

쿠버네티스 Concept

MAKE CLOUD NATIVE UBIQUITOUS

CNCF is the open source, vendor-neutral hub of **cloud native computing**, hosting projects like Kubernetes and Prometheus to make cloud native universal and sustainable.

[ABOUT CNCF](#)

212
Projects

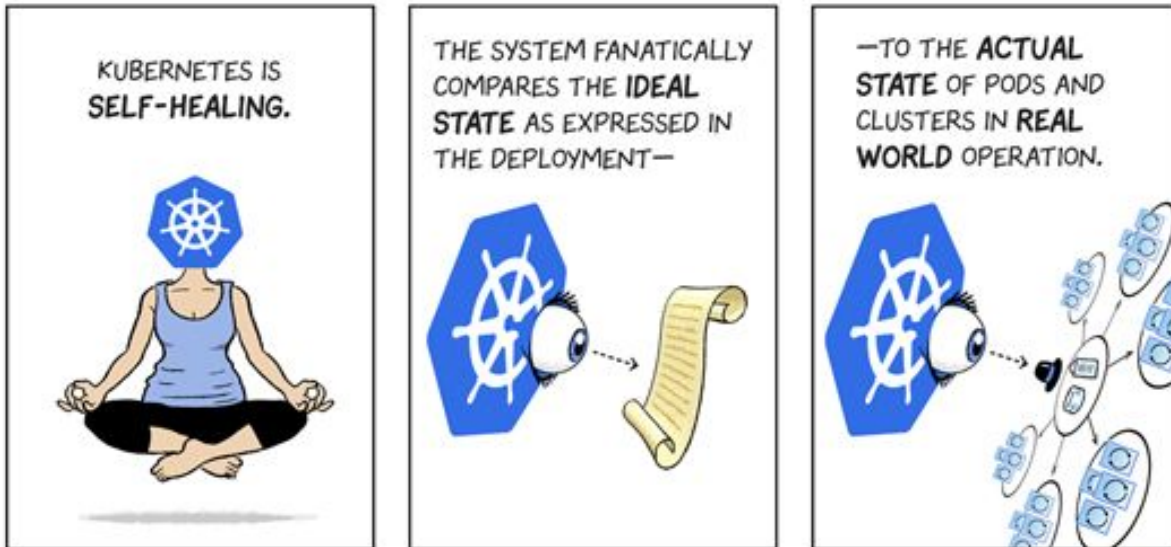
287K
Contributors

20M
Contributions

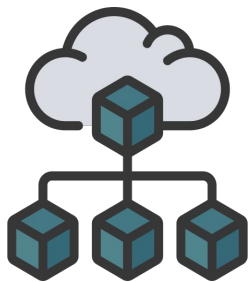
192
Countries

쿠버네티스란 무엇인가?

- 구글 Borg(이후 Omega 로 바뀐 시스템) 내부 시스템을 개발해 2014년 K8s로 공개 (10년간 비밀 유지)
- CNCF에 인큐베이팅 되어 첫번째로 졸업 함
- 쿠버네티스의 핵심기능
 - 애플리케이션 기능에 집중할 수 있도록 서비스를 추상화 함
 - 수천에서 수만개의 컨테이너를 효과적으로 관리 가능
 - 운영팀이 인프라를 효과적으로 관리할 수 있도록 효과적이고 손쉬운 도구 제공
 - 상태확인 및 자가 치유, 오토스케일링, 어플리케이션 개발 단순화 등



