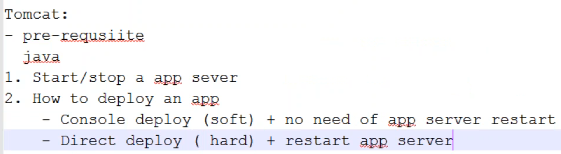
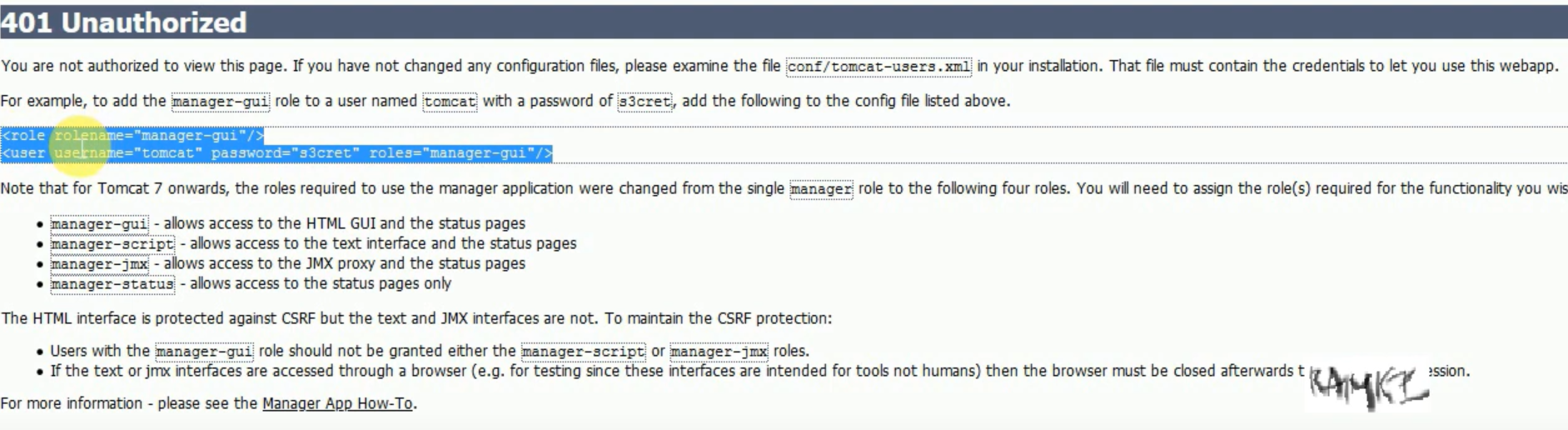
**Deployment:**



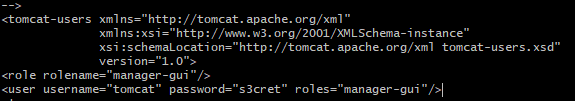
* Console deployment, we call it as soft deployment and direct deployment is called hard deployment

**Console deployment:**

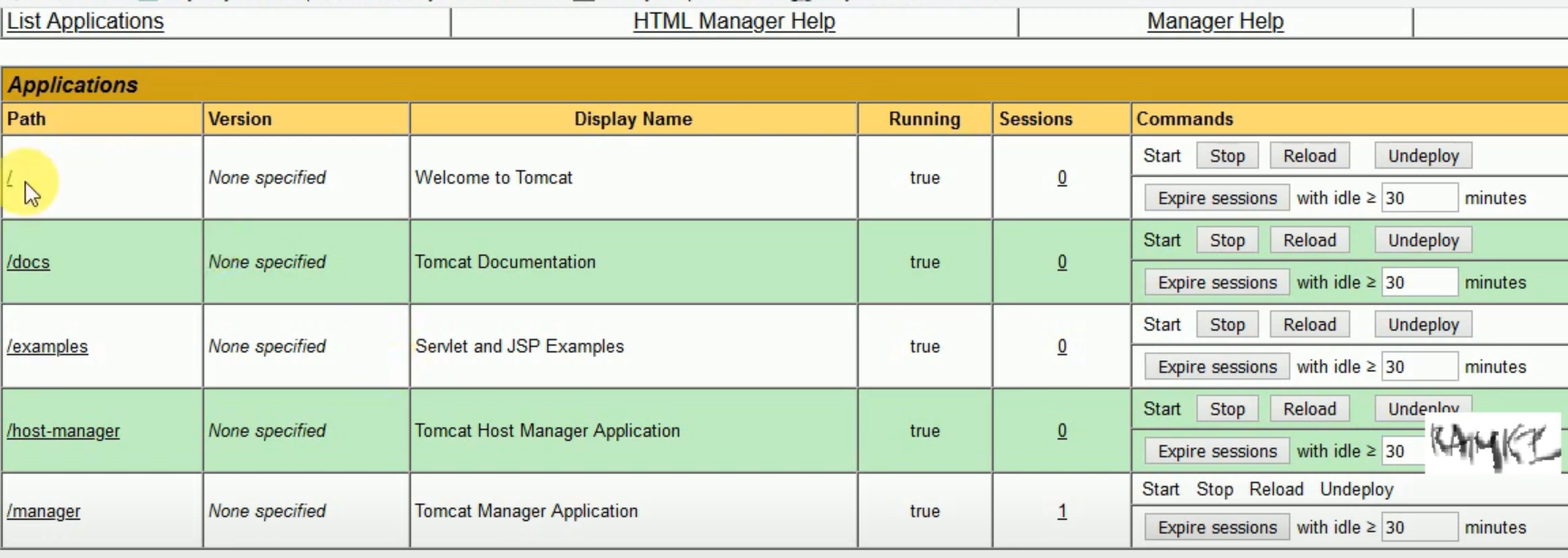
* Open tomcat portal using IP:PORT on the browser
* Then click on manager app. By default, we don’t have access for that, so just cancel the login prompt. It will show some error as below.



