Experiment - Build Pipelines

Jenkins pipelines help you align the build process of a project. This is done by specifying tasks and the order in which they are executed. There are all kinds of possible tasks that a Jenkins pipeline can do for you. For example, build assets, send an email on error, send the build artifacts via SSH to your application server, etc.

Setup of Pipelines

Jenkins allows to specify pipelines using a `Jenkinsfile`. This is just a textfile that contains the necessary data for Jenkins to execute the pipeline. It is called `Jenkinsfile` (notice: no file extension) and should be placed in the root of your project.

> This file should be checked into version control as it is needed on your Jenkins instance.

Jenkins supports two different syntaxes.

1. Declarative (since Pipeline version 2.5)
2. Scripted

For this tutorial we will focus on the declarative approach.

The following example shows a pipeline with 2 stages, this would go into a project file named “Jenkinsfile”. This is case sensitive and must be in this format. Place this in your GitHub, BitBucket or GitLab account. The example below would use GitHub, but any git environment will work.

```
pipeline {
    agent any

    stages {
        stage('Build Assets') {
            agent any
            steps {
                echo 'Building Assets...'
            }
        }
        stage('Test') {
            agent any
            steps {
                echo 'Testing stuff...'
            }
        }
```

```
    }
}
```

The agent directive tells Jenkins to allocate a workspace and an executor for the pipeline. Without it, the pipeline is not valid and therefore required.

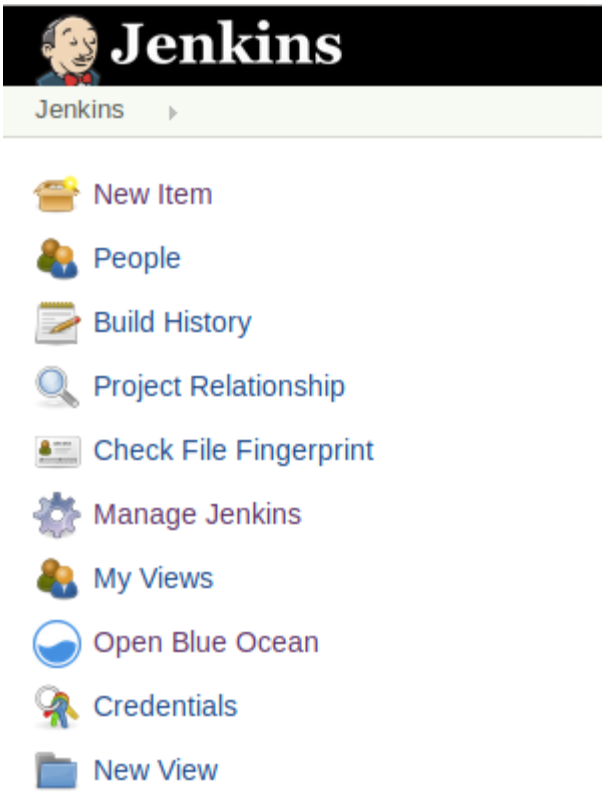
Setup using the Blue Ocean Plugin

The above process can also be done using the Blue Ocean *Jenkins* Plugin.

Installation

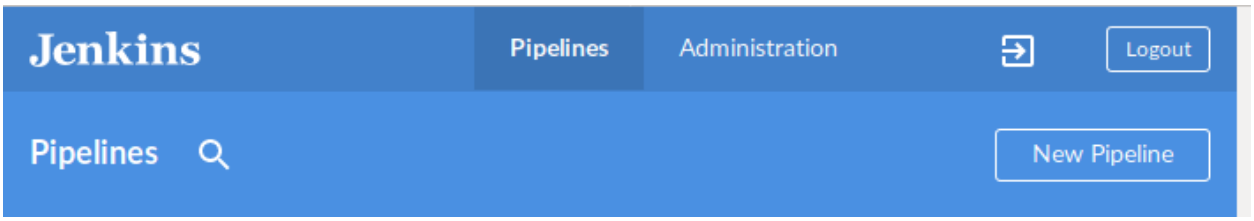
To install the Plugin go to **Manage Jenkins Manage Plugins Available** and select the *Blue Ocean* Plugin.

