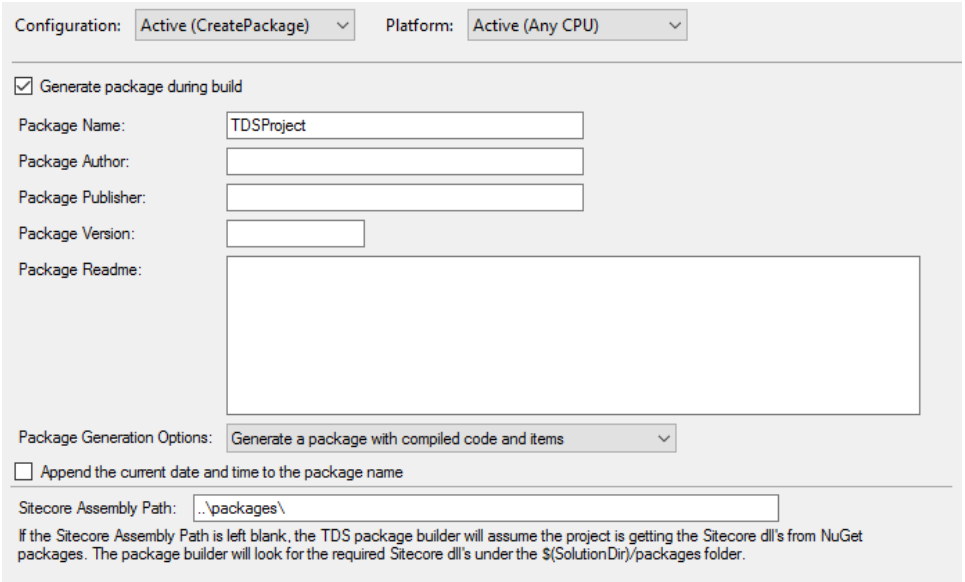
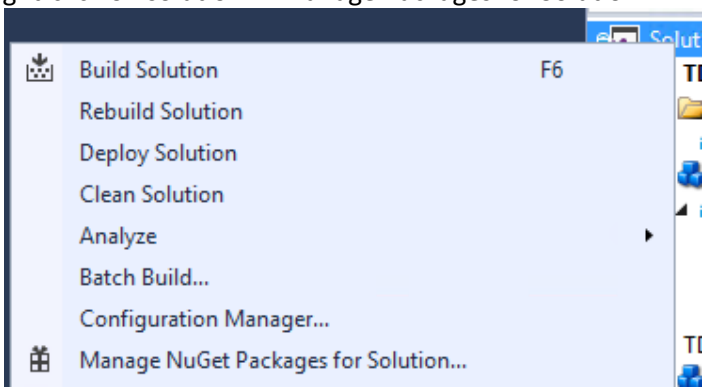
Example setup of TC and Octopus with TDS

On the developers' machines:

1. Install TDS
2. Configure the source control repository
3. Configure the TDS Project
    a. Configure TDS to create .update packages during build.

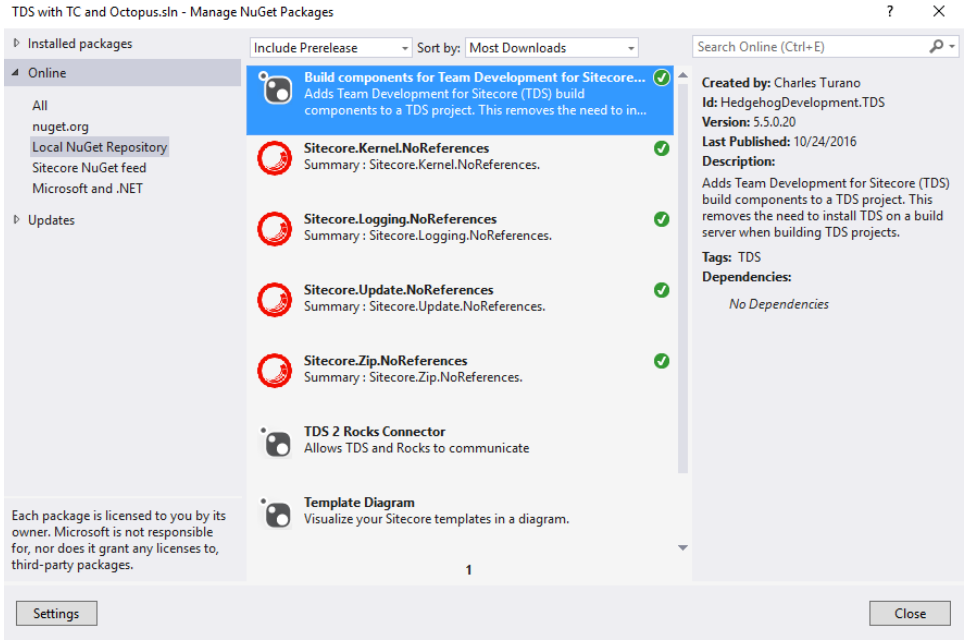


    b. Add the Solution to source control (if you haven't done that during the creation of the projects)
    c. Install the TDS Build Components NuGet Package
        a. Right-click on solution -> Manage Packages for Solution



        b. Add custom NuGet repository, pointing at the TDS NuGet package folder -or- in case you have your own NuGet repo, upload the package to it
        c. Add the Sitecore.Kernel, Sitecore.Logging, Sitecore.Update and Sitecore.Zip NuGet packages for the used Sitecore version in the same NuGet Repository. You can download them from: https://sitecore.myget.org/gallery/sc-packages
        d. Switch to the custom repository and select the HedgehogDevelopment.TDS package that holds the build components
        e. Install the package to ***every*** TDS project inside the Solution

