

교육가이드

교육 가이드

오픈소스 기업 강의

운영체제

운영체제 교육

현재 리눅스는 시스템을 운영하는 부분을 담당하지만, 서비스에 대한 라이브러리 역할 및 컨테이너/가상화에 대한 런타임을 제공하는 런타임 라이브러리 제공하는 시스템 소프트웨어이다.

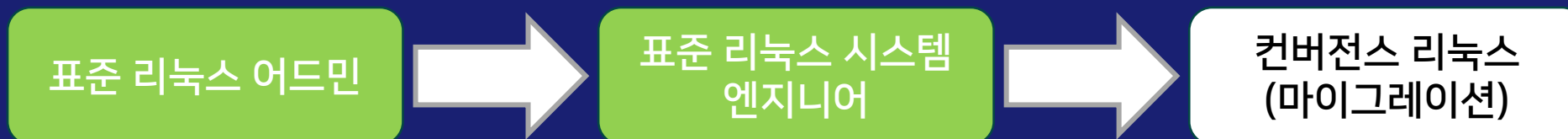
어떠한 리눅스 배포판을 사용하여도, 시스템 운영에 상관은 없지만 핵심 컴포넌트에 대한 이해 및 활용은 매우 중요하다. 현재 리눅스 시스템은 다음과 같은 기능을 제공하고 있다. 더 이상 **명령어 학습은 중요하지 않다**.

1. systemd
2. XFS/LVM2/BTRFS/Stratis/VDO
3. SELinux/AppArmor
4. 기본적인 운영 명령어
5. 컨테이너 및 가상화(podman, container-tools, libvirt, libguestfs)

운영체제 기본 교육

현재 리눅스는 systemd 기반으로 통합이 되고 있다. 많은 리눅스 시스템 엔지니어 그리고 벤더사 교육에서 이러한 부분을 많이 알려주지 않는다.

오픈소스 배포판을 통해서 systemd 및 통합 및 변경된 시스템 블록 영역에 대해서 확인 하도록 한다.



고급 리눅스 교육

고급 리눅스 교육은 보통 다음과 같은 부분에 대해서 강화를 한다.

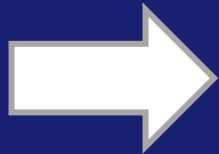
1. 시스템 트러블 슈팅
2. 시스템 보안(SELinux/AppArmor 등)
3. 리눅스 시스템 하드닝(NIST 기준 베이스라인 보안)
4. 클러스터링 시스템

현재는 컨테이너 및 가상화를 인하여 이러한 부분이 이전에 비해서 많이 연해지고 있지만, 여전히 고급 기술이며 시스템 운영 시, 꼭 필요한 기술이기도 한다.

고급 리눅스

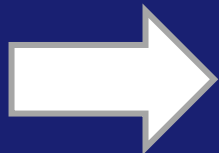
많은 리눅스 기술이 컨테이너 및 가상화로 넘어가면서 이전에 비해서 스크립트 및 트러블 슈팅이 중요도가 다시 많이 올라가기 시작 하였다. 특히, 클러스터링 시스템은 여전히 많은 인프라에서 사용하기 때문에 시스템 엔지니어라면 학습을 권장 한다.

셸 스크립트 파워툴즈



표준 리눅스 클러스터링 시스템

표준 리눅스 장애처리



리눅스 보안 하드닝

컨테이너

이전에 다양하게 제공이 되었던 컨테이너 솔루션이 지금은 CNCF로 통합이 되면서 오픈소스의 큰 단점인 파편화가 드디어 어느정도 해결이 되었다. 대다수 빅-테크 기업들은 현재 다음과 같은 기술을 사용하고 있다.

