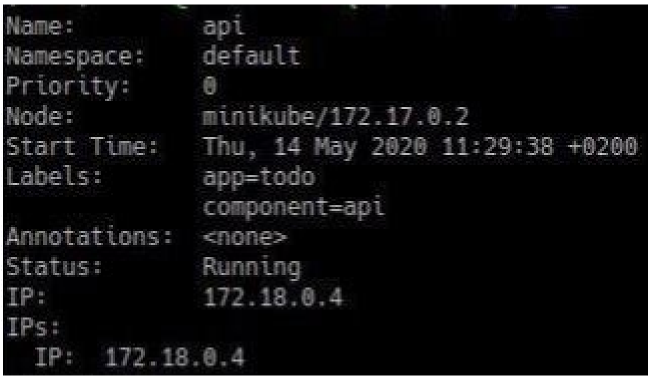
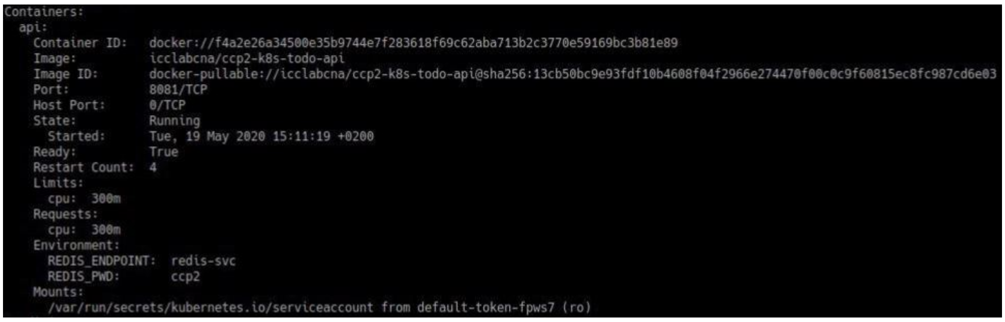
CLD - Lab 06 : Kubernetes

Task 1 - Deliverables

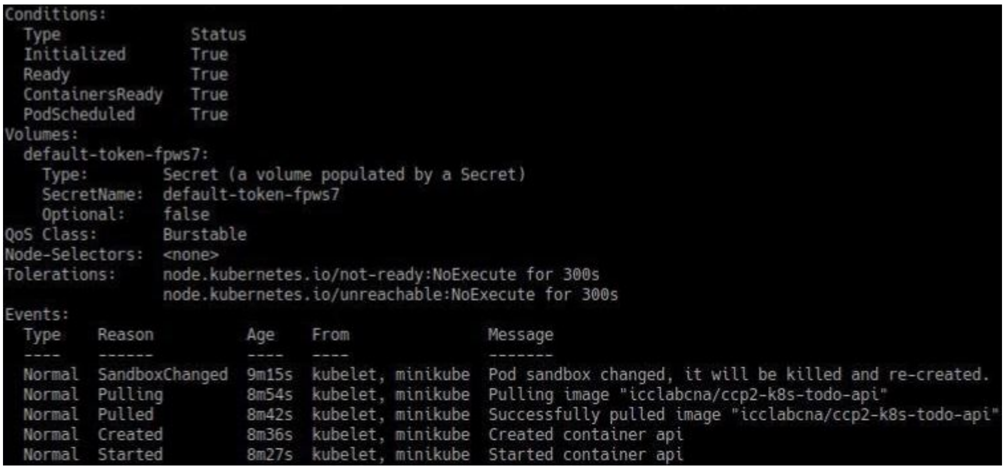
- Document any difficulties you faced and how you overcame them.
  - We didn't know at first how to assign a value for the endpoint of the localhost. By trying things out we managed to find a solution.
- Copy the object descriptions into the lab report.
  - Description of the Pods:
    - The screenshots below are the description of the Pod API. We make 3 screenshots to have bigger screenshots. This seperation has been done because we cannot edit the size of the image in a markdown document.



```
Name:         api
Namespace:    default
Priority:     0
Node:         minikube/172.17.0.2
Start Time:   Thu, 14 May 2020 11:29:38 +0200
Labels:       app=todo
              component=api
Annotations:  <none>
Status:       Running
IP:           172.18.0.4
IPs:
  IP:  172.18.0.4
```

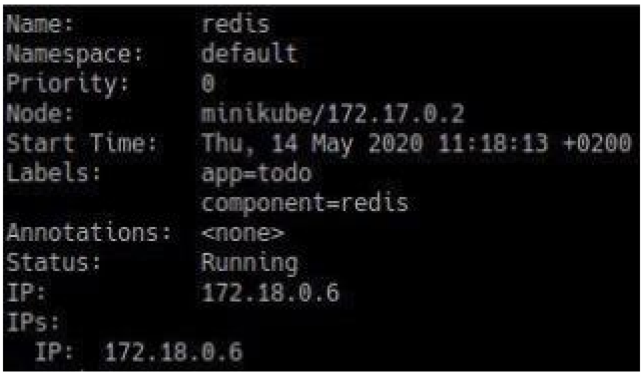


```
Containers:
  api:
    Container ID:   docker://f4a2e26a34500e35b9744e7f283618f69c62aba713b2c3770e59169bc3b81e89
    Image:          icclabcna/ccp2-k8s-todo-api
    Image ID:       docker-pullable://icclabcna/ccp2-k8s-todo-api@sha256:13cb50bc9e93fdf10b4608f04f2966e274470f00c0c9f60815ec8fc987cd6e03
    Port:           8081/TCP
    Host Port:      0/TCP
    State:          Running
      Started:      Tue, 19 May 2020 15:11:19 +0200
    Ready:          True
    Restart Count:  4
    Limits:
      cpu:  300m
    Requests:
      cpu:  300m
    Environment:
      REDIS_ENDPOINT:  redis-svc
      REDIS_PWD:       ccp2
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from default-token-fpws7 (ro)
```

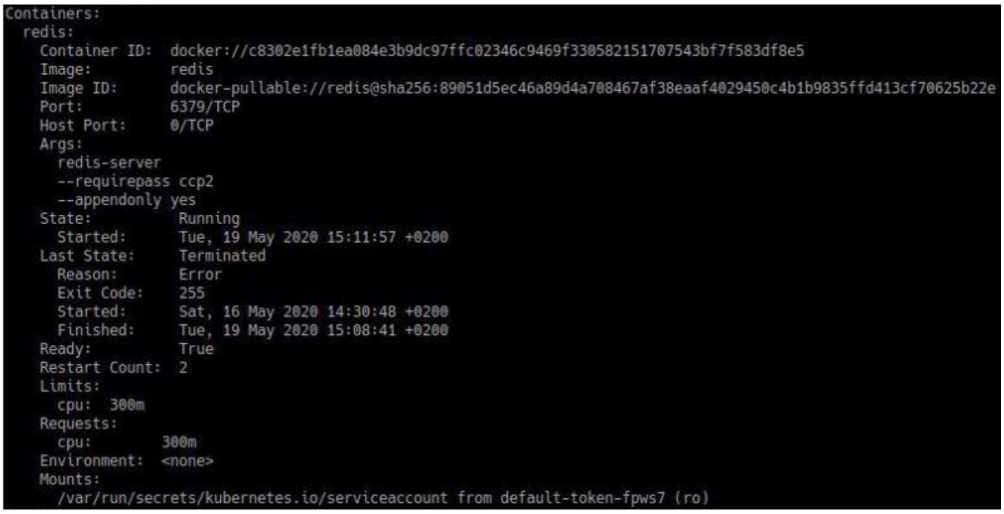


```
Conditions:
  Type              Status
  Initialized       True
  Ready             True
  ContainersReady   True
  PodScheduled      True
Volumes:
  default-token-fpws7:
    Type:        Secret (a volume populated by a Secret)
    SecretName:  default-token-fpws7
    Optional:    false
QoS Class:       Burstable
Node-Selectors:  <none>
Tolerations:     node.kubernetes.io/not-ready:NoExecute for 300s
                 node.kubernetes.io/unreachable:NoExecute for 300s
Events:
  Type    Reason          Age    From               Message
  ----    ------          ----   ----               -------
  Normal  SandboxChanged  9m15s  kubelet, minikube  Pod sandbox changed, it will be killed and re-created.
  Normal  Pulling         8m54s  kubelet, minikube  Pulling image "icclabcna/ccp2-k8s-todo-api"
  Normal  Pulled          8m42s  kubelet, minikube  Successfully pulled image "icclabcna/ccp2-k8s-todo-api"
  Normal  Created         8m36s  kubelet, minikube  Created container api
  Normal  Started         8m27s  kubelet, minikube  Started container api
```

