|  |  |  |
| --- | --- | --- |
| ITU Logo | INTERNATIONAL TELECOMMUNICATION UNION**TELECOMMUNICATIONSTANDARDIZATION SECTOR**STUDY PERIOD 2017-2020 | FG-AI4H-E-008 |
| **ITU-T Focus Group on AI for Health** |
| **Original: English** |
| **WG(s):** | N/A | Geneva, 29 May – 1 June 2019 |
| **DOCUMENT** |
| **Source:** | AHG Benchmarking Platform |
| **Title:** | Status Report of the Ad hoc Group “Benchmarking Platform” |
| **Purpose:** | Discussion |
| **Contact:** | Markus WenzelFraunhofer Heinrich-Hertz-InstitutGermany | Email: markus.wenzel@hhi.fraunhofer.de |
| **Contact:** | Oguz ErogluAda Health GmbHGermany | ‎Email: oguz.eroglu@ada.com |
| **Contact:** | Henry Hoffmann Ada Health GmbHGermany | Email: henry.hoffmann@ada.com |

|  |  |
| --- | --- |
| **Abstract:** | The ad hoc group “Benchmarking Platform” studies the suitability of *AIcrowd* to serve as benchmarking platform software. This document contains a progress report for the period April to May 2019. |

# Introduction

This ad hoc group studies the suitability of *AIcrowd* to serve as benchmarking platform software and summarizes the current findings (April and May 2019) in this document. At the Shanghai meeting of the Focus Group on Artificial Intelligence for Health (FG-AI4H), a contribution from AIcrowd was discussed ([FGAI4H-D-011](https://extranet.itu.int/sites/itu-t/focusgroups/ai4h/docs/FGAI4H-D-011.docx)). This contribution proposes an open-source benchmarking platform that can be used by the Focus Group. Upon initial inspection, the benchmarking platform “AIcrowd” appeared to be in conformance with the requirements for a benchmarking platform for the Focus Group, as detailed in the “Call for proposals of open-source software that enables the FG-AI4H to run the benchmarking procedure on computing infrastructure of the United Nations” ([FGAI4H-C-106](https://extranet.itu.int/sites/itu-t/focusgroups/ai4h/docs/FGAI4H-C-106.docx)). The ad hoc group would like to thank AIcrowd for their offer to allow the focus group to use their open source code.

# Methods

According to the mandate ([FGAI4H-D-035](https://extranet.itu.int/sites/itu-t/focusgroups/ai4h/docs/FGAI4H-D-035.docx)), the ad hoc group inspected the information on <https://www.aicrowd.com/> and the software code and documentation on <https://github.com/AIcrowd/AIcrowd>, tested the software, examined the capabilities and limitations of the software, and determined the implications for the computing infrastructure.

# Results

## Tests

The AIcrowd platform comprises the system components “Webapp”, “Client”, “Grading Server”, “Storage”, and the standard repository manager “Gitlab” (according to the current documentation on <https://github.com/Aicrowd/Aicrowd/blob/master/doc/architecture.md>).

The ad hoc group managed to run the AIcrowd *Webapp* / front-end application and Gitlab in order to find out how the AICrowd platform works. Due to the limited documentation and time, it was not feasible yet to test the other system components "Client", "Grading Server", and "Storage”.

The Webapp is written in Ruby on Rails and available from the AIcrowd GitHub repository. The AHG has used a Virtual Private Server with Fedora Linux version 30 to install AIcrowd (from the master branch). The installation instructions are available on the GitHub repository in the form of markdown documents. The AHG found that several additional steps were not documented, it has prepared a more complete set of installation instructions in the appendix of this document.

In addition, the AIcrowd platform requires a GitLab instance for submission acceptance, which is an open-source platform that provides functionality similar to GitHub, but can be self-hosted. AIcrowd uses the GitLab instance to accept new submissions. Installation instructions (on CentOS Linux version 7) are also documented in the annex of this document.

## Capabilities

First, the ad hoc group would like to acknowledge the achievements and track record of AIcrowd, which is a spin-off company from the university EPFL:

* Having successfully run machine learning challenges at workshops of the (elite) NIPS/NeurIPS conference (<https://nips.cc/>), where cutting edge machine learning research is presented.
* AIcrowd takes defense measures against malicious predict routines or cheating with the execution in a closed environment and cleaned error messages. The functionality to evaluate the AI models in a closed environment brings them ahead of the well known AI evaluation systems Kaggle owned by Google or Topcoder, as it becomes evident in the comparison by their competitor "EvalAI": Yadav et al. (2019). EvalAI: Towards Better Evaluation Systems for AI Agents. *arXiv*:1902.03570. <https://arxiv.org/abs/1902.03570>
* Working together with companies such as Swiss Federal Railways and Unity.

