



Eclipse Che on K8S

Manage projects and distributed teams with open source in-browser IDE

\$ whoami
Natale Vinto
Developer Advocate at Red Hat

Agenda

- What is Eclipse Che
- Architecture
- Integrations with K8S
- Runtimes and Workspaces
- Factory
- Demo



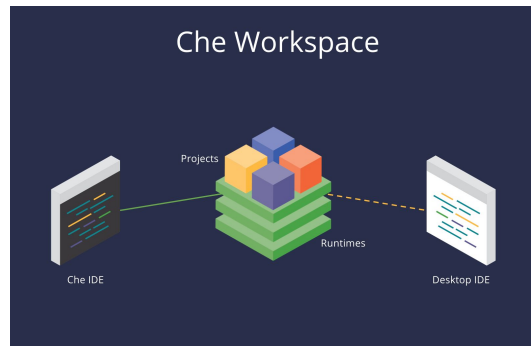
Eclipse Che

The Kubernetes-Native IDE for Developer Teams



Eclipse Che

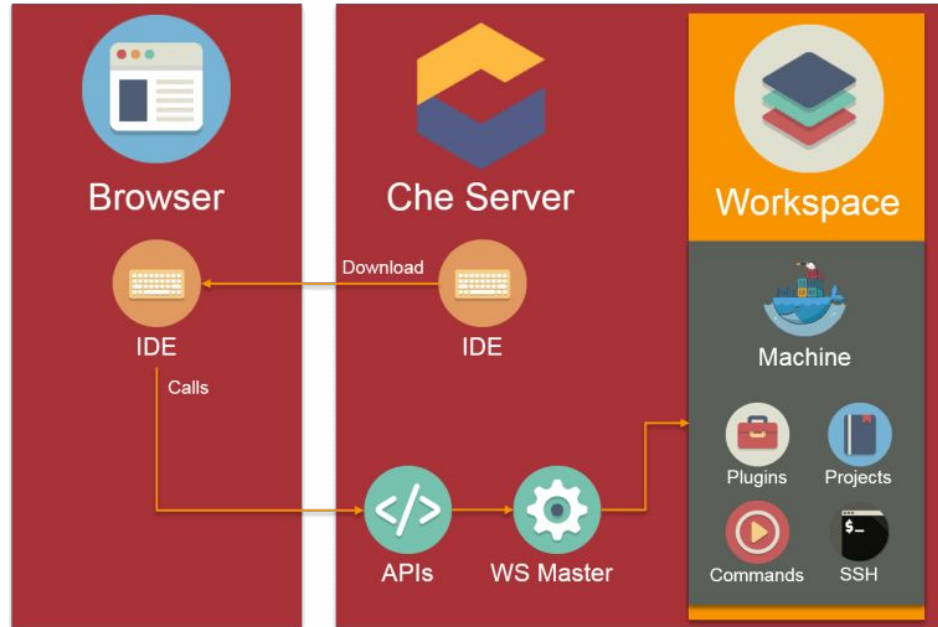
- In-Browser IDE
- Containerized extendible Workspaces
- Support most used Runtimes and Framework (and more!)
- Support Eclipse Theia for Visual Studio Code look'n'feel
- Support for Multi-tenancy
- **Open Source**



Eclipse Che



Eclipse Che Architecture



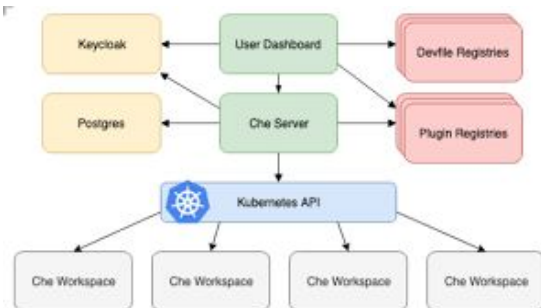
Integrations with K8S

- Helm charts
 - You can use **chectl** to install on any k8s
- Operator
 - Available from OperatorHub.io
- Minikube addon
 - minikube addons enable olm



