

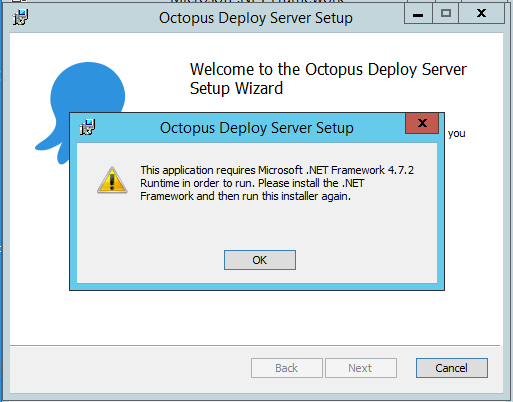
The Octopus Deploy server is available as a self-hosted instance that you install and manage on your infrastructure. This section walks you through installing the **self-hosted** Octopus Deploy server.

Download the Octopus Installer

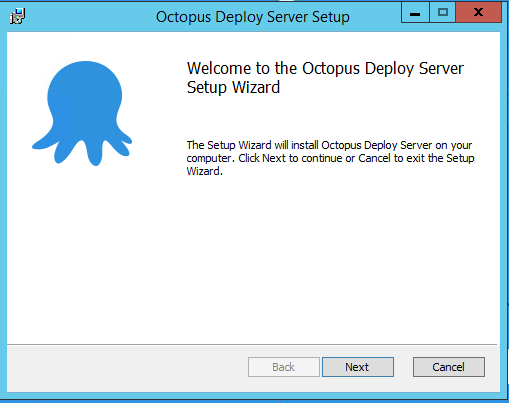
**URL:** **https://octopus.com/downloads/server**

Install Octopus:-

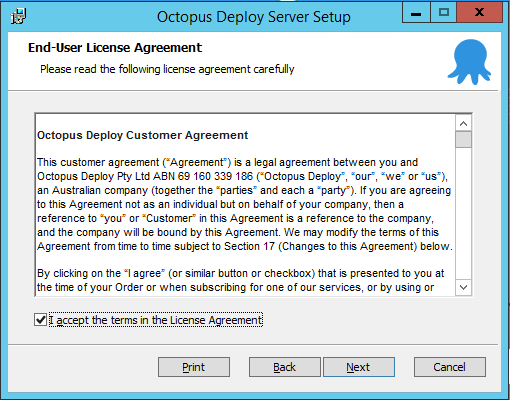
Start the Octopus Installer



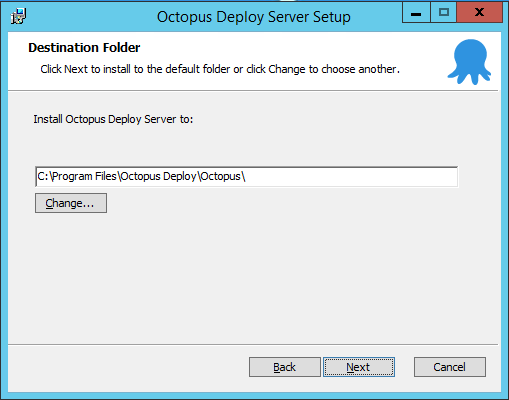
**Note**: .Net Framework 4.7.2 or Higher is required for this installation



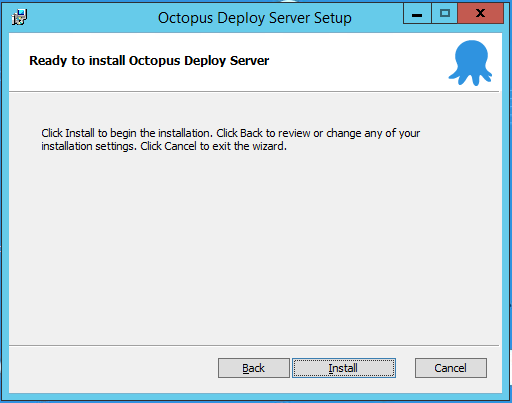
Click **Next**, accept the **Terms in the License Agreement** and click **Next**.

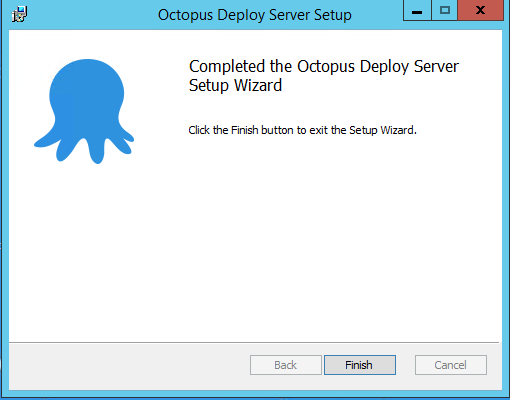


Accept the default **Destination Folder** or choose a different location and click **Next**.

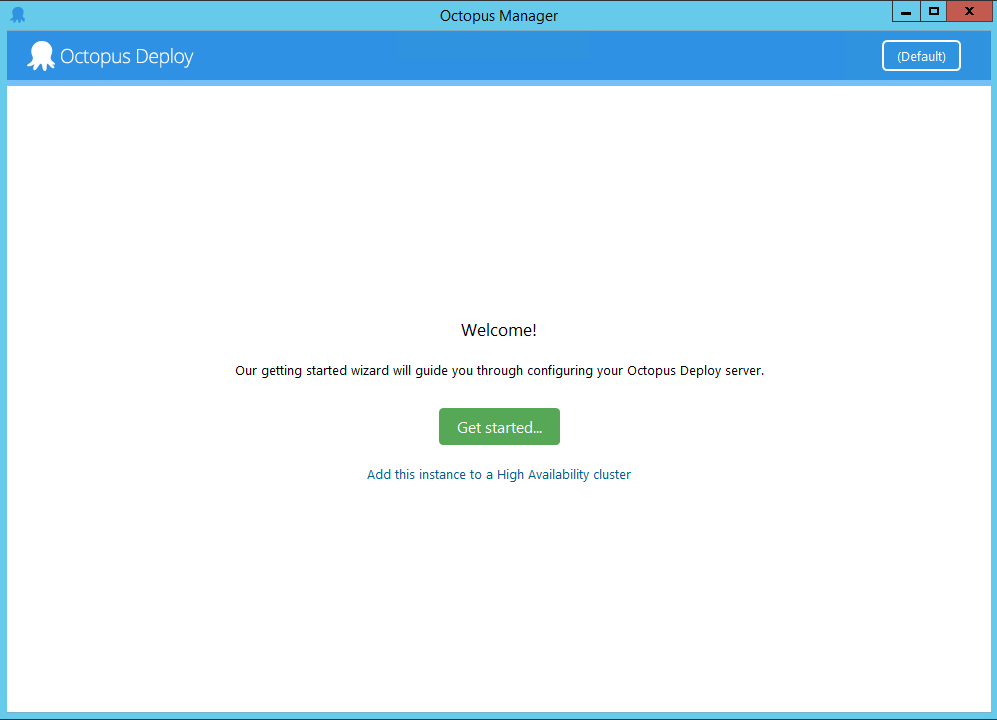


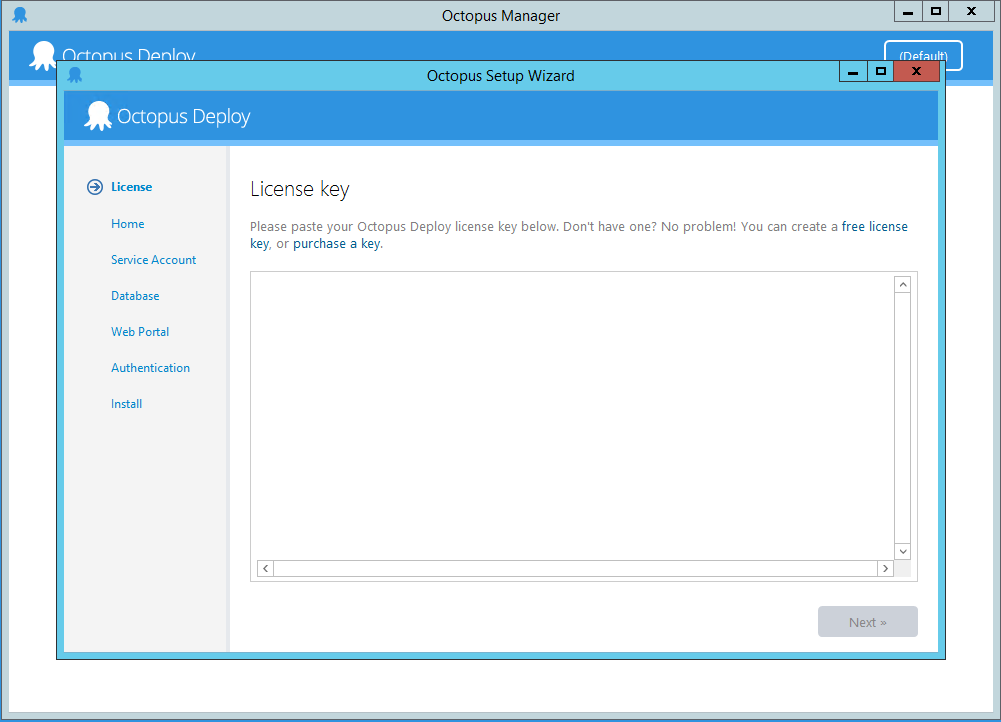
Click **Install**, and give the app permission to **make changes to your device**.





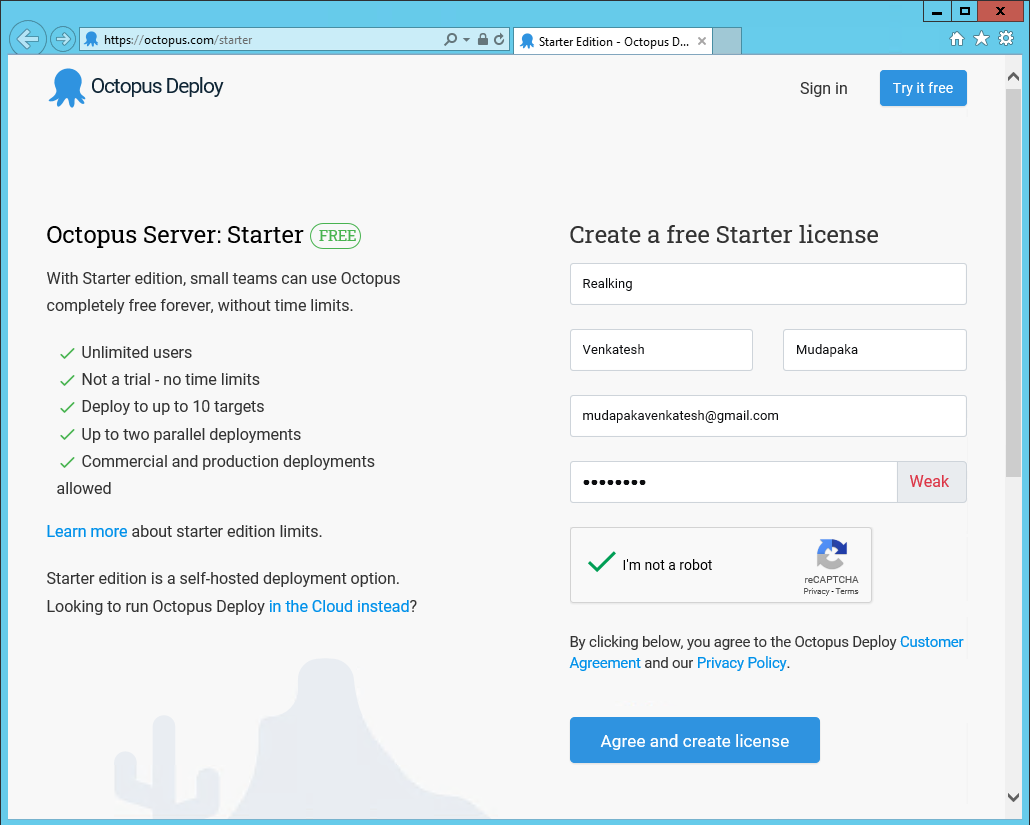
Click **Finish** to exit the installation wizard and launch the **Getting started wizard** to configure your Octopus Deploy Server.



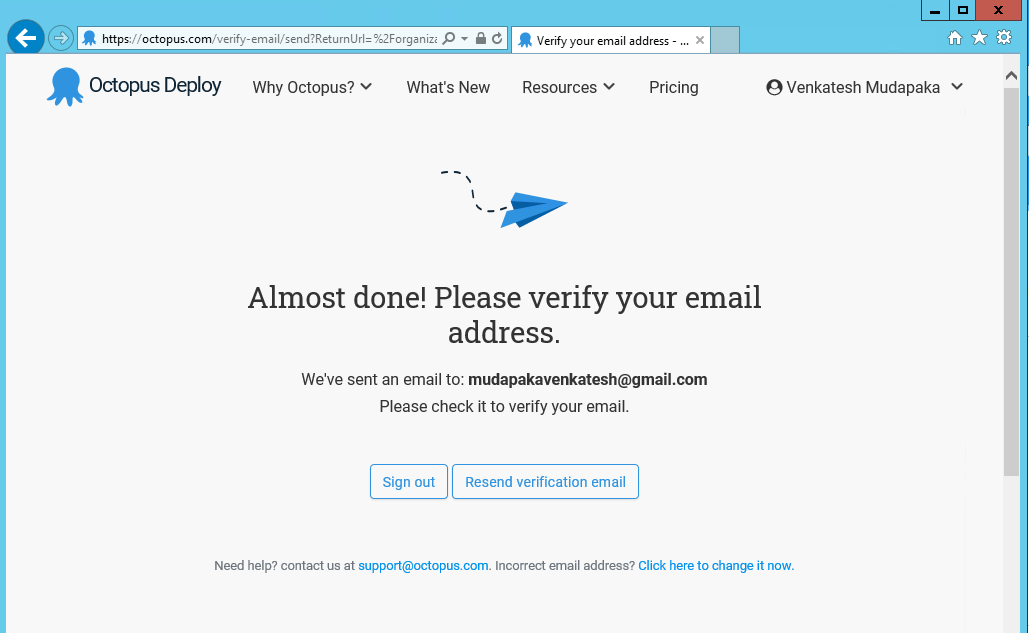
Click **Get started...** and either enter your details to start a free trial of Octopus Deploy or enter your **license key** and click **Next**.

## Step to Create a Free License key:-

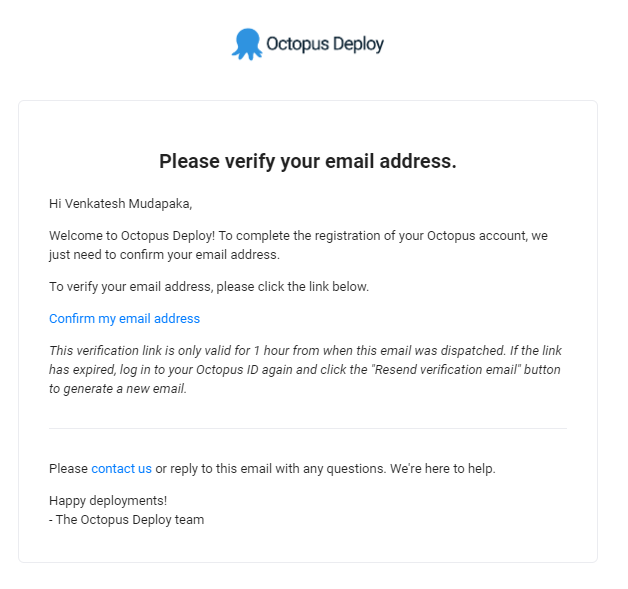
Enter your **Org name**, **First name**, **Last name**, **email** and **desired password** to sign up. Click **Agree and create license**.



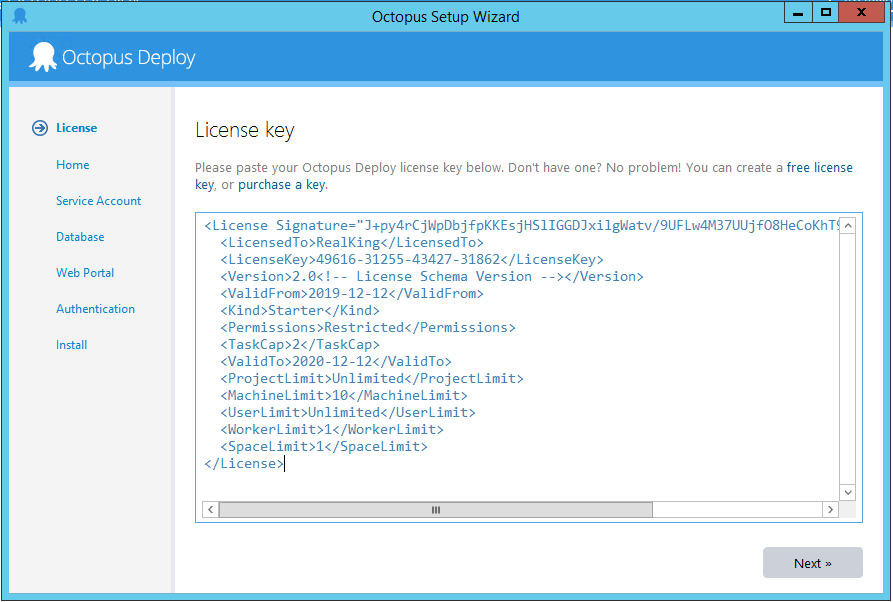
A verification email will be sent out to your email and need to verify the same to activate your account.



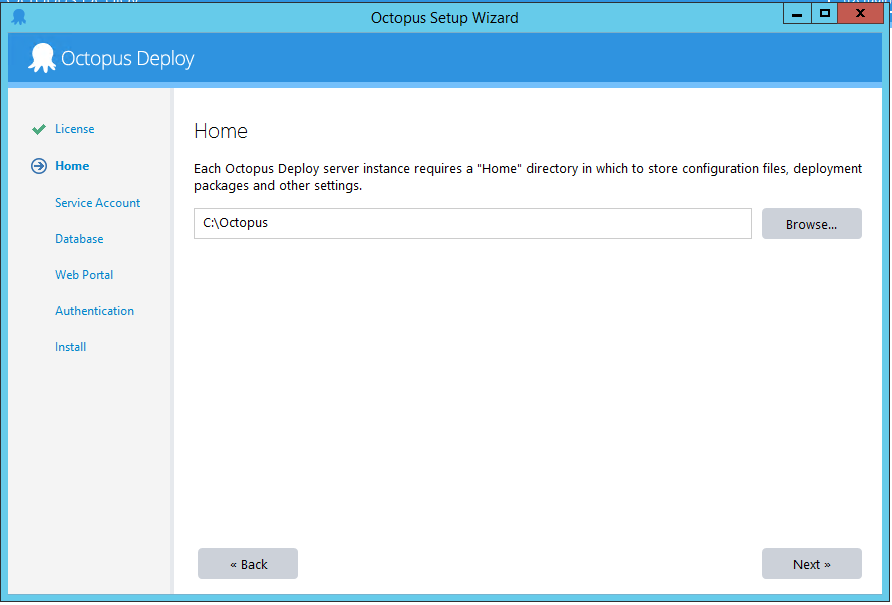
Open the email and click on the **Confirm my email address**.



