**O que é o Jenkins?**

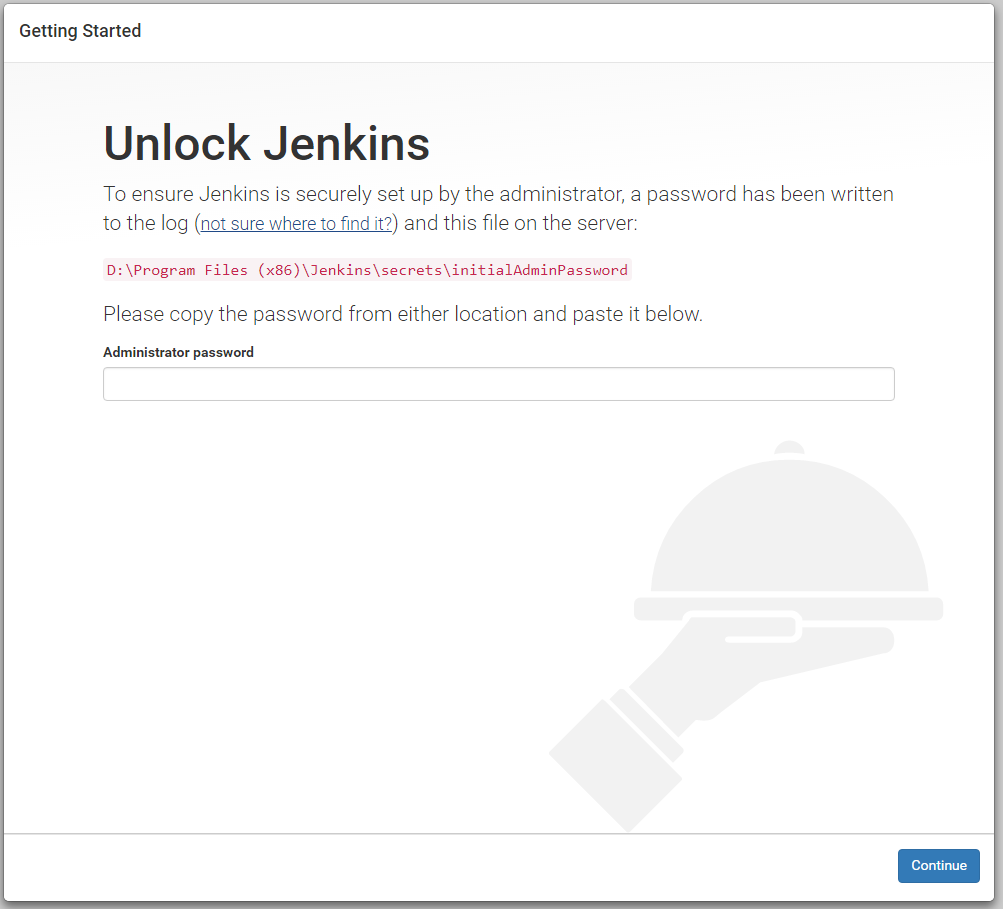
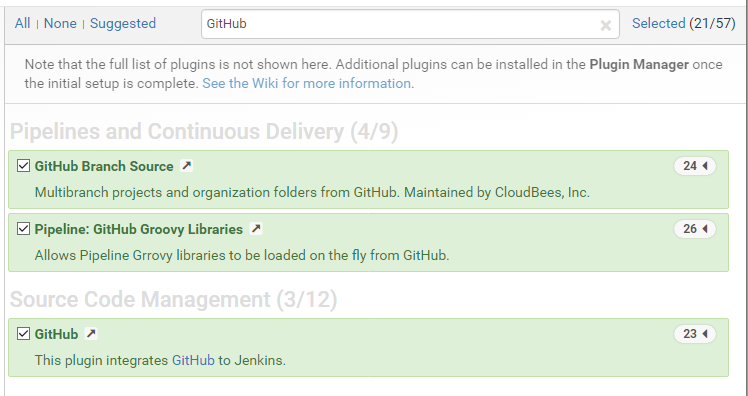
Jenkins é uma aplicação java que permite implementar Continuous Integration e Continuous Delivery ao nosso projeto.

