

# 2019 KBS 미디어창의기술전 우수상 작품 소개

## 파일 기반 방송시스템 원격관리솔루션 자체 개발 웹 기반 원격접속관리 솔루션 Guacamole

글: 권영부 KBS 청주방송총국 기술국

파일 기반 시스템으로의 전환은 제작 워크플로우에 많은 변화를 가져왔다. 가장 큰 변화는 영상원본을 저장하는 방식이다. 물리적 저장매체를 기반으로 하는 리니어한 편집에서 대규모 네트워크 스토리지로 영상을 분산 저장하는 방식으로 바뀌면서, 네트워크에 연결된 어느 NLE에서든 영상파일을 바로 가져와 편집할 수 있게 되었다. 제작진은 오롯이 편집업무에 집중할 수 있게 되었지만, 동시에 한정된 자원(네트워크 트래픽, 스토리지) 안의 무분별한 남용을 막기 위해, 프록시 편집, 스토리지 계층화, 사용자별 권한 관리 등 다양한 컴포넌트(서비스)들이 요구되었고, 이를 체계적으로 운용 관리하는 시스템엔지니어라는 새로운 Job도 등장하게 되었다.

(나열한 서비스를 패키징한 시스템이 NPS(Network Production System)이며 서비스들은 자동화 워크플로우를 통해 유기적으로 동작한다) 초기에는 효율적인 자원관리시스템(MAM)이 정립되기까지 NPS 시스템의 외연을 넓혀갔다면, 지금에는 ALL-IP, ALL-FILE 기반의 제작환경으로 변화가 요구되면서 NPS에 통합 연계되는 시스템이 늘어났다. 그리고 시스템관리자의 고민도 늘었다. 시스템관리자의 업무부담이 지속해서 증가하기 때문이다. 그에 대한 원인을 개인적인 관점에서 분석해보았다.

첫째, IP 기반 시스템은 관리 부담이 어렵다. IP 기반 장비들은 유기적으로 연결되어 있어, 장비별로 담당자를 나누기보다 시스템별로 소수의 인원에 의해 중앙집중형 방식으로 관리되고 있다. 중앙집중형 관리방식이 가지는 의미 자체가 효율성과 경제성을 함의하기 때문에 소수의 인원이 광범위한 시스템을 관리하는 것은 어찌 보면 당연한 일이다. ALL-IP 기반의 제작 워크플로우의 변화에 맞춰 파일 기반 방송시스템은 나날이 확장하고 있다. 문제는 확장되다 보면 유관부서 간에 관리영역이 중복되기도 할 텐데 업무분담이 되지 않고 대개 한쪽으로 책임과 업무가 쏠리게 된다. 설명했듯이 파일 기반 시스템은 유기적으로 연결되어 있어, 시스템 유관부서 간 공조하여 풀어나가야 할 일은 항상 존재하나 현실적으로 쉽지가 않다.

두 번째는 효율적인 시스템 운영관리가 이뤄지기 위해 다양한 편의와 기능을 제공하는 소프트웨어를 사용하면 되겠지만, 대부분 고가의 소프트웨어이기에 개인의 편리함을 위해 회사에 구매해달라고 요청하기가 쉽지가 않다. 예로 다양한 원격프로토콜(RDP, VNC, SSH, FTP 등)을 지원하고, 원격접속 및 암호를 관리해주는 대표적 원격관리 소프트

웨어인 Devolutions 사의 Remote Desktop Manager(Windows 계열)과 royal apps 사의 royal ts(mac OS 계열)가 있다. 또한, 인터넷만 연결되어 있다면 어디서든 사내 컴퓨터에 접속 및 원격제어를 할 수 있게 하는 Teamviewer, anydesk와 국산 소프트웨어인 알서포트도 있다.

필자는 2년간 파일 기반 시스템 유지보수 업무를 맡으면서, 이러한 고민을 조금이라도 덜기 위해 다양한 방법을 찾아보려 했고, 결국 오픈소스에서 그 답을 찾았다. 바로 Apache Project 중 하나인 Apache Guacamole을 이용해보는 것이었다.

