Cloud Native Operations : Kubernetes and Helm

Martin Chabot

Group Exercise #1

Group Warm-up questions GE#1

# Instructions

1. Login as ericsson username and start your Kubernetes Nocentino Lab. These exercises should **ONLY** be done on the Nocentino lab.
2. Be mindful of the words used throughout this document. 😉
3. Ensure you have at least 3 worker nodes running and ensure they are configured to accept the scheduling of pods.
4. Ensure you have no leftovers of resources from previous exercises.
5. Work as a team, help each other’s and try not to do all the exercises on your own.
6. Every question should be discussed with your peers to ensure all the options have been looked at.
7. Some answers will vary between labs. Some answer will be the same. Don’t cut’n paste someone else’ answer, find the answer from your own lab.
8. When you answer a question, your instructor **must** see the prompt of your lab, the command you used, the output it generated and highlight the section you want the instructor grading your exercises to focus on. Don’t let your instructor guess for your answer in a long output. Be specific.
9. Some of the questions may be tricky, sometimes an error output may be a valid answer.

# Example of a good answer:

Question: What ip address the pod busybox1 has?



# Section 0

Your name: Md Rabiul Ahamed Bin Hanif

Your team leader is: Aurunakumari Yedurupaka

# Lab Preparation

Let’s create some application: an eCommerce website that sells some products. It needs catalogues, carts, order managements, a web front-end and few databases in the backend.

Process the following command to deploy your eCommerce application:

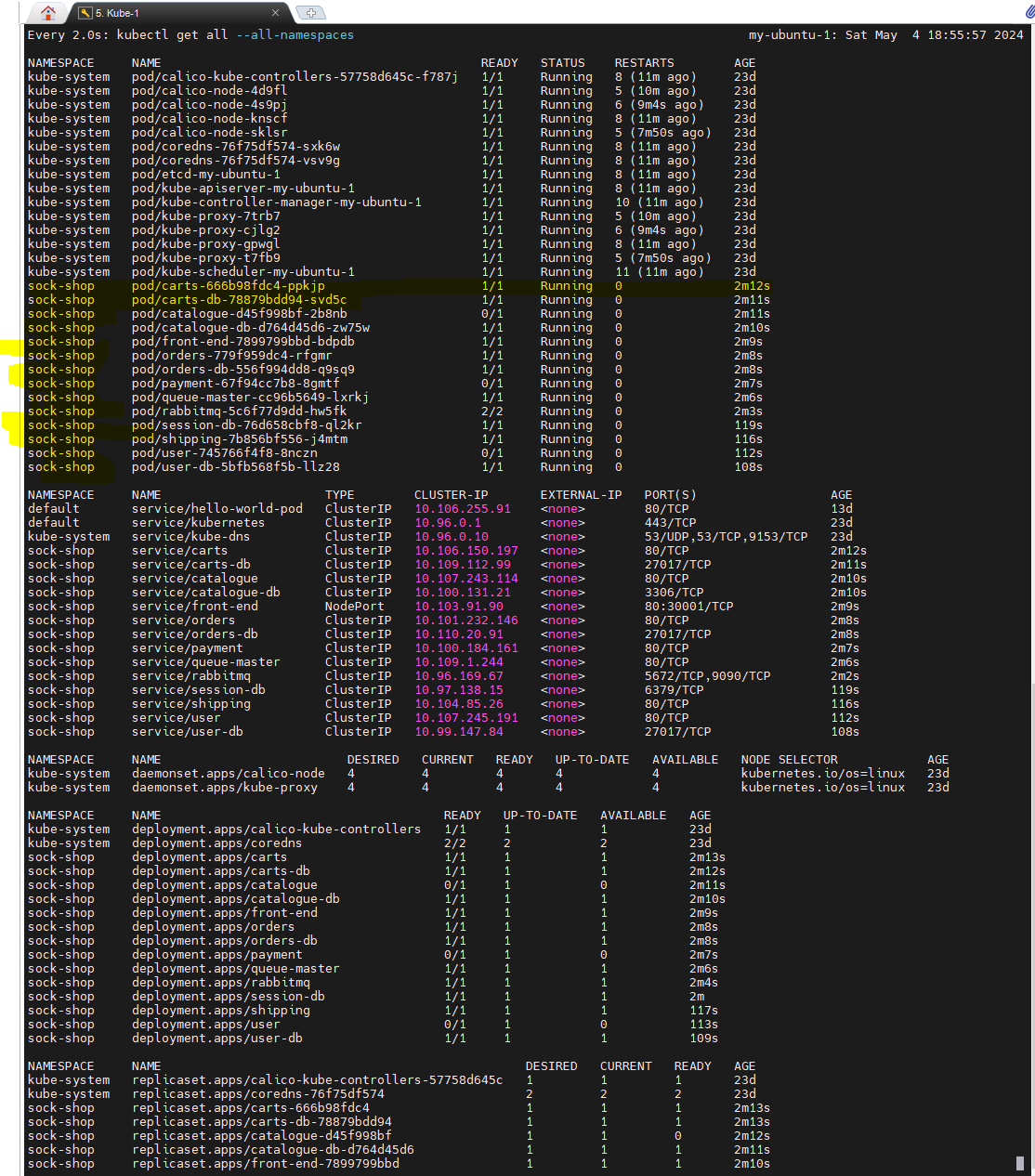
cd; kubectl apply -f https://raw.githubusercontent.com/martin-eric/cn-class/main/martin-sock-shop-complete-demo-2023.yaml

NOTE: You are responsible of the commands you process, understand the above before pressing enter. Maybe you should download a copy locally of the deployment file. It might be useful later on 😉.

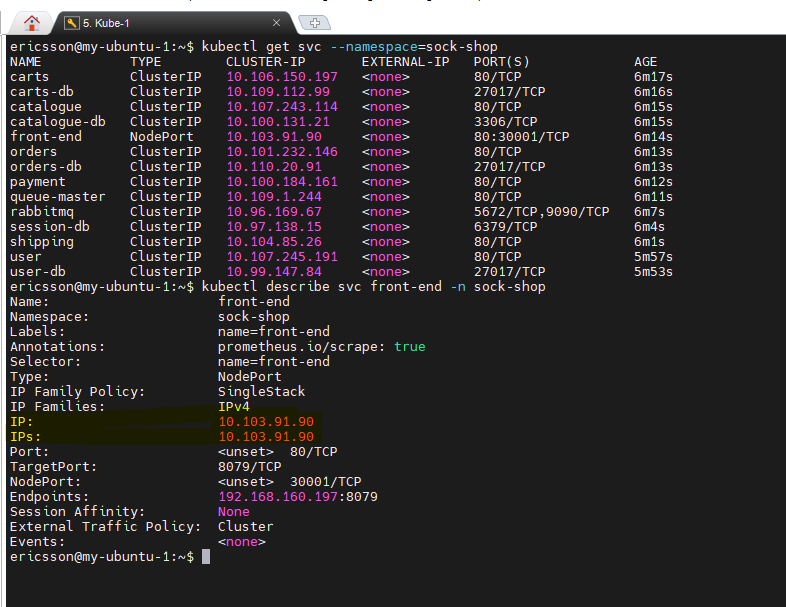
# Section 1

1. How can you confirm that all the pods and services are in a healthy state. Show the commands you used.

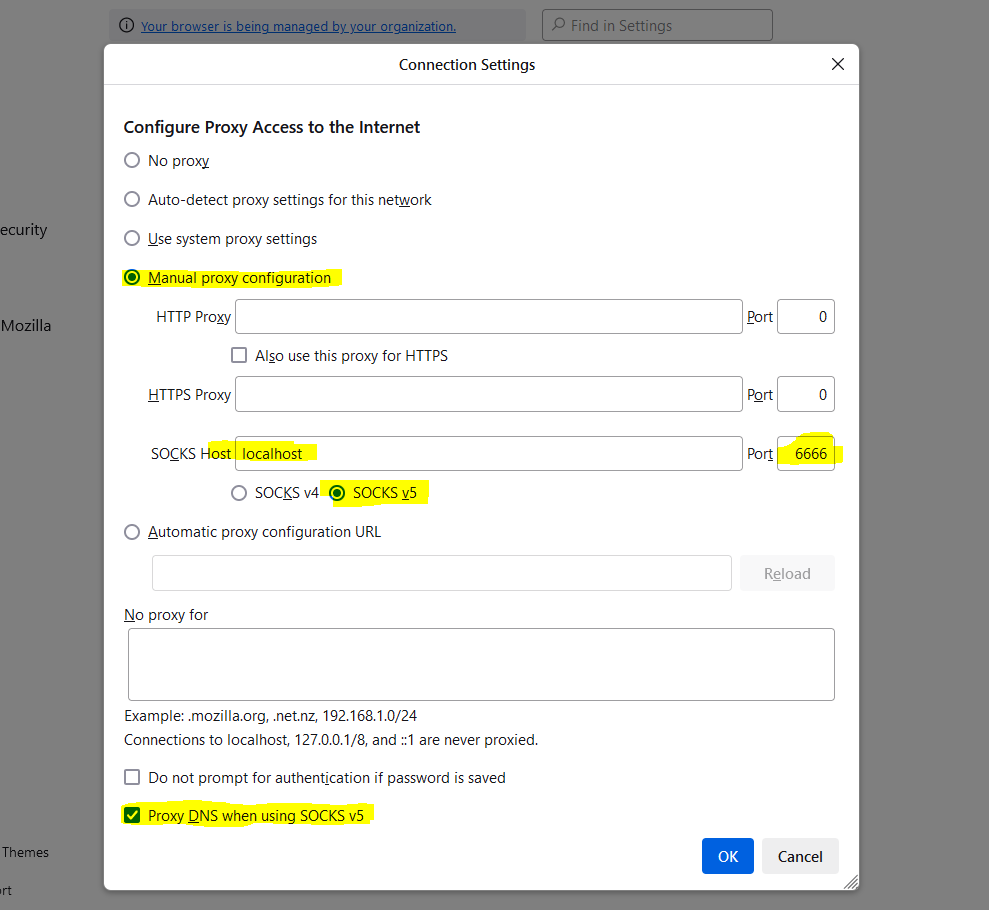
NOTE: You might want to run your kubectl commands every 10-20 seconds. It might take up to 5 mins after the deployment is launched before everything is ready. Provide output when all is in a healthy state.



