

# DevOps Culture and Practice Enablement

## Exercise 1 - Introduction



# 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 Setup

## Tech Exercises Links!!

[Exercise docs!](#)

[OpenShift Console.](#)

[Code Ready Workspaces](#)

[Gitlab](#)

## Course Material

Repo  
CI/CD

   
rht-labs/enablenment-  
ci-cd  
A collection of scripts designed by RHT Labs to help you get started with OpenShift CI/CD. This repository contains the scripts and documentation for the CI/CD setup.

Repo  
Todolist

   
rht-labs/todolist  
A collection of scripts designed by RHT Labs to help you get started with OpenShift. This repository contains the scripts and documentation for the OpenShift setup.

Repo  
Docs

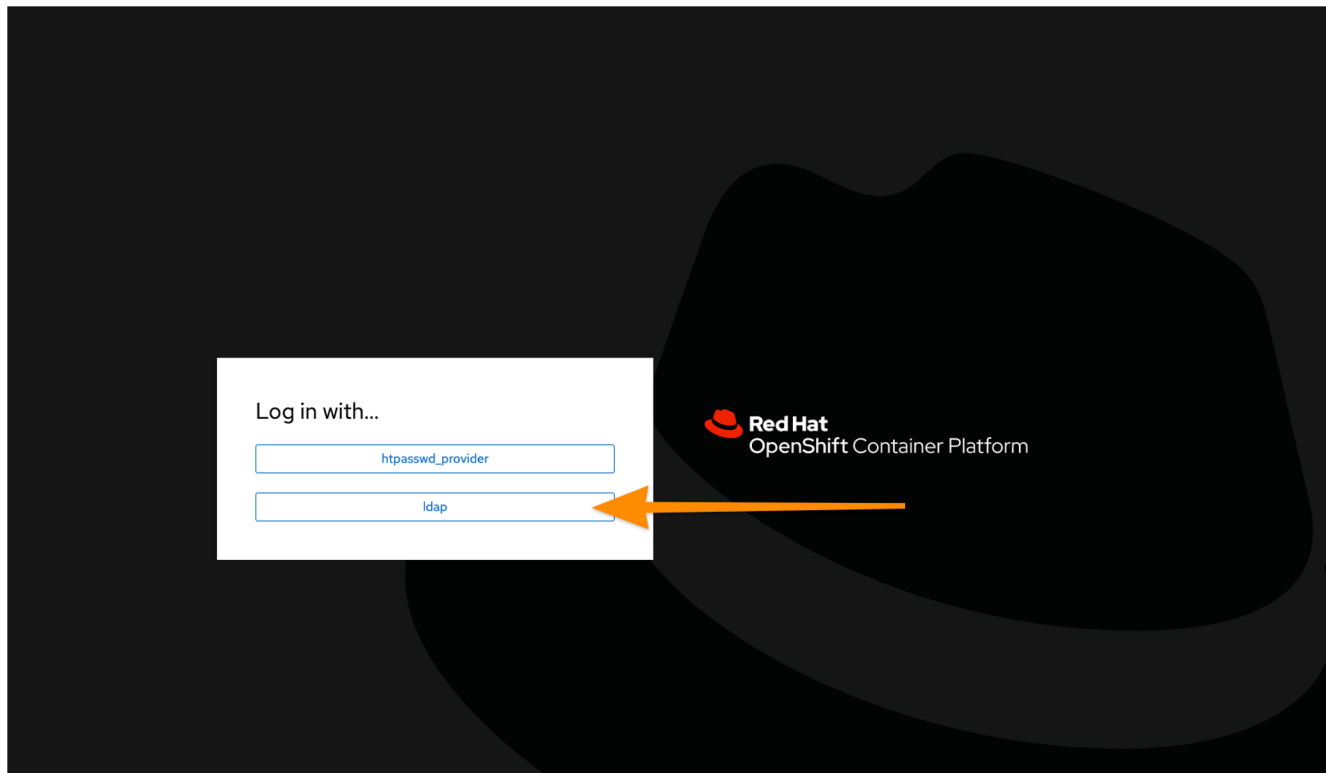
   
rht-labs/enablenment-  
docs  
A collection of scripts designed by RHT Labs to help you get started with OpenShift. This repository contains the scripts and documentation for the OpenShift setup.

