Deployment and Operations Guide (Runbook)

University of Maryland Global Campus

SWEN 670 Team B (Best)

Summer Semester

Version 2.0

August 4, 2023

Contributors

David Babers, Mohamed Ben Lakbir, Robson De Souza, Collin Hicks

Johnny Huynh, Benny Iko, Scott McCrillis, Jonathan Nagy, Amol Thomare, Max McCool

Version	Issue Date	Description
1.0	07/22/2023	Initial Implementation
2.0	08/04/2023	Milestone 4 Version

Table of Contents

Introduction

This deployment and operations guide is for use in navigating the backend services for the ConvoBuddy Application delivered by Team B during the Summer 2023 semester.

Purpose

The purpose of this document is to explain how the ConvoBuddy application operates from a backend perspective. This manual is to be used by any future graduate students or engineers who are interested in executing the application to see how it functions.

Intended Audience

The intended audience for this document would be the stakeholders of this application and any future graduate students in the SWEN 670 course. In addition, this document would interest any engineers interested in a test case of using ChatGPT to assist people with short-term memory loss.

Stakeholders

Table 1, Stakeholders

Stakeholder Name	**Project Role(s)**
Dr. Mir Assadullah	Client/Professor
Roy Gordon	Project Mentor
Robert Wilson	DevSecOps Mentor
Jonathan Nagy	Team Lead, Developer, Tester, Writer
Amol Thomare	Developer, Tester
David Babers	Project Manager, Developer, BA.
Mohamed BEN LAKBIR	All around, Writer, Tester, QA
Scott McCrillis	BA, Developer
Johnny Huynh	Developer, Writer, Tester, BA
Robson De Souza	Developer, BA, QA, Writer
Benny Iko	Development, Tester.
Collin Hicks	Development, Writer, All around
Kidanu Mekonnen	QA, Writer, BA

Project Documents

Table 2, Project documents.

	Document	**Version**	**Date**
1	Project Management Plan (PMP)	4.0	8/04/2023
2	Software Requirements Specification (SRS)	4.0	7/24/2023

3	Technical Design Document (TDD)	3.0	7/24/2023
4	Programmers Guide (PG)	2.0	7/24/2023
5	Deployment and Operations (DevOps)	2.0	8/04/2023
6	User Guide (UG)	1.0	8/05/2023
7	Test Report (TR)	1.0	8/05/2023

Abbreviations and Acronyms

Table 3, Acronyms and abbreviations.

Abbreviation/Acronym	Description
AMI	Amazon Machine Image
AWS	Amazon Web Services
CA	Certificate Authority
EC2	Elastic Cloud Compute
SSL	Secure Socket Layer
TLS	Transport Layer Security
JDK	Java Development Kit

Definition of Terms

Table 4, Definition of terms.

Terms	Definition

References

Amazon. (n.d.a). Make an AMI public. Retrieved July 18, 2023, from https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/sharingamis-intro.html

Amazon. (n.d.b). What is Corretto 17? Retrieved July 17, 2023, from https://docs.aws.amazon.com/corretto/latest/corretto-17-ug/what-is-corretto-17.html

Amazon. (n.d.c). Run command on your Linux instance at launch. Retrieved July 18, 2023, from https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/user-data.html

Amazon. (n.d.d). What's the difference between SSL and TLS? Retrieved July 29, 2023, from https://aws.amazon.com/compare/the-difference-between-ssl-and-tls/

Apache. (2023a). Reverse proxy guide. Retrieved July 29, 2023, from https://httpd.apache.org/docs/2.4/howto/reverse_proxy.html

Apache. (2023b). Apache module mod_proxy. Retrieved July 29, 2023, from https://httpd.apache.org/docs/2.4/mod/mod_proxy.html

GitHub. (n.d.b). Cloning a Repository. Retrieved July 20, 2023, from https://docs.github.com/en/repositories/creating-and-managing-repositories/cloning-a-repository

GoDaddy.com. (n.d.). What factors affect DNS propagation time? Retrieved July 29, 2023, from https://www.godaddy.com/help/what-factors-affect-dns-propagation-time-1746

Man.archlinux.org. (n.d.). Systemd.unit(5). Retrieved July 18, 2023, from https://man.archlinux.org/man/systemd.unit.5.en

Morel, B. (September 5, 2017). Creating a Linux service with 3rivil. Retrieved July 20, 2023, from https://medium.com/@benmorel/creating-a-linux-service-with-systemd-611b5c8b91d6

NoIP.com. (n.d.). How long until the domain I just registered is active? https://www.noip.com/support/knowledgebase/how-long-until-the-domain-i-just-registered-will-become-active

Workflow

Continuous Integration (CI)

GitHub actions are utilized to validate the build on push and pull requests to the backend_services or backend_test_utility folders in the development branch of repository umgc/summer2023. The following configuration file describes the github actions that result in a build. The build outputs to .apk, as Android is the primary testing platform along with iOS. Unfortunately, there was insufficient time to create and test a iOS CI actions file.

Figure 1. backend_ci.yml Build File.

```
Name: Backend Services CI

on:
  push:
    branches: [“development”]
    paths:
      - “backend_services/**”
      - “backend_test_utility/**”

  pull_request:
    branches: [“development”]
    paths:
      - “backend_services/**”
      - “backend_test_utility/**”

  # allow action to be run from github UI
  workflow_dispatch:

jobs:
  build:
    name: Build backend_services
    runs-on: ubuntu-latest

    steps:
      - name: checkout source
        uses: actions/checkout@v3

      - name: Setup Java environment for Android build
        uses: actions/setup-java@v3
        with:
          distribution: “zulu”
          java-version: “12.x”
          cache: “gradle” # speed up java setup

      - name: Setup flutter SDK
        uses: subosito/flutter-action@v2
        with:
          flutter-version: “3.10.5”
          channel: “stable”
          cache: true # speed up flutter sdk setup

      # backend_services
      - name: Get backend_services dependencies
        run: flutter pub get
        working-directory: ./backend_services

      # Allow violation of info suggestions
      - name: Analyze backend_services project
        run: flutter analyze –no-fatal-infos
        working-directory: ./backend_services


      - name: Run backend_services unit tests
        run: flutter test –coverage
        working-directory: ./backend_services

      # backend_test_utility
      - name: Copy .env.release to .env
        run: |
          cp .env.release .env
        working-directory: ./backend_test_utility

      - name: Get backend_test_utility dependencies
        run: flutter pub get
        working-directory: ./backend_services

      # Allow violation of info suggestions
      - name: Analyze backend_test_utility project
        run: flutter analyze –no-fatal-infos
        working-directory: ./backend_test_utility

      - name: Run backend_test_utility unit tests
        run: flutter test –coverage
        working-directory: ./backend_test_utility

      - name: Build backend_test_utility apk
        run: flutter build apk –dart-define=OPENAI_API_KEY=”${{secrets.OPENAI_API_KEY}}”
        working-directory: ./backend_test_utility

      - name: Upload generated apk to workflow artifacts
        uses: actions/upload-artifact@v1
        with:
          name: backend_test_utility-apk
          path: backend_test_utility/build/app/outputs/flutter-apk/app-release.apk
```

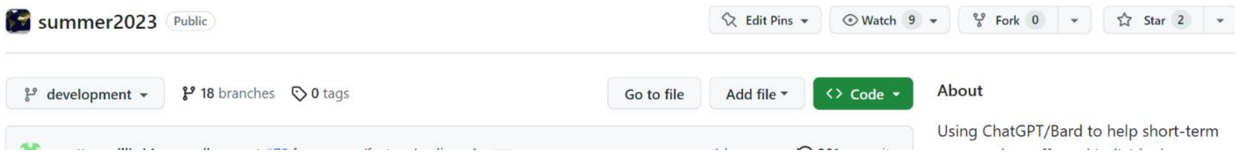
Preparations for Use

Cloning the GitHub ConvoBuddy Repository

In order to have the application to run, the following instructions must be followed to get the application on the user's local machine:

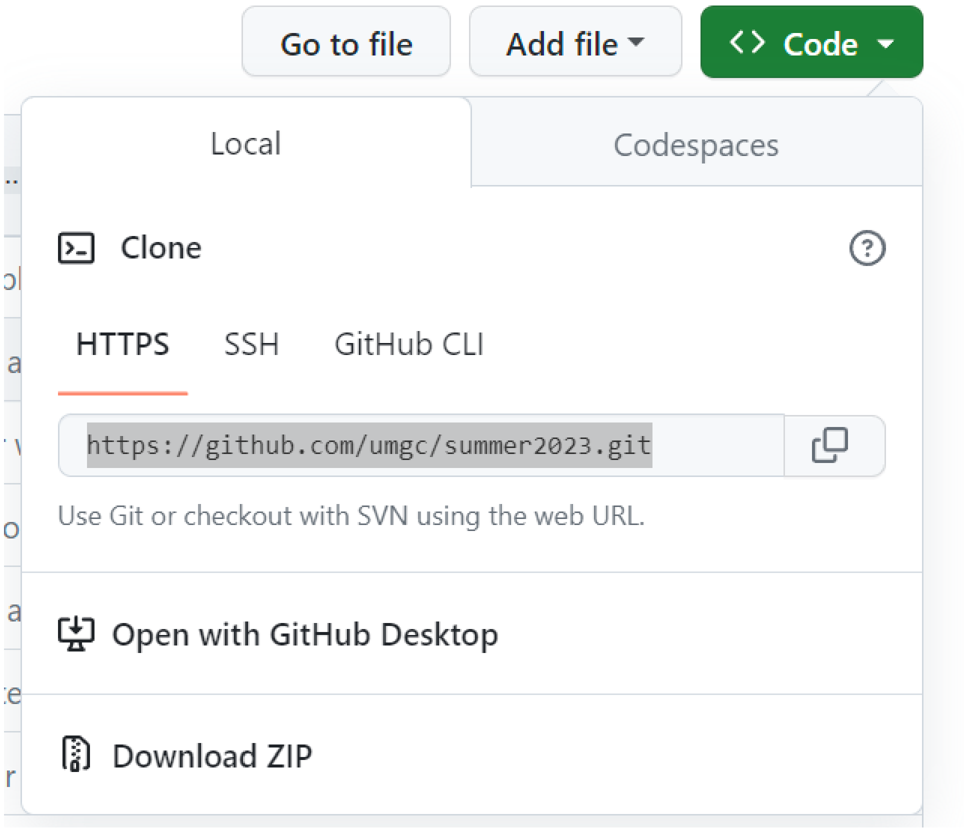
- Go the repository location at GitHub.com
- Click on the green [Code < >] button in the upper right corner

Figure 2. Code button location



- Copy the url link for the repository

Figure3. URL link for the repository



```
https://github.com/umgc/summer2023.git
```

Use Git or checkout with SVN using the web URL.

- Open up Git Bash or a terminal in your local machine
- Go to the directory you want to clone
- Type out the command 'git clone https://github.com/url/to/the/repository' and click enter (Github, n.d.b.)

Setting Up the .env File

ConvoBuddy and the backend_services package use a .env file deployed with the application to configure certain items such as AWS URLs, BESie topics, and the OpenAI API key. As the API key must be private, it cannot be pushed to github. Thus the .env file is in the .gitignore and is not to be checked in. To build and run the project with full functionality, a complete copy of the .env file is required.

The completely filled in development environment .env file with OpenAI API Key can be downloaded from Team B Files, filename **.env.txt**.

Note: Teams downloads the file as env.txt without the leading ".". Copy it to the talker-mobile-app folder and rename to .env.

