

KUBERNETES

Introducción y comportamiento imperativo

Pods

Gestión de un pod individual con imagen custom con networking

Levantar el servicio web estático de node proporcionado en la práctica 1, usando Kubernetes. Este servicio:

- *Deberá ser levantado usando un comando de "kubectl run", para solo levantar el pod*
- *Tendrá que tener expuesto el puerto 32170 del host, y poder verse el sitio web estático desde allí. Usar el servicio web que vea conveniente para conseguir eso.*

Se deberá registrar en el PDF de la memoria los comandos utilizados, así como una captura de kubectl get y kubectl describe sobre el pod.

Se solicita utilizar la imagen de la primera actividad. Esta fue publicada en Docker Hub y la mía se llama “devandmus/p1-node”

The screenshot shows the Docker Hub interface for the repository `devandmus/p1-node`. The page includes a search bar, navigation links (Explore, Repositories, Organizations, Help), and a user profile for `devandmus`. The repository page has tabs for General, Tags, Builds, Collaborators, Webhooks, and Settings. A banner for Advanced Image Management is visible. The repository details show it was created 17 days ago and has 1 tag. The 'Tags and Scans' section displays a table with one tag, `1.0`, which was pushed 17 days ago and pulled 3 minutes ago. The 'Automated Builds' section is also visible, indicating that builds are disabled.

TAG	OS	PULLED	PUSHED
1.0	linux	3 minutes ago	17 days ago

Una vez creada, con la ayuda de “kubectl run” levanto el pod utilizando dicha imagen. Dentro de las opciones, expongo el puerto 8080 ya que ahí es donde vive la instancia de dicha imagen.

```
# amaldonado @ Macbook-Air in ~/Development/master/master-fullstack
$ kubectl run activity --image=devandmus/p1-node:1.0 --port=8080
pod/activity created
```

Verifico que el pod está corriendo con el comando “kubectl get all”:

```
# amaldonado @ Macbook-Air in ~/Development/master/master-fullstack/09-contene
$ kubectl get all
```

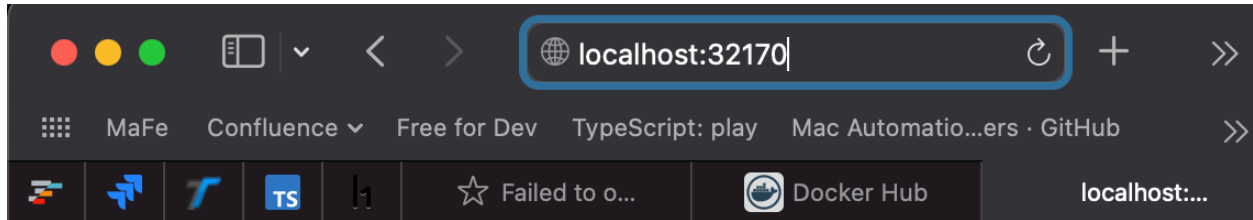
NAME	READY	STATUS	RESTARTS	AGE
pod/activity	1/1	Running	0	58s

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	108m

Procedo a exponer el servicio, como se solicita, en el puerto 32170 del host utilizando el comando “port-forward”:

```
# amaldonado @ Macbook-Air in ~/Development/M
$ kubectl port-forward activity 32170:8080
Forwarding from 127.0.0.1:32170 -> 8080
Forwarding from [::1]:32170 -> 8080
```

Se obtiene el resultado esperado y la app se consume en el puerto 32170.



Hello World!

Para terminar, adjunto captura del “get” y “describe” sobre el pod:

```
# amaldonado @ Macbook-Air in ~/Development/master
$ kubectl get pod/activity
```

NAME	READY	STATUS	RESTARTS	AGE
activity	1/1	Running	0	3m10s

```
# amaldonado @ Macbook-Air in ~/Development/master/master-fullstack/09-contenedores/actividad-0
$ kubectl describe pod/activity
```

