ACKNOWLEDGEMENT

I would like to express my gratitude towards my University as well as to my teacher for providing me the golden opportunity to do this wonderful Project regarding DevOps Which helped me in doing a lot of homework and learning, As a result, came to know about so many new concepts So, I am really thank full to them.

Moreover, I would like to thank my friends who helped me a lot whenever I got stuck in some problem related to my course I am really thankful to have such good support from them as they always have my back whenever I need it.

Also, I would like to mention the support system and consideration of my parents who have always been there in my life to make me choose the right thing and oppose the wrong. Without them, I could never have learned and become the person I am now.

I have taken an effort in this project, however, it would not have been possible without the support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them.

Table of Content

Introduction

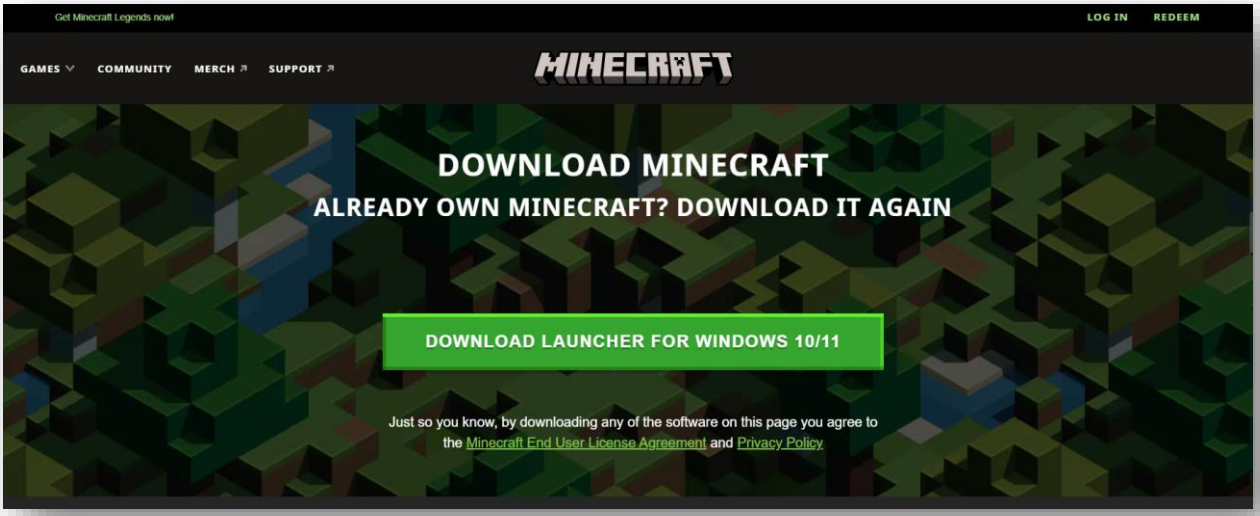
DevOps is a set of practices that combines software development and IT operations to shorten the systems development life cycle, improve deployment frequency, and ensure reliability and scalability of software applications. DevOps emphasizes collaboration, automation, and monitoring across development and operations teams to deliver high-quality software products efficiently.

One technology that has become increasingly popular in the DevOps space is Docker, a platform that allows developers to package their applications and dependencies into portable containers that can be deployed anywhere. Docker simplifies the deployment process by providing a consistent and reliable environment for running applications, and it also enables teams to easily scale their applications up or down as needed.

Minecraft, a popular sandbox video game, can be used with Docker to create and manage Minecraft servers. With Docker, players can easily set up and configure their own Minecraft servers, customize the server settings, and manage the server resources. Docker allows players to easily deploy and manage multiple Minecraft servers on a single machine or across multiple machines, which can be particularly useful for running large Minecraft communities or hosting gaming events. By leveraging Docker's containerization technology, Minecraft players can enjoy a more streamlined and efficient gaming experience.