## 오픈소스 Apache Guacamole

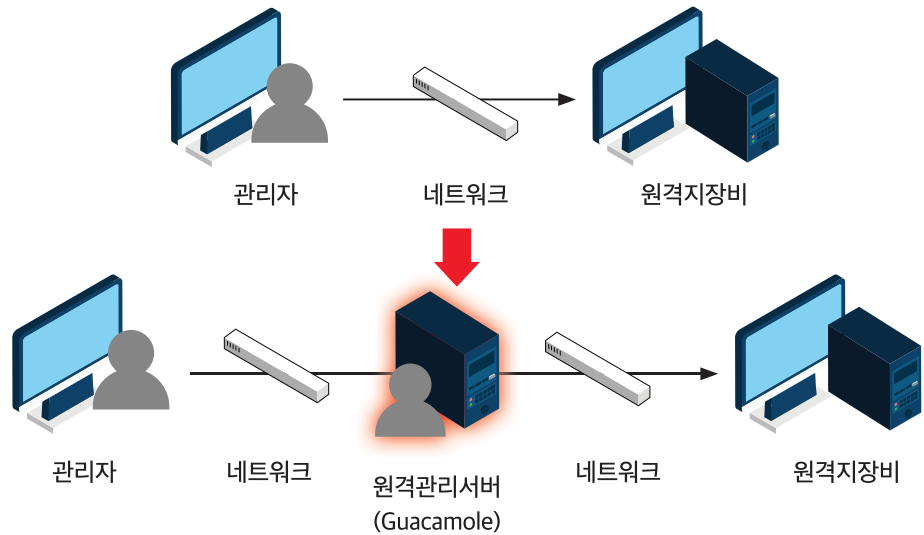


그림 1. 게이트웨이 역할을 하는 Guacamole

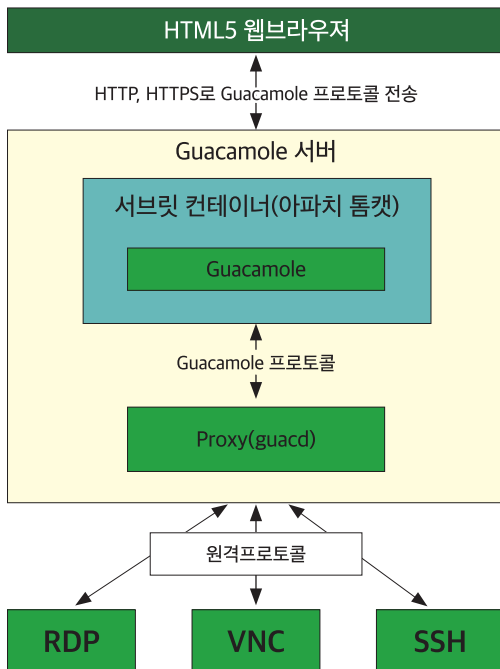


그림 2. Guacamole 아키텍처

Apache Guacamole은 웹 기반 원격관리 솔루션이다. 기존 원격접속방식은 클라이언트가 원격 장비에 접속하기 위해 별도도구가 필요했는데, 웹앱 기반 Guacamole 솔루션을 이용하면 클라이언트는 웹브라우저만 있어도 쉽게 원격접속이 가능하다. Guacamole 서버는 클라이언트 대신 원격 장비와 연결을 맺고, 주고받은 데이터를 중개해주는 게이트웨이 역할을 하게 된다.

## 아키텍처

Guacamole은 Guacamole server, Guacamole Client, 그리고 Database로 구성된다. 서버는 Tomcat(WAS), client는 서블릿 컨테이너가 배포한 웹앱이 된다. DB는 MYSQL, MARIADB와 postgresql을 지원하며, 사용자 계정, 원격호스트, 접속 이력 등을 관리한다. 서버에 설치되는 guacd 데몬은 원격프로토콜(RDP, VNC, SSH)을 Guacamole 서버가 이해하는 프로토콜로 변환하는 프록시 역할을 한다.

제공하는 기능 및 특징

앞서 언급했듯 필자가 고민했던 내용을 다시 적어보면, 시스템이 커지면서 유관부서 간 협업이 어렵고, 효율적인 업무수행을 위한 원격관리 툴이 필요하다고 했다. 이러한 고민을 Guacamole 솔루션이 해결해주었다. 다음은 Guacamole이 제공하는 주요 특징들이다.

