www.cnblogs.com

wadeson

github：https://github.com/jsonhc/

昵称：wadeson
园龄：1年5个月
粉丝：29
关注：2
+加关注

<			2018年11月			>
日	一	二	三	四	五	六
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	1
2	3	4	5	6	7	8

搜索

找找看

谷歌搜索

常用链接

我的随笔
我的评论
我的参与
最新评论
我的标签

我的标签

docker(18)
openstack(8)
python(8)
tomcat(8)
saltstack(6)
nginx(5)
ansible(5)
ELK(4)
ftp(3)
jenkins(3)
更多

随笔档案

2018年2月 (1)
2018年1月 (4)
2017年12月 (3)
2017年11月 (20)
2017年10月 (17)
2017年9月 (5)
2017年8月 (20)
2017年7月 (23)

最新评论

1. Re:nginx之配置proxy_set_header
写得很不错。
--xiahui400951

2. Re:docker之docker-machine用法
@zhongqingshen你做ssh认证了吗？...
--wadeson

3. Re:docker之docker-machine用法
亲：我有遇见这个错误：Error creating machine: Error detecting OS: Too many retries waiting for SSH to be availa......
--zhongqingshen

4. Re:jenkins配置从节点
@紫色光的记忆贴图呗，其他环境是否一致，像Jenkins版本之类的环境...
--wadeson

5. Re:jenkins配置从节点

博客园 首页 新随笔 联系 订阅XML 管理

随笔-93 评论-10 文章-0

docker之Dockerfile实践

上一篇介绍了Dockerfile中使用的指令，现在开始进行指令实践

先查看下本地的镜像，选一个作为base image：

```
[root@docker ~]# docker images
REPOSITORY              TAG                 IMAGE ID            CREATED             SIZE
wadeson/centos_nginx    v1                  210a202d37b8        2 hours ago         464MB
nginx                   latest              c59f17fe53b0        4 days ago          108MB
ubuntu                  latest              747cb2d60bbe        3 weeks ago         122MB
centos                  latest              196e0ce0c9fb        6 weeks ago         197MB
```

在某一个目录下面创建一个专门存放此demo的目录，也就是Dockerfile所在的context：

```
[root@docker ~]# mkdir docker_demo
[root@docker ~]# cd docker_demo/
[root@docker docker_demo]# touch Dockerfile
[root@docker docker_demo]# pwd
/root/docker_demo
[root@docker docker_demo]# ll
total 0
-rw-r--r--. 1 root root 0 Nov  1 04:34 Dockerfile
```

接下来就开始编写Dockerfile文件了（注意Dockerfile的D需要大写）

这里以编译nginx提供web服务来构建新的镜像

1、下载nginx源码包到docker_demo这个目录下：

```
[root@docker docker_demo]# ll
total 960
-rw-r--r--. 1 root root      0 Nov  1 04:34 Dockerfile
-rw-r--r--. 1 root root 981687 Oct 17 09:20 nginx-1.12.2.tar.gz
```

2、以下是编写好的Dockerfile v1版：

```
[root@docker docker_demo]# cat Dockerfile
# base image
FROM centos

# MAINTAINER
MAINTAINER json_hc@163.com

# put nginx-1.12.2.tar.gz into /usr/local/src and unpack nginx
ADD nginx-1.12.2.tar.gz /usr/local/src

# running required command
RUN yum install -y gcc gcc-c++ glibc make autoconf openssl openssl-devel
RUN yum install -y libxslt-devel -y gd gd-devel GeoIP GeoIP-devel pcre pcre-devel
RUN useradd -M -s /sbin/nologin nginx

# change dir to /usr/local/src/nginx-1.12.2
WORKDIR /usr/local/src/nginx-1.12.2

# execute command to compile nginx
RUN ./configure --user=nginx --group=nginx --prefix=/usr/local/nginx --with-file-aio --with-
http_ssl_module --with-http_realip_module --with-http_addition_module --with-http_xslt_module --with-
http_image_filter_module --with-http_geoip_module --with-http_sub_module --with-http_dav_module --
with-http_flv_module --with-http_mp4_module --with-http_gunzip_module --with-http_gzip_static_module -
-with-http_auth_request_module --with-http_random_index_module --with-http_secure_link_module --with-
http_degradation_module --with-http_stub_status_module && make && make install

EXPOSE 80
```

3、查看docker_demo目录情况：

你好，我参照你的配置的，但是节点连接SSN被拒绝了是怎么回事，显示一个红叉叉

--紫色光的记忆

阅读排行榜

1. docker之Dockerfile实践(39826)
2. nginx之配置proxy_set_header(21010)
3. python3.6连接oracle数据库(17456)
4. docker-compose初试及命令基础(9738)
5. jenkins配置从节点(7420)

