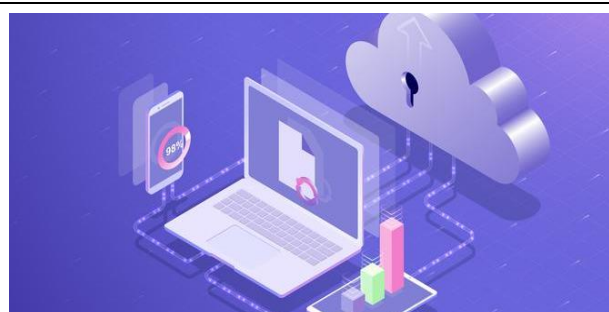




Universidad de Guadalajara  
CUCEI – Centro Universitario de Ciencias Exactas e  
Ingenierías  
Mtro. Michel Emanuel López Franco  
Computación tolerante a fallas

# Reporte 15



Velasco Hernandez Victor  
Manuel

Código: 216598879

| D06 | 2022A |

| 02/Mayo/2022 |

Docker, Kubernetes, Apache Mesos,  
OpenShift, Rancher y Istio





Universidad de Guadalajara  
CUCEI  
Computación tolerante a fallas

Victor Manuel Velasco Hernández  
|Código: 216598879 | | 2022A |  
|02/ Mayo /2022 | |D06|



## Reporte 15-Docker, Kubernetes, Apache Mesos, OpenShift, Rancher y Istio

### Objetivo:

Contestar preguntas y generar ejemplo.

**Diferencias entre Docker, Kubernetes, Apache Mesos, OpenShift, Rancher y Istio y que aplicación le podrías dar.**

Herramienta	Diferencias	Aplicación
Docker	<ul style="list-style-type: none"><li>• Conteneriza aplicaciones.</li><li>• No escalable.</li><li>• Alta tolerancia a fallos.</li></ul>	Es importante separar un proyecto de software por imágenes, para que en caso de que se necesite solo una parte de este, no se cargue completamente, como lo sería el sistema login de una pagina web.
Kubernetes	<ul style="list-style-type: none"><li>• Es un software administrador de contenedores.</li><li>• Auto-escalable.</li><li>• Baja tolerancia a fallos.</li><li>• Es open-source.</li></ul>	En el desarrollo de una pagina web con inicios de sesión, es importante que los cluster sean replicados en caso de que los contenedores hayan alcanzado su máxima cantidad de despliegues, ya sea por la gran afluencia de usuarios.
Apache Mesos	<ul style="list-style-type: none"><li>• Es un administrador de Cluster.</li><li>• Provee aplicaciones con APIs para el manejo de recursos y scheduling.</li><li>• Usa Workflows de contenedores.</li></ul>	En caso de que sea necesario que se desplieguen aplicaciones cada cierto tiempo, ya sea para que las aplicaciones estén deshabilitadas y sean habilitadas dependiendo la planificación, es



		importante agendar esta herramienta.
OpenShift	<ul style="list-style-type: none"><li>• Es un software administrador de contenedores.</li><li>• Es un producto open-source de RedHat.</li><li>• Incluye Docker y Kubernetes.</li><li>• Ofrece el servicio de tecnologías de productividad y seguridad.</li></ul>	El desarrollo de servicios y aplicaciones en gobierno en el cuál es posible visualizar los beneficios con alta fidelidad.
Rancher	<ul style="list-style-type: none"><li>• Tecnología que administra clusters de Kubernetes en masa.</li><li>• Simplifica las operaciones complejas de Kubernetes mientras mantiene su flexibilidad.</li></ul>	En caso de que se quiere eficientemente el computo distribuido, es importante que se de una correcta gestión de los recursos que disponga.
Istio	<ul style="list-style-type: none"><li>• Capa de servicios configurables que conecta, supervisa y protege contenedores en un clúster Kubernetes,</li><li>• Gestiona el tráfico.</li></ul>	En caso de querer desplegar una tienda online, se pueden supervisar todos los servicios con los que se cuentan.

