

Kubernetes Notes:

🎯 أول إشي: شو هو ال Deployment؟

✅ تعريف مبسّط:

Deployment هو مثل "مدير ذكي" مسؤول عن تشغيل تطبيقك داخل Kubernetes، ويحرص على إنه:

- عدد النسخ (Pods) اللي طلبتها شغالة.
- النسخة الصحيحة من التطبيق مستخدمة.
- وإذا وقع واحد من ال Pods، يعيده تلقائيًا.
- وإذا غيرتي شي (زي الصورة أو العدد)، يعمل التحديث بطريقة ذكية وبدون توقف.

💡 طيب خلينا نرجع خطوة لورا:

كيف بيشتغل Kubernetes عادةً؟

- أنت بدك تشغلي تطبيق ويب مثلاً.
 - فيتنشني Pod يحوي الحاوية (container) اللي فيها تطبيقك.
 - لكن:
 - Pod واحد ممكن يوقع.
 - ما فيه تكرار (replication).
 - التحديثات لازم تعملها يدوي.
- وهنا بييجي ال Deployment علشان يحل كل هالمشاكل.

💡 تشبيه واقعي:

👤 ال Pod مثل "عامل" أو "موظف" واحد شغال.

🗨️ ال Deployment مثل "مدير الفريق"، مسؤوليته:

- يوظف العدد المطلوب.
- يطرد ويعيّن حسب الحاجة.
- يطور الفريق.
- يحتفظ بنسخة احتياطية من كل تغييراتك.

❓ ليش مش أشتغل دايمًا بـ Pod فقط؟

لأنه ال Pod:

- بيتوقف فجأة إذا صار خطأ.
- ما يعرف يتحدث.
- ما فيه نسخ متعددة.
- ما فيه سجل تغييرات.

لكن ال Deployment يعمل كل هاد عنك بشكل ذكي وآلي. ✨

Web App with Mongo Database

تشغيل تطبيق ويب (Web App) يستخدم قاعدة بيانات MongoDB، وكل الإعدادات رح تنعمل من العقدة الرئيسية (master node) فقط.

المطلوبة لتشغيل التطبيق Kubernetes هذا أمر لتنزيل مشروع جاهز فيه ملفات الـ

git clone <https://github.com/IbrahimTalaatAh/kubernetes.git>

شو اللي بيصير هون؟

- هو أداة لإدارة الأكواد والمشاريع : Git

- (GitHub) معناها "انسخ نسخة كاملة من مشروع موجود على الإنترنت `clone`".
- YAML فيه ملفات `kubernetes` بتنزل مجلد اسمه.

: لملفات اللي بتنزل

- `mongo-config.yaml`
- `mongo-secret.yaml`
- `mongo.yaml`
- `webapp.yaml`

`cd kubernetes/`

`kubectl apply -f mongo-config.yaml`

◆ **4. `kubectl apply -f mongo-config.yaml`**

🔧 هذا بيستخدم أداة Kubernetes (اللي اسمها `kubectl`) علشان يضيف إعدادات معينة تتعلق بـ MongoDB.

طيب، شو بيحتوي الملف `mongo-config.yaml` ؟

- ConfigMap: ملف بيخزن معلومات عادية (زي اسم الداتابيس، اسم المستخدم...).
- هاي المعلومات مش سرّية.

`kubectl apply -f mongo-secret.yaml`

◆ **5. `kubectl apply -f mongo-secret.yaml`**

🔒 هون بنعمل نفس الشي، بس مع ملف أسرار (Secrets).

شو يعني Secrets ؟

- معلومات حساسة مثل كلمات المرور وبيانات الدخول.

kubectl apply -f mongo.yaml

◆ 6. kubectl apply -f mongo.yaml

📦 هذا الملف هو المسؤول عن تشغيل قاعدة بيانات MongoDB نفسها على Kubernetes.

شو بيحتوي هذا الملف؟

- تعريف Deployment لتشغيل MongoDB.
- يربطه مع ال ConfigMap و ال Secret اللي عملناهم قبل شوي.
- ممكن يحتوي أيضًا على Service علشان نقدر نتواصل مع Mongo.

kubectl apply -f webapp.yaml

◆ 7. kubectl apply -f webapp.yaml

📦 هذا الملف بيشغل تطبيق الويب نفسه.

شو فيه؟

- Deployment لتطبيق الويب (مثلًا باستخدام Node.js أو Python أو أي لغة).
- Service من نوع `NodePort` تفتح بورت (30100) علشان نقدر نفتح التطبيق من المتصفح.

◆ "Open the webapp website from your host machine using any node IP with port 30100."

