Red Hat CodReady Workspaces 2

What’s New

At A Glance

Red Hat CodeReady Workspaces 2.x

a Cloud-Native IDE for OpenShift

Kubernetes with Zero Effort

Embed a CodeReady Workspaces link in a project repo or issue tracker and anyone with a browser can be contributing code in <2 minutes.

Protect your Code

Source code is never cloned to a hard-to-secure laptop. Code stays in an IT controlled sandbox, but one that reacts as quickly as a developer’s laptop.

Developer Environment as Code

Developer environments are codified with a Devfile: consistent; reproducible. Store them in the repo for auditability and to provide a GitOps experience.

Openshift Developer Perspective Integration

Workspaces are accessible from the OpenShift console, making the onboarding even smoother.

Air-Gap Install

Deploy on your OpenShift cluster, behind your firewall. Air-Gap capabilities. Easy to monitor and administrate.

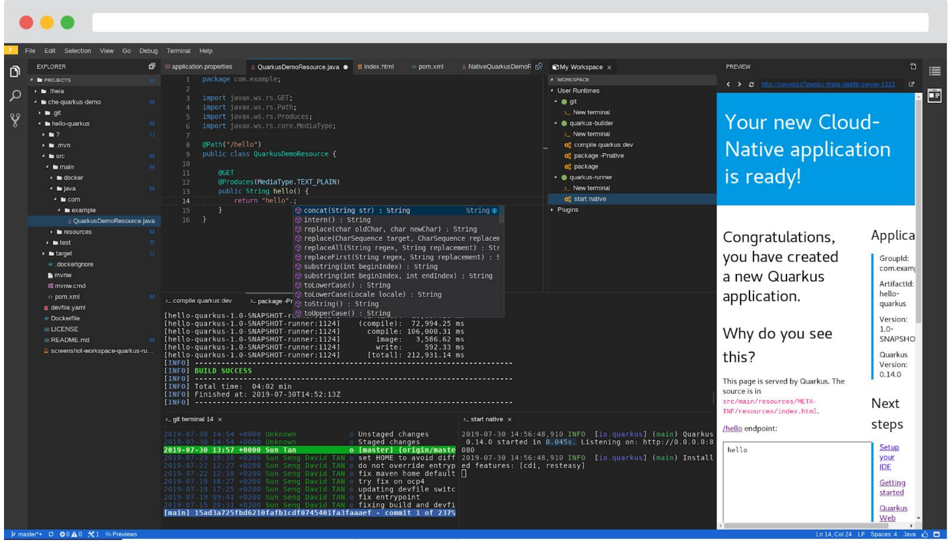
New Built-in Editor

New browser based editor, providing a fast desktop like experience. Compatibility with Visual Studio Code extensions.

CodeReady Workspaces 2.0

Based on Eclipse Che 7

- **Kubernetes-based developer workspaces:** Leverage fully containerized developer workspaces, and bring your kubernetes application runtime in your development environment.
- **New Editor**: Get a top-of-the-art desktop like experience in the browser.
- **VSCode extension compatibility:** Benefits from existing extensions
- **Devfile:** Codified definition of replicable developer environments
- **OpenShift VSCode Plug-in:** Speeds up OpenShift development
- **Easier to Monitor and Operate:** Prometheus and Grafana dashboards.



```java
package com.example;

import javax.ws.rs.GET;
import javax.ws.rs.Path;
import javax.ws.rs.Produces;
import javax.ws.rs.core.MediaType;

@Path("/hello")
public class QuarkusDemoResource {

    @GET
    @Produces(MediaType.TEXT_PLAIN)
    public String hello() {
        return "hello".;
    }
}
```

```
concat(String str) : String                       String
intern() : String
replace(char oldChar, char newChar) : String
replace(CharSequence target, CharSequence replacem
replaceAll(String regex, String replacement) : Str
replaceFirst(String regex, String replacement) : S
substring(int beginIndex) : String
substring(int beginIndex, int endIndex) : String
toLowerCase() : String
toLowerCase(Locale locale) : String
toString() : String
toUpperCase() : String
```

```
[hello-quarkus-1.0-SNAPSHOT-r
[hello-quarkus-1.0-SNAPSHOT-runner:1124]    (compile):  72,994.25 ms
[hello-quarkus-1.0-SNAPSHOT-runner:1124]      compile: 106,000.31 ms
[hello-quarkus-1.0-SNAPSHOT-runner:1124]        image:   3,586.62 ms
[hello-quarkus-1.0-SNAPSHOT-runner:1124]        write:     592.33 ms
[hello-quarkus-1.0-SNAPSHOT-runner:1124]      [total]: 212,931.14 ms
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  04:02 min
[INFO] Finished at: 2019-07-30T14:52:13Z
[INFO] ------------------------------------------------------------------------
```

```
2019-07-30 14:54 +0000 Unknown                o Unstaged changes
2019-07-30 14:54 +0000 Unknown                o Staged changes
2019-07-30 13:57 +0000 Sun Tan                o [master] {origin/maste
2019-07-23 19:10 +0200 Sun Seng David TAN o set HOME to avoid diff
2019-07-22 12:27 +0200 Sun Seng David TAN o do not override entryp
2019-07-22 12:18 +0200 Sun Seng David TAN o fix maven home default
2019-07-19 18:27 +0200 Sun Seng David TAN o try fix on ocp4
2019-07-19 17:25 +0200 Sun Seng David TAN o updating devfile switc
2019-07-19 09:41 +0200 Sun Seng David TAN o fix entrypoint
2019-07-15 20:31 +0200 Sun Seng David TAN o fixing build and devfi
[main] 15ad3a725fbd6210fafb1cdf0745401fa3faaaef - commit 1 of 237%
```

```
2019-07-30 14:56:48,910 INFO  [io.quarkus] (main) Quarkus
 0.14.0 started in 0.045s. Listening on: http://0.0.0.0:8
080
2019-07-30 14:56:48,910 INFO  [io.quarkus] (main) Install
ed features: [cdi, resteasy]
```

Your new Cloud-Native application is ready!

Congratulations, you have created a new Quarkus application.

Why do you see this?

This page is served by Quarkus. The source is in `src/main/resources/META-INF/resources/index.html`.

