GIC-Game Inside Container

A report submitted for the course named Project II (CS-300)

By

Pranav Birendra Pathak

Bachelor of Technology, IV Semester

Roll No. 17010112

Under the Supervision and Guidance of

Dr. Kabita Thaoroijam

Department of Computer Science and Engineering

Indian Institute of Information Technology Manipur

June, 2019

Abstract

The aim of this project is to deploy the game on container using Docker. The main focus area through this project is to find a quick and efficient way to deploy new versions of the Games as well as application for production environment. Working objective will be based on creation of Dockerfile, which is one of way to create a docker image. This image will contain all the requirement for game to run inside it and it deployed for use. The traditional technique uses the virtualization techniques to isolate users and share resources among dedicated servers, however it inflict the performance and has lots of defect. To overcome this, we use here Container which is a Operating-system-level virtualization and very fast to deploy.

Keywords - Docker, Containers, Wine, Virtualization, Open-Source, Linux, Networking

Declaration

I declare that this submission represents my idea in my own words and where others' idea or words have been included, I have adequately cited and referenced the original source. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/sources in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and can also evoke penal action from the sources which have thus not been properly cited or from proper permission has not been taken when needed.

(Signature)
(Pranav Birendra Pathak)
Date: (17010112)

Department of Computer Science & Engineering
Indian Institute of Information Technology Manipur

Certificate

This is to certify that the project report entitled **"GIC-Game Inside Container"** submitted to the Department of Computer Science and Engineering, Indian Institute of Information Technology, Manipur, in partial fulfillment for the award of the degree of **Bachelor of Technology** in Computer Science and Engineering, is a record of bona fide work carried out by **Mr. Pranav Birendra Pathak**, Roll No. 17010112.

No part of this report has been submitted elsewhere for award of any other degree.

(Dr. Kabita Thaoroijam)

Assistant Professor
Supervisor

Date:

Department of Computer Science & Engineering
Indian Institute of Information Technology Manipur

Certificate

This is to certify that the project report entitled **"GIC-Game Inside Container"** submitted to the Department of Computer Science and Engineering, Indian Institute of Information Technology, Manipur, in partial fulfillment for the award of the degree of **Bachelor of Technology** in Computer Science and Engineering, is a record of bona fide work carried out by **Mr. Pranav Birendra Pathak**, Roll No. 17010112.

No part of this report has been submitted elsewhere for award of any other degree.

Dr. Nongmeikapam Kishorjit Singh

Assistant Professor and Head,

Department of CSE

IIIT, Manipur

Date:

Department of Computer Science & Engineering
Indian Institute of Information Technology Manipur

Certificate

This is to certify that the project report entitled **"GIC-Game Inside Container"** submitted to the Department of Computer Science and Engineering, Indian Institute of Information Technology, Manipur, in partial fulfillment for the award of the degree of **Bachelor of Technology** in Computer Science and Engineering, is a record of bona fide work carried out by **Mr. Pranav Birendra Pathak**, Roll No. 17010112.

Examiners Signature:

Acknowledgement

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals. I would like to extend my sincere thanks to all of them. I am highly indebted to **Dr. Kabita Thaoroijam** for his guidance and constant supervision on my project **"GIC-Game Inside Container"** as well as for providing necessary information regarding the project and also for their support in completing the project. I would like to express my gratitude towards my parents for their kind co-operation and encouragement which help me in completion of this project. I would like to express my special gratitude and thanks to my mentor for giving me such attention and time. My thanks and appreciations also go to my colleague in developing the project and people who have willingly helped me out with their abilities.

- Mr. Pranav Birendra Pathak

Contents

List of Figures

Chapter 1

Introduction

“Use Your Knowledge For Solution, Not For Criticism”

- Pranav Birendra Pathak

First images comes in mind wen we heard the word container is BOX, well container is a box where an application is packed inside and shipped. prior to container technologies, deploying an application usually took quite a long time, which cost the time and resources. with the popularity of containerization through Docker and Kubernetes, the whole process became more streamlined and standardized. The container technologies can be used to automate the deployment process quite effortlessly and therefore the value of a well configured container platform is intangible. Docker is a tool to create an image of an application and the dependencies needed to run it. The image can then later be used on a containerization platform such as Kubernetes.

1.1 Need of this project

Imagine a situation where one want to test his or her application with different dependencies.It's obvious a time consuming idea to create a multiple VM or boot n their base machine. With this container technology one can create an Operating-system-level virtualizationwith one click , which will we isolated to each other, more reliable and secure and solves the dependencies problem.

The idea behind creating this GUI application is to to give the fast and efficient way of deploying application in Operating-system-level virtualization called "container" through Game As A Service platform.

1.2 Project overview

This major focus on this project will be given on:

1. Docker (A containerisation tool)
2. Wine (A layer on which windows application can be run)

Since this project is upon container, Setup can be made on any linux-based os.

1.2.1 Timeline

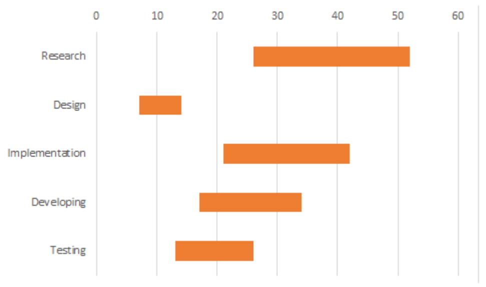


Figure 1.1: Timeline of completiton

The Figure 3.3 shows the steps taken to implement the system.The box represent that the work has been done untill the end of that block.The Whole system is divided into 5 phases some of them i have done in a parallel fashion which can be seen in the graph.

Chapter 2

Existing System Study

Outline: This chapter presents the following:

1. A brief introduction about existing system.
2. Problems in existing VM vs Container.

2.1 Introduction

To deploy an application or any kind of services , normally used nowadays is cloud provider VM or manually configured . Normally, many services are deployed on a single server for better utilization of resources which is not a good technique due to maby security and other reason. It leads conflict to resources and dependencies problem. There are many cases where resources gone to wasted and more chances of getting data loses with slight disturbances. When times comes to scale the server,if developers got stuck with time which lead to slow deliveries of product, customer unsatisfactory and loss, this will lead as great barrier to companies.

