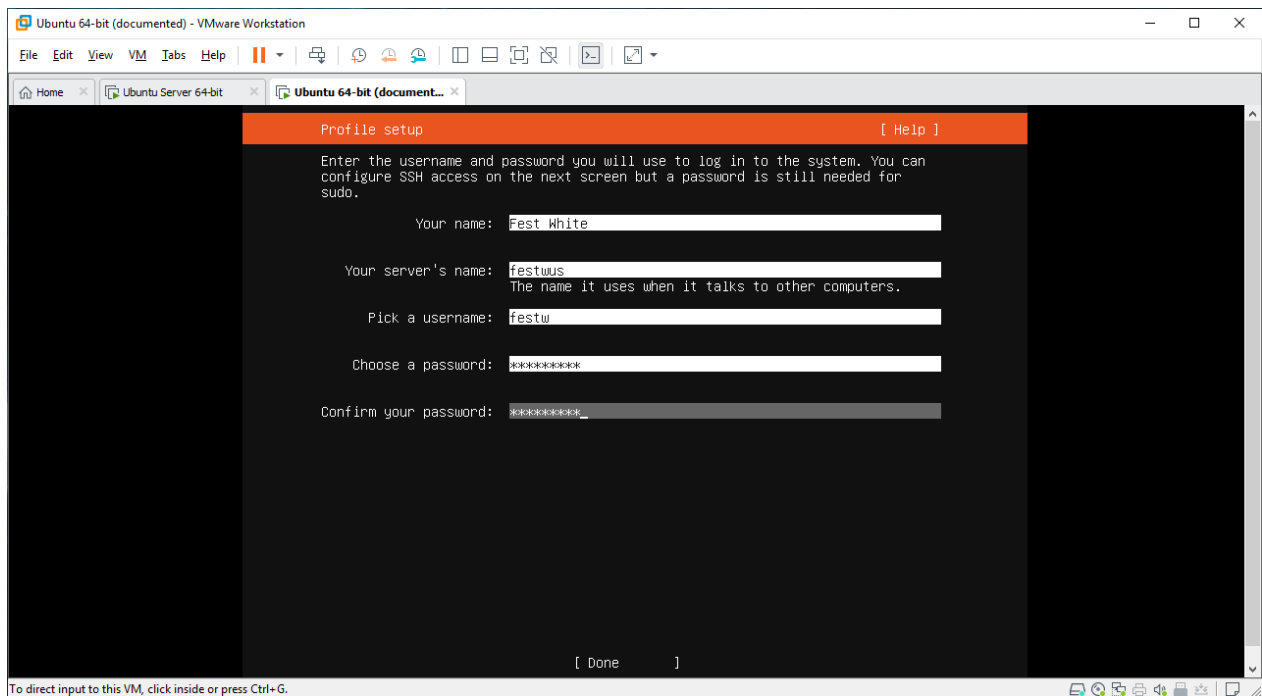