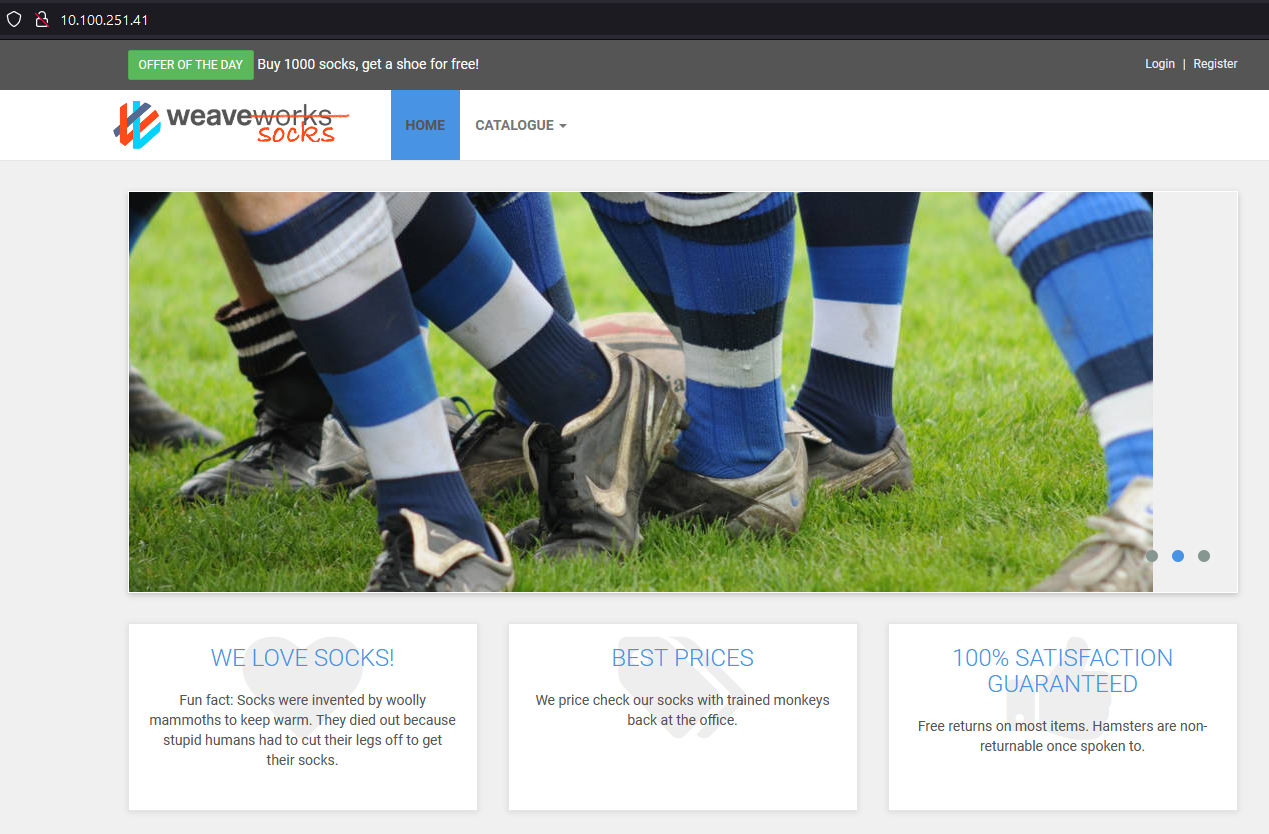
1. After the deployment has successfully completed and all the resources are in a normal state, investigate which ip address you should connect to this e-commerce website. Look for an ip that could be reachable with the standard port for http. Show the commands you used.



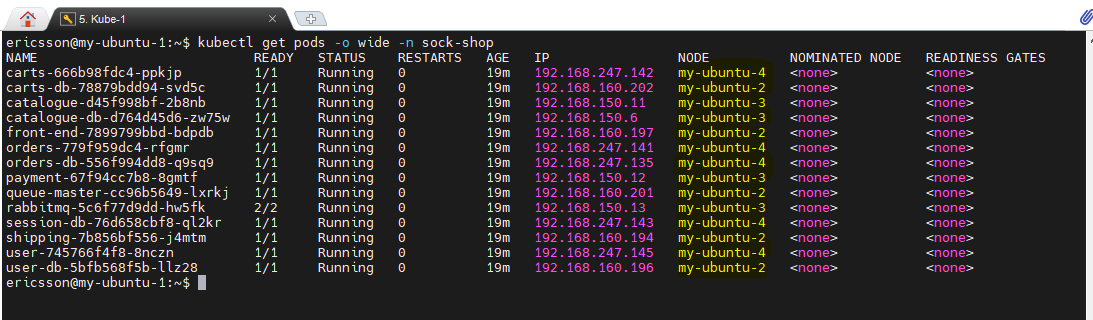
1. When identified, connect your web browser to it and take a snipit of the url you used, port forwarding knowledge may help here 😉. An example from my lab:







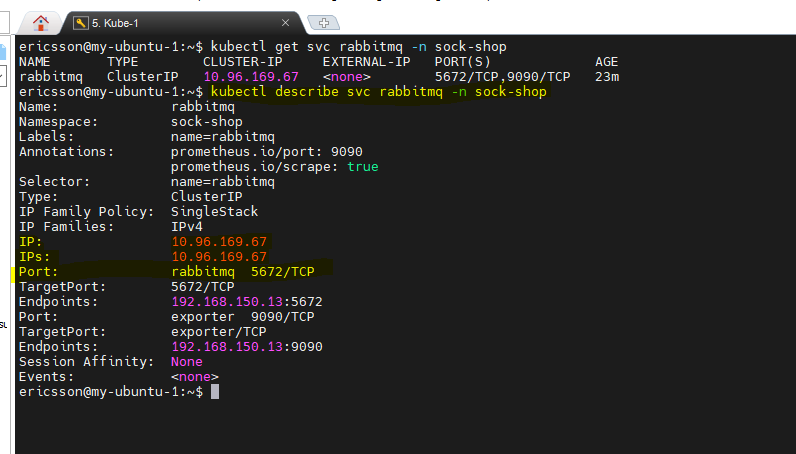
1. Confirm that every worker node has some eCommerce resources scheduled to it. Show your work and output how you determined the answer.



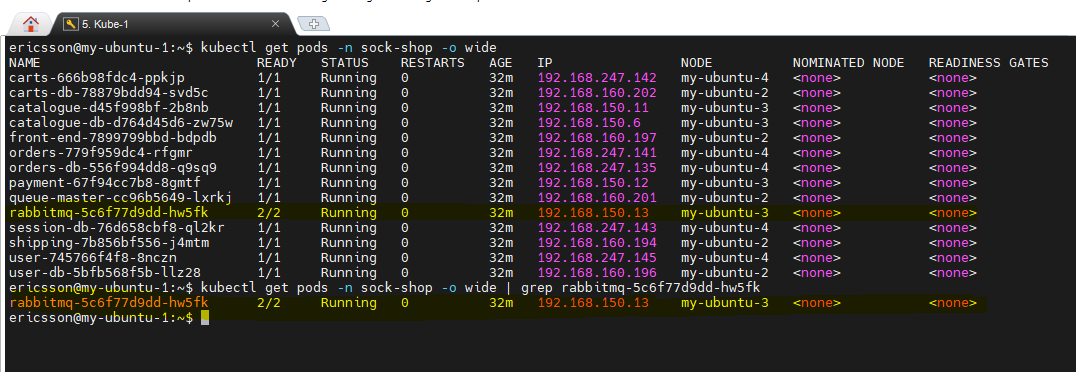
# Section 2

Focus only on the **service** called rabbitmq :

1. What IP and port this **service** owns? Show your work and output how you determined the answer.

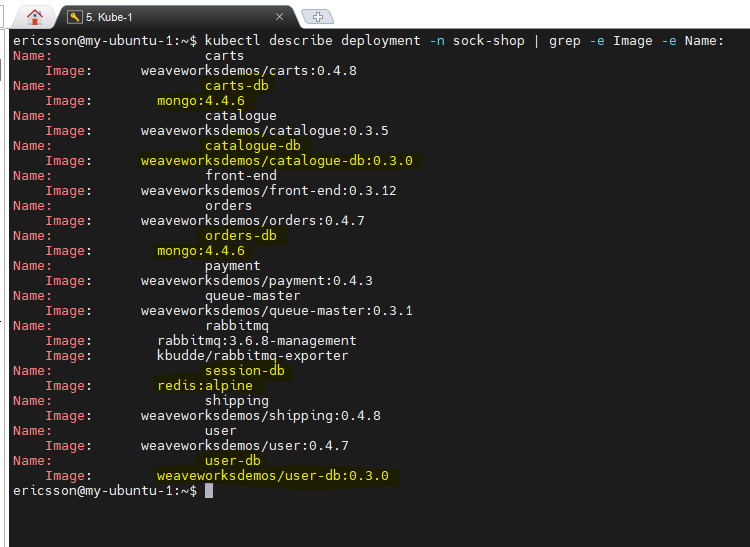


1. Which POD name and IP address and port this service will distribute the traffic to and from? Show your work and output how you determined the answer.

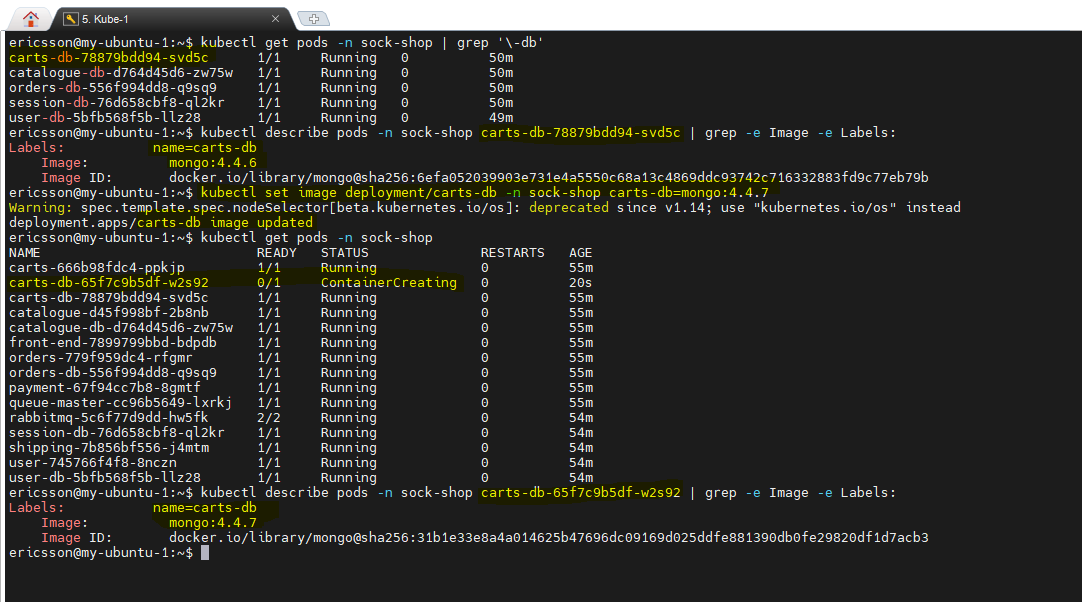


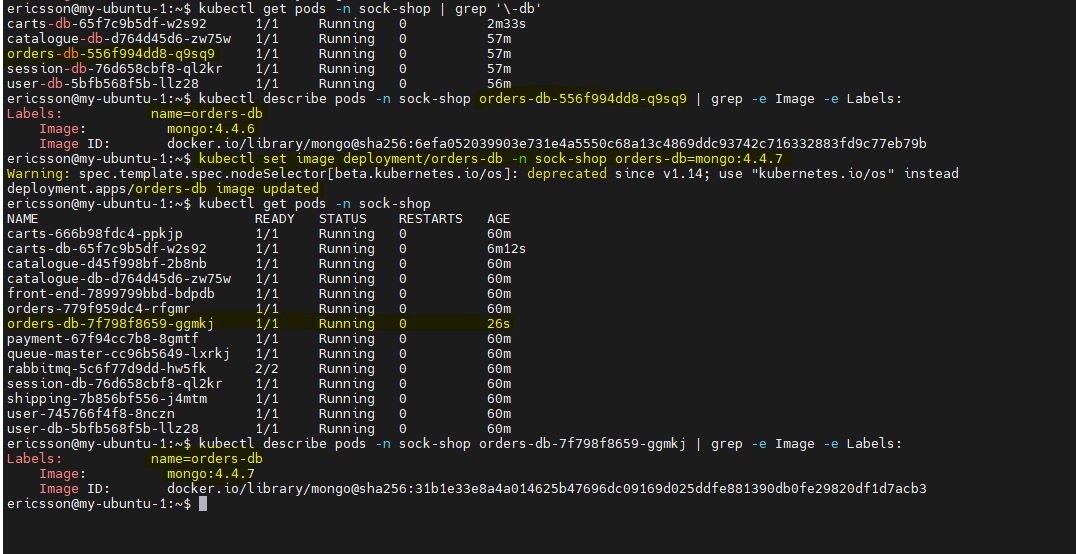
# Section 3

1. There are many different **deployments** that have the suffix “-db” in their name. What image each of them are using?



1. You want to upgrade the image of only the -db deployments that are using the mongodb image to the next minor version. For example, if the current image is mongo:1.2.3, then the next minor upgrade would be mongo:1.2.4. There are different ways to upgrade the deployments: there are implicit ways, explicit ways and a kubectl command just for image update. Chose the option you prefer. Show the commands you used and relevant output with highlights to upgrade the -db deployments to the next minor version of mongodb.