`http://<IP_of_any_node>:30100` → `http://192.168.1.110:30100`

Managing your deployment:

kubectl get deployments

`kubectl get deployments` 1.1

هذا الأمر يعرض كل ال Deployments في الكلاستر.

شو هو ال Deployment؟

هو الشي اللي بيقول لـ Kubernetes:

"شغّلي هذا التطبيق، وخليه شغال دائماً، وإذا واحد من ال Pods وقف، شغّله من جديد."

النتائج المتوقعة:

Edit ↗ Copy ↗

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
webapp-deployment	3/3	3	3	5m
mongo-deployment	1/1	1	1	5m

kubectl get pods

kubectl get pods 2.2 ✓

هذا يعرض كل الوحدات الصغيرة (Pods) التي بتشغل التطبيقات الفعالية.

شو هو ال Pod؟

هي أصغر وحدة تشغيل في Kubernetes، وتشغل container (مثلاً: تطبيق ويب، MongoDB ... إلخ).

النتائج:

					Edit ↗	Copy 📄
NAME	READY	STATUS	RESTARTS	AGE		
webapp-deployment-xxxxx	1/1	Running	0	3m		
mongo-deployment-yyyyy	1/1	Running	0	3m		

kubectl get services

kubectl get services 3.3 ✓

هذا الأمر يعرض كل الخدمات (Services) التي بتوصل بين التطبيقات وبعضها أو بتفتحها للعالم الخارجي.

شو يعني Service؟

هو مثل "موزّع" يحدد كيف الناس توصل لتطبيقك.

النتائج:

						Edit ↗	Copy 📄
NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE		
webapp-service	NodePort	10.0.0.123	<none>	80:30100/TCP	3m		
mongo-service	ClusterIP	10.0.0.124	<none>	27017/TCP	3m		

kubectl describe <type> <type name>

<type> could be deployment, service, pod, ..

هذا يعطيك تفاصيل شاملة عن أي كائن بالكلاستر

الأمثلة:

- `kubectl describe deployment webapp-deployment`
- `kubectl describe pod mongo-deployment-xxxx`
- `kubectl describe service webapp-service`

Edit your deployment to get 6 webapp pods, use the **kubectl edit deployment webapp-deployment** command for that.

:Edit your deployment to get 6 webapp pods

Edit Copy

```
kubectl edit deployment webapp-deployment
```

هذا الأمر يفتح لك ملف ال Deployment في محرر نصي مباشر (غالبًا nano أو vi).

شو تعمل في فيه؟

1. دوري على السطر اللي فيه:

Edit Copy

```
replicas: 3
```

2. وغيّره لـ:

Edit Copy

```
replicas: 6
```

3. احفظي واخرجي من المحرر.

Done ↓

النتيجة: Kubernetes رح يزيد عدد ال pods لتطبيق الويب لـ 6 بدل 3.

Check how many pods are running in the cluster

kubectl get pods

Remove all deployments and services

```
kubectl delete deployment webapp-deployment
```

```
kubectl delete deployment mongo-deployment
```

حذف services:

```
kubectl delete service webapp-service
```

```
kubectl delete service mongo-service
```

Open `mongo.yaml` and change the replicas to 4 and do the same for `webapp.yaml` setting the replicas to 10 and redeploy mongo then webapp.

```
nano mongo.yaml
```

Find the line: replicas: 1 and change it to: replicas: 4

Same thing for `webapp.yaml` —→ Open it, change it to replicas: 10

Then restart them:

```
kubectl apply -f mongo.yaml
```

```
kubectl apply -f webapp.yaml
```

Get running pods with the `-o wide` option —→ `-o wide`

ex: `kubectl get pods -o wide`

What is the difference between

```
kubectl edit deployment webapp-deployment
```


and

Open mongo.yaml and change the replicas to 4 and do the same for webapp.yaml setting the replicas to 10 and redeploy mongo then webapp.

الخيار الأول (تعديل مباشر بالكلاستر):

Edit Copy

```
kubectl edit deployment webapp-deployment
```

- تفتحي المحرر
- تغيري replicas: 3 إلى replicas: 6
- تحفظي ☒ النتيجة: Kubernetes يعدّل النسخ فوراً.

الخيار الثاني (تعديل يدوي في الملف yaml):

webapp.yaml تفتحي الملف

- تغيري:

Edit Copy

```
replicas: 10
```

- تحفظي الملف
- تعملي:

Edit Copy

```
kubectl apply -f webapp.yaml
```

☒ النتيجة: سيتم تحديث ال deployment حسب الملف. Done

Another way to edit your deployment is using the following commands:

```
kubectl scale deployment/webapp-deployment --replicas=5
```

```
kubectl scale deployment/webapp-deployment --replicas=5 2. ✓
```

✂ هذا الأمر مسؤول عن:

تغيير عدد النسخ (replicas) لتطبيق معين (هنا webapp-deployment) إلى 5 نسخ.