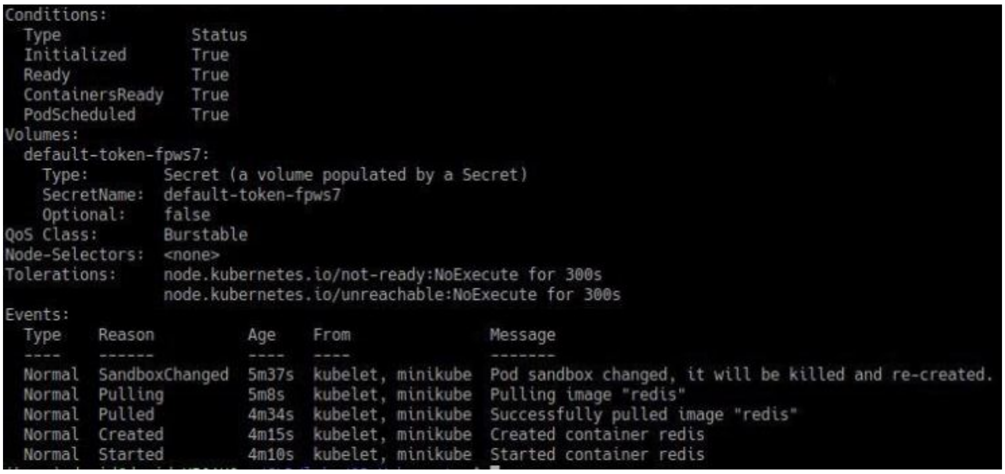
    - The screenshots below are the description of the Pod Redis.



```
Name:           redis
Namespace:      default
Priority:       0
Node:           minikube/172.17.0.2
Start Time:     Thu, 14 May 2020 11:18:13 +0200
Labels:         app=todo
                component=redis
Annotations:    <none>
Status:         Running
IP:             172.18.0.6
IPs:
  IP:  172.18.0.6
```

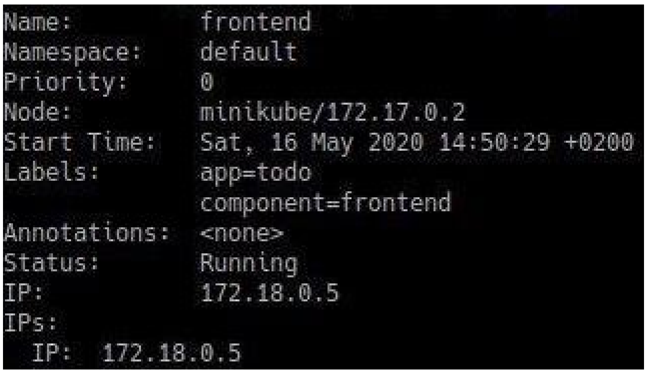


```
Containers:
  redis:
    Container ID:  docker://c8302e1fb1ea084e3b9dc97ffc02346c9469f330582151707543bf7f583df8e5
    Image:         redis
    Image ID:      docker-pullable://redis@sha256:89051d5ec46a89d4a708467af38eaaf4029450c4b1b9835ffd413cf70625b22e
    Port:          6379/TCP
    Host Port:     0/TCP
    Args:
      redis-server
      --requirepass ccp2
      --appendonly yes
    State:          Running
      Started:      Tue, 19 May 2020 15:11:57 +0200
    Last State:     Terminated
      Reason:       Error
      Exit Code:    255
      Started:      Sat, 16 May 2020 14:30:48 +0200
      Finished:     Tue, 19 May 2020 15:08:41 +0200
    Ready:          True
    Restart Count:  2
    Limits:
      cpu:  300m
    Requests:
      cpu:        300m
    Environment:  <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from default-token-fpws7 (ro)
```

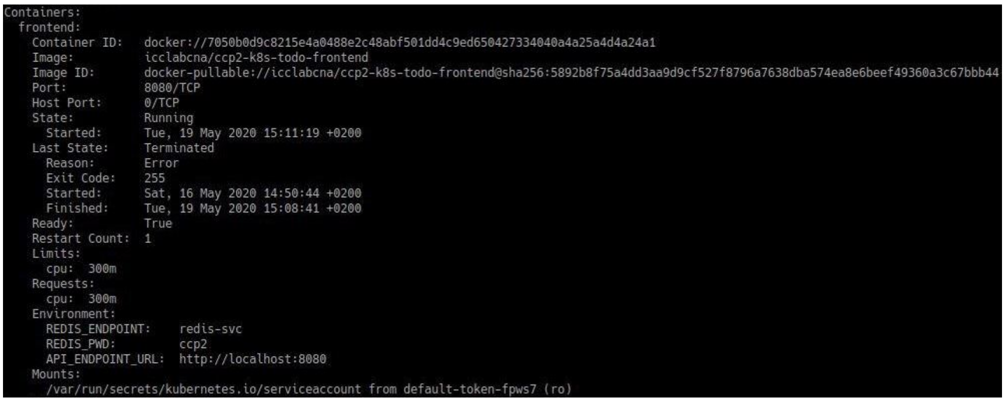


```
Conditions:
  Type              Status
  Initialized       True
  Ready             True
  ContainersReady   True
  PodScheduled      True
Volumes:
  default-token-fpws7:
    Type:        Secret (a volume populated by a Secret)
    SecretName:  default-token-fpws7
    Optional:    false
QoS Class:       Burstable
Node-Selectors:  <none>
Tolerations:     node.kubernetes.io/not-ready:NoExecute for 300s
                 node.kubernetes.io/unreachable:NoExecute for 300s
Events:
  Type    Reason          Age    From               Message
  ----    ------          ----   ----               -------
  Normal  SandboxChanged  5m37s  kubelet, minikube  Pod sandbox changed, it will be killed and re-created.
  Normal  Pulling         5m8s   kubelet, minikube  Pulling image "redis"
  Normal  Pulled          4m34s  kubelet, minikube  Successfully pulled image "redis"
  Normal  Created         4m15s  kubelet, minikube  Created container redis
  Normal  Started         4m10s  kubelet, minikube  Started container redis
```