- 이 솔루션이 필요했던 가장 큰 이유이기도 한 원격접속정보 관리기능은 유지보수업무에 상당한 도움을 주었다. 기억력이 좋지 않았던 필자는 원격작업을 할 때마다 장비 IP, 암호를 정리한 엑셀 파일을 보면서 작업했는데, 이 솔루션을 이용하면서 그럴 필요가 없어 업무효율이 많이 향상되었다.
- 급히 원격작업을 해야 할 일이 발생하였을 때, 방화벽 정책이 적용된 관리자 PC까지 갈 필요 없이, guacamole 서버로 접속 가능한 사내 PC가 있는 어느 곳에서나 원격작업을 할 수 있게 됐다. 사내 네트워크 정책상 웹서비스인 80 포트는 막을 수 없기에 가능한 일이었다. 일례로 제작진에서 민원이 들어와 문제 파악을 위해 현장에 직접 찾아가는 일이 빈번하게 있다, 결국, 해결법이 원격지에 있는 서버의 설정을 바꾸어주면 되는 건데, 관리자 PC까지 다시 돌아가 처리를 하던 매번 번거로운 일이었다. 이젠 guacamole을 이용하여 현장에서 문제를 즉시 해결할 수 있게 되었다.

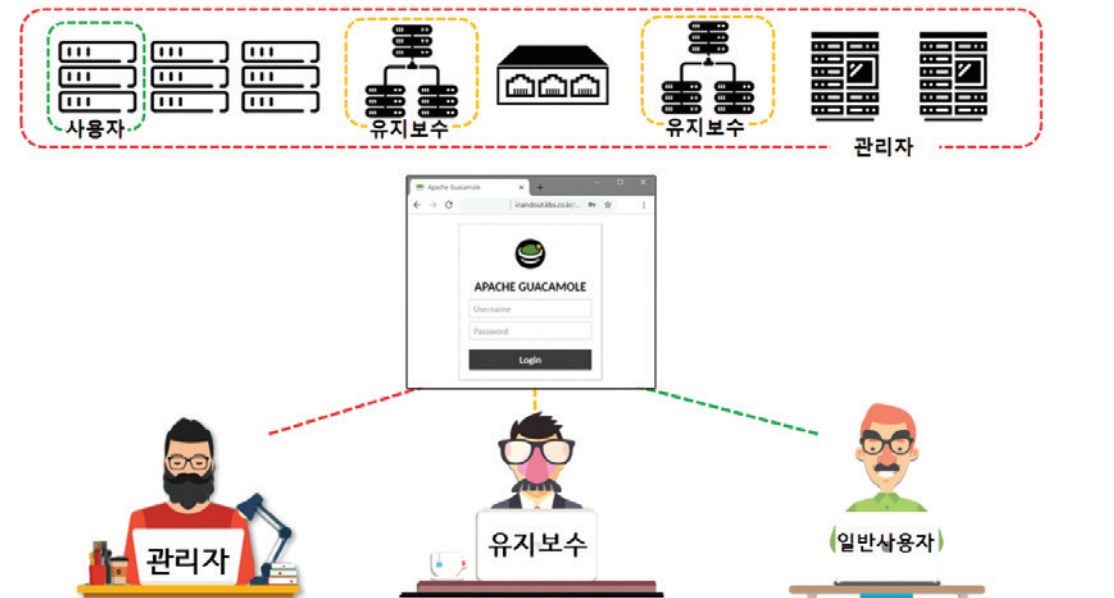


그림 3. 계정별 시스템 접근 권한 구분

- 타부서직원이 특정 장비에 원격접속을 할 필요가 있을 때, 일반적으로 원격 장비에 대한 원격프로토콜, IP, 암호를 알려줘서 보안이 노출되는 문제가 있는데, Guacamole을 이용하면 그럴 필요 없이 해당 장비만 접속 가능한 임시 계정을 생성하여 접근 가능토록 해주면 보안이 노출되는 문제를 차단할 수 있다.
- Guacamole은 클라이언트-서버 모델이다. 이 모델을 따르면 클라이언트 간에 session 정보를 공유할 수 있다. 이는 태생적으로 화면공유가 되지 않던 RDP와 SSH 원격접속화면을 VNC처럼 공유할 수 있게 하는 기능을 제공한다. 이 기능을 사용하여 유관부서 간에 작업내용을 실시간으로 공유할 수 있어, 실제로 효율적으로 업무를 완수했던 사례가 있다.

