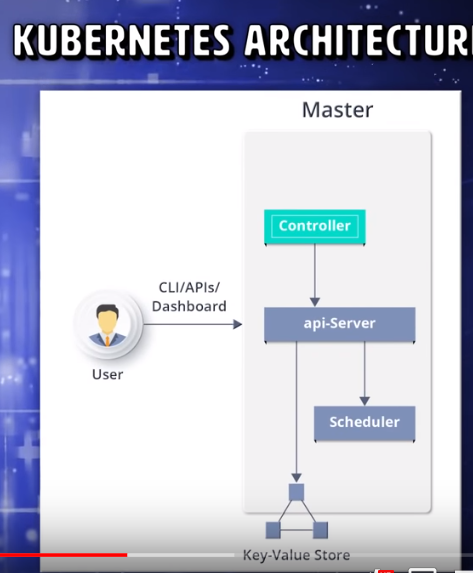
Kubernetes notes ::==========================

1. 3 default name spaces: kube-system public and default
2. Important terms :prod release , deploy , staging
3. Continuous delivery can be fully automated with a workflow process or partially automated with manual steps at critical points
4. When continuous delivery is properly implemented, developers always have a deployment-ready build artifact that has passed through a standardized test process. With continuous deployment, revisions are deployed to a production environment automatically without explicit approval from a developer, making the entire software release process automated. This, in turn, allows for a continuous customer feedback loop early in the product lifecycle.
5. Continuous Delivery Is Not Continuous Deployment

CICD pipeline:

1. 

K8S : Master node , worker node , distributed Key value pair ( ETCD)



API Server >

Scheduler > which assign work to the worker node

Controller: watch the current state of the nodes and make sure that it is exactly same as desired state

If not it will take control and adjust it through API servers

ETCd > cluster state and configuration details

=====================================================================================K8s > Horizontal Pod auto scalar > based on load , cpu utilization scale the pods , when CPU utilization is above the threshold it will auto scale

Metric server : disk io, cpu utilization , memory usage , HPA will check the matrix servers and based on that autos scale : we need to set the threshold

Cooling period : 3 min for up, 5 min for down.. that means once the load increase it wait 3 min and launch the new pods.. and it will wait another 3 min before next auto scale, same for scale down

And in 15 seconds HPA will kick start

Deploying metrics server : Go to github and clone it and edit the server.deployment.yaml

And resolve the certificate issue before you proceed further

Resource limit is must to work HPA .. without setting that it wont work

**DOCKERS :**

How to build envrionment-agnostic systems with Docker?

Using these 3 features:

Volumes

environment variable injection

Read-only file systems

**RUN** lets you execute commands inside **of** your **Docker** image. ... **CMD** lets you define a default **command** to **run** when your container starts. You could say that**CMD** is a **Docker run**-time operation, meaning it's not something that gets executed at build time. It happens when you **run** an image.

The key **difference** between **stateful** and **stateless** applications is that **stateless**applications don't “store” data whereas **stateful** applications require backing storage.

**Docker Flow:**

\Everything starts with the **Dockerfile**. The Dockerfile is the source code of the Image.Once the Dockerfile is created, you build it to create the **image** of the container. The image is just the "compiled version" of the "source code" which is the Dockerfile.Once you have the image of the container, you should redistribute it using the **registry**. The registry is like a git repository -- you can push and pull images.

Next, you can use the image to run **containers**. A running container is very similar, in many aspects, to a virtual machine (but without the hypervisor).

+------------+ docker build +--------------+ docker run -dt +-----------+ docker exec -it +------+

| Dockerfile | --------------> | Image | ---------------> | Container | -----------------> | Bash |

+------------+ +--------------+ +-----------+ +------+

^

| docker pull

Difference between image and layer :

Image: A Docker image is built up from a series of read-only layers

Layer: Each layer represents an instruction in the image’s Dockerfile.

**How will you monitor docker in prod ?**

Docker State > tells us cpu utilization, memory uti. etc

Docker event > tells us set of events running in docker daemon

**Orphant volume : volume without a container**

**How to remove it :** docker image prune

Paravirtualization > type 1 hypervisor which runs directly on bare-metal

**What is the use of Onbuild instruction :?** The ONBUILD instruction adds to the image a trigger instruction to be executed at a later time, when the image is used as the base for another build. This is useful if you are building an image which will be used as a base to build other images, for example an application build environment or a daemon which may be customized with user-specific configuration.

**Is it good practice to run stateful applications on Docker? What are the scenarios where Docker best fits in?**

the problem with statefull docker aplications is that they by default store their state (data) in the containers filesystem. Once you update your software version or want to move to another machine its hard to retrieve the data from there.

What you need to do is bind a volume to the container and store any data in the volume.

if you run your container with: docker run -v hostFolder:/containerfolder any changes to /containerfolder will be persisted on the hostfolder. Something similar can be done with a nfs drive. Then you can run you application on any host machine and the state will be saved in the nfs drive.

**How containers works at low level :**

This is how Docker works: Each container runs in its own namespace but uses exactly the same kernel as all other containers. The isolation happens because kernel knows the namespace that was assigned to the process and during API calls it makes sure that process can only access resources in its own namespace.

**We can create docker image tar file using below command**

docker save - o <path for generated tar file> <image name>

Then copy your image to a new system with regular file transfer tools such as cp or scp. After that you will have to load the image into Docker:

docker load -i <path to image tar file>

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ***:* What is Build Cache in Docker?** | |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Add to PDF/md** |  |  |  | **Junior** | |

When we build an Image, Docker will process each line in Dockerfile. It will execute the commands on each line in the order that is mentioned in the file. But at each line, before running any command, Docker will check if there is already an existing image in its cache that can be reused rather than creating a new image.

**What is the difference between ‘docker run’ and ‘docker create’**

Create > container will be in stoped state and later we can run it using docker id

**docker inspect NAME|ID >> It will show where docker files are storred > default /var/lib/docker**

Docker file is not case sensitive

Entrypoint command will override CMD command

Multiple container can share the same volume.. Just use VOLUME command in docker file and specify the path

How to stop and restart the Docker container ?? > Docker start/stop <container ID >

Out of the box, **Docker** on **Windows** only **run Windows** container. To use Linux containers on**Windows Server**, you need to use the **Docker**Enterprise Edition Preview which includes a full LinuxKit system for **running Docker** Linux

What is Scrum ?

Scrum is basically used to divide your complex software and product development task into smaller chunks, using iterations and incremental practises. Each iteration is of two weeks. Scrum consists of three roles: Product owner, scrum master and Team

git branch –merged

The above command lists the branches that have been merged into the current branch.

git branch –no-merged

this command lists the branches that have not been merged.

How to check the exit status of the commands ?

Echo $

How to pass the parameters to the script and how can I get those parameters?

Scriptname.sh parameter1 parameter2

I will use $\* to get the parameters.

How you will do the releases?

There are some steps to follow.