Repo  
Installation  
OCP

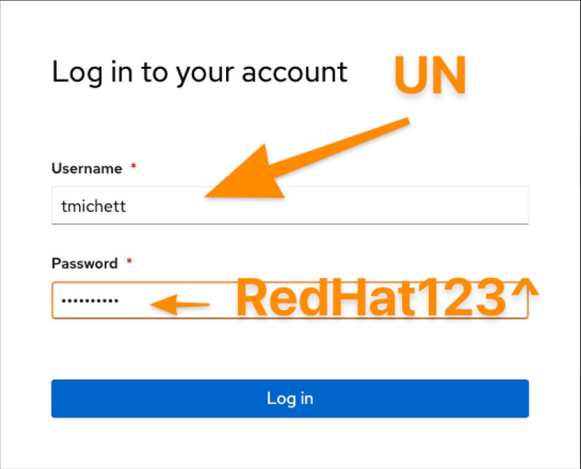
   
rht-labs/enablenment-  
framework  
A collection of scripts designed by RHT Labs to help you get started with OpenShift. This repository contains the scripts and documentation for the OpenShift setup.



# CodeReady Workspace Setup



# CodeReady Workspace Setup Cont.



The screenshot shows the login interface for the Red Hat OpenShift Container Platform. It features a white login box on a dark background. The text 'Log in to your account' is at the top left of the box. To its right, the letters 'UN' are written in large orange font. Below this, the 'Username' field is labeled with a red asterisk and contains the text 'tmichett'. An orange arrow points from the 'UN' text to this field. The 'Password' field is also labeled with a red asterisk and contains a series of dots. An orange arrow points from the text 'RedHat123' (written in large orange font next to the field) to the password field. Below the password field is a blue 'Log in' button. To the right of the login box, the Red Hat logo and 'OpenShift Container Platform' text are displayed, followed by the welcome message 'Welcome to Red Hat OpenShift Container Platform.'

