



Module 4 - AWS architecture

🕒 Created At	@Dec 17, 2020 4:21 PM
👤 Person	👤 송 용 석
🏷️ Tags	
🕒 Updated At	@Dec 23, 2020 9:19 AM
👤 마지막 수정	👤 민 경 환
📅 발표일	
👤 발표자	
📁 분류	스터디
👤 작성자	👤 송 용 석
👤 참석 인원	
🏷️ 태그	AWS Cloud Practitioner Essentials
🏷️ 프로젝트	

AWS Well-Architected Framework



안정적이고 안전하며 효율적이고 경제적인 시스템을 설계하고 운영하기 위한 설계 모범 사례를 제공

워크로드 검토를 위한 무료 서비스: AWS WA Tool

AWS Well-Architected Tool - Amazon Web Services

AWS Well-Architected Tool은 워크로드 상태를 검토하고 AWS 아키텍처 모범 사례와 비교하는 데 도움이 됩니다. 이 도구는 클라우드 아키텍처가 안전하고, 성능이 뛰어나며, 복원력을 갖춘, 효율적인 애플리케이션 아키텍처를 구축할 수 있도록 개발된 AWS Well-Architected Framework 를 기반으로 합니다.

🔗 <https://aws.amazon.com/ko/well-architected-tool/?whats-new-cards.sort-by=item.additionalFields.postDateTime&whats-new-cards.sort-order=desc>



The screenshot displays the AWS Well-Architected Tool interface. The top navigation bar includes the AWS logo, 'Services', and a search bar. The left sidebar shows the 'Well-Architected Tool' menu with options like 'Dashboard', 'Workload invitations', 'Workloads', and 'musma-general' (selected). The main content area shows the 'musma-general' workload overview. A notification banner indicates 'Lens version not current' with a 'View available upgrades' button. Below this, tabs for 'Overview', 'Milestones', 'Properties', and 'Shares' are visible. The 'Overview' tab is active, showing 'Workload overview' details: 'Last updated' on Aug 19, 2020 at 12:31 PM UTC+9, 'Questions answered' 4/46, and 'Risks' (2 High risk, 0 Medium risk).

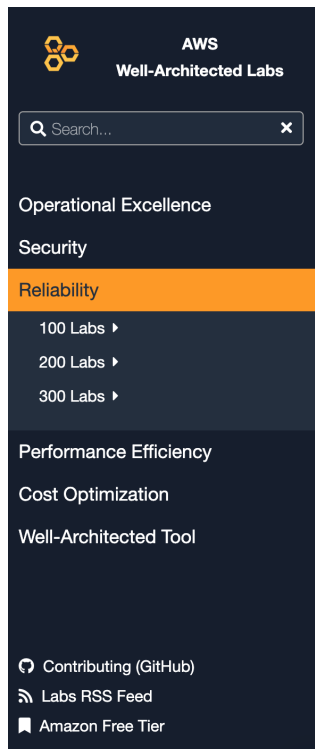
모범 사례를 구현한 실무 경험이 포함된 설명서


AWS Well-Architected Labs :: AWS Well-Architected Labs

The Well-Architected framework has been developed to help cloud architects build the most secure, high-performing, resilient, and efficient infrastructure possible for their applications. This framework provides a consistent approach for customers and partners to evaluate

 <https://wellarchitectedlabs.com/?ref=wellarchitected-wp>





 | [AWS Well-Architected Labs](#) > Reliability

Reliability Labs

Reliability Labs by Level

- [100 Labs](#)
 - [Level 100: Deploy a Reliable Multi-tier Infrastructure using CloudFormation](#)
- [200 Labs](#)
 - [Level 200: Implementing Bi-Directional Cross-Region Replication \(CRR\) for Amazon Simple S](#)
 - [Level 200: Deploy and Update CloudFormation](#)
 - [Level 200: Testing Backup and Restore of Data](#)
 - [Level 200: Testing for Resiliency of EC2 instances](#)
- [300 Labs](#)
 - [Level 300: Implementing Health Checks and Managing Dependencies to improve Reliability](#)
 - [Level 300: Testing for Resiliency of EC2, RDS, and AZ](#)
 - [Level 300: Fault Isolation with Shuffle Sharding](#)

Reliability Labs by tag (topic)

- [implement_change](#)
 - [Level 100: Deploy a Reliable Multi-tier Infrastructure using CloudFormation](#)
 - [Level 200: Deploy and Update CloudFormation](#)
- [data_backup](#)
 - [Level 200: Implementing Bi-Directional Cross-Region Replication \(CRR\) for Amazon Simple S](#)

The Five Pillars of the WA Framework



다섯 가지 필라를 하나씩 살펴볼 텐데, 이를 모두 꿰뚫는 핵심 키워드는 **자동화(automation)**임.