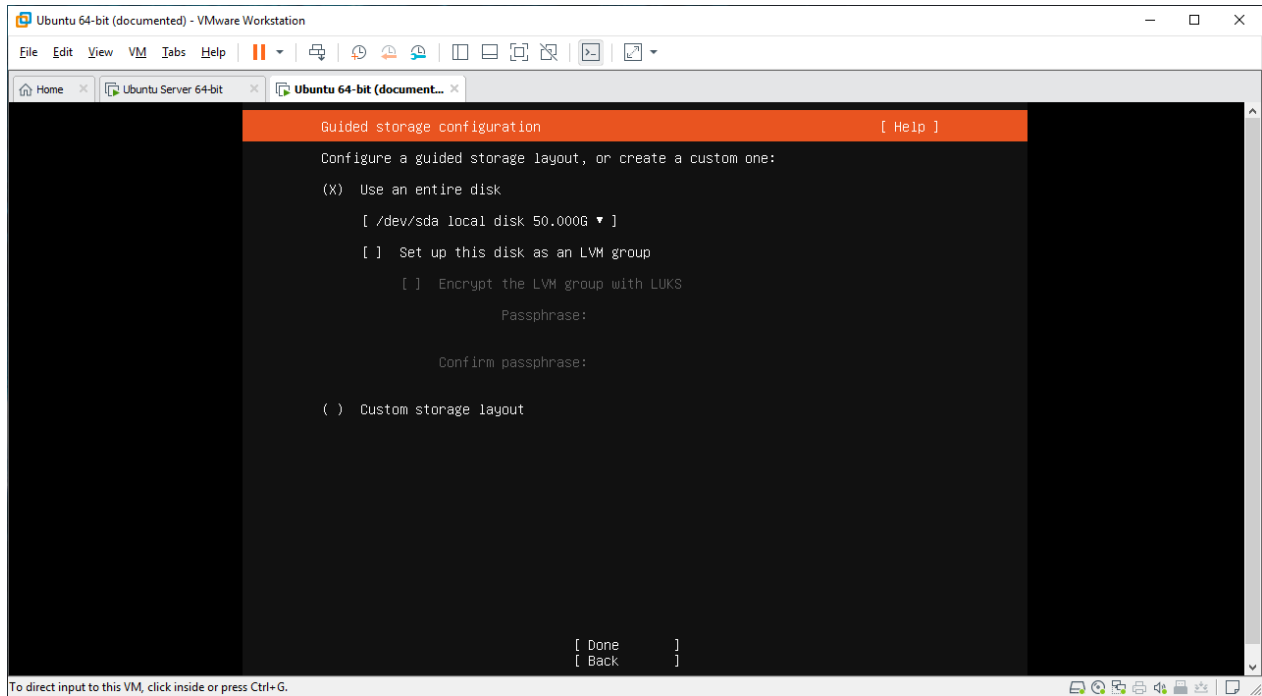
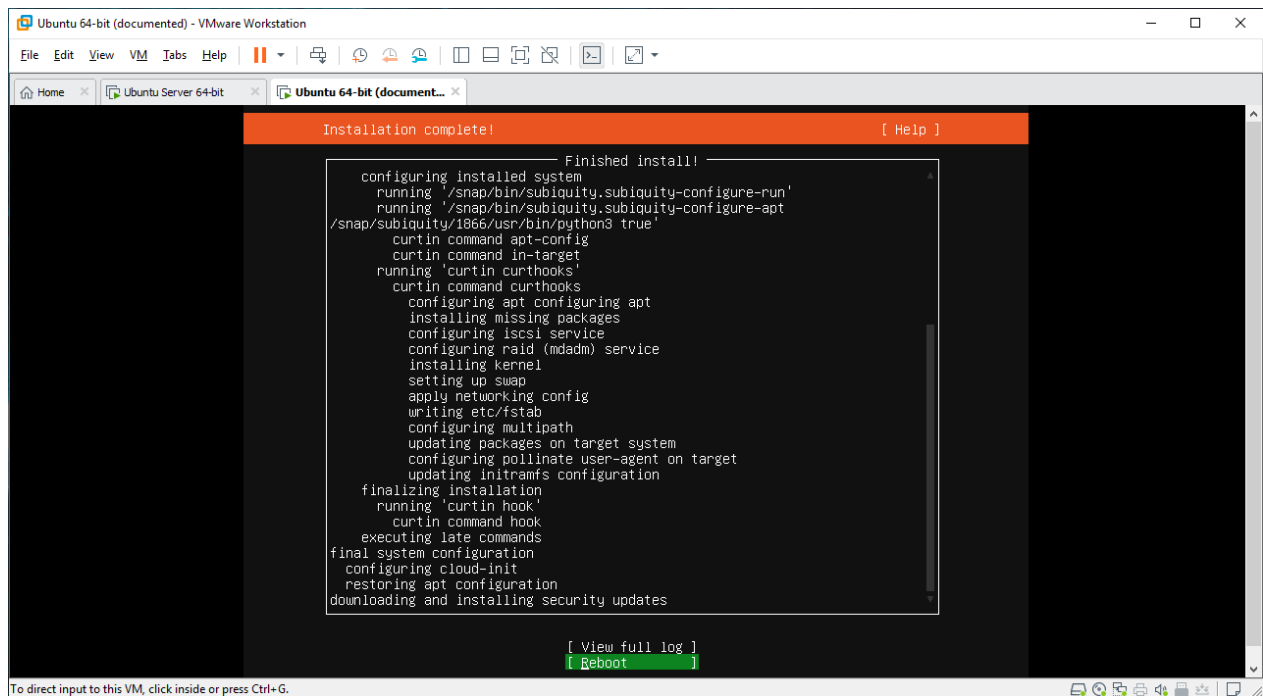
