Red Hat
Training and
Certification

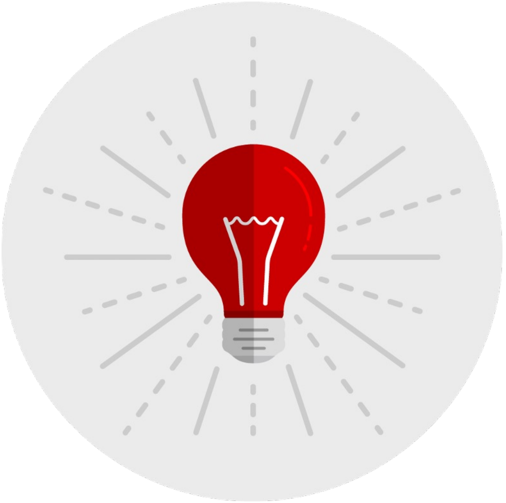
RED HAT
OPEN INNOVATION LABS

DevOps Culture and Practice Enablement

Exercise 1 - Introduction

Red Hat

Topics



Lab Setup and Access

How to setup your account and access CodeReady Workspaces and the OpenShift Environment

Codeready Workspaces

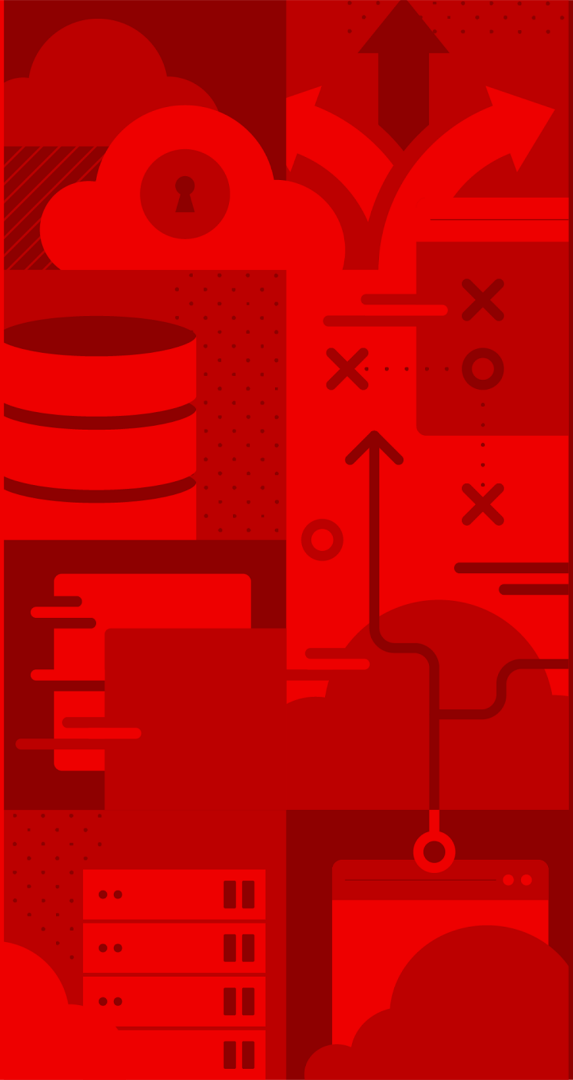
Accessing and understanding CodeReady Workspaces

Technical Exercise Introduction

Introduction to the technical exercise and what is to be accomplished

Technical Hands-On Exercises

- Jenkins
- Nexus
- Gitlab
- ArgoCD
- OpenShift / HELM
- Eclipse Che (CodeReady Workspace)

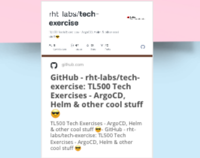


Lab Setup & Access

CodeReady Workspace and Environment Setup

Course Material

Repo Tech Exercises - https://github.com/rht-labs/tech-exercise



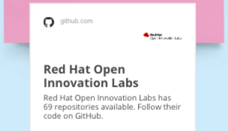
rht labs/**tech-exercise**

github.com

GitHub - rht-labs/tech-exercise: TL500 Tech Exercises - ArgoCD, Helm & other cool stuff 😎

TL500 Tech Exercises - ArgoCD, Helm & other cool stuff 😎- GitHub - rht-labs/tech-exercise: TL500 Tech Exercises - ArgoCD, Helm & other cool stuff 😎

Repo PetBattle - https://github.com/petbattle



github.com

Red Hat Open Innovation Labs

Red Hat Open Innovation Labs has 69 repositories available. Follow their code on GitHub.

Repo Installation OCP - https://github.com/rht-labs/enablement-framework

github.com

rht-labs/enablement-framework

Contribute to rht-labs/enablement-framework development by creating an account on GitHub.