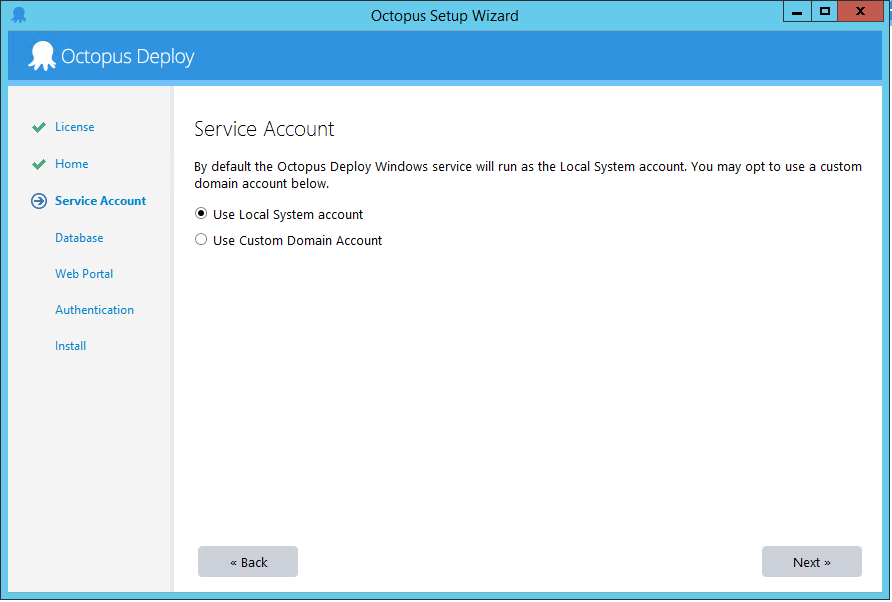
Once you Confirm the license a **license key** will be provided to continue the installation.



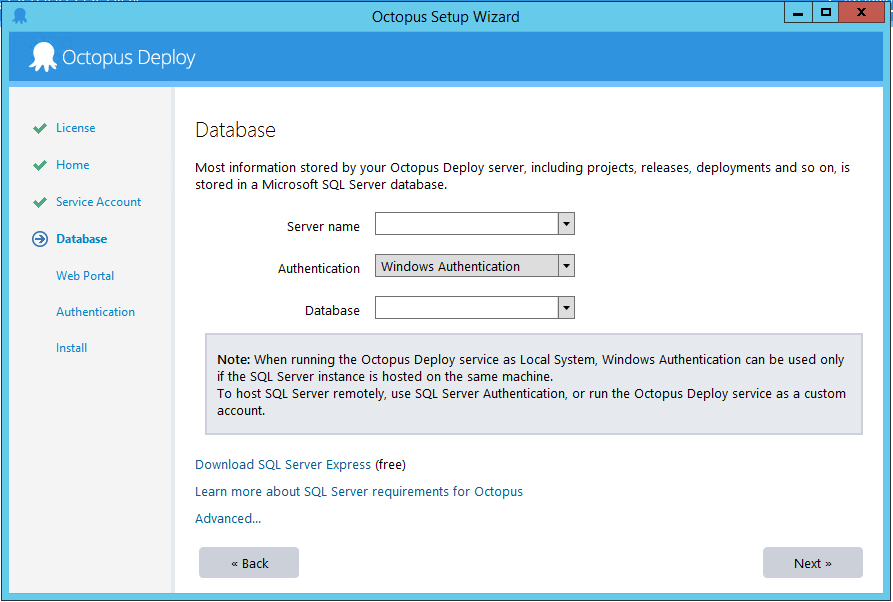
Accept the default **Home Directory** or enter a location of your choice and click **Next**.



Decide whether to use a **Local System Account** or a **Custom Domain Account**.



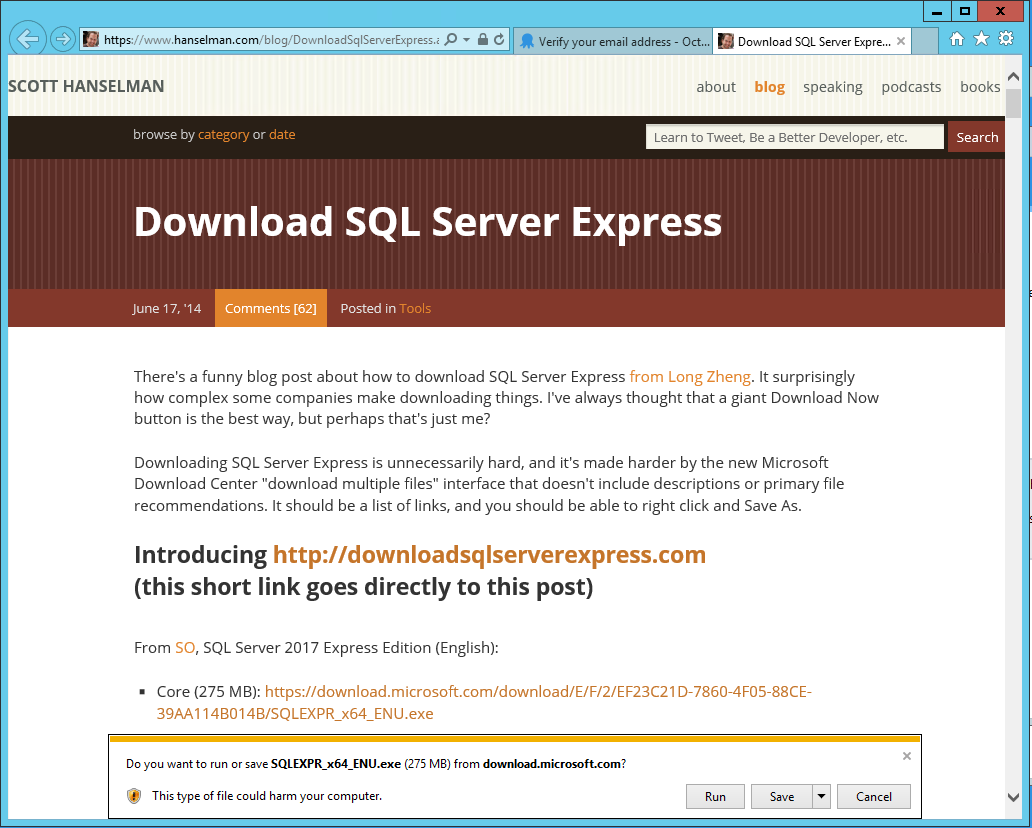
On the **Database** page, click the dropdown arrow in the **Server Name** field to detect the SQL Server Database. Octopus will create the database for you which is the recommended process; however, you can also [create your own database](https://octopus.com/docs/installation/sql-server-database#creating-the-database).



Octopus Deploy stores **projects**, **environments**, and **deployment** **history** in a **Microsoft SQL Server database**.

## Install Your Own Database:-

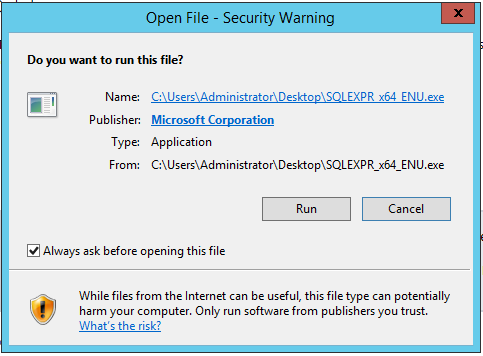
**Go to:** <https://www.hanselman.com/blog/DownloadSqlServerExpress.aspx>



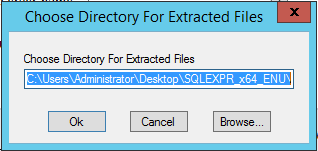
Once you download, open the **.exe**



**Accept** the security warnings if any.

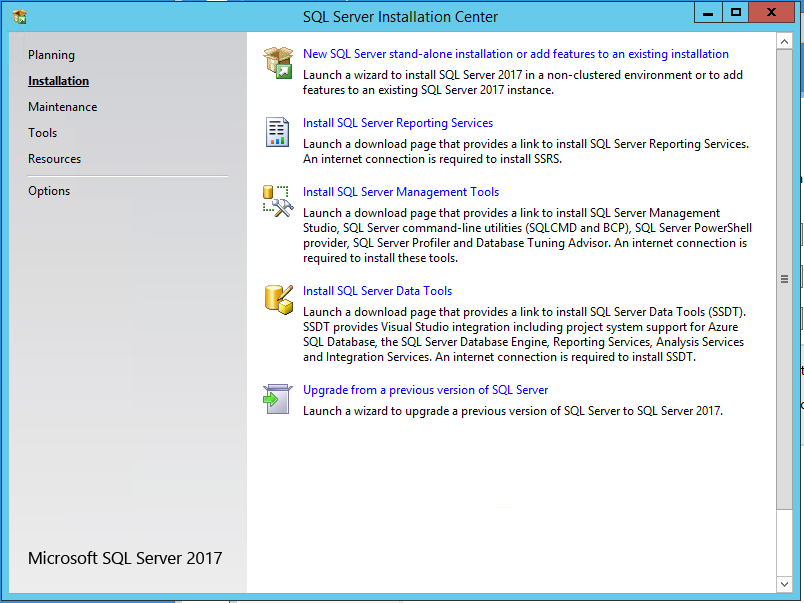


Select the **location** where the files need to be extracted and continue with the installation.

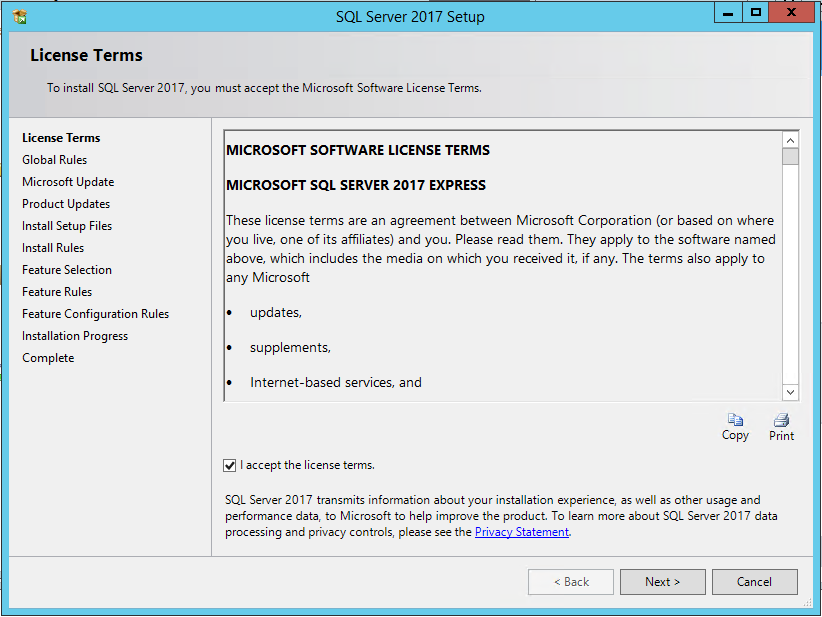


Once it is extracted, it will open the **SQL Server Installation Center**

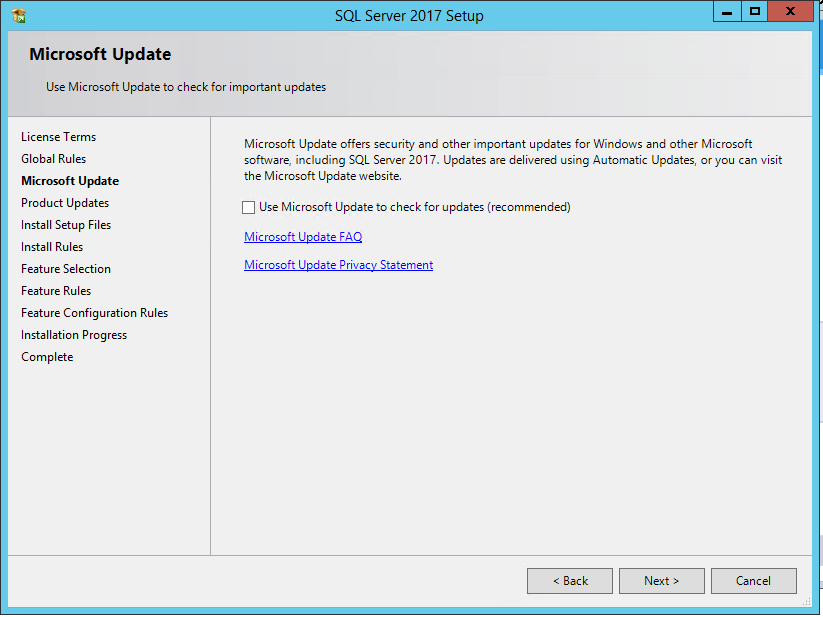
Select first option: **New SQL Server…**



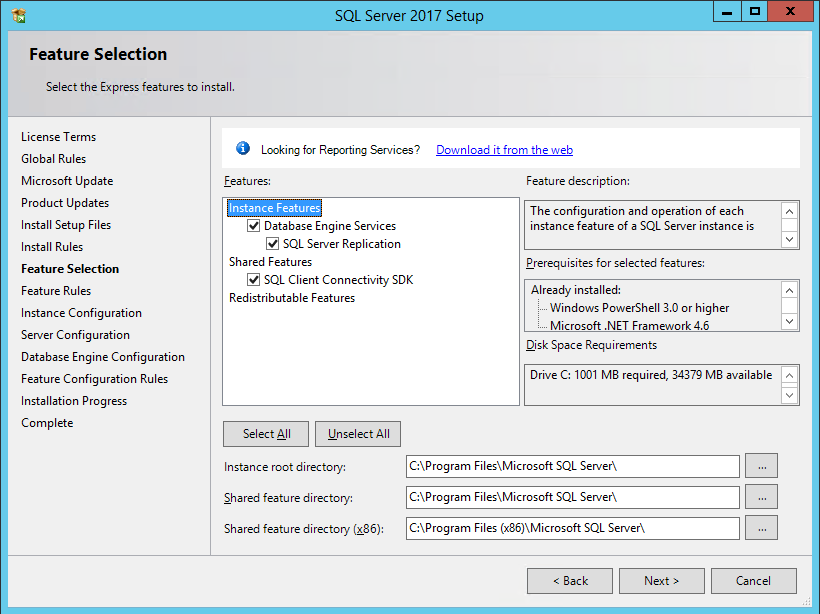
Accept the **Terms in the License Agreement** and click **Next**.



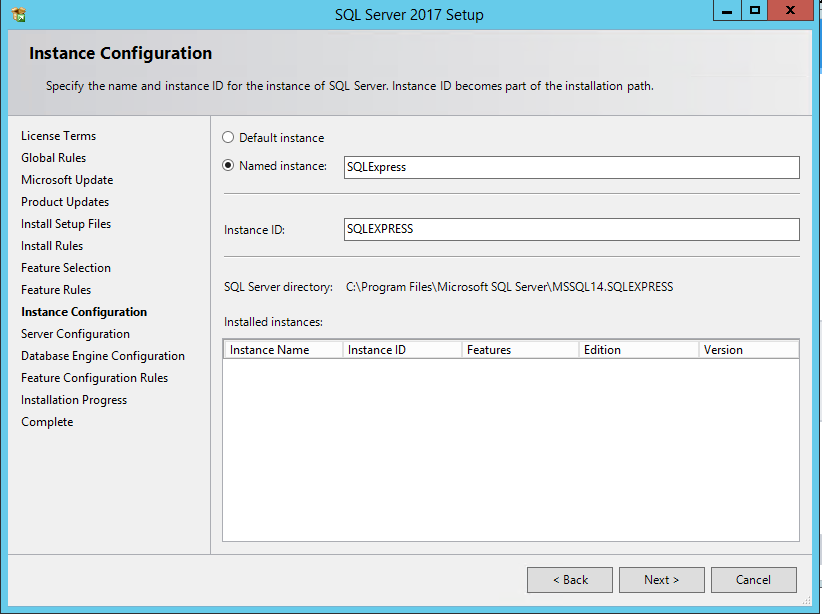
Leave it **default** and Click **Next**.



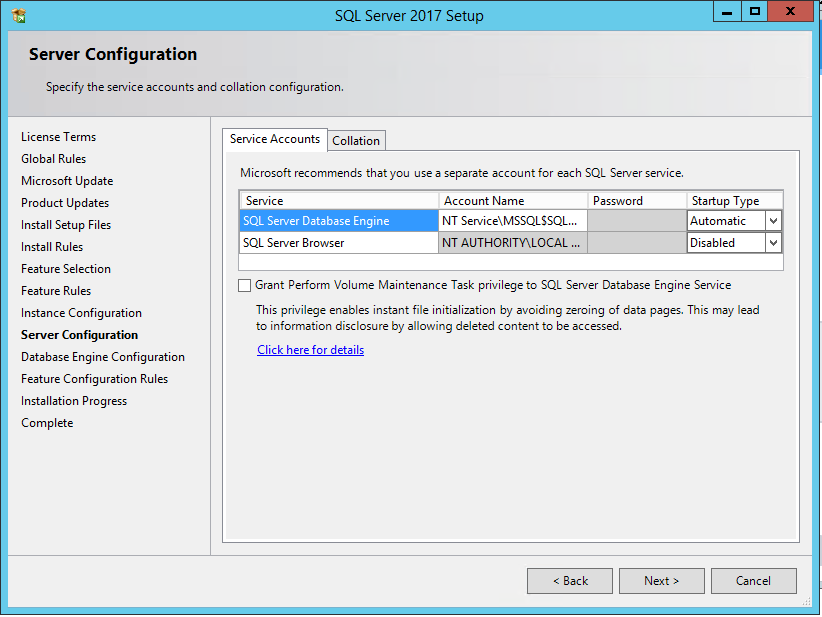
Leave it **default** and Click **Next**.



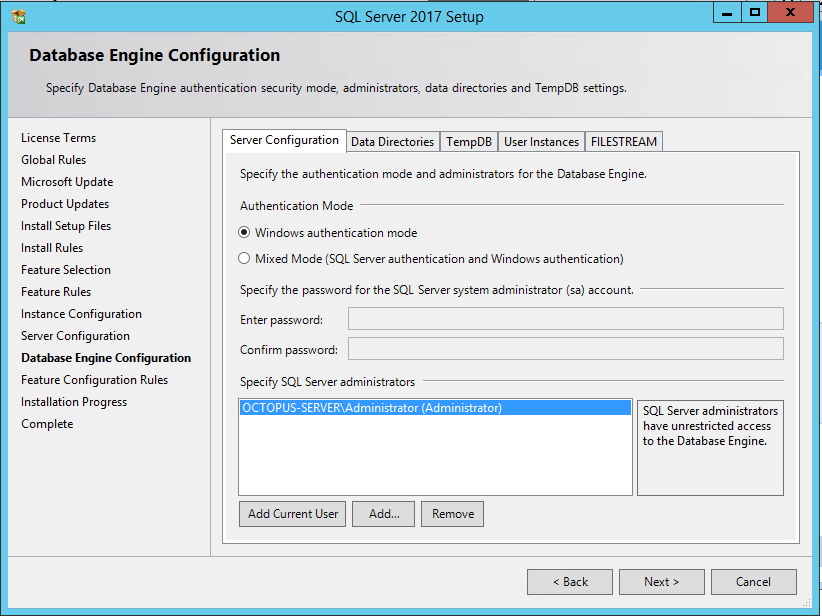
Give a **name** to **Instance** or Leave it **default** and Click **Next**.



