V.S.B ENGINEERING COLLEGE

ELECTRONICS AND COMMUNICATION ENGINEERING

Team ID	PNT2022TMID33612
Project Name	IOT Based Smart Crop Protection System For Agriculture

Create Node-Red services

Learning objectives

In this tutorial, we will learn how to:

- Create a Node-RED starter application running in the IBM Cloud
- Secure the application
- Customize the Node-RED Starter Kit by adding additional nodes

Prerequisites

To complete this tutorial, we need an IBM Cloud account.

This tutorial requires an IBM Cloud Pay-As-You-Go account. To upgrade your Lite account, go to our account settings. In the Account Upgrade section, click **Add credit card** to upgrade to a Pay-As-You-Go account, or click **Upgrade** for a Subscription account.
See Upgrading our account for more information.

This Node-RED starter tutorial provides instructions on deploying the app to IBM Cloud Code Engine, which is a fully managed, serverless platform that runs our containerized workloads and manages the underlying infrastructure for us. IBM Cloud Code Engine provides 100000 vCPU seconds per month at no charge. Our Node-RED flow will often scale to 0, which means that you won't incur any charges for light to moderate usage. Review our consumption and confirm our billing on a regular basis.

Estimated time

Steps

1. Find the Node-RED Starter Kit in the IBM Cloud catalog
2. Create our application
3. Enable the Continuous Delivery feature
4. Open the Node-RED application

5. Configure our Node-RED application
6. Add extra nodes to our Node-RED palette

Step1. Find the Node-RED Starter Kit in the IBM Cloud catalog

1. Log in to IBM Cloud.
2. Open the catalog and search for **node-red**.
3. Click on the **Node-RED App** tile.

   This will show us an overview of the Starter Kit and what it provides.

Step2. Create your application

Now we need to create the Node-RED starter application.

1. On the *Create* tab, a randomly generated **App name** will be suggested. Either accept that default name or provide a unique name for your application. This will become part of the application URL.

2. The Node-RED starter application requires an instance of the **Cloudant database service** with IBM Cloud IAM and Cloudant credentials to store our application flow configuration. Select the region the service should be created in and what pricing plan it should use.

3. Click the **Create** button to continue. This will create our application and, if necessary a Cloudant database service instance, but it is not yet deployed to IBM Cloud.

Step3. Enable the Continuous Delivery feature

1. On the next screen, click the **Deploy your app** button to enable the *Continuous Delivery* feature for our application.

2. On the next screen, click the **Code Engine** tile.

3. Scroll down after selecting the **Code Engine** tile. We will need to create an **IBM Cloud API** key to allow the deployment process to access your resources. Click the **New** button to create the key. A message dialog will appear. We can accept the default values and confirm to close the dialog.

4. Select the **Region** and **Container registry region**, to deploy your application to. This should match the region created our Cloudant instance in.

5. Provide a unique **Project** name or accept the default 'project-name'

   Click **Next** to continue.

6. Configure the **DevOps toolchain** by selecting the **region** it should be created in. Again, try to match the region we selected previously.

7. After a few moments, the **Deployment Automation** section will refresh with the details of our newly created Delivery Pipeline. The Status field of the pipeline will eventually show **In progress**. That means our application is being built and deployed.

8. The Deploy stage will take a few minutes to complete. we can click on the ci-pipeline **Status** link to check the progress of the Delivery Pipeline. Eventually the Deploy stage will display a green checkmark and a **Success** message to show it has passed. This means our Node-RED starter application is now running.

Step 4. Open the Node-RED application

Now deployed our Node-RED application, let's open it up! May have to refresh our page.

On the application details page, we should now see the **App URL**, **Source** and **Deploymenttarget** fields filled in.

Click on the **App URL** to open up our Node-RED application in a new browser tab.

Step 5. Configure our Node-RED application

The first time we open your Node-RED app, we'll need to configure it and set up security.

1. A new browser tab will open with the Node-RED start page.

2. On the initial screen, click **Next** to continue.

3. Secure our Node-RED editor by providing a **username** and **password**. If we need to change these at any point, we can either edit the values in the Cloudant database, or override them using *environment variables*. The documentation on nodered.org describes how to do this. Click **Next** to continue.

4. The final screen summarizes the options you've made and highlights the environment variables we can use to change the options in the future. Click **Finish** to proceed.

5. Node-RED will save our changes and then load the main application. From here we can click the **Go to our Node-RED flow editor** button to open the editor.

   The Node-RED editor opens showing the default flow.

Step 6. Add extra nodes to your Node-RED palette

The recommended approach is to edit your application's package.json file to include theadditional node modules and then redeploy the application.

This step shows how to do that in order to add the **node-red-dashboard** module.

1. On our application's details page, click **Source** url. This will take us to a git repository where we can edit the application source code from your browser.

2.. Scroll down the list of files and click on **package.json**. This file lists the module dependencies of your application.

3. Click the **Edit** button

4. Add the following entry to the top of the dependencies section (1):

5. ```
   "node-red-dashboard": "2.x",
   ```

   Add a **Commit message** (2) and click **Commit changes** (3)

6. At this point, the Continuous Delivery pipeline will automatically run to build and deploy that change into your application. If you view the Delivery Pipeline you can watch its progress. The Build section
   ```

shows the last commit made and the Deploy section shows the progress of redeploying theapplication.

7. Once the Deploy stage completes, our application will have restarted and now have the node-red-dashboard nodes preinstalled.

Summary

Now created a Node-RED application that is hosted in the IBM Cloud. We have also learned how to edit the application source code and automatically deploy changes.