



docker

4장

GUI로 Docker 운영하기

Docker를 GUI로
관리하기

Docker를 GUI로 관리하기

- docker 관리를 위한 container를 실행하여 GUI로 관리하기
 - **docker volume create** portainer_data
 - **docker run -d -p 9000:9000 -v /var/run/docker.sock:/var/run/docker.sock -v portainer_data:/data portainer/portainer-ce**
 - 또는 **curl http://down.cloudshell.kr/docker/portainer.sh | bash**
 - 웹브라우저에서
 - **http://10.0.2.250:9000**
 - 이제 **GUI를 통하여 docker를 관리한다**
 - ID와 암호를 기억해 둔다



portainer.io

▼ New Portainer installation

Please create the initial administrator user.

Username **admin**

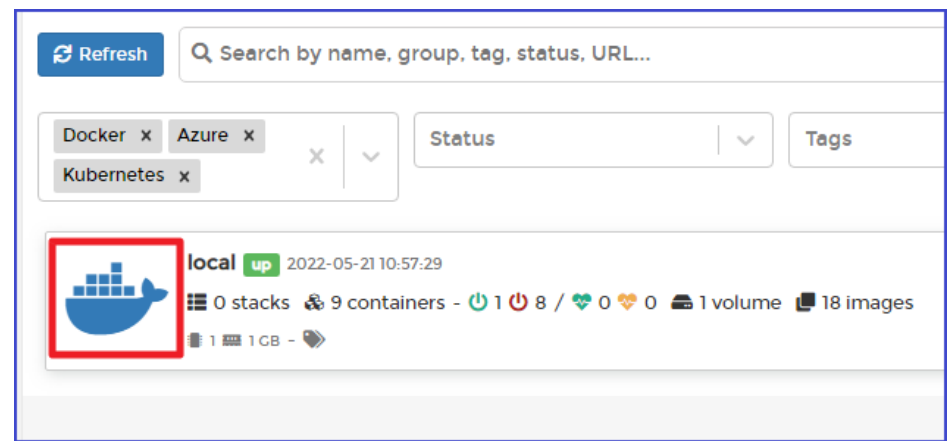
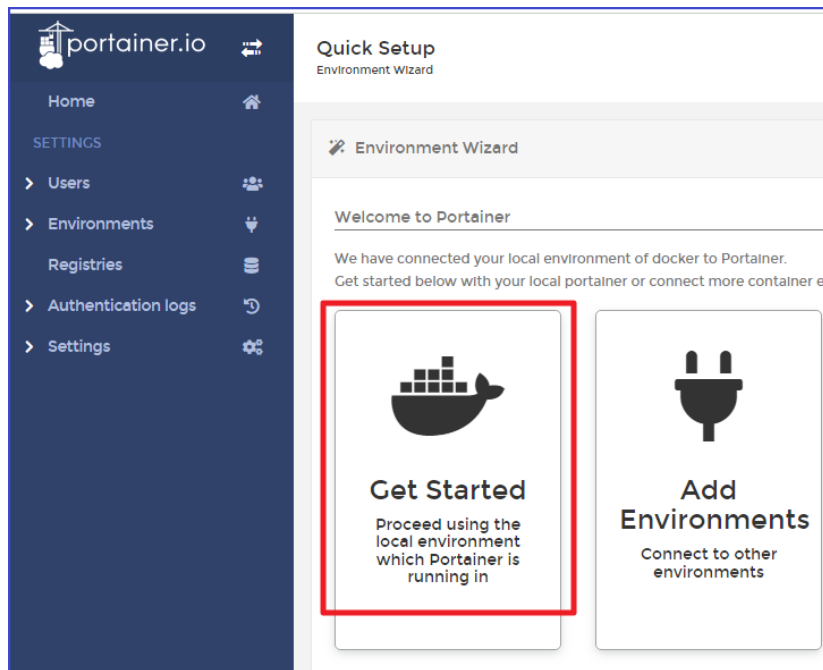
Password **P@ssw0rd1234**

Confirm password **P@ssw0rd1234** ✓

☒ Allow collection of anonymous statistics. You can find more information about this in our [privacy policy](#).

Docker를 GUI로 관리하기

- docker 관리를 위한 container를 실행하여 GUI로 관리하기



Docker를 GUI로 관리하기

- docker 관리를 위한 container를 실행하여 GUI로 관리하기

The screenshot displays the Portainer.io web interface. On the left is a dark blue sidebar with navigation links: Home, LOCAL, Dashboard (selected), App Templates, Stacks, Containers, Images, Networks, Volumes, Events, Host, SETTINGS, Users, Environments, Registries, Authentication logs, and Settings. The main content area is titled 'Dashboard' with the subtitle 'Environment summary'. It features an 'Environment info' table and several summary cards.

