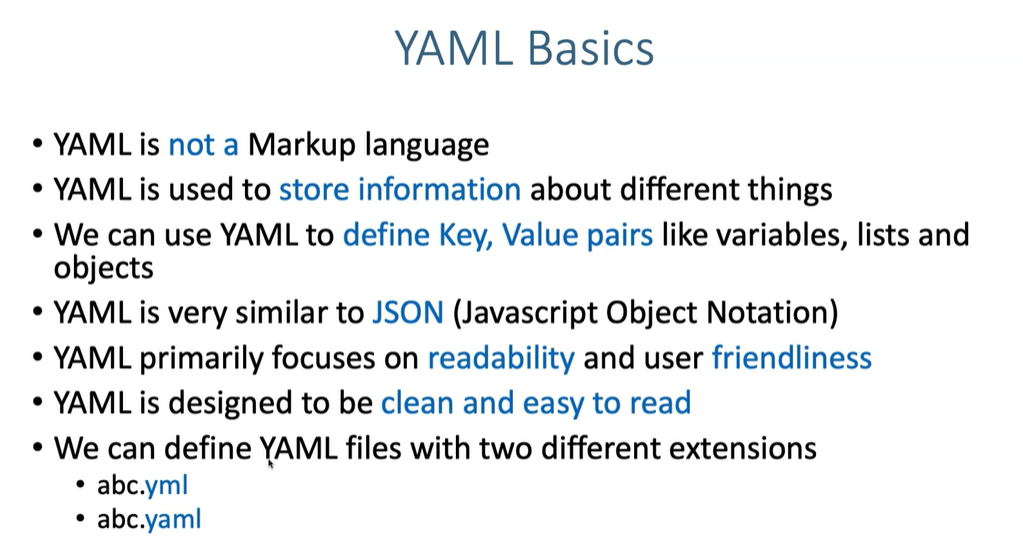
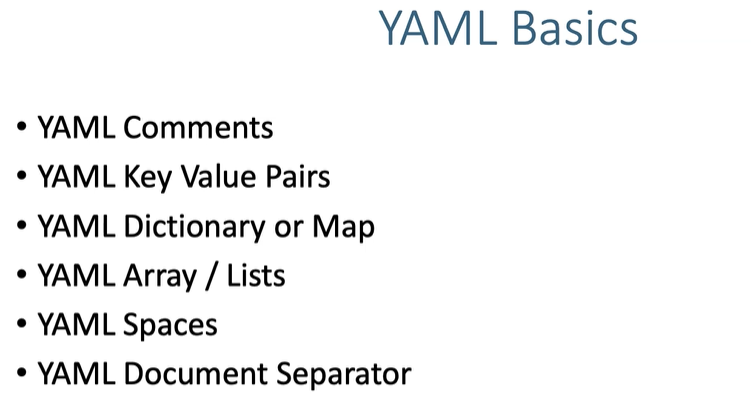
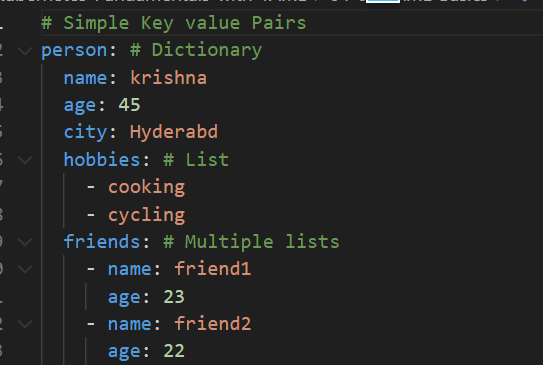
**KUBERNETES-YAML and Helm Charts**

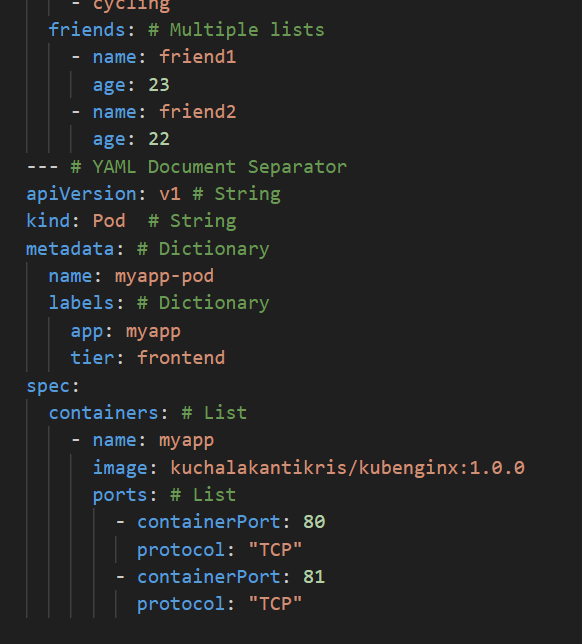
****

****

****

**Sepratrator**

**---**

**You can add multiple containers  
**

**HELM**

Helm is the package manager for K8s Application

For linux [ yum, aptget ]

