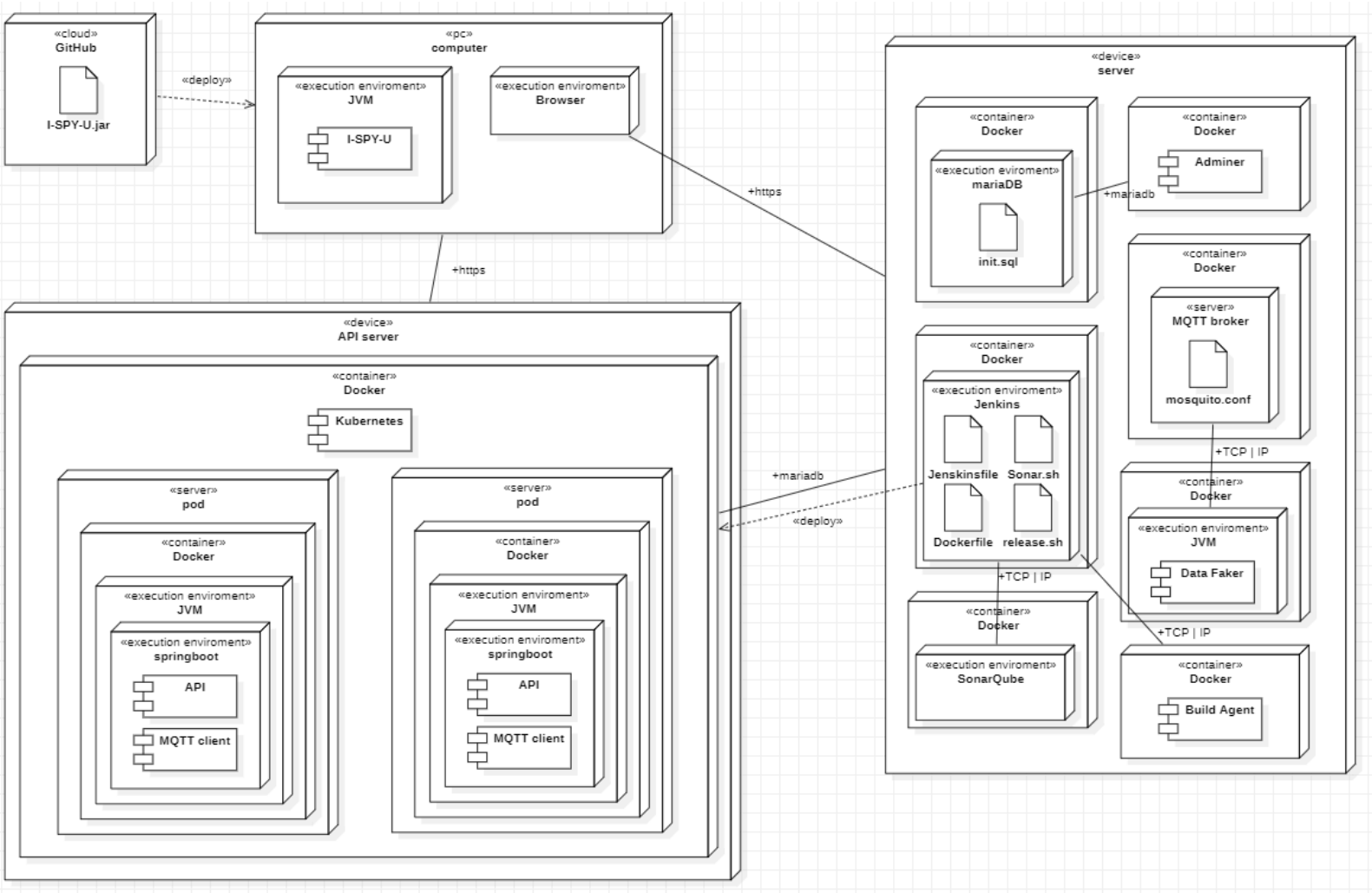
OTP Deployment diagram

Eino Ruuth, Niko Meriluoto, Toni Hirvikallio ja Joonas Karppinen

development environment



Server 1 - server:

Has Jenkins for CI/CD, SonarQube for Code Analysis, MariaDB for DB, Adminer for DB panel, Mosquitto MQTT broker, Java Data faker for pseudo data, JDK/Maven build agent

Production would have MQTT broker, MariaDB and Adminer probably

Jenkins has connection to SonarQube for code analysis, build agent for building and to the Kubernetes in server 2 for deployment, it also technically accesses GitHub for pulling latest code, releasing built artifact and Docker Hub for pushing built containers but these were left out of the diagram.

Data Faker has access to MQTT broker for creating pseudo data, in production it would be the sensors accessing the broker.

Server 2 - API server:

This environment runs Kubernetes in Rancher Kubernetes in Docker setup, it has 2 pods with currently one docker container inside per pod. These are the REST API serving the client and have access to the MariaDB and MQTT broker. The MQTT broker is only accessed by the MQTT client module.

Technically there is also an nginx Docker container proxying all network access to Kubernetes but it felt unnecessary clutter to include as would TomCat server also serving the Spring Boot.

PC:

Runs the client program and in development environment the needed tools, IDE with access to SonarQube, browser to manage Jenkins, SonarQube and Adminer panels.
It also gets the built release from GitHub releases deployed to it.

GitHub:

GitHub has the code repo and published release artifacts for the client

Docker Hub:

Docker Hub is used for storing the built containers: REST API, data faker, client with VNC and client with X display connection