

1. Workshop	2
1.1 Prerequisites	2
1.2 Basic Scenarios	4
1.2.1 01 - Rancher Hosts Setup	5
1.2.1.1 Docker Machine	5
1.2.1.2 VirtualBox	8
1.2.2 02 - Rancher Server Setup	11
1.2.3 03 - Setup Access Control	12
1.2.4 04 - Create a Rancher Environment	12
1.2.5 05 - Rancher Agents Setup	13
1.2.6 06 - Run Standalone Container	15
1.2.7 07 - Create a Stack	22
1.2.8 08 - Create Stack From a Custom Catalog	23
1.3 Advanced Scenarios	25
1.3.1 01 - Monitor an Environment	25
1.3.2 02 - Use Secrets	27
1.3.3 03 - Setup a Load Balancer	30
1.3.3.1 Load Balancer for HTTP Only	30
1.3.3.2 Load Balancer for HTTP and HTTPS	35
1.3.3.3 Selector Rules	37
1.3.4 04 - Scheduling	38
1.3.5 05 - Use Rancher CLI	40
1.3.5.1 Create and Destroy Stack & Services Using Rancher CLI	42
1.3.5.2 Rancher CLI - Examples	43
1.3.6 06 - Use Rancher Compose	59
1.3.7 07 - Use Web API	61
1.3.8 08 - Webhooks	83

Workshop

- Prerequisites
- Basic Scenarios
 - 01 - Rancher Hosts Setup
 - 02 - Rancher Server Setup
 - 03 - Setup Access Control
 - 04 - Create a Rancher Environment
 - 05 - Rancher Agents Setup
 - 06 - Run Standalone Container
 - 07 - Create a Stack
 - 08 - Create Stack From a Custom Catalog
- Advanced Scenarios
 - 01 - Monitor an Environment
 - 02 - Use Secrets
 - 03 - Setup a Load Balancer
 - 04 - Scheduling
 - 05 - Use Rancher CLI
 - 06 - Use Rancher Compose
 - 07 - Use Web API
 - 08 - Webhooks

Prerequisites

In order to have a pleasant experience during Rancher Workshop, please ensure the followings are already done BEFORE the workshop begins (ideally, at least 1 day before):

1. Good connection to internet, since we're going to pull some Docker images!
2. Latest version of Oracle Virtualbox (v5.1.26 at the time of writing)
 - a. <https://www.virtualbox.org/wiki/Downloads>

In case you are using Windows 10 Pro with Hyper-V, please remove this hypervisor before installing Virtualbox.

3. Latest version of Docker Machine (v0.12.2 at the time of writing)
 - a. <https://docs.docker.com/machine/install-machine/>

Do not touch your current Docker installation, if you have one.

4. Once you have these in place, please run the following commands in a terminal of your choice to validate your setup:
▼ [Click here to expand...](#)

```
# Create a VM named "rs", running Rancher OS v1.0.4 with Docker
v17.0.3.1-ce
# using docker-machine and VirtualBox provider (2 CPU cores, 2 GB
RAM and 10 GB disk space)
docker-machine create -d virtualbox --virtualbox-boot2docker-url
https://releases.rancher.com/os/v1.0.4/rancheros.iso
--virtualbox-cpu-count 2 --virtualbox-memory 2048
--virtualbox-disk-size 10000 rs
```

```
# Once the above command completes successfully, display the newly
created VM
docker-machine ls

## Command output (or something similar, in case you have other VMs
created using docker-machine)
# NAME      ACTIVE    DRIVER      STATE      URL
SWARM      DOCKER    ERRORS
# rs        -         virtualbox   Running    tcp://192.168.99.100:2376
Unknown    Unable to query docker version: Cannot connect to the
docker engine endpoint
```

```
# Enter the newly created VM
docker-machine ssh rs
```

```
# Once inside the VM, display Docker version
docker version

## Command output
# Client:
# Version:      17.03.1-ce
# API version:  1.27
# Go version:   go1.7.5
# Git commit:   c6d412e
# Built:        Tue Mar 28 00:40:02 2017
# OS/Arch:      linux/amd64
#
# Server:
# Version:      17.03.1-ce
# API version:  1.27 (minimum version 1.12)
# Go version:   go1.7.5
# Git commit:   c6d412e
# Built:        Tue Mar 28 00:40:02 2017
# OS/Arch:      linux/amd64
# Experimental: false
```

```
# To ensure you have a working version of Docker, run the
hello-world container
docker container run hello-world

## Command output
# Unable to find image 'hello-world:latest' locally
# latest: Pulling from library/hello-world
# b04784fba78d: Pull complete
# Digest:
sha256:f3b3b28a45160805bb16542c9531888519430e9e6d6ffc09d72261b0d26f
f74f
# Status: Downloaded newer image for hello-world:latest
#
# Hello from Docker!
# This message shows that your installation appears to be working
correctly.
#
# To generate this message, Docker took the following steps:
# 1. The Docker client contacted the Docker daemon.
# 2. The Docker daemon pulled the "hello-world" image from the
Docker Hub.
# 3. The Docker daemon created a new container from that image
which runs the
#     executable that produces the output you are currently reading.
# 4. The Docker daemon streamed that output to the Docker client,
which sent it
#     to your terminal.
#
# To try something more ambitious, you can run an Ubuntu container
with:
# $ docker run -it ubuntu bash
#
# Share images, automate workflows, and more with a free Docker ID:
# https://cloud.docker.com/
#
# For more examples and ideas, visit:
# https://docs.docker.com/engine/userguide/
```

5. If all of the above completed successfully, congrats! You are good to go for the Rancher Workshop!

Basic Scenarios

This page displays basic scenarios one must cover in order to start using Rancher.

- [01 - Rancher Hosts Setup](#)
- [02 - Rancher Server Setup](#)
- [03 - Setup Access Control](#)
- [04 - Create a Rancher Environment](#)

- 05 - Rancher Agents Setup
- 06 - Run Standalone Container
- 07 - Create a Stack
- 08 - Create Stack From a Custom Catalog

01 - Rancher Hosts Setup

This page show the different ways one can use for creating hosts used for running Rancher server and agents.

This workshop will make use of VMs created using Docker Machine.

- Docker Machine
- VirtualBox

Docker Engine supports [plenty of platforms](#), but there are some operating systems built having in mind the specific purpose of acting as Docker container hosts.

Here are some of them:

- [boot2docker](#)
 - Currently, unable to use Rancher secrets, see more here: <https://github.com/rancher/rancher/issues/7707#issuecomment-277841482>
- CoreOS
- RancherOS **[TO BE USED DURING THIS WORKSHOP]**
- Others

There are also other ways of creating Docker hosts to be used by Rancher server and its agents:

- Plain-old VMs + Docker classic installation (<https://docs.docker.com/engine/installation/>)
- 10 acre ranch for macOS (<https://github.com/rancher/10acre-ranch#docker-machine-on-macos>)
 - Based on [Docker Machine for Mac](#)
 - Supports [xhyve](#) (a macOS native hypervisor) only
 - Not very difficult to add support for other docker-machine drivers - just check <https://github.com/rancher/10acre-ranch/blob/master/bin/mac-ranch>
- Vagrant box with pre-installed Docker
 - <https://github.com/phusion/open-vagrant-boxes>
 - Others
- Others

Docker Machine

This page details how to run Rancher server and 2 agents on VMs created via Docker Machine.

This is the approach to be used during Java Tech Group Day #5 Rancher workshop!

- Prerequisites
- RancherOS
- Create VMs Using Docker Machine
 - Create Rancher server Docker machine
 - Create Rancher agent #1 Docker machine
 - Create Rancher agent #2 Docker machine
 - Get Docker machine IP address
 - Connect to Docker machine via SSH
 - Change Docker machine system parameters

Prerequisites

- Check [Prerequisites](#) section.

RancherOS

This OS was built specifically for hosting Docker containers and it's officially supported by Rancher server.

- RancherOS Home: <http://rancher.com/rancher-os/>
- RancherOS Official Documentation: <http://rancher.com/docs/os/v1.0/en/>
- This workshop will use RancherOS v1.0.4, available here: <https://releases.rancher.com/os/v1.0.4/rancheros.iso>
- Latest version can be found here: <https://releases.rancher.com/os/latest/rancheros.iso>

- All versions can be found here: <https://github.com/rancher/os/releases>

You can control a RancherOS instance via ros command:

```
$ sudo ros
# NAME:
#   ros - Control and configure RancherOS
#built: '2017-08-08T00:05:15Z'
#
#USAGE:
#   ros [global options] command [command options] [arguments...]
#
#VERSION:
#   v1.0.4
#
#AUTHOR(S):
#   Rancher Labs, Inc.
#
#COMMANDS:
#   config, c    configure settings
#   console      manage which console container is used
#   engine       manage which Docker engine is used
#   service, s
#   os          operating system upgrade/downgrade
#   tls          setup tls configuration
#   install      install RancherOS to disk
#   selinux
#   help, h      Shows a list of commands or help for one command
#
#GLOBAL OPTIONS:
#   --help, -h    show help
#   --version, -v print the version
```

Create VMs Using Docker Machine

More information can be found [here](#) and [here](#).

On Windows, please run docker-machine as admin!

Create Rancher server Docker machine

```
docker-machine create -d virtualbox --virtualbox-boot2docker-url
https://releases.rancher.com/os/v1.0.4/rancheros.iso --virtualbox-cpu-count
2 --virtualbox-memory 2048 --virtualbox-disk-size 10000 rs
```

Create Rancher agent #1 Docker machine

```
docker-machine create -d virtualbox --virtualbox-boot2docker-url
https://releases.rancher.com/os/v1.0.4/rancheros.iso --virtualbox-cpu-count
2 --virtualbox-memory 2048 --virtualbox-disk-size 20000 ra1
```

Create Rancher agent #2 Docker machine

```
docker-machine create -d virtualbox --virtualbox-boot2docker-url
https://releases.rancher.com/os/v1.0.4/rancheros.iso --virtualbox-cpu-count
2 --virtualbox-memory 2048 --virtualbox-disk-size 20000 ra2
```

Get Docker machine IP address

In order to obtain the IP address of a VM created via docker-machine, run the following command:

```
# Solution #1
docker-machine ip <DOCKER_MACHINE_NAME>

# Solution #2
docker-machine inspect <DOCKER_MACHINE_NAME> --format={{.Driver.IPAddress}}
```

Connect to Docker machine via SSH

```
docker-machine ssh rs
docker-machine ssh ra1
docker-machine ssh ra2
```

Change Docker machine system parameters

In case your Rancher server is keep showing an error page, check the rs machine logs.

You might see:

```
...
time="2017-08-20T16:29:03Z" level=info msg="Starting Catalog Service
(port 8088, refresh interval 300 seconds)"
fatal error: runtime: out of memory
...
```

If this is the case, you may want to allocated more RAM to this machine.

```
# stop VM
docker-machine stop <DOCKER_MACHINE_NAME>

# set number of CPU cores
VBoxManage modifyvm <DOCKER_MACHINE_NAME> --cpus 2

# set amount of RAM in MB
VBoxManage modifyvm <DOCKER_MACHINE_NAME> --memory 4096

# start VM
docker-machine start <DOCKER_MACHINE_NAME>
```

VirtualBox

This page details how to run Rancher server and 2 agents on VMs created via Oracle VirtualBox.

This is an alternative approach for [Docker Machine](#).

Error rendering macro 'toc' : For input string: " 2 "

Prerequisites

- Install VirtualBox, if you haven't done so already.

In case your host is running Windows 10 Pro + Hyper-V, please remove this hypervisor before running any VirtualBox VMs!

- Have the ability of accessing a machine using SSH
 - On Windows, you can use Git Bash, which is included in [Git for Windows](#)

VirtualBox Virtual Machines

#	VM Name	Role	OS	CPU Cores	RAM (MB)	Disk* (GB)	Comments
1	rs	Rancher server	CoreOS	2	1024	10	<ul style="list-style-type: none"> • VirtualBox VM Settings Network Adapter 1: NAT • VirtualBox VM Settings Network Adapter 2: Bridges Adapter
2	ra1	Rancher agent #1			2048	20	
3	ra2	Rancher agent #2			2048	20	

Hard disk file type: VDI (VirtualBox Disk Image)

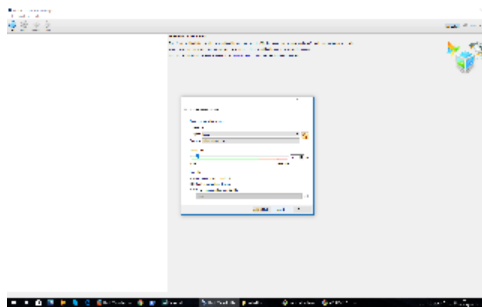
Storage on physical hard disk: Dynamically allocated

VirtualBox VM Installation Steps

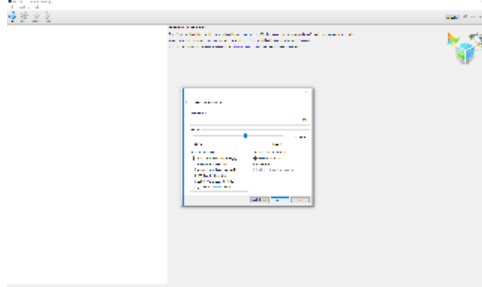
In case your host is running Windows 10 Pro + Hyper-V, please remove this hypervisor before running any VirtualBox VMs!

▼ [Click here to expand...](#)

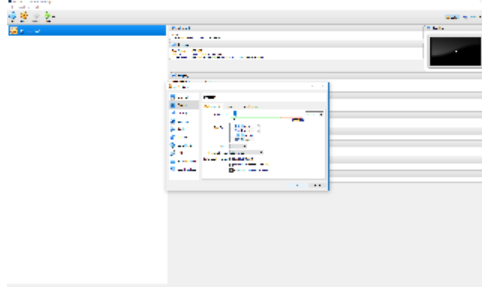
1. Specify name and operating system



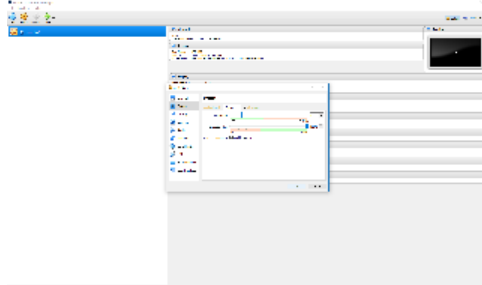
2. Specify file location



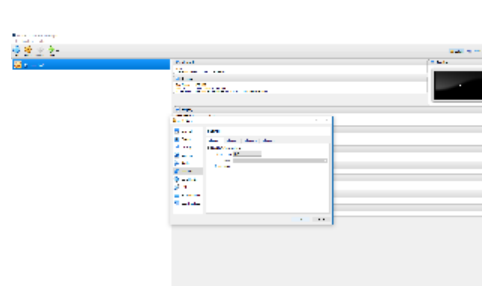
3. Specify amount of RAM



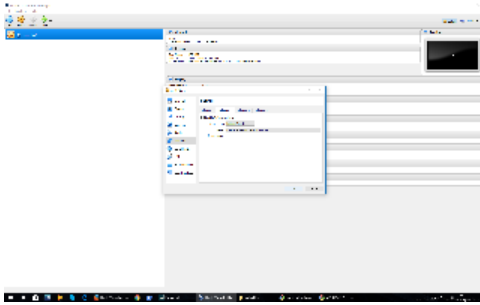
4. Specify number of CPU cores



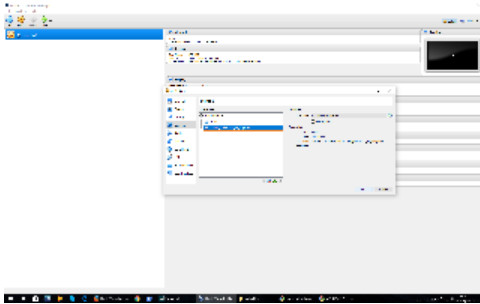
5. Enable network adapter: NAT



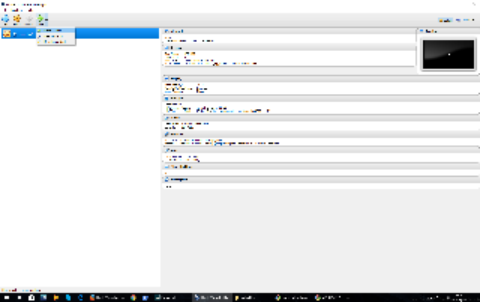
6. Enable network adapter: Bridged Adapter



7. Mount CoreOS ISO



8. Start VM



CoreOS Installation Steps

CoreOS is a OS which was built specifically for hosting Docker containers; it not officially supported by Rancher server.

- CoreOS Home: <https://coreos.com/>
- CoreOS Official Documentation: <https://coreos.com/os/docs/latest>
- CoreOS Quick Guide (Unofficial): www.liberidu.com/blog/2015/04/11/basic-newbie-install-coreos-on-virtualbox-getting-started-with-docker/

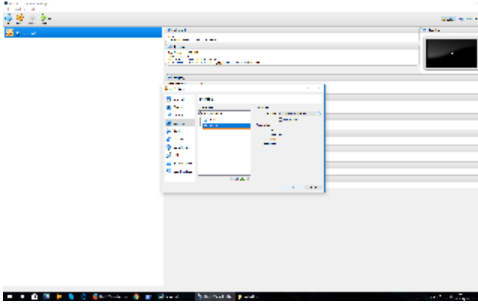
Follow the steps below when creating one VM for hosting Rancher server and another one for hosting one Rancher agent, then, for each additional agent, just clone the VM hosting the first agent!

Click here to expand...

1. Start VM from VirtualBox GUI
2. Generate password inside cloud-config.yml file
Error rendering macro 'code': Invalid value specified for parameter 'lang'
3. Edit cloud-config.yml file
Error rendering macro 'code': Invalid value specified for parameter 'lang'
4. Cloud-config.yml contents
Error rendering macro 'code': Invalid value specified for parameter 'lang'
5. Install CoreOS
Error rendering macro 'code': Invalid value specified for parameter 'lang'

6. Shutdown machine
Error rendering macro 'code': Invalid value specified for parameter 'lang'

7. Unmount CoreOS ISO from VirtualBox GUI



8. Start VM from VirtualBox GUI
9. Login using rancher/rancher (or the credentials you have previously set)
10. Display VM IP v4 Address (something like 192.168....)
Error rendering macro 'code': Invalid value specified for parameter 'lang'
11. Optionally, change the hostname accordingly
Error rendering macro 'code': Invalid value specified for parameter 'lang'

02 - Rancher Server Setup

This page details how to start an instance of a Rancher server.

More information can be found [here](#).

1. Connect to the VM to host Rancher server using ssh

```
docker-machine ssh rs
```

2. Start Rancher server Docker container

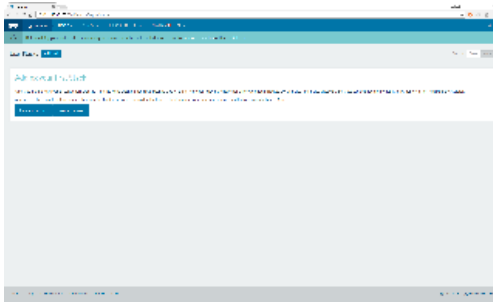
This workshop uses Rancher Server v1.6.7, the latest version at the time of writing this phrase.

```
sudo docker run -d --name rs -p 8080:8080 --restart=unless-stopped  
rancher/server:v1.6.7
```

3. Monitor Rancher server logs

```
docker logs -f rs
```

4. Go to Rancher UI: http://<RANCHER_SERVER_HOST_IP_ADDRESS>:8080

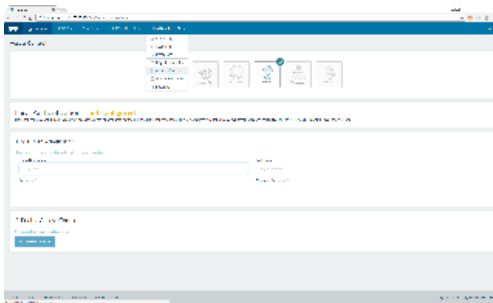


03 - Setup Access Control

This page details how to secure access to a running instance of Rancher server.

More information can be found [here](#).

1. Go to Rancher UI
2. Go to menu ADMIN Access Control
3. Select "LOCAL" as Access Control and fill in the fields appropriately



4. Click "Enable Local Auth" bottom left button
5. Logout
6. Login using the newly created user account

04 - Create a Rancher Environment

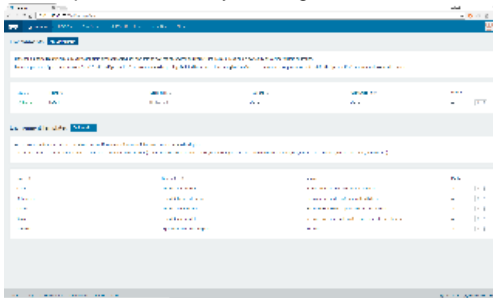
This page details how to create an environment inside a running instance of Rancher server.