评论排行榜

1. 照着官网来安装openstack pike之environment设置(5)
2. jenkins配置从节点(2)
3. docker之docker-machine用法(2)
4. nginx之配置proxy_set_header(1)

推荐排行榜

1. docker之Dockerfile实践(4)
2. nginx之配置proxy_set_header(2)
3. jenkins配置从节点(1)
4. tomcat+svn+maven+jenkins实现自动构建(1)
5. tomcat结合httpd和nginx(1)

```
[root@docker docker_demo]# ll
total 964
-rw-r--r--. 1 root root   1112 Nov  1 04:58 Dockerfile
-rw-r--r--. 1 root root 981687 Oct 17 09:20 nginx-1.12.2.tar.gz
```

4、执行docker build进行构建：

```
docker build -t centos_nginx:v1 .
```

后面的.代表的是相对路径的当前目录，如果需要全路径则为/root/docker_demo（就是找到Dockerfile文件）

构建成功后，查看新构建的镜像：

```
[root@docker docker_demo]# docker images
REPOSITORY              TAG                 IMAGE ID            CREATED             SIZE
centos_nginx            v1                  78d18f16e757        4 hours ago         464MB
wadeson/centos_nginx    v1                  210a202d37b8        7 hours ago         464MB
nginx                   latest              c59f17fe53b0        4 days ago          108MB
ubuntu                  latest              747cb2d60bbe        3 weeks ago         122MB
centos                  latest              196e0ce0c9fb        6 weeks ago         197MB
```

5、然后使用构建的镜像启动一个container并开启nginx服务：

```
[root@docker docker_demo]# docker run -d centos_nginx:v1 /usr/local/nginx/sbin/nginx -g "daemon off;"
ea5af922378356a5ebff60992f000b186b09d1e8d6a4915b0b8ccf997ca12404
```

然后查看启动的container是否在运行：

```
[root@docker docker_demo]# docker ps -l
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS
PORTS               NAMES
ea5af9223783        centos_nginx:v1     "/usr/local/nginx/..."   7 seconds ago       Up 6 seconds
80/tcp              flamboyant_carson
```

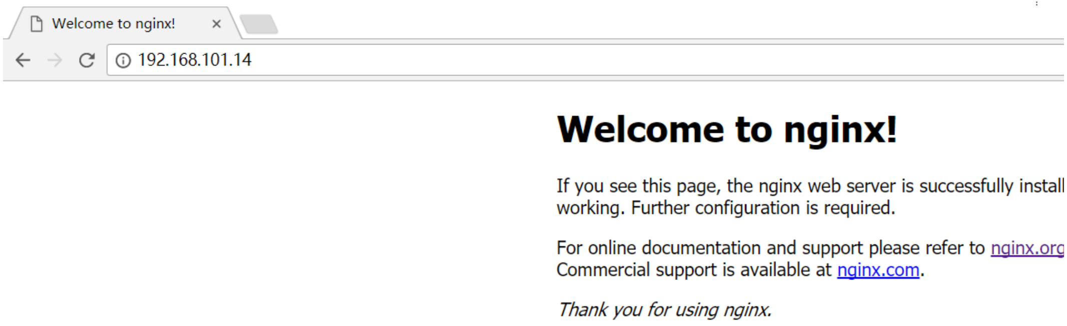
虽然nginx服务开启了，但是port并没有进行映射到本机host，所以这个container并不能进行访问，重新启动一个进行了映射端口的容器

```
[root@docker docker_demo]# docker run -d -p80:80 centos_nginx:v1 /usr/local/nginx/sbin/nginx -g
"daemon off;"
e4b6e4846dedc6f130e028701c84828a635f3b367c0d500ebd947de16b1be0b2
```

再次查看端口映射信息：

```
[root@docker docker_demo]# docker port e4b6e4846ded
80/tcp -> 0.0.0.0:80
```

于是进行访问：



Welcome to nginx!

If you see this page, the nginx web server is successfully install
working. Further configuration is required.

For online documentation and support please refer to nginx.org
Commercial support is available at nginx.com.

Thank you for using nginx.

