Lightweight virtualization -- docker in practice

- Baidu BAE team -- Chen YiFei
- 2013-10.25

Outline

- 1. Lightweight virtualization
  - What is Lightweight virtualization?
  - What technology is behind Lightweight virtualization?
- 2. BAE and docker
  - What is docker?
  - Why does BAE choose docker?
  - How does BAE use docker?
- 3. Docker developments and forecast

What is Lightweight virtualization technology

- **From Linux-process perspective**
  - What are the surrounding environments and resources involved when the processes are running?
    - Linux Kernel
    - File System
    - Network System
    - PID, UID, IPC and other resources
    - memory, disk, CPU and other resources
  - Each process sees the same surroundings
  - All processes share these same resources

What is Lightweight virtualization technology

- With development in technology, new requirements are formed:
  - **Resource isolation:** different processes need their own independent surroundings
  - **Resource constrains:** some processes can only have limited resources
  - Ability to isolate **a group of processes** and set limitations to them

What is Lightweight virtualization technology

- Requirements are summarized as follows:
  - for a group of processs
  - Allocate separate operating environment
    - File System
    - Network System
    - PID, UID, UTS, mount, IPC namespace
  - Able to limit resources they can use
    - Memory
    - CPU
    - Network Traffic
    - Disk Space
    - Disk read and write frequency
  - It would ideal if interference between process groups can be eliminated

What is Lightweight virtualization technology

- Lightweight virtualization is the technology used to fulfill these requirements
- The process groups that meet the above restrictions are called "lightweight virtual machine" or Container
- Process Container concept was first introduced by Google engineers in 2006
  - http://lwn.net/Articles/199643/
  - http://lwn.net/Articles/236038/
- wikipedia definition
  - http://en.wikipedia.org/wiki/Operating_system-level_virtualization

What is Lightweight virtualization technology

- Demonstration

```
root@c5cec4b035ec:~# ps axf
  PID TTY      STAT   TIME COMMAND
    1 ?        S      0:00 /usr/sbin/sshd -D
    7 ?        Ss     0:00 sshd: root@pts/0
   19 pts/0    Ss     0:00  \_ -bash
  115 pts/0    R+     0:00      \_ ps axf
  110 ?        Ss     0:00 /usr/bin/redis-server /etc/redis/redis.conf
root@c5cec4b035ec:~#
```