An environment uses one or more registered hosts (Docker hosts, aka Rancher agents) to run Docker containers.

An environment groups stacks, which are composed of services, which are made out of one or more containers.

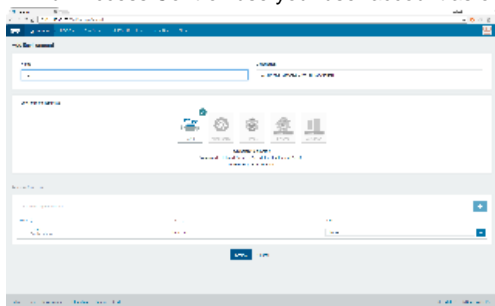
More information can be found [here](#).

1. Go to Rancher UI
2. Go to top left menu entry Manage Environments

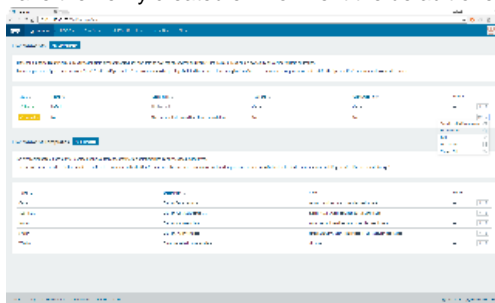


3. Click the "Add Environment" button
4. Fill in the fields

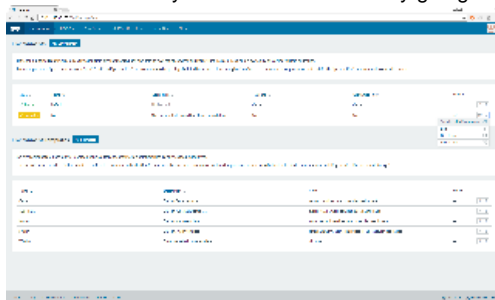
- a. Name: `tgd`
- b. Description: Used by Java Tech Group Day #5 Rancher workshop
- c. Environment Template: Cattle
- d. Access Control: use your user account as owner for this new environment



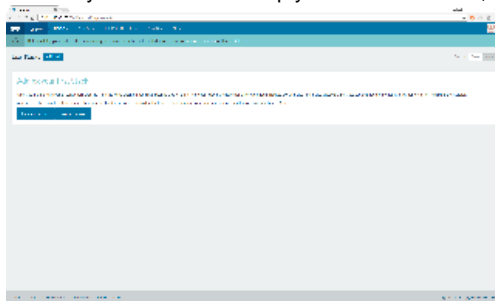
5. Click "Create" button to have Rancher create this environment
6. Make the newly created environment the default one by going to environment menu Set as Default



7. Switch to the newly created environment by going to environment menu Switch to this Environment



8. The newly environment is empty: it hosts no stacks, it has no registered hosts



9. Optionally, you can delete "Default" Rancher environment, if you no longer use it

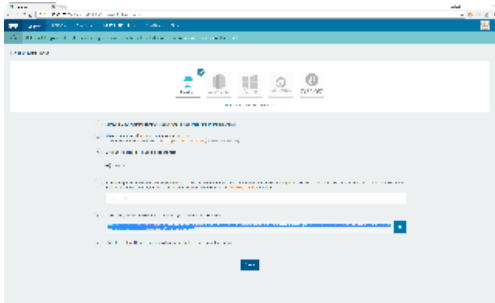
05 - Rancher Agents Setup

This page details how to register several hosts (aka Rancher agents) to a running instance of Rancher server.

A Rancher agent is a Docker host which runs one or more containers.

More information can be found [here](#).

1. Go to Rancher UI
2. Select "tgd" environment, if it's not already the selected one
3. Go to menu Infrastructure Hosts click "Add Host" button



4. Copy the above selected text
5. Connect to the VM to host Rancher agent #1 using ssh

```
docker-machine ssh ra1
```

6. Paste and run the previously copied command

```
## The command looks something like this:  
# sudo docker run --rm --privileged -v  
/var/run/docker.sock:/var/run/docker.sock -v  
/var/lib/rancher:/var/lib/rancher rancher/agent:v1.2.5  
http://192.168.99.100:8080/v1/scripts/E16CD86C33A1E2399AA1:1483142400  
000:uMn9PmfeLyOZLJQckI3uLnQ0s
```

The above command will create a Docker container named **rancher-agent**.

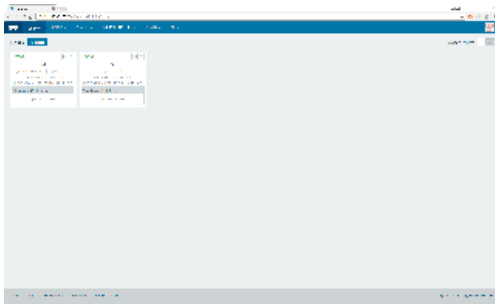
7. Connect to the VM to host Rancher agent #2 using ssh

```
docker-machine ssh ra2
```

8. Paste and run the previously copied command

```
## The command looks something like this:  
# sudo docker run --rm --privileged -v  
/var/run/docker.sock:/var/run/docker.sock -v  
/var/lib/rancher:/var/lib/rancher rancher/agent:v1.2.5  
http://192.168.99.100:8080/v1/scripts/E16CD86C33A1E2399AA1:1483142400  
000:uMn9PmfeLyOZLJQckI3uLnQ0s
```

9. Go to Rancher UI
10. Go to menu Infrastructure Hosts and you should see 2 hosts: ra1 and ra2, registered to "tgd" environment



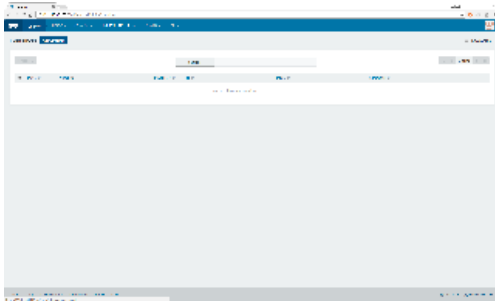
06 - Run Standalone Container

This page details how to run a container on a running instance of Rancher server, without creating a stack or service.

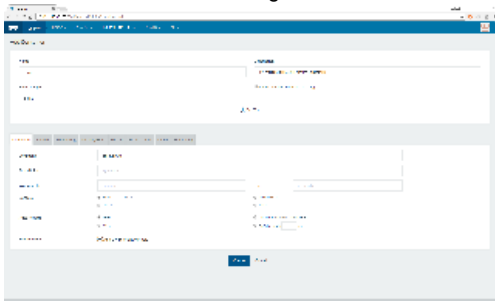
Running a standalone container may prove useful for maintenance tasks, where you are running specialized containers.

More information can be found [here](#).

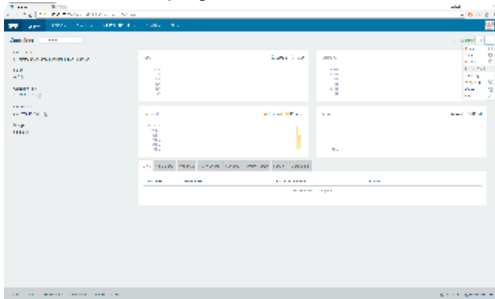
1. Go to Rancher UI
2. Go to menu Infrastructure Containers



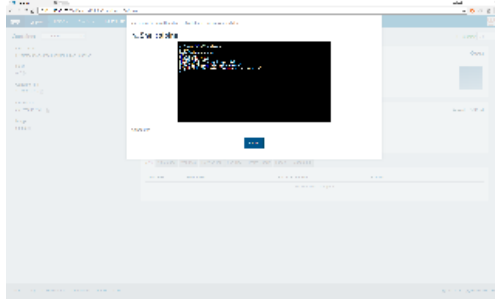
3. Click the top left "Add Container" button
4. Fill in the fields in order to start a container based on the [alpine](#) Docker image
 - a. Name: alpine
 - b. Description: My first adhoc container running on Rancher!
 - c. Image: alpine
 - d. Command tab Command: tail -f /dev/null
 - This will force the container to remain in running state after start-up so that one may execute shell commands, view container logs or interact with it in other any way



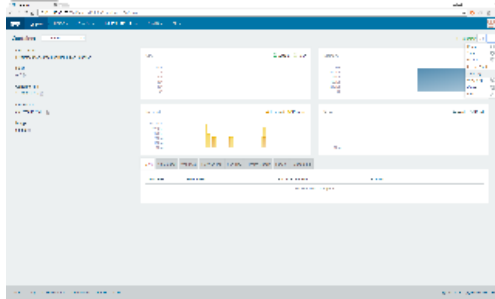
5. Click the bottom button "Create" to have Rancher starting the container
6. Go to container top right menu Execute Shell



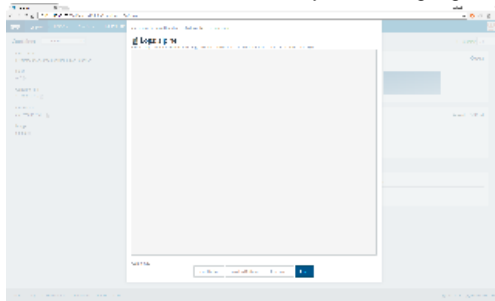
- Execute "cat /etc/*-release" command in the opened terminal to display the current OS name and version



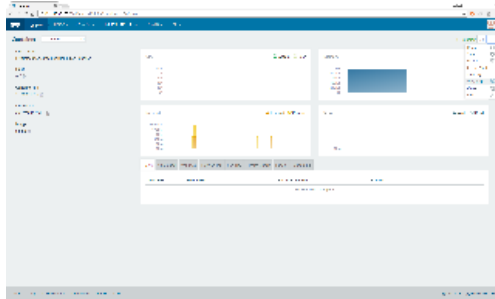
- Close the shell by clicking the "Close" button
- Go to the container top right menu View Logs



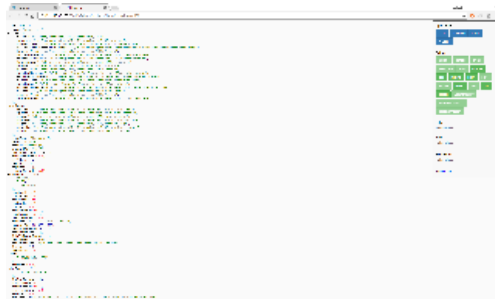
- This container does not show very interesting log entries



- Close the log view by clicking the "Close" button
- Go to container top right menu View in API



- API container view



Container JSON




```
{
  "id": "li38",
  "type": "container",
  "links": {

    "self": "http://192.168.99.100:8080/v2-beta/projects/1a16/containers/li38",

    "account": "http://192.168.99.100:8080/v2-beta/projects/1a16/containers/li38/account",

    "credentials": "http://192.168.99.100:8080/v2-beta/projects/1a16/containers/li38/credentials",

    "healthcheckInstanceHostMaps": "http://192.168.99.100:8080/v2-beta/projects/1a16/containers/li38/healthcheckinstancehostmaps",

    "hosts": "http://192.168.99.100:8080/v2-beta/projects/1a16/containers/li38/hosts",

    "instanceLabels": "http://192.168.99.100:8080/v2-beta/projects/1a16/containers/li38/instancelabels",

    "instanceLinks": "http://192.168.99.100:8080/v2-beta/projects/1a16/containers/li38/instancelinks",

    "instances": "http://192.168.99.100:8080/v2-beta/projects/1a16/containers/li38/instances",

    "mounts": "http://192.168.99.100:8080/v2-beta/projects/1a16/containers/li38/mounts",

    "ports": "http://192.168.99.100:8080/v2-beta/projects/1a16/containers/li38/ports",

    "serviceEvents": "http://192.168.99.100:8080/v2-beta/projects/1a16/containers/li38/serviceevents",

    "serviceExposeMaps": "http://192.168.99.100:8080/v2-beta/projects/1a16/containers/li38/serviceexposemaps",

    "serviceLogs": "http://192.168.99.100:8080/v2-beta/projects/1a16/containers/li38/servicelogs",

    "services": "http://192.168.99.100:8080/v2-beta/projects/1a16/containers/li38/services",

    "targetInstanceLinks": "http://192.168.99.100:8080/v2-beta/projects/1a16/containers/li38/targetinstancelinks",

    "volumes": "http://192.168.99.100:8080/v2-beta/projects/1a16/containers/li38/volumes",
```

```
"stats": "http://192.168.99.100:8080/v2-beta/projects/1a16/containers/li38/stats",

"containerStats": "http://192.168.99.100:8080/v2-beta/projects/1a16/containers/li38/containerstats"
},
"actions": {

"restart": "http://192.168.99.100:8080/v2-beta/projects/1a16/containers/li38/?action=restart",

"update": "http://192.168.99.100:8080/v2-beta/projects/1a16/containers/li38/?action=update",

"stop": "http://192.168.99.100:8080/v2-beta/projects/1a16/containers/li38/?action=stop",

"migrate": "http://192.168.99.100:8080/v2-beta/projects/1a16/containers/li38/?action=migrate",

"logs": "http://192.168.99.100:8080/v2-beta/projects/1a16/containers/li38/?action=logs",

"execute": "http://192.168.99.100:8080/v2-beta/projects/1a16/containers/li38/?action=execute",

"proxy": "http://192.168.99.100:8080/v2-beta/projects/1a16/containers/li38/?action=proxy"
},
"baseType": "instance",
"name": "alpine",
"state": "running",
"accountId": "1a16",
"blkioDeviceOptions": null,
"blkioWeight": null,
"build": null,
"capAdd": [

],
"capDrop": [

],
"cgroupParent": null,
"command": [
    "tail",
    "-f",
    "/dev/null"
],
"cpuCount": null,
"cpuPercent": null,
"cpuPeriod": null,
"cpuQuota": null,
"cpuSet": null,
```

```
"cpuSetMems":null,
"cpuShares":null,
"createIndex":null,
"created":"2017-08-20T13:16:07Z",
"createdTS":1503234967000,
"dataVolumeMounts":{

},
"dataVolumes":[

],
"dataVolumesFrom":[

],
"deploymentUnitUuid":null,
"description":"My first adhoc container running on Rancher!",
"devices":[

],
"diskQuota":null,
"dns":[
    "169.254.169.250"
],
"dnsOpt":null,
"dnsSearch":[
    "rancher.internal"
],
"dockerPorts":[

],
"domainName":null,
"entryPoint":null,
"environment":null,
"expose":null,

"externalId":"a017fbdb676d9016e6881a0cd1c31a4926367c597284f57460d49cf9c9fd494f",
"extraHosts":null,
"firstRunning":"2017-08-20T13:16:09Z",
"firstRunningTS":1503234969000,
"groupAdd":null,
"healthCheck":null,
"healthCmd":null,
"healthInterval":null,
"healthRetries":null,
"healthState":null,
"healthTimeout":null,
"hostId":"1h4",
"hostname":null,
"imageUuid":"docker:alpine",
"instanceTriggeredStop":"stop",
"ioMaximumBandwidth":null,
"ioMaximumIOps":null,
```

```
"ip":null,
"ip6":null,
"ipcMode":null,
"isolation":null,
"kernelMemory":null,
"kind":"container",
"labels":{"
    "io.rancher.container.ip":"10.42.230.36\16",

"io.rancher.container.uuid":"07a9a647-08fa-45ce-bb64-222783712ed2",
    "io.rancher.cni.network":"ipsec",
    "io.rancher.container.name":"alpine",
    "io.rancher.cni.wait":"true",
    "io.rancher.container.mac_address":"02:69:76:ad:8f:aa"
},
"logConfig":{"
    "type":"logConfig",
    "config":{"

    },
    "driver":""
},
"lxcConf":null,
"memory":null,
"memoryReservation":null,
"memorySwap":null,
"memorySwappiness":null,
"milliCpuReservation":null,
"mounts":null,
"nativeContainer":false,
"netAlias":null,
"networkContainerId":null,
"networkMode":"managed",
"oomKillDisable":null,
"oomScoreAdj":null,
"pidMode":null,
"pidsLimit":null,
"ports":[

],
"primaryIpAddress":"10.42.230.36",
"primaryNetworkId":"ln10",
"privileged":false,
"publishAllPorts":false,
"readOnly":false,
"registryCredentialId":null,
"removed":null,
"requestedHostId":null,
"restartPolicy":null,
"runInit":false,
"secrets":[

],
```

```
"securityOpt":null,
"serviceId":null,
"serviceIds":null,
"shmSize":null,
"stackId":null,
"startCount":1,
"startOnCreate":true,
"stdinOpen":true,
"stopSignal":null,
"storageOpt":null,
"sysctls":null,
"system":false,
"tmpfs":null,
"transitioning":"no",
"transitioningMessage":null,
"transitioningProgress":null,
"tty":true,
"ulimits":null,
"user":null,
"userPorts":[

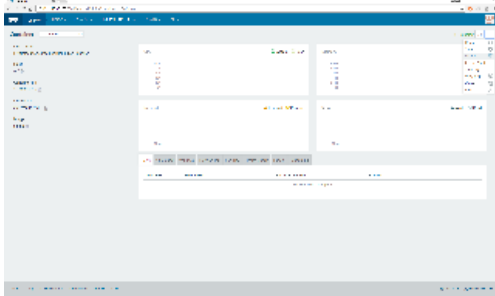
],
"usersnsMode":null,
"uts":null,
"uuid":"07a9a647-08fa-45ce-bb64-222783712ed2",
"version":"0",
```

```

    "volumeDriver":null,
    "workingDir":null
  }

```

14. Optionally, you can remove this container, if you do not use it anymore; go to container top right menu Delete



07 - Create a Stack

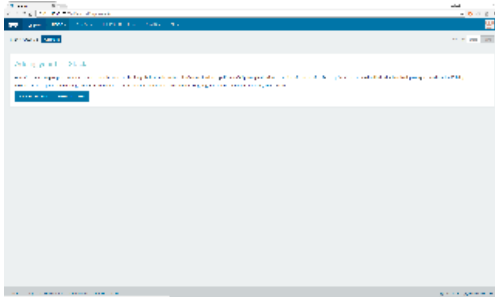
This page details how to create a stack on a running instance of Rancher server.

A stack groups one or more service, which are composed of one or more containers.

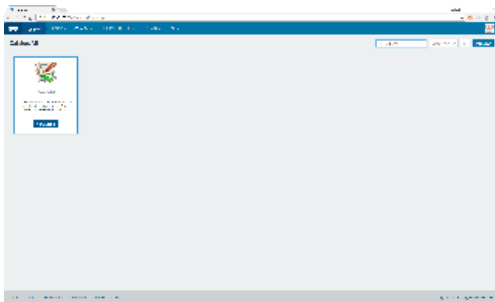
Usually, one would associate a stack with an application and the stack services with application subsystem (database, back-end, cache, etc.).

More information can be found [here](#).

1. Go to Rancher UI
2. Go to menu Stacks

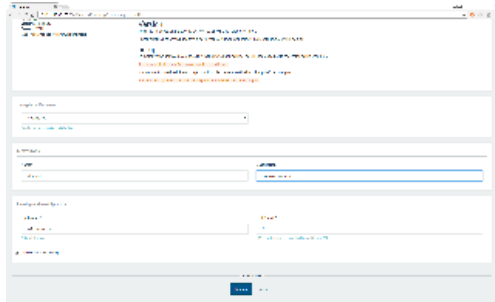


3. Click "Browse Catalog" button
4. Take a look all those catalog items ...
5. Search catalog by "DokuWiki"
6. Click the "View Details" button

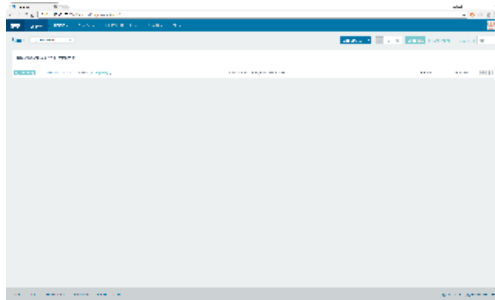


7. Fill in the fields
 - a. Template Version: 2016-06-26a
 - b. Name: dokuwiki1
 - c. Description: DokuWiki instance #1

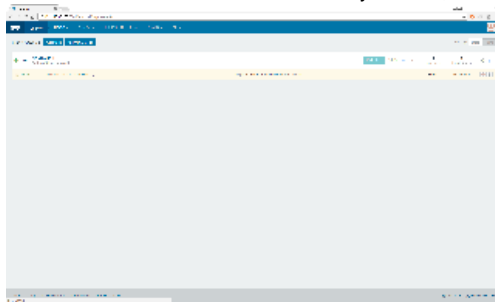
- d. Hostname: dw1.example.com
- e. HTTP port: 80



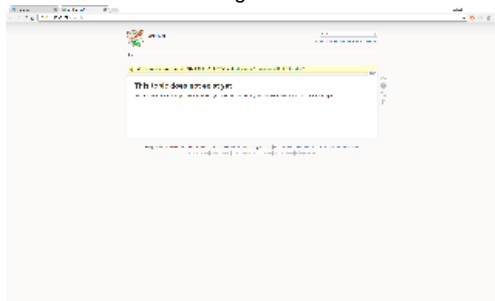
- 8. Click the "Launch" button
- 9. See the stack being created and its services being started



- 10. After all services has been successfully started, click the link "80" from the Ports field



- 11. Take a look at the running DokuWiki instance



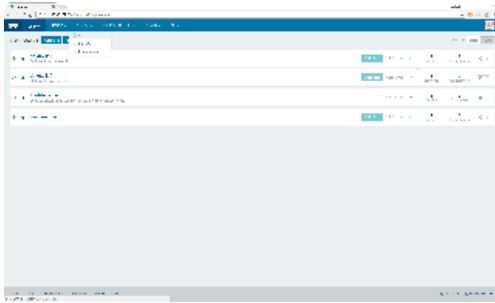
08 - Create Stack From a Custom Catalog

This page details how to register a custom catalog into a running instance of Rancher server and how to create a stack based on this catalog.

