

Build and Deploy Node.js Apps With Microsoft Azure

Troubleshooting Guide



Welcome to the troubleshooting guide of HackerLog: Build and Deploy Node.js Apps with Microsoft Azure! This guide will help you with troubleshooting any issues you may face while following the workshop materials.

Please choose your Operating System below to see the troubleshooting guide:

[WINDOWS](#)

[OSX](#)

This is the troubleshooting guide for Windows. If that is not your operating system, please go back to the first page to choose the right troubleshooting guide.

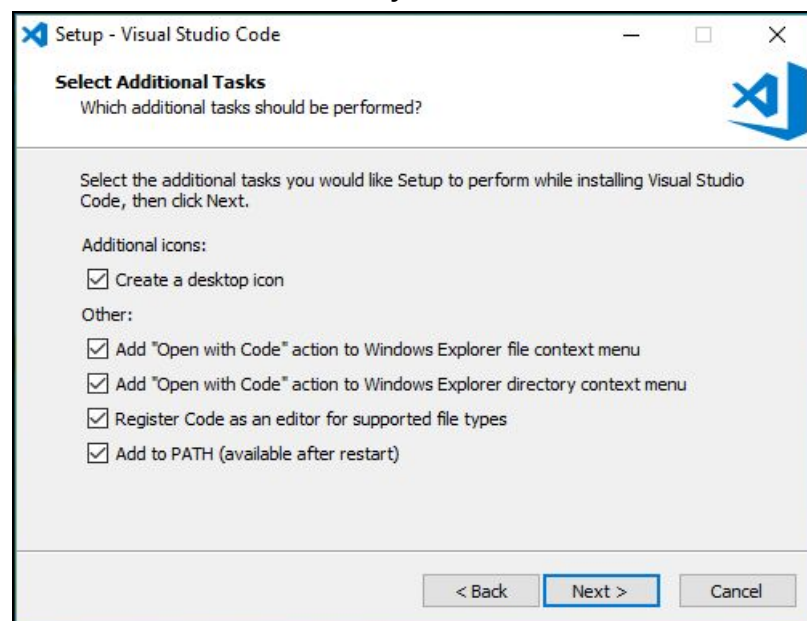
The following sections describe the possible problems that could occur or things to be aware of when following the workshop.

SECTION 1: CREATING AN AZURE ACCOUNT

1. It is possible that there are some errors while creating an Azure account. Most of these errors disappear when you **create the account in an incognito tab**.
2. During signup, you can verify your mobile number by providing an OTP. If you choose to receive the OTP via text message, note that it might take some time. Simply wait for a minute and if you do not receive an OTP by then, **click on the "I did not receive a code" button**.
3. **Do not use the OTP via call option**, as this simply redirects you to the beginning of the form.
4. If you have previously used Azure and have used your free trial credits, you will need to input your credit card information and sign up for a Pay-As-You-Go Account.
5. If you do not have a student email address, you will need to input your credit card information and sign up for a Pay-As-You-Go Account.

