



# **Cross-organizational distributed systems and Clouds**

Solution for Exercise 6

Christopher B. Hauser

Institute of Information Resource Management

2018-06-21

## Answers to questions

### Lesson 1: Container Orchestration with Docker Swarm

#### Questions: Docker Swarm

*In the Swarm terminology, what are services, tasks, and containers?*

A service is a declarative description of a task, executed by a container. Several (replicated) tasks may serve as a service, while the task uses a container to run software to fulfil its purpose.

*Where in our Cloud Stack do you place Docker Swarm?*

The virtual machines have to be created externally. Docker engines have to be assigned to a Swarm cluster. Swarm automates from Containers on upwards.

Cloud Stack	Example	Deployment Tool
<b>Application Component</b>	Mediawiki	Dockerfile/Bash
<b>Containers</b>	Docker	Docker Swarm
<b>Virtual Resource</b>	Instance m1.small	Terraform
<b>Cloud Platform</b>	OpenStack	-

### Lesson 2: Container Orchestration with Rancher

#### Questions: Rancher

*Where in our Cloud Stack do you place Rancher?*

Rancher offers the full cloud stack: from allocating resources to container placement and triggers application deployment via Docker.

Cloud Stack	Example	Deployment Tool
<b>Application Component</b>	Mediawiki	Dockerfile/Bash
<b>Containers</b>	Docker	Rancher
<b>Virtual Resource</b>	Instance m1.small	Rancher

Cloud Stack	Example	Deployment Tool
<b>Cloud Platform</b>	OpenStack	-

Yet, Rancher does not automate the resource allocation depending on demands (e.g. http requests per second, or cpu load). This feature has to be added separately.

## Solution for practical part

### Docker Swarm

Docker Swarm works without additional software, since it is integrated in Docker. Yet it does not automate the creation of nodes when all available nodes are fully packed with containers. Scaling and updating containers works within seconds. Loadbalancing is partially replaced by Swarm's networking: services are accessible from any of the joined nodes.

### Rancher

Rancher starts virtual machines in bwcloud, and adds them as hosts to Rancher.

Rancher defines so called Stacks, which contain services. A service refers to a docker image, which is used to deploy a container to serve the service.

Rancher provides monitoring, control, and overview of hosts and containers.

The screenshot shows the OpenStack dashboard interface. The main content area is titled 'Instances' and contains a table with the following data:

Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
rancher-hosts3	Ubuntu Server 16.04 RAW	192.168.5.11 Floating IPs: 134.60.47.163	m1.small	rancher	Active	nova	None	Running	19 minutes	Create Snapshot
rancher-hosts1	Ubuntu Server 16.04 RAW	192.168.5.10 Floating IPs: 134.60.47.176	m1.small	rancher	Active	nova	None	Running	19 minutes	Create Snapshot
rancher	Ubuntu Server 16.04 RAW	192.168.5.3 Floating IPs: 134.60.47.164	m1.small	christopher-uulm	Active	nova	None	Running	1 hour, 7 minutes	Create Snapshot

The table is displayed with 3 items. The left sidebar shows the navigation menu with 'Instances' highlighted. The top bar shows the project name 'Projekt\_ehx27@uni-ulm.de' and the user 'ehx27@uni-ulm.de'.

