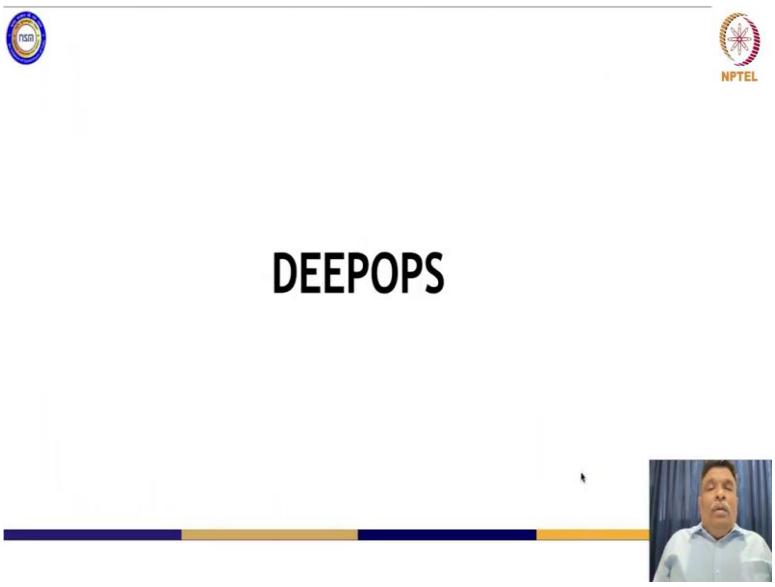
Applied Accelerated Artificial Intelligence
Prof. Satyadhyan Chickerur
School of Computer Science and Engineering
Indian Institute of Technology, Madras

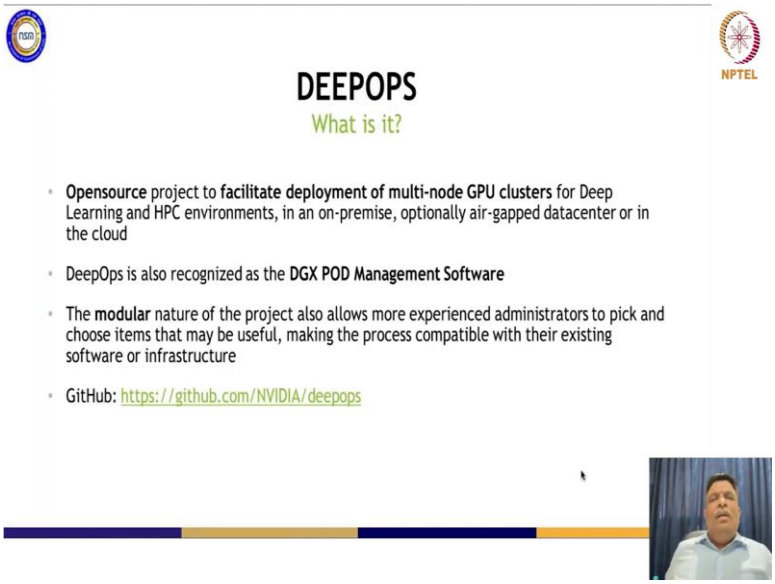
Lecture - 13
DeepOps: Deep Dive into Kubernetes with deployment of various AI based Services
Session II - Kubernetes Part - 2

(Refer Slide Time: 00:15)



So, coming now from the concept of DevOps we reached the concept of MLOps and then now there is next concept of DeepOps which actually involves deep learning. And since deep learning is involved we would be having lot of GPUs with us right. So, those GPUs can be of any make anything.

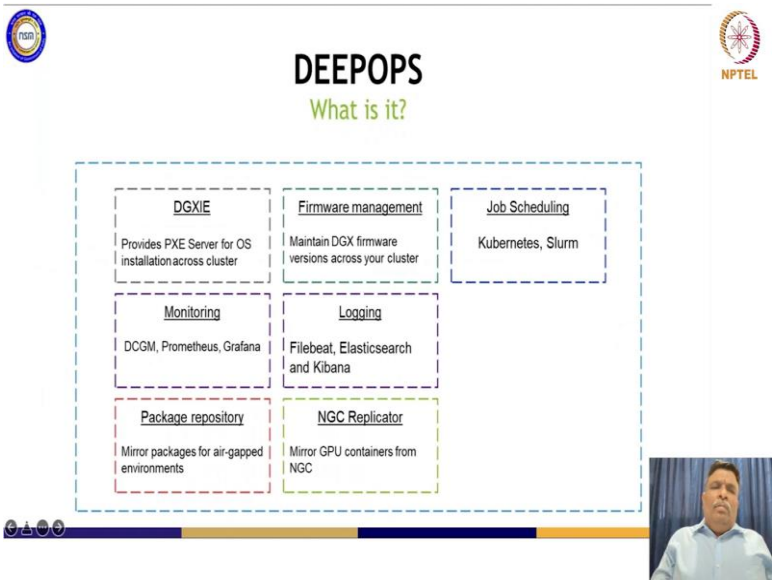
(Refer Slide Time: 00:40)



But DeepOps is basically a open source project ok from NVIDIA which facilitates deployment of multi node GPU clusters for deep learning and HPC environment in an on-premise optionally air gapped data center or in the cloud. So, the idea is you can have a physical on premise data centre or basically in the cloud. So, this is a open source project.

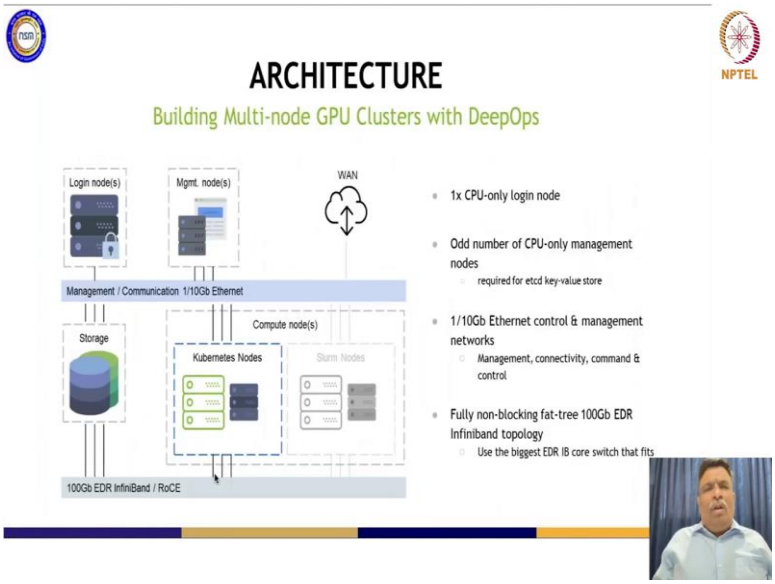
Now this actually is used for managing pods on the DGX. So, at present if you are not able to understand the architecture of DGX cluster, let us not worry too much about it. But let us try to understand that any node which is a worker node has to run some POD on it ok. It may use a GPU it may not use a GPU and that modularity of the project ok will be of lot of help to these administrators who actually maintain right this infrastructure and software. So, this is the GitHub link to DeepOps.

(Refer Slide Time: 02:07)



Now there are so many things which DeepOps can do. So, it can do the job scheduling using Kubernetes Slurm you can maintain the firmware of the DGX you can do monitoring using DCGM and Prometheus and Grafana. So, we will see this actually today monitoring and then you can do Filebeat Elasticsearch and all of this as to how do you log then you can maintain and manage various packages ok so on and so forth.

(Refer Slide Time: 02:44)

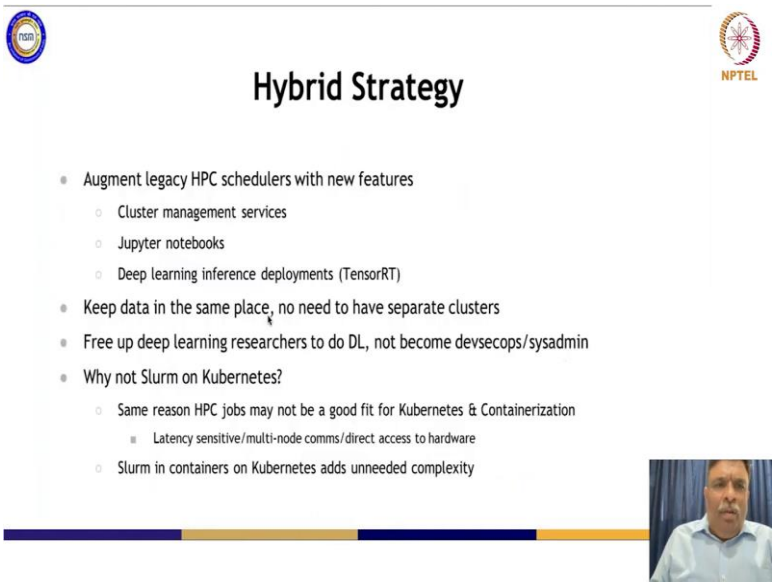


So, now this is the architecture you have got the login node you have got the management node and you have got the Kubernetes node and these Kubernetes node are

actually a part of the compute node. So, you will have the Slurm nodes you will have Kubernetes nodes because we know that any docker thing which is not generated by Kubernetes is not going to be maintained by the Kubernetes cluster.

So, you can actually take a docker image and run it, but if Kubernetes has to manage it it has to be actually what to say started or be done on a Kubernetes through a Kubernetes image right. So, something like that. So, here I am just going into the basic of this as to how DeepOps came into this thing.

(Refer Slide Time: 03:36)



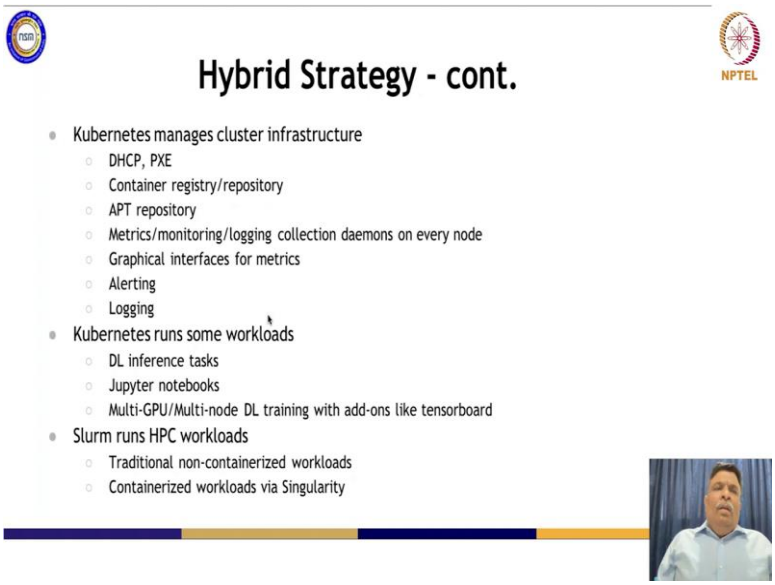
And now there is a idea of hybrid strategy wherein we talk of ok. So, we have this augment legacy of HPC schedulers with new features. So, we have cluster management services we have Jupyter notebooks we have deep learning inference deployments right.

So, you have to do inferencing. Inferencing is nothing, but whatever is being done by the servicing pipeline ok or you need to actually deploy it. So, we use Tensor RT and then. So, that basically means you should ensure how do you deploy. So, we need to also see that how the data has to be kept in the same place no need to have separate clusters free up deep learning researchers to do DL not become devsecops or system admins why not Slurm on kubernetes.

So, there these are all things which are still in discussion or debatable based on the flexibility of the system based on how the user wants to use it and perceive it right. So,

that is how it is, but the idea is that all of this are going to make our lives easy for maintaining things which we need to do ok.

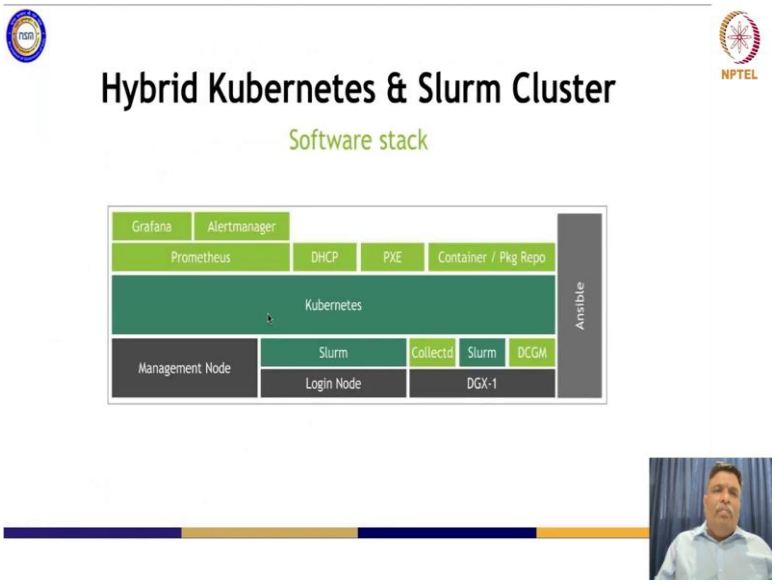
(Refer Slide Time: 04:59)



So, we are talking of Kubernetes managing the cluster infrastructure it can run some workloads we will have DL inference tasks we can run Jupyter notebooks we can do multi GPU multi node deep learning training right and we can do tensorboards there can. So, many graphs. So, many charts ok all of this. So, Slurm runs HPC workload right.

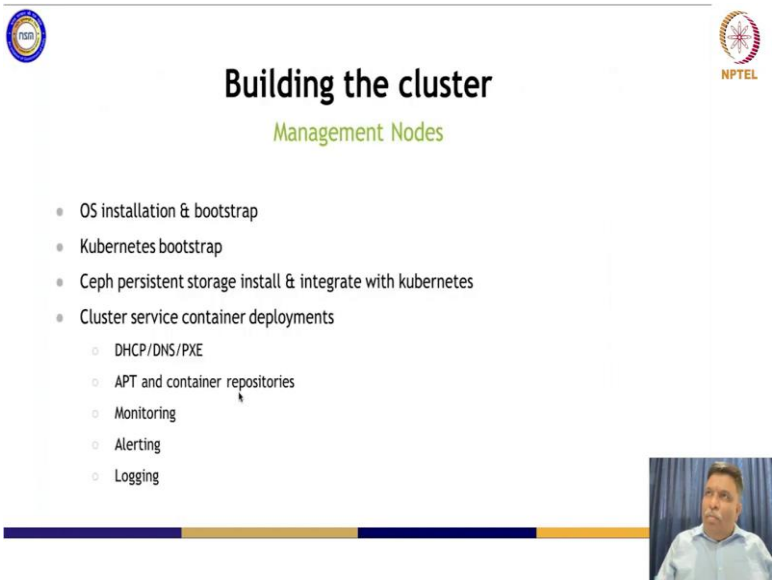
So, these HPC workload traditionally are non containerized workloads. And but nowadays you get lot of HPC containers right. So, whatever people used to do GROMACS this that everything still you have got a lot of containers which you can use for that matter ok. So, now, it is changing a bit. So, no issue on that part actually.

(Refer Slide Time: 05:53)



So, how do you actually stock up the stack up the software stack. You are trying to develop a hybrid Kubernetes and Slurm cluster. So, you have got prometheus here DHCP containers Grafana alert manager Kubernetes then Slurm login node this is your compute node or DGX or whatever then you have this management node. So, something like this right.