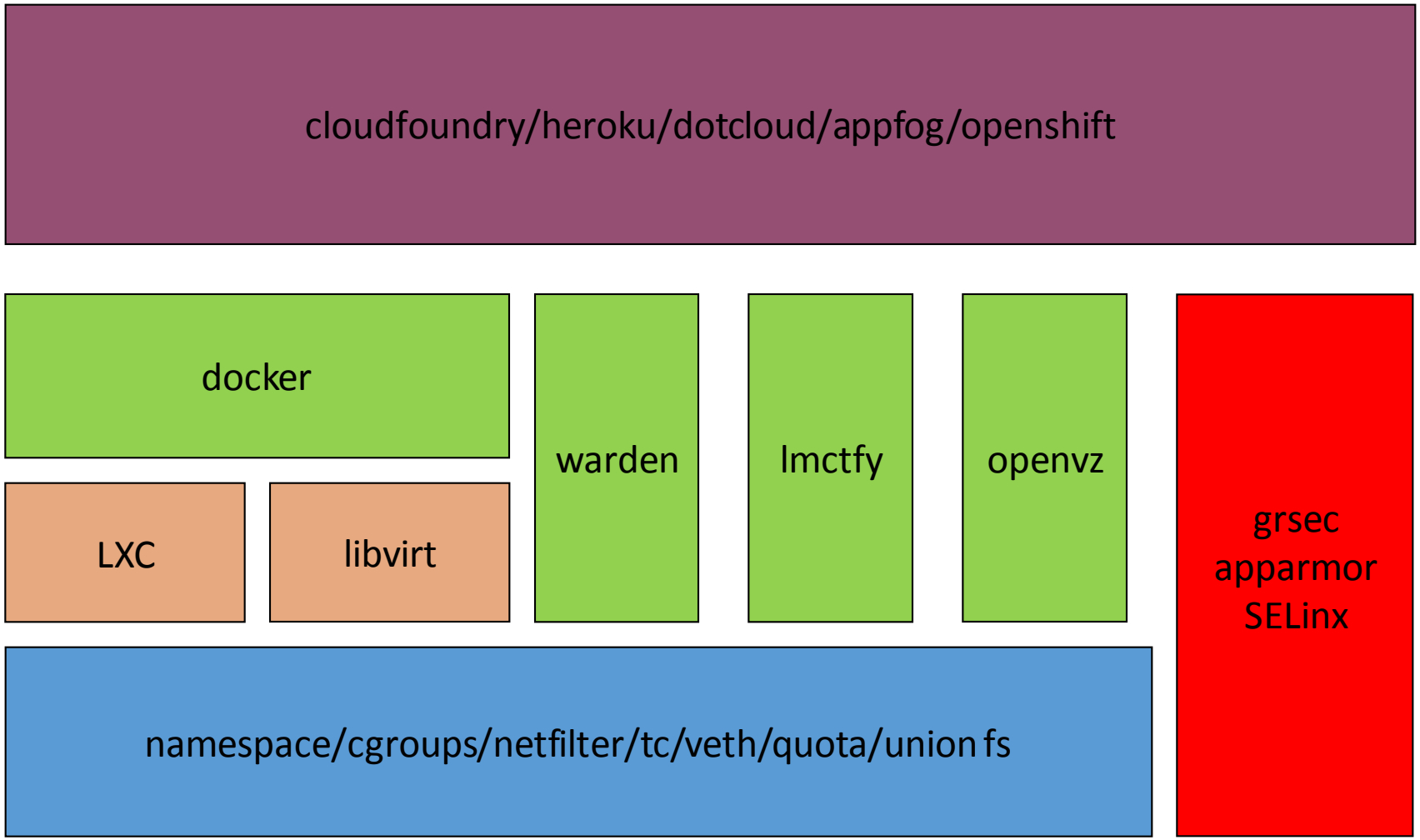
```
11869 ?        S      0:00      \_ lxc-start -n c5cec4b035ec1e7a1816fadaec9432eaa16d165
11881 ?        S      0:00          \_ /usr/sbin/sshd -D
20626 ?        Ss     0:00              \_ /usr/bin/redis-server /etc/redis/redis.conf
```

```
11869 ?        S      0:00      \_ lxc-start -n c5cec4b035ec1e7a1816fadaec9432eaa16d165d5
11881 ?        S      0:00      |   \_ /usr/sbin/sshd -D
20626 ?        Ss     0:00      |       \_ /usr/bin/redis-server /etc/redis/redis.conf
17248 ?        S      0:00      \_ lxc-start -n a972e88e548618d6eea534782dc8b5d99e876eb84
17263 ?        S      0:00      |   \_ /usr/sbin/sshd -D
```

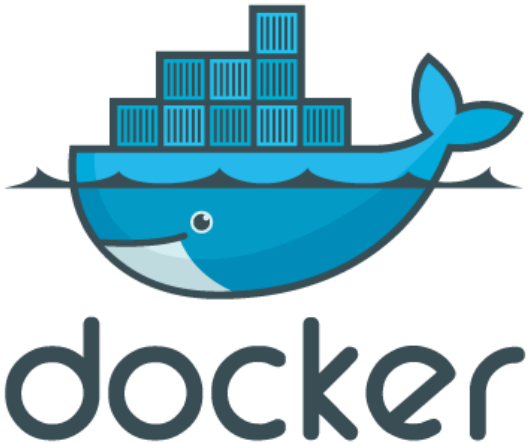
Lightweight virtualization --technologies and projects

- Underlying technology:
  - namespace/cgroups
  - veth
  - union fs(AUFS)
  - netfilter/chroot/tc/quota
- Low-level container management
  - LXC/libvirt
- Security related
  - grsec/apparmor/SELinux
- High-level container/image management
  - docker/warden/lmctfy/openVZ

Lightweight virtualization -- technologies and projects



BAE and docker
--what is docker



- A complete solution for Lightweight virtual machine.
- Open source project developed by dotCloud
- https://www.docker.io/
- https://github.com/dotcloud/docker
- Introduced about 6 months ago, ranked first in language activity by Github GO language

BAE and docker
--what is docker

- Based on LXC tools , but easier to use
- AUFS: speedy creation of container, cool image management.
- Client-Server Architecture
- REST API: Clear interface
- Command-line tool: easy to use

