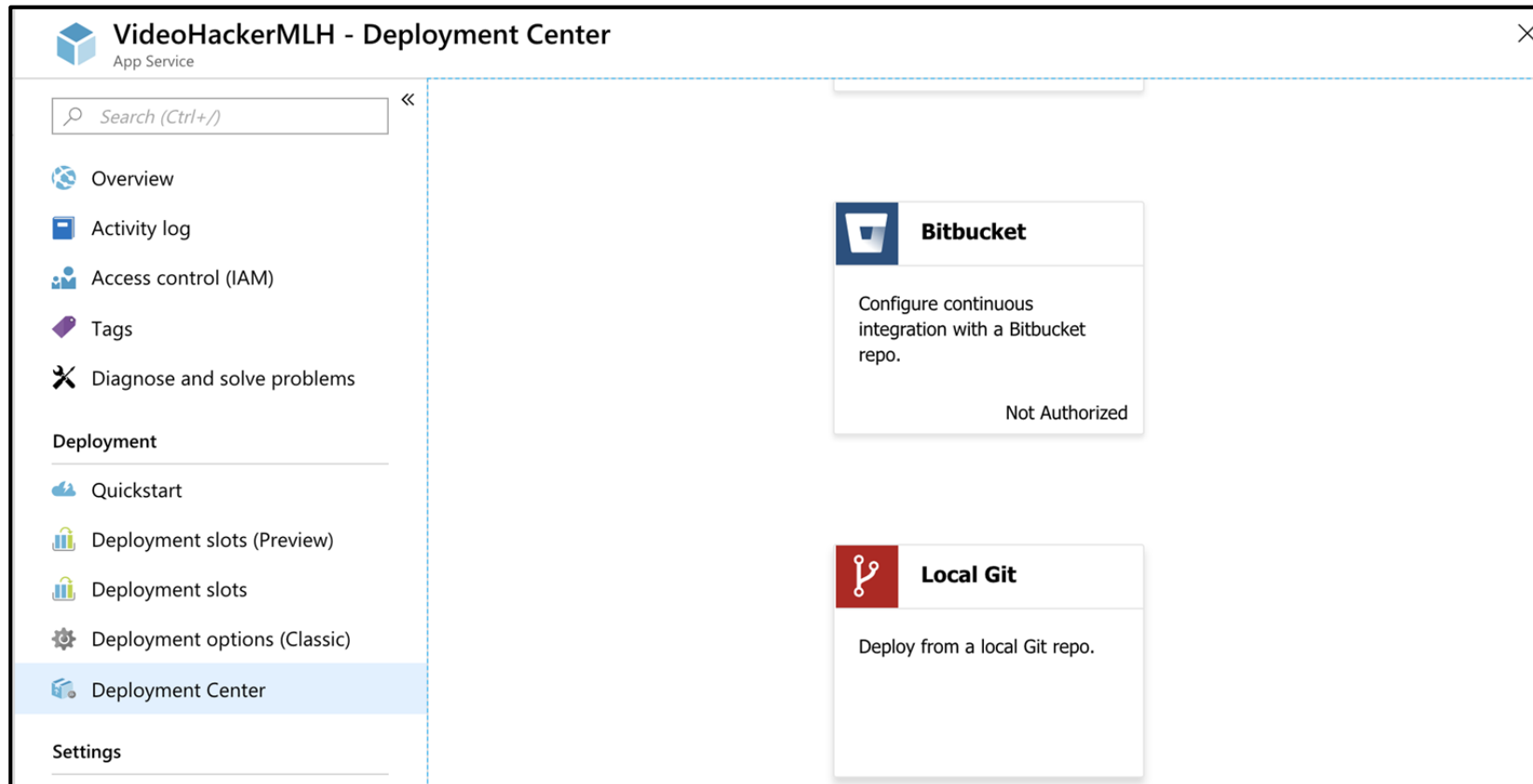
1. On the left side of your Azure portal, select **All resources**.
2. Click the App Service you just made.
3. Select **Deployment Center**.



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**.




Deployment Center


App Service Deployment Center enables you to choose the location of your code as well as options for build and deployment to the cloud. [Learn more](#)




SOURCE CONTROL



BUILD PROVIDER




SUMMARY



App Service Kudu build server

Use App Service as the build server. The App Service Kudu engine will automatically build your code during deployment when applicable with no additional configuration required.




Azure Pipelines (Preview)


Configure a robust deployment pipeline for your application using Azure Pipelines, part of Azure DevOps Services (formerly known as VSTS). The pipeline builds, runs load tests and deploys to staging slot and then to production.


