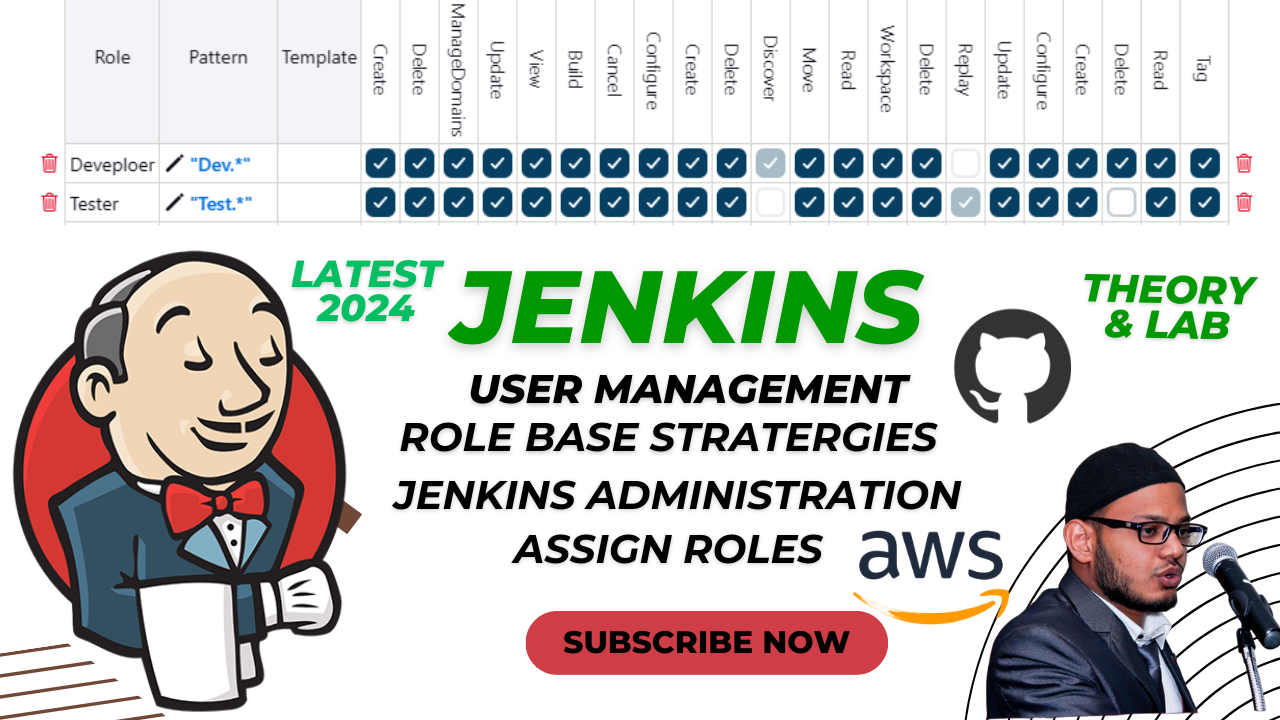
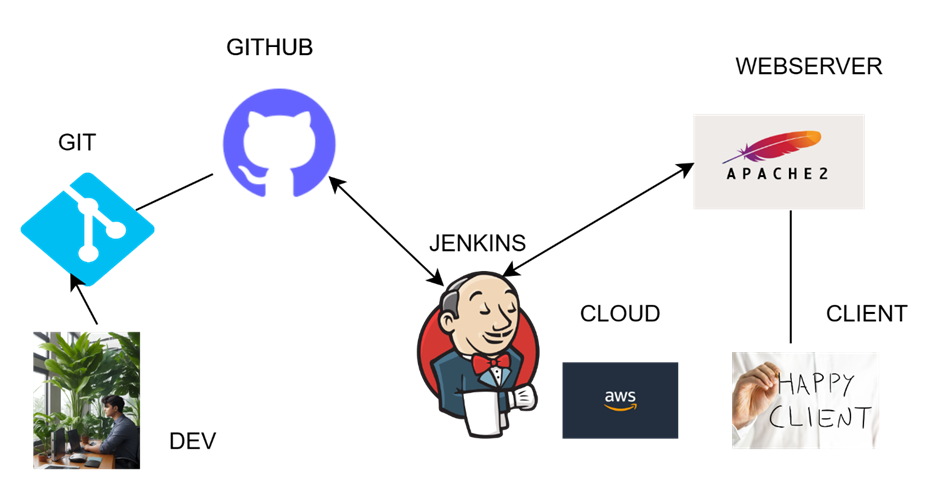
**JENKINS ADMINISTRATION – USER MANAGEMENT**