Log in to your account **UN**

Username \*  
tmichett

Password \*  
..... **RedHat123**

Log in

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>

Allow selected permissions

Deny



# CodeReady Workspace Setup Cont.

## CODEREADY WORKSPACES

### Update Account Information

**Username**

**Email**

**First name**

**Last name**

**Submit**






# CodeReady Workspace Setup Cont.

CODEREADY WORKSPACES

Log In

UN

 Authenticate to link your account with openshift-v4

Username or email

tmichett

Password

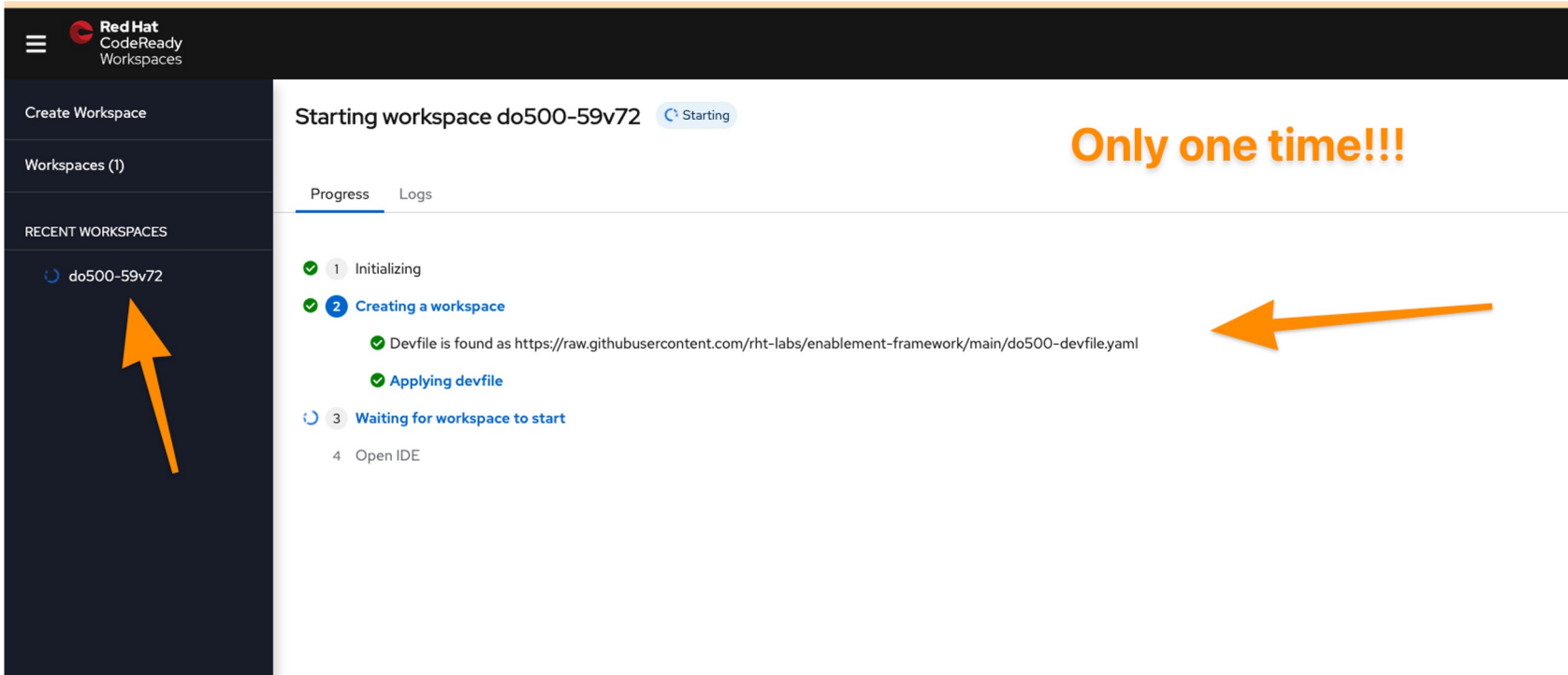
RedHat123^

[Forgot Password?](#)

Log In



# CodeReady Workspace Setup Cont.



The screenshot displays the Red Hat CodeReady Workspaces interface. On the left sidebar, under 'RECENT WORKSPACES', the workspace 'do500-59v72' is listed with a refresh icon. An orange arrow points to this entry. The main panel shows the 'Starting workspace do500-59v72' status with a 'Starting' button. Below this, a progress bar indicates the steps: 1. Initializing, 2. Creating a workspace, 3. Waiting for workspace to start, and 4. Open IDE. Step 2 is currently active and highlighted with a green checkmark. An orange arrow points to the details of step 2, which include: 'Devfile is found as <https://raw.githubusercontent.com/rht-labs/enablement-framework/main/do500-devfile.yaml>' and 'Applying devfile'. The text 'Only one time!!!' is written in orange above the progress bar.

Red Hat CodeReady Workspaces

Create Workspace

Workspaces (1)

