

Institute of Computer Technology

B. Tech Computer Science and Engineering

Sub: Identity and Access Management (2CSE507)

Name - Nisarg Prajapati

Branch - Cyber Security

Enrollment no. – 23162171019

Semester - 5

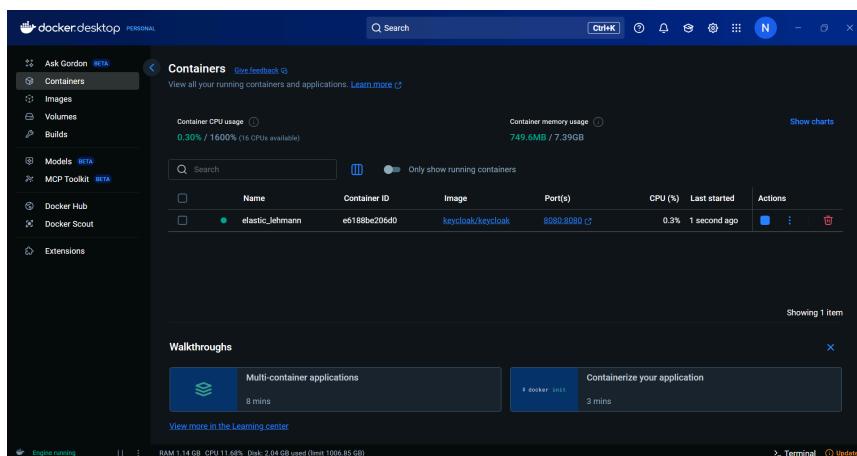
Class - A

Batch – 52

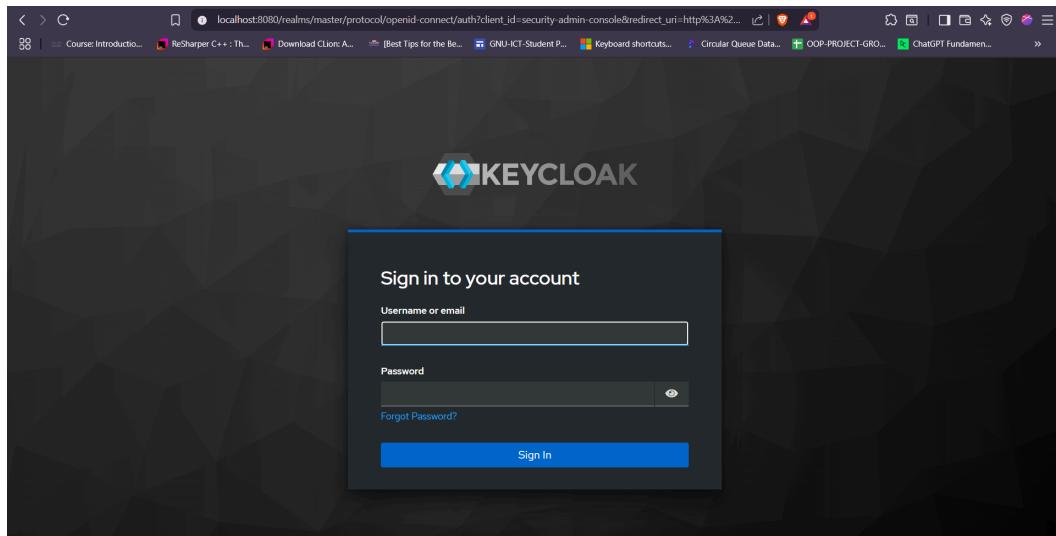
PRACTICAL NO:- 2

Starting the Container:

- Open your Docker Desktop.
- In the containers tab, click the start button of the container you want to start.