K8s를 사용하는 이유



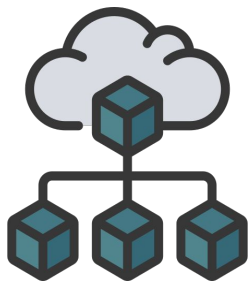
생성



설치 및 설정

Imperative

K8s를 사용하는 이유



생성

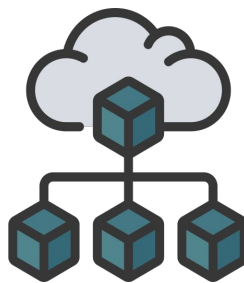


설치 및 설정

Imperative



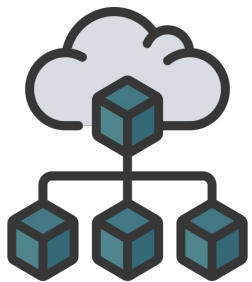
설치 및 설정



생성

Declarative

K8s를 사용하는 이유



생성



설치 및 설정

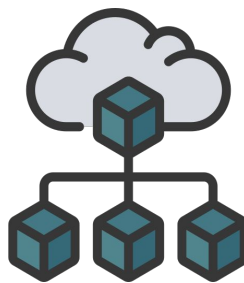
Imperative



변경 수반



설치 및 설정



생성

Declarative



