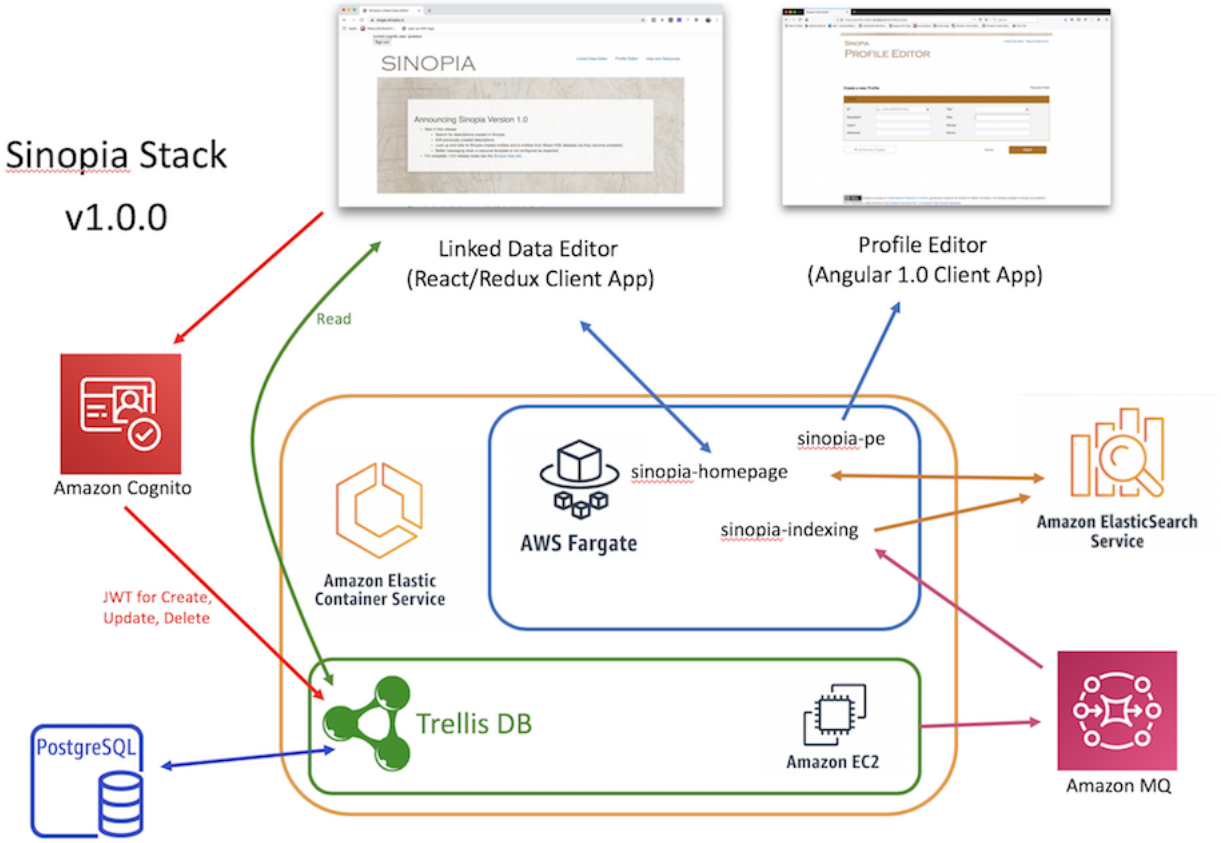
2019 BIBFRAME Workshop - Stockholm, Sweden

https://ld4p.github.io/bf-workshop-2019/sinopia-stack-aws/

Running the Sinopia Stack on Amazon Web Services



Background

An early requirement for the open-source Sinopia project was to build a cloud-based collaborative editing environment. The team choose Amazon Web Services, one of the most popular commercial cloud provider, by utilizing a number of different AWS services and products to host Sinopia and its dependent technologies. Although the first version of Sinopia is closely tied to specific AWS cloud services, we are looking at more generic infrastructure options for hosting Sinopia on other commercial cloud services as well as in hybrid environments.

Trellis - Linked Data Platform

In Sinopia's early analysis, we determined that having a RDF triplestore was not necessary for meeting the requirements of a create-update-read-delete (CRUD) editor for RDF.

Amazon Web Services

Cognito

To handle authentication and authorization for create, update, and delete operations on the Linked Data RDF and JSON resource templates in Sinopia, we are using the AWS Cognito secure user sign-up and access control service.

Amazon Web Services

Elasticsearch Service

After a user successfully signs-up via the Amplify SDK, a JSON Web Token.

Top

Amazon Web Services

Elastic Container Service

The deployment of Sinopia on AWS relies on pre-built Docker images hosted on DockerHub that are then run in a Elastic Container Service (ECS) cluster. Sinopia is run on three ECS clusters; development, staging, and production with corresponding Docker images for each environment.

Top

Amazon Web Services

Fargate

Fargate is an AWS service that runs Docker containers in a state-less fashion with in Virtual Elastic cluster. A Docker is a lightweight running linux process that is generated in a deterministic way from a Docker image

In Sinopia, we run the following Docker Images as Fargate tasks:

- Sinopia Linked Data Editor Docker image
- Sinopia Profile Editor Docker image
- Sinopia Indexing Pipeline Docker image
- Sinopia ACL Docker image

Top

Amazon EC2

The other method we use to run a Docker image is by hosting as part of a virtual machine running as a EC2 Docker container.

For the initial `1.0.0` release, a simple search index is indexed and searched through the AWS hosted Elasticsearch Service. Elasticsearch is a full-text search engine, based on Lucene, that has been optimized for running on Amazon's cloud.

Amazon Web Services

Apache ActiveMQ

Trellis publishes events like creating, updating, or deleting resource as they occur to a AWS Messenging Queue that is monitored by a process called `sinopia-indexing` pipeline that then updates the Elasticsearch search index.

```
message = {
  payload: '2345'
}
```

Top

Amazon Web Services

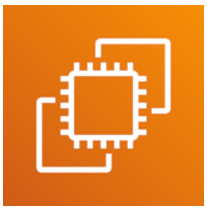
Relational Database Service (RDS)

Sinopia server's foundation is a variant of Trellis Linked Data Platform that uses a PostgreSQL relational database to manage the Sinopia Editor's RDF and JSON payloads instead of a RDF Triplestore.

The AWS RDS services offers the following advantages:

- Significantly cheaper than AWS Triplestore Neptune
- Automatic backups
- Administrative overhead reduced

Top



When we started Sinopia, Fargate does not persist data so we could not mount a permanent disk volume to store the RDF entities Memento metadata as needed by our Linked Data Platform, Trellis.

Top