## Role Based Access Control-----Server Security

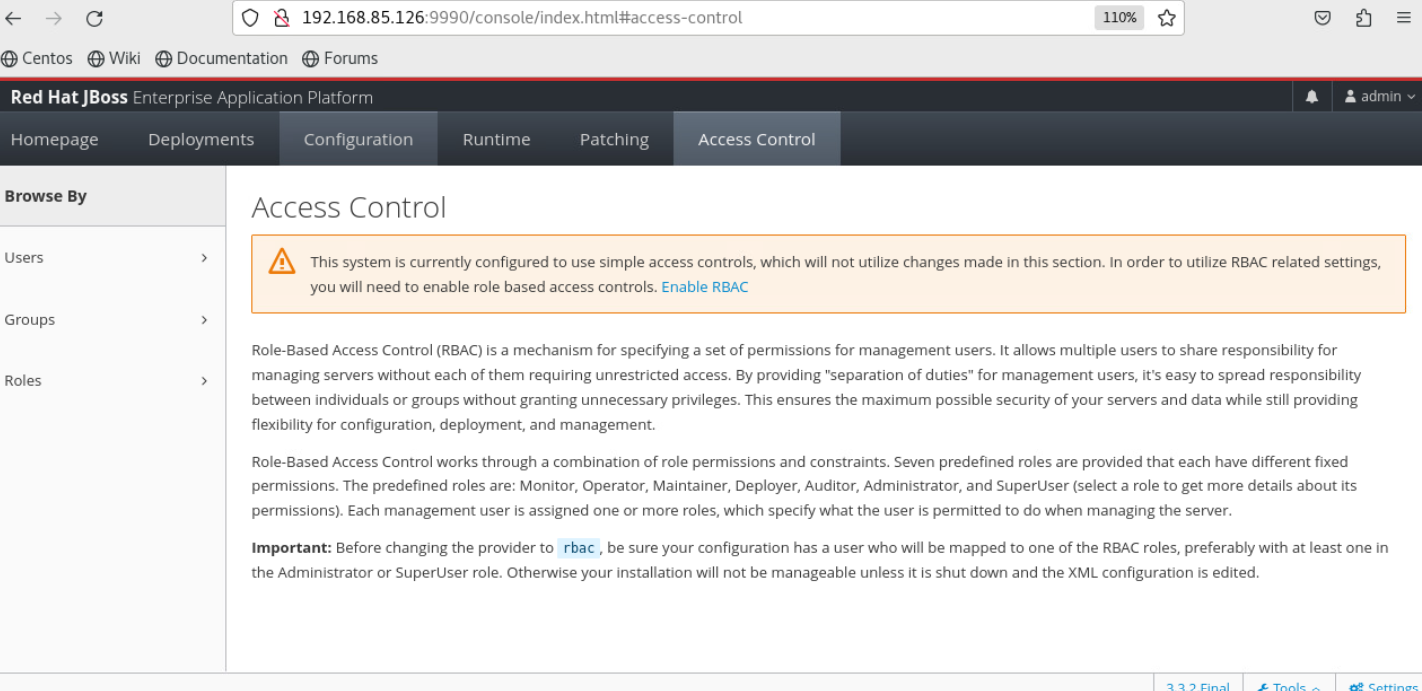
Role Based Access Control (RBAC) is the ability to restrict access to system or certain portions of it to authorized users. For JBoss AS 7.x, the web-based administrative console had an all-or-nothing approach. This means user authenticated with management security realm will have all the privileges. This may not be appropriate for mission-critical deployments and a finer-grained control may be required. WildFly 8 introduces RBAC using different roles.

**Purpose**: This section explains how to configure and use RBAC for WildFly.

There are seven pre-defined roles in two different categories. First four roles are where users are locked out of sensitive data and the next three level roles where users are allowed to deal with sensitive data.

The pre-defined roles are explained below:

| **Role** | **Permissions** |
| --- | --- |
| Monitor | * Has the fewest permissions * Can only read configuration and current runtime state * No access to sensitive resources or data or audit logging resources |
| Operator | * All permissions of Monitor * Can modify the runtime state, e.g. reload or shutdown the server, pause/resume JMS destination, flush database connection pool. * Does not have permission to modify persistent state. |
| Maintainer | * All permissions of Operator * Can modify the persistent state, e.g. deploy an application, setting up new data sources, add a JMS destination |
| Deployer | * All permissions of Maintainer * Permission is restricted to applications only, cannot make changes to container configuration |
| Administrator | * All permissions of Maintainer * View and modify sensitive data such as access control system * No access to administrative audit logging system |
| Auditor | * All permissions of Monitor * View and modify resources to administrative audit logging system * Cannot modify sensitive resources or data outside auditing, can read any sensitive data |
| Super User | * Has all the permissions * Equivalent to administrator in previous versions   Server Security |