온-프레미스가 아닌 클라우드인 만큼, 가능한 한 모든 부분에서 자동화 가능한 기능들을 적극 활용함으로써 이익을 극대화

표 1. AWS Well-Architected 프레임워크 부문

이름	설명
운영 우수성	효과적인 개발 및 워크로드 실행을 지원하고, 작업에 대한 인사이트를 얻고, 지원 프로세스 및 절차를 지속적으로 개선하여 비즈니스 가치를 제공할 수 있는 능력입니다.
보안	보안 원칙에는 클라우드 기술을 활용하여 보안을 강화하고 데이터, 시스템 및 자산을 보호하는 능력이 포함됩니다.
안정성	워크로드의 의도한 기능이 수행될 것으로 예상되는 시기에 이 기능을 올바르게 일관적으로 수행하는 기능을 말하며 총 수명 주기에 걸쳐 워크로드를 운영 및 테스트할 수 있는 기능이 포함됩니다.
성능 효율성	컴퓨팅 리소스를 시스템 요구 사항에 맞게 효율적으로 사용하고, 수요 변화 및 기술 변화에 따라 이러한 효율성을 유지하는 기능이 포함됩니다.
비용 최적화	시스템을 실행하여 최저 가격으로 비즈니스 가치를 제공할 수 있는 기능이 포함됩니다.

1. 운영 우수성



효과적인 개발 및 워크로드 실행을 지원하고, 작업에 대한 인사이트를 얻고, 지원 프로세스 및 절차를 지속적으로 개선하여 비즈니스 가치를 제공할 수 있는 능력

- 백서

<https://d1.awsstatic.com/whitepapers/architecture/AWS-Operational-Excellence-Pillar.pdf?ref=wellarchitected-ws>

- 설계 원칙

1. 코드를 통한 운영: 최대한 코드로 구현(e.g. IaC) → 자동화 → 인적 오류(human error) 제한
2. 되돌릴 수 있는 사소한 내용 적용: 실패할 경우 되돌릴 수 있는 작은 증분으로 변경 내용 적용
3. 운영 절차를 수시 정제
4. 실패 예측: 사전 분석 → 실패 시나리오 테스트
5. 모든 운영상 실패로부터 학습: ~~아제 실패로부터는 그만 배우고 성공을 통해서도 배우도록 하자~~

2. 보안



데이터, 시스템 및 자산을 보호

- 백서

<https://d0.awsstatic.com/whitepapers/architecture/AWS-Security-Pillar.pdf?ref=wellarchitected-wp>

- 설계 원칙

1. 강력한 자격 증명 기반 구현: 권한 최소화
2. 추적 기능 활성화: 시스템 모니터링 (지표 수집) → 알림 전송을 자동화
3. 모든 계층에 보안 적용: 네트워크 엣지, VPC, 로드 밸런싱, 모든 인스턴스 및 컴퓨팅 서비스, 운영 체제, 애플리케이션, 코드
4. 모범 사례 자동 적용: 자동화된 소프트웨어 기반의 보안 메커니즘
5. 전송 및 보관 중인 데이터 보호: 암호화, 토큰화 및 액세스 제어
6. 사람들이 데이터에 쉽게 액세스할 수 없도록 유지: 데이터의 직접 액세스 또는 수동 처리의 필요성을 줄임
7. 보안 이벤트에 대비: 조직의 요구 사항에 부합하는 인시던트 관리 및 조사 정책과 프로세스를 마련하여 인시던트에 대비

