Session 1371 - Get Started with Heroku and Slack App Integration

Exercise Guide

TDX 2025
Hands On Workshop

Introduction

Welcome to the *Digital Frontier*, where we're saddling up to integrate Heroku with Slack and lasso some real-time updates, automated tasks, and user-friendly features for a cattle management system, Trail Boss. By the end of this ride, you'll have the beginnings of an app to manage your own herd of cloud-based cattle, ready for anything the digital wilderness throws at you.

Workshop Goals

By the end of this Hands On Workshop, participants will:

1. Deploy a Python Flask + Slack Bolt SDK application to Heroku.
2. Configure Slack to interact with the application.
3. Implement a Slack Workflow that triggers an automatic feed order reminder.

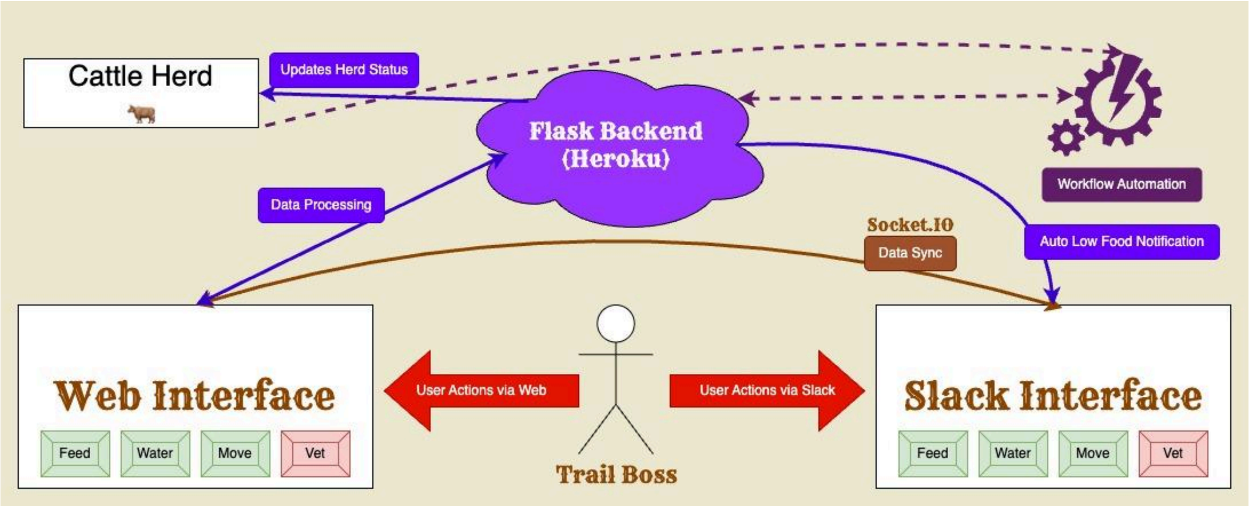
Prerequisites

1. Python installed, version 3.13.2 is used for the workshop.
2. Git / GitHub installed.
3. A Heroku Account and the Heroku CLI installed locally.
    a. Join the Heroku team here with your Heroku account
4. Access to a Slack workspace with a Pro or Business + Plan.
    a. You can request to join the TDX Heroku Slack Demo workspace here:
        - https://join.slack.com/t/tdxherokuslackdemo/shared_invite/zt-30zbvefrn-YTmZxWXOnlQccjVj1dC_rg2
5. Code editor of your choice to view the code files. We won't be writing application code in this workshop, but we'll be briefly walking through it and you can review the code to get a better sense of the application or to expand on it later.

Agenda

Setting up Camp: Preparing for the Ride

Welcome, Partner! Let's start this trail ride with a quick look at what you'll learn today: herding cattle in a western-themed app and integrating it with Slack and Heroku. The app is built in Python using the Flask micro web framework and leverages the Slack Bolt SDK and is deployed on Heroku. Let's take a quick look at the application architecture:



Claimin' Your Land: Cloning the Application from GitHub

Before you can build and maintain your cattle empire, you need digital land - let's grab the code from GitHub

- https://github.com/heroku-examples/heroku-slack-tdx25-hot
- Clone the repository. In a terminal:

Unset

```
git clone
https://github.com/heroku-examples/heroku-slack-tdx25-hot
your-repo-folder

cd your-repo-folder
```

Building the Ranch: Deploying to Heroku

Now that you've got the code, it's time to set up shop on Heroku, your digital cattle ranch:

- Ensure the Heroku CLI is installed

Unset

```
heroku --version
```

- Login to your Heroku account.

Unset

```
heroku login
```

- Create a new Heroku app, give it a name that'd make any cowpoke proud.

Unset

```
heroku create yourLastName-tdx25-trail-boss -t tdx25-heroku-slack
```

- Add and commit the files to Git and deploy the app to Heroku!

Unset

```
git add .
git commit -m "First deploy to Heroku"
git push heroku main
```

- Get the live Heroku URL

Unset

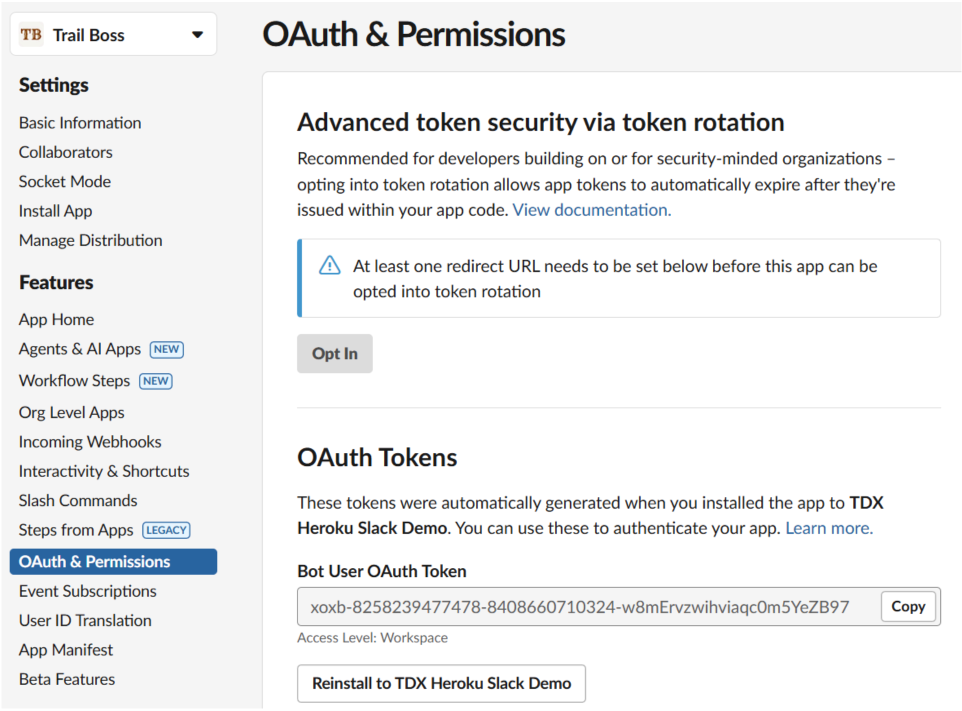
```
heroku open
```

  - This URL (e.g., https://your-app-name.herokuapp.com/) will be needed for Slack configuration
  - Note, the application will not work yet because it's looking for configuration variables that are provided in the next few steps. However, the app URL is needed at this point, so keep track of it.

Ropin' in the Herd: Setting Up the Application in Slack

Now that our digital ranch is set up, we need a trusty messenger to keep us updated - enter Slack!

- Go to Slack API Apps and create a new app. Name it something unique like yourLastName-TrailBoss
- Get the following settings:
  - **Slack Bot Token** found under the *OAuth & Permissions* section



OAuth & Permissions

Advanced token security via token rotation

Recommended for developers building on or for security-minded organizations – opting into token rotation allows app tokens to automatically expire after they're issued within your app code. View documentation.

⚠ At least one redirect URL needs to be set below before this app can be opted into token rotation

OAuth Tokens

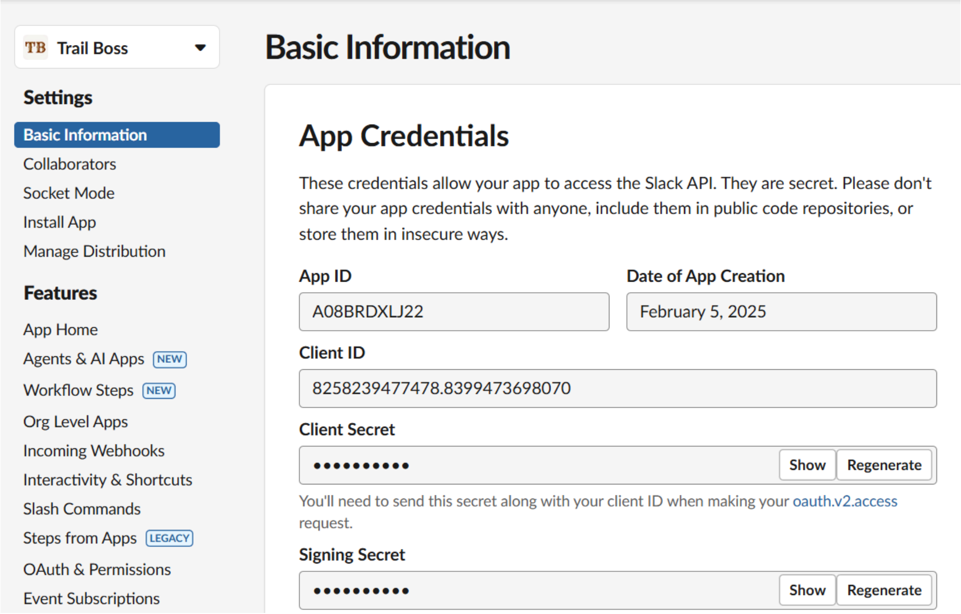
These tokens were automatically generated when you installed the app to **TDX Heroku Slack Demo**. You can use these to authenticate your app. Learn more.

Bot User OAuth Token

xoxb-8258239477478-8408660710324-w8mErvzwihviaqc0m5YeZB97

Access Level: Workspace

- ○ **Slack Signing Secret** found under the *Basic Information* section



Basic Information

App Credentials

These credentials allow your app to access the Slack API. They are secret. Please don't share your app credentials with anyone, include them in public code repositories, or store them in insecure ways.

App ID: A08BRDXLJ22

Date of App Creation: February 5, 2025

Client ID: 8258239477478.8399473698070

Client Secret: ••••••••••

You'll need to send this secret along with your client ID when making your oauth.v2.access request.

Signing Secret: ••••••••••

- Set up environment variables in Heroku for Slack credentials - the digital equivalent of a sheriff's badge.

Unset

```
heroku config:set SLACK_BOT_TOKEN='xoxb-your-bot-token'
heroku config:set SLACK_SIGNING_SECRET='your-signing-secret'
```

- Enable **Interactivity & Shortcuts**
  - ○ Set the *Request URL* to: https://your-app-name.herokuapp.com/slack/actions
- Enable **Event Subscriptions**
  - ○ Subscribe to the following bot events
    - ■ `app_mentions`
    - ■ `message.channels`
    - ■ `block_actions`
  - ○ Set the *Request URL* to: https://your-app-name.herokuapp.com/slack/events
- Lasso the right permissions and OAuth settings
  - ○ Add OAuth Scopes:
    - ■ `chat:write`
    - ■ `commands`
    - ■ `im:history`
    - ■ `channels:history`
- Install the app to your workspace

Keepin' the Herd in Line: Integrating a Slack Workflow for Feed Order Reminders

- Go to Slack Automations
- Create a new Workflow using the Slack Workflow Builder
    - Name the workflow something meaningful: Feed Reorder Reminder
- Set Trigger - lots of different options here, but we're going to do an event based workflow using a Webhook.
    - Trigger: Webhook
    - Copy the generated webhook URL
- Add the webhook URL to your Heroku Config Settings

Unset

```
heroku config:set SLACK_WEBHOOK_URL='your-Slack-Webhook-URL'
```

- Add a Slack Message Step in the Workflow & select a channel/recipient
    - Message: "The herd's feed supply is running low. Time to order more feed!"
    - Channel: `#supply-orders`
- Test the Workflow
    - Currently the app starts with a feed level of 40% and the notification is triggered when the feed level goes below 25%.
    - "Feed" the cattle twice to take the feed level down to 20% and check Slack for the automated response.

Wrappin' Up the Trail: Next Steps & Final Thoughts

- Recap
- Potential app enhancements
- Q&A

Thank You!

You're now a Heroku + Slack wrangler, ready to ride the digital frontier. Whether you're keepin' track of cattle or automating business updates, you've got the tools to handle the job.

Now go on, partner - ride tall in the saddle.

Application Enhancement Ideas

Additional Features

- Add a database for data management and persistence. Store historical herd data including past feed orders and health reports.
- Similar to the architecture for feeding the cattle, implement a workflow for watering the herd.
- Implement a Slack Workflow for requesting a vet visit.
- Implement a GPS tracking feature to update and visualize herd location on a map.
- Introduce a weather API integration to provide real-time weather alerts that could impact cattle movement and/or health.
- Automate the scheduling system for planning vet visits and supply restocks.
- Create an automated daily check-in for herd status and deliver summaries to key ranch personnel in Slack.
- Expand the Slack bot with role-based permissions so only authorized users can approve feed orders, move the herd, or call the vet.

AI Enhancements

- Implement AI-powered herd health monitoring using machine learning to analyze patterns and predict potential issues.
- Integrate national language processing (NLP) to allow users to interact with the Slack bot using free-text commands.
- Use AI-driven image recognition to detect cattle from drone footage and monitor herd movement and health.