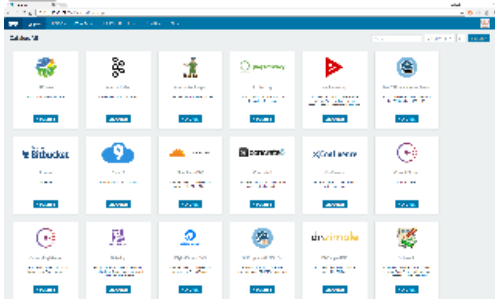
Catalogs allow packing complex Docker Compose setups which can easily be started even by non-technical persons using a wizard-like setup form.

More information can be found [here](#) and [here](#).

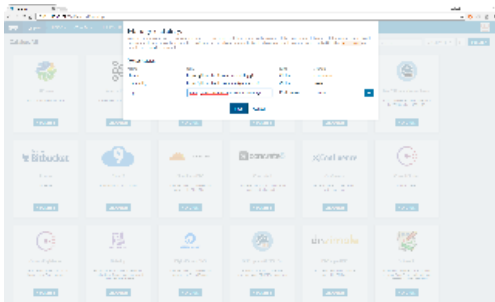
- 1. Go to Rancher UI
- 2. Go to menu Catalog All



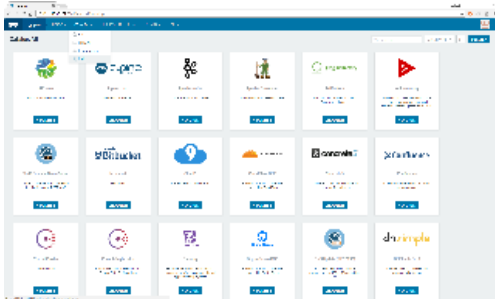
3. Click the top right "Manage" button



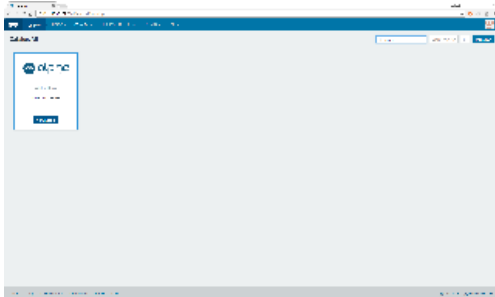
4. Click the "Add Catalog" link and fill in the fields accordingly
 - a. Name: `tgd`
 - b. URL: `https://github.com/satrapu/rancher-workshop.git`
 - c. Branch: `master`



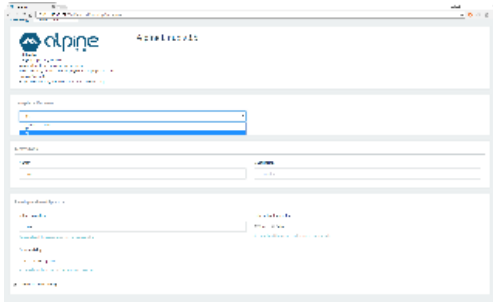
5. Click "Save" button and you should now see a new catalog, "tgd"



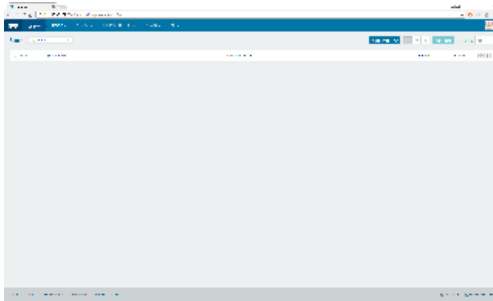
6. Filter catalog items by text "Alpine"



7. Click "View Details"
8. Fill in the mandatory and optional fields as you see fit; play around to see how this UI behaves



- Click the "Launch" button and see the new stack being created and started



Advanced Scenarios

This page displays advanced scenarios one should cover in order to start using Rancher beyond its basics.

- 01 - Monitor an Environment
- 02 - Use Secrets
- 03 - Setup a Load Balancer
- 04 - Scheduling
- 05 - Use Rancher CLI
- 06 - Use Rancher Compose
- 07 - Use Web API
- 08 - Webhooks

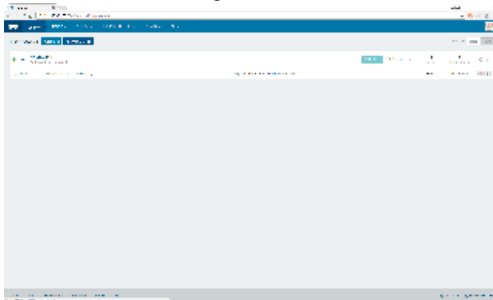
01 - Monitor an Environment

This page details how to monitor a Rancher environment using Weavescope.

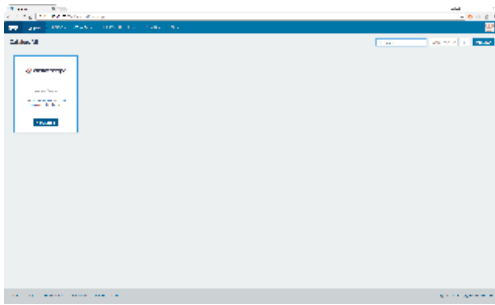
Monitoring is essential for keeping your environment and its hosts in good shape.

More information can be found [here](#).

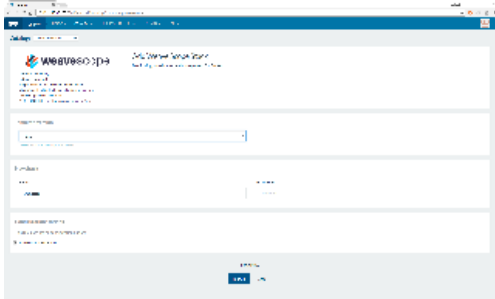
- Go to Rancher UI
- Click "Add from Catalog" button



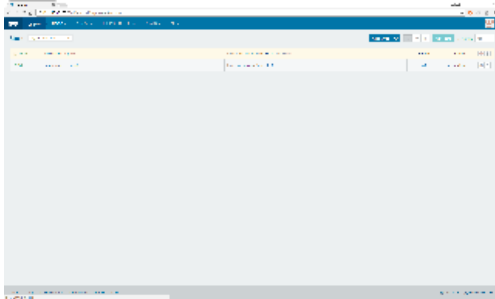
- Filter catalog items by "weavescope"



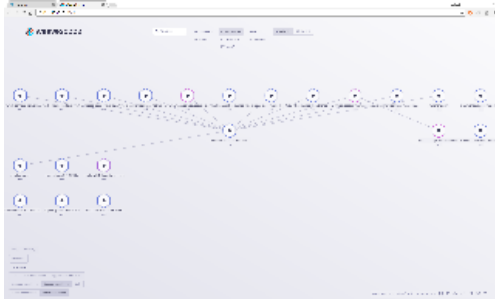
4. Click the "View Details" button
5. Leave fields with default values and click "Launch" button



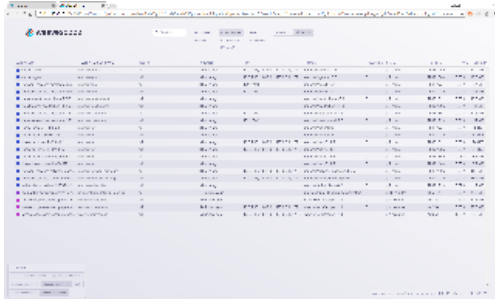
6. Wait for all WeaveScope services to start, then click the "4040" link from Ports field



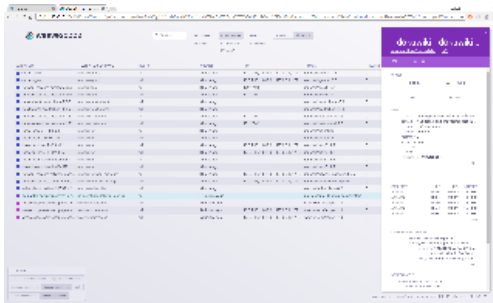
7. See WeaveScope in action



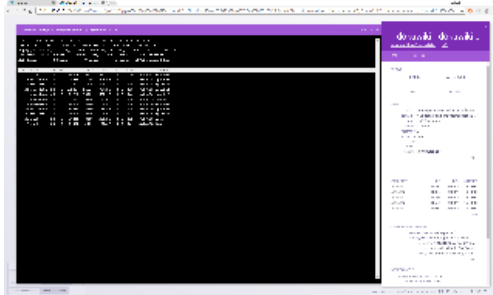
8. Organize WeaveScope data by Containers displayed as table



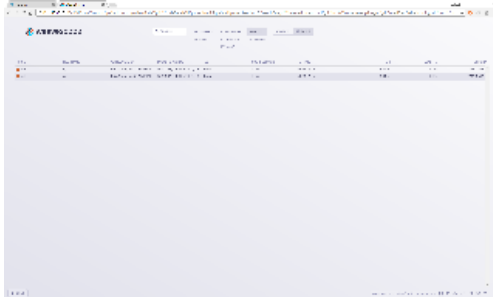
9. Click the containers running DokuWiki
10. Click the "Exec shell" link



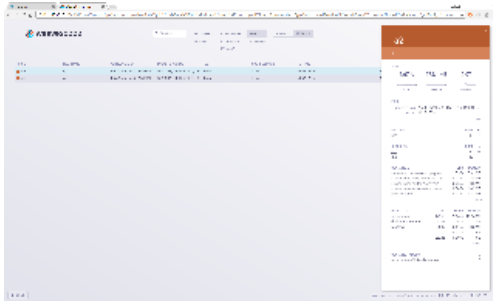
11. Inside the console, type "top" command and take a peek inside this container processes



12. Close the console when you're done
13. Organize WeaveScope data by Hosts displayed as table



14. Click the row displaying ra2 host to open an overview of this host



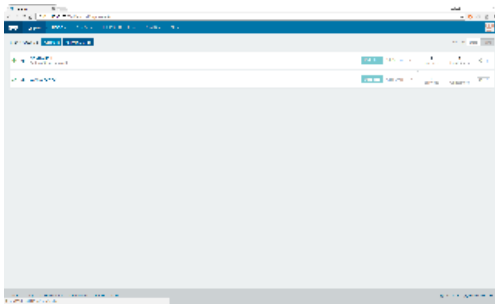
02 - Use Secrets

This page details how to provide secrets to a container.

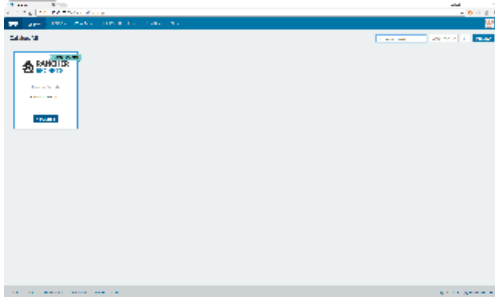
Passing sensitive data (e.g. credentials, certificates, etc.) to containers should be done in a secured manner and Rancher offers support for this kind of activity.

More information can be found [here](#).

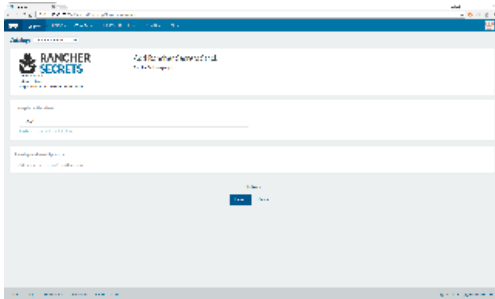
1. Go to Rancher UI
2. Go to menu Stacks User
3. Click "Add from catalog"



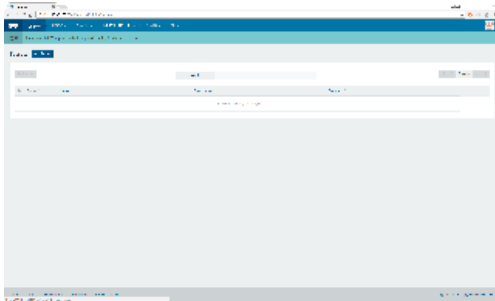
4. Filter catalog items by "rancher secrets"



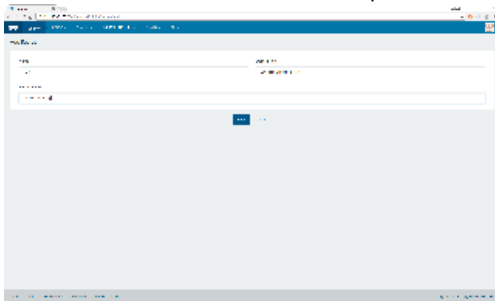
5. Click the "View Details" button
6. Leave all fields set to default values and click "Launch" button



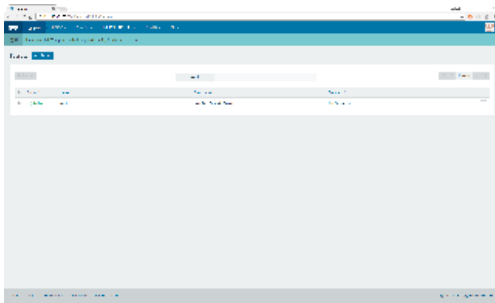
7. Go to menu Infrastructure Secrets



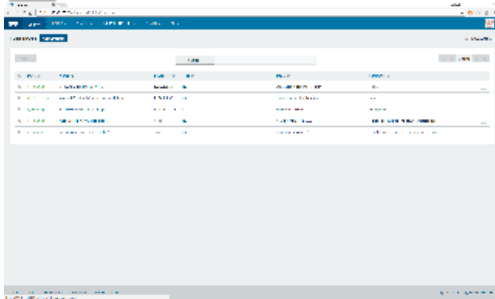
8. Click "Add Secret" button
9. Fill in the fields accordingly
 - a. Name: tgd5
 - b. Description: Java Tech Group #5 Secret
 - c. Secret Value: rancher-workshop



10. Click "Save" button and you should see the newly added secret



11. Go to menu Infrastructure Containers



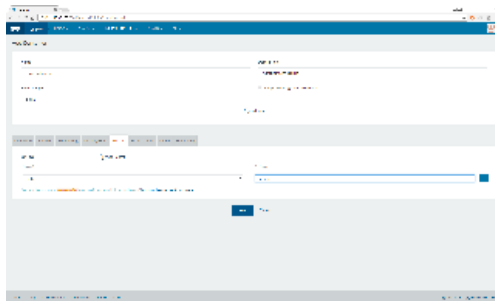
12. Click the top left "Add Container" button

13. Fill in the fields in order to start a container based on the [alpine](#) Docker image

- Name: alpine
- Description: My first adhoc container running on Rancher!
- Image: alpine
- Command tab Command: `tail -f /dev/null`
 - This will force the container to remain in running state after start-up so that one may execute shell commands, view container logs or interact with it in other any way
- Secrets tab Add Secret: choose `tgds` and optionally can set an alias, via `As Name` field

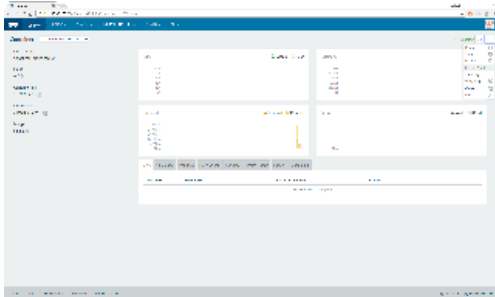
If the "As Name" field was not set, the secret will be available at path: `/run/secrets/tgds`.

In the "As Name" field was set to "my_secret", the secret value will be available at path: `/run/secrets/my_secret`.

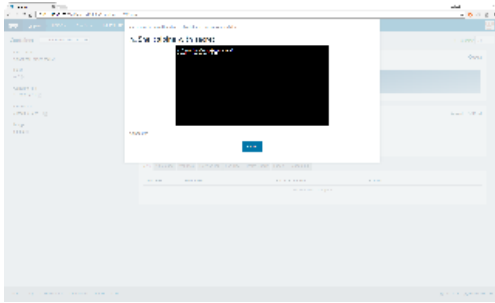


14. Click the "Create" button

15. Once the container is running, go to container top right menu and select "Execute Shell"



16. Inside the console, type "`cat /run/secrets/my_secret`" to print the secret value



03 - Setup a Load Balancer

This page details how to setup a load balancer in a running instance of Rancher server.

A load balancer can be used for distributing requests between several hosts, can be used for providing access via HTTPS and can additionally perform SSL termination.

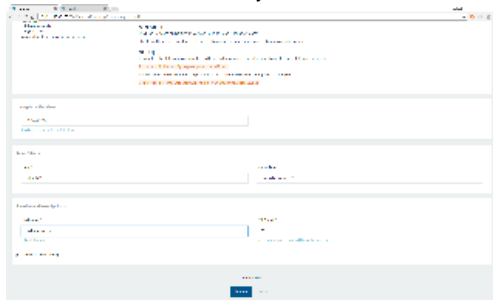
- [Load Balancer for HTTP Only](#)
- [Load Balancer for HTTP and HTTPS](#)
- [Selector Rules](#)

Load Balancer for HTTP Only

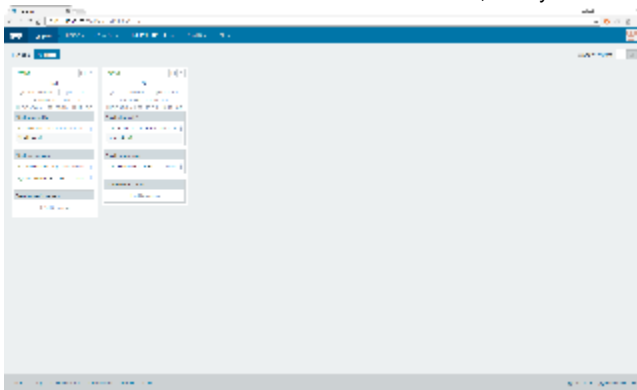
This page details how to setup a load balancer to distribute HTTP requests made against several instances of DokuWiki hosted on a running instance of Rancher server.

More information can be found [here](#).

