Create a Node.js web app in Azure

Article • 03/24/2022 • 12 minutes to read • 13 contributors

Choose a development environment

Visual Studio Code | Command-line interface | Azure portal

In this article

In this quickstart, you'll learn how to create and deploy your first Node.js (Express) web app to Azure App Service. App Service supports various versions of Node.js on both Linux and Windows.

This quickstart configures an App Service app in the **Free** tier and incurs no cost for your Azure subscription.

Set up your initial environment

- Have an Azure account with an active subscription. Create an account for free.
- Install Node.js LTS and npm. Run the command `node --version` to verify that Node.js is installed.
- Have a FTP client (for example, FileZilla), to connect to your app.

Create your Node.js application

In this step, you create a basic Node.js application and ensure it runs on your computer.

Tip

If you have already completed the **Node.js tutorial** , you can skip ahead to **Deploy to Azure**.

1. Create a Node.js application using the Express Generator , which is installed by default with Node.js and NPM.

   Bash Copy

   ```bash
   npx express-generator myExpressApp --view ejs
   ```

2. Change to the application's directory and install the NPM packages.

   Bash Copy

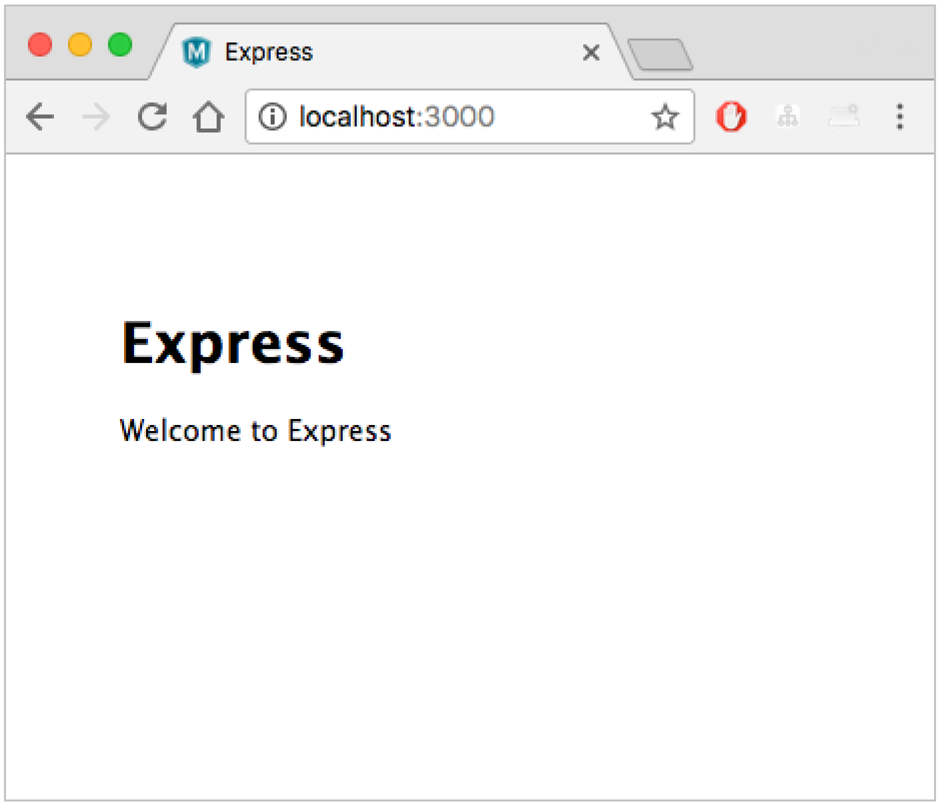
   ```bash
   cd myExpressApp && npm install
   ```

3. Start the development server with debug information.

   Bash Copy

   ```bash
   DEBUG=myexpressapp:* npm start
   ```

4. In a browser, navigate to `http://localhost:3000`. You should see something like this:



Express

Welcome to Express

Deploy to Azure

Before you continue, ensure that you have all the prerequisites installed and configured.

