Problem

- Need: Extend Augur's new frontend and update the getting started document.
- Purpose: Users can view more health metrics about a project and a clearer getting started document of Augur.
- Client Base: Students, developers, researchers, and data scientists.

Deliverable

- Show the number of pull requests on Augur's new frontend.
- Show more health and security metrics about repositories or groups on Augur's new frontend.
- Make an outline for what we plan to change in the getting started document.

Diagrams

- Create design diagrams like the ones from class. Our first step is to find a software program that we can use to draw these diagrams: https://www.diagrams.net/
- Each individual contributes to all diagrams, through independent design, design collaboration, and careful peer review of finished diagrams.
- Compile the diagrams into the group document.
- Each diagram should have a title and caption describing what it shows.

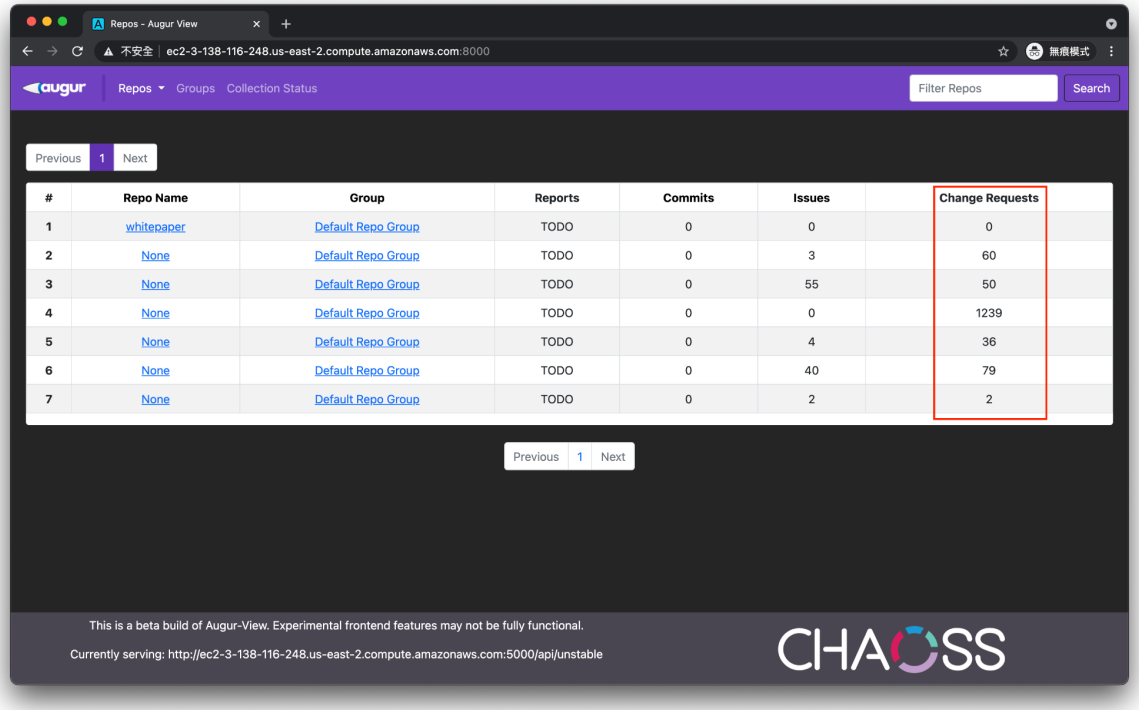
System Constraints

- Web server software.
- Database management system.
- Physical servers with compatible OS installed to host the webserver and DBMS with network and internet connectivity.
- Client computers with a web browser and Internet access.

