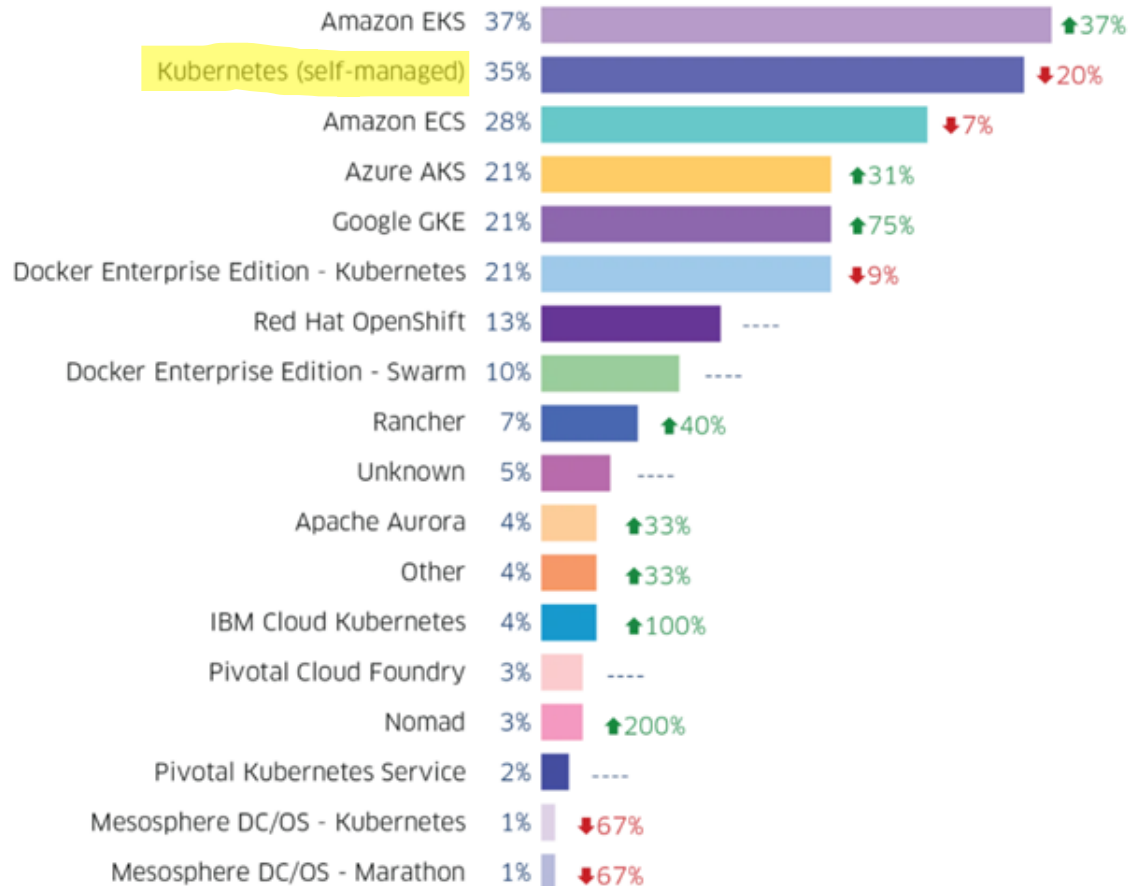


On-Premise

k8s를 활용한 Wordpress 구축 및 배포

시장조사

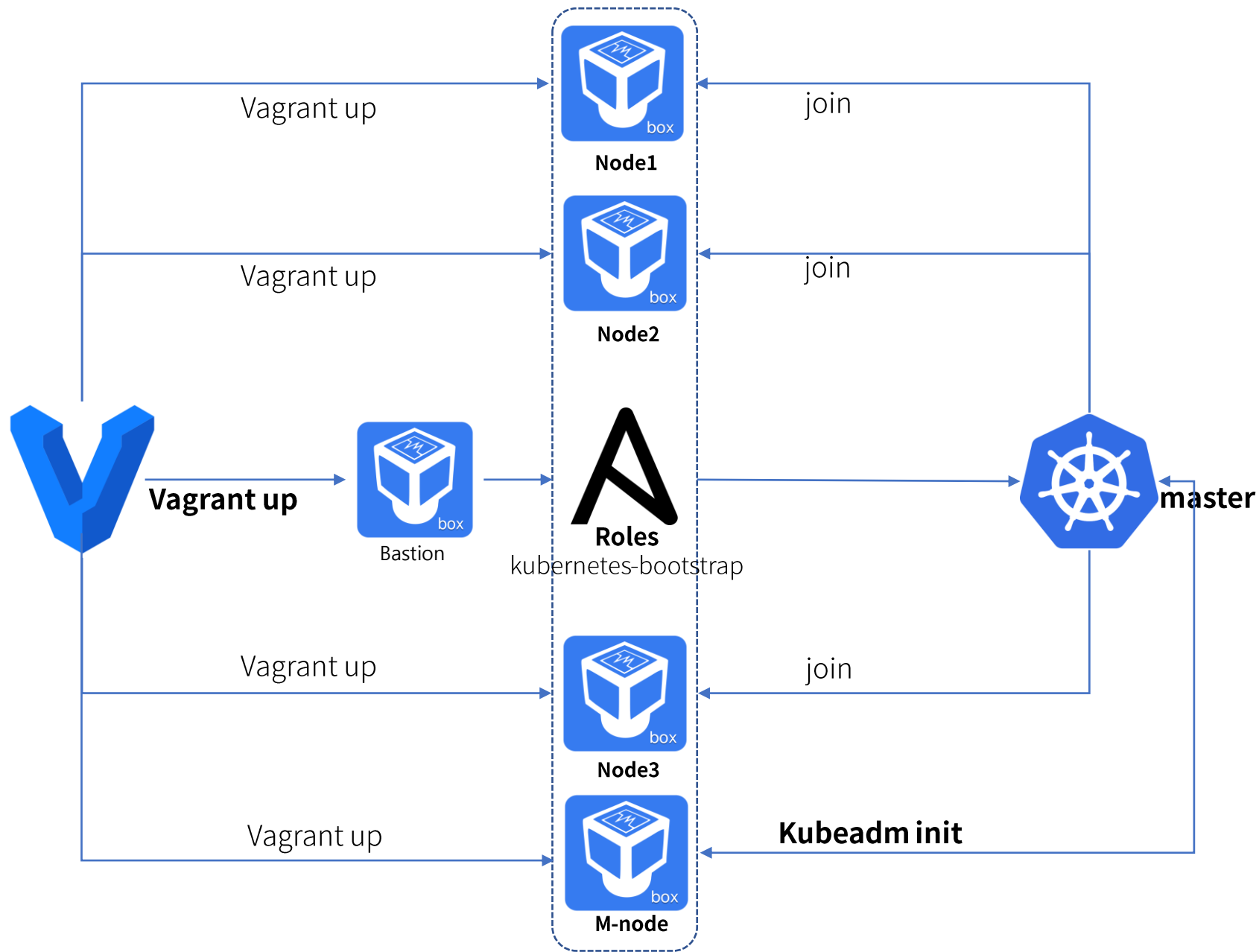
Which of the following container orchestrators do you use? (pick as many as apply)



<출처: StackRox>

온프레미스 환경에서 컨테이너 오케스트레이션을 제공하는 대표적인 솔루션은 도커 스웜, 메소스, 노매드, 쿠버네티스 등이 있다.

Vagrant + Ansible



```

C: > waplz > Vagrantfile
4
5 Vagrant.configure("2") do |config|
6
7   #Web-Master
8   config.vm.define "m1" do |cfg|
9     config.vm.box = "bento/rockylinux-8"
10    cfg.vm.provider "virtualbox" do |vb|
11      vb.name = "m1"
12      vb.cpus = 2
13      vb.memory = 4092
14      vb.customize ["modifyvm", :id, "--groups", "/TEAM"]
15    end
16
17    cfg.vm.host_name = "m1"
18    cfg.vm.network "public_network", ip: "211.100.2.50"
19    cfg.vm.network "forwarded_port", guest: 22, host: 60500, auto_correct: true, id: "ssh"
20    cfg.vm.synced_folder "../data", "/vagrant", disabled: true
21    cfg.vm.provision "shell", path: "bash_ssh_conf_4_CentOS.sh"
22  end
23
24  #Web-Worker1
25  config.vm.define "w1" do |cfg|
26    config.vm.box = "bento/rockylinux-8"
27    cfg.vm.provider "virtualbox" do |vb|
28      vb.name = "w1"
29      vb.cpus = 1
30      vb.memory = 1024
31      vb.customize ["modifyvm", :id, "--groups", "/TEAM"]
32    end
33
34    cfg.vm.host_name = "w1"
35    cfg.vm.network "public_network", ip: "211.100.2.60"
36    cfg.vm.network "forwarded_port", guest: 22, host: 60600, auto_correct: true, id: "ssh"
37    cfg.vm.synced_folder "../data", "/vagrant", disabled: true
38    cfg.vm.provision "shell", path: "bash_ssh_conf_4_CentOS.sh"
39  end

```

```

C: > waplz > $ bash_ssh_conf_4_CentOS.sh
1  #!/usr/bin/env bash
2
3  now=$(date +%m_%d_%Y)
4  cp /etc/ssh/sshd_config /etc/ssh/sshd_config_${now}.backup
5  sed -i -e 's/PasswordAuthentication no/PasswordAuthentication yes/g' /etc/ssh/sshd_config
6  systemctl restart sshd
7
8  cat << EOF > /etc/hosts
9  127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
10 ::1 localhost localhost.localdomain localhost6 localhost6.localdomain6
11 211.100.2.63 b1
12 211.100.2.50 m1
13 211.100.2.60 w1
14 211.100.2.61 w2
15 211.100.2.62 w3
16 EOF
17
18 cat << EOF > /etc/resolv.conf
19 nameserver 1.1.1.1
20 nameserver 8.8.8.8
21 EOF

```

Vagrantfile

가상환경 구축을 위한 vagrantfile 작성
Master Node와 Worker Node 1,2,3 /
Bastion Node 생성

bash_ssh_conf_4_CentOS.sh

ssh 연결 설정 및 hosts 파일에 노드 정보 추가

```

75 #Bastion Host
76 config.vm.define "Bastion-Host1" do |cfg|
77   config.vm.box = "bento/rockylinux-8"
78   cfg.vm.provider "virtualbox" do |vb|
79     vb.name = "Bastion-Host1"
80     vb.cpus = 1
81     vb.memory = 1024
82     vb.customize ["modifyvm", :id, "--groups", "/TEAM"]
83   end
84
85   cfg.vm.host_name = "Bastion-Host1"
86   cfg.vm.network "public_network", ip: "211.100.2.63"
87   cfg.vm.network "forwarded_port", guest: 22, host: 60630, auto_correct: true, id: "ssh"
88   cfg.vm.synced_folder "../data", "/vagrant", disabled: true
89   cfg.vm.provision "shell", inline: "dnf install -y epel-release"
90   cfg.vm.provision "shell", inline: "dnf install -y ansible"
91   cfg.vm.provision "shell", inline: "dnf install -y git"
92   cfg.vm.provision "shell", inline: "dnf install -y tree"
93   cfg.vm.provision "file", source: "ansible_env_ready.yml", destination: "ansible_env_ready.yml"
94   cfg.vm.provision "shell", inline: "ansible-playbook ansible_env_ready.yml"
95   cfg.vm.provision "file", source: "auto_pass.yml", destination: "auto_pass.yml"
96   cfg.vm.provision "shell", inline: "ansible-playbook auto_pass.yml", privileged: false
97   cfg.vm.provision "shell", path: "bash_ssh_conf_4_CentOS.sh"
98   cfg.vm.provision "file", source: "timezone.yml", destination: "timezone.yml"
99   cfg.vm.provision "shell", inline: "ansible-playbook timezone.yml"
100 end
101 end

```

Ansible_env_ready.yml

Inventory에 Node IP 추가

```

C: > waplz > ! ansible_env_ready.yml
1 ---
2 - name: Setup
3   hosts: localhost
4   connection: local
5   gather_facts: no
6
7   tasks:
8     - name: Add "/etc/ansible/hosts"
9       blockinfile:
10         path: /etc/ansible/hosts
11         block: |
12           [master]
13           211.100.2.50
14           [worker]
15           211.100.2.60
16           211.100.2.61
17           211.100.2.62
18     - name: Install git
19       yum:
20         name: git
21         state: present

```

```

75 #Bastion Host
76 config.vm.define "Bastion-Host1" do |cfg|
77   config.vm.box = "bento/rockylinux-8"
78   cfg.vm.provider "virtualbox" do |vb|
79     vb.name = "Bastion-Host1"
80     vb.cpus = 1
81     vb.memory = 1024
82     vb.customize ["modifyvm", :id, "--groups", "/TEAM"]
83   end
84
85   cfg.vm.host_name = "Bastion-Host1"
86   cfg.vm.network "public_network", ip: "211.100.2.63"
87   cfg.vm.network "forwarded_port", guest: 22, host: 60630, auto_correct: true, id: "ssh"
88   cfg.vm.synced_folder "../data", "/vagrant", disabled: true
89   cfg.vm.provision "shell", inline: "dnf install -y epel-release"
90   cfg.vm.provision "shell", inline: "dnf install -y ansible"
91   cfg.vm.provision "shell", inline: "dnf install -y git"
92   cfg.vm.provision "shell", inline: "dnf install -y tree"
93   cfg.vm.provision "file", source: "ansible_env_ready.yml", destination: "ansible_env_ready.yml"
94   cfg.vm.provision "shell", inline: "ansible-playbook ansible_env_ready.yml"
95   cfg.vm.provision "file", source: "auto_pass.yml", destination: "auto_pass.yml"
96   cfg.vm.provision "shell", inline: "ansible-playbook auto_pass.yml", privileged: false
97   cfg.vm.provision "shell", path: "bash_ssh_conf_4_CentOS.sh"
98   cfg.vm.provision "file", source: "timezone.yml", destination: "timezone.yml"
99   cfg.vm.provision "shell", inline: "ansible-playbook timezone.yml"
100 end
101 end

