

EPAM Systems, RD Dep., R&D Dep.  
DevOps LAB

Docker. Day 1

REVISION HISTORY					
Ver.	Description of Change	Author	Date	Approved	
				Name	Effective Date
<1.0>	According to training notes	Siarhei Beliakou	27-Feb-2019		

Legal Notice

This document contains privileged and/or confidential information and may not be disclosed, distributed or reproduced without the prior written permission of EPAM Systems.

## Lab work. Day 1

### Tasks

#### 1. Install Docker Service on CentOS 7

##### 1.1 Install Docker.

##### 1.2 Verify that Docker has been installed and running correctly. (1)

##### 1.3 Check that service runs on Host start. (1)

```
File Edit Selection Find View Goto Tools Project Preferences Help
Vagrantfile docker.sh
1 yum install -y \
2 yum-utils \
3 jq \
4 net-tools
5 yum-config-manager \
6 --add-repo \
7 https://download.docker.com/linux/centos/docker-ce.repo
8 yum -y install docker-ce
9 systemctl enable docker
10 systemctl start docker
11 usermod -ag docker vagrant
```

```
Vagrantfile docker.sh
1 Vagrant.configure("2") do |config|
2
3   ip=["192.168.68.69"]
4
5   servers=[
6     {
7       :hostname => "BLEDAI",
8       :ip => ip[0],
9       :box => "sbeliakou/centos",
10      :ram => 1024,
11      :cpu => 2,
12      :script => "docker.sh"
13    }
14  ]
15  servers.each do |machine|
16    config.vm.define machine[:hostname] do |node|
17      node.vm.box = machine[:box]
18      node.vm.hostname = machine[:hostname]
19      node.vm.network "private network", ip: machine[:ip]
20      node.vm.provider "virtualbox" do |vb|
21        vb.customize ["modifyvm", :id, "--memory", machine[:ram]]
22      end
23      node.vm.provision "shell", path: machine[:script]
24    end
25  end
26 end
27
28 end
```

```
[vagrant@BLEDAI ~]$ ps -ef |grep docker
root      6307      1   0 07:38 ?           00:00:00 /usr/bin/dockerd -H fd:// --con
ainerd=/run/containerd/containerd.sock
vagrant   6540   6506   0 07:40 pts/0    00:00:00 grep --color=auto docker
[vagrant@BLEDAI ~]$
```

```
[vagrant@BLEDAI ~]$ docker info
Containers: 0
  Running: 0
  Paused: 0
  Stopped: 0
Images: 0
Server Version: 18.09.7
Storage Driver: overlay2
  Backing Filesystem: xfs
  Supports d type: true
  Native Overlay Diff: true
Logging Driver: json-file
Cgroup Driver: cgroupfs
Plugins:
  Volume: local
  Network: bridge host macvlan null overlay
  Log: awslogs fluentd gcplogs gelf journald json-file local logentries splunk sy
slog
Swarm: inactive
Runtimes: runc
Default Runtime: runc
Init Binary: docker-init
containerd version: 894b81a4b802e4eb2a91d1ce216b8817763c29fb
runc version: 425e105d5a03fabd737a126ad93d62a9eede87f
init version: fec3683
Security Options:
  seccomp
    Profile: default
Kernel Version: 3.10.0-957.21.3.el7.x86_64
Operating System: CentOS Linux 7 (Core)
OSType: linux
Architecture: x86_64
CPUs: 2
Total Memory: 991MiB
Name: BLEDAI
```

- ## 2.1 Create Dockerfile and build custom image of nginx (on centos). (3)

## 2.2 Create Dockerfile and build custom image of nginx (on ubuntu). (3)

### 2.3 Create Dockerfile and build custom image of nginx (on alpine). (3)



```
FROM ubuntu
LABEL maintainer="Hanna Bledai"
RUN apt-get -y update && apt-get -y upgrade && \
    apt-get -y install openjdk-8-jdk wget && \
    cd /opt && \
    wget http://ftp.byfly.by/pub/apache.org/tomcat/tomcat-9/v9.0.20/bin/apache-tomcat-9.0.20.tar.gz && \
    tar xvfz /opt/apache-tomcat-9.0.20.tar.gz && \
    mv /opt/apache-tomcat-9.0.20 /opt/tomcat && \
    rm -rf /opt/*.tar.gz

EXPOSE 8080

CMD ["/opt/tomcat/bin/catalina.sh","run"]
```

## 2.6 Create Dockerfile and build custom image of tomcat (on alpine). (3)

```
FROM alpine
LABEL maintainer="Hanna Bledai"
RUN apk update && \
    apk add openjdk8 && \
    cd /opt && \
    wget http://ftp.byfly.by/pub/apache.org/tomcat/tomcat-9/v9.0.20/bin/apache-tomcat-9.0.20.tar.gz && \
    tar xvzf /opt/apache-tomcat-9.0.20.tar.gz && \
    mv /opt/apache-tomcat-9.0.20 /opt/tomcat && \
    rm -rf /opt/*.tar.gz

EXPOSE 8080

CMD ["/opt/tomcat/bin/catalina.sh","run"]
```

