

A stylized landscape illustration featuring a dark green background. In the foreground, there are several layers of jagged, mountain-like shapes in shades of blue and grey, creating a sense of depth. On the left side, a multi-colored rainbow (red, orange, yellow, red) is partially visible, appearing to rise behind the mountains. The overall aesthetic is clean and modern.

# What's New in WildFly and Upcoming Roadmap

Brian Stansberry | [brian.stansberry@redhat.com](mailto:brian.stansberry@redhat.com)



About me



I'm the WildFly project lead

and the Principal Architect for Red Hat JBoss EAP

I've been involved with JBoss AS /WildFly/EAP for over 20 years



# A brief history lesson

# What's New...ish



- 2019 -- provisioning a slim server [using Galleon layers](#)
- 2020 -- MicroProfile Platform support
- 2021
  - WildFly 22 -- first WildFly Preview release, Jakarta EE 9
  - WildFly 25 -- legacy Picketbox security support removed
- 2022
  - Ansible Collection for WildFly
  - New cloud architecture
  - WildFly 27 and Jakarta EE 10



# What's New



## What's new in the main server

- WildFly 28 -- Observability -- new Micrometer and MicroProfile Telemetry subsystems
- WildFly 29 -- Configuration customization using YAML
  - Start with a shared base config and customize it in an idempotent way
  - `./standalone.sh -c standalone-full.xml -y=jaas-realm.yml:demo-app-queues.yml`
- WildFly 30 -- Preliminary SE 21 support (more on this later)
- WildFly 31
  - Jakarta MVC in WildFly Preview
  - Hibernate 6.4 and Hibernate Search 7.0
  - Updates to support the MicroProfile 6.1 spec versions

# What's new beyond the main server



## Galleon ecosystem

- Jakarta EE 10 Core Profile with the `ee-core-profile-server` Galleon layer
- New feature packs to let you add onto your standard WildFly:
  - Keycloak SAML Adapter
  - MyFaces 4
  - EE8- $\rightarrow$ 9 transformation
- WildFly Glow beta
  - Be sure to stay for Jean-Francois Denise's talk on this!

## WildFly Maven Plugin

- Dev mode

# Education and Community Outreach



Best practices for using WildFly have changed *a lot* over the last 5 years.

Helping our users understand the best way to use WildFly has been a big focus in recent months.

- You heard earlier from Harald Pehl about <https://wildfly.org/get-started/>
- WildFly Mini-Conference!
- and...



# Quickstart Improvements



Quickstarts now demonstrate provisioning a WildFly server or bootable jar with the quickstart deployment included.

Wherever possible the quickstart is able to run on OpenShift, using a Helm chart to deploy.

Quickstarts include basic smoke and quickstart-specific tests, with the tests able to run against:

- an externally managed WildFly server
- the server or bootable jar provisioned by the quickstart
- the quickstart application + server deployed on OpenShift

# User guides on wildfly.org



Keep an eye on the new <https://wildfly.org/guides> page!

The focus here is on step-by-step instructions for accomplishing particular use cases, particularly operational ones. For example:

- [Configuring Logging for your Application](#)
- [Securing WildFly Apps with OIDC on OpenShift](#)
- [Deploying WildFly using Ansible](#)

Continuing to add more guides is an important priority for the WildFly developer community.

Please try them out and give us your suggestions for improvements and for other guides!



# What's Coming

# Quarterly Release Cadence



WildFly will continue its practice of producing feature releases roughly every three months. WildFly releases will be loosely time boxed.

- WildFly 32 - April 2024
- WildFly 33 - July 2024
- WildFly 34 - October 2024
- WildFly 35 - January 2025

We will produce one bug-fixing micro release roughly one month after each feature release.

For each feature release we produce a beta. So that's 12 releases a year.

# Jakarta EE 11



Jakarta is aiming to have EE 11 out in June or July. (*JMHO: June seems like a real stretch.*)

Some highlights:

- New spec: Jakarta Data
  - Split persistence from the model with the Repository pattern.
- Virtual thread support in Jakarta Concurrency (SE 21 only).
  - EE 11 requires SE 17 as a baseline but supports SE 21
- Quite a lot of enhancements to Jakarta Persistence
  - including Java Record support

EE 11 **removes support** for running with a SecurityManager.

- Java SE has deprecated but still supports the SM, but deprecated SM APIs are being removed from EE 11 APIs, and many APIs and impls will remove internal SM calls.
- *If you try and run an EE 11 server with the SM enabled, you'll have problems. Don't do it.*

# WildFly Support for Jakarta EE 11



We'll initially bring EE 11 into WildFly by incorporating EE 11 specs into WildFly Preview.

- WildFly Preview 32 -- a mix of EE 11 and EE 10 specs.
- WildFly Preview 33 -- more EE 11 specs
- WildFly Preview 34 -- if necessary even more EE 11 specs

The timing of the EE 11 GA release *might* align with the WildFly Preview 33 release but it's too early to say if WildFly Preview 33 will be EE 11 compatible. Probably not.

# WildFly Support for Jakarta EE 11



We're not planning on having EE 11 support in standard WildFly prior to WildFly 34.

We are exploring supporting both EE 10 and EE 11 in standard WildFly for some period, and we aim to have this in place before introducing EE 11 in standard WildFly.

Perhaps not until WildFly 35.

## 'Dual Support' -- EE 10 and EE 11



We're evaluating producing parallel feature packs for standard WildFly, one with EE 10 APIs and the other with EE 11.

