ISLD Sentiment Bot Deployment Guide

Step-by-Step Deployment

Prerequisites

- Python 3.8+ environment.
- Tweepy, Requests, aiohttp, and python-dotenv installed:

```
pip install tweepy requests aiohttp python-dotenv
```

- Twitter/X Developer account with API credentials:
- Consumer Key (API Key)
- Consumer Secret (API Secret Key)
- Access Token
- Access Token Secret
- xAI API credentials (Grok): Obtain from x.ai/api.

Step 1: Setup API Credentials Securely

- Use `.env` file for credentials:

```
TWITTER_API_KEY=your_api_key
TWITTER_API_SECRET=your_api_secret
TWITTER_ACCESS_TOKEN=your_access_token
TWITTER_ACCESS_SECRET=your_access_secret
GROK_API_KEY=your_xai_api_key
```

- Load credentials securely in your Python script.

Step 2: Integrate LLM (Grok via xAI API)

- Add async functions to your Python script to generate dynamic tweets:

```
import aiohttp

async def generate_tweet_content(price: str, topic: str):
    prompt_templates = {
        "charity":
f"Generate a tweet highlighting ISLD's charitable impact for SIDS at price
${price}.",
        "market": f"Generate a tweet with a Raydium market update for ISLD
at price ${price}.",
        "engagement":
```

```
f"Generate a tweet to drive ISLD community engagement at price ${price}."
    }
    prompt = prompt_templates.get(topic, prompt_templates["engagement"])

    async with aiohttp.ClientSession() as session:
        async with session.post(
            "https://api.x.ai/v1/chat/completions",
            headers={"Authorization": f"Bearer
{os.getenv('GROK_API_KEY')}", "Content-Type": "application/json"},
            json={"model": "grok-3", "messages": [{"role": "user",
"content": prompt}], "max_tokens": 280}
        ) as response:
            data = await response.json()
            return data['choices'][0]['message']['content']
```

Step 3: Test Script Locally

- Run your script locally to verify LLM integration and tweet posting:

```
python isld_sentiment_bot.py
```

- Ensure tweets are successfully posted and generated dynamically.

Step 4: Hosting & Automation

- **Serverless (AWS Lambda, Heroku)**:
- Package your script with dependencies.
- Schedule executions with EventBridge or Heroku Scheduler.
- **VPS (AWS EC2, DigitalOcean)**:
- Install dependencies and deploy your script.
- Schedule automated runs using cron jobs:

```
0 */4 * * * python3 /path/to/isld_sentiment_bot.py
```

Step 5: Monitoring & Maintenance

- Regularly check logs and tweet engagements.
- Adjust scheduling frequency based on engagement and platform guidelines.

Potential Enhancements

- Further enhance tweet variety by tuning the LLM prompt parameters (e.g., temperature).
- Implement analytics for deeper insights into engagement metrics.
- Ensure compliance with platform policies and adapt to changes promptly.