Spark-on-K8S로 제품 만들기

kt NexR, 민영근

순서

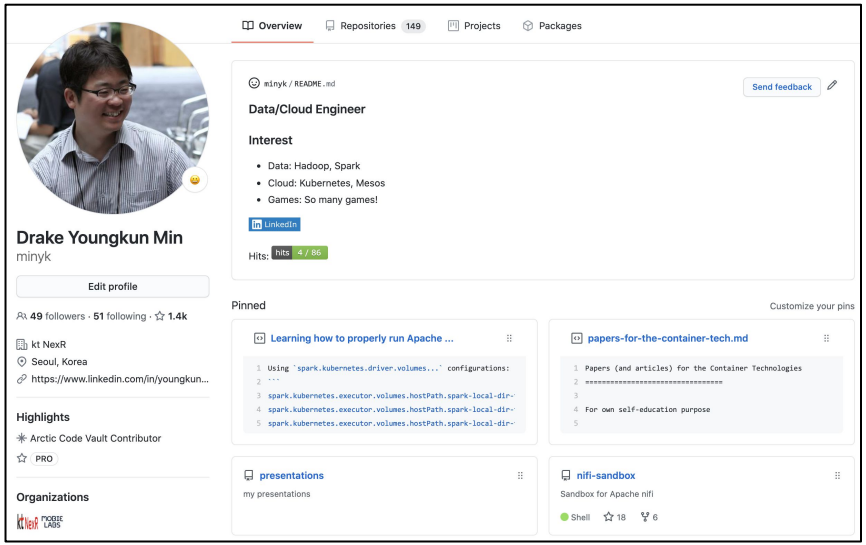
- 자기소개/ 회사소개
- Apache Spark
  - 소개/ Spark 3.0의 새 기능들
- Spark on Kubernetes
  - 기본 개념
  - Spark 3.0에 추가된 K8S 지원 기능들
- Spark on Kubernetes 이슈들
  - Spark 3.0/3.1의 이슈들
- Spark on Kubernetes 제품화의 문제들
- 마무리
- 참고

자기소개

- 2019.4 ~ kt NexR R&D 센터
- 2016.6 ~ 2019.4 AJ 네트웍스 IT센터
- 2013.8 ~ 2016.6 kt NexR TA팀
- 2011.8 ~ 2013.7 단국대 연구전담교수
- 2011.2 단국대학교 공학박사

- 관심사: 데이터 처리, 컨테이너

https://github.com/minyk



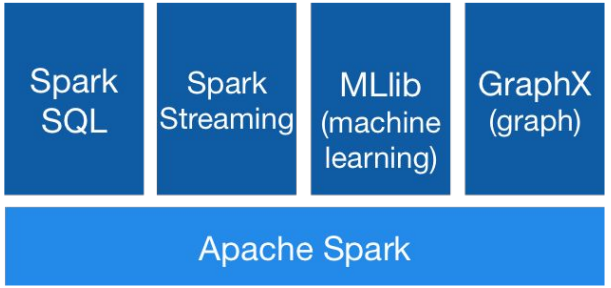
kt NexR

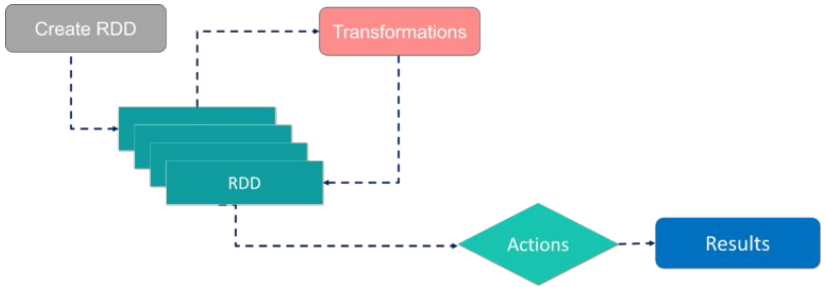
- 2020: K8S 기반의 분석 플랫폼 NexR Enterprise v1.2.0 출시
- 2016: 실시간 처리 플랫폼 Lean Stream 1.0 출시
- 2014: Hadoop 2.0 적용된 NDAP 4.0 출시
- 2012: Hadoop 기반의 빅데이터 플랫폼 NDAP 출시, kt 빅데이터 플랫폼 개발
- 2011: kt 그룹사 편입
- 2007: NexR 설립


