



심층 분석: Well-Architected 설계 원칙

학습 내용

강의의 핵심

배울 내용은 다음과 같습니다.

- AWS Well-Architected Framework의 설계 원칙 식별하기
- 주요 Well-Architected 설계 원칙을 자세히 설명하기



Well-Architected 설계 원칙

Well-Architected Framework는 클라우드에서 우수한 설계를 쉽게 구현하는 데 도움이 되는 일반적인 설계 원칙 세트를 제공합니다.

- 필요 용량을 추측할 필요가 없습니다.
- 프로덕션 규모로 시스템을 테스트합니다.
- 자동화로 아키텍처 관련 실험이 더욱 쉬워집니다.
- 혁신적인 아키텍처를 허용합니다.
- 데이터를 사용하여 아키텍처를 구동합니다.
- 실전을 통해 개선합니다.



필요 용량에 대한 추측 불필요

기존 환경

- 시스템을 배포하기 전에 용량을 결정하면 값비싼 유향 리소스를 낭비할 수 있습니다.
- 제한된 용량의 성능 영향을 처리해야 할 수도 있습니다.



클라우드 환경

- 필요한 인프라 용량을 추측할 필요가 없습니다.
- 많은 적은 용량을 필요한 만큼만 사용하고, 자동으로 확장 및 축소할 수 있습니다.



프로덕션 규모로 시스템을 테스트

기존 환경

- 테스트만을 위해 중복 환경을 생성하려면 보통 엄청난 비용이 듭니다.
- 대부분의 테스트 환경은 프로덕션 수요의 실제 수준에서 테스트되지 않습니다.

클라우드 환경

- 필요에 따라 중복 환경을 생성하고, 테스트를 완료한 후, 리소스를 폐기할 수 있습니다.
- 실행 중일 때만 테스트 환경에 대한 비용을 지불하므로, 온프레미스에서 테스트하는 비용의 극히 일부만으로 실제 환경을 시뮬레이션할 수 있습니다.



자동화

기존 환경

- 온프레미스 환경은 자동화하려면 추가 작업이 필요한 개별 구조 및 구성 요소를 가지고 있습니다(모든 인프라 부분에 사용할 수 있는 공통 API가 없음).

클라우드 환경

- 저렴한 비용으로 시스템을 생성 및 복제할 수 있습니다(수동 작업 없음).
- 자동화의 변경 사항을 추적하고, 영향을 감사하며, 필요하면 이전 파라미터로 되돌릴 수 있습니다.



혁신적인 아키텍처 허용

기존 환경

- 아키텍처 관련 결정은 대부분 정적인 일회성 이벤트로 구현됩니다.
- 시스템 수명 동안 시스템의 메이저 버전은 몇 개밖에 되지 않을 것입니다.
- 비즈니스가 변화하면서, 초기 의사 결정은 변화하는 비즈니스 요구 사항을 충족하는데 방해가 될 수 있습니다.

클라우드 환경

- 필요에 따라 자동화 및 테스트할 수 있는 능력이 설계 변경에 따른 위험을 낮춰줄 수 있습니다.
- 시간이 지나면서 시스템이 진화할 수 있으므로 비즈니스에서는 새로운 혁신을 표준 사례로 활용할 수 있습니다.



데이터를 사용하여 아키텍처를 구동

기존 환경

- 아키텍처 결정은 종종 조직의 기본값에 따라 선택됩니다.
- 일반적으로 데이터 세트를 생성할 수 없습니다.
- 아키텍처의 크기를 결정하기 위한 모델 및 가정이 사용될 것입니다.

클라우드 환경

- 아키텍처 선택이 워크로드의 동작에 미치는 영향에 대한 데이터를 수집합니다.
- 워크로드 개선 방법에 대한 사실에 기반한 결정을 내립니다.
- 이 데이터를 사용하여 시간이 지남에 따라 아키텍처 선택 및 개선 사항을 파악합니다.



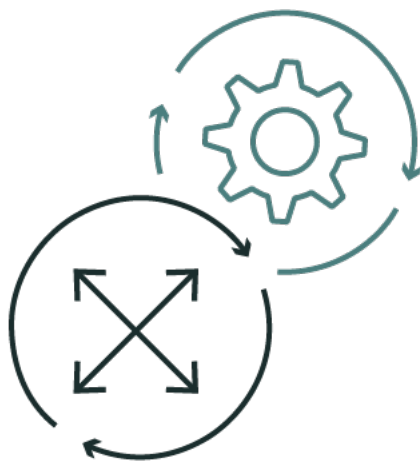
실전을 통해 개선

기존 환경

- 프로덕션에서 문제가 발생했을 때만 런북을 실행할 것입니다.

클라우드 환경

- 프로덕션 환경에서 이벤트를 시뮬레이션하기 위해 실전 일정을 예약하여 아키텍처와 프로세스가 어떻게 작동하는지 테스트합니다.



핵심 사항



AWS Well-Architected Framework에서 추진하는 주요 설계 원칙은 다음과 같습니다.

- 필요 용량을 추측하지 않습니다.
- 프로덕션 규모로 시스템을 테스트합니다.
- 자동화로 아키텍처 관련 실험이 더욱 쉬워집니다.
- 데이터를 사용하여 혁신을 허용합니다.
- 데이터를 사용하여 아키텍처를 구동합니다.
- 실전을 통해 개선합니다.