Creating and Deploying a Personal Package Archive (PPA) for SSH Log Monitoring

ENPM818P Final Project - Group 4

Bavan Mooganahally Yadunath
Derick Ansignia
Piyush Goenka
Tanvi Kanchan

UNIVERSITY OF MARYLAND FEARLESSLY FORWARD

Introduction

Introduction

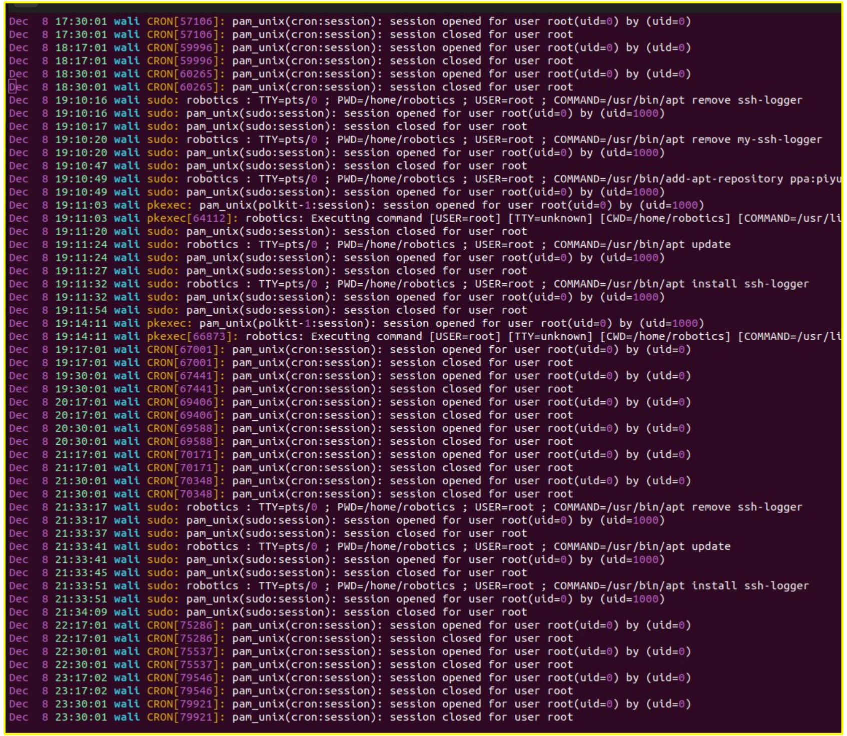
Organizations infrastructures **demand** robust **log monitoring** to ensure security, operational efficiency, and compliance.

Traditional decentralized log management is **insufficient** for scalable environments requiring real-time insights.

Our solution provides a **centralized**, **automated log monitoring** system for Ubuntu servers, integrating seamlessly with **Firebase** for comprehensive log analysis. We specifically capture **SSH login** attempt details in order to continuously monitor and prevent security breaches.

We use the following software tools in order to achieve our objective:

- **Software:** Python3 package to capture critical login metadata (IP, timestamp, username)
- **Storage:** Firebase
- **Software distribution:** Personal Package Archive (PPA)
- **Package Management:** Launchpad PPA
- **Deployment:** Ansible



```
Dec  8 17:30:01 wali CRON[57106]: pam_unix(cron:session): session opened for user root(uid=0) by (uid=0)
Dec  8 17:30:01 wali CRON[57106]: pam_unix(cron:session): session closed for user root
Dec  8 18:17:01 wali CRON[59996]: pam_unix(cron:session): session opened for user root(uid=0) by (uid=0)
Dec  8 18:17:01 wali CRON[59996]: pam_unix(cron:session): session closed for user root
Dec  8 18:30:01 wali CRON[60265]: pam_unix(cron:session): session opened for user root(uid=0) by (uid=0)
Dec  8 18:30:01 wali CRON[60265]: pam_unix(cron:session): session closed for user root
Dec  8 19:10:16 wali sudo: robotics : TTY=pts/0 ; PWD=/home/robotics ; USER=root ; COMMAND=/usr/bin/apt remove ssh-logger
Dec  8 19:10:16 wali sudo: pam_unix(sudo:session): session opened for user root(uid=0) by (uid=1000)
Dec  8 19:10:17 wali sudo: pam_unix(sudo:session): session closed for user root
Dec  8 19:10:20 wali sudo: robotics : TTY=pts/0 ; PWD=/home/robotics ; USER=root ; COMMAND=/usr/bin/apt remove my-ssh-logger
Dec  8 19:10:20 wali sudo: pam_unix(sudo:session): session opened for user root(uid=0) by (uid=1000)
Dec  8 19:10:47 wali sudo: pam_unix(sudo:session): session closed for user root
Dec  8 19:10:49 wali sudo: robotics : TTY=pts/0 ; PWD=/home/robotics ; USER=root ; COMMAND=/usr/bin/add-apt-repository ppa:piyu
Dec  8 19:10:49 wali sudo: pam_unix(sudo:session): session opened for user root(uid=0) by (uid=1000)
Dec  8 19:11:03 wali pkexec: pam_unix(polkit-1:session): session opened for user root(uid=0) by (uid=1000)
Dec  8 19:11:03 wali pkexec[64112]: robotics: Executing command [USER=root] [TTY=unknown] [CWD=/home/robotics] [COMMAND=/usr/li
Dec  8 19:11:20 wali sudo: pam_unix(sudo:session): session closed for user root
Dec  8 19:11:24 wali sudo: robotics : TTY=pts/0 ; PWD=/home/robotics ; USER=root ; COMMAND=/usr/bin/apt update
Dec  8 19:11:24 wali sudo: pam_unix(sudo:session): session opened for user root(uid=0) by (uid=1000)
Dec  8 19:11:27 wali sudo: pam_unix(sudo:session): session closed for user root
Dec  8 19:11:32 wali sudo: robotics : TTY=pts/0 ; PWD=/home/robotics ; USER=root ; COMMAND=/usr/bin/apt install ssh-logger
Dec  8 19:11:32 wali sudo: pam_unix(sudo:session): session opened for user root(uid=0) by (uid=1000)
Dec  8 19:11:54 wali sudo: pam_unix(sudo:session): session closed for user root
Dec  8 19:14:11 wali pkexec: pam_unix(polkit-1:session): session opened for user root(uid=0) by (uid=1000)
Dec  8 19:14:11 wali pkexec[66873]: robotics: Executing command [USER=root] [TTY=unknown] [CWD=/home/robotics] [COMMAND=/usr/li
Dec  8 19:17:01 wali CRON[67001]: pam_unix(cron:session): session opened for user root(uid=0) by (uid=0)
Dec  8 19:17:01 wali CRON[67001]: pam_unix(cron:session): session closed for user root
Dec  8 19:30:01 wali CRON[67441]: pam_unix(cron:session): session opened for user root(uid=0) by (uid=0)
Dec  8 19:30:01 wali CRON[67441]: pam_unix(cron:session): session closed for user root
Dec  8 20:17:01 wali CRON[69406]: pam_unix(cron:session): session opened for user root(uid=0) by (uid=0)
Dec  8 20:17:01 wali CRON[69406]: pam_unix(cron:session): session closed for user root
Dec  8 20:30:01 wali CRON[69588]: pam_unix(cron:session): session opened for user root(uid=0) by (uid=0)
Dec  8 20:30:01 wali CRON[69588]: pam_unix(cron:session): session closed for user root
Dec  8 21:17:01 wali CRON[70171]: pam_unix(cron:session): session opened for user root(uid=0) by (uid=0)
Dec  8 21:17:01 wali CRON[70171]: pam_unix(cron:session): session closed for user root
Dec  8 21:30:01 wali CRON[70348]: pam_unix(cron:session): session opened for user root(uid=0) by (uid=0)
Dec  8 21:30:01 wali CRON[70348]: pam_unix(cron:session): session closed for user root
Dec  8 21:33:17 wali sudo: robotics : TTY=pts/0 ; PWD=/home/robotics ; USER=root ; COMMAND=/usr/bin/apt remove ssh-logger
Dec  8 21:33:17 wali sudo: pam_unix(sudo:session): session opened for user root(uid=0) by (uid=1000)
Dec  8 21:33:37 wali sudo: pam_unix(sudo:session): session closed for user root
Dec  8 21:33:41 wali sudo: robotics : TTY=pts/0 ; PWD=/home/robotics ; USER=root ; COMMAND=/usr/bin/apt update
Dec  8 21:33:41 wali sudo: pam_unix(sudo:session): session opened for user root(uid=0) by (uid=1000)
Dec  8 21:33:45 wali sudo: pam_unix(sudo:session): session closed for user root
Dec  8 21:33:51 wali sudo: robotics : TTY=pts/0 ; PWD=/home/robotics ; USER=root ; COMMAND=/usr/bin/apt install ssh-logger
Dec  8 21:33:51 wali sudo: pam_unix(sudo:session): session opened for user root(uid=0) by (uid=1000)
Dec  8 21:34:09 wali sudo: pam_unix(sudo:session): session closed for user root
Dec  8 22:17:01 wali CRON[75286]: pam_unix(cron:session): session opened for user root(uid=0) by (uid=0)
Dec  8 22:17:01 wali CRON[75286]: pam_unix(cron:session): session closed for user root
Dec  8 22:30:01 wali CRON[75537]: pam_unix(cron:session): session opened for user root(uid=0) by (uid=0)
Dec  8 22:30:01 wali CRON[75537]: pam_unix(cron:session): session closed for user root
Dec  8 23:17:02 wali CRON[79546]: pam_unix(cron:session): session opened for user root(uid=0) by (uid=0)
Dec  8 23:17:02 wali CRON[79546]: pam_unix(cron:session): session closed for user root
Dec  8 23:30:01 wali CRON[79921]: pam_unix(cron:session): session opened for user root(uid=0) by (uid=0)
Dec  8 23:30:01 wali CRON[79921]: pam_unix(cron:session): session closed for user root
```

/var/log/auth.log

UNIVERSITY OF MARYLAND FEARLESSLY FORWARD

Objectives

1

Centralized Log Monitoring System with Near Real-time Monitoring Capabilities

2

Efficient Log Parsing and Categorization

3

Packaging and distribution through PPA

4

Automated Deployment with Ansible

5

Scalability and Adaptability

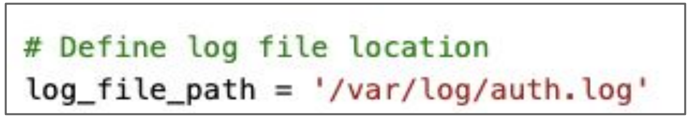
Implementation Process

Log Monitoring Script

Our log monitoring script (main.py) is a Python-based solution designed to parse authentication logs, classify login events, and upload categorized data to Firebase Firestore for centralized analysis and monitoring.

a. Functionality:

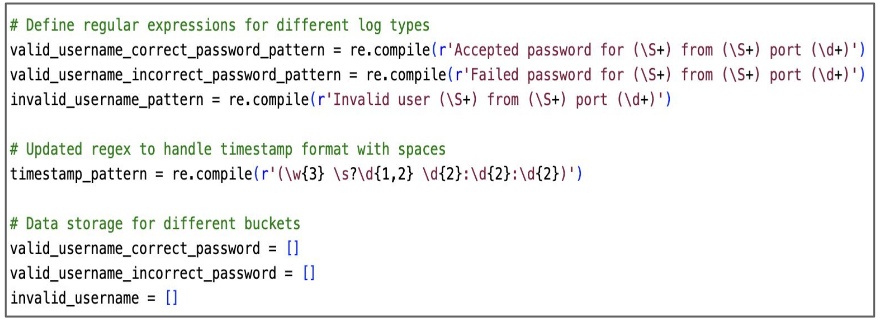
- It processes authentication logs from ***/var/log/auth.log.***
- It extracts SSH-related data from all the logs and uploads it to Firebase Firestore database.



```
# Define log file location
log_file_path = '/var/log/auth.log'
```

b. Key Features:

- The script parses authentication logs to identify successful (***Valid Username Correct Password***), failed (***Valid Username Incorrect Password***), and unauthorized login (***Invalid username***) attempts.

