CH 4.8 Katib 실습

---

1. Prerequisite

- 이전 시간에 생성한 minikube + kubeflow v1.4 환경
  - katib v0.12.0
- 편한 IDE
  - yaml 파일을 읽기 편한 에디터면 충분합니다.

---

2. Katib Experiment 예제 1

- Katib Experiment 리소스 yaml 예제를 함께 보며, 각 필드가 어떤 의미를 가지는지 확인해보겠습니다.

  ▼ random-example.yaml

```
apiVersion: "kubeflow.org/v1beta1"
kind: Experiment
metadata:
  namespace: kubeflow-user-example-com # namespace
  name: demo # experiment name

# Experiment 관련 메타 정보 작성
spec:
  # Objective Function
  # 최적화하기 위한 metric, type, early stopping goal 등을 포함
  objective:
    type: maximize
    goal: 0.99

    # Trial 에서 출력할 때, 정해진 형식으로 StdOut 으로 출력하면 name 을 parsing 할 수 있
음
    # https://www.kubeflow.org/docs/components/katib/experiment/#metrics-coll
ector
```

```
    # objectiveMetricName 은 hp search 를 수행할 objective metric 의 이름
    # addtionalMetricName 은 hp search 와는 관계없지만 함께 출력할 metric 의 이름
    objectiveMetricName: Validation-accuracy
    additionalMetricNames:
      - Train-accuracy

  # Hyperparameter Search Algorithm
  algorithm:
    # Katib 에서는 현재 지원하고 있는 search algorithm 이 다음과 같이 정해져 있습니다.
    # https://www.kubeflow.org/docs/components/katib/experiment/#search-algor
ithms

    # 각각의 algorithm 은 정해진 HP search package 를 사용하여 동작하며,
    # 어떤 docker image 를 사용할 것인지는 katib 설치 당시 배포한 configmap 에 적혀있습니
다.
    # 다음 명령을 통해서 어떤 algorithm 이 어떤 package 를 사용하는지 확인할 수 있습니다.
    # `kubectl get configmap katib-config -o yaml` 의 suggestion 필드 확인
    algorithmName: random

  # 병렬로 실행할 Trial 의 개수
  parallelTrialCount: 2

  # 최대 Trial 개수 (도달하면 실험 종료 : Succeeded status 로 종료)
  maxTrialCount: 2

  # 최대 failed Trial 개수 (도달하면 실험 종료 : Failed status 로 종료)
  maxFailedTrialCount: 2

  # HP Search 를 수행할 space 정의
  # 각각의 hyperparameter 마다 type 은 무엇인지, space 는 무엇인지를 정의
  # https://github.com/kubeflow/katib/blob/195db292374dcf3b39b55dcb3fcd14b3a5
5d5942/pkg/apis/controller/experiments/v1beta1/experiment_types.go#L186-L207
  parameters:
    - name: lr # 뒤의 필드 중 trialTemplate.trialParameters[x].reference 와 일치해
야 합니다. (실수하기 쉬운 부분)
      parameterType: double
      feasibleSpace:
        min: "0.01"
        max: "0.03"
    - name: num-layers
      parameterType: int
      feasibleSpace:
        min: "2"
        max: "5"
    - name: optimizer
      parameterType: categorical
      feasibleSpace:
        list:
          - sgd
          - adam
          - ftrl

  # Suggestion 에 의해 생성된 HP 후보 조합 하나를 input 으로 받아서 학습 및 평가를 진행할 T
rial 의 템플릿
  trialTemplate:

    # 아래 trialSpec.spec.template.spec.containers[x].name 중에서 metric 을 출력하
는 container 의 이름
```

```
    # 지금 예시에서는 container 가 하나뿐이므로 해당 container 의 이름으로 출력
    primaryContainerName: training-container

    # 아래 trialSpec.spec.template.spec.containers[x].command (or args) 에서 사
용할 Hyperparameter 에 대한 메타 정보 정의
    # trialParameters[x].name 은 아래 trialSpec 에서의 값과 매핑되며,
    # trialParameters[x].reference 는 위의 parameters[x].name 과 매핑됩니다.
    trialParameters:
      - name: learningRate
        description: Learning rate for the training model
        reference: lr
      - name: numberLayers
        description: Number of training model layers
        reference: num-layers
      - name: optimizer
        description: Training model optimizer (sdg, adam or ftrl)
        reference: optimizer

    # trialSpec 으로는 Job, TfJob 등의 리소스를 사용할 수 있으며, 본 예시는 Job 을 사용합
니다.
    # https://www.kubeflow.org/docs/components/katib/trial-template/
    trialSpec:
      apiVersion: batch/v1
      kind: Job
      spec:
        template:
          # 현재 버전의 katib 는 istio sidecar 와 함께 사용할 수 없습니다.
          # 자세한 내용은 다음 페이지를 확인하시기 바랍니다.
          # https://www.kubeflow.org/docs/components/katib/hyperparameter/#ex
ample-using-random-search-algorithm
          # https://github.com/kubeflow/katib/issues/1638
          metadata:
            annotations:
              sidecar.istio.io/inject: 'false'
          spec:
            containers:
              - name: training-container

                # 해당 이미지는 미리 docker build and push 되어있어야 사용 가능
                # 해당 docker image 를 빌드한 Dockerfile 및 소스코드는 다음 경로에서 확
인
                # https://github.com/kubeflow/katib/tree/983a867/examples/v1b
eta1/trial-images/mxnet-mnist
                image: docker.io/kubeflowkatib/mxnet-mnist:v1beta1-45c5727
                command:
                  - "python3"
                  - "/opt/mxnet-mnist/mnist.py"
                  - "--batch-size=64"
                  - "--lr=${trialParameters.learningRate}"
                  - "--num-layers=${trialParameters.numberLayers}"
                  - "--optimizer=${trialParameters.optimizer}"
                  - "--num-epochs=1" # 테스트 시 시간 소요를 줄이기 위해 epoch 은 1
 회만 수행하겠습니다.
            restartPolicy: Never
```

