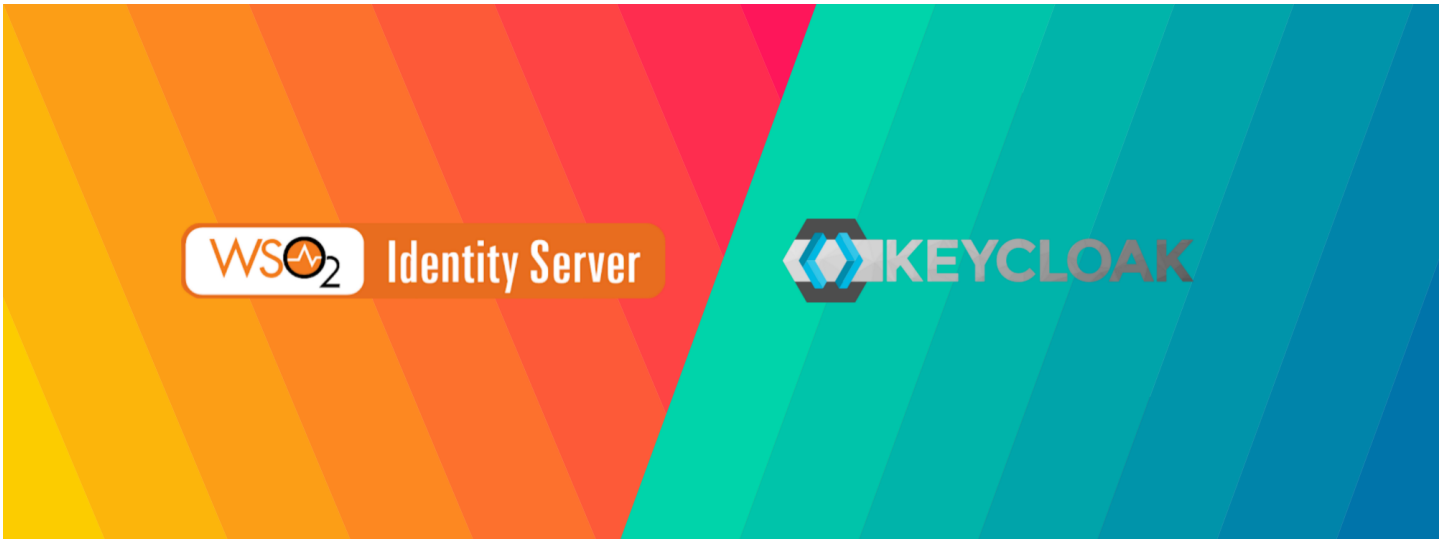
Chamath

70 Followers About Follow

WSO2 Identity Server vs. Keycloak : A Comparison of the Two Leading Open Source Identity Servers

Chamath Jul 24, 2020 · 9 min read



If you are looking for an open-source identity provider for your organization, two of the main candidates that exist in the market right now are **WSO2 Identity Server** and **Keycloak**.

Both of these are widely adopted enterprise grade identity management solutions that offer the users with great flexibility and a wide range of features. In this piece, we will see how these two open-source Identity management offerings stack against one another.

If we consider WSO2 Identity Server, it is developed by WSO2 Inc. and it is distributed under the terms of Apache License 2.0 which is pretty nonrestrictive and allows for commercial use. WSO2 Identity Server was first released in 2008 and it is written in Java. And it runs on WSO2 Carbon middle-ware which also enables many other products of WSO2 Inc.

As per Keycloak, it is currently being developed by JBoss which is a division of Red Hat. It had its first release in 2014. Similar to WSO2 Identity Server, Keycloak is also distributed under Apache License 2.0. It is written predominantly in Java and runs on WildFly middle-ware.

Commercial Support

Disclaimer: The pricing details were last updated on 24/07/2020 from the vendor websites and may be different from actual. Please confirm with the vendor website before purchasing.

If you are planning to use this software for a business, then you should definitely consider the commercial support option.

WSO2 Identity Server support subscription fees are based on the region. For North America, subscriptions start at **$25,400 per year for 2 cores** and the prices are scaled based on the number of cores you use. A subscription gives you 24x7x365 incident support, reactive maintenance for any incidents affecting the production or pre-production systems. They offer response and resolution times based on the severity of the issue. A subscription includes a level of query support proportional to the subscription fees; 1 hour for each $1000 of subscription fees. With each subscription, the first 3 pre-production environments are included for free. With a commercial subscription, you get patches for their releases which you don't get with the community version. You can find the latest pricing details for your region and for the type of your deployment here.