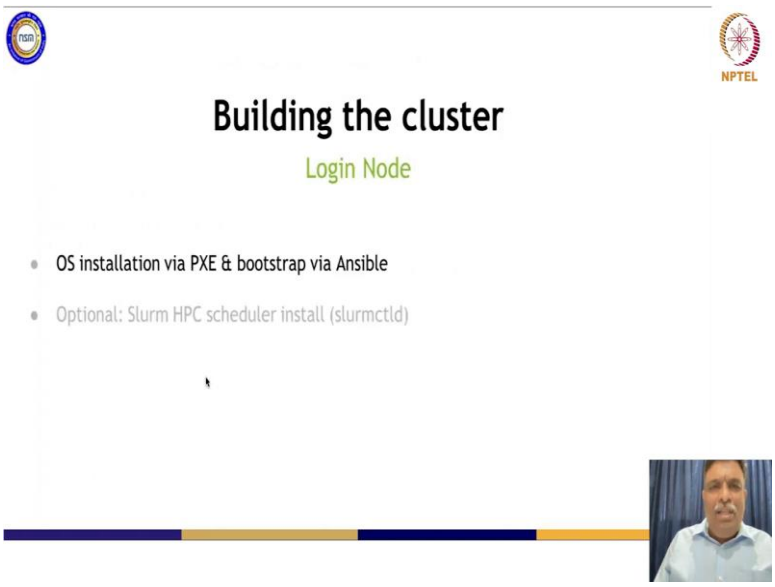
(Refer Slide Time: 06:25)



So, let us try to do some demo and before that just for people who want to understand right how do you actually develop your own cluster. So, I am just going in very brief as

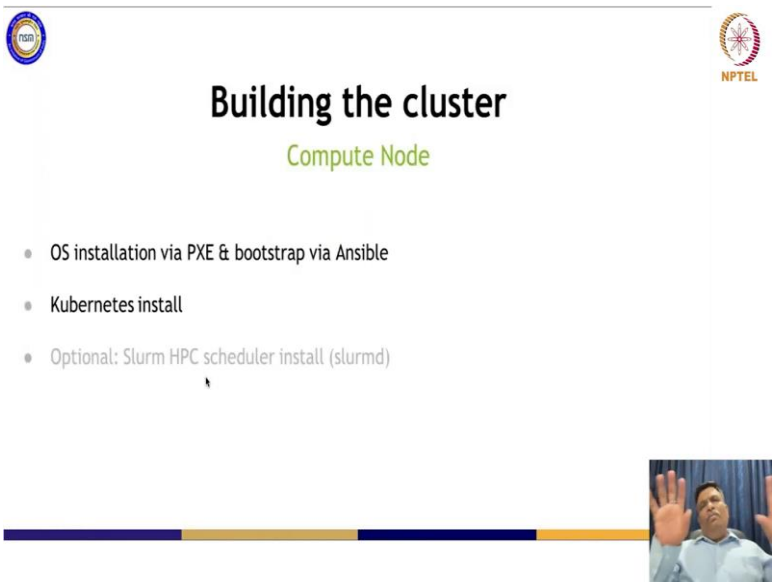
to how do you develop your own cluster you should have a management node which actually has the OS installation and Kubernetes bootstrap you have to install and integrate ceph persistent storage persistent storage is very very important and then there are container deployment.

(Refer Slide Time: 06:57)



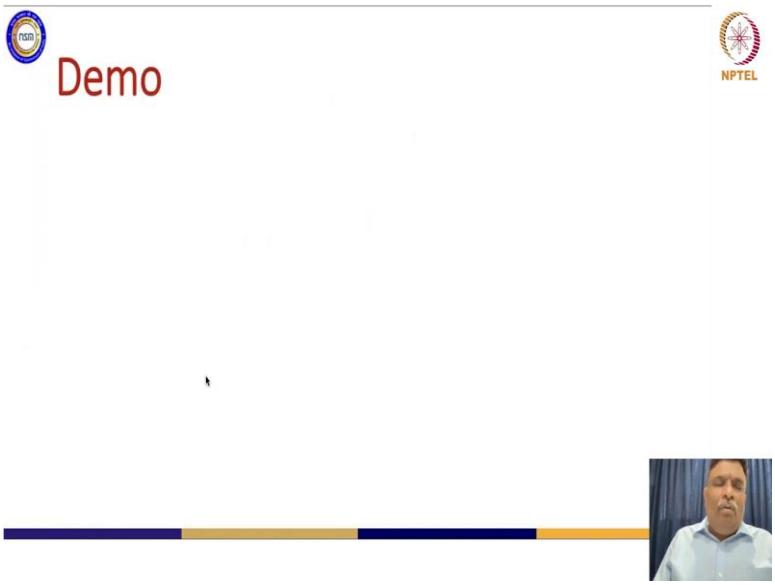
So, you have this management node you have the login node there are various ways of trying to actually do OS installation and bootstrapping of the login node.

(Refer Slide Time: 07:07)



And then the compute node wherein you need to install Kubernetes again and then bootstrapping it.

(Refer Slide Time: 07:13)

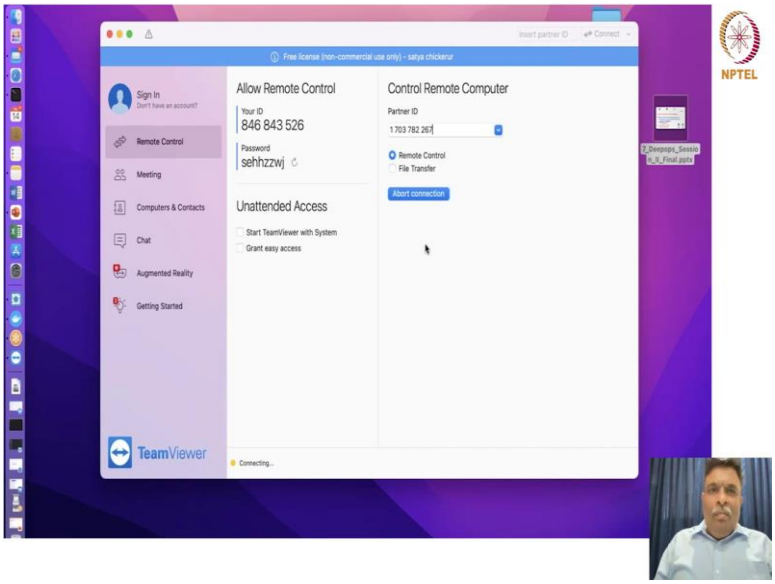


So, this is in brief actually what I thought we will do it in the theory portion. Now, let us go for the demo. So, what we are going to do in the demo now today is I will show you two ways of trying to see the same type of the setup right. We have actually one master node two worker nodes ok.

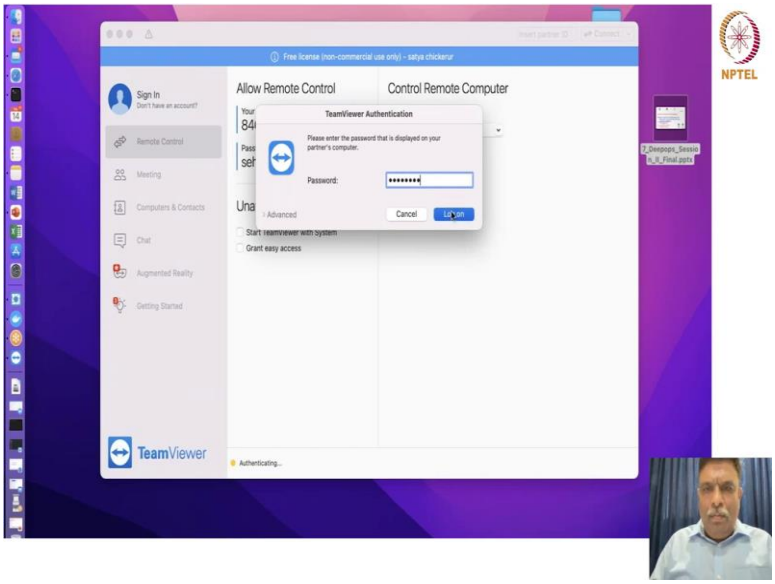
And then we will try to do it from the standard CLI type of thing using the Kubernetes dashboard try to analyze it try to run machine learning application. Wherein we are trying to do machine learning application development wherein by seeing the marks of the students the previous performance of the students we are going to predict as to what is his probability of getting placed right. This is a small ml application which we are going to develop.

And next we will see another type of a dashboard which is far more easier than the previous one. So, that you can appreciate as to how this particular thing of Kubernetes the GUI development the real time analysis all is actually improving ok for us to be better deep learning scientists or programmers.

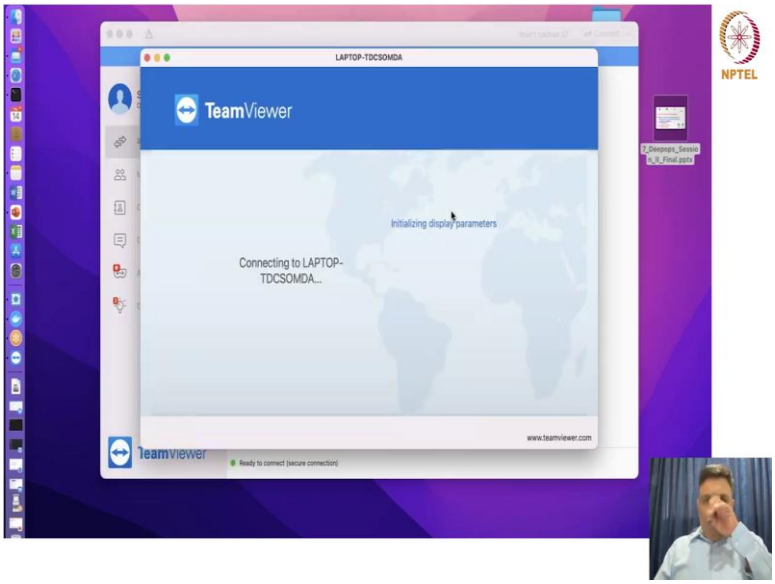
(Refer Slide Time: 08:56)



(Refer Slide Time: 09:12)

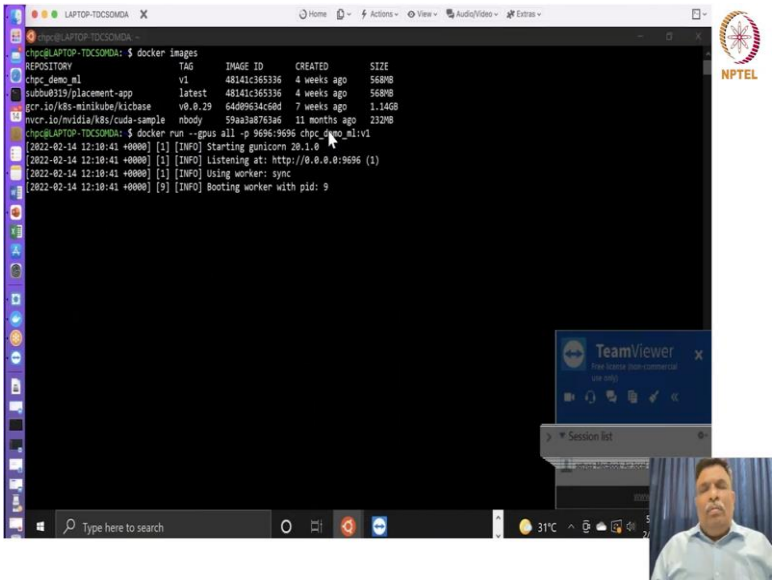


(Refer Slide Time: 09:23)



So, let us try to go to the demo. Let me just log on to it ok.

(Refer Slide Time: 09:24)



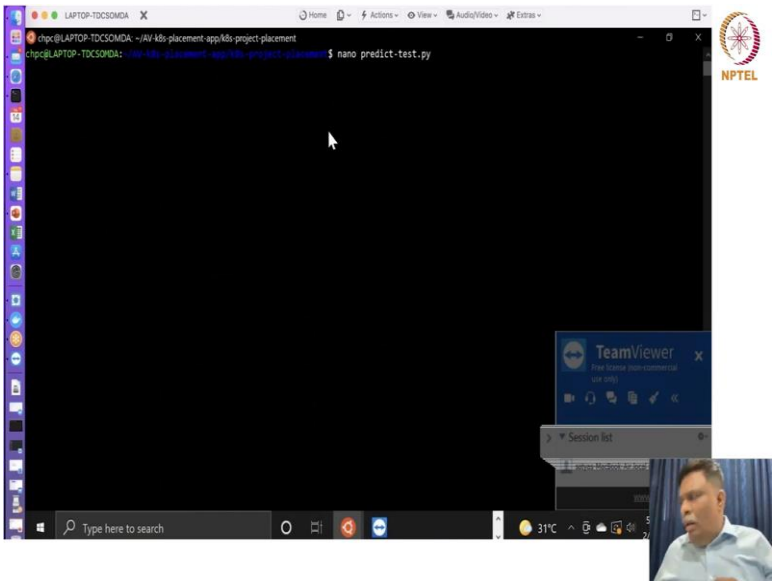
So, now, let us try to see whether what dockers we have ok. What dockers we have? So, yeah so, we have got so many docker images right. So, what we are trying to do is we are trying to use a demo docker which is CHPC demo for machine learning.

It has a tag v1 it has got it is this image id right and it was created 4 weeks ago and this is the size of that particular CHPC demo ml. So, how we have done it? We have accessed

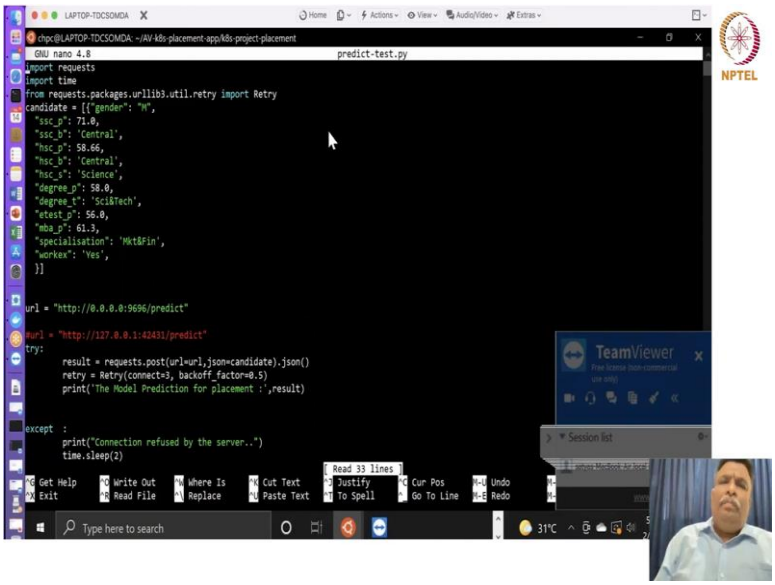
the docker from this we will share the link we did some modification onto it and then tried to rename it and repackage it as a CHPC demo ml docker ok yeah.

So, the next thing is to see and run this docker. So, how do we run? We use this command docker run GPUs all these are the port numbers, local host port numbers then this is forwarding and then this is this particular program ok which we need to run ok.

(Refer Slide Time: 11:08)



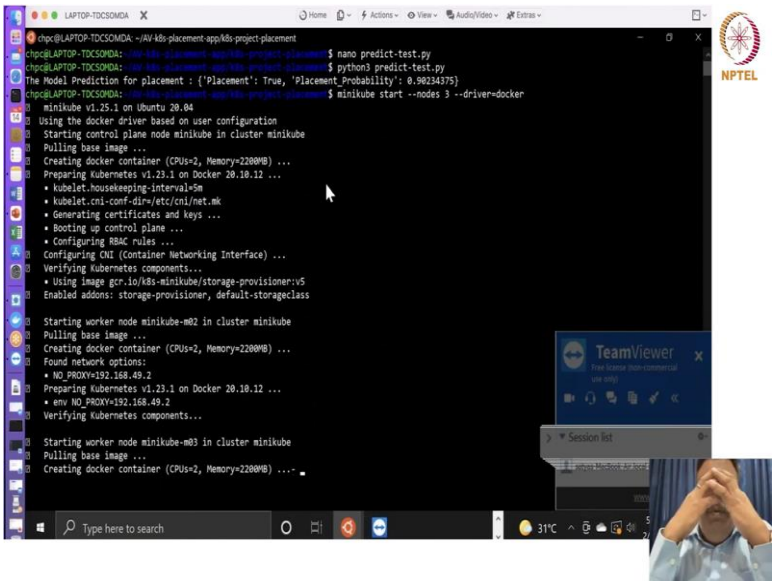
(Refer Slide Time: 11:27)



So, now what we need to do is try to understand and try to see excuse me. Try to see the program to do some prediction which basically is something like this. So, the candidate

gender, his marks ok and what has to be done very simple rudimentary type of stuff just to show you people as to what are we going to do right.

(Refer Slide Time: 12:00)

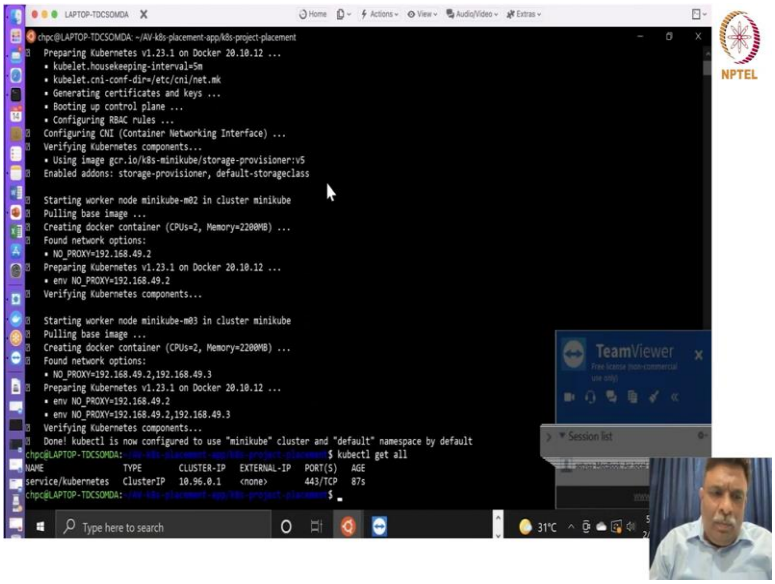


So, now let us try to run this program. So, once we run this program the probability ok for placement is 0.90234375. Now let us try to see this in the dashboards right. So, the idea is that we are going to start ok the minikube nodes as the three things. So, what we have tried to do here is we have tried to just see ok that the dockers which we have downloaded runs using a python.

