2025 하반기 커리큘럼 목차

표준 리눅스(4일)

1. 표준 리눅스 시스템 어드민

교육명	표준 리눅스 시스템 운영
교육개요	리눅스를 문화적 그리고 기술적으로 접근하여 리눅스 시스템 이해도를 높인다. 대다수 리눅스 시스템 애플리케이션은 현재 유기적으로 통합이 되고 있다. 어떤 식으로 유기적으로 통합이 되고 있는지, 그리고 시스템 영역에서 어떤 식으로 현대화가 되고 있는지 확인한다.
목표	systemd 중심으로 통합된 시스템 systemd기반으로 관리를 위한 기본적인 명령어 systemd기반으로 디스크 관리를 위한 명령어 systemd기반으로 컨테이너 및 가상머신을 위한 리눅스 리눅스 자동화를 위한 YAML, TOML그리고 앤서블
기간	총 4일, 3일로 조정가능 과정
랩	오픈스택 기반, 인터넷 사용 가능한 노트북 요구
교육대상	1년 이상 혹은 미만의 리눅스 관련 시스템 관리자 혹은 네트워크 엔지니어 가상화 혹은 컨테이너 관련 프로젝트 시스템 엔지니어 혹은 소프트웨어 엔지니어
1일	**리눅스와 오픈소스 관계** **리눅스 쉘 및 커널의 구성** - 리눅스 표준 디렉터리 구성 및 역할 설명 - FSH VS LSB의 차이점 - 왜 리눅스 배포판은 디렉터리가 다른가? **systemd기반으로 통한 시스템 블록 이해** - systemd와 system-v의 차이점 - 왜 대다수 리눅스 배포판은 systemd로 전환하였는가? - systemd에서 제공하는 주요 리소스 유형 - 앞으로 시스템 자원 유닛(/etc/fstab, block device) **서비스 및 로그 관리를 위한 명령어(로깅 분석)** - rsyslog는 - systemd, journald기반으로 시스템 블록 로깅 분석 - systemd기반으로 중앙 로깅 시스템 구성 **자주 사용하는 기본적인 리눅스 명령어** - systemd-booted기반으로 부팅 관리 - systemd-kernel 관리 - systemd-locale, loginctl - systemd-machine-id **리눅스 커널 동작 구조(간단하게)** - 현재 리눅스 커널의 아키텍처 및 기능
2일	**새로운 네트워크 구성 및 관리자 사용** - systemd-network

	- NetworkManager - ip, ss, route - rh-ifcfg-scripts - systemd-resolved, systemd-hostnamed - systemd-timesyncd, chronyd **새로운 방화벽 시스템(nftables, firewalld)** - netfilters는 무엇인가? - iptables vs nftables - firewalld와 nftables의 관계 - firewalld-cmd명령어 사용 방법 - nft명령어 사용 방법 **프로세스 및 사용자 관리를 위한 명령어** - 사용자 생성 및 비밀번호 설정 - 사용자 관리를 위한 usermod - 특정 사용자 그룹 생성 및 설정 - 그룹 관리를 위한 groupmod - 패스워드 생성 - 시스템 프로세스 및 사용자 프로세스 차이점 - 프로세스 관리를 위한 명령어 **파일 시스템 퍼미션** - 사용자/그룹 및 시스템 퍼미션 - ACL기반 퍼미션 관리 - 현재 리눅스에서 umask사용 및 관리 방법 **패키지 관리자** - 리눅스 배포판에서 제공하는 패키지(RPM, DEB) - 고급 패키지 관리자(DNF/YUM/APT) - 리눅스 커널 패키지 관리(소스코드 및 바이너리) - 패키지 리 빌드는 필요한가? 왜 법적으로 문제가 발생하는가?
3일	**LVM개념 및 구성 및 구현** - LVM2의 개념 - 왜 레드햇 계열은 여전히 LVM2를 사용하는가? - LVM2와 파티션 영역 관계 - PV/VG/LV생성 및 관리 - LVM2의 확장 기능 사용 - 백업 및 복구 **BTRFS** - BTRFS 파일 시스템 소개 및 구성 - Pool 생성 및 관리 - Volume 생성 및 관리 - 파일 시스템 수정 도구

	XFS - XFS 파일 시스템 소개 및 구성 - XFS 관리 도구 - Stratis for XFS 설명 및 구성 - VDO for XFS 설명 및 구성 - 파일 시스템 수정도구 **성능 모니터링** - systemd기반에서 시스템 모니터링 - sysstat와 iostat, vmstat - 네트워크 모니터링 방법
4일	**리눅스 커널과 모듈** - 리눅스 커널에서 모듈의 역할 - 모듈 튜닝 및 설정 - 모듈 디버깅 - 커널 파레메터 조정하기(tuned, systemd-sysctld) - udev와 그리고 dbus의 관계 **백업 도구 및 응급 복구** - 리눅스에서 제공하는 백업 방법 및 도구(tar, rsync, dd) - 이미지 기반 백업 도구(disk, partition) - 응급 복구 - single, emergency mode **가상머신 및 컨테이너 기술 소개** - systemd-machine - libvirtd, libguest-tools - podman과 그리고 conmon, crun - podman기반으로 pod, container 생성 - buildah, skopeo를 통한 이미지 구성 및 이미지 서버 구성 **시스템 자동화** - ansible/terraform/salt 비교 - 각 도구별 자동화 예제 및 구성

2. 표준 리눅스 시스템 엔지니어

교육명	표준 리눅스 시스템 엔지니어
교육개요	리눅스를 문화적 그리고 기술적으로 접근하여 리눅스 시스템 이해도를 높인다. 대다수 리눅스 시스템 애플리케이션은 현재 유기적으로 통합이 되고 있다. 어떤 식으로 유기적으로 통합이 되고 있는지, 그리고 시스템 영역에서 어떤 식으로 현대화가 되고 있는지 확인한다.
목표	systemd 중심으로 통합된 시스템 systemd기반으로 관리를 위한 기본적인 명령어 systemd기반으로 디스크 관리를 위한 명령어

	systemd기반으로 컨테이너 및 가상머신을 위한 리눅스 리눅스 자동화를 위한 YAML, TOML그리고 앤서블
기간	총 4일, 3일로 조정가능 과정
랩	오픈스택 기반, 인터넷 사용 가능한 노트북 요구
교육대상	1년 이상 혹은 미만의 리눅스 관련 시스템 관리자 혹은 네트워크 엔지니어 가상화 혹은 컨테이너 관련 프로젝트 시스템 엔지니어 혹은 소프트웨어 엔지니어
1일	**리눅스와 오픈소스 관계** **리눅스 쉘 및 커널의 구성** - 리눅스 표준 디렉터리 구성 및 역할 설명 - FSH VS LSB의 차이점 - 왜 리눅스 배포판은 디렉터리가 각기 다른가? **systemd기반으로 통한 시스템 블록 이해** - systemd와 system-v의 차이점 - 왜 대다수 리눅스 배포판은 systemd로 전환하는가? - systemd에서 제공하는 주요 리소스 유형 - 앞으로 시스템 자원 유닛(/etc/fstab, block device) **서비스 및 로그 관리를 위한 명령어(로깅 분석)** - rsyslog는 왜 더 이상 주요 로깅 시스템이 아닌가? - systemd, journald기반으로 시스템 블록 로깅 분석 - systemd기반으로 중앙 로깅 시스템 구성 **자주 사용하는 기본적인 리눅스 명령어(일 부분만 목록)** - systemd-booted기반으로 부팅 관리 - systemd-kernel 관리 - systemd-locale, loginctl - systemd-machine-id - systemd-* 그 이외 확장 기능 **리눅스 커널 동작 구조(간단하게)** - 현재 리눅스 커널의 아키텍처 및 기능
2일	**새로운 네트워크 구성 및 관리자 사용** - systemd-network - NetworkManager - ip, ss, route - rh-ifcfg-scripts - systemd-resovled, systemd-hostnamed - systemd-timesycnd, chronyd **새로운 방화벽 시스템(nftables, firewalld)** - netfilters는 무엇인가? - iptables vs nftables - firewalld와 nftables의 관계 - firewalld-cmd명령어 사용 방법

	- nft명령어 사용 방법 **프로세스 및 사용자 관리를 위한 명령어** - 사용자 생성 및 비밀번호 설정 - 사용자 관리를 위한 usermod - 특정 사용자 그룹 생성 및 설정 - 그룹 관리를 위한 groupmod - 패스워드 생성 - 시스템 프로세스 및 사용자 프로세스 차이점 - 프로세스 관리를 위한 명령어 **파일 시스템 퍼미션** - 사용자/그룹 및 시스템 퍼미션 - ACL기반 퍼미션 관리 - 현재 리눅스에서 umask사용 및 관리 방법 **패키지 관리자** - 리눅스 배포판에서 제공하는 패키지(RPM, DEB) - 고급 패키지 관리자(DNF/YUM/APT) - 리눅스 커널 패키지 관리(소스코드 및 바이너리) - 패키지 리빌드는 필요한가? 왜 법적으로 문제가 발생하는가?
3일	**LVM개념 구성 및 구현** - LVM2의 개념 - 왜 레드햇 계열은 여전히 LVM2를 사용하는가? - LVM2와 파티션 영역 관계 - PV/VG/LV생성 및 관리 - LVM2의 확장 기능 사용 - 백업 및 복구 **BTRFS** - BTRFS 파일 시스템 소개 및 구성 - Pool 생성 및 관리 - Volume 생성 및 관리 - 파일 시스템 수정 도구 **XFS** - XFS 파일 시스템 소개 및 구성 - XFS 관리 도구 - Stratis for XFS 설명 및 구성 - VDO for XFS 설명 및 구성 - 파일 시스템 수정도구 **성능 모니터링** - systemd기반에서 시스템 모니터링 - sysstat와 iostat, vmstat - 네트워크 모니터링 방법

4일	**리눅스 커널과 모듈** - 리눅스 커널에서 모듈의 역할 - 모듈 튜닝 및 설정 - 모듈 디버깅 - 커널 파라메터 조정하기(tuned, systemd-sysctl) - udev와 그리고 dbus의 관계 **백업 도구 및 응급 복구** - 리눅스에서 제공하는 백업 방법 및 도구(tar, rsync, dd) - 이미지 기반 백업 도구(disk, partition) - 응급 복구 - single, emergency mode **가상머신 및 컨테이너 기술 소개** - systemd에서 컨테이너 및 가상머신 통합(systemd-machine) - libvirtd, libguest-tools - podman과 그리고 conmon, crun - podman기반으로 pod, container 생성 - buildah, skopeo를 통한 이미지 구성 및 이미지 서버 구성

3. 컨버전스 리눅스(마이그레이션)

교육명	**컨버전스 리눅스(리눅스 윈백 마이그레이션)**
교육개요	현재 레드햇에서 배포하는 RHEL의 GPL 라이선스의 루프-홀(Loop Hole)를 통해서 더 이상 다른 배포판에서 SRPM를 재배포할 수 없도록 하였다. 이로써, 로키 리눅스 및 알마 리눅스는 더 이상 레드햇과 호환성을 100%맞출수는 없으면 ABI호환성만 유지하도록 결정하였다. 어떠한 배포판이 기업에 적절한지, 랩을 통해서 각 제품들을 비교하면서 적절한 배포판을 선택할 수 있도록 한다.
목표	기존 레드햇 기반의 배포판에서 SuSE기반으로 전환하는 방법에 대해서 학습한다.
기간	총 4일
랩	오픈스택 기반, 인터넷 사용 가능한 노트북 요구
교육대상	데비안 기반의 배포판과 레드햇 계열 배포판 비교를 원하는 사용자 레드햇 계열 배포판에서 데비안 계열 및 레드햇 호환 배포판으로 전환을 원하는 사용자
1일	수세/알마(로키/센트 스트림) 리눅스 비교 레드햇 ABI(Application Binary Interface)/kABI와 SuSE ABI/kABI비교 수세/레드햇/데비안 리눅스 설치 기본적으로 제공하는 무인 설치 도구 수세 리눅스에서의 패키지 관리자(DNF, Zypper, rpm) TUI/GUI 도구 비교(YaST2, RedHat TUI tools) 리눅스 커널 및 파일 시스템의 차이 및 호환성 구성 방법(xfs, btrfs, ext3/4) 수세/레드햇/데비안 LSB비교
2일	리눅스 표준 명령어(레드햇/수세/데비안) 바이너리 및 라이브러리 비교 커널 모듈 관리 및 운영방법 블록장치 관리 및 구성 명령어(fuse, devicemapper, udev, multipath등)

	수세/레드햇/데비안 계열의 systemd 비교 네트워크 구성 및 설정(wicked, NetworkManager, YaST2)
3일	방화벽(firewalld, nftables, iptables) 패키지 저장소 관리 및 확장 방법 자동화 도구(앤서블/Salt/Terraform) 커널 라이브 패치(Kgraft vs kpatch) Cockpit기반으로 서버 관리(레드햇/수세 리눅스)
4일	소프트웨어 솔루션(미들웨어, 스토리지, 백업) 메모리/가상화 기술 비교 표준 컨테이너 시스템 사용 설치 및 사용 방법 수세 리눅스에 바닐라버전 쿠버네티스 설치 가이드 제안 자동화를 위한 솔루션 하드웨어 및 소프트웨어 트러블 슈팅을 위한 가이드

고급 리눅스(4일)

1. 쉘 스크립트 파워툴즈

교육명	**표준 쉘 스크립트**
교육개요	리눅스에서 많이 사용하는 배시 쉘은 오랫동안 사용한 쉘 도구이다. 기본적인 함수 및 변수를 학습하면서
목표	배시 스크립트를 현대적 쉘 스크립트로 작성하는 방법 학습 대화형 쉘 스크립트를 작성하기 위한 방법 학습 쉘 스크립트기반으로 쿠버네티스 활용 도구
기간	3일
랩	오픈스택 기반, 인터넷 사용 가능한 노트북 요구
교육대상	최소 1년 이상 리눅스 시스템 운영자
1일	리눅스에서 제공하는 쉘 서비스 표준 쉘 스크립트 문법 및 가이드 변수 선언 및 함수 선언 변경된 쉘 조건문 및 중복 중괄호 대괄호와 이전 괄호의 차이점 시스템 변수 형식 선언 간단한 쉘 스크립트 프로그램 쿠버네티스 클러스터(SNK) 구성
2일	kubectl명령어를 배시 스크립트로 랩핑 - 쿠버네티스에서 필요한 기능 추가 - 네임스페이스 고정 - 사용자 로그인 - 클러스터 정보 확인 및 전환 - 이벤트 로그 확인 및 구성
3일	노드 관리 - 노드 관리를 위한 kubectl 랩핑 - sshless를 위한 관리 명령어 확장