RH SSO, you need to have a contract for Red Hat JBoss® Enterprise Application Platform which starts at **$8000 for 16 cores per year**. Red Hat supports migrating from the Keycloak community version to the RH SSO in case you ever wish to get a subscription later. There are different support categories that you can pay for, which provide different support levels. For an example, there is quick response if you pay a higher amount, where the response time is within a few hours. Those come under the add-on section for Red Hat products.

Setting Up

To set up WSO2 Identity Server, you need to download the binary distribution from the WSO2 website and extract it. Once downloaded and extracted, you need to run the following command in the terminal.

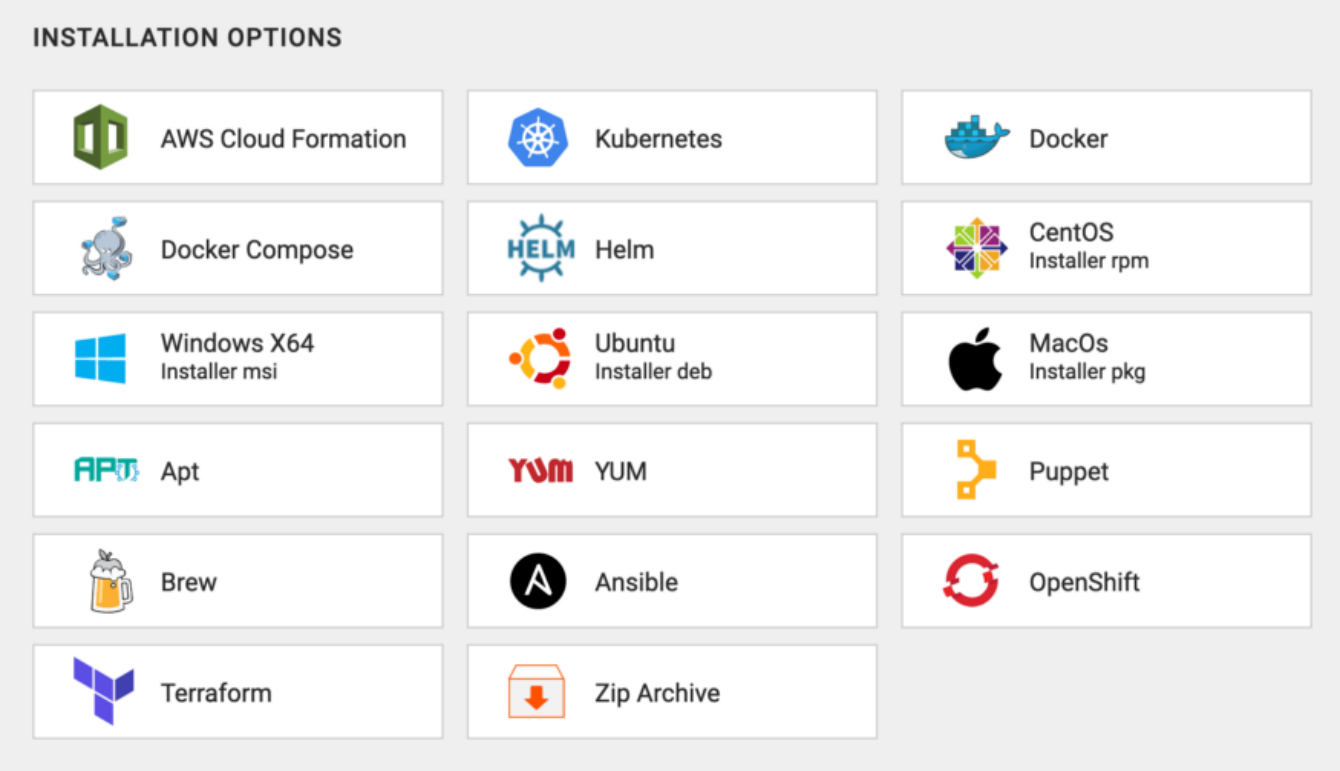
- On Windows: `<IS_HOME>/bin/wso2server.bat --run`
- On Linux/Mac OS: `sh <IS_HOME>/bin/wso2server.sh`

Also, you can setup the community version of the WSO2 Identity Server via its public docker registry by running the following command in the terminal.

```
docker run -it -p 9443:9443 --name is wso2/wso2is:5.7.0
```

However, with a commercial account, you have the option of using the WSO2 private docker registry which has updates and patches to their releases. To set up WSO2 Identity Server docker instance, run the following commands in the terminal.

```
docker login docker.wso2.com
docker pull docker.wso2.com/wso2is:<version>
docker run -d docker.wso2.com/<pulled image>
```