3. 안정성



의도한 기능이 수행될 것으로 예상되는 시기에 올바르게 일관적으로 수행하는 기능

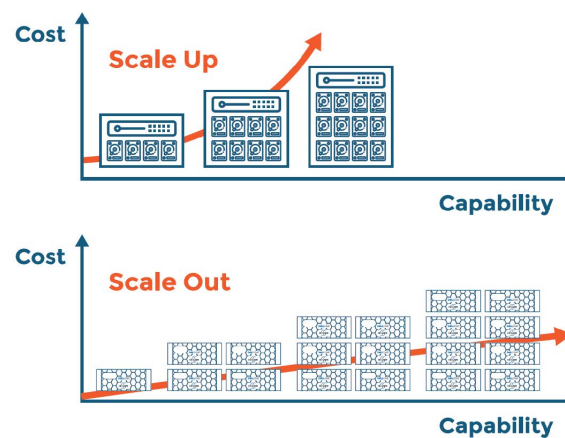
- 백서

<https://d0.awsstatic.com/whitepapers/architecture/AWS-Reliability-Pillar.pdf?ref=wellarchitected-wp>

- 설계 원칙

1. 장애 자동 복구: 장애 추적 및 자동 알림을 지원. 보다 정교한 자동화를 구현할 경우 장애가 발생하기 전에 예측하여 해결하는 것도 가능.
2. 복구 절차 테스트: 자동화를 사용하여 다양한 장애를 시뮬레이션하거나 이전에 장애로 이어졌던 시나리오를 재현
3. 수평적 확장(Scale-Out)으로 워크로드 전체 가용성 증대: 단일의 큰 리소스를 다수의 작은 리소스로 대체하여 단일 장애(Single Point of Failure)가 전체 워크로드에 미치는 영향을 축소

Scale Up vs. Scale Out



4. 용량 추정 불필요: 사용량을 모니터링하고 리소스 추가 또는 제거를 자동화함으로써 프로비저닝 과다 또는 부족 현상 없이 최적의 수준으로 수요를 충족
5. 자동화 변경 사항 관리: 인프라 변경은 자동화를 통해 수행되어야 함. 추후에는 자동화 변경 이력을 관리(추적하고 검토)

4. 성능 효율성



컴퓨팅 리소스를 시스템 요구 사항에 맞게 효율적으로 사용

- 백서

<https://d0.awsstatic.com/whitepapers/architecture/AWS-Performance-Efficiency-Pillar.pdf?ref=wellarchitected-wp>

- 설계 원칙

1. 고급 기술의 대중화: 복잡한 태스크를 클라우드 공급업체(AWS)에 위임 → 고난도 신기술을 직접 구현하는 대신 서비스 형태로 사용하고, 제품 개발에 더 집중
2. 몇 분 만에 전 세계에 배포: 최소한의 비용으로 지연 시간을 줄이고 고객 경험을 개선

3. 서버리스 아키텍처 사용: 물리적 서버 관리로 인한 운영 부담 소거. 어쨌든 가능한 한 관리형 서비스 쓰자.
4. 실험 횟수 증가: 자동화할 수 있는 가상 리소스를 활용하며 여러 가지 인스턴스, 스토리지 또는 구성에 대한 비교 테스트를 신속하게 수행
5. 기계적 동조 고려: 클라우드 서비스가 어떻게 사용되는지 파악하고 항상 워크로드 목표에 가장 부합하는 기술 접근 방식을 사용.
 - (원문) Consider *mechanical sympathy*: Use the technology approach that aligns best with your goals.
 - Mechanical sympathy is when **you use a tool or system with an understanding of how it operates best.**

You don't have to be an engineer to be a racing driver, but you do have to have **Mechanical Sympathy**.