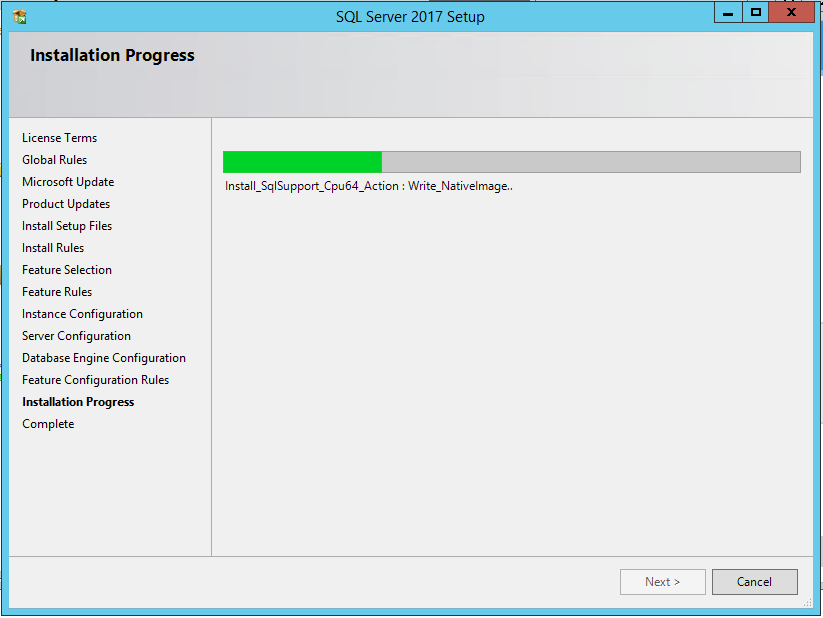
**Create accounts** for SQL or Leave it **default** and Click **Next**.



I’m using **Windows Authentication mode**, Leave it **default** and Click **Next**.



Let it install…

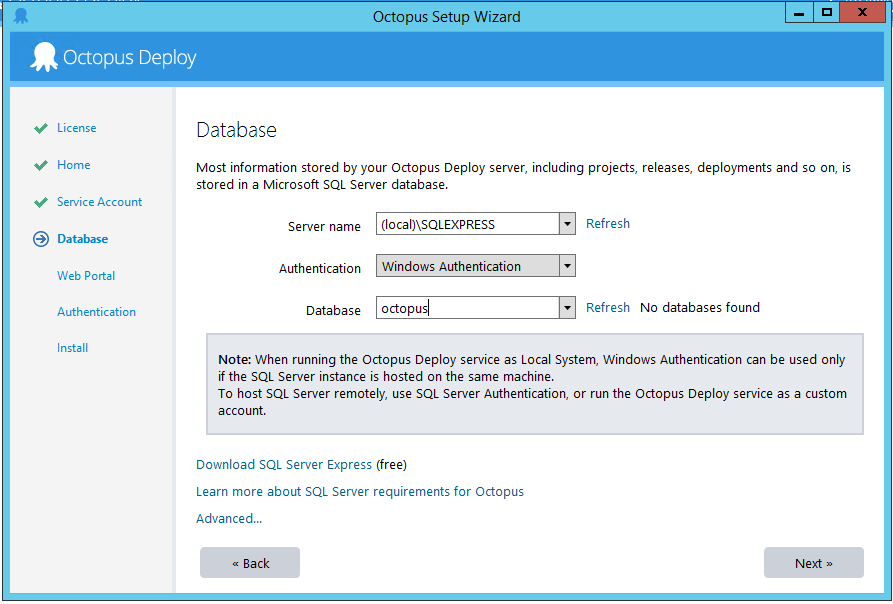


You will be displayed below when the installation is successful.

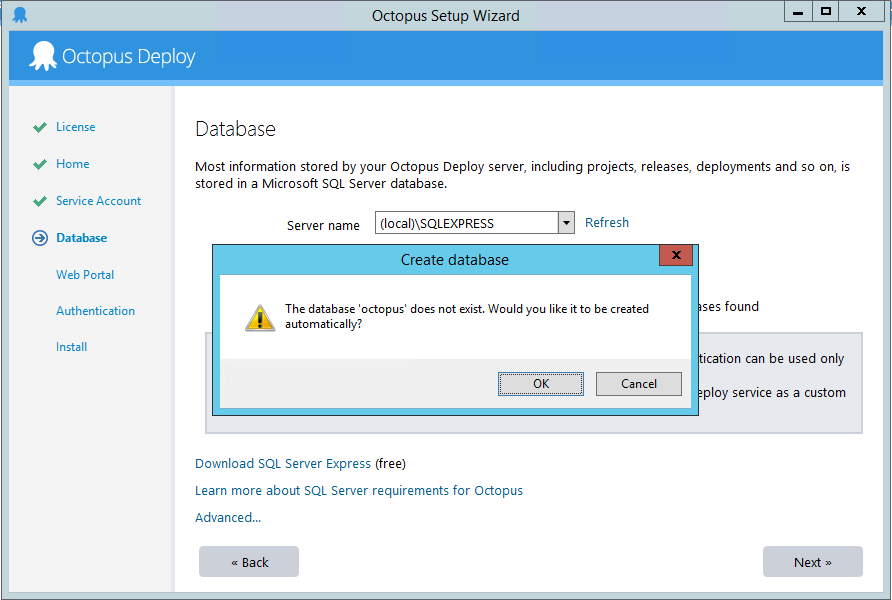


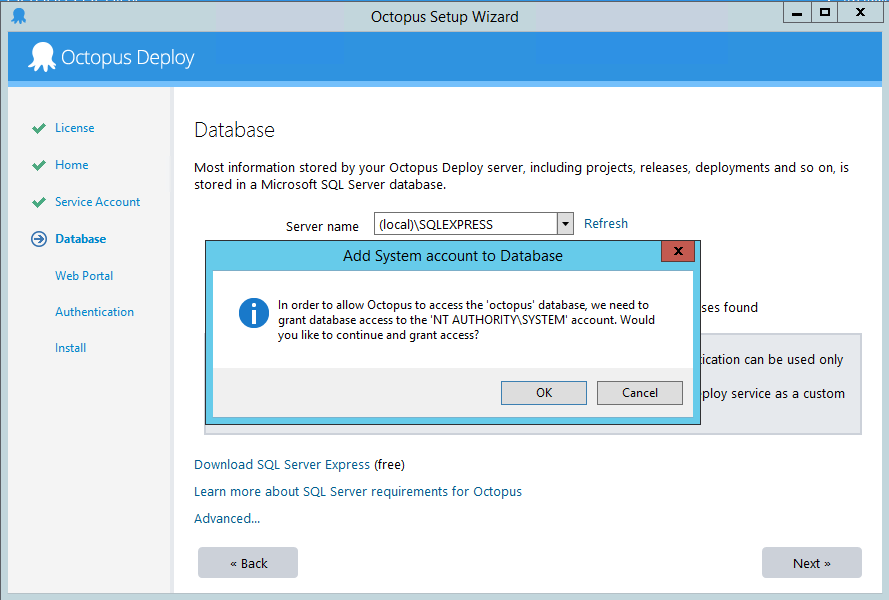
## Select or Create Your Own Database:-

Enter a name for the database, and click **Next** and **OK** to **create the database**.

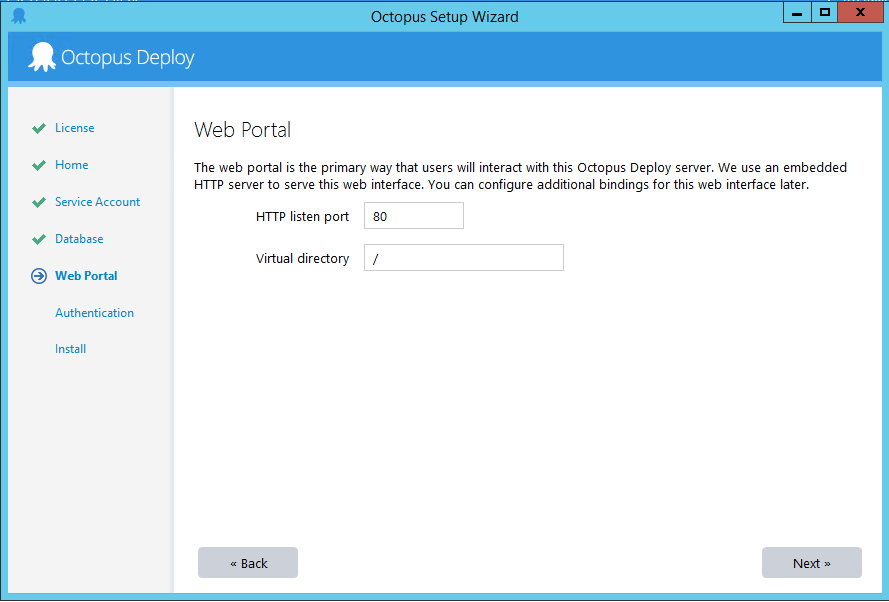


Be careful **not** to use the name of an existing database as the setup process will install Octopus into that pre-existing database.

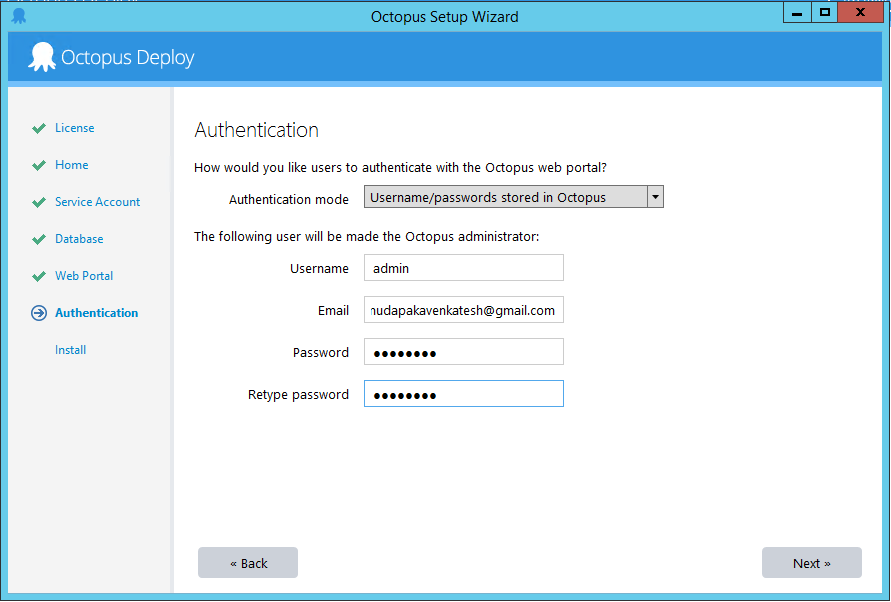




Accept the **default port** and **directory** or enter your own and click **Next**.

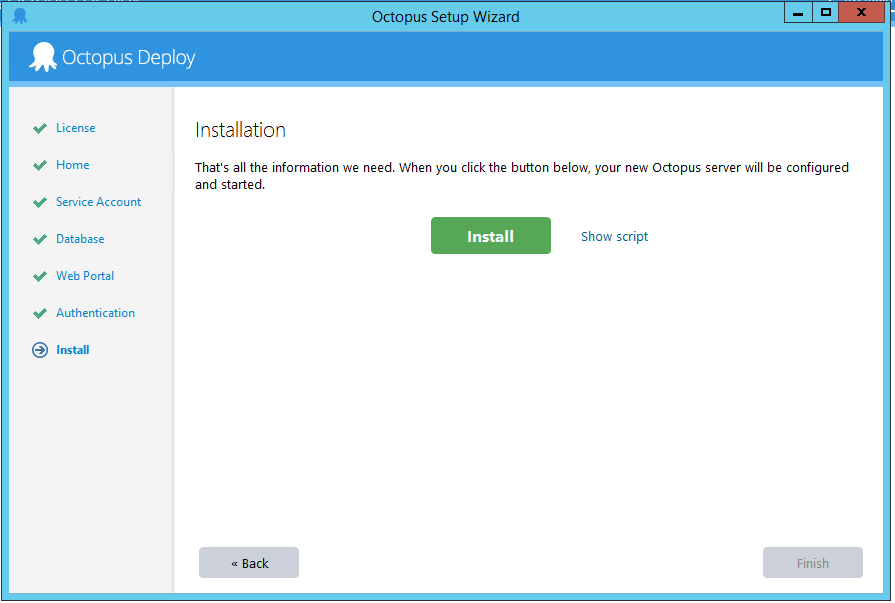


If you’re using **username and passwords stored in Octopus** authentication mode, enter the username and password that will be used for the Octopus administrator. If you are using active directory, enter the active directory user details.

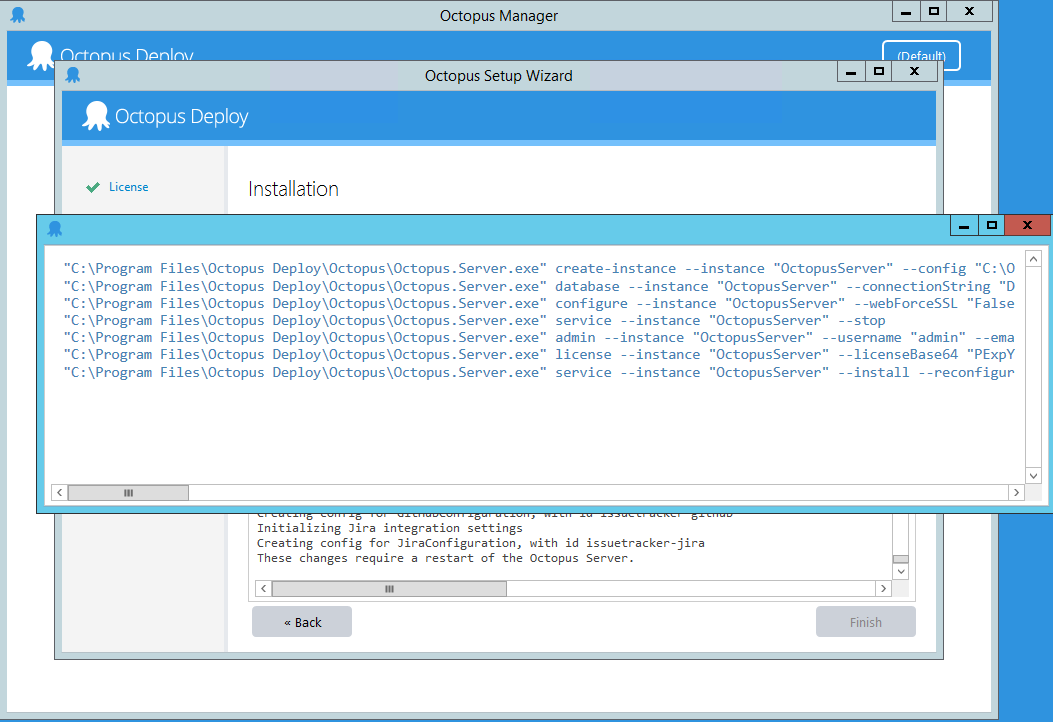


You can configure addition Authentication Providers for the Octopus Deploy server after the server has been installed.

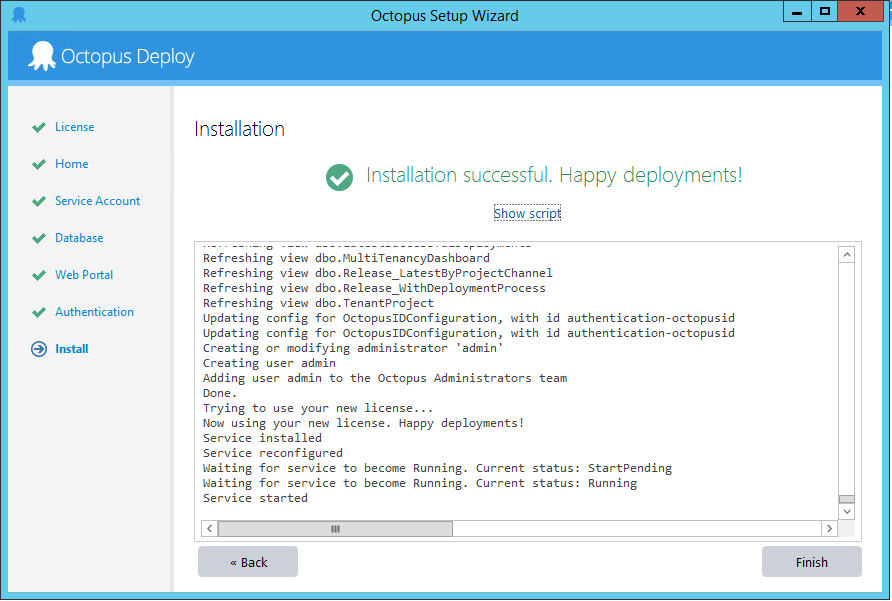
Click **Install**.



This will run some **scripts** to install Octopus Server.



Once Done, you can see the below screen.



When the installation has completed, click **Finish** to launch the **Octopus Manager**.

# Octopus Manager

Before you launch the **Octopus Web Portal**, it's worth taking note of the other settings such as controlling the Octopus Windows Service, importing and exporting the data Octopus stores in the SQL server, and viewing the master key.

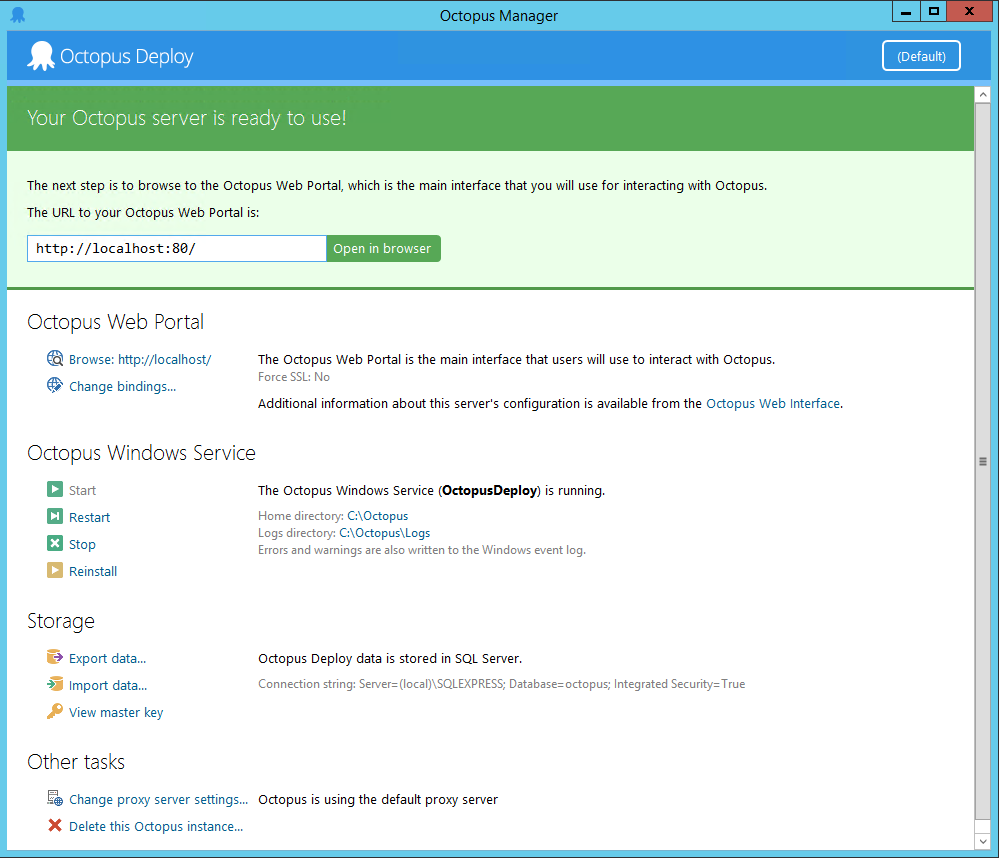
You can launch the Octopus Web Portal from the Octopus Manager, by clicking **Open in Browser**.

### Save Your Master Key

Under the storage section, you will see a link to **View Master Key**.

When Octopus is installed, it generates a master key which is a random string that is used to encrypt sensitive data in your Octopus database. You will need the master key if you ever need to restore Octopus.