BAE and docker
--what is BAE

- http://developer.baidu.com
- Baidu PAAS platform for developers

BAE and docker

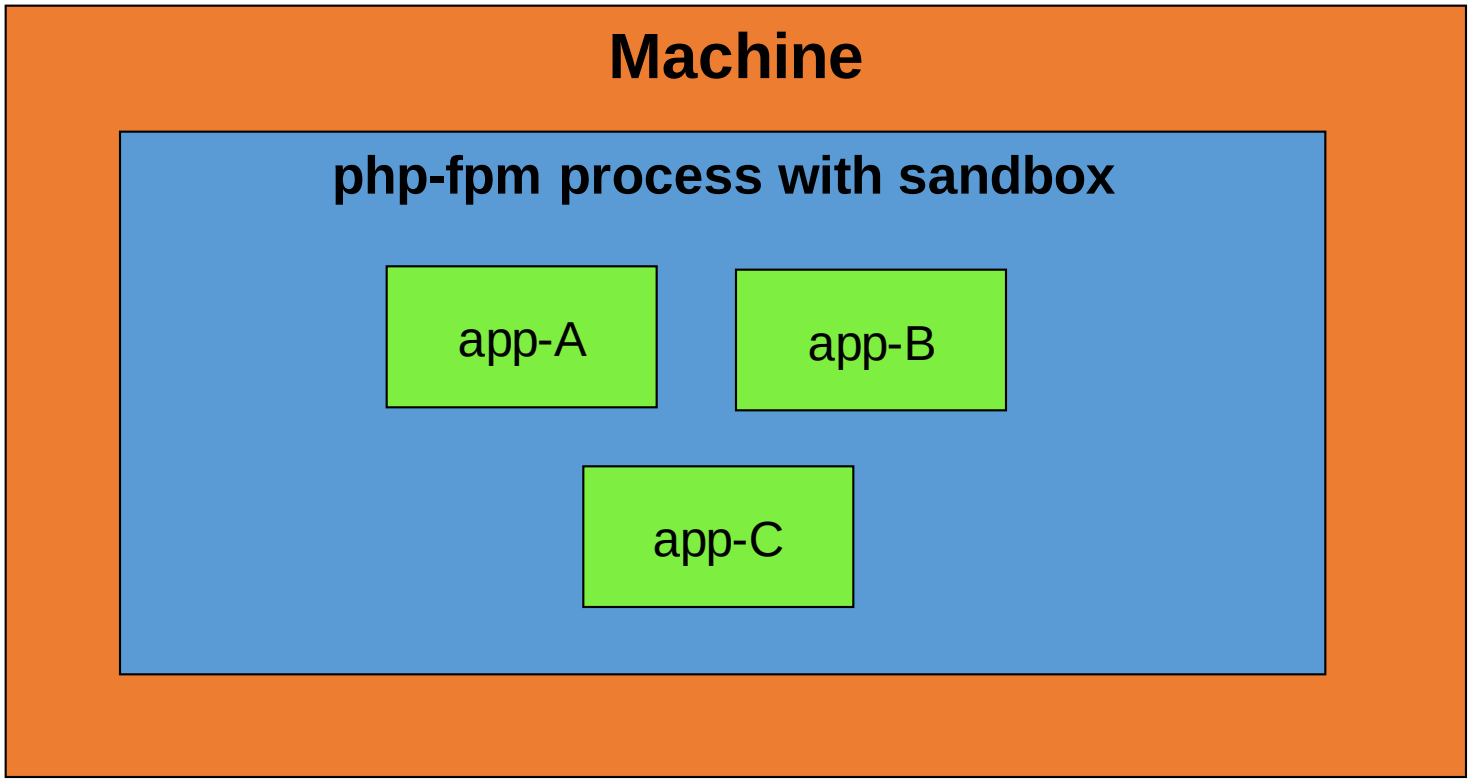
--why does BAE choose docker

- The dilemma of traditional PAAS
  - GAE is the instigator
  - Resource isolation and resource constraints are achieved through **sandbox technology**
  - High cost of platform development and maintenance
  - Many limitations, and high cost of learning
  - High cost of application migration and development
  - Developers complain much