## Limitations

As noted on their Github page, the documentation is “work in progress, and portions may be incomplete or out of date”. Understandably, documentation for non-members is not of highest priority for a startup. However, with the given instructions, installation was difficult/impossible. (Feedback from the ad hoc group on the installation instructions was included in the documentation.) While example challenges are published, setting up a challenge/benchmark is not really explained so far yet.

Apparently, an additional Kubernetes orchestrator is required in addition to the documented system components, which is currently *not published as open-source software*. The AIcrowd CEO Mohanty Sharada has indicated that it would be *possible to share* the Kubernetes files with the Focus Group. (This Kubernetes orchestrator monitors the GitLab instance for new submissions, which come in the form of new commits to repositories that contain a Docker image. Once a new submission is detected, the orchestrator downloads a copy of this submission, and evaluates the model on the testing data, which are stored on the server that runs the Kubernetes instance. The Kubernetes orchestrator is designed to have a very limited output, it connects to another server only to publish the results of the evaluation.)

AIcrowd has been and is hosting challenges at NeurIPS, which can be considered as ideal security test bed, because of the fierce competition at this high profile conference and the monitoring by global machine learning experts. AIcrowd takes crucial defense measures against malicious predict routines or cheating with the execution in a closed environment and cleaned error messages. Still, it could be worthwhile that third party cybersecurity experts review and audit the system at some point.

Finally, the following issues were regarded as in part unclear:

* AIcrowd originates from the project/repository CrowdAI that contains some crucial APIs such as the WebApp client API (<https://github.com/crowdAI/crowdai_api>) and the Grader client API (<https://github.com/crowdAI/crowdai-client-py>). These client applications do not exist in the AICrowd repository and it's not clear if they are still being used or not. It is also not clear if these APIs are designed for which part of the overall user story (are they for the challenge creator or the participants?).
* It is not clear if we are able to modify the reports for our needs, such as grouping the results by certain diseases/countries/ethnicities etc. Given the defined input/output objects of the Grader API, the output does not contain application level specific details, but only the primary/secondary scores.
* There seems to be no support for mocking API\_KEYS in order to test the Grader and Client architectures in a local environment before launching the project. It seems that these applications are trying to connect deployed servers with actual API\_KEYS. This makes it hard to start developing before actually receiving an API KEY/Challenge ID from AICrowd. If there's a dev environment support, it's not clear how to achieve this, given lack of documentation.

## Implications for the computing infrastructure

Customization from public cloud services (i.e. Amazon) to on-premises computing infrastructure of the Focus Group seems to be possible, but requires an unknown investment of work as becomes evident from the documentation (“This architecture could be replicated by other parties on private or public architectures, in line with the crowdAI open source approach. Eventually an install / developer guide will be written.” “Amazon S3 is used to store files required for the application. Note that this could be replaced with a S3 compatible file store, such as OpenStack Swift S3.”).

# Summary

Software of AIcrowd is an option to serve as benchmarking platform software for the Focus Group, given the proven track record and the functionality to evaluate AI models in a closed environment. The limitations and preliminary test results discussed above must be taken into account, and it should be noted that the software requires support and customization before it would be ready for on-premises computing infrastructure of the focus group.

Appendix A:
Installation instructions for the benchmarking platform “AIcrowd” on a Virtual Private Server (VPS) with a public IP address running Fedora Server version 30

AIcrowd provides concise installation instructions in the documentation in the form on markdown (GFM) in the main repository (<https://github.com/AIcrowd/AIcrowd>). This appendix is based on those installation instructions and aims to provide some additional clarifications. AIcrowd is a web application (webapp) that is mostly written in Ruby on Rails (RoR / Rails). It relies on a large number of dependencies called “gems”, all of which are open source and can be installed using package managers such as “npm” (nodejs) or “yarn”.

Given the high number of dependencies, with very specific version requirements, it is likely that general installation instructions would result in specific version issues, therefore these installation instructions are specifically for Fedora Server 30. The same procedure has also been tested on Ubuntu 18.04, the difference being only in the package managed (i.e. apt instead of dnf/yum) and the fact that Fedora requires extra steps to setup PostgreSQL.

The following installation instructions use SSH into a clean Fedora 30 VPS (DigitalOcean was used), it is assumed that the SSH pubkey of the local machine was already added to the root account of the VPS using the Digital Ocean setup web interface. These installation instructions were tested on 13 May 2019 with the main branch of AIcrowd.

