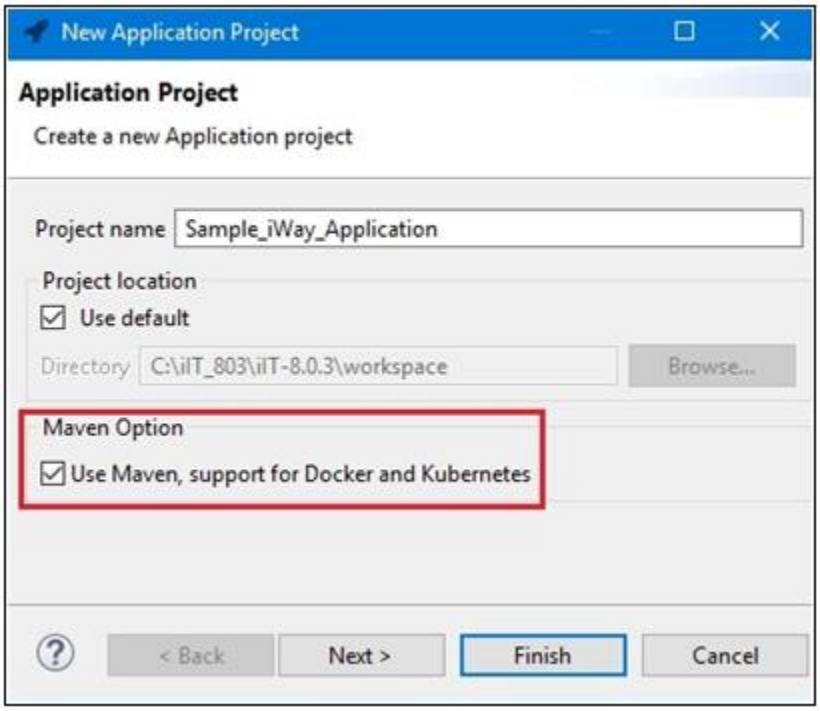
Deploying an iWay Application to a Docker Container

Packaging an iWay application into a Docker image enables streamlined deployments into a cloud-based environment of your choice.

This how-to includes the following topics:

Overview

When an iWay application is created in iWay Integration Tools (iIT), you can select an option to enable Docker support in the New Application Project dialog box, as shown in the following image.



Selecting *Use Maven, support for Docker and Kubernetes* under the Maven Option, creates all of the required structures, such as a Docker file and a Project Object Model (POM) file. The application image can easily be created using the Maven plug-in within iIT.

You can use the Docker Explorer in iIT to monitor and manage your Docker instances. Full support for Kubernetes enables you to take a Docker image and easily deploy it into Docker, or cloud-based environment. Kubernetes is an open-source system for automating deployment, scaling, and management of containerized applications.

Installing Docker

This section describes how to install a Docker environment locally on your system.

Note: If your organization already has a Docker environment set up, the following steps may not be required. The installation steps for Docker will vary depending on the Docker version and edition you are installing for local testing. Consult the Docker documentation for your specific version for installation steps.

1. Download the *Docker Desktop for Windows* application from the following Docker website:

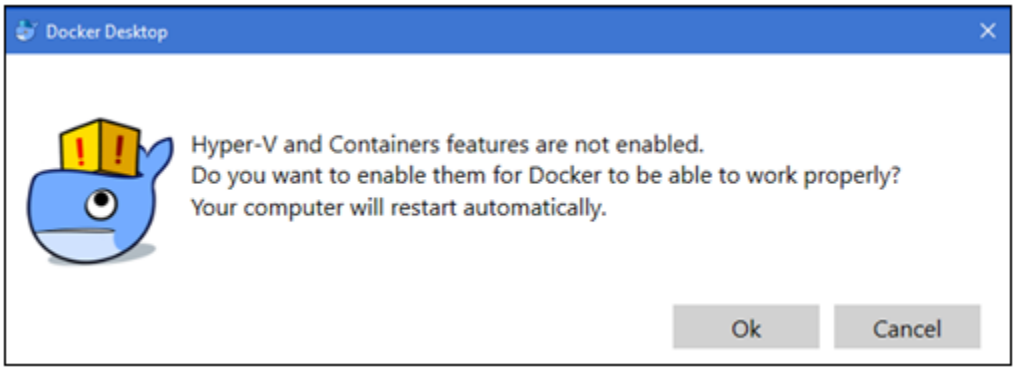
   https://hub.docker.com/editions/community/docker-ce-desktop-windows

2. Double-click the *Docker for Windows Installer* executable (.exe) to run the installer.

   Follow the installation wizard prompts. You can choose to use a Windows container instead of a Linux container. For the purposes of this How-to, a Windows-based container is used.

3. After the Docker installation is complete, you are prompted to sign out and then sign back into Windows.

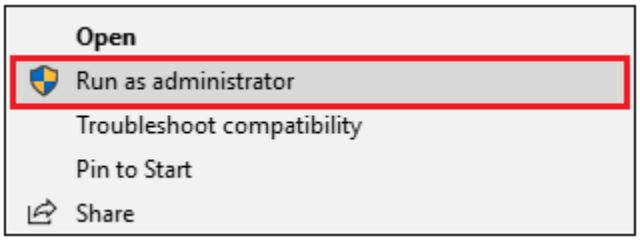
   After you sign back into Windows, you may be prompted to enable the Hyper-V and Containers features, as shown in the following image.



4. Click *OK*.

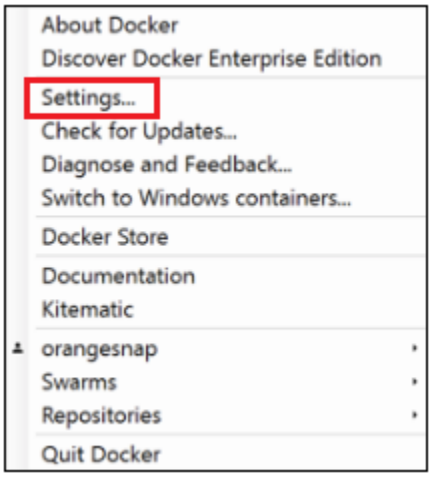
5. Reboot your system, which may take a few minutes.

6. After rebooting the system, start the Docker application using the *Run as administrator* option.



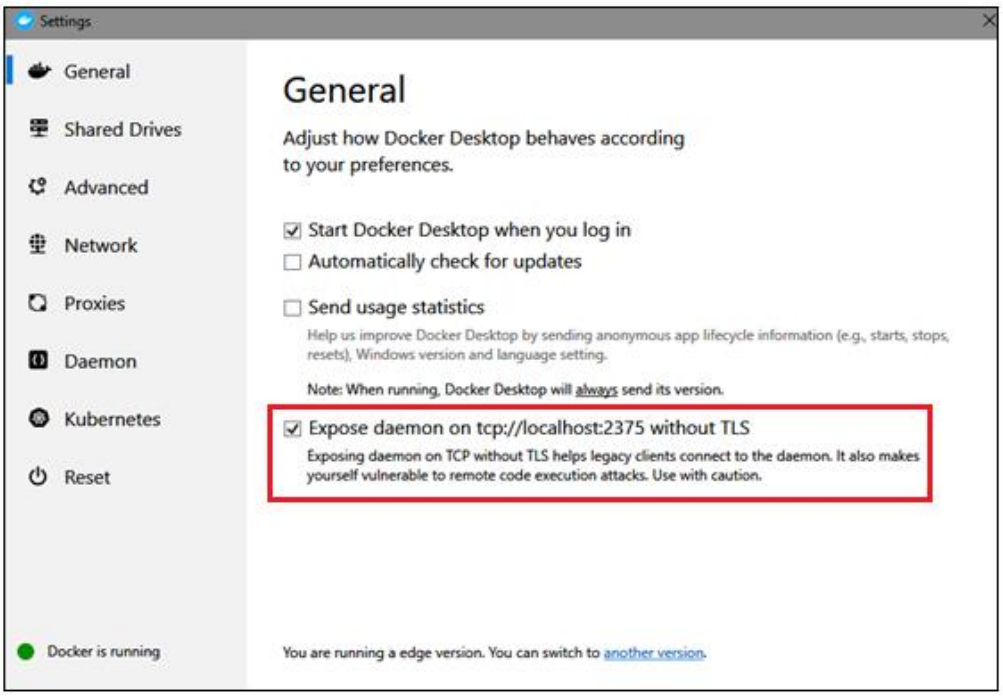
When Docker starts, you will see the whale icon on the Windows taskbar, as shown in the following image.

