

Subplan 10 — July 20, 2024

- **DevOps & Deployment Labels: Deploying Social Media Automation Tools:** Completed production rollout on July 18, 2024, with zero downtime and 99.95% uptime in first 48 hours.
- **Integration & API Labels: Event-Driven Architecture for Social Media Automation:** Fully implemented AWS SNS event bus with Lambda v3.2.1 for asynchronous post status updates.
- **Testing & Quality Assurance Labels: End-to-End Testing for Instagram Automation:** Verified Instagram automation stability with 1000+ posts over 7 days, no failures reported.

Generated Dialogues (Exemplars, Truncated)

Turn 1 (Mar 1, 2024)

User: I'm trying to set up a Python 3.10 environment for automating social media posts, and I've chosen Tweepy v4.10.1 and Facebook SDK v3.1.0 for API integrations, but I'm not sure how to structure my project for better outreach. Can you help me plan this out in terms of scheduling and deadlines?

Assistant:

Certainly! Setting up an automated posting system with Tweepy and Facebook SDK is a great start.

Project Structure:

1. Project Setup (Mar 1–7) [Truncated ...]
2. Content Creation (Mar 8–14) [Truncated ...]
3. Automation Development (Mar 15–21) [Truncated ...]
4. Deployment and Monitoring (Mar 22–28) [Truncated ...]

User (Follow-up)

That sounds like a great plan! One thing though—could we add a step for testing posts on a smaller scale before full deployment? Just to make sure everything looks good and functions as expected.

Assistant

Absolutely, adding a step for testing the posts on a smaller scale is a great idea to ensure everything looks good and functions as expected before full deployment. Here's how you can incorporate that into the existing plan:

Updated Project Structure

- #### 1. **Project Setup [Truncated ...]
- #### 4. **Testing Phase (March 22 - March 24, 2024)**
- **Days 22-23:** Conduct a small-scale test.
 - Post a few test messages to a limited audience or a test account.
 - Verify that the posts look good and function as expected.
 - **Day 24:** Review the test results.
 - Address any issues found during the test phase.
 - Make final adjustments to the automation script and content.

[Truncated ...]

User (Mid-turn) -> Code Sharing

I'm trying to implement a feature to collect engagement metrics for Instagram posts, but I'm not sure how to integrate it with the existing PostgreSQL schema. I've designed a table for `instagram_posts` with foreign key to `posts` table, but I'm not sure how to handle the data migration. Can you help me with that? Here's my code:

```
import psycopg2
conn = psycopg2.connect(dbname="engagement_metrics", user="username",
```