**Click2Cloud DockM**

**User Manual**

Created Date: 02/08/2017

Created By: Click2Cloud

**Location - India**

s

Table of Contents

[1. Click2Cloud DocM Application 3](#_Toc492485685)

[1.1 Active Endpoints: 4](#_Toc492485686)

[1.2 Endpoint Actions: 4](#_Toc492485687)

[1.2.1 Dashboard: 4](#_Toc492485688)

[1.2.2 App Templates: 5](#_Toc492485689)

[1.2.3 Stacks 6](#_Toc492485690)

[1.2.4 Services 8](#_Toc492485691)

[1.2.5 Containers 9](#_Toc492485692)

[1.2.6 App To Container 13](#_Toc492485693)

[1.2.7 Docker Images 13](#_Toc492485694)

[1.2.8 Networks 14](#_Toc492485695)

[1.2.9 Volumes 16](#_Toc492485696)

[1.2.10 Secrets 17](#_Toc492485697)

[1.2.11 Swarm 18](#_Toc492485698)

[1.3 DockM Setting: 19](#_Toc492485699)

[1.3.1 User Management: 19](#_Toc492485700)

[1.3.2 Endpoints 22](#_Toc492485701)

[1.3.3 Registries 23](#_Toc492485702)

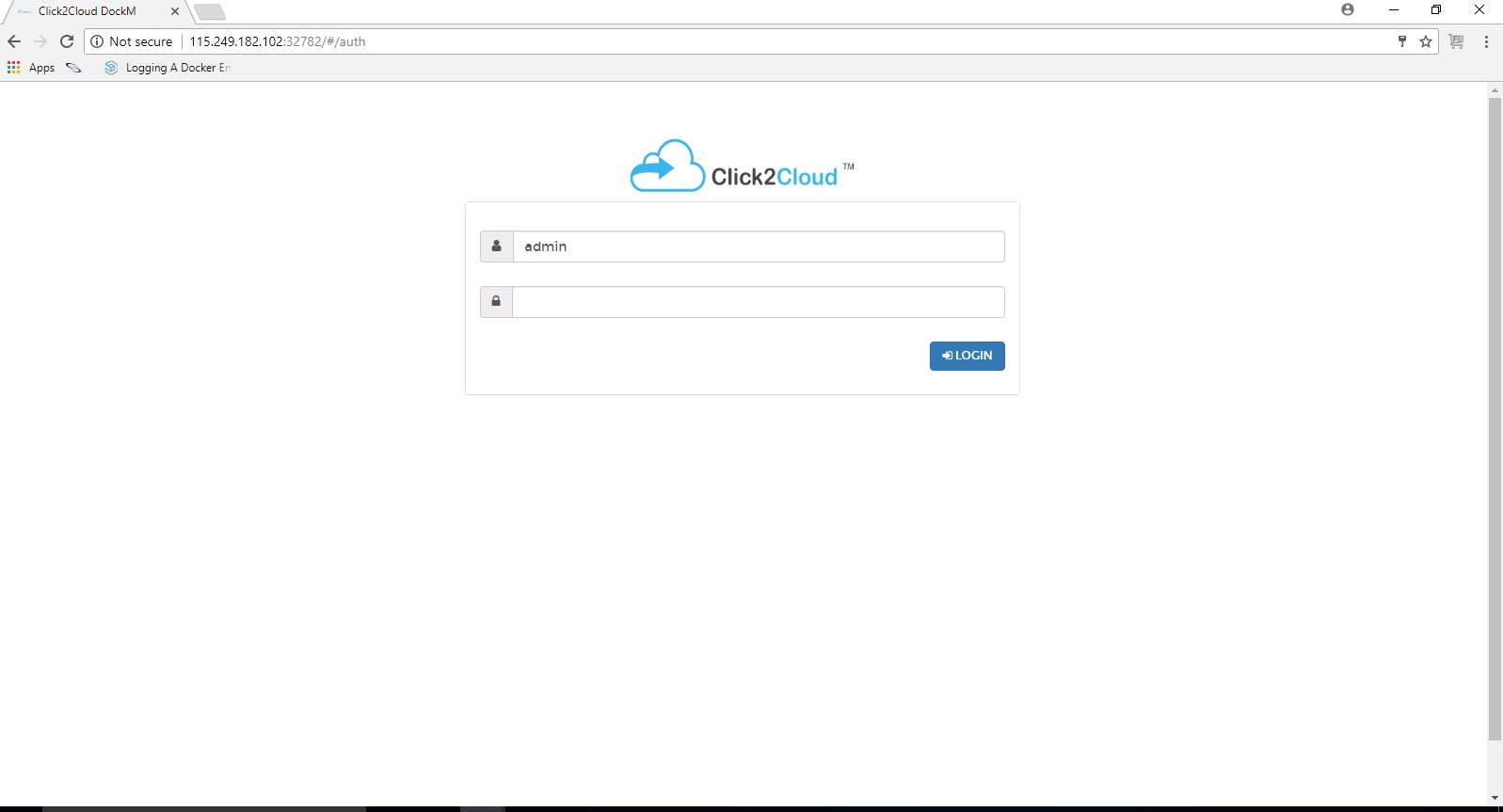
[1.3.4 Settings 24](#_Toc492485703)

# 1. Click2Cloud DocM Application

**Click2Cloud DocM** is lightweight management **UI** which allows you to **easily** manage your different Docker environments like Docker hosts or Swarm clusters.

It allows you to manage your Docker containers, images, volumes, networks! It is compatible with the *Docker* engine and with *Docker Swarm mode*.

Enter your credentials to log into the **Click2Cloud** **DockM** application.



## **1.1 Active Endpoints:**

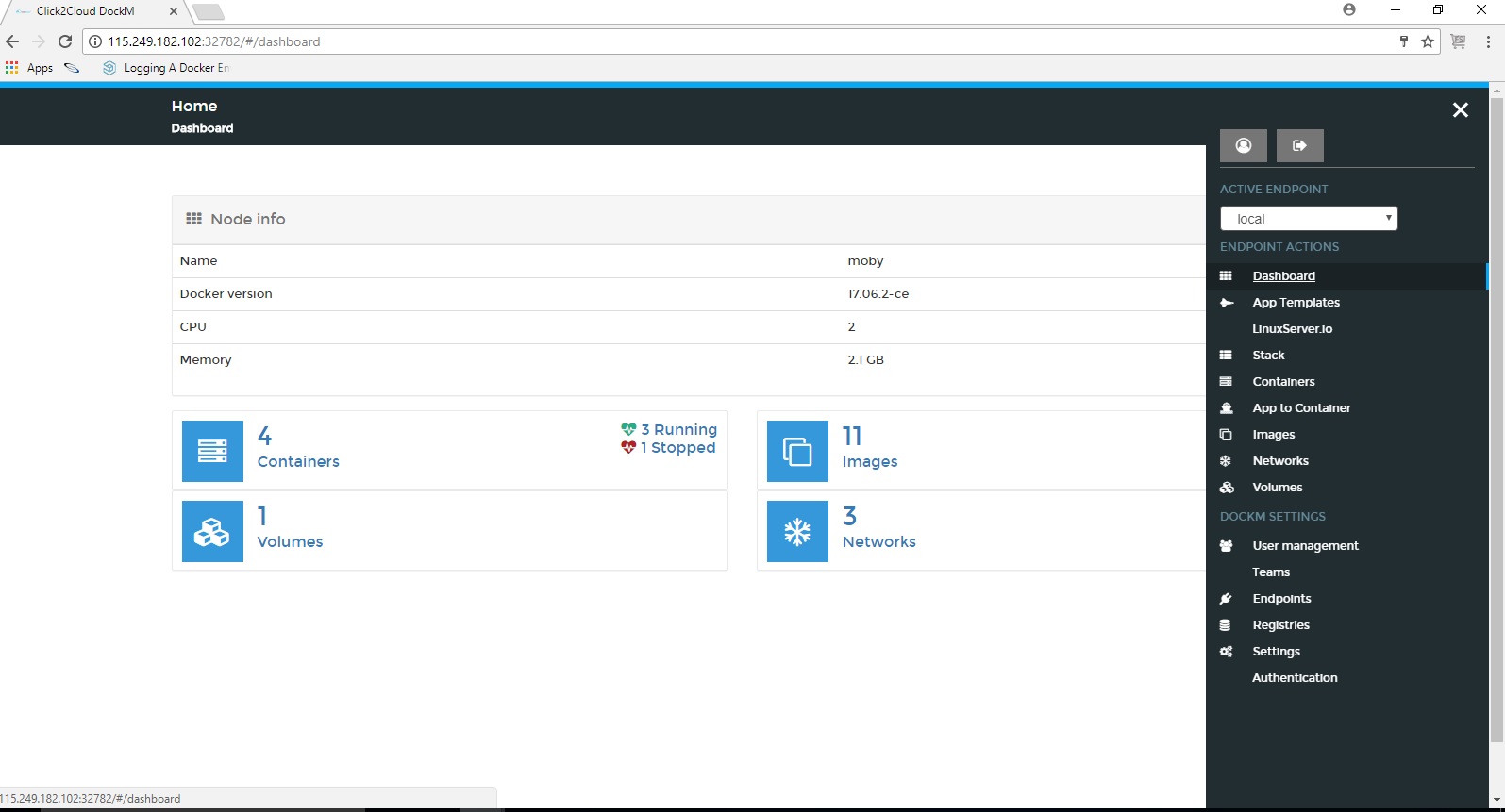
Lists the endpoints for which the user is authorized by the Administrator.

## **1.2 Endpoint Actions:**

This section provides the detailed overview of Docker and allows you to manage stacks, services, containers, images, networks, volumes and swarm.

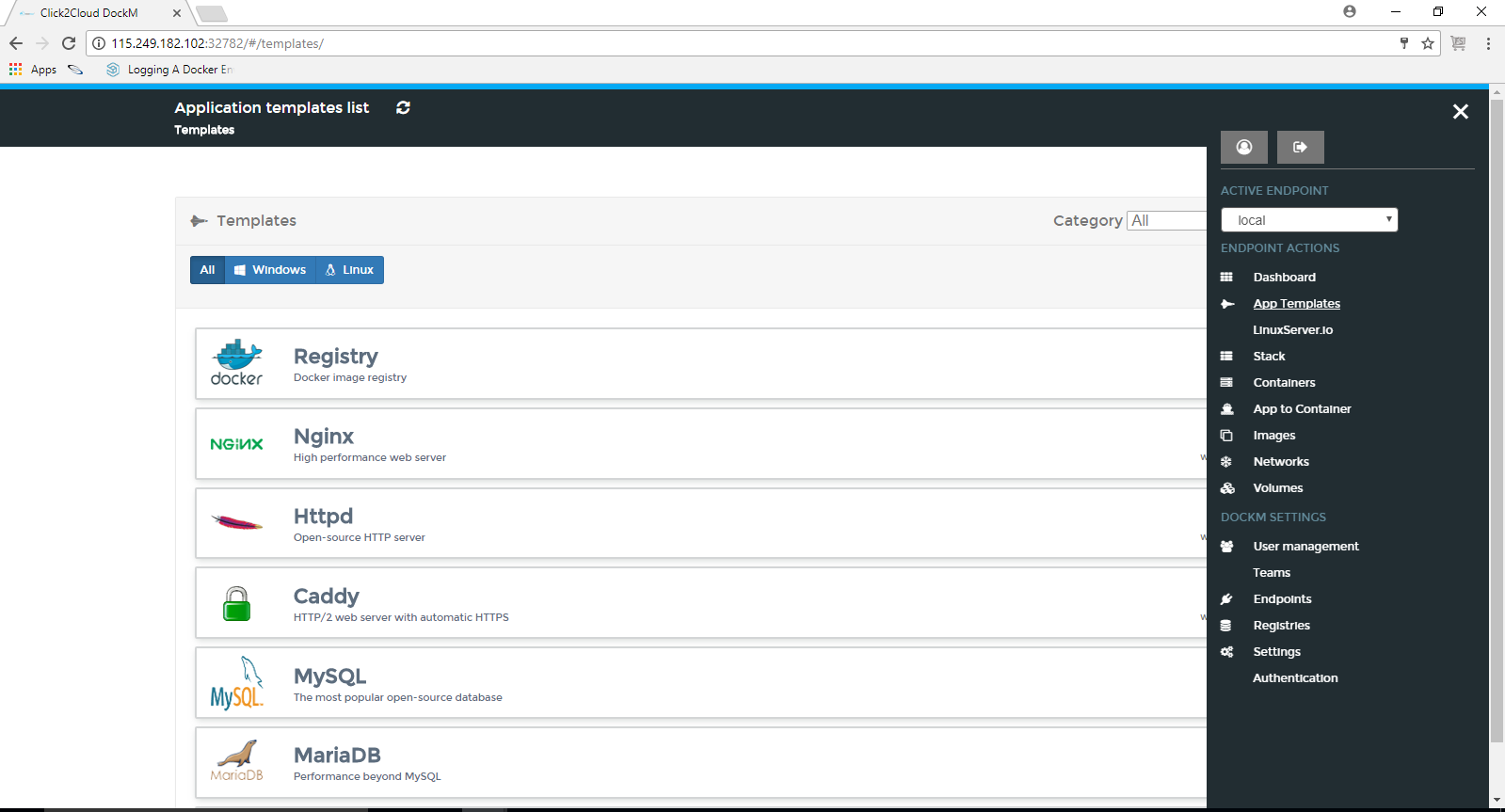
### **1.2.1 Dashboard:**

The dashboard home screen displays the running and stopped containers, number of downloaded Docker images, number of volumes and networks. You don’t have memorize Docker commands. Everything can be done from the Dashboard itself.



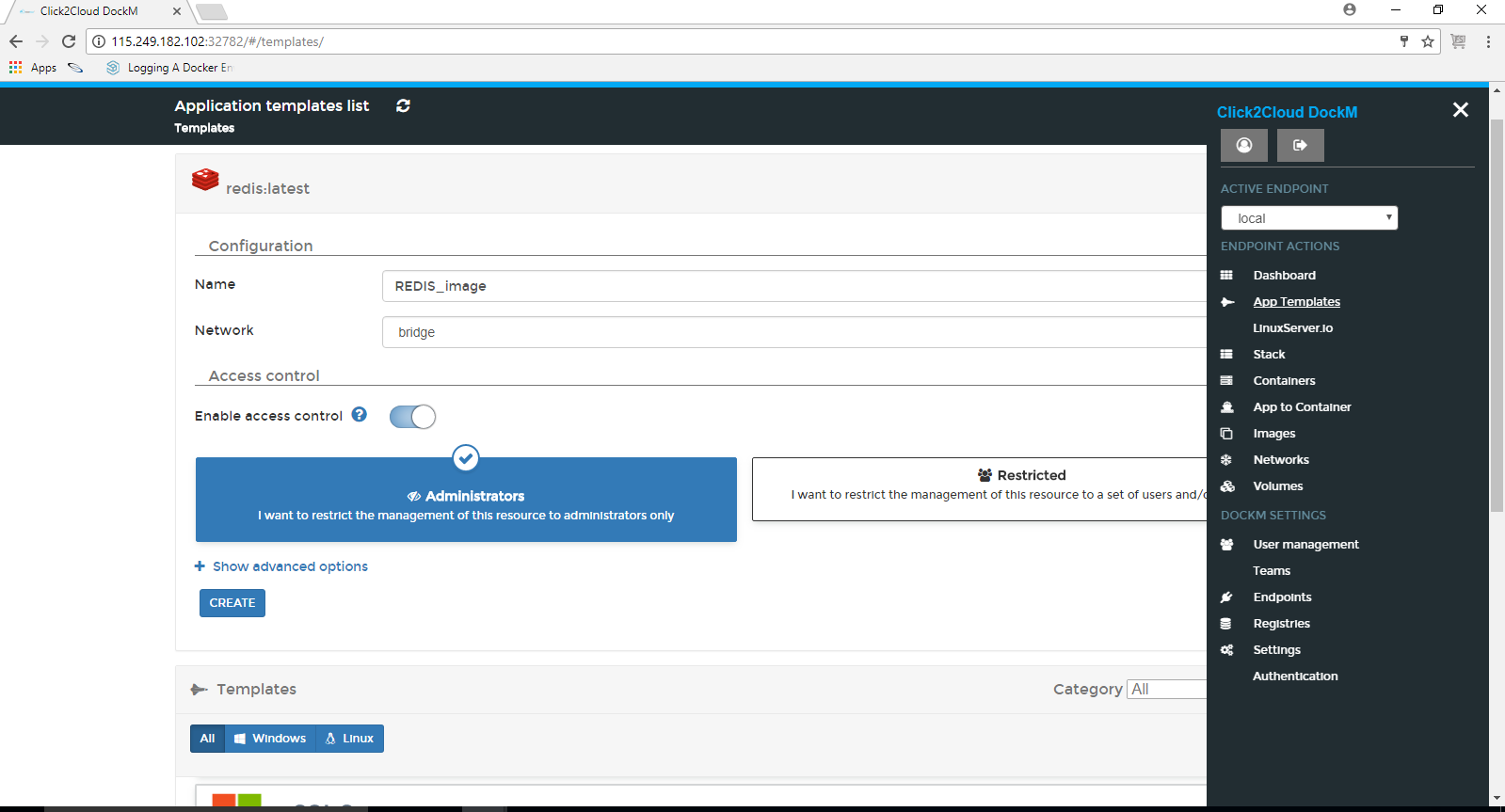
### **1.2.2 App Templates:**

The App Templates screen shows some ready-made available templates such as Docker image registry, Nginx, Httpd, MySql, WordPress etc. You can create your own custom templates by specifying the path of JSON file of GitHub repository, local system from Settings tab.



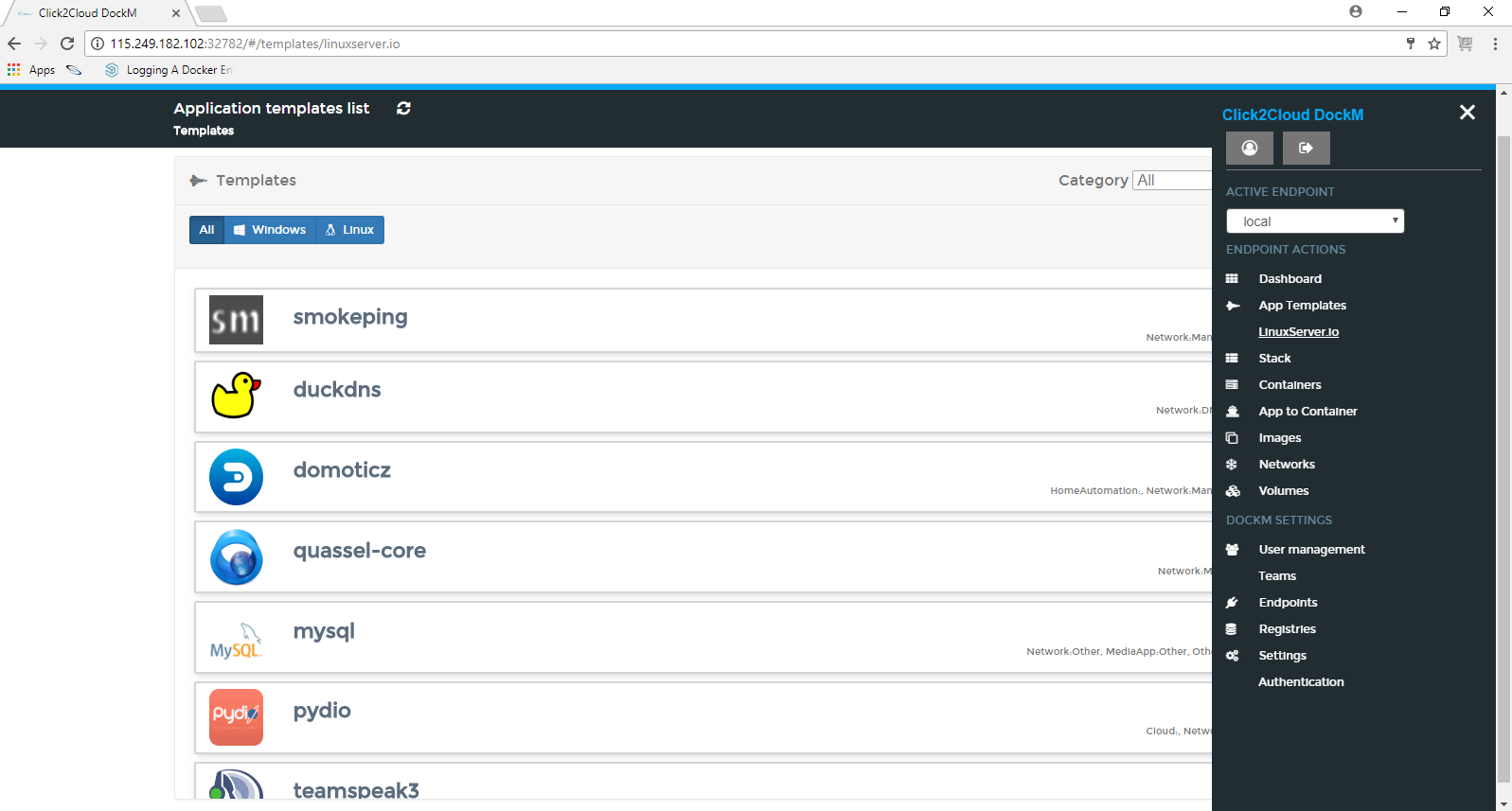
To deploy a Container, just click on the respective template. Let us launch **Redis** Container. To do so, click on the **Redis** template. Enter the Container name, select network type (ex. bridge mode). click on **Show advanced options** and set port number.

Finally, Click **Create**.