/hello endpoint:

hello

Applica

GroupId: com.examp

ArtifactId: hello-quarkus

Version: 1.0-SNAPSHO

Quarkus Version: 0.14.0

Next steps

Setup your IDE

Getting started

Quarkus Web

New Architectural Components

Kubernetes Based Workspaces

Workspaces: Developer Sandbox centrally hosted on OpenShift

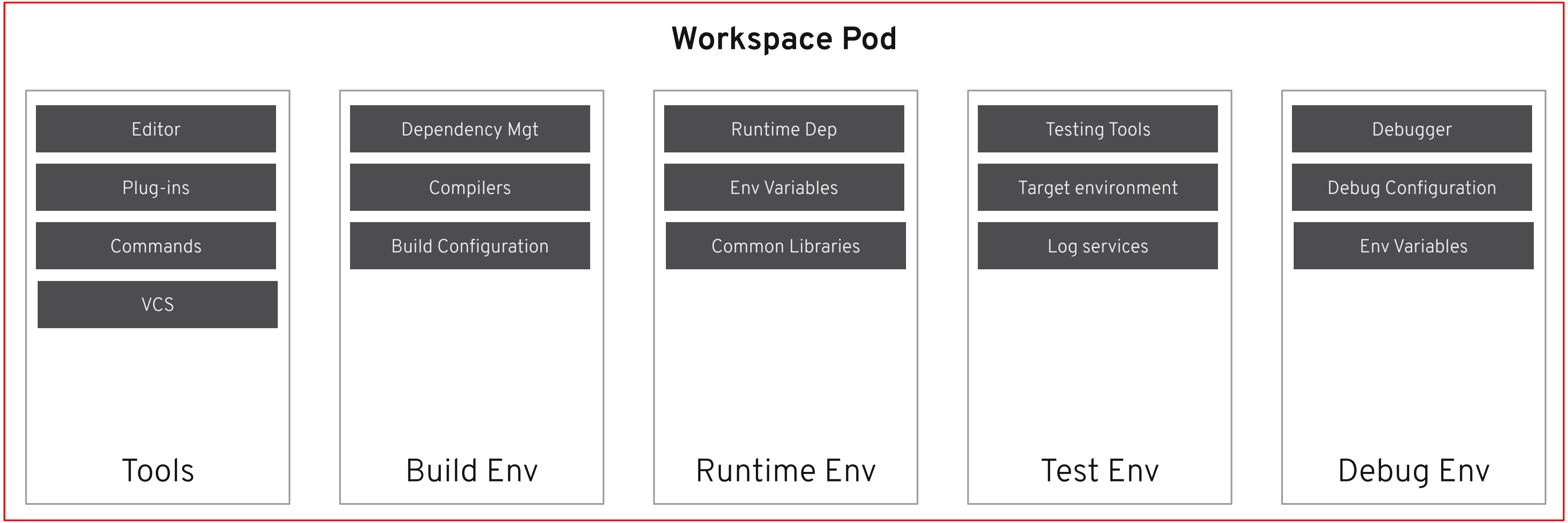
A CRW workspace is a **developer sandbox running on OpenShift**, where everything needed to code on a project is packaged into containers.

It provides:

- Application runtimes
- Build tools
- Development tools: Browser based editor + plugins
- Project source code repositories

Workspaces: Developer Sandbox

Containerizing everything you need to develop, build, run, test and debug your application.



Workspace Pod

Tools
- Editor
- Plug-ins
- Commands
- VCS

Build Env
- Dependency Mgt
- Compilers
- Build Configuration

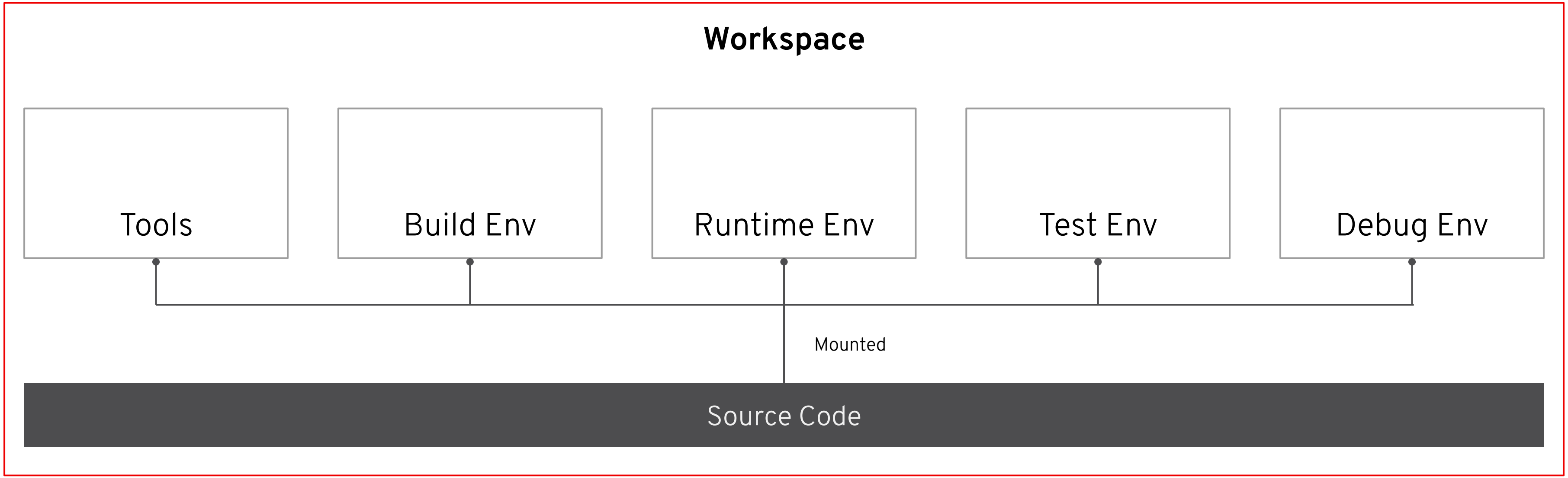
Runtime Env
- Runtime Dep
- Env Variables
- Common Libraries

Test Env
- Testing Tools
- Target environment
- Log services

Debug Env
- Debugger
- Debug Configuration
- Env Variables

Workspaces: Developer Sandbox

Source code is getting mounted in your workspace, accessible from all containers.