DOCKER + MINECRAFT

Step 1 – Install Minecraft from official Site -



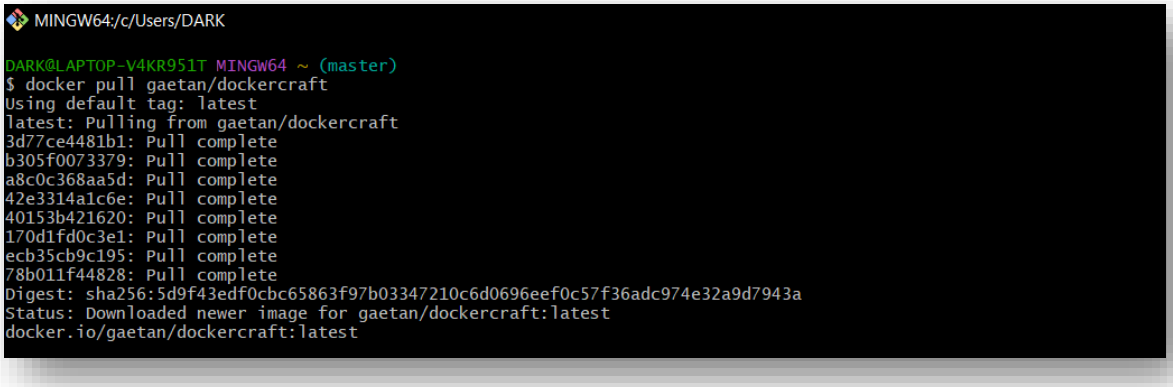
Step 2 - Pull or build Docker craft image –

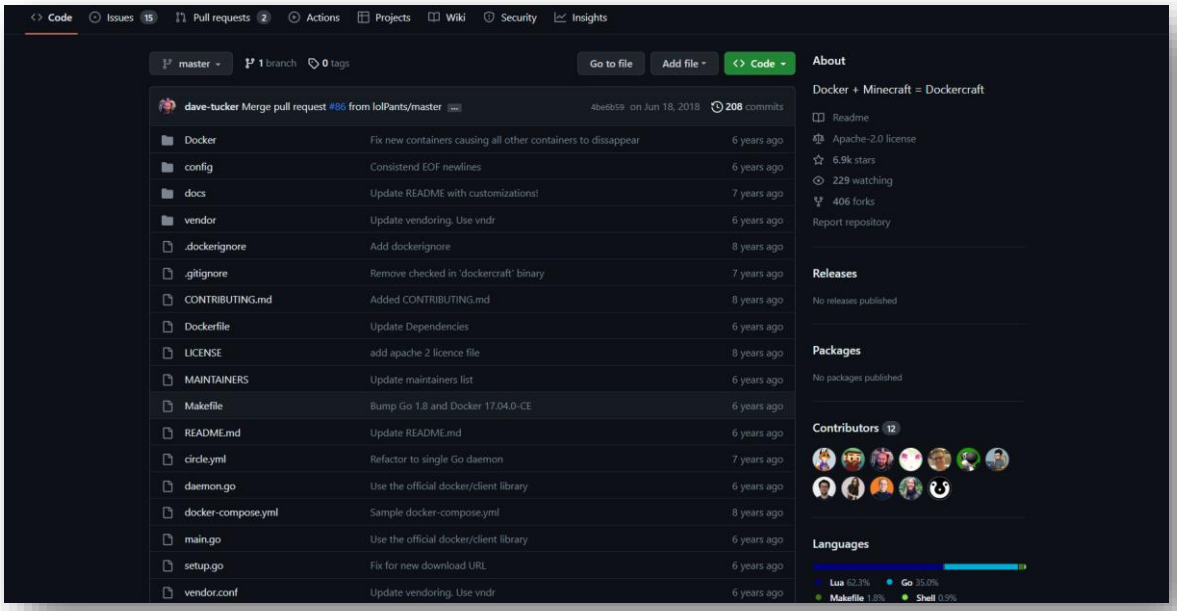
- docker pull gaetan/dockercraft

  or

- git clone git@github.com:docker/dockercraft.git
  docker build -t gaetan/dockercraft dockercraft

- Now the repo is cloned and can be used inside a container





Step 3 - Run Dockercraft container –

- The image will create a container which we will configure according to out need the port which we want to provide

- **Command –**
  docker run -t -i -d -p 25565:25565 \
  -v /var/run/docker.sock:/var/run/docker.sock \
  --name dockercraft \
  gaetan/dockercraft

  - Mounting (/var/run/docker.sock) inside the container is necessary to send requests to the Docker remote API.
  - The default port for a Minecraft server is 25565, if you prefer a different one: -p <port>:25565

