

An Intro to Eclipse Che

June 2015

Tyler Jewell, Eclipse Che Project Lead

@TylerJewell

The Application Trinity

Develop Apps

issue mgmt

source code repo

agile

ide



workspace

Deliver Apps

check out &
build code

unit test

quality control

package &
archiving

integration
testing

deploy to test
environment

deploy to pre-
production

acceptance
testing

deploy to
production

continuous
integration

packaging &
build tools

code
quality
analysis

testing
frameworks

artifact
repository

release
mgmt

100 firms including Atlassian, JetBrains, Puppet, Chef, Ansible, ZeroTurnaround, IBM, Electric Cloud, Coverity, Sonatype, JFrog, and so on...

Host Apps

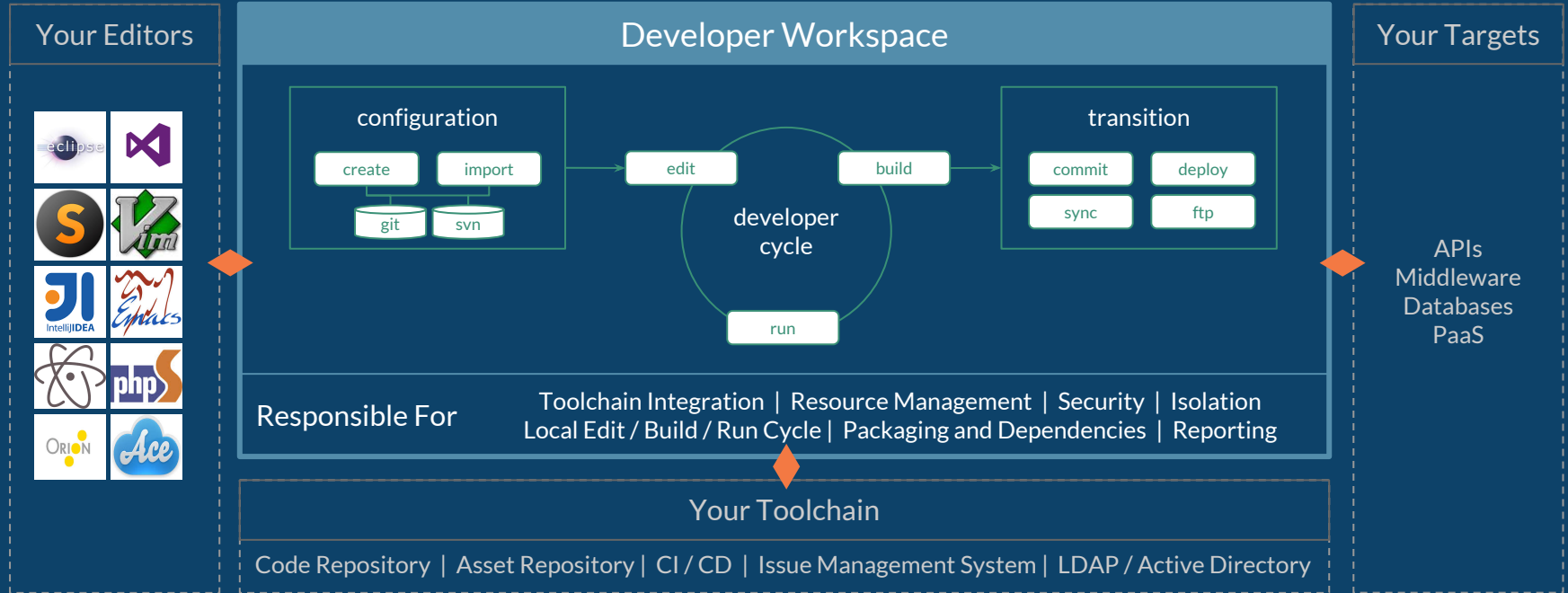
test

pre-production

production

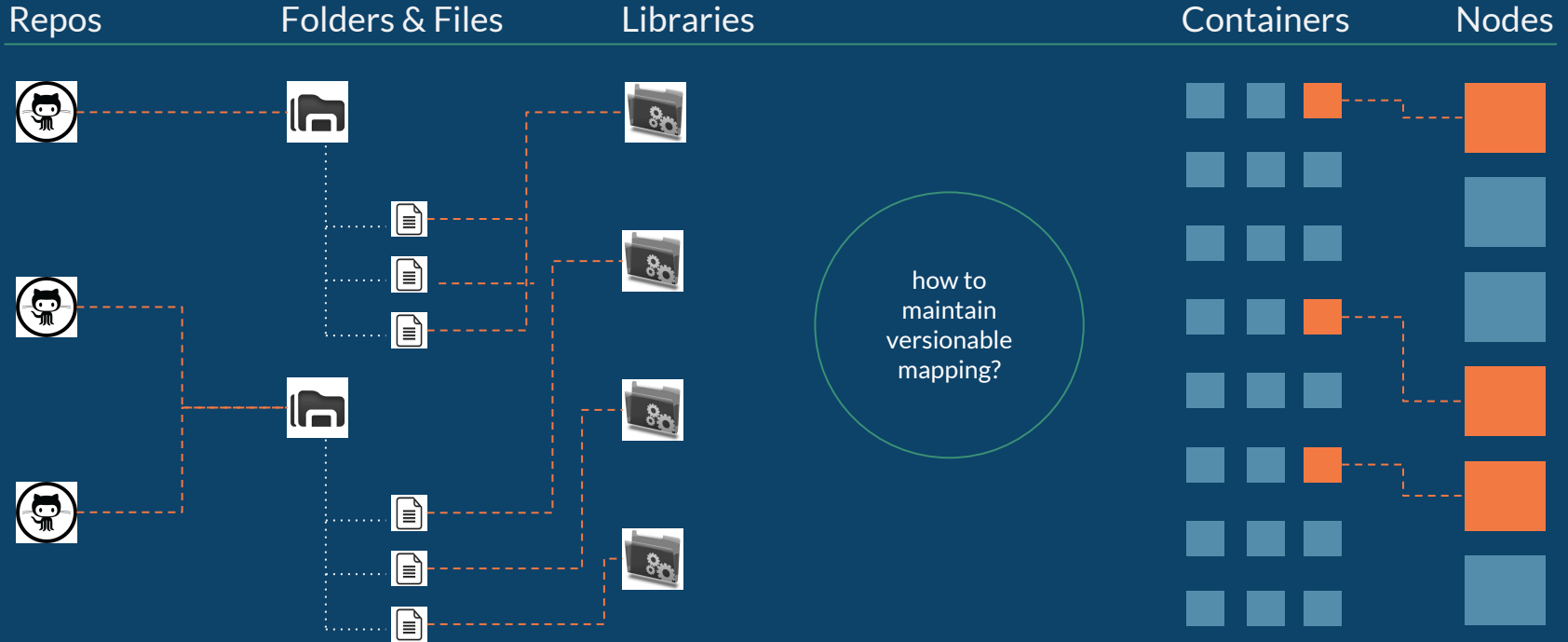


The Developer Workspace



Localhost is not cloneable, shareable, scalable or compliant.

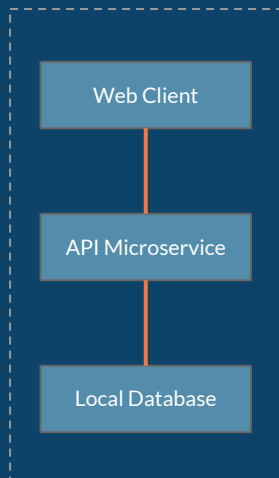
Containers Shift Burdens Away From DevOps



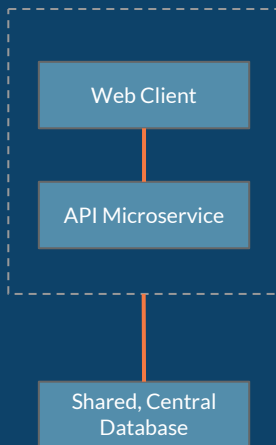
Containers and Composition

Imagine a developer working on a 3-tier application using containers. The container configuration can change frequently.

