Day 1

WorldSkills Korea Training Camp
EMR on EKS

우 준 후

He/Him
Cloud Support Engineer Intern
Amazon Web Service

최 윤 진

She/Her
Cloud Support Engineer Intern
Amazon Web Service

목차

1. Amazon EMR
2. Amazon EKS
3. Karpenter
4. EMR on EKS
5. QnA

Amazon EMR

Amazon EMR이란?

(이전명칭: Amazon Elastic MapReduce)

Apache Hadoop 관련 애플리케이션(Hadoop 에코시스템)을
AWS에서 구축하여 빅데이터를 처리하고 분석하기 위한 서비스

Amazon EMR이란?

EMR의 주요 특징

- **클러스터 생성 및 관리 자동화**
- **다양한 빅데이터 도구 지원**
- Apache Spark, Hadoop, Hive, Pig 등
- **AWS 서비스와의 통합**
- S3, Glue, Lake Formation 연동
- **보안 및 모니터링**

Apache Spark

Apache Spark란?

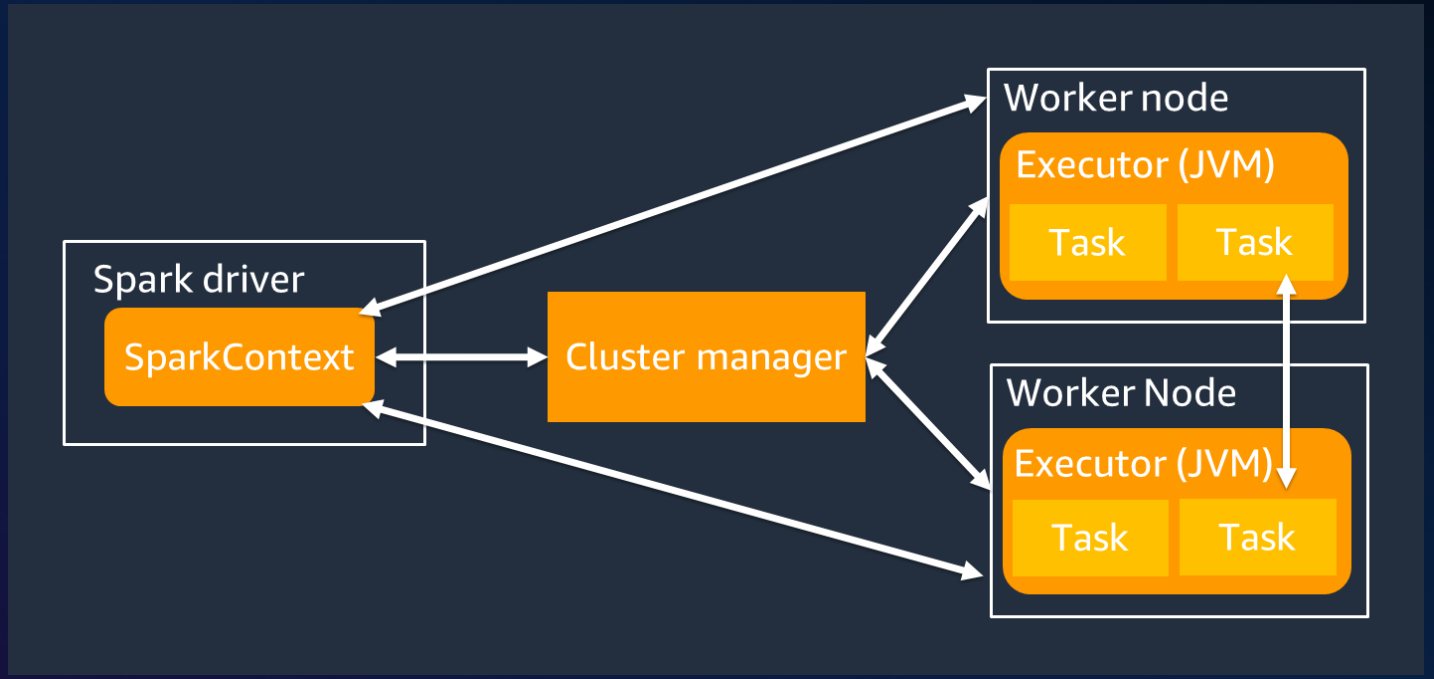
- EMR에서 가장 많이 사용되는 도구
- 대용량 데이터 처리를 위한 통합 분석 엔진
- 인메모리 처리로 빠른 성능
- SQL, 머신러닝, 그래프 처리 등 다양한 워크로드를 지원함
- Python, Java, Scala 등 다양한 프로그래밍 언어를 지원

