### **Prerequisites**

1. **Kubernetes Cluster:** Ensure you have a running Kubernetes cluster.EKS
2. **kubectl:** Ensure you have kubectl installed and configured to interact with your Kubernetes cluster.
3. **Helm:** Install Helm to manage Kubernetes applications.
4. AWS cloud

I have experimented with the below in the AWS EKS cluster. Please install the Nginx ingress controller as required (or) make the service type “LoadBalancer” in the Values. yaml file.  
I have used cluster issuer as **Letencrypt** for the SSL cert TLS. If you want you can use **AWSPCAissuer** also as required.   
  
Modify anything as required after extraction and do helm package and push to your Centralized registry like Nexus/Jfrog, or Gitlab registry.   
  
Few helm commands for reference:   
  
ramkumar.v@ip-172-16-25-1 rvs\_pocs % **helm package wordpress**

**/Users/ramkumar.v/rvs\_pocs/my-wordpress-0.1.0.tgz**

ramkumar.v@ip-172-16-25-1 rvs\_pocs %   
  
  
Render the Helm chart to ensure that the templates are correctly expanded:

**helm template my-wordpress wordpress --namespace my-wordpress**

**Step 1:**   
  
I have packaged the required deployment specs into below tar,gz file as a helm package.  
File name: **my-wordpress-0.1.0.tgz**Extract the package using the below command :   
  
# tar -xvf **my-wordpress-0.1.0.tgz**  
  
 **This extracted folder contains Chart.yaml with version,Values.yaml for the required Key value pair updates for templates  
Templates Folder content:**Persistent-volume.yaml:

Persistent-volume-claim.yaml:

MySQL Deployment

Mysql-service

WordPress Deployment

Wordpress-service.yaml:

Ingress

HPA   
storgaeclass.yaml

Letsencrypt.yaml

**Installation command : Dry run with Debug enabled**helm upgrade --install my-wordpress wordpress --namespace my-wordpress --create-namespace --debug –dry-run   
 **Installation after Dry run success.**   
helm upgrade --install my-wordpress wordpress --namespace my-wordpress --create-namespace --debug –dry-run

**my-wordpress – Release name  
wordpress – Chart name**

**namespace – namespace name**

**--create-namespace – if not already exists   
  
#### Values.yaml #####**replicaCount: 3

service:

port: 80

image:

repository: wordpress

tag: 4.8-apache

pullPolicy: IfNotPresent

mysql:

image: mysql:5.6

rootPassword: rvs@2024@123

database: wordpress

persistence:

enabled: true

**storageClass: ebs-sc**

accessMode: ReadWriteOnce

size: 10Gi

ingress:

enabled: true

annotations:

cert-manager.io/cluster-issuer: cert-manager-letsencrypt-prod-r53

kubernetes.io/ingress.class: nginx

meta.helm.sh/release-name: my-wordpress

meta.helm.sh/release-namespace: my-wordpress

hosts:

- host: wordpress.eks.us-east-1.product-dev.ram.link

http:

paths:

- path: /

pathType: ImplementationSpecific

backend:

service:

name: wordpress

port: 80

tls:

- hosts:

- wordpress.eks.us-east-1.product-dev.ram.link

secretName: wordpress-tls

autoscaling:

enabled: true

minReplicas: 3

maxReplicas: 10

targetCPUUtilizationPercentage: 50

**#####  
  
## If we don't need to have ingress and want to use service as a “LoadBalancer type”, Then please use service enabled and Ingress enabled false. If you want to increase the persistence storage of MySQL DB, we can update the required values in Values.yaml and I have a separate storage class defined in the helm package. It would automatically create the storage class “ebs-sc” in AWS and use that as for PV.**The Output of the command would be like below. **ramkumar.v@ip-172-16-25-1 rvs\_pocs % helm upgrade --install my-wordpress wordpress --namespace my-wordpress --create-namespace --debug**

history.go:56: [debug] getting history for release my-wordpress

upgrade.go:153: [debug] preparing upgrade for my-wordpress

upgrade.go:161: [debug] performing update for my-wordpress

upgrade.go:354: [debug] creating upgraded release for my-wordpress

client.go:393: [debug] checking 10 resources for changes

client.go:684: [debug] Looks like there are no changes for Secret "mysql-secret"