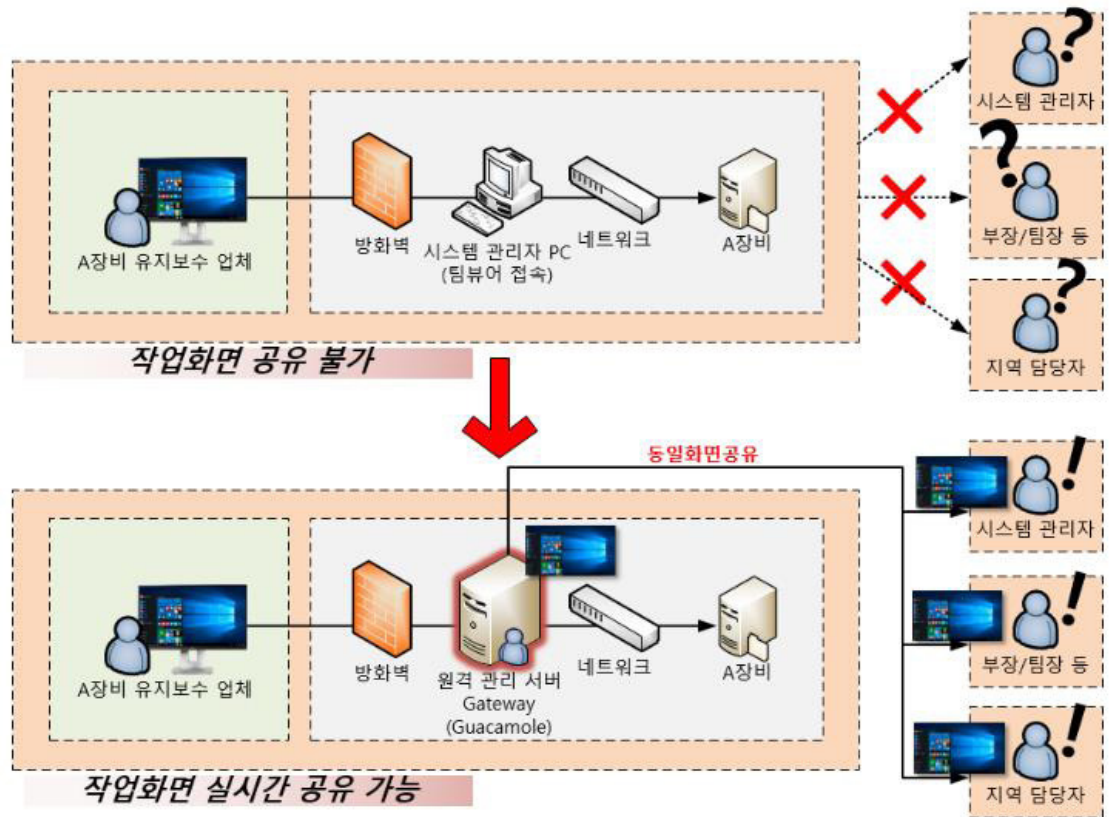


그림 4. 실시간 원격작업화면 공유

그 외 화면녹화와 가상 네트워크 드라이브 마운트 등 다수의 유용한 기능을 제공한다. 자세한 설명은 정식 홈페이지의 Documentation 페이지를 참고하면 된다.

## 설치 방법

설치 방법은 Guacamole 1.0 버전으로 집필하였으며, 최근 출시한 1.1 버전(2020.01.26.)에도 적용 가능할 것으로 생각한다.