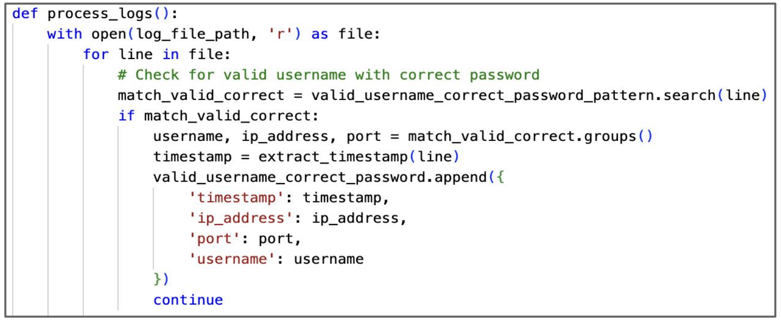


```
# Define regular expressions for different log types
valid_username_correct_password_pattern = re.compile(r'Accepted password for (\S+) from (\S+) port (\d+)')
valid_username_incorrect_password_pattern = re.compile(r'Failed password for (\S+) from (\S+) port (\d+)')
invalid_username_pattern = re.compile(r'Invalid user (\S+) from (\S+) port (\d+)')

# Updated regex to handle timestamp format with spaces
timestamp_pattern = re.compile(r'(\w{3} \s?\d{1,2} \d{2}:\d{2}:\d{2})')

# Data storage for different buckets
valid_username_correct_password = []
valid_username_incorrect_password = []
invalid_username = []
```

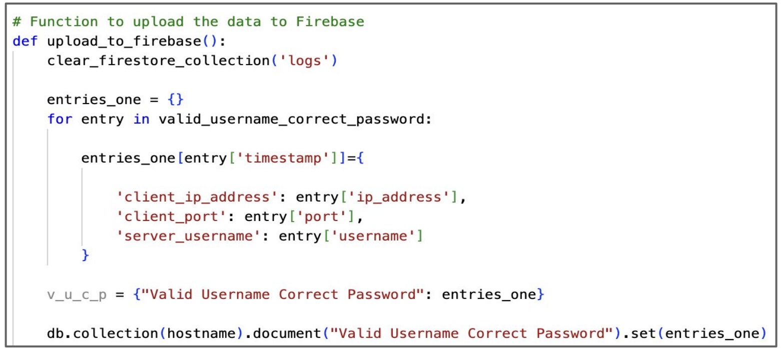
Log Monitoring Script

Key Features (cont):

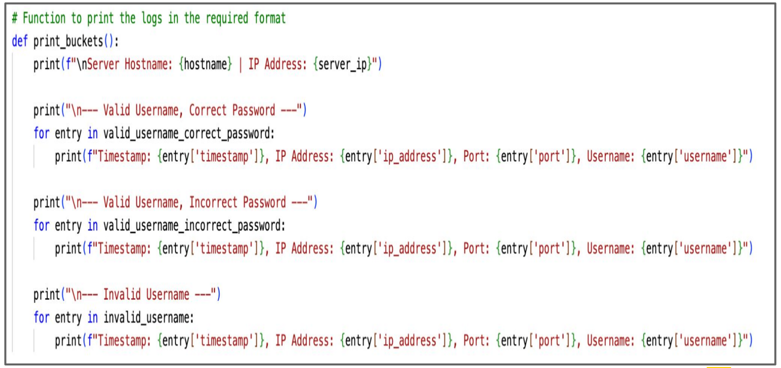
- It extracts timestamps for when the login attempt was made, IP addresses from where the login attempt was made, ports to which the user tried to connect on the system, and usernames.
- It gives local console output for immediate review.
- It pushes the parsed data to Firebase Firestore for centralized analysis.



```python
def process_logs():
    with open(log_file_path, 'r') as file:
        for line in file:
            # Check for valid username with correct password
            match_valid_correct = valid_username_correct_password_pattern.search(line)
            if match_valid_correct:
                username, ip_address, port = match_valid_correct.groups()
                timestamp = extract_timestamp(line)
                valid_username_correct_password.append({
                    'timestamp': timestamp,
                    'ip_address': ip_address,
                    'port': port,
                    'username': username
                })
                continue
```



```python
# Function to upload the data to Firebase
def upload_to_firebase():
    clear_firestore_collection('logs')

    entries_one = {}
    for entry in valid_username_correct_password:

        entries_one[entry['timestamp']]={

            'client_ip_address': entry['ip_address'],
            'client_port': entry['port'],
            'server_username': entry['username']
        }

    v_u_c_p = {"Valid Username Correct Password": entries_one}

    db.collection(hostname).document("Valid Username Correct Password").set(entries_one)
```



```python
# Function to print the logs in the required format
def print_buckets():
    print(f"\nServer Hostname: {hostname} | IP Address: {server_ip}")

    print("\n--- Valid Username, Correct Password ---")
    for entry in valid_username_correct_password:
        print(f"Timestamp: {entry['timestamp']}, IP Address: {entry['ip_address']}, Port: {entry['port']}, Username: {entry['username']}")

    print("\n--- Valid Username, Incorrect Password ---")
    for entry in valid_username_incorrect_password:
        print(f"Timestamp: {entry['timestamp']}, IP Address: {entry['ip_address']}, Port: {entry['port']}, Username: {entry['username']}")

    print("\n--- Invalid Username ---")
    for entry in invalid_username:
        print(f"Timestamp: {entry['timestamp']}, IP Address: {entry['ip_address']}, Port: {entry['port']}, Username: {entry['username']}")
```