2.2 Problems In Existing VM vs Container

You may know how fast and efficient is to boot an OS in Virtual Machine as compare to boot on physical baseOS. Though comparing Virtual machines with current container technology there are many upsides to it. VM's are bulky and slow to serup and configure whereas when taking account to container, it is fast to setup and configure whithin in a click. Due to agile nature of container, it is recommended as best practices for DevOps Process. Integrating CI-CD pipeline with container, enables the fast testing of the newly developed versions of the application and pushing new versions to production environments. Even though containers and virtual machines are quite different, they can be combined to get the good sides of them both.

The robustness of the virtual machine and the agility of containers, provide a good basis for the deployment process. (Docker container 2018, cited 13.1.2018.)

"Virtual machines (VMs) are an abstraction of physical hardware turning one server into many servers." (Docker container 2018, cited 13.1.2018).

Virtual machines are built on top of a hypervisor, which allows several virtual machines to run on one machine . Each virtual machine Figure 2.1 instance contains a full copy of an operating system and all the dependencies needed to run, which take up several gigabytes of storage space

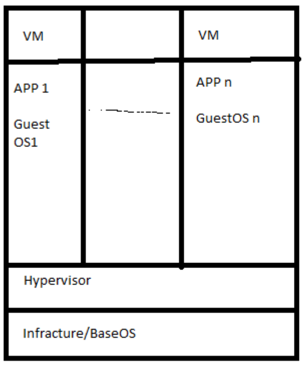


Figure 2.1: Virtual Machine Architecture

2.3 Observations

I have observed that my project that is GIC-Game Inside Containe would be of immense help to the client as it is fully GUI deployed on container which will give a new, better and fast approach to deploy their product using container.curently everything is done vm, which results in a lot of time consumption, and it is not reliable.This project will be an ideal solution for cloud providers and App deployment

2.4 Summary

In this chapter, we describe the existing system of deploying Application through VM technology and the problems faced by it. One of the Key Factors in this fast running era is speed and automation. You must be always be ready for scale in out your production. Here comes the role of this project. With one click entire production will create and can be deleted.

Chapter 3

System Analysis, Design & Implementation

3.1 Introduction

In this ever growing technology and advancement world, there was a myth in the market that GUI can't be brought on Docker or container. This was due to the purpose for creation. They are not meant for the deployment of Game inside it, but it does'nt meant that this cant be done rather it proved to be a great way to deploy the GUI app on it.

The main component used in this project is Docker. The procedure is as, Through the Dockerfile of Docker for deployment of the application Dockerimage will be create as the requirement. A Dockerfile has all the instructions on how to build the final image for deployment and distribution. The images that are made are reusable perpetually. The image is then used by Kubernetes for the deployment. The benefits of Docker are, for example, the reusability of once created resources and the fast setup of the target environment, whether it is for testing or production purposes. This is achieved through container technologies made possible by Docker and Kubernetes. Container technology is a quite new technology which has been growing for the past five years.

3.2 Objective

The key objective for this project is to build game container and based on it create a private cloud on Linux based OS where user can access the game services without any cost of license and can experience the power of container.

1. With one click, entire OS is booted and ready with gaming console
2. With one click , entire system is destroyed

3.3 System Designing

This Game service is a kind of Internet-based computing that provides shared pro-cessing resources and data to computers and other devices on demand. The user canselect the software of his/her choice and directly uses the software as if it is installed onhis own system but actually not.In reality, the game application runs on the server but provide the graphical window on the client's system so the client can interact with it and play.

Designing is the most important and the most efficient function while software development. Without a proper design, it is very difficult to develop an appropriate software that fulfills nearly all user demands. Therefore, managing designing part in an organization is a critical activity. A designer needs to ensure that the design created by him can be easily understood by the all the members of developing a team. A proper design will allow the coder to implement the system development planning properly. Similarly, the backend part that includes the database management plays a key role in any of the systems. Thus, the team working on this field must know proper handling and management of database and its tools

The system is developed using the Wine, Docker as a container tool, Python language for Interface.Module used are:

1. Tkinter: This module is used for the Client GUI to interact with the services
2. Hashlib: This module is used to encrypt and decrypt the Client Data

The data of user is stored in the file called ”accounts.txt”. We have used file as a database to store data

Dockerfile is used to create the Docker image, which is used to launch the new container

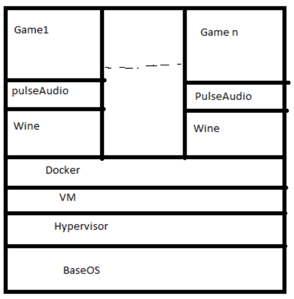


Figure 3.1: System working Design model Architecture Overview

Figure 3.1 shows all the layer upon which games is running.The layer upon docker the the basic requirement to run the windows application.

3.3.1 Wine

Wine is not an emulator rather it is a layer that has capability to run windows application on several POSIX-compliant operating systems, such as Linux, macOS, BSD. Basic Overview of wine is it translates Windows API calls into POSIX calls on-the-fly, eliminating the performance and memory penalties of other methods and allowing you to cleanly integrate Windows applications into your desktop.

Here we use wine layer inside docker to run the windows games.Not to forget that alone Wine is not sufficient to run windows games and have a fun of gui. Prior to Wine layer there is necessity to have a graphical entity present Upon the wine layer, Linux achieve the power to run windows application.

3.3.2 Pulse-audio

In simple words, pulse-audio is used to transfer the audio from one system to other. This tool is used here to transfer the audio of games and application generated to client. Pulse-Audio is a network-capable sound server program distributed via the freedesktop.org project. It runs mainly on Linux, various BSD distributions such as FreeBSD and OpenBSD, macOS, as well as illumous distributions and the Solaris operating system

3.4 Methodology and Implementation

Figure 3.2 there are 5 layers upon which game will run. Each layer act as a support layer to other layer. Each layer can be described as follows:

1. Layer 1 (BaseOS): In this layer, there will be compute engine i.e RAM and CPU or all the hardware requirement to boot the os upon with graphical card.A complete infracture needed to run any application.
2. This layer is called Hypervisor layer. This layer act as a layer for vm's to run. This layer support the upper layer.
3. Layer 3: In this layer, VMs are installed, where upon work will be done. In this case oracle VMware is used. It is a good practice to have vm as it will act as the good utilization of resources
4. Layer 4: This is the layer of DOcker where dockerfiles are kept with all the docker requirement. Inside Dockerfile there will be several setup done for docker images . This Dockerfiles is used to boot or create the images.
5. Layer 5: This is the application layer upon which the games launched through the docker images which will be available for the users to play.