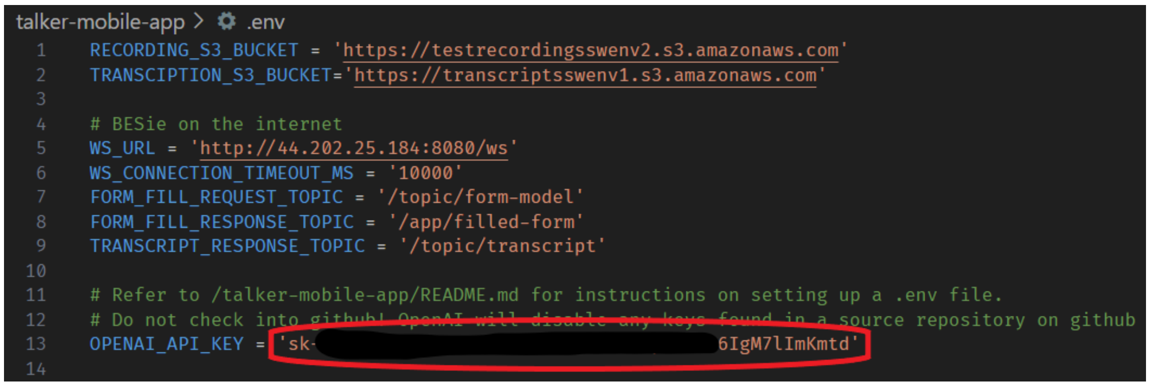
Getting an OpenAI API Key

This short video explains the process of creating a new key: https://youtu.be/EQQjdwdVQ-M?t=24

The following outlines setting up a new key and adding it to the .env created with the instructions above.

1. Go to https://openai.com and Login or Sign up for a new account.
2. Click on API
3. At top right, click on Personal and then View API Keys menu item.
4. Click "+ Create new secret key" button
5. Give the key a name relating to ConvoBuddy or SWEN670
6. The new secret key will appear in a text box with copy icon button next to it.
7. Copy the key and paste it into your .env file.

Figure 2. Populated .env File with API Key Redacted



talker-mobile-app > .env

```
RECORDING_S3_BUCKET = 'https://testrecordingsswenv2.s3.amazonaws.com'
TRANSCIPTION_S3_BUCKET='https://transcriptsswenv1.s3.amazonaws.com'

# BESie on the internet
WS_URL = 'http://44.202.25.184:8080/ws'
WS_CONNECTION_TIMEOUT_MS = '10000'
FORM_FILL_REQUEST_TOPIC = '/topic/form-model'
FORM_FILL_RESPONSE_TOPIC = '/app/filled-form'
TRANSCRIPT_RESPONSE_TOPIC = '/topic/transcript'

# Refer to /talker-mobile-app/README.md for instructions on setting up a .env file.
# Do not check into github! OpenAI will disable any keys found in a source repository on github
OPENAI_API_KEY = 'sk-[redacted]6IgM7lImKmtd'

```

8. Save the key somewhere like OneNote where it will be safe. It cannot be retrieved once the dialog is closed. Additional keys can be made, however.
9. Click done.

A new account should have $5 in credits to start with, which is ample for testing with the ChatGPT model 3.5 ConvoBuddy uses.

Deployment, Testing, and Running

Testing

The following section describes the setup and installation of the test environment for a ConvoBuddy application tester.

Installing on an Android Emulator on Windows

If you already have an emulator setup, skip the next section and go to Installing ConvoBuddy on Android Emulator.

Setting up an Android Emulator

You will need Android Studio installed to create an emulated Android device.

Install Android Studio

1. Download and install Android Studio.
2. Start Android Studio and go through the 'Android Studio Setup Wizard'. This installs the latest Android SDK, Android SDK Command-line Tools, and Android SDK Build-Tools, which are required by Flutter when developing for Android.
3. Run flutter doctor to confirm that Flutter has located your installation of Android Studio. If Flutter cannot locate it, run flutter config –android-studio-dir=<directory> to set the directory that Android Studio is installed to.

(Flutter.dev, n.d.)

Optionally enable Hardware Acceleration

You have the option to enable hardware acceleration to increase performance of the Android Emulator, with the caveat that you may run into hardware incompatibility or graphics issues. You can read me and find instructions at https://developer.android.com/studio/run/emulator-acceleration#accel-vm.

Set up an Emulator

1. Launch **Android Studio**, click the **Device Manager** icon, and select **Create Device** under **Virtual** tab…
   - In older versions of Android Studio, you should instead launch **Android Studio > Tools > Android > AVD Manager** and select **Create Virtual Device…**. (The **Android** submenu is only present when inside an Android project.)
   - If you do not have a project open, you can choose **3-Dot Menu / More Actions > Virtual Device Manager** and select **Create Device…**
2. Choose a device definition and select **Next**.
3. Select one or more system images for the Android versions you want to emulate and select **Next**. An *x86* or *x86_64* image is recommended.
4. Under Emulated Performance, select **Hardware – GLES 2.0** to enable hardware acceleration.

5. Verify the AVD configuration is correct and select **Finish**.

For details on the above steps, see Managing AVDs.

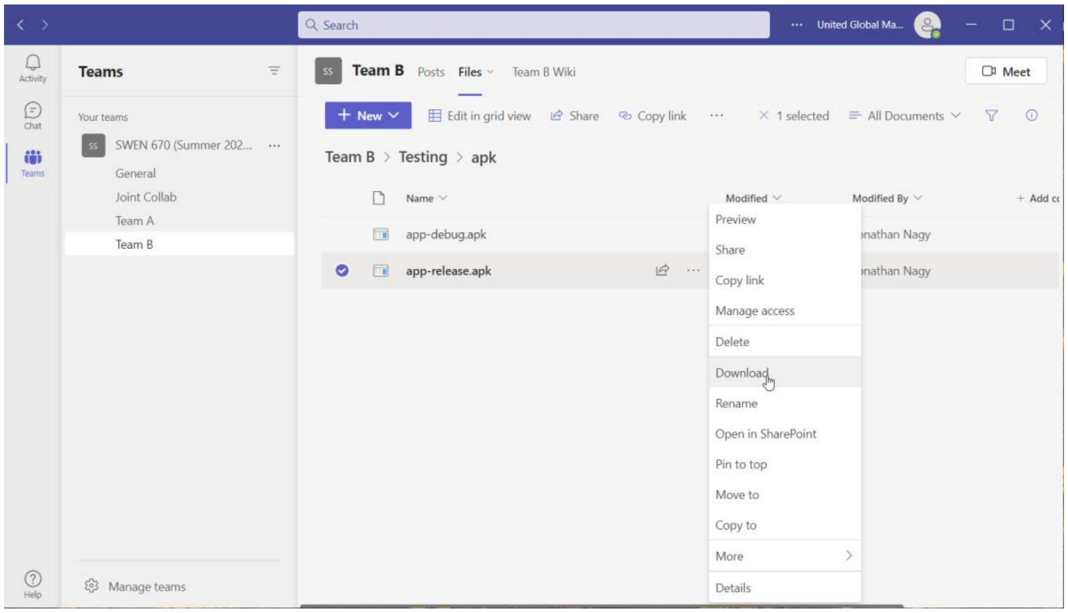
6. In Android Virtual Device Manager, click **Run** in the toolbar. The emulator starts up and displays the default canvas for your selected OS version and device.

(Flutter.dev, n.d.)

Installing ConvoBuddy APK on Android Emulator

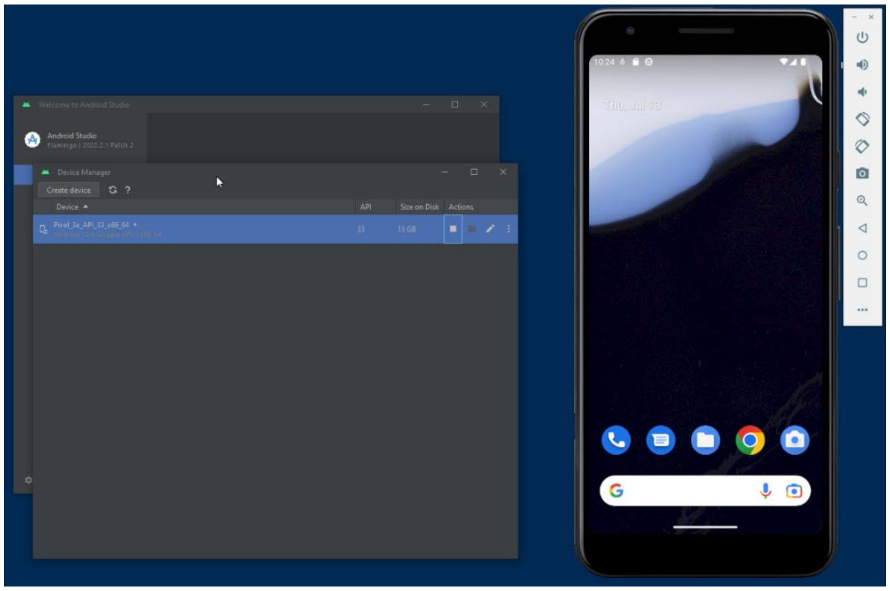
1. To install the apk on an emulator, simply download the app-release.apk or app-debug.apk from Team B channels Files tab at “Team B > Testing > apk” folder. If you need access to the app documents folder for testing, use the app-debug version, as the release version restricts access to app documents.

Figure 3. Download ConvoBuddy APK



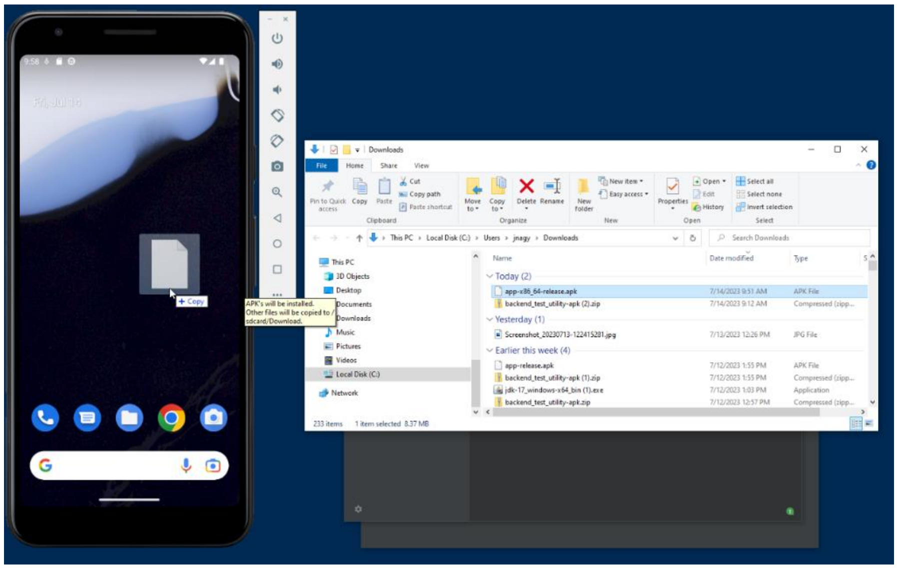
2. Start up your android emulator from Android Studio or VS code.

Figure 4. Android Emulator Started



3. If you have a previous version of ConvoBuddy installed, uninstall it first to avoid an incompatible version error. To install, drag and drop the downloaded app from your Downloads folder to the emulator.