Packaging and Distribution

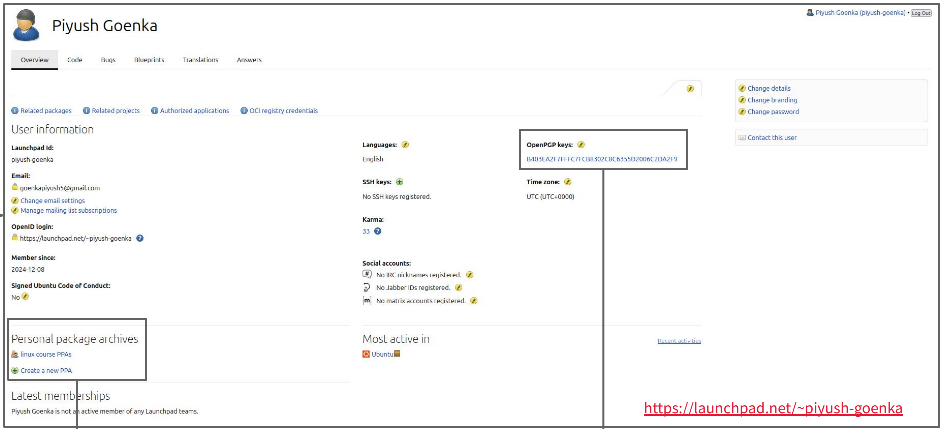
In Order to package and distribute our log monitoring script (main.py) as a python package into our PPA, we need to follow two main steps:

a. Setting up a PPA

- Creating a launchpad account
- Creating a PPA
- Creating and registering an OpenPGP key
- Importing the key to the launchpad account

b. Setting up and deploying the python package

- Creating a standard directory structure for .deb packages
- Building the python package
- Publishing the python package in PPA



Piyush Goenka (piyush-goenka) • Log Out

Piyush Goenka

Overview | Code | Bugs | Blueprints | Translations | Answers

Related packages | Related projects | Authorized applications | OCI registry credentials

User information

Launchpad Id:
piyush-goenka

Email:
goenkapiyush5@gmail.com
Change email settings
Manage mailing list subscriptions

OpenID login:
https://launchpad.net/~piyush-goenka

Member since:
2024-12-08

Signed Ubuntu Code of Conduct:
No

Languages:
English

SSH keys:
No SSH keys registered.

Karma:
33

Social accounts:
No IRC nicknames registered.
No Jabber IDs registered.
No matrix accounts registered.

OpenPGP keys:
B403EA2F7FFFC7FCB8302C8C6355D2006C2DA2F9

Time zone:
UTC (UTC+0000)

Change details
Change branding
Change password

Contact this user

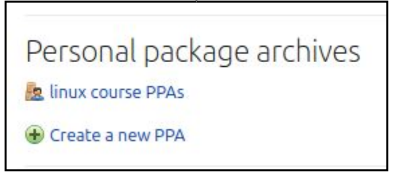
Personal package archives
linux course PPAs
Create a new PPA

Most active in
Ubuntu

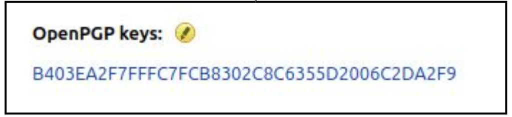
Recent activities

Latest memberships
Piyush Goenka is not an active member of any Launchpad teams.

https://launchpad.net/~piyush-goenka



Personal package archives
linux course PPAs
Create a new PPA



OpenPGP keys:
B403EA2F7FFFC7FCB8302C8C6355D2006C2DA2F9

Packaging and Distribution

Setting up and deploying the python package

- Creating a standard directory structure for .deb packages
- Building the python package
- Publishing the python package in PPA

ssh-logger python package structure:

```
ssh-logger
    |---- debian
    |       |---- changelog
    |       |---- compat
    |       |---- control
    |       |---- rules
    |---- src
    |         |---- __init__.py
    |         |---- main.py
    |---- setup.py
```

changelog

```
ssh-logger (1.0.0) jammy; urgency=medium

 * jammy release

-- Piyush Goenka <goenkapiyush5@gmail.com>  Wed, 04 Dec 2024 14:47:24 -0400
```

control

```
Source: ssh-logger
Maintainer: Piyush Goenka <goenkapiyush5@gmail.com>
Build-Depends: debhelper,dh-python,python3-all,python3-setuptools
Section: devel
Priority: optional
Standards-Version: 3.9.6
X-Python3-Version: >= 3.6

Package: ssh-logger
Architecture: any
Description: Log SSH events
Depends: ${python3:Depends},python3-requests
```

setup.py

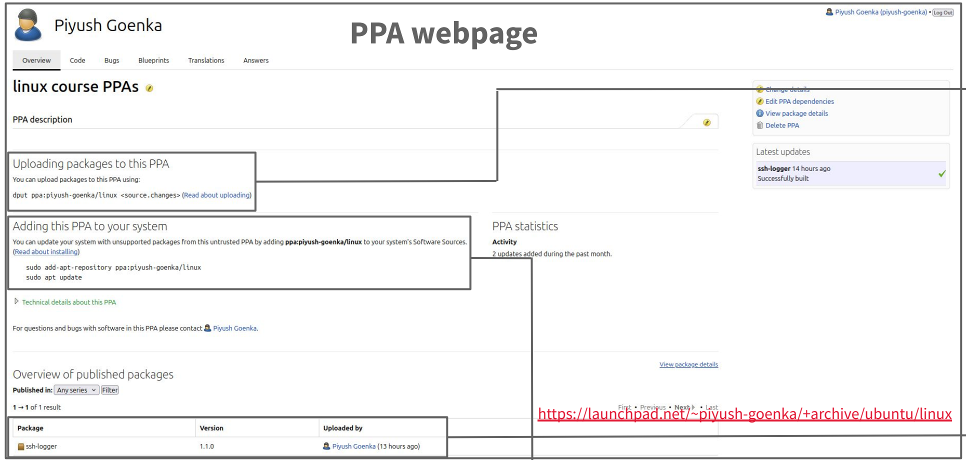
```
from setuptools import setup, find_packages


setup(
  name='ssh-logger',
  version='1.0.0',
  description='logging of ssh authentication events',
  author='Piyush Goenka',
  author_email='goenkapiyush5@gmail.com',
  license='MIT',
  install_requires=['firebase-admin>=6.0.0'],
  packages=find_packages(),
  entry_points=dict(
              console_scripts=['ssh_logger=src.main:run_logger']
  )
)
```