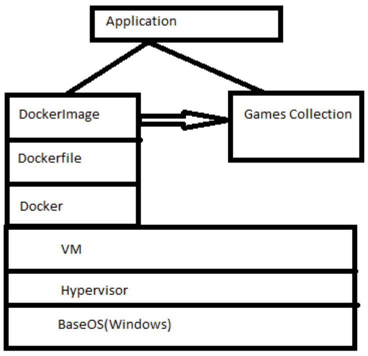
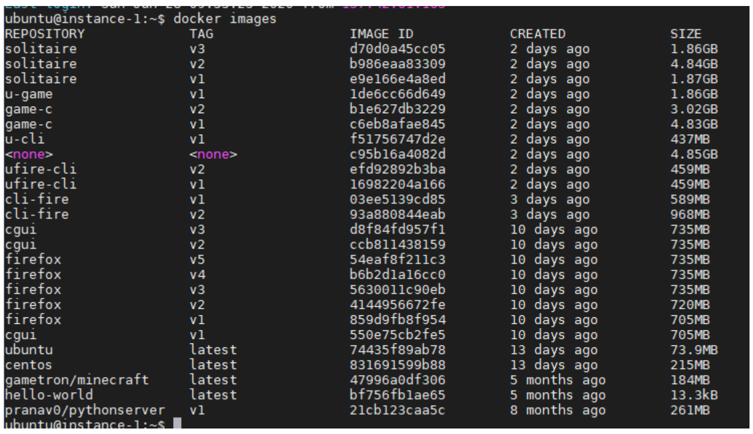


Figure 3.2: working Overview of GIC

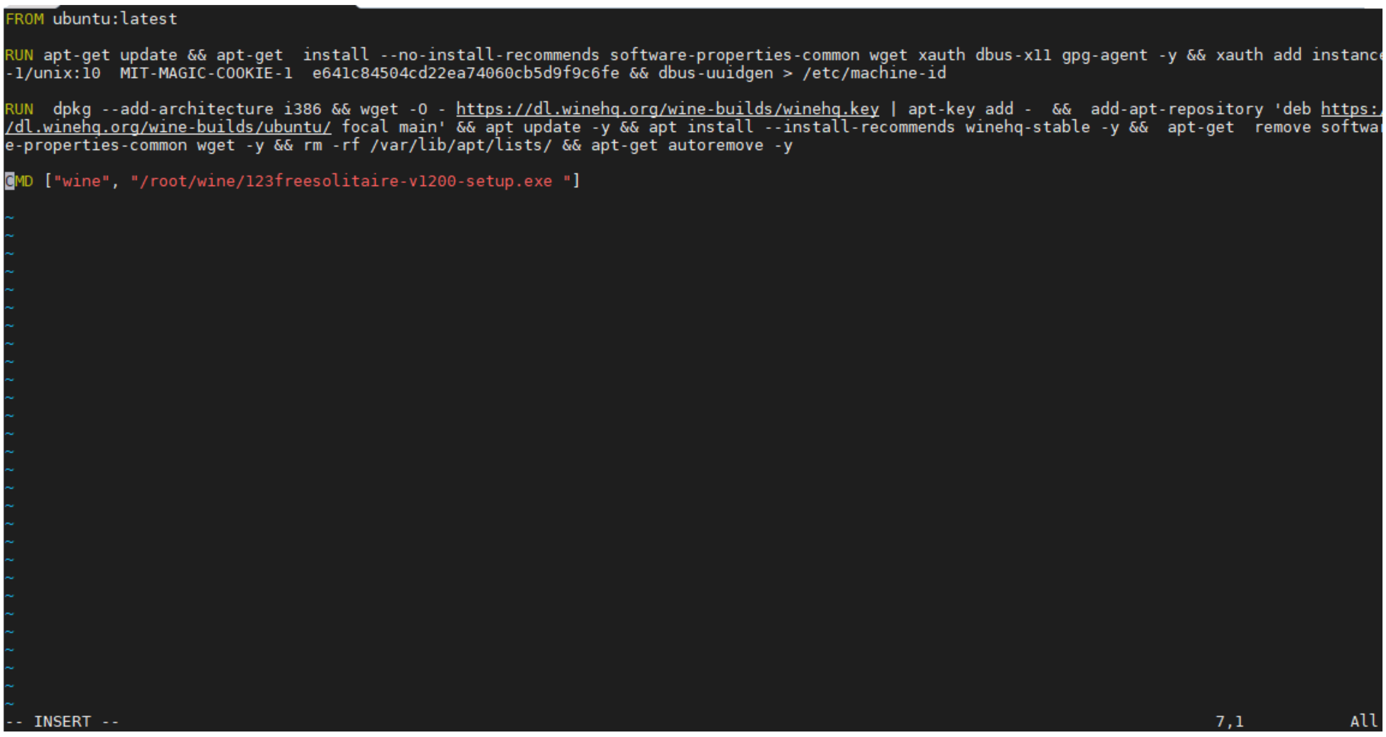
3.4.1 Dockerfile

DOckerfile is a best option to craete your own image from an application.The purpose of the Dockerfile is to automate the image building process,in which all the necessary dependencies and libraries are installed.It makes the build process more efficient through reducing the size of the final image, which is determined in the last phase of the build (FIGURE 5) (Dockerfile 2017, cited 25.12.2017.) Building images by your own requires multiple stage, testing and verifying. DOckerfile just automate the process.



```
ubuntu@instance-1:~$ docker images
REPOSITORY              TAG                 IMAGE ID            CREATED             SIZE
solitaire               v3                  d70d0a45cc05        2 days ago          1.86GB
solitaire               v2                  b986eaa83309        2 days ago          4.84GB
solitaire               v1                  e9e166e4a8ed        2 days ago          1.87GB
u-game                  v1                  1de6cc66d649        2 days ago          1.86GB
game-c                  v2                  b1e627db3229        2 days ago          3.02GB
game-c                  v1                  c6eb8afae845        2 days ago          4.83GB
u-cli                   v1                  f51756747d2e        2 days ago          437MB
<none>                  <none>              c95b16a4082d        2 days ago          4.85GB
ufire-cli               v2                  efd92892b3ba        2 days ago          459MB
ufire-cli               v1                  16982204a166        2 days ago          459MB
cli-fire                v1                  03ee5139cd85        3 days ago          589MB
cli-fire                v2                  93a880844eab        3 days ago          968MB
cgui                    v3                  d8f84fd957f1        10 days ago         735MB
cgui                    v2                  ccb811438159        10 days ago         735MB
firefox                 v5                  54eaf8f211c3        10 days ago         735MB
firefox                 v4                  b6b2d1a16cc0        10 days ago         735MB
firefox                 v3                  5630011c90eb        10 days ago         735MB
firefox                 v2                  4144956672fe        10 days ago         720MB
firefox                 v1                  859d9fb8f954        10 days ago         705MB
cgui                    v1                  550e75cb2fe5        10 days ago         705MB
ubuntu                  latest              74435f89ab78        13 days ago         73.9MB
centos                  latest              831691599b88        13 days ago         215MB
gametron/minecraft      latest              47996a0df306        5 months ago        184MB
hello-world             latest              bf756fb1ae65        5 months ago        13.3kB
pranav0/pythonserver    v1                  21cb123caa5c        8 months ago        261MB
ubuntu@instance-1:~$
```