于是基于Dockerfile的一个简单实例构建完成，现在基于这个Dockerfile文件依次添加其他的指令进行构建

添加ENV环境变量指令：

```
[root@docker docker_demo]# cat Dockerfile
# base image
FROM centos

# MAINTAINER
MAINTAINER json_hc@163.com

# put nginx-1.12.2.tar.gz into /usr/local/src and unpack nginx
ADD nginx-1.12.2.tar.gz /usr/local/src

# running required command
RUN yum install -y gcc gcc-c++ glibc make autoconf openssl openssl-devel
RUN yum install -y libxslt-devel -y gd gd-devel GeoIP GeoIP-devel pcre pcre-devel
RUN useradd -M -s /sbin/nologin nginx

# change dir to /usr/local/src/nginx-1.12.2
WORKDIR /usr/local/src/nginx-1.12.2
```

```
# execute command to compile nginx
RUN ./configure --user=nginx --group=nginx --prefix=/usr/local/nginx --with-file-aio  --with-
http_ssl_module  --with-http_realip_module    --with-http_addition_module    --with-http_xslt_module
--with-http_image_filter_module    --with-http_geoip_module  --with-http_sub_module  --with-
http_dav_module --with-http_flv_module    --with-http_mp4_module --with-http_gunzip_module  --with-
http_gzip_static_module  --with-http_auth_request_module  --with-http_random_index_module   --with-
http_secure_link_module   --with-http_degradation_module   --with-http_stub_status_module && make &&
make install

ENV PATH /usr/local/nginx/sbin:$PATH

EXPOSE 80
```

然后进行构建：

```
[root@docker docker_demo]# docker build -t centos_nginx:v2 .
Sending build context to Docker daemon  985.6kB
Step 1/10 : FROM centos
 ---> 196e0ce0c9fb
Step 2/10 : MAINTAINER json_hc@163.com
 ---> Using cache
 ---> cde1d7830106
Step 3/10 : ADD nginx-1.12.2.tar.gz /usr/local/src
 ---> Using cache
 ---> 1e4d16340af0
Step 4/10 : RUN yum install -y gcc gcc-c++ glibc make autoconf openssl openssl-devel
 ---> Using cache
 ---> 405835ad9b0b
Step 5/10 : RUN yum install -y libxslt-devel -y gd gd-devel GeoIP GeoIP-devel pcre pcre-devel
 ---> Using cache
 ---> 4002738cf7a6
Step 6/10 : RUN useradd -M -s /sbin/nologin nginx
 ---> Using cache
 ---> 02961c5c564d
Step 7/10 : WORKDIR /usr/local/src/nginx-1.12.2
 ---> Using cache
 ---> f1da71a93c5e
Step 8/10 : RUN ./configure --user=nginx --group=nginx --prefix=/usr/local/nginx --with-file-aio  --
with-http_ssl_module  --with-http_realip_module    --with-http_addition_module    --with-
http_xslt_module   --with-http_image_filter_module    --with-http_geoip_module  --with-http_sub_module
--with-http_dav_module --with-http_flv_module    --with-http_mp4_module --with-http_gunzip_module  --
with-http_gzip_static_module  --with-http_auth_request_module  --with-http_random_index_module   --
with-http_secure_link_module   --with-http_degradation_module   --with-http_stub_status_module && make
&& make install
 ---> Using cache
 ---> cd2ad4c45004
Step 9/10 : ENV PATH /usr/local/nginx/sbin:$PATH
 ---> Running in 07ba2f7129bc
 ---> 9588fa1058aa
Removing intermediate container 07ba2f7129bc
Step 10/10 : EXPOSE 80
 ---> Running in 473cd847154a
 ---> 2031faf8894a
Removing intermediate container 473cd847154a
Successfully built 2031faf8894a
Successfully tagged centos_nginx:v2
```

由于在构建的过程中docker会采用缓存的机制，上面的构建过程中包含很多using cache，所以这次构建非常快，如果需要重新构建，不想使用cache需要添加--no-cache

--no-cache   Do not use cache when building the image

查看v2版本的镜像：

```
[root@docker docker_demo]# docker images
REPOSITORY              TAG                 IMAGE ID            CREATED             SIZE
centos_nginx            v2                  2031faf8894a        2 minutes ago       464MB
```

