쿠버네티스의 이해

1. 쿠버네티스 : 오픈소스 시스템, 컨테이너화 된 앱을 자동으로 배포와 스케일링 및 관리해주는.
   1. 특징 : 서비스 디스커버리(검색/맵핑), 로드 밸런싱, 스토리지 오케스트레이션, 자동화된 롤아웃/백, 자동 복구, 자동 빈 패킹, 구성관리.
2. 클러스터 : 일련의 노드 머신의 집합, 컨테이너화 된 앱을 실행하기 위한.
   1. 구성요소들 : 마스터 노드 하나와 다수의 워커 노드 그리고 kubectl과 cloud
3. 마스터 노드 : 가상 머신, 클러스터 전체를 컨트롤(스케일링, 변경 반영 등)하며 내부에 있는 모든 노드를 관리하는.
   1. 구성요소, kube-api-server : 쿠버네티스의 모든 통신 맡음.
   2. 구성요소, kube-controller-manager : 파드들을 관리하는 각각의 컨트롤러를 제어.
   3. 구성요소, kube-scheduler : 리소스를 자원 할당 가능한 노드에 할당. 실행순서 제어.
   4. 구성요소, etcd : 클러스터 내의 모든 세부적인 데이터 저장소. 키-값 타입.
   5. 구성요소, cloud-controll-manager : 컨트롤러들을 클라우드 서비스와 연결하여 관리.
4. 워커 노드 : 가상 머신, 마스터에 의해 명령을 받고 실제 컨테이너들이 생성되고 일을 하는.
   1. 구성요소, kubelet : 컨테이너 실행. 지속적인 상태 체크. Kube-api-server와 통신하는 에이전트. 하나의 워커 노드상에서 동작하는 파드와 컨테이너 관리
   2. 구성요소, kube-proxy : 클러스터 내부의 가상 네트워크 설정/관리.
   3. 구성요소, 컨테이너 런타임 : 컨테이너 운영/실행 담당. Containerd, 도커 등등.
5. Container runtime interface : 쿠버네티스에서 다양한 컨테이너 런타임을 사용할 수 있게 해주는 API.
   1. 과정 : 유저 🡪 Kubernetes API 🡪 master node 🡪 worker node 🡪 CRI-O 🡪 runc 🡪 container host’s kernel 🡪 container.
6. 쿠버네티스 배포 도구 : kubeadm, kops, kubespray.

쿠버네티스의 리소스 종류

1. 파드 실행 컨트롤러 : 디플로이먼트, 스테이트풀셋, 잡, 크론잡, 데몬셋.
2. 서비스 관련 : 인그레스, 서비스.
3. 스토리지 : PV, PVC.
4. 설정 정보 : ConfigMap, Secret.

쿠버네티스의 리소스

1. 네임 스페이스 : 물리적인 마스터/워커 노드를 논리적인 단위로 분리한 리소스. 각 사용자는 권한이 부여된 네임 스페이스만 접근 가능.
2. 디플로이먼트 : 전체 MSA를 관리하는 세트. 레플리카셋을 관리하면서 실행시켜야 할 파드의 배포 및 지속적 관리를 하는 리소스. Stateless한 앱에 적용. 하나 이상의 파드를 관리. 배포시 롤링 업데이트. 서비스를 통한 외부 노출. 모든 파드가 1개의 pvc에 모두 연결.
   1. 레플리카셋 : 파드의 개수를 유지하고 관리하는 리소스. 실행되는 파드 안정적 유지.
   2. 롤링 업데이트 : 각기 다른 노드에 퍼져 있는 파드들을 부분부분 점진적으로 업데이트(중단 없이).
3. 레이블/셀렉터
4. 파드 : 실제로 컨테이너가 실행되는 공간. 컨테이너를 돌리는 최소한의 단위. 파드 마다 ip가 따로 있음. 캡슐화(컨테이너+스토리지+IP+실행방식 옵션+환경 등).
5. 서비스 : 클러스터 외부에서 클러스터 내부 파드에 지속적으로 접근할 수 있는 ip(방법)를 제공하는 리소스. 파드 간 로드밸런싱(요청 분배)과 서비스 디스커버리(검색 매핑) 함.
   1. 서비스 디스커버리 : 분산 네트워크 환경에서 서비스(실행중인 인스턴스)들을 자동으로 검색하고 서로 연결할 수 있도록 하는 프로세스(매핑 게이트웨이). 대규모 MSA에서 중요한 역할. 쿠버네티스와 같은 오케스트레이션 시스템에서 핵심 기능 담당.
6. 인그레스 : 클러스터 외부에서 클러스터 내부로 들어오는 http(s) 트래픽 관리(로드밸런싱). 도메인 이름을 통해 특정 웹 앱 서비스로 트래픽 라우팅.
7. 볼륨 : 파드의 일부분으로 동일한 파드 내의 컨테이너끼리의 공유. 컨테이너의 데이터를 보관하여 유지. 파드가 삭제되거나 재생성될 때도 데이터 유지. 리소스 제공자. 관리자에 의해 생성되거나 동적 프로비저닝을 통해 자동 생성됨. pvc로부터 요청이 들어오면 할당됨.
8. 볼륨 클레임 : 사용자가 pv에 스토리지를 요청하는 방법. 필요한 볼륨(스토리지)를 요청. 소비자. 클러스터 관리자가 프로비저닝한 PV 중에서 PVC의 요구사항을 충족하는 것이 자동으로 바인딩.
9. 스토리지 클래스 : pv로 확보한 스토리지 종류를 정의하는 리소스. Pv의 동적 프로비저닝일 때 사용됨. 미리 정의되어 있으면 그에 맞는 pv를 바로 생성 가능.
   1. 정적 프로비저닝 볼륨 구조 : 파드 -볼륨 사용-> PVC <-바인딩-> 이미 생성된 PV -> 스토리지.
   2. 동적 프로비저닝 볼륨 구조 : 파드 -볼륨 사용-> PVC <-바인딩-> PV <-프로비저닝 생성-> SC -> 스토리지.
10. 스테이트풀셋 : 재스케줄링 되어도 상태 유지해야하는 앱에 사용. 하나 이상의 파드를 관리. 식별자가 변하지 않음. 파드들의 순서 및 고유성 보장. 헤드리스 서비스를 통한 외부 노출. 파드마다 각각의 고유한 pvc를 생성하여 고유한 pv를 가짐.
11. 잡 : 일회성 작업(데이터 처리, 배치 계산, 백업 작업 등)을 관리하기 위해 사용되는 리소스. 성공적으로 작업 완료될 때까지 파드의 실행을 계속 관리.
12. 크론잡 : 주기적이고 반복적인 작업을 자동으로 실행하기 위한 리소스.
13. 데몬셋 : 클러스터 내의 모든 노드에 파드를 자동으로 배포하는데 사용되는 리소스. 각 노드에 대해 정확히 하나의 파드 복사본(로그 수집, 모니터링 등등)이 실행되도록 보장하는 것이 주 목적.
14. ConfigMap : 앱 코드(컨테이너)와 설정 데이터를 분리할 수 있게 해줌. 앱 설정을 유연하게 관리하고 환경 간 이동 쉬움. 키-값 형태의 설정 정보 저장소.
15. 시크릿 : 민감한 데이터 정보를 소스코드나 앱 설정에서 분리하여 안전하게 저장/관리. 모든 파드에는 자동으로 연결된 시크릿 볼륨이 존재. 메모리에 저장됨.

