Edge Agent Guide

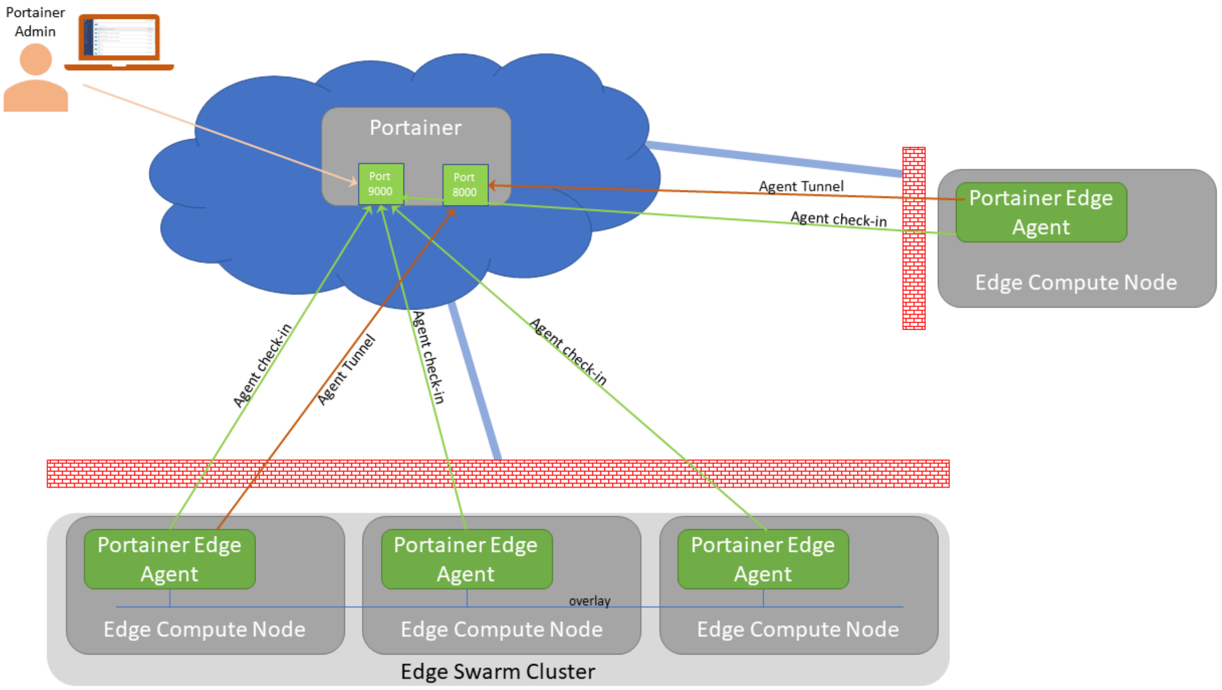
The edge agent was created as a way to manage an edge compute environment where devices typically lack the networking capability to run the traditional Portainer agent.

Introduction to the edge agent

The traditional Portainer agent was built for a standard networking environment with a static IP and port forwarding enabled, such that Portainer can easily connect to an agent. In response to the advanced use cases we are seeing from our users, we re-thought the architecture of the agent.

With the edge agent; the deployment process and the way Portainer and the agent communicate has changed. The agent can now communicate out to the internet and back to Portainer (eliminating the need for a static IP and port forwarding). Some examples where the edge agent can now be used include:

- Behind an internet connection that does not have a fixed public IP, or cannot be port forwarded
- Via a 4G network, with a IP address assigned on the ISP's network and not the internet
- Inside a corporate network with no way to forward a public IP:port to the agent



Overview of the edge agent's architecture (swarm cluster on bottom, standalone right)

Configuring Portainer for the edge agent

Portainer must be updated to version 1.22.0 and have port 8000 exposed (alongside the port used to access the webUI) to use the edge agent. When using an edge agent endpoint, port 8000 is used by an agent to open a tunnel with Portainer.

How to add port 8000 to Portainer running as a standalone container:
To add port 8000 to an existing Portainer container, it must be deleted and redeployed with port 8000 added to the command.

- **Step 1:** Stop your existing Portainer container using the following command (assuming your container is named *portainer)*:
  ```
  docker container stop portainer
  ```

Note: If you have a different name for your container replace 'portainer' with the name you used.

- **Step 2:** Pull the latest *portainer/portainer* image using the following command:

  ```
  docker pull portainer/portainer
  ```

- **Step 3:** Create a new Portainer container using the same command you used to create the previous one with the additional flag -p 8000:8000.

  For example:

  ```
  docker container run -p 9000:9000 -v
  /var/run/docker.sock:/var/run/docker.sock portainer/portainer
  ```

  Becomes:

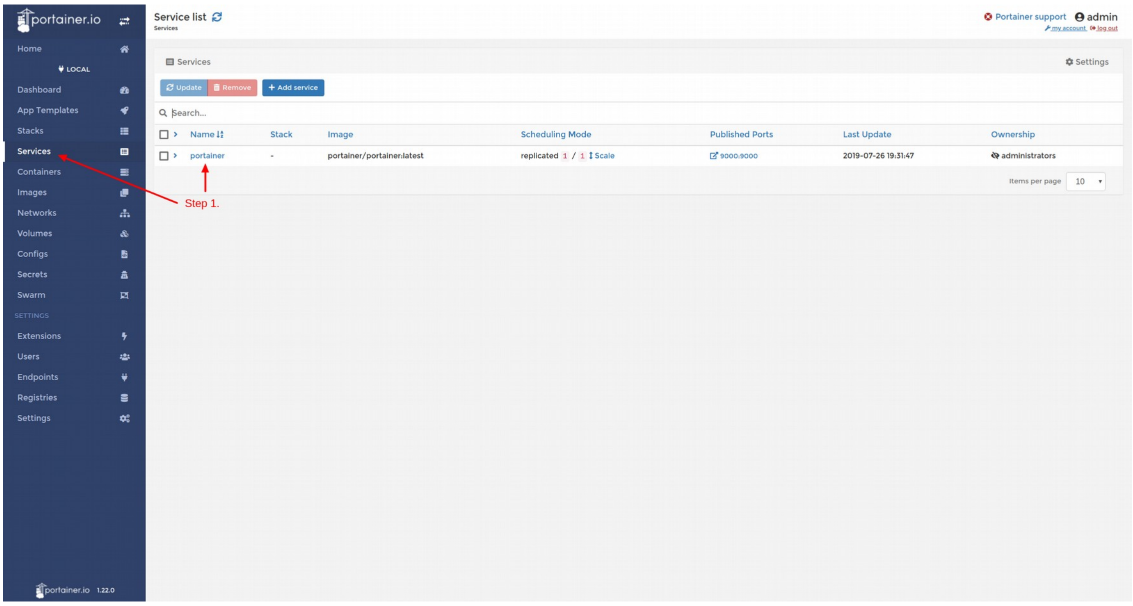
  ```
  docker container run -p 9000:9000 -p 8000:8000 -v
  /var/run/docker.sock:/var/run/docker.sock portainer/portainer
  ```

How to add port 8000 to a Portainer service via the Portainer UI:

Using the Portainer UI, you can update the image the service is using to the latest version and add port 8000 to this service.

Updating the image:

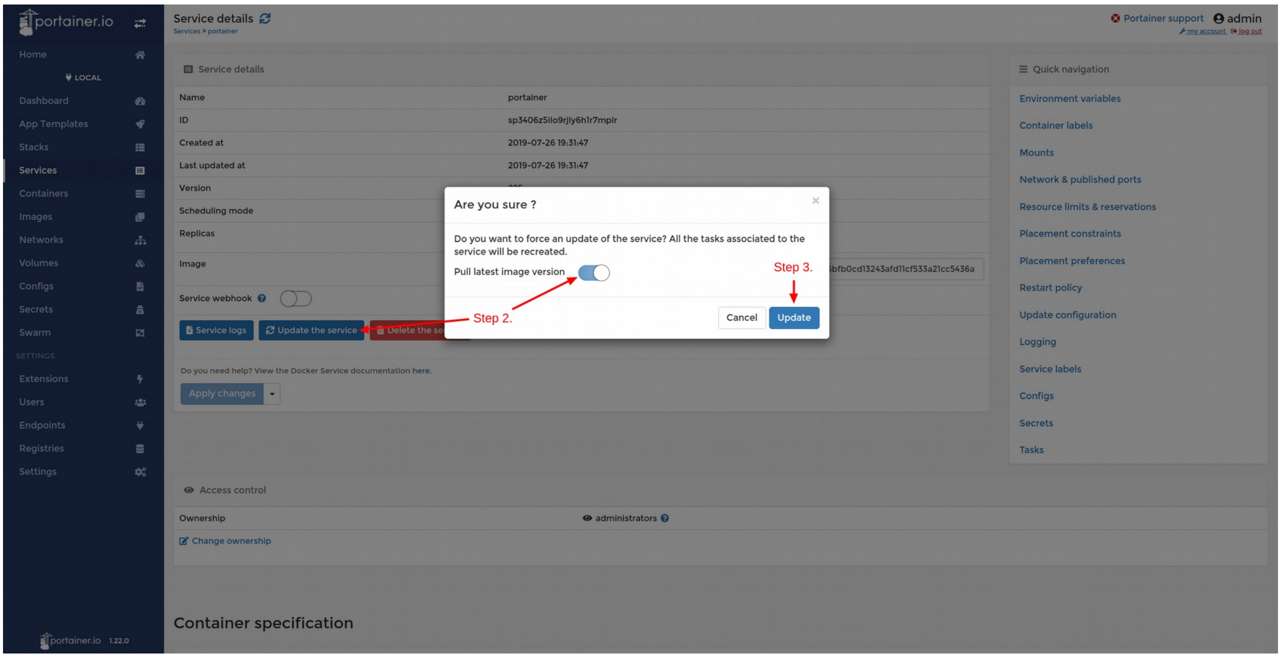
- **Step 1:** Navigate to the Services view and click on your service.