```

Auto_pass.yml

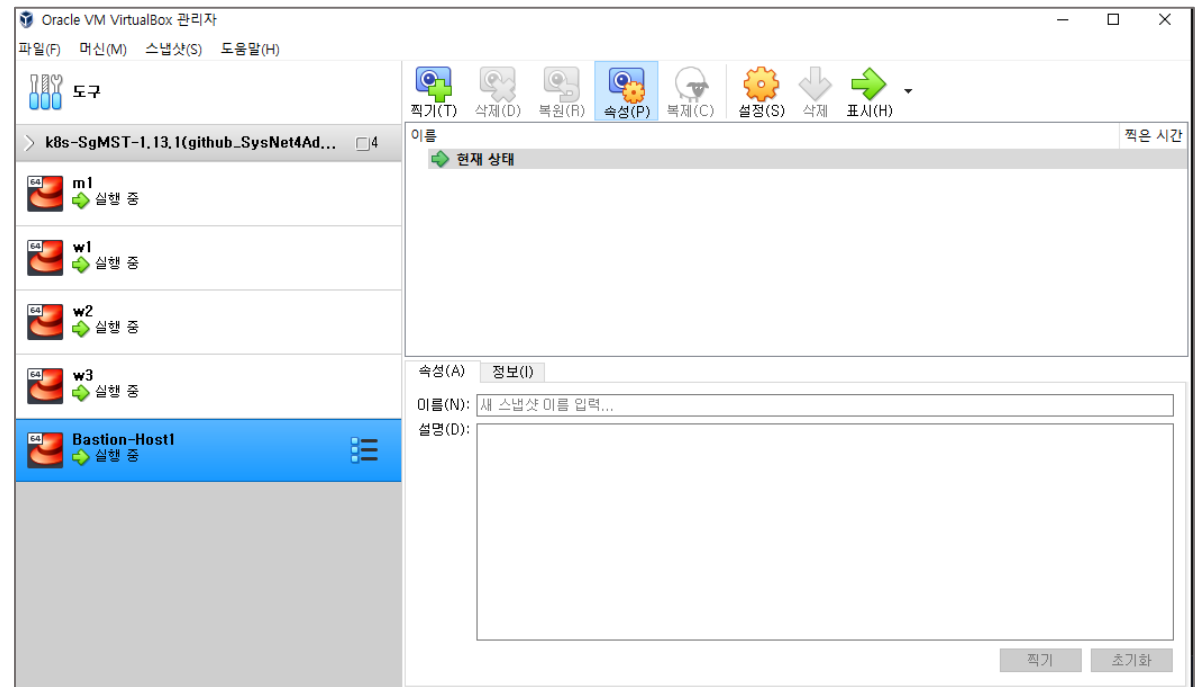
ssh 접속을 위한 key 생성 및 배포

```

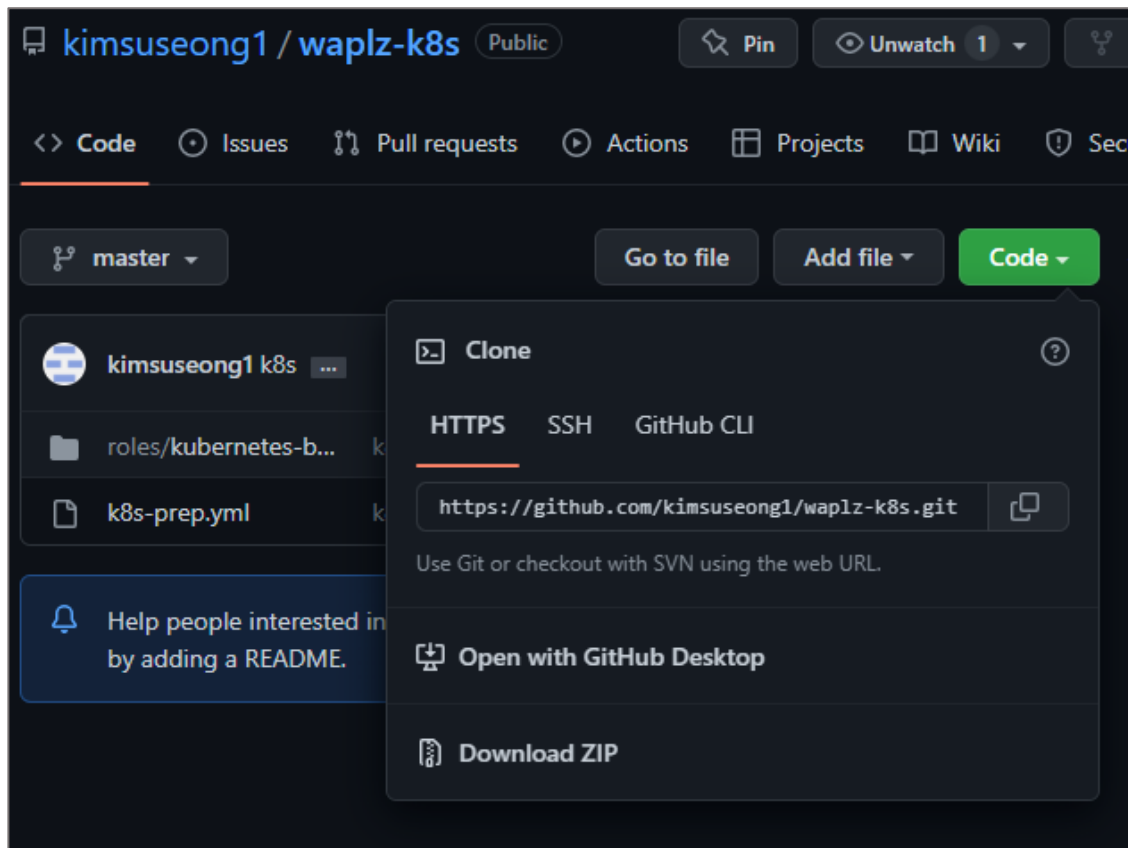
C: > waplz > ! auto_pass.yml
1 |--
2 - name: Create authority between server and nodes
3   hosts: all
4   connection: local
5   serial: 1
6   gather_facts: no
7   vars:
8     ansible_password: vagrant
9
10  tasks:
11    - name: ssh-keyscan for known_hosts file
12      command: /usr/bin/ssh-keyscan -t ecdsa {{ ansible_host }}
13      register: keyscan
14
15
16    - name: input key
17      lineinfile:
18        path: ~/.ssh/known_hosts
19        line: "{{ item }}"
20        create: yes
21      with_items:
22        - "{{ keyscan.stdout_lines }}"
23
24    - name: ssh-keygen for authorized_keys file
25      command: "ssh-keygen -b 2048 -t rsa -f ~/.ssh/id_rsa -q -N ''"
26      ignore_errors: yes
27      run_once: true
28
29    - name: input key for each node
30      connection: ssh
31      authorized_key:
32        user: vagrant
33        state: present
34        key: "{{ lookup('file', '~/.ssh/id_rsa.pub') }}"

```

```
C:\Windows\system32\cmd.exe
Bastion-Host1: ...ignoring
Bastion-Host1: TASK [input key for each node] *****
Bastion-Host1: changed: [211.100.2.61]
Bastion-Host1: PLAY [Create authority between server and nodes] *****
Bastion-Host1: TASK [ssh-keyscan for known_hosts file] *****
Bastion-Host1: changed: [211.100.2.62]
Bastion-Host1: TASK [input key] *****
Bastion-Host1: changed: [211.100.2.62] => (item=211.100.2.62 ecdsa-sha2-nistp256 AAAAE2YjZHNhLlNoYTl1bmlzdHAyNTYAAAAIbmlzdHAyNTYAAAB88Hu8gca0wrUs4NbOkP44r1DWeF0x6
0aC3W3P009RM9Ng2oxrSr5J7/4yb5i2p0Imy8fjMOcBCTe3vr6OG108=)
Bastion-Host1: TASK [ssh-keygen for authorized_keys file] *****
Bastion-Host1: fatal: [211.100.2.62]: FAILED! => {"changed": true, "cmd": ["ssh-keygen", "-b", "2048", "-t", "rsa", "-f", "~/.ssh/id_rsa", "-q", "-N", ""], "delta
: 0:00:00.349484", "end": "2022-10-28 01:29:30.341581", "msg": "non-zero return code", "rc": 1, "start": "2022-10-28 01:29:29.992147", "stderr": "", "stderr_lines":
[], "stdout": "/home/vagrant/.ssh/id_rsa already exists.\nOverwrite (y/n)? ", "stdout_lines": ["/home/vagrant/.ssh/id_rsa already exists.", "Overwrite (y/n)? "]}
Bastion-Host1: ...ignoring
Bastion-Host1: TASK [input key for each node] *****
Bastion-Host1: changed: [211.100.2.62]
Bastion-Host1: PLAY RECAP *****
Bastion-Host1: 211.100.2.50      : ok=4   changed=4   unreachable=0   failed=0   skipped=0   rescued=0   ignored=0
Bastion-Host1: 211.100.2.60      : ok=4   changed=4   unreachable=0   failed=0   skipped=0   rescued=0   ignored=1
Bastion-Host1: 211.100.2.61      : ok=4   changed=4   unreachable=0   failed=0   skipped=0   rescued=0   ignored=1
Bastion-Host1: 211.100.2.62      : ok=4   changed=4   unreachable=0   failed=0   skipped=0   rescued=0   ignored=1
=> Bastion-Host1: Running provisioner: shell...
Bastion-Host1: Running: C:/Users/49L/AppData/Local/Temp/vagrant-shell120221028-15220-yy8vcs.sh
=> Bastion-Host1: Running provisioner: file...
Bastion-Host1: timezone.yml => timezone.yml
=> Bastion-Host1: Running provisioner: shell...
Bastion-Host1: Running: inline script
Bastion-Host1: PLAY [setup timezone] *****
Bastion-Host1: [WARNING]: Could not match supplied host pattern, ignoring: rocky
Bastion-Host1: TASK [set timezone to Asia/Seoul] *****
Bastion-Host1: changed: [localhost]
Bastion-Host1: PLAY RECAP *****
Bastion-Host1: localhost      : ok=1   changed=1   unreachable=0   failed=0   skipped=0   rescued=0   ignored=0
Bastion-Host1:
```



Vagrant up 완료 후 Bastion node로 접속



Git repository

미리 waplz-k8s에 push 해 놓은
Yaml파일들을 clone

```
[vagrant@Bastion-Host1 ~]$  
[vagrant@Bastion-Host1 ~]$ git clone https://github.com/kimsuseong1/waplz-k8s.git  
Cloning into 'waplz-k8s'...  
remote: Enumerating objects: 22, done.  
remote: Counting objects: 100% (22/22), done.  
remote: Compressing objects: 100% (19/19), done.  
Receiving objects: 100% (22/22), 4.38 KiB | 4.38 MiB/s, done.  
remote: Total 22 (delta 0), reused 22 (delta 0), pack-reused 0  
[vagrant@Bastion-Host1 ~]$ ll  
total 12  
-rw-rw-r--. 1 vagrant vagrant 417 Oct 31 11:41 ansible_env_ready.yml  
-rw-rw-r--. 1 vagrant vagrant 857 Oct 31 11:41 auto_pass.yml  
-rw-rw-r--. 1 vagrant vagrant 170 Oct 31 11:42 timezone.yml  
drwxrwxr-x. 4 vagrant vagrant 51 Oct 31 11:47 waplz-k8s  
[vagrant@Bastion-Host1 ~]$ cd waplz-k8s/  
[vagrant@Bastion-Host1 waplz-k8s]$ tree  
  
.  
├── k8s-prep.yml  
├── roles  
│   └── kubernetes-bootstrap  
│       ├── defaults  
│       │   └── main.yml  
│       ├── tasks  
│       │   ├── configure_firewalld.yml  
│       │   ├── configure_timezone_ntp.yml  
│       │   ├── disable_swap.yml  
│       │   ├── install_k8s_packages.yml  
│       │   ├── load_kernel_modules_sysctl.yml  
│       │   ├── main.yml  
│       │   ├── pre_setup.yml  
│       │   └── setup_crio.yml  
│       ├── templates  
│       │   ├── crio.repo.j2  
│       │   ├── kernel_modules.conf.j2  
│       │   └── kubernetes.repo.j2  
│       └── vars  
│           └── RedHat8.yml  
└──
```

6 directories, 14 files

Git clone, Tree

Github 링크로 모든 파일 실행
Kubernetes Master Node, Worker Node
자동 설치


```
[vagrant@bastion-host1 k8s]$
[vagrant@bastion-host1 k8s]$ ansible-playbook k8s-prep.yml

PLAY [Setup proxy] *****

TASK [Gathering Facts] *****
ok: [211.100.2.50]
ok: [211.100.2.60]
ok: [211.100.2.61]
ok: [211.100.2.62]

TASK [kubernetes-bootstrap : Add the OS specific variables] *****
ok: [211.100.2.50] => (item=/home/vagrant/k8s/roles/kubernetes-bootstrap/vars/RedHat8.yml)
ok: [211.100.2.60] => (item=/home/vagrant/k8s/roles/kubernetes-bootstrap/vars/RedHat8.yml)
ok: [211.100.2.61] => (item=/home/vagrant/k8s/roles/kubernetes-bootstrap/vars/RedHat8.yml)
ok: [211.100.2.62] => (item=/home/vagrant/k8s/roles/kubernetes-bootstrap/vars/RedHat8.yml)

TASK [kubernetes-bootstrap : Include Pre-raps setup task] *****
included: /home/vagrant/k8s/roles/kubernetes-bootstrap/tasks/pre_setup.yml for 211.100.2.50, 211.100.2.60, 211.100.2.61, 211.100.2.62

TASK [kubernetes-bootstrap : Put SELinux in permissive mode] *****
ok: [211.100.2.50]
ok: [211.100.2.61]
ok: [211.100.2.62]
ok: [211.100.2.60]

TASK [kubernetes-bootstrap : Install some packages needed to configure the nodes] *****
changed: [211.100.2.50] => (item=['vim', 'bash-completion', 'wget', 'curl', 'firewalld', 'python3-firewall', 'yum-utils', 'lvm2', 'device-mapper', 'persistent-data', 'iproute-tc'])
changed: [211.100.2.60] => (item=['vim', 'bash-completion', 'wget', 'curl', 'firewalld', 'python3-firewall', 'yum-utils', 'lvm2', 'device-mapper', 'persistent-data', 'iproute-tc'])
changed: [211.100.2.62] => (item=['vim', 'bash-completion', 'wget', 'curl', 'firewalld', 'python3-firewall', 'yum-utils', 'lvm2', 'device-mapper', 'persistent-data', 'iproute-tc'])
changed: [211.100.2.61] => (item=['vim', 'bash-completion', 'wget', 'curl', 'firewalld', 'python3-firewall', 'yum-utils', 'lvm2', 'device-mapper', 'persistent-data', 'iproute-tc'])

TASK [kubernetes-bootstrap : Disable firewalld service] *****
ok: [211.100.2.50]
ok: [211.100.2.61]
ok: [211.100.2.60]
ok: [211.100.2.62]

TASK [kubernetes-bootstrap : Include task to disable swap] *****
included: /home/vagrant/k8s/roles/kubernetes-bootstrap/tasks/disable_swap.yml for 211.100.2.50, 211.100.2.60, 211.100.2.61, 211.100.2.62

TASK [kubernetes-bootstrap : Disable SWAP since kubernetes can't work with swap enabled(1/2)] *****
changed: [211.100.2.50]
changed: [211.100.2.60]
```

```
changed: [211.100.2.62]

TASK [kubernetes-bootstrap : Start and enable crio service] *****
changed: [211.100.2.50]
changed: [211.100.2.60]
changed: [211.100.2.61]
changed: [211.100.2.62]

TASK [kubernetes-bootstrap : Include task to configure containerd container runtime] *****
skipping: [211.100.2.50]
skipping: [211.100.2.60]
skipping: [211.100.2.61]
skipping: [211.100.2.62]

TASK [kubernetes-bootstrap : Include task to install k8s packages] *****
included: /home/vagrant/k8s/roles/kubernetes-bootstrap/tasks/install_k8s_packages.yml for 211.100.2.50, 211.100.2.60, 211.100.2.61, 211.100.2.62

TASK [kubernetes-bootstrap : Add kubernetes repository] *****
changed: [211.100.2.50]
changed: [211.100.2.60]
changed: [211.100.2.61]
changed: [211.100.2.62]

TASK [kubernetes-bootstrap : Install kubernetes packages] *****
changed: [211.100.2.50]
changed: [211.100.2.61]
changed: [211.100.2.60]
changed: [211.100.2.62]

TASK [kubernetes-bootstrap : Enable kubelet service] *****
changed: [211.100.2.50]
changed: [211.100.2.61]
changed: [211.100.2.62]
changed: [211.100.2.60]

TASK [kubernetes-bootstrap : Include task to configure firewalld] *****
skipping: [211.100.2.50]
skipping: [211.100.2.60]
skipping: [211.100.2.61]
skipping: [211.100.2.62]

PLAY RECAP *****
211.100.2.50      : ok=28  changed=15  unreachable=0  failed=0  skipped=3  rescued=0  ignored=0
211.100.2.60      : ok=28  changed=15  unreachable=0  failed=0  skipped=3  rescued=0  ignored=0
211.100.2.61      : ok=28  changed=15  unreachable=0  failed=0  skipped=3  rescued=0  ignored=0
211.100.2.62      : ok=28  changed=15  unreachable=0  failed=0  skipped=3  rescued=0  ignored=0

[vagrant@bastion-host1 k8s]$
```

Ansible-playbook 명령어 실행

```
C: > role > k8s > ! k8s-prep.yml
1  ---
2  - name: Setup proxy
3    hosts: all
4    become: yes
5    become_method: sudo
6
7  vars:
8    k8s_version: "1.22"
9    selinux_state: permissive
10   timezone: "Asia/Seoul"
11   k8s_cni: calico
12   container_runtime: cri-o
13   pod_network_cidr: "172.16.0.0/16"
14   configure_firewalld: false
15   setup_proxy: false
16   proxy_server: "proxy.example.com:8080"
17   docker_proxy_exclude: "localhost,127.0.0.1"
18   roles:
19     - kubernetes-bootstrap
```

K8s-prep.yml

Pod network cidr
172.16.0.0/16으로 고정

```

C: > role > k8s > roles > kubernetes-bootstrap > tasks > ! main.yml
1  ---
2  - name: Add the OS specific variables
3    include_vars: "{{ item }}"
4    with_first_found:
5      - "{{ ansible_os_family }}{{ ansible_distribution_major_version }}.yaml"
6      - "{{ ansible_os_family }}.yaml"
7
8  - name: Include Pre-reps setup task
9    include_tasks: pre_setup.yml
10
11 - name: Include task to disable swap
12   include_tasks: disable_swap.yml
13
14 - name: Include task to configure timezone and ntp
15   include_tasks: configure_timezone_ntp.yml
16
17 - name: Include task to load required kernel modules and sysctl configs
18   include_tasks: load_kernel_modules_sysctl.yml
19
20 - name: Include task to configure cri-o container runtime
21   include_tasks: setup_crio.yml
22   when: container_runtime == "cri-o"
23
24 - name: Include task to install k8s packages
25   include_tasks: install_k8s_packages.yml
26
27 - name: Include task to configure firewalld
28   include_tasks: configure_firewalld.yml
29   when: configure_firewalld
30