설치 방법은 두 가지다. Native 방식과 Docker 방식이 있다. Native 방식으로는 리눅스 OS에만 설치할 수 있고 가상화 기술인 Docker 방식을 선택하면 윈도우에서도 설치할 수 있다. Native 방식은 수동으로 소스 컴파일하는 방식이어서 설치과정에 유틸리티와 프로토콜을 선택적으로 설치할 수 있다. 제공하는 유틸리티, 프로토콜 모두 유용하므로 전체 설치를 권장한다. 다만 설치 시간이 길어, 번거롭다고 생각하면 Docker 방식으로 설치해보자. 도커 허브에서 Apache가 미리 제작한 최신 도커 이미지를 내려받아 설치하는 방식으로 10분 안에 모든 설치를 마칠 수 있다. 또한, Docker 네트워크 기능을 활용하여 간단하게 포트포워딩 작업도 할 수 있으니 Docker 방식이 여러모로 편리하다.

도커(Docker)에 대해 간단히 설명을 덧붙이자면, 도커는 컨테이너 기반의 오픈소스 가상화 기술로, 기존의 하이퍼바이저와 베어메탈 방식의 가상화 기술보다 더 경량화된 버전으로 생각하면 된다.

필자는 스크립트 형식으로 설치하는 Docker-compose 방식으로 진행하였다. 그 이유는 리눅스사용에 능숙하지 못한 사람이라 하더라도 스크립트 파일만 있으면 쉽게 설치 할 수 있어 추후 많은 곳에서 사용할 수 있길 기대하였다. Guacamole을 실행하는데 사용되었던 도커 이미지를 정리해보았다.

| 도커 이미지명                   | 도커 이미지 설명                                                                                                                                                                                                                      |
|---------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| guacamole/guacamole(필수)   | 톰캣(WAS)과 모든 기능을 포함하는 웹앱(클라이언트)을 패키징한 이미지                                                                                                                                                                                       |
| guacamole/guacd(필수)       | 프록시 역할을 하는 guacd 서비스가 설치된 이미지                                                                                                                                                                                                  |
| mariadb(필수)               | 계정, 시스템 환경설정 등을 관리하는 DB 이미지(MYSQL, postgresql도 가능함)                                                                                                                                                                            |
| phpmyadmin/phpmyadmin(옵션) | guacamole 솔루션에서 환경설정 등을 import/export 하는 메뉴가 없어서, DB SQL로 DB 백업해야하는 번거로움이 있다. SQL이 익숙하지 않다면 phpmyadmin과 같은 웹 기반 DB 관리 도구를 사용하길 권장한다.                                                                                           |
| guacenc(옵션_자체제작)          | 자체적으로 만든 Ubuntu 기반 도커이미지로, guacenc와 ffmpeg 유틸리티가 설치되어 있다. guacamole 솔루션은 원격접속화면을 녹화하는 기능이 있는데, 녹화 후 생성되는 Raw 데이터를 영상코덱으로 변환하는 유틸리티가 guacenc이다. 컨버팅 포맷은 m4v로 고정되어 있어 활용률이 높은 영상코덱으로 변환하기 위해 2차 컨버팅이 필요하며, 이는 ffmpeg 유틸리티가 수행한다. |

만약 Ubuntu로 설치를 진행한다면, phpmyadmin이 점유하는 80 포트와 OS에 디폴트로 동작하는 웹서비스(18 버전 apache, 19버전 nginx) 간 포트 충돌을 막기 위해 기존 웹서비스는 중지해야 한다. 설치 완료 후 웹브라우저에서 `http://localhost:8080/guacamole`로 들어가면 guacamole 메인 페이지가 보인다. 디폴트 계정/비밀번호는 `guacadmin/guacadmin`이다.

