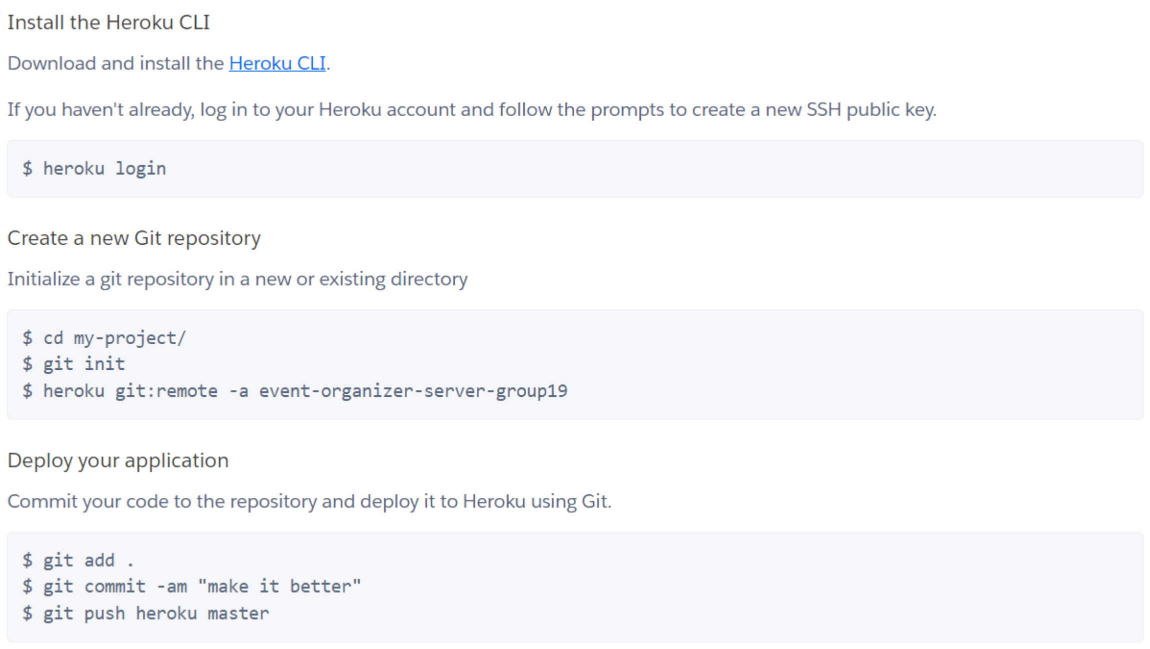
Platform
Heroku
Required components:
1. How to set up the server, like which remote machine to use?
2. How to initiate the database there? Like the SQL server or what other setups you need to do.
3. What are other frameworks you need to install?
4. How to do the testing? Black box testing, white box testing, and unit testing...
5. How to make your application publicly accessible? Like what is the URL to access it, and how to run this?

1. Cloud platform: Heroku. https://www.heroku.com/platform. Need to sign up for an account. Then, create a new app. Next, download Heroku CLI (https://devcenter.heroku.com/articles/heroku-cli)to your computer and follow the steps below:



Install the Heroku CLI

Download and install the Heroku CLI.

If you haven't already, log in to your Heroku account and follow the prompts to create a new SSH public key.

```
$ heroku login
```

Create a new Git repository

Initialize a git repository in a new or existing directory

```
$ cd my-project/
$ git init
$ heroku git:remote -a event-organizer-server-group19
```

