

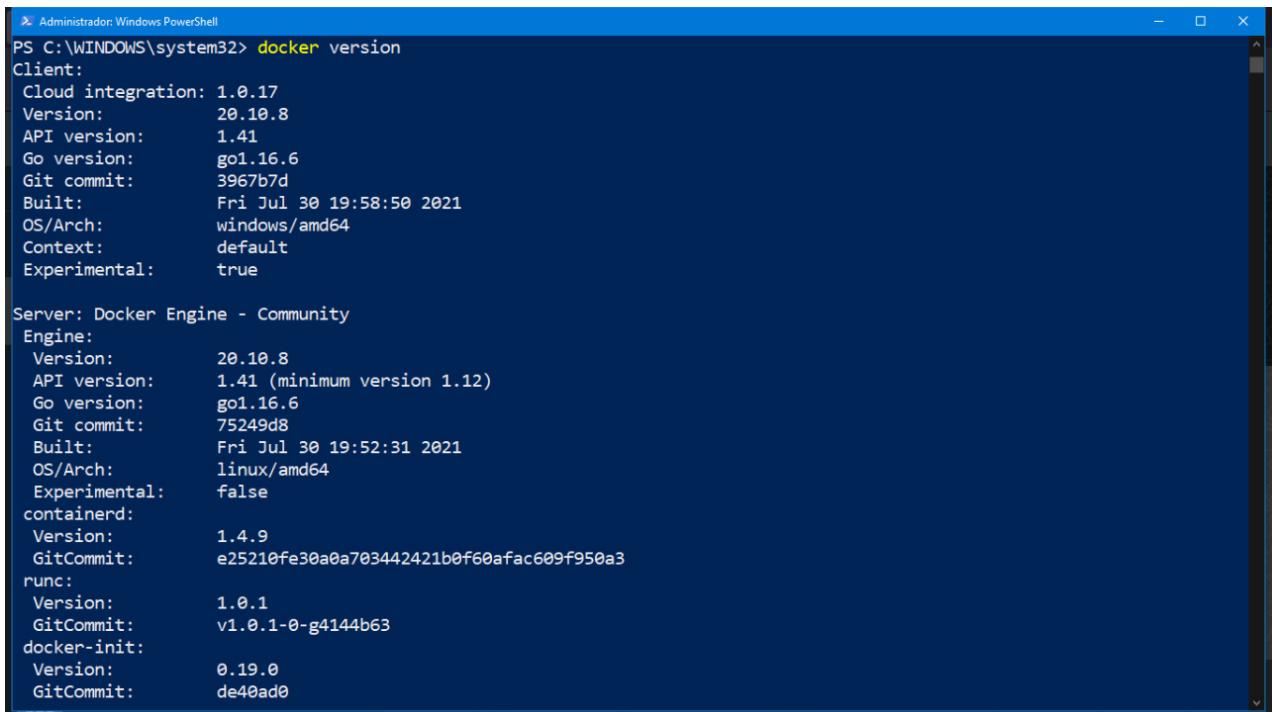
Cloud Computing:

Microarea 1

Integrantes:

- Joe Andre Acuna Torres
- Bryan Joseph Diaz Varillas

1. Check your docker version:

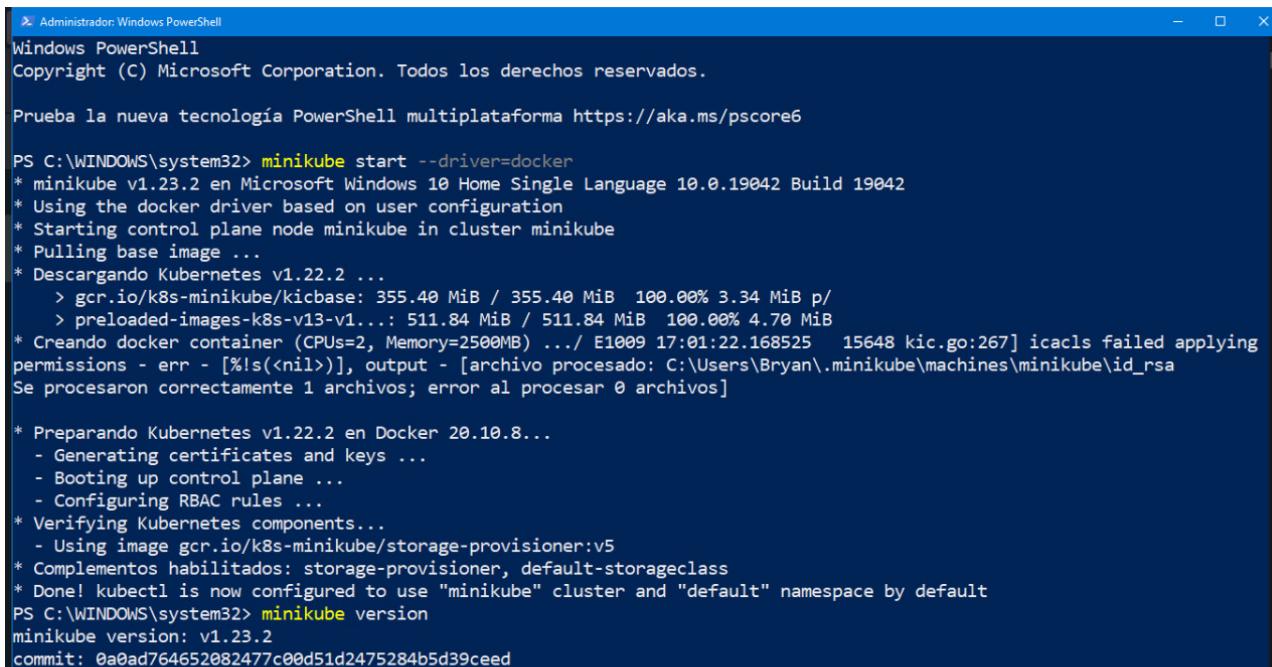


```
PS C:\WINDOWS\system32> docker version
Client:
  Cloud integration: 1.0.17
  Version:          20.10.8
  API version:      1.41
  Go version:       go1.16.6
  Git commit:       3967b7d
  Built:            Fri Jul 30 19:58:50 2021
  OS/Arch:          windows/amd64
  Context:          default
  Experimental:    true

Server: Docker Engine - Community
Engine:
  Version:          20.10.8
  API version:     1.41 (minimum version 1.12)
  Go version:       go1.16.6
  Git commit:       75249d8
  Built:            Fri Jul 30 19:52:31 2021
  OS/Arch:          linux/amd64
  Experimental:    false
containerd:
  Version:          1.4.9
  GitCommit:        e25210fe30a0a703442421b0f60afac609f950a3
runc:
  Version:          1.0.1
  GitCommit:        v1.0.1-0-g4144b63
docker-init:
  Version:          0.19.0
  GitCommit:        de40ad0
```

La versión del docker es la 20.10.8

2. Install Minikube (alternative you could try Kind, microk8s, or other).



```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS C:\WINDOWS\system32> minikube start --driver=docker
* minikube v1.23.2 en Microsoft Windows 10 Home Single Language 10.0.19042 Build 19042
* Using the docker driver based on user configuration
* Starting control plane node minikube in cluster minikube
* Pulling base image ...
* Descargando Kubernetes v1.22.2 ...
  > gcr.io/k8s-minikube/kicbase: 355.40 MiB / 355.40 MiB 100.00% 3.34 MiB p/
  > preloaded-images-k8s-v13-v1...: 511.84 MiB / 511.84 MiB 100.00% 4.70 MiB
* Creando docker container (CPUs=2, Memory=2500MB) .../ E1009 17:01:22.168525 15648 kic.go:267] icacls failed applying permissions - err - [%!s(<nil>)], output - [archivo procesado: C:\Users\Bryan\.minikube\machines\minikube\id_rsa
Se procesaron correctamente 1 archivos; error al procesar 0 archivos]

* Preparando Kubernetes v1.22.2 en Docker 20.10.8...
  - Generating certificates and keys ...
  - Booting up control plane ...
  - Configuring RBAC rules ...
* Verifying Kubernetes components...
  - Using image gcr.io/k8s-minikube/storage-provisioner:v5
* Complementos habilitados: storage-provisioner, default-storageclass
* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
PS C:\WINDOWS\system32> minikube version
minikube version: v1.23.2
commit: 0a0ad764652082477c00d51d2475284b5d39ceed
```