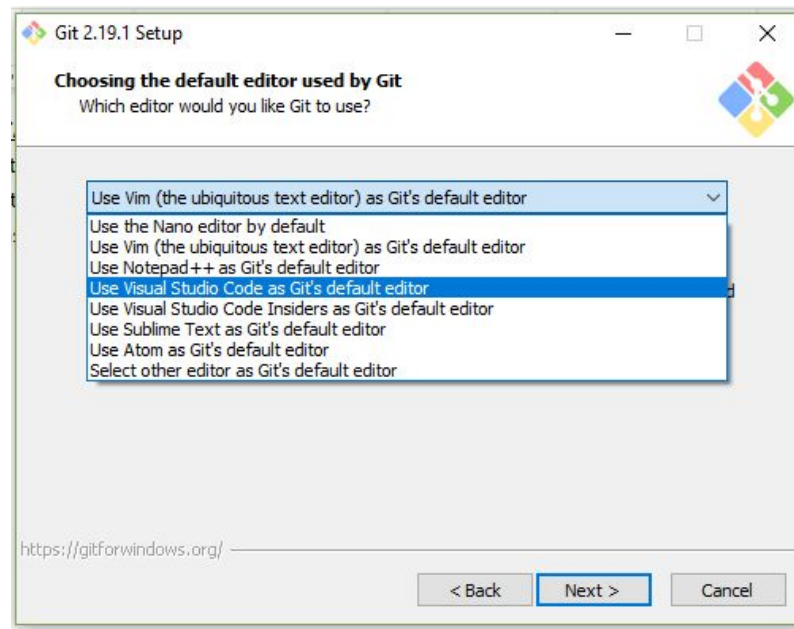
SECTION 2: DOWNLOAD AND INSTALL VS CODE

1. If the site does not show the correct version for your OS, scroll down to the bottom of the site and manually select the version that matches your OS.
2. Restart your PC after installation to ensure the PATH variable is set (not essential, but may be useful).
3. During install, choose all the options in this dialog box to make it easy for you to work with VS code (not essential, but may be useful).





SECTION 3: INSTALLING GIT

1. Note that Git always installs the latest version for your OS automatically so if you do not want that you need to manually select which version of git you want.
2. During install, you could choose VS code to be Git's default editor (not essential, but may be useful).



3. After installing git, if your VS Code is open, **first close it**. Reopen VS code, create a new terminal and try to run the git --version. If that fails (shows "git is not recognized"), **kill the terminal**, create another terminal and try the command again.

```
git : The term 'git' is not recognized as the name of a cmdlet, function, script file, or operable program. Check the spelling of the
name, or if a path was included, verify that the path is correct and try again.
At line:1 char:1
+ git init
+ ~~~~~
+ CategoryInfo          : ObjectNotFound: (git:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException
```

To kill the terminal, click on the icon that looks like this -  and to add a new terminal, click on the icon that looks like this - 

SECTION 4: INSTALLING NODE

1. After installing node, if your VS Code is open, **first close it**. Reopen VS code, create a new terminal and try to run the npm install (ensure that you are in the proper directory first). If that fails (shows "npm is not recognized"), **kill the terminal**, create another terminal and try the command again.

```
npm : The term 'npm' is not recognized as the name of a cmdlet, function, script file, or operable program. Check the spelling of the
name, or if a path was included, verify that the path is correct and try again.
At line:1 char:1
+ npm install
+ ~~~~~
+ CategoryInfo          : ObjectNotFound: (npm:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException
```

2. ACCEPTABLE NODE VERSIONS: As of 2019. February 15, Node versions 8.11, 9.4, 10.11, and 10.14 work. Please use these versions. =}

SECTION 5: ADDING CODE

1. Note inside console.log for printing the port we are listening on, we do not use the normal single quotes `"`, but the other quote which is next to key 1 on the keyboard - ```
2. Ensure you **delete the mlh-hackerlog-master part in the extract all dialog box**, or else there will be a nesting of folders.
3. While entering the code, one thing to note is that the line numbers may change if you enter the code below the comments so the best thing to do would be to **replace the comments with the code**. (Slide 66)

SECTION 6: RUNNING THE CODE

1. When you open up a terminal, ensure you are in the correct path i.e. you should see something like `\Downloads\mlh-localhost-hackerlog-master>` and the cursor waiting there. If that doesn't happen, **kill the terminal and create a new one**. If that still doesn't help, **restart VS code**. Otherwise you might get an error like "ENOENT: no such file or directory, open '..\package.json'"
2. Wait till **"HackerLog app listening on port 3000!"** is shown on the console before trying to open `localhost:3000` in your browser or it will say "This site can't be reached"



This site can't be reached

localhost refused to connect.

- Did you mean <http://localhost3000.org/>?
- Search Google for **localhost 3000**

ERR_CONNECTION_REFUSED

3. Type **clear** in your terminal to delete all the text shown, so that there is less clutter (not necessary, but recommended).
4. If you cancel the extraction midway (or if the extraction doesn't complete successfully for some reason) it will still create a folder but it might not have all the files so it may throw errors when you run npm start. If this happens to you, close VS code, **delete the extracted folder, extract again** and repeat all the steps from there. One common error is "failed to lookup..."
5. If you get the error "listen EADDRINUSE 0.0.0.0:3000", ensure there are no other terminals where the node index.js command is still running. If there are, **kill those terminals** and **re-run the command in a new terminal**.

```
Error: listen EADDRINUSE 0.0.0.0:3000
    at Object._errnoException (util.js:1022:11)
    at _exceptionWithHostPort (util.js:1044:20)
    at Server.setupListenHandle [as _listen2] (net.js:1367:14)
    at listenInCluster (net.js:1408:12)
    at doListen (net.js:1517:7)
    at _combinedTickCallback (internal/process/next_tick.js:141:11)
    at process._tickCallback (internal/process/next_tick.js:180:9)
    at Function.Module.runMain (module.js:695:11)
    at startup (bootstrap_node.js:188:16)
    at bootstrap_node.js:609:3
```