Note

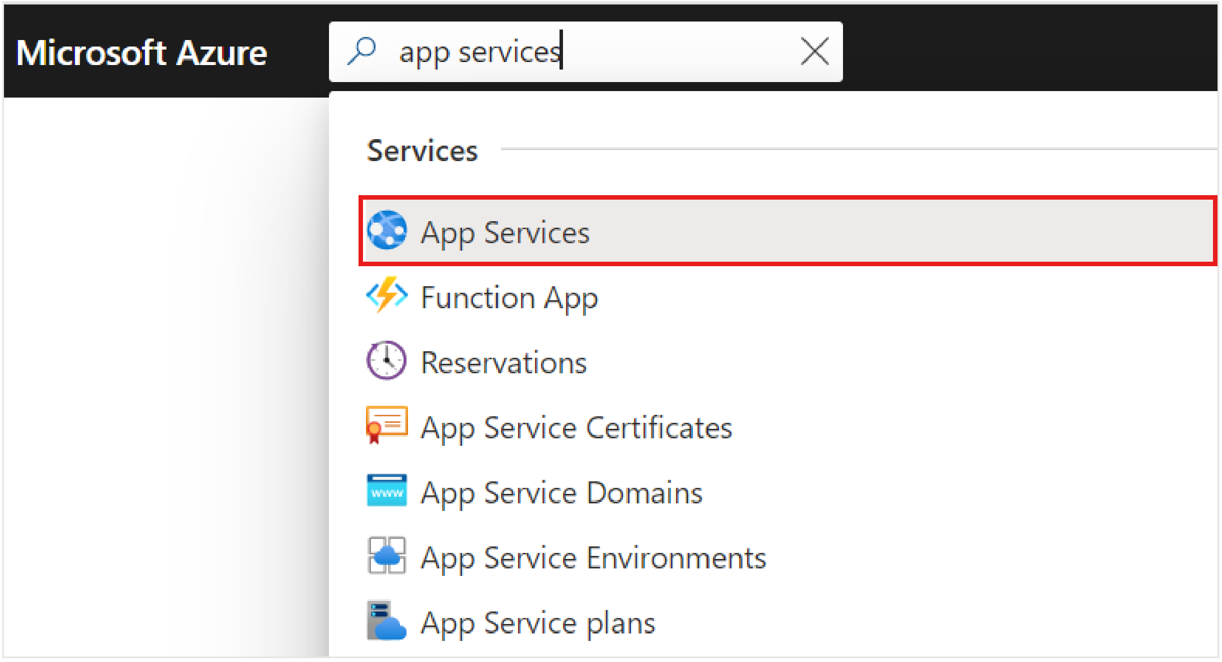
For your Node.js application to run in Azure, it needs to listen on the port provided by the `PORT` environment variable. In your generated Express app, this environment variable is already used in the startup script *bin/www* (search for `process.env.PORT`).

Sign in to Azure portal

Sign in to the Azure portal at https://portal.azure.com .

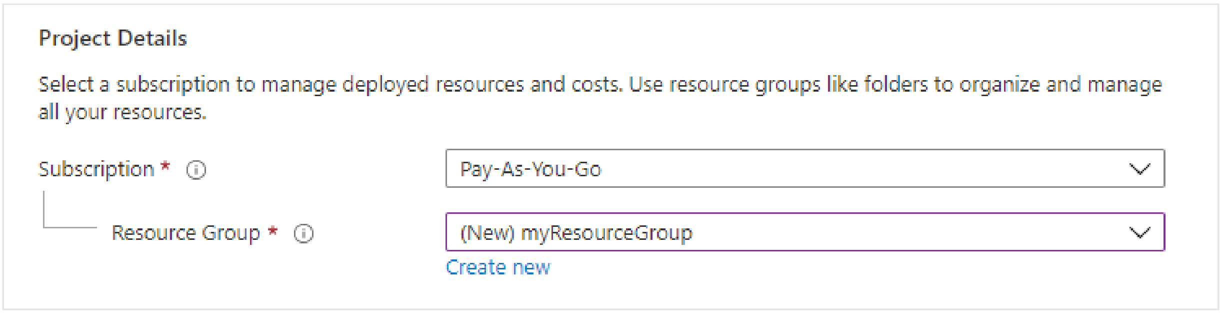
Create Azure resources

1. Type **app services** in the search. Under **Services**, select **App Services**.



2. In the **App Services** page, select **Create**.

3. In the **Basics** tab, under **Project details**, ensure the correct subscription is selected and then select to **Create new** resource group. Type *myResourceGroup* for the name.