* **Understanding Jenkins User Management: What, Why, and Where**

**Link for youtube video** [**https://www.youtube.com/watch?v=9278t8ykuGo**](https://www.youtube.com/watch?v=9278t8ykuGo)





Jenkins user management involves controlling and managing user access to Jenkins, an open-source automation server. Here are some key aspects of Jenkins user management:

**Title: Understanding Jenkins User Management: What, Why, and Where**

**Introduction:** Jenkins, an open-source automation server, plays a crucial role in facilitating continuous integration and continuous delivery (CI/CD) in software development. Effective Jenkins user management is essential for ensuring secure and streamlined processes within the CI/CD pipeline.

**What is Jenkins User Management?** Jenkins User Management involves the creation, configuration, and administration of user accounts within the Jenkins environment. This includes defining user roles, permissions, and access levels to various Jenkins’s functionalities.

**Why is Jenkins User Management Important?**

1. **Access Control:** User management allows administrators to control access to Jenkins resources. Different users may have different levels of authorization, ensuring that only authorized personnel can perform critical actions.
2. **Security:** By managing user accounts, Jenkins administrators can enforce security measures, reducing the risk of unauthorized access or malicious activities. This is particularly crucial when dealing with sensitive build and deployment processes.
3. **Collaboration:** In a collaborative development environment, multiple team members may interact with Jenkins. User management ensures that each team member has the appropriate permissions, fostering collaboration without compromising security.
4. **Audit Trail:** User management provides an audit trail, allowing administrators to track changes made within Jenkins. This helps in identifying and rectifying any unauthorized or suspicious activities.

**Where is Jenkins User Management Applicable?**

1. **Software Development Teams:** Jenkins is widely used in software development pipelines. User management is crucial in such environments to define roles for developers, testers, and administrators, ensuring each has the necessary access for their tasks.
2. **Enterprise Environments:** In large enterprises, multiple teams may use Jenkins for various projects. Effective user management ensures that each team has its own space and permissions, preventing interference between projects.
3. **Continuous Integration and Continuous Delivery (CI/CD):** Jenkins is a key component of CI/CD pipelines. User management is applied to control who can trigger builds, deploy applications, and access reports, aligning with the principles of CI/CD.

**User Management Process in Jenkins:**

1. **User Creation:** Administrators create user accounts for individuals involved in the Jenkins environment.
2. **Role Definition:** Different roles (e.g., admin, developer, tester) are defined, each with specific permissions.
3. **Permission Assignment:** Users are assigned to appropriate roles, determining their access levels.
4. **Regular Audits:** Periodic audits are conducted to review and update user roles based on project requirements.
5. **Password Policies:** Administrators implement strong password policies to enhance security.
6. **Education and Training:** Users are educated on security best practices and the importance of adhering to the user management policies.

In conclusion, Jenkins User Management is a fundamental aspect of maintaining a secure and efficient CI/CD environment. By defining roles, controlling access, and regularly auditing user accounts, organizations can harness the full potential of Jenkins while ensuring a robust and secure software development lifecycle.