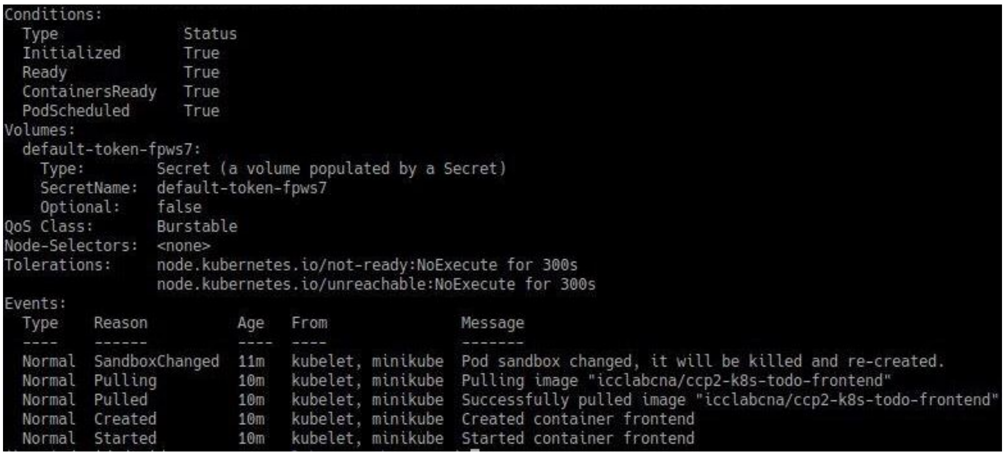
- The screenshots below are the description of the Pod Frontend.



```
Name:           frontend
Namespace:      default
Priority:       0
Node:           minikube/172.17.0.2
Start Time:     Sat, 16 May 2020 14:50:29 +0200
Labels:         app=todo
                component=frontend
Annotations:    <none>
Status:         Running
IP:             172.18.0.5
IPs:
  IP:  172.18.0.5
```

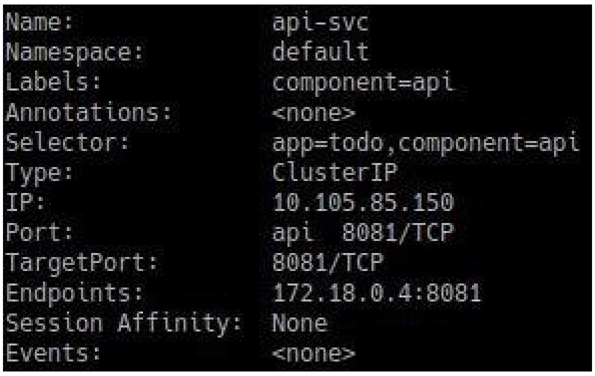


```
Containers:
  frontend:
    Container ID:   docker://7050b0d9c8215e4a0488e2c48abf501dd4c9ed650427334040a4a25a4d4a24a1
    Image:          icclabcna/ccp2-k8s-todo-frontend
    Image ID:       docker-pullable://icclabcna/ccp2-k8s-todo-frontend@sha256:5892b8f75a4dd3aa9d9cf527f8796a7638dba574ea8e6beef49360a3c67bbb44
    Port:           8080/TCP
    Host Port:      0/TCP
    State:          Running
      Started:      Tue, 19 May 2020 15:11:19 +0200
    Last State:     Terminated
      Reason:       Error
      Exit Code:    255
      Started:      Sat, 16 May 2020 14:50:44 +0200
      Finished:     Tue, 19 May 2020 15:08:41 +0200
    Ready:          True
    Restart Count:  1
    Limits:
      cpu:  300m
    Requests:
      cpu:  300m
    Environment:
      REDIS_ENDPOINT:    redis-svc
      REDIS_PWD:         ccp2
      API_ENDPOINT_URL:  http://localhost:8080
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from default-token-fpws7 (ro)
```



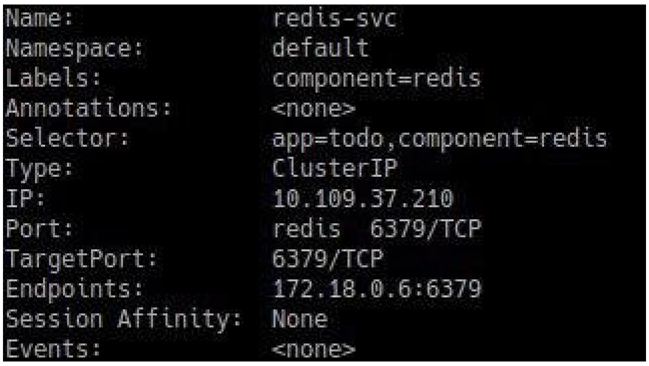
```
Conditions:
  Type              Status
  Initialized       True
  Ready             True
  ContainersReady   True
  PodScheduled      True
Volumes:
  default-token-fpws7:
    Type:        Secret (a volume populated by a Secret)
    SecretName:  default-token-fpws7
    Optional:    false
QoS Class:       Burstable
Node-Selectors:  <none>
Tolerations:     node.kubernetes.io/not-ready:NoExecute for 300s
                 node.kubernetes.io/unreachable:NoExecute for 300s
Events:
  Type    Reason          Age   From               Message
  ----    ------          ----  ----               -------
  Normal  SandboxChanged  11m   kubelet, minikube  Pod sandbox changed, it will be killed and re-created.
  Normal  Pulling         10m   kubelet, minikube  Pulling image "icclabcna/ccp2-k8s-todo-frontend"
  Normal  Pulled          10m   kubelet, minikube  Successfully pulled image "icclabcna/ccp2-k8s-todo-frontend"
  Normal  Created         10m   kubelet, minikube  Created container frontend
  Normal  Started         10m   kubelet, minikube  Started container frontend
```

- Description of the Services:
  - The screenshot below is the description of the service API.



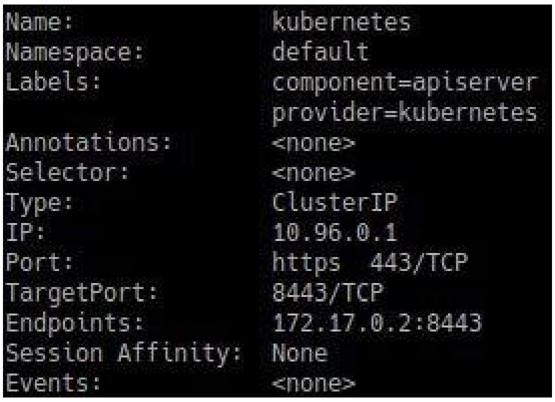
```
Name:              api-svc
Namespace:         default
Labels:            component=api
Annotations:       <none>
Selector:          app=todo,component=api
Type:              ClusterIP
IP:                10.105.85.150
Port:              api  8081/TCP
TargetPort:        8081/TCP
Endpoints:         172.18.0.4:8081
Session Affinity:  None
Events:            <none>
```

  - The screenshot below is the description of the service Redis.



```
Name:              redis-svc
Namespace:         default
Labels:            component=redis
Annotations:       <none>
Selector:          app=todo,component=redis
Type:              ClusterIP
IP:                10.109.37.210
Port:              redis  6379/TCP
TargetPort:        6379/TCP
Endpoints:         172.18.0.6:6379
Session Affinity:  None
Events:            <none>
```

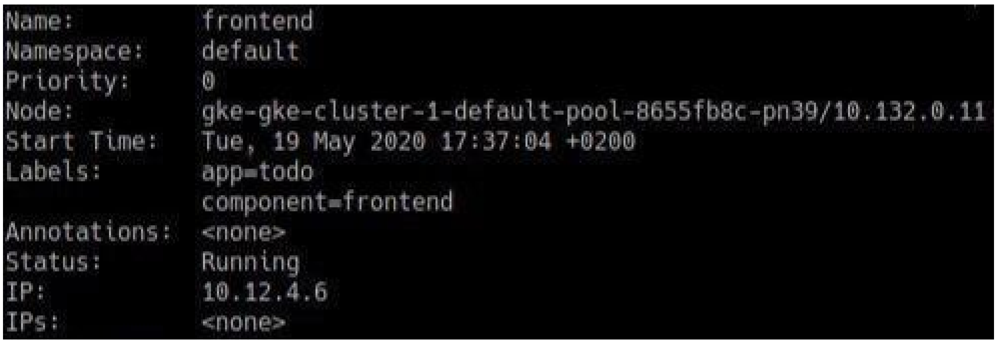
  - The screenshot below is the description of the service Kubernetes.



