oVirt and Docker Integration

October 2014

Federico Simoncelli
Principal Software Engineer – Red Hat

Agenda

- Deploying an Application (Old-Fashion and Docker)
- Ecosystem: Kubernetes and Project Atomic
- Current Status of Integration
  - oVirt Docker User-Interface Plugin
  - “Dockerized” oVirt Engine
  - Docker on Virtualization
- Possible Future Integration
  - Managing Containers as VMs
  - Future Multi-Purpose Data Center

Deploying an Application (Old-Fashion)

- Deploying an instance of Etherpad

```
# yum search etherpad
Warning: No matches found for: etherpad
No matches found

$ unzip etherpad-lite-1.4.1.zip
$ cd etherpad-lite-1.4.1
$ vim README.md
...
## GNU/Linux and other UNIX-like systems
You'll need gzip, git, curl, libssl develop libraries, python and gcc.
*For Debian/Ubuntu*: `apt-get install gzip git-core curl python libssl-dev pkg-
config build-essential`
*For Fedora/CentOS*: `yum install gzip git-core curl python openssl-devel && yum
groupinstall "Development Tools"`
*For FreeBSD*: `portinstall node, npm, git (optional)`

Additionally, you'll need [node.js](http://nodejs.org) installed, Ideally the
latest stable version, be careful of installing nodejs from apt.
...
```

Installing Dependencies (Old-Fashion)

- **134** new packages required

```
$ yum install gzip git-core curl python openssl-devel
Transaction Summary
================================================================================
Install  2 Packages (+14 Dependent packages)

$ yum groupinstall "Development Tools"
Transaction Summary
================================================================================
Install  7 Packages (+19 Dependent packages)

$ yum install nodejs
Transaction Summary
================================================================================
Install  1 Package (+4 Dependent packages)

$ yum install npm
Transaction Summary
================================================================================
Install  1 Package  (+86 Dependent packages)
```

Few dependencies later finally...

```
$ ./bin/run.sh
Ensure that all dependencies are up to date...  If this is the first time you
have run Etherpad please be patient.
npm WARN engine helenus@0.6.2: wanted: {"node":">=0.6.0 <0.9.0"} (current:
{"node":"v0.10.30","npm":"1.3.6"})
...
```

- Will it work for me?
- The warning is coming from a third-party library, will it really affect Etherpad?
- What was the reason to not support node > 0.9.0?
- What should I do now?

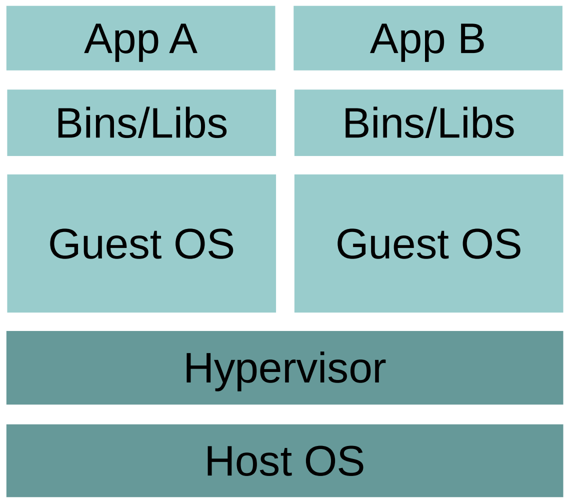
Building and Deploying Requirements

- Distributing your application should be easy (one packaging system fits all)
- Freedom for the developer to choose the platform
- Dependencies should be magically available on all platforms
- The platform of the developer should be the same used by QA and the same used in production
- Rebuilding your appliance or application should be as easy as running one single command

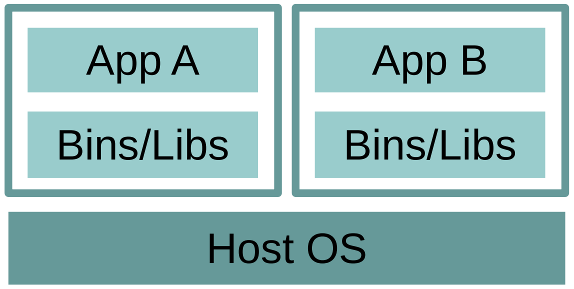
What is docker ?

- Open platform for developers and sysadmins to build, ship, and run distributed applications
- **Docker Engine** is a portable lightweight runtime and packaging tool
- **Docker Hub** is a cloud service for sharing applications and automating workflows (13,000+ applications available)
- Enables applications to be quickly assembled from components (eliminating the friction between development, QA, and production)
- The same application can run unchanged on laptops, data center VMs, and any cloud

Virtual Machine vs.