SSH into the VPS at the root account using (replacing XXX.XXX.XXX.XXX with the IP address of the VPS)

ssh root@XXX.XXX.XXX.XXX

Create a new user account (here newuser).

adduser newuser

Set a password for newuser.

passwd newuser

A new password will be asked for twice.

Make the new user a superuser (add to wheel).

usermod -aG wheel newuser

Switch to the new user account.

su - newuser

Update the system

sudo dnf update

The sudo prefix will require the password (of newuser) to be entered.

Install rvm (Ruby version manager) using:

gpg2 --recv-keys 409B6B1796C275462A1703113804BB82D39DC0E3 7D2BAF1CF37B13E2069D6956105BD0E739499BDB

Note that the exact key value may change, for the up-to-date key see the command at <https://rvm.io/>.

Next install rvm.

curl -sSL https://get.rvm.io | bash -s stable

Clone the AIcrowd repository using

git clone https://github.com/AIcrowd/AIcrowd.git

Move the working directory to the AIcrowd directory

cd AIcrowd/

Set the version of Ruby to use to 2.5.5.

rvm install 2.5.5

It may be required to exit the newuser session using Control + D and to login again using:

su - newuser

Install postgresql (note these package have different names on different distributions such as Debian / Ubuntu).

sudo dnf install postgresql-server postgresql-contrib

Enable postgresql.

sudo systemctl enable postgresql

Setup postgresql.

sudo postgresql-setup initdb

Start postgresql.

sudo systemctl start postgresql

Switch to the postgres user and open psql.

sudo -u postgres psql

Create a new database.

CREATE DATABASE crowdai\_development;

Exit the postgres psql session using Control + D.

Install npm.

sudo dnf install npm

Install all the npm packages (note that the period is part of the command).

npm install .

Install docker.

sudo yum install docker

Enable and start docker.

sudo systemctl enable docker

sudo systemctl start docker

Run the docker container in the AIcrowd folder.

sudo docker run -p 4572:4572 localstack/localstack

Create fog folder

mkdir ~/fog

Then

db:migrate

db:seed

Then

ENV=development rails server

Appendix B:
Installation instructions for the Git platform “GitLab” on a Virtual Private Server (VPS) with a public IP address running CentOS version 7

One of the requirements of the AIcrowd platform is a GitLab instance for taking submissions. This appendix B outlines the procedure for installing the GitLab platform on a VPS running CentOS (binaries are provided through a repository for the CentOS distribution). These installation instructions use SSH into a clean CentOS 7 VPS (DigitalOcean was used), it is assumed that the SSH pubkey of the local machine was already added to the root account of the VPS using the Digital Ocean setup web interface.

SSH into the VPS at the root account using (replacing XXX.XXX.XXX.XXX with the IP address of the VPS)

ssh root@XXX.XXX.XXX.XXX

Create a new user account (here newuser).

adduser newuser

Set a password for newuser.

passwd newuser

A new password will be asked for twice.

Make the new user a superuser (add to wheel.

usermod -aG wheel newuser

Switch to the new user account.

su - newuser

Update the system

sudo yum update

The sudo prefix will require the password (of newuser) to be entered.

Install several dependencies and tools used during setup.

sudo yum -y install epel-release curl nano policycoreutils-python

The sudo prefix will require the password (of newuser) to be entered.

Install postfix.

sudo yum -y install postfix

The sudo prefix will require the password (of newuser) to be entered.

Enable and start postfix.

sudo systemctl enable postfix

sudo systemctl start postfix

The sudo prefix will require the password (of newuser) to be entered.

Create a repository file for the GitLab repository (and open it in the nano editor).

sudo nano /etc/yum.repos.d/gitlab-ce.repo

In the file paste the following contents:

[gitlab-ce]

name=gitlab-ce

baseurl=https://packages.gitlab.com/gitlab/gitlab-ce/el/7/$basearch

repo\_gpgcheck=1

gpgcheck=1

enabled=1

gpgkey=https://packages.gitlab.com/gitlab/gitlab-ce/gpgkey

https://packages.gitlab.com/gitlab/gitlab-ce/gpgkey/gitlab-gitlab-ce-3D645A26AB9FBD22.pub.$

metadata\_expire=300

Save the file using Control+O and types “yes” + Enter.

Exit the editor using Control+X.

Update the package manager cache.

sudo yum makecache

Install GitLab (replacing XXX.XXX.XXX.XXX with either a public URL or a public IP address).

sudo EXTERNAL\_URL="http://XXX.XXX.XXX.XXX" yum install -y gitlab-ce

Upon successful completion a GitLab logo show be show in the Command Line Interface, at which point, GitLab can be accessed through the public URL/IP address.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_