AUGUST 10, 2015

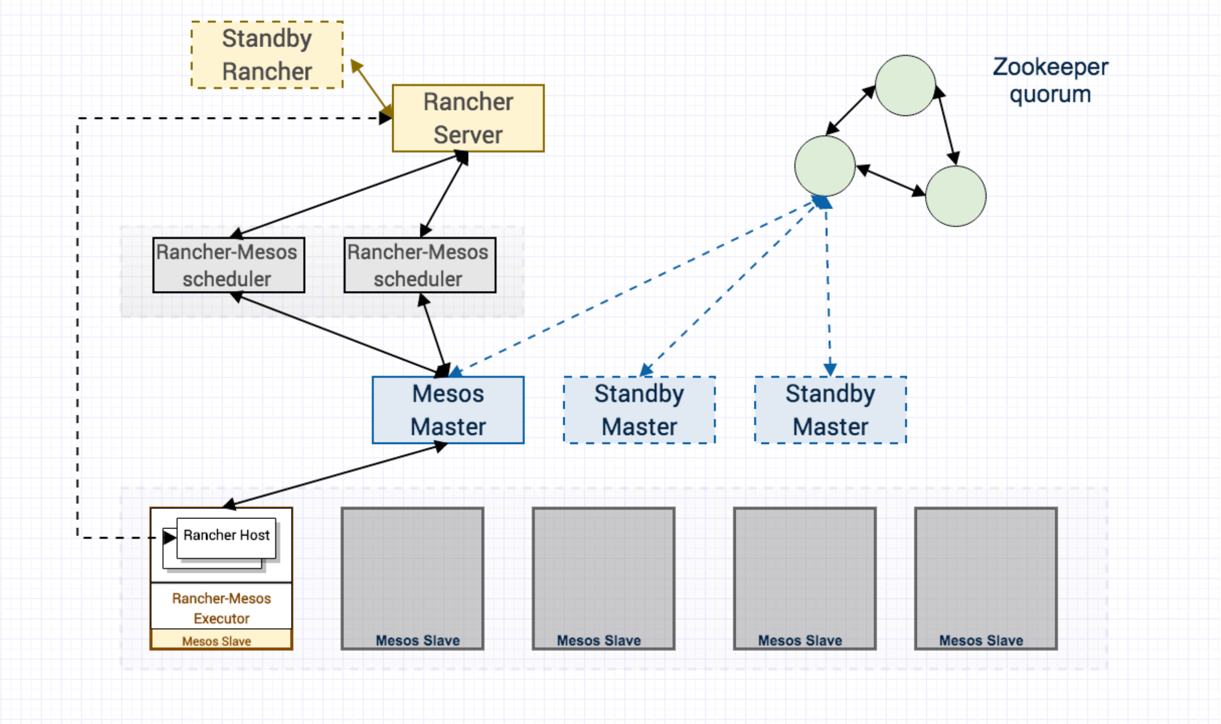
Building a Rancher Container Service on Mesos

DOCKER MESOS ORCHESTRATION

*This post is now a bit out of date. Since posting this article we've added full support for Mesos environments directly into Rancher. You can read more about it at rancher.com/mesos. * Hi, I'm Sidhartha Mani, one of the engineers here at Rancher Labs. Over the last few months I've been working with Apache Mesos, an open source resource manager and scheduler, which can be used to deploy workloads on infrastructure. Mesos is very good at understanding resources and capacity management, but for more advanced workload or container management, many users choose a framework such as Marathon. Frameworks provide capabilities like load balancing, service discovery, rolling upgrades, application composability, continuous deployment and others. A number of Rancher's community members who are also using Mesos have felt the need for a system that combines the fault tolerance, and scheduling capabilities of Mesos with the multi-tenant, private container service capabilities provided by Rancher. One community user, Marcel Neuhausler from AT&T Foundry, took the initiative to chart a broad design and envisioned a workflow for such an integration. He had a very interesting insight to combine Mesos' ability to schedule VMs with Rancher's ability to then manage those VMs and deploy containers on them. He wrote a Mesos framework for Rancher on Mesos, which proved to be a great starting point for writing the framework I'm going to talk about today. In this blog, I am going to describe the ideas and the software - Rancher Mesos framework, that resulted through the act of collaboration with Marcel. This framework can be used to setup large scale production jobs like Hadoop, Kafka, ElasticSearch etc. in docker containers, as well as any part of the DevOps pipeline, such as build, test, staging or production application environments. In the sections below, I'll discuss the architecture of the framework, and show you how to set it up on your local environment.

[](https://gist.github.com/wlan0/b5379d9d727579496c5b#the-rancher-mesos-architecture)The Rancher Mesos Architecture

Here’s a diagram explaining the Rancher Mesos Framework



As you can see, Rancher integrates with Mesos using standard Mesos interfaces - A Mesos framework. The various components are

1: Mesos Master: The Mesos Master is a cluster of machines that run the `mesos-master` process. It maintains, and monitors the Mesos slaves, and handles resource offering, task launching, task monitoring, fault tolerance and message passing etc.