	자원 관리 - POD 생성 및 관리를 위한 명령어 확장 - 서비스 생성 및 관리를 위한 명령어 확장 - 인그레스 생성 및 관리를 위한 명령어 확장 - 로드 밸런서(metallb) 생성 및 관리를 위한 명령어 확장

2. 표준 리눅스 클러스터링 시스템(Pacemaker)

교육명	**표준 리눅스 클러스터링 시스템**
교육개요	현 리눅스 시스템은 물리/가상적으로 여러 운영체제에 설치하여 사용한다. 이러한 이유로 특정 노드에 장애가 발생시, 다른 노드에서 중지된 서비스를 그때로 이전을 시켜야 한다. 현재 리눅스 시스템은 Pacemaker라는 표준적인 High Availability시스템을 사용하고 있다. 어떠한 방식으로 설치 및 운영하는 기본적인 교육을 통해서 확인한다.
목표	리눅스 기반으로H/A시스템 구축 및 운영에 대해서 학습한다.
기간	총 4/5일
랩	오픈스택 기반, 인터넷 사용 가능한 노트북 요구
교육대상	최소 1년 이상의 리눅스 시스템 운영자 혹은 서비스 아키텍트
1일	페이스메이커 소개 - 페이스메이커 구조 및 기능 설명 - Corosync 및 Agent설명 - LSB/OCF에이전트 설명 페이스메이커 랩 구성 - 하이퍼브이 기반으로 구성 - 랩 스토리지 생성 페이스메이커 설치 - 페이스 메이커 설치 - 기본적인 클러스터 구성 및 설치 - 노드 추가 및 삭제 방법 - pcs명령어 사용방법
2일	표준 서비스 기반으로 리소스 구성 - 기본 리소스 생성 및 관리 - 리소스 설정 및 구성 DRBD기반 스토리지 관리 - DRBD구성 - DRBD기반으로 스토리지 복제 및 공유 펜싱 장치 - 펜싱 장치 설명 - 펜싱 장치 생성 및 구성 페이스메이커 웹 관리자 - 웹 관리자 구성 및 접근 - 사용방법

	페이스메이커 ACL/ALERT - ACL 설명 - ACL기반으로 사용자 접근 관리 - ALERT 설명 - ALERT기반으로 알람 설정
3일	페이스메이커 Booth - Booth설명 - 간단한 설치 및 구성 설명(실제로 설치하지 않음. 고급정에서 추후) 리소스 점수(Resource score/infinity) - Score, infinity 설명 - Location, co-location, resources, stickiness 페이스메이커 DR - DR설명 - 간단한 설치 및 구성 설명(실제로 설치하지 않음. 고급정에서 추후) QDEVICE - QDEVICE 설명 - 간단하게 설치 및 구현 리소스 생성 - 리소스 생성 - 리소스 관리 및 펜싱 LVM2기반 스토리지 관리 - LVM2 리소스 설명 - LVM2 구성 및 랩 실험 NFS기반 스토리지 관리 - NFS 리소스 설명 - NFS 구성 및 랩 실험
4일	아파치 서비스 - 설명 - 서비스 구성 및 실험 GFS2 스토리지 - 설명 - 서비스 구성 및 실험 톰캣 서비스 - 설명 - 서비스 구성 및 실험 PgSQL/Mariadb - 설명 - 서비스 구성 및 실험 DRBD+TOMCAT+MARIADV

	- 설명 - 서비스 구성 및 실험 TWO Node 구성 - 설명 - 액티브/스탠바이 구성 및 실험 - 액티브/액티브 구성 및 실험

3. 표준 리눅스 장애처리

교육명	표준 리눅스 장애처리
교육개요	현재 모든 리눅스는 표준적인 시스템 블록 관리자 그리고 파일 시스템을 사용한다. 특히, systemd도입을 통해서 파일 시스템에서 많은 부분들이 변경이 되었다. 파일 시스템은 경우에는 기존의 XFS 및 BTRFS를 기반으로 각 배포판들은 기업용 파일 시스템을 도입하고 있다. 이러한 변경점들에 대해서 학습 및 확인하고 어떠한 방식으로 장애처리가 가능한지 확인한다.
목표	systemd에서 발생하는 시스템 블록 장애에 대해서 살펴본다. 기본적인 파일 시스템 및 확장 파일 시스템 영역에서 발생하는 장애에 대해서 살펴본다. 네트워크 블록에서 발생하는 장애에 대해서 살펴본다. 시스템 영역에서 콜 제한 프로그램에 발생할 수 있는 문제에 대해서 확인한다. 패키지 관리자에서 발생하는 문제에 대해서 살펴본다.
기간	3일(가급적 4일 권장)
랩	오픈스택 기반, 인터넷 사용 가능한 노트북 요구
교육대상	최소 1년 이상의 리눅스 시스템 운영자 혹은 서비스 아키텍트
교육대상	리눅스 운영 담당자 리눅스 사용 경험이 2년 이상 운영자
1일	systemd에 통합된 시스템 영역 확인 램 디스크를 통한 운영체제 복구 및 파일 시스템 확인 grub2, bootrec를 통한 부트로더 재구성 및 복구 부트 업 장치 복구 및 영구적 마운트 장치 수정(/etc/fstab, .mount) 부트 업 속도 확인 및 속도 개선 파일 시스템 복구(xfs 및 LVM2) 블록장치 용량 확장(LVM2, Stratis) 블록장치 장애 확인 및 처리 및 수정
2일	systemd-journald통한 시스템 이벤트 분석 및 로그 분석 coredump활성화 간단한 장애 발생 및 분석 SELinux 컨텍스트 및 시스템 콜 장애처리 네트워크 설정 및 장애처리(NetworkManager) 성능 향상 및 분석을 위한 모니터링 방법
3일	커널 모듈 및 파라메터 확인 및 수정 메모리 페이징 및 스왑 장애처리 그리고 확장 리눅스 시스템 사용자 세션 관리 및 분석(사용자 콘솔 로깅 포함) 보고 및 분석을 위한 시스템 기록 수집 및 분석
4일	QEMU/KVM, Libvirt 장애 분석 및 처리를 위한 방법 제안 컨테이너 장애 분석 및 처리를 위한 방법 제안 메모리 관리 OOM, CGROUP 확인 및 장애처리

	RPM 및 DNF3 장애 처리 확장 퍼미션(attr)통한 장애 처리 및 확인

4. Foreman 기반, 리눅스 노드 관리 자동화

교육명	리눅스 노드 관리 자동화
교육개요	온프레미스 및 퍼블릭 클라우드에 많은 컴퓨팅 자원들이 생성 및 구성이 된다. 특히, 많은 회사들이 온프레미스로 다시 롤백 혹은 전환하면서 대용량 노드 관리에 대해서 많은 어려움이 있다. 이를 해결하기 위해서 오픈소스에서 많이 사용하는 Foreman기반으로 큰 규모의 컴퓨팅 노드 구성 및 관리를 학습하도록 한다.
목표	여러 물리적 서버 및 가상서버 관리 방법에 대해서 학습 프로비저닝이 완료된 서버에서 앤서블과 같은 도구로 구성하는 방법학습 foreman기반으로 여러 리눅스 배포판 구성 및 업데이트에 대해서 학습
기간	3일(권장은 4일)
랩	오픈스택 기반, 인터넷 사용 가능한 노트북 요구
교육대상	대형 인프라를 관리 및 운영하는 시스템 관리자 인프라 설치 및 자동화가 필요한 시스템 관리자 프로비저닝 및 디플로이먼트 통합이 필요한 시스템 관리자
1일	The Foreman 설명 - Foreman 아키텍처 - Foreman 주요 기능 - 기존 관리 방식과 Foreman의 차이점 및 장점 The Foreman 설치 - 설치 전 요구사항 - 통합 부분 * puppet, ansible, docker(podman) 통합 * 네트워크 및 호스트 The Foreman 설치 - 레드햇 계열 - 데비안 계열 - 수세 계열 - 설치 및 기본설정 - Foreman 웹 인터페이스 사용 The foreman 기본 구성 - 호스트 등록 - O/S 템플릿 구성
2일	호스트 관리 - 호스트 생성 및 등록 - 호스트 그룹 생성 및 등록 - OS 템플릿 기반으로 프로비저닝 구성 프로비저닝 환경 설정 및 구성

	- 환경 분리 구성하기(dev, test, product) - 환경별 호스트 그룹 및 설정 - 프로비저닝 이후 puppet/ansible/docker(podman) 자동화 도구와 통합 호스트 배포 및 관리 - OS배포 - 호스트 상태 관리 및 모니터링 - 클라우드 시스템에서 프로비저닝 자동화 Puppet/ansible 통합 - 통합 시나리오 설명 - 앤서블 기반으로 설치 후 디플로이먼트 구성
3일	Foreman 확장 기능 - 확장 모듈 - 템플릿 구성 및 관리 - RPM 저장소 관리 및 구성 사용자 관리

5. 리눅스 보안 하드닝

교육명	**리눅스 보안 하드닝** **NIST/KISA 기반 Linux 하드닝 및 자동화 보안 실습**
교육개요	본 교육은 NIST SP 800-53 및 KISA ISMS-P 기반 보안 정책을 실무 Linux 시스템에 적용하는 방법을 학습합니다. 기초 보안 하드닝부터 Ansible 자동화, 시스템 무결성 검증 및 CVE 기반 모의 해킹까지 포함되어 있으며, 보안성과 운영 안정성을 동시에 확보할 수 있는 실습 중심 교육입니다.
목표	- NIST와 KISA 기준의 보안 정책을 이해하고 비교할 수 있다. - 시스템 접근 제어 및 서비스 하드닝을 수행할 수 있다. - Ansible을 활용한 자동화 보안 설정을 수행할 수 있다. - 실제 취약점을 바탕으로 침해 분석 및 대응 실습을 수행할 수 있다.
기간	4일 (총 28시간)
랩	오픈스택 기반, 인터넷 사용 가능한 노트북 요구
교육대상	Linux 시스템 관리자, 보안 담당자, DevSecOps 입문자
1일	**보안 하드닝 개요 및 접근제어** - NIST와 KISA 보안 정책 구조 비교 - Linux 보안 사고 사례 분석 - 계정 관리 및 패스워드 정책 설정 - su/sudo 제한, PAM 설정 이해 **[실습]** - /etc/passwd, /etc/shadow 보호 - login.defs, pam_limits.conf 설정 - 사용자 권한 감사

2일	**네트워크 보안 & Ansible 자동화** - SSH 하드닝, 포트 변경, 키 인증 - iptables/firewalld 설정 - logrotate, journald 로그 정책 - Ansible로 하드닝 정책 자동화 **[실습]** - Ansible로 SSH 설정 배포 - firewall-cmd 구성 - logrotate 정책 작성 및 테스트
3일	**파일 시스템 보호 및 불변 시스템 이해** - chattr, lsattr를 통한 파일 보호 - /usr, /boot 읽기 전용 마운트 실습 - MicroOS, CoreOS vs 일반 배포판 비교 - SELinux/AppArmor 보안 정책 이해 **[실습]** - chattr +i 적용 실습 - SELinux 설정 및 로그 확인
4일	**CVE 기반 모의 해킹 & 무결성 검증** - CVE-2021-4034(polkit) 권한 상승 실습 - auditd, aide, rpm -Va 무결성 검증 이해 - Ansible로 패치 적용 자동화 - 침해 흔적 포렌식 및 대응 전략 **[실습]** - polkit CVE 실습 - 무결성 검사 및 포렌식 대응 시나리오

컨테이너(4일)

1. 포드만 컨테이너 어드민

교육명	운영 및 개발을 위한 고급 표준 컨테이너 런타임 활용
교육개요	오픈소스에서는 고수준 컨테이너를 포드만(Podman)으로 사용한다. 기존에 사용하였던 도커(Docker)는 많은 배포판에서 더 주요 런타임으로 지원하지 않는다. 현재 도커는 미란티스(Mirantis)회사로 인수가 되어, CRI-Docker로 다시 재작성 하고 있다. 커뮤니티는 Podman기반으로 사용하고 있다. 고수준 컨테이너 포드만과 쿠버네티스에서 사용하는 표준 컨테이너 CRI-O에 대해서 학습한다. 아울러, 간단하게 Pod와 Container의 차이점, 그리고 OCI, CRI 인터페이스 및 도구에 대해서 학습한다.
목표	Docker/Containerd와 Podman/CRI-O의 차이점에 대해서 학습한다. Podman기반으로 Pod 및 Container생성 및 관리한다. Podman API 서비스에 대해서 알아본다.

기간	총 4일, 3일로 조정가능 과정
랩	오픈스택 기반, 인터넷 사용 가능한 노트북 요구
교육대상	최소 1년 이상 컨테이너 운영 경험자 도커에서 표준 컨테이너로 이전하려는 개발자 혹은 인프라 엔지니어
1일	표준 컨테이너는 무엇인가? - 기존 컨테이너와 차이점 - Podman, CRI-O, Docker그리고 Containerd - 가상화와 컨테이너의 차이점 쿠버네티스와 컨테이너 런타임 관계 - 쿠버네티스에서 지원하는 컨테이너 런타임 - Podman은 왜 쿠버네티스에서 사용하지 못하는가? - 쿠버네티스에서 사용이 가능한 표준 컨테이너 도구 - 표준 컨테이너 설정 containers, CNI 그리고 csi 포드만 설치하기 - 포드만 패키지 살펴보기 - 포드만 패키지 설치 - 포드만 서비스 확인 포드만 명령어 - 컨테이너 명령어 - 포드 명령어 - 네트워크 명령어 - 스토리지 명령어
2일	기본 컨테이너 개념 및 기술 - c-group, namespace - unshare - container monitor, runc(crun) - Pod(pause) - Kernel, init 그리고 Pod - NSPID,SUBID 그리고 UID, GID(rootless, rootful) 특수권한 컨테이너 생성하기 Pod기반으로 컨테이너 생성 및 관리 - Pod생성 및 관리 방법 - Pod와 Container 연결 - Pod에서 Container Application과(의) 관계 네임스페이스 네트워크 확인하기 표준 컨테이너 디렉터리 역할 - /run/containers/ - /run/pod - /var/lib/containers 오버레이 디스크 동작 방식 표준 컨테이너 기반 이미지 생성

	- podman build - buildah 이미지 검색 및 복사 - /etc/containers/ - skopeo
3/4일	다중 Podman Node 관리 및 운영 - Podman API 서버 - 다중 서버 접근/관리 및 서비스 배포 포드만에서 쿠버네티스로 서비스 테스트 - 쿠버네티스 서비스로 전환방법 컨테이너 기반으로 systemd서비스 구성 - 사용자 기반 systemd container 서비스 - 관리자 기반(root) systemd container 서비스