선언에 따른 생성
추상화 가능

Declarative

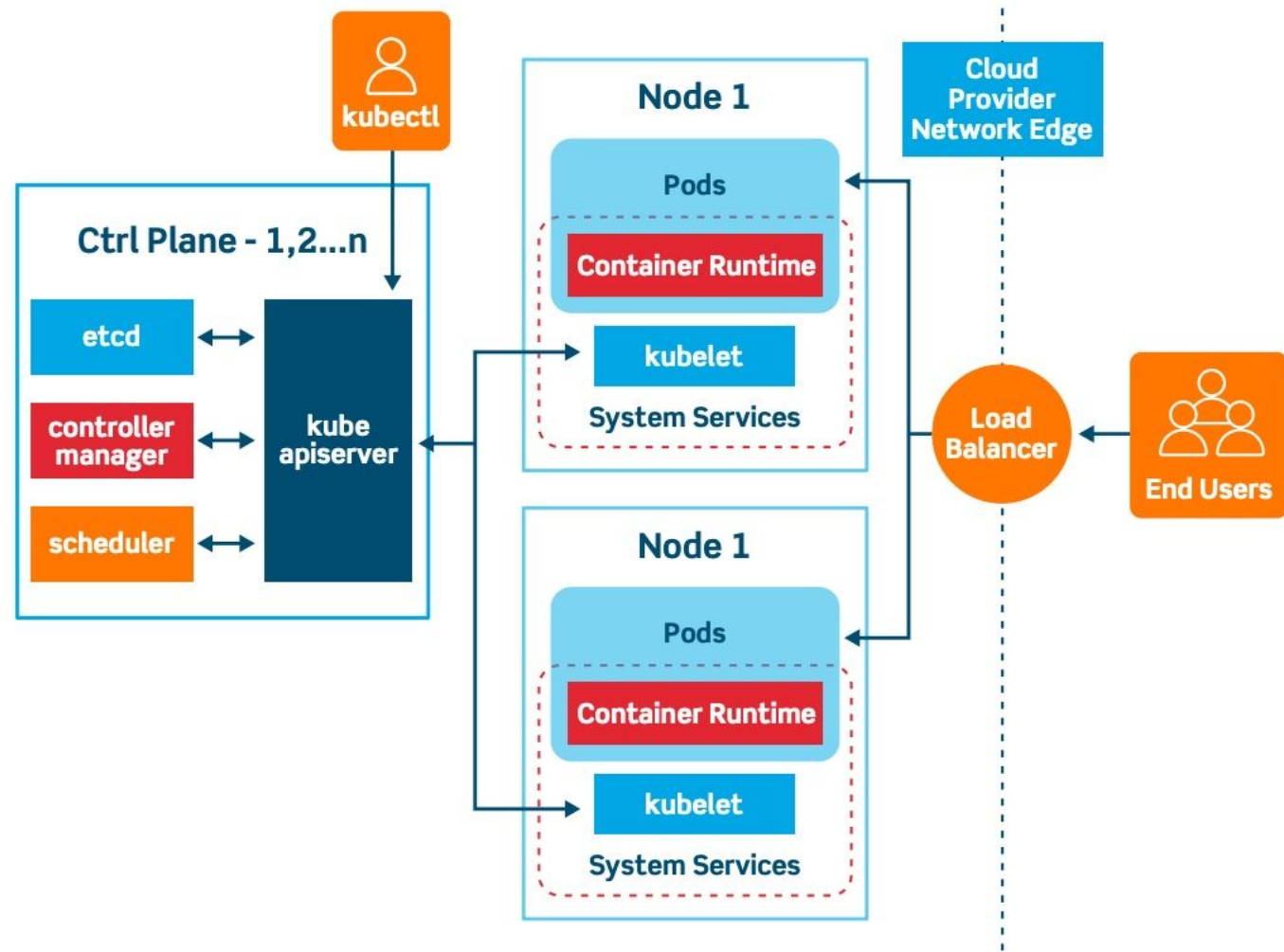
=



Declarative = Exist API

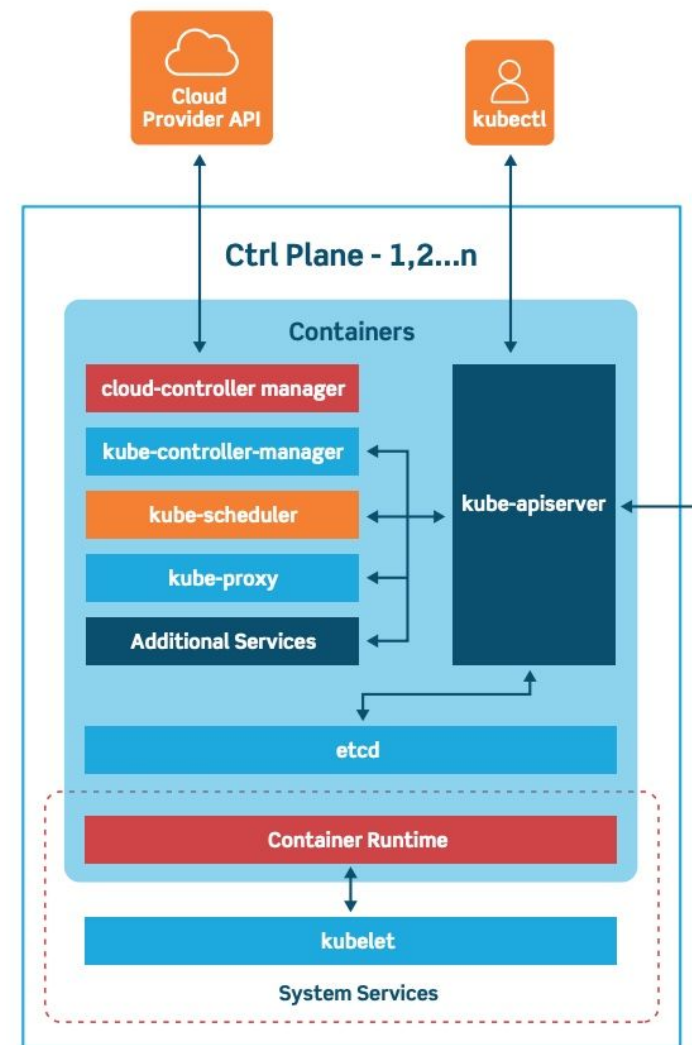
K8s 의 기본 구성 단위

- Control Plane
 - etcd
 - controller manager
 - scheduler
- Nodes
 - kubelet
 - kube-proxy
 - container-runtime



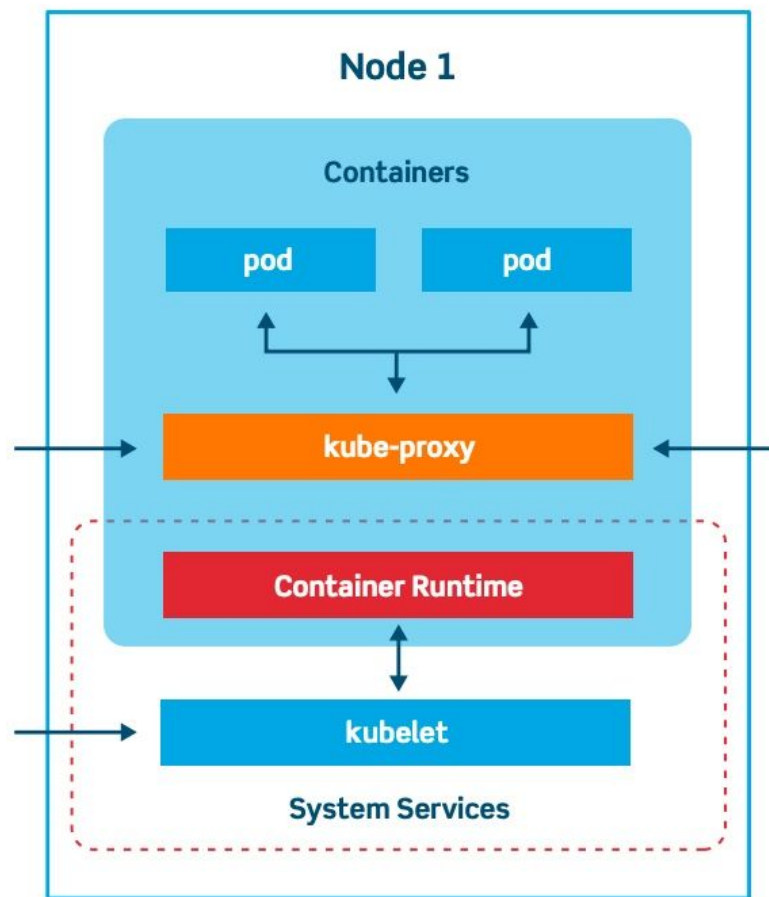
K8s 의 기본 구성 단위 – Contorl Plane(Master)

