



# Making Teuthology Friendly

Google Summer of Code 2023 Proposal

## Personal Information

**Name:** Devansh Singh

**Email:** [devanshamity@gmail.com](mailto:devanshamity@gmail.com)

**GitHub:** [@Devansh3712](https://github.com/Devansh3712)

**LinkedIn:** [@devanshsingh3](https://www.linkedin.com/in/devanshsingh3)

**Slack:** @devansh

**University:** [Jaypee Institute of Information Technology, Noida](#)

**Course:** Bachelor of Technology in Computer Science Engineering

**Timezone:** Indian Standard Time (GMT +5:30)

# Table of Contents

<b>Personal Information</b>	<b>1</b>
<b>Project Information</b>	<b>3</b>
<b>Project Proposal</b>	<b>3</b>
API Workflow	5
Continuous Integration	7
UI/UX Mockups for Pulpito	8
User Authentication	8
Scheduling Job(s)	9
Killing Job(s)	9
Time Commitment	10
Deliverables	10
<b>Timeline</b>	<b>11</b>
<b>About Me</b>	<b>12</b>
<b>Why Me?</b>	<b>13</b>
<b>References</b>	<b>13</b>

# Project Information

**Mentors:** [Aishwarya Mathuriya](#), [Zack Cerza](#), [Kamoltat Sirivadhna](#)

**Goals:**

- **Ease of Use:** Implement a set of APIs for Teuthology and integrate it with Pulpito to make workflow easier
- **Documentation:** Document the API and create a Postman collection for the endpoints
- **Automation:** Automate testing and formatting of source code
- **Better UI/UX:** Work on Pulpito's user interface

## Project Proposal

Teuthology commands, in general, require a bit of a learning curve and usually requires someone who's a veteran with teuthology to help out, this can interrupt the productivity of the engineering team. The complexity of teuthology can lead to new users scheduling bad runs that can hold up the queue and block other jobs.

By creating an API and integrating it with Pulpito, developers can directly schedule or kill jobs without SSH-ing into teuthology through the website. The API will have endpoints based on the [teuthology](#) modules. The backend will majorly use the following libraries:

- **FastAPI:** Making the endpoints for the Teuthology API
- **Pydantic:** Validating request JSON payload for passing arguments to teuthology functions
- **Pytest:** Running unit-tests for the API endpoints
- **Tox:** Automate testing and formatting of the source code
- **Pre-Commit:** Running checks on a branch before committing or creating a pull request

A pre-commit config such as the following can be used for running basic checks. The checks will run on the developer's local machine before committing to the repository.

```
repos:
- repo: https://github.com/pre-commit/pre-commit-hooks
  rev: v4.1.0
  hooks:
  - id: check-ast
  - id: detect-private-key
  - id: end-of-file-fixer
  - id: fix-byte-order-marker
  - id: mixed-line-ending
  - id: trailing-whitespace
```

Tox will be used for automating the formatting and testing of the source code. In the future, multiple Python versions can be used for testing.

```
[tox]
env_list = format, py310
isolated_build = True

[testenv]
description = Run unit tests
passenv = *
deps =
    -r requirements.txt
commands = pytest -v

[testenv:format]
description = Format code
skip_install = true
deps = black
commands = black {posargs:.}
```

Therefore, the **main tasks** for the project will be:

- Implement backend APIs using teuthology modules
- Create unit tests and integration tests for the API
- Create documentation and create a Postman collection for API endpoints
- Adding user authentication to Pulpito
- Implement killing and scheduling jobs through Pulpito
- Create unit-tests for frontend widgets
- Integrate the frontend with backend

