Table of Contents

* [Homework Assignment](https://www.opentlc.com/labs/ocp_advanced_deployment/06_1_Assignment_Lab.html#_homework_assignment)
* [1. Business Use Case](https://www.opentlc.com/labs/ocp_advanced_deployment/06_1_Assignment_Lab.html#_business_use_case)
* [2. POC Requirements](https://www.opentlc.com/labs/ocp_advanced_deployment/06_1_Assignment_Lab.html#_poc_requirements)
  + [2.1. Basic and HA Requirements](https://www.opentlc.com/labs/ocp_advanced_deployment/06_1_Assignment_Lab.html#_basic_and_ha_requirements)
  + [2.2. Environment Configuration](https://www.opentlc.com/labs/ocp_advanced_deployment/06_1_Assignment_Lab.html#_environment_configuration)
  + [2.3. CICD Workflow](https://www.opentlc.com/labs/ocp_advanced_deployment/06_1_Assignment_Lab.html#_cicd_workflow)
  + [2.4. Multitenancy](https://www.opentlc.com/labs/ocp_advanced_deployment/06_1_Assignment_Lab.html#_multitenancy)

Homework Assignment

**Goals**

* Assess hands-on proficiency with Red Hat OpenShift Container Platform Deployment advanced topics
* Complete course leading to **Red Hat Delivery Specialist - Advanced Platform-as-a-Service (PaaS) Administration**accrediation.

**Criteria**

* Assignments should take the average student 30-40 hours to complete
* Assignments are an individual effort
  + Each student completes his or her own assignment without collaboration
* Assignments should simulate a challenge typically encountered in a Red Hat Consulting engagement
  + Assignment requirements are intentionally a bit vague
* Grading is as follows:
  + **20%** : Basic Requirements section
  + **20%** : HA Deployment section
  + **20%** : Environment Configuration section
  + **20%** : CICD Workflow section
  + **20%** : Multitenancy section
* Passing grade: 80%

**Submission - Engagement Journal and Git Repo**

1. Engagement Journal
   * Submit your documentation like you would submit a client engagement journal.
     + Provide sufficient documentation for each section.
     + Sections without documentation will not be graded
     + [Red Hat LMS](https://docs.google.com/document/d/1nxlvAOlSdNs3-y8AkmDjnc8vtCH9rJdI5zbN9deCK50/edit) if you are a Red Hat employee
     + [Red Hat Connect](https://partner.redhat.com/) (partner.redhat.com) if you are a business partner
2. Git Repo
   * The Git repo should be your own work
   * Deployment must run end-to-end without error.
   * Use as much ansible as possible. Shell is acceptable, but discouraged.
   * Use the best tools of the Containers and PaaS Community of Practice, detailed below

1. Business Use Case

You are a consultant assigned to a telecommunications company called MitziCom. MitziCom provides hosting and cloud services to a variety of clients, from medium size companies to enterprise giants.

MitziCom has asked you to lead a 30-40 hour proof-of-concept (POC) using Red Hat OpenShift Container Platform. The purpose of the POC is to determine the feasibility of using Red Hat OpenShift Container Platform as a target for internal and client workloads.

2. POC Requirements

MitziCom management requires that you include all of the items listed in these subsections in your POC.

Most importantly, MitziCom management wants to be able to deploy your work on their infrastructure in an **automated fashion.**

* Create a public github repository with all your work which can be cloned onto a homework bastion host and executed to execute all the steps below:
  + Follow the structure created by the CoP, as displayed here in this example effort: <https://github.com/redhat-cop/openshift-toolkit/tree/master/quota-management>
* Create an ansible inventory file based on the which deploys the desired OpenShift and its components.
* Create a script or process that completes the following:
  + Customizses the ansible inventory file for different hostnames
  + Sets up storage, networking, and other environment configurations
  + Use <https://github.com/redhat-cop/openshift-applier> to automate the creation of OpenShift objects
    - Deploys app and executes the CICD Workflow
      * Example application deployment: <https://github.com/redhat-cop/container-pipelines/tree/master/basic-spring-boot>
    - Creates all OpenShift objects necessary for multitenancy
* Provide instructions for the MitziCom administrator to deploy all the above in a similar environment in a **single** command

2.1. Basic and HA Requirements

* Ability to authenticate at the master console
* Registry has storage attached and working
* Router is configured on each infranode
* PVs of different types are available for users to consume
* Ability to deploy a simple app (**nodejs-mongo-persistent**)
* There are three masters working
* There are three **etcd** instances working
* There is a load balancer to access the masters called loadbalancer.$GUID.$DOMAIN
* There is a load balancer/DNS for both infranodes called \*.apps.$GUID.$DOMAIN
* There are at least two infranodes, labeled env=infra

2.2. Environment Configuration

* NetworkPolicy is configured and working with projects isolated by default (simulate Multitenancy)
* Aggregated logging is configured and working
* Metrics collection is configured and working
* Router and Registry Pods run on Infranodes
* Metrics and Logging components run on Infranodes
* Service Catalog, Template Service Broker, and Ansible Service Broker are all working

2.3. CICD Workflow

* Jenkins pod is running with a persistent volume
* Jenkins deploys **openshift-tasks** app
* Jenkins OpenShift plugin is used to create a CICD workflow
* HPA is configured and working on production deployment of **openshift-tasks**

2.4. Multitenancy

* Multiple Clients (customers) created
  + Clients will be named Alpha Corp and Beta Corp (client=alpha, client=beta), and a "client=common" for unspecified customers.
  + Alpha Corp will have two users, Amy and Andrew
  + Beta Corp will have two users, Brian and Betty
* Dedicated node for each Client
* **admissionControl** plugin sets specific limits per label (client/customer)
* The new project template is modified so that it includes a LimitRange
* The new user template is used to create a user object with the specific label value
* On-boarding new client documentation explains how to create a new client/customer

Build Version: c3147ce9f77191e30b447cc423f2f68a0c40fc03 : Last updated 2018-07-31 01:29:02 EDT