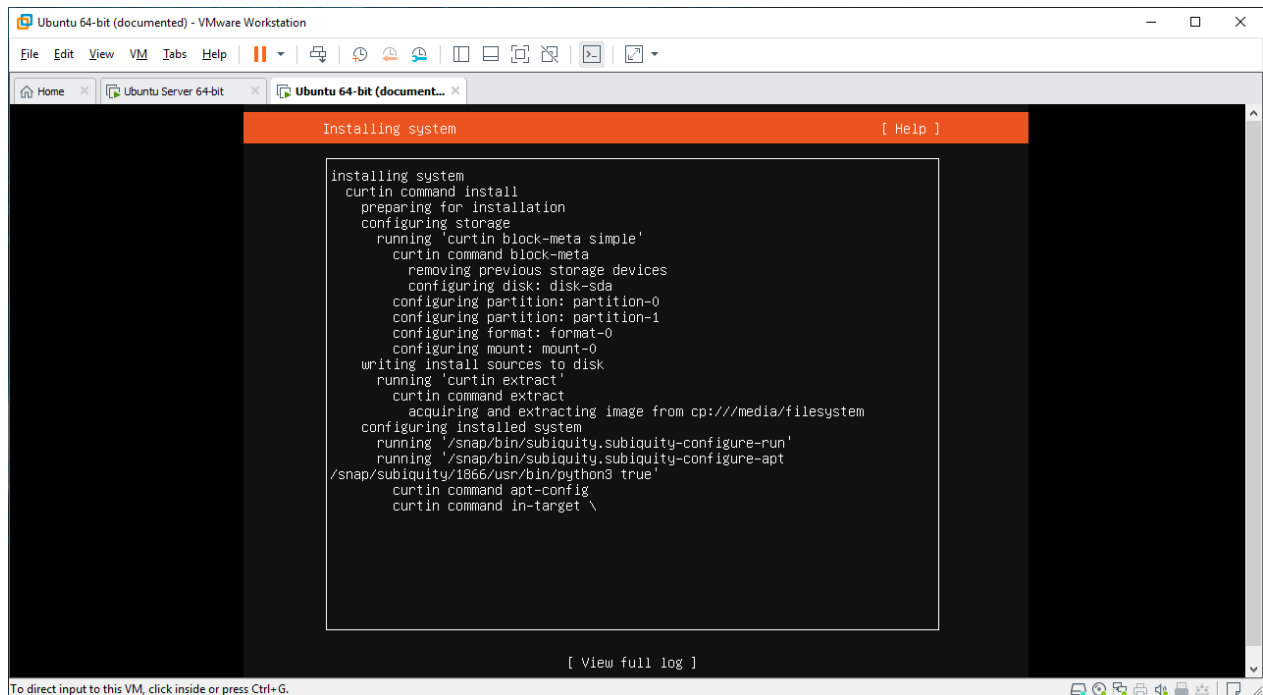