- Virtual Machine
  - Application
  - Necessary binaries and libraries
  - Entire guest operating system



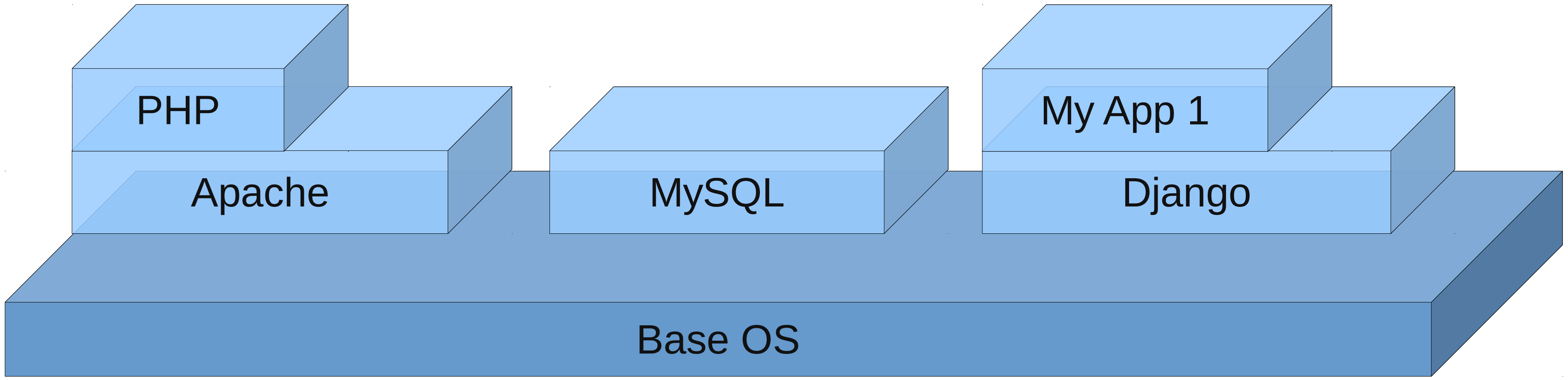
- Docker Container
  - Application
  - Necessary binaries and libraries
  - Uses the same kernel of the host

Deploying with docker

```
$ docker search etherpad
NAME                            DESCRIPTION    STARS     OFFICIAL    AUTOMATED
johbo/etherpad-lite                            1                     [OK]
mnagaku/docker-etherpad-lite                   1                     [OK]
...

$ docker run johbo/etherpad-lite
Generating settings file /data/etherpad-settings.json
start...
...

$ docker ps
CONTAINER ID         IMAGE                            COMMAND
d41cc9e20757         johbo/etherpad-lite:latest       "bin/configure_and_r
    ...CREATED              STATUS                 PORTS                   NAMES
    ...2 minutes ago        Up 2 minutes           9001/tcp                sharp_poincare

$ docker inspect d41cc9e20757
...
        "ExposedPorts": {
            "9001/tcp": {}
        },
...
```