Name: activity
Namespace: default
Priority: 0
Node: docker-desktop/192.168.65.4
Start Time: Fri, 11 Mar 2022 19:11:43 -0300
Labels: run=activity
Annotations: <none>
Status: Running
IP: 10.1.0.14
IPs:
IP: 10.1.0.14
Containers:
activity:
Container ID: docker://5967b446e3b07f48209c3f2534a36fd4f9841726f8eb312bcd015dea616294a7
Image: devandmus/p1-node:1.0
Image ID: docker-pullable://devandmus/p1-node@sha256:34cbad6d540e6fa789906b79d5098211
Port: 8080/TCP
Host Port: 0/TCP
State: Running
Started: Fri, 11 Mar 2022 19:11:44 -0300
Ready: True
Restart Count: 0
Environment: <none>
Mounts:
/var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-759c9 (ro)

```

Mounts:
  /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-759c9 (ro)
Conditions:
  Type                Status
  Initialized          True
  Ready                True
  ContainersReady      True
  PodScheduled         True
Volumes:
  kube-api-access-759c9:
    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
Node-Selectors:        <none>
Tolerations:           node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                      node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type    Reason      Age    From          Message
  ----    -
  Normal  Scheduled   4m1s   default-scheduler Successfully assigned default/activity to docker-desktop
  Normal  Pulled      4m     kubelet       Container image "devandmus/p1-node:1.0" already present on machine
  Normal  Created     4m     kubelet       Created container activity
  Normal  Started     4m     kubelet       Started container activity

```

Deployments

Re-escalado de replica sets

Este simple ejercicio practicaremos con los replica sets. Usando un 'httpd' (apache server) como imagen base, realizaremos un deployment que tenga replica set 5.

Tras ello, eliminaremos dos de los pods contenidos en el replica-set. Una vez hecho, bajaremos el replica set a 2. ¿Cuáles son los contenedores que han quedado en el deployment, y por qué motivo?

La entrega de este ejercicio debe ser compuesta por las capturas de los comandos ejecutados, así como una explicación a lo sucedido en la propia memoria

Comienzo ejecutando el comando para crear las réplicas necesarias desde la imagen “httpd”:

```
# amaldonado @ Macbook-Air in ~/Development/master/master-fullstack
$ kubectl create deployment activity2 --replicas=5 --image=httpd
deployment.apps/activity2 created
```

Verifico la correcta creación e inicialización de todas las instancias con “get all”:

```
# amaldonado @ Macbook-Air in ~/Development/master/master-fullstack/09-content
$ kubectl get all
```

NAME	READY	STATUS	RESTARTS	AGE
pod/activity2-6dd4bc879-42fwd	1/1	Running	0	11s
pod/activity2-6dd4bc879-47964	1/1	Running	0	11s
pod/activity2-6dd4bc879-8d9dg	1/1	Running	0	11s
pod/activity2-6dd4bc879-nfp6z	1/1	Running	0	12s
pod/activity2-6dd4bc879-vfv6h	1/1	Running	0	11s

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	22m

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
deployment.apps/activity2	5/5	5	5	12s

NAME	DESIRED	CURRENT	READY	AGE
replicaset.apps/activity2-6dd4bc879	5	5	5	12s

Como se solicita claramente, procedo a **eliminar** dos de los pods, el primero y el segundo:

```
# amaldonado @ Macbook-Air in ~/Development/master/
$ kubectl delete pod/activity2-6dd4bc879-42fwd
pod "activity2-6dd4bc879-42fwd" deleted

# amaldonado @ Macbook-Air in ~/Development/master/
$ kubectl delete pod/activity2-6dd4bc879-47964
pod "activity2-6dd4bc879-47964" deleted
```

Con “get all” podemos observar que luego de ser eliminados, fueron recreados y se diferencian de sus hermanos por su tiempo de vida:

```
# amaldonado @ Macbook-Air in ~/Development/master/master-fullstack/0
$ kubectl get all
```

NAME	READY	STATUS	RESTARTS	AGE
pod/activity2-6dd4bc879-8d9dg	1/1	Running	0	103s
pod/activity2-6dd4bc879-nfp6z	1/1	Running	0	104s
pod/activity2-6dd4bc879-pfnfr	1/1	Running	0	28s
pod/activity2-6dd4bc879-spj6m	1/1	Running	0	40s
pod/activity2-6dd4bc879-vfv6h	1/1	Running	0	103s