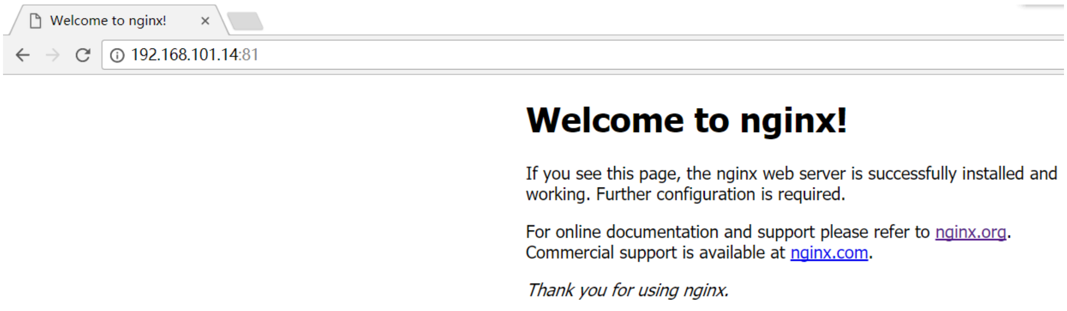
使用v2版本的镜像启动一个container：

```
[root@docker docker_demo]# docker run -d -p81:80 centos_nginx:v2 nginx -g "daemon off;"
da48b465b1b1a14824497d724eee52b8408270b3b5223c5dd7094b7c0cef211d
```

查看container运行状态：

```
[root@docker docker_demo]# docker ps -l
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS
PORTS               NAMES
da48b465b1b1        centos_nginx:v2     "nginx -g 'daemon ..."   23 seconds ago      Up 22 seconds
0.0.0.0:81->80/tcp   determined_neumann
```

进行访问：



Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

上述启动容器的过程中使用的命令docker run -d -p81:80 centos_nginx:v2 nginx -g "daemon off;"为什么这里使用的nginx而不是

/usr/local/nginx/sbin/nginx，因为在Dockerfile文化中执行了PATH=/usr/local/nginx/sbin:$PATH，添加到了环境变量

添加指令CMD：

```
[root@docker docker_demo]# cat Dockerfile
# base image
FROM centos

# MAINTAINER
MAINTAINER json_hc@163.com

# put nginx-1.12.2.tar.gz into /usr/local/src and unpack nginx
ADD nginx-1.12.2.tar.gz /usr/local/src

# running required command
RUN yum install -y gcc gcc-c++ glibc make autoconf openssl openssl-devel
RUN yum install -y libxslt-devel -y gd gd-devel GeoIP GeoIP-devel pcre pcre-devel
RUN useradd -M -s /sbin/nologin nginx

# change dir to /usr/local/src/nginx-1.12.2
WORKDIR /usr/local/src/nginx-1.12.2

# execute command to compile nginx
RUN ./configure --user=nginx --group=nginx --prefix=/usr/local/nginx --with-file-aio  --with-http_ssl_module  --with-http_realip_module    --with-http_addition_module    --with-http_xslt_module   --with-http_image_filter_module    --with-http_geoip_module  --with-http_sub_module  --with-http_dav_module --with-http_flv_module    --with-http_mp4_module --with-http_gunzip_module  --with-http_gzip_static_module  --with-http_auth_request_module  --with-http_random_index_module  --with-http_secure_link_module   --with-http_degradation_module   --with-http_stub_status_module && make && make install

ENV PATH /usr/local/nginx/sbin:$PATH

EXPOSE 80

CMD /bin/sh -c 'nginx -g "daemon off;"'
```

然后进行构建：

```
[root@docker docker_demo]# docker build -t centos_nginx:v3 .
```

查看v3版本的镜像：

```
[root@docker docker_demo]# docker images
REPOSITORY             TAG                 IMAGE ID            CREATED             SIZE
centos_nginx           v3                  0e49a2c0562f        11 seconds ago      464MB
```

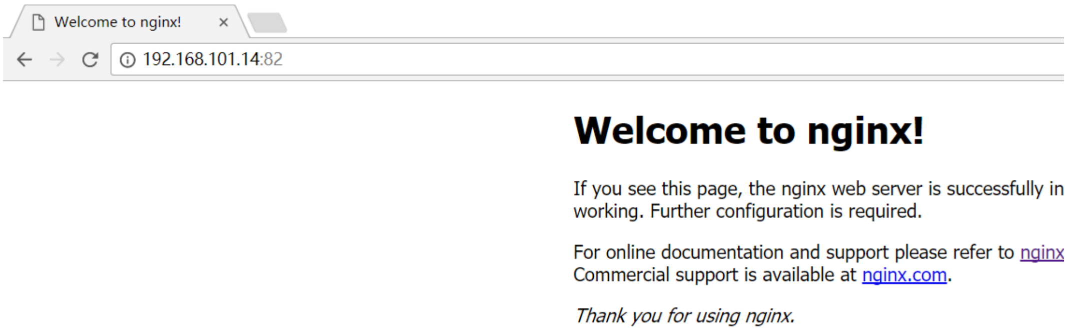
然后基于v3版本的镜像启动一个container：

```
[root@docker docker_demo]# docker run -d -p82:80 centos_nginx:v3
d988c04d04f49b909f28e7b664be3959a0d51b951f1c1b04fcf5c716552b7c41
```

查看启动的容器状态：

```
[root@docker docker_demo]# docker ps -l
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS
PORTS                NAMES
d988c04d04f4        centos_nginx:v3     "/bin/sh -c '/bin/..."   6 seconds ago       Up 5 seconds
0.0.0.0:82->80/tcp   optimistic_saha
```

最后进行访问：



Welcome to nginx!