Figure 1: rancher-bwcloud

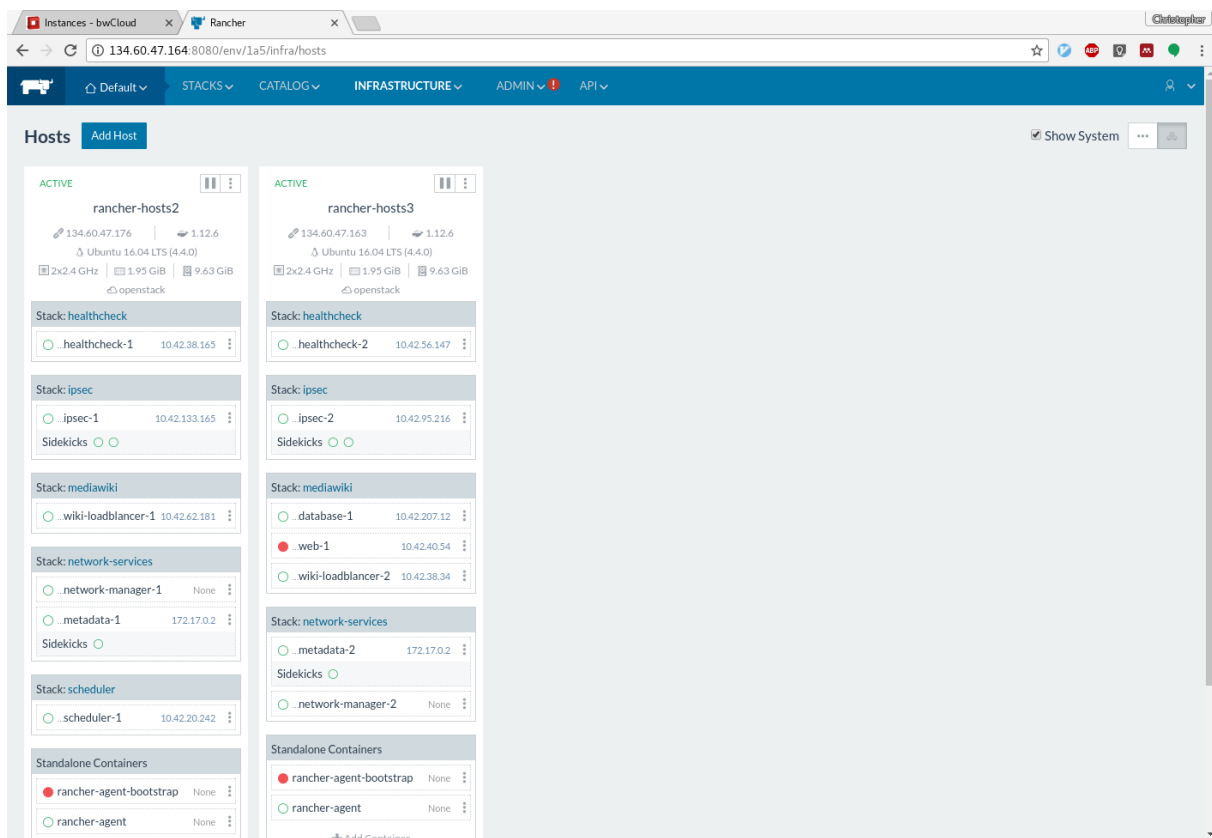


Figure 2: rancher-hosts

The screenshot displays the Rancher management interface for a stack named 'mediawiki'. The top navigation bar includes links for Default, STACKS, CATALOG, INFRASTRUCTURE, ADMIN, and API. The main content area shows a table of services within the stack:

Service	Image	Ports	Scale
database	bwcloud-fip164.rz.uni-ulm.de:5000/database		1
web	bwcloud-fip164.rz.uni-ulm.de:5000/mediawiki	80	1

Below the table, the 'web' service is selected, showing its details:

- Info:** Active, Image: bwcloud-fip164.rz.uni-ulm.de:5000/mediawiki, Entrypoint: None, Command: None.
- Containers:** 1 container, Scale: 1.
- Ports:** 134.60.47.163:80.
- Links:** No Links.

Figure 3: rancher-stackg

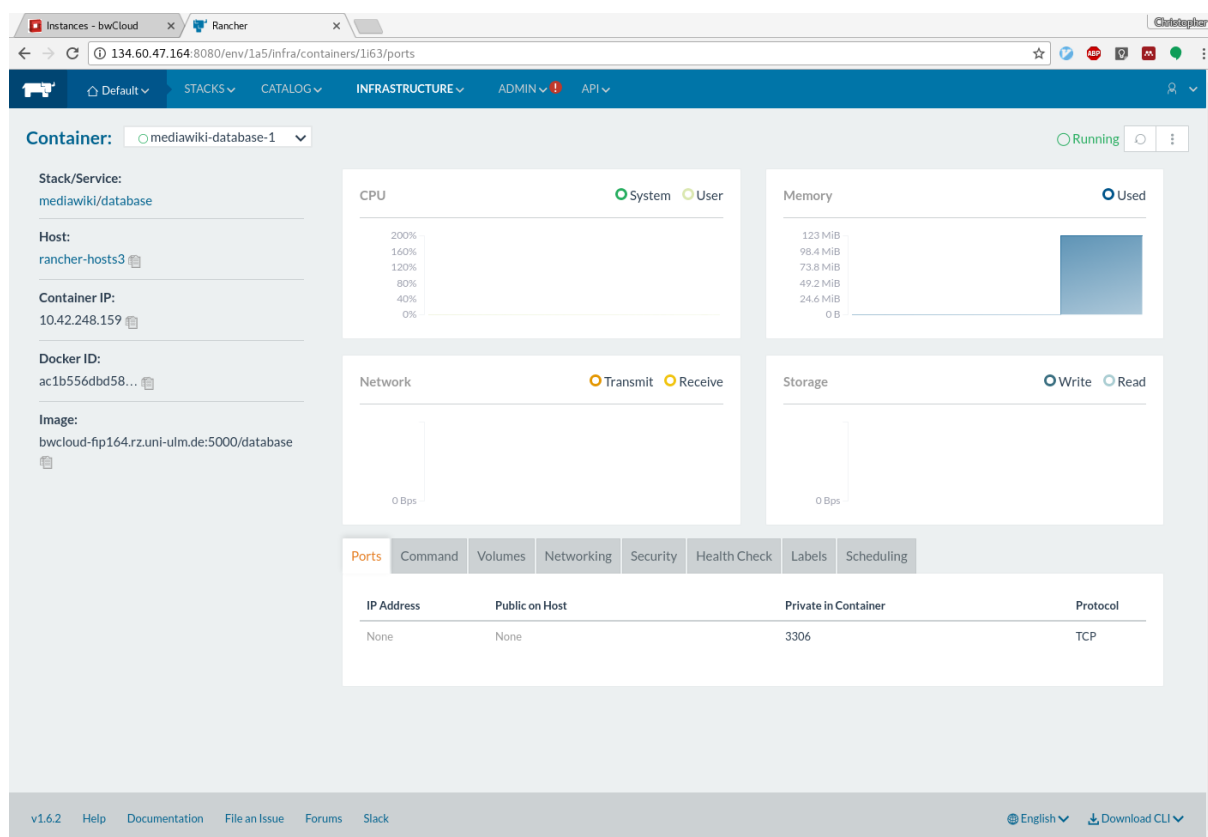


Figure 4: rancher-containerview