Jenkins user management involves controlling and managing user access to Jenkins, an open-source automation server. Here are some key aspects of Jenkins user management:

1. **Authentication:**
   * Jenkins supports various authentication methods, such as:
     + **Jenkins Internal Database:** You can create and manage user accounts within Jenkins itself.
     + **LDAP (Lightweight Directory Access Protocol):** Integrate Jenkins with an LDAP server for centralized user authentication.
     + **Single Sign-On (SSO):** Use external authentication providers for seamless login.
2. **Authorization:**
   * Once users are authenticated, you can define their permissions and access levels based on roles.
   * Jenkins supports matrix-based security, where you can assign specific permissions to users or groups for different job actions.
3. **Roles and Groups:**
   * Organize users into groups and assign roles to these groups for easier management.
   * Define specific roles such as administrators, developers, testers, etc., and assign appropriate permissions.
4. **Project-based Security:**
   * Jenkins allows you to configure security settings on a per-project basis.
   * You can control who has access to specific jobs, builds, and other project-related actions.
5. **User Self-Registration:**
   * Optionally, Jenkins can allow users to register themselves, or administrators can manually create user accounts.
6. **Password Policies:**
   * Define password policies to enhance security, such as password complexity requirements and expiration periods.
7. **Audit Trails:**
   * Jenkins provides audit trails to track user activities and changes made to security settings.
8. **API Access:**
   * Users with the necessary permissions can interact with Jenkins programmatically through its API.

To manage users in Jenkins, you typically navigate to the "Manage Jenkins" section in the Jenkins web interface and access the "Configure Global Security" page. From there, you can set up authentication, define authorization strategies, and configure other security-related settings.

Login with admin administration passwords

**LAB :**

Prerequisite :

User: admin

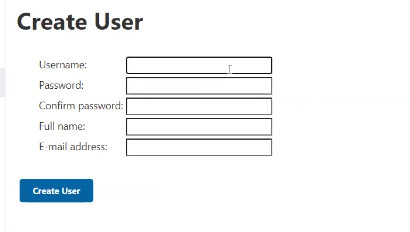
Pass : admin123

Manage Jenkins:

Read access , full access , create access , superadmin , job level

Creating a Jenkins user -1

Users click & create user

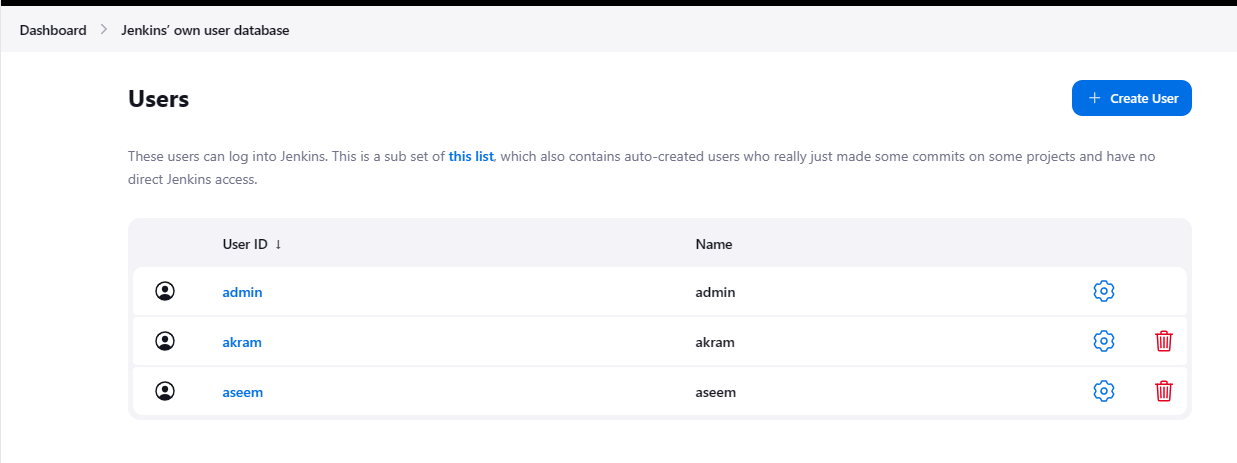


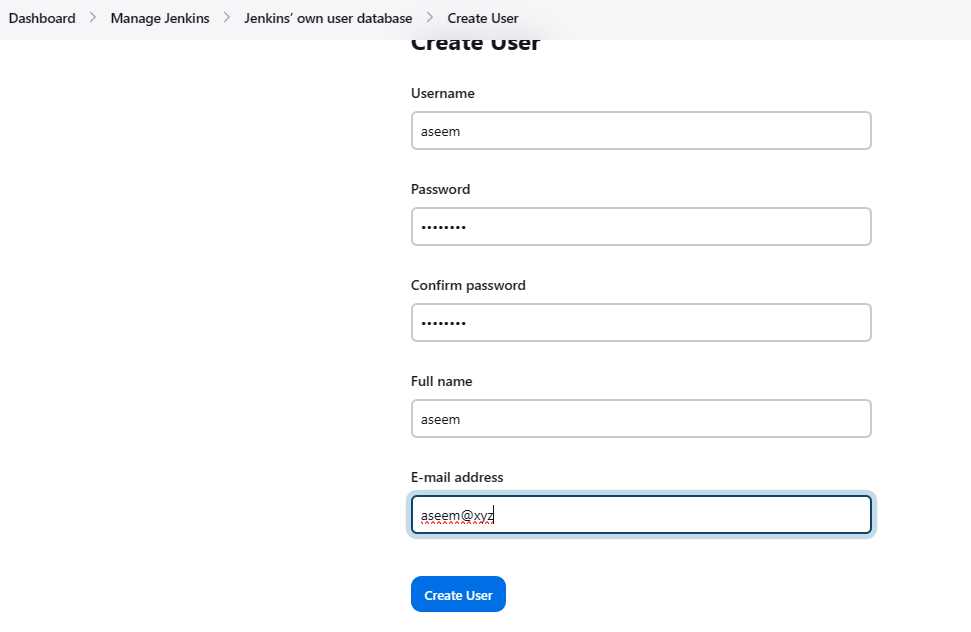
**Username : aseem**

Password : aseem123

Fullname : aseemakram

Email : [aseem@gmail.com](mailto:aseem@gmail.com)





Logout of the Jenkins

Akram

Akram123

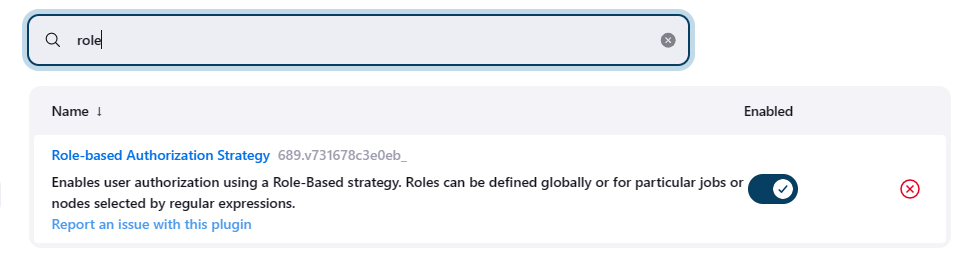
Test the user

1. revoke completes access from them and provide them limited or restricted permissions

Login with above credentials. acts like as admin privileges

manage Jenkins –

plugins – check for role (plugin – **Role-based authorized strategy**)



