Jenkins Authentication

Jenkins is an open-source automation server that can be used to automate different stages of the software delivery process. It provides various authentication mechanisms to secure access to Jenkins resources.

Here are some of the authentication mechanisms that Jenkins supports:

Jenkins' own user database: Jenkins has its own user database, which stores usernames and passwords. You can create user accounts in Jenkins, and users can log in using their credentials.

LDAP: Jenkins supports Lightweight Directory Access Protocol (LDAP), which allows Jenkins to authenticate users against an LDAP server. This is useful in organizations where user accounts are centrally managed.

SAML: Jenkins supports Security Assertion Markup Language (SAML), which is an XML-based authentication and authorization protocol. SAML allows single sign-on (SSO) authentication, where users can log in once and access multiple applications.

OAuth: Jenkins also supports OAuth, an authentication protocol that allows users to grant third-party applications access to their resources without sharing their credentials.

To configure authentication in Jenkins, you need to navigate to the "Configure Global Security" page, where you can select the desired authentication mechanism and configure the settings accordingly.

LDAP

1. Install the LDAP plugin in Jenkins. Go to "Manage Jenkins" -> "Manage Plugins" -> "Available" and search for "LDAP Plugin". Install the plugin and restart Jenkins.
2. Go to "Manage Jenkins" -> "Configure Global Security". Under the "Security Realm" section, select "LDAP" as the authentication method.
3. Configure the LDAP server details. Here's an example of what the settings might look like:
    - Server: ldap://myldapserver.com
    - Root DN: dc=mydomain,dc=com
    - User search base: ou=users,dc=mydomain,dc=com
    - User search filter: (uid={0})

    In this example, we're connecting to an LDAP server at "myldapserver.com" and using the domain "mydomain.com". The root DN is the base DN for our LDAP directory, and the user search base is the location in the directory where user accounts are stored. The user search filter specifies how to search for user accounts. The "{0}" will be replaced with the username when a user logs in.
4. Test the LDAP connection. Click the "Test LDAP settings" button to verify that Jenkins can connect to the LDAP server and retrieve user information.
5. Save the configuration and test authentication. Once the LDAP settings are saved, try logging in to Jenkins with an LDAP user account. If everything is configured correctly, Jenkins should authenticate the user and allow access to the Jenkins resources.

SAML

Here's an example of how to configure Jenkins to use SAML for authentication:

1. Install the SAML plugin in Jenkins. Go to "Manage Jenkins" -> "Manage Plugins" -> "Available" and search for "SAML Plugin". Install the plugin and restart Jenkins.
2. Configure the SAML Identity Provider (IdP). You'll need to obtain the metadata for your IdP, which typically includes information such as the IdP entity ID, the Single Sign-On (SSO) URL, and the public key used for signing SAML messages.

3. Go to "Manage Jenkins" -> "Configure Global Security". Under the "Security Realm" section, select "SAML 2.0" as the authentication method.
4. Configure the SAML settings. Here's an example of what the settings might look like:
    - Identity Provider Metadata: paste the metadata for your IdP
    - Display Name Attribute: specify the SAML attribute that should be used as the user's display name
    - Email Address Attribute: specify the SAML attribute that should be used as the user's email address
5. Test the SAML connection. Click the "Test SAML settings" button to verify that Jenkins can connect to the IdP and retrieve user information.
6. Save the configuration and test authentication. Once the SAML settings are saved, try logging in to Jenkins using SSO with a user account from the IdP. If everything is configured correctly, Jenkins should authenticate the user and allow access to the Jenkins resources.

OAuth

Here's an example of how to configure Jenkins to use OAuth for authentication:

1. Install the OAuth plugin in Jenkins. Go to "Manage Jenkins" -> "Manage Plugins" -> "Available" and search for "OAuth Plugin". Install the plugin and restart Jenkins.
2. Register Jenkins as a client with the OAuth provider. This typically involves creating an OAuth application in the provider's developer console and obtaining a client ID and secret.
3. Go to "Manage Jenkins" -> "Configure Global Security". Under the "Security Realm" section, select "OAuth" as the authentication method.
4. Configure the OAuth settings. Here's an example of what the settings might look like:
    - Authorization URL: the URL for the OAuth provider's authorization endpoint
    - Token URL: the URL for the OAuth provider's token endpoint
    - User Info URL: the URL for the OAuth provider's user info endpoint
    - Client ID: the ID of the OAuth client that Jenkins registered with the provider
    - Client Secret: the secret key of the OAuth client
    - Scopes: a comma-separated list of OAuth scopes that Jenkins should request from the provider

5. Test the OAuth connection. Click the "Test OAuth settings" button to verify that Jenkins can connect to the OAuth provider and retrieve user information.
6. Save the configuration and test authentication. Once the OAuth settings are saved, try logging in to Jenkins with an OAuth user account. If everything is configured correctly, Jenkins should authenticate the user and allow access to the Jenkins resources.