* Create a check list
* Create a release branch
* Bump the version
* Merge release branch to master & tag it.
* Use a Pull request to merge the release merge
* Deploy master to Prod Environment
* Merge back into develop & delete release branch
* Change log generation
* Communicating with stack holders
* Grooming the issue tracker

How can I integrate all the tools with Jenkins?

I have to navigate to the manage Jenkins and then global tool configurations there you have to provide all the details such as Git URL , Java version, Maven version , Path etc.

My application is not coming up for some reason? How can you bring it up?

We need to follow the steps

* Network connection
* The Web Server is not receiving users’s request
* Checking the logs
* Checking the process id’s whether services are running or not
* The Application Server is not receiving user’s request(Check the Application Server Logs and Processes)
* A network level ‘connection reset’ is happening somewhere

I want to copy the artifacts from one location to another location in cloud. How?

Create two S3 buckets, one to use as the source, and the other to use as the destination and then create policies.

[**Delete Files Older Than x Days on Linux**](https://www.howtogeek.com/howto/ubuntu/delete-files-older-than-x-days-on-linux/)

**find /path/to/\*log -mtime +5 -exec rm {} \;** it delete log files older than 5 days

How you will take backup for Jenkins?

Copy  JENKINS\_HOME directory and “jobs” directory to replicate it in another server

How to deploy docker container to aws?

Amazon provides the service called Amazon Elastic Container Service; By using this creating and configuring the task definition and services we will launch the applications.

I want to change the default port number of apache tomcat. How?

Go to the tomcat folder and navigate to the conf folder there you will find a server.xml file. You can change connector port tag as you want

How to access variable names in Ansible?

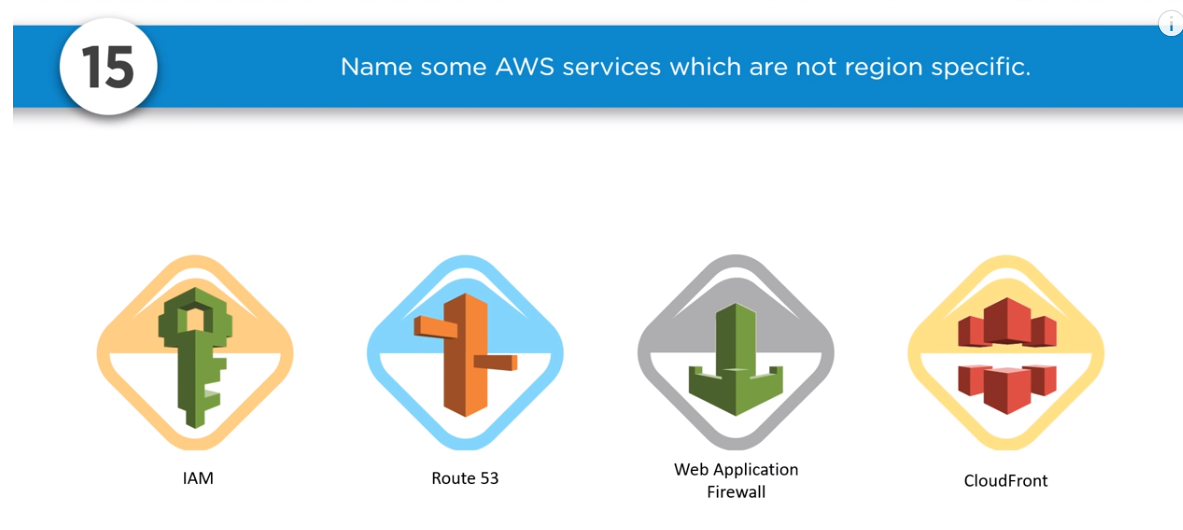
Using hostvars method we can access and add the variables like below

{{ hostvars[inventory\_hostname][‘ansible\_’ + which\_interface][‘ipv4’][‘address’] }}

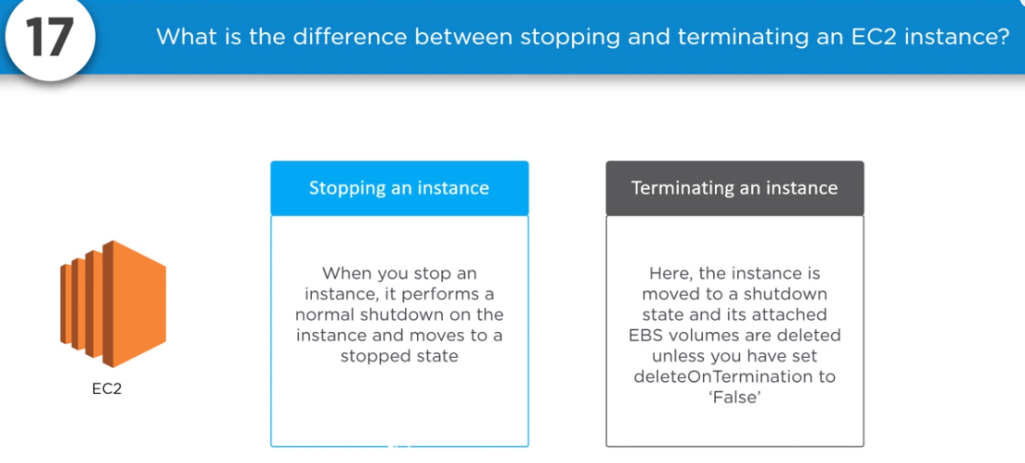
**<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<AWS >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>**

**Geo targeting >**  for ex: use languages used in that area to increase popularity

CloudFront > It’s a caching servers which help user to get data from nearest available server(low latency)



NAT gateway : upto 45gbps managed by amazon, canot assign security group



3 types of EC2 instance :

On demand: fixed rate , good for testing and dev

Spot instance : based on bidding , not permanent , supply and demand market , it will not with u all the time

Reserved instance : for long term cheap > but instance for 1 year at at time , low medium high 3 type

Solaris and AIX not supported by AWS

Jenkins test modules : Junit > for unit testing report, Sonarqube > static code analysis , Selenium >computerization test

 How to Version control Docker images ? >> Use docker tag <image id> command

Difference between “poll scm” and “build periodically” ?????

Poll SCM will trigger the build only if it detects the change in SCM, whereas Build Periodically will trigger the build once the given time period is elapsed.

syntax for building docker image

docker build –f <filename> -t imagename:version

running docker image

docker run –dt –restart=always –p <hostport>:<containerport> -h <hostname> -v <hostvolume>:<containervolume> imagename:version

How to log into a container

docker exec –it <containerID> /bin/bash

How Can Make a Sure New Service Is Ready For The Products Launched?

* Backup System
* Recovery plans
* Load Balancing
* Monitoring
* Centralized logging

What is git rebase?