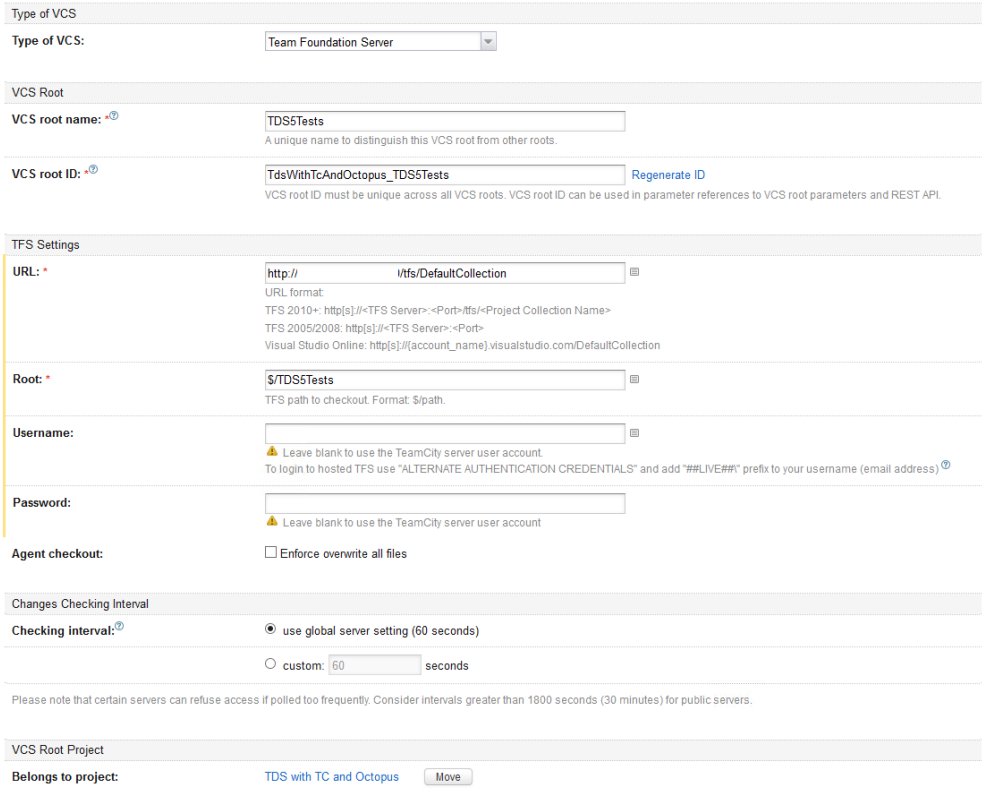
f. Install all assembly packages to the Web Project



