

Version - **2024.01**

Dernière mise-à-jour : 2024/12/20 15:20

DOF309 - Gestion de Paquets pour Kubernetes avec Helm

Contenu du Module

- **DOF309 - Gestion de Paquets pour Kubernetes avec Helm**

- Contenu du Module
 - Présentation de Helm
 - LAB #1 - Travailler avec Helm
 - 1.1 - Installation de Helm
 - 1.2 - La Commande helm search hub
 - 1.3 - Rechercher dans le Helm Hub
 - 1.4 - Ajouter et supprimer un Dépôt
 - 1.5 - La Commande helm search repo
 - 1.6 - La Commande helm show
 - 1.7 - Installation d'un Chart
 - 1.8 - La Commande helm get
 - 1.9 - Utilisation des NOTES
 - 1.10 - La Commande helm upgrade
 - 1.11 - La Commande helm history
 - 1.12 - La Commande helm rollback
 - 1.13 - La Commande helm uninstall
 - LAB #2 - Supervision de Kubernetes avec le Stack EFK
 - 2.1 - Présentation du Stack EFK
 - 2.2 - Installation du Chart elasticsearch
 - 2.3 - Installation du Chart fluentd-elasticsearch
 - 2.4 - Installation du Chart kibana
 - 2.5 - Génération de Traces dans Kubernetes
-

- 2.6 - Visualisation des Données avec Kibana

Ressources

Lab #1

- <https://www.dropbox.com/scl/fi/mw6afaitlz26tvt31dw33/wordpress-values.yaml?rlkey=eii2mhu99bo5ita8whf5e1ihq&dl=0>

Lab #2

- <https://www.dropbox.com/scl/fi/rmgz23mutsbvxe5qk8099/kibana-values.yaml?rlkey=rqq9zelrjjw9mkf7lefh9ohi&dl=0>
- <https://www.dropbox.com/scl/fi/3tiwgd80xpi387g0mdlc7/counter.yaml?rlkey=pphutka93v5mlikgicrzc6p9&dl=0>

Présentation de Helm

Helm est un gestionnaire de paquets pour Kubernetes. Helm 3.x est compatible avec les versions **n-3** de Kubernetes où **n** est la version de Kubernetes contre laquelle la version de Helm a été compilée, tandis que le gestionnaire Helm 2.x n'est compatible qu'avec les versions **n-1** de Kubernetes où **n** est la version de Kubernetes contre laquelle la version de Helm a été compilée.

LAB #1 - Travailler avec Helm

Placez-vous dans votre Gateway. Commencez par vérifier le status de Minikube :

```
trainee@gateway:~$ minikube status
minikube
type: Control Plane
host: Running
```

```
kubelet: Running
apiserver: Running
kubeconfig: Configured
```

1.1 - Installation de Helm

Installez **helm** :

```
trainee@gateway:~$ su -
Password:

root@gateway:~# curl https://baltocdn.com/helm/signing.asc | gpg --dearmor | sudo tee
/usr/share/keyrings/helm.gpg > /dev/null
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100  1699  100  1699    0     0  17161      0  --:--:--  --:--:--  --:--:-- 17336

root@gateway:~# echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/helm.gpg]
https://baltocdn.com/helm/stable/debian/ all main" | sudo tee /etc/apt/sources.list.d/helm-stable-debian.list
deb [arch=amd64 signed-by=/usr/share/keyrings/helm.gpg] https://baltocdn.com/helm/stable/debian/ all main

root@gateway:~# apt-get update
...

root@gateway:~# apt-get install helm
...
```

Vérifiez ensuite votre travail :

```
root@gateway:~# which helm
/usr/sbin/helm

root@gateway:~# helm version
```

```
version.BuildInfo{Version:"v3.16.3", GitCommit:"cfd07493f46efc9debd9cc1b02a0961186df7fdf", GitTreeState:"clean", GoVersion:"go1.22.7"}
```

Pour permettre l'auto-complétion de la ligne de commande avec la commande `helm`, similaire à celle sous `bash`, exécutez la commande suivante en tant que trainee :

```
root@gateway:~# exit
logout
trainee@gateway:~$

trainee@gateway:~$ source <(helm completion bash)
```

Pour visualiser les variables d'environnement de la commande **helm**, exécutez la commande **helm env** :

```
trainee@gateway:~$ helm env
HELM_BIN="helm"
HELM_BURST_LIMIT="100"
HELM_CACHE_HOME="/home/trainee/.cache/helm"
HELM_CONFIG_HOME="/home/trainee/.config/helm"
HELM_DATA_HOME="/home/trainee/.local/share/helm"
HELM_DEBUG="false"
HELM_KUBEAPISERVER=""
HELM_KUBEASGROUPS=""
HELM_KUBEASUSER=""
HELM_KUBECAFILE=""
HELM_KUBECONTEXT=""
HELM_KUBEINSECURE_SKIP_TLS_VERIFY="false"
HELM_KUBETLS_SERVER_NAME=""
HELM_KUBETOKEN=""
HELM_MAX_HISTORY="10"
HELM_NAMESPACE="default"
HELM_PLUGINS="/home/trainee/.local/share/helm/plugins"
HELM_QPS="0.00"
HELM_REGISTRY_CONFIG="/home/trainee/.config/helm/registry/config.json"
```

```
HELM_REPOSITORY_CACHE="/home/trainee/.cache/helm/repository"  
HELM_REPOSITORY_CONFIG="/home/trainee/.config/helm/repositories.yaml"
```



Important : Notez que la valeur de la variable **\$HELM_NAMESPACE** est **default**. Pour modifier le Namespace utilisé lors de l'utilisation de la commande helm, vous pouvez soit modifier la valeur de cette variable, soit utiliser l'option **-namespace** ou **-n** lors de chaque exécution de la commande.

Pour visualiser l'environnement de la commande **kubectl** exécutez la commande **kubectl config view** :

```
trainee@gateway:~$ kubectl config view  
apiVersion: v1  
clusters:  
- cluster:  
  certificate-authority-data: DATA+OMITTED  
  server: https://192.168.42.99:8443  
  name: 192-168-42-99:8443  
- cluster:  
  certificate-authority: /home/trainee/.minikube/ca.crt  
  extensions:  
  - extension:  
    last-update: Fri, 20 Dec 2024 14:48:36 CET  
    provider: minikube.sigs.k8s.io  
    version: v1.34.0  
    name: cluster_info  
  server: https://192.168.59.100:8443  
  name: minikube  
contexts:  
- context:  
  cluster: minikube  
  extensions:
```

```
- extension:
  last-update: Fri, 20 Dec 2024 14:48:36 CET
  provider: minikube.sigs.k8s.io
  version: v1.34.0
  name: context_info
  namespace: default
  user: minikube
name: minikube
- context:
  cluster: 192-168-42-99:8443
  namespace: myproject
  user: developer/192-168-42-99:8443
name: minishift
- context:
  cluster: 192-168-42-99:8443
  namespace: myproject
  user: developer/192-168-42-99:8443
name: myproject/192-168-42-99:8443/developer
- context:
  cluster: 192-168-42-99:8443
  namespace: myproject
  user: system:admin/192-168-42-99:8443
name: myproject/192-168-42-99:8443/system:admin
current-context: minikube
kind: Config
preferences: {}
users:
- name: developer/192-168-42-99:8443
  user:
    token: REDACTED
- name: minikube
  user:
    client-certificate: /home/trainee/.minikube/profiles/minikube/client.crt
    client-key: /home/trainee/.minikube/profiles/minikube/client.key
```

```
- name: system:admin/192-168-42-99:8443
  user:
    client-certificate-data: DATA+OMITTED
    client-key-data: DATA+OMITTED
```

1.2 - La Commande helm search hub

Un grand nombre de Charts Helm sont regroupés sur le site [The Artifact Hub](#). Il est possible de lister le contenu du Hub en ligne de commande :

```
trainee@gateway:~$ helm search hub | more
URL                                     CHART VERSION
APP VERSION
DESCRIPTION
https://artifacthub.io/packages/helm/mya/12factor 24.1.2
  Easily deploy any application that conforms to ...
https://artifacthub.io/packages/helm/gabibbo97/... 0.1.0
fedora-32
  389 Directory Server
https://artifacthub.io/packages/helm/four-allpo... 20.3.4
3.10.38
  A Helm chart for 4ALLPORTAL version 3.10.0 and up
https://artifacthub.io/packages/helm/helm-7dtd/... 0.1.0
0.4.4
  7 Days to Die is an open-world game that is a u...
https://artifacthub.io/packages/helm/openshift/... 0.2.0
1.16.0
  A Helm chart for A10 Thunder Kubernetes Connector
https://artifacthub.io/packages/helm/aad-pod-id... 4.1.18
1.8.17
  Deploy components for aad-pod-identity
https://artifacthub.io/packages/helm/smo-helm-c... 6.0.0
  ONAP Application Authorization Framework
https://artifacthub.io/packages/helm/smo-helm-c... 6.0.0
```

```

    ONAP Active and Available Inventory
https://artifacthub.io/packages/helm/statcan/aa...    0.1.7
1.0
    A Helm chart for AAW Financial Data Reconciliation
https://artifacthub.io/packages/helm/arhatdev/a...    0.1.0
latest
    Network Manager Living at Edge
https://artifacthub.io/packages/helm/apfelplant...    0.1.0
1.16.0
--More--

```

Recherchez ensuite tout Chart contenant la chaîne **mysql** :

```

trainee@gateway:~$ helm search hub mysql
URL                                CHART VERSION  APP VERSION    DESCRIPTION
https://artifacthub.io/packages/helm/mysql/mysql    2.1.3          8.0.26         deploy mysql
standalone or group-replication He...
https://artifacthub.io/packages/helm/kubegems/m...   8.9.6          8.0.29         MySQL is a fast,
reliable, scalable, and easy t...
https://artifacthub.io/packages/helm/bitnami-ak...   9.4.3          8.0.31         MySQL is a fast,
reliable, scalable, and easy t...
https://artifacthub.io/packages/helm/dify-tidb/...   11.1.17        8.4.2          MySQL is a fast,
reliable, scalable, and easy t...
https://artifacthub.io/packages/helm/kubesphere...   1.0.2          5.7.33        High Availability
MySQL Cluster, Open Source.
https://artifacthub.io/packages/helm/ygqygq2/mysql   9.5.0          8.0.32         MySQL is a fast,
reliable, scalable, and easy t...
https://artifacthub.io/packages/helm/cloudnativ...   5.0.1          8.0.16         Chart to create a
Highly available MySQL cluster
https://artifacthub.io/packages/helm/warjiang/m...   9.10.5         8.0.33         MySQL is a fast,
reliable, scalable, and easy t...
https://artifacthub.io/packages/helm/stakater/m...   1.0.6          mysql chart that
runs on kubernetes
https://artifacthub.io/packages/helm/choerodon/...   8.5.1          8.5.1          Chart to create a

```

Highly available MySQL cluster https://artifacthub.io/packages/helm/saber/mysql	8.8.21	8.0.27	Chart to create a
Highly available MySQL cluster https://artifacthub.io/packages/helm/groundhog2...	3.0.1	8.4.3	A Helm chart for
MySQL on Kubernetes https://artifacthub.io/packages/helm/bitnami/mysql	12.2.0	8.4.3	MySQL is a fast, reliable, scalable, and easy t...
https://artifacthub.io/packages/helm/kvalitetsi...	9.10.4	8.0.33	MySQL is a fast, reliable, scalable, and easy t...
https://artifacthub.io/packages/helm/druid-helm...	1.6.4	5.7.30	Fast, reliable, scalable, and easy to use open-...
https://artifacthub.io/packages/helm/devopspilo...	1.0.0	1.0.0	Helm chart for running mysql database
https://artifacthub.io/packages/helm/gengxianku...	0.2.0	1.16.0	MySQL is an open- source relational database man...
https://artifacthub.io/packages/helm/t3n/mysql	1.0.0	5.7.32	Fast, reliable, scalable, and easy to use open-...
https://artifacthub.io/packages/helm/helmcourse...	0.2.4	5.7	MySQL packaged in Rajware Helm course
https://artifacthub.io/packages/helm/ot-contain...	0.1.0	1.16.0	A Helm chart for Kubernetes to deploy mysql.
https://artifacthub.io/packages/helm/kubeblocks...	1.0.0-alpha.0	5.7.44	MySQL is a widely used, open-source relational ...
https://artifacthub.io/packages/helm/someblackm...	1.7.1	5.7.33	Fast, reliable, scalable, and easy to use open-...
https://artifacthub.io/packages/helm/kubesphere...	1.6.8	5.7.31	Fast, reliable, scalable, and easy to use open-...
https://artifacthub.io/packages/helm/kubegemsap...	4.5.2	5.7.26	Chart to create a Highly available MySQL cluster
https://artifacthub.io/packages/helm/wso2/mysql	1.6.9	5.7.30	Fast, reliable, scalable, and easy to use open-...
https://artifacthub.io/packages/helm/homeenterp...	0.5.0	8.0.29	MySQL Database Server
https://artifacthub.io/packages/helm/helm-chart...	1.0.0	5.7.26	Chart to create a

Highly available MySQL cluster			
https://artifacthub.io/packages/helm/anvibo/mysql	0.1.0	1.16.0	A Helm chart for Kubernetes
https://artifacthub.io/packages/helm/huangcheng...	0.1.0	1.16.0	A Helm chart for Kubernetes
https://artifacthub.io/packages/helm/mysql-oper...	2.2.2	9.1.0	MySQL InnoDB Cluster Helm Chart for deploying M...
https://artifacthub.io/packages/helm/mysql-oper...	2.2.2	9.1.0-2.2.2	MySQL Operator Helm Chart for deploying MySQL I...
https://artifacthub.io/packages/helm/softonic/m...	2.2.3	0.3.0	Take mysql backups from any mysql instance to A...
https://artifacthub.io/packages/helm/kfirfer/my...	0.0.4	8.0.20	A Helm chart for MySQL Cluster
https://artifacthub.io/packages/helm/wordpress-...	1.0.0	1.1	This is a package for configuring wordpress and...
https://artifacthub.io/packages/helm/kubeblocks...	1.0.0-alpha.0	8.0.30	ApeCloud MySQL is a database that is compatible...
https://artifacthub.io/packages/helm/kubeblocks...	1.0.0-alpha.0	8.0.33	MySQL is a widely used, open-source relational ...
https://artifacthub.io/packages/helm/bitpoke/my...	0.6.3	v0.6.3	A Helm chart for easy deployment of a MySQL clu...
https://artifacthub.io/packages/helm/bitpoke/my...	0.6.3	v0.6.3	A helm chart for Bitpoke Operator for MySQL
https://artifacthub.io/packages/helm/stakater/m...	1.0.6		mysql-storage chart that runs on kubernetes
https://artifacthub.io/packages/helm/kubesphere...	1.0.1	5.7.33	High Availability MySQL Cluster, Open Source.
https://artifacthub.io/packages/helm/appscode/s...	8.0.21	8.0.21	stash-mysql - MySQL database plugin for Stash b...
https://artifacthub.io/packages/helm/wordpressm...	1.0.0		This is the Helm Chart that creates the Wordpre...
https://artifacthub.io/packages/helm/kubeblocks...	0.5.3	8.0.30	ApeCloud MySQL Scale is ApeCloud MySQL proxy. A...
https://artifacthub.io/packages/helm/kubeblocks...	1.0.0-alpha.0	8.0.30	ApeCloud MySQL is

a database that is compatible...			
https://artifacthub.io/packages/helm/kube-wordp...	0.1.0	0.0.1-alpha	Helm Chart for
Wordpress installation on MySQL ...			
https://artifacthub.io/packages/helm/debeando/a...	0.10.0	0.6.0	A Helm chart for
DeBeAndo Agent for MySQL			
https://artifacthub.io/packages/helm/appscode/m...	2024.12.18	v0.3.0	A Helm chart for
MySQL Alert by AppsCode			
https://artifacthub.io/packages/helm/wso2/mysql-am	4.2.0-1	5.7	A Helm chart for
MySQL based deployment of WS02...			
https://artifacthub.io/packages/helm/choerodon/...	1.0.0	5.7	mysql client
https://artifacthub.io/packages/helm/kubesphere...	0.5.6	v0.11.0	A Helm chart for
prometheus mysql exporter with...			
https://artifacthub.io/packages/helm/openshift/...	0.0.1	0.0.1	This content is
expermental, do not use it in p...			
https://artifacthub.io/packages/helm/wso2is-pat...	5.11.0-2	5.7	A Helm chart for
MySQL based deployment of WS02...			
https://artifacthub.io/packages/helm/wso2/mysql-is	5.11.0-5	5.7	A Helm chart for
MySQL based deployment of WS02...			
https://artifacthub.io/packages/helm/wso2/mysql-ob	1.5.0-1	5.7	A Helm chart for
MySQL based deployment of WS02...			
https://artifacthub.io/packages/helm/banzaiclou...	0.1.0	0.2.0	A Helm chart for
deploying the Oracle MySQL Ope...			
https://artifacthub.io/packages/helm/grdscloud/...	0.1.0	1.0.0	A MySQL Database
Operator Helm chart for Kubern...			
https://artifacthub.io/packages/helm/openshift/...	0.0.4	0.0.4	This content is
expermental, do not use it in p...			
https://artifacthub.io/packages/helm/openshift/...	0.1.0	0.1.0	A Helm chart for
MySQL Service Endpoint Definit...			
https://artifacthub.io/packages/helm/ygqygq2/my...	1.0.0	5.7.24	A Helm chart
servicemonitor for mysql exporter			
https://artifacthub.io/packages/helm/kubeblocks...	0.9.0	8.0.32	MySQL is a widely
used, open-source relational ...			
https://artifacthub.io/packages/helm/kubeblocks...	0.8.0	8.0.32	MySQL is a widely
used, open-source relational ...			