Project Details

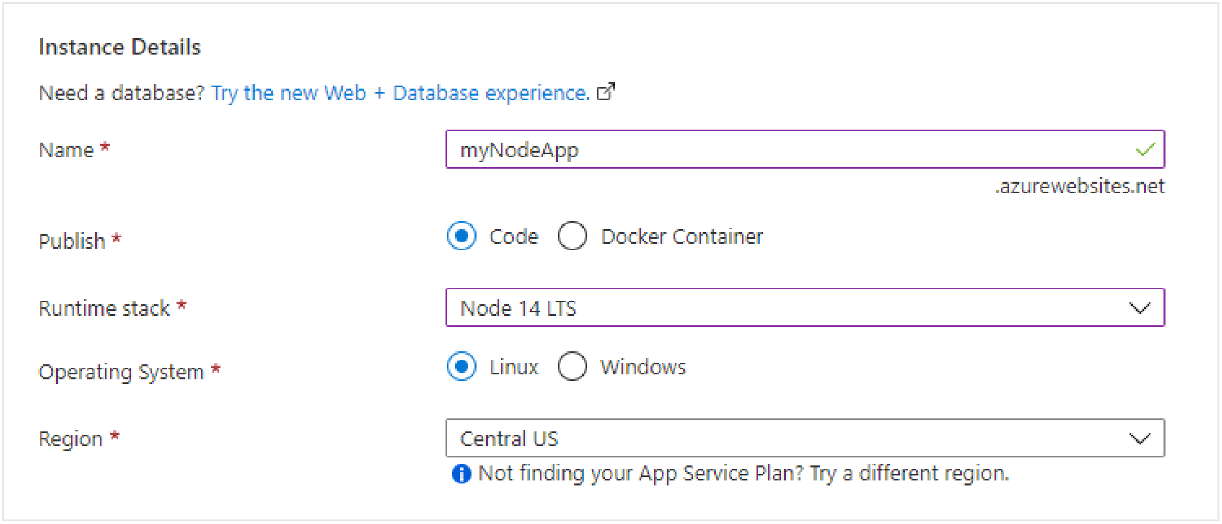
Select a subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * : Pay-As-You-Go

Resource Group * : (New) myResourceGroup

Create new

4. Under **Instance details**, type a globally unique name for your web app and select **Code**. Select *Node 14 LTS* **Runtime stack**, an **Operating System**, and a **Region** you want to serve your app from.

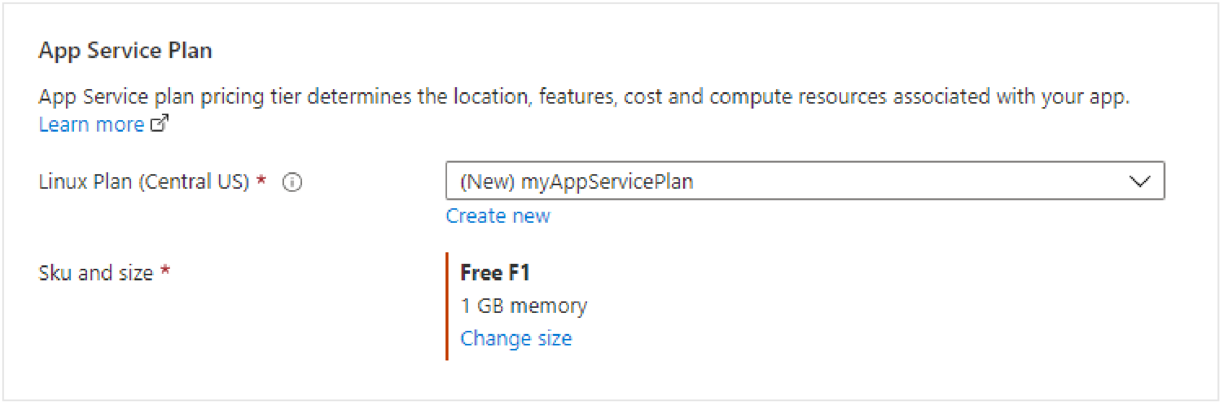

Instance Details

Need a database? Try the new Web + Database experience.

Name *	myNodeApp .azurewebsites.net
Publish *	[✓ Code] Docker Container
Runtime stack *	Node 14 LTS
Operating System *	[✓ Linux] Windows
Region *	Central US

Not finding your App Service Plan? Try a different region.

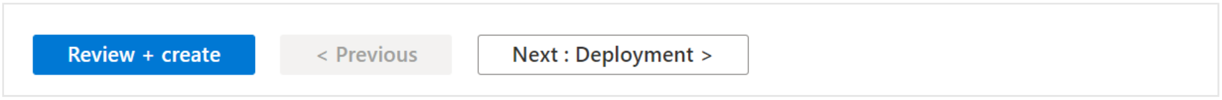
5. Under **App Service Plan**, select **Create new** App Service Plan. Type *myAppServicePlan* for the name. To change to the Free tier, select **Change size**, select **Dev/Test** tab, select **F1**, and select the **Apply** button at the bottom of the page.


App Service Plan

App Service plan pricing tier determines the location, features, cost and compute resources associated with your app. Learn more

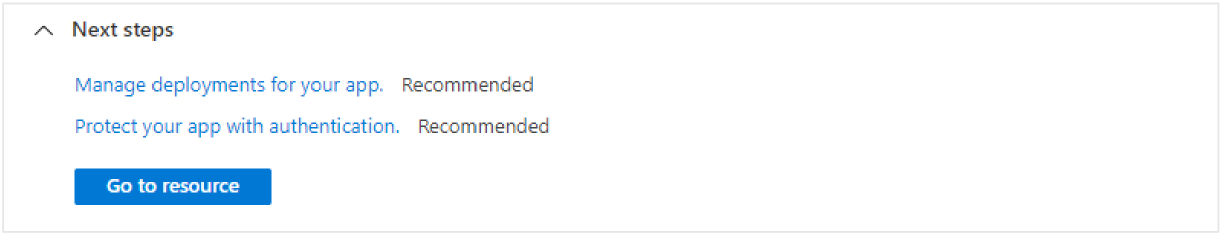
Linux Plan (Central US) *	(New) myAppServicePlan
Sku and size *	Free F1 1 GB memory