Up and running

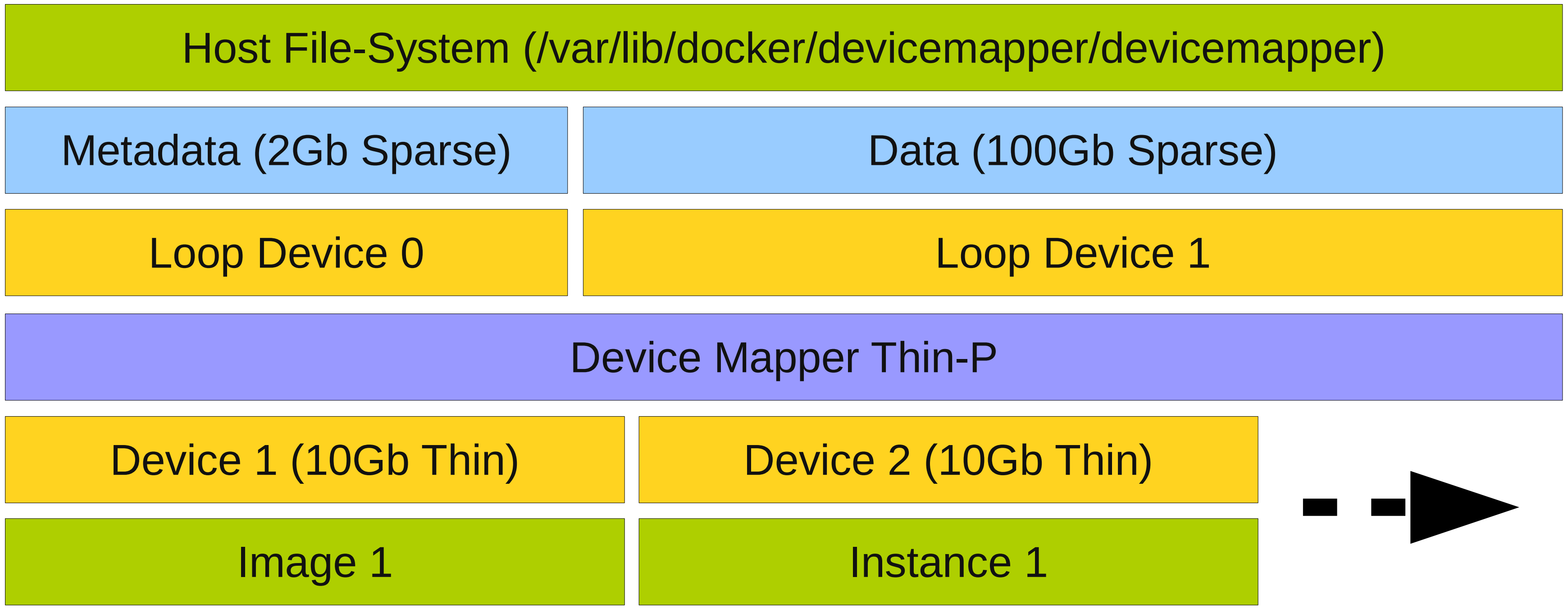
Docker Images Dependencies



- Each image may depend on another image which forms the layer beneath it
- All images are identified by a 64 hexadecimal digit string (internally a 256bit value)
- Images can be tagged

Docker Under The Hood – Images

- Graph Drivers (aufs, btrfs, devmapper, vfs)
  - Ability to quickly clone an image and apply changes
  - Default is devmapper



Docker Ecosystem Overview

- Ecosystem has an extremely fast pace
- April 2014 – Red Hat announces Project Atomic
  http://www.projectatomic.io
- June 2014 – Google announces Kubernetes
  https://github.com/GoogleCloudPlatform/kubernetes
- Hundreds of companies and projects joined the ecosystem in the last few months
  - https://github.com/google/cadvisor
  - https://github.com/zettio/weave
- oVirt contributors are actively monitoring the ecosystem and researching possible integration points