3. Run Containers in different modes. Prove settings to be applied correctly.
  - 3.1 Run Nginx and Tomcat with port mapping to Host node (automatically assigned port). (1)
  - 3.2 Run Nginx and Tomcat with port mapping to Host node (statically defined mapping). (1)

```
[vagrant@BLEDIAI tomcat2.6]$ docker run -P -d hbledai/tomcat:2.6
9632678173a2eaf2b10035ca886506ee220ff5e251649fc8af76218ec9af31e7
[vagrant@BLEDIAI tomcat2.6]$ vim Dockerfile
[vagrant@BLEDIAI tomcat2.6]$ docker run -P -d hbledai/tomcat:2.6
336d19439ae8e72c5584025c0c04864f03b8ebc0ace7075df09b63a26e13d368
[vagrant@BLEDIAI tomcat2.6]$ docker run -p 8080:8080 -d hbledai/tomcat:2.5
61f9a7da5cf80a1b75f2e556a19042fe87c97cb3997e0f1ca04ea8017d98c6cc
[vagrant@BLEDIAI tomcat2.6]$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED
61f9a7da5cf8	hbledai/tomcat:2.5	"/opt/tomcat/bin/cat..."	15 seconds ago
Up 14 seconds	0.0.0.0:8080->8080/tcp	awesome_edison	
336d19439ae8	hbledai/tomcat:2.6	"/opt/tomcat/bin/cat..."	28 seconds ago
Up 26 seconds	0.0.0.0:32880->8080/tcp	cranky_taussig	

```
[vagrant@BLEDIAI tomcat2.6]$
```

### 3.3 Run Nginx and Tomcat with Specified names. (1)

```
[vagrant@BLEDIAI tomcat2.6]$ docker run -P -d --name tomcat hbledai/tomcat:2.3
eb8c276de368edc87bfe2236cf442b01da7f2230ea09bclab2e19fe953cafbal
[vagrant@BLEDIAI tomcat2.6]$ docker run -P -d --name nginx hbledai/nginx:2.2
e08e962e2440cb7de07354c18be7c9f2cce60a5461b57326f90ceade57ede97f
[vagrant@BLEDIAI tomcat2.6]$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
e08e962e2440	hbledai/nginx:2.2	"nginx -g 'daemon of..."	8 seconds ago	Up 6 seconds	0.0.0.0:32882->80/tcp	nginx
eb8c276de368	hbledai/tomcat:2.3	"/opt/tomcat/bin/cat..."	26 seconds ago	Up 25 seconds	0.0.0.0:32881->8080/tcp	tomcat
61f9a7da5cf8	hbledai/tomcat:2.5	"/opt/tomcat/bin/cat..."	6 minutes ago	Up 6 minutes	0.0.0.0:8080->8080/tcp	awesome_edison
336d19439ae8	hbledai/tomcat:2.6	"/opt/tomcat/bin/cat..."	7 minutes ago	Up 7 minutes	0.0.0.0:32880->8080/tcp	cranky_taussig

```
[vagrant@BLEDIAI tomcat2.6]$
```

### 3.4 Run Nginx and Tomcat with different restart policies: always/unless-stopped (3)

```
[vagrant@BLEDIAI tomcat2.6]$ docker run -P -d --restart=always hbledai/nginx:2.2
6fbc30084204f98a8fb49186433d87193e6c392417b5c20ad5634be2986c
[vagrant@BLEDIAI tomcat2.6]$ docker run -P -d --restart=unless-stopped hbledai/tomcat:2.5
149a5036edc181fa55a7a6bd5f268eb90d10f767998ce4baea57b45be07e863
[vagrant@BLEDIAI tomcat2.6]$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
149a5036edc1	hbledai/tomcat:2.5	"/opt/tomcat/bin/cat..."	13 seconds ago	Up 12 seconds	0.0.0.0:32884->8080/tcp	condescending_bell
6fbc30084204	hbledai/nginx:2.2	"nginx -g 'daemon of..."	37 seconds ago	Up 36 seconds	0.0.0.0:32883->80/tcp	cranky_booth

## 4. Operate with running containers and available images.

### 4.1 Get List of Running Containers. (1)

