Zabbix Web & Log 모니터링

구축 목적

- Web 사이트의 상태확인 및 호스트의 Log 모니터링

[Zabbix 서버 구축 후 웹에서 진행!]

WEB 모니터링 설정

1. 웹 모니터링 템플릿 생성

- 설정 - 템플릿 - 템플릿 작성

  < 템플릿 탭 >

  - 이름 : WebSite URL Monitoring
  - 그룹 : Templates

  < 매크로 탭 >

  - 매크로 : {$WEB.URL}

- 생성 !

---

2. 트리거 생성

- 설정 - 템플릿 - WebSite URL Monitoring의 트리거 - 트리거 작성

  < 웹사이트 응답코드 에러 - 트리거 >

  - 이름 : 웹사이트 응답코드 에러 : {$WEB.URL}
  - Operational data : Response code : {ITEM.LASTVALUE1}
  - 심각도 : 가벼운 장애
  - 조건식 : last(/WebSite URL Monitoring/web.test.rspcode[Website Health Check: {$WEB.URL},Webpage Availability])<>200
  - 생성 !

  < 웹사이트 접근 불가 - 트리거 >

  - 이름 : 웹사이트 접근 불가: {$WEB.URL}
  - 심각도 : 가벼운 장애
  - 조건식 : last(/WebSite URL Monitoring/web.test.fail[Website Health Check: {$WEB.URL}])<>0
  - 생성 !

  < 웹사이트 접속 지연 - 트리거 >

  - 이름 : 웹사이트 접속 지연 : ({ITEM.LASTVALUE}): {$WEB.URL}
  - 심각도 : 가벼운 장애

    - 조건식 : avg(/WebSite URL Monitoring/web.test.time[Website Health Check: {$WEB.URL},Webpage Availability,resp],10m)>2
    - 생성 !

---

3. 호스트 생성

- 설정 - 호스트 - 호스트 작성

  < 호스트 탭 >

    - 호스트 명 : URL_test1
    - 템플릿 : WebSite URL Monitoring
    - 그룹 : url

  < 매크로 탭 >

    - 매크로 : {$WEB.URL}
    - 값 : https://www.google.com (모니터링 하고싶은 URL 입력)

- 생성 !

---

4. 웹 시나리오 작성

- 호스트 - 설정 - URL_test1 웹 - Web 시나리오 작성

  < 시나리오 탭 >

    - 이름 : Website Health Check: {$WEB.URL}
    - 갱신 간격 : 20s
    - 시도 횟수 : 5
    - 에이전트 : Chrome 80 (Windows)

  < 스텝 탭 >

    - 추가
    - 이름 : Webpage Availability
    - URL : {$WEB.URL}
    - 리다이렉트를 따라간다 체크
    - 타임아웃 : 15s
    - 요구 스테이터스 코드 : 200

- 생성 !

TEST

- 웹 감시에서 모니터링 값이 정상적으로 나오는지 확인!

< Error 발생 >

- Connection timeout : 해당 URL의 IP로 나갈수 있게 방화벽 설정되었는지 확인 (80 , 443)

★ 정상적으로 작동된다면 WEB 모니터링 설정 완료 ! ★

Log 모니터링 설정

1. Log 모니터링 템플릿 생성

- 설정 - 템플릿 - 템플릿 작성

  < 템플릿 탭 >

  - 이름 : System Log Template
  - 그룹 : [Log 모니터링을 원하는 서버 그룹]

  < 매크로 탭 >

  - 매크로 : {$SYSLOG}
  - 값 : /var/log/messages
    - ubuntu의 경우 messages 파일이 없으면 아래 내용 진행

```
# vi /etc/rsyslog.d/50-default.conf
-------------------------------------------------
*.=info;*.=notice;*.=warn;\
    auth,authpriv.none;\
    cron,daemon.none;\
    mail,news.none          -/var/log/messages
-------------------------------------------------
( 해당 부분 주석 해제 )

# systemctl restart rsyslog

* messages 파일 생겼는지 확인!
```

- 생성 !