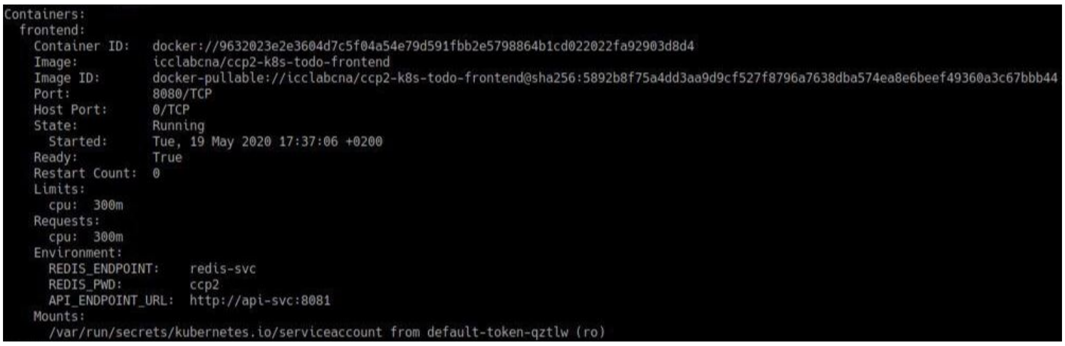
```
Name:              kubernetes
Namespace:         default
Labels:            component=apiserver
                   provider=kubernetes
Annotations:       <none>
Selector:          <none>
Type:              ClusterIP
IP:                10.96.0.1
Port:              https  443/TCP
TargetPort:        8443/TCP
Endpoints:         172.17.0.2:8443
Session Affinity:  None
Events:            <none>
```

Task 2 - Deliverables

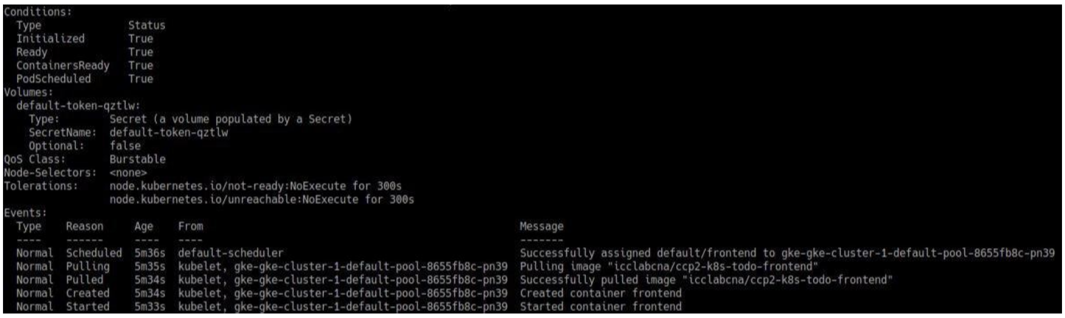
- Document any difficulties you faced and how you overcame them.
    - The first difficulty was finding what was a target port and how to use it. Then we had to correctly map the ip addresses and ports of the different pods/services so that everything matches. We found the details in the download files about how to do it.
    - The second problem was connecting the pod to the service when creating it. It said the cluster had not enough nodes. We had to modify the default pool in the cluster details at the bottom of the page.
    - We couldn't connect to the frontend service so we had to increase the number of nodes again. We used the clusterIP instead of the load-balancer so this was also something wrong we did
- Copy the object descriptions into the lab report (if they are unchanged from the previous task just say so).
    - Only the Pod Frontend has been modified. All other services/Pods are unchanged. The screenshots below show the new description of the Pod Frontend.



```
Name:         frontend
Namespace:    default
Priority:     0
Node:         gke-gke-cluster-1-default-pool-8655fb8c-pn39/10.132.0.11
Start Time:   Tue, 19 May 2020 17:37:04 +0200
Labels:       app=todo
              component=frontend
Annotations:  <none>
Status:       Running
IP:           10.12.4.6
IPs:          <none>
```

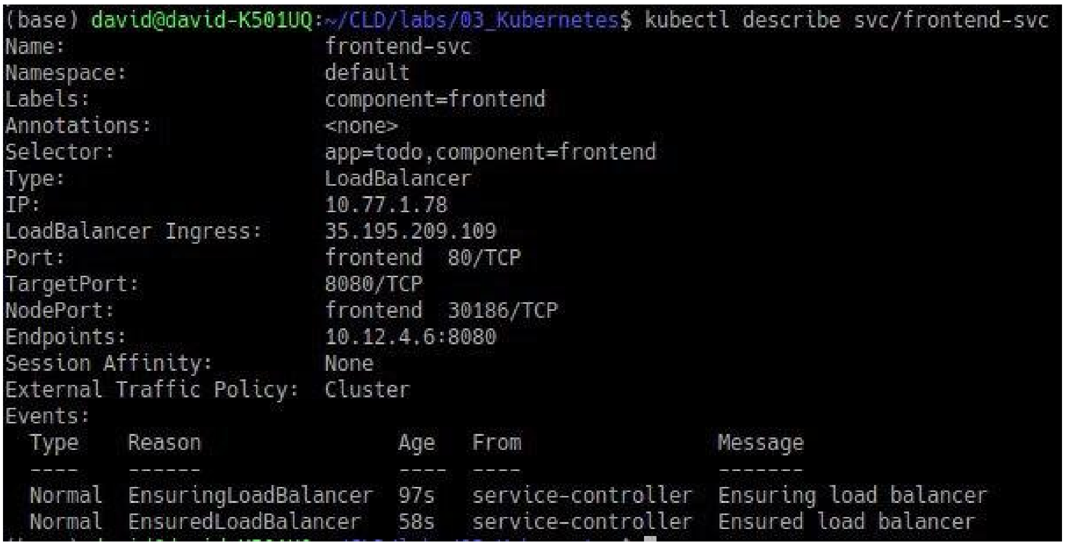


```
Containers:
  frontend:
    Container ID:   docker://9632023e2e3604d7c5f04a54e79d591fbb2e5798864b1cd022022fa92903d8d4
    Image:          icclabcna/ccp2-k8s-todo-frontend
    Image ID:       docker-pullable://icclabcna/ccp2-k8s-todo-frontend@sha256:5892b8f75a4dd3aa9d9cf527f8796a7638dba574ea8e6beef49360a3c67bbb44
    Port:           8080/TCP
    Host Port:      0/TCP
    State:          Running
      Started:      Tue, 19 May 2020 17:37:06 +0200
    Ready:          True
    Restart Count:  0
    Limits:
      cpu:  300m
    Requests:
      cpu:  300m
    Environment:
      REDIS_ENDPOINT:    redis-svc
      REDIS_PWD:         ccp2
      API_ENDPOINT_URL:  http://api-svc:8081
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from default-token-qztlw (ro)
```



