

# Airavata Release Testing

This page is to describe the integration testing steps for Airavata release verification and effort to automate those test for future release.

## SCOPE

Airavata testing for running Grid (XSEDE, Campus Clusters) and Cloud jobs to verify the release. You need to have knowledge about running applications on XSEDE or other cluster to perform part of these tests.

## Verify Build

1. Compile the project with an empty Maven repository `mvn install -Dmaven.repo.local=/tmp/cleanrepo`
2. Verify the modules/distribution/ (Make sure size of each distribution is less than 100MB)
  - airavata-server
    - a. /apache-airavata-server-<version>-bin.tar.gz
    - b. target/apache-airavata-server-<version>-bin.zip
    - c. target/apache-airavata-server-<version>.war.tar.gz
    - d. target/apache-airavata-server-<version>.war.zip
  - airavata-client
    - a. target/apache-airavata-client-<version>-bin.tar.gz
    - b. target/apache-airavata-client-<version>-bin.zip
  - xbaya-gui
    - a. target/apache-airavata-xbaya-gui-<version>-bin.tar.gz
    - b. target/apache-airavata-xbaya-gui-<version>-bin.zip
3. Untar/zip the Airavata server bin distribution to verify the embedded version.
4. Untar/zip the Xbaya GUI bin distribution and work with embedded version of Airavata server.
  - Follow the instructions to run [5 min](#) and [10 min](#) tutorials.
5. Deploy the war distribution to Tomcat 7. Follow [Deployment Guide](#) to configure Airavata on Mysql and Tomcat. Guide also have details on server property configuration.
6. Start the Tomcat to perform tests below.
7. Use Xbaya GUI bin distribution to perform test.

## Test Registry (Xbaya currently)

1. Connect to registry using Registry > Setup Airavata Registry tab.
2. Add a new host to the registry.

3. Edit the existing host by updating all the properties and verify the changes
4. Try to add/edit different types of hosts.
5. Add applications for each type of host configuration. You can use local/SSH/GSISSH/GRAM5 protocols in Airavata to run your applications. You need to update airavate-server.properties for security setting for above protocols.
6. Edit Applications to make sure all the parameter can be updated and saved to registry.

#### **Test Running Workflow (XBaya currently)**

1. Add new workflow and add an application configured in previous step.
2. Configure inputs and outputs for selected application. [10 min](#) tutorials can help you with how to configure inputs and outputs.
3. Run the workflow using Run workflow button and monitor the workflow using monitor tab in XBaya.
4. Look at the Tomcat server logs to make sure there is no exception on the console while running the workflow
5. Look at XBaya console for exceptions also and report if there is any error.
6. Workflow progress and results should be reported to XBaya
7. Use View > Airavata Registry for provenance information about experiment workflow.
8. Make sure you are able to browse through registry data tree.
9. Run the workflow several times to make sure consistent behavior.
10. Test all different hosts and applications setting to make sure different configurations are working fine.

#### **Test Airavata API ([API Samples](#))**

1. Register a new host and get host. ([Sample](#))
2. Register an application and get created application. ([Sample](#))
3. Access workflows registered already in registry.
4. Running an workflow by setting inputs. ([Sample](#))
5. Monitor the running workflow ([Sample](#))
6. Get workflow results
7. Set workflow header to select host dynamically.
8. Update HPC setting of current workflow application.

**Note:** Report any issues to [Airavata JIRA](#)

**Note:** Add test scenarios for other components like Messaging, XBaya, Registry