

## Tomcat 9.x / Linux: Postavljanje aplikacija kao web usluga




Kad smo se već potrudili pa instalirali i zaštili naš Tomcat serverčić, podignut ćemo na njemu par aplikacija. Svojom defaultnom internom strukturom Tomcat podržava "portalski pristup", znači, početnoj stranici portala pristupamo domenskim imenom (FQDN, poput `www.google.com`), potom za pristup aplikacijama rabimo poveznice prikazane u pregledniku. Dobar primjer je na nižoj slici - s FQDN URL-om `http://primjeri.corp.hr:8080` pristupili smo Tomcatovoj početnoj stranici (portalu), potom, da bismo došli do njegovih oglednih aplikacija kliknemo na linku Examples. Naravno, isto bismo postigli uporabom punog URL-a: `http://primjeri.corp.hr:8080/examples ...` i tako to.

Home Documentation Configuration Examples Wiki Mailing Lists

### Apache Tomcat/9.0.12

If you're seeing this, you've successfully installed



**Recommended Reading:**

- [Security Considerations HOW-TO](#)
- [Manager Application HOW-TO](#)
- [Clustering/Session Replication HOW-TO](#)

**Developer Quick Start**

- [Tomcat Setup](#)
- [First Web Application](#)
- [Realms & AAA](#)
- [JDBC DataSources](#)
- [Examples](#)

Naš je zadatak ovaj:

a) ostaviti defaultni virtualni host, poznat kao "localhost", za administratore;

b) podići tri aplikacije za potrebe radnika firme, ali tako da im se pristupa kao tipičnim web uslugama, znači, upisom imena `app1.corp.hr`, `app2.corp.hr` i `app3.corp.hr` u preglednik.

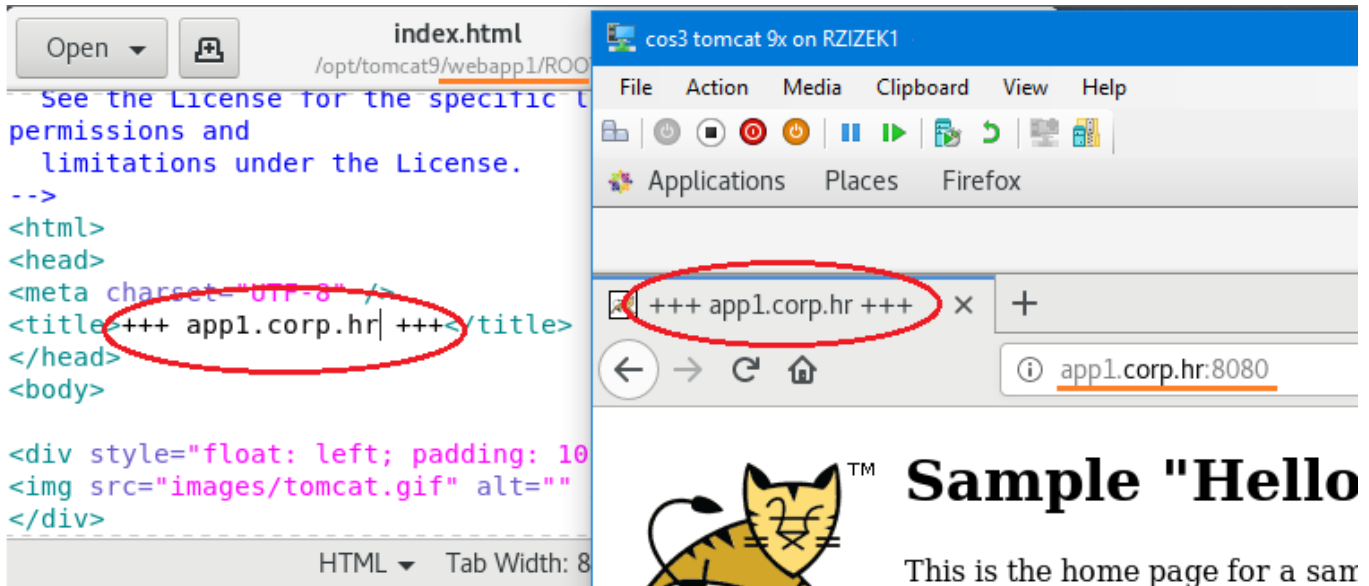
Stavka b) sugerira nam da je Tomcat na LAN-u, IP neka bude 192.168.10.11. U ovoj pokaznoj vježbi, budući da se koncentriramo na rad s virtualnim serverima - to je zato jer njima rješavamo zahtjev - iskoristit ćemo aplikaciju `sample.war` koju ćemo instalirati najjednostavnijom metodom `auto-deploy`.