https://artifacthub.io/packages/helm/wener/prometheus-mysql-exporter-with-...	2.8.0	v0.16.0	A Helm chart for
https://artifacthub.io/packages/helm/banzaicloud/prometheus-mysql-exporter-with-...	0.2.4	v0.11.0	A Helm chart for
https://artifacthub.io/packages/helm/prometheus-mysql-exporter-with-...	1.0.1	v0.12.1	A Helm chart for
https://artifacthub.io/packages/helm/prometheus-mysql-exporter-with-...	2.8.0	v0.16.0	A Helm chart for
https://artifacthub.io/packages/helm/kubegemsap/prometheus-mysql-exporter-with-...	1.8.1	v0.12.1	A Helm chart for
https://artifacthub.io/packages/helm/cloudnative/prometheus-mysql-exporter-with-...	0.3.2	v0.11.0	A Helm chart for
https://artifacthub.io/packages/helm/braedon/prometheus-mysql-exporter-with-...	0.2.0		Run queries on
https://artifacthub.io/packages/helm/wernerme/prometheus-mysql-exporter-with-...	2.8.0	v0.16.0	A Helm chart for
https://artifacthub.io/packages/helm/openshift/prometheus-mysql-exporter-with-...	0.0.2	0.0.2	This content is
https://artifacthub.io/packages/helm/bytebuilders/prometheus-mysql-exporter-with-...	2021.3.11	v2021.03.11	Stash MySQL Addon
https://artifacthub.io/packages/helm/fermosit/zabbix-server	3.0.2	6.4.0-latest	Zabbix monitoring
https://artifacthub.io/packages/helm/kubeblocks/mysql-scale-cluster	0.5.3	8.0.30	An ApeCloud
https://artifacthub.io/packages/helm/clusterpedia/clusterpedia	0.1.1	v0.6.0	A Helm chart for
https://artifacthub.io/packages/helm/t3n/mysql-exporter	2.1.0		
https://artifacthub.io/packages/helm/kfirfer/mysql-exporter	0.3.0	0.0.4	A Helm chart for
https://artifacthub.io/packages/helm/wso2/mysql-enterprise-integrator	6.6.0-3	5.7	A Helm chart for
https://artifacthub.io/packages/helm/sikalabs/wso2-enterprise-integrator	0.1.2		
https://artifacthub.io/packages/helm/choerodon/mysql-exporter	0.1.0	1.0	A Helm chart for
https://artifacthub.io/packages/helm/choerodon/mysql-exporter			Kubernetes

https://artifacthub.io/packages/helm/phppgadmin... Kubernetes	0.1.0		A Helm chart for
https://artifacthub.io/packages/helm/azure-samp... Kubernetes	0.1.0		A Helm chart for
https://artifacthub.io/packages/helm/mysql-ndb-... deploying the Oracle MySQL NDB...	1.6.0	9.1.0-1.6.0	A Helm chart for
https://artifacthub.io/packages/helm/sonu-wordp... chart to deploy wordpress and...	1.0.0	2	This is my custom
https://artifacthub.io/packages/helm/bitnami-ak... free software tool written in P...	10.3.6	5.2.0	phpMyAdmin is a
https://artifacthub.io/packages/helm/bitnami/ph... free software tool written in P...	18.1.0	5.2.1	phpMyAdmin is a
https://artifacthub.io/packages/helm/paradeum-t... full-featured database management ...	0.1.9	4.8.1	Adminer is a
https://artifacthub.io/packages/helm/cetic/adminer full-featured database management ...	0.2.1	4.8.1	Adminer is a
https://artifacthub.io/packages/helm/startechni... full-featured database management ...	0.1.8	4.8.1	Adminer is a
https://artifacthub.io/packages/helm/inspur/dble scalability middle-ware for MySQL...	0.0.2	2.19.09	DBLE is a high
https://artifacthub.io/packages/helm/wordpress-... chart to deploy wordpress and...	1.0.0	2	This is my custom
https://artifacthub.io/packages/helm/cloudnativ... help backup MySQL databases usi...	2.4.0	2.4.0	A Helm chart to
https://artifacthub.io/packages/helm/kfirfer/my... help backup MySQL databases usi...	2.8.0	1.0.3	A Helm chart to
https://artifacthub.io/packages/helm/kokuwa/mys... help backup MySQL databases usi...	6.4.0	v3.2.0	A Helm chart to
https://artifacthub.io/packages/helm/cloudnativ... with a single master and zero or ...	0.5.1	5.7.13	MySQL cluster
https://artifacthub.io/packages/helm/cloudnativ... compatible, enhanced, open source d...	1.0.0	5.7.17	free, fully
https://artifacthub.io/packages/helm/cloudnativ... compatible, enhanced, open source d...	1.0.0	5.7.19	free, fully

https://artifacthub.io/packages/helm/kfirfer/pe... compatible, enhanced, open source d...	1.5.10	8.0.32-24.2	free, fully
https://artifacthub.io/packages/helm/someblackm... mysql administration frontend	6.5.2	5.0.2	phpMyAdmin is an
https://artifacthub.io/packages/helm/percona/ps... Deploying the Percona Operator...	0.8.0	0.8.0	A Helm chart for
https://artifacthub.io/packages/helm/wordpress-... create the wordpress with suitabl...	0.1.0	1.1	this is chart
https://artifacthub.io/packages/helm/kubesphere... High-Availability MySQL cluster, o...	1.0.0	5.7.33	XenonDB is a
https://artifacthub.io/packages/helm/evolves/ba... open-source backup tools written i...	0.6.1	0.6.1	Backup is an
https://artifacthub.io/packages/helm/schmitzis/... made by Bookstack and MySQL	0.1.1	21.12	Wiki application
https://artifacthub.io/packages/helm/xxl-job-ad... capable of parsing MySQL binlog...	0.1.2	1.1.6	Canal Server is
https://artifacthub.io/packages/helm/t3n/clouds... Proxy	2.0.3	1.16	Google Cloud SQL
https://artifacthub.io/packages/helm/makaira/co... MariaDB database from one serve...	1.3.0	1.0.0	Copy a MySQL or
https://artifacthub.io/packages/helm/helmcourse... end, MySQL backend packaged in Ra...	0.3.3	adminer: 4.7,mysql: 5.7	Adminer front-
https://artifacthub.io/packages/helm/djjudas21/... help backup SQL databases using...	0.0.16	0.0.9	A Helm chart to
https://artifacthub.io/packages/helm/db-operato... creates databases and make them...	1.30.0	2.10.0	The DB Operator
https://artifacthub.io/packages/helm/ju4t/django 的完美主义者网络框架!	0.1.0	4.1.4	Django是一个有期限
https://artifacthub.io/packages/helm/edgelessy... MySQL-compatible database for c...	0.3.2	v0.3.2	EdgelessDB is a
https://artifacthub.io/packages/helm/monosql-op... deploying Monograph Operator t...	2.0.0	2.0.0	A Helm chart for
https://artifacthub.io/packages/helm/sqream-cha... accessing managed redis & mysql	0.6.0	1.16.0	A Helm chart for

https://artifacthub.io/packages/helm/featurehub... Enterprise Grade, Cloud Native ...	4.1.4	1.8.0	FeatueHub is an
https://artifacthub.io/packages/helm/sqream-cha... Flows service, part of java back...	0.6.2	1.16.0	Helm chart for
https://artifacthub.io/packages/helm/kubeblocks... is a fast & low-cost analytica...	0.9.0	0.2.0	ApeCloud FoxLake
https://artifacthub.io/packages/helm/startechni... modular, high performance free ...	1.0.1	3.2.3	FreeRADIUS is a
https://artifacthub.io/packages/helm/getindata/... Proxy	0.20.5	1.16	Google Cloud SQL
https://artifacthub.io/packages/helm/rimusz/gcl... Proxy	0.25.6	2.6.0	Google Cloud SQL
https://artifacthub.io/packages/helm/datarepo/g... Proxy	0.19.13	1.17	Google Cloud SQL
https://artifacthub.io/packages/helm/ibm-charts... a multi-master solution for M...	1.1.0		Galera Cluster is
https://artifacthub.io/packages/helm/ibm-charts... developed as open source software an...	1.1.2		MariaDB is
https://artifacthub.io/packages/helm/cloudposse... Linux, Apache, MySQL, PHP (aka...	0.1.4		A Helm chart for
https://artifacthub.io/packages/helm/login-test... login_test_backend with MySQL ...	0.1.0	1.0.0	A Helm chart for
https://artifacthub.io/packages/helm/bitnami/ma... open source, community-developed ...	20.2.1	11.4.4	MariaDB is an
https://artifacthub.io/packages/helm/riftbit/ma... scalable, and easy to use open-...	9.6.0	10.5.12	Fast, reliable,
https://artifacthub.io/packages/helm/cloudnativ... scalable, and easy to use open-...	6.1.0	10.3.15	Fast, reliable,
https://artifacthub.io/packages/helm/camptocamp... scalable, and easy to use open-...	1.0.0		Fast, reliable,
https://artifacthub.io/packages/helm/bitnami-ak... open source, community-developed ...	11.4.0	10.6.11	MariaDB is an
https://artifacthub.io/packages/helm/kvalitetsi... open source, community-developed ...	11.2.2	10.6.9	MariaDB is an

https://artifacthub.io/packages/helm/petersando...	15.2.5	11.4.3	Fast, reliable, scalable, and easy to use open-...
https://artifacthub.io/packages/helm/kubeblocks...	1.0.0-alpha.0	10.6.15	MariaDB is an open-source relational database m...
https://artifacthub.io/packages/helm/kvalitetsi...	7.4.1	10.6.9	MariaDB Galera is a multi-primary database clus...
https://artifacthub.io/packages/helm/smo-helm-c...	6.0.0		Chart for MariaDB Galera cluster
https://artifacthub.io/packages/helm/bitnami/ma...	14.1.0	11.4.4	MariaDB Galera is a multi-primary database clus...
https://artifacthub.io/packages/helm/bitnami-ak...	7.4.7	10.6.10	MariaDB Galera is a multi-primary database clus...
https://artifacthub.io/packages/helm/appuio/mar...	1.2.6	10.5.12	MariaDB Galera is a multi-master database clust...
https://artifacthub.io/packages/helm/riftbit/ma...	5.13.5	10.5.12	MariaDB Galera is a multi-master database clust...
https://artifacthub.io/packages/helm/mariadb-op...	0.36.0	0.36.0	Run and operate MariaDB in a cloud native way
https://artifacthub.io/packages/helm/mariadb-op...	0.36.0	0.0.0	mariadb-operator CRDs
https://artifacthub.io/packages/helm/moco/moco	0.15.1	0.25.1	MySQL operator on Kubernetes using GTID-based s...
https://artifacthub.io/packages/helm/sqream-cha...	0.7.1	1.16.0	Helm chart for monitor service, part of java ba...
https://artifacthub.io/packages/helm/monosql-op...	1.0.5	1.0.4	A Helm chart for deploying Monograph Operator t...
https://artifacthub.io/packages/helm/viveksahu2...	1.0.0	2	This is my custom chart to deploy wordpress and...
https://artifacthub.io/packages/helm/mysql2s3bk...	0.1.0	alpha	MySQL database JOBS
https://artifacthub.io/packages/helm/kubeblocks...	3.2.6	3.2.6	Orchestrator is a MySQL high availability and r...
https://artifacthub.io/packages/helm/kubeblocks...	1.0.0-alpha.0	3.2.6	orchestrator
https://artifacthub.io/packages/helm/christianh...	1.13.2		A Helm chart to

deploy Passbolt with MySQL and ... https://artifacthub.io/packages/helm/vquie/phpipam	1.0.3	1.5.2	phpipam is an
open-source web IP address manage... https://artifacthub.io/packages/helm/fsdrw08/po...	0.1.3	4.5.3	PowerDNS all in
one (powerdns + powerdns-admin ... https://artifacthub.io/packages/helm/percona/px...	1.16.0	1.16.0	A Helm chart for
deploying the Percona Operator... https://artifacthub.io/packages/helm/kubesphere...	0.1.0	5.7_1.0.1	A Helm chart for
Kubernetes to run StoneDB stan... https://artifacthub.io/packages/helm/stonedb/st...	0.1.0	5.7_1.0.1	A Helm chart for
Kubernetes to run StoneDB https://artifacthub.io/packages/helm/banzaiclou...	0.0.2		A TiDB Helm chart
for Kubernetes https://artifacthub.io/packages/helm/kubeblocks...	1.0.0-alpha.0	v7.1.5	TiDB is an open-
source, cloud-native, distribut... https://artifacthub.io/packages/helm/webapp-db-...	0.1.0	1.0	Java frontend
with a MySQL backend https://artifacthub.io/packages/helm/sikademo/w...	2020.11.18		Example Wordpress
Chart (with MySQL) https://artifacthub.io/packages/helm/wordpressm...	0.1.0	1.1	
https://artifacthub.io/packages/helm/rock8s/xtradb	1.12.0	0.71.0	enterprise MySQL
https://artifacthub.io/packages/helm/xxl-job-ad...	0.2.3	v3.1.8	☐ A most popular
sql audit platform for mysql			



Important : Notez que la chaîne recherchée peut se retrouver dans l'URL aussi bien que dans la description.

