

Docker 기초

운행도

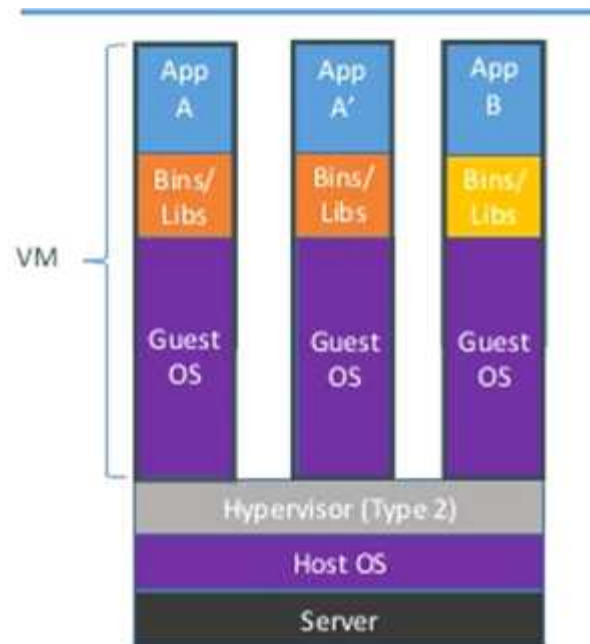
서버 환경을 구축해보자

서버 환경을 구축해보자

1. OS설치 후 소프트웨어 및 기본 설정

서버 환경을 구축해보자

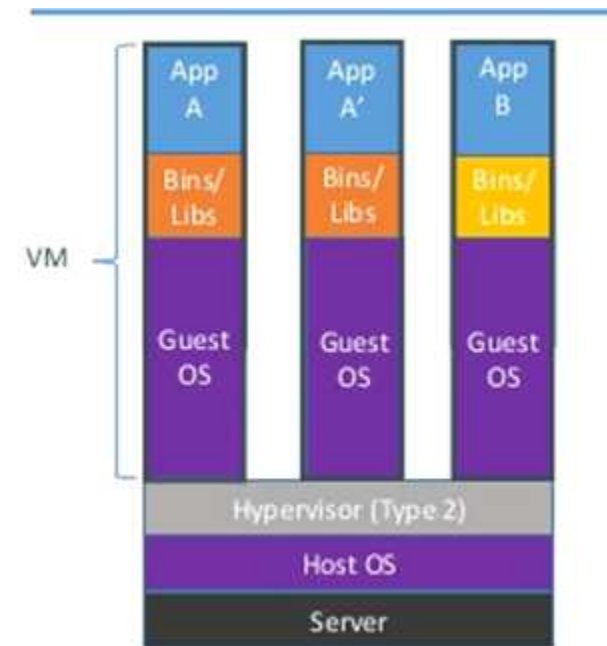
VIRTUREBOX,VMWARE에 이미지로 스냅샷 등 저장을하고
로드 후 설정만 바꿔서 다시 사용



서버 환경을 구축해보자

설치하는 방법이 동일하나
수동으로 IP등 설정값을 바꿔주는게 IT스럽지 않네여..

이 과정을 Script로 정리해서 가상머신에 새 이미지를 생성하고
Script에 정리된 설치 및 IP,도메인,
가상머신 설정을 자동화하게한다.(Vagrant)
여전히 GHEST OS가 생성됨 무거움



Vagrant

1. 이미지 기반
2. 이미지(MYSQL,Tomcat까지 깔린 각각의 이미지)를 로드후 설정을 세팅(Chef)
3. 각 이미지는 Box라는 툴을 통해 저장하고
원격에서 VagrantFile 과 함께 사용가능

```
$ Vagrant.configure (VAGRANTFILE_API_VERSION) do | config |  
  config.vm.box = "centos64"
```

```
  // 네트워크 설정
```

```
  config.vm.network : private_network, ip : "192.168.33.10"
```

```
  // GUI 모드의 설정
```

```
  config.vm.provider : virtualbox do | vb |
```

```
    vb.gui = true
```

```
  end
```

```
end
```

