쿠버네티스 실습 1 - YAML

---

1. YAML 이란?

- 데이터 직렬화에 쓰이는 포맷/양식 중 하나
    - 데이터 직렬화란?
        - 서비스간에 Data 를 전송할 때 쓰이는 포맷으로 변환하는 작업
            - ex) 쿠버네티스 마스터에게 요청을 보낼 때 사용
    - 다른 데이터 직렬화 포맷
        - XML, JSON
- 파일 포맷
    - `.yaml`, `.yml`

---

2. YAML 특징

가독성

- YAML 은 **사람이 읽기 쉽도록** 디자인
  - YAML 포맷

```
apiVersion: v1
kind: Pod
metadata:
  name: example
spec:
  containers:
    - name: busybox
      image: busybox:1.25
```

  - JSON 포맷

```
{
  "apiVersion": "v1",
  "kind": "Pod",
  "metadata": {
    "name": "example"
  },
  "spec": {
    "containers": [
      {
        "name": "busybox",
        "image": "busybox:1.25"
      }
    ]
  }
}
```

Widely-use

- kubernetes manifests 명세
- docker compose 명세
- ansible playbook 명세
- github action workflow 명세

Strict-Validation

- 줄 바꿈

- 들여쓰기
  - `Tab` VS `Space`

---

3. 문법

1) Key-Value

- Recursive 한 key-value pair 의 집합

```
apiVersion: v1
kind: Pod
metadata:
  name: example
spec:
  containers:
    - name: busybox
      image: busybox:1.25
```

2) 주석

- `#` 를 줄의 맨 앞에 작성하면 주석 처리됩니다.

```
# kubernetes pod exmaple 입니다.
apiVersion: v1
kind: Pod
metadata:
  name: example
# 중간에 작성해도 됩니다.
spec:
  # 여기도 주석을 달 수 있습니다.
  containers:
    - name: busybox
      image: busybox:1.25
```

3) 자료형

string

```
# 일반적인 문자열은 그냥 작성해도 되고, 따옴표로 감싸도 됩니다.
example: this is 1st string
example: "this is 1st string"
```

```
# 반드시 따옴표로 감싸주어야 하는 경우 :
# 1) 숫자를 문자열 타입으로 지정하고 싶은 경우
example: 123
example: "123"

# y, yes, true 등의 YAML 예약어와 겹치는 경우
example: "y"

# :, {, }, ,, #, *, =, \n 등의 특수 문자를 포함한 경우
example: "a : b"
example: "a#bc*"
```

integer

```
# integer type
example: 123

# hexadecimal type: 0x 로 시작
example: 0x1fff
```

float

```
# float type
example: 99.9

# exponential type
example: 1.23e+03 # 1.23 x 10^3 = 1230
```

boolean

```
# True
example: true
example: yes
example: on

# False
example: false
example: no
example: off
```

4) List

```
# - 를 사용하여 list 를 명시할 수 있습니다.
examples:
  - ex_one: 1
  - ex_two: 2

# [ ] 로 입력해도 됩니다.
examples: ["1", "2", "3"]

# list 의 원소는 어떤 자료형이든 가능합니다.
spec:
  containers:
    - name: busybox
      image: busybox:1.25
    - name: ubuntu
      image: ubuntu
      commands:
        - sleep
        - 3600
    - name: python
      image: python:3.9
```

5) Multi-line strings

`|`

- 중간에 위치한 **빈 줄**을 `\n` 으로 처리하며, **문자열의 맨 마지막**에 `\n` 을 붙입니다.

```
example: |
  Hello
  Fast
  Campus.
# "Hello\nFast\nCampus.\n" 으로 처리
```

`>`

- 중간에 위치한 **빈 줄**을 제외하고, **문자열의 맨 마지막**에 `\n` 을 붙입니다.

```
example: >
  Hello
  Fast
  Campus.
# "Hello Fast Campus.\n" 으로 처리
```

`|-` , `>-`

- 각각 `|`, `>` 와 동일하되, **문자열의 맨 마지막**에 `\n` 이 추가되지 않습니다.

6) Multi-document yaml

- `---` 라는 구분선을 통해 하나의 yaml 파일에 여러 개의 yaml document 를 작성 가능

```
apiVersion: v1
kind: Pod
metadata:
  name: one
---
apiVersion: v1
kind: Service
metadata:
  name: two
---
apiVersion: v1
kind: Deployment
metadata:
  name: three
```

- 3 개의 yaml document 로 인식

7) 복습

- Pod 의 명세를 작성한 yaml 예시

```
# key-value pairs
apiVersion: v1
kind: Pod
metadata:
  name: example
  labels:
    hello: bye
spec:
  containers:
    # list
    - name: busybox
      image: busybox:1.25
      # list
      ports:
      - containerPort: 80
    - name: another-container
      image: curlimages/curl
```

- 선언형 인터페이스를 위해서, Desired State 를 명시하는 용도로 사용