🧠 شرح تفصيلي:

- `kubectl`: الأداة المسؤولة عن التحكم بكلاستر Kubernetes.
 - `scale`: يعني "غيّر الحجم" — الحجم هنا هو عدد النسخ من التطبيق (ال Pods).
 - `deployment/webapp-deployment`: هذا هو اسم ال Deployment اللي بدنا نعدله.
 - `replicas=5--`: معناها "خلي عدد ال Pods يكون 5".
- 🏠 النتيجة: Kubernetes بيزيد أو يقلل عدد ال Pods عشان يوصلوا إلى 5.
- ✓ إذا كانوا 3، بيضيف 2. ✓ إذا كانوا 6، بيحذف 1. ✓ إذا ما كان في أي Pods شغالة، بيبدأ يشغل 5 جديدة.

```
kubectl scale deployment/mongo-deployment --replicas=2
```

show running pods after that:

```
kubectl get pods
```

```
kubectl get rs
```

kubectl get rs 5.5 ✓

هذا الأمر يعرض ReplicaSets الموجودة.

شو هو ال ReplicaSet؟

- هو الكائن اللي يحافظ على عدد النسخ المطلوب من ال Pods في ال deployment.
- لما تعملي scale, ال ReplicaSet بيتغير داخليًا لي مطابق العدد الجديد.

النتائج المتوقعة:

Edit Copy				
NAME	DESIRED	CURRENT	READY	AGE
webapp-deployment-xxx	5	5	5	2m
mongo-deployment-yyy	2	2	2	2m


ملاحظة مهمة:

- DESIRED : عدد النسخ اللي طلبتها.
- CURRENT : عدد النسخ اللي فعليًا شغالة.
- إذا في فرق، معناته فيه Pods عم تشتغل أو تنحذف.
- لازم الاثنين يتطابقوا غالبًا خلال ثواني.

Done ↓

Update Mongo to version 5.0.14 from 5.0

kubectl set image deployment/mongo-deployment mongodb=mongo:5.0.14

```
kubectl set image deployment/mongo-deployment 7   
mongodb=mongo:5.0.14
```

🔧 هذا أمر بحدث الصورة (image) التي ال deployment يستخدمها.


شرح تفصيلي:

- `kubectl set image`: يعني "غير صورة ال container"
 - Deployment هو اسم ال : `deployment/mongo-deployment`
 - : `mongodb=mongo:5.0.14`
 - `mongodb` هو اسم ال container داخل ال pod.
 - `mongo:5.0.14` هو اسم الصورة الجديدة (من DockerHub غالبًا).
- 🎯 الهدف: Kubernetes يعمل **rolling update**: يعني وحدة وحدة. يوقف ال pod القديم، ويشغل جديد بالصورة الجديدة.

Check the status using the commands:

```
kubectl rollout status deployment/mongo-deployment
```

:Check the status using 8

```
kubectl rollout status deployment/mongo-deployment 
```

🔍 هذا الأمر يعطيك حالة عملية التحديث.

- رح يضل يطبعلك رسالة مثل:

```
Waiting for deployment "mongo-deployment" rollout to finish: 1 old replicas are pending terminati
```

- ولما يخلص:

```
deployment "mongo-deployment" successfully rolled out
```



`kubectl describe deployments` notice the mongo version in the output. You can also rollback and return to a previous version of the container image. Let's rollback mongo to version 5.0 by editing the deployment using the command `kubectl edit deployment mongo-deployment` and at the same time increasing the number of pods by 2. run `kubectl get pods` and `kubectl get rs` to see things in action (be fast).

Rollback Mongo to version 5.0 and increase replicas by 2 ← BACK

الهدف: 🎯

- نرجع Mongo لإصدار 5.0 (رجوع للخلف = rollback)
- ونزود عدد النسخ بـ 2

`kubectl edit deployment mongo-deployment` ✓

هذا يعرض ملف ال deployment بمحرر مباشر. ➡

- دوري على `image` وعدليها من:

Edit ✎ Copy 📄

```
image: mongo:5.0.14
```

إلى:

Edit ✎ Copy 📄

```
image: mongo:5.0
```

- وعدلي `replicas` ، زيديها مثلاً من 2 إلى 4.

To check the history of rollouts use the command:

`kubectl rollout history deployment/mongo-deployment`

deployment. هذا الأمر يعرض تاريخ كل التحديثات التي عملتها على هذا الـ

you can check the details of a specific revision using the command:

`kubectl rollout history deployment/mongo-deployment --revision=2`