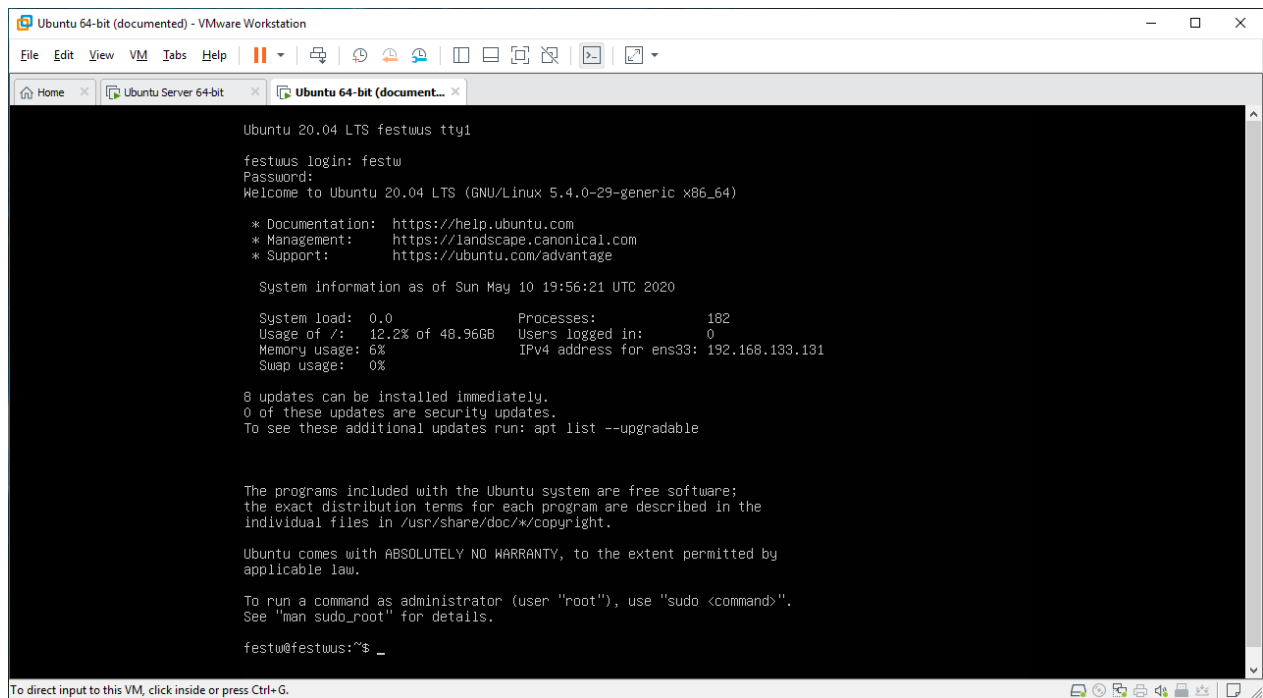
TASK 0

Ubuntu Server install;





Restart



```
Ubuntu 64-bit (documented) - VMware Workstation
File Edit View VM Tabs Help
Home x Ubuntu Server 64-bit x Ubuntu 64-bit (documented... x

Ubuntu 20.04 LTS festwus tty1
festwus login: festw
Password:
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-29-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Sun May 10 19:56:21 UTC 2020

System load:  0.0               Processes:    182
Usage of /:   12.2% of 48.96GB   Users logged in: 0
Memory usage: 6%               IPv4 address for ens33: 192.168.133.131
Swap usage:  0%

8 updates can be installed immediately.
0 of these updates are security updates.
To see these additional updates run: apt list --upgradable

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

festw@festwus:~$ _
```

To direct input to this VM, click inside or press Ctrl+G.

SNAPSHOT1 .

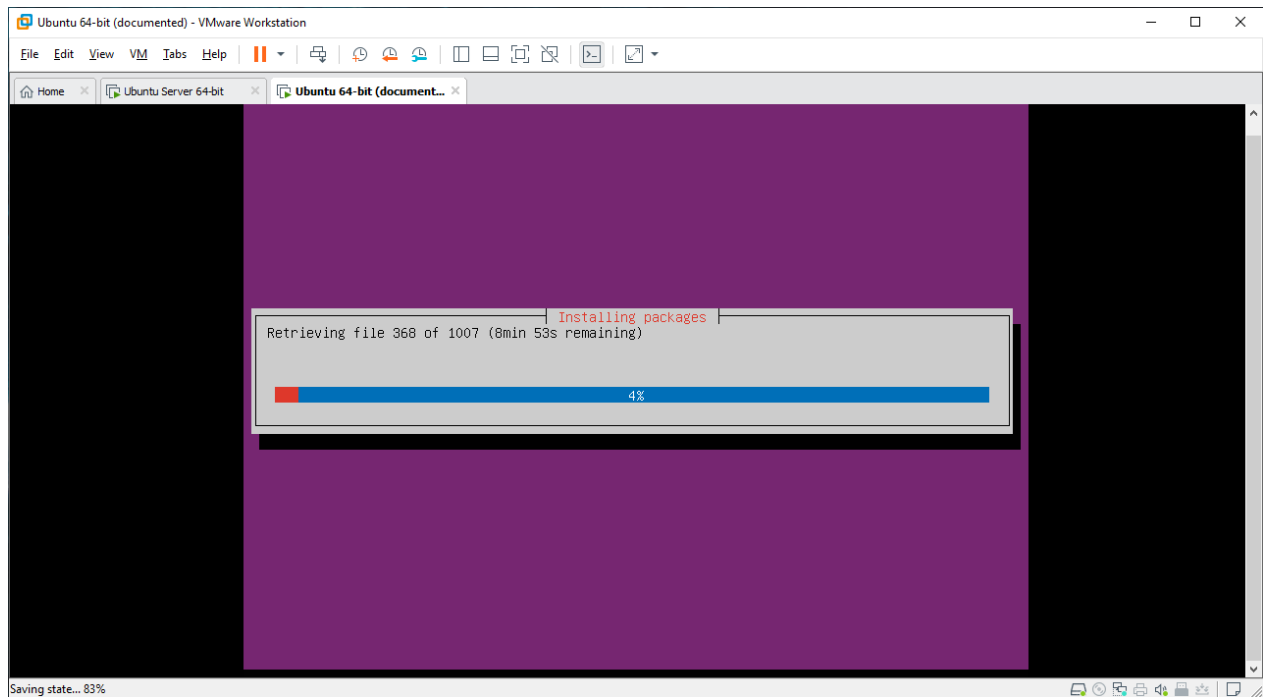
sudo -i

apt-get update

sudo apt install tasksel

tasksel --list-tasks

sudo tasksel install ubuntu-desktop



reboot

Install VMware tools;

```
sudo apt install open-vm-tools-desktop open-vm-tools
```