If you see this page, the nginx web server is successfully in working. Further configuration is required.

For online documentation and support please refer to nginx Commercial support is available at nginx.com.

Thank you for using nginx.

新增加的CMD /bin/sh -c 'nginx -g "daemon off;"'表示

当启动一个container时默认运行的命令，如果在启动container时赋予了command的化，那么

定义的CMD中的命令将不会被执行，而会去执行command的命令

添加entrypoint指令：

```
[root@docker docker_demo]# cat Dockerfile
# base image
FROM centos

# MAINTAINER
MAINTAINER json_hc@163.com

# put nginx-1.12.2.tar.gz into /usr/local/src and unpack nginx
ADD nginx-1.12.2.tar.gz /usr/local/src

# running required command
RUN yum install -y gcc gcc-c++ glibc make autoconf openssl openssl-devel
RUN yum install -y libxslt-devel -y gd gd-devel GeoIP GeoIP-devel pcre pcre-devel
RUN useradd -M -s /sbin/nologin nginx

# change dir to /usr/local/src/nginx-1.12.2
WORKDIR /usr/local/src/nginx-1.12.2

# execute command to compile nginx
RUN ./configure --user=nginx --group=nginx --prefix=/usr/local/nginx --with-file-aio  --with-
http_ssl_module  --with-http_realip_module    --with-http_addition_module    --with-http_xslt_module
--with-http_image_filter_module    --with-http_geoip_module  --with-http_sub_module  --with-
http_dav_module --with-http_flv_module    --with-http_mp4_module --with-http_gunzip_module  --with-
http_gzip_static_module  --with-http_auth_request_module  --with-http_random_index_module  --with-
http_secure_link_module   --with-http_degradation_module   --with-http_stub_status_module && make &&
make install

ENV PATH /usr/local/nginx/sbin:$PATH

EXPOSE 80

ENTRYPOINT ["nginx"]

CMD ["-g","daemon off;"]
```

当ENTRYPOINT和CMD连用时，CMD的命令是ENTRYPOINT命令的参数，两者连用相当于nginx -g "daemon off;"

而当一起连用的时候命令格式最好一致（这里选择的都是json格式的是成功的，如果都是sh模式可以试一下）

开始进行构建v4版本：

```
[root@docker docker_demo]# docker build -t centos_nginx:v4 .
```

查看新建的镜像：

```
[root@docker docker_demo]# docker images
REPOSITORY              TAG                 IMAGE ID            CREATED             SIZE
centos_nginx            v4                  6c5128aaff05        4 minutes ago       464MB
```

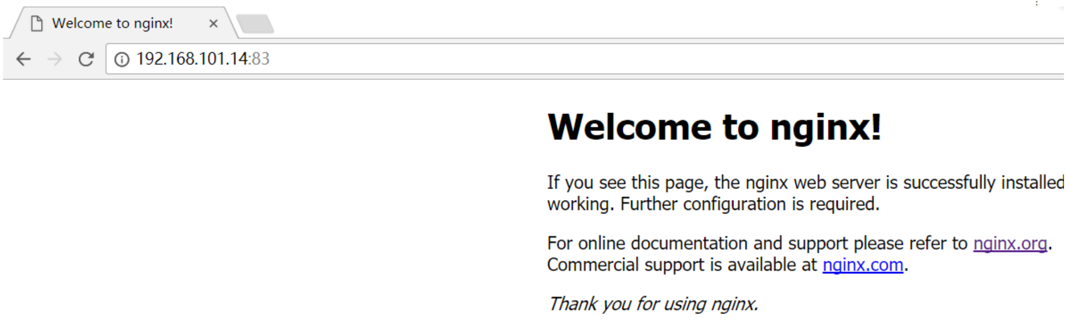
为v4版本的镜像启动一个container：

```
[root@docker docker_demo]# docker run -d -p83:80 centos_nginx:v4
6933c78255f3cebe44d4d5d080caf8a2fde45ded2f9b333ec01cdfe98cd5f417
```

然后查看容器状态：

```
[root@docker docker_demo]# docker ps -l
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS
PORTS               NAMES
6933c78255f3        centos_nginx:v4     "nginx -g 'daemon ..."   6 seconds ago       Up 5 seconds
0.0.0.0:83->80/tcp   zealous_euclid
```

最后进行访问：



Welcome to nginx!