[1]–[8]

### ¿Que es K3D?

Es un ligero empaquetador para correr K3S, es el equivalente a Rancher con Kubernetes, estos se encuentran dentro de contenedores Docker, cuando se despliega K3D se despliegan dos contenedores, rancher y kubernetes(k3s). [9]

### ¿Qué es K3S?

Creado por Rancher, es una distribución de Kubernetes reducida, se aloja dentro de un contenedor Docker, pensada para funcionar en equipos de limitados recursos, quita los componentes innecesarios para proyectos de pequeña escala en Kubernetes, siendo inclusivo con procesadores ARM y los tradicionales de 64 bits, útil para internet de las cosas. [10]



## Introducción:

El uso de herramientas de manejo de clúster que complementan a Kubernetes brinda la posibilidad de tener sistemas de computo o clúster capaces de manejar una gran cantidad de procesos de manera planificada e interconectada, lo cuál hace que sea posible la realización de procesos cada vez más complejos con mayor seguridad, lo cuál resulta ser una excelente herramienta para lograr una mejora en el manejo de operaciones realizadas dentro de un clúster.

## Desarrollo:

En esta actividad se va a desplegar un servidor de Minecraft 1.12.2, con ayuda de Rancher contenido en K3D y Kubernetes (K3S), todo esto dentro de contenedores Docker, por lo cuál, es importante tener instalado K3S, K3D y Docker, basados en el tutorial "[Running Minecraft on Kubernetes with Rancher K3s - YouTube](#)". [11]

Comando para instalar K3S:

`"curl -sL https://get.k3s.io | sh -"`

Comando para intalar K3D:

`"curl -s https://raw.githubusercontent.com/k3d-io/k3d/main/install.sh | bash"`

```
root@kali: ~  
File Actions Edit View Help  
root@kali)~  
# curl -sL https://get.k3s.io | sh -  
[INFO] Finding release for channel stable  
[INFO] Using v1.22.7+k3s1 as release  
[INFO] Downloading hash https://github.com/k3s-io/k3s/releases/download/v1.22.7+k3s1/sha256sum-amd64.txt  
[INFO] Downloading binary https://github.com/k3s-io/k3s/releases/download/v1.22.7+k3s1/k3s  
[INFO] Verifying binary download  
[INFO] Installing k3s to /usr/local/bin/k3s  
[INFO] Skipping installation of SELinux RPM  
[INFO] Skipping /usr/local/bin/kubectrl symlink to k3s, already exists  
[INFO] Creating /usr/local/bin/crictl symlink to k3s  
[INFO] Skipping /usr/local/bin/ctr symlink to k3s, command exists in PATH  
at /usr/bin/ctr  
[INFO] Creating killall script /usr/local/bin/k3s-killall.sh  
[INFO] Creating uninstall script /usr/local/bin/k3s-uninstall.sh  
[INFO] env: Creating environment file /etc/systemd/system/k3s.service.env  
[INFO] systemd: Creating service file /etc/systemd/system/k3s.service  
[INFO] systemd: Enabling k3s unit  
Created symlink /etc/systemd/system/multi-user.target.wants/k3s.service → /etc/systemd/system/k3s.service.  
[INFO] systemd: Starting k3s  
root@kali)~  
# curl -s https://raw.githubusercontent.com/k3d-io/k3d/main/install.sh | bash  
Preparing to install k3d into /usr/local/bin  
k3d installed into /usr/local/bin/k3d  
Run 'k3d --help' to see what you can do with it.
```