Process of moving or combining a sequence of commits to a new base commit

Cloud watch .. > Used to launch automatic instance when it fails .. and threshold is met

Types of AWS AMIs : Amazon Machine Image

Fully baked > ready to use everything pre built

Just enough AMI > minimal OS that is it

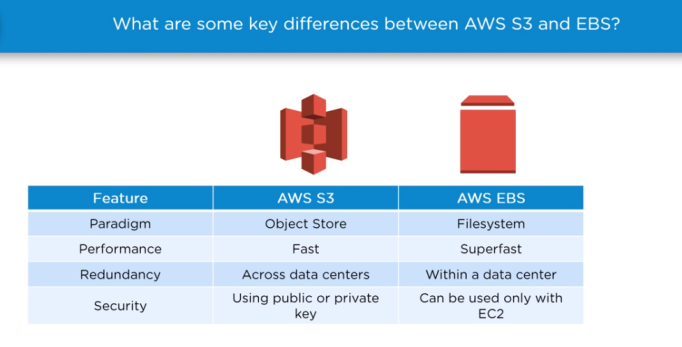
Hybrid

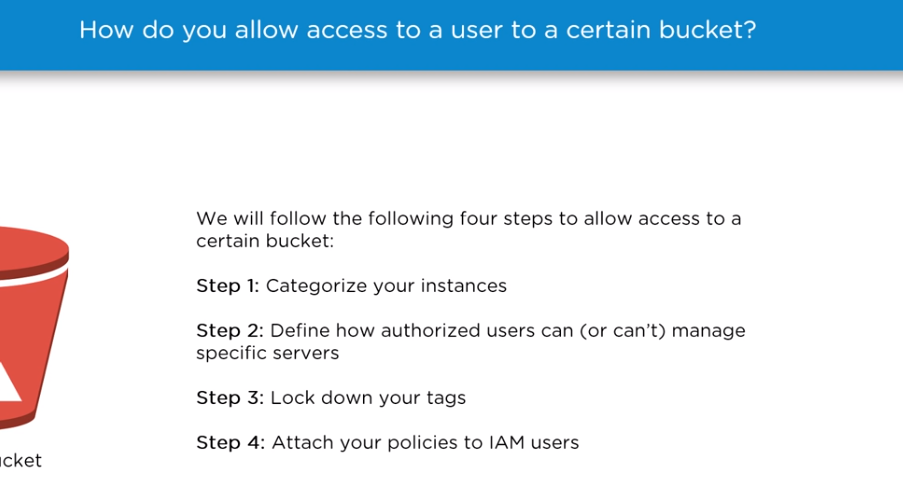
If you lost AWS key how will you login > ?? instead of key if it was a password then we can Use AD (active directory ) to reset it

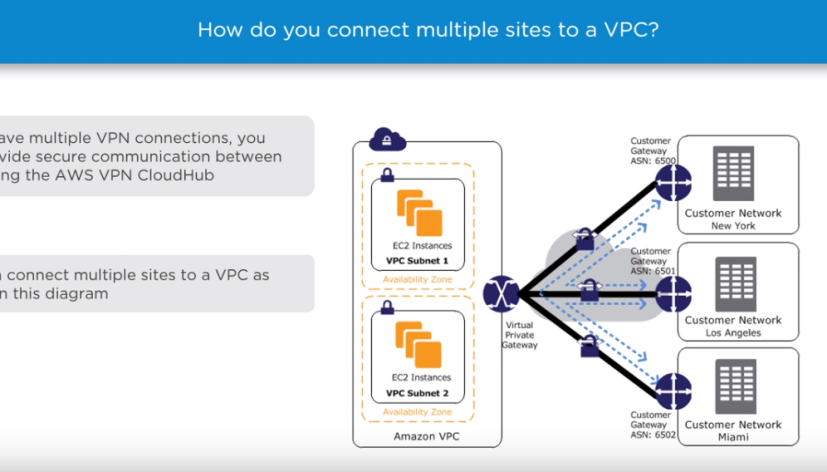
Or we can take back up of instance key (mirroring)

S3 > Object store > cant install anything.. it is just for a storage

EBS > install store everything







Sercurity in VPC > security groups – acts as a firewall

ACLs > control both inbound and outbound subnet

<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<Maven >>>>>>>>>>>>>>>>>>>>>>>>>>>>

Groupid > com.comanyname.project

Artfifact id > application

Mvn archetype:generate

POM>XML > groupid artifactid dependencey version credentials. And remote repo url mention all these

Some of the Important goals :

Clean validate complie test package verify install deploy

DskipTest=true > compile but will not run it

Dmaven.test.skip=true > skip compiling the test and running them

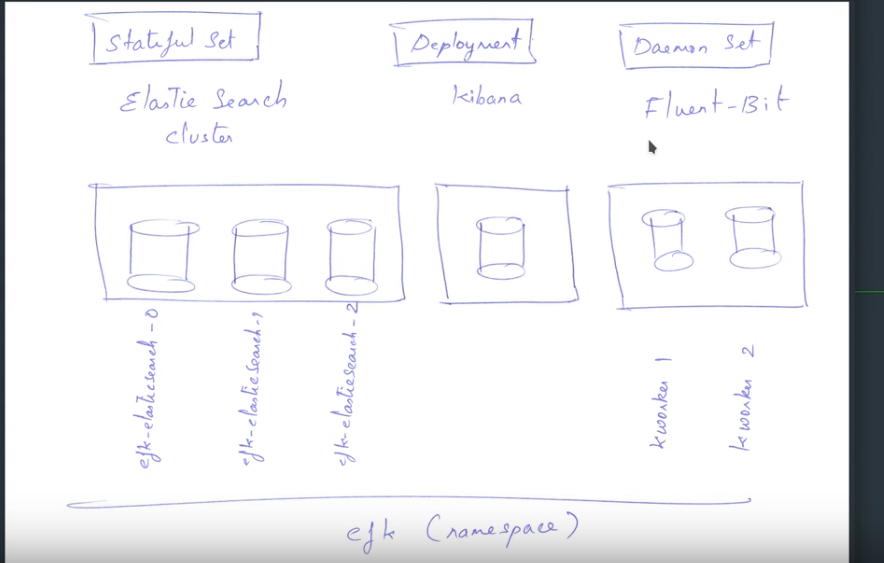
Mvn dependency:analyze > alalyze the dependency

Mvn dependency:tree > print the dependency

<<<<<<<<<<<<<<<<<<<<<<<<<EFK>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

If you got a client then sending logs and monitoring logs is necessary ☺

Deployed the EFK and promethius as a HELM chart



Fluent bit will be in each worker node which will collect the logs

<< Side Note > Add cluster to rancher >> follow instruction and rancher will give one Token / long commands and you run it on master node and it will setup automatically >>

Cluster IP > to communicate between pods

NodePort > if u wan to access it from outside network

<<<<<<<<<<<<<<<<<<<<<<HELM >>>>>>>>>>>>>>>>>>

It contains the vaules file ex: name.values we can modify this ans start our own helm chart

Ex : helm install stable/grafana --name grafana1 --values /tmp/graf.values -n grafana\_namespace

<<<<<<<<<<<<<<<<<<KIBANA>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

You can search the logs from particular pod

To install just follow the on screen steps and select time stramp

You can add various filters as well

<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<EFK >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