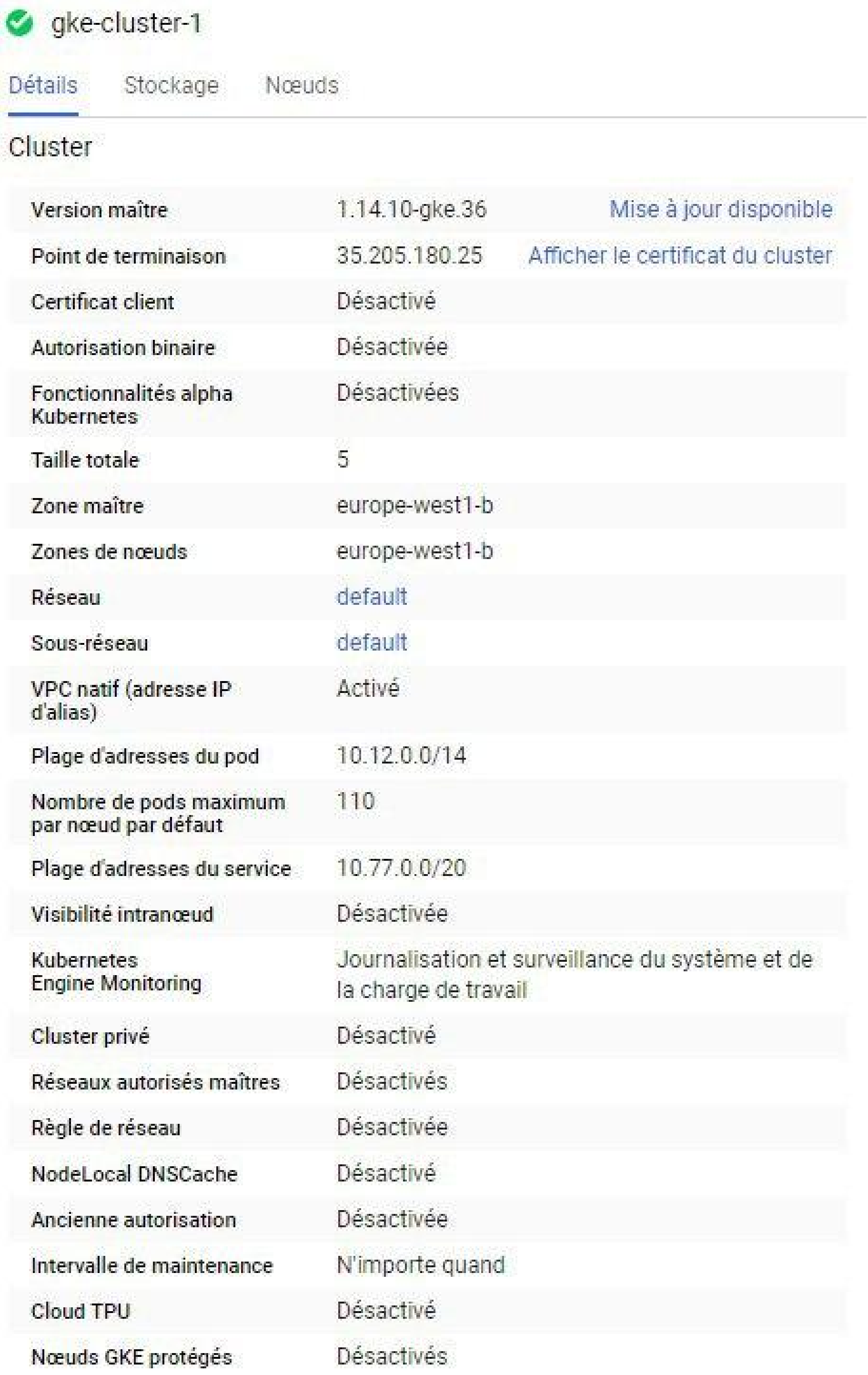
```
Conditions:
  Type              Status
  Initialized       True
  Ready             True
  ContainersReady   True
  PodScheduled      True
Volumes:
  default-token-qztlw:
    Type:        Secret (a volume populated by a Secret)
    SecretName:  default-token-qztlw
    Optional:    false
QoS Class:       Burstable
Node-Selectors:  <none>
Tolerations:     node.kubernetes.io/not-ready:NoExecute for 300s
                 node.kubernetes.io/unreachable:NoExecute for 300s
Events:
  Type    Reason     Age    From                                                 Message
  ----    ------     ----   ----                                                 -------
  Normal  Scheduled  5m36s  default-scheduler                                    Successfully assigned default/frontend to gke-gke-cluster-1-default-pool-8655fb8c-pn39
  Normal  Pulling    5m35s  kubelet, gke-gke-cluster-1-default-pool-8655fb8c-pn39  Pulling image "icclabcna/ccp2-k8s-todo-frontend"
  Normal  Pulled     5m34s  kubelet, gke-gke-cluster-1-default-pool-8655fb8c-pn39  Successfully pulled image "icclabcna/ccp2-k8s-todo-frontend"
  Normal  Created    5m34s  kubelet, gke-gke-cluster-1-default-pool-8655fb8c-pn39  Created container frontend
  Normal  Started    5m33s  kubelet, gke-gke-cluster-1-default-pool-8655fb8c-pn39  Started container frontend
```

  - Description of the service Frontend. This service has been created during this task:



```
(base) david@david-K501UQ:~/CLD/labs/03_Kubernetes$ kubectl describe svc/frontend-svc
Name:                     frontend-svc
Namespace:                default
Labels:                   component=frontend
Annotations:              <none>
Selector:                 app=todo,component=frontend
Type:                     LoadBalancer
IP:                       10.77.1.78
LoadBalancer Ingress:     35.195.209.109
Port:                     frontend  80/TCP
TargetPort:               8080/TCP
NodePort:                 frontend  30186/TCP
Endpoints:                10.12.4.6:8080
Session Affinity:         None
External Traffic Policy:  Cluster
Events:
  Type    Reason                Age   From                Message
  ----    ------                ----  ----                -------
  Normal  EnsuringLoadBalancer  97s   service-controller  Ensuring load balancer
  Normal  EnsuredLoadBalancer   58s   service-controller  Ensured load balancer
```

- Take a screenshot of the cluster details from the GKE console.
  - The screenshots below show the details from the GKE console. It is important to notice that the screenshots have been taken after we have done the task 3.



Cluster

Version maître	1.14.10-gke.36	Mise à jour disponible
Point de terminaison	35.205.180.25	Afficher le certificat du cluster
Certificat client	Désactivé	
Autorisation binaire	Désactivée	
Fonctionnalités alpha Kubernetes	Désactivées	
Taille totale	5	
Zone maître	europe-west1-b	
Zones de nœuds	europe-west1-b	
Réseau	default	
Sous-réseau	default	
VPC natif (adresse IP d'alias)	Activé	
Plage d'adresses du pod	10.12.0.0/14	
Nombre de pods maximum par nœud par défaut	110	
Plage d'adresses du service	10.77.0.0/20	
Visibilité intranœud	Désactivée	
Kubernetes Engine Monitoring	Journalisation et surveillance du système et de la charge de travail	
Cluster privé	Désactivé	
Réseaux autorisés maîtres	Désactivés	
Règle de réseau	Désactivée	
NodeLocal DNSCache	Désactivé	
Ancienne autorisation	Désactivée	
Intervalle de maintenance	N'importe quand	
Cloud TPU	Désactivé	
Nœuds GKE protégés	Désactivés	

Chiffrement des codes secrets au niveau de la couche d'application	Désactivé
Workload Identity	Désactivé
Google Groupes pour RBAC	Désactivée
Autoscaling des pods verticaux	Désactivé
Mesure de l'utilisation de GKE	Désactivée
Istio	Désactivé
Cloud Run pour Anthos	Désactivé

