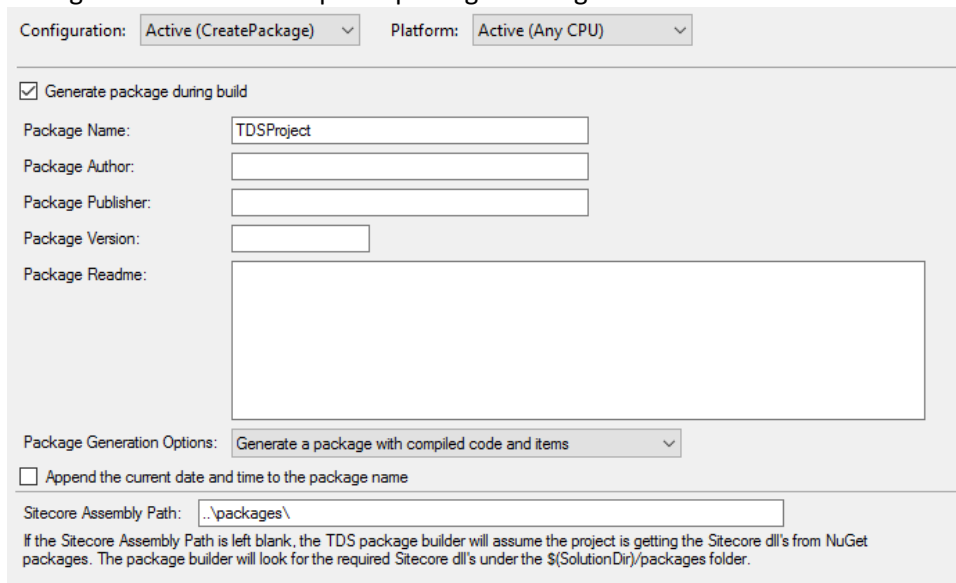


Example setup of TC and Octopus with TDS

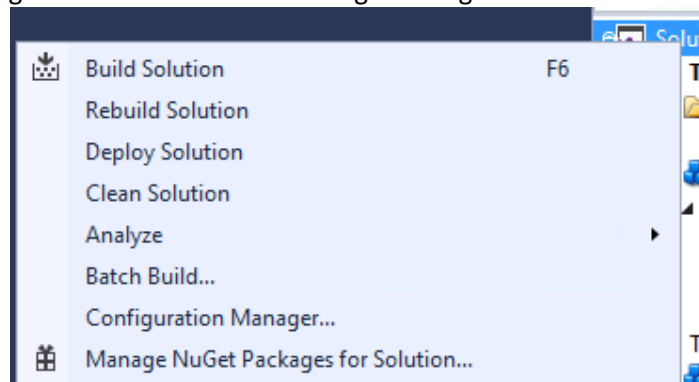
On the developers' machines:

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



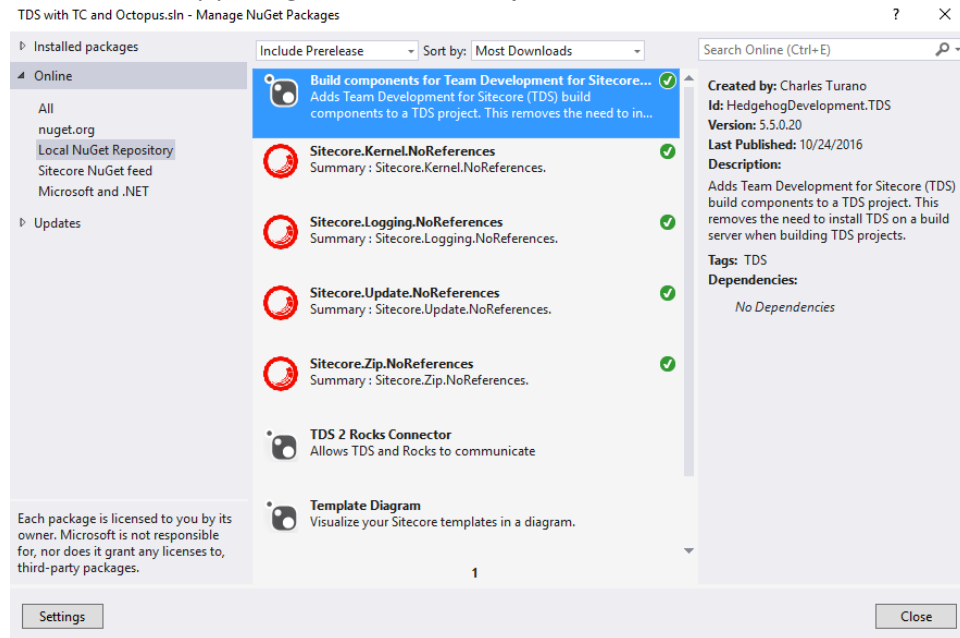
The screenshot shows the 'Generate package during build' configuration window. At the top, there are two dropdown menus: 'Configuration: Active (CreatePackage)' and 'Platform: Active (Any CPU)'. Below these, there is a checked checkbox labeled 'Generate package during build'. Underneath, there are several input fields: 'Package Name' (containing 'TDSProject'), 'Package Author', 'Package Publisher', and 'Package Version'. A large text area for 'Package Readme' is also present. At the bottom, there is a dropdown for 'Package Generation Options' set to 'Generate a package with compiled code and items', a checkbox for 'Append the current date and time to the package name' which is unchecked, and a text field for 'Sitecore Assembly Path' containing '..\packages\'. A small note below this field explains that if left blank, the builder will look for Sitecore dll's in the \$(SolutionDir)/packages folder.

- 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

Type of VCS

Type of VCS:

VCS Root

VCS root name:
A unique name to distinguish this VCS root from other roots.

VCS root ID: [Regenerate ID](#)
VCS root ID must be unique across all VCS roots. VCS root ID can be used in parameter references to VCS root parameters and REST API.

TFS Settings

URL:
URL format:
TFS 2010+: http[s]://<TFS Server>:<Port>/tfs/<Project Collection Name>
TFS 2005/2008: http[s]://<TFS Server>:<Port>
Visual Studio Online: http[s]://[account_name].visualstudio.com/DefaultCollection

Root:
TFS path to checkout. Format: \$/path.

Username:
⚠ Leave blank to use the TeamCity server user account.
To login to hosted TFS use "ALTERNATE.AUTHENTICATION.CREDENTIALS" and add "##LIVE##" prefix to your username (email address) Ⓞ

Password:
⚠ Leave blank to use the TeamCity server user account

Agent checkout: Enforce overwrite all files

Changes Checking Interval

Checking interval: use global server setting (60 seconds)
 custom: seconds

Please note that certain servers can refuse access if polled too frequently. Consider intervals greater than 1800 seconds (30 minutes) for public servers.

VCS Root Project

Belongs to project:

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.

Build Step (1 of 5): NuGet Restore | ▾

| | |
|-------------------------|--|
| Runner type: | NuGet Installer <small>Installs and updates missing NuGet packages</small> |
| Step name: | NuGet Restore <small>Optional, specify to distinguish this build step from other steps.</small> |
| Restore Packages | |
| Path To Solution File*: | TDS with TC and Octopus/TDS with TC and Octopus.sln <small>The path to Visual Studio solution file (.sln)</small> |