**Universidad de Guadalajara**  
**CUCEI**  
**Computación tolerante a fallas**

**Victor Manuel Velasco Hernández**  
**| Código: 216598879 | | 2022A |**  
**| 02/ Mayo /2022 | | D06 |**

Comenzamos creando el cluster de Rancher, el cuál empaquetara nuestra versión K3S de Kubernetes, con el comando “*k3d cluster create minecraft*”.

```
root@kali: ~  
File Actions Edit View Help  
root@kali)~  
# k3d cluster create minecraft  
INFO[0000] Prep: Network  
INFO[0000] Created network 'k3d-minecraft'  
INFO[0000] Created image volume k3d-minecraft-images  
INFO[0000] Starting new tools node...  
INFO[0001] Creating node 'k3d-minecraft-server-0'  
INFO[0002] Pulling image 'ghcr.io/k3d-io/k3d-tools:5.4.1'  
INFO[0003] Pulling image 'docker.io/rancher/k3s:v1.22.7-k3s1'  
INFO[0008] Starting Node 'k3d-minecraft-tools'  
INFO[0019] Creating LoadBalancer 'k3d-minecraft-serverlb'  
INFO[0020] Pulling image 'ghcr.io/k3d-io/k3d-proxy:5.4.1'  
INFO[0055] Using the k3d-tools node to gather environment information  
INFO[0056] HostIP: using network gateway 172.18.0.1 address  
INFO[0056] Starting cluster 'minecraft'  
INFO[0056] Starting servers...  
INFO[0057] Starting Node 'k3d-minecraft-server-0'  
INFO[0071] All agents already running.  
INFO[0071] Starting helpers...  
INFO[0073] Starting Node 'k3d-minecraft-serverlb'  
INFO[0082] Injecting records for hostAliases (incl. host.k3d.internal) and for 2 network members into CoreDNS configmap ...  
INFO[0086] Cluster 'minecraft' created successfully!  
INFO[0087] You can now use it like this:  
kubectl cluster-info  
  
root@kali)~  
# docker ps  
CONTAINER ID   IMAGE                                     COMMAND                                     CREATED        STATUS        PORTS  
d98b71089e9e   ghcr.io/k3d-io/k3d-proxy:5.4.1          "/bin/sh -c nginx-pr..."              43 seconds ago Up 20 seconds  80/tcp, 0.0.0.0:45727->6  
443/tcp       k3d-minecraft-serverlb  
113a360cd847   rancher/k3s:v1.22.7-k3s1              "/bin/k3d-entrypoint..."              About a minute ago Up 38 seconds  
k3d-minecraft-server-0
```

Podremos contemplar con el comando “*docker ps*” que tras haber creado el cluster dentro de Docker se generaron dos contenedores, uno para rancher y otro para Kubernetes(KS3), como podremos ver con el comando “*kubectl get nodes*” dentro del clúster está el correspondiente pod para Minecraft.

```
root@kali: ~  
File Actions Edit View Help  
root@kali)~  
# docker ps  
CONTAINER ID   IMAGE                                     COMMAND                                     CREATED        STATUS        PORTS  
d98b71089e9e   ghcr.io/k3d-io/k3d-proxy:5.4.1          "/bin/sh -c nginx-pr..."              43 seconds ago Up 20 seconds  80/tcp, 0.0.0.0:45727->6  
443/tcp       k3d-minecraft-serverlb  
113a360cd847   rancher/k3s:v1.22.7-k3s1              "/bin/k3d-entrypoint..."              About a minute ago Up 38 seconds  
k3d-minecraft-server-0  
  
root@kali)~  
# kubectl cluster-info  
Kubernetes control plane is running at https://0.0.0.0:45727  
CoreDNS is running at https://0.0.0.0:45727/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy  
Metrics-server is running at https://0.0.0.0:45727/api/v1/namespaces/kube-system/services/https:metrics-server:https/proxy  
  
To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.  
  
root@kali)~  
# kubectl get nodes  
NAME                STATUS    ROLES    AGE     VERSION  
k3d-minecraft-server-0 Ready     control-plane,master  6m30s   v1.22.7+k3s1
```



