쿠버네티스 기반 대화형 애플리케이션 스트리밍 아키텍처 설계

주정현, 손재기, 김동민*

*한국전자기술연구원 메디컬IT융합연구센터

jujeong0108@keti.re.kr, jgson@keti.re.kr, *dmkim@keti.re.kr

Design of a Kubernetes-based Application Streaming Architecture

Jeonghyeon Ju, Jaegi Son, Dongmin Kim

*Korea Electronics Technology Institute Medical IT Convergence Research Center

요 약

본 논문은 대화형 애플리케이션을 쿠버네티스 기반 클라우드 환경에서 컨테이너화하여 배포·스트리밍하고, 브라우저 및 전용 클라이언트에서 원격 조작이 가능하도록 하는 아키텍처를 설계 및 구현한다. 해당 아키텍처는 SteamCMD를 기반으로 클라우드 환경에서 다양한 Steam 컨텐츠 실행 표준 환경을 구축하고 이를 사용자 기기 환경에 구애받지 않는 스트리밍 환경을 구상하여 단말 이질성과 설치 및 업데이트 부담을 최소화하고, 다양한 환경에서의 접근성을 보장한다. 이러한 방식으로 재현할 수 있는 파이프라인과 설계 원칙과 운영 패턴을 체계화하여 클라우드 환경에서의 애플리케이션 스트리밍의 초석을 마련하는 것을 목표로 한다.

1. 서 론

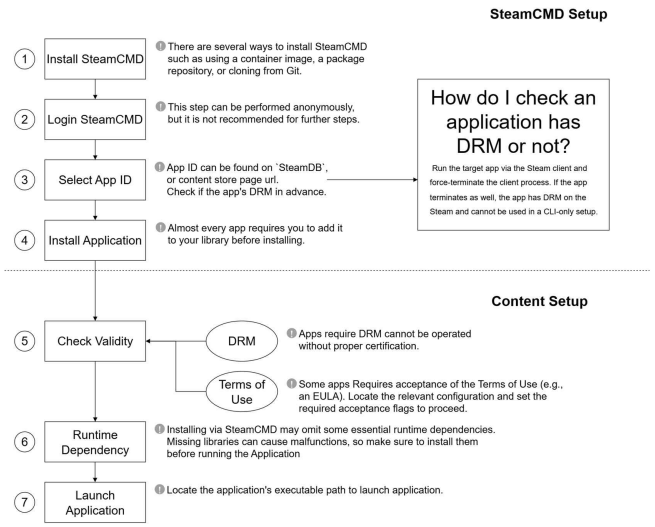
최근 사용자 단말의 사양과 관계없이 고성능 애플리케이션을 사용 가능하게 하는 원격·하이브리드 협업이 일반화되면서 원격 소프트웨어 사용은 다양한 분야에서 자리를 잡고 있다[1]. 이러한 배경에서 클라우드 서버 측 실행과 스트리밍은 실행 환경을 중앙에서 표준화하고, 사용자는 브라우저 혹은 경량 클라이언트만으로 접속하게 하는 방식으로, 접근성을 향상하고 사용자 디바이스에 대한 부담을 간소화한다. 이에 빌드-배포-운영 전 과정을 클라우드 네이티브 환경에서 구성하는 규격화된 설계를 갖추는 것은 원격 소프트웨어 사용 환경에서 필수적인 요소가 되었다[2].

이에 본 논문에서는 대화형 애플리케이션을 쿠버네티스 기반 클라우드 환경에서 표준화된 파이프라인으로 실행하도록 하는 참조 아키텍처를 구상하여 다양한 실행 환경에 일관되게 제공할 수 있는 스트리밍 서비스 구조를 설계하고자 한다. 이를 통해 재현 가능한 파이프라인과 설계 원칙 및 운영 패턴을 정식화하고, 참조 구성요소와 운영 가이드를 함께 제시함으로써 실무 적용 가능성을 뒷받침한다.