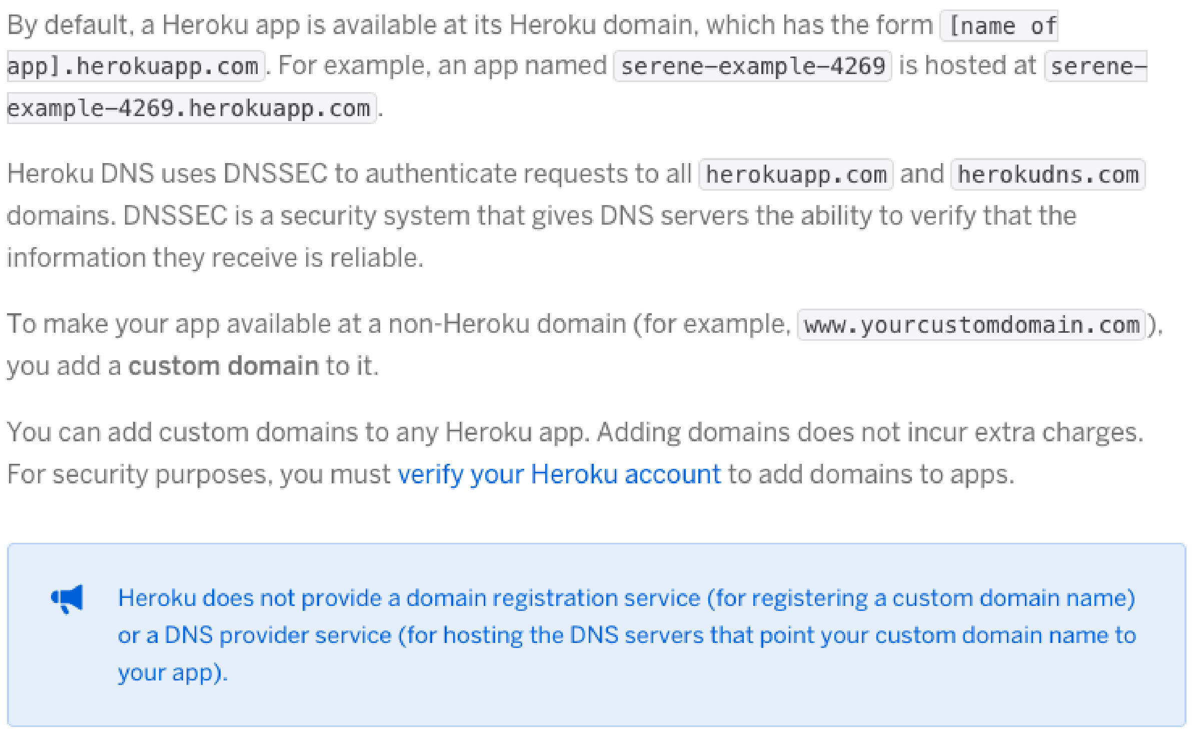
Deploy your application

Commit your code to the repository and deploy it to Heroku using Git.

```
$ git add .
$ git commit -am "make it better"
$ git push heroku master
```

2. If the app writes data to its local filesystem for persistent storage (including to a local SQLite database), on Heroku it must instead write that data to one of the following: A managed database service (MySQL with ClearDB in our case) A managed object storage service (such as Amazon S3)
There's no need of mysql script to initialize the database. Database is provided automatically when you use their database services such as ClearDB. Tables are generated automatically by spring boot server.
3. We need to have ClearDB, Maven 3, and Java 8 installed
    a. For more info: https://devcenter.heroku.com/start
4. Black/White box testing will be implemented by directly using the url created by Heroku. Unit testing will be implemented in the test folder in our maven project.

5. A URL would be provided so that others can have access to the application. Using Heroku, we could either use the default URL (“`[name of app].herokuapp.com`”) or we could create a custom domain through “`heroku domains:add`” (https://devcenter.heroku.com/articles/custom-domains). Based on this decision, the user can then enter the url to view the application.



By default, a Heroku app is available at its Heroku domain, which has the form `[name of app].herokuapp.com`. For example, an app named `serene-example-4269` is hosted at `serene-example-4269.herokuapp.com`.

Heroku DNS uses DNSSEC to authenticate requests to all `herokuapp.com` and `herokudns.com` domains. DNSSEC is a security system that gives DNS servers the ability to verify that the information they receive is reliable.

To make your app available at a non-Heroku domain (for example, `www.yourcustomdomain.com`), you add a **custom domain** to it.

You can add custom domains to any Heroku app. Adding domains does not incur extra charges. For security purposes, you must verify your Heroku account to add domains to apps.

Heroku does not provide a domain registration service (for registering a custom domain name) or a DNS provider service (for hosting the DNS servers that point your custom domain name to your app).

https://devcenter.heroku.com/articles/git
Heroku Partners with GitHub to Offer Student Developer Program | Heroku

Deployment Document (due November 13)

This document should include detailed instructions of what steps to take in order to deploy the code. If any outside libraries, specific server settings, or other elements are needed, this is where to explain those to your grader.

Total Grade	1.0%
0.5%	Step-by-step instructions for getting the application running in a new environment are comprehensive and seem complete
0.5%	Formatted professionally with no grammatical mistakes