****

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

****

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**Select Add Button**

**A screenshot of a computer

Description automatically generated**

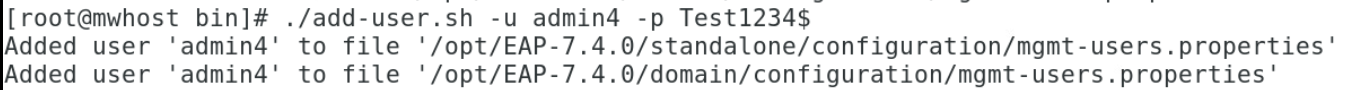
**Repeat for adding remaining users**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

****

****

**A close-up of a computer screen

Description automatically generated**

**Disabling RBAC**

* To disable RBAC with jboss-cli.sh use the write-attribute operation of the access authorization resource to set the provider attribute to simple.

/core-service=management/access=authorization:write-attribute(name=provider, value=simple)

[standalone@localhost:9999 /] /core-service=management/access=authorization:write-attribute(name=provider, value=simple)

{

"outcome" => "success",

"response-headers" => {

"operation-requires-reload" => true,

"process-state" => "reload-required"

}

}

[standalone@localhost:9999 /] /:reload

{

"outcome" => "success",

"result" => undefined

}

[standalone@localhost:9999 /]

If the server is off-line the XML configuration can be edited to enabled or disable RBAC. To do this, edit the provider attribute of the access-control element of the management element. Set the value to rbac to enable, and simple to disable.