SECTION 7: CREATE A COSMOS RESOURCE

1. **Be sure to do this before creating your app!**

SECTION 8: DEPLOYING THE APP

1. In deployment option of the web app, you need to enter a username and a password. VS code will ask for this password every time you want to publish to this git repository so don't forget it!

Deployment option	Set up basic authentication
* Choose Source Local Git Repository	New user name and password Git and FTP cannot use your Windows account to authenticate, so this dialog lets you specify a user name and password that can be used when using those technologies. * Deployment user name <input type="text"/> * Password <input type="password"/> * Confirm Password <input type="password"/>
Setup connection <i>Set up basic authentication</i>	
Performance Test Not Configured	
OK	OK

2. Give your app a name. This name must be unique across Azure. Wait for the green check mark. If you get a red exclamation mark, you must rename your app.
3. Format for Connection String is
`mongodb://<username>:<primary_password>@<hostname>:<port>/?ssl=true&replicaSet=globaldb`
 Or simply copy the **primary connection string** from the Connection Strings menu.
4. After you initialize the git repository in VS code, the following dialog box appears. Click on the add to workspace button and continue with the other steps.

Would you like to open the initialized repository, or add it to the current workspace?

Source: Git (Extension)

Open Repository
 Add to Workspace

This is the troubleshooting guide for OSX. If that is not your operating system, please go back to the first page to choose the right troubleshooting guide.

The following sections describe the possible problems that could occur or things to be aware of when following the workshop.

SECTION 1: CREATING AN AZURE ACCOUNT

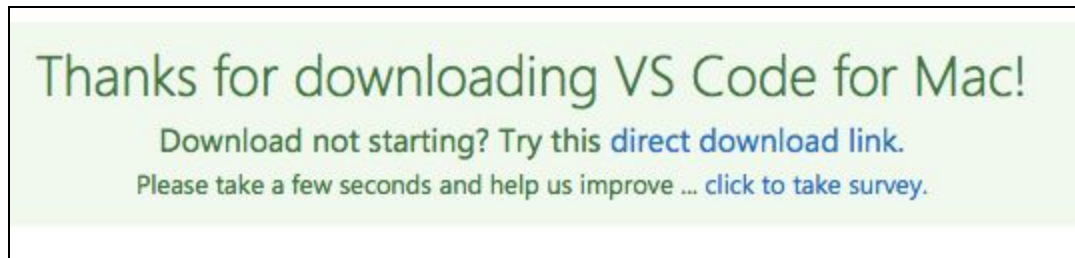
1. It is possible that there are some errors while creating an Azure account. Most of these errors disappear when you **create the account in an incognito tab**.
2. During signup, you can verify your mobile number by providing an OTP. If you choose to receive the OTP via text message, note that it might take some time. Simply wait for a minute and if you do not receive an OTP by then, **click on the “I did not receive a code” button**.
3. **Do not use the OTP via call option**, as this simply redirects you to the beginning of the form.
4. If you have previously used Azure and have used your free trial credits, you will need to input your credit card information and sign up for a Pay-As-You-Go Account.
5. If you do not have a student email address, you will need to input your credit card information and sign up for a Pay-As-You-Go Account.

SECTION 2: DOWNLOAD AND INSTALL VS CODE

1. Select **Mac** as your operating system on Visual Studio Code's website and the Stable version as the package you want to download.



2. If the Visual Studio Code package does not start automatically downloading, you can select **Direct Download**.



3. Restart your computer after installation to ensure the PATH variable is set (not essential, but may be useful).

SECTION 3: INSTALLING GIT



1. There are several ways to install Git on a Mac. The easiest way is to install the XCode Command Line tools. On Mavericks (10.9) or above you can run Git from the terminal. Open your terminal and type `$ git --version` in and then press Enter.