h. After installing, restart Visual Studio to apply the changes

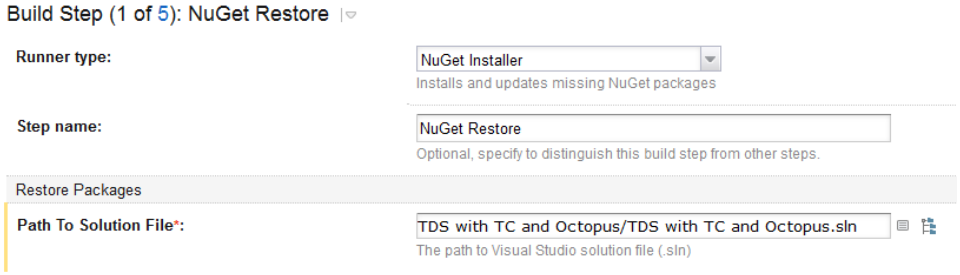
On the TC server

1. Create a new project in TC
2. Configure TC to work with your source control repository



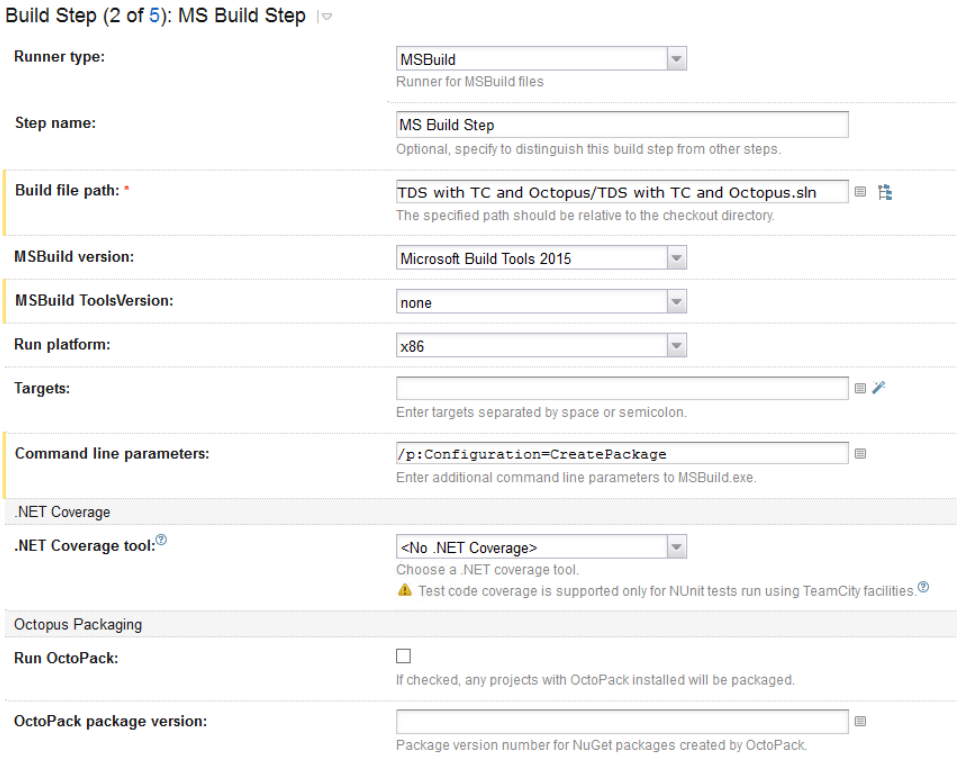
3. Configure the NuGet Restore step

   This step restores the NuGet packages, containing the TDS Build Components and the assembly files, which will be later used in the MS Build step.

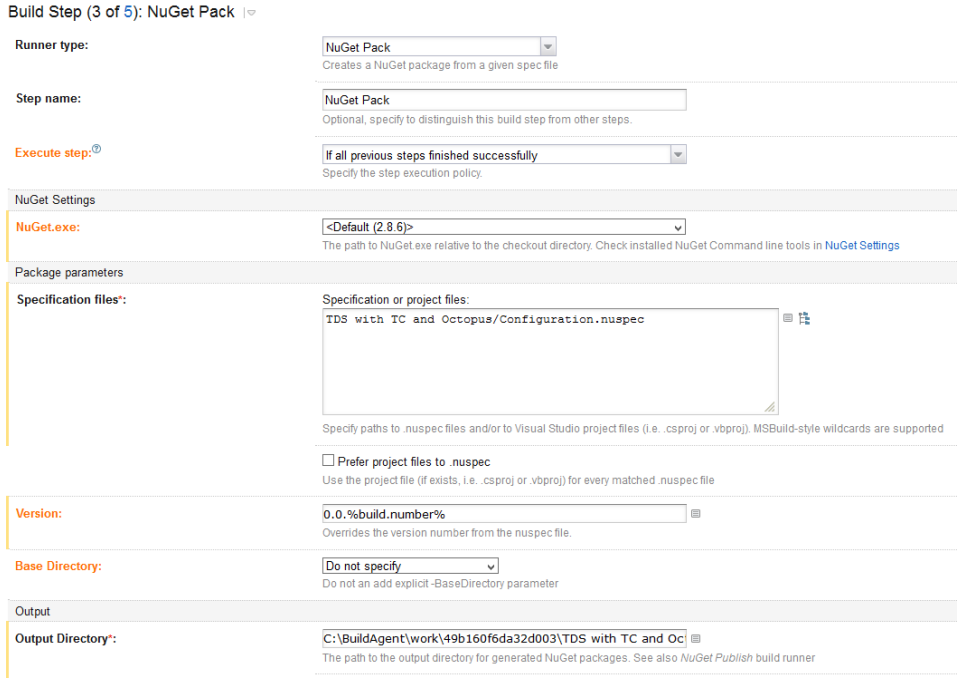


4. Configure the MS Build Step

   Find the relative path to the Solution file and add a command line parameter to select the right Build configuration