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

Build Step (2 of 5): MS Build Step | ▾

| | |
|----------------------------------|--|
| Runner type: | MSBuild <small>Runner for MSBuild files</small> |
| Step name: | MS Build Step <small>Optional, specify to distinguish this build step from other steps.</small> |
| Build file path: * | TDS with TC and Octopus/TDS with TC and Octopus.sln <small>The specified path should be relative to the checkout directory.</small> |
| MSBuild version: | Microsoft Build Tools 2015 |
| MSBuild ToolsVersion: | none |
| Run platform: | x86 |
| Targets: | <input type="text"/> <small>Enter targets separated by space or semicolon.</small> |
| Command line parameters: | /p:Configuration=CreatePackage <small>Enter additional command line parameters to MSBuild.exe.</small> |
| .NET Coverage | |
| .NET Coverage tool: [Ⓜ] | <No .NET Coverage> <small>Choose a .NET coverage tool. ⚠ Test code coverage is supported only for NUnit tests run using TeamCity facilities. [Ⓜ]</small> |
| Octopus Packaging | |
| Run OctoPack: | <input type="checkbox"/> <small>If checked, any projects with OctoPack installed will be packaged.</small> |
| OctoPack package version: | <input type="text"/> <small>Package version number for NuGet packages created by OctoPack.</small> |

5. Configure TC to create a NuGet Package

Build Step (3 of 5): NuGet Pack ▾

Runner type:
Creates a NuGet package from a given spec file

Step name:
Optional, specify to distinguish this build step from other steps.

Execute step:
Specify the step execution policy.

NuGet Settings

NuGet.exe:
The path to NuGet.exe relative to the checkout directory. Check installed NuGet Command line tools in [NuGet Settings](#)

Package parameters

Specification files:
Specification or project files:

Specify paths to .nuspec files and/or to Visual Studio project files (i.e. .csproj or .vbproj). MSBuild-style wildcards are supported

Prefer project files to .nuspec
Use the project file (if exists, i.e. .csproj or .vbproj) for every matched .nuspec file

Version:
Overrides the version number from the nuspec file.

Base Directory:
Do not add an explicit -BaseDirectory parameter

Output

Output Directory:
The path to the output directory for generated NuGet packages. See also [NuGet Publish](#) build runner

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

Build Step (4 of 5): Push | ▾

Runner type:
 Pushes package files (.nupkg, .zip, .tar.gz, etc.) to an Octopus Deploy server

Step name:
 Optional, specify to distinguish this build step from other steps.

Execute step:
 Specify the step execution policy.

Octopus Connection

Octopus URL:
 Specify Octopus web portal URL

API key:
 Specify Octopus API key. You can get this from your user page in the Octopus web portal.

Package Push

Package paths:
 Package path patterns:

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

Build Step (5 of 5): Octopus Release | ▾

Runner type:
 Creates and, optionally, deploys releases in Octopus Deploy

Step name:
 Optional, specify to distinguish this build step from other steps.

Execute step:
 Specify the step execution policy.

Octopus Connection

Octopus URL:
 Specify Octopus web portal URL

API key:
 Specify Octopus API key. You can get this from your user page in the Octopus web portal.

Octopus version:
 Which version of the Octopus Deploy server are you using?

Release

Project:
 Enter the name of the Octopus project to create a release for

Release number:
 The number to use for this release, e.g., 1.0.%build.number%.

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:

The screenshot shows the Octopus Server interface. At the top, there's a navigation bar with 'Dashboard', 'Environments', 'Projects', 'Library', 'Tasks', and 'Configuration'. Below this, the 'Environments' section is visible, showing two environments: 'Dev' and 'QA'. The 'Dev' environment is selected, and the page title is 'Environments > Deployment Targets > Development'. The 'Settings' tab is active, and the 'Connectivity' sub-tab is selected. The main content area is titled 'Deployment Target settings' and contains three sections: 'Deployment', 'Communication', and 'Communication'. The 'Deployment' section has 'Environments' set to 'Dev' and 'Roles' set to 'Development server'. The 'Communication' section has 'Style' set to 'Listening Tentacle', 'Thumbprint' set to '29F93D0E27C592D674A554', and 'Tentacle URL' set to 'https://:10933/'. There is also a checkbox for 'Disable this deployment target'.