#### **1.2.2.1 LinuxServer.io**

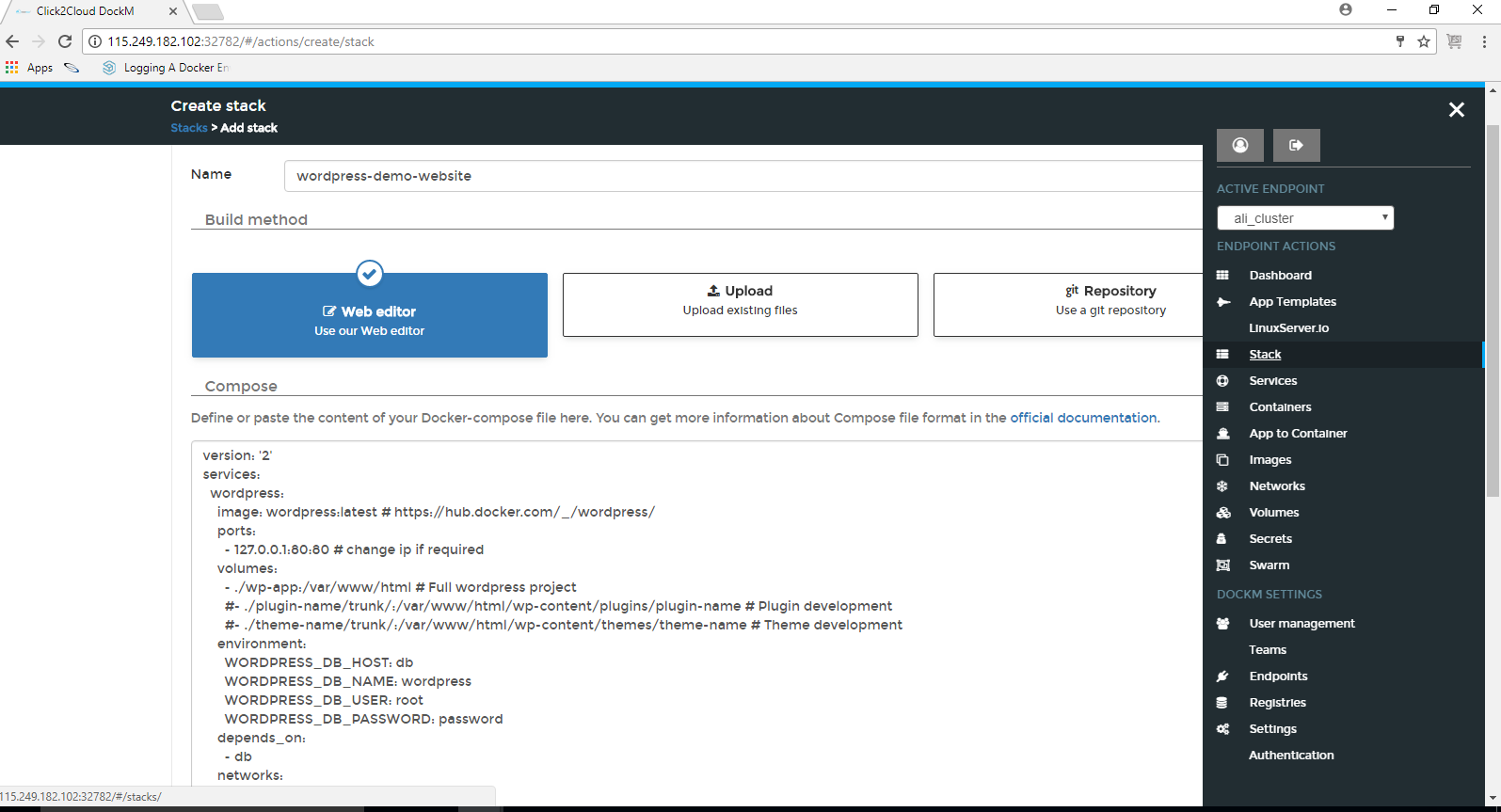
App Templates contains another tab LinuxServer.io which shows images related to Linux environment.



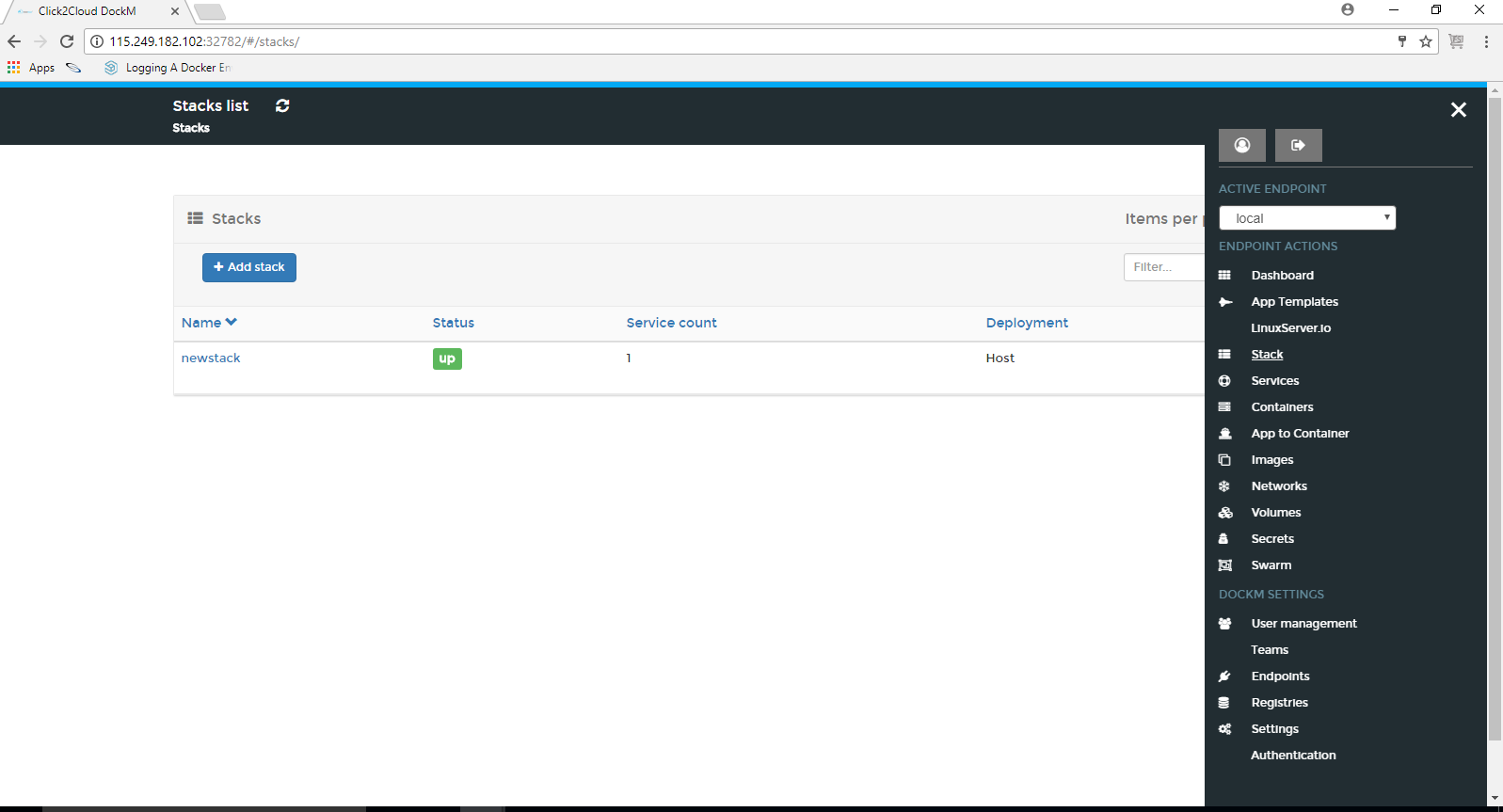
### **1.2.3 Stacks**

Multi-container Docker applications can be run on a single Docker engine by using stack files. Stack file defines environment variables, deployment tags, the number of containers, and related environment-specific configuration.

Copy the contents of docker-compose.yml file in the provided web-editor text box and create the stack. We can also upload or specify the path of GitHub repository for your docker-compose.yml file.

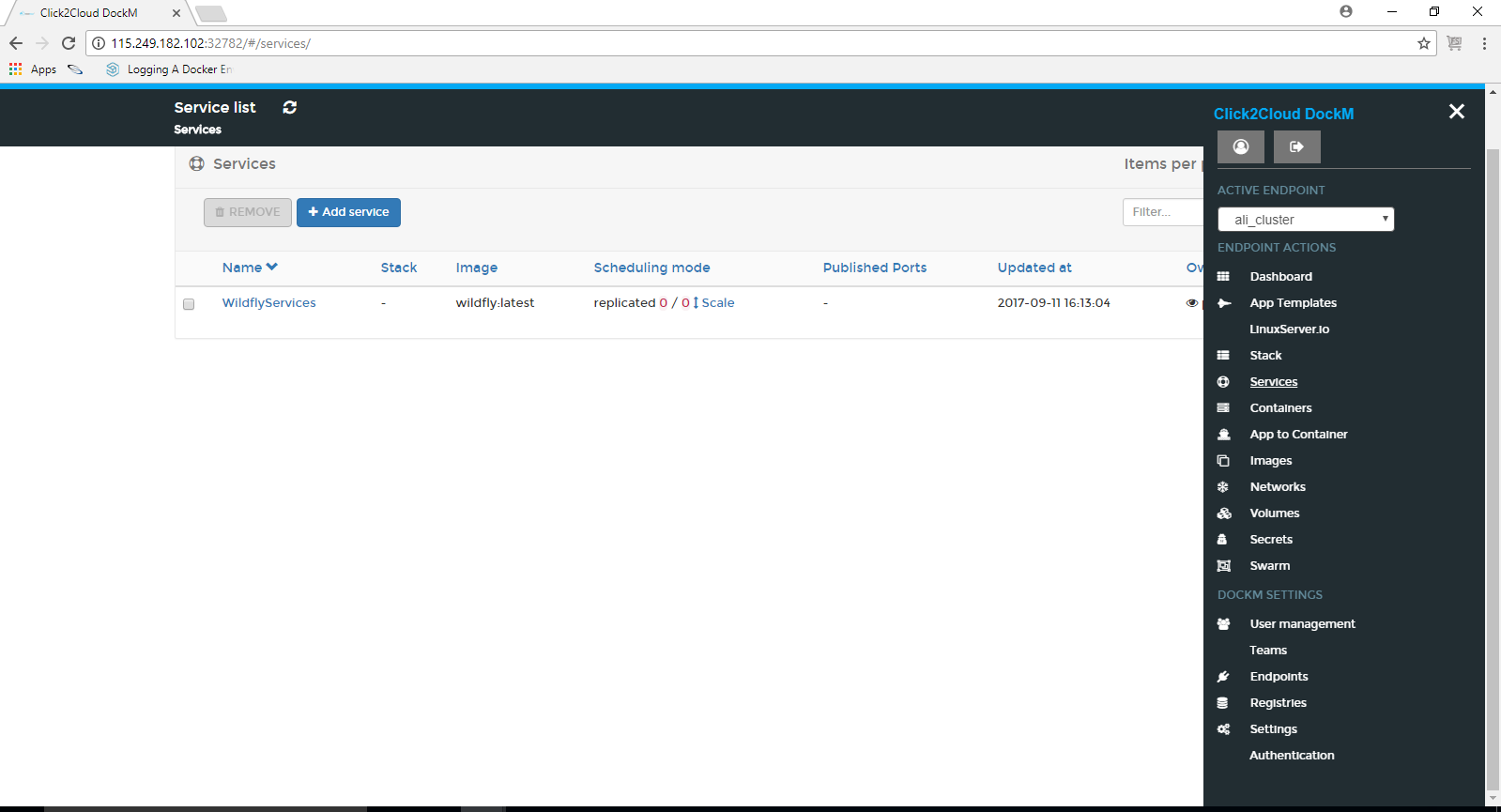


The stacks list displays the created stacks, status, service count and deployment hosts.

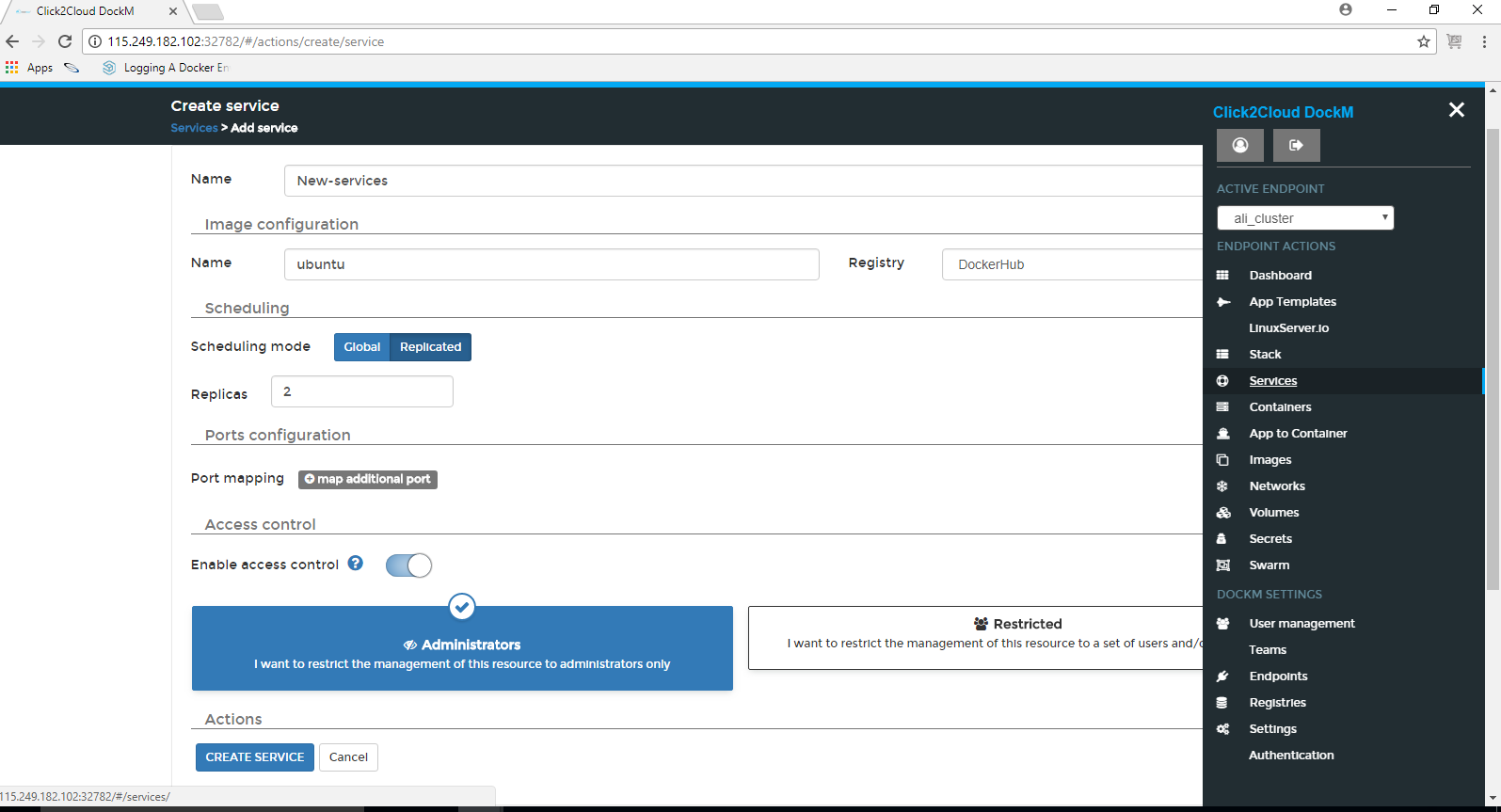


### **1.2.4 Services**

Services allows the Docker engine to know about the state of the services that we want up at all times. Now that the engine knows about the desired state, it can take immediate measures to correct things when the actual state differs. If a container dies unexpectedly at any time, the engine can bring another one up automatically. If an entire node goes down taking a few containers with it, the engine detects it and can bring these containers back up on a healthy node.

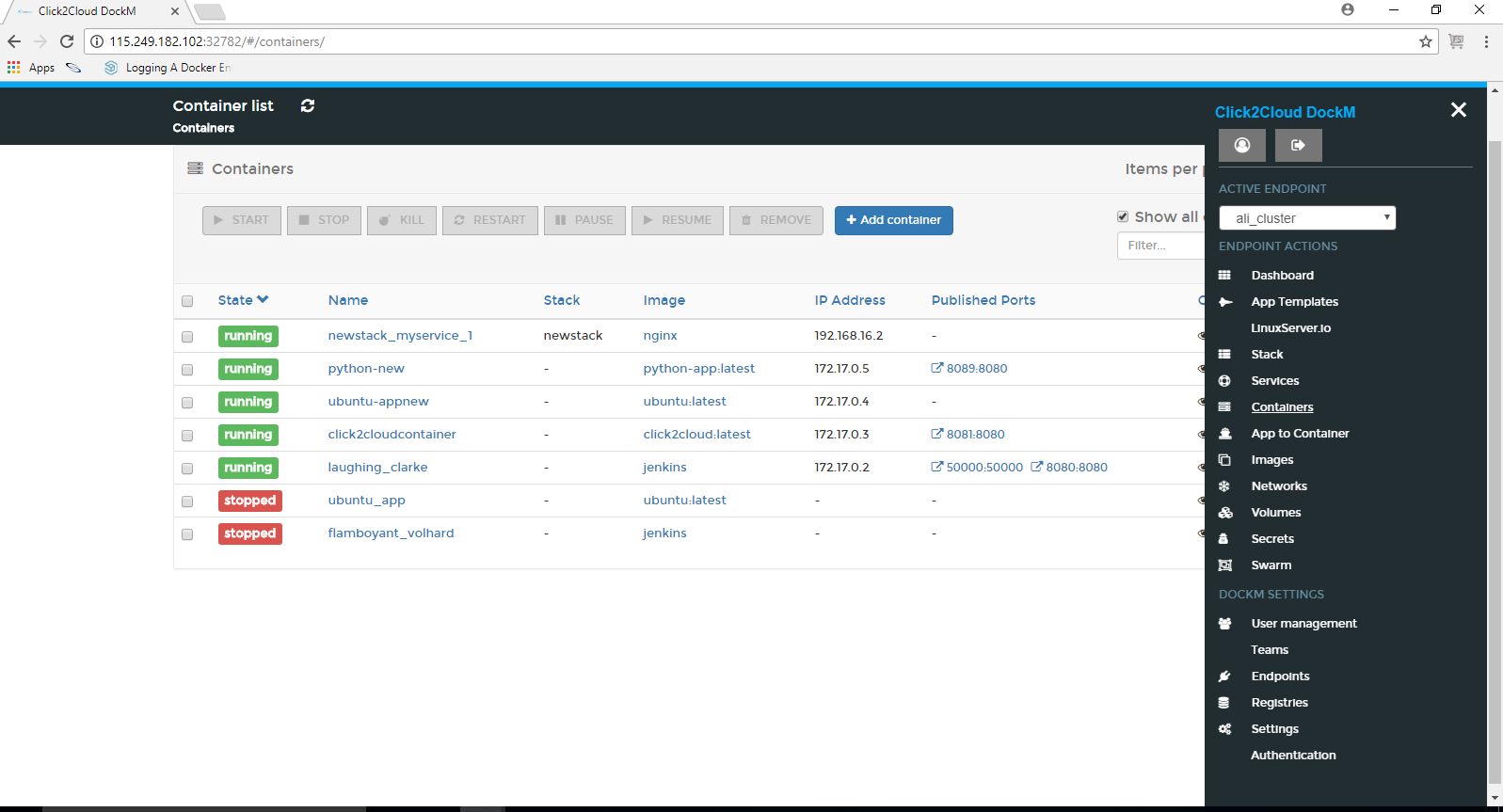


The service created by using service option can be schedule as Global, Replicated. We can also create the entry points, working directory, networks, mapped volumes etc.



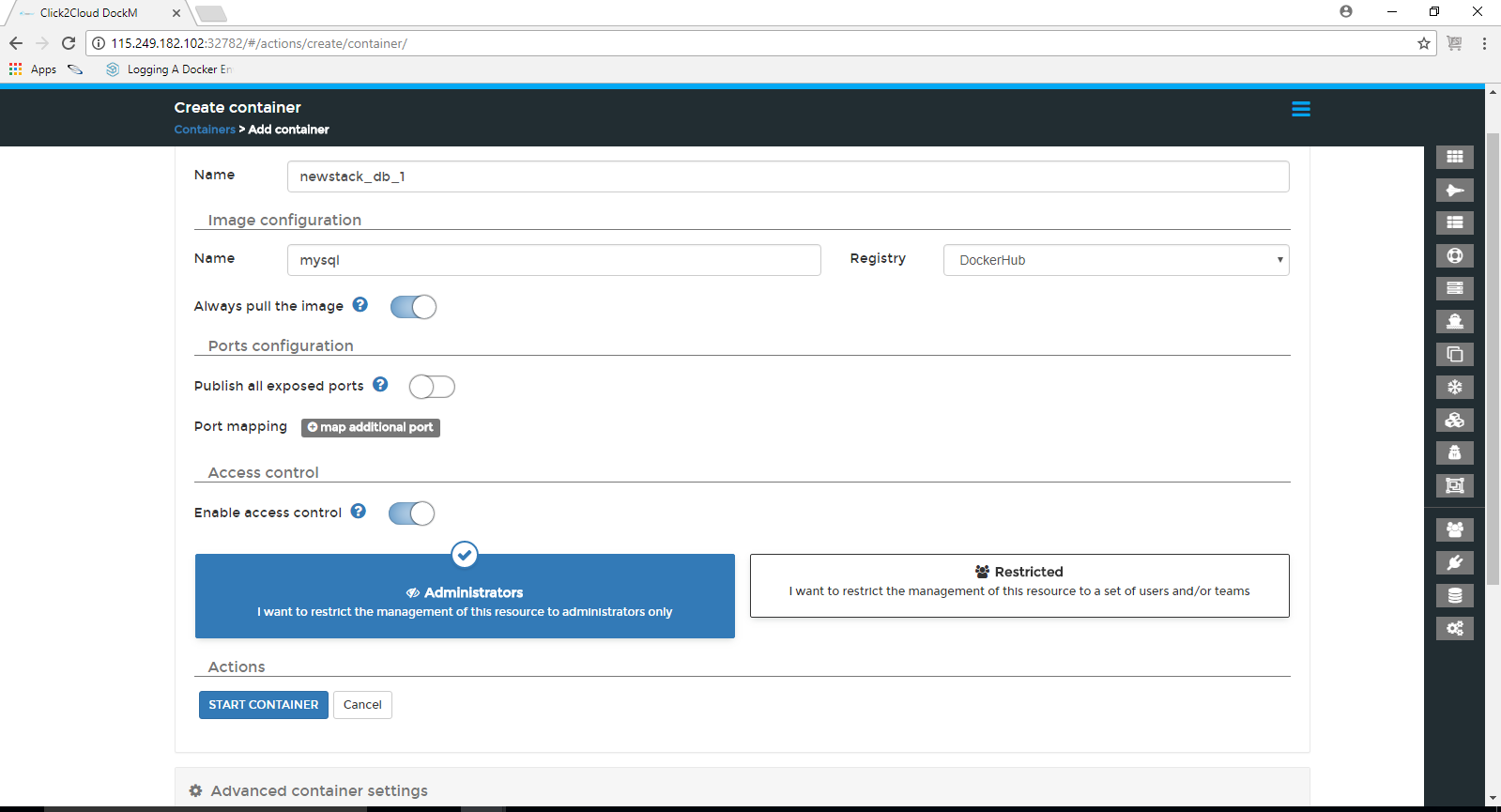
### **1.2.5 Containers**

The **Containers** page that shows the list of running and Stopped containers.