Course slides - https://rht-labs.com/tech-exercise/slides

rht-labs.com

TL500 - Slides TOC

Tech Exercise Instructions - https://rht-labs.com/tech-exercise/#/

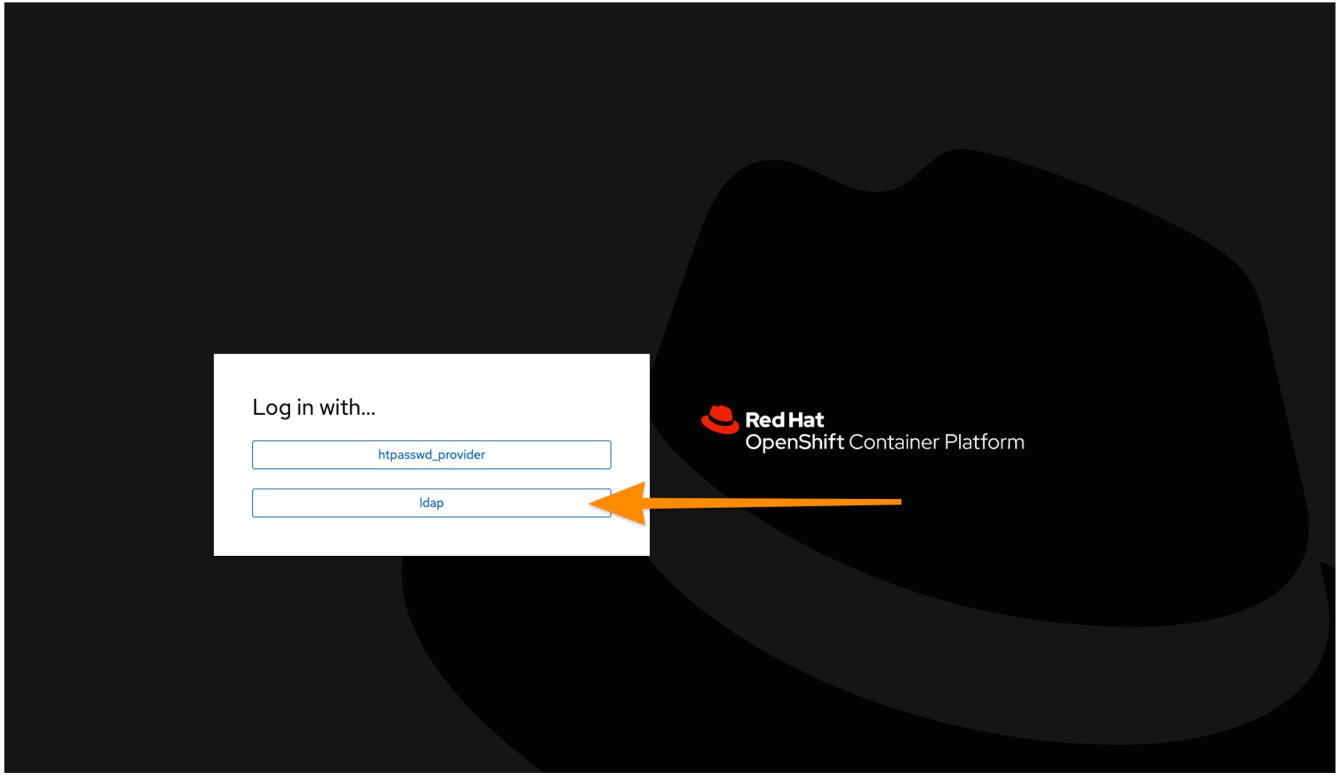
DevOps Culture & Practice (TL500)

CodeReady Workspaces https://codeready-tl500-workspaces.apps.clp-1.ls-na.ole.redhat.com/

Openshift Console https://console-openshift-console.apps.clp-1.ls-na.ole.redhat.com/

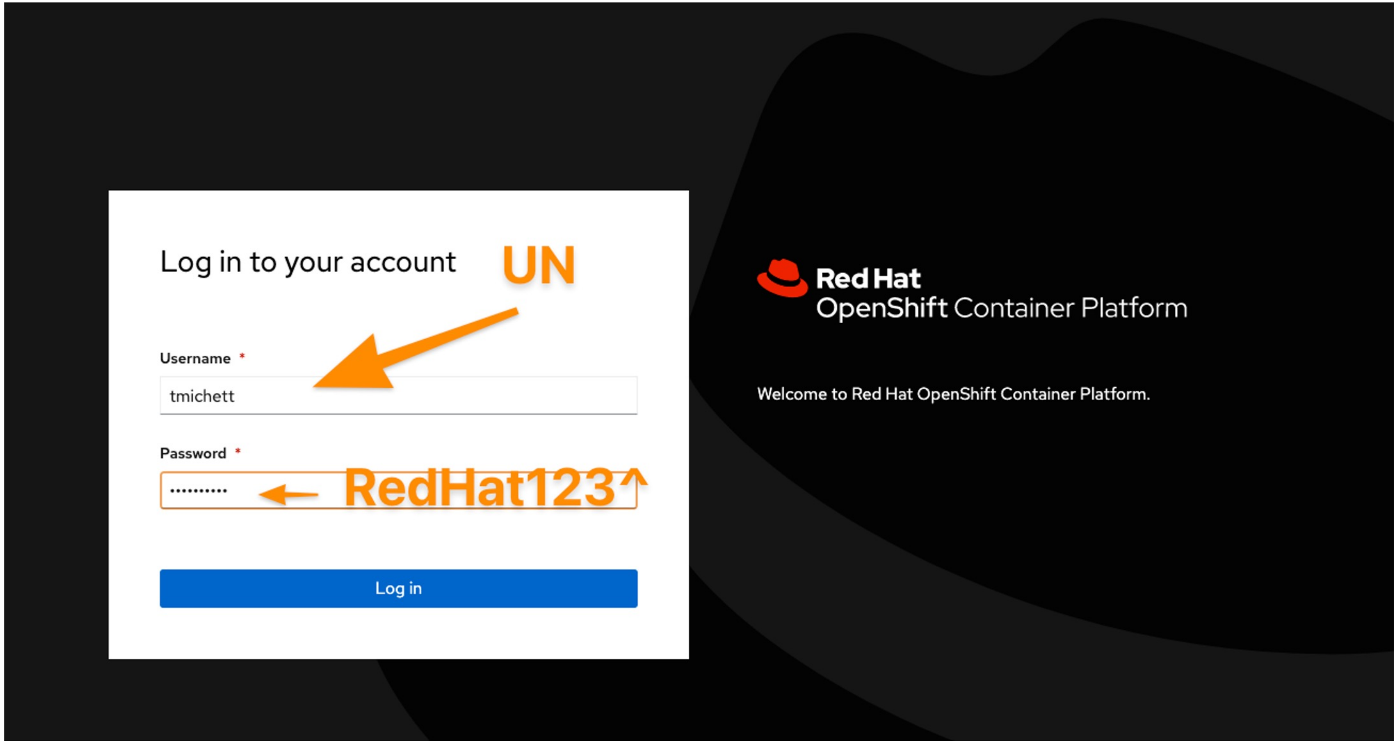
Gitlab https://gitlab-ce.apps.clp-1.ls-na.ole.redhat.com/

CodeReady Workspace Setup



Log in with...

CodeReady Workspace Setup Cont.