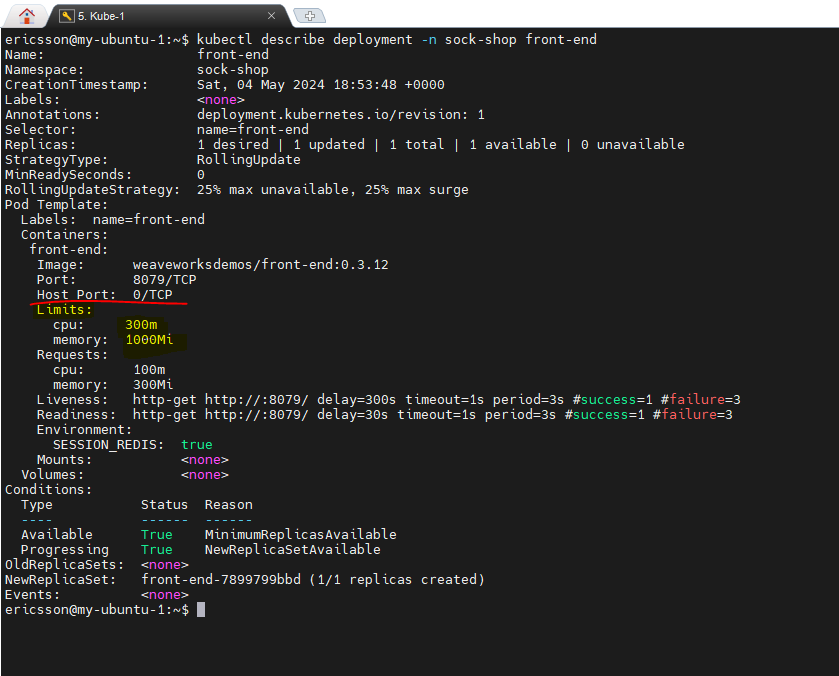




# Section 4

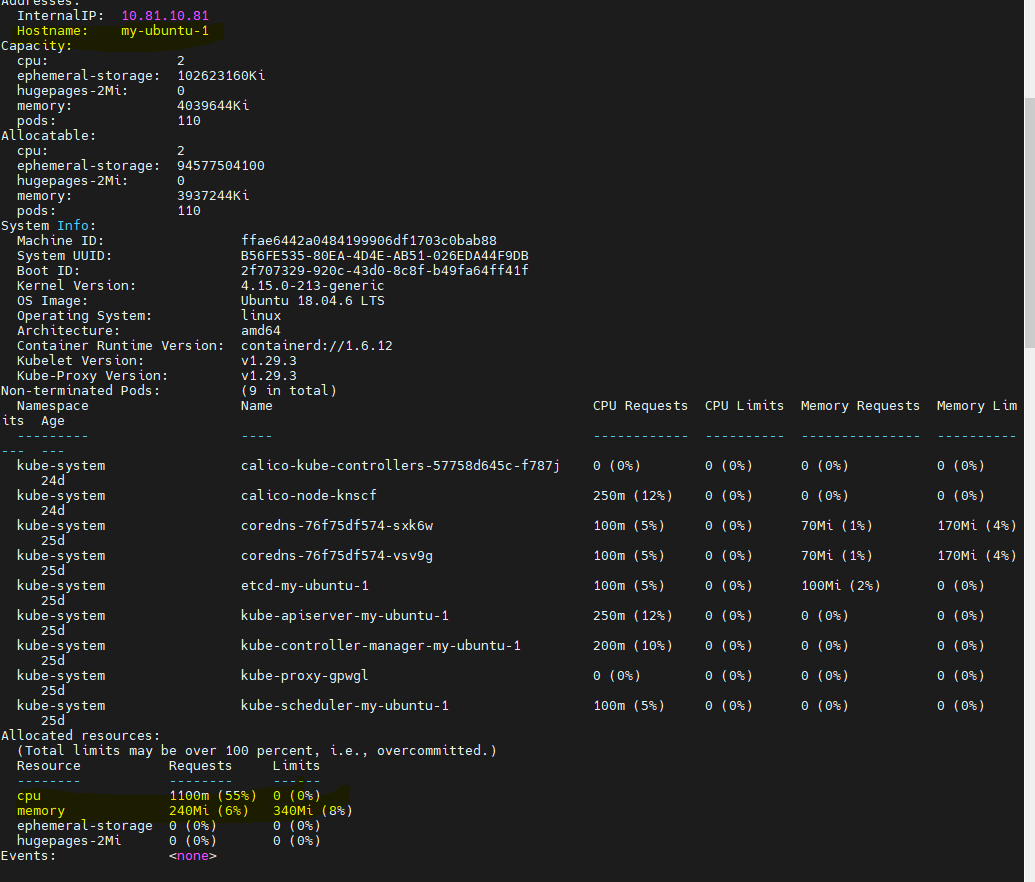
Focus only on the deployment *front-end*:

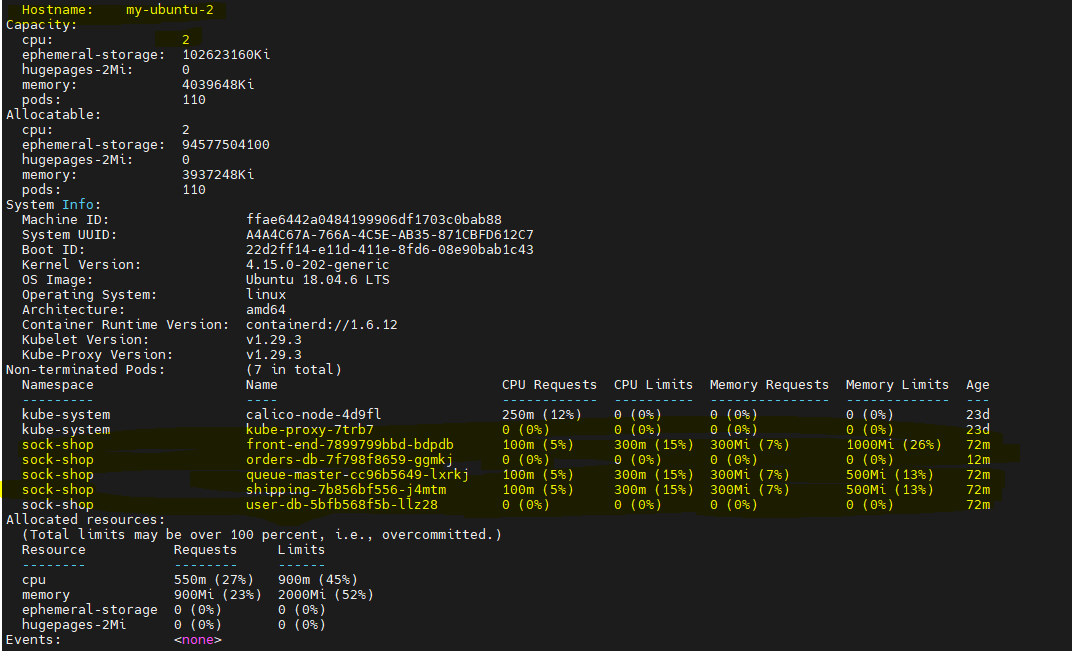
1. How much memory and cpu the deployment is limited to from a declarative point of view?