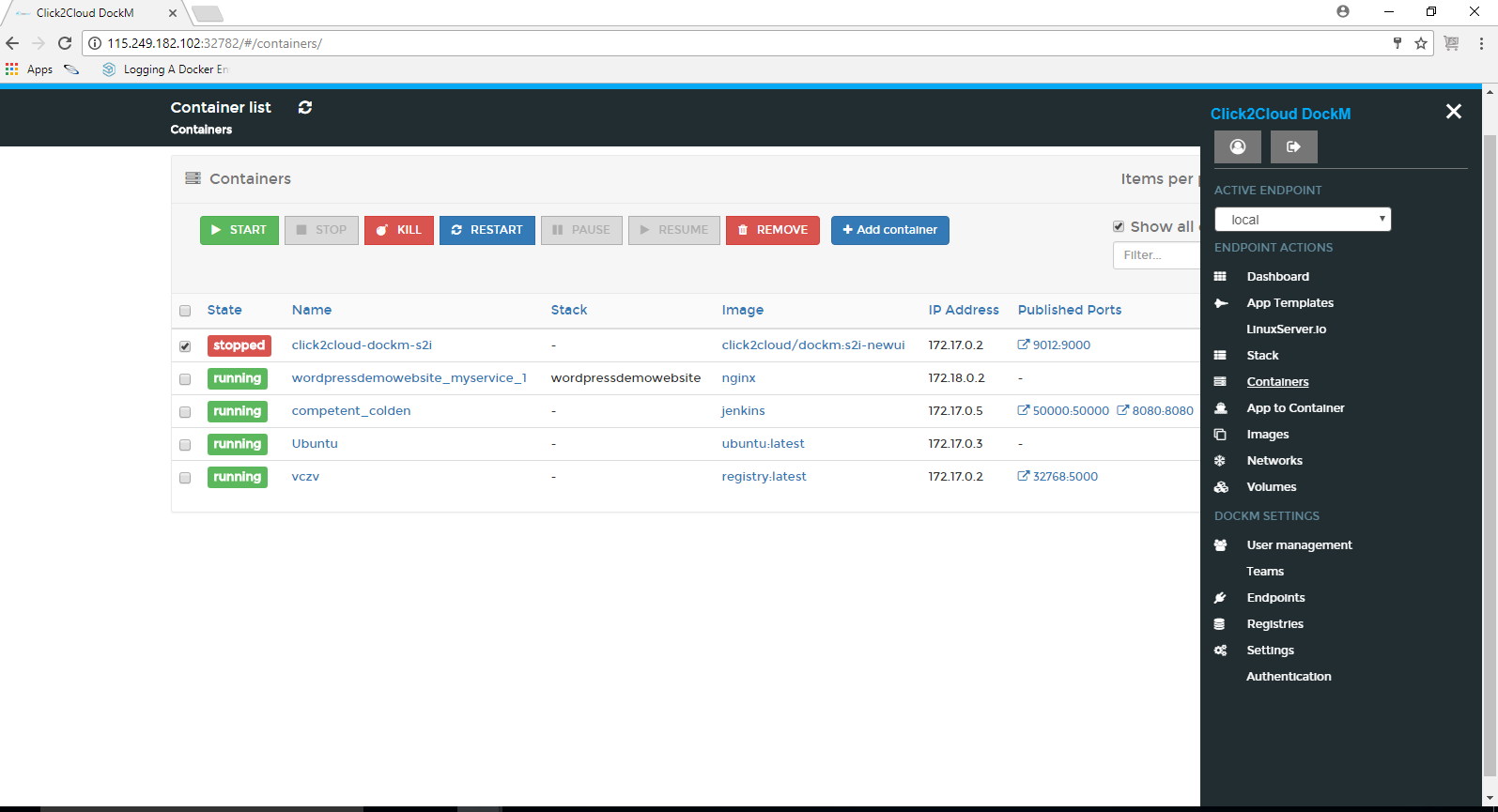


To create a Container, just hit Add Container button on the top. Enter the Container name and Image name. click on **map additional port** and set port number.

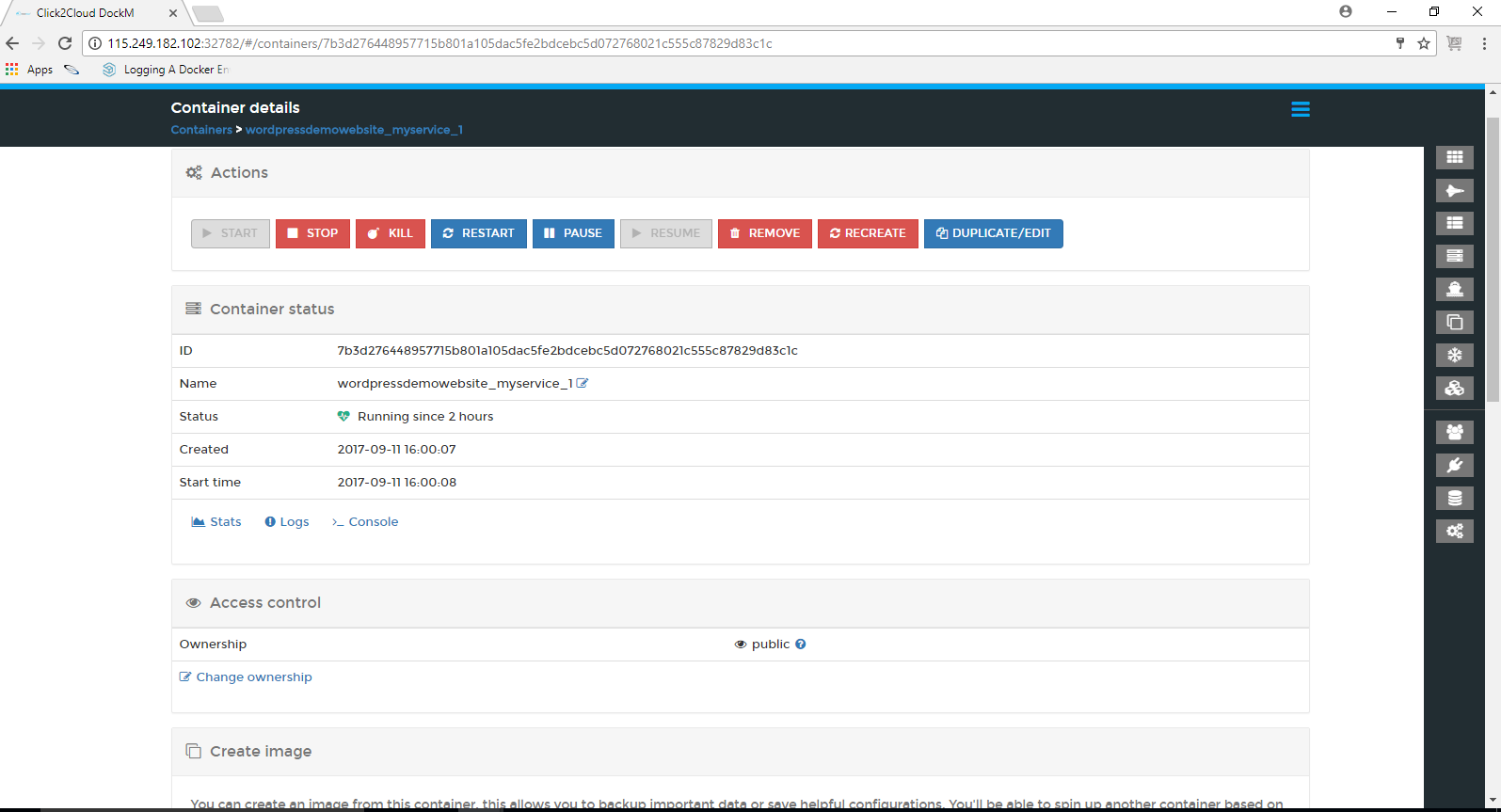
Finally, Click **Start Container**.



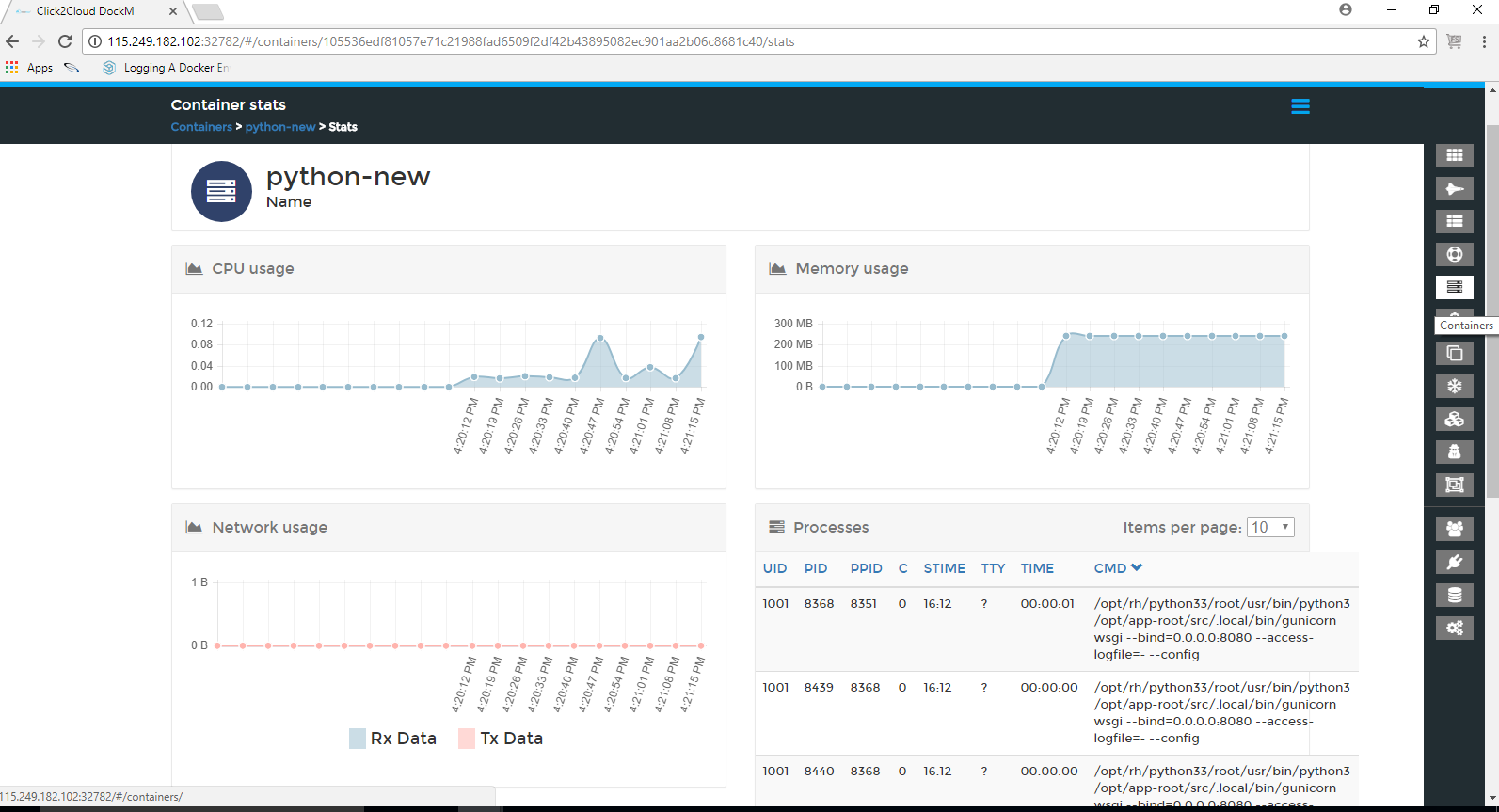
To start a newly created container, just select it and hit Start button on the top. We can restart, stop, pause, and remove any Containers from this section.



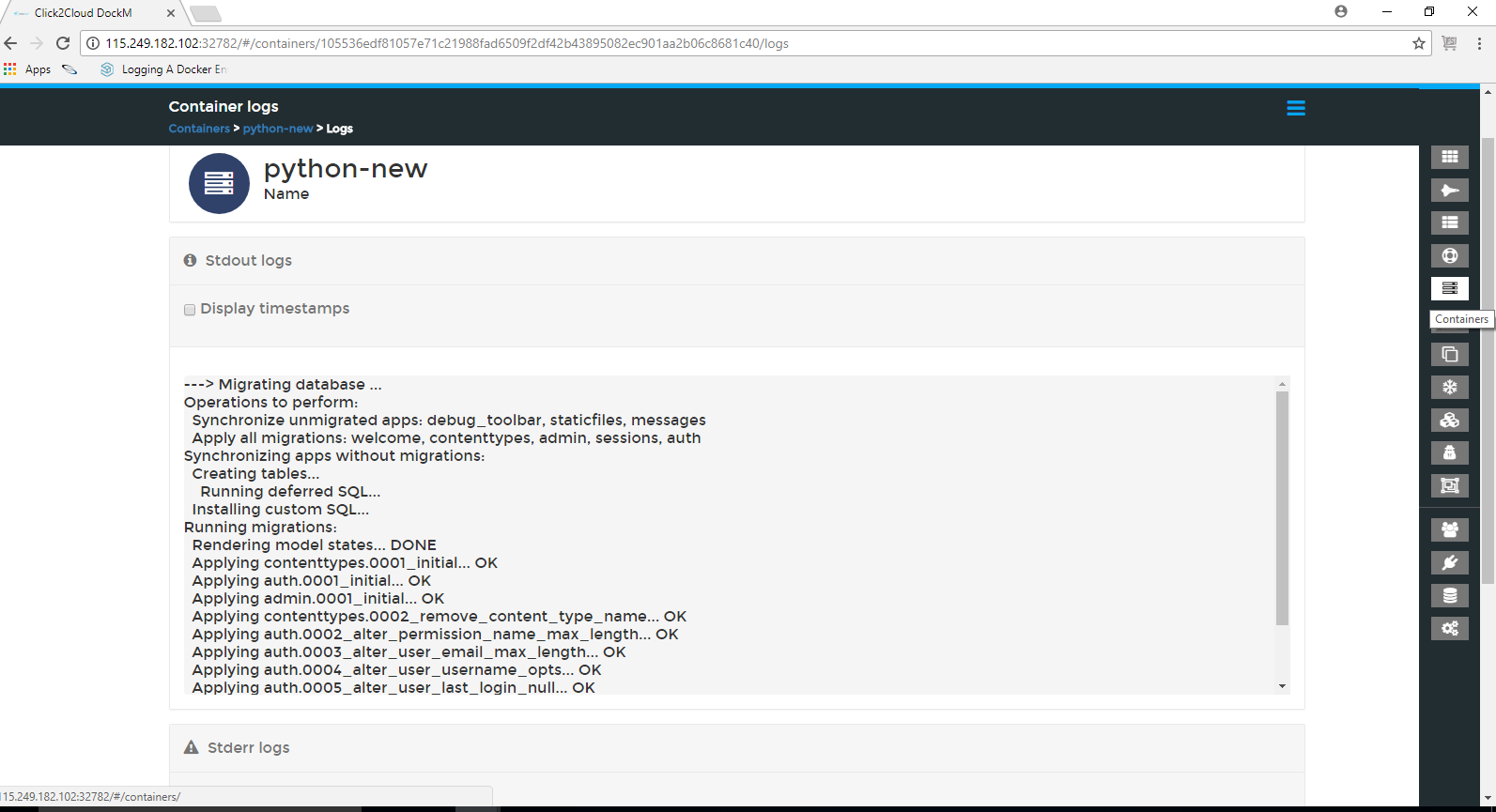
To view the complete overview of any container, just click on the name of the container.



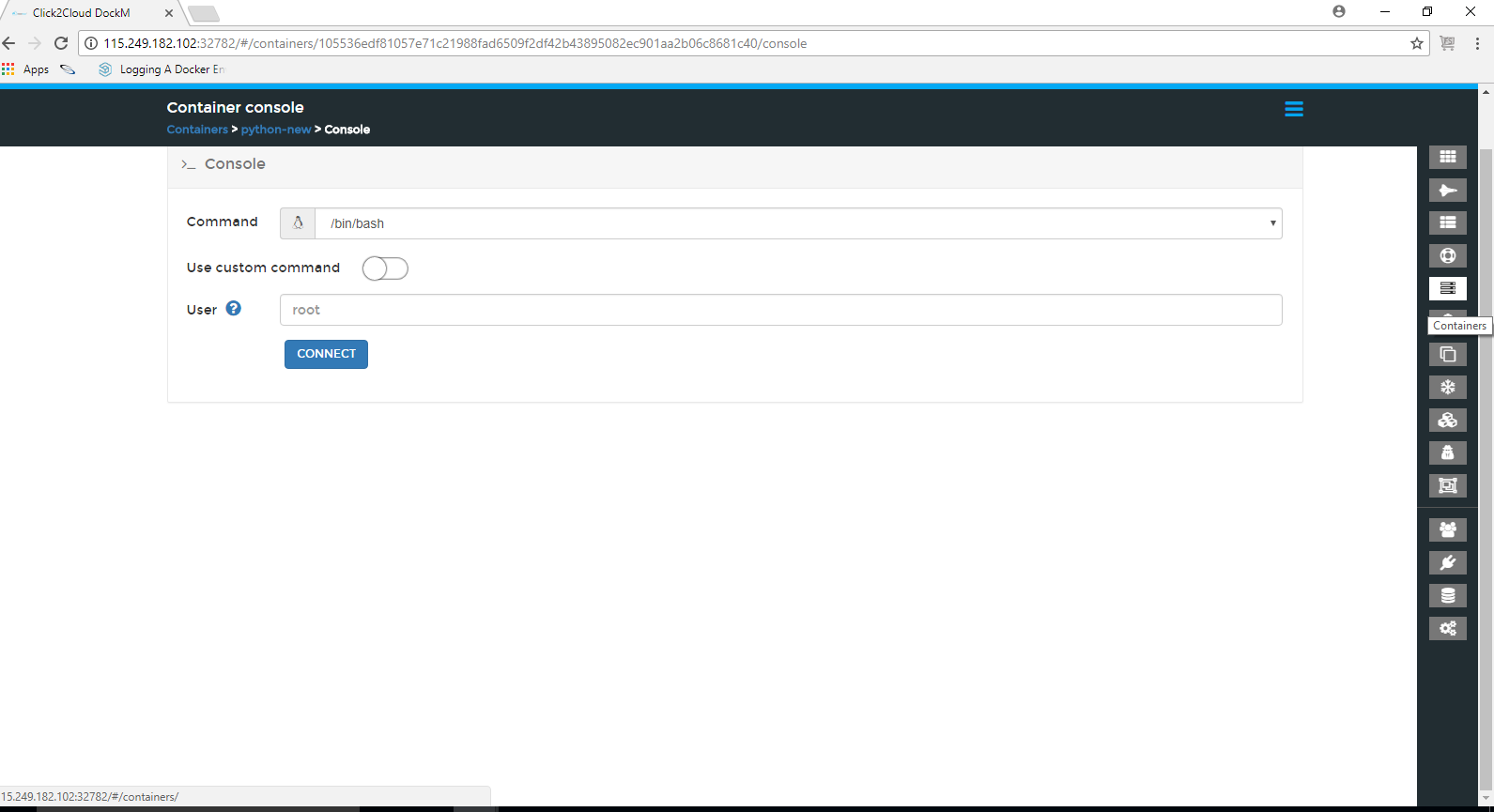
Click on the Stats button to view what’s happening in the newly launched Container.



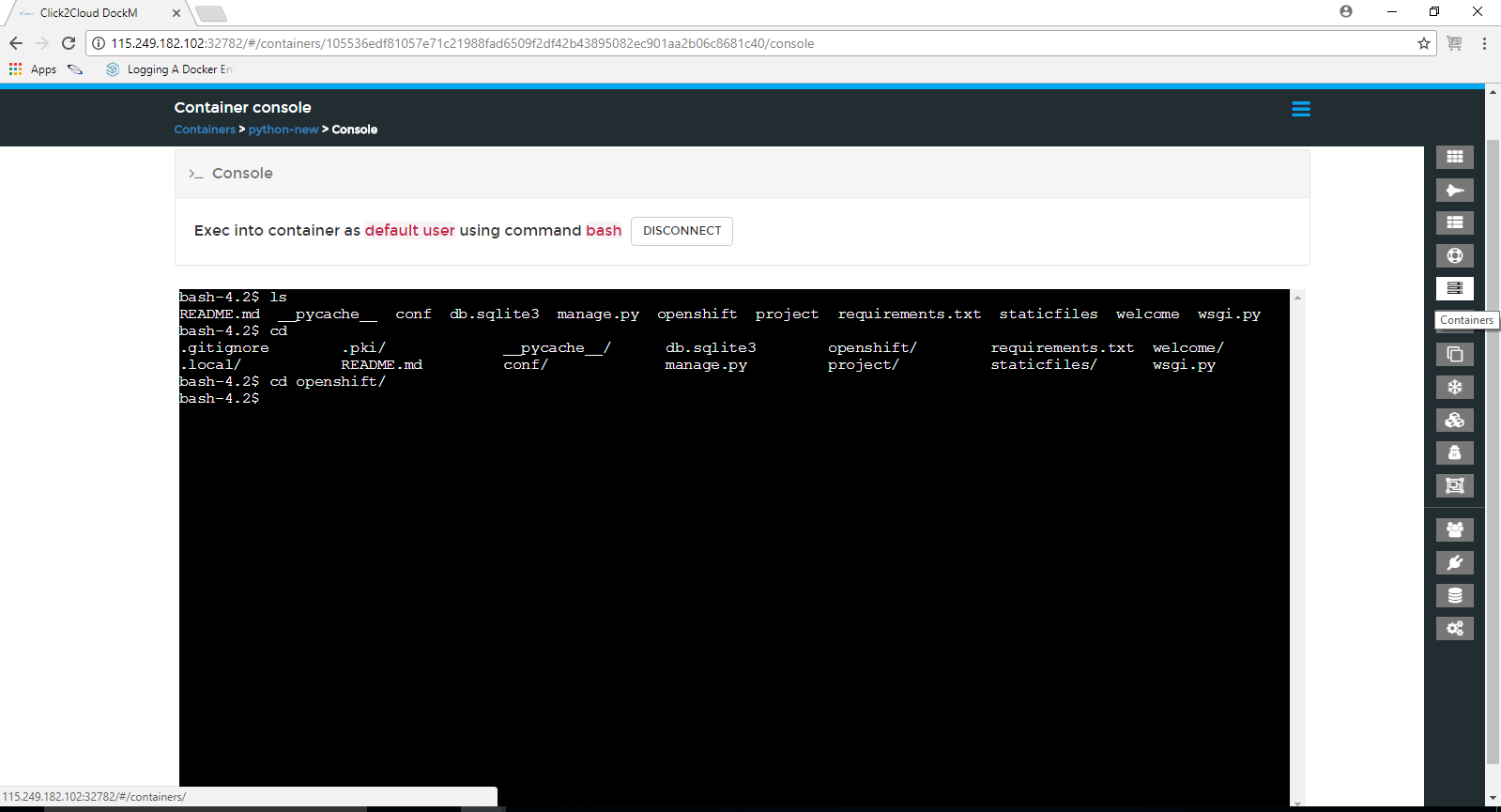
Similarly, click on the **Logs** button to view complete log details of the Container.