2. 표준 쿠버네티스 어드민

교육명	**쿠버네티스 기본과정**
교육개요	본 과정은 CNCF가 주도하는 클라우드 네이티브 기술 중 핵심인 Kubernetes를 중심으로, 설치부터 운영, 확장까지의 실무 전반을 다룹니다. 쿠버네티스 구성 요소, 컨테이너 생태계(OCI), 주요 오픈소스 도구들(Gogs, ArgoCD, Tekton 등)과의 연계, 그리고 DevOps 기반 운영 환경 구성 방법을 단계별 실습을 통해 학습합니다.
목표	Kubernetes의 핵심 구성 요소 및 설치 구조 이해 쿠버네티스 클러스터 설치 및 기본 네임스페이스/오브젝트 활용 Pod, 서비스, 볼륨 등 핵심 리소스의 생성 및 관리 실습 DevOps를 위한 자동화 도구 및 애플리케이션 배포 흐름 이해 클러스터 확장, 모니터링, 백업 등 운영에 필요한 관리 역량 습득
기간	총 4/5일, 3일 불가능.
랩	오픈스택 랩 기반 진행
교육대상	컨테이너 기반 인프라에 대한 기초 지식을 보유한 시스템 엔지니어 온프레미스 또는 클라우드 환경에서 Kubernetes 기반 운영을 도입하려는 인프라 관리자 DevOps 및 CI/CD 환경에 관심이 있는 실무자 또는 교육 담당자
1일	**쿠버네티스 소개** - 쿠버네티스는 무엇인가? - 쿠버네티스 구성요소 - 쿠버네티스 표준 사양 - 인프라 코드화 및 자원 코드화 - 클라우드 네이티브 **설치준비** - 권장사양 - 에디터 및 작업 도구 설정 - 하이퍼브이 설치 **쿠버네티스 도구** **쿠버네티스 아키텍처**

	- 가상화vs컨테이너 - RUNC/CRUN - 클러스터 구성원 설명 - 주요자원 설명 **시작하기 전에** - 최종확인 - 명령어 자동완성 - YAML문법 **기본기능** - 네임스페이스 - 생성/삭제/확인 - 멀티 컨테이너 구성 - cp - exec - expose/service/endpoint
2일	**기본기능** - port, port-forward - label - set - edit - delete - diff - debug/log - explain - replace - patch **프로젝트 애플리케이션** - 시나리오 - 인프라 소프트웨어 - 시나리오 소프트웨어 **쿠버네티스 인프라 소프트웨어** - 패키지 관리자 - 쿠버네티스 미터링 서비스 - 쿠버네티스 컨텍스트 및 사용자 구성 - 역할구성 및 서비스 계정
3일	**쿠버네티스 스토리지** - 스토리지 서버 구성 및 설명 - CSI-DRIVER-NFS 설치 - GENERAL-NFS 설치 - 스토리지 클래스 및 PV/PVC **Scale/rollout/rollback/history**

	컨테이너 자동확장(autoscale) **선호도 및 예외처리** **변수전달** **Drain/taint/cordon/uncordon** **노드 추가 및 제거** **컨테이너 및 포드 상태 확인** **Label, annotation, selectors** - 이름표 - 선택자 - 주석 - 노드 셀렉터 **데몬 서비스** **상태 POD 서비스** **작업** - 작업 - 크론잡 **설정파일(configmap)** **비밀(secret)** **배포 및 배치** - Deployment - Batch POD **복제자** - replicaset - replication controller **서비스 확장** - 외부아이피 주소(ExternalIP) - 외부이름(ExternalName) **네트워크 정책(NetworkPolicy)**
4일	**쿠버네티스 기능 확장** - 도메인 서버 - 로드 밸런서 - 인그레스 - Gogs - Docker-registry - ArgoCD - kube-virt - minio **DevOps를 위한 준비** - 데브옵스는 무엇인가? - 표준 컨테이너 도구 - Ansible 기반 클러스터 운영 자동화

	- Tekton - Istio - Knative **관리 및 운영** - 클러스터 업그레이드 - 소프트웨어 L4기반 컨트롤러 로드 밸런서 - ETCD백업 및 복원 - 쿠버네티스 스케줄러 수정 및 변경 - 오프라인 설치 - 이미지 복사 및 배포 - 모니터링(Prometheus+Grafana)

3. 수세 렌처 어드민

교육명	쿠버네티스/렌처
교육개요	쿠버네티스에 대한 경험이 없는 사용자 혹은 기존 경험이 있는 사용자 대상으로, 렌처 기반으로 손쉽게 쿠버네티스 클러스터 관리 방법에 대해서 학습한다. 쿠버네티스는 두 가지 API를 제공하는 기본 kubernetes, kubefed이다. 렌처에서 지원하는 멀티 클러스터와 기존 OCM, Karmada와 어떠한 차이점이 있는지 확인한다.
목표	쿠버네티스 설치 및 운영 명령어 학습한다. 하나 이상의 쿠버네티스 클러스터 구성 및 운영한다. 렌처 기반으로 한 개 이상의 클러스터에 디플로이먼트 및 서비스 관리에 대해서 학습한다.
기간	총 4일, 3일로 조정가능 과정
랩	오픈스택 기반, 인터넷 사용 가능한 노트북 요구
교육대상	간단하게 쿠버네티스 설치 및 명령어 학습을 원하는 사용자 클러스터 다중 운영 및 서비스 배포에 관심이 있는 사용자.
1일	**랩 및 솔루션 소개** - 쿠버네티스와 랜처와 관계 - K8S와 K3S의 차이점 - 업/다운 스트림 버전 차이점 - 수세와 랜처의 관계 - 랜처 아키텍처 **랜처 설치** - 운영체제 설정 - 랜처 및 쿠버네티스 설치 - 네트워크 구성 및 확인 **랜처 관리자 도구 설치** - 관리자 도구 설치 - 랜처 대시 보드 설명
2일	**랜처 활용(기본 자원 관리)**

	- namespace 및 POD 개념 - POD 생성 및 관리 - 자원 생성방법(apply, create) - POD와 Container관계 - POD 배포를 관리하는 Deployment, 그리고 RelicaSet - POD 구성을 위한 기본 활용 자원들 - 다중컨테이너 구성 - 서비스 생성 및 노출 - 레이블 및 선택조건(labels, match) - debug 및 log 확인 **랜처 스토리지 구현** - CSI와 쿠버네티스 스토리지 관계 - NFS기반으로 구성 및 연결 - PV/PVC그리고 스토리지 클래스 간단한 설명 **쿠버네티스 패키지**
3일	고급 자원 관리 - 쿠버네티스 고급 자원관리 랜처에서 CI/CD구성 및 구현

4. OCM & RKE2기반 멀티 클러스터 운영

교육명	**OCM·Rancher 기반 멀티 클러스터 운영 및 실전 관리**
교육개요	쿠버네티스 환경이 복잡해짐에 따라, 멀티 클러스터 관리 수요가 급증하고 있다. 본 교육은 바닐라 K8s 환경에서도 적용 가능한 Open Cluster Management(OCM)와 Rancher를 중심으로, 클러스터 통합, 정책 관리, GitOps 배포, 장애 대응까지 실습 중심으로 구성된다.
목표	- 바닐라 Kubernetes에서 OCM 설치 및 구성 능력 습득 - Rancher 기반 클러스터 통합 관리 비교 - PlacementRule/Subscription 활용 앱 배포 실습 - ArgoCD·Tekton·Fleet 연동 GitOps 환경 구성 - 클러스터 간 통신, 보안 정책, 장애 복구 전략 학습
기간	총 5일 (기본 4일 + 심화 1일 선택)
랩	- 3개 이상의 Kubernetes 클러스터 환경 (Kubeadm, Kind, RKE2 등 가능) - Git 서버 및 Helm Chart 저장소 - 네트워크 연결 실습 (Submariner 등)
교육대상	- Kubernetes 기반 서비스 운영 관리자 - 클러스터 통합 운영 전략이 필요한 SRE, DevOps 담당자 - GitOps 및 보안 정책 적용 환경을 설계하고자 하는 개발자 및 운영자

1일	**주요 내용** - 멀티 클러스터 운영이 필요한 이유 - 싱글 vs 멀티 클러스터 아키텍처 비교 - kubeconfig 병합 및 컨텍스트 전환 - 클러스터 통신 흐름 이해 (Ingress, DNS) **실습** - Kubernetes 클러스터 2~3개 생성 (Kubeadm or Kind) - kubectl config를 통한 context 병합 및 스위칭 - 각 클러스터에 샘플 애플리케이션 배포 및 상태 확인
2일	**2일차: OCM & Rancher를 이용한 클러스터 통합** **주요 내용** - OCM 아키텍처: Hub, Managed, Component 소개 - Helm을 이용한 OCM 설치 및 등록 - Rancher 개요 및 Project, Cluster 구조 이해 - OCM과 Rancher 등록 방식 비교 **실습** - OCM 설치 및 클러스터 등록 (Hub → Managed) - Rancher 설치 (Docker 또는 Helm 기반) - 동일한 클러스터를 Rancher에도 등록하고 비교 분석 - Web UI를 통한 클러스터 정보 시각화
3일	**3일차: 앱 배포 및 정책 관리** **주요 내용** - OCM에서의 PlacementRule, Channel, Subscription 구조 - Rancher에서의 Helm Chart, 앱 배포 방식 이해 - Kyverno를 통한 정책 작성 및 OCM에 적용 - Submariner를 이용한 클러스터 간 서비스 연결 **실습** - OCM 기반 애플리케이션 배포 (PlacementRule 구성) - Rancher에서 Helm Chart 배포 실습 - Kyverno 정책: 특정 Namespace에만 배포 허용 정책 작성 - Submariner를 통한 클러스터 간 ping, curl 테스트
4~5일	**4일차: GitOps 및 CI/CD 연동** **주요 내용** GitOps 개념과 ArgoCD/Fleet 비교 ArgoCD ApplicationSet을 통한 멀티 배포 Tekton 기반 Git → 빌드 → 배포 파이프라인 구성 Rancher Fleet 구조 및 Git 연동 방식 **실습** ArgoCD + ApplicationSet으로 클러스터별 자동 배포 Tekton Pipeline을 구성해 Git 푸시 → 이미지 빌드

	Rancher Fleet Git 연결 및 배포 자동화 클러스터별 버전 구분 배포 전략 (Blue/Green)

5. 표준 KNATIVE

교육명	클라우드 네이티브를 위한 MSA 구현
교육개요	CNCF 및 Knative를 기반으로 직접 서비스 빌드 및 배포를 한다. 표준적인 도구와 구성 기반으로 플랫폼 구성 후, 자동화를 직접적인 인프라 자원 구성 없이 인프라를 구축한다. 개발자는 Knative를 통해서 애플리케이션을 배포 및 활용하는 방법에 대해서 직접 체험/구축한다.
목표	쿠버네티스 인프라 자원을 코드로 자동화한다. MSA기반 애플리케이션 구성 및 애플리케이션 패키징. Knative기반으로 서비스 배포 및 관리한다.
기간	총 4일, 3일로 조정가능 과정
랩	오픈스택 기반, 인터넷 사용 가능한 노트북 요구
교육대상	쿠버네티스 기반으로 MSA서비스 구성 및 운영하는 인프라 관리자 및 서비스 관리자.
1일	쿠버네티스 설치 및 구성 - 간단하게 마스터 및 워크 노드 구성 CI/CD 인터페이스 구성 - ArgoCD - Tekton 이미지 빌드 및 테스트 - buildah 명령어로 컨테이너 패키지 구성 - podman으로 컨테이너 패키지 테스트(로컬 Infra Container, Application Container)
2일	레거시 프로그램 쿠버네티스로 마이그레이션 하기 - 레거시 프로그램을 쿠버네티스 마이그레이션 - 레거시 프로그램을 마이그레이션을 위한 방법 레거시 프로그램을 컨테이너 구조에 맞게 수정 및 변경하기 - 프로그램 수정 및 변경 - 테스트 및 검증하기 MSA 프로그램 구성 - MSA 서비스 구성을 위한 필요한 기술 - 간단하게 MSA기반의 애플리케이션 작성 및 배포
3일	Istio서비스 구성하기 - Istio 서비스 설명 - Istio 서비스 구성을 위한 API 구성 - Istio 서비스 배포 - API서비스 추적 및 스케일링
4일	Knative를 위한 쿠버네티스 구성

	- knative 소개 - knative와 쿠버네티스 서비스 관계 - knative 문법 및 구성 - event driven 서비스 구성 - 오토스케일링 서비스 배포

6. 쿠버네티스 자동화

교육명	**쿠버네티스 자동화 및 DEVOPS**
교육개요	쿠버네티스 기반으로 DevOps를 위한 서비스 설계 및 배포를 직접 진행한다. 사용자는 가상의 부서에서 기본만 구축이 되었는 쿠버네티스 클러스터에 어떠한 방식으로 DEVOPS를 구축하는 직접 경험하면서 애플리케이션 빌드 및 서비스 배포까지 직접 한다. 이 교육은 인프라부터 쿠버네티스까지 자동화를 구성 및 구현한다.
목표	Podman/Kubernetes/Tekton/ArgoCD/GitLab CI/Knative/Prometheus/Grafana 통합 운영 실습
기간	총 4일, 조정가능 과정
랩	오픈스택 기반 랩
교육대상	
1일	**DevOps 기반 환경 구성 및 Podman** 이론 - DevOps 개요 및 컨테이너 기반 개발 흐름 - Podman과 Docker 비교 - 쿠버네티스 클러스터 구성 이행(기 구축 환경 설명) 실습 - 포드만으로 컨테이너 이미지 빌드 및 테스트 - 포드만으로 시스템 서비스 관리(systemd 연동) - 포드만으로 애플리케이션 패키징 및 이미지 푸시(GitLab Container Registry)
2일	**쿠버네티스에 서비스 배포** 이론 - 쿠버네티스 기본 구성 요소 복습(Pod, Service, Deployment) - GitOps개념 및 Helm 소개 실습 - 기본 클러스터에 배포 환경 연동 - YAML파일 기반 수동 배포 - GitLab Container Registry에서 이미지 pull 및 deploy - Helm 차트로 간단한 서비스 배포
3일	**GIt기반 CI/CD파이프라인 구축** 이론 - GitLab CI/CD개요 - .gitlab-ci.yaml 구성 설명 실습

	- GitLab runner 등록 및 연동 - 컨테이너 빌드/테스트/푸시 자동화 - GitLab Registry와 Tekton 연동 준비 - 파이프라인 실행 후 결과 확인(이메일/메신저 알림)
4일	**Tekton, ArgoCD를 이용한 배포 자동화** 이론 - Tekton Pipeline과 Task이해 - AgroCD 구조 및 실시간 GitOps 배포 실습 - Tetkon으로 이미지 빌드 및 서비스 배포 구성 - GitLab -> Tekton -> 클러스터로 이어지는 자동 배포 구성 - ArgoCD로 Git 기반 지소적 배포 자동화 구성 - GitLab Webhook 연동 및 실시간 배포 확인(ArgoCD, Tekton)
5일	**고급 자동화 및 Knative 활용** 이론 - Knative Serving과 Eventing 개요 - Knative와 Tekton 통합 시나리오 - 시스템 알람 고도화 실습 - 모니터링 시스템 구축(Prometheus, Grafana) - Knative로 이벤트 기반 애플리케이션 배포 - Tekton -> Knative 파이프라인 설계 - Podman, Tekton, GitLab ArgoCD, Knative 종합 랩 - 배포 완료 후 직접 알림 시스템 구축