rules

```
#! /usr/bin/make -f
#export DH_VERBOSE = 1
export PYBUILD_NAME = ssh-logger

%:
  dh $@ --with python3 --buildsystem=pybuild
```

Packaging and Distribution

PPA webpage



linux course PPAs

PPA description

Uploading packages to this PPA

You can upload packages to this PPA using:

```
dput ppa:piyush-goenka/linux <source.changes>
```
(Read about uploading)

Adding this PPA to your system

You can update your system with unsupported packages from this untrusted PPA by adding **ppa:piyush-goenka/linux** to your system's Software Sources. (Read about installing)

```
sudo add-apt-repository ppa:piyush-goenka/linux
sudo apt update
```

Technical details about this PPA

For questions and bugs with software in this PPA please contact Piyush Goenka.

Latest updates

ssh-logger 14 hours ago
Successfully built

PPA statistics

Activity

2 updates added during the past month.

Overview of published packages

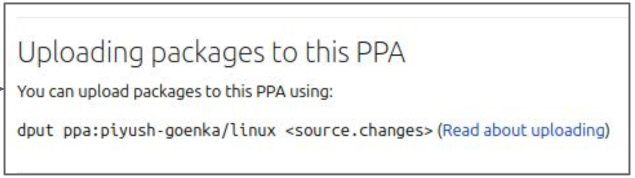
View package details

Published in:

1 → 1 of 1 result

Package	Version	Uploaded by
ssh-logger	1.1.0	Piyush Goenka (13 hours ago)

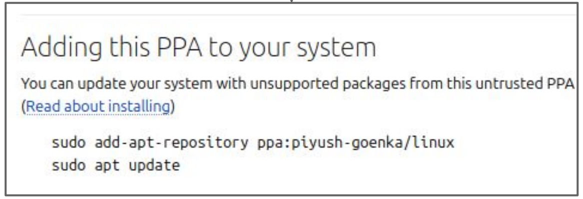
https://launchpad.net/~piyush-goenka/+archive/ubuntu/linux



Uploading packages to this PPA

You can upload packages to this PPA using:

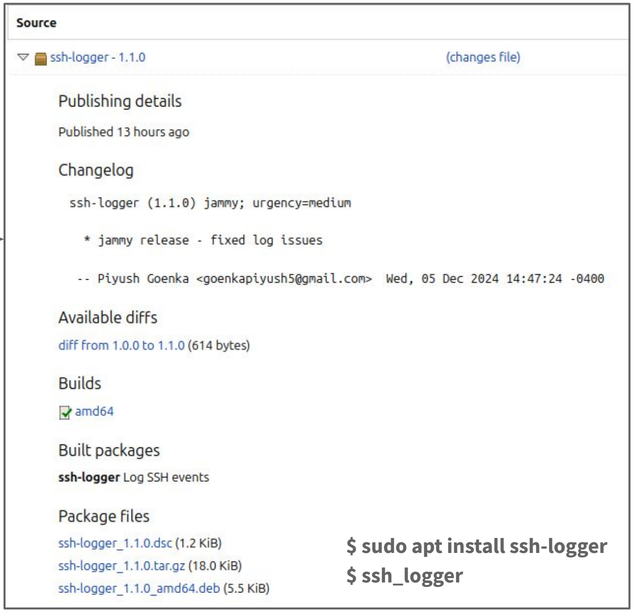
```
dput ppa:piyush-goenka/linux <source.changes>
```
(Read about uploading)



Adding this PPA to your system

You can update your system with unsupported packages from this untrusted PPA (Read about installing)

```
sudo add-apt-repository ppa:piyush-goenka/linux
sudo apt update
```



Source

ssh-logger - 1.1.0 (changes file)

Publishing details

Published 13 hours ago

Changelog

```
ssh-logger (1.1.0) jammy; urgency=medium

  * jammy release - fixed log issues

 -- Piyush Goenka <goenkapiyush5@gmail.com>  Wed, 05 Dec 2024 14:47:24 -0400
```

Available diffs

diff from 1.0.0 to 1.1.0 (614 bytes)