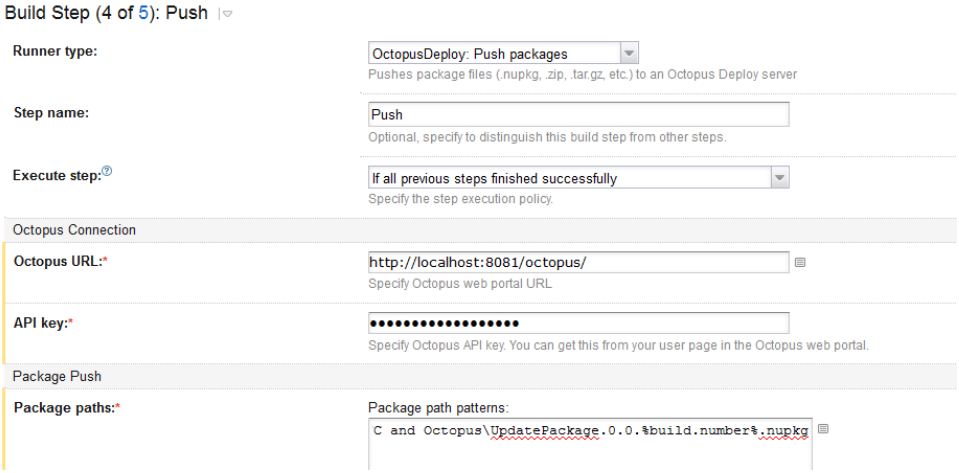
5. Configure TC to create a NuGet Package



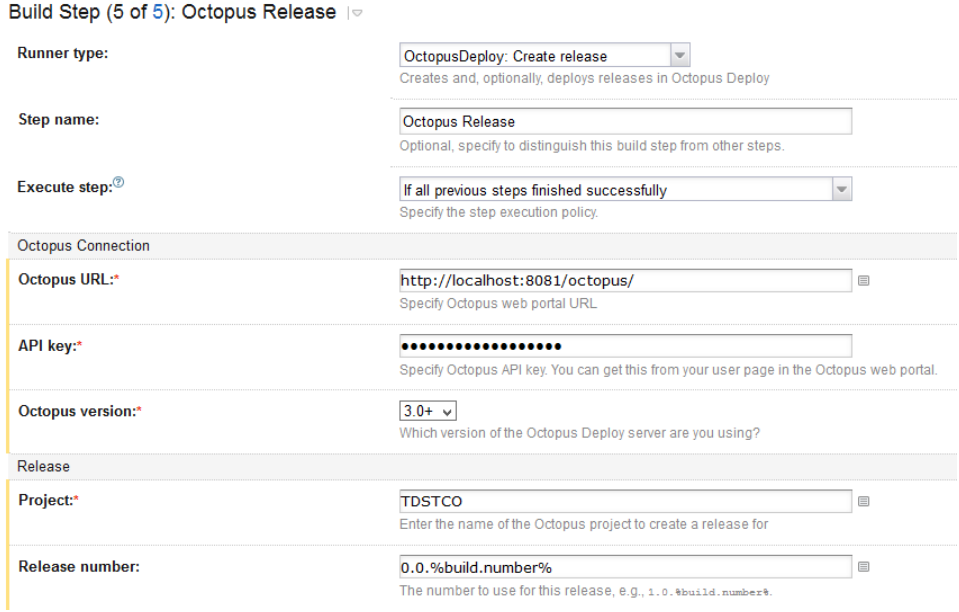
It's important to create and point to a .nuspec file. Here's an example one:

```
<?xml version="1.0"?>
<package xmlns="http://schemas.microsoft.com/packaging/2011/08/nuspec.xsd">
 <metadata>
  <id>UpdatePackage</id>
  <version>1.0.0</version>
  <title>Update package</title>
  <authors>Me</authors>
  <owners>Hedgehog Development</owners>
  <requireLicenseAcceptance>false</requireLicenseAcceptance>
  <description>Contains .update package</description>
  <copyright>Copyright ©</copyright>
  <tags></tags>
 </metadata>
  <files>
  <file src="C:\BuildAgent\work\49b160f6da32d003\TDS with TC and
Octopus\TDSProject\bin\Package_CreatePackage\TDSProject.update" />
 </files>
</package>
```

