K8s

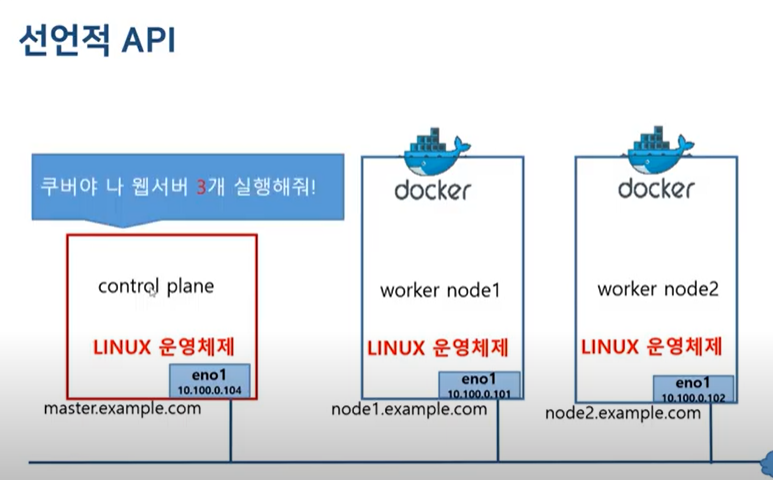
-> Linux Container 작업을 자동화 및 관리하는 오픈소스 플랫폼

-> 어디서나 실행가능 (On-premise, Public Cloud(AKS, EKS, GKE))

-> Container화된 워크로드와 서비스를 관리하기 위한 이식성이 있고, 확장가능한 오픈소스 플랫폼

-> k8s는 선언적 구성과 자동화를 모두 용이하게 해준다. K8s는 크고, 빠르게 성장하는 생태계를 가지고 있다. K8s 서비스, 기술 지원 및 도구는 어디서나 쉽게 이용할 수 있다.

선언적 API



Container 란

* 컨테이너는 VM과 유사하지만 격리 속성을 완화하여 애플리케이션 간에 운영체제(OS)를 공유한다. 그러므로 컨테이너는 가볍다고 여겨진다. VM과 마찬가지로 컨테이너에는 자체 파일 시스템, CPU 점유율, 메모리, 프로세스 공간 등이 있다. 기본 인프라와의 종속성을 끊었기 때문에, 클라우드나 OS 배포본에 모두 이식할 수 있다.

Ex) code

# cat > app.js

Const http = require(‘http’);

Const os = require(‘os’);

Console.log(“Test server starting…”);

var handler = function(req, res) {

res.writeHead(200);

res.end(“Container Hostnae: “ + os.hostname() + “\n”);

};

Var www = http.createServer(handler);

[www.listen(8080)](http://www.listen(8080));

