Installing an OAuth authentication server offline involves several steps, including setting up a server environment, configuring the OAuth server software, and ensuring all necessary dependencies are in place without internet access. Below is a step-by-step guide to achieve this:

**Step 1: Prepare the Environment**

1. **Select a Machine:**
   * Choose a server or a virtual machine that will host your OAuth server. Ensure it has the required specifications for running your OAuth server software.
2. **Install the Operating System:**
   * If not already done, install an operating system like Ubuntu, CentOS, or any other preferred OS.

**Step 2: Download Necessary Software and Dependencies**

1. **Choose an OAuth Server Software:**
   * Popular options include Keycloak, IdentityServer, or OAuth2 Proxy. For this example, let's use Keycloak.
2. **Download Keycloak:**
   * Download the Keycloak server from a machine with internet access. You can get the archive from the official Keycloak downloads page.
3. **Download JDK (Java Development Kit):**
   * Keycloak requires a JDK to run. Download the appropriate version of the JDK from [AdoptOpenJDK](https://adoptopenjdk.net/) or Oracle's website.
4. **Download Database Software:**
   * Keycloak supports various databases like PostgreSQL, MySQL, etc. Download the necessary database software or JDBC driver if required.
5. **Download Required Dependencies:**
   * If Keycloak has other dependencies (like specific libraries or tools), download these as well. Check the documentation for any additional requirements.
6. **Transfer Files to the Offline Server:**
   * Use a USB drive or any other method to transfer the downloaded files to the server that will host the OAuth server.

**Step 3: Install and Configure the Software**

1. **Install JDK:**
   * Transfer the JDK installation file to the offline server and install it. For example:

bash

Copy code

sudo mkdir -p /usr/local/java

sudo tar -xzf jdk-xx\_linux-x64\_bin.tar.gz -C /usr/local/java

1. **Set Up Environment Variables:**
   * Set JAVA\_HOME and update the PATH variable.

bash

Copy code

export JAVA\_HOME=/usr/local/java/jdk-xx

export PATH=$JAVA\_HOME/bin:$PATH

1. **Install the Database:**
   * Install the database software. For PostgreSQL:

bash

Copy code

sudo rpm -ivh postgresql-xx.rpm

sudo systemctl enable postgresql

sudo systemctl start postgresql

1. **Configure the Database:**
   * Set up the database for Keycloak. Create a user and a database:

sql

Copy code

CREATE USER keycloak WITH PASSWORD 'password';

CREATE DATABASE keycloakdb;

GRANT ALL PRIVILEGES ON DATABASE keycloakdb TO keycloak;

1. **Install Keycloak:**
   * Extract the Keycloak server:

bash

Copy code

tar -xzf keycloak-xx.tar.gz -C /opt

cd /opt/keycloak-xx

1. **Configure Keycloak:**
   * Update the Keycloak configuration files to point to your database. Edit standalone/configuration/standalone.xml or use environment variables:

xml

Copy code

<datasource>

<connection-url>jdbc:postgresql://localhost:5432/keycloakdb</connection-url>

<driver>postgresql</driver>

<username>keycloak</username>

<password>password</password>

</datasource>

**Step 4: Start and Test the OAuth Server**

1. **Start Keycloak:**
   * Run the Keycloak server:

bash

Copy code

cd /opt/keycloak-xx/bin

./standalone.sh

1. **Create Admin User:**
   * Open a terminal and run:

bash

Copy code

./add-user-keycloak.sh -r master -u admin -p password

1. **Access Keycloak:**
   * Open a web browser on a machine that can access the server and navigate to http://<your-server-ip>:8080/auth.
   * Log in with the admin credentials you created.
2. **Configure OAuth Clients and Users:**
   * Use the Keycloak admin console to configure OAuth clients, users, and roles as needed.

**Step 5: Secure and Maintain the Server**

1. **Secure the Server:**
   * Configure SSL/TLS for secure communication.
   * Set up firewalls and security groups to restrict access.
2. **Regular Backups:**
   * Ensure regular backups of the database and Keycloak configuration.
3. **Monitor and Update:**
   * Regularly monitor the server's health and apply offline updates as necessary.

By following these steps, you should be able to set up an OAuth authentication server in an offline environment.