Deploy Your App to Microsoft Azure


Instructions

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

 Refresh

 Disconnect

 Sync

 Deployment Credentials

Source
Local Git

Build
Kudu

Git Clone Uri
https://null@videohackerm1h.scm.azurewebsites.net:4

TIME	STATUS	COMMIT ID (AUTHOR)
No data to display		

App Credentials

User Credentials

User Credentials are defined by you, the user, and can be used with all the apps to which you have access. These credentials can be used with FTP, Local Git and WebDeploy.

Username
localhost

Password
.....





Confirm Password
.....

Save Credentials

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>`.

 Refresh  Disconnect  Sync  Deployment Credentials

Source

Local Git

Build

Kudu

Git Clone Uri

`https://mlh-localhost@videohackermih.scm.azurewebsites.net:443/videohackermih.git`

TIME	STATUS	COMMIT ID (AUTHOR)	CHECKIN MESSAGE	LOGS
No data to display				


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>:<primary_password>@<hostname>:<port>/?ssl=true&replicaSet=globaldb

Application settings

 Application Settings are encrypted at rest and transmitted over an encrypted channel. You can choose to display them in plain text in your browser by using the controls below.

Hide Values

Show Values

APP SETTING NAME	VALUE	SLOT SETTING	DELETE
MONGODB_URI	<input type="text" value="Enter a value"/>	<input type="checkbox"/>	

[+ Add new setting](#)

Deploy Your App to Microsoft Azure

Instructions

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


Before:

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

After:


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

Application settings

 Application Settings are encrypted at rest and transmitted over an encrypted channel. You can choose to display them in plain text in your browser by using the controls below.

Hide Values

Show Values

APP SETTING NAME	VALUE	SLOT SETTING	DELETE
MONGODB_URI	<input type="text" value="Enter a value"/>	<input type="checkbox"/>	

[+ Add new setting](#)

You just:

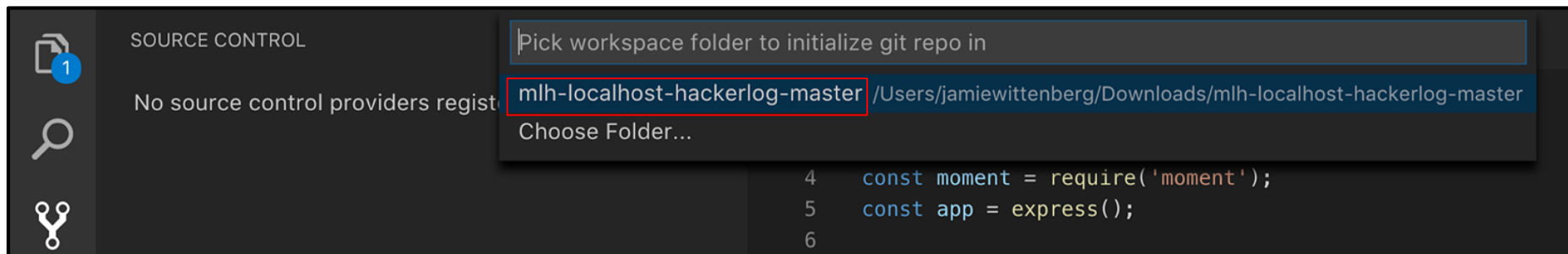
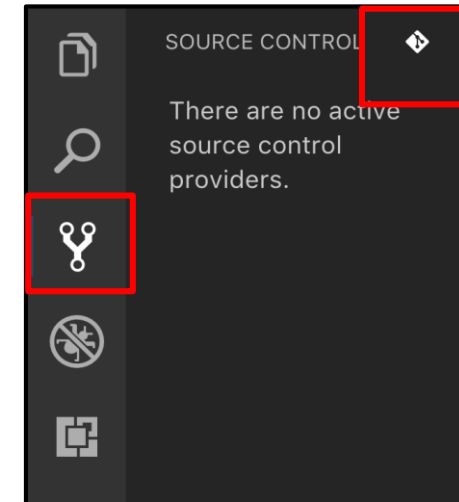
- 1. Created a database on Microsoft Azure.**
- 2. Created an App Service Plan.**
- 3. Created a Node app environment to deploy your app to.**

Head back to Visual Studio Code for the rest!

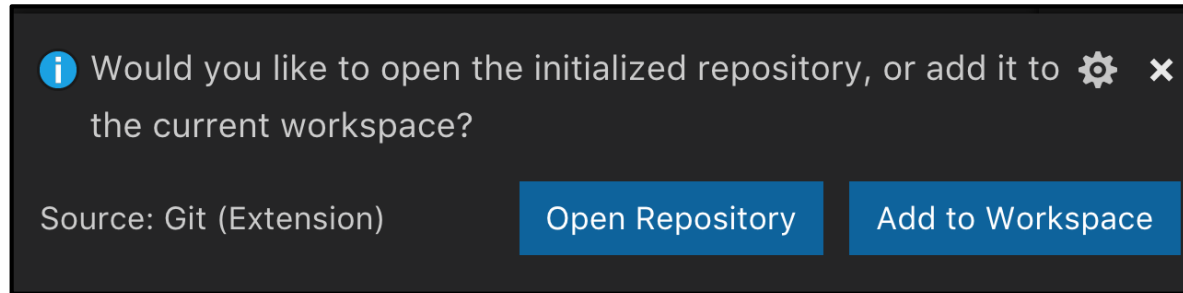
Deploy Your App to Microsoft Azure

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.