1. Zaustaviti Tomcat: **systemctl stop tomcat9**.
2. U conf/server.xml koristimo definiciju localhosta kao predložak za dodatna tri virtualna servera... napravimo točno kako je na nižoj slici te, dakako, spremimo.

```
<Host name="localhost" appBase="webapps" unpackWARs="true" autoDeploy="true">
<!-- SingleSignOn valve, share authentication between web applications
Documentation at: /docs/config/valve.html -->
<!--
<Valve className="org.apache.catalina.authenticator.SingleSignOn" />
-->
<!-- Access log processes all example.
Documentation at: /docs/config/valve.html
Note: The pattern used is equivalent to using pattern="common" -->
<Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"
prefix="localhost_access_log" suffix=".txt"
pattern="%h %l %u %t &quot;%r&quot; %s %b" />
<Host name="app1.corp.hr" appBase="webapp1" unpackWARs="true" autoDeploy="true">
</Host>
<Host name="app2.corp.hr" appBase="webapp2" unpackWARs="true" autoDeploy="true">
</Host>
<Host name="app3.corp.hr" appBase="webapp3" unpackWARs="true" autoDeploy="true">
</Host>
</Engine>
</Service>
</Server>
```

3. U direktorij tomcat9 kreiramo direktorije webapp1, webapp2 i webapp3.
4. Iz direktorija /opt/tomcat9/webapps/docs/appdev/sample kopiramo paket sample.war u svaki novokreirani webappX.
5. Preimenujemo kopirane sample.war u ROOT.war (velika slova su obavezna).
6. Pobriniti se da vlasnik svih webappX direktorija i njihovih sadržaja bude tomsvc:  
**chown -R tomsvc:tomsvc webapp1 webapp2 webapp3**
7. Pokrenuti Tomcat: **systemctl start tomcat9 ->** Tomcat raspakirava .war paket i čini te aplikacije spremnima za pristup.
8. U Hosts korisničke stanice upisati tri retka:  
**192.168.10.11 app1.corp.hr**  
**192.168.10.11 app2.corp.hr**  
**192.168.10.11 app3.corp.hr**
9. Spojiti se s korisničkog računala na prvu aplikaciju: **http://app1.corp.hr:8080**, kliketati po pristupnim linkovima da vidimo radi li kako treba; isto ponoviti s preostale dvije aplikacije.
10. Kako bismo si olakšali zahtjevnija rekonfiguriranja i neizbježni tshooting (tko radi, taj griješi, jel'te), svakoj ćemo aplikaciji dati njen jedinstveni on-line identitet zahvatom u njen index.html, tako ćemo tijekom svakojakih isprobavanja i probnih spajanja uvijek znati na koju smo aplikaciju i virtualni

host spojeni. Nakon izmjene datoteke nije potreban restart Tomcata, on prati promjene i učitava ih.



**11.** Uvidom u /logs/catalina.out vidjeli smo da Tomcat uredno podiže virtualne hostove s našim aplikacijama ali nigdje nema logova o našim pristupanjima tim aplikacijama. Omogućit ćemo logiranje za svaki virtualni host (aplikaciju) zasebno: konfigurirajte server.xml kako je na nižoj slici, rabeći definiciju od localhosta kao osnovicu + restart Tomcata, potom pristupajte aplikacijama i pratite zapise u logovima tih aplikacija.

```

<Host name="localhost" appBase="webapps" unpackWARs="true" autoDeploy="true">
    <Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"
        prefix="localhost_access_log" suffix=".txt"
        pattern="%h %l %u %t &quot;%r&quot; %s %b" />
</Host>
<Host name="app1.corp.hr" appBase="webapp1" unpackWARs="true" autoDeploy="true">
    <Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"
        prefix="app1_access_log" suffix=".txt"
        pattern="%h %l %u %t &quot;%r&quot; %s %b" />
</Host>
<Host name="app2.corp.hr" appBase="webapp2" unpackWARs="true" autoDeploy="true">
    <Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"
        prefix="app2_access_log" suffix=".txt"
        pattern="%h %l %u %t &quot;%r&quot; %s %b" />
</Host>
<Host name="app3.corp.hr" appBase="webapp3" unpackWARs="true" autoDeploy="true">
    <Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"
        prefix="app3_access_log" suffix=".txt"
        pattern="%h %l %u %t &quot;%r&quot; %s %b" />
    </Host>
</Engine>
</Service>
</Server>
    
```

Lijepo smo to odradili pa su apetiti nadređenih porasli, sad žele da taj Tomcat server udomi i aplikaciju za HR, sa zaštitom podataka u transmisiji jer ipak se posredstvom te aplikacije radi s osobnim i poslovno senzitivnim podacima, i tako to. Možemo podići novu instancu Tomcata pa aplikaciju zavrtjeti u zasebnom procesu, možemo podići zasebnu instalaciju Tomcata pa će ta

aplikacija biti u vlastitoj JVM... ali mi ćemo ovaj zahtjev riješiti na najbrži način, dupliciranjem Tomcatove systemske komponente Service.