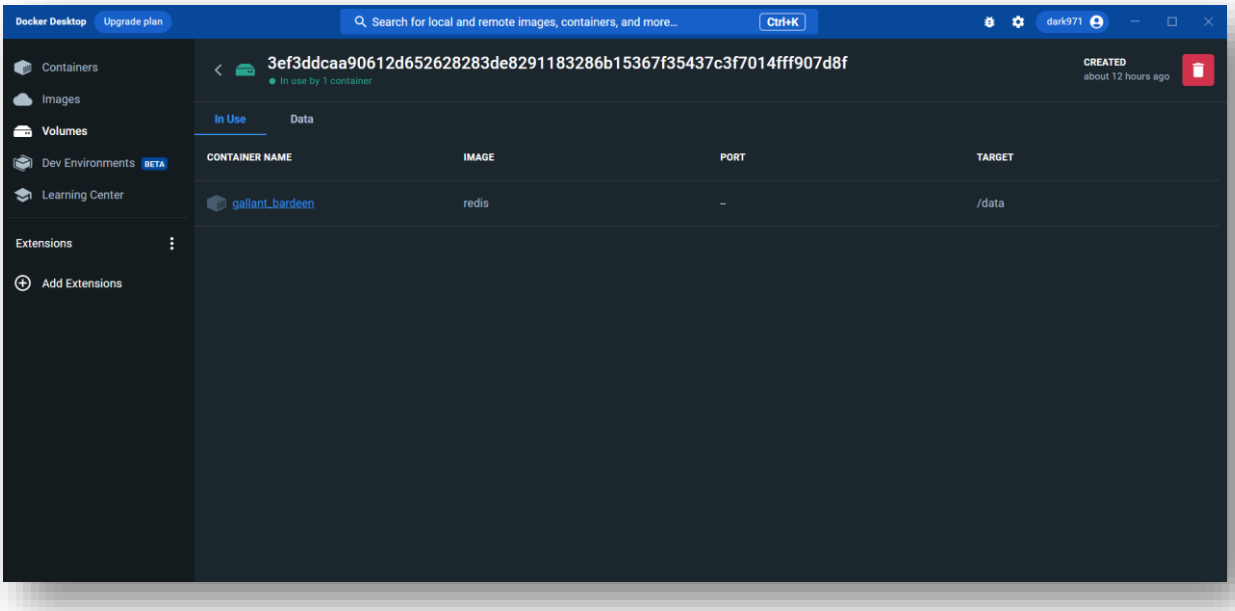
```
DARK@LAPTOP-V4KR951T MINGW64 ~ (master)
$ docker run -t -i -d -p 25565:25565 \
> -v /var/run/docker.sock/var/run/docker.sock \
> --name dockercraft \
> gaetan/dockercraft
fe00cc5db8ee284b3aec0c8074b15792a9ad7110348d07bed1df820a04fb93fd
```