APACHE
Spark
TM

Apache Spark: 소개

- Unified Analytics Engine for Large-scale Data Processing
  - 2009년 시작, Hadoop MapReduce 프로그래밍 모델의 대안
  - API(scala/java/python/R), SQL, Streaming, GraphX, ML 지원
- Hadoop Mapreduce의 뒤를 잇는 대용량 데이터 처리 엔진
  - RDD(Resilient Distributed DataSet): 메모리에 저장되는 변경되지 않는 데이터집합
  - Lazy execution이 특징





Apache Spark: 소개

- Hadoop 생태계의 한 축
  - Hadoop DFS, Hadoop YARN 호환
  - Hive JDBC Driver 호환
  - Hive Metastore 호환
- 높은 확장성
  - Datasource V2 API, Custom Catalog
  - SQL 확장 명령 개발 가능
- 강력한 Python 지원
  - Scala와 거의 동일한 수준에서 지원
  - Jupyter 등 Python 환경과도 자연스럽게 통합
  - Pandas 호환 라이브러리 Databricks Koala

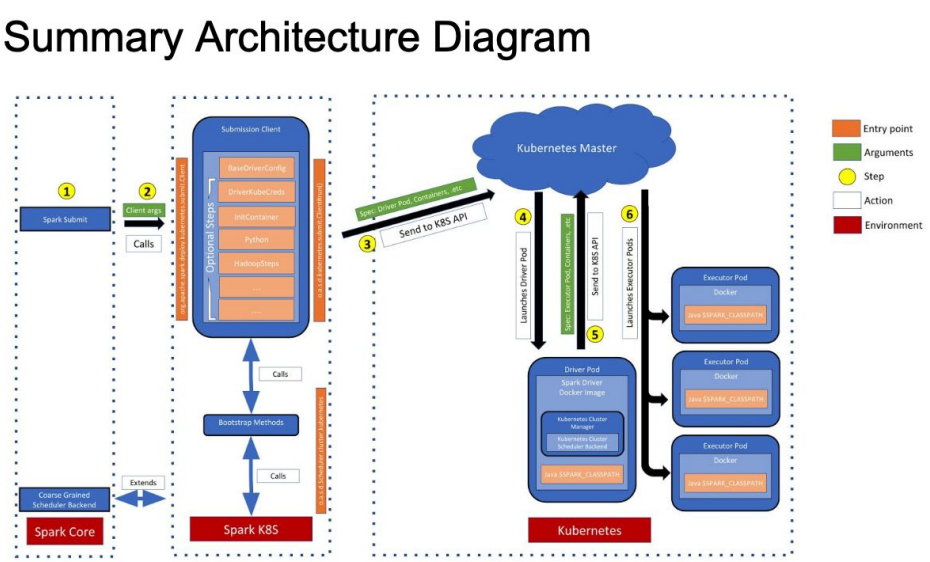
Apache Spark: 3.0의 새 기능

- Accelerator-aware Scheduler (SPARK-24615)
- Adaptive Query Execution (SPARK-31412)
- Dynamic Partition Pruning (SPARK-11150)
- Structured Streaming UI (SPARK-29543)
- Catalog plugin API (SPARK-31121)
- Java 11 support (SPARK-24417)
- Hadoop 3 support (SPARK-23534)

Spark on Kubernetes

Spark on Kubernetes

- 2016.11~2018.3 진행
  - SPARK-18278 SPIP: Support native submission of spark jobs to a kubernetes cluster
- 별도의 repo에서 개발되어 2.3.0에 병합
- 현재(3.0.1) 아직은 실험적 기능
  - Dynamic Allocation 미지원
  - Shuffle 서비스 미지원



