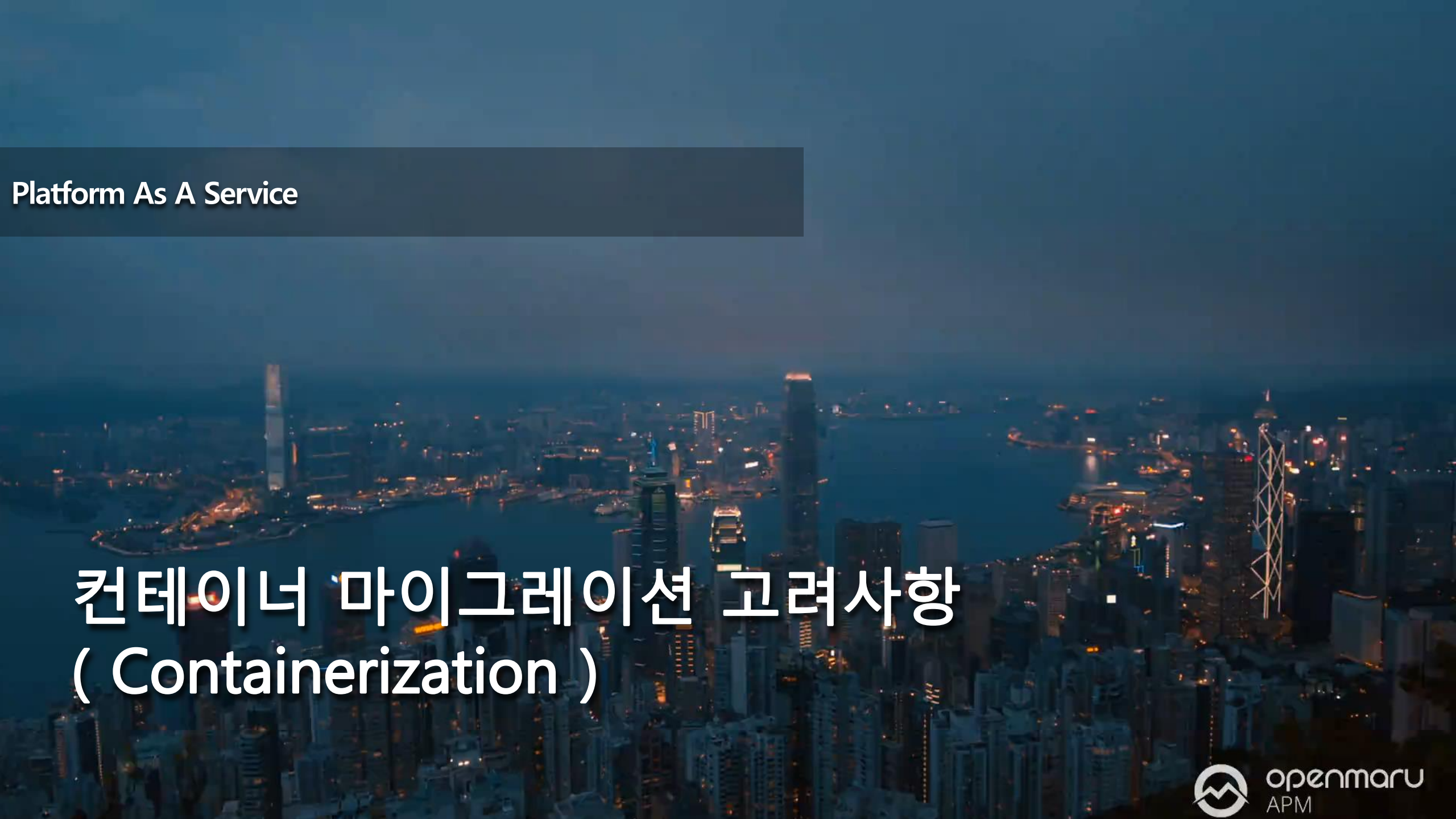

Platform As A Service

컨테이너 마이그레이션 고려사항
(Containerization)