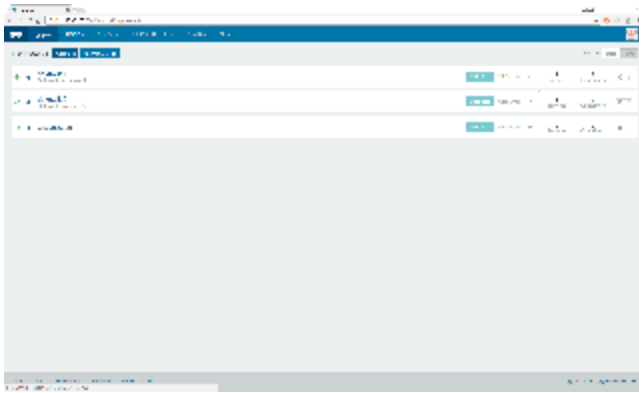
1. Go to Rancher UI
2. Create 2 stacks based on DokuWiki catalog item
 - a. The first one was created while performing [07 - Create a Stack](#)
 - b. The second one should only contain minor differences



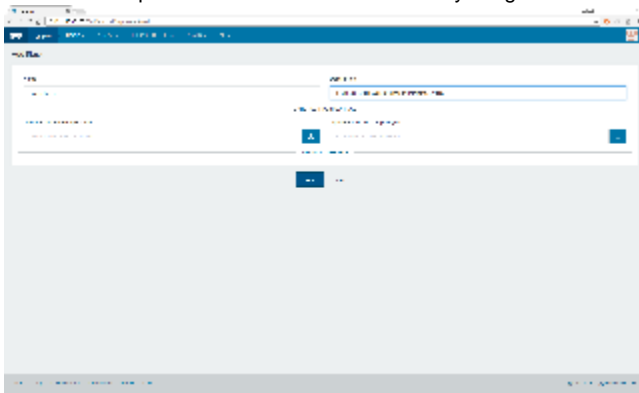
3. Wait for the second stack to be created and started, then you should see they are running on different hosts



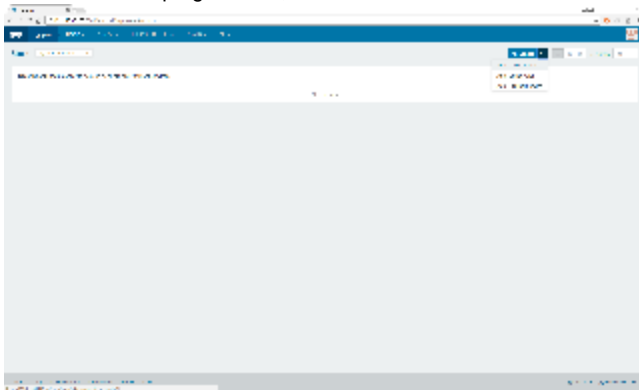
4. Go to menu Stacks User
5. Click "Add Stack" button



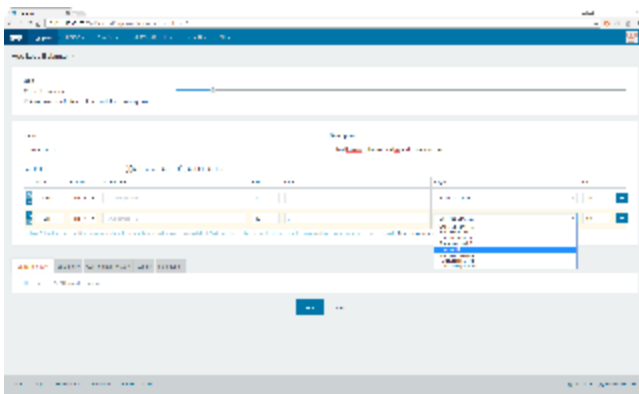
6. Fill in the fields accordingly
 - a. Name: load-balancer
 - b. Description: Load balancer to be used by all tg4 environment services.



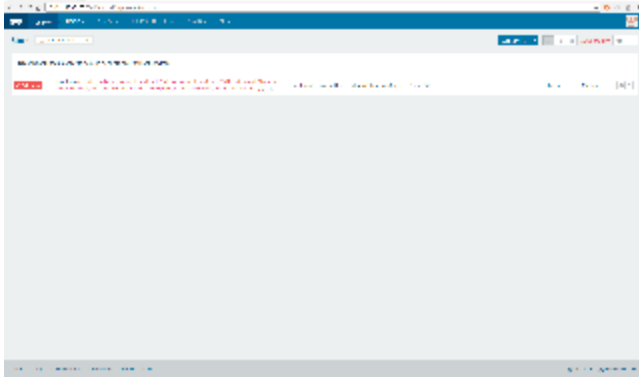
7. Click "Create" button
8. From the stack top right menu, select "Add Load Balancer Service"



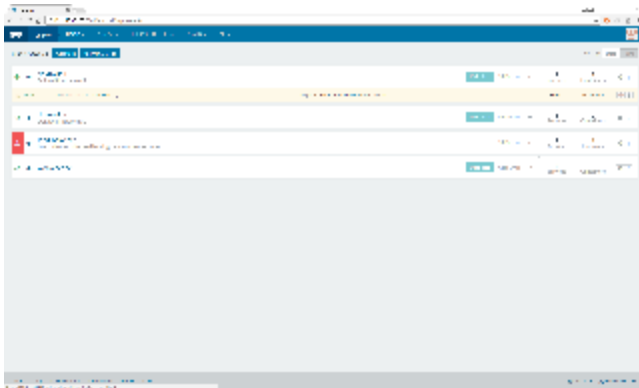
9. Fill in the fields accordingly
 - a. Add a Service Rule
 - i. Access: Public
 - ii. Protocol: HTTP
 - iii. Request Host: leave empty
 - iv. Port: 80
 - v. Path: /
 - vi. Target: dokuwiki-server from stack dokuwiki1
 - vii. Port: 80 (this is the container internal port, where its Apache process will listen for any incoming requests)
 - b. Add another Service Rule with the same fields, just change Target to dokuwiki-server from stack dokuwiki2



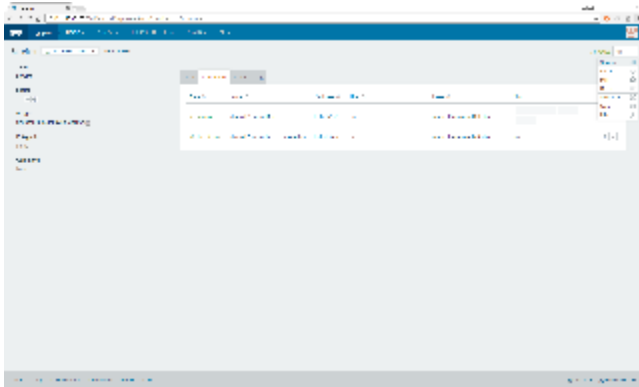
10. Click the "Create" button
11. This will fail, since we have 2 services on a host, trying to use the same port 80



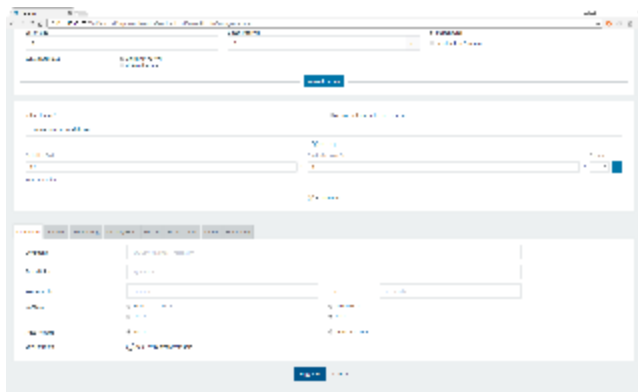
12. Update both DokuWiki services to use a different HTTP port: 8081, respectively 8082
13. Go to menu Stacks User
14. Click the dokuwiki-server link from dokuwiki1 stack



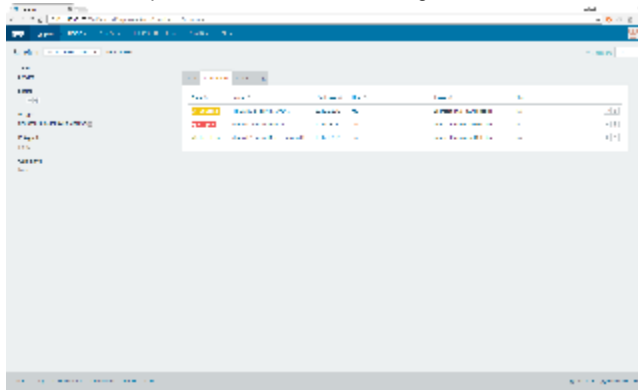
15. From the service top right menu, click "Upgrade"



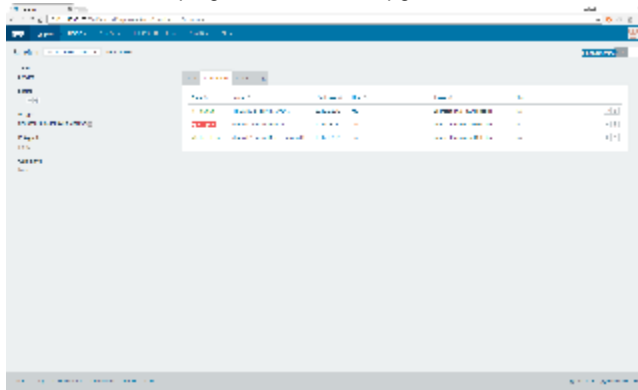
16. Change only one field: Public Host Port to 8081, then click "Upgrade" button



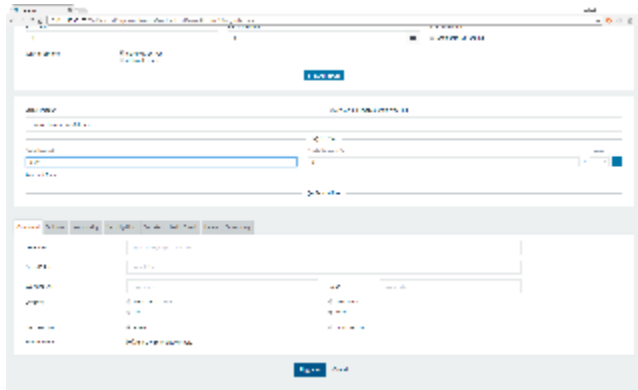
17. Rancher will stop the container of the existing service and will create another one with the new settings



18. Click the service top right menu Finish Upgrade; this will remove the old container

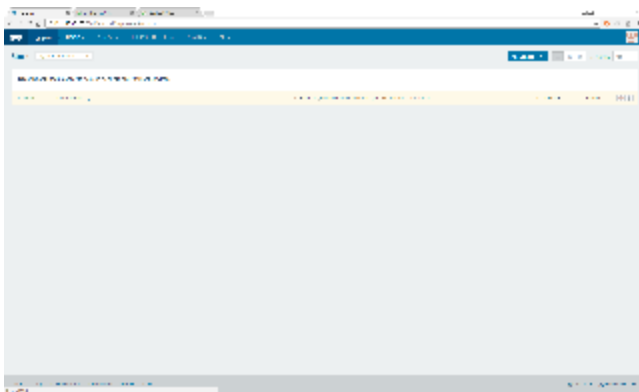


19. Do the same for dokuwiki-server link from dokuwiki2 stack; set Public Host Port field to 8082

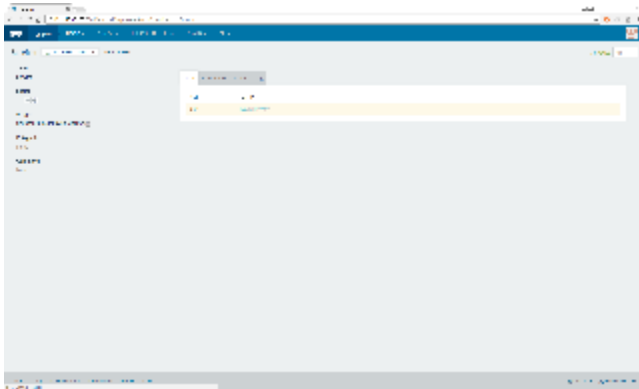


20. The load balancer service should be running OK by now

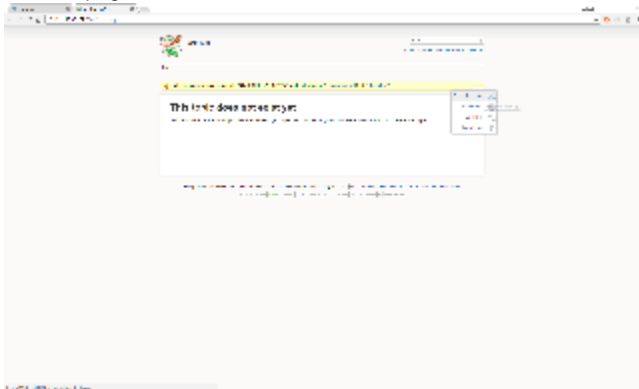
21. Click the "80/tcp" link from its Port field to navigate to one DokuWiki instance



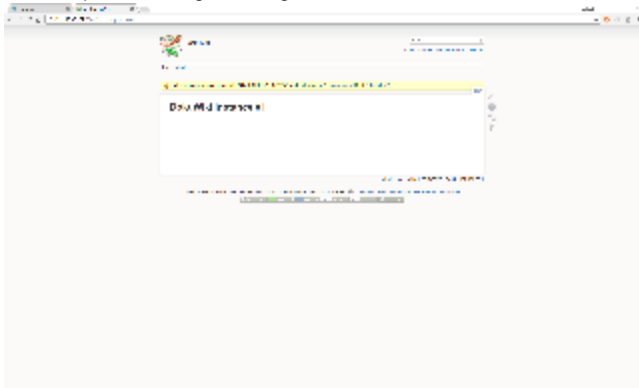
22. Refresh page several times and you'll see you cannot tell which DokuWiki instance is in fact serving the request, so we need to visit each DokuWiki instance and create a page with a specific title to be used as a discriminator
23. Go to dokuwiki-server service from dokuwiki1 stack and click the Host IP link associated with Port 8081



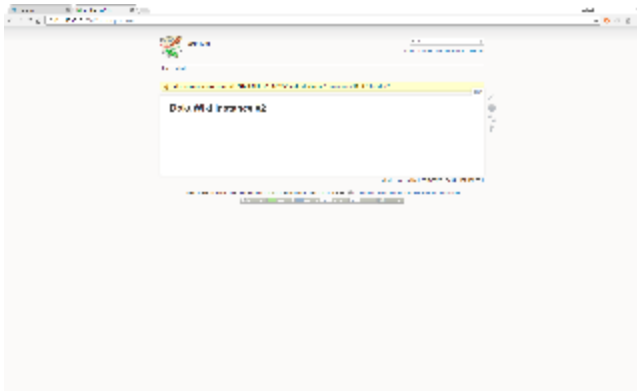
24. Create a page inside the DokuWiki instance



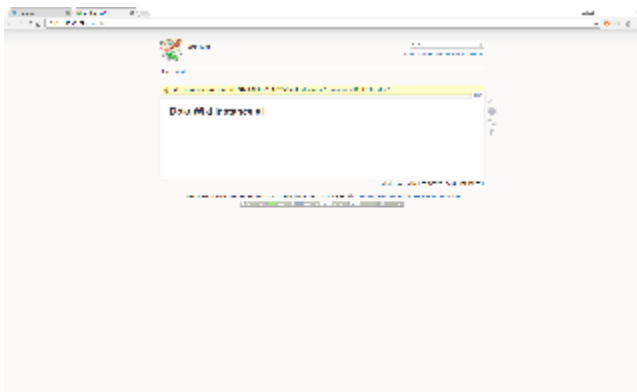
25. Add one phrase using Heading 1: DokuWiki Instance #1



26. Do the same for dokuwiki-server service from dokuwiki2 stack, just change the phrase to "DokuWiki Instance #2"



27. Go back to the load balancer URL and refresh the page several times; you should notice that each request is processed by a different DokuWiki instance, thus the load balancer is working and it's using the round-robin algorithm



Load Balancer for HTTP and HTTPS

This page details how to setup a load balancer to distribute HTTPS requests made against several instances of DokuWiki hosted on a running instance of Rancher server.

More information can be found [here](#) and [here](#).

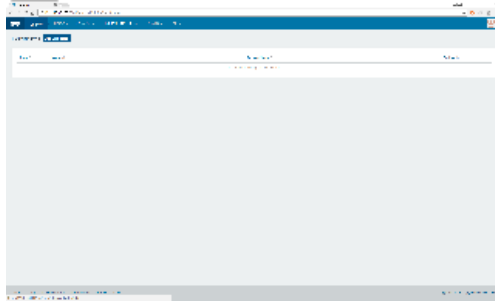
In order for the load balancer previously added via [Load Balancer for HTTP Only](#) to server HTTPS requests, a certificate must be added to Rancher server

1. Generate a certificate
 - a. Follow the steps described here: https://www.akadia.com/services/ssh_test_certificate.html or
 - b. Use the one provided by Docker Machine
 - i. On Windows, this certificate is found here: %USER%\docker\machine\certs\ca.pem, while its private key is found here: %USER%\docker\machine\certs\ca-key.pem

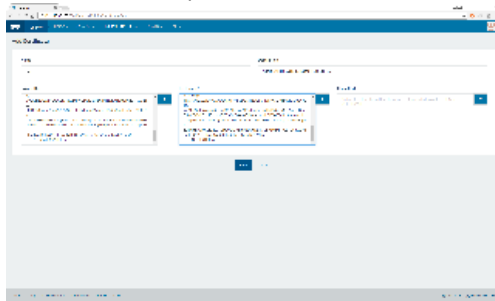
For production grade Rancher server, use VALID certificates!

For the purpose of this workshop, any certificate will do.

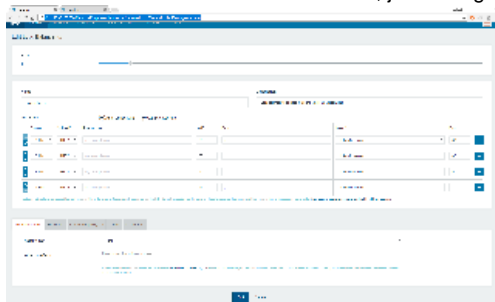
2. Go to Rancher UI
3. Go to menu Infrastructure Certificates
4. Click "Add Certificate" button



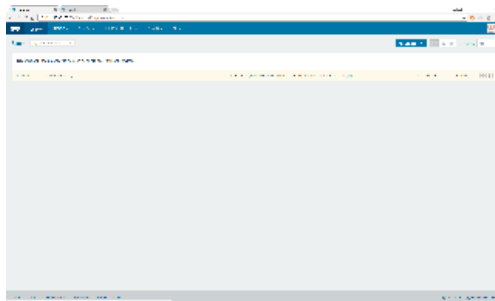
5. Fill in the field appropriately
 - a. Name: `tgd`
 - b. Description: Certificate to be used by the `tgd` load balancer.
 - c. Private Key: paste here the contents of the private key file
 - d. Certificate: paste here the contents of the certificate file



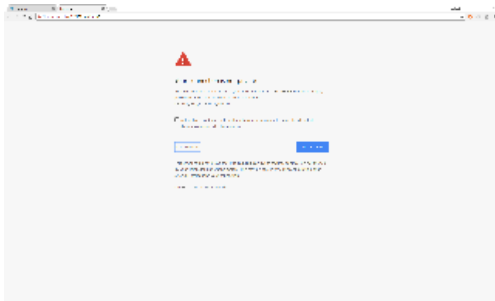
6. Click the "Save" button
7. Edit the **Load Balancer for HTTP Only** and add 2 new Service Rules, both using HTTPS protocol
 - a. Service Rule 1
 - i. Access: Public
 - ii. Protocol: HTTPS
 - iii. Request Host: leave empty
 - iv. Port: 443 (the default port used for serving content over HTTPS)
 - v. Path: /
 - vi. Target: `dokuwiki-server` from stack `dokuwiki1`
 - vii. Port: 80
 - b. Do the same for Service Rule 2, just change Target to `dokuwiki-server` from stack `dokuwiki2`



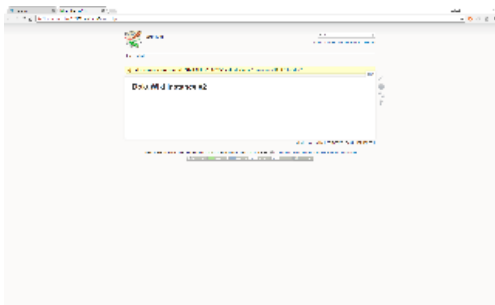
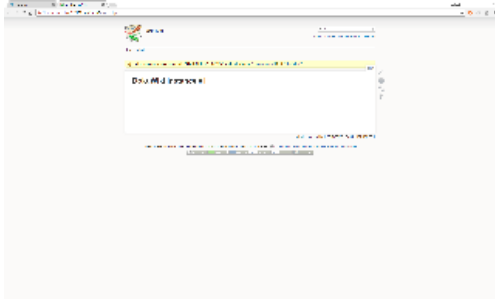
8. Click the "Edit" button
9. Go to the load balancer service and click the new "443/tcp" link from Ports field



10. In case the certificate leads to the browser displaying a warning, just ignore it (for production, the load balancer should be reachable via a domain name which must be included inside the certificate)



11. You should now see both DokuWiki instances over HTTPS

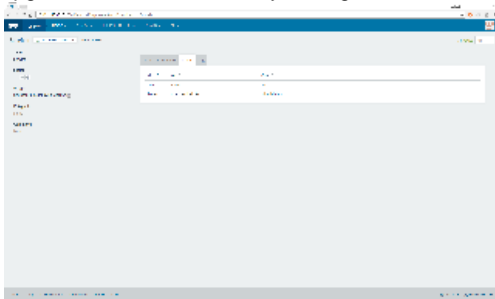


Selector Rules

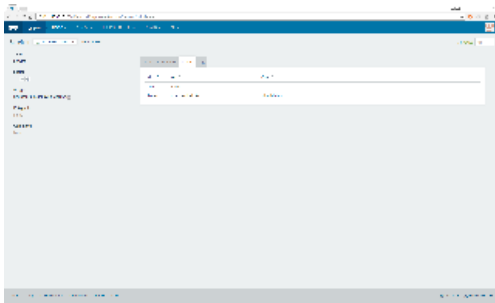
This page details how to setup a load balancer to distribute requests made against several instances of DokuWiki hosted on a running instance of Rancher server using labels attached to these instances.