Figure 5. Install ConvoBuddy APK

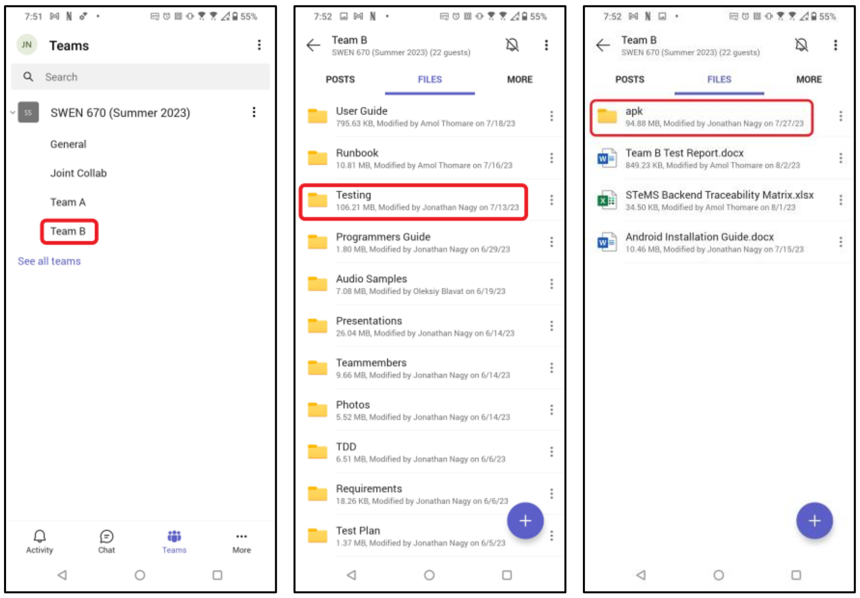


Installing on an Android Physical Device

Downloading and Installing the ConvoBuddy APK

1. Open Teams and go to Team B channel and then Files tab. From there open the Testing folder, and then the apk folder.

Figure 6. ConvoBuddy APK Location



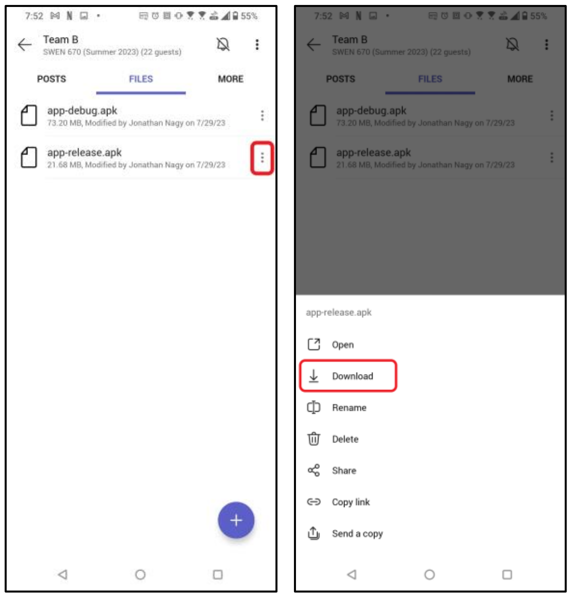
Teams
Search
SWEN 670 (Summer 2023)
General
Joint Collab
Team A
Team B
See all teams
Activity Chat Teams More

Team B
SWEN 670 (Summer 2023) (22 guests)
POSTS FILES MORE
User Guide
795.63 KB, Modified by Amol Thomare on 7/18/23
Runbook
10.81 MB, Modified by Amol Thomare on 7/16/23
Testing
106.21 MB, Modified by Jonathan Nagy on 7/13/23
Programmers Guide
1.80 MB, Modified by Jonathan Nagy on 6/29/23
Audio Samples
7.08 MB, Modified by Oleksiy Blavat on 6/19/23
Presentations
26.04 MB, Modified by Jonathan Nagy on 6/14/23
Teammembers
9.66 MB, Modified by Jonathan Nagy on 6/14/23
Photos
5.52 MB, Modified by Jonathan Nagy on 6/14/23
TDD
6.51 MB, Modified by Jonathan Nagy on 6/6/23
Requirements
18.26 KB, Modified by Jonathan Nagy on 6/6/23
Test Plan
1.37 MB, Modified by Jonathan Nagy on 6/5/23

Team B
SWEN 670 (Summer 2023) (22 guests)
POSTS FILES MORE
apk
94.88 MB, Modified by Jonathan Nagy on 7/27/23
Team B Test Report.docx
849.23 KB, Modified by Amol Thomare on 8/2/23
STeMS Backend Traceability Matrix.xlsx
34.50 KB, Modified by Amol Thomare on 8/1/23
Android Installation Guide.docx
10.46 MB, Modified by Jonathan Nagy on 7/15/23

2. Tap the triple dot action button to the right of the APK version you want to install. On the triple dot menu, tap Download.

Figure 7. Download ConvoBuddy APK

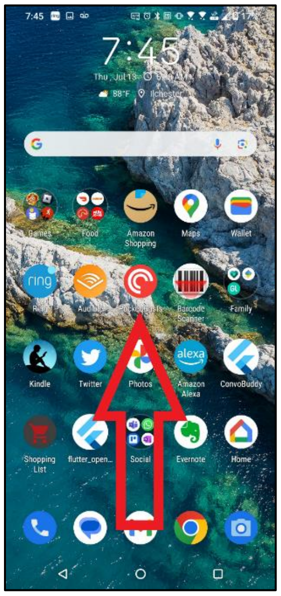


Team B
SWEN 670 (Summer 2023) (22 guests)
POSTS FILES MORE
app-debug.apk
73.20 MB, Modified by Jonathan Nagy on 7/29/23
app-release.apk
21.68 MB, Modified by Jonathan Nagy on 7/29/23

app-release.apk
Open
Download
Rename
Delete
Share
Copy link
Send a copy

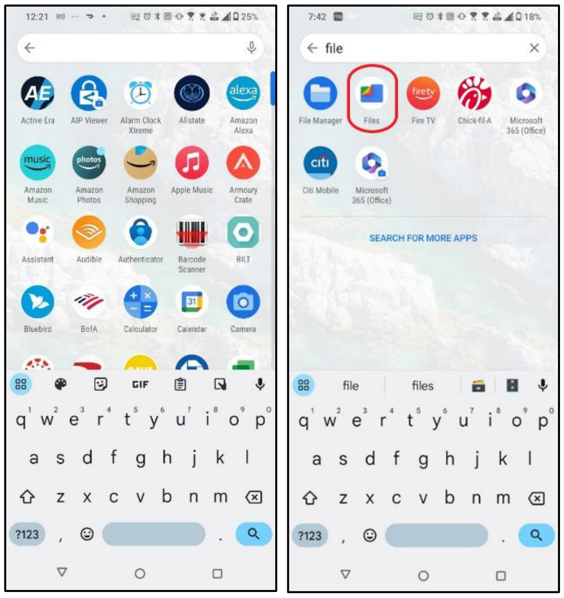
3. Swipe up from the app tray at the bottom of the home screen.

Figure 8. Swipe Up for App Drawer



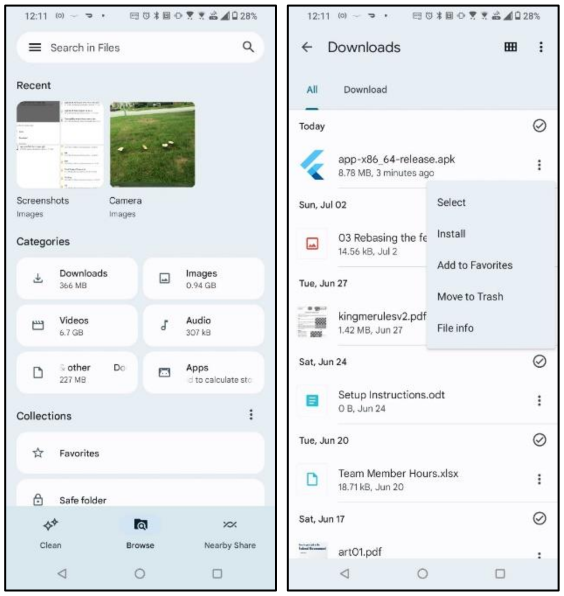
4. Search for the file app (install from play store if you don't have it). Open the files app.

Figure 9. Open Files App



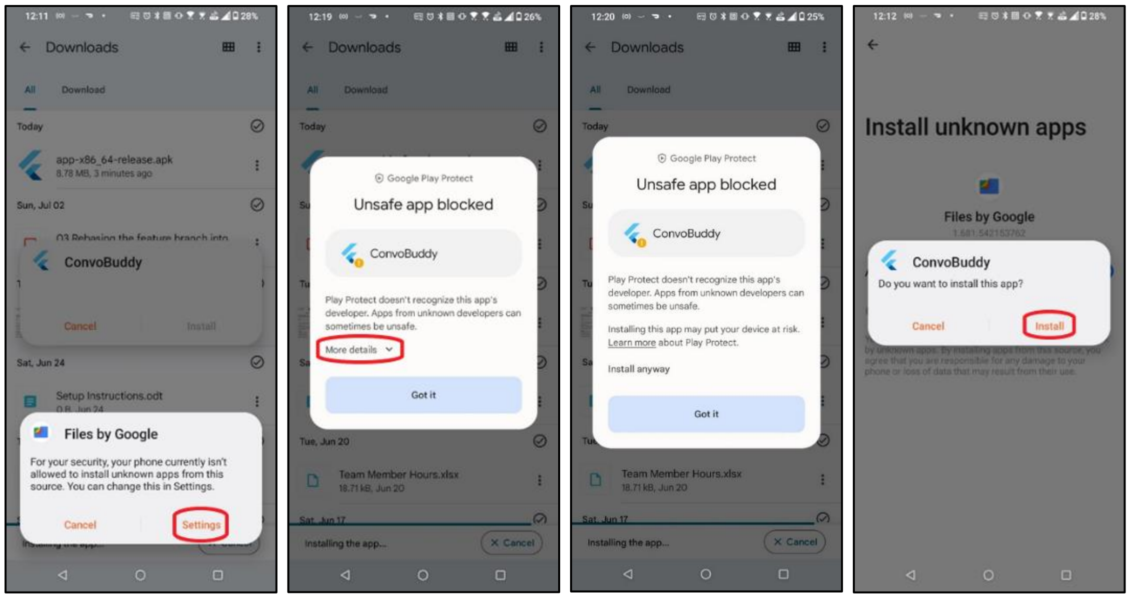
5. From the Files app, open the Downloads folder. Click the triple dot action button next to the download apk and select install.

Figure 10. Install ConvoBuddy APK



6. Most likely you will get a warning about unknown apps. Tap on settings. You will be shown an Unsafe app block dialog. Tap on More details to expand the warning details. Tap on install anyway and then tap install on the ConvoBuddy installation dialog.

Figure 11. Confirm ConvoBuddy Installation


Files by Google

For your security, your phone currently isn't allowed to install unknown apps from this source. You can change this in Settings.

Google Play Protect

Unsafe app blocked

Play Protect doesn't recognize this app's developer. Apps from unknown developers can sometimes be unsafe.

Installing this app may put your device at risk. Learn more about Play Protect.

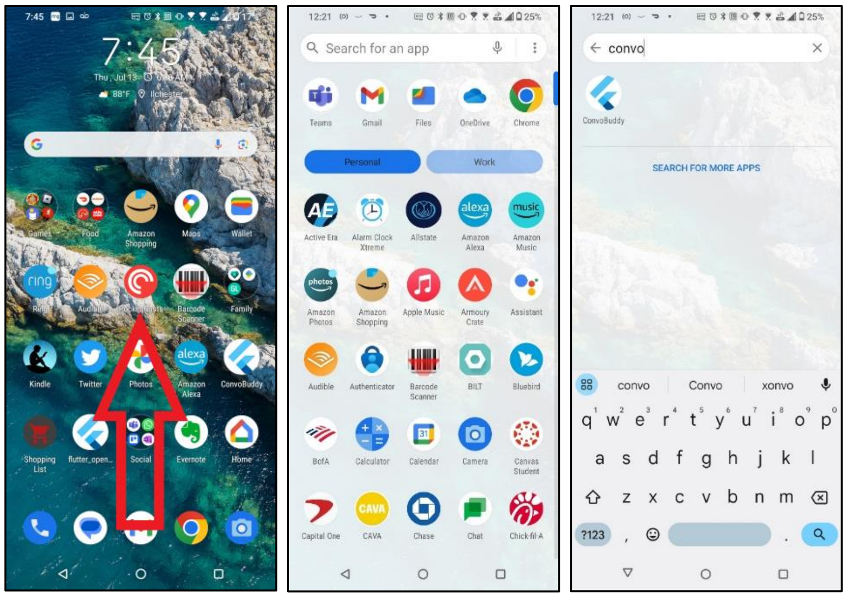
Install unknown apps

ConvoBuddy

Do you want to install this app?