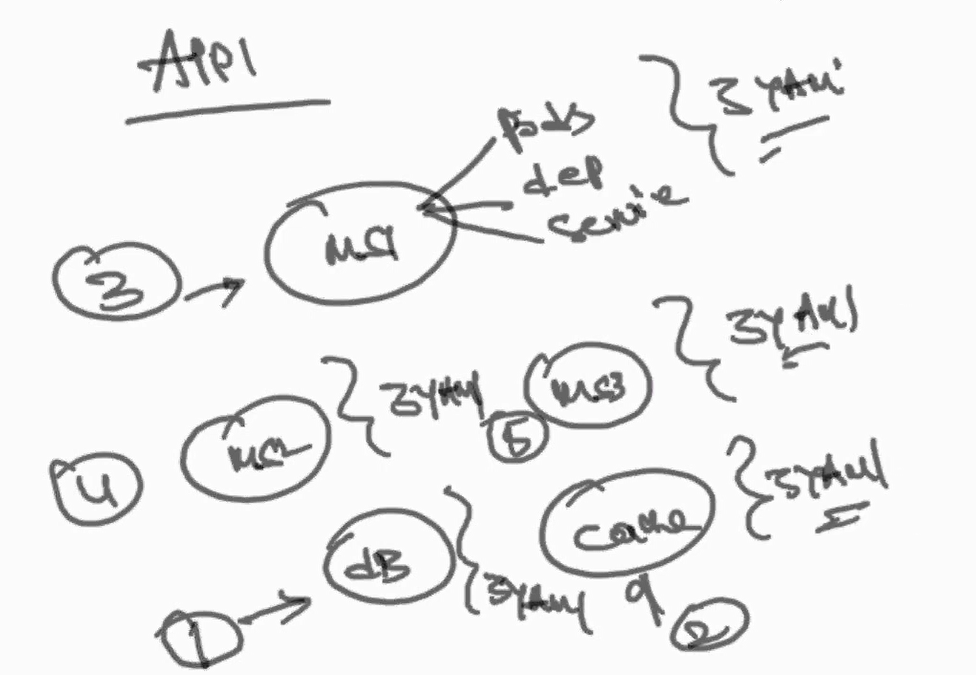
What is the advantage of yum and aptget

Lets take an example

They will offer tomcat as a package, when I am trying to install tomcat, => this requires JDK, when you install tomcat it automatically install JDK.

* A proper version control of your deployments

For example:



I have application which is using microservice1, ms2,ms3,db and cache management and each has 3 yamls file. Now I would like to install first db, second cache, 3rd ms1, 4th ms2, 5th ms3. That is there is order of deployment and I need to install all yaml files sepearately.

We can package everything using helm charts , helm is very important for automation.



In the Helm, we will have charts 🡺 it is like a package, like application package 🡺 inside that we will have templates🡺 inside templates we will have all yaml files in this.

We can paramentise the template.



curl https://baltocdn.com/helm/signing.asc | sudo apt-key add -

sudo apt-get install apt-transport-https --yes

echo "deb https://baltocdn.com/helm/stable/debian/ all main" | sudo tee /etc/apt/sources.list.d/helm-stable-debian.list

