

1. Prometheus

1-1. 개요: 모니터링 및 알람을 보내주는 오픈소스 시스템이다. 2012년도부터 개발이 시작되었고 지금은 독립 오픈소스 프로젝트이며 각 회사에서 독립적으로 운영되고 있다. Prometheus는 2016년에 Kubernetes 참가 이후에 CNCF에 참가하였다.

1-2. 작성 언어: Go

1-3. 기능:

1-3-1. 이름/키 또는 이름/값 쌍 메트릭으로 식별되는 시계열 데이터가 있는 다차원 데이터 모델이다.

1-3-2. Prometheus용 SQL인 PromQL을 사용할 수 있다.

1-3-3. 분산 스토리지가 아닌 싱글 서버 노드를 사용한다.

1-3-4. 시계열 데이터는 HTTP를 통해 각 서버에서 Pull 방식으로 수집한다.

1-3-5. 중간 게이트웨이에 의한 시계열 데이터 Pushing을 지원한다.

1-3-6. service discovery 또는 static configuration에 의해 모니터링할 대상을 찾는다.

1-3-7. 그래프와 대시보드를 지원한다.



2. Grafana

2-1. 개요: Grafana는 query, 시각화, 알림을 사용할 수 있게 해주며 메트릭이 어디에 저장되어 있더라도 해당 메트릭을 분석할 수 있다. 대시보드를 생성, 확장, 공유하며 데이터 기반 문화를 가속시킨다.

2-2. 작성언어: Go

2-3. 기능:

2-3-1. Grafana는 데이터를 이해하는데 도움이 되는 다양한 시각화 옵션을 제공한다.

2-3-2. 시각적으로 임계치를 지정할 수 있으며, Slack, PagerDuty 등 다양한 도구들을 통해 알림을 받을 수 있다.

2-3-3. Grafana는 수십개의 데이터베이스를 지원하며 그것들을 혼합하여 같은 대시보드 상에 시각화 할 수 있다.



3. Jenkins

3-1. 개요: 소프트웨어 빌드, 테스트, 배포 등 모든 종류의 작업을 자동화하는 자체 컨테이너화된 오픈 소스 자동화 서버이다. Jenkins는 네이티브 시스템 패키지로 설치할 수 있으며 docker나 어떠한 독립실행형 서버에서도 설치가 가능하다.

3-2. 작성언어: Java

3-3. 기능:

3-3-1. 확장가능한 자동화 서버로 CI 서버나 CD 허브로 어떠한 프로젝트에도 사용될 수 있다.

3-3-2. 자바 기반 자체 컨테이너화되어 Windows, Linux, macOS, Unix 계열 OS에서 패키지를 통해 쉽게 사용할 수 있다.

3-3-3. 웹 인터페이스를 이용하여 쉽게 설치 및 구성이 가능하다.

3-3-4. Update Center에서 수백개의 플러그인들을 통해 지속적으로 CI/CD toolchain의 모든 도구와 통합한다.

3-3-5. 다양한 머신에 배포할 수 있고 빌드, 테스트, 배포 등을 돕는다.



Jenkins

4. Terraform

4-1. 개요: Terraform은 수백개의 클라우드 서비스를 관리하는 지속적인 CLI 워크플로우를 제공하는 IaC(Infrastructure as Code) 소프트웨어이다. Terraform은 클라우드 API를 선언적 구성 파일로 코드화 한다.

4-2. 작성언어: Go

4-3. 기능

4-3-1. 새로운 리소스를 생성하고 기존의 리소스를 관리하며 필요하지 않은 리소스를 제거하여 전반적인 생명주기를 관리하는 IaC를 정의한다.

4-3-2. terraform init를 이용하여 레지스트리로부터 자동적으로 커뮤니티나 파트너 모듈을 다운로드하고 설치한다.

4-3-3. 운영자가 인프라에 변화에 대해 안전하고 예측 가능할 수 있게 해준다.

4-3-4. 그래프를 이용하여 계획, 상태 등을 쉽게 생성할 수 있다.

4-3-5. 메타데이터의 히스토리를 유지하며 대규모의 인프라의 성능을 개선시킨다.



5. Istio

5-1. 개요: Istio는 마이크로서비스 간 데이터 공유를 제어하는 오픈소스 서비스 메쉬 플랫폼이다. Istio는 배포의 복잡성을 줄이고 개발 팀의 부담을 덜게 해주며, 모든 로깅 플랫폼이나 텔레메트리 또는 정책 시스템을 통합할 수 있는 API를 포함한 플랫폼이기도 하다. Istio의 다양한 기능 세트를 사용하면 분산 마이크로 서비스 아키텍처를 성공적이고 효율적으로 실행할 수 있으며 마이크로 서비스를 보호, 연결 및 모니터링하는 일관된 방법을 제공합니다.

5-2. 작성언어: Go

5-3. 기능:

5-3-1. 트래픽 관리(Traffic management): Istio는 룰 설정과 트래픽 라우팅을 통해 트래픽 흐름과 API 호출을 제어할 수 있다. 문제가 발생하기 전에 트래픽과 실패 복구 기능에 대해 더 나은 가시성과 함께 이슈를 확인할 수 있다.

5-3-2. 보안(Security): Istio는 기본 통신 채널을 제공하고 스케일에 따른 인증, 권한 부여 및 서비스 통신 암호화를 관리한다. Istio를 활용하면 애플리케이션 변경을 최소화하면서 여러 프로토콜 및 런타임 전반에서 정책을 일관되게 실행할 수 있다. Istio와 쿠버네티스(또는 인프라) 네트워크 정책을 함께 사용하면 네트워크 및 애플리케이션 레이어에서 파드 간 또는 서비스 간 커뮤니케이션을 보안할 수 있습니다.

5-3-3. 관찰 가능성(Observability): Istio의 강력한 추적 기능, 모니터링, 로깅 기능은 서비스 메시 배포에 대해 깊은 통찰력을 제공한다. Istio의 모니터링 기능을 통해 서비스 성능이 업스트림 및 다운스트림에 어떤 영향을 미치는지 실제로 이해하고 사용자 지정 대시 보드는 모든 서비스의 성능에 대한 가시성을 제공하고 해당 성능이 다른 프로세스에 어떤 영향을 미치는지 확인할 수 있다.