Vagrant

```
config.vm.provision "shell", inline: <<-SHELL
  echo "Install system packages..."
  yum -y update
  yum -y install wget
  yum -y install epel-release
  wget http://rpms.famillecollet.com/enterprise/remi-release-7.rpm
  rpm -Uvh remi-release-7.rpm
  echo "Install redis..."
  yum -y --enablerepo=remi install redis
  systemctl enable redis
  echo "Install mysql..."
  wget http://dev.mysql.com/get/mysql57-community-release-el7-7.noarch.rpm
  yum -y install mysql57-community-release-el7-7.noarch.rpm
  yum -y install mysql-community-server
  systemctl enable mysqld
  echo "Dont forget mysql secure..."
  echo "Install node.js..."
  curl --silent --location https://rpm.nodesource.com/setup_6.x | bash -
  yum install -y nodejs
  echo "Install nginx..."
```

서버 환경을 구축해보자

1. Guset OS를 생성하지 않고 원 호스트 OS의 커널을 사용하면 좋겠다.
톰캣, MYSQL의 기능만 쓸건데 OS가 갯수만큼 설치되니까 낭비.
2. 어썬 apt-get으로 다 설치하고 설정하는데..
이미지 공유 말고 스크립트만 공유하면 안될까?
3. 설치를 하는데 우분투, NANO, VIM, JDK까지 설치하는 동일하다.
근데 다른 이미지에 같은 사양의 소프트웨어가
이미 다운받아져 있는데 캐싱 해서 더 빠르게 설치 하면 좋지 않을까?

서버 환경을 구축해보자(요약)

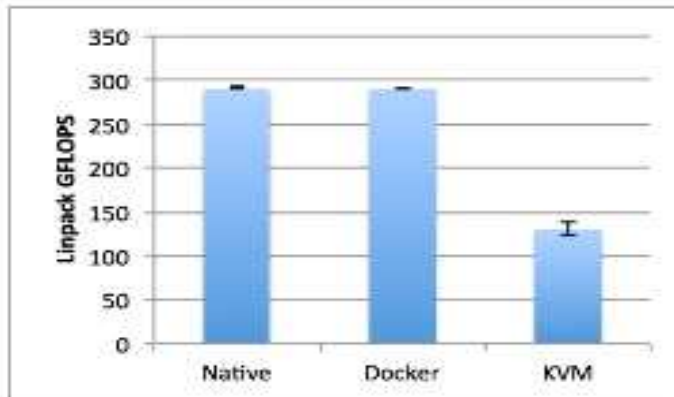
1. OS설치 후 소프트웨어 및 기본 설정
2. VIRTUREBOX,VMWARE에 이미지로 스냅샷 등 저장을하고
로드 후 설정만 바꿔서 다시 사용
3. 설치하는 방법이 동일하니 이 과정을 Script로 정리해서 가상머신에 새 이미지를 생성하
고 Script에 정리된 설치 및 설정을 자동화하게한다.(Vagrant)
계속 GHEST OS가 생성됨 무거움
- 4.설치를 하는데 우분투,NANO,VIM,JDK까지 설치는 동일하다.
그 다음에 오라클서버는 오라클,톰캣은 톰캣 을 설치하고 환경구축을하는데
계속 우분투,VIM,JDK를 새로 받아야하나,,
설치 과정을 이력관리하여 바뀐부분 부터만 새로 받도록 하면 좋겠다.(Docker)
원 호스트 OS의 커널을 사용함

Docker 성능

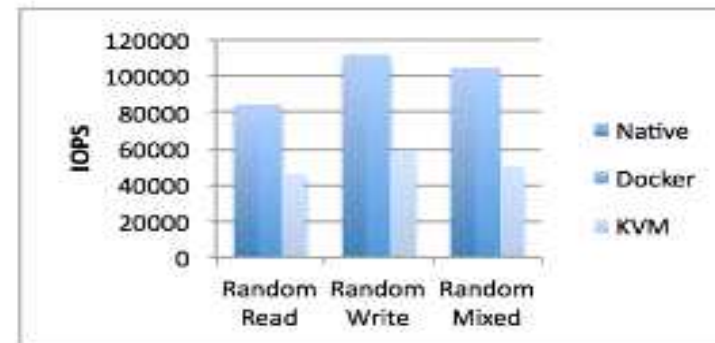
■ Docker vs VM(성능 비교)

■ Docker vs KVM: CPU 및 IO 성능 비교

- Linpack 틀을 이용해 **CPU**의 연산 부하를 준 경우
 - Docker: Native와 비슷
 - KVM: 큰 성능 저하 발생
- **IOPS**(Input/Output Per Second) 측정
 - Docker: Native와 비슷
 - KVM: 큰 성능 저하 발생