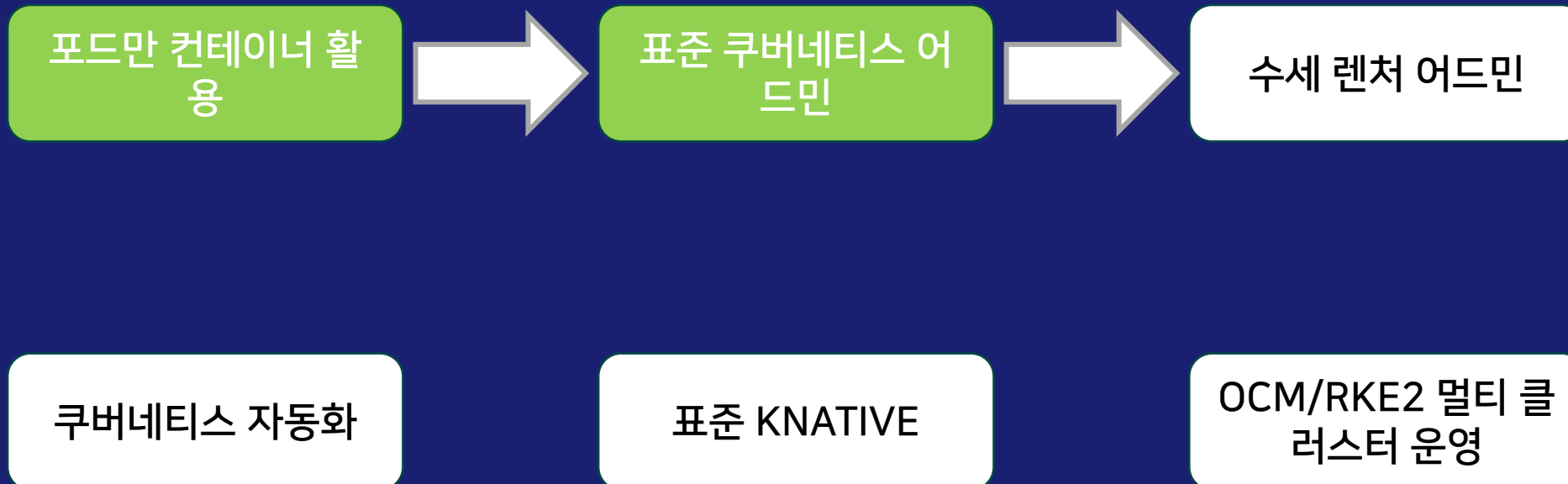
1. Docker/Docker-Swarm
2. Podman
3. Kubernetes

도커의 Docker-Swarm은 더 이상 개발을 하지 않으며, Docker는 기존 프로그램 유지 및 개발은 계속 하고 있으나, 현재 이 부분도 Dockerd에서 Containerd로 분리가 되면서 CRI Adaptor 기반으로 쿠버네티스를 지원하고 있다.

이러한 이유로 기존 Docker사용자는 Kubernetes와 높은 호환성 기반으로 개발/테스트/마이그레이션을 위해서 Podman기반으로 개발 환경 전환을 권장한다.

컨테이너

현재 컨테이너 시장은 포드만/쿠버네티스/테크톤 이외 CNCF도구로 통합이 되고 있다. 기존 Docker사용자는 여전히 기존 컨테이너 환경 사용이 가능하지만, 가급적이면 포드만 기반으로 전환을 권장하고 있다.



가상화

리눅스 가상화 운영을 위해서 다음과 같은 기술을 보통 권장한다.

1. 리눅스 시스템 블록 이해(어드민 영역)
2. 가상머신 아키텍처 및 메커니즘(QEMU, KVM 등)
3. Libvirtd와 libguestfs 도구
4. 컨테이너와 가상머신의 아키텍처 차이점
5. 쿠버네티스 가상화