7. Right-click the whale icon and select *Settings* from the context menu, as shown in the following image.



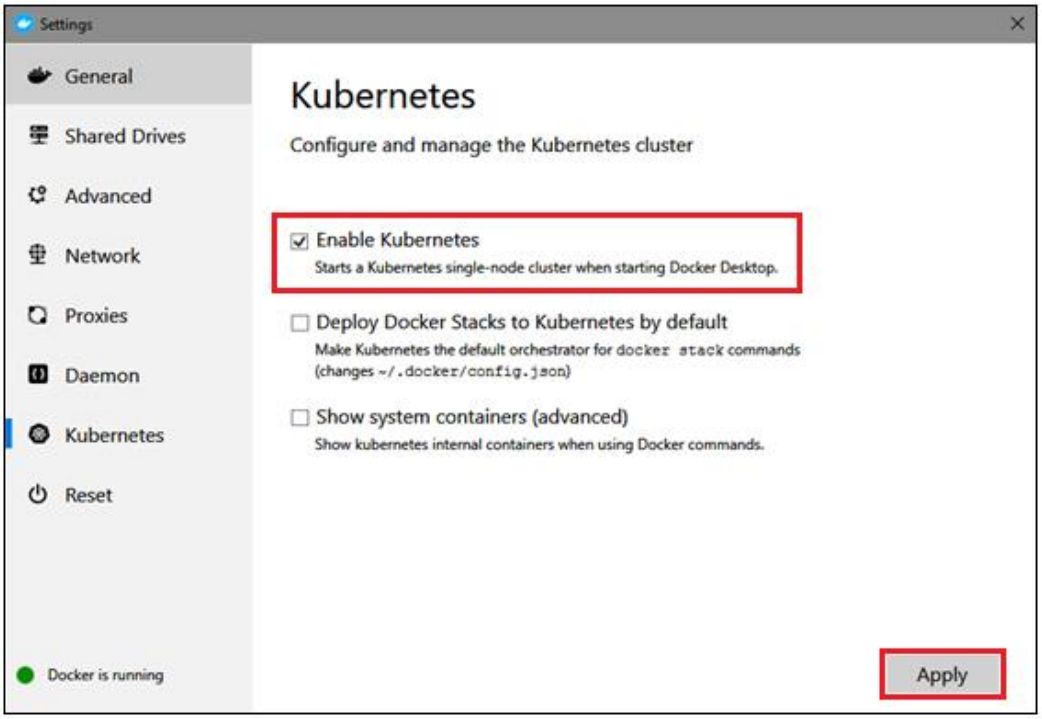
The Settings dialog box opens.

8. Under the General preferences area, select the *Expose daemon on tcp://localhost:2375 without TLS* option, as shown in the following image.



9. Click *Kubernetes* in the left pane of the Settings dialog box.

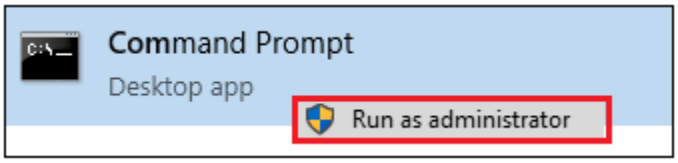
10. Under the Kubernetes preferences, select the *Enable Kubernetes* option, as shown in the following image.



11. Click *Apply*.

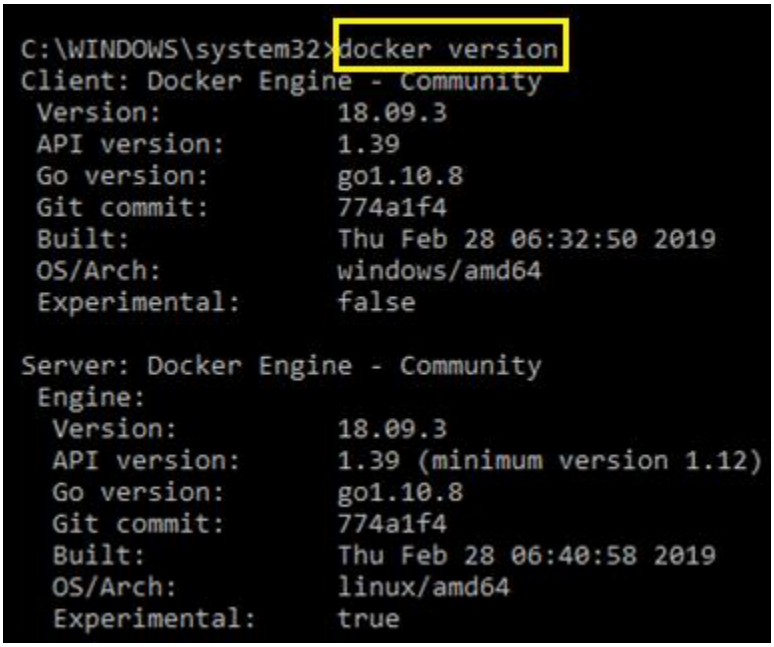
    This process may take a few minutes as all of the services are being enabled.

12. Open a Windows Command Prompt with the *Run as administrator* option, as shown in the following image.

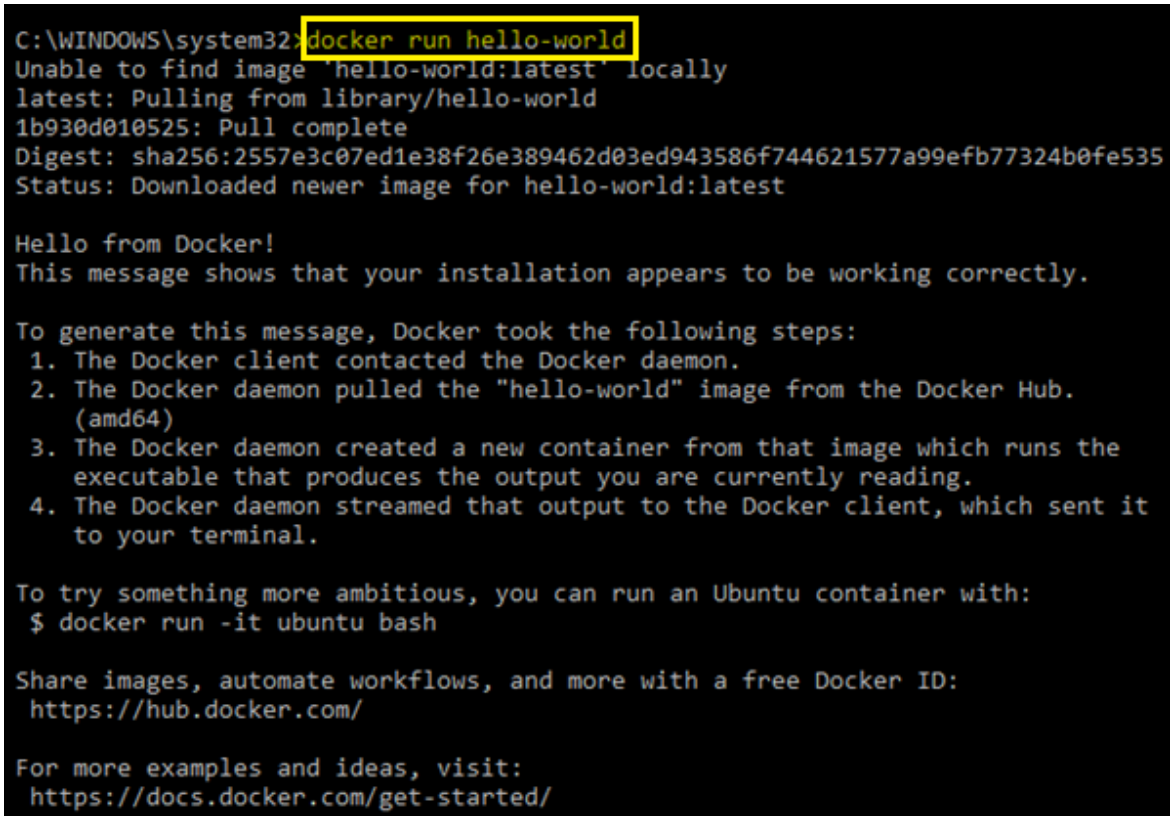


13. Type test commands to test your Docker installation and ensure that Docker is functional.

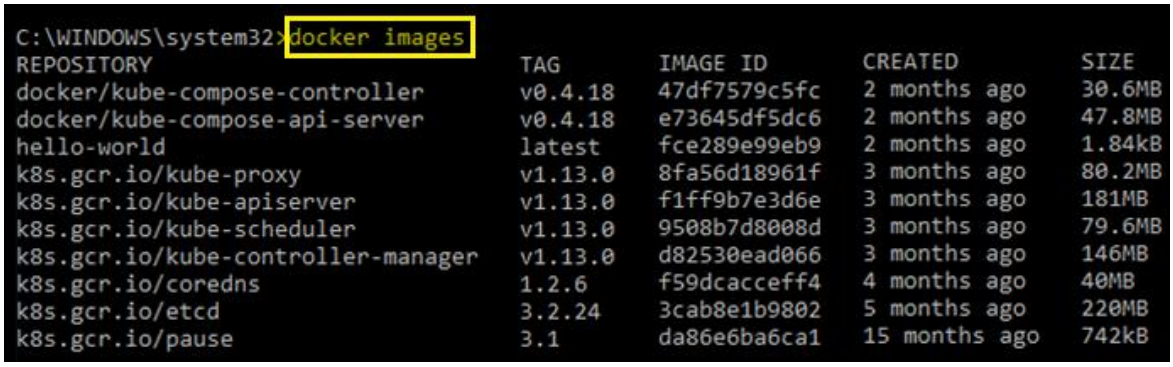
    a. Type *docker version*, as shown in the following image.



