Kubernetes와 SageMaker를 활용하여
기계학습 워크로드 관리하기

강성문
솔루션즈 아키텍트
AWS
Agenda

Fully managed machine learning with Amazon SageMaker

ML Workload management with Kubernetes
  - Amazon SageMaker Operator for Kubernetes
  - Kubeflow

Other solutions (Step Functions, Apache Airflow)

Machine Learning Lifecycle Management
Fully managed machine learning with Amazon SageMaker
Amazon SageMaker is a fully managed service that covers the entire machine learning workflow.

- Jupyter notebook instances
- High-performance algorithms
- Large-scale training
- Optimization
- One-click deployment
- Fully managed with auto-scaling
Amazon SageMaker manages ML infrastructure

Build
- Pre-built notebook instances
- Highly optimized machine learning algorithms

Train
- One-click training for ML, deep learning, and custom algorithms
- Automatic model tuning (hyperparameter optimization)

Deploy
- Deployment without engineering effort
- Fully managed hosting at scale
Demo – Managed ML development with Amazon SageMaker

Amazon SageMaker
Build, train, and deploy machine learning models at scale
The quickest and easiest way to get ML models from idea to production.

Get started
Explore AWS data in your notebooks, and use algorithms to create models via training jobs. Leverage Notebook instances in the cloud to begin.

Create notebook instance

Start with an overview

How it works

Pricing (US)
With Amazon SageMaker, you pay only for what you use. Authoring, training and testing is billed by the second, with no minimum fees and no upfront commitments.

Learn more
Building ML workflows with Amazon SageMaker and Kubernetes
• Focuses on data science, business outcome and speed to market

• **Wants minimum dependency on the DevOps team** for experimentation and model development

• **No or limited K8s and infrastructure knowledge**

• Wants to **leverage existing K8s** investment and best practices

• Wants to manage using familiar K8s **construct and syntax**

• Limited ML knowledge & engineering experience with ML workloads
Under the hood – Amazon SageMaker and Kubernetes

Key Features

• Amazon SageMaker Operators for training, tuning, inference
• Natively interact with Amazon SageMaker jobs using K8s tools (e.g., get pods, describe)
• Stream and view logs from Amazon SageMaker in K8s
• Helm Charts to assist with setup and spec creation
Why together?

Orchestration
Repeatable Pipelines
Automation
CI / CD

+---------------------------------+
<table>
<thead>
<tr>
<th>aws</th>
</tr>
</thead>
<tbody>
<tr>
<td>ML Operations &amp; troubleshooting</td>
</tr>
<tr>
<td>Optimization, monitoring, logging</td>
</tr>
<tr>
<td>Model hosting</td>
</tr>
<tr>
<td>Model training</td>
</tr>
<tr>
<td>Security &amp; compliance</td>
</tr>
<tr>
<td>Service provisioning</td>
</tr>
<tr>
<td>Infrastructure &amp; configuration</td>
</tr>
</tbody>
</table>

Rancher

Amazon Web Services
Demo – XGBoost Kubernetes Amazon SageMaker Operator
Amazon SageMaker Operators for Kubernetes

Why GitHub?  Enterprise  Explore  Marketplace  Pricing

Search

AWS / amazon-sagemaker-operator-for-k8s

Join GitHub today
GitHub is home to over 40 million developers working together to host and review code, manage projects, and build software together.

Sign up

Amazon SageMaker operator for Kubernetes
https://sagemaker.readthedocs.io/en/s...

79 commits 9 branches 0 packages 2 releases 5 contributors Apache-2.0

Branch: master  New pull request

github

Initial commit

2 months ago

api/v1

Renamed Go module (966)

20 days ago

bin

Renamed Go module (966)

20 days ago

codebuild

Adding Travis CI related files for SMInk tests (863)

last month

Posts

Official AWS Podcast

AWS Case Studies

amazon-sagemaker-operator-for-k8s

Resources

Getting Started

What's New

Top Posts

https://github.com/aws/amazon-sagemaker-operator-for-k8s


Introducing Amazon SageMaker Operators for Kubernetes

by Aditya Bindal | on 02 DEC 2019 | in Artificial Intelligence, AWS Reinvent, SageMaker | Mentions | Comments | Share

AWS is excited to introduce Amazon SageMaker Operators for Kubernetes, a new capability that makes it easier for developers and data scientists using Kubernetes to train, tune, and deploy machine learning (ML) models in Amazon SageMaker. Customers can install these Amazon SageMaker Operators on their Kubernetes clusters to create Amazon SageMaker jobs natively using the Kubernetes API and command-line Kubernetes tools such as `kubectl`. In addition to this blog post, we also published a whitepaper about how to do machine learning on Amazon SageMaker and Kubernetes.