6. Create a step, which pushes the NuGet Package to the Octopus NuGet Repository



7. Create a step, which will create a Release in Octopus



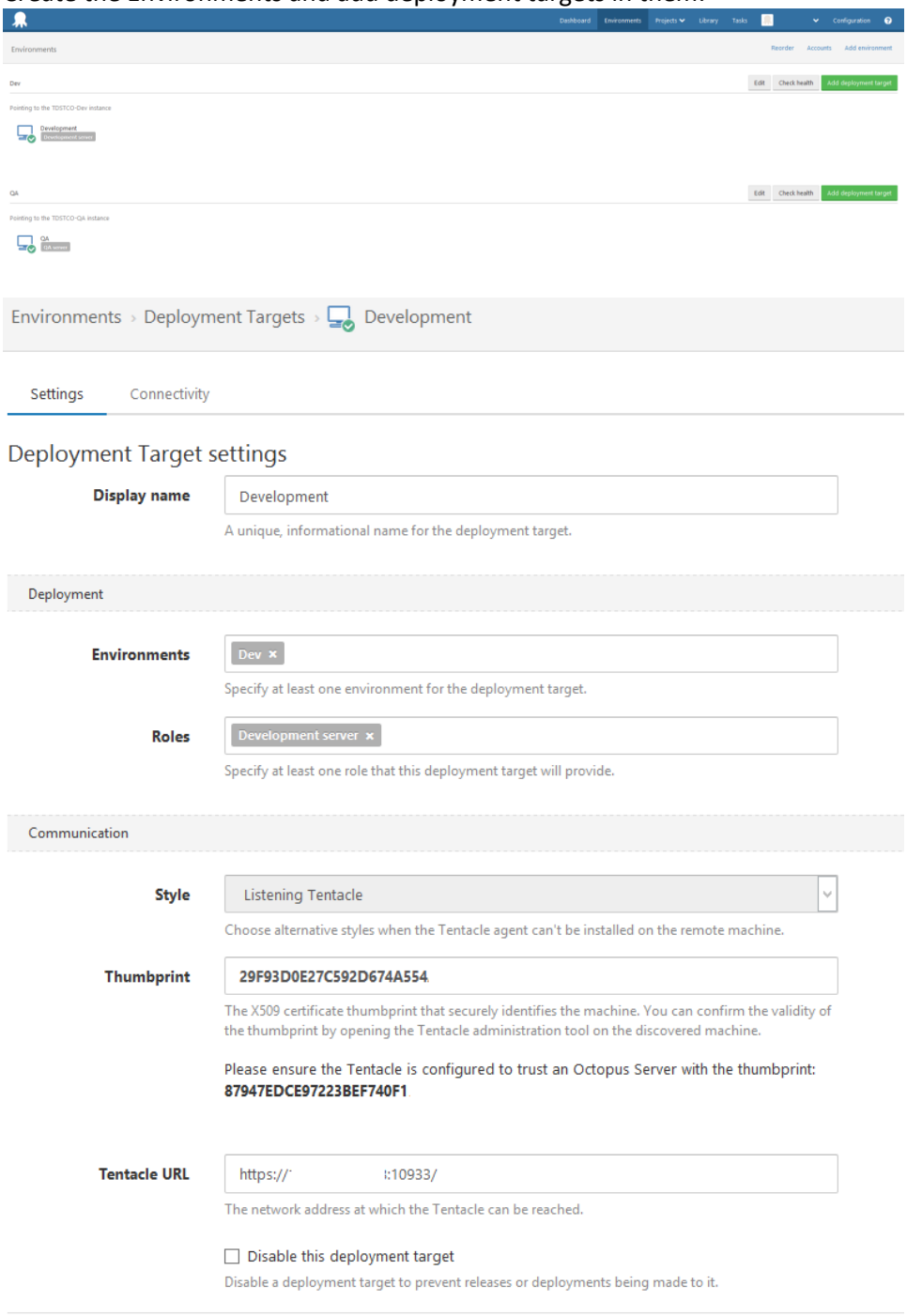
It's important to select the Project, created in Octopus. The API key is created from the user profile in Octopus.

8. Licensing

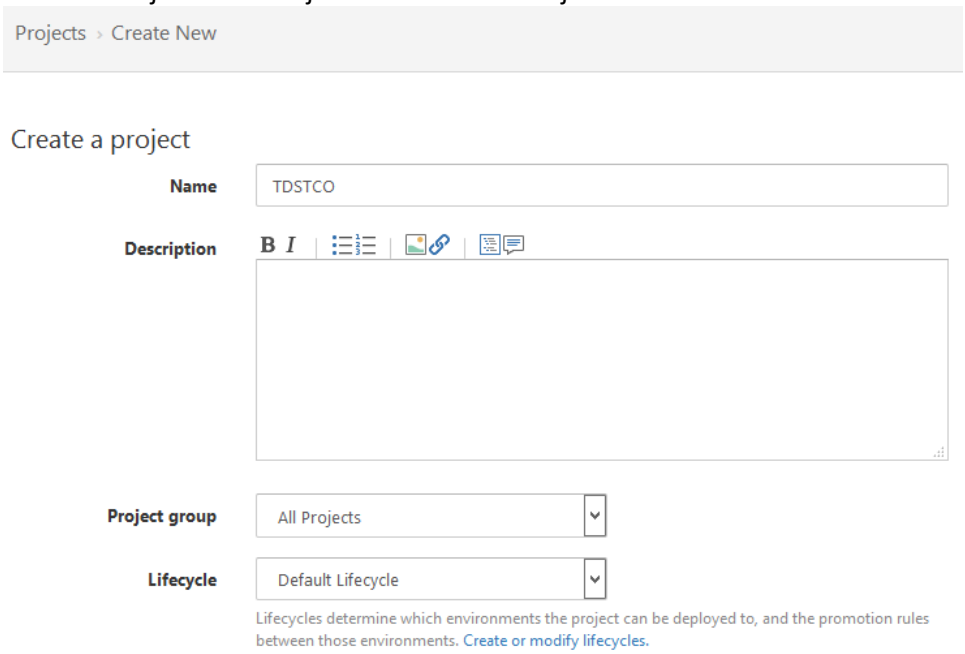
    Add two environment variables with your license information as values:
    TDS_Owner
    TDS_Key