Figure 3.3: Docker images



```
FROM ubuntu:latest

RUN apt-get update && apt-get  install --no-install-recommends software-properties-common wget xauth dbus-x11 gpg-agent -y && xauth add instance
-1/unix:10  MIT-MAGIC-COOKIE-1  e641c84504cd22ea74060cb5d9f9c6fe && dbus-uuidgen > /etc/machine-id

RUN  dpkg --add-architecture i386 && wget -O - https://dl.winehq.org/wine-builds/winehq.key | apt-key add -  &&  add-apt-repository 'deb https:/
/dl.winehq.org/wine-builds/ubuntu/ focal main' && apt update -y && apt install --install-recommends winehq-stable -y &&  apt-get  remove softwa
e-properties-common wget -y && rm -rf /var/lib/apt/lists/ && apt-get autoremove -y

CMD ["wine", "/root/wine/123freesolitaire-v1200-setup.exe "]
~
~
-- INSERT --                                                                                              7,1           All
```

Figure 3.4: Dockerfile

Here you can see in figure 3.4 contain the Dockerfile used to launch the final image which contain all the necessary requirement to run the games.

3.4.2 Container

Launching the container from the image created above requires additional support to run the game. These requirements can be passed only while launching the container from the image created.
These are given below as:

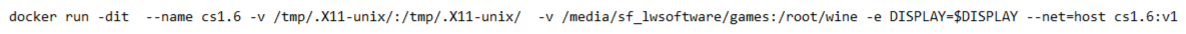
GUI: For the graphical need for games, there is the need for the GUI. GUI is transferred from the base VMs to docker.The file which is responsible for GUI has to be exported to Docker so here we create the link or tunnel through which docker get the sharing of gui from base os on which docker is running. via
-v /tmp/.X11-unix:/tmp/.X11-unix
and mount to same location.

Display: Providing GUI support alone is not enough for any graphical Apps to run, there is need to to tell docker where is to display. For that , it is required to export the Display Variable of the system where we want to Display that is done via

-e DISPLAY=localhost=:0

networking: To Connect with host network we provide one more variable

–net=host



```
docker run -dit  --name cs1.6 -v /tmp/.X11-unix/:/tmp/.X11-unix/  -v /media/sf_lwsoftware/games:/root/wine -e DISPLAY=$DISPLAY --net=host cs1.6:v1
```

Figure 3.5: Final command to run the Docker will all requirement

Chapter 4

Conclusion

After understanding the wholes sole purpose of container and vm's, the best way to manage resource for now is to deploy the container on vm. there is no doubt that vm is a good way to manage and use the resources but due to bulkyness and slow, one should use container in vm to deploy the application.

4.1 Overall Description

Time Saving: Since GIC is based on the container technology it is fast secure and reliable.

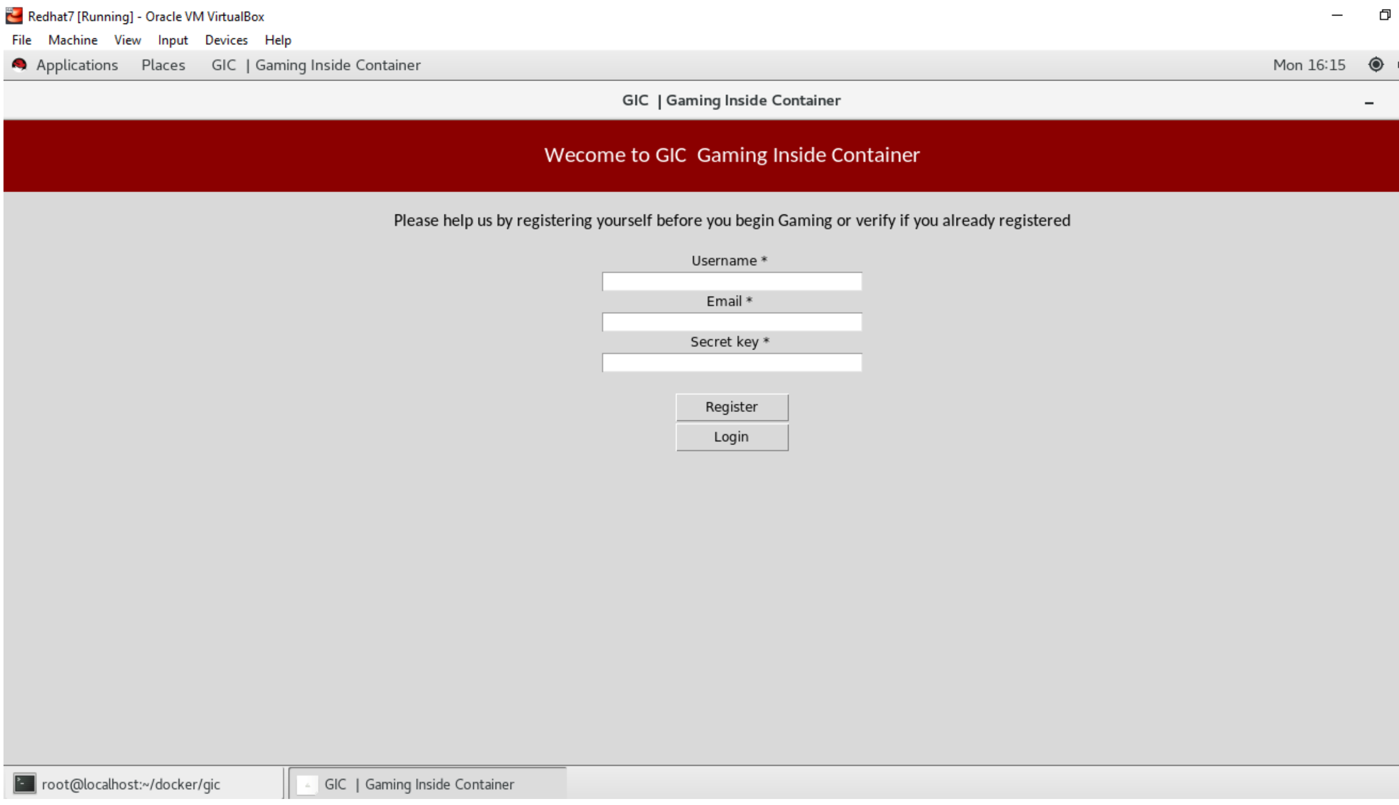
No Complication: Due to atomocity feature of container, it reolves all the dependencies problems

