Shepherd - course CI/CD

Participants

- Developer: Sami, Mikko
- Technical instructor: Jaakko
- Customer: Markku (EDIT, A+ admins)
- Additional work:
    - Mikko Hakala - consults operations perspective
- Weekly: -

Process description

No formal team, hence open process between EDIT and CS IT

Project description

Course material CI/CD pipeline for A+ and other LMSes (e.g. mycourses). The project cosists of few parts:

- Roman: Python lib and CLI tool to compile with local docker. Possibility to develope a GUI app for teachers and include `./docker-up.sh` parts. Conseptually same as `drone exec` - https://github.com/apluslms/roman
- Shepherd: Web service and the core project. - https://github.com/apluslms/shepherd - Shepherd cositst of three major areas:
    1. User and course management. Users are stored with tress, so we can model real organisation. Supports defining rules like "if you are member of cs-staff, you can create new courses with prefix `cs-*`". Additionally, teachers can manage other teachers part of the course (i.e. write persmission)
    2. Automatic course compilation as reaction to changes in the remote git: 1) make a local copy of the data (protection against removing repo or removing access), 2) compile the course material using containers 3) upload configurations to different services and/or configure them
    3. Grant permissions for course build steps to manage their virtual area in different services

Problems the project aims to solve:

- Remove administration tasks by allowing course creation and management for the teacher themself
- Manage SSH keys, thus removing requirement to add Teemu Lehtinen to the version.aalto.fi project. Additionally, allows using other git remotes.
- Supporting more complicated course compilation by using containers for steps
- Scaling course compilation by using Kubernetes
- Make it easier to add more assessment tools (currently it's only easy to use a single mooc-grader, which blocks migration to new installations and services)

Project goals:

- step 1 - replace mooc-grader/gitmanager, requirements:
    - web UI
    - pipeline runner: we have celery prototype, look into better alternatives
    - roman for running or creating configuration for workflow engine
    - deployment containers: work done on static file uploading client and server. container for coniguring A+ is required
- step 2 - more than that

Status (done):

- User management in Shepherd (Proto/ok)
- Course managment in Shepherd (Proto/ok)
- Shepherd uses celery to download git material, git subtree to create a build workingtree and to run Roman to build it
- Roman can build courses using docker in local machine using filesystem mount
- We have static material upload flask application (files can be shared with nginx)
- We have deployment container to upload to the flask app
- PoC works with roman building, uploading static files to docker-compose network including static server and files can be accessed from there

To-do

- See GitHub issues
- Sami and Mikko will look into container/CI/workflow engine:
    - Argo - https://argoproj.github.io/
    - Drone.io - https://drone.io/
    - Tekton - https://cloud.google.com/tekton
- Migrate relevant cards from Trello - https://trello.com/b/pCNgbxyI/shepherd and https://trello.com/b/rbkEpxTp/roman