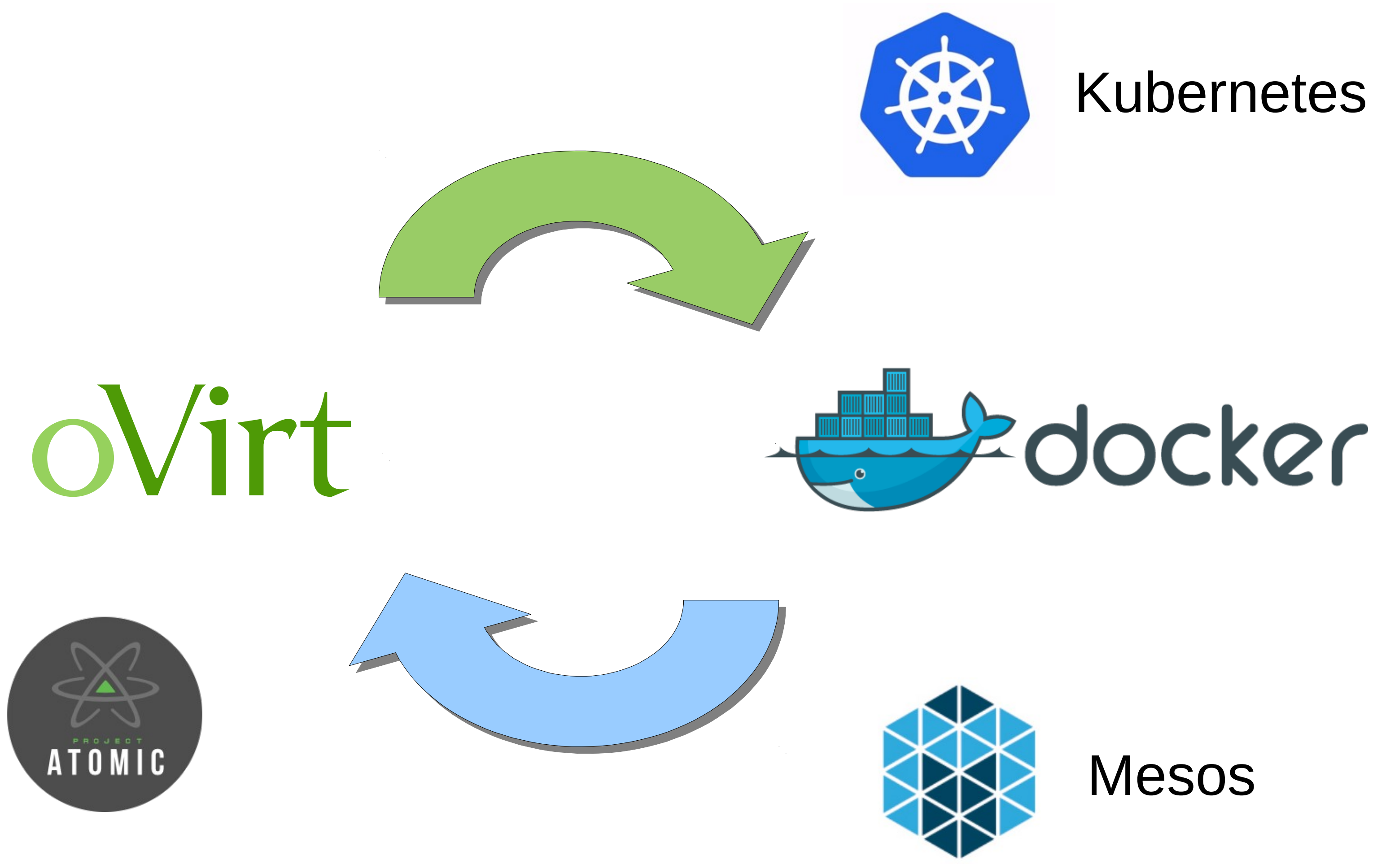
Ecosystem: Project Atomic

- Project Atomic Host: lightweight operating system that has been assembled out of upstream RPM content
- Integrates the tools and patterns of container-based application
- Providing an end-to-end solution for deploying containerized applications quickly and reliably
- Uses rpm-OSTree, an open-source tool for managing bootable, immutable, versioned filesystem trees from upstream RPM content

Ecosystem: Kubernetes

- Open source implementation of container cluster management
- Uses Docker to package, instantiate, and run containerized applications (**Pods**)
- Establishes robust declarative primitives for maintaining the desired state requested by the user
- Automatically chooses hosts (**Minions**) to run those containers on (**Scheduler**)
- Architecturally, It is built as a collection of pluggable components and layers (ability to use alternative schedulers, storage systems, and distribution mechanisms)

Co-Existing with Containers

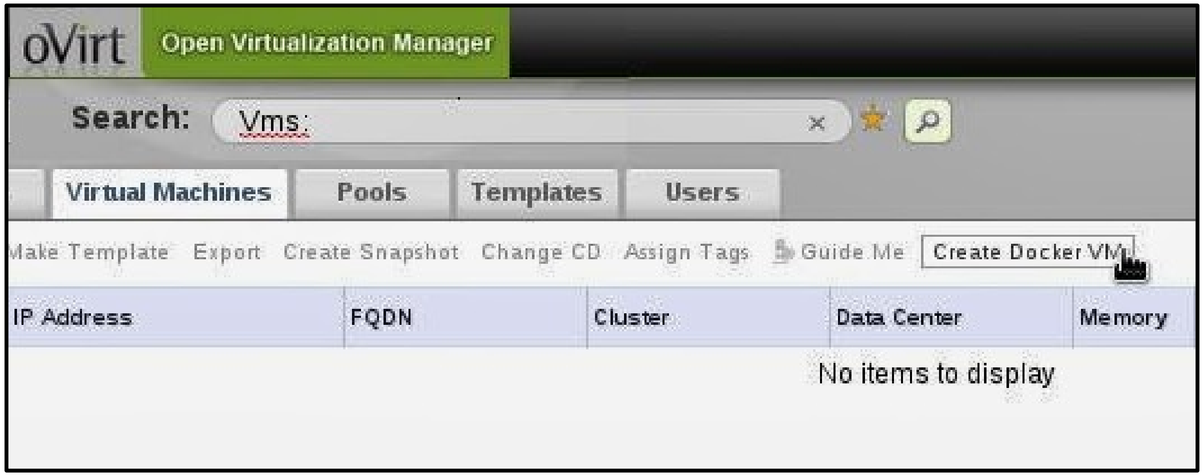


Integration with Containers

1. Utilities and tools to automate and simplify the deployment of Containers
   - UI Plugin to run Containers in VMs
   - Docker VM image available on public Glance repository
   - oVirt Engine deployment as a Container
2. Enabling Containers Managers to use oVirt as IaaS to orchestrate Containers
3. Containers on oVirt Nodes
4. Possible evolution to a Multi-Purpose Data Center (different types of workloads)