More information can be found [here](#).

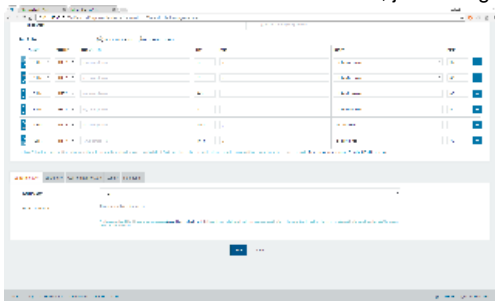
1. Go to Rancher UI
2. Upgrade service dokuwiki1 by adding one label: dokuwiki=blue



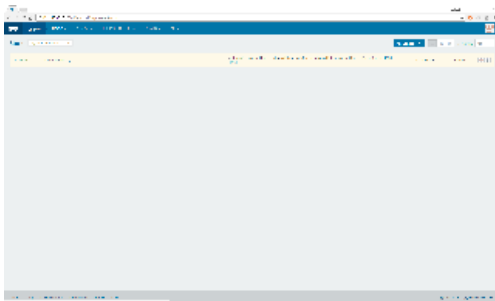
3. Upgrade service dokuwiki2 by adding one label: dokuwiki=red



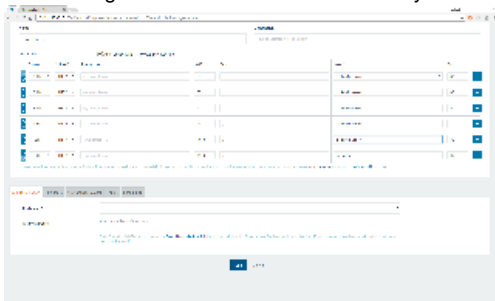
4. Edit the **Load Balancer for HTTP and HTTPS** and add 2 Selector Rules
 - a. Selector Rule 1
 - i. Access: Public
 - ii. Protocol: HTTPS
 - iii. Request Host: leave empty
 - iv. Port: 4443
 - v. Path: /
 - vi. Target: dokuwiki=blue
 - vii. Port: 80
 - b. Do the same for Selector Rule 2, just change Target to dokuwiki=red



5. Click the "Edit" button
6. Go to the load balancer service and click the new "4443/tcp" link from Ports field



7. You should now see both DokuWiki instances over HTTPS
8. Edit once again the load balancer and modify Selector Rule 1, by changing Target to dokuwiki=yellow



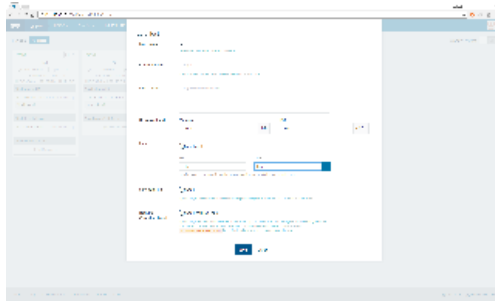
9. Click the "Edit" button
10. Go to the load balancer service and click the new "4443/tcp" link from Ports field
11. Try refreshing the page several times and you will notice DokuWiki Instance #1 no longer appears

04 - Scheduling

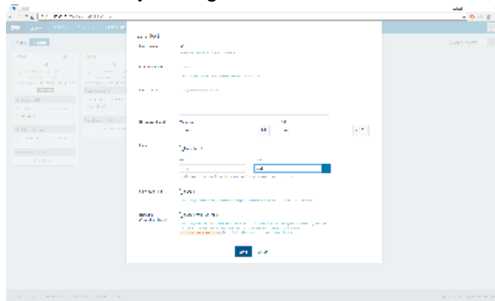
This page details how to ensure a container is run on a particular host by using labels.

More information can be found [here](#).

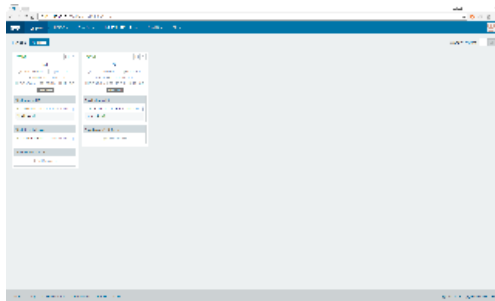
1. Go to Rancher UI
2. Go to menu Infrastructure Hosts
3. Edit host ra1 by adding a label: color=blue



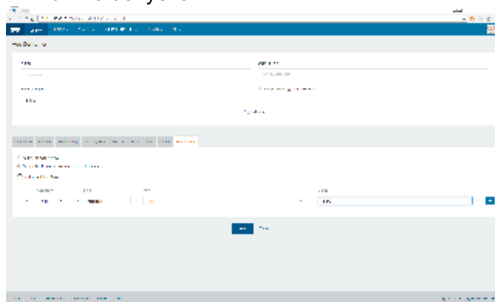
4. Edit host ra2 by adding a label: color=red



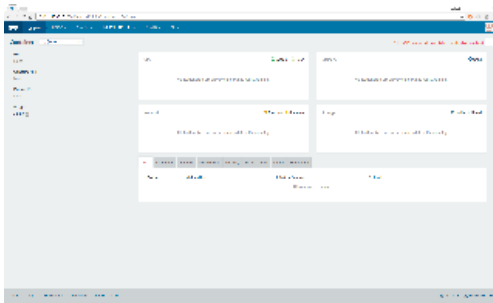
5. You should see one label on each host



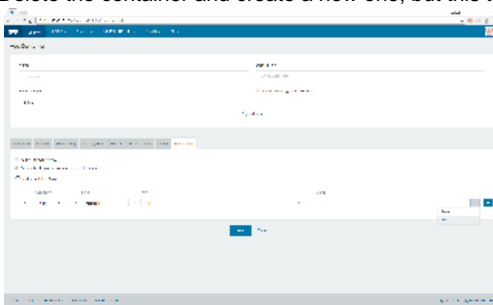
6. Go to menu Infrastructure Containers
7. Add container from Docker image alpine and inside Scheduling tab, fill in the fields of a Scheduling Rule
 - a. Condition: must
 - b. Field: host label
 - c. Key: color
 - d. Value: yellow



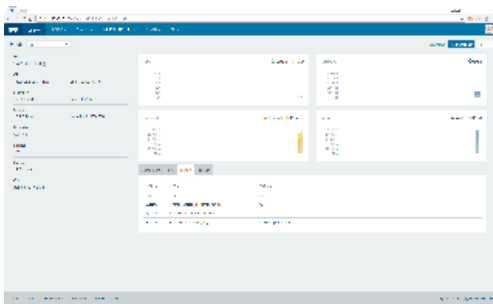
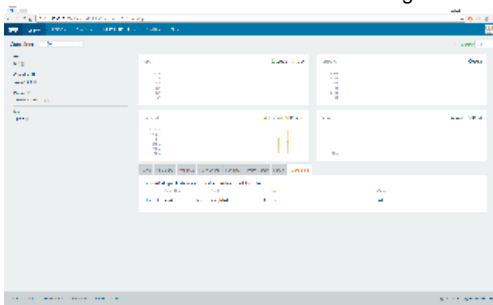
8. You should see an error



9. Delete the container and create a new one, but this time, set the color label to red



10. You should see that the container is running on host ra2, where color=red



05 - Use Rancher CLI

This page details how to use Rancher CLI against a running instance of Rancher server to perform CRUD tasks against its resources (e.g. environments, stacks, etc.).

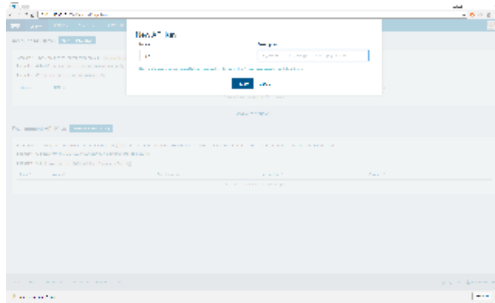
This tool can be used to automate things, so that they may be invoked by a CI server like Jenkins.

More information can be found [here](#) and [here](#).

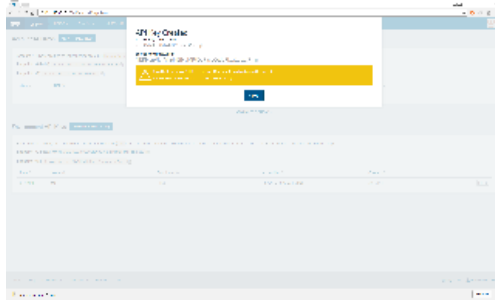
Make sure you visit the examples too:

- [Create and Destroy Stack & Services Using Rancher CLI](#)
- [Rancher CLI - Examples](#)

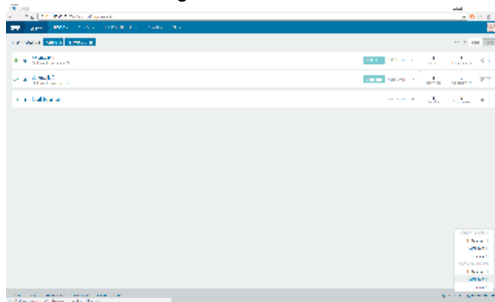
1. Go to Rancher UI
2. Go to menu API Keys
3. Click the Add Environment button and give the API key a meaningful name



4. Click the Create button and save the Access Key and Secret Key in a safe place for future references



5. Go to the bottom right link "Download CLI" and click the appropriate Rancher CLI download link



6. Unzip the downloaded file (rancher-windows-386-v0.6.3.zip) somewhere on your machine
7. Create a new Rancher environment

```
# Rancher environment is passed via --env <ENVIRONMENT_ID>
.\rancher.exe --url http://192.168.99.101:8080 --access-key
F830A6D384AF8C497D19 --secret-key
NERNbgVkmVhpHEfNyMRtDd9avbQzJePkJxldLxg7 env create newEnvironment
```

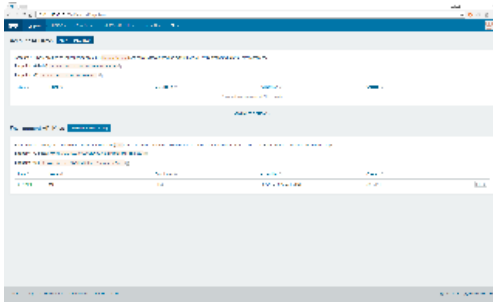
This command will fail with the following error message:

FATA[0000] Resource type [project] is not creatable

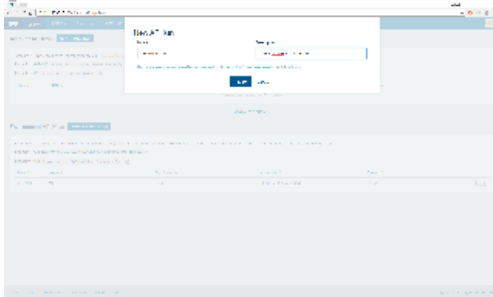
This happens since we're using an environment API key which grants access only to the resources of a specific environment.

In order to create a new environment, we need to use an Account API Key!

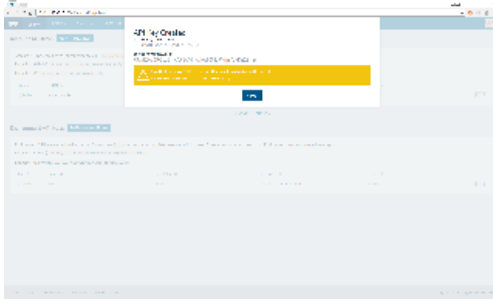
8. Go to menu, API Keys and click the Add Account API Key



9. Give the key a meaningful name and description and click the Create button



10. Click the Create button and save the Access Key and Secret Key in a safe place for future references



11. Re-run the command used for creating an environment and the result (the id of the newly created Rancher environment) should be printed inside the current terminal

```
.\rancher.exe --url http://192.168.99.101:8080 --access-key
80805BE0AC79342F72C6 --secret-key
PJygtZyy6RGcaBhKAF3cMmcU6dKMLFYfmYzvKq2S env create newEnvironment
# 1a28
```

12. As an exercise, use Rancher CLI commands for creating some of the Rancher objects you have previously created via Rancher UI

Create and Destroy Stack & Services Using Rancher CLI

1. Clone [rancher-workshop](#) git repository on your dev machine
2. Open a terminal from where you can run Rancher CLI commands
3. Create stack named "my-stack" in a given environment, and then start it (will use default values for catalog item)

```
./rancher --url http://192.168.99.100:8080 --access-key
9E27BDAEBC51FF4249D9 --secret-key
QWpmxggKe8toiCA123g2xddtAhhYnBH93slmupqC stacks create my-stack
--docker-compose
<RANCHER_WORKSHOP_HOME>/templates/alpine/0/docker-compose.yml
--rancher-compose
<RANCHER_WORKSHOP_HOME>/templates/alpine/0/rancher-compose.yml --start
# 1st15
```


[%RANCHER_URL%]	
--access-key value	Specify Rancher API access key
[%RANCHER_ACCESS_KEY%]	
--secret-key value	Specify Rancher API secret key
[%RANCHER_SECRET_KEY%]	
--host value	Host used for docker command
[%RANCHER_DOCKER_HOST%]	
--wait, -w	Wait for resource to reach resting state
--wait-timeout value	Timeout in seconds to wait (default: 600)
--wait-state value	State to wait for (active, healthy, etc)
--help, -h	show help
--version, -v	print the version

Commands:

catalog	Operations with catalogs
config	Setup client configuration
docker	Run docker CLI on a host
environment, env	Interact with environments
events, event	Displays resource change events
exec	Run a command on a container
export	Export configuration yaml for a stack as a tar archive or to local files
hosts, host	Operations on hosts
logs	Fetch the logs of a container
ps	Show services/containers
restart	Restart service, container
rm	Delete service, container, stack, host, volume,
secret	
run	Run services
scale	Set number of containers to run for a service
secrets, secret	Operations on secrets
ssh	SSH into host
stacks, stack	Operations on stacks
start, activate	Start or activate service, container, host, stack
stop, deactivate	Stop or deactivate service, container, host, stack
up	Bring all services up
volumes, volume	Operations on volumes
inspect	View details for service, container, host,
environment, stack, volume, secret	
wait	Wait for resources service, container, host, stack,
machine, projectTemplate	

help Shows a list of commands or help for one command

Run 'rancher COMMAND --help' for more information on a command.

Fetch All Hosts By Environment

▼ [Click here to expand...](#)

```
.\rancher.exe --url http://192.168.99.103:8080 --access-key
74FF92AEC4691F7D8AF4 --secret-key
KywlkfNSJZXrSk8nK1f93P8bhxNMaPDEGEpsRPcT hosts
# ID          HOSTNAME    STATE    CONTAINERS    IP          LABELS
DETAIL
# 1h3         ra2         active   11            192.168.99.103  color=red
# 1h4         ra1         active   11            192.168.99.102
color=blue
```

Fetch All Stacks By Environment

▼ [Click here to expand...](#)

```
.\rancher.exe --url http://192.168.99.103:8080 --access-key
74FF92AEC4691F7D8AF4 --secret-key
KywlkfNSJZXrSk8nK1f93P8bhxNMaPDEGEpsRPcT stacks
# ID          NAME          STATE    CATALOG
SERVICES      SYSTEM        DETAIL
# 1st9         dokuwiki1     healthy  catalog://community:dokuwiki:0    1
false
# 1st10        dokuwiki2     healthy  catalog://community:dokuwiki:0    1
false
# 1st11        weavescope    healthy  catalog://community:weavescope:0  2
false
```

Fetch Stack Details By Name

▼ [Click here to expand...](#)

```
.\rancher.exe --url http://192.168.99.103:8080 --access-key
74FF92AEC4691F7D8AF4 --secret-key
KywlkfNSJZXrSk8nK1f93P8bhxNMaPDEGEpsRPcT inspect dokuwiki1
```

▼ [Click here to expand...](#)

```
{
  "id": "1st9",
  "type": "stack",
  "links": {
    "self": ".../v2-beta/projects/1a16/stacks/1st9",
```

```

        "account": ".../v2-beta/projects/1a16/stacks/1st9/account",
        "hosts": ".../v2-beta/projects/1a16/stacks/1st9/hosts",
        "instances": ".../v2-beta/projects/1a16/stacks/1st9/instances",
        "scheduledUpgrades":
".../v2-beta/projects/1a16/stacks/1st9/scheduledupgrades",
        "secrets": ".../v2-beta/projects/1a16/stacks/1st9/secrets",
        "services": ".../v2-beta/projects/1a16/stacks/1st9/services",
        "volumeTemplates":
".../v2-beta/projects/1a16/stacks/1st9/volumetemplates",
        "volumes": ".../v2-beta/projects/1a16/stacks/1st9/volumes",
        "composeConfig":
".../v2-beta/projects/1a16/stacks/1st9/composeconfig"
    },
    "actions": {
        "deactivateservices":
".../v2-beta/projects/1a16/stacks/1st9/?action=deactivateservices",
        "activateservices":
".../v2-beta/projects/1a16/stacks/1st9/?action=activateservices",
        "upgrade":
".../v2-beta/projects/1a16/stacks/1st9/?action=upgrade",
        "addoutputs":
".../v2-beta/projects/1a16/stacks/1st9/?action=addoutputs",
        "update":
".../v2-beta/projects/1a16/stacks/1st9/?action=update",
        "exportconfig":
".../v2-beta/projects/1a16/stacks/1st9/?action=exportconfig",
        "remove": ".../v2-beta/projects/1a16/stacks/1st9/?action=remove"
    },
    "baseType": "stack",
    "name": "dokuwiki1",
    "state": "active",
    "accountId": "1a16",
    "answers": null,
    "binding": null,
    "created": "2017-08-23T18:01:30Z",
    "createdTS": 1503511290000,
    "description": "DokuWiki Instance #1",
    "dockerCompose": "dokuwiki-server:\n  ports:\n  -
${http_port}:80/tcp\n  labels:\n    io.rancher.sidekicks:
dokuwiki-data\n  hostname: ${dokuwiki_hostname}\n  image:
ununseptium/dokuwiki-docker\n  volumes_from:\n    - dokuwiki-data\n
\ndokuwiki-data:\n  labels:\n    io.rancher.container.start_once:
'true'\n  entrypoint:\n    - /bin/true\n  hostname: dokuwikidata\n
image: ununseptium/dokuwiki-docker\n  volumes:\n    -
/var/www/html/data\n    - /var/www/html/lib/plugins\n",
    "environment": {
        "dokuwiki_hostname": "dwl.example.com",
        "http_port": "8081"
    },
    "externalId": "catalog://community:dokuwiki:0",
    "group": null,
    "healthState": "healthy",
    "kind": "stack",

```

```
"outputs": null,
"previousEnvironment": null,
"previousExternalId": null,
"rancherCompose": ".catalog:\n  name: Dokuwiki\n  version: 2016-06-26a\n  description: |\n    DokuWiki is a simple to use and highly versatile Open Source wiki software that doesn't require a database.\n  minimum_rancher_version: v0.56.0\n  maintainer: \"Tim Kopplow <tim@arctium.io>\"\n  uuid: dokuwiki-0\n  questions:\n    - variable: \"dokuwiki_hostname\"\n      description: \"Dokuwiki hostname\"\n      label: \"Hostname\"\n      required: true\n      default: \"dw.example.com\"\n      type: \"string\"\n      - variable: \"http_port\"\n        description: \"HTTP port to expose on host. Will be used to bind TCP\"\n        label: \"HTTP port\"\n        required: true\n        default: 80\n        type: \"string\"\n      ndokuwiki-server:\n        scale: 1\n        retain_ip: true\n        health_check:\n          port: 80\n          interval: 30000\n          unhealthy_threshold: 3\n          strategy: recreate\n          response_timeout: 2000\n          healthy_threshold: 2\n      ndokuwiki-data:\n        scale: 1\n        retain_ip: true\n",
"removed": null,
"serviceIds": [
  "ls13"
],
"startOnCreate": true,
"system": false,
"templates": null,
"transitioning": "no",
"transitioningMessage": null,
```

```
"transitioningProgress": null,  
"uuid": "b66e5dec-7eb0-4202-9d83-bb0c5f1c97a3"  
}
```

Fetch Services By Environment

▼ [Click here to expand...](#)

```
# Rancher environment is passed via --env <ENVIRONMENT_ID>  
.\rancher.exe --url http://192.168.99.103:8080 --access-key  
74FF92AEC4691F7D8AF4 --secret-key  
Kw1kfNSJZXrSk8nK1f93P8bhxNMaPDEGEpsRPcT --env 1a7 ps  
  
# Command output  
# ID          TYPE          NAME  
IMAGE          STATE    SCALE    SYSTEM  
ENDPOINTS    DETAIL  
# 1s13        service      dokuwiki1/dokuwiki-server  
ununseptium/dokuwiki-docker    healthy    2/1      false  
# 1s13        sidekick  
dokuwiki1/dokuwiki-server/dokuwiki-data    ununseptium/dokuwiki-docker  
healthy    2/1      false  
# 1s14        service      dokuwiki2/dokuwiki-server  
ununseptium/dokuwiki-docker    healthy    2/1      false  
# 1s14        sidekick  
dokuwiki2/dokuwiki-server/dokuwiki-data    ununseptium/dokuwiki-docker  
healthy    2/1      false  
# 1s15        loadBalancerService    load-balancer/load-balancer  
rancher/lb-service-haproxy:v0.7.6    healthy    1/1      false
```

Fetch Service Details By Name

▼ [Click here to expand...](#)

```
.\rancher.exe --url http://192.168.99.103:8080 --access-key  
74FF92AEC4691F7D8AF4 --secret-key  
Kw1kfNSJZXrSk8nK1f93P8bhxNMaPDEGEpsRPcT inspect  
dokuwiki1/dokuwiki-server
```

▼ [Click here to expand...](#)

```
{  
  "id": "1s13",  
  "type": "service",  
  "links": {  
    "self": ".../v2-beta/projects/1a16/services/1s13",  
    "account": ".../v2-beta/projects/1a16/services/1s13/account",  
    "consumedbyservices":  
      ".../v2-beta/projects/1a16/services/1s13/consumedbyservices",  
    "consumedservices":
```



```

"/v2-beta/projects/1a16/services/1s13/consumedservices",
    "instances":
"/v2-beta/projects/1a16/services/1s13/instances",
    "networkDrivers":
"/v2-beta/projects/1a16/services/1s13/networkdrivers",
    "serviceExposeMaps":
"/v2-beta/projects/1a16/services/1s13/serviceexposemaps",
    "serviceLogs":
"/v2-beta/projects/1a16/services/1s13/servicelogs",
    "stack": "/v2-beta/projects/1a16/services/1s13/stack",
    "storageDrivers":
"/v2-beta/projects/1a16/services/1s13/storagedrivers",
    "containerStats":
"/v2-beta/projects/1a16/services/1s13/containerstats"
},
    "actions": {
        "upgrade":
"/v2-beta/projects/1a16/services/1s13/?action=upgrade",
        "restart":
"/v2-beta/projects/1a16/services/1s13/?action=restart",
        "update":
"/v2-beta/projects/1a16/services/1s13/?action=update",
        "remove":
"/v2-beta/projects/1a16/services/1s13/?action=remove",
        "deactivate":
"/v2-beta/projects/1a16/services/1s13/?action=deactivate",
        "removeservicelink":
"/v2-beta/projects/1a16/services/1s13/?action=removeservicelink",
        "addservicelink":
"/v2-beta/projects/1a16/services/1s13/?action=addservicelink",
        "setservicelinks":
"/v2-beta/projects/1a16/services/1s13/?action=setservicelinks"
},
    "baseType": "service",
    "name": "dokuwiki-server",
    "state": "active",
    "accountId": "1a16",
    "assignServiceIpAddress": false,
    "createIndex": 6,
    "created": "2017-08-23T18:01:30Z",
    "createdTS": 1503511290000,
    "currentScale": 1,
    "description": null,
    "externalId": null,
    "fqdn": null,
    "healthState": "healthy",
    "instanceIds": [
        "1i35",
        "1i47"
    ],
    "kind": "service",
    "launchConfig": {
        "type": "launchConfig",

```

```
"blkioWeight": null,
"capAdd": [],
"capDrop": [],
"cgroupParent": null,
"count": null,
"cpuCount": null,
"cpuPercent": null,
"cpuPeriod": null,
"cpuQuota": null,
"cpuSet": null,
"cpuSetMems": null,
"cpuShares": null,
"dataVolumes": [],
"dataVolumesFrom": [],
"description": null,
"devices": [],
"diskQuota": null,
"dns": [],
"dnsSearch": [],
"domainName": null,
"healthCheck": {
    "type": "instanceHealthCheck",
    "healthyThreshold": 2,
    "initializingTimeout": null,
    "interval": 30000,
    "name": null,
    "port": 80,
    "reinitializingTimeout": null,
    "requestLine": null,
    "responseTimeout": 2000,
    "strategy": "recreate",
    "unhealthyThreshold": 3
},
"healthInterval": null,
"healthRetries": null,
"healthTimeout": null,
"hostname": "dwl.example.com",
"imageUuid": "docker:ununseptium/dokuwiki-docker",
"instanceTriggeredStop": "stop",
"ioMaximumBandwidth": null,
"ioMaximumIOps": null,
"ip": null,
"ip6": null,
"ipcMode": null,
"isolation": null,
"kernelMemory": null,
"kind": "container",
"labels": {
    "io.rancher.sidekicks": "dokuwiki-data",
    "dokuwiki": "blue"
},
"logConfig": {
    "type": "logConfig",
```

```

        "config": {},
        "driver": null
    },
    "memory": null,
    "memoryMb": null,
    "memoryReservation": null,
    "memorySwap": null,
    "memorySwappiness": null,
    "milliCpuReservation": null,
    "networkMode": "managed",
    "oomScoreAdj": null,
    "pidMode": null,
    "pidsLimit": null,
    "ports": [
        "8081:80/tcp"
    ],
    "privileged": false,
    "publishAllPorts": false,
    "readOnly": false,
    "requestedIpAddress": null,
    "runInit": false,
    "secrets": [],
    "shmSize": null,
    "startOnCreate": true,
    "stdinOpen": false,
    "stopSignal": null,
    "system": false,
    "tty": false,
    "user": null,
    "userdata": null,
    "usersnsMode": null,
    "uts": null,
    "version": "19d935c8-cd4a-4683-81e9-41ae8d93212f",
    "volumeDriver": null,
    "workingDir": null,
    "dataVolumesFromLaunchConfigs": [
        "dokuwiki-data"
    ],
    "networkLaunchConfig": null,
    "vcpu": 1
},
"lbConfig": null,
"linkedServices": null,
"metadata": {
    "io.rancher.service.hash":
"440add56ce917c9d318510d271fe1d12019f6550"
},
"publicEndpoints": [
    {
        "type": "publicEndpoint",
        "hostId": "lh4",
        "instanceId": "li47",
        "ipAddress": "192.168.99.102",

```

```

        "port": 8081,
        "serviceId": "ls13"
    }
],
"removed": null,
"retainIp": true,
"scale": 1,
"scalePolicy": null,
"secondaryLaunchConfigs": [
    {
        "type": "secondaryLaunchConfig",
        "dataVolumes": [
            "/var/www/html/data",
            "/var/www/html/lib/plugins"
        ],
        "entryPoint": [
            "/bin/true"
        ],
        "hostname": "dokuwikidata",
        "imageUuid": "docker:ununseptium/dokuwiki-docker",
        "instanceTriggeredStop": "stop",
        "kind": "container",
        "labels": {
            "io.rancher.container.start_once": "true",
            "io.rancher.service.hash":
"1c19dc96c1a9666e6e7f7939914145cc12cd0a34"
        },
        "logConfig": {
            "type": "logConfig"
        },
        "name": "dokuwiki-data",
        "networkMode": "managed",
        "privileged": false,
        "publishAllPorts": false,
        "readOnly": false,
        "runInit": false,
        "startOnCreate": true,
        "stdinOpen": false,
        "system": false,
        "tty": false,
        "vcpu": 1,
        "version": "0"
    }
],
"selectorContainer": null,
"selectorLink": null,
"stackId": "1st9",
"startOnCreate": false,
"system": false,
"transitioning": "no",
"transitioningMessage": null,
"transitioningProgress": null,
"upgrade": {

```

```
"type": "serviceUpgrade",
"inServiceStrategy": {
  "type": "inServiceUpgradeStrategy",
  "batchSize": 1,
  "intervalMillis": 2000,
  "launchConfig": {
    "type": "launchConfig",
    "blkioWeight": null,
    "capAdd": [],
    "capDrop": [],
    "cgroupParent": null,
    "count": null,
    "cpuCount": null,
    "cpuPercent": null,
    "cpuPeriod": null,
    "cpuQuota": null,
    "cpuSet": null,
    "cpuSetMems": null,
    "cpuShares": null,
    "dataVolumes": [],
    "dataVolumesFrom": [],
    "description": null,
    "devices": [],
    "diskQuota": null,
    "dns": [],
    "dnsSearch": [],
    "domainName": null,
    "healthCheck": {
      "type": "instanceHealthCheck",
      "healthyThreshold": 2,
      "initializingTimeout": null,
      "interval": 30000,
      "name": null,
      "port": 80,
      "reinitializingTimeout": null,
      "requestLine": null,
      "responseTimeout": 2000,
      "strategy": "recreate",
      "unhealthyThreshold": 3
    },
    "healthInterval": null,
    "healthRetries": null,
    "healthTimeout": null,
    "hostname": "dwl.example.com",
    "imageUuid": "docker:ununseptium/dokuwiki-docker",
    "instanceTriggeredStop": "stop",
    "ioMaximumBandwidth": null,
    "ioMaximumIOps": null,
    "ip": null,
    "ip6": null,
    "ipcMode": null,
    "isolation": null,
    "kernelMemory": null,
```

```
"kind": "container",
"labels": {
  "io.rancher.sidekicks": "dokuwiki-data",
  "dokuwiki": "blue"
},
"logConfig": {
  "type": "logConfig",
  "config": {},
  "driver": null
},
"memory": null,
"memoryMb": null,
"memoryReservation": null,
"memorySwap": null,
"memorySwappiness": null,
"milliCpuReservation": null,
"networkMode": "managed",
"oomScoreAdj": null,
"pidMode": null,
"pidsLimit": null,
"ports": [
  "8081:80/tcp"
],
"privileged": false,
"publishAllPorts": false,
"readOnly": false,
"requestedIpAddress": null,
"runInit": false,
"secrets": [],
"shmSize": null,
"startOnCreate": true,
"stdinOpen": false,
"stopSignal": null,
"system": false,
"tty": false,
"user": null,
"userdata": null,
"usersnsMode": null,
"uts": null,
"version": "19d935c8-cd4a-4683-81e9-41ae8d93212f",
"volumeDriver": null,
"workingDir": null,
"dataVolumesFromLaunchConfigs": [
  "dokuwiki-data"
],
"networkLaunchConfig": null,
"vcpu": 1
},
"previousLaunchConfig": {
  "type": "launchConfig",
  "blkioWeight": null,
  "capAdd": [],
  "capDrop": [],
```

```
"cgroupParent": null,
"count": null,
"cpuCount": null,
"cpuPercent": null,
"cpuPeriod": null,
"cpuQuota": null,
"cpuSet": null,
"cpuSetMems": null,
"cpuShares": null,
"dataVolumes": [],
"dataVolumesFrom": [],
"description": null,
"devices": [],
"diskQuota": null,
"dns": [],
"dnsSearch": [],
"domainName": null,
"healthCheck": {
  "type": "instanceHealthCheck",
  "healthyThreshold": 2,
  "initializingTimeout": null,
  "interval": 30000,
  "name": null,
  "port": 80,
  "reinitializingTimeout": null,
  "requestLine": null,
  "responseTimeout": 2000,
  "strategy": "recreate",
  "unhealthyThreshold": 3
},
"healthInterval": null,
"healthRetries": null,
"healthTimeout": null,
"hostname": "dwl.example.com",
"imageUuid": "docker:ununseptium/dokuwiki-docker",
"instanceTriggeredStop": "stop",
"ioMaximumBandwidth": null,
"ioMaximumIOps": null,
"ip": null,
"ip6": null,
"ipcMode": null,
"isolation": null,
"kernelMemory": null,
"kind": "container",
"labels": {
  "io.rancher.sidekicks": "dokuwiki-data",
  "dokuwiki-instance": "1"
},
"logConfig": {
  "type": "logConfig",
  "config": {},
  "driver": null
},
```

```

    "memory": null,
    "memoryMb": null,
    "memoryReservation": null,
    "memorySwap": null,
    "memorySwappiness": null,
    "milliCpuReservation": null,
    "networkMode": "managed",
    "oomScoreAdj": null,
    "pidMode": null,
    "pidsLimit": null,
    "ports": [
        "8081:80/tcp"
    ],
    "privileged": false,
    "publishAllPorts": false,
    "readOnly": false,
    "requestedIpAddress": null,
    "runInit": false,
    "secrets": [],
    "shmSize": null,
    "startOnCreate": true,
    "stdinOpen": false,
    "stopSignal": null,
    "system": false,
    "tty": false,
    "user": null,
    "userdata": null,
    "usernsMode": null,
    "uts": null,
    "version": "31b0b60f-80c9-4451-a337-97bd3c0fe66c",
    "volumeDriver": null,
    "workingDir": null,
    "dataVolumesFromLaunchConfigs": [
        "dokuwiki-data"
    ],
    "networkLaunchConfig": null,
    "vcpu": 1
},
"previousSecondaryLaunchConfigs": [
    {
        "type": "secondaryLaunchConfig",
        "dataVolumes": [
            "/var/www/html/data",
            "/var/www/html/lib/plugins"
        ],
        "entryPoint": [
            "/bin/true"
        ],
        "hostname": "dokuwikidata",
        "imageUuid": "docker:ununseptium/dokuwiki-docker",
        "instanceTriggeredStop": "stop",
        "kind": "container",
        "labels": {

```



```

        "io.rancher.container.start_once": "true",
        "io.rancher.service.hash":
"1c19dc96c1a9666e6e7f7939914145cc12cd0a34"
    },
    "logConfig": {
        "type": "logConfig"
    },
    "name": "dokuwiki-data",
    "networkMode": "managed",
    "privileged": false,
    "publishAllPorts": false,
    "readOnly": false,
    "runInit": false,
    "startOnCreate": true,
    "stdinOpen": false,
    "system": false,
    "tty": false,
    "vcpu": 1,
    "version": "0"
  }
],
"secondaryLaunchConfigs": [
  {
    "type": "secondaryLaunchConfig",
    "dataVolumes": [
      "/var/www/html/data",
      "/var/www/html/lib/plugins"
    ],
    "entryPoint": [
      "/bin/true"
    ],
    "hostname": "dokuwikidata",
    "imageUuid": "docker:ununseptium/dokuwiki-docker",
    "instanceTriggeredStop": "stop",
    "kind": "container",
    "labels": {
      "io.rancher.container.start_once": "true",
      "io.rancher.service.hash":
"1c19dc96c1a9666e6e7f7939914145cc12cd0a34"
    },
    "logConfig": {
      "type": "logConfig"
    },
    "name": "dokuwiki-data",
    "networkMode": "managed",
    "privileged": false,
    "publishAllPorts": false,
    "readOnly": false,
    "runInit": false,
    "startOnCreate": true,
    "stdinOpen": false,
    "system": false,
    "tty": false,

```

```
        "vcpu": 1,  
        "version": "0"  
    }  
    ],  
    "startFirst": false  
},  
"toServiceStrategy": null  
},
```

```
"uuid": "b4cf33a7-7cad-4507-bdce-210201741d41",  
"vip": null  
}
```

Fetch Service ID By Name

▼ [Click here to expand...](#)

```
.\rancher.exe --url http://192.168.99.101:8080 --access-key  
F830A6D384AF8C497D19 --secret-key  
NERNbgVkmVhpHEfNyMRtDd9avbQzJePkJxldLxg7 inspect --format '{{.id}}'  
dokuwiki1/dokuwiki-server  
# ls13
```

Fetch Service Name By ID

▼ [Click here to expand...](#)

```
.\rancher.exe --url http://192.168.99.101:8080 --access-key  
F830A6D384AF8C497D19 --secret-key  
NERNbgVkmVhpHEfNyMRtDd9avbQzJePkJxldLxg7 inspect --format '{{.name}}'  
ls13  
# dokuwiki-server
```

06 - Use Rancher Compose

This page details how to use Rancher Compose against a running instance of Rancher server.

Rancher Compose tool is a multi-host version of Docker Compose; it operates in the context of a stack.

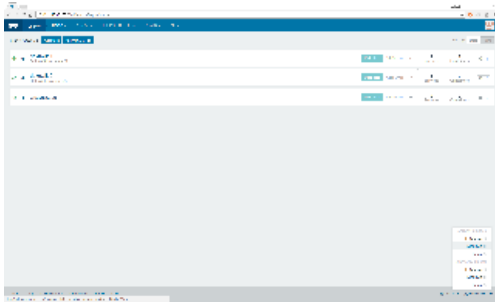
More information can be found [here](#) and [here](#).

1. Clone [rancher-workshop](#) git repository on your dev machine, if you haven't done so already
2. Create a file, **answers.txt**, under **<RANCHER_WORKSHOP_HOME>\templates\alpine\0** folder

```
<RANCHER_WORKSHOP_HOME>\templates\alpine\0\answers.txt
```

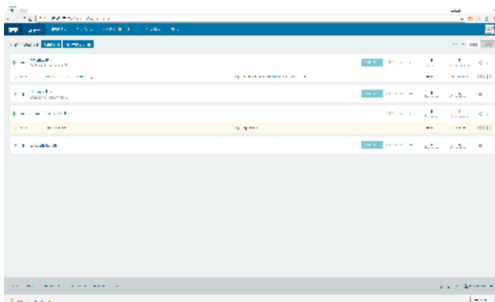
```
options_enum=Group  
options_boolean=false  
options_string=Custom stack created using Rancher Compose tool
```

3. Go to Rancher UI
4. Go to the bottom right link "Download CLI" and click the appropriate Rancher Compose download link



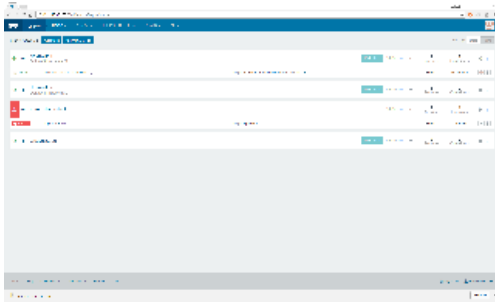
5. Unzip the downloaded file (rancher-compose-windows-386-v0.12.5.zip) somewhere on your machine
6. Open a terminal from where you can run Rancher Compose commands
7. Create stack named "my-stack" in a given environment, and then start it

```
./rancher-compose.exe --url http://192.168.99.103:8080 --access-key
74FF92AEC4691F7D8AF4 --secret-key
KywlkfNSJZXRsk8nK1f93P8bhxNMaPDEGEpsRPcT --file
<RANCHER_WORKSHOP_HOME>\templates\alpine\0\docker-compose.yml
--rancher-file
<RANCHER_WORKSHOP_HOME>\templates\alpine\0\rancher-compose.yml
--env-file <RANCHER_WORKSHOP_HOME>\templates\alpine\0\answers.txt
--project-name my-stack up -d
# INFO[0000] Creating stack my-custom-stack
# INFO[0000] [0/1] [alpine-3-5]: Creating
# INFO[0000] Creating service alpine-3-5
# INFO[0000] [0/1] [alpine-3-5]: Created
# INFO[0000] [0/1] [alpine-3-5]: Starting
# INFO[0004] [1/1] [alpine-3-5]: Started
```



8. Stop stack "my-stack"

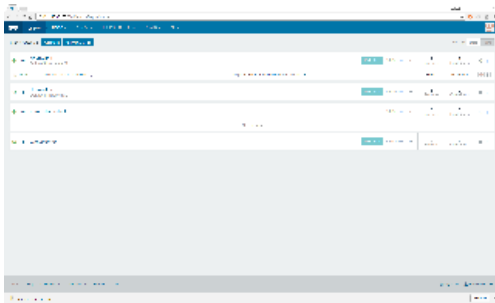
```
.\rancher-compose.exe --url http://192.168.99.103:8080 --access-key
74FF92AEC4691F7D8AF4 --secret-key
KywlkfNSJZXRsk8nK1f93P8bhxNMaPDEGEpsRPcT --file
<RANCHER_WORKSHOP_HOME>\templates\alpine\0\docker-compose.yml
rancher-file
<RANCHER_WORKSHOP_HOME>\templates\alpine\0\rancher-compose.yml
--env-file <RANCHER_WORKSHOP_HOME>\templates\alpine\0\answers.txt
--project-name my-custom-stack stop
# INFO[0000] [0/1] [alpine-3-5]: Stopping
# INFO[0000] [0/1] [alpine-3-5]: Stopped
```



9. Remove all services from stack "my-stack"

This command will not remove the stack, only its services!

```
.\rancher-compose.exe --url http://192.168.99.103:8080 --access-key  
74FF92AEC4691F7D8AF4 --secret-key  
KywlkfNSJZXrSk8nK1f93P8bhxNMaPDEGEpsRPcT --file  
<RANCHER_WORKSHOP_HOME>\templates\alpine\0\docker-compose.yml  
rancher-file  
<RANCHER_WORKSHOP_HOME>\templates\alpine\0\rancher-compose.yml  
--env-file <RANCHER_WORKSHOP_HOME>\templates\alpine\0\answers.txt  
--project-name my-custom-stack rm  
# INFO[0000] [0/1] [alpine-3-5]: Deleting  
# INFO[0000] [0/1] [alpine-3-5]: Deleted
```



07 - Use Web API

This page details how to use Rancher Web API in order to perform CRUD operations against all server resources.