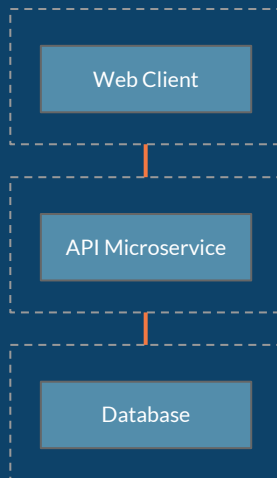
Rapid Authoring



Staged Data



Networked

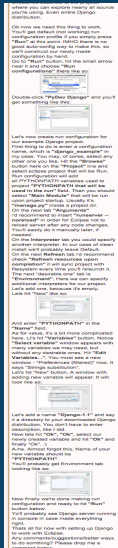


Docker Compose is a step forward...

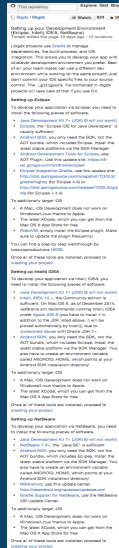
```
web:  
  build: .  
  links:  
    - db  
  ports:  
    - "8000:8000"  
db:  
  image: postgres
```

Reality: Workspace Configuration via Wiki

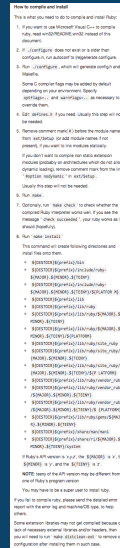
Django and PHP with Eclipse



Tomcat with IntelliJ



Compile and install Ruby



Mozilla Webmaker with Vagrant



Continuous delivery demands repeatability - but developer workspace configuration is individual & complex.

Eclipse Che is an SDK, hosted workspace, and cloud IDE.

Package Che extensions (you author) with custom machines into new assemblies.

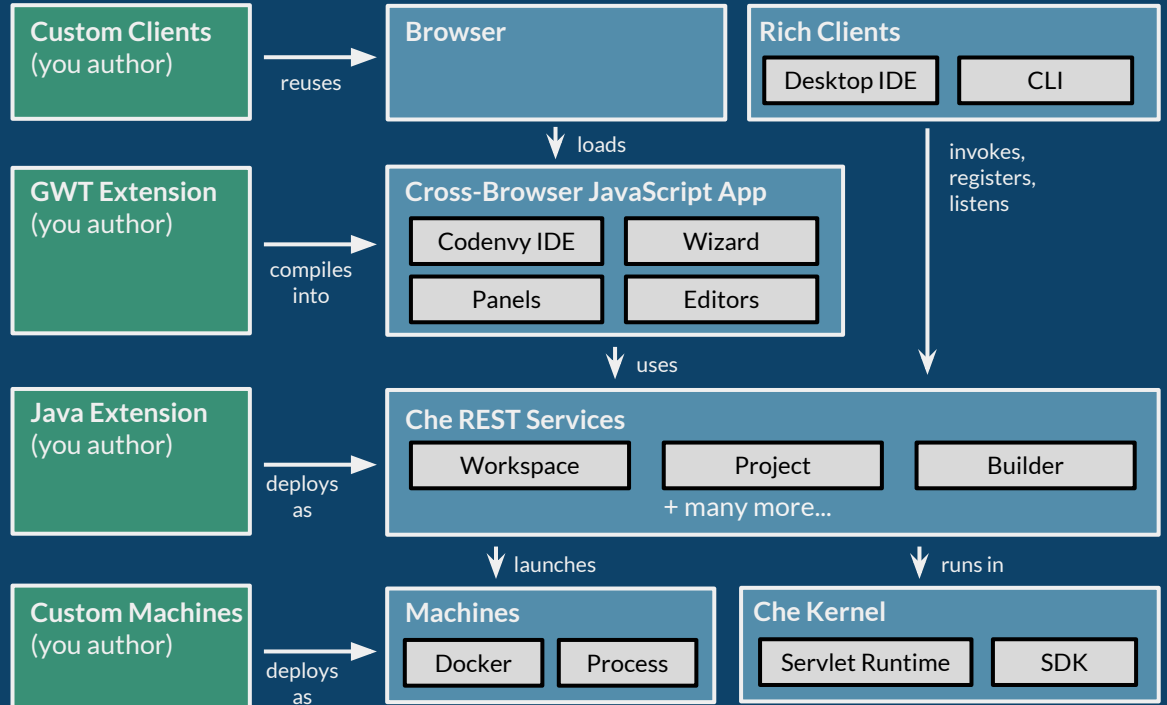
Che can be used as a desktop workspace, embedded workspace, or hosted workspace.