Cost Effective: It's cheaper. All the cloud services are on the pay-as-you go model so you pay as per the time period you use.you dont have to pay any fee for license and all.

Appendix A

Screenshot and Description of the Implemented System

A.1 First Page


Redhat7 [Running] - Oracle VM VirtualBox

GIC | Gaming Inside Container

Wecome to GIC Gaming Inside Container

Please help us by registering yourself before you begin Gaming or verify if you already registered

Username *

Email *

Secret key *

File
Machine
View
Input
Devices
Help
Applications
Places
GIC | Gaming Inside Container
Mon 16:15
Register
Login
root@localhost:~/docker/gic
GIC | Gaming Inside Container

Figure A.1: First page

Figure A.1 is the first page when you run the project.

A.2 User Login form

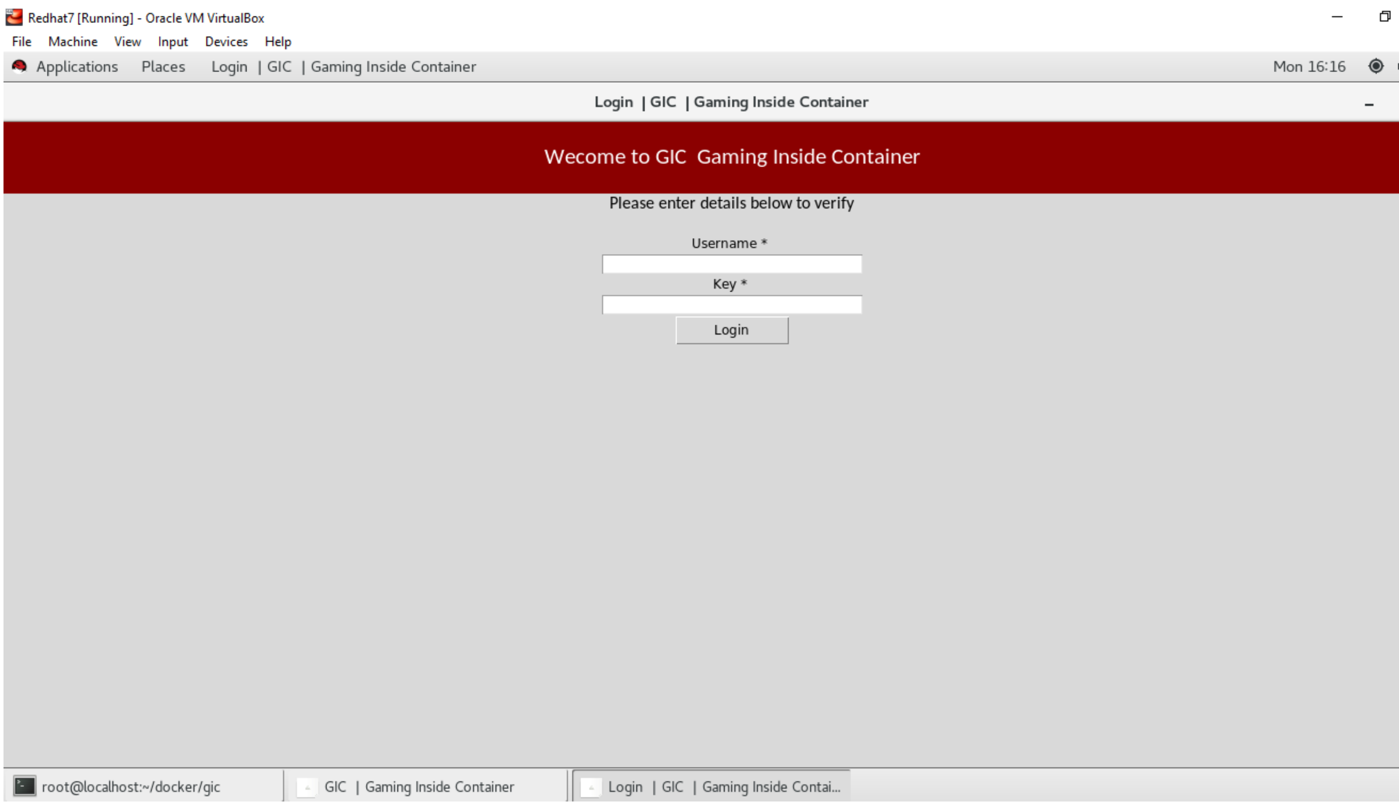

Redhat7 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places Login | GIC | Gaming Inside Container
Mon 16:16
Login | GIC | Gaming Inside Container
Wecome to GIC Gaming Inside Container
Please enter details below to verify
Username *
Key *
Login
root@localhost:~/docker/gic
GIC | Gaming Inside Container
Login | GIC | Gaming Inside Contai...

Figure A.2: Login form

Figure A.2 User have to insert there Email and Password for login in the System.