## API Workflow

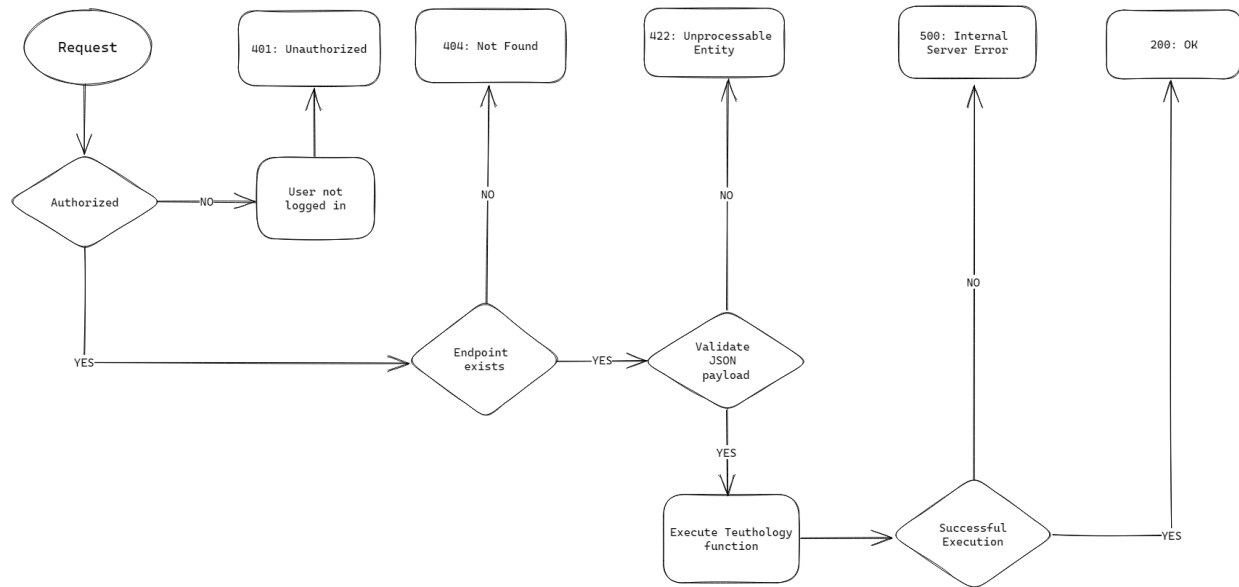
The workflow of a request would be as follows:

- The user should be authorized before making a request to the endpoints (GitHub OAuth)
- If the user session has expired, re-authorize the user
- The request payload should be a valid JSON (validated using pydantic)
- The arguments from the payload should be passed to the respective teuthology function and executed
- Fetch job execution details from Paddles
- Return the JSON success or error response

A JWT can be used for storing the user session which will be validated on every request, and the payload will include the following keys:

```
{
  "iat": "<unix_epoch>", // Issued At Time
  "exp": "<unix_epoch>", // Expiration
  "username": "<github_username>"
}
```

The whole workflow can be visualized as follows:



The following table contains the endpoints that will be implemented, along with the supported HTTP method and its description. The teuthology endpoints will require a user to be logged in, or else an `AuthorizationError` will be returned.

Endpoint	Method	Description
/login	GET	Login user using GitHub
/logout	GET	Logout user
/suite	POST	Run a suite of ceph integration tests
/schedule	POST	Schedule ceph integration tests
/kill	POST	Kill running teuthology jobs