Jackie Stewart, racing driver

5. 비용 최적화



최저 가격으로 비즈니스 가치 제공

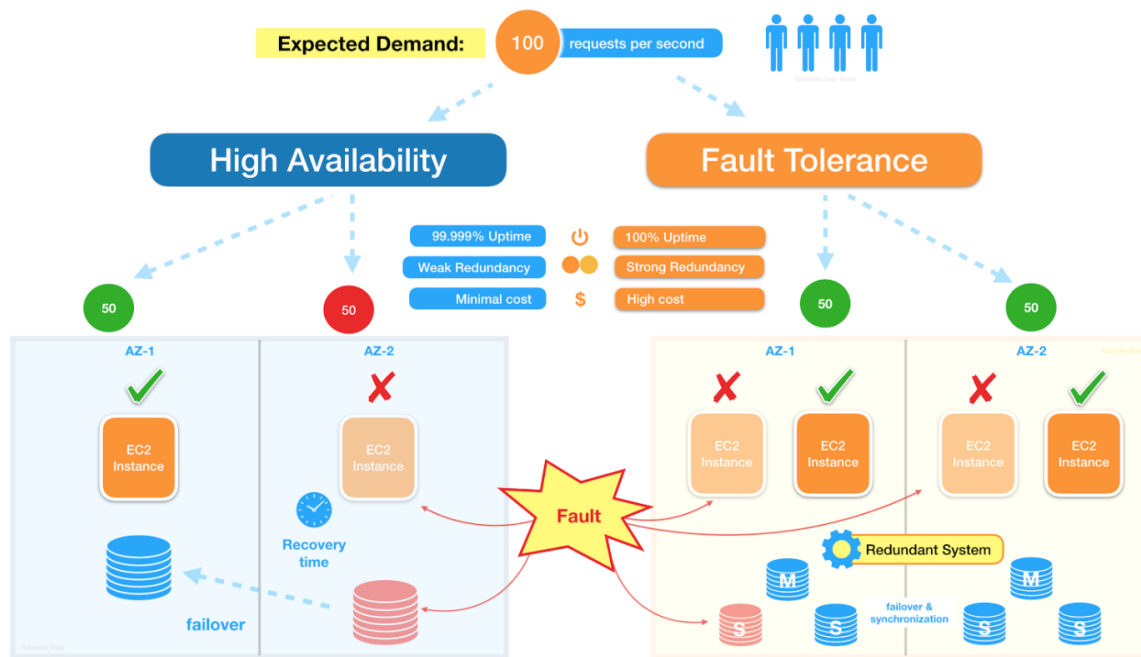
• 백서

<https://d0.awsstatic.com/whitepapers/architecture/AWS-Cost-Optimization-Pillar.pdf?ref=wellarchitected-wp>

• 설계 원칙

1. 클라우드 재무 관리 구현: 조직이 이 새로운 기술 및 사용 관리 영역에서 역량을 쌓기 위해서는 시간과 리소스를 할애해야 함. 보안 또는 운영 역량과 마찬가지로 지식 강화, 프로그램, 리소스 및 프로세스를 통해 역량을 쌓아 비용 효율적인 조직이 되어야.
2. 소비 모델 도입: 예를 들어 개발 및 테스트 환경은 주로 주중 근무일 중에 하루 8시간 동안만 사용. 사용되지 않는 동안 이러한 리소스를 중단.
3. 전반적인 효율성 측정: 워크로드의 비즈니스 결과 및 워크로드 제공과 관련된 비용을 측정
4. 획일적인 업무 부담에 대한 비용 지출 중단: 힘든 데이터 센터 운영 작업을 AWS가 처리 & 관리형 서비스를 통해 운영 부담 감소 → 고객과 비즈니스 프로젝트에 집중
5. 비용 분석 및 기여도 파악: 시스템의 사용량과 비용을 정확하게 식별 → ROI(투자 대비 수익률) 측정

Fault Tolerance & High Availability



내(耐)결함성: 결함을 견뎌내는 성질. 결함(fault) 또는 고장(failure)이 발생해도 정상적 또는 부분적으로 기능을 수행.

- 목적: 작동 가능 상태를 유지
- 구현: 구성 요소 중복 설계

고가용성: 쓸(用) 수(可) 있는 상태가 잘 유지된다.

- 목적: 작동 가능 상태를 유지
- 구현: 가동 중단을 최소화. fast recovery.