Builds

✓ amd64

Built packages

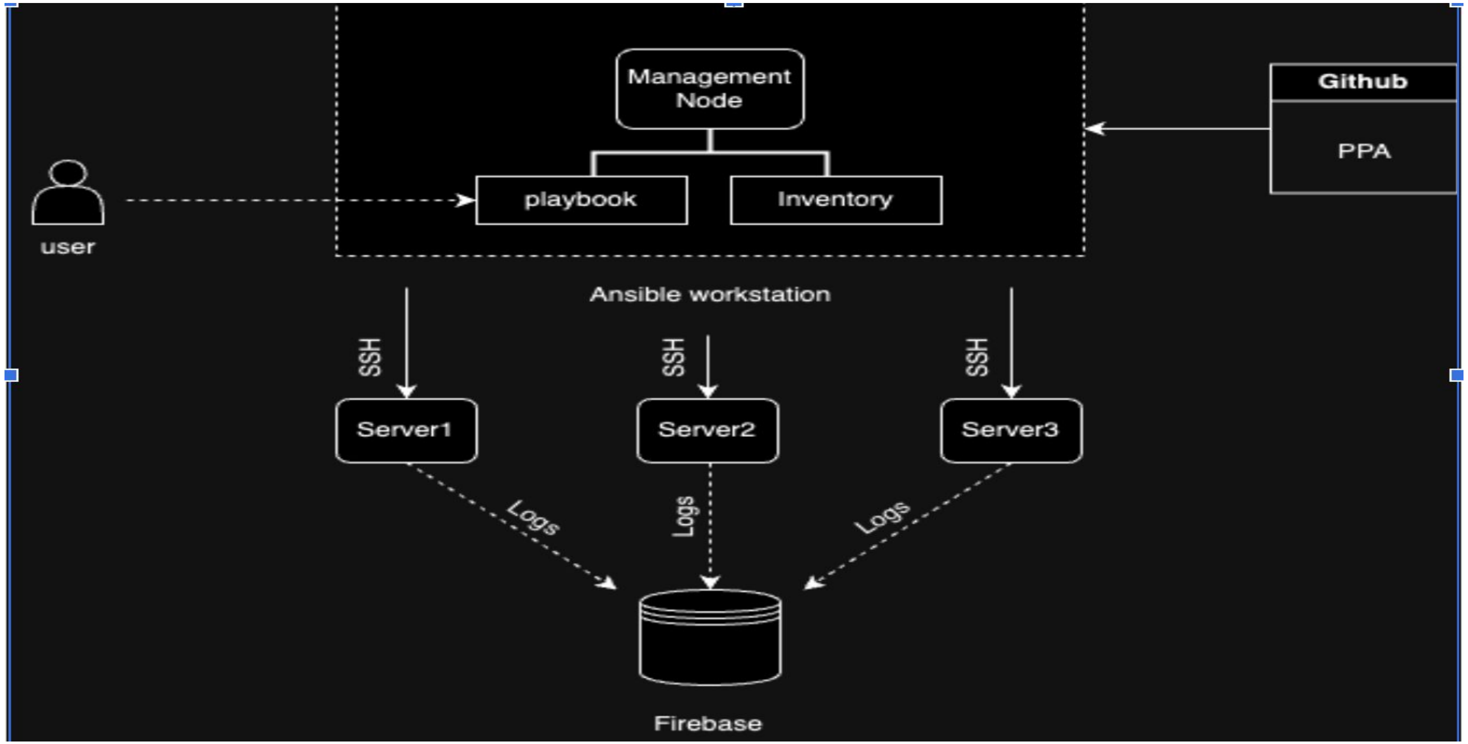
ssh-logger Log SSH events

Package files

ssh-logger_1.1.0.dsc (1.2 KiB)
ssh-logger_1.1.0.tar.gz (18.0 KiB)
ssh-logger_1.1.0_amd64.deb (5.5 KiB)

$ sudo apt install ssh-logger
$ ssh_logger

Workflow



Management Node
playbook
Inventory
user
Github
PPA
Ansible workstation
SSH
SSH
SSH
Server1
Server2
Server3
Logs
Logs
Logs
Firebase

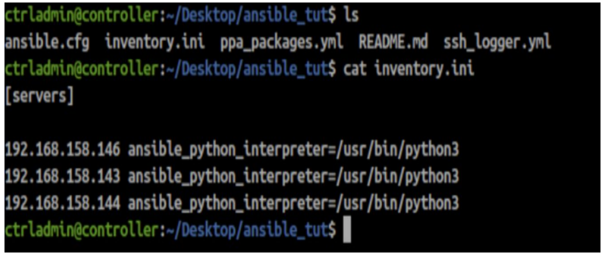
Workflow

a. **Playbook**

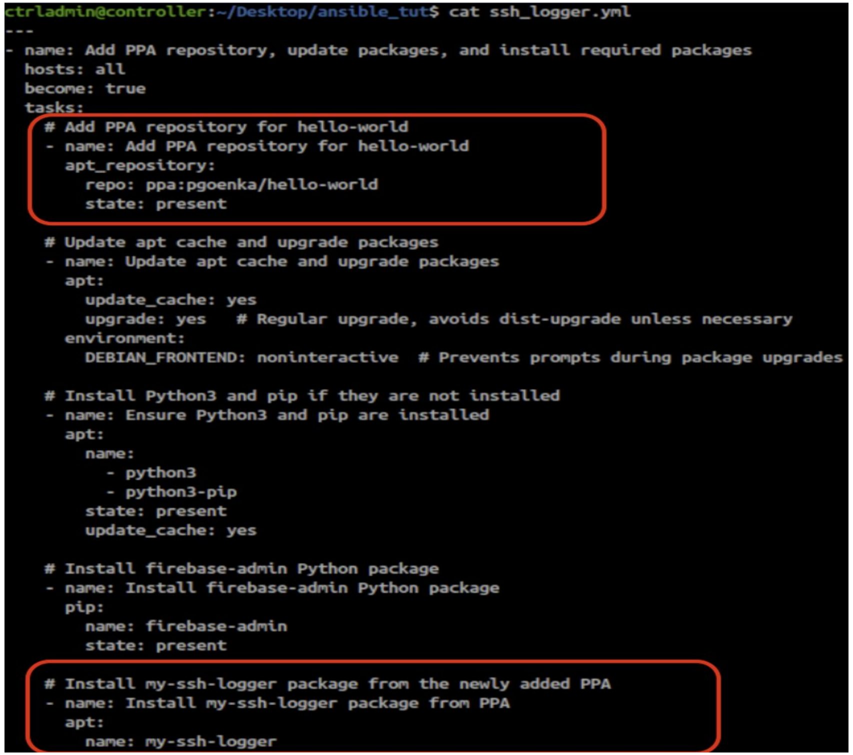
- Contains automation scripts and configuration definitions.
- Defines the desired state and tasks for server management.
- Executes configuration instructions across all three servers.