sudo apt-get update

sudo apt-get install helm

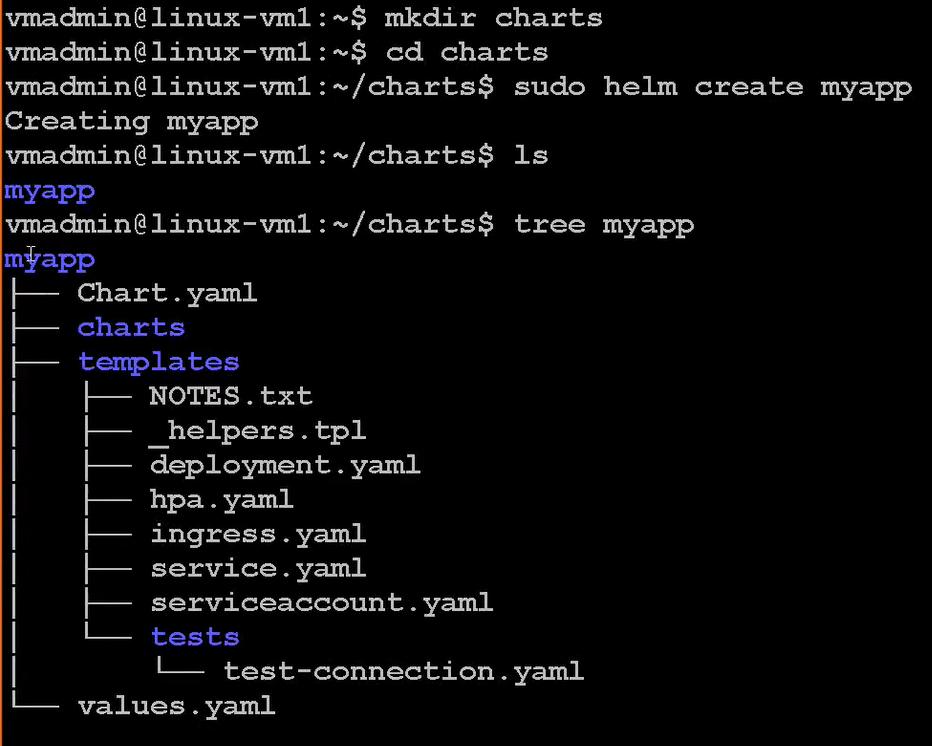
helm version --short

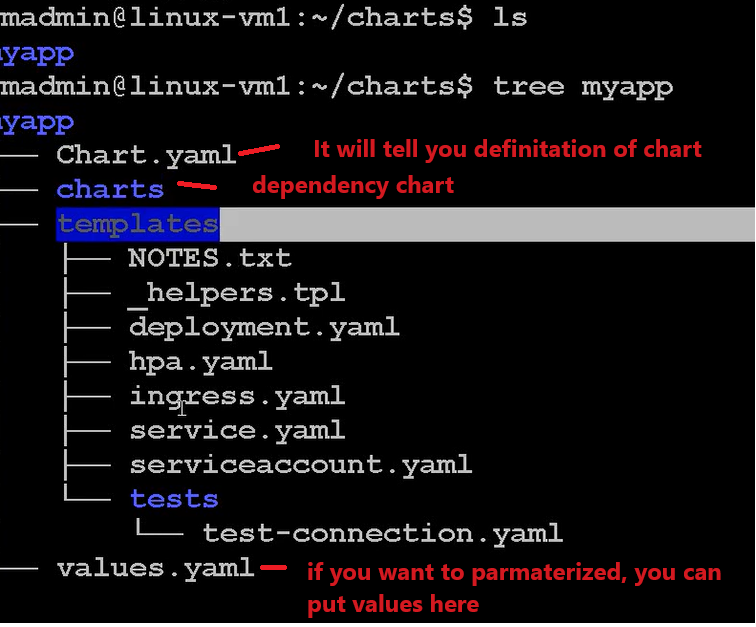
curl -sL https://aka.ms/InstallAzureCLIDeb | sudo bash

az –version

# mkdir charts

# cd charts





Let us delete this and create step by step



Let us create manually.

# mkdir myapp

# cd myapp

Myapp# touch chart.yaml

Myapp# mkdir templates



# vi chart.yaml

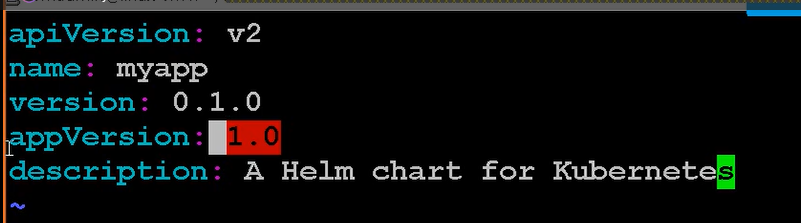
vi myapp/Chart.yml

apiVersion: v2

name: myapp

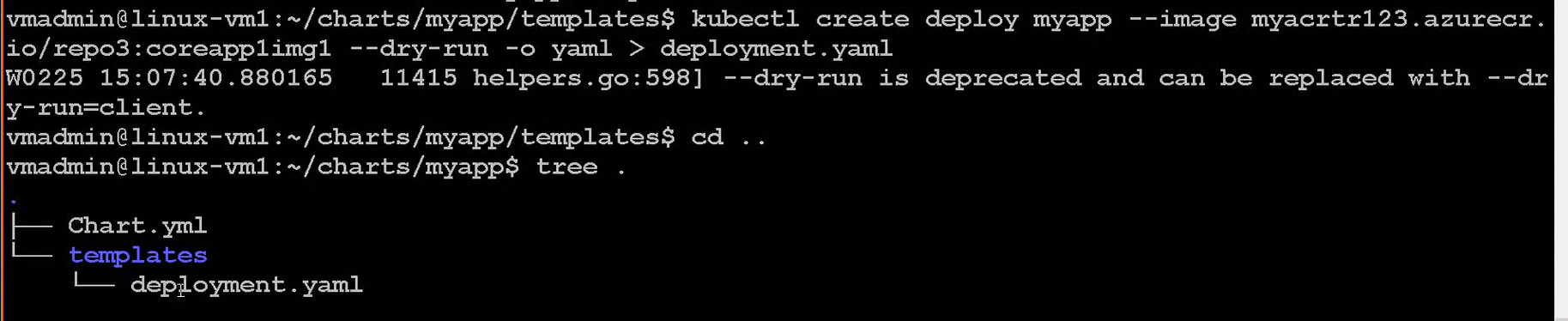
version: 0.1.0

appVersion: 1.0

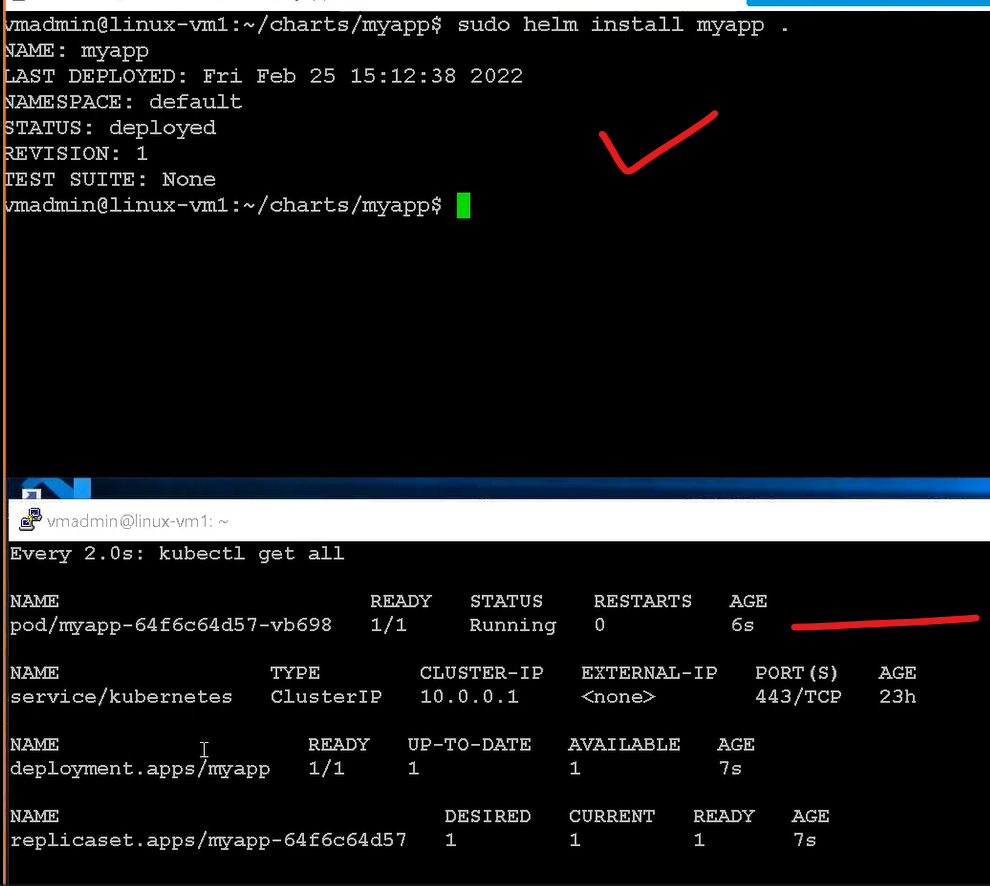
description: A Helm chart for Kubernetes  
  