Log in to your account

UN

Username *

tmichett

Password *

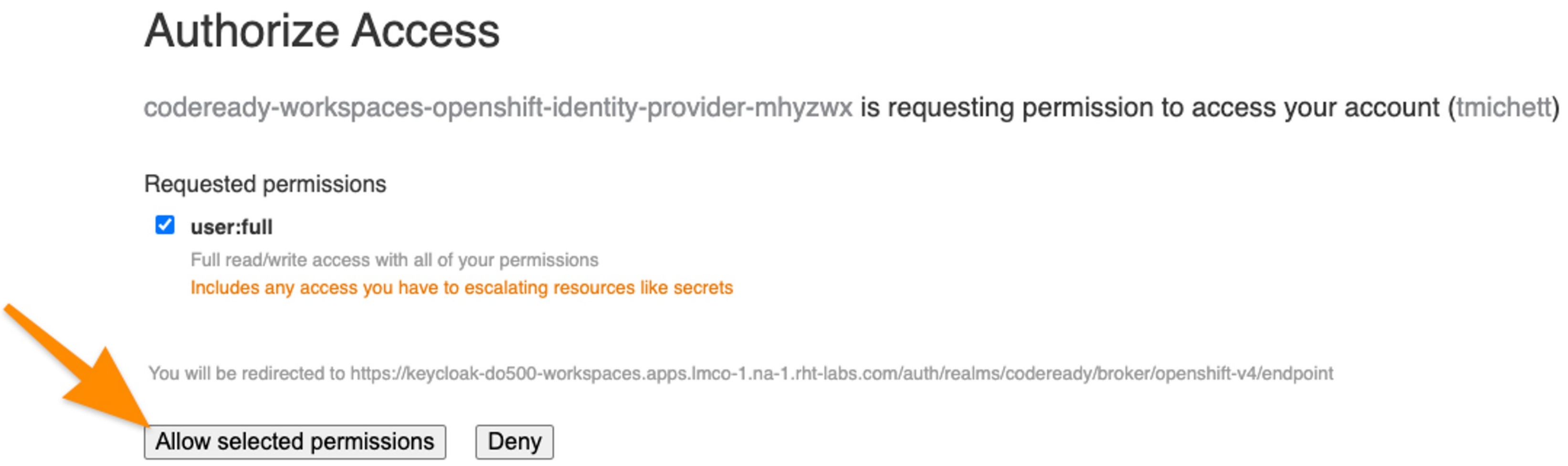
••••••••••

RedHat123^

Red Hat OpenShift Container Platform

Welcome to Red Hat OpenShift Container Platform.

CodeReady Workspace Setup Cont.



Authorize Access

codeready-workspaces-openshift-identity-provider-mhyzwx is requesting permission to access your account (tmichett)

Requested permissions

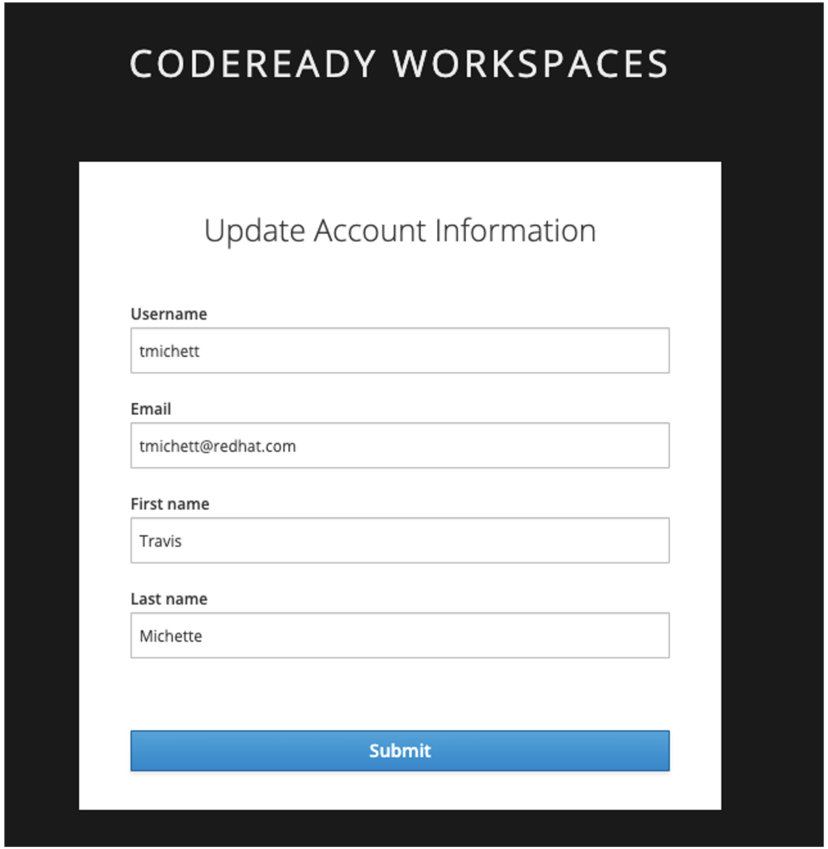
☑ **user:full**

Full read/write access with all of your permissions

Includes any access you have to escalating resources like secrets

You will be redirected to https://keycloak-do500-workspaces.apps.lmco-1.na-1.rht-labs.com/auth/realms/codeready/broker/openshift-v4/endpoint

CodeReady Workspace Setup Cont.



CODEREADY WORKSPACES

Update Account Information

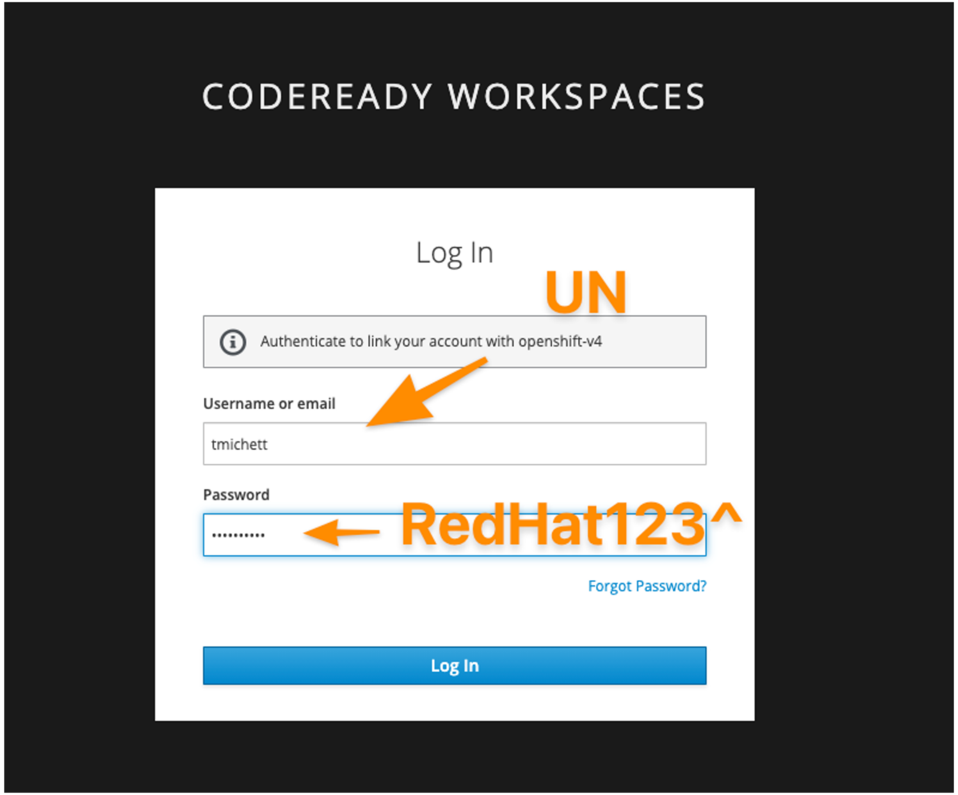
Username: tmichett

Email: tmichett@redhat.com

First name: Travis

Last name: Michette

CodeReady Workspace Setup Cont.