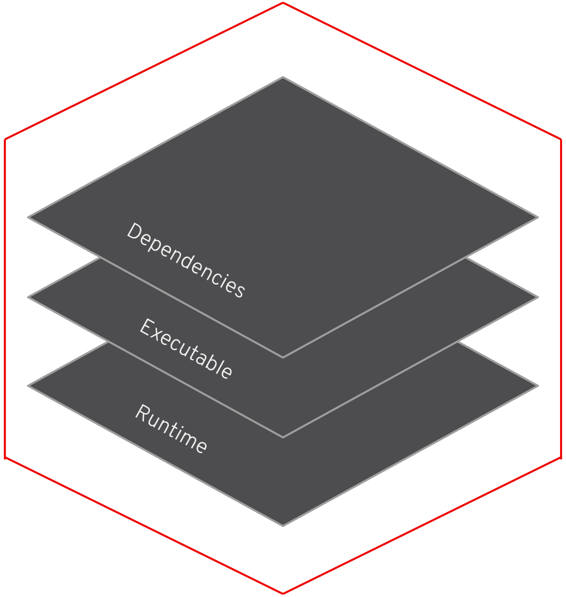


Containerized Development Tools

Development tools (editors and plug-ins) are packaged with their runtimes and dependencies:

- Zero dependency installation
- Isolated execution
- Own lifecycle
- Easy upgrade/switch
- Simpler packaging
- Scalable

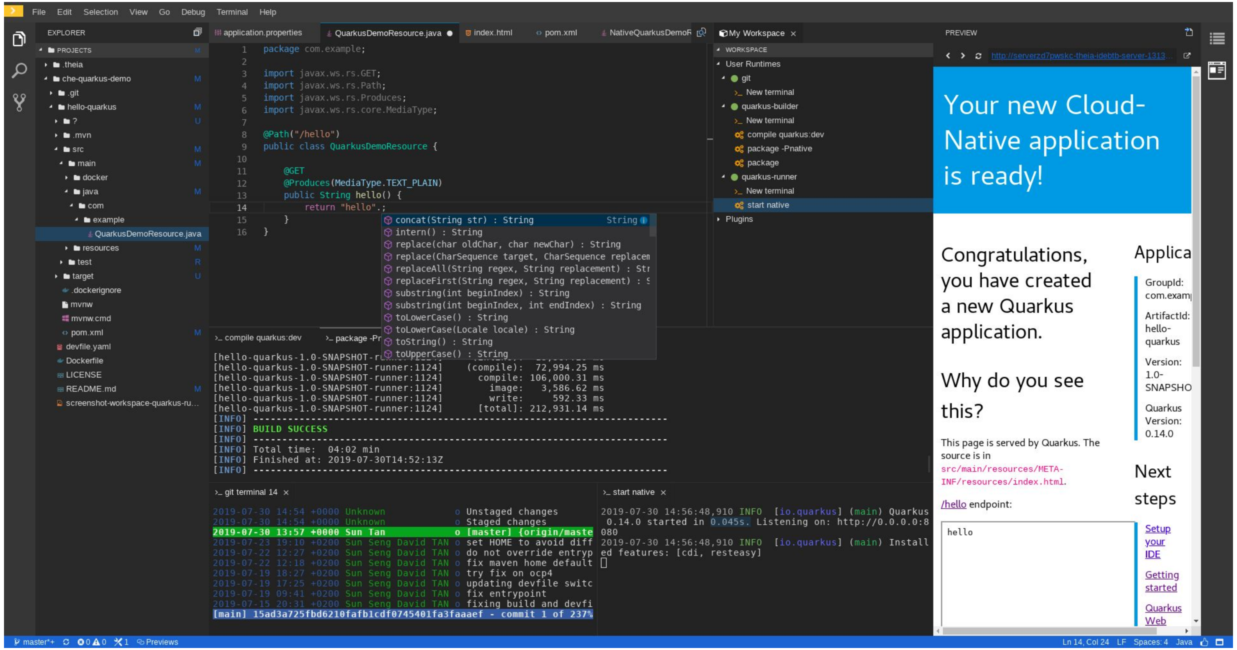
Package VSCode extensions in the container.



New Editor

New Editor: VSCode in the browser

Containerizing the Editor: zero install and automate configuration



```java
package com.example;

import javax.ws.rs.GET;
import javax.ws.rs.Path;
import javax.ws.rs.Produces;
import javax.ws.rs.core.MediaType;

@Path("/hello")
public class QuarkusDemoResource {

    @GET
    @Produces(MediaType.TEXT_PLAIN)
    public String hello() {
        return "hello".;
    }
}
```

```
concat(String str) : String                        String
intern() : String
replace(char oldChar, char newChar) : String
replace(CharSequence target, CharSequence replacem
replaceAll(String regex, String replacement) : Str
replaceFirst(String regex, String replacement) : S
substring(int beginIndex) : String
substring(int beginIndex, int endIndex) : String
toLowerCase() : String
toLowerCase(Locale locale) : String
toString() : String
toUpperCase() : String
```

```
[hello-quarkus-1.0-SNAPSHOT-r
[hello-quarkus-1.0-SNAPSHOT-runner:1124]    (compile):  72,994.25 ms
[hello-quarkus-1.0-SNAPSHOT-runner:1124]      compile: 106,000.31 ms
[hello-quarkus-1.0-SNAPSHOT-runner:1124]        image:   3,586.62 ms
[hello-quarkus-1.0-SNAPSHOT-runner:1124]        write:     592.33 ms
[hello-quarkus-1.0-SNAPSHOT-runner:1124]      [total]: 212,931.14 ms
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  04:02 min
[INFO] Finished at: 2019-07-30T14:52:13Z
[INFO] ------------------------------------------------------------------------
```

```
2019-07-30 14:54 +0000 Unknown             o Unstaged changes
2019-07-30 14:54 +0000 Unknown             o Staged changes
2019-07-30 13:57 +0000 Sun Tan             o [master] {origin/maste
2019-07-23 19:10 +0200 Sun Seng David TAN o set HOME to avoid diff
2019-07-22 12:27 +0200 Sun Seng David TAN o do not override entryp
2019-07-22 12:18 +0200 Sun Seng David TAN o fix maven home default
2019-07-19 18:27 +0200 Sun Seng David TAN o try fix on ocp4
2019-07-19 17:25 +0200 Sun Seng David TAN o updating devfile switc
2019-07-19 09:41 +0200 Sun Seng David TAN o fix entrypoint
2019-07-15 20:31 +0200 Sun Seng David TAN o fixing build and devfi
[main] 15ad3a725fbd6210fafb1cdf0745401fa3faaaef - commit 1 of 237%
```

```
2019-07-30 14:56:48,910 INFO  [io.quarkus] (main) Quarkus
 0.14.0 started in 0.045s. Listening on: http://0.0.0.0:8
080
2019-07-30 14:56:48,910 INFO  [io.quarkus] (main) Install
ed features: [cdi, resteasy]
```

Your new Cloud-Native application is ready!

