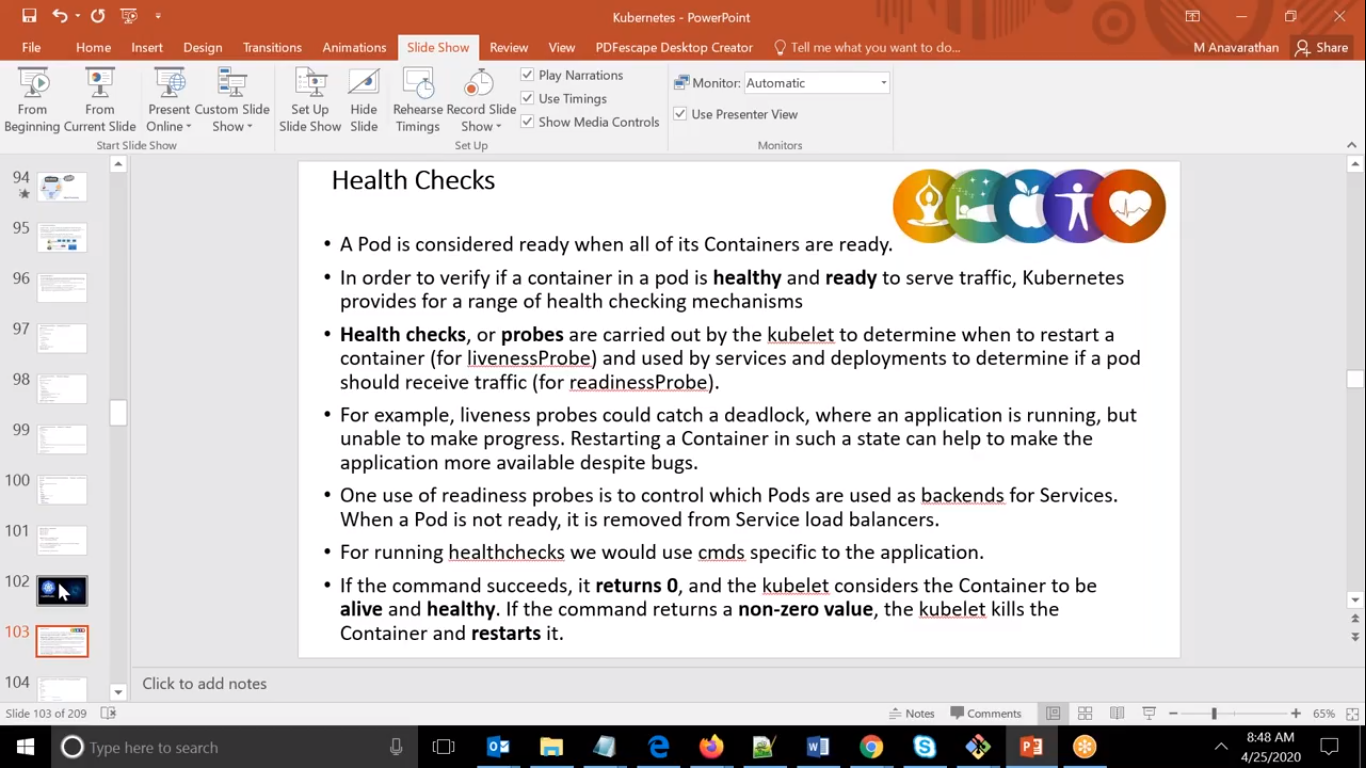
## Health Checks:-



We will be creating Pod for our work, And pod is said to be healthy if it is running successfully.

So we have one way we can check the health of container by verifying either it is healthy or not and that process is called is Health Checks.

So Health Checks are carried out by kubelet whichwill run some commands which will be running in all the pods available.

But we have to mention the frequency also when to run so probe will be carried out by the kubelet.