User story:

*"I'm an architect who emphasizes keeping up with WildFly releases in order to get CVE and general bug fixes, but I need a longer time to adapt to Jakarta EE changes."*

The EE 10 variant of standard WildFly would provide that "longer time".



# 'Dual Support' Caveats



We can only continue releases of the 'older' (i.e. EE 10) variant as long as all the components that are part of it have compatible releases that are acceptable.

- No CVEs. (Or perhaps none with a severity score greater than some TBD #)
- No other bugs with critical impact on WildFly.

We integrate hundreds of libraries, and many of those only maintain their 'main' branch. They might not produce bug-fix releases for their EE 10 variants.

Think of dual support as buying you an extra quarter or two or three to transition to EE 11.

# MicroProfile



MicroProfile is planning to release MicroProfile 7.0 in June.

- Updates to Rest Client, JWT, Fault Tolerance, Telemetry, OpenAPI
- Removal of MicroProfile Metrics (which WildFly doesn't support)
  - MicroProfile Telemetry will add support for OpenTelemetry Metrics and Logging
- Based on EE 10, but it *may* work on EE 11 as well, when that comes out.

We *might* provide support for this in WildFly 33, perhaps with a few weeks delay in the WildFly 33 release date. We'll see; it's too early to say.

- We've successfully navigated this kind of tight timing in the past.
- If not, we'll target WildFly 34 (October).

# Java SE and WildFly



When we talk about which SE releases are 'supported' for WildFly, we classify releases in four different categories:

- *Qualified* -- LTS releases where we've fully qualified the SE version as compatible. **Currently SE 11 and SE 17.**
- *Recommended* -- The latest '*Qualified*' release. **Currently SE 17.**
- *Qualifying* -- LTS releases that we're actively qualifying. **Currently SE 21.**
- *Non-LTS* -- where we've done enough basic testing to be able to say it's useful for users to experiment with WildFly on it. **Currently None.**

# Java SE 21



Java SE 21 is currently in the '*Qualifying*' category. We'll see if we can complete the qualification for the WildFly 32 release and make SE 21 the '*Recommended*' version.

If not we should get there for WildFly 33.

Criteria to move to '*Recommended*' are:

- **Done:** Pass the WildFly and WildFly Core testsuites.
  - Main testsuite, Galleon layer tests, bootable jar tests, SecurityManager enabled, Windows, etc.
- **In Progress:** Pass the relevant Jakarta EE and MP TCKs.
  - Failing tests that cannot be passed because the test itself doesn't work right on SE 21 are ok.
- **In Progress:** Review key component support for SE 21.
  - Do they test, any known problems etc.

# Java SE 11



An open topic among the WildFly developers is how long we can continue to support Java SE 11.

The broader ecosystem we rely upon is starting to use SE 17 as the baseline.

- EE 11 will require SE 17

WildFly Preview 32 will require SE 17.

*Perhaps* the 'dual support' EE 10 / EE 11 approach will allow us to support SE 11 for a bit longer in the EE 10 variant.

**WildFly users should be actively working to move off of SE 11; it's only a matter of time until we cannot provide SE 11 support.**

# Stability Levels



Started in WildFly 31 and expanding in a major way in WildFly 32 and 33 is the concept of WildFly features having different levels of 'stability'.

Let's look at some user stories related to this:

*"I'm a WildFly developer and I want to make a tech preview variant of my feature available in standard WildFly so I can get feedback from real users."*

*"I run WildFly servers and I want to investigate tech preview features but I don't want to use them by accident."*

*"I set policy for how WildFly is used in my organization and I want to ensure tech preview modules are not present in our production WildFly installations."*

# Stability Levels



Features can have one of four stability levels, with the WildFly developers demanding more from feature teams depending on the stability level. Features can be 'promoted' over time to higher stability levels.

- *Experimental*: True bleeding edge stuff. Real likelihood of future incompatible changes, including being dropped altogether.
- *Preview*: Typical 'tech preview' notion. Lower likelihood of future incompatible changes. Definite intent (but no promise) to eventually move to *Community* level.
- *Community*: Going forward this will be the typical level for new standard WildFly features. Expectation for quality and long term compatibility matches historical WildFly norms.
- *Default*: Features that have undergone additional validation, typically involving engineers with a quality engineering focus. Difference from *Community* is only about validation, not expectations.

All but one feature in WildFly 31 is at '*Default*' level, but going forward we plan to add many more features at other levels.

# Controlling Stability Levels in the Runtime



Use the '--stability' command line argument to specify the desired stability level:

```
bin/standalone.sh --stability=preview
```

Values are 'experimental', 'preview', 'community', 'default'

Features at a lower stability level are not visible and can't be used.

If you don't set this, the OOTB setting for the WildFly installation is used.

- For standard WildFly that's 'community'.
- For WildFly Preview that's 'preview'.



# Limiting Stability Levels when Provisioning



In WF 32+, you'll be able to tell Galleon not to provision config elements and JBoss Modules modules that have a stability level below your desired minimum.

```
<plugin>
  <groupId>org.wildfly.plugins</groupId>
  <artifactId>wildfly-maven-plugin</artifactId>
  <configuration>
    [...]
    <galleon-options>
      [...]
      <minimum-stability>default</minimum-stability>
    </galleon-options>
    [...]
  </plugin>
```

[wildfly.org](http://wildfly.org)



@WildFlyAS

@wildflyas@fosstodon.org