Congratulations, you have created a new Quarkus application.

Why do you see this?

This page is served by Quarkus. The source is in `src/main/resources/META-INF/resources/index.html`.

/hello endpoint:

```
hello
```

Applica

GroupId: com.examp

ArtifactId: hello-quarkus

Version: 1.0-SNAPSHO

Quarkus Version: 0.14.0

Next steps

Setup your IDE

Getting started

Quarkus Web

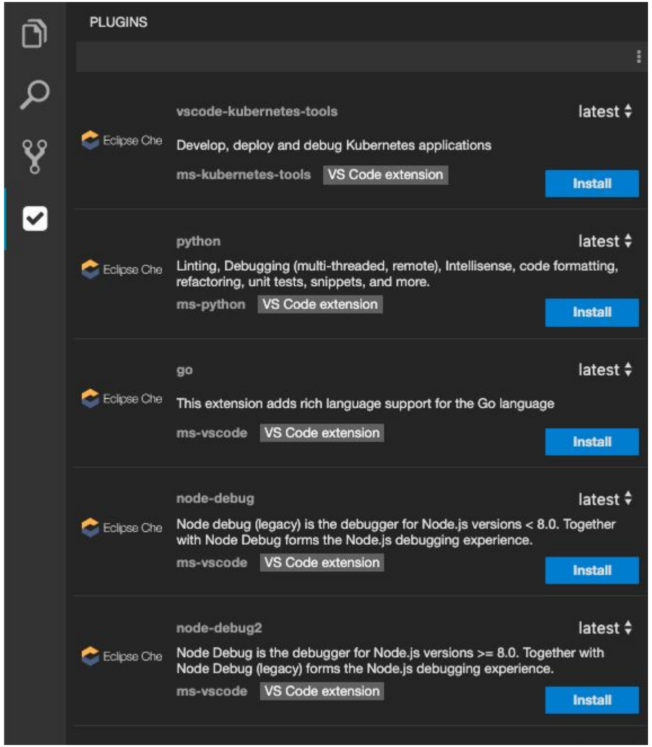
Extended Eclipse Theia, to provide a top-of-the-art editor experience.

Built-in:

- Languages Server Protocol
- Debug Adapter Protocol

Compatible with VSCode extensions

Customize with VSCode extensions



vscode-kubernetes-tools
Develop, deploy and debug Kubernetes applications
ms-kubernetes-tools VS Code extension

python
Linting, Debugging (multi-threaded, remote), Intellisense, code formatting, refactoring, unit tests, snippets, and more.
ms-python VS Code extension

go
This extension adds rich language support for the Go language
ms-vscode VS Code extension

node-debug
Node debug (legacy) is the debugger for Node.js versions < 8.0. Together with Node Debug forms the Node.js debugging experience.
ms-vscode VS Code extension

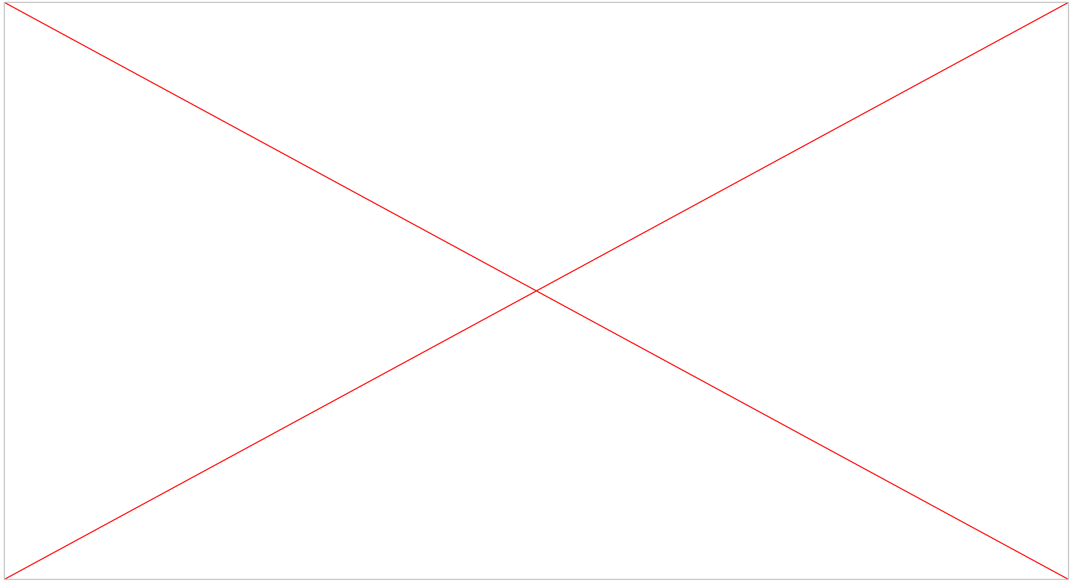
node-debug2
Node Debug is the debugger for Node.js versions >= 8.0. Together with Node Debug (legacy) forms the Node.js debugging experience.
ms-vscode VS Code extension

- Visual Studio Code extensions compatibility
- Extensions packaged with their dependencies
- Plug-in registry with predefined set of plug-ins

All Red Hat extensions available for VSCode will be available with CodeReady Workspaces 2.