CODEREADY WORKSPACES

Log In

UN

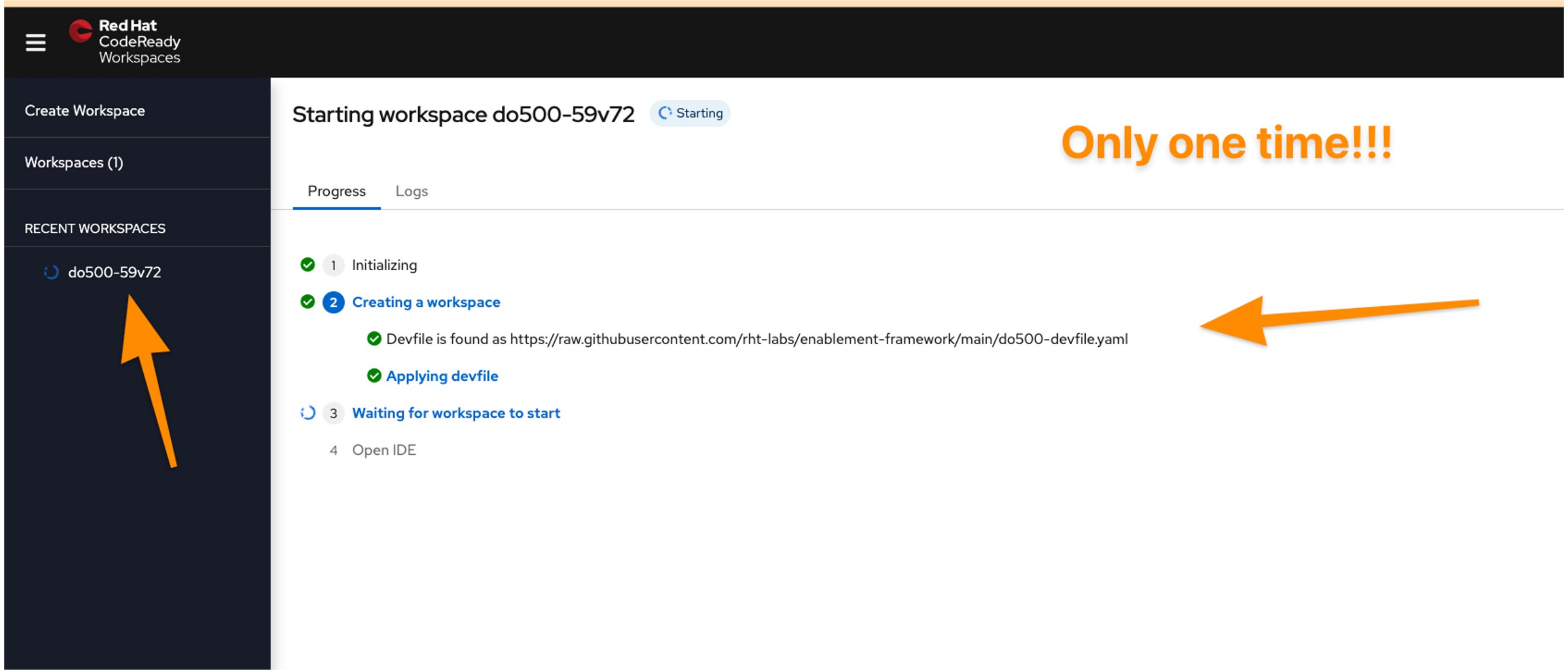
ⓘ Authenticate to link your account with openshift-v4

Username or email: tmichett

Password: ••••••••••

RedHat123^

CodeReady Workspace Setup Cont.



Starting workspace do500-59v72 Starting

✓ 1 Initializing

✓ 2 Creating a workspace

✓ Devfile is found as https://raw.githubusercontent.com/rht-labs/enablement-framework/main/do500-devfile.yaml