가상화

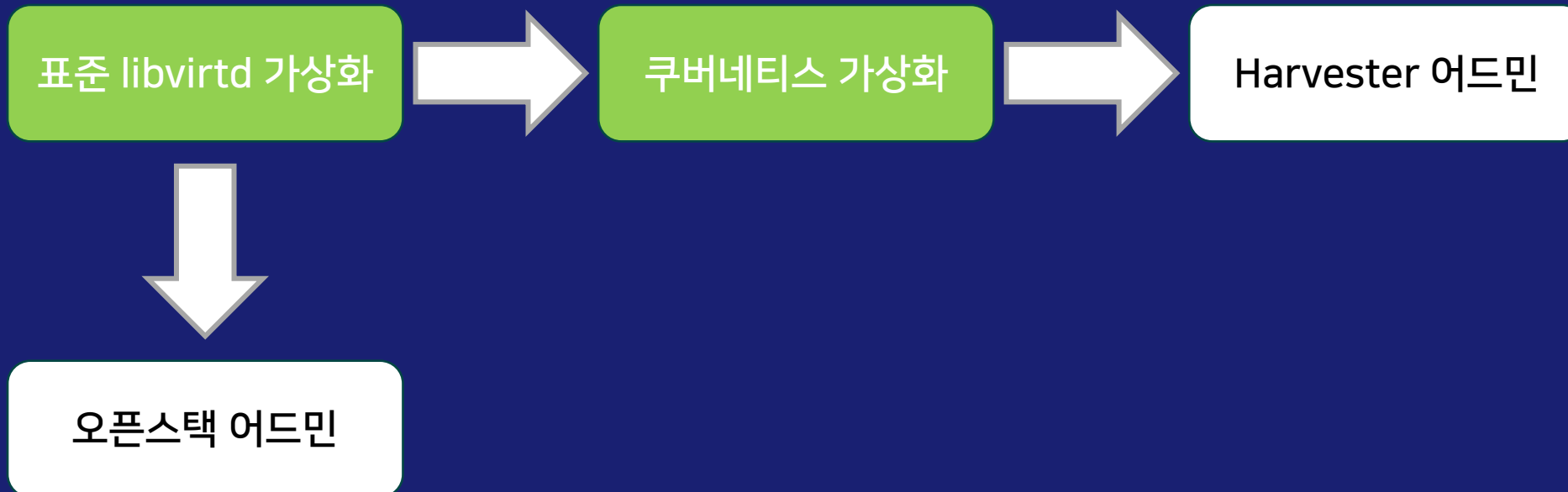
오픈소스 기반의 가상화 서비스는 실제로 위에서 언급한 기술 기반으로 동작 및 구성이 되기 때문에, 해당 영역에 대한 지식이 없는 경우, 트러블 슈팅이 어려운 부분이 있다.

현재 쿠버네티스는 가상화 기능도 통합을 하고 있기 때문에, 쿠버네티스 기반으로 운영 시 컨테이너+가상화를 동시에 운영이 가능하다.

가상화

현재 오픈소스 가상화는 크게 두 가지로 나누어지고 있다.

IaaS영역에서는 오픈스택, PaaS+ 영역에서는 쿠버네티스 가상화(OpenShift Virtualization, SuSE Harvester)와 같은 솔루션이 시장에 확대가 되고 있다. 다만, 이 두 개 기술을 활용하기 위해서 Libvirt 관련 기술에 대해서 학습 및 교육을 권장한다.



자동화

모든 리눅스 시스템의 제일 큰 우선순위는 현재 자동화 이다. 인프라를 자동화 혹은 AI 기반 자동화를 위해서 IaaC 도구가 많이 필요하다. 다음과 같은 도구가 오픈소스에서 많이 사용하는 인프라 자동화 도구.

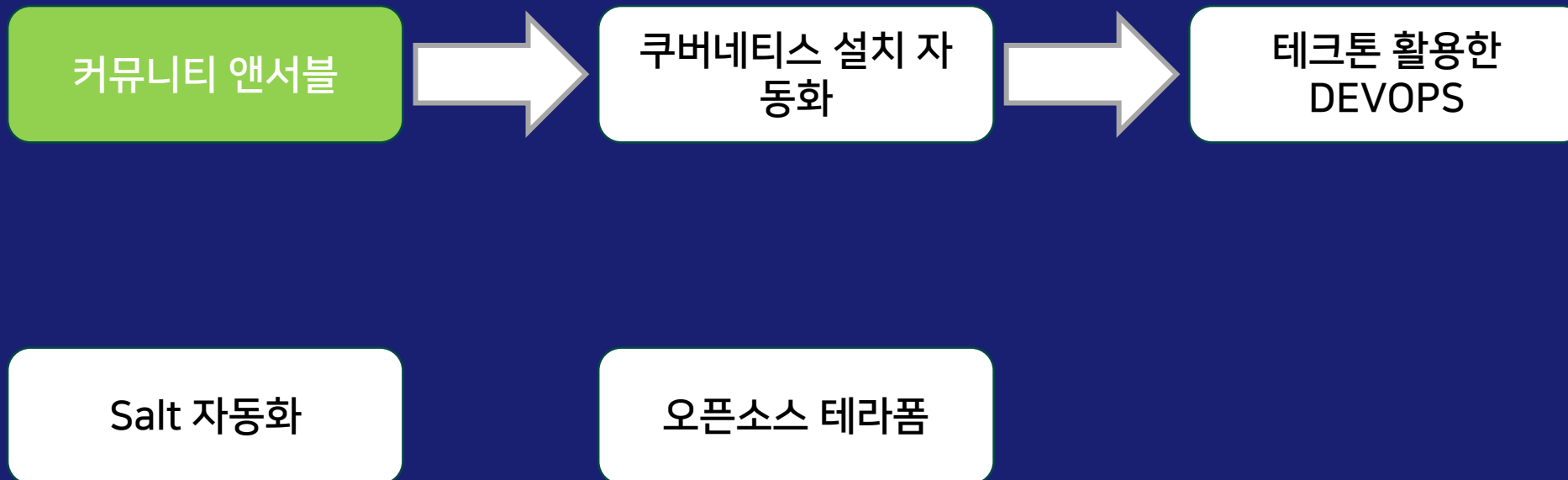
1. Ansible
2. Terraform(Open Tofu)
3. Salt
4. Crossplane