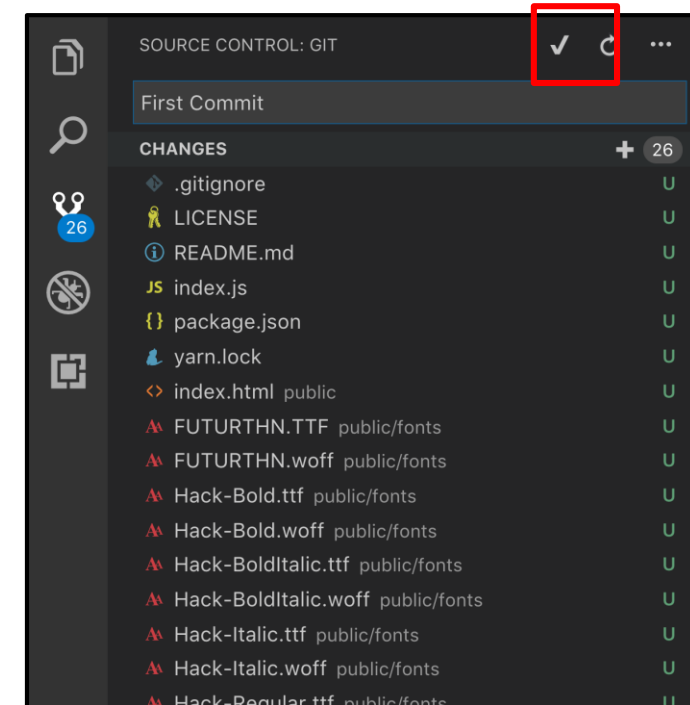
Note: Make sure the folder is NOT your Desktop or entire Documents folder.



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.

Refresh Disconnect Sync Deployment Credentials

Source

Local Git

Build

Kudu

Git Clone Uri

`https://mlh-localhost@videohackermih.scm.azurewebsites.net:443/videohackermih.git`