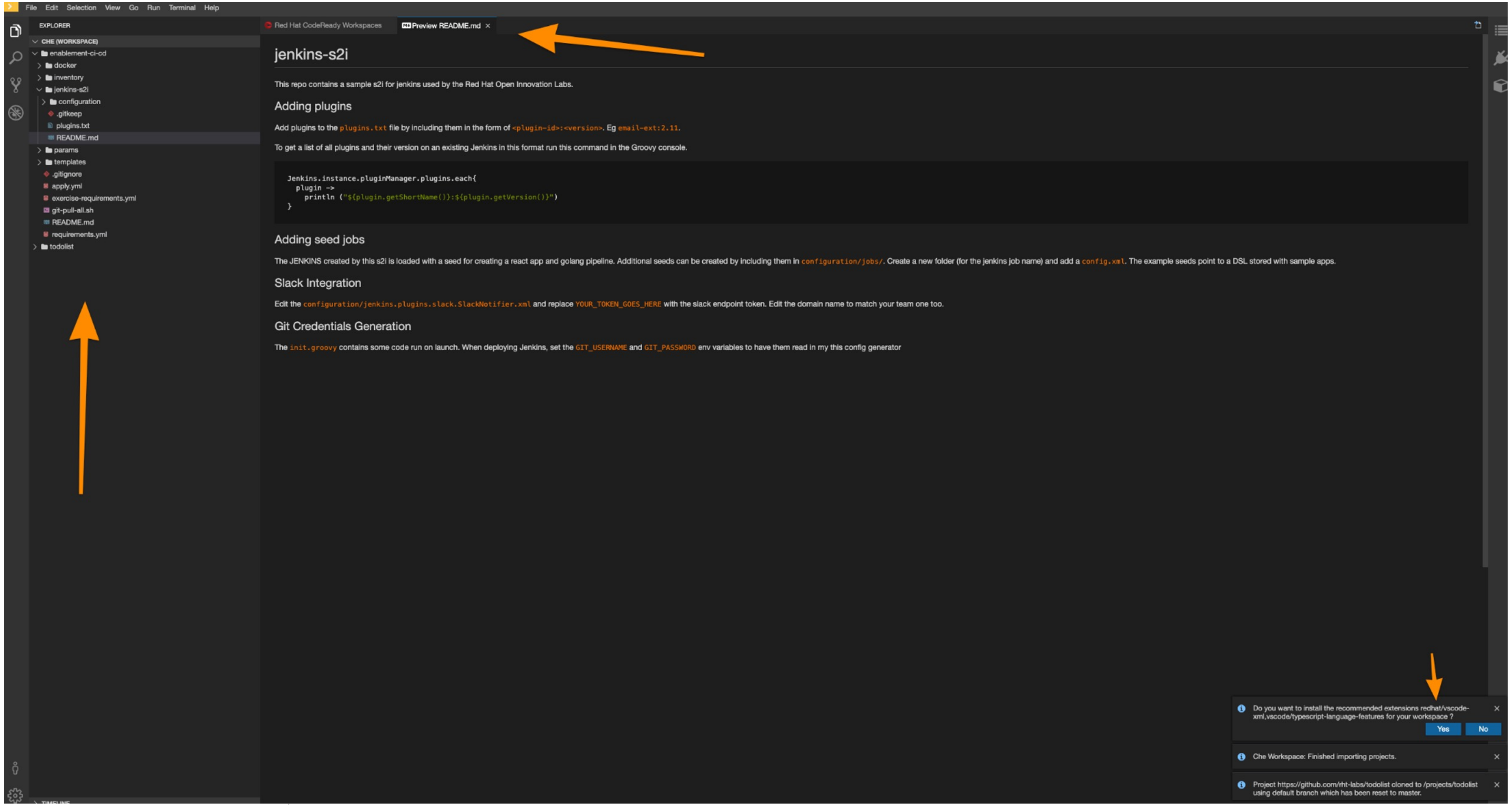
✓ Applying devfile

3 Waiting for workspace to start

4 Open IDE

Only one time!!!

CodeReady Workspace Setup Completed



jenkins-s2i

This repo contains a sample s2i for jenkins used by the Red Hat Open Innovation Labs.

Adding plugins

Add plugins to the `plugins.txt` file by including them in the form of `<plugin-id>:<version>`. Eg `email-ext:2.11`.

To get a list of all plugins and their version on an existing Jenkins in this format run this command in the Groovy console.

```
Jenkins.instance.pluginManager.plugins.each{
  plugin ->
    println ("${plugin.getShortName()}:${plugin.getVersion()}")
}
```

Adding seed jobs

The JENKINS created by this s2i is loaded with a seed for creating a react app and golang pipeline. Additional seeds can be created by including them in `configuration/jobs/`. Create a new folder (for the jenkins job name) and add a `config.xml`. The example seeds point to a DSL stored with sample apps.

Slack Integration

Edit the `configuration/jenkins.plugins.slack.SlackNotifier.xml` and replace `YOUR_TOKEN_GOES_HERE` with the slack endpoint token. Edit the domain name to match your team one too.

Git Credentials Generation

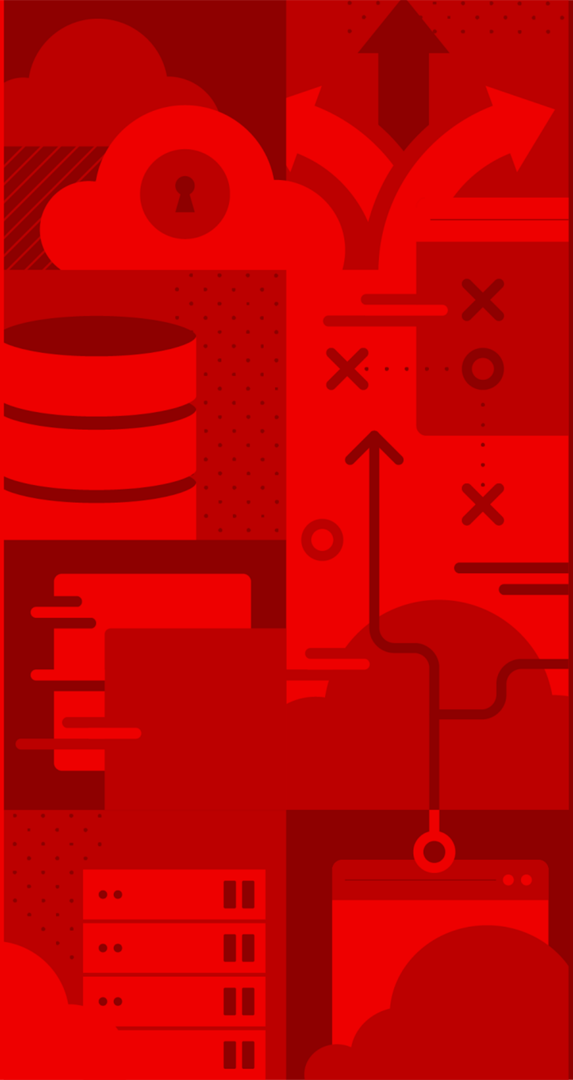
The `init.groovy` contains some code run on launch. When deploying Jenkins, set the `GIT_USERNAME` and `GIT_PASSWORD` env variables to have them read in my this config generator

Do you want to install the recommended extensions redhat/vscode-xml,vscode/typescript-language-features for your workspace ?