Another notable feature is you can easily connect to the console of your Container by clicking on the **CONNECT** button.



You can see the Console of the container, and disconnect it by clicking on **DISCONNECT** button.



### **1.2.6 App To Container**

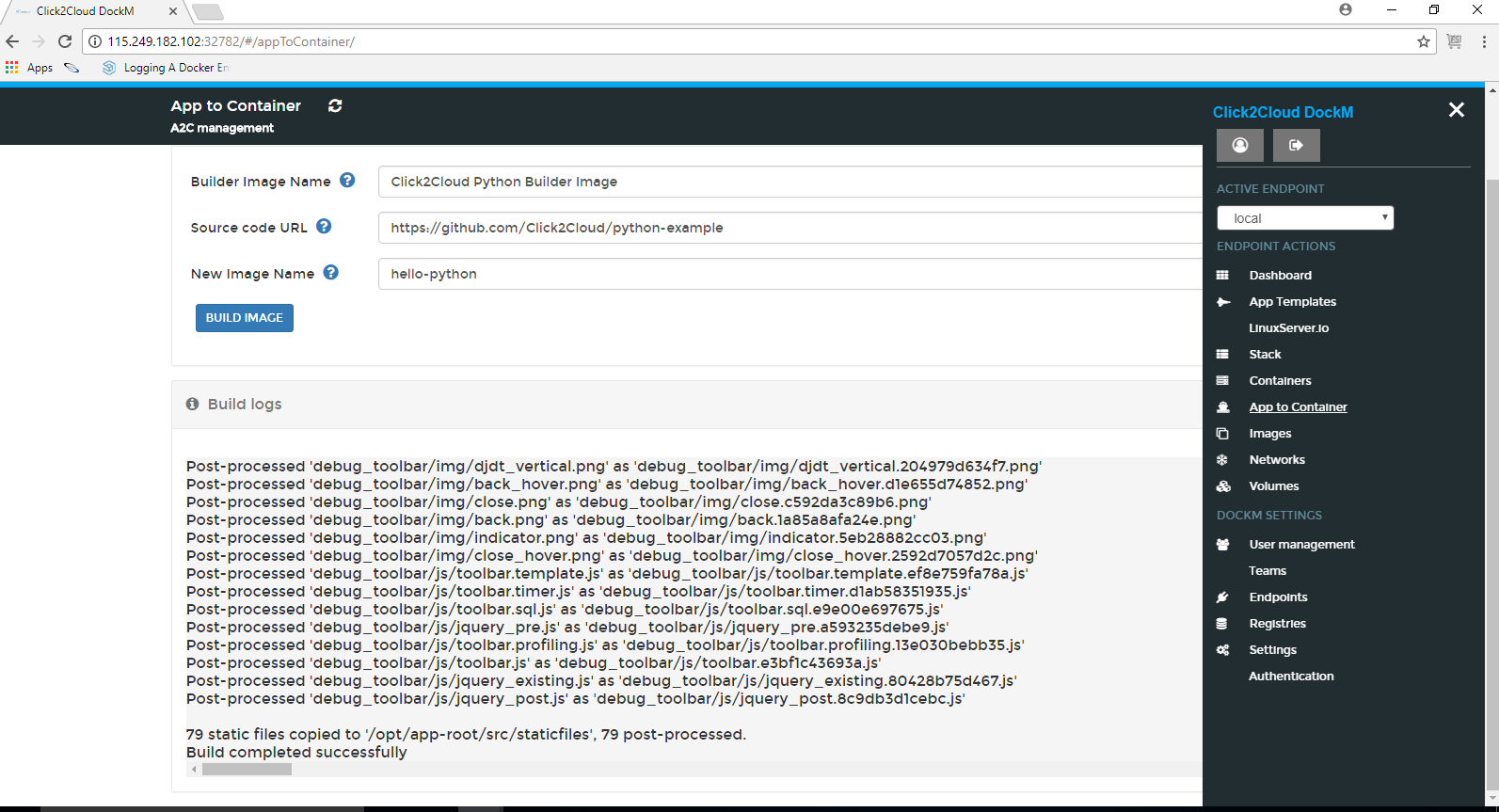
App to Container can be used for building Docker images from source code. It produces ready-to-run images by injecting source code into a Container and letting the Container prepare that source code for execution. By creating self-assembling **builder images**, we can version and control your build environments exactly like we use Docker images to version our runtime environments.

The **Builder image name** is nothing but the Docker image to be used in building the final image.

The **Source Code** **URL** is the URL of a Git repository or a local path to the source code.

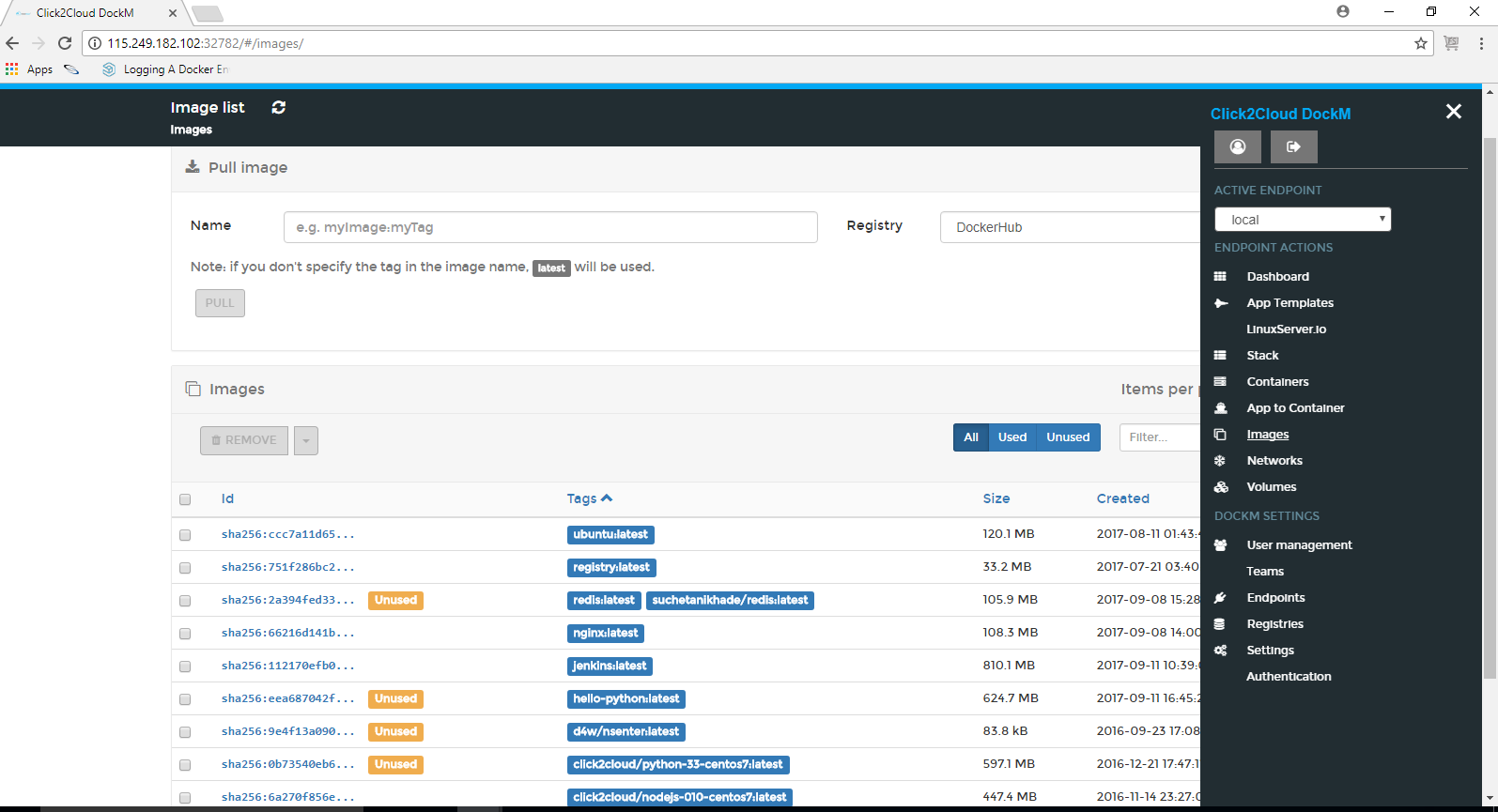
The **New image name** is the name of the final Docker image.

Here We can see the Complete Builder logs.

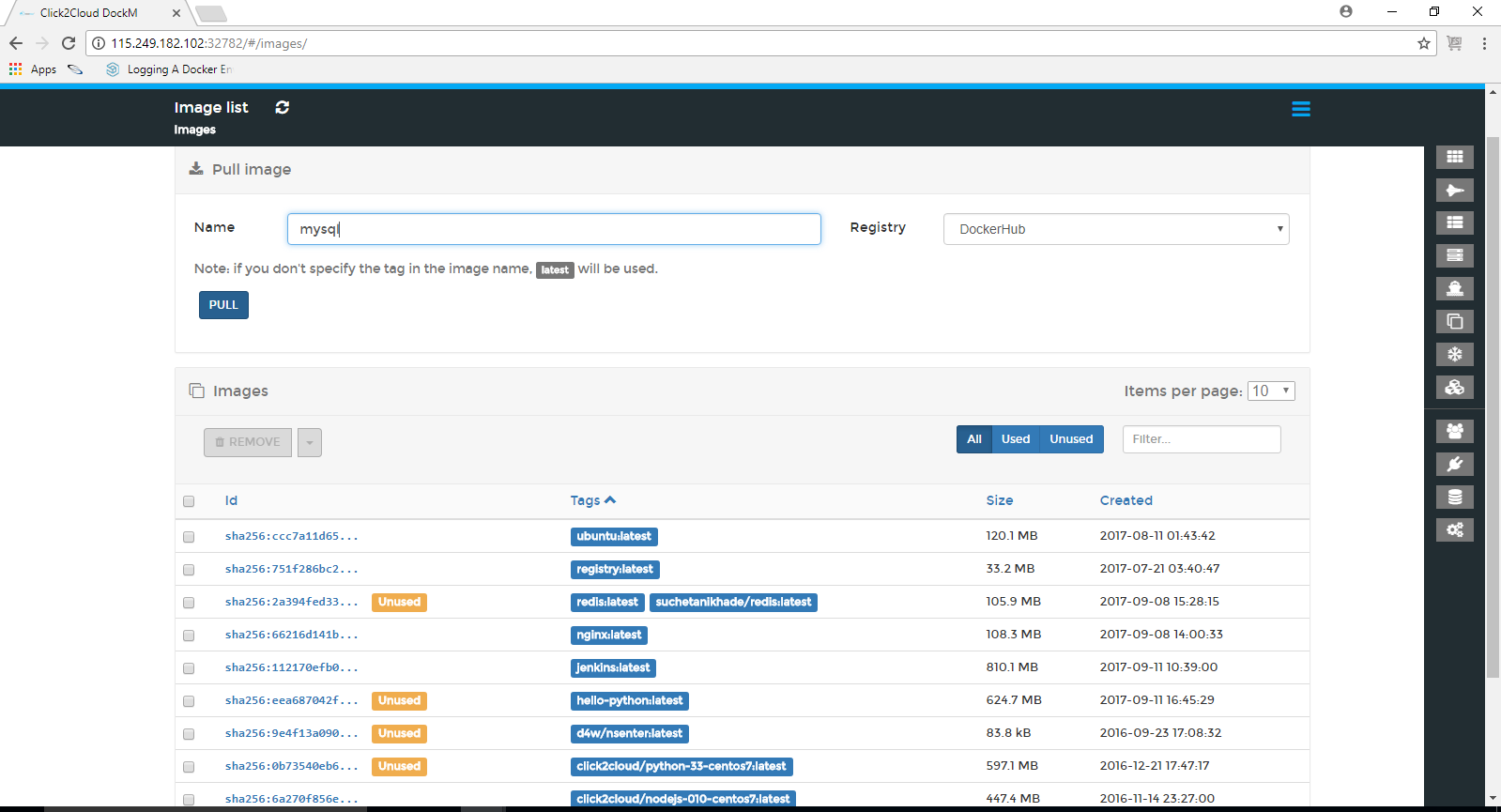


### **1.2.7 Docker Images**

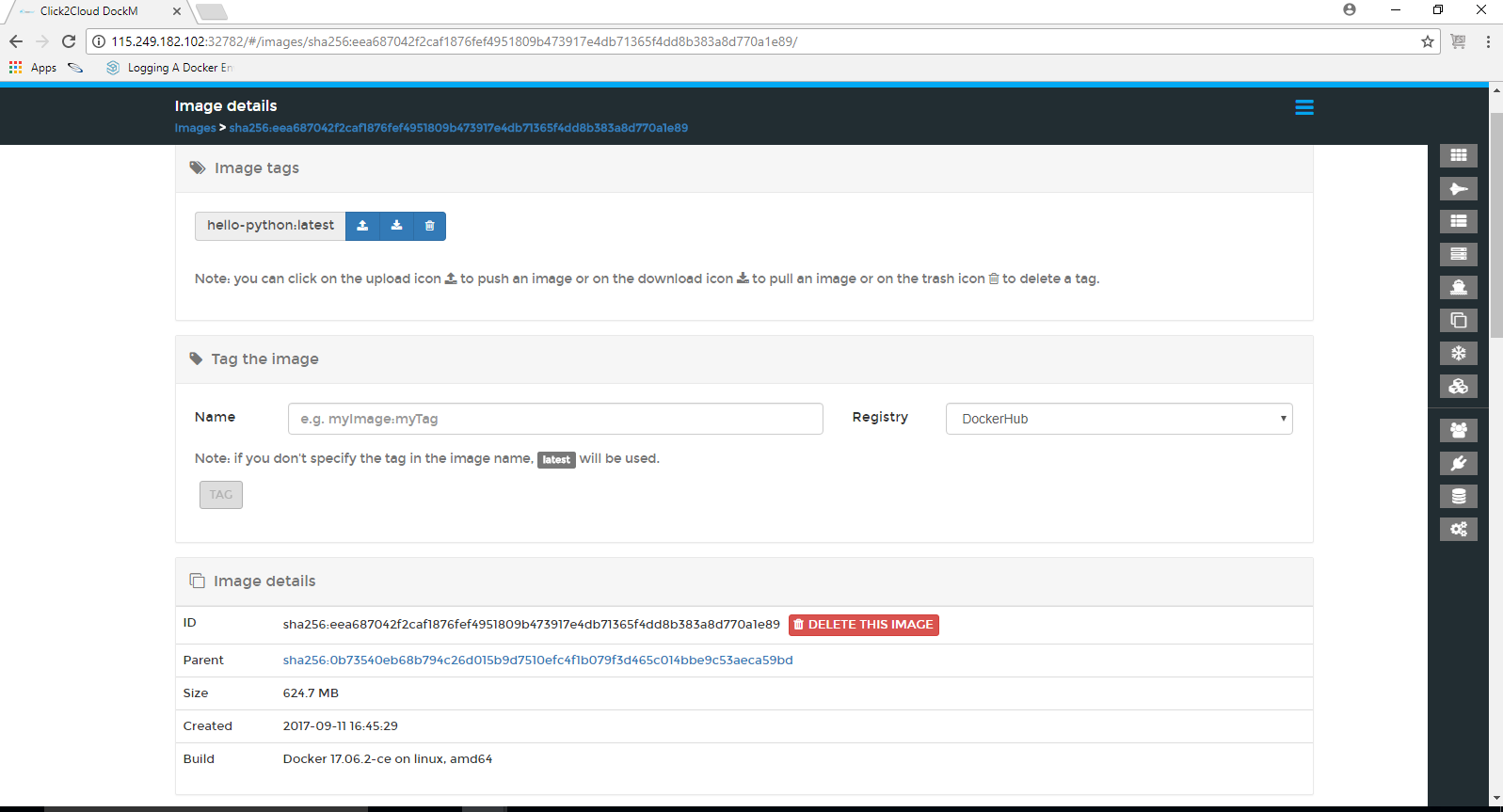
In this section, we can view the list of downloaded docker images.



You can pull the image from any docker registry by specifying the image name.



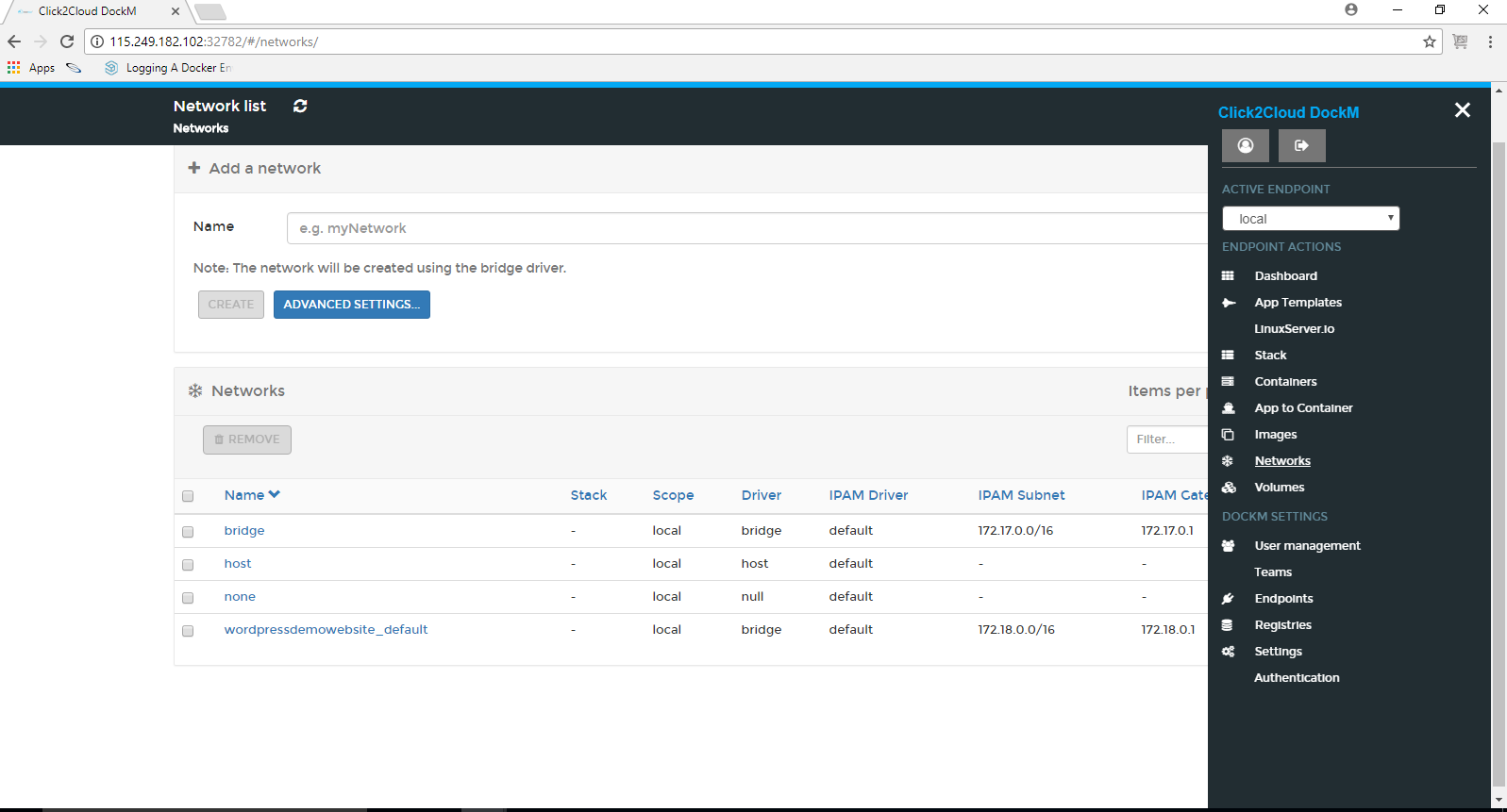
To view the complete overview of any image, just click on the name of the image.