Environment info	
Environment	local 1 1 GB - Standalone 20.10.14
URL	/var/run/docker.sock
Tags	-

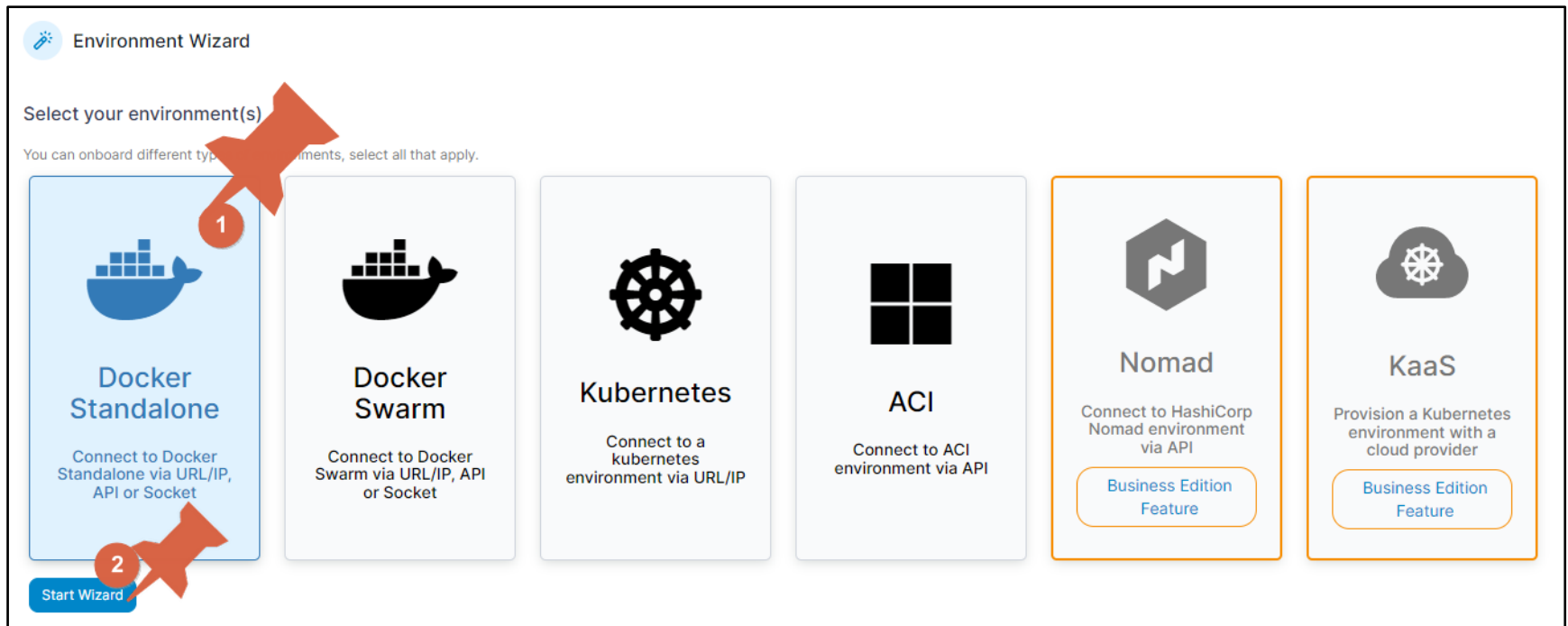
Resource Type	Count	Usage
Stacks	0	
Images	18	3.3 GB
Containers	9	
Volumes	1	
Networks	3	

Docker를 GUI로 관리하기

- 원격 docker host 관리하기
 - Portainer는 원격에 있는 docker host, Kubernetes, Azure Container Instance를 추가하여 하나의 Console에서 한꺼번에 관리할 수 있다
- 원격 docker host 추가하기
 - 먼저 관리될 원격 docker host(node1)에서 다음과 같이 원격 접속을 허용하는 설정을 한다(centos:7 기준)
 - **vi /usr/lib/systemd/system/docker.service**
ExecStart=/usr/bin/dockerd -H fd:// -H **tcp://0.0.0.0:2375**
 - **systemctl daemon-reload**
 - **systemctl restart docker**
 - 참고로 Ubuntu 16.04인 경우에는 다음과 같이 한다
 - **vi /etc/systemd/system/multi-user.target.wants/docker.service**
ExecStart=/usr/bin/dockerd -H fd:// -H **tcp://0.0.0.0:2375**
 - **systemctl daemon-reload**
 - **systemctl restart docker**

Docker를 GUI로 관리하기

- node2의 Portainer에서 원격 docker host(node1) **추가하기**
 - **Environments → Add Environment → Docker Standalone** 선택



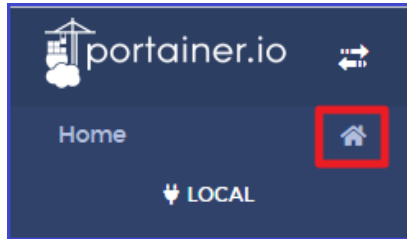
Docker를 GUI로 관리하기

- node2의 Portainer에서 원격 docker host(node1) **추가하기**
 - 관리할 도커 호스트 이름(별칭 가능)과 **"IP주소:2375"**를 입력한다