Many AWS customers use Kubernetes, an open-source general-purpose container orchestration system, to deploy and manage containerized applications, often via a managed service such as Amazon Elastic Kubernetes Service (EKS). This enables data scientists and developers, for example, to set up repeatable ML pipelines and maintain greater control over their training and inference workloads. However, to support ML workloads these customers still need to write custom code to optimize the underlying ML infrastructure, ensure high availability and reliability, provide data science productivity tools, and comply with appropriate security and regulatory requirements. For example, when Kubernetes customers use GPUs for training and inference, they often need to change how Kubernetes schedules and scales GPU workloads in order to increase utilization, throughput, and availability. Similarly, for deploying trained models to production for inference, Kubernetes customers have to spend additional time in setting up and optimizing their auto-scaling clusters across multiple Availability Zones.

Amazon SageMaker Operators for Kubernetes bridges this gap, and customers are now spared all the heavy lifting of integrating their Amazon SageMaker and Kubernetes workflows. Starting today, customers using Kubernetes can make a simple call to Amazon SageMaker, a modular and fully-managed service that makes it easier to build, train, and deploy machine learning (ML) models at scale. With workflows in Amazon SageMaker, compute resources are pre-configured and optimized, only provisioned when requested, scaled as needed, and shut down automatically when jobs complete, offering near 100% utilization. Now with Amazon SageMaker Operators for Kubernetes, customers can continue to enjoy the scalability and productivity benefits of Kubernetes and EKS, along with integrating the many additional benefits.
Building ML workflows with Amazon SageMaker and Kubeflow
What is Kubeflow

Containerized machine learning platform

Makes it easy to develop, deploy, and manage portable, scalable end-to-end ML workflows on k8s

“Toolkit” – loosely coupled tools and blueprints for ML

End to End ML workflow – ML code is only a small component

Pipelines – Machine Learning Job Orchestrator

- Compose, deploy, and manage end-to-end ML workflows
  - End-to-end orchestration
  - Easy, rapid, and reliable experimentation
  - Easy re-use

- Built using Pipelines SDK
  - kfp.compiler, kfp.components, kfp.Client

- Uses Argo under the hood to orchestrate resources
Creating Kubeflow Pipeline Components

```python
@dsl.pipeline(
    name='Sample Trainer',
    description=''
)
def sample_train_pipeline(...):
    create_cluster_op = CreateClusterOp('create-cluster', ...)
    analyze_op = AnalyzeOp('analyze', ...)
    transform_op = TransformOp('transform', ...)
    train_op = TrainerOp('train', ...)
    predict_op = PredictOp('predict', ...)
    confusion_matrix_op = ConfusionMatrixOp('confusion-matrix', ...)
    roc_op = RocOp('roc', ...)

kfp.compiler.Compiler().compile(sample_train_pipeline, 'my-pipeline.zip')
```
Demo – KMeans Classification with Kubeflow Pipeline
MACHINE LEARNING USING KUBEFLOW

Kubeflow provides a simple, portable, and scalable way of running Machine Learning workloads on Kubernetes.

In this module, we will install Kubeflow on Amazon EKS, run a single-node training and inference using TensorFlow, train and deploy model locally and remotely using Fairing, setup Kubeflow pipeline and review how to call AWS managed services such as Sagemaker for training and inference.
Kubernetes workshop on EKS. Mainly focus on AWS integration examples. Please go check kubeflow website http://kubeflow.org for other examples.

Latest commit 4ffe115 on Dec 2, 2019

- Doc: Address workshop feedbacks and remove unready notebooks
- Notebooks: Ask users to interrupt logs update if they want to move forward (#21)
- .gitignore: Add python gitignore
- README.md: Update kubeflow workshop
Building ML workflows with Amazon SageMaker and Step functions
What is Step Functions?

Mountains
People
Snow

Task
Choice
Fail
Parallel

Start
ExtractImageMetadata
ImageTypeCheck
StoreImageMetadata
Rekognition
Thumbnail
AddRecognizedTags
End

NotSupportedImageType
Run tasks with any compute resource

Activity

Start ➔ HelloWorld ➔ End

Worker requests tasks from Step Functions

AWS Lambda function

Start ➔ HelloWorld ➔ End

Step Functions invokes Lambda function

Worker long poll

Synchronous request

Amazon EC2

Amazon ECS

Traditional server

Amazon SageMaker
AWS Service integration with Step functions

Step Functions 사용을 통한 Amazon SageMaker 관리

PDF

Step Functions can control certain AWS services directly from the Amazon States 언어. For more information, see the following:

- 서비스 통합
- 파라미터를 서비스 API에 전달

지원되는 Amazon SageMaker API 및 구문:

- CreateEndpoint
  - 요청 구문
  - 지원되는 파라미터:
    - EndpointConfigName
    - EndpointName
    - Tags
  - 응답 구문
- CreateEndpointConfig
  - 요청 구문
  - 지원되는 파라미터:
    - EndpointConfigName
    - KmsKeyId
    - ProductionVariants
    - Tags
Add Amazon SageMaker training and transform jobs in your workflows