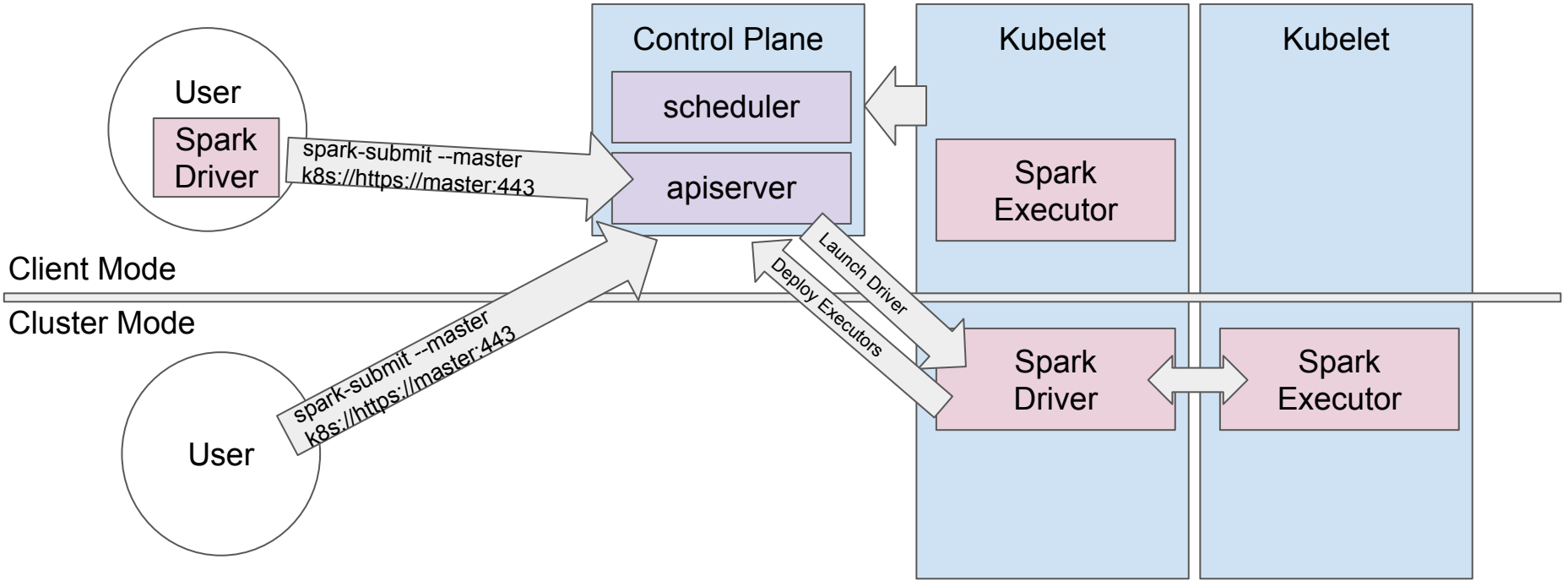
From Design Proposal Doc.

Spark-on-K8S 두가지 방법

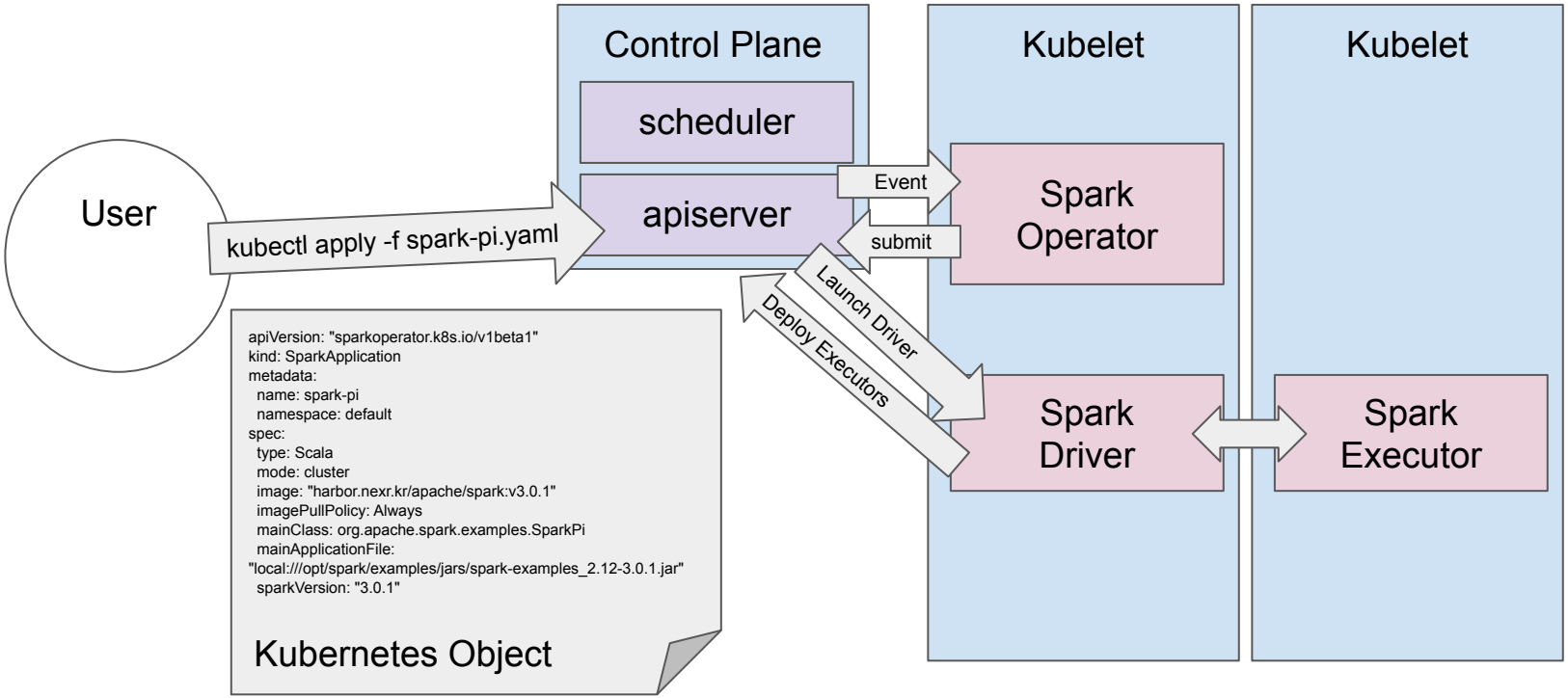
- Spark native
  - 스파크의 K8S 지원 기능
  - spark-submit 사용