Ahora, como solicitado, **bajaré** el replica set a 2, esto significa que escalaré las réplicas usando “scale”

```
# amaldonado @ Macbook-Air in ~/Development/master/
$ kubectl scale --replicas=2 deployment activity2
deployment.apps/activity2 scaled
```

Ahora verifico las réplicas:

```
# amaldonado @ Macbook-Air in ~/Development/master/master-fullstack/09-conte
$ kubectl get all
```

NAME	READY	STATUS	RESTARTS	AGE
pod/activity2-6dd4bc879-nfp6z	1/1	Running	0	3m53s
pod/activity2-6dd4bc879-pfnfr	1/1	Running	0	2m37s

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	25m

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
deployment.apps/activity2	2/2	2	2	3m53s

NAME	DESIRED	CURRENT	READY	AGE
replicaset.apps/activity2-6dd4bc879	2	2	2	3m53s

Tras evaluar esta captura y contrastar los IDs, los contenedores que se quedaron fueron tanto el primero como el último en correr. El terminado en **6z** fue el primero en levantarse cuando se crearon las réplicas y luego de borrar 2 instancias, el terminado en **fr** fue el último en levantarse. Investigando, la eliminación de los Pods se basa en una serie de factores que son evaluados, no es un proceso al azar. Teniendo esto presente, asumo que se guarda el mas viejo ya que su tiempo de ejecución sin fallos lo define como “el mas estable” y se guarda el mas nuevo por ser la réplica más fresca.

Deployment con networking

Por último, se pedirá realizar un deployment de una aplicación usando además networking. Para ello se pedirá exponer de un Deployment de httpd con replica-set 2 el puerto 80 de las aplicaciones. Describir el servicio. A continuación, crear para uno de los dos https un NodePort para ser accedido desde el Host.

En la memoria deberá ir contenido el proceso para desplegar dicho servicio, así como un kubectl describe de los servicios levantados, y de los pods.

Primero que todo, como solicitado, se expone un deployment de httpd con replica 2 en el puerto 80

```
# amaldonado @ Macbook-Air in ~/Development/master/master-fullstack/09-conten
$ kubectl create deployment activity3 --replicas=2 --image=httpd --port=80
deployment.apps/activity3 created
```

Verifico la correcta creación con “get all”

```
# amaldonado @ Macbook-Air in ~/Development/master/master-fullstack/09-contene
$ kubectl get all
```

NAME	READY	STATUS	RESTARTS	AGE
pod/activity3-79bf9d6879-6d79k	1/1	Running	0	106s
pod/activity3-79bf9d6879-kpq7f	1/1	Running	0	106s

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	51m

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
deployment.apps/activity3	2/2	2	2	106s

NAME	DESIRED	CURRENT	READY	AGE
replicaset.apps/activity3-79bf9d6879	2	2	2	106s

Como solicitado creo para uno de los nodes un servicio con “type: NodePort” al que llamo “activity3-cluster”

```
# amaldonado @ Macbook-Air in ~/Development/master/master-fullstack/09-contenedores/actividad-03 on
$ kubectl expose pod/activity3-79bf9d6879-6d79k --port=80 --name=activity3-cluster --type=NodePort
service/activity3-cluster exposed
```

Con “get all” verifico su existencia

```
# amaldonado @ Macbook-Air in ~/Development/master/master-fullstack/09-contenedores/activi
$ kubectl get all
```

NAME	READY	STATUS	RESTARTS	AGE
pod/activity3-79bf9d6879-6d79k	1/1	Running	0	15m
pod/activity3-79bf9d6879-kpq7f	1/1	Running	0	15m

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/activity3-cluster	NodePort	10.104.78.36	<none>	80:30755/TCP	15s
service/kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	64m

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
deployment.apps/activity3	2/2	2	2	15m

NAME	DESIRED	CURRENT	READY	AGE
replicaset.apps/activity3-79bf9d6879	2	2	2	15m

Como solicitado, “describe” del servicio:

```
# amaldonado @ Macbook-Air in ~/Development/master/master-fullstack/09
$ kubectl describe service/activity3-cluster
```

