Version - **2025.01**

Last update : 2025/01/18 17:09

DOE309 - Package Management for Kubernetes with Helm

Contents

Overview

Helm is a package manager for Kubernetes. Helm 3.x is compatible with Kubernetes versions **n-3** where **n** is the Kubernetes version against which the Helm version has been compiled, while the Helm 2.x manager is only compatible with Kubernetes versions **n-1**.

LAB #1 - Working with Helm

Go to your Gateway VM. Start by checking the status of Minikube:

```
trainee@gateway:~$ minikube status
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured
```

1.1 - Installing Helm

Install **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
...
```

Then check the location of the binary and its version:

```
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"}
```

To enable command-line auto-completion with the helm command, similar to that under bash, run the following command as trainee:

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

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

To view the environment variables for the **helm** command, run the **helm env** command:

```
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**: Note that the value of the **$HELM_NAMESPACE** variable is **default**. To change the namespace used when using the helm command, you can either modify the value of this variable, or use the **–namespace** or **-n** option each time you run the command.

To view the **kubectl** command environment, run the **kubectl config view** command:

```
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 - The helm search hub Command

A large number of Helm Charts are grouped together on the **The Artifact Hub** website. The contents of the Hub can be listed on the command line:

```
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--
```

Then search for any Chart containing the string **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... Rajware Helm course	0.2.4	5.7	MySQL packaged in
https://artifacthub.io/packages/helm/ot-contain... Kubernetes to deploy mysql.	0.1.0	1.16.0	A Helm chart for
https://artifacthub.io/packages/helm/kubeblocks... used, open-source relational ...	1.0.0-alpha.0	5.7.44	MySQL is a widely
https://artifacthub.io/packages/helm/someblackm... scalable, and easy to use open-...	1.7.1	5.7.33	Fast, reliable,
https://artifacthub.io/packages/helm/kubesphere... scalable, and easy to use open-...	1.6.8	5.7.31	Fast, reliable,
https://artifacthub.io/packages/helm/kubegemsap... Highly available MySQL cluster	4.5.2	5.7.26	Chart to create a
https://artifacthub.io/packages/helm/wso2/mysql scalable, and easy to use open-...	1.6.9	5.7.30	Fast, reliable,
https://artifacthub.io/packages/helm/homeenterp... Server	0.5.0	8.0.29	MySQL Database
https://artifacthub.io/packages/helm/helm-chart... Highly available MySQL cluster	1.0.0	5.7.26	Chart to create a
https://artifacthub.io/packages/helm/anvibo/mysql Kubernetes	0.1.0	1.16.0	A Helm chart for
https://artifacthub.io/packages/helm/huangcheng... Kubernetes	0.1.0	1.16.0	A Helm chart for
https://artifacthub.io/packages/helm/mysql-oper... Cluster Helm Chart for deploying M...	2.2.2	9.1.0	MySQL InnoDB
https://artifacthub.io/packages/helm/mysql-oper... Helm Chart for deploying MySQL I...	2.2.2	9.1.0-2.2.2	MySQL Operator
https://artifacthub.io/packages/helm/softonic/m... backups from any mysql instance to A...	2.2.3	0.3.0	Take mysql
https://artifacthub.io/packages/helm/kfirfer/my... MySQL Cluster	0.0.4	8.0.20	A Helm chart for
https://artifacthub.io/packages/helm/wordpress-... for configuring wordpress and...	1.0.0	1.1	This is a package
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 WSO2...

| 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 WSO2...

https://artifacthub.io/packages/helm/wso2/mysql-is MySQL based deployment of WSO2...	5.11.0-5	5.7	A Helm chart for
https://artifacthub.io/packages/helm/wso2/mysql-ob MySQL based deployment of WSO2...	1.5.0-1	5.7	A Helm chart for
https://artifacthub.io/packages/helm/banzaiclou... deploying the Oracle MySQL Ope...	0.1.0	0.2.0	A Helm chart for
https://artifacthub.io/packages/helm/grdscloud/... Operator Helm chart for Kubern...	0.1.0	1.0.0	A MySQL Database
https://artifacthub.io/packages/helm/openshift/... expermental, do not use it in p...	0.0.4	0.0.4	This content is
https://artifacthub.io/packages/helm/openshift/... MySQL Service Endpoint Definit...	0.1.0	0.1.0	A Helm chart for
https://artifacthub.io/packages/helm/ygqygq2/my... servicemonitor for mysql exporter	1.0.0	5.7.24	A Helm chart
https://artifacthub.io/packages/helm/kubeblocks... used, open-source relational ...	0.9.0	8.0.32	MySQL is a widely
https://artifacthub.io/packages/helm/kubeblocks... used, open-source relational ...	0.8.0	8.0.32	MySQL is a widely
https://artifacthub.io/packages/helm/wener/prom... prometheus mysql exporter with...	2.8.0	v0.16.0	A Helm chart for