Environments > Deployment Targets > Development

Settings Connectivity

Deployment Target settings

Display name
A unique, informational name for the deployment target.

Deployment

Environments
Specify at least one environment for the deployment target.

Roles
Specify at least one role that this deployment target will provide.

Communication

Style ▼
Choose alternative styles when the Tentacle agent can't be installed on the remote machine.

Thumbprint
The X509 certificate thumbprint that securely identifies the machine. You can confirm the validity of the thumbprint by opening the Tentacle administration tool on the discovered machine.
Please ensure the Tentacle is configured to trust an Octopus Server with the thumbprint:
87947EDCE97223BEF740F1

Tentacle URL
The network address at which the Tentacle can be reached.


Disable this deployment target
Disable a deployment target to prevent releases or deployments being made to it.

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

Projects > Create New

Create a project

Name

Description **B I** 

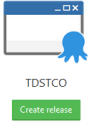
Project group ▼

Lifecycle ▼

Lifecycles determine which environments the project can be deployed to, and the promotion rules between those environments. [Create or modify lifecycles.](#)

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

TDSTCO · Process



TDSTCO
[Create release](#)


Overview
Process

Deployment process

- 1. Deploy to Dev**
Deploy NuGet package UpdatePackage from Octopus Server (built-in) to machines in roles: [Development server](#)
Only in: [Dev](#)
- 2. Deploy to QA**
Deploy NuGet package UpdatePackage from Octopus Server (built-in) to machines in roles: [QA server](#)
Only in: [QA](#)

[Add step](#) [Reorder steps](#)

One deploying to Dev



TDSTCO

Create release

- Overview
- Process
- Variables
- Channels
- Releases
- Settings

Step details

Step name

Runs on target roles

By default, this step will be run simultaneously on all deployment targets. [Configure a rolling deployment.](#)

Package

This step is used to deploy the contents of a NuGet package to one or more machines. You can configure the remote machines to deploy to in the **environments** tab. The packages should contain, and how to [create them](#).

NuGet feed

Select the NuGet feed that this package will be found in.

NuGet package ID

Enter the ID of a NuGet package to deploy.

Custom install directory

Install to

After the NuGet package is extracted, it will be copied to this location on the remote machine.

Purge Purge this directory before installation

Before the contents of the extracted NuGet package is copied, all files in this location will be removed.

Octopus comes with a variety of built-in features that help to make your deployments easier. You can enable these features by adding them below.


[Configure features](#)

Conditions

Environments

When you deploy your project to a given environment, this step will be skipped if the environment is not in the list above.

And another one, deploying to QA



TDSTCO

Create release

- Overview
- Process
- Variables
- Channels
- Releases
- Settings

Step details

Step name

Runs on target roles

By default, this step will be run simultaneously on all deployment targets. [Configure a rolling deployment.](#)

Package

This step is used to deploy the contents of a NuGet package to one or more machines. You can configure the remote machines to deploy to in the **environments** tab. The NuGet should contain, and how to [create them](#).

NuGet feed

Select the NuGet feed that this package will be found in.

NuGet package ID

Enter the ID of a NuGet package to deploy.

Custom install directory

Install to

After the NuGet package is extracted, it will be copied to this location on the remote machine.

Purge Purge this directory before installation

Before the contents of the extracted NuGet package is copied, all files in this location will be removed.

Octopus comes with a variety of built-in features that help to make your deployments easier. You can enable these features by adding them below.

[Configure features](#)

Conditions

Environments

When you deploy your project to a given environment, this step will be skipped if the environment is not in the list above.

Run condition

Limit when this step runs by setting a condition.