- Now the Container is in running state with the port specified

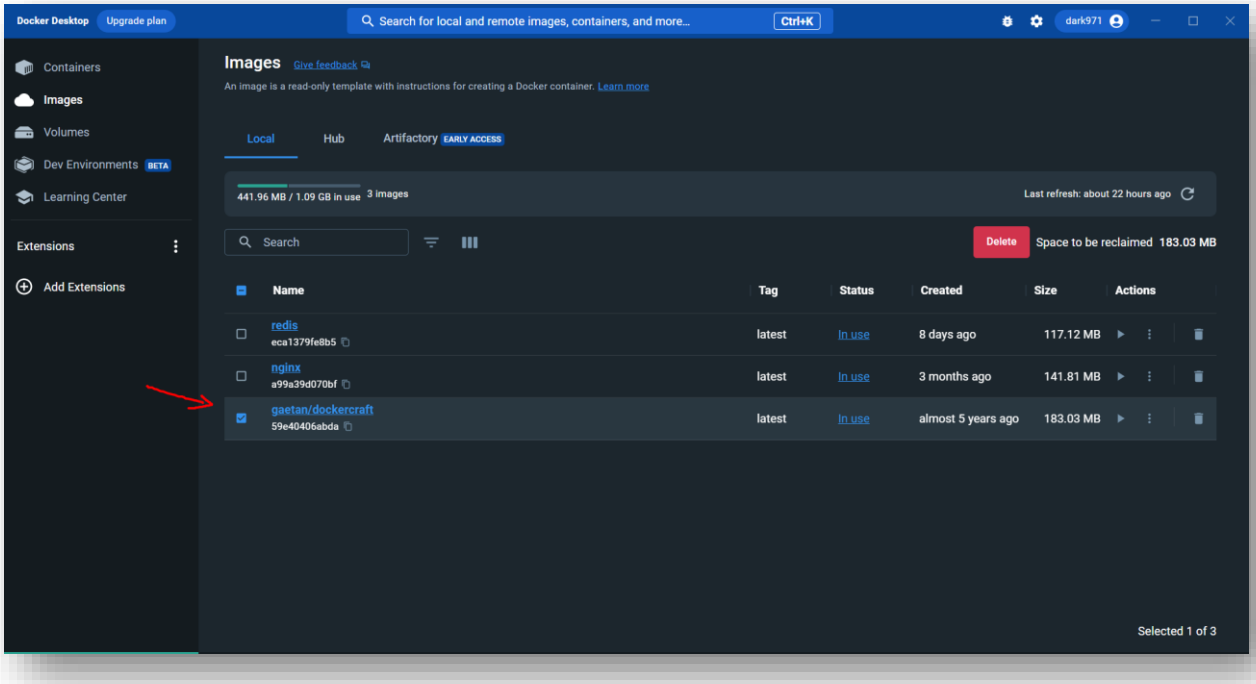
Docker Desktop ->

1. Volume -

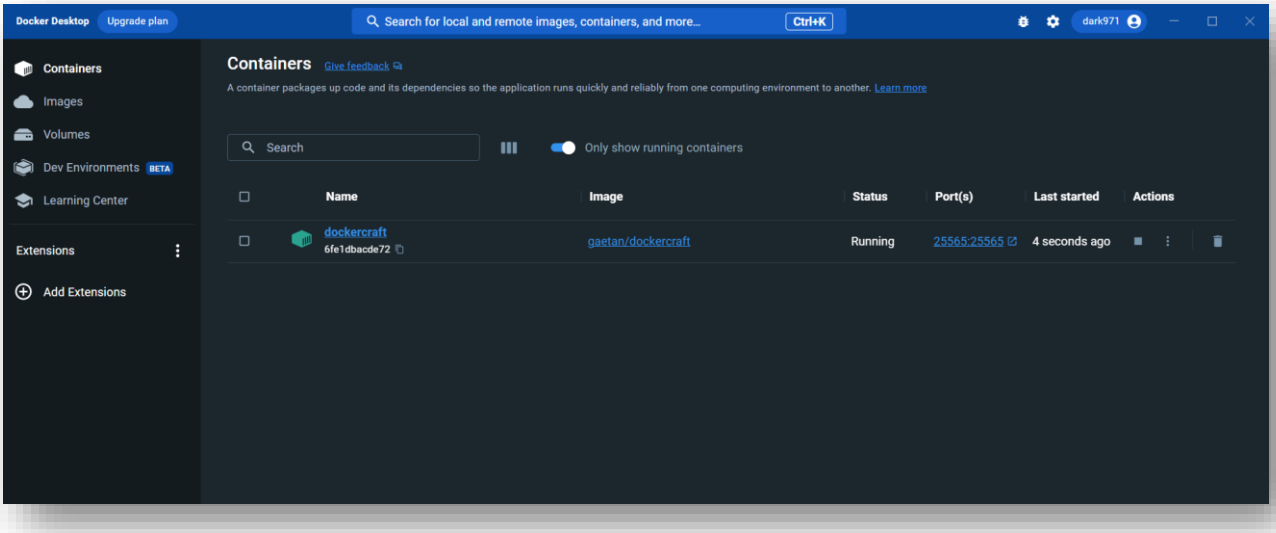
- The New volume is created to store the server data