Users / Activities / Relevant Data / Constraints

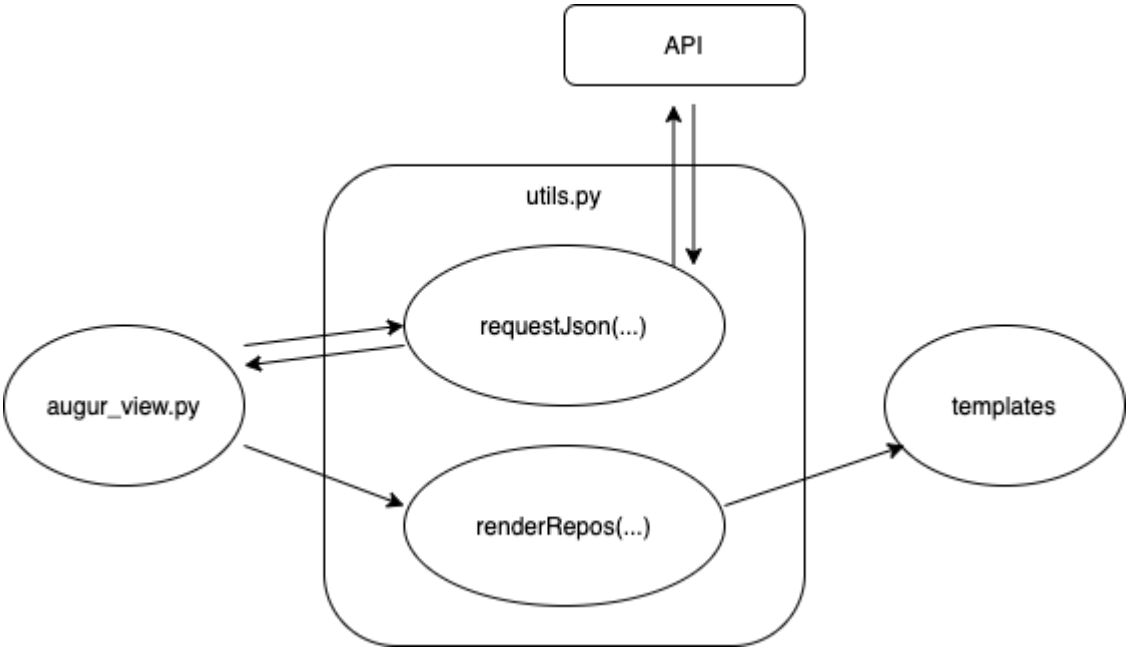
- Users can view Augur backend: http://ec2-3-138-116-248.us-east-2.compute.amazonaws.com:5000/api/unstable
- Users can view API about repositories: http://ec2-3-138-116-248.us-east-2.compute.amazonaws.com:5000/api/unstable/repos
- Users can view API about groups: http://ec2-3-138-116-248.us-east-2.compute.amazonaws.com:5000/api/unstable/repo-groups
- Users can view API about repo_info: http://ec2-3-138-116-248.us-east-2.compute.amazonaws.com:5000/api/unstable/metadata/repo_info
- Users can view Augur's new frontend: http://ec2-3-138-116-248.us-east-2.compute.amazonaws.com:8000/
- Users can view the hello world page: http://ec2-3-138-116-248.us-east-2.compute.amazonaws.com:8000/hello-world
- Users can view the number of pull requests on the repos-table page.
- Users can view more health and security metrics about repositories and groups on Augur's new frontend.
- Users can view a clearer getting started document about Augur.

Show the number of pull requests on the frontend (Hello World version of our project)



Architecture

- Get API data using the requestJson("metadata/repo_info") function.
- Store the information about repositories and feed them to templates.
- Render the data on templates.
- Displaying other health and safety metrics can be done in similar steps, but using different APIs.



Read the number of pull requests from a JSON file and add that info to the table

Name Column	Info Column	Info Column	Info Column	Pull Requests
Name	Info	Info	Info	5
Name	Info	Info	Info	12
Name	Info	Info	Info	1,262

Update the getting started document - Ahmad Imtiaz

For setting up the DevOps environment the instructions use the Ubuntu 20.4 version, rather than the 18x version which is supported no longer because of the Python 3. The Ubuntu 20.04 is preferred for the PostgreSQL Installation because it has long-term support. It can also be available for the 18x versions so that the availability is gigantic. For the Git Configuration, every platform should have the command line login to cache the Git Credentials for the LINUX user who is operating the Augur. By performing this step the Facade Commit Counting Diesel is prevented from stalling the command line prompt when the repositories move or when they disappear.

The limitation for the Python Virtual Environment Configuration is that the software version of the python should be more than 3.8, or else it won't support it. The Augur installation is only done on the macOS, Ubuntu, and the Fedora. But it does not support the windows directly, for the installation in windows, it is required to use the docker images, or else the user should install the virtual machine with the supported operating system. For the installation from the source, PostgreSQL should be set up for the storage of the data collected by Augur. After that, the Augur application server should be installed and configured. The Augur data collected workers should be installed and configured.

For the deployment of the augur’s frontend, we must first run the augur config init-frontend with the python environment. The user has to enter the Augur’s home directory and run the npm install and then the npm run build in the frontend directory. The good thing is that the command line interface is provided in the Augur for interaction with the Augur installation.