Name: activity3-cluster
Namespace: default
Labels: app=activity3
pod-template-hash=79bf9d6879
Annotations: <none>
Selector: app=activity3,pod-template-hash=79bf9d6879
Type: NodePort
IP Family Policy: SingleStack
IP Families: IPv4
IP: 10.104.78.36
IPs: 10.104.78.36
LoadBalancer Ingress: localhost
Port: <unset> 80/TCP
TargetPort: 80/TCP
NodePort: <unset> 30755/TCP
Endpoints: 10.1.0.31:80,10.1.0.32:80
Session Affinity: None
External Traffic Policy: Cluster
Events: <none>

DISCLAIMER

Estuve durante horas (sin exagerar, horas) intentando terminar este enunciado. Trabajo con un **Mac M1** y se reporta que existen problemas con el Minikube en la comunidad

<https://github.com/kubernetes/minikube/issues/12036>

Seguí decenas de tutoriales, eliminé las dependencias una y otra vez, instalé y desinstalé sin éxito, cambié de versiones, etc...

Adjunto las múltiples instancias que quedan aún en mi local de la última versión de Docker y Kubernetes que instalé

```
# amaldonado @ Macbook-Air in ~/Development/Master/master-fullstack/09-contenedores/actividad-03 on
$ kubectl get all
```

NAME	READY	STATUS	RESTARTS	AGE
pod/activity3-757b7dff9-6nz59	1/1	Running	1 (69m ago)	97m
pod/activity3-757b7dff9-7bjrq	1/1	Running	1 (69m ago)	97m
pod/hello-node-6b89d599b9-55frv	0/1	CrashLoopBackOff	10 (4m35s ago)	30m
pod/test-m1-5898b7848c-tllff	1/1	Running	0	7m41s

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/activity3-cluster	NodePort	10.96.157.166	<none>	80:31653/TCP	97m
service/hello-node	LoadBalancer	10.101.5.70	127.0.0.1	8080:30485/TCP	28m
service/kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	101m
service/test-m1	NodePort	10.97.153.248	<none>	8080:32614/TCP	5m55s

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
deployment.apps/activity3	2/2	2	2	97m
deployment.apps/hello-node	0/1	1	0	30m
deployment.apps/test-m1	1/1	1	1	7m41s

NAME	DESIRED	CURRENT	READY	AGE
replicaset.apps/activity3-757b7dff9	2	2	2	97m
replicaset.apps/hello-node-6b89d599b9	1	1	0	30m
replicaset.apps/test-m1-5898b7848c	1	1	1	7m41s

Intenté hacerlo hasta con un LoadBalancer para generar un tunnel como indica la documentación del mismo Minikube. Ni eso funcionó:

```
# amaldonado @ Macbook-Air in ~/Development/Master/master-fullstack/09-contenedores/actividad-03 on git:main x [22:11:05]
$ minikube tunnel
✓ Tunnel successfully started

🚧 NOTE: Please do not close this terminal as this process must stay alive for the tunnel to be accessible ...

🚧 Starting tunnel for service hello-node.
```

El URL nunca funcionó.

Cuando corro el servicio según lo documentado para obtener el URL, se queda pegado

```
# amaldonado @ Macbook-Air in ~/Development/Master/master-fullstack/09-contenedores/actividad-03 on git:main x [22:11:05]
$ minikube service activity3-cluster
🚧 Starting tunnel for service activity3-cluster.
🚧 Opening service default/activity3-cluster in default browser...
! Because you are using a Docker driver on darwin, the terminal needs to be open to run it.
```

Esta es la página que intenta abrir

Safari Can't Open the Page

Safari can't open the page "192.168.49.2:31653" because the server where this page is located isn't responding.

Igualmente, es incapaz de devolver el url cuando solicito solo eso. Seguí muchos tutoriales donde con ese flag devuelve un url. A mi jamás me devolvió algo más que timeout.