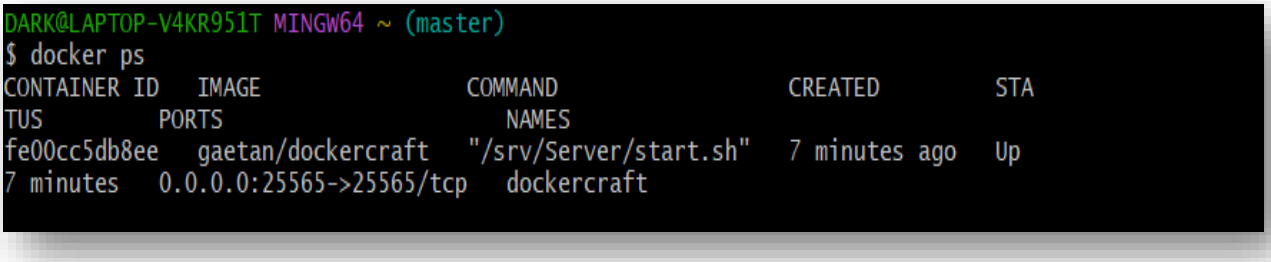
2. Docker Image -



3. Container -



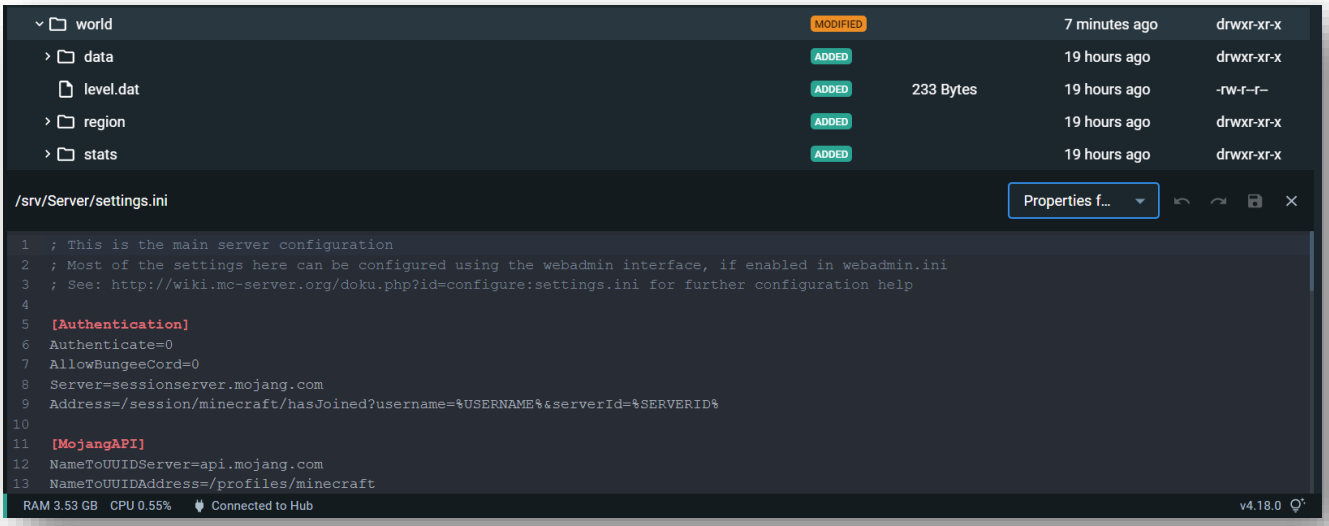
- **Container running Ready to use on Minecraft server**



Configuration ->

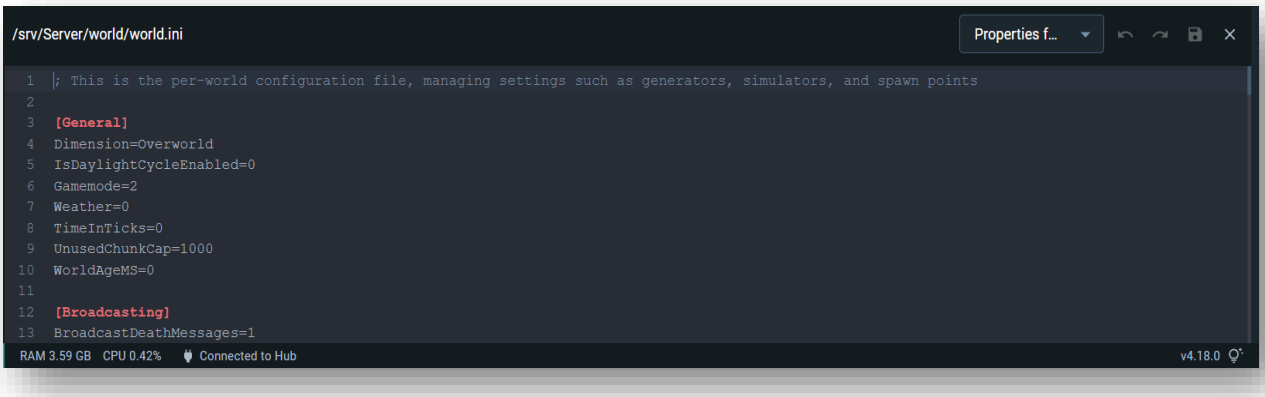
- The authentication and other several changes can be made as per the client requirements