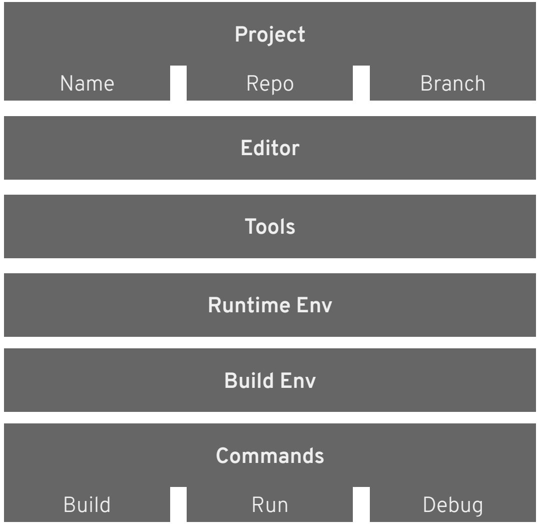
Enriched Developer Experience

- Keyboard Navigation: Command Palette
- Rich Editor
  - Find/Replace instances
  - Peek Definition
  - Outline
- Improved Debug
- Git Integration
- Layout customization
- Theming
- Port detection Plugin



Devfile

Devfile: Developer environment as code



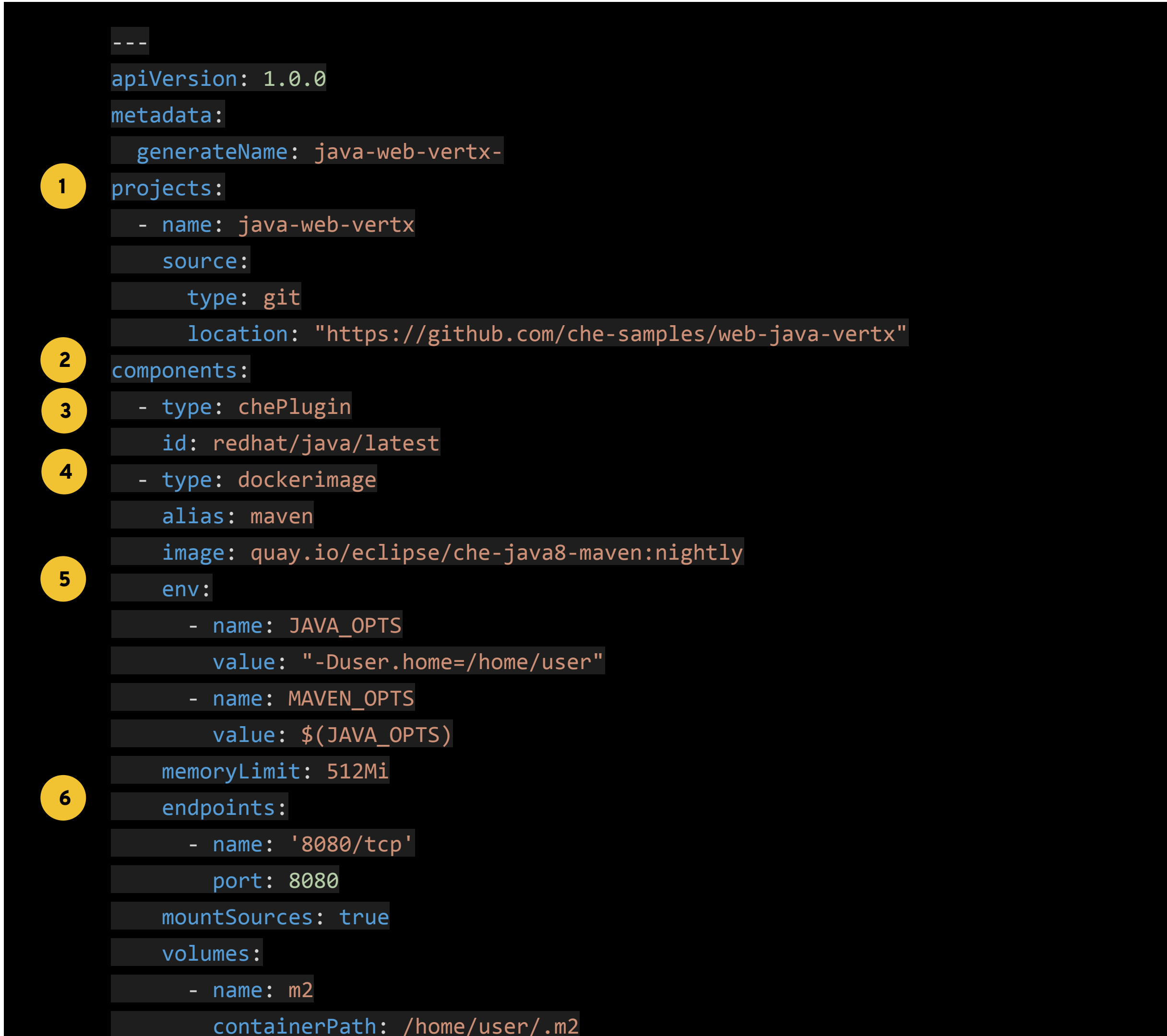
Devfile Definition

The devfile provides easy-to-configure, highly reproducible definitions of portable developer environments.

It is a declarative abstraction of a replicable developer workspaces, which includes the runtime environments, the source code of the projects mapped to repositories and the tools, plugins and commands needed to code, build, test, run and debug a project.

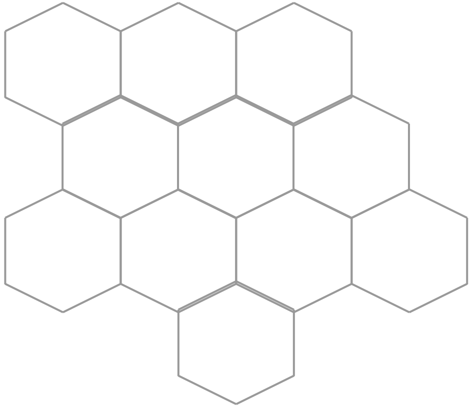
Devfile Example

1. Project information
2. List of components of the workspace
3. Plugin component
4. Runtime image
5. Env variables to configure the container
6. End-points definition