Name	Stack	Image	Scheduling Mode	Published Ports	Last Update	Ownership
portainer	-	portainer/portainer:latest	replicated 1 / 1 Scale	9000:9000	2019-07-26 19:31:47	administrators

Selecting your Portainer service from the services view

- **Step 2:** Click the *Update the service* button, then click the *Pull latest image version* toggle
- **Step 3:** Click the *Update* button to pull the latest Portainer image.



Service details

Services > portainer

Service details	
Name	portainer
ID	sp3406z5ilo9rjly6h1r7mplr
Created at	2019-07-26 19:31:47
Last updated at	2019-07-26 19:31:47
Version	
Scheduling mode	
Replicas	
Image	...bfb0cd13243afd11cf533a21cc5436a

Are you sure ?

Do you want to force an update of the service? All the tasks associated to the service will be recreated.

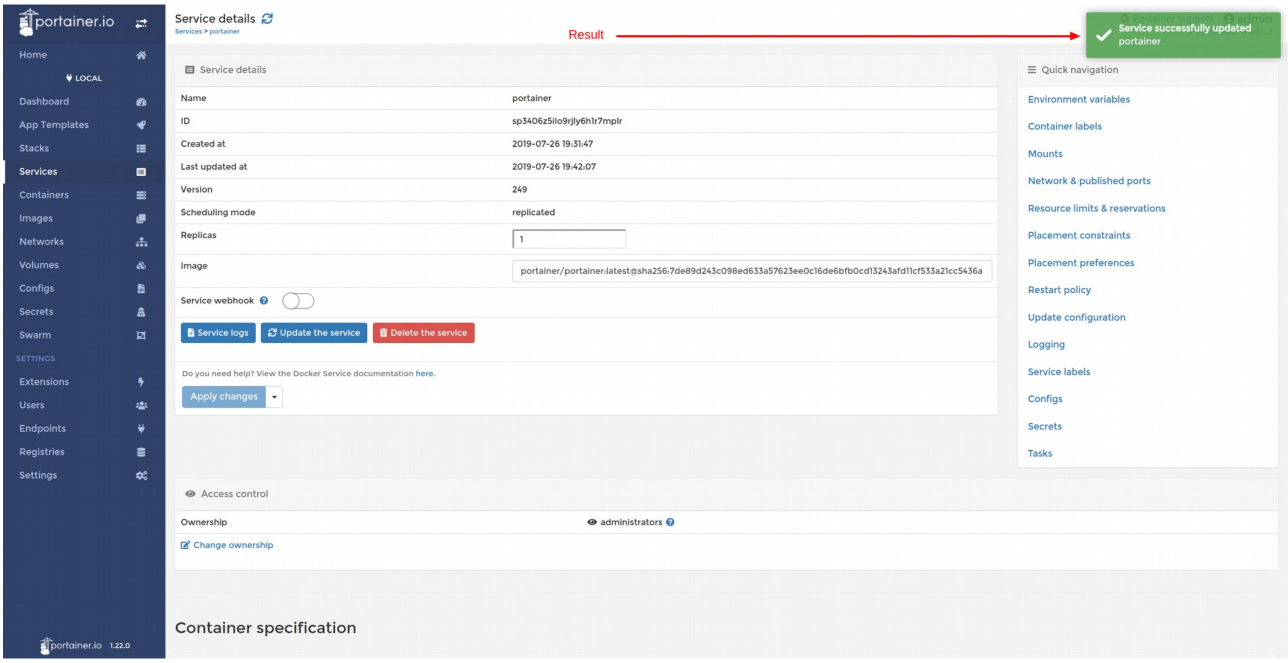
Pull latest image version

Do you need help? View the Docker Service documentation here.

Ownership: administrators

Choosing to update the service with the latest image version

Result: You should see a green notification message appear in the top right of the screen if the service was successfully updated.



Service details

Services > portainer

Service successfully updated
portainer

Service details	
Name	portainer
ID	sp3406z5ilo9rjly6h1r7mplr
Created at	2019-07-26 19:31:47
Last updated at	2019-07-26 19:42:07
Version	249
Scheduling mode	replicated
Replicas	1
Image	portainer/portainer:latest@sha256:7de89d243c098ed633a57623ee0c16de6bfb0cd13243afd11cf533a21cc5436a

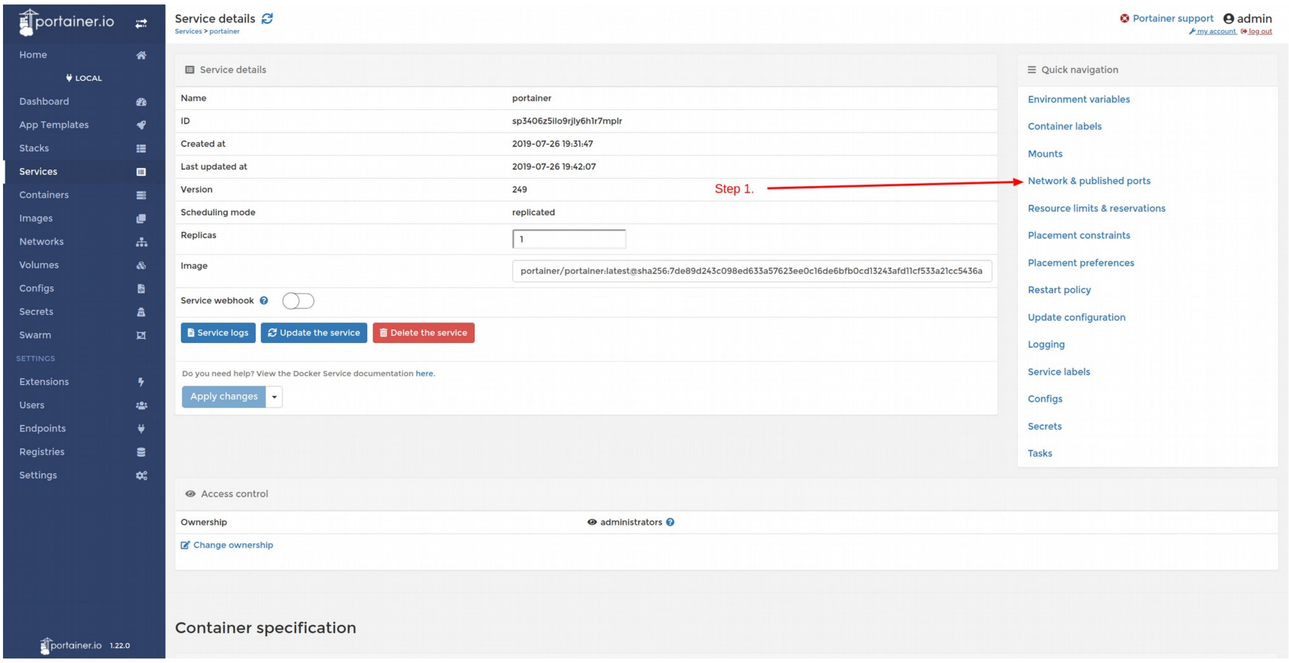
Do you need help? View the Docker Service documentation here.

Ownership: administrators

Service successfully updated with latest image

Adding port 8000:

- **Step 1:** From the Services details view of your service, refresh the page then click on the *Network & published* ports section of your sidebar


Service details

Services > portainer

Name	portainer
ID	sp3406z5ilo9rjly6h1r7mplr
Created at	2019-07-26 19:31:47
Last updated at	2019-07-26 19:42:07
Version	249
Scheduling mode	replicated
Replicas	1
Image	portainer/portainer:latest@sha256:7de89d243c098ed633a57623ee0c16de6bfb0cd13243afd11cf533a21cc5436a
Service webhook	

Do you need help? View the Docker Service documentation here.

Step 1.