    - Anti Cheat
    - Authentication
    - Change in Player Data

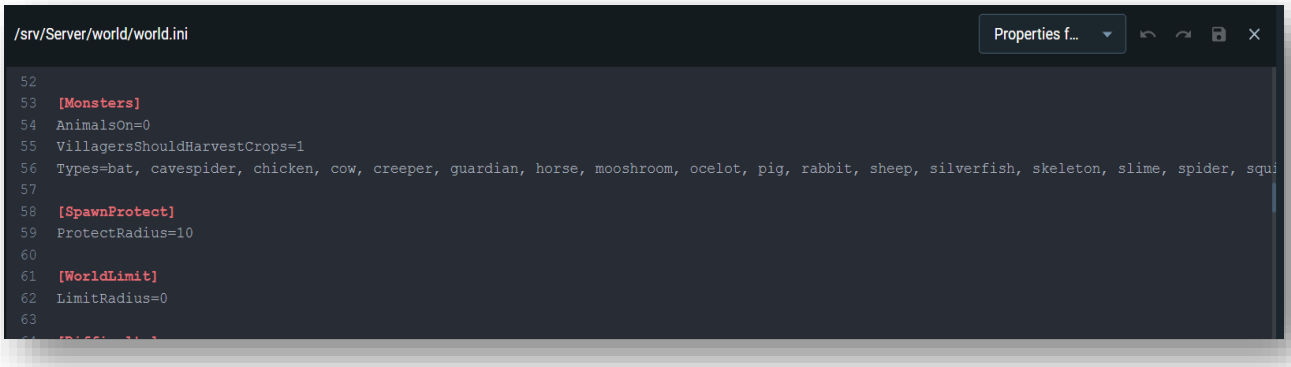


World Generation ->

- Change to world can be made by provider end to maintain the server
    - Game mode
    - View distance
    - Change in sea.
    - Adding or changing the types of mobs