Now, we are trying to link it up to these three nodes which we are running ok. So, I hope it is clear that minikube is a local Kubernetes which gives me the option of generating a master and two slaves or workers before that I have my python program which can do this prediction right.

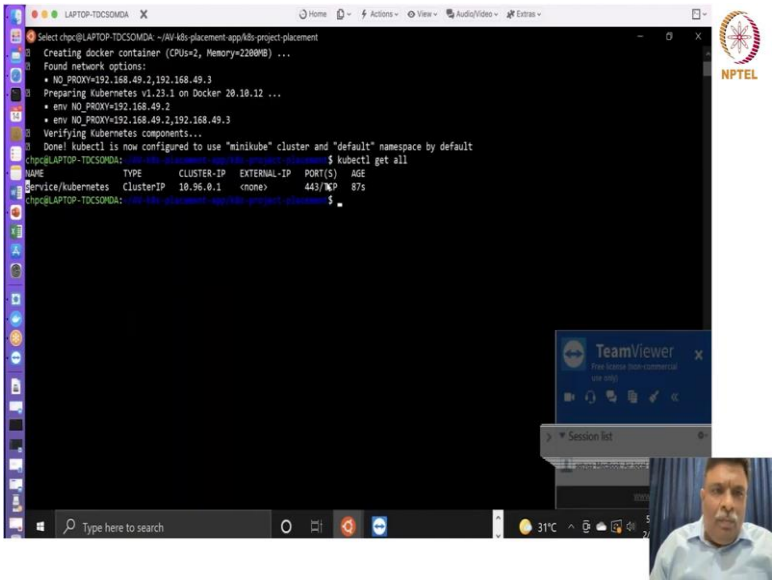
So, now I am going to take this python program take it as a docker ok and then try to run it. So, that it is serviced for people or people can access it when it is running ok on one of these worker nodes right.

(Refer Slide Time: 14:33)



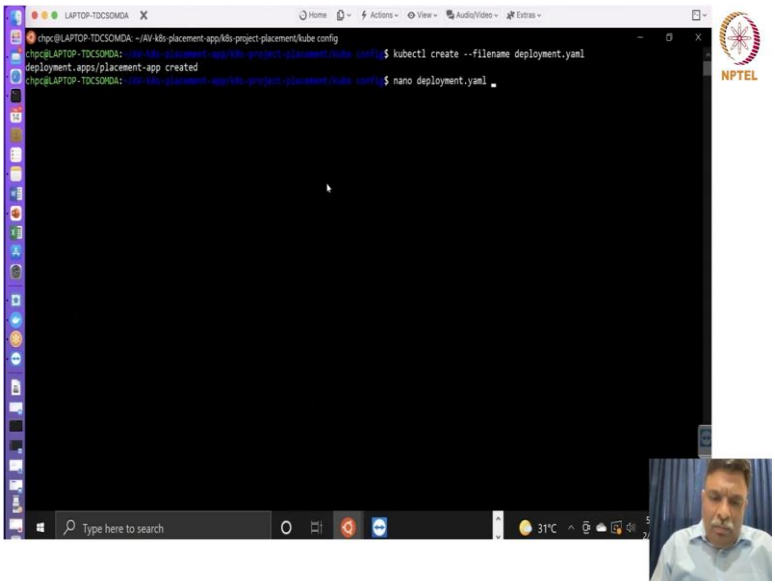
So, the idea is that when you work on a desktop, you can just run your program see it and then you tell that you will test it. So, you give the input it gives you the prediction. So, it all ok done. But when you are giving it as a service to people in a sense that when you are trying to develop application trying to sell it or then trying to develop it for the use of people then basically you need to actually give it as a service. So, we will have to deploy it somewhere right. Now how are we going to deploy it we are going to deploy it using kubectl see here.

(Refer Slide Time: 15:20)



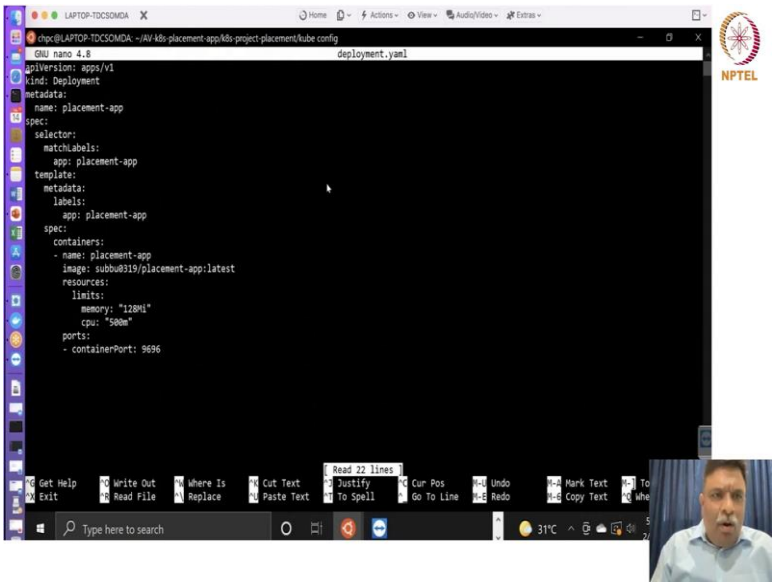
So, this is actually the service being provided by the Kubernetes on that particular IP cluster IP right.

(Refer Slide Time: 15:38)



So, now we will run this and then try to deploy it right. So, we have created we will try to see the deployment file now.

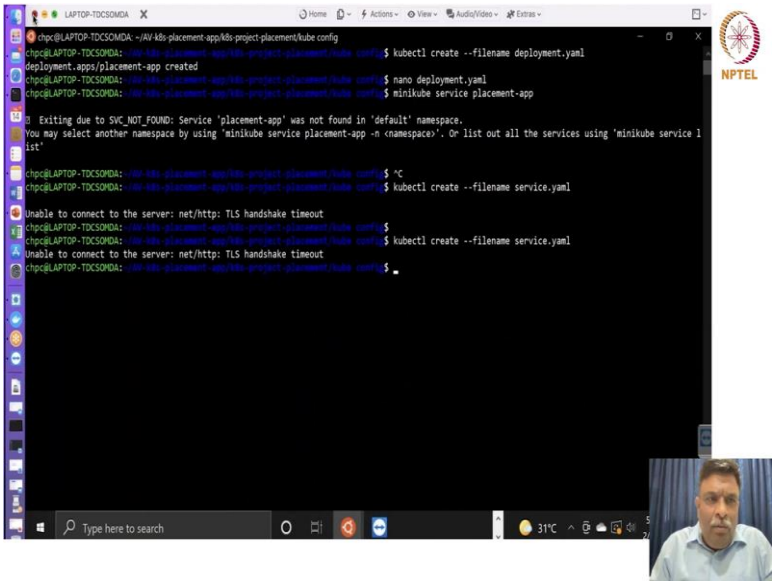
(Refer Slide Time: 15:59)



So, what does this deployment file mean? We are trying to deploy this is a kind of deployment metadata tells that it is some name by name placement app application is a placement app the containers name is placement app, but this is the container which we

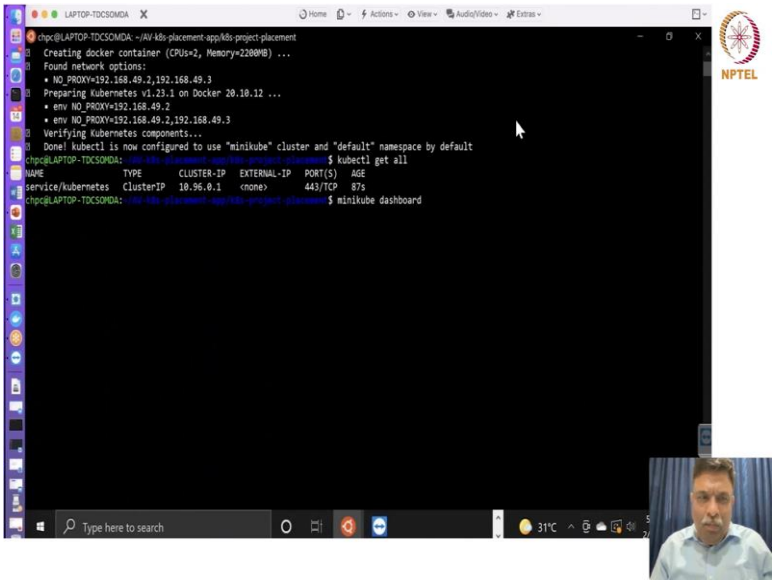
have tried to take the image from. I have this limit of my memory decided the CPU this much ok and then the container pod on which this particular pod actually is supposed to be deployed right. So, this is what we have tried to do it.

(Refer Slide Time: 16:43)



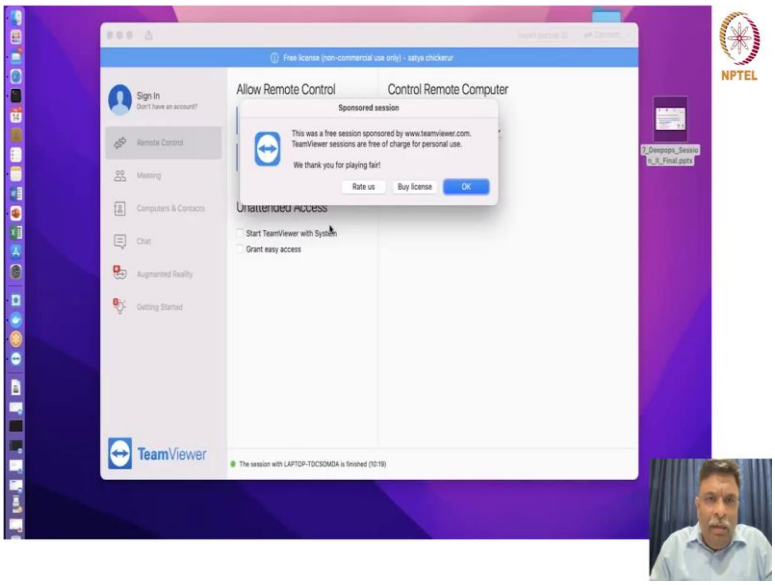
Now, once we deploy it we will actually make it available as a service correct. So, then we run this minikube service placement app right. So, now, it is running ok. Let us wait for some time. 1 minute, we got some error it seems we have not been able to, yes. I do not know why this has become so slow. We will see the Kubernetes dashboard and see whether the previous one is running.

(Refer Slide Time: 19:01)



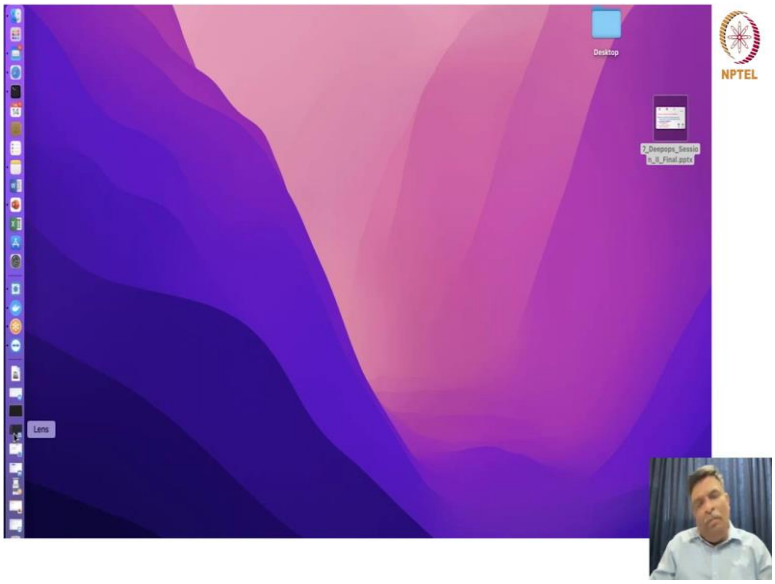
So, there is some issue with the TLS handshake. So, let me just go to the next thing in the meantime we will try to debug it.

(Refer Slide Time: 19:42)



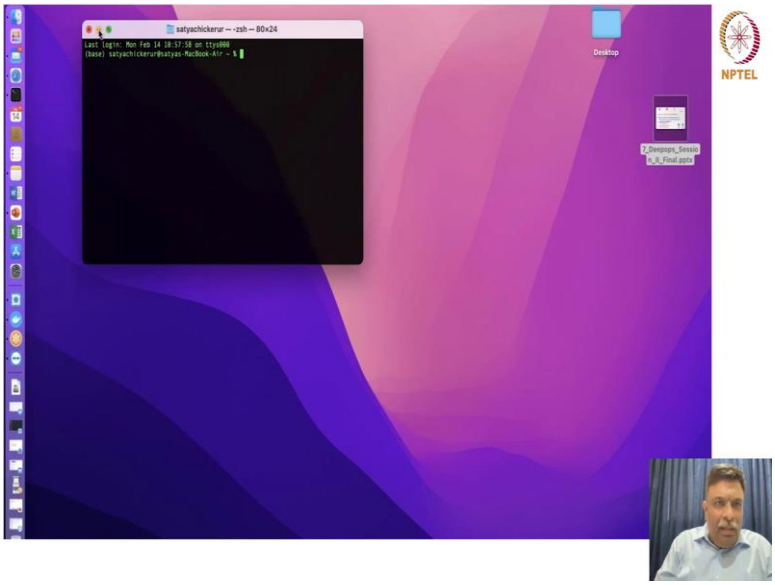
But let us try to actually go and see a situation here.

(Refer Slide Time: 19:51)



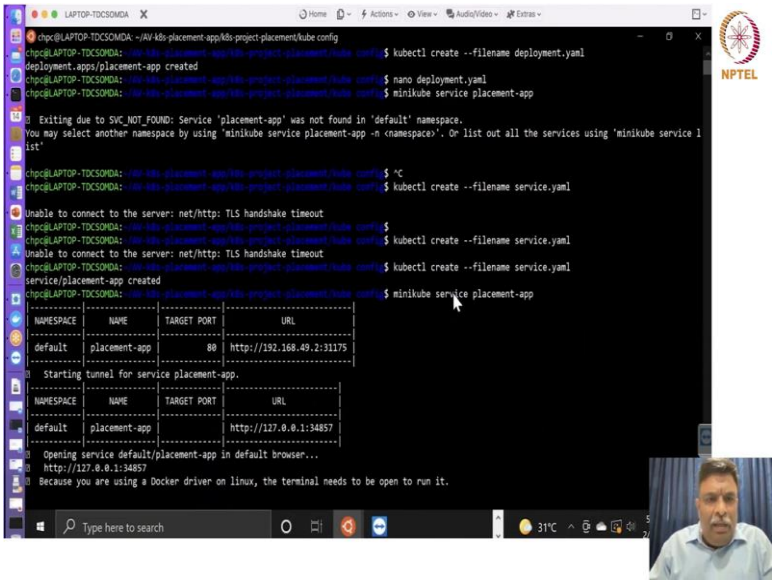
See once you do it using this right there will be certain issues like this popping up what is now actually happening is this kubectl has got some issues right. So, these type of issues to avoid yes.

(Refer Slide Time: 20:11)



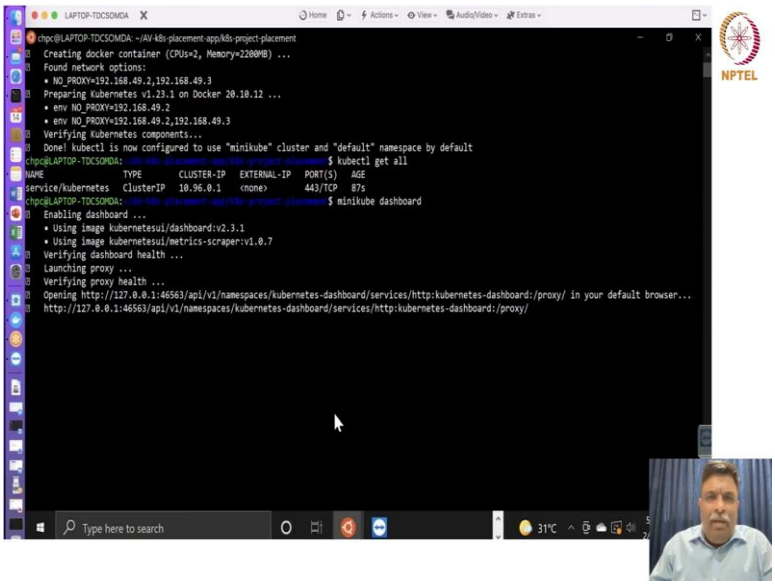
So, we will try to actually see that it is done up and running it takes time. So, sometimes what happens is you are and again connect it where is this r x 3 4 0 6.