Recherchez maintenant le Chart **WordPress**. Par défaut, les colonnes plus large que 50 caractères sont tronquées :

```
trainee@gateway:~$ helm search hub wordpress
```

URL	CHART VERSION	APP VERSION	DESCRIPTION
-----	---------------	-------------	-------------

https://artifacthub.io/packages/helm/kube-wordp... wordpress package	0.1.0	1.1	this is my
https://artifacthub.io/packages/helm/wordpress-... deploying Wordpress+Mariadb st...	1.0.2	1.0.0	A Helm chart for
https://artifacthub.io/packages/helm/shubham-wo... Kubernetes	0.1.0	1.16.0	A Helm chart for
https://artifacthub.io/packages/helm/bitnami-ak... world's most popular blogging ...	15.2.13	6.1.0	WordPress is the
https://artifacthub.io/packages/helm/bitnami/wo... world's most popular blogging ...	24.1.5	6.7.1	WordPress is the
https://artifacthub.io/packages/helm/sikalabs/w...	0.2.0		Simple Wordpress
https://artifacthub.io/packages/helm/riftbit/wo... platform for building blogs and ...	12.1.16	5.8.1	Web publishing
https://artifacthub.io/packages/helm/schichtel/... WordPress	0.7.1	6.6.1	A Helm chart for
https://artifacthub.io/packages/helm/skywordpre... WordPress	0.1.0	1.16.0	A Helm chart for
https://artifacthub.io/packages/helm/rock8s/wor... content management system	0.12.4	v0.12.4	An open source
https://artifacthub.io/packages/helm/sikademo/w... Chart (with MySQL)	2020.11.18		Example Wordpress
https://artifacthub.io/packages/helm/groundhog2... Wordpress on Kubernetes	0.13.0	6.7.1-apache	A Helm chart for
https://artifacthub.io/packages/helm/devops/wor... chart	0.12.0	1.16.0	Wordpress helm
https://artifacthub.io/packages/helm/camptocamp... platform for building blogs and ...	0.6.10	4.8.1	Web publishing
https://artifacthub.io/packages/helm/homeenterp...	0.5.0	5.9.3-php8.1-apache	Blog server
https://artifacthub.io/packages/helm/mcouliba/w... Kubernetes	0.1.0	1.16.0	A Helm chart for
https://artifacthub.io/packages/helm/wordpress-... Wordpress installation with additio...	6.1.1-3.0rc5	master	Lightweight
https://artifacthub.io/packages/helm/wordpress-... for configuring wordpress and...	1.0.0	1.1	This is a package

https://artifacthub.io/packages/helm/securecode... Outdated Wordpress Instance: Never e...	4.11.0	4.0	Insecure &
https://artifacthub.io/packages/helm/wordpressm... Chart that creates the Wordpre...	1.0.0		This is the Helm
https://artifacthub.io/packages/helm/bitpoke/wo... Operator Helm Chart	0.12.4	v0.12.4	Bitpoke WordPress
https://artifacthub.io/packages/helm/bitpoke/wo... deploying a WordPress site on Bi...	0.12.4	v0.12.4	Helm chart for
https://artifacthub.io/packages/helm/kube-wordp... Wordpress installation on MySQL ...	0.1.0	0.0.1-alpha	Helm Chart for
https://artifacthub.io/packages/helm/riotkit-or... Wordpress installation with additio...	2.1.0-alpha4	v2.1-alpha4	Lightweight
https://artifacthub.io/packages/helm/phntom/bin... www.binaryvision.co.il static wordpress	0.0.4	0.0.3	
https://artifacthub.io/packages/helm/gh-shessel... platform for building blogs and ...	2.1.12	6.1.1	Web publishing
https://artifacthub.io/packages/helm/rock8s/wor...	6.7.1		
https://artifacthub.io/packages/helm/sikalabs/w...	0.1.2		
https://artifacthub.io/packages/helm/gh-shessel... platform for building blogs and ...	6.3.1	6.3.1	Web publishing
https://artifacthub.io/packages/helm/wordpressh... Kubernetes	0.1.0	1.16.0	A Helm chart for
https://artifacthub.io/packages/helm/wordpress-... create the wordpress with suitabl...	0.1.0	1.1	this is chart
https://artifacthub.io/packages/helm/wordpress-... wordpress-gatsby	0.0.1		Helm Chart for
https://artifacthub.io/packages/helm/bitpoke/bi... for WordPress provides a versat...	1.8.18	1.8.18	The Bitpoke App
https://artifacthub.io/packages/helm/sonu-wordp... chart to deploy wordpress and...	1.0.0	2	This is my custom
https://artifacthub.io/packages/helm/wordpress/... Kubernetes	0.2.0	1.1.0	Wordpress for
https://artifacthub.io/packages/helm/uvaise-wor... Kubernetes	0.2.0	1.1.0	Wordpress for

https://artifacthub.io/packages/helm/wordpress-...	1.0.0	2	This is my custom chart to deploy wordpress and...
https://artifacthub.io/packages/helm/bitpoke/stack	0.12.4	v0.12.4	Your Open-Source, Cloud-Native WordPress Infrac...
https://artifacthub.io/packages/helm/securecode...	4.11.0	v3.8.27	A Helm chart for the WordPress security scanner...
https://artifacthub.io/packages/helm/wordpresss...	1.1.0	5.8.2	Web publishing platform for building blogs and ...
https://artifacthub.io/packages/helm/viveksahu2...	1.0.0	2	This is my custom chart to deploy wordpress and...
https://artifacthub.io/packages/helm/six/wordress	0.2.0	1.1.0	Wordpress for Kubernetes
https://artifacthub.io/packages/helm/jinchi-cha...	0.2.0	1.1.0	Wordpress for Kubernetes
https://artifacthub.io/packages/helm/projet-dev...	0.1.0	1.16.0	A Helm chart for wordpress deployed on Azure Ku...
https://artifacthub.io/packages/helm/wordpressm...	0.1.0	1.1	

Pour visualiser l'ensemble de chaque ligne, utilisez l'option **-max-col-width=0** :

```

trainee@gateway:~$ helm search hub wordpress --max-col-width=0
URL                                     CHART VERSION  APP VERSION
DESCRIPTION
https://artifacthub.io/packages/helm/kube-wordpress/wordpress 0.1.0          1.1
this is my wordpress package
https://artifacthub.io/packages/helm/wordpress-mariadb/wordpress 1.0.2          1.0.0
A Helm chart for deploying Wordpress+Mariadb stack on kubernetes.
https://artifacthub.io/packages/helm/shubham-wordpress/wordpress 0.1.0          1.16.0
A Helm chart for Kubernetes
https://artifacthub.io/packages/helm/bitnami-aks/wordpress 15.2.13        6.1.0
WordPress is the world's most popular blogging and content management platform. Powerful yet simple, everyone
from students to global corporations use it to build beautiful, functional websites.
https://artifacthub.io/packages/helm/bitnami/wordpress 24.1.5         6.7.1
WordPress is the world's most popular blogging and content management platform. Powerful yet simple, everyone

```

```

from students to global corporations use it to build beautiful, functional websites.
https://artifacthub.io/packages/helm/sikalabs/wordpress 0.2.0
Simple Wordpress
https://artifacthub.io/packages/helm/riftbit/wordpress 12.1.16 5.8.1
Web publishing platform for building blogs and websites.
https://artifacthub.io/packages/helm/schichtel/wordpress 0.7.1 6.6.1
A Helm chart for WordPress
https://artifacthub.io/packages/helm/skywordpress/wordpress 0.1.0 1.16.0
A Helm chart for WordPress
https://artifacthub.io/packages/helm/rock8s/wordpress 0.12.4 v0.12.4
An open source content management system
https://artifacthub.io/packages/helm/sikademo/wordpress 2020.11.18
Example Wordpress Chart (with MySQL)
https://artifacthub.io/packages/helm/groundhog2k/wordpress 0.13.0 6.7.1-apache
A Helm chart for Wordpress on Kubernetes
https://artifacthub.io/packages/helm/devops/wordpress 0.12.0 1.16.0
Wordpress helm chart
https://artifacthub.io/packages/helm/camptocamp3/wordpress 0.6.10 4.8.1
Web publishing platform for building blogs and websites.
https://artifacthub.io/packages/helm/homeenterpriseinc/wordpress 0.5.0 5.9.3-php8.1-
apache Blog server
https://artifacthub.io/packages/helm/mcouliba/wordpress 0.1.0 1.16.0
A Helm chart for Kubernetes
https://artifacthub.io/packages/helm/wordpress-hardened/wordpress-hardened 6.1.1-3.0rc5 master
Lightweight Wordpress installation with additional security fixes
https://artifacthub.io/packages/helm/wordpress-mysql-1/wordpress-mysql 1.0.0 1.1
This is a package for configuring wordpress and mysql on kubertes automatically
https://artifacthub.io/packages/helm/securecodebox/old-wordpress 4.11.0 4.0
Insecure & Outdated Wordpress Instance: Never expose it to the internet!
https://artifacthub.io/packages/helm/wordpressmysql/wordpress-mysql 1.0.0
This is the Helm Chart that creates the Wordpress and MySQL service with dynamic parameters
https://artifacthub.io/packages/helm/bitpoke/wordpress-operator 0.12.4 v0.12.4
Bitpoke WordPress Operator Helm Chart
https://artifacthub.io/packages/helm/bitpoke/wordpress-site 0.12.4 v0.12.4

```

Helm chart for deploying a WordPress site on Bitpoke Stack		
https://artifacthub.io/packages/helm/kube-wordpress-mysql/kube-wordpress-mysql	0.1.0	0.0.1-alpha
Helm Chart for Wordpress installation on MySQL database		
https://artifacthub.io/packages/helm/riotkit-org/wordpress-hardened	2.1.0-alpha4	v2.1-alpha4
Lightweight Wordpress installation with additional security fixes		
https://artifacthub.io/packages/helm/phntom/binaryvision-static-wordpress	0.0.4	0.0.3
www.binaryvision.co.il static wordpress		
https://artifacthub.io/packages/helm/gh-shesselink81-public/wordpress-apache	2.1.12	6.1.1
Web publishing platform for building blogs and websites.		
https://artifacthub.io/packages/helm/rock8s/wordpress-crds	6.7.1	
https://artifacthub.io/packages/helm/sikalabs/wordpress-mysql	0.1.2	
https://artifacthub.io/packages/helm/gh-shesselink81-public/wordpress-nginx	6.3.1	6.3.1
Web publishing platform for building blogs and websites.		
https://artifacthub.io/packages/helm/wordpresshelmchart/wordpress-helm-chart	0.1.0	1.16.0
A Helm chart for Kubernetes		
https://artifacthub.io/packages/helm/wordpress-mysql/wordpressmysqlenv	0.1.0	1.1
this is chart create the wordpress with suitable wordpress		
https://artifacthub.io/packages/helm/wordpress-gatsby/wp-gats-helm	0.0.1	
Helm Chart for wordpress-gatsby		
https://artifacthub.io/packages/helm/bitpoke/bitpoke	1.8.18	1.8.18
The Bitpoke App for WordPress provides a versatile dashboard to create, deploy, scale, manage and monitor		
WordPress sites in a Kubernetes cluster.		
https://artifacthub.io/packages/helm/sonu-wordpress-mysql/mychart	1.0.0	2
This is my custom chart to deploy wordpress and mysql		
https://artifacthub.io/packages/helm/wordpress/wordress	0.2.0	1.1.0
Wordpress for Kubernetes		
https://artifacthub.io/packages/helm/uvaise-wordpress/wordress	0.2.0	1.1.0
Wordpress for Kubernetes		
https://artifacthub.io/packages/helm/wordpress-mysql-helm-chart/mychart	1.0.0	2
This is my custom chart to deploy wordpress and mysql		
https://artifacthub.io/packages/helm/bitpoke/stack	0.12.4	v0.12.4
Your Open-Source, Cloud-Native WordPress Infrastructure		
https://artifacthub.io/packages/helm/securecodebox/wpscan	4.11.0	v3.8.27
A Helm chart for the WordPress security scanner that integrates with the secureCodeBox.		

https://artifacthub.io/packages/helm/wordpressss-app/app Web publishing platform for building blogs and websites.	1.1.0	5.8.2
https://artifacthub.io/packages/helm/viveksahu26/mychart This is my custom chart to deploy wordpress and mysql	1.0.0	2
https://artifacthub.io/packages/helm/six/wordpress Wordpress for Kubernetes	0.2.0	1.1.0
https://artifacthub.io/packages/helm/jinchi-chart/wordpress Wordpress for Kubernetes	0.2.0	1.1.0
https://artifacthub.io/packages/helm/projet-devops/wp-chart A Helm chart for wordpress deployed on Azure Kuberne service	0.1.0	1.16.0
https://artifacthub.io/packages/helm/wordpressmysqlapp/wpmysql	0.1.0	1.1

Il est aussi possible d'obtenir la sortie au format YAML :

```

trainee@gateway:~$ helm search hub wordpress --output yaml | more
- app_version: "1.1"
  description: this is my wordpress package
  repository:
    name: kube-wordpress
    url: https://harsh-del.github.io/wordpress-charts/charts/
  url: https://artifacthub.io/packages/helm/kube-wordpress/wordpress
  version: 0.1.0
- app_version: 1.0.0
  description: A Helm chart for deploying Wordpress+Mariadb stack on kubernetes.
  repository:
    name: wordpress-mariadb
    url: https://noor-muradi.github.io/Wordpress/
  url: https://artifacthub.io/packages/helm/wordpress-mariadb/wordpress
  version: 1.0.2
- app_version: 1.16.0
  description: A Helm chart for Kubernetes
  repository:
    name: shubham-wordpress
    url: https://shubhambhardwaj007.github.io/Kubernetes-helm-chart/

```

```
url: https://artifacthub.io/packages/helm/shubham-wordpress/wordpress
version: 0.1.0
- app_version: 6.1.0
description: WordPress is the world's most popular blogging and content management
  platform. Powerful yet simple, everyone from students to global corporations use
  it to build beautiful, functional websites.
repository:
  name: bitnami-aks
  url: https://marketplace.azurecr.io/helm/v1/repo
url: https://artifacthub.io/packages/helm/bitnami-aks/wordpress
version: 15.2.13
- app_version: 6.7.1
description: WordPress is the world's most popular blogging and content management
  platform. Powerful yet simple, everyone from students to global corporations use
  it to build beautiful, functional websites.
repository:
  name: bitnami
  url: https://charts.bitnami.com/bitnami
url: https://artifacthub.io/packages/helm/bitnami/wordpress
version: 24.1.5
- app_version: ""
description: Simple Wordpress
repository:
  name: sikalabs
  url: https://helm.sikalabs.io
url: https://artifacthub.io/packages/helm/sikalabs/wordpress
version: 0.2.0
- app_version: 5.8.1
description: Web publishing platform for building blogs and websites.
repository:
  name: riftbit
  url: https://charts.riftbit.com/
url: https://artifacthub.io/packages/helm/riftbit/wordpress
version: 12.1.16
```

```
- app_version: 6.6.1
  description: A Helm chart for WordPress
  repository:
    name: schichtel
    url: https://schich.tel/helm-charts
--More--
```

1.3 - Rechercher dans le Helm Hub

Rendez-vous ensuite à l'adresse du Hub en utilisant un navigateur Web et recherchez **mysql** :

Artifact Hub

SIGN UP SIGN IN

Find, install and publish Kubernetes packages

Q Search packages

Tip: Use **multiple words** to refine your search. Example: `kafka operator`

You can also [browse all packages](#) - or - try one of the sample queries:

[Falco rules for CVE](#) [Helm Charts provided by Bitnami](#) [Helm Charts in the storage category](#)

[OPA policies with MIT license](#) [Packages with Apache-2.0 license](#)

1964 PACKAGES | 29456 RELEASES

Artifact Hub is an **Open Source** project

[GitHub](#) [Slack](#) [Twitter](#)

Explore and discover packages

pihole Updated 10 days ago oes Updated 2 days ago

Dans le résultat retourné, cliquez sur **mysql** du dépôt **Bitnami** :

The screenshot shows the Artifact Hub search results for 'mysql'. The search bar at the top contains 'mysql' and shows 1 - 20 of 44 results. The results are displayed in a grid format. The first result, 'mysql' by Bitnami, is highlighted with a red box. It is a Helm chart for creating a highly available MySQL cluster, with version 7.0.0 and app version 8.0.22. Other results include 'mysql' by Choerodon, 'mysql-backup' by softonic, 'mysql-cluster' by kfirfer, 'stash-mysql' by AppCode Inc., and 'kanister-mysql' by kanister. The left sidebar contains filters for Official repositories, Verified publishers, Category (Database, Integration and Delivery, Logging and Tracing, Machine learning, Monitoring, Networking, Security, Storage, Streaming and Messaging, Web applications), Kind (Helm charts, OLM operators), Publisher, and Repository.