Provisionnement automatique des nœuds

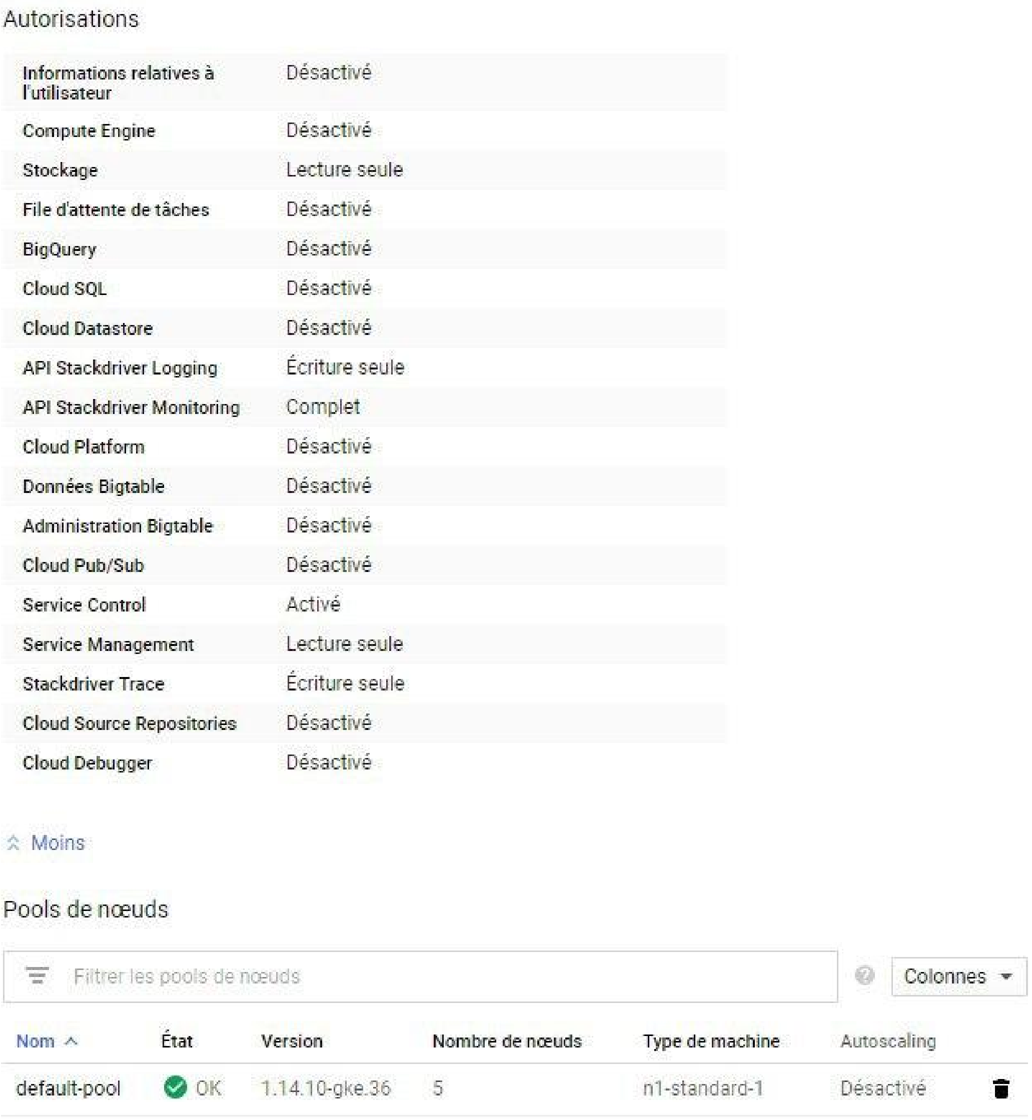
Désactivé

Libellés

Aucun

Modules complémentaires

Tableau de bord Kubernetes	Désactivé
Gestionnaire d'applications	Désactivé
Équilibrage de charge HTTP	Activé
Pilote CSI de disque persistant Compute Engine	Désactivé



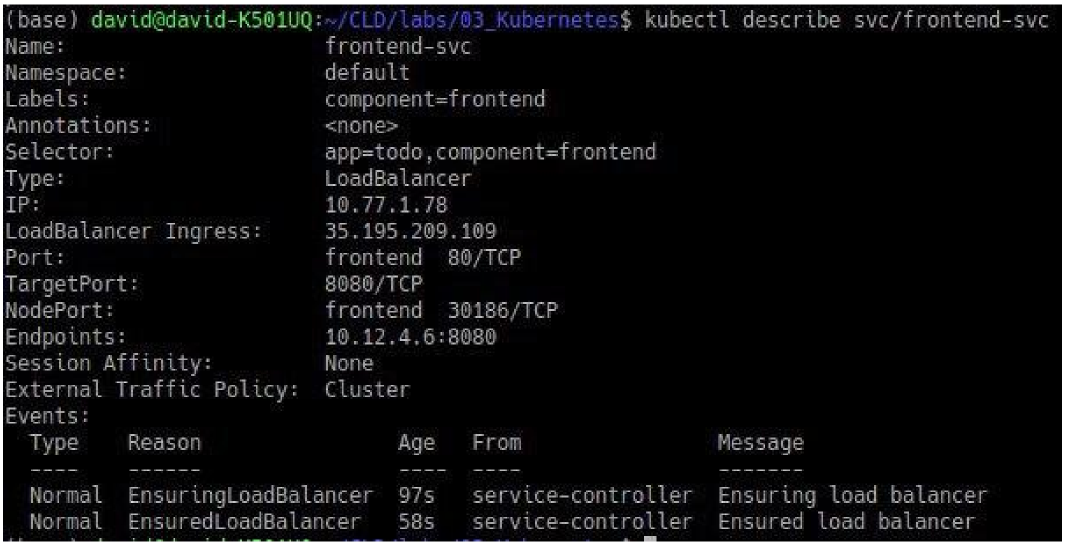
Autorisations

Informations relatives à l'utilisateur	Désactivé
Compute Engine	Désactivé
Stockage	Lecture seule
File d'attente de tâches	Désactivé
BigQuery	Désactivé
Cloud SQL	Désactivé
Cloud Datastore	Désactivé
API Stackdriver Logging	Écriture seule
API Stackdriver Monitoring	Complet
Cloud Platform	Désactivé
Données Bigtable	Désactivé
Administration Bigtable	Désactivé
Cloud Pub/Sub	Désactivé
Service Control	Activé
Service Management	Lecture seule
Stackdriver Trace	Écriture seule
Cloud Source Repositories	Désactivé
Cloud Debugger	Désactivé

Pools de nœuds

Nom	État	Version	Nombre de nœuds	Type de machine	Autoscaling
default-pool	✓ OK	1.14.10-gke.36	5	n1-standard-1	Désactivé

- Copy the output of the `kubectl describe` command to describe your load balancer once completely initialized.
  - It's located in the `svc-frontend`



```
(base) david@david-K501UQ:~/CLD/labs/03_Kubernetes$ kubectl describe svc/frontend-svc
Name:                     frontend-svc
Namespace:                default
Labels:                   component=frontend
Annotations:              <none>
Selector:                 app=todo,component=frontend
Type:                     LoadBalancer
IP:                       10.77.1.78
LoadBalancer Ingress:     35.195.209.109
Port:                     frontend  80/TCP
TargetPort:               8080/TCP
NodePort:                 frontend  30186/TCP
Endpoints:                10.12.4.6:8080
Session Affinity:         None
External Traffic Policy:  Cluster
Events:
  Type    Reason                Age   From                Message
  ----    ------                ----  ----                -------
  Normal  EnsuringLoadBalancer  97s   service-controller  Ensuring load balancer
  Normal  EnsuredLoadBalancer   58s   service-controller  Ensured load balancer
```