가상화(4일)

1. 표준 Libvirtd 가상화

교육명	**표준 Libvirtd 가상화**
교육개요	리눅스에서 사용하는 QEMU/KVM기반의 가상화는 Libvirtd기반으로 관리 및 운영이 된다. 이 서비스는 독립적으로 사용이 가능하다. Libvirtd에 대한 이해도를 높이고, 이를 통해서 오픈스택 및 쿠버네티스와 같은 가상화 시스템을 운영 시 장애 처리에 도움이 된다. 이와 더불어 표준 가상머신 디스크 관리 도구 사용방법 및 이미지 관리 도구에 대해서도 학습한다.
목표	Libvirtd명령어 및 기능에 대해서 이해한다. 동작 방법 및 기존 오픈스택 및 쿠버네티스와 어떠한 관계가 있는지 확인한다. 표준 도구를 통한 이미지 관리 및 배포 방법에 대해서 학습한다
기간	총 4일, 3일로 조정가능 과정
랩	오픈스택 기반 랩
교육대상	최소 1년 이상의 리눅스 운영 경험이 있는 사용자

	VMware혹은 Xen 가상화에서 QEMU/KVM기반의 가상화로 기술 전환이 필요한 가상화 인프라 엔지니어
1일	**랩 구성 및 설명** **QEMU/KVM 설명** - 동작 방식 및 QEMU/KVM관계 - 설치 및 서비스 확인 - KVM의 역할 - QEMU가 다루는 자원 **Libvirtd 설명** - Libvirtd와 하이퍼바이저 차이점 - Libvirtd 서비스/설정 및 디렉터리 - QEMU/KVM과 관계 **virsh명령어 학습** - 자주 사용하는 명령어 활용 - virsh명령어 기반으로 XML파일 수정 - 자원 조회 및 활용 **저장소 위치 구성** - 기본 저장소 위치 - 저장소 추가 및 드라이버 **네트워크 설정 구성** - Linux Bridge Network - OpenvSwitch - VLAN **가상머신 디스플레이** - VNC - Spice
2일	**가상머신 관리** - 가상머신 생성 - 가상머신 마이그레이션 - 가상머신 디스크 이미지 * 스냅샷 생성 및 관리 * 이미지 편집 및 수정 * 가상머신 템플릿 이미지 * 이미지 저장소 구성 - 가상머신 자동 설정 * cloud-init 설명 * cloud-init 스크립트 및 명령어 **마이그레이션** - 다중 노드에서 Libvirtd기반으로 마이그레이션 구현
3일	**반가상화 드라이버 구성 및 사용**

	- 네트워크 - 스토리지 **Guest Tools 설명** - 가상 자원 생성을 위한 QEMU - 디스크 관리를 위한 도구 - 이미지 배포 및 배포서버 - 자동 이미지 빌드 시스템 **API기반으로 Libvirtd서버 관리 및 서비스 배포** - 다중 노드 기반으로 가상머신 구성 **성능 개선을 위한 설정 및 튜닝** - 네트워크 장치 - 블록 디스크 장치 - 반가상화 장치 - NUMA - KVM KSM - De-Duplicate Block Service (VDO) **libvirtd 및 가상머신 트러블 슈팅**

2. 쿠버네티스 가상화

교육명	쿠버네티스 가상화를 위한 확장
교육개요	현재 컨테이너 기반의 PaaS시스템이 기존 PasS시장을 섭렵하고 있다. 하지만, 여전히 가상화는 컨테이너 시스템보다 장점이 많으며, 서로 다루는 영역이 다르다. 이러한 이유로 가상화 시스템을 쿠버네티스와 통합하는 방법에 대해서 학습한다.
목표	쿠버네티스 이해 및 쿠버네티스 가상화에 대해서 학습
기간	총 4일, 3일로 조정가능 과정
랩	오픈스택 기반 랩
교육대상	최소 1년 이상의 쿠버네티스 운영 경험 및 학습한 사용자 혹은 엔지니어
1일	**kube-virt 소개** - kube-virt는 무엇인가? - kubernetes와 kube-virt의 관계 - kube-virt 아키텍처 및 주요 구성 요소 - 컨테이너+가상머신 통합 **쿠버네티스 설치 및 구성** - kube-virt 서비스 아키텍트 - 설치 전 준비사항 - 쿠버네티스/kube-virt 요구사항 - kube-virt리소스와 스토리지 및 네트워크 - helm기반으로 kube-virt설치 - kubectl으로 kube-virt 자원관리

	Kube-virt환경 설정 및 관라 - 네임스페이스 생성 및 관리 - PV/PVC 구성 및 SC 활용 - kube-virt YAML - kube-virt 리소스 모니터링
2일	**Kube-virt에서 가상머신 생성 및 관리** - KubeVirt의 VM객체와 템플릿 관리 - virtctl, kubectl을 이용한 가상머신 생성 - 가상머신 구성을 위한 YAML 파일 작성 및 배포 - 디스크 이미지 가져오기 및 디스크 관리 **가상머신에서 사용하는 네트워크 관리** - 가상머신 네트워크 관리 - 네트워크 및 가상머신 인터페이스 설정 - 가상머신과 쿠버네티스 클러스터 네트워크 통신 **가상머신 운영 및 관리** - 가상머신 상태 관리 - 가상머신 로그 및 상태 확인(가상머신 자원) - 가상머신 자원 관리(CPU/MEM/DISK등) - 가상머신 스냅샷 및 리스토어 - 가상머신 상태 정보 복사 - 가상머신 롤백
3일	**고급기능** - 가상머신 스케줄링 - 다중 노드에서 가상머신 분배 - GPU 및 특정한 하드웨어 직접 사용하기 및 접근 - 가상머신 템플릿 이미지 생성 및 관리 **보안** - 쿠버네티스 RBAC기반으로 가상머신 자원 보안 - 가상머신 접근 제어(네트워크 및 사용자) - kubevirt 보안 설정(SELinux, AppArmor) **성능 최적화** - CPU 및 메모리 최적화 - 스토리지 성능 최적화 - 가상머신 오버커밋 - 모니터링(Prometheus, Metrics, Grafana) **트러블 슈팅** - 로그 및 이벤트 분석 - H/A 기반으로 가상머신 마이그레이션 - VM 상태 모니터링 및 디버깅

3. 오픈스택 어드민

교육명	표준 오픈스택 서비스 설치 및 구성
교육개요	본 과정은 표준 오픈스택 플랫폼의 설치 및 기본 사용법을 학습하는 입문 과정입니다. 오픈스택의 핵심 구성 요소에 대한 이해를 바탕으로, 클라우드 인프라 환경을 직접 구축하고 운영할 수 있는 기초 역량을 다집니다. 또한, 컨테이너 기반으로 동작하는 오픈스택 컴포넌트의 구조를 이해하고, Kolla 및 Kayobe를 활용한 클러스터 프로비저닝 및 배포 과정을 실습을 통해 익힙니다.
목표	오픈스택 설치를 kolla기반으로 진행한다. 기존에 사용하였던 프로비저닝 시스템인 triple-o는 더 이상 kolla를 부분적으로 사용하기 때문에 적합하지 않는 프로비저닝 시스템이다. 새로운 Kyobe를 통해서 프로비저닝 후, 오픈스택 명령어를 학습한다.
기간	총 5일
랩	오픈스택 기반, 인터넷 사용 가능한 노트북 요구
교육대상	최소 2년의 리눅스 운영 및 사용 경험이 있는 엔지니어 혹은 사용자 아마존 서비스에서 온프레미스 서비스로 환경을 옮겨오려는 서비스 설계자 혹은 엔지니어
1일	**오픈스택 과정 및 랩 소개** - 랩 구성 및 목적 **오픈스택 역사 및 OpenInfra** - 오픈스택의 시작 - 오픈스택과 OpenInfra그리고 CNCF - 현재 오픈스택의 방향과 목적 **설치를 위한 랩 준비 및 구성** - 오프라인 설치를 위한 설명 - Product vs PoC모델 - 운영 및 구성을 위한 기술 **오픈스택 기본 컴포넌트 소개** - Keystone, Nova, Neutron, Cinder, Heat, Glance, Ceilometer **대표적인 확장 컴포넌트 소개** - Octavia, Designated... **오픈스택 CLI 및 Dashboard(horizon) 확인하기** **차세대 UI Skyline과 horizon의 차이점** **오픈스택 설치** - 오픈스택 설치가 가능한 배포판 - 오픈스택과 SELinux 관계 - 클러스터에서 사용할 임시 스토리지 구성 - kolla-ansible 기반으로 오픈스택 설치
2일	**오픈스택 설치** - Podman vs Docker - Bifrost와 Director의 차이점 - kolla-ansible와 openstack-ansible차이점 그리고 활용.

	- 기존 오픈스택과 달라진 설정 및 로그 파일 구성 - 플레이북 설정 및 설치 **오픈스택 keystone 자원 다루기** - 사용자/프로젝트/역할/그룹 생성 및 관리 - 엔드포인트/카달로그와 오픈스택 서비스 관계(service project) - 서비스 계정과 일반 계정의 차이점 - 사용자/프로젝트/그룹/역할 활용하여 계정 구성 **오픈스택 이미지** - 오픈스택에서 말하는 이미지(컨테이너 및 가상머신) - 이미지 빌드를 위한 도구 및 방법 - guestfs-tools와 libvirtd관계 - 이미지 빌드 및 업로드
3일	**오픈스택 네트워크** - 오픈스택에서 제공하는 SDN유형 - OVS/OVN그리고 Appliance Network의 차이 * OVN의 장점과 한계점 * Openvswitch는 왜 사용하는가? * Linuxbridge와 OVS의 차이점 및 관계 - Provider/Flat/Tenant Network 구성 - vlan통한 Provider네트워크 구성 - 오픈스택 네트워크/서브넷 생성 * 네트워크 생성 * 서브넷 생성 * 라우터 구성 * FloatingIP와 IP Address Pairing * nftables/OVS/OVN 사용 시, 자원 구성 차이 - 오픈스택 외부 네트워크 생성 * 오픈스택 외부 네트워크 구성 * Provider/External 네트워크 차이
4일	**오픈스택 블록** - 블록 스토리지 서비스 설명 - 블록 스토리지 백-엔드와 드라이버 타입 - 백-엔드 드라이버 설명 * 왜, LVM2는 진짜로 사용하면 안 되는가? * NFS드라이버는 무엇으로? (NFS vs Ganesha) * 이외 확장한 가능한 스토리지 - 블록 장치 생성 및 할당 그리고 확인 **오픈스택 미터링 서비스** - 미터링 서비스 설명 - 미터링 서비스를 통한 서비스 사용량 확인

	- 미터링 서비스는 어렵다! * Gnocchi 서비스 설명 및 확인 * Panko 서비스 설명 및 확인 * Adho 서비스 설명 및 확인 - 이제는 미터링 그만! * 현대적으로 운영 및 구성하기 위한 Prometheus, Grafana 운영.
5일	**오픈스택 가상머신** - 오픈스택 nova와 nova-api 그리고 nova-conductor의 관계 및 차이점 - 가상머신 구성을 위한 준비 * Flavor * Security Group * Firewall와 Security Group의 차이점 그리고 OVN * 가상머신 관리를 위한 명령어 훑어보기 * libvirt, sasl관계 그리고, 마이그레이션 * libvirt디버깅을 위한 접근 방법 - 가상머신 생성 * 가상머신 콘솔에 접근하기 * 사용이 가능한 콘솔 유형 - 외부에서 접근 가능하도록 FloatingIP할당 **오픈스택 자동화** - Heat 서버스 설명 * Ansible과 Heat의 차이점. * 왜 둘 다 YAML기반으로 사용하는가? - Heat 문법 설명 - Heat 기반으로 간단한 서비스 배포 **오픈스택 서비스 확장** - 서비스 확장을 위한 플레이북 훑어보기 - 서비스 확장 및 확인 - 오픈스택에서 CI/CD **오픈스택 기반 쿠버네티스 구성 및 구축** - 구축 시 필요한 컴포넌트 - 오픈스택 오퍼레이터 - 이해가 필요한 확장 기술 - 쿠버네티스 클러스터 구성

4. Harvester 어드민

교육명	**Harvester 기반 클라우드 네이티브 HCI 운영 실습**
교육개요	Harvester는 Rancher가 주도한 오픈소스 HCI(Hyper-Converged Infrastructure) 플랫폼으로, KVM 기반 가상화와 Kubernetes 통합 환경을 제공한다. 본 교육 과정은

	Harvester 설치부터 VM 배포, 네트워크 구성, Rancher 연동, 멀티 노드 고가용성 구성 및 Kubernetes 통합 운영까지 실무 중심으로 구성되며, 글로벌 트렌드에 따라 GitOps 및 클라우드 인프라 자동화도 함께 실습한다.
목표	- Harvester의 구조 및 작동 원리 이해 - 하이퍼컨버지드 VM 환경 직접 구축 및 운영 - Harvester 내 PXE, ISO, 이미지 저장소 구성 - Rancher 연동 및 RKE2 클러스터 배포 - 멀티 노드 HCI 고가용성 구성 - VM 기반 워크로드와 K8s 네이티브 워크로드 통합 운용
기간	총 5일
랩	오픈스택 기반, 인터넷 사용 가능한 노트북 요구 - 최소 5대 이상의 노드(VM 또는 실 서버) 구성 - PXE + DHCP + TFTP + HTTP 부트 환경 - 외부 DNS, NFS, VM 이미지 저장소, Rancher 설치 환경
교육대상	- HCI 도입을 고려 중인 인프라 관리자 및 DevOps - KVM 기반 VM 운영자에서 클라우드 인프라로 확장하고자 하는 실무자 - Rancher 및 RKE2 환경의 통합 운영을 목표로 하는 기술 리더
1일	**Harvester 이해 및 설치 환경 준비** **이론** Harvester 개요 및 아키텍처 소개 HCI와 전통 가상화(KVM, Proxmox 등)의 차이 PXE 부팅 구조: DHCP/TFTP/HTTP 구성 원리 ISO, Cloud-Init, 이미지 저장소 구조 설명 **실습** PXE 환경 준비 (dnsmasq + httpd + TFTP) Harvester ISO PXE 자동 설치 준비 Harvester 3~5노드 클러스터 설치 실습
2일	**2일차 – VM 인프라 구성 및 스토리지 연동** **이론** Harvester 내 VM 생성 방식 및 Cloud-Init 이해 Virtual IP, VLAN, 브리지 네트워크 구성 내장 Longhorn 스토리지 구조 이해 **실습** Harvester 기반 VM 생성 (Cloud-Init, ISO) ISO/이미지 업로드 및 템플릿 구성 Longhorn 상태 확인 및 볼륨 관리 실습
3일	**Rancher 연동 및 RKE2 클러스터 통합** **이론** Harvester ↔ Rancher 통합 구조 설명 VM 워크로드에서 K8s 워크로드로 확장

