

Server architecture and deployment

General part

Write 2-5 lines of text for each bullet:

- What is a virtual environment?
- Which benefits does a virtual environment give us?
- Explain why Java is running in a virtual environment and how it relates to Java's motto: write once, run everywhere.
- Explain what load balancing means.
- Give an example of one load balancing technique.

Practical part

For this exercise we will write a simple Java chat server and deploy it behind a Nginx proxy on a server.

You will need

- A server and your login credentials to log in to the server
- An installation of Nginx listening on port 80
- An installation of Tomcat listening on another port than 80 (preferably 8080)

1. Deploying a server

First write a small Java HTTP server with two methods:

- One HTTP POST method which accepts text from incoming requests (either by a URL parameter or from the request body) and inserts that text into a list on the server.
- One HTTP GET method which returns the content of that list in the body of the HTTP response to the client.

Now package the project to a .war file, ready to be deployed on your remote server.

When you have the .war file, copy it to the server and deploy it on Tomcat. Demonstrate that you can access the server by putting content into the list with the HTTP POST method, and getting it out with the HTTP GET method.

2. Setting up a reverse proxy

For this exercise we would like to forward incoming traffic from Nginx to the Tomcat server. Both the incoming traffic to the Nginx server and the communication from Nginx to Tomcat should be in plain HTTP. It should not be encrypted.

Implement a reverse proxy in Nginx on port 80 that forwards all traffic to the Tomcat server on another port (probably 8080). Demonstrate that you can still reach your HTTP server that you just deployed through Nginx on port 80.

Hints:

1. Use Netbeans automatised generation of Web Applications.
2. Setup the server and use Postman to see if you receive what you expect.
3. Use Netbeans to package your .war file and the 'scp' command to move it to the server.
4. When deployed, Nginx simply just needs to forward all http traffic to the Tomcat port. The command for this is 'proxy_pass' in the Nginx configuration file.