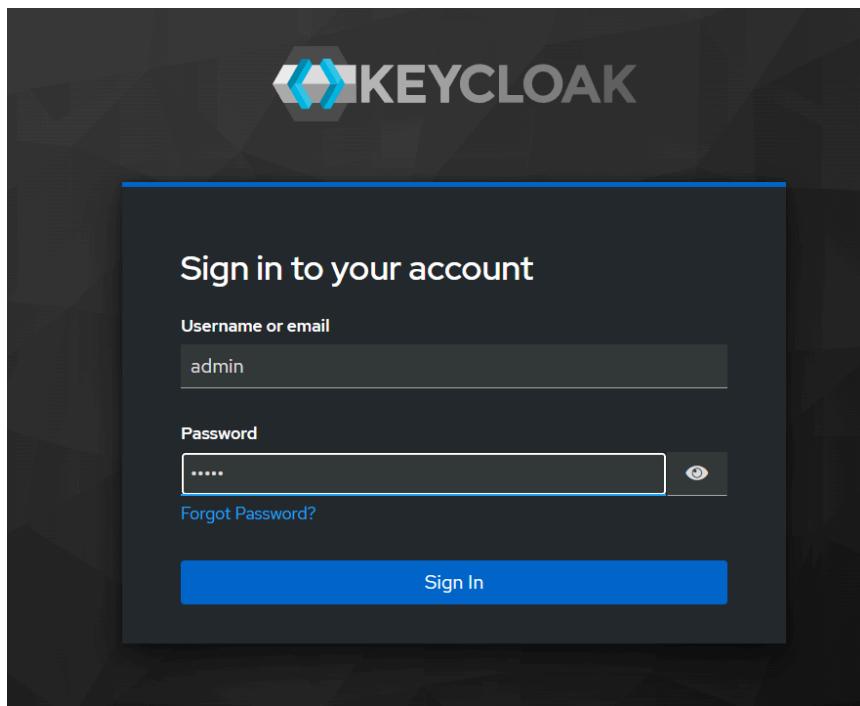


- Now open your browser and go to '<http://localhost:8080/>'.



Creating a new user:

- Login as admin.



- Go to the user tab and click on ‘Add User’.

The screenshot shows the Keycloak 'Users' page. The left sidebar has 'Manage' selected, and 'Users' is highlighted. The main area displays a table of users with columns: Username, Email, Last name, and First name. Two users are listed: 'admin' and 'user1'. A blue 'Add user' button is located at the top right of the table area.

- Fill all required fields like username, password, etc.
- Also select actions that the user must do after logging-in first like Update passwords, Update profile, email verification, etc.

The screenshot shows the 'Create user' form. The left sidebar has 'Manage' selected, and 'Users' is highlighted. The main form has a 'Required user actions' section with a dropdown set to 'Select action'. Below it is a 'Email verified' toggle switch set to 'Off'. The 'General' section contains fields for 'Username' (with a red asterisk), 'Email', 'First name', and 'Last name'. A 'Groups' section with a 'Join Groups' button is also present. At the bottom are 'Create' and 'Cancel' buttons.

Updating User Details:

- Go to the user tab.
- Click on the user, whose details you want to change
- You will find following subtabs :
 - **Details** : View and edit basic user info (username, email, status, etc.).
 - **Credentials** : Manage user passwords and other credentials (reset, view status).
 - **Role Mapping** : Assign or remove roles to define user permissions.
 - **Groups** : View and manage group memberships for the user.
 - **Consents** : Review user consents for client scopes and revoke if needed.
 - **Identity Provider Links** : Link external identity providers (e.g., Google, GitHub) to the user.
 - **Sessions** : View active user sessions and optionally revoke them.
 - **Events** : View login, logout, and error events related to the user.

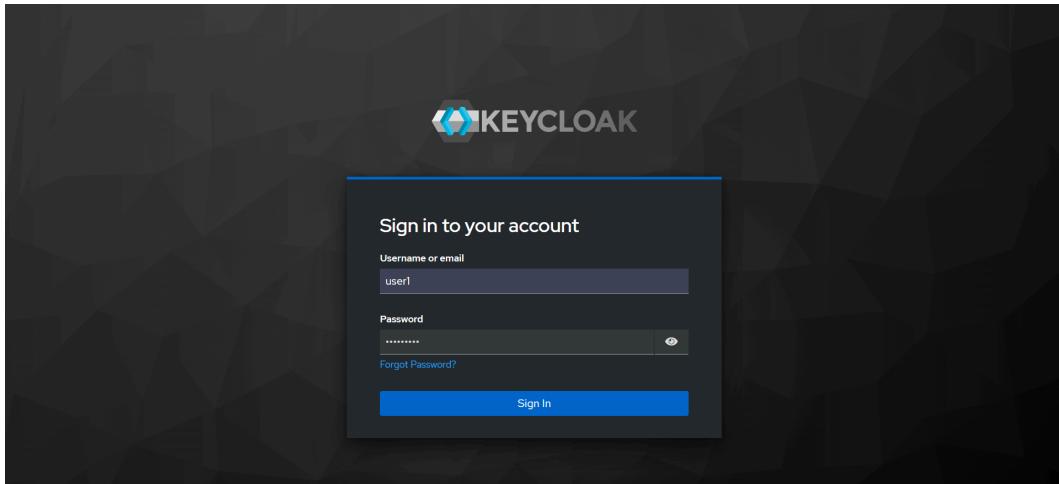
The screenshot shows the Keycloak User Details page for a user named "user1". The user is listed with the following details:

- ID**: 6laecblf-16b0-4c6f-90dc-3b9b6c488455
- Created at**: 7/28/2025, 11:40:05 AM
- Email verified**: Off

The page features a navigation bar with tabs for Details, Credentials, Role mapping, Groups, Consents, Identity provider links, Sessions, and Events. The "Details" tab is currently selected. There are "Save" and "Revert" buttons at the bottom of the form.

