

**Lab Manual- Install Wordpress using Helm Chart on AKS**

**Prepared for**:

**Date:** 18th Dec 2023

**Prepared by:**

Document Name: Lab Manual **Document Number** AZLabn916

**Contributor:**

Contents

[1. Objective 3](#_Toc155432767)

[2. Use the default bridge network 3](#_Toc155432768)

[3. Use the User-defined bridge network 6](#_Toc155432769)

# Objective

A Helm chart is a package that contains all the necessary resources to deploy an application to a Kubernetes cluster. This includes YAML configuration files for deployments, services, secrets, and config maps that define the desired state of your application

# Add Repository

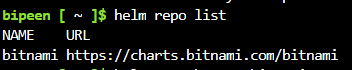
1. Type below command to add bitnami repo

helm repo add bitnami https://charts.bitnami.com/bitnami



1. List the added repo

helm repo list



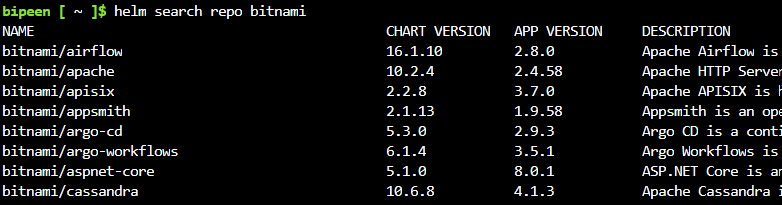
# helm search': Finding Charts

Helm comes with a powerful search command. It can be used to search two different types of source:

* helm search hub searches [the Artifact Hub](https://artifacthub.io/), which lists helm charts from dozens of different repositories.
* helm search repo searches the repositories that you have added to your local helm client (with helm repo add). **This search is done over local data, and no public network connection is needed.**

1. Search the added repo

helm search repo bitnami



1. Search the Hub for WordPress package

helm search hub wordpress

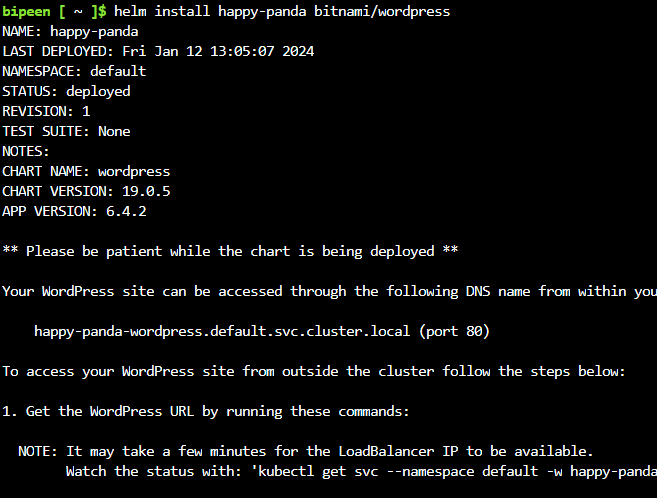


# Install the WordPress

To install a new package, use the helm install command. At its simplest, it takes two arguments: A release name that you pick, and the name of the chart you want to install.

1. Install Wordpress

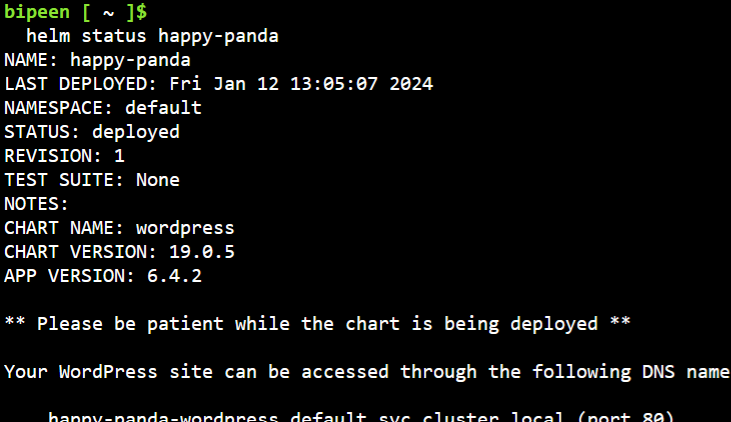
helm install happy-panda bitnami/wordpress



Now the wordpress chart is installed. Note that installing a chart creates a new release object. The release above is named happy-panda. (If you want Helm to generate a name for you, leave off the release name and use --generate-name.)