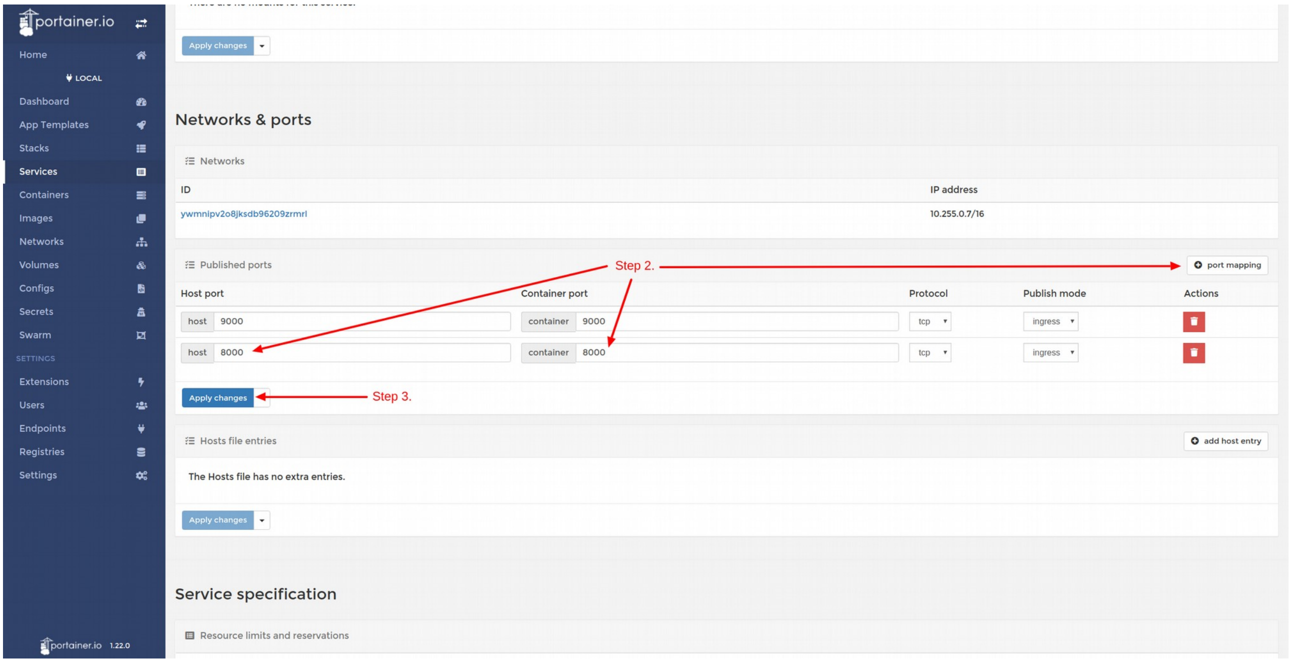
Access control

Ownership	administrators

Container specification

Navigating to the Network & ports section

- **Step 2:** Click on the port mapping button to add a new port entry and fill in 8000 for the host and container ports.
- **Step 3:** Click the *Apply changes* button to apply this port addition to your service.


Networks & ports

Networks

ID	IP address
ywmnipv2o8jksdb96209zrmrl	10.255.0.7/16

Published ports

Host port	Container port	Protocol	Publish mode	Actions
host 9000	container 9000	tcp	ingress	
host 8000	container 8000	tcp	ingress	

Step 2.

Step 3.

Hosts file entries

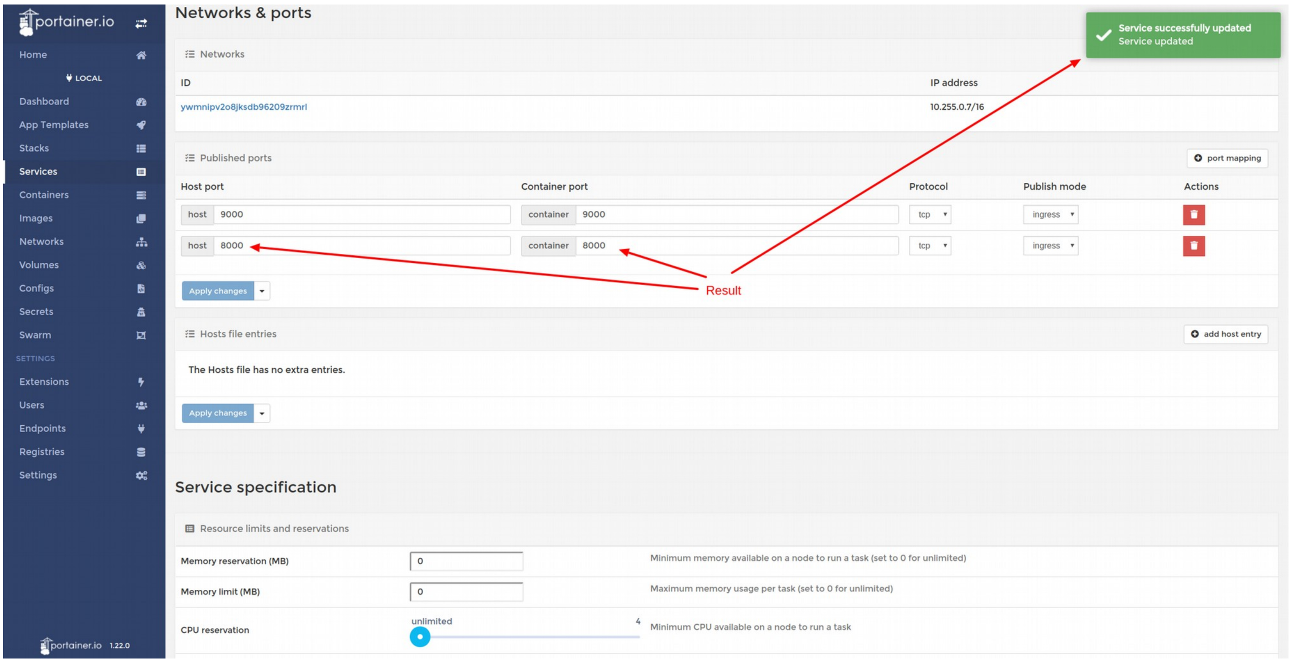
The Hosts file has no extra entries.

Service specification

Resource limits and reservations

Applying port 8000 as an additional port to the Portainer service

Result: You should see a green notification message appear in the top right of the screen if the service was successfully updated with port 8000 as an additional port.



Networks & ports

Service successfully updated
Service updated

Networks

ID	IP address
ywmnipv2o8jksdb96209zrmrl	10.255.0.7/16

Published ports

Host port	Container port	Protocol	Publish mode	Actions
host 9000	container 9000	tcp	ingress	
host 8000	container 8000	tcp	ingress	

Hosts file entries

The Hosts file has no extra entries.

Service specification

Resource limits and reservations

Memory reservation (MB)	0	Minimum memory available on a node to run a task (set to 0 for unlimited)
Memory limit (MB)	0	Maximum memory usage per task (set to 0 for unlimited)
CPU reservation	unlimited 4	Minimum CPU available on a node to run a task

Port 8000 successfully applied to the Portainer service

How to add port 8000 to a Portainer service via CLI:

If you deployed Portainer as a service in a Swarm cluster or as part of a stack, you can issue the following command to update the image (assuming your Docker service is called *portainer*):

```
docker service update --publish-add published=8000,target=8000 --image
portainer/portainer:latest portainer
```

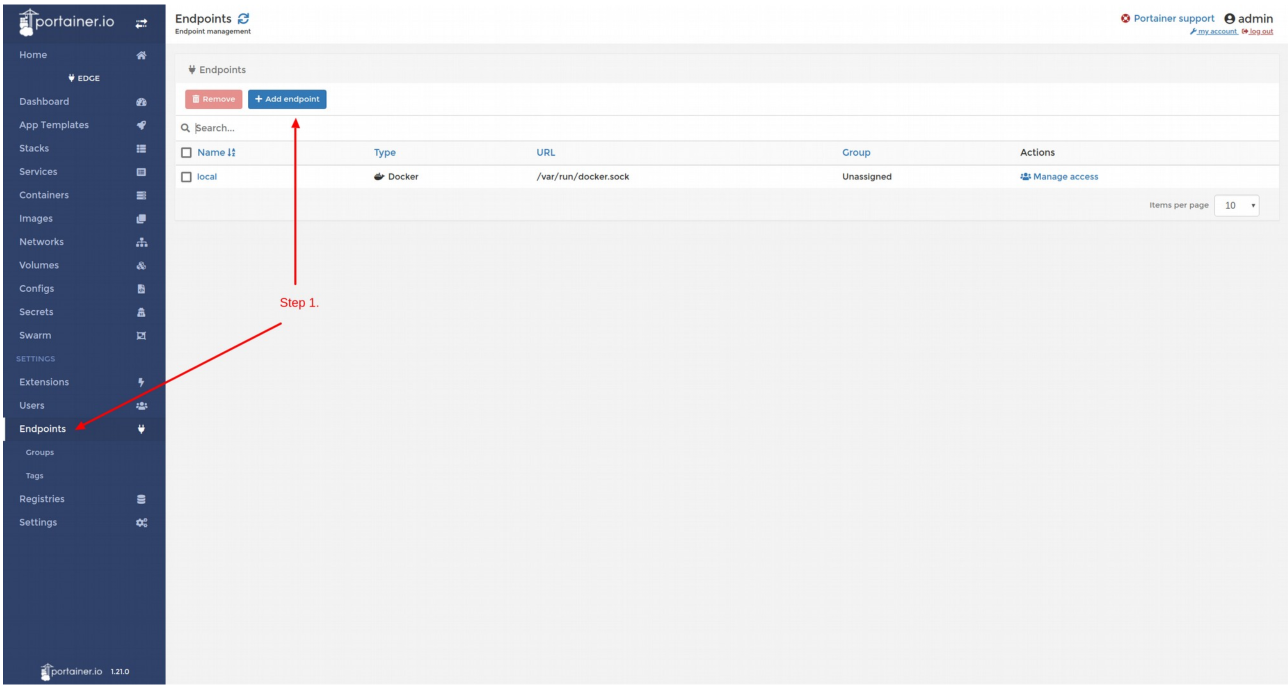
Securing Portainer for use with the agent:

With the introduction of the edge agent, we now recommend to deploy Portainer with TLS. This is to ensure that the Portainer communication with the agent is secured.

