

# 영구 컨테이너에 데이터 스토리지의 발전이 필요한 이유

백서

오늘날 기업들이 사용하는 마이크로서비스 지원 애플리케이션과 컨테이너화된 애플리케이션의 유형이 복잡해지면서 이런 컨테이너에 영구 스토리지를 사용해야 할 필요성도 더욱 커지고 있습니다.

이 백서에서는 단기적 컨테이너에서 영구 컨테이너로 어떻게 발전하였고, 개발자와 DevOps 팀이 모든 지역에서 (기존의 프로비저닝된 컨테이너와 서버리스 컨테이너를 비롯한) 컨테이너에 성능이 우수하고 저렴한 확장형 스토리지를 활용할 수 있는 방법에 대해 논의합니다.

## 컨테이너와 민첩성

조직에서 가상 머신 대신 컨테이너를 택하는 이유는 여러 가지가 있습니다. 우선 컨테이너는 빠르게 시작할 수 있고, 개발자가 하나의 컨테이너 이미지에 모든 코드 종속성을 패키지로 묶을 수 있습니다. 또한, 컨테이너는 가상 머신보다 시스템 리소스를 효율적으로 사용하고 이동이 간편합니다. 게다가 각 컨테이너에서 함수 또는 전체 애플리케이션의 독립적인 조각을 실행할 수 있기 때문에 마이크로서비스를 위한 완벽한 플랫폼을 제공하므로 민첩한 개발이 가능합니다. [IDC](#)

는 2023년까지 IT 조직의 70%가 민첩성을 개선하기 위해 컨테이너를 도입하게 될 것으로 예측한 것처럼 컨테이너 시장의 폭발적인 성장은 놀라운 일이 아닙니다.

컨테이너는 Unix chroot와 BSD jails의 루트에서 지금의 Docker로 발전하였으며, 그와 동시에 단기적인 개발 플랫폼에서 민첩하고 영구적인 애플리케이션을 위한 플랫폼으로 진화했습니다. 이런 발전 덕분에 최신 컨테이너 기술을 이용해서 새로운 애플리케이션을 구축하는 것이 일반화되었습니다. 하지만 컨테이너가 제공하는 민첩성을 활용하는 일부 애플리케이션의 경우, 데이터에 대한 접근성과 공유 기능을 향상해야 합니다. 영구 스토리지가 없다면 컨테이너는 상태 저장 워크로드, 개발 도구, 공유된 노트북 시스템에 필요한 높은 가용성과 내구성을 제공할 수 없게 됩니다.



Custom Media

## 컨테이너의 실행 위치

Nucleus Research의 최근 조사 결과에 따르면, 컨테이너 5개 중 4개가 클라우드에서 실행되고 그중 80%가 Amazon Web Services(AWS)에서 실행됩니다. 전체 Kubernetes 워크로드의 82%가 Amazon Web Services(AWS)에서 실행됩니다. 컨테이너에 관련해 AWS가 인기 있는 이유

먼저, 배포 옵션이 다양합니다. 조직에서는 Amazon Elastic Compute Cloud(Amazon EC2)를 사용하거나 두 개의 컨테이너 전용 오케스트레이션 도구(Amazon Elastic Container Service(Amazon ECS) 또는 Amazon Elastic Kubernetes Service(Amazon EKS)) 중에서 선택할 수 있습니다. 또한, AWS Fargate는 Amazon ECS와 Amazon EKS에 서버리스 컴퓨팅 엔진을 제공하고 모든 컨테이너화 애플리케이션에 동적으로 확장 가능한 컴퓨팅을 제공하여 컴퓨팅 인프라를 관리할 필요성을 완전히 제거하기 때문에 조직에서는 과도하게 프로비저닝하거나 사용하지 않은 리소스에 비용을 지불할 필요가 없습니다.

Amazon Elastic File System(Amazon EFS)과 마찬가지로 Amazon ECS와 Amazon EKS는 리전에서 실행되도록 아키텍처를 설계합니다. 즉, AWS 리전의 모든 가용 영역(AZ)에서 실행되므로 사실상 모든 환경에 가용성과 확장성을 지원합니다.