<CPU 테스트>

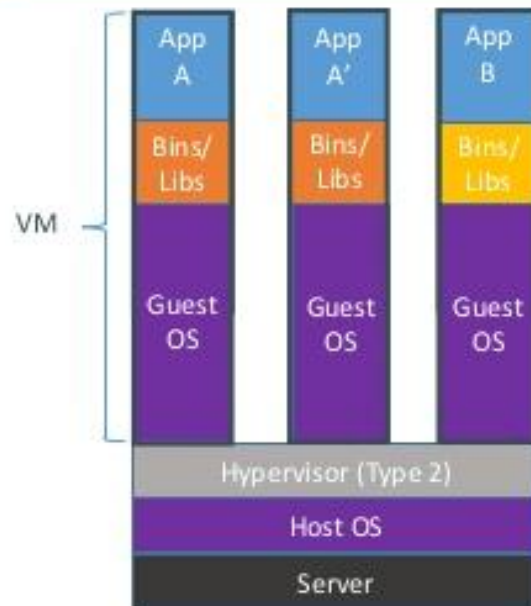


<Block I/O 테스트>

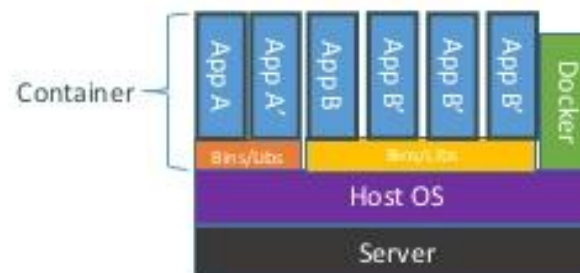
자료 출처: [http://domino.research.ibm.com/library/cyberdig.nsf/papers/0929052195DD819C85257D2300681E7B/\\$File/rc25482.pdf](http://domino.research.ibm.com/library/cyberdig.nsf/papers/0929052195DD819C85257D2300681E7B/$File/rc25482.pdf)

Docker 빠른 이유

Containers vs. VMs



Containers are isolated, but share OS and, where appropriate, bins/libraries

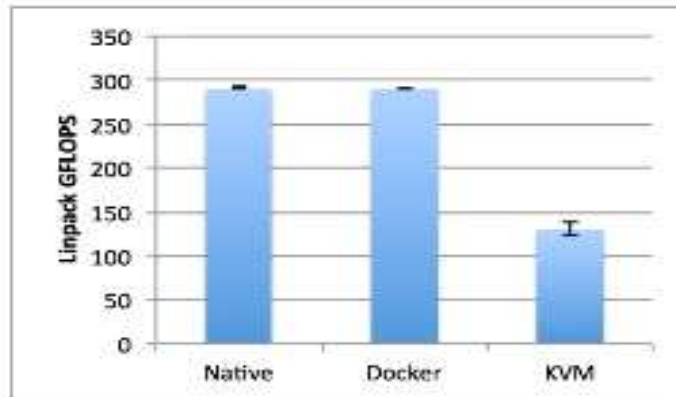


Docker 성능

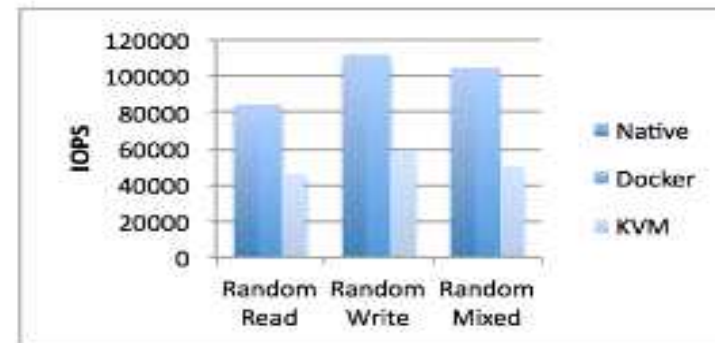
■ Docker vs VM(성능 비교)

■ Docker vs KVM: CPU 및 IO 성능 비교

- Linpack 틀을 이용해 **CPU**의 연산 부하를 준 경우
 - Docker: Native와 비슷
 - KVM: 큰 성능 저하 발생
- **IOPS**(Input/Output Per Second) 측정
 - Docker: Native와 비슷
 - KVM: 큰 성능 저하 발생



<CPU 테스트>



<Block I/O 테스트>

자료 출처: [http://domino.research.ibm.com/library/cyberdig.nsf/papers/0929052195DD819C85257D2300681E7B/\\$File/rc25482.pdf](http://domino.research.ibm.com/library/cyberdig.nsf/papers/0929052195DD819C85257D2300681E7B/$File/rc25482.pdf)

