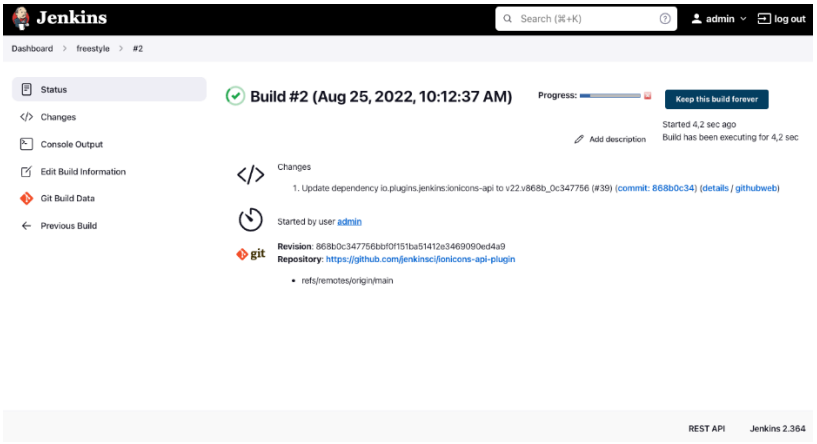
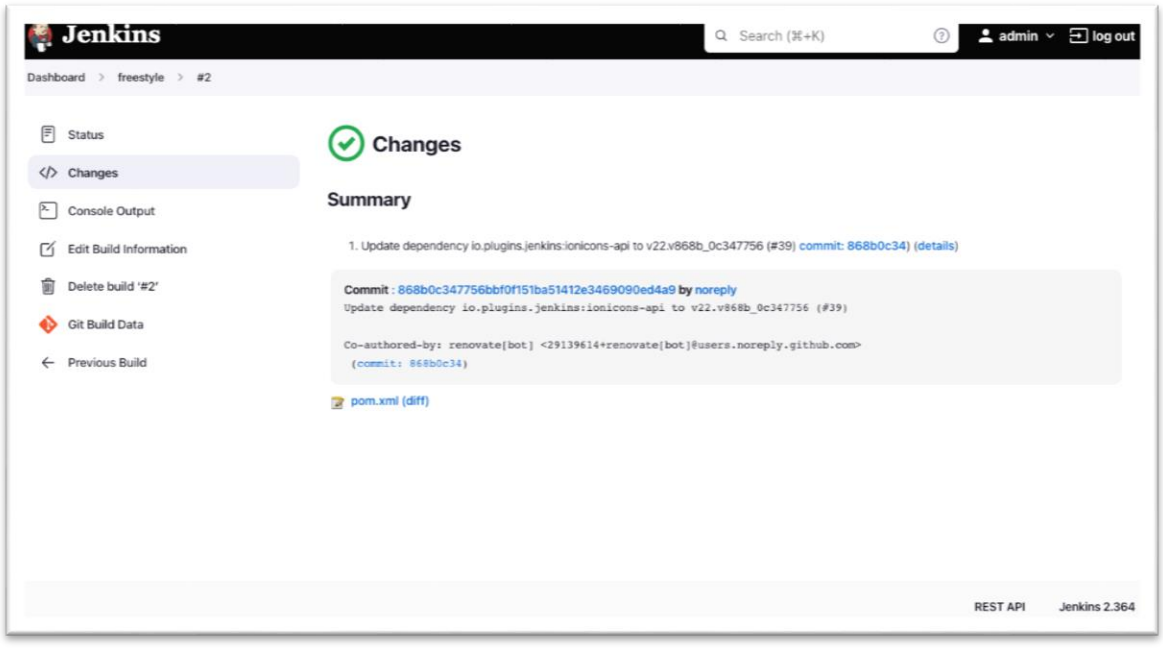
Jenkins : Web-hook

Web hooks are an essential component of continuous integration and continuous delivery (CI/CD) systems like Jenkins.





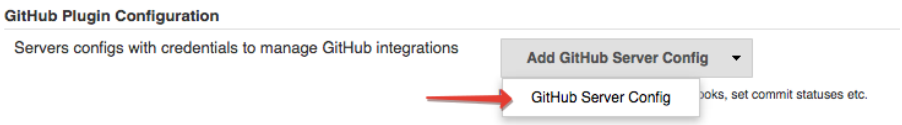
GitHub hook trigger for GITScm polling :

This feature enables builds after [post-receive hooks in your GitHub repositories](). This trigger only kicks git-plugin internal polling algo for every incoming event against matched repo.

Manual Mode

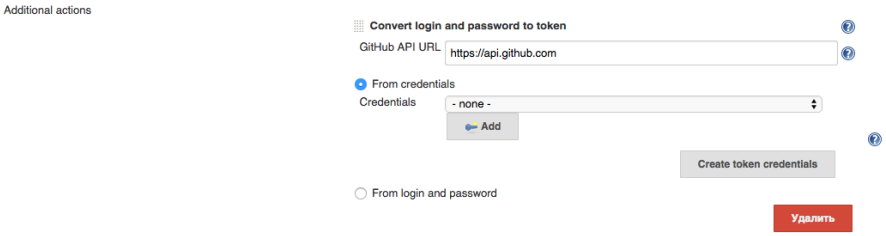
In this mode, you'll be responsible for registering the hook URLs to GitHub. Click the icon (under Manage Jenkins > Configure System > GitHub) to see the URL in Jenkins that receives the post-commit POSTs — but in general the URL is of the form

Step 1. Go to the global configuration and add GitHub Server Config.

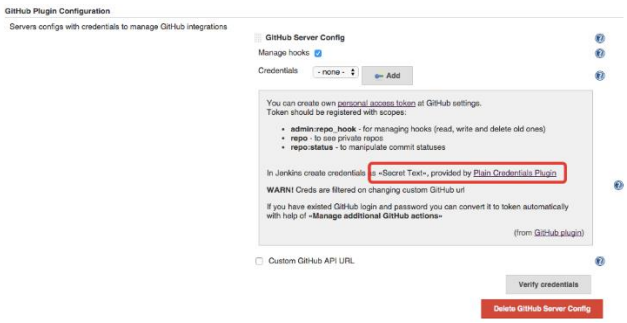


Step 2.1. Create your personal access token in GitHub.

Plugin can help you to do it with all required scopes. Go to **Advanced** -> **Manage Additional GitHub Actions** -> **Convert Login and Password to token**



Step 2.2.: **Select previously created "Secret Text" credentials with GitHub OAuth tokTo be able manage hooks your token should have** admin:org_hook **scope.**



Step 3. Once that configuration is done, go to the project config of each job you want triggered automatically and simply check "GitHub hook trigger for GIT Scm polling" under "Build Triggers".

Security Implications

This plugin requires that you have an HTTP URL reachable from GitHub, which means it's reachable from the whole internet. So it is implemented carefully with the possible malicious fake post-receive POSTS in mind.

Jenkins inside a firewall:

In case your Jenkins run inside the firewall and not directly reachable from the internet, this plugin lets you specify an arbitrary endpoint URL as an override in the automatic mode. The plugin will assume that you've set up reverse proxy or some other means so that the POST from GitHub will be routed to the Jenkins.