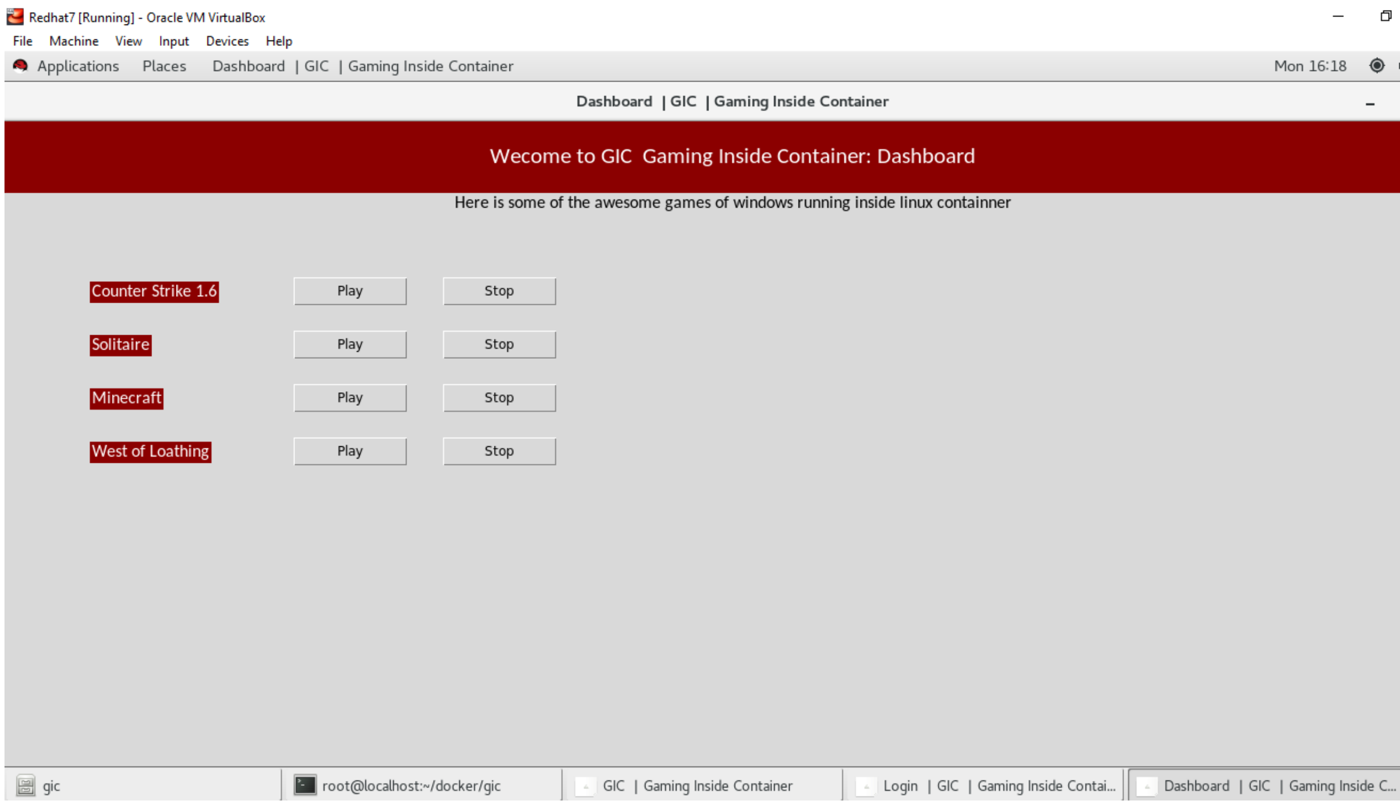
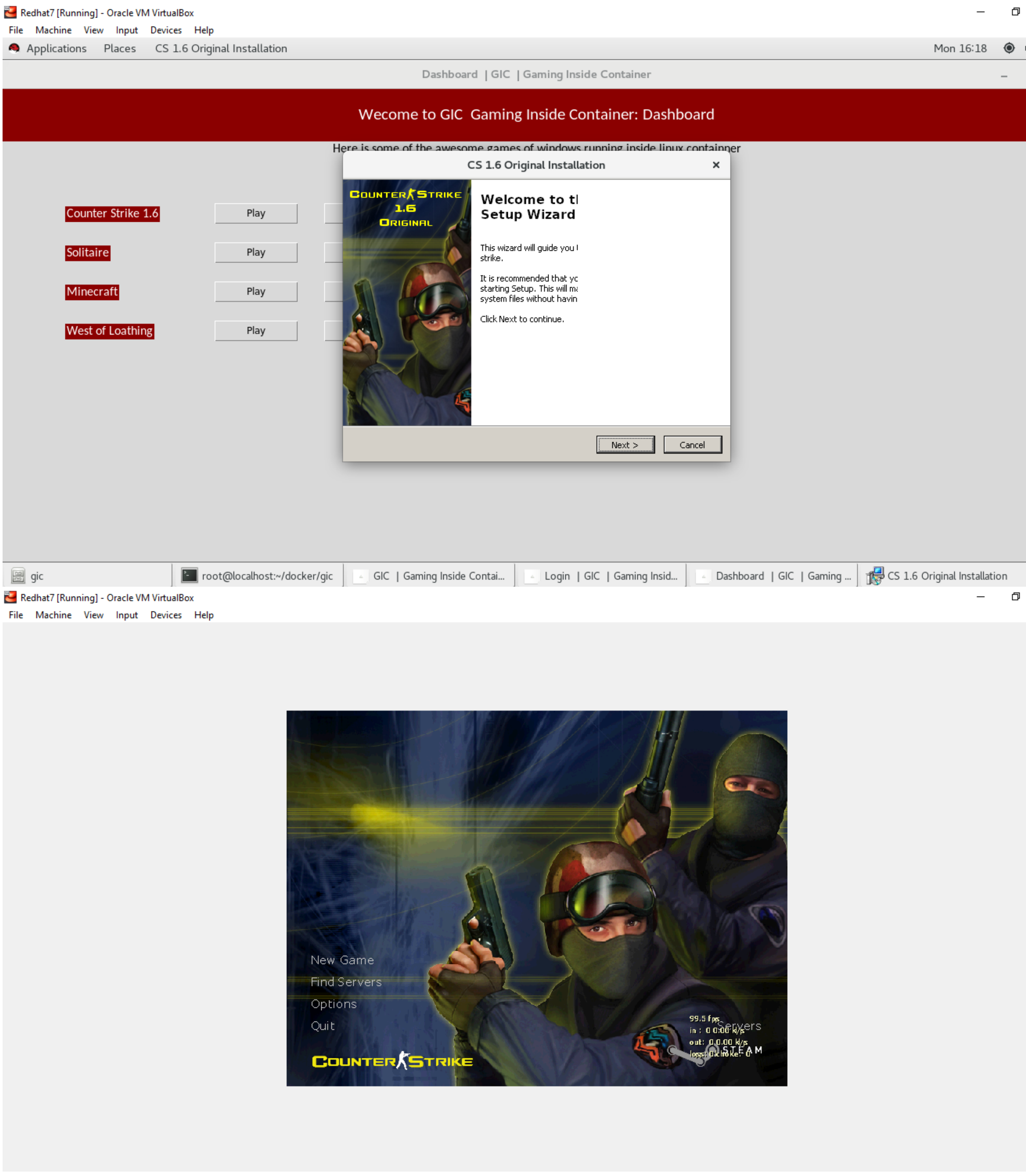