- Kubernetes native
  - 별도의 spark operator
  - kubectl / YAML manifest 사용

Spark on Kubernetes 구조



Spark Operator 구조



Spark 3.0에서 추가된 K8S 지원 기능들

- Driver/Executor Pod template 지원
  - pod spec 을 커스텀할 수 있도록 지원. Ambassador 패턴 같은 추가 컨테이너를 붙이기 쉬움

- Driver의 service에 annotation 추가 가능
  - 추가 메타데이터 기록 가능

```
apiVersion: v1
 Kind: Pod
 metadata:
   labels:
     podtemplate: sidekick
 spec:
   containers:
   - name: spark-kubernetes-executor
   - name: side-kick
     image: "minio/sidekick:v0.1.4"
     imagePullPolicy: Always
     args:
       - --address
       - :9000
       - http://10.0.3.19:8080
       - http://10.0.3.20:8080
     ports:
         - containerPort: 9000
```

Spark 3.0에서 추가된 K8S 지원 기능들

- local 저장소로 tmpfs를 mount 하는 설정 추가
  - 설정 한 줄

```
spark.kubernetes.local.dirs.tmpfs=true
```

- local 저장소로 지정된 volume을 mount 할 수 있도록 개선

```
spark.kubernetes.driver.volumes.hostPath.spark-local-dir-1.mount.path=/data
spark.kubernetes.driver.volumes.hostPath.spark-local-dir-1.mount.readOnly=false
```

Spark on K8S 연관 있는 현재 이슈들

- Add support for dynamic resource allocation(SPARK-24432)
  - K8S에서 동적할당을 제공하기 위한 개발 진행 중
  - 별도 저장소를 사용하는 (SPARK-25299) 이슈가 블럭 중
  - 실험 기능(spark.dynamicAllocation.shuffleTracking.enabled)을 사용해서 “soft”한 방법은 있음
- hostPath로 SSD를 Executor로 mount 시 관리 문제 발생
  - 비정상 종료된 Executor가 SSD의 데이터를 삭제하지 못함 -> 디스크 공간 부족
  - 별도 관리하기 어려움: GUID를 사용해서 로그를 봐야만 파악 가능
- Volume에서 PVC를 사용하는 경우 claimName을 하나만 지정할 수 있음
  - 하나의 Volume을 여러개의 Executor가 mount 해서 사용
  - 3.1에서 개선

Spark 3.1(early 2021)의 K8S 지원 이슈들

- SPARK-33005: Kubernetes GA
  - GA 될 예정
  - 동적 할당 기능은 일단 지원하지 않는 것으로.
- SPARK-30949 Driver cores in kubernetes are coupled with container resources, not spark.driver.cores
  - Driver가 실제로 사용하는 코어의 수를 설정에서 읽어오던 것을 K8S 컨테이너 자원에서 사용하도록 개선
- SPARK-32971 Support dynamic PVC creation/deletion for K8s executors
  - PVC 동적 생성/삭제
  - StorageClass 지정 가능: 다양한 StorageClass 를 사용하는 활용 방법 예상

+

+ α = ???