- 컨트롤 플레인은 쿠버네티스 전체 클러스터를 관리.
- 쿠버네티스 오브젝트의 레코드를 유지 및 관리 (제어루프)
- 컨트롤 플레인의 상세 구성
 - API 서버 : 쿠버네티스 오브젝트에 대한 데이터를 설정하고 검증. REST 서비스 제공
 - 스케줄러 : 클러스터의 노드에 파드를 배포 하는 역할. feasibility 계산 하여 배포
 - 컨트롤러 매니저 : 제어 루프를 이용해 설정된 상태를 지속적으로 유지하도록 함
 - ETCD : 클러스터 오브젝트의 데이터를 저장

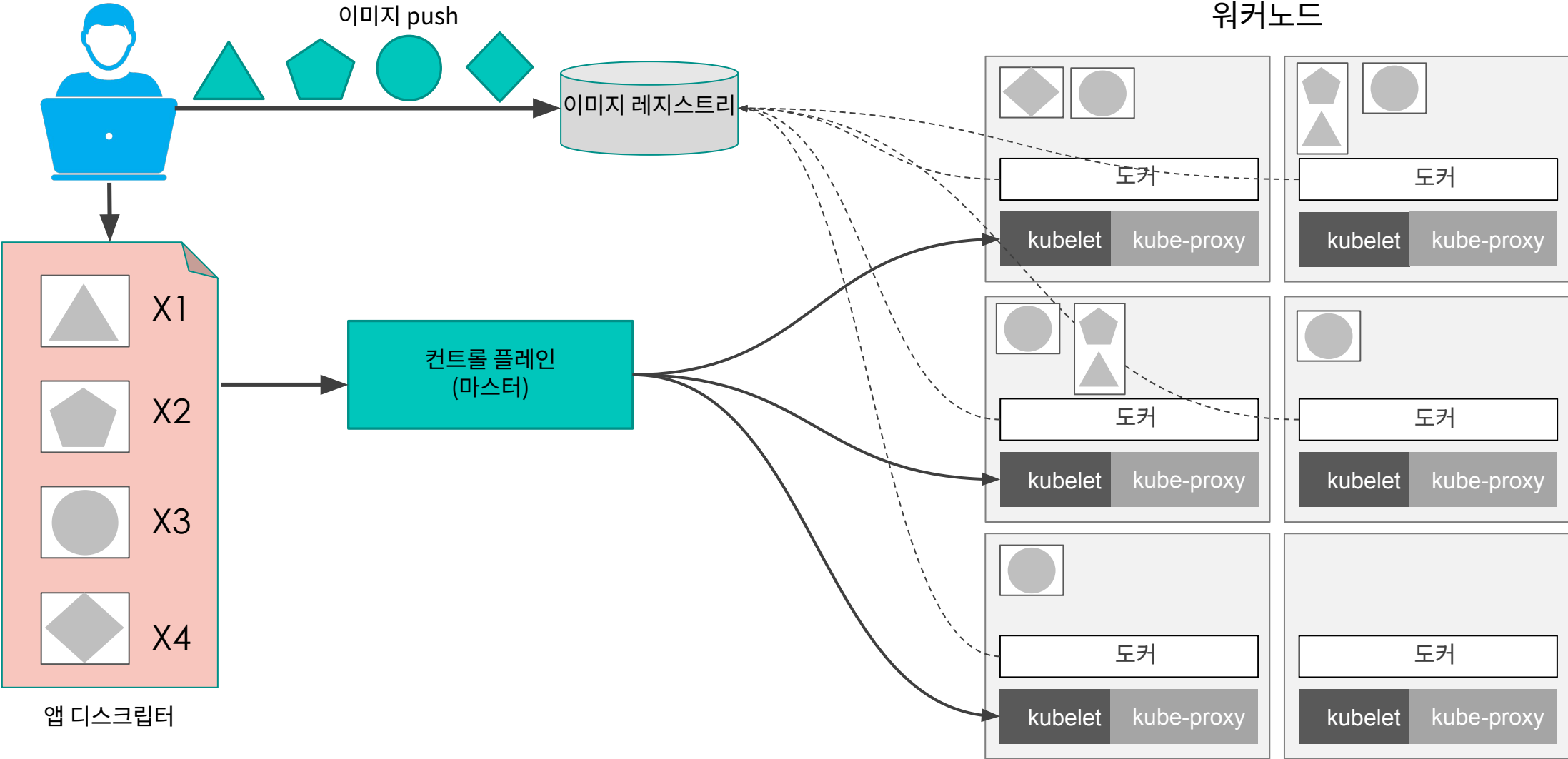


K8s의 기본 구성 단위 - Node(Worker)

- 워커 노드는 컨테이너화된 애플리케이션을 실행하는 시스템
- 애플리케이션을 실행, 모니터링하며 다음의 구성 요소로 이루어짐
 - 컨테이너 런타임 : 컨테이너를 실행하는 엔진 (Docker, rkt 등)
 - Kubelet : API 서버와 통신하며 워커 노드의 컨테이너를 관리
 - kube-proxy : 애플리케이션의 구성 요소간에 네트워크 트래픽을 분산