```
[vagrant@BLEDIAI tomcat2.6]$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
ecd29a3ef8f8	hbledai/nginx:2.2	"nginx -g 'daemon of..."	7 seconds ago	Up 6 seconds	0.0.0.0:32888->80/tcp	inspiring_rubin
7fb82de12e96	hbledai/nginx:2.1	"nginx -g 'daemon of..."	10 seconds ago	Up 8 seconds	0.0.0.0:32887->80/tcp	heuristic_noether
3f9bd6c64c1b	hbledai/tomcat:2.3	"/opt/tomcat/bin/cat..."	21 seconds ago	Up 20 seconds	0.0.0.0:32886->8080/tcp	nostalgic_lehmann
8370e7a6ceb7	hbledai/tomcat:2.6	"/opt/tomcat/bin/cat..."	27 seconds ago	Up 26 seconds	0.0.0.0:32885->8080/tcp	focused_joliot
149a5036edc1	hbledai/tomcat:2.5	"/opt/tomcat/bin/cat..."	3 minutes ago	Up 3 minutes	0.0.0.0:32884->8080/tcp	condescending_bell
6fbc30084204	hbledai/nginx:2.2	"nginx -g 'daemon of..."	3 minutes ago	Up 3 minutes	0.0.0.0:32883->80/tcp	cranky_booth

```
[vagrant@BLEDIAI tomcat2.6]$
```

### 4.2 Get List of Running Containers, apply custom filter and format output (Name, Image, Ports, Status). (2)

```
[vagrant@BLEDIAI tomcat2.6]$ docker ps --format "table {{.Names}}\t{{.Image}}\t{{.Ports}}\t{{.Status}}"
```

NAMES	IMAGE	PORTS	STATUS
inspiring_rubin	hbledai/nginx:2.2	0.0.0.0:32888->80/tcp	Up About a minute
heuristic_noether	hbledai/nginx:2.1	0.0.0.0:32887->80/tcp	Up About a minute
nostalgic_lehmann	hbledai/tomcat:2.3	0.0.0.0:32886->8080/tcp	Up About a minute
focused_joliot	hbledai/tomcat:2.6	0.0.0.0:32885->8080/tcp	Up 2 minutes
condescending_bell	hbledai/tomcat:2.5	0.0.0.0:32884->8080/tcp	Up 5 minutes
cranky_booth	hbledai/nginx:2.2	0.0.0.0:32883->80/tcp	Up 5 minutes

```
[vagrant@BLEDIAI tomcat2.6]$
```

### 4.3 Stop several containers (docker stop/docker kill). Get details about them. (1)

```
[vagrant@BLEDIAI tomcat2.3]$ docker stop $(docker ps -a -q)
0d9fae014cfe
036aa31bc3ef
dcb8db0f52b4
f57366ac4341
[vagrant@BLEDIAI tomcat2.3]$ docker inspect $(docker ps -a -q) | jq '.[].State'.Status
"exited"
"exited"
"exited"
"exited"
[vagrant@BLEDIAI tomcat2.3]$
```

### 4.4 Get List of stopped Containers. (1)

```
0c189a376308
[vagrant@BLEDIAI tomcat2.3]$ docker ps -f "status=exited"
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
95cab4d9295a	hbledai/nginx:2.3	"nginx -g 'daemon of..."	4 minutes ago	Exited (1)	4 minutes ago	vigilant_r
ide						
ecd20a3eff8f	hbledai/nginx:2.2	"nginx -g 'daemon of..."	5 minutes ago	Exited (0)	About a minute ago	inspiring_
rubin						
7fb82de12e96	hbledai/nginx:2.1	"nginx -g 'daemon of..."	5 minutes ago	Exited (0)	About a minute ago	heuristic_
noether						
3f9bd6c64c1b	hbledai/tomcat:2.3	"/opt/tomcat/bin/cat..."	5 minutes ago	Exited (143)	About a minute ago	nostalgic_
lehmann						
8370e7a6ceb7	hbledai/tomcat:2.6	"/opt/tomcat/bin/cat..."	5 minutes ago	Exited (143)	About a minute ago	focused_jo
liot						
149a5036edc1	hbledai/tomcat:2.5	"/opt/tomcat/bin/cat..."	8 minutes ago	Exited (143)	About a minute ago	condescend
ing_bell						
6fbc30084204	hbledai/nginx:2.2	"nginx -g 'daemon of..."	8 minutes ago	Exited (0)	About a minute ago	cranky_boo
th						
e08e962e2440	hbledai/nginx:2.2	"nginx -g 'daemon of..."	11 minutes ago	Exited (0)	10 minutes ago	nginx
eb8c276de368	hbledai/tomcat:2.3	"/opt/tomcat/bin/cat..."	11 minutes ago	Exited (143)	10 minutes ago	tomcat
61f9a7da5cf8	hbledai/tomcat:2.5	"/opt/tomcat/bin/cat..."	18 minutes ago	Exited (143)	10 minutes ago	awesome_ed
ison						

## 4.5 Remove all stopped containers. (2)