Web Hosting

- 비용 효율적 대안: on-demand provisioning 등
- 확장성: 예상치 못한 트래픽 처리 등
- 다양한 환경을 위한 솔루션: 사용자 트래픽 시뮬레이션 등

Additional materials

강의 4 - AWS 아키텍처 설계 (조재구 테크니컬 트레이너, AWS) :: AWSome Day 온라인 컨퍼런스 2018

Published on 강의 4 - AWS 아키텍처 설계 (조재구 테크니컬 트레이너, AWS) Well-Architected 프레임워크와 내결함성 및 고가용성, 그리고 웹호스팅을 위한 참조 아키텍처를 포함한 AWS 아키텍처에 대해 알아봅니다. 1. © 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved. 모듈 4: AWS 아키텍처 설계 2. © 2018, Amazon Web Services, <https://www.slideshare.net/awskorea/4-aws-aws-awsome-day-2018>

모듈 4: AWS 아키텍처 설계

AWS training and certification


 <https://www.aws.training/Details/Curriculum?id=42037>

References

https://d1.awsstatic.com/whitepapers/ko_KR/architecture/AWS_Well-Architected_Framework.pdf

AWS Well-Architected Tool - Amazon Web Services


AWS Well-Architected Tool은 워크로드 상태를 검토하고 AWS 아키텍처 모범 사례와 비교하는 데 도움이 됩니다. 이 도구는 클라우드 아키텍처가 안전하고, 성능이 뛰어나며, 복원력을 갖춘, 효율적인 애플리케이션 아키텍처를 구축할 수 있도록 개발된 AWS Well-Architected Framework 를 기반으로 합니다.

 <https://aws.amazon.com/ko/well-architected-tool/?whats-new-cards.sort-by=item.additionalFields.postDateTime&whats-new-cards.sort-order=desc>



AWS Well-Architected Labs :: AWS Well-Architected Labs


The Well-Architected framework has been developed to help cloud architects build the most secure, high-performing, resilient, and efficient infrastructure possible for their applications. This framework provides a consistent approach for customers and partners to evaluate

 <https://wellarchitectedlabs.com/>



The Five Pillars of the Framework

Creating a software system is a lot like constructing a building. If the foundation is not solid, structural problems can undermine the integrity and function of the building. When architecting technology solutions, if you neglect the five pillars of operational excellence, security, reliability, performance efficiency, and cost optimization, it can become challenging to build a system that delivers on your expectations and requirements.

 <https://docs.aws.amazon.com/wellarchitected/latest/framework/the-five-pillars-of-the-framework.html>

Conclusion

The AWS Well-Architected Framework provides architectural best practices across the pillars for designing and operating reliable, secure, efficient, and cost-effective systems in the cloud for IoT applications. The framework provides a set of questions that you can use to review an existing or proposed IoT architecture, and also a set of AWS best practices for each pillar.

 <https://docs.aws.amazon.com/wellarchitected/latest/iot-lens/conclusion.html>

Mechanical Sympathy

Mechanical sympathy is when you use a tool or system with an understanding of how it operates best. You don't have to be an engineer to be a racing driver, but you do have to have Mechanical Sympathy.

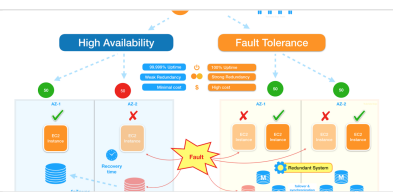
 <https://wa.aws.amazon.com/wat.concept.mechanical-sympathy.en.html>

AWS Well-Archi

High Availability vs. Fault Tolerance

Many IT Professionals nowadays do not have the right understanding of High Availability and Fault Tolerance, even those who are already in the IT industry for a long time. These two concepts are quite similar to each other, and sometimes, people use these two terms

<https://www.linkedin.com/pulse/high-availability-vs-fault-tolerance-jon-bonso/>



고가용성

High Availability; 高可用性 시스템에 장애가 생기더라도 빠르게 복구하는 성질 시스템을 장애 없이(중단 시간 없이) 오랫동안 가용한 상태로 운영할 수 있는 성질로 표현하기도 한다. 99%의 고가용성 시스템이란, 100시간 동안 운영 시 장애 시간이 1시간 이하임을 의미한다.

http://itwiki.kr/w/%EA%B3%A0%EA%B0%80%EC%9A%A9%EC%84%B1#.EC.9E.A5.EC.95.A0_ED.97.88.EC.9A.A9.28Fault_Tolerant.29.EC.99.80.EC.9D.98_EC.B0.A8.EC.9D.B4