```

tasks_main.yml

Include_tasks 를 이용해서
이하 yaml 파일을 실행

```

C: > role > k8s > roles > kubernetes-bootstrap > tasks > ! pre_setup.yml
1  |---
2  - name: Put SELinux in permissive mode
3    selinux:
4      policy: targeted
5      state: "{{ selinux_state }}"
6
7  #- name: Update system packages
8  # package:
9  #   name: "*"
10 #   state: latest
11
12 - name: Install some packages needed to configure the nodes
13   ansible.builtin.package:
14     name: "{{ item }}"
15   loop:
16     - "{{ basic_packages }}"
17
18 - name: Disable firewalld service
19   ansible.builtin.service:
20     name: firewalld
21     state: stopped
22     enabled: no

```

pre_setup.yml

-기초 설정-

- * SELinux 의 동작 모드를 permissive로 변경
- * 필수 패키지 다운로드
- * 방화벽 사용 중지

```

C: > role > k8s > roles > kubernetes-bootstrap > tasks > ! main.yml
1  ---
2  - name: Add the OS specific variables
3    include_vars: "{{ item }}"
4    with_first_found:
5      - "{{ ansible_os_family }}{{ ansible_distribution_major_version }}.yaml"
6      - "{{ ansible_os_family }}.yaml"
7
8  - name: Include Pre-reps setup task
9    include_tasks: pre_setup.yml
10
11 - name: Include task to disable swap
12   include_tasks: disable_swap.yml
13
14 - name: Include task to configure timezone and ntp
15   include_tasks: configure_timezone_ntp.yml
16
17 - name: Include task to load required kernel modules and sysctl configs
18   include_tasks: load_kernel_modules_sysctl.yml
19
20 - name: Include task to configure cri-o container runtime
21   include_tasks: setup_crio.yml
22   when: container_runtime == "cri-o"
23
24 - name: Include task to install k8s packages
25   include_tasks: install_k8s_packages.yml
26
27 - name: Include task to configure firewalld
28   include_tasks: configure_firewalld.yml
29   when: configure_firewalld
30

```

```

C: > role > k8s > roles > kubernetes-bootstrap > tasks > ! disable_swap.yml
1  |---
2  | - name: Disable SWAP since kubernetes can't work with swap enabled(1/2)
3  |   ansible.builtin.shell: |
4  |     swapoff -a
5  |
6  | - name: Disable SWAP in fstab since kubernetes can't work with swap enabled(2/2)
7  |   ansible.builtin.replace:
8  |     path: /etc/fstab
9  |     regexp: '^(#[^#].*?\sswap\s+.*)\$'
10 |     replace: '# \1'

```

Disable_swap.yml

Kubernetes 설치 시 swap 메모리를
사용하면 안되기 때문에 swapoff 설정

/etc/fstab에도 swap 메모리가 마운트 되지
않게 적용

```

C: > role > k8s > roles > kubernetes-bootstrap > tasks > ! main.yml
1 ---
2 - name: Add the OS specific variables
3   include_vars: "{{ item }}"
4   with_first_found:
5     - "{{ ansible_os_family }}{{ ansible_distribution_major_version }}.yaml"
6     - "{{ ansible_os_family }}.yaml"
7
8 - name: Include Pre-reps setup task
9   include_tasks: pre_setup.yml
10
11 - name: Include task to disable swap
12   include_tasks: disable_swap.yml
13
14 - name: Include task to configure timezone and ntp
15   include_tasks: configure_timezone_ntp.yml
16
17 - name: Include task to load required kernel modules and sysctl configs
18   include_tasks: load_kernel_modules_sysctl.yml
19
20 - name: Include task to configure cri-o container runtime
21   include_tasks: setup_crio.yml
22   when: container_runtime == "cri-o"
23
24 - name: Include task to install k8s packages
25   include_tasks: install_k8s_packages.yml
26
27 - name: Include task to configure firewalld
28   include_tasks: configure_firewalld.yml
29   when: configure_firewalld
30

```

```

C: > role > k8s > roles > kubernetes-bootstrap > tasks > ! configure_timezone.yml
1 |---
2 |  - name: Configure timezone on all nodes
3 |  community.general.timezone:
4 |    name: "{{ timezone }}"
5 |
6 |  - name: Ensure chrony package is installed
7 |  ansible.builtin.package:
8 |    name: chrony
9 |    state: present
10 |
11 |  - name: Enable and start chronyd service
12 |  ansible.builtin.service:
13 |    name: chronyd
14 |    state: started
15 |    enabled: yes
16 |
17 |  - name: Synchronize time manually
18 |  ansible.builtin.shell: chronyc sources

```

Configure_timezone.yml

시간 동기화를 위해 chrony service
설치 및 가동

```

C: > role > k8s > roles > kubernetes-bootstrap > tasks > ! main.yml
1  ---
2  - name: Add the OS specific variables
3    include_vars: "{{ item }}"
4    with_first_found:
5      - "{{ ansible_os_family }}{{ ansible_distribution_major_version }}.yaml"
6      - "{{ ansible_os_family }}.yaml"
7
8  - name: Include Pre-reps setup task
9    include_tasks: pre_setup.yml
10
11 - name: Include task to disable swap
12   include_tasks: disable_swap.yml
13
14 - name: Include task to configure timezone and ntp
15   include_tasks: configure_timezone_ntp.yml
16
17 - name: Include task to load required kernel modules and sysctl configs
18   include_tasks: load_kernel_modules_sysctl.yml
19
20 - name: Include task to configure cri-o container runtime
21   include_tasks: setup_crio.yml
22   when: container_runtime == "cri-o"
23
24 - name: Include task to install k8s packages
25   include_tasks: install_k8s_packages.yml
26
27 - name: Include task to configure firewalld
28   include_tasks: configure_firewalld.yml
29   when: configure_firewalld
30

```

```

C: > role > k8s > roles > kubernetes-bootstrap > tasks > ! load_kernel_modules_sysctl.yml
1  |---
2  - name: Load required modules
3    community.general.modprobe:
4      name: "{{ item }}"
5      state: present
6    with_items:
7      - br_netfilter
8      - overlay
9      - ip_vs
10     - ip_vs_rr
11     - ip_vs_wrr
12     - ip_vs_sh
13     - nf_conntrack
14
15 - name: Create the .conf file to load the modules at bootup
16   ansible.builtin.template:
17     src: kernel_modules.conf.j2
18     dest: /etc/modules-load.d/k8s_kernel_modules.conf
19
20 - name: Modify sysctl entries
21   ansible.posix.sysctl:
22     name: '{{ item.key }}'
23     value: '{{ item.value }}'
24     sysctl_set: yes
25     state: present
26     reload: yes
27     ignore_errors: True
28   with_items:
29     - {key: net.bridge.bridge-nf-call-ip6tables, value: 1}
30     - {key: net.bridge.bridge-nf-call-iptables, value: 1}
31     - {key: net.ipv4.ip_forward, value: 1}

```

```

C: > role > k8s > roles > kubernetes-bootstrap > templates > ≡ kernel_modules.conf.j2
1  |overlay
2  |br_netfilter
3  |ip_vs
4  |ip_vs_rr
5  |ip_vs_wrr
6  |ip_vs_sh
7  |nf_conntrack

```

Load_kernel_modules_sysctl.yml

쿠버네티스 실행 시 필요한 모듈 로드
및 활성화를 위한 모듈 파라미터 수정

Setup_crio.yml

```
C: > role > k8s > roles > kubernetes-bootstrap > tasks > ! main.yml
```

```
1 ---
2 - name: Add the OS specific variables
3   include_vars: "{{ item }}"
4   with_first_found:
5     - "{{ ansible_os_family }}{{ ansible_distribution_major_version }}.yaml"
6     - "{{ ansible_os_family }}.yaml"
7
8 - name: Include Pre-reps setup task
9   include_tasks: pre_setup.yml
10
11 - name: Include task to disable swap
12   include_tasks: disable_swap.yml
13
14 - name: Include task to configure timezone and ntp
15   include_tasks: configure_timezone_ntp.yml
16
17 - name: Include task to load required kernel modules and sysctl configs
18   include_tasks: load_kernel_modules_sysctl.yml
19
20 - name: Include task to configure cri-o container runtime
21   include_tasks: setup_crio.yml
22   when: container_runtime == "cri-o"
23
24 - name: Include task to install k8s packages
25   include_tasks: install_k8s_packages.yml
26
27 - name: Include task to configure firewalld
28   include_tasks: configure_firewalld.yml
29   when: configure_firewalld
30
```

```
C: > role > k8s > roles > kubernetes-bootstrap > tasks > ! setup_crio.yml
```

```
1 ---
2 - name: Configure Cri-o YUM repository
3   ansible.builtin.template:
4     src: crio.repo.j2
5     dest: /etc/yum.repos.d/crio.repo
6
7 - name: Setup required sysctl params
8   ansible.posix.sysctl:
9     name: '{{ item.key }}'
10    value: '{{ item.value }}'
11    sysctl_set: yes
12    state: present
13    reload: yes
14    with_items:
15      - {key: net.bridge.bridge-nf-call-ip6tables, value: 1}
16      - {key: net.bridge.bridge-nf-call-iptables, value: 1}
17      - {key: net.ipv4.ip_forward, value: 1}
18
19 - name: Install cri-o
20   yum:
21     name: cri-o
22     state: latest
23     update_cache: yes
24
25 - name: Configure cri-o subnet
26   ansible.builtin.replace:
27     path: /etc/cni/net.d/100-crio-bridge.conf
28     regexp: '10\.85\.0\.0/16'
29     replace: '{{ pod_network_cidr }}'
30     ignore_errors: True
31
32 - name: Start and enable crio service
33   ansible.builtin.service:
34     name: crio
35     state: restarted
36     enabled: yes
```

* Cri-o repository, 파라미터 설정

* Cri-o 다운로드 및 서비스 시작

* '10\.85\.0\.0/16'으로 표시되어 있는 서브넷을 k8s-prep.yml에서 설정한 pod_network_cidr로 치환 (172.16.0.0/16)

```
crio.repo.j2 x kernel_modules.conf.j2 kubernetes.repo.j2 ! RedHat8.yml
C: > role > k8s > roles > kubernetes-bootstrap > templates > crio.repo.j2
1 [devel_kubic_libcontainers_stable]
2 name=Stable Releases of Upstream github.com/containers packages
3 type=rpm-md
4 baseurl=https://download.opensuse.org/repositories/devel:/kubic:/libcontainers:/stable/CentOS_{{ ansible_distribution_major_version }}/
5 gpgcheck=1
6 gpgkey=https://download.opensuse.org/repositories/devel:/kubic:/libcontainers:/stable/CentOS_{{ ansible_distribution_major_version }}/repodata/repomd.xml.key
7 enabled=1
8
9 [devel_kubic_libcontainers_stable_cri-o_{{ k8s_version }}]
10 name=devel:kubic:libcontainers:stable:cri-o:{{ k8s_version }} (CentOS_{{ ansible_distribution_major_version }})
11 type=rpm-md
12 baseurl=https://download.opensuse.org/repositories/devel:/kubic:/libcontainers:/stable:/cri-o:{{ k8s_version }}/CentOS_{{ ansible_distribution_major_version }}/
13 gpgcheck=1
14 baseurl=https://download.opensuse.org/repositories/devel:/kubic:/libcontainers:/stable:/cri-o:{{ k8s_version }}/CentOS_{{ ansible_distribution_major_version }}/repodata/repomd.key
15 enabled=1
16
```

Crio.repo.j2

Cri-o repository 주소를
포함하고 있는 진자 파일


```
C: > role > k8s > roles > kubernetes-bootstrap > tasks > ! main.yml
1 ---
2 - name: Add the OS specific variables
3   include_vars: "{{ item }}"
4   with_first_found:
5     - "{{ ansible_os_family }}{{ ansible_distribution_major_version }}.yaml"
6     - "{{ ansible_os_family }}.yaml"
7
8 - name: Include Pre-reps setup task
9   include_tasks: pre_setup.yml
10
11 - name: Include task to disable swap
12   include_tasks: disable_swap.yml
13
14 - name: Include task to configure timezone and ntp
15   include_tasks: configure_timezone_ntp.yml
16
17 - name: Include task to load required kernel modules and sysctl configs
18   include_tasks: load_kernel_modules_sysctl.yml
19
20 - name: Include task to configure cri-o container runtime
21   include_tasks: setup_crio.yml
22   when: container_runtime == "cri-o"
23
24 - name: Include task to install k8s packages
25   include_tasks: install_k8s_packages.yml
26
27 - name: Include task to configure firewalld
28   include_tasks: configure_firewalld.yml
29   when: configure_firewalld
30
```

```
C: > role > k8s > roles > kubernetes-bootstrap > tasks > ! install_k8s_packages.yml
1 ---
2 - name: Add kubernetes repository
3   ansible.builtin.template:
4     src: 'kubernetes.repo.j2'
5     dest: /etc/yum.repos.d/kubernetes.repo
6
7 - name: Install kubernetes packages
8   yum:
9     name: [kubelet,kubeadm,kubect1]
10    disabled_excludes: kubernetes
11
12 - name: Enable kubelet service
13   ansible.builtin.service:
14     name: kubelet
15     enabled: yes
```

```
C: > role > k8s > roles > kubernetes-bootstrap > templates > ≡ kubernetes.repo.j2
1 [kubernetes]
2 name=kubernetes
3 baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-x86_64
4 enabled=1
5 gpgcheck=1
6 repo_gpgcheck=1
7 gpgkey=https://packages.cloud.google.com/yum/doc/yum-key.gpg
8 https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg
9 exclude=kubelet kubeadm kubect1
```

Install_k8s_packages.yml

Kubernetes repository 추가,
패키지 다운로드 및 kubelet 서비스 가동

Kubernetes.repo.j2

Kubernetes repository 주소를
포함하고 있는 진자 파일

```

C: > role > k8s > roles > kubernetes-bootstrap > tasks > ! main.yml
1 ---
2 - name: Add the OS specific variables
3   include_vars: "{{ item }}"
4   with_first_found:
5     - "{{ ansible_os_family }}{{ ansible_distribution_major_version }}.yaml"
6     - "{{ ansible_os_family }}.yaml"
7
8 - name: Include Pre-reps setup task
9   include_tasks: pre_setup.yml
10
11 - name: Include task to disable swap
12   include_tasks: disable_swap.yml
13
14 - name: Include task to configure timezone and ntp
15   include_tasks: configure_timezone_ntp.yml
16
17 - name: Include task to load required kernel modules and sysctl configs
18   include_tasks: load_kernel_modules_sysctl.yml
19
20 - name: Include task to configure cri-o container runtime
21   include_tasks: setup_crio.yml
22   when: container_runtime == "cri-o"
23
24 - name: Include task to install k8s packages
25   include_tasks: install_k8s_packages.yml
26
27 - name: Include task to configure firewalld
28   include_tasks: configure_firewalld.yml
29   when: configure_firewalld
30

```

```

C: > role > k8s > roles > kubernetes-bootstrap > tasks > ! configure_firewalld.yml
1 ---
2 - name: Install firewalld
3   ansible.builtin.package:
4     name: firewalld
5     state: present
6
7 - name: Start and enable firewalld
8   ansible.builtin.package:
9     name: firewalld
10    state: started
11    enabled: yes
12
13 - name: Configure firewalld on worker nodes
14   ansible.posix.firewalld:
15     port: "{{ item }}/tcp"
16     permanent: yes
17     state: enabled
18   with_items: '{{ k8s_master_ports }}'
19   when: ("node" in ansible_hostname or "worker" in ansible_hostname)
20
21 - name: Open flannel ports on the firewall
22   ansible.posix.firewalld:
23     port: "{{ item }}/udp"
24     permanent: yes
25     state: enabled
26   with_items: "{{ flannel_udp_ports }}"
27   when: k8s_cni == "flannel"
28
29 - name: Open calico UDP ports on the firewall
30   ansible.posix.firewalld:
31     port: "{{ item }}/udp"
32     permanent: yes
33     state: enabled
34   with_items: "{{ calico_udp_ports }}"
35   when: k8s_cni == "calico"
36
37 - name: Open calico TCP ports on the firewall
38   ansible.posix.firewalld:
39     port: "{{ item }}/TCP"
40     permanent: yes
41     state: enabled
42   with_items: "{{ calico_tcp_ports }}"
43   when: k8s_cni == "calico"
44
45 - name: Reload firewalld
46   shell: firewall-cmd --reload