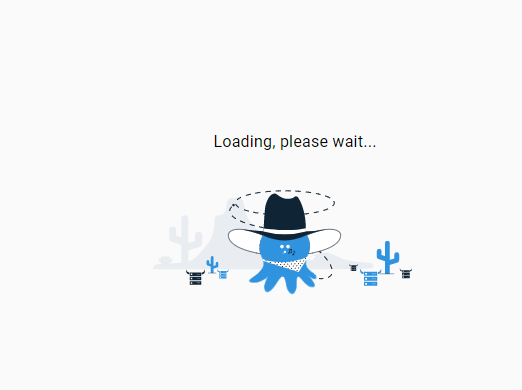
Make a copy of the master key and save it in a **secure** location.



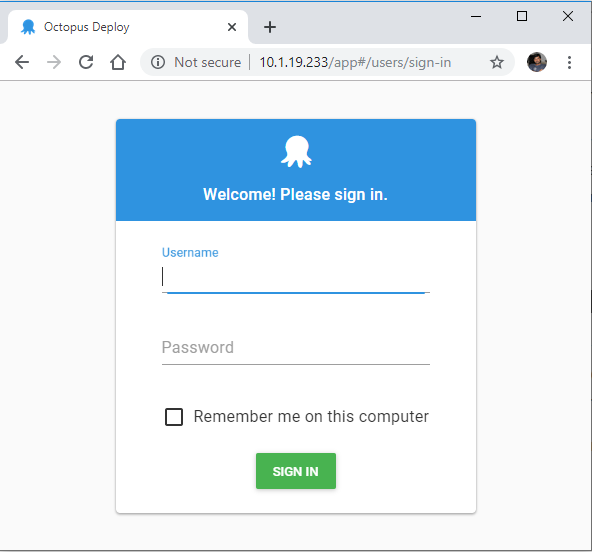
## Launch the Octopus Web Portal

The **Octopus Web Portal** is where you'll manage your infrastructure, projects, deployment process, access the built-in repository, and manage your deployments and releases.

**Troubleshooting**: If you've had any problems with the installation, review the [troubleshooting page](https://octopus.com/docs/installation/troubleshooting).



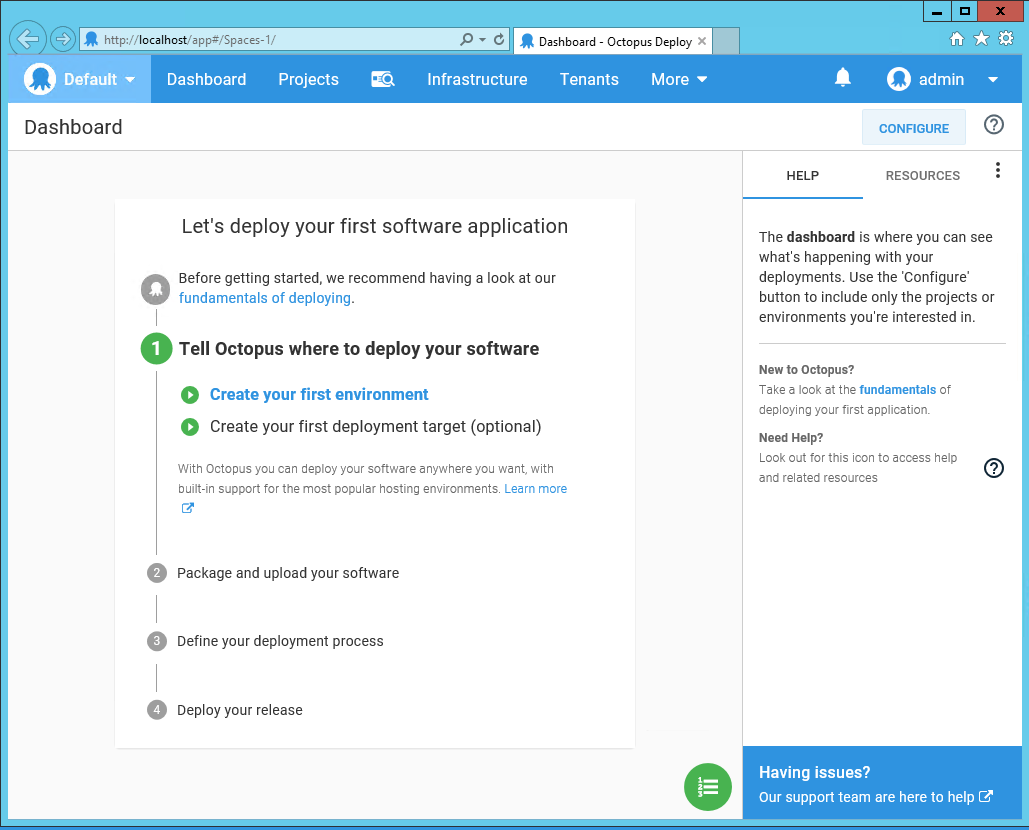
Click **Open in browser** to launch the **Octopus Web Portal** and log in using the authentication details you set up during the configuration process.



# Deploy Software

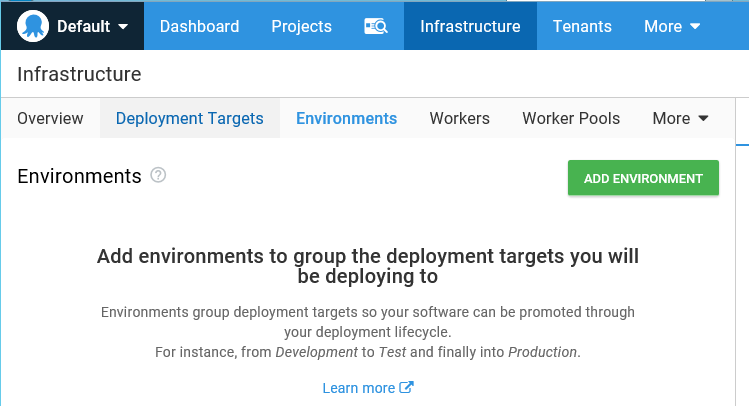
## Create your first environment:-

Environments are how you organize your deployment targets (whether on-premises servers or cloud services) into groups that represent the different stages of your deployment pipeline, for instance, **development**, **test**, and **production**.



### Infrastructure:-

With Octopus Deploy the machines and services you deploy your software to are managed on the **Infrastructure** tab of the Octopus Web Portal:



### Environments:-

Before you can add your deployment targets to Octopus, you need to configure your [environments](https://octopus.com/docs/infrastructure/environments). Environments represent the different stages of your deployment pipeline and ensure that the software that's deployed is the same as it moves through those stages from, for instance, **development**, into **test**, and finally to **production**.

Typical environments include:

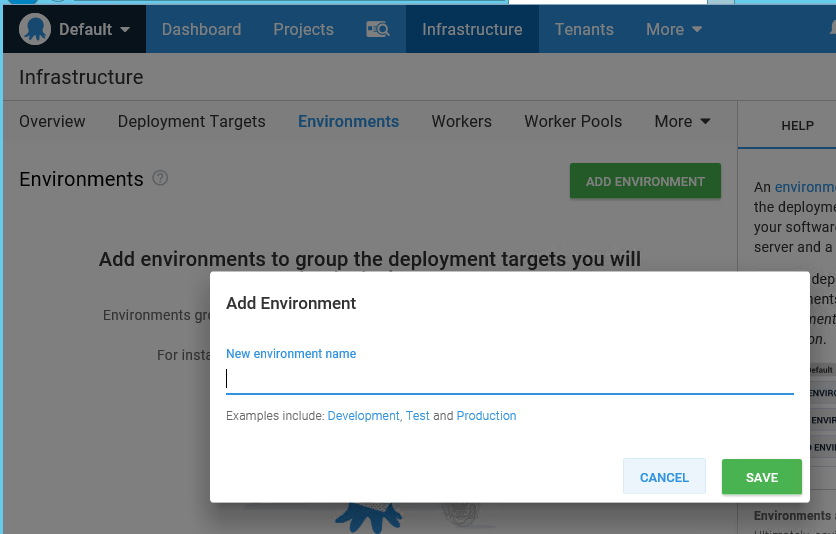
* Development
* Test
* Production

You can add as many or as few environments as you require in your deployment pipeline, and you can add as many deployment targets to each environment as you need.

Add New Environments

1. Navigate to **Infrastructure ➜ Environments** and click **ADD ENVIRONMENT**.
2. Give your new environment a meaningful name and click **SAVE**.
3. Add a description for the environment.
4. Select the checkbox in the **Default Guided Failure Mode** section if you want Octopus Deploy to prompt users for intervention if a deployment to this environment fails. Learn more about [guided failure mode](https://octopus.com/docs/deployment-process/releases/guided-failures).
5. Select the checkbox in the **Dynamic Infrastructure** section if deployments to this environment are allowed to create infrastructure such as targets and accounts. Learn more about [Dynamic Infrastructure](https://octopus.com/docs/infrastructure/deployment-targets/dynamic-infrastructure).
6. Click **SAVE**.

You can add as many environments as you need.

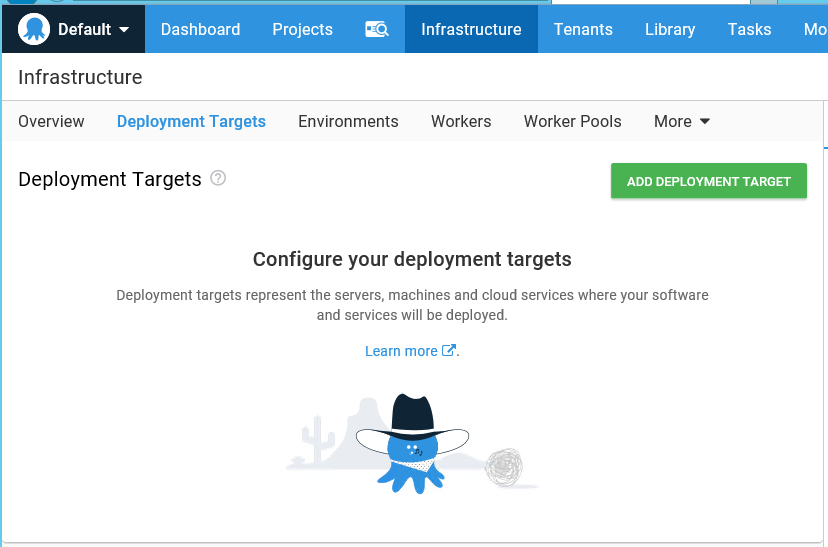


## Create your first deployment target:-

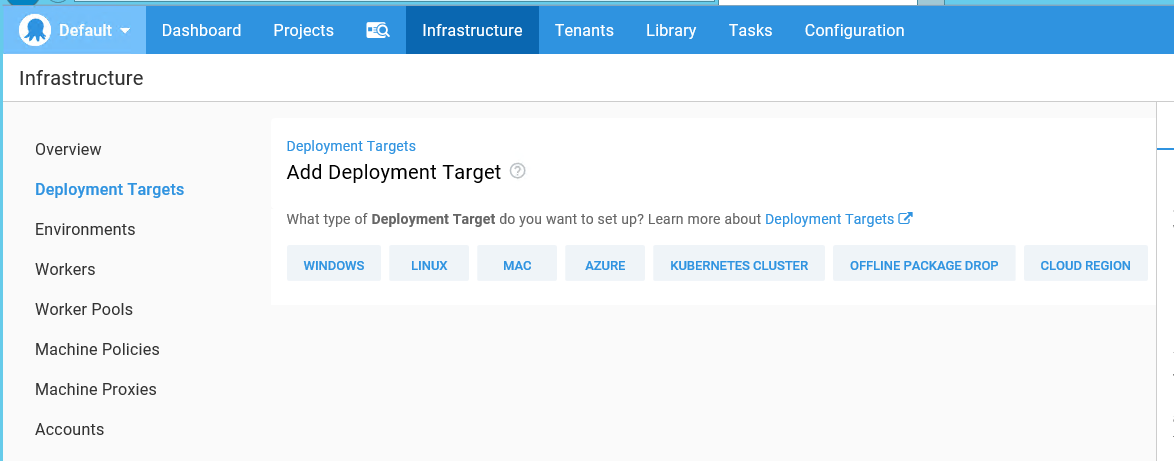
Deployment Targets

With Octopus Deploy, you can deploy software to Windows servers, Linux servers, Microsoft Azure, AWS, Kubernetes Clusters, Cloud Regions, or an Offline Package Drop. Regardless of where you're deploying your software, these machines and services are known as your deployment targets.

You can manage your deployment targets by navigating to **Infrastructure ➜ Deployment Targets** in the Octopus Web Portal:



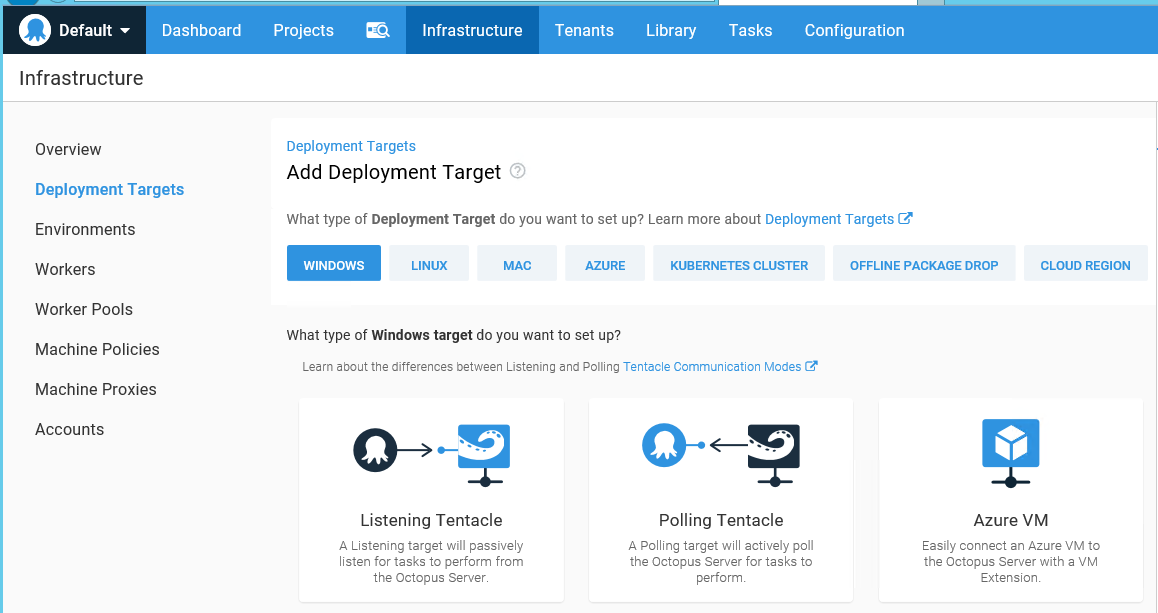
From the **Deployment Targets** tab you can add new deployment targets, disable or delete deployment targets, check on the status of your targets, and run health checks.



### Windows Targets

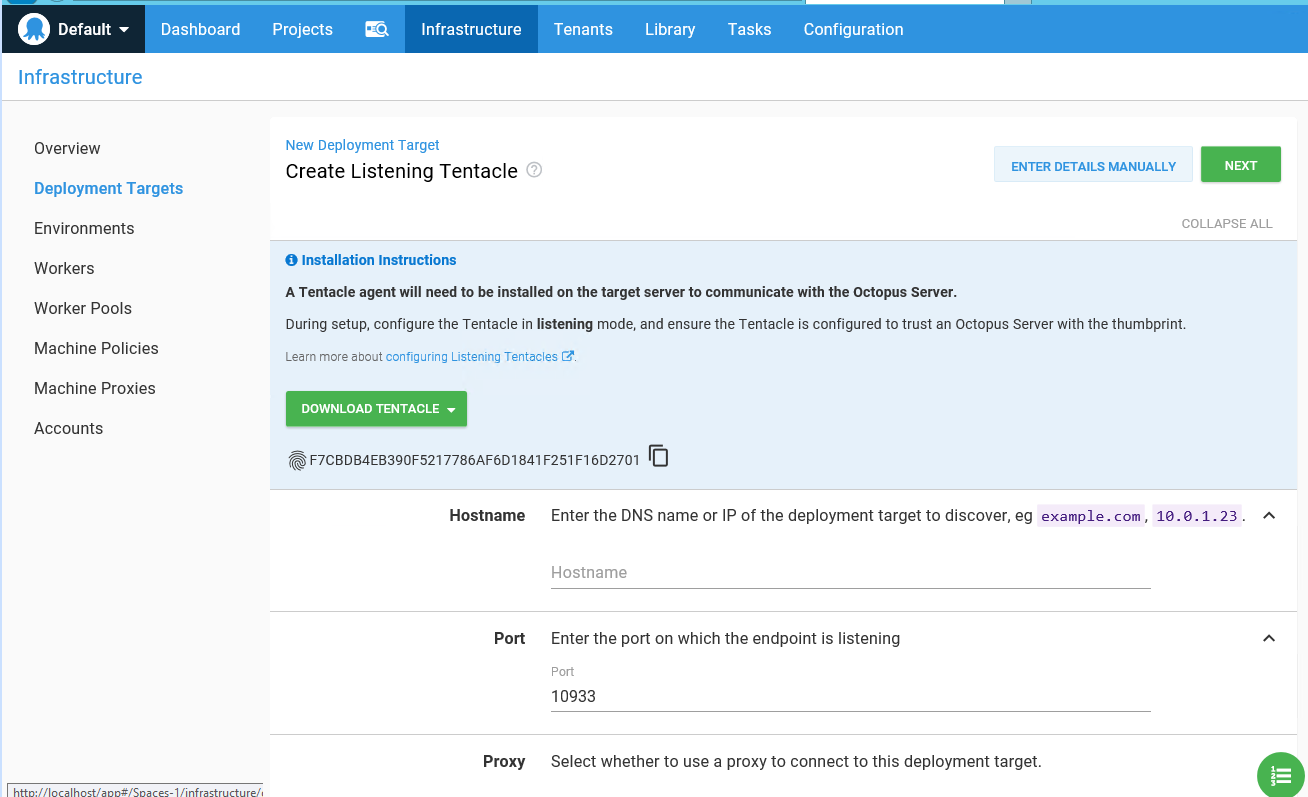
In the **Octopus Web Portal**, navigate to the **Infrastructure** tab, select **Deployment Targets** and click **ADD DEPLOYMENT TARGET ➜ WINDOWS**, and select **Listening Tentacle**.

Copy the **Thumbprint** (the long alphanumerical string).



When you **deploy software** to **Windows servers**, you need to **install Tentacle**, a **lightweight agent service**, on your **Windows** servers so they can **communicate** with the **Octopus server**.

Back on the Tentacle server, accept the default listening port **10933** and paste the **Thumbprint** into the **Octopus Thumbprint** field and click **Next**.