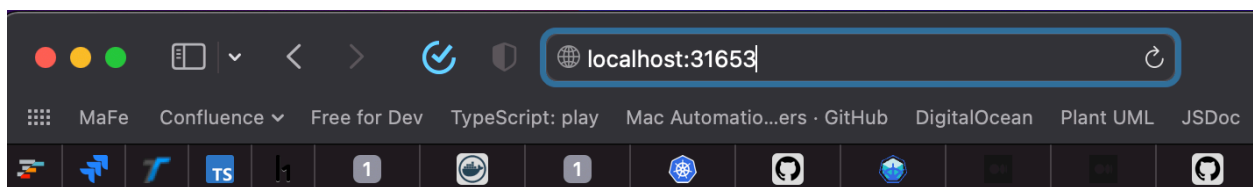
```
# amaldonado @ Macbook-Air in ~/Development/Master/master-fullstack/09-contenedores/actividad-0
$ minikube service activity3-cluster --url

🏃 Starting tunnel for service activity3-cluster.
! Because you are using a Docker driver on darwin, the terminal needs to be open to run it.
```

La única solución que encontré para poder exponer la página fue lamentablemente con el port-forward:

```
# amaldonado @ Macbook-Air in ~/Development/Master/master-fullst
$ kubectl port-forward service/activity3-cluster 31653:80
Forwarding from 127.0.0.1:31653 -> 80
Forwarding from [::1]:31653 -> 80
Handling connection for 31653
Handling connection for 31653
```

Esto si establece la conexión con el host como corresponde y devuelve la página:



It works!

Reintento por última vez, adjunto evidencia que todos los elementos son nuevos:

```
# amaldonado @ Macbook-Air in ~/Development/Master/master-fullstack/09-contenedores/actividad-03 on g
$ kubectl get all
```

NAME	READY	STATUS	RESTARTS	AGE
pod/activity3-5ccfc6f9c5-fkwgr	1/1	Running	0	4m15s
pod/activity3-5ccfc6f9c5-kj9pk	1/1	Running	0	4m19s
pod/hello-node-6b89d599b9-55frv	0/1	CrashLoopBackOff	14 (4m6s ago)	50m
pod/test-m1-5898b7848c-tllff	1/1	Running	0	27m

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/activity3-cluster	NodePort	10.96.157.166	<none>	80:31653/TCP	116m
service/hello-node	LoadBalancer	10.101.5.70	<pending>	8080:30485/TCP	48m
service/kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	121m
service/test-m1	NodePort	10.97.153.248	<none>	8080:32614/TCP	25m

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
deployment.apps/activity3	2/2	2	2	117m
deployment.apps/hello-node	0/1	1	0	50m
deployment.apps/test-m1	1/1	1	1	27m

NAME	DESIRED	CURRENT	READY	AGE
replicaset.apps/activity3-5ccfc6f9c5	2	2	2	4m19s
replicaset.apps/activity3-757b7dffb9	0	0	0	117m
replicaset.apps/hello-node-6b89d599b9	1	1	0	50m
replicaset.apps/test-m1-5898b7848c	1	1	1	27m

Sin exito, dejo como versión final del enunciado la solución con “port-forward”