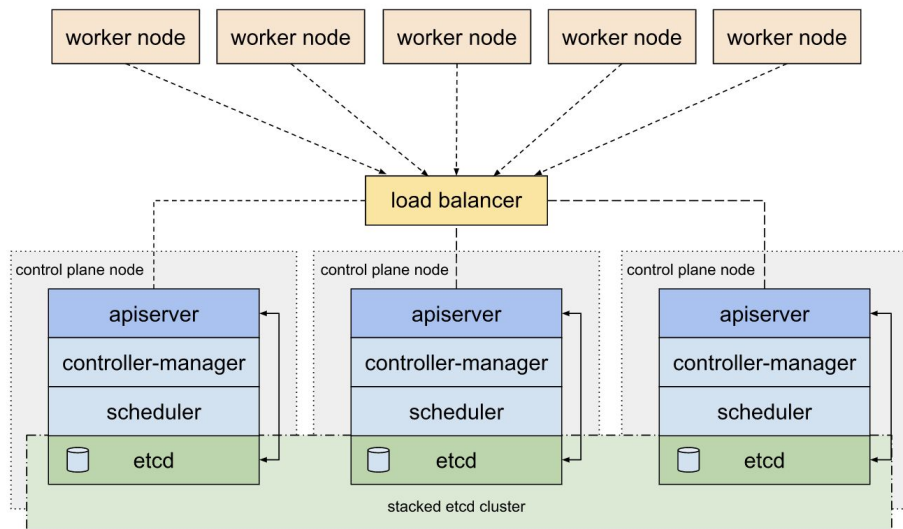
K8s 에서 애플리케이션 실행



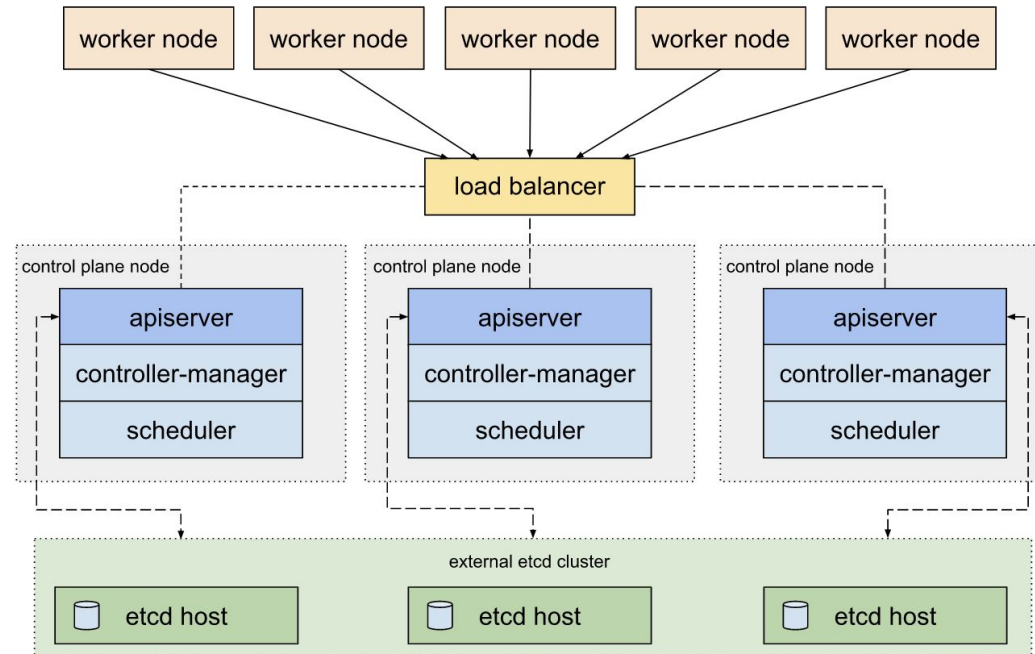
ETCD란

- etcd 는 클러스터 구성으로 데이터를 분산 해서 보관함
- 모든 데이터는 모든 노드에 동일하게 존재
- RAFT 알고리즘으로 리더를 선출 하며, 모든 데이터는 리더에 먼저 쓰여지고 나머지 노드에 복제됨(RAFT 상세 URL)
- ETCD의 고가용성을 위해서는 최소 3대 구성이며, Contol Plane 내부 또는 외부에 구성 가능함