Task 3 - Deliverables

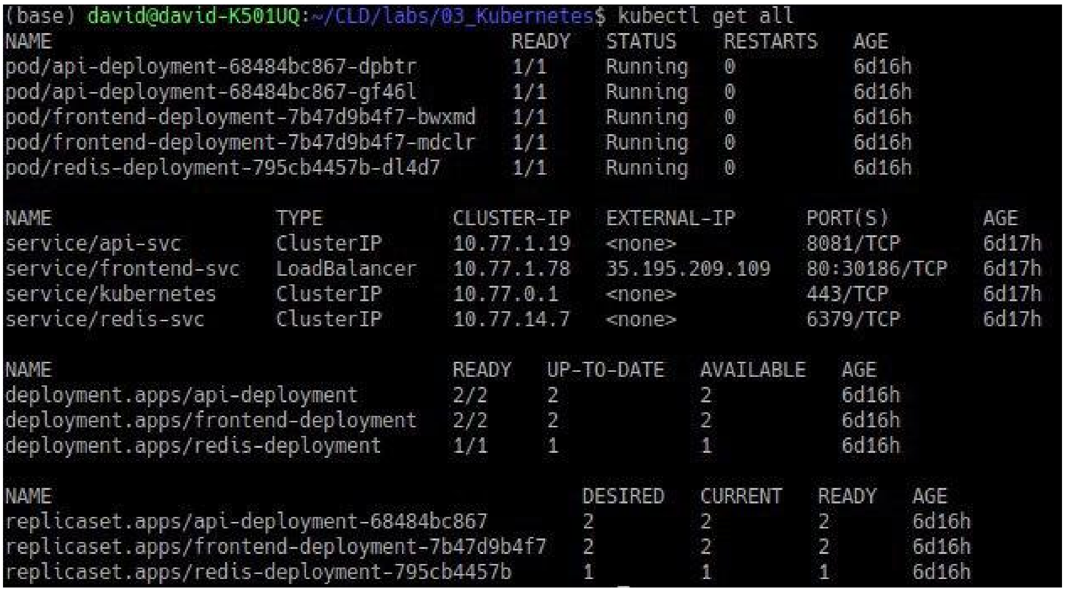
Subtask 3.1

- Use only 1 instance for the Redis-Server. Why?

- Redis is a master-slave architecture. The main Redis cache can set up muéltiple slaves to hold the data and act according to what the main instance would do. Since it seems we don't use a lot of data, we only have one instance of Redis server.

- Other observations/difficulties
  - The Kubernetes official documentation helped us a lot so we didn't encounter any difficulties
  - The Replica were all given IDs and they were created successfully

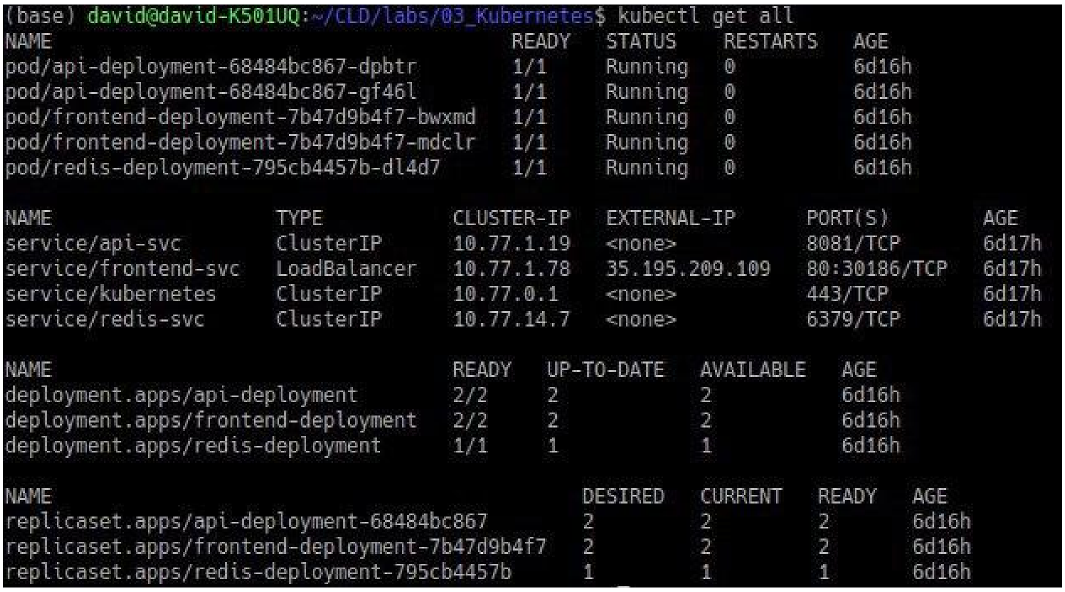
    kubectl get all



```
(base) david@david-K501UQ:~/CLD/labs/03_Kubernetes$ kubectl get all
NAME                                       READY   STATUS    RESTARTS   AGE
pod/api-deployment-68484bc867-dpbtr        1/1     Running   0          6d16h
pod/api-deployment-68484bc867-gf46l        1/1     Running   0          6d16h
pod/frontend-deployment-7b47d9b4f7-bwxmd   1/1     Running   0          6d16h
pod/frontend-deployment-7b47d9b4f7-mdclr   1/1     Running   0          6d16h
pod/redis-deployment-795cb4457b-dl4d7      1/1     Running   0          6d16h

NAME                   TYPE           CLUSTER-IP   EXTERNAL-IP      PORT(S)        AGE
service/api-svc        ClusterIP      10.77.1.19   <none>           8081/TCP       6d17h
service/frontend-svc   LoadBalancer   10.77.1.78   35.195.209.109   80:30186/TCP   6d17h
service/kubernetes     ClusterIP      10.77.0.1    <none>           443/TCP        6d17h
service/redis-svc      ClusterIP      10.77.14.7   <none>           6379/TCP       6d17h

NAME                                  READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/api-deployment        2/2     2            2           6d16h
deployment.apps/frontend-deployment   2/2     2            2           6d16h
deployment.apps/redis-deployment      1/1     1            1           6d16h

NAME                                             DESIRED   CURRENT   READY   AGE
replicaset.apps/api-deployment-68484bc867        2         2         2       6d16h
replicaset.apps/frontend-deployment-7b47d9b4f7   2         2         2       6d16h
replicaset.apps/redis-deployment-795cb4457b      1         1         1       6d16h
```

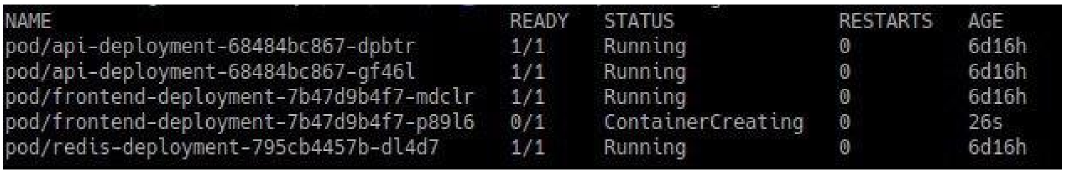
Subtask 3.2

- What happens if you delete a Frontend or API Pod? How long does it take for the system to react?
  - It deletes successfully and another replaces it immediatly. It took 30 seconds to launch the new one.
  - Command was `kubectl delete pod frontend-deployment-7b47d9b4f7-bwxmd`