* [Fluentd](https://www.fluentd.org/) is an open source data collector providing a unified logging layer, supported by 500+ plugins connecting to many types of systems.
* [Elasticsearch](https://www.elastic.co/products/elasticsearch) is a distributed, RESTful search and analytics engine.
* [Kibana](https://www.elastic.co/products/kibana) lets you visualize your Elasticsearch data

<<<<<<<<<<<<<<<<<<<<<<<Prometheus and Grafana >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

Rbac > role based access control

Grafana > add data source > and follow the onscreen steps to complete the installation

And create new dashboard > get the dashboard id from grafana website select which one you want to choose and in grafana create dashboard, add the ID and continue

Or you can create your own dashboard using JSON file upload ( custom dash board )

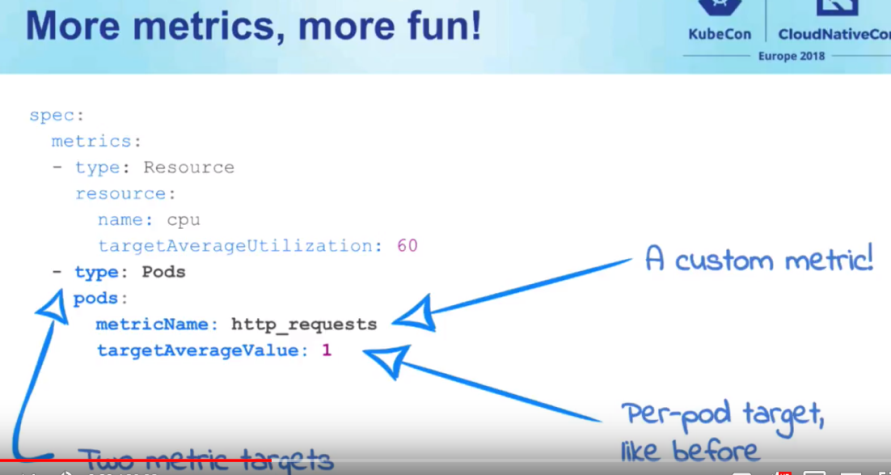
**We use custom matrix to auto scale the application in kubernetes and ptometheus helps us to do that**

<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<HPA >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

V1 > only based on cpuutilization

V2 > in beta > it support more custom matrix

Matrix > it is a list now



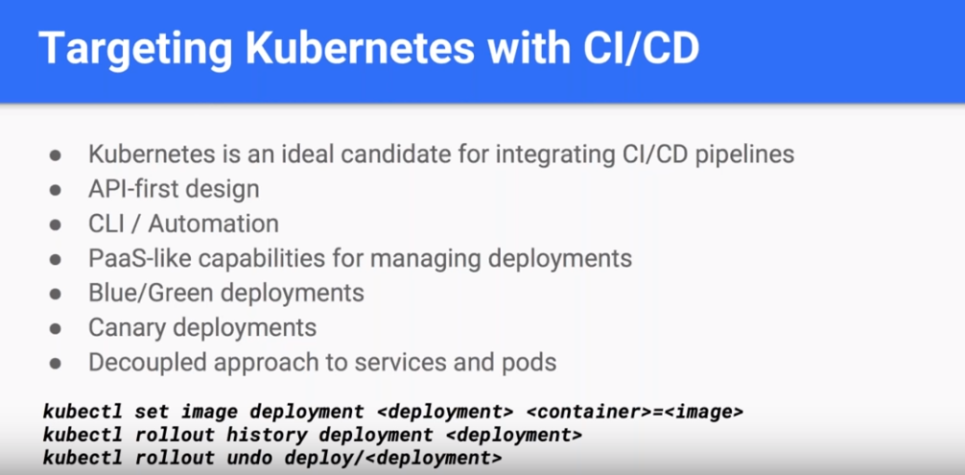
HPA with Prometheus

1. Update and install the Prometheus-Adapter helm chart for custom metrics to be consumed by HPA.
2. Define and create the custom metric
3. Create the HPA object
4. Testing with the scale out.

<<<<<<<<<<<<<<<<<<<<<<<<<<<**CICD**>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

Earlier we used to build JAR WAR files and deploy it in prod, but now we just create a docker image instead

Now we make standardized image and dploy in containers

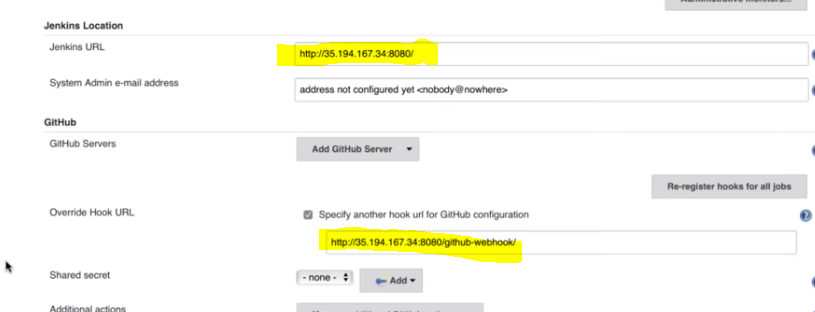


Kubectl set image >>>> Used to upgrade pod with new version / new update > its like live update

Rollout based on policy, and we can undo rollout as well ☺

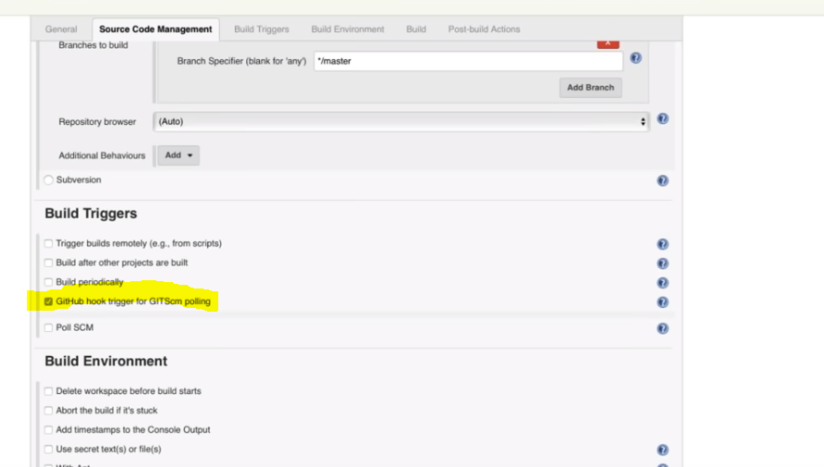
<<<<<<<<<<<<<<<<Docker login –u <username> -p <password> >>>>>>>>>