157 cd myapp/templates/

158 kubectl create deploy myapp --image myacrtr123.azurecr.io/repo3:coreapp1img1 --dry-run -o yaml > deployment.yaml



now let us install

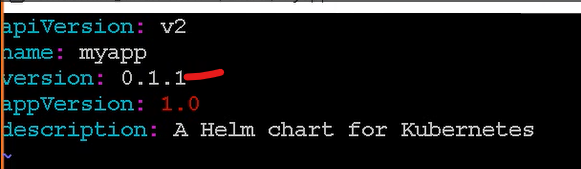


Now let us modify replica set to 3

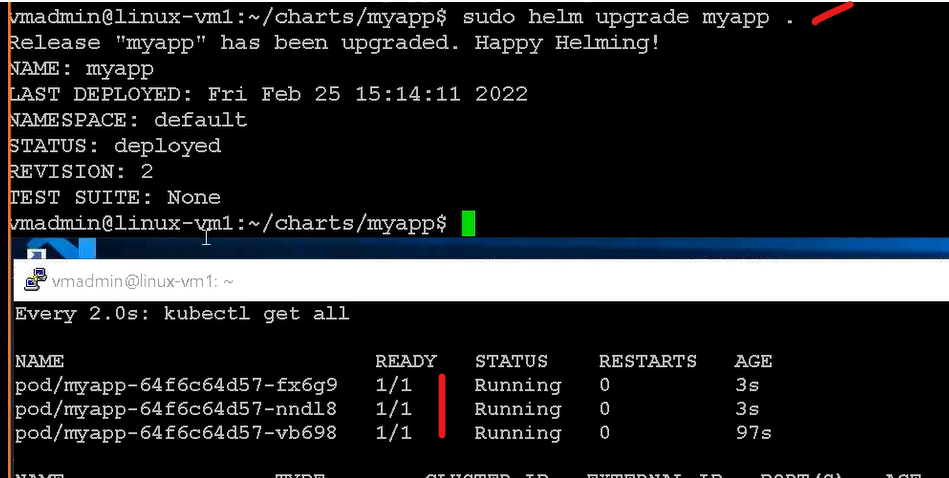
Vi deployment.yaml and change replica to 3