On the Octopus Server

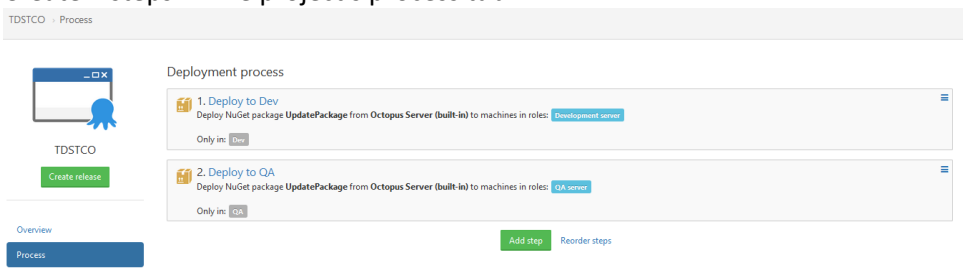
1. Create the Environments and add deployment targets in them:



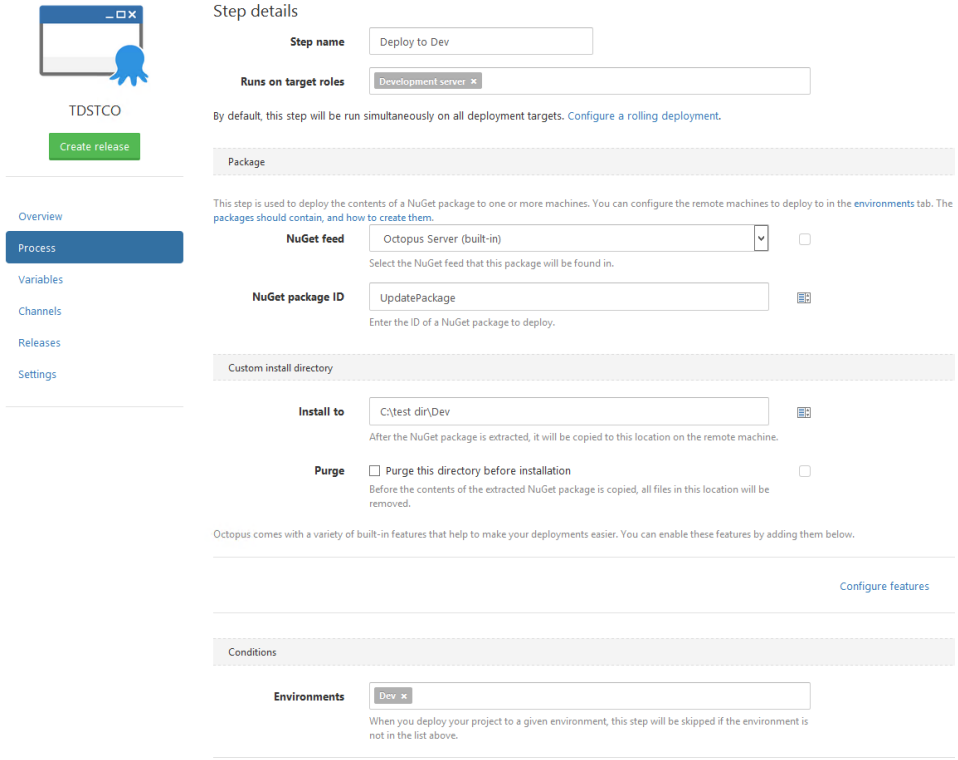
2. Create a Project from Projects -> All -> Add Project



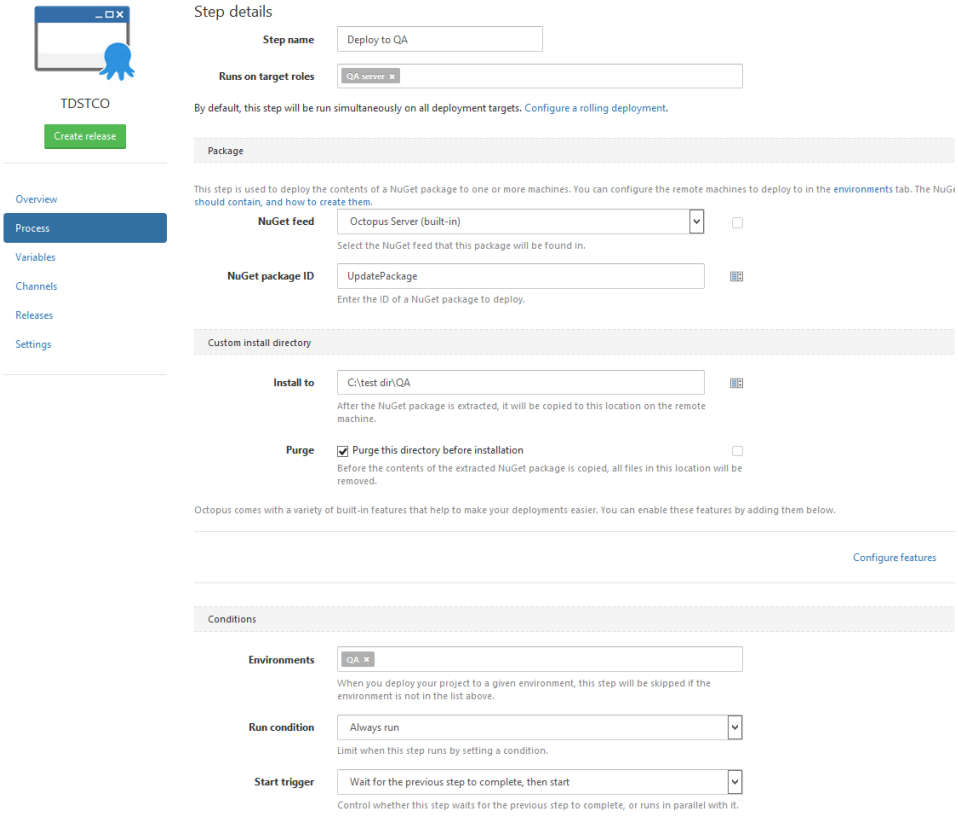
3. Create 2 steps in The project's process tab:



One deploying to Dev



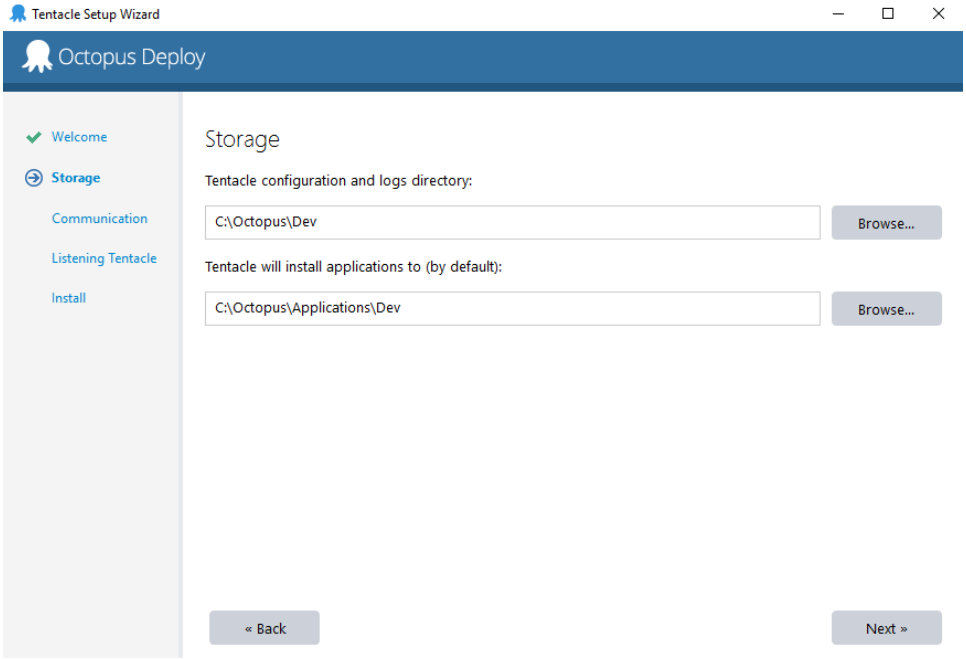
And another one, deploying to QA



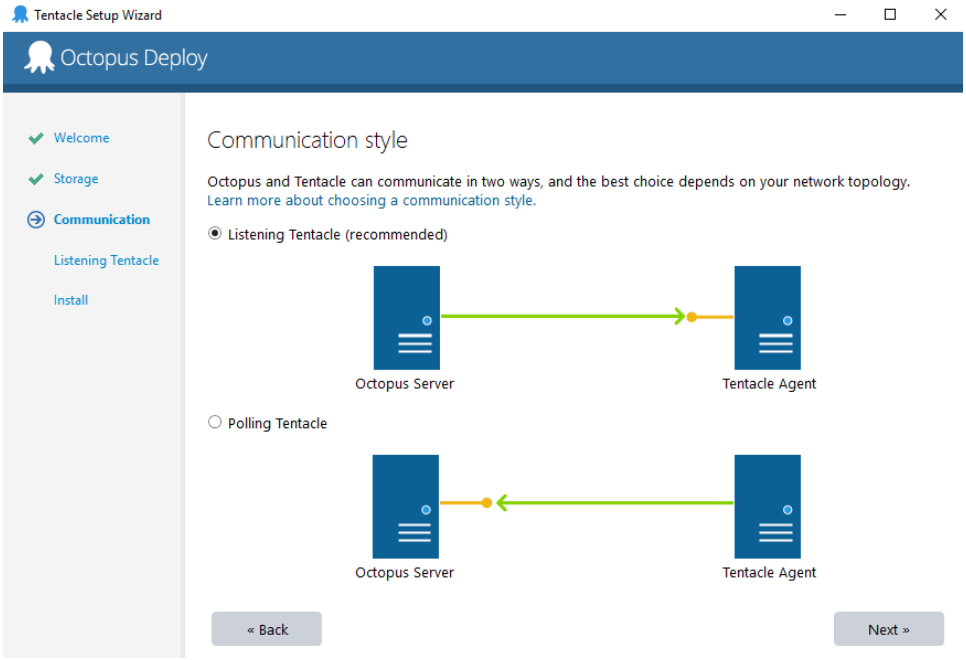
In the "Install to" textbox, insert the path where you want your .update package to be deployed.