<management>

<access-control provider="rbac">

<role-mapping>

<role name="SuperUser">

<include>

<user name="$local"/>

</include>

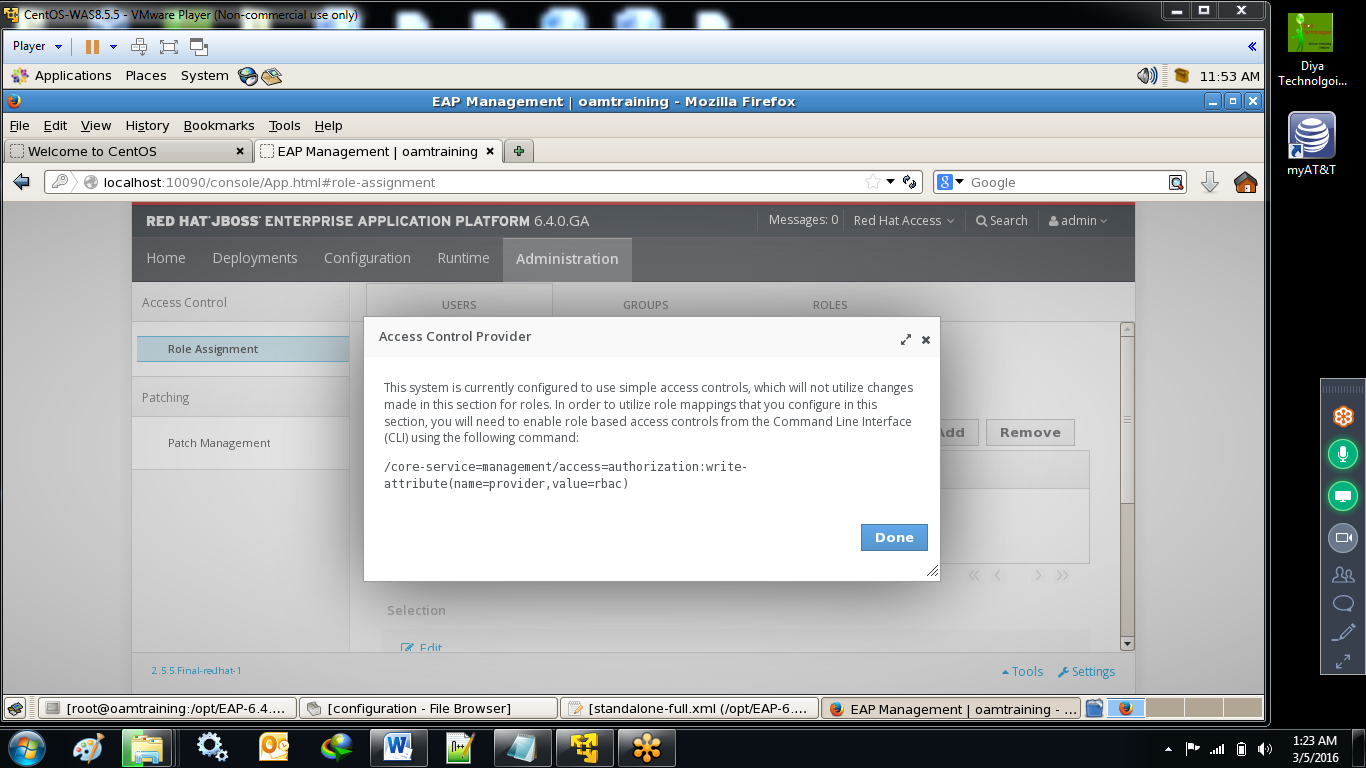
</role>

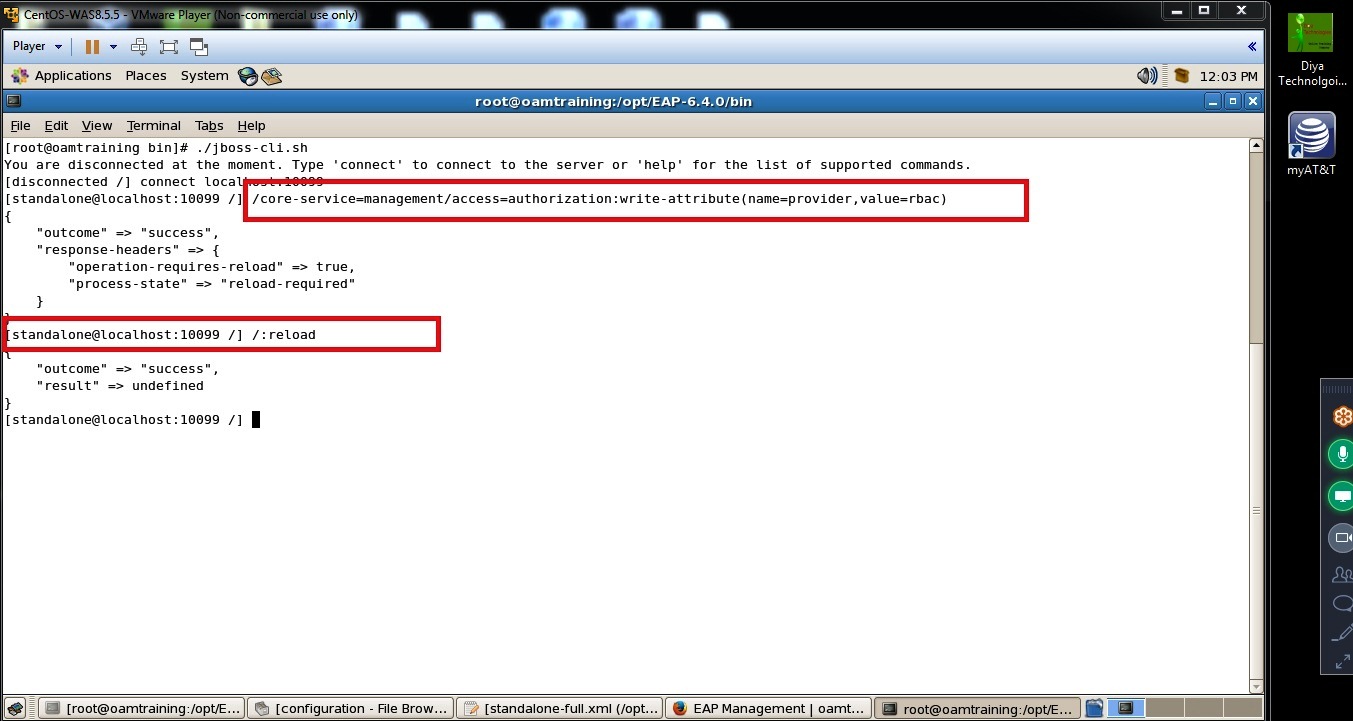
</role-mapping>

</access-control>

</management>