b. **Inventory**

- Stores server information and host details.
- Maintains list of all managed servers (Server1, Server2, Server3).
- Contains connection details and server groupings.



```
ctrladmin@controller:~/Desktop/ansible_tut$ ls
ansible.cfg  inventory.ini  ppa_packages.yml  README.md  ssh_logger.yml
ctrladmin@controller:~/Desktop/ansible_tut$ cat inventory.ini
[servers]

192.168.158.146 ansible_python_interpreter=/usr/bin/python3
192.168.158.143 ansible_python_interpreter=/usr/bin/python3
192.168.158.144 ansible_python_interpreter=/usr/bin/python3
ctrladmin@controller:~/Desktop/ansible_tut$
```



```
ctrladmin@controller:~/Desktop/ansible_tut$ cat ssh_logger.yml
---
- name: Add PPA repository, update packages, and install required packages
  hosts: all
  become: true
  tasks:
    # Add PPA repository for hello-world
    - name: Add PPA repository for hello-world
      apt_repository:
        repo: ppa:pgoenka/hello-world
        state: present

    # Update apt cache and upgrade packages
    - name: Update apt cache and upgrade packages
      apt:
        update_cache: yes
        upgrade: yes   # Regular upgrade, avoids dist-upgrade unless necessary
      environment:
        DEBIAN_FRONTEND: noninteractive  # Prevents prompts during package upgrades

    # Install Python3 and pip if they are not installed
    - name: Ensure Python3 and pip are installed
      apt:
        name:
          - python3
          - python3-pip
        state: present
        update_cache: yes

    # Install firebase-admin Python package
    - name: Install firebase-admin Python package
      pip:
        name: firebase-admin
        state: present

    # Install my-ssh-logger package from the newly added PPA
    - name: Install my-ssh-logger package from PPA
      apt:
        name: my-ssh-logger
```

Workflow

c. **Primary Function of Management Node:**

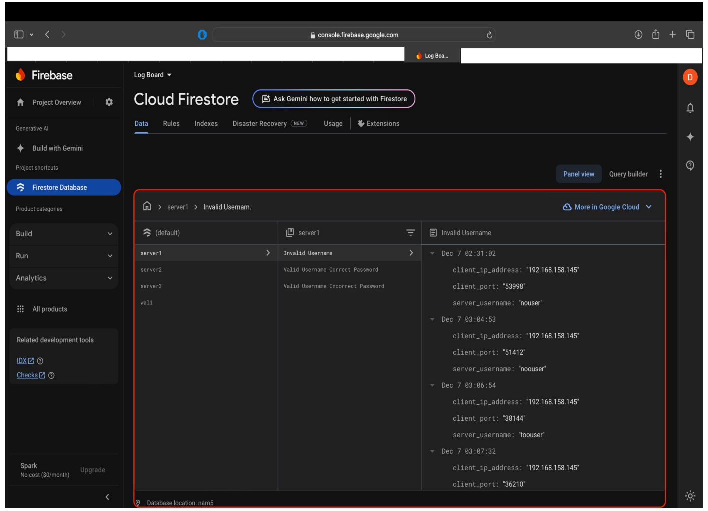
- Coordinating deployments to all three servers via SSH connections.
- Processing configuration management tasks defined in playbooks.
- Managing server states and configurations uniformly.
- Ensuring consistent configuration across the infrastructure.

d. **Server-Side Logging:**

- Each server (Server1, Server2, Server3) generates its own logs during operation.
- Logs are sent independently through dedicated channels to Firebase.
- Dotted lines in the diagram indicate asynchronous log transmission.

e. **Firebase Integration:**

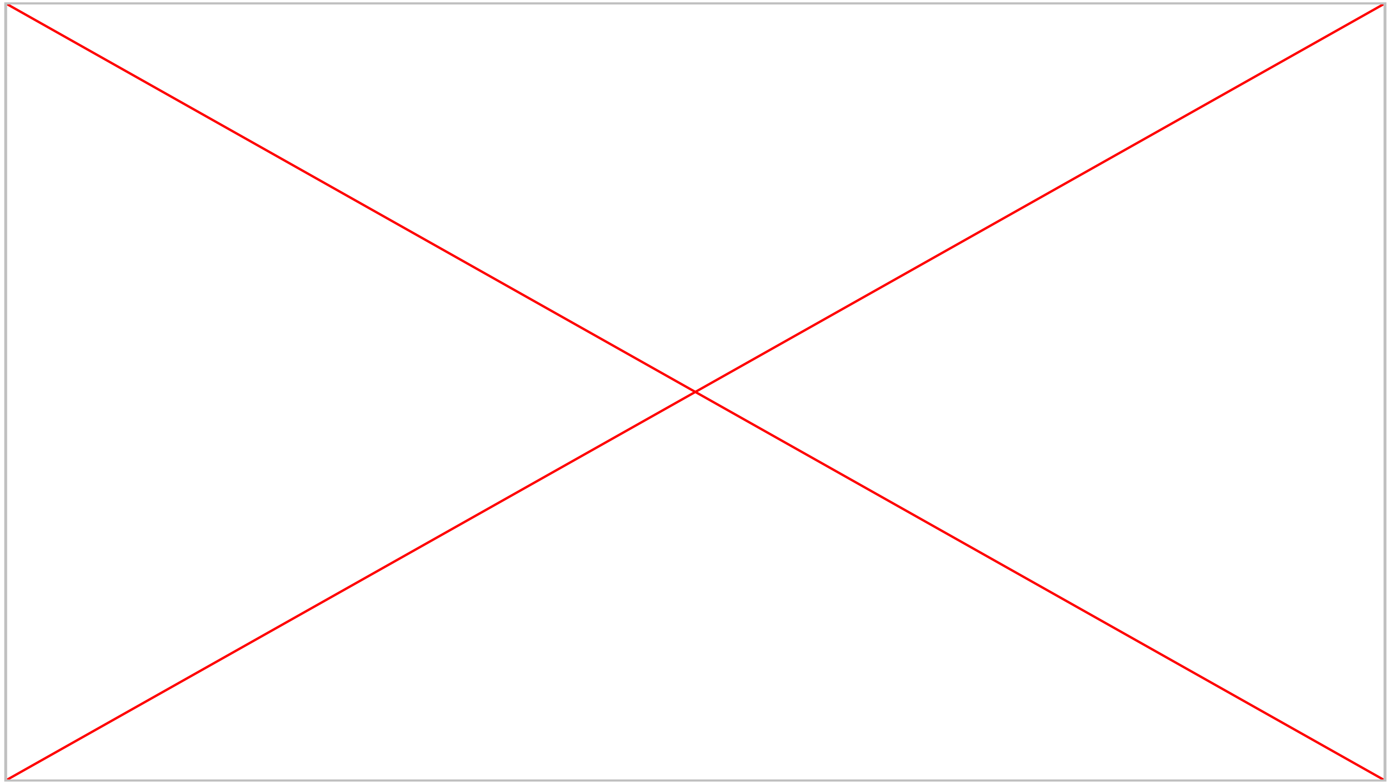
- Firebase acts as a centralized logging database.
- All three servers push their logs simultaneously.
- Near real-time log collection.