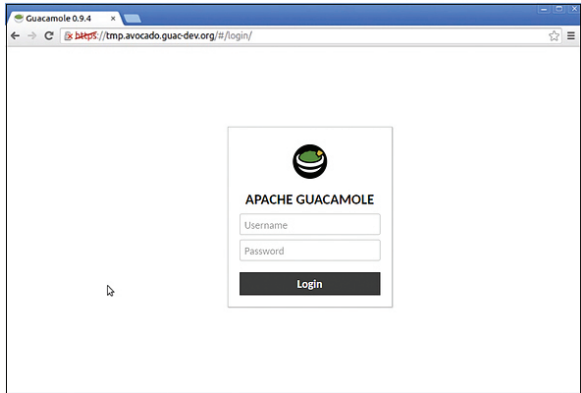


그림 5. Guacamole 메인 페이지

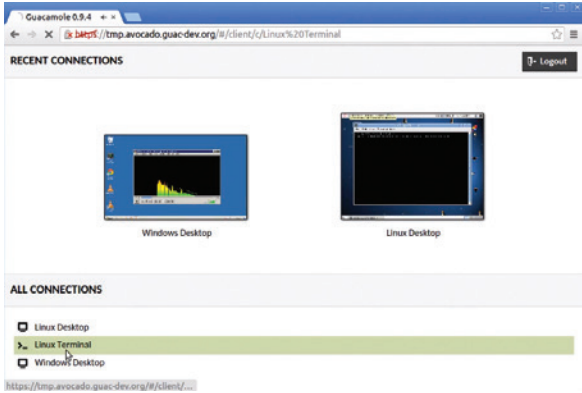


그림 6. Guacamole 대시보드 화면

### 설치 운용 시 참고 사항

- Guacamole의 VNC 인증방식은 username/password가 아닌, only password 인증방식이다. VNC 기반의 Apple ARD(Apple Remote Desktop)는 디폴트로 username/password 인증방식만 허용한다. Guacamole의 VNC 프로토콜로 MacOS에 접속하고자 할 때 password 인증방식을 활성화하면 된다. (MacOS - 시스템환경설정 - 공유 - 화면공유 - password 인증방식 활성화)
- Guacamole에서 화면녹화 기능을 사용하고자 할 때, 화면 해상도 값을 지정해줘야 한다. 빈칸으로 두면 dynamic resolution이 자동활성화되어 웹브라우저 창사이즈에 따라 원격화면 해상도가 바뀐다. 문제는 화면녹화 raw 파일에 해상도 정보가 없으면 guacenc 유틸리티를 이용한 컨버팅 작업 시 에러가 발생한다.
- windows 10과 windows server 2016 이후 버전에서 RDP 접속이 원활하지 않을 경우, 레지스트리 설정을 변경해야 한다.

