

6.4 Student self-assessment

The `checkwork` command allows students to assess their own work against the criteria used by instructors for automated assessment of lab performance. This can be disabled on a deployment-wide basis using the `CHECKWORK no` directive in the `config/labtainers.config` file. Of course this assumes you have separately provided access control over that file, e.g., through use of a custom VM appliance.²⁰

6.4.1 Current state assessment

The lab designer can define a subset of goals and results that inform the student whether the *current* system state is as desired. This greatly differs from typical Labtainer goal assessment, which measure whether the student ever achieved expected results, regardless of the system's current state. These *current state* goals are intended to guide the student with potentially more information than is found (or is practical) in the standard goals. The current state goals are not intended to replace other goals, and they are not displayed to instructors.

The current state goals and results must have a prefix of `cw_`, and they are required to have documentation directives of `CHECK_TRUE` or `CHECK_FALSE`. Text included within a directive will be displayed to students if the value of the associated goals at the time `checkwork` was run does not match the directive value. In the example below the documentation directive will be displayed if the `cw_ssh_open` value is `False`.

```
#CHECK_TRUE: The SSH port is not open.  
cw_ssh_open = client:prestop.stdout : FILE_REGEX : 22/tcp.*open
```

A `CHECK_OK` documentation directive can be added display text in the event that all of the `cw_` goals match their documentation directives.

Current state goals are expected to reflect the current state of the computers as described below.

6.5 Current state artifacts

Results and goals used for current state assessment should primarily be derived from artifacts generated by `prestop` scripts described in 6.1.7. The system uses the most recent timestamp found for any files named by current state results, i.e., those with the `cw_` prefix. The designer can name any file for a current state result – but note it may be difficult to divine current state solely from previous artifacts, e.g., the state may have changed. For this reason, we suggest use of `prestop` scripts.

To highlight the differences between current state assessment and standard Labtainers assessment, consider an example lab that requires the student to enforce an access control policy on a database having several users with differing authorizations. To support the instructor, we'd like to report on whether the student ever managed to configure the database permissions within a single configuration such that all users were prevented from exceeding their authorization and yet were able to access data to which they were authorized. Providing the instructor with point details of whether individual modes of access were permitted or denied at any time in the lab might not be very helpful because the context of such access would not be known. For example, a goal might reflect that John was denied access to some table at some point, but was it due to everyone being denied access? Or due to John being denied access to everything? Such intermediate results can be presented to instructors (or they can delve into intermediate results themselves on the grader container), but those results lack context within the grading

²⁰ Disabling self-assessment might be useful if Labtainers was repurposed for skills assessment testing.