	RKE2 클러스터 설치 방식 소개 **실습** Rancher 설치 및 Harvester 클러스터 연결 Harvester UI에서 RKE2 클러스터 생성 생성된 RKE2 클러스터에 GitOps 앱 배포
4일	**네트워크, 고가용성, 정책** **이론** Multus/CNI 연동 구조 설명 LoadBalancer, Ingress 구성 네트워크 격리 및 보안 정책 구성 HA 구성 요소 (ETCD, Harvester HA 구조) **실습** VM에 Static IP 설정 및 외부 통신 확인 RKE2 + Harvester 연계 Ingress 서비스 배포 스토리지 장애/네트워크 장애 복구 실습
5일	**GitOps 및 운영 자동화** **이론** VM 배포 자동화 전략 (Cloud-Init, REST API, Salt 등) GitOps 연계 (ArgoCD + Rancher Fleet) 모니터링/알림/로깅 연동 전략 **실습** Rancher Fleet 기반 GitOps 구성 Git 푸시 → RKE2 + VM 환경 자동 반영 Prometheus + Grafana를 통한 VM/K8s 통합 모니터링 알림: Alertmanager 또는 웹훅 기반 Slack 연동

자동화(4일)

1. 커뮤니티 앤서블 기본 과정

교육명	커뮤니티 앤서블 기본과정
교육개요	앤서블 처음 사용자를 위한 교육. 앤서블 설치 및 활용 방법에 대해서 랩 기반으로 학습한다.
목표	앤서블 YAML문법 학습 앤서블 명령어 사용 방법 플레이북 및 역할(role) 작성방법 및 활용 앤서블 프레임워크 디렉터리 및 컬렉션 앤서블 모듈 활용
기간	총 4일, 3일로 조정가능 과정
랩	오픈스택 기반, 인터넷 사용 가능한 노트북 요구

교육대상	앤서블 기반으로 리눅스 및 윈도우 시스템 자동화 플레이북 기반으로 스크립트 작업을 앤서블로 마이그레이션 ROLE기반으로 기본적인 대용량 작업을 분리 작성 및 구성
1일	YAML작성을 위한 에디터 설정 YAML/JSON/TOML의 차이점 그리고 적용 부분 앤서블 명령어 사용 방법(애드혹, 플레이북, 네비게이터) 인벤토리 구성 및 설정 시스템 변수 및 앤서블 변수 관리 인벤토리 호스트 선택(패터닝)
2일	주요 모듈 사용 방법 및 다중 플레이북 작성 - 앤서블에서 많이 사용하는 모듈 사용 방법(copy, shell, register) - 변수 선언 및 활용 방법 - 시스템 변수 및 앤서블 Facts - 조건문 및 오류 핸들링 다중 플레이북 및 ROLE 구성 및 작성 - ROLE 구성을 위한 프레임워크 디렉터리 구성 - 템플릿 구성을 위한 Jinja2 문법 학습 앤서블 컬렉션 - 앤서블 컬렉션 검색 및 설치하기 - 사용자 컬렉션 구성 및 배포하기
3일	플레이북 기반 리눅스 서비스 구성 - systemd 서비스 관리 - RPM패키지 관리 및 저장소 배포 - 디스크 파티션 및 파일 시스템 생성 - 네트워크 인터페이스 구성 및 관리 - 방화벽 관리 - SELinux 컨텍스트 및 레이블링 구성 네트워크 설정 및 구성을 위한 플레이북 - NetworkManager - Netplan - General Network - teamd - Linux Bridge Podman위한 앤서블 플레이북 생성 - 컨테이너 기반 통합 애플리케이션 배포 - 배포된 서비스에 대한 검증 및 리포트 작성
4일	앤서블 YAML파일 암호화 및 보호하기 Podman위한 앤서블 플레이북 생성 - 컨테이너 기반 통합 애플리케이션 배포 - 배포된 서비스에 대한 검증 및 리포트 작성 앤서블 기반으로 리눅스 튜닝 배포 - kernel parameter - module parameter Libvirtd위한 앤서블 플레이북 생성

	- 가상머신 이미지 배포 - 스냅샷 생성 및 관리 - 마이그레이션

2. 쿠버네티스 설치 자동화

교육명	쿠버네티스 설치 자동화
교육개요	쿠버네티스 설치는 기본적으로 kubeadm명령어를 통해서 부트스트랩 진행한다. 앤서블을 활용하여 쿠버네티스 설치 및 서비스 배포 자동화가 가능한지 확인한다.
목표	앤서블 및 kubeadm를 사용하여 쿠버네티스 클러스터 설치 및 구성을 자동화한다. 이를 통해서 운영자 개입 없이, 서비스 구성이 가능한지 확인한다.
기간	총 4일, 3일로 조정가능 과정
랩	오픈스택 기반, 인터넷 사용 가능한 노트북 요구
교육대상	쿠버네티스 클러스터 설치 및 서비스 배포 자동화
1일	kubeadm명령어 및 기능 확인 사용할 명령어 및 설치 절차 정리 kubeadm Configuration 확인 - InitConfiguration - ClusterConfiguration - KubeletConfiguration - KubeProxyConfiguration - JoinConfiguration Kubeadm Configuration 작성 - 기본 설정 파일 작성 및 테스트 - 앤서블 템플릿 형식으로 변경 시스템 영역 - 운영체제 배포 준비 - 저장소 구성(RPM 및 컨테이너 이미지) - 설정 파일 수정 내용 확인 - 모든 작업 내역 앤서블 플레이북으로 작성
2일	시스템 영역(계속) kubeadm/OS 구성 - 작성된 템플릿 기반으로 컨트롤 플레인 구성 확인 - 추가적인 Kubelet설정 확인 - 클러스터 구성 확인 - 컴퓨트 노드 구성 및 클러스터 추가 인프라 노드 확장 - 스토리지 노드 구성(NFS/GlusterFS) - 라이브러리 노드 구성(rpms, container images)
3일	기본 서비스 설치 및 확장

	- HelmChart - kubernetes operator - 앤서블로 전환 및 통합 쿠버네티스 표준 패키지 설치 - 네트워크 기능 (Flanned, calico) - 클러스터 모니터링 - StorageClass (NFS, GlusterFS) - 앤서블로 전환 및 통합 roles통합 - kubeadm/OS구성 - 인프라 노드 - 기본 서비스 설치 및 확장 - 쿠버네티스 표준 패키지 설치
4일	확장 클러스터 패키지 구성 - 로드밸런서 - 인그레스 서비스 - 클러스터 라이브러리 포드(registry, git) - CI/CD(Tekton, ArgoCD) - Knative - 앤서블로 전환 및 통합 배포용 애플리케이션 구성 - 테스트용 애플리케이션 아키텍처 확인 - 패키징 및 배포 테스트 - 앤서블로 배포파일 구성 및 테스트 최종 통합 - 쿠버네티스 클러스터 설치 통합 실행 - 배포된 클러스터 애플리케이션 배포 실행

3. 테크톤 활용한 DEVOPS

교육명	**CNCF 테크톤을 활용한 DEVOPS**
교육개요	**"YAML 기반 Kubernetes 네이티브 CI/CD – Tekton 실습 중심 통합 과정"** 쿠버네티스 시대의 표준 CI/CD는 이제 **Tekton**입니다. 본 과정은 기존 Jenkins 기반의 복잡한 플러그인 구조에서 벗어나, **Google, Red Hat, SUSE** 등 업계 리더들이 공동 개발 중인 **CNCF 공인 Tekton**을 기반으로 한 **클라우드 네이티브 CI/CD 운영 체계**를 실습 중심으로 학습합니다.
목표	Jenkins/Jenkins X에서 Kubernetes 기반 CI/CD로 전환을 고려 중인 분 YAML 기반으로 파이프라인을 관리하고 싶은 GitOps 사용자

	Tekton의 구조, 실행 모델, 외부 Git 서버(SCM)와의 통합 연동이 궁금한 분 Tekton을 기반으로 한 **경량화되고 확장 가능한 DevOps 자동화 환경**을 구축하고자 하는 분
기간	총 5일, 4일로 조정가능 과정
랩	오픈스택 기반, 인터넷 사용 가능한 노트북 요구
교육대상	최소 1년 이상의 쿠버네티스 운영 경험 및 학습한 사용자 혹은 엔지니어
1일	**랩 소개** - 하이퍼브이 기반 - 오픈스택 기반 **쿠버네티스 설치 및 테크톤 설치** - 자동화를 통한 쿠버네티스 설치 - 테크톤 설치 **테크톤 탄생 및 소개** - 쿠버네티스 혹은 Cloud Native에서의 CI/CD입장 **테크톤 기본 블록 문법** - "핼로우 월드" 작성하기 **테크톤 태스크 설명**
2일	**테크톤 태스크 설명(계속)** - 태스크(tasks) 및 스탭(steps)이해하기 - 첫 태스크 작성하기 - 태스크 파라메타 구성 및 사용하기 - 데이터 공유(쿠버네티스 볼륨) **테크톤 대시보드** - 대시보드 설치 및 접근 **태스크 디버깅** - tkn 명령어를 통한 자원 디버깅 - kubectl 명령어를 통한 자원 디버깅 **GIT/REGISTRY 서버 구성** - Gogs 기반으로 GIT OPs서버 구축 - docker-registry 기반으로 컨테이너 레지스트리 서버 구성
3일	**테크톤 파이프라인 작성하기** - 파이프라인 설명 - 첫 번째 파이프라인 작성 - 파이프라인 파라메터 - 파이프라인 순서 지정 및 "파이프라인 런즈(runs)"를 통한 결과 확인 **워크스페이스** - 워크스페이스를 통한 데이터 공유 - 데이터 공유가 가능한 쿠버네티스 자원(storageclass,pv,pvc) - 파이프라인을 통한 영구자료(Persistent Data) - 워크스페이스 사용하기 - 워크스페이스 템플릿 생성

4일	**조건식(operator)** - 테크톤 조건식 설명 - 조건식 지시자 - 조건식 및 파라메터 **인증 처리** - 테크톤 인증 처리 설명 - 기본 인증 - SSH 인증 - 컨테이너 레지스트리 인증
5일	**트리거** - 테크톤 트리거 설명 및 설치 - 클러스터에 트리거 설정 - 트리거 템플릿/바인딩 및 이벤트 리스너 - 트리거 템플릿 기반으로 트리거 생성 - 트리거 및 웹훅 **ArgoCD와 연동 및 설정** - ArgoCD 설명 - ArgoCD 설치 - ArgoCD와 Tekton의 차이점 - ArgoCD와 테크톤 연동 및 배포

4. Salt 자동화

교육명	**인프라 자동화를 위한 SaltStack 실전 과정**
교육개요	SaltStack(Salt)은 대규모 인프라를 효율적으로 관리하고 자동화하기 위한 오픈소스 구성 관리 도구입니다. 본 과정에서는 Salt의 핵심 구성 요소와 사용법을 실습 중심으로 학습하며, Salt SSH 기반의 Agentless 방식도 함께 다룹니다.
목표	- Salt의 아키텍처 및 핵심 개념 이해 - 상태 기반 인프라 자동화 구성 - Salt Master/Minion 설치 및 설정 - Agentless 방식(Salt SSH) 활용법 습득 - Jinja 템플릿과 Pillar를 통한 유연한 배포 - Salt Reactor, Beacon을 이용한 이벤트 기반 자동화 구현
기간	4일 또는 5일
랩	Master-Minion 구성, 상태 적용 실습, 이벤트 자동화 실습 등 실습 위주
교육대상	시스템 관리자, DevOps 엔지니어, 자동화 도구에 관심 있는 개발자 및 운영자
1일	✅ **1일차: Salt 입문과 구조 이해** SaltStack 개요 및 구성 요소 (Master, Minion, Returner 등) Salt 설치 및 환경 구축 (Podman 또는 VM 기반 실습) Salt 통신 방식 (ZeroMQ, RAET 등)

	기본 명령어 사용 (salt, salt-call, test.ping, cmd.run 등) 실습: Master-Minon 구축 및 ping 테스트
2일	✅ **2일차: 상태 기반 관리와 Jinja 템플릿** Salt State 개념 및 구조 (.sls, top.sls) 패키지 설치, 파일 배포, 서비스 관리 등 상태 관리 실습 Jinja 템플릿 문법 소개 및 활용 실습: 사용자 계정, 패키지, 서비스 상태 정의
3일	✅ **3일차: 고급 기능 (Pillar, Grains, Orchestration)** Grains와 Pillar를 통한 정보 분리 및 환경 구성 Orchestration: 여러 노드 제어 워크플로우 구성 Salt SSH를 활용한 Agentless 자동화 방식 SSH 방식 장단점 및 운영 시 고려사항 실습: SSH 기반 상태 실행, hybrid 환경 구성
4일	✅ **4일차: 이벤트 기반 자동화와 외부 시스템 연동** Salt Reactor와 Beacon을 이용한 이벤트 기반 자동화 Salt와 외부 시스템 연동: REST API, Git 연동 실습: 로그 이벤트 발생 시 자동 조치, GitHub 기반 상태 배포 마무리 프로젝트: 자동화된 웹 서버 환경 구성 및 실시간 이벤트 대응

5. 오픈소스 테라폼

교육명	**OpenTofu(테라폼 기초 과정)**
교육개요	테라폼의 오픈소스 버전인 오픈토푸의 기본적인 문법 및 활용 방법에 대해서 학습한다. 현재 리눅스 파운데이션은 공식적으로 오픈토푸을 기본 자동화 도구로 지원하고 있다.
목표	- OpenTofu 문법 및 기본적인 사용 방법 - OpenTofu 아키텍처 이해 - 운영체제 애플리케이션 설치 및 설정 - 하드웨어(디스크·및·네트워크) 설정
기간	총· 4일·조정가능·과정
랩	오픈스택 기반, 인터넷 사용 가능한 노트북 요구
교육대상	DevOps 초급자, 인프라 엔지니어, 클라우드 초급 관리자 등
1일	IaC 개념 소개 및 OpenTofu 등장 배경 HCL(HashiCorp Configuration Language) 문법 이해 OpenTofu 설치 및 환경 구성 실습 첫 번째 .tf 파일 작성 및 리소스 배포 (local file 등)
2일	**Provider** 구조와 블록 이해 Variable과 Output 설정 실습 상태 관리(State)와 tofu state 분석 terraform.tfstate와 백엔드 개념