And update chart version

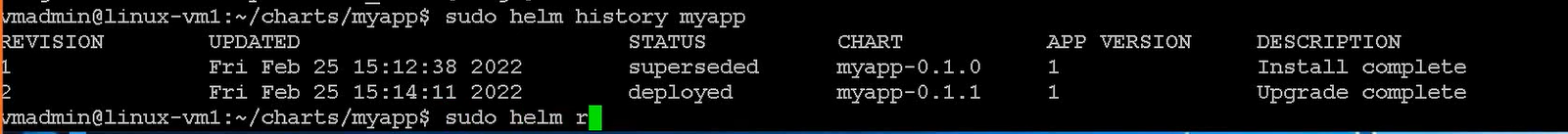
Vi chart.yaml



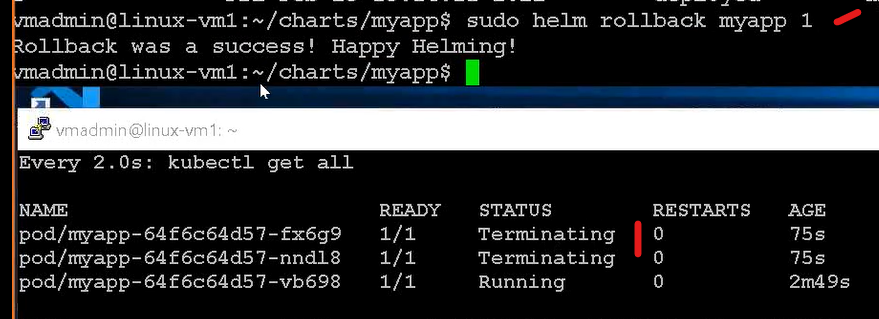
Now I need to upgrade

# sudo helm upgrade myapp .  


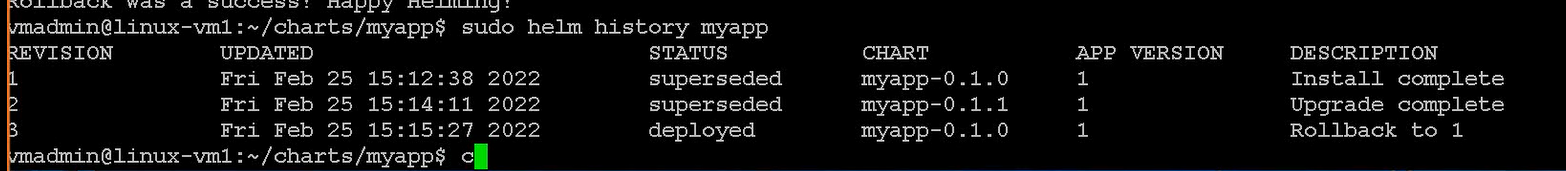
# sudo helm history



# we can roll back to older version

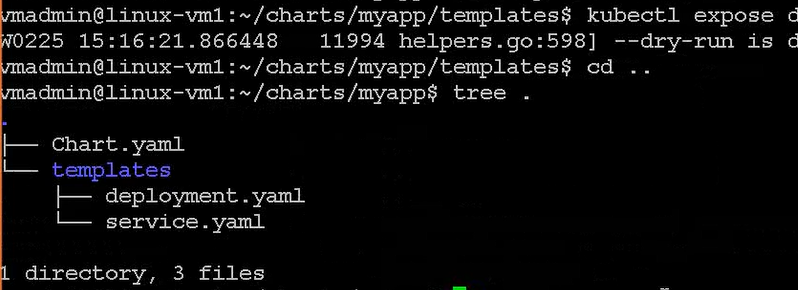


# sudo helm history



Now I will add revision 4 , as I want to add service.yaml file

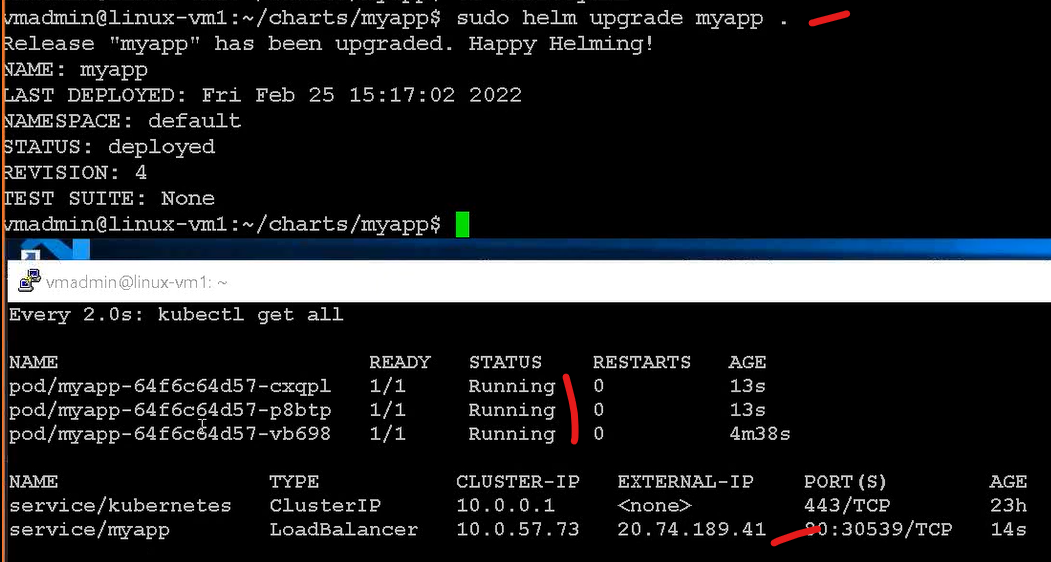
Cd templates

Templates#   
kubectl expose deploy myapp --port 80 --type=LoadBalancer --dry-run -o yaml > service.yaml  
  