Integrations with K8S

- Persistence (PVC)
- Multi-tenancy (Keycloak SSO)
- Custom Resource



Runtimes and Workspaces

The screenshot displays the Eclipse Che web interface. On the left is a dark sidebar with navigation links: Workspaces (6), Get Started (active), Stacks, Factories, Administration, and Organizations. Below these are 'RECENT WORKSPACES' with links to create a workspace and several workspace IDs. The main content area is titled 'Getting Started with Eclipse Che' and has a sub-header 'Get Started' with a link to 'Custom Workspace'. A 'Select a Sample' section prompts the user to 'Select a sample to create your first workspace.' with a 'Filter by' input field. A grid of 20 workspace templates is shown, each with an icon, title, and description. The templates include: NodeJS Angular Web Application, Apache Camel K, Apache Camel based on Spring Boot, Mainframe Basic Stack, C/C++, .NET Core, ASP.NET Core Web Application, Go, Java Gradle, Java Maven, Java with Spring Boot and MongoDB, Java with Spring Boot and MySQL, Java Spring Boot, Java Vert.x, NodeJS Express Web Application, NodeJS MongoDB Web Application, NodeJS React Web Application, NodeJS Web Application based on Yarn, PHP Laravel with MySQL, PHP with MySQL, PHP Symfony, PHP Simple, Python, and Python Django. At the bottom right, there are buttons for 'Make a wish', 'Docs', and 'Community'. The status bar at the very bottom shows 'Eclipse Che - 7.13.2' and the user 'Natalie Vintro'.

Eclipse Che

Workspaces (6)

Get Started

Stacks

Factories

Administration

Organizations

RECENT WORKSPACES

- Create Workspace
- react-che-example65d2
- react-gitter
- frontend42oo7
- webpack nodejs react4d0
- nodejs-7aige

Getting Started with Eclipse Che


Get Started Custom Workspace


Select a Sample


Select a sample to create your first workspace.


Filter by


Temporary Storage 26 items


**NodeJS Angular Web Application**
Stack for developing NodeJS Angular Web Application


**Apache Camel K**
Stack with tooling ready to develop integration projects with Apache Camel K


**Apache Camel based on Spring Boot**
Stack with environment ready to develop integration projects with Apache Camel based on Spring Boot.


**Mainframe Basic Stack**
Check! Mainframe Basic Stack is an all-in-one extension pack for developers working with z/OS applications, suitable for all levels of mainframe experience, even beginners.


**C/C++**
Stack with C/C++ and Clang 8


**.NET Core**
Stack with .Net 2.2


**ASP.NET Core Web Application**
Stack for developing ASP.NET Core Web Application


**Go**
Stack with Go 1.12.10


**Java Gradle**
Java Stack with OpenJDK 11 and Gradle 6.2.1


**Java Maven**
Java Stack with OpenJDK 11 and Maven 3.6.0


**Java with Spring Boot and MongoDB**
Java stack with OpenJDK 8, MongoDB and Spring Boot Guestbook demo application


**Java with Spring Boot and MySQL**
Java stack with OpenJDK 8, MySQL and Spring Boot Petclinic demo application


**Java Spring Boot**
Java stack with OpenJDK 8 and Spring Boot Petclinic demo application


**Java Vert.x**
Java stack with OpenJDK 8 and Vert.x demo application


**NodeJS Express Web Application**
Stack with NodeJS 10


**NodeJS MongoDB Web Application**
Stack with NodeJS 10 and MongoDB 3.4


**NodeJS React Web Application**
Stack for developing NodeJS React Web Application


**NodeJS Web Application based on Yarn**
Stack for developing NodeJS Web Application based on Yarn


**PHP Laravel with MySQL**
PHP Stack with Laravel and MySQL real world application


**PHP with MySQL**
PHP Stack with MySQL and simple database application


**PHP Symfony**
PHP Stack with Symfony Demo Application <https://symfony.com/>

**PHP Simple**
PHP Stack with PHP 7.1 and simple web application

**Python**
Python Stack with Python 3.7

**Python Django**
Python Stack with Python 3.7 and Django application

**Quarkus Tools**
Quarkus Tools with OpenJDK 8 and Maven 3.6.3

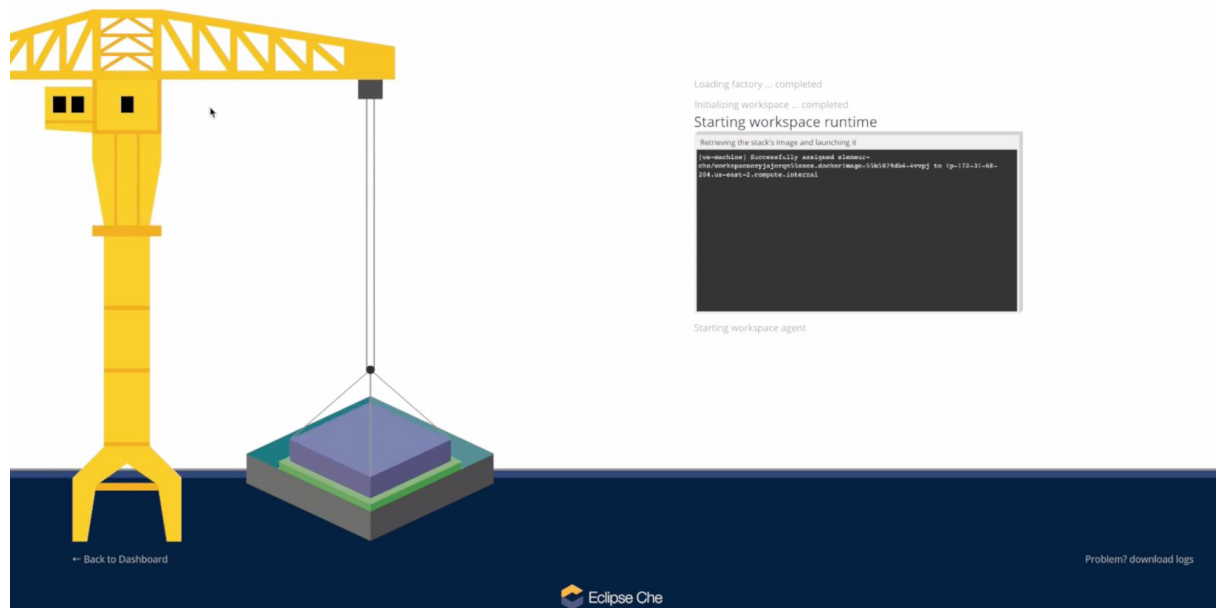
**Rust**
Rust Stack with Rust 1.39


Eclipse Che - 7.13.2

Natalie Vintro

Make a wish Docs Community

Factory & Devfile



- Create Workspaces from devfile
- Devfile is a YAML containing description of Workspace
- Easy to extend
- Fork and Click 

theia — Theia Browser Example x +

localhost:3000/workspace/theia

File Edit Selection View Go Debug Terminal Help

EXPLORER

- THEIA
- .github
- .theia
- .vscode
- configs
- dev-packages
- doc
- examples
- logo
- node_modules
- packages
- bunyan
- callhierarchy
- console
- core
- lib
- node_modules
- src
- browser
- icons
- menu
- browser-context-menu-r...
- browser-menu-module.ts
- browser-menu-plugin.ts
- messaging
- preferences
- quick-open
- shell
- source-tree
- status-bar
- style
- test
- tree
- widgets
- window

browser-menu-plugin.ts

```
41 .....@inject(MenuModelRegistry) protected readonly menuProvider: MenuModelRegistry
42 ..... { }
43 .....
44 .....createMenuBar(): MenuBarWidget {
45 .....const menuBar = new DynamicMenuBarWidget();
46 .....menuBar.id = 'theia:menuBar';
47 .....const menuModel = this.menuProvider.getMenu(MAIN_MENU_BAR);
48 .....const phosphorCommands = this.createPhosphorCommands(menuModel);
49 .....// for the main menu we want all items to be visible.
50 .....phosphorCommands.isVisible = () => true;
51 .....
52 .....for (const menu of menuModel.children) {
53 .....if (menu instanceof CompositeMenuNode) {
54 .....const menuWidget = new DynamicMenuWidget(menu, { commands: phosphorCommands }, this.co
55 .....menuBar.addMenu(menuWidget);
56 .....menuBar.act
57 .....}
58 .....}
59 .....return menuBar;
60 .....}
61 .....
62 .....createContextMenu(path: MenuPath, anchor?: Anchor): MenuWidget {
63 .....const menuModel = this.menuProvider.getMenu(path);
64 .....const phosphorCommands = this.createPhosphorCommands(menuModel, anchor);
65 .....
66 .....const contextMenu = new DynamicMenuWidget(menuModel, { commands: phosphorCommands }, this.co
```

activate (method) Widget.activate(): void

- activeIndex
- activeMenu
- openActiveMenu
- isAttached

Problems x Output Debug Console > ~/workspace/theia

- [typescript] Property 'act' does not exist on type 'DynamicMenuBarWidget'. [2339] (56, 25)
- [tslint] unused expression, expected an assignment or function call (no-unused-expression) [100000] (56, 17)
- [tslint] Missing semicolon (semicolon) [100000] (56, 28)

OUTLINE

- BrowserMainMenuFactory
- logger
- contextKeyService
- constructor
- commandRegistry
- keybindingRegistry
- menuProvider
- createMenuBar
- createContextMenu
- createPhosphorComma...
- addPhosphorCommands
- addPhosphorCommand
- DynamicMenuBarWidget
- constructor
- DynamicMenuWidget
- constructor
- menu
- options
- contextKeyService
- aboutToShow
- open
- updateSubMenus
- buildSubMenus
- BrowserMenuBarContrib...
- constructor
- factory
- onStart
- createLogo

Ln 56, Col 28 LF Spaces: 4 TypeScript 3.1.3

Demo

- Installation on Minikube
- Using and Extending React Workspace
- Test into a workspace with auto generated ingress to access app
- Push changes to Github, triggering Quay.io image build and use K8S resource for deployment of the app on Minikube cluster

<https://github.com/blues-man/eclipse-che-minikube-demo>



Thank you!

- Visit **OperatorHub.io**
- Get more about Eclipse Che and CodeReady Workspaces at **learn.openshift.com**