6. Select the **Review + create** button at the bottom of the page.



7. After validation runs, select the **Create** button at the bottom of the page.

8. After deployment is complete, select **Go to resource**.


Next steps

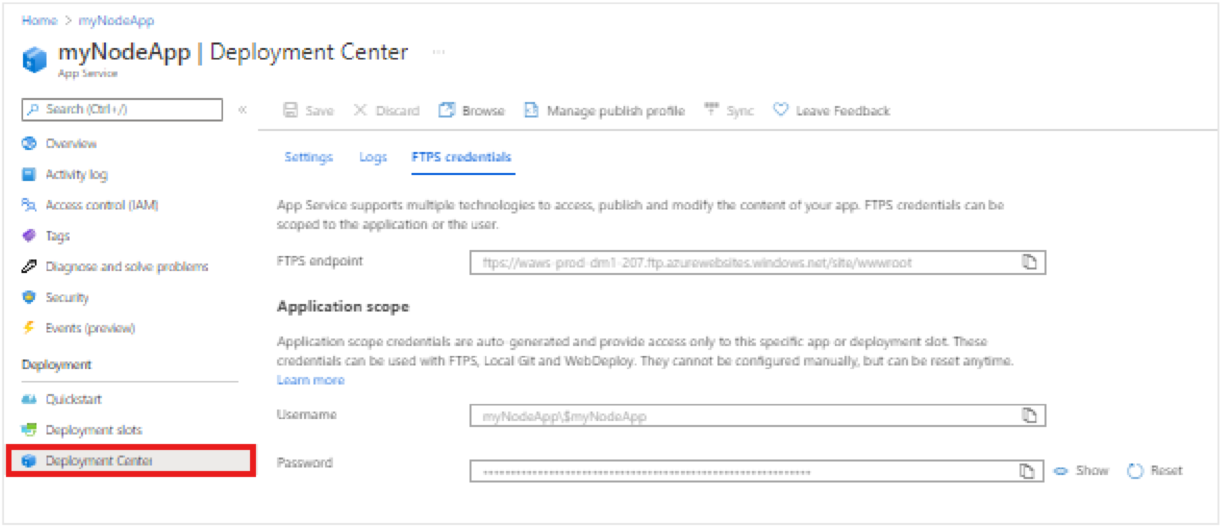
Manage deployments for your app. Recommended

Protect your app with authentication. Recommended

Get FTP credentials

Azure App Service supports **two types of credentials** for FTP/S deployment. These credentials aren't the same as your Azure subscription credentials. In this section, you get the *application-scope credentials* to use with FileZilla.

1. From the App Service app page, select **Deployment Center** in the left-hand menu and select **FTPS credentials** tab.



App Service supports multiple technologies to access, publish and modify the content of your app. FTPS credentials can be scoped to the application or the user.

FTPS endpoint: ftps://waws-prod-dm1-207.ftp.azurewebsites.windows.net/site/wwwroot

Application scope

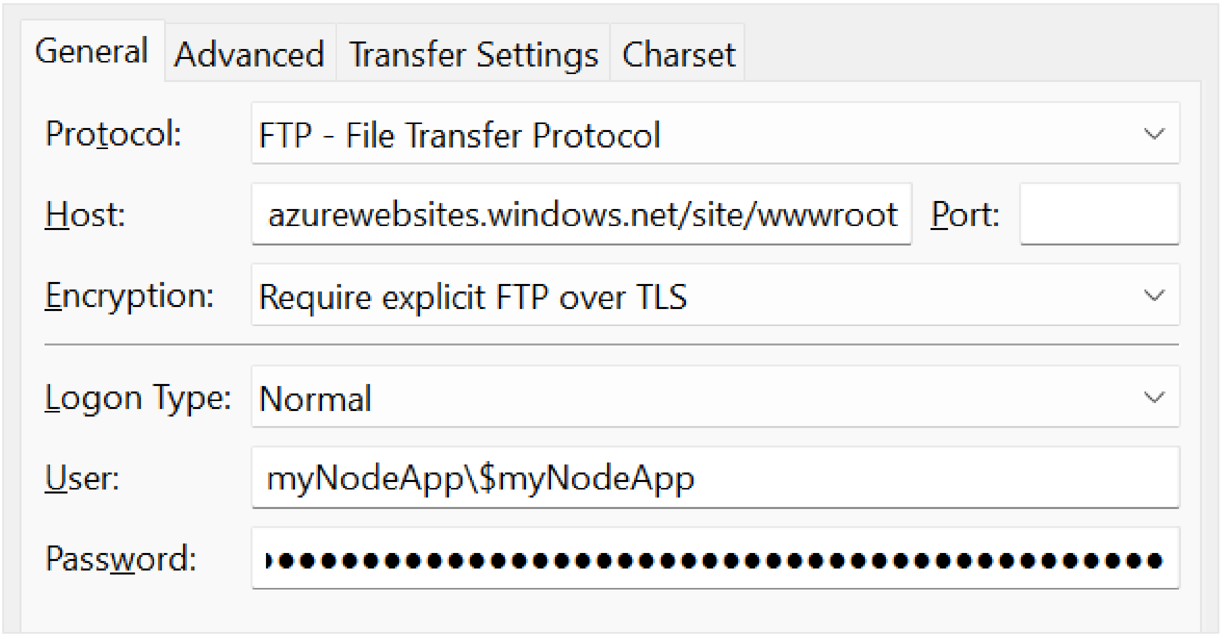
Application scope credentials are auto-generated and provide access only to this specific app or deployment slot. These credentials can be used with FTPS, Local Git and WebDeploy. They cannot be configured manually, but can be reset anytime.