Ya está instalado el minikube es por eso que ahora toca inicializarlo con el driver de docker

a. Single-Node

```
PS C:\WINDOWS\system32> minikube node list
minikube      192.168.49.2
PS C:\WINDOWS\system32> kubectl cluster-info
Kubernetes control plane is running at https://127.0.0.1:65204
CoreDNS is running at https://127.0.0.1:65204/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
PS C:\WINDOWS\system32> kubectl get nodes
NAME      STATUS    ROLES     AGE   VERSION
minikube  Ready     control-plane,master  17m   v1.22.2
```

Un nodo único viene a ser solo el nodo master.

b. multi-node (1master, 2 workers)

```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> minikube start --nodes 3 -p multinode-demo
* [multinode-demo] minikube v1.23.2 en Microsoft Windows 10 Home Single Language 10.0.19042 Build 19042
* Controlador docker seleccionado automáticamente. Otras opciones: hyperv, virtualbox, ssh
* Starting control plane node multinode-demo in cluster multinode-demo
* Pulling base image ...
* Creando docker container (CPUs=2, Memory=2200MB) ...| E1009 17:31:59.451085    3196 kic.go:267] icacls failed applying permissions - err - [%!s(<nil>)], output - [archivo procesado: C:\Users\Bryan\.minikube\machines\multinode-demo\id_rsa
Se procesaron correctamente 1 archivos; error al procesar 0 archivos]

* Preparando Kubernetes v1.22.2 en Docker 20.10.8...
- Generating certificates and keys ...
- Booting up control plane ...
- Configuring RBAC rules ...
* Configurando CNI CNI ...
* Verifying Kubernetes components...
- Using image gcr.io/k8s-minikube/storage-provisioner:v5
* Complementos habilitados: storage-provisioner, default-storageclass

* Starting node multinode-demo-m02 in cluster multinode-demo
* Pulling base image ...
* Creando docker container (CPUs=2, Memory=2200MB) ...| E1009 17:33:18.935611    3196 kic.go:267] icacls failed applying permissions - err - [%!s(<nil>)], output - [archivo procesado: C:\Users\Bryan\.minikube\machines\multinode-demo-m02\id_rsa
Se procesaron correctamente 1 archivos; error al procesar 0 archivos]

* Se han encontrado las siguientes opciones de red:
- NO_PROXY=192.168.58.2
- no_proxy=192.168.58.2
* Preparando Kubernetes v1.22.2 en Docker 20.10.8...
- env NO_PROXY=192.168.58.2
* Verifying Kubernetes components...

* Starting node multinode-demo-m03 in cluster multinode-demo
* Pulling base image ...
* Creando docker container (CPUs=2, Memory=2200MB) ...| E1009 17:34:59.255180    3196 kic.go:267] icacls failed applying permissions - err - [%!s(<nil>)], output - [archivo procesado: C:\Users\Bryan\.minikube\machines\multinode-demo-m03\id_rsa
Se procesaron correctamente 1 archivos; error al procesar 0 archivos]

* Se han encontrado las siguientes opciones de red:
- NO_PROXY=192.168.58.2,192.168.58.3
- no_proxy=192.168.58.2,192.168.58.3
* Preparando Kubernetes v1.22.2 en Docker 20.10.8...
- env NO_PROXY=192.168.58.2
- env NO_PROXY=192.168.58.2,192.168.58.3
* Verifying Kubernetes components...
* Done! kubectl is now configured to use "multinode-demo" cluster and "default" namespace by default
```