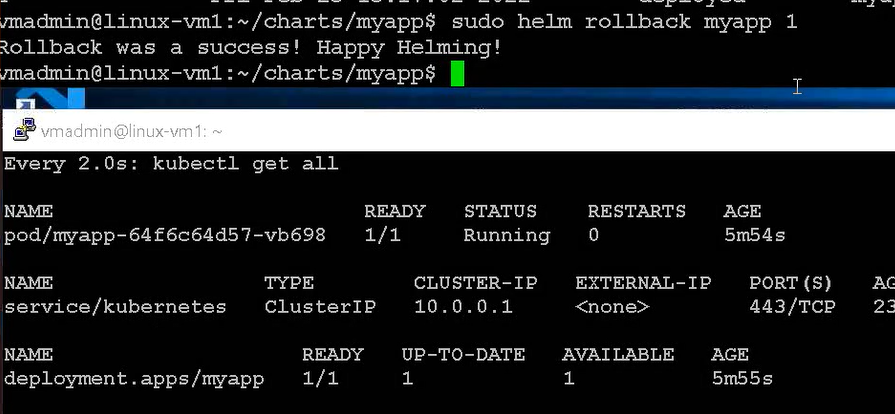
Now let me change chart version

Vi chart.yaml  

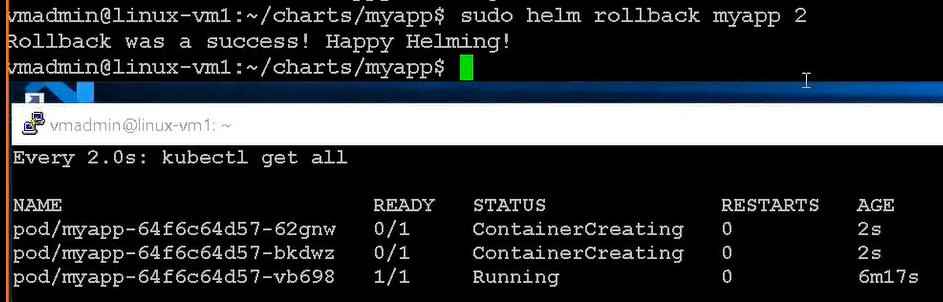

# sudo helm upgrade myapp .



Now you can roll back to version 1, where you will have only one pod



# now you can roll back to version 2, where you will increase only pods



This is how , you can do complete rollbacks and upgrades, everything we can run as a one unit, no need to install service, deployment separetly

Let us completely uninstall

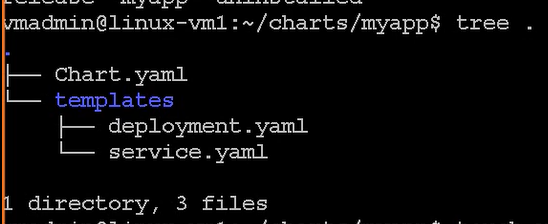
# sudo helm uninstall myapp



You can completely templatize this ( parameterise)

You can specify variables as well

I am using same location



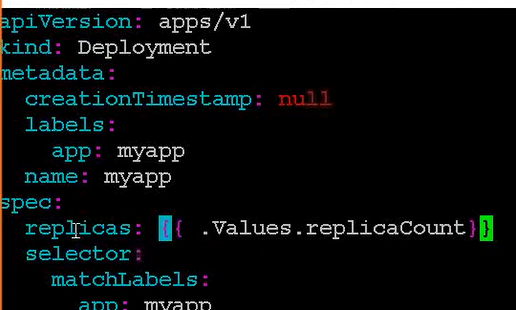
Now let us create values.yaml



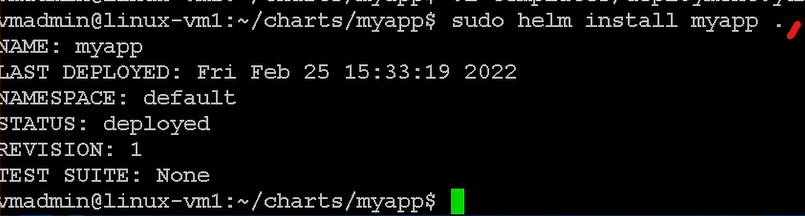
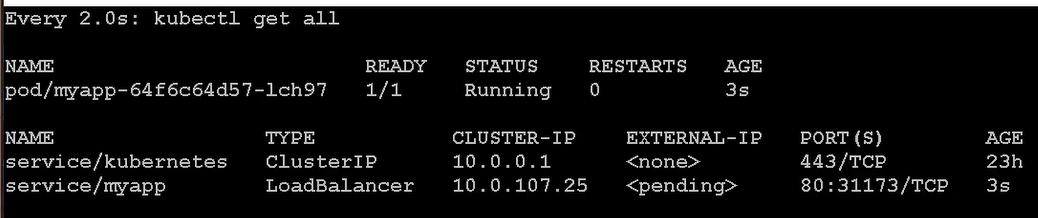
# in this values.yaml, we can define variable  
# vi values.yaml  


This variable I want to refer in deployment.yaml file  
vi template\deployment.yaml

replicas: {{ .Values.replicaCount}}

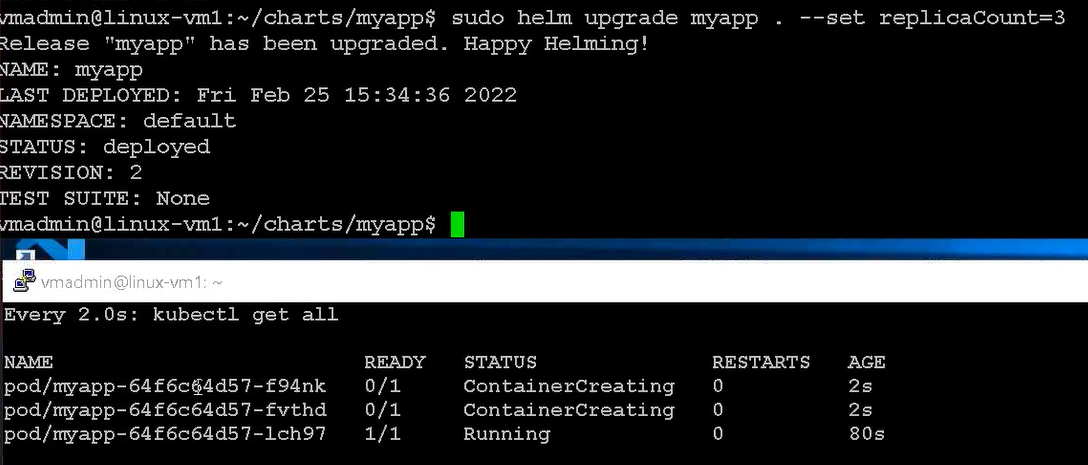


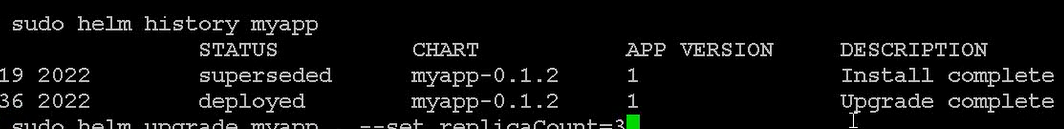
# sudo helm install myapp .