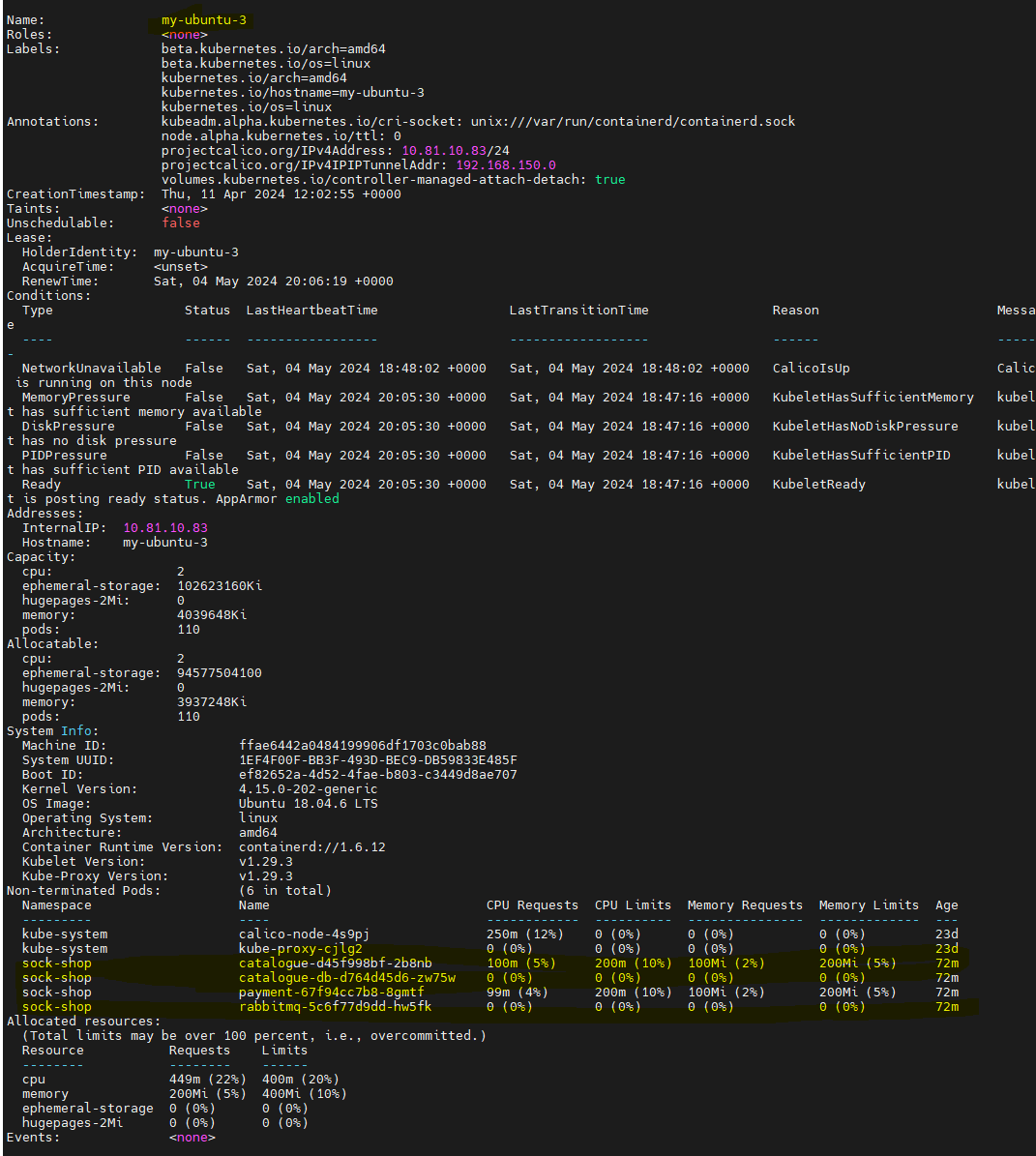


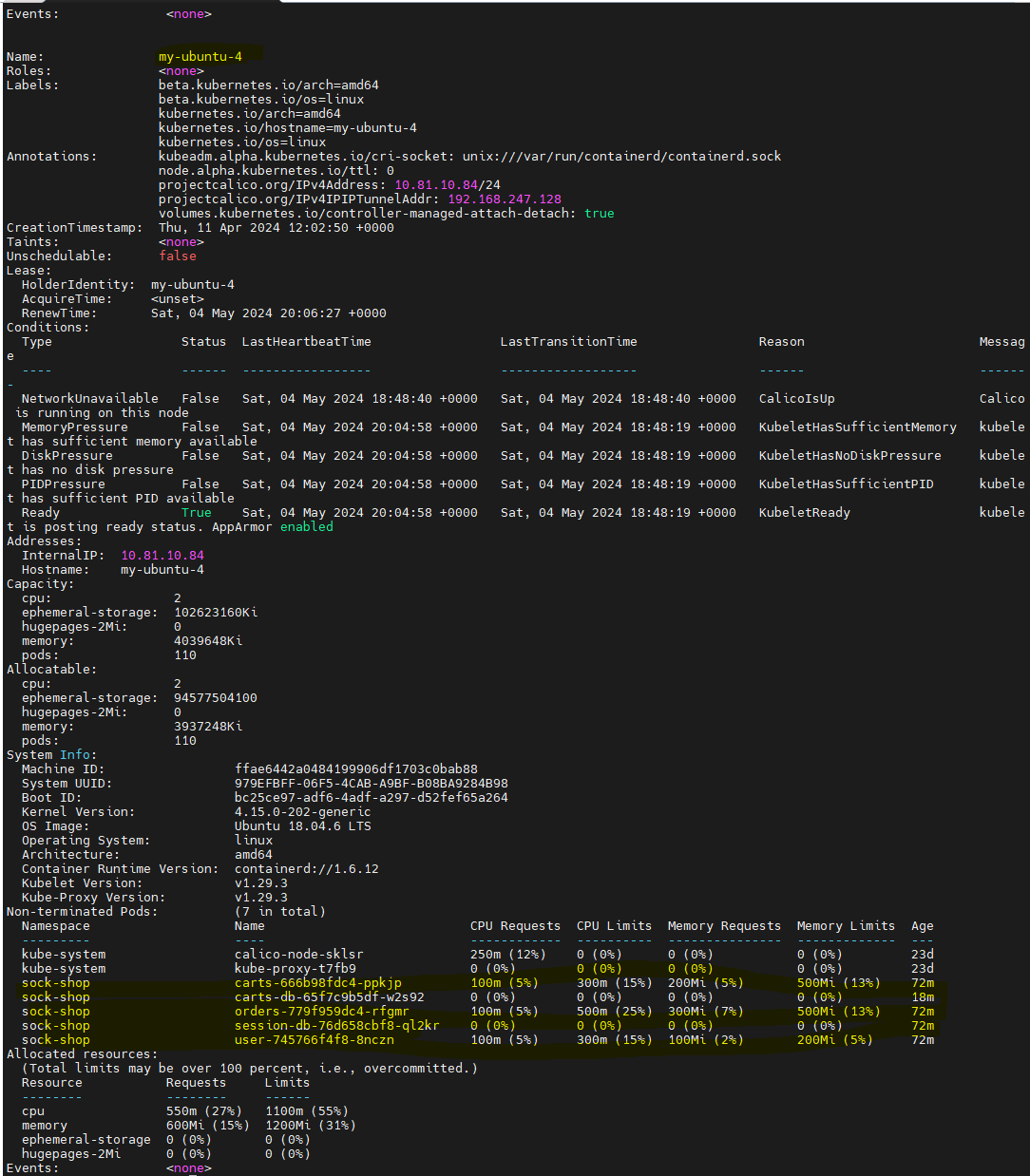
1. How much percentage of cpu and memory this deployment is consuming on each of the nodes?

kubectl describe nodes

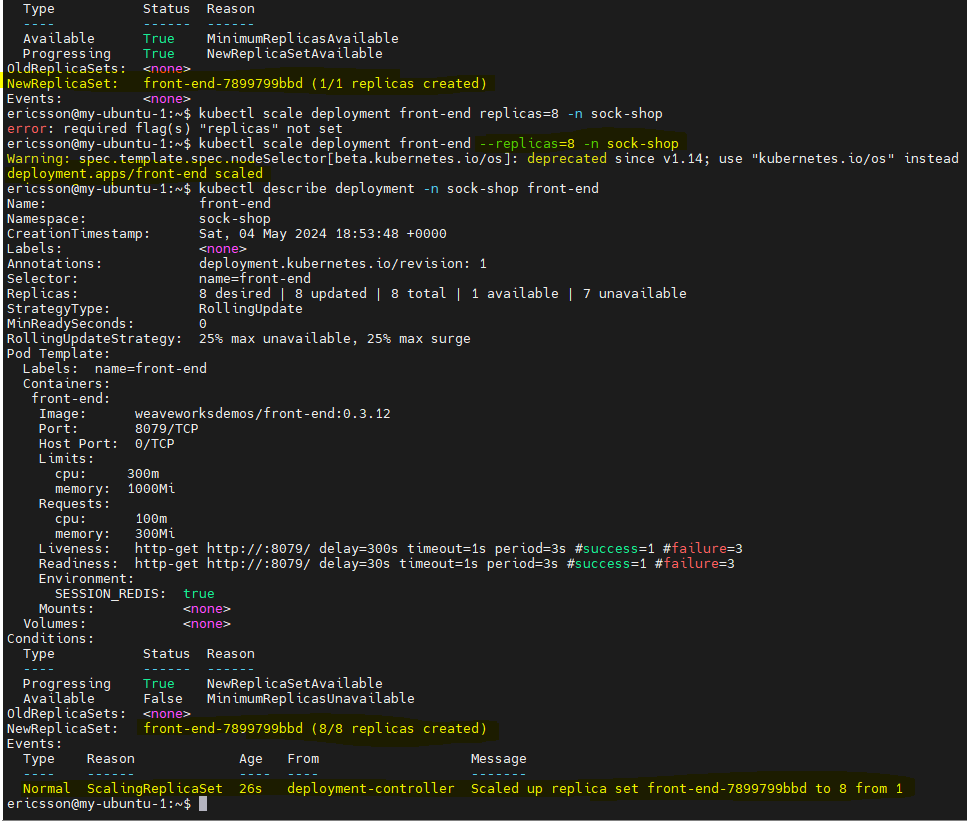






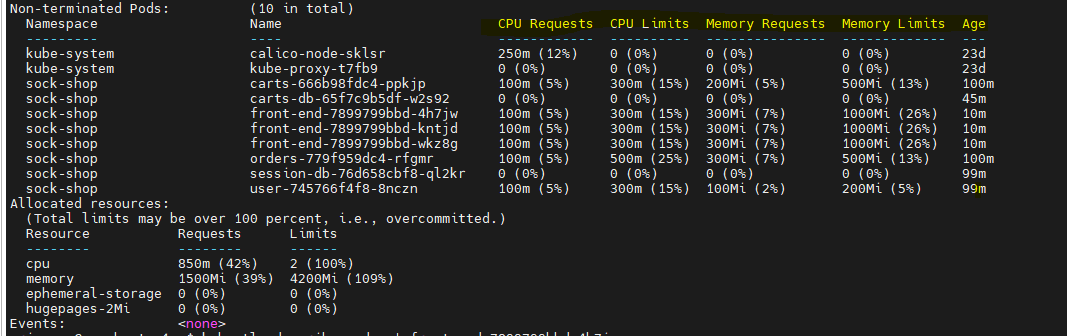


1. Your test team is sending a high amount of http traffic to your eCommerce and your Architect suggests redoing the tests with the front-end scaled out to 16. What command will you use to perform the expansion? Show the commands you used and highlight relevant output.   
   NOTE: if your Windows platform is the 16Gig version, you can scale up to 8 instead of 16



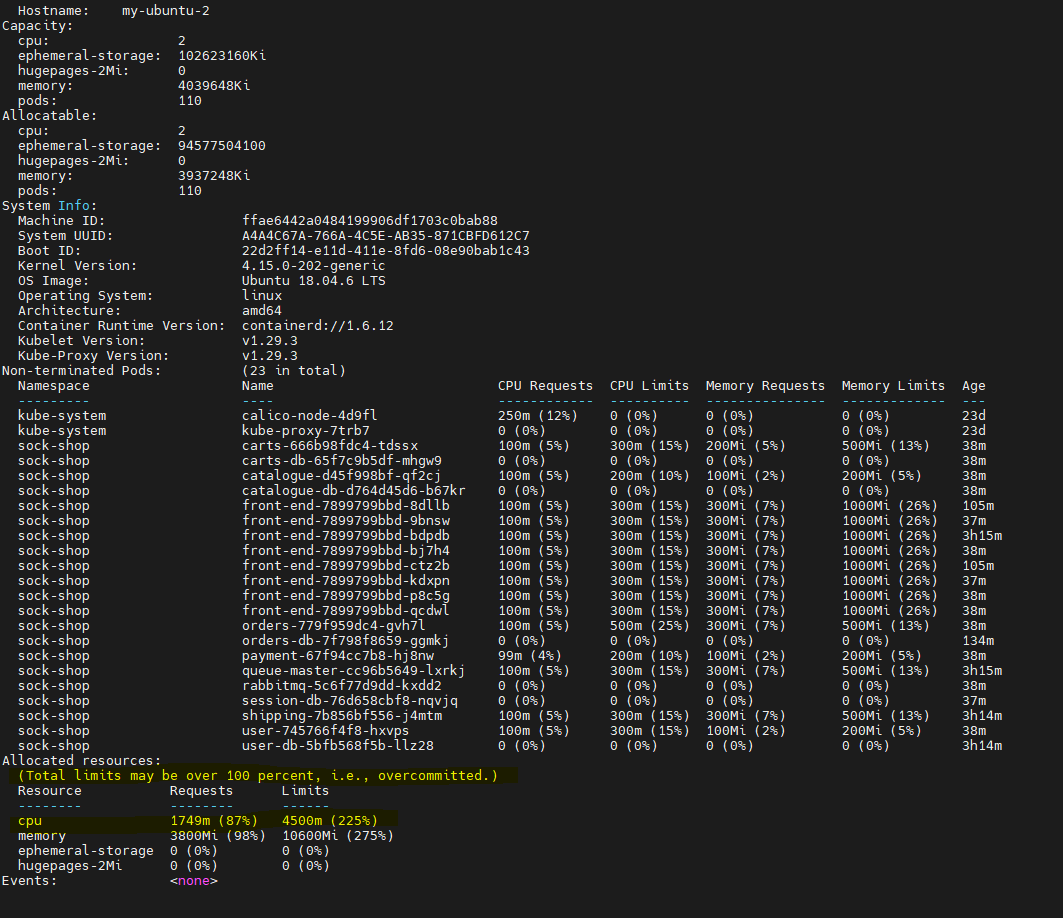
1. Verify on one of the worker nodes how much memory and cpu each of the front-end pods are limited to. Show the commands you used and highlight relevant output.