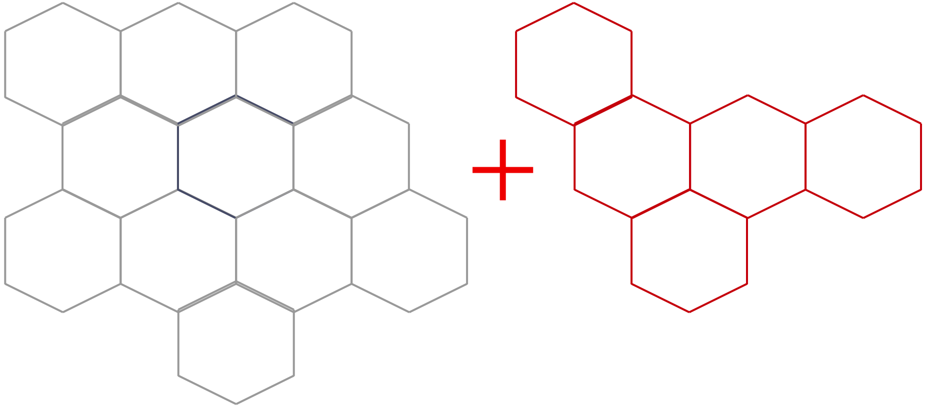


```
---
apiVersion: 1.0.0
metadata:
  generateName: java-web-vertx-
projects:
  - name: java-web-vertx
    source:
      type: git
      location: "https://github.com/che-samples/web-java-vertx"
components:
  - type: chePlugin
    id: redhat/java/latest
  - type: dockerimage
    alias: maven
    image: quay.io/eclipse/che-java8-maven:nightly
    env:
      - name: JAVA_OPTS
        value: "-Duser.home=/home/user"
      - name: MAVEN_OPTS
        value: $(JAVA_OPTS)
    memoryLimit: 512Mi
    endpoints:
      - name: '8080/tcp'
        port: 8080
    mountSources: true
    volumes:
      - name: m2
        containerPath: /home/user/.m2
```

“Dev mode” your kubernetes application



Use your K8S application definition



Supercharge your tools and start coding

Devfile: Made for Team

Manage Consistency

First class support of K8S, manage the complexity of developer environments take out the pain.

Easy to Integrate

Devfiles are easy to integrate with any tools. Developer environments can be created from anywhere at anytime.

Simple to Share

Devfile live with source code, are easy to modify, fork and share.

Extensible

Customize per task, with plug-in and developer preferences

DEMO

Photo by Unsplash

Stacks

List of out-of-the-box stacks

	Versions
.NET	.Net 2.1
C/C++	C/C++
GO	Go 1.11.5
JAVA-EAP-MAVEN	OpenJDK 8, Maven 3.5, EAP 7.2
JAVA-MAVEN	OpenJDK 8, Maven 3.5
JAVA-SPRINGBOOT	OpenJDK 8, Maven 3.5
JAVA-VERTX	OpenJDK 8, Maven.35
JAVA-THORNTAIL	OpenJDK 8, Thorntail 2.5
NODEJS	NodeJS 10
NODEJS-MONGODB	NodeJS 10, MongoDB 3.4
PHP	PHP 7.1
PYTHON	Python 3.6
RED HAT FUSE	

All stacks are built from the available images on the container catalog and UBI based.

All default stacks are available here:

Plug-ins

List of Plug-ins

	Description	Version
JAVA LS	Java Language Support (Intellisense and Debugger)	
TYPESCRIPT LS	Typescript Language Support (Intellisense and Debugger)	
NODE DEBUG	Node >= 8.0 Support (Debugger)	
OPENSHIFT CONNECTOR	Interact with an OpenShift Cluster	
KUBERNETES	Develop, deploy and debug Kubernetes applications	
DEPENDENCY ANALYTICS	Red Hat Dependency Analytics plug-in	
CAMEL TOOLING	Camel Language Support	
YAML LS	YAML Language Support	
XML LS	XML Language Support	

	Description	Version
OMNISHARP	.Net Language Support	
GO	GO Language Support	
PHP	PHP Language Support (Intellisense and Debugger)	
CPPTOOLS	C/C++ Language Support (Intellisense and Debugger)	
PROJECT INITIALIZER	A lightweight extension based on Red Hat launcher to generate	
WSDL2REST	Extension supporting the mapping of an existing SOAP service to a REST service.	
PYTHON	Python Language Support	

OpenShift 4.2 Integration

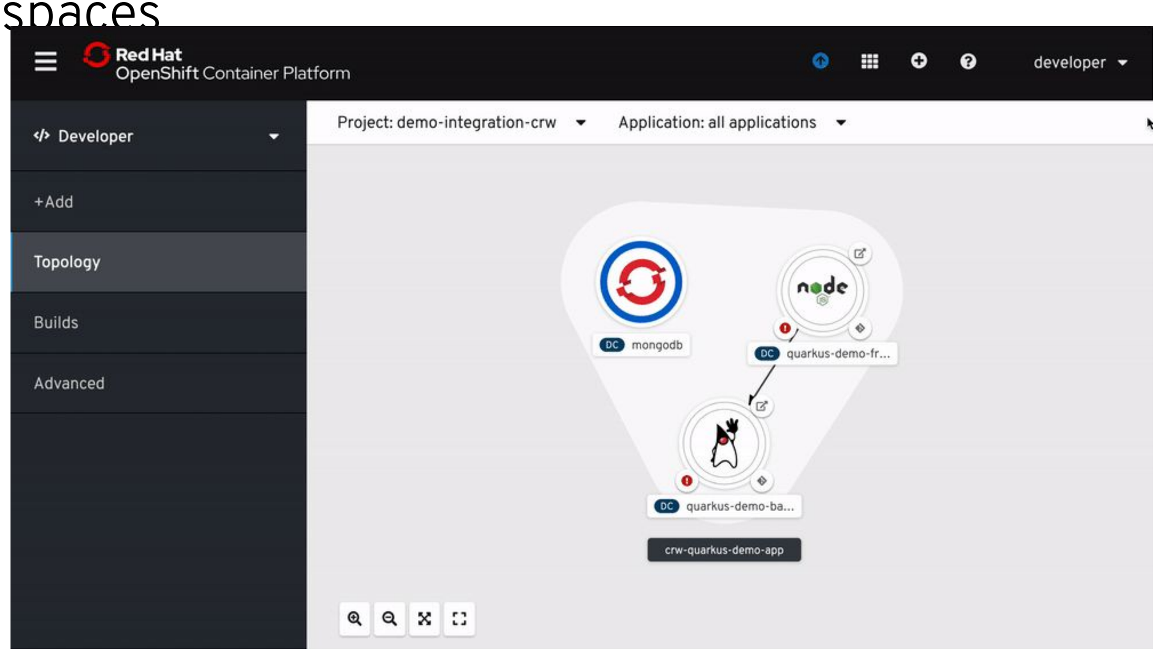
OpenShift Edit with CodeReady Workspaces