- **A less secure way is to use Portainer with self-signed certs:** If you are using a self signed Portainer instance, the edge agent must be deployed with *the flag: -e EDGE_INSECURE_POLL=1*. If you do not deploy Portainer with this flag, then the agent will not be able to communicate with Portainer.
- **Portainer over HTTP:** If Portainer is not configured with either of the above options, it will fallback to using HTTP. This is insecure and NOT recommended as Portainer will respond to agent polling over HTTP. This could potentially expose the encrypted tunnel credentials to a malicious actor.

Add an edge agent endpoint

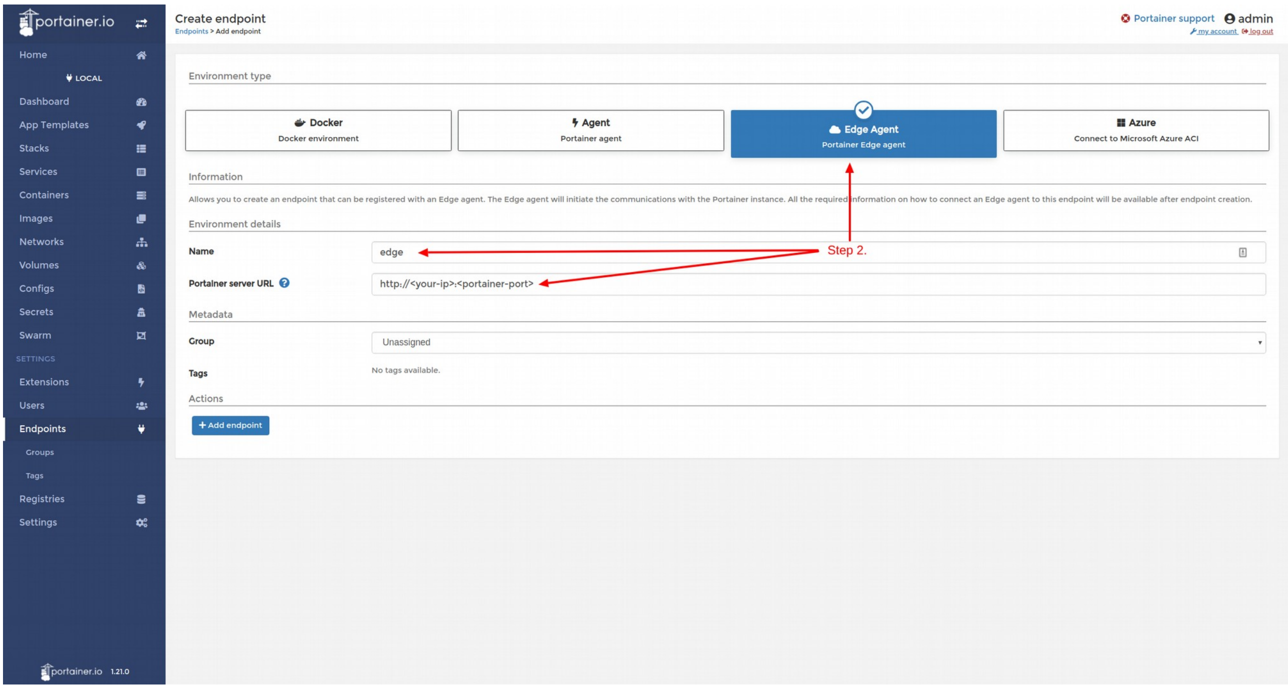
- **Step 1:** Navigate to the Endpoints view and click on the *Add endpoint* button.



☐ Name	Type	URL	Group	Actions
☐ local	Docker	/var/run/docker.sock	Unassigned	Manage access

Adding a new endpoint from the Endpoints view

- **Step 2:** Click on the *edge agent* tab and enter the name and the URL for your Portainer instance (Used by the agent to connect to Portainer)
  Note: Portainer prefils the URL with the domain name of the machine it is on. If it defaults to localhost, set the URL to *http://<your-ip>:<portainer-port>*, (where <portainer-port> is the port you use to access the Portainer UI).



Allows you to create an endpoint that can be registered with an Edge agent. The Edge agent will initiate the communications with the Portainer instance. All the required information on how to connect an Edge agent to this endpoint will be available after endpoint creation.

Name: edge

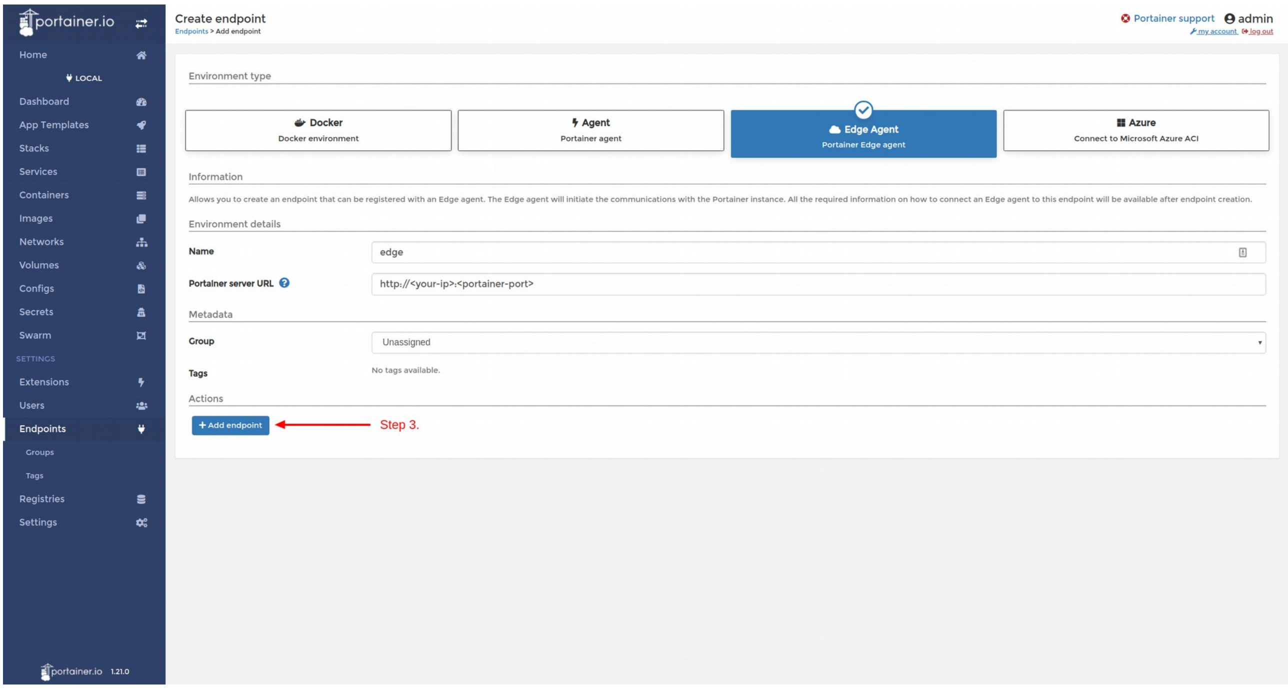
Portainer server URL: http://<your-ip>:<portainer-port>

Group: Unassigned

Tags: No tags available.

Configuring an edge endpoint with name and <IP:Port> of your Portainer instance

- **Step 3:** Click the *Add endpoint* button



Create endpoint
Endpoints > Add endpoint

Environment type: Edge Agent

Information

Allows you to create an endpoint that can be registered with an Edge agent. The Edge agent will initiate the communications with the Portainer instance. All the required information on how to connect an Edge agent to this endpoint will be available after endpoint creation.