Yes No

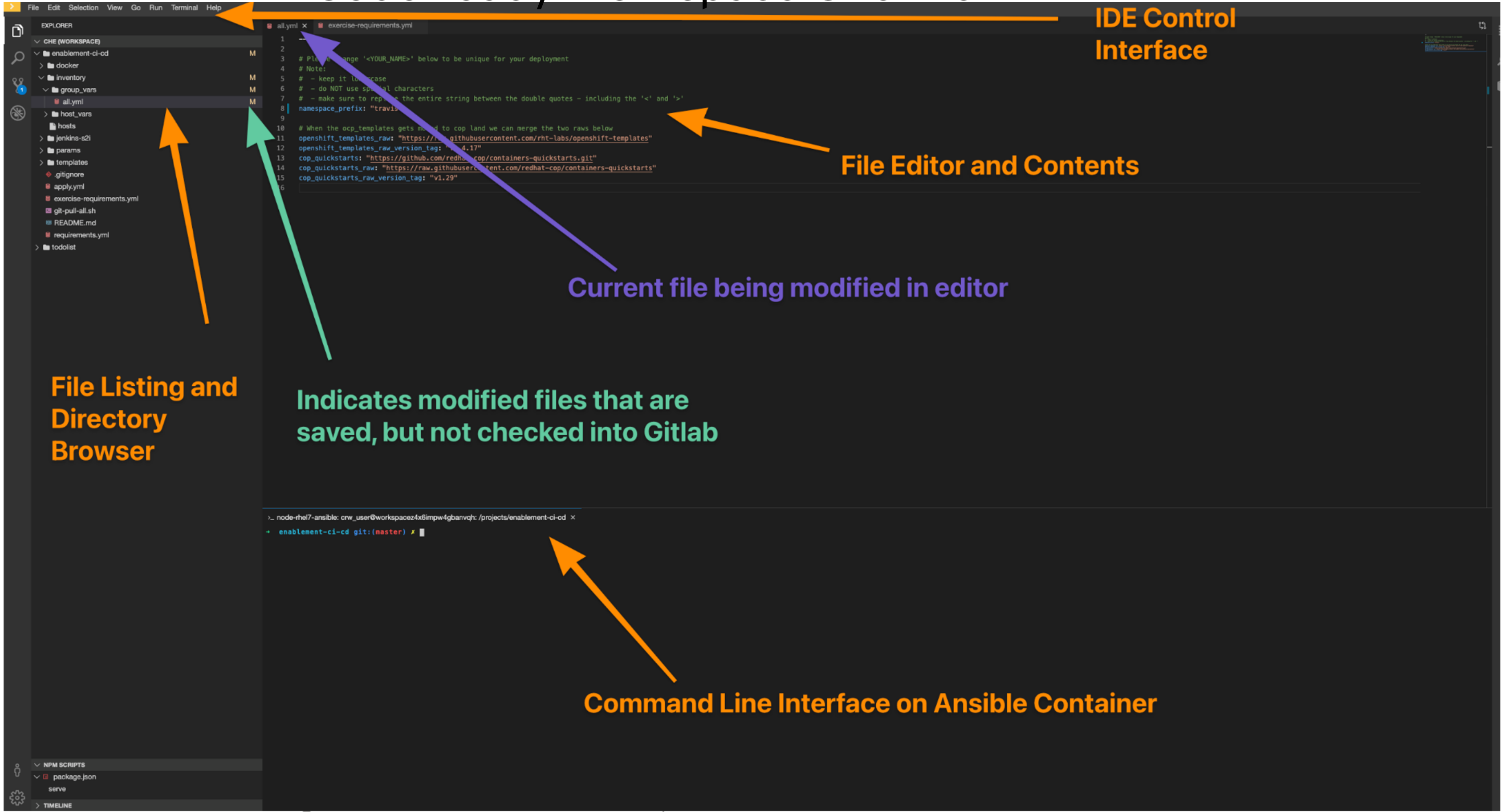
Che Workspace: Finished importing projects.

Project https://github.com/rht-labs/todolist cloned to /projects/todolist using default branch which has been reset to master.



CodeReady Workspace Introduction

CodeReady Workspace Overview



```
---

# Please change '<YOUR_NAME>' below to be unique for your deployment
# Note:
#  - keep it lowercase
#  - do NOT use special characters
#  - make sure to replace the entire string between the double quotes - including the '<' and '>'
namespace_prefix: "travis

# When the ocp_templates gets moved to cop land we can merge the two raws below
openshift_templates_raw: "https://raw.githubusercontent.com/rht-labs/openshift-templates"
openshift_templates_raw_version_tag: "[illegible]4.17"
cop_quickstarts: "https://github.com/redhat-cop/containers-quickstarts.git"
cop_quickstarts_raw: "https://raw.githubusercontent.com/redhat-cop/containers-quickstarts"
cop_quickstarts_raw_version_tag: "v1.29"
```

```
>_ node-rhel7-ansible: crw_user@workspacez4x6impw4gbanvqh: /projects/enablement-ci-cd ×
➜ enablement-ci-cd git:(master) ✗ 
```