Aquí estamos inicializando el minikube con 3 nodos (multi-node)

```
Administrator: Windows PowerShell
- env NO_PROXY=192.168.58.2
- env NO_PROXY=192.168.58.2,192.168.58.3
* Verifying Kubernetes components...
* Done! kubectl is now configured to use "multinode-demo" cluster and "default" namespace by default
PS C:\WINDOWS\system32> kubectl get nodes
NAME           STATUS   ROLES      AGE     VERSION
multinode-demo   Ready    control-plane,master  6m2s   v1.22.2
multinode-demo-m02  Ready    <none>    4m44s   v1.22.2
multinode-demo-m03  Ready    <none>    2m29s   v1.22.2
PS C:\WINDOWS\system32> minikube status -p multinode-demo
! Executing "docker container inspect multinode-demo --format={{.State.Status}}" took an unusually long time: 2.9806619s
* Restarting the docker service may improve performance.
multinode-demo
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured

multinode-demo-m02
type: Worker
host: Running
kubelet: Running

multinode-demo-m03
type: Worker
host: Running
kubelet: Running
```

Con estos comandos podemos comprobar que hay un nodo maestro (Node Master) y dos nodos trabajadores (Node Worker)

3. Use a simple Pod to test your environment

```
deployment.yaml: Bloc de notas
Archivo Edición Formato Ver Ayuda
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  selector:
    matchLabels:
      app: nginx
  replicas: 2 # tells deployment to run 2 pods matching the template
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: nginx:1.14.2
          ports:
            - containerPort: 80
```

Código yaml del deployment de nginx

```
C:\Users\Bryan
λ kubectl describe deployment nginx-deployment
Name:           nginx-deployment
Namespace:      default
CreationTimestamp:  Sun, 10 Oct 2021 10:42:48 -0500
Labels:          <none>
Annotations:    deployment.kubernetes.io/revision: 1
Selector:        app=nginx
Replicas:       2 desired | 2 updated | 2 total | 2 available | 0 unavailable
StrategyType:   RollingUpdate
MinReadySeconds: 0
RollingUpdateStrategy: 25% max unavailable, 25% max surge
Pod Template:
  Labels:  app=nginx
  Containers:
    nginx:
      Image:  nginx:1.14.2
      Port:   80/TCP
      Host Port:  80/TCP
      Environment: <none>
      Mounts:  <none>
      Volumes: <none>
  Conditions:
    Type     Status  Reason
    ----  -----  -----
    Available  True    MinimumReplicasAvailable
    Progressing  True   NewReplicaSetAvailable
OldReplicaSets: <none>
NewReplicaSet:  nginx-deployment-66b6c48dd5 (2/2 replicas created)
Events:
  Type     Reason          Age   From            Message
  ----  -----  ----  ----
  Normal  ScalingReplicaSet 2m45s  deployment-controller  Scaled up replica set nginx-deployment-66b6c48dd5 to 2
```

```
λ Cmdr
λ kubectl get pods -l app=nginx
NAME          READY   STATUS    RESTARTS   AGE
nginx-deployment-66b6c48dd5-g9jzp  1/1     Running   0          4m1s
nginx-deployment-66b6c48dd5-vq4gw  1/1     Running   0          4m1s

λ kubectl describe pod nginx-deployment-66b6c48dd5-g9jzp
Name:           nginx-deployment-66b6c48dd5-g9jzp
Namespace:      default
Priority:      0
Node:          multinode-demo-m03/192.168.58.4
Start Time:    Sun, 10 Oct 2021 10:42:49 -0500
Labels:         app=nginx
Annotations:   pod-template-hash=66b6c48dd5
Status:        Running
IP:            10.244.2.3
IPs:           IP: 10.244.2.3
Controlled By: ReplicaSet/nginx-deployment-66b6c48dd5 (rc)  deployment-controller scaled up replica set nginx-deployment-66b6c48dd5
Containers:
  nginx:
    Container ID:  docker://edb4b16aab01c8d32f6eeba495db19802d673513e2bc16ec76a39c0b9108e6dd
    Image:         nginx:1.14.2
    Image ID:     docker-pullable://nginxsha256:f7988fb6c02e0ce69257d9bd9cf37ae28a60f1df7563c3a2a6abe24160306b8d
    Port:          80/TCP
    Host Port:    80/TCP
    State:        Running
      Started:   Sun, 10 Oct 2021 10:43:20 -0500
    Ready:        True
    Restart Count: 0
    Environment:  <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-5mb9k (ro)
  Conditions:
    Type     Status
    Initialized  True
    Ready      True
    ContainersReady  True
    PodScheduled  True
  Volumes:
    kube-api-access-5mb9k:
      Type:      Projected (a volume that contains injected data from multiple sources)
      TokenExpirationSeconds: 3607
      ConfigMapName:  kube-root-ca.crt
      ConfigMapOptional: <nil>
      DownwardAPI:   true
  QoS Class:  BestEffort (multi-node)
  Node-Selectors:  <none>
  Tolerations:   node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                 node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
```