Start trigger

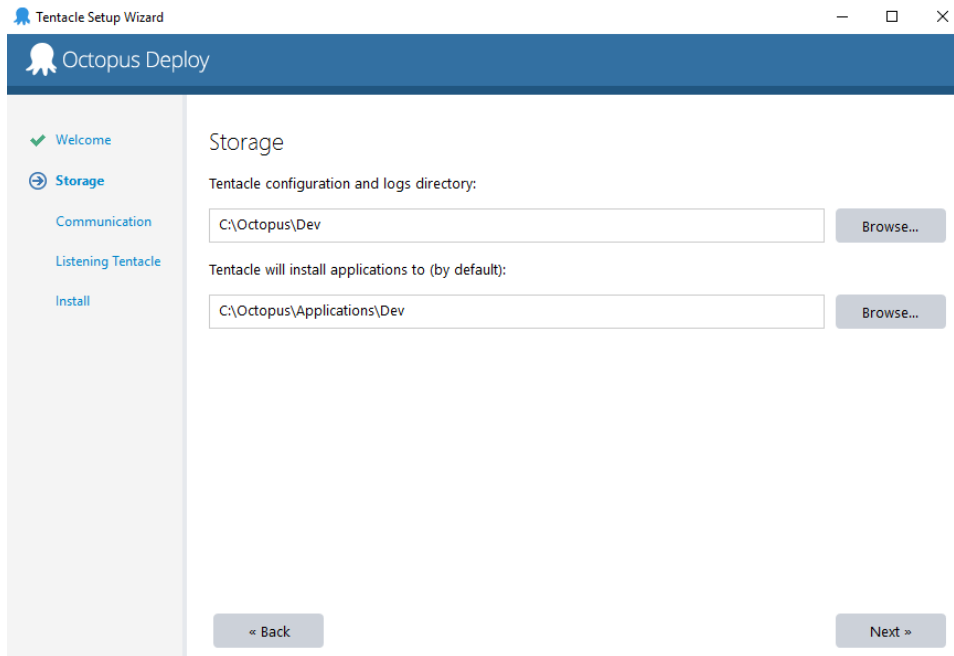
Control whether this step waits for the previous step to complete, or runs in parallel with it.

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

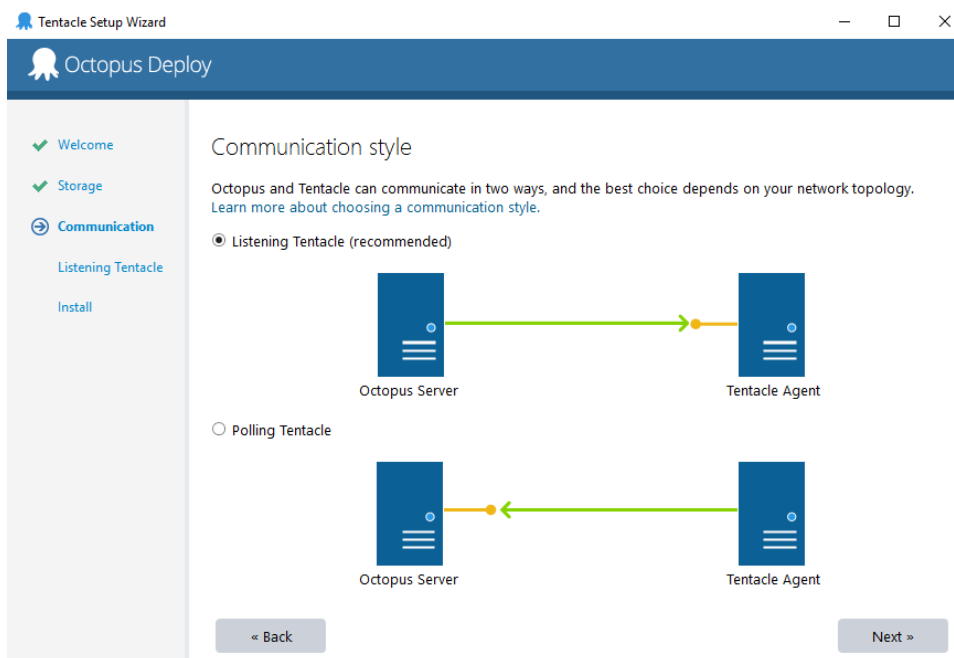
On the Sitecore servers

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.