**Universidad de Guadalajara**  
**CUCEI**  
**Computación tolerante a fallas**

**Victor Manuel Velasco Hernández**  
**|Código: 216598879 | | 2022A |**  
**|02/ Mayo /2022 | |D06|**

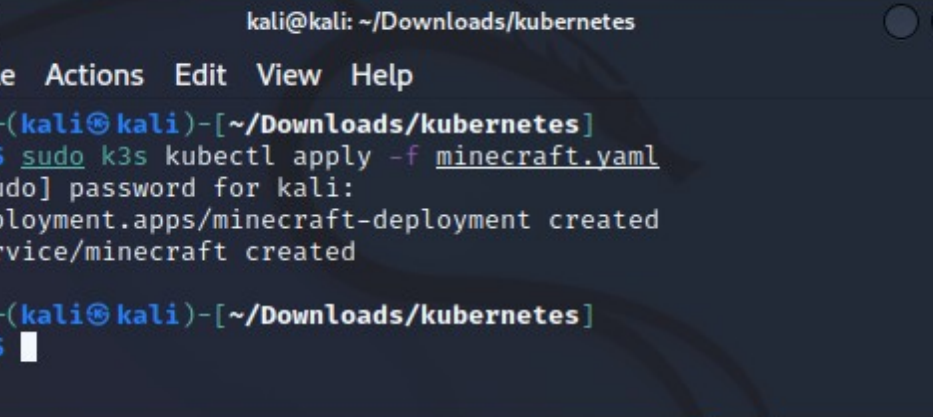
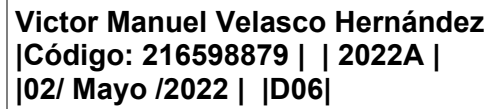
Para poder empezar a desplegar nuestro servidor, necesitamos generar un archivo YAML para darle la configuración al clúster, en este archivo, declararemos un contenedor con una imagen llamada “hashicraft/minecraft:v1.12.2” y dentro de su entorno, un link con el mapa que se desea incluir:

```
! minecraft.yaml x
home > kali > Downloads > kubernetes > ! minecraft.yaml
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: minecraft-deployment
5    labels:
6      app: minecraft
7  spec:
8    replicas: 1
9    selector:
10     matchLabels:
11       app: minecraft
12  template:
13    metadata:
14      labels:
15        app: minecraft
16    spec:
17      containers:
18        - name: minecraft
19          image: hashicraft/minecraft:v1.12.2
20          ports:
21            - containerPort: 25565
22          env:
23            - name: WORLD_BACKUP
24              value: https://github.com/nicholasjackson/hashicraft/releases/download/v0.0.0/world2.tar.gz
25            - name: WHITELIST_ENABLED
26              value: false
```

