1. **Create a pod using file method:-**
    - Image name = nginx
    - Pod name = myapp
    - Set label on pod --- app=website
    - Define containerport property in spec section
    - Container name = mysecondcontainer

2. **Create a pod using file method:-**
    - Image name = ubuntu
    - Pod name = sleeper
    - Set label on pod --- app=sleeper
    - Container name = mythirdcontainer
    - command should be run at the time of pod starting === sleep 10000
    - make sure container is in running state

3. **Create a pod using file method:-**
    - image name = mysql
    - Pod name = mydb
    - Set label on pod --- app=database
    - Container name = container4
    - Make sure container is in running state
    - Define containerport property in spec section

4. **Perform action on previously created pod named container4:-**
    - Get a shell into container and login to database
    - Create a DB user named tgindia
    - Allow super user permission to tgindia user
    - Create a database named Kubernetes

- Make sure tgindia db user should have full access on all databasesand tables

5. **Perform action on previously created pod named mysecondcontainer:-**
    - Get a shell into container
    - Create a index.html file in document root and this file should contains a line “I am a Kubernetes Engineer”
    - On your machine curl pod’s ip and this line should print “I am a Kubernetes Engineer”

6. **Perform action on previously created pod named myfirstcontainter:-**
    - create an index.html file on your machine and write a line “I am a DevOps engineer”
    - copy this index.html at document root location into pod
    - on your machine, curl pod’s ip and it should print a line “I am a DevOps engineer”

7. **Create a pod using file method:-**
    - Pod name = nonstoppod
    - Image name = alpine
    - Set label on pod --- app=nonstop
    - Container name = autorestart
    - Set a restart policy on pod = always
    - command should be run at the time of pod starting === sleep 15
    - make sure this pod will restart in every 15 seconds (you can use watch command for live check)

8. **Check which pod is using the most ram in cluster and on which machine that pod is running**

9. **Check how many numbers of pod is running in your cluster in all namespcaces**

10. **Check the log of pod named mysecondcontainer and save those logs in /tmp/2log file**

11. **Describe pod named mydb and check Events section, save that section in a file /tmp/dbevent file**

12. **Check the status of node which are running in your cluster:-**
    - Check the cluster version and save into a file /tmp/clusterversion.txt
    - Check the label of master node machine and save all labels in /tmp/controlplane-label.txt file
13. **Describe all the worker node and check which node have less cpu and available ram**

14. **Check the worker node capacity and allocatable resources**