Spark 아키텍처

Driver Program
- 애플리케이션의 main 함수 실행
- SparkContext 생성 및 관리

Cluster Manager
- 리소스 할당 및 관리

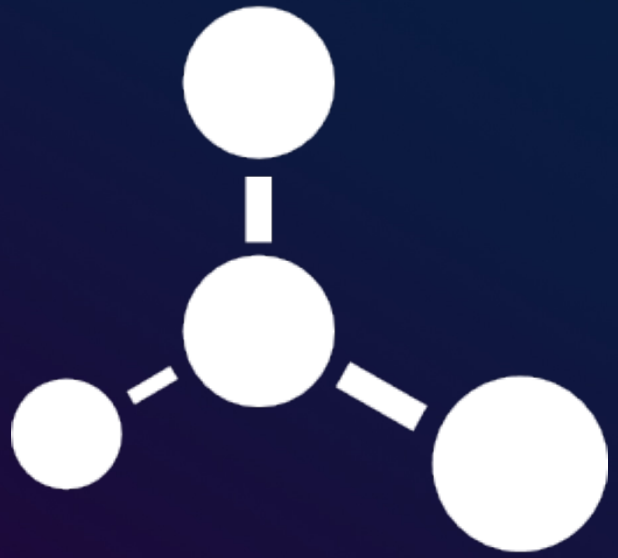
Executor
- 실제 작업 수행
- 데이터 저장 및 캐싱



Amazon EKS

Amazon EKS란?

- Amazon Elastic Kubernetes Service
- 완전 관리형 Kubernetes 서비스
- 다양한 컴퓨팅 옵션 지원
- 클라우드 및 온프레미스 환경 모두 지원



EKS 노드 관리

- 관리형 노드 그룹 생성
- 노드 자동 복구 기능
- 노드 상태 모니터링
- CLI 및 콘솔을 통한 관리

Karpenter란?

기존 노드 관리의 한계

- 인스턴스 타입 선택의 제한
- 용량 확보의 어려움
- 복잡한 자동화 정책

Karpenter의 등장

- 실시간 워크로드 기반 확장
- 자동 인스턴스 타입 선택
- 신속한 노드 프로비저닝

Karpenter란?

기존 Cluster Autoscaler와의 비교

- 노드 그룹 제한 없음
- 빠른 스케일링 속도
- 정확한 리소스 매칭

EMR 워크로드에서의 이점

- Spark 작업에 최적화된 노드 선택
- 리소스 사용률 향상
- 비용 최적화

가상 클러스터와 네임스페이스 활용

가상 클러스터란?

실제 EKS 클러스터의 특정 네임스페이스를 EMR 서비스에 등록한 것

기능:

EMR 서비스가 해당 네임스페이스 내에서 Apache Spark 작업을 실행할 수 있게 하는 논리적 연결 고리 역할

1. **다중 가상 클러스터 지원**: 하나의 EKS 클러스터에 여러 개의 가상 클러스터를 생성할 수 있습니다
2. **네임스페이스 연결**: 각 가상 클러스터는 EKS 클러스터의 특정 네임스페이스와 1:1로 연결됩니다
3. **논리적 추상화**: 물리적 EKS 클러스터와 EMR 서비스 간의 논리적 연결점을 제공합니다
4. **리소스 활용**: EMR 서비스가 EKS 클러스터의 컴퓨팅 리소스를 활용하여 Spark 작업을 실행할 수 있게 합니다