## Continuous Integration

A GitHub action will be created for running Tox whenever a new commit or pull request is made to the repository. This will make sure that the new code that is being committed passes all the tests, and adheres to the format style of the source code.

```
name: Tox CI

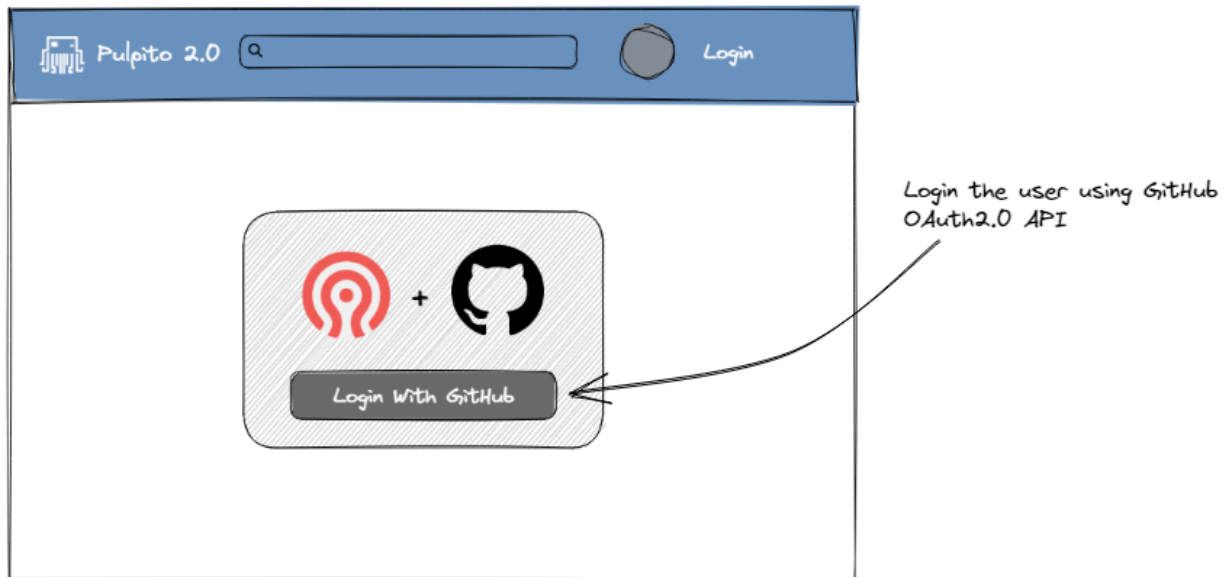
on:
  push:
    branches: '**'
  pull_request:
    branches: '**'

jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v3
      - name: Setup Python
        uses: actions/setup-python@v3
        with:
          python-version: '3.10'
      - name: Install Dependencies
        run: |
          python -m pip install --upgrade pip
          pip install tox
      - name: Run Tox
        run: tox
```

## UI/UX Mockups for Pulpito

### User Authentication

A user will log in using their GitHub account, and a user session will be created for them.



Once a user has logged in, they will have the following profile options as a menu:

- **Scheduled Jobs:** See the currently scheduled jobs
- **Past Jobs:** See the jobs that have been executed
- **Logout**

A navigation bar can be added, which will have the options for searching by status, branch, suite, machine type and scheduling job(s).



## Scheduling Job(s)

An Authenticated user can schedule jobs using the Pulpito UI.

The screenshot shows the 'Schedule Integration Tests' form in the Pulpito 2.0 UI. The header includes the Pulpito 2.0 logo, a search bar, a GitHub profile picture, and the username 'Devansh3712'. A dropdown menu on the right contains 'Scheduled Jobs', 'Past Jobs', and 'Logout'. The form fields are organized into two columns. The left column contains 'Suite Name', 'Worker', 'Number of Runs', 'Email', and 'Seed'. The right column contains 'Description', 'Priority', 'Timeout', 'Subset', and 'Conf File' (with an 'Upload' button). Below these are four checkboxes: 'No Nested Subset', 'Dry Run', 'First in Suite', and 'Last in Suite'. A green 'Schedule' button is at the bottom. Annotations include: a red asterisk indicating required fields; a note that the user's email ID can be fetched using the GitHub API; a note pointing to the GitHub profile picture; a dashed box for the navigation menu; and a note about checkboxes for boolean values.

\*: Required Fields

User's email ID can be fetched using GitHub API

GitHub Profile Picture

Scheduled Jobs  
Past Jobs  
Logout

Checkboxes for boolean values

Schedule

## Killing Job(s)

Job(s) can be killed using the kill button by the owner of the job.

The screenshot shows the 'Job Name' page in the Pulpito 2.0 UI. The header is identical to the previous screenshot. The main content area includes links for '<Job Name>', 'See other runs of <job> on <branch>?', 'See other runs on branch <name>?', and 'See other runs scheduled on <date>?'. Below these is a 'Detailed Jobs View' section with buttons for 'X Queued', 'X Total', 'Expand All', 'Description', and 'Kill Jobs'. A table with 12 columns (Status, Job ID, Links, Posted, Started, Updated, Runtime, Duration, Machine, Branch, Nodes, Kill) displays three rows of job data. Each row has a trash can icon in the 'Kill' column. Annotations include: a note pointing to the GitHub profile picture; a dashed box for the navigation menu; a note that only the owner of the jobs can kill them; and an arrow pointing to the trash can icons.

GitHub Profile Picture

Scheduled Jobs  
Past Jobs  
Logout

<Job Name>  
See other runs of <job> on <branch> ?  
See other runs on branch <name> ?  
See other runs scheduled on <date> ?

