



# **Tools & Concepts for Cloud Deployments**

Solution for Exercise 6

Christopher B. Hauser

Institute of Information Resource Management, Ulm  
University

## 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
<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.

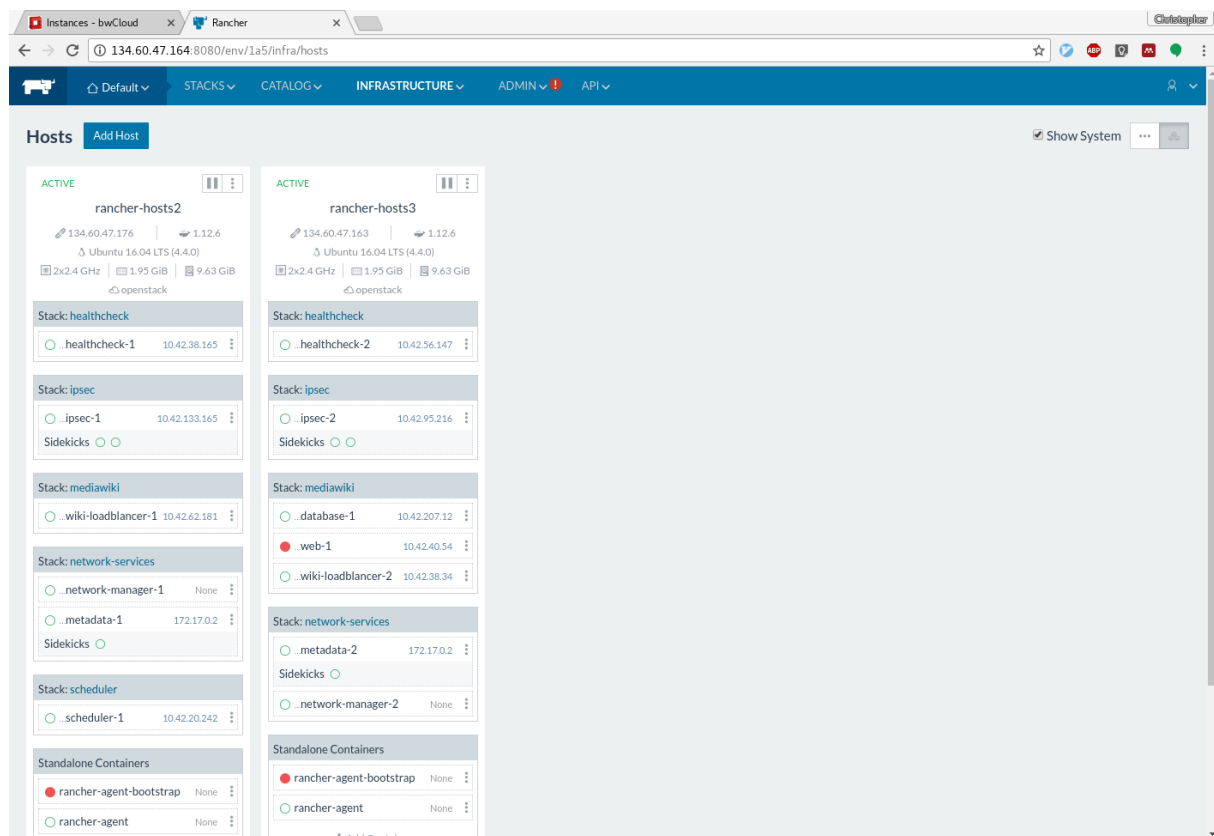
### Rancher

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

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-ulm	Active	nova	None	Running	1 hour, 7 minutes	Create Snapshot

Figure 1: rancher-bwcloud

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.