Eclipse Che Extensibility

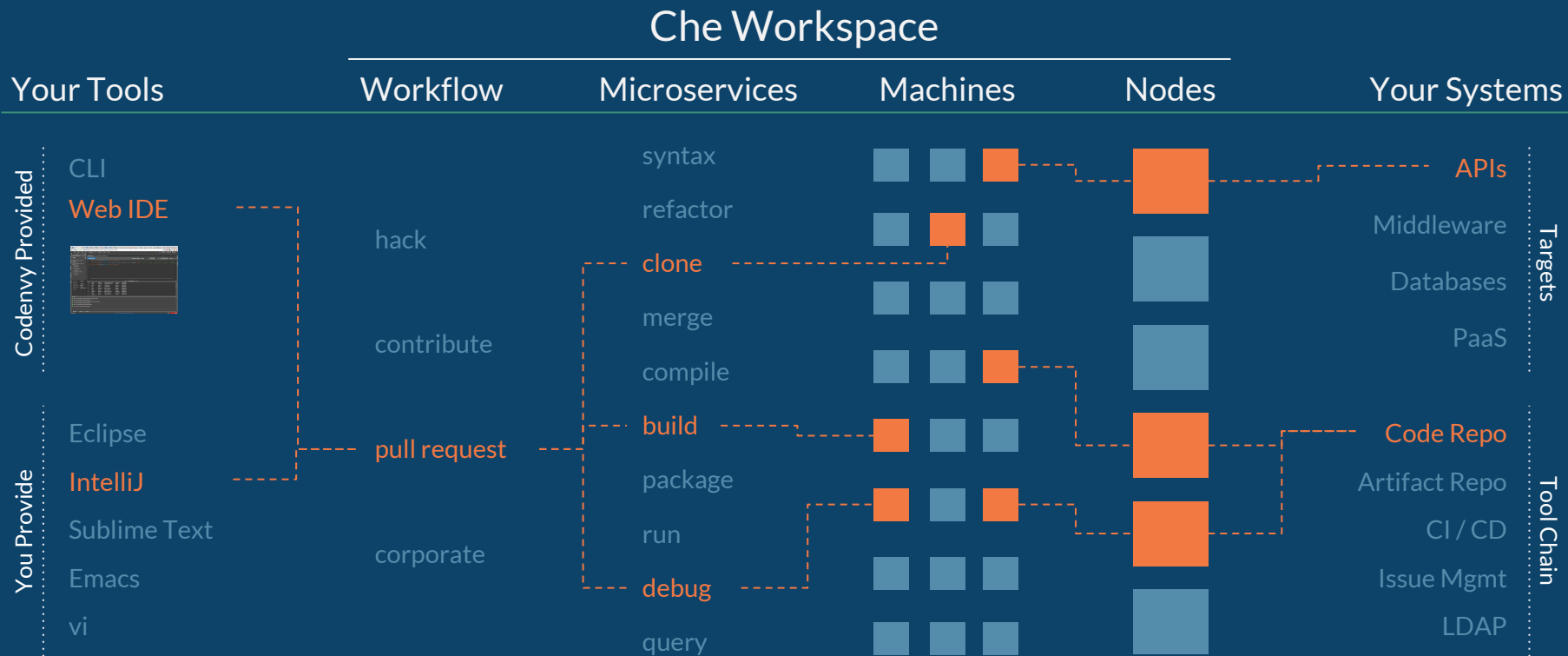
Che combines a kernel, developer REST services, machines and a JavaScript cloud IDE.