EKS 클러스터와의 관계

1. 리소스 분리 및 멀티 테넌시

- 여러 팀이나 워크로드를 네임스페이스로 분리
- 동일한 EKS 클러스터에서 여러 EMR 워크로드를 독립적으로 실행
- 각 팀별 격리된 환경 제공

```
EKS 클러스터
 ├── 네임스페이스 A (가상 클러스터 1) → 개발 팀
 ├── 네임스페이스 B (가상 클러스터 2) → 데이터 팀
 └── 네임스페이스 C (가상 클러스터 3) → ML 팀
```

2. 권한 관리 및 보안

- IAM 역할 분리
- **RBAC 적용**
- 서비스 계정 연결

EKS 클러스터와의 관계

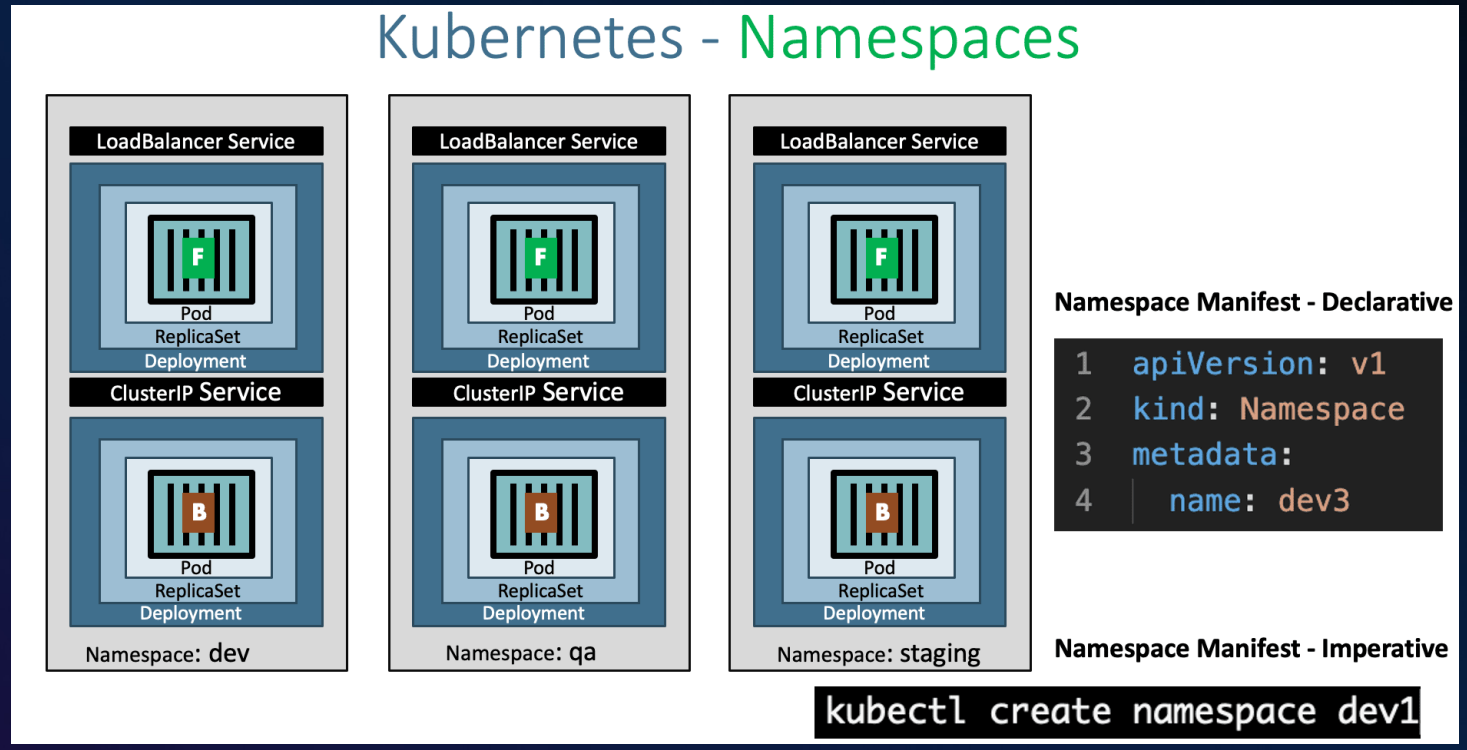
3. 리소스 할당 및 관리

- **네임스페이스별 리소스 쿼터**: CPU, 메모리, 스토리지 할당량 설정
- **노드 선택**: 특정 노드 그룹에 워크로드 배치 가능
- **리소스 격리**: 각 가상 클러스터 간 리소스 경합 방지