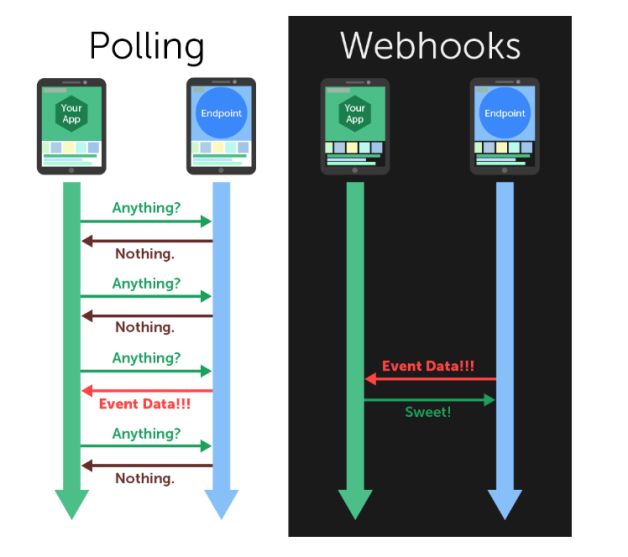




Githubweb hook url and Jenkins url both are same

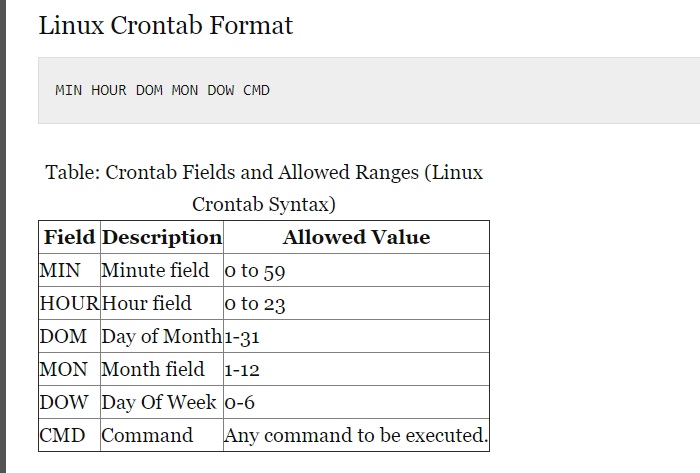
GITHUB repo > setting > integration and service > Add service >> select Jenkins > configure webhook





Poll SCM > it send data in predefined interval and most of them are wasted

Webhook > when there is only new content , it send the data

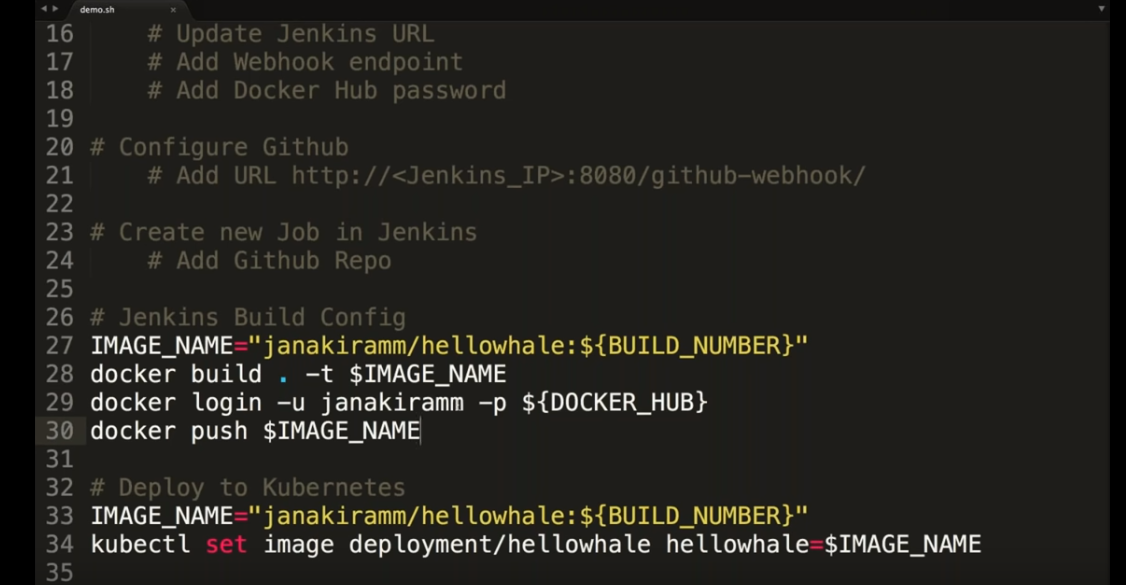


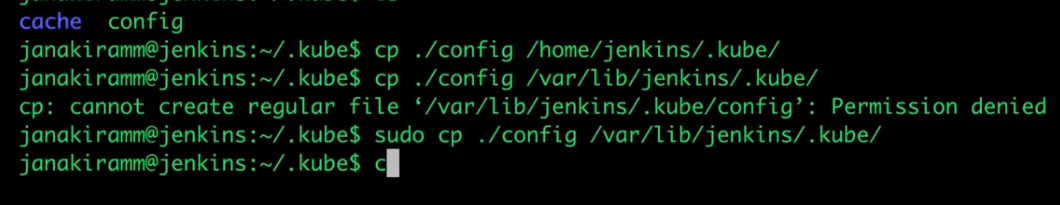
<https://www.thegeekstuff.com/2009/06/15-practical-crontab-examples>

How to Execute a Linux Command After Every Reboot ???????

@reboot CMD

<<<<<<<<<<<<<<<<<<<CICD – Git >Jenkins build to Docker image and to K8s>>>>>>>>>>>>





Copy these file to above specified location so that Jenkins can now communicate with Kubernetes

<<<<<<<<<<<<<<<<<<<<<<<<<<<<<Jenkins File integration with GIT >>>>>>>>>>>>>>>>>>>>>>>>>