## To keep track of a release's state, or to re-read configuration information, you can use helm status

helm status happy-panda



# Connect the WordPress

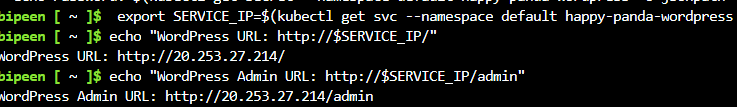
To access your WordPress site from outside the cluster, follow the steps below:

## Get the WordPress URL by running these commands:

export SERVICE\_IP=$(kubectl get svc --namespace default happy-panda-wordpress --template "{{ range (index .status.loadBalancer.ingress 0) }}{{ . }}{{ end }}")

echo "WordPress URL: <http://$SERVICE_IP/>"

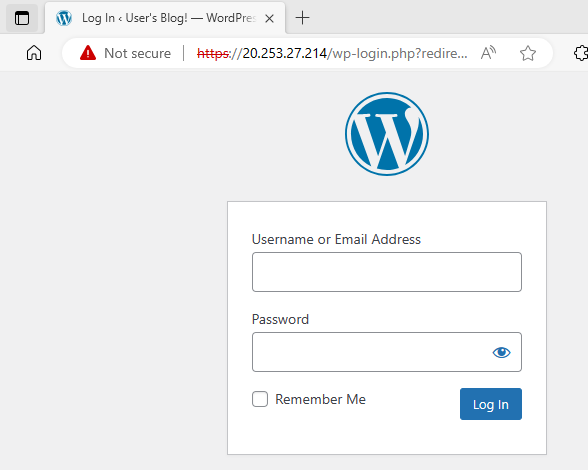
echo "WordPress Admin URL: http://$SERVICE\_IP/admin"



NOTE: It may take a few minutes for the LoadBalancer IP to be available.

Watch the status with: 'kubectl get svc --namespace default -w happy-panda-wordpress'

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



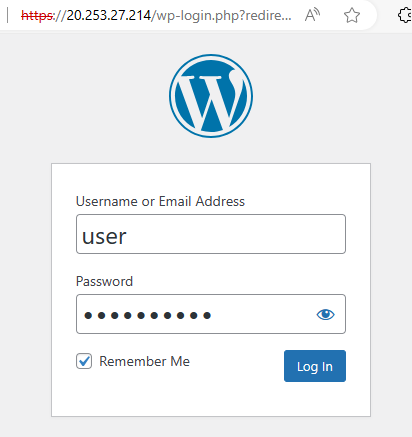
## Login with the following credentials below to see your blog:

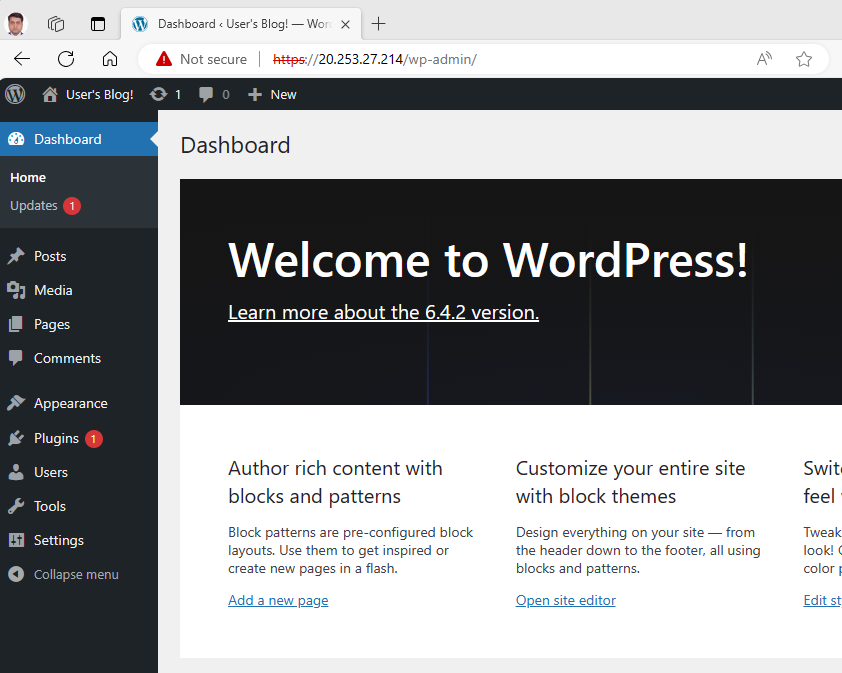
echo Username: user



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







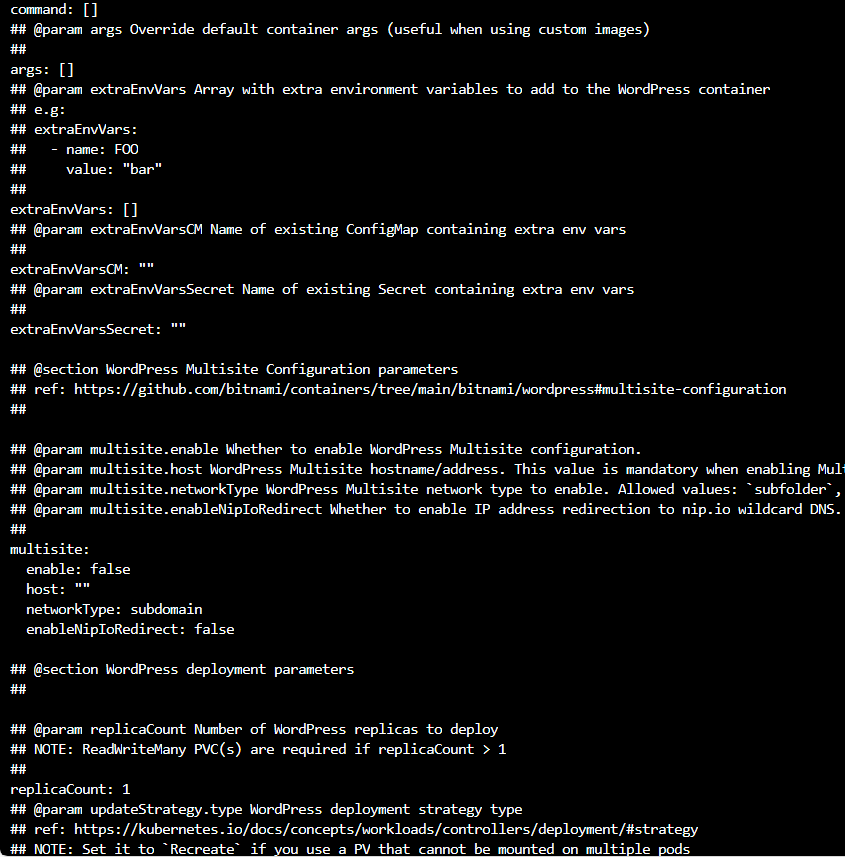
# Customizing the Chart Before Installing

Installing the way we have here will only use the default configuration options for this chart. Many times, you will want to customize the chart to use your preferred configuration.

To see what options are configurable on a chart, use helm show values:

helm show values bitnami/wordpress

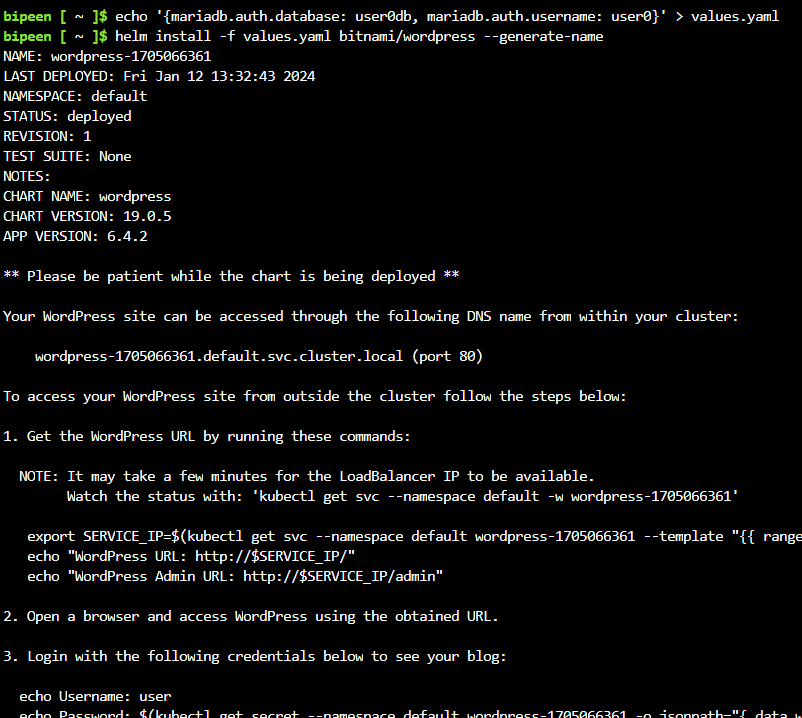




You can then override any of these settings in a YAML formatted file, and then pass that file during installation.