reboot

.....

```
sudo nautilus
```

Java installation;

```
sudo apt install default-jdk
```

```
sudo apt install default-jre
```

For configuring specific Java version;

```
sudo update-alternatives --config java
```

Configure JAVA_HOME env. var:

```
sudo nano /etc/environment
```

add the following:

```
JAVA_HOME="/usr/lib/jvm/java-11-openjdk-amd64"
```

check if it's set:

```
echo $JAVA_HOME
```

Tomcat installation;

Download>Web>

Add/Edit following paramaters;

server.xml :

```
<Connector port="8081" protocol="HTTP/1.1"
    connectionTimeout="20000"
    redirectPort="8443" />
```

tomcat users.xml :

```
<role rolename="manager-gui"/>
<user username="admin" password="admin" roles="manager-gui"/>
```

In Tomcat's directory "/bin" we can start or stop the server;

```
sh startup.sh
```

```
sh.shutdown.sh
```

*or create a service that will be start the same on startup

Login to Tomcat web server;

/manager - Mozilla Firefox

/manager

localhost:8081/manager/html

Recommendation

Tomcat Web Application Manager

Message: OK

Manager

[List Applications](#) [HTML Manager Help](#) [Manager Help](#) [Server Status](#)

Applications

Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/docs	None specified	Tomcat Documentation	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/examples	None specified	Servlet and JSP Examples	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/host-manager	None specified	Tomcat Host Manager Application	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/manager	None specified	Tomcat Manager Application	true	1	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes

Deploy

Deploy directory or WAR file located on server

Context Path:

Version (for parallel deployment):

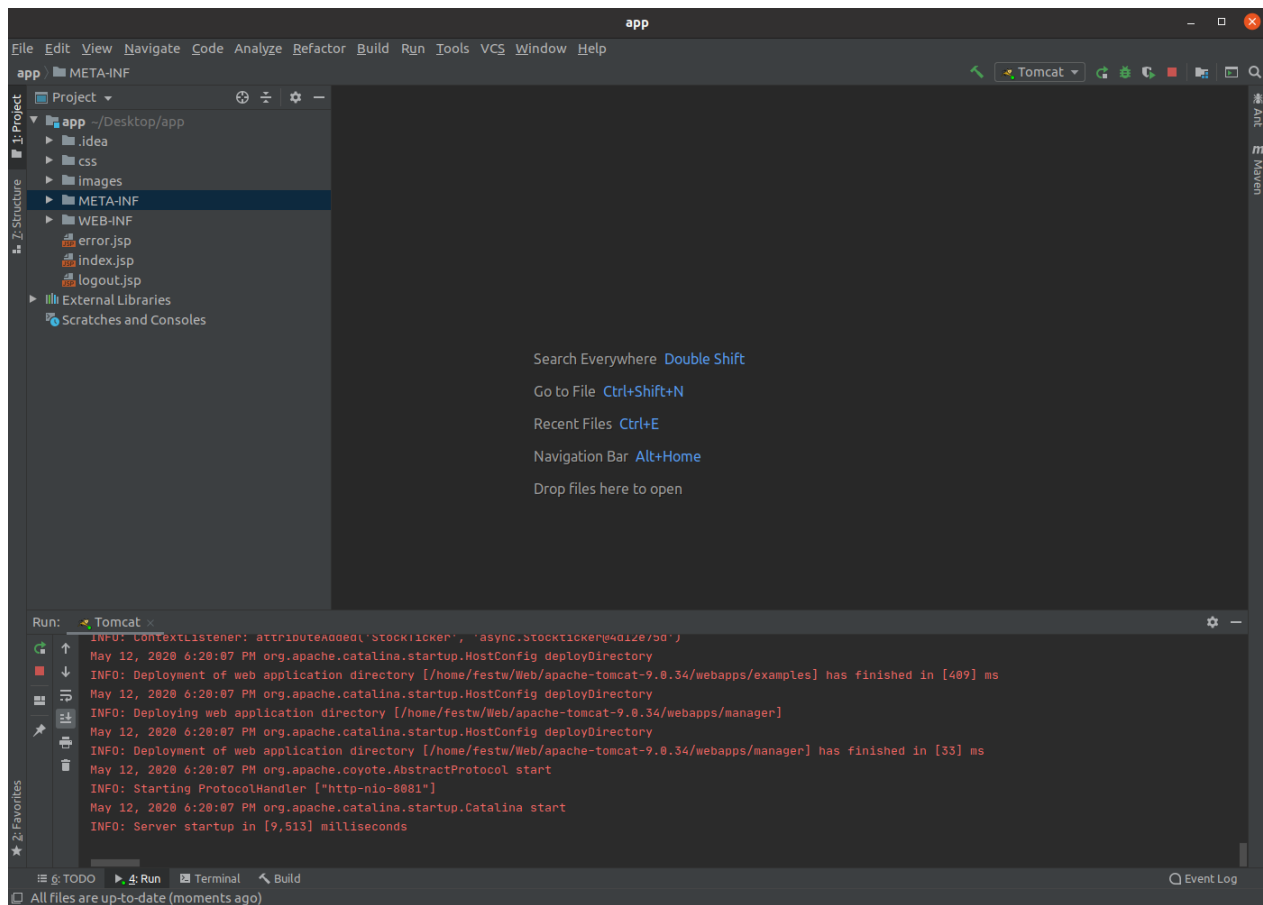
XML Configuration file path:

