    - be in your dev machine, e.g. a fresh VPS or container (recommend Ubuntu 20+ with >=2 GB RAM)
    - clone your remote repo to your dev machine
    - `cd` into the project folder
    - install rbenv with ruby-build: `curl -fsSL https://github.com/rbenv/rbenv-installer/raw/HEAD/bin/rbenv-installer | bash`
    - reload profile: `source ~/.bashrc`
    - install ruby 3.x.y: `rbenv install 3.x.y`, where 3.x.y is the version of ruby specified in the Gemfile, or 3.2.2 if none.
    - set ruby 3.x.y as the local default version: `rbenv local 3.x.y`
    - install bundler: `gem install bundler`
    - configure bundler to skip production gems: `bundle config set --local without 'production'`
    - install dependencies: `bundle install`
- install heroku cli: `curl https://cli-assets.heroku.com/install-ubuntu.sh | sh`
- login to heroku: `heroku login -i`
    - `username: <your username>`
    - `password: <your API key>`
        - get your API key from your heroku account
- create an app on heroku: `heroku create [appname]`, where `[appname]` is an optional name for the app
- create an s3 bucket
- create iam role for app to access s3 bucket
    - take note of the access key id and secret access key
    - create access policy: in your iam s3 user, under permissions, click add permission, then create inline policy
        - choose `s3` as the service
        - specify the actions allowed:
            - `ListBucket`
            - `PutObject`
            - `GetObject`
            - `DeleteObject`
        - specify bucket resource ARN for the ListBucket action: click add ARN to restrict access
            - put name of your s3 bucket in the bucket name field
        - specify object resource ARN for the PutObject and 2 more actions:

            - put name of your s3 bucket in the bucket name field
            - click any next to object
        - click review policy at the bottom
        - make sure it looks right and then create it
- in `config/storage.yml`, make sure `region` and `bucket` fields match your bucket's region and name
- Inside the codebase in app/controllers/items_controller, edit the upload_to_s3, and change the name 'campuscloset' to the name of your s3 bucket.
- in `config/storage.yml`, make sure `region` and `bucket` fields match your bucket's region and name
- create google oauth2 client id:
    - go to google cloud apis & services
    - if you've never been here before, you'll need to make a project first and configure your oauth consent screen
        - make the project internal
        - only fill in the required fields:
            - name: your app's name
            - email: your email
            - authorized domains: your apps domain, e.g. `appname.herokuapp.com`
            - developer contact info: your email
        - go to credentials, then click create credentials at the top and select oauth client id
            - application type: web application
            - name: your app's name
            - authorized redirect uris, add: `https://appname.herokuapp.com/auth/google_oauth2/callback`
            - take note of the client id and client secret
- Create Mapbox API token
    - https://docs.mapbox.com/help/getting-started/access-tokens/
    - Create an account and find your token on your profile page
    - https://account.mapbox.com/
        - Using the default public token should be fine
- remove encrypted credentials that you cannot decrypt: `rm -f config/credentials.yml.enc`
- `Create a .env file and add the following credentials for local use:`
    - AWS_ACCESS_KEY_ID=<your aws token>

    - AWS_SECRET_ACCESS_KEY=<your aws key>
    - GOOGLE_CLIENT_ID=<your google client id>
    - GOOGLE_CLIENT_SECRET= <your google client secret>
    - MAPBOX_ACCESS_TOKEN= <your mapbox access token>
- For deployment use, put the same environment variables into heroku like so:
    - https://devcenter.heroku.com/articles/config-vars
    - They can be added from the command line by running:
    - `heroku config:set AWS_ACCESS_KEY_ID=8N029N81 AWS_SECRET_ACCESS_KEY=9s83109d3+583493190`
- configure email account for sending emails
    - use gmail (because why not?)
    - create an app password
- set sendmail config vars on heroku
    - `heroku config:set SENDMAIL_USERNAME=the email address you just created/configured`
    - `heroku config:set SENDMAIL_PASSWORD=the app password you just created`
    - `heroku config:set MAIL_HOST=https://appname.herokuapp.com`
- setup the database: `rails db:migrate`
- prepare the test database: `rails db:test:prepare`
- run rspec tests: `rails spec`
- run cucumber tests: `rails cucumber`
    - `Some tests require selenium which might require chrome driver`
    - `To install the necessary items, follow these steps`
    - `https://www.gregbrisebois.com/posts/chromedriver-in-ws12/`
- stage changes: `git add .`
- commit changes: `git commit -m "ready to push to heroku"`
- deploy to heroku: `git push heroku master`
- run migrations on heroku: `heroku run rails db:migrate`

- seed database on heroku: `heroku run rails db:seed`
- poke around the deployed app
- don't forget to also push to your own github repo: `git push`