현재 Ansible이 많은 사용자를 가지고 있으며, 인프라 측면에서 많이 사용한다. 이와 비슷하게 Salt도 동일하게 기능을 제공하지만, VMWare로 인수가 된 후, 커뮤니티 활동이 많이 줄어든 상태이다.

Terraform 경우에는 IBM으로 인수 직전, Open Tofu라는 이름으로 Fork하였고, 기존 Terraform사용자들이 Open Tofu로 마이그레이션을 많이 고려하고 있다.

자동화

현재 모든 리눅스 시스템은 자동화 가지고 가고 있다. 인공지능 시스템이 들어 오면서 점점 리눅스 시스템 엔지니어에게 더 높은 수준을 요구하고 있다. 현재는 대다수 리눅스 배포판은 자체 패키지로 앤서블을 지원하고 있으며, 사용이 어려운 경우 파이썬 PIP를 통해서 지원이 가능하다.



스토리지

오픈소스에서 제공하는 스토리지는 Software Defined Storage 개념으로 제공하고 있다. 현재 많은 개수의 SDS 시스템이 있으며, 보통 엔터프라이즈 수준에서 다음과 같은 도구를 많이 사용한다.

1. Ceph
2. Ganesha(NFS)
3. GlusterFS

Ceph 오픈소스의 대표적인 기업용 스토리지 시스템이다. 현재는 IBM이 레드햇 인수 후, IBM에서 Ceph 스토리지 판매를 담당하고 있다. 하지만, 여전히 컨테이너 및 가상화 고객들은 Rook과 같은 오픈소스 솔루션 기반으로 시스템을 운영을 하고 있다.

스토리지

스토리지 교육은 기업 규모로 넘어가면 CEPH/NFS/GlusterFS를 많이 사용한다. 대다수 기업들은 일반적으로 NFS를 많이 사용하지만, 현재 사용하는 NFS는 레거시로 전환이 되고 있으며, 현재는 Ganesha NFS 기반으로 pNFS 구현을 권장한다.



언어

리눅스 시스템 엔지니어는 더 이상 **코드 이해도 없이 운영 및 구성**은 어려운 상황이 되었다. 대다수 시스템이 자동화 도구를 기반으로 대규모 시스템을 운영하고 있으며, 대규모가 아니라고 하여도 어플라이언스 시스템에서 자동화 도구는 많이 활용한다. 대표적으로 Salt가 그렇다.

해외에서는 리눅스 시스템 엔지니어가 코드를 읽지 못하는 경우 보통 Runbook Engineer 혹은 Cargo Culture Engineer라고 부르고 있다.

이러한 이유로 리눅스 시스템 엔지니어는 최소 한 두 가지 언어 학습을 권장한다.

1. Python 언어
2. Go 언어

언어

현재 국내 리눅스 시스템 엔지니어는 Runbook Engineer 형태를 많이 가지고 있다. 다른 말로, 장애가 발생하였을 때 로그나 혹은 소스코드 분석이 되지 않으며, 심지어 시스템 로그도 분석을 못하는 경우가 늘어나고 있다.

이러한 부분을 해결하기 위해서 언어 교육을 통해서 좀 더 Read-Intensive task를 할 수 있도록 한다.

GO 언어

파이썬

메소드 개발