

VoIP Test

Part 1

Design and describe a scalable highly-available cloud VoIP system, based on FreeSWITCH, with the following requirements:

1. The system should provide all the basic features a modern multi-tenant VoIP/PBX should have, such as, inbound, outbound and local calls, IVR, call queues, conferencing, ring groups, voicemail, instant messaging, call recording, CDR storage, WebRTC, etc.
2. Users should configure the system, listen to recordings, view reports and statistics, etc., through a web application
3. Calls should be distributed between FreeSWITCH nodes for load balancing
4. Call recordings should be stored in a scalable and low-cost storage system
5. The system should be secured
6. The system should recover active calls in case a FreeSWITCH node failed
7. The system should trigger webhooks on call events such as originate, answer and hang-up
8. Logs, node metrics (e.g., CPU, RAM, disk usage, etc.) and SIP packets should be monitored and alerts should be triggered in the case of errors/issues

What to deliver?

1. An architecture diagram depicting the components of the system
2. Description of the architecture. Please also refer to any limitations and/or issues that the proposed system may have.

Part 2

Develop a micro-service that will trigger web-hooks to notify external applications/systems about call events, such as, originate, answer and hang-up.

Requirements:

- The endpoints of the external application/system will be called by the micro-service using the POST method. The user should be able to provide connection timeout (in seconds).
- Event description

Event	Example
Event	Example
Originate	<pre>{ "uuid":"460a36b6-09e7-466b-97d8-903c8cf4fc6c", "callee_number":"0035799123456", "caller_id_number":"+442035040000", "ani":"1002", "direction":"outbound", "call_start":"2020-01-29T07:45:19.000+00:00", "call_start_uepoch":1580283919489709, "action":"originate" }</pre>

Answer	<pre>{ "uuid":"460a36b6-09e7-466b-97d8-903c8cf4fc6c", "callee_number":"0035799123456", "caller_id_number":"+442035040000", "ani":"1002", "direction":"outbound", "answer_start":"2020-01-29T07:45:35.000+00:00", "answer_start_uepoch":1580283935169797, "action":"answer" }</pre>
Hangup	<pre>{ "uuid":"460a36b6-09e7-466b-97d8-903c8cf4fc6c", "callee_number":"0035799123456", "caller_id_number":"+442035040000", "ani":"1002", "direction":"outbound", "call_start":"2020-01-29T07:45:19.000+00:00", "call_start_uepoch":1580283919489709, "answer_start":"2020-01-29T07:45:35.000+00:00", "answer_start_uepoch":1580283935169797, "call_end":"2020-01-29T07:45:39.000+00:00", "call_end_uepoch":1580283939549758, "duration_sec":4, "hangup_cause":"NORMAL_CLEARING", "action":"hangup" }</pre>

- The micro-service should use the FreeSWITCH's event socket
- The endpoints of the external application/system should be provided through a REST API. Endpoints should be stored in a PostgreSQL database.
- The micro-service should resend failed webhooks. Failed webhooks should be stored in a PostgreSQL database.
- The micro-service should store logs
- The micro-service should be deployed using Docker Compose

What to deliver?

Please submit your solution through email. The submission must include:

1. Source code archive
2. Database design & implementation
3. Docker file for the micro-service
4. Docker compose file that will include the micro-service and the database
5. Document with any assumptions made & project guidelines in a [README.MD](#) file