b. Type *docker run hello-world*, as shown in the following image**.**



c. Type *docker images*, as shown in the following image**.**



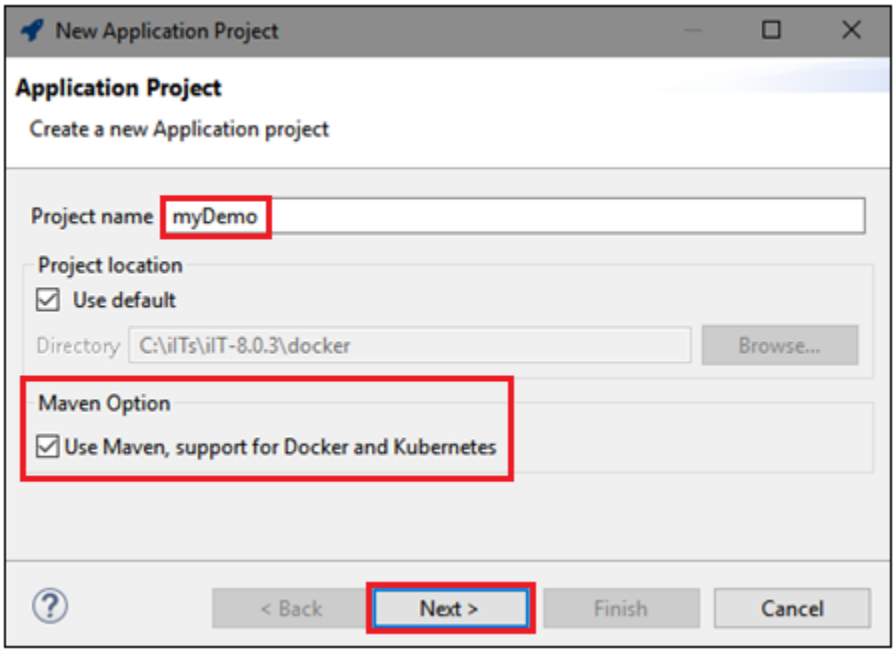
Creating a Sample iWay Application With Docker Support

This section describes how to create an iWay application, which can be deployed as a Docker container. A simple API will be used as an example to validate that the container is deployed successfully.

1. Open iWay Integration Tools (iIT) in the desired workspace.

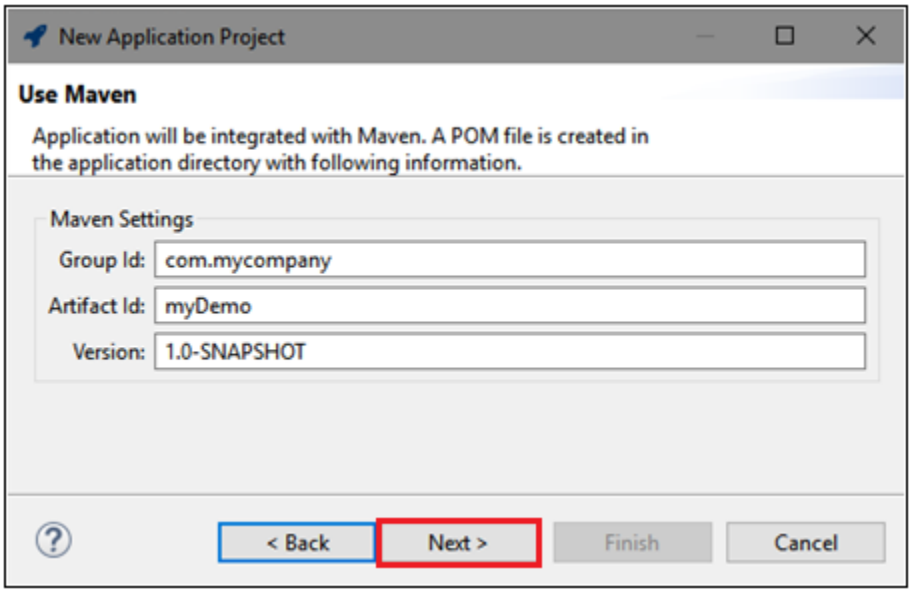
2. Create a sample iWay application project (for example, *myDemo*) with support for Docker and Kubernetes.

During the project creation, ensure that the *Use Maven, support for Docker and Kubernetes* option is selected in the New Application Project dialog box, as shown in the following image.



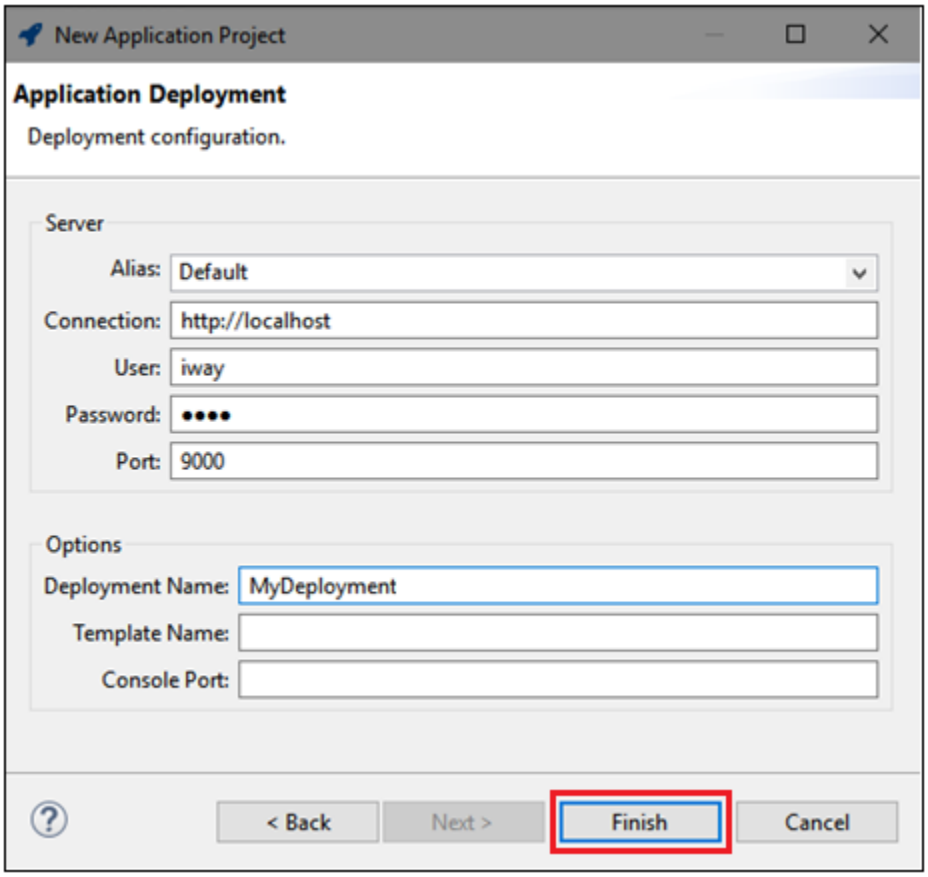
3. Click *Next*.