```

Configure_firewalld.yml

master node, worker node
방화벽 설치 및 포트 허용, 재시작

```
C: > role > k8s > roles > kubernetes-bootstrap > vars > ! RedHat8.yml
```

```
1  |--  
2  basic_packages:  
3    - vim  
4    - bash-completion  
5    - wget  
6    - curl  
7    - firewalld  
8    - python3-firewalld  
9    - yum-utils  
10   - lvm2  
11   - device-mapper-persistent-data  
12   - iproute-tc
```

RedHat8.yml

기본적인 명령어 packages가 담긴 yaml 파일

```
[vagrant@m1 waplz-web]$
[vagrant@m1 waplz-web]$ sudo sysctl -p
net.bridge.bridge-nf-call-ip6tables = 1
net.bridge.bridge-nf-call-iptables = 1
net.ipv4.ip_forward = 1
[vagrant@m1 waplz-web]$ systemctl status cri-o
● cri-o.service - Container Runtime Interface for OCI (CRI-O)
   Loaded: loaded (/usr/lib/systemd/system/crio.service; enabled; ve>
   Active: active (running) since Mon 2022-10-31 23:42:23 KST; 10h a>
     Docs: https://github.com/cri-o/cri-o
    Main PID: 824 (crio)
         Tasks: 15
        Memory: 173.5M
         CGroup: /system.slice/crio.service
                └─824 /usr/bin/crio
```

lines 1-9/9 (END)

```
[vagrant@w2 ~]$
[vagrant@w2 ~]$ sudo sysctl -p
net.bridge.bridge-nf-call-ip6tables = 1
net.bridge.bridge-nf-call-iptables = 1
net.ipv4.ip_forward = 1
[vagrant@w2 ~]$ systemctl status cri-o
● cri-o.service - Container Runtime Interface for OCI (CRI-O)
   Loaded: loaded (/usr/lib/systemd/system/crio.service; enabled; ve>
   Active: active (running) since Mon 2022-10-31 23:42:38 KST; 10h a>
     Docs: https://github.com/cri-o/cri-o
    Main PID: 766 (crio)
         Tasks: 13
        Memory: 80.4M
         CGroup: /system.slice/crio.service
                └─766 /usr/bin/crio
```

lines 1-9/9 (END)

☐ Exclude "211.100.2.50" from MultiExec mode

```
[vagrant@w3 ~]$
[vagrant@w3 ~]$ sudo sysctl -p
net.bridge.bridge-nf-call-ip6tables = 1
net.bridge.bridge-nf-call-iptables = 1
net.ipv4.ip_forward = 1
[vagrant@w3 ~]$ systemctl status cri-o
● cri-o.service - Container Runtime Interface for OCI (CRI-O)
   Loaded: loaded (/usr/lib/systemd/system/crio.service; enabled; ve>
   Active: active (running) since Mon 2022-10-31 23:42:42 KST; 10h a>
     Docs: https://github.com/cri-o/cri-o
    Main PID: 760 (crio)
         Tasks: 14
        Memory: 120.7M
         CGroup: /system.slice/crio.service
                └─760 /usr/bin/crio
```

lines 1-9/9 (END)

☐ Exclude "211.100.2.61" from MultiExec mode

```
[vagrant@w1 ~]$
[vagrant@w1 ~]$ sudo sysctl -p
net.bridge.bridge-nf-call-ip6tables = 1
net.bridge.bridge-nf-call-iptables = 1
net.ipv4.ip_forward = 1
[vagrant@w1 ~]$ systemctl status cri-o
● cri-o.service - Container Runtime Interface for OCI (CRI-O)
   Loaded: loaded (/usr/lib/systemd/system/crio.service; enabled; ve>
   Active: active (running) since Mon 2022-10-31 23:42:35 KST; 10h a>
     Docs: https://github.com/cri-o/cri-o
    Main PID: 769 (crio)
         Tasks: 15
        Memory: 115.2M
         CGroup: /system.slice/crio.service
                └─769 /usr/bin/crio
```

lines 1-9/9 (END)

작동 확인

각 노드에 sysctl -p , systemctl status cri-o 명령어로
의도한 role이 제대로 설치되었고 작동하는지 확인

```
[vagrant@m1 ~]$ sudo kubeadm init --apiserver-advertise-address=211.100.2.50 --pod-network-cidr=172.16.0.0/16
Your Kubernetes control-plane has initialized successfully!

To start using your cluster, you need to run the following as a regular user:

  mkdir -p $HOME/.kube
  sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
  sudo chown $(id -u):$(id -g) $HOME/.kube/config

Alternatively, if you are the root user, you can run:

  export KUBECONFIG=/etc/kubernetes/admin.conf

You should now deploy a pod network to the cluster.
Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:
https://kubernetes.io/docs/concepts/cluster-administration/addons/

Then you can join any number of worker nodes by running the following on each as root:

kubeadm join 211.100.2.50:6443 --token cq93h6.7og90l3nr71lrxrg \
  --discovery-token-ca-cert-hash sha256:5d3bf5af1d2e00d1a0fd1a1f970df6b0636a2061c3c870dbafa2c564a1c964eb
[vagrant@m1 ~]$
```

Kubernetes 시작

master node에서 kubeadm init 코드 입력으로
Kubernetes 시작

```
[vagrant@m1 ~]$ mkdir -p $HOME/.kube
[vagrant@m1 ~]$ sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
[vagrant@m1 ~]$ sudo chown $(id -u):$(id -g) $HOME/.kube/config
[vagrant@m1 ~]$
```

Kubectl 명령어 준비

```
[vagrant@m1 ~]$ curl https://raw.githubusercontent.com/projectcalico/calico/v3.24.1/manifests/calico.yaml -O
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100 229k  100 229k    0     0  498k      0 --:--:-- --:--:-- --:--:-- 497k
[vagrant@m1 ~]$
```

Calico CNI 설치를 위해 링크 입력

```

- name: CALICO_IPV4POOL_CIDR
  value: "172.16.0.0/16"
# Disable file logging so `kubectl logs` works.
- name: CALICO_DISABLE_FILE_LOGGING
  value: "true"
# Set Felix endpoint to host default action to ACCEPT.
- name: FELIX_DEFAULTENDPOINTTOHOSTACTION
  value: "ACCEPT"
# Disable IPv6 on Kubernetes.
- name: FELIX_IPV6SUPPORT
  value: "false"
- name: FELIX_HEALTHENABLED
  value: "true"
securityContext:
  privileged: true
resources:
  requests:
    cpu: 250m
lifecycle:
  preStop:
    exec:
      command:
        - /bin/calico-node
        - -shutdown
livenessProbe:
--

```

```

[vagrant@m1 ~]$ kubectl apply -f calico.yaml
poddruptionbudget.policy/calico-kube-controllers created
serviceaccount/calico-kube-controllers created
serviceaccount/calico-node created
configmap/calico-config created
customresourcedefinition.apiextensions.k8s.io/bgpconfigurations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/bgppeers.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/blockaffinities.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/caliconodestatuses.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/clusterinformations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/felixconfigurations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/globalnetworkpolicies.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/globalnetworksets.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/hostendpoints.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipamblocks.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipamconfigs.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipamhandles.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ippools.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipreservations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/kubecontrollersconfigurations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/networkpolicies.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/networksets.crd.projectcalico.org created
clusterrole.rbac.authorization.k8s.io/calico-kube-controllers created
clusterrole.rbac.authorization.k8s.io/calico-node created
clusterrolebinding.rbac.authorization.k8s.io/calico-kube-controllers created
clusterrolebinding.rbac.authorization.k8s.io/calico-node created
daemonset.apps/calico-node created
deployment.apps/calico-kube-controllers created
[vagrant@m1 ~]$

```

Calico.yaml

Pod_network_cidr 주소 변경

설치 시 지정한 Pod_network 주소로 설정 후
매니페스트를 실행하여 Calico CNI 설치


```
[vagrant@w2 ~]$ sudo kubeadm join 211.100.2.50:6443 --token cq93h6.7o
g90l3nr71lrxrg --discovery-token-ca-cert-hash sha256:5d3bf5af1d2e00d1
a0fd1a1f970df6b0636a2061c3c870dbafa2c564a1c964eb
[preflight] Running pre-flight checks
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n ku
be-system get cm kubeadm-config -o yaml'
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubel
et/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "
/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Starting the kubelet
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap.
..

This node has joined the cluster:
* Certificate signing request was sent to apiservert and a response wa
s received.
* The Kubelet was informed of the new secure connection details.

Run 'kubectl get nodes' on the control-plane to see this node join th
e cluster.

[vagrant@w2 ~]$
```

```
[vagrant@w3 ~]$ sudo kubeadm join 211.100.2.50:6443 --token cq93h6.7o
g90l3nr71lrxrg --discovery-token-ca-cert-hash sha256:5d3bf5af1d2e00d1
a0fd1a1f970df6b0636a2061c3c870dbafa2c564a1c964eb
[preflight] Running pre-flight checks
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n ku
be-system get cm kubeadm-config -o yaml'
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubel
et/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "
/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Starting the kubelet
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap.
..

This node has joined the cluster:
* Certificate signing request was sent to apiservert and a response wa
s received.
* The Kubelet was informed of the new secure connection details.

Run 'kubectl get nodes' on the control-plane to see this node join th
e cluster.

[vagrant@w3 ~]$
```

```
[vagrant@w1 ~]$ sudo kubeadm join 211.100.2.50:6443 --token cq93h6.7o
g90l3nr71lrxrg --discovery-token-ca-cert-hash sha256:5d3bf5af1d2e00d1
a0fd1a1f970df6b0636a2061c3c870dbafa2c564a1c964eb
[preflight] Running pre-flight checks
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n ku
be-system get cm kubeadm-config -o yaml'
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubel
et/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "
/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Starting the kubelet
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap.
..

This node has joined the cluster:
* Certificate signing request was sent to apiservert and a response wa
s received.
* The Kubelet was informed of the new secure connection details.

Run 'kubectl get nodes' on the control-plane to see this node join th
e cluster.

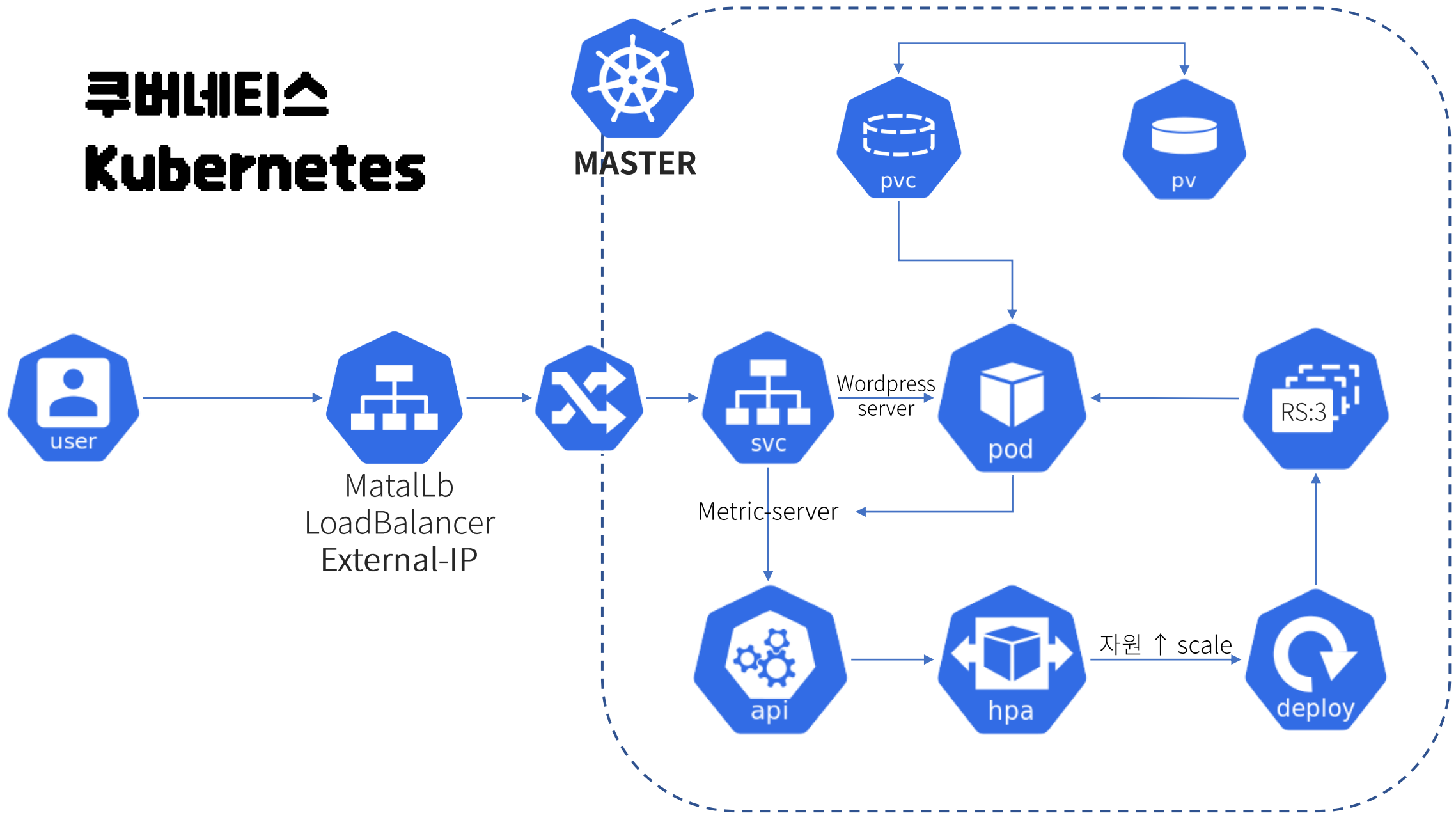
[vagrant@w1 ~]$
```

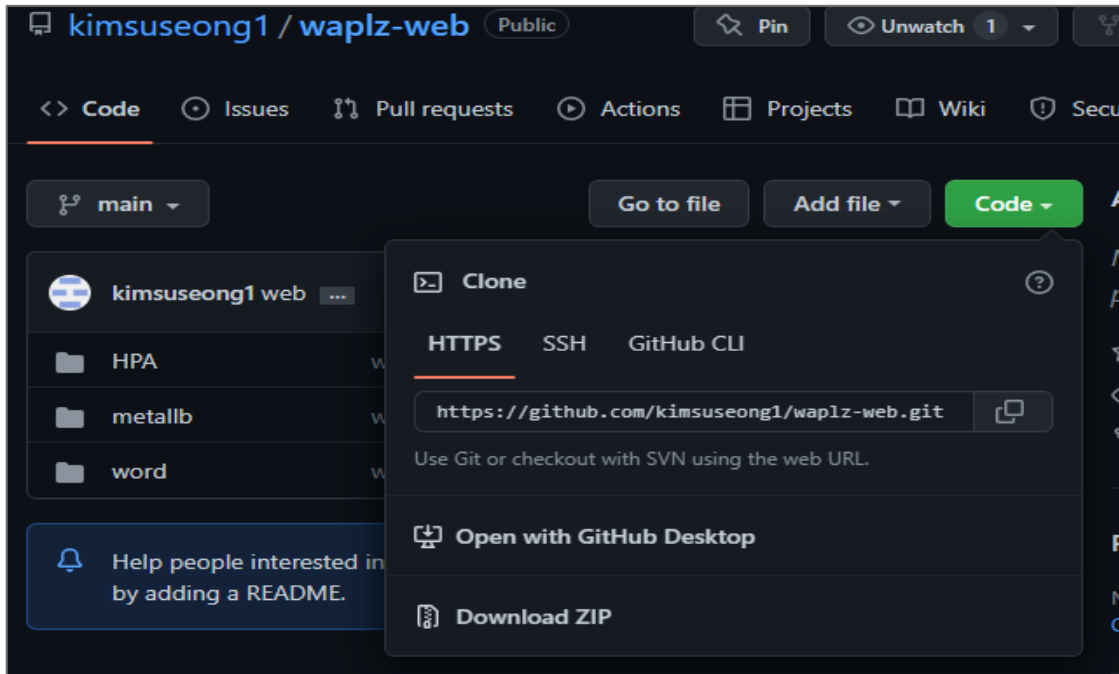
```
[vagrant@m1 ~]$ kubectl get nodes -o wide
NAME STATUS ROLES AGE VERSION INTERNAL-IP
AINER-RUNTIME
m1 Ready control-plane 49m v1.25.3 211.100.2.50
o://1.22.5
w1 Ready <none> 28m v1.25.3 211.100.2.60
o://1.22.5
w2 Ready <none> 28m v1.25.3 211.100.2.61
o://1.22.5
w3 Ready <none> 28m v1.25.3 211.100.2.62
o://1.22.5
[vagrant@m1 ~]$
```