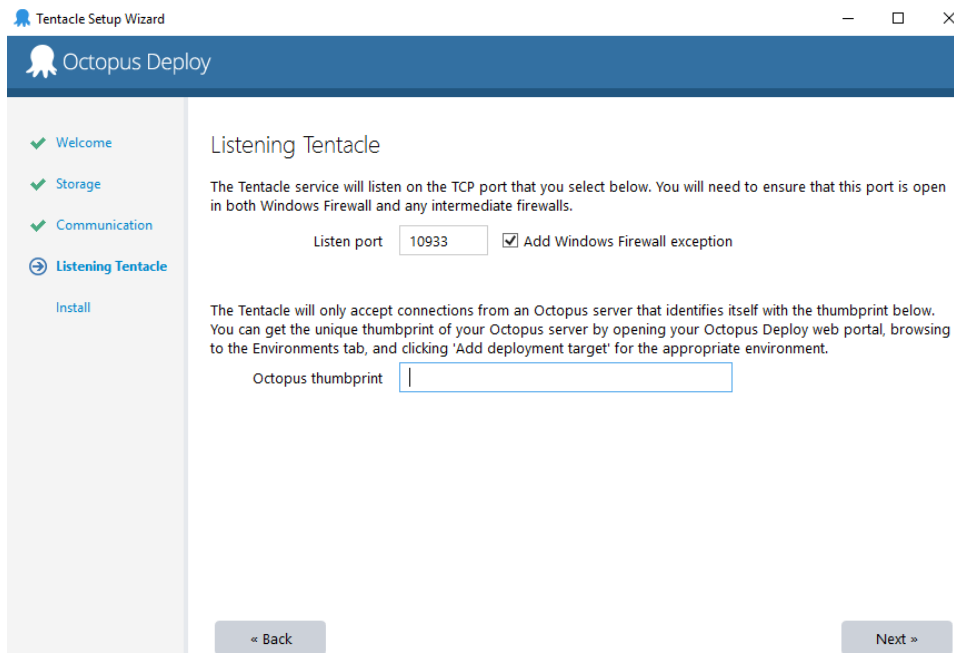
The Thumbprints in Octopus and the Tentacle must be the same.



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



Select "Listening Tentacle"

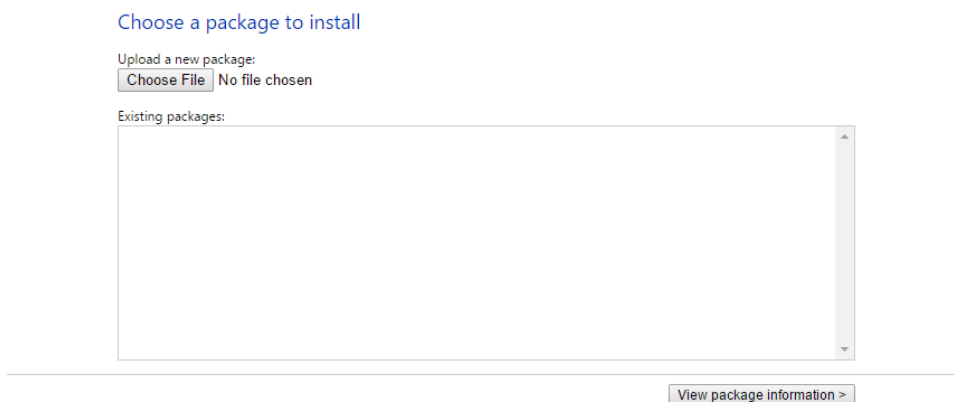


Select the port and paste the thumbprint from the Deployment Target settings in Octopus

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.