(Refer Slide Time: 20:45)



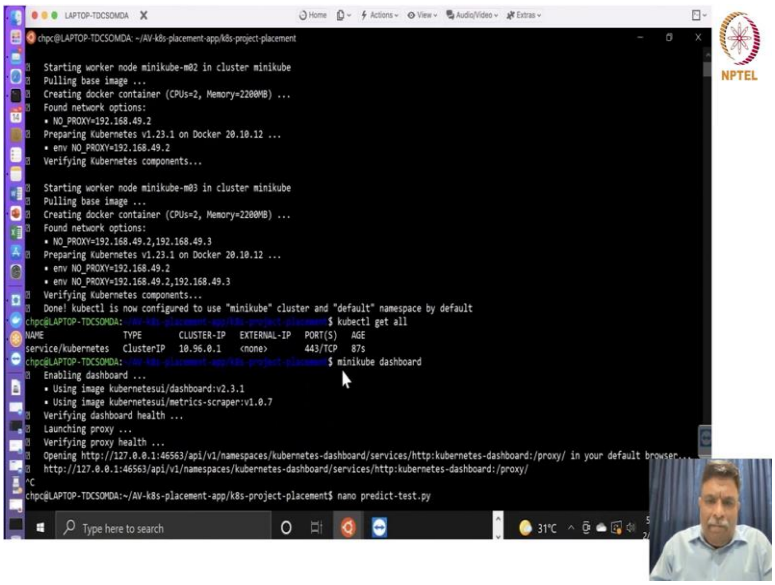
Yeah so, the service placement app got created we were running the same thing, but it was giving us the TLS handshake timeout. So, sometimes it happens, but anyway once we are able to do it. Now yes so, if you see this we are able to run it in the default name placement space right the default name space this placement app with this particular target port using this URL and then we have to actually do a tunnelling for this service placement app to this URL right.

(Refer Slide Time: 21:33)



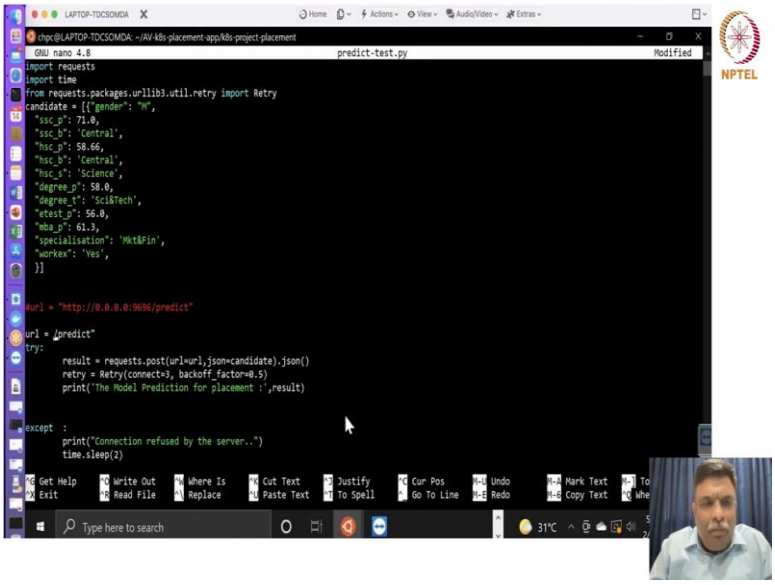
So, we will now try to open this dashboard ok and see as to what we get and then we will discuss.

(Refer Slide Time: 21:44)



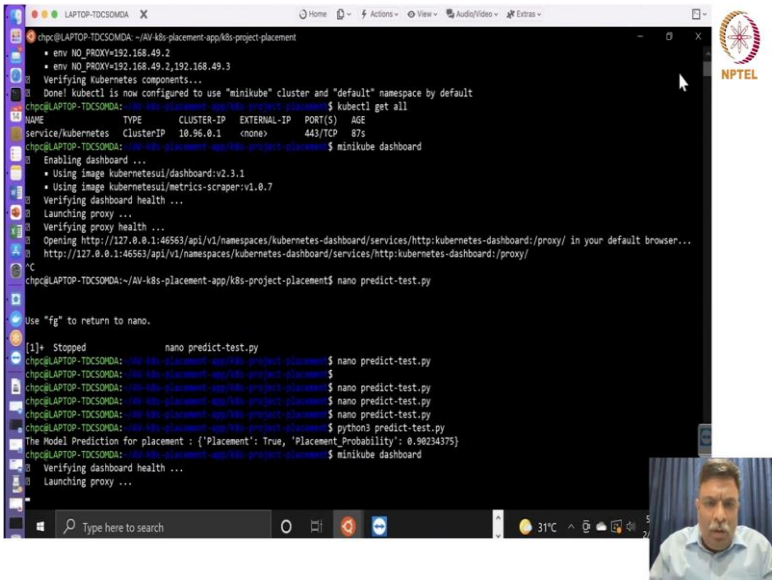
So, what effectively happens is we are trying to do two things; one is the old URL one is the new URL and we will have to do the tunnelling. So, we will now try to run the prediction ok using that URL because it will be serviced there now ok. So, we will we have changed this and then we will now try to run it yeah.

(Refer Slide Time: 22:17)

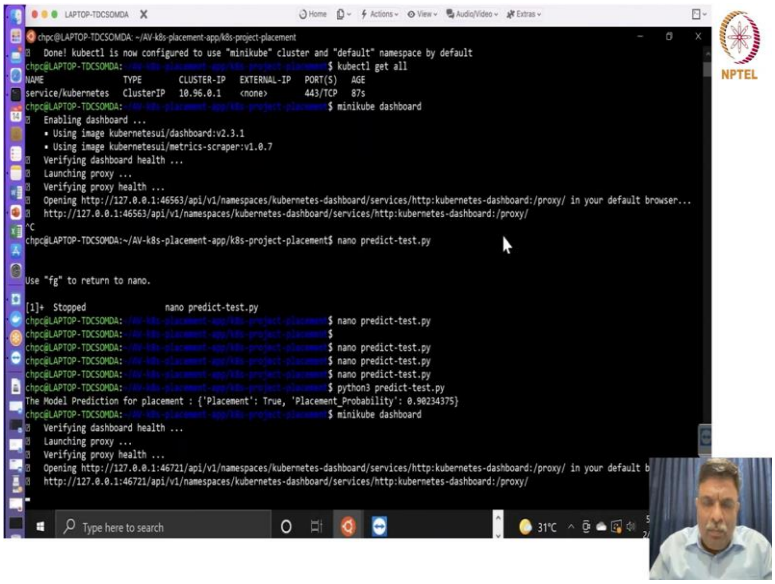


We will have to put that IP.

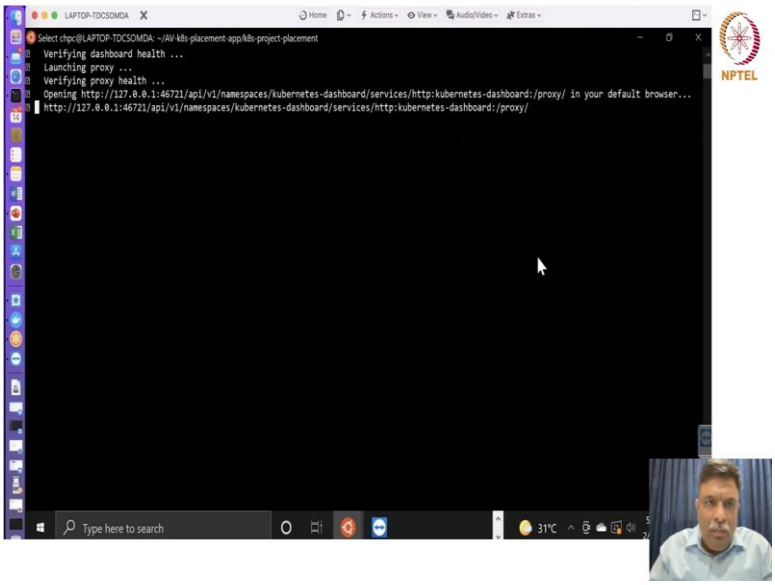
(Refer Slide Time: 23:06)



(Refer Slide Time: 23:34)

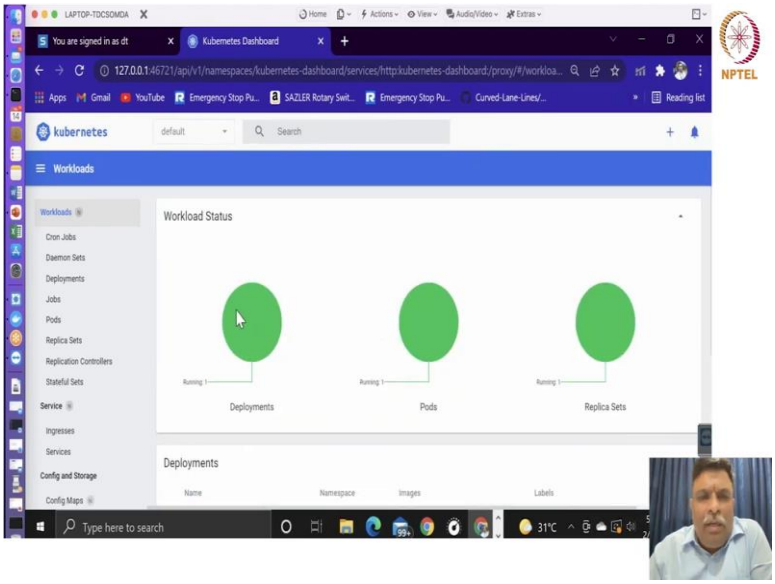


(Refer Slide Time: 23:35)

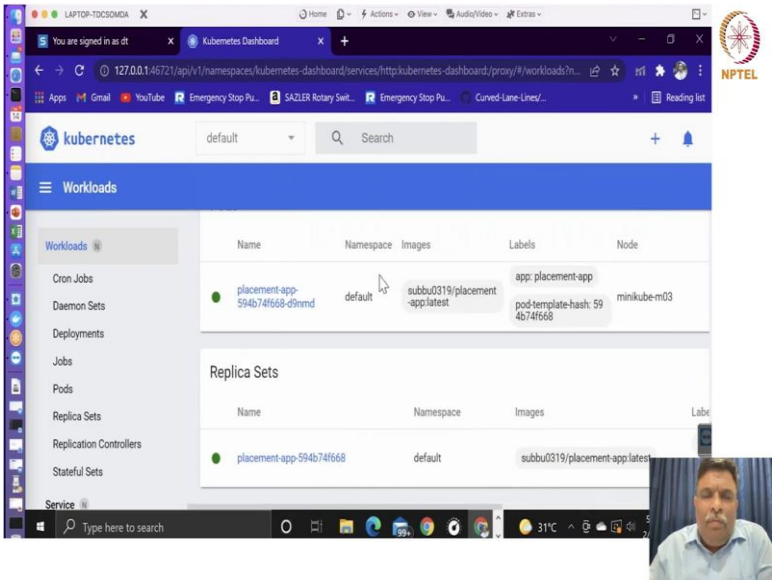


Yeah So, now, we will go to the dashboard and see as to the same result right just running on dockers and then running on Kubernetes right as a pod now how will we get it right. So, we will try to see this yeah.

(Refer Slide Time: 23:58)

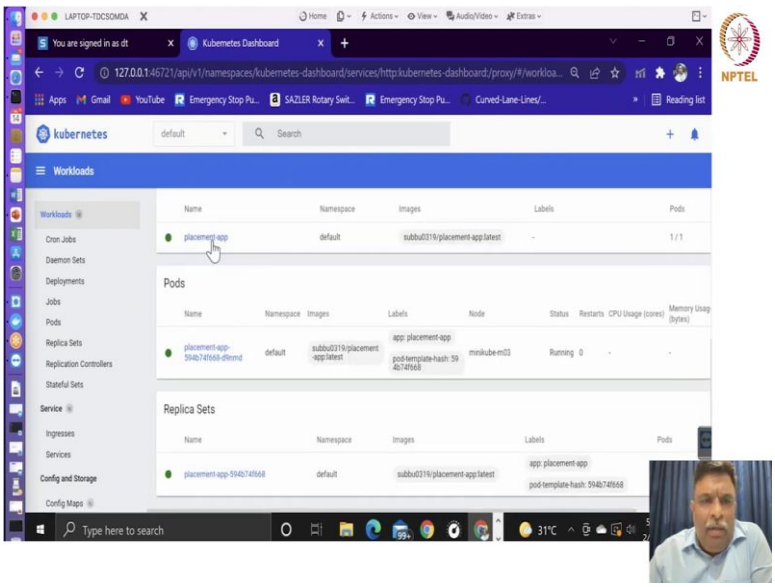


(Refer Slide Time: 23:59)

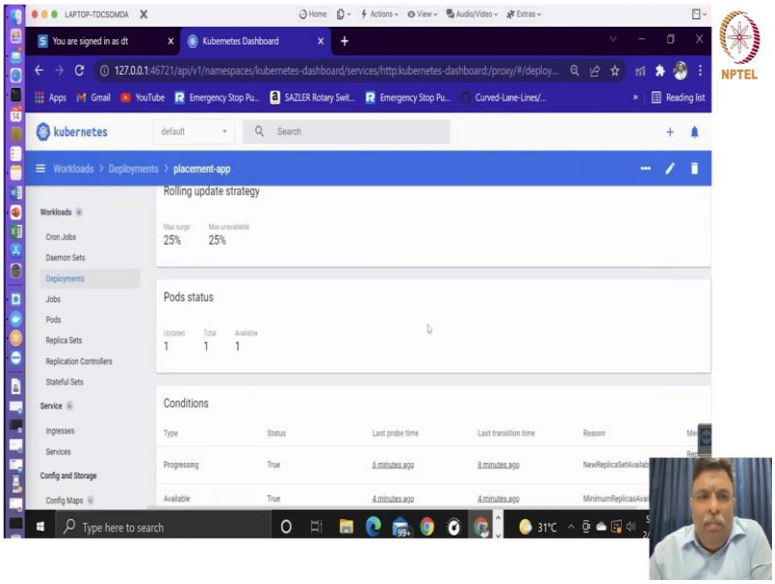


So, see here if you see what happened if you see the workload status there is one deployment, there is one pod, there is one replica set and then if you see what are those right, the deployments this is the placement app.