#### Download the Tentacle Installer:-

##### Long-term Support Octopus Server (Slow lane):-

The version of Tentacle that works with the latest long-term support version of Octopus Deploy:

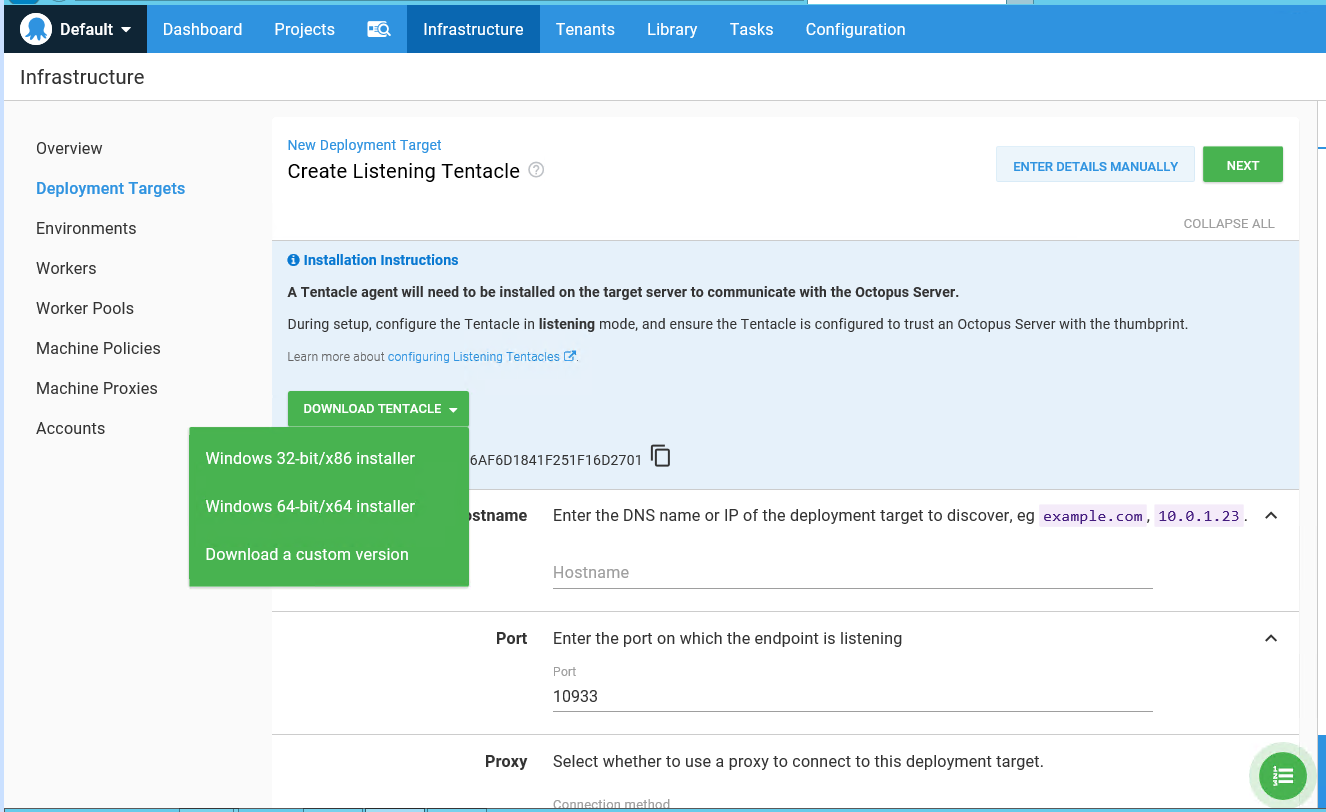
* Octopus Tentacle x64
* Octopus Tentacle 32-bit/x86
* Octopus Tentacle for Linux

If you need an older version of Tentacle that works with a previous long-term support release, visit the previous downloads page, locate the specific version of Octopus you're working with, and click your version to visit the downloads page for that version.

##### Fast Lane Tentacle:-

The version of Tentacle that works with the latest fast lane version of Octopus Deploy:

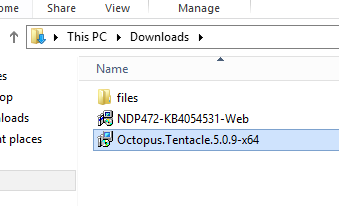
* Octopus Tentacle x64
* Octopus Tentacle 32-bit/x86
* Octopus Tentacle for Linux



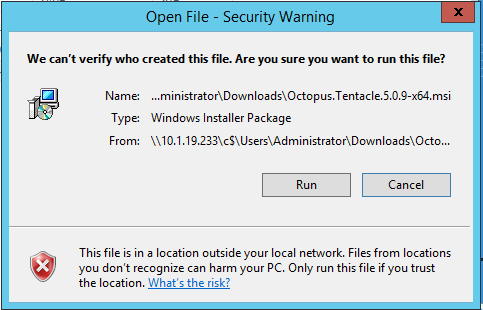
#### Configure a Listening Tentacle (recommended):-

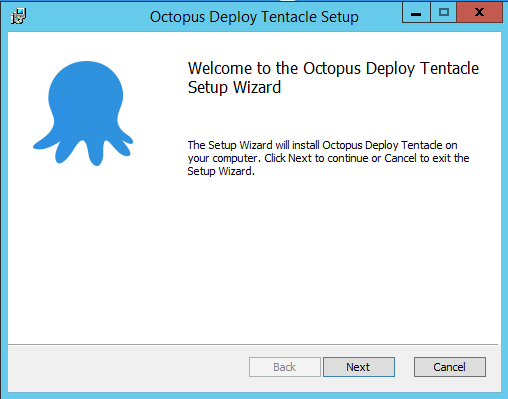
Before you can configure your **Windows servers as Tentacles**, you need to **install Tentacle Manager** on the machines that you plan to use as Tentacles.

**Tentacle Manager** is the **Windows application** that **configures your Tentacle**. Once installed, you can access it from your start menu/start screen. **Tentacle Manager** can configure Tentacles to use a [proxy](https://octopus.com/docs/infrastructure/deployment-targets/proxy-support), delete the Tentacle, and show diagnostic information about the Tentacle.



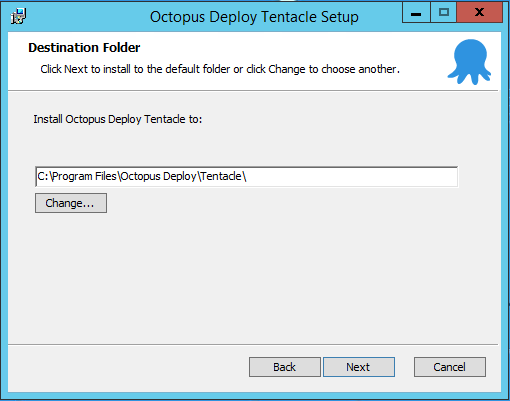
Start the **Tentacle installer**

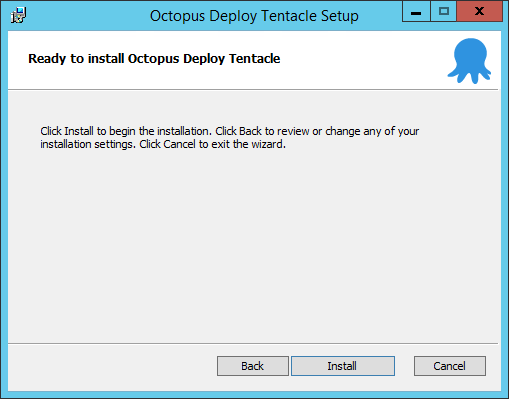




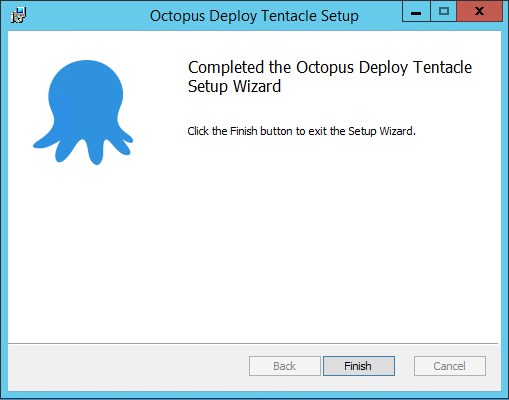
Click **Next**. **Accept** the license agreement, and follow the onscreen prompts.



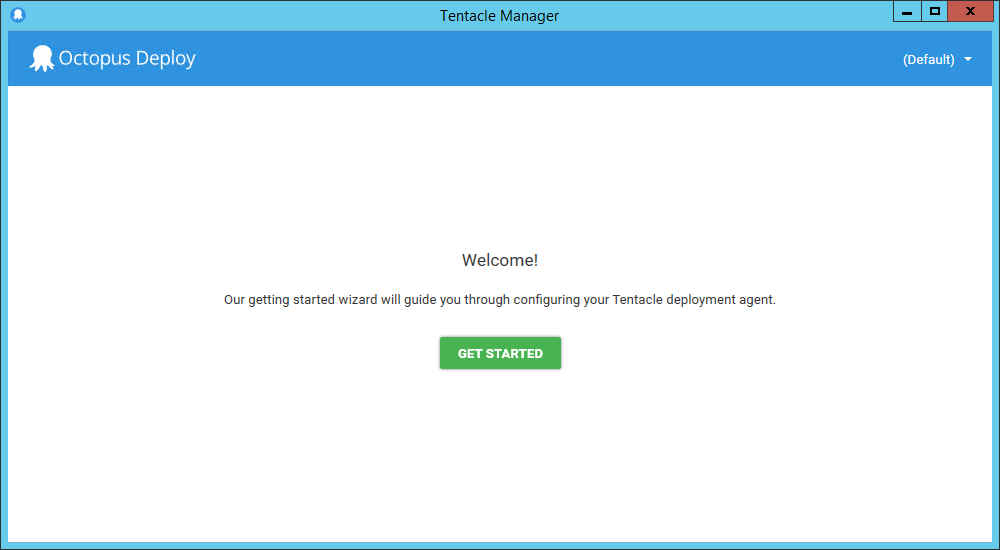




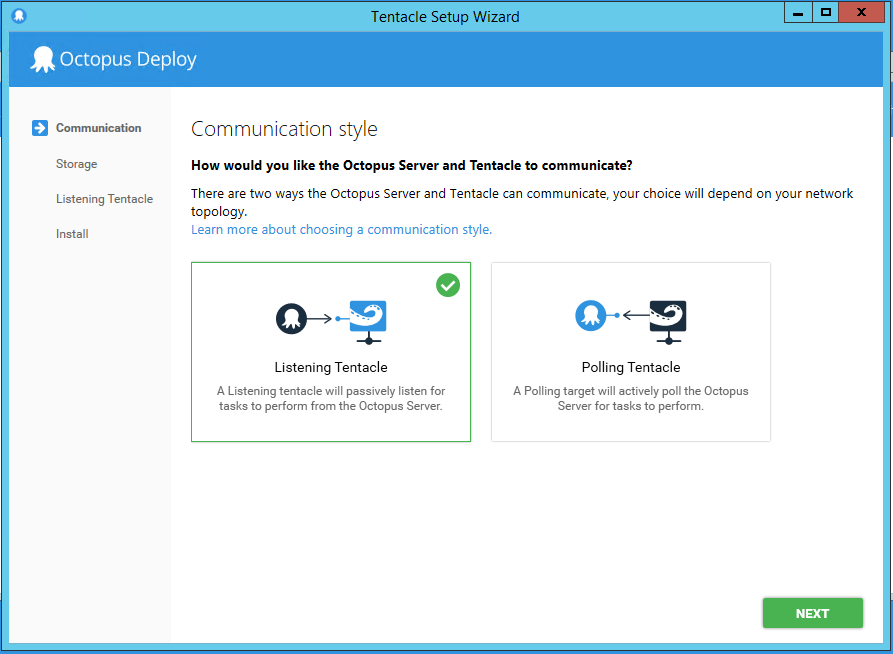
When the Octopus Deploy Tentacle Setup Wizard has completed, click **Finish** to exit the wizard.



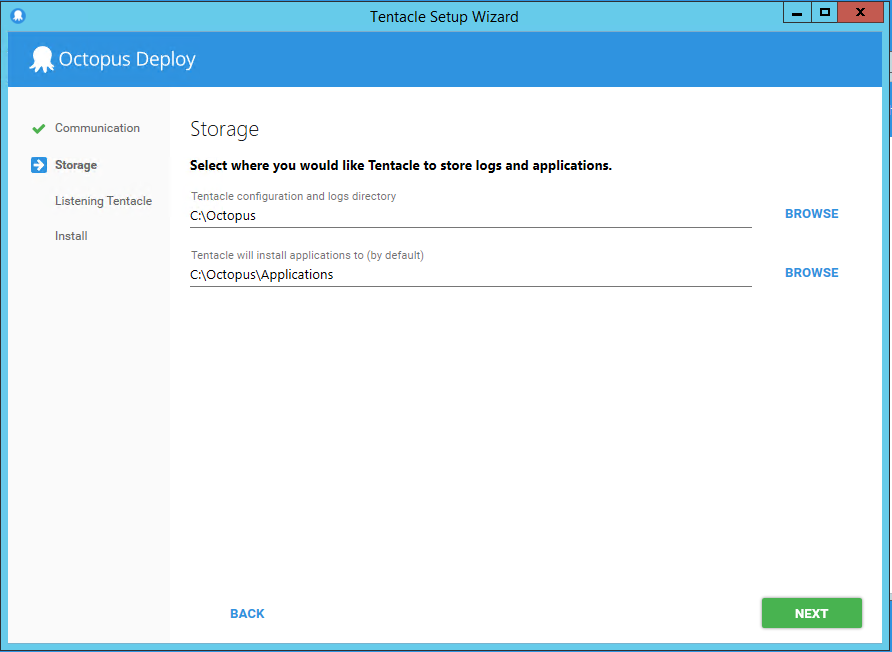
When the Tentacle Manager launches, click **GET STARTED**.



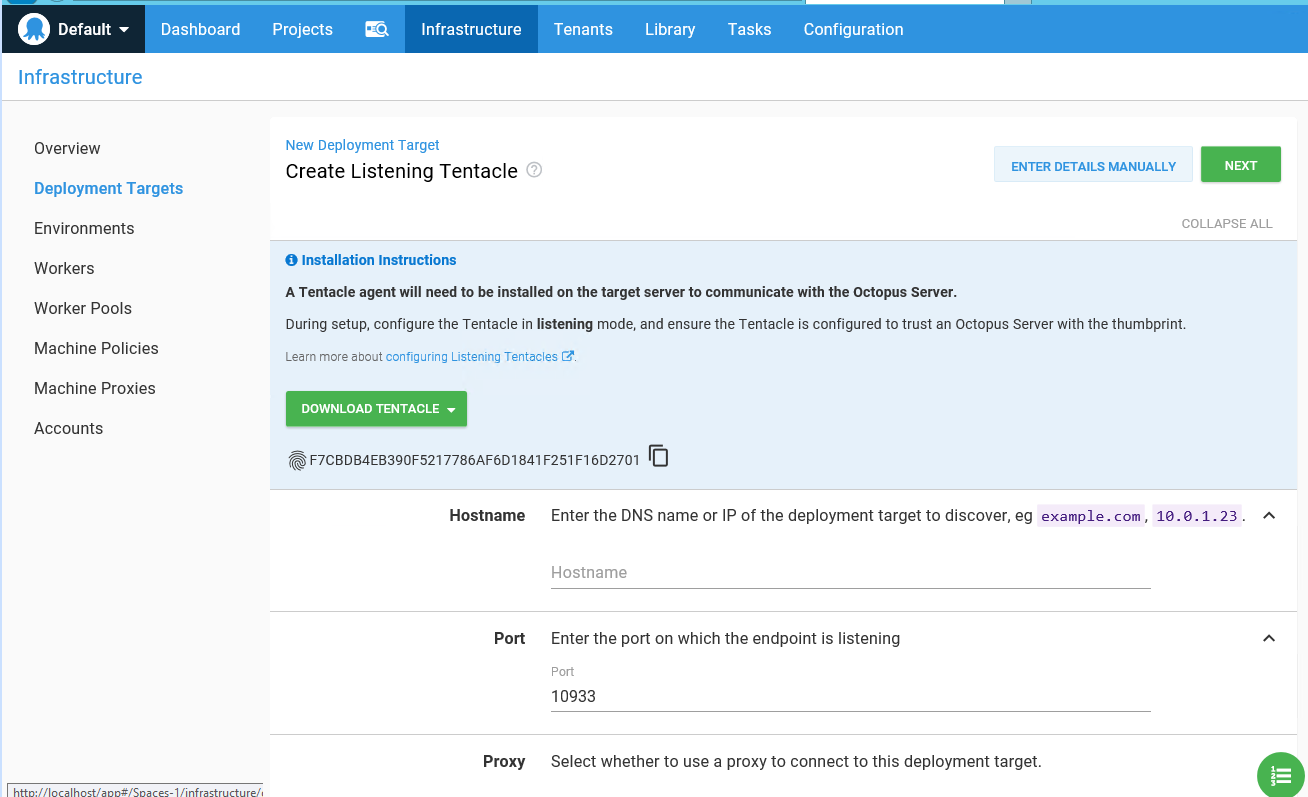
On the communication style screen, select **Listening Tentacle** and click **Next**.



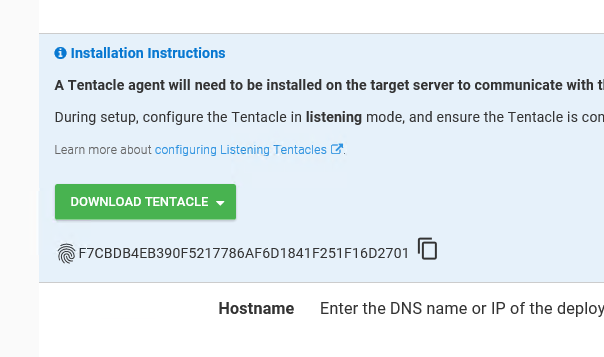
Click **Next**.

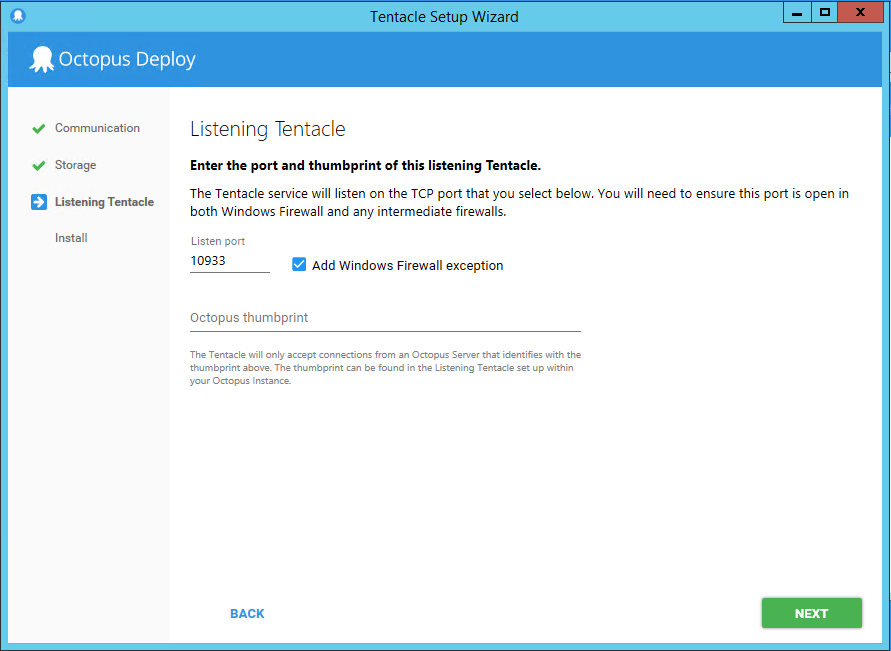


Click **Next**.