IDE Control Interface

File Editor and Contents

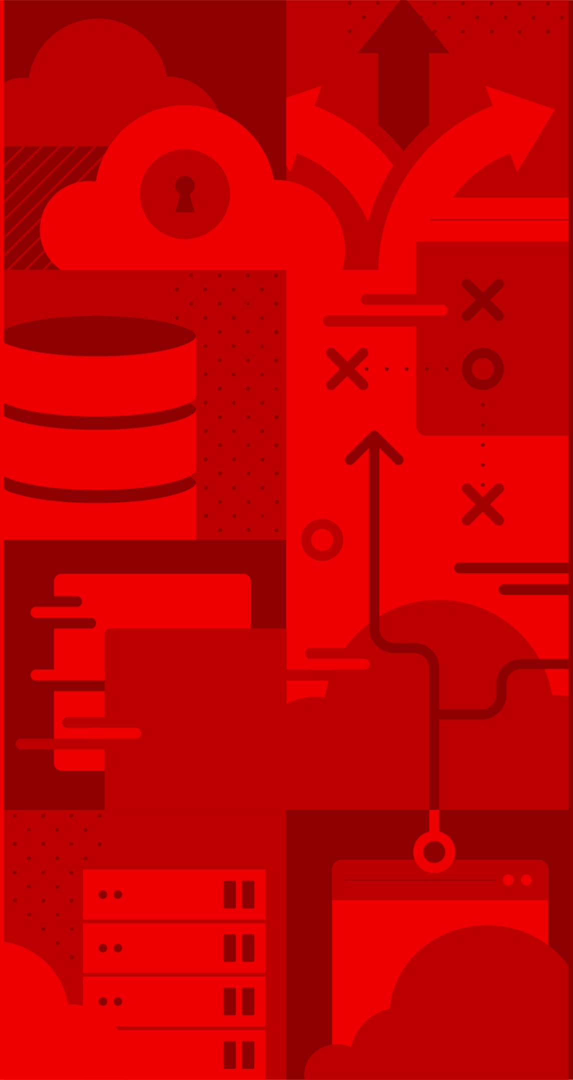
Current file being modified in editor

File Listing and Directory Browser

Indicates modified files that are saved, but not checked into Gitlab

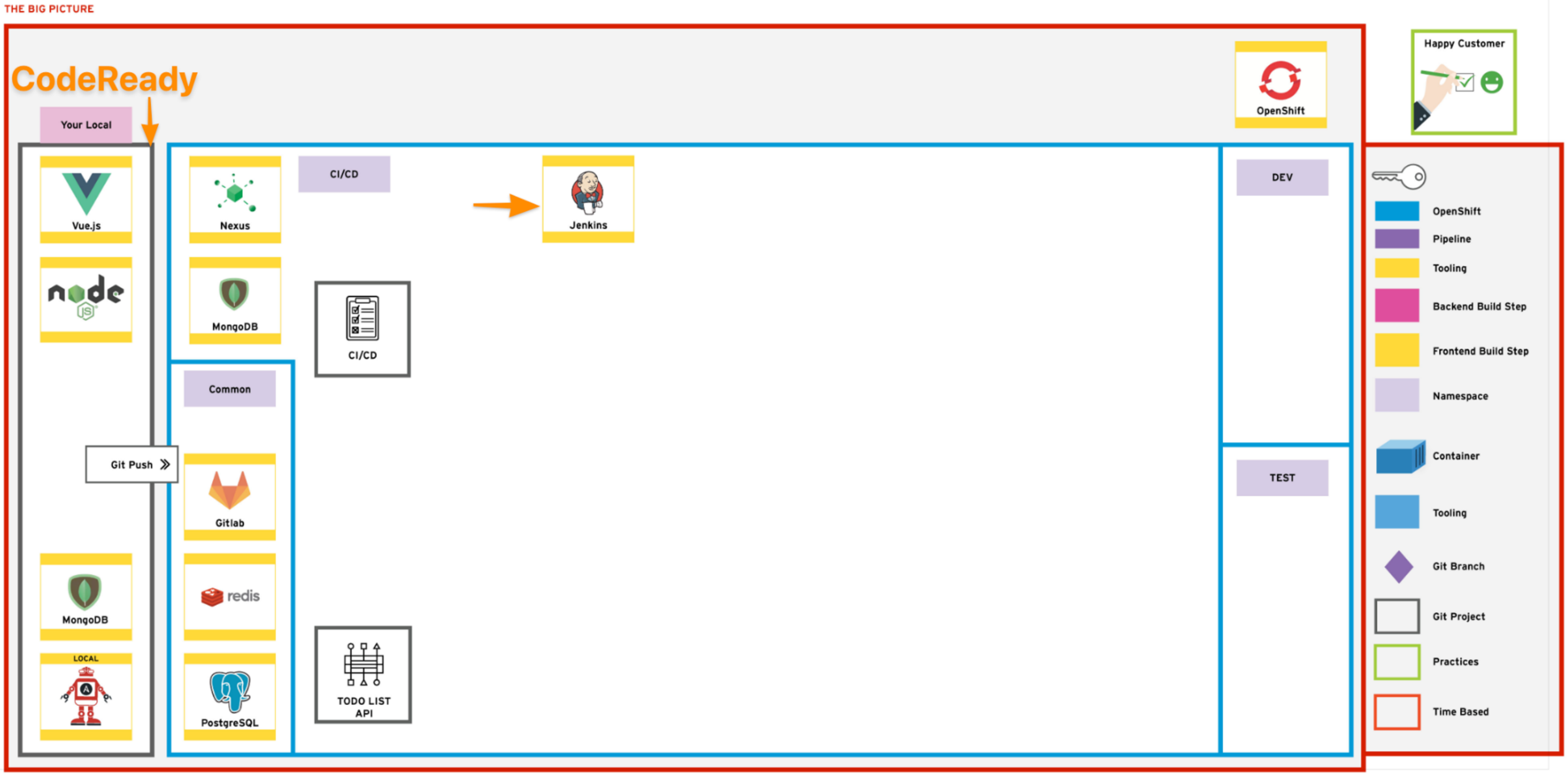
Command Line Interface on Ansible Container

Red Hat



Technical Exercise Introduction

The Big Picture



Exercise Objectives

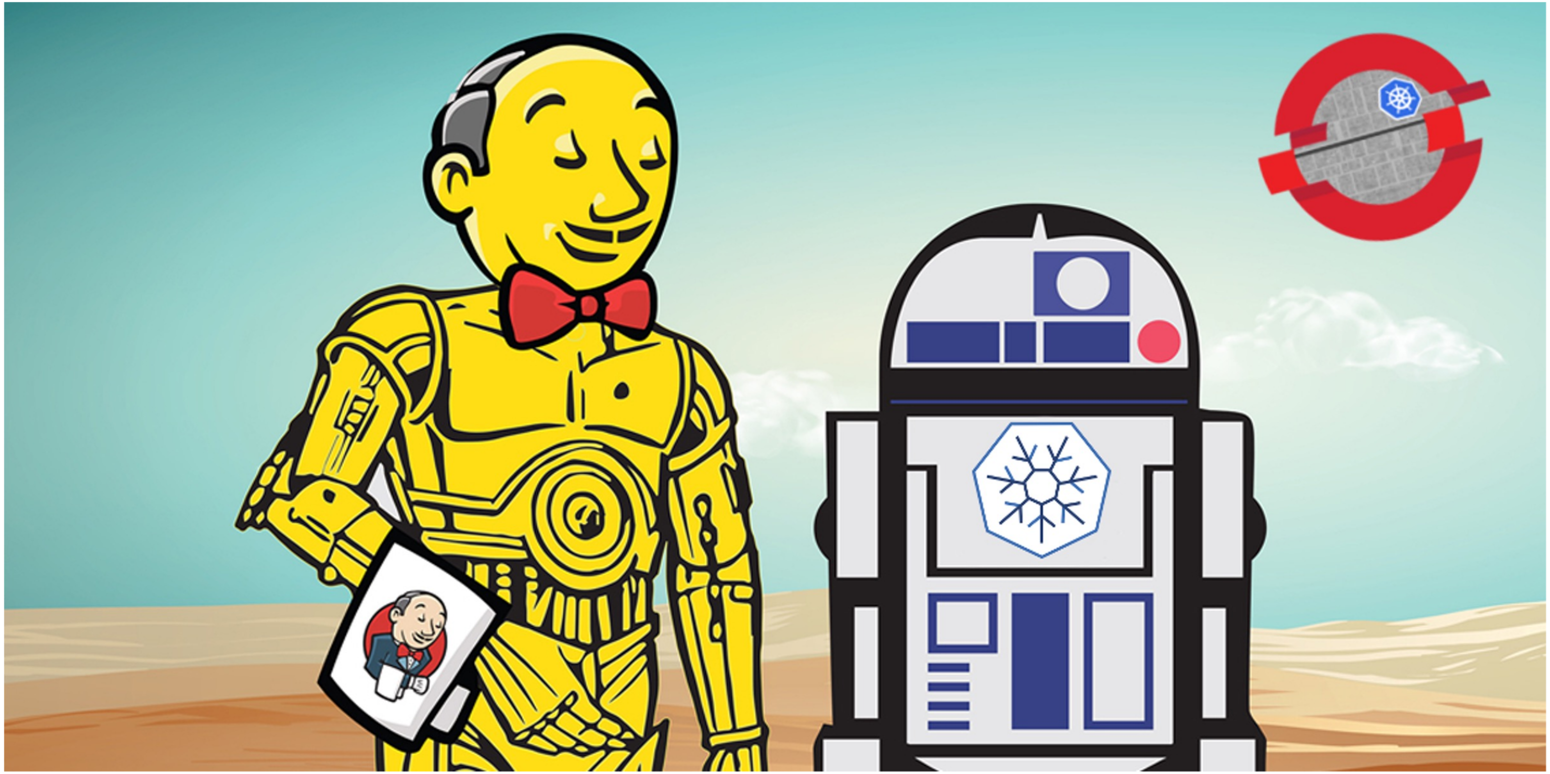
1. Setup CodeReady Workspace and OpenShift environment including user accounts
2. Access CodeReady/Che Workspace for hands-on technical exercises
3. Create initial automation environment with Configuration-as-Code
4. Building initial foundation tooling to facilitate deployment and lifecycle management of a custom application which can be automated
   - ➢ Create Project Namespaces for:
     - ■ CI/CD Tooling
     - ■ Development Environment
     - ■ Test Environment
5. Create an Infrastructure-as-Code environment for a sample DevOps deployment