WAR or Directory path:

Deploy

Deploy downloaded .war via "War file to deploy"

Alternative is through IDE;



Setup Tomcat plugin and point it to installation folder.

TASK 1

The screenshot shows a web browser window displaying the "Spring SAML Sample application" interface. The page has a header with the Spring logo and the application title. Below the header, there are two tabs: "SAML Login" and "Metadata Administration", with the latter being the active tab. The main content area is titled "Login to administration" and includes a login form with fields for "User:" (containing "admin") and "Password:" (containing "****"), and a "Login" button. To the right of the login form is a "Useful links" section with links to "Homepage (Spring)", "Sources (GitHub)", "Documentation (Spring)", "Issue tracking (Jira)", and "Continuous integration (Bamboo)". Below this is a "Support" section with links to "Community (Stackoverflow)" and "Commercial (V7 Security)". Further down is a "Releases" section with links to "Final releases (Spring)", "Milestone releases (Spring)", and "Snapshot releases (Spring)". The footer contains copyright information: "© 2014 Vladimir Schäfer" and "Website template by Arcsin".

The browser's developer tools are open at the bottom, showing the "Elements" panel on the left and the "Styles" panel on the right. The "Elements" panel displays the HTML structure of the page, with the following snippet highlighted:

```
<a href="/test/saml/web/metadata">Metadata Administration</a> == $0
```

The "Styles" panel shows the CSS rules applied to the selected element, including:

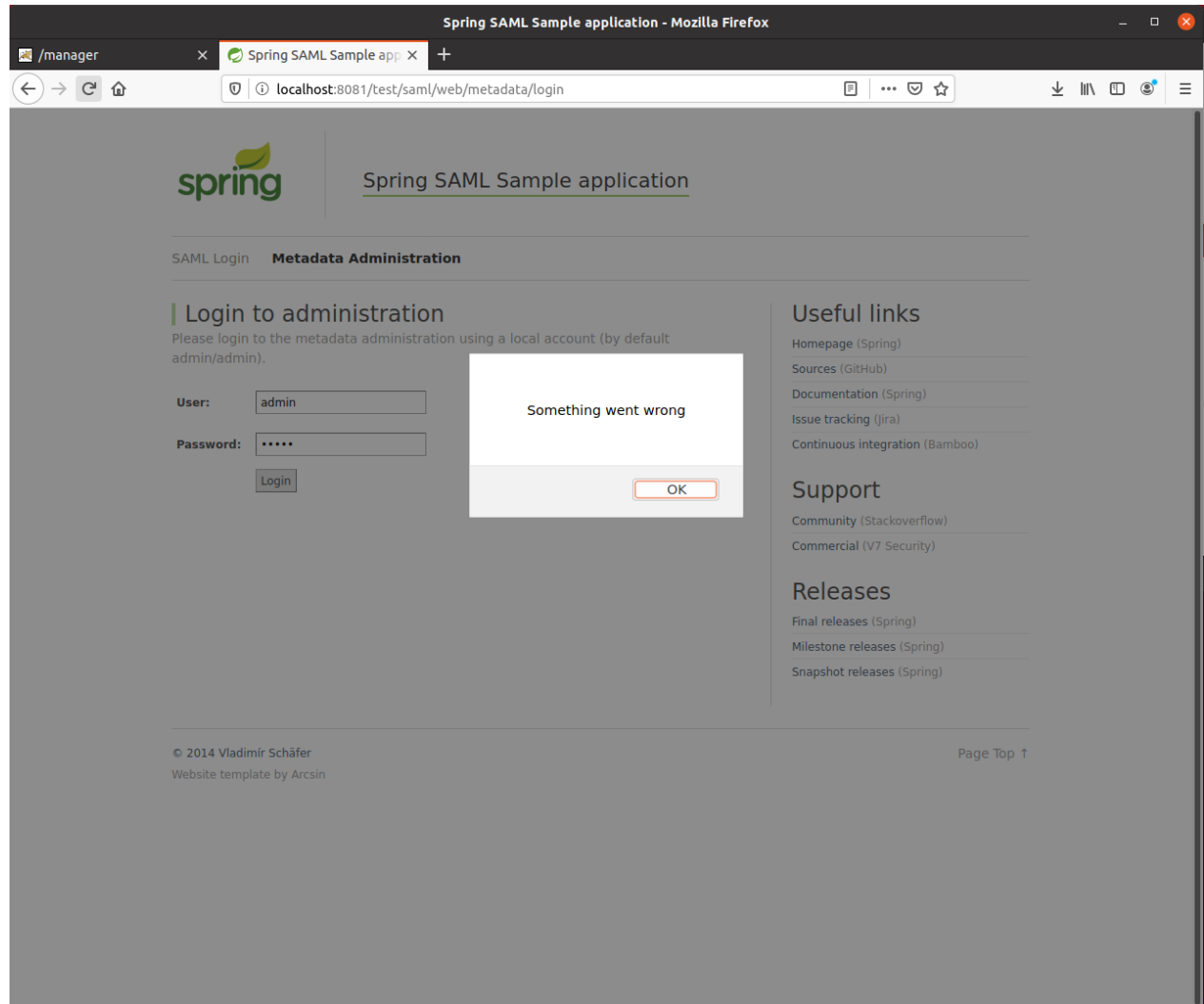
- `#sub-nav li.current-tab a { font-weight: bold; }` (style.css:386)
- `#sub-nav a { font-size: 1.2em; text-decoration: none; }` (style.css:381)
- `.navigation li.current-tab a { color: #e61212; }` (style.css:359)
- `.navigation a { color: #888; text-decoration: none; }` (style.css:354)
- `a { color: #456; text-decoration: none; }` (style.css:67)

The breadcrumb at the bottom of the developer tools shows the path: `html > body > div#site-wrapper > div#header > div#sub-nav > navigation > ul > li.current-tab > a`.

Proposing to decorate our CSS file with background color settings. so that "Metadata Administration" looks similar to this but with red background color attribute settings.

TASK 2

Find "Metadata Administration" and press "Login"



read article;

<https://docs.spring.io/spring-security-saml/docs/current/reference/html/chapter-sample-app.html>

Next Inspect Login Element.

Change this line so it looks like this;

```
<input name="submit" class="button" type="submit" value="Login" onclick="Login">
```

The screenshot shows a web browser window with the URL `localhost:8081/test/saml/web/metadata/login`. The page is titled "Spring SAML Sample application" and features a "Login to administration" section. The login form has fields for "User:" (containing "admin") and "Password:" (containing "****"), and a "Login" button. To the right of the login form is a "Useful links" section with links to "Homepage (Spring)", "Sources (GitHub)", "Documentation (Spring)", "Issue tracking (Jira)", and "Continuous integration (Bamboo)". Below this is a "Support" section with links to "Community (Stackoverflow)" and "Commercial (V7 Security)". At the bottom of the page, there is a "Releases" section with links to "Final releases (Spring)", "Milestone releases (Spring)", and "Snapshot releases (Spring)".

The browser's developer tools are open, showing the "Elements" panel. The selected element is the "Login" button, which is an `input` element with the following attributes: `name="submit", class="button", type="button", value="Login", onclick="alert('Something went wrong')"`. The "Styles" panel shows the default styles for the `input.button` class, including `background-color: #DDD;`, `border: 1px outset #AAA;`, and `padding: 4px 5px;`.