Docker on oVirt UI Plugin

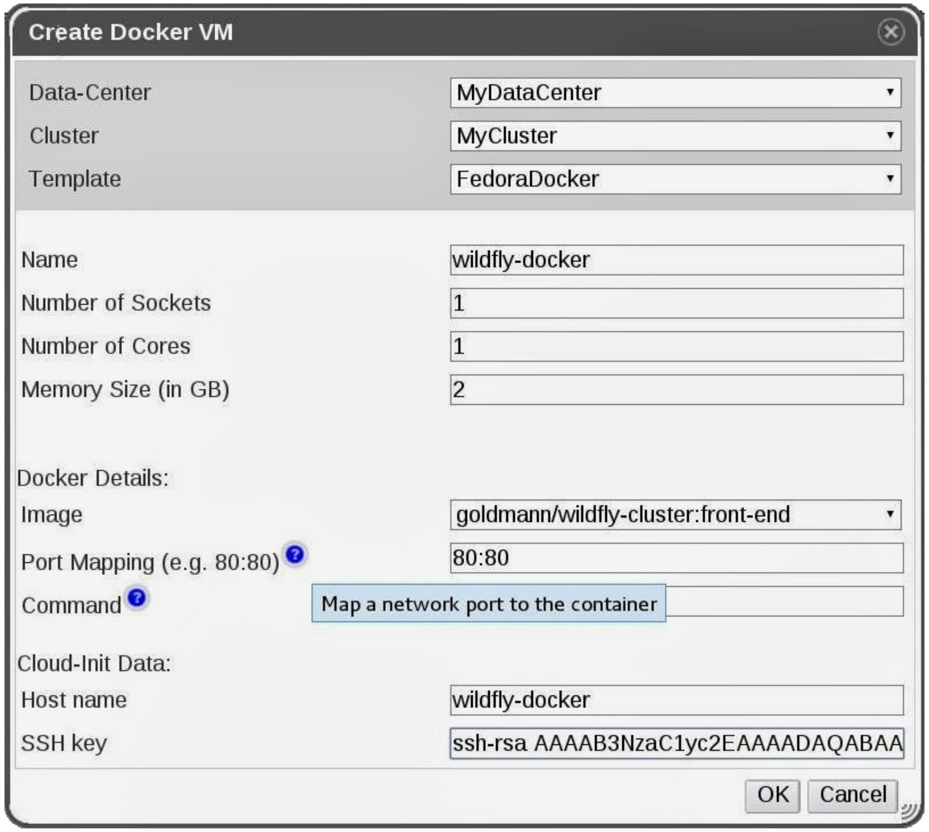
- Allows the user to create a new oVirt VM, that runs a selected Docker image running a specified command
- Uses the Cloud-Init integration in order to pass the Docker commands to the guest
- Docker image is downloaded from the public registry to the VM on first launch



http://ovedou.blogspot.co.il/2014/03/running-docker-container-in-ovirt.html

Docker on oVirt UI Plugin

- Code available in the oVirt samples-uiplugins repository
- In order to use it you need the Docker Service, Cloud-Init, and ovirt-guest-agent ("CentOS 6.5 64-Bit Docker" on Public Glance Repository)
- It works only in Cluster Level 3.4 (persisting the Cloud-Init properties)



Create Docker VM

Data-Center	MyDataCenter
Cluster	MyCluster
Template	FedoraDocker
Name	wildfly-docker
Number of Sockets	1
Number of Cores	1
Memory Size (in GB)	2

Docker Details:

Image	goldmann/wildfly-cluster:front-end
Port Mapping (e.g. 80:80)	80:80
Command	

Map a network port to the container

Cloud-Init Data:

Host name	wildfly-docker
SSH key	ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAA

http://ovedou.blogspot.co.il/2014/03/running-docker-container-in-ovirt.html

“Dockerized” oVirt Engine

- oVirt Engine instance inside a container:

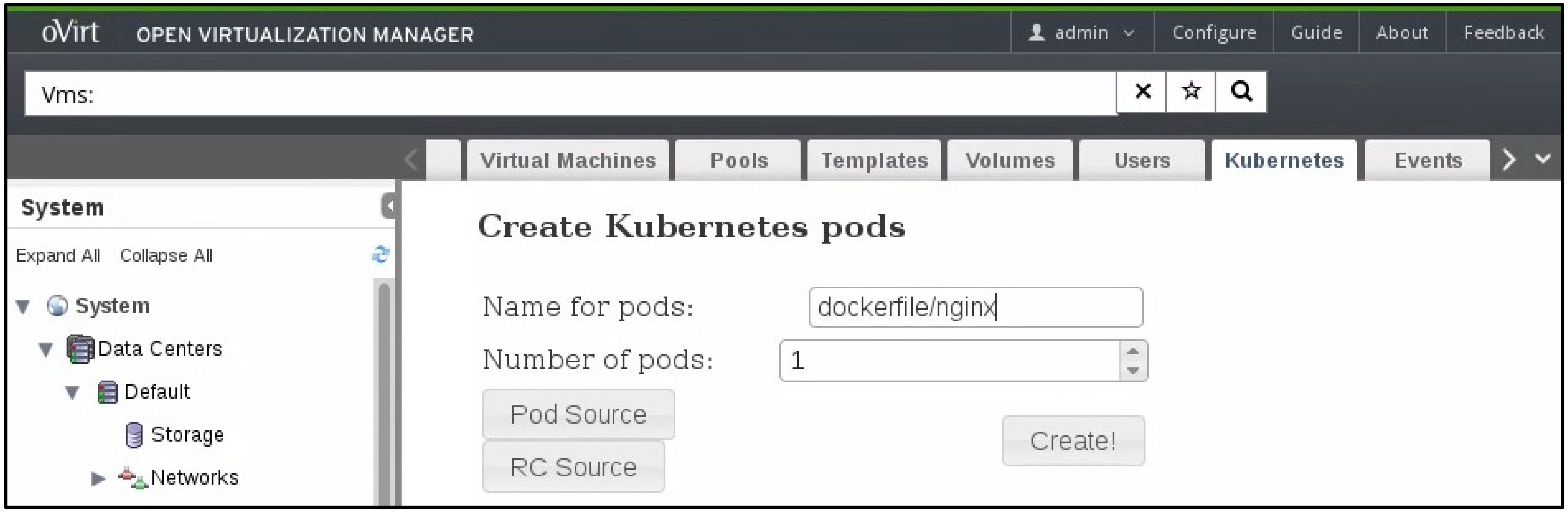
```
docker run --privileged -dt -p 80:80 -p 443:443 \
        --name ovirt mgoldboi/ovirt-sa-configured-3.5.0
```