```
[vagrant@BLEDIAI tomcat2.3]$ docker rm $(docker ps -a -q)
596dce954eb2
adc7df7e81ed
1df822a2036c
bc30bc6f249c
268d075f46ec
fede8471e60b
79b8cdcd3482
bb83b85fbb2b
344beed967b9
[vagrant@BLEDIAI tomcat2.3]$
```

## 4.6 Remove all unused images. (2)

```
[vagrant@BLEDIAI tomcat2.3]$ docker image prune
WARNING! This will remove all dangling images.
Are you sure you want to continue? [y/N] y
Deleted Images:
deleted: sha256:06d5e1538e9fa203f76d369308dd3ad8b83c2d7419219c46649111c48357fe66
deleted: sha256:2635cd5c3393c77e34373eeb931f601e2a1d8a384a8658763ab20b706ecb6941
deleted: sha256:9c53bcac910057ecc3371ad32cc10cef6e98161f1be361e8fa9aca52c36f1c001
deleted: sha256:5e59f5eb9ad8e45f5e87b99e82afdc6c68f0bab9bc9d079868085fc7a644b77
deleted: sha256:7c7ea7f238b458dd21dd171eb12c65d93dff61b1de8a4d4d50f923b42dda9020
deleted: sha256:8a1c39164369ac0d3a64707bfc2a73f7c2129c0dc08613337a1831a24014ff16
deleted: sha256:6cb38549f9e170fb2ef537de5d20356e53cb243ef19c816ca274c3ee7e45a1
deleted: sha256:0f7bf0d5a5b5366a9b5461da9b88be6f76eed175e644b80cbac7ac8fbfb4f8e
deleted: sha256:2a03ff31e6268b7265cf87b74b7b99d1bae5f0a99515fdc48f8545c3e41a92
deleted: sha256:6b0fcl1bef123fe7f5e95ca8983714ad92473c0c8388904de5cad1eb26717fe4e
deleted: sha256:9efa79bc151130852025a7dfdec4978099e8d72307b1ac8485b9fa3bbef9771
deleted: sha256:4de912107764af0c30aaba0fcc592f7817d7e7a0e2a029046dcf332f17194737
deleted: sha256:7385b466c2660a6b1ba92ac0bf8eb23f1e06dcdcb97a84ff850d37cc161e617
deleted: sha256:c4ab7d8f7286005aaf075610bb179b091474eb5a3800a81b4b87940d28248a17
deleted: sha256:4648419d732202b7e29f32f7229208297bb1f252b113538c6275b2c687b29f
deleted: sha256:929040e1bce951eeb5223976a2f14cc3d0793c875ddc32d228156b22e36a1a83
deleted: sha256:2b62edb48a0cb581415718deb29d3624a36db340289e843fec1aad8b64fd772
deleted: sha256:ea6eba253bb8d305d0357c54265fa1f4949aa19b1067a9baaddecc2dfb5d56c3
```

## 5. Get logs from all containers.

### 5.1 Get logs from Containers with "docker log" command (1)

```
[vagrant@BLEDIAI tomcat2.3]$ docker run -P -d --name tomcat hbledai/tomcat:2.4
0d9fae014cfec5b4725408fc2f4a2bde61c286c53ba20c4530013261def43f25
[vagrant@BLEDIAI tomcat2.3]$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
0d9fae014cfe	hbledai/tomcat:2.4	"/opt/tomcat/bin/cat..."	4 seconds ago	Up 3 seconds	0.0.0.0:32895->8080/tcp	tomcat
036aa31bc3ef	hbledai/nginx:2.2	"nginx -g 'daemon of..."	34 minutes ago	Up 34 minutes	0.0.0.0:32892->80/tcp	eloquent_cori
dcb8db0f52b4	hbledai/nginx:2.1	"nginx -g 'daemon of..."	34 minutes ago	Up 34 minutes	0.0.0.0:32891->80/tcp	amazing_mendel
f57366ac4341	hbledai/tomcat:2.6	"/opt/tomcat/bin/cat..."	34 minutes ago	Up 34 minutes	0.0.0.0:32890->8080/tcp	pedantic_wu

```
[vagrant@BLEDIAI tomcat2.3]$
[vagrant@BLEDIAI tomcat2.3]$ docker logs tomcat
03-Jul-2019 12:57:37.577 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Server version name: Apache Tomcat/9.0.20
03-Jul-2019 12:57:37.580 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Server built: May 3 2019 22:26:00 UTC
03-Jul-2019 12:57:37.580 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Server version number: 9.0.20.0
03-Jul-2019 12:57:37.580 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log OS Name: Linux
03-Jul-2019 12:57:37.580 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log OS Version: 3.10.0-957.21.3.el7.x86_64
03-Jul-2019 12:57:37.580 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Architecture: amd64
03-Jul-2019 12:57:37.580 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Java Home: /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.212.b04-0.el7.x86_64/jre
03-Jul-2019 12:57:37.581 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log JVM Version: 1.8.0_212-b04
03-Jul-2019 12:57:37.585 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log JVM Vendor: Oracle Corporation
03-Jul-2019 12:57:37.585 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log CATALINA_BASE: /opt/tomcat
03-Jul-2019 12:57:37.585 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log CATALINA_HOME: /opt/tomcat
```