설치

- <https://store.docker.com/editions/community/docker-ce-desktop-windows>
- 선택사항에서 linux base로 설치

시작(정석)

1. Dockerfile
 빌드
2. Docker Image
 run(vm의 mount)
3. bash에서 개발 후
4. commit
5. push (레파지토리에 이미지 저장)
6. 2번으로 ...

Dockerfile

```
FROM ruby:2.3.0
```

```
RUN apt-get update -qq && apt-get install -y build-essential libmysqlclient-dev
```

```
RUN mkdir /inventory_manager
```

```
WORKDIR /inventory_manager
```

```
ADD Gemfile /inventory_manager/Gemfile
```

```
ADD Gemfile.lock /inventory_manager/Gemfile.lock
```

```
RUN bundle install
```

```
ADD . /inventory_manager
```

Dockerfile Build

```
● docker-compose build
db uses an image, skipping
Building web
Step 1 : FROM ruby:2.3.0
---> ca7e58463fba
Step 2 : RUN apt-get update -qq && apt-get install -y build-essential libmysqlclient-dev
---> Using cache
---> 87968d506955
Step 3 : RUN mkdir /inventory_manager
---> Using cache
---> e052938af823
Step 4 : WORKDIR /inventory_manager
---> Using cache
---> 024956478f8f
Step 5 : ADD Gemfile /inventory_manager/Gemfile
---> 9aa16b7f91bf
Removing intermediate container dedceb09a509
Step 6 : ADD Gemfile.lock /inventory_manager/Gemfile.lock
---> b22f4626893d
Removing intermediate container e83ebcc210a0
Step 7 : RUN bundle install
---> Running in 5bf14735f4bc
Don't run Bundler as root. Bundler can ask for sudo if it is needed, and
installing your bundle as root will break this application for all non-root
users on this machine.
Fetching gem metadata from https://rubygems.org/.....
Fetching version metadata from https://rubygems.org/...
Fetching dependency metadata from https://rubygems.org/..
Resolving dependencies...
```


간편하게 시작

1. Kinematic에서 원하는 이미지 실행
2. 설정(볼륨 등) 맞춤
3. EXEC클릭하여 bash에서 개발 후
4. commit(로컬에는 계속 저장되어있음)
5. push (레파지토리에 이미지 저장)

DockerHub

다른 사람이 공유해논 곳 부터 시작가능함
FROM davidcaste/alpine-java~~



84)

All			
	2.0K STARS	10M+ PULLS	> DETAILS
ed build	21 STARS	100K+ PULLS	> DETAILS
	1 STARS	1M+ PULLS	> DETAILS
ed build	3 STARS	1M+ PULLS	> DETAILS
urce	0 STARS	1M+ PULLS	> DETAILS

davidcaste/alpine-tomcat ☆

Last pushed: 8 months ago

Repo Info Tags Dockerfile Build Details

Dockerfile

```
FROM davidcaste/alpine-java-unlimited-jce:jre8
```

```
MAINTAINER David Castellanos <davidcaste@gmail.com>
```