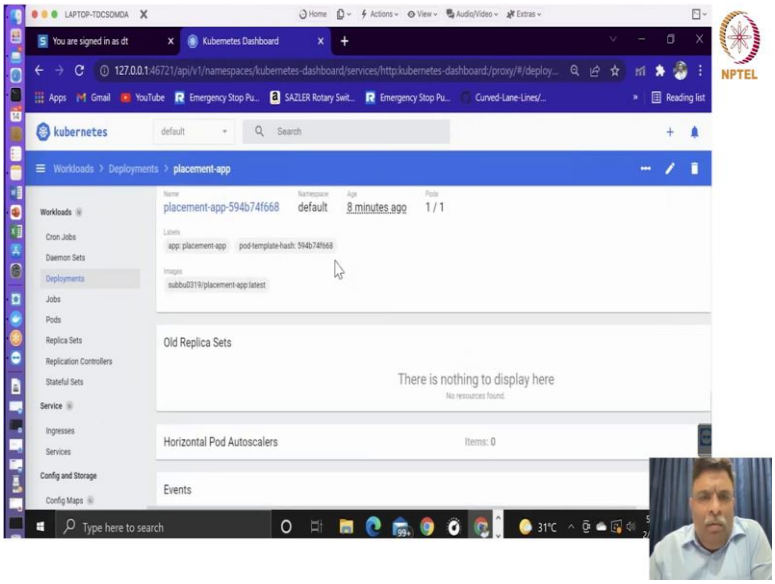
(Refer Slide Time: 24:19)



(Refer Slide Time: 24:22)

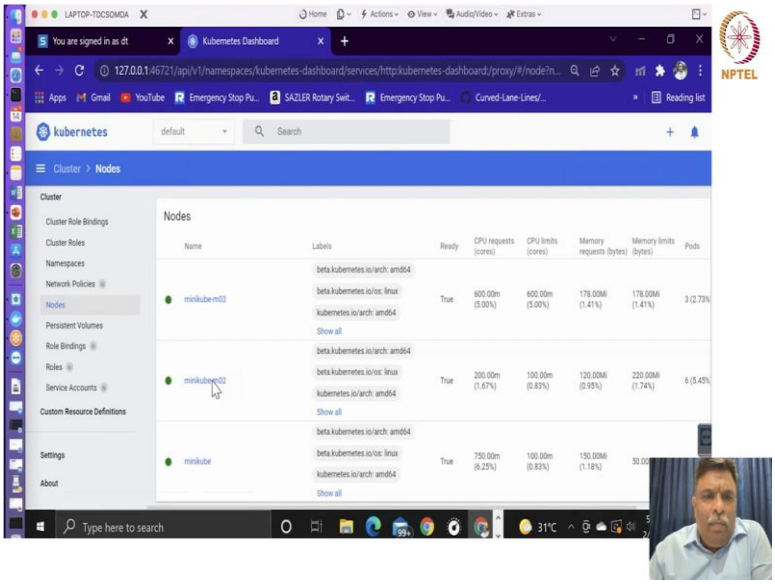


(Refer Slide Time: 24:24)



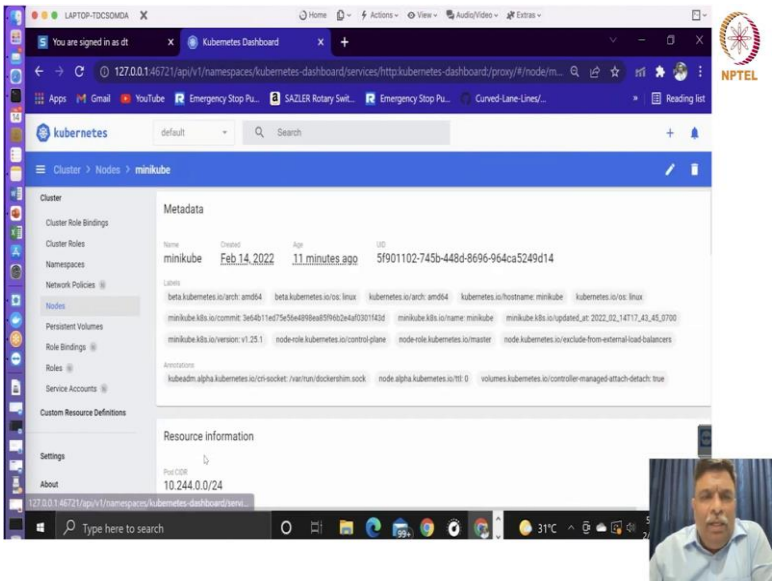
What is the status, when did it start right? So, what type of label it uses? Which is the image you are trying to run on it right? All these details you get.

(Refer Slide Time: 24:41)



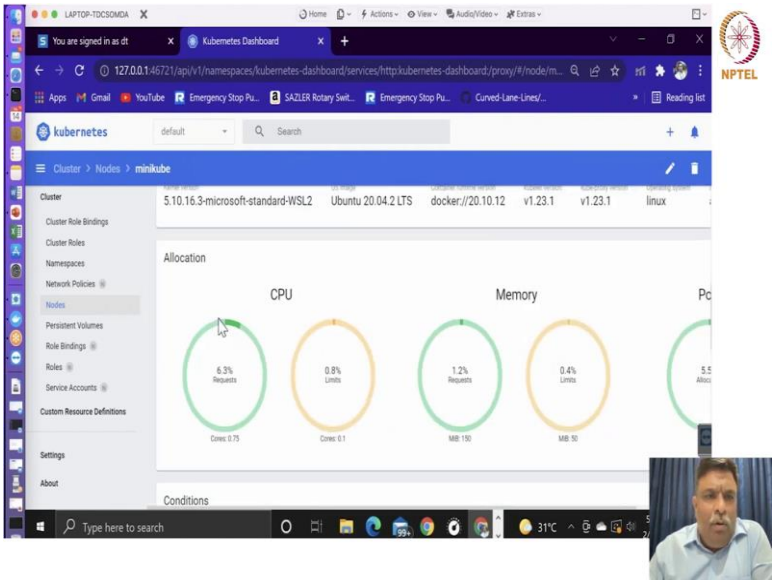
Similarly, if you see the nodes right. So, yesterday we saw that there were three nodes right; the one is the master there are two worker nodes.

(Refer Slide Time: 24:52)



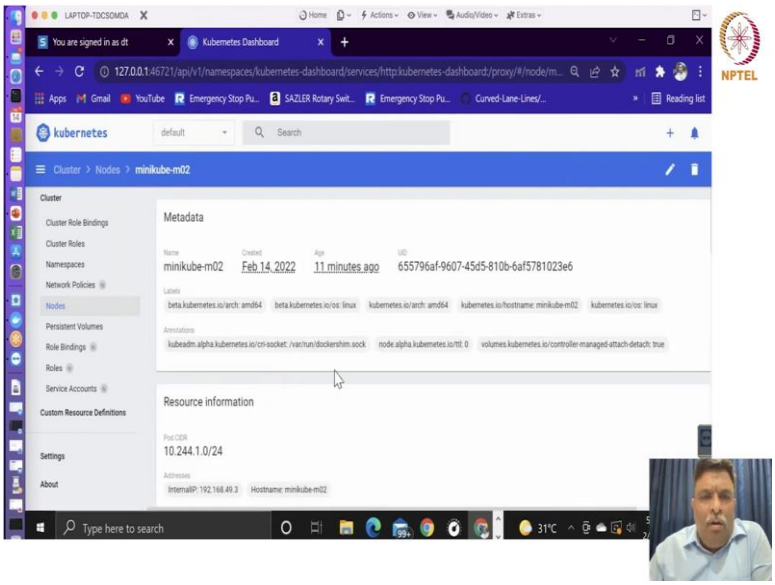
So, if you see now this minikube which is the master node.

(Refer Slide Time: 24:56)



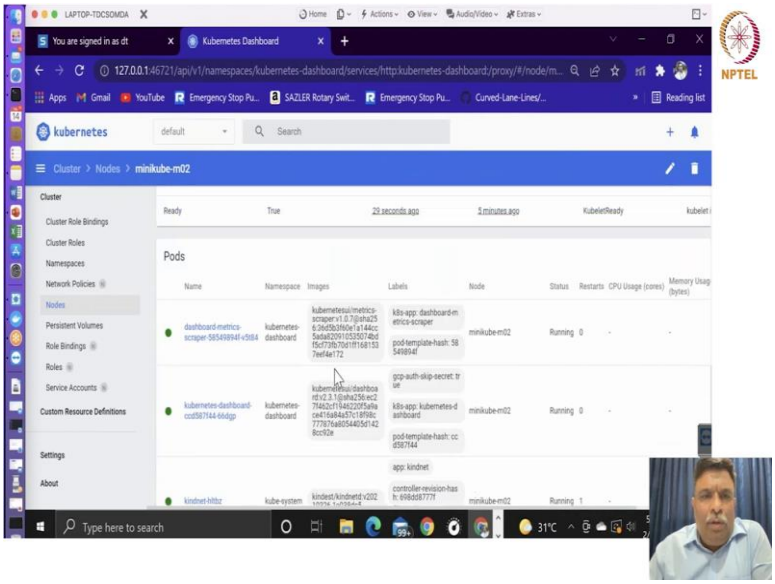
You have got details about this right the CPU the memory and how many pods it is running. Now if you go to this worker node ok.

(Refer Slide Time: 25:12)

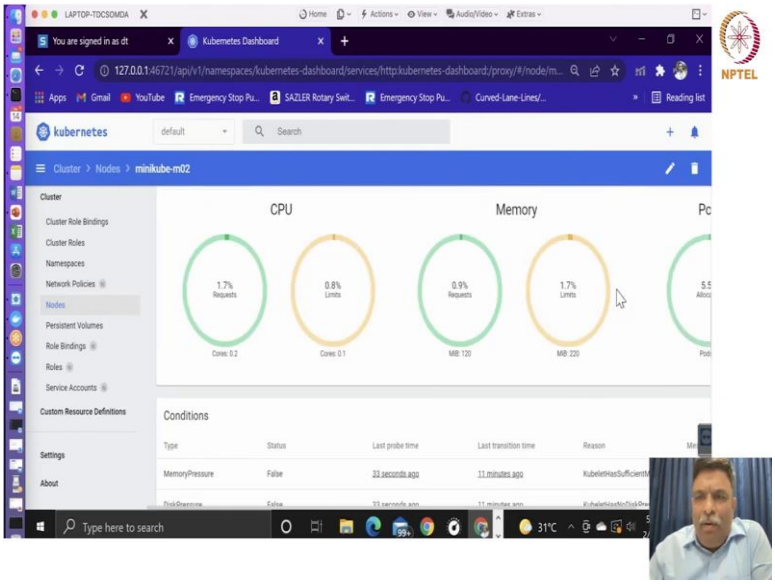


Minikube m 02 again it has got two CPUs right so many pods running ok.

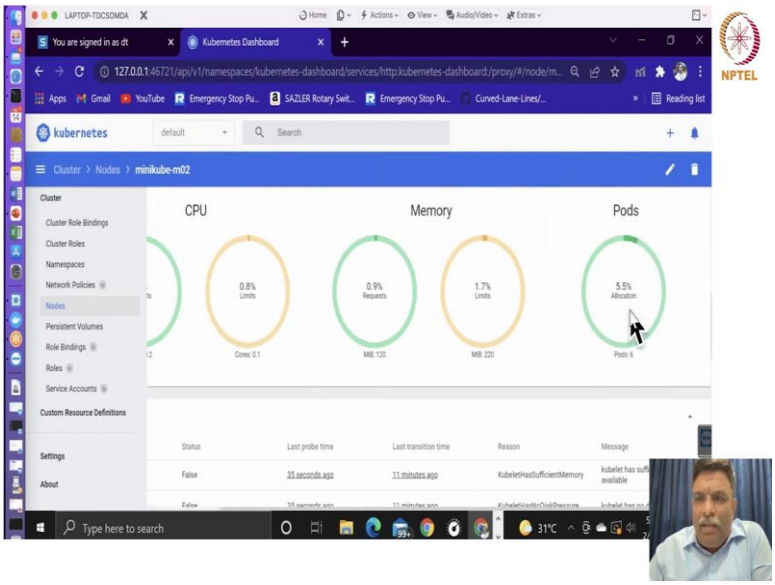
(Refer Slide Time: 25:16)



(Refer Slide Time: 25:18)

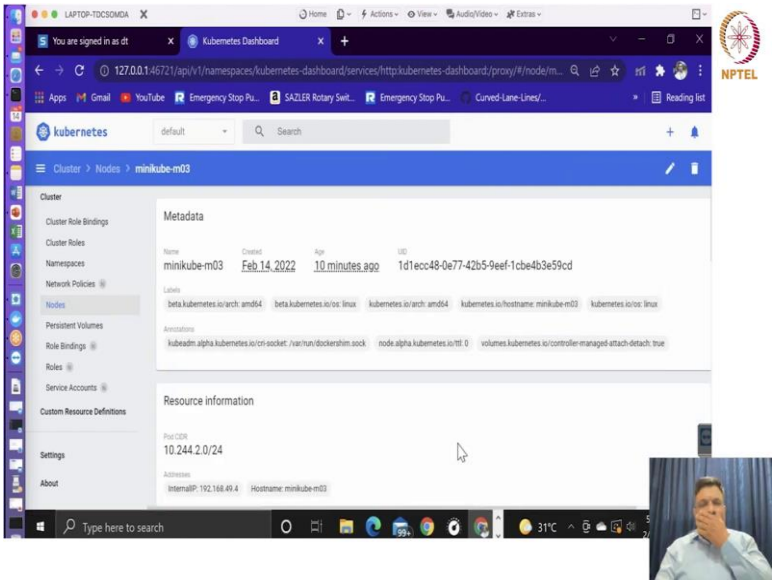


(Refer Slide Time: 25:21)



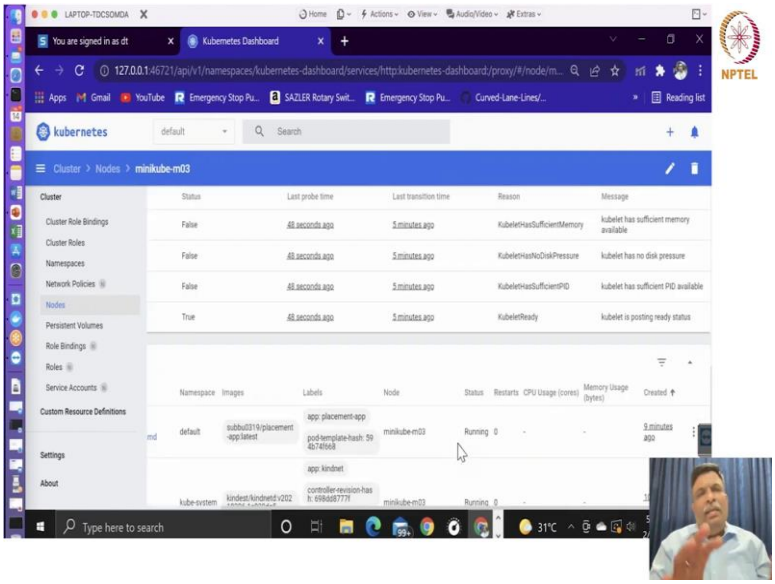
So, much memory allocated and stuffs like that.

(Refer Slide Time: 25:30)



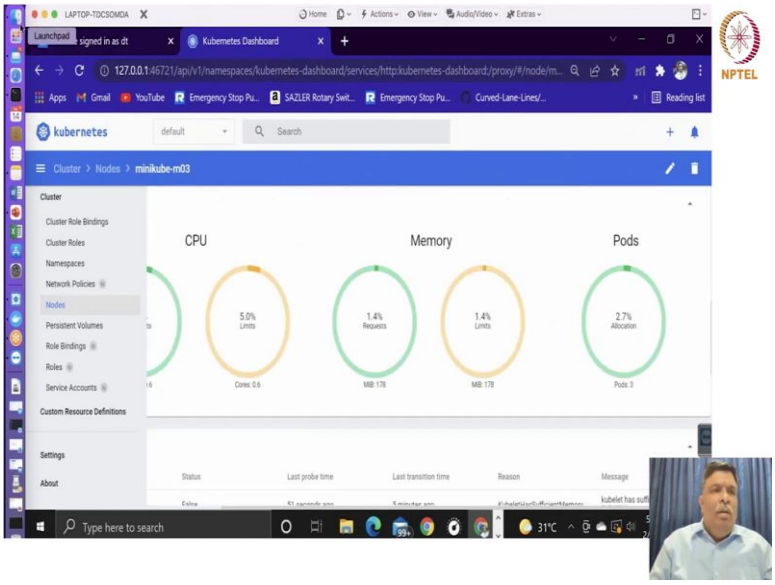
And then if you see this minikube m3 it again has this ok.

(Refer Slide Time: 25:36)



So, this is the Kubernetes dashboard.