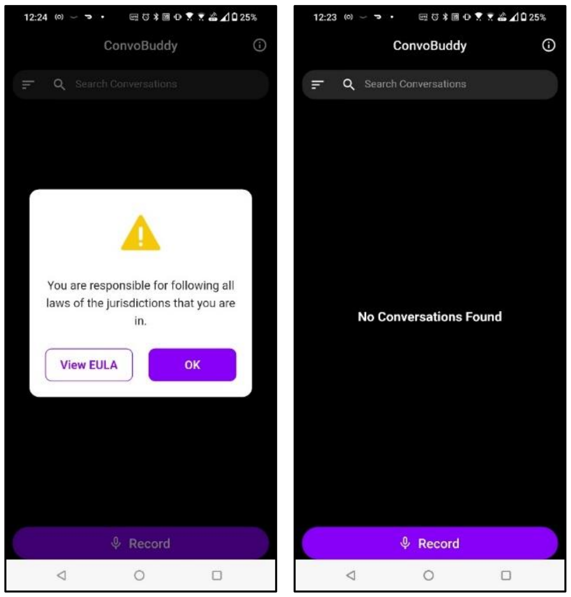
7. Swipe up again from the home screen and search for the ConvoBuddy app.

Figure 12. Open ConvoBuddy App



8. Open the ConvoBuddy app and accept the EULA.

Figure 13. Accept ConvoBuddy EULA


ConvoBuddy
Search Conversations
You are responsible for following all laws of the jurisdictions that you are in.
View EULA
OK
No Conversations Found
Record

Deployment

The following section describes the deployment of applications and services for the SteMS backend services.

Deploying BESie Service to Development/Test

Please refer to the Team A Deployment Guide for instructions on deploying to developer machines and the test environment.

Deploying BESie Service to Production

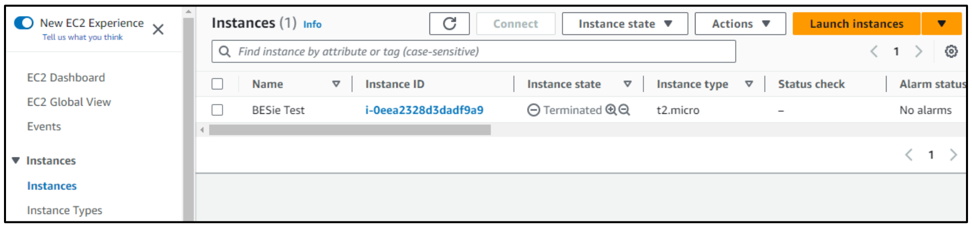
To deploy BESie to production, AWS was chosen due to ease of use and availability of limit 12-month free trail periods. The deployment of BESie may be accomplished in two ways, either building a AWS Elastic Cloud Compute (EC2) machine instance from an Amazon Linux Amazon Machine Image (AMI) or deploying a saved AMI with BESie already installed and configured.

Create EC2 Instance from Amazon Linux AMI

This section will describe the creation of a BESie server from a clean Amazon Linux AMI.

1. Go to AWS Console and login (https://aws.amazon.com)
2. Navigate to the EC2 service
3. In Instances, click Launce instances button

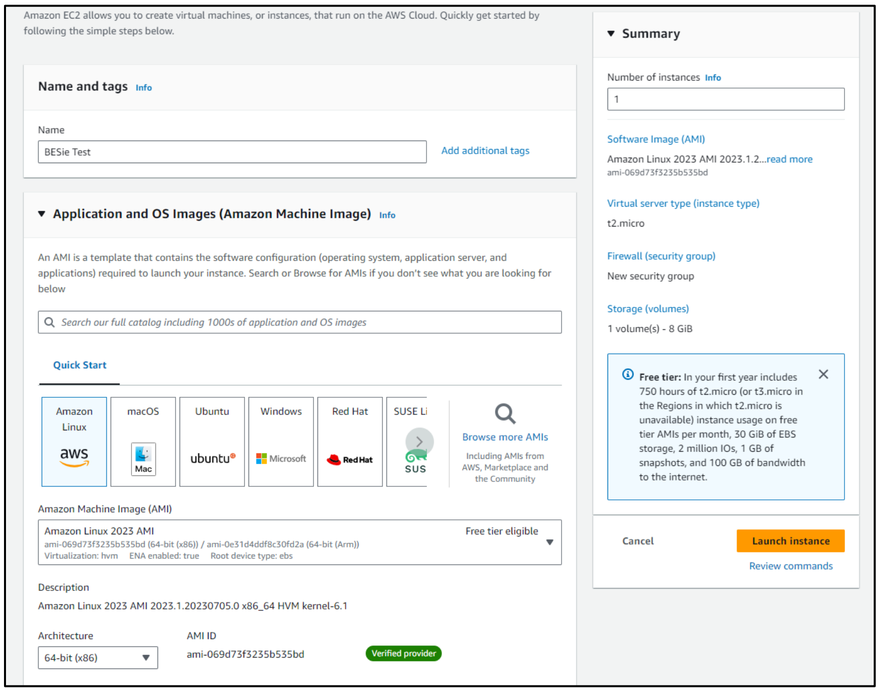
Figure 14. AWS Instances screen



	Name	Instance ID	Instance state	Instance type	Status check	Alarm status
	BESie Test	i-0eea2328d3dadf9a9	Terminated	t2.micro	–	No alarms

4. Pick Amazon Linux AMI

Figure 15. Create Amazon Linux Instance


Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags

Name: BESie Test

Application and OS Images (Amazon Machine Image)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Add additional tags
Search our full catalog including 1000s of application and OS images
Quick Start
Amazon Linux
macOS
Ubuntu
Windows
Red Hat
SUSE Li
Browse more AMIs
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI): Amazon Linux 2023 AMI — Free tier eligible
ami-069d73f3235b535bd (64-bit (x86)) / ami-0e31d4ddf8c30fd2a (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Description: Amazon Linux 2023 AMI 2023.1.20230705.0 x86_64 HVM kernel-6.1

Architecture: 64-bit (x86)

AMI ID: ami-069d73f3235b535bd — Verified provider

Summary

Number of instances: 1

Software Image (AMI): Amazon Linux 2023 AMI 2023.1.2...read more
ami-069d73f3235b535bd

Virtual server type (instance type): t2.micro

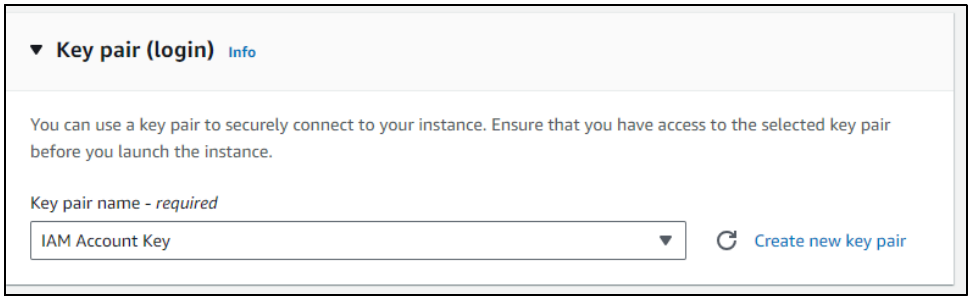
Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

5. Select an existing Key Pair or Create a new one

Figure 16. Select Key Pair



▼ **Key pair (login)** Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

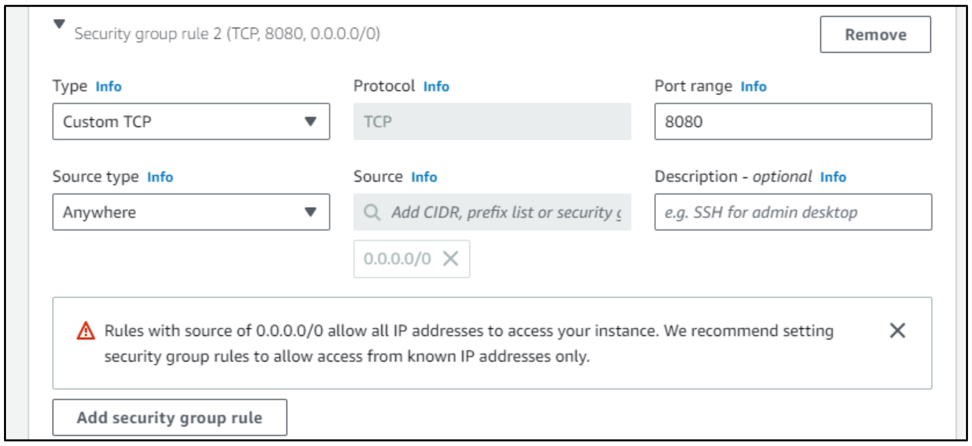
Key pair name - *required*

IAM Account Key

Create new key pair

6. Click Edit under Network Settings
7. Click "Add security group rule" and set to Type to **Custom TCP**, Port range to **8080**, Source type **Anywhere**.

Figure 17. Port 8080 Security Group Rule



Security group rule 2 (TCP, 8080, 0.0.0.0/0)

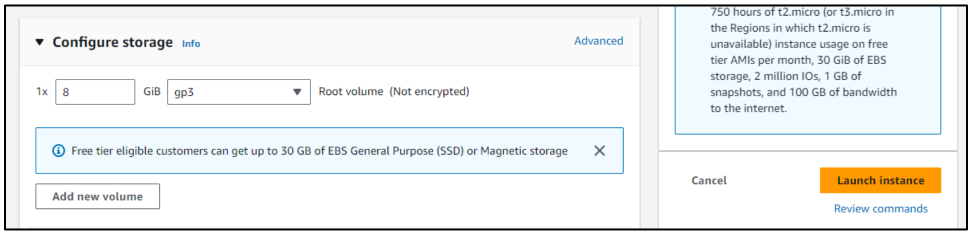
Type	Protocol	Port range
Custom TCP	TCP	8080

Source type	Source	Description - *optional*
Anywhere	Add CIDR, prefix list or security g 0.0.0.0/0	e.g. SSH for admin desktop

Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

8. Click Launch Instance

Figure 18. Launce Instance



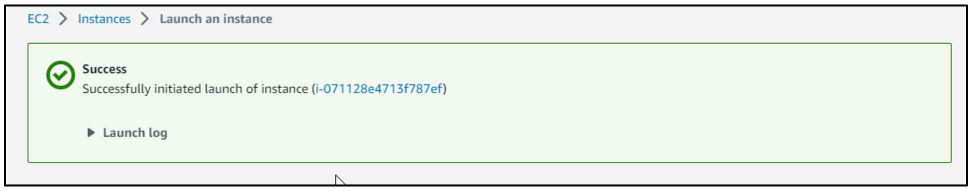
▼ **Configure storage** Info

1x 8 GiB gp3 Root volume (Not encrypted)

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage

750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Figure 19. Instance Created



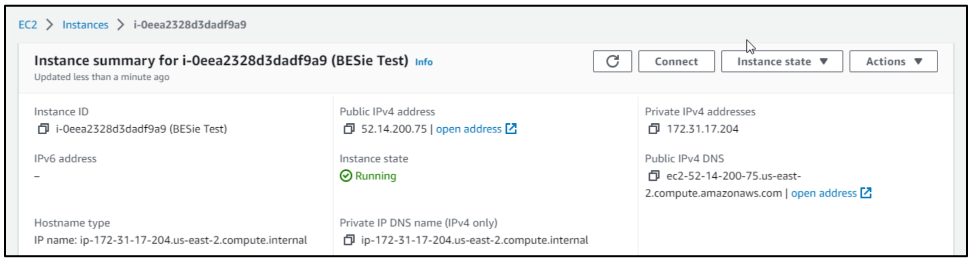
Success
Successfully initiated launch of instance (i-071128e4713f787ef)

Connect and Install Prerequisites

To make things simpler, you can connect to the new instance with AWS web console.

