Ubuntu Docker 使用笔记

**Docker安装：**

apt-get update

apt-get install apt-transport-https ca-certificates

apt-get install curl software-properties-common

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | apt-key add -

add-apt-repository \

       "deb [arch=amd64] https://download.docker.com/linux/ubuntu \

       $(lsb\_release -cs) \

       stable"

apt-get update

apt-cache madison docker-ce

apt-get install docker-ce

docker run hello-world

**加速器：**

注册阿里云账号，获得加速地址

编辑/etc/docker/daemon.json

{

"registry-mirrors": [

"https://13jscqda.mirror.aliyuncs.com"

],

"insecure-registries": [],

"debug": true,

"experimental": true

}

**Mysql安装配置**

docker search mysql

docker pull mysql

docker images

docker run --name mysql -e MYSQL\_ROOT\_PASSWORD=123456 -p 3306:3306 -d mysql

将数据库持久化

docker run --name mysql -e MYSQL\_ROOT\_PASSWORD=123456 -p 3306:3306 -v /home/myproject/databases:/var/lib/mysql -d mysql

1.查看mysql的版本

mysql> status

2. 进行远程连接授权；

Use mysql

Select host,user,plugin from user;

mysql> grant all on \*.\* to 'root'@'%';

mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY 'password' PASSWORD EXPIRE NEVER;

ALTER USER 'root'@'%' IDENTIFIED WITH mysql\_native\_password BY '123456';

（密码改为123456）

5. 设置完成可以连接数据库

**防火墙配置**

apt-get install firewalld

firewall-cmd --zone=public --list-port

firewall-cmd --zone=public --add-port=3306/tcp –permanent

firewall-cmd –reload

docker run -ti --name node-1 -v $(pwd):/workspace node bash

docker run -ti --name vue-1 -v $(pwd):/workspace ebiven/vue-cli bash

**docker图形界面管理工具**

一、docker UI

docker search dockerui

docker pull abh1nav/dockerui

docker run -d --privileged --name dockerui -p 9000:9000 -v /var/run/docker.sock:/var/run/docker.sock abh1nav/dockerui

1. portainer

docker search portainer

docker pull portainer/portainer

docker run --name portainer -d -p 9000:9000 -v /var/run/docker.sock:/var/run/docker.sock portainer/portainer

<http://192.168.9.67:9000>

**VUE环境部署**

docker pull ebiven/vue-cli

#docker run -v /home/myproject/workspace:/code -p 8080:8080 -d ebiven/vue-cli

docker run --name myvue -v /home/workspace/vue/:/web -p 8080:8080 -itd ebiven/vue-cli

docker exec –it <container\_id> bash

vue create myproject

cd myproject

yarn serve or npm run serve

VUE更改配置端口号

创建vue.config.js文件

module.exports = {

devServer: {

port: 9090, // 端口

},

// lintOnSave: false // 取消 eslint 验证

}

Oracle部署

拉取镜像

docker pull absolutapps/oracle-12c-ee

#运行容器

时间圈TZ=Asia/Shanghai

docker run -d -p 8080:8080 -p 1521:1521 --name oracle12c --privileged absolutapps/oracle-12c-ee

docker run -itd -p 8080:8080 -p 1521:1521 --name oracle12c -v /home/workspace/db/oracle12c:/u01/app/oracle --privileged absolutapps/oracle-12c-ee

#容器开启时间很长，需要打开日志查看安装情况

docker logs -f oracle12c

#进入容器内部

docker exec -it oracle12c /bin/bash

lsnrctl status

echo $ORACLE\_SID

oracle12c 数据文件位置：/u01/app/oracle/oradata/orcl

#登录

sqlplus system/oracle@localhost:1521/orcl

**alter user scott account unlock;**

**MSSQLServer部署**

**docker pull mcr.microsoft.com/mssql/server**

docker run --name mssqlserver \

-e 'ACCEPT\_EULA=Y' -e 'MSSQL\_SA\_PASSWORD=Zeda$1234' \

-e 'MSSQL\_PID=Enterprise' -p 1433:1433 \

-d mcr.microsoft.com/mssql/server

docker run --name mssqlserver -v /home/workspace/db/mssql:/var/opt/mssql/data -e 'ACCEPT\_EULA=Y' -e 'MSSQL\_SA\_PASSWORD=Zeda$1234' -e 'ID=Enterprise' -p 1433:1433 -d mcr.microsoft.com/mssql/server