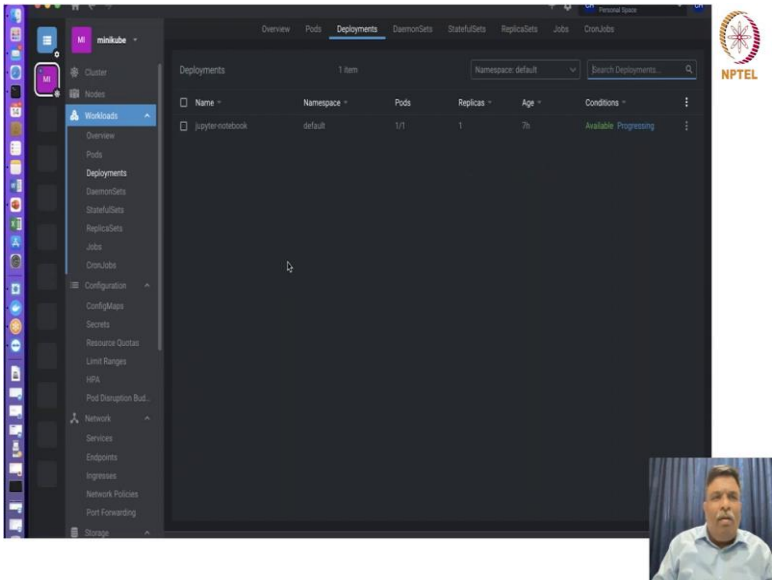
(Refer Slide Time: 25:38)



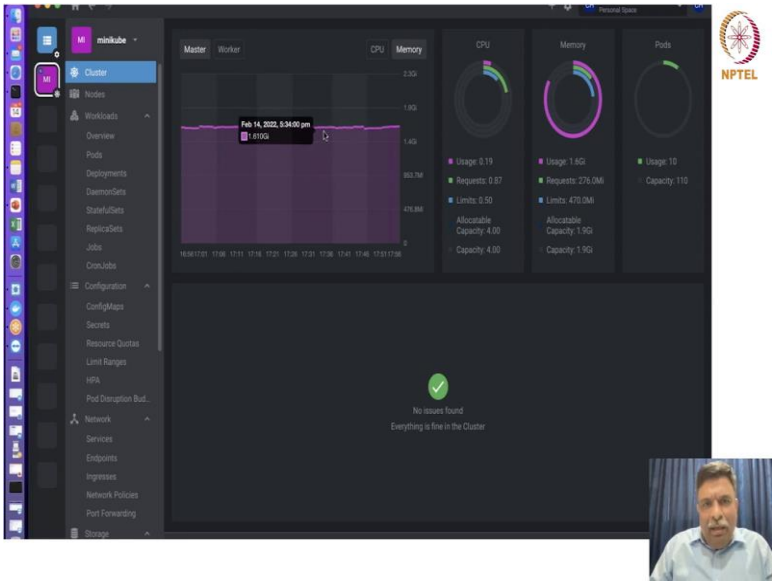
Now, before we end today's lecture maybe I will take 5 to 10 minutes more I will show you another very good GUI right. So, even if you are not understood it totally at least you should be in a position to understand that there is a node on that node you are running the pod that pod runs a docker it can be any application. So, for the time being it is a machine learning application which we saw.

Now, when you see the dashboard it is not that user friendly in a sense that you will have to scroll you will have to see everything. Now I am going to show you another way of seeing the same thing right which is one of the very very newest and the latest things which is called as lens.

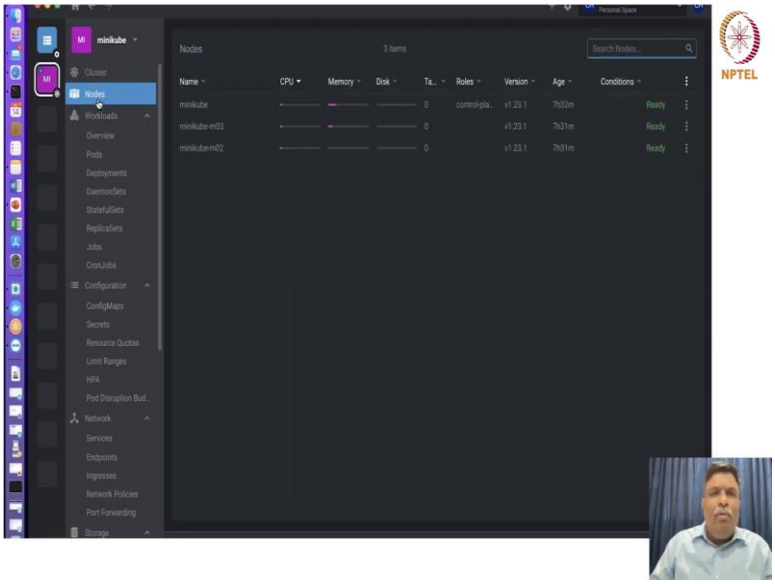
(Refer Slide Time: 26:25)



(Refer Slide Time: 26:33)

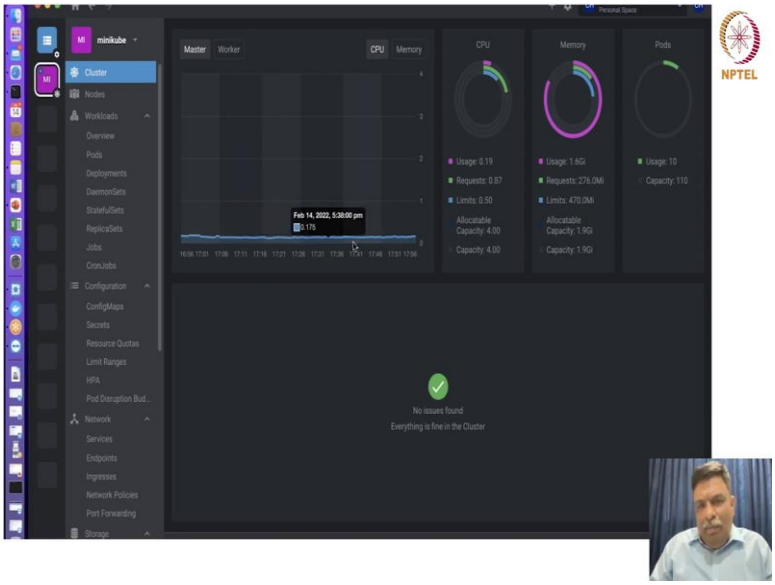


(Refer Slide Time: 26:33)



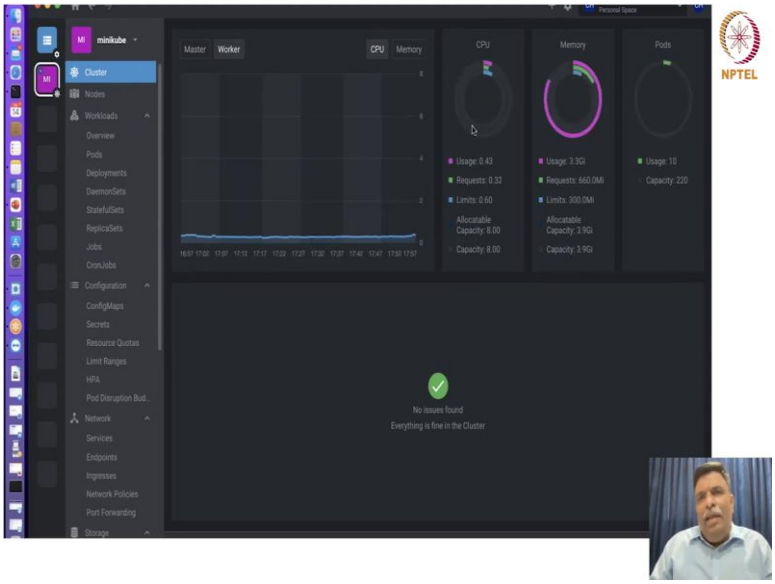
Now this helps me to actually see my cluster information in a very very user friendly way. For example, it shows me the master the master what is the CPU utilization, how much memory is being utilized, how many pods are running right and how much of the memory in real time actually how is it being utilized.

(Refer Slide Time: 27:01)



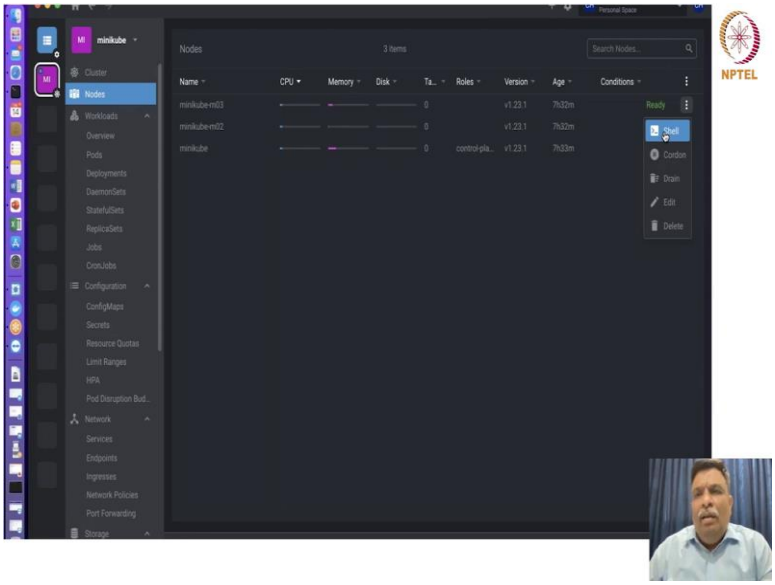
CPU performance or how is CPU actually being used right. So, usage wise on the master.

(Refer Slide Time: 27:10)



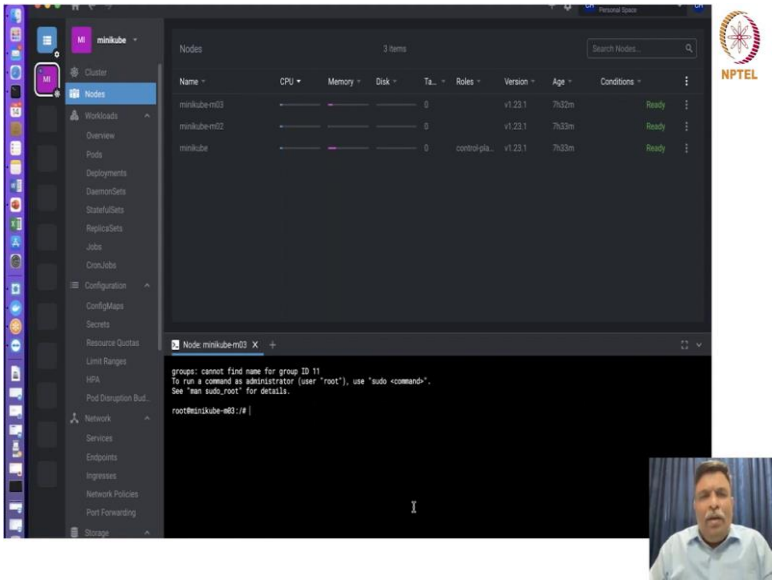
If you go to worker again you get the information about how what is the utilization of the CPU, how much memory is being utilized, how many pods are running right. So, if you see this is a better way of trying to understand it in a better way right.

(Refer Slide Time: 27:31)



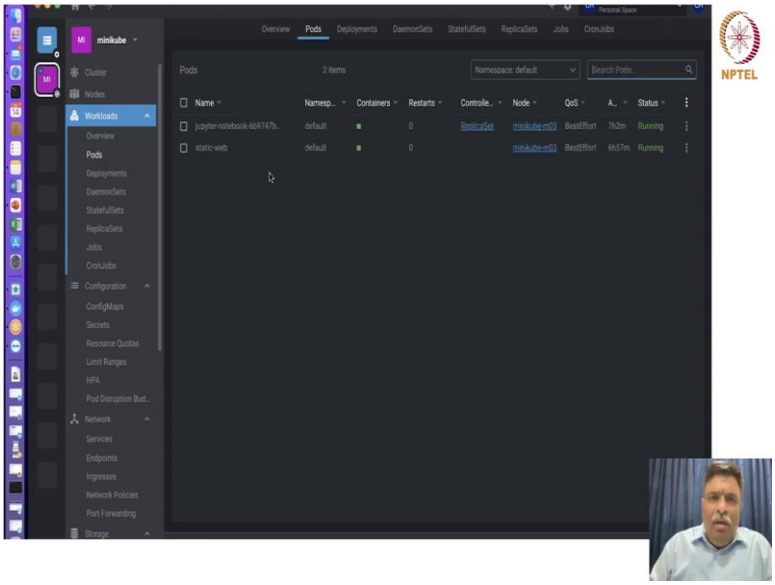
Here also you can see three nodes; one is the master node, one is the worker node, another one is the other worker node. And if you see this you can work at the shell prompt of that particular node right.

(Refer Slide Time: 27:49)



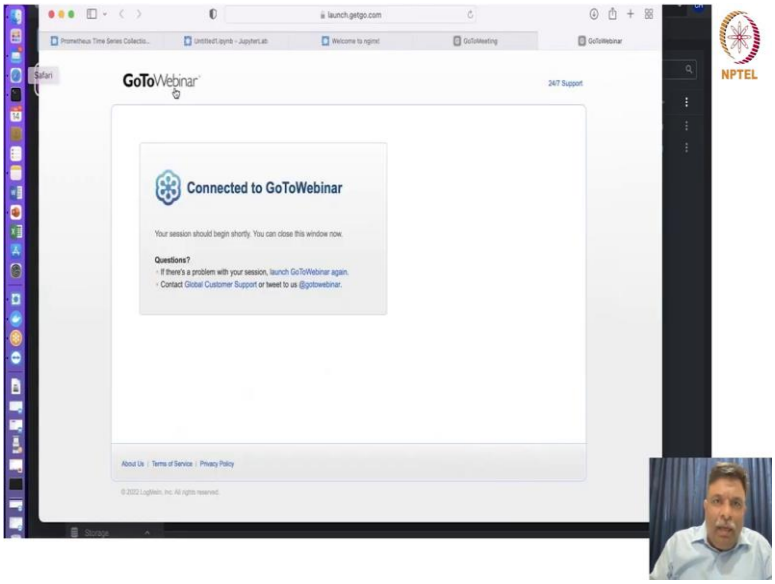
So, let us try to understand this how it will be useful ok in your days to come when you are trying to use it right. So, you have actually entered the node minikube 3 now right. I am not using it at present, but I am just trying to tell you the facilities available in this. Now you see the jobs ok.

(Refer Slide Time: 28:18)



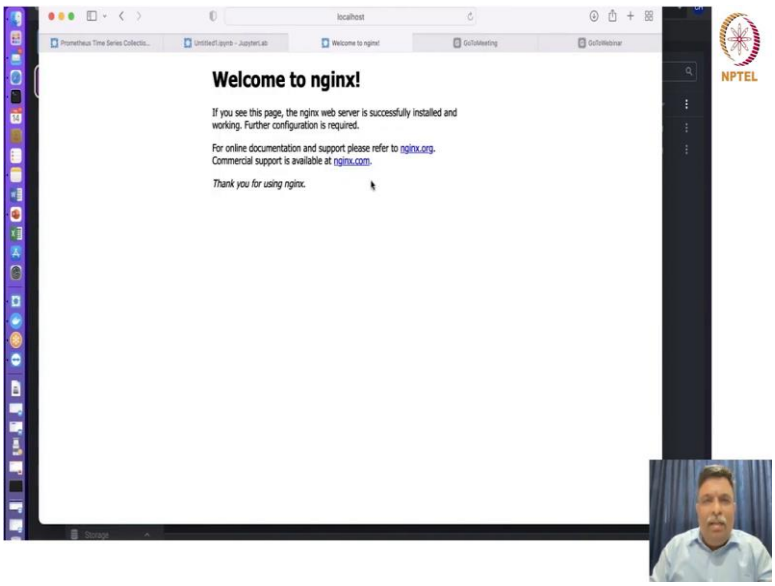
Let us see the pods we have got two pods running I will tell you what are those pods actually ok. So, one is a static web which is this nginx thing which we saw in the previous this thing. So, if you see the nginx thing.

(Refer Slide Time: 28:40)



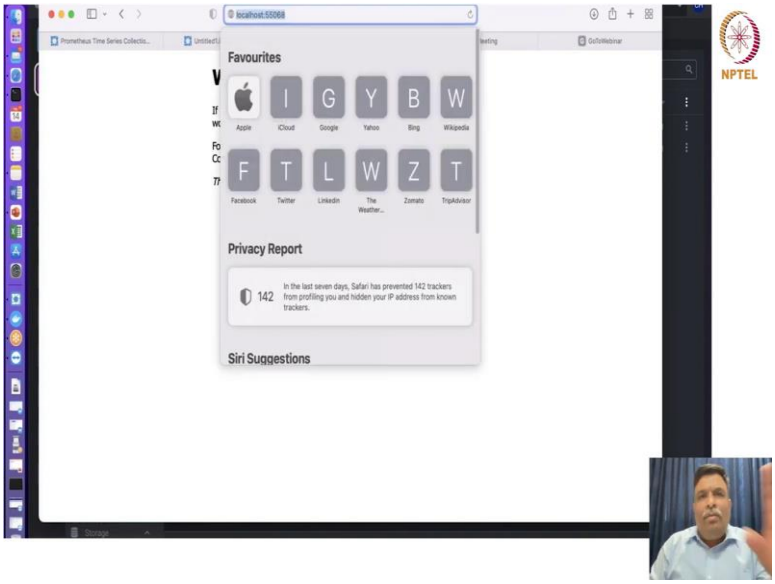
Let me just show you that yeah.