Rancher currently supports 2 API versions: v1 and v2-beta.

More information can be found [here](#) (v2-beta) and [here](#) (v1).

- [Fetch Stacks by Environment](#)
- [Fetch Stack](#)

Fetch Stacks by Environment

v2-beta v1

▼ [Click here to expand...](#)

```
curl -u 74FF92AEC4691F7D8AF4:KywlkfNSJZXrSk8nK1f93P8bhxNMaPDEGEpsRPcT  
http://192.168.99.103:8080/v2-beta/projects/1a7/stacks
```

```
{
  "type": "collection",
  "resourceType": "stack",
  "links": {

"self": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks"
  },
  "createTypes": {

"composeProject": "http://192.168.99.103:8080/v2-beta/projects/1a7/composeprojects",

"kubernetesStack": "http://192.168.99.103:8080/v2-beta/projects/1a7/kubernetesstacks",

"stack": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks"
  },
  "actions": {

  },
  "data": [
    {
      "id": "1st5",
      "type": "stack",
      "links": {

"self": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st5",

"account": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st5/account",

"hosts": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st5/hosts",

"instances": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st5/instances",

"scheduledUpgrades": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st5/scheduledupgrades",

"secrets": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st5/secrets",

"services": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st5/services",

"volumeTemplates": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st5/volumetemplates",

"volumes": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st5/volumes",
```

```

"composeConfig": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st5/composeconfig"
    },
    "actions": {

"deactivateservices": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st5/?action=deactivateservices",

"activateservices": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st5/?action=activateservices",

"upgrade": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st5/?action=upgrade",

"addoutputs": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st5/?action=addoutputs",

"update": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st5/?action=update",

"exportconfig": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st5/?action=exportconfig",

"remove": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st5/?action=remove"
    },
    "baseType": "stack",
    "name": "healthcheck",
    "state": "active",
    "accountId": "1a7",
    "answers": null,
    "binding": null,
    "created": "2017-08-27T13:15:48Z",
    "createdTS": 1503839748000,
    "description": null,
    "dockerCompose": "version: '2'\nservices:\n  healthcheck:\nlabels:\n  io.rancher.container.create_agent: 'true'\n  io.rancher.scheduler.global: 'true'\n  image:\n  rancher/healthcheck:v0.3.1\n  logging:\n  driver: json-file\n  options:\n    max-size: 25m\n    max-file: '2'\n",
    "environment": null,
    "externalId": "catalog://library:infra*healthcheck:3",
    "group": null,
    "healthState": "healthy",
    "kind": "stack",
    "outputs": null,
    "previousEnvironment": null,
    "previousExternalId": null,
    "rancherCompose": ".catalog:\n  name: \"Healthcheck Service\"\n  version: \"v0.3.0\"\n  description: \"Rancher healthcheck service\"\n  minimum_rancher_version: v1.6.0-rc1\n  healthcheck:\n    health_check:\n      port: 42\n      interval: 2000\n"

```

```
response_timeout: 2000\n      unhealthy_threshold: 3\nhealthy_threshold: 2\n",
    "removed": null,
    "serviceIds": [
        "ls12"
    ],
    "startOnCreate": null,
    "system": true,
    "templates": null,
    "transitioning": "no",
    "transitioningMessage": null,
    "transitioningProgress": null,
    "uuid": "baf8b05-3ed6-4dee-b967-f38bd603abb6"
},
{
    "id": "1st6",
    "type": "stack",
    "links": {

"self": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st6",

"account": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st6/account",

"hosts": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st6/hosts",

"instances": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st6/instances",

"scheduledUpgrades": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st6/scheduledupgrades",

"secrets": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st6/secrets",

"services": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st6/services",

"volumeTemplates": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st6/volumetemplates",

"volumes": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st6/volumes",

"composeConfig": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st6/composeconfig"
    },
    "actions": {

"deactivateservices": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st6/?action=deactivateservices",
```



```

"activateservices":"http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st6/?action=activateservices",

"upgrade":"http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st6/?action=upgrade",

"addoutputs":"http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st6/?action=addoutputs",

"update":"http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st6/?action=update",

"exportconfig":"http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st6/?action=exportconfig",

"remove":"http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st6/?action=remove"
},
"baseType":"stack",
"name":"ipsec",
"state":"active",
"accountId":"1a7",
"answers":null,
"binding":null,
"created":"2017-08-27T13:15:48Z",
"createdTS":1503839748000,
"description":null,
"dockerCompose":"version: '2'\n
services:\n
  ipsec:\n
    # IMPORTANT!!!! DO NOT CHANGE VERSION ON UPGRADE\n
    image: rancher/net:holder\n
    command: sh -c \"echo Refer to router sidekick for logs; mkfifo f; exec cat f\"\n
    network_mode: ipsec\n
    ports:\n
      - 500:500/udp\n
      - 4500:4500/udp\n
    labels:\n
      io.rancher.sidekicks: router\n
      io.rancher.scheduler.global: 'true'\n
      io.rancher.cni.link_mtu_overhead: '0'\n
      io.rancher.network.macsync: 'true'\n
      io.rancher.network.arpsync: 'true'\n
      router:\n
        cap_add:\n
          - NET_ADMIN\n
        image: rancher/net:v0.11.7\n
        network_mode: container:ipsec\n
        environment:\n
          RANCHER_DEBUG: '${RANCHER_DEBUG}'\n
        labels:\n
          io.rancher.container.create_agent: 'true'\n
          io.rancher.container.agent_service.ipsec: 'true'\n
          logging:\n
            driver: json-file\n
            options:\n
              max-size: 25m\n
            max-file: '2'\n
            sysctls:\n
              net.ipv4.conf.all.send_redirects: '0'\n
              net.ipv4.conf.default.send_redirects: '0'\n
              net.ipv4.conf.eth0.send_redirects: '0'\n
              net.ipv4.xfrm4_gc_thresh: '2147483647'\n
        cni-driver:\n
          privileged: true\n
        image: rancher/net:v0.11.7\n
        command: sh -c \"touch /var/log/rancher-cni.log && exec tail ---disable-inotify -F /var/log/rancher-cni.log\"\n
        network_mode: host\n
        pid: host\n
        labels:\n
          io.rancher.scheduler.global: 'true'\n
          io.rancher.network.cni.binary: 'rancher-bridge'\n
          io.rancher.container.dns: 'true'\n
          logging:\n
            driver: json-file\n
            options:\n
              max-size: 25m\n
            max-file:

```

```

'2'\n    network_driver:\n        name: Rancher IPsec\n
default_network:\n        name: ipsec\n        host_ports: {{
.Values.HOST_PORTS }}\n        subnets:\n            - network_address:
$SUBNET\n            dns:\n            - 169.254.169.250\n            dns_search:\n
- rancher.internal\n        cni_config:\n            '10-rancher.conf':\n
name: rancher-cni-network\n            type: rancher-bridge\n
bridge: $DOCKER_BRIDGE\n            bridgeSubnet: $SUBNET\n
logToFile: \var\log\rancher-cni.log\n            isDebugLevel:
${RANCHER_DEBUG}\n            isDefaultGateway: true\n            hostNat:
true\n            hairpinMode: {{ .Values.RANCHER_HAIRPIN_MODE }}\n
promiscMode: {{ .Values.RANCHER_PROMISCUOUS_MODE }}\n            mtu:
${MTU}\n            linkMTUOverhead: 98\n            ipam:\n
type: rancher-cni-ipam\n            subnetPrefixSize: \/{
.Values.SUBNET_PREFIX }}\n            logToFile:
\var\log\rancher-cni.log\n            isDebugLevel:
${RANCHER_DEBUG}\n            routes:\n            - dst:
169.254.169.250\32\n",
        "environment":null,
        "externalId":"catalog:\\\\library:infra*ipsec:11",
        "group":null,
        "healthState":"healthy",
        "kind":"stack",
        "outputs":null,
        "previousEnvironment":null,
        "previousExternalId":null,
        "rancherCompose": ".catalog:\n    name: \"Rancher IPsec\"\n\n
version: \"0.1.3\"\n    minimum_rancher_version: v1.6.6-rc1\n
questions:\n    - variable: \"DOCKER_BRIDGE\"\n        label: \"Docker
Bridge\"\n        description: \"Name of Docker Bridge. Default is
`docker0`\"\n        type: \"string\"\n        default: \"docker0\"\n
required: true\n    - variable: MTU\n        label: \"MTU for the
network\"\n        description: \"Adjust the MTU for the network,
according to your needs. Ex: GCE(1460), AWS(1500), etc\"\n
required: true\n        default: 1500\n        type: int\n        - variable:
\"SUBNET\"\n        label: \"Subnet\"\n        description: \"The subnet to
use for the managed IPSEC network.\"\n        type: \"string\"\n
default: '10.42.0.0\16'\n        required: true\n        - variable:
\"SUBNET_PREFIX\"\n        label: \"IPAM Subnet Prefix\"\n
description: \"The Subnet prefix to use for the managed network. For
most users this is same as the bridge subnet prefix. Ex: 16 if using
10.42.0.0\16 for bridge subnet.\"\n        type: int\n        default: 16\n
required: true\n        - variable: \"RANCHER_DEBUG\"\n        label: \"Enable
Debug Logs\"\n        description: \"This will enable very verbose debug
logs.\"\n        type: \"boolean\"\n        default: \"false\"\n
required: true\n        - variable: \"RANCHER_HAIRPIN_MODE\"\n        label:
\"Enable Hairpin mode\"\n        description: \"If this is enabled,
Promiscuous mode needs to be disabled.\"\n        type: \"boolean\"\n
default: \"false\"\n        required: true\n        - variable:
\"RANCHER_PROMISCUOUS_MODE\"\n        label: \"Enable Promiscuous mode on
the bridge\"\n        description: \"If this is enabled, Hairpin mode
needs to be disabled.\"\n        type: \"boolean\"\n        default:
\"true\"\n        required: true\n        - variable: \"HOST_PORTS\"\n
label: \"Enable Host Ports\"\n        description: \"Flag to

```

```

enable\/disable publishing the ports on the hosts\"\\n      type:
\"boolean\"\\n      default: \"true\"\\n      required: true\\n\",
    \"removed\":null,
    \"serviceIds\":[
        \"1s7\",
        \"1s9\"
    ],
    \"startOnCreate\":null,
    \"system\":true,
    \"templates\":null,
    \"transitioning\":\"no\",
    \"transitioningMessage\":null,
    \"transitioningProgress\":null,
    \"uuid\":\"47604724-1a59-4bb5-97ba-e04fe14e9b21\"
},
{
    \"id\":\"1st7\",
    \"type\":\"stack\",
    \"links\":{

\"self\":\"http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st7\",

\"account\":\"http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st7/account\",

\"hosts\":\"http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st7/hosts\",

\"instances\":\"http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st7/instances\",

\"scheduledUpgrades\":\"http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st7/scheduledupgrades\",

\"secrets\":\"http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st7/secrets\",

\"services\":\"http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st7/services\",

\"volumeTemplates\":\"http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st7/volumetemplates\",

\"volumes\":\"http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st7/volumes\",

\"composeConfig\":\"http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st7/composeconfig\"
    },
    \"actions\":{

\"deactivateservices\":\"http://192.168.99.103:8080/v2-beta/projects/1

```

```

a7\stacks\1st7\?action=deactivateservices",

"activateservices":"http:\\\\192.168.99.103:8080\v2-beta\projects\1a7\stacks\1st7\?action=activateservices",

"upgrade":"http:\\\\192.168.99.103:8080\v2-beta\projects\1a7\stacks\1st7\?action=upgrade",

"addoutputs":"http:\\\\192.168.99.103:8080\v2-beta\projects\1a7\stacks\1st7\?action=addoutputs",

"update":"http:\\\\192.168.99.103:8080\v2-beta\projects\1a7\stacks\1st7\?action=update",

"exportconfig":"http:\\\\192.168.99.103:8080\v2-beta\projects\1a7\stacks\1st7\?action=exportconfig",

"remove":"http:\\\\192.168.99.103:8080\v2-beta\projects\1a7\stacks\1st7\?action=remove"
    },
    "baseType":"stack",
    "name":"network-services",
    "state":"active",
    "accountId":"1a7",
    "answers":null,
    "binding":null,
    "created":"2017-08-27T13:15:48Z",
    "createdTS":1503839748000,
    "description":null,
    "dockerCompose":"version: '2'\n
services:\n
network-manager:\n    image: rancher/network-manager:v0.7.7\n
privileged: true\n    network_mode: host\n    pid: host\n    command:
plugin-manager --metadata-url http:\\\\169.254.169.250\\2016-07-29\n
environment:\n    DOCKER_BRIDGE: docker0\n    METADATA_IP:
169.254.169.250\n    volumes:\n    -
/var/run/docker.sock:/var/run/docker.sock\n    -
/var/lib/docker:/var/lib/docker\n    -
/var/lib/rancher/state:/var/lib/rancher/state\n    -
/lib/modules:/lib/modules:ro\n    - /run:/run\n    -
/var/run:/var/run:ro\n    - rancher-cni-driver:/etc/cni\n    -
rancher-cni-driver:/opt/cni\n    labels:\n
io.rancher.scheduler.global: 'true'\n    logging:\n    driver:
json-file\n    options:\n    max-size: 25m\n    max-file:
'2'\n    metadata:\n    cap_add:\n    - NET_ADMIN\n    image:
rancher/metadata:v0.9.3\n    network_mode: bridge\n    command:
start.sh rancher-metadata -subscribe\n    labels:\n
io.rancher.sidekicks: dns\n    io.rancher.container.create_agent:
'true'\n    io.rancher.scheduler.global: 'true'\n
io.rancher.container.agent_service.metadata: 'true'\n    logging:\n
driver: json-file\n    options:\n    max-size: 25m\n
max-file: '2'\n    sysctls:\n    net.ipv4.conf.all.send_redirects:
'0'\n    net.ipv4.conf.default.send_redirects: '0'\n
net.ipv4.conf.eth0.send_redirects: '0'\n    dns:\n    image:

```

```

rancher\/\dns:v0.15.1\
network_mode: container:metadata\
command:
rancher-dns --listen 169.254.169.250:53 --metadata-server=localhost
--answers=\/etc\/rancher-dns\/answers.json --recursor-timeout
${DNS_RECURSER_TIMEOUT} --ttl ${TTL}\
labels:\
io.rancher.scheduler.global: 'true'\
logging:\
driver:
json-file\
options:\
max-size: 25m\
max-file:
'2'\
,
    "environment":null,

"externalId":"catalog:\/\/library:infra*network-services:23",
    "group":null,
    "healthState":"healthy",
    "kind":"stack",
    "outputs":null,
    "previousEnvironment":null,
    "previousExternalId":null,
    "rancherCompose":".catalog:\
name: Network Services\
version: v0.2.5\
minimum_rancher_version: v1.6.6-rc1\
questions:\
- variable: DNS_RECURSER_TIMEOUT\
label: Timeout
for Rancher DNS Recursor\
description: Specify timeout in seconds
for DNS Recursor.\
required: true\
default: 2\
type:
int\
- variable: TTL\
label: TTL for service discovery
answers\
description: How long answers for *.rancher.internal
responses are valid\
required: true\
default: 1\
type:
int\
",
    "removed":null,
    "serviceIds":[
        "ls10",
        "ls11"
    ],
    "startOnCreate":null,
    "system":true,
    "templates":null,
    "transitioning":"no",
    "transitioningMessage":null,
    "transitioningProgress":null,
    "uuid":"aec4cb8-d65e-4ab7-9081-3a4a77ab3d46"
},
{
    "id":"1st8",
    "type":"stack",
    "links":{

"self":"http:\/\/192.168.99.103:8080\/v2-beta\/projects\/1a7\/stacks\/1st8",

"account":"http:\/\/192.168.99.103:8080\/v2-beta\/projects\/1a7\/stacks\/1st8\/account",

"hosts":"http:\/\/192.168.99.103:8080\/v2-beta\/projects\/1a7\/stacks\/1st8\/hosts",

"instances":"http:\/\/192.168.99.103:8080\/v2-beta\/projects\/1a7\/stack

```

```

s\1st8\instances",

"scheduledUpgrades": "http:\\\\192.168.99.103:8080\v2-beta\projects\1a7\stacks\1st8\scheduledupgrades",

"secrets": "http:\\\\192.168.99.103:8080\v2-beta\projects\1a7\stacks\1st8\secrets",

"services": "http:\\\\192.168.99.103:8080\v2-beta\projects\1a7\stacks\1st8\services",

"volumeTemplates": "http:\\\\192.168.99.103:8080\v2-beta\projects\1a7\stacks\1st8\volumetemplates",

"volumes": "http:\\\\192.168.99.103:8080\v2-beta\projects\1a7\stacks\1st8\volumes",

"composeConfig": "http:\\\\192.168.99.103:8080\v2-beta\projects\1a7\stacks\1st8\composeconfig"
    },
    "actions": {

"deactivateservices": "http:\\\\192.168.99.103:8080\v2-beta\projects\1a7\stacks\1st8\?action=deactivateservices",

"activateservices": "http:\\\\192.168.99.103:8080\v2-beta\projects\1a7\stacks\1st8\?action=activateservices",

"upgrade": "http:\\\\192.168.99.103:8080\v2-beta\projects\1a7\stacks\1st8\?action=upgrade",

"addoutputs": "http:\\\\192.168.99.103:8080\v2-beta\projects\1a7\stacks\1st8\?action=addoutputs",

"update": "http:\\\\192.168.99.103:8080\v2-beta\projects\1a7\stacks\1st8\?action=update",

"exportconfig": "http:\\\\192.168.99.103:8080\v2-beta\projects\1a7\stacks\1st8\?action=exportconfig",

"remove": "http:\\\\192.168.99.103:8080\v2-beta\projects\1a7\stacks\1st8\?action=remove"
    },
    "baseType": "stack",
    "name": "scheduler",
    "state": "active",
    "accountId": "1a7",
    "answers": null,
    "binding": null,
    "created": "2017-08-27T13:15:48Z",
    "createdTS": 1503839748000,
    "description": null,
    "dockerCompose": "version: '2'\n\nservices:\n  scheduler:\n

```

```

image: rancher\scheduler:v0.8.2\n    command: scheduler
--metadata-address 169.254.169.250\n    environment:\n
RANCHER_DEBUG: '${RANCHER_DEBUG}'\n    labels:\n
io.rancher.container.create_agent: \"true\"\n
io.rancher.container.agent_service.scheduling: \"true\"\n    logging:\n
driver: json-file\n    options:\n    max-size: 25m\n
max-file: '2'\n",
    "environment":null,
    "externalId":"catalog://\n/library:infra*scheduler:9",
    "group":null,
    "healthState":"healthy",
    "kind":"stack",
    "outputs":null,
    "previousEnvironment":null,
    "previousExternalId":null,
    "rancherCompose": ".catalog:\n    name: Scheduler\n
description: A resource based scheduler plugin for Rancher.\n
version: v0.6.3\n    minimum_rancher_version: v1.6.1-rc1\n
questions:\n    - variable: \"RANCHER_DEBUG\"\n    label:
\"Enable Debug Logs\"\n    description: \"This will enable very
verbose debug logs.\"\n    type: \"boolean\"\n    default:
\"false\"\n    required: true\n\nscheduler:\n    health_check:\n
request_line: GET \n/healthcheck HTTP\n/1.0\n    port: 80\n
interval: 2000\n    initializing_timeout: 10000\n
reinitializing_timeout: 10000\n    response_timeout: 2000\n
unhealthy_threshold: 3\n    healthy_threshold: 2\n",
    "removed":null,
    "serviceIds":[
        "1s8"
    ],
    "startOnCreate":null,
    "system":true,
    "templates":null,
    "transitioning":"no",
    "transitioningMessage":null,
    "transitioningProgress":null,
    "uuid":"66ff376d-3af8-4067-b103-0e8c02d4f558"
},
{
    "id":"1st9",
    "type":"stack",
    "links":{

"self":"http://\n/192.168.99.103:8080\n/v2-beta\n/projects\n/1a7\n/stacks\n/1st9",

"account":"http://\n/192.168.99.103:8080\n/v2-beta\n/projects\n/1a7\n/stacks\n/1st9\n/account",

"hosts":"http://\n/192.168.99.103:8080\n/v2-beta\n/projects\n/1a7\n/stacks\n/1st9\n/hosts",

"instances":"http://\n/192.168.99.103:8080\n/v2-beta\n/projects\n/1a7\n/stack