Finalmente, como se solicita, termino por adjuntar los “kubectl describe” del servicio y los pods:

```
# amaldonado @ Macbook-Air in ~/Development/Master/master-fullstack/09-contenedores/actividad-03 on git:main
$ kubectl describe pod/activity3-5ccfc6f9c5-fkwgr
Name:          activity3-5ccfc6f9c5-fkwgr
Namespace:     default
Priority:       0
Node:          minikube/192.168.49.2
Start Time:    Fri, 11 Mar 2022 22:51:06 -0300
Labels:        app=activity3
               pod-template-hash=5ccfc6f9c5
Annotations:   kubectl.kubernetes.io/restartedAt: 2022-03-11T22:51:02-03:00
Status:        Running
IP:            172.17.0.10
IPs:
  IP:          172.17.0.10
Controlled By: ReplicaSet/activity3-5ccfc6f9c5
Containers:
  httpd:
    Container ID:  docker://cde1ad49e749a890af4d4317921a73c0f447d5d64fbe97273e7b623ef407b907
    Image:         httpd
    Image ID:      docker-pullable://httpd@sha256:b7907df5e39a98a087dec5e191e6624854844bc8d0202307428dd90b1
    Port:         80/TCP
    Host Port:    0/TCP
    State:        Running
      Started:    Fri, 11 Mar 2022 22:51:08 -0300
    Ready:        True
    Restart Count: 0
    Environment:  <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-tm7qp (ro)
Conditions:
  Type             Status
  Initialized       True
  Ready            True
  ContainersReady  True
  PodScheduled     True
Volumes:
  kube-api-access-tm7qp:
    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
Node-Selectors:    <none>
Tolerations:       node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                   node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
```

```
node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type    Reason      Age   From          Message
  ----    -
  Normal  Scheduled   5m42s default-scheduler Successfully assigned default/activity3-5ccfc6f9c5-fkwgr to minikube
  Normal  Pulling     5m41s kubelet       Pulling image "httpd"
  Normal  Pulled      5m40s kubelet       Successfully pulled image "httpd" in 1.76043525s
  Normal  Created     5m40s kubelet       Created container httpd
  Normal  Started     5m40s kubelet       Started container httpd
```

```
# amaldonado @ Macbook-Air in ~/Development/Master/master-fullstack/09-contenedores/actividad-03 on git
$ kubectl describe pod/activity3-5ccfc6f9c5-kj9pk
Name:          activity3-5ccfc6f9c5-kj9pk
Namespace:     default
Priority:       0
Node:          minikube/192.168.49.2
Start Time:    Fri, 11 Mar 2022 22:51:02 -0300
Labels:        app=activity3
               pod-template-hash=5ccfc6f9c5
Annotations:   kubectl.kubernetes.io/restartedAt: 2022-03-11T22:51:02-03:00
Status:        Running
IP:            172.17.0.9
IPs:
  IP:          172.17.0.9
Controlled By: ReplicaSet/activity3-5ccfc6f9c5
Containers:
  httpd:
    Container ID:  docker://a5012a0331cd6986972725ad6f186bdd898bc7ce3cd41ca46989efb3247f2d39
    Image:         httpd
    Image ID:      docker-pullable://httpd@sha256:b7907df5e39a98a087dec5e191e6624854844bc8d0202307428dd
    Port:         80/TCP
    Host Port:    0/TCP
    State:        Running
      Started:    Fri, 11 Mar 2022 22:51:05 -0300
    Ready:        True
    Restart Count: 0
    Environment:  <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-dfq9k (ro)
Conditions:
  Type          Status
  Initialized    True
  Ready         True
  ContainersReady True
  PodScheduled   True
Volumes:
  kube-api-access-dfq9k:
    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
Node-Selectors:  <none>
Tolerations:     node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                  node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
```

```
node.kubernetes.io/unreachable:NoExecute op=Exists for 300s

Events:
  Type    Reason      Age   From          Message
  ----    -
  Normal  Scheduled   6m36s default-scheduler Successfully assigned default/activity3-5ccfc6f9c5-kj9pk to minikube
  Normal  Pulling     6m36s kubelet       Pulling image "httpd"
  Normal  Pulled      6m34s kubelet       Successfully pulled image "httpd" in 2.290775293s
  Normal  Created     6m34s kubelet       Created container httpd
  Normal  Started     6m34s kubelet       Started container httpd
```

```
# amaldonado @ Macbook-Air in ~/Development/Master/master-fullstack/0
$ kubectl describe service/activity3-cluster
Name: activity3-cluster
Namespace: default
Labels: app=activity3
        pod-template-hash=757b7dffb9
Annotations: <none>
Selector: app=activity3,pod-template-hash=757b7dffb9
Type: NodePort
IP Family Policy: SingleStack
IP Families: IPv4
IP: 10.96.157.166
IPs: 10.96.157.166
Port: <unset> 80/TCP
TargetPort: 80/TCP
NodePort: <unset> 31653/TCP
Endpoints: <none>
Session Affinity: None
External Traffic Policy: Cluster
Events: <none>
```