Otra parte importante, es la declaración dentro del mismo archivo, el servicio para la conexión del contenedor del mapa de Minecraft, se declaran los puertos a utilizar:

```
! minecraft.yaml x
home > kali > Downloads > kubernetes > ! minecraft.yaml
24  value: https://github.com/nicholasjackson/hashicraft/releases/download/v0.0.0/world2.tar.gz
25  - name: WHITELIST_ENABLED
26    value: "false"
27
28  ---
29  apiVersion: v1
30  kind: Service
31  metadata:
32    labels:
33      app: minecraft
34    name: minecraft
35  spec:
36    ports:
37      - name: minecraft
38        nodePort: 30001
39        port: 25565
40        protocol: TCP
41        targetPort: 25565
42    selector:
43      app: minecraft
44    type: NodePort
```





```
kali@kali: ~/Downloads/kubernetes
File Actions Edit View Help
(kali@kali)-[~/Downloads/kubernetes]
$ sudo k3s kubectl apply -f minecraft.yaml
[sudo] password for kali:
deployment.apps/minecraft-deployment created
service/minecraft created
(kali@kali)-[~/Downloads/kubernetes]
$
```

```
(root@kali)~# k3s kubectcl get pods
NAME                                READY   STATUS
minicraft-deployment-59759b64c9-hxz92  1/1     Running
0                                     111s

(root@kali)~# k3s kubectcl logs
error: expected 'logs [-f] [-p] (POD | TYPE/NAME) [-c CONTAINER]'.
POD or TYPE/NAME is a required argument for the command
See 'kubectcl logs -h' for help and examples

(root@kali)~# k3s kubectcl logs minicraft-deployment-59759b64c9-hxz92
ls: cannot access '/minicraft/world': No such file or directory
Installing default world https://github.com/nicholasjackson/hashicraft/releases/download/v0.0.0/world2.tar.gz
--2022-04-30 06:38:23-- https://github.com/nicholasjackson/hashicraft/releases/download/v0.0.0/world2.tar.gz
Resolving github.com (github.com)... 140.82.114.4
Connecting to github.com (github.com)[140.82.114.4]:443... connected.
HTTP request sent, expecting response... 302 Found
https://objects.githubusercontent.com/github-production-release-asset-2a65be/254305888/6fe88980-7b1d-11ea-8c8f-ac5c96d23dcd?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWNJYAX4CSVEH53A2F20220430%2Fus-east-1%2F%3F2Faws4_request&X-Amz-Date=20220430T063828Z&X-Amz-Expires=300&X-Amz-Signature=8b3db64393d71cc19538a5757b79de6445880b2b5faaba82da253f65b1be826X-Amz-SignedHeaders=host&actor_id=0&key_id=0&repo_id=254305888&response-content-disposition=attachment%3B%20filename%3Dworld2.tar.gz&response-content-type=application%2Foctet-stream [following]
--2022-04-30 06:38:24-- https://objects.githubusercontent.com/github-production-release-asset-2a65be/254305888/6fe88980-7b1d-11ea-8c8f-ac5c96d23dcd?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWNJYAX4CSVEH53A2F20220430%2Fus-east-1%2F%3F2Faws4_request&X-Amz-Date=20220430T063828Z&X-Amz-Expires=300&X-Amz-Signature=8b3db64393d71cc19538a5757b79de6445880b2b5faaba82da253f65b1be826X-Amz-SignedHeaders=host&actor_id=0&key_id=0&repo_id=254305888&response-content-disposition=attachment%3B%20filename%3Dworld2.tar.gz&response-content-type=application%2Foctet-stream
Connecting to objects.githubusercontent.com (objects.githubusercontent.com)... 185.199.111.133, 185.199.109.133, ...
HTTP request sent, giving response... 200 OK
Length: 9315421 (8.9M) [application/octet-stream]
```



Universidad de Guadalajara  
CUCEI  
Computación tolerante a fallas

Victor Manuel Velasco Hernández  
|Código: 216598879 | | 2022A |  
|02/ Mayo /2022 | |D06|

Solo que a estas alturas, hay un pequeño defecto, el clúster no está vinculado a los puertos declarados, por lo que procederemos a borrarlo con “*k3d cluster delete minecraft*” y crearemos otro con “*k3d cluster create minecraft -p 25565:30001*”, ya con los puertos correctos:

```
kali@kali: ~/Downloads/kubernetes
File Actions Edit View Help
(kali@kali)~[~/Downloads/kubernetes]
$ sudo k3d cluster delete minecraft
INFO[0001] Deleting cluster 'minecraft'
INFO[0004] Deleting cluster network 'k3d-minecraft'
INFO[0004] Deleting 2 attached volumes...
WARN[0004] Failed to delete volume 'k3d-minecraft-images' of cluster 'minecraft': failed to find volume 'k3d-minecraft-images': Error: No such volume: k3d-minecraft-images → Try to delete it manually
INFO[0004] Removing cluster details from default kubeconfig...
INFO[0004] Removing standalone kubeconfig file (if there is one)...
INFO[0004] Successfully deleted cluster minecraft!

(kali@kali)~[~/Downloads/kubernetes]
$ sudo k3d create cluster minecraft -p 25565:30001
Error: unknown command "create" for "k3d"
Run 'k3d --help' for usage.
FATA[0000] unknown command "create" for "k3d"

(kali@kali)~[~/Downloads/kubernetes]
$ sudo k3d cluster create minecraft -p 25565:30001
INFO[0000] portmapping '25565:30001' lacks a nodefilter, but there's more than one node: defaulting to [servers:*:proxy agents:*:proxy]
INFO[0000] Prep: Network
INFO[0000] Created network 'k3d-minecraft'
INFO[0000] Created image volume k3d-minecraft-images
INFO[0000] Starting new tools node...
INFO[0001] Starting Node 'k3d-minecraft-tools'
INFO[0001] Creating node 'k3d-minecraft-server-0'
INFO[0001] Creating LoadBalancer 'k3d-minecraft-serverlb'
INFO[0002] Using the k3d-tools node to gather environment information
INFO[0003] HostIP: using network gateway 172.19.0.1 address
INFO[0003] Starting cluster 'minecraft'
INFO[0003] Starting servers...
INFO[0004] Starting Node 'k3d-minecraft-server-0'
INFO[0016] All agents already running.
INFO[0016] Starting helpers...
```