## 컨테이너에 영구 스토리지를 사용해야 하는 이유

조직들이 컨테이너 오케스트레이션 시스템에서 애플리케이션을 점차 표준화하면서 컨테이너에서 상태 저장 워크로드를 실행하는 경우도 늘어나고 있습니다. 이는 컨테이너 오케스트레이터가 영구 스토리지를 지원하지 못할 때 문제가 되는 경우가 많습니다. 영구 액세스가 불가능하더라도 최신 상태 저장 마이크로서비스에 방해가 되는 것은 아니지만, 상태 저장 애플리케이션이 시간에 따른 변경 사항을 추적하고 서로 데이터를 공유하는 데는 장애물이 됩니다.

데이터베이스, 웹 서버를 비롯한 여러 가지 애플리케이션이 스토리지 계층에서 데이터를 영구 저장할 것을 요구합니다. 여기에는 콘텐츠 관리 시스템과 같이 상태를 저장하고 장기 실행되는 애플리케이션이 포함됩니다. 이런 애플리케이션은 편집자가 새 콘텐츠를 게시하거나 관리자가 플러그인을 설치할 때마다 파일이 자주 변경됩니다. 또한, 기계 학습 프레임워크나 협업적 데이터 과학 작업 영역과 같이 서로 데이터를 공유해야 하는 애플리케이션도 포함되는데, 이들은 여러 가용 영역에 걸쳐 있을 가능성이 큼니다.

Amazon Elastic File System(Amazon EFS)과 마찬가지로 Amazon ECS와 Amazon EKS는 리전에서 실행되도록 아키텍처를 설계합니다. 즉, AWS 리전의 모든 가용 영역(AZ)에서 실행되므로 사실상 모든 환경에 가용성과 확장성을 지원합니다.

조직들이 점차 클라우드 네이티브로 옮겨가면서, IT와 사업 부서 모두 "단순히 작동하기만 하는" 애플리케이션을 벗어나 다음과 같은 애플리케이션을 요구하고 있습니다.

- 내구성
- 모든 프로덕션 요구 사항에 따라 확장 가능
- 수요 변동에 탄력적으로 적응
- 일상적인 운영 문제 해결 지원

이제 막 클라우드 여정을 시작한 기업들도 개발자, DevOps, 프로덕션 환경의 요구 사항을 따르면서도 컴퓨팅과 스토리지에 탄력성을 제공하려면 컨테이너가 필요하다는 것을 점차 깨닫고 있습니다.



기업들이 앱을 현대화하고 컨테이너화된 마이크로서비스 방식을 도입해서 관리 간접비를 절감하면서도 애플리케이션의 탄력성과 확장성을 높일 수는 있지만, 모든 애플리케이션이 마이크로서비스처럼 빠르게 다시 배포될 수 있는 것은 아닙니다. 이런 엔터프라이즈 애플리케이션은 컨테이너와 서버리스 컴퓨팅 아키텍처의 속도와 효율성을 따라가지 못합니다.

### 컨테이너를 위한 영구 스토리지:

#### 이상향

IT의 관점에서는 컨테이너화된 앱과 운영을 구축하여 매끄럽게 실행하는 능력만 있으면 됩니다. 안타깝게도 온프레미스에 머물며 고립된 상태에 있는 IT는 이러한 요구 사항에 부응할 수 있을 만큼 발전하지 못했습니다. 제한적 확장 능력, 관리 간접비, 끊임없는 온프레미스 기술 재생 사이클 등으로 인해 민첩성과 확장성을 발휘할 수 없습니다.

조직들이 컨테이너화된 워크로드에 영구적인 스토리지를 최대한 활용할 수 있으려면 다음과 같은 장점을 갖춘 솔루션이 필요합니다.