워커 노드 조인

master node에서 kubeadm init
명령어로 받은 토큰과 해시 값으로
worker node들에서 조인

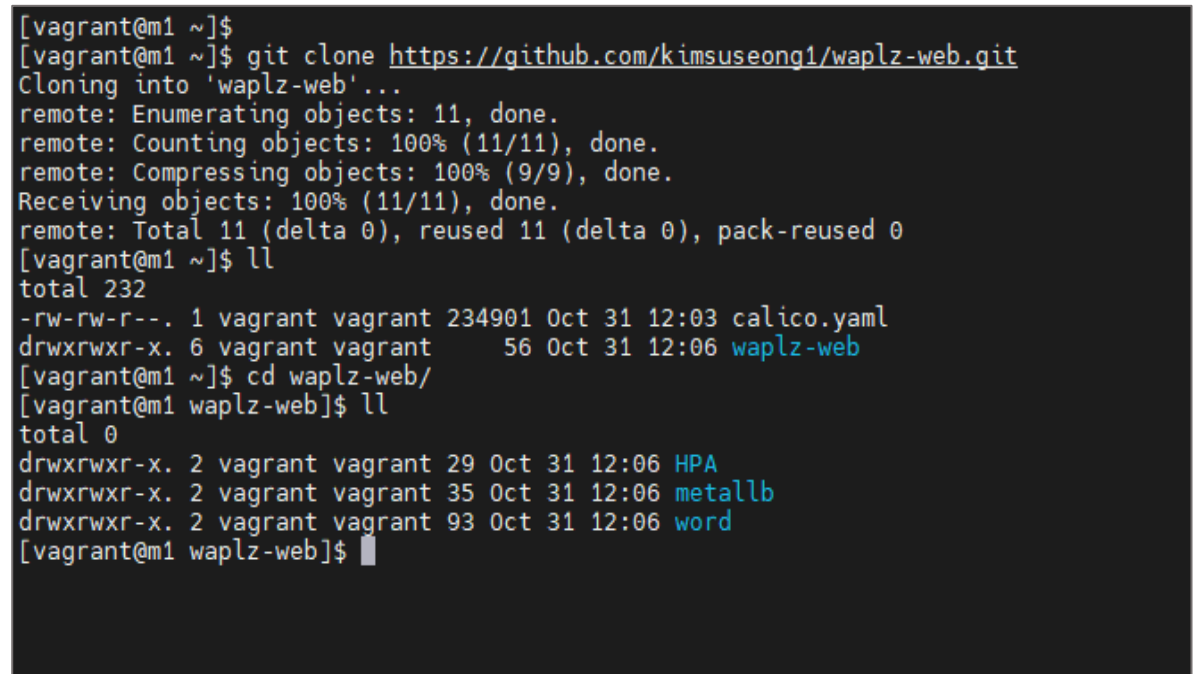
쿠버네티스 Kubernetes





GitHub Repositories

GitHub 에서 미리 Waplz-web 리포지터리에 push 해놓은 (HPA, metallb, word) 파일 HTTPS Code 를 복사



Git Clone

master node 에서 git clone 명령어로 복사한 code로 waplz-web 디렉토리를 복사

```
[vagrant@m1 metallb]$ kubectl apply -f https://raw.githubusercontent.com/metallb/metallb/v0.12.1/manifests/namespace.yaml
namespace/metallb-system created
[vagrant@m1 metallb]$ kubectl apply -f https://raw.githubusercontent.com/metallb/metallb/v0.12.1/manifests/metallb.yaml
serviceaccount/controller created
serviceaccount/speaker created
clusterrole.rbac.authorization.k8s.io/metallb-system:controller created
clusterrole.rbac.authorization.k8s.io/metallb-system:speaker created
role.rbac.authorization.k8s.io/config-watcher created
role.rbac.authorization.k8s.io/pod-lister created
role.rbac.authorization.k8s.io/controller created
clusterrolebinding.rbac.authorization.k8s.io/metallb-system:controller created
clusterrolebinding.rbac.authorization.k8s.io/metallb-system:speaker created
rolebinding.rbac.authorization.k8s.io/config-watcher created
rolebinding.rbac.authorization.k8s.io/pod-lister created
rolebinding.rbac.authorization.k8s.io/controller created
daemonset.apps/speaker created
deployment.apps/controller created
resource mapping not found for name: "controller" namespace: "" from "https://raw.githubusercontent.com/metallb/metallb/v0.12.1/manifests/metallb.yaml": no matches for kind "PodSecurityPolicy" in version "policy/v1beta1"
ensure CRDs are installed first
resource mapping not found for name: "speaker" namespace: "" from "https://raw.githubusercontent.com/metallb/metallb/v0.12.1/manifests/metallb.yaml": no matches for kind "PodSecurityPolicy" in version "policy/v1beta1"
ensure CRDs are installed first
[vagrant@m1 metallb]$
```

온프레미스에서 로드밸런서를 제공하는 metallb, namespace 배포

```
[vagrant@m1 metallb]$ kubectl get pods -n metallb-system -o wide
```

| NAME | READY | STATUS | RESTARTS | AGE | IP | NODE | NOMINATED | NODE | READINESS | GATES |
|-----------------------------|-------|---------|----------|-------|---------------|------|-----------|------|-----------|-------|
| controller-6658b8446c-sqmxj | 1/1 | Running | 0 | 3m54s | 172.16.80.193 | w2 | <none> | | <none> | |
| speaker-48kbt | 1/1 | Running | 0 | 3m54s | 211.100.2.62 | w3 | <none> | | <none> | |
| speaker-gtgjs | 1/1 | Running | 0 | 3m54s | 211.100.2.61 | w2 | <none> | | <none> | |
| speaker-r6plg | 1/1 | Running | 0 | 3m54s | 211.100.2.60 | w1 | <none> | | <none> | |

```
[vagrant@m1 metallb]$
```

```
apiVersion: v1
kind: ConfigMap
metadata:
  namespace: metallb-system
  name: config
data:
  config: |
    address-pools:
    - name: default
      protocol: layer2
      addresses:
      - 211.100.2.50-211.100.2.50
```

~/waplz-web/metallb/metallb-l2config.yaml

```
[vagrant@m1 metallb]$
[vagrant@m1 metallb]$ ll
total 4
-rw-rw-r--. 1 vagrant vagrant 219 Oct 31 12:06 metallb-l2config.yaml
[vagrant@m1 metallb]$ kubectl apply -f metallb-l2config.yaml
configmap/config created
[vagrant@m1 metallb]$ kubectl get configmap -n metallb-system
```

| NAME | DATA | AGE |
|------------------|------|-----|
| config | 1 | 15s |
| kube-root-ca.crt | 1 | 74m |

```
[vagrant@m1 metallb]$
```

MetallB 스피커 확인 및 L2/ARP 경로 생성

- 정해진 작동 방식 L2(ARP)에 따라 경로를 만들 수 있도록 경로를 제공하는 Speaker 동작을 확인합니다.
- MetallB 설정을 적용 하기 위해 오브젝트는 ConfigMap을 사용 하고, L2 네트워크(ARP/NDP)에서 211.100.2.50 고정 대역으로 로드밸런서를 구현 합니다.
- ConfigMap이 생성됐는지 다음 명령으로 확인 합니다.

```

1  apiVersion: apps/v1
2
3  kind: Deployment
4  metadata:
5    name: wordpress
6    labels:
7      app: wordpress
8  spec:
9    replicas: 3
10   selector:
11     matchLabels:
12       app: wordpress
13   template:
14     metadata:
15       labels:
16         app: wordpress
17     spec:
18       containers:
19         - image: wordpress
20           name: wordpress
21           resources:
22             requests:
23               memory: "200Mi"
24               cpu: "50m"
25             limits:
26               memory: "500Mi"
27               cpu: "100m"
28       env:
29         - name: WORDPRESS_DB_HOST
30           value: "211.100.2.51"
31         - name: WORDPRESS_DB_NAME
32           value: "waplz_DB"
33         - name: WORDPRESS_DB_USER
34           value: "admin"
35         - name: WORDPRESS_DB_PASSWORD
36           value: "admin"
37       ports:
38         - containerPort: 80
39           name: wordpress

```

wordpress.yaml

Wordpress 배포를 위한 yaml 코드 분석

- 파드 수를 보장하는 replicas 오브젝트를 제공.
- 파드마다 주어진 부하량을 결정하는 requests, limits 항목에 값을 추가합니다.
- Wordpress에 연결할 DB-Server의 mariadb에서 생성한 DB_NAME, DB_USER, DB_PASSWORD를 입력합니다.

Wordpress 배포를 위한 yaml 코드 분석

```
volumeMounts:
  - name: wordpress-persistent-storage
    mountPath: /var/www/html
volumes:
  - name: wordpress-persistent-storage
    persistentVolumeClaim:
      claimName: wordpress-volumeclaim
```

wordpress.yaml

PVC로 생성된 Persistent-storage 를 /var/www/html 경로를 volum에 mount 합니다.

```
1 kind: PersistentVolumeClaim
2 apiVersion: v1
3 metadata:
4   name: wordpress-volumeclaim
5 spec:
6   accessModes:
7     - ReadWriteOnce
8   resources:
9     requests:
10
11     storage: 10Gi
```

PVC.yaml

지속적으로 사용 가능한 볼륨을 요청 하고 준비된 볼륨에서 10G 공간을 할당합니다.

```
1 kind: PersistentVolume
2 apiVersion: v1
3 metadata:
4   name: pv0001
5   labels:
6     type: local
7 spec:
8   capacity:
9     storage: 25Gi
10  accessModes:
11    - ReadWriteOnce
12  hostPath:
13
14    path: "/data001/pv0001"
```

PV.yaml

지속적으로 사용 가능한 볼륨으로 선언 하며 pv0001 공간에 볼륨을 사용할 수 있게 준비합니다.

```
1 ---
2 kind: Service
3 apiVersion: v1
4 metadata:
5   name: wordpress-service
6 spec:
7   type: LoadBalancer
8 selector:
9   app: wordpress
10 ports:
11   - name: http
12     protocol: TCP
13     port: 80
14     targetPort: 80
15
```

Wordpress-service.yaml

MetalLB를 구성했으므로 Wordpress 서비스를 로드밸런서 서비스로 설정 합니다.

파드에서 생성한 내용을 기록하고 보관하는 PV, PVC yaml

```
[vagrant@m1 wordpress]$ ll
total 16
-rw-rw-r--. 1 vagrant vagrant 202 Oct 28 12:49 pvc1.yaml
-rw-rw-r--. 1 vagrant vagrant 210 Oct 28 12:49 wd-svc.yaml
-rw-rw-r--. 1 vagrant vagrant 173 Oct 28 12:49 wordpress-volume.yaml
-rw-rw-r--. 1 vagrant vagrant 1130 Oct 28 13:35 wordpress.yaml
[vagrant@m1 wordpress]$ kubectl apply -f ./
persistentvolume/pv0001 created
service/wordpress-service created
persistentvolumeclaim/wordpress-volumeclaim created
deployment.apps/wordpress created
[vagrant@m1 wordpress]$
```

service

```
[vagrant@m1 word]$ kubectl get svc,pod,pv,pvc -o wide
NAME                                TYPE             CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE      SELECTOR
service/kubernetes                  ClusterIP         10.96.0.1       <none>           443/TCP          4h23m    <none>
service/wordpress-service          LoadBalancer     10.109.34.197   211.100.2.50    80:32315/TCP     25s      app=wordpress
```

pod

```
NAME                                READY    STATUS    RESTARTS   AGE    IP              NODE
wordpress-85bb4855bc-6p2j9         1/1     Running   0          15m    172.16.80.195   w2
wordpress-85bb4855bc-pdmlj         1/1     Running   0          15m    172.16.193.194   w3
wordpress-85bb4855bc-qfjwl         1/1     Running   0          15m    172.16.190.66    w1
[vagrant@m1 word]$
```

pv ,pvc

```
NAME      CAPACITY  ACCESS MODES  RECLAIM POLICY  STATUS  CLAIM
pv0001    25Gi      RWO           Retain          Bound   default/wordpress-volumeclaim
[vagrant@m1 word]$ kubectl get pvc
NAME                                STATUS  VOLUME  CAPACITY  ACCESS MODES  STORAGECLASS  AGE
wordpress-volumeclaim              Bound   pv0001  25Gi      RWO           default        6m2s
```

Wordpress 관련 서비스, 파드 배포

- Kubectl apply -f ./ 로 해당 경로 yaml 파일들을 created 시켜 줍니다.
- LoadBalancer 로 구현 되어 EXTERNAL-IP 는 ConfigMap layer2 동작 방식으로 부여한 IP로 관리 되고 있습니다.
- Replicas 수에 맞게 각 노드들에게 알맞게 배포 되었고 각 노드IP는 지정 해준 172.16.0.0/16 대역으로 할당 되었습니다.
- Pvc가 사용 가능한 볼륨을 요청하여 pv에서 pv0001 공간에 볼륨 선언으로 wordpress-volumeclaim을 할당 받았습니다.



Welcome

Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

Information needed

Please provide the following information. Do not worry, you can always change these settings later.

Site Title

Username

Username can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.

Password [Hide](#)

Strong

Important: You will need this password to log in. Please store it in a secure location.

Your Email

Double-check your email address before continuing.

Search engine visibility ☐ Discourage search engines from indexing this site

It is up to search engines to honor this request.

[Install WordPress](#)



Welcome

Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

Information needed

Please provide the following information. Do not worry, you can always change these settings later.

Site Title

Username

Username can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.

Password [Hide](#)

Very weak

Important: You will need this password to log in. Please store it in a secure location.

Confirm Password ☒ Confirm use of weak password

Your Email

Double-check your email address before continuing.


Search engine visibility ☐ Discourage search engines from indexing this site

It is up to search engines to honor this request.

[Install WordPress](#)

Wordpress 접속

EXTERNAL-IP로 접속 한 다음 성공적으로 DB정보를 받았으면
위와 같이 임의로 관리해줄 관리자 정보를 적고 install wordpress를 눌러
줍니다.



You are now logged out.

Username or Email Address

waplzadmin

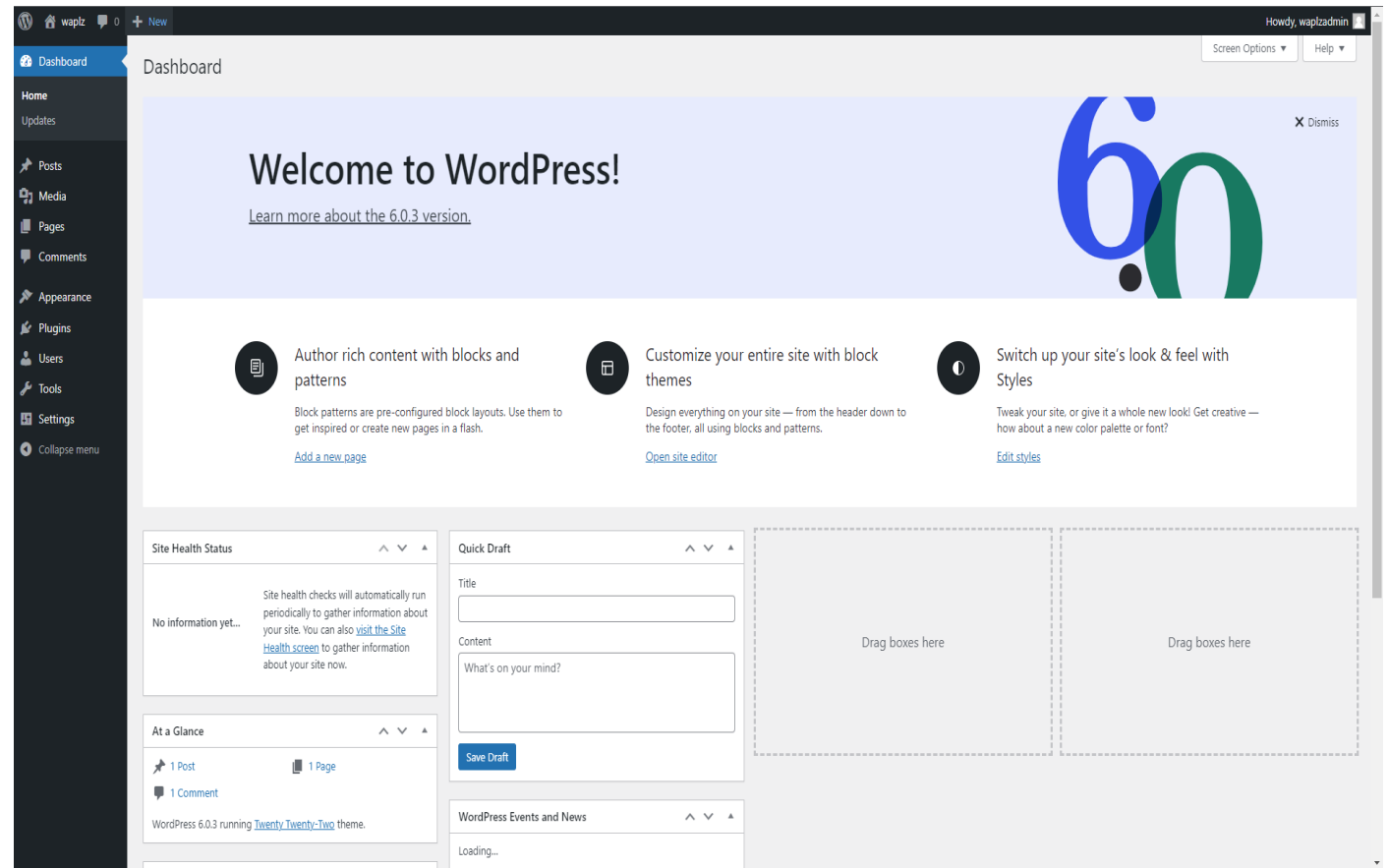
Password

☐ Remember Me

Log In

Lost your password?