BAE and docker

--why does BAE choose docker

- PAAS based on sandbox techology



BAE and docker

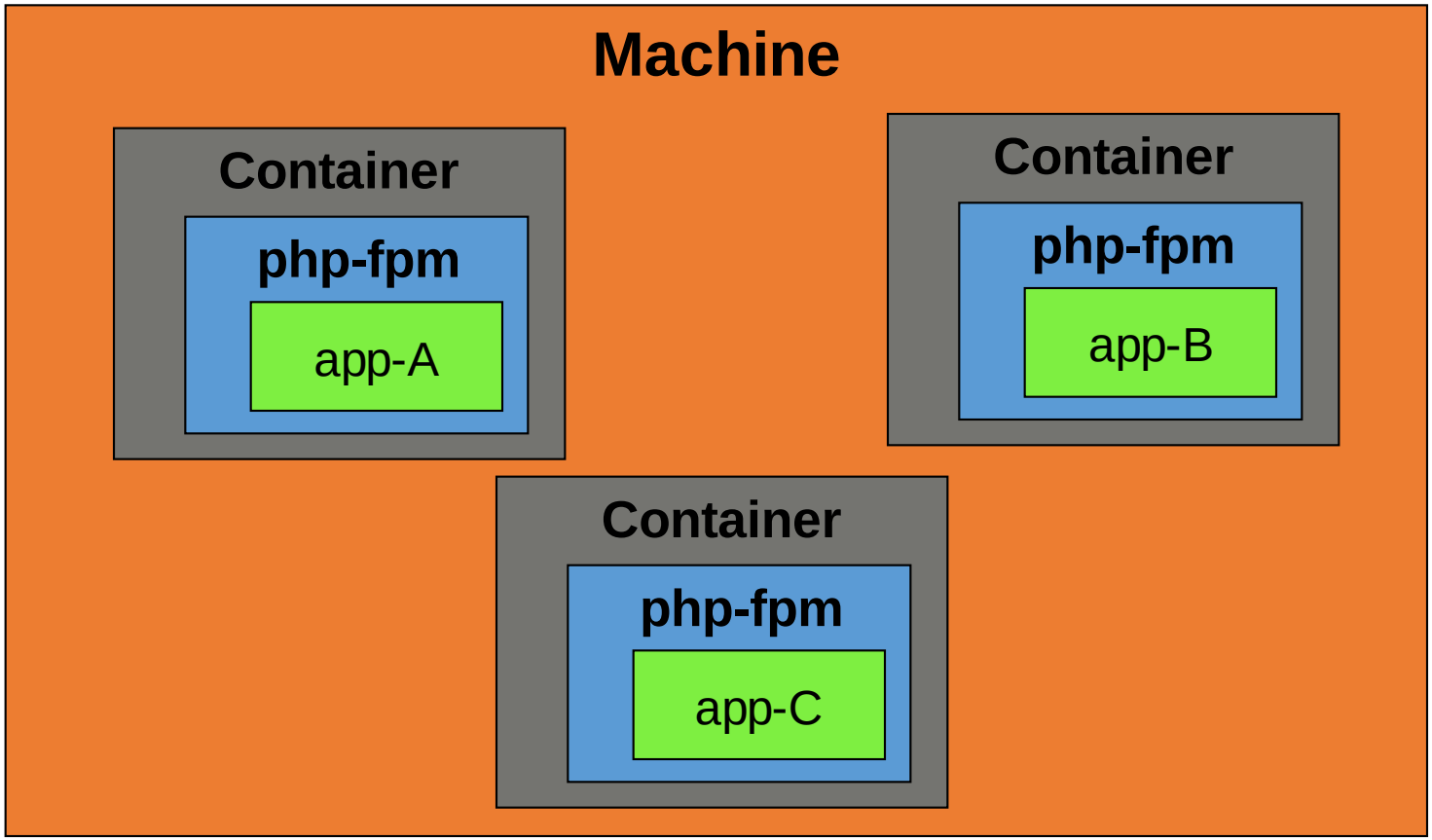
--why does BAE choose docker

- Container technology brought the light:
  - Get rid of the sandbox; through **Container technology**, resource is isolated and limited in the outer layer
  - No language level restrictions, significantly reduce development costs
  - Supporting new programming language made extremely simple
  - cost of platform development and maintenance is significantly lowered
- Industry trends: Emerging PAAS platform have chosen Container
  - Cloudfoudry/openshift/heroku/dotcloud/appfog