Y nuevamente desplegaremos nuestro archivo YAML, con el comando “*k3s kubectl apply -f minecraft.yaml*” dentro de su carpeta raíz:

```
kali@kali: ~/Downloads/kubernetes
File Actions Edit View Help
(kali@kali)~[~/Downloads/kubernetes]
$ sudo k3s kubectl apply -f minecraft.yaml
[sudo] password for kali:
deployment.apps/minecraft-deployment created
service/minecraft created

(kali@kali)~[~/Downloads/kubernetes]
$
```





Universidad de Guadalajara  
CUCEI  
Computación tolerante a fallas

Victor Manuel Velasco Hernández  
|Código: 216598879 | | 2022A |  
|02/ Mayo /2022 | |D06|

Podremos observar con el comando “*k3s kubectl get pods*” que el proyecto nuevamente ha sido desplegado, lo ultimo, es ver la IP del servicio para podernos conectar en Minecraft, esto con el comando “*k3s kubectl get svc*”:

```
kali@kali: ~/Downloads/kubernetes
File Actions Edit View Help
(kali@kali)-[~/Downloads/kubernetes]
$ sudo k3s kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
minecraft-deployment-59759b64c9-hxz92 1/1     Running   0           8m59s

(kali@kali)-[~/Downloads/kubernetes]
$ sudo k3s kubectl get svc
NAME          TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
kubernetes    ClusterIP   10.43.0.1     <none>         443/TCP          65m
minecraft     NodePort    10.43.130.76 <none>         25565:30001/TCP  9m30s
```

Luego procedemos a abrir Minecraft, y seleccionamos “Multiplayer”:





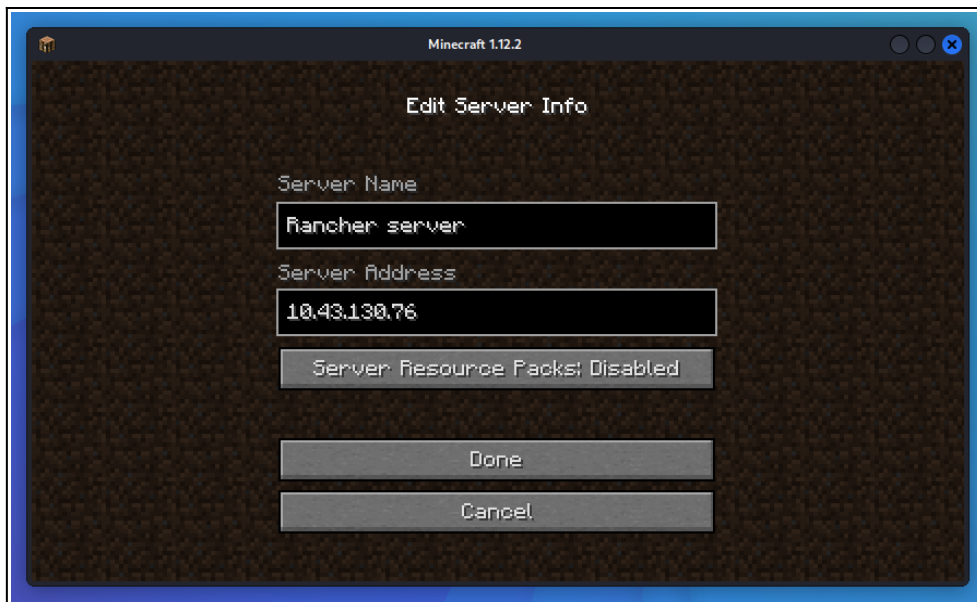
**Universidad de Guadalajara**  
**CUCEI**  
**Computación tolerante a fallas**

**Victor Manuel Velasco Hernández**  
**|Código: 216598879 | | 2022A |**  
**|02/ Mayo /2022 | |D06|**

Nos aparecerá este apartado, seleccionaremos “Add Server”:



Después, procederemos a introducir la IP del servidor:

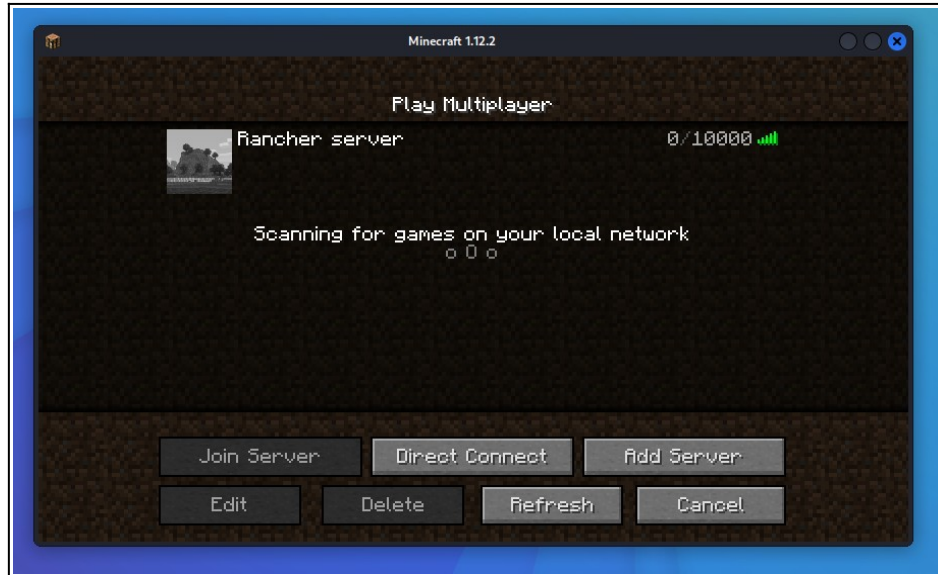




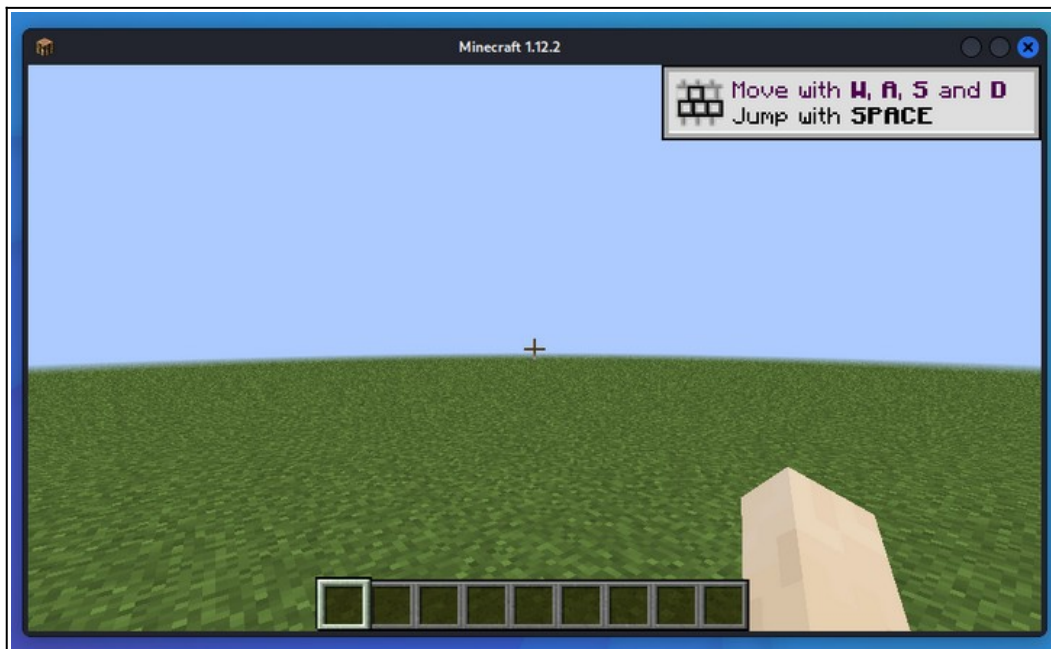
Universidad de Guadalajara  
CUCEI  
Computación tolerante a fallas

Victor Manuel Velasco Hernández  
|Código: 216598879 | | 2022A |  
|02/ Mayo /2022 | |D06|

Y al guardarlo, finalmente, nos aparecerá la capacidad del servidor, lo seleccionamos para poder entrar:



Y con esto, tendríamos listo nuestro servidor de Minecraft:





### **Conclusiones:**

El uso de herramientas para la gestión de clúster resulta ser muy útil para desplegar y replicar los contenedores necesarios dependiendo la demanda de los usuarios, además de que es útil preservando el código y la integridad de la información del clúster, esto hace un sistema tolerante a fallos, por lo que resulta ser fundamental conocer la naturaleza de este servicios para desarrollar cada vez, mejores funciones que garantizan el mejor acceso a la información de los programas, y la presencia de ejecución sin tenerlo en el equipo, lo cuál ayuda a tener una mejor comprensión de la gestión de los clúster, esto hace que podamos evitar posibles problemas de computo distribuido en el futuro.

### **Link de repositorio:**

<https://github.com/Victor012396/ComputacionTolerante.git>

### **Bibliografía:**

- [1] nishanil, “¿Qué es Docker?” <https://docs.microsoft.com/es-es/dotnet/architecture/microservices/container-docker-introduction/docker-defined> (consultado el 18 de marzo de 2022).
- [2] K. Chandrakant, “Mesos vs. Kubernetes | Baeldung”, el 27 de agosto de 2019. <https://www.baeldung.com/ops/mesos-kubernetes-comparison> (consultado el 26 de abril de 2022).
- [3] “¿Qué es Kubernetes? | Microsoft Azure”. <https://azure.microsoft.com/es-es/topic/what-is-kubernetes/> (consultado el 2 de abril de 2022).
- [4] “OpenShift vs. Kubernetes: What’s the Difference?”, el 16 de septiembre de 2021. <https://www.ibm.com/cloud/blog/openshift-vs-kubernetes> (consultado el 26 de abril de 2022).
- [5] “Kubernetes vs. Rancher: The differences all devs should know”, TheServerSide.com. <https://www.theserverside.com/answer/Kubernetes-vs-Rancher-The-differences-all-devs-should-know> (consultado el 28 de abril de 2022).
- [6] sixe, “Todo lo que necesita saber sobre Rancher: gestión de Kubernetes para empresas”, SiXe Ingeniería, el 4 de mayo de 2021. <https://sixe.es/noticias/suse-rancher-kubernetes-toda-la-informacion> (consultado el 2 de abril de 2022).
- [7] “¿Qué es Istio?” <https://www.redhat.com/es/topics/microservices/what-is->



**Universidad de Guadalajara**  
**CUCEI**  
**Computación tolerante a fallas**

**Victor Manuel Velasco Hernández**  
**|Código: 216598879 | | 2022A |**  
**|02/ Mayo /2022 | |D06|**

[istio](#) (consultado el 28 de abril de 2022).

- [8] “Istio”, Istio. <https://istio.io/latest/> (consultado el 19 de abril de 2022).
- [9] “k3d”. <https://k3d.io/v5.4.1/> (consultado el 30 de abril de 2022).
- [10]“K3s: Lightweight Kubernetes”. <https://k3s.io/?msclkid=b457f925c84611ec871691bb55a68a96> (consultado el 30 de abril de 2022).
- [11]HashiCraft, Running Minecraft on Kubernetes with Rancher K3s, (el 29 de mayo de 2020). Consultado: el 30 de abril de 2022. [En línea Video]. Disponible en: <https://www.youtube.com/watch?v=qeEhv8SLQGQ>