Environment details

Name: edge

Portainer server URL: http://<your-ip>:<portainer-port>

Metadata

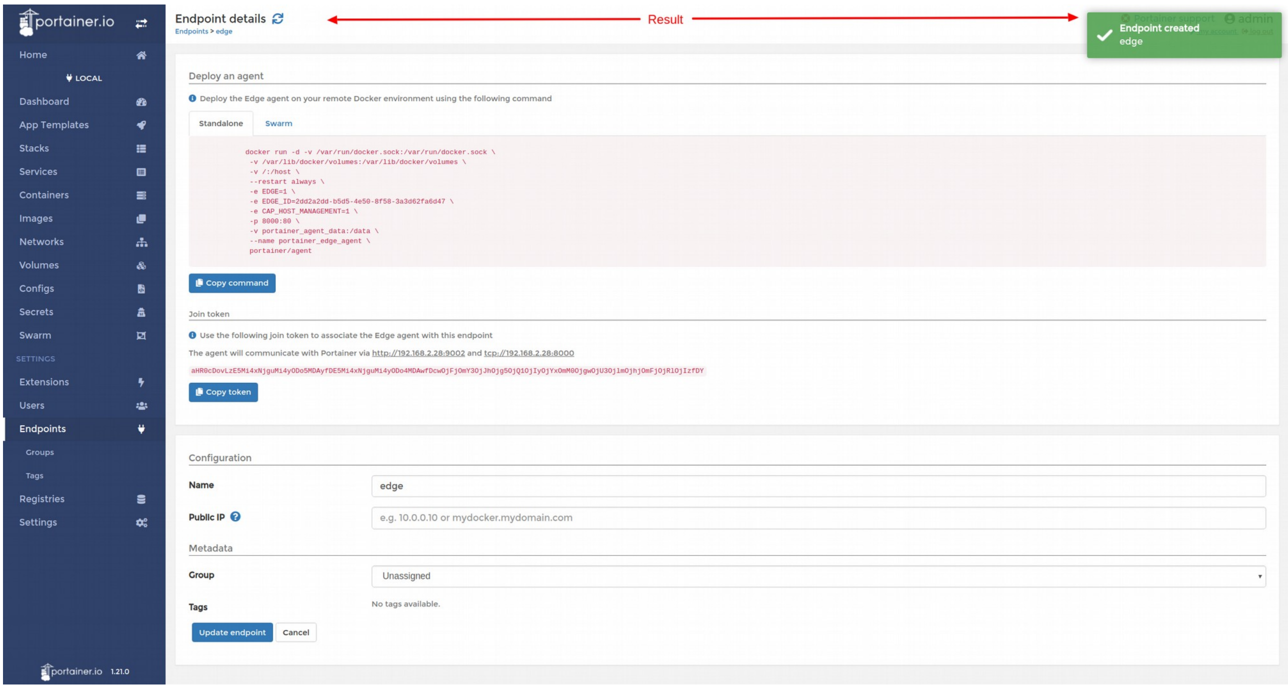
Group: Unassigned

Tags: No tags available.

Actions

Clicking the Add endpoint button

Result: You should see a green notification message appear in the top right of the screen if the endpoint was successfully initialized. You should now also be redirected to the Endpoint details view to configure this endpoint.



Endpoint details
Endpoints > edge

Endpoint created
edge

Deploy an agent

Deploy the Edge agent on your remote Docker environment using the following command

```
docker run -d -v /var/run/docker.sock:/var/run/docker.sock \
 -v /var/lib/docker/volumes:/var/lib/docker/volumes \
 -v /:/host \
 --restart always \
 -e EDGE=1 \
 -e EDGE_ID=2dd2a2dd-b5d5-4e50-8f58-3a3d62fa6d47 \
 -e CAP_HOST_MANAGEMENT=1 \
 -p 8000:80 \
 -v portainer_agent_data:/data \
 --name portainer_edge_agent \
 portainer/agent
```

Join token

Use the following join token to associate the Edge agent with this endpoint

The agent will communicate with Portainer via http://192.168.2.28:9002 and tcp://192.168.2.28:8000

aHR0cDovLzE5Mi4xNjguMi4yODo5MDAyfDE5Mi4xNjguMi4yODo4MDAwfDcwOjFjOmY3OjJhOjg5OjQ1OjIyOjYxOmM0OjgwOjU3OjlmOjhjOmFjOjRlOjIzfDY

Configuration

Name: edge

Public IP: e.g. 10.0.0.10 or mydocker.mydomain.com

Metadata

Group: Unassigned

Tags: No tags available.

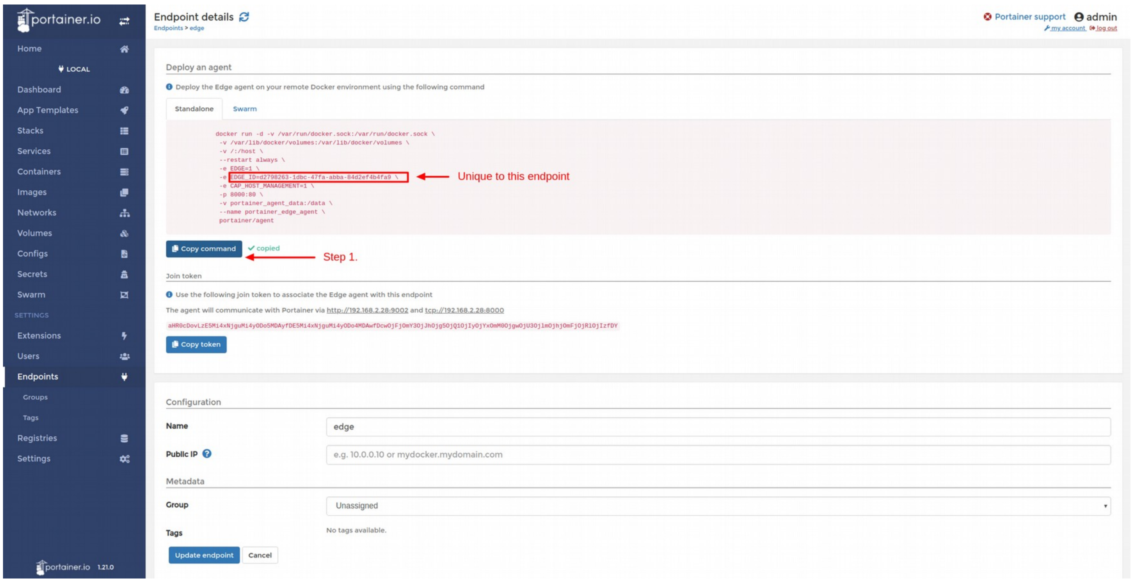
Endpoint successfully added and browser redirect to the endpoint configuration view

Deploy the edge agent

The next step is to deploy the edge agent on your edge environment. You can do this on a standalone environment or as a service across the nodes in a Docker swarm.

Deploy the edge agent in a standalone environment

- **Step 1:** On the standalone tab, click on the *Copy command* button
  Note: This command will contain an EDGE_ID that is unique to the endpoint you created, so don't use the one in this example.



Endpoint details

Endpoints > edge

Deploy an agent

Deploy the Edge agent on your remote Docker environment using the following command

```
docker run -d -v /var/run/docker.sock:/var/run/docker.sock \
 -v /var/lib/docker/volumes:/var/lib/docker/volumes \
 -v /:/host \
 --restart always \
 -e EDGE=1 \
 -e EDGE_ID=d2798263-1dbc-47fa-abba-84d2ef4b4fa9 \
 -e CAP_HOST_MANAGEMENT=1 \
 -p 8000:80 \
 -v portainer_agent_data:/data \
 --name portainer_edge_agent \
 portainer/agent
```

Join token

Use the following join token to associate the Edge agent with this endpoint

The agent will communicate with Portainer via http://192.168.2.28:9002 and tcp://192.168.2.28:8000

```
aHR0cDovLzE5Mi4xNjguMi4yODo5MDAyfDE5Mi4xNjguMi4yODo4MDAwfDcw0jFjOmY30jJh0jg50jQ10jIy0jYxOmM00jgw0jU30jlm0jhjOmFj0jRl0jIzfDY
```

Configuration

Name: edge

Public IP: e.g. 10.0.0.10 or mydocker.mydomain.com

Metadata

Group: Unassigned

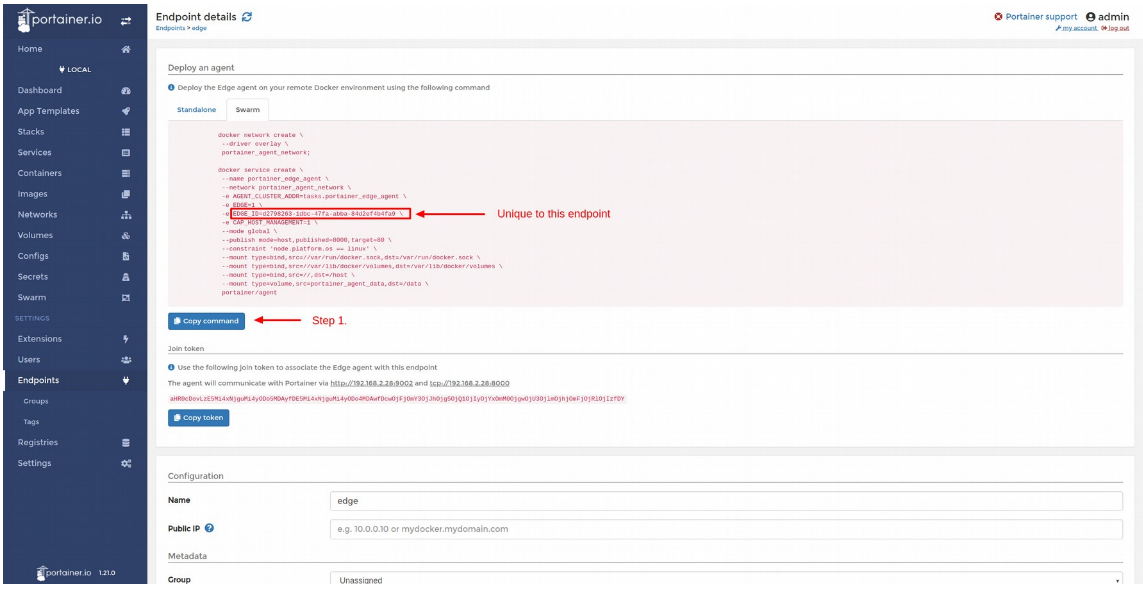
Tags: No tags available.