Artifact HUB SIGN UP SIGN IN

1 - 20 of 44 results for "mysql" Show: 20

FILTERS

Official repositories Verified publishers

CATEGORY

Database
 Integration and Delivery
 Logging and Tracing
 Machine learning
 Monitoring
 Networking
 Security
 Storage
 Streaming and Messaging
 Web applications

KIND

Helm charts (40)
 OLM operators (4)

PUBLISHER
No publisher selected

REPOSITORY
No repository selected

mysql (highlighted)
ORG: Bitnami REPO: Bitnami
VERSION: 7.0.0 APP VERSION: 8.0.22
Updated a day ago
★ 4
Chart to create a Highly available MySQL cluster

mysql
ORG: Helm REPO: choerodon
VERSION: 0.1.4 APP VERSION: 0.1.4
Updated 6 months ago
★ 0
mysql for Choerodon

mysql-backup
ORG: Helm REPO: softonic
VERSION: 2.1.4 APP VERSION: 0.2.0 LICENSE: Apache-2.0
Updated 4 days ago
★ 1
Take mysql backups from any mysql instance to AWS S3. Supports GCP CloudSQL

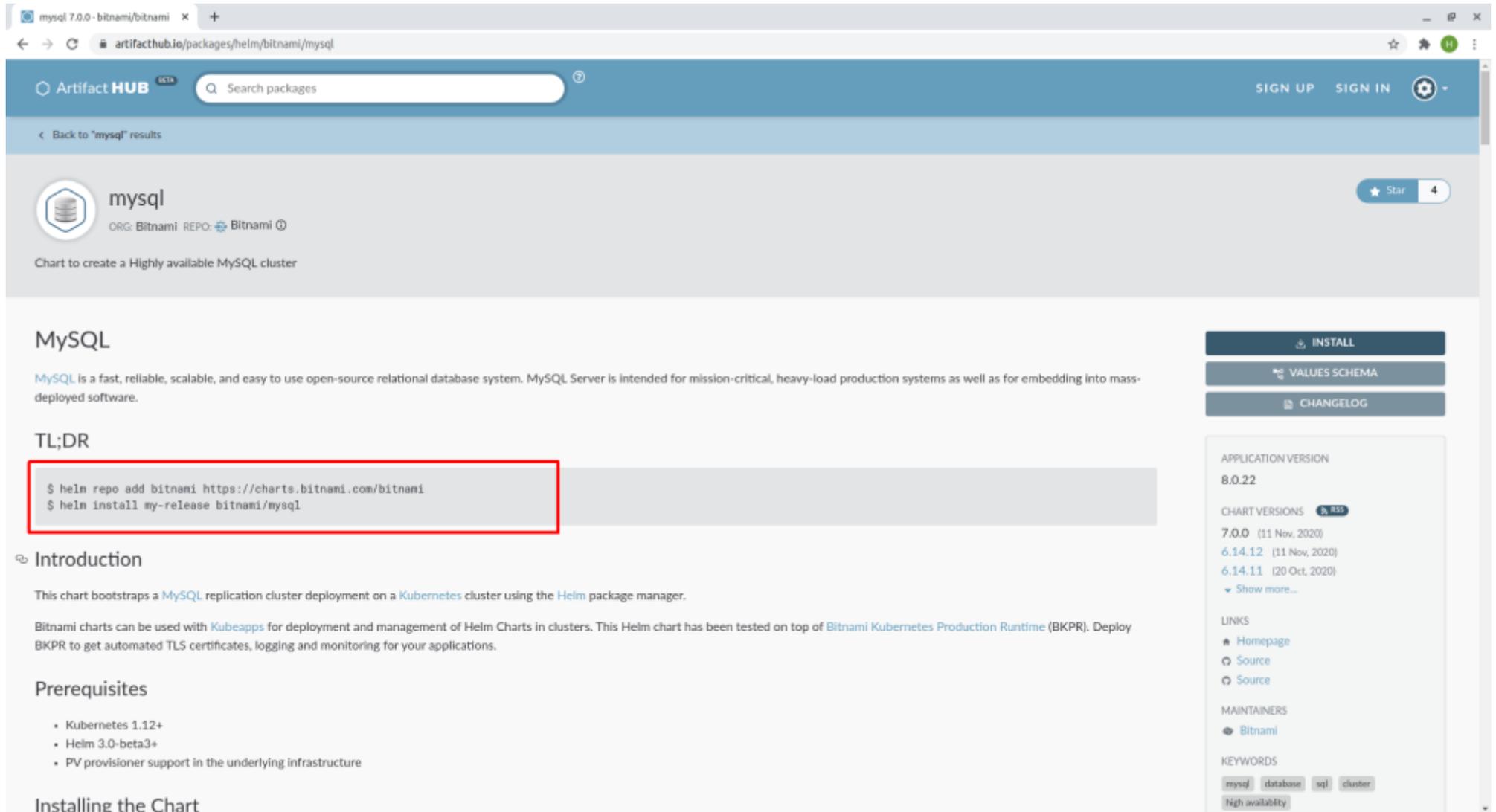
mysql-cluster
USER: kfirfer REPO: kfirfer
VERSION: 0.0.3 APP VERSION: 8.0.20
Updated 2 months ago
★ 1
A Helm chart for MySQL Cluster

stash-mysql
ORG: AppCode Inc. REPO: appcode
VERSION: 8.0.14 APP VERSION: 8.0.14
Updated 5 months ago
★ 0
stash-mysql - MySQL database backup and restore plugin for Stash by AppCode
Verified Publisher

kanister-mysql
ORG: Helm REPO: kanister
VERSION: 0.32.0 APP VERSION: 5.7.14
Updated 4 months ago
★ 0
MySQL w/ Kanister support based on stable/mysql

mysql-cluster (bottom row)
Updated 2 months ago

Vous constaterez des informations concernant ce Chart ainsi que des instructions pour l'installer :



mysql 7.0.0 - bitnami/bitnami

artifacthub.io/packages/helm/bitnami/mysql

Artifact HUB

Search packages

SIGN UP SIGN IN

Back to "mysql" results

mysql

ORG: Bitnami REPO: Bitnami

Chart to create a Highly available MySQL cluster

MySQL

MySQL is a fast, reliable, scalable, and easy to use open-source relational database system. MySQL Server is intended for mission-critical, heavy-load production systems as well as for embedding into mass-deployed software.

TL;DR

```
$ helm repo add bitnami https://charts.bitnami.com/bitnami
$ helm install my-release bitnami/mysql
```

INSTALL

VALUES SCHEMA

CHANGELOG

APPLICATION VERSION

8.0.22

CHART VERSIONS

7.0.0 (11 Nov, 2020)

6.14.12 (11 Nov, 2020)

6.14.11 (20 Oct, 2020)

Show more...

LINKS

Homepage

Source

Source

MAINTAINERS

Bitnami

KEYWORDS

mysql database sql cluster

high availability

Introduction

This chart bootstraps a MySQL replication cluster deployment on a Kubernetes cluster using the Helm package manager.

Bitnami charts can be used with Kubeapps for deployment and management of Helm Charts in clusters. This Helm chart has been tested on top of Bitnami Kubernetes Production Runtime (BKPR). Deploy BKPR to get automated TLS certificates, logging and monitoring for your applications.

Prerequisites

- Kubernetes 1.12+
- Helm 3.0-beta3+
- PV provisioner support in the underlying infrastructure

Installing the Chart

1.4 - Ajouter et supprimer un Dépôt

En utilisant les informations prises du Hub, installez maintenant le dépôt **bitnami** :

```
trainee@gateway:~$ helm repo add bitnami https://charts.bitnami.com/bitnami
```

```
"bitnami" has been added to your repositories
```

Pour lister les dépôts installés, utilisez la commande **helm repo list** :

```
trainee@gateway:~$ helm repo list
NAME      URL
bitnami   https://charts.bitnami.com/bitnami
```

Pour mettre à jour les dépôts, utilisez la commande **helm repo update** :

```
trainee@gateway:~$ helm repo update
Hang tight while we grab the latest from your chart repositories...
...Successfully got an update from the "bitnami" chart repository
Update Complete. *Happy Helming!*
```

Pour supprimer un dépôt, utilisez la commande **helm repo remove <repo>**

```
trainee@gateway:~$ helm repo remove bitnami
"bitnami" has been removed from your repositories
```

```
trainee@gateway:~$ helm repo list
Error: no repositories to show
```

1.5 - La Commande **helm search repo**

Ajoutez de nouveau le dépôt Bitnami :

```
trainee@gateway:~$ helm repo add bitnami https://charts.bitnami.com/bitnami
"bitnami" has been added to your repositories
```

Pour voir le contenu de ce dépôt au format YAML, utilisez la commande **helm search repo** :

```
trainee@gateway:~$ helm search repo bitnami --output yaml | more
- app_version: 2.10.4
  description: Apache Airflow is a tool to express and execute workflows as directed
    acyclic graphs (DAGs). It includes utilities to schedule tasks, monitor task progress
    and handle task dependencies.
  name: bitnami/airflow
  version: 22.4.3
- app_version: 2.4.62
  description: Apache HTTP Server is an open-source HTTP server. The goal of this
    project is to provide a secure, efficient and extensible server that provides
    HTTP services in sync with the current HTTP standards.
  name: bitnami/apache
  version: 11.3.0
- app_version: 3.11.0
  description: Apache APISIX is high-performance, real-time API Gateway. Features
    load balancing, dynamic upstream, canary release, circuit breaking, authentication,
    observability, amongst others.
  name: bitnami/apisix
  version: 3.7.0
- app_version: 1.53.0
  description: Appsmith is an open source platform for building and maintaining internal
    tools, such as custom dashboards, admin panels or CRUD apps.
  name: bitnami/appsmith
  version: 5.1.2
- app_version: 2.13.2
  description: Argo CD is a continuous delivery tool for Kubernetes based on GitOps.
  name: bitnami/argo-cd
  version: 7.1.1
- app_version: 3.6.2
  description: Argo Workflows is meant to orchestrate Kubernetes jobs in parallel.
    It uses DAG and step-based workflows
  name: bitnami/argo-workflows
  version: 11.1.0
- app_version: 8.0.11
```

```
description: ASP.NET Core is an open-source framework for web application development
  created by Microsoft. It runs on both the full .NET Framework, on Windows, and
  the cross-platform .NET Core.
name: bitnami/aspnet-core
version: 6.3.0
- app_version: 5.0.2
description: Apache Cassandra is an open source distributed database management
  system designed to handle large amounts of data across many servers, providing
  high availability with no single point of failure.
name: bitnami/cassandra
version: 12.1.0
- app_version: 1.16.2
description: cert-manager is a Kubernetes add-on to automate the management and
  issuance of TLS certificates from various issuing sources.
name: bitnami/cert-manager
version: 1.4.0
- app_version: 0.146.0
description: Chainloop is an open-source Software Supply Chain control plane, a
  single source of truth for metadata and artifacts, plus a declarative attestation
  process.
name: bitnami/chainloop
version: 2.1.9
- app_version: 1.16.5
description: Cilium is an eBPF-based networking, observability, and security for
  Linux container management platforms like Docker and Kubernetes.
--More--
```

Recherchez maintenant Wordpress dans le dépôt Bitnami :

```
trainee@gateway:~$ helm search repo wordpress
NAME                CHART VERSION  APP VERSION  DESCRIPTION
bitnami/wordpress  24.1.5         6.7.1        WordPress is the world's most popular blogging ...
bitnami/wordpress-intel 2.1.31        6.1.1        DEPRECATED WordPress for Intel is the most popu...
```

La commande précédemment exécutée ne retourne que la dernière version de Wordpress. Pour voir la liste des versions disponibles, utilisez l'option **-versions** :

```
trainee@gateway:~$ helm search repo wordpress --versions | more
NAME                CHART VERSION  APP VERSION  DESCRIPTION
bitnami/wordpress  24.1.5         6.7.1       WordPress is the world's most popular blogging ...
bitnami/wordpress  24.1.4         6.7.1       WordPress is the world's most popular blogging ...
bitnami/wordpress  24.1.3         6.7.1       WordPress is the world's most popular blogging ...
bitnami/wordpress  24.1.2         6.7.1       WordPress is the world's most popular blogging ...
bitnami/wordpress  24.1.1         6.7.1       WordPress is the world's most popular blogging ...
bitnami/wordpress  24.1.0         6.7.1       WordPress is the world's most popular blogging ...
bitnami/wordpress  24.0.11        6.7.1       WordPress is the world's most popular blogging ...
bitnami/wordpress  24.0.10        6.7.1       WordPress is the world's most popular blogging ...
bitnami/wordpress  24.0.9         6.7.1       WordPress is the world's most popular blogging ...
bitnami/wordpress  24.0.8         6.7.1       WordPress is the world's most popular blogging ...
bitnami/wordpress  24.0.7         6.7.1       WordPress is the world's most popular blogging ...
bitnami/wordpress  24.0.6         6.7.0       WordPress is the world's most popular blogging ...
bitnami/wordpress  24.0.5         6.7.0       WordPress is the world's most popular blogging ...
bitnami/wordpress  24.0.4         6.7.0       WordPress is the world's most popular blogging ...
bitnami/wordpress  24.0.3         6.7.0       WordPress is the world's most popular blogging ...
bitnami/wordpress  24.0.2         6.7.0       WordPress is the world's most popular blogging ...
bitnami/wordpress  24.0.1         6.6.2       WordPress is the world's most popular blogging ...
bitnami/wordpress  24.0.0         6.6.2       WordPress is the world's most popular blogging ...
bitnami/wordpress  23.1.29        6.6.2       WordPress is the world's most popular blogging ...
bitnami/wordpress  23.1.28        6.6.2       WordPress is the world's most popular blogging ...
bitnami/wordpress  23.1.27        6.6.2       WordPress is the world's most popular blogging ...
bitnami/wordpress  23.1.26        6.6.2       WordPress is the world's most popular blogging ...
bitnami/wordpress  23.1.25        6.6.2       WordPress is the world's most popular blogging ...
bitnami/wordpress  23.1.24        6.6.2       WordPress is the world's most popular blogging ...
bitnami/wordpress  23.1.23        6.6.2       WordPress is the world's most popular blogging ...
bitnami/wordpress  23.1.22        6.6.2       WordPress is the world's most popular blogging ...
bitnami/wordpress  23.1.21        6.6.2       WordPress is the world's most popular blogging ...
bitnami/wordpress  23.1.20        6.6.2       WordPress is the world's most popular blogging ...
bitnami/wordpress  23.1.19        6.6.2       WordPress is the world's most popular blogging ...
```

```
bitnami/wordpress      23.1.18      6.6.2      WordPress is the world's most popular blogging ...
bitnami/wordpress      23.1.17      6.6.2      WordPress is the world's most popular blogging ...
bitnami/wordpress      23.1.16      6.6.2      WordPress is the world's most popular blogging ...
bitnami/wordpress      23.1.15      6.6.2      WordPress is the world's most popular blogging ...
bitnami/wordpress      23.1.14      6.6.2      WordPress is the world's most popular blogging ...
bitnami/wordpress      23.1.13      6.6.2      WordPress is the world's most popular blogging ...
bitnami/wordpress      23.1.12      6.6.1      WordPress is the world's most popular blogging ...
bitnami/wordpress      23.1.11      6.6.1      WordPress is the world's most popular blogging ...
bitnami/wordpress      23.1.10      6.6.1      WordPress is the world's most popular blogging ...
bitnami/wordpress      23.1.9       6.6.1      WordPress is the world's most popular blogging ...
bitnami/wordpress      23.1.8       6.6.1      WordPress is the world's most popular blogging ...
bitnami/wordpress      23.1.7       6.6.1      WordPress is the world's most popular blogging ...
bitnami/wordpress      23.1.6       6.6.1      WordPress is the world's most popular blogging ...
bitnami/wordpress      23.1.5       6.6.1      WordPress is the world's most popular blogging ...
bitnami/wordpress      23.1.4       6.6.1      WordPress is the world's most popular blogging ...
bitnami/wordpress      23.1.3       6.6.1      WordPress is the world's most popular blogging ...
bitnami/wordpress      23.1.2       6.6.1      WordPress is the world's most popular blogging ...
bitnami/wordpress      23.1.1       6.6.1      WordPress is the world's most popular blogging ...
bitnami/wordpress      23.1.0       6.6.1      WordPress is the world's most popular blogging ...
bitnami/wordpress      23.0.15      6.6.1      WordPress is the world's most popular blogging ...
bitnami/wordpress      23.0.14      6.6.1      WordPress is the world's most popular blogging ...
bitnami/wordpress      23.0.13      6.6.1      WordPress is the world's most popular blogging ...
bitnami/wordpress      23.0.12      6.6.1      WordPress is the world's most popular blogging ...
bitnami/wordpress      23.0.11      6.6.1      WordPress is the world's most popular blogging ...
bitnami/wordpress      23.0.10      6.6.1      WordPress is the world's most popular blogging ...
bitnami/wordpress      23.0.9       6.6.0      WordPress is the world's most popular blogging ...
bitnami/wordpress      23.0.8       6.6.0      WordPress is the world's most popular blogging ...
bitnami/wordpress      23.0.7       6.6.0      WordPress is the world's most popular blogging ...
--More--
```