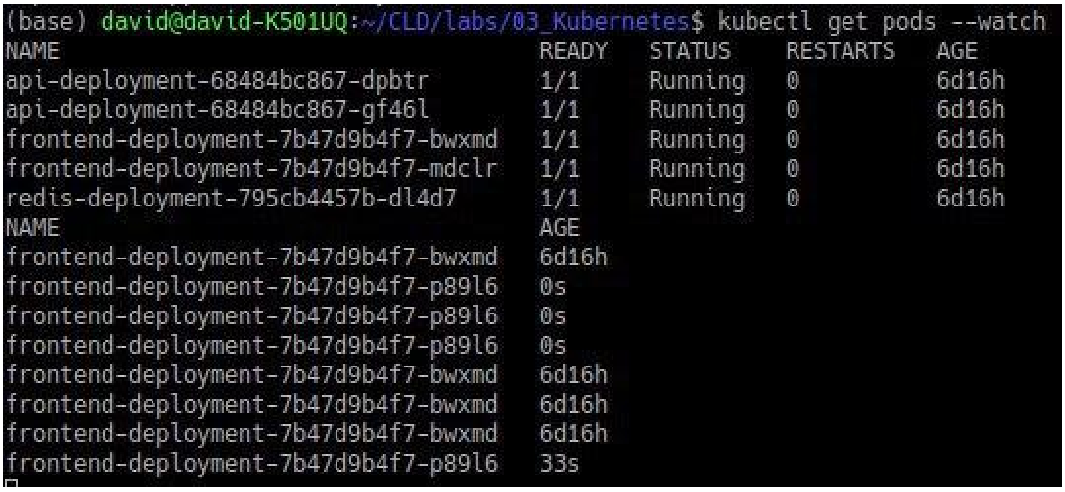
```
(base) david@david-K501UQ:~/CLD/labs/03_Kubernetes$ kubectl get all
NAME                                       READY   STATUS    RESTARTS   AGE
pod/api-deployment-68484bc867-dpbtr        1/1     Running   0          6d16h
pod/api-deployment-68484bc867-gf46l        1/1     Running   0          6d16h
pod/frontend-deployment-7b47d9b4f7-bwxmd   1/1     Running   0          6d16h
pod/frontend-deployment-7b47d9b4f7-mdclr   1/1     Running   0          6d16h
pod/redis-deployment-795cb4457b-dl4d7      1/1     Running   0          6d16h

NAME                   TYPE           CLUSTER-IP   EXTERNAL-IP      PORT(S)        AGE
service/api-svc        ClusterIP      10.77.1.19   <none>           8081/TCP       6d17h
service/frontend-svc   LoadBalancer   10.77.1.78   35.195.209.109   80:30186/TCP   6d17h
service/kubernetes     ClusterIP      10.77.0.1    <none>           443/TCP        6d17h
service/redis-svc      ClusterIP      10.77.14.7   <none>           6379/TCP       6d17h

NAME                                  READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/api-deployment        2/2     2            2           6d16h
deployment.apps/frontend-deployment   2/2     2            2           6d16h
deployment.apps/redis-deployment      1/1     1            1           6d16h

NAME                                             DESIRED   CURRENT   READY   AGE
replicaset.apps/api-deployment-68484bc867        2         2         2       6d16h
replicaset.apps/frontend-deployment-7b47d9b4f7   2         2         2       6d16h
replicaset.apps/redis-deployment-795cb4457b      1         1         1       6d16h
```

Before the delete



```
NAME                                        READY   STATUS              RESTARTS   AGE
pod/api-deployment-68484bc867-dpbtr         1/1     Running             0          6d16h
pod/api-deployment-68484bc867-gf46l         1/1     Running             0          6d16h
pod/frontend-deployment-7b47d9b4f7-mdclr    1/1     Running             0          6d16h
pod/frontend-deployment-7b47d9b4f7-p89l6    0/1     ContainerCreating   0          26s
pod/redis-deployment-795cb4457b-dl4d7       1/1     Running             0          6d16h
```

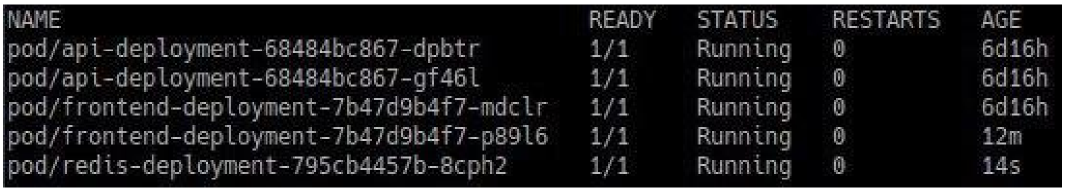
While the Pod is recreating



```
(base) david@david-K501UQ:~/CLD/labs/03_Kubernetes$ kubectl get pods --watch
NAME                                  READY   STATUS    RESTARTS   AGE
api-deployment-68484bc867-dpbtr       1/1     Running   0          6d16h
api-deployment-68484bc867-gf46l       1/1     Running   0          6d16h
frontend-deployment-7b47d9b4f7-bwxmd  1/1     Running   0          6d16h
frontend-deployment-7b47d9b4f7-mdclr  1/1     Running   0          6d16h
redis-deployment-795cb4457b-dl4d7     1/1     Running   0          6d16h
NAME                                  AGE
frontend-deployment-7b47d9b4f7-bwxmd  6d16h
frontend-deployment-7b47d9b4f7-p89l6  0s
frontend-deployment-7b47d9b4f7-p89l6  0s
frontend-deployment-7b47d9b4f7-p89l6  0s
frontend-deployment-7b47d9b4f7-bwxmd  6d16h
frontend-deployment-7b47d9b4f7-bwxmd  6d16h
frontend-deployment-7b47d9b4f7-bwxmd  6d16h
frontend-deployment-7b47d9b4f7-p89l6  33s
```

Running pods after the delete and recreation of the new Pod is complete

- What happens when you delete the Redis Pod?
  - It does the same as before but the creation of the new Redis Pod is much more faster. While after 30 seconds the pod was still being launch before, here after 14 seconds the new Redis Pod is already running



```
NAME                                        READY   STATUS    RESTARTS   AGE
pod/api-deployment-68484bc867-dpbtr         1/1     Running   0          6d16h
pod/api-deployment-68484bc867-gf46l         1/1     Running   0          6d16h
pod/frontend-deployment-7b47d9b4f7-mdclr    1/1     Running   0          6d16h
pod/frontend-deployment-7b47d9b4f7-p89l6    1/1     Running   0          12m
pod/redis-deployment-795cb4457b-8cph2       1/1     Running   0          14s
```

- How can you change the number of instances temporarily to 3? Hint: look for scaling in the deployment documentation

```
kubectl scale deployment.v1.apps/nginx-deployment --replicas=3
```

- What autoscaling features are available? Which metrics are used?
  - We can use [horizontal Pod autoscaling]() where we specify the minimum and maximum number of pods in our cluster and it is based on the CPU utilization of our existing Pods.
- How can you update a component? (see "Updating a Deployment" in the deployment documentation)
  - We can update what image a Deployment is using: `kubectl --record deployment.apps/nginx-deployment set image deployment.v1.apps/nginx-deployment nginx=nginx:1.16.1`
    - Here we told the nginx Pod to use image 1.16.1 instead of the image it was using
  - We can also edit the Deployment and change the image manually by typing `kubectl edit deployment.v1.apps/nginx-deployment`

Other Deliverables

- Document any difficulties you faced and how you overcame them.

  - No particular difficulties
- Copy the object descriptions into the lab report.
  - Every other Pod/Service are the same as they were previously on Task 2
  - Otherwise the objects descriptions are in the report under Task 3.2