### 5.2 Run Tomcat so that its logs (catalina.out) will be available on the host system by the path of /var/log/tomcat/catalina.out. Check logs. (1)

I had to make some changes to the file of catalina.sh so that the changes were written to the log and the container did not fall. Added lines "touch \$CATALINA\_OUT" and \$CATALINA\_OUT 2>&1

```

elif [ "$1" = "run" ]; then
    shift
    touch "$CATALINA_OUT"
    if [ "$1" = "-security" ] ; then
        if [ $have tty -eq 1 ]; then
            echo "Using Security Manager"
        fi
        shift
        eval exec "\"$ RUNJAVA\" \"\$LOGGING_CONFIG\" \$LOGGING_MANAGER $JAVA_OPTS $CATALINA_OPTS \
        -D$ENDORSED_PROP=\"\$JAVA_ENDORSED_DIRS\" \" \
        -classpath \"\$CLASSPATH\" \" \
        -Djava.security.manager \
        -Djava.security.policy=\"\$CATALINA_BASE/conf/catalina.policy\" \" \
        -Dcatalina.base=\"\$CATALINA_BASE\" \" \
        -Dcatalina.home=\"\$CATALINA_HOME\" \" \
        -Djava.io.tmpdir=\"\$CATALINA_TMPDIR\" \" \
        org.apache.catalina.startup.Bootstrap \"$@\" start \
        >> \"$CATALINA_OUT\" 2>&1
    else
        eval exec "\"$ RUNJAVA\" \"\$LOGGING_CONFIG\" \$LOGGING_MANAGER $JAVA_OPTS $CATALINA_OPTS \
        -D$ENDORSED_PROP=\"\$JAVA_ENDORSED_DIRS\" \" \
        -classpath \"\$CLASSPATH\" \" \
        -Dcatalina.base=\"\$CATALINA_BASE\" \" \
        -Dcatalina.home=\"\$CATALINA_HOME\" \" \
        -Djava.io.tmpdir=\"\$CATALINA_TMPDIR\" \" \
        org.apache.catalina.startup.Bootstrap \"$@\" start \
        >> \"$CATALINA_OUT\" 2>&1
    fi
fi

```

```

student@EPBYMINW5249:~/BLEDAI/... x vagrant@BLEDAI:~ x
FROM centos:7
LABEL maintainer="Hanna Bledai"
RUN yum install -y wget java-1.8.0-openjdk-devel && \
    cd /opt && \
    wget http://ftp.byfly.by/pub/apache.org/tomcat/tomcat-9/v9.0.20/bin/apache-tomcat-9.0.20.tar.gz && \
    tar xvzf /opt/apache-tomcat-9.0.20.tar.gz && \
    mv /opt/apache-tomcat-9.0.20 /opt/tomcat && \
    rm -rf /opt/*.tar.gz
COPY catalina.sh /opt/tomcat/bin/catalina.sh
EXPOSE 8080

CMD ["/opt/tomcat/bin/catalina.sh", "run"]

"tomcat2.3/Dockerfile" 12L, 449C 10,1 All

```

```

[vagrant@BLEDAI tomcat2.3]$ docker run -P -v /var/log/tomcat/catalina.out:/opt/tomcat/logs/catalina.out hbledai/tomcat:2.3
^C[vagrant@BLEDAI tomcat2.3]$ dockerun -P -d -v /var/log/tomcat/catalina.out:/opt/tomcat/logs/catalina.out hbledai/tomcat:2.3
d22276d3f6ae8e9db7d33a92d87224b32fa137e116935ee59dde698f776fb20
[vagrant@BLEDAI tomcat2.3]$ cat /var/log/tomcat/catalina.out
03-Jul-2019 14:51:31.832 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Server version name: Apache Tomcat/9.0.20
03-Jul-2019 14:51:31.834 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Server built: May 3 2019 22:26:00 UTC
03-Jul-2019 14:51:31.834 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Server version number: 9.0.20.0
03-Jul-2019 14:51:31.834 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log OS Name: Linux
03-Jul-2019 14:51:31.834 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log OS Version: 3.10.0-957.21.3.el7.x86_64
03-Jul-2019 14:51:31.834 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Architecture: amd64
03-Jul-2019 14:51:31.834 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Java Home: /usr/lib64/java