etcd 내부 구성

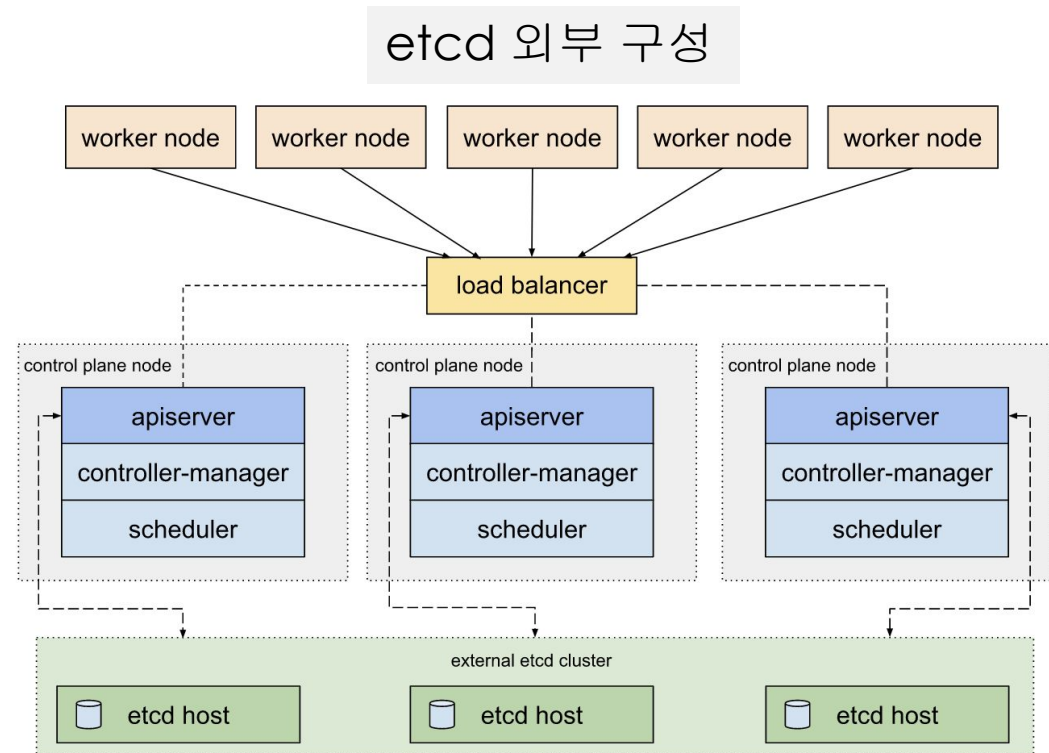
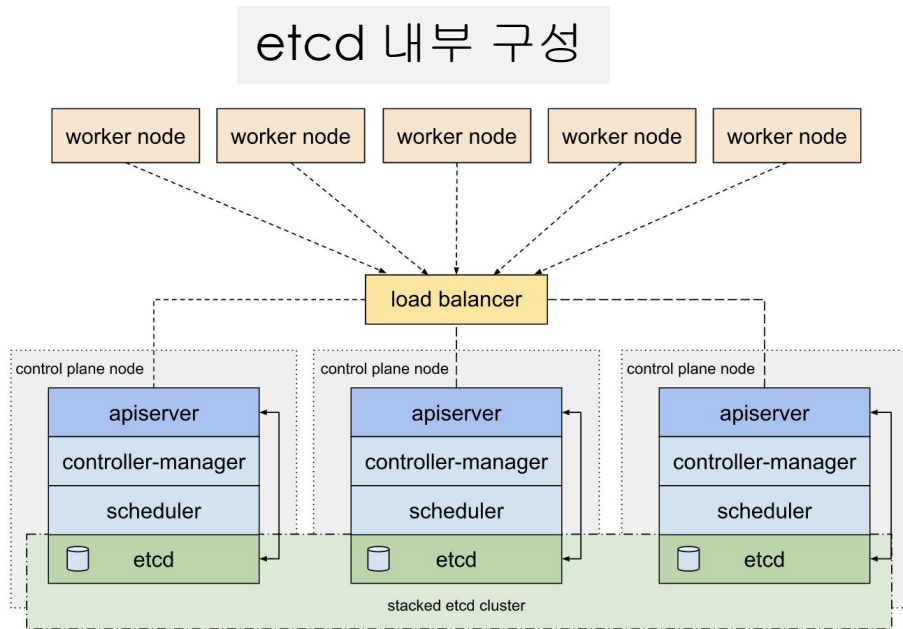


etcd 외부 구성



ETCD고가용성 구성

- etcd 는 클러스터 구성으로 데이터를 분산 해서 보관함
- 모든 데이터는 모든 노드에 동일하게 존재
- RAFT 알고리즘으로 리더를 선출 하며, 모든 데이터는 리더에 먼저 쓰여지고 나머지 노드에 복제됨(RAFT 상세 URL)
- ETCD의고가용성을 위해서는 최소 3대 구성이며, Contol Plane 내부 또는 외부에 구성 가능함



XML, JSON, YAML

XML, JSON, BSON, YAML

- XML은 XSD 를 통한 제약 사항이 강함.
- JSON은 XML 에 비해 제약사항이 매우 적고, 파싱 속도가 빨라 XML의 특정 영역을 대체.
- YAML 의 가장 큰 목적은 사람이 읽을 수 있도록 하는 것

항목	XML	JSON	YAML
Birth	1996	2002	2006
Full Name	eXtensible Markup Language	Java Script Object Notation	Y Ain't Markup Language
가독성	일기 어려움	읽을 만함	읽기 쉬움
파싱속도	느림	빠름	빠름
자료구조	트리	맵	맵

<https://www.json2yaml.com/>

JSON to YAML

Convert JSON to YAML online

YAML vs JSON

determine which format is right for you

JSON

- stands for javascript object notation
- records separated by commas
- keys & strings wrapped by double quotes
- good choice for data transport

YAML

stands for Y Ain't Markup Language and is a superset of JSON

- lists begin with a hyphen
- dependent on whitespace / indentation
- better suited for configuration than json

privacy notice

- analytics and adsense are run on the site which utilize cookies
- here is a version of the site without adsense and analytics
- submissions are not stored or logged
- contact: json2yaml@protonmail.com