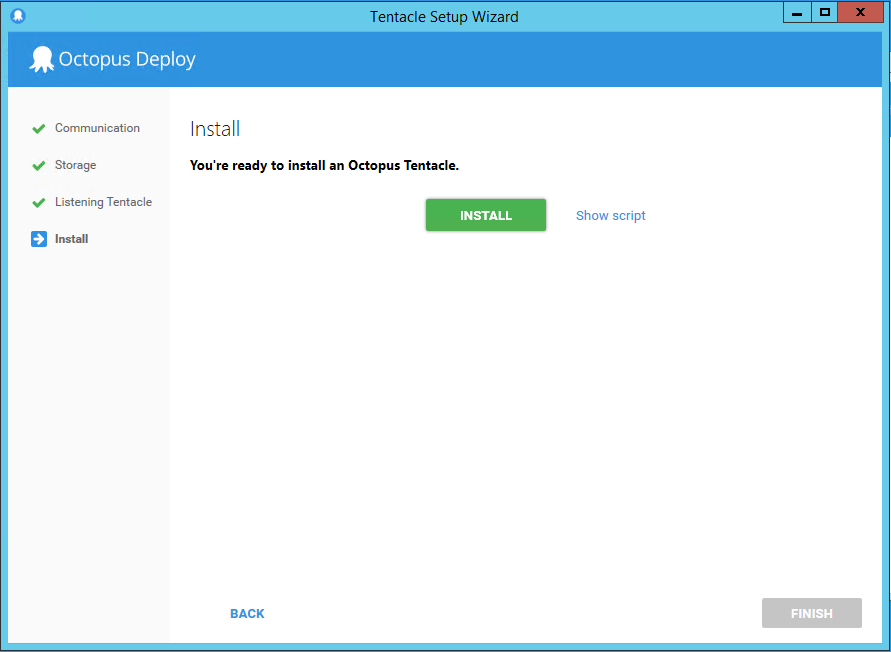


Back on the Tentacle server, accept the default listening port **10933** and paste the **Thumbprint** into the **Octopus Thumbprint** field and click **Next**.



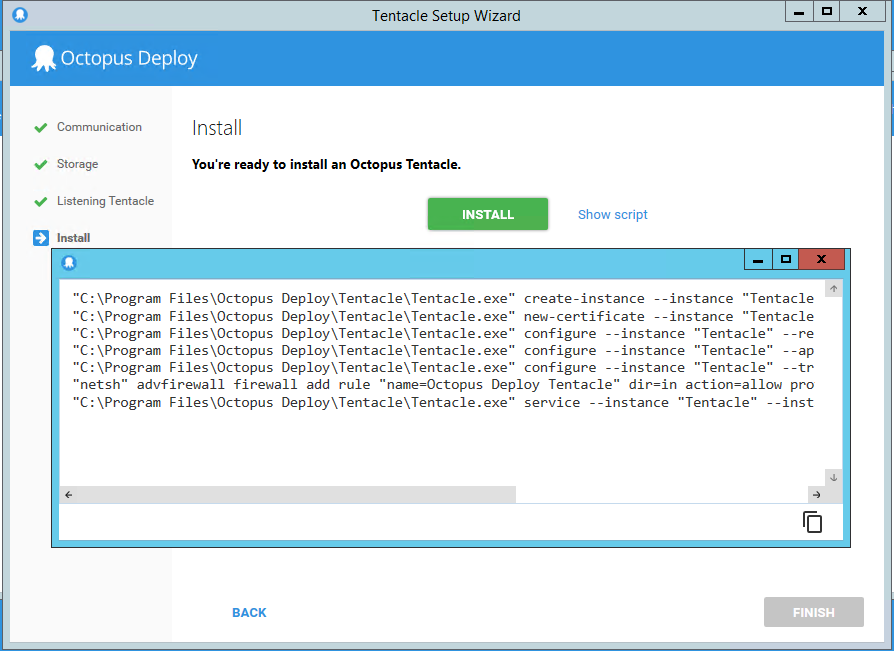


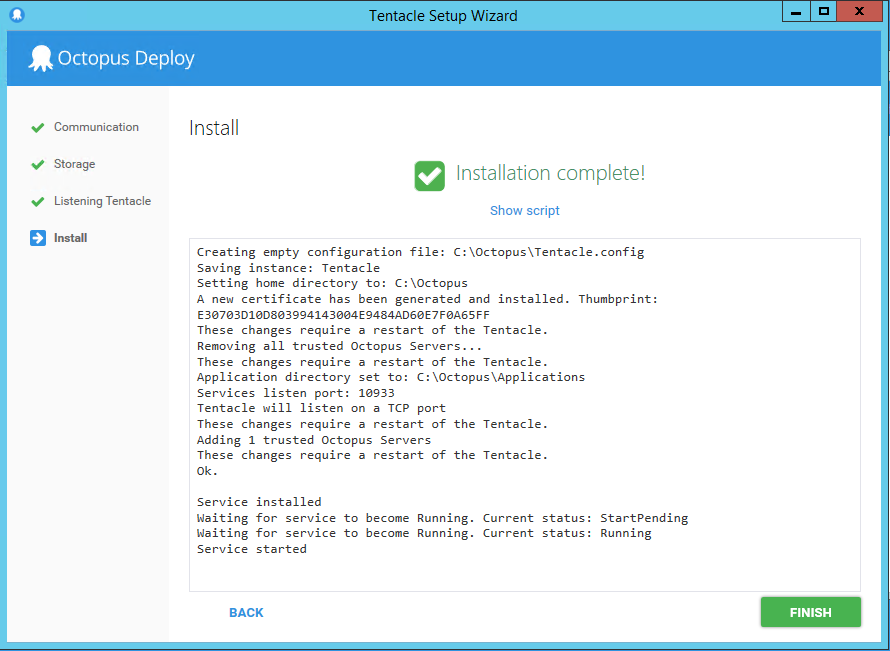
Click **INSTALL**, and after the installation has finished click **Finish**.

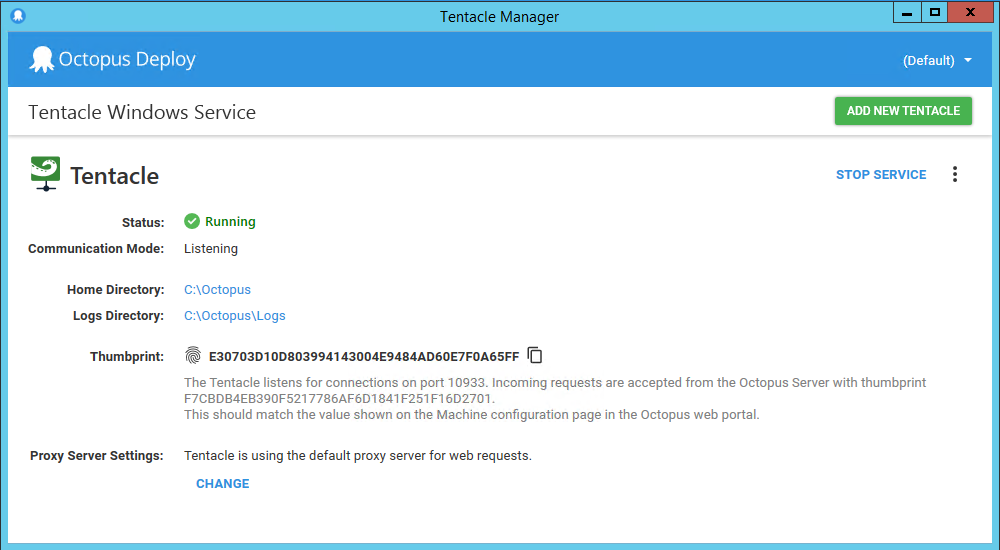


Once installed, Tentacles:

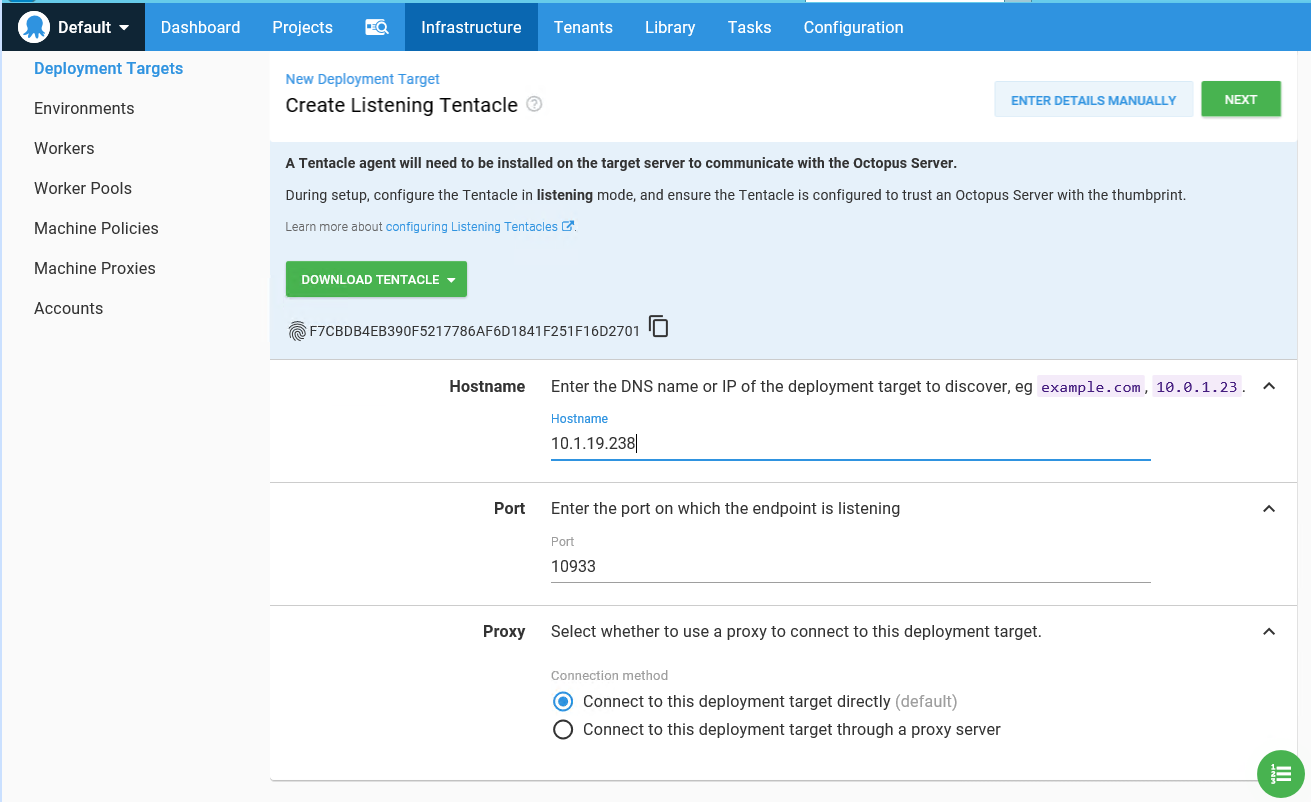
* Run as a Windows service called **OctopusDeploy Tentacle**.
* Wait for tasks from Octopus (deploy a package, run a script, etc).
* Report the progress and results back to the Octopus server.







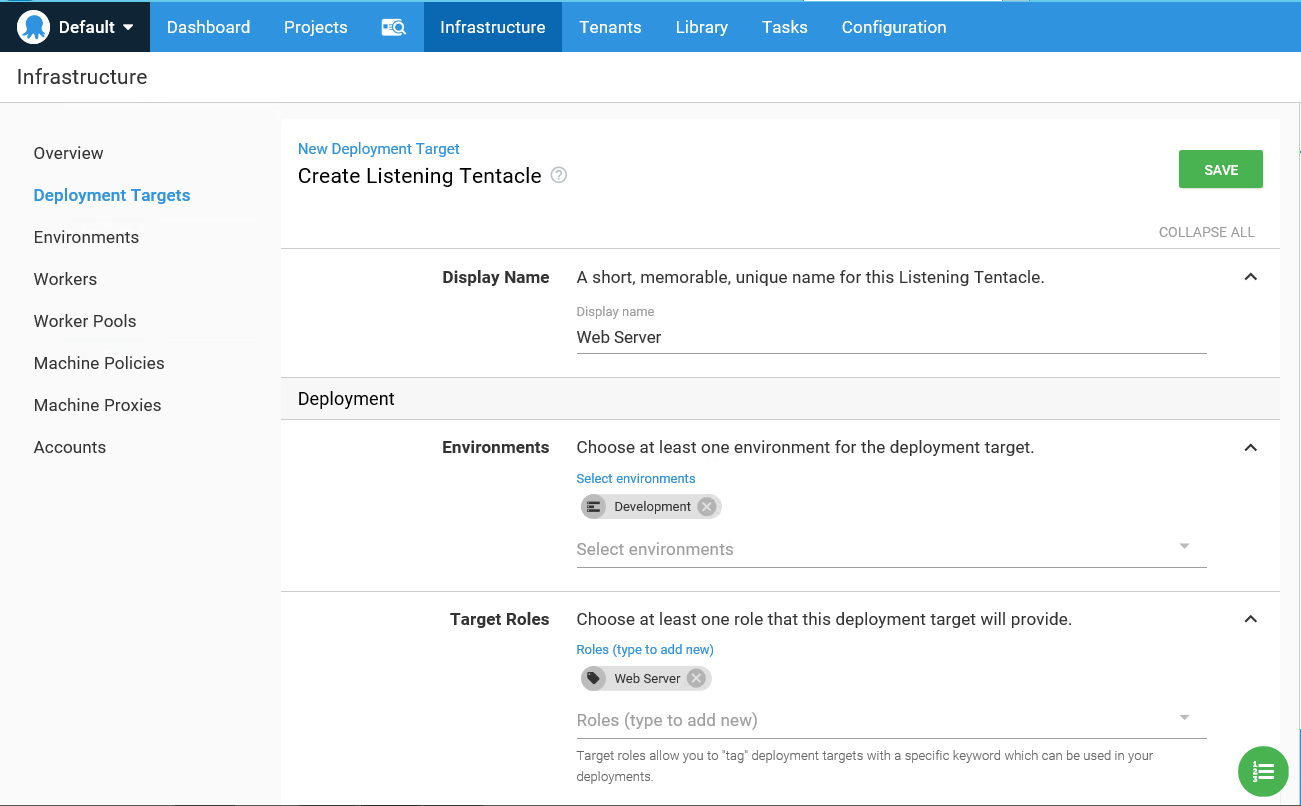
Back in the **Octopus Web Portal**, enter the hostname or IP address of the machine the Tentacle is installed on, i.e., example.com or 10.0.18.238, and click **NEXT**.

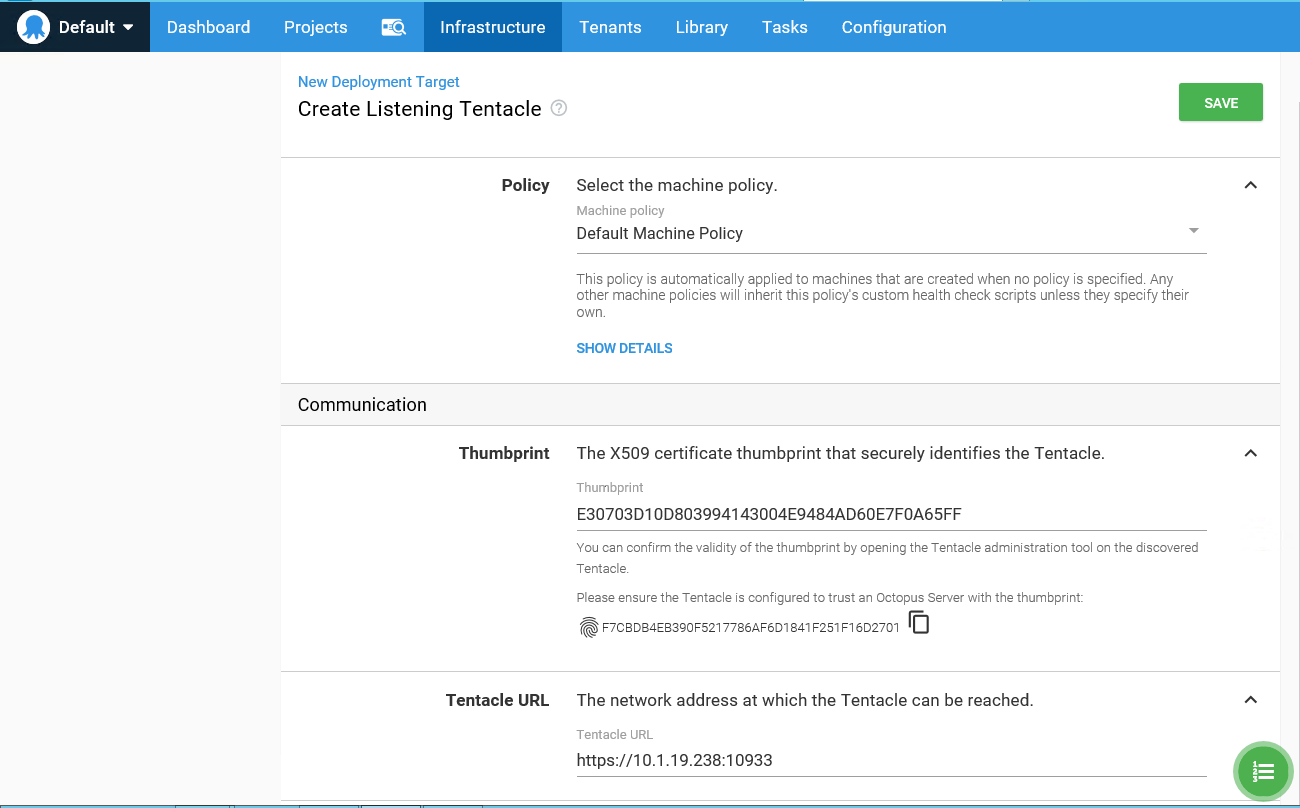


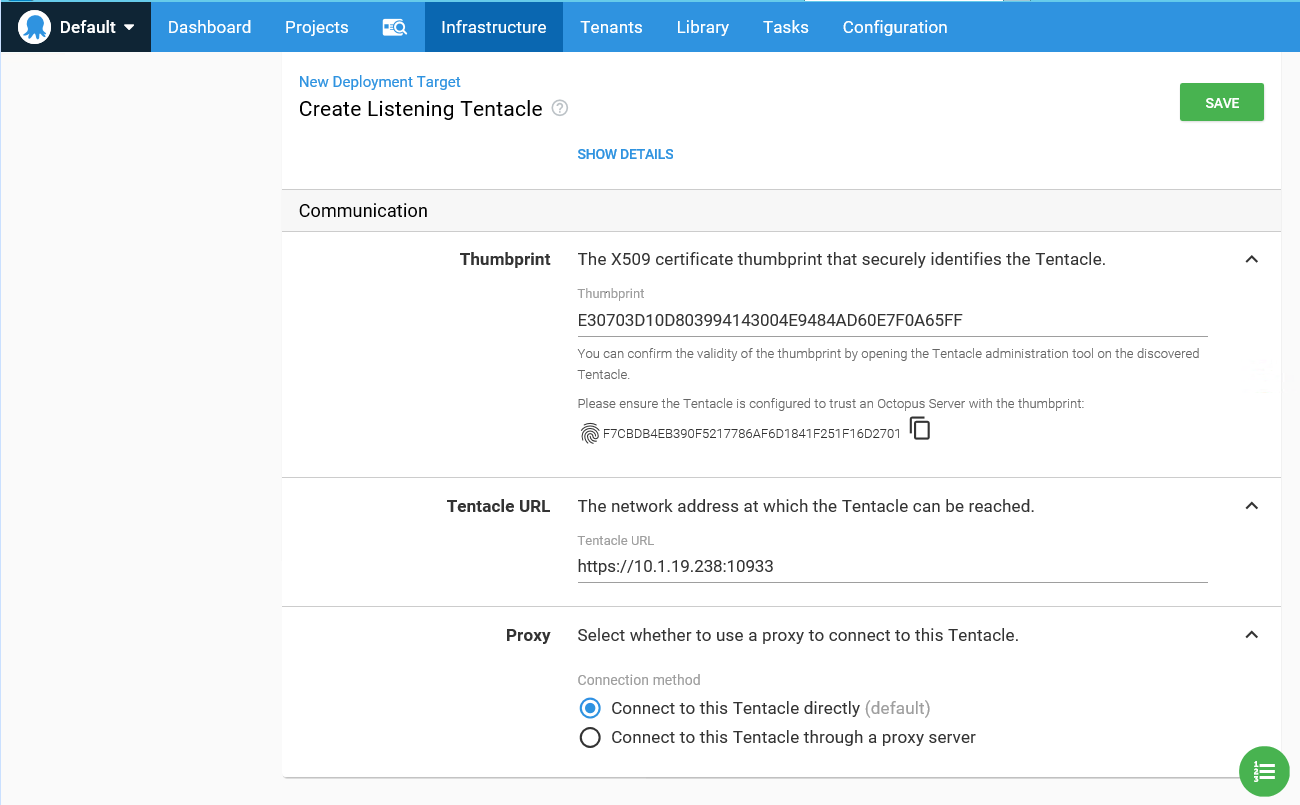
Add a display name for the deployment target (the server where you just installed the Listening Tentacle).