install

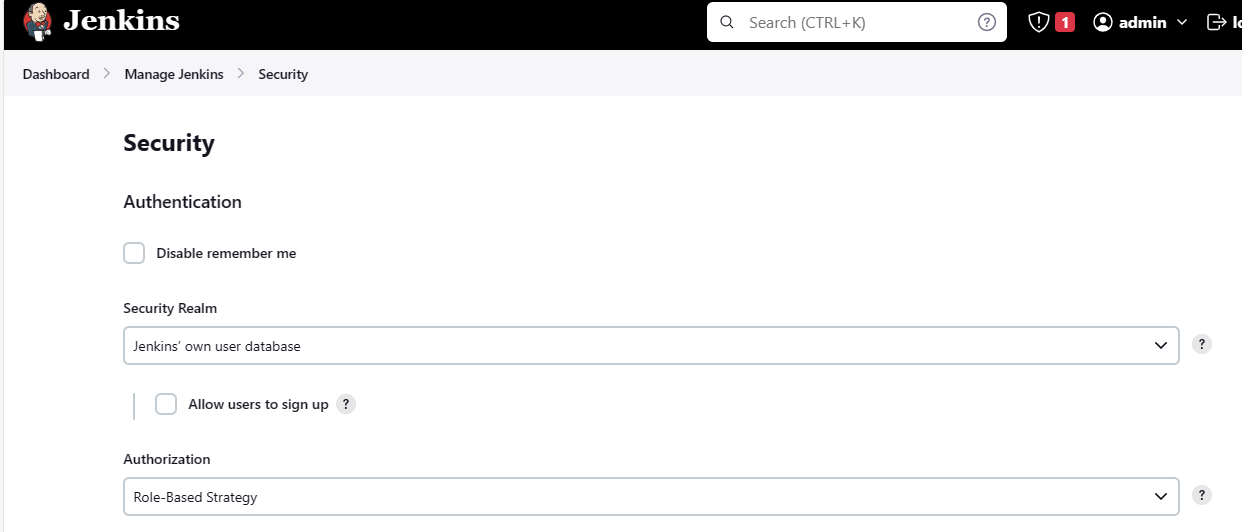
Restart the Jenkins:

* manage Jenkins – security

Authentication

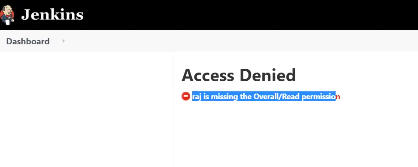
Security Realm – Jenkins own user DB

Authorization – click on Role-based-strategy/



Click on apply & save

Login with user credential that the permissions are revoked .

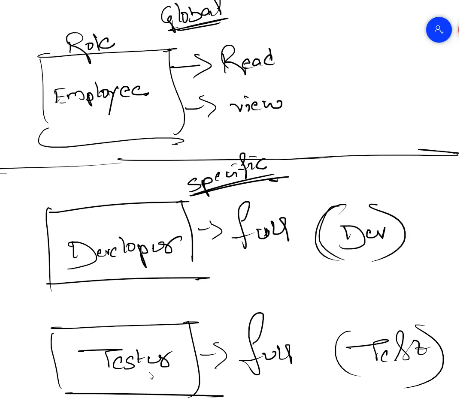


Drill to check the user aseem login attempt : access read restricted.

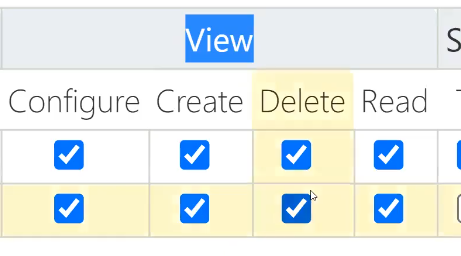
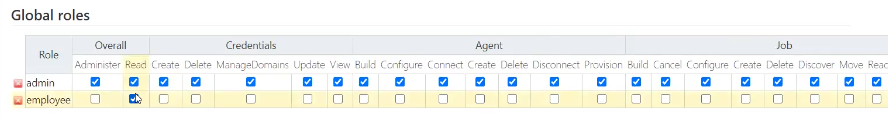
**Username : aseem**

Login with admin in to Jenkins

Manage Jenkins: manage and assign roles



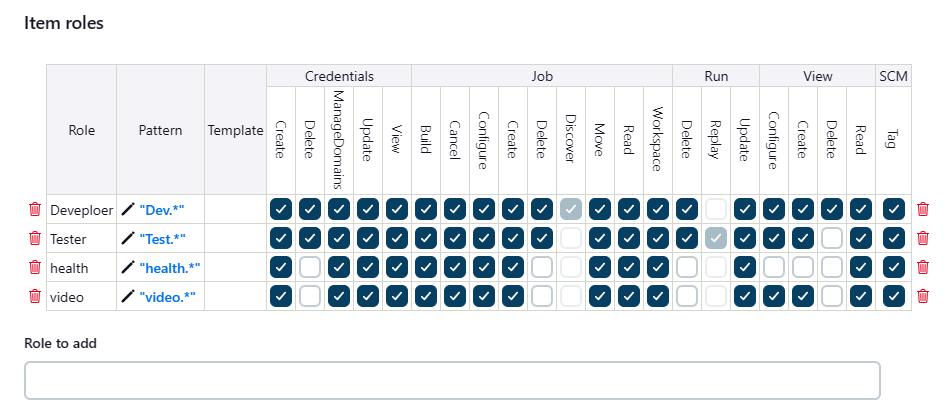
* Create employee role – global rule that is globals with limited acess – only read & view view