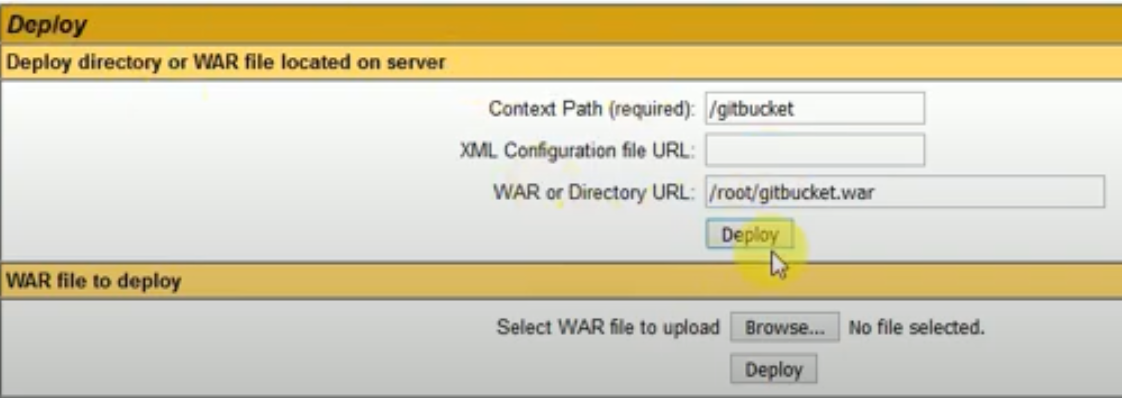
* so, we need to copy the lines mentioned in error and paste them in conf/tomcat-users.xml



* By this, we are creating a user and assigning roles
* These roles are specific to access something on the tomcat portal. We can try to access something and if we don’t have access, the error message will let us know what role is required to access the particular feature in portal.
* Then after we need to restart the tomcat
* Now, when we login to manager app with the credentials, we can be able to see the applications deployed in tomcat.



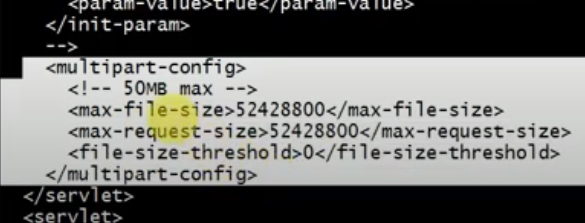
* We can see the docs, examples, manager applications because these folders come with tomcat installation by default in webapps directory. We can delete them as well. If we delete that, we can’t access the manager portal.
* We can also see the sessions in console as above. If we go inside, we can see the session ID and details as well.
* We can stop and start the applications from portal as well from console
* When we access an application from here. The address of the application will be like **IP:port/<application\_name> or <context>**
* We can also do the deployment from browser as below.
* We can either upload the war file from local machine which is also called client-side deployment, or we can give the path to the war file inside the server and give the context path. So that the application can pick it automatically. This is also called server-side deployment.



* Manager application is also called as console application. This we can find mainly in app servers
* Once we deploy it, we can able to see it in applications
* When we do console deployment, it creates another folder with the war file name in webapps directory.
* From the console, we can undeploy also. When we undeploy the directories on the server will be disappeared
* We can also reload the application from console.
* Manager app is available in console because it’s been deployed in webapps by default. We don’t need it for production environments, but it can be helpful for DEV environments.

**Increase the war file size for deployment:**

* If we go to webapps/manager/WEB\_INF/web.xml file, we can see below configuration.



* This means, it allows maximum 50MB size of war file for the deployment from console. We can modify it.

**Direct deployment:**

* Here, we need to copy the war file manually to the webapps folder. Then restart the tomcat
* After restarting it creates the folder with war file name inside webapps folder
* And, we can see the same under applications in tomcat console
* Whenever we restart tomcat, it will redeploy all the deployed applications