# cat > Dockerfile

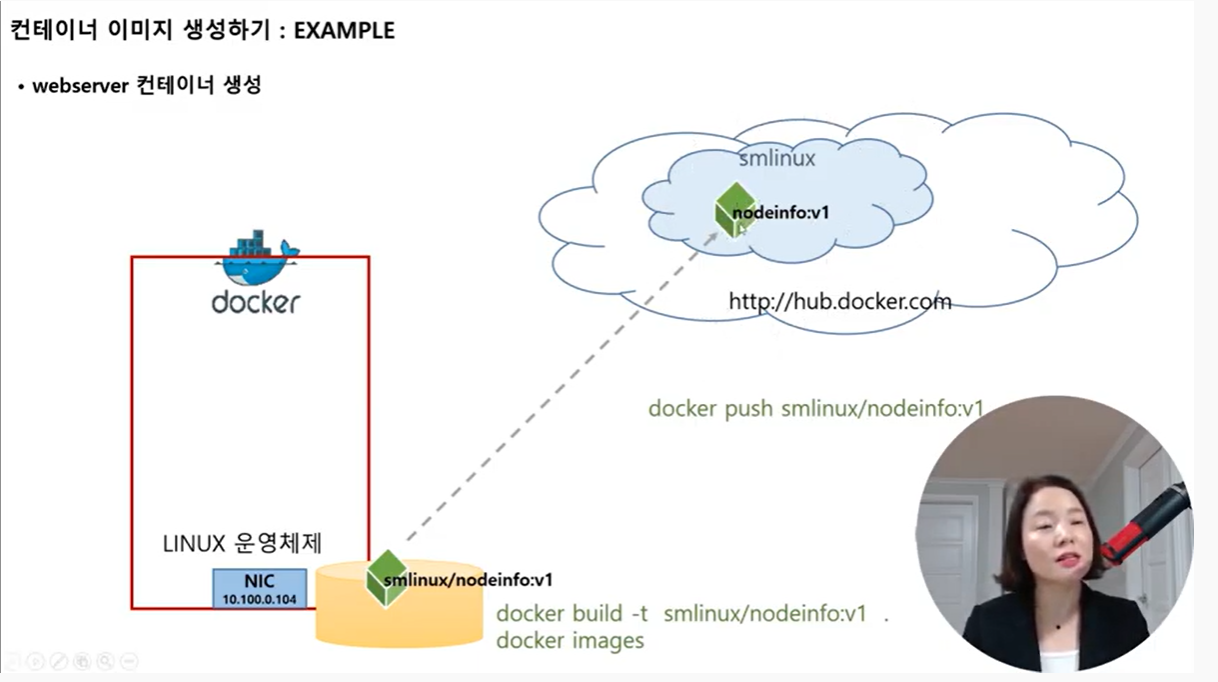
FROM node:12

COPY app.js /app.js

ENTRYPOINT [“node”, “appjs”]

<Ctrk><d>

Webserver Container 생성



컨테이너 이름

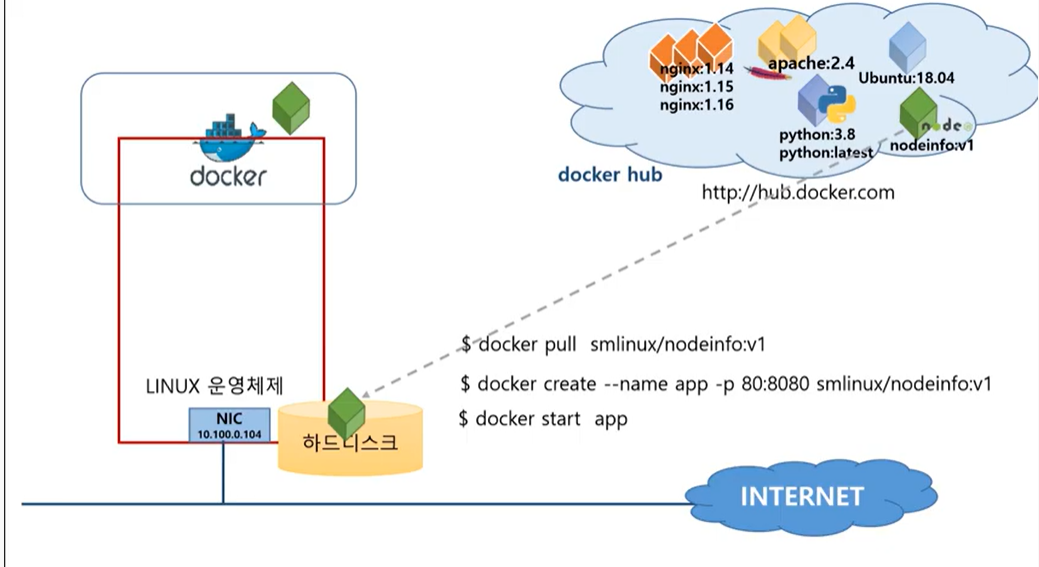
컨테이너 이름

1. 리눅스 시스템에 docker 설치
2. docker daemon을 실행
3. docker platfrom 생성
4. Container build 및 start

Build 명령어를 통해 컨테이너 생성

Push 명령어를 통해 repository에 업로드

\*\* 허브에는 저장한 수많은 컨테이너들이 존재



80포트로 들어오면 8080포트로 포워딩

Pull 명령어를 통해 컨테이너 다운로드

Create 및 start 명령어를 통해 앱을 구동

Virtual machine vs Container

Virtual machine

|  |  |  |
| --- | --- | --- |
| Virtual machine  (Guest Operating System) | Virtual machine  (Guest Operating System) | Virtual machine  (Guest Operating System) |
| Hypervisor  (Vmware, VirtualBox) | | |
| Infrastructure(Local Host) | | |

각각 목적에 맞게 용량을 설정하여 Virtual machine을 운영할 수 있다.

* 실제 Infra에 있는 하드웨어를 효율적이고 유연하게 운영 가능
* 만약 시스템에 장애가 발생하면 정상적으로 운영이 불가능
* 따라서, Scale Out(수평적으로 확장)을 통해 문제를 해결
* Scale Out으로 구성하면 하나의 시스템에서 장애가 발생하여도 나머지 시스템들로 원활하게 비즈니스 연속성이 가능

\*\* 비즈니스 연속성(BCP)