1.6 - La Commande helm show

Pour voir les méta-données d'une version du Chart, il convient d'utiliser la commande **helm show chart** :

```
trainee@gateway:~$ helm show chart bitnami/wordpress --version=24.1.5
annotations:
  category: CMS
  images: |
    - name: apache-exporter
      image: docker.io/bitnami/apache-exporter:1.0.9-debian-12-r6
    - name: os-shell
      image: docker.io/bitnami/os-shell:12-debian-12-r34
    - name: wordpress
      image: docker.io/bitnami/wordpress:6.7.1-debian-12-r9
  licenses: Apache-2.0
apiVersion: v2
appVersion: 6.7.1
dependencies:
- condition: memcached.enabled
  name: memcached
  repository: oci://registry-1.docker.io/bitnamicharts
  version: 7.x.x
- condition: mariadb.enabled
  name: mariadb
  repository: oci://registry-1.docker.io/bitnamicharts
  version: 20.x.x
- name: common
  repository: oci://registry-1.docker.io/bitnamicharts
  tags:
  - bitnami-common
  version: 2.x.x
description: WordPress is the world's most popular blogging and content management
  platform. Powerful yet simple, everyone from students to global corporations use
  it to build beautiful, functional websites.
home: https://bitnami.com
icon: https://bitnami.com/assets/stacks/wordpress/img/wordpress-stack-220x234.png
keywords:
- application
```

```
- blog
- cms
- http
- php
- web
- wordpress
maintainers:
- name: Broadcom, Inc. All Rights Reserved.
  url: https://github.com/bitnami/charts
name: wordpress
sources:
- https://github.com/bitnami/charts/tree/main/bitnami/wordpress
version: 24.1.5
```

Pour consultez le fichier README du Chart, il convient d'utiliser la commande **helm show readme** :

```
trainee@gateway:~$ helm show readme bitnami/wordpress --version 24.1.5 | more
<!-- app-name: WordPress -->
```

```
# Bitnami package for WordPress
```

```
WordPress is the world's most popular blogging and content management platform. Powerful yet simple, everyone from students to global corporations use it to build beautiful, functional websites.
```

```
[Overview of WordPress](http://www.wordpress.org)
```

```
## TL;DR
```

```
```console
```

```
helm install my-release oci://registry-1.docker.io/bitnamicharts/wordpress
```

```
```
```

```
Looking to use WordPress in production? Try [VMware Tanzu Application Catalog](https://bitnami.com/enterprise), the commercial edition of the Bitnami catalog.
```

Introduction

This chart bootstraps a [WordPress](https://github.com/bitnami/containers/tree/main/bitnami/wordpress) deployment on a [Kubernetes](https://kubernetes.io) cluster using the [Helm](https://helm.sh) package manager.

It also packages the [Bitnami MariaDB chart](https://github.com/bitnami/charts/tree/main/bitnami/mariadb) which is required for bootstrapping a MariaDB deployment for the database requirements of the WordPress application, and the [Bitnami Memcached chart](https://github.com/bitnami/charts/tree/main/bitnami/memcached) that can be used to cache database queries.

Bitnami charts can be used with [Kubeapps](https://kubernetes.dev/) for deployment and management of Helm Charts in clusters.

Prerequisites

- Kubernetes 1.23+
- Helm 3.8.0+
- PV provisioner support in the underlying infrastructure
- ReadWriteMany volumes for deployment scaling

Installing the Chart

To install the chart with the release name `my-release`:

```
```console
helm install my-release oci://REGISTRY_NAME/REPOSITORY_NAME/wordpress
```
```

> Note: You need to substitute the placeholders `REGISTRY_NAME` and `REPOSITORY_NAME` with a reference to your Helm chart registry and repository. For example, in the case of Bitnami, you need to use `REGISTRY_NAME=registry-1.docker.io` and `REPOSITORY_NAME=bitnamicharts`.

The command deploys WordPress on the Kubernetes cluster in the default configuration. The

```
[Parameters](#parameters) section lists the parameters that can be configured during installation.
```

```
> **Tip**: List all releases using `helm list`
```

```
## Configuration and installation details
```

```
### Resource requests and limits
```

Bitnami charts allow setting resource requests and limits for all containers inside the chart deployment. These are inside the `resources` value (check parameter table). Setting requests is essential for production workloads and these should be adapted to your specific use case.

To make this process easier, the chart contains the `resourcesPreset` values, which automatically sets the `resources` section according to different presets. Check these presets in [the bitnami/common chart](https://github.com/bitnami/charts/blob/main/bitnami/common/templates/_resources.tpl#L15). However, in production workloads using `resourcesPreset` is discouraged as it may not fully adapt to your specific needs.

Find more information on container resource management in the [official Kubernetes documentation](<https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>).

```
--More--
```

Pour consultez les valeurs du Chart, il convient d'utiliser la commande **helm show values** :

```
trainee@gateway:~$ helm show values bitnami/wordpress --version 24.1.5 | more
```

```
# Copyright Broadcom, Inc. All Rights Reserved.
```

```
# SPDX-License-Identifier: APACHE-2.0
```

```
## @section Global parameters
```

```
## Global Docker image parameters
```

```
## Please, note that this will override the image parameters, including dependencies, configured to use the global value
```

```
## Current available global Docker image parameters: imageRegistry, imagePullSecrets and storageClass
```

```
##
```

```
## @param global.imageRegistry Global Docker image registry
```

```
## @param global.imagePullSecrets Global Docker registry secret names as an array
## @param global.defaultStorageClass Global default StorageClass for Persistent Volume(s)
##
global:
  imageRegistry: ""
  ## E.g.
  ## imagePullSecrets:
  ##   - myRegistryKeySecretName
  ##
  imagePullSecrets: []
  defaultStorageClass: ""
  ## Security parameters
  ##
  security:
    ## @param global.security.allowInsecureImages Allows skipping image verification
    allowInsecureImages: false
  ## Compatibility adaptations for Kubernetes platforms
  ##
  compatibility:
    ## Compatibility adaptations for Openshift
    ##
    openshift:
      ## @param global.compatibility.openshift.adaptSecurityContext Adapt the securityContext sections of the
      ## deployment to make them compatible with Openshift restricted-v2 SCC: remove runAsUser, runAsGroup and
      ## fsGroup and let the platform use their allowed default IDs. Possible values: auto (apply if the detected
      ## running cluster is Openshift), force (perform the adaptation always), disabled (do not perform adaptati
      ## on)
      ##
      adaptSecurityContext: auto
  ## @section Common parameters
  ##

## @param kubeVersion Override Kubernetes version
##
```

```
kubeVersion: ""
## @param nameOverride String to partially override common.names.fullname template (will maintain the release
name)
##
nameOverride: ""
## @param fullnameOverride String to fully override common.names.fullname template
##
fullnameOverride: ""
## @param commonLabels Labels to add to all deployed resources
##
commonLabels: {}
## @param commonAnnotations Annotations to add to all deployed resources
##
commonAnnotations: {}
## @param clusterDomain Kubernetes Cluster Domain
##
clusterDomain: cluster.local
--More--
```



Important : Les valeurs par défaut peuvent être surchargées par des valeurs dans un fichier au format YAML.

Pour consultez les méta-données, le fichier README ainsi que les valeurs du Chart, il convient d'utiliser la commande **helm show all** :

```
trainee@gateway:~$ helm show all bitnami/wordpress --version 24.1.5 | more
annotations:
  category: CMS
  images: |
    - name: apache-exporter
      image: docker.io/bitnami/apache-exporter:1.0.9-debian-12-r6
    - name: os-shell
```

```
  image: docker.io/bitnami/os-shell:12-debian-12-r34
- name: wordpress
  image: docker.io/bitnami/wordpress:6.7.1-debian-12-r9
  licenses: Apache-2.0
apiVersion: v2
appVersion: 6.7.1
dependencies:
- condition: memcached.enabled
  name: memcached
  repository: oci://registry-1.docker.io/bitnamicharts
  version: 7.x.x
- condition: mariadb.enabled
  name: mariadb
  repository: oci://registry-1.docker.io/bitnamicharts
  version: 20.x.x
- name: common
  repository: oci://registry-1.docker.io/bitnamicharts
  tags:
  - bitnami-common
  version: 2.x.x
description: WordPress is the world's most popular blogging and content management
  platform. Powerful yet simple, everyone from students to global corporations use
  it to build beautiful, functional websites.
home: https://bitnami.com
icon: https://bitnami.com/assets/stacks/wordpress/img/wordpress-stack-220x234.png
keywords:
- application
- blog
- cms
- http
- php
- web
- wordpress
maintainers:
```

```
- name: Broadcom, Inc. All Rights Reserved.  
  url: https://github.com/bitnami/charts  
name: wordpress  
sources:  
- https://github.com/bitnami/charts/tree/main/bitnami/wordpress  
version: 24.1.5  
  
---  
# Copyright Broadcom, Inc. All Rights Reserved.  
# SPDX-License-Identifier: APACHE-2.0  
  
## @section Global parameters  
## Global Docker image parameters  
## Please, note that this will override the image parameters, including dependencies, configured to use the  
## global value  
## Current available global Docker image parameters: imageRegistry, imagePullSecrets and storageClass  
##  
  
## @param global.imageRegistry Global Docker image registry  
--More--
```

1.7 - Installation d'un Chart

Créez un Namespace appelé **dof309** dans lequel vous allez installer WordPress :

```
trainee@gateway:~$ kubectl create namespace dof309  
namespace/dof309 created
```

Créez le fichier **wordpress-values.yaml** contenant des valeurs qui surchargeront celles du Chart WordPress :

```
trainee@gateway:~$ vi wordpress-values.yaml  
trainee@gateway:~$ cat wordpress-values.yaml  
wordpressUsername: helm-user
```

```
wordpressPassword: my-pass
wordpressEmail: helm-user@i2tch.loc
wordpressFirstName: Helm
wordpressLastName: User
wordpressBlogName: Learn Helm!
service:
  type: NodePort
```

Installez maintenant le Chart WordPress :

```
trainee@gateway:~$ helm install wordpress bitnami/wordpress --values=wordpress-values.yaml --namespace dof309 --
version 24.1.5
NAME: wordpress
LAST DEPLOYED: Fri Dec 20 15:14:17 2024
NAMESPACE: dof309
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
CHART NAME: wordpress
CHART VERSION: 24.1.5
APP VERSION: 6.7.1
```

Did you know there are enterprise versions of the Bitnami catalog? For enhanced secure software supply chain features, unlimited pulls from Docker, LTS support, or application customization, see Bitnami Premium or Tanzu Application Catalog. See <https://www.arrow.com/globalecs/na/vendors/bitnami> for more information.

**** Please be patient while the chart is being deployed ****

Your WordPress site can be accessed through the following DNS name from within your cluster:

```
wordpress.dof309.svc.cluster.local (port 80)
```

To access your WordPress site from outside the cluster follow the steps below:

1. Get the WordPress URL by running these commands:

```
export NODE_PORT=$(kubectl get --namespace dof309 -o jsonpath="{.spec.ports[0].nodePort}" services wordpress)
export NODE_IP=$(kubectl get nodes --namespace dof309 -o jsonpath="{.items[0].status.addresses[0].address}")
echo "WordPress URL: http://$NODE_IP:$NODE_PORT/"
echo "WordPress Admin URL: http://$NODE_IP:$NODE_PORT/admin"
```

2. Open a browser and access WordPress using the obtained URL.

3. Login with the following credentials below to see your blog:

```
echo Username: helm-user
echo Password: $(kubectl get secret --namespace dof309 wordpress -o jsonpath="{.data.wordpress-password}" |
base64 -d)
```

WARNING: There are "resources" sections in the chart not set. Using "resourcesPreset" is not recommended for production. For production installations, please set the following values according to your workload needs:

```
- resources
+info https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/
```

Vérifiez le statut du deployment de l'application :

```
trainee@gateway:~$ helm list --namespace dof309
```

| NAME | NAMESPACE | REVISION | UPDATED | STATUS | CHART |
|------------------|-----------|----------|---|----------|-------|
| wordpress | dof309 | 1 | 2024-12-20 15:14:17.143771785 +0100 CET | deployed | |
| wordpress-24.1.5 | 6.7.1 | | | | |

1.8 - La Commande helm get

Helm utilise un système de **Hooks** qui permet au développeurs d'exécuter des tâches à des moments précis. Les Hooks sont :

- pre-install

- post-install
- pre-delete
- post-delete
- pre-upgrade
- post-upgrade
- pre-rollback
- post-rollback
- test

Pour visualiser les Hooks, s'ils existent, il convient d'utiliser la commande **helm get hooks** :

```
trainee@gateway:~$ helm get hooks wordpress --namespace dof309
trainee@gateway:~$
```

Pour consulter les manifests, utilisez la commande **helm get manifest** :

```
trainee@gateway:~$ helm get manifest wordpress --namespace dof309 | more
---
# Source: wordpress/charts/mariadb/templates/networkpolicy.yaml
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
  name: wordpress-mariadb
  namespace: "dof309"
  labels:
    app.kubernetes.io/instance: wordpress
    app.kubernetes.io/managed-by: Helm
    app.kubernetes.io/name: mariadb
    app.kubernetes.io/version: 11.4.4
    helm.sh/chart: mariadb-20.2.1
    app.kubernetes.io/part-of: mariadb
spec:
  podSelector:
    matchLabels:
```

```
  app.kubernetes.io/instance: wordpress
  app.kubernetes.io/managed-by: Helm
  app.kubernetes.io/name: mariadb
  app.kubernetes.io/version: 11.4.4
  helm.sh/chart: mariadb-20.2.1
policyTypes:
- Ingress
- Egress
egress:
- {}
ingress:
- ports:
  - port: 3306
  - port: 3306
---
# Source: wordpress/templates/networkpolicy.yaml
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
  name: wordpress
  namespace: "dof309"
  labels:
    app.kubernetes.io/instance: wordpress
    app.kubernetes.io/managed-by: Helm
    app.kubernetes.io/name: wordpress
    app.kubernetes.io/version: 6.7.1
    helm.sh/chart: wordpress-24.1.5
spec:
  podSelector:
    matchLabels:
      app.kubernetes.io/instance: wordpress
      app.kubernetes.io/name: wordpress
  policyTypes:
  - Ingress
```

```
- Egress
egress:
- {}
ingress:
- ports:
  - port: 8080
  - port: 8443
--More--
```

Il est possible de revoir les **NOTES** imprimées à l'écran lors de l'exécution de la commande helm install en utilisant le comande **help get notes** :

```
trainee@gateway:~$ helm get notes wordpress --namespace dof309
NOTES:
CHART NAME: wordpress
CHART VERSION: 24.1.5
APP VERSION: 6.7.1
```

Did you know there are enterprise versions of the Bitnami catalog? For enhanced secure software supply chain features, unlimited pulls from Docker, LTS support, or application customization, see Bitnami Premium or Tanzu Application Catalog. See <https://www.arrow.com/globalecs/na/vendors/bitnami> for more information.