If you see this page, the nginx web server is successfully installed
working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

这里增加一个案例，如果Dockerfile修改成下面：

```
[root@docker docker_demo]# cat Dockerfile
# base image
FROM centos

# MAINTAINER
MAINTAINER json_hc@163.com

# put nginx-1.12.2.tar.gz into /usr/local/src and unpack nginx
ADD nginx-1.12.2.tar.gz /usr/local/src

# running required command
RUN yum install -y gcc gcc-c++ glibc make autoconf openssl openssl-devel
RUN yum install -y libxslt-devel -y gd gd-devel GeoIP GeoIP-devel pcre pcre-devel
RUN useradd -M -s /sbin/nologin nginx

# change dir to /usr/local/src/nginx-1.12.2
WORKDIR /usr/local/src/nginx-1.12.2

# execute command to compile nginx
RUN ./configure --user=nginx --group=nginx --prefix=/usr/local/nginx --with-file-aio  --with-
http_ssl_module  --with-http_realip_module    --with-http_addition_module    --with-http_xslt_module
--with-http_image_filter_module    --with-http_geoip_module  --with-http_sub_module  --with-
http_dav_module --with-http_flv_module    --with-http_mp4_module --with-http_gunzip_module  --with-
http_gzip_static_module  --with-http_auth_request_module  --with-http_random_index_module   --with-
http_secure_link_module   --with-http_degradation_module   --with-http_stub_status_module && make &&
make install

ENV PATH /usr/local/nginx/sbin:$PATH

EXPOSE 80

ENTRYPOINT ["nginx"]

CMD ["-g","daemon on;"]
```

CMD的命令修改为了后台，我们知道如果容器内的进程在后台运行那么容器将不会运行，现在以此构建v5版本镜像：

```
[root@docker docker_demo]# docker build -t centos_nginx:v5 .
```

查看v5版本的新镜像：

```
[root@docker docker_demo]# docker images
REPOSITORY             TAG                  IMAGE ID            CREATED              SIZE
centos_nginx           v5                   5c1131306686        29 seconds ago       464MB
```

现在利用v5版本镜像启动一个container，但是在启动的时候添加后面的command：

```
[root@docker docker_demo]# docker run -d -p85:80 centos_nginx:v5 -g "daemon off;"
5e11edbbd2a0e184f1766c435c0d9b01ef5d74b57e2e2c3a1efed0cf2a2e037b
```

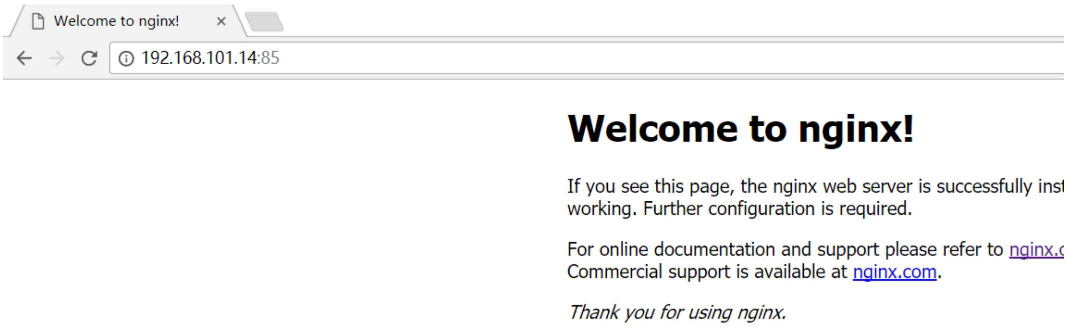
可以看见在后面新增了-g "daemon off;"，前面说过如果增加了命令那么Dockerfile中的CMD中的命令将不会生效

查看容器运行状态：

```
[root@docker docker_demo]# docker ps -l
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS
PORTS                NAMES
5e11edbbd2a0        centos_nginx:v5     "nginx -g 'daemon ..."   3 seconds ago       Up 3 seconds
0.0.0.0:85->80/tcp   nifty_bartik
```

可以看见容器的运行丝毫没有问题，于是nginx的服务依然还是在前台运行，没有被CMD影响，而新增加的command（-g "daemon off;"）

将作为ENTRYPOINT的新的参数以此为准，于是进行访问端口85



Welcome to nginx!

If you see this page, the nginx web server is successfully inst
working. Further configuration is required.