If this does not work, you can go to: <https://git-scm.com/download/mac> to download Git via a binary installer. This will also make sure you have the most up to date version.



2. Still not working? It might be that your computer's operating system is not compatible with the latest Git version because your operating system is out of date. Go to your **System Preferences**, then Select **Software Updates**. If there are any updates that you need to download, they will be here.
3. After installing git, if your VS Code is open, **first close it**. Reopen VS code, create a new terminal and try to run the `git --version`. If that fails (shows "git not recognized"), **kill the terminal** and create another one.

```
git : The term 'git' is not recognized as the name of a cmdlet, function, script file, or operable program. Check the spelling of the
name, or if a path was included, verify that the path is correct and try again.
At line:1 char:1
+ git init
+ ~~~~~
+ CategoryInfo          : ObjectNotFound: (git:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException
```

To kill the terminal, click on the icon that looks like this -  and to add a new terminal, click on the icon that looks like this - 

SECTION 4: INSTALLING NODE

1. Node demands two different packages before you can install it. Many developers already have these packages on their computer, but if Node is having a hard time running, it may be because these packages are not installed. You can go to NodeJS.org and download the Node installer for Mac.

Node.js® is a JavaScript runtime built on **Chrome's V8 JavaScript engine**.

Download for macOS (x64)

10.13.0 LTS

Recommended For Most Users

11.1.0 Current

Latest Features

[Other Downloads](#) | [Changelog](#) | [API Docs](#) [Other Downloads](#) | [Changelog](#) | [API Docs](#)

Or have a look at the **Long Term Support (LTS) schedule**.

Sign up for **Node.js Everywhere**, the official Node.js Monthly Newsletter.

2. Or, if you are worried about any packages you download wreaking havoc on your computer, you can download Homebrew, a package manager for Mac. To install Homebrew, open your Terminal and type in:

```
ruby -e "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

This is especially useful if you don't have administrator access in the system.

SECTION 5: ADDING CODE

1. Note inside console.log for printing the port we are listening on, we do not use the normal single quotes `"`, but the other quote which is next to key 1 on the keyboard - ```
2. While entering the code, one thing to note is that the line numbers may change if you enter the code below the comments so the best thing to do would be to **replace the comments with the code**. (Slide 66)

SECTION 6: RUNNING THE CODE

1. When you open up a terminal, ensure you are in the correct path i.e. you should see something like `\Downloads\mlh-localhost-hackerlog-master>` and the cursor waiting there. If that doesn't happen, **kill the terminal and create a new one**. If that still doesn't help, **restart VS code**. Otherwise you might get an error like "ENOENT: no such file or directory, open '..\package.json'"
2. Wait till "**HackerLog app listening on port 3000!**" is shown on the console before trying to open `localhost:3000` in your browser or it will say "This site can't be reached."



This site can't be reached

localhost refused to connect.

- Did you mean <http://localhost3000.org/>?
- Search Google for **localhost 3000**