**** Please be patient while the chart is being deployed ****

Your WordPress site can be accessed through the following DNS name from within your cluster:

```
wordpress.dof309.svc.cluster.local (port 80)
```

To access your WordPress site from outside the cluster follow the steps below:

1. Get the WordPress URL by running these commands:

```
export NODE_PORT=$(kubectl get --namespace dof309 -o jsonpath="{.spec.ports[0].nodePort}" services wordpress)
export NODE_IP=$(kubectl get nodes --namespace dof309 -o jsonpath="{.items[0].status.addresses[0].address}")
echo "WordPress URL: http://$NODE_IP:$NODE_PORT/"
```

```
echo "WordPress Admin URL: http://$NODE_IP:$NODE_PORT/admin"
```

2. Open a browser and access WordPress using the obtained URL.

3. Login with the following credentials below to see your blog:

```
echo Username: helm-user
```

```
echo Password: $(kubectl get secret --namespace dof309 wordpress -o jsonpath="{.data.wordpress-password}" | base64 -d)
```

WARNING: There are "resources" sections in the chart not set. Using "resourcesPreset" is not recommended for production. For production installations, please set the following values according to your workload needs:

- resources

+info <https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Pour visualiser les valeurs spécifiées par l'utilisateur dans le fichier de valeurs, utilisez la commande **helm get values** :

```
trainee@gateway:~$ helm get values wordpress --namespace dof309
```

```
USER-SUPPLIED VALUES:
```

```
service:
```

```
  type: NodePort
```

```
wordpressBlogName: Learn Helm!
```

```
wordpressEmail: helm-user@i2tch.loc
```

```
wordpressFirstName: Helm
```

```
wordpressLastName: User
```

```
wordpressPassword: my-pass
```

```
wordpressUsername: helm-user
```

Pour voir toutes les valeurs, il convient d'utiliser la commande **helm get values** avec l'option **-all** :

```
trainee@gateway:~$ helm get values wordpress --all --namespace dof309 | more
```

```
COMPUTED VALUES:
```

```
affinity: {}
```

```
allowEmptyPassword: true
```

```
allowOverrideNone: false
apacheConfiguration: ""
args: []
automountServiceAccountToken: false
autoscaling:
  enabled: false
  maxReplicas: 11
  minReplicas: 1
  targetCPU: 50
  targetMemory: 50
clusterDomain: cluster.local
command: []
common:
  exampleValue: common-chart
  global:
    compatibility:
      openshift:
        adaptSecurityContext: auto
    defaultStorageClass: ""
    imagePullSecrets: []
    imageRegistry: ""
    security:
      allowInsecureImages: false
commonAnnotations: {}
commonLabels: {}
containerPorts:
  http: 8080
  https: 8443
containerSecurityContext:
  allowPrivilegeEscalation: false
  capabilities:
    drop:
      - ALL
  enabled: true
```

```
privileged: false
readOnlyRootFilesystem: true
runAsGroup: 1001
runAsNonRoot: true
runAsUser: 1001
seLinuxOptions: {}
seccompProfile:
  type: RuntimeDefault
customHTAccessCM: ""
customLivenessProbe: {}
customPostInitScripts: {}
customReadinessProbe: {}
customStartupProbe: {}
diagnosticMode:
  args:
  - infinity
  command:
  - sleep
  enabled: false
existingApacheConfigurationConfigMap: ""
existingSecret: ""
--More--
```

Dernièrement pour visualiser toutes les informations, utilisez la commande **helm get all** :

```
trainee@gateway:~$ helm get all wordpress --namespace dof309 | more
NAME: wordpress
LAST DEPLOYED: Fri Dec 20 15:14:17 2024
NAMESPACE: dof309
STATUS: deployed
REVISION: 1
CHART: wordpress
VERSION: 24.1.5
APP_VERSION: 6.7.1
```

```
TEST SUITE: None
USER-SUPPLIED VALUES:
service:
  type: NodePort
wordpressBlogName: Learn Helm!
wordpressEmail: helm-user@i2tch.loc
wordpressFirstName: Helm
wordpressLastName: User
wordpressPassword: my-pass
wordpressUsername: helm-user

COMPUTED VALUES:
affinity: {}
allowEmptyPassword: true
allowOverrideNone: false
apacheConfiguration: ""
args: []
automountServiceAccountToken: false
autoscaling:
  enabled: false
  maxReplicas: 11
  minReplicas: 1
  targetCPU: 50
  targetMemory: 50
clusterDomain: cluster.local
command: []
common:
  exampleValue: common-chart
global:
  compatibility:
    openshift:
      adaptSecurityContext: auto
  defaultStorageClass: ""
  imagePullSecrets: []
```

```
imageRegistry: ""
security:
  allowInsecureImages: false
commonAnnotations: {}
commonLabels: {}
containerPorts:
  http: 8080
  https: 8443
containerSecurityContext:
  allowPrivilegeEscalation: false
  capabilities:
    drop:
      - ALL
  enabled: true
  privileged: false
  readOnlyRootFilesystem: true
--More--
```

1.9 - Utilisation des NOTES

Vérifiez le nombre de Pods qui ont été démarrés :

```
trainee@gateway:~$ kubectl get pods -n dof309
NAME                                READY   STATUS    RESTARTS   AGE
wordpress-75b57d8d9b-9t4jw         0/1     Running   0           4m22s
wordpress-mariadb-0                0/1     Running   0           4m22s
```

Pour se connecter à WordPress, il convient de suivre les instructions de la section **NOTES** imprimée à l'écran lors de l'exécution de la commande **helm install** :

```
trainee@gateway:~$ export NODE_PORT=$(kubectl get --namespace dof309 -o jsonpath="{.spec.ports[0].nodePort}"
services wordpress)
```

```
trainee@gateway:~$ export NODE_IP=$(kubectl get nodes --namespace dof309 -o
jsonpath="{.items[0].status.addresses[0].address}")

trainee@gateway:~$ echo "WordPress URL: http://$NODE_IP:$NODE_PORT/"
WordPress URL: http://192.168.59.100:32236/

trainee@gateway:~$ echo "WordPress Admin URL: http://$NODE_IP:$NODE_PORT/admin"
WordPress Admin URL: http://192.168.59.100:32236/admin
```

Si vous oubliez le mot de passe de l'interface d'administration de l'installation de WordPress, utilisez la commande suivante :

```
trainee@gateway:~$ echo Password: $(kubectl get secret --namespace dof309 wordpress -o
jsonpath="{.data.wordpress-password}" | base64 -d)
Password: my-pass
```

1.10 - La Commande helm upgrade

La mise-à-jour d'un Chart se fait en modifiant les valeurs dans le fichier YAML. Dans notre cas, le fichier s'appelle **wordpress-values.yaml**. Modifiez ce fichier pour porter le nombre de **Replicas** à 2 et pour diminuer la mémoire et le nombre de **millicores** du processeur par rapport à leurs valeurs par défaut de 512Mi et de 300m respectivement :

```
trainee@gateway:~$ vi wordpress-values.yaml
trainee@gateway:~$ cat wordpress-values.yaml
wordpressUsername: helm-user
wordpressPassword: my-pass
wordpressEmail: helm-user@i2tch.loc
wordpressFirstName: Helm
wordpressLastName: User
wordpressBlogName: Learn Helm!
service:
  type: NodePort
replicaCount: 2
resources:
```

```
requests:  
  memory: 256Mi  
  cpu: 100m
```

Utilisez maintenant la commande **helm upgrade** pour mettre à jour le deployment :

```
trainee@gateway:~$ helm upgrade wordpress bitnami/wordpress --values wordpress-values.yaml -n dof309 --version 24.1.5  
Release "wordpress" has been upgraded. Happy Helming!  
NAME: wordpress  
LAST DEPLOYED: Fri Dec 20 15:33:45 2024  
NAMESPACE: dof309  
STATUS: deployed  
REVISION: 2  
TEST SUITE: None  
NOTES:  
CHART NAME: wordpress  
CHART VERSION: 24.1.5  
APP VERSION: 6.7.1
```

Did you know there are enterprise versions of the Bitnami catalog? For enhanced secure software supply chain features, unlimited pulls from Docker, LTS support, or application customization, see Bitnami Premium or Tanzu Application Catalog. See <https://www.arrow.com/globalecs/na/vendors/bitnami> for more information.

**** Please be patient while the chart is being deployed ****

Your WordPress site can be accessed through the following DNS name from within your cluster:

```
wordpress.dof309.svc.cluster.local (port 80)
```

To access your WordPress site from outside the cluster follow the steps below:

1. Get the WordPress URL by running these commands:

```
export NODE_PORT=$(kubectl get --namespace dof309 -o jsonpath="{.spec.ports[0].nodePort}" services wordpress)
export NODE_IP=$(kubectl get nodes --namespace dof309 -o jsonpath="{.items[0].status.addresses[0].address}")
echo "WordPress URL: http://$NODE_IP:$NODE_PORT/"
echo "WordPress Admin URL: http://$NODE_IP:$NODE_PORT/admin"
```

2. Open a browser and access WordPress using the obtained URL.

3. Login with the following credentials below to see your blog:

```
echo Username: helm-user
echo Password: $(kubectl get secret --namespace dof309 wordpress -o jsonpath="{.data.wordpress-password}" |
base64 -d)
```

Vérifiez maintenant le nombre de Pods qui ont été démarrés :

```
trainee@gateway:~$ kubectl get pods -n dof309
```

| NAME | READY | STATUS | RESTARTS | AGE |
|----------------------------|-------|---------|---------------|-------|
| wordpress-75b57d8d9b-9t4jw | 0/1 | Running | 1 (18m ago) | 24m |
| wordpress-75b57d8d9b-jtx58 | 0/1 | Running | 1 (103s ago) | 5m18s |
| wordpress-779bd5776b-9jz8v | 0/1 | Running | 1 (56s ago) | 5m18s |
| wordpress-mariadb-0 | 0/1 | Running | 7 (3m31s ago) | 24m |

Visualisez les valeurs qui ont été utilisées lors de la mise-à-jour :

```
trainee@gateway:~$ helm get values wordpress -n dof309
```

USER-SUPPLIED VALUES:

```
replicaCount: 2
resources:
  requests:
    cpu: 100m
    memory: 256Mi
service:
  type: NodePort
wordpressBlogName: Learn Helm!
```

```
wordpressEmail: helm-user@i2tch.loc
wordpressFirstName: Helm
wordpressLastName: User
wordpressPassword: my-pass
wordpressUsername: helm-user
```

Exécutez de nouveau la mise-à-jour, cette fois **sans** spécifier le fichier de valeurs **wordpress-values.yaml** sur la ligne de commande :

```
trainee@gateway:~$ helm upgrade wordpress bitnami/wordpress -n dof309 --version 24.1.5
Release "wordpress" has been upgraded. Happy Helming!
NAME: wordpress
LAST DEPLOYED: Fri Dec 20 15:44:46 2024
NAMESPACE: dof309
STATUS: deployed
REVISION: 3
TEST SUITE: None
NOTES:
CHART NAME: wordpress
CHART VERSION: 24.1.5
APP VERSION: 6.7.1
```

Did you know there are enterprise versions of the Bitnami catalog? For enhanced secure software supply chain features, unlimited pulls from Docker, LTS support, or application customization, see Bitnami Premium or Tanzu Application Catalog. See <https://www.arrow.com/globalecs/na/vendors/bitnami> for more information.

**** Please be patient while the chart is being deployed ****

Your WordPress site can be accessed through the following DNS name from within your cluster:

```
wordpress.dof309.svc.cluster.local (port 80)
```

To access your WordPress site from outside the cluster follow the steps below:

1. Get the WordPress URL by running these commands:

```
export NODE_PORT=$(kubectl get --namespace dof309 -o jsonpath="{.spec.ports[0].nodePort}" services wordpress)
export NODE_IP=$(kubectl get nodes --namespace dof309 -o jsonpath="{.items[0].status.addresses[0].address}")
echo "WordPress URL: http://$NODE_IP:$NODE_PORT/"
echo "WordPress Admin URL: http://$NODE_IP:$NODE_PORT/admin"
```

2. Open a browser and access WordPress using the obtained URL.

3. Login with the following credentials below to see your blog:

```
echo Username: helm-user
echo Password: $(kubectl get secret --namespace dof309 wordpress -o jsonpath="{.data.wordpress-password}" |
base64 -d)
```

Visualisez les valeurs qui ont été utilisées lors de la mise-à-jour :

```
trainee@gateway:~$ helm get values wordpress -n dof309
USER-SUPPLIED VALUES:
replicaCount: 2
resources:
  requests:
    cpu: 100m
    memory: 256Mi
service:
  type: NodePort
wordpressBlogName: Learn Helm!
wordpressEmail: helm-user@i2tch.loc
wordpressFirstName: Helm
wordpressLastName: User
wordpressPassword: my-pass
wordpressUsername: helm-user
```



Important : Notez que la commande helm a réutilisé les valeurs de la première mise-à-



jour.