1. Click on the new instance on the Instances screen and then click the Connect button.

Figure 20. Connect to Instance from AWS Console

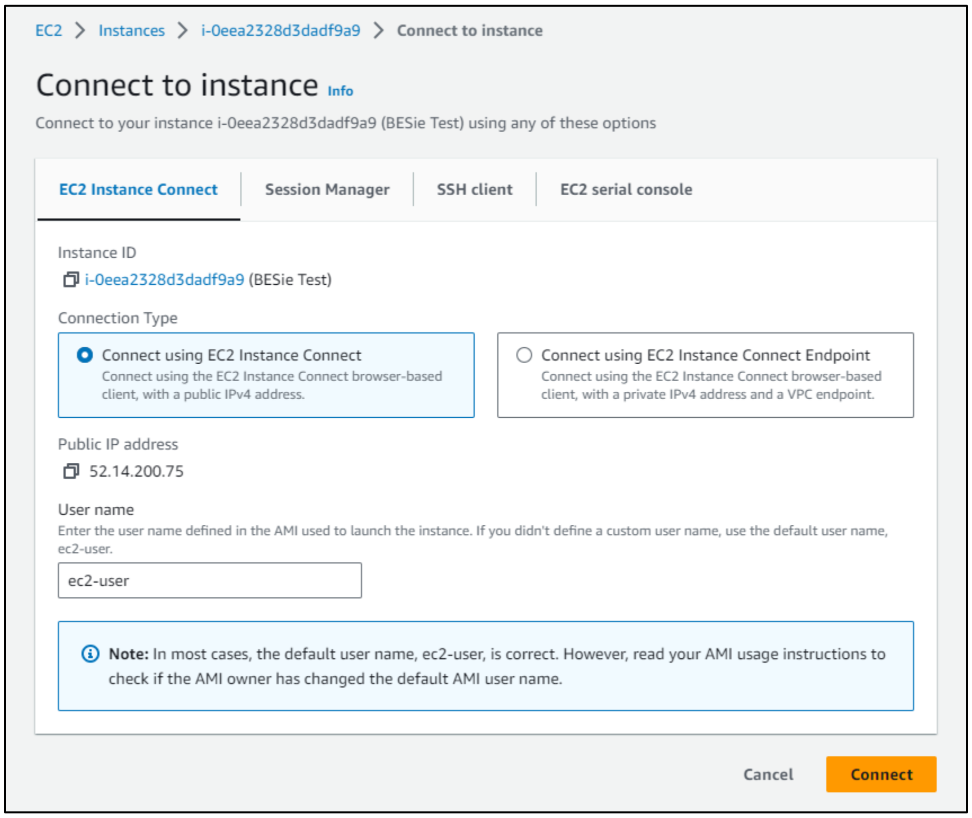


Instance summary for i-0eea2328d3dadf9a9 (BESie Test) Info
Updated less than a minute ago

Instance ID: i-0eea2328d3dadf9a9 (BESie Test)
Public IPv4 address: 52.14.200.75 | open address
Private IPv4 addresses: 172.31.17.204
IPv6 address: –
Instance state: Running
Public IPv4 DNS: ec2-52-14-200-75.us-east-2.compute.amazonaws.com | open address
Hostname type: IP name: ip-172-31-17-204.us-east-2.compute.internal
Private IP DNS name (IPv4 only): ip-172-31-17-204.us-east-2.compute.internal

2. Defaults are appropriate. Make sure ec2-user is the selected user.

Figure 21. Connect to Instance



Connect to instance Info

Connect to your instance i-0eea2328d3dadf9a9 (BESie Test) using any of these options

Instance ID: i-0eea2328d3dadf9a9 (BESie Test)

Connection Type:
[✓ Connect using EC2 Instance Connect] Connect using the EC2 Instance Connect browser-based client, with a public IPv4 address.
Connect using EC2 Instance Connect Endpoint: Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.

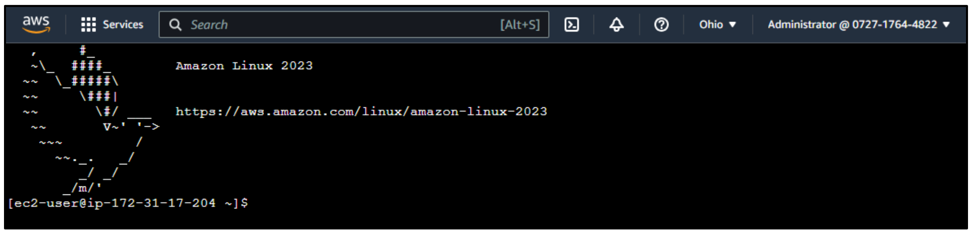
Public IP address: 52.14.200.75

User name
Enter the user name defined in the AMI used to launch the instance. If you didn't define a custom user name, use the default user name, ec2-user.
ec2-user

Note: In most cases, the default user name, ec2-user, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.

3. You should be connected and logged in automatically

Figure 22. Connected to Instance



```
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-17-204 ~]$
```

4. To install Amazon's distribution of Java Development Kit (JDK) 17, type **sudo yum –y install java-17-amazon-corretto-devel**.

The JDK version 17 is required to build BESie, and yum package manager makes it easy to install Amazon's distribution of the package.

5. Install git by typing **sudo yum –y install git.**
6. Clone the umgc/summer2023 repository by typing **git clone https://github.com/umgc/summer2023.git**.
7. Build BESie service by typing **cd summer2023/BESie/** and then **./mvnw clean package**.

This will invoke Maven java package manager to pull in all required dependencies and package a BESie executable in the form of a jar file.

8. Open the nano editor to create a 18rivil service entry by typing **sudo nano /etc/18rivil/system/besie.service**

Systemd is an effective way to turn any long running program in Linux into a service (Morel, B., 2017). Elevatated 18rivileges are required to create a file under /etc., so the sudo command is used to invoke the nano command. The besie.service naming convention allows control of the BESie service by the "besie" short name and viewing of logs by short name with journalctl.

9. Add the following contents to the new besie.service file. Update the snapshot jar version to the current version output by the build.

Figure 23. /etc/18rivil/system/besie.service File

```
[Unit]
Description=BESie Service
After=network.target
StartLimitIntervalSec=0

[Service]
Type=simple
Restart=always
RestartSec=1
User=ec2-user
ExecStart=java -jar /home/ec2-user/summer2023/BESie/target/BESie-0.0.1-SNAPSHOT.jar

[Install]
WantedBy=multi-user.target
```

10. Type **Ctrl-O** and followed by **Enter** to save and then **Ctrl-X** to exit.

The line “Restart=always” means BESie will restart if it crashes, or the system is rebooted. After and WantedBy lines determine when in the startup process the service is started, after the network components are set up and before going multi-user. The line “RestartSec=1” specifies a 1 second delay in restart attempts. The user is sent to non-privileged “ec2-user” to reduce security risks. The line “StartLimitIntervalSec=0” disables any kind of startup rate limiting (Man.archlinux.org, n.d.).

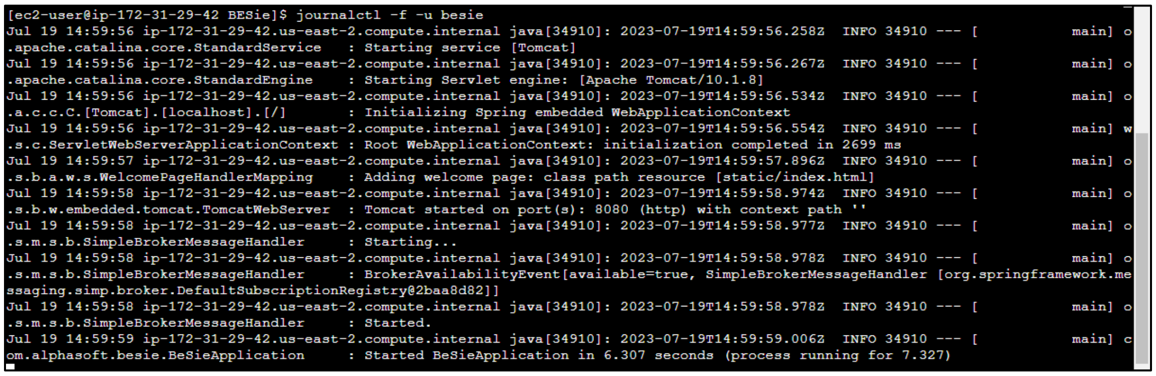
11. Enable the BESie service by typing **sudo systemctl enable besie**

Again, elevated 19rivileges are required, this time to run systemctl.

12. Start up the BESie service by typing **sudo systemctl start besie**
13. Check BESie logs by typing **journalctl –f –u besie**.

The –f flag “follows” output as it is generated and added to the logs. The “–u besie” flag specifies log output should be filter to the BESie unit, i.e. besie.

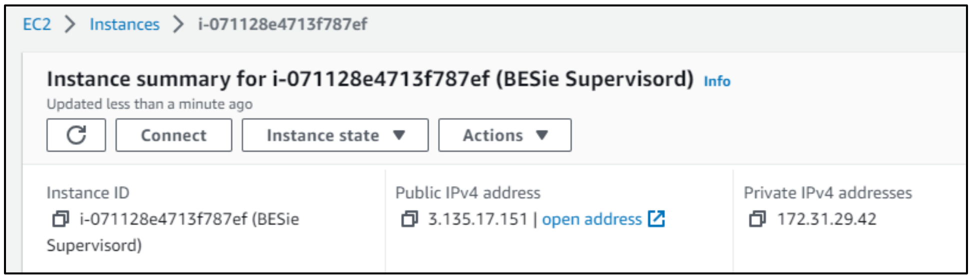
Figure 24. BESie Service Logs



```
[ec2-user@ip-172-31-29-42 BESie]$ journalctl -f -u besie
Jul 19 14:59:56 ip-172-31-29-42.us-east-2.compute.internal java[34910]: 2023-07-19T14:59:56.258Z  INFO 34910 --- [           main] o
.apache.catalina.core.StandardService    : Starting service [Tomcat]
Jul 19 14:59:56 ip-172-31-29-42.us-east-2.compute.internal java[34910]: 2023-07-19T14:59:56.267Z  INFO 34910 --- [           main] o
.apache.catalina.core.StandardEngine     : Starting Servlet engine: [Apache Tomcat/10.1.8]
Jul 19 14:59:56 ip-172-31-29-42.us-east-2.compute.internal java[34910]: 2023-07-19T14:59:56.534Z  INFO 34910 --- [           main] o
.a.c.c.C.[Tomcat].[localhost].[/]        : Initializing Spring embedded WebApplicationContext
Jul 19 14:59:56 ip-172-31-29-42.us-east-2.compute.internal java[34910]: 2023-07-19T14:59:56.554Z  INFO 34910 --- [           main] w
.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 2699 ms
Jul 19 14:59:57 ip-172-31-29-42.us-east-2.compute.internal java[34910]: 2023-07-19T14:59:57.896Z  INFO 34910 --- [           main] o
.s.b.a.w.s.WelcomePageHandlerMapping     : Adding welcome page: class path resource [static/index.html]
Jul 19 14:59:58 ip-172-31-29-42.us-east-2.compute.internal java[34910]: 2023-07-19T14:59:58.974Z  INFO 34910 --- [           main] o
.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port(s): 8080 (http) with context path ''
Jul 19 14:59:58 ip-172-31-29-42.us-east-2.compute.internal java[34910]: 2023-07-19T14:59:58.977Z  INFO 34910 --- [           main] o
.s.m.s.b.SimpleBrokerMessageHandler      : Starting...
Jul 19 14:59:58 ip-172-31-29-42.us-east-2.compute.internal java[34910]: 2023-07-19T14:59:58.978Z  INFO 34910 --- [           main] o
.s.m.s.b.SimpleBrokerMessageHandler      : BrokerAvailabilityEvent[available=true, SimpleBrokerMessageHandler [org.springframework.me
ssaging.simp.broker.DefaultSubscriptionRegistry@2baa8d82]]
Jul 19 14:59:58 ip-172-31-29-42.us-east-2.compute.internal java[34910]: 2023-07-19T14:59:58.978Z  INFO 34910 --- [           main] o
.s.m.s.b.SimpleBrokerMessageHandler      : Started.
Jul 19 14:59:59 ip-172-31-29-42.us-east-2.compute.internal java[34910]: 2023-07-19T14:59:59.006Z  INFO 34910 --- [           main] c
om.alphasoft.besie.BeSieApplication      : Started BeSieApplication in 6.307 seconds (process running for 7.327)
```

14. Test BESie by entering “http://<PublicIP>:8080/” in a browser, where Public IP is taken from the EC2 Instance information.