* Developer -item role - dev related jobs , after dev it could be anything

Pattern - Dev.\* -

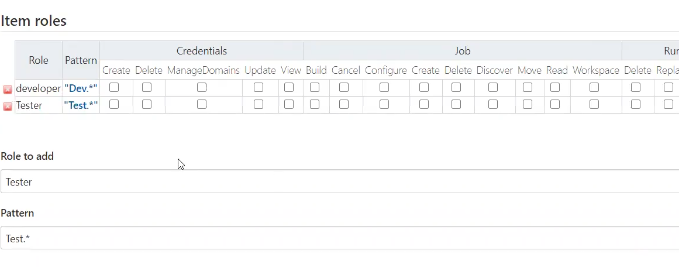
Provide full access to dev item role

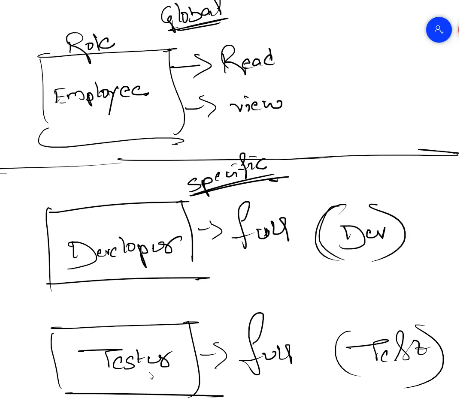


* Tester- item role- test related jobs, after test can see only test

Pattern - Test.\* -

Provide full access to dev item role all





**Agenda**

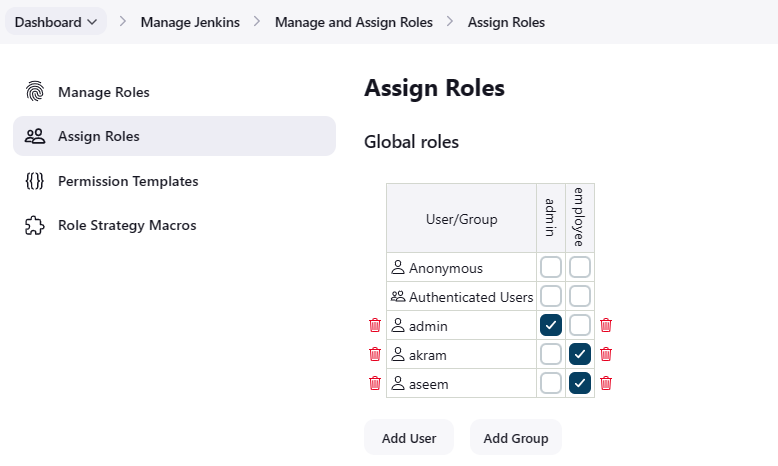
aseem– dev team only access to dev jobs only

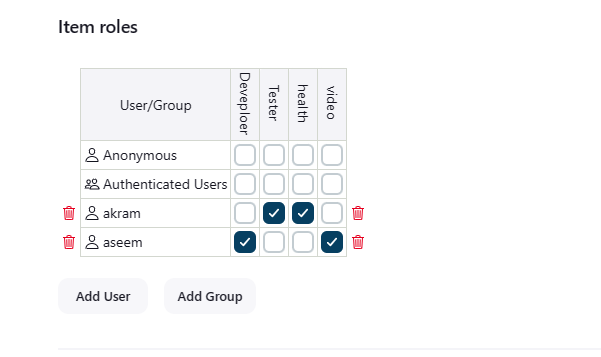
akram – test team only access to dev jobs only