Click to copy edge agent deployment command for a standalone environment

- **Step 2:** Execute the command on the standalone Docker edge environment.

Deploy the edge agent as a swarm service

- **Step 1:** On the swarm tab, click on the Copy command button.
  Note: This command will contain an EDGE_ID that is unique to the endpoint you created, so don't use the one in this example.



Endpoint details

Endpoints > edge

Deploy an agent

Deploy the Edge agent on your remote Docker environment using the following command

```
docker network create \
 --driver overlay \
 portainer_agent_network;

docker service create \
 --name portainer_edge_agent \
 --network portainer_agent_network \
 -e AGENT_CLUSTER_ADDR=tasks.portainer_edge_agent \
 -e EDGE=1 \
 -e EDGE_ID=d2798263-1dbc-47fa-abba-84d2ef4b4fa9 \
 -e CAP_HOST_MANAGEMENT=1 \
 --mode global \
 --publish mode=host,published=8000,target=80 \
 --constraint 'node.platform.os == linux' \
 --mount type=bind,src=//var/run/docker.sock,dst=/var/run/docker.sock \
 --mount type=bind,src=//var/lib/docker/volumes,dst=/var/lib/docker/volumes \
 --mount type=bind,src=//,dst=/host \
 --mount type=volume,src=portainer_agent_data,dst=/data \
 portainer/agent
```

Join token

Use the following join token to associate the Edge agent with this endpoint

The agent will communicate with Portainer via http://192.168.2.28:9002 and tcp://192.168.2.28:8000

```
aHR0cDovLzE5Mi4xNjguMi4yODo5MDAyfDE5Mi4xNjguMi4yODo4MDAwfDcw0jFjOmY30jJh0jg50jQ10jIy0jYxOmM00jgw0jU30jlm0jhjOmFj0jRl0jIzfDY
```

Configuration

Name: edge

Public IP: e.g. 10.0.0.10 or mydocker.mydomain.com

Metadata

Group: Unassigned

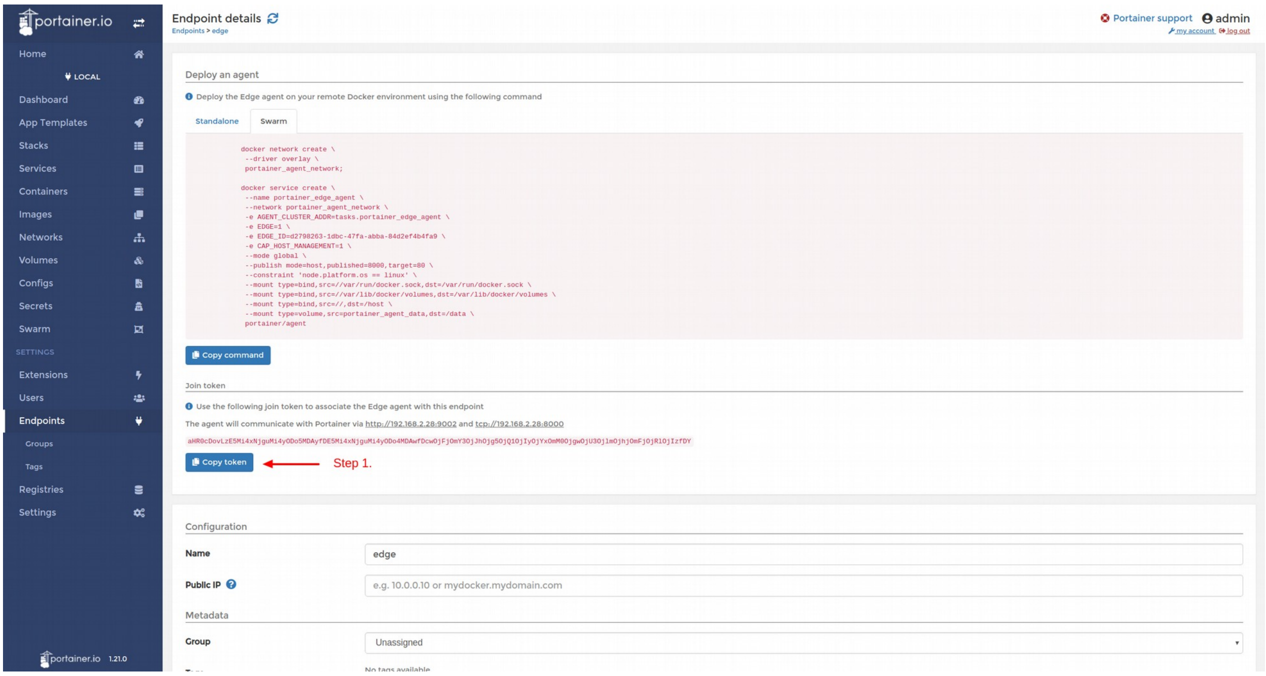
Copying the deployment command for a swarm environment

- **Step 2:** Execute the command on a manager node in the edge environment.

Pair the edge agent with Portainer

The final step of setting up the edge agent is to pair it to Portainer using the join token that Portainer provides in the edge endpoint details.

- **Step 1:** In the Endpoint details view of your endpoint, click the *Copy token* button.



Deploy an agent

Deploy the Edge agent on your remote Docker environment using the following command

```
docker network create \
 --driver overlay \
 portainer_agent_network;

docker service create \
 --name portainer_edge_agent \
 --network portainer_agent_network \
 -e AGENT_CLUSTER_ADDR=tasks.portainer_edge_agent \
 -e EDGE=1 \
 -e EDGE_ID=d2798263-1dbc-47fa-abba-84d2ef4b4fa9 \
 -e CAP_HOST_MANAGEMENT=1 \
 --mode global \
 --publish mode=host,published=8000,target=80 \
 --constraint 'node.platform.os == linux' \
 --mount type=bind,src=//var/run/docker.sock,dst=/var/run/docker.sock \
 --mount type=bind,src=//var/lib/docker/volumes,dst=/var/lib/docker/volumes \
 --mount type=bind,src=//,dst=/host \
 --mount type=volume,src=portainer_agent_data,dst=/data \
 portainer/agent
```

Join token

Use the following join token to associate the Edge agent with this endpoint

The agent will communicate with Portainer via http://192.168.2.28:9002 and tcp://192.168.2.28:8000

```
aHR0cDovLzE5Mi4xNjguMi4yODo5MDAyfDE5Mi4xNjguMi4yODo4MDAwfDcwOjFjOmY3OjJhOjg5OjQ1OjIyOjYxOmM0OjgwOjU3OjlmOjhjOmFjOjRlOjIzfDY
```

Configuration

Name: edge

Public IP: e.g. 10.0.0.10 or mydocker.mydomain.com

Metadata

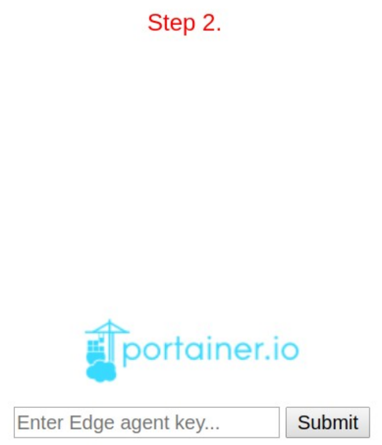
Group: Unassigned

No tags available

Copying the join token from Portainer

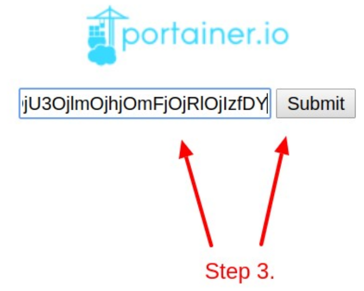
Note: If you have navigated off the page, you can click on the unassociated endpoint in the home view and you will be redirected to the endpoint details page. Alternatively you can click on the endpoint in the endpoints view.