- 배포, 사용, 관리가 간편한 서비스형 셀프 서비스 설계

- 인프라에 대한 걱정 없이 한 번 설정으로 사용할 수 있는 단순성
- 정책 중심적 자동화
- 컨테이너화된 서버리스 플랫폼과의 호환성
- 성능, 중복성 및 규정 준수 문제에 대한 지역적 가용성 및 내구성
- 필요한 데이터 액세스만 제공하는 철저한 보안
- 모든 애플리케이션 인스턴스에서 상태 저장을 지원하는 기능
- 단일 확장 애플리케이션의 동시 인스턴스나 여러 애플리케이션 사이에서 간편하게 데이터 공유

#### 중요한 점

지금 상태에 머물 수는 없습니다. 애플리케이션을 현대화하고 개발 및 배포 방식을 민첩하게 바꾸지 않는 기업은 경쟁에서 불이익을 경험하게 됩니다. 또한, 다양한 애플리케이션 유형을 지원하려면 영구 스토리지가 필요합니다. 컨테이너에 마이크로서비스를 도입하지 못하거나 컨테이너 간의 상태 저장 트랜잭션을 지원하지 못한다면 현대화 노력을 지연시킬 수 있습니다.



게다가 데이터에 영구적으로 액세스하지 못할 경우, 데이터 사이언티스트, 데이터 분석 전문가 및 다른 사업 부서가 컨테이너화된 애플리케이션 워크로드에서 생성되는 데이터를 공유하고 공동 작업을 하는 자유를 누릴 수 없게 됩니다.

궁극적으로는 영구 스토리지에 대한 탄탄한 전략을 세우지 못한다면 디지털 혁신 노력을 저해해 시장에 대한 준비 상태에도 영향을 미칠 수 있습니다.

## 컨테이너를 위한 상태 저장 스토리지

좋은 방법이 있습니다. Amazon Elastic File System(Amazon EFS)는 상태 저장 컨테이너에 영구 스토리지를 제공하고 DevOps 입장까지 배려합니다. Amazon ECS와 Amazon EKS의 핵심적 특성을 똑같이 적용하여 설계한 Amazon EFS는 탄력적이고 지역적이며, Amazon EKS 및 Amazon ECS와 찰떡궁합처럼 매끄럽게 통합되고 파일 시스템을 다시 구성하지 않고도 탄력적으로 확장됩니다. Amazon EFS는 AWS Fargate와 함께 서버리스 컨테이너에 영구 스토리지를 제공합니다.

Amazon EFS는 모든 컨테이너 pod와 AZ에 일관적인 공유된 스토리지를 제공함으로써, 모든 컨테이너가 언제나 모든 파일의 최신 버전에 안전하게 액세스하도록 보장합니다. 완전관리형 서비스인 Amazon EFS는고가용성과 빠른 장애 조치를 제공합니다.

또한, AZ 전체에 컨테이너가 클러스터링된 경우에는 모든 AZ에서 동시에 공유된 볼륨에 액세스할 수 있습니다. 따라서 조직에서는 저렴한 스팟 요금으로 해당 AZ에서 애플리케이션을 실행하고, 스토리지 계층을 다시 구성할 필요 없이 비용을 절감할 수 있습니다. Amazon EFS는 지역적으로 설계되었기 때문에 컨테이너를 이동하거나 다시 시작할 경우 다른 AZ에서도 공유된 파일 시스템에 다시 연결됩니다.

## T-Mobile®

국제적인 통신 대기업 T-Mobile은 고객에게 제공하는 앱이 하루 사용 시간이나 연간 사용 기간(월) 측면에서 모두 사용량이 급증하는 바람에 어려움을 겪었습니다. 기존 인프라는 과도하게 프로비저닝하지 않고는 피크 수요를 지원하는 데 필요한 확장성을 지원할 수 없었습니다.

T-Mobile은 마이크로서비스 방식을 사용하도록 애플리케이션을 현대화하고 Amazon EFS를 사용하는 컨테이너를 배포하여 영구 스토리지와 스토리지 관리 간접비 없이도 동적인 확장이 가능한 기능을 제공했습니다. 그 덕분에 얻은 혜택은 분명했습니다.

- Amazon EFS로 16,000개 컨테이너 관리
- DIY에 비해 스토리지 비용 70% 절감, 스토리지 관리 간접비 절약
- 새로운 애플리케이션 서비스 배포 사이클 시간 개선

T-Mobile의 수석 아키텍트인 Amreth Chandrasehar는 "우리는 가용성과 성능에 대한 요구 사항이 다양한 애플리케이션을 사용하는 대기업입니다. EFS는 회사 전체적으로 이런 요구 사항을 충족하는 공통적인 스토리지 플랫폼을 제공해주었습니다."라고 말합니다.