← Go to waplz



Dashboard

Welcome to WordPress!

[Learn more about the 6.0.3 version.](#)

Author rich content with blocks and patterns

Customize your entire site with block themes

Switch up your site's look & feel with Styles

Site Health Status

Quick Draft

At a Glance

WordPress Events and News

Wordpress 접속

임의로 생성해준 관리자 정보를 입력 후
성공적으로 구축된 사이트를 확인 합니다.

```

132 spec:
133   containers:
134   - args:
135     - --cert-dir=/tmp
136     - --secure-port=4443
137     - --kubelet-preferred-address-types=InternalIP,ExternalIP,Hostname
138     - --kubelet-insecure-tls
139     - --kubelet-use-node-status-port
140     - --metric-resolution=15s
141   image: k8s.gcr.io/metrics-server/metrics-server:v0.6.1
142   imagePullPolicy: IfNotPresent
143   livenessProbe:

```

Metric-server CA 인증서

프로덕션 환경에서는 CA 인증서를 신뢰하는 과정을 수행해야 하지만 현재는 테스트 과정이므로 CA 인증서를 확인하지 않도록 추가 해줍니다.

```

[vagrant@m1 ~]$
[vagrant@m1 ~]$ cd HPA/
[vagrant@m1 HPA]$ ll
total 8
-rw-rw-r--. 1 vagrant vagrant 4214 Oct 28 14:03 components.yaml
[vagrant@m1 HPA]$ clear
[vagrant@m1 HPA]$
[vagrant@m1 HPA]$ ll
total 8
-rw-rw-r--. 1 vagrant vagrant 4214 Oct 28 14:03 components.yaml
[vagrant@m1 HPA]$ kubectl apply -f components.yaml
serviceaccount/metrics-server created
clusterrole.rbac.authorization.k8s.io/system:aggregated-metrics-reader created
clusterrole.rbac.authorization.k8s.io/system:metrics-server created
rolebinding.rbac.authorization.k8s.io/metrics-server-auth-reader created
clusterrolebinding.rbac.authorization.k8s.io/metrics-server:system:auth-delegator created
clusterrolebinding.rbac.authorization.k8s.io/system:metrics-server created
service/metrics-server created
deployment.apps/metrics-server created
apiservice.apiregistration.k8s.io/v1beta1.metrics.k8s.io created

```

Metric-server 배포

미리 받아온 HPA 디렉토리에 파라미터 값을 추가 해준 components.yaml 을 created 시켜 줍니다. 그리고 잘 배포 되었는지 명령어로 확인 합니다.

```

[vagrant@m1 HPA]$
[vagrant@m1 HPA]$
[vagrant@m1 HPA]$ kubectl get svc --all-namespaces

```

| NAMESPACE | NAME | TYPE | CLUSTER-IP |
|-------------|-------------------|--------------|----------------|
| default | kubernetes | ClusterIP | 10.96.0.1 |
| default | wordpress-service | LoadBalancer | 10.100.140.162 |
| kube-system | kube-dns | ClusterIP | 10.96.0.10 |
| kube-system | metrics-server | ClusterIP | 10.102.156.80 |

```
17 spec:
18   containers:
19     - image: wordpress
20       name: wordpress
21       resources:
22         requests:
23           memory: "500Mi"
24           cpu: "100m"
25         limits:
26           memory: "1024Mi"
27           cpu: "300m"
```

wordpress.yaml

Wordpress.yaml에 requests, limits 항목과 CPU, Memory 값은 파드마다 주어진 부하량을 결정하는 기준이 됩니다.

```
[vagrant@m1 HPA]$
[vagrant@m1 HPA]$ kubectl autoscale deployment wordpress --min=1 --max=9 --cpu-percent=50
horizontalpodautoscaler.autoscaling/wordpress autoscaled
[vagrant@m1 HPA]$
```

Autoscale

Autoscale을 설정해 특정 조건이 만족되는 경우에 자동으로 scale 명령이 수행되도록 min(최소 파드수) max(최대 파드 수) 이고 CPU 사용량이 50% 넘게 되면 autoscale하겠다는 뜻입니다.

```
Windows PowerShell
PS C:\Users\4GL>
PS C:\Users\4GL> $i=0; while($true)
>> {
>>   % { $i++; write-host -NoNewLine "$i $_" }
>>   (Invoke-RestMethod "http://211.100.2.50")-replace '#n', " "
>> }
```

Wordpress 부하를 주는 명령어

HPA를 테스트하기 위해 왼쪽에 있는 파워셸 창에서 반복문을 실행 합니다.

```
Every 2.0s: kubectl top pods m1: Sun Oct 30 15:44:23 2022
NAME CPU(cores) MEMORY(bytes)
wordpress-85bb4855bc-f6m6v 1m 13Mi
wordpress-85bb4855bc-h56s6 1m 13Mi
wordpress-85bb4855bc-l8vj5 100m 50Mi
wordpress-85bb4855bc-rjj9h 1m 13Mi
wordpress-85bb4855bc-s18z9 101m 50Mi
wordpress-85bb4855bc-zksxp 61m 55Mi

Every 2.0s: kubectl get pods m1: Sun Oct 30 15:44:23 2022
NAME READY STATUS RESTARTS AGE
wordpress-85bb4855bc-f6m6v 1/1 Running 0 97s
wordpress-85bb4855bc-h56s6 1/1 Running 0 5m53s
wordpress-85bb4855bc-l8vj5 1/1 Running 0 8m29s
wordpress-85bb4855bc-rjj9h 1/1 Running 0 5m53s
wordpress-85bb4855bc-rz66w 0/1 Pending 0 97s
wordpress-85bb4855bc-s2pvl 0/1 Pending 0 97s
wordpress-85bb4855bc-s18z9 1/1 Running 0 8m29s
wordpress-85bb4855bc-zksxp 1/1 Running 0 8m29s
```

```

      wp-block-column"></div> </div>
    </li></ul> </main> <footer class
    top:var(--wp--custom--spacing--lar
  onwide" style="padding-top:4rem;pad
  'home" aria-current="page">waplz</
  href="https://wordpress.org" rel="n
    </div>
  reen-reader-text {
  path: inset(50%);
    padding: 0;
    word-wrap: normal !impor
    background-color: #eee;
    color: #444;
    left: 5px;
  decoration: none;
  } </style>
ector( 'main' ),
    // Early exit if a skip-
  n;
  it, sibling = docum
  ment was not found.
  target's ID, and generate one if it
  ! skipLinkTargetID ) {
    t.id = skipLinkTargetID;
    ment( 'a' ); skipLink
    skipLinkTargetID;
    sibling.parentElement.in
```

HPA(Horizontal Pod Autoscaler) 테스트

master node 창을 두개 띄운 후,
그 다음 watch kubectl top pods, get
pods 실행 후 2초에 한 번씩 자동으로 상
태를 확인 합니다.

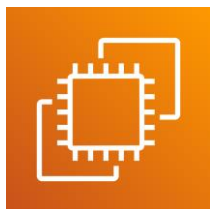
Cloud

AWS EKS를 활용한 Wordpress 구축 및 배포



VPC

격리된 클라우드
리소스



EC2

클라우드의 가상 서버



EKS

Kubernetes를 시작,
실행 및 조정하는 가장
신뢰할 수 있는 방법



IAM

AWS 리소스에 대한
엑세스 관리



RDS

관리되는 관계형
데이터 베이스 서비스



VPC : 10.0.0.0/16
bastion : 10.0.0.0/24
EKS-01 : 10.0.1.0/24
EKS-02 : 10.0.2.0/24

| <input type="checkbox"/> | Name | VPC ID | 상태 | IPv4 CIDR |
|-------------------------------------|---------|-----------------------|-----------|-------------|
| <input checked="" type="checkbox"/> | EKS-VPC | vpc-0829f9ebdff4e99be | Available | 10.0.0.0/16 |

서브넷 (3) 정보

Q 서브넷 필터링

서브넷 ID: subnet-00d09d954e0a74731 X

서브넷 ID: subnet-016ce8119d6ea04a7 X

서브넷 ID: subnet-0fc7b873e6716b95f X

필터 지우기

| <input type="checkbox"/> | Name | 서브넷 ID | 상태 | VPC | IPv4 CIDR |
|--------------------------|---------|--------------------------|-----------|--------------------------------|-------------|
| <input type="checkbox"/> | Bastion | subnet-00d09d954e0a74731 | Available | vpc-0829f9ebdff4e99be EKS... | 10.0.0.0/24 |
| <input type="checkbox"/> | EKS-01 | subnet-016ce8119d6ea04a7 | Available | vpc-0829f9ebdff4e99be EKS... | 10.0.1.0/24 |
| <input type="checkbox"/> | EKS-02 | subnet-0fc7b873e6716b95f | Available | vpc-0829f9ebdff4e99be EKS... | 10.0.2.0/24 |

EKS Cluster를 구축할 네트워크를 구성하기 위해 VPC와 Subnet을 지정한 CIDR로 만듭니다.

라우팅 테이블 (4) 정보

Q 라우팅 테이블 필터링

| <input type="checkbox"/> | Name ▼ | 라우팅 테이블 ID ▼ | 명시적 서브넷 연결 | 엣지 연결 | 기본 ▼ | VPC ▼ |
|--------------------------|------------|-----------------------|-------------------------|-------|------|-------------------------------|
| <input type="checkbox"/> | EKS-RT | rtb-0a6613ac2bf33ea53 | 2 서브넷 | - | 예 | vpc-090662c8988b6d27a EK... |
| <input type="checkbox"/> | Bastion_RT | rtb-04b631c33c9368327 | subnet-07ae1cdb6dcd9... | - | 아니요 | vpc-090662c8988b6d27a EK... |

rtb-0c4ed820e6892a68c / Bastion-RT

세부 정보 | 라우팅 | 서브넷 연결 | 엣지 연결 | 라우팅 전파 | 태그

라우팅 (2)

Q 라우팅 필터링

모두 ▼

| 대상 ▼ | 대상 ▼ | 상태 |
|-------------|-----------------------|-------|
| 0.0.0.0/0 | igw-02bc8c610f0fa2251 | ✔ 활성화 |
| 10.0.0.0/16 | local | ✔ 활성화 |

라우팅 테이블은 Public Subnet 이며 외부로 향하는 트래픽은 Internet Gateway를 통해 밖에 나갑니다.

sg-0e3bc4be5809a59df - Bastion_SG

세부 정보

보안 그룹 이름
Bastion_SG

보안 그룹 ID
sg-0e3bc4be5809a59df

설명
bastion ssh

VPC ID
vpc-090662c8988b6d27a

소유자
547576513274

인바운드 규칙 수
1 권한 항목

아웃바운드 규칙 수
1 권한 항목

인바운드 규칙

아웃바운드 규칙

태그

이제 Reachability Analyzer를 사용하여 네트워크 연결을 확인할 수 있습니다.

Reachability A

인바운드 규칙 (1/1)



태그 관리

보안 그룹 규칙 필터


| <input checked="" type="checkbox"/> | Name ▼ | 보안 그룹 규칙 ID ▼ | IP 버전 ▼ | 유형 ▼ | 프로토콜 ▼ | 포트 범위 ▼ | 소스 ▼ |
|-------------------------------------|--------|------------------------|---------|------|--------|---------|-------------------|
| <input checked="" type="checkbox"/> | - | sgr-036ce0b77db4008... | IPv4 | SSH | TCP | 22 | 203.233.95.154/32 |

Bastion 보안 그룹을 생성 하고 인바운드 규칙은 SSH 22번 포트에 내IP로 접근 할 수 있게 설정 합니다.


sg-0599f8886e186d1ef - Bastion-EKS_SG

세부 정보


보안 그룹 이름

 Bastion-EKS_SG

보안 그룹 ID

 sg-0599f8886e186d1ef

설명

 Bastion to EKS

VPC ID

 vpc-090662c8988b6d27a

소유자

 547576513274

인바운드 규칙 수

4 권한 항목


아웃바운드 규칙 수

1 권한 항목

인바운드 규칙

아웃바운드 규칙

태그

 이제 Reachability Analyzer를 사용하여 네트워크 연결을 확인할 수 있습니다.

Reachability An

인바운드 규칙 (4)



태그 관리

인

 보안 그룹 규칙 필터

| <input type="checkbox"/> | Name ▾ | 보안 그룹 규칙 ID ▾ | IP 버전 ▾ | 유형 ▾ | 프로토콜 ▾ | 포트 범위 ▾ | 소스 ▾ |
|--------------------------|--------|-----------------------|---------|------------|--------|---------|-----------------------|
| <input type="checkbox"/> | - | sgr-02fe71640cfb30320 | - | HTTPS | TCP | 443 | sg-0e3bc4be5809a59... |
| <input type="checkbox"/> | - | sgr-0a8e8f21c5c089f17 | - | DNS (TCP) | TCP | 53 | sg-0e3bc4be5809a59... |
| <input type="checkbox"/> | - | sgr-0ca83bc54d3fe18ff | - | DNS (UDP) | UDP | 53 | sg-0e3bc4be5809a59... |
| <input type="checkbox"/> | - | sgr-06700f50dcfaa698d | - | 사용자 지정 TCP | TCP | 10250 | sg-0e3bc4be5809a59... |

EKS 보안 그룹 생성 하고 인바운드 규칙은 443/TCP, 10250/TCP, 53/TCP, 53/UDP에 대한 접근을 허용합니다.
443/TCP는 Kube-api-server 컴포넌트가 동작할 때 사용하는 포트



IAM

eks-worker-role

eks worker role

요약

생성 날짜

November 10, 2022, 11:50 (UTC+09:00)

마지막 활동

없음

ARN

arn:aws:iam::547576513274:role/eks-worker-role

인스턴

arn

최대 세션 지속 시간

1시간

권한

신뢰 관계

태그

액세스 관리자

세션 취소

권한 정책 (6) 정보

최대 10개의 관리형 정책을 연결할 수 있습니다.

🔍 속성 또는 정책 이름을 기준으로 정책을 필터링하고 Enter를 누릅니다.



정책 이름 ↗



유형



AmazonEKSWorkerNodePolicy

AWS 관리형



AmazonEC2ContainerRegistryReadOnly

AWS 관리형



CloudWatchLogsFullAccess

AWS 관리형



AmazonElasticFileSystemFullAccess

AWS 관리형



AmazonEKS_CNI_Policy

AWS 관리형



AmazonRoute53FullAccess

AWS 관리형

Worker node role에 대한 권한 정책을 적용 합니다.

eks-cluster-role

eks cluster role

요약

생성 날짜

November 10, 2022, 12:01 (UTC+09:00)

마지막 활동

없음

ARN

arn:aws:iam::547576513274:role/eks-cluster-role

최대 세션 지속 시간

1시간

권한

신뢰 관계

태그

액세스 관리자

세션 취소

권한 정책 (2) 정보

최대 10개의 관리형 정책을 연결할 수 있습니다.

Q 속성 또는 정책 이름을 기준으로 정책을 필터링하고 Enter를 누릅니다.



정책 이름



유형



AmazonEKSClusterPolicy

AWS 관리형



AmazonEKSServicePolicy

AWS 관리형

EKS Cluster role에 대한 권한 정책을 적용 합니다.

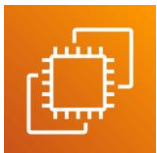
역할 (4) 정보

IAM 역할은 단기간 동안 유효한 자격 증명을 가진 특정 권한이 있는 자격 증명입니다. 신뢰할 수 있는 개체가 역할을 맡을 수 있습니다.

