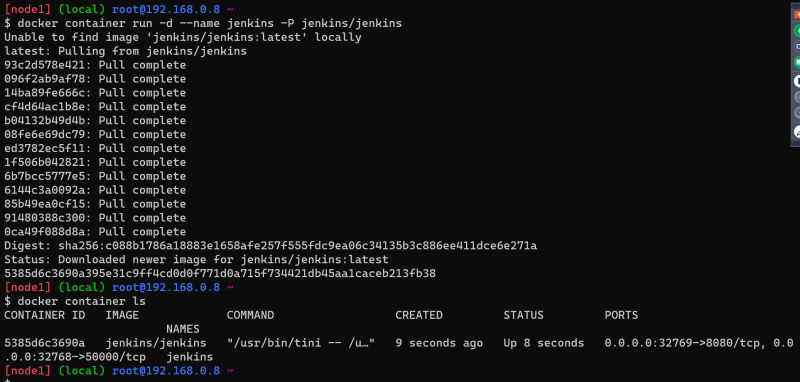
**Interactions**

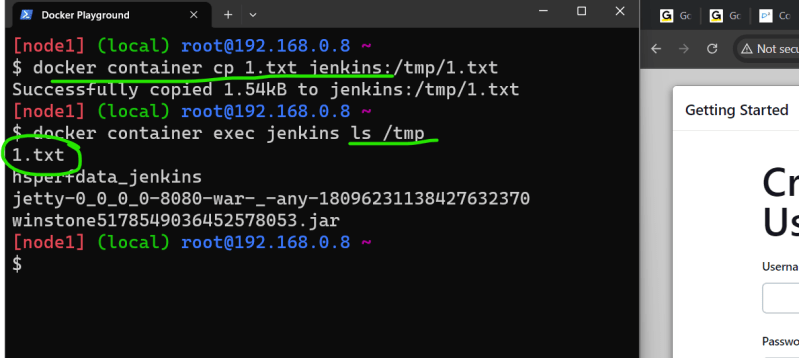
* We can interact with running or stopped containers
  + attach: We will be able to connect to containers main process’s STDIN/STDOUT/STDERR.
  + cp: This allows to send or recieve content to/from container
  + exec: Execute a command inside containers’s isolation.
  + logs: Here we can review all the STDERR and STDOUT
* Playground

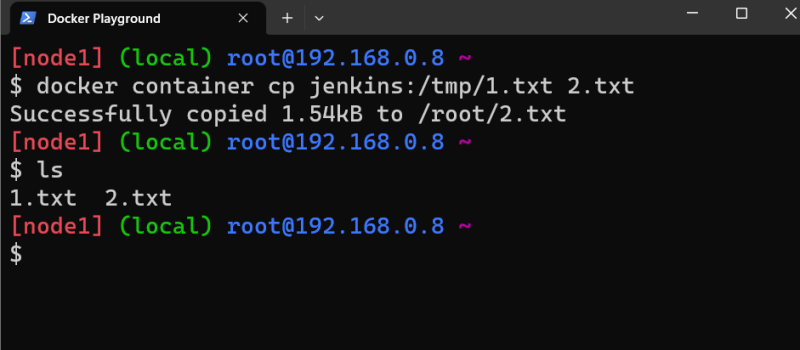
Run a jenkins container in a detached mode  


now attach to the jenkins container

now if we execute ctrl+c i.e. exit the main process will be exited and the container will go into exited state

To view logs use docker container logs <cont-id/cont-name>

Copy files from docker host into contianer  


Copy files from container into docker host  


**Limiting Host resources**

* Lets review some options for container resource consumption. We will be able to limit access to CPU,memory and block devices
* There are two types of limits
  + soft limits: These represent reservation i.e container could consume more than declared but mininum soft limit
  + hard limits: Thse represent container will not get more than declared value
* Container by default can consume all of host resources (no limits)
* Options:
  + –cpu-period and –cpu-quota: This is specified in micro seconds and will modify cpu limits
  + –cpu-shares: This manages weight for containers main process. This is soft limit.
  + –cpus
  + –memory: maximum amount of memory for your container (hard limit)
  + –memory-reservation: this is soft limit
  + –blkio-weight
  + –blkio-weight-device

**Building Docker Images**

* Docker image is required to create container.
* Docker images are OCI compliant i.e. they can be executed in other container runtimes as well
* Docker image will have all the necessary contents to run an application
* We need to build docker images to run our application inside container. This process is called as *Containerization*
* To Build docker images we have 3 ways
  + Start from scratch and add components file by file
  + Create a container with some image, make changes to run your application and create a image from the running contianer (commit)
  + Using a file with all the instructions to create image (Dockerfile)

**Creating images from scratch**

* Docker has a reserved image (base-image) that is is empty & is known as scratch
* Even os components which are necessary to start the main process in container has to be copied by us

**Creating Images interactively**

**Game of life**

* <https://referenceapplicationskhaja.s3.us-west-2.amazonaws.com/gameoflife.war> for the war file
* To run this application, we need tomcat 8 or 9 with jdk 8 installed
* Copy the war file into webapps folder

sudo apt update

sudo apt install openjdk-8-jdk tomcat9 -y

* Create a container in an interactive way or detached mode

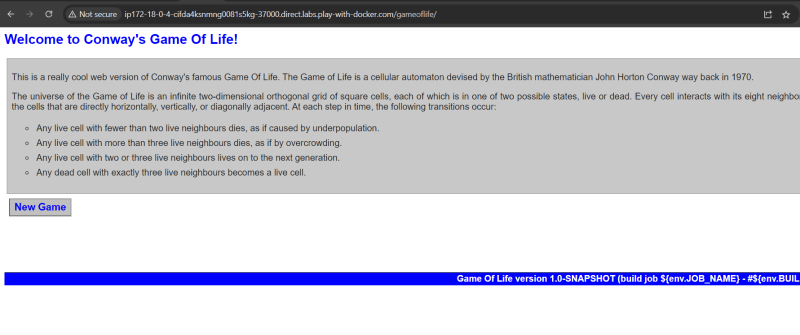
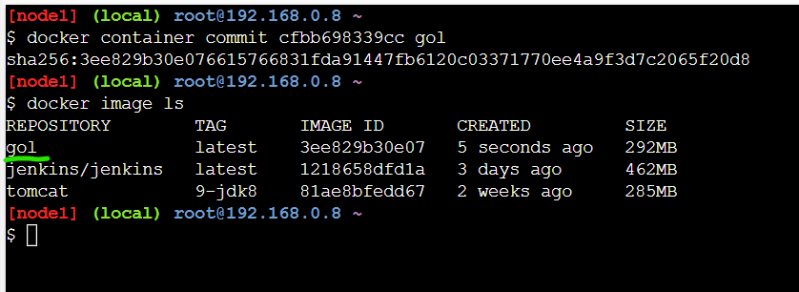
docker container run -d -p 37000:8080 --name gol tomcat:9-jdk8

docker container exec -it gol /bin/bash

cd webapps

wget https://referenceapplicationskhaja.s3.us-west-2.amazonaws.com/gameoflife.war

# now access the application over http://<ip-docker-host>:37000/gameoflife

  
  
\* Now let’s create a new container with this image and it works as shown in class

**Exercise**

* Create a docker image using commit command for
  + spring petclinic
  + nopcommerce