https://artifacthub.io/packages/helm/banzaiclou... prometheus mysql exporter with...	0.2.4	v0.11.0	A Helm chart for
https://artifacthub.io/packages/helm/prometheus... prometheus mysql exporter with...	1.0.1	v0.12.1	A Helm chart for
https://artifacthub.io/packages/helm/prometheus... 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/cloudnativ... prometheus mysql exporter with...	0.3.2	v0.11.0	A Helm chart for
https://artifacthub.io/packages/helm/braedon/pr... Mysql and export the results as ...	0.2.0		Run queries on
https://artifacthub.io/packages/helm/wenerme/pr... prometheus mysql exporter with...	2.8.0	v0.16.0	A Helm chart for

https://artifacthub.io/packages/helm/openshift/... experimental, do not use it in ...	0.0.2	0.0.2	This content is
https://artifacthub.io/packages/helm/bytebuilde... Community	2021.3.11	v2021.03.11	Stash MySQL Addon
https://artifacthub.io/packages/helm/fermosit/z... server	3.0.2	6.4.0-latest	Zabbix monitoring
https://artifacthub.io/packages/helm/kubeblocks... MySQL-Scale Cluster Helm chart for ...	0.5.3	8.0.30	An ApeCloud
https://artifacthub.io/packages/helm/clusterped... Clusterpedia	0.1.1	v0.6.0	A Helm chart for
https://artifacthub.io/packages/helm/t3n/mysql-...	2.1.0		
https://artifacthub.io/packages/helm/kfirfer/my... Kubernetes	0.3.0	0.0.4	A Helm chart for
https://artifacthub.io/packages/helm/wso2/mysql-ei WSO2 Enterprise Integrator Dat...	6.6.0-3	5.7	A Helm chart for
https://artifacthub.io/packages/helm/sikalabs/w...	0.1.2		
https://artifacthub.io/packages/helm/choerodon/... Kubernetes	0.1.0	1.0	A Helm chart for
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 MySQ...	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/edgelesssy... 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...	1.1.2		MariaDB is developed as open source software an...
https://artifacthub.io/packages/helm/cloudposse...	0.1.4		A Helm chart for Linux, Apache, MySQL, PHP (aka...
https://artifacthub.io/packages/helm/login-test...	0.1.0	1.0.0	A Helm chart for login_test_backend with MySQL ...
https://artifacthub.io/packages/helm/bitnami/ma...	20.2.1	11.4.4	MariaDB is an open source, community-developed ...
https://artifacthub.io/packages/helm/riftbit/ma...	9.6.0	10.5.12	Fast, reliable, scalable, and easy to use open-...
https://artifacthub.io/packages/helm/cloudnativ...	6.1.0	10.3.15	Fast, reliable, scalable, and easy to use open-...
https://artifacthub.io/packages/helm/camptocamp...	1.0.0		Fast, reliable, scalable, and easy to use open-...
https://artifacthub.io/packages/helm/bitnami-ak...	11.4.0	10.6.11	MariaDB is an open source, community-developed ...
https://artifacthub.io/packages/helm/kvalitetsi...	11.2.2	10.6.9	MariaDB is an open source, community-developed ...
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... MariaDB in a cloud native way	0.36.0	0.36.0	Run and operate
https://artifacthub.io/packages/helm/mariadb-op... CRDs	0.36.0	0.0.0	mariadb-operator
https://artifacthub.io/packages/helm/moco/moco Kubernetes using GTID-based s...	0.15.1	0.25.1	MySQL operator on
https://artifacthub.io/packages/helm/sqream-cha... monitor service, part of java ba...	0.7.1	1.16.0	Helm chart for
https://artifacthub.io/packages/helm/monosql-op... deploying Monograph Operator t...	1.0.5	1.0.4	A Helm chart for
https://artifacthub.io/packages/helm/viveksahu2... chart to deploy wordpress and...	1.0.0	2	This is my custom
https://artifacthub.io/packages/helm/mysql2s3bk... JOBS	0.1.0	alpha	MySQL database
https://artifacthub.io/packages/helm/kubeblocks... MySQL high availability and r...	3.2.6	3.2.6	Orchestrator is a
https://artifacthub.io/packages/helm/kubeblocks...	1.0.0-alpha.0	3.2.6	orchestrator
https://artifacthub.io/packages/helm/christianh... deploy Passbolt with MySQL and ...	1.13.2		A Helm chart to
https://artifacthub.io/packages/helm/vquie/phpipam open-source web IP address manage...	1.0.3	1.5.2	phpipam is an
https://artifacthub.io/packages/helm/fsdrw08/po... one (powerdns + powerdns-admin ...	0.1.3	4.5.3	PowerDNS all in
https://artifacthub.io/packages/helm/percona/px... deploying the Percona Operator...	1.16.0	1.16.0	A Helm chart for
https://artifacthub.io/packages/helm/kubesphere... Kubernetes to run StoneDB stan...	0.1.0	5.7_1.0.1	A Helm chart for
https://artifacthub.io/packages/helm/stonedb/st... Kubernetes to run StoneDB	0.1.0	5.7_1.0.1	A Helm chart for
https://artifacthub.io/packages/helm/banzaiclou... for Kubernetes	0.0.2		A TiDB Helm chart
https://artifacthub.io/packages/helm/kubeblocks... source, cloud-native, distribut...	1.0.0-alpha.0	v7.1.5	TiDB is an open-
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**: Note that the string you're looking for can be found in the URL as well as in the description.

Now look for the Chart **WordPress**. By default, columns wider than 50 characters are truncated:

```
trainee@gateway:~$ helm search hub wordpress
URL                                                     CHART VERSION    APP VERSION         DESCRIPTION
https://artifacthub.io/packages/helm/kube-wordp...      0.1.0            1.1                 this is my
wordpress package
https://artifacthub.io/packages/helm/wordpress-...      1.0.2            1.0.0               A Helm chart for
deploying Wordpress+Mariadb st...
https://artifacthub.io/packages/helm/shubham-wo...      0.1.0            1.16.0              A Helm chart for
Kubernetes
https://artifacthub.io/packages/helm/bitnami-ak...      15.2.13          6.1.0               WordPress is the
world's most popular blogging ...
https://artifacthub.io/packages/helm/bitnami/wo...      24.1.5           6.7.1               WordPress is the
world's most popular blogging ...
https://artifacthub.io/packages/helm/sikalabs/w...      0.2.0                                Simple Wordpress
https://artifacthub.io/packages/helm/riftbit/wo...      12.1.16          5.8.1               Web publishing
platform for building blogs and ...
https://artifacthub.io/packages/helm/schichtel/...      0.7.1            6.6.1               A Helm chart for
WordPress
https://artifacthub.io/packages/helm/skywordpre...      0.1.0            1.16.0              A Helm chart for
```

WordPress

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...     6.3.1      6.3.1              Web publishing
platform for building blogs and ...
https://artifacthub.io/packages/helm/wordpressh...     0.1.0      1.16.0             A Helm chart for
Kubernetes
https://artifacthub.io/packages/helm/wordpress-...     0.1.0      1.1                this is chart
create the wordpress with suitabl...
https://artifacthub.io/packages/helm/wordpress-...     0.0.1                         Helm Chart for
wordpress-gatsby
https://artifacthub.io/packages/helm/bitpoke/bi...     1.8.18     1.8.18             The Bitpoke App
for WordPress provides a versat...
https://artifacthub.io/packages/helm/sonu-wordp...     1.0.0      2                  This is my custom
chart to deploy wordpress and...
https://artifacthub.io/packages/helm/wordpress/...     0.2.0      1.1.0              Wordpress for
Kubernetes
https://artifacthub.io/packages/helm/uvaise-wor...     0.2.0      1.1.0              Wordpress for
Kubernetes
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 Infras...
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
```

To view all of each line, use the **--max-col-width=0** option:

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

It is also possible to obtain the output in YAML format:

```
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 - Searching the Artifact Hub

Now go to the **The Artifact Hub** website using a web browser and search for **mysql** :

In the returned result, click on **mysql** from the **Bitnami** repository:

You will find information about this Chart as well as instructions on how to install it:

1.4 - Adding and Deleting a Repository

Using the information taken from the Hub, now install the **bitnami** repository:

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

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

To list installed repositories, use the **helm repo list** command:

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

To update repositories, use the **helm repo update** command:

```
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!❀
```

To remove a repository, use the **helm repo remove <repo>** command.

```
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 - The helm search repo Command

Add the Bitnami repository again:

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

To view the contents of this repository in YAML format, use the **helm search repo** command:

```
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--
```

Now look for Wordpress in the Bitnami repository:

```
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...
```

The previously executed command returns only the latest version of Wordpress. To see the list of available versions, use the **–versions** option:

```
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 - The helm show Command

To view the metadata of a Chart version, use the **helm show chart** command:

```
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
```

To view the Chart README file, use the **helm show readme** command:

```
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 manag
er.

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://kubeapps.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 produc
tion 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](h
ttps://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--
```

To view the Chart values, use the **helm show values** command:

```
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
```

```yaml
## @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 an
d 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** : Default values can be overridden by values in a YAML format file.