Install options for WSO2 Identity Server

With Keycloak, you can setup a server just by running,

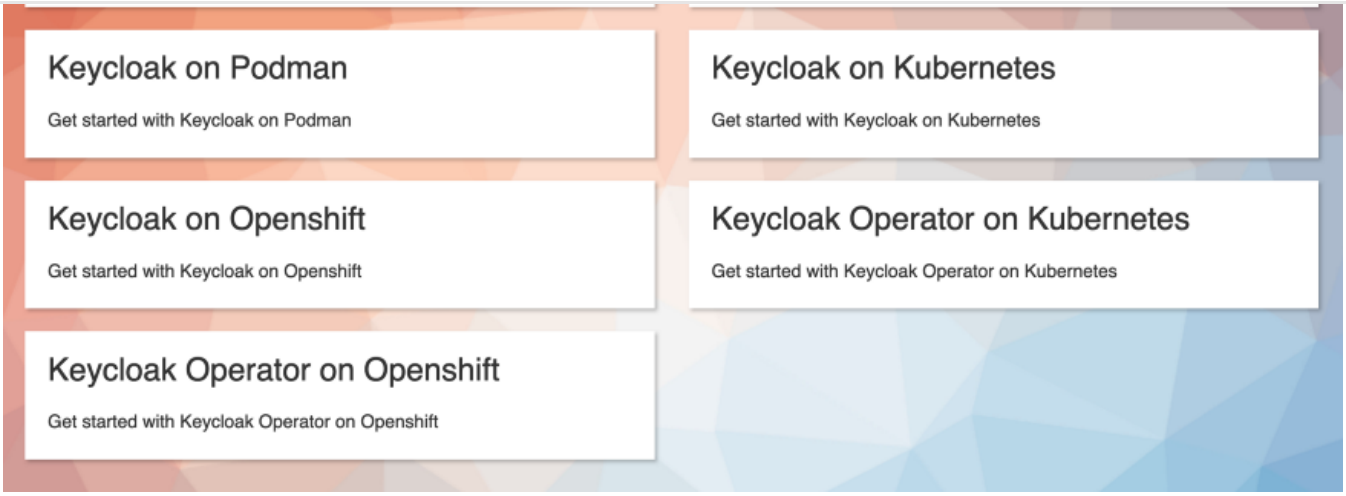
```
docker run jboss/keycloak
```

If you want to get the binary files and start the server locally, you can download the binary distribution from the Keycloak website. After extracting you should have a directory named keycloak-<version>. To start the Keycloak server, open the directory keycloak-<version>, and run the following command in a terminal.

- On Windows: `bin/standalone.bat`
- On Linux/Mac OS: `sh bin/standalone.sh`

Other than that, you can set up a Keycloak server on Kubernetes, Openshift or on Podman if you wish to do so. More on that here.

Getting Started with Keycloak



Install options for Keycloak

Functionality

Getting to the meat of the comparison, I will try to compare the functionalities that each of the products offer on a few areas which I feel is most significant for a user looking to adopt one of these. For the comparison, following product distributions were used.

WSO2 Identity Server: 5.10.0

Keycloak: 10.0.2

User Stores

A user store is one of the fundamental components in any Identity Management solution. It allows you to persist users and roles. Both WSO2 Identity Server and Keycloak has been configured to use the embedded H2 database out of the box. But they both discourage to use that in production.

Keycloak offers only one persistence option in a single data source that is a JDBC data source. Out of the box, they have support for connecting external LDAP and Active Directory and Kerberos servers. In operation, what happens is that the authenticated users' data is mapped to a common user model in the local Keycloak user database. So, you don't have the capacity to actively manage external user stores.

even another WSO2 Identity Server instance for persistence. So, WSO2 Identity Server allows you to configure multiple data sources and multiple user stores (domains) whereas in Keycloak, you are limited to a single data source and a single user store.

About databases, both these products support many popular databases like MySQL, PostgreSQL, Oracle, etc.

Users and Roles

Users and roles is a fundamental concept to Identity Management and it is supported by both of them. Apart from users and roles, Keycloak also has the notion of "groups" which WSO2 Identity Server does not have. What this additional functionality "groups" adds is that, it allows you to assign attributes to multiple users at once.

Single-Sign-On

Single-Sign-On is probably one of the main reasons why someone would opt to use an identity server solution because it allows a user to authenticate only once and get access to multiple applications. The commercial version of Keycloak in fact goes by the name RedHat Single-Sign-On. So, it goes without saying, SSO functionality is supported in both of these solutions.