  
I

I can even upgrade by helm passing this arugement as well, as we have parametrized values

# sudo helm upgrade myapp . --set replicaCount=3  
  
the number of pods will increase to 3



# sudo helm history myapp  


sudo helm upgrade myapp . --set replicaCount=5  
  


now I am rolling back to revision 1



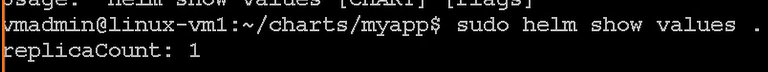
Let me uninstall myapp

sudo helm uninstall myapp

for example , if we have 10 variables to pass to my template, is this possible

I want to see , what are the variable they have defined in the values

myapp# sudo helm show values .

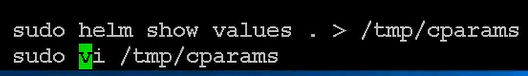


I can write all these variable into my own file

sudo helm show values . > /tmp/cparams

Modify replicaCount in cparams

# vi /tmp/cparams





# sudo helm install myapp .

It will install normally

# sudo helm upgrade myapp .   
  
sudo helm upgrade myapp . --values /tmp/cparams



sudo helm uninstall myapp

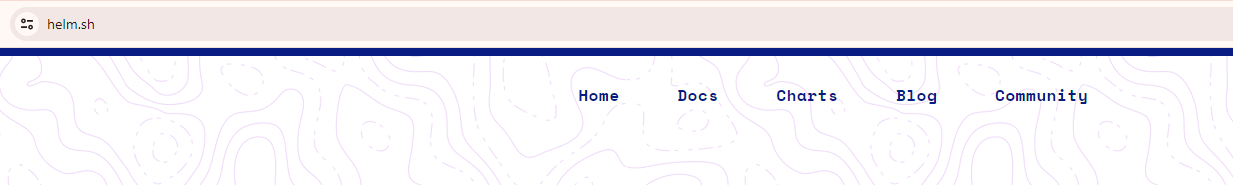
Now will talk about dependency management..

for example my chart is depending on another chart

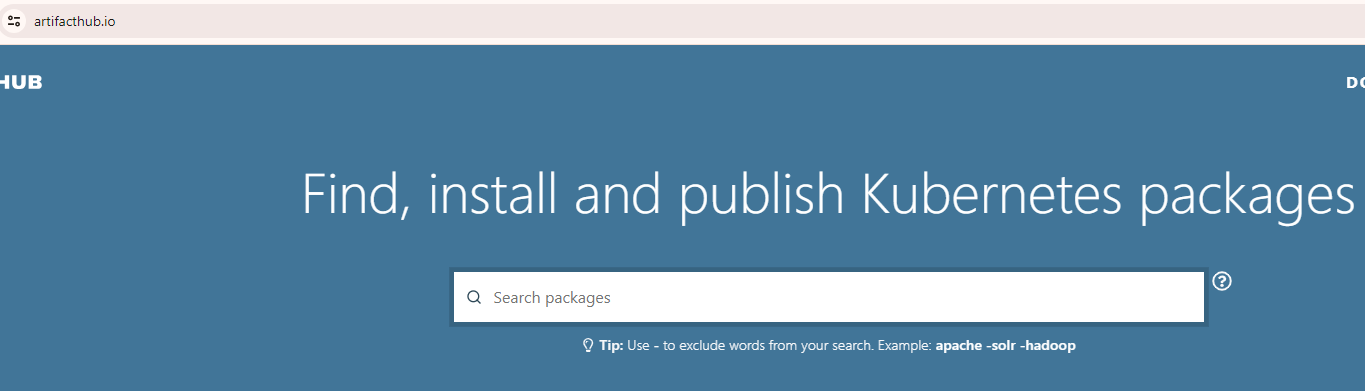
before deploying my chart, I need to deploy another chart.

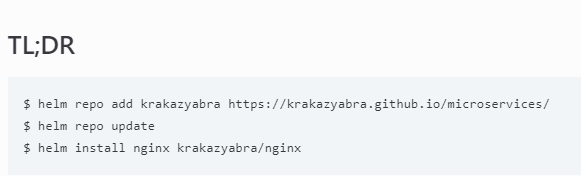
You don’t need to install, but you can sepecify..