3. Quick Start

- current namespace 를 user namespace 로 변경합니다.
  - `kubens kubeflow-user-example-com`
- random-example 을 `kubectl` 로 생성해봅니다.
  - `kubectl apply -f random-example.yaml`
- 리소스별 진행 상황을 확인합니다. (experiment, suggestion, trial, pod)
  - `watch -n1 'kubectl get experiment,suggestion,trial,pod'`
  - `kubectl logs <pod-name> -c metrics-logger-and-collector`
- UI 에서도 확인해봅니다.
  - 포트포워딩
    - `k port-forward svc/istio-ingressgateway -n istio-system 8080:80`
  - UI 접속 후 Experiments(AutoML) 탭 접속
    - localhost:8080

---

4. Katib UI 를 통한 생성

- UI 접속 후, New Experiment 버튼을 클릭하여 생성해보겠습니다.
  - `Hyper Parameters` 와 `Trial Template` 부분을 일치시켜야함에 주의해주시기 바랍니다.
- 단, UI 에서는 istio-sidecar inject false 를 추가한 템플릿을 미리 만들어두지 않으면 사용할 수 없으므로, kubeflow-user-example-com ns 에서는 istio-sidecar inject 를 항상 false 로 수행하도록 변경합니다.
  - `kubectl edit ns kubeflow-user-example-com`
    - `istio-injection: enabled` 를 `istio-injection: disabled` 로 변경합니다.
- 또한, epoch 도 변경할 수 없으므로 Experiment 가 종료되는데 시간이 다소 소요될 수 있습니다.
  - 각 trial pod 의 log 를 확인하여 정상적으로 진행 중인지 확인할 수 있습니다.

---

5. Katib Experiment 예제 2

- 이번에는 또 다른 Katib Experiment 리소스 yaml 예제를 함께 보며, hyperparameter search algorithm 을 변경해보겠습니다.

  ▼ bayesian-example.yaml

```
apiVersion: kubeflow.org/v1beta1
kind: Experiment
metadata:
  namespace: kubeflow-user-example-com
  name: bayesian-optimization
spec:
  objective:
    type: maximize
    goal: 0.99
    objectiveMetricName: Validation-accuracy

  algorithm:
    algorithmName: bayesianoptimization
    # 아래 부분이 추가된 것을 확인할 수 있습니다.
    # algorithmSetting 에서 지원하는 key-value 는 알고리즘마다 다르며,
    # 각 알고리즘에서 지원하는 algorithmSetting 은 다음 페이지에서 확인할 수 있습니다.
    # https://www.kubeflow.org/docs/components/katib/experiment/#search-algorithms-in-detail

    # bayesian optimization 에서 지원하는 algorithmSetting 은 다음 페이지에서 확인할 수 있습니다.
    # https://www.kubeflow.org/docs/components/katib/experiment/#bayesian-optimization
    # 지정하지 않은 value 는 Default value 로 설정됩니다.
    algorithmSettings:
      - name: "random_state" # random seeed
        value: "1234"
      - name: "n_initial_points" # 근사를 위한 inital point 의 개수
        value: "5"
  parallelTrialCount: 1
  maxTrialCount: 1
  maxFailedTrialCount: 1
  parameters:
    - name: lr
      parameterType: double
      feasibleSpace:
        min: "0.001"
        max: "0.05"
    - name: num-layers
      parameterType: int
      feasibleSpace:
        min: "2"
        max: "7"
    - name: optimizer
      parameterType: categorical
      feasibleSpace:
        list:
          - sgd
          - adam
  trialTemplate:
```

```
    primaryContainerName: training-container
    trialParameters:
      - name: learningRate
        description: Learning rate for the training model
        reference: lr
      - name: numberLayers
        description: Number of training model layers
        reference: num-layers
      - name: optimizer
        description: Training model optimizer (sdg, adam or ftrl)
        reference: optimizer
    trialSpec:
      apiVersion: batch/v1
      kind: Job
      spec:
        template:
          metadata:
            annotations:
              sidecar.istio.io/inject: 'false'
          spec:
            containers:
              - name: training-container
                image: docker.io/kubeflowkatib/mxnet-mnist:v1beta1-45c5727
                command:
                  - "python3"
                  - "/opt/mxnet-mnist/mnist.py"
                  - "--batch-size=64"
                  - "--lr=${trialParameters.learningRate}"
                  - "--num-layers=${trialParameters.numberLayers}"
                  - "--optimizer=${trialParameters.optimizer}"
                  - "--num-epochs=1" # 성능이 개선되는 걸 확인하고 싶다면 이 값을 변경해
주세요.
            restartPolicy: Never
```