<https://www.youtube.com/watch?v=56jtwSrNvrs>

<<<<<<<<<<<<<Types of images Used >>>>>>>>>>>>>>>>

If it is java based then openjdk image

If it is go lang based then golang image

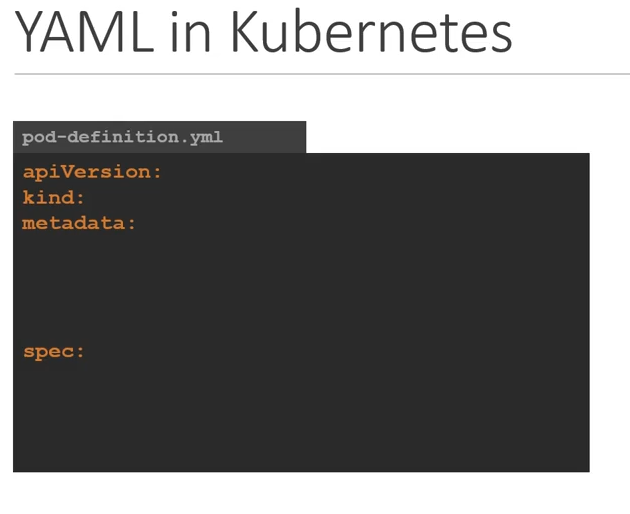
Otherwise ubuntu or alpine

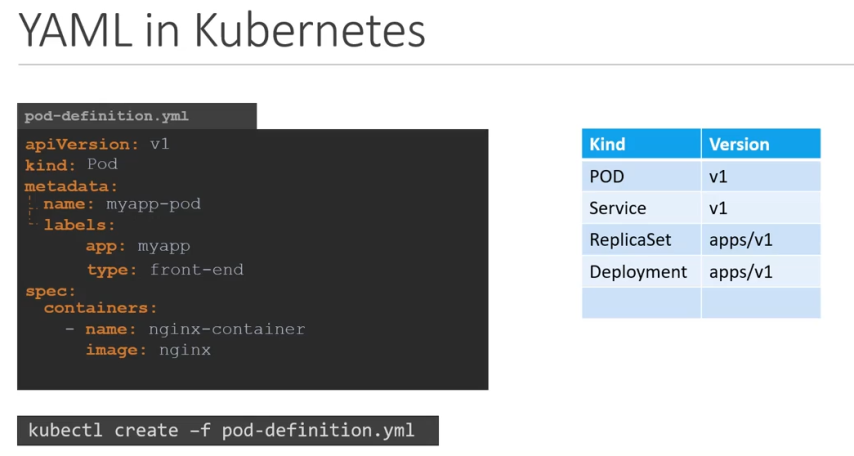
<<<<<<<<<<<<<<<<<<<<<<Jenkins security >>>>>>>>>>>>>>>>>>>>>>>>

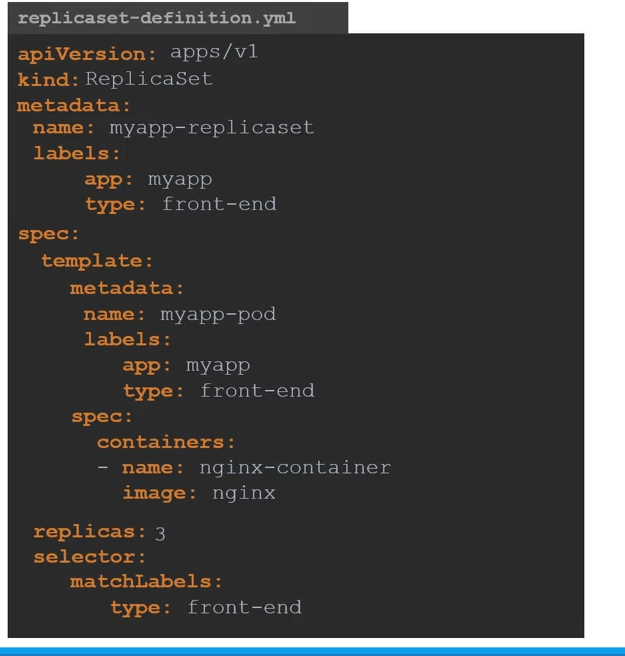
Used Ldap integration and matrix based security ☺

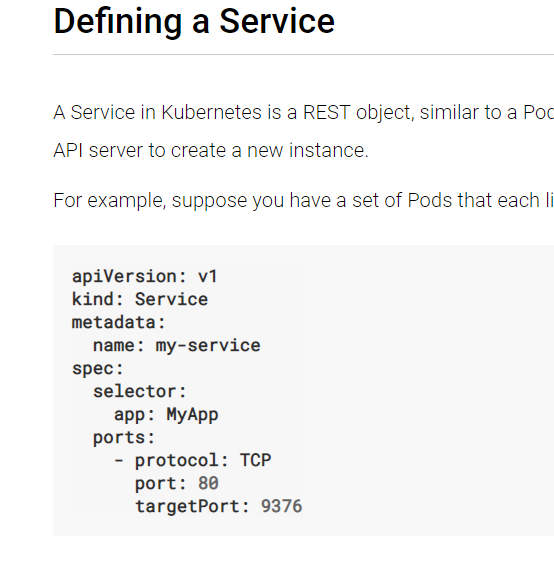
:Install LDAP server in environment and then integrate with Jenkins > manage Jenkins and global security > provide root dn and manager dn and all the necessary details and click apply

<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<YAML in K8S >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>