* Kubectl describe nodes



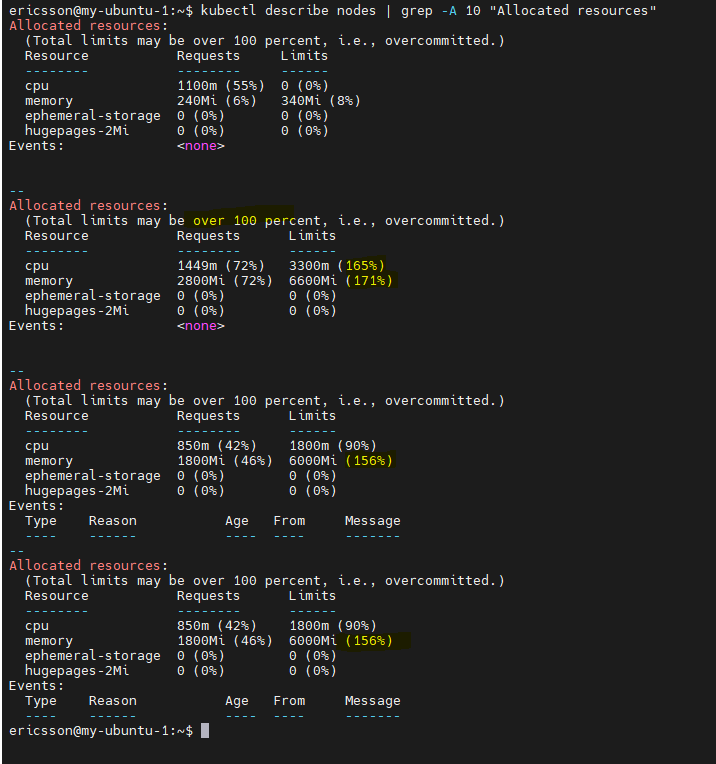
1. Your co-worker notifies you that we may have overcommitted the nodes by scaling out too much. Can you confirm or deny if ANY of the worker nodes are overcommitted?

* By using the following command, it has been found that none of the CPU or memory is overcommitted.

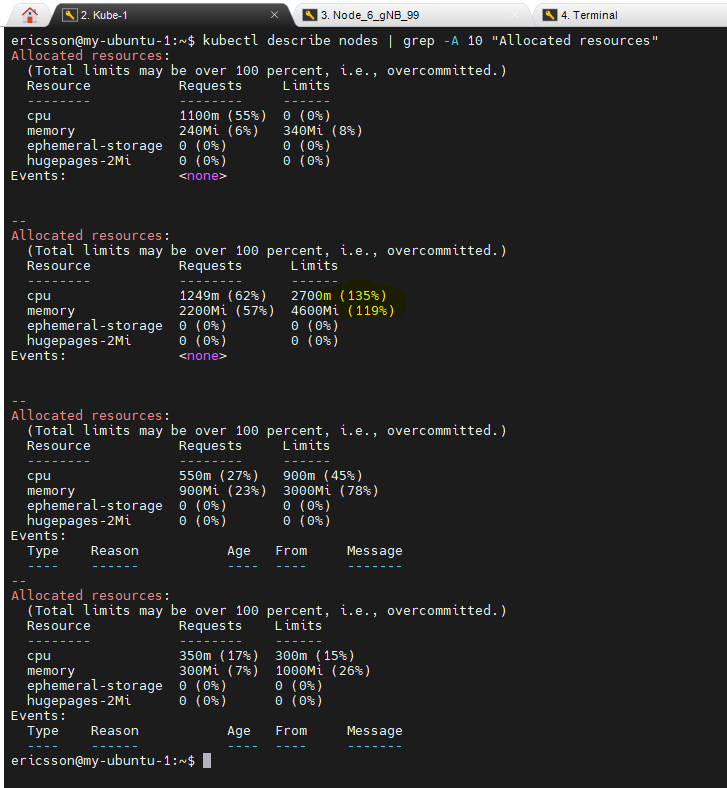


1. Scale the front-end deployment down until we are not overcommitted on any of the worker node. Identify that magic number.

