

```

#include <boost/test/unit_test.hpp>

#include "config.hpp"

BOOST_AUTO_TEST_CASE( config_contains_server_type )
{
    std::stringstream ss("{ \"server\" : { \"type\" : \"jenkins\", \"url\" : \"https://example.com\" } }");

    auto config = cic::config::load_from(ss);

    BOOST_CHECK_EQUAL(config.server_type, "jenkins");
}

BOOST_AUTO_TEST_CASE(config_contains_server_url) {
    auto config = cic::config::from_string("{ \"server\" : { \"type\" : \"jenkins\", \"url\" : \"https://example.com\" } }");

    BOOST_CHECK_EQUAL(config.server_url, "https://example.com");
}

BOOST_AUTO_TEST_CASE(config_contains_username_if_specified) {
    auto config = cic::config::from_string("{ \"server\" : { \"type\" : \"jenkins\", \"url\" : \"https://example.com\", \"username\" : \"a@b.com\" } }");

    BOOST_CHECK_EQUAL(config.username, "a@b.com");
}

BOOST_AUTO_TEST_CASE(config_contains_password_if_specified) {
    auto config = cic::config::from_string("{ \"server\" : { \"type\" : \"jenkins\", \"url\" : \"https://example.com\", \"password\" : \"jasfdh\" } }");

    BOOST_CHECK_EQUAL(config.password, "jasfdh");
}

BOOST_AUTO_TEST_CASE(merge_favors_lhs) {
    cic::config a("jenkins", "foo");

    cic::config b("hudson", "bar", "graham", "let me in");

    auto c = a.merge(b);

    BOOST_CHECK_EQUAL(c.server_type, "jenkins");
    BOOST_CHECK_EQUAL(c.server_url, "foo");
    BOOST_CHECK_EQUAL(c.username, "graham");
    BOOST_CHECK_EQUAL(c.password, "let me in");
}

```