"Synchronously Run a Training Job": {
    "Type": "Task",
    "Resource": "arn:aws:states:::sagemaker:CreateTrainingJob.sync",
    "Parameters": {
        "AlgorithmSpecification": {...},
        "HyperParameters": {...},
        "InputDataConfig": [...],
        ...
    },
    "Catch": [
        {
            "ErrorEquals": ["States.TaskFailed"],
            "ResultPath": "$\cdot cause",
            "Next": "Notify on Error"
        },
        "ResultPath": "$\cdot jobInfo",
        "Next": "Report Success"
    ]
}

"Synchronously Run a Transform Job": {
    "Type": "Task",
    "Resource": "arn:aws:states:::sagemaker:CreateTransformJob.sync",
    "Parameters": {
        "TransformJobName.$": "$\cdot transform",
        "ModelName.$": "$\cdot model",
        "MaxConcurrentTransforms": 8,
        ...
    },
    "Catch": [
        {
            "ErrorEquals": ["States.TaskFailed"],
            "ResultPath": "$\cdot cause",
            "Next": "Notify on Error"
        },
        "ResultPath": "$\cdot jobInfo",
        "Next": "Report Success"
    ]
}
Define workflows in JSON

```json
{
  "StartAt": "Download",
  "States": {
    "Download": {
      "Type": "Task",
      "Next": "Train"
    },
    "Train": {
      "Type": "Task",
      "Resource": "arn:aws:states:::sagemaker:createTrainingJob.sync",
      "ResultPath": "$.training_job",
      "Parameters": {
        "AlgorithmSpecification": {
          "TrainingImage": "811284229777.dkr.ecr.us-east-1.amazonaws.com/image-classification:latest",
          "TrainingInputMode": "File"
        }
      }
    }
  }
}
```
Demo – ML Workflow with Step functions

```
"Parallel": {
  "Type": "Parallel",
  "Next": "CompareResult",
  "Branches": [
    {
      "StartAt": "XGBoost(Built-in)",
      "States": {
        "XGBoost(Built-in)": {
          "Type": "Task",
          "Resource": "arn:aws:states:::sagemaker:createTrainingJob.sync",
          "End": true,
          "Parameters": {
            "TrainingJobName": "XGBoost-SM-StepFunctions-003",
            "ResourceConfig": {
              "InstanceCount": 1,
              "InstanceType": "ml.c5.xlarge",
              "VolumeSizeInGB": 20
            },
            "HyperParameters": {
              "max_depth": "5",
              "eta": "0.2",
              "gamma": "4",
              "min_child_weight": "6",
              "max_delta_feature": "0.0"
            }
          }
        }
      }
    }
  ]
}
```

Diagram:
- Start
  - ReadAndPreprocess
  - XGBoost(Built-in)
  - Tensorflow(BYOS)
  - CompareResult
  - ChooseTransform
    - ConditionFail
    - XGBoostTransform
    - TensorflowTransform
  - End
Building ML workflows with Amazon SageMaker and Apache Airflow
Apache Airflow

• Built by Airbnb, open sourced under Apache
• Platform to programmatically author, schedule and monitor workflows
• Rich user interface to visualize job pipelines, monitor progress and troubleshoot issues
• Need to be installed separately on Amazon EC2
Apache Airflow Amazon SageMaker Operators

- sagemaker_training_operator
- sagemaker_tuning_operator
- sagemaker_model_operator
- sagemaker_endpoint_config_operator
- sagemaker_endpoint_operator
- sagemaker_transform_operator
- segment_track_event_operator

https://airflow.apache.org/docs/1.10.3/_api/airflow/contrib/operators/index.html
Demo – Recommendation batch job with Airflow
ML workflows with Amazon SageMaker and Apache Airflow

Summary
(Machine Learning Lifecycle)
MLOps vs DevOps

MLOps

DevOps

Source Codes

Data

Models
AWS 머신러닝(ML) 교육 및 자격증

Amazon의 개발자와 데이터 과학자들 교육하는 데 직접 활용 되었던 커리큘럼을 기반으로 학습하세요!

전체 팀을 위한
머신러닝 교육

비즈니스 의사 결정자,
데이터 과학자, 개발자,
데이터 플랫폼 엔지니어 등
역할에 따라 제공되는
맞춤형 학습 경로를
확인하세요.

원하는 방법으로!
교육 유연성 제공

약 65개 이상의
온라인 과정 및
AWS 전문 강사를 통해
실습과 실적용의 기회가
제공되는 강의실 교육이
준비되어 있습니다.

전문성에 대한
검증

업계에서 인정받는
‘AWS 공인 머신러닝 – 전문분야’
자격증을 통해
머신러닝 모델을 구축, 학습,
튜닝 및 배포하는 데 필요한
전문 지식이 있음을
입증할 수 있습니다.

https://aws.amazon.com/ko/training/
learning-paths/machine-learning/
여러분의 소중한 피드백을 기다립니다!
강연 평가 및 설문 조사에 참여해 주세요.
감사합니다