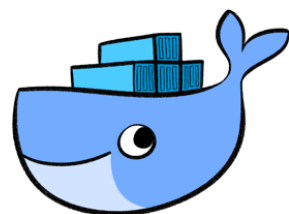




kubernetes

Wordpress in Kubernetes (App Container runtime) July 30, 2020

by @loanshark (Changhyun Kim)



kubernetes

Open-source system for automation deployment

word-press

Open-source website creation platform that is written in
PHP and a MySQL database

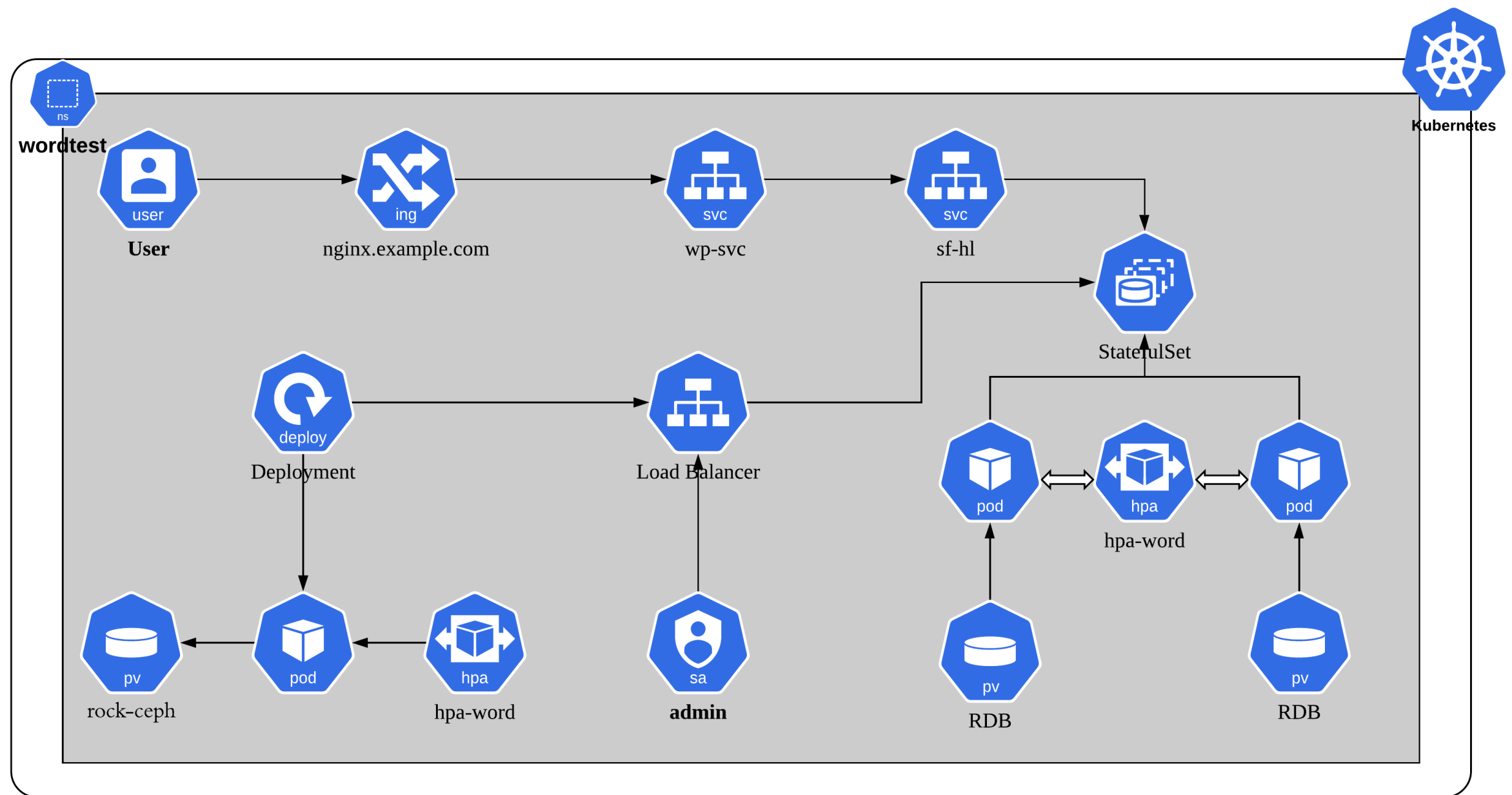
vscode

streamlined code editor

Requirement

1. Ingress TLS Termination
2. Service: ClusterIP
3. Deployment: Wordpress(Replica:2, Liveness, Readiness)
4. PVC: StorageClass(cephfs)
5. HPA: Deployment
6. Service: Headless
7. Statefulset: Mysql (Replica : 2, Liveness, Readiness)
8. PVC: StorageClass(rbd)
9. HPA: Statefulset
10. PodAffinity 및 PodAntiAffinity (wp/db <-> wp/db)
11. ConfigMap, Secret

Architecture



DEMO

Setting

namespace wordtest

dir-name wordtest

```
$ kubectl create ns wordtest
```

```
$ kubcctl config set-context -current -namespace=wordtest # 선호하는 네임스페이스 설정
```

```
$ kubectl config view -minify | grep namespace: # 확인하기
```

```
$ mkdir wordtest
```

```
$ kubectl get all -namespace rock-ceph # ceph 스토리지 확인
```

```
$ kubectl get CephFileSystem -A
```

Deployment

```
$ kubectl create -f deploy.yaml  
$ kubectl get deployments.apps
```

작성한 yml 실행
deployment 확인

```
~/Desktop/test » kubectl get deployments.apps  
NAME      READY   UP-TO-DATE   AVAILABLE   AGE  
wp-dp     3/3     3            3           124m
```



여기서 Deployment 을 생성할 때 resources.requests 를 설정해 Metric 을 받아야, HPA 설정이 가능하다.
또 한, 환경 변수 받을 값을 'env' 로 정의한다.

Service

Service 생성 및 Loadbalancer 생성

```
$ kubectl apply -f svc_dir  
$ kubectl get svc