```

### 5.3 Configure Containers to send logs to Journald. Get logs with journalctl. (1)



```
[vagrant@BLEDIAI tomcat2.3]$ docker run --log-driver=journald -P -d hbledai/nginx:2.1
4f858a5b12cf77d544e9291b208d4cb5bfa9bd00b7bd245cbf335818fc7aba98
[vagrant@BLEDIAI tomcat2.3]$ sudo journalctl -u docker.service
-- Logs begin at Wed 2019-07-03 07:32:25 UTC, end at Wed 2019-07-03 15:06:13 UTC. --
Jul 03 07:38:16 BLEDIAI systemd[1]: Starting Docker Application Container Engine...
Jul 03 07:38:17 BLEDIAI dockerd[6307]: time="2019-07-03T07:38:17.054050186Z" level=info msg="parsed scheme: \"unix\" module=grpc
Jul 03 07:38:17 BLEDIAI dockerd[6307]: time="2019-07-03T07:38:17.054096138Z" level=info msg="scheme \"unix\" not registered, fallback to default scheme
Jul 03 07:38:17 BLEDIAI dockerd[6307]: time="2019-07-03T07:38:17.054153601Z" level=info msg="parsed scheme: \"unix\" module=grpc
Jul 03 07:38:17 BLEDIAI dockerd[6307]: time="2019-07-03T07:38:17.054160274Z" level=info msg="scheme \"unix\" not registered, fallback to default scheme
Jul 03 07:38:17 BLEDIAI dockerd[6307]: time="2019-07-03T07:38:17.055360531Z" level=info msg="ccResolverWrapper: sending new addresses to cc: [{unix:///
Jul 03 07:38:17 BLEDIAI dockerd[6307]: time="2019-07-03T07:38:17.055401631Z" level=info msg="ClientConn switching balancer to \"pick_first\" module=gr
Jul 03 07:38:17 BLEDIAI dockerd[6307]: time="2019-07-03T07:38:17.055464870Z" level=info msg="pickfirstBalancer: HandleSubConnStateChange: 0xc42078c180,
Jul 03 07:38:17 BLEDIAI dockerd[6307]: time="2019-07-03T07:38:17.055582154Z" level=info msg="pickfirstBalancer: HandleSubConnStateChange: 0xc42078c180,
Jul 03 07:38:17 BLEDIAI dockerd[6307]: time="2019-07-03T07:38:17.055642129Z" level=info msg="ccResolverWrapper: sending new addresses to cc: [{unix:///
Jul 03 07:38:17 BLEDIAI dockerd[6307]: time="2019-07-03T07:38:17.055651055Z" level=info msg="ClientConn switching balancer to \"pick_first\" module=gr
Jul 03 07:38:17 BLEDIAI dockerd[6307]: time="2019-07-03T07:38:17.055668669Z" level=info msg="pickfirstBalancer: HandleSubConnStateChange: 0xc42078c510,
Jul 03 07:38:17 BLEDIAI dockerd[6307]: time="2019-07-03T07:38:17.055744958Z" level=info msg="pickfirstBalancer: HandleSubConnStateChange: 0xc42078c510,
Jul 03 07:38:17 BLEDIAI dockerd[6307]: time="2019-07-03T07:38:17.104582899Z" level=info msg="Graph migration to content-addressability took 0.00 second
Jul 03 07:38:17 BLEDIAI dockerd[6307]: time="2019-07-03T07:38:17.105156475Z" level=info msg="Loading containers: start."
Jul 03 07:38:17 BLEDIAI dockerd[6307]: time="2019-07-03T07:38:17.235175222Z" level=info msg="Default bridge (docker0) is assigned with an IP address 17
Jul 03 07:38:17 BLEDIAI dockerd[6307]: time="2019-07-03T07:38:17.285523047Z" level=info msg="Loading containers: done."
Jul 03 07:38:17 BLEDIAI dockerd[6307]: time="2019-07-03T07:38:17.316592322Z" level=info msg="Docker daemon" commit=2d0083d graphdriver(s)=overlay2 vers
```

## 5.4 Configure Docker daemon to send logs from all containers to journald. (1)

```
vagrant@BLEDIAI:~/tomcat2.6
"log-driver": "journald"

/etc/docker/daemon.json [readonly] 3L, 31C
```