Username: myNodeApp\$myNodeApp

Password: ••

2. Open **FileZilla** and create a new site.

3. From the **FTPS credentials** tab, copy **FTPS endpoint**, **Username**, and **Password** into FileZilla.

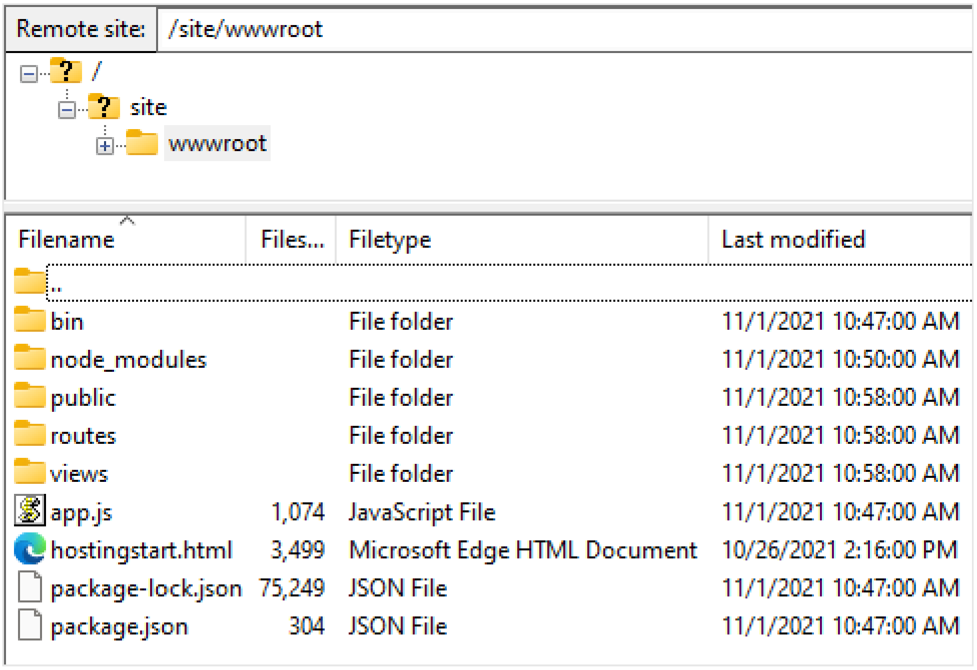


Protocol:	FTP - File Transfer Protocol
Host:	azurewebsites.windows.net/site/wwwroot
Port:	
Encryption:	Require explicit FTP over TLS
Logon Type:	Normal
User:	myNodeApp\$myNodeApp
Password:	•••

4. Select **Connect** in FileZilla.

Deploy files with FTP

1. Copy all files and directories files to the **/site/wwwroot** directory in Azure.



Remote site: /site/wwwroot

- /
  - site
    - wwwroot

Filename	Files...	Filetype	Last modified
..			
bin		File folder	11/1/2021 10:47:00 AM
node_modules		File folder	11/1/2021 10:50:00 AM
public		File folder	11/1/2021 10:58:00 AM
routes		File folder	11/1/2021 10:58:00 AM
views		File folder	11/1/2021 10:58:00 AM
app.js	1,074	JavaScript File	11/1/2021 10:47:00 AM
hostingstart.html	3,499	Microsoft Edge HTML Document	10/26/2021 2:16:00 PM
package-lock.json	75,249	JSON File	11/1/2021 10:47:00 AM
package.json	304	JSON File	11/1/2021 10:47:00 AM