- bayesian-example 을 `kubectl` 로 생성해봅니다.
  - `kubectl apply -f bayesian-example.yaml`
- UI 에서도 확인해봅니다.

---

6. Katib Experiment 예제 3

- 이번에는 또 다른 Katib Experiment 리소스 yaml 예제를 함께 보며, metrics-collector 를 Default(StdOut) 에서 File 로 변경해보겠습니다.

▼ file-metrics-collector-example.yaml

```
apiVersion: kubeflow.org/v1beta1
kind: Experiment
metadata:
  namespace: kubeflow-user-example-com
```

```
  name: file-metrics-collector
spec:
  objective:
    type: maximize
    goal: 0.99
    objectiveMetricName: accuracy
    additionalMetricNames:
      - loss
  # 이 부분이 추가되었습니다.
  metricsCollectorSpec:
    # Metrics collector 의 type 을 지정할 수 있습니다.
    # Default 는 StdOut 이며, File, TesnroflowEvent 등을 지원합니다.
    # 자세한 내용은 다음 페이지를 참고하시기 바랍니다.
    # https://www.kubeflow.org/docs/components/katib/experiment/#metrics-coll
ector

    # 이번 예제에서는 File 타입을 사용해봅니다.
    collector:
      kind: File
    # File Metrics Collector 가 Metrics 를 어떻게 가져올 것인지를 정의하는 부분입니다.
    source:
      # metrics 를 어느 경로에서 가져올 것인지를 작성합니다.
      # 각 Trial 에서는 반드시 해당 경로에 metrics 를 저장해야 정상적으로 parsing 할 수 있
습니다.
      fileSystemPath:
        path: "/katib/mnist.log"
        kind: File
      filter:
        # 해당 파일에서 어떤 format 을 metrics name 과 metrics value 로 인식하고 pars
ing 해올 것인지를 정의합니다.
        metricsFormat:
          - "{metricName: ([\\w|-]+), metricValue: ((-?\\d+)(\\.\\d+)?)}"


  algorithm:
    algorithmName: random
  parallelTrialCount: 1
  maxTrialCount: 1
  maxFailedTrialCount: 1
  parameters:
    - name: lr
      parameterType: double
      feasibleSpace:
        min: "0.01"
        max: "0.03"
    - name: momentum
      parameterType: double
      feasibleSpace:
        min: "0.3"
        max: "0.7"
  trialTemplate:
    primaryContainerName: training-container
    trialParameters:
      - name: learningRate
        description: Learning rate for the training model
        reference: lr
      - name: momentum
        description: Momentum for the training model
```

```
        reference: momentum
    trialSpec:
      apiVersion: batch/v1
      kind: Job
      spec:
        template:
          metadata:
            annotations:
              sidecar.istio.io/inject: 'false'
          spec:
            containers:
              - name: training-container
                # 해당 이미지의 Dockerfile 과 src code 는 다음 페이지에서 확인할 수 있습
니다.
                # https://github.com/kubeflow/katib/tree/983a867/examples/v1b
eta1/trial-images/pytorch-mnist
                image: docker.io/kubeflowkatib/pytorch-mnist:v1beta1-45c5727
                command:
                  - "python3"
                  - "/opt/pytorch-mnist/mnist.py"
                  - "--log-path=/katib/mnist.log"
                  - "--lr=${trialParameters.learningRate}"
                  - "--momentum=${trialParameters.momentum}"
                  - "--epochs=1"
            restartPolicy: Never
```

- file-metrics-collector-example 을 `kubectl` 로 생성해봅니다.
  - `kubectl apply -f file-metrics-collector-example.yaml`
- UI 에서도 확인해봅니다.