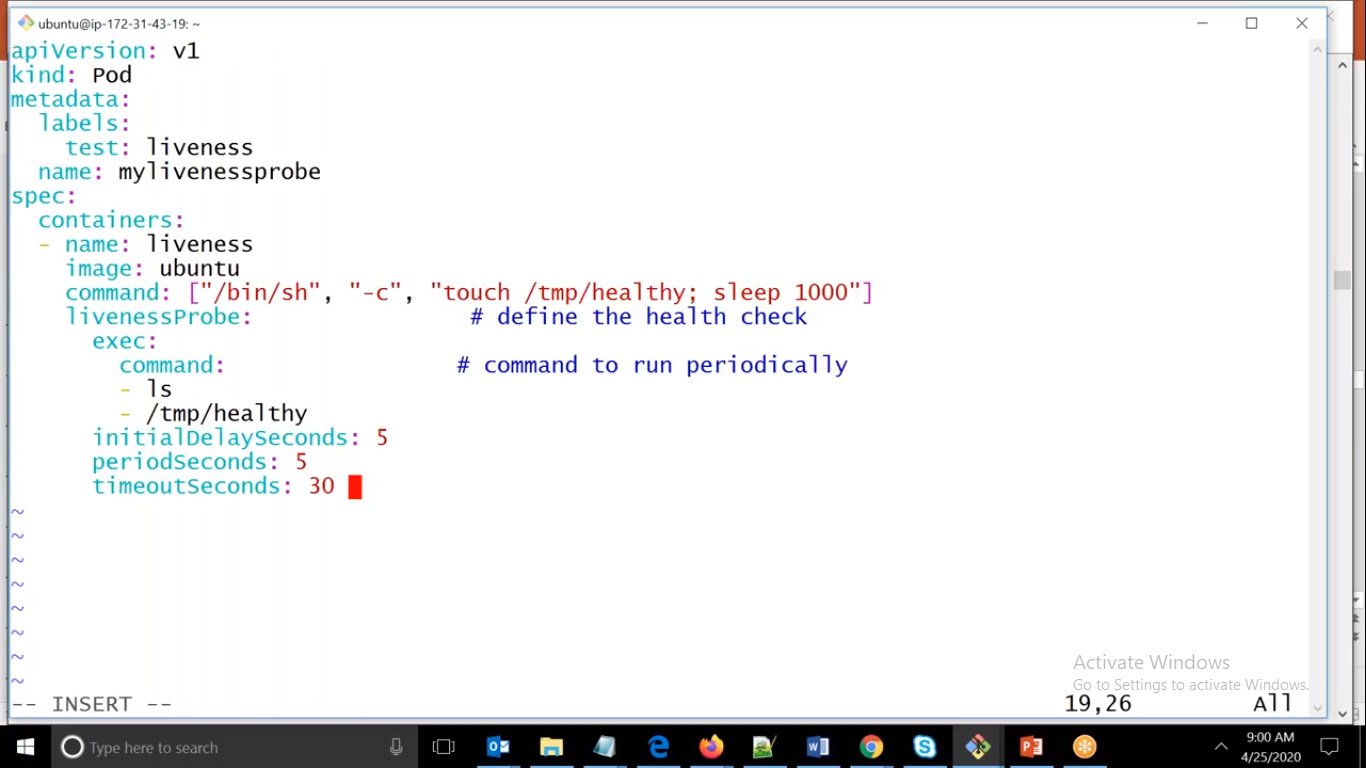
So that k8s will be running that command in all the pod.

And if it returns non zero then ks8 will considered it as unhealthy.

It will stop and delete the container.

We have 2 types of Health Checks.

1. Liveness
2. Readyness



Here we can see that we have created pod object.

and running one command, if we see in down we have given **periodseconds** : 5 what it means is after every 5th second it will start the health check and by that time if machine starts take more than 5 seconds then it will send non zero status and it will considered as unhealthy.

so again it will stop that container and recreated it again and it will again takes more than 5 seconds then it will be again exit..so it will be endless loop.

To overcome from this only we have given **initialDelaySeconds** so that it will wait for first 5 seconds then only it will start health checks..

so that it will not give wrong result and exit..to give initial delay seconds we should know how much time our container take to restart itself.

timeoutSeconds:30 means it will wait for the command to execute for 30 seconds if it does not able to do it in that time it will return non zero and it will be considered as un healthy.

ubuntu@ip-172-31-41-23:~$ ls /tmp/healthy

ls: cannot access '/tmp/healthy': No such file or directory

ubuntu@ip-172-31-41-23:~$ echo $?

2

ubuntu@ip-172-31-41-23:~$ touch /tmp/healthy

ubuntu@ip-172-31-41-23:~$ echo $?

0

ubuntu@ip-172-31-41-23:~$ ls /tmp/healthy

/tmp/healthy

ubuntu@ip-172-31-41-23:~$ echo $?

0

ubuntu@ip-172-31-41-23:~$

Here we can see that first time folder was not there so if we check the status of echo $? it is returning non zero i.e. unhealthy.