Redhat7 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places Dashboard | GIC | Gaming Inside Container
Mon 16:18
Dashboard | GIC | Gaming Inside Container
Wecome to GIC Gaming Inside Container: Dashboard
Here is some of the awesome games of windows running inside linux containner
Counter Strike 1.6
Solitaire
Minecraft
West of Loathing
Play
Stop
gic
root@localhost:~/docker/gic
GIC | Gaming Inside Container
Login | GIC | Gaming Inside Contai...
Dashboard | GIC | Gaming Inside C...

Figure A.3: Gaming Dashboard

A.3 Gaming Dashboard

Figure A.3 shows the list of all available games to play. Here to remember with one click on play, a new container will launch with and you will see the gaming console. And with other click on stop, the container will stop and then get deleted within seconds.


Redhat7 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places CS 1.6 Original Installation
Mon 16:18
Dashboard | GIC | Gaming Inside Container
Counter Strike 1.6
Solitaire
Minecraft
West of Loathing
Play
CS 1.6 Original Installation
Next >
Cancel
gic
root@localhost:~/docker/gic
GIC | Gaming Inside Contai...
Login | GIC | Gaming Insid...
Dashboard | GIC | Gaming ...
CS 1.6 Original Installation
Redhat7 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
New Game
Find Servers
Options
Quit
COUNTER-STRIKE

Wecome to GIC Gaming Inside Container: Dashboard

Here is some of the awesome games of windows running inside linux containner

Welcome to tl
Setup Wizard

This wizard will guide you t
strike.

It is recommended that yc
starting Setup. This will ma
system files without havin

Click Next to continue.

Figure A.4: These are the images when one click on play to CS1.6

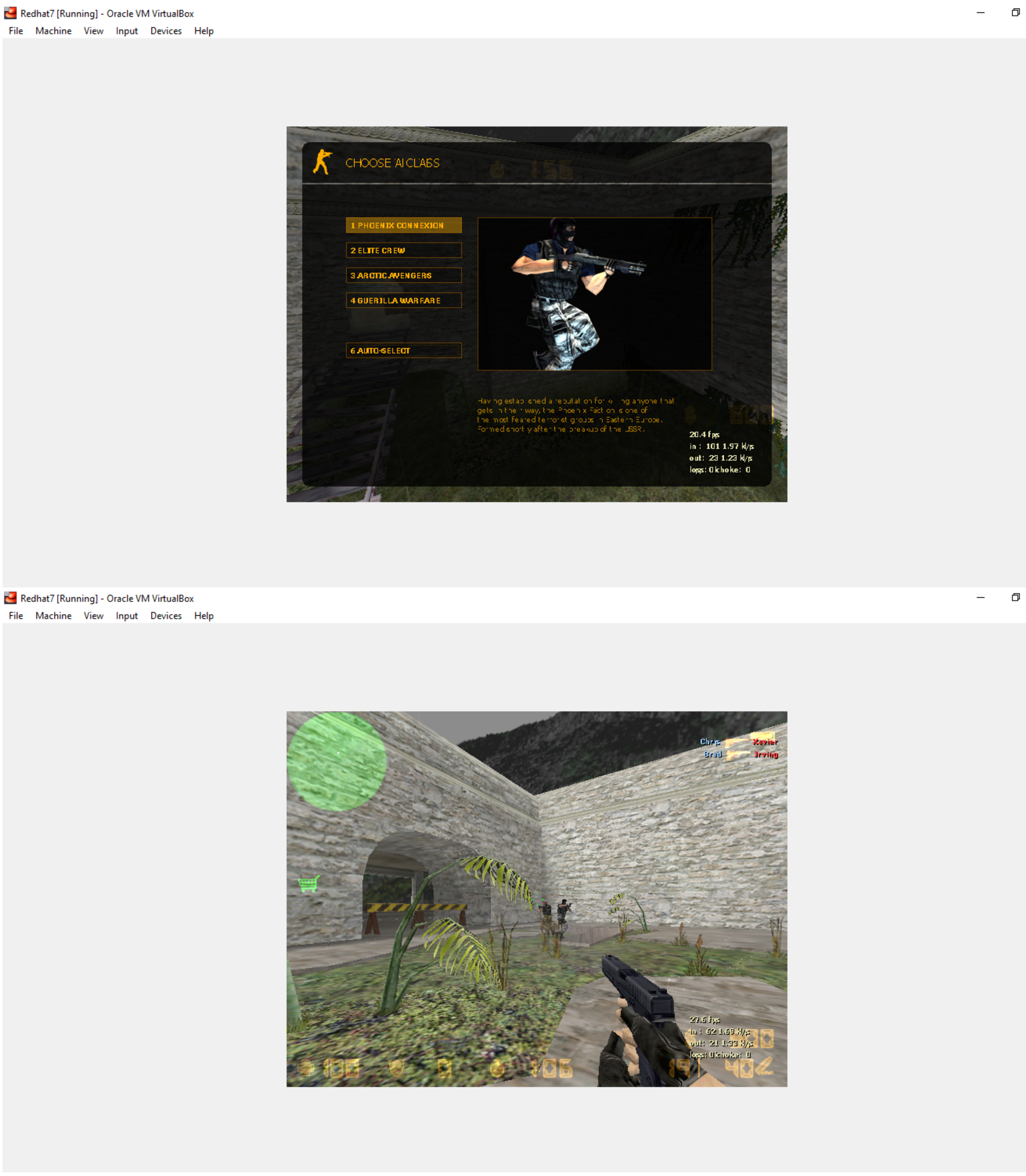
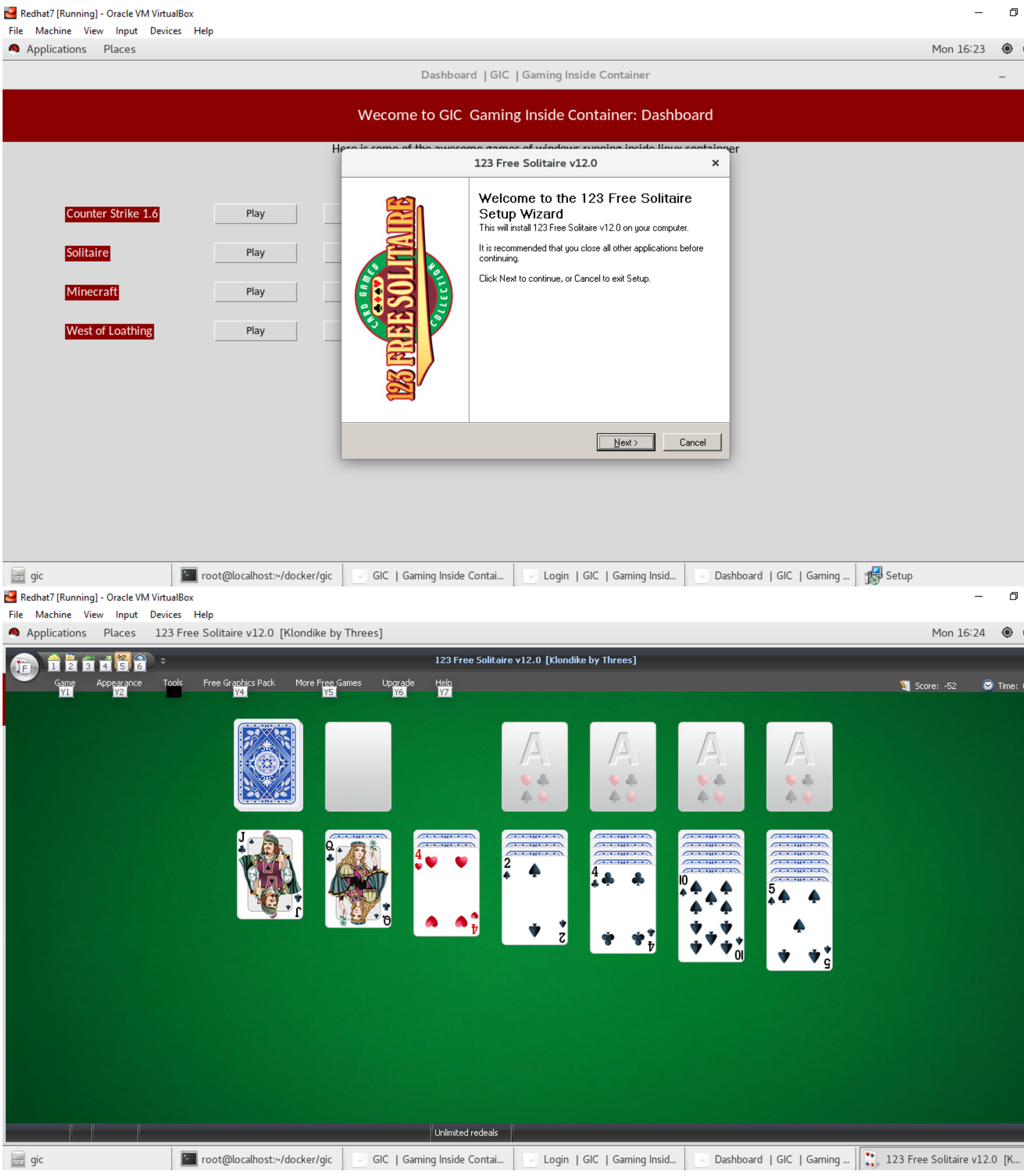

