AWS Essentials

4. 클라우드 모범사례

Tacademy

CONTENTS

학습 목표

- 클라우드 컴퓨팅의 모범 사례들에 대한 개념을 이해할 수 있습니다.
- 장애를 고려한 디자인, 컴포넌트의 소결합에 대한 개념을 이해할 수 있습니다.
- 탄력성 및 동시성을 가진 서비스의 구현과 보안을 고려한 구성 방안에 대해 이해할 수 있습니다.

1. 장애를 고려한 디자인

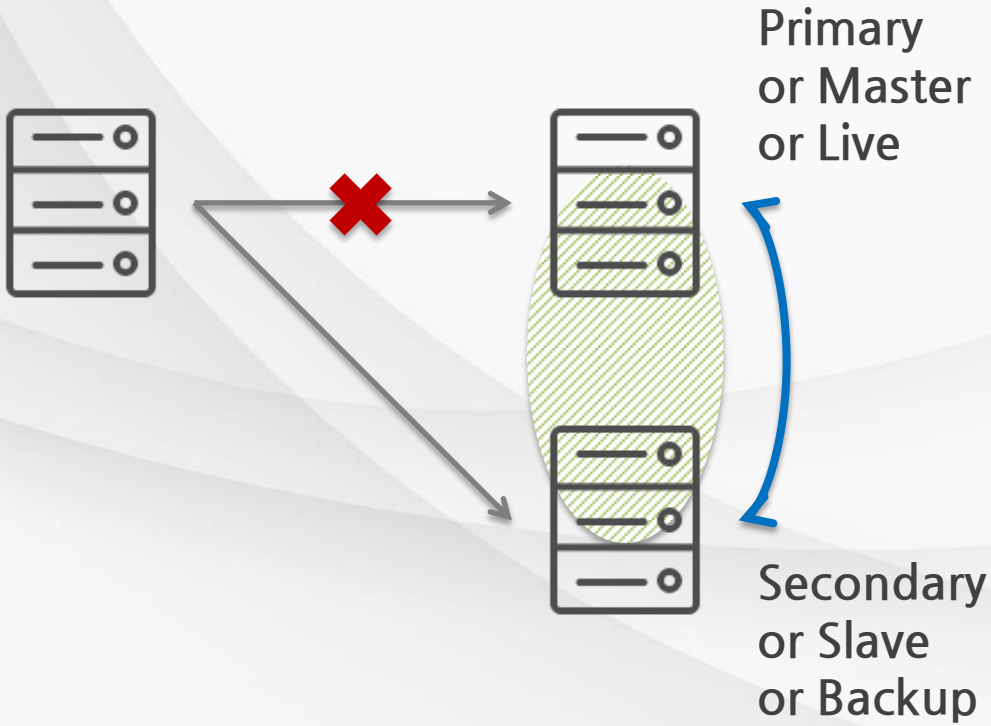
장애를 고려한 디자인 개요

클라우드 환경에서 설계를 할 때 **장애가 발생 한다는 가정** 하에 보수적으로 설계를 진행하고, 장애로부터 자동으로 **복구, 복원**이 가능하도록 시스템을 디자인 한다.

- 하드웨어의 장애가 발생하더라도 어플리케이션이나 서비스 레벨의 장애로 직결되지 않도록 단일 부분의 **장애 요소(SPOF)**들을 제거한다.