Developers customize Che by writing extensions at any tier, overriding any behavior.

Codenvy is a multi-tenant system that runs Eclipse Che extensions.



Che Workspaces Distribute Actions



Eclipse Che & Codenvy IP Relationship

Eclipse Che

IDE + SDK

REST API

project builder runner
auth factory analytics

Che Extensions

java javascript c++
maven gradle 55+

Your Extensions

Plug-In | Plug-In | Plug-In | Plug-In | Plug-In

Kernel

Codenvy Platform

Automation  | Management 
Factories Contrib Admin Analytics

API CLI Browser Mobile

Operations

HA Tenancy Metering Security

Eclipse Che

Languages Editors Projects

External Integration Plug-Ins

Source CI PaaS Issue Identity

Developer Microservice Plug-Ins

Code Build Run Source Query

Elastic Docker Node Pool



Distribution

update.codenvy.com

Notices Accounts Binaries Installers

Offerings

SaaS@ codenvy.com On-Prem @ Your Datacenter Managed @ Any Datacenter

Installation

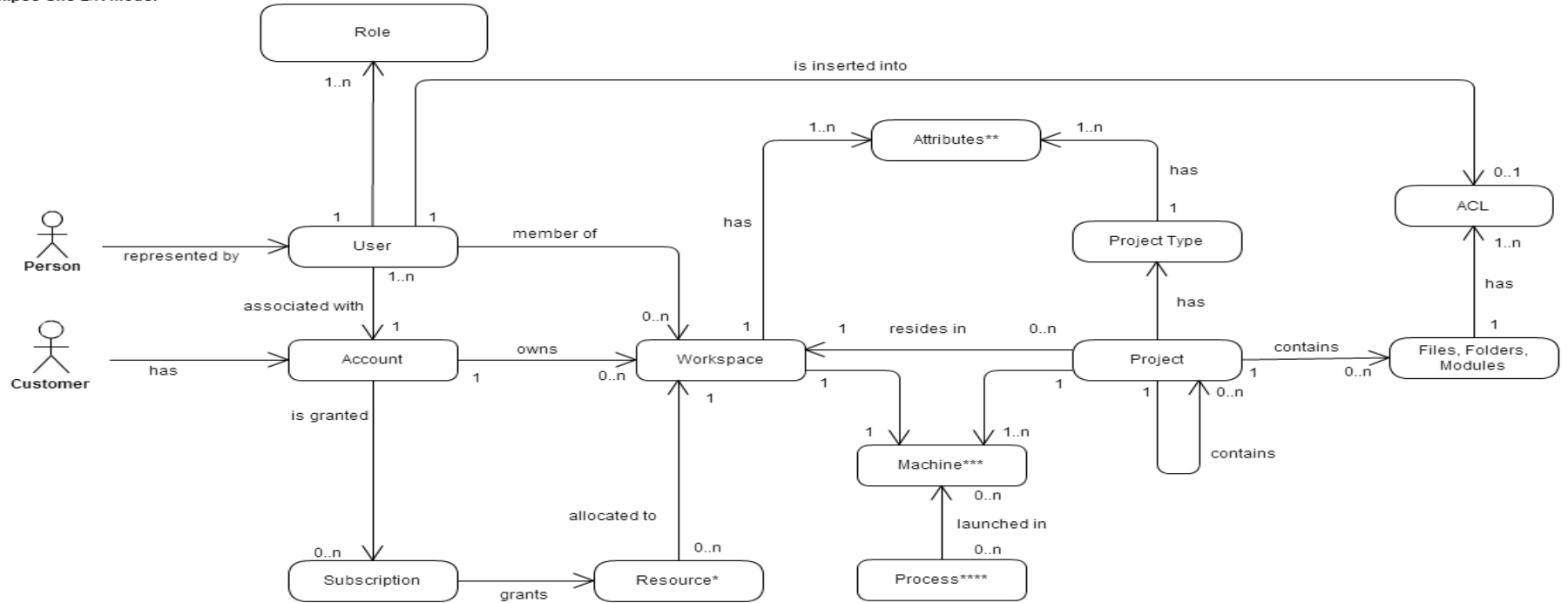
1-Node n-Node CentOS 6 CentOS 7

Cloud Marketplaces

IBM AWS Google ...