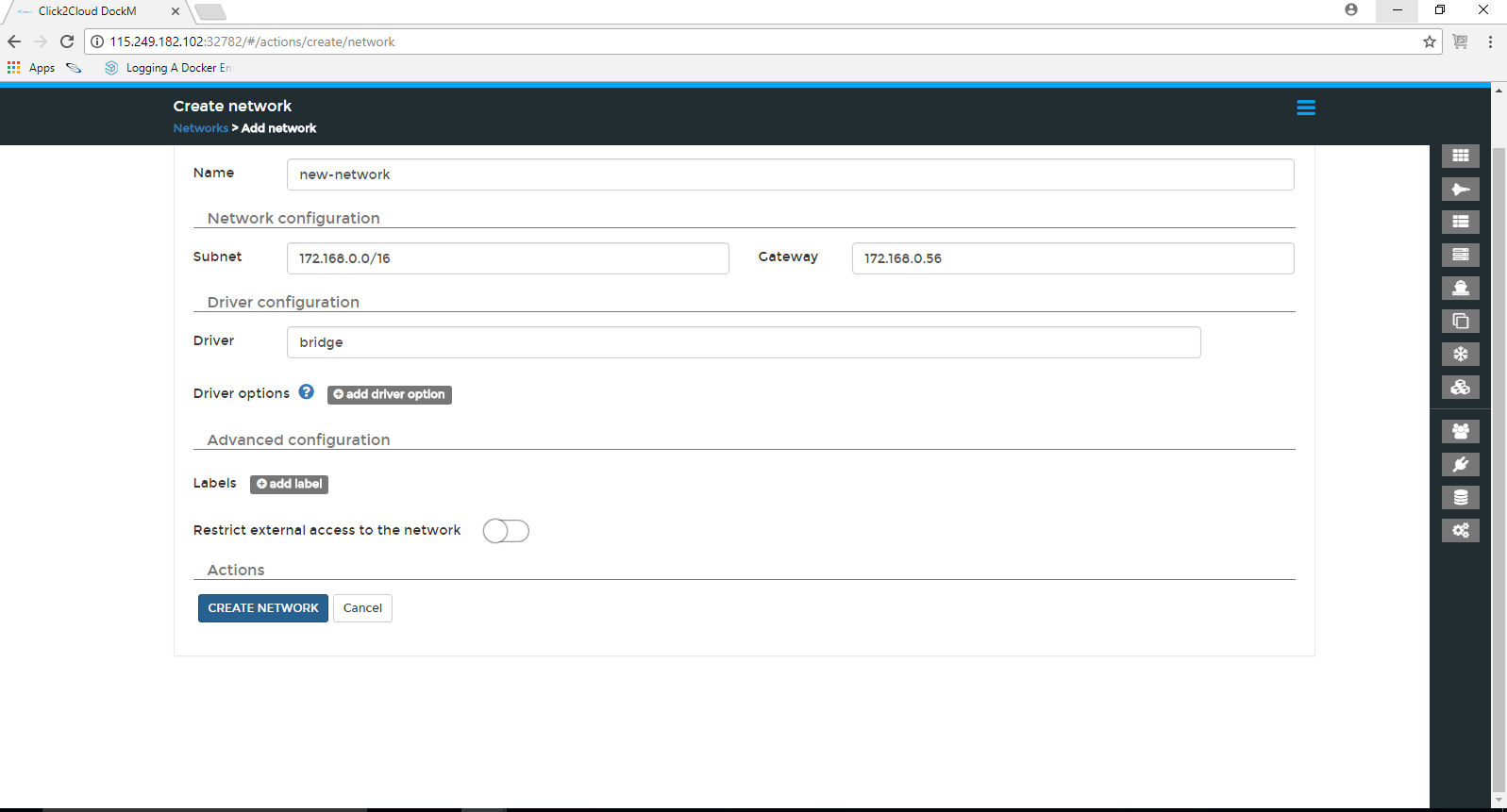
To remove any image, just select it and click **Remove**. The selected image will be removed.

### **1.2.8 Networks**

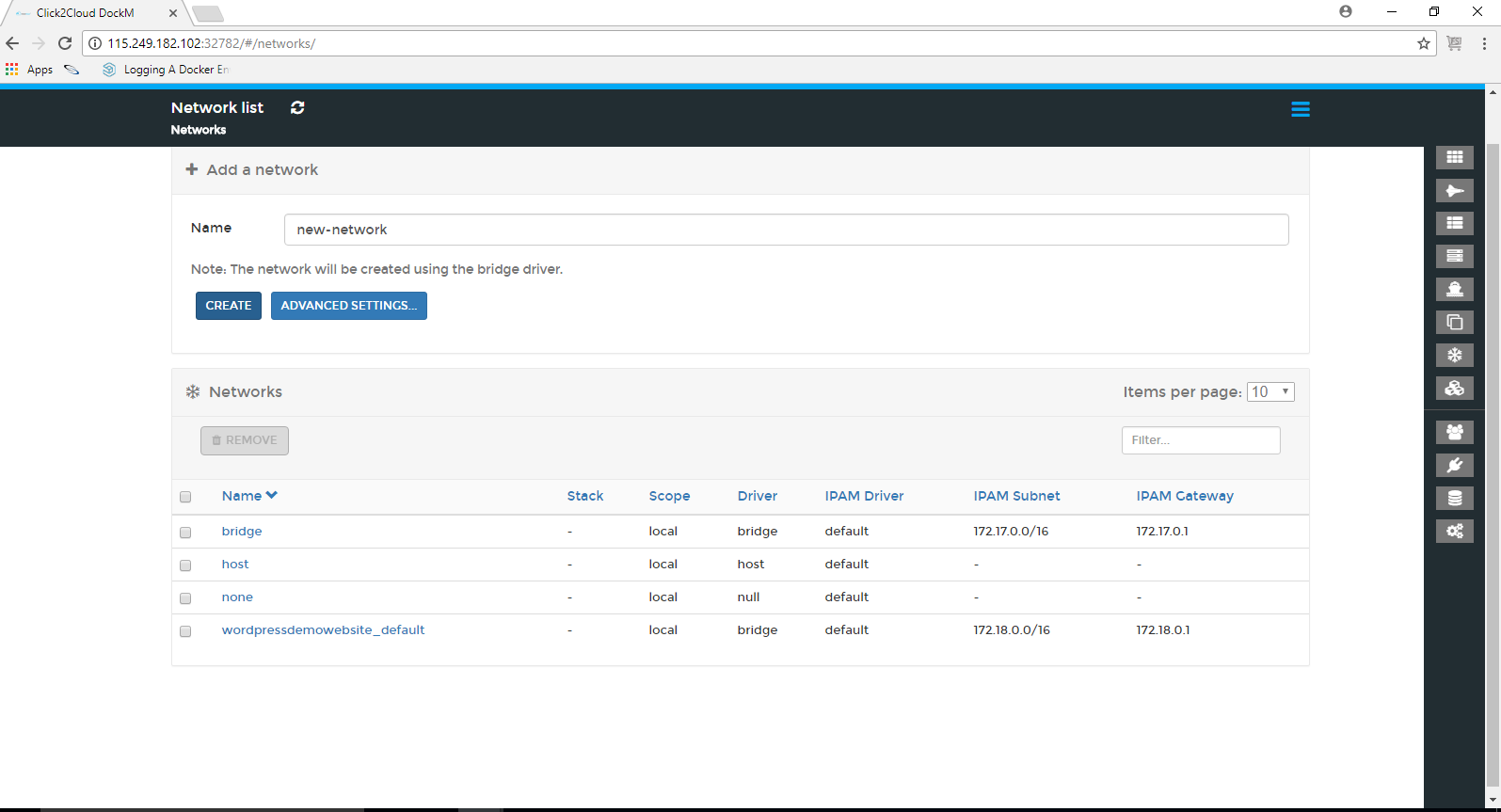
This section allows to add a new network, change the network type, assign/change IP address, remove existing networks.



Clicking on Advanced settings buttons shows the Create network page providing the options for Name, Subnet, Gateway, Driver, Driver options, Labels for the network.

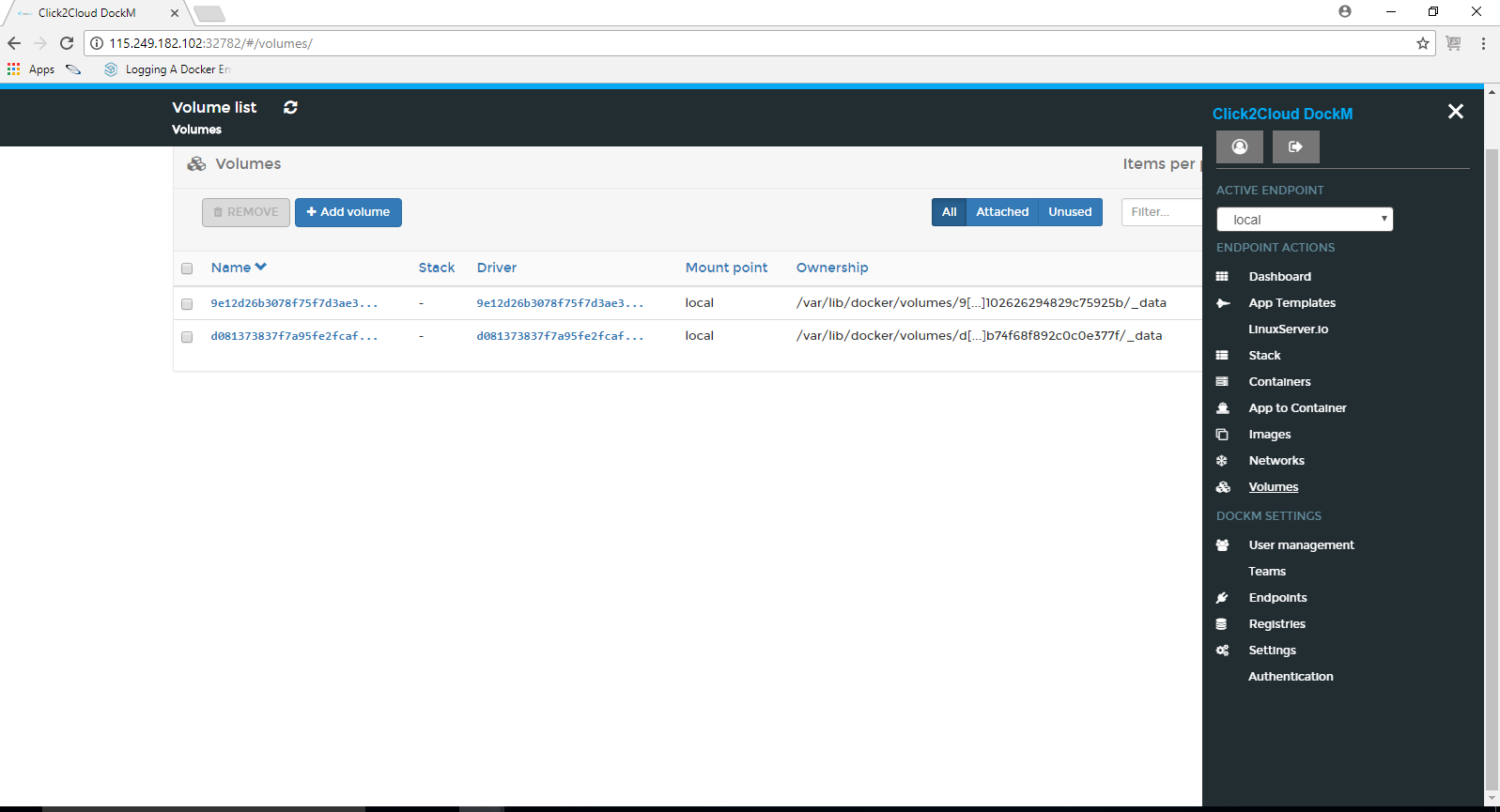


To create a Network without adding the Network configuration, Driver Configuration and Advanced Configuration. Just enter the Network name and click on **CREATE** button. The Network will be created using the bridge driver.

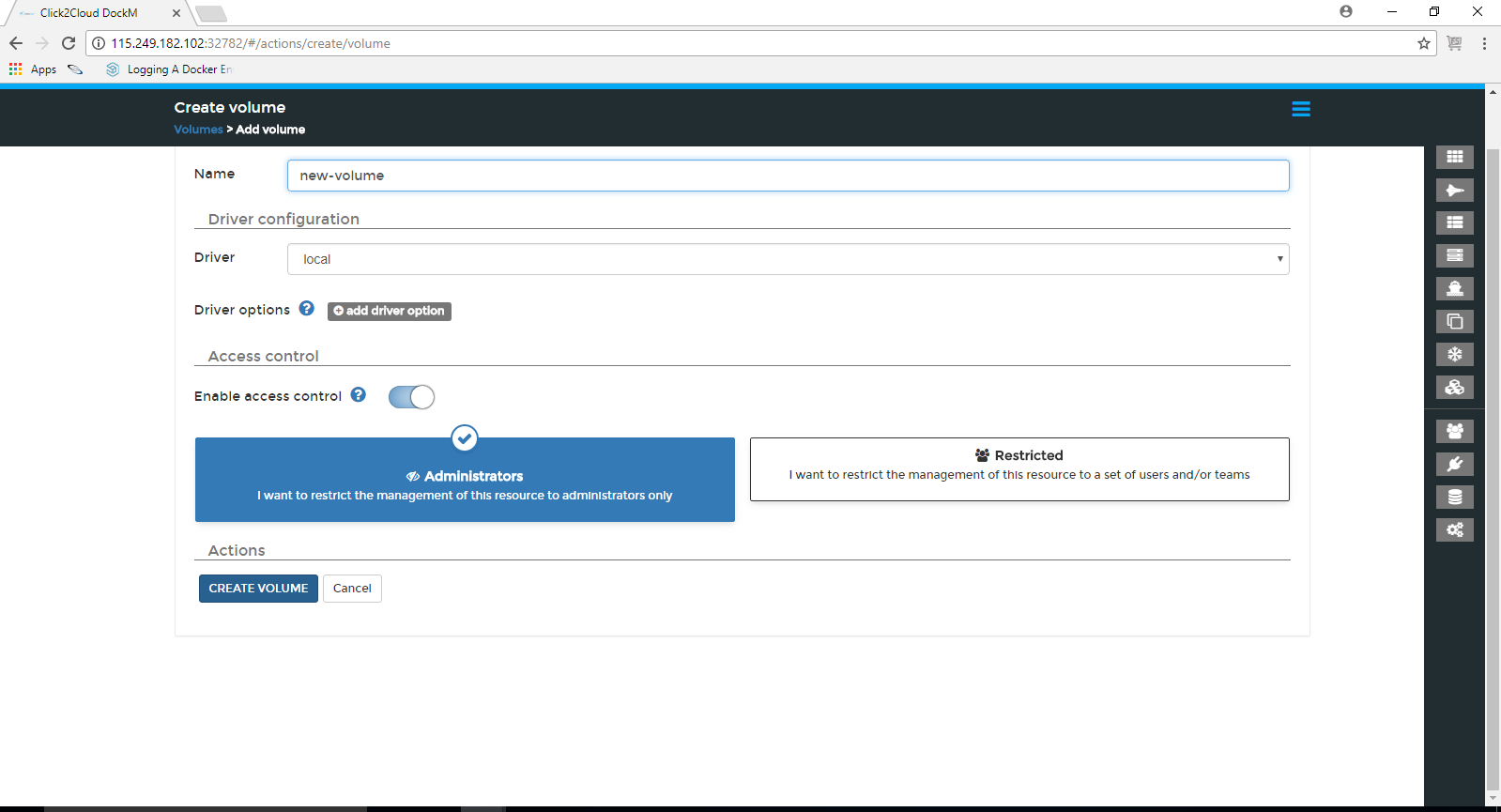


### **1.2.9 Volumes**

Here we can view existing Docker volumes, create new one, delete them if we no longer need them.

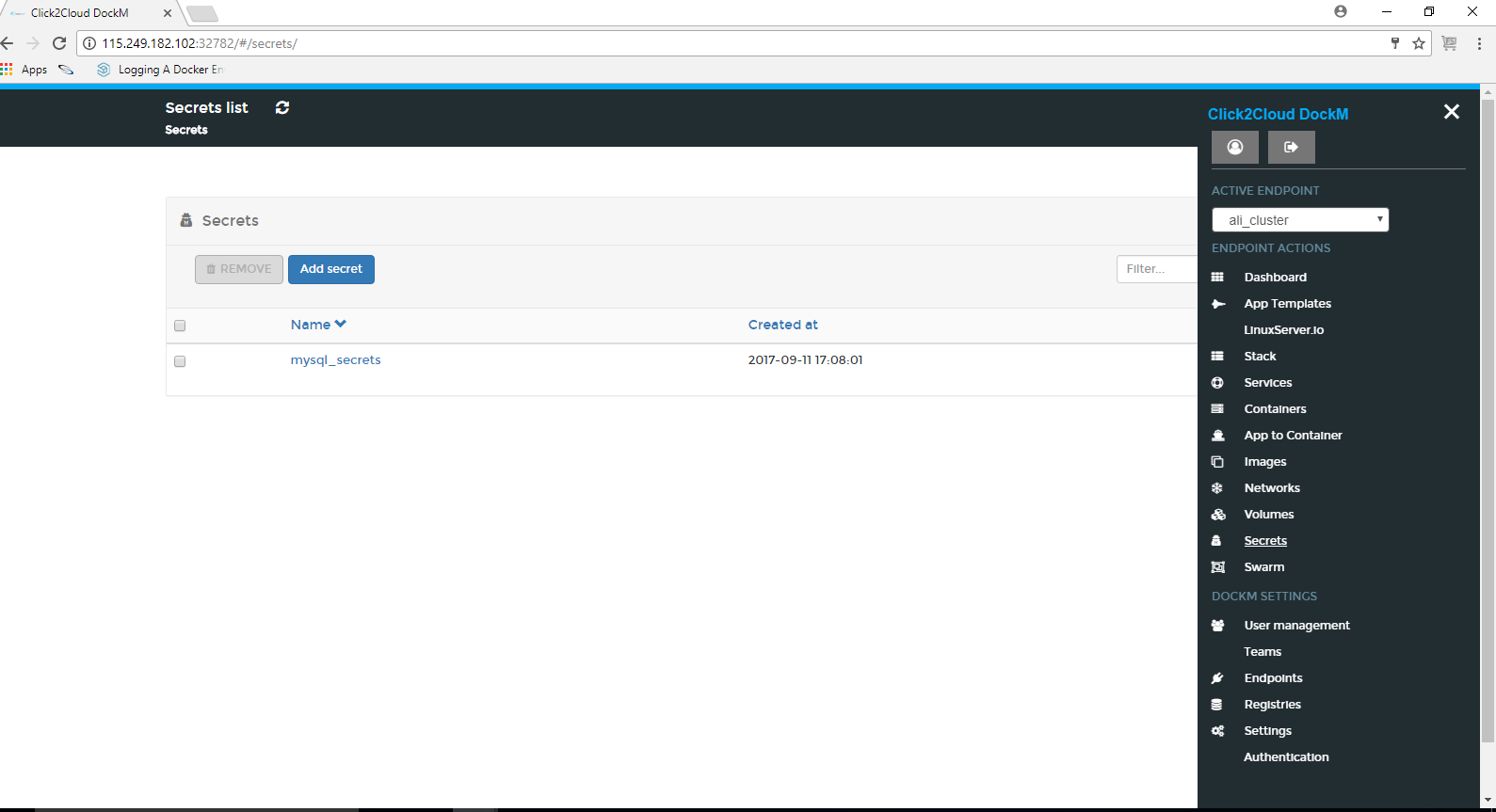


Clicking on Add volume button shows the Create volume page containing Name, Driver, Driver options, Access control options.

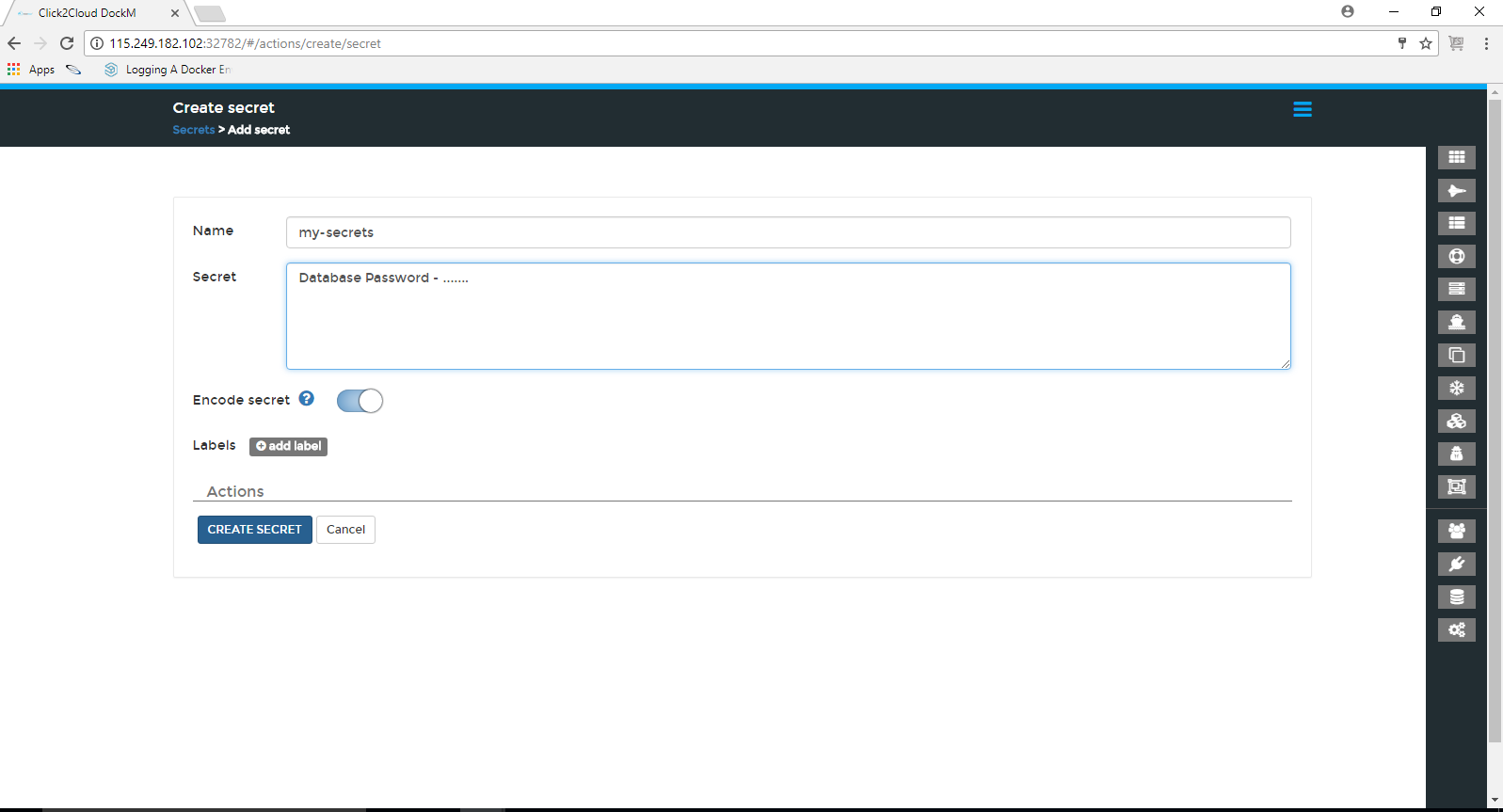


### **1.2.10 Secrets**

This section allows to add a secret and remove existing secrets. We can see the added secrets list with the created date and time.