4. 운영 독립성

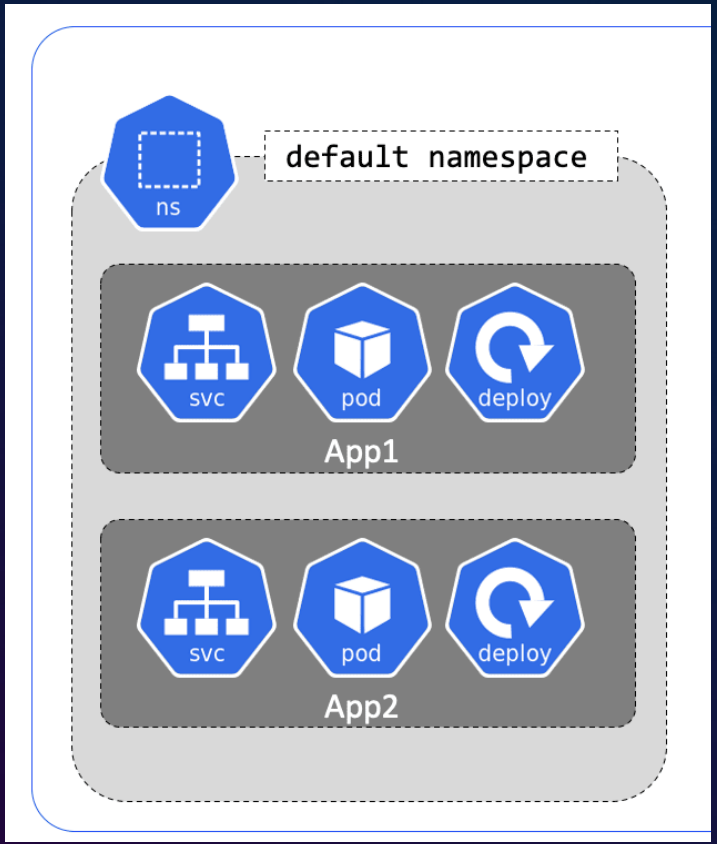
- **독립적 모니터링**: 각 가상 클러스터별 메트릭 및 로그 수집
- **개별 스케일링**: 워크로드별 독립적인 확장/축소
- **버전 관리**: 가상 클러스터별 EMR 릴리스 버전 선택 가능

EKS 클러스터 환경에서의 네임스페이스 개념



다양한 유형의 노드에서 실행되는 워크로드를 논리적으로 분리하는 핵심 메커니즘

Kubernetes 기본 네임스페이스



- **default:** 기본 네임스페이스
- **kube-system:** Kubernetes 시스템 컴포넌트용
- **kube-public:** 모든 사용자가 접근 가능한 공용 네임스페이스
- **kube-node-lease:** 노드 하트비트 정보 저장

EKS 클러스터 환경에서의 네임스페이스 개념

다양한 유형의 노드에서 실행되는 워크로드를 논리적으로 분리하는 핵심 메커니즘

EKS 클러스터는 다음과 같은 컴퓨팅 옵션을 지원

- EKS Auto Mode 관리형 노드
- 자체 관리형 노드
- Amazon EKS 관리형 노드 그룹
- AWS Fargate
- **Amazon EKS 하이브리드 노드**

하이브리드 환경에서의 네임스페이스 전략

분리 전략 옵션:

- **지리적 분리**: 클라우드와 온프레미스 워크로드를 별도 네임스페이스로 관리
- **환경별 분리**: 개발, 스테이징, 프로덕션 환경을 네임스페이스로 구분
- **팀별 분리**: 각 개발팀이나 프로젝트별로 독립적인 네임스페이스 할당

하이브리드 환경에서의 네임스페이스 전략

다양한 애드온을 지원:

- Kubernetes Metrics Server
- cert-manager
- Prometheus Node Exporter
- kube-state-metrics

EMR on EKS

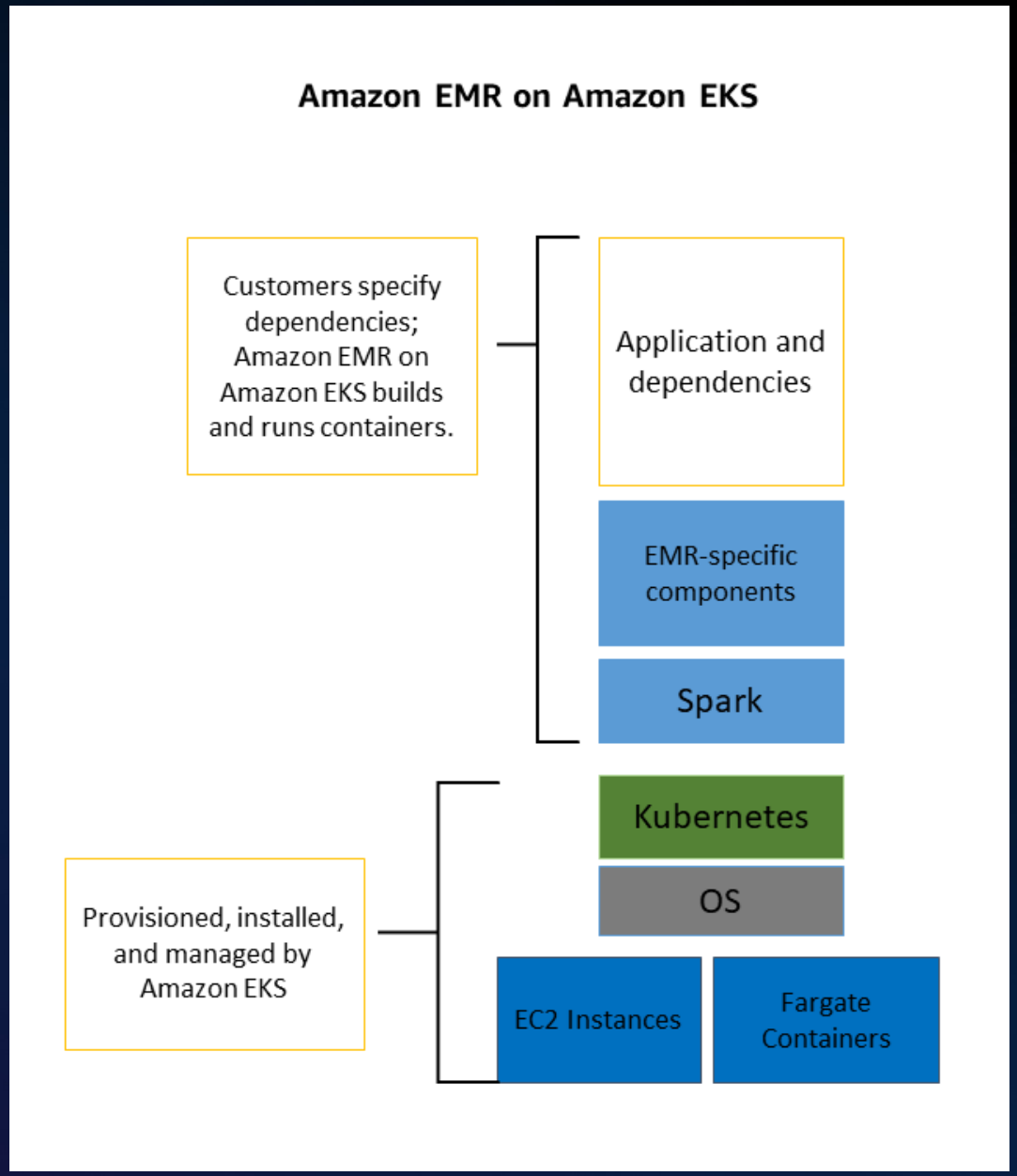
EMR on EKS란?