SO we have created it and again checked it then it returns zero.

**So in real scenario we have to create a such a script for health check so that based on success or failure it should return zero or non zero.**

Here if we delete the pod so nothing will be happened since it is pod.

SO we can check it by deleting the folder and it will return non zero so that it will recreated the new container.

Events:

Type Reason Age From Message

---- ------ ---- ---- -------

Normal Scheduled <unknown> default-scheduler Successfully assigned default/mylivenessprobe to ip-172-31-37-102

Warning Unhealthy 71s (x6 over 2m41s) kubelet, ip-172-31-37-102 Liveness probe failed: ls: cannot access '/tmp/healthy': No such file or directory

Normal Killing 71s (x2 over 2m31s) kubelet, ip-172-31-37-102 Container liveness failed liveness probe, will be restarted

Normal Pulling 41s (x3 over 19m) kubelet, ip-172-31-37-102 Pulling image "ubuntu"

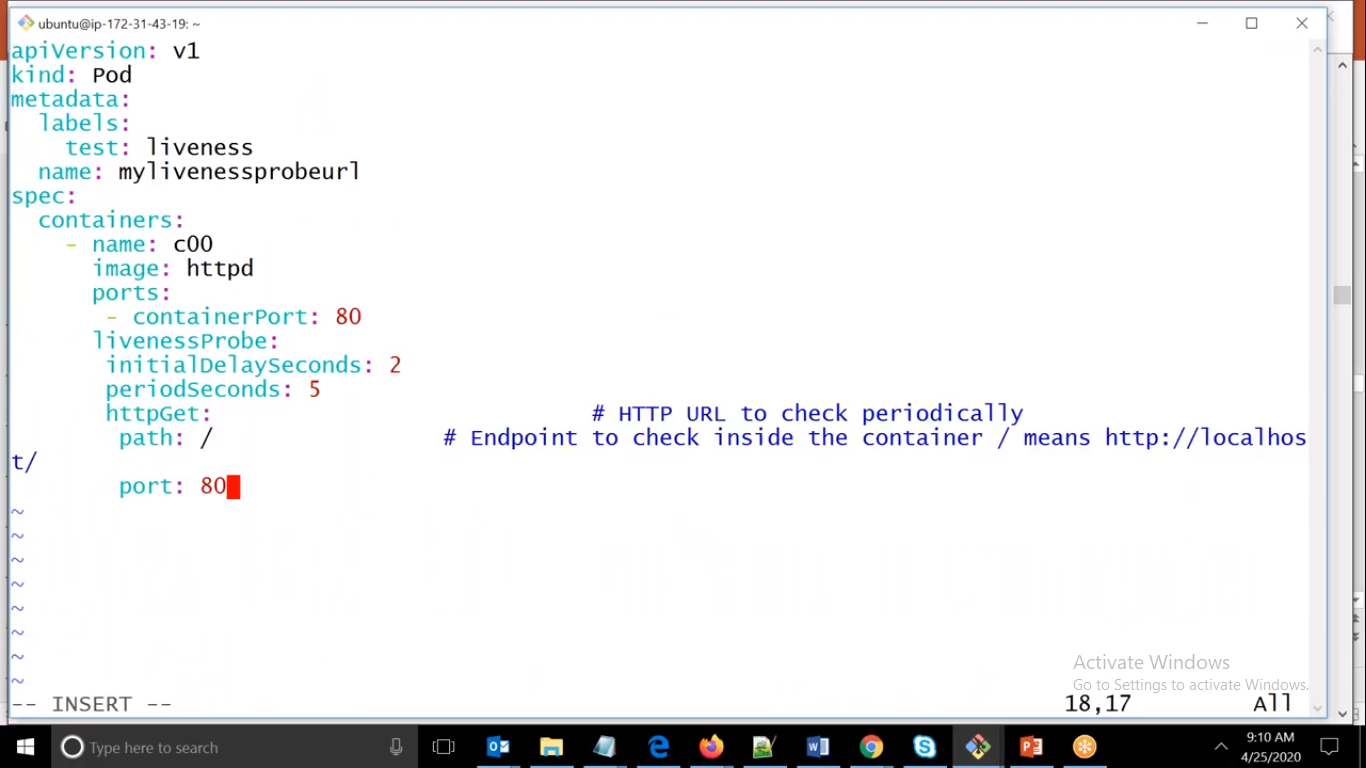
Normal Pulled 38s (x3 over 18m) kubelet, ip-172-31-37-102 Successfully pulled image "ubuntu"