**O que significa Continuous Integration e Continuous Delivery?**

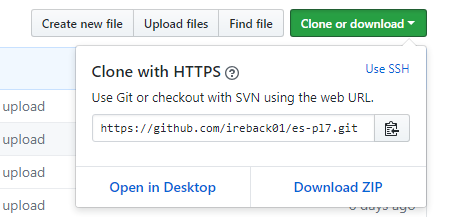
Visto estarmos a trabalhar em equipa, temos vários devolopers a trabalhar nos seus computadores e posteriormente a fazer upload do trabalho realizado para o GitHub. A probabilidade de um código ter erros está sempre presente e portanto, quantos mais utilizadores, mais probabilidade de ocorrerem erros.

É aqui que o Jenkins entra: Depois de associado ao GitHub, ele vai pegar nos últimos ficheiros de código submetidos, fazer a build da plataforma (app, exe, website,…) e correr vários testes (Nunit, MSTest, Gradle,…). Por fim, é nos fornecido um report dos erros e problemas que ocorreram.

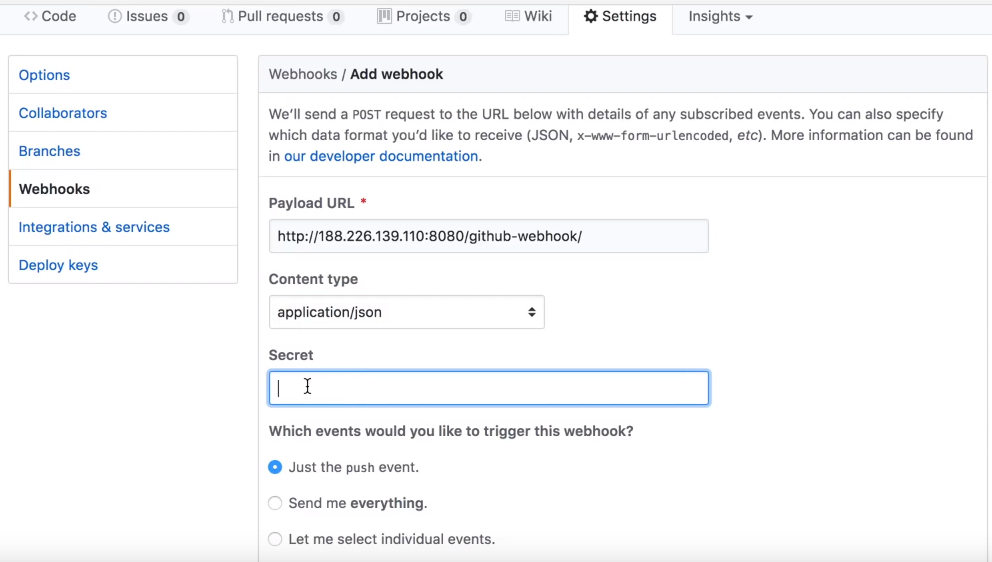
**Como instalar?**

1. <https://jenkins.io/download/> , escolher versão LTS adequada ao teu SO
2. Instalar o ficheiro transferido
3. Ao abrir o Jenkins pela 1ª vez será necessário inserir uma password que se encontra no diretório que é indicado
4. Escolher a 2ª opção que permite selecionar os plugins individualmente
5. Pesquisar por “GitHub” e incluir na instalação

Após a inicialização, temos o Jenkins a funcionar com o plugin do GitHub. Agora falta apenas criar um projeto linkado com o repositório GitHub.

1. Selecionar “Crie um novo projeto”
2. Dar um nome e escolher “Freestyle Project”
3. Em “General” escolher “GitHub Project” e introduzir a url do repositório a usar
4. De seguida, em “Source Code Management” selecionar “Git” e adicionar o url obtida no GitHub como na imagem:
5. Em “Build Triggers” selecionar “GitHub hook trigger for GITScm polling”
6. Após estes passos podemos carregar em “Save” e neste momento temos o Jenkins à espera de qualquer alteração no GitHub para voltar a dar “Build” ao projeto.

Agora falta só fazer com que o GitHub notifique o Jenkins de alterações.

1. No site do GitHub em Settings - > Webhooks -> Add Webhook
2. Payload URL será a url do Jenkins, seguido de “/github-webhook”, Content type será “application/json” e o resto permanecerá default.

Por fim, a configuração está concluída e cada vez que existir um evento “push” no GitHub, o Jenkins será notificado e procederá a fazer a build do projeto atualizado.