RECENT WORKSPACES

do500-59v72

Starting workspace do500-59v72 Starting

Progress Logs

1 Initializing

2 Creating a workspace

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

Applying devfile

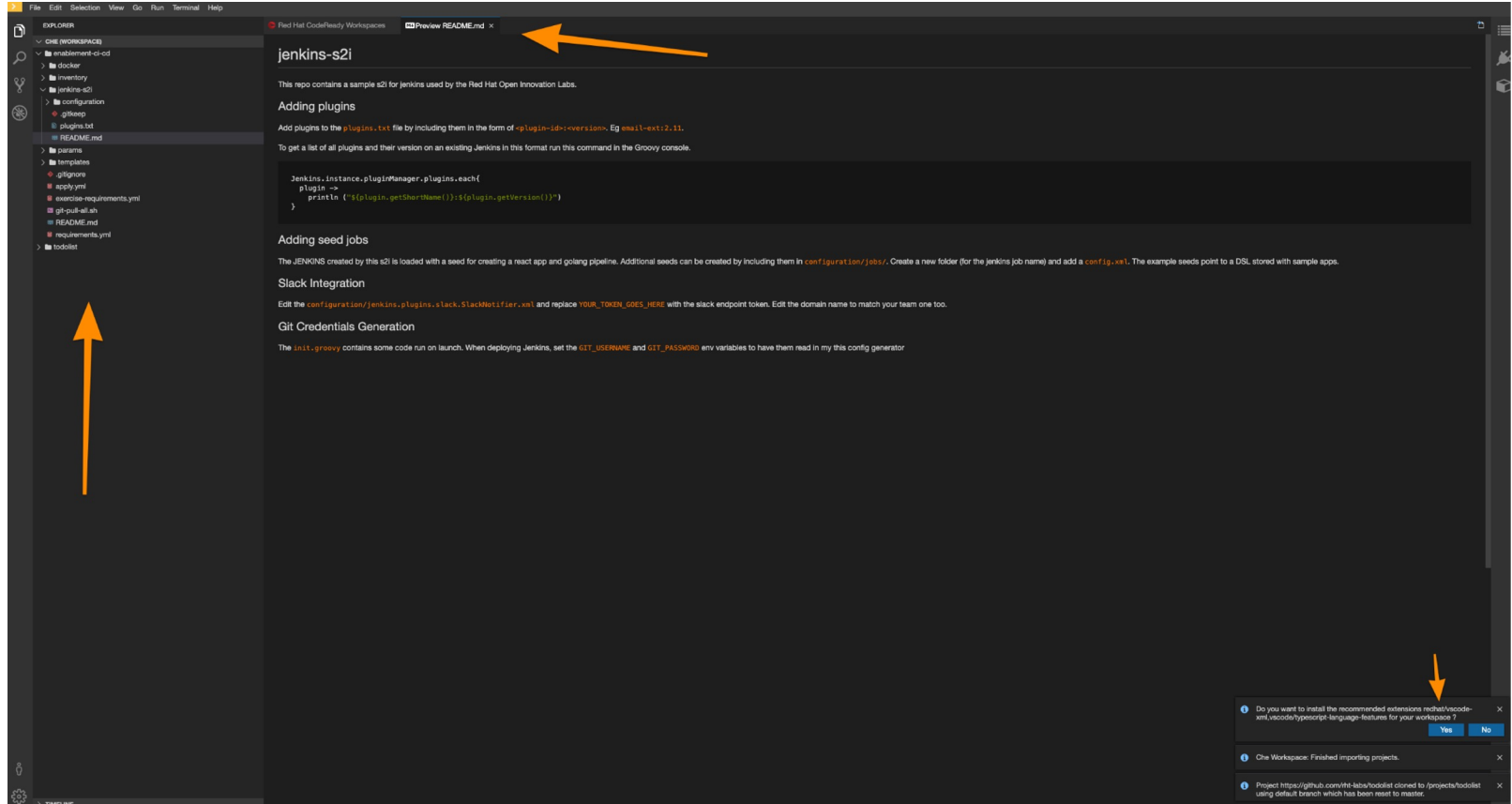
3 Waiting for workspace to start

4 Open IDE

Only one time!!!



# CodeReady Workspace Setup Completed



# CodeReady Workspace Introduction

# CodeReady Workspace Overview

The screenshot displays the CodeReady Workspace IDE interface. On the left is the Explorer panel showing a file tree with folders like 'ONE (WORKSPACE)', 'enablement-ci-cd', 'docker', 'inventory', 'group\_vars', 'host\_vars', 'hosts', 'jenkins-cd', 'params', 'templates', 'gitignore', 'app.py', 'exercise-requirements.yml', 'git-pull-all.sh', 'README.md', 'requirements.yml', and 'todoist'. The main editor area shows a file named 'exercise-requirements.yml' with YAML content. At the bottom is a terminal window showing a command prompt.

**IDE Control Interface**: Points to the top menu bar (File, Edit, Selection, View, Go, Run, Terminal, Help).

**File Editor and Contents**: Points to the main editor area displaying the 'exercise-requirements.yml' file.

**Current file being modified in editor**: Points to the 'exercise-requirements.yml' file in the Explorer panel.

**File Listing and Directory Browser**: Points to the Explorer panel on the left.

**Indicates modified files that are saved, but not checked into Gitlab**: Points to the 'M' (modified) icon next to the 'exercise-requirements.yml' file in the Explorer panel.