- **Step 2:** Navigate to <edge-ip>:8000 in a different tab in your browser (replacing <edge-ip> with the IP of your edge environment). You should see the Agent Pairing screen with the Portainer.io logo, a textbox and the submit button.



Edge agent waiting for the join token to be entered

- **Step 3:** Paste the join token in the text box and then click the *Submit* button.



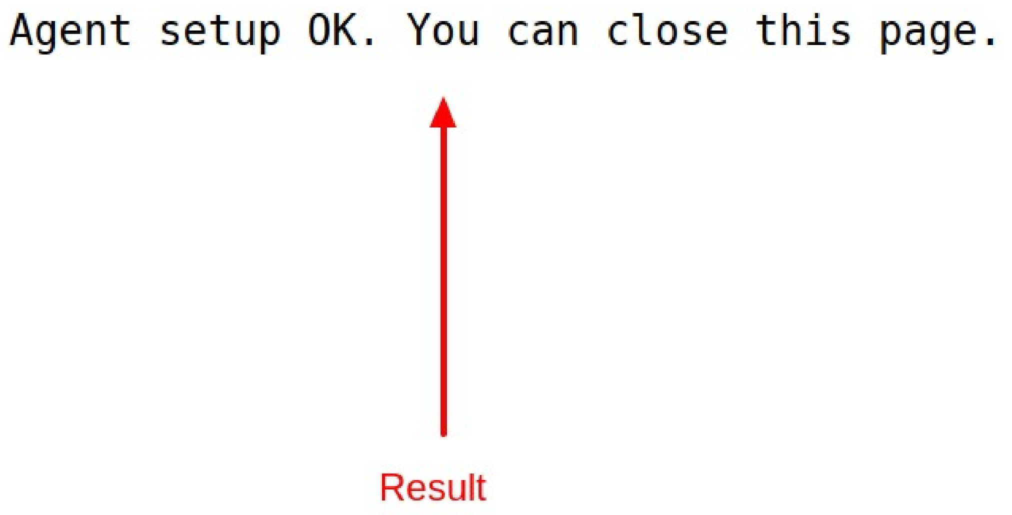
jU3OjlmOjhjOmFjOjRlOjIzfDY

Step 3.

Submitting the join token to pair the edge agent with Portainer

Note: If you do not enter the join token within 15minutes, this page will go offline for security reasons. To get it back, simply restart the edge agent service.

Result: You should now be redirected to <edge-ip>:8000/init and see confirmation text that the agent has successfully been set up. It will now initiate a connection with your Portainer instance.

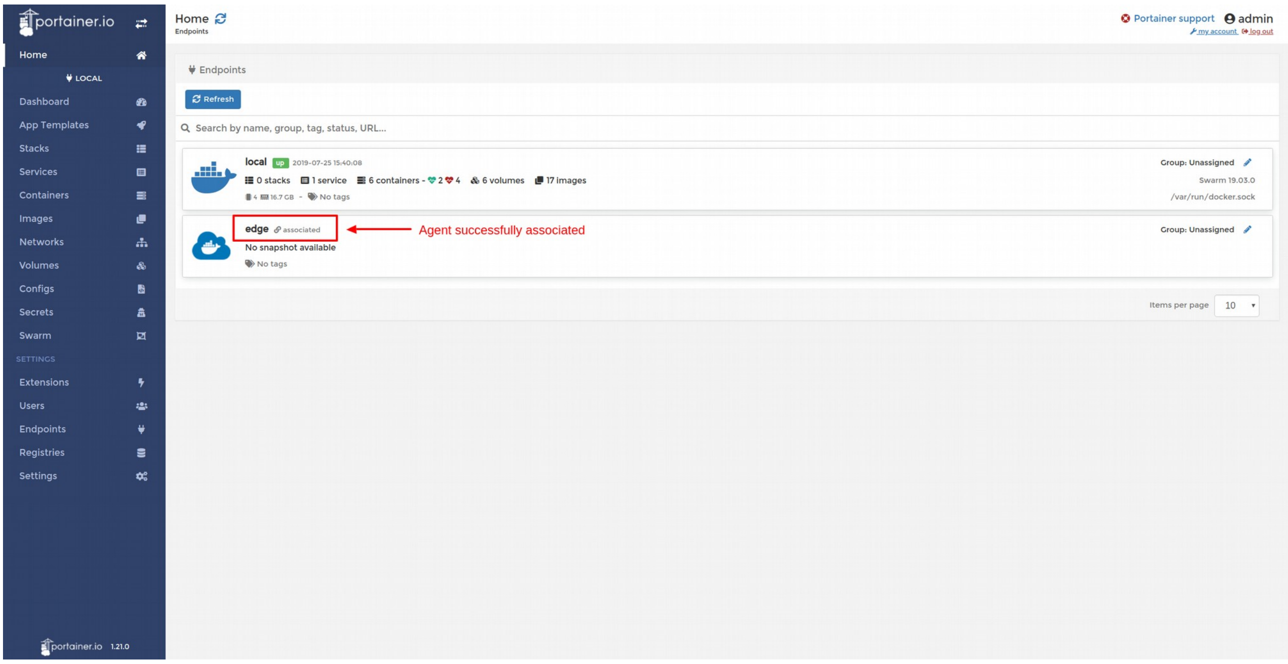


```
Agent setup OK. You can close this page.
```

Result

Agent has successfully saved the token and completed setup

You should now also see in the home view that the agent has been successfully associated with your endpoint.



local up 2019-07-25 15:40:08 — Group: Unassigned
0 stacks 1 service 6 containers - 2 4 6 volumes 17 images — Swarm 19.03.0
4 16.7 GB - No tags — /var/run/docker.sock

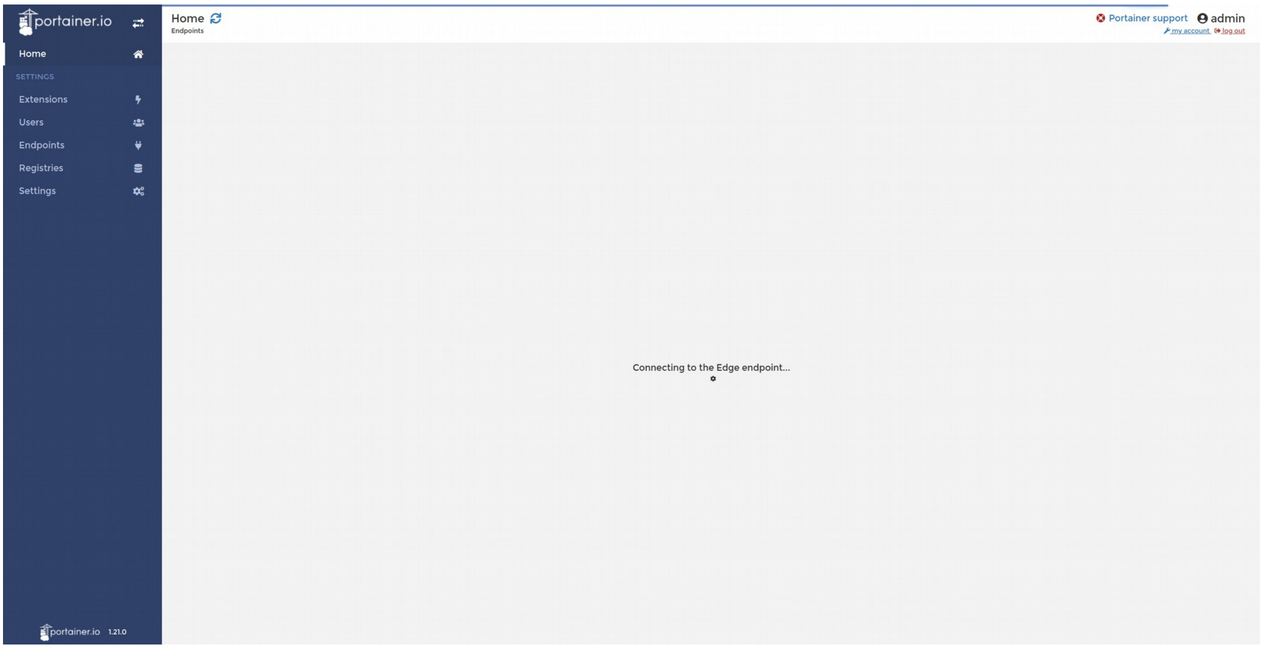
edge associated — Group: Unassigned
No snapshot available
No tags

Agent successfully associated

The agent was successfully associated to the edge endpoint

Using the Agent endpoint

Once you have successfully associated the edge agent to Portainer, you can click on the endpoint to initiate the connection to the edge environment.