2. 본론

본 논문에서 '대화형 애플리케이션'은 사용자 입력에 즉시 반응하여 화면이 프레임 수준으로 갱신되는 소프트웨어(게임·시뮬레이션·VR 등)를 의미한다. 해당 설계는 이러한 대화형 애플리케이션의 특성을 폭넓게 포괄하는 서비스 제공 플랫폼인 Steam을 대상 풀로 삼고, 해당 플랫폼의 컨텐츠를 쿠버네티스 환경에서 구동이 가능하도록 하는 SteamCMD 콘솔 유틸리티를 활용한다.

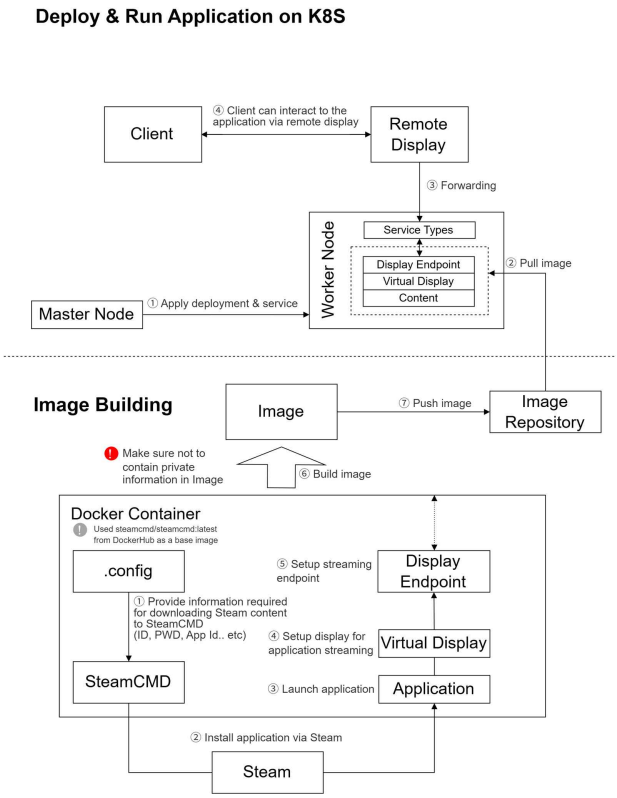
2-1. 컨텐츠 구성 절차



<그림 1> SteamCMD를 통한 컨텐츠 구성 절차

그림 1은 SteamCMD 활용에 필요한 요구사항과 컨텐츠 유효성 검증 등의 실행에 필요한 조건과 과정을 절차적으로 정리한 것이다. SteamCMD는 비대화형 설치를 지원하는 콘솔 유틸리티이므로, 본 설계에서는 SteamCMD 전용 컨테이너 이미지를 활용하여 환경 구성을 표준화하고 재현성을 확보하였다. 대상 Steam 컨텐츠 선택은 식별자와 메타데이터를 바탕으로 이루어지며, 이를 통해 실행 애플리케이션을 결정한다. 본 설계에서 컨텐츠 접근은 합법적 권한을 전제로 한다. 따라서 플랫폼의 권리 관리(DRM) 및 계정 기반 라이선스 정책을 충족하지 않는 항목은 실행 대상에서 제외하고, 기술적 실행 요건을 충족하는 컨텐츠만을 다룬다.