Redhat7 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
CHOOSE A CLASS
1 PHOENIX CONNEXION
2 ELITE CREW
3 ARCTIC AVENGERS
4 GUERILLA WARFARE
6 AUTO-SELECT
Redhat7 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Chris
Brad
Xavier
Irving

Figure A.5: These are the images when one click on play to CS1.6


Redhat7 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places
Mon 16:23
Dashboard | GIC | Gaming Inside Container
Wecome to GIC Gaming Inside Container: Dashboard
Counter Strike 1.6
Solitaire
Minecraft
West of Loathing
Play
123 Free Solitaire v12.0
123 FREE SOLITAIRE
CARD GAMES COLLECTION
Next >
Cancel
gic
root@localhost:~/docker/gic
GIC | Gaming Inside Contai...
Login | GIC | Gaming Insid...
Dashboard | GIC | Gaming ...
Setup
Mon 16:24
123 Free Solitaire v12.0 [Klondike by Threes]
Game
Appearance
Tools
Free Graphics Pack
More Free Games
Upgrade
Help
Score: -52
Time:
Unlimited redeals
123 Free Solitaire v12.0 [K...

Here is some of the awesome games of windows running inside linux container

Welcome to the 123 Free Solitaire Setup Wizard

This will install 123 Free Solitaire v12.0 on your computer.

It is recommended that you close all other applications before continuing.

Click Next to continue, or Cancel to exit Setup.

Figure A.6: These are the images when one click on play to solitaire

Appendix B

User manual

B.1 Introduction

The web application will be giving a very user friendly approach for all user.Records can be easily accessed and store and other information respectively.

Since all games are running on seperate container, these are the most secure and efficient way to manage and play the games.

B.2 Step to setup the server for the GIC

1. First create the main docker images by running the command **docker build -t gameset:latest .**
run this command on the working directory or main directory where you will see the dockerfile ie inside ./gic
2. Create individual images by going into the sperate folder of each game.

cd ./gic/cs1.6

docker build -t cs1.6:latest
Give tag name to each game as that of directory name, because Script uses the game as primary
3. Your setup is done. Start running App.py by
python3 App.py

Bibliography

1. Shipman, John W. (2010-12-12), Tkinter reference: a GUI for Python, New Mexico Tech Computer Center, retrieved 2012-01-11 dc
2. Cane, B (2017). Leveraging the Dockerignore file to create smaller images. Cited 18.01.2018, https://blog.codeship.com/leveraging-the-dockerignore-file-to-create-smaller-images/
3. Coleman, M. (2016). Containers and VMs together. Cited 18.01.2018, https://blog.docker.com/20
and-vms-together/