echo '{mariadb.auth.database: user0db, mariadb.auth.username: user0}' > values.yaml

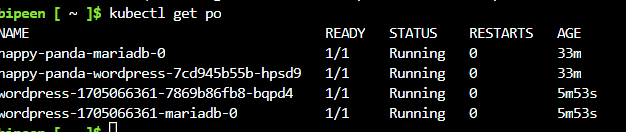
helm install -f values.yaml bitnami/wordpress --generate-name



The above will create a default MariaDB user with the name user0, and grant this user access to a newly created user0db database, but will accept all the rest of the defaults for that chart.

There are two ways to pass configuration data during install:

* --values (or -f): Specify a YAML file with overrides. This can be specified multiple times and the rightmost file will take precedence
* --set: Specify overrides on the command line.



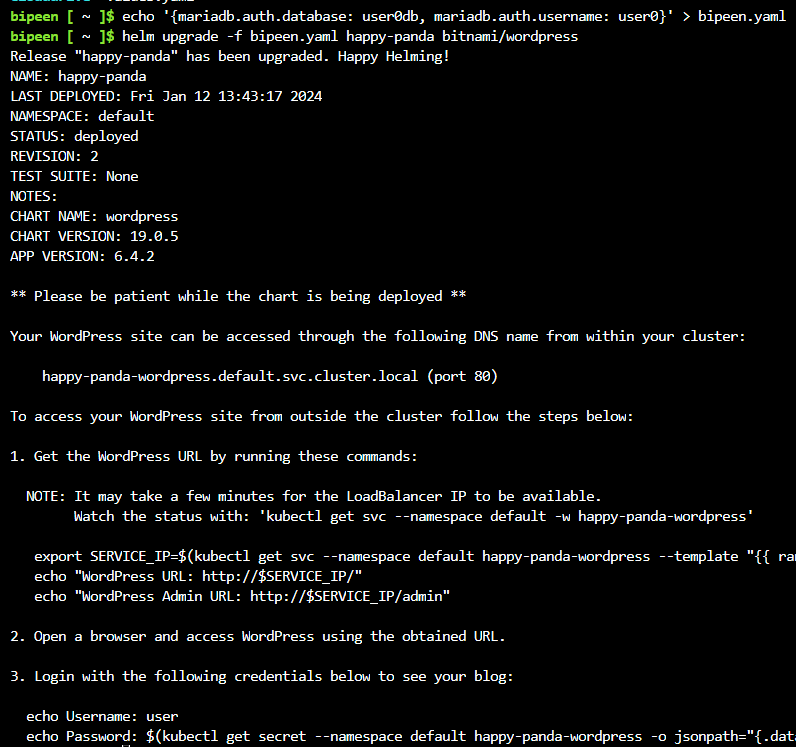
# 'helm upgrade' and 'helm rollback': Upgrading a Release, and Recovering on Failure

When a new version of a chart is released, or when you want to change the configuration of your release, you can use the helm upgrade command.

An upgrade takes an existing release and upgrades it according to the information you provide. Because Kubernetes charts can be large and complex, Helm tries to perform the least invasive upgrade. It will only update things that have changed since the last release.

echo '{mariadb.auth.database: user0db, mariadb.auth.username: user0}' > bipeen.yaml

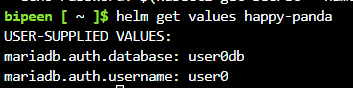
helm upgrade -f bipeen.yaml happy-panda bitnami/wordpress



In the above case, the happy-panda release is upgraded with the same chart, but with a new YAML file:

* We can use helm get values to see whether that new setting took effect.

helm get values happy-panda



* Now, if something does not go as planned during a release, it is easy to roll back to a previous release using helm rollback [RELEASE] [REVISION].

helm rollback happy-panda 1