From the Developer perspective

- See the applications components
- Get details about all components

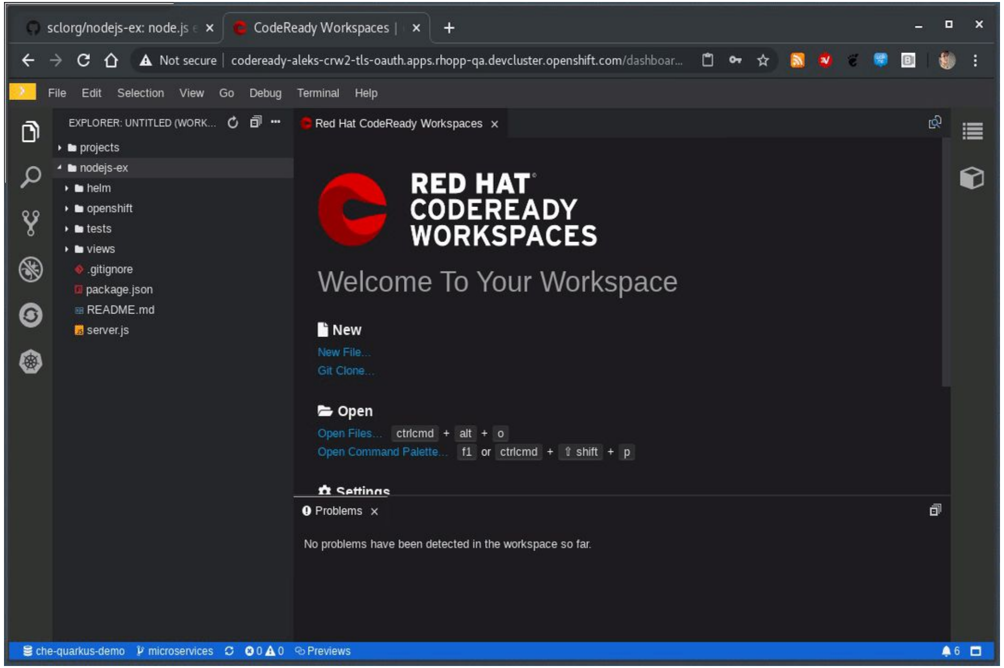
If CodeReady Workspaces is deployed:

- Quick access link to Workspaces
- Edit button is surfacing in the UI to allow coding on application's components



OpenShift Plug-in

- Speeds up OpenShift development. Connect to any OpenShift cluster and create, debug, and deploy from CodeReady Workspaces itself.
- Uses OpenShift Do (odo) to simplify inner-loop development for cloud infrastructure and OpenShift CLI (oc) to help you interact with the OpenShift instance and complete the inner-loop experience.
- Compatible with OpenShift instances (3.x or 4.x) and supporting public cloud instances such as Red Hat OpenShift on Azure and AWS.



sclorg/nodejs-ex: node.js

CodeReady Workspaces |

Not secure | codeready-aleks-crw2-tls-oauth.apps.rhopp-qa.devcluster.openshift.com/dashboar...

EXPLORER: UNTITLED (WORK...

- projects
- nodejs-ex
  - helm
  - openshift
  - tests
  - views
  - .gitignore
  - package.json
  - README.md
  - server.js

Red Hat CodeReady Workspaces

RED HAT CODEREADY WORKSPACES

Welcome To Your Workspace

New

New File...

Git Clone...

Open

Open Files... ctrlcmd + alt + o

Open Command Palette... f1 or ctrlcmd + ⇧ shift + p

Settings

Problems

No problems have been detected in the workspace so far.

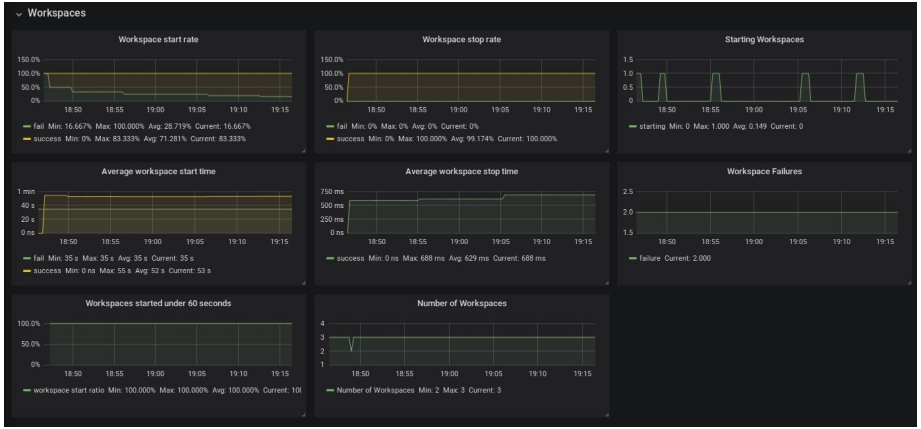
che-quarkus-demo microservices 0 0 Previews 6

OPTIONAL SECTION MARKER OR TITLE

Monitoring and Administering

Monitoring Capabilities

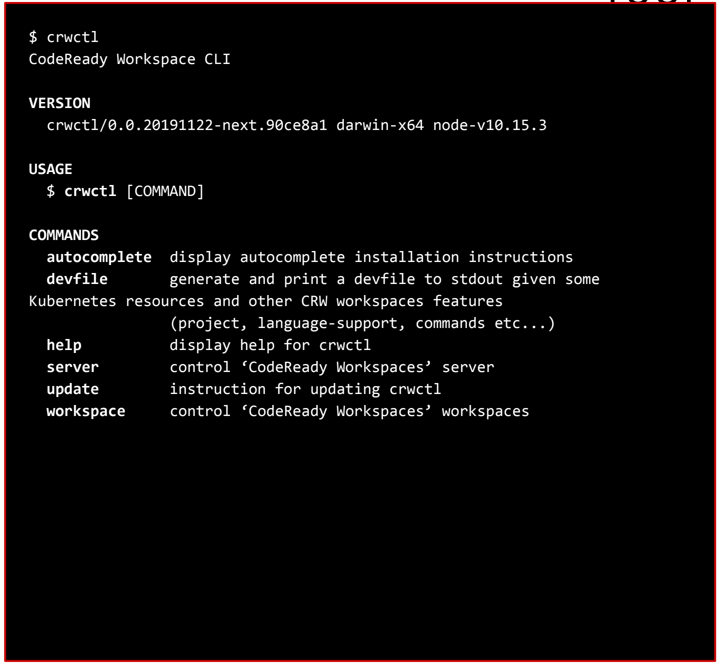
- CodeReady Workspaces provides support for OpenTracing API. Connect to tracing tools (ex:Jaeger) to analyze the behaviour of transactions happening when Workspaces is deployed on your infrastructure.
- Combined with Grafana and Prometheus, get CodeReady Workspaces Metrics to help you monitor your deployment. Number of users, workspaces running, startup time, JVM state and many other aspects to help operating and monitoring your installation are getting available.


Workspaces
Workspace start rate
150.0%
100.0%
50.0%
0%
18:50
18:55
19:00
19:05
19:10
19:15
fail Min: 16.667% Max: 100.000% Avg: 28.719% Current: 16.667%
success Min: 0% Max: 83.333% Avg: 71.281% Current: 83.333%
Workspace stop rate
150.0%
100.0%
50.0%
0%
fail Min: 0% Max: 0% Avg: 0% Current: 0%
success Min: 0% Max: 100.000% Avg: 99.174% Current: 100.000%
Starting Workspaces
1.5
1.0
0.5
0
starting Min: 0 Max: 1.000 Avg: 0.149 Current: 0
Average workspace start time
1 min
40 s
20 s
0 ns
fail Min: 35 s Max: 35 s Avg: 35 s Current: 35 s
success Min: 0 ns Max: 55 s Avg: 52 s Current: 53 s
Average workspace stop time
750 ms
500 ms
250 ms
0 ns
success Min: 0 ns Max: 688 ms Avg: 629 ms Current: 688 ms
Workspace Failures
2.5
2.0
1.5
failure Current: 2.000
Workspaces started under 60 seconds
100.0%
50.0%
0%
workspace start ratio Min: 100.000% Max: 100.000% Avg: 100.000% Current: 10(
Number of Workspaces
4
3
2
1
Number of Workspaces Min: 2 Max: 3 Current: 3

Command line tool

NEW COMMAND LINE TOOL

CodeReady Workspaces Command Line Tool



```
$ crwctl
CodeReady Workspace CLI

VERSION
  crwctl/0.0.20191122-next.90ce8a1 darwin-x64 node-v10.15.3

USAGE
  $ crwctl [COMMAND]

COMMANDS
  autocomplete  display autocomplete installation instructions
  devfile       generate and print a devfile to stdout given some
Kubernetes resources and other CRW workspaces features
                (project, language-support, commands etc...)
  help          display help for crwctl
  server        control ‘CodeReady Workspaces’ server
  update        instruction for updating crwctl
  workspace     control ‘CodeReady Workspaces’ workspaces
```

CLI tool which helps install, manage and use CodeReady Workspaces on your OpenShift cluster.

Install the CLI locally and manage remote clusters.

Pre-check installation of Workspaces.

Administration tasks for Workspaces.

Stack and Plug-ins Registries

Stack and Plug-ins Registries

Tools Registry

In-house deployment

Define plug-ins

Define IDEs

Devfile Registry

Provide ready-to-use developer stacks

Manage stacks

AirGap Capabilities

AirGap Capabilities

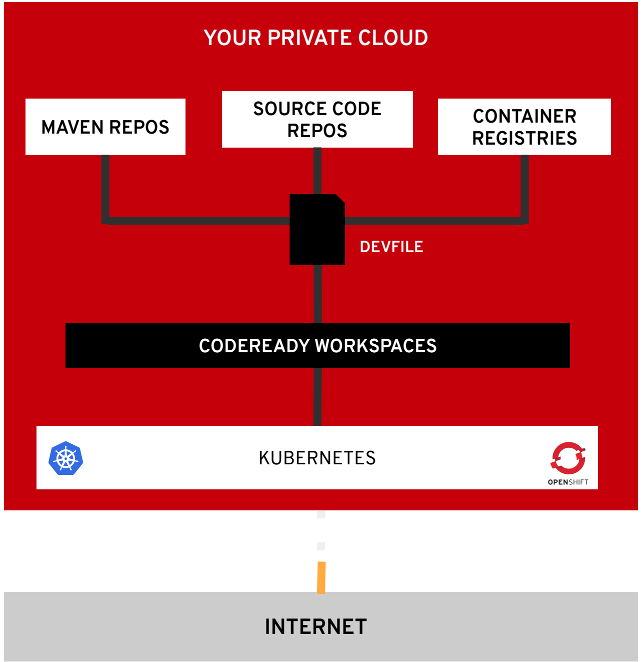
CodeReady Workspaces can be installed and used in an AirGap environment, on your own private cloud.

Stacks will be configured to point to your own image registry.

Plug-ins will be hosted with the plug-ins registry deployed on your private cloud.

Workspaces can be configured to:

- Point to your private git repository
- Leverage images from your container registries.
- Use your own maven, npm or any dependency repository you are using

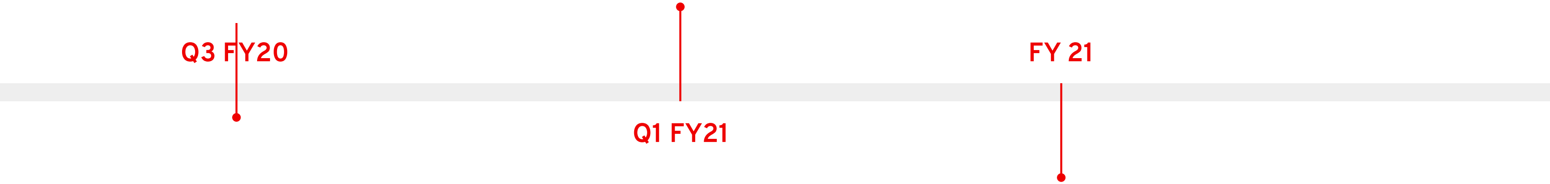


OPTIONAL SECTION MARKER OR TITLE

Roadmap

DRAFT

CodeReady Workspaces Roadmap



Q3 FY20

CRW 2.0

Based on Che 7

Support for OSD 4

Offline installation

Console Dev Integration

Monitoring Dashboard

Q1 FY21

CRW 2.1

Telemetry

Installation Diagnostic

New Create Workspace Flow

Commands/Tasks Consolidation

Devfile lifecycle

Firefox support

Performances improvements

FY 21

CRW 2.x

Integration with ODO, Pipelines, Serverless

Toolchain integration plug-ins

Admin command-Line

On-the-go Workspace Configuration

Thank you

- linkedin.com/showcase/red-hat-developer
- youtube - bit.ly/2YRIWTk
- facebook.com/redhatdeveloperprogra m
- twitter.com/rhdevelopers