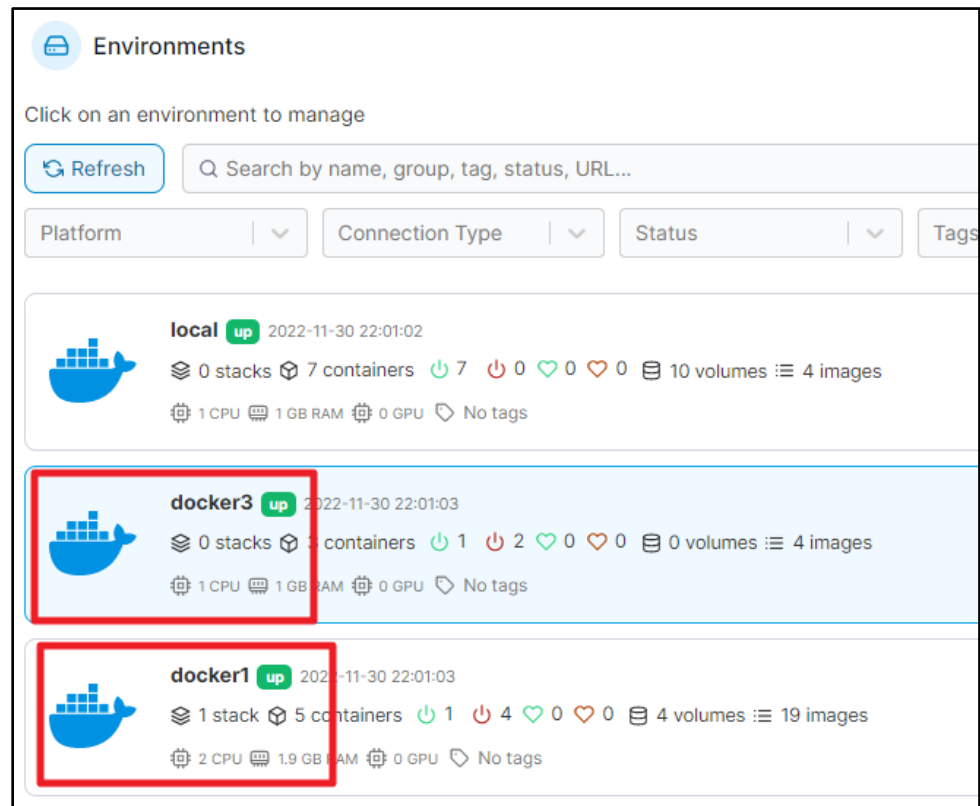
The screenshot shows the 'Environment Wizard' interface in Portainer. The first step is 'Docker Standalone'. Below the title, it says 'Connect to your Docker Standalone environment'. There are four options: 'Agent', 'API', 'Socket', and 'Edge Agent'. The 'API' option is selected, indicated by a blue checkmark and a red circle with the number '1'. Below these options, there are two input fields: 'Name*' with the value 'docker2' and 'Docker API URL*' with the value '192.168.56.123:2375'. A red circle with the number '2' is next to the 'Name' field. Below the input fields, there is a 'TLS' toggle switch which is currently off. At the bottom, there is a 'More settings' link and a 'Connect' button. A red circle with the number '3' is next to the 'Connect' button.

Docker를 GUI로 관리하기

- node2의 Portainer에서 원격 docker host(node1) 관리하기
 - Home을 클릭



- 관리할 node1을 클릭하여 Container를 관리한다



Docker를 GUI로 관리하기

- Portainer에서 Container 추가하기
 - `jesuswithme/nginxdemos`라는 이미지를 사용하여 Container를 생성한다
 - **Containers → Add container**
 - **name**과 **Image** 그리고 **Manual network port publishing**을 입력한 후 **Deploy the container**를 클릭한다

The screenshot shows the 'Add container' form in Portainer. Several fields are highlighted with red boxes to indicate where to enter information:

- Name:** A text input field containing 'myweb'.
- Image:** A dropdown menu showing 'docker.io' and a text input field containing 'jesuswithme/nginxdemos'.
- Manual network port publishing:** A section containing a 'publish a new network port' button and a table for port mapping.

The port mapping table is as follows:













host	container
8080	80

Docker를 GUI로 관리하기


- Portainer에서 Container 추가하기


- 생성한 Container를 실행한다


- myweb 컨테이너의 제일 우측의 8080:80을 클릭하여 접속한다

<input type="checkbox"/>	myweb	running	    	-	jesuswithme/nginxdemos.latest	2022-05-21 11:31:42	172.17.0.3	 8080:80
<input type="checkbox"/>	portainer	running	    	-	portainer/portainer-ce	2022-05-21 10:55:16	172.17.0.2	 9000:9000

- 결과는 실패!! 접속이 안된다.
 - 왼쪽 주메뉴의 **Environments**를 클릭한 후 local을 클릭하여 다음과 같이 해결한 후 다시 시도하면 접속된다(**##docker host의 IP 주소를 입력해야 하는 것이다**)

 Remove

 Add environment

<input type="checkbox"/>	Name	Type
<input type="checkbox"/>	local	 Docker

Configuration

Name

local

Environment URL ?

/var/run/docker.sock

Public IP ?

172.30.1.52

Docker를 GUI로 관리하기

- Portainer에서 Container 추가하기
 - 생성한 Container(myweb)에 성공적으로 접속