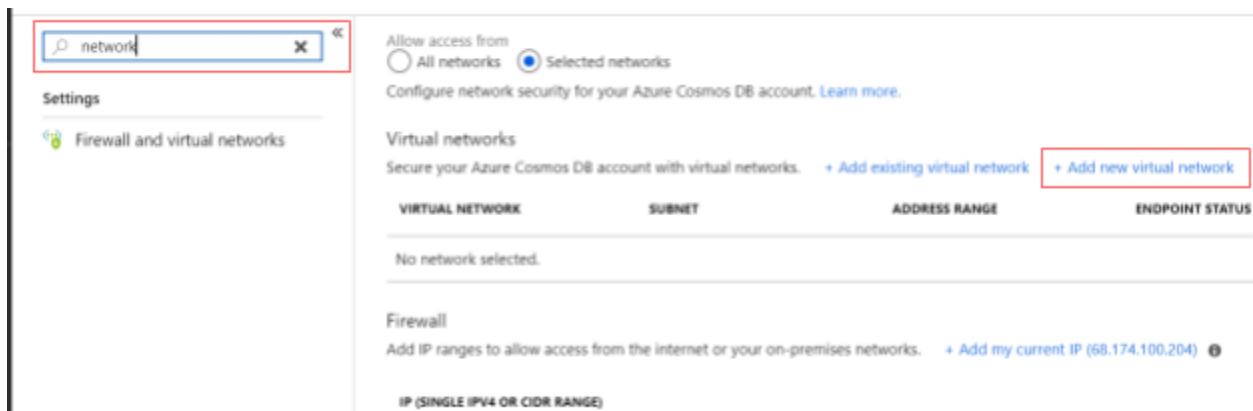
ERR_CONNECTION_REFUSED

3. Type **clear** in your terminal to delete all the text shown, so that there is less clutter (not necessary, but recommended).
4. If you get the error "listen EADDRINUSE 0.0.0.0:3000", ensure there are no other terminals where the node index.js command is still running. If there are, **kill those terminals** and **re-run the command in a new terminal**.

```
Error: listen EADDRINUSE 0.0.0.0:3000
    at Object._errnoException (util.js:1022:11)
    at _exceptionWithHostPort (util.js:1044:20)
    at Server.setupListenHandle [as _listen2] (net.js:1367:14)
    at listenInCluster (net.js:1408:12)
    at doListen (net.js:1517:7)
    at _combinedTickCallback (internal/process/next_tick.js:141:11)
    at process._tickCallback (internal/process/next_tick.js:180:9)
    at Function.Module.runMain (module.js:695:11)
    at startup (bootstrap_node.js:188:16)
    at bootstrap_node.js:609:3
```

SECTION 7: CREATE A COSMOS RESOURCE

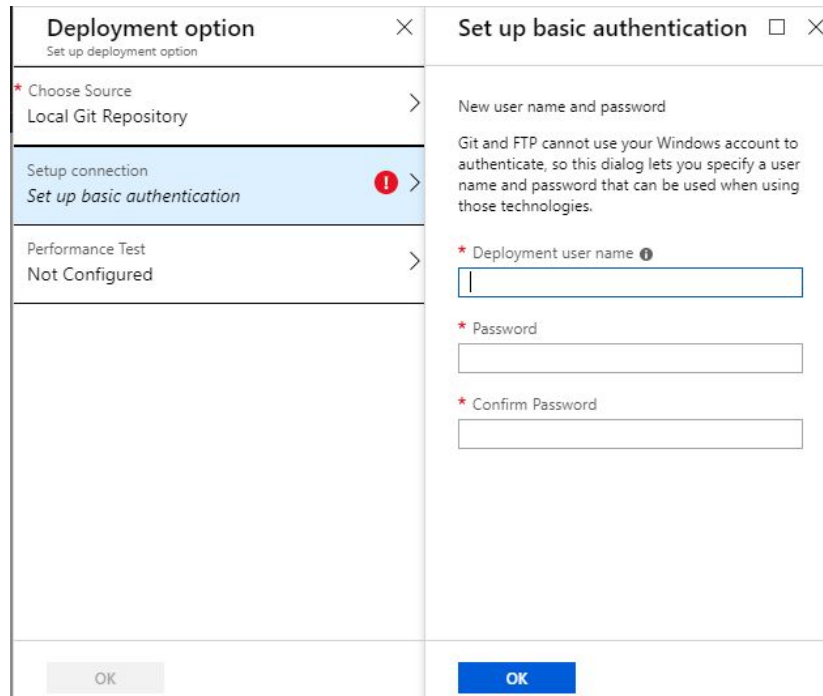
1. **instructions/MacCommands.md** will be helpful here!
2. Remember to change all the firewall back to selected resources after the workshop to prevent random people from sending requests to your app.
3. After creating the Cosmo DB and someone accidentally forgets to add a virtual network, they can always search for it and add it there. After clicking on "Add new virtual network" it will ask you about address spaces. Address space = 10.1.0.0/16, subnet Address range = 10.1.0.0/24



4.

SECTION 8: DEPLOYING THE APP

1. In deployment option of the web app, you need to enter a username and a password. VS code will ask for this password every time you want to publish to this git repository so don't forget it!



2. Give your app a name. This name must be unique across Azure. Wait for the green check mark. If you get a red exclamation mark, you must rename your app.
3. Format for Connection String is
`mongodb://<username>:<primary_password>@<hostname>:<port>/?ssl=true&replicaSet=globaldb`
Or simply copy the **primary connection string** from the Connection Strings menu.
4. After you initialize the git repository in VS code, the following dialog box appears. Click on the add to workspace button and continue with the other steps.