Figure 2: *rancher-hosts*

The screenshot shows the Rancher web interface in a browser. The top navigation bar includes links for Default, STACKS, CATALOG, INFRASTRUCTURE, ADMIN, and API. The main content area displays a stack named 'mediawiki'. Below the stack name, there is a table listing the services within the stack:

Status	Service Name	Image	Service Type	Containers
Active	database	Image: bwcloud-fip164.rz.uni-ulm.de:5000/database	Service	1 Container
Active	web	Image: bwcloud-fip164.rz.uni-ulm.de:5000/mediawiki Ports: 80	Service	1 Container

Below the table, there is a detailed view for the 'web' service. It includes the following information:

- Info (View Details):** Active, Image: bwcloud-fip164.rz.uni-ulm.de:5000/mediawiki, Entrypoint: None, Command: None.
- Containers (1):** Scale 1, with a circular progress indicator.
- Ports:** 134.60.47.163:80.
- Links:** No Links.

Figure 3: rancher-stackg

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

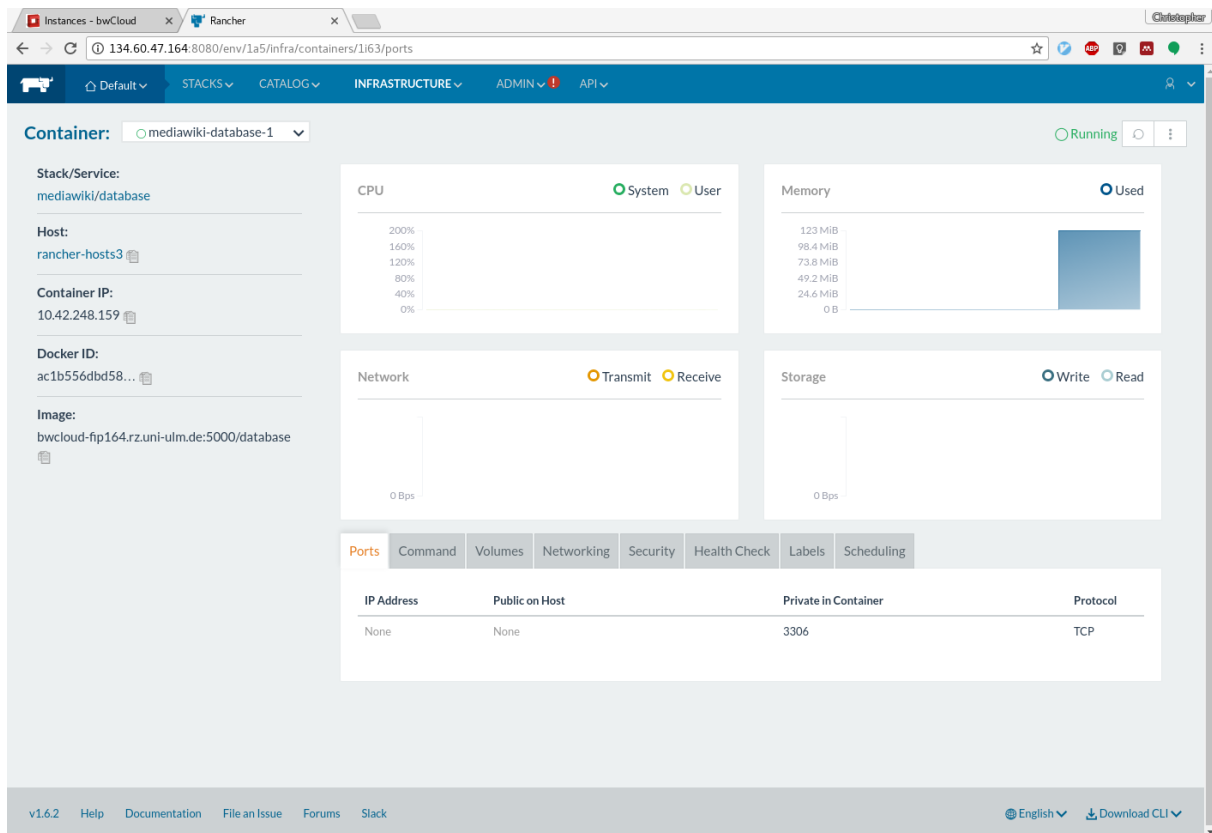


Figure 4: rancher-containerview