Vérifiez maintenant le nombre de Pods qui ont été démarrés :

```
trainee@gateway:~$ kubectl get pods -n dof309
NAME                                READY   STATUS              RESTARTS   AGE
wordpress-5f88d486d7-6hnbp         1/1     Running             0           6m52s
wordpress-mariadb-0                0/1     CrashLoopBackOff    12 (41s ago) 49m
```

1.11 - La Commande helm history

Helm utilise un système de **révisions**. Une nouvelle révision est créée à chaque installation, mise-à-jour ou quand il y a un **rollback**. Pour voir la liste des révisions, utilisez la commande suivante :

```
trainee@gateway:~$ helm history wordpress -n dof309
REVISION      UPDATED              STATUS          CHART          APP VERSION
DESCRIPTION
1             Fri Dec 20 15:14:17 2024    superseded     wordpress-24.1.5  6.7.1      Install
complete
2             Fri Dec 20 15:33:45 2024    superseded     wordpress-24.1.5  6.7.1      Upgrade
complete
3             Fri Dec 20 15:44:46 2024    superseded     wordpress-24.1.5  6.7.1      Upgrade
complete
4             Fri Dec 20 15:57:01 2024    deployed       wordpress-24.1.5  6.7.1      Upgrade
complete
```

La commande **helm get values** peut être utilisée pour consulter la liste des valeurs pour chaque révision :

```
trainee@gateway:~$ helm get values wordpress --revision 3 -n dof309
USER-SUPPLIED VALUES:
replicaCount: 2
```

```
resources:
  requests:
    cpu: 100m
    memory: 256Mi
service:
  type: NodePort
wordpressBlogName: Learn Helm!
wordpressEmail: helm-user@i2tch.loc
wordpressFirstName: Helm
wordpressLastName: User
wordpressPassword: my-pass
wordpressUsername: helm-user

trainee@gateway:~$ helm get values wordpress --revision 4 -n dof309
USER-SUPPLIED VALUES:
replicaCount: 1
```

La commande **helm get manifest** peut être utilisée pour consulter l'état des ressources Kubernetes pour chaque révision :

```
trainee@gateway:~$ helm get manifest wordpress --revision 4 -n dof309 | more
---
# Source: wordpress/charts/mariadb/templates/networkpolicy.yaml
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
  name: wordpress-mariadb
  namespace: "dof309"
  labels:
    app.kubernetes.io/instance: wordpress
    app.kubernetes.io/managed-by: Helm
    app.kubernetes.io/name: mariadb
    app.kubernetes.io/version: 11.4.4
    helm.sh/chart: mariadb-20.2.1
    app.kubernetes.io/part-of: mariadb
```

```
spec:
  podSelector:
    matchLabels:
      app.kubernetes.io/instance: wordpress
      app.kubernetes.io/managed-by: Helm
      app.kubernetes.io/name: mariadb
      app.kubernetes.io/version: 11.4.4
      helm.sh/chart: mariadb-20.2.1
  policyTypes:
    - Ingress
    - Egress
  egress:
    - {}
  ingress:
    - ports:
        - port: 3306
        - port: 3306
  ---
# Source: wordpress/templates/networkpolicy.yaml
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
  name: wordpress
  namespace: "dof309"
  labels:
    app.kubernetes.io/instance: wordpress
    app.kubernetes.io/managed-by: Helm
    app.kubernetes.io/name: wordpress
    app.kubernetes.io/version: 6.7.1
    helm.sh/chart: wordpress-24.1.5
spec:
  podSelector:
    matchLabels:
      app.kubernetes.io/instance: wordpress
```

```
    app.kubernetes.io/name: wordpress
policyTypes:
- Ingress
- Egress
egress:
- {}
ingress:
- ports:
    - port: 8080
    - port: 8443
--More--
```

1.12 - La Commande helm rollback

Pour revenir à une révision précédente il convient d'utilise la commande **helm rollback** :

```
trainee@gateway:~$ helm rollback wordpress 3 -n dof309
Rollback was a success! Happy Helming!
```

Il convient ensuite de vérifier que les valeurs actuelles sont bien celles de la révision ciblée :

```
trainee@gateway:~$ helm get values wordpress -n dof309
USER-SUPPLIED VALUES:
replicaCount: 2
resources:
  requests:
    cpu: 100m
    memory: 256Mi
service:
  type: NodePort
wordpressBlogName: Learn Helm!
wordpressEmail: helm-user@i2tch.loc
wordpressFirstName: Helm
```

```
wordpressLastName: User
wordpressPassword: my-pass
wordpressUsername: helm-user
```

Le Rollback est clairement indiqué dans la sortie de la commande **helm history** :

```
trainee@gateway:~$ helm history wordpress -n dof309
REVISION      UPDATED              STATUS      CHART              APP VERSION
DESCRIPTION
1             Fri Dec 20 15:14:17 2024      superseded  wordpress-24.1.5   6.7.1      Install
complete
2             Fri Dec 20 15:33:45 2024      superseded  wordpress-24.1.5   6.7.1      Upgrade
complete
3             Fri Dec 20 15:44:46 2024      superseded  wordpress-24.1.5   6.7.1      Upgrade
complete
4             Fri Dec 20 15:57:01 2024      superseded  wordpress-24.1.5   6.7.1      Upgrade
complete
5             Fri Dec 20 16:09:02 2024      deployed   wordpress-24.1.5   6.7.1      Rollback
to 3
```

1.13 - La Commande helm uninstall

La désinstallation d'un Chart se fait en utilisant la commande **helm uninstall**. Par défaut cette commande supprime aussi l'historique des révisions sauf si l'option **-keep-history** est spécifiée sur la ligne de commande. Supprimez donc le Chart wordpress ainsi que l'historique des révisions :

```
trainee@gateway:~$ helm uninstall wordpress -n dof309
release "wordpress" uninstalled
```

La commande **helm list** démontre cette suppression :

```
trainee@gateway:~$ helm list -n dof309
NAME      NAMESPACE      REVISION      UPDATED STATUS  CHART      APP VERSION
```

ainsi qua la sortie de la commande **kubectl get deployments** :

```
trainee@gateway:~$ kubectl get deployments -l app=wordpress -n dof309
No resources found in dof309 namespace.
```

Par contre la commande **helm uninstall** ne supprime **pas** le **PersistentVolumeClaim** créé lors de l'installation du Chart :

```
trainee@gateway:~$ kubectl get pvc -n dof309
NAME                                STATUS    VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS          AGE
data-wordpress-mariadb-0            Bound    pvc-71bba44c-9af2-44c9-9e59-103efdf1b29b  8Gi        RWO             standard              56m
<unset>                             56m
```

Pour supprimer le **PersistentVolumeClaim**, il est nécessaire d'utilise la commande **kubectl delete pvc** avec l'option **-field-selector** :

```
trainee@gateway:~$ kubectl delete pvc --field-selector metadata.name=data-wordpress-mariadb-0 -n dof309
persistentvolumeclaim "data-wordpress-mariadb-0" deleted
```

Le **PersistentVolumeClaim** se trouve de cette manière supprimé :

```
trainee@gateway:~$ kubectl get pvc -n dof309
No resources found in dof309 namespace
```

LAB #2 - Supervision de Kubernetes avec le Stack EFK

2.1 - Présentation du Stack EFK

Le stack EFK est composé de trois éléments :

- **Elasticsearch** qui assure :
 - le stockage des métriques et journaux,

- **Fluentd** qui assure :
 - la collecte des données brutes de Kubernetes,
 - l'envoi des métriques et journaux à Elasticsearch,
- **Kibana** qui permet :
 - la visualisation des données stockées dans Elasticsearch.

Avant de poursuivre, arrêtez minikube et augmentez la mémoire de sa machine virtuelle à 12 294 Mo et le nombre de vCPUs à 4 :

```
trainee@gateway:~$ minikube stop
[ ] Stopping node "minikube" ...
[ ] 1 node stopped.

trainee@gateway:~$ minikube delete
[ ] Deleting "minikube" in virtualbox ...
[ ] Removed all traces of the "minikube" cluster.

trainee@gateway:~$ minikube config set memory 12294
[ ] These changes will take effect upon a minikube delete and then a minikube start

trainee@gateway:~$ minikube config set cpus 4
[ ] These changes will take effect upon a minikube delete and then a minikube start

trainee@gateway:~$ minikube start
[ ] minikube v1.26.0 on Debian 11.3 (kvm/amd64)
[ ] Using the virtualbox driver based on user configuration
[ ] Starting control plane node minikube in cluster minikube
[ ] Creating virtualbox VM (CPUs=4, Memory=12294MB, Disk=20000MB) ...
[ ] Preparing Kubernetes v1.24.1 on Docker 20.10.16 ...
  ■ Generating certificates and keys ...
  ■ Booting up control plane ...
  ■ Configuring RBAC rules ...
```

You have selected "virtualbox" driver, but there are better options !

```
For better performance and support consider using a different driver:
```

- kvm2
- qemu2

```
To turn off this warning run:
```

```
$ minikube config set WantVirtualBoxDriverWarning false
```

```
To learn more about on minikube drivers checkout https://minikube.sigs.k8s.io/docs/drivers/  
To see benchmarks checkout https://minikube.sigs.k8s.io/docs/benchmarks/cpuusage/
```

- Verifying Kubernetes components...
 - Using image gcr.io/k8s-minikube/storage-provisioner:v5
- Enabled addons: storage-provisioner, default-storageclass
- Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default

2.2 - Installation du Chart elasticsearch

Commencez par installer le **repository** stable de <https://charts.helm.sh> :

```
trainee@gateway:~$ helm repo add stable https://charts.helm.sh/stable  
"stable" has been added to your repositories
```

Installez ensuite le Chart helm **elasticsearch** :

```
trainee@gateway:~$ helm install elasticsearch stable/elasticsearch  
WARNING: This chart is deprecated  
NAME: elasticsearch  
LAST DEPLOYED: Sat Jul 16 11:27:22 2022  
NAMESPACE: default  
STATUS: deployed
```

REVISION: 1

NOTES:

This Helm chart is deprecated. Please use <https://github.com/elastic/helm-charts/tree/master/elasticsearch> instead.

The elasticsearch cluster has been installed.

Elasticsearch can be accessed:

* Within your cluster, at the following DNS name at port 9200:

elasticsearch-client.default.svc

* From outside the cluster, run these commands in the same shell:

```
export POD_NAME=$(kubectl get pods --namespace default -l
"app=elasticsearch,component=client,release=elasticsearch" -o jsonpath="{.items[0].metadata.name}")
echo "Visit http://127.0.0.1:9200 to use Elasticsearch"
kubectl port-forward --namespace default $POD_NAME 9200:9200
```

Patientez quelques minutes afin que tous les Pods affichent un état de **READY** :

```
trainee@gateway:~$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
elasticsearch-client-7869bf97bb-tb25p 1/1     Running   0           5m50s
elasticsearch-client-7869bf97bb-zvl6h 1/1     Running   0           5m49s
elasticsearch-data-0                  1/1     Running   0           5m49s
elasticsearch-data-1                  1/1     Running   0           2m37s
elasticsearch-master-0                1/1     Running   0           5m49s
elasticsearch-master-1                1/1     Running   0           2m37s
elasticsearch-master-2                1/1     Running   0           65s
```

Vérifiez ensuite la présence des services **elasticsearch-client** et **elasticsearch-discovery** :

```
trainee@gateway:~$ kubectl get services
NAME                TYPE          CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
elasticsearch-client ClusterIP      10.106.34.82    <none>           9200/TCP         8m23s
elasticsearch-discovery ClusterIP      None            <none>           9300/TCP         8m23s
kubernetes           ClusterIP      10.96.0.1       <none>           443/TCP          10m
```

2.3 - Installation du Chart fluentd-elasticsearch

Installez le Chart helm **fluentd-elasticsearch** à partir du repository **stable** :

```
trainee@gateway:~$ helm install fluentd stable/fluentd-elasticsearch --set elasticsearch.host=elasticsearch-client
WARNING: This chart is deprecated
W0716 12:20:08.692464    7135 warnings.go:70]
spec.template.metadata.annotations[scheduler.alpha.kubernetes.io/critical-pod]: non-functional in v1.16+; use the
"priorityClassName" field instead
NAME: fluentd
LAST DEPLOYED: Sat Jul 16 12:20:07 2022
NAMESPACE: default
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
1. To verify that Fluentd has started, run:

    kubectl --namespace=default get pods -l "app.kubernetes.io/name=fluentd-
elasticsearch,app.kubernetes.io/instance=fluentd"

THIS APPLICATION CAPTURES ALL CONSOLE OUTPUT AND FORWARDS IT TO elasticsearch . Anything that might be
identifying,
```

including things like IP addresses, container images, and object names will NOT be anonymized.



Important : Fluentd a besoin de connaître l'emplacement de l'hôte Elasticsearch. Dans notre cas, la valeur est **elasticsearch-client**. Notez donc l'utilisation sur la ligne de commande de l'option **-set elasticsearch.host=**.

Patientez quelques secondes afin que le Pod **fluentd-fluentd-elasticsearch-xxxxx** affiche un état de **READY** :

```
trainee@gateway:~$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
elasticsearch-client-7869bf97bb-tb25p 1/1     Running   0          10m
elasticsearch-client-7869bf97bb-zvl6h 1/1     Running   0          10m
elasticsearch-data-0                   1/1     Running   0          10m
elasticsearch-data-1                   1/1     Running   0          6m51s
elasticsearch-master-0                 1/1     Running   0          10m
elasticsearch-master-1                 1/1     Running   0          6m51s
elasticsearch-master-2                 1/1     Running   0          5m19s
fluentd-fluentd-elasticsearch-znvq5    1/1     Running   0          62s
```



Important : Notez que dans le cas de l'utilisation de minikube, nous n'avons qu'un seul nœud dans le cluster Kubernetes. Pour cette raison, nous avons installé qu'un seul Pod Fluentd avec Helm sans passer par un DaemonSet.

2.4 - Installation du Chart kibana

Kibana a aussi besoin de connaître l'emplacement de l'hôte Elasticsearch. Cette fois-ci, la valeur sera incluse dans un fichier de **values**. Créez donc le fichier **kibana-values.yaml** :

```
trainee@gateway:~$ vi kibana-values.yaml
trainee@gateway:~$ cat kibana-values.yaml
files:
  kibana.yml:
    ## Default Kibana configuration from kibana-docker.
    server.name: kibana
    server.host: "0"
    ## For kibana < 6.6, use elasticsearch.url instead
    elasticsearch.hosts: http://elasticsearch-client:9200
service:
  type: NodePort
```



Important : Notez la création d'un service de type **NodePort**. Dans le cas de l'utilisation du DaemonSet, nous aurions créé un service **ClusterIP*** de type **LoadBalancer**.

Installez donc le Chart Helm kibana à partir du repository **stable** en référençant le fichier **values kibana-values.yaml** :

```
trainee@gateway:~$ helm install kibana stable/kibana -f kibana-values.yaml
WARNING: This chart is deprecated
NAME: kibana
LAST DEPLOYED: Sat Jul 16 12:22:57 2022
NAMESPACE: default
STATUS: deployed
REVISION: 1
NOTES:
THE CHART HAS BEEN DEPRECATED!
```

Find the new official version @ <https://github.com/elastic/helm-charts/tree/master/kibana>

To verify that kibana has started, run:

```
kubectl --namespace=default get pods -l "app=kibana"
```

Kibana can be accessed:

* From outside the cluster, run these commands in the same shell:

```
export NODE_PORT=$(kubectl get --namespace default -o jsonpath="{.spec.ports[0].nodePort}" services kibana)
export NODE_IP=$(kubectl get nodes --namespace default -o jsonpath="{.items[0].status.addresses[0].address}")
echo http://$NODE_IP:$NODE_PORT
```

Notez les instructions à la fin de la sortie précédente qui indiquent comment atteindre Kibana de l'extérieur du cluster. Créez donc les deux variables **NODE_PORT** et **NODE_IP** et exécutez la commande **echo** afin d'obtenir le socket de connexion à Kibana. Notez cette information :

```
trainee@gateway:~$ export NODE_PORT=$(kubectl get --namespace default -o jsonpath="{.spec.ports[0].nodePort}"
services kibana)
trainee@gateway:~$ export NODE_IP=$(kubectl get nodes --namespace default -o
jsonpath="{.items[0].status.addresses[0].address}")
trainee@gateway:~$ echo http://$NODE_IP:$NODE_PORT
http://192.168.59.101:30812
```

Patientez quelques minutes afin que le Pod **kibana-yyyyyyyyyy-xxxxx** affiche un état de **READY** :

```
trainee@gateway:~$ kubectl get pods
```

| NAME | READY | STATUS | RESTARTS | AGE |
|---------------------------------------|-------|---------|----------|-------|
| elasticsearch-client-7869bf97bb-tb25p | 1/1 | Running | 0 | 14m |
| elasticsearch-client-7869bf97bb-zvl6h | 1/1 | Running | 0 | 14m |
| elasticsearch-data-0 | 1/1 | Running | 0 | 14m |
| elasticsearch-data-1 | 1/1 | Running | 0 | 11m |
| elasticsearch-master-0 | 1/1 | Running | 0 | 14m |
| elasticsearch-master-1 | 1/1 | Running | 0 | 11m |
| elasticsearch-master-2 | 1/1 | Running | 0 | 9m28s |
| fluentd-fluentd-elasticsearch-znvq5 | 1/1 | Running | 0 | 5m11s |
| kibana-74cd958bd4-qxmtc | 1/1 | Running | 0 | 2m21s |

Vérifiez ensuite la présence du service **NodePort** :

```
trainee@gateway:~$ kubectl get services
NAME                TYPE          CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
elasticsearch-client ClusterIP      10.106.34.82    <none>        9200/TCP         14m
elasticsearch-discovery ClusterIP      None            <none>        9300/TCP         14m
kibana              NodePort      10.97.180.37    <none>        443:30812/TCP   2m49s
kubernetes          ClusterIP      10.96.0.1       <none>        443/TCP         16m
```