After the installation is finished you have an additional menu entry called `Open Blue Ocean` in your main *Jenkins* navigation.

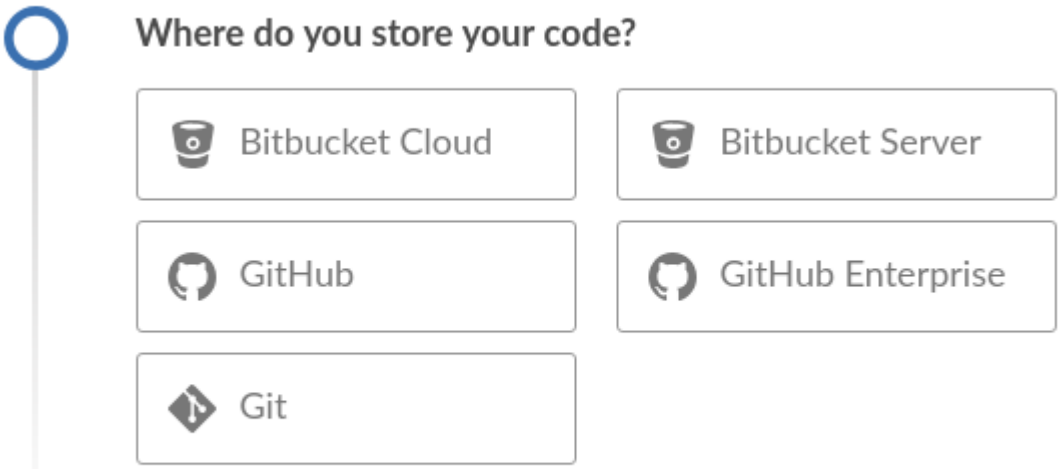


5.2.2. Creating a new Pipeline

Click on `New Pipeline` to create a new Pipeline.



Select your version control provider.



> For this example we will use a GitHub repository, but any git or other version control repo could be used.

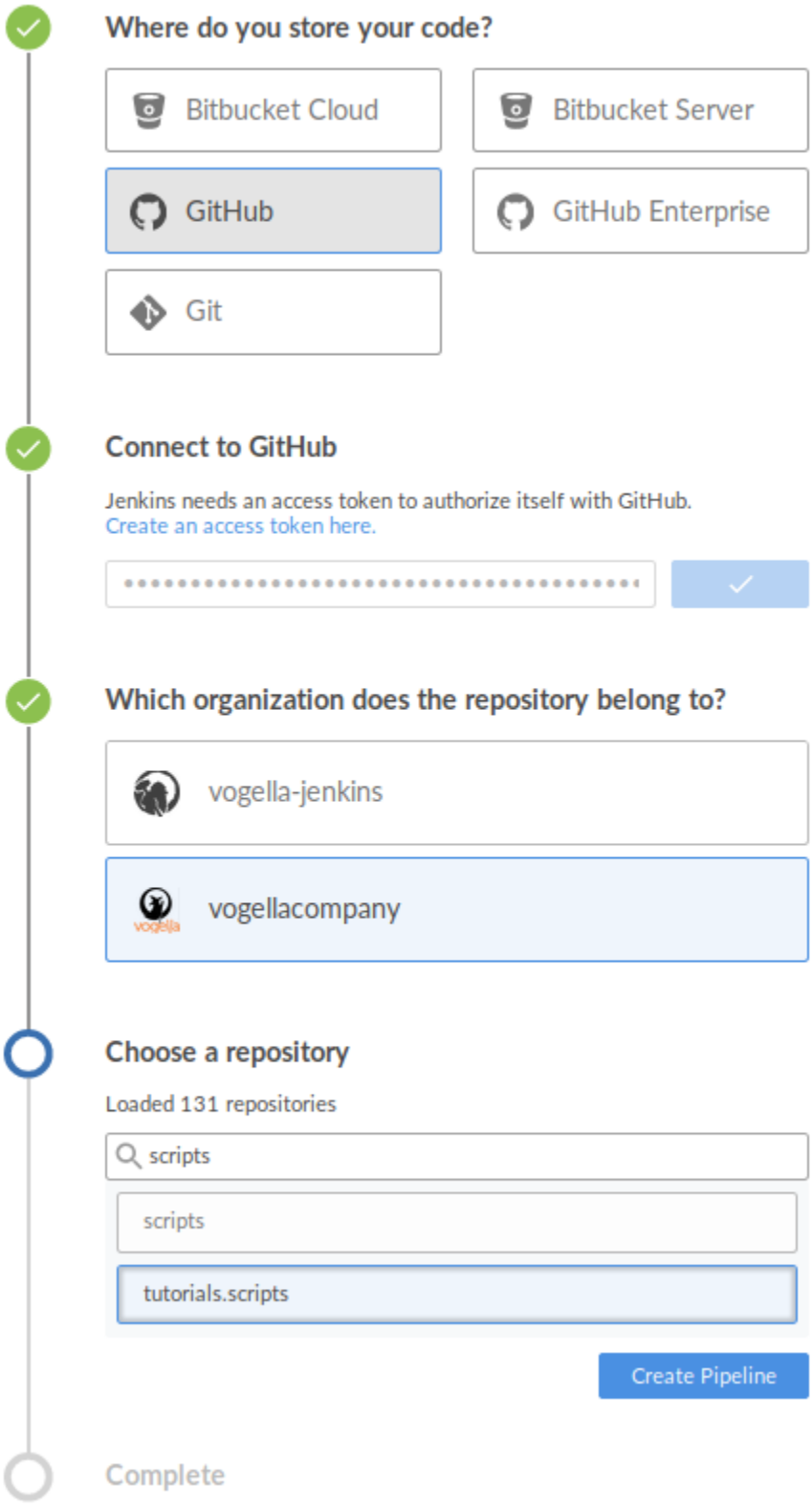
Depending on your provider you will need to pass some kind of credentials.