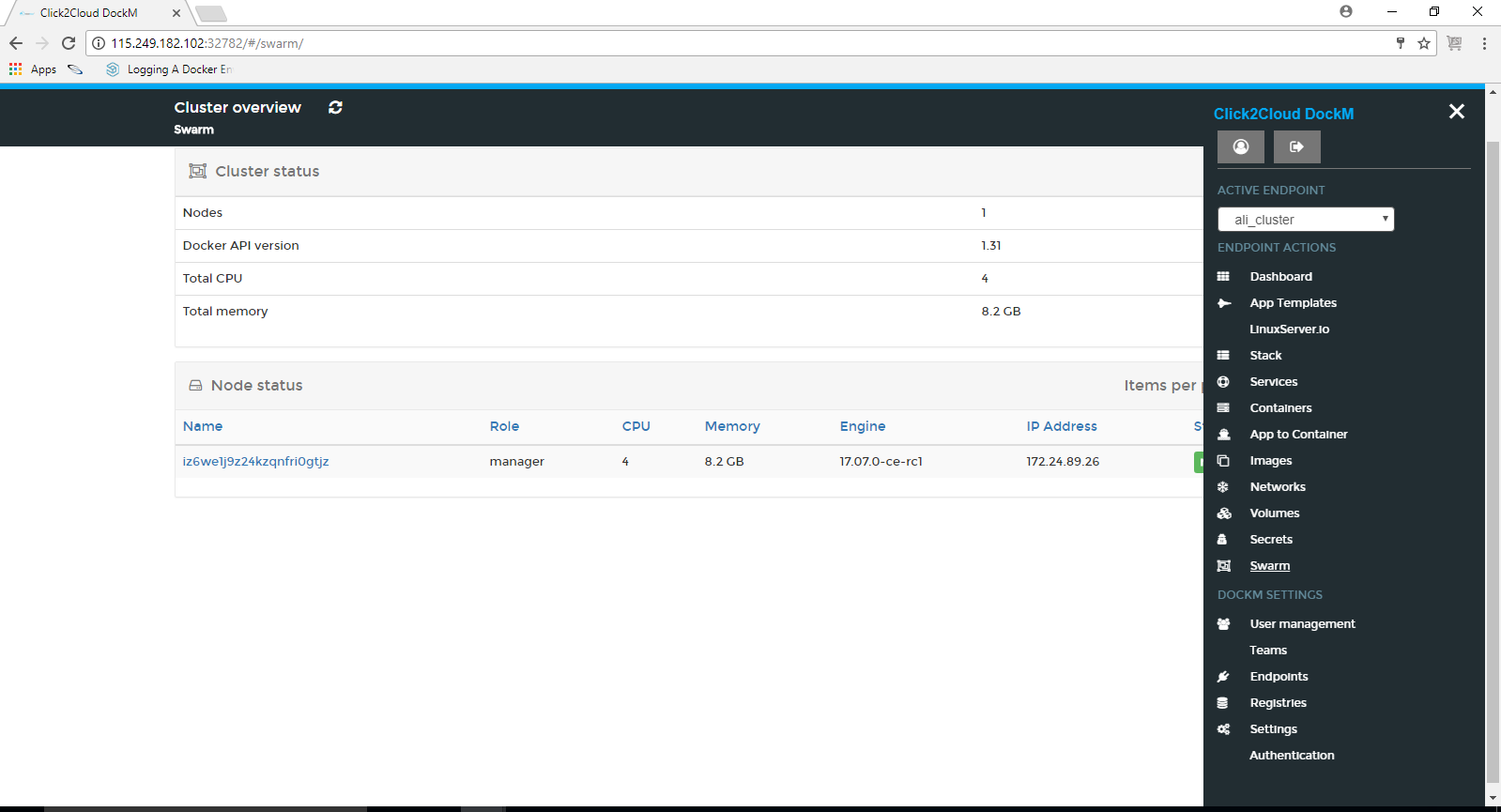
Clicking on Add secret button shows the Create secret page containing Name, Secret, Encode options, Labels.



To remove any secrets, just select it and click **Remove**. The selected secrets will be removed.

### **1.2.11 Swarm**

In Swarm, we can see the Cluster overview which is nothing but Cluster status and Node status, which Manage the swarm.



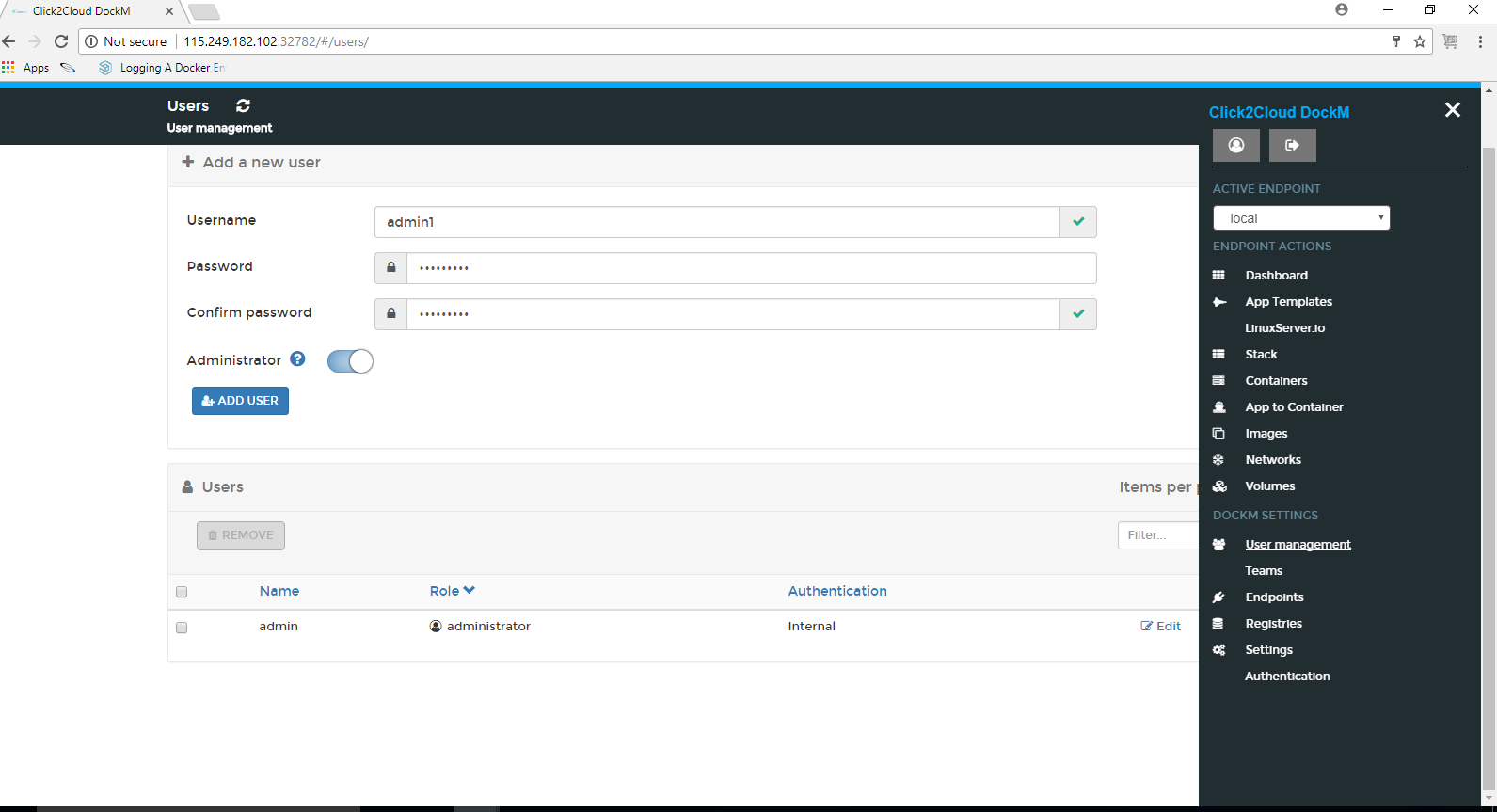
## **1.3 DockM Setting:**

This section provides the option for User Management, Endpoints, Registries and, Application Settings.

### **1.3.1 User Management:**

**Administrator User:**

Administrator User can create other Administrator Users, Normal Users. Administrator user can be created by enabling the Administrator button, and clicking on ADD USER button.



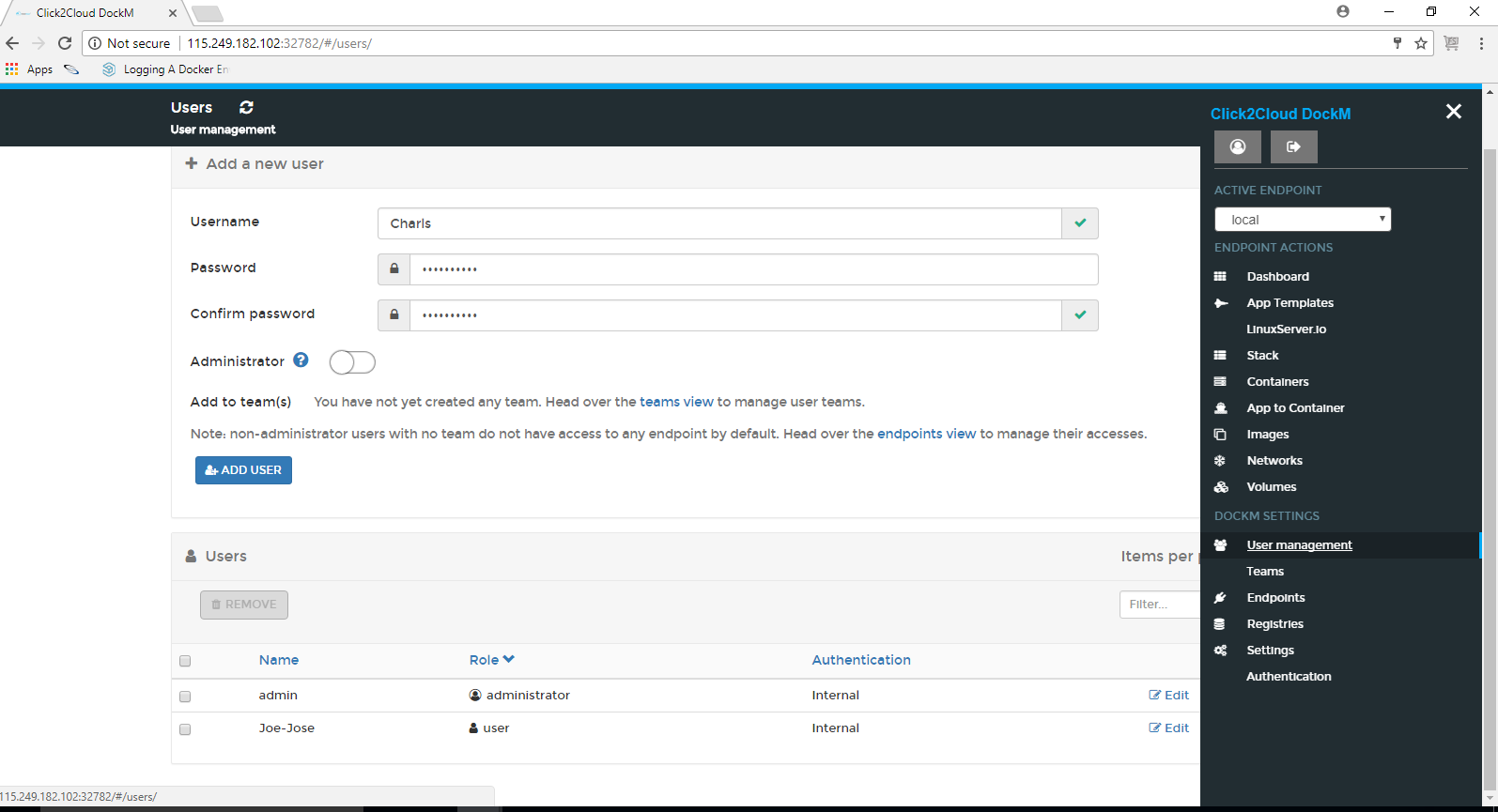
Only the Administrator User is responsible to Authorize Users, Teams, Team Leaders for the specific Endpoint.

Administrator User is the only person who has control over all the tabs in the DockM application.

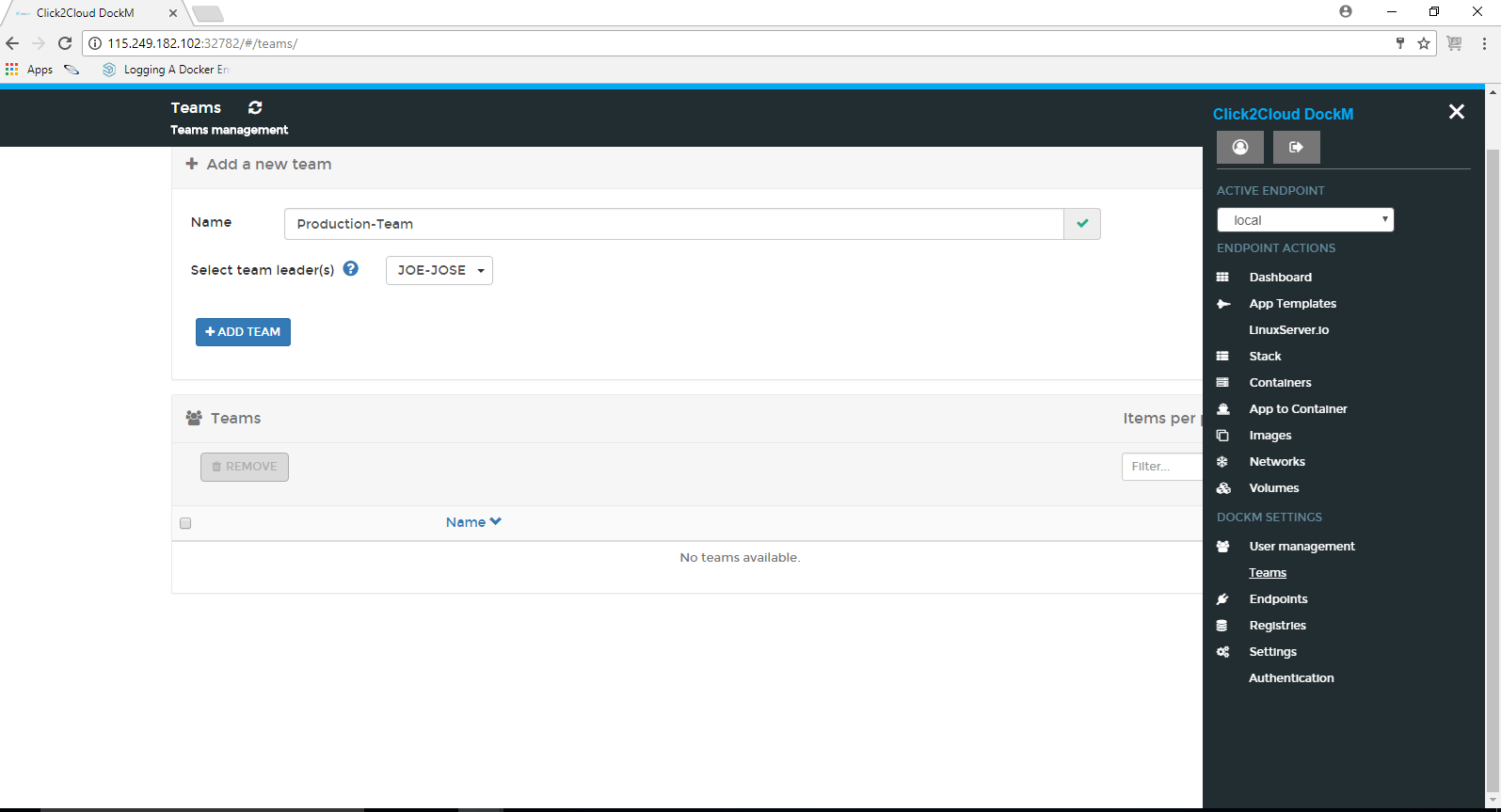
#### **1.3.1.1 Team**

**Users:**

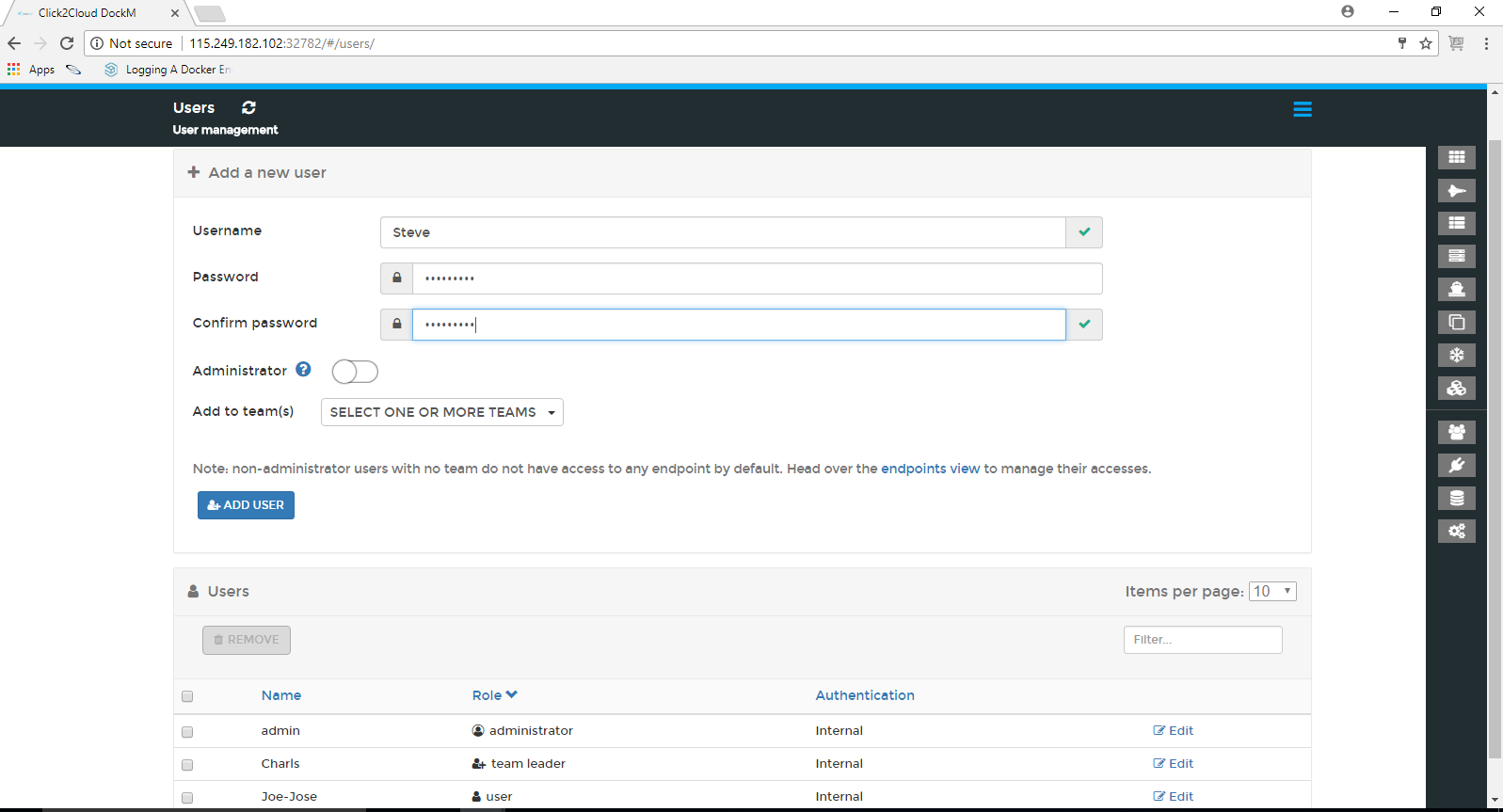
Administrator User can create many normal and Administrator User from the Users tab. A normal user can be created without enabling the Administrator button. Enter the Username and Password. click on **ADD USER**.



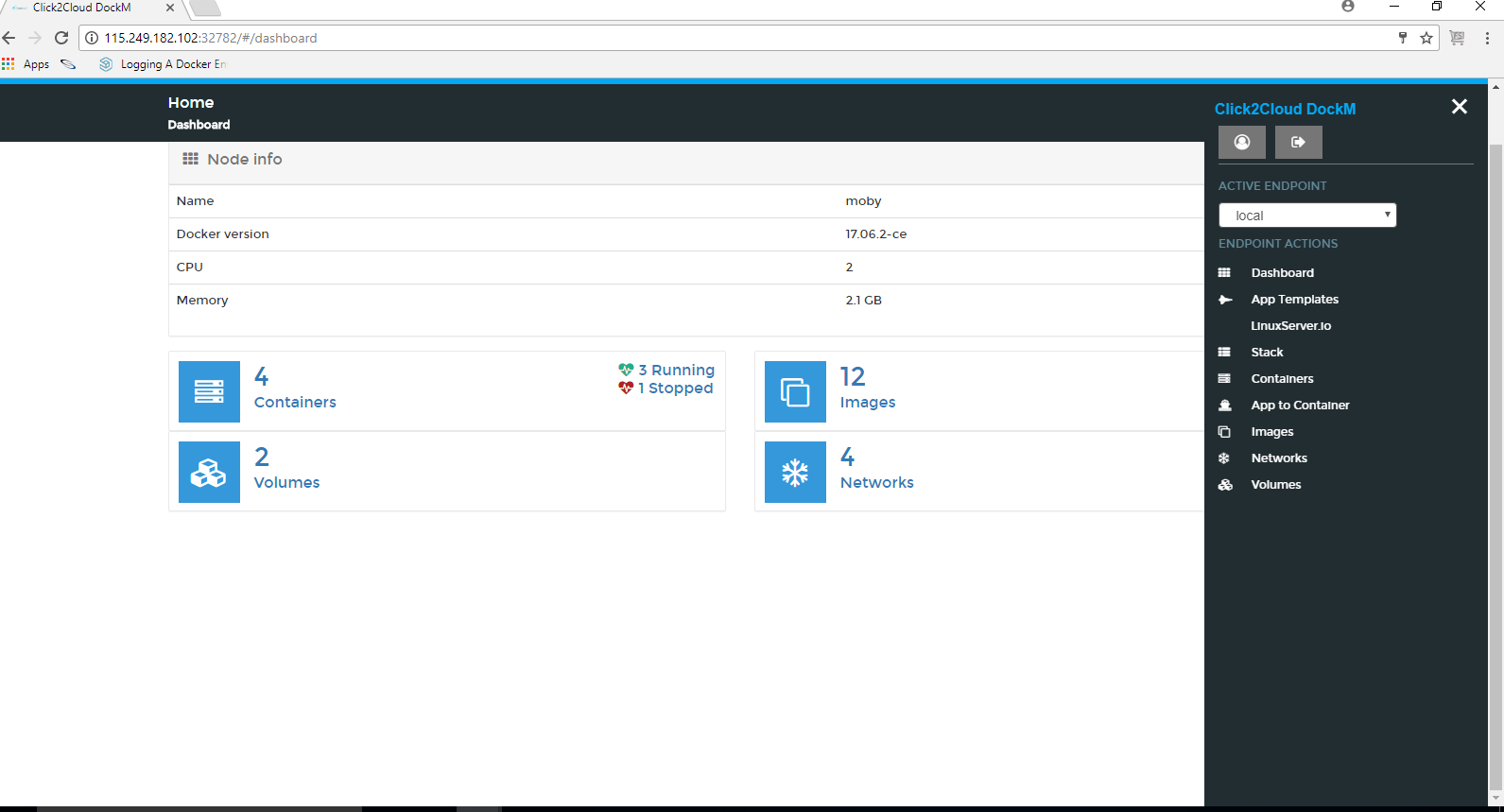
Clicking on **teams view** shows the Teams page providing the options for Name and select box for the Selecting team leader.