4. The Use Maven dialog box opens, as shown in the following image.



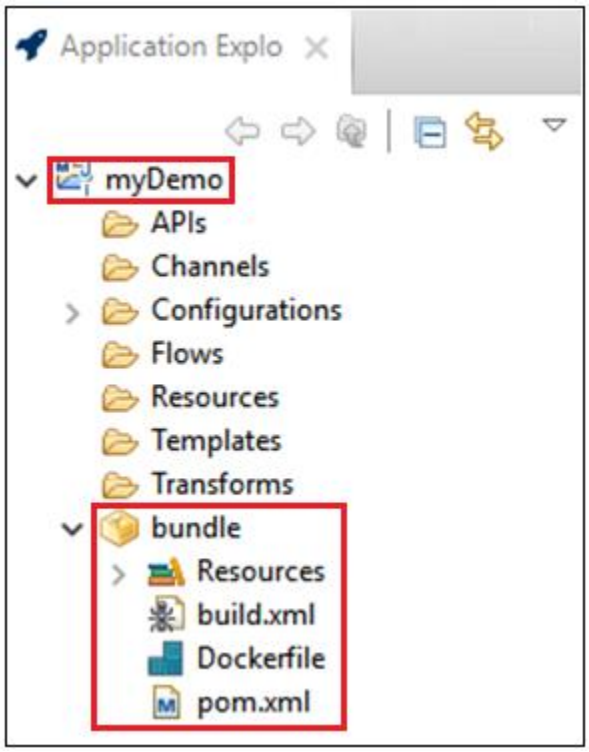
5. Click *Next*.

The Application Deployment dialog box opens, as shown in the following image.



6. Provide values for the required parameters, and then click *Finish*.

   The application is created and you can browse its components, as shown in the following image.

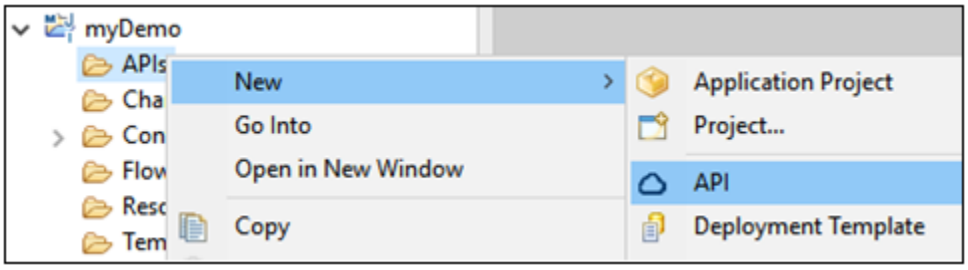


   If you expand the *bundle* folder, which represents the application, you will see the *Dockerfile* and *pom.xml* files, which were previously created for use.

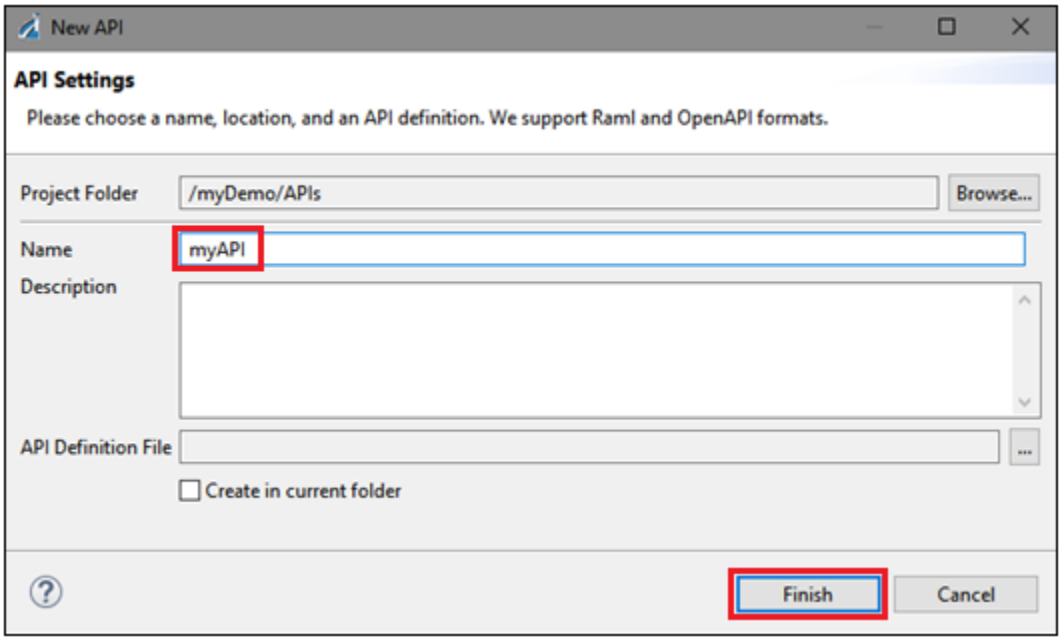
Configuring an API to Test the Docker Container

This section describes how to configure a simple API to test your Docker container. Note that for demonstration purposes only, this API will be created without any externalized definitions, such as RESTful API Modeling Language (RAML) or Swagger. Production applications should have externalized definitions for the endpoints.

1. Right-click the *APIs* subfolder in your iWay application, select *New*, and then select *API* from the context menu, as shown in the following image.

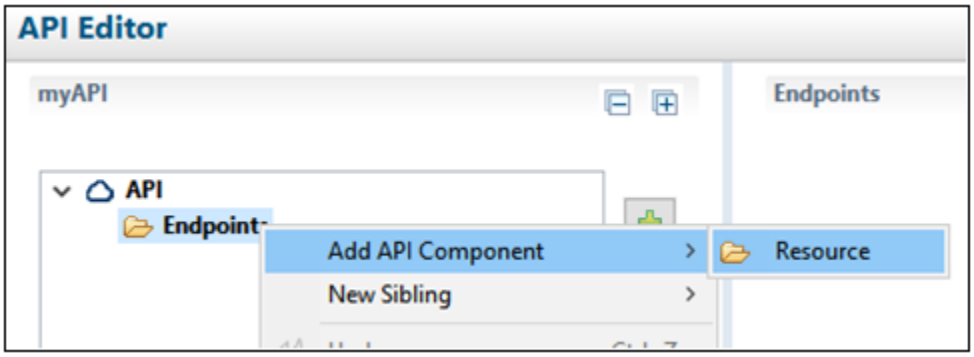


The API Settings dialog box opens, as shown in the following image.



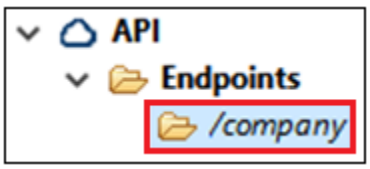
2. Type a name for your new API (for example, *myAPI*), and then click *Finish*.

   The API Editor opens, as shown in the following image.

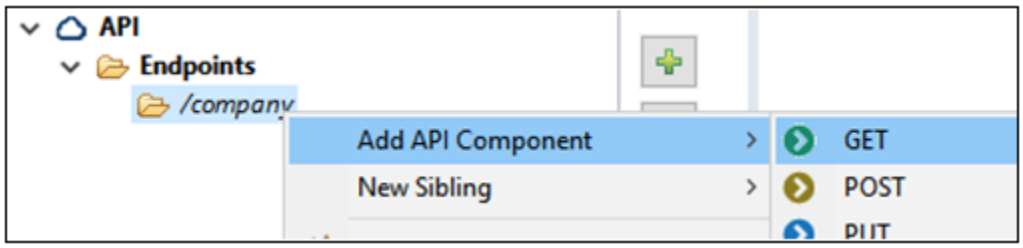


3. Right-click *Endpoints,* select *Add API Component,* and then select *Resource* from the context menu.

4. Rename the added resource to */company*, as shown in the following image.

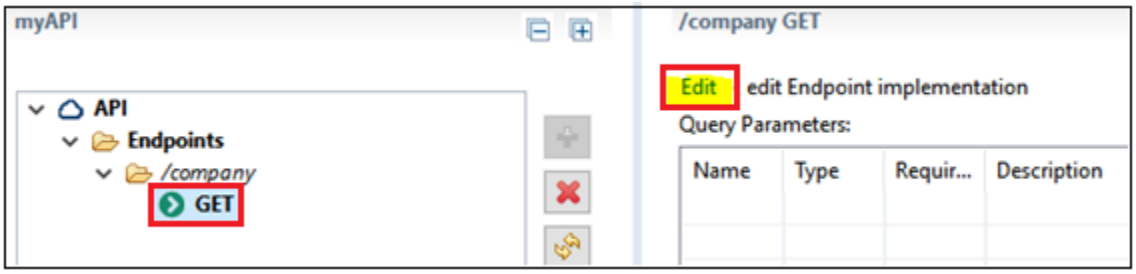


5. Right-click the */company* resource, select *Add API* Component, and then select *GET* from the context menu, as shown in the following image.



This will add a GET method for this API endpoint.