Detailed Jobs View v

X Queued X Total Expand All Description Kill Jobs

Only the owner of the jobs can kill them

Status	Job ID	Links	Posted	Started	Updated	Runtime	Duration	Machine	Branch	Nodes	Kill

## Time Commitment

I will be able to devote 35-40 hours per week during my semester break, and 20-25 hours per week while my college is open. I plan to contribute around 175 hours towards this project.

[Here](#) is my detailed semester schedule. I will have my end-semester examinations from 16 - 26 May, during which I cannot contribute much time. Other than that, I will make sure to have no commitments other than GSoC during my break and I haven't applied for any organization other than Ceph nor do I intend to.

## Deliverables

By the end of the project, the following will be delivered:

- A robust and well-documented API for teuthology
- A Postman collection of the API endpoints
- New teuthology API integrated with Pulpito
- Tests for frontend and backend
- Continuous integration for running tests using GitHub Actions

# Timeline

- **Before May 4**
  - Communicate with mentors to understand the details and goal of the project
  - Get familiar with Ceph and Teuthology codebase
  - Get the [GSoC task](#) done
- **May 4 - May 28**
  - Read documentation and working of Ceph, Teuthology test suite, Paddles and Pulpito
  - Go through the current implementation of the [Teuthology API](#)
  - Get in touch with mentors and discuss about how I intend to approach the problem
  - Work further on Pulpito mockups
- **May 29 - July 14**
  - Work on the remaining endpoints of the module, write documentation
  - Setup Pre-Commit and Tox for testing and formatting
  - Setup GitHub Action CI for on new commits/pull requests
  - Work on unit-tests and integration tests for all endpoints
  - Create Postman collection for the API endpoints
  - Prepare for midterm evaluation
- **July 14 - August 21**
  - Work on Pulpito 2.0 user authentication using GitHub
  - Add widgets for scheduling and killing jobs using Pulpito, integrate API with frontend
  - Work on unit-tests for widgets

- **August 21 - August 28**
  - Buffer week for any unpredictable problems
  - Code style and quality review, developer documentation
  - Prepare for the final evaluation

## About Me

I am a second-year CS undergrad and an intermediate Python & Go developer. I have been working with Python for the last four years. My main field of interest is backend development and I like to create and work on web servers. I have done freelancing, worked as a backend intern for a couple of startups, and been a part of Major League Hacking's prep fellowship in which I contributed to several open-sourced projects.

I also have essential experience in C, C++, HTML, CSS, and Javascript. Some technologies I use daily include Docker, Git & GitHub, Continuous Integration (GitHub Actions, TravisCI), SQL databases (MySQL, PostgreSQL), MongoDB, Redis, etc.

Some of my Python projects are:

- [tsuki](#): Open Sourced Social Media Platform
- [torweather](#): Email Notification System for TOR Relay Nodes
- [cmc-py](#): Unofficial CoinMarketCap API and Wrapper

My resume can be found [here](#) as well.

## Why Me?

I like to develop backend servers and APIs, and Python is the language I'm most familiar with. I also have industry experience from working as an intern and I think my technology stack matches the requirements of this project. I did some research on Ceph, and how it separates the storage software from the hardware and I am interested to know more about it. By being a part of Ceph under GSoC 2023, I will be able to learn about software-defined storage.

I love how open-source works, and participating in open-source activities has introduced me to things I would have never known before and has drastically improved my coding skills. I hope to contribute more to the open-source world. I first got involved in open source while participating in Google Code-In 2019, and I haven't stopped since. I participated in the MLH Prep fellowship which focuses mainly on open-source contributions.

## References

- <https://docs.ceph.com/projects/teuthology/en/latest/index.html>
- <https://www.youtube.com/playlist?list=PLrBUGilNAakNsOwHaIM27OBGKezQbUdM->
- [https://docs.ceph.com/en/latest/dev/developer\\_guide/testing\\_integration\\_tests/tests-integration-testing-teuthology-intro/](https://docs.ceph.com/en/latest/dev/developer_guide/testing_integration_tests/tests-integration-testing-teuthology-intro/)
- <https://pulpito.ceph.com/>
- <https://github.com/VallariAg/teuthology-api>