

# Docker API. Настройка. Включение в Portainer

Материалы: <https://docs.docker.com/engine/daemon/remote-access/#additional-information> - делал по `systemctl edit`, с `docker.json` не заводится  
<https://www.devopsschool.com/blog/how-to-configure-docker-daemon-with-a-configuration-file/> - отсюда не брал, прост куча конфигов какой-то

---

## 1. Включение API на удаленном сервере с Docker.

---

В консоли прописываем `systemctl edit docker`, после чего попадаем в неведомый файл, явно не основной, но и хер пойми какой.

В нем прописываем (добавляем):

```
[Service]
ExecStart=
ExecStart=/usr/bin/dockerd -H fd:// -H
tcp://0.0.0.0:2375
```

После рестартим `systemd` и `docker`:

```
systemctl daemon-reload
systemctl restart docker
```

```
root@docker-portainer-client:~# ss -tulnp
```

Netid	State	Recv-Q	Send-Q	Local Address:Port	Peer Address:Port	Process
udp	UNCONN	0	0	127.0.0.53%lo:53	0.0.0.0:*	users: (("systemd-resolve", pid=98, fd=13))
udp	UNCONN	0	0	10.6.1.213%eth0:68	0.0.0.0:*	users: (("systemd-network", pid=80, fd=14))
tcp	LISTEN	0	100	127.0.0.1:25	0.0.0.0:*	users: (("master", pid=453, fd=13))
tcp	LISTEN	0	4096	127.0.0.1:32853	0.0.0.0:*	users: (("containerd", pid=122, fd=10))
tcp	LISTEN	0	4096	127.0.0.53%lo:53	0.0.0.0:*	users: (("systemd-resolve", pid=98, fd=14))
tcp	LISTEN	0	100	:::1:25	:::1:*	users: (("master", pid=453, fd=14))
tcp	LISTEN	0	4096	*:22	*:*	users: (("systemd", pid=1, fd=63))
tcp	LISTEN	0	4096	*:2375	*:*	users: (("dockerd", pid=682, fd=3))

```
root@docker-portainer-client:~#
```

```
curl http://ip:port/version --verbose
```

```
root@docker-portainer-client:~# curl 10.6.1.213:2375/version --verbose
* Trying 10.6.1.213:2375...
* Connected to 10.6.1.213 (10.6.1.213) port 2375 (#0)
> GET /version HTTP/1.1
> Host: 10.6.1.213:2375
> User-Agent: curl/7.81.0
> Accept: */*

* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< Api-Version: 1.43
< Content-Type: application/json
< Docker-Experimental: false
< Ostype: linux
< Server: Docker/24.0.7 (linux)
< Date: Sat, 28 Dec 2024 11:40:10 GMT
< Content-Length: 773

{"Platform":{"Name":"","Components":[{"Name":"Engine","Version":"24.0.7","Details":{"ApiVersion":"1.43","Arch":"amd64","BuildTime":"2024-03-13T20:23:54.000000000+00:00","Experimental":"false","GitCommit":"24.0.7-0ubuntu2-22.04.1","GoVersion":"go1.21.1","KernelVersion":"6.8.12-4-pve","MinAPIVersion":"1.12","Os":"linux"}},{"Name":"containerd","Version":"1.7.12","Details":{"GitCommit":"","Name":"runc","Version":"1.1.12-0ubuntu2-22.04.1","Details":{"GitCommit":"","Name":"docker-init","Version":"0.19.0","Details":{"GitCommit":"","Name":"runc","Version":"1.1.12-0ubuntu2-22.04.1","GoVersion":"go1.21.1","Os":"linux","Arch":"amd64","KernelVersion":"6.8.12-4-pve","BuildTime":"2024-03-13T20:23:54.000000000+00:00"}},{"Name":"docker-init","Version":"0.19.0","Details":{"GitCommit":"","Name":"runc","Version":"1.1.12-0ubuntu2-22.04.1","GoVersion":"go1.21.1","Os":"linux","Arch":"amd64","KernelVersion":"6.8.12-4-pve","BuildTime":"2024-03-13T20:23:54.000000000+00:00"}]}]}
* Connection #0 to host 10.6.1.213 left intact
root@docker-portainer-client:~#
```

Заходим в `Environment-related - Enviornments`, выбираем `Add environment`.

Environments					Q Search...	Remove	+ Add environment
<input type="checkbox"/>	Name ↓↑	Type ↓↑	URL ↓↑	Group Name ↓↑	Actions		
<input type="checkbox"/>	docker-ubuntu	Docker	tcp://10.6.1.213:2375	Unassigned	<a href="#">Manage access</a>		
<input type="checkbox"/>	local	Docker	unix:///var/run/docker.sock	Unassigned	<a href="#">Manage access</a>		
					Items per page	10	▼

Выбираем **Docker Standalone**, жмем **Start Wizard**.

Environment Wizard

Select your environment(s)  
You can onboard different types of environments, select all that apply.

Connect to existing environments

**Docker Standalone**  
Connect to Docker Standalone via URL/IP, API or Socket

☒

**Docker Swarm**  
Connect to Docker Swarm via URL/IP, API or Socket

☐

**Kubernetes**  
Connect to a Kubernetes environment via URL/IP

☐

**ACI**  
Connect to ACI environment via API

☐

Set up new environments

**Provision KaaS Cluster**  
Provision a Kubernetes cluster via a cloud provider's Kubernetes as a Service


☐ Business Feature

**Create Kubernetes cluster**  
Create a Kubernetes cluster on existing infrastructure

☐ Business Feature

Start Wizard


Попадаем в мастер настройки. Выбираем **API**, дальше указываем исходя из ваших данных, после жмем **Connect**.

 Environment Wizard


1

Docker Standalone


Connect to your Docker Standalone environment




Agent



API



Socket



Edge Agent Standard

Name\*

Имя сервера, которое будет отображаться

Docker API URL\*

IP адрес:порт сервера, который будет добавлен

TLS

☐

> More settings

⚡ Connect

← Previous

Close →

После бежим в **Home** , смотрим на наши среды, их должно быть больше 1 =)

portainer.io  
COMMUNITY EDITION

Home

docker-ubuntu

Dashboard

Templates

Stacks

Containers

Images

Networks

Volumes

Events

Host

Details

Setup

Registries

Home

Environments

Search by name, group, tag, status, URL... x

Refresh

Kubeconfig

Click on an environment to manage

Platform

Connectio...

Status

Tags



Groups

Agent Vers...

Clear all

Sort By

↓↑

 local  2024-12-28 16:31:23 Standalone 20.10.24+dfsg1 /var/run/docker.sock

Group: Unassigned

No tags

Local

0 stacks

3 containers

3

0

0

0

1 volume



2 images

1 CPU

536.9 MB RAM

Live connect

Disconnected

 docker-ubuntu  2024-12-28 16:31:23 Standalone 24.0.7 10.6.1.213:2375

Group: Unassigned

No tags

0 stacks

0 containers

0 volumes

0 images

1 CPU

536.9 MB RAM

Disconnect

Connected

Items per page 10