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server\
WinStations\WDP-Tcp]
"SecurityLayer"=dword:00000001
"UserAuthentication"=dword:00000000
```

위의 항목처럼 디폴트값으로 설정되어있으면 접속에 문제가 없을 것이다. guacamole 1.1은 freerdp가 2.0 버전으로 업데이트되어 문제가 해결되었을 수도 있지만, 직접 확인은 못 해보았다.

## 활용 방안

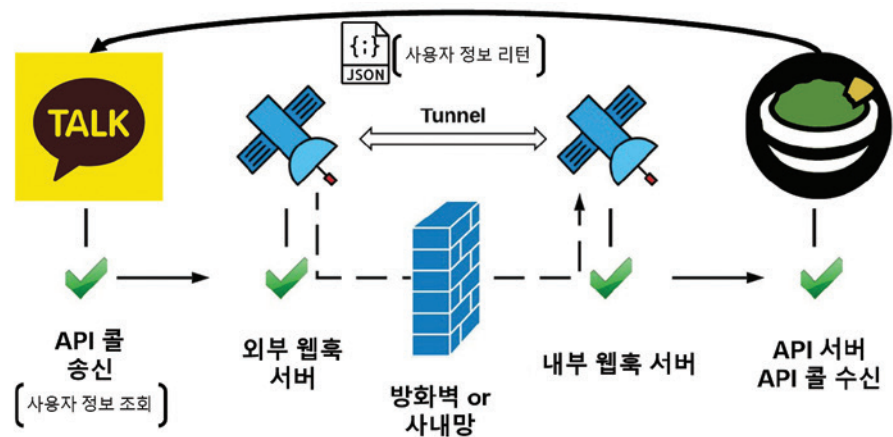


그림 7. webhookrelay 서비스를 활용한 외부 통신

Guacamole에 접속하는 사용자도 많아지고 관리대상이 늘어나게 되며, Guacamole 원격접속기록 및 로그인한 사용자, 활성화된 세션 정보 등을 서버에 접속하지 않고도 모바일기기로 알림서비스로 받을 수 있다면 유지보수업무에 도움이 될 것으로 생각했다. 그런 역할을 하는 서비스를 구현하고자 했을 때 3가지를 고려해야 했다.

첫째는 어떤 방식이 가장 간단하고 빠르게 정보를 전달할 수 있을까였다. 생각해낸 것은 카카오톡을 활용하는 방법이였다. 예전에 회사 선배께서 카카오톡을 활용한 구내식당 메뉴 알리미 서비스 개발한 것을 보고 생각해낸 아이디어였다. 카카오톡 채널을 개설해서 카카오톡 챗봇에 사전 정의된 질의를 하면 자체 개발한 API 서버가 요청을 받아 관련 정보를 DB에서 읽어와 응답하는 방식을 구현하기로 했다.

둘째는 카카오 서버와 회사 내부에 있는 API 서버가 서로 통신하기 위한 퍼블릭 서비스를 구현하는 방법이었다. API 서버에 공인 IP를 부여하는 방식이 가장 간단하지만, 그보다 더 저렴하게 구현할 수 있는 방법을 찾아 보니 webhookrelay라는 tunneling service를 제공하는 솔루션을 이용하기로 했다. 방화벽에 막혀 있는 API 서버는 일반적으로 외부에서 접근이 불가능하지만, 이 솔루션을 이용하면 외부에서 접근 가능한 URL을 제공하여, web API로 데이터를 주고받을 수 있게 한다. webhookrelay에서 지원하는 데이터 형식은 Body, Header, URL query 등이 있다. 카카오톡 역시 데이터를 JSON 형식으로 주고받기 때문에 구현하고자 하는 서비스에 충족했다. 가격도 free plane이 월 150 API call로 시작해서, basic 플랜은 \$4.5에 1500 API call을 제공하기에 아주 저렴하게 퍼블릭 서비스를 구현할 수 있었다.

셋째는 DB에서 정보를 가져오기 위한 API 서버 개발이었다. API 서버는 python 기반의 flask 프레임워크를 이용하기로 했는데, 이유는 간단하게 REST API를 구현할 수 있었기 때문이다. guacamole의 DB명은 guacamole\_db이다. 여기서 필요로 하는 정보를 가진 테이블은 2개다.

- guacamole\_user\_history : 사용자 접속기록 및 현재 접속자
- guacamole\_connection\_history : 원격접속 기록 및 현재 연결된 세션

