# Server Frontend and Backend Setup Guide

# This document provides step-by-step instructions for running the frontend and backend of your server using PM2, as well as instructions for installing PM2 and including a PPK file for SSH authentication.

# Frontend Setup

# Step 1: Log into the Server

# Open PuTTY. If you don’t have putty on your system, you can use <https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html> to download it.

# Enter the server's IP address and port.

# In the left pane, navigate to "Connection" -> "SSH" -> "Auth" -> "Credentials".

# Under "Private key file for authentication," click "Browse" and select your PPK file.

# Return to the "Session" tab, provide a session name in the "Saved Sessions" field, and click "Save" for future use.

# Click "Open" to connect to the server.

# Step 2: Install PM2

# If PM2 is not installed, you can install it globally using npm:

# npm install -g pm2

# Step 3: Navigate to the Frontend Directory

# Example,

# cd Workspace/SAMS/SAMS-Frontend

# Step 4: Build the Frontend Application

# npm run build

# // It generates a dist folder. The "dist" folder is often generated as part of a build process, which takes the source code of a frontend application (written in languages like HTML, JavaScript, and CSS) and compiles, optimizes and bundles it for deployment to a production server. This process helps reduce file sizes and improve the loading speed of the application.

# In the context of your instructions, running `npm run build` likely triggers the build process for your frontend application and the resulting optimized and bundled files are placed in the "dist" folder. Once the "dist" folder is generated, it can be served by a web server to make the frontend of your application accessible to users.

# Step 5: Check if the Frontend is Already Running

# pm2 list

# If the frontend is running and you wanna update changes, follow these steps:

# Run `npm run build` to build the frontend.

# Identify the process ID (ID) of the frontend process in the PM2 list.

# Stop the existing frontend process:

# pm2 delete ID

# Save the changes.

# pm2 save

# Step 6: Start the Frontend

# To start the frontend, run the following command:

# pm2 serve dist 3000 –spa

# Step 7: Save the Configuration

# After starting the frontend, save the PM2 configuration:

# pm2 save

# Backend Setup

# Step 1: Navigate to the Backend Directory

# Example,

# cd Workspace/SAMS/SAMS-Backend/backend

# Step 2: Set the Django Secret Key

# Set the Django secret key as an environment variable. Replace <your\_secret\_key> with your actual secret key.

# export DJANGO\_SECRET\_KEY="django-insecure-<your\_secret\_key>"

# i.e,

# export DJANGO\_SECRET\_KEY="django-insecure-x^s-eh+r4es@\*(57wstt=brrv$w5pqrze^#gp%&&#ie=8znm59"

# Step 3: Check if the Backend is Already Running

# pm2 list

# If the backend is running and you wanna update changes, follow these steps:

# Identify the process ID (ID) of the backend process in the PM2 list.

# Stop the existing backend process:

# pm2 delete ID

# Save the changes.

# pm2 save

# Step 4: Start the Backend

# To start the backend, run the following command, replacing <your\_secret\_key> with your actual secret key:

# pm2 start python3 --name "backend-app" -- manage.py runserver 0.0.0.0:8000

# Step 5: Save the Configuration

# After starting the backend, save the PM2 configuration:

# pm2 save

# Installation of Python and Node

# Installing Python:

# Visit the official Python website: https://www.python.org/downloads/windows/

# Download the latest Python installer for Windows. Choose the installer that matches your system architecture (usually 64-bit is recommended).

# Run the installer and follow the installation wizard.

# Make sure to check the box that says "Add Python X.Y to PATH" during installation (replace X.Y with the version number you downloaded).

# After the installation is complete, you can open a command prompt and type python --version to verify the installation.

# Installing Node.js:

# Visit the official Node.js website: https://nodejs.org/

# Download the LTS (Long Term Support) version, which is recommended for most users.

# Run the Node.js installer and follow the installation wizard, accepting the default settings.

# After the installation is complete, you can open a command prompt or terminal and verify the installation by running:

# node -v

# npm -v

# Cloning the file:

# Open the terminal and navigate to the target directory (e.g., workspace)

# Let’s say,

# cd workspace

# Then clone the repository by using the following command,

# git clone <https://NigunSanjai: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@github.com/NigunSanjai/SAMS.git>

# Running the backend server:

# Navigate to `sams-backend` and install the dependencies.

# i.e.,

# cd sams-backend

# pip install -r requirements.txt

# Once, the installation has been done, navigate to the backend folder,

# cd backend

# Enter the secret key,

# $env:DJANGO\_SECRET\_KEY = 'django-insecure-x^s-eh+r4es@\*(57wstt=brrv$w5pqrze^#gp%&&#ie=8znm59'

# Run the server using.,

# python manage.py runserver

# Running the Frontend server:

# Now, navigate to `sams-frontend` folder and install the dependencies,

# cd sams-frontend

# npm install

# Once, the installation has been done, run the server using,

# npm run dev