> GitHub provides the ability to generate access-tokens that applications can use to access the platform with your user. You can also restrict what the access-token can do.

The *Blue Ocean* application will provide a link to the GitHub page you need to visit. The necessary permissions that *Blue Ocean* needs to operate are already selected. Add a description and click on `Generate Token` at the bottom of the page.

Copy the generated token and paste it in the *Blue Ocean* mask.

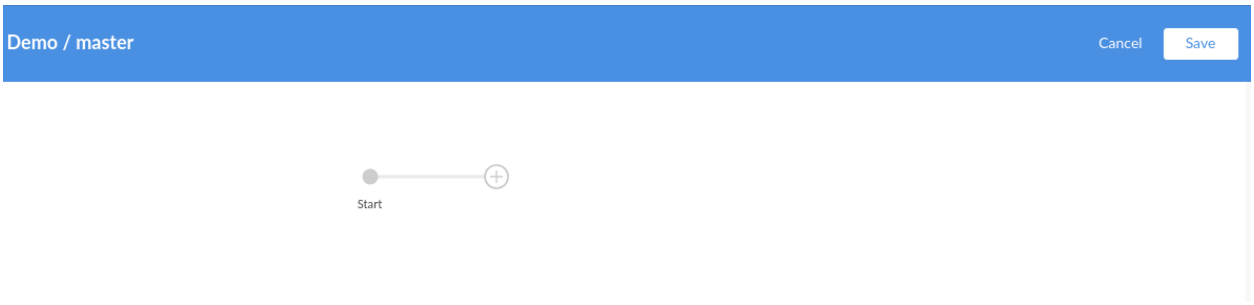
Select the account the repository belongs to and select the repository.