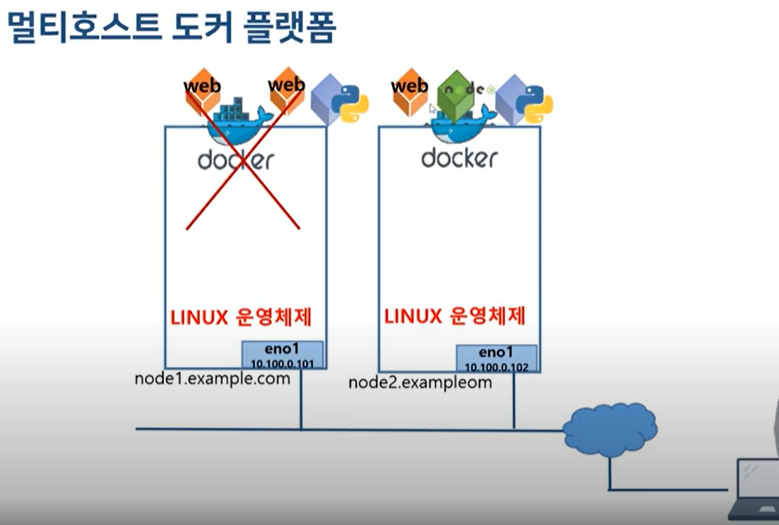
* 조직이 심각한 운영 중단 사태를 예방하고, 해당 사태가 발생했을 때 신속하게 회복할 수 있는 능력을 확보하기 위해 마련하는 상세한 전략 및 일련의 시스템

Container

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| App A | App B | App C | App D | App E |
| docker | | | | |
| Host operation  (Windows, Linux) | | | | |
| Infrastructure | | | | |

* Virtual Machin에 비해 앱 용량이 가벼움
* 확장 및 축소가 용이함
* 주 목적은 배포(deploy)

Multi Host Docker Platform



* 만약 docker에 장애가 발생할 경우 앱 구동이 불가능하다. 따라서 여러 대의 docker에 앱을 구동 시켜야 한다.
* 하지만, Engineer가 모든 컨테이너를 모니터링하면서 관리하는 것은 많이 어려움

Container Orchestration

\*\* 자동화된 컨테이너 배포 및 Scaling management

* Master Node가 Woker Node의 App을 적절하게 배치하여 운영

Container Layer

|  |  |  |
| --- | --- | --- |
| Layer 6 | Development Workflow opinionated Containers | OpenShift, Cloud Foundary, Docker Cloud, Deis, Flynn….. |
| Layer 5 | Orchestration/Scheduling  Service Model | Kubernetes, Docker Swarm, Marathon/Mesos, Nomad, Diego |
| Layer 4 | Container Engine | Docker, Rocket RunC(OCI), Osv, LXC, LXD |
| Layer 3 | Operation System | Ubuntu, RHEL, CoreOS, Unikernels |
| Layer 2 | Virtual Infrastructure | vSphere, EC2, GCP, Azure, OpenStack |
| Layer 1 | Physical Infrastructure | Raw Computer, Network, Storage |

설치 없이 k8s 실습

* <https://www.katacoda.com/courses/kubernetes/playground>

(katacoda – Kubernetes playground)

* <https://labs.play-with-k8s.com/p/c84uc841r0f00095cjc0#c84uc841_c84ucck1r0f00095cjcg>

(Docker Playground)

1. Initializes cluster master node:

kubeadm init --apiserver-advertise-address $(hostname -i) --pod-network-cidr 10.5.0.0/16

2. Initialize cluster networking:

Kubectl apply -f https://raw.githubusercontent.com/cloudnativelabs/kube-router/master/daemonset/kubeadm-kuberouter.yaml

K8s Cluster 직접 구성

1. Kubeadm
   * 쿠버네티스에서 공식 제공하는 클러스터 생성/관리 도구
2. Kubespray
   * 쿠버네티스 클러스터를 배포하는 오픈소스 프로젝트
   * 다양한 형식으로 쿠버네티스 클러스터 구성가능
   * 온프레미스에서 상용 서비스 클러스터 운영 시 유용

CNI(Container Network Interface)

* + Container간 통신을 지원하는 VxLAN. Pod Network라고도 부름
  + 다양한 종류의 플러그 인이 존재
  + flannel, calico, weavenet 중 한 개가 반드시 설치되어 있어야 Pod간 통신이 가능함

k8s Cluster 구성

* Control plane(Master node)
  + 워커 노드들의 상태를 관리하고 제어
  + Single master
  + Multi master(3, 5개의 master node)
* Worker node
  + Docker platform을 통해 컨테이너를 동작하며 실제 서비스 제공

Kubeadm을 이용한 쿠버네티스 설치 – On premise (vm3 master, node1, node2)

1. Docker install
   * All system : docs.docker.com

* sudo yum install -y yum-utils
* sudo yum-config-manager \
* --add-repo \
* https://download.docker.com/linux/centos/docker-ce.repo

1. Kubernetes install
   * 설치 전 환경설정
   * Kubeadm, kubectl, kubelet 설치
   * Control-plane 구성
   * Worker node 구성
   * 설치 확인