Select which [environments](https://octopus.com/docs/infrastructure/environments) the deployment target will be assigned to.

Choose or create at least one [target roles](https://octopus.com/docs/infrastructure/deployment-targets#target-roles) for the deployment target and click **Save**.

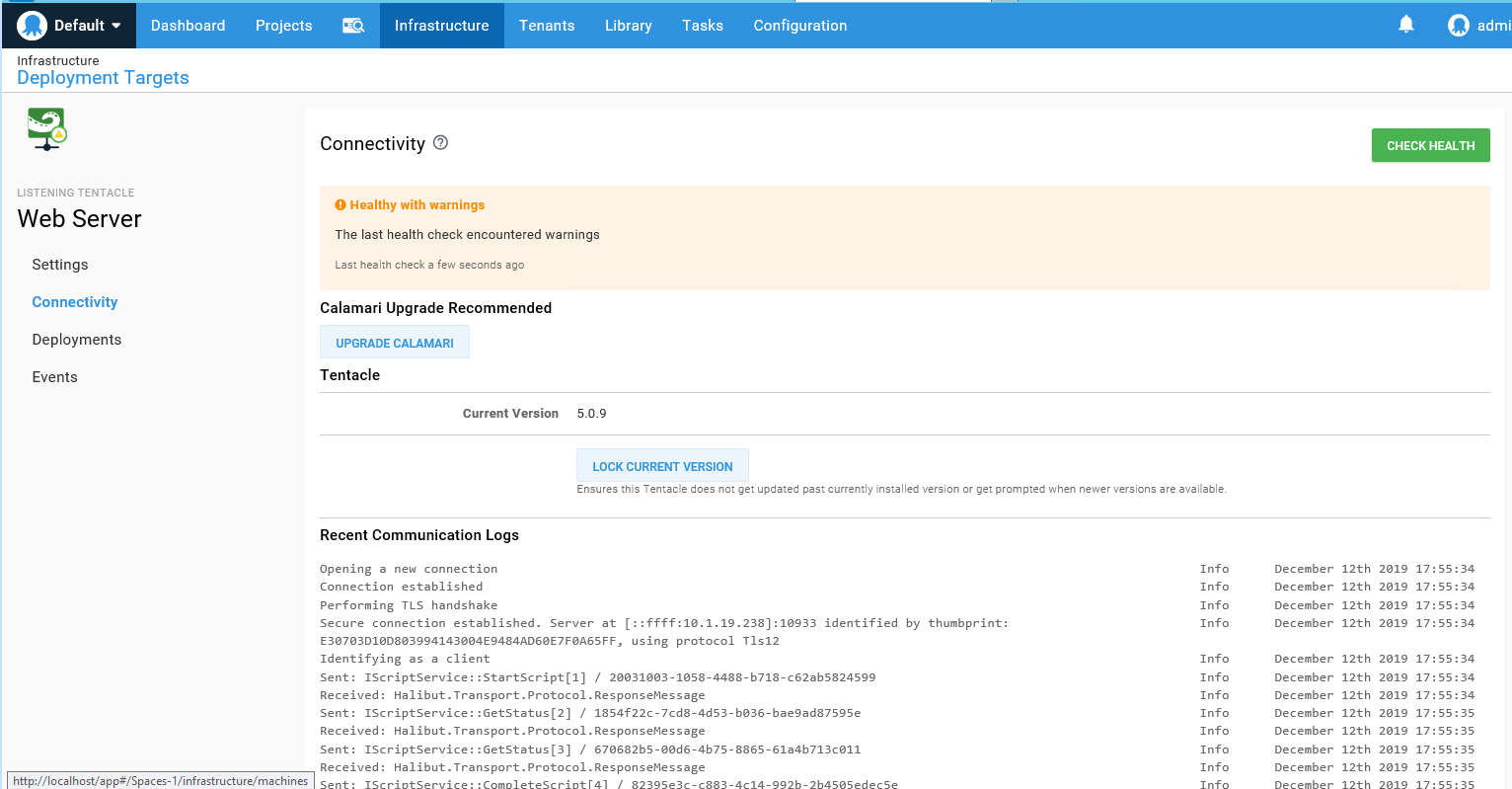


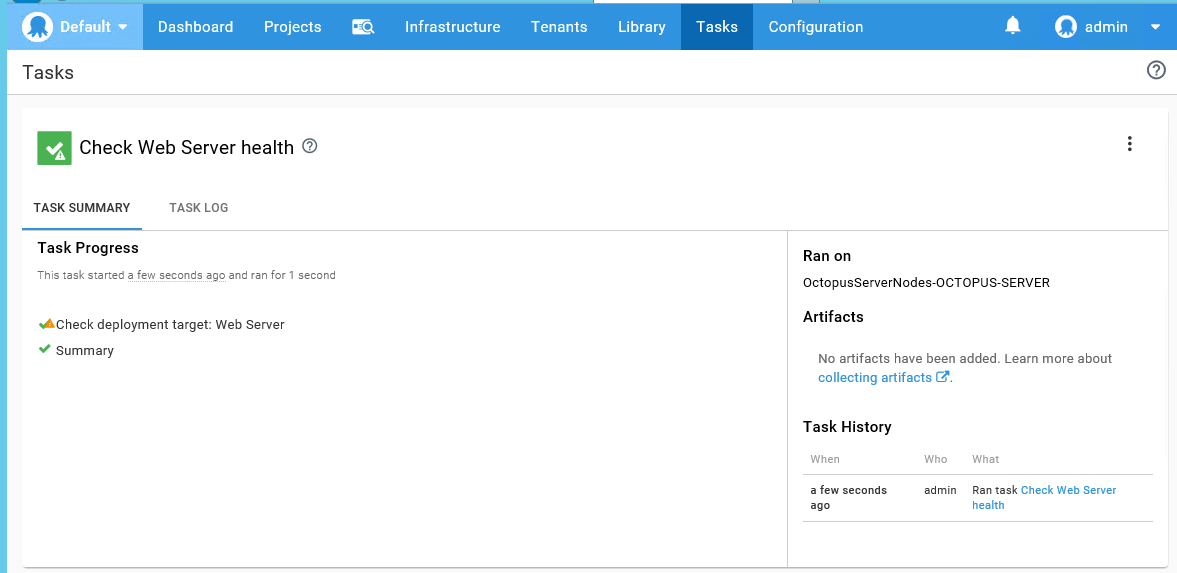




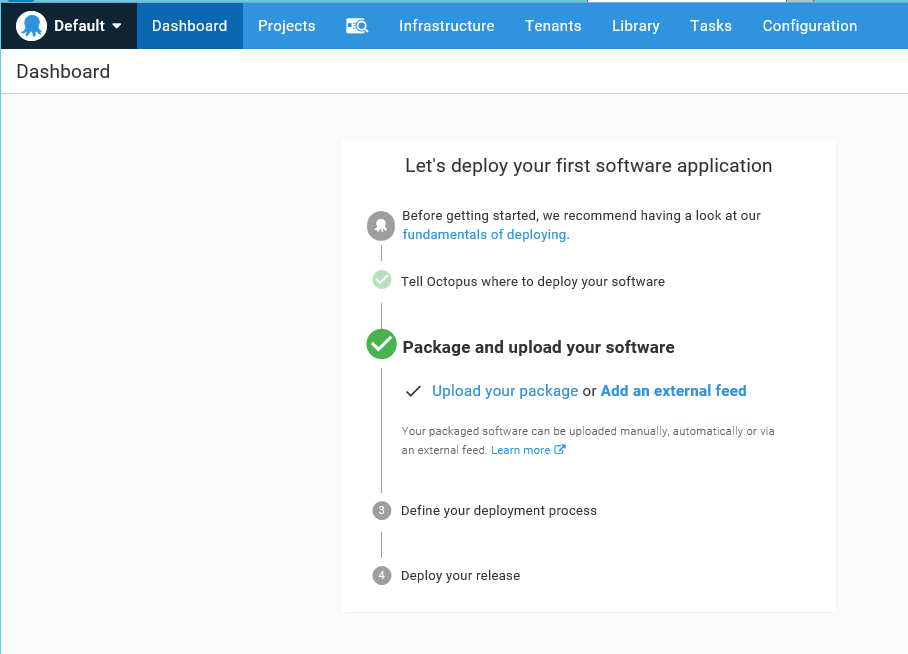
Your deployment target is configured, next you need to perform a [health check and update Calamari](https://octopus.com/docs/infrastructure/deployment-targets/machine-policies#health-check).

If the Tentacle isn't connecting, try the steps on the [troubleshooting page](https://octopus.com/docs/infrastructure/deployment-targets/windows-targets/troubleshooting-tentacles).





# Packaging Applications



Deploying software with Octopus often involves deploying packages. This section explains how to package your applications for deployment with Octopus.

Before you can deploy a package you need to:

1. Give your package a [package ID](https://octopus.com/docs/packaging-applications#package-id).
2. Choose and apply a [versioning scheme](https://octopus.com/docs/packaging-applications#version-numbers).
3. Create the package in a [supported format](https://octopus.com/docs/packaging-applications#supported-formats).
4. Host the package in a [package repository](https://octopus.com/docs/packaging-applications/package-repositories).

#### Package ID

Package IDs must conform to the following specifications:

* Package IDs must be unique within your Octopus Deploy instance.
* Package IDs consist of one or more segments separated by one of the following separator characters: - . \_.
* Segments contain only alphanumeric characters.

For instance, the package ID in this sample package is hello-world.

[hello-world.1.0.0.zip](https://octopus.com/images/docs/hello-world.1.0.0.zip)

Avoid using numbers in your package ID as it could result in the version number being incorrectly parsed.

#### Version Numbers

Octopus supports [Semantic Versioning](https://octopus.com/docs/packaging-applications/create-packages/versioning#semver), unless you are deploying artifacts to a [Maven repository](https://octopus.com/docs/packaging-applications/package-repositories/maven-feeds) in which case you will need to use [Maven Versions](https://octopus.com/docs/packaging-applications/create-packages/versioning#maven).

The version number needs to be applied to your package after the package ID and before the format. For instance. The version number in our sample package is **1.0.0**.

[hello-world.1.0.0.zip](https://octopus.com/images/docs/hello-world.1.0.0.zip)

Learn more about [versioning schemes](https://octopus.com/docs/packaging-applications/create-packages/versioning).

Package Dependencies and Structure

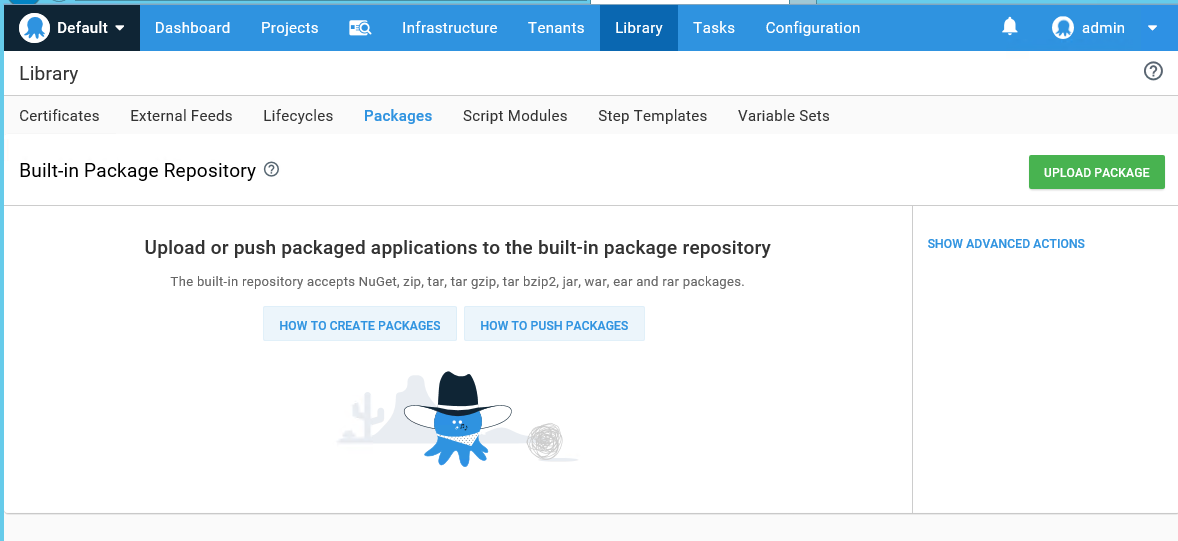
When you package your applications, you need to include all the binaries that are required to run the application, and structure the package the way you want it to appear after it has been extracted.

#### Supported Formats

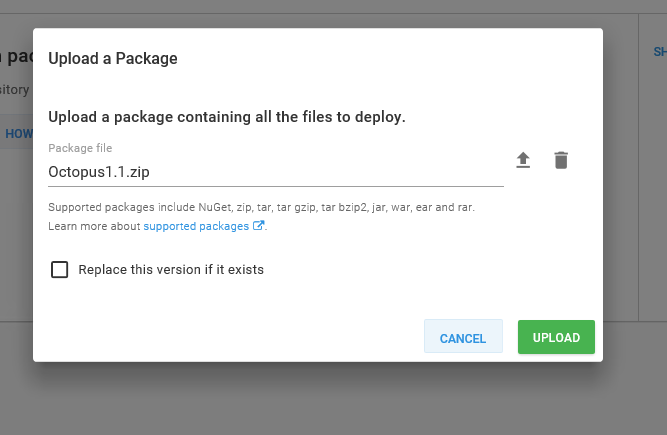
It is important that your packages have the correct **file extension** because Octopus uses the **file extension** to determine the correct extraction algorithm to use with your packages.

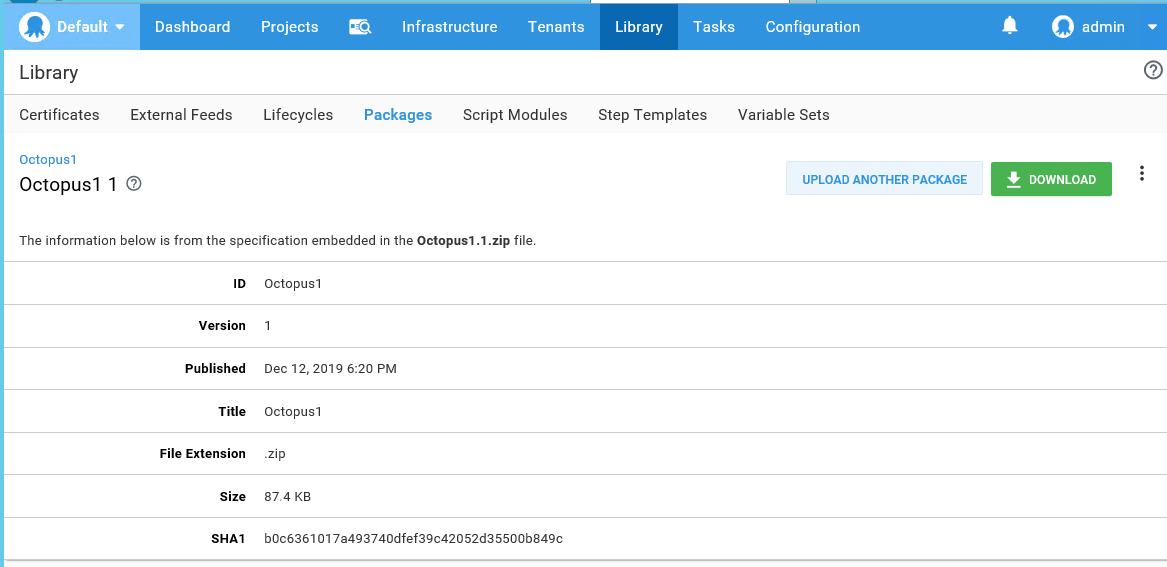
| **Package Type** | **File Extensions** | **Notes** |
| --- | --- | --- |
| NuGet | .nupkg | Compatible with any NuGet repository (including the [Built-In repository](https://octopus.com/docs/packaging-applications/package-repositories/built-in-repository)). Currently only NuGet packages will have extra metadata like release notes and description extracted from the package metadata. Learn about NuGet on the [official NuGet website](http://docs.nuget.org/docs/start-here/overview). |
| Zip | .zip | Standard zip file as created through most common zip programs. Compatible with the [built-In repository](https://octopus.com/docs/packaging-applications/package-repositories/built-in-repository) only. |
| JAR WAR EAR RAR | .jar, .war, .ear, .rar | Compatible with the [built-In repository](https://octopus.com/docs/packaging-applications/package-repositories/built-in-repository#pushing-packages-to-the-built-in-repository) and [Maven Feeds](https://octopus.com/docs/packaging-applications/package-repositories/maven-feeds) from **Octopus 4.1**. |
| Tar | .tar | Compatible with the [Built-In repository](https://octopus.com/docs/packaging-applications/package-repositories/built-in-repository) only. |
| Tar + Gzip | .tgz, .tar.gz, .tar.Z | Compatible with the [built-In repository](https://octopus.com/docs/packaging-applications/package-repositories/built-in-repository) only. |
| Tar + Bzip2 | .tar.bz, .tar.bz2, .tbz\* | Compatible with the [built-In repository](https://octopus.com/docs/packaging-applications/package-repositories/built-in-repository) only. |
| Docker Image |  | [Docker Registries](https://octopus.com/docs/packaging-applications/package-repositories/docker-registries). Learn about [Docker](https://octopus.com/docs/deployment-examples/docker-containers) and Octopus Deploy. |