Docker –exec –it mssqlserver bash

**/opt/mssql-tools/bin/sqlcmd -S localhost -U SA -P 'Zeda$1234'**

**dotNet core 2.0部署**

**docker pull microsoft/dotnet**

**docker run -itd --rm -p 8000:8000 --name dotnet --privileged -v /home/workspace/web:/web microsoft/dotnet**

dotnet new mvc -n MyWeb

cd MyWeb

修改Program.cs

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Threading.Tasks;

using Microsoft.AspNetCore.Hosting;

namespace myweb

{

public class Program

{

public static void Main(string[] args)

{

var host = new WebHostBuilder()

.UseKestrel()

.UseContentRoot(Directory.GetCurrentDirectory())

.UseIISIntegration()

.UseUrls("http://\*:8000")

.UseStartup<Startup>()

.Build();

host.Run();

}

}

}

**Dotnet run**

**Windows Server 2008 R2 安装Docker**

1. **下载安装Docker ToolBox**

**下载地址：（**<http://mirrors.aliyun.com/docker-toolbox/windows/docker-toolbox/>**）**

1. **点击【Docker Quickstart Terminal】图标启动**
2. **镜像加速(如果需要pull，每次都需要进行如下操作):**

**docker-machine ssh default**

**sudo vi /etc/docker/daemon.json**

**{**

**"registry-mirrors": [ "https://13jscqda.mirror.aliyuncs.com"**

**],**

**"insecure-registries": [],**

**"debug": true,**

**"experimental": true**

**}**

**docker-machine restart default**

**Tomcat部署**

**docker pull tomcat**

**docker run -itd --name mycat -p 8080:8080 tomcat**

**Subversion Egde 部署**

**Docker pull prong/subversion-edge**

**docker run -d -p 3343:3343 -p 4434:4434 -p 18080:18080 -v /home/svn/data:/opt/csvn/data --name svn-server prong/subversion-edge**

**http：//localhost:3343/csvn**

* 3343 - HTTP CSVN 管理界面
* 4434 - HTTPS CSVN 管理界面 (如果启用了SSL)
* 18080 - Apache Http SVN

这个镜像将csvn的数据目录暴露为卷/opt/csvn/data，如果将其映射为宿主机的一个空文件夹，则初始化脚本将负责将基本配置复制到这个卷。

**GitLab安装部署**

docker pull gitlab/gitlab-ee

mkdir -p /home/gitlab/config

mkdir -p /home/gitlab/logs

mkdir -p /home/gitlab/data

docker run --name gitlab -p 443:443 -p 8088:8088 -v /home/gitlab/config:/etc/gitlab:Z -v /home/gitlab/logs:/var/log/gitlab:Z -v /home/gitlab/data:/var/opt/gitlab:Z -e TZ=Asia/Shanghai -d gitlab/gitlab-ee

docker run --name gitlab -p 443:443 -p 8088:8088 -v /mnt/git/repos/config:/etc/gitlab:Z -v /mnt/git/repos/logs:/var/log/gitlab:Z -v /mnt/git/repos/data:/var/opt/gitlab:Z -e TZ=Asia/Shanghai -d gitlab/gitlab-ee

**Redis安装部署**

docker pull redis

mkdir –p /home/workspace/db/redis/data

docker run -p 6379:6379 -v /home/workspace/db/redis/data:/data --name redis -d redis redis-server --appendonly yes --requirepass "Zeda$1234"

docker exec -it redis redis-cli

**MongoDB安装部署**

docker pull mongo

mkdir –p /home/workspace/db/mongo/configdb

mkdir –p /home/workspace/db/mongo/data

docker run --name mongo -p 27017:27017 -v /home/workspace/db/mongo/configdb:/data/configdb -v /home/workspace/db/mongo/data:/data/db -d mongo --auth

docker exec -it mongo bash

mongo admin

db.createUser({user:'admin',pwd:'Zeda$1234',roles:[{role:'userAdminAnyDatabase',db:'admin'}]})

db.auth(‘admin’,’Zeda$1234’);