3일	모듈(Module) 구조와 재사용성 실습 Git 기반 모듈 관리 외부 변수 파일 활용 및 워크스페이스(Workspace) 실습 디버깅 및 로깅 (TF_LOG, plan, validate)
4일	OpenTofu를 활용한 AWS/OpenStack 리소스 실습 네트워크 구성, 가상 머신 배포 CI/CD 연동 개요 (예: GitHub Actions, GitLab CI 등과의 연결) OpenTofu + Ansible 연동 구조 소개

퍼블릭 클라우드(3일)

1. AWS 기본 서비스(업데이트 중)

교육명	아마존 기본 서비스
교육개요	대표적인 퍼블릭 클라우드 서비스인 아마존 AWS기반으로 어떠한 서비스를 구성 및 구축할 수 있는지 확인한다. 이를 통해서 데이터센터에서 구성하는 인프라와 퍼블릭 클라우드의 차이점을 확인한다.
목표	아마존 AWS 기본적인 서비스를 이해한다. IDC와 동일하게 인프라 구성 및 구축한다. 기본적인 서비스 이해를 한다.
기간	총 4일, 3일로 조정은 가능하나, 몇 교육 모듈은 제외해야 됨.
랩	아마존 계정 필요
교육대상	1년 미만 리눅스 사용경험이 있는 엔지니어 혹은 개발자 아마존 퍼블릭 클라우드 서비스에 대해서 관심이 있는 엔지니어 혹은 개발자
1일	**클라우드 및 AWS소개** - 클라우드 개념 및 AWS서비스 소개 - 서비스 가입과 웹 콘솔 사용 **컴퓨팅 서비스** - EC2 가상머신 서비스 - 인스턴스 유형과 비용 - 가상머신 연결 및 작업 - 외부에서 서비스 요청
2일	**아마존 기본 보안** - IAM 서비스로 사용자 계정 관리 - 권한 정의 및 제어(RBAC) **네트워크** - VPC 기반으로 망 분리(Public/Private) - 방화벽으로 트래픽 제어 - 고정 IP로 지속적인 통신 구현
3일	**AWS Storage S3** - Simple Storage Service -S3 - Buket으로 데이터 저장 - 정적 웹 호스팅 서비스 구현 - Storage Class(Glacier 서비스 통합) **AWS EBS/EFS** - 블록 스토리지 구현 - 공유 스토리지 기반으로 데이터 공유
4일	Serverless 시스템 - Serverless Platform 소개 - Lambda서비스 운영 자동화

	데이터베이스 - 다양한 데이터베이스 서비스 - RDS기반으로 데이터베이스 구성 및 설정
5일	**모니터링** - CloudWatch로 모니터링 - Metric 및 Alarm서비스 **서비스 스케일링** - 오토 스케일링 서비스 - ELB(로드 밸런서)와 부하 분산 - 고가용성(High Availability) 서비스 구성

2. AWS 기본 보안(업데이트 중)

교육명	**아마존 기본 보안**
교육개요	아마존의 기본 서비스를 구성 및 구현 후, 각각 구성요소에 어떠한 방식으로 보안 적용이 가능한지 확인한다.
목표	좀 더 안전하게 아마존 자원을 운영 및 관리하기 위해서 사용자 권한 설정 및 할당 및 자원 관리에 대해서 학습한다. 기존에 구성된 자원에 어떠한 영역까지 보안 설정이 가능한지 확인 후 직접 적용한다.
기간	총 4일, 3일로 조정은 가능하나, 몇 교육 모듈은 제외해야 됨.
랩	아마존 계정 필요
교육대상	1년 미만 리눅스 사용경험이 있는 엔지니어 혹은 개발자 아마존 퍼블릭 클라우드 서비스에 대해서 관심이 있는 엔지니어 혹은 개발자
1일	아마존 기본 서비스 구성 - VPC - Subnet생성 - Internet Gateway - NAT Gateway 정책보안 - 사용자 관리 - 정책 할당 및 구성 - 인증서 조정
2일	VPC보안 - 네트워크 ACL - 보안그룹 - 도메인 보안 - 방화벽 - 엔드포인트 - 웹 보안 - ELB/CloudFront/WAF - TLS(Private)

	데이터보안 - EBS - S3 데이터베이스 보안 - RDS
3일	모니터링 - CloudWatch - CloudTrail

3. AWS ML시스템 구성 및 활용하기(업데이트 중)

교육명	**아마존에서 ML시스템 구성 및 활용하기**
교육개요	아마존에서 SaaS형태로 제공해주는 ML서비스 사용 및 활용 방법에 대해서 학습한다. 아마존에서 제공하는 ML SaaS에 대해서 알아보고, 어떠한 방법으로 사용하는지, 그리고 기존 베어메탈 혹은 호스티드(hosted) 서비스와 어떠한 차이점이 있는지 확인한다.
목표	아마존에서 제공하는 AI SaaS서비스 확인 및 학습한다. 아마존에서 제공하는 ML SaaS서비스 확인 및 학습한다. 위의 두 가지 AI/ML서비스를 사용하여 간단하게 서비스 구성 및 활용한다.
기간	3일
랩	아마존 계정 필요
교육대상	아마존 기반으로 AI/ML서비스를 처음 접하는 소프트웨어 사용자.
1일	아마존 머신러닝 서비스 소개 ML서비스와 아마존 EC2/AIM 설정 머신 러닝을 위한 모델링 설명 ML서비스 사용 시, 비용 발생 예측 EC2기반으로 ML구현 및 사용하기 위한 Cloud9 환경 테스트
2일	서버리스 기반으로 데이터 관리(SageMaker) 실용적인 데이터 분석 및 처리 학습용 데이터 처리 및 SageMaker Modeling Registry SageMaker Model모델링 모니터링 및 스케줄링 캡쳐된 데이터 분석하기
3일	ML를 위한 보안 및 거버넌스 그리고 규정준수 데이터 보안 및 사생활 모델링 Amazone EKS기반으로 Kubeflow 구현 SageMaker 파이프 라인 구성

4. ONE DAY AWS AI

교육명	**아마존 Sagemaker를 통한 AI맛보기.**
교육개요	현재 대다수 서비스 및 하드웨어는 인공지능과 융합이 되고 있는 사회이다. 이러한 인공지능 서비스에 대해서 쉽게 접근 및 이해하기 위해서 아마존 서비스 기반으로 간단하게 AI서비스를 구축 및 구성한다. 이 랩은 최소 코드로 진행하는 AI 랩.

목표	AWS 기반으로 간단하게 AI서비스 구성 및 맛보기
기간	1일
랩	AWS 개인 계정이 필요
교육대상	AI에 관심이 있는 사용자 코드를 잘 모르는 사용자
1일	AWS AI/ML 서비스 소개 - AI/ML의 기본개념: 머신러닝/딥러닝/모델 훈련 - Amazone Sage maker, Recognition, Lex, Polly, Comprehend - 클라우드 AI/ML과 온 프레미스의 AI/ML의 차이점 AI/ML 모델 학습과 배포 - Amazone SageMaker기반으로 준비 - 데이터 준비: S3를 통해서 SageMaker에 연결 - 모델 훈련: SageMaker의 내장 알고리즘을 사용하여 기본 모델 학습 - 모델 배포: 훈련된 모델을 기반으로 엔드포인트 배포 및 예측 수행 - 모델 튜닝 및 최적화(SageMaker, 하이퍼 파라메터 튜닝 기반으로 튜닝) AI 모델 추론 및 운영 - 모델 배포 후, 실시간 예측 - 모델 모니터링 방법

네트워크

1. OVS/OVN 기반 인프라 엔지니어 네트워크 실무 교육 커리큘럼

교육명	**OVS/OVN 기반 인프라 네트워크 실무 마스터 과정**
교육개요	OVS 및 OVN의 내부 구조와 사용법을 중심으로 OpenStack 및 Kubernetes 환경에서의 SDN 네트워크 구성, 보안 제어, 트러블슈팅까지 실습 중심으로 다루는 과정
목표	- OVS의 구조 및 구성법 이해 - OVN의 L2/L3 구성 및 정책 설정 - OpenStack과 OVN의 통합 구성 실습 - Kubernetes-OVN 연동을 통한 CNI 제어 이해 - ACL, NAT, Flow 분석 기반 트러블슈팅 역량 확보
기간	총 5일, 4일 조정가능. 다만, kubernetes-ovn 제외
랩	가상화 기반 실습 환경 (OpenStack VM 기반 OVN 구성 / Kubernetes 클러스터 + OVN-CNI)
교육대상	네트워크 인프라 엔지니어, OpenStack/Kubernetes 운영자, 클라우드 네트워크 설계자
1일	**OVS 구조 및 VLAN 실습** - OVS vs Linux Bridge 구조 비교 - Bridge / Port / VLAN / Bonding 구성 실습 - 주요 명령어: `ovs-vsctl`, `ovs-ofctl` **Mini Lab:**

	- `br0` 브릿지를 만들고 eth1 포트를 추가 - VLAN 10, 20 설정 후 VM 간 통신 테스트 **VM 실습:** - VM1 (vlan10): 192.168.10.10 - VM2 (vlan20): 192.168.20.10 - ping 테스트로 브로드캐스트 도달 확인
	OVN 논리 네트워크 구성 - OVN 아키텍처 이해 (northd, NB/SB DB) - Logical Switch, Logical Router 실습 - ACL 정책 기본 구조 구성 Mini Lab: - `ls0`, `lr0` 생성 및 포트 연결 - ACL로 ping 차단 후 `ovn-trace` 확인 VM 실습: - VM1: Logical Switch A - VM2: Logical Switch B - 라우터 연결로 통신 시도 후 ACL 적용 여부 확인
	Geneve 터널 구성 및 MTU 확인 - NAT(SNAT/DNAT) 흐름 및 Floating IP 이해 - 흐름 분석 도구: `ovn-trace`, `ovn-sbctl` Mini Lab: - 외부 연결을 위한 NAT 구성 - `ovn-trace`로 8.8.8.8 ping 트래픽 흐름 분석 VM 실습: - 내부 VM에서 외부로 ping 후 Floating IP 적용 - 외부에서 VM으로 ssh 시도 → DNAT 검증
	Day 4: ACL 고급 정책 및 장애 처리 - ACL 우선순위와 예외 규칙 구성 - 장애 시 흐름 차단/허용 정책 실습 - 정책 오류 및 흐름 복원 실습 **Mini Lab:** - 특정 IP에 대해서만 HTTP 허용 ACL 구성 - 정책 우선순위에 따라 흐름 차단 테스트 **VM 실습:** - VM1: 허용 IP - VM2: 차단 대상

	- curl로 통신 여부 확인 및 로그 분석
	Kubernetes OVN 연동 - OVN-Kubernetes 구성 요소 이해 - OVN-K8s pod ↔ pod, pod ↔ external 흐름 분석 - K8s NetworkPolicy 구성 실습 Mini Lab: - allow-http NetworkPolicy 적용 후 결과 비교 - pod 간 통신과 외부 연결 차단 여부 확인 VM 실습: - pod-A: frontend (80 open) - pod-B: backend (deny all) - `kubectl exec`로 HTTP curl 확인

인프라 AI/ML

1. OpenAI를 활용한 인프라 관리 및 분석

교육명	**OpenAI를 활용한 인프라 관리 및 분석**
교육개요	OpenAI를 기반으로 일반적인 인프라 시스템 관리에 어떠한 방식으로 활용이 가능한지 알아본다. 대다수 리눅스 기반의 시스템들은 systemd-journald, syslogd를 통해서 이벤트 기록을 남기며, 많은 양의 이벤트가 발생하였을 때, 분석하기가 어려운 부분이 있다. 이를 OpenAI를 통해서 손쉽게 분석 및 조회할 수 있는지 알아본다.
목표	생성형 AI 및 ML과 어떠한 차이점이 있는지 확인한다. OpenAI의 간단한 활용 방법을 학습한다. API를 통해서 시스템에서 발생한 기록을 OpenAI를 통해서 분석한다.
기간	1일
랩	리눅스 가상머신 그리고, OpenAI 계정
교육대상	OpenAI기반으로 인프라 관리 및 활용에 어떠한 도움이 되는지 알고 싶은 인프라 엔지니어
1일	OpenAI 소개 및 서비스 등록 및 설명. - OpenAI에서 제공하는 서비스 확인 - 기본적인 API호출 방법 웹에서 기본적으로 활용방법 연습. - 파이썬 코드 기반으로 OpenAI와 연결- - 파이썬 코드로 OpenAI API를 호출 및 데이터 업로드 구성 - systemd-journald, syslogd로그 파일 - OpenAI를 통하여 로그 파일 조회 - OpenAI를 기반으로 쿠버네티스 자원 구성

2. Llama.cpp기반 GPT구성 및 활용(준비중)

교육명	**llama.cpp기반으로 CPU기반ㅇ LLM모델 구현**
교육개요	많은 회사들이 GPU기반으로 LLM모델을 구성 및 구현하고 있다. 하지만, 모든 사용자가 LLM를 위해서 값비싼 GPU를 사용하기가 어렵기 때문에, CPU 혹은 저성능의 GPU기반으로 LLM모델을 구성하는 방법을 학습한다. 이를 기반으로 간단하게 프롬프트 구성 및 활용을 한다.
목표	LLM을 CPU기반으로 구성한다.
기간	3일
랩	오픈스택 기반, 인터넷 사용 가능한 노트북 요구
교육대상	CPU기반으로 LLM모델에 관심있는 엔지니어
1일	LLM과 llama.cpp 설명 - LLM개념 및 활용 - llama.cpp의 특징 - GPU 및 CPU모델의 차이점 - llama.cpp의 내부 구조 Llama.cpp 설치 및 환경설정 - 필수 패키지 및 의존성 확인 - 왜, 컨테이너 기반으로 설치하지 않는가? - llama.cpp설치 - CMake통한 컴파일 및 빌드 - 모델 파일(.gguf) 내려받기 및 저장 기본적인 모델 실행 - llama.cpp를 통한 실행 - 기본적인 프롬프트 입력과 출력 - 주요 옵션 확인 - 모델 비교 및 실행(7B, 13B, 30B) 모델 튜닝 및 입력 최적화 - 토큰 개념 및 이해 - --temp, --top_k, --top_p 실험 및 조절 - 단어 반복 방지 –repeat_penalty - 채팅 모드 및 스트리밍 출력 설정
2일	CPU환경에서 성능 최적화 - Thread 개수 설정 - Batch 크기 조정 - NUMA 메모리 최적화 양자화(Quantization)방식 및 활용 - QK_K_M, Q5_0, Q8_o