```

디렉토리 안에 있는 yamI 실행
service 확인

```
~ » kubectl get svc
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
sf-hl	ClusterIP	None	<none>	3306/TCP	17h
wp-lb	LoadBalancer	10.233.38.224	192.168.122.200	80:31919/TCP	17h
wp-svc	ClusterIP	10.233.41.132	<none>	4000/TCP	4h7m

생성된 Service와 Endpoint를 확인

```
~ » kubectl get ep -o wide
```

NAME	ENDPOINTS	AGE
sf-hl	10.233.92.31:3306	17h
wp-lb	<none>	17h
wp-svc	<none>	4h13m

Persistent Volume

```
$ kubectl patch storageclasses.storage.k8s.io csi-cephfs \
  -p '{"metadata":
{"annotations":{"storageclass.kubernetes.io/is-default-class":"true"}}}'
# 스토리지의 default 를 'ceph' 로 변경한다.
```

```
$ kubectl apply -f pvc_dir
$ kubectl get pvc
```

디렉토리안에 있는 yml 실행
pvc 확인

```
~/Desktop/test/word » kubectl get pvc
NAME                                STATUS    VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   AGE
data-wp-statefulset-0              Bound    pvc-c39733eb-9862-4f68-bcc8-105587a868d1   10Gi       RWO             rook-ceph-block 19h
data-wp-statefulset-1              Bound    pvc-f05fc33d-e837-4f42-85eb-3d7a2787375b   10Gi       RWO             rook-ceph-block 18h
db-pvc                             Bound    pvc-b2e10989-7299-4c19-b6c8-9fb1c6b76139   20Gi       RWO             csi-cephfs      19h
wp-lb-pvc                          Bound    pvc-2287dcfc-9f67-4b9f-ac80-371640db7e81   20Gi       RWX             csi-cephfs      20h
```



data-wp-statefulset 으로 명시된 'pvc'는 Statefulset.yaml 에서 volumeClaimTemplates 으로 정의된 볼륨이다. 또한, 'pv' 는 동적 스토리지인 ceph 에서 자동으로 할당하기 때문에 만들 필요가 없다.

StatefulSet (MySQL)

StatefulSet 생성