2. Browse to your app's URL to verify the app is running properly.

Redeploy updates

You can deploy changes to this app by making edits in Visual Studio Code, saving your files, and then redeploy to your Azure app. For example:

1. From the sample project, open *views/index.ejs* and change

   HTML Copy
   ```html
   <p>Welcome to <%= title %></p>
   ```

   to

   HTML Copy
   ```html
   <p>Welcome to Azure</p>
   ```

2. Save your changes, then redeploy the app using your FTP client again.

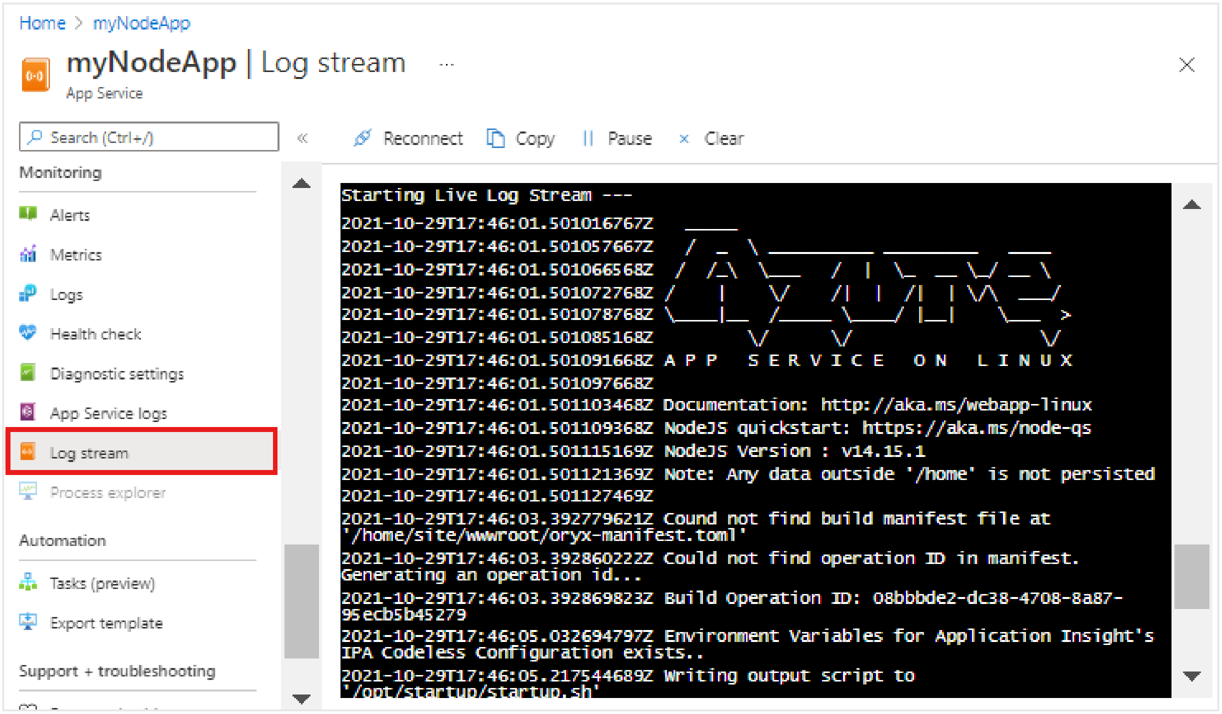
3. Once deployment is complete, refresh the webpage `http://<app-name>.azurewebsites.net`. You should see that the `Welcome to Express` message

has been changed to `Welcome to Azure!`.

Stream Logs

You can access the console logs generated from inside the app and the container in which it runs. You can stream log output (calls to `console.log()`) from the Node.js app directly in the Azure portal.

1. In the same **App Service** page for your app, use the left menu to scroll to the *Monitoring* section and select **Log stream**.



```
Starting Live Log Stream ---
2021-10-29T17:46:01.501016767Z   _____
2021-10-29T17:46:01.501057667Z  /  _  \ __________ _________   ____
2021-10-29T17:46:01.501066568Z /  /_\  \___   /  |  \_  __ \_/ __ \
2021-10-29T17:46:01.501072768Z /    |    \/    /|  |  /|  | \/\  ___/
2021-10-29T17:46:01.501078768Z \____|__  /_____ \____/ |__|    \___  >
2021-10-29T17:46:01.501085168Z         \/      \/                  \/
2021-10-29T17:46:01.501091668Z A P P   S E R V I C E   O N   L I N U X
2021-10-29T17:46:01.501097668Z
2021-10-29T17:46:01.501103468Z Documentation: http://aka.ms/webapp-linux
2021-10-29T17:46:01.501109368Z NodeJS quickstart: https://aka.ms/node-qs
2021-10-29T17:46:01.501115169Z NodeJS Version : v14.15.1
2021-10-29T17:46:01.501121369Z Note: Any data outside '/home' is not persisted
2021-10-29T17:46:01.501127469Z
2021-10-29T17:46:03.392779621Z Cound not find build manifest file at
'/home/site/wwwroot/oryx-manifest.toml'
2021-10-29T17:46:03.392860222Z Could not find operation ID in manifest.
Generating an operation id...
2021-10-29T17:46:03.392869823Z Build Operation ID: 08bbbde2-dc38-4708-8a87-
95ecb5b45279
2021-10-29T17:46:05.032694797Z Environment Variables for Application Insight's
IPA Codeless Configuration exists..
2021-10-29T17:46:05.217544689Z Writing output script to
'/opt/startup/startup.sh'
```

2. After a few seconds, the output window shows a message indicating that you're connected to the log-streaming service. You can generate more output activity by refreshing the page in the browser.

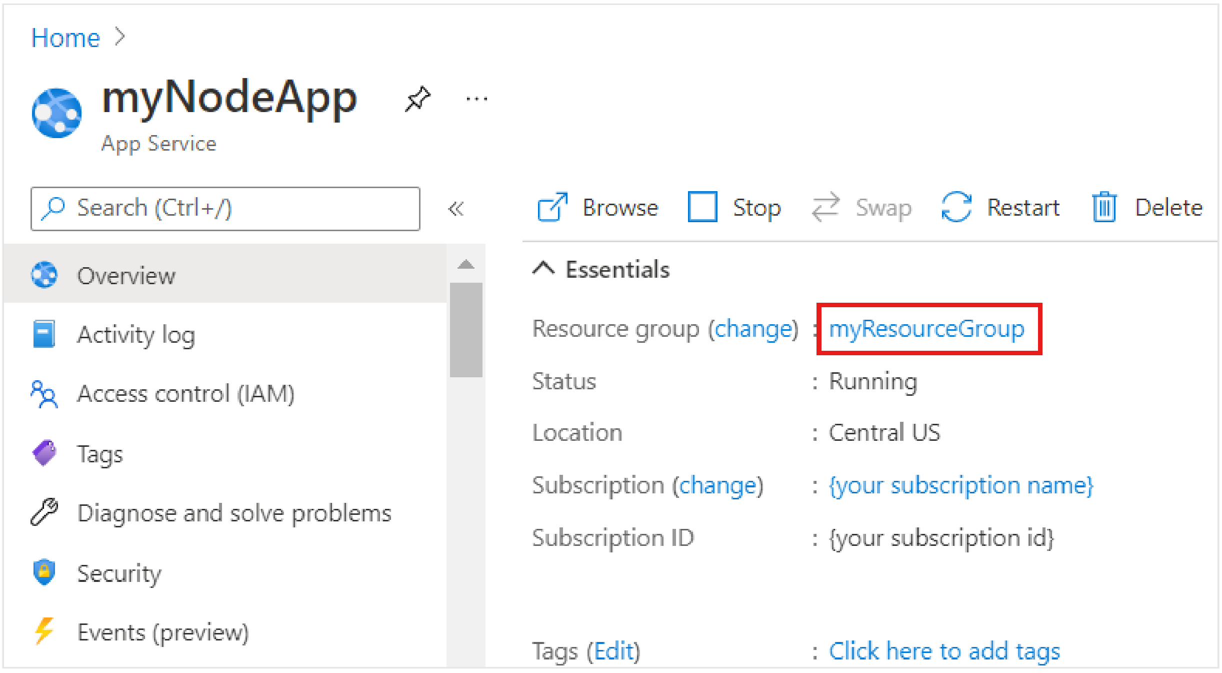
```
 Connecting...
 2021-10-26T21:04:14  Welcome, you are now connected to log-stream-
ing service.
 Starting Log Tail -n 10 of existing logs ----

/appsvctmp/volatile/logs/runtime/81b1b83b27ea1c3d598a1cdec28c71c4074
ce66c735d0be57f15a8d07cb3178e.log
 2021-10-26T21:04:08.614384810Z: [INFO]
 2021-10-26T21:04:08.614393710Z: [INFO]  # Enter the source directo-
ry to make sure the script runs where the user expects
 2021-10-26T21:04:08.614399010Z: [INFO]  cd "/home/site/wwwroot"
 2021-10-26T21:04:08.614403210Z: [INFO]
 2021-10-26T21:04:08.614407110Z: [INFO]  export NODE_PATH=/usr/lo-
cal/lib/node_modules:$NODE_PATH
 2021-10-26T21:04:08.614411210Z: [INFO]  if [ -z "$PORT" ]; then
 2021-10-26T21:04:08.614415310Z: [INFO]          export PORT=8080
 2021-10-26T21:04:08.614419610Z: [INFO]  fi
 2021-10-26T21:04:08.614423411Z: [INFO]
 2021-10-26T21:04:08.614427211Z: [INFO]  node /opt/startup/default-
static-site.js
 Ending Log Tail of existing logs ---
```

