Plugging Docker-based visualizations into Django with \texttt{django\_docker\_engine}

Chuck McCallum, Scott Ouellette, and Nils Gehlenborg
Department of Biomedical Informatics, Harvard Medical School

Problem
Big applications don’t need more complexity:
- Visualizations may have own backends
- Unit tests for tool lost in tests for application
- Different stacks, or dependency conflicts

One-off visualizations are hard to maintain:
- How to install?
- How is input provided?
- How do I authenticate users?

Solution

Developers need to configure how Docker and Django will interact. The method Refinery uses is in \textbf{bold}.

Where should containers run?
- On the same host as django
  - ... but this could cause contention for limited resources
- On a separate host
  - ... but this is another host to set up

How to pass input?
- As mounted JSON file
  - ... but this requires SSH access
- As environment variable containing JSON
  - ... but this may hit a shell size limit
- As a URL pointing to JSON
  - ... but this requires you to create new URLs

How should routing work?
- By container name in URL path
  - ... but this requires that all URLs be relativized
- By container name in subdomain of URL host
  - ... but this requires appropriate wildcard DNS to be set up

Additional constraints
Refinery imposes additional constraints on the format of the input.json, but these are not enforced by \texttt{django\_docker\_engine} itself.

Is \texttt{django\_docker\_engine} a good tool for the job?
Probably yes, if:
- Stateless
- Many different visualizations
- Loosely coupled to application
- Read-only
- Visualizations require their own backends
- Medium levels of use, and just a few different inputs

Perhaps not, if:
- Stateful
- Just one visualization
- Tightly coupled to application
- Read-write
- JS-only (overkill)
- High levels of use, with many different inputs

Limitations/TODOs
- Our proxy server does not support WebSocket connections: R Shiny apps and Jupyter notebooks do not work.
- Only one docker container is launched at a time: No provision for breaking visualizations into microservices with Docker Compose.
- Visualization authors need to make sure that AJAX requests preserve session cookies from Django.
- Demo works only on Mac.
- The input.json structure is overly restrictive.
- We would like to use ECS or Kubernetes as the backend, but have not demonstrated this.
- No provision for persisting work done in the container.