https://github.com/mgoldboi/oVirt-Dockerized/

- Configuration layer on top of base image with oVirt packages deployed (Fedora 20)
- Options to run stand-alone or connected to an external database

Docker on Virtualization

- Running Containers inside Virtual Machines
- oVirt is not aware of Containers
- oVirt may include tools and plugins to help you visualizing containers in the Data Center



Create Kubernetes pods

Name for pods: dockerfile/nginx

Number of pods: 1

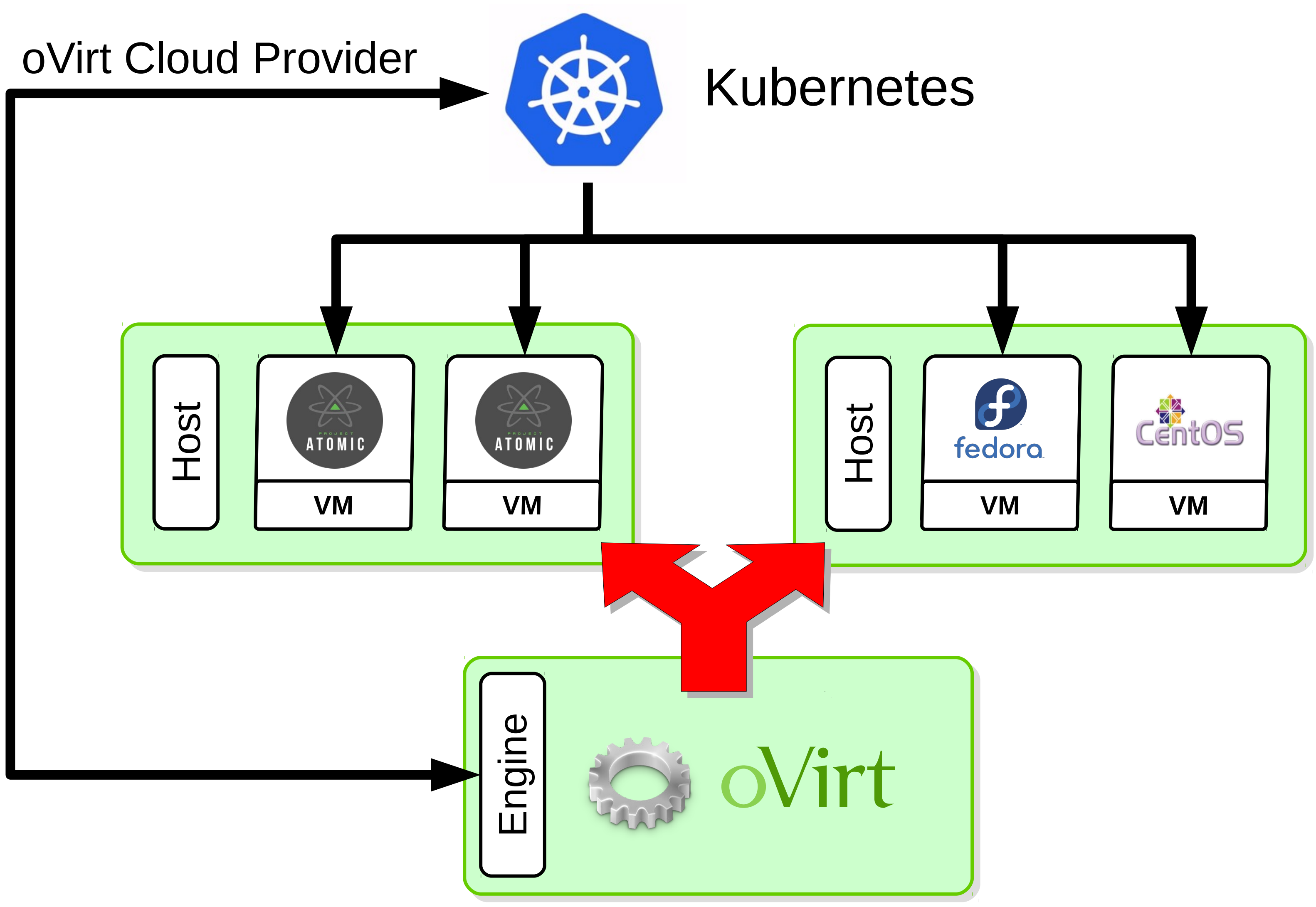
Kubernetes Cloud Provider for oVirt

- Merged in Kubernetes master the 12th of Sep 2014
  https://github.com/GoogleCloudPlatform/kubernetes/pull/1189
- Allows Kubernetes to discover Docker VMs (Minion) in oVirt
- Simple configuration:

```
[connection]
uri = https://ovirt-engine:8443/ovirt-engine/api
username = admin@internal
password = admin

[filters]
vms = tags=kubernetes
```