BAE and docker

--why does BAE choose docker

- PAAS based on container technology



BAE and docker

--why does BAE choose docker

- BAE2.0 platform is a sandbox-based PAAS
- We are deeply troubled by the sandbox
- Noting Container technology, began preliminary research and exploration

BAE and docker

-- technology option one: to develop by ourself

- Internal virtual machine team gave the following solution:
  - openstack + libvirt
  - Functionally, it meets the basic needs
- Problems:
  - the most important requirements are not addressed:
    - Need frequent creation and deletion of Container
    - Need to create and delete container within a few seconds
    - Actually creation of the Container takes more than 15 seconds (not acceptable)
  - Openstack superfluous
  - Not enough confidence in the quality of the code
  - Lack of follow-up technical support

BAE and docker
-- technology option two: warden

- The warden from cloudfoundry
  - warden is a more complete solution
  - However:
    - not familiar with Ruby
    - deang tightly coupled with warden
    - Not active enough in the technical community

BAE and docker
-- get to know docker

- By chance, talked to docker evangelist Jerome Petazzoni
- Assessment for docker
  - Functionally, it meets the main needs
  - Intrepid: Virtual machine can be quickly created and deleted
  - Intrepid: incremental image management capabilities
  - Code easier to read; has confidence to solve problem
  - tech community are very active and all view it with good future

BAE and docker
-- docker in practice

- Docker's resource isolation meet the basic needs
- the main development work:
  - better resource limitation
  - More comprehensive security restrictions
    - for the public PAAS platform, security is the most important.

BAE and docker
-- docker in practice

- Better resource limitation:
  - memory
  - CPU
  - Disk space
    - quota
  - Disk read and write
    - blkio
  - Network bandwidth
    - tc
  - setrlimit

BAE and docker
-- docker in practice

- More comprehensive security restrictions
  - **grsec: most important**
  - **apparmor:**
  - **LXC tools:**
    - lxc.drop_capabilities
    - Lxc.device.deny
  - **strict iptables rules**
  - **account managment**
    - random root password
    - deny root login
  - **scanning for suspicious running processes**

BAE and docker
-- docker in practice

- using docker private image registry to take care of image management problem

BAE and docker
-- docker in practice

- the problem we meet and solved:
  - containers can't work after docker server restart
    - our patch has been accepted
  - unstable during pressure test:
    - create iptable rule may fail
    - container can't be stopped or deleted sometimes
  - too many threads created by docker server

the development of docker
-- ecosystem thriving

- CoreOS
- Yandex – Cocaine
- Flynn - https :/ / flynn.io /
- The latest version of OpenStack Havana has the native support for docker
- ...

the development of docker
-- the cooperation with Redhat

- Docker’s key step:
  - take full advantage of the powerful network management capabilities of libvirt
  - Use the device-mapper technology to remove the dependence on AUFS
  - Use SELinux to resolve security issues
  - will be able to run on Red Hat's Linux distributions
  - Openshift integrated support for docker

the development of docker
-- the main problem to be solved

- Security
  - user namespace is used to solve root privilege problem
  - using SELinux for security
- Support more Linux distributions
  - With AUFS constraints, currently only supports ubuntu
- Support more architectures
  - Currently only supports x86-64

docker forecast

- Will become the leader in this field
- Cloudfoundry, openshift will support docker
- open source PAAS project based on docker will be born, and be developed with Go language
- A growing number of emerging open source project will commence on docker

Reference

- http://www.infoq.com/articles/docker-containers
- http://en.wikipedia.org/wiki/Operating_system-level_virtualization
- http://en.wikipedia.org/wiki/LXC
- http://en.wikipedia.org/wiki/Cgroups
- http://en.wikipedia.org/wiki/Linux_Containers
- http://en.wikipedia.org/wiki/Aufs
- http://libvirt.org/
- http://linuxcontainers.org/
- https://wiki.ubuntu.com/LxcSecurity
- http://lwn.net/Articles/236038/

Reference

- http://blog.dotcloud.com/under-the-hood-linux-kernels-on-dotcloud-part
- http://blog.dotcloud.com/kernel-secrets-from-the-paas-garage-part-24-c
- http://www.nsnam.org/wiki/index.php/HOWTO_Use_Linux_Containers_to_set_up_virtual_networks
- http://openvz.org/Main_Page
- http://aufs.sourceforge.net/
- http://blog.docker.io/2013/08/containers-docker-how-secure-are-they/