Proposing changing mentioned line of code to fix the issue in file;

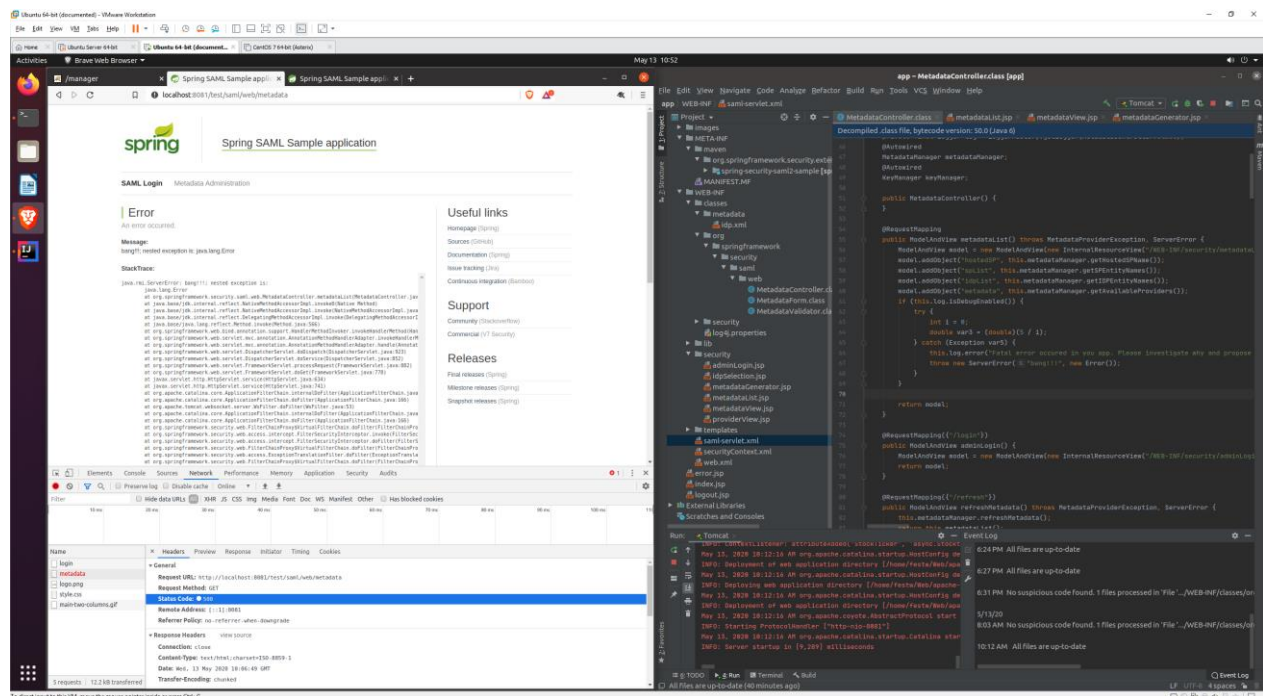
adminlogin.jsp

Check logs of log4j

TASK 3

Extract .war in some folder and open a project in favorite IDE(IntelliJ) for this purpose;

Error 500 is server side error – investigating this MVC app?



Metadatacontroller.java has some weird code inside it – seems that "@RequestMapping" annotation points to right .jsp file but i'm not sure if that weird code belongs here;

```
if (this.log.isDebugEnabled()) {
    try {
        int i = 0;
        double var3 = (double)(5 / i);
    } catch (Exception var5) {
        this.log.error("Fatal error occurred in you app. Please investigate why and propose 'quick and dirty' solution that will GO AROUND this problem!", var5);
        throw new ServerError("bang!!!", new Error());
    }
}
```

Why dividing by 0 (* / 0) ?

Is everything OK with our ID provider ?

Maybe issue that first routing only returns a "model" and not "modelandview" ?

When checking "metadatadatalist.jsp" i found that it looks fine, besides that "url" attributes inside paragraphs are empty ?

Can the following error tell smth about this error;

- No default metadata configured, generating with default values, please pre-configure metadata for production use*
- Generated default entity base URL <http://localhost:8081/test> based on values in the first server request. Please set property entityBaseURL on MetadataGenerator bean to fixate the value.*
- Created default metadata for system with entityID: <http://localhost:8081/test/saml/metadata>*