1. U direktoriju Tomcat9 kreiramo direktorij hrwebapp, ukopiramo u njega ROOT.war, predamo vlasništvo računom tomsvc.
2. Zaustavimo Tomcat.
3. Server.xml rekonfiguriramo prema nižoj slici.

Slika namjerno obuhvaća i definiciju app3.corp.hr. Prvi je razlog bolja orijentacija u kodu. Drugi je razlog usporedba sekcija za zapisivanje događaja, naime, pripremamo pristup aplikaciji kroz reverzni proxy koji će biti konfiguriran tako da u polju X-Forwarded-For HTTP *headera* prosljedi Tomcatu IP adresu klijenta, zato smo redefinirali logiranje kako bi Tomcat zapisivao te izvorne IP adrese.

```
<Host name="app3.corp.hr" appBase="webapp3" unpackWARs="true" autoDeploy="true">
  <Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"
    prefix="app3_access_log" suffix=".txt"
    pattern="%h %l %u %t &quot;%r&quot; %s %b" />
</Host>
</Engine>
</Service>
<Service name="hr-app">
<Connector port="8448" protocol="org.apache.coyote.http11.Http11NioProtocol"
  maxThreads="100" enableLookups="false" SSLEnabled="true" scheme="https"
  secure="true" clientAuth="false" sslProtocol="TLSv1.2" sslEnabledProtocols="TLSv1.2"
  keystoreType="PKCS12" keystoreFile="/opt/tomcat9/hr-kstore" keystorePass="Passw0rd0" />
<Engine name="hraplikacija" defaultHost="hrweb">
<Realm className="org.apache.catalina.realm.LockOutRealm">
<Realm className="org.apache.catalina.realm.UserDatabaseRealm" resourceName="UserDatabase">
</Realm>
</Realm>
<Host name="hrweb" appBase="hrwebapp" unpackWARs="true" autoDeploy="true">
<Alias>hrapp.corp.hr</Alias>
<Valve className="org.apache.catalina.valves.RemoteIpValve" />
<Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"
  prefix="hrapp_access_log" suffix=".txt"
  pattern="%{x-forwarded-for}i %h %l %u %t &quot;%r&quot; %s %b" />
</Host>
</Engine>
</Service>
</Server>
```

4. Sredimo si spremište digitalnih certifikata kako je opisano u prethodnom članku (vidi <https://sysportal.carnet.hr/node/1836> [1]), s time da nam u ovom scenariju treba poslužiteljski certifikat za ime hrapp.corp.hr.
5. Na vatrozidu servera otvorimo port TCP 8448 - novoj mašini (Engine) dodjeljujemo konektore sa zasebnim portovima jer je optimizacija konektora zasebna priča.
6. U Hosts korisničke stanice dodamo mrežno ime web usluge i IP: **192.168.10.11 hrapp.corp.hr**.
7. Startamo Tomcat, potom se URL-om **https://hrapp.corp.hr:8448** spojimo na HR aplikaciju.

Primjećujete da se u gornjim situacijama bavimo naprednijom razinom administriranja, ali bez ikakog teoretiziranja. Razlog je jednostavan: zainteresira li vas sve ovo, lako ćete se sami upoznati sa Tomcatovim komponentama poput Service, Engine, Connector, Context..., na službenom siteu Tomcat projekta.

\*

pon, 2019-06-17 09:08 - Ratko Žižek **Kuharice:** [Za sistemce](#) [2]  
**Kategorije:** [Software](#) [3]  
**Vote:** 0

No votes yet

**story\_tag:** [tomcat](#) [4]  
[Linux](#) [5]  
[aplikacija](#) [6]

**Source URL:** <https://sysportal.carnet.hr/node/1840>

#### Links

- [1] <https://sysportal.carnet.hr/node/1836>
- [2] <https://sysportal.carnet.hr/taxonomy/term/22>
- [3] <https://sysportal.carnet.hr/taxonomy/term/25>
- [4] <https://sysportal.carnet.hr/taxonomy/term/285>
- [5] <https://sysportal.carnet.hr/taxonomy/term/119>
- [6] <https://sysportal.carnet.hr/taxonomy/term/308>