

Tomcat nədir?

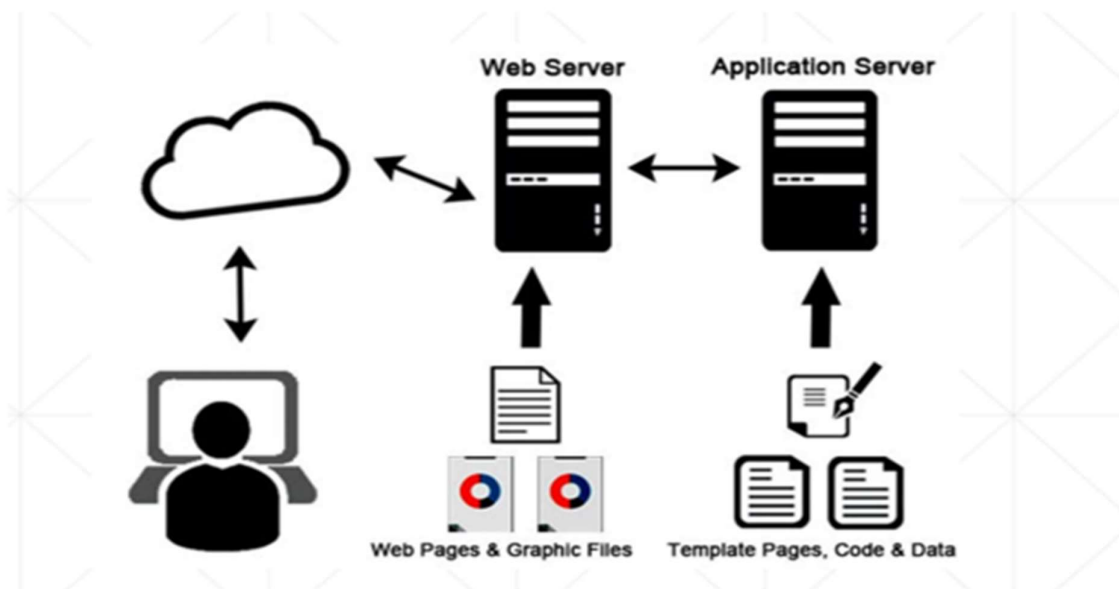
Application serverdir. Javayla yazılan kodları deploy etmək üçün istifadə olunur. Javayla yazılan kodu istifadəçilərə çatdırmaq üçün application server olan Tomcat-dan istifadə edirik. Tomcat javanı dəstəkləyən application serverdir. Application server-ə sorğu gəlir, özündə olan build(war,jar), web apps altına ataraq deploy edir. Azərbaycanda ən çox istifadə olunan programlaşdırma dili javadır. Buna görə tomcat-dan istifadə edirik.

Web serverlə Application server arasında fərq nədir? Web server həm static, həm də dinamik contexti bizə qaytaran fayldır. Application server isə özündən nəyisə əməl edir. "Səbətə at" kimi funksiyalar buna nümunə göstərmək olar. Tomcat-ı nə cür install edə bilərik?

1) yum install tomcat (servis automatic olaraq gəlir)

2) rəsmi saytdan tar.gz faylını götürərək install etmək (bu zaman service gəlmir, service ya özümüz qururuq ki, bu məsləhət olunmur, yada scriptdən istifadə edirik.)

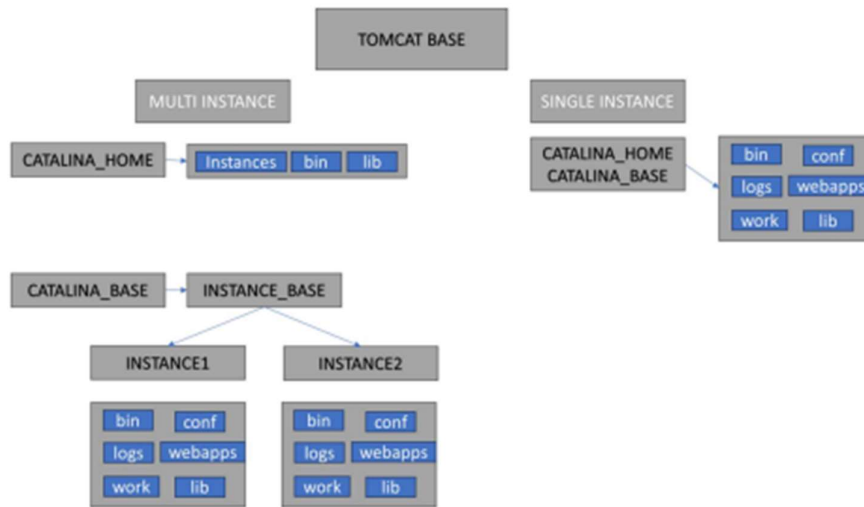
Ancaq Tomcat istifadə etmək üçün bizə ilk növbədə Javayı install etmək lazımdır. Daha sonra Tomcat install etmək olar. Heç bir halda Tomcat başqa programlaşdırma dili ilə istifadə olunmur. Yalnızca Java dili ilə istifadə olunur.



Application Server-dən söhbət gədirsə ilkin olaraq onun databesi olmalıdır. Ora sorğu gedəcək, userlər oradan istifadə edəcək. Best practise, Developer kod-da yazır, biz içəridən value set edib götürürük.

Tomcat-in fayl structure necədir?