JSON

```
1 {
2   "json": [
3     "rigid",
4     "better for data interchange"
5   ],
6   "yaml": {
7     "slim and flexible",
8     "better for configuration"
9   },
10  "object": {
11    "key": "value",
12    "array": [
13      {
14        "null_value": null
15      },
16      {
17        "boolean": true
18      },
19      {
20        "integer": 1
21      }
22    ],
23    "alias": "aliases are like variables"
24  },
25  {
26    "alias": "aliases are like variables"
27  }
28 }
```

YAML

```
1 ---
2 # -- yaml supports comments, json does not
3 # did you know you can embed json in yaml?
4 # try uncommenting the next line
5 # { foo: 'bar' }
6
7 json:
8   - rigid
9   - better for data interchange
10
11 yaml:
12   - slim and flexible
13   - better for configuration
14
15 object:
16   key: value
17   array:
18     - null_value:
19       - null_value:
20         - boolean: true
21         - integer: 1
22         - alias: &example aliases are like variables
23         - alias: *example
24
25 paragraph: >
26   Blank lines denote
27   paragraph breaks
28
29 content: |-
30   Or we
31   can auto
```

Xml

yaml 비교.JPG

```
<?xml version="1.0" encoding="UTF-8" ?>
<root>
  <apiVersion>v1</apiVersion>
  <kind>Pod</kind>
  <metadata>
    <name>hello-pod</name>
    <labels>
      <app>hello</app>
    </labels>
  </metadata>
  <spec>
    <containers>
      <name>hello-container</name>
      <image>tmkub/hello</image>
      <ports>
        <containerPort>8000</containerPort>
      </ports>
    </containers>
  </spec>
</root>
```

Json

```
{
  "apiVersion": "v1",
  "kind": "Pod",
  "metadata": {
    "name": "hello-pod",
    "labels": {
      "app": "hello"
    }
  },
  "spec": {
    "containers": [
      {
        "name": "hello-container",
        "image": "tmkub/hello",
        "ports": [
          {
            "containerPort": 8000
          }
        ]
      }
    ]
  }
}
```

Yaml

```
apiVersion: v1
kind: Pod
metadata:
  name: hello-pod
  labels:
    app: hello
spec:
  containers:
  - name: hello-container
    image: tmkub/hello
    ports:
    - containerPort: 8000
```

XML	JSON	YAML
<pre><Servers> <Server> <name>Server1</name> <owner>John</owner> <created>123456</created> <status>active</status> </Server> </Servers></pre>	<pre>{ Servers: [{ name: Server1, owner: John, created: 123456, status: active }] }</pre>	<pre>Servers: - name: Server1 owner: John created: 123456 status: active</pre>

YAML 바로 알기

YAML ✓

```
1 ---
2 # YAML 은 주석을 허용 합니다. JSON은 허용하지 않습니다.
3 json:
4   - rigid
5   - better for data interchange
6 yaml:
7   - slim and flexible
8   - better for configuration
9 object:
10  key: value
11  array:
12    - null_value:
13      - boolean: true
14      - integer: 1
15      - alias: &example aliases are like variables
16      - alias: *example
17 paragraph: >
18   Blank lines denote
19
20   paragraph breaks
21 content: |-
22   Or we
23   can auto
24   convert line breaks
25   to save space
26 alias: &foo
27 bar: baz
28 alias_reuse: *foo
```

JSON

```
1 {
2   "json": [
3     "rigid",
4     "better for data interchange"
5   ],
6   "yaml": [
7     "slim and flexible",
8     "better for configuration"
9   ],
10  "object": {
11    "key": "value",
12    "array": [
13      {
14        "null_value": null
15      },
16      {
17        "boolean": true
18      },
19      {
20        "integer": 1
21      },
22      {
23        "alias": "aliases are like variables"
24      },
25      {
26        "alias": "aliases are like variables"
27      }
28    ]
29  },
30  "paragraph": "Blank lines denote\nparagraph breaks\n",
31  "content": "Or we\ncan auto\nconvert line breaks\nto save space",
32  "alias": {
33    "bar": "baz"
34  },
35  "alias_reuse": {
36    "bar": "baz"
37  }
38 }
```