	- 양자화된 모델 실행 성능 비교 외부 API 연동 - 파이선을 통한 llama.cpp - OpenAI 형식으로 llama.cpp사용 및 활용하기 - FastAPI를 통해서 구성 - ollama와 비교 및 활용 Embedding통한 외부 데이터 연동 - RAG기반으로 llama.cpp활용하기 - 임베딩 활용하기 - 검색 기반 답변 시스템 구축(chromadb, faiss) - PDF/TXT 문서 내용을 llama.cpp로 검색
3일	챗봇 구축 - CLI기반으로 작성 - Flask/llama.cpp로 기반으로 간단한 챗봇 구현 - steaming모드 활용 및 프롬프트 엔지니어링

스토리지

1. CEPH 기초

교육명	CEPH기초 과정
교육개요	오픈소스에서 유일하게 커뮤니티 및 기업용 환경에서 인증 받은 Ceph오브젝트 스토리지에 대해서 설치 및 운영 방법에 대해서 학습한다. Ceph사용자는 보통 오브젝트 복제와 CephFS에 대해서 관심이 많다. 이를 어떠한 방식으로 구성 및 활용하는지 학습한다.
목표	Ceph 스토리지 설치 OSD/MDS/MON 그리고 Ceph스토리지 아키텍처 확인 스토리지 인증 및 사용자 관리 Object/CephFS(fuse) 직접 구현 및 활용
기간	총 4일, 3일로 조정은 가능하나, 몇 교육 모듈은 제외해야 됨.
랩	오픈스택 기반, 인터넷 사용 가능한 노트북 요구
교육대상	CEPH에 관심있는 스토리지 엔지니어 클라우드 시스템에 오브젝트 스토리지 구성이 필요한 사용자
1일	**Ceph 개념 및 아키텍처** - 분산 스토리지의 장점 - Ceph에서 제공하는 CephFS - Ceph알고리즘(rados, crush) - 클러스터 설계 규칙 **Ceph 설치** - 저장소 살펴보기(배포판 별) - 패키지 설치 * Ceph RPM 혹은 DEB * cephadm도구와 앤서블 관계 * Ceph Cluster 설치 - Ceph에서 제공하는 API * 쿠버네티스에서 Ceph * 오픈스택에서 Ceph - 쿠버네티스 기반으로 Ceph설치 * okd에서 ceph스토리지 구성(데모, 실습 없음)
2일	**Ceph 자원 단위** - Ceph PG - PG계산 방식 및 적용 - Ceph Pool과 PG 관계 - CRUSH 설정 - OSD, MDS, MON의 기능 및 설명 **Ceph 명령어** - 기본적인 명령어 사용 방법

	* 클러스터 자원 조회 * 클러스터 OSD자원 조회 * 클러스터 pool자원 조회 **Ceph 사용자** - Ceph 사용자 생성 및 관리 - Ceph에서 제공하는 권한/접근 제어 정책 **Ceph Pool** - Ceph Pool개념 - Pool과 OSD의 관계 **OSD 확장하기** - OSD 확인 - OSD 확장 및 확인 **OSD 기반으로 자원 관리** - OSD 기반 자원 관리 (업로드/조회 등) - 자원 조회 및 접근
3일	**Ceph 블록 스토리지** - RADOS Block Device 설명 * POSIX 호환성은 무엇인가? - OSD와 RADOS Block Device 차이점 - RADOS기반으로 장치 생성 및 구성 - RADOS와 CephFS의 연관 구조 이해 * CephFS의 FUSE 기반 마운트 방식 이해 * FuseFS와 Ceph관계 * MDS와 OSD/MON 그리고 RADSO * CephFS기반으로 블록장치 구성 및 연결 - RADOS Gateway * RADOS 게이트웨이 설명 * RADOS기반으로 자원 활용
4일	**스토리지 통합** - 오픈스택에서 Ceph스토리지 사용 - 쿠버네티스에서 Ceph스토리지 사용 **모니터링** - 클러스터 상태 모니터링 **튜닝** - 수정 가능한 커널 파라메터 - 수정 가능한 네트워크 파라메터 - 수정 가능한 Ceph Server/Client

2. GlusterFS 기초

교육명	GlusterFS기초 과정
교육개요	오픈소스에서 많이 사용하는 NAS혹은 SDS(Software Defined Storage)의 대표적인 GluterFS에 대해서 학습한다. 다른 스토리지보다 느리다는 평가가 있지만, NFS기반에서 활용도가 높으며, OpenStack 및 Kubernetes계열 및 BareMetal 환경에서도 낮은 장애로 신뢰성이 높은 SDS 오픈소스 저장소이다.
목표	GlusterFS설치 GlusterFS기반으로 Brick생성 및 구성
기간	총 3일, 4일로 권장
랩	오픈스택 기반, 인터넷 사용 가능한 노트북 요구
교육대상	SDS 스토리지 엔지니어 Kubernetes 및 OpenStack에서 NAS 스토리지 구성 NFS4기반의 pNFS 시스템이 필요한 스토리지 엔지니어
1일	**GlusterFS 무엇인가?** - GlusterFS 아키텍처 - GlusterFS에서 사용하는 표준 오픈소스 컴포넌트 - GlusterFS와 NFS/CephFS/GFS2의 차이점 - 다른 NAS와 GlusterFS와 비교 **GlusterFS 설치** - 사용가능한 운영체제 확인 - 설치방법 및 구성 **GlusterFS 명령어** - gluster 명령어 사용
2일	**블록 생성 및 구성** - GlusterFS에서 제공하는 Brick/Volume그리고 GlusterFS Daemon - 기본 볼륨 및 블록 생성하기 - 볼륨 디스크 연결 및 구성 **볼륨 관리** - 구성된 볼륨 기반으로 복제 및 백업 - 볼륨 확장 및 축소, 그리고 리밸런싱 **브릭/블록 복제** - 볼륨 복제 설명 - Geo Replication 복제 - Geo Replication 확인 **볼륨 쿼터** - 볼륨 쿼터 설명 - 볼륨 쿼터 설정 및 확인
3일	**브릭 추가 및 제거** - 브릭과 노드의 관계

	- 노드추가 및 브릭 확인 - 노드제거 및 브릭 확인 **배포 및 복제 설정** - 클러스터 노드 복제 구성 * 배포 브릭 구성 - 분산 브릭 구성 **클러스터 모니터링** **웹 관리자 페이지 접근**
4일	**고급 기능 및 컨테이너 연동** - GlusterFS 성능 튜닝 - 캐시, ping-timeout 등 성능 옵션 확인 및 설정 - 장애 복구 실습 - split-brain 재현 및 해결 - brick 삭제 후 복구 실습 **Kubernetes 연동** - Heketi 구성 - Kubernetes StorageClass 생성 및 PVC 테스트 **RESTful API 구성** - Heketi API 연동 실습 - CLI 외 볼륨 생성 자동화

언어

1. GO 언어 기초

교육명	**GO 언어 기초**
교육개요	Go 언어의 문법, 병행성, HTTP 서버 개발까지 기본기를 익히고, 실습으로 간단한 블로그 애플리케이션을 개발하는 4일 과정
목표	- Go 언어 문법 및 구조 습득 - 병행성(concurrency) 프로그래밍 이해 - HTTP 기반 서버 구현 능력 확보 - 파일 기반 블로그 프로젝트 완성
기간	4일 (총 24시간, 1일 6시간 기준)
랩	오픈스택 기반, 인터넷 사용 가능한 노트북 요구
교육대상	프로그래밍을 처음 접하는 사람, 백엔드 입문자, 시스템 프로그래밍에 관심 있는 분들
1일	Go 설치 및 개발환경 구성 기본 문법: 변수, 상수, 타입, 조건문 패키지 구조 이해 간단한 CLI 출력 실습 🛠️ [미니 프로젝트 시작] →"블로그 글 목록 CLI" 만들기 (글 데이터는 slice로 저장)

2일	함수, 배열, 슬라이스, 맵 구조체와 메서드 활용 파일 읽기/쓰기 기본 실습 🛠️ [미니 프로젝트 계속] →"글 작성/저장 기능" 추가 (파일 기반 저장소 구현)
3일	인터페이스와 타입 임베딩 고루틴과 채널, select 문 🛠️ [미니 프로젝트 계속] →"병렬 글 검색 기능" 구현 →"CLI 기반 간단 검색 기능" 추가
4일	net/http 패키지 기본 사용법 REST API 구조 이해 및 구축 JSON 응답 및 상태코드 처리 🛠️ [미니 프로젝트 완성] → "HTTP 서버 기반 블로그 목록/쓰기 API" 완성 → Postman 또는 curl로 테스트

2. MicroJava 기초

교육명	**마이크로 자바 기초(Quarkus)**
교육개요	클라우드 환경에 최적화된 경량 자바 프레임워크인 Quarkus를 활용하여 RESTful API 개발, 데이터 연동, Kubernetes 배포 및 관측 기능을 실습하며 마이크로서비스 기반 개발에 필요한 기초 역량을 함양하는 과정
목표	- Quarkus 기반 REST API 개발 역량 습득 - GraalVM을 활용한 네이티브 빌드 실습 - Kubernetes 배포 및 관측 자동화 이해 - DevOps 연계 및 보안 구성 학습
기간	3일 또는 4일 과정 (옵션에 따라 선택 가능)
랩	오픈스택 기반, 인터넷 사용 가능한 노트북 요구
교육대상	- 클라우드 기반 자바 개발자 - Kubernetes 환경에서 자바 서비스 운영을 담당하는 인프라 엔지니어 - 경량화 및 빠른 배포가 필요한 자바 기반 시스템 개발자
1일	[기초 개념 및 개발 환경 구축] - Quarkus 개요 및 장점, JVM과 네이티브 모드 비교 - Quarkus Dev Mode, Dev UI, 테스트 환경 실습 - REST API 기본 구조 작성 (@GET, @POST 등) - application.properties, 프로파일 분기 실습 - Live Coding & 실시간 리로드 확인
2일	[데이터 처리 및 관측 기능 이해] - PostgreSQL과 연동한 Panache ORM 기반 JPA CRUD

	- Repository 패턴 및 트랜잭션 처리 실습 - Health Check(`smallrye-health`), Metrics(`micrometer`) 구성 - OpenTelemetry 기반 분산 추적 기본 적용 - Prometheus + Grafana 대시보드 실습
3일	[클라우드 통합 및 배포 실습] - Kubernetes 통합: config, health, readiness probe - ConfigMap, Secret, 환경변수 동적 주입 실습 - GraalVM 기반 네이티브 이미지 빌드 및 비교 - Deployment YAML 자동 생성 및 minikube 배포 실습 - 미니 프로젝트 구성 (도서 API 구조 설계 및 코드 작성 시작)
4일	[CI/CD 및 보안 연계 - 선택] - GitHub Actions 또는 Tekton으로 Quarkus 빌드 & 배포 자동화 - Keycloak 기반 OAuth2 보안 및 사용자 인증 실습 - `@Retry`, `@Timeout`, `@Fallback` 기반 회복력 패턴 적용 - 마이크로서비스 관측 구성: Trace + Metrics + Logs 통합 - 미니 프로젝트 완료 및 결과 공유

3. 파이썬 기초

교육명	**파이썬 언어 기초**
교육개요	인프라 운영 및 시스템 자동화를 위한 파이썬 언어 기초 교육입니다. 반복 작업을 효율적으로 처리하고 API 호출, 로그 분석, 시스템 제어를 파이썬으로 수행하는 기초 역량을 배웁니다.
목표	- 파이썬 문법과 구조 이해 - 시스템/네트워크 제어 자동화 스크립트 작성 - API 및 로그 분석 실습 수행
기간	4일 (1일 6시간 기준, 총 24시간)
랩	오픈스택 기반, 인터넷 사용 가능한 노트북 요구
교육대상	시스템 관리자, 인프라 운영자, DevOps 입문자, 자동화에 관심 있는 일반 IT 종사자
1일	파이썬 개요 및 설치 (VS Code, Jupyter 환경 구성) 변수, 자료형 (문자열, 리스트, 딕셔너리, 튜플, 셋) 조건문 (if, else, elif), 반복문 (for, while) 리스트 컴프리헨션 실습 실습: 사용자 입력을 받아서 정렬/필터하는 리스트 필터링기
2일	함수 선언, 반환값, 스코프 개념 모듈/패키지 import 방법 파일 입출력(open, with, read, write) 예외 처리 (try, except, finally) 실습: 로그 파일 분석기, 파일 백업 자동화 스크립트
3일	시스템 명령어 실행(os, subprocess 활용)

	CLI 인자 파싱(argparse) 디스크 사용량 점검, 네트워크 체크 자동화 실습: ping, traceroute 기반 네트워크 상태 수집기 실습: 파일 정리 스크립트 (확장자별 자동 분류)
4일	REST API 기본 (requests, json) Prometheus, Telegram 등 연동 실습 JSON 응답 처리 및 조건 필터링 실습: 시스템 모니터링 자동화 및 경고 전송 종합 미니랩 (4개 중 선택 실습) 1. Prometheus 기반 상태 점검기 2. 파일 자동 정리 도구 3. 네트워크 점검 도구 4. 텔레그램 봇으로 경고 전송

4. 메소드 개발(BDD/TDD)

교육명	**Go 언어 기반 BDD 블로그 개발과 CI/CD 실무** ***(프리코스 포함)***
교육개요	본 과정은 Go 언어로 간단한 블로그 백엔드 애플리케이션을 개발하며, BDD 테스트 기반과 PostgreSQL 연동, CI/CD 파이프라인 구축까지 실습 중심으로 학습하는 과정입니다. 시스템 엔지니어를 위해 사전 Go 언어 및 웹 개념을 익힐 수 있는 프리코스를 함께 제공합니다.
목표	- Go 언어와 Gin 프레임워크 기반 웹 API 개발 - Ginkgo 기반 BDD 테스트 작성 - PostgreSQL 연동 실습 - CI/CD 개념 이해 및 Tekton 또는 GitLab CI 실습
기간	4+α일 (프리코스 0.5~1일 분량 + 본과정 4일, 1일 6시간 기준)
랩	오픈스택 기반, 인터넷 사용 가능한 노트북 요구
교육대상	DevOps 실습을 위한 개발자 KNATIVE기반의 개발 과정에 관심 있는 시스템 엔지니어 CI/CD에 관심 있는 교육생
프리코스	Go 언어 기초 문법 실습 (변수, 조건문, 함수 등) REST API, HTTP 메서드 기본 개념 JSON 포맷 이해 및 실습 Git 기본 명령어와 GitHub 리포지토리 사용법
1일	- 개발 방법론 개요 (Agile, TDD, BDD) - VSCode로 개발환경 구성 - Hello World 예제 및 기본 함수 테스트 - testing 패키지로 테스트 기본 실습 - Ginkgo/Gomega 설치 및 기초 테스트 작성