Where do you store your code?
Bitbucket Cloud
Bitbucket Server
GitHub
GitHub Enterprise
Git
Connect to GitHub
Jenkins needs an access token to authorize itself with GitHub.
Create an access token here.
Which organization does the repository belong to?
vogella-jenkins
vogellacompany
Choose a repository
Loaded 131 repositories
scripts
scripts
tutorials.scripts
Create Pipeline
Complete

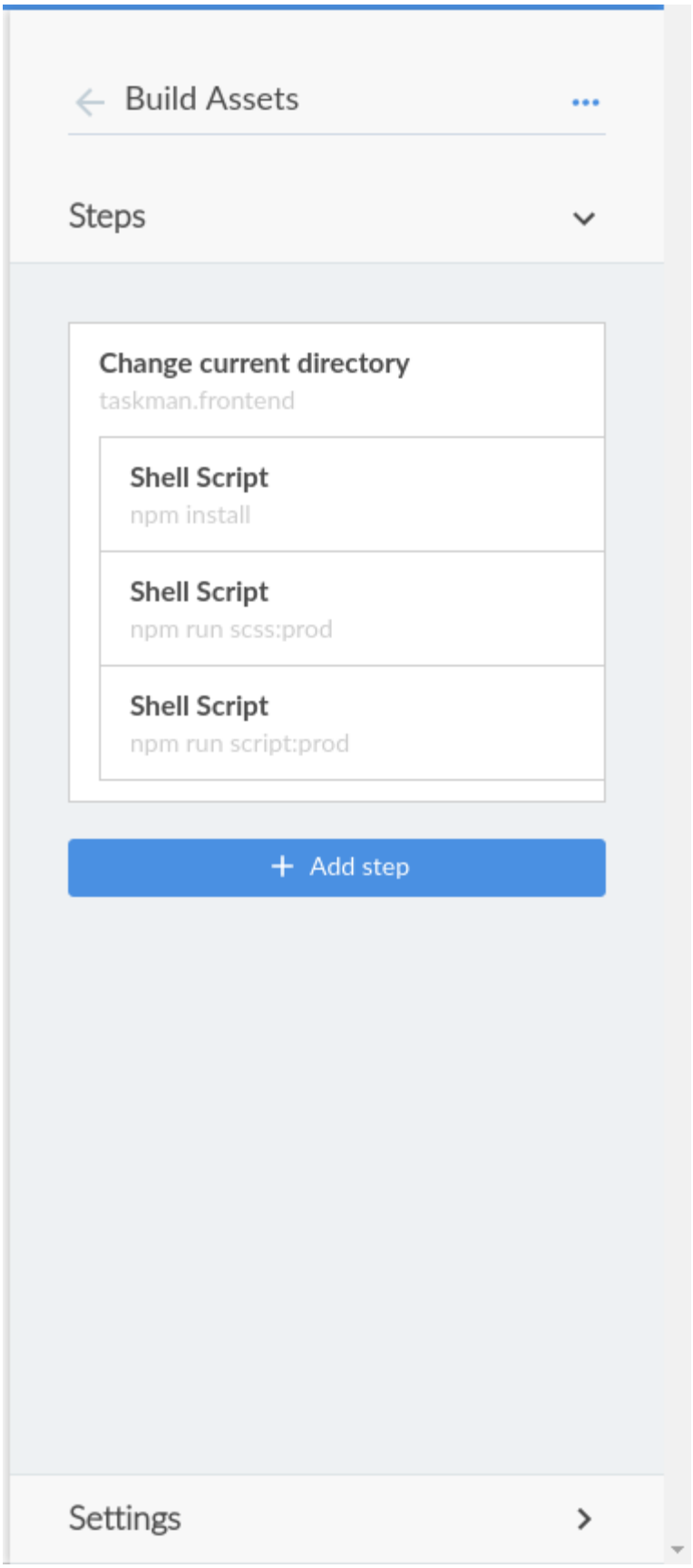
If you already have a Jenkinsfile with pipelines in the repository it will show up in the last step. If not, *Blue Ocean* will offer to create one for you.

Adding steps to your Pipeline

In the next screen you will see a visual representation of your Pipeline. Here you can add or remove steps.



To create a new step click on + in the canvas. A menu will open on the right that lets you specify a name and what steps you want to perform.


Build Assets
Steps
Change current directory
taskman.frontend
Shell Script
npm install
Shell Script
npm run scss:prod
Shell Script
npm run script:prod
Add step
Settings

After you have finished editing the Pipeline Blue Ocean will offer you to commit the newly created pipeline to your repository.

Under the hood *Blue Ocean* only created a valid `Jenkinsfile` to be used by Jenkins.

After committing *Jenkins* will build the project using the newly modified Pipelines.