**Step -1**





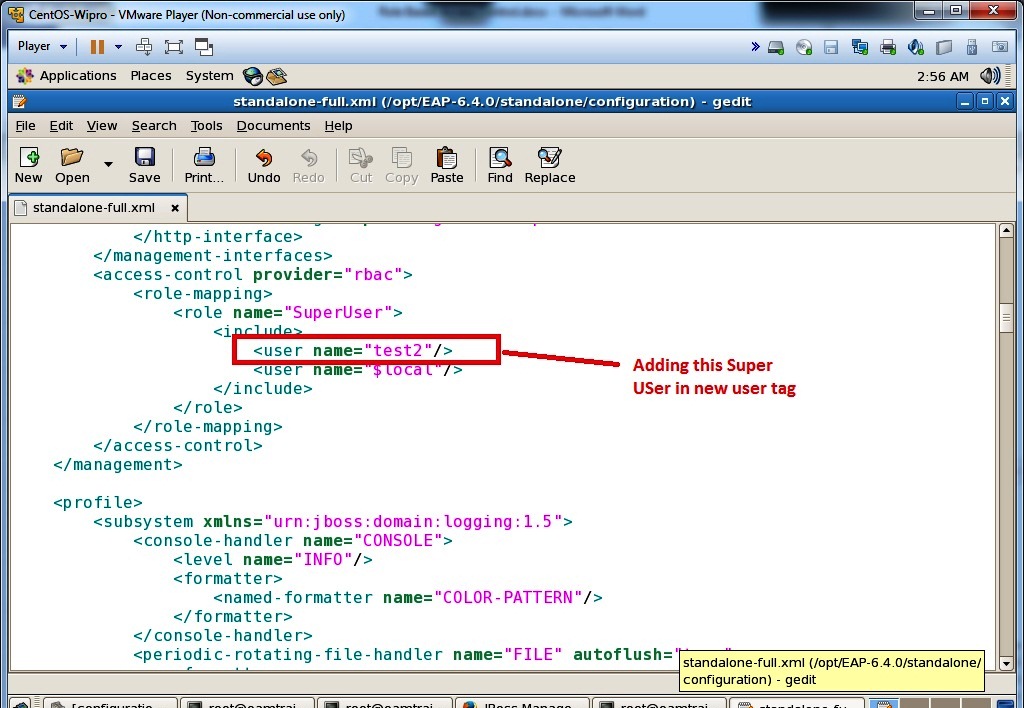
**Step -2**

First Create a Super user using

./add-user.sh –u testsuperuser1 -p test1234$

And add that user in standalone.xml or domain.xml

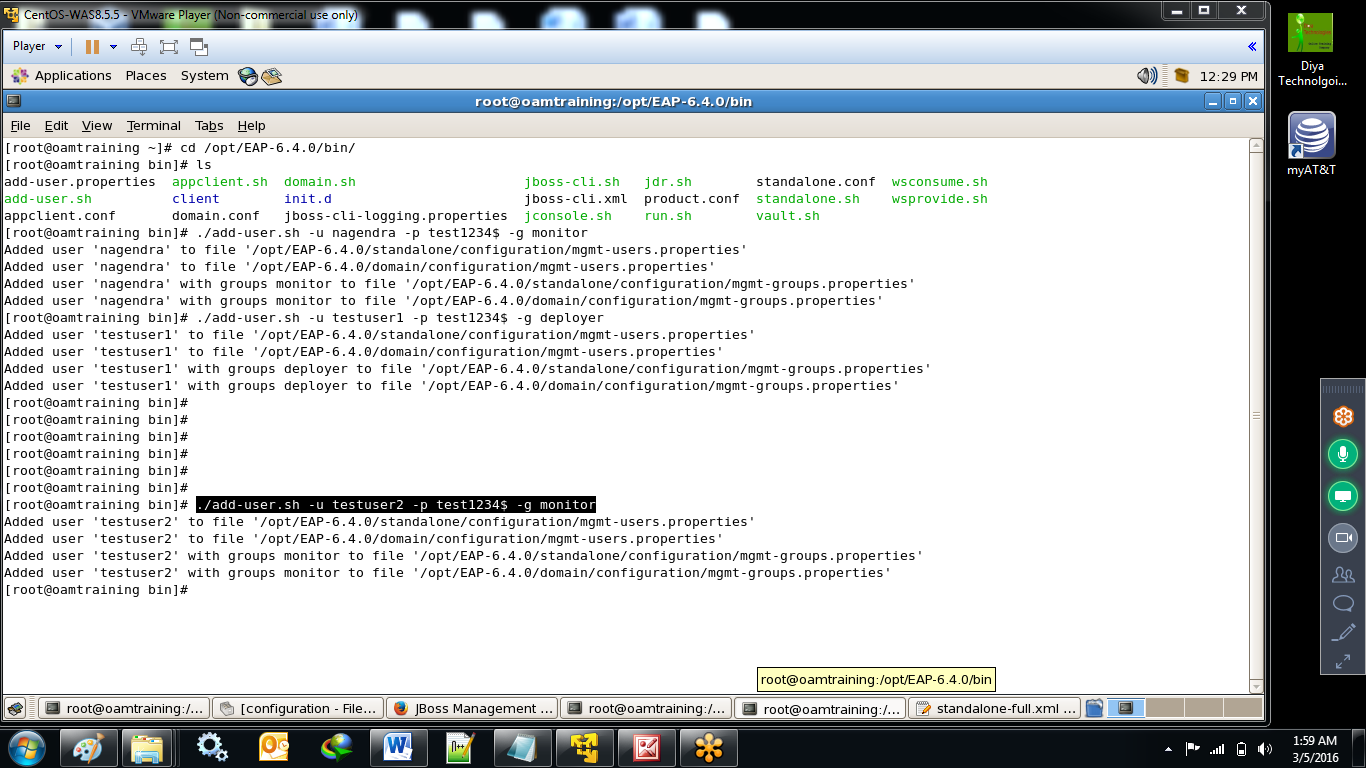
example: add-user.sh -u leonard -p leonard1 -g geek