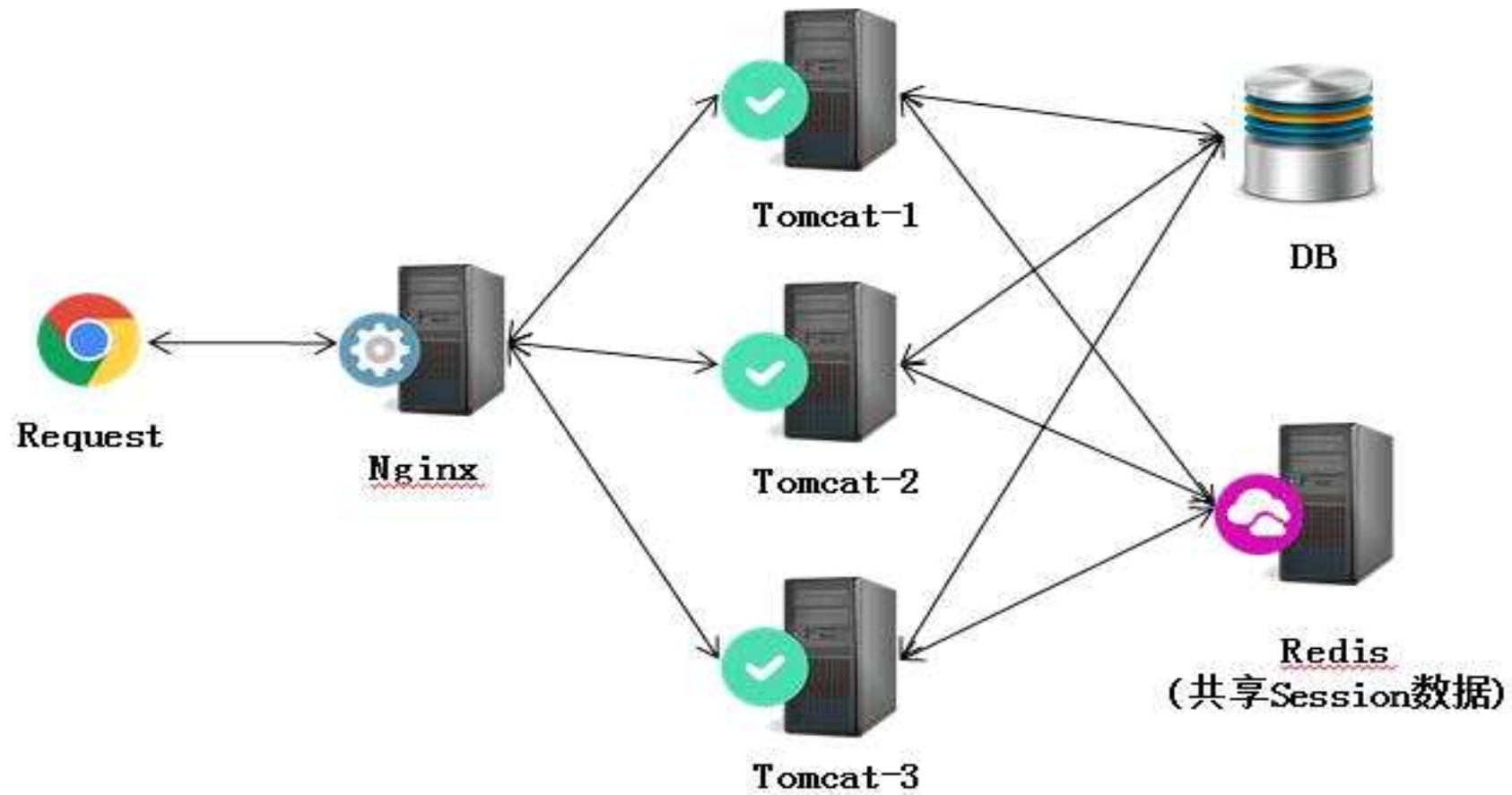
```
ENV TOMCAT_MAJOR=8 \  
    TOMCAT_VERSION=8.5.3 \  
    TOMCAT_HOME=/opt/tomcat \  
    CATALINA_HOME=/opt/tomcat \  
    CATALINA_OUT=/dev/null
```

```
RUN apk upgrade --update && \  
    apk add --update curl && \  
    curl -jksSL -o /tmp/apache-tomcat.tar.gz http://archive.apache.org/dist/tomcat/tomcat-${  
gunzip /tmp/apache-tomcat.tar.gz && \  
    tar -C /opt -xf /tmp/apache-tomcat.tar && \  
    ln -s /opt/apache-tomcat-${TOMCAT_VERSION} ${TOMCAT_HOME} && \  
    rm -rf ${TOMCAT_HOME}/webapps/* && \  
    mv /opt/apache-tomcat-${TOMCAT_VERSION}/webapps/* ${TOMCAT_HOME}/webapps/ &&
```

실습 1

1. MYSQL 설치 후 HediSql로 접속(3306)
2. Tomcat설치
3. Nginx를 Tomcat 연동 (link)
4. Redis(세션관리) 싱크

속제



마치며

1. 이미지 공유 보다는
DockerFile로 관리를 추천
2. 리모트 서버들을 관리하기 위한 툴
(Kubernetes(google), Swarm)
<https://portainer.io/> or Swarmkit
3. AWS 및 GCP에서 Docker컨테이너 기반 클
라우드 서비스 운영중

참고-1

가장 빨리 만나는 Docker

<http://pyrasis.com/docker.html>

Vagrant와 Docker

<http://khanrc.tistory.com/entry/Docker-Vagrant>

참고-2 명령어

```
docker tag nginx henen/nginx:0.2
```

```
docker commit -m "add apt-get install nano" 47ee9c3a9717  
enen/nginx:0.2
```

```
docker push henen/nginx:0.2
```

```
docker diff b728d2e2ab00
```

```
docker ps
```

```
docker ps -al
```