Click on Users tab,to check whether the team leader gets assigned role or not.

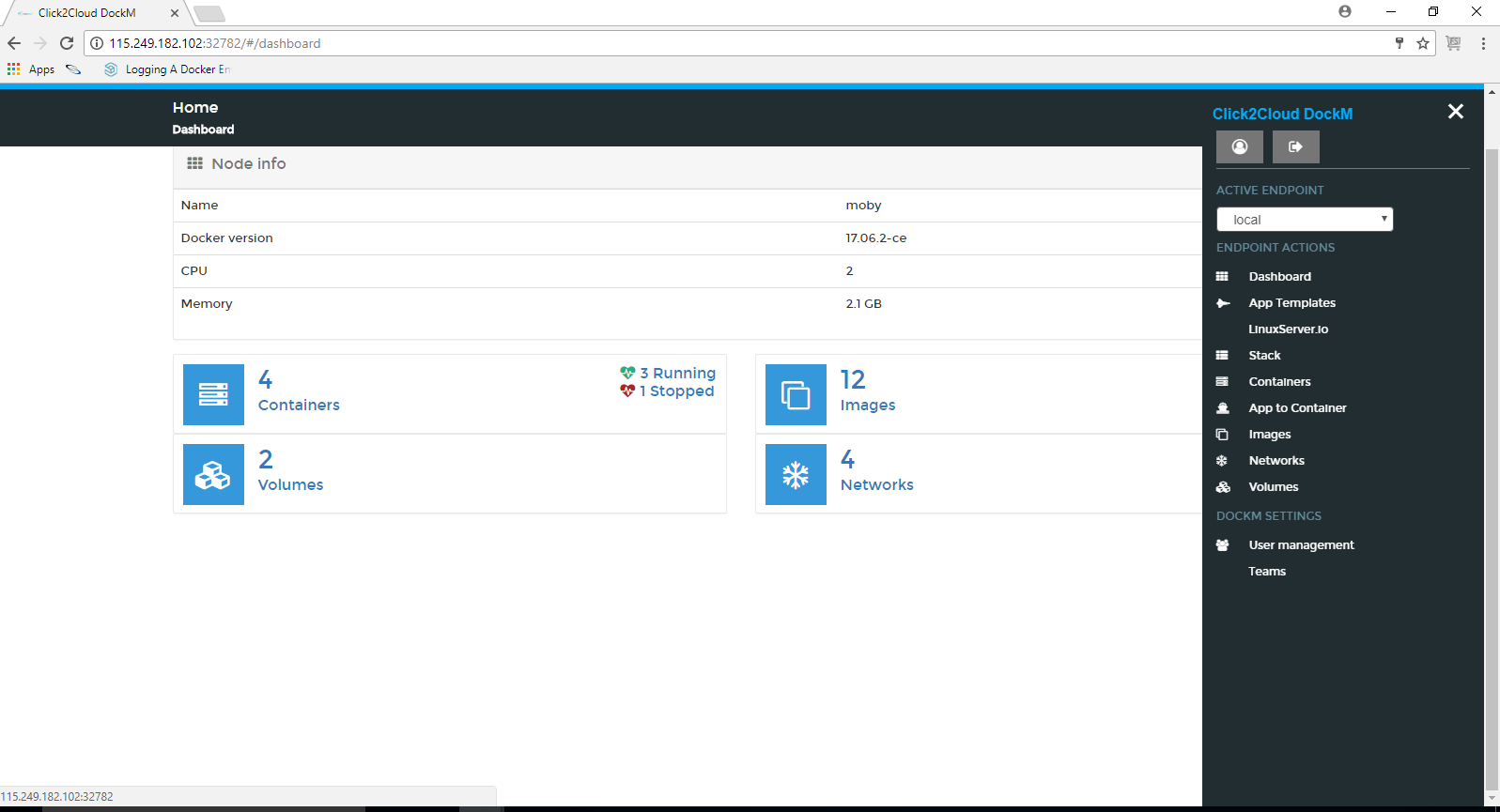


After logging in into the application, Normal User can only see the tabs present under Endpoint Actions view and cannot see the tabs under DockM setting view. Normal User can switch in between the Endpoints which are authorized by the Administrator.

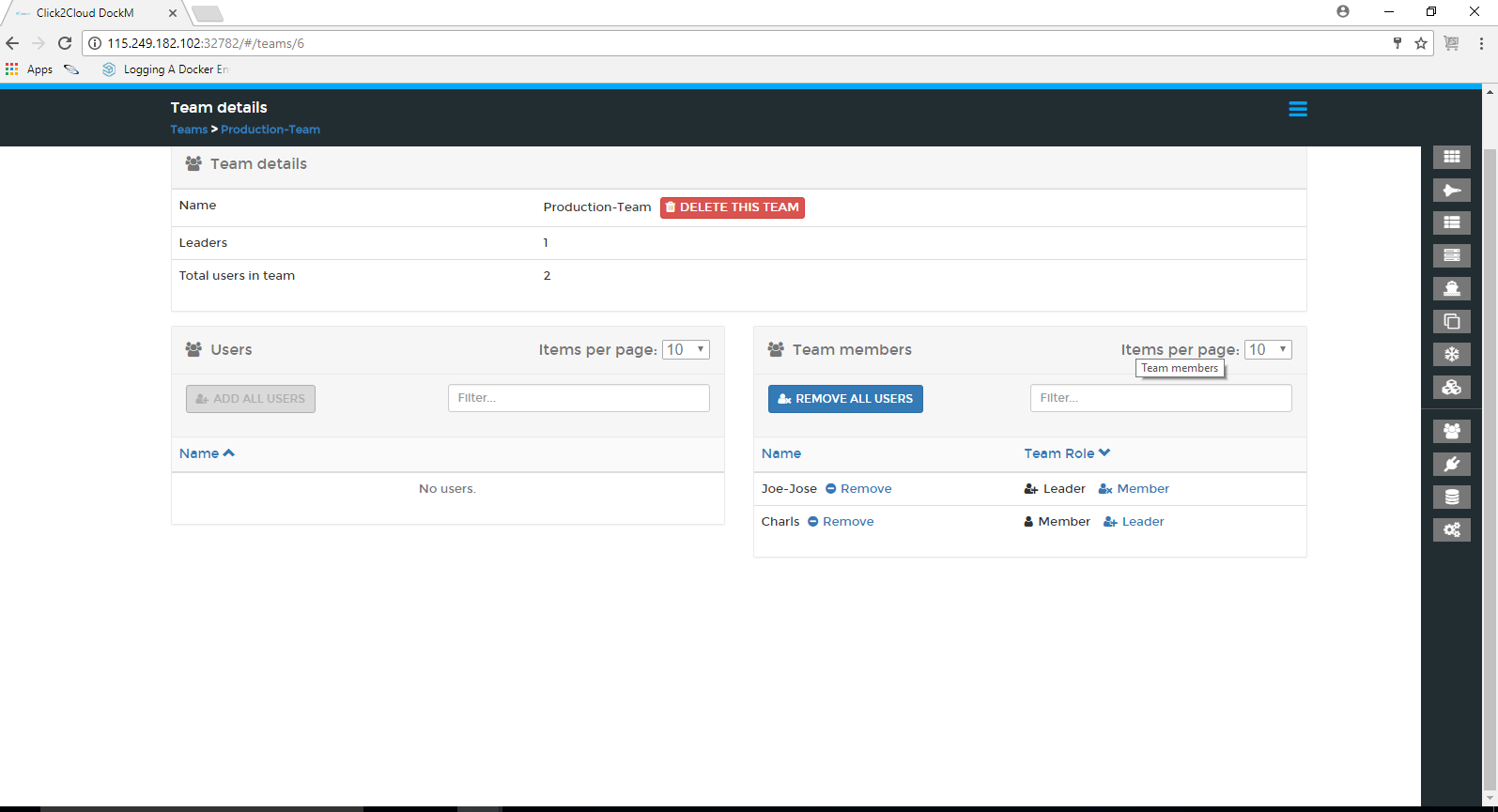


**Team Leader:**

Team Leader can only see the User Management, Teams tab from the DockM Settings view, Team Leader can add other users in the team which are created by Administrator, unless the user is authorized for the endpoint by Administrator the user will not be able to log in into the application.

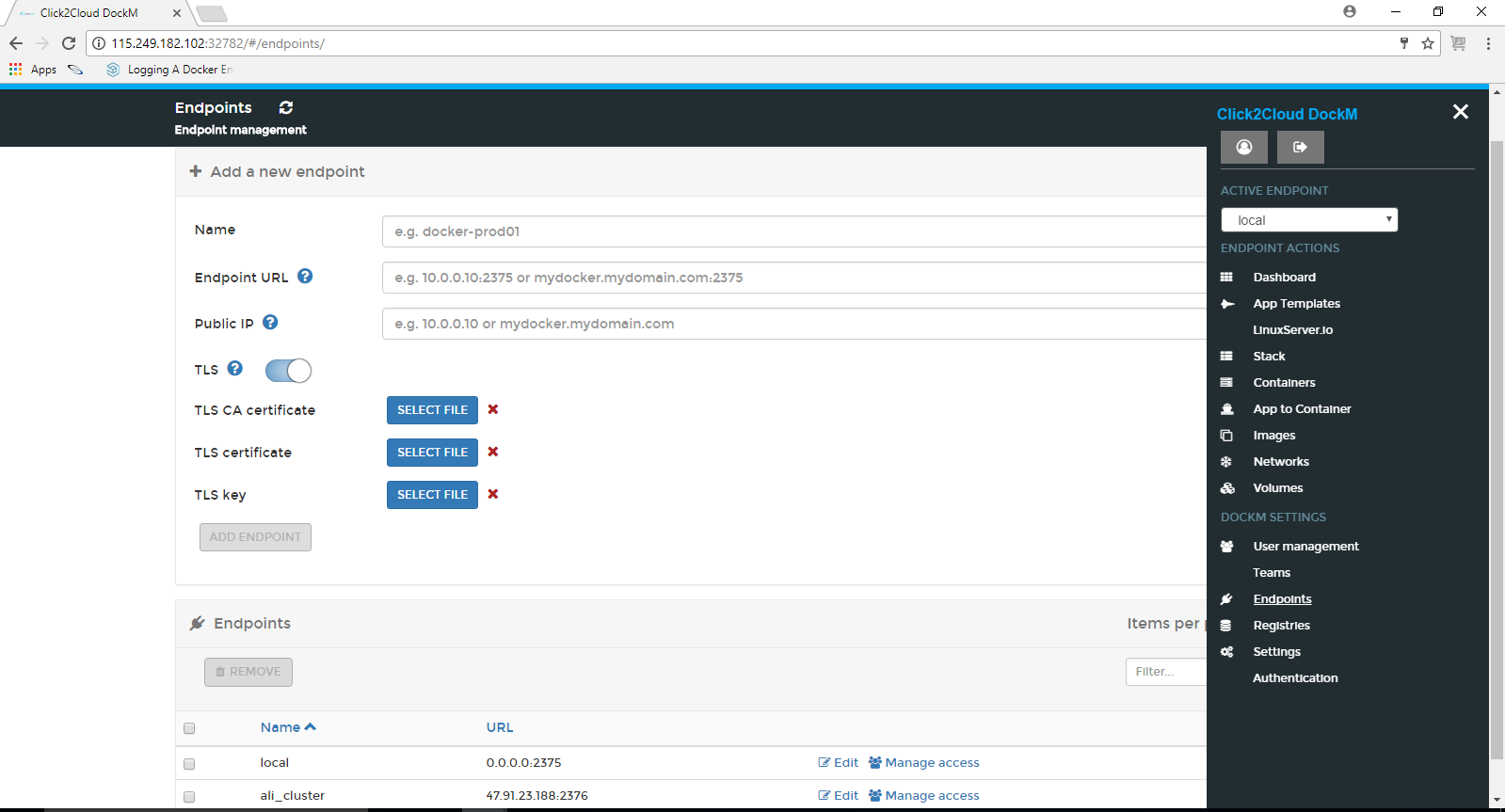


Clicking on Teams tab, shows the Teams page. In this page click on the Edit button of the added team, shows the Team Details page. We can add and remove the Users from the Team. Simply click on a user to move it from one table to the other.

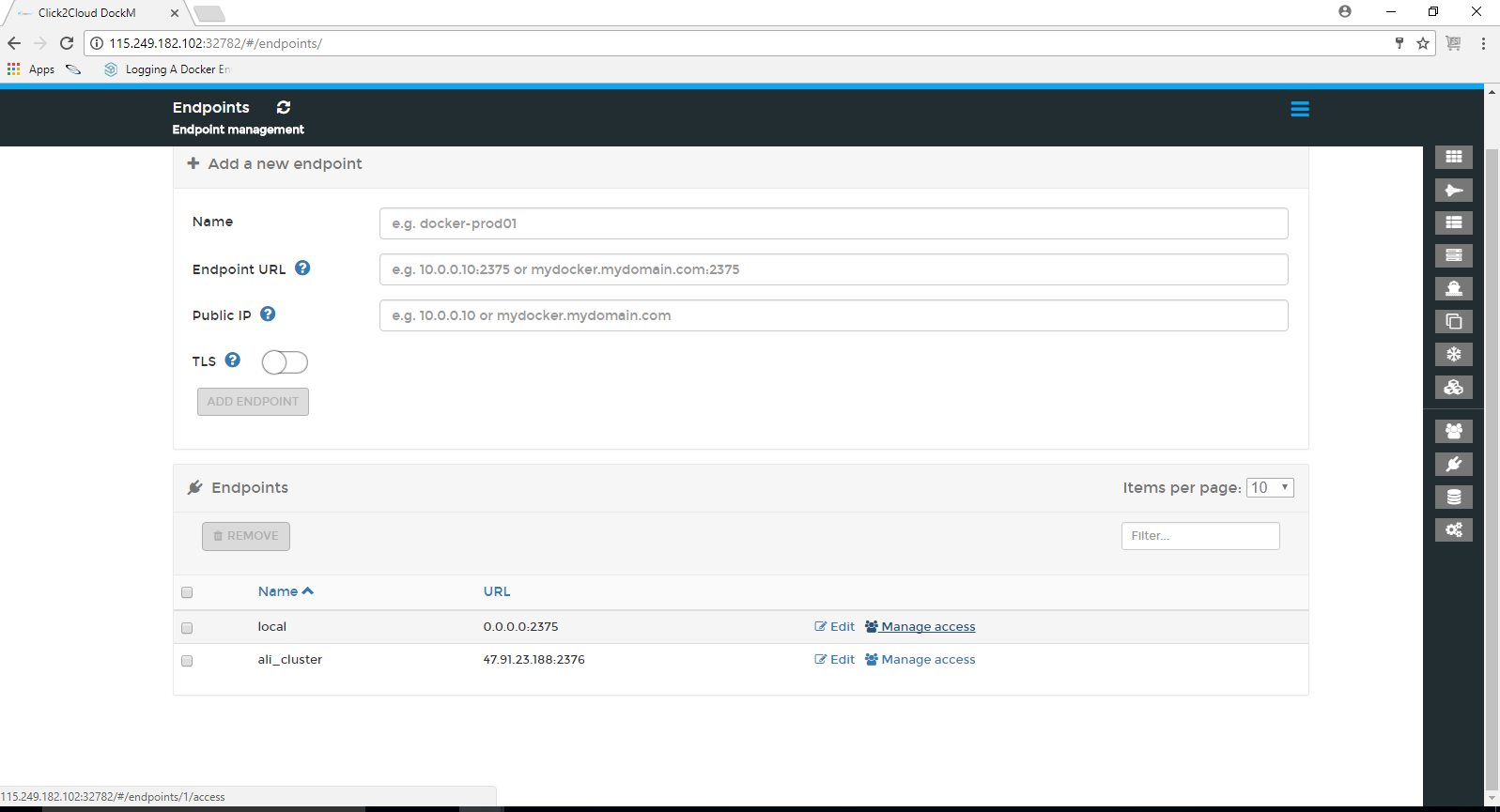


### **1.3.2 Endpoints**

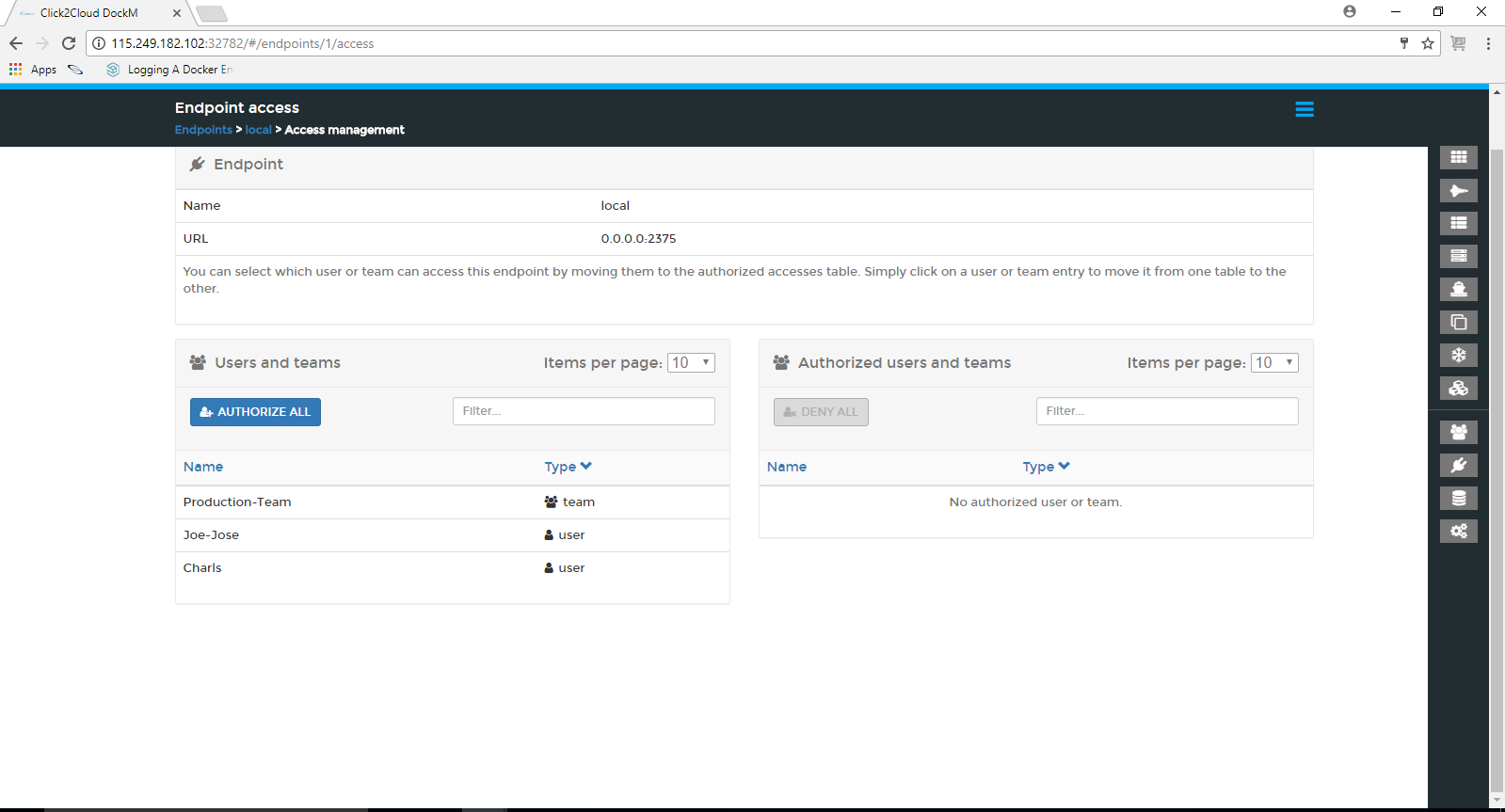
This tab provides the option to create the Endpoints, only Administrator User can create the Endpoints and validate TLS certificates.