To view the metadata, README file and Chart values, use the **helm show all** command:

```
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 - Installing a Chart

Create a namespace called **doe309** in which you will install WordPress :

```
trainee@gateway:~$ kubectl create namespace doe309
namespace/doe309 created
```

Create the file **wordpress-values.yaml** containing values that will override those in the WordPress Chart:

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

```
wordpressEmail: helm-user@ittraining.loc
wordpressFirstName: Helm
wordpressLastName: User
wordpressBlogName: Learn Helm!
service:
  type: NodePort
```

Now install the Chart WordPress :

```
trainee@gateway:~$ helm install wordpress bitnami/wordpress --values=wordpress-values.yaml --namespace doe309 --
version 24.1.5
NAME: wordpress
LAST DEPLOYED: Fri Dec 20 15:14:17 2024
NAMESPACE: doe309
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.doe309.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=$(kubectl get --namespace doe309 -o jsonpath="{.spec.ports[0].nodePort}" services wordpress)
   export NODE_IP=$(kubectl get nodes --namespace doe309 -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 doe309 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/
```

Check application deployment status:

```
trainee@gateway:~$ helm list --namespace doe309
NAME            NAMESPACE       REVISION        UPDATED                                     STATUS          CHART
APP VERSION
wordpress       doe309          1               2024-12-20 15:14:17.143771785 +0100 CET deployed
wordpress-24.1.5        6.7.1
```

1.8 - The helm get Command

Helm uses a system of **Hooks** that allows developers to execute tasks at specific times. These Hooks are :

- pre-install

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

To view the Hooks, if any exist, use the **helm get hooks** command:

```
trainee@gateway:~$ helm get hooks wordpress --namespace doe309
trainee@gateway:~$
```

To view the manifests, use the **helm get manifest** command:

```
trainee@gateway:~$ helm get manifest wordpress --namespace doe309 | more
---
# Source: wordpress/charts/mariadb/templates/networkpolicy.yaml
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
  name: wordpress-mariadb
  namespace: "doe309"
  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: "doe309"
  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--
```

It is possible to review the **NOTES** printed on the screen when executing the helm install command by using the **help get notes** command:

```
trainee@gateway:~$ helm get notes wordpress --namespace doe309
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.doe309.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=$(kubectl get --namespace doe309 -o jsonpath="{.spec.ports[0].nodePort}" services wordpress)
    export NODE_IP=$(kubectl get nodes --namespace doe309 -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 doe309 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/
```

To view user-specified values in the value file, use the **helm get values** command:

```
trainee@gateway:~$ helm get values wordpress --namespace doe309
USER-SUPPLIED VALUES:
service:
  type: NodePort
wordpressBlogName: Learn Helm!
wordpressEmail: helm-user@ittraining.loc
wordpressFirstName: Helm
wordpressLastName: User
wordpressPassword: my-pass
wordpressUsername: helm-user
```

To see all the values, use the **helm get values** command with the **–all** option:

```
trainee@gateway:~$ helm get values wordpress --all --namespace doe309 | 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--
```

Lastly, to view all the information, use the **helm get all** command:

```
trainee@gateway:~$ helm get all wordpress --namespace doe309 | more
NAME: wordpress
LAST DEPLOYED: Fri Dec 20 15:14:17 2024
NAMESPACE: doe309
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@ittraining.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 - Using NOTES

Check the number of Pods that have been started:

```
trainee@gateway:~$ kubectl get pods -n doe309
NAME                         READY   STATUS    RESTARTS   AGE
wordpress-75b57d8d9b-9t4jw   0/1     Running   0          4m22s
wordpress-mariadb-0          0/1     Running   0          4m22s
```

To connect to WordPress, follow the instructions in the **NOTES** section printed on the screen when running the **helm install** command:

```
trainee@gateway:~$ export NODE_PORT=$(kubectl get --namespace doe309 -o jsonpath="{.spec.ports[0].nodePort}"
services wordpress)

trainee@gateway:~$ export NODE_IP=$(kubectl get nodes --namespace doe309 -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
```

If you forget the password for the WordPress installation administration interface, use the following command:

```
trainee@gateway:~$ echo Password: $(kubectl get secret --namespace doe309 wordpress -o
jsonpath="{.data.wordpress-password}" | base64 -d)
Password: my-pass
```

1.10 - The helm upgrade Command

Updating a Chart is done by modifying the values in the YAML file. In our case, the file is called **wordpress-values.yaml**. Modify this file to increase the number of **Replicas** to 2 and to decrease the processor's memory and **millicores** from their default values of 512Mi and 300m respectively:

> 🚧 To do: Copy the content from **here** and paste it into your file.

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

```
    type: NodePort
replicaCount: 2
resources:
  requests:
    memory: 256Mi
    cpu: 100m
```

Now use the **helm upgrade** command to update the :

```
trainee@gateway:~$ helm upgrade wordpress bitnami/wordpress --values wordpress-values.yaml -n doe309 --version
24.1.5
Release "wordpress" has been upgraded. Happy Helming!
NAME: wordpress
LAST DEPLOYED: Fri Dec 20 15:33:45 2024
NAMESPACE: doe309
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.doe309.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=$(kubectl get --namespace doe309 -o jsonpath="{.spec.ports[0].nodePort}" services wordpress)
   export NODE_IP=$(kubectl get nodes --namespace doe309 -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 doe309 wordpress -o jsonpath="{.data.wordpress-password}" |
base64 -d)
```