2: Mesos Slave: These are the hosts on which jobs are to be scheduled. In this case, these will be used to launch VMs that register with Rancher. We use VMs instead of containers because we provision hosts using Mesos, unlike other frameworks that schedule jobs on it. These VMs can then be orchestrated using Rancher, and jobs can be scheduled on them using containers.

3: Rancher Server: This is a cluster of machines that run the `rancher/server` docker container. It maintains, and monitors the Rancherhost, while providing a multi-tenant container service to users which includes a number of container management and infrastructure features, such as private deployment environments, container networking, Docker compose support, service discovery, rolling upgrades, registry management and more.

4: Rancher Hosts: These are hosts provisioned using Mesos’ resource offers. These hosts run docker and have the `rancher/agent` container running, which is used for Rancher’s private networking, container scheduling and for various tasks involving the hosts.

5: Rancher-Mesos Scheduler (github): The scheduler is a two tiered application. It is a Rancher external event handler, as well as a Mesos scheduler. The event handler is used to listen on `create host` event from Rancher. The scheduler is used to listen for `resource offers` from Mesos. When the Rancher-Mesos scheduler receives a `create host` event, it adds that event to an event queue. Once Mesos provides a suitable slave to schedule tasks on, it de-queues events , and the Rancher Host is created on that Mesos slave, if it has sufficient capacity.

6: Rancher-Mesos Executor (github): This is the process that is invoked when an available slave is provided to Rancher for creating hosts. This process uses QEMU-KVM to create

VMs with bridge networking. Docker is installed on these VMs and then `rancher/agent` is started to make it register with Rancher Server.

7: Rancher-Mesos Framework: The Rancher-Mesos Framework is used to refer to Rancher-Mesos Scheduler and Rancher-Mesos Executor collectively.

[](https://gist.github.com/wlan0/b5379d9d727579496c5b#the-rancher-mesos-workflow)The Rancher Mesos Workflow

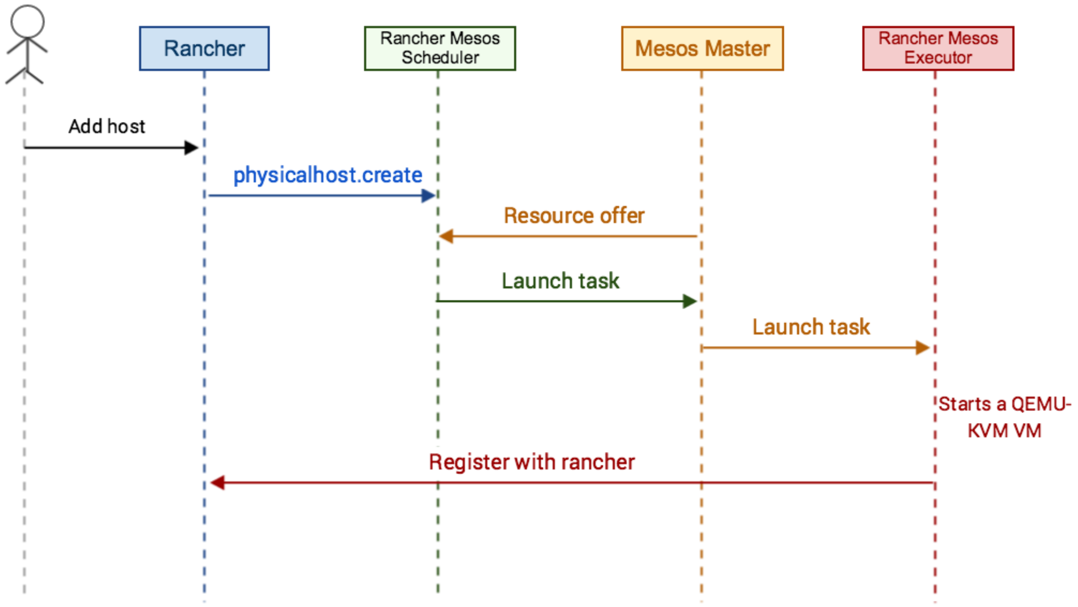
The user's point of view of working with the Rancher Mesos framework would be no different from using Rancher today.

1: The user would click on `Add Host` in the UI, which would provision a host in one of the available Mesos slaves. The slave on which it is provisioned is determined by the Mesos master.

2: The host, once provisioned will register itself with the rancher server. It will show up in the UI and the user can view stats, execute shell or start/stop containers like normal.

[](https://gist.github.com/wlan0/b5379d9d727579496c5b#looking-under-the-hood)Looking under the hood

This picture explains the sequence of events to provision a host using Rancher Mesos



1: When you click on `Add Host` in the UI, Rancher server creates a `physicalhost.create` event.

2: This event is received by all the external handlers that have subscribed to this event. In this case, the Rancher-Mesos scheduler subscribes to this event.

3: On Receiving the event, the scheduler saves the event in an event queue.

4: Then the scheduler waits for a resource offer of a free host from Mesos Master.