Spark on Kubernetes 제품화의 문제들

Spark on Kubernetes 제품화의 문제들

- 자원 효율성
- 로컬 디스크 사용으로 인한 성능 저하
- 별도의 사용자 “빅”데이터 저장 공간
- 파일/ 데이터 형식
- User Interface/eXperience

Spark on Kubernetes 제품화의 문제들

자원 효율성

- K8S의 기본 스케쥴러 사용: 대규모의 Batch 처리 시 문제 발생 가능
  - “spark.scheduler.minRegisteredResourcesRatio”: 요청한 Executor 중 80%(기본)가 구동/등록되어야 연산 시작
  - 여러개의 Spark Job 들로 클러스터 포화시 Dead-lock 발생 가능
  - 대규모 연산 시에는 Gang 스케쥴링을 지원하는 별도의 스케쥴러(kube-batch 등) 사용 필요
  - 아직 테스트 중…
- Executor의 request.cores를 얼마로?
  - K8S의 오버헤드를 감안한 CPU 자원을 지정해야 함
  - 다른 Workload와 중복되지 않는다면 Node CPU의 80% ~ 85% 수준으로 설정
  - 4 cores -> 3400m request.cores

Spark on Kubernetes 제품화의 문제들

로컬 디스크 사용으로 인한 성능 저하

- emptyDir: container들이 사용하는 Volume에 생성
  - K8S “emptyDir”만 위치를 지정할 수 있나요?
- tmpfs 사용: Mem 부족으로 인한 OOM 주의
  - 대용량의 RAM 필요
- Spark의 K8S Volume mount 기능으로 SSD 사용
  - RAM과 DISK의 사이
  - 3.0 현재 hostPath 로 mount만 가능: SSD 공간의 별도 관리가 필요
  - 3.1에서는 PVC 동적생성 가능: rancher/local-path-provisioner 사용 가능

Spark on Kubernetes 제품화의 문제들

별도의 사용자 (빅)데이터 저장 공간

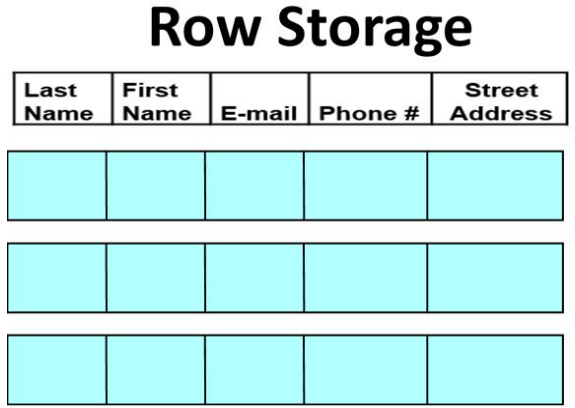
- HDFS-on-Kubernetes
  - HDFS의 구조적인 문제: JBOD 구성이 어려움
  - K8S 구성 시 Data Locality를 얻기 어려움(Datanode의 socket을 각각 mount 해야 함)
  - Public Cloud에서 구성 시, 비용에서 장점이 없음

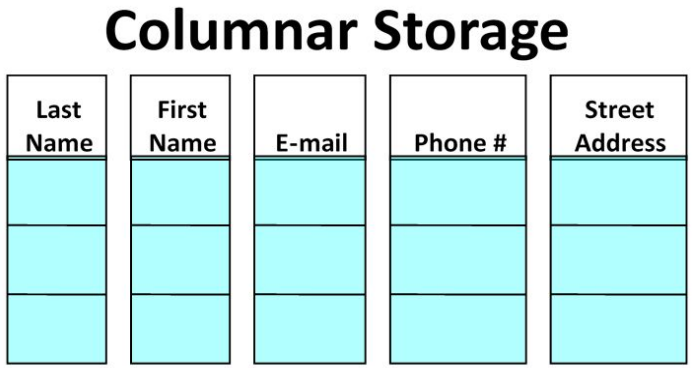
- S3 API 호환 저장소(minio, ceph rgw 등)
  - ceph은 구성이 복잡하나 rook operator로 설치 가능, minio는 구성이 비교적 단순
  - POSIX와는 다른 방식의 Rename 연산으로 인해 성능 저하 발생
  - Public Cloud는 대부분 S3 API 호환의 오브젝트 저장소 제공