Now check the number of Pods that have been started:

```
trainee@gateway:~$ kubectl get pods -n doe309
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
```

View the values used during the update:

```
trainee@gateway:~$ helm get values wordpress -n doe309
USER-SUPPLIED VALUES:
replicaCount: 2
resources:
  requests:
    cpu: 100m
    memory: 256Mi
service:
```

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

Run the update again, this time **without** specifying the values file **wordpress-values.yaml** on the command line:

```
trainee@gateway:~$ helm upgrade wordpress bitnami/wordpress -n doe309 --version 24.1.5
Release "wordpress" has been upgraded. Happy Helming!
NAME: wordpress
LAST DEPLOYED: Fri Dec 20 15:44:46 2024
NAMESPACE: doe309
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.doe309.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=$(kubectl get --namespace doe309 -o jsonpath="{.spec.ports[0].nodePort}" services wordpress)
   export NODE_IP=$(kubectl get nodes --namespace doe309 -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 doe309 wordpress -o jsonpath="{.data.wordpress-password}" |
base64 -d)
```

View the values used during the update:

```
trainee@gateway:~$ helm get values wordpress -n doe309
USER-SUPPLIED VALUES:
replicaCount: 2
resources:
  requests:
    cpu: 100m
    memory: 256Mi
service:
  type: NodePort
wordpressBlogName: Learn Helm!
wordpressEmail: helm-user@ittraining.loc
wordpressFirstName: Helm
wordpressLastName: User
wordpressPassword: my-pass
wordpressUsername: helm-user
```

⚠️ **Important**: Note that the helm command has reused the values from the first update.

Now check the number of Pods that have been started:

```
trainee@gateway:~$ kubectl get pods -n doe309
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 - The helm history Command

Helm uses a system of **revisions**. A new revision is created for each installation, update or **rollback**. To see the list of revisions, use the following command:

```
trainee@gateway:~$ helm history wordpress -n doe309
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
```

The **helm get values** command can be used to consult the list of values for each revision:

```
trainee@gateway:~$ helm get values wordpress --revision 3 -n doe309
USER-SUPPLIED VALUES:
```

```
replicaCount: 2
resources:
  requests:
    cpu: 100m
    memory: 256Mi
service:
  type: NodePort
wordpressBlogName: Learn Helm!
wordpressEmail: helm-user@ittraining.loc
wordpressFirstName: Helm
wordpressLastName: User
wordpressPassword: my-pass
wordpressUsername: helm-user

trainee@gateway:~$ helm get values wordpress --revision 4 -n doe309
USER-SUPPLIED VALUES:
replicaCount: 1
```

The **helm get manifest** command can be used to view the status of Kubernetes resources for each revision:

```
trainee@gateway:~$ helm get manifest wordpress --revision 4 -n doe309 | more
---
# Source: wordpress/charts/mariadb/templates/networkpolicy.yaml
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
  name: wordpress-mariadb
  namespace: "doe309"
  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: "doe309"
  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 - The helm rollback Command

To return to a previous revision, use the **helm rollback** command:

```
trainee@gateway:~$ helm rollback wordpress 3 -n doe309
Rollback was a success! Happy Helming!
```

Next, check that the current values are those of the targeted revision:

```
trainee@gateway:~$ helm get values wordpress -n doe309
USER-SUPPLIED VALUES:
replicaCount: 2
resources:
  requests:
    cpu: 100m
    memory: 256Mi
service:
  type: NodePort
wordpressBlogName: Learn Helm!
wordpressEmail: helm-user@ittraining.loc
```

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

The rollback is clearly indicated in the output of the **helm history** command:

```
trainee@gateway:~$ helm history wordpress -n doe309
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 - The helm uninstall Command

Uninstall a Chart using the **helm uninstall** command. By default, this command also deletes the revision history unless the **–keep-history** option is specified on the command line. So delete the wordpress Chart as well as the revision history:

```
trainee@gateway:~$ helm uninstall wordpress -n doe309
release "wordpress" uninstalled
```

The **helm list** command demonstrates this deletion:

```
trainee@gateway:~$ helm list -n doe309
```

```
NAME       NAMESPACE        REVISION         UPDATED STATUS    CHART     APP VERSION
```

as well as the output of the **kubectl get deployments** command:

```
trainee@gateway:~$ kubectl get deployments -l app=wordpress -n doe309
No resources found in doe309 namespace.
```

However, the **helm unistall** command does **not** delete the **PersistentVolumeClaim** created when the Chart was installed:

```
trainee@gateway:~$ kubectl get pvc -n doe309
NAME                          STATUS    VOLUME                                       CAPACITY    ACCESS MODES
STORAGECLASS    VOLUMEATTRIBUTESCLASS    AGE
data-wordpress-mariadb-0    Bound      pvc-71bba44c-9af2-44c9-9e59-103efdf1b29b    8Gi          RWO          standard
<unset>                     56m
```

To delete the **PersistentVolumeClaim**, use the **kubectl delete pvc** command with the **–field-selector** option:

```
trainee@gateway:~$ kubectl delete pvc --field-selector metadata.name=data-wordpress-mariadb-0 -n doe309
persistentvolumeclaim "data-wordpress-mariadb-0" deleted
```

This deletes the **PersistentVolumeClaim**:

```
trainee@gateway:~$ kubectl get pvc -n doe309
No resources found in doe309 namespace
```

LAB #2 - Monitoring Kubernetes with the EFK Stack

2.1 - Overview

The EFK stack consists of three elements:

- **Elasticsearch** which provides:
  - the storage of metrics and logs,
- **Fluentd** which provides:
  - the collection of raw data from Kubernetes,
  - sends metrics and logs to Elasticsearch,
- **Kibana** which enables:
  - the visualization of the data stored in Elasticsearch.

Before continuing, stop minikube and increase its virtual machine memory to 12,294 MB and the number of vCPUs to 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 - Installing the elasticsearch Chart

Start by installing the stable **repository** of **https://charts.helm.sh** :

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

Then install the Chart **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
```

Wait a few minutes until all Pods display a **READY** status:

```
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
```

Next, check for the presence of the **elasticsearch-client** and **elasticsearch-discovery** services:

```
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 - Installing fluentd-elasticsearch Chart

Install the Chart **fluentd-elasticsearch** from the **stable** repository:

```
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 needs to know the location of the Elasticsearch host. In our case, the value is **elasticsearch-client**. Note therefore the use on the command line of the **–set elasticsearch.host=** option.

Wait a few seconds until the **fluentd-fluentd-elasticsearch-xxxxx** Pod displays a status of **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**: Note that when using minikube, we only have one node in the Kubernetes cluster. For this reason, we have installed only one Fluentd Pod with Helm, without using a DaemonSet.

2.4 - Installing the kibana Chart

Kibana also needs to know the location of the Elasticsearch host. This time, the value will be included in a **values** file. Create the file **kibana-**

values.yaml:

To do: Copy the content from **here** **and paste it into your file.**

```
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**: Note the creation of a **NodePort** service. When using a DaemonSet, we would have created a **ClusterIP*** service of the type **LoadBalancer**.

Install the Chart kibana from the **stable** repository, referencing the **values kibana-values.yaml** file:

```
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
```

Note the instructions at the end of the previous output which indicate how to reach Kibana from outside the cluster. Create the two variables **NODE_PORT** and **NODE_IP** and run the **echo** command to obtain the Kibana connection socket. Note this 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
```

Wait a few minutes until the Pod **kibana-yyyyyyyy-xxxxx** displays a status of **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
```

Next, check for the presence of the **NodePort** service:

```
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 - Generating Logs in Kubernetes

Now create a **counter.yaml** file to create a Pod that will generate continuous demo logs:

> To do: Copy the content from **here** and paste it into your file.

```
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']
```

Apply this file with the **kubectl** command:

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

Check that the **counter** Pod is in **READY** state:

```
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
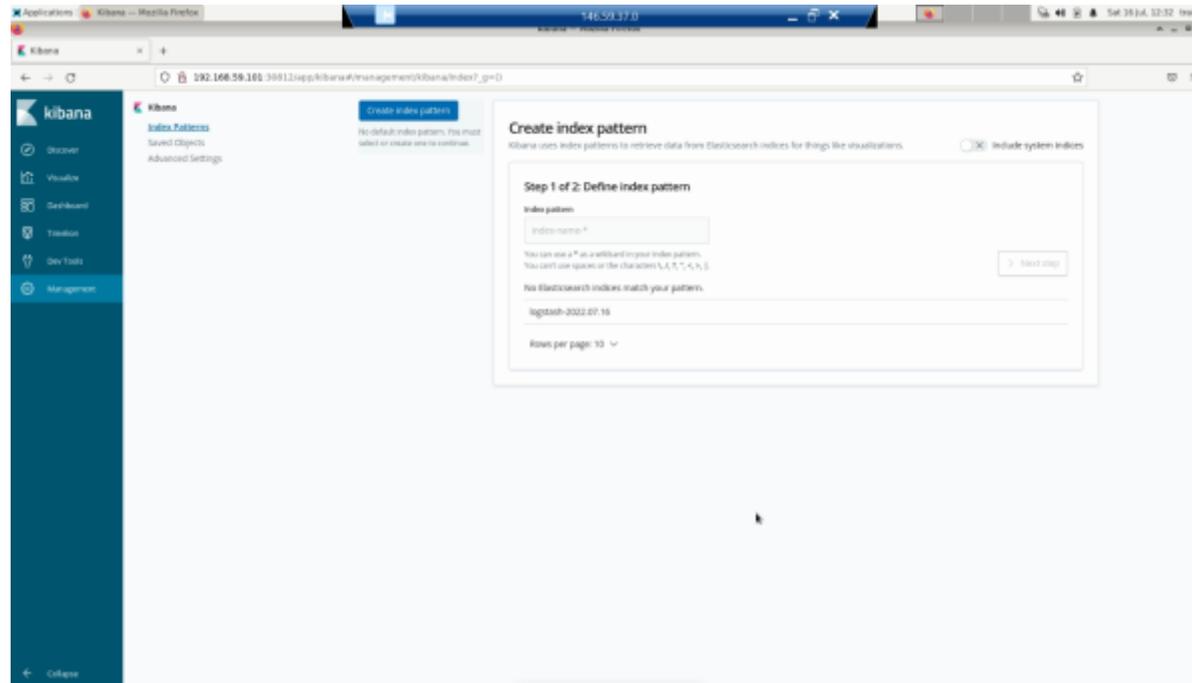```

## 2.6 - Visualizing Data with Kibana

Connect to your VM Gateway in VNC mode using Apache Guacamole or via an RDP connection.

Open the FireFox browser and enter the address you obtained by executing the **echo** command above.
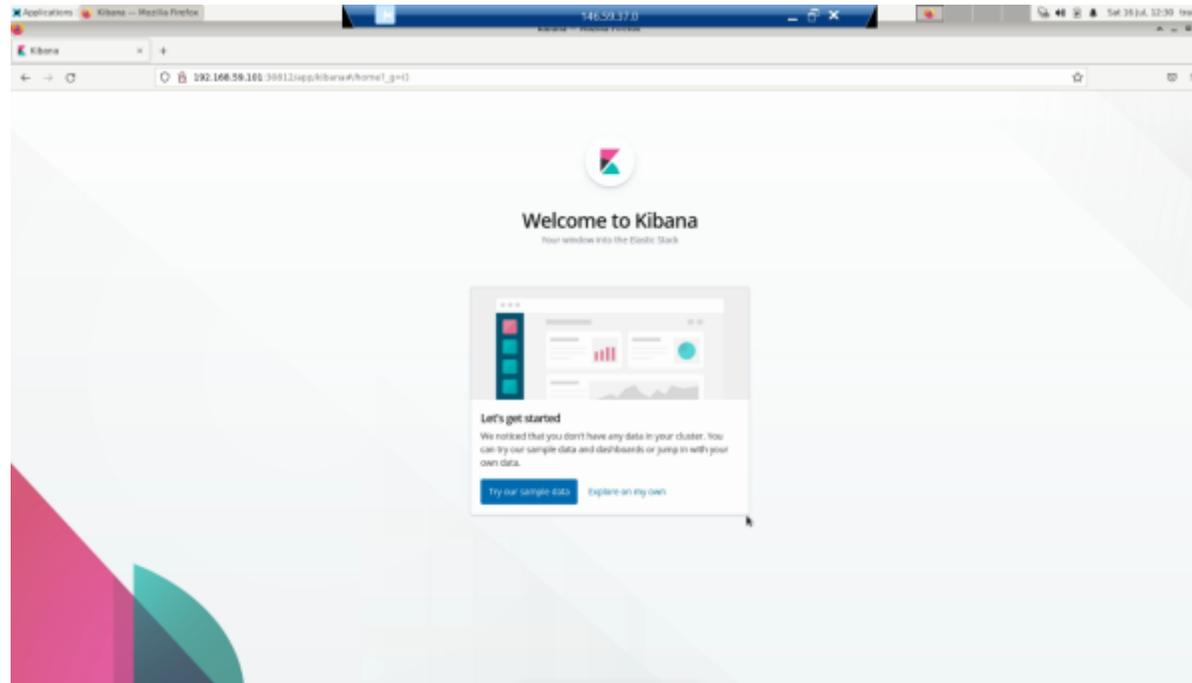
Click on **Explore on my own** > **Management** > **Index Patterns**, then, in **Create index pattern**, enter the value **logstash-**\* and click on the **Next step** button:

In step 2/2, enter the value of **@timestamp** then click on the **Create index pattern** button:
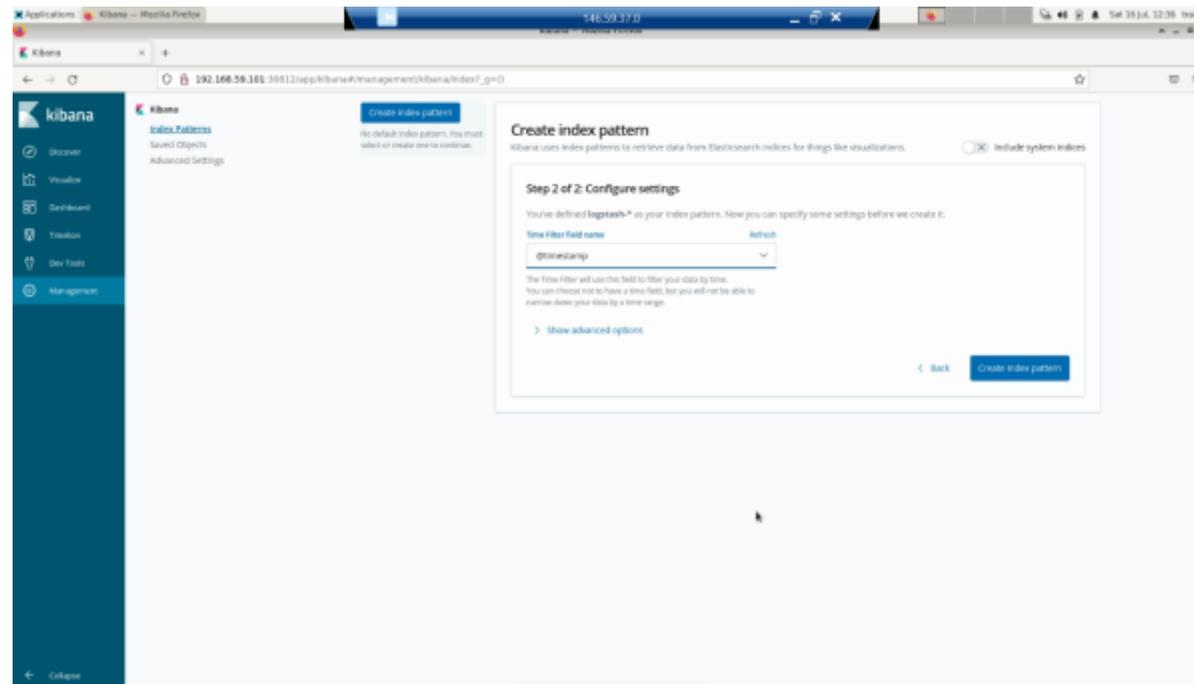
Click on the **Discover** link in the Kibana menu. Select **logstash-**\* then choose **@timestamp** in **Available fields** and click on the **Add** button:
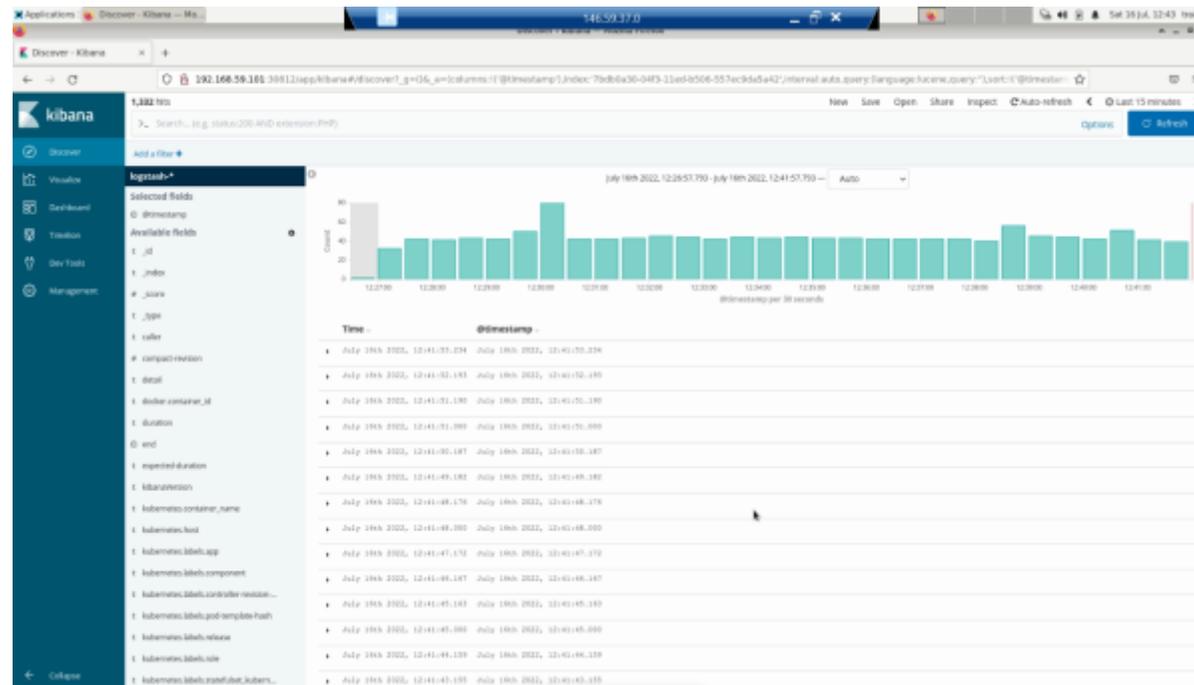
Apply a filter by clicking on the **kubernetes.pod_name** field:

Click on the **+** magnifying glass opposite the **counter** line, then on the **Add** button opposite the field name: