Conserve water with the Internet of Things, Part 2

Secure your Node-RED application and automate deployment to IBM Cloud

Bram Havers
Kai Robbin Weller
Robert-Jan Sips
Bill Hymas

July 18, 2017
(First published December 16, 2014)

This second tutorial in a three-part series shows how to build a low-cost plant-monitoring solution based on Raspberry Pi. Secure the Node-RED app and set up automated deployment to IBM Cloud in response to code changes.

Important: IBM Internet of Things Foundation (IoT Foundation) is now named IBM Watson IoT Platform. The IBM Cloud service names have also changed. This article was written using the previous names. The content and images have not been updated. It is provided "as is." Given the rapid evolution of technology, some steps and illustrations may have changed as well.

This second of a three-part tutorial series takes you through construction and configuration of a water-conservation solution — a houseplant soil-moisture monitor — that runs on the Internet of Things. In Part 1 you set up the system's hardware and used the Internet of Things Foundation Starter on IBM Cloud™ to build a Node-RED app that stores incoming sensor data in a Cloudant database and sends it from there to Twitter.

The water-monitoring solution originated with the Poseidon project, a voluntary initiative that aims to reduce global water usage.

Your Node-RED instance is currently accessible by everybody on the Internet, so in this tutorial you'll use the IBM Cloud DevOps Services environment to change the app's security settings. You'll also configure the DevOps Services environment to deploy code changes to your IBM Cloud instance automatically.

“Changes made to the configuration will be versioned (committed) in Git and automatically deployed to your IBM Cloud instance.”
What you'll need to build your solution

- A IBM Cloud account linked to your IBM ID.
- Completion of the steps in Part 1 of this tutorial series.

Step 1. Create a Git repository from your IBM Cloud instance

In this step, you'll set a password for Node-RED, which involves modifying the configuration settings of your IBM Cloud instance. You'll copy the IBM Cloud boilerplate data (configuration) to a Git repository on DevOps Services and enable the autodeploy option. Changes made to the configuration will then be versioned (committed) in Git and automatically deployed to your IBM Cloud instance.

1. Log in to IBM Cloud and click your application in the dashboard.
2. Click the ADD GIT button on the right.
3. Click the Git URL or the EDIT CODE button:
4. Click CONTINUE to confirm that all your changes will automatically be pushed to IBM Cloud.
5. After the DevOps Services setup is done, you might need to click the Git URL again to log in.

Step 2. Configure and deploy Node-RED security

1. Generate an MD5 hash of the password for Node-RED, preferably using a computer that's under your control. (For educational purposes, you can use an MD5 hash generator on the Internet.)
2. In the IBM Cloud dashboard, click your app's EDIT CODE button to go to the web IDE in DevOps Services.
3. In the IDE, click bluemix-settings.js to open the file for editing.
4. In bluemix-settings.js, uncomment the `httpAdminAuth` property and set your user name and the MD5 hash that you just created.

**Step 3. Test security configuration changes**

1. In the web IDE of your DevOps Services project, click the DEPLOY button in the upper left.
2. Click DEPLOY in the Configure Application Deployment dialog box:

![Configure Application Deployment dialog box](image)

The changes to the code are stored automatically.

3. In the IBM Cloud dashboard, click the link that takes you to the home page of your application.

4. Click **Go to your Node-RED flow editor**. If you followed the preceding steps correctly, you'll be asked for a user ID and password:
5. Log in with user ID and password that you created in Step 2.

**Step 4. Commit code to Git and trigger autodeploy**

So far, you've changed the bluemix-settings.js file on your "local" working directory in the DevOps Services IDE. The changes have not been committed to your code repository yet. Now you'll commit the code to Git to make it visible to everybody on your project team. It will be autodeployed (again) to your IBM Cloud instance.

1. In your DevOps Services project, select the Git repository (the second icon on the far left side of the page).
2. On the right side, you can see the bluemix-settings.js file, with the line of code that you changed highlighted:

3. Select the Select All check box and enter a commit message to make clear to other developers on your team what you have changed:

4. Click COMMIT:

By clicking COMMIT, you added (committed) the changes to your "local" working directory, but you still haven't uploaded (pushed) them to the Git repository. This way, you can collect all of your changes (commits) and push them to the repository simultaneously — a convenient
approach if you are working offline for a longer term. But it's also a good practice to commit and push often so others are aware of your changes.

5. Click **PUSH** to push your changes to the Git repository, which also triggers autodeployment on your IBM Cloud instance.

6. Check deployment status:
   a. Click your project name to go to the home of your DevOps Services project.
   b. Click the **BUILD & DEPLOY** button. In the top line of the status list, you can see the last commit deployed:

   ![Deployment Status](image)

5. Validate once more that Node-RED is password-protected.

**Conclusion to Part 2**

In this tutorial, you learned how to secure your Node-RED application. And you created a Git repository for your running IBM Cloud instance. You tested the deployment of changes from DevOps Services to IBM Cloud, and you put these changes under version control. As a last step, you configured autodeployment of commits to your Git repository to your running IBM Cloud environment.

You now have a fully operational system for monitoring the moisture that your houseplants consume and a reliable way to update the code collaboratively. In **Part 3**, you'll enhance the solution by building a client app and API for viewing the sensor data as a graph.
Related topics

- The Internet of Things service
- The Cloudant NoSQL service
- Open toolchains in the Continuous Delivery service
- Internet of Things
- Cloudant
- Node-RED

© Copyright IBM Corporation 2014, 2017
Trademarks
(www.ibm.com/developerworks/ibm/trademarks/)

Conserve water with the Internet of Things, Part 2
Page 6 of 6