2.5 - Génération de Traces dans Kubernetes

Créez maintenant un fichier **counter.yaml** afin de créer un Pod qui générera des logs de démonstration en continu :

```
trainee@gateway:~$ vi counter.yaml
trainee@gateway:~$ cat counter.yaml
## counter.yaml
apiVersion: v1
kind: Pod
metadata:
  name: counter
spec:
  containers:
  - name: count
    image: busybox
    args: [/bin/sh, -c, 'i=0; while true; do echo "Demo log $i: $(date)"; i=$((i+1)); sleep 1; done']
```

Appliquez ce fichier avec la commande **kubectl** :

```
trainee@gateway:~$ kubectl apply -f counter.yaml
pod/counter created
```

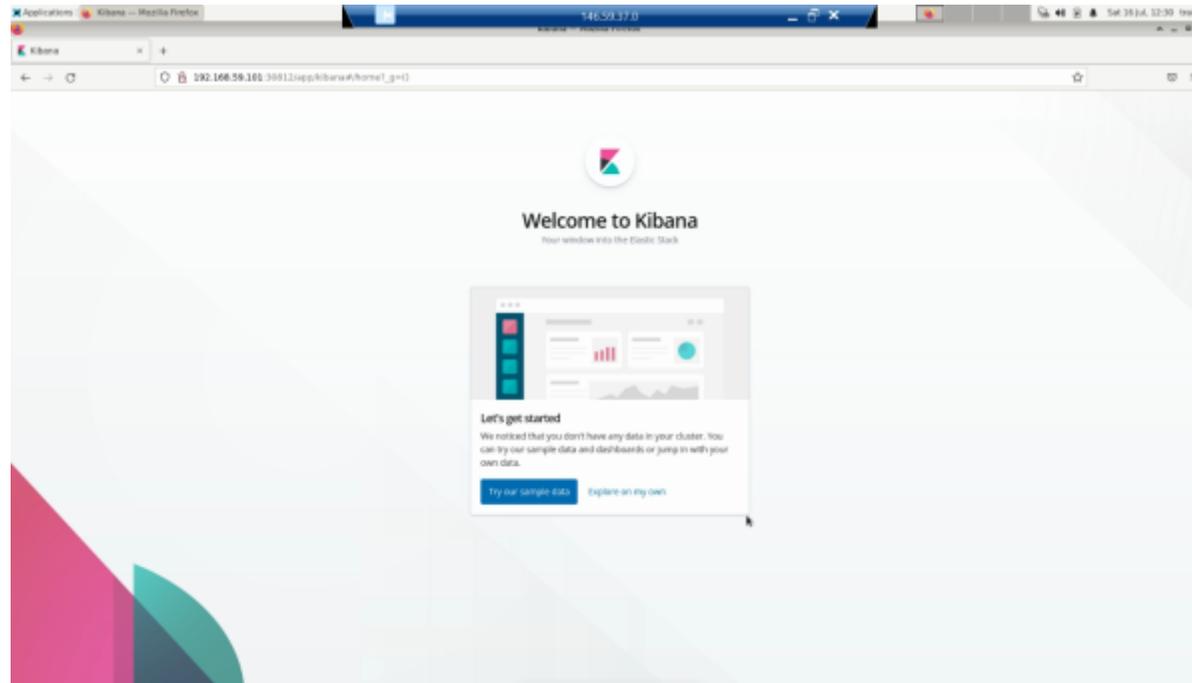
Vérifiez que le Pod **counter** soit bien en état de **READY** :

```
trainee@gateway:~$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
counter                             1/1     Running   0           17s
elasticsearch-client-7869bf97bb-tb25p 1/1     Running   0           16m
elasticsearch-client-7869bf97bb-zvl6h 1/1     Running   0           16m
elasticsearch-data-0                 1/1     Running   0           16m
elasticsearch-data-1                 1/1     Running   0           13m
elasticsearch-master-0               1/1     Running   0           16m
elasticsearch-master-1               1/1     Running   0           13m
elasticsearch-master-2               1/1     Running   0           11m
fluentd-fluentd-elasticsearch-znvq5  1/1     Running   0           7m15s
kibana-74cd958bd4-qxmtc              1/1     Running   0           4m25s
```

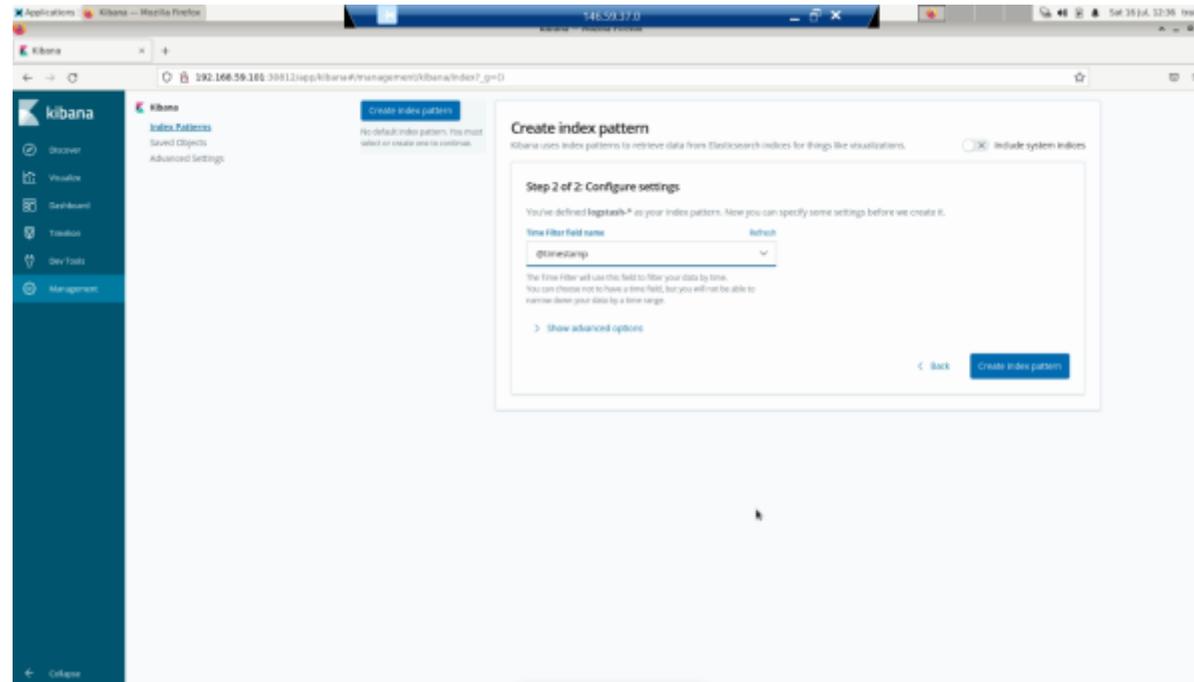
2.6 - Visualisation des Données avec Kibana

Connectez-vous à votre VM Gateway en mode VNC en utilisant Apache Guacamole ou via une connexion RDP.

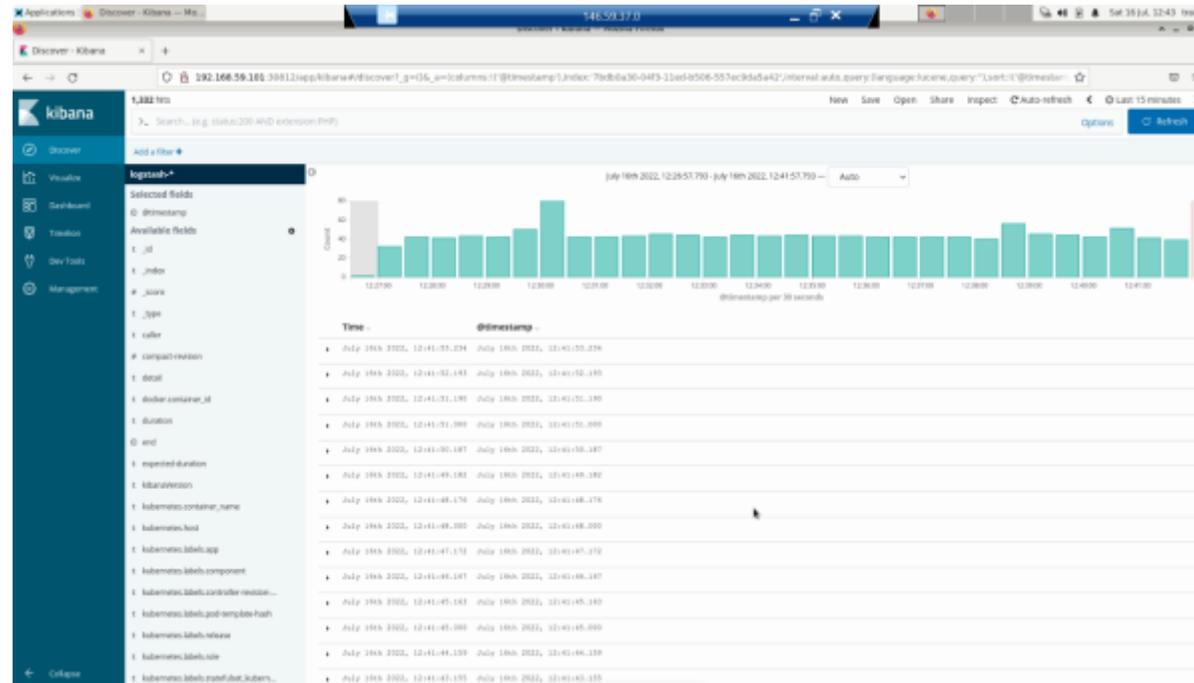
Ouvrez le navigateur FireFox et saisissez l'adresse que vous avez obtenu en exécutant la commande **echo** ci-dessus.



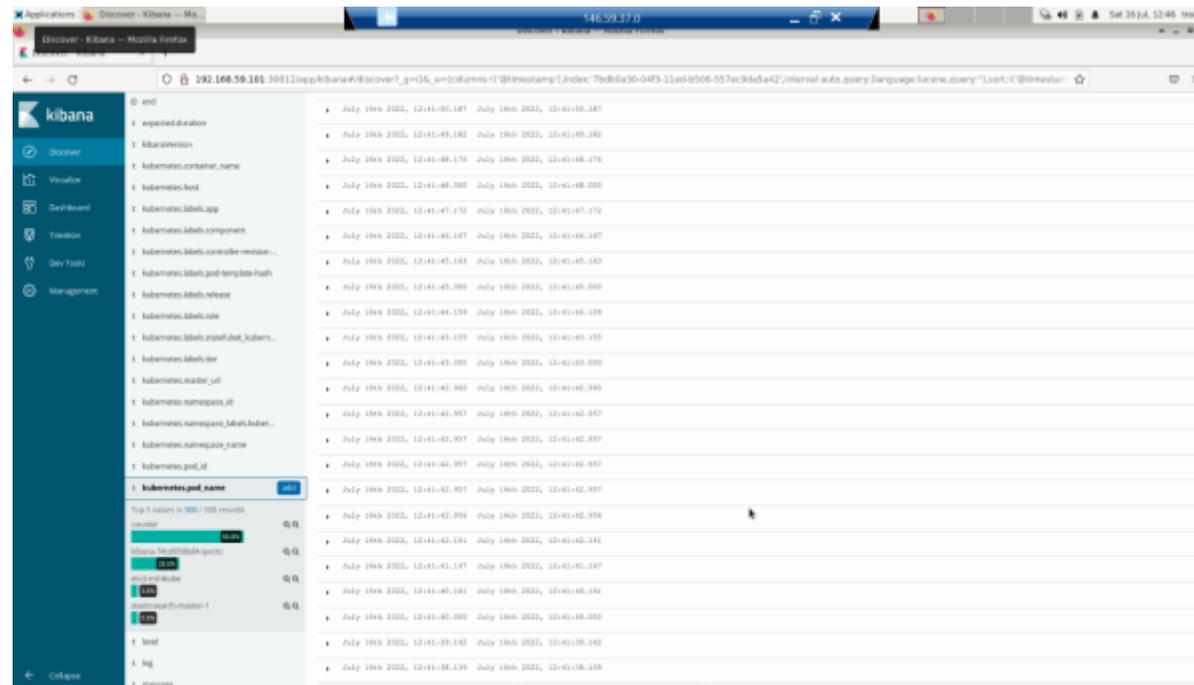
Cliquez sur **Explore on my own** > **Management** > **Index Patterns**, ensuite, dans **Create index pattern**, renseignez la valeur **logstash-*** et cliquez sur le bouton **Next step** :



Cliquez sur le lien **Discover** dans le menu de Kibana. Sélectionnez **logstash-*** puis choisissez **@timestamp** dans **Available fields** et cliquez sur le bouton **Add** :



Appliquez un filtre en cliquant sur le champ **kubernetes.pod_name** :



The screenshot shows the Kibana interface with a list of Kubernetes pod names. The 'counter' field is highlighted in green, and the 'Add' button is visible next to it. The list of pods includes:

| Pod Name | Creation Time | Last Update Time |
|--------------------------------------|------------------------------|------------------------------|
| kubernetes.container_name | July 19th 2022, 12:41:49.182 | July 19th 2022, 12:41:49.182 |
| kubernetes.labels.app | July 19th 2022, 12:41:47.172 | July 19th 2022, 12:41:47.172 |
| kubernetes.labels.component | July 19th 2022, 12:41:47.147 | July 19th 2022, 12:41:47.147 |
| kubernetes.labels.controller-version | July 19th 2022, 12:41:47.143 | July 19th 2022, 12:41:47.143 |
| kubernetes.labels.pod-template-hash | July 19th 2022, 12:41:47.000 | July 19th 2022, 12:41:47.000 |
| kubernetes.labels.release | July 19th 2022, 12:41:47.000 | July 19th 2022, 12:41:47.000 |
| kubernetes.labels.role | July 19th 2022, 12:41:46.139 | July 19th 2022, 12:41:46.139 |
| kubernetes.labels.role.kubern... | July 19th 2022, 12:41:47.155 | July 19th 2022, 12:41:47.155 |
| kubernetes.labels.dir | July 19th 2022, 12:41:47.000 | July 19th 2022, 12:41:47.000 |
| kubernetes.master_pod | July 19th 2022, 12:41:47.000 | July 19th 2022, 12:41:47.000 |
| kubernetes.name.namespace.kuber... | July 19th 2022, 12:41:42.907 | July 19th 2022, 12:41:42.907 |
| kubernetes.name.namespace | July 19th 2022, 12:41:42.907 | July 19th 2022, 12:41:42.907 |
| kubernetes.pod_id | July 19th 2022, 12:41:42.907 | July 19th 2022, 12:41:42.907 |
| kubernetes.pod_name | July 19th 2022, 12:41:42.907 | July 19th 2022, 12:41:42.907 |
| counter | July 19th 2022, 12:41:42.908 | July 19th 2022, 12:41:42.908 |
| status.phase | July 19th 2022, 12:41:42.151 | July 19th 2022, 12:41:42.151 |
| status.pod | July 19th 2022, 12:41:41.147 | July 19th 2022, 12:41:41.147 |
| status.pod.phase | July 19th 2022, 12:41:40.181 | July 19th 2022, 12:41:40.181 |
| status.pod.phase | July 19th 2022, 12:41:40.000 | July 19th 2022, 12:41:40.000 |
| status.pod.phase | July 19th 2022, 12:41:39.142 | July 19th 2022, 12:41:39.142 |
| status.pod.phase | July 19th 2022, 12:41:38.129 | July 19th 2022, 12:41:38.129 |

Cliquez sur la loupe + en face de la ligne **counter** puis sur le bouton **Add** en face du nom du champs :