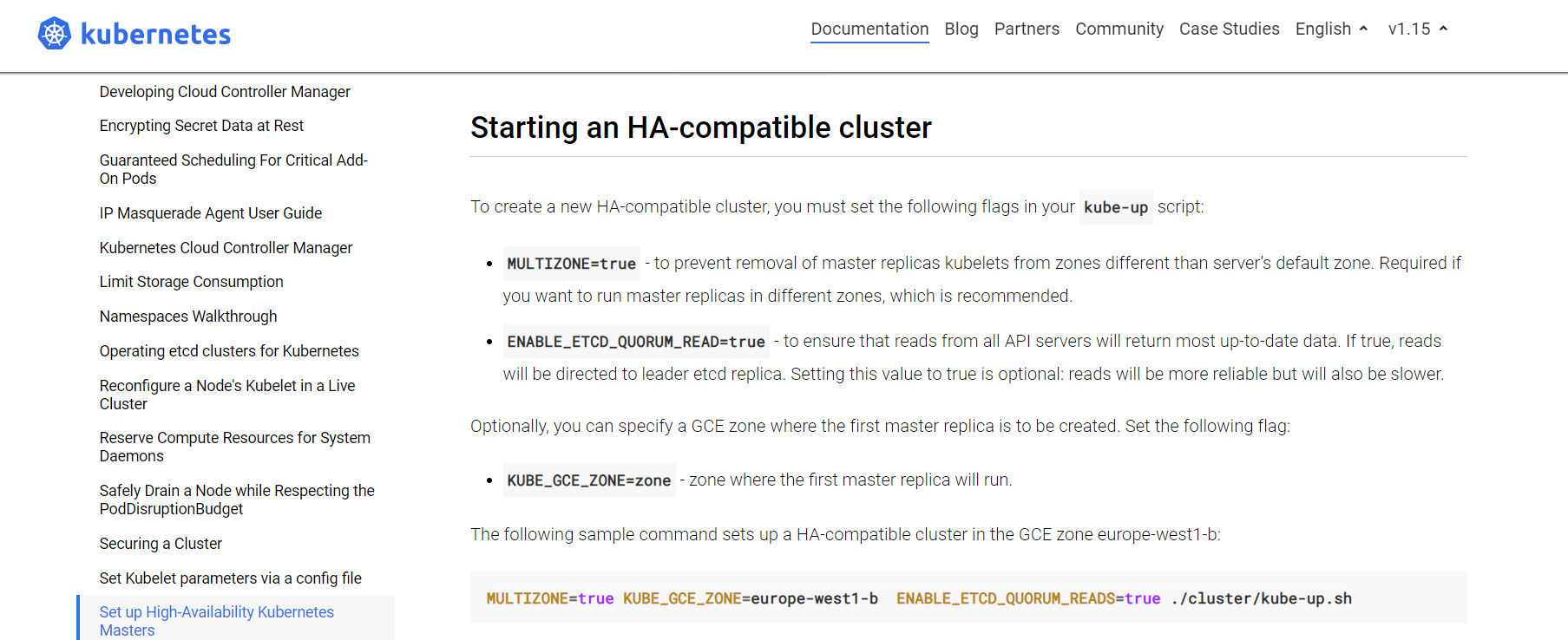


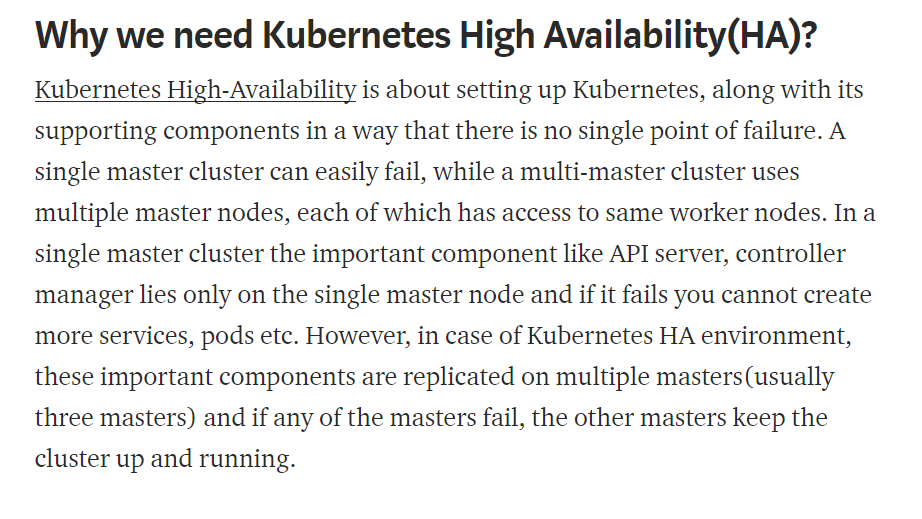




Replica set creates only replicas

Deployment helps us to do rolling update and undo and many other thing

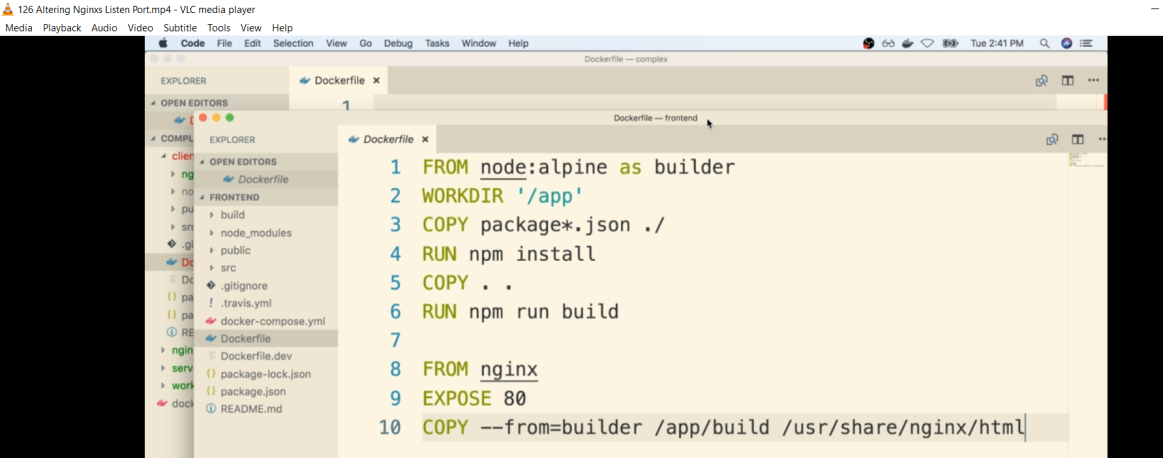




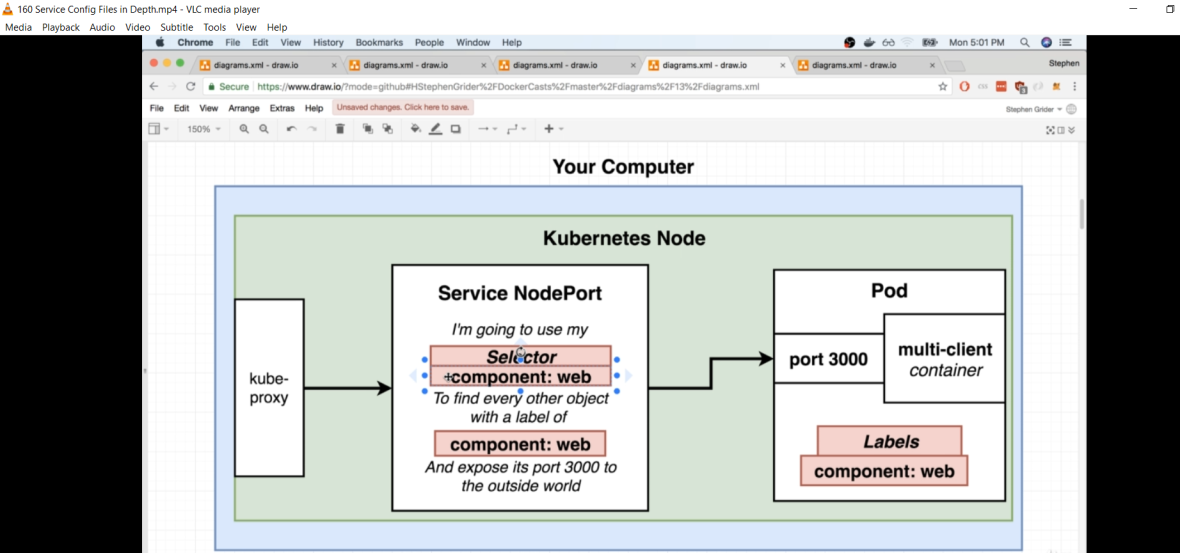
<<<<<<<<<<<<<<<<<<<<<<SonarQube>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

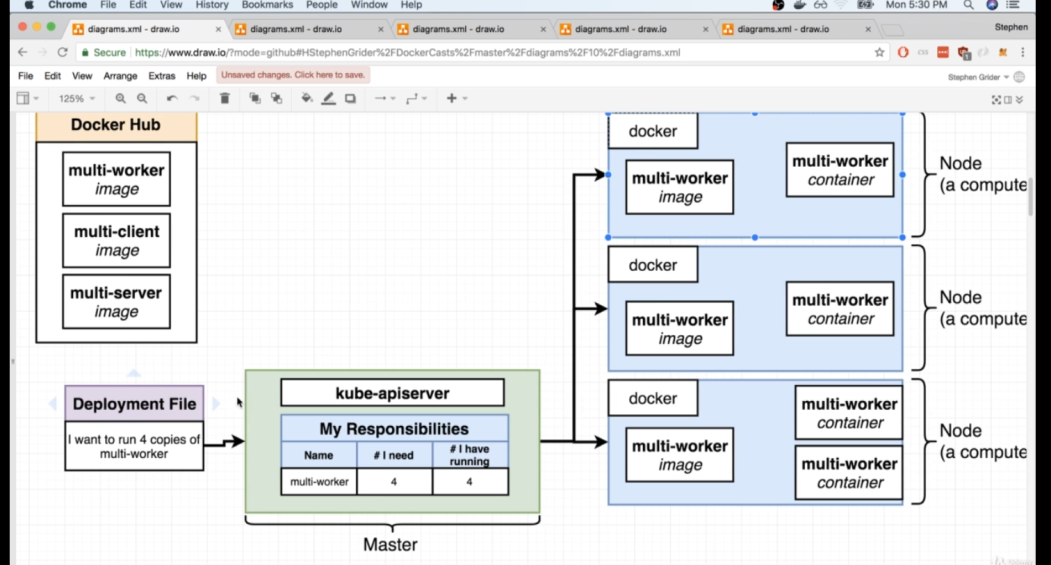
SonarQube cannot be run as root on Unix-based systems, so create a dedicated user account to use for SonarQube if necessary.

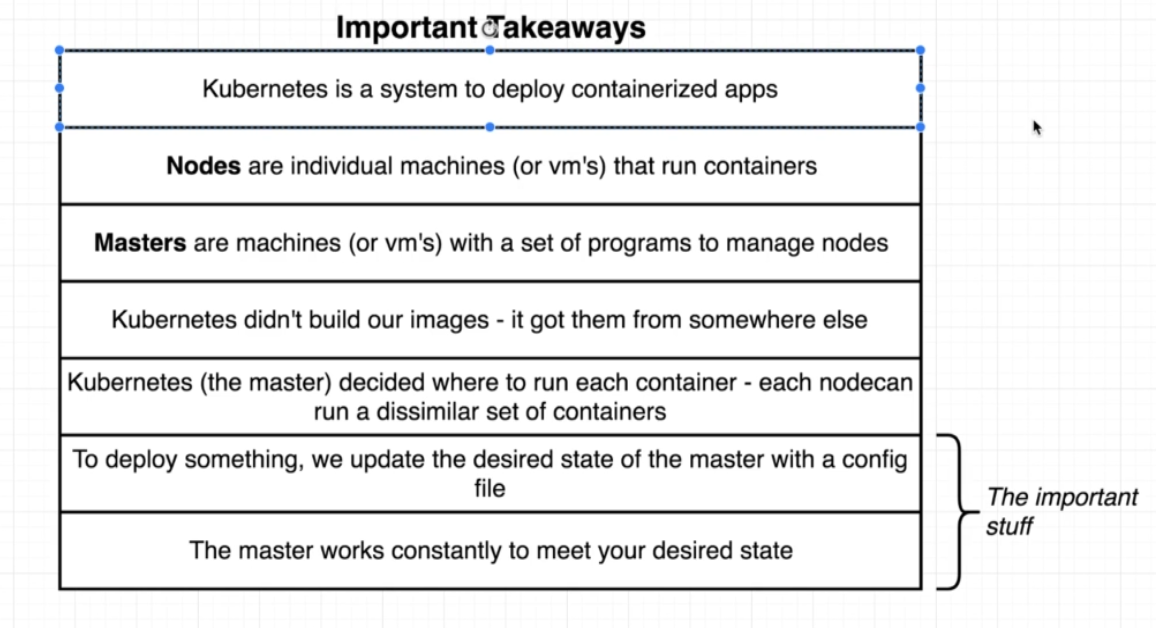
Multiple container build and some code example



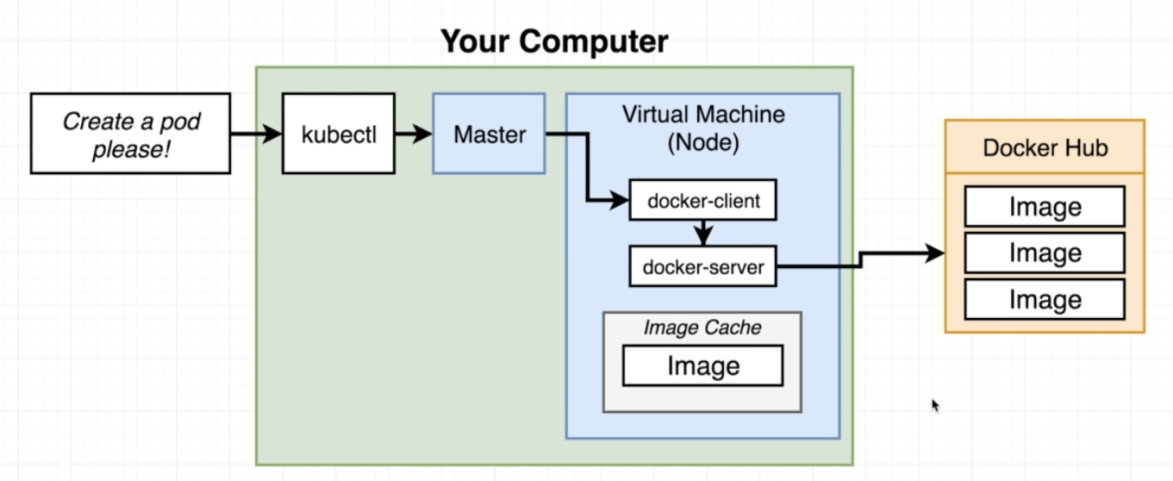
K8S <<<<<<<<<<<<<<<<<<<<<SELECTOR and LABEL >>>>>>>>>>>>>>>>>>>>>>>>

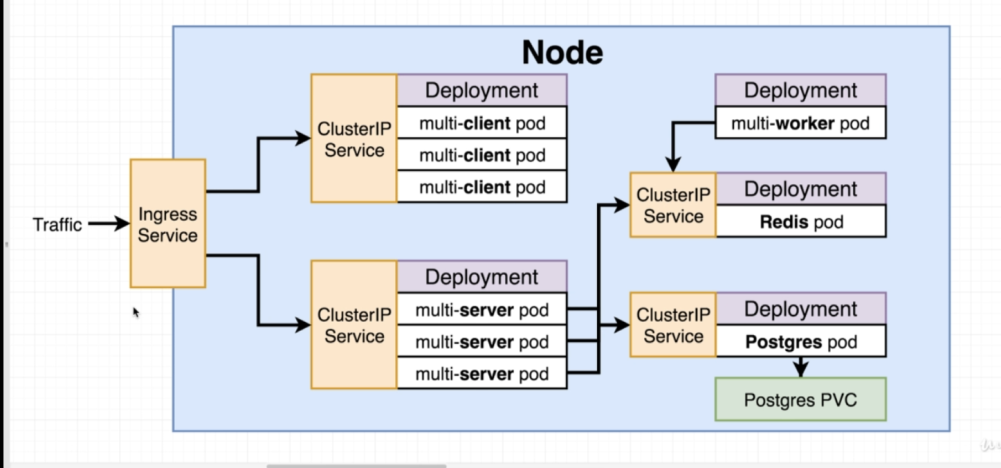












INGRESS <<<<<<<<<<<<<<<<<<<<<<<<>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