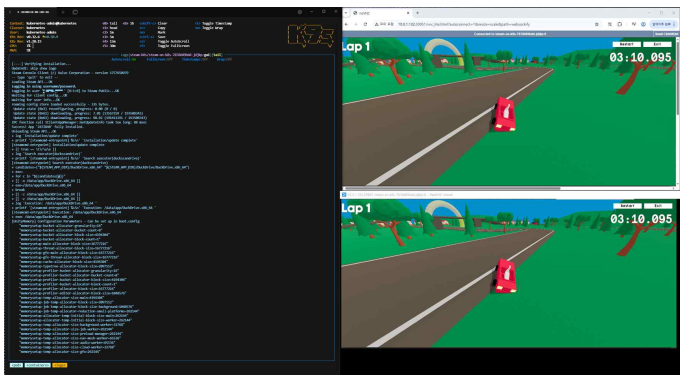
2-2. 스트리밍 환경 구축



<그림 2> 쿠버네티스 스트리밍 애플리케이션 구축 절차

그림 2는 아키텍처에서 앞선 절차를 통해 구성한 Steam 컨텐츠를 활용하여 스트리밍 세션을 형성하는 과정을 나타낸다. 이 과정은 이미지 빌드와 배포, 스케줄링, 가상 디스플레이 실행, Remote Display를 통한 사용자 접속으로 이어지는 일련의 파이프라인을 완성하며, 제안된 아키텍처의 핵심 작동 방식을 설명한다. 애플리케이션 실행 환경은 컨테이너 이미지로 빌드되어 이미지 레포지토리에 저장되며, 보안과 재현성을 위해 계정·토큰 등 민감한 정보는 이미지에 포함하지 않고 런타임에 외부에서 주입한다. 이후 Deployment 및 Service 객체를 통해 실행 정책과 접근 방식을 정의하고, 스케줄러는 해당 Pod를 노드에 할당한다. 정의된 가상 디스플레이의 출력은 Display Endpoint를 통해 외부로 연결되며, Service 객체는 이를 ClusterIP, NodePort, LoadBalancer 등 다양한 방식으로 노출할 수 있다. 이러한 과정을 통해 클라이언트는 Remote Display Client를 기반으로 구축된 서비스 실행 화면에 일관된 실행 환경을 통해 접근하고 상호작용을 할 수 있다.

2-3. 샘플 애플리케이션 실행 결과



<그림 3> 샘플 애플리케이션 'Ducks Can Drive' 스트리밍 예시

그림 3은 제안된 아키텍처를 적용하여 샘플 애플리케이션 'Ducks Can Drive[3]'를 스트리밍 전용 원격 클라이언트 및 브라우저를 통해 실행한 결과를 보여준다. 컨텐츠는 쿠버네티스 환경에서 배포되어 가상 디스플레이 상에서 구동되며, Remote Display Client를 통해 클라이언트 단에서 접근 및 조작이 가능하다. VNC와 noVNC 두 가지 방식을 통해 동일한 컨텐츠가 스트리밍으로 제공됨을 확인할 수 있으며, 이를 통해 제안된 파이프라인이 단말 환경과 관계없이 일관된 실행 경험을 제공함을 확인하였다.

3. 결론

본 논문에서는 SteamCMD를 활용하여 구성한 대화형 애플리케이션을 쿠버네티스 기반 클라우드 환경에서 표준화된 절차로 컨테이너화하고, 원격 디스플레이를 통해 스트리밍하는 아키텍처를 설계 및 구현하였다. 또한 가상 디스플레이 및 Remote Display Client를 통한 상호작용을 지원함으로써 사용자 환경에 제약이 최소화된 접근성을 확보하였다.

제안된 아키텍처는 원격·하이브리드 및 클라우드 네이티브 애플리케이션 실행 환경의 참조 모델로 활용될 수 있다. 특히 단말 이질성과 설치·업데이트 부담을 최소화하고, 다양한 환경에서 동일한 실행 경험을 제공할 수 있다는 점에서 실용성을 갖는다. 향후 연구 및 설계를 통해 GPU 가속 환경, 지원 플랫폼 확장 등을 통해 보다 폭넓은 응용 시나리오를 수행하는 고성능 스트리밍 아키텍처로의 발전을 도모하고자 한다. 이를 통해 클라우드 기반 대화형 애플리케이션의 발전에 이바지할 수 있을 것이다.

ACKNOWLEDGMENT

이 논문은 2024년 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임(No. RS-2024-00460194), XR 디바이스 기반의 원격 파일럿 및 투시형 서비스 기술 개발

참 고 문 헌

[1] Daga, M. V., Daga, M. P., & Badak, A. "OPTIMIZING FPS FOR LOW-LATENCY LIVE STREAMING PIPELINES USING KUBERNETES AND CONTAINERS,"International Journal of Recent Research and Review, Special Issues-2, pp. 11-21, 2025.

[2] Poetra, F. R., et al. "Performance analysis of video streaming service migration using container orchestration." IOP Conference Series: Materials Science and Engineering. Vol. 830. No. 2. IOP Publishing, 2020.

[3] Joseph Cook, "Ducks Can Drive," SteamDB, Available: https://steamdb.info/app/2472840/ (accessed Sep. 12, 2025).