Infrastructure as Code : 스크립트를 통해 인프라 및 구성 관리를 자동화하는 방법론. 수동 프로세스가 아닌 코드를 통해 인프라를 관리하고 프로비저닝. 정보 관리를 위해 별도의 인프라 세트가 필요하지 않음.

1. 특징 : 비용 절감. 배포 속도 향상. 일관성 향상. 물리환경 세부정보 추상화. 오류 감소. 코드에 집중 가능. 구성 드리프트 제거.
2. 도구, 테라폼 : 인프라를 손쉽게 구축하고 안전하게 변경하고 효율적으로 관리할 수 있는 오픈 소스 도구. 플랫폼에 구애 받지 않음. 가변적인 인프라(인프라 자체 변경X, 재 프로비저닝).
   1. 특징 : IaC, 인프라를 코드로 정의하여 생산성, 투명성, 공유성, 효율성 확보.   
      변경 계획과 변경 적용을 분리하여 실수 줄음.   
      종속성 그래프를 작성하여 미리 변경을 적용했을 때의 인프라 상태 알 수 있음.   
      여러 장소에서 같은 구성의 인프라 구축/변경 자동화 가능하여 시간/실수 절약.
   2. 오케스트레이션 도구(복잡한 자동화와 조정 작업 처리. 서비스 전체에 걸친 작업 수행/여러 컴포넌트와 서비스 통합.). 선언적(원하는 상태 설명).
   3. 집을 설계하고 짓는 건축가. AWS, GCP, Azure에서 서버, 네트워크 등 기반 구조 세팅.
   4. 우수한 스케줄링 기능. 쾌적한 환경 제공. 도커와 통합 우수. 과정 불투명함.
3. 도구, Ansible : 여러 수동 IT 프로세스(구성 관리, 앱 배포, 프로비저닝, 오케스트레이션 등)을 자동화하는 오픈소스 IT 자동화 도구. 불변적인 인프라
   1. 컨피규레이션 관리 도구(서버나 다른 IT 시스템 설정과 관리 자동화. 단일 서버나 시스템 수준에서 더 세밀한 설정 제공.). 절차적(어떻게 도달할지 설명).
   2. 집 안을 꾸미고 요리를 하는 준비하는 셰프. 클라우드에 앱 설치 및 환경설정 수행.
   3. 안전한 보안과 ACL 기능. 자동화 프레임워크와 조정 가능. 종속성/상호/오케스트레이션과 연결된 특정 응용 프로그램에는 부적합.
4. 도구, vault : 크로스 플랫폼 패스워드 및 인증 관리 시스템. 다양한 인터페이스, 방법론 제공. 토큰과 시크릿 경로 🡨🡪 시크릿(데이터)
5. 도구, ceph : 오픈소스 소프트웨어 스토리지 플랫폼.
   1. 소프트웨어 정의 스토리지 이점 : 비용 대비 성능 절충. 유연한 인터페이스. 다양한 스토리지 구현.
   2. 구조 : DSD 데몬. 모니터. 매니저. 메타데이터 서버.