Amreth Chandrasehar

선임 엔터프라이즈 아키텍트, T-Mobile

## Amazon EFS이 제공하는 혜택

AWS에서 컨테이너를 지원하는 모든 주요 구성 요소 (Amazon ECS, Amazon EKS, AWS Fargate 및 Amazon EFS)는 가용성이 높고, 확장 가능하며, 성능이 우수하기 때문에 개발자는 이런 기능을 설계할 필요가 없습니다. 이미 AWS 플랫폼에 내장되어 있습니다.

Amazon EFS는 영구 컨테이너를 도입하는 조직에 여러 가지 장점을 제공합니다. 먼저, 다른 DevOps 모범 사례에 맞게 컨테이너 스토리지 배포 및 관리에 민첩성을 제공합니다.

또한, Amazon EFS는 컨테이너 스토리지를 "민주화"합니다. 개발자는 이와 관련해서 OS, 스토리지 유형 또는 인프라와 관련된 그 무엇도 걱정하지 않아도 됩니다. IT와 사용자도 요구 사항 변화에 따라 확장, 성장 또는 계약이 가능하고 간접비가 전혀 없는 설계가 제공하는 혜택을 누릴 수 있습니다.

조직에는 지속적으로 컨테이너 스토리지를 최적화해서 최상의 성능을 유지할 수 있다는 장점을 제공하며, AWS Fargate와 통합하여 컨테이너와 영구 스토리지를 배포하는 동시에 서버, 클러스터, 인프라를 관리해야 할 필요가 없습니다. Amazon EFS 구성이 Amazon ECS 작업 정의에 포함되고 자동으로 연결되기 때문에 상태 저장 애플리케이션 개발과 데이터 공유가 간단하고 자동화됩니다.

조직에서는 데이터에 안전하게 액세스하기 위해 Amazon EFS Access Points를 사용할 수 있는데, 이는 AWS Identity and Access Management(IAM)와 통합되어 공유된 데이터에 간단히 액세스할 수 있도록 지원합니다. 그에 따라 세밀한 보안이 적용되어 애플리케이션, 사용자, 기타 마이크로서비스가 필요한 데이터에만 액세스하고 권한이 없는 애플리케이션은 데이터에 전혀 액세스할 수 없게 됩니다. 각 마이크로서비스는 자체적인 디렉터리가 있고 파일 시스템의 해당 부분에만 제한적, 또는 부분적으로 액세스할 수 있습니다. AWS Fargate를 사용하면 각

작업 또는 pod가 자체 커널에서 실행되어 워크로드가 분리되고 기본적으로 보안이 개선됩니다.

## 요약

기업의 컨테이너 여정은 상태를 저장하거나 저장하지 않는 모든 애플리케이션에 확대되고 있으며, 컨테이너를 애플리케이션을 개발하기 배포하는 현대적인 수단으로 바라보는 시각도 늘어나고 있습니다. 컨테이너를 새로운 표준으로 인정하는 사람도 많아졌습니다. 컨테이너로 배포되는 미션 크리티컬 애플리케이션이 늘어날수록 컨테이너에 영구 스토리지를 사용해야 할 필요성은 더욱 커집니다.

개발 팀, IT 및 운영 팀 모두 민첩한 방식으로 컨테이너를 배포하기를 원하며, AWS는 모든 지역에서 안전하고 내구성 있게 제공되는 스토리지와 컴퓨팅에 정책 기반 자동화와 운영이 필요 없는 관리를 적용합니다.

AWS 컨테이너 에코시스템은 Amazon EFS와 영구 스토리지 외에도 새로운 마이크로서비스 기반 앱을 기업에 통합하려는 개발 및 DevOps 노력을 지원합니다. Amazon EFS는 상태 비저장, 상태 저장, 서버리스, AWS Fargate를 사용한 완전 관리에 이르기까지 AWS 클라우드의 모든 컨테이너 환경을 자연스럽게 지원합니다.

---

Amazon EFS가 어떻게 컨테이너에 영구 스토리지를 간단히 제공하는지 알아보려면 [여기](#)를 클릭하세요.

---