```

```

s\1st9\instances",

"scheduledUpgrades": "http:\\\\192.168.99.103:8080\v2-beta\projects\1a7\stacks\1st9\scheduledupgrades",

"secrets": "http:\\\\192.168.99.103:8080\v2-beta\projects\1a7\stacks\1st9\secrets",

"services": "http:\\\\192.168.99.103:8080\v2-beta\projects\1a7\stacks\1st9\services",

"volumeTemplates": "http:\\\\192.168.99.103:8080\v2-beta\projects\1a7\stacks\1st9\volumetemplates",

"volumes": "http:\\\\192.168.99.103:8080\v2-beta\projects\1a7\stacks\1st9\volumes",

"composeConfig": "http:\\\\192.168.99.103:8080\v2-beta\projects\1a7\stacks\1st9\composeconfig"
    },
    "actions": {

"deactivateservices": "http:\\\\192.168.99.103:8080\v2-beta\projects\1a7\stacks\1st9\?action=deactivateservices",

"activateservices": "http:\\\\192.168.99.103:8080\v2-beta\projects\1a7\stacks\1st9\?action=activateservices",

"upgrade": "http:\\\\192.168.99.103:8080\v2-beta\projects\1a7\stacks\1st9\?action=upgrade",

"addoutputs": "http:\\\\192.168.99.103:8080\v2-beta\projects\1a7\stacks\1st9\?action=addoutputs",

"update": "http:\\\\192.168.99.103:8080\v2-beta\projects\1a7\stacks\1st9\?action=update",

"exportconfig": "http:\\\\192.168.99.103:8080\v2-beta\projects\1a7\stacks\1st9\?action=exportconfig",

"remove": "http:\\\\192.168.99.103:8080\v2-beta\projects\1a7\stacks\1st9\?action=remove"
    },
    "baseType": "stack",
    "name": "dokuwiki1",
    "state": "active",
    "accountId": "1a7",
    "answers": null,
    "binding": null,
    "created": "2017-08-27T13:23:13Z",
    "createdTS": 1503840193000,
    "description": "DokuWiki Instance #1",
    "dockerCompose": "dokuwiki-server:\n  ports:\n    -

```



```

${http_port}:80/tcp\n labels:\n io.rancher.sidekicks:
dokuwiki-data\n hostname: ${dokuwiki_hostname}\n image:
ununseptium\/dokuwiki-docker\n volumes_from:\n - dokuwiki-data\n
\ndokuwiki-data:\n labels:\n io.rancher.container.start_once:
'true'\n entrypoint:\n - \/bin\/true\n hostname: dokuwikidata\n
image: ununseptium\/dokuwiki-docker\n volumes:\n -
\/var\/www\/html\/data\n - \/var\/www\/html\/lib\/plugins\n",
  "environment":{
    "dokuwiki_hostname":"dwl.example.com",
    "http_port":"8081"
  },
  "externalId":"catalog:\/\/community:dokuwiki:0",
  "group":null,
  "healthState":"healthy",
  "kind":"stack",
  "outputs":null,
  "previousEnvironment":null,
  "previousExternalId":null,
  "rancherCompose": ".catalog:\n name: Dokuwiki\n version:
2016-06-26a\n description: |\n DokuWiki is a simple to use and
highly versatile Open Source wiki software that doesn't require a
database.\n minimum_rancher_version: v0.56.0\n maintainer: \"Tim
Kopplow <tim@arctium.io>\"\n uuid: dokuwiki-0\n questions:\n -
variable: \"dokuwiki_hostname\"\n description: \"Dokuwiki
hostname\"\n label: \"Hostname:\"\n required: true\n
default: \"dw.example.com\"\n type: \"string\"\n - variable:
\"http_port\"\n description: \"HTTP port to expose on host. Will be
used to bind TCP\"\n label: \"HTTP port:\"\n required: true\n
default: 80\n type: \"string\"\n\ndokuwiki-server:\n scale: 1\n
retain_ip: true\n health_check:\n port: 80\n interval: 30000\n
unhealthy_threshold: 3\n strategy: recreate\n response_timeout:
2000\n healthy_threshold: 2\n\ndokuwiki-data:\n scale: 1\n
retain_ip: true\n",
    "removed":null,
    "serviceIds":[
      "1s13"
    ],
    "startOnCreate":true,
    "system":false,
    "templates":null,
    "transitioning":"no",
    "transitioningMessage":null,
    "transitioningProgress":null,
    "uuid":"c9d13752-9c66-418e-9560-28325b34c754"
  },
  {
    "id":"1st10",
    "type":"stack",
    "links":{

"self":"http:\/\/192.168.99.103:8080\/v2-beta\/projects\/1a7\/stacks\/1s
t10",

```

```
"account": "http:\\\\192.168.99.103:8080\\v2-beta\\projects\\1a7\\stacks\\1st10\\account",

"hosts": "http:\\\\192.168.99.103:8080\\v2-beta\\projects\\1a7\\stacks\\1st10\\hosts",

"instances": "http:\\\\192.168.99.103:8080\\v2-beta\\projects\\1a7\\stacks\\1st10\\instances",

"scheduledUpgrades": "http:\\\\192.168.99.103:8080\\v2-beta\\projects\\1a7\\stacks\\1st10\\scheduledupgrades",

"secrets": "http:\\\\192.168.99.103:8080\\v2-beta\\projects\\1a7\\stacks\\1st10\\secrets",

"services": "http:\\\\192.168.99.103:8080\\v2-beta\\projects\\1a7\\stacks\\1st10\\services",

"volumeTemplates": "http:\\\\192.168.99.103:8080\\v2-beta\\projects\\1a7\\stacks\\1st10\\volumetemplates",

"volumes": "http:\\\\192.168.99.103:8080\\v2-beta\\projects\\1a7\\stacks\\1st10\\volumes",

"composeConfig": "http:\\\\192.168.99.103:8080\\v2-beta\\projects\\1a7\\stacks\\1st10\\composeconfig"
    },
    "actions": {

"deactivateservices": "http:\\\\192.168.99.103:8080\\v2-beta\\projects\\1a7\\stacks\\1st10\\?action=deactivateservices",

"activateservices": "http:\\\\192.168.99.103:8080\\v2-beta\\projects\\1a7\\stacks\\1st10\\?action=activateservices",

"upgrade": "http:\\\\192.168.99.103:8080\\v2-beta\\projects\\1a7\\stacks\\1st10\\?action=upgrade",

"addoutputs": "http:\\\\192.168.99.103:8080\\v2-beta\\projects\\1a7\\stacks\\1st10\\?action=addoutputs",

"update": "http:\\\\192.168.99.103:8080\\v2-beta\\projects\\1a7\\stacks\\1st10\\?action=update",

"exportconfig": "http:\\\\192.168.99.103:8080\\v2-beta\\projects\\1a7\\stacks\\1st10\\?action=exportconfig",

"remove": "http:\\\\192.168.99.103:8080\\v2-beta\\projects\\1a7\\stacks\\1st10\\?action=remove"
    },
    "baseType": "stack",
    "name": "dokuwiki2",
    "state": "active",
```

```

        "accountId": "1a7",
        "answers": null,
        "binding": null,
        "created": "2017-08-27T13:23:33Z",
        "createdTS": 1503840213000,
        "description": "DokuWiki Instance #2",
        "dockerCompose": "dokuwiki-server:\n  ports:\n    -
    ${http_port}:80/tcp\n  labels:\n    io.rancher.sidekicks:
    dokuwiki-data\n  hostname: ${dokuwiki_hostname}\n  image:
    ununseptium/dokuwiki-docker\n  volumes_from:\n    - dokuwiki-data\n
    \ndokuwiki-data:\n  labels:\n    io.rancher.container.start_once:
    'true'\n  entrypoint:\n    - /bin/true\n  hostname: dokuwikidata\n
    image: ununseptium/dokuwiki-docker\n  volumes:\n    -
    /var/www/html/data\n    - /var/www/html/lib/plugins\n",
        "environment": {
            "dokuwiki_hostname": "dw2.example.com",
            "http_port": "8082"
        },
        "externalId": "catalog://community:dokuwiki:0",
        "group": null,
        "healthState": "healthy",
        "kind": "stack",
        "outputs": null,
        "previousEnvironment": null,
        "previousExternalId": null,
        "rancherCompose": ".catalog:\n  name: Dokuwiki\n  version:
    2016-06-26a\n  description: |\n    DokuWiki is a simple to use and
    highly versatile Open Source wiki software that doesn't require a
    database.\n  minimum_rancher_version: v0.56.0\n  maintainer: \"Tim
    Kopplow <tim@arctium.io>\"\n  uuid: dokuwiki-0\n  questions:\n    -
    variable: \"dokuwiki_hostname\"\n    description: \"Dokuwiki
    hostname\"\n    label: \"Hostname\"\n    required: true\n
    default: \"dw.example.com\"\n    type: \"string\"\n    - variable:
    \"http_port\"\n    description: \"HTTP port to expose on host. Will be
    used to bind TCP\"\n    label: \"HTTP port\"\n    required: true\n
    default: 80\n    type: \"string\"\n  \ndokuwiki-server:\n  scale: 1\n
    retain_ip: true\n  health_check:\n    port: 80\n    interval: 30000\n
    unhealthy_threshold: 3\n    strategy: recreate\n    response_timeout:
    2000\n  healthy_threshold: 2\n  \ndokuwiki-data:\n  scale: 1\n
    retain_ip: true\n",
        "removed": null,
        "serviceIds": [
            "1s14"
        ],
        "startOnCreate": true,
        "system": false,
        "templates": null,
        "transitioning": "no",
        "transitioningMessage": null,
        "transitioningProgress": null,
        "uuid": "15b7b650-c2f2-4157-bb4a-d0f53db72891"
    },
    {

```

```
        "id": "1st11",
        "type": "stack",
        "links": {

"self": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st11",

"account": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st11/account",

"hosts": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st11/hosts",

"instances": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st11/instances",

"scheduledUpgrades": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st11/scheduledupgrades",

"secrets": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st11/secrets",

"services": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st11/services",

"volumeTemplates": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st11/volumetemplates",

"volumes": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st11/volumes",

"composeConfig": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st11/composeconfig"
        },
        "actions": {

"deactivateservices": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st11/?action=deactivateservices",

"activateservices": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st11/?action=activateservices",

"upgrade": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st11/?action=upgrade",

"addoutputs": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st11/?action=addoutputs",

"update": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st11/?action=update",

"exportconfig": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st11/?action=exportconfig",
```

```

"remove": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st11/?action=remove"
},
"baseType": "stack",
"name": "weavescope",
"state": "active",
"accountId": "1a7",
"answers": null,
"binding": null,
"created": "2017-08-27T13:23:41Z",
"createdTS": 1503840221000,
"description": null,
"dockerCompose": "weavescope-probe:\n image:
weaveworks/scope:1.1.0\n privileged: true\n net: host\n pid: host\n
labels:\n   io.rancher.scheduler.global: true\n
io.rancher.container.dns: true\n links:\n - weavescope-app\n
volumes:\n - \"/var/run/docker.sock:/var/run/docker.sock"\n
tty: true\n command:\n - "--probe.docker"\n - "true"\n -
"--no-app"\n - "weavescope-app"\nweavescope-app:\n image:
weaveworks/scope:1.1.0\n ports:\n - "4040:4040"\n command:\n -
"--no-probe"\n",
"environment": {

},
"externalId": "catalog://community:weavescope:0",
"group": null,
"healthState": "healthy",
"kind": "stack",
"outputs": null,
"previousEnvironment": null,
"previousExternalId": null,
"rancherCompose": ".catalog:\n name: weavescope\n version:
1.1.0\n description: \"Monitoring, visualisation and management for
Docker\"\n",
"removed": null,
"serviceIds": [
  "1s15",
  "1s16"
],
"startOnCreate": true,
"system": false,
"templates": null,
"transitioning": "no",
"transitioningMessage": null,
"transitioningProgress": null,
"uuid": "f93baeab-385c-47b9-8eeb-add16e948964"
},
{
  "id": "1st23",
  "type": "stack",
  "links": {

```

```
"self": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st23",

"account": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st23/account",

"hosts": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st23/hosts",

"instances": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st23/instances",

"scheduledUpgrades": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st23/scheduledupgrades",

"secrets": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st23/secrets",

"services": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st23/services",

"volumeTemplates": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st23/volumetemplates",

"volumes": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st23/volumes",

"composeConfig": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st23/composeconfig"
    },
    "actions": {

"deactivateservices": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st23/?action=deactivateservices",

"activateservices": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st23/?action=activateservices",

"upgrade": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st23/?action=upgrade",

"addoutputs": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st23/?action=addoutputs",

"update": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st23/?action=update",

"exportconfig": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st23/?action=exportconfig",

"remove": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st23/?action=remove"
    },
}
```

```

        "baseType": "stack",
        "name": "my-custom-stack",
        "state": "active",
        "accountId": "1a7",
        "answers": null,
        "binding": null,
        "created": "2017-08-27T19:28:39Z",
        "createdTS": 1503862119000,
        "description": null,
        "dockerCompose": null,
        "environment": null,
        "externalId": null,
        "group": null,
        "healthState": "healthy",
        "kind": "stack",
        "outputs": null,
        "previousEnvironment": null,
        "previousExternalId": null,
        "rancherCompose": null,
        "removed": null,
        "serviceIds": [
            "1s26"
        ],
        "startOnCreate": null,
        "system": false,
        "templates": null,
        "transitioning": "no",
        "transitioningMessage": null,
        "transitioningProgress": null,
        "uuid": "6f86f5f2-8e34-4b87-b371-1fb6bba0ed99"
    },
    "sortLinks": {

```

```

"accountId": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks?sort=accountId",

```

```

"created": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks?sort=created",

```

```

"description": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks?sort=description",

```

```

"externalId": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks?sort=externalId",

```

```

"group": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks?sort=group",

```

```

"healthState": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks?sort=healthState",

```

```

"id": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks?sort="

```

```
id",

"kind": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks?sort=kind",

"name": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks?sort=name",

"removeTime": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks?sort=removeTime",

"removed": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks?sort=removed",

"state": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks?sort=state",

"system": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks?sort=system",

"uuid": "http://192.168.99.103:8080/v2-beta/projects/1a7/stacks?sort=uuid"
  },
  "pagination": {
    "first": null,
    "previous": null,
    "next": null,
    "limit": 100,
    "total": null,
    "partial": false
  },
  "sort": null,
  "filters": {
    "accountId": null,
    "created": null,
    "description": null,
    "externalId": null,
    "group": null,
    "healthState": null,
    "id": null,
    "kind": null,
    "name": null,
    "removeTime": null,
    "removed": null,
    "state": null,
    "system": null,
    "uuid": null
  },
  "createDefaults": {
```



```
}  
}
```

Fetch Stack

v2-beta v1

▼ [Click here to expand...](#)

```
curl -u 74FF92AEC4691F7D8AF4:Kyw1kfNSJZXrSk8nK1f93P8bhxNMaPDEGEpsRPcT  
http://192.168.99.103:8080/v2-beta/projects/1a7/stacks/1st23
```

```
{  
  "id": "1st23",  
  "type": "stack",  
  "links": {  
    "self": ".../v2-beta/projects/1a7/stacks/1st23",  
    "account": ".../v2-beta/projects/1a7/stacks/1st23/account",  
    "hosts": ".../v2-beta/projects/1a7/stacks/1st23/hosts",  
    "instances": ".../v2-beta/projects/1a7/stacks/1st23/instances",  
  
    "scheduledUpgrades": ".../v2-beta/projects/1a7/stacks/1st23/scheduledupgrades",  
    "secrets": ".../v2-beta/projects/1a7/stacks/1st23/secrets",  
    "services": ".../v2-beta/projects/1a7/stacks/1st23/services",  
  
    "volumeTemplates": ".../v2-beta/projects/1a7/stacks/1st23/volumetemplates",  
    "volumes": ".../v2-beta/projects/1a7/stacks/1st23/volumes",  
  
    "composeConfig": ".../v2-beta/projects/1a7/stacks/1st23/composeconfig"  
  },  
  "actions": {  
  
    "deactivateservices": ".../v2-beta/projects/1a7/stacks/1st23/?action=deactivateservices",  
  
    "activateservices": ".../v2-beta/projects/1a7/stacks/1st23/?action=activate  
services",  
    "upgrade": ".../v2-beta/projects/1a7/stacks/1st23/?action=upgrade",  
  
    "addoutputs": ".../v2-beta/projects/1a7/stacks/1st23/?action=addoutputs",  
    "update": ".../v2-beta/projects/1a7/stacks/1st23/?action=update",  
  
    "exportconfig": ".../v2-beta/projects/1a7/stacks/1st23/?action=exportconfig",  
    "remove": ".../v2-beta/projects/1a7/stacks/1st23/?action=remove"  
  },  
  "baseType": "stack",  
  "name": "my-custom-stack",  
}
```

```
"state": "active",
"accountId": "1a7",
"answers": null,
"binding": null,
"created": "2017-08-27T19:28:39Z",
"createdTS": 1503862119000,
"description": null,
"dockerCompose": null,
"environment": null,
"externalId": null,
"group": null,
"healthState": "healthy",
"kind": "stack",
"outputs": null,
"previousEnvironment": null,
"previousExternalId": null,
"rancherCompose": null,
"removed": null,
"serviceIds": [
    "1s26"
],
"startOnCreate": null,
"system": false,
"templates": null,
"transitioning": "no",
"transitioningMessage": null,
```

```
"transitioningProgress":null,  
"uuid":"6f86f5f2-8e34-4b87-b371-1fb6bba0ed99"  
}
```

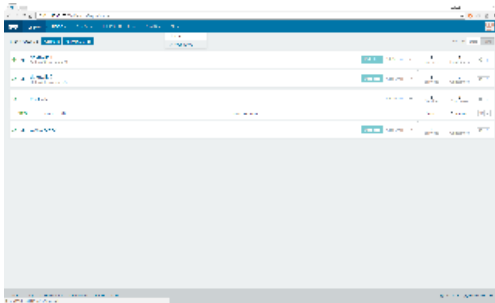
08 - Webhooks

This page details how to use webhooks in order to have a running instance of Rancher server respond to outside world events.

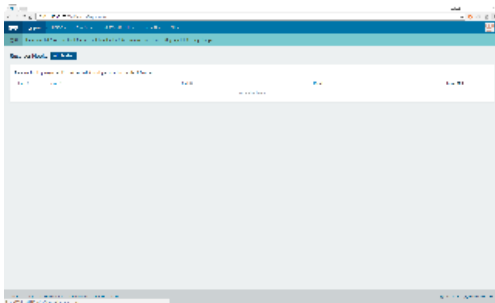
A webhook is an endpoint publicly available, so whenever Rancher should take a specific action, notify it by posting a request with no body to a specific endpoint.

More information can be found [here](#).

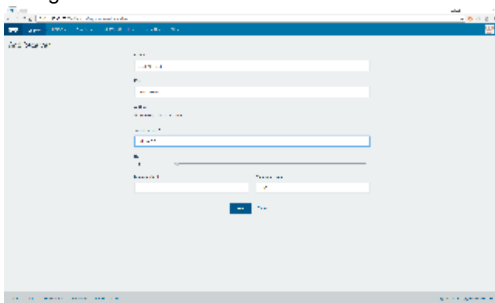
1. Create stack my-stack, as documented here: [Create and Destroy Stack & Services Using Rancher CLI](#)
2. Go to Rancher UI
3. Go to menu API Webhooks



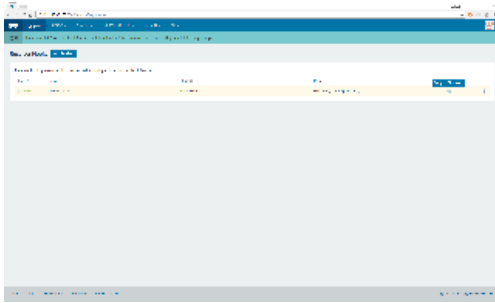
4. Click the "Add Receiver" button



5. Fill in the fields appropriately, then click the "Create" button
 - a. Name: scale-my-stack
 - b. Kind: Scale a service
 - c. Action: Scale Up
 - d. Target Service: alpine-3-5 from stack my-stack
 - e. By: 1
 - f. Minimum Scale: 1
 - g. Maximum Scale: 100



6. See the newly added webhook



7. Copy the trigger URL
8. Open a terminal and run the following command **3 times**

```
curl -X POST  
"http://192.168.99.103:8080/v1-webhooks/endpoint?key=qIgRKbKri2MEUMoK  
U3hUC5uieK9DWYqbApw9VgGT&projectId=1a7"
```

Please make sure the URL above is put inside double quotes.

9. Go to Rancher UI and you should see that my-stack contains 4 containers (the default one + **3 added** by calling the webhook)