- Change in game Difficulty

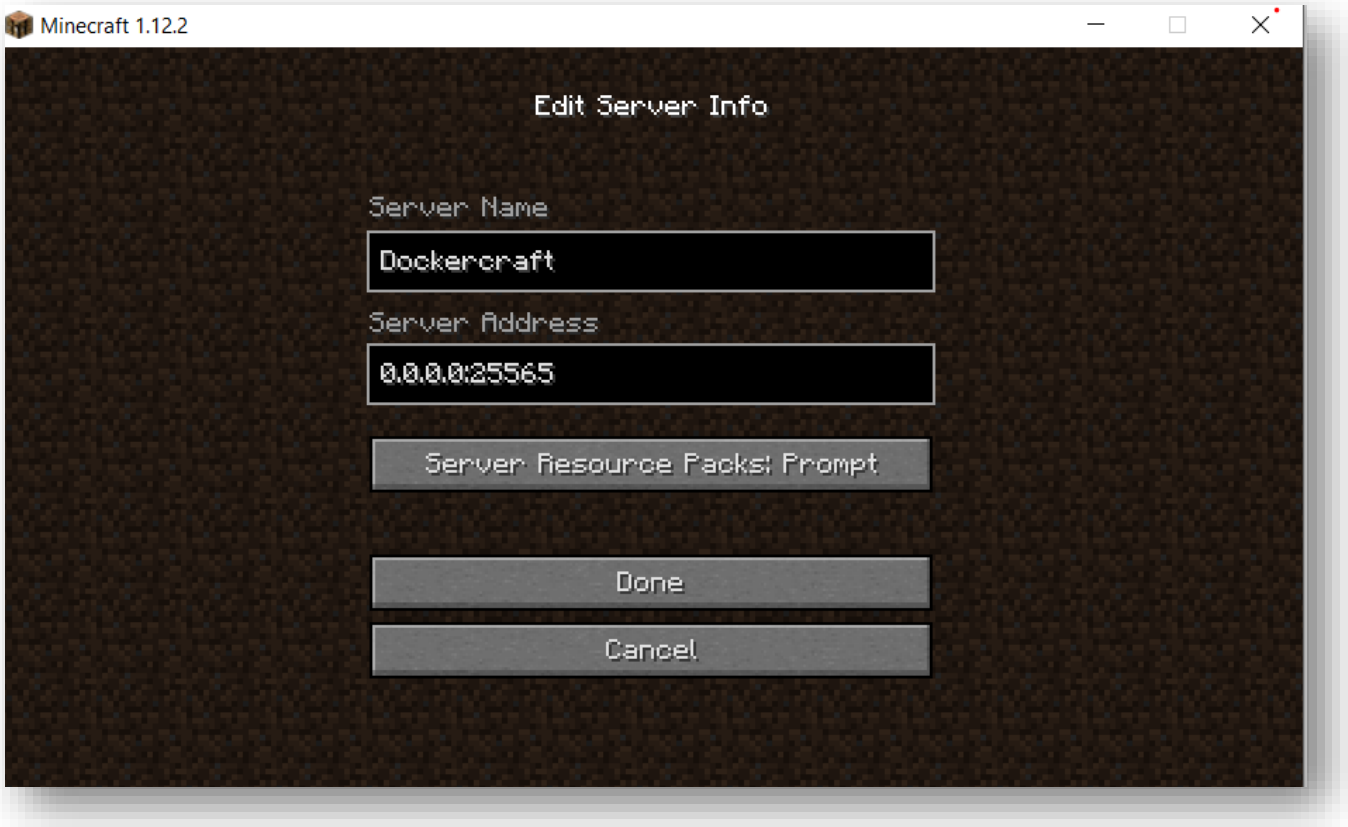

/srv/Server/world/world.ini
Properties f...
52
53 [Monsters]
54 AnimalsOn=0
55 VillagersShouldHarvestCrops=1
56 Types=bat, cavespider, chicken, cow, creeper, guardian, horse, mooshroom, ocelot, pig, rabbit, sheep, silverfish, skeleton, slime, spider, squi
57
58 [SpawnProtect]
59 ProtectRadius=10
60
61 [WorldLimit]
62 LimitRadius=0
63

Now everything is configured and ready to launch the server

1. Open Minecraft > Multiplayer > Add Server

- The server address is the IP of Docker host. No need to specify a port if you used the default one.
- In our case Ip address is – 0.0.0.0:25565

- Here Name of server can be provided and the server Address



Step 4 - Join **Server!**

- You should see at least one container in your world, which is the one hosting your Dockercraft server.
- You can start, stop and remove containers interacting with levers and buttons. Some Docker commands are also supported directly via Minecraft's chat window, which is displayed by pressing the T key (default) or / key.

- Here you can see the name of our image
- CPU usage
- Memory usage

- The container can be removed from here too and the server will turn off

- This button in game can be used to start of stop the containers according to the need

Creating a Demo Container

- The container can be made while being inside the game using (/docker run redis)

```
C:\Users\DARK>docker run -d --name hello_redis redis
901e38d93b01ec2ea5872d1446859814a069f5cdbea908956593cea4c2797a0d
```

```
C:\Users\DARK>docker ps
CONTAINER ID   IMAGE              COMMAND                  CREATED             STATUS             PORTS                     NAMES
901e38d93b01   redis              "docker-entrypoint.s…"   About a minute ago  Up About a minute  6379/tcp                  hello_redis
9db8c2e8eeb7   redis              "docker-entrypoint.s…"   3 minutes ago       Up 3 minutes       6379/tcp                  gallant_bardeen
6fe1dbacde72   gaetan/dockercraft "/srv/Server/start.sh"   7 hours ago         Up 10 minutes      0.0.0.0:25565->25565/tcp  dockercraft
```

- As the snipt show the container name same this can be removed and start and stop

- Admin can perform all this function on the player he will be managing

Summary of Report

DevOps is a set of practices that combines software development and IT operations to deliver high-quality software products efficiently. Docker, a platform for packaging and deploying applications, is popular in the DevOps space for its ability to simplify deployment and scaling. Minecraft, a popular video game, can be used with Docker to easily set up and manage Minecraft servers, making it ideal for running large gaming communities or hosting events. By leveraging Docker's containerization technology, Minecraft players can enjoy a streamlined and efficient gaming experience.

In this report I have discussed in detail my project, its working, making, features, and application, I have explained each and every step of making an Minecraft server using docker and applying DevOps fundamentals.
This report highlights all the process involved in making serial order from using several features to create and manage the game servers
I have also attached the preview of the steps performed