ROITraining
Maximize Your Training Investment

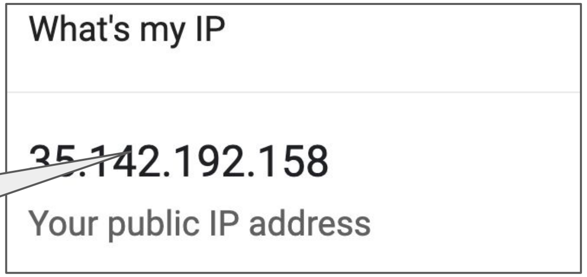
Cheat Sheet: Docker Jenkins

Prerequisites

- To use the Jenkins Docker Image and pipeline, you must have:
  - a kubernetes cluster in your project
  - at least one nodejs deployment already running in the cluster

Creating a Firewall Rule

- Locate your computer's public IP address:
  - Open a new browser window and perform a Google search for "what is my ip"
  - Copy your IP address displayed



What's my IP

35.142.192.158

Your public IP address

Your IP will be different

- Run the following command in your cloud shell to create a firewall rule allowing traffic on port 80 (http) from your IP address
  - IMPORTANT: be sure to replace *PutYourPublicIPHere* with the IP just copied

```
gcloud compute firewall-rules create default-allow-http --direction=INGRESS
--priority=1000 --network=default --action=ALLOW --rules=tcp:80
--source-ranges=PutYourPublicIPHere/32 --target-tags=http-server
```

- If you get an error the firewall rule already exists, delete it and try again:

```
gcloud compute firewall-rules delete default-allow-http
```

Creating The VM

- Create a new Compute Engine instance
    - Give the instance a name that you can recognize as being your Jenkins instance
    - Make it a **small** machine type
    - Select **Deploy a container image to this VM instance**
        - Use `hello-world` for the container
            - This is just to ensure docker is installed on the vm
            - You will ssh in a moment and run a jenkins container in the vm
    - Select **Allow full access to all Cloud APIs**
    - Select **Allow HTTP traffic**

Running the Docker Jenkins

- After the VM starts, click the **SSH** button on the console to SSH into the vm
- Run the following command from the SSH session to run a container we have provided that has jenkins installed (it will take a couple minutes to download):

```
docker run -d -p 80:8080 -p 50000:50000 --name jenkinsdemo -v jenkins_home:/var/jenkins_home drehnstrom/jenkins-gcp:v0.2
```

- When it completes, get the admin password:

```
docker exec jenkinsdemo cat /var/jenkins_home/secrets/initialAdminPassword
```

- If you run the above command and it complains that it cannot find the file, wait a while and try again. The underlying process may not have finished even though you got your command prompt back.
- Copy the Jenkins admin password displayed
  - You will need it in the next section

Jenkins 1 of 3

- In a browser, go to the external IP of the VM
  - Copy the external IP from Compute Engine
  - NOTE: It may take a while to be serving properly - wait a minute and try again if not working yet
- When Jenkins is running, enter the admin password
- Click on “install suggested plugins” and let it finish
- Create an Admin User - you can use any username and password
  - Follow the prompts to complete the setup
- Click “New Item” on the left hand side or click the “create new jobs” link under the Welcome
- Give it a name and select “Pipeline”, click OK
- Continued on next slide

Jenkins 2 of 3

- Create a pipeline using this startpoint and replacing [vars] with your values: https://storage.googleapis.com/deloitte-training/Jenkinsfile
- Use wget on your Cloud Shell to download the above file and then open in the editor and follow the instructions at the top of the file.
    - You do NOT need to change the way the images are being versioned, the use of :2.${env.BUILD_ID} generates a dynamic version number that will auto increment with each Jenkins build
- Copy/Paste to the Pipeline Box (click Pipeline in your Jenkins Window)
- Click Save
- Click Build Now

Jenkins 3 of 3

- To run the pipeline, click **Build Now**
- You will see the stages complete
  - If you get an error, click on the build date in the Build History section on bottom left and then click on **Console output**
  - Review the output and correct any errors
- If the pipeline successfully completes, run the following command in your cloudshell
  - `kubectl get pods`
  - You should see the pod is being replaced