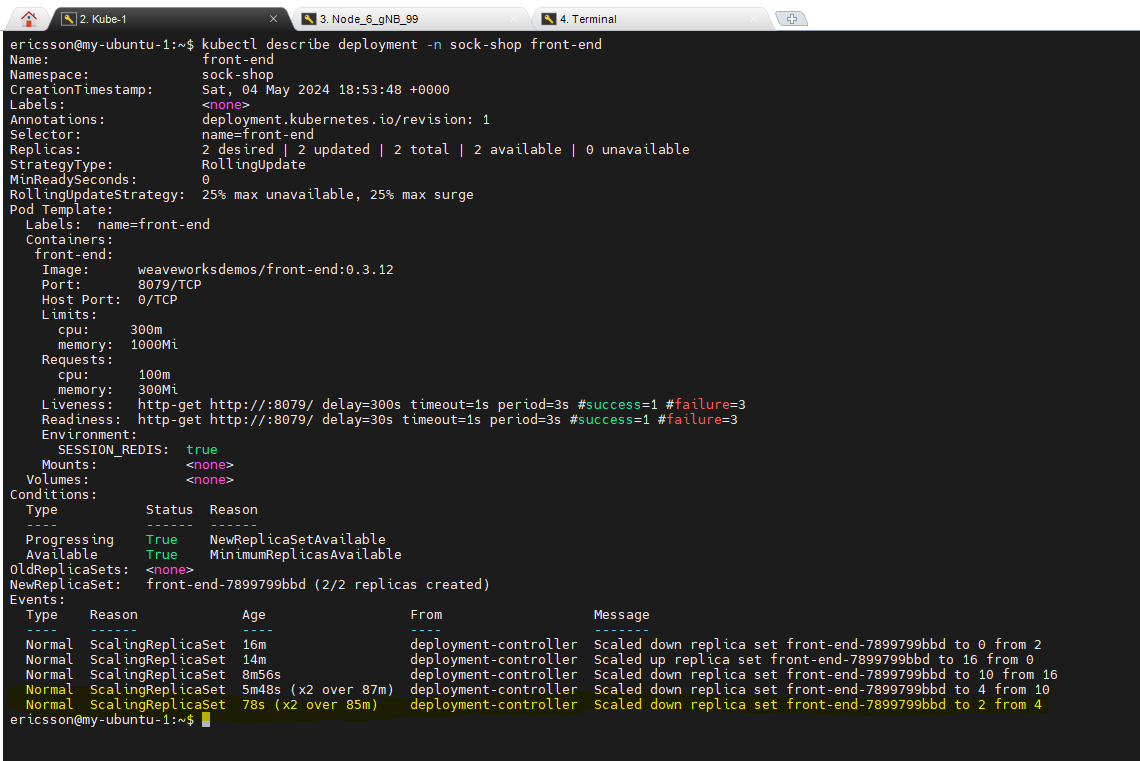
* **For 16 replicas**

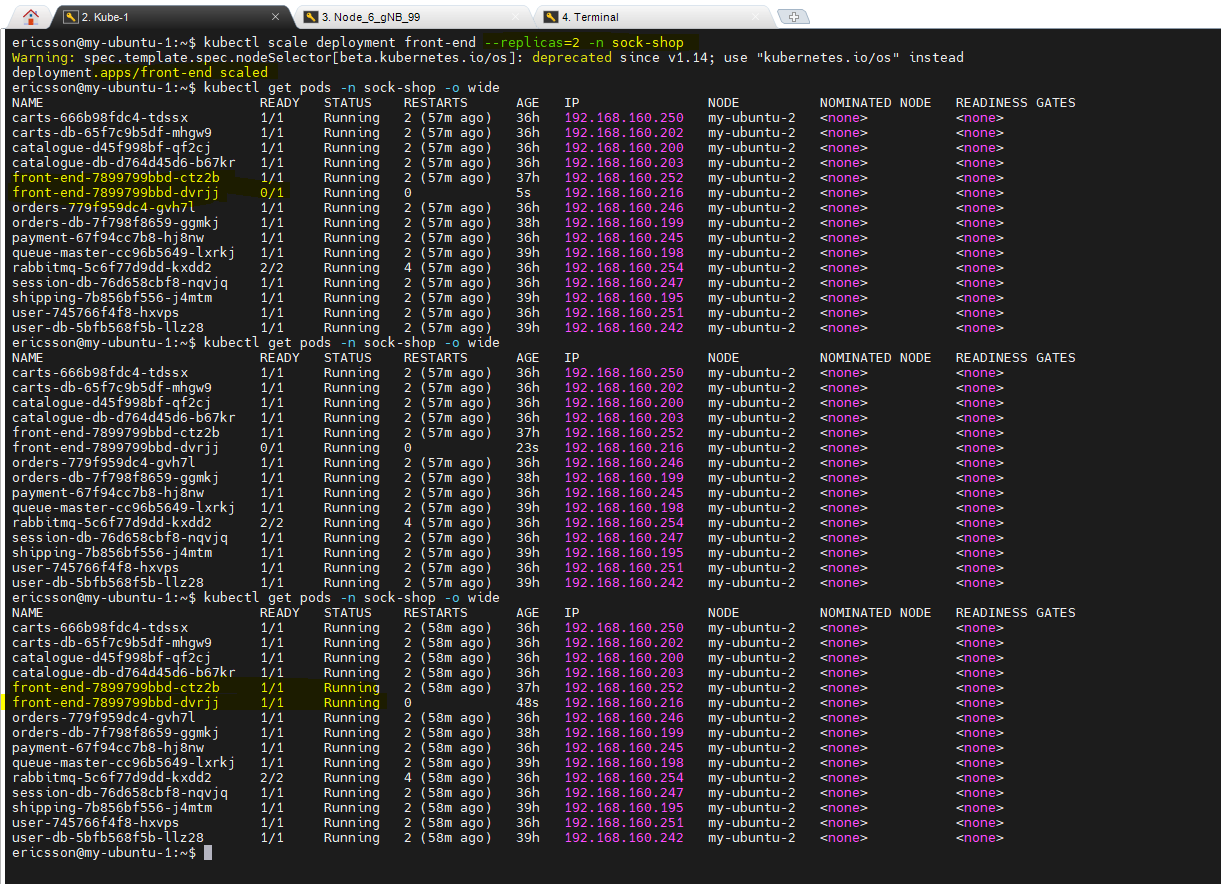


* For replicas up to 4 looks my-ubuntu-2 is not going



1. Scale back the front-end deployment to 2.

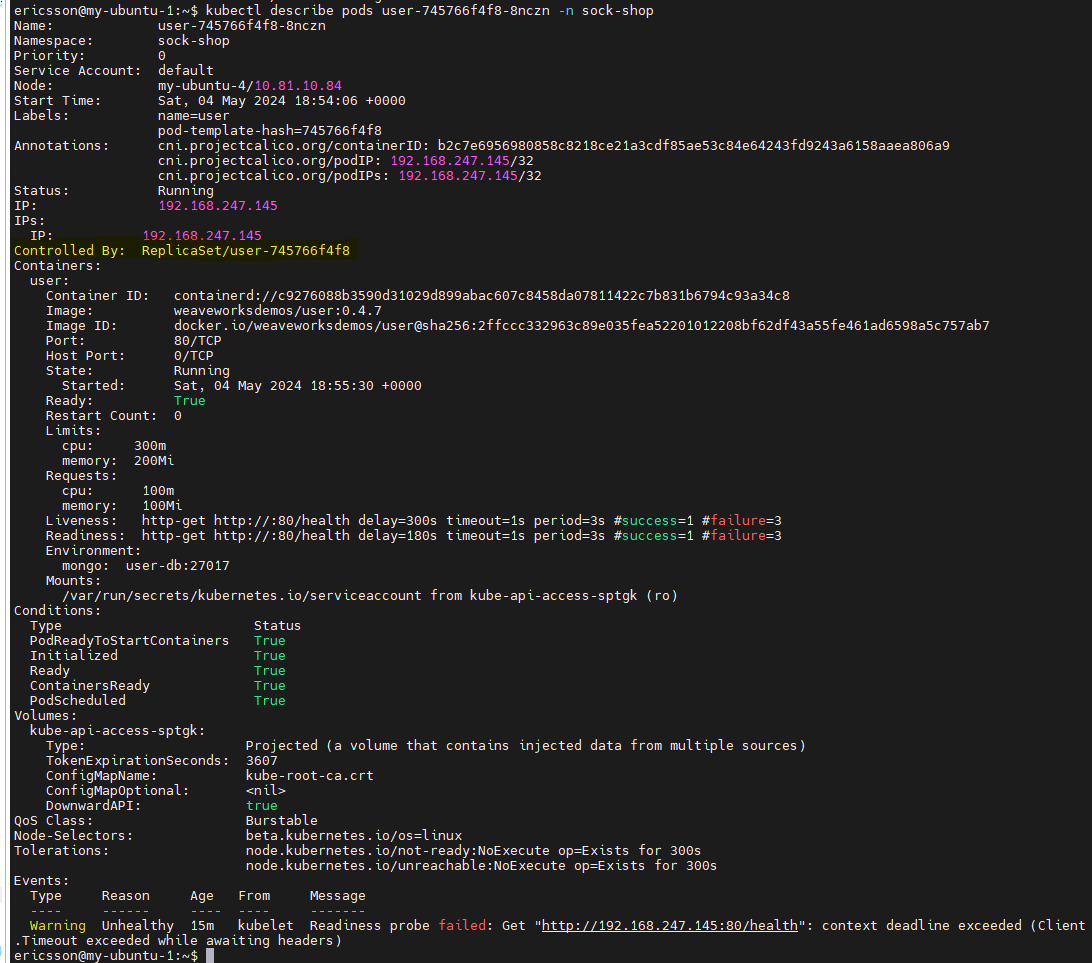




# Section 5

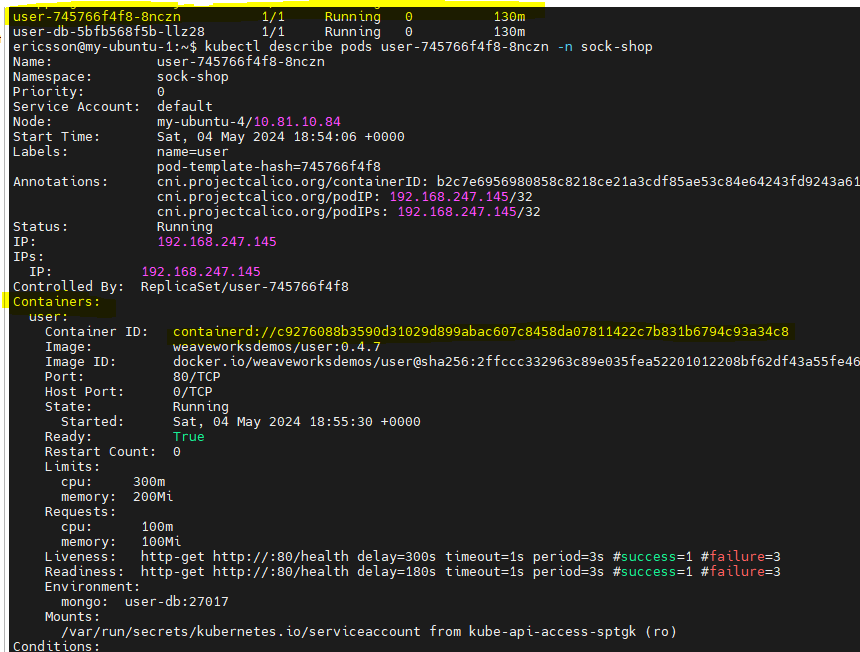
Focus only on the pod created by the *user* deployment, answer the following questions:

1. Is this pod controlled by a replicaset, a statefulset, or something else?



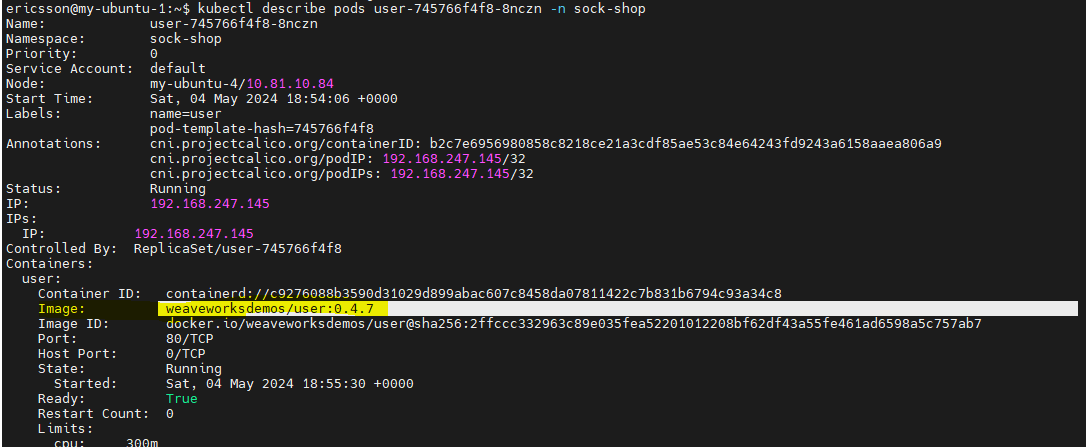
1. How many containers are running in this pod?

* There is one container named user.



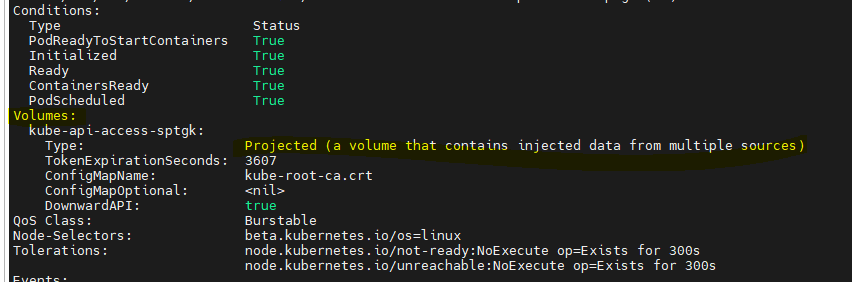
1. Which image is used by each container?

* kubectl describe pods user-745766f4f8-8nczn -n sock-shop



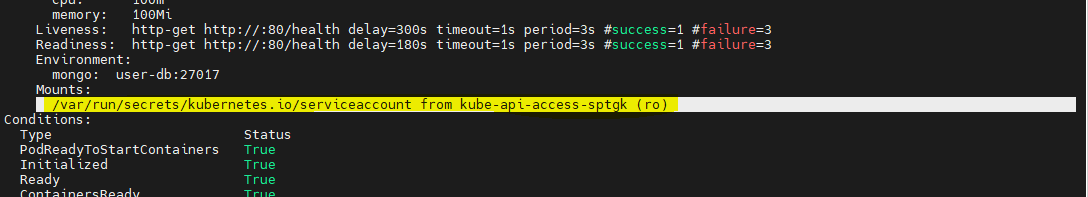
1. There is a volume attached to this pod, what is providing the data to that volume? A PermanentVolume ? A PermanentVolumeClaim ? Something else?

* kubectl describe pods user-745766f4f8-8nczn -n sock-shop



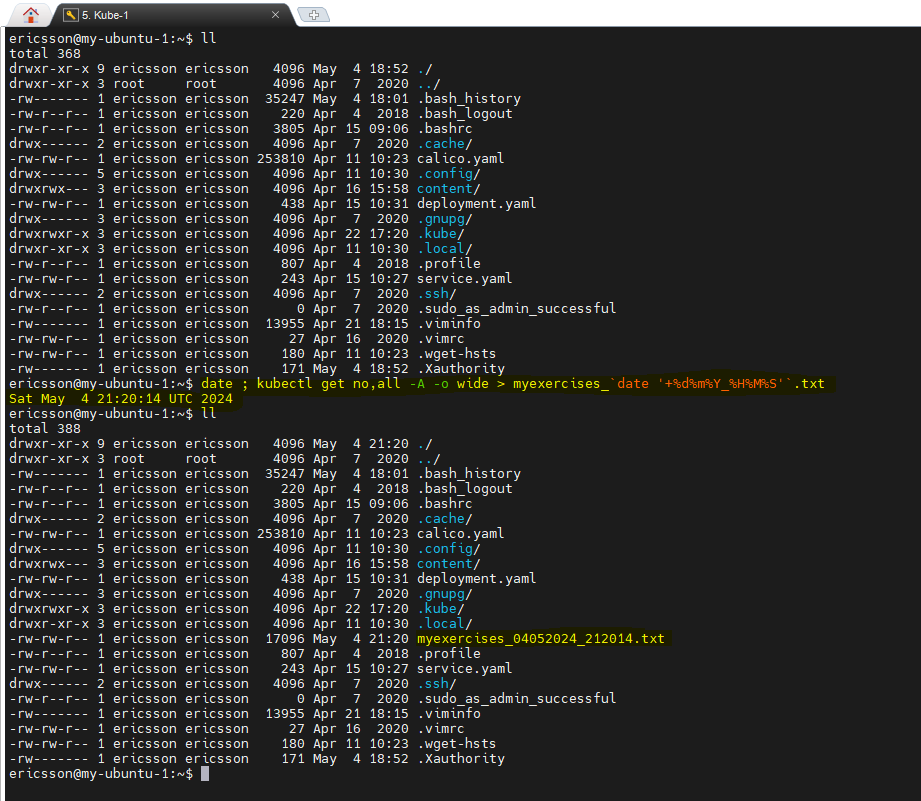
1. What is the full path of where this volume is mounted inside the container?

* kubectl describe pods user-745766f4f8-8nczn -n sock-shop



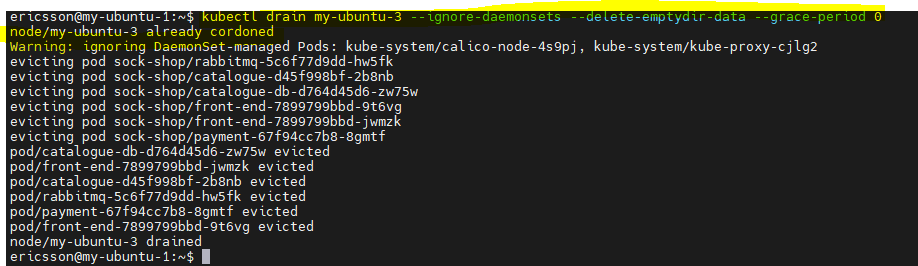
# Section 6

1. process this command: date ; kubectl get no,all -A -o wide > myexercises\_`date '+%d%m%Y\_%H%M%S'`.txt

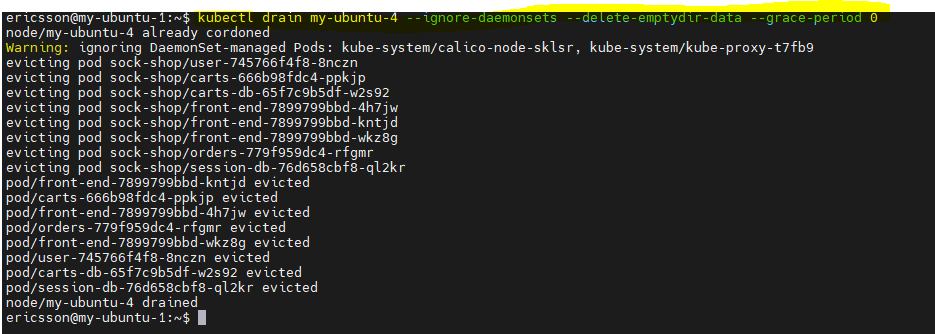


1. Prevent future scheduling on your last 2 worker nodes **and** empty them of any user pods, leaving one worker node available. It’s okay to delete the local data if asked, it’s a lab. (--delete-emptydir-data --ignore-daemonsets). Which command did you use to accomplish these tasks?