```
$ kubectl apply -f state_dir  
$ kubectl get statefulsets.apps
```

디렉토리 안에 있는 yamI 실행
statefulset 확인

```
~/Desktop/test » kubectl get statefulsets.apps  
NAME                READY   AGE  
wp-statefulset       1/2     32m
```



StatefulSet 로 인해 생성된 Pod 는 개별적인 PVC(RDB) 를 가진다. 현재, 기본스토리지 ceph-fs 로 설정을하여 storageClassName 를 rook-ceph-block 으로 정의한다. Pod 는 순차적으로 실행되며 마스터 및 슬레이브 형태의 아키텍처로 실행된다.

ConfigMap

Master 와 Slave 를 정의할 함수를 ConfigMap 으로 생성한다.

```
$ kubectl create-f mysql-config.yaml  
$ kubectl get deployments.apps
```

*# 작성한 yamI 실행
deployment 확인*

```
~/Desktop/test » kubectl get configmaps  
NAME      DATA   AGE  
mysql     2       20h
```

Secret

Ingress TLS Termination 을 위한 Secret 및 TLS 인증서를 생성한다.

```
$ mkdir ingress-tls
```

디렉토리 생성 및 TLS 키 생성

```
$ openssl genrsa -out ingress/ingress.key 2048
```

```
$ openssl req -new -x509 -key ingress/ingress.key \
  -out ingress/ingress.crt \
  -days 3650 -subj /CN=nginx.example.com
```

```
$ kubectl create secret tls ingress-tls-secret \
  --key=ingress/ingress-tls.key \
  --cert=ingress/ingress-tls.crt
```

```
~/Desktop/test » kubectl get secrets
```

NAME	TYPE	DATA	AGE
default-token-qvrd7	kubernetes.io/service-account-token	3	21h
ingress-tls-secret	kubernetes.io/tls	2	18h

Ingress TLS

생성된 Secret 을 정의하여 Ingress yaml 을 생성

```
$ kubectl create -f ingress-tls.yaml
```

```
$ kubectl get deployments.apps
```

작성한 yaml 실행

ingress-tls 확인

```
~/Desktop/test » kubectl get ingresses.extensions
```

NAME	HOSTS	ADDRESS	PORTS	AGE
happy-ingress	nginx.example.com		80, 443	18h

curl 로 Ingress 를 테스트

```
~/Desktop/test/word/ingress-tls » curl nginx.example.com/hi:80
```

```
<!DOCTYPE html>
```

```
<html class="no-js" lang="ko-KR">
```

```
  <head>
```

```
    <meta charset="UTF-8">
```

```
    <meta name="viewport" content="width=device-width, initial-scale=1.0" >
```

```
    <link rel="profile" href="https://gmpg.org/xfn/11">
```

Horizontalpodautoscalers (HPA)

Deployment & StatefulSet 의 각 각 HPA 를 생성

```
$ kubectl apply -f hpa_dir
```

```
$ kubectl get horizontalpodautoscalers.autoscaling
```

디렉토리 있는 yaml 실행

HPA 확인

```
~/Desktop/test » kubectl get horizontalpodautoscalers.autoscaling
```

NAME	REFERENCE	TARGETS	MINPODS	MAXPODS	REPLICAS	AGE
hpa-stat	StatefulSet/wp-statefulset	<unknown>/70%	1	5	0	20h
hpa-word	Deployment/wp-dp	<unknown>/70%	1	5	0	20h

💡 Deployment & Statefulset 의 request 값을 임의로 주었기 때문에 차이가 있다. 스케일링의 필요한 계산 공식은 이렇다(목표 복제본 수 = $\text{ceil}[\text{현재 복제본 수} \times (\text{현재 메트릭 값} / \text{목표 메트릭 값})]$)

Liveness Probe & Readiness Probe

Deployment \rightarrow Liveness Probe & Readiness Probe