Figure 25. EC2 Instance Public IP



Instance summary for i-071128e4713f787ef (BESie Supervisord)

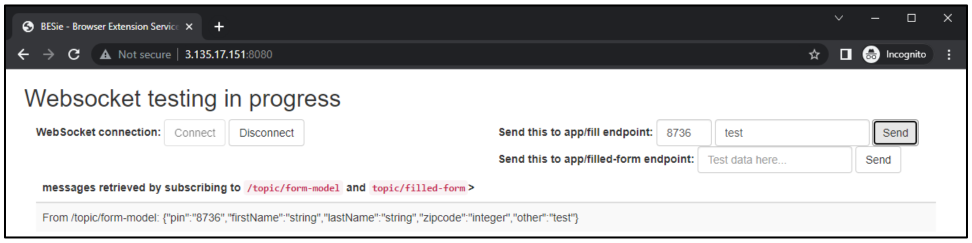
Updated less than a minute ago

Instance ID: i-071128e4713f787ef (BESie Supervisord)

Public IPv4 address: 3.135.17.151

Private IPv4 addresses: 172.31.29.42

Figure 26. BESie Test Page



Websocket testing in progress

WebSocket connection: Connect Disconnect

Send this to app/fill endpoint: 8736 test Send

Send this to app/filled-form endpoint: Test data here... Send

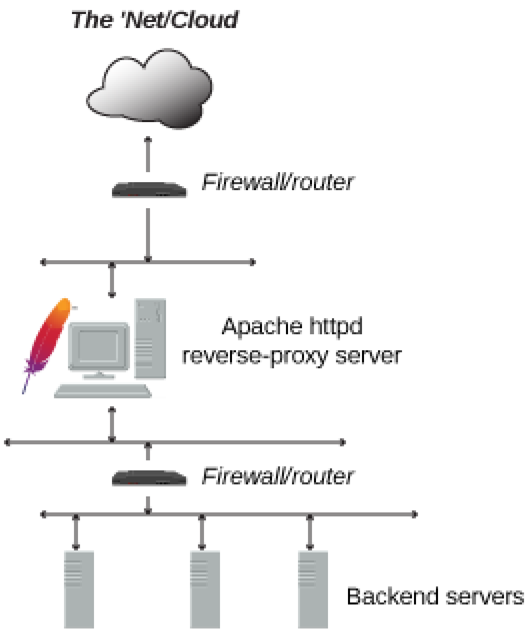
messages retrieved by subscribing to `/topic/form-model` **and** `topic/filled-form` **>**

```
From /topic/form-model: {"pin":"8736","firstName":"string","lastName":"string","zipcode":"integer","other":"test"}
```

Setting up SSL/TLS

To protect user information in the form of recordings and transcripts, BESie should be put behind a Secure Socket Layer/Transport Layer Security (SSL/TLS) protected web server. SSL is protocol for creating a secure connection between two devices or applications on a network, and TLS is an upgraded version of SSL that fixes security vulnerabilities (Amazon, n.d.d). The AWS instance will be configured such that BESie is only visible to processes running on the machine or the instance's virtual network and will be exposed to the outside through the Apache web server configured as reverse proxy for port 8080 on localhost.

Figure 27. Generic Apache Reverse Proxy Setup



(Apache, 2023a)

Apache version 2.4.47 and higher feature the Protocol Upgrade option to the mod_proxy module. This option allows support for end-to-end encryption of websockets using tunnelling (Apache, 2023b).

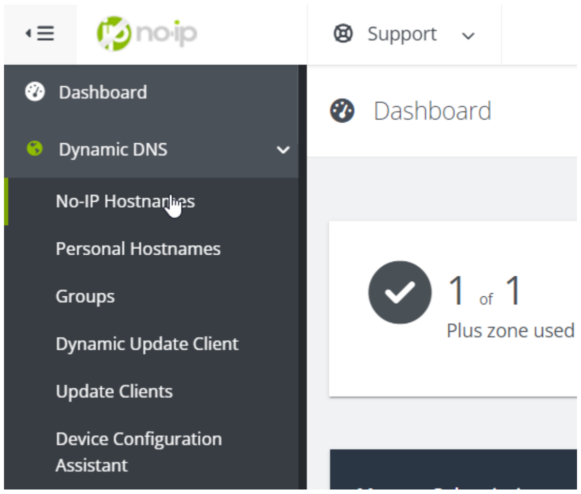
To enable SSL/TLS encryption on the Apache server, we must obtain a domain name, configure it for our AWS instance's public IP address, and obtain an SSL certificate from a certificate authority (CA).

Obtaining a Domain Name

For a domain name, the free No-IP service was used. Other services could be used, and a paid service would provide more options as far as domain name and configuration. To configure a domain name:

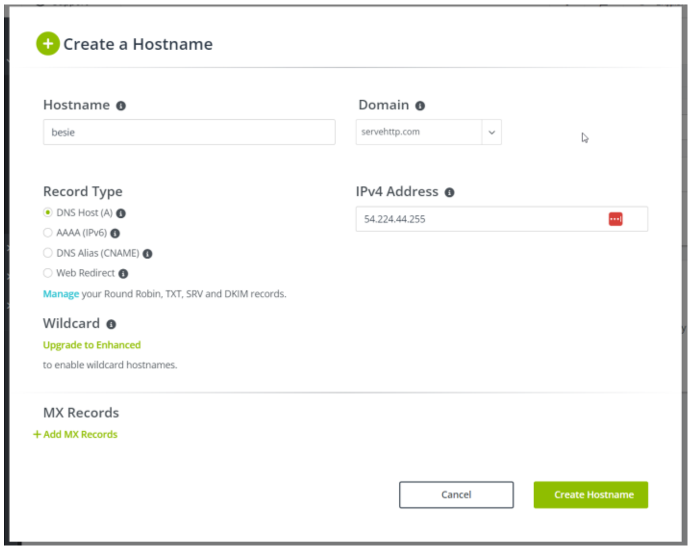
1. Go to https://www.noip.com/
2. Sign up with an account
3. Open menu item Dynamic DNS and then click on NO-IP Hostnames.

Figure 28. No-IP Menu



4. Click Create Hostname
5. Enter the hostname, such as “besie.servehttp.com”, and public IP address from the BESie instance information from the AWS console.

Figure 29. No-IP Create a Hostname Dialog



Create a Hostname

Hostname: besie

Domain: servehttp.com

Record Type
- (●) DNS Host (A)
- ( ) AAAA (IPv6)
- ( ) DNS Alias (CNAME)
- ( ) Web Redirect

Manage your Round Robin, TXT, SRV and DKIM records.

IPv4 Address: 54.224.44.255

Wildcard
Upgrade to Enhanced
to enable wildcard hostnames.

MX Records
+ Add MX Records

Cancel | Create Hostname

6. Click Create Hostname and you should see a new Hostname entry in the No-IP dashboard.

The hostname should propagate it may take 24-48 hours for the new hostname to propagate to all of the root DNS servers (noip.com, n.d.). However, in most cases DNS updates only take a few hours to propagate (godaddy.com, n.d.), and in the initial setup done for this guide the domain was returning from DNS lookup within 10 minutes.

Configuring SSL/TLS and Obtaining a Certificate

Another free service that was used to configure SSL/TLS is Let's Encrypt, a non-profit certificate authority. The CertBot application is used to configure the web server using a Let's Encrypt certificate. Before we can use CertBot, we will need to configure a virtual host on port 80.

1. Connect to the BESie AWS instance command line as before from the AWS instance page.
2. Type the following commands in succession, ensuring no errors are emitted:

   ```
   sudo dnf install -y httpd
   sudo systemctl start httpd
   sudo systemctl enable httpd
   ```

3. Type the following command to create a new configuration file:

   ```
   sudo nano /etc/httpd/conf.d/besie.conf
   ```

4. Enter the following text into the editor and update the ServerName to match the domain name created above. When done, type `Ctrl-O` followed by `Enter` and `Ctrl-X` when done to save and exit.

   ```
   <VirtualHost *:80>
       ServerName besie.servehttp.com
       DocumentRoot /var/www/html
       ErrorLog logs/error.log
       CustomLog logs/requests.log combined
   </VirtualHost>
   ```

5. Enable SSL/TLS by entering the following:

   ```
   sudo dnf install -y mod_ssl
   ```

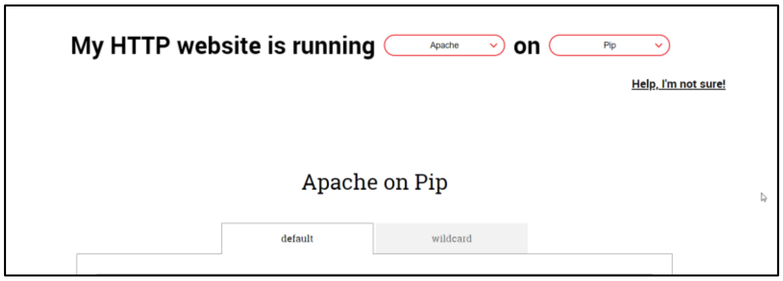
6. Delete the ssl.conf file provided by the mod_ssl package. CertBot will provide a new ssl configuration file with best-practice settings for mod_ssl.

   ```
   sudo rm /etc/httpd/conf.d/ssl.conf
   ```

7. Restart the web server with the command "`sudo systemctl restart httpd`". Assuming there is no error with the new configuration, continue to configuring the server with CertBot.

8. Go to https://certbot.eff.org/
9. Next to "My HTTP website is running", select Apache and then next to "on" select Pip. The webpage should update with Apache on Pip header followed by instructions.

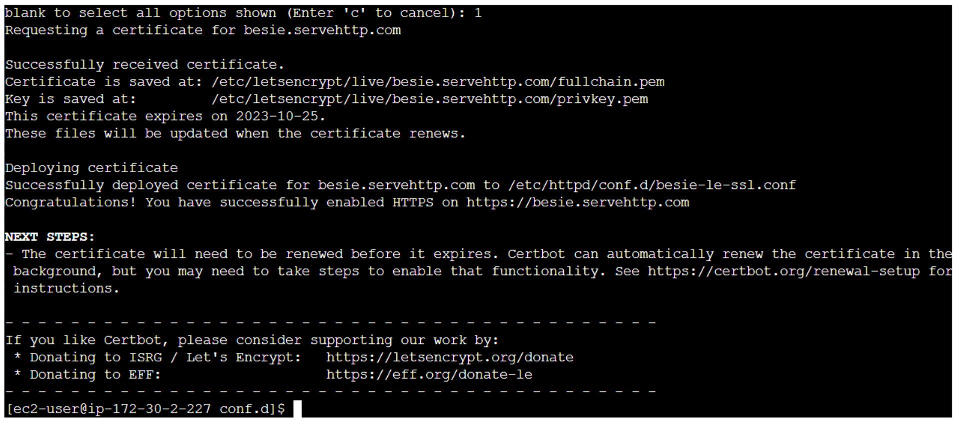
Figure 30. Certbot Instructions Selection



10. Connect to the BESie AWS instance command line as before from the AWS instance page.
11. Follow the instructions, using "dnf" version of the package manager commands.
12. After running "`sudo certbot --apache`", you will be prompted for several things:
    a. Enter the adminstrator's email address who will be responsible for BESie.
    b. Agree to terms and sharing the email with EFF and "Y" to terms and "Y" or "N" to share the email with EFF.
    c. CertBot should detect the virtual host we created above on port 80 as the first and only option. If so, type 1 to select the hostname.