Clean up resources

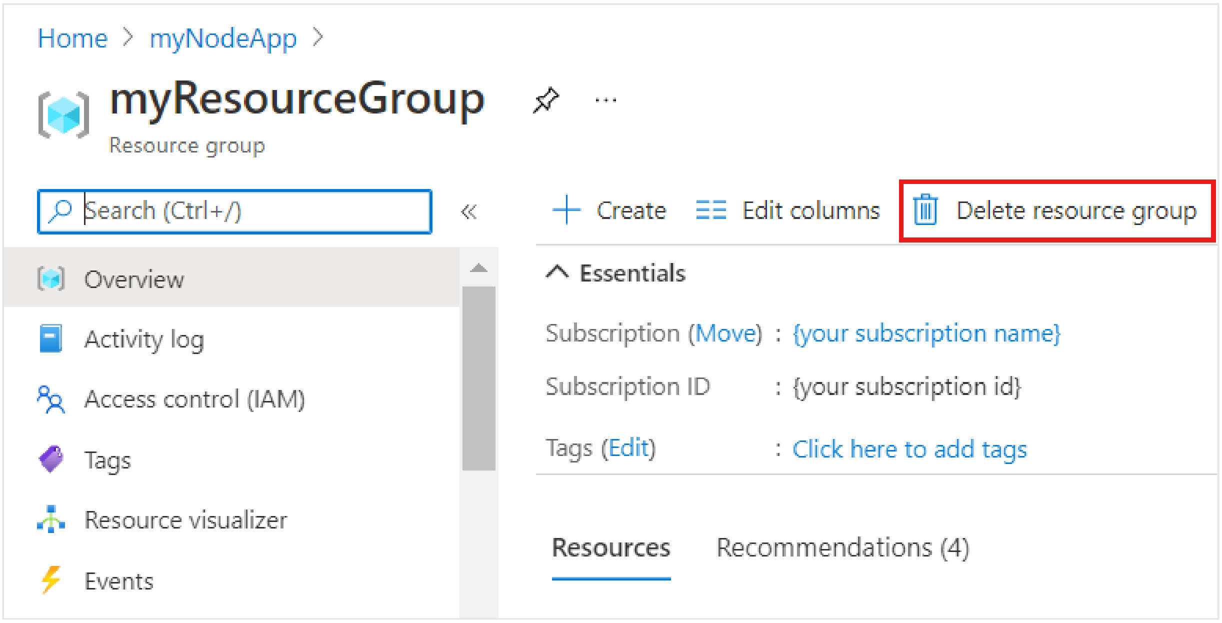
When no longer needed, you can delete the resource group, App service, and all related resources.

1. From your App Service *overview* page, select the *resource group* you created in the Create Azure resources step.



Resource group (change) : myResourceGroup
Status : Running
Location : Central US
Subscription (change) : {your subscription name}
Subscription ID : {your subscription id}
Tags (Edit) : Click here to add tags

2. From the *resource group* page, select **Delete resource group**. Confirm the name of the resource group to finish deleting the resources.



Next steps

Congratulations, you've successfully completed this quickstart!

Tutorial: Node.js app with MongoDB

Configure Node.js app

Check out the other Azure extensions.

- Cosmos DB
- Azure Functions
- Docker Tools
- Azure CLI Tools
- Azure Resource Manager Tools

Or get them all by installing the Node Pack for Azure extension pack.

Recommended content

Deploy a Node.js web app using MongoDB to Azure - Azure App Service

This article shows you have to deploy a Node.js app using Express.js and a MongoDB database to Azure. Azure App Service is used to host the web application and Azure Cosmos DB to host the database using the 100% compatible MongoDB API built into Cosmos DB.

Configure Node.js apps - Azure App Service

Learn how to configure a Node.js app in the native Windows instances, or in a pre-built Linux container, in Azure App Service. This article shows the most common configuration tasks.

Build and deploy a Node.js Express app to Azure Cloud Services (classic)

Use this tutorial to create a new application using the Express module, which provides an MVC framework for creating Node.js web applications.

Deploy from local Git repo - Azure App Service

Learn how to enable local Git deployment to Azure App Service. One of the simplest ways to deploy code from your local machine.

QuickStart: Create a static HTML web app - Azure App Service

Deploy your first HTML Hello World to Azure App Service in minutes. You deploy using Git, which is one of many ways to deploy to App Service.

Deployment options for Azure hosting - Azure

Deploying your apps to Azure hosting services means moving a file or set of files to Azure to be served via an HTTP endpoint.

Deploy apps from GitHub to Azure

Support to deploy apps from GitHub to Azure

Working with Node.js Modules

Learn how to work with Node.js modules when using Azure App Service or Cloud Services.

Show more