Item role -specific roles

Aseem - to be add to dev

Akram- to be add to test





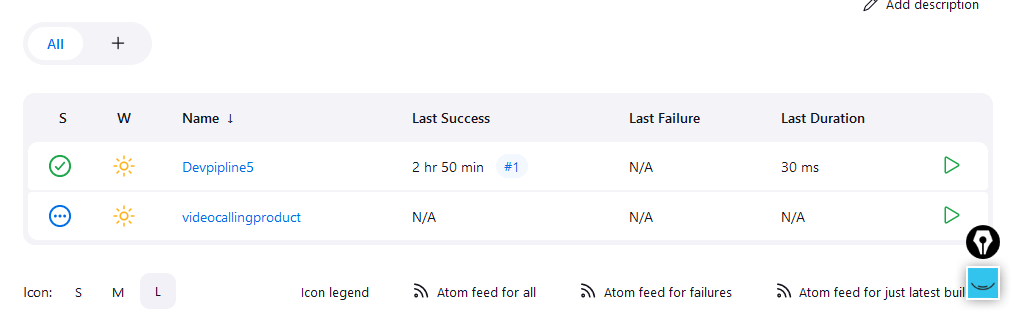
Apply & save.

Assign role = add user created in step 1 – aseem

Employees role in assign

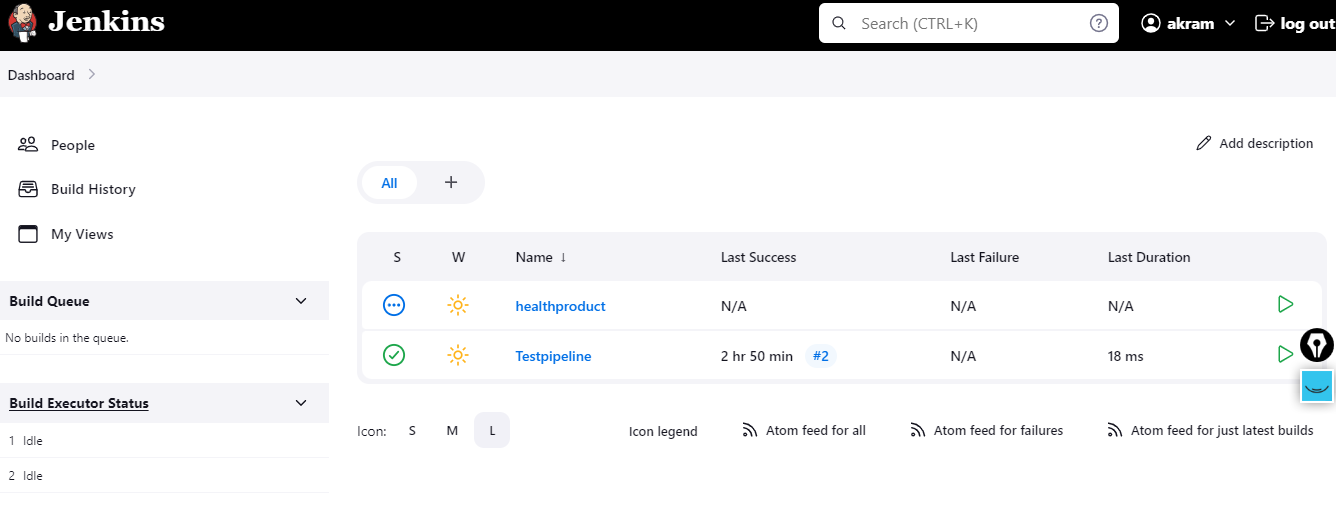
**TEST cases**

* Login with aseem user to view the Jenkins dashboard

Able to see DEV & video product jobs in the dashbpard  


* Login with akram user to view the Jenkins dashboard

Test jobs & Health product related jobs are visible.   
able to see all the jobs & create & delete , jobs etc .



\*\*\*\*\*\* Drill to remove delete , build to full access role etc. restricted via admin privileges .

Login with user , restricted permission are applied .