On the Sitecore servers

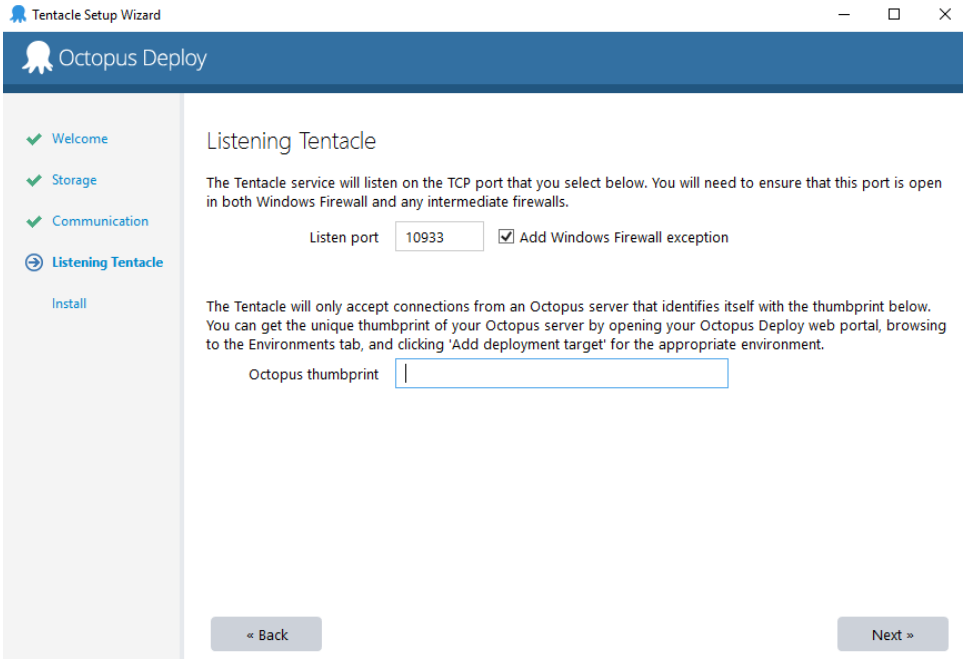
1. Install the Listening Tentacles (agents) on the machines where your Sitecore instances are hosted. The Tentacles can be downloaded from Octopus during the creation of the New Deployment Target.
   The Thumbprints in Octopus and the Tentacle must be the same.



The directories at this tab can be left default, since we choose a custom directory for the deployment



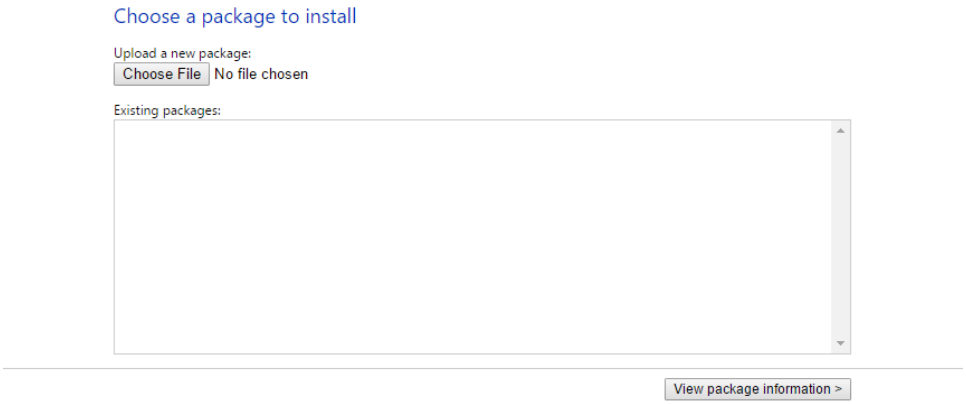
Select "Listening Tentacle"



Select the port and paste the thumbprint from the Deployment Target settings in Octopus

Automating with Sitecore Package Deployer

The Dev environment will show changes automatically, using the Sitecore Package Deployer (http://www.hhogdev.com/blog/2015/september/sitecore-package-deployer.aspx). It's an open source tool, which install .update packages automatically.
It is installed as from http://<sitecore-instance>/sitecore/admin/UpdateInstallationWizard.aspx



The Sitecore Package Deployer checks for .update package every minute in: \Data\SitecorePackageDeployer, so this path must be set in the "install to" text box in the Deployment steps settings.

Once this is set up you can deploy changes to the targeted Sitecore instances.