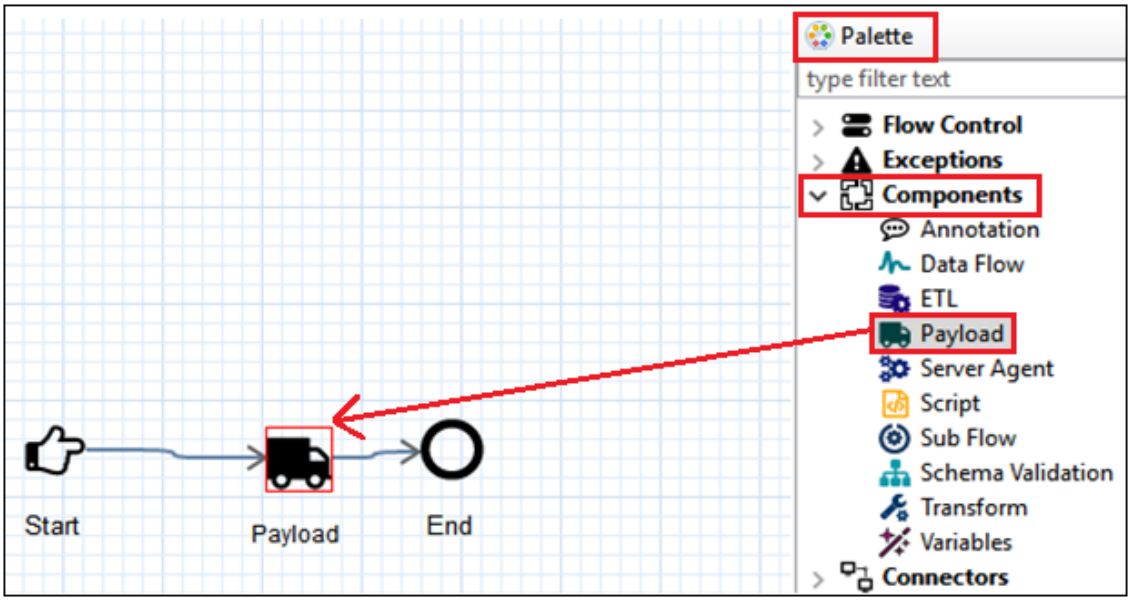
6. Click the *Edit* hyperlink for the */company GET* action, as shown in the following image.



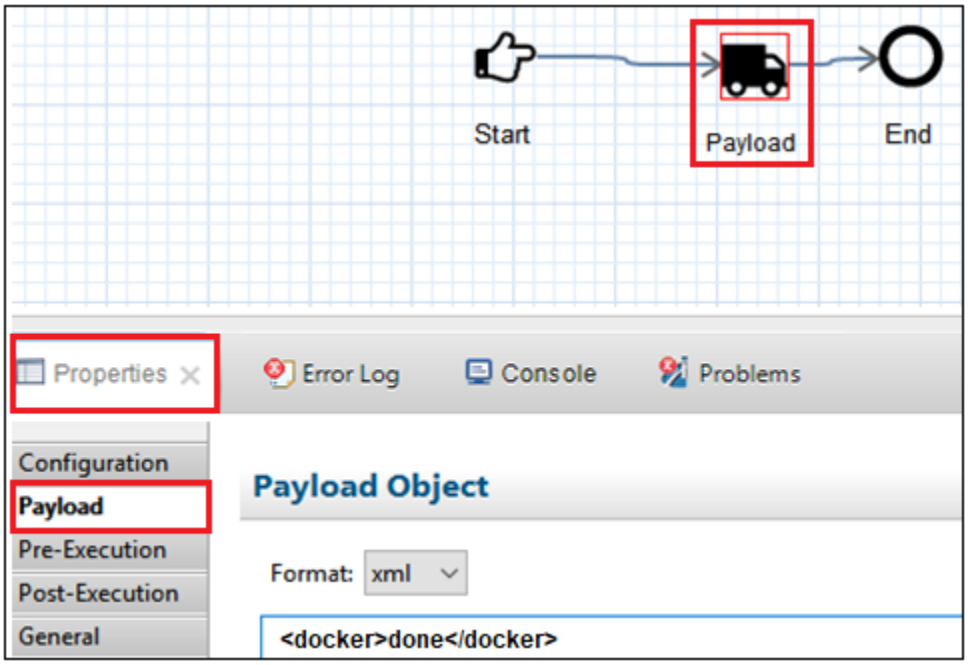
This enables you to create or assign a process flow representing the business logic to be executed for this endpoint.

7. For demonstration purposes, reply to this GET call only with a static message, by performing the following steps:

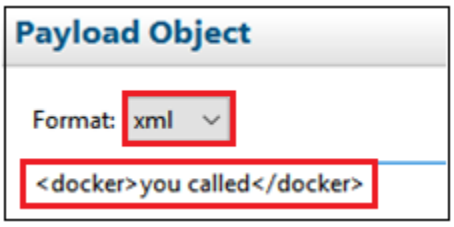
   a. From the Palette, under the *Components group*, select *Payload,* and drag it onto the line between the Start and End objects, as shown in the following image.



   b. Click the *Payload* object and then select the *Payload* tab under *Properties*, as shown in the following image.

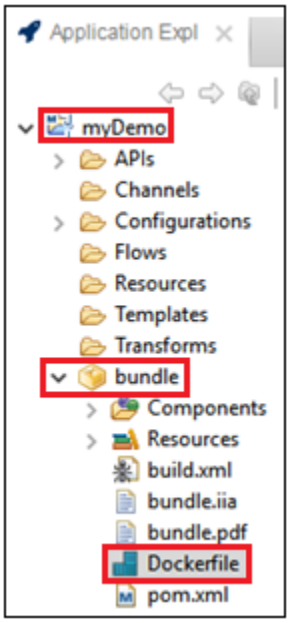


c. In the Payload Object section, specify the format of the payload (for example, XML) and a sample text message, as shown in the following image.

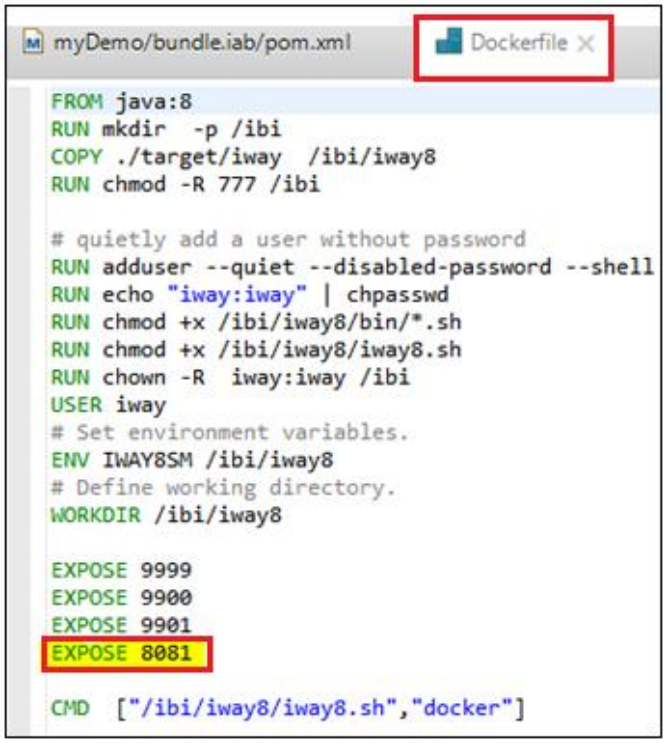


8. Click the *Save All* icon  to save the changes made to your process flow.

9. Double-click *Dockerfile*, which is located in the *bundle* subfolder of your iWay application project, as shown in the following image.



The Dockerfile opens as a new tab in your main workspace area, as shown in the following image.



10. Add the line *EXPOSE 8081* to the Dockerfile.

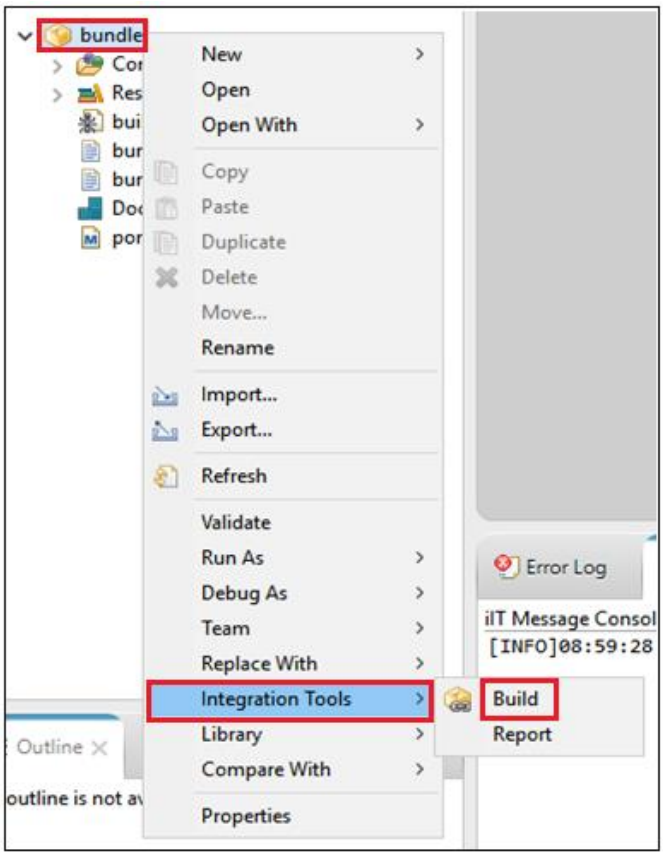
    This line is required since the API is running internally on default port 8081. The port for the API can be configured in the settings area of the APIs.