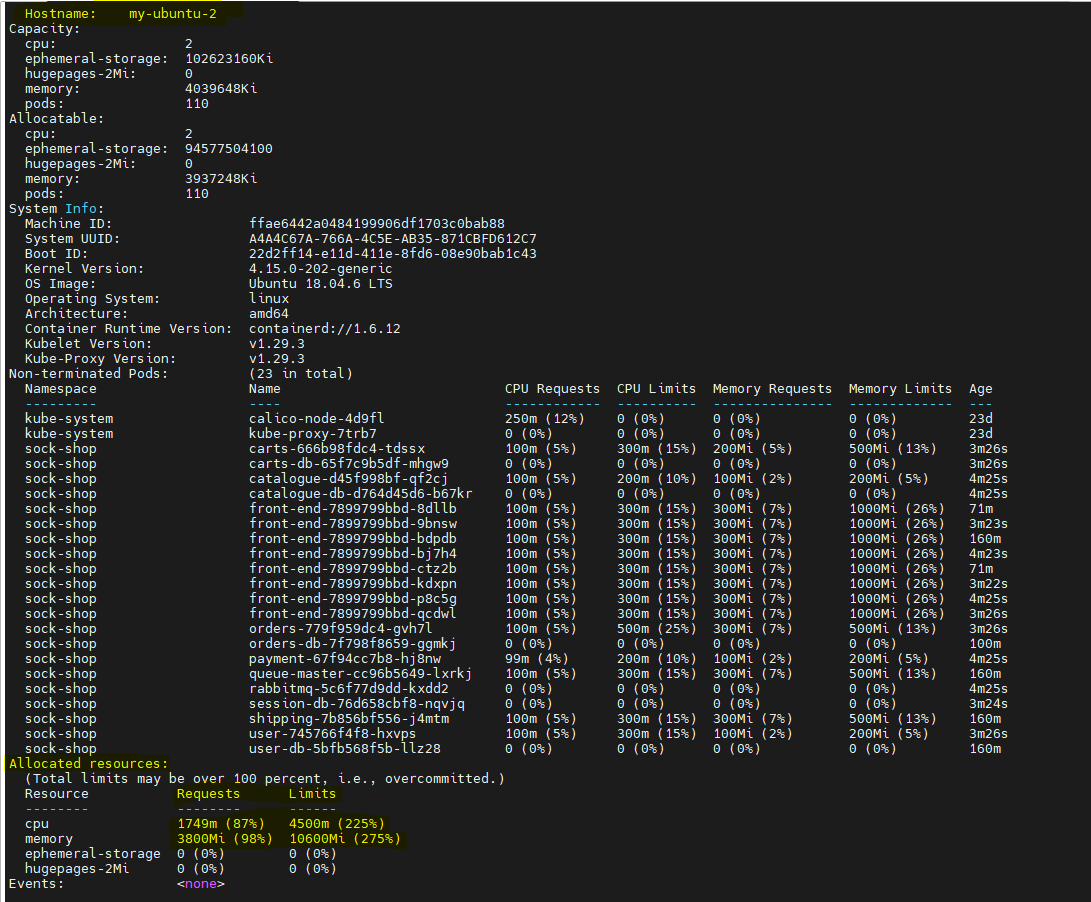






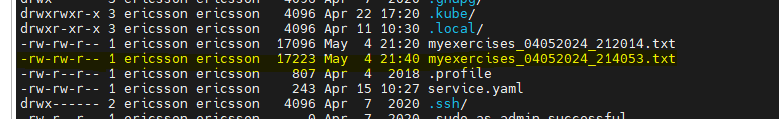
1. Now that your only remaining worker node has all the sock-shop resources scheduled to it, do we have another overcommitment situation on it? If yes, is it on Requests, on Limits ? on both ?

* Node is overcommitted
* Kubectl describe nodes.

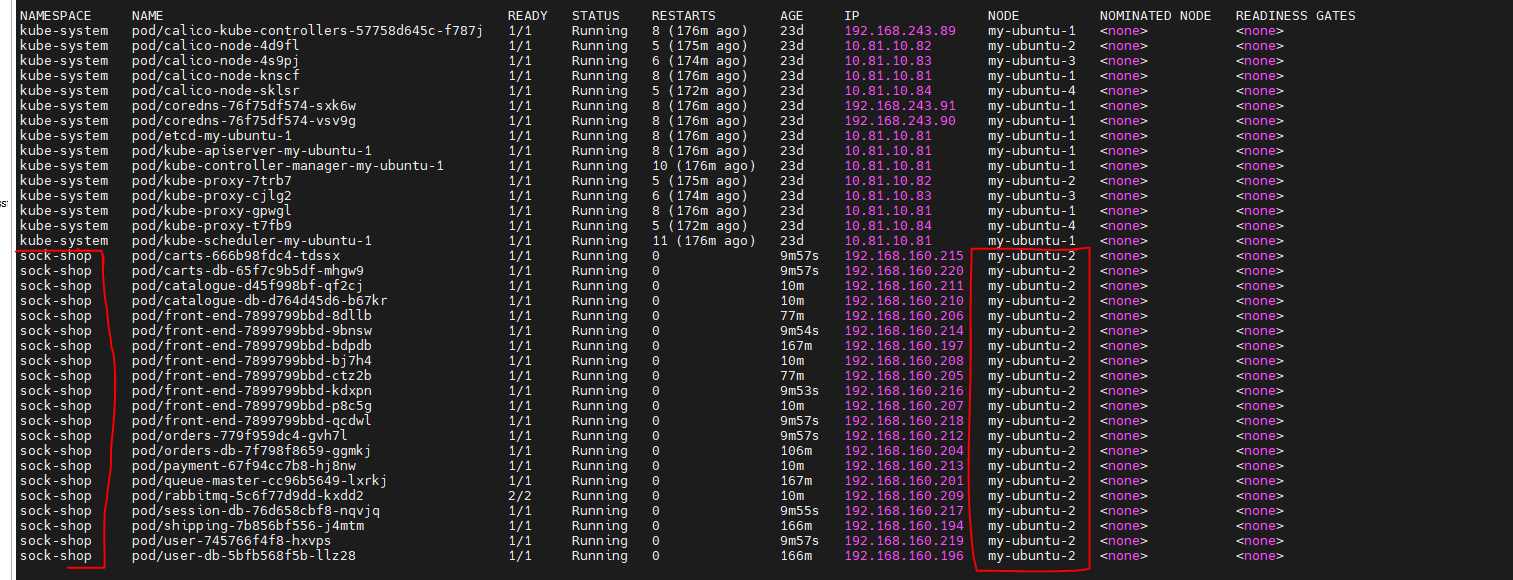


1. Rerun the same command as a) to document that your last 2 worker nodes are not carrying any application pods.

* We Rerun same command as a) and newly created **myexercises\_04052024\_214053.txt**

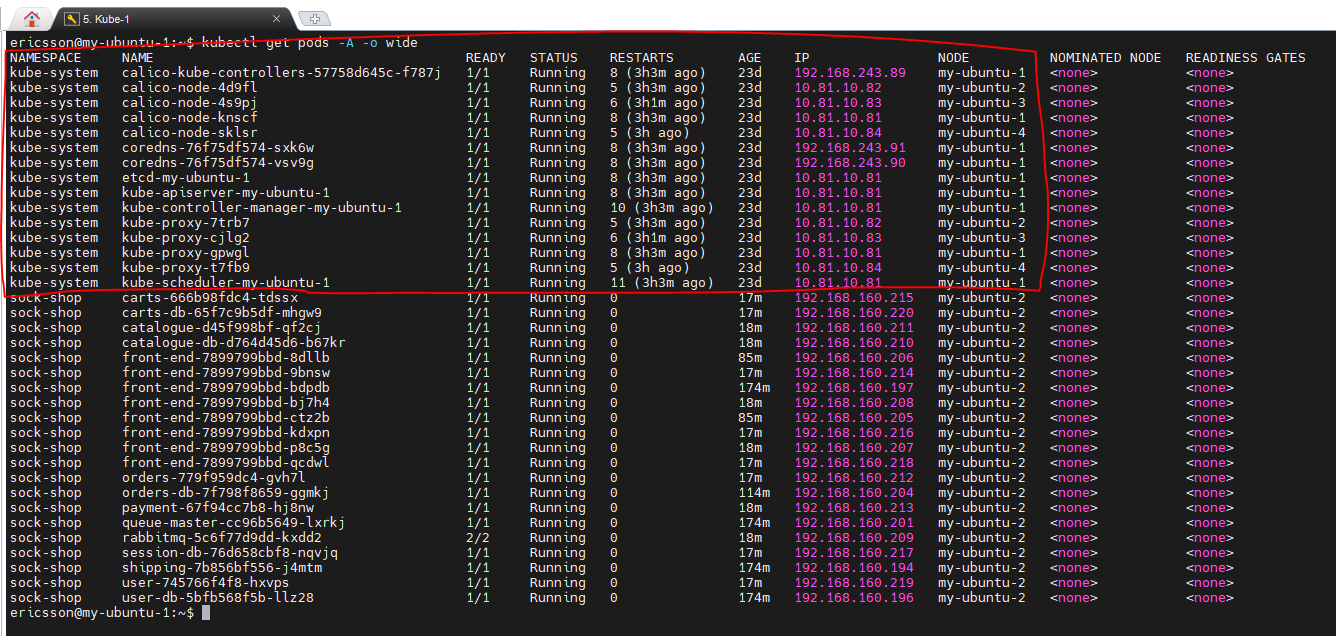


* Inside of the **myexercises\_04052024\_214053.txt** all the sock-shop are running under my-ubuntu-2



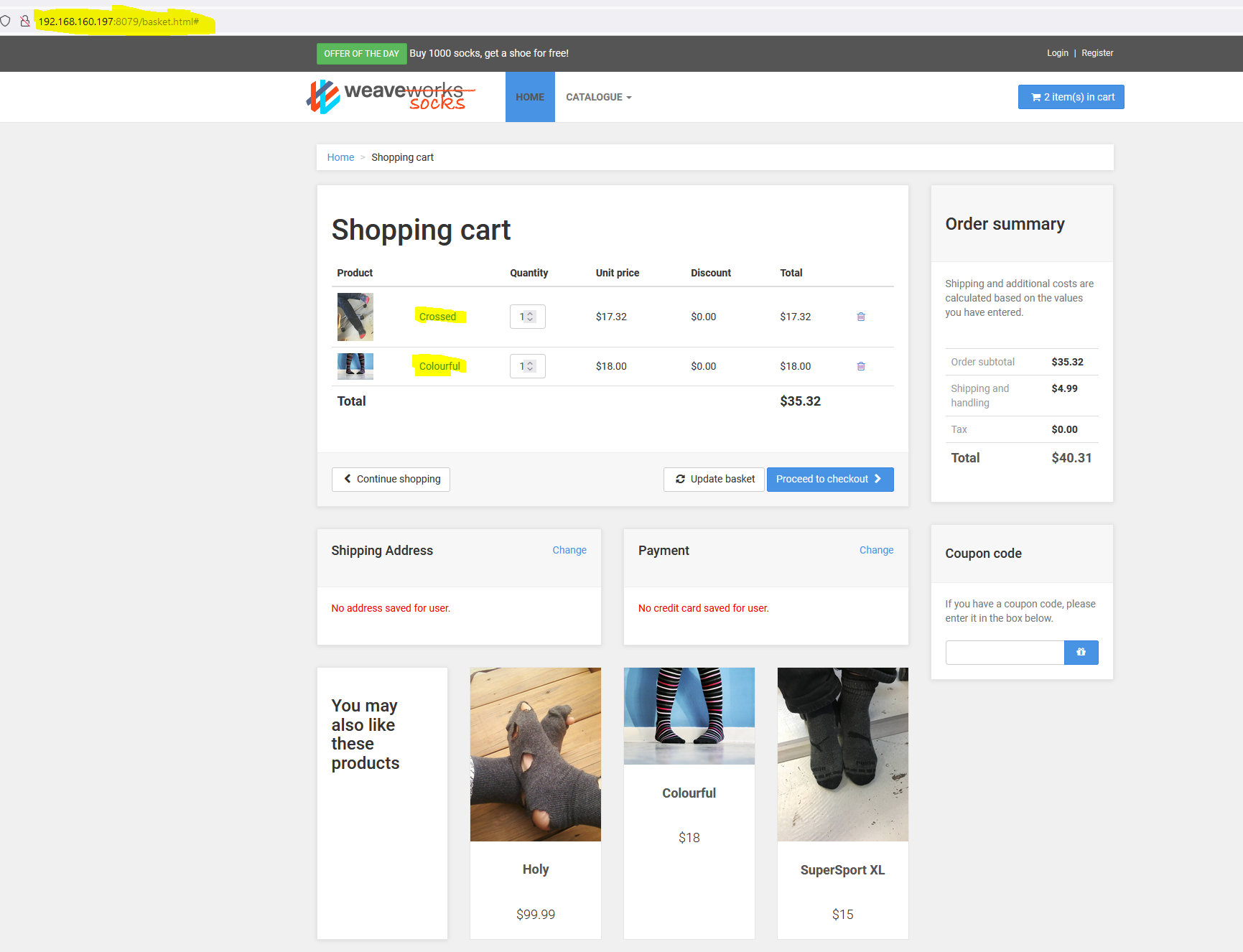
1. Do you see remaining pods on last 2 worker nodes when processing :   
   kubectl get pods -A -o wide’ ?   
   if yes how come? I though we drained the last two worker nodes.