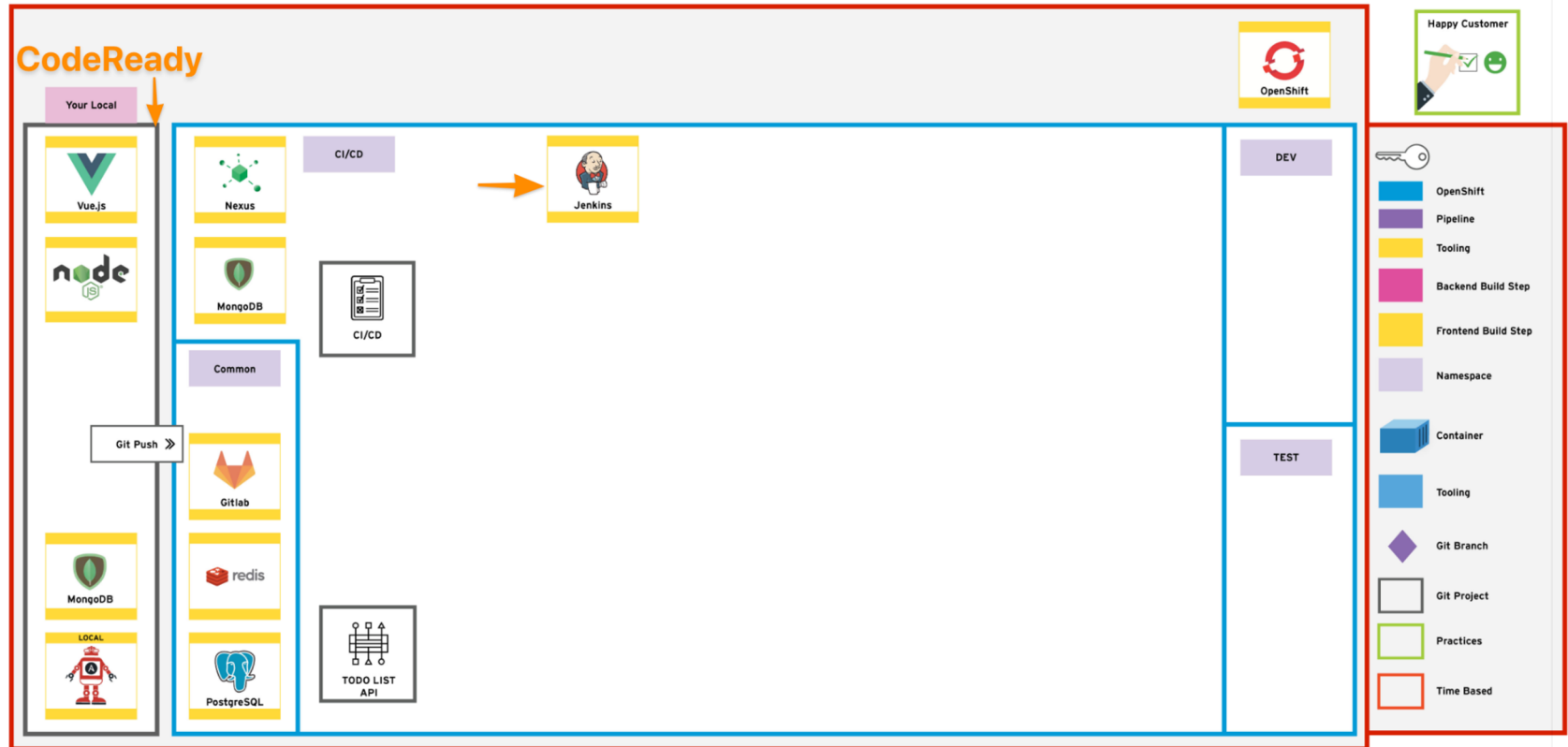
**Command Line Interface on Ansible Container**: Points to the terminal window at the bottom showing the command prompt: `node-rhel7-ansible: cwe_user@workspace4x8mpw4gburgh: /projects/enabement-ci-cd x`.