11. Click the *Save All* icon to save your work.

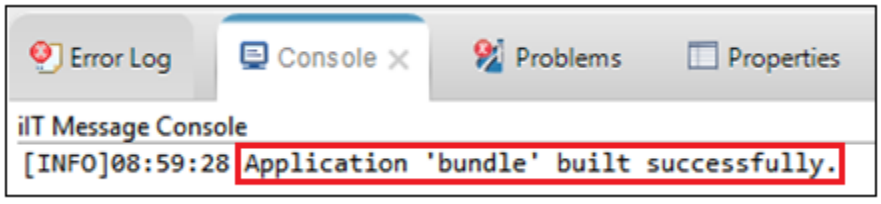
Building the Application

This section describes how to build the iWay application to create a deployable package.

1. Right-click the *bundle* subfolder of your iWay application project, select *Integration Tools,* and then *Build* from the context menu, as shown in the following image.

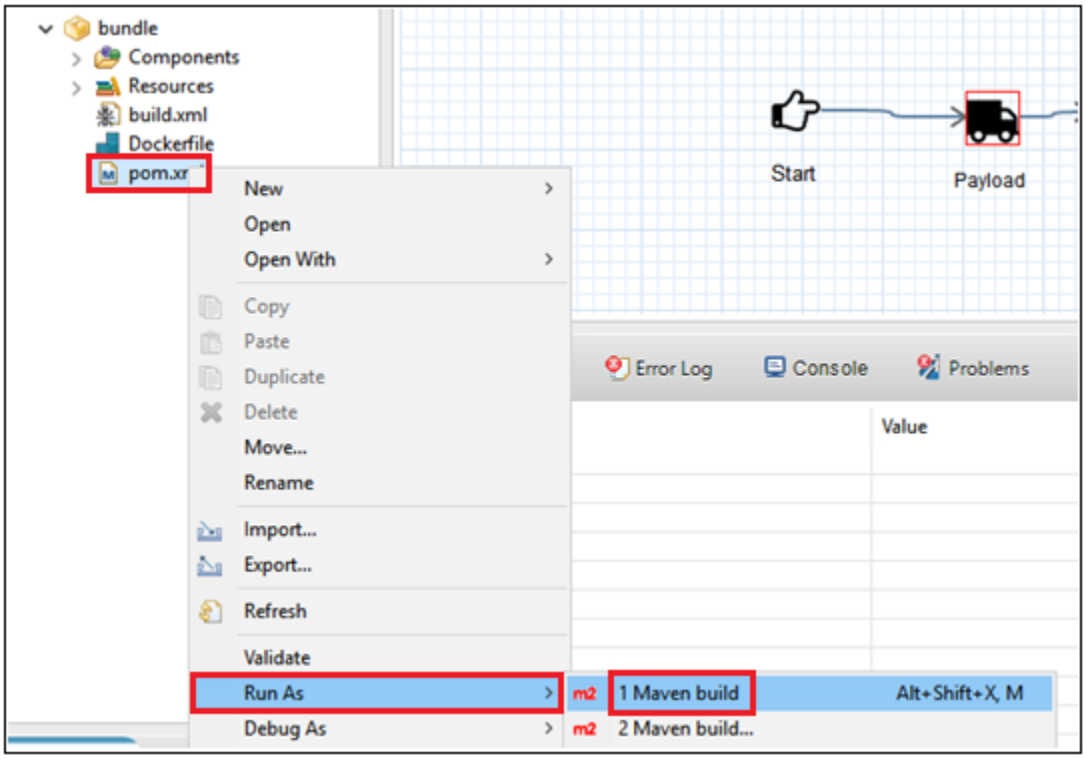


An *Application 'bundle' built successfully* message should appear in the Console tab after the build is complete, as shown in the following image.

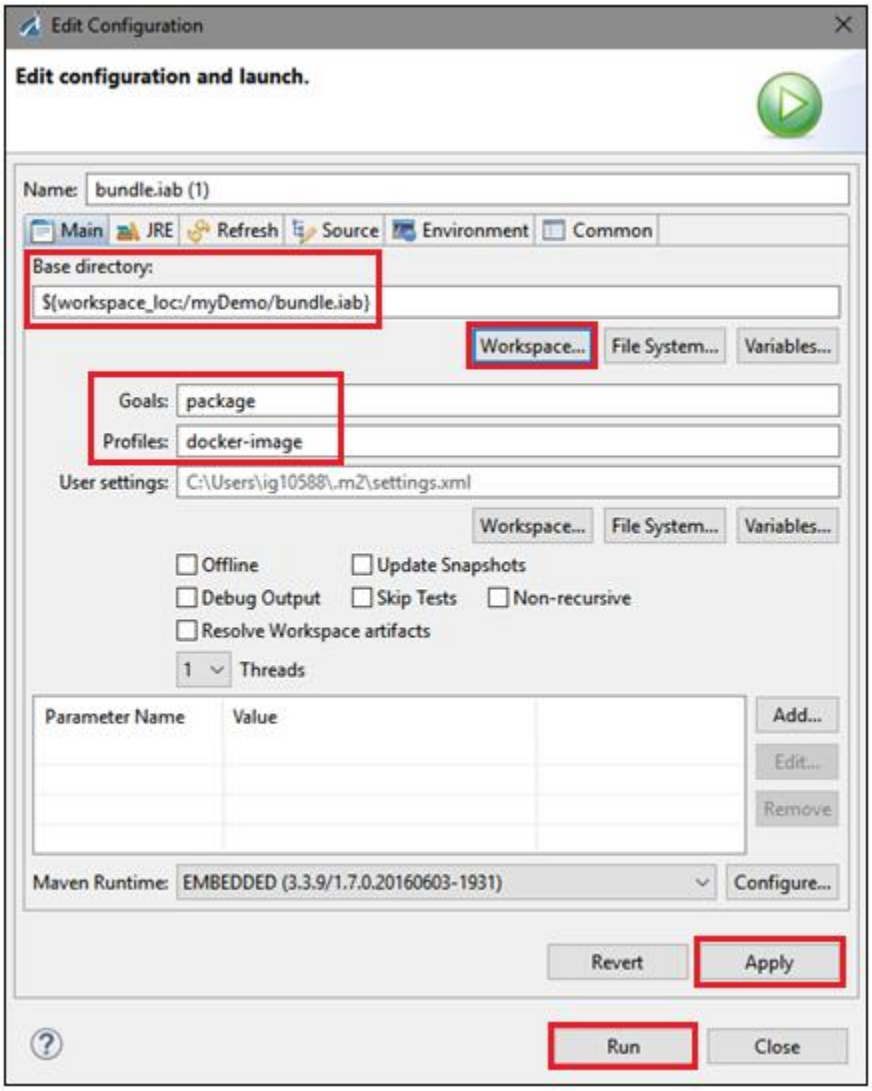


You are now ready to package your application using the Maven build script.

2. Right-click the *pom.xml* file in the *bundle* subfolder of your iWay application project, select *Run As*, and then select *Maven build* from the context menu, as shown in the following image.



The Edit Configuration dialog box opens, as shown in the following image.



