**Azure Linux App Service HOL**

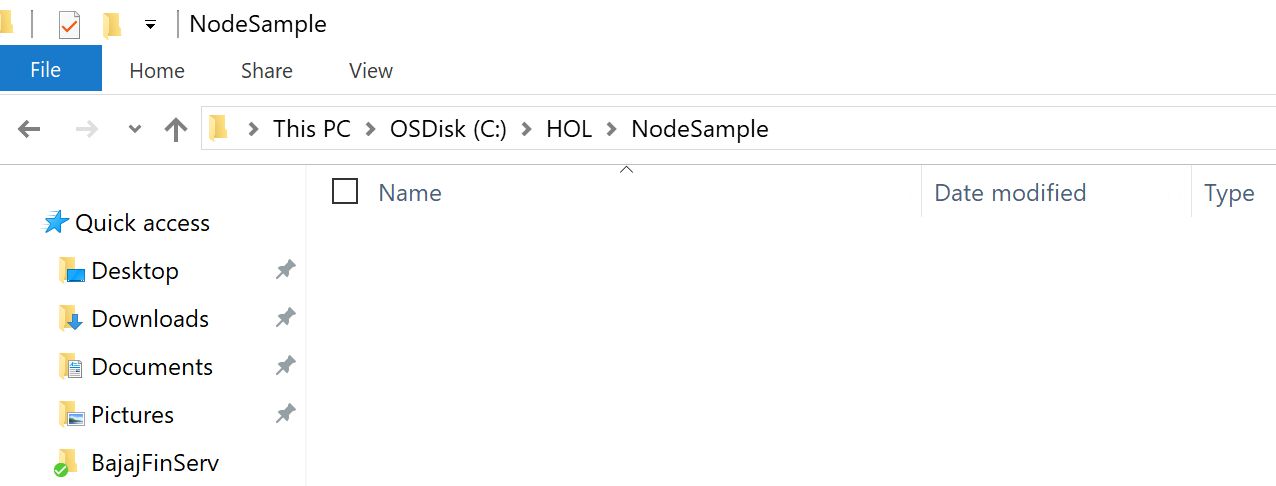
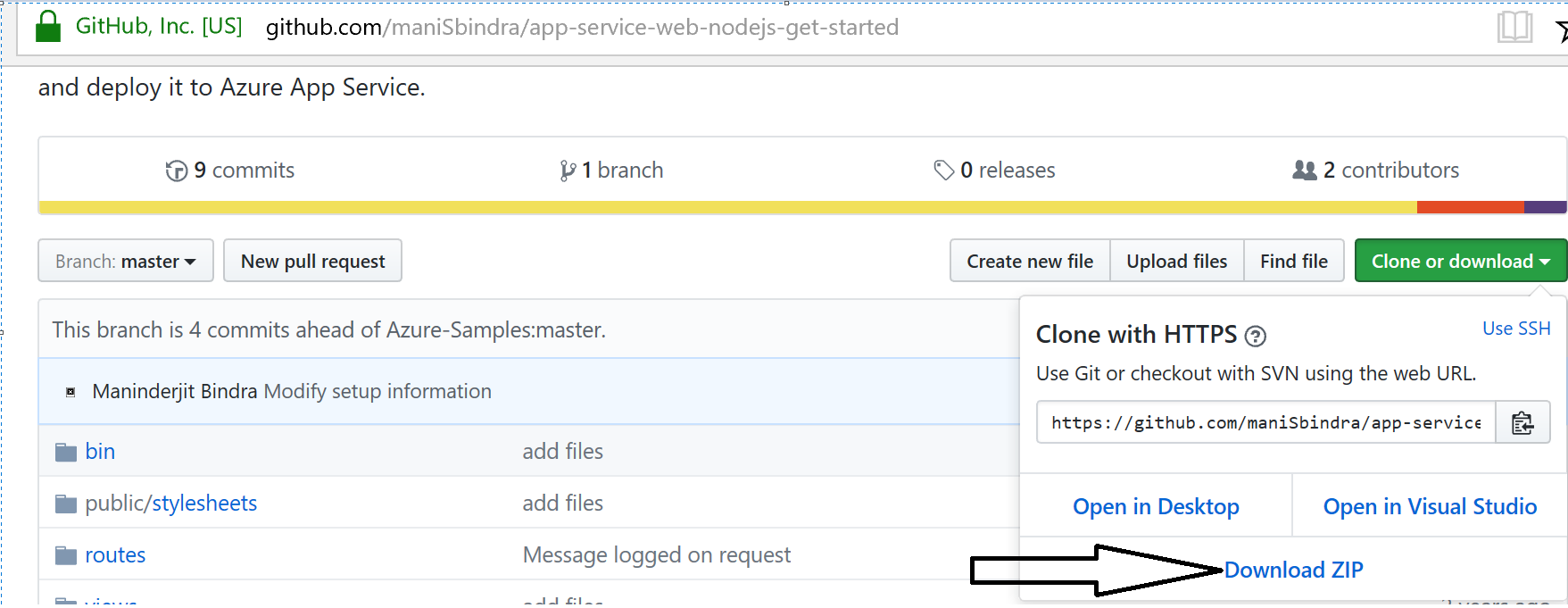
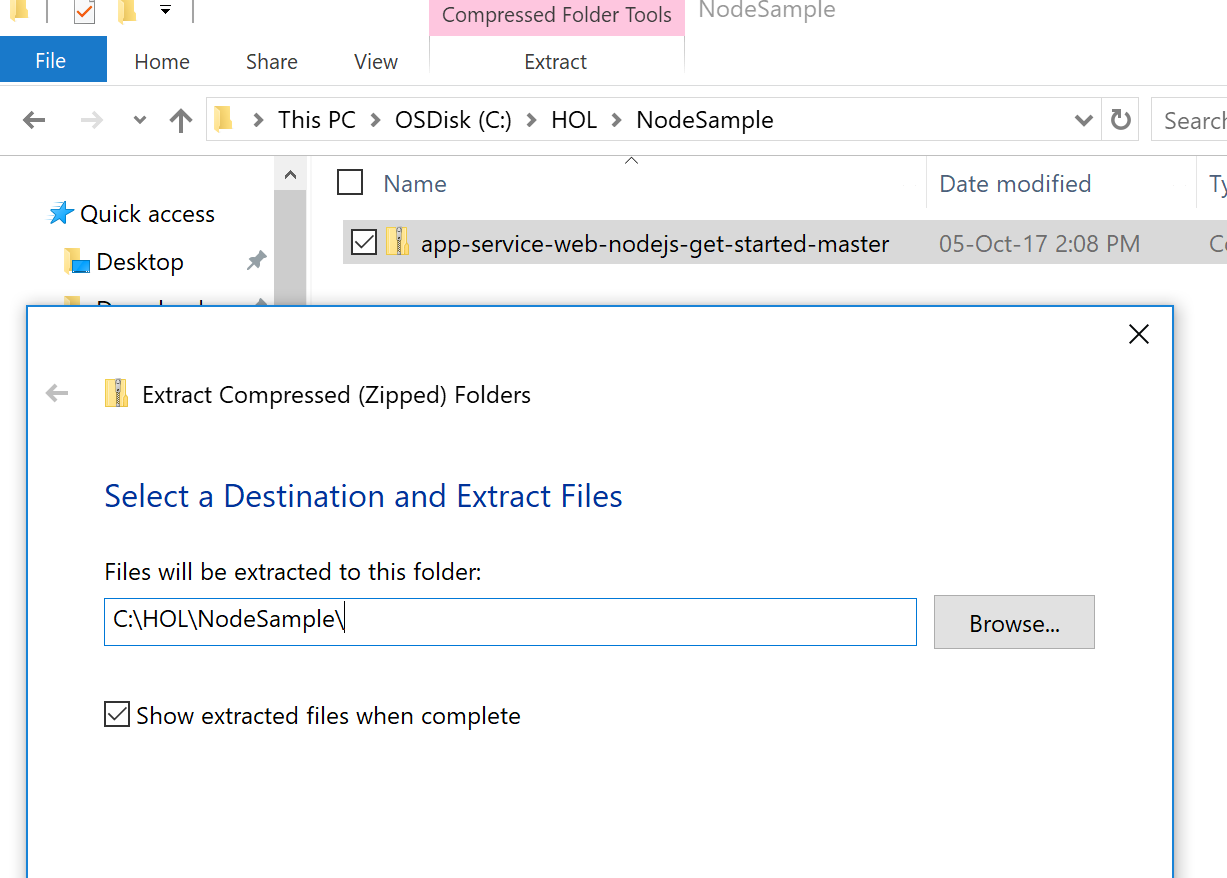
The purpose of the HOL is to help the developers to build a simple NodeJS web application and host the same using Azure Linux App service. As developer, you should be able to build the NodeJS application, testing it locally, Dockerize the application, push the docker image to Docker Hub, and finally create a web application in Azure Linux App Service using the image you published to Docker Hub. Using this approach, you will be able to move any web application running in Linux to Azure App Service.

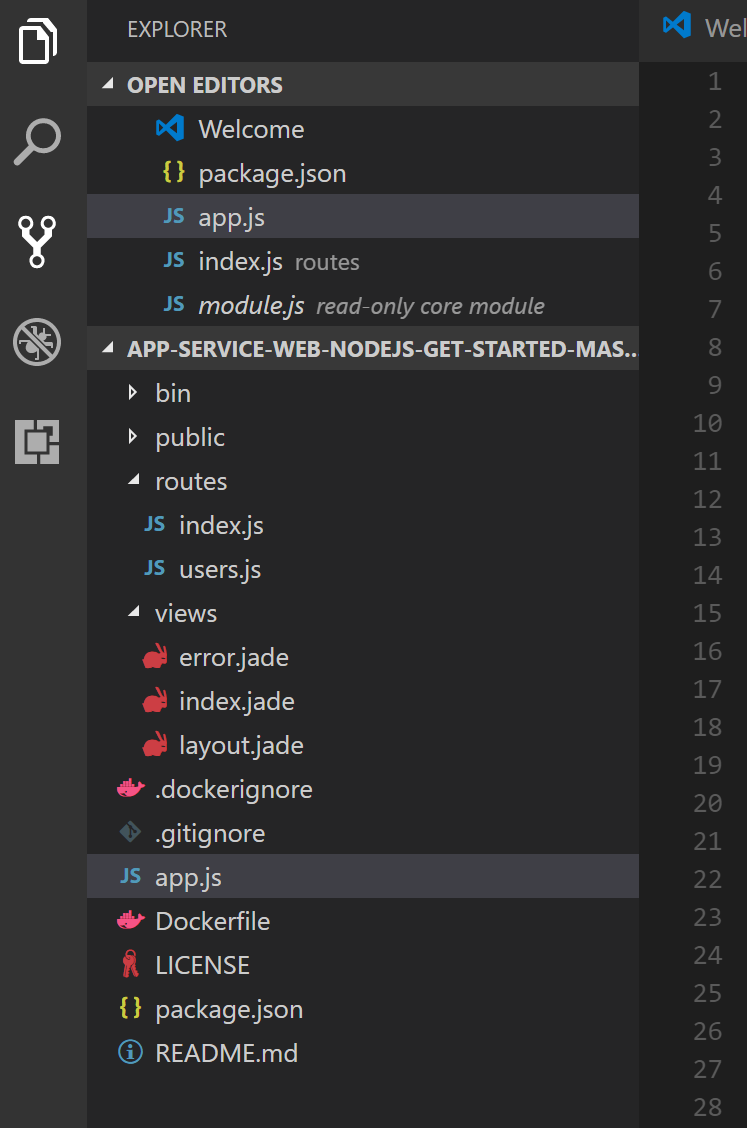
**Prerequisites**

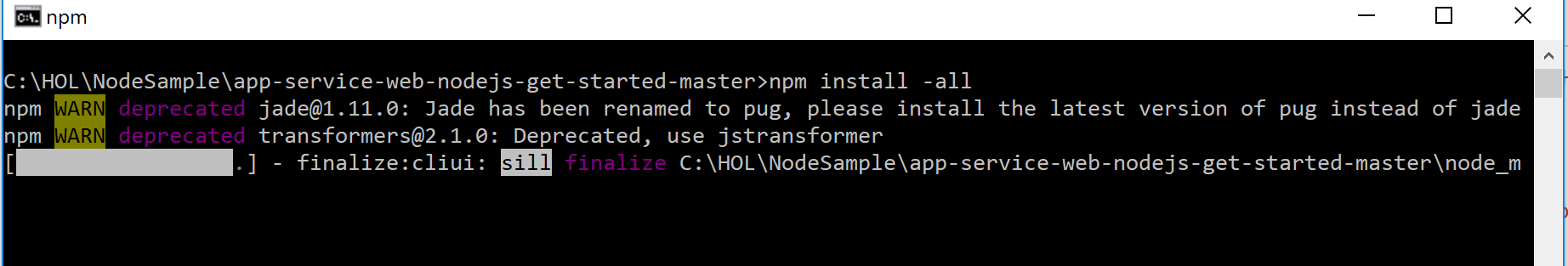
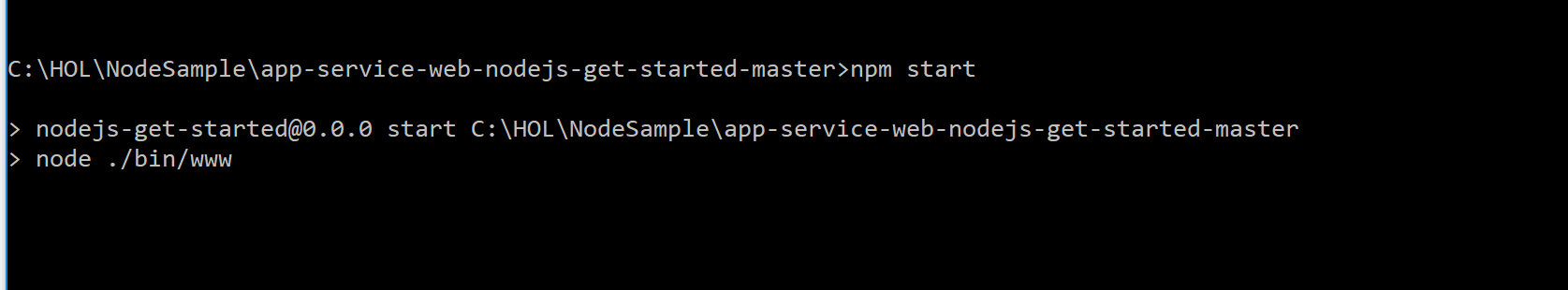
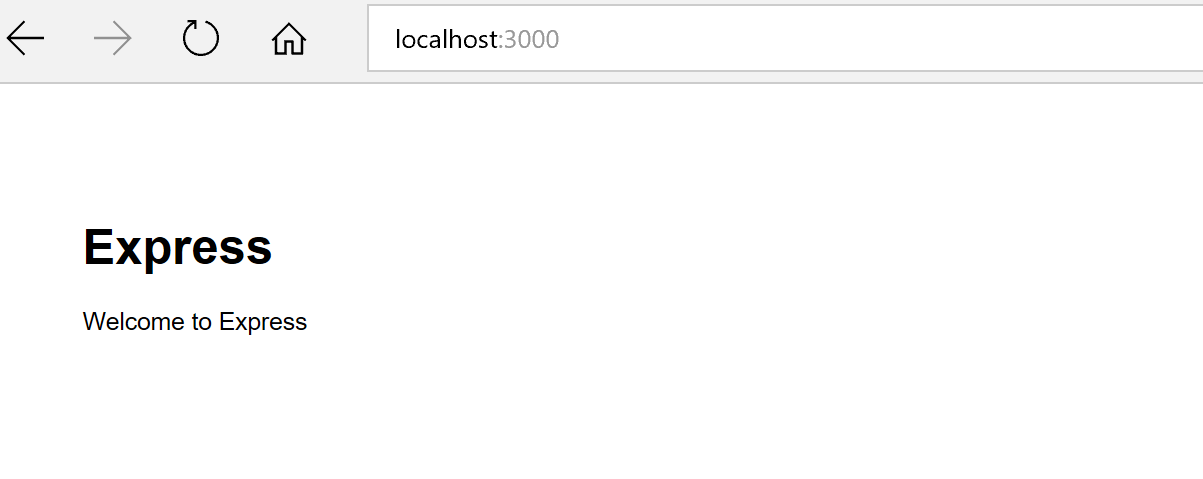
The minimum prerequisites to run this sample are given below, please ensure your developer workstation has the following components pre-installed.

1. Latest Node.js with NPM. Download it from <https://nodejs.org/en/download/>.
2. **[Recommended]** Visual Studio Code for IntelliSense and debugging, download it from <https://code.visualstudio.com/> for free.
3. Docker Community Edition
4. An account created for you in Docker HUB
5. Clone or download code from <https://github.com/maniSbindra/app-service-web-nodejs-get-started>

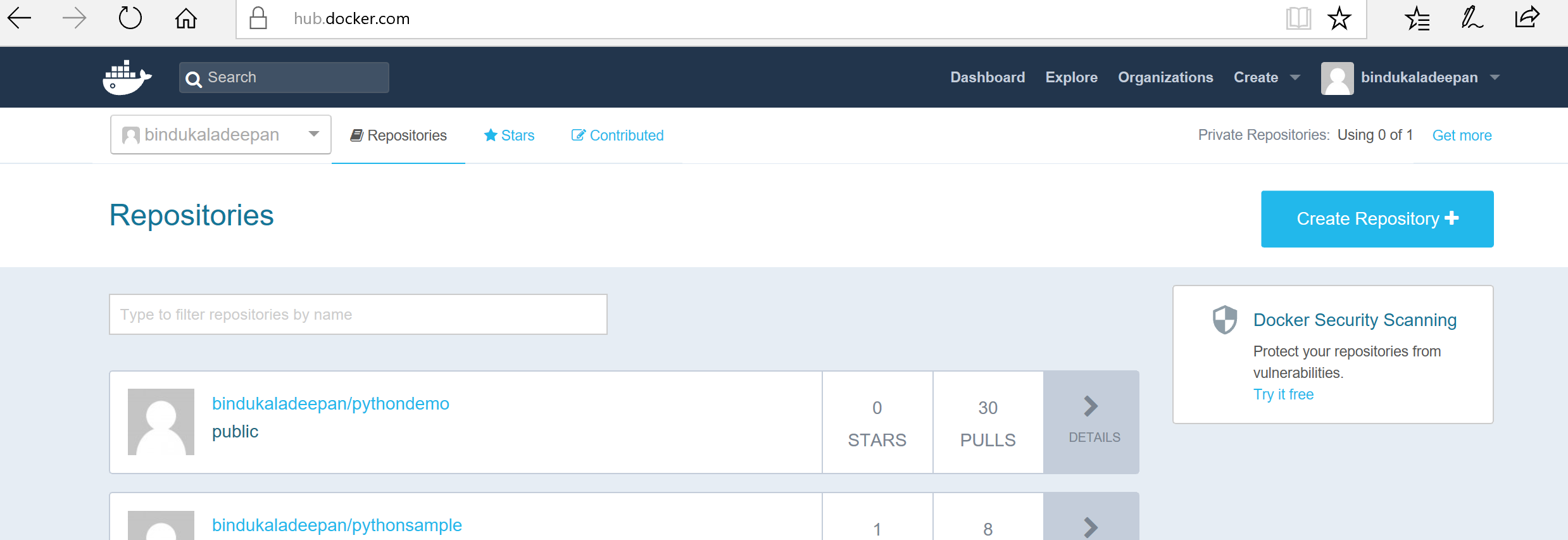
**Step 1 – Building NodeJS application locally**

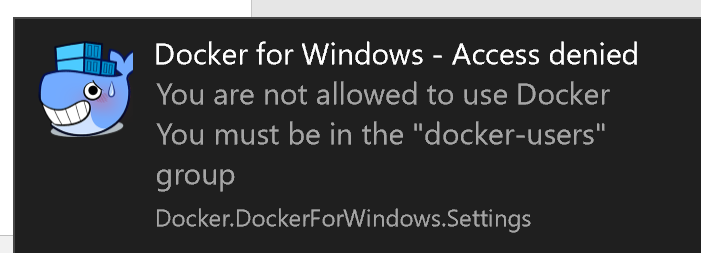
1. The first step in the HOL will be creating a NodeJS application and test the same locally
2. Create a folder in your local computer (**Example**: C:\HOL\NodeSample). 
3. In the browser go to <https://github.com/maniSbindra/app-service-web-nodejs-get-started> and clone or download the application to the folder you have created in the above step (#2) . This is a simple Node.JS and Express based web site
4. Once you download and unzip the file in the folder you have created in the step #2
5. Open visual studio code (or any editor of your choice) and open the folder (in Visual Studio Code -> **File ->Open Folder** and chose the folder where you unzipped the files in last step, Once you open the folder, the file structure should look like below



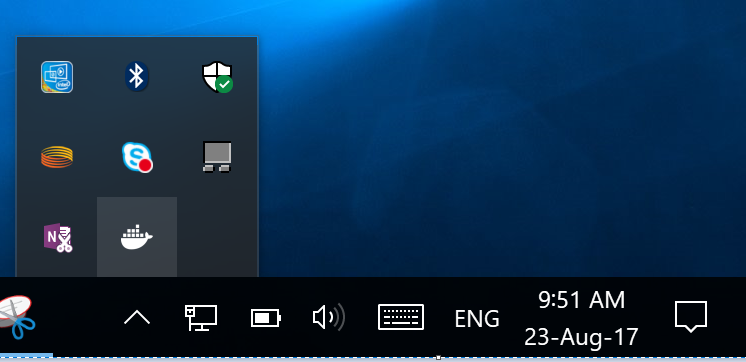
1. Open NodeJS command prompt (desktop application) and cd to your working folder as shown below, and run “**npm install -all**” command, this will install all the required node.js modules for the project 
2. Once all the required node packages got downloaded and installed, in the Node.js command promote (while at your working directory), run “npm start” command, this is command will start Express 
3. Now launch browser and **go to** <http://localhost:3000>, you should find an empty Express page as shown below, this indicates the express node.js application running correctly in locally 
4. Now, go ahead and **close** the Node.JS command prompt to stop the website

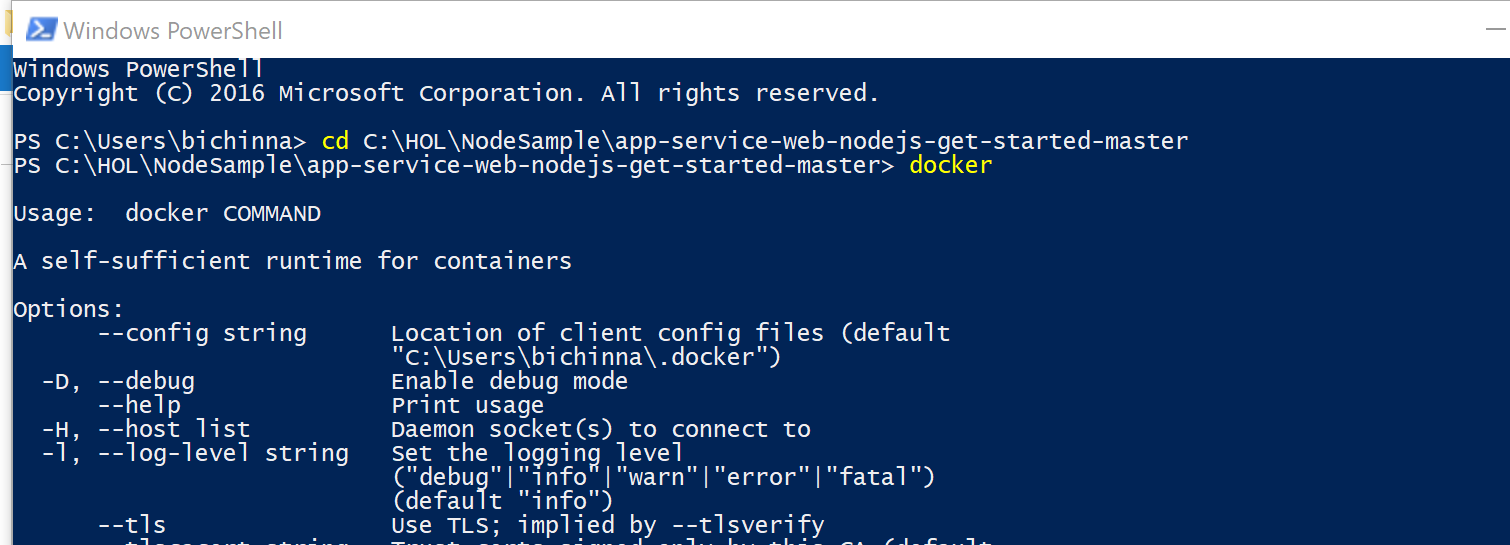
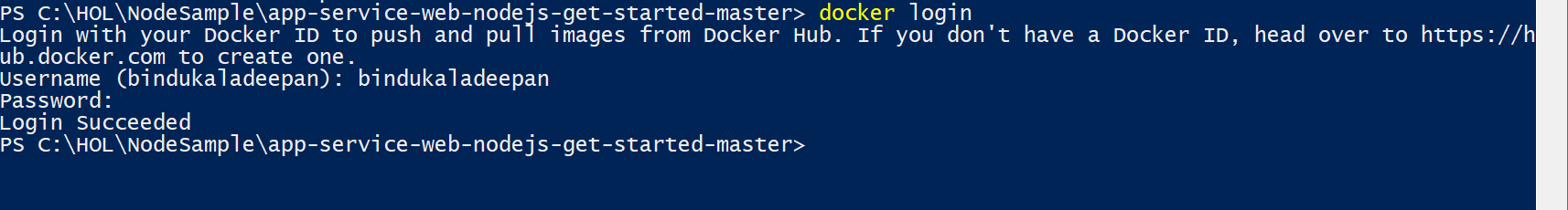
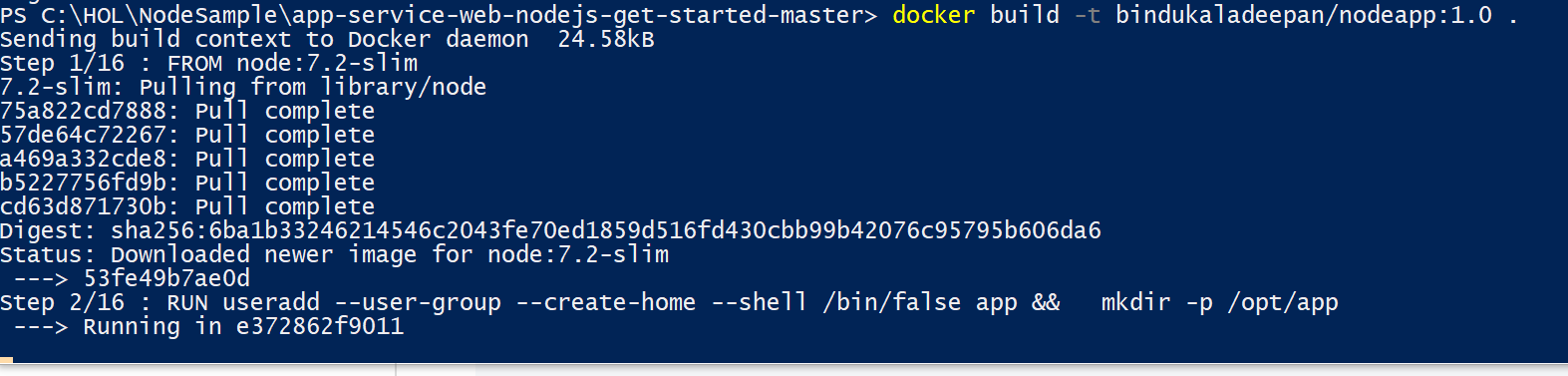
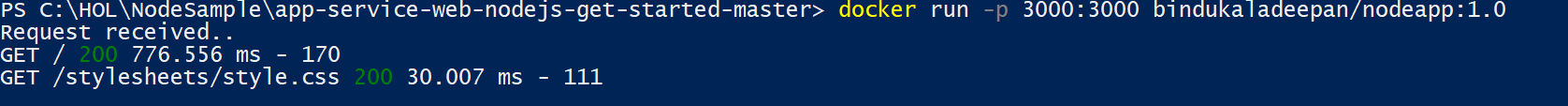
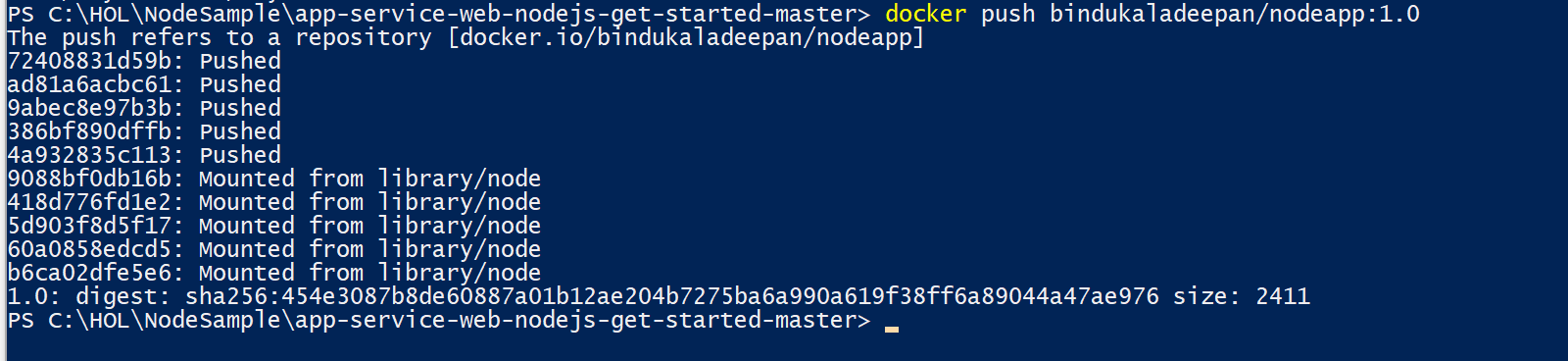
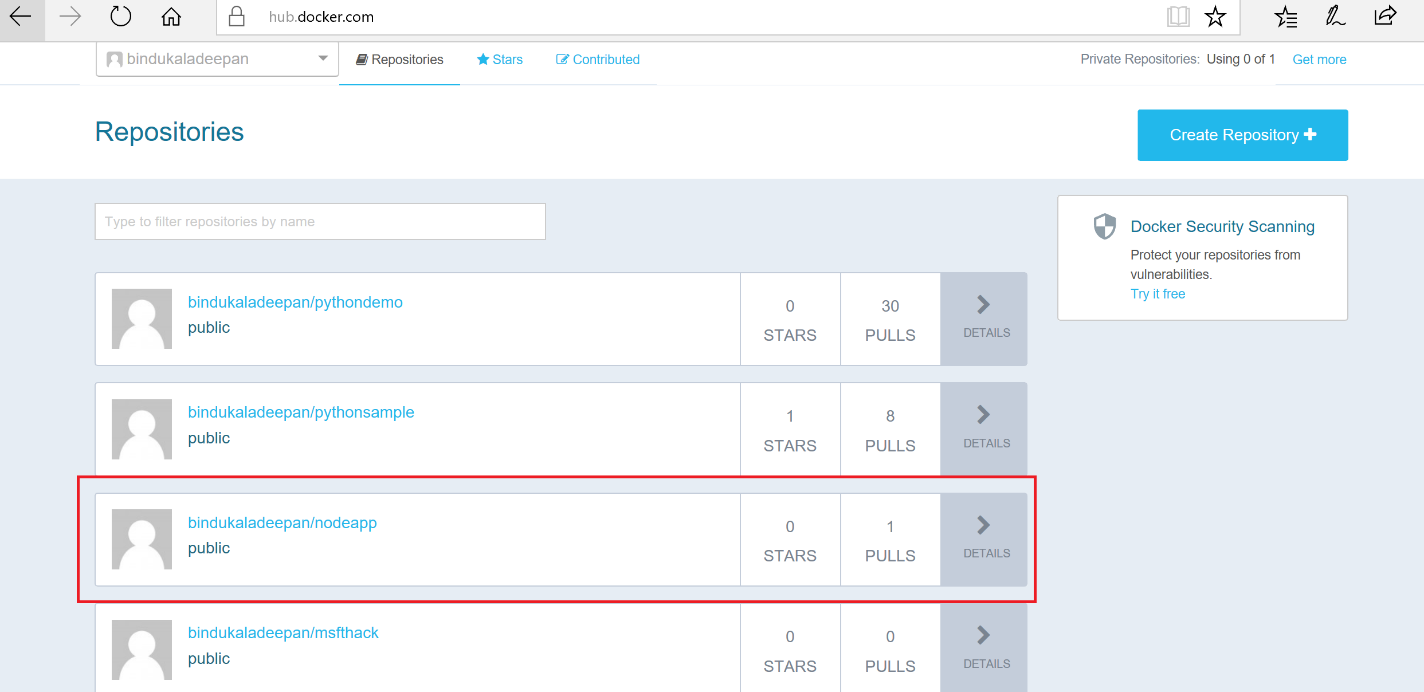
**Step 2 – Dockerize the application and publish the image to Docker Hub**

1. In this step, we will be creating custom Docker image and publish the image to Docker hub. In order to publish the custom images to Docker Hub, therefore at this stage, you should create an account for yourself in [**https://hub.docker.com/**](https://hub.docker.com/) if you don’t have an account with Docker Hub. Go ahead and sign-up for an account in Docker Hub and login with the account you have created. Yould show see a screen ling below when you log-in with your account to <https://hub.docker.com> 
2. **Docker for windows should be already installed and running in your Lab machine**, if yes, jump to step 6 directly and **skip step 2 to 5**. If not, Download and install Docker CE for windows from - <https://www.docker.com/community-edition>
3. Try to launch “Docker for Windows” from your desktop, if you get an error as shown below, go to step 4, otherwise skip step 4

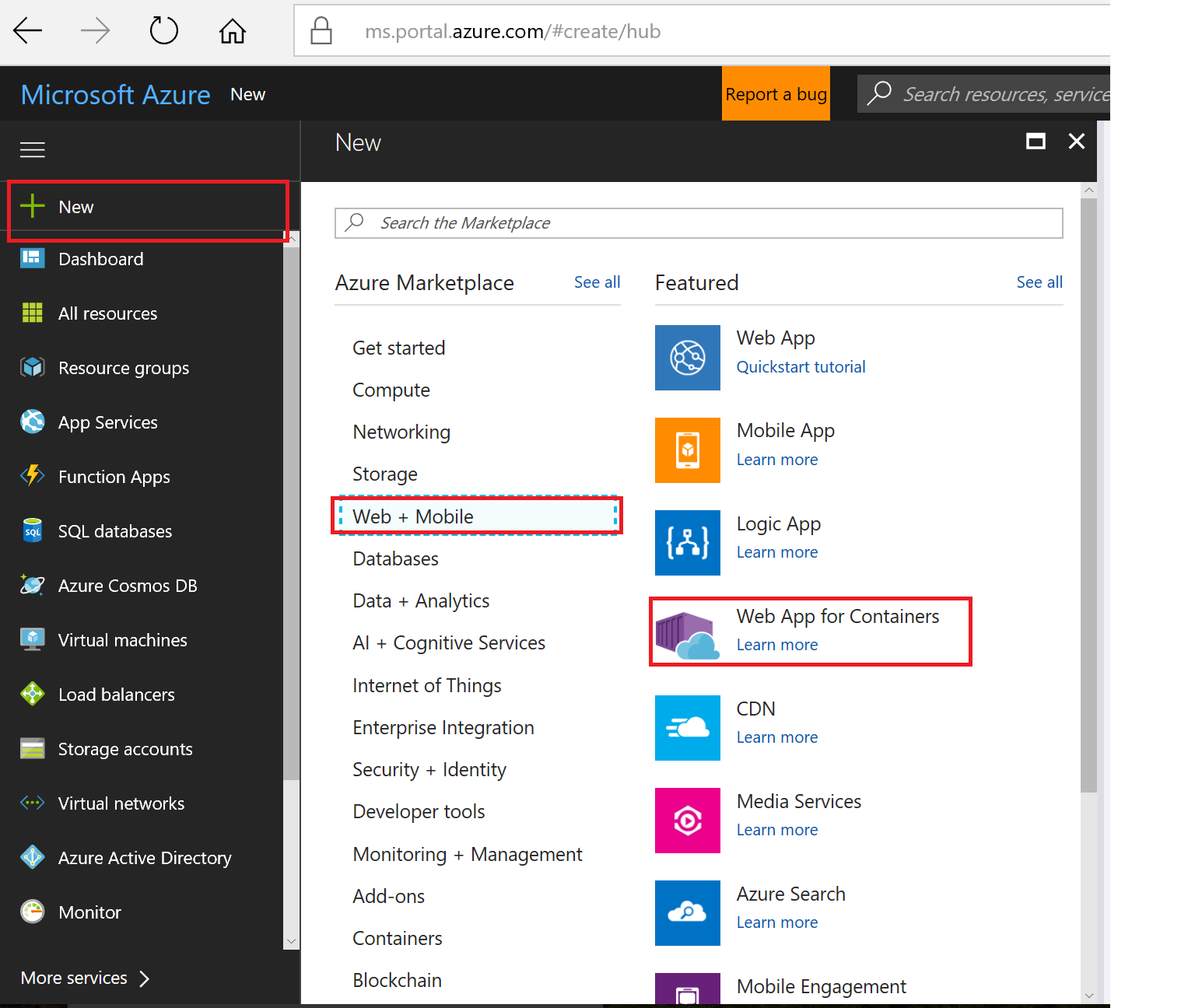
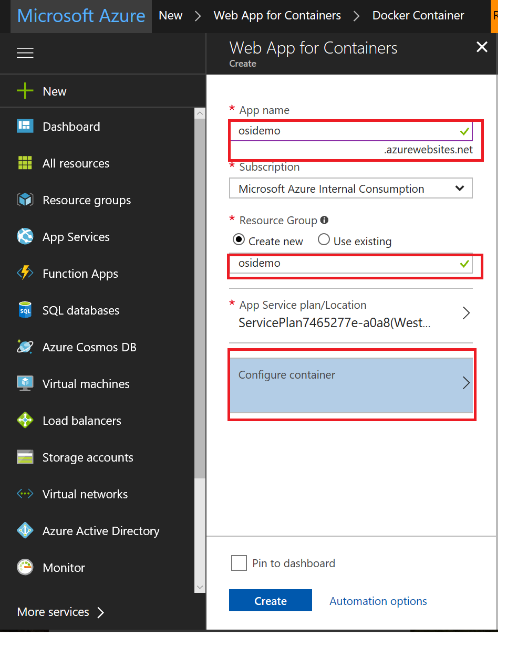
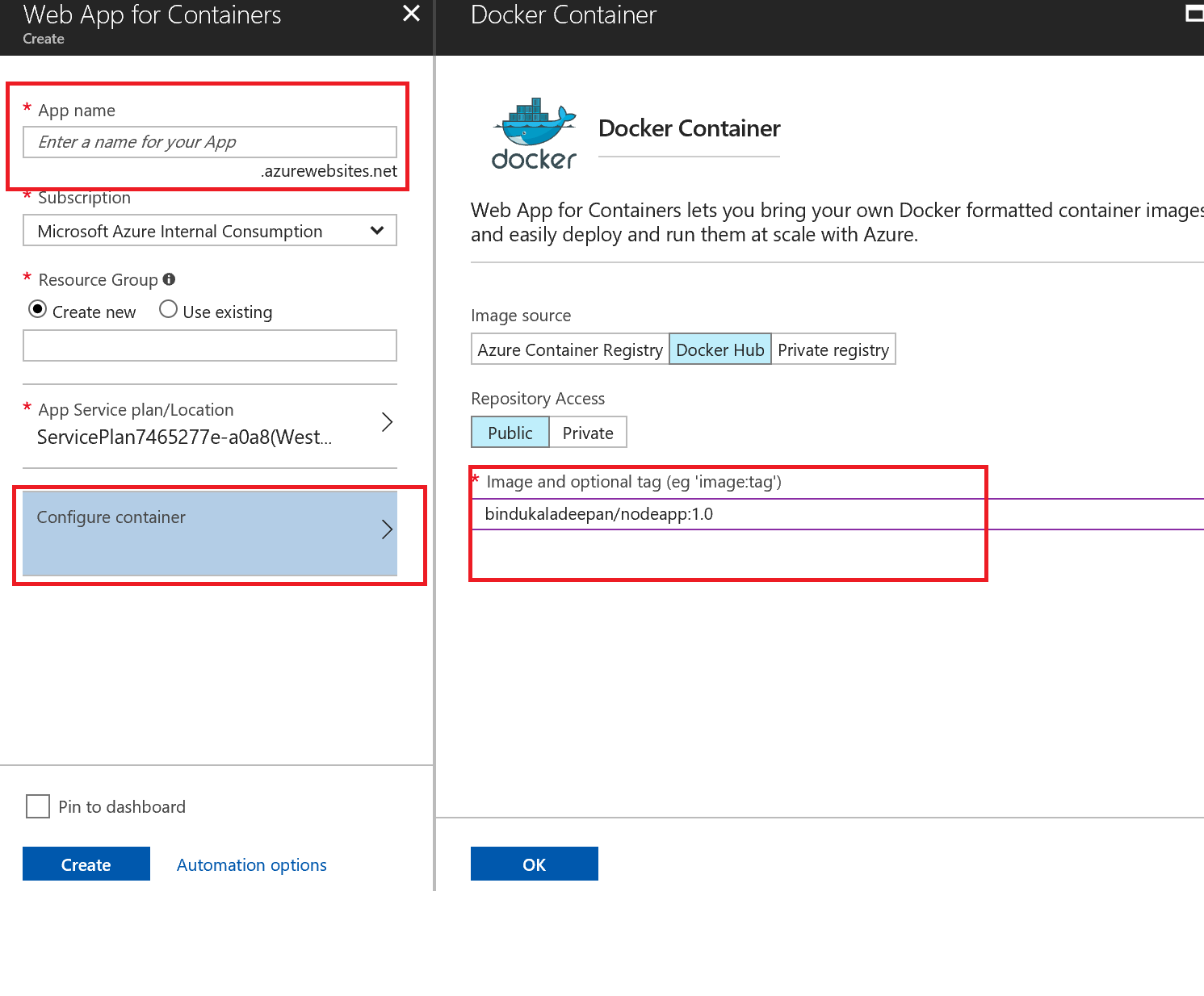
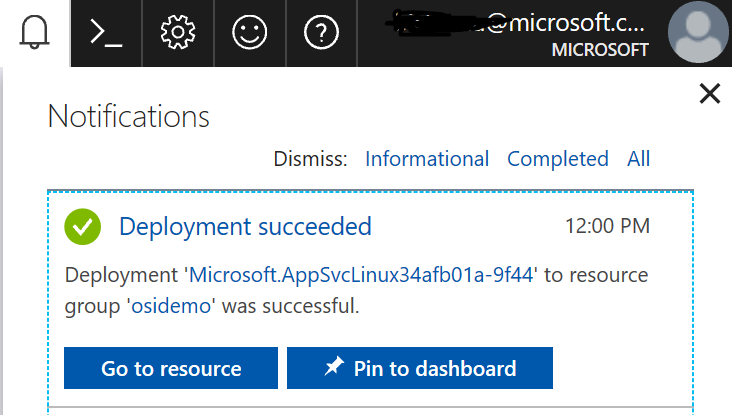
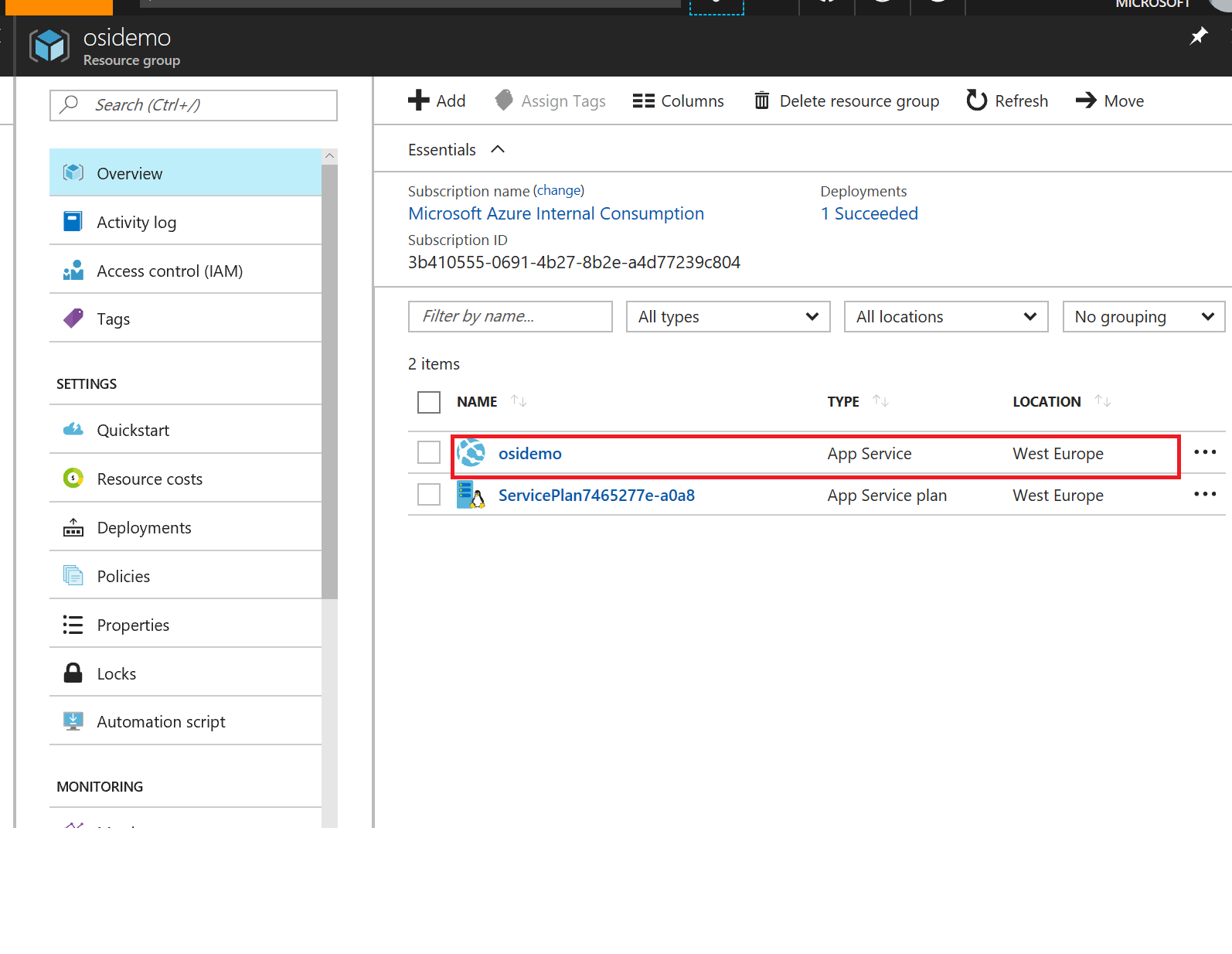
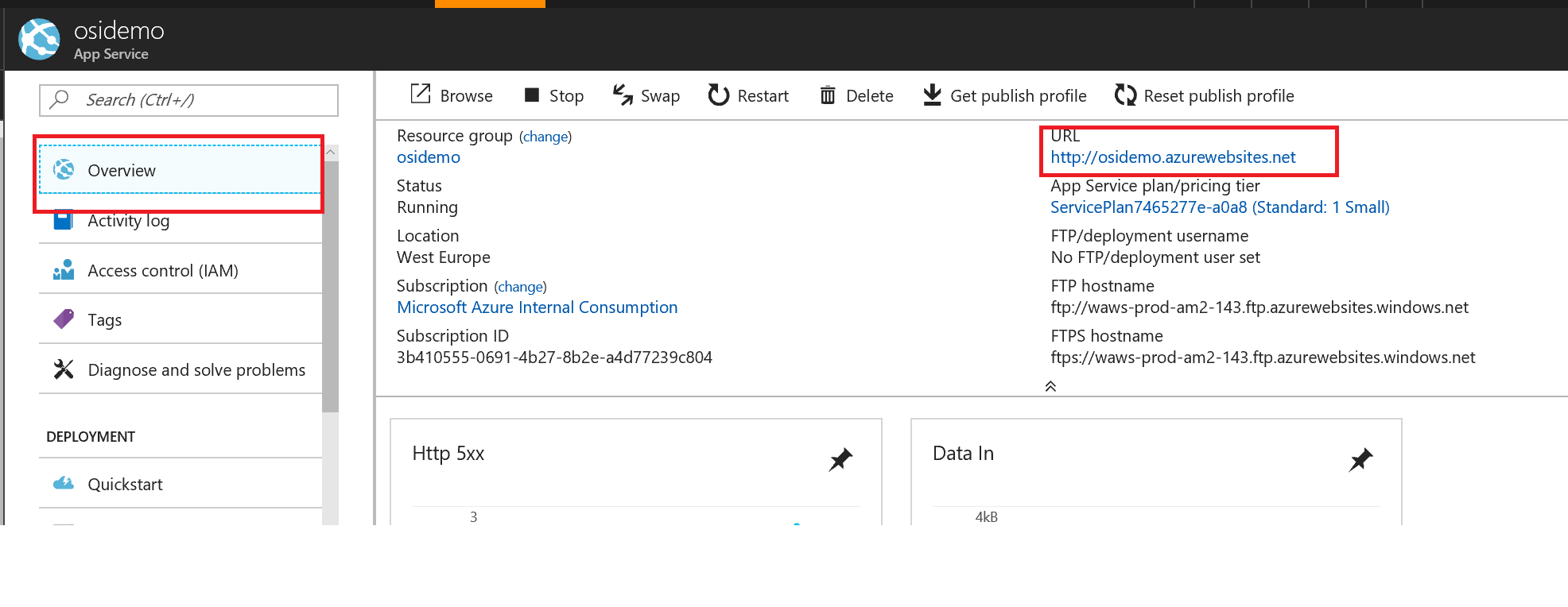


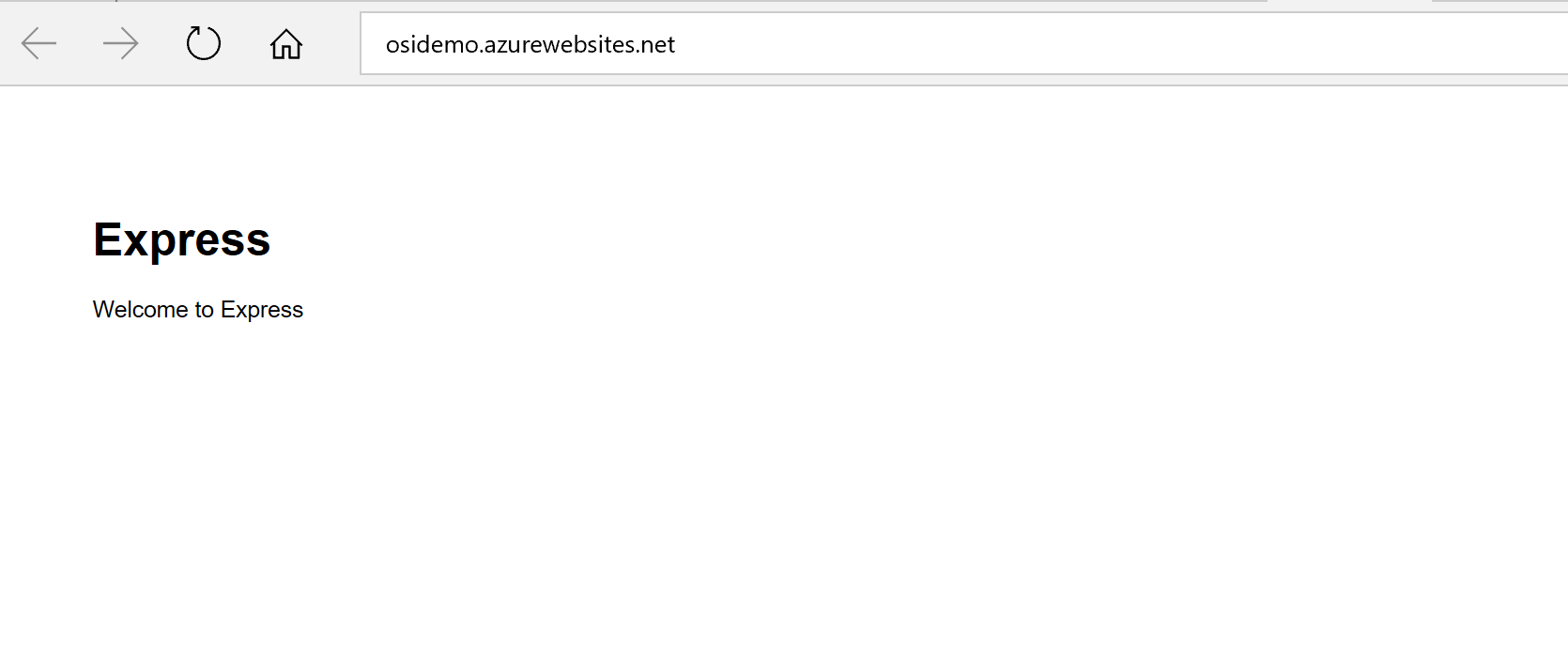
1. Go to control panel and access “Edit local users and group” and find a group called “docker-users”
2. After successful installation, you should have Docker up and running, verify the status by checking in status bar



1. Open Windows PowerShell and navigate to the working folder as shown below, and type commend “**docker**”, you should find list of docker options, if so, it indicates that Docker is up and running correctly 
2. In the PowerShell, issue “**docker login**” command as shown below, and when prompted enter your Docker Hub account user name and password. You should see a “login Succeeded” message when you get yourself logged in 
3. Run the docker command “**docker build -t <<your docker hub userid>>/nodeapp:1.0 .**” This command will build a docker image. Note the “**.**” at the end of the command. Once you run the command, Docker will download all the pre-requisites and create an Image for the application 
4. Once you are able to successfully build the Docker Image, run command “**docker run -p 3000:3000 <<Name of the docker image your have created in the last step >>”.** In this step, we are running the Node.JS application within Docker Container (Linux based Operating system). -p in the command is to map our local machine’s port to Docker’s port. So that we can access the site running within Docker from local machine
5. Now try accessing [**http://localhost:3000**](http://localhost:3000) in browser, you should see the application up and running 
6. Run “**docker push <<Name of your docker image >>**” this will upload the image we have created in the above step to Docker Hub, you should see something like shown in the below image when Docker uploading the image to Docker Hub 
7. Go to <https://hub.docker.com> and login with your docker user name and password, now you should see the image that we have published in the last step

**Step 3 – Create Web App using Azure Linux App service using image available in Docker Hub**

1. Go to Azure portal <https://portal.azure.com> click “**+New -> Web+Mobile -> Web App for Containers**” as shown below 
2. In the next screen, provide valid values for **App Name** and **Resource Group**, and you may create a new resource group, you may change the App service plan or leave the default plan automatically created, and clcik “Configure Container” as shown below 
3. In the container configuration page, Select “Docker Hub” in Image Source and “Public” in Repository access, and enter the name of the image you published to your docker hub repository, and click “**OK**”, and clock “**Create**”. Ensure you give the full name of the docker image including any tag name you may have given. “**image name:tag**” Example **bindukaladeepan/nodeapp:1.0** 
4. After clicking “Create” wait until the resources gets deployed, you should see a notification message as shown below in Azure portal
5. Once the resource got created, go the resource group and find and click the web application we just crated as shown below, go to “**Overview**” and obtain “**URL**”
6. Obtain URL as shown below and access the public facing site. Now you should see the site you just deployed. 



**Conclusion**

We have completed creating a simple NodeJS & Express based web application, and created custom docker container image out of the application to run in Linux OS, published the images to Docker HUB, and used the image available in Docker HUB and created a web application using “Web App for Containers” . Using this approach you can bring any custom docker image and make it running in Azure