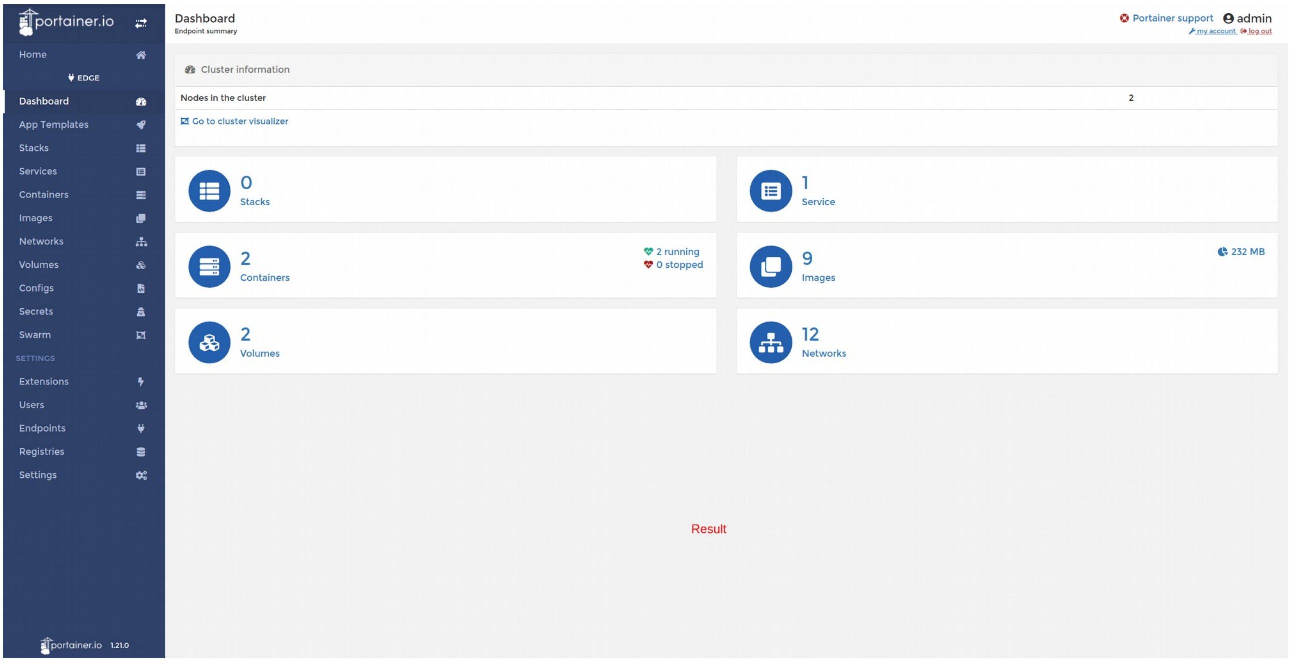


Connecting to the Edge endpoint...

Connecting to the edge environment

Note: Depending on the edge agent poll frequency applied in the Portainer settings, this can take between 10 seconds to a minute (refer to the next section for more info on why).

Result: You should now be redirected to the dashboard view for the edge endpoint once the connection is established. You can now manage the endpoint as you would for any other endpoint.



Nodes in the cluster 2

Successfully connected to the edge endpoint

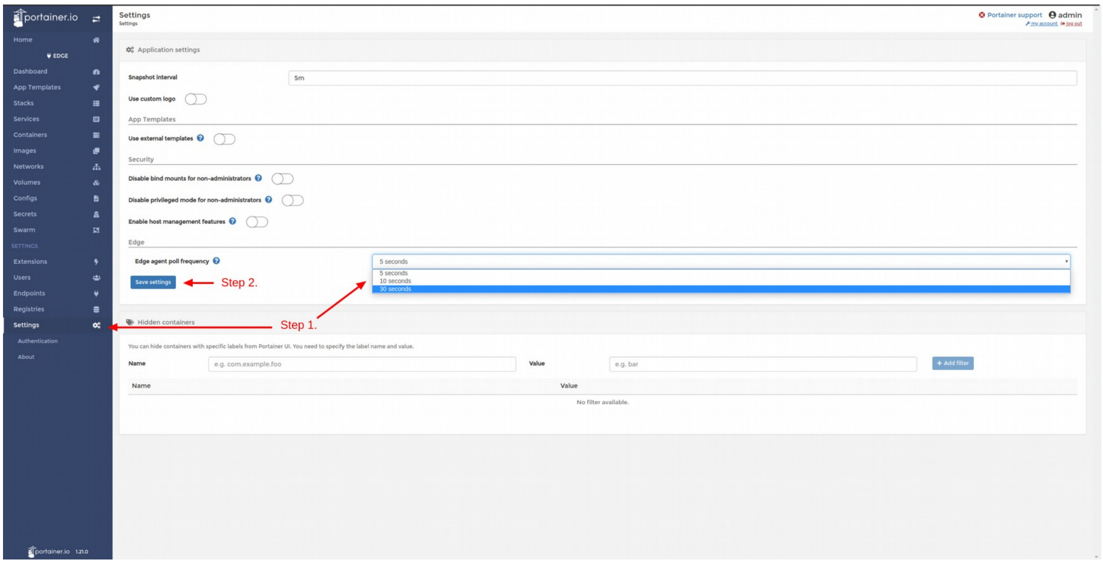
The edge agent poll frequency

The edge agent has something called a poll frequency, this determines how often the edge agent checks-in with Portainer. In a check-in, the edge agent will check to see if it is needed or for changes to endpoint config. The poll frequency will also affect the time it takes to connect to the agent endpoint at the start of a management session.

The connection time will take twice as long as the poll frequency that you have set i.e if it is set to 30seconds Portainer will have to wait 1 minute for the tunnel to be established.

How to change the edge agent poll frequency

- **Step 1:** Navigate to the settings view, then select a poll frequency from the *Edge agent poll frequency* drop-down (default is 5 seconds)
- **Step 2:** Click the *Save settings* button to apply.



Settings
Settings

Application settings

Snapshot interval: 5m

Use custom logo ○

App Templates

Use external templates ○

Security

Disable bind mounts for non-administrators ○

Disable privileged mode for non-administrators ○

Enable host management features ○

Edge

Edge agent poll frequency: 5 seconds
5 seconds
10 seconds
30 seconds

Hidden containers

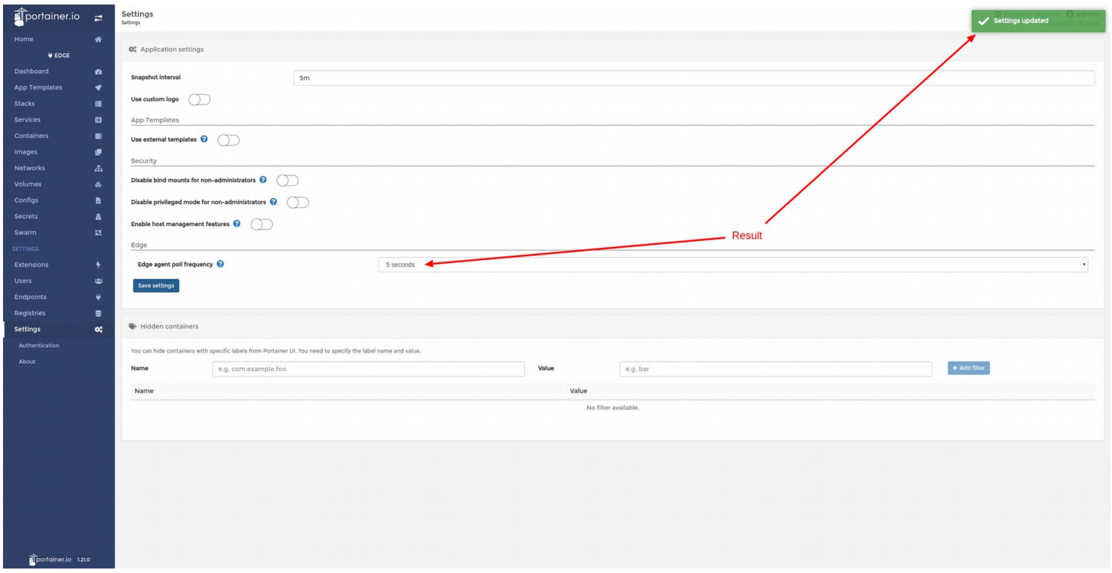
You can hide containers with specific labels from Portainer UI. You need to specify the label name and value.

Name: e.g. com.example.foo Value: e.g. bar

Name	Value
No filter available.	

Configuring the edge agent poll frequency

Result: You should see the page refresh and a green notification message appear in the top right of the screen if *Edge agent poll frequency* was successfully updated. You should now also see that the option you selected has been saved after the page refreshed.



Settings
Settings

Settings updated

Application settings

Snapshot interval: 5m

Use custom logo ○

App Templates

Use external templates ○

Security

Disable bind mounts for non-administrators ○

Disable privileged mode for non-administrators ○

Enable host management features ○

Edge

Edge agent poll frequency: 5 seconds

Hidden containers

You can hide containers with specific labels from Portainer UI. You need to specify the label name and value.

Name: e.g. com.example.foo Value: e.g. bar

Name	Value
No filter available.	

Edge agent poll frequency successfully updated