
STIGMA

OSS Project 9TH / Team STIGMA
Lee, Suhyun (ilyf28@gmail.com)

-
- 1 Overview
 - 2 About STIGMA
 - 3 Pros and Cons
 - 4 Demo
 - 5 Next Step

Overview

1 필요에 의해

- 기존 Nagios 모니터링 시스템에 대해 사용상 불편함을 호소
- Agentless에 대한 needs 발생

Nagios 모니터링 시스템에 대한 요구사항

	As is	To be
Config	.cfg 파일 수정시 오타가 있는 경우 nagios 실행 안됨	Web UI를 통해 설정 변경 방식 개선
History	현재 상태 값만 제공	이전 상태 값 제공
Graph	검색 조건에 의해 검색된 시간대의 상태 값만 제공	실시간으로 변경되는 상태 값 제공

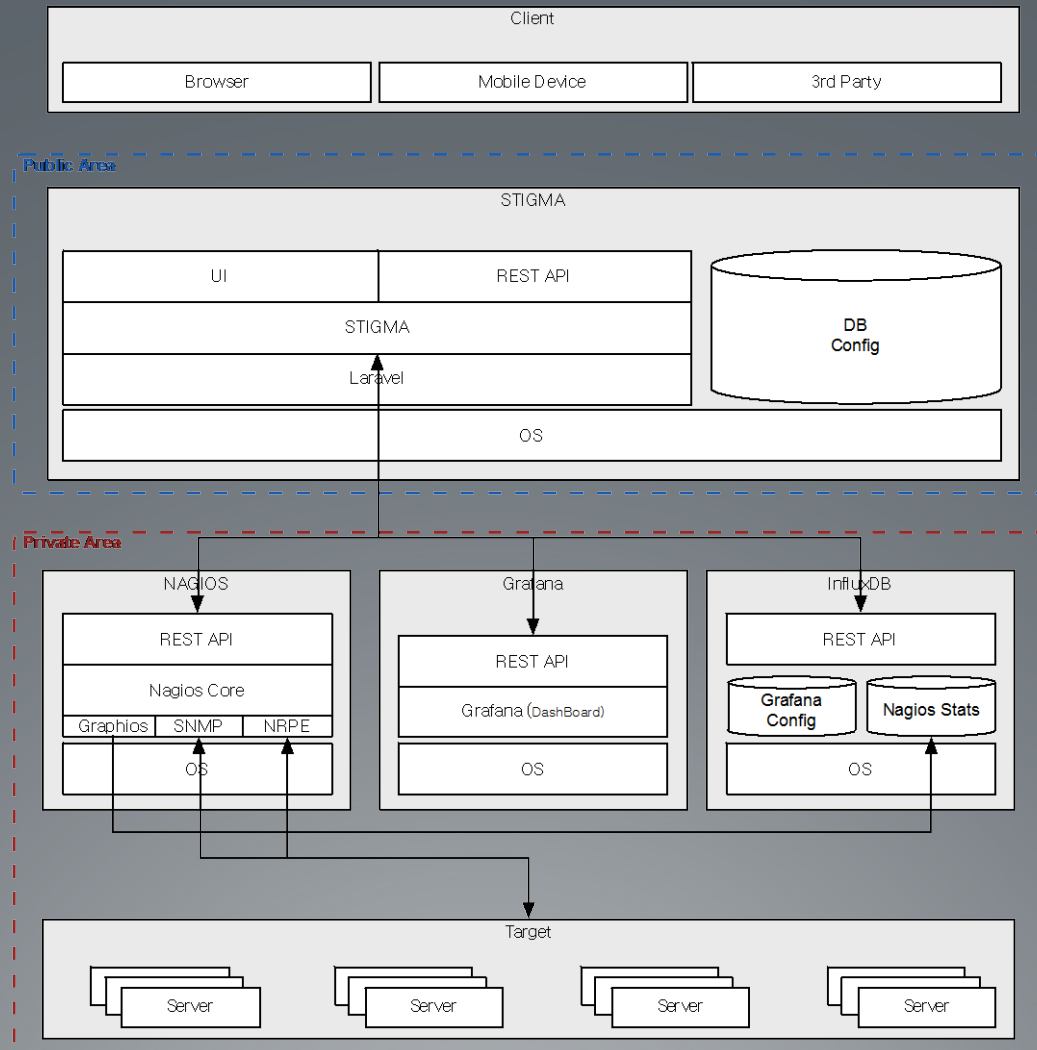
2 OpenSource

- 오픈소스 개발 방식
 - 공통의 플랫폼을 공유
 - 제품을 공동의 성과로 만들어냄
- Package
 - 기능별로 패키지 개발
- REST API
 - 자원의 식별
 - 메시지를 통한 리소스의 조작

About STIGMA

About STIGMA

1 Architecture



2 개발 환경

- OpenShift
- Amazon AWS EC2
- Nagios
- MySQL
- Grafana
- InfluxDB
- Slack
- Trello
- Github
- Omnigraffle
- Balsamiq Mockups

3 개발 언어

- PHP
- Laravel
- AngularJS
- Foundation

Pros and Cons

- Provisioning 가능
 - Nagios에 대한 기술력이 없어도 STIGMA를 배포함과 동시에 IaaS, PaaS 플랫폼에 필요한 서비스를 Deploy하여 짧은 시간 내에 모니터링 환경을 구성할 수 있음
 - 스타트업, 팀 / 프로젝트 단위 모니터링 환경을 구성하고자 할 때 유용
- 각 모듈별 API 가능
 - STIGMA를 이루고 있는 각각의 서비스들은 모듈화 되어 있으며 단독 서버로 배포됨
 - REST API를 통해서 제 3의 서비스를 개발 할 수 있을 뿐 아니라 별도의 솔루션으로 대체 가능
- OpenShift (RedHat PaaS Solution)
 - 프로세스 자동화로 서비스 제공 절차를 간소화
 - 빌드 및 배포를 자동화시킴으로써 애플리케이션 개발시 개발자의 업무 효율성 강화

Pros and Cons

- Custom Cartridges
 - OpenShift v2.2 기반에서 Custom Cartridge를 생성하고 적용하기 어려움
 - OpenShift v3 부터 Docker Container 도입

Demo

04

Demo

Next Step

- Docker
 - Amazon AWS EC2에 provisioning 되어있는 부분을 Docker 환경으로 migration
 - STIGMA의 모든 서비스가 OpenShift 3.0에서 구동되도록 지원
- 서비스 모니터링
 - NRPE(Nagios Remote Plugin Executor)를 통해 다양한 서비스를 모니터링 할 수 있도록 command 개발
- Graph
 - Dynamic dashboard API
- Report
 - 모니터링 대상에 대해 성능 및 이벤트 내용을 보고서 형태로 제공

감사합니다