## 5.5 Get logs container logs from journald by:

### Container name (1)

```
[vagrant@BLEDIAI tomcat2.3]$ docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED          STATUS          PORTS          NAMES
056be577192a      hbledai/tomcat:2.6 "/opt/tomcat/bin/cat..." 34 seconds ago   Up 33 seconds   0.0.0.0:32768->8080/tcp   happy_benz
[vagrant@BLEDIAI tomcat2.3]$ sudo journalctl CONTAINER_NAME=happy_benz
-- Logs begin at Wed 2019-07-03 07:32:25 UTC, end at Wed 2019-07-03 15:13:25 UTC. --
Jul 03 15:12:43 BLEDIAI 056be577192a[1748]: 03-Jul-2019 15:12:43.715 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Server version n
Jul 03 15:12:43 BLEDIAI 056be577192a[1748]: 03-Jul-2019 15:12:43.819 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Server built:
Jul 03 15:12:43 BLEDIAI 056be577192a[1748]: 03-Jul-2019 15:12:43.819 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Server version n
Jul 03 15:12:43 BLEDIAI 056be577192a[1748]: 03-Jul-2019 15:12:43.819 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log OS Name:
Jul 03 15:12:43 BLEDIAI 056be577192a[1748]: 03-Jul-2019 15:12:43.819 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log OS Version:
Jul 03 15:12:43 BLEDIAI 056be577192a[1748]: 03-Jul-2019 15:12:43.819 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Architecture:
Jul 03 15:12:43 BLEDIAI 056be577192a[1748]: 03-Jul-2019 15:12:43.819 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Java Home:
Jul 03 15:12:43 BLEDIAI 056be577192a[1748]: 03-Jul-2019 15:12:43.819 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log JVM Version:
Jul 03 15:12:43 BLEDIAI 056be577192a[1748]: 03-Jul-2019 15:12:43.819 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log JVM Vendor:
Jul 03 15:12:43 BLEDIAI 056be577192a[1748]: 03-Jul-2019 15:12:43.819 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log CATALINA_BASE:
Jul 03 15:12:43 BLEDIAI 056be577192a[1748]: 03-Jul-2019 15:12:43.820 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log CATALINA_HOME:
Jul 03 15:12:43 BLEDIAI 056be577192a[1748]: 03-Jul-2019 15:12:43.833 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Command line arg
Jul 03 15:12:43 BLEDIAI 056be577192a[1748]: 03-Jul-2019 15:12:43.833 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Command line arg
```

### Container tag (1)

```
[vagrant@BLEDIAI tomcat2.3]$ docker run -P -d --log-opt tag=tomcat hbledai/tomcat:2.6
cdccff1a5a9629dc193c97e66d1aa67113302a1429744e12c0b586837ce1d5b1
[vagrant@BLEDIAI tomcat2.3]$ sudo journalctl CONTAINER_TAG=tomcat
-- Logs begin at Wed 2019-07-03 07:32:25 UTC, end at Wed 2019-07-03 15:14:49 UTC. --
Jul 03 15:14:31 BLEDIAI tomcat[1748]: 03-Jul-2019 15:14:31.868 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Server version name:
Jul 03 15:14:31 BLEDIAI tomcat[1748]: 03-Jul-2019 15:14:31.875 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Server built:
Jul 03 15:14:31 BLEDIAI tomcat[1748]: 03-Jul-2019 15:14:31.885 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Server version number:
Jul 03 15:14:31 BLEDIAI tomcat[1748]: 03-Jul-2019 15:14:31.885 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log OS Name:
Jul 03 15:14:31 BLEDIAI tomcat[1748]: 03-Jul-2019 15:14:31.885 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log OS Version:
Jul 03 15:14:31 BLEDIAI tomcat[1748]: 03-Jul-2019 15:14:31.885 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Architecture:
Jul 03 15:14:31 BLEDIAI tomcat[1748]: 03-Jul-2019 15:14:31.885 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Java Home:
Jul 03 15:14:31 BLEDIAI tomcat[1748]: 03-Jul-2019 15:14:31.885 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log JVM Version:
Jul 03 15:14:31 BLEDIAI tomcat[1748]: 03-Jul-2019 15:14:31.885 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log JVM Vendor:
Jul 03 15:14:31 BLEDIAI tomcat[1748]: 03-Jul-2019 15:14:31.885 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log CATALINA_BASE:
Jul 03 15:14:31 BLEDIAI tomcat[1748]: 03-Jul-2019 15:14:31.885 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log CATALINA_HOME:
Jul 03 15:14:31 BLEDIAI tomcat[1748]: 03-Jul-2019 15:14:31.887 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Command line argument:
Jul 03 15:14:31 BLEDIAI tomcat[1748]: 03-Jul-2019 15:14:31.887 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Command line argument:
Jul 03 15:14:31 BLEDIAI tomcat[1748]: 03-Jul-2019 15:14:31.888 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Command line argument:
Jul 03 15:14:31 BLEDIAI tomcat[1748]: 03-Jul-2019 15:14:31.888 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Command line argument:
Jul 03 15:14:31 BLEDIAI tomcat[1748]: 03-Jul-2019 15:14:31.888 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Command line argument:
Jul 03 15:14:31 BLEDIAI tomcat[1748]: 03-Jul-2019 15:14:31.888 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Command line argument:
```

6. Containerize Python-FLASK application developed in previous LAB Module Python. You should create 2 images of python 2.x and 3.x versions and the same python/flask application. Run containers from both images in detached mode.