2일	Gin 기반 블로그 구조 설계 글 작성/조회/삭제 기능 구현 PostgreSQL 모델링 및 연동 (GORM)
3일	BDD 테스트 시나리오 추가 테스트 자동화 구성 Dockerfile 작성 및 로컬 테스트
4일	CI/CD 개념 설명 Jenkins와 Tekton 비교 GitLab CI 또는 Tekton으로 자동화 파이프라인 구축 및 실행

세미나/부트캠프

1. OKD 부트캠프

교육명	**OKD 부트캠프**
교육개요	레드햇 오픈시프트 커뮤니티 버전 OKD기반으로, 오픈시프트 설치 및 구성에 대해서 확인한다. 설치 후, okd기반으로 서비스 배포 및 서비스 구성을 하면서, 쿠버네티스와 어떠한 차이점이 있는지 확인한다. okd는 커뮤니티 버전이기 때문에, 기술 지원은 받을 수 없지만, 다운 스트림 버전 OpenShift와 기능적으로 동일하기 때문에 Bad Test용도 혹은 중소규모로 운영 시, 적절한 제품이다.
목표	okd기반으로 오픈시프트 기능 확인 및 설치 클러스터 운영 및 CI/CD구성
기간	총 4일, 3일로 조정가능 과정
랩	오픈스택 기반, 인터넷 사용 가능한 노트북 요구
교육대상	리눅스 명령어 및 사용 경험자 쿠버네티스 명령어 및 사용 경험자
1일	**OKD 소개** - OKD와 OpenShift의 차이점 - 쿠버네티스와 OKD의 차이점 및 역할 - 표준 도구 **클러스터 설치** - OKD설치 방법(IPI/UPI방식) - 클러스터 아키텍처(컨트롤러 및 컴퓨트) - 설치 전 L4 및 DNS 준비 - OKD 설치(프로덕트 모델 기반) - 기본 명령어 및 서비스 살펴보기
2일	**자원 관리** - 기존 쿠버네티스 자원 명령어 확인 * kubectl, oc 명령어 * 기존 쿠버네티스 자원 생성 및 관리

	* okd project/kubernetes namespace **POD 및 Deployment그리고 DeploymentConfig의 차이점** - 애플리케이션 생성 - 애플리케이션 배포 및 관리 - 배포전략 **네트워크** - okd SDN 네트워크 * Kubernetes SDN, OVN - okd router * router기반으로 서비스 라우팅 구성 - okd ingress * okd에서 ingress 서비스 사용하기 * ingress와 router의 차이점 **okd 스토리지** - okd에서 제공하는 스토리지 * 스토리지 백엔드 추가 * PV/PVC * StroageClass **okd 사용자** - okd 사용자 관리 * 사용자 백엔드 구성 * 사용자 추가 - rbac기반으로 자원 접근 제한
3일	**빌드** - 이미지 빌드를 위한 S2I - 컨테이너 이미지 빌드 - 빌드 및 애플리케이션 배포 - 템플릿 기반으로 이미지 배포 **okd 오퍼레이터** - okd 오퍼레이터 설명 - 오퍼레이터 기반으로 클러스터 서비스 확장 **okd helm** - okd와 helm 활용하기 - helm chart 설치 및 관리 **okd package** - kustomized기반으로 패키징 - helm chart기반으로 애플리케이션 패키징 - 사용자 operator 패키징
4일	**모니터링 및 디버깅**

	- Prometheus, Grafana, EKF 기반으로 리소스 모니터링 - POD 및 노드 자원 상태 확인 - 노드 접근 및 디버깅 **CI/CD 구성** - okd기반에서 CI/CD 구성 - okd build 사용하기 - Tekton기반으로 CI/CD - ArgoCD기반으로 자원 배포 및 확인하기 - GIT 및 이미지 서버

2. CKA 부트캠프

교육명	CKA 부트캠프
교육개요	CKA 대비를 위해서 직접 랩 구축 후, 랩 기반으로 쿠버네티스의 기본 기능 기반으로 시험 문제에 대해서 대비한다.
목표	CKA 시험 환경에 편하게 사용 및 적용할 수 있도록 핸즈-온-랩 기반으로 태스크 단위의 작업을 수행한다.
기간	총 4일, 3일로 조정가능 과정
랩	오픈스택 기반, 인터넷 사용 가능한 노트북 요구
교육대상	리눅스 명령어 및 사용 경험자 쿠버네티스 명령어 및 사용 경험자
1일	**랩을 위한 간단한 환경 구축** - Xfce기반으로 X-ORG Session사용하기 - 랩 설명 및 구성 **기본적인 kubectl 사용 방법** - 컨텍스트 변경 및 네임스페이스 **명령어 사용법 및 자원 활용** - 기본적인 쿠버네티스 컴포넌트 확인 - 쿠버네티스 추상자원 **쿠버네티스 문서 훑어보기** - 공인 문서와 블로그 그리고 토론의 차이점 - docker/containerd/cri-docker 그리고 CRI-O **대비 1** - 클러스터 살펴보기 - 사용자 로그인 및 클러스터 확인하기 - 각 클러스터의 자원 살펴보기 **대비 2** - ETCD와 쿠버네티스 관계 - 쿠버네티스 스케줄러 * 스케줄러 장애 처리 - kubelet의 설명 * kubelet 역할 및 장애처리 - kube-proxy * kube-proxy의 역할 및 발생 가능한 장애 * kube-proxy와 네트워크, 그리고 커널 관계 - POD * Static POD와 General POD의 차이 * POD 개념 설명 및 구조 * POD를 YAML기반으로 생성 * YAML파일 사양

	- namespace * 쿠버네티스에서 namespace와 커널의 namespace * namespace 생성 및 관리
2일	**대비 3** - 쿠버네티스에서 제공하는 복제 서비스 * ReplicaSet/ReplicationController * Deployment * DaemonSet * StateFul * 랩 **대비 4** - 쿠버네티스 서비스 * 서비스 설명 * ClusterIP * Load balancer * 랩 **대비 5** - 쿠버네티스 스케줄링 * 레이블 및 셀렉터 * Taints/Toleration * 노드 셀렉터 및 선호도 * 랩 - 자원 제한 * 리미트 및 쿼터 차이 * 랩 - 정적 포드(Static POD) * Static POD 구성 및 관리 * 랩 **대비 6** - RBAC * 설명 * RBAC 기반으로 자원 관리 및 접근제한 * 랩 **대비 7** - 로깅 * 쿠버네티스 로깅 설명 * 모니터링 방법 및 기능 설명 * 애플리케이션 로그 확인 * 랩

	대비 8 - 애플리케이션 생명주기 * 애플리케이션 생명 주기 관리 방법 * 롤링 업데이트 * 애플리케이션 인자 값 설정 및 Containerfile * ConfigMap, Secret과 관계 그리고 Pod + Containerfile(Dockerfile)과 ENV과 그리고 설정 변수 * 보안이 필요한 정보(data)관리 * 랩
3일	**대비 9** - 멀티 컨테이너 * 설명 * 다중컨테이너 구성 및 사용 방법 * 초기화 컨테이너(init-container) * 애플리케이션 힐링 * 랩 **대비 10** - 운영체제 및 클러스터 관리 * 운영체제 업데이트 * 클러스터 업데이트 * ETCD 백업 * 랩 **대비 11** - 쿠버네티스에서 제공하는 보안 설명 * 쿠버네티스에서 TLS 사용하기 * 쿠버네티스 kubeconfig 사용하기 및 설정 * RBAC 및 서비스 계정 + 일반 RBAC + 클러스터 RBAC + 서비스 계정 * 컨테이너 이미지 보안 + docker-registry + registry token 그리고 인증 * Network Policy (Namespace Network Policy) * 랩
4일	**대비 12** - CRD 개념 및 구성 * custom resource는 무엇인가? * CRD컨트롤러

	* CRD자원 생성 및 관리 * 랩 **대비 13** - PV/PVC/StorageClass - PV/PVC 구성 및 드라이버 - PV/PVC를 구성하여 POD에 연결 - SC를 기반으로 구성 및 POD에 연결 - 랩 **대비 14** - Ingress 소개 - Ingress 서비스 구성 - 랩 **대비 15** - 쿠버네티스 패키지 소개 - Helm 설치 및 사용하기 - 랩 **대비 16** - 쿠버네티스 커스터마이즈 소개 - 커스터마이즈 구성 - 랩 **총 예상 문제**

3. 하이브리드 클라우드를 위한 전략 실습

교육명	하이브리드 클라우드를 위한 전략
교육개요	온프레미스와 퍼블릭 클라우드를 통합 운영하기 위한 전략적 접근 방법 및 기술 요소를 학습합니다. Kubernetes, Keycloak, MinIO, CI/CD 파이프라인을 중심으로 하이브리드 환경 구성 실습을 포함합니다.
목표	- 하이브리드 클라우드의 개념과 구성 요소 이해 - 인증 및 스토리지 통합 전략 습득 - 멀티 클라우드 환경과의 차이점 구분 - 실습을 통한 연동 구성 능력 배양
기간	2일
랩	오픈스택 기반 랩 및 퍼블릭 클라우드 계정(없어도 됨)
교육대상	인프라 관리자, DevOps 엔지니어, 클라우드 아키텍트, 하이브리드 인프라를 설계하고자 하는 실무자
1일	**하이브리드 클라우드 개념 및 기본 개념** - 하이브리드 클라우드의 정의와 개념 소개(이 부분에서 우리는 Container/VirtualMachine 하이브리드로 촛점) - 기업의 IT 인프라 변화에 대한 배경(퍼블릭 클라우드 및 컨터이너 기반의 인프라 시스템 영향) - 클라우드 서비스 모델(퍼블릭/프라이빗/하이브리드(멀티 클라우드와 동일한 뜻. 용어만 다름) - IaaS/PaaS/SaaA의 특징 및 차이점 - Container/VirtualMachine 기반으로 하이브리드 클라우드
2일	**클라우드 주요 도구 및 오케스트레이션 소개** - 표준 컨테이너 도구 소개 - 표준 가상머신 도구 소개 - API는 왜 중요한가? 시스템 엔지니어 입장에서 API - 하이브리드 클라우드 아키텍처 소개 및 충족조건 * 애플리케이션 및 데이터 포털 전략 - 온프레미스+퍼블릭클라우드=하이브리드? 멀티클라우드? - 만약, 하이브리드 클라우드 구현을 해야한다면...?

4. 테크톤 세미나

교육명	테크톤 세미나
교육개요	쿠버네티스 기반의 컨테이너 시스템에 CNCF의 표준 기술인 Tekton를 구축한다. 이 세미나에서는 Gogs기반으로 간단한 GitOps구축 후, 쿠버네티스 기반의 컨테이너 시스템에 CD 통해서 이미지 빌드 후 docker-registry 배포 및 kubernetes-action를 통해서 특정 네임스페이스까지 배포를 직접 구현 및 실행한다.
목표	쿠버네티스 기반으로 테크톤 서비스 구축 및 서비스를 배포한다.
기간	1일

코드	SEM-010
교육대상	모든 소프트웨어/시스템 엔지니어 및 기획자
1일	**테크톤 설명** 테크톤 설치 및 명령어 학습 - HelmChart/Operators - tkn/kubectl CLI **간단한 자원 설명** - tasks, taskruns - pipepline, pipelineruns **간단하게 CI/CD 구축** - tomcat - postgresql - maven **서비스 배포 및 확인** - buildah - kubernetes-action

5. Kube-virt/libvirtd 세미나

교육명	**Kube-virt Libvirt 세미나**
교육개요	쿠버네티스는 컨테이너에서 가상머신 지원 및 제공한다. 쿠버네티스 기반으로 가상머신 배포 시, 어떠한 방법으로 이미지를 생성 및 배포해야 하는지 학습한다. 또한, CI/CD를 통해서 가상머신 이미지를 컨테이너 이미지처럼 자동 패키지/빌드/디플로이가 가능한지 확인한다.
목표	쿠버네티스 가상머신 기반으로 가상머신 이미지 빌드 및 배포에 대해서 학습한다.
기간	1일
코드	SEM-010
교육대상	모든 소프트웨어/시스템 엔지니어 및 기획자
1일	Kube-virt 설명 Libvirtd와 가상머신과 관계 이미지 빌드 도구 및 빌드 방식 CI/CD와 가상머신 관계 가상머신 이미지 쿠버네티스 가상머신으로 배포

6. SELinux/AppArmor 실전 운영 가이드(제발 SELinux/AppArmor 끄지 마세요!)

교육명	**SELinux/AppArmor 실전 운영 가이드**
교육개요	SELinux/AppArmor는 리눅스 운영체제 입장에서 마지막 프로세스 제어 기능이다. 해당 기능을 사용하지 않고 운영하여도 문제가 없다. 다만, 허용되지 실행 혹은 사용자 접근이 발생하는 경우, 시스템에서는 어떠한 제어를 할 수 없다. 이 교육에서는 SELinux 개

	념 및 사용 방법. 그리고 컨테이너 및 오픈스택과 같은 가상머신을 운영 시, 어떠한 방식으로 정책을 구성해야 하는지 학습한다.
목표	SELinux/AppArmor를 어떠한 방식으로 운영 및 관리하는지 학습 사용자가 원하는 정책을 작성 및 추가 학습 다중 사용자 권한에 대한 학습
기간	2일
코드	SEM-010
교육대상	모든 소프트웨어/시스템 엔지니어 및 보안 기획자
1일	**SELinux 개념 및 구조** - MAC 개념 - SELinux 아키텍처(정책 서버/보안 컨텍스트/보안 레이블) **SELinux 기본 명령어** - getenforce, setenforce, setstatus - Z context - chcon, restorecon - semanage, setsebool **AppArmor와 SELinux 비교** - AppArmor 프로파일 경로 기반 제어 방식 이해 - aa-status, aa-complain, aa-enforce, aa-disable **AppArmor 기본 명령어** - apparmor_parser, aa-genprof, aa-logprof - 프로파일 추가/삭제/편집 방법 **사용자 정책 작성 및 적용** - SELinux .te, .pp파일 - AppArmor 프로파일 작성 방법
2일	**중앙 관리 및 알림 시스템** - setroubleshoot-server 개념 및 설치 - 실시간 AVC 경고 메일, rsyslog, journald 연동 **Libvirtd, container 보안적용** - libvir SELinux 정책 관리(virtd_t) - Podman, Docker에서 SELInux/AppArmor Label 설정 **사용자 정책 작성 및 BPF통한 공격 방어** - 사용자 정의 정책 작성 - BPF 해킹 도구를 이용해 우회 시도 후 차단 테스트 쿠버네티스에 SELinux/AppArmor 적용 - POD 및 애플리케이션 보안 - 볼륨 디스크 접근 보안 **최종 실습** - SELinux/AppArmor 종합 적용 - 모드 설정 변경/로그 분석/정책 수정