Logging-in as the new user:

- Open ‘localhost:8080’ in the new browser.
- Enter the username and password of the new user that you just created.



- Complete all the tasks that must be completed after first login like Update passwords, Update profile, email verification, etc.

Resetting a user's password:

- Go to the user tab.
- Click on the user, whose password you want to change.

- Go to the ‘credentials’ tab.

The screenshot shows the Keycloak administration interface. On the left, a sidebar menu is visible with options like 'Manage realms', 'Clients', 'Client scopes', 'Realm roles', 'Users' (which is selected), 'Groups', 'Sessions', 'Events', 'Configure', 'Realm settings', and 'Authentication'. The main content area is titled 'User details' for a user named 'user1'. The 'Credentials' tab is active, showing a table with one row: a password entry. A 'Reset password' button is located at the bottom right of the table. The top right corner shows the user is 'Enabled' and has an 'Action' dropdown.

- Click on ‘Reset password’ and enter a new temporary password.

This screenshot shows the same Keycloak interface as the previous one, but with a modal dialog box overlaid. The dialog is titled 'Set password for user1' and contains three fields: 'Password' (marked with a red asterisk), 'New password confirmation' (also marked with a red asterisk), and a 'Temporary' toggle switch which is set to 'On'. Below the fields are 'Save' and 'Cancel' buttons. The background of the main interface is dimmed to indicate it is not active while the modal is open.

Setting password policies:

- Go to the ‘authentication’ tab.

The screenshot shows the Keycloak 'Authentication' page. On the left, there's a sidebar with 'Manage' and 'Authentication' sections. Under 'Authentication', 'Flows' is selected. The main area has tabs for 'Flows', 'Required actions', and 'Policies'. Below is a table of flows:

Flow name	Used by	Description
browser	Built-in	Browser based authentication
clients	Built-in	Client authentication flow
direct grant	Built-in	Direct grant flow
docker auth	Built-in	Docker authentication flow
first broker login	Built-in	First broker login flow
registration	Built-in	Registration flow
reset credentials	Built-in	Reset credentials flow

- Go to the ‘policies’ tab and in the ‘password policy’ tab you can add the policies you want from the dropdown like min and max length, min amount of special character(s), etc.

The screenshot shows the Keycloak 'Authentication' page with the 'Policies' tab selected. The sidebar shows 'Manage realms' and 'Authentication' sections. Under 'Authentication', 'Policies' is selected. There are tabs for 'Password policy', 'OTP Policy', 'Webauthn Policy', 'Webauthn Passwordless Policy', and 'CIBA Policy'. The 'Password policy' tab is active. It has a 'Save' and 'Reload' button at the bottom. The configuration fields are:

- Minimum Length: 6
- Special Characters: 1
- Uppercase Characters: 1
- Digits: 1
- Lowercase Characters: 1

What is the difference between docker and Virtualization? Why do we use docker?

Features	Docker	Hypervisor
Type	Container-based virtualization	Hardware-based (VM) virtualization
Function	Runs applications in isolated containers	Runs entire OS instances (VMs)
Abstraction Level	OS-level virtualization	Hardware-level virtualization
What It Virtualizes	Only the application and its dependencies	Entire operating system and hardware
Host OS Shared?	Yes (containers share the same host OS kernel)	No (each VM runs its own OS instance)
Guest OS Required	No	Yes
Resource Usage	Lightweight (no guest OS overhead)	Heavy (each VM needs its own OS)
Performance	Near-native	Slightly lower due to VM overhead
Isolation Level	Process-level (less isolated)	Full OS-level (more isolated)
Security	Shared kernel (risk if compromised)	Better security due to full isolation
Portability	Very portable (container images)	Less portable (VM images are larger)
Typical Use Case	Microservices, DevOps, CI/CD, lightweight apps	Full OS testing, legacy app support, multiple OS