🔍 검색

| <input type="checkbox"/> | 역할 이름 | ▼ | 신뢰할 수 있는 개체 |
|--------------------------|------------------|---|--------------|
| <input type="checkbox"/> | eks-worker-role | | AWS 서비스: ec2 |
| <input type="checkbox"/> | eks-cluster-role | | AWS 서비스: eks |

IAM 역할 생성 완료 후 확인 합니다.



EC2

▼ 요약

인스턴스 개수 정보

1

소프트웨어 이미지(AMI)

Amazon Linux 2 Kernel 5.10 AMI...[더 보기](#)
ami-09cf633fe86e51bf0

가상 서버 유형(인스턴스 유형)

t2.micro

방화벽(보안 그룹)

Bastion_SG

스토리지(볼륨)

1개의 볼륨 - 8GiB

❗ 프리 티어: 첫 해에는 월별 프리 티어 AMI에 대한 t2.micro(또는 t2.micro를 사용할 수 없는 리전의 t3.micro) 인스턴스 사용량 750시간, EBS 스토리지 30GiB, IO 2백만 개, 스냅샷 1GB, 인터넷 대역폭 100GB가 포함됩니다.

취소

인스턴스 시작

EC2 > [인스턴스](#) > i-0fba4ad438b6ac99b > 인스턴스에 연결

인스턴스에 연결 정보

다음 옵션 중 하나를 사용하여 인스턴스 i-0fba4ad438b6ac99b (Bastion EC2)에 연결

EC2 인스턴스 연결

Session Manager

SSH 클라이언트

EC2 직렬 콘솔

인스턴스 ID

i-0fba4ad438b6ac99b (Bastion EC2)

1. SSH 클라이언트를 엽니다.

2. 프라이빗 키 파일을 찾습니다. 이 인스턴스를 시작하는 데 사용되는 키는 project.pem입니다.

3. 필요한 경우 이 명령을 실행하여 키를 공개적으로 볼 수 없도록 합니다.

❏ `chmod 400 project.pem`

4. 퍼블릭 IP을(를) 사용하여 인스턴스에 연결:

❏ 13.124.14.234

예:

❏ `ssh -i "project.pem" ec2-user@13.124.14.234`

❗ 참고: 대부분의 경우 추정된 사용자 이름은 정확합니다. 하지만 AMI 사용 지침을 읽고 AMI 소유자가 기본 AMI 사용자 이름을 변경했는지 확인하십시오.

Bastion-serve로 사용할 EC2 생성 후 SSH 클라이언트에 연결할 정보를 확인 합니다.

```
10/11/2022 12:42.09 /drives/c/Users/4GL/Downloads ssh -i "project.pem" ec2-user@13.124.14.234
Warning: Permanently added '13.124.14.234' (RSA) to the list of known hosts.
X11 forwarding request failed on channel 0

  _ | _ | _ |
  _ | ( _ | _ |
  _ | \ _ | _ |
                Amazon Linux 2 AMI

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-10-0-0-240 ~]$
```

SSH 인바운드 규칙을 허용 한 EC2에 원격 접속 툴을 이용하여 접속 합니다.



RDS

RDS > 데이터베이스 생성

데이터베이스 생성


데이터베이스 생성 방식 선택 정보


☒ 표준 생성
가용성, 보안, 백업 및 유지 관리에 대한 옵션을 포함하여 모든 구성 옵션을 설정합니다.


☐ 손쉬운 생성
권장 모범 사례 구성을 사용합니다. 일부 구성 옵션은 데이터베이스를 생성한 후 변경할 수 있습니다.


엔진 옵션


엔진 유형 정보


☐ Amazon Aurora


☐ MySQL


☒ MariaDB


☐ PostgreSQL


☐ Oracle


☐ Microsoft SQL Server


버전

MariaDB 10.6.10 ▼

CMS에 연결할 RDS를 MariaDB 엔진 옵션으로 선택 합니다.

연결 정보

컴퓨팅 리소스

이 데이터베이스의 컴퓨팅 리소스에 대한 연결을 설정할지를 선택합니다. 연결을 설정하면 컴퓨팅 리소스가 이 데이터베이스에 연결할 수 있도록 연결 설정이 자동으로 변경됩니다.

☒ EC2 컴퓨팅 리소스에 연결 안 함
이 데이터베이스의 컴퓨팅 리소스에 대한 연결을 설정하지 않습니다. 나중에 컴퓨팅 리소스에 대한 연결을 수동으로 설정할 수 있습니다.

☐ EC2 컴퓨팅 리소스에 연결
이 데이터베이스의 EC2 컴퓨팅 리소스에 대한 연결을 설정합니다.

Virtual Private Cloud(VPC) 정보

VPC를 선택합니다. VPC는 이 DB 인스턴스의 가상 네트워킹 환경을 정의합니다.

EKS-VPC (vpc-090662c8988b6d27a)

해당 DB 서브넷 그룹이 있는 VPC만 나열됩니다.

데이터베이스를 생성한 후에는 VPC를 변경할 수 없습니다.

DB 서브넷 그룹 정보

DB 서브넷 그룹을 선택합니다. DB 서브넷 그룹은 선택한 VPC에서 DB 인스턴스가 어떤 서브넷과 IP 범위를 사용할 수 있는지를 정의합니다.

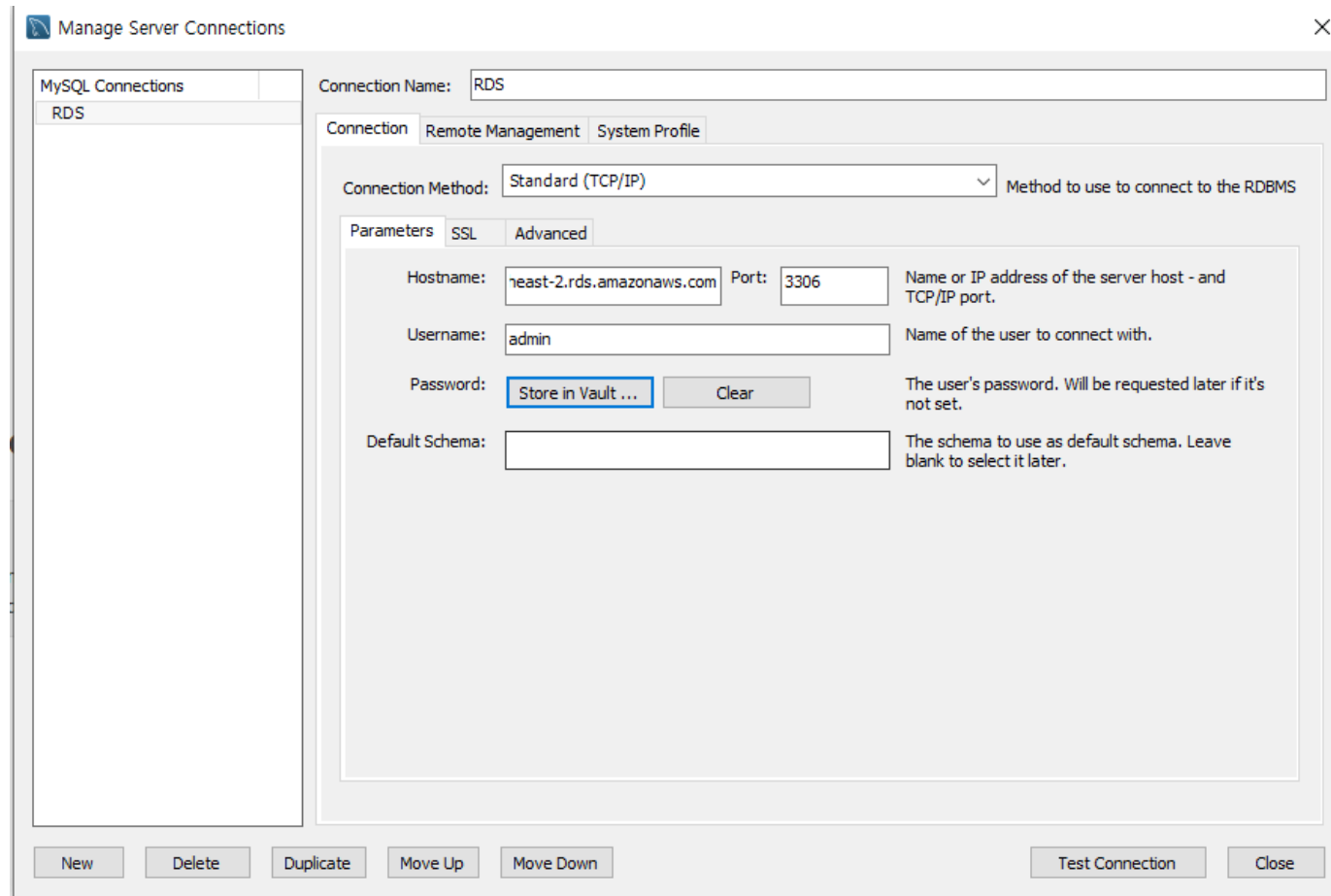
새 DB 서브넷 그룹 생성

퍼블릭 액세스 정보

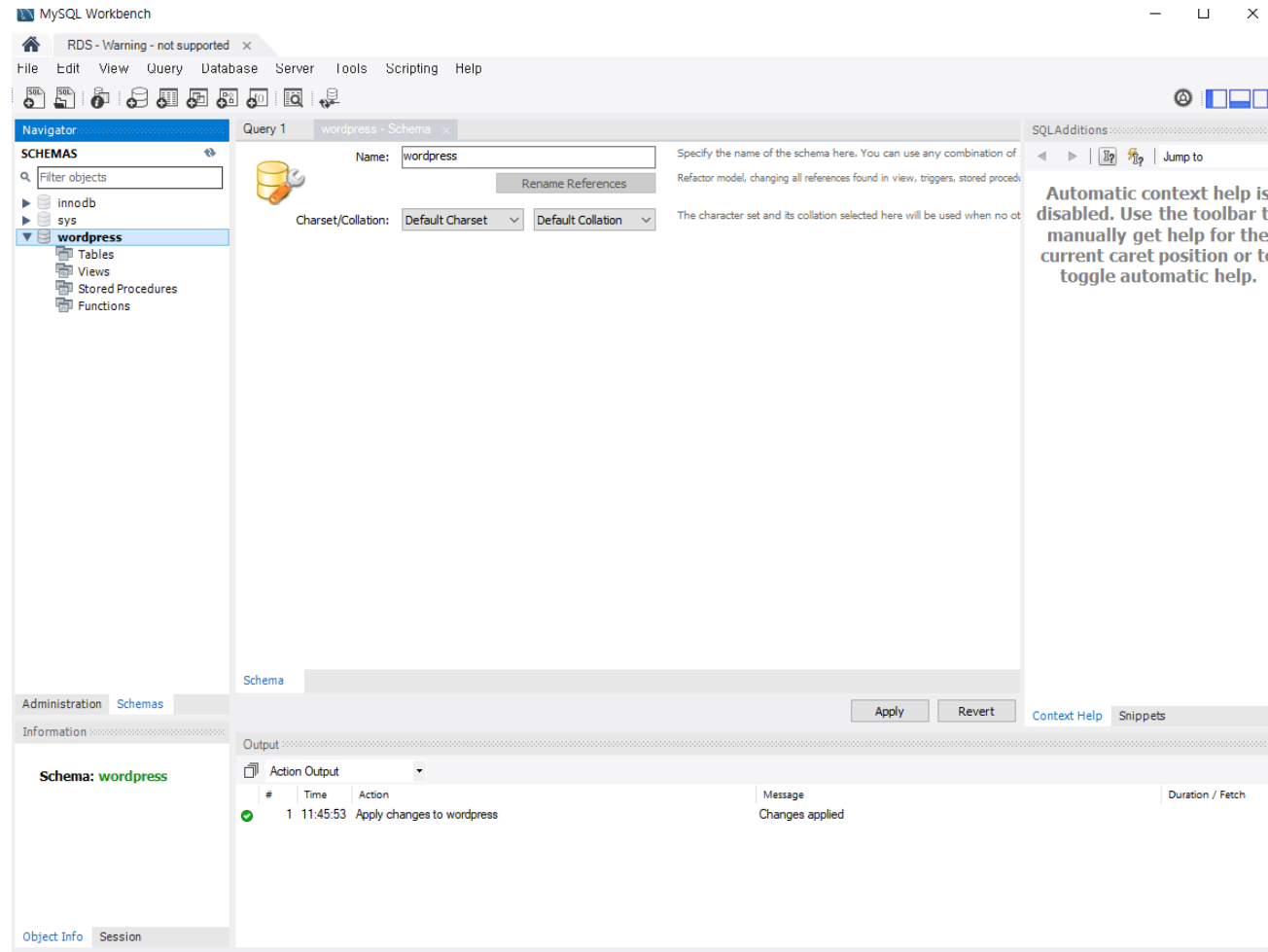
☒ 예
RDS는 데이터베이스에 퍼블릭 IP 주소를 할당합니다. VPC 외부의 Amazon EC2 인스턴스 및 다른 리소스가 데이터베이스에 연결할 수 있습니다. VPC 내부의 리소스도 데이터베이스에 연결할 수 있습니다. 데이터베이스에 연결할 수 있는 리소스를 지정하는 VPC 보안 그룹을 하나 이상 선택합니다.

☐ 아니요
RDS는 퍼블릭 IP 주소를 데이터베이스에 할당하지 않습니다. VPC 내부의 Amazon EC2 인스턴스 및 다른 리소스만 데이터베이스에 연결할 수 있습니다. 데이터베이스에 연결할 수 있는 리소스를 지정하는 VPC 보안 그룹을 하나 이상 선택합니다.

RDS를 EKS-VPC에 연결 후 새 서브넷 그룹을 생성하고, 퍼블릭 액세스를 허용 후 생성 합니다.



Mysql Workbench를 통해 생성한 RDS에 접속 합니다.



Wordpress 사용을 위한 스키마를 생성 합니다.



EKS

클러스터 (1) 정보

🔄

삭제

클러스터 추가 ▼

🔍

이름, 상태, Kubernetes 버전 또는 공급자를 기준으로 클러스터 필터링

<

1

>

| 클러스터 이름 | 상태 | Kubernetes 버전 | 공급자 |
|---|--|---------------|-----|
| <div><div>○</div><div>EKS-Cluster</div></div> | <div><div>⋮</div><div>생성 중</div></div> | 1.21 | EKS |

k8s를 사용하기 위해 EKS Cluster를 생성 합니다.

노드 그룹 구성 정보

노드 그룹은 Amazon EKS 클러스터에 컴퓨팅 용량을 제공하는 EC2 인스턴스의 그룹입니다. 클러스터에는 여러 노드 그룹을 추가할 수 있습니다.

노드 그룹 구성

노드 그룹을 생성한 후에는 이러한 속성을 변경할 수 없습니다.

이름

이 노드 그룹에 대한 고유한 이름을 할당합니다.

노드 그룹 이름은 문자 또는 숫자로 시작해야 하며 유니코드 문자 세트, 숫자, 하이픈 및 밑줄을 포함할 수 있습니다. 최대 길이는 63자입니다.

노드 IAM 역할 정보

노드에서 사용할 IAM 역할을 선택합니다. 새 역할을 생성하려면 [IAM 콘솔](#)(으)로 이동합니다.



관리형 노드 그룹 삭제 시 서비스가 중단될 수 있기 때문에 선택한 역할은 자체 관리형 노드 그룹에서 사용하지 않아야 합니다.

[자세히 알아보기](#)

EKS Node Group에 생성해 두었던 EKS-Worker-Role를 노드 IAM 역할로 설정 합니다.

노드 그룹 컴퓨팅 구성

노드 그룹을 생성한 후에는 이러한 속성을 변경할 수 없습니다.

AMI 유형 [정보](#)

노드에 대한 EKS 최적화 Amazon Machine Image를 선택합니다.

Amazon Linux 2 (AL2_x86_64) ▼

용량 유형

이 노드 그룹에 대한 용량 구매 옵션을 선택합니다.

On-Demand ▼

인스턴스 유형 [정보](#)

이 노드 그룹에 대해 선호하는 인스턴스 유형을 선택합니다.

선택 ▼

t3.medium ✕

vCPU: Up to 2 vCPUs memory: 4.0 GiB

디스크 크기

각 노드에 연결되는 EBS 볼륨의 크기를 선택합니다.

20

GiB

Node Group Computer 구성 입니다.

노드 그룹 조정 구성

원하는 크기

그룹에서 처음에 시작할 노드 수를 설정합니다.

노드

최소 크기

그룹에서 축소할 수 있는 최소 노드 수를 설정합니다.

노드

최대 크기

그룹에서 확장할 수 있는 최대 노드 수를 설정합니다.