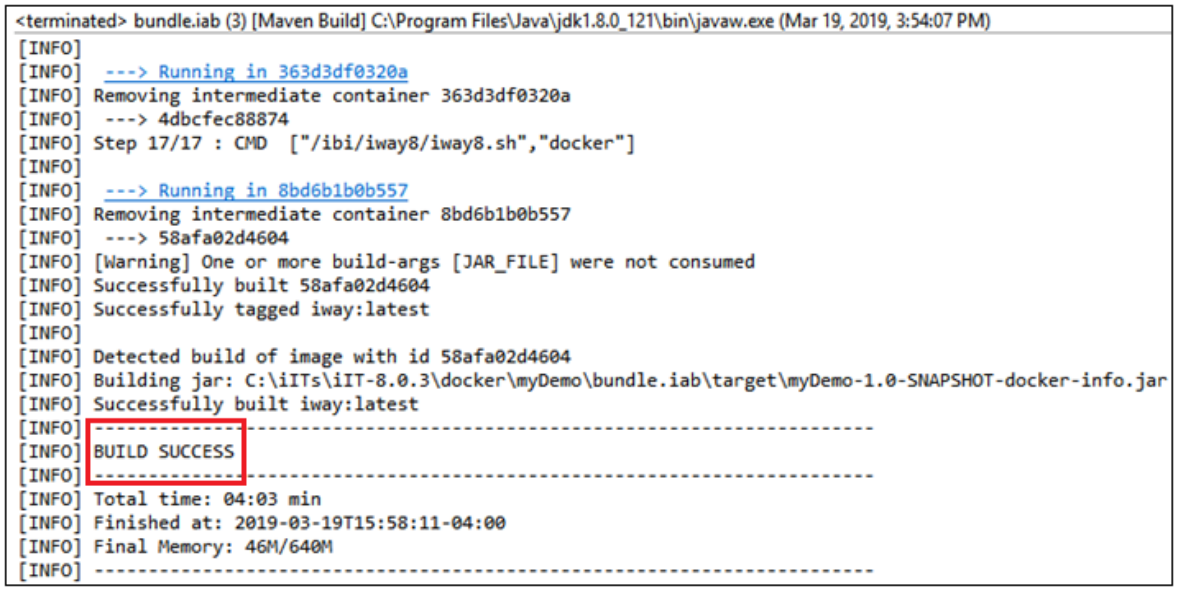
3. Provide values for the parameters, as shown in the following table.

Parameter	Value
Base directory	Click the *Workspace* button and browse to the location of the .iab file. For example: `${workspace_loc:/myDemo/bundle.iab}`
Goals	`package`
Profiles	`docker-image`

4. Click *Apply* and then click *Run*.

   Note: The first time your application is built, all of the components have to be acquired from the online repository. This process may take a few minutes.

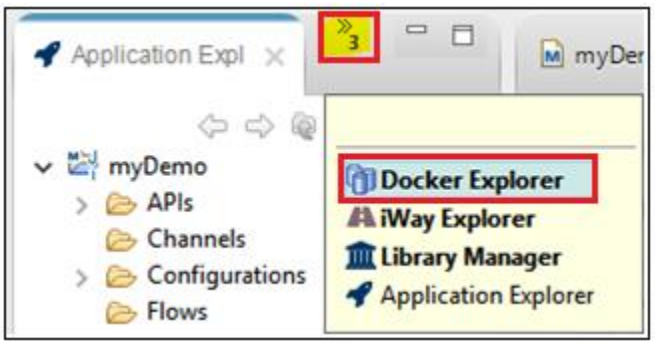
5. After the build is complete, you will see a success message, as shown in the following image.



Connecting to the Docker Explorer

This section describes how to connect to the Docker Explorer in iIT.

1. From the available views in iIT, select *Docker Explorer*, as shown in the following image.

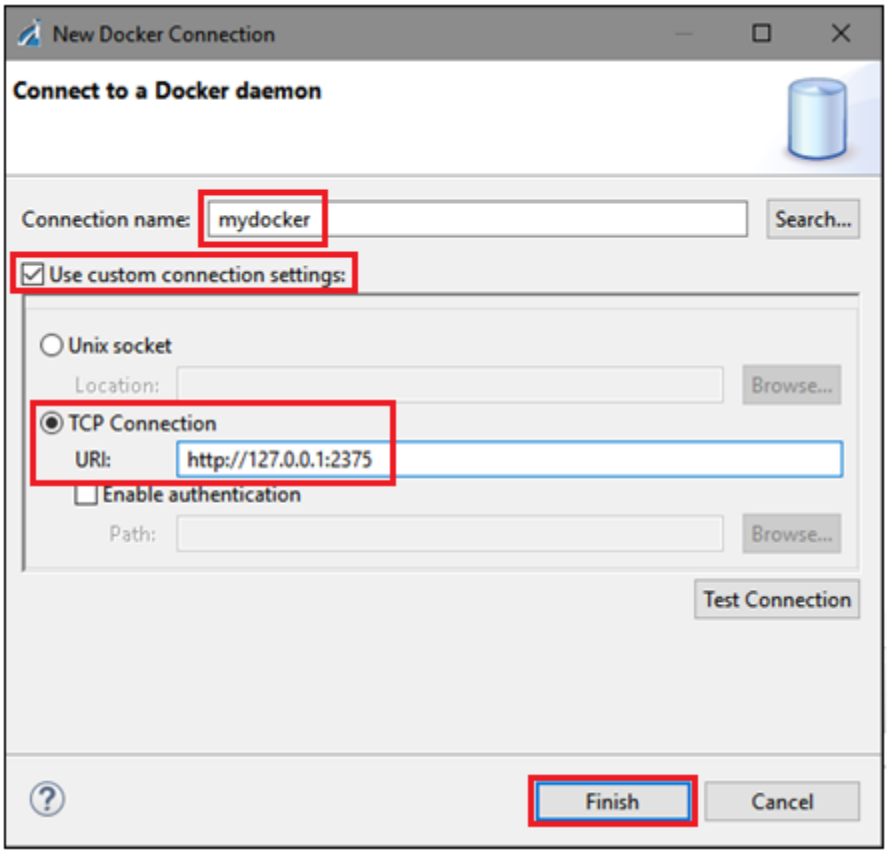


   Note: The first time you open the Docker Explorer, you need to create a new connection to your Docker environment. Since a local instance of Docker is being used in this How-to for demonstration purposes, connect to it locally.

2. In the New Docker Connection dialog box:
    a. Type a name for your new Docker connection (for example, *mydocker*).

    b. Click the *Use custom connection settings* checkbox.

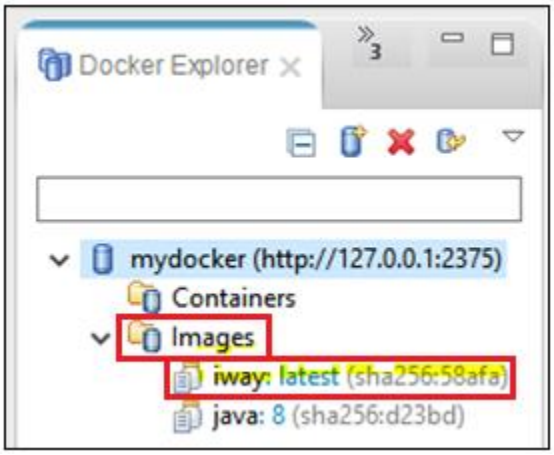
    c. Click *TCP Connection* and specify the URI as **http://127.0.0.1:2375**, as shown in the following image.



Note: This URI is the same URI being used for the Docker installation.

3. Click *Finish*.

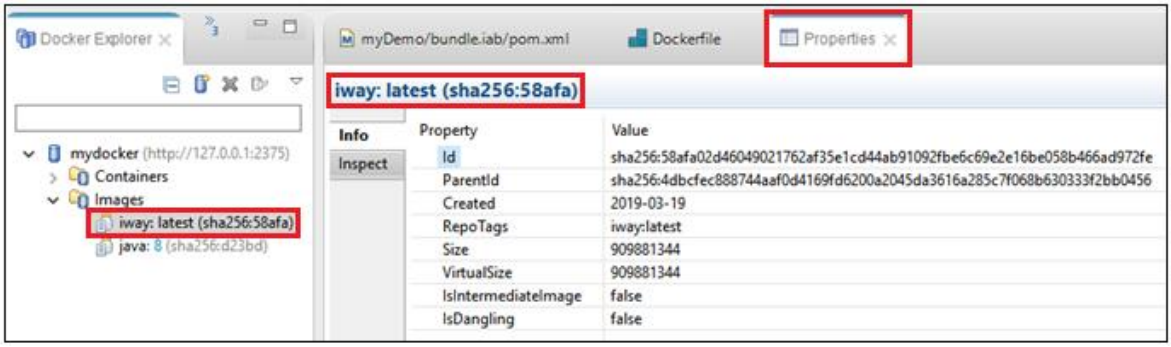
   Your Docker connection provides two folders (Containers and Images) in the Docker Explorer, as shown in the following image.



   Under the *Images* folder, you should see *iway:latest***,** which is the Docker image you have just created.

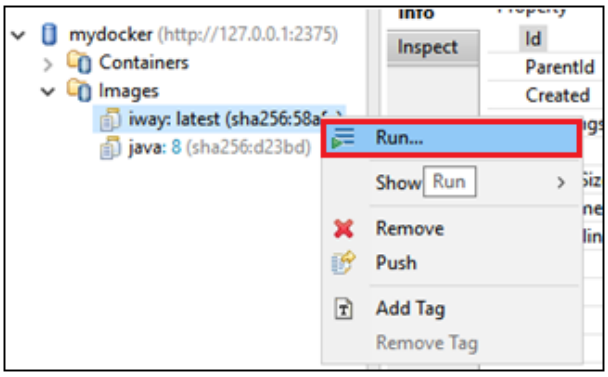
4. Right-click *iway:latest*, select *Show In,* and then select *Properties*.

   Detailed information for your selected Docker image is displayed in the Properties tab, as shown in the following image.

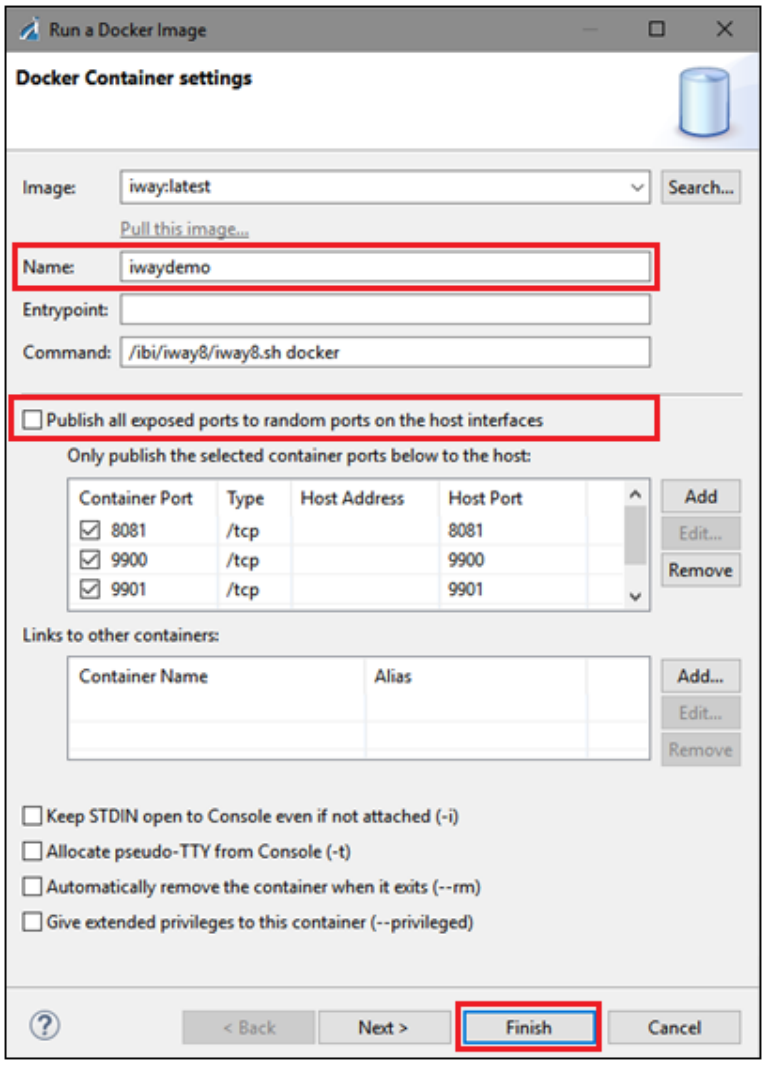


   You are now ready to run this Docker image as a container.

5. Right-click *iway:latest* and then select *Run* from the context menu, as shown in the following image.



The Run a Docker Image dialog box opens, prompting you for additional information, as shown in the following image.

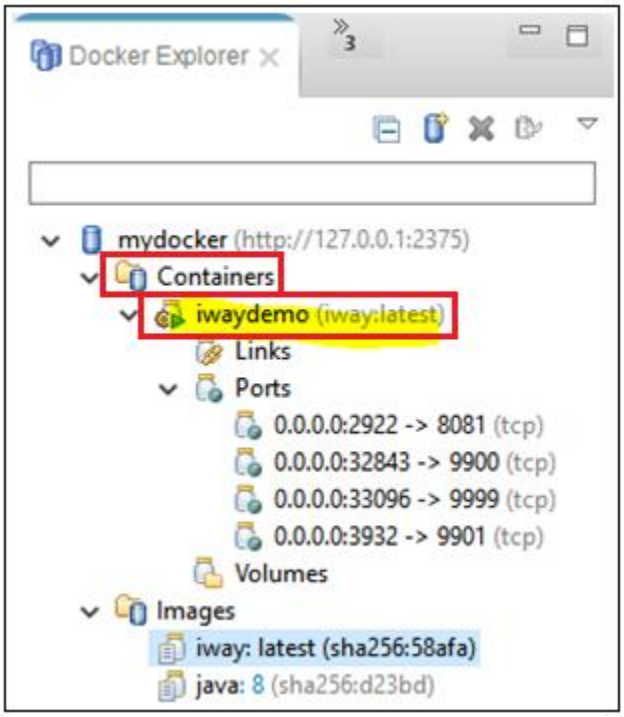


6. Provide the deployment name (for example *iwaydemo*).

7. Clear the *Publish all exposed ports to random ports on the host interfaces* checkbox, since specific ports will be used for testing.

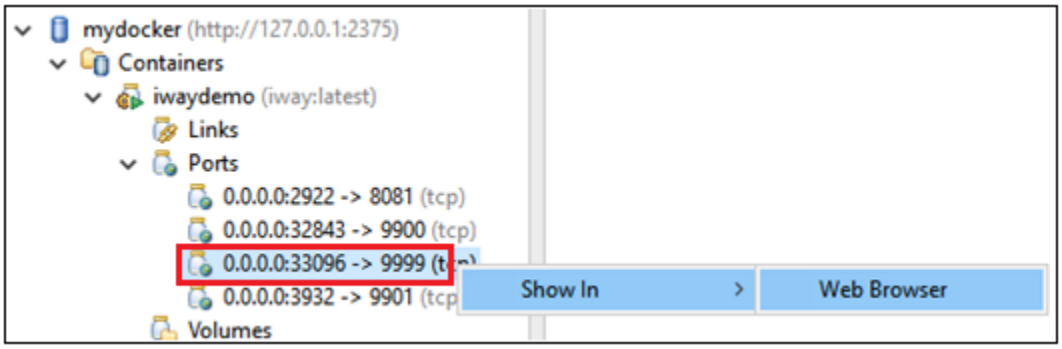
8. Click *Finish*.

9. Expand the *Containers* folder, as shown in the following image.

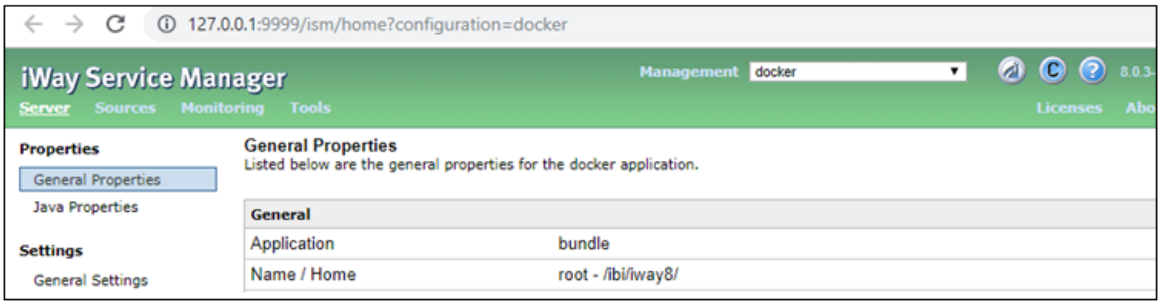


Your Docker container including all of the relevant information is displayed.

10. Right-click port *9999*, select *Show In*, and then select *Web Browser* from the context menu, as shown in the following image.



The iWay Service Manager (iSM) Administration Console opens, as shown in the following image.

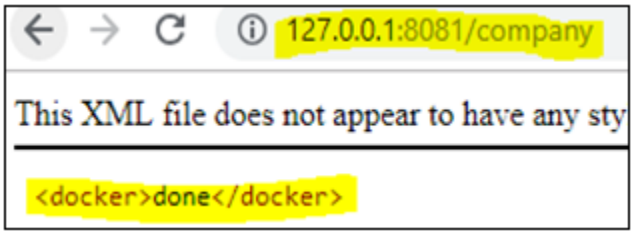


11. Ensure that you can log in to the iSM Administration Console using the following credentials:

    - User name: **admin**
    - Password: **admin**

12. In a browser, you can test the API call by typing the following URL:

    `http://127.0.0.1:8081/company`



Note: You can also use any external API testing tool of your choice.