You should see output like the following:

Figure 31. Certbot Configured



```
blank to select all options shown (Enter 'c' to cancel): 1
Requesting a certificate for besie.servehttp.com

Successfully received certificate.
Certificate is saved at: /etc/letsencrypt/live/besie.servehttp.com/fullchain.pem
Key is saved at:         /etc/letsencrypt/live/besie.servehttp.com/privkey.pem
This certificate expires on 2023-10-25.
These files will be updated when the certificate renews.

Deploying certificate
Successfully deployed certificate for besie.servehttp.com to /etc/httpd/conf.d/besie-le-ssl.conf
Congratulations! You have successfully enabled HTTPS on https://besie.servehttp.com

NEXT STEPS:
- The certificate will need to be renewed before it expires. Certbot can automatically renew the certificate in the
 background, but you may need to take steps to enable that functionality. See https://certbot.org/renewal-setup for
 instructions.

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
If you like Certbot, please consider supporting our work by:
 * Donating to ISRG / Let's Encrypt:   https://letsencrypt.org/donate
 * Donating to EFF:                    https://eff.org/donate-le
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
[ec2-user@ip-172-30-2-227 conf.d]$
```

13. Next we will configure the reverse proxy to point to BESie on localhost. Type:

```
sudo nano /etc/httpd/conf.d/besie-le-ssl.conf
```

And in the editor add the following highlighted portion and then type `Ctrl-O` followed by `Enter` and `Ctrl-X` when done to save and exit.

```
<IfModule mod_ssl.c>
<VirtualHost *:443>
    ServerName besie.servehttp.com
    DocumentRoot /var/www/html
    ErrorLog logs/error.log
    CustomLog logs/requests.log combined
```

```
  SSLCertificateFile /etc/letsencrypt/live/besie.servehttp.com/fullchain.pem
  SSLCertificateKeyFile /etc/letsencrypt/live/besie.servehttp.com/privkey.pem
  Include /etc/letsencrypt/options-ssl-apache.conf

  ProxyPass / http://localhost:8080/ upgrade=websocket
  ProxyPassReverse / http://localhost:8080/

</VirtualHost>
</IfModule>
```

The ProxyPass and ProxyPassReverse lines are standard for configuring a reverse proxy with a backend service running at port 8080 on localhost. The “upgrade=websocket” property configures an end-to-end websocket workflow built into the mod_proxy module as of Apache version 2.4.47 (Apache, 2023b). The instance created for this guide was Apache 2.4.56 for Amazon Linux.

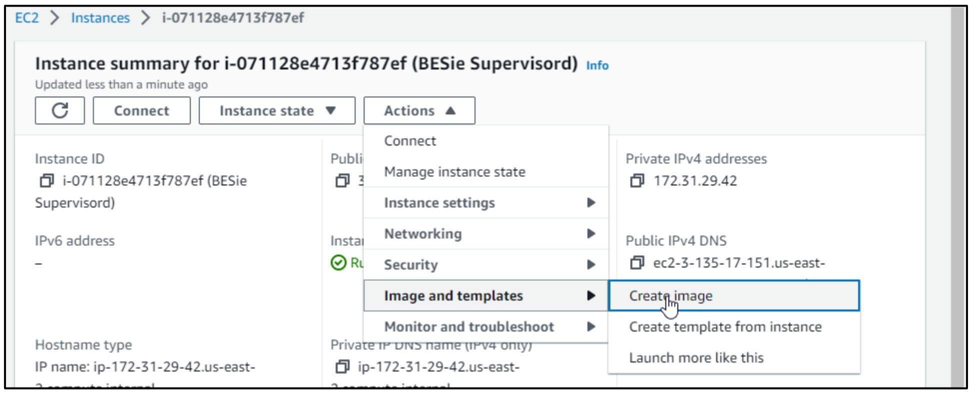
Authentication headers along with a secret key for BESie would be ideal, but this was not implemented due to time constraints.

Save BESie Service AMI

This section will describe how you can save the Amazon machine instance created above as an Amazon Machine Image (AMI) for easy deployment later. Note, the AMI can be created whether the virtual machine is running or stopped.

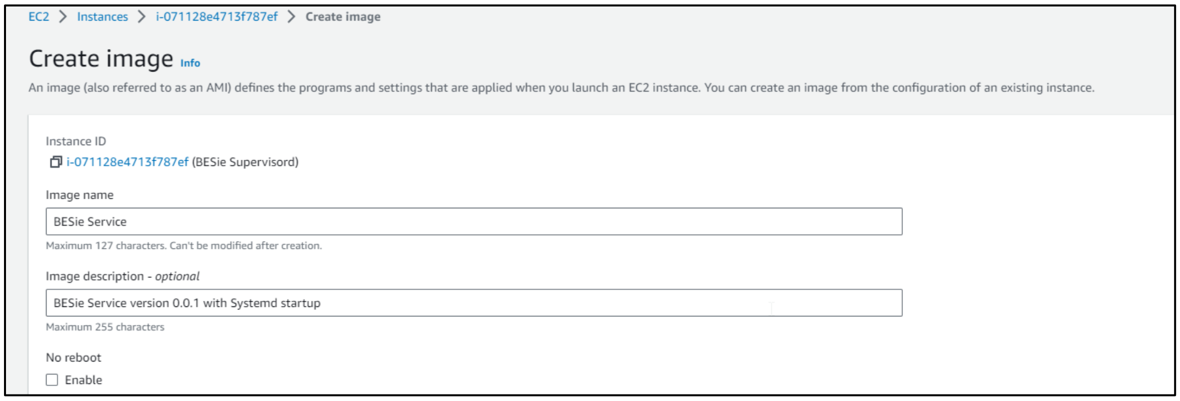
1. On the BESie EC2 Instance defails screen, click “Actions” then hover over “Images and templates“ and then click “Create Image”.

Figure 32. Create Image Command



2. Enter an Image name and Image description

Figure 33. Create Image Screen



EC2 > Instances > i-071128e4713f787ef > Create image

Create image Info

An image (also referred to as an AMI) defines the programs and settings that are applied when you launch an EC2 instance. You can create an image from the configuration of an existing instance.

Instance ID
i-071128e4713f787ef (BESie Supervisord)

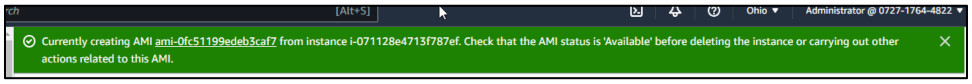
Image name
BESie Service
Maximum 127 characters. Can't be modified after creation.

Image description - *optional*
BESie Service version 0.0.1 with Systemd startup
Maximum 255 characters

No reboot
☐ Enable

3. Click the "Create Image" button at the bottom. A message stating "Currently creating AMI" should be shown.

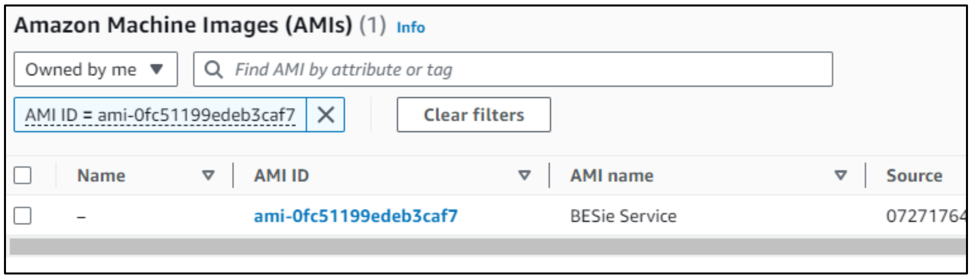
Figure 34. Image Created Message



Currently creating AMI ami-0fc51199edeb3caf7 from instance i-071128e4713f787ef. Check that the AMI status is 'Available' before deleting the instance or carrying out other actions related to this AMI.

4. The AMI will be pending for a bit. When the Status is no longer "pending", but instead "available", proceed.

Figure 35. AMIs

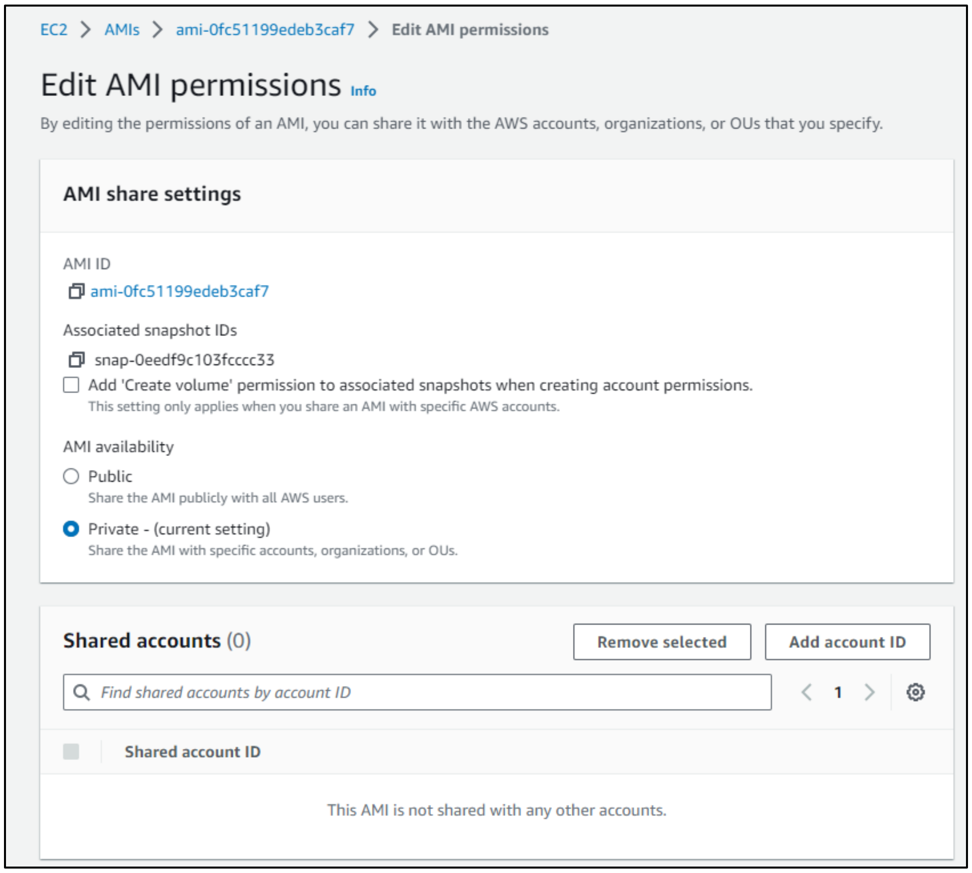


Amazon Machine Images (AMIs) (1) Info

	Name	AMI ID	AMI name	Source
☐	–	ami-0fc51199edeb3caf7	BESie Service	07271764

5. On the AMI details, under the Permissions tab click the "Edit AMI permissions" button. From there you can set it to public or search for AWS accounts to share the image with.

Figure 36. AMI Permissions


EC2 > AMIs > ami-0fc51199edeb3caf7 > Edit AMI permissions
Info
Remove selected
Add account ID
Find shared accounts by account ID
1

Edit AMI permissions

By editing the permissions of an AMI, you can share it with the AWS accounts, organizations, or OUs that you specify.

AMI share settings

AMI ID
ami-0fc51199edeb3caf7

Associated snapshot IDs
snap-0eedf9c103fcccc33
☐ Add 'Create volume' permission to associated snapshots when creating account permissions.
This setting only applies when you share an AMI with specific AWS accounts.

AMI availability
○ Public
Share the AMI publicly with all AWS users.
● Private - (current setting)
Share the AMI with specific accounts, organizations, or OUs.

Shared accounts (0)

Shared account ID

This AMI is not shared with any other accounts.