client.go:684: [debug] Looks like there are no changes for StorageClass "ebs-sc"

client.go:684: [debug] Looks like there are no changes for PersistentVolume "wp-pv"

client.go:684: [debug] Looks like there are no changes for PersistentVolumeClaim "wp-pvc"

client.go:684: [debug] Looks like there are no changes for Service "mysql"

client.go:684: [debug] Looks like there are no changes for Service "wordpress"

client.go:684: [debug] Looks like there are no changes for Deployment "mysql"

client.go:684: [debug] Looks like there are no changes for Deployment **"wordpress"**

**client.go:684: [debug] Looks like there are no changes for HorizontalPodAutoscaler "wordpress"**

**W0614 00:45:37.337527 16483 warnings.go:70] annotation "kubernetes.io/ingress.class" is deprecated, please use 'spec.ingressClassName' instead**

**client.go:414: [debug] Created a new Ingress called "my-wordpress" in my-wordpress**

**upgrade.go:169: [debug] updating status for upgraded release for my-wordpress**

**Release "my-wordpress" has been upgraded. Happy Helming!**

**NAME: my-wordpress**

**LAST DEPLOYED: Fri Jun 14 00:45:15 2024**

**NAMESPACE: my-wordpress**

**STATUS: deployed**

**REVISION: 5**

**TEST SUITE: None**

**USER-SUPPLIED VALUES:**

**{}**

**COMPUTED VALUES:**

**autoscaling:**

**enabled: true**

**maxReplicas: 10**

**minReplicas: 3**

**targetCPUUtilizationPercentage: 50**

**image:**

**pullPolicy: IfNotPresent**

**repository: wordpress**

**tag: 4.8-apache**

**ingress:**

**annotations:**

**cert-manager.io/cluster-issuer: cert-manager-letsencrypt-prod-r53**

**kubernetes.io/ingress.class: nginx**

**meta.helm.sh/release-name: my-wordpress**

**meta.helm.sh/release-namespace: my-wordpress**

**enabled: true**

**hosts:**

**- host: wordpress.eks.us-east-1.product-dev.socure.link**

**http:**

**paths:**

**- backend:**

**service:**

**name: wordpress**

**port: 80**

**path: /**

**pathType: ImplementationSpecific**

**tls:**

**- hosts:**

**- wordpress.eks.us-east-1.product-dev.socure.link**

**secretName: wordpress-tls**

**mysql:**

**database: wordpress**

**image: mysql:5.6**

**rootPassword: rvs@2024@123**

**persistence:**

**accessMode: ReadWriteOnce**

**enabled: true**

**size: 10Gi**

**storageClass: ebs-sc**

**replicaCount: 3**

**service:**

**port: 80**

**HOOKS:**

**MANIFEST:**

**---**

**# Source: my-wordpress/templates/secrets.yaml**

**apiVersion: v1**

**kind: Secret**

**metadata:**

**name: mysql-secret**

**type: Opaque**

**data:**

**mysql-root-password: cnZzQDIwMjRAMTIz**

**---**

**# Source: my-wordpress/templates/storageclass.yaml**

**apiVersion: storage.k8s.io/v1**

**kind: StorageClass**

**metadata:**

**name: ebs-sc**

**provisioner: kubernetes.io/aws-ebs**

**parameters:**

**type: gp2**

**fsType: ext4**

**encrypted: "true"**

**---**

**# Source: my-wordpress/templates/persistent-volume.yaml**

**apiVersion: v1**

**kind: PersistentVolume**

**metadata:**

**name: wp-pv**

**spec:**

**storageClassName: ebs-sc**

**capacity:**

**storage: 10Gi**

**accessModes:**

**- ReadWriteOnce**

**hostPath:**

**path: "/mnt/data"**

**---**

**# Source: my-wordpress/templates/persistent-volume-claim.yaml**

**apiVersion: v1**

**kind: PersistentVolumeClaim**

**metadata:**

**name: wp-pvc**

**spec:**

**storageClassName: ebs-sc**

**accessModes:**

**- ReadWriteOnce**

**resources:**

**requests:**

**storage: 10Gi**

**---**

**# Source: my-wordpress/templates/mysql-service.yaml**

**apiVersion: v1**

**kind: Service**

**metadata:**

**name: mysql**

**spec:**

**ports:**

**- port: 3306**

**selector:**

**app: mysql**

**---**

**# Source: my-wordpress/templates/wordpress-service.yaml**

**apiVersion: v1**

**kind: Service**

**metadata:**

**name: wordpress**

**spec:**

**type: ClusterIP**

**ports:**

**- port: 80**

**selector:**

**app: wordpress**

**---**

**# Source: my-wordpress/templates/mysql-deployment.yaml**

**apiVersion: apps/v1**

**kind: Deployment**

**metadata:**

**name: mysql**

**spec:**

**selector:**

**matchLabels:**

**app: mysql**

**strategy:**

**type: Recreate**

**template:**

**metadata:**

**labels:**

**app: mysql**

**spec:**

**containers:**

**- image: mysql:5.6**

**name: mysql**

**env:**

**- name: MYSQL\_ROOT\_PASSWORD**

**valueFrom:**

**secretKeyRef:**

**name: mysql-secret**

**key: mysql-root-password**

**- name: MYSQL\_DATABASE**

**value: wordpress**

**ports:**

**- containerPort: 3306**

**name: mysql**

**volumeMounts:**

**- name: mysql-persistent-storage**

**mountPath: /var/lib/mysql**

**volumes:**

**- name: mysql-persistent-storage**

**persistentVolumeClaim:**

**claimName: wp-pvc**

**---**

**# Source: my-wordpress/templates/wordpress-deployment.yaml**

**apiVersion: apps/v1**

**kind: Deployment**

**metadata:**

**name: wordpress**

**spec:**

**replicas: 3**

**selector:**

**matchLabels:**

**app: wordpress**

**template:**

**metadata:**

**labels:**

**app: wordpress**

**spec:**

**containers:**

**- image: wordpress:4.8-apache**

**name: wordpress**

**env:**

**- name: WORDPRESS\_DB\_HOST**

**value: mysql:3306**

**- name: WORDPRESS\_DB\_PASSWORD**

**valueFrom:**

**secretKeyRef:**

**name: mysql-secret**

**key: mysql-root-password**

**ports:**

**- containerPort: 80**

**volumeMounts:**

**- name: wordpress-persistent-storage**

**mountPath: /var/www/html**

**volumes:**

**- name: wordpress-persistent-storage**

**persistentVolumeClaim:**

**claimName: wp-pvc**

**---**

**# Source: my-wordpress/templates/hpa.yaml**

**apiVersion: autoscaling/v2**

**kind: HorizontalPodAutoscaler**

**metadata:**

**name: wordpress**

**spec:**

**scaleTargetRef:**

**apiVersion: apps/v1**

**kind: Deployment**

**name: wordpress**

**minReplicas: 3**

**maxReplicas: 10**

**metrics:**

**- type: Resource**

**resource:**

**name: cpu**

**target:**

**type: Utilization**

**averageUtilization: 50**

**---**

**# Source: my-wordpress/templates/ingress.yaml**

**apiVersion: networking.k8s.io/v1**

**kind: Ingress**

**metadata:**

**name: my-wordpress**

**annotations:**

**cert-manager.io/cluster-issuer: "cert-manager-letsencrypt-prod-r53"**

**kubernetes.io/ingress.class: "nginx"**

**meta.helm.sh/release-name: "my-wordpress"**

**meta.helm.sh/release-namespace: "my-wordpress"**

**spec:**

**rules:**

**- host: wordpress.eks.us-east-1.product-dev.socure.link**

**http:**

**paths:**

**- path: /**

**pathType: ImplementationSpecific**

**backend:**

**service:**

**name: wordpress**

**port:**

**number: 80**

**tls:**

**- secretName: wordpress-tls**

**hosts:**

**- wordpress.eks.us-east-1.product-dev.socure.link**