Se puede ver que se han creado los pods y la descripción del primero de ellos

4. Use a non-trivial application to validate your configuration (test on single node and multi-node)

```
cassandra-statefulset.yaml: Bloc de notas
Archivo Edición Fmtato Ver Ayuda
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: cassandra
  labels:
    app: cassandra
spec:
  serviceName: cassandra
  replicas: 3
  selector:
    matchLabels:
      app: cassandra
  template:
    metadata:
      labels:
        app: cassandra
    spec:
      terminationGracePeriodSeconds: 1800
      containers:
        - name: cassandra
          image: gcr.io/google-samples/cassandra:v13
          imagePullPolicy: Always
          ports:
            - containerPort: 7000
              name: intra-node
            - containerPort: 7001
              name: tls-intra-node
            - containerPort: 7199
              name: jmx
            - containerPort: 9042
              name: cql
          resources:
            limits:
              cpu: "500m"
              memory: 1Gi
            requests:
              cpu: "500m"
              memory: 1Gi
          securityContext:
            capabilities:
              add:
                - IPC_LOCK
          lifecycle:
            preStop:
              exec:
                command:
                  - /bin/sh
                  - -c
                  - nodetool drain
            env:
              - name: MAX_HEAP_SIZE
                value: 512M
              - name: HEAP_NEWSIZE
                value: 100M
              - name: CASSANDRA_SEEDS
                value: "cassandra-0.cassandra.default.svc.cluster.local"
              - name: CASSANDRA_CLUSTER_NAME
                value: "K8Demo"
              - name: CASSANDRA_DC
                value: "DC1-K8Demo"
              - name: CASSANDRA_RACK
                value: "Rack1-K8Demo"
              - name: POD_IP
                valueFrom:
                  fieldRef:
                    fieldPath: status.podIP
            readinessProbe:
              exec:
                command:
                  - /bin/bash
                  - -c
                  - /ready-probe.sh
            initialDelaySeconds: 15
            timeoutSeconds: 5
          volumeMounts:
            - name: cassandra-data
              mountPath: /cassandra_data
  volumeClaimTemplates:
    - metadata:
        name: cassandra-data
      spec:
        accessModes: [ "ReadWriteOnce" ]
        storageClassName: fast
        resources:
          requests:
            storage: 1Gi
    ...
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: fast
  provisioner: k8s.io/minikube-hostpath
  parameters:
    type: pd-ssd
```

Con esto podemos ver que cassandra está listo para ser ejecutado

```
$ kubectl get statefulset cassandra
NAME          READY   AGE
cassandra     3/3    11m
```

```
$ kubectl get pods -l="app=cassandra"
NAME        READY   STATUS    RESTARTS   AGE
cassandra-0 1/1     Running   0          11m
cassandra-1 1/1     Running   0          5m6s
cassandra-2 1/1     Running   0          3m25s
```

```
$ kubectl exec -it cassandra-0 -- nodetool status
Unable to use a TTY - input is not a terminal or the right kind of file
Datacenter: DC1-K8Demo
=====
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
-- Address      Load      Tokens  Owns (effective)  Host ID
   Rack
UN 172.17.0.3  104.55 KiB  32        75.2%           a14aa372-4fc6-41e9-a7
37-90ee35df6ef3  Rack1-K8Demo
UN 172.17.0.5  65.87 KiB  32        72.4%           1930c321-f5d8-4668-ac9
2-eca07ec92d45  Rack1-K8Demo
UN 172.17.0.4  70.89 KiB  32        70.4%           c1e4bac5-13a4-4dcd-ba1
d-e9176b913c97  Rack1-K8Demo
```

Cuando ejecutamos, nos muestra el estado del anillo.