The Che Glossary

Eclipse Che E/R Model



Che 4.0: Project - Machine - Env Abstraction

Object	Purpose	Composition	Manages	For	User Creates In
Account	Object contains the information about billing, subscription, and organizations owned. An administrative unit, used by managers and admins to bucket groups of users and resources together, with different allocation polices & controls.	has 1 .. n account/owner has 0 .. n account/member has 0 .. n account/manager	Name Billing Owner Members Resource limits (gbh) via subs Resource limits (ram) via subs	Codenvy	System UD
Workspace	A thin layer that binds the set of projects and their source code to a single working machine aggregate.	has 1 .. 1 dev machine has 0 .. n projects has 0 .. n environments has 1 .. n ws/admin has 0 .. n ws/member has 0 .. n ws/stakeholder	Dev machine recipe Dev machine instance per user	Codenvy / Che	UD IDE
Project	A single project, mapped to a single source repository, that has type. A module is a project that is a child of another project.	has 1 .. 1 type has 1 .. 1 source repository has 0 .. n ACLs has 0 .. n modules	Source repository Public / private	Codenvy / Che	UD IDE
Environment	An abstraction that contains a networked group of machines used to provide runtime capabilities for a set of projects.	has 1 .. 1 recipe has 1 .. n machines has 1 .. 1 network has 0 .. n proj-machine-map	Run machines Run networks Linkage between project & machine	Codenvy / Che	IDE

Workspace Behavior in 4.0

A development machine is a special purpose machine that is bound to a Workspace for the purposes of development. It can hold a content assist system, the source code, and also receive commands. All commands - by default - are routed to the dev machine, including run commands. Dev machines are managed by Codenvy. Codenvy will start / stop / activate / suspend / snapshot against our own policy based upon when a user open / closes a workspace.

Separately, a developer will be able to define an Environment. An environment is a runtime environment that maps a set of projects in a single workspace against a set of machines. The environment has a recipe for how the machines should boot, how the source code / artifacts are placed into the environment, and the network connections in between them. Essentially, it's a type of Docker Compose, but belonging to the Workspace. Users of the Workspace will have control over starting / stopping an environment. The environment will have runtime isolation from the dev machine.

All development machines start with ½ GB RAM. Workspace admins can modify the RAM and Dockerfile for the development machine. Workspace members will be able to edit and run environments. Any changes to a development machine configuration will restart the development machine with loss of any state.

Getting Started with Che

Source: github.com/codenvy/che
Developer Mailing List: che-dev@eclipse.org
Planning Meetings: wiki.eclipse.org/che
IRC: [#eclipseche](#) (routes to slack)

Note: Currently 1000 issues open on Che, hosted at Codenvy's private Jira. We are working to export into bugzilla.

Che Extension

`plugin-angularjs/core/client/src/main/java/com/codenvy/plugin/angularjs/core/client/`

```
@Singleton
@Extension(title = "AngularJS")
public class AngularJsExtension extends JsExtension {
    @Inject
    public AngularJsExtension(IconRegistry iconRegistry, AngularJSResources resources) {
        super(Const.ANGULAR_JS_ID, iconRegistry, resources);
    }
}
```

`plugin-angularjs/core/server/src/main/java/com/codenvy/plugin/angularjs/core/server/project/type`

```
@Singleton
public class AngularJSProjectType extends ProjectType {
    public AngularJSProjectType() {
        super("AngularJS", "AngularJS Project", true, false);
        setDefaultRunner("system:/javascript/webapp/grunt");
        addRunnerCategories(Arrays.asList(RunnerCategory.JAVASCRIPT.toString()));
    }
}
```

Eclipse Che

www.eclipse.org/che
Try it live: codenvy.com