# Technical Exercise Introduction

# The Big Picture

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

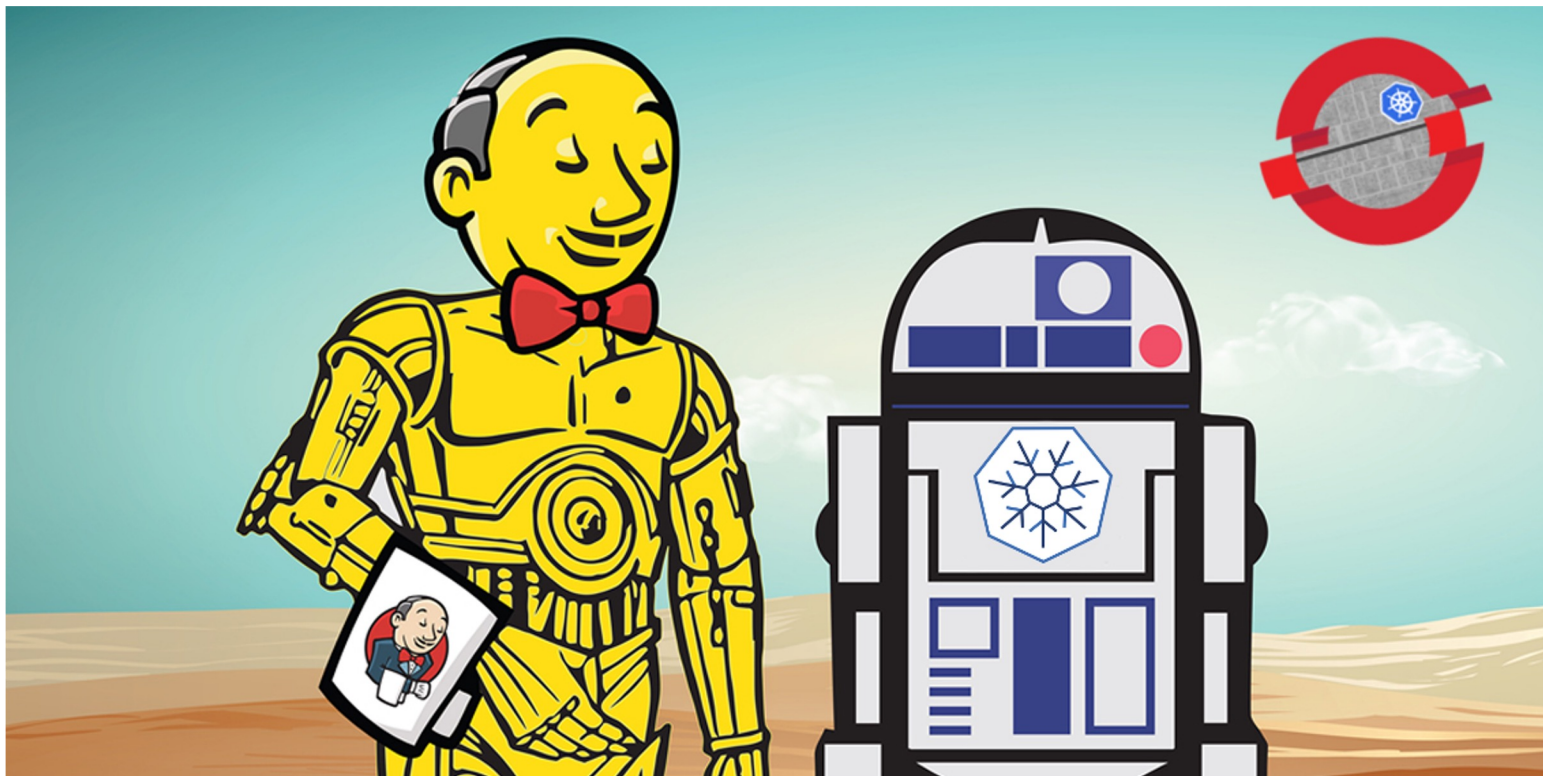
- [Openshift Container Platform](#) - Red Hat's private, on-premise cloud application deployment and hosting platform
- [GitLab](#) - Community driven Git server now with integrated DevOps Toolchain.
- [Nexus](#) - Repository manager for storing lots of application types. Can also host npm and Docker 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 - Committing **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.linkedin.com/company/red-hat)



<https://www.facebook.com/RedHatServices>



[https://twitter.com/RedHat\\_Services](https://twitter.com/RedHat_Services)

E-Mail: [tmichett@redhat.com](mailto:tmichett@redhat.com)