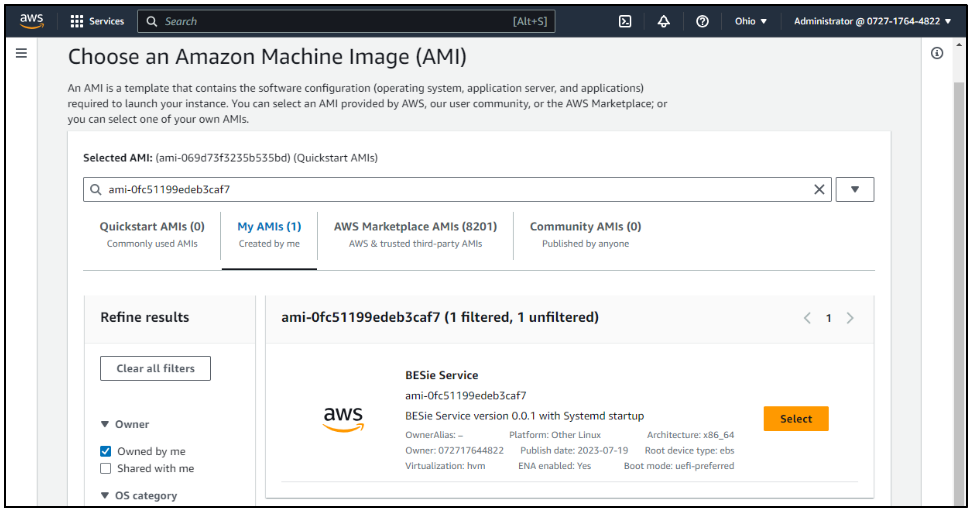
Create EC2 Instance from Saved BESie Service AMI

AMIs are created at a region level, and so when attempting to create an instance ensure you are in the same region as the AMI was originally created. The AMI created in this guide for example was created in the us-east-2 region. Additionally, AMIs may take a while to appear in the Marketplace but may appear more quickly under My AMIs on the Create instance screen. More information on sharing AMIs can be found in the User Guide for Linux Instances (Amazon, n.d.a).

To create a new instance from a previously created BESie Service AMI:

1. From Instances in EC2, click the "Launch instances" button.
2. Enter the AMI created previously in the Search box. For example, "ami-0fc51199edeb3caf7" was created for this guide. A result should be visible for either My AMIs the current account created the AMI originally, or Community AMIs if the AMI was shared. Sharing an AMI with a specific user was not tested, but may should show up in the My AMIs section. Once found, click the Select button next to the desired AMI to start creation of an instance of that image.

Figure 37. Select Previously Created AMI



Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

Selected AMI: (ami-069d73f3235b535bd) (Quickstart AMIs)

ami-0fc51199edeb3caf7

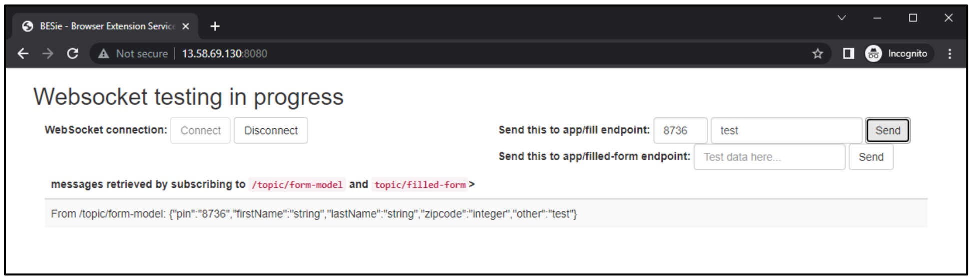
Quickstart AMIs (0) Commonly used AMIs
My AMIs (1) Created by me
AWS Marketplace AMIs (8201) AWS & trusted third-party AMIs
Community AMIs (0) Published by anyone
Refine results
Clear all filters
Owner
Owned by me
Shared with me
OS category

ami-0fc51199edeb3caf7 (1 filtered, 1 unfiltered)

BESie Service
ami-0fc51199edeb3caf7
BESie Service version 0.0.1 with Systemd startup
OwnerAlias: – Platform: Other Linux Architecture: x86_64
Owner: 072717644822 Publish date: 2023-07-19 Root device type: ebs
Virtualization: hvm ENA enabled: Yes Boot mode: uefi-preferred

3. Select or create Key Pair (see instructions and figures in the Create EC2 Instance from Amazon Linux AMI section)
4. Click “Add security group rule” and set to Type to **Custom TCP**, Port range to **8080**, Source type **Anywhere**. (See instructions and figures in the Create EC2 Instance from Amazon Linux AMI section)
5. Click the “Launch Instance” button.
6. Verify service is running in the browser by going to http://<PublicIP>:8080/.

Figure 38. Verify BESie Instance from Test Page


BESie - Browser Extension Servic
Not secure 13.58.69.130:8080
Incognito

Websocket testing in progress

WebSocket connection: Connect Disconnect

Send this to app/fill endpoint: 8736 test Send

Send this to app/filled-form endpoint: Test data here... Send

messages retrieved by subscribing to /topic/form-model and topic/filled-form >

From /topic/form-model: {"pin":"8736","firstName":"string","lastName":"string","zipcode":"integer","other":"test"}

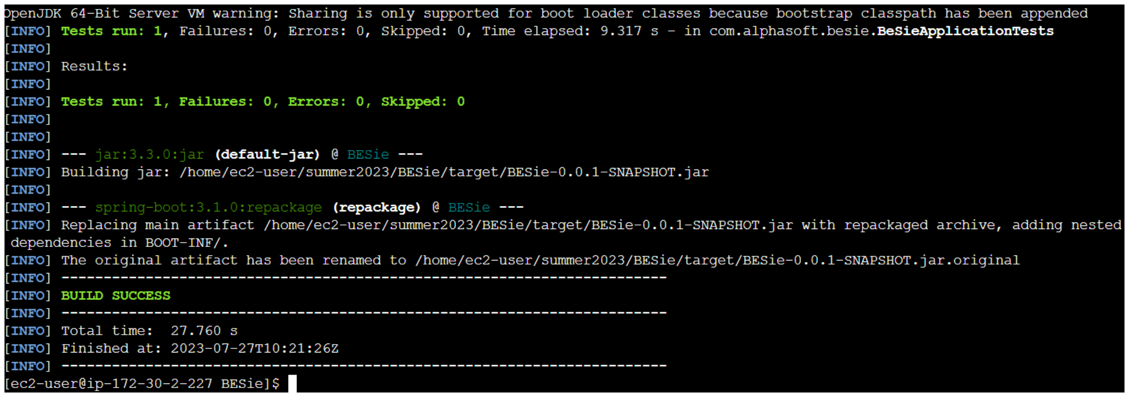
Updating the BESie Service

To update BESie, connect to the AWS BESie instance through the AWS console’s instance page. Once connected, type the following commands on the command line:

```
1. sudo systemctl stop besie
2. cd ~/summer2023/BESie
3. git pull
4. ./mvnw clean package
```

You should see the following output indicating a successful build and an output JAR file, here it is `target/BESie-0.0.1-SNAPSHOT.jar`.

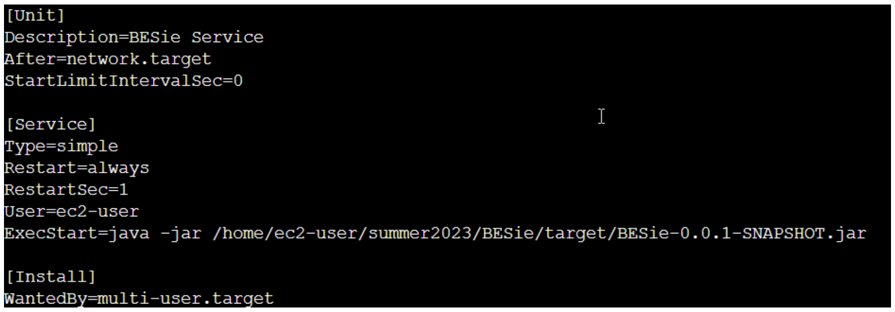
Figure 39. BESie Built Successfully



```
OpenJDK 64-Bit Server VM warning: Sharing is only supported for boot loader classes because bootstrap classpath has been appended
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 9.317 s - in com.alphasoft.besie.BeSieApplicationTests
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO]
[INFO] --- jar:3.3.0:jar (default-jar) @ BESie ---
[INFO] Building jar: /home/ec2-user/summer2023/BESie/target/BESie-0.0.1-SNAPSHOT.jar
[INFO]
[INFO] --- spring-boot:3.1.0:repackage (repackage) @ BESie ---
[INFO] Replacing main artifact /home/ec2-user/summer2023/BESie/target/BESie-0.0.1-SNAPSHOT.jar with repackaged archive, adding nested
 dependencies in BOOT-INF/.
[INFO] The original artifact has been renamed to /home/ec2-user/summer2023/BESie/target/BESie-0.0.1-SNAPSHOT.jar.original
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  27.760 s
[INFO] Finished at: 2023-07-27T10:21:26Z
[INFO] ------------------------------------------------------------------------
[ec2-user@ip-172-30-2-227 BESie]$
```

5. Type "`sudo nano /etc/system/system/besie.service`" and Ensure that output jar matches file path shown in ExecStart line.

Figure 40. Verifying BESie ExecStart Command



```
[Unit]
Description=BESie Service
After=network.target
StartLimitIntervalSec=0

[Service]
Type=simple
Restart=always
RestartSec=1
User=ec2-user
ExecStart=java -jar /home/ec2-user/summer2023/BESie/target/BESie-0.0.1-SNAPSHOT.jar

[Install]
WantedBy=multi-user.target
```

Once the ExecStart line is verified, or updated to the correct path of the new JAR file, run the following to start BESie up again:

6. `sudo systemctl start besie`

Deploying Transcription Services

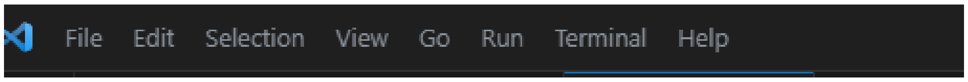
1. From the application root directory, navigate to /backend_services/aws-resources.
2. Ensure the following are installed globally on your machine: aws-cli and aws-cdk. Please see the following links for cli or cdk installation: https://aws.amazon.com/cli/, https://docs.aws.amazon.com/cdk/v2/guide/getting_started.html
3. Ensure that the AWS access id and key associated with your aws account are properly configured. To verify please refer to the following document: https://docs.aws.amazon.com/cli/latest/userguide/cli-chap-configure.html
4. Run "aws –version" and "cdk –version" commands in the terminal.
5. Execute the command "cdk bootstrap" and ensure there are no errors in the command line
6. Then execute the "cdk deploy" command and ensure there are no errors in the command line.
7. Verify the two s3 buckets are deployed by executing the command "aws s3 ls" in the terminal.
8. Verify the two buckets are named "convo-budy-transcript" and "convo-budy-audio-files".
9. Verify the API gateway is deployed by running the command "aws apigateway get-rest-apis".

10. Verify the API endpoint is named “convobuddy”.
11. Verify the lambda functions are deployed by running aws lambda list-functions. The output should show two functions named convoBudyStartTranscript and convoBudySendTranscript.

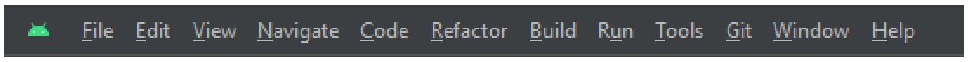
Troubleshooting

Troubleshooting from VS Code/Android Studio

Troubleshooting from VS Code involves using the debugging console to determine any errors that may have occurred. This is done by having the repository open in VS Code, clicking Run on the upper bar, and click Debugging from the dropdown that appears.



A similar process is done in Android Studio. The user has the project open within Android Studio, proceeds to select Run from the upper bar, and click on Debug from the dropdown menu.



Additionally, users can use the terminal in either VS Code or Android Studio and go to specific directories and run the command flutter analyze to get any specific errors that occur in a portion of the project.

Troubleshooting from AWS

Troubleshooting from AWS involves looking up the specific errors that are outputted by AWS. This can be done by going to https://docs.aws.amazon.com/AmazonS3/latest/userguide/troubleshooting.html and looking at what error is being outputted by the S3 bucket service. Additionally, AWS can be troubleshooted by contacting AWS Support with the errors being outputted by the servi.

Additionally, troubleshooting AWS transcription from the local machine involves following these instructions:

Troubleshooting Transcription Services

1. If cdk fails to deploy verify the following
   a. Valid AWS credentials are in the .aws/credentials file on your machine.
   b. Services are deployed in the same region (example: us-east-1).
   c. Verify the AWS account listed has admin permissions(this user should have console privileges).

If the transcription service still fails contact the appropriate point of contact on the team to ensure AWS services are not impacted by other factors.