※ SPOF(Single Point Of Failure): 단일 장애 요소, 시스템 단일 구성 요소가 동작하지 않았을 때 전체 시스템에 영향을 주는 요소

서비스의 이중화 사례



대표적인 정책 요소

- 일관성 있는 백업과 복구 정책과 자동화
- 재기동 이후에 진행되는 프로세스의 정립
- 데이터를 리로드 하여 다시 동기화 할수 있는 상태 허용
- 이미지에 표준화된 사전 설정(pre-config), 최적화 정책 관리
- 인메모리 세션이나 유저 정보 상태를 저장하는 방식을 지양

2. 컴포넌트에 대한 소결합

컴포넌트에 대한 소결합 개요

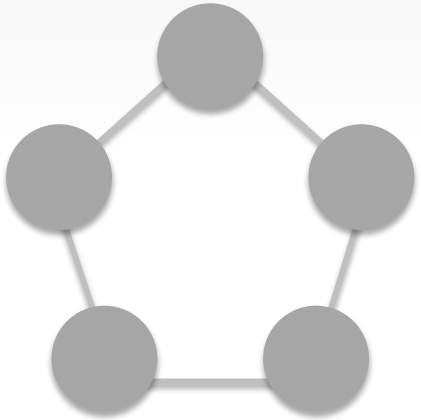
클라우드에서는 SOA의 디자인 원칙 중에 하나인 **"더 큰 확장을 위해서 시스템의 구성요소들을 보다 소결합 시킨다"** 를 따르는 설계 방식을 가져간다.

다양한 요소들을 구성할 때 서로간에 **밀결합(tight coupled)을 지양**하여 하나의 구성요소에 문제가 발생하였을 때 다른 구성요소는 해당 문제와 상관없이 **지속적으로 서비스 가능해야 한다**는 개념이다.

소결합 VS 밀결합

Loose Coupled

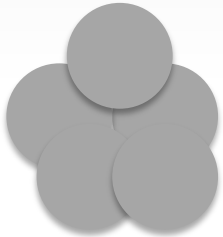
(단계별 독립된 방식)



VS

Tight Coupled

(절차지향적인 방식)



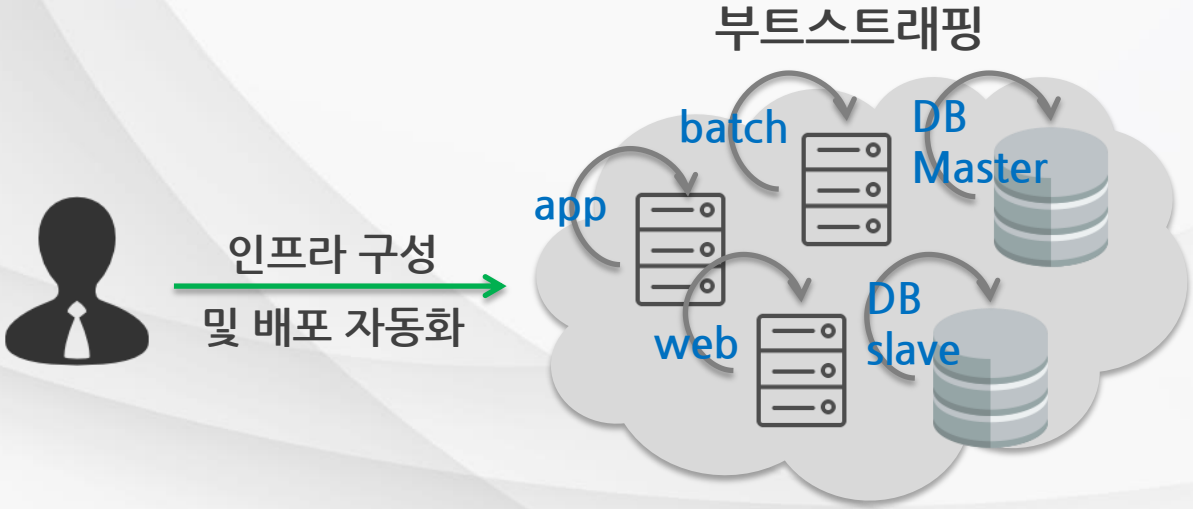
3. 탄력성/동시성/보안을 고려한 구성

탄력성/동시성 있는 구성

탄력성 있는 구성	동시성 있는 구성
◉ 탄력성을 구현하기 위해 배포 절차를 자동화하고 구성을 간소화하는 프로세스 구성 필요 ◉ 사용자의 개입 없이 확장 필요 ◉ 실 사용량에 부합되는 자원을 제공 ◉ 자원의 효율성과 비용 효율성을 가져 온다.	◉ 클라우드에서는 동시성을 구현하는데 어렵지 않음 - 예를 들어, 동일한 시점에 데이터를 저장하면서 요청을 수행하는 작업을 실행할 수 있음 ◉ 클라우드는 작업을 병렬화 하고 자동화 할 수 있음

