ROI TRAINING
MAXIMIZE YOUR TRAINING INVESTMENT

# Cheat Sheet: Using Git

# Using Git (Prerequisites)

- In Google Cloud Shell
  - If using Cloud Shell Editor, select View | Toggle Hidden Files, so hidden (.) files are visible
- In your **`external`** folder, create a new file named `.gitignore`
  - Add node_modules to the .gitignore

```
node_modules
```

- When done, copy the `.gitignore` file to the `internal` folder
  - Both the `internal` and `external` folders need a copy of `.gitignore`

# Using Git

- The following slides provide steps on using GitHub
- Additionally, the last two optional slides show how to use Google Cloud Source repos
  - You could choose to add Google Cloud repos as a second remote

# Using Git (GitHub)

- Prerequisites:
  - Join GitHub if you are not already a member (www.github.com)
  - Create a public repository called `events-app-internal` in your GitHub account
    - Do NOT add anything (e.g., a ReadMe)
    - Make a note of the repo address (copy and save it somewhere)
  - Create a second repository called `events-app-external` in your GitHub account
    - Do NOT add anything (e.g., a ReadMe)
    - Make a note of the repo address (copy and save it somewhere)

# Using Git (GitHub) (continued)

- Create a GitHub personal access token:
  - From the github.com page, in the upper-right corner, click your profile photo, then click **Settings**
  - In the left sidebar, click **Developer settings**
  - In the left sidebar, click **Personal access tokens**
  - Click **Generate new token**
    - Provide a **Note** for your token of `classtoken`
    - Set the expiration to `7 days`
    - Select the **Repo** scope
  - Click **Generate token**
- Copy the generated token and save it somewhere secure. You cannot view it again.
  - You will use the token as your GitHub password when using the **git** command
  - You will need it multiple times

# Using Git (GitHub) (continued)

- Switch to the browser with Google Cloud Shell
  - Open a new Cloud Shell tab by clicking the + button
  - In the terminal window, change to the **`events-app`** folder
  - Execute the following commands:
    - `git config --global user.email "your_email_on_github"`
    - `git config --global user.name "your_github_user_name"`
    - `git config --global init.defaultBranch main`
    - Verify with: `git config --global --list`

# Using Git (GitHub) (continued)

- In the same Cloud Shell tab, change to the **`internal`** folder and run:
  - `git init`
  - `git add .`
  - `git commit -m "Initial commit"`
  - `git remote add origin` *your-git-internal-repo-address*
  - `git push -u origin main`
    - You will be asked for your GitHub user id and password (Use your token as the password)
- Change to the **`external`** folder and run:
  - `git init`
  - `git add .`
  - `git commit -m "Initial commit"`
  - `git remote add origin` *your-git-external-repo-address*
  - `git push -u origin main`

# Making Changes to Code

- Go make a change to your code
- In Cloud Shell, change to the service folder with the change (**`internal`** or **`external`**)

```
git add .
git commit -m "My first change"
```

  - This commits it to your local repo
  - The remote repo has not been updated yet
- `git push origin main`
  - This pushes the changes to the main branch of the remote repo named origin

# Implementing Twelve-Factor App

- As a group, revisit how case study application conforms to the twelve factors
  - Refine the slide in your Google Slides document

# Caching your Username and Token

- These steps are completely optional and potentially dangerous
  - **Warning**: Performing these steps stores your git account password in the global .git-credentials file
- In Cloud Shell, run the following command:
  git config --global credential.helper store
- After running this command, the first time you pull or push from the remote repository, you'll get asked for the username and password/token
  - For future commands you don't have to provide the username and password/token
  - The credentials are storage in ~/.git-credentials file

# Success!

- **Congratulations**! You have successfully stored your code in git

# Optional: Using Git (Google Cloud Source Repos)

For Reference – SKIP FOR NOW

- Switch to the browser with Google Cloud Shell
- Open a new Cloud Shell tab by clicking the + button 
  - Change to the `sample-master` folder and execute the following commands:
  - `export PROJECT=$(gcloud info --format='value(config.project)')`
  - `git config --global user.email "(gcloud config get-value core/account)"`
  - `git config --global user.name "Your-Name-Here"`
- Create two source repos (for internal and external):
  - `gcloud source repos create events-app-external`
    - Type Y if asked to enable API
  - `gcloud source repos create events-app-internal`
- Configure Git to use `gcloud` for authentication
  - `git config credential.helper gcloud.sh`
- You have just created two source repos on Google Cloud and configured Git
  - On the next slide, you will save your code to the appropriate repo

# Optional: Using Git (Google Cloud Source Repos) (continued)

For Reference – SKIP FOR NOW

- In the same Google Cloud Shell tab:
  - Change to the `internal` folder
  - `git config credential.helper gcloud.sh`
  - `git commit -m "Initial commit"`
  - `git remote add second https://source.developers.google.com/p/$PROJECT/r/events-app-internal`
  - `git push -u second main`
- Change to the `external` folder
  - `git config credential.helper gcloud.sh`
  - `git commit -m "Initial commit"`
  - `git remote add second https://source.developers.google.com/p/$PROJECT/r/events-app-external`
  - `git push -u second main`