Tools and Frameworks Used

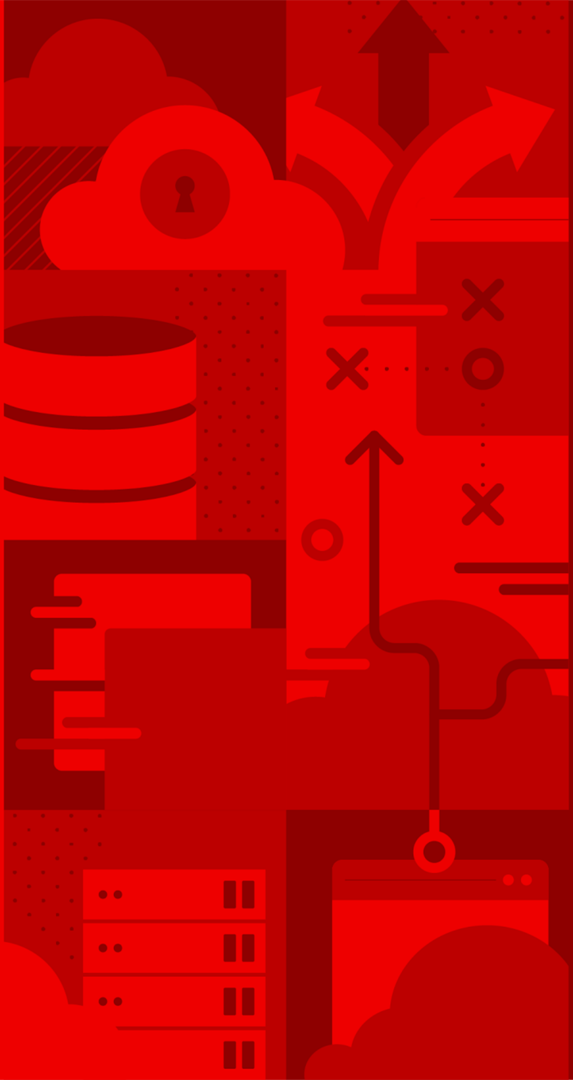
- Openshift Container Platform - Red Hat's private, on-premise cloud application deployment and hosting platform
- GitLab - Community driven Git server now with integrated DevOps Toolchain.
- Nexus - Repository manager for storing lots of application types. Can also host npm and Docker registries.
- Jenkins - OpenSource Build automation server. Highly customisable with plugins.
- Helm - Helps us to define, install, and upgrade Kubernetes application.
- ArgoCD - A controller which continuously monitors application and compare the current state against the desired.

Exercise Steps (10,000’ View)

1. Clone the repo https://github.com/rht-labs/enablement-ci-cd which contains the scaffold of the project. *(Already completed for you as part of CodeReady Workspace)*
2. Create <your-name>-ci-cd, <your-name>-dev and <your-name>-test project namespaces using the inventory and run them with the OpenShift Applier to populate the cluster.
3. Use the templates provided to create build and deployment configs in <your-name>-ci-cd project to be leveraged throughout the rest of the guided exercises.
   a. Nexus
   b. GitLab
   c. Jenkins (using an s2i to pre-configure jenkins)
4. Commit your enablement-ci-cd repository to the GitLab Instance you've created.
5. Burn it all down and re-apply your inventory proving config-as-code works.



[Exercise 1 - Introduction Crawl]



Technical Hands-on Exercises

What is this exercise about?

- This exercise is all about automation and **configuration-as-code**.
- This exercise is aimed at the creation of the supporting tooling that will be used to support the rest of the exercises **using code.**
- The high level goal is to create a collection of project namespaces and populate them with Git, Jenkins & Nexus **using code.**
- As a learner, you will be able to:

  1. Understand the benefits gained from GitOps approach
  2. Deploy helm charts manually
  3. Drive tool installations through GitOps

Why is automation important and how does it help?

- Assurance - Prevents unwanted config changes from people making arbitrary changes to environments. No more Snowflake servers!
- Traceability - Commiting **configuration-as-code** means a user has approved and changes can be tracked.
- Phoenix Server - Burn it all to the ground and bring it back; exactly the way it was!

Technical Hands-On Exercises

Hands-On Activities

- Infrastructure-as-Code (Gitlab)
- Pipelines (Jenkins)
- Containers and Orchestration (OpenShift)
- Container Registries (Nexus)
- Automation (Ansible)
- Local development environment (CodeReady Workspace)
- … and more

Thank you

Red Hat is the world's leading provider of
enterprise open source software solutions.
Award-winning support, training, and consulting
services make
Red Hat a trusted adviser to the Fortune 500.

linkedin.com/company/red-hat

https://www.facebook.com/RedHatServices

https://twitter.com/RedHat_Services

E-Mail: tmichett@redhat.com