For online documentation and support please refer to nginx.c
Commercial support is available at nginx.com.

Thank you for using nginx.

添加VOLUME指令：

```
[root@docker docker_demo]# cat Dockerfile
# base image
FROM centos

# MAINTAINER
MAINTAINER json_hc@163.com

# put nginx-1.12.2.tar.gz into /usr/local/src and unpack nginx
ADD nginx-1.12.2.tar.gz /usr/local/src

# running required command
RUN yum install -y gcc gcc-c++ glibc make autoconf openssl openssl-devel
RUN yum install -y libxslt-devel -y gd gd-devel GeoIP GeoIP-devel pcre pcre-devel
RUN useradd -M -s /sbin/nologin nginx

# mount a dir to container
VOLUME ["/data"]

# change dir to /usr/local/src/nginx-1.12.2
WORKDIR /usr/local/src/nginx-1.12.2

# execute command to compile nginx
RUN ./configure --user=nginx --group=nginx --prefix=/usr/local/nginx --with-file-aio  --with-http_ssl_module  --with-http_realip_module    --with-http_addition_module    --with-http_xslt_module   --with-http_image_filter_module    --with-http_geoip_module  --with-http_sub_module  --with-http_dav_module --with-http_flv_module    --with-http_mp4_module --with-http_gunzip_module  --with-http_gzip_static_module  --with-http_auth_request_module  --with-http_random_index_module   --with-http_secure_link_module   --with-http_degradation_module   --with-http_stub_status_module && make && make install

# setup PATH
ENV PATH /usr/local/nginx/sbin:$PATH

# EXPOSE
EXPOSE 80

# the command of entrypoint
ENTRYPOINT ["nginx"]

CMD ["-g"]
```

开始进行构建：

```
[root@docker docker_demo]# docker build -t centos_nginx:v6 .
```

查看v6版本的镜像：

```
[root@docker ~]# docker images
REPOSITORY            TAG                 IMAGE ID            CREATED             SIZE
centos_nginx          v6                  959fdf4d4288        35 seconds ago      464MB
```

利用该镜像启动一个container：

```
[root@docker ~]# docker run -d -p 86:80 --name=nginx6 centos_nginx:v6 -g "daemon off;"
6c15a9e93fb1421bdb7eddaabe439996e97415e85a003f80c1d8b4b2c5ee3ffa
```

查看启动的容器状态：

```
[root@docker ~]# docker ps -l
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS
PORTS                NAMES
6c15a9e93fb1        centos_nginx:v6     "nginx -g 'daemon ..."   4 seconds ago       Up 4 seconds
0.0.0.0:86->80/tcp   nginx6
```

利用docker exec进入到container中，查看是否存在卷/data：

```
[root@6c15a9e93fb1 /]# ll
total 16
-rw-r--r--.  8 root root 15836 Sep 11 15:53 anaconda-post.log
lrwxrwxrwx.  1 root root     7 Sep 11 15:51 bin -> usr/bin
drwxr-xr-x.  2 root root     6 Nov  2 01:42 data
```

这个/data挂载的目录对应本机host的这个目录：

```
[root@docker _data]# pwd
/var/lib/docker/volumes/3490a9b7f9773b020343e82c1c4236d976b69eb6a80121eb80612d8c39e02820/_data
```

现在在本机host上的这个目录创建一个文件：

```
[root@docker _data]# touch wadeson.sh
[root@docker _data]# ll
total 0
-rw-r--r--. 1 root root 0 Nov  1 21:45 wadeson.sh
```

然后切换到container中查看是否存在这个文件：

```
[root@6c15a9e93fb1 /]# ll /data/
total 0
-rw-r--r--. 1 root root 0 Nov  2 01:45 wadeson.sh
```