```
livenessProbe:
  httpGet:
    path: /
    port: 80
readinessProbe:
  httpGet:
    path: /
    port: 80
```



✓ Liveness & Readiness 둘다 HTTP Probe
로 '/' 에 request 를 보내고 '200' 코드를
받으면 실행된다

StatefulSet \rightarrow Liveness Probe & Readiness Probe

```
livenessProbe:
  exec:
    command: ["mysqladmin", "ping"]
  initialDelaySeconds: 30
  periodSeconds: 10
  timeoutSeconds: 5
readinessProbe:
  exec:
    # Check we can execute queries over TCP (skip-networking is off).
    command: ["mysql", "-h", "127.0.0.1", "-e", "SELECT 1"]
  initialDelaySeconds: 5
  periodSeconds: 2
  timeoutSeconds: 1
```



✓ Liveness & Readiness 둘다 exec Probe 로 해당
'command' 의 값이 올바르게 실행되는지 테스트한
다

* Option

InitialDelaySeconds: 해당 숫자만큼 대기

PeriodSecond: 정해진 주기로 확인

TimeoutSecond: 숫자만큼 동작하지 않으면 Probe
가 실행된다.

EXECUTE



환영합니다

유명한 5분 워드프레스 설치 과정에 오신 것을 환영합니다! 아래에서 정보를 입력만 하면 세계에서 가장 확장성 높고 강력한 개인 출판 플랫폼을 사용하는 길로 들어서게 됩니다.

필요한 정보

다음 정보들을 제공해주세요. 나중에 다시 변경할 수 있으니 걱정하지 않아도 됩니다.

사이트 제목

사용자명

사용자명은 알파벳, 숫자, 스페이스, 밑줄, 하이픈, 마침표, @ 심볼만 가능합니다.

암호

[숨기기](#)

매우 약함

중요: 로그인하려면 이 비밀번호가 필요할 것입니다. 안전한 곳에 보관하십시오.

비밀번호 확인

☒ 취약한 암호 사용 확인

이메일 주소:

계속하기 전에 이메일 주소를 한 번 더 확인하세요.

검색 엔진 접근 여부

☐ 검색 엔진이 이 사이트 검색 차단하기

이 요청이 받아들여지는 것은 전적으로 검색 엔진에 좌우됩니다.

[워드프레스 설치하기](#)



사용자명 또는 이메일 주소

test

암호

••••



☐ 기억하기

로그인

암호를 분실하셨나요?

← [changhyun\(으\)로 돌아가기](#)

워드프레스에 오신 것을 환영합니다!

시작을 위해 몇개의 링크를 마련했습니다:

무시

시작하기

사이트를 사용자 정의 하기

또는 [테마를 완전히 변경하세요](#)

다음 단계

- 첫 블로그 글을 작성하세요
- 자기 소개 페이지를 추가하세요
- 홈페이지 설정하기
- 사이트 보기

추가 작업

- 위젯 관리
- 메뉴 관리
- 댓글 켜기 또는 끄기
- 시작하기에 대해 더 배우기

사이트 건강 상태

No information yet...

Site health checks will automatically run periodically to gather information about your site. You can also [visit the Site Health screen](#) to gather information about your site now.

사이트 현황

1 글

1 페이지

1개의 댓글

워드프레스 5.4.2이(가) [Twenty Twenty](#) 테마를 사용하고 있습니다.

활동

최근 발행

오늘, 7:47 오후

안녕하세요!

최고 대그

빠른 임시글

타이틀

내용

무슨 생각을 하고 계신가요?

임시 글로 저장

워드프레스 이벤트 및 뉴스

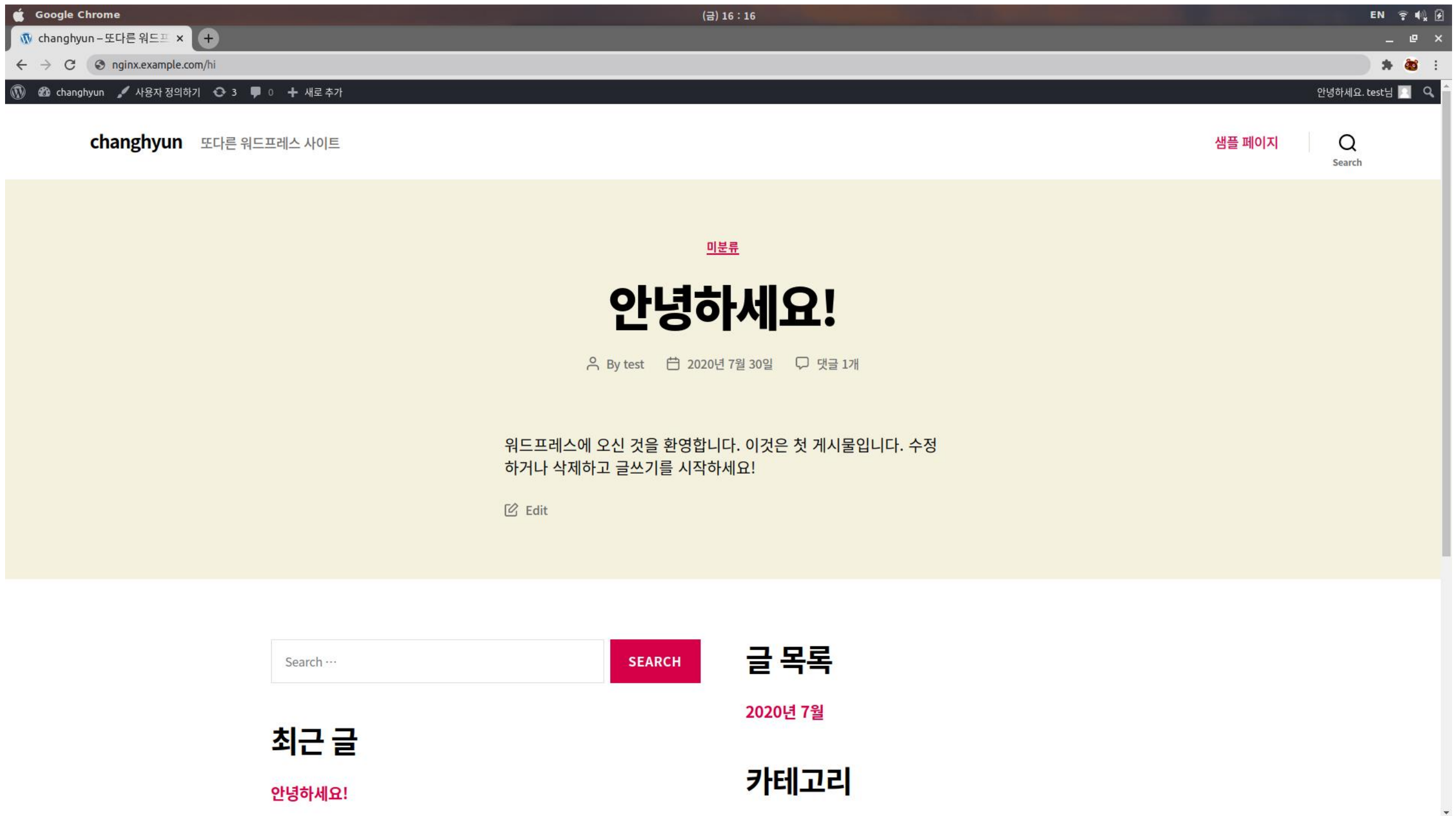
인근의 예정된 이벤트 참석하기. [📅](#)

현재 인근에 예정된 이벤트가 없습니다. 만드시겠습니까?

[WordPress 5.5 Release Candidate](#)

[WPTavern: Ring Launches IIRI Submissions Plugin for WordPress](#)

Final check



THANK YOU

by @loanshark (Changhyun Kim)
yaml 🐻 <https://github.com/changhyuni>