Installing and Configuring the Sauce OnDemand Plugin for Jenkins

Important Steps for All Continuous Integration Configurations

When you are configuring your continuous integration platform to work with Sauce, always make sure to follow these steps:

1. Update tests to reference the environment variables set by the plugin. See the topic on **Setting Environment Variables** for more information.
2. Output the Sauce session ID to the standard output (`stdout`) to allow the Sauce plugin to associate test results to Sauce Jobs. See the topic on **Outputting Session IDs to stdout** for more information.

Changelog for Plugin Versions

You can find the complete changelog for each version of the Jenkins Plugin at GitHub: https://github.com/saucelabs/jenkins-sauce-ondemand-plugin/blob/master/CHANGELOG.md

Requirements

- You need to be able to access the IP range `162.222.72.0/21`, `saucelabs.com`, and `ondemand.saucelabs.com` from your Jenkins server. Check out the Jenkins documentation if you need to set up a proxy or other advanced network configuration.

Install the Plugin

You can install the Sauce OnDemand plugin for Jenkins though the Jenkins Administration page.

1. In Jenkins, go to the **Administration** page.
2. Click **Manage Jenkins**.
3. Click **Manage Plugins**.
4. Click the **Available** tab.
5. In the list of plugins, find and select **Sauce OnDemand Plugin**.
6. Click **Download now and install after restart**.
   The plugin file is fairly large, so download may take several minutes.
7. In the plugin installation dialog, select **Restart Jenkins when installation is complete and no jobs are running**.

Configure Your Sauce Labs Credentials

The Jenkins plugin provides an interface for storing your Sauce Labs authentication credentials as environment variables on the Jenkins server, which is one of our best practices for testing with Sauce. This allows you to reference your credentials without having to hardcode them into your tests, and because the plugin manages authentication at the global level, you can have multiple jobs running at the same time that use these credentials.

1. After the plugin has installed and Jenkins has restarted, go to the **Administration** page in Jenkins.
2. Click **Credentials**.
3. You can click an existing domain or click **Add domain.**
4. Once in your domain of choice, click **Add Credentials.**
5. Under **Kind**, select **Sauce Labs.**
6. Enter the **Username** and **API Access Key** for your Sauce account.
7. Click **OK** to save.
   Your credentials are now saved.

You can now use these credentials on a per project basis. To do so:

1. Go to the **Administration** page for one of your projects.
2. Click **Configure.**
3. Under **Sauce Labs Options,** you will see the credentials you have stored as an option under **Credentials**.

   You can also add credentials here on the fly.

Versions Before 1.147

In versions of the plugin prior to 1.147 credential management is handled slightly differently. For older versions of the plugin you will find credential management in the **Manage Jenkins > Configure System** section:

1. After the plugin has installed and Jenkins has restarted, go to the **Administration** page in Jenkins.
2. Click **Manage Jenkins**.
3. Click **Configure System**.
4. Under **Sauce Support**, enter the **Username** and **API Access Key** for your Sauce account.

Setting Authentication Environment Variables in Your Tests

Once you've set up your authentication credentials as environment variables, you can refer to them in your tests like this:

```
WebDriver driver = new RemoteWebDriver(
            new URL("http://"+System.getenv("SAUCE_USERNAME")+":"
+System.getenv("SAUCE_ACCESS_KEY")+"@ondemand.saucelabs.com:80/wd
/hub",
            desiredCapabilities);
```

Check out the topic Setting Desired Capabilities for Jenkins Projects for more information about using environment variables.