1.통합된 환경

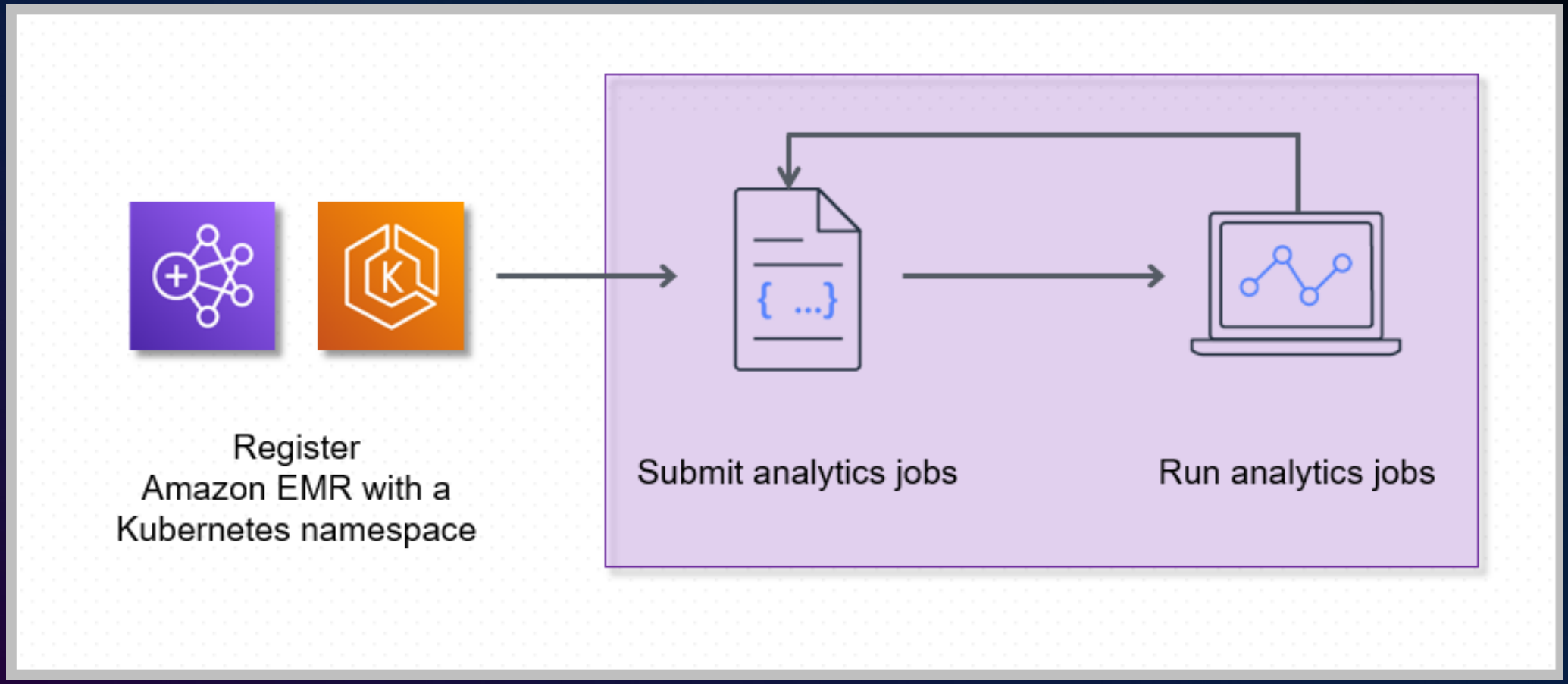
2.리소스 최적화

3.자동화된 관리

4.유연한 컴퓨팅 옵션



EMR on EKS 가상 클러스터에 작업을 제출하면 어떻게 되나요?



Thank you!