Response page should display following data (at least):

- Python Version it is running on (1)
- Flask Server version (1)
- Systems OS Name, Release Version, Family Name (3)
- Kernel Version (1)
- Container build date/time (2)
- Page generation date/time in zones UTC0, Europe/Minsk (2)
- System Host Name, IP address (2)
- Student (Developer) Name (1)
- Custom Environment Variable PARAMETER (1)
- Container Life time (2)

```
(venv) [student@EPBYMINW5249 Flask-App]$ docker run -p 5001:5000 hbledai/flaskapp:1
* Serving Flask app "app" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
```

0.0.0.0:5001

## TASK 6

Python version 3.7.3,  
Flask version 1.0.3,  
OS name posix Linux, Release Version: 3.10.0-957.12.2.el7.x86\_64,  
Family Name: GNU/Linux ,  
Kernel Version #1 SMP Tue May 14 21:24:32 UTC 2019,  
Date Create 2019-07-03 23:00:37  
, Page generation date/time in zones UTC0: 2019-07-03 20:02:36.492848+00:00,  
Page generation date/time in zones Europe/Minsk: 2019-07-03 23:02:36.492915+03:00 System Host Name, IP address e5de594e471f, 172.17.0.3  
Student name Hanna Bledai  
Container Life time: 2:58:01

41 point – 8

57 points – 10

```
^[[A^(venv) [student@EPBYMINW5249 Flask-App]$ docker build --build-arg created_date="$(date '+%Y-%m-%d %H:%M:%S')" --build-arg name_creator="Hanna Bledai" -t hb
flaskapp:1 .
```

```
FROM python:3
ARG name_creator
ARG created_date

LABEL maintainer=$name_creator

RUN pip install Flask
RUN pip install pytz

COPY . /

ENV DATE_CREATED=$created_date
ENV NAME_CREATOR=$name_creator

EXPOSE 5000

CMD ["python3", "app.py"]
```

```

import os
import sys
import subprocess
import re
import platform
import pytz
import socket
from datetime import datetime
from flask import Flask
app = Flask(__name__)
def get_install_package(package_name):
    p = {}
    instPack = subprocess.run('pip freeze', shell=True,
                              stdout=subprocess.PIPE).stdout.decode('utf-8')
    instPack = re.findall(r"[\w']+.*", instPack)
    for i in range(len(instPack)):
        instPack[i] = instPack[i].split('==')
    for iP in instPack:
        p.update({iP[0]: iP[1]})
    return p[package_name]
def get_family_name():
    return subprocess.run('uname -o', shell=True,
                          stdout=subprocess.PIPE).stdout.decode('utf-8')
def get_date_created():
    return os.getenv('DATE_CREATED')
def get_host_and_ip():
    host_name = socket.gethostname()
    host_ip = socket.gethostbyname(host_name)
    return (host_ip, host_name)
@app.route("/")
def content():
    dateNow = subprocess.run('date "+%Y-%m-%d %H:%M:%S"', shell=True,
                              stdout=subprocess.PIPE).stdout.decode('utf-8')
    time2 = parse_date_bash(dateNow)
    time1 = parse_date_bash(os.getenv('DATE_CREATED'))
    page = '<html><body><h1>'
    page += 'TASK 6'
    page += '</h1>'
    page += f'Python version {sys.version[:5]}, <br /> '
    page += f'\nFlask version {get_install_package("Flask")}, <br />'

```

```

page += f'\nOS name {os.name} {platform.system()}, Release Version:
{platform.release()},<br />' \
    f'Family Name: {get_family_name()},<br /> '
page += f'\n Kernel Version {platform.version()},<br /> '
page += f'\n Date Create {get_date_created()}<br />, '
page += f'Page generation date/time in zones UTC0: {datetime.now(pytz.utc)}, <br
/>' \
    f'Page generation date/time in zones Europe/Minsk:
{datetime.now(pytz.timezone("Europe/Minsk"))} '
page += f'System Host Name, IP address {socket.gethostname()},
{socket.gethostbyname(socket.gethostname())}<br />'
page += f'Student name {os.getenv("NAME_CREATOR")}<br />'
page += f'Container Life time: {time1 - time2}<br />'
page += '</body></html>'
return page

def parse_date_bash(date_null):
    date_arr = date_null.split()
    date_date = date_arr[0].split('-')
    date_time = date_arr[1].split(':')
    dateCteated = datetime(int(date_date[0]), int(date_date[1]), int(date_date[2]),
        int(date_time[0]), int(date_time[1]), int(date_time[2]))
    return dateCteated

if __name__ == "__main__":
    app.run(host='0.0.0.0', port=os.getenv('PORT')) # port 5000 is the default

```