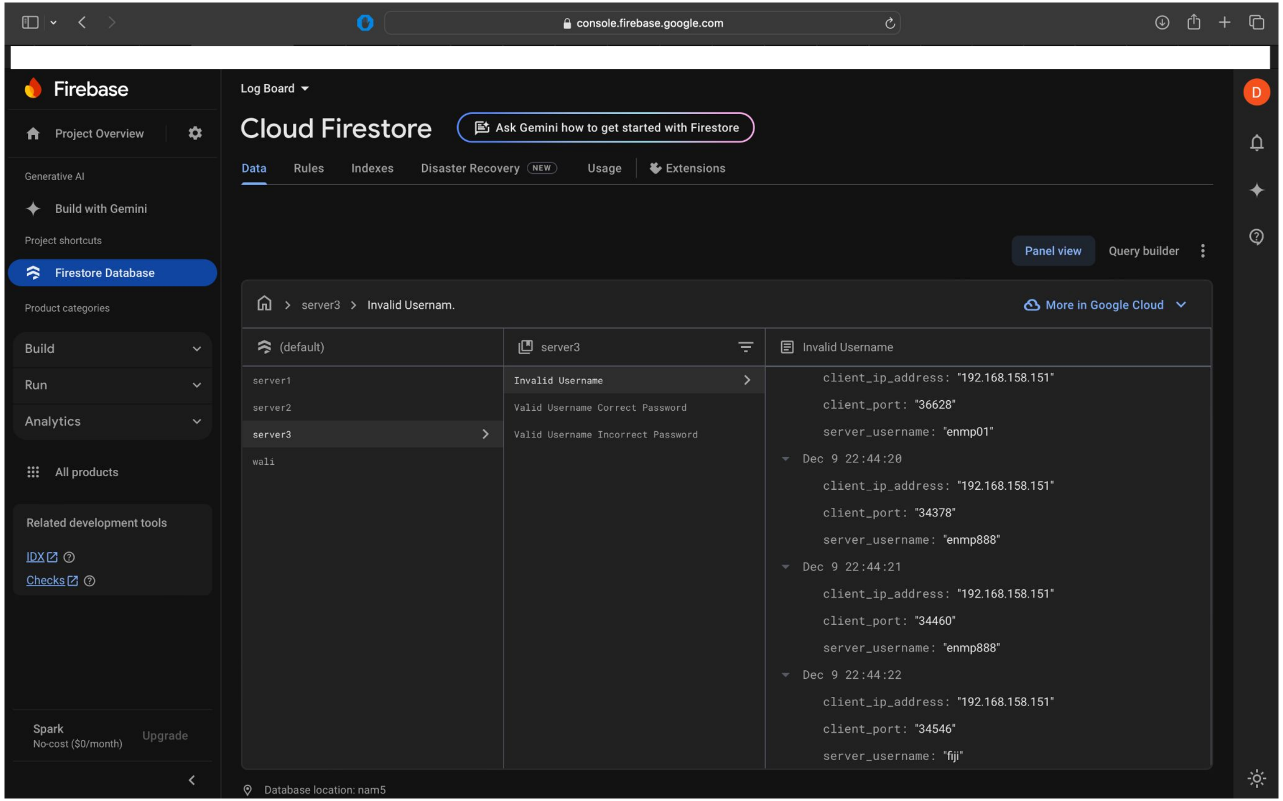
```
server1
server2
server3
wali

Invalid Username
Valid Username Correct Password
Valid Username Incorrect Password

Dec 7 02:31:02
    client_ip_address: "192.168.158.145"
    client_port: "53998"
    server_username: "nouser"
Dec 7 03:04:53
    client_ip_address: "192.168.158.145"
    client_port: "51412"
    server_username: "noouser"
Dec 7 03:06:54
    client_ip_address: "192.168.158.145"
    client_port: "38144"
    server_username: "toouser"
Dec 7 03:07:32
    client_ip_address: "192.168.158.145"
    client_port: "36210"
```

Demonstration

Demo Video: (Click here)



Results:



```
(default)        server3                                   Invalid Username
server1          Invalid Username                              client_ip_address: "192.168.158.151"
server2          Valid Username Correct Password               client_port: "36628"
server3          Valid Username Incorrect Password             server_username: "enmp01"
wali                                                       Dec 9 22:44:20
                                                               client_ip_address: "192.168.158.151"
                                                               client_port: "34378"
                                                               server_username: "enmp888"
                                                           Dec 9 22:44:21
                                                               client_ip_address: "192.168.158.151"
                                                               client_port: "34460"
                                                               server_username: "enmp888"
                                                           Dec 9 22:44:22
                                                               client_ip_address: "192.168.158.151"
                                                               client_port: "34546"
                                                               server_username: "fiji"
```

Conclusion