Security Cam Cloud Service

Cloud Service for Hosting Multiple Security Cam Instances

### Introduction The Security Cam (NVR, Network Video Recorder) project is primarily designed to run without the need for a cloud service, inside a secure LAN, with access from WAN being made available with port forwarding. The Cloud Server can be used to provide access to a number of NVR instances without the port forwarding set up, as the NVR makes a client connection to the Cloud Service through which the web interactions are multiplexed.

The Cloud Server is intended to be run at a public internet address, to which instances of Security Cam are configured to make client connections. The NVR does not need to have a local user account set up to connect to the Cloud Service. Instead a user account is created on the Cloud Service for each NVR connected to it. When the user account is created, the required username, password and email address are entered along with the NVR's unique product ID by which the Cloud Service identifies the specific NVR, When an NVR has no local account, it will attempt to connect to the Cloud Service by default.

Description

The Cloud server application consists of a Java (Grails) server side with an Angular 12 web application for the client side. The Angular client has two operating modes, as an admin application (for the admin user), and client mode for users who login to use their NVR. In client mode, the server side acts as a proxy to the connected NVRs with user accounts, so most restful API calls are made through this proxy to the users NVR. The Cloud Server can handle multiple NVRs and each connected NVR must have a user account set up on the Cloud Server to give client access to it. Each NVR is associated with its user account with a unique product ID that is set on the NVR on initial installation, and is entered along with the account data when the account is originally set up. Each connected NVR has its own individual proxy interface to its web server backend. The Cloud Server logs onto the NVRs using a special secure access Cloud Account.

Cloud Service Features

- Hosts multiple NVRs with each one having its own user account
- Admin Access
    - Change admin account password.
    - Show list of connected NVRs
    - Indicate which NVRs have Cloud accounts
    - Indicate which NVRs with Cloud accounts are connected to the Cloud.
    - For each NVR, show number of users viewing the NVR through the Cloud Service.
    - Change users Cloud account password.
    - Change users Cloud account email address.
    - Enable/Disable users Cloud account.
    - Delete Users Cloud Account.
    - Show only accounts where the NVR is offline (not connected to Cloud)
    - Show only connected NVRs with no Cloud user account set up.
    - NVR list filter for user name/product id
- Client NVR access

  - Most features present through direct access to the NVR are present with client access on the Cloud. Camera web admin pages are not accessible through the Cloud as they are with direct NVR access. The Admin functions are not present, though you can add or remove the local NVR account.
  - NVR configuration
  - Add/Remove local NVR account.

Run time platform, for Cloud Server

The current build configuration (as created with./gradlew buildDebFile) is for Raspberry pi V4 running headless (server) version of Ubuntu 23.04 (Lunar Lobster). The application runs on Java on the server side, so it can easily be adapted to other platforms.

Tomcat Web Server

Tomcat 9 (https://tomcat.apache.org/) hosts the server (Web Back End) and client (Web Front End) of the NVR, giving access to these through port 8080.

Web Front End

The Web Front End (client) is an Angular application using Angular CLI version 12.0.5 or later. This forms the user interface of the web application. To get more help on the Angular CLI use `ng help` or go check out the Angular CLI Overview and Command Reference page.

Web Back End

The Web Back End (server) is a Grails application (https://grails.org/), which provides a Restful API for the Angular Web Front End in admin mode. In client mode, the Restful API is mainly from the NVR via the account proxy.

The project is verified to build with the following:-

- Angular CLI: 15.2.0 or greater
- Node: 18.17.1
- npm: 9.9.7
- Package Manager: npm 9.6.7
- Grails Version: 5.3.2
- openjdk version "19.0.2" 2023-01-17
- Gradle 7.6

Using other versions may cause build issues in some cases.

Set up build environment

```
git clone git@github.com:richard-austin/cloud-server.git
cd cloud-server
gradle init
```

Build for deployment to Raspberry pi

```
./gradlew buildDebFile
```

This will create a deb file with a name of the form cloud_*VERSION-nn-ID-dirty*_arm64.deb Where:-

- *VERSION* is the most recent git repo tag
- *nn* Is the number of commits since the last git tag (not present if no commits since last tag.)
- *ID* The last git commit ID (not present if no commits since last tag.)
- *dirty* "dirty" is included in the name if there were uncommitted changes to the source code when built.

When the build completes navigate to where the .deb file was created:-

```
cd xtrn-scripts-and-config/deb-file-creation
```

scp the .deb file to the Raspberry pi

Installation on the Raspberry pi

```
sudo apt update
sudo apt upgrade
```

(restart if advised to after upgrade)

Navigate to where the .deb file is located

```
sudo apt install ./deb_file_name.deb
```

- Wait for installation to complete.
- The Tomcat web server will take 1 - 2 minutes to start the application.
- *If this is the first installation on the Raspberry pi..*
    - *Generate the site certificate..*

      ```
      cd /var/cloud
      sudo ./install-cert.sh
      ```

      Fill in the details it requests (don't put in any information you are not happy with being publicly visible, for example you may want to put in a fake email address etc.)
    - nginx will not have started in the absence of the site certificate, so restart nginx.

```
sudo systemctl restart nginx
```

Initial Setup

Set up admin user account password

The admin account is set up with the default password *elementary*, this should be changed first of all.

- Set a browser to https://*cloud-server_ip_addr*
- Ignore the warning which may be given as a result of the home generated site certificate and continue to application which will show the menu bar.
- Click on the *Log in* option on the meny bar
- Enter *admin* as the user name and *elementary* as the password.
- Click confirm.
- Click on *General* on the right had side of the menu bar.
- Click on *Change Password*
- Enter *elementary* as the current password
- Enter you new password and again in the confirm box.
- Click on *Change Password*
- The new password is now set up.