THE SPACE RACE

A DO180 Practice Scenario

Scenario

You've recently been assigned to engagement at Kosmos Galaxy Ballistics, a highly secretive rocket building agency. Consulting hours have been purchased to assist the agency in building out a flagship OpenShift capability, which comes on the back of a failed attempt to get Kubernetes running without Red Hat's assistance. Your new manager Valentina Tereshkova has informed you that they have high expectations after the poorly implemented Kubernetes capability was identified as a contributing cause of the catastrophic failure during a recent simulated launch. The intern previously responsible for cluster administration, Alan Shepard, had begun work to set up an OpenShift cluster. However, Alan has mysteriously disappeared and you are to take his place. Valentina informs you that upper management is determined to forge ahead with a real launch in just one month's time in which Valentina's dog Laika will be sent into orbit. It is thus your responsibility to set up their OpenShift cluster and save Laika's life, it goes without saying, the stakes are astronomical.

Setup

1. Login to your OpenShift cluster
    a. Install instructions for the CLI can be found here: Getting started with the OpenShift CLI, any OpenShift cluster can be used
    b. The easiest method is to log in on your local machine to the cluster provisioned by the DO180 course using the credentials provided
    c. This activity cannot be done in the DO180 lab VM as it requires podman v3.1.1 or later.
2. Fork the repository at https://github.com/rohandry/DO180PracticeAssignment and create a local clone on your machine
3. Run the buildscenario.sh script
    a. To avoid activity spoilers, **dont** open this script unless something is broken
    b. The script can be rerun to reset the environment if you would like to restart the exercise

Challenge Points

Valentina has asked you to perform the following tasks and briefly record your findings and methods for each. She has also requested that given the strange circumstances of Alan's disappearance, you keep your eye out for any clues as to his whereabouts. Place any files you create through this scenario within the **exerciseFiles** directory.

Part A

1. Get a lay of the land, see what containers and images exist
    a. Be sure to note any containers both running and stopped
2. Get rid of any test images or containers
3. Was he completing his training exercises? Determine what changes have been made to the practiceContainer
    a. What scripts have been added
    b. Has he made any other documentation? (Hint: Do some forensics. Stopped containers keep their files until they are removed. Can you access what Alan was working on?)
4. We've noticed that the container running the devsql database stopped recently, do you mind looking to see if you can figure out why the container isn't running? You can find the script used to generate it in 'providedFiles/devSqlLaunch.sh'. Once you've figured out what's wrong, fix the script and restart it
5. One of our new network devs Yuri has asked that you set up an Nginx environment for him. He has the following requirements:
    a. Follow security best practices, none of the following instructions should be done with root privileges
    b. Create a container according to the following specifications
        i. Use the latest Debian version of the docker bitnami image and name it yuriContainer
        ii. Configure port forwarding so he can curl the webpage through localhost (hint: view the image's documentation)
        iii. Create the 'excerciseFiles/yurisFiles/utilities' directory and mount it into the container's /app directory so that yuri has the following utilities when he curls:
            1. *address*/utilities/birthday.html page which prints your birthday
            2. *address*/utilities/metime/yuri.html page which prints 'yuri is cool'
        iv. Launch the container and test the above web pages

    c. With the container running, insert the following files in the 'providedFiles/yurisFiles' directory into Yuri's container:
        i. salesdata.xml
        ii. userreport.pdf
        iii. The entire 'data' directory
    d. Replace the default nginx web page with a fun message and curl it
    e. Save the completed setup as a tar file and as an image
    f. Restart the container from the tar file and image to make sure everything works
6. Alan had begun work on a containerfile, please:
    a. Apply optimization strategies to minimize the image size
    b. Where applicable, ensure all commands implement the Exec form
    c. Build the image tagged as version 1 and push it to your quay.io registry
    d. Pull it, update the version number to 2.0 in the containerfile and push it again, updating the image tag appropriately
7. Create a Containerfile for Yuri that performs actions 5a to 5d
    a. Additionally, give access inside the container to the following commands (hint: documentation)
        i. less
        ii. vim
    b. Test it works

Part B.

For this part of the scenario, we will use many applications found in the https://github.com/RedHatTraining/DO288-apps repository. From here DO288 will refer to this repository.

1. Determine the state of the OpenShift Cluster.
    a. Delete the mytest app in a single command
    b. Delete the test-project project
    c. Create a new project and launch your own version of the app onto OpenShift
        i. The app is found in the 'providedFiles/hello-world-nginx' folder of the https://github.com/rohandry/DO180PracticeAssignment repository
    d. Expose the application and test its route
    e. Update the index file with a personal message and trigger a new build
        i. Check that your change is reflected in the new page by checking the route link
        ii. Double-check this by printing the content of the index file inside the running container

2. Get some practice building an environment; Deploy the following onto your cluster.
    a. build-app from DO288 named buildapp
    b. html-helloworld from DO288 named htmlhelloworld
    c. Your Containerfile from part A7 named yuricontainer
    d. container-build from DO288 named containerbuild
    e. mariadb-persistent template from the OpenShift namespace
        i. Named mariadbpersistent
        ii. All required parameters should be specified
        iii. Where the parameter has a generated value you can set it to an arbitrary valid value
        iv. Where the parameter has a default value set it to this default value
    f. Make your own copy of https://github.com/rohandry/httpd-ex in Github
        i. Named mytemplate
        ii. Remove the route from the provided template and deploy the app via the template
        iii. Expose the application and test the route works
        iv. Update index.html to include only the text, 'Hello World - *personal message*' and trigger a new build
        v. Check the route endpoint again
        vi. Update the template to use a different non-privileged port and trigger a new build
        vii. Map a different non-privileged port on your workstation to the app's port (specified in the previous step) and access it at localhost:HOST_PORT
    g. Use the CLI to scale the app to 3 pods and write a script to write the logs of the application pods to a text file
        i. Ensure this script works without any adjustment after pods have been deleted and remade

Part C.

Well done on making it this far. Upper management is pleased with your work and has decided to give you the honor of launching the rocket. Please run the script 'providedFiles/ countDown.sh' to trigger the launch. The launch code is 190995.

End of exercise

By Rohan Drysdale
rohan@redhat.com

https://www.linkedin.com/in/rohan-drysdale-895744192/