5: Once the scheduler receives the resource offer, it can retrieve the earliest event from the queue, and launch that task on the offered host.

6: The task starts Rancher Mesos Executor. The executor uses QEMU-KVM to start a new VM.

7: Then it install docker on the new VM.

8: The executor then instructs the new VM to registers itself as a host with rancher server.

[](https://gist.github.com/wlan0/b5379d9d727579496c5b#setting-up-and-running-rancher-mesos-framework)Setting up and running Rancher Mesos framework

In this section, I'll show you how to setup this architecture on your laptop to try it out. We'll use VMware fusion pro to virtualize the setup as it requires changing networking configuration, and its easier to work this way. Download the iso for Ubuntu Desktop 14.04.2. In VMware fusion, select `Add > Install from disk or image`. Make sure you enable nested virtualization, and have at least 2GB of Memory before booting. To enable nested virtualization

```
    Click on settings >
        Processors and Memory >
            Advanced Options >
                Enable Hypervisor Applications
```

Now boot it up.

1: The first step in setting up is network configuration. We need to setup bridge networking for `eth0`. Before continuing, ensure that `bridge-utils` is installed, using `sudo apt-get install bridge-utils`. Setup your `/etc/network/interfaces` as follows :-

```
auto lo
iface lo inet loopback


auto eth0
iface eth0 inet manual


auto br0
iface br0 inet dhcp
    bridge_ports eth0
    bridge_stp off
    bridge_fd 0
    bridge_maxwait 0
```

Then run `ifup -a`, which reads the config file and sets up the bridge interface. If you run `ifconfig` now, you'll notice there is no IP address on `eth0`, and there is a `br0` interface with a configured IP address. From here onwards, when I refer to `$IP`, it is the IP address on `br0` on this machine

2: The next step is installing the necessary packages. First, you'll need git `sudo apt-get install git` To install QEMU-KVM, use this command,

```
sudo apt-get install -y qemu-kvm libvirt-bin ubuntu-vm-builder bridge-utils
echo 'allow br0' > /etc/qemu/bridge.conf
```

Then, install the executor (You need golang, mercurial , and Godeps)

```
go get -d github.com/wlan0/rancher-mesos-executor
cd $GOPATH/src/github.com/wlan0/rancher-mesos-executor && ./scripts/build
sudo cp build/rancher-mesos-executor /bin/
```

Then replace executor with scheduler in the previous steps to install rancher-mesos-scheduler. Finally, install docker

```
wget -qO- https://get.docker.com | sh
```

3: Start rancher-server

```
sudo docker run -p 8080:8080 -d wlan0/rancher-server
```

This will start rancher-server on port 8080

4: Install Mesos master and slave

```
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv E56151BF
echo "deb http://repos.mesosphere.io/ubuntu/ trusty main" > /etc/apt/sources.list.d/mesosphere.list
sudo apt-get -y update
sudo apt-get -y install mesos
service zookeeper stop
sudo apt-get -y remove --purge zookeeper
echo manual > /etc/init/mesos-master.override
```

5: Start Mesos master and slave

```
sudo nohup mesos-master --work_dir=$(pwd) --ip=$IP &
sudo nohup mesos-slave --master=$IP:5050 --ip=$IP &
```

6: Start the rancher-mesos scheduler.

```
CATTLE_URL=http://$IP:8080/v1 CATTLE_ACCESS_KEY=service CATTLE_SECRET_KEY=servicepass MESOS_MASTER=$IP:5050 IP_CIDR=$
```

7: Now from a browser, go to `$IP:8080` to see the Rancher UI. Now, I had to do a bit of a hack to get the UI to call to mesos, as Rancher uses Docker Machine for adding resources from clouds, and I haven't had time to create a custom Mesos machine driver. So, to add a host, click on add host within any environment and select the \"Rackspace\" icon, use any dummy credentials, and hit create. You should see a host get added to the Infrastructure tab of Rancher. Wait for a few minutes for the host to connect to cattle. Once it does, you'll be able to use this host to start containers. Note: I have short circuited the authentication part in the external handler(rancher-mesos-scheduler) to ignore the cloud type and always provision Mesos hosts, so this will work from any of the different cloud drivers or from the Rancher API. In the future I'll add a proper Mesos driver for the create host function.

8: Note that everytime you provision a host, the console for the created VM will pop up on your screen. This can be disabled for production environments.

Hopefully this gives you an idea of how to deploy Rancher as a framework on Mesos. Thanks again to Marcel from AT&T Foundry for workign with our team on this, and all the other community members who have attempted or suggested this integration. With Rancher on Mesos, creating a multi-tenant private container service, on top of your Mesos cluster.

[](https://gist.github.com/wlan0/b5379d9d727579496c5b#next-steps)Next Steps

1: If you have any questions, please post them on our forums

2: If you like to reach out to me, or have any questions, email me at sid@rancher.com

Also, If you'd like help setting up your environment please join our Rancher beta program, request a demo, or register for our next online meetup.