CATALINA_HOME, CATALINE_BASE nədir? TOMCAT_BASE nədir? Tutaq ki /opt/tomcat qurmuşuq. Yeni mənim tomcatlarım burada olacaq. Bu bizim TOMCAT_BASE adlanır. Single_instance zamanı CAATALINE_HOME və CATALINA_BASE eyni olur. Single_instance dediyimiz tək fayl daxilində Tomcatlar toplanır. Ancaq bunun mənfi xüsusiyyətləridə var. Tutaq ki, 5 tomcat deploy olunur. Yeni developer-lər bu tomcat üzərində işləyir. Ancaq bu developer bu tomcatların birində dəyişiklik etməyimizi istəsə biz gərək bütün deploy olan tomcat-ları stop edək. Bunun qarşısını almaq üçün biz Multi_instance qura bilərik. Bunun sayəsində biz bir OS daxilində 5 fərqli tomcat deploy edə bilərik. Ancaq burada qeyd etməli olduğumuz bir neçə nüans var. Bunlardan biri odurki, Linuxda bir port-dan yalnız bir service istifadə edə bilər. Multi_Instance zamanı CATALINA_HOME-da bin və lib faylları saxlanılır və fərqli versiyalarda tomcat-ları deploy etmək olur.



Library daxilində biz çox işləmirik. Developer kodu yazıb hansı ki, biz onu deploy etməliyik, ona lazım olan library ya sənə deyir, ya da qeyd edirik ki, bunlar mütləq olmalıdı. Webapps hansı application işləməsinə istəyiriksə onu qeyd edirik. Conf daxilində Tomcat-in əsas configləri yerləşir. Onlardan ən önəmlilərdən biri isə server.xml-dir. Burada port, ssl configləri yerləşir. Bin daxilində necə start, stop setenv.sh yerləşir. Deploying .war file Hansısa developer java-da kod yazıb. Build commandasıyla jar faylı quraşdırılır. Javadaki kodu build etmək üçün iki tool var. Maven və Gradle. Oda yalnız developer tərəfindən müəyyən edilir.

Single_Instance necə qururuq?

İlkin olaraq Tomcat istifadə etmək üçün bizə java lazımdır. Javanı install etmək lazımdır. yum install openjdk edərsək bütün user-lər üçün javanı qurduğu üçün fərqli user-ə fərqli java qurmağın qarşısını alır. JDK və JR var. JDK ümumi javanı gətirir. JR install zamanı bütün paketləri və library gətirmir.

Tomcat 9 requires Java SE 8 or later. We will install OpenJDK, the open-source implementation of the Java Platform ,which is the default Java development and runtime in CentOS 9

<https://openjdk.java.net/>

```
mkdir /usr/share/java
```

```
cd /usr/share/java
```

Download Java to this directory

```
vim .bash_profile
```

```
    JAVA_HOME=/usr/share/java
```

```
    PATH=$PATH:$JAVA_HOME/bin
```

```
    export PATH
```

Running Tomcat as the root user is a security risk and not considered best practice

We'll create a new system user

```
useradd tomcat1
```

```
passwd tomcat1
```

Create Directory for all tomcat files and prepare user for tomcat (TOMCAT_BASE)

```
mkdir /Tomcat
```

```
cd /Tomcat
```

Download Tomcat

We will download the latest version of Tomcat 9.0.x from the Tomcat downloads page .

```
cd /Tomcat
```

```
wget https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.100/bin/apache-tomcat-9.0.100.tar.gz
```

```
tar -xvf apache-tomcat-9.0.100.tar.gz - rm -rf apache apache-tomcat-9.0.100.tar.gz
```

```
mv apache-tomcat-9.0.100 Single-Instance
```

```
cd apache-tomcat-9.0.100
```

```
wget https://tomcat.apache.org/tomcat-7.0-doc/appdev/sample/sample.war --no-check-certificate
```

```
chown tomcat1:tomcat1 -R /Tomcat/
```

```
bash ./bin/catalina.sh
```

<http://localhost:8080/sample>

Configure multiple instances of Tomcat server

```
cd /Tomcat
```

```
mkdir Multi-Instances
```

```
cd Multi-Instances (startup\shutdown.sh)
```

```
cp -r /Tomcat/Single-Instance/bin . - cp -r /Tomcat/Single-Instance/lib .
```

```
mkdir instances
```

```
cd instances
```

```
mkdir instance1
```

```
mkdir instance2
```

```
cp -r /Tomcat/Single-Instance ./instance1
```

```
cp -r /Tomcat/Single-Instance ./instance2
```

```
cd instance1/2 - Delete unused files
```

Edit /conf/server.xml file for all new instance

Startup\shutdown.sh

```
#!/bin/bash
```

```
instance_name=$1
```

```
TOMCAT_BASE=/Tomcat
```

```
CATALINA_HOME=$TOMCAT_BASE/Multi-Instances
```

```
INSTANCE_BASE=$TOMCAT_BASE/Multi-Instances/instances
```

```
CATALINA_BASE=$INSTANCE_BASE/$instance_name
```

```
export CATALINA_HOME CATALINA_BASE
```

```
$CATALINA_BASE/bin/startup\shutdown.sh
```

```
Bash /Tomcat/Multi-Instances/startup\shutdown.sh instance1\instance2
```

Redirect http to https

```
<security-constraint>  
  <web-resource-collection>  
    <web-resource-name>Protected Context</web-resource-name>  
    <url-pattern>/*</url-pattern>  
  </web-resource-collection>  
</security-constraint>  
  
<user-data-constraint>  
  <transport-guarantee>CONFIDENTIAL</transport-guarantee>  
</user-data-constraint>  
</security-constraint>
```

Add certificate java keystore:

```
keytool --list -keystore $JAVA_HOME/lib/security/cacerts
```

```
keytool -import -alias ca -file somecert.cer -keystore $JAVA_HOME/lib/security/cacerts -  
storepass changeit
```