- May allow to discover hosts as well in the future

Docker on Virtualization



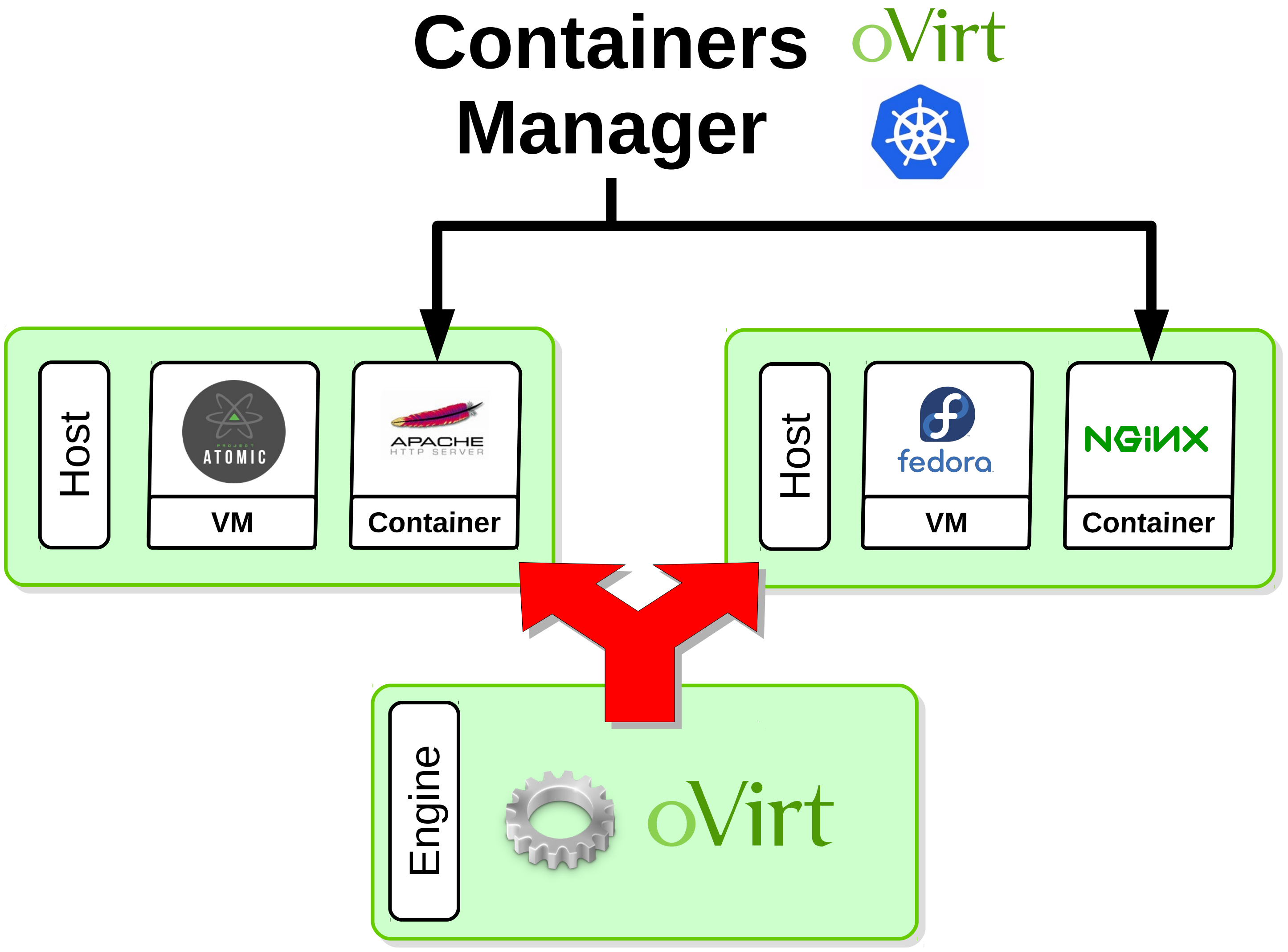
oVirt Kubernetes and Docker

Live Demo Video

Managing Containers as VMs

- Are VMs and Containers alike?
  - Do they share the same operations, can they be managed seamlessly?
  - Container Live Migration? (CRIU: checkpoint and restore functionality for Linux in userspace)
- What about Security? (Wider surface of attack, SELinux)
- Would a Monolithic Scheduler be sufficient on large scale Data Center? (vs. Two-Level / Shared-State)
- What agent should manage the Containers? (VDSM, Kubelet?)