Normal Created 38s (x3 over 18m) kubelet, ip-172-31-37-102 Created container liveness

Normal Started 38s (x3 over 18m) kubelet, ip-172-31-37-102 Started container liveness

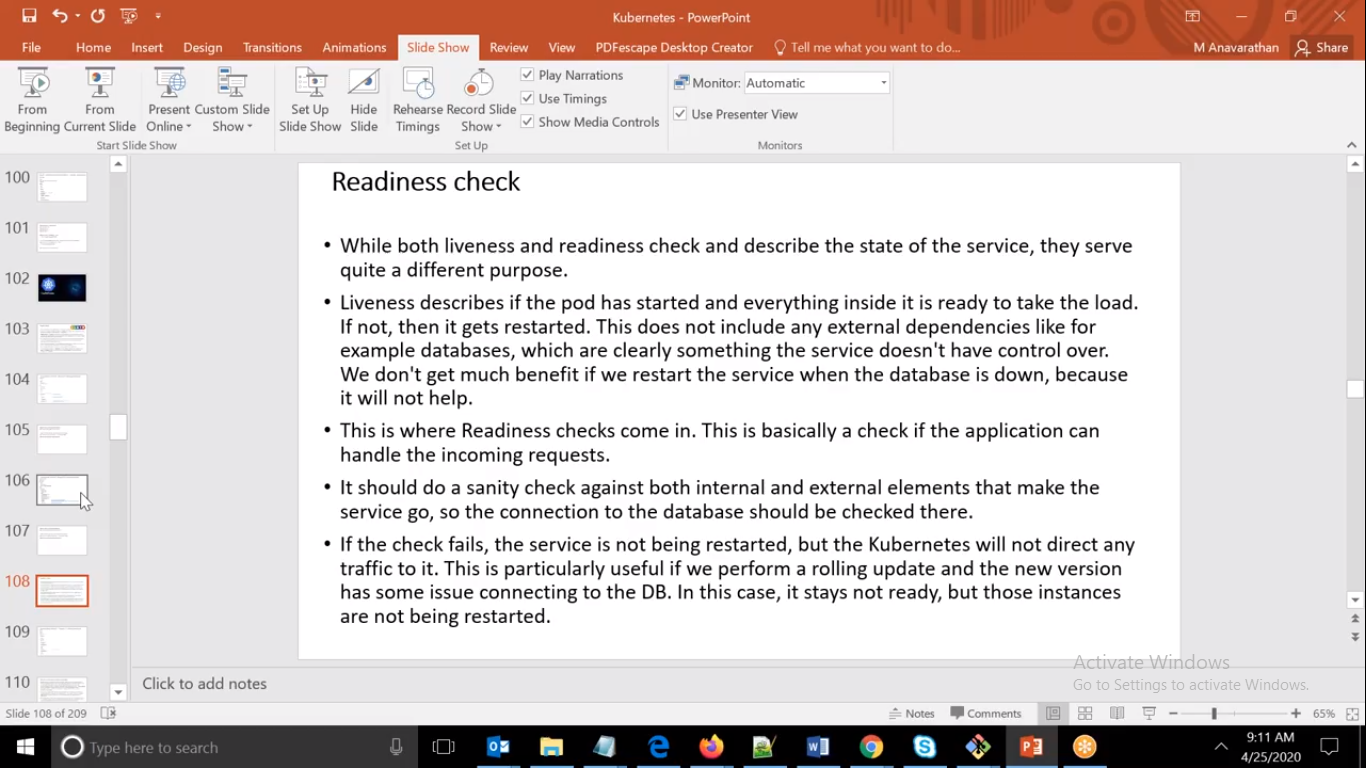
## 2.Liveness with URL

If we want to check any web application checking its port is available or not we can do that alco.



Here we have given containerPort as 80 and to check it we have given httpGet so that it will go inside the container and check if the port is available or not and if available either it is listening it or not.

Another way to check the health by using **Readiness**



Liveness check means whether application is started or not.

Is it listening on port or not so more or less it will check the status of application i.e. it is live or not.

But we want to check even our application is started successfully but is it ready for use or not.

i.e. is it able to connect to the database or not to check that health we will be using another health check called as readiness check.

In general we will be using Liveness checks only.

But in certain scenario we have to check either internally is it able to connect other application or not.