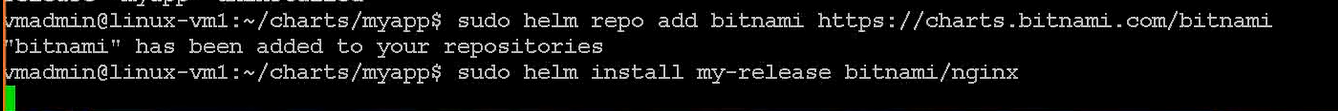
You have lot of charts like browse helm.sh in browser



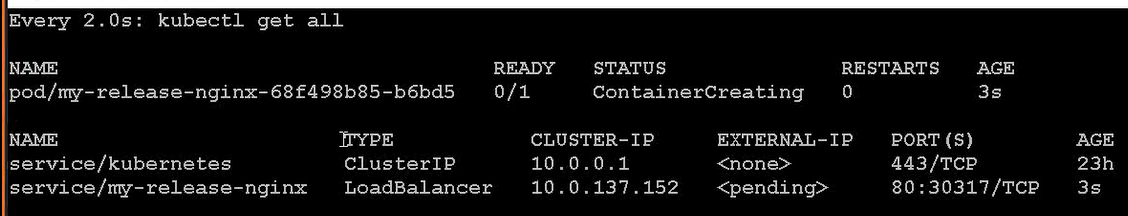
There are thousands of charts available here



Let me search nginx official  


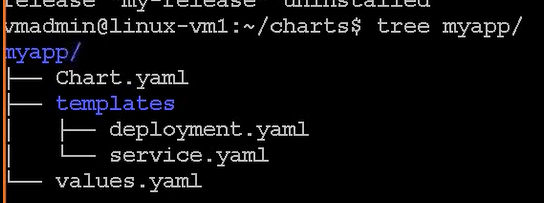


Everything is define the chart, it will create pod and service

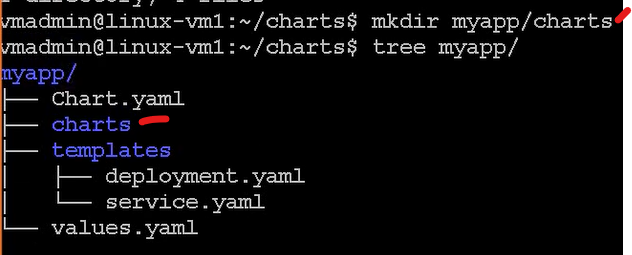


# sudo helm repo list



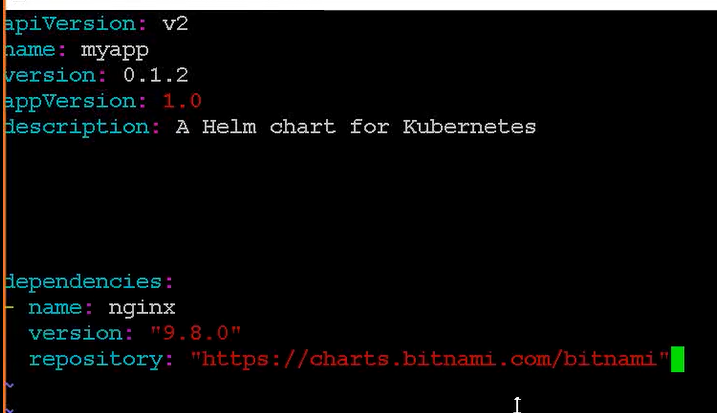


I will create chart folder



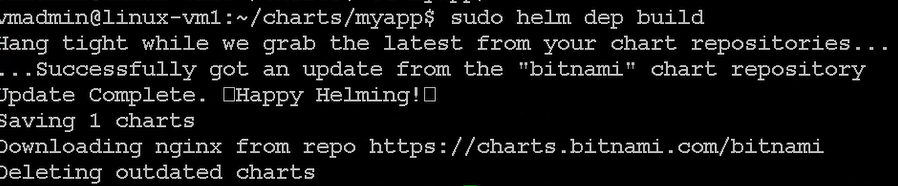
Add dependency in the chart.yaml file





Now you need to build dependnecies

sudo helm dep build

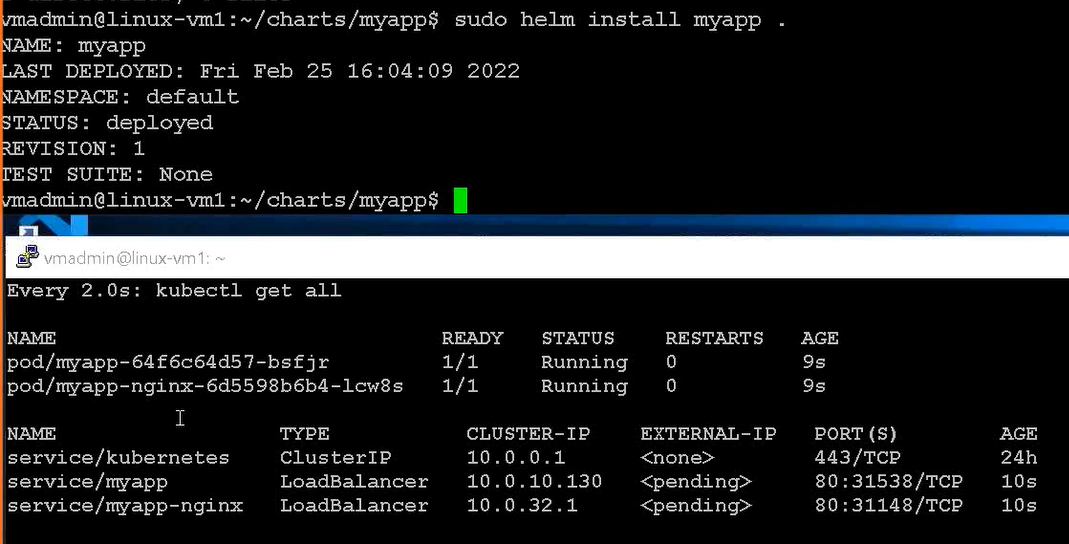


Now you can see, in the charts folder, you wil have zip file



# when you install..

First it will install nginx pod and then myapp pod.



# sudo helm uninstall myapp