Spark on Kubernetes 제품화의 문제들

파일/ 데이터 형식

- Columnar 형식(ORC, Parquet) 사용 필수
  - 최소한의 데이터를 읽어 들이기 위하여
  - Spark는 Predicate Push Down 능력이 좋은 편
- Databrick Delta, Apache Iceberg 등의 테이블 포맷 사용 고려
  - GDPR 등의 데이터 규제 대응 가능
  - Storage로의 I/O 부담 감소

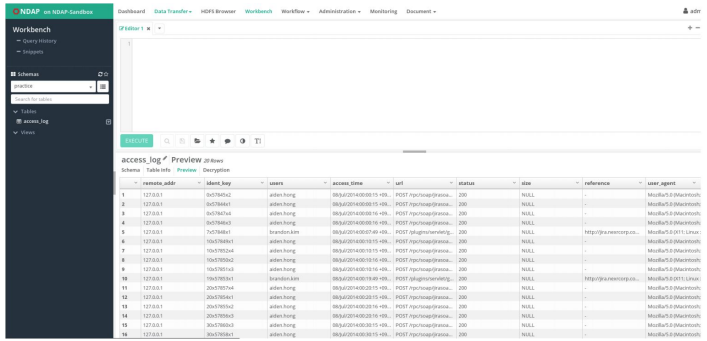




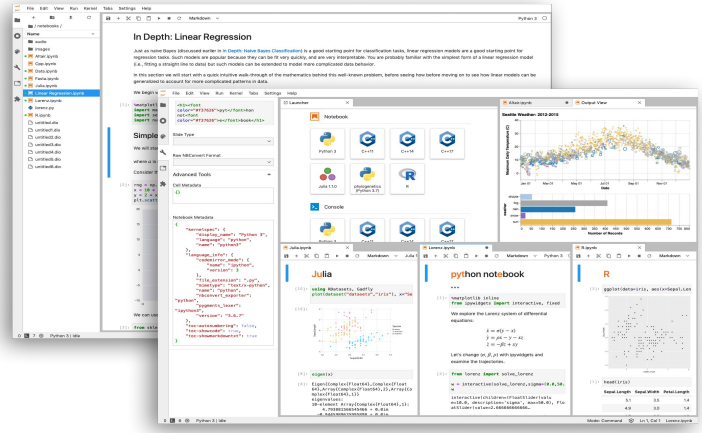
Spark on Kubernetes 제품화의 문제들

User Interface/eXperience

- Workbench 스타일
  - Good old friend
  - 주로 SQL-like
  - Data transform, DW



- Notebook 스타일
  - Jupyter, Zeppelin 등
  - 주로 Python, Interpreter
  - EDA, Science



마무리

마무리

- Apache Spark: 메모리 기반의 빅데이터 처리 엔진
- Spark on Kubernetes: 2018년부터 지원 시작, GA는 내년에 예정
  - Dynamic Allocation, Shuffle 제외
- Spark on Kubernetes 제품화
  - 어려움이 있으나 최신 버전에서는 극복하고 있음
  - 데이터 저장소: S3 API 호환의 오브젝트 저장소
  - 성능을 놓치지 않는 구성 필요
  - 제품의 성격에 맞는 선택과 집중

참고

- Spark: Cluster Computing with Working Sets, Matei Z. et al, UC Berkeley, HotCloud'10
- Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing, Matei Z. et al, UC Berkeley, NSDI'12
- Spark on Kubernetes, https://issues.apache.org/jira/browse/SPARK-18278
- Apache Spark on K8S Best Practice and Performance in the Cloud, Junping Du, Tencent, Spark+AI Summit NA 2019
- Running Apache Spark on Kubernetes: Best Practices and Pitfalls, Jean-Yves Stephan, Data Mechanics, Spark+AI Summit NA 2020
- A Thorough Comparison of Delta Lake, Iceberg and Hudi, Junjie Chen, Tencent, Spark+AI Summit NA 2020

We’re Hiring

http://ktnexr.com/about/job.html

모든 포지션 Open 중!