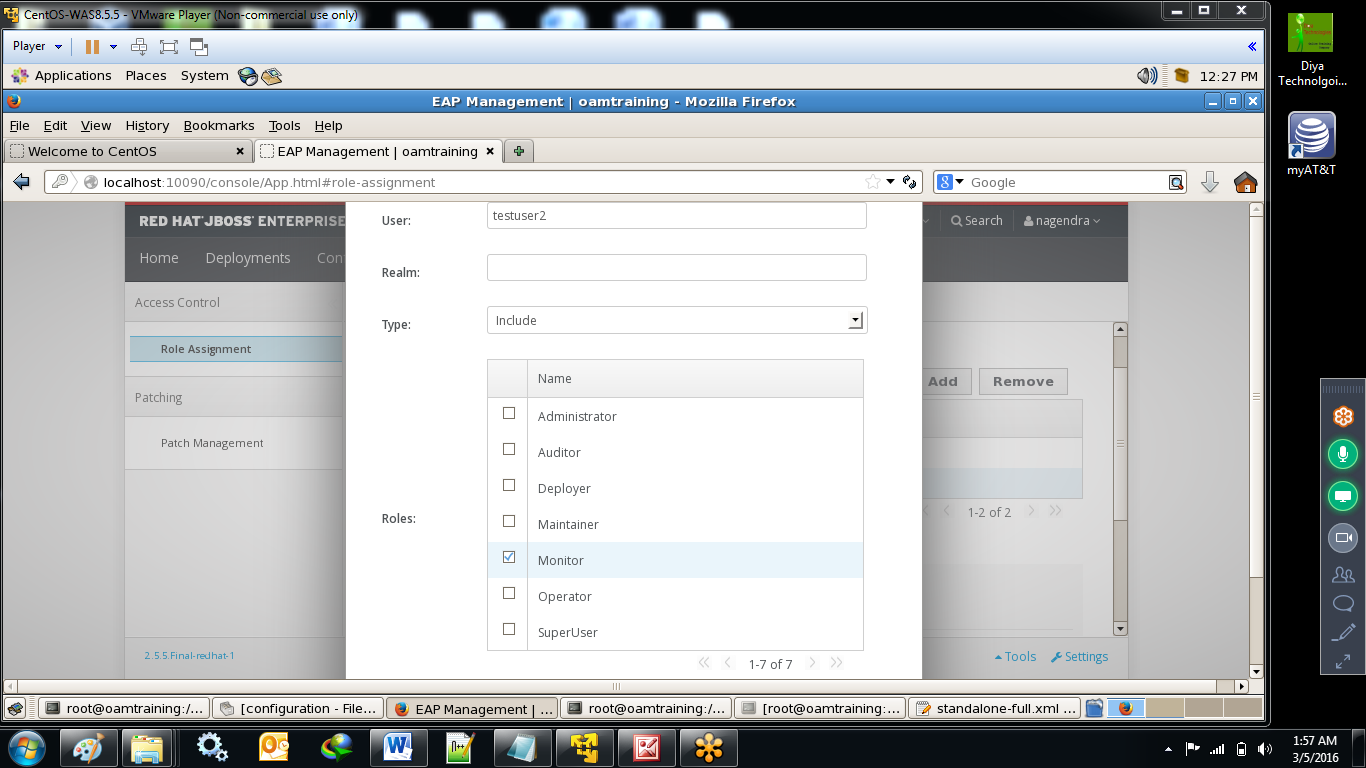
And Login using this User Only

**Step -3:**Assign the username/password for the user using add-user.sh

./add-user.sh –u username –p password –g monitor



Step-4 :map the created user to role in admin console (No Password for created user) username: testuser2 role: monitor



Step-3-Login and check the privileges