Virtualization and Docker



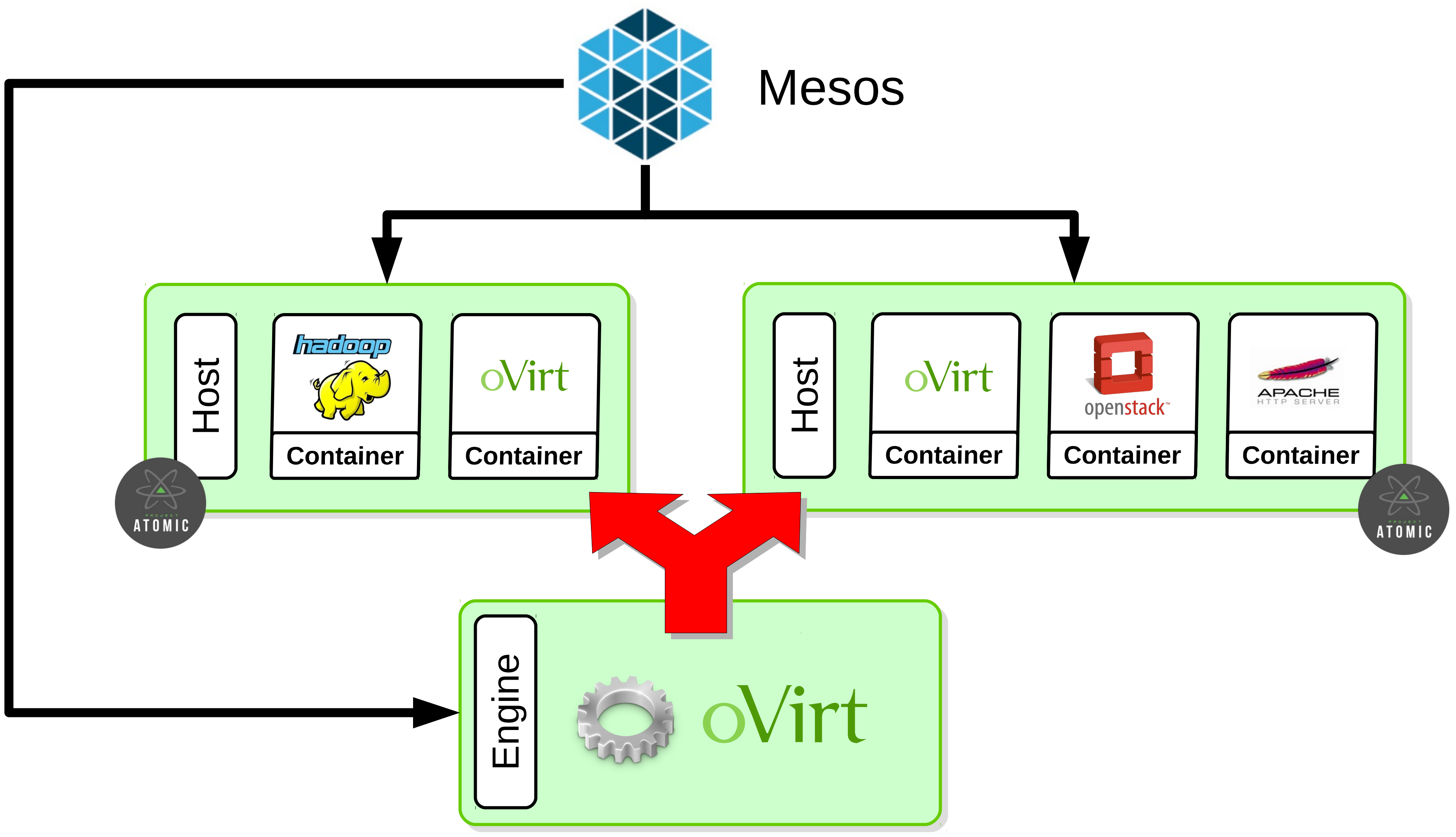
Ecosystem: Mesos

- Provides the fine-grained resource allocations for pods across nodes in a cluster
- Makes Kubernetes play nicely with other frameworks running on the same cluster resources
- Offers to the Kubernetes scheduler sets of available resources from the cluster nodes (slaves/minions)

Future Multi-Purpose Data Center

- Multiple Workloads and Managers (oVirt, OpenStack, Hadoop)
- Hosts are Multi-Purpose running Project Atomic and Containers
- Hosts are dynamically assigned to a certain type of Workload by a Scheduler (e.g. Mesos)
- oVirt required resources (Hosts to run VMs for a certain Cluster) will be assigned by Mesos

Multi-Purpose Data Center



THANK YOU!

devel@ovirt.org

#ovirt irc.oftc.net