2. 아이템 생성

- 설정 - 템플릿 - System Log Template 아이템 - 아이템 작성

  < 아이템 탭 >

  - 이름 : {$SYSLOG} Log
  - 종류 : ZABBIX 에이전트 (액티브)
  - 키 : log[{$SYSLOG},,,,skip]
  - 데이터형 : 로그
  - 갱신 간격 : 10s

  < 태그 탭 >

- 이름 : Application
- 값 : Log

- 생성 !

---

3. 트리거 생성

- 설정 - 템플릿 - System Log Template 트리거 - 트리거 작성

< anomaly status 트리거 >

- 이름 : {$SYSLOG} anomaly status
- Operational data : System Error : {ITEM.VALUE}
- 심각도 : 가벼운 장애
- 조건식 : count(/System Log Template/log[{$SYSLOG},,,,skip],#30,"regexp","\b[eE]rror")>=1
- 수동으로 클로즈 허가 체크
- 생성 !

< warn_test 트리거 >

- 이름 : warn_test
- 심각도 : 경고
- 장애의 조건식 : find(/System Log Template/log[{$SYSLOG},,,,skip],5s,"like","warn") =1
- 정상 이벤트를 생성 : 복구조건식
- 복구조건식 : find(/System Log Template/log[{$SYSLOG},,,,skip],5s,"like","warn") =0
- 생성 !

< LOGFILE Alert 트리거 >

- 이름 : LOGFILE : {$SYSLOG} Alert
- 심각도 : 경고
- 장애의 조건식 : find(/System Log Template/log[{$SYSLOG},,,,skip],10s,"like","alert") =1
- 정상 이벤트를 생성 : 복구조건식
- 복구조건식 : find(/System Log Template/log[{$SYSLOG},,,,skip],10s,"like","alert") =0
- 장애 이벤트의 생성모드 : 복수
- 생성 !

< fault_test 트리거 >

- 이름 : fault_test
- 심각도 : 경고
- 장애의 조건식 : find(/System Log Template/log[{$SYSLOG},,,,skip],5s,"like","fault") =1
- 정상 이벤트를 생성 : 복구조건식
- 복구조건식 : find(/System Log Template/log[{$SYSLOG},,,,skip],5s,"like","fault") =0
- 생성 !

< fail_test 트리거 >

- 이름 : fail_test
- 심각도 : 경고

    - 장애의 조건식 : find(/System Log Template/log[{$SYSLOG},,,,skip],5s,"like","fail") =1
    - 정상 이벤트를 생성 : 복구조건식
    - 복구조건식 : find(/System Log Template/log[{$SYSLOG},,,,skip],5s,"like","fail") =0
    - 생성 !
    -

< err_test 트리거 >

    - 이름 : err_test
    - 심각도 : 경고
    - 장애의 조건식 : find(/System Log Template/log[{$SYSLOG},,,,skip],5s,"like","error") =1
    - 정상 이벤트를 생성 : 복구조건식
    - 복구조건식 : find(/System Log Template/log[{$SYSLOG},,,,skip],5s,"like","error") =0
    - 생성 !

< critical_test 트리거 >

    - 이름 : critical_test
    - 심각도 : 경고
    - 장애의 조건식 : find(/System Log Template/log[{$SYSLOG},,,,skip],5s,"like","critical") =1
    - 정상 이벤트를 생성 : 복구조건식
    - 복구조건식 : find(/System Log Template/log[{$SYSLOG},,,,skip],5s,"like","critical") =0
    - 생성 !

---

4. 호스트 설정

- 설정 - 호스트 - 해당 호스트 클릭
    - 템플릿 : System Log Template 추가
- 갱신 !

TEST

- 모니터링 - 호스트 - 최근 데이터 - log 검색 - 해당 아이템 클릭 or 이력 확인

> 정상적으로 Log 올라오는지 확인

★ 정상적으로 작동된다면 Log 모니터링 설정 완료 ! ★