위 테이블에서 얻은 값을 적절히 가공하여 카카오톡 챗봇으로 전달하였다. 참고로 JWT(Jason Web Token) 방식을 사용하여 로그인 사용자, 접속기록, 사용자 접속 끊기 등을 구현할 수 있는 법도 있다. 그렇게 개발한 API 서버를 통해 카카오톡 챗봇으로 guacamole의 DB에서 원하는 정보를 가공해서 표출할 수 있었다.

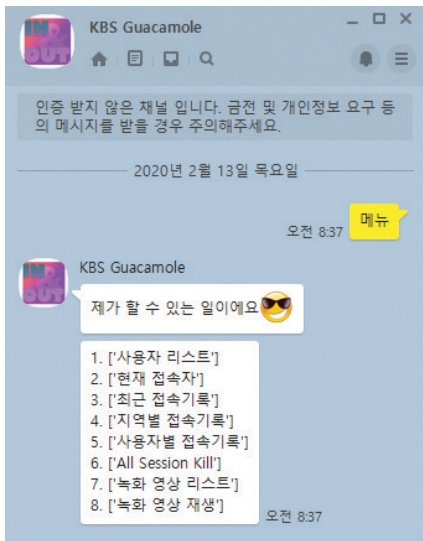


그림 8. 카카오톡 챗봇 메뉴

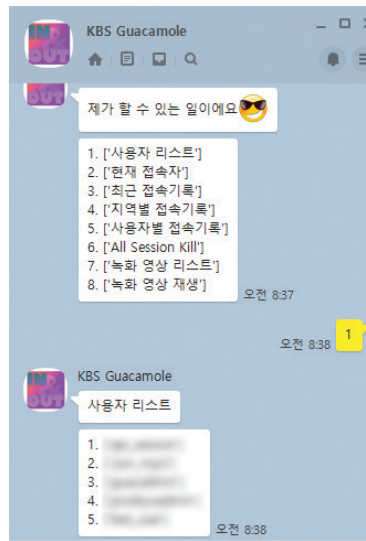


그림 9. 카카오톡 챗봇에서 사용자 정보 불러오기

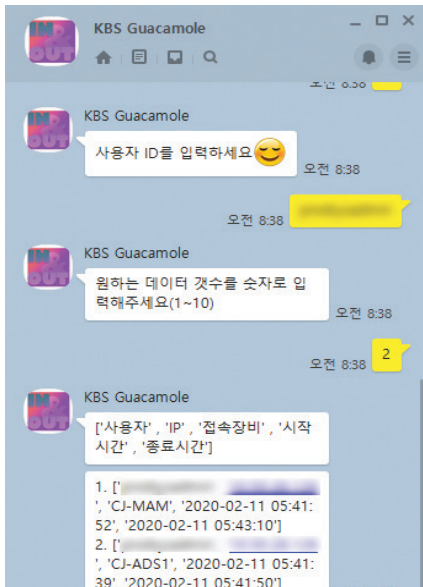


그림 10. 카카오톡 챗봇에서 접속기록 가져오기

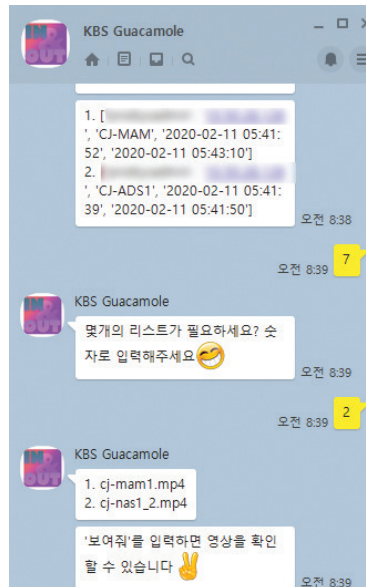


그림 11. 녹화된 영상 재생하기

## 향후 목표

guacamole을 활용해서 구축하고자 했던 최종목표는 원격작업을 녹화한 영상을 일목요연하게 웹페이지에서 보여주는 동영상 공유 웹사이트를 만드는 것이었다. 유지보수업무를 하면서 느꼈던 아쉬운 점은 중요한 원격작업내용을 영상으로 만들어 많은 사람과 공유했으면 하는 개인적인 바람이 있었다. 그리고 문서형식의 작업일지보다 실제 작업 영상을 본다면 이해도 빠르고, 교육자료로도 활용될 수 있을 것으로 생각했다. 앞서 guacamole을 이용해 카카오톡으로 녹화된 영상을 스트리밍한 것을 구현한 바 있다. 시간이 허락한다면 이보다 더 업그레이드된 동영상 공유 웹사이트 구축을 실행으로 옮기고 싶다.

## 마치며

IP 기반 방송장비가 소형화, 경량화되면서 레거시 방송시스템에 비해 차지하는 시스템 공간이 작아졌다. 다만 물리적으로 크기가 작아졌다고 시스템이 단순해진 것은 아니다. 시스템 통합연계와 꾸준한 고도화 사업을 통해 시스템 복잡도는 더 높아진 상황이며, 이로 인해 관리 포인트도 훨씬 많아졌다. ALL-IP, ALL-FILE 기반의 방송시스템이 도입되면 이보다 더욱 복잡한 시스템이 출현할 것이다. 그렇다면 시스템 관리방식에도 지금과는 다른 새로운 접근법이 필요할 것이며, 오퍼레이터와 시스템관리자로 업무를 나눌 게 아니라 모두가 협업하는 분위기가 조성되어야 할 것이다. 그런 어려움과 고민이 상존하는 이 시점에 Guacamole 솔루션이 보여주는 가능성은 긍정적인 변화를 이끌어주길 기대한다.

설치과정에서 수많은 시행착오를 겪으면서 많은 것을 공부하게 되었다. docker와 webhook에 대한 개념이 없었던 나로서 새로운 기술에 대해 공부할 수 있게 되어 미디어창기술전을 준비했던 게 귀중한 경험이 되었다. 무엇보다 같이 고생하고 도와준 팀원들이 없었다면 수상의 영예를 누리보지 못했을 것이다.

참고로 Docker-compose 스크립트 파일과 JWT 방식으로 세션 정보를 가져오는 코드를 다운받을 수 있는 링크 (<https://github.com/gkisanet/docker-compose/>)를 첨부하니 설치에 참고하기 바란다. 🙏