Both WSO2 Identity Server and Keycloak support the two most common SSO protocols, SAML 2.0 and OpenID Connect. While SSO is very well supported by both of these products, there are slight terminology differences between them. In WSO2 Identity Server, they call their applications as "service providers", whereas in Keycloak they are called as "clients". With WSO2 Identity Server, you also get a "resident service provider" which can be used for inbound and outbound identity provisioning. More about provisioning later.

Attribute Mapping

Attribute mapping is something that you need if you have diverse applications because different applications call the same attribute in different ways. Like for an example, "email" and "email address", or "dob" and "date of birth". Because of that, you might want to map user attributes to different entities.

In WSO2 Identity Server, these attributes are mapped through "claims". For that, it supports standard and custom claim dialects. In Keycloak, they call these "custom attributes" and they also support standard and custom scopes for attributes. Also, it is

Identity Federation

Identity federation is about relying on another identity provider for authenticating your users. This enables things like social log in, where users can log in to applications via Facebook or Twitter or from some other platform.

While both of these products do support configuring external identity providers, WSO2 Identity Server allows you to do that in a more flexible way. With WSO2 Identity Server, you can configure external identity providers per-application whereas in Keycloak, per-application IdP binding is not supported.

Identity Provisioning

In simple terms, identity provisioning means creating users on-the-fly as they are authenticated. And it comes in two variations; inbound provisioning and outbound provisioning. Inbound provisioning means you create users locally while they are authenticated externally. Outbound provisioning means you create users elsewhere while they are authenticated locally. Keycloak only supports the first variation; inbound user provisioning. With WSO2 Identity Server, you get both inbound and outbound user provisioning capabilities. Also, per-application user provisioning is another feature that is supported only by WSO2 Identity Server.

Another important factor to note here is that WSO2 Identity Server supports the SCIM (System for Cross-domain Identity Management) protocol. Unfortunately, neither per-application user provisioning nor the SCIM protocol for provisioning is supported by Keycloak currently.

Multi Tenancy

Multi tenancy is a way of creating virtual identity servers within a single server instance. The main reason why you want to have such a setup is cheaper implementation. Both these servers do support this mechanism. But there is a slight difference in terminology. WSO2 Identity Server calls them “tenants” and Keycloak calls them “realms”.

Keycloak lets you manage tenant easily when compared to WSO2 Identity Server because in Keycloak, Superuser can manage all the realms. But in WSO2 Identity Server, you need to login to every tenant separately as tenant admin and make changes

One-Time Passwords

“One-Time password” is a well known security enhancement which you can implement. It is often a part of multi-step authentication flows. When comparing the two servers, both of them support Time-based OTPs (TOTP) and Google Authenticator(QR code) which is a standard time based code implementation. Keycloak also has support for Counter-based OTP (HOTP) which WSO2 Identity Server does not support. But on the other hand, WSO2 Identity Server allows you to generate OTPs and send them over on e-mail or SMS, which Keycloak does not support out of the box.

Multi Step Authentication

Multi step authentication is used mainly for two purposes. It can be used for enhancing security just like the one-time passwords or it can be used to impose specific actions on users such as password updates after logging in. The support for multi step authentication on Keycloak is a bit limited. The only security enhancement they offer as of now is one-time passwords and the rest are some pre-defined actions which you can mandate the user to execute. And the multi step authentication policy is configurable only per realm with Keycloak.

With WSO2 Identity Server, you can configure the multi step authentication policy as you wish. The authentication policy configurations are extremely flexible. You can add as many steps as you want. You can add multiple authenticators per step. And you can configure authentication policies per application which is a huge benefit in terms of customization which you don’t get with Keycloak.

An important feature that’s worth mentioning is the script based authentication flow configuration which WSO2 Identity Server offers. Basically, this allows you to script the authentication flow. You can make use of the different authentication script templates to cater to your needs. Or you can use them as a reference and write your own custom authentication script.

Verdict

functionality it offers is rigid and limited. On the other hand, with WSO2 Identity Server, the UI feels a bit outdated and convoluted. It requires a significant amount of knowledge around the product to do things properly. And the extended configurations can get overwhelming for a first time user. Also, it is pricey in terms of commercial support. But I believe it can offer anything in terms of functionality when compared to Keycloak which has a limited and a rigid set of functionalities.

Ultimately, it boils down to what you require out of an identity server. I would recommend using the WSO2 Identity Server if your application landscape is very diverse and complex, and require enterprise level customizations. On the other hand, Keycloak would be better suited if you are a non-profit organization or a university where you don't want to invest a lot of money, and want a very basic set of functionalities with customer support.

Identity Management Wso2 Identity Server Wso2is Wso2 Keycloak