탄력성 있는 구성

❖ 탄력적인 구성 방법 사례



※ 부트스트랩(Bootstrap) :
자체 동작에 의해서 어떤 소정의 상태로 이행하도록 설정하는 방법

동시성 있는 구성

❖ 작업의 병렬적인 구성 방법 사례

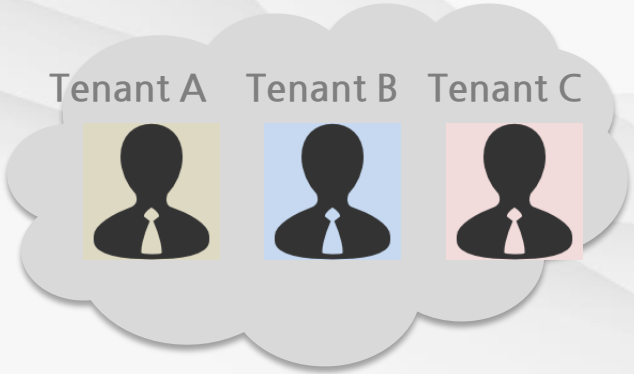
보안을 고려한 구성

보안은 모든 어플리케이션 계층에 적용되어야 하는 필수적인 조건이며 특히 클라우드와 같은 **멀티테넌시 환경에서는 보안이 가장 중요한 고려사항**이다.

- 사용자가 네트워크 구성과 어플리케이션 레벨의 보안 책임이 있으며 다양한 보안 사례 및 툴을 고려하여 보안 레벨을 높이고 서비스의 보안 요구사항에 맞게 구성해야 한다.

보안을 고려한 구성

보안은 모든 어플리케이션 계층에 적용되어야 하는 필수적인 조건이며 특히 클라우드와 같은 **멀티테넌시 환경에서는 보안이 가장 중요한 고려사항**이다.



※ Multi-tenancy:

특정 단위의 사용자들이 특정 자원들 혹은 어플리케이션들의 물리적인 자원을 공유하며 서로의 환경에 대해 논리적으로 분리하고 접근을 제어하며 사용하는 구성 환경

보안을 고려한 구성

❖ 기본적인 보안 모범 사례

- 데이터 전송시에 정보 보호
- 저장에 대한 정보 보호
- 어플리케이션의 보호
- 클라우드 자원의 보호
- 다수의 사용자 혹은 권한에 대한 보호

학습정리

지금까지 [클라우드 모범사례]에 대해서 살펴보았습니다.

장애를 고려한 디자인

장애로부터 자동으로 **복구·복원**이 가능하도록 시스템을 디자인 하며, 어플리케이션이나 서비스 레벨 단일 부분의 **장애 요소(SPOF)들을 제거**한다.

컴포넌트에 대한 소결합

서비스의 높은 확장성을 위해 하나의 구성요소에 문제가 발생했을 때 다른 구성요소는 그 문제와 상관없이 지속적으로 서비스가 가능하도록 **소결합하여 설계**한다.

탄력성/동시성/보안을 고려한 구성

- **탄력성**을 통해 자원의 효율성과 비용의 효율성을 극대화 시키며 **동시성** 있는 시스템 설계를 통해 작업의 물리적인 소요 시간을 단축하고 자동화 한다.
- 클라우드 환경의 다양한 보안 사례 및 툴을 고려 하여 **보안 레벨**을 높이고 서비스의 **보안 요구사항**에 맞게 구성 한다.