* The pod belong to namespaces sock-shop, has moved to my-ubuntu-2, however, calico nodes still on my-ubuntu-3 and my-ubuntu-4, due to ignore daemon sets.



# Section 7

1. Which product on the Sock-Shop website you like the most? add it to your cart.
2. Crossed
3. Colourful

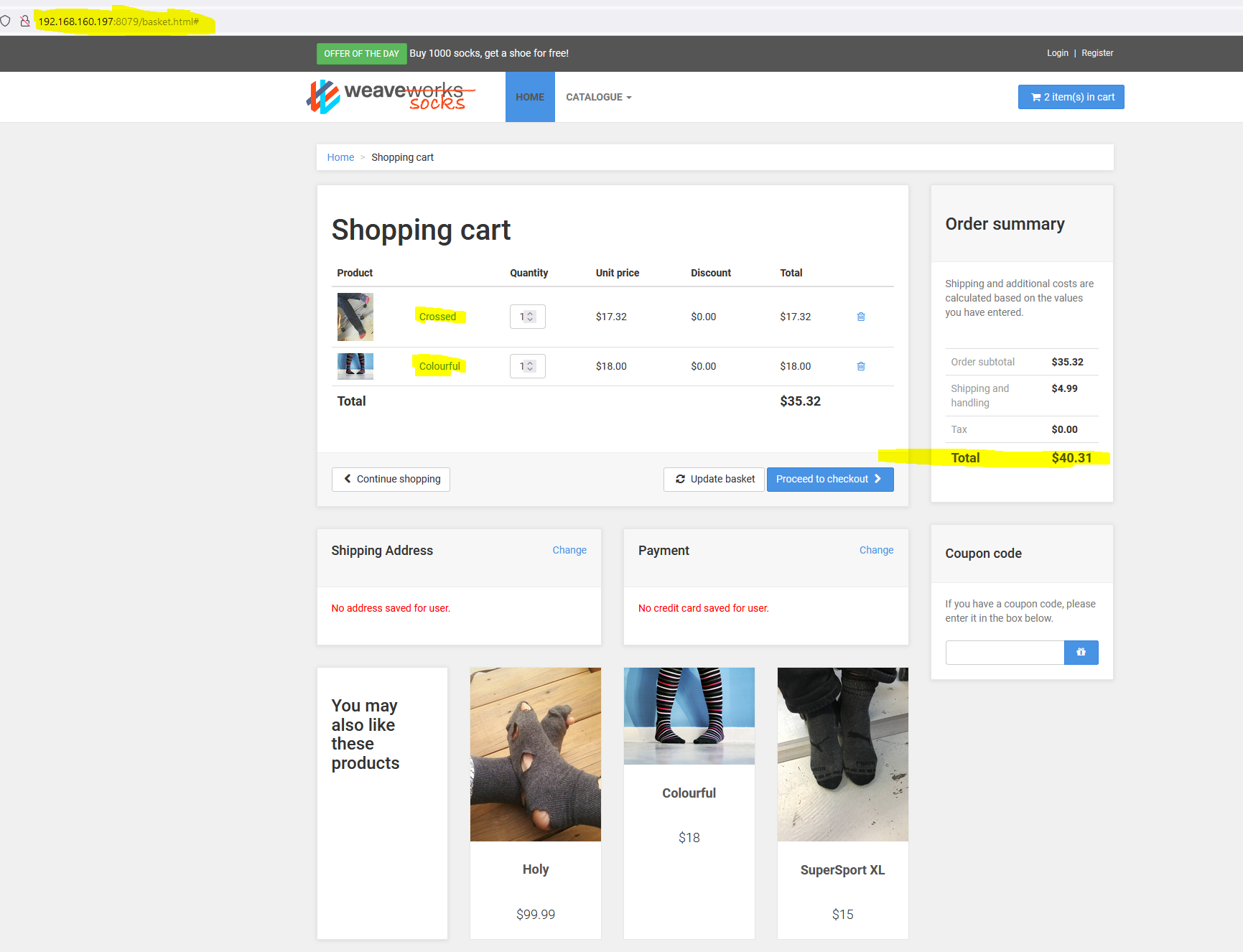


1. Which one would you like to order for your manager? Add it to your cart.

* **I want to order for my manager. See above screen capture cart.**

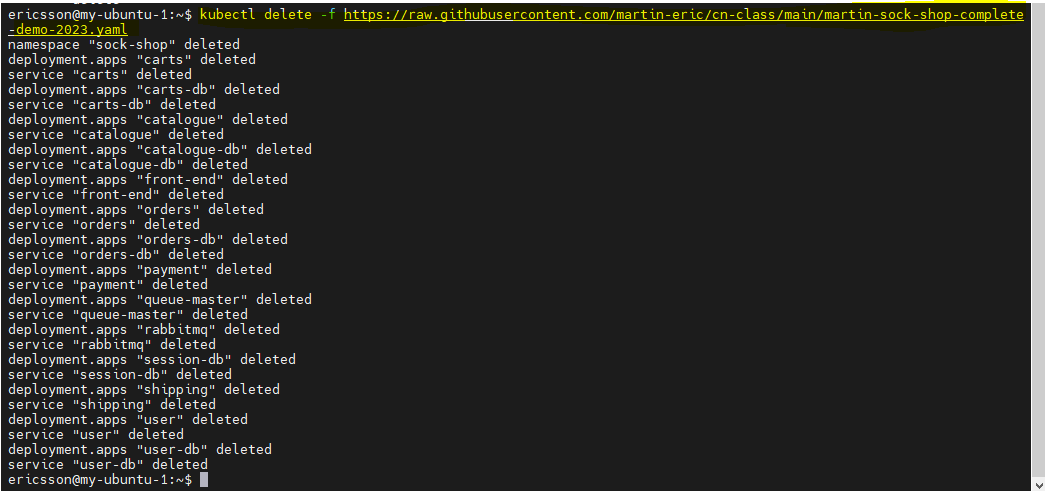
1. What is the total price of your cart? You can supply screen capture.

* Total price is $40.31.



# Section 8

1. Delete all the resources this exercise created with the least number of commands. Show your work.

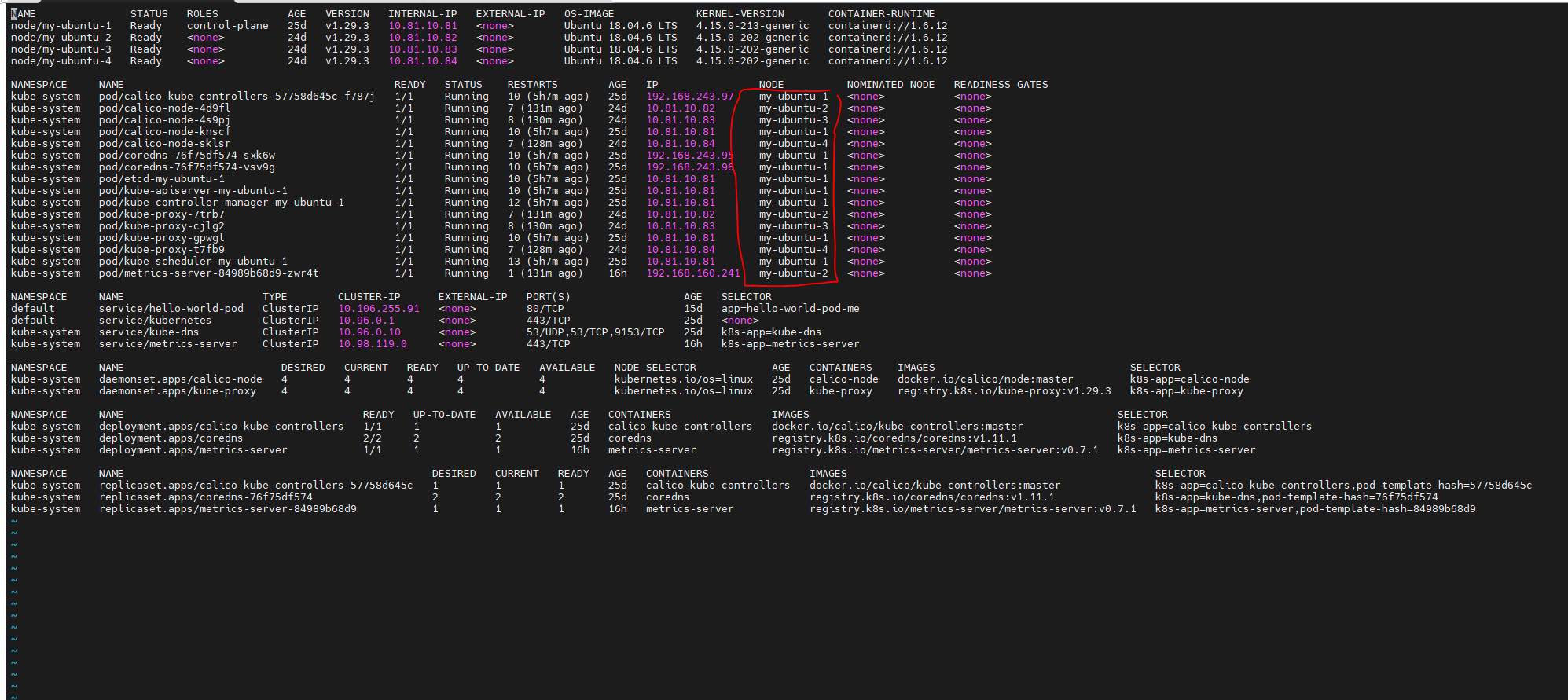


1. Ensure your last two worker nodes can accept future deployment scheduled to them 😉

* Now my-ubuntu-3 and my-ubuntu-4 will accept future deployment.  
  

1. run this command again : kubectl get no,all -A -o wide > myexercises\_`date '+%d%m%Y\_%H%M%S'`.txt

* Inside of the folder



# Conclusion:

1. Send all your answers and the **3 files created from 6a, 6d, and 8.3** by email to your instructors. Ensure the title of the email is the name of this exercise.
2. If the file is too big, use zip to compress it.