#### Package Repositories



Click **Upload a Package** and select the package to upload. Click **UPLOAD.**

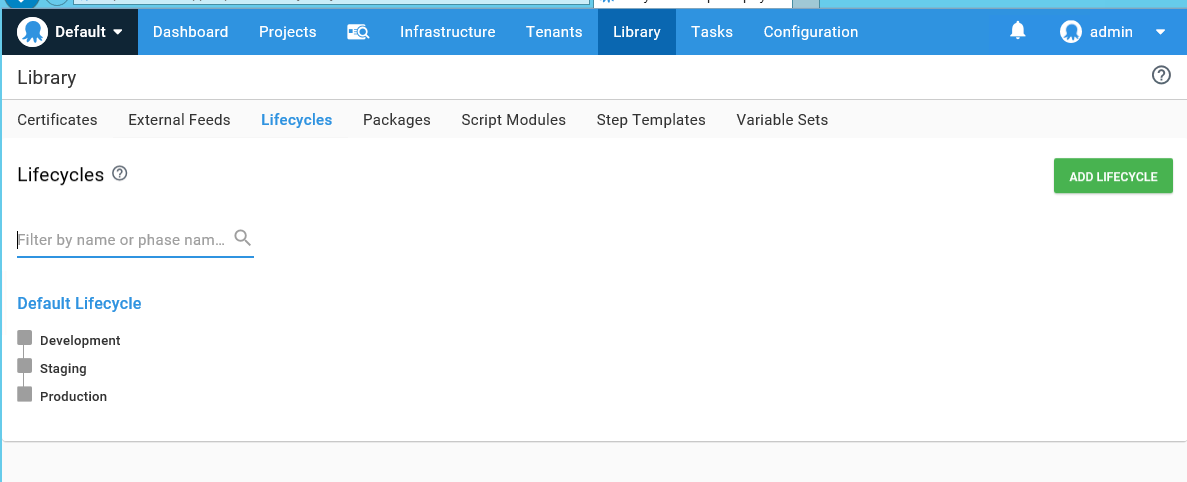




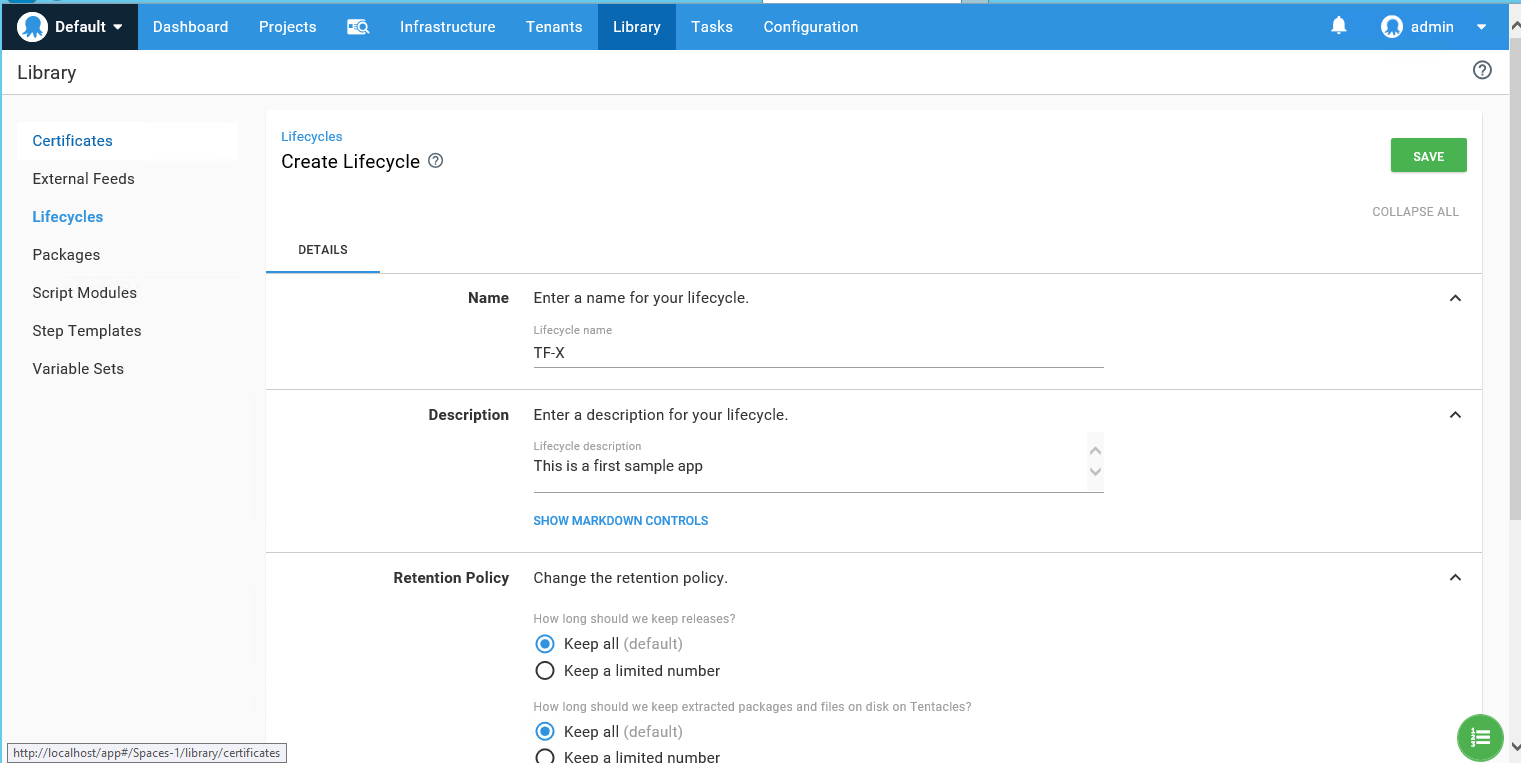
#### Lifecycles

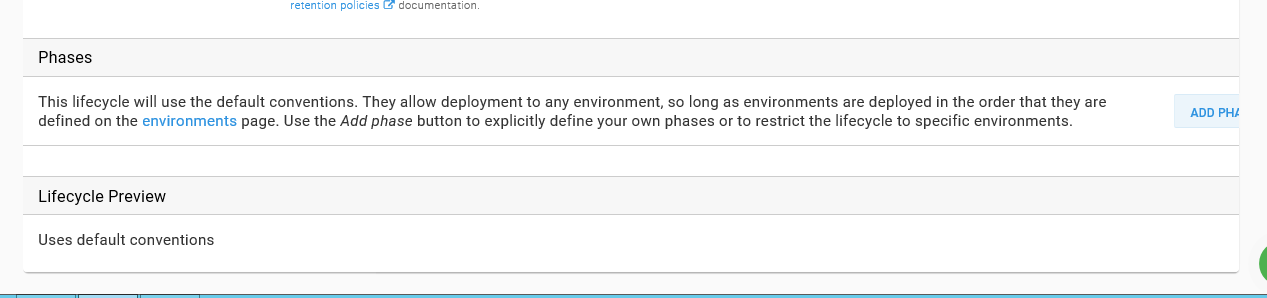
Lifecycles give you control over the way releases of your software are promoted between your environments. Lifecycles enable a number of advanced deployment workflow features:

* **Control the order of promotion**: for example, to prevent a release being deployed to *production* if it hasn't been deployed to *staging*.
* **Automate deployment to specific environments**: for example, automatically deploy to *test* as soon as a release is created.
* **Retention policies**: specify the number of releases to keep depending on how far they have progressed through the lifecycle.



Click on **ADD LIFECYCLE** and give a name, description and come down to ADD Phases

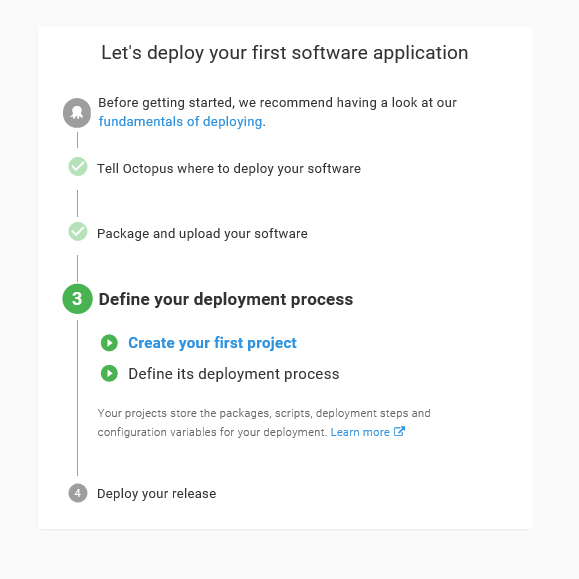




# Deployment Process

Now that you have access to an [Octopus Server](https://octopus.com/docs/getting-started#octopus-deploy-server), your [infrastructure is configured](https://octopus.com/docs/infrastructure), and your [applications packaged](https://octopus.com/docs/packaging-applications), you're ready to start deploying your software.

The deployment process is the steps the Octopus Deploy server orchestrates to deploy your software.

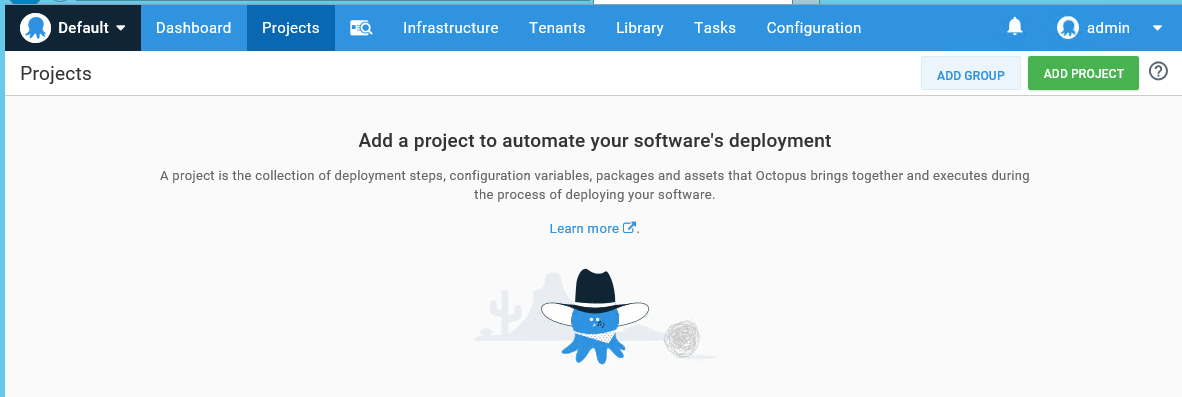


A Simple Hello World Deployment Process

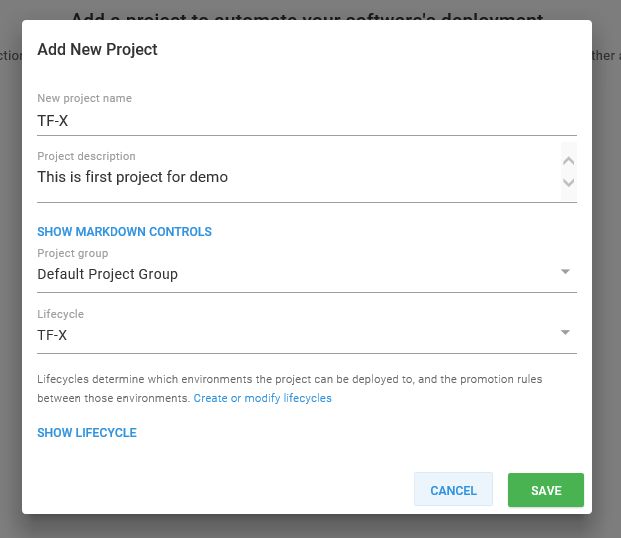
For this example process, we'll use Octopus Deploy to create a simple process with only one step that runs a script on a deployment target. The process uses a **Test** environment with one **Deployment Target** that has the target role **server**.

From the Octopus Web portal, select **Projects** from the main navigation, and click **ADD PROJECT**.

## Create your first project:-

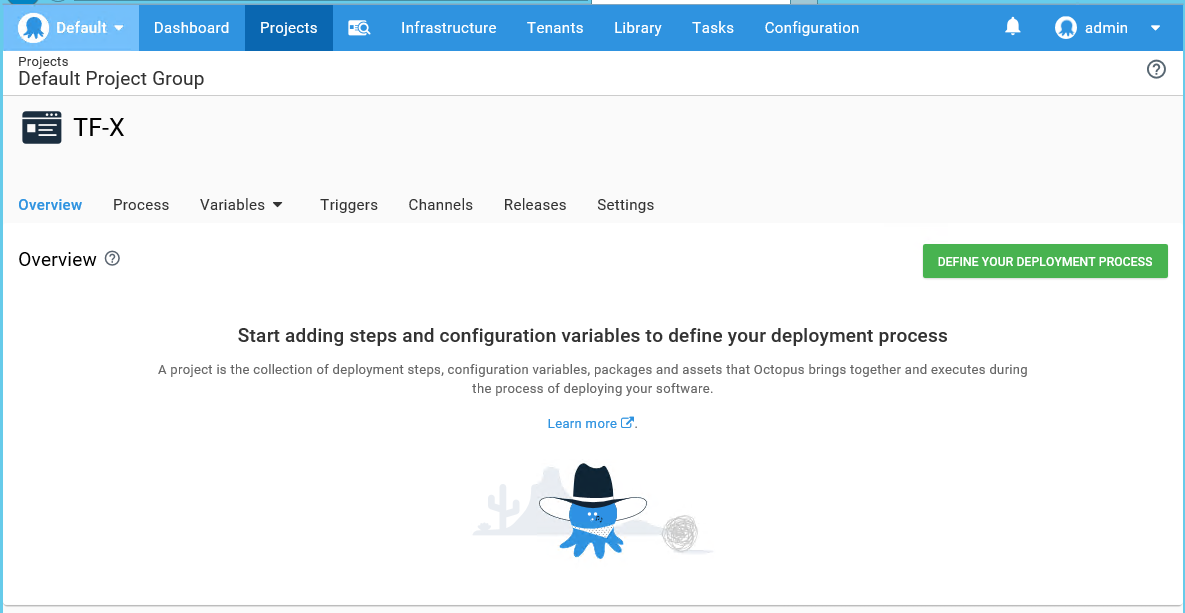


Name the project, for instance, Hello World, and click **Save**.

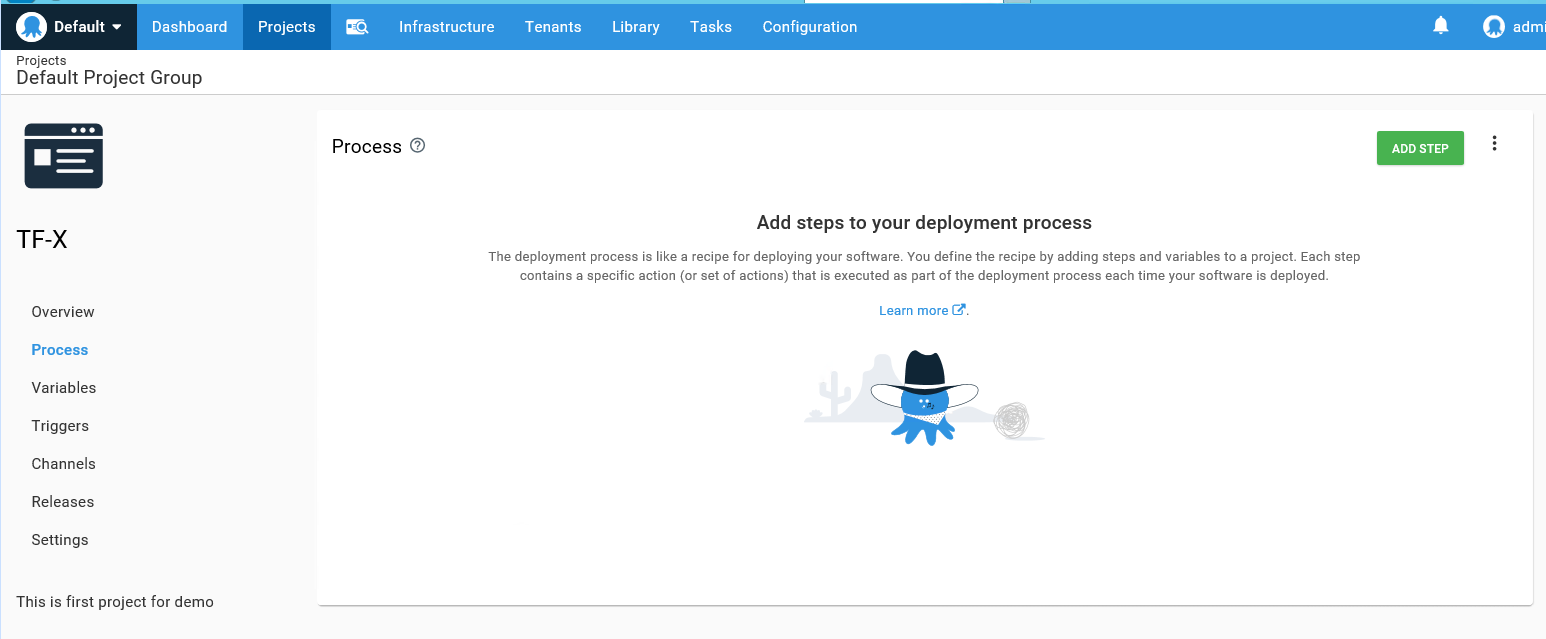


From your new project's overview page, click **DEFINE YOUR DEPLOYMENT PROCESS**.

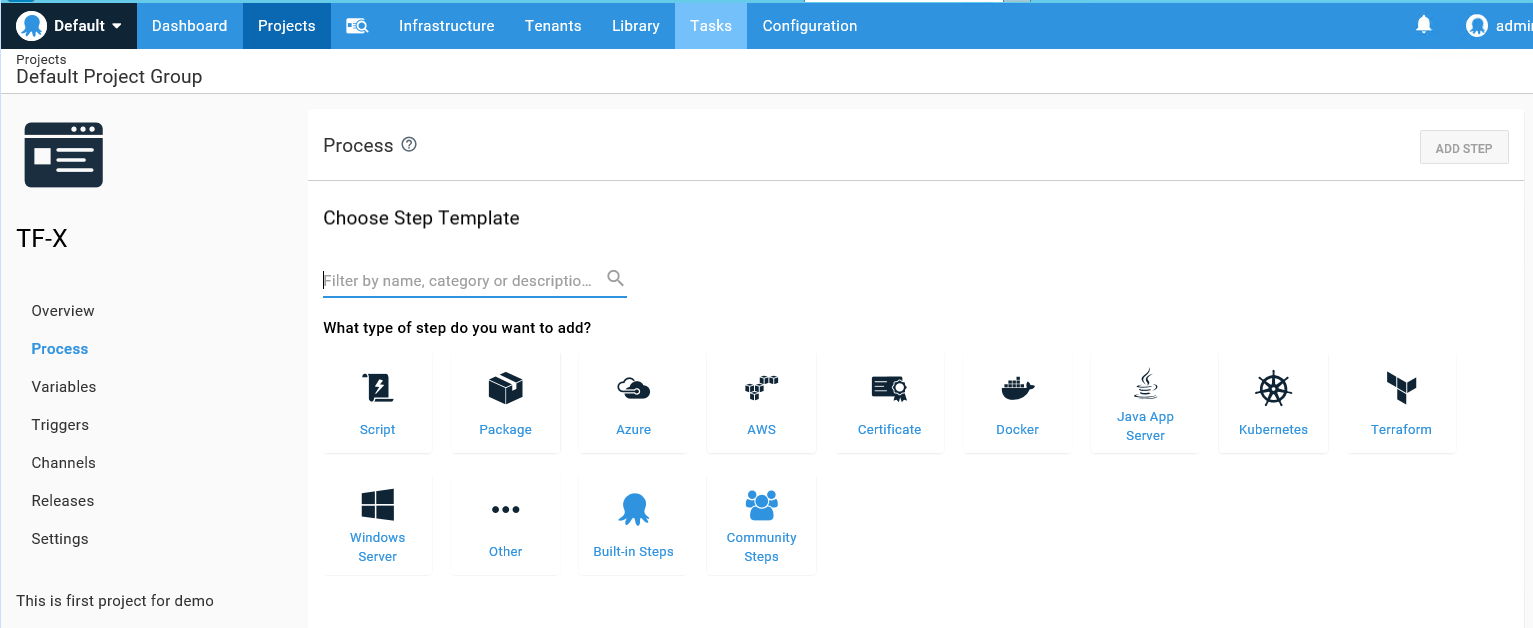
## ****Define Your Deployment Process****