TIME	STATUS	COMMIT ID (AUTHOR)	CHECKIN MESSAGE	LOGS
No data to display				

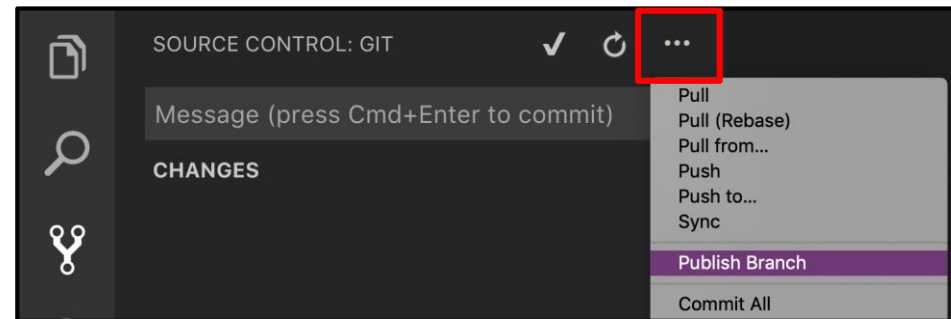
8. In your terminal, type `git remote add origin` and paste what you just copied. This is how you tell Git where to send your code.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  1: bash  +  [ ]  [X]  ^  X
@wonderwoman [mlh-localhost-video-hacker-master] ♥ git remote add origin https://null@videohackermih
.scm.azurewebsites.net:443/videohackermih.git
@wonderwoman [mlh-localhost-video-hacker-master] (master) ♥ |
```

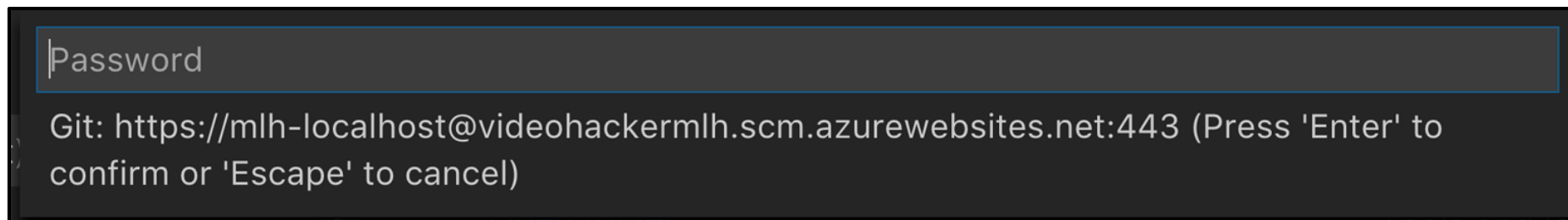
Deploy Your App to Microsoft Azure

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

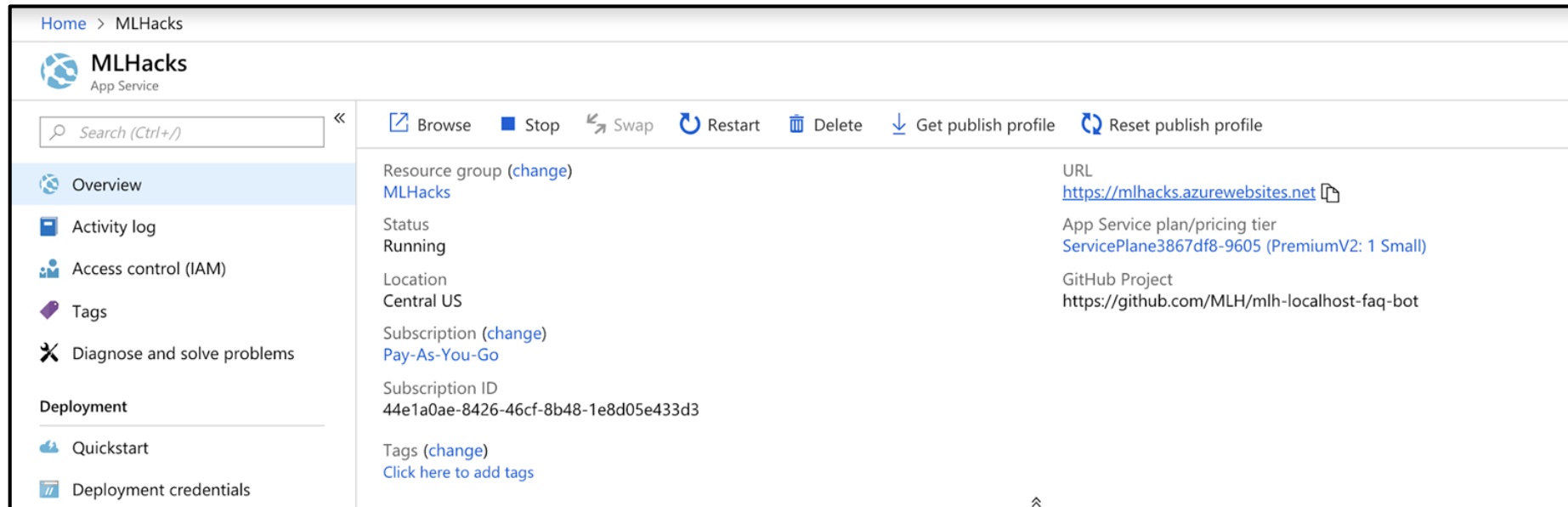
13. Back in Azure Deployment Centre, you can see that your deploy is in progress!

Source Local Git				
Build Kudu		Git Clone Uri https://mlh-localhost@videohackermih.scm.azurewebsites.net:443/videohackermih.git		
TIME	STATUS	COMMIT ID (AUTHOR)	CHECKIN MESSAGE	LOGS
Sunday, November 18, 2018				
7:52:16 PM GMT-5	Running deployment comm	ed2dd4f (JamieMLH)	First Commit	

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!



The screenshot displays the Microsoft Azure portal interface for an App Service named "MLHacks". The breadcrumb navigation at the top shows "Home > MLHacks". The left sidebar contains a search bar and a list of navigation options: Overview (selected), Activity log, Access control (IAM), Tags, Diagnose and solve problems, Deployment, Quickstart, and Deployment credentials. The main content area features a toolbar with actions: Browse, Stop, Swap, Restart, Delete, Get publish profile, and Reset publish profile. Below the toolbar, the Overview section displays the following information:

- Resource group: [MLHacks](#) (change)
- Status: Running
- Location: Central US
- Subscription: [Pay-As-You-Go](#) (change)
- Subscription ID: 44e1a0ae-8426-46cf-8b48-1e8d05e433d3
- Tags: [Click here to add tags](#) (change)
- URL: <https://mlhacks.azurewebsites.net>
- App Service plan/pricing tier: [ServicePlane3867df8-9605 \(PremiumV2: 1 Small\)](#)
- GitHub Project: <https://github.com/MLH/mlh-localhost-faq-bot>