添加ONBUILD指令：

Dockerfile1中base image为A镜像，并在Dockerfile1中定义ONBUILD指令，构建成新镜像B镜像

Dockerfile2中base image为B镜像，构建成新镜像C

当使用镜像B启动的container1不会执行OBNUILD中定义的内容，而使用C镜像启动的container2则会执行ONBUILD定义的内容

```
[root@docker docker_demo]# cat Dockerfile
# base image
FROM centos

# MAINTAINER
MAINTAINER json_hc@163.com

# put nginx-1.12.2.tar.gz into /usr/local/src and unpack nginx
ADD nginx-1.12.2.tar.gz /usr/local/src

# running required command
RUN yum install -y gcc gcc-c++ glibc make autoconf openssl openssl-devel
RUN yum install -y libxslt-devel -y gd gd-devel GeoIP GeoIP-devel pcre pcre-devel
RUN useradd -M -s /sbin/nologin nginx

# mount a dir to container
ONBUILD VOLUME ["/data"]

# change dir to /usr/local/src/nginx-1.12.2
WORKDIR /usr/local/src/nginx-1.12.2

# execute command to compile nginx
RUN ./configure --user=nginx --group=nginx --prefix=/usr/local/nginx --with-file-aio  --with-
http_ssl_module  --with-http_realip_module    --with-http_addition_module    --with-http_xslt_module
--with-http_image_filter_module    --with-http_geoip_module  --with-http_sub_module  --with-
http_dav_module --with-http_flv_module    --with-http_mp4_module --with-http_gunzip_module  --with-
http_gzip_static_module  --with-http_auth_request_module  --with-http_random_index_module   --with-
http_secure_link_module   --with-http_degradation_module   --with-http_stub_status_module && make &&
make install

# setup PATH
ENV PATH /usr/local/nginx/sbin:$PATH

# EXPOSE
EXPOSE 80

# the command of entrypoint
ENTRYPOINT ["nginx"]

CMD ["-g"]
```

使用上面Dockerfile构建镜像版本v7：

```
[root@docker docker_demo]# docker run -d -p 87:80 --name=nginx7 centos_nginx:v7 -g "daemon off;"
48f1fe5c71aefc0f9513e8085af8f5b7cdf14fa986fb3b11f2050f18ceefd26e
```

现在进入到容器内查看是否存在/data：

```
[root@docker ~]# docker exec -it nginx7 /bin/bash
[root@48f1fe5c71ae nginx-1.12.2]# ll /data
```

```
ls: cannot access /data: No such file or directory
```

现在修改上面Dockerfile的FROM基于的base image：

```
[root@docker docker_demo]# cat Dockerfile
# base image
FROM centos_nginx:v7

# MAINTAINER
MAINTAINER json_hc@163.com
```

利用此Dockerfile构建镜像v8：

```
[root@docker docker_demo]# docker build -t centos_nginx:v8 .
Sending build context to Docker daemon  986.6kB
Step 1/2 : FROM centos_nginx:v7
# Executing 1 build trigger...
Step 1/1 : VOLUME /data
 ---> Running in a6187867513d
 ---> 5ac07930be4c
Removing intermediate container a6187867513d
Step 2/2 : MAINTAINER json_hc@163.com
 ---> Running in e02dbf8219cf
 ---> 6f792dc07c35
Removing intermediate container e02dbf8219cf
Successfully built 6f792dc07c35
Successfully tagged centos_nginx:v8
```

可以看见卷/data已经执行了操作，现在启动一个container：

```
[root@docker docker_demo]# docker run -d -p 88:80 --name=nginx8 centos_nginx:v8 -g "daemon off;"
6c2a847c5f6b59b02f91afecadbfc15c88d1217a477c0421a424bce6e5eb317a
```

查看容器状态，并进入到容器验证/data目录：

```
[root@docker docker_demo]# docker ps -l
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              
PORTS                NAMES
6c2a847c5f6b        centos_nginx:v8     "nginx -g 'daemon ..."   37 seconds ago      Up 37 seconds       
0.0.0.0:88->80/tcp   nginx8
[root@docker docker_demo]# docker exec -it nginx8 /bin/bash
[root@6c2a847c5f6b nginx-1.12.2]# ll /data/
total 0
```

由此可见镜像v8包含了v7 的所有内容，并且增加了ONBUILD的内容

标签: docker

好文要顶 关注我 收藏该文

wadeson
关注 - 2
粉丝 - 29
+加关注

4
推荐

0
反对

« 上一篇：docker之Dockerfile指令介绍
» 下一篇：docker之数据卷管理

posted on 2017-11-01 17:50 wadeson 阅读(39841) 评论(0) 编辑 收藏

刷新评论 刷新页面 返回顶部

发表评论

昵称： c332030

评论内容：

提交评论 退出 订阅评论

[Ctrl+Enter快捷键提交]

相关博文：

· Docker探索系列2之镜像打包与DockerFile
· Centos7 docker 常用指令
· Docker基础-搭建本地私有仓库
· Docker Commands
· docker之手动构建新的镜像

最新新闻：

· 天猫精灵的目标，不是诗和远方，是「智联万物」
· 保卫豆瓣：今日互联网不允许有中立者
· 专访网易有道CEO：有道翻译王将登陆日本 发力K12教育
· 不要钱还免安装！PS杀手火了，网友：作者是上帝么？
· 6000首歌曲下架KTV，音集协成“众矢之的”？
» 更多新闻...