(Refer Slide Time: 28:41)



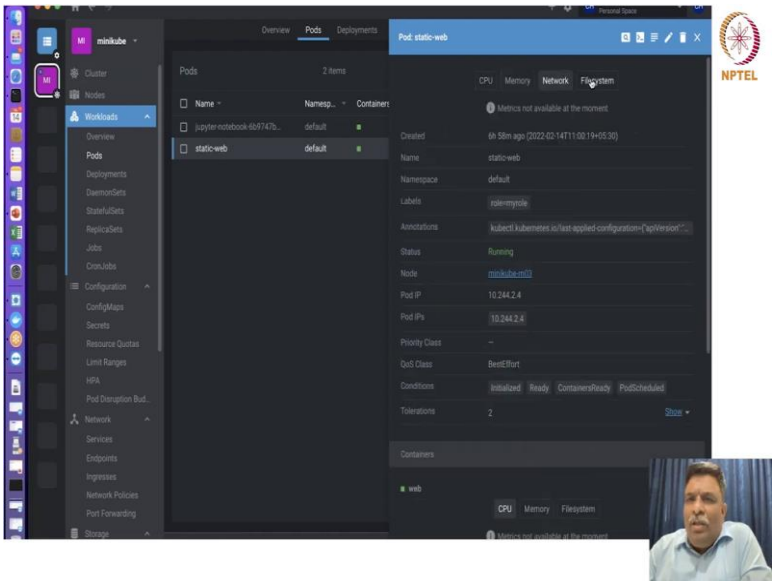
So, see this is running on that particular pod right. So, I will tell you what; that means, ok.

(Refer Slide Time: 28:50)



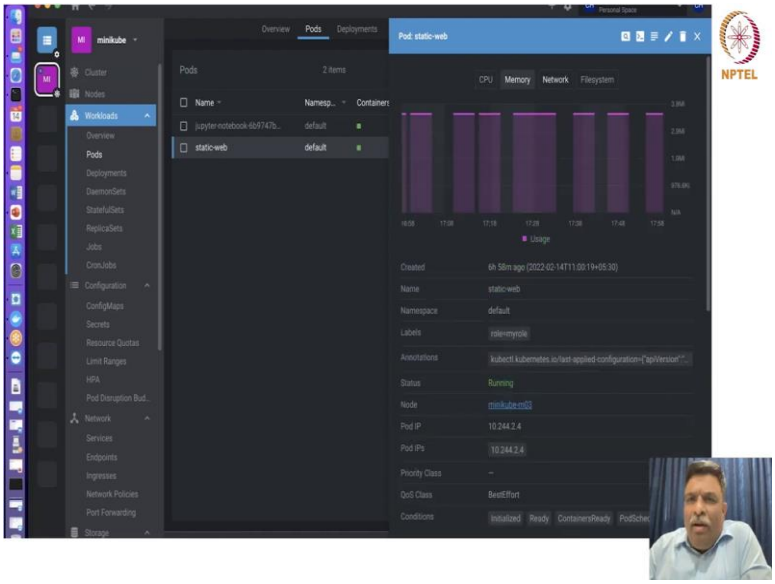
So, this is actually getting serviced on the local host it is a web server installed ok and it is running in this pod this is this pod ok this particular pod.

(Refer Slide Time: 29:00)

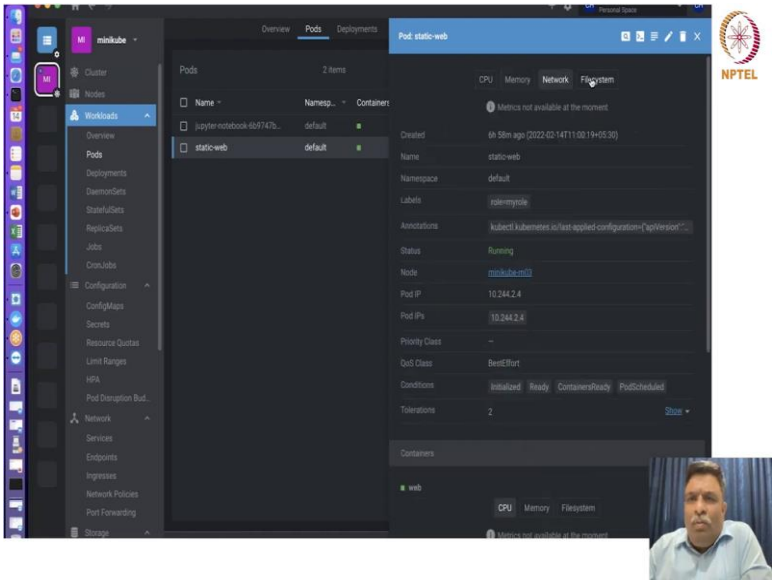


If you see this how much CPU utilization, what is the network it uses.

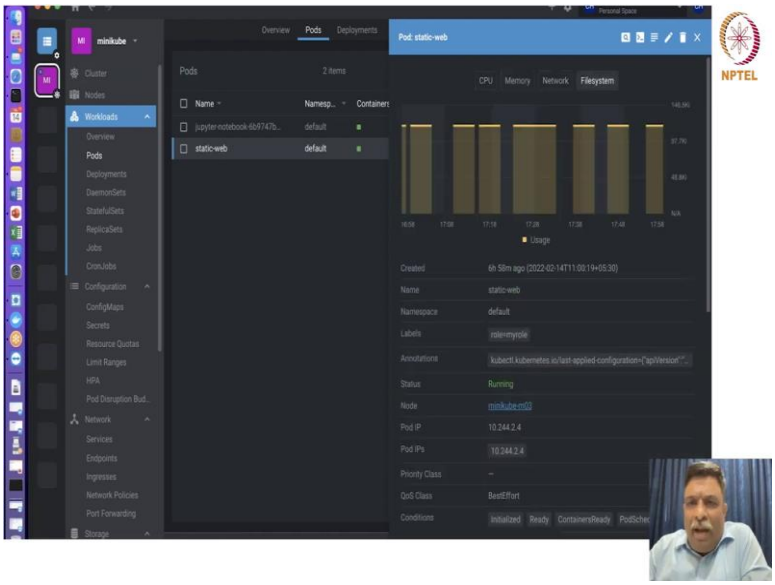
(Refer Slide Time: 29:08)



(Refer Slide Time: 29:08)

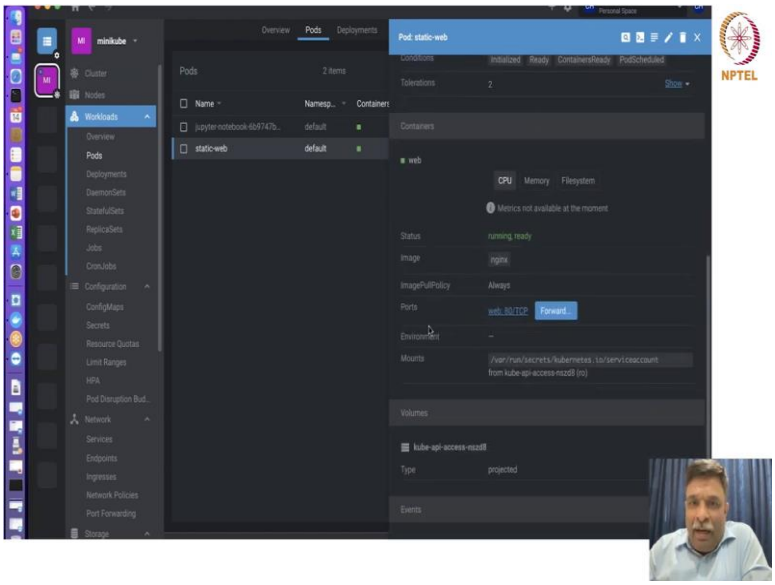


(Refer Slide Time: 29:09)



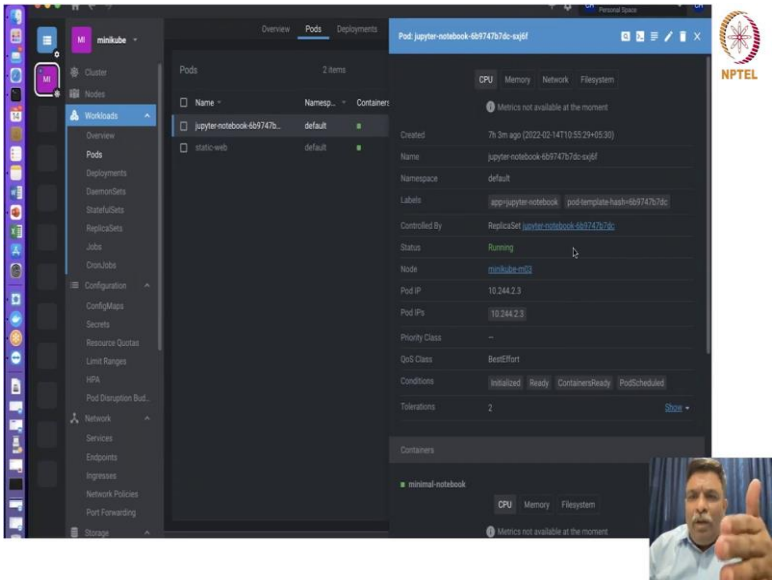
What is the file system ok. Where, what are the pods on which you can view it what is the status right.

(Refer Slide Time: 29:11)

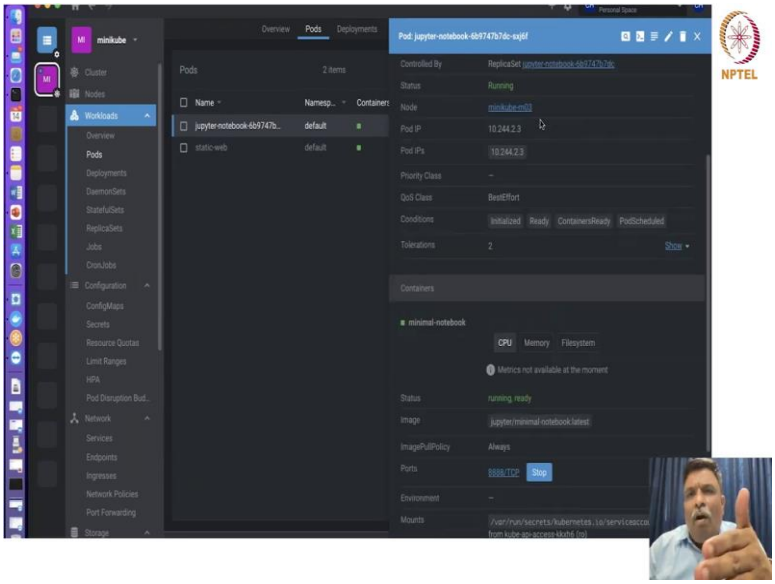


All of this is available to you ok. Similarly this is Jupyter Notebook.

(Refer Slide Time: 29:27)

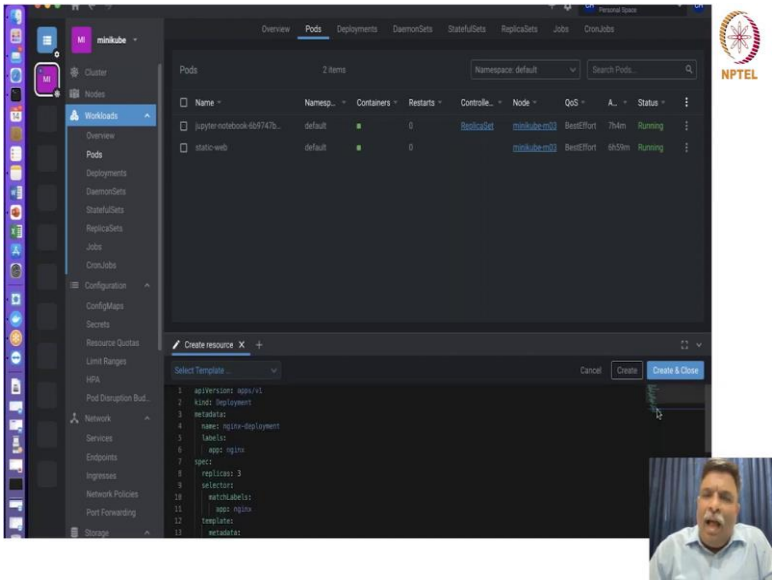


(Refer Slide Time: 29:28)



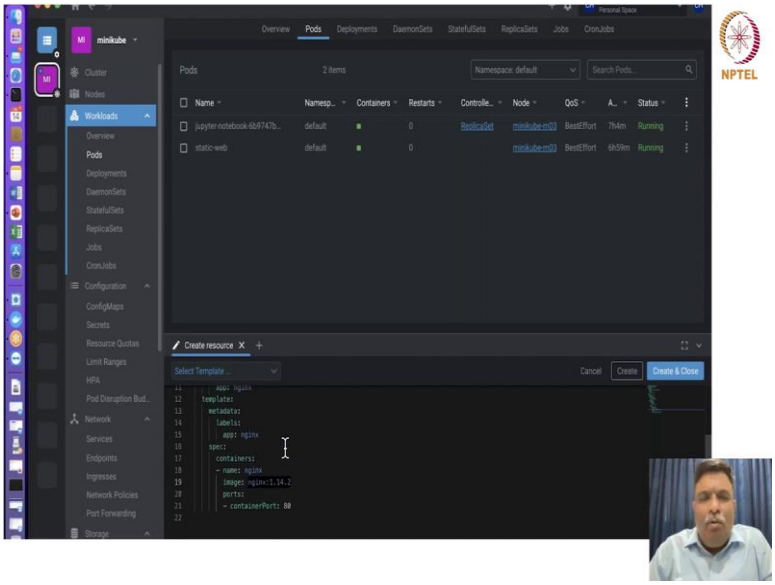
So, this Jupyter Notebook is running on node 3 whereas, this is also running on node 3 it is its not necessary that it has to run on node 2. It has actually allocated right based on some decision which it makes ok. This is also running on node 3. So, if you see this memory file system the CPU ok how its running ready at what pod. So, you will see that also. But before that let me just tell you if you want to create another pod right.

(Refer Slide Time: 30:04)



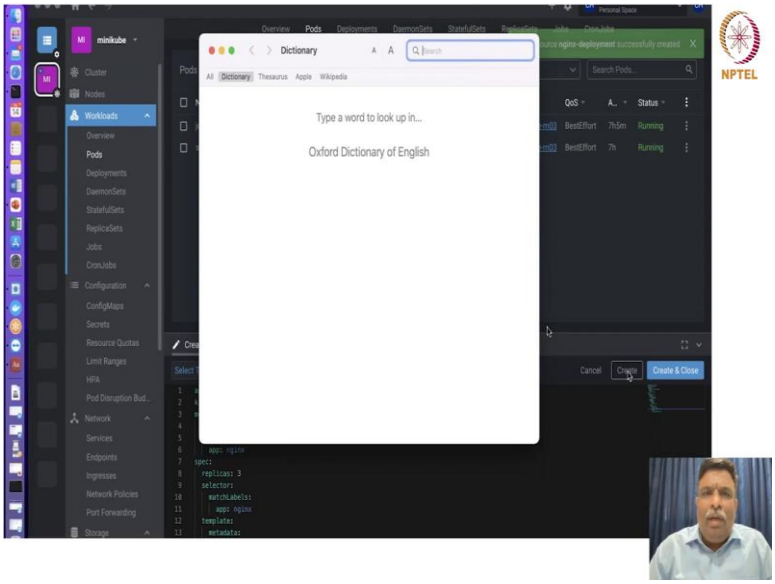
You have this resource right creating a pod and then you can use deployment, you can use something. So, I am just telling to showing to I am just showing you this that this is the API version right YAML file this what type of deployment name I have given.

(Refer Slide Time: 30:24)



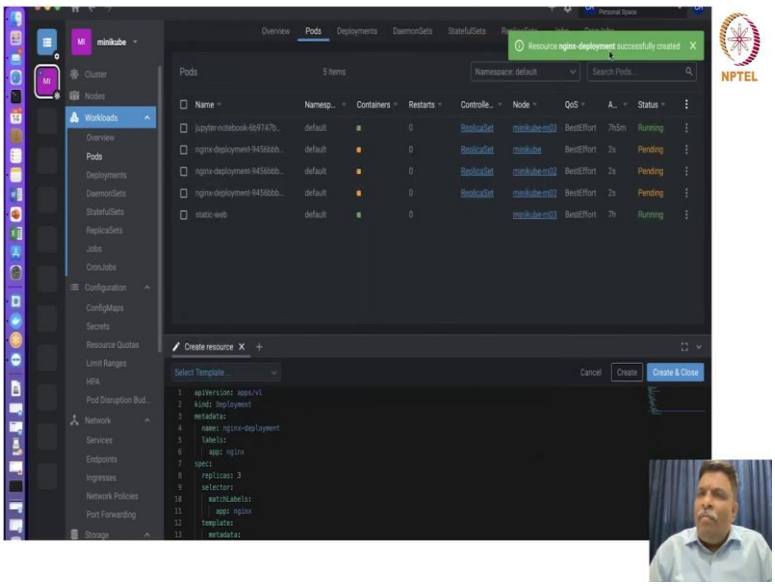
Which is the image it is running see here. So, this is a docker image right. So, I can actually when I am trying to run the Jupyter Notebook ok I will put the docker image which actually has a Jupyter Notebook and then I actually create it.