노드

Node Group 조정 구성에 노드는 원하는 크기 2개, 최소 크기 2개, 최대 크기 3개로 설정 합니다.

네트워킹 지정

노드 그룹 네트워크 구성

노드 그룹을 생성한 후에는 이러한 속성을 변경할 수 없습니다.

서브넷 [정보](#)

노드가 실행될 VPC의 서브넷을 지정합니다. 새 서브넷을 생성하려면 [VPC 콘솔](#)의 해당 페이지로 이동합니다.

서브넷 선택



subnet-008d6411947a161d0 ✕

subnet-095255affbf95fe10 ✕

☒ 노드에 대한 SSH 액세스 구성 [정보](#)

SSH 키 페어

SSH 키 페어를 선택하여 노드에 대한 보안 원격 액세스를 허용합니다. 새 SSH 키 페어를 생성하려면 [EC2 콘솔](#)의 해당 페이지로 이동합니다.

BstionKey



SSH 원격 액세스 권한 허용 대상

노드에 원격으로 액세스할 수 있는 SSH 클라이언트 소스 IP 범위를 구성합니다.

☒ 선택한 보안 그룹

노드에 원격으로 액세스할 수 있는 소스 IP를 제한하는 보안 그룹을 지정합니다.

☐ 모두

노드에 원격으로 액세스할 수 있는 소스 IP를 제한하지 않습니다.

보안 그룹

새 보안 그룹을 생성하려면 [EC2 콘솔](#)의 해당 페이지로 이동합니다.

보안 그룹 선택



sg-0e2aa775a21ef973d ✕

취소

이전

다음

Node Group에 서브넷과 보안 그룹을 설정 합니다.


```
[ec2-user@ip-10-0-0-144 ~]$ aws configure
AWS Access Key ID [None]: AKIA34D6V0MJIQFYX6X4
AWS Secret Access Key [None]: 7bXrZFjSdvmKHKWf/3YYB80TA7qaZid6PVEXimQD
Default region name [None]: ap-northeast-2
Default output format [None]: json
[ec2-user@ip-10-0-0-144 ~]$
[ec2-user@ip-10-0-0-144 ~]$
[ec2-user@ip-10-0-0-144 ~]$ mkdir -p ~/.kube
[ec2-user@ip-10-0-0-144 ~]$ curl -LO https://dl.k8s.io/release/v1.21.0/bin/linux/amd64/kubec
tl
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100  138    100  138    0     0   526      0  --:--:-- --:--:-- --:--:--   528
100 44.2M  100 44.2M    0     0 25.2M      0  0:00:01  0:00:01 --:--:-- 42.5M
[ec2-user@ip-10-0-0-144 ~]$ chmod +x ./kubectl
[ec2-user@ip-10-0-0-144 ~]$ sudo mv ./kubectl /usr/bin/kubectl
[ec2-user@ip-10-0-0-144 ~]$ kubectl version --short --client
Client Version: v1.21.0
[ec2-user@ip-10-0-0-144 ~]$ aws eks update-kubeconfig --region ap-northeast-2 --name EKS-Clu
ster
Added new context arn:aws:eks:ap-northeast-2:816309433106:cluster/EKS-Cluster to /home/ec2-u
ser/.kube/config
[ec2-user@ip-10-0-0-144 ~]$ kubectl get node
NAME                                                    STATUS    ROLES    AGE     VERSION
ip-10-0-2-187.ap-northeast-2.compute.internal        Ready     <none>    4h29m   v1.21.14-eks-ba743
26
ip-10-0-3-199.ap-northeast-2.compute.internal        Ready     <none>    4h30m   v1.21.14-eks-ba743
26
[ec2-user@ip-10-0-0-144 ~]$ kubectl get pod --all-namespaces
NAMESPACE      NAME                                READY     STATUS    RESTARTS   AGE
kube-system    aws-node-d4ss5                     1/1       Running   0           4h30m
kube-system    aws-node-xdvk4                     1/1       Running   0           4h30m
kube-system    coredns-6dbb778559-5qwl           1/1       Running   0           4h50m
kube-system    coredns-6dbb778559-m8qxg          1/1       Running   0           4h50m
kube-system    kube-proxy-fmcd8                   1/1       Running   0           4h30m
kube-system    kube-proxy-kzfqk                   1/1       Running   0           4h30m
[ec2-user@ip-10-0-0-144 ~]$
```

**configure를 통해 IAM 계정을 연결 후 EKS를 활용하기 위한 도구들을 설치하고
EKS를 통해 설치된 Node들의 연결을 확인 합니다**

```

cpu: 100m
env:
- name: WORDPRESS_DB_HOST
  value: mariadb.co7xwio47rv1.ap-northeast-2.rds.amazonaws.com
- name: WORDPRESS_DB_NAME
  value: wordpress
- name: WORDPRESS_DB_USER
  value: admin
- name: WORDPRESS_DB_PASSWORD
  value: ss100421
ports:
- containerPort: 80
  name: wordpress
volumeMounts:
- name: wordpress-persistent-storage
  mountPath: /var/www/html
volumes:
- name: wordpress-persistent-storage
  persistentVolumeClaim:
    claimName: wordpress-volumeclaim

```

```

-rw-r--r-- 1 root root 202 Nov 11 05:05 pvc1.yaml
-rw-r--r-- 1 root root 210 Nov 11 05:05 wd-svc.yaml
-rw-r--r-- 1 root root 173 Nov 11 05:05 wordpress-volume.yaml
-rw-r--r-- 1 root root 1154 Nov 11 05:21 wordpress.yaml
[ec2-user@ip-10-0-0-144 word]$ kubectl apply -f ./
persistentvolume/pv0001 created
service/wordpress-service created
persistentvolumeclaim/wordpress-volumeclaim created
deployment.apps/wordpress created
[ec2-user@ip-10-0-0-144 word]$

```

Git을 통해 코드 다운로드 후 wordpress.yaml 코드에 RDS에 대한 정보를 기입 후 다운로드 한 디렉토리에 있는 yaml 코드 들을 apply 합니다.

```
[ec2-user@ip-10-0-0-144 word]$ kubectl get pod,svc
```

| NAME | READY | STATUS | RESTARTS | AGE |
|--------------------------------|-------|-------------------|----------|-----|
| pod/wordpress-7f595b49f6-jc2ss | 0/1 | ContainerCreating | 0 | 22s |
| pod/wordpress-7f595b49f6-lf4bm | 0/1 | ContainerCreating | 0 | 22s |
| pod/wordpress-7f595b49f6-tddvq | 0/1 | ContainerCreating | 0 | 22s |

| NAME | TYPE | CLUSTER-IP | EXTERNAL-IP |
|---------------------------|--------------|----------------|--|
| service/kubernetes | ClusterIP | 172.20.0.1 | <none> |
| service/wordpress-service | LoadBalancer | 172.20.108.219 | a2409131bcb664cf9bf89e83b15f3655-1570243123.ap-northeast-2.elb.amazonaws.com |

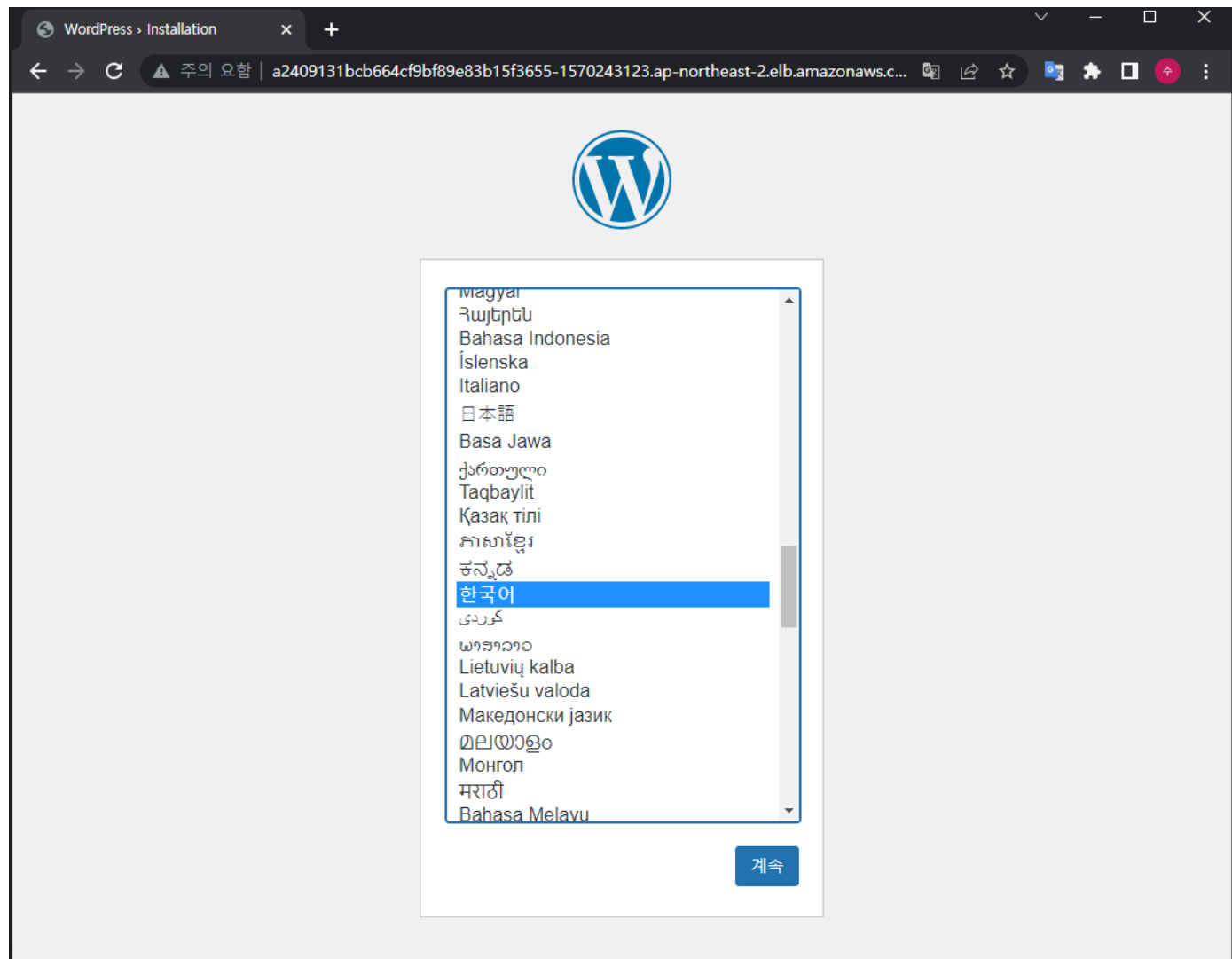
```
[ec2-user@ip-10-0-0-144 word]$ kubectl get pod
```

| NAME | READY | STATUS | RESTARTS | AGE |
|----------------------------|-------|-------------------|----------|-----|
| wordpress-7f595b49f6-jc2ss | 0/1 | ContainerCreating | 0 | 29s |
| wordpress-7f595b49f6-lf4bm | 0/1 | ContainerCreating | 0 | 29s |
| wordpress-7f595b49f6-tddvq | 0/1 | ContainerCreating | 0 | 29s |

```
[ec2-user@ip-10-0-0-144 word]$ kubectl get pod
```

| NAME | READY | STATUS | RESTARTS | AGE |
|----------------------------|-------|---------|----------|-----|
| wordpress-7f595b49f6-jc2ss | 1/1 | Running | 0 | 35s |
| wordpress-7f595b49f6-lf4bm | 1/1 | Running | 0 | 35s |
| wordpress-7f595b49f6-tddvq | 1/1 | Running | 0 | 35s |

배포 된 wordpress pod와 service를 확인 합니다.



EXTERNAL-IP 를 통해 접속하여 정상 가동이 되는지 확인 합니다.



환영합니다

유명한 워드프레스 5분 설치 절차에 오신 것을 환영합니다! 아래의 정보를 입력하기만 하면 세계에서 가장 확장성 있고 강력한 개인 발행 플랫폼을 사용할 수 있습니다.

정보가 필요합니다

다음 정보를 제공해주세요. 걱정하지 마세요. 이 설정을 나중에 언제든지 바꿀 수 있습니다.

사이트 제목

사용자명

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

비밀번호

[숨기기](#)

Very weak

중요: 로그인할 비밀번호가 필요할 것입니다. 안전한 위치에서 저장해주세요.

비밀번호 확인

☒ 약한 비밀번호 사용 확인

이메일 주소

계속하기 전에 이메일 주소를 다시 확인하세요.

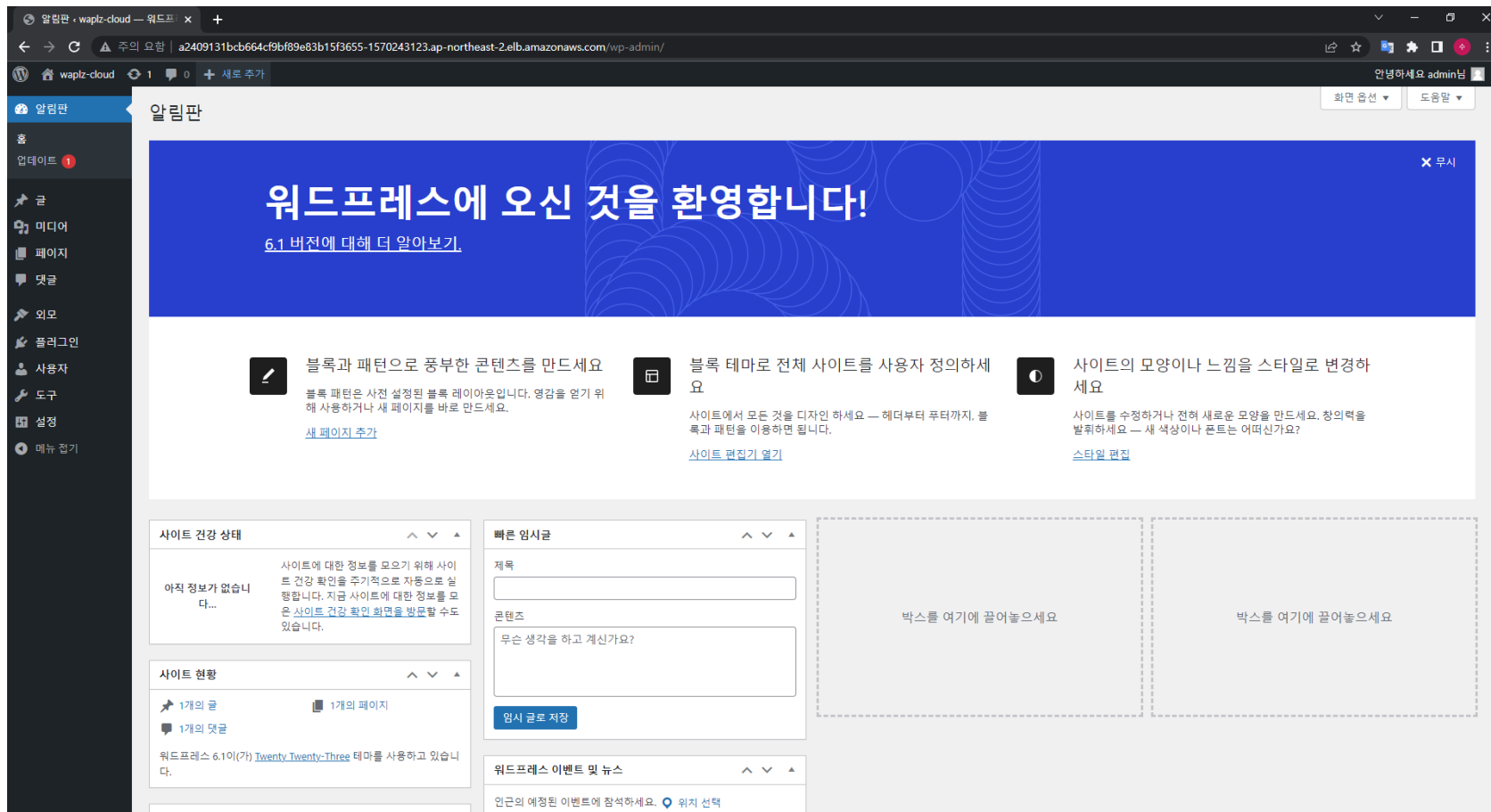
검색 엔진 가시성

☐ 검색 엔진이 이 사이트를 검색하는 것을 차단

이 요청이 받아들여지는 것은 전적으로 검색 엔진에 좌우됩니다.

[워드프레스 설치](#)

웹 페이지 정보 임의로 설정 합니다.



EKS를 활용한 Wordpress 구축 완료.