By using manage access option Administrator User gives the permission to Users, Team Leaders

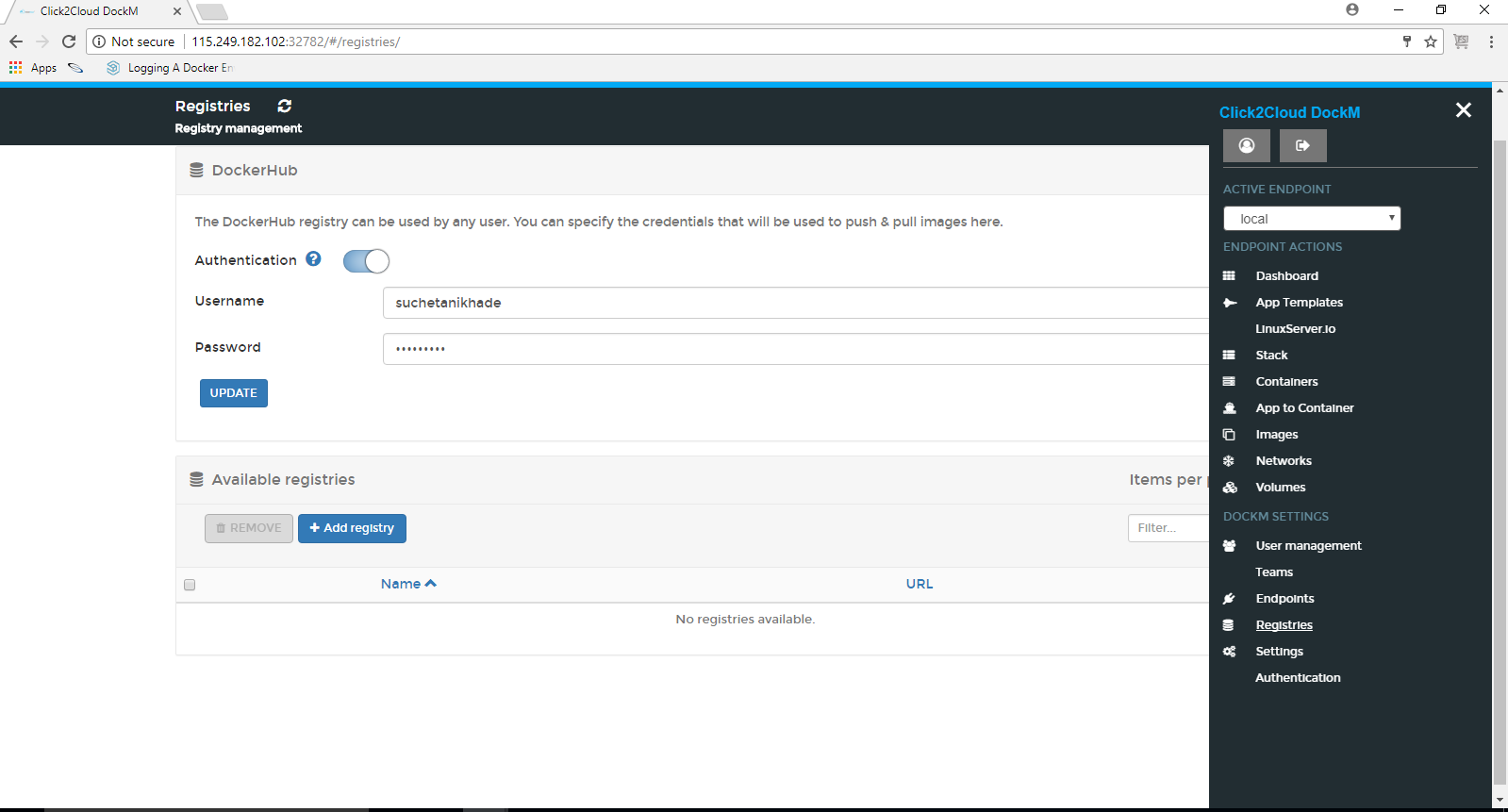


Clicking on Manage Access, shows the Endpoint Access page. In this page, we can select which user or team can access this endpoint by moving them to the authorized access table. Simply click on a user or team entry to move it from one table to the other.

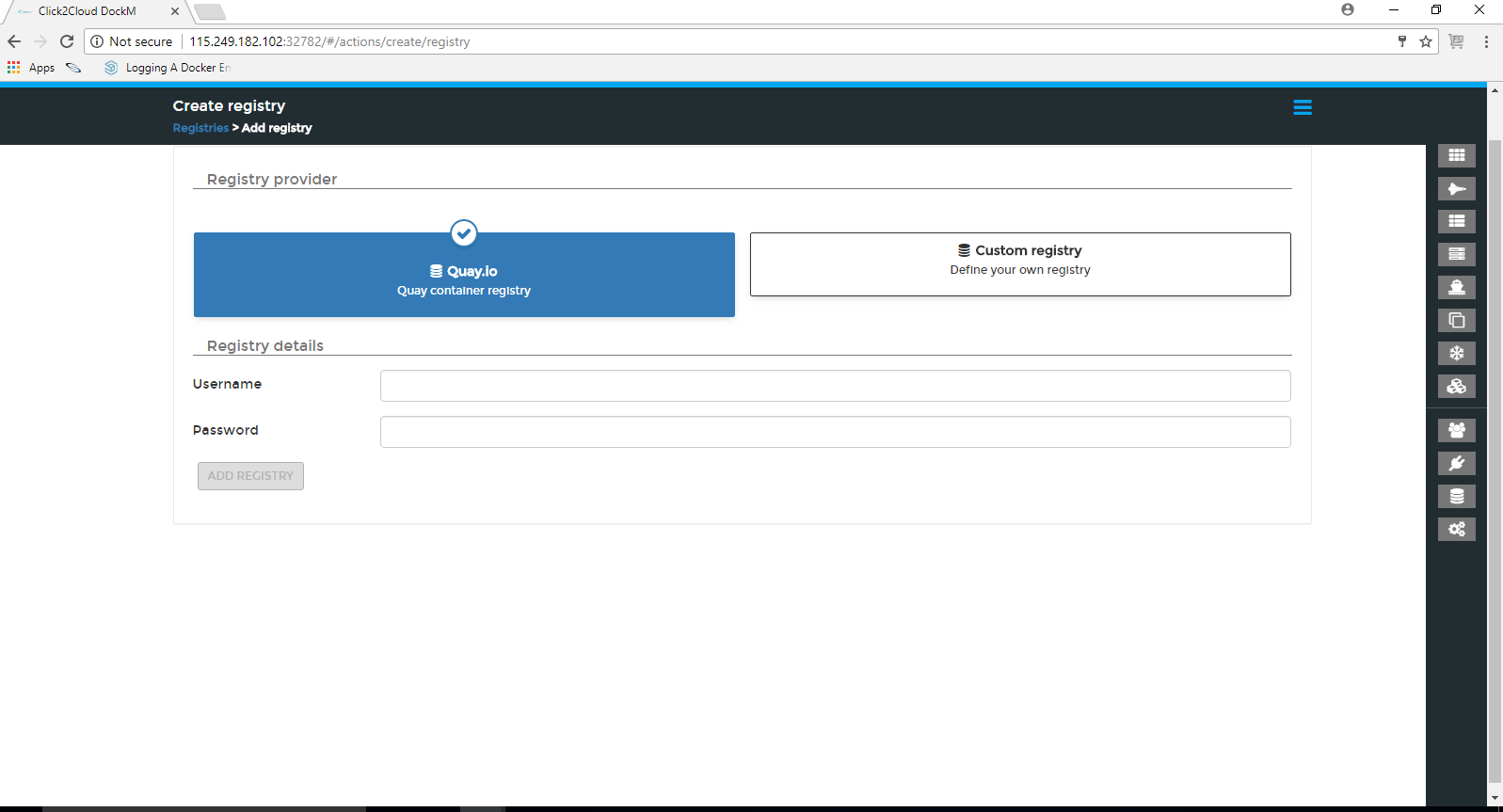


### **1.3.3 Registries**

Registries contains the all available registries. Administrator can Update and add registries.

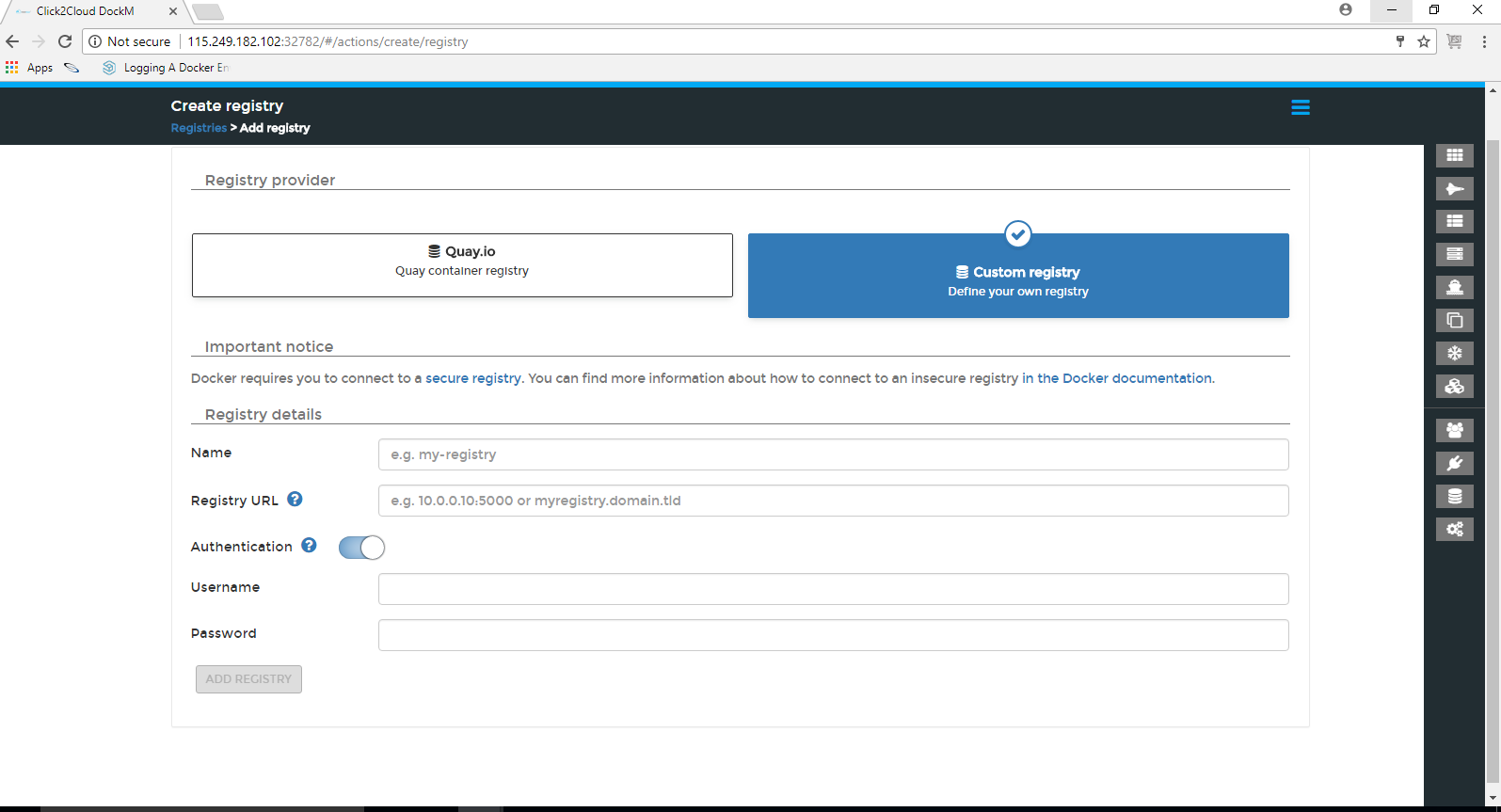


Administrator can create another custom registry. Clicking on Add Registry buttons shows the Create Registry page with the options Quay.io and Custom registry. For adding Quay.io i.e. Quay container registry provide Registry Details as Username and Password.



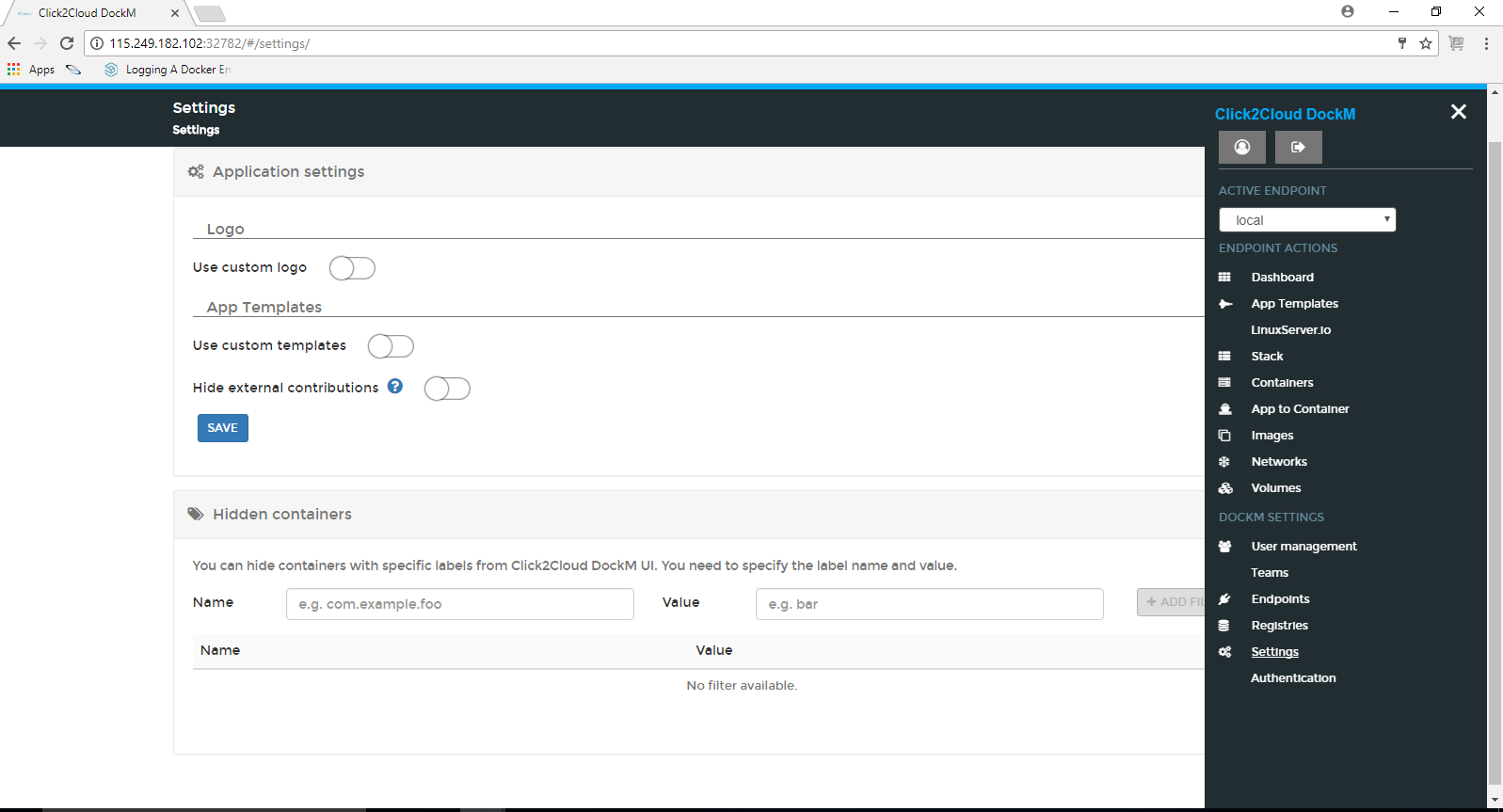
You can create your own custom registry by providing the options for Registry Details as Name, Registry URL and Authentication for Custom registry.

In Authentication tab provide the Username and Password.

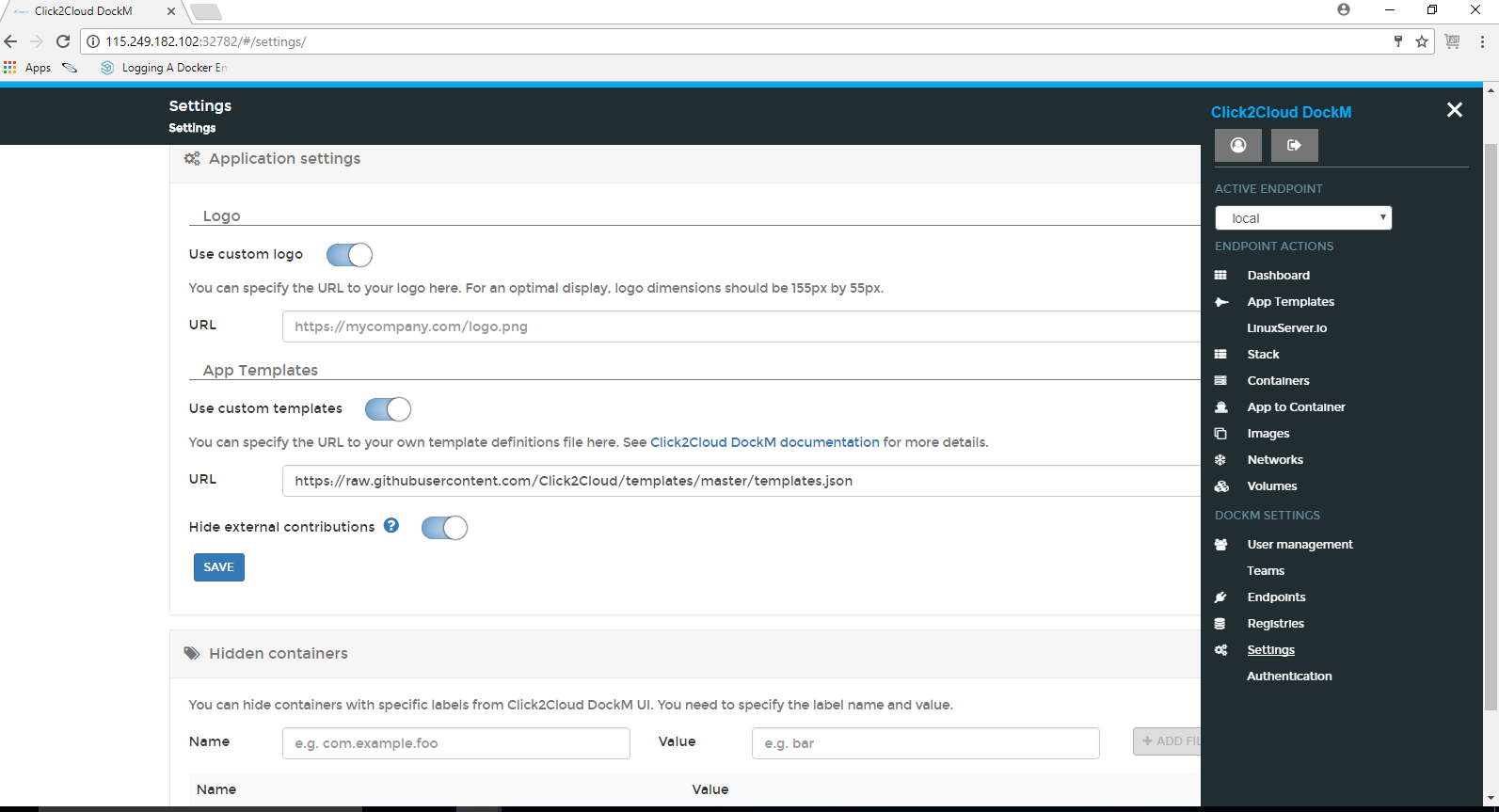


### **1.3.4 Settings**

Settings contains all the Application settings and all hidden container settings



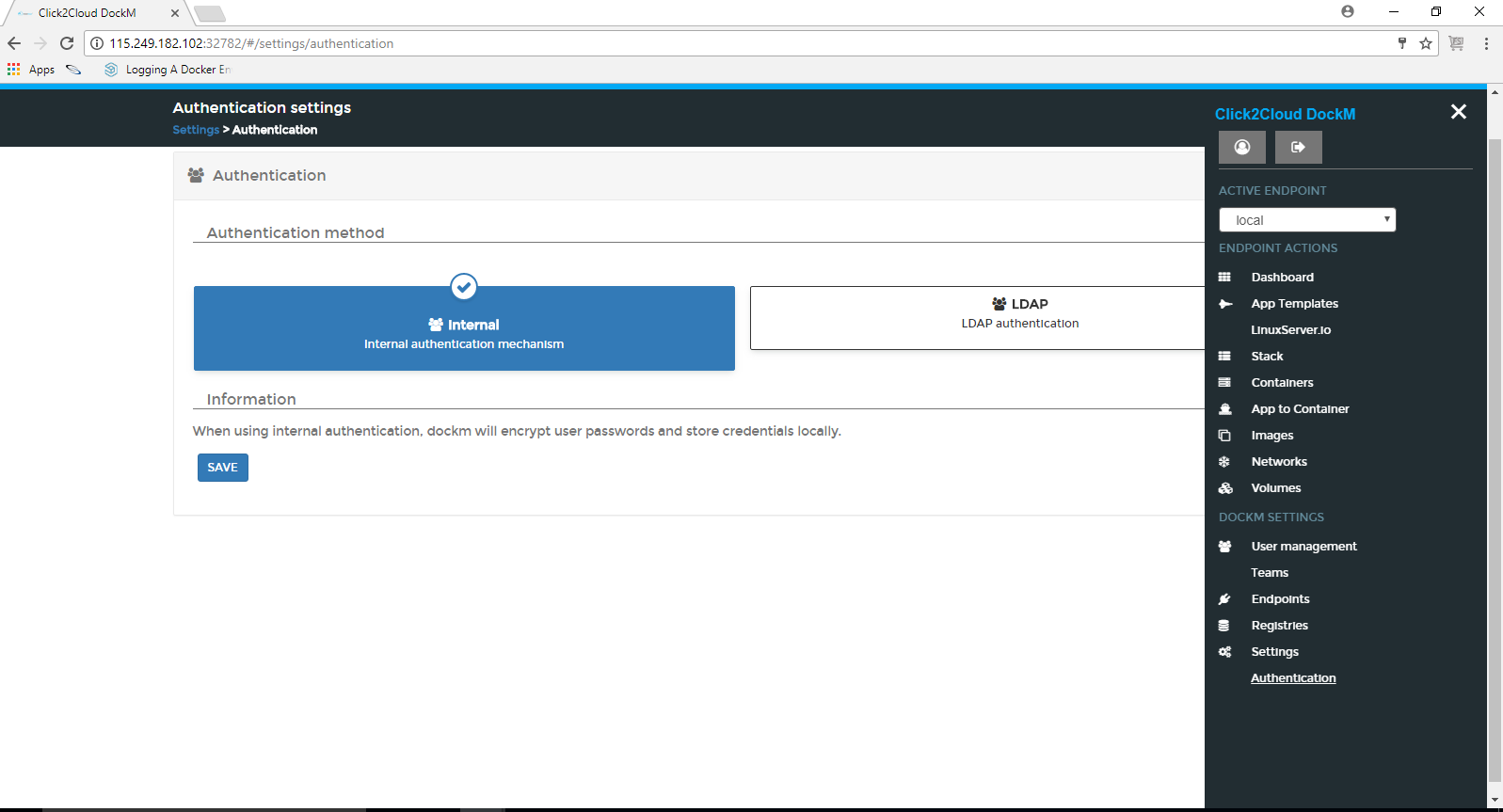
Administrator User can add the Custom Logo by specifying the URL with specified image dimensions. By enabling the Use custom templates option, you can create your own custom templates by specifying the path of JSON file of GitHub repository. If you don’t want external contributions such as Linuxserver.io in the side bar, you can hide it by enabling the Hide external contributions.



#### 1.3.4.1 Authentication

There are two type of Authentication methods one is Internal and second is LDAP authentication.

When you are using Internal authentication DockM will encrypt user passwords and store credentials locally.



When you are using LDAP authentication, DockM will delegate user authentication to a LDAP server. Exception for the admin user that always uses internal authentication.

