

# Intermine Boot

An InterMine, commonly referred to as an InterMine **instance**, is one of many biological data warehouses based on the InterMine open source software. They provide a webapp, and a webservice that multiple InterMine clients in different programming languages (eg. Python and JavaScript) can query, to receive integrated biological data.

Building and running an InterMine is an arduous process which requires Linux system administration skills, and provisioned servers if you want your InterMine to be publicly available. InterMine Cloud attempts to solve this and lower the barrier to building and running an InterMine instance.

InterMine Cloud is composed of three main parts. The [wizard](#) provides web pages where you can configure your new InterMine without having to touch a shell or text editor. The [configurator](#) is the backend of the wizard; it receives the HTTP requests communicating how the user wants to configure their InterMine, and creates the necessary configuration files to build it. These configuration files are then used by [compose](#) to create, deploy and manage intermine instances on cloud. In the beginning, we aim to use Google Cloud Platform, but we also wish to support Amazon Web Services and local clouds in the future.

The question in your head now is probably, "where does InterMine Boot fit in this?" The answer to this is that we want to support users who wish to build and run an InterMine locally, without having to use our publicly hosted services. In addition to its role in InterMine Cloud, we also want it to be a convenience tool to easily start test InterMine instances inside Docker containers (to avoid messing with the user's environment) when developing or testing, for instance automated in Continuous Integration.

So InterMine Boot has different use cases. An outline of the use cases are mentioned below.

- Local
  - User downloads intermine\_boot
  - Then user runs "imboot check env"... this will check for the tools it require (docker)
  - User then runs "imboot start / quickstart local"
    - If quickstart then a demo biotestmine will be created
    - If start then intermine\_boot will search locally for previously saved configs and give options to choose from those configs or create a new config
      - If creating a new config is chosen then local instances of wizard and intermine\_configurator will be created and user is redirected to the default browser to complete the config process.
      - .Once a config is created user comes back to cli and stop the local instances of wizard and intermine\_configurator and continue

- Once again user is shown config options to choose from
  - User chooses a config
  - `intermine_boot` then creates a mines using the config and finally redirects the user to the mine
- CI
  - Check for dependencies
    - return error if dependencies not met
  - Load Config or parse option from cmd
  - Download latest container images or the images specified in the config
  - Download the snapshots if user wants to
  - Start containers
  - Wait for the mine build to finish
  - run tests
  - stop the mine
  - compress and upload the data if user wants to
    - we also need to version this data

## New notes

Invocations:

*intermine\_boot start local* - builds and runs biotestmine

*intermine\_boot start local path/to/mine/dir* - builds and runs a mine

*intermine\_boot build local* - builds biotestmine and creates archive of it in cwd as filename

``${project.title}-${project.releaseVersion}.zip``, e.g. BioTestMine-tutorial.zip

*intermine\_boot build local path/to/mine/dir* - builds a mine and creates archive of it ""

*intermine\_boot load local path/to/archive* - loads a previously built archive and runs the mine