(Refer Slide Time: 30:46)



I can create another nginx now ok. I do not know why it is coming.

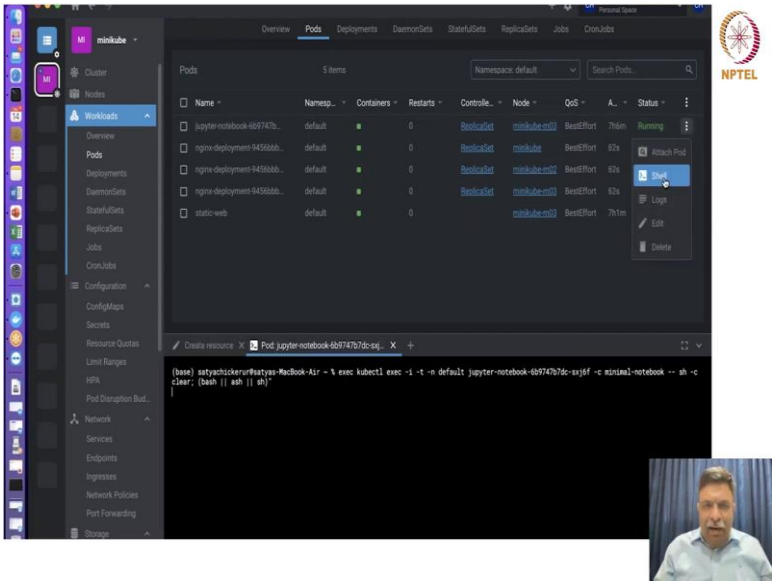
(Refer Slide Time: 30:49)



See resource nginx development deployment successfully created. So, see here now I suppose by mistake I clicked it three times maybe I do not know. So, it has created three instances right and its creating a replica set also of it and its pending you will wait for some two three minutes then we will see that it will actually all the three of them will be running. And if you see here this minikube sorry this particular deployment is running on node 2 and this is running on node 1.

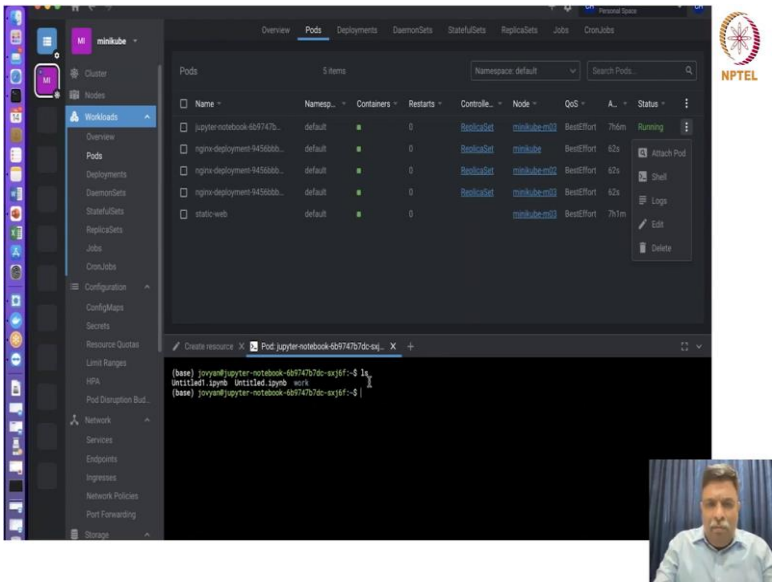
Now, ok or the master node ok and this was running on node 3. So, this also is now trying to run on node 3 and if you see all of them have the status now running. So, you can actually see that Jupyter Notebook is running.

(Refer Slide Time: 31:47)



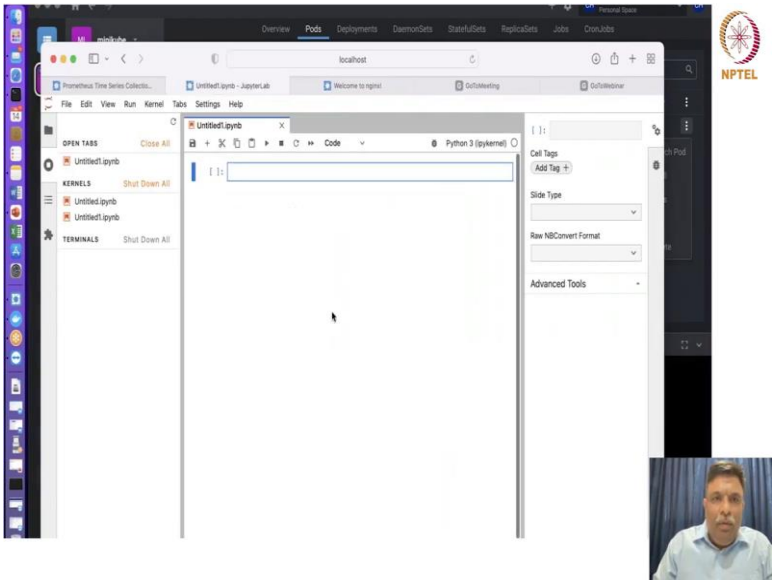
You can go to the shell.

(Refer Slide Time: 31:50)

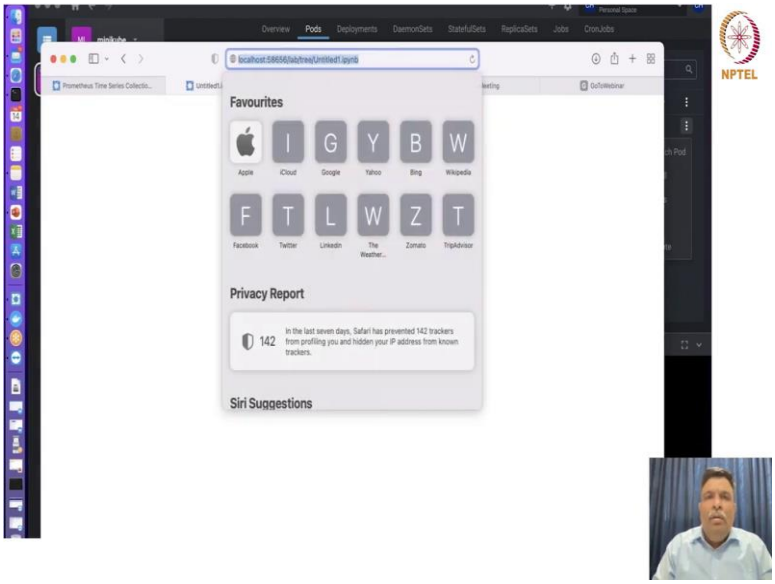


So, you can go to the root of that particular container you have got all of this files and folders there and if you see this Jupyter Notebook it is here its running its running on local host ok with this.

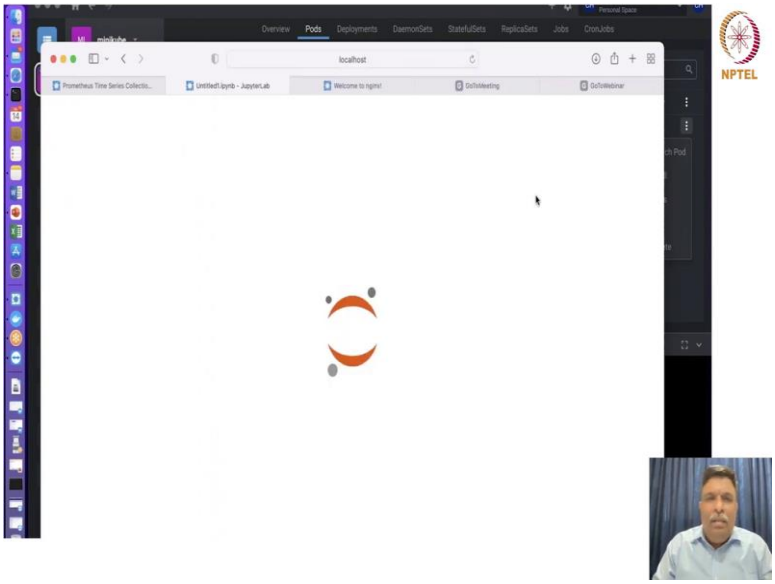
(Refer Slide Time: 32:05)



(Refer Slide Time: 32:09)

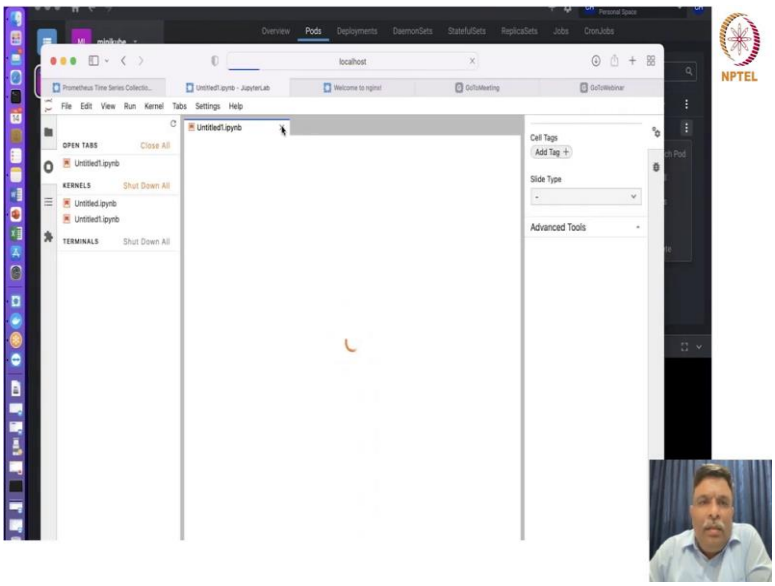


(Refer Slide Time: 32:13)



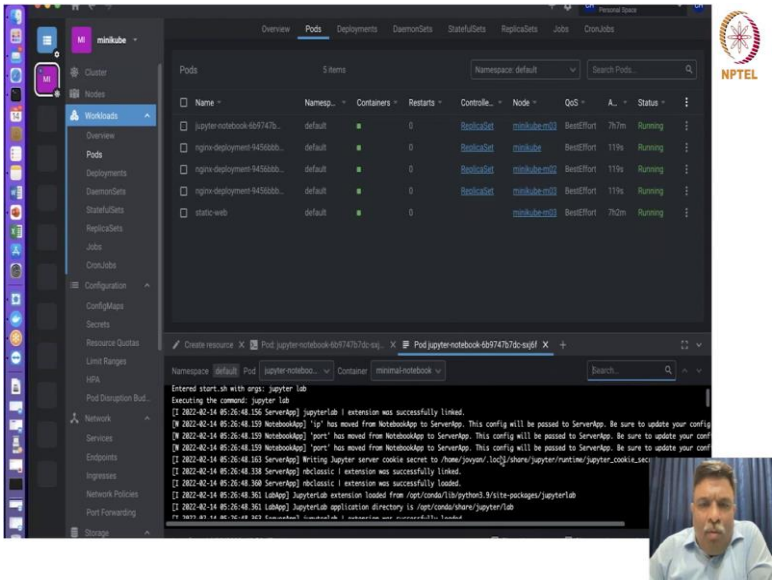
So, let me just try to connect it, yes.

(Refer Slide Time: 32:21)

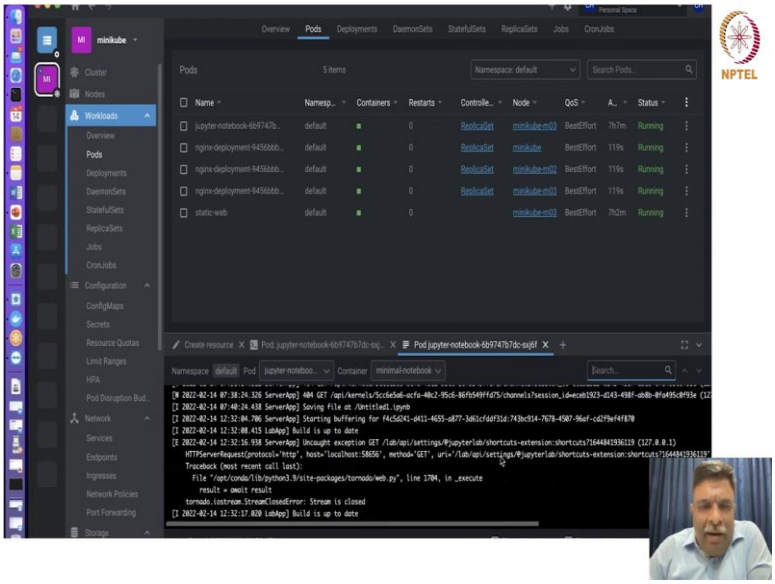


So, it will open its initializing. So, this is how actually it is going to happen. So, my Jupyter Notebook is running on node three you can attach a pod to a different this thing you can shift it you can see the logs.

(Refer Slide Time: 32:46)

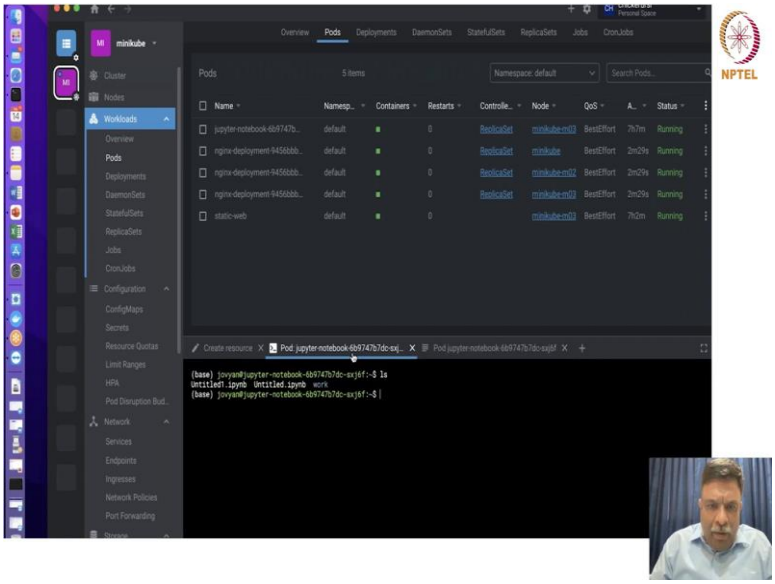


(Refer Slide Time: 32:47)

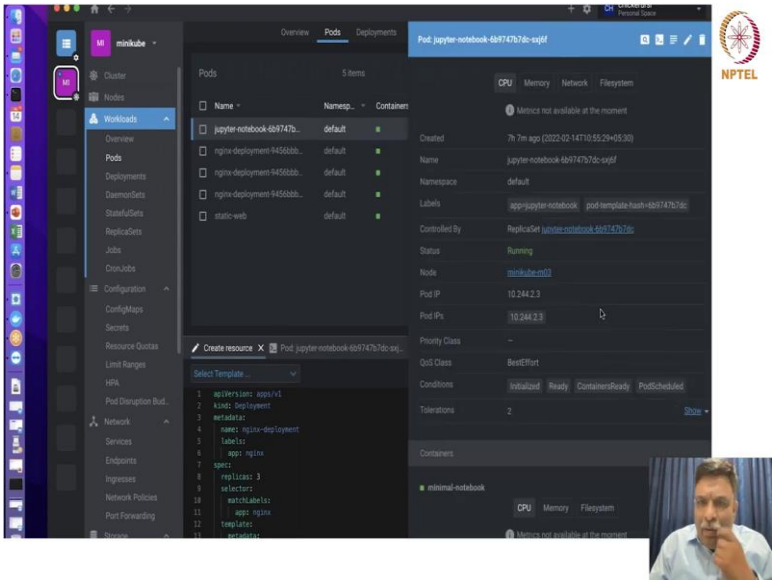


And if you see the logs, you will try to know right as to how much I have run it and at what particular pod it is running ok all of this actually can be known. So, let me just show you that then most probably that is what I thought I will share it with you today. So, you can actually create ok.

(Refer Slide Time: 33:17)



(Refer Slide Time: 33:17)



Resource ok by a template and if you want to see how we did it. So, you can see the image which we pulled we pulled the Jupyter Notebook minimal to latest from that ok that is how actually it has been done. So, I hope this is a bit of idea to you. Because from tomorrow onwards we will be developing your own applications right.

Deep learning and machine learning applications using pytorch using Tensorflow and all. So, ultimately once you develop those type of applications using dockers then you can

actually put into the pods and do the things very easily you can actually do such use such templates and do things very easily ok. So, that is it.

Thank you so much.