openmaru
APM


Cloud Ready System
openmaru
APM

컨테이너의 특징에 따른 고려사항

특징	상세
POD 단위로 확장이 쉬움 (IP와 Hostname이 임의로 부여됨)	• 기존 이미지에서 동일한 POD를 만들기 때문에 쉽게 Scale Out 가능 • POD는 Scale Out이 가능하도록 지정된 IP Range에서 임의로 부여됨. 고정된 IP를 사용할 수 없음. • 컨테이너 Hostname도 확장을 고려하여 [애플리케이션명]-XXXX 형태로 부여됨 • 컨테이너를 임의로 확장할 수 있기 때문에 확장할 수 있는 애플리케이션 구조가 되어야 함
POD 재생성시 컨테이너 내부 데이터는 초기화 됨	• 반드시 저장해야 업로드 데이터와 같은 데이터는 NFS와 같은 공유 스토리지를 연결해야 함 • NFS는 로컬보다 느림
보안상 컨테이너 실행 시 임의의 사용자가 만들어 짐	• root 사용자 및 특정 User는 사용이 제한됨(옵션으로 허용 가능하지만 보안상 취약) • OpenShift POD내의 User는 임의의 숫자 형태로 사용됨 (예 : 10000020000) Group 권한으로 root가 지정됨) • 컨테이너 내에서 Host의 Network 및 리소스에 접근할 수 없음 (옵션으로 허용 가능하지만 보안상 취약)
표준출력으로 전달되는 로그는 Web Console에서 조회가능	• 컨테이너는 Foreground로 실행되기 때문에 표준 출력 파일을 별도로 지정할 수 없음 • 표준 출력에 나오는 메시지는 CLI나 웹 콘솔에서 조회 가능 • 일반 로그 File을 NFS를 사용하여 저장 가능하지만, 같은 디렉터리에 생기기 때문에 hostname 을 파일명 앞에 사용하여 동일 파일이 생성되지 않도록 해야 함

애플리케이션 Containerize시 고려사항 (1/2)

구분	내용	비고
Containerize 대상선정	시스템 중 컨테이너로 옮길 대상을 검토	• Windows 전용 시스템은 Linux로 전환 후 컨테이너화 • 무거운 Legacy WAS는 가벼운 오픈소스 WAS 전환 고려 • DB는 추천하지 않음(성능이 Critical 하지 않은 경우만 사용) • 배포 WAR 별 컨테이너로 구성 • 세션 클러스터링이 필요없는 애플리케이션이 확장시 더 효율적
Base Image선택	Official WAS 이미지 확인	• WAS 벤더에서 제공하는 이미지가 있는지 우선 확인
WEB/WAS 연동 아키텍처	WAS IP가 동적으로 변경되며, Scale Out 을 고려하여 Web 서버에서 WAS IP를 지 정하는 연결방식은 사용할 수 없음	• OpenShift 환경에서는 WEB은 제외하고 WAS로만 서비스 하는 단순한 아키텍처가 효율적임
내부 인터페이스 주소	OpenShift 내부에 인터페이스 대상 서버 가 있을 경우 대상 IP가 동적으로 변경되 기 때문에 연동방법 확인 필요	• OpenShift의 Service명 또는 환경변수를 이용하여 해결 가능
세션클러스터링	IP가 임의로 변경되기 때문에 IP 지정방식 의 클러스터링은 사용할 수 없음	• 세션 클러스터링 방식은 WAS 마다 다르기 때문에 벤더에 확인 필요함 • JBoss EAP는 세션 클러스터링 가능(JGroups DNS_PING, KUBE_PING사용) • 세션을 Data Grid에 저장하는 방식 추천
애플리케이션 설정값	변경이 필요한 값은 이미지에 넣지 않고 별도로 관리(환경변수, 파일등)	• 변경 필요한 대상 확인 • 변경 값 적용 방식 확인(애플리케이션 설정값, Java 옵션, 환경변수 등)
애플리케이션 소 스	소스에서 Hostname이나 IP에 기반한 로 직이 있으면 수정해야 함	• localhost나 환경변수를 사용

애플리케이션 Containerize시 고려사항 (2/2)

구분	내용	비고
보안	• root 유저를 사용하는 경우, ssh client 사용하는 경우, Host의 자원에 접근하는 경우 보안에 취약할 수 있음	• 설정 변경하여 사용가능
3rd Party Library	• 애플리케이션의 라이선스가 IP나 Hostname기반일 경우 변경 필요(벤더 확인) • C로 작성된 so 파일을 사용하는 경우 해당 Image에 사용가능한 버전으로 변경 필요 • IP, Hostname 기반으로 라이선스를 사용 하는 상용 S/W는 Any IP 기반으로 교환	• 사용하는 3rd Party 라이브러리의 종류 확인 • 해당 벤더에 문의
대용량 컨텐츠	• 대용량 컨텐츠를 이미지로 만들 경우 • Build / 배포 속도가 매우 느려짐	• 공유 볼륨에 대용량 컨텐츠를 분리하여야 함 • 공유 볼륨을 PV(Persistent Volume)로 컨테이너에 마운트하여 사용
Logging	• Standard Out으로 출력되는 로그는 오픈 시프트에서 확인가능	• 파일로 저장되는 로그는 기존과 동일하게 파일로 출력 • 로그가 저장되는 디렉터리를 PV(Persistent Volume)로 연결하여 사용 가능
Batch 형태 애플리케이션	• 클러스터링이 고려되지 않는 Batch 형태 의 애플리케이션은 POD 확장시 여러 번 실행될 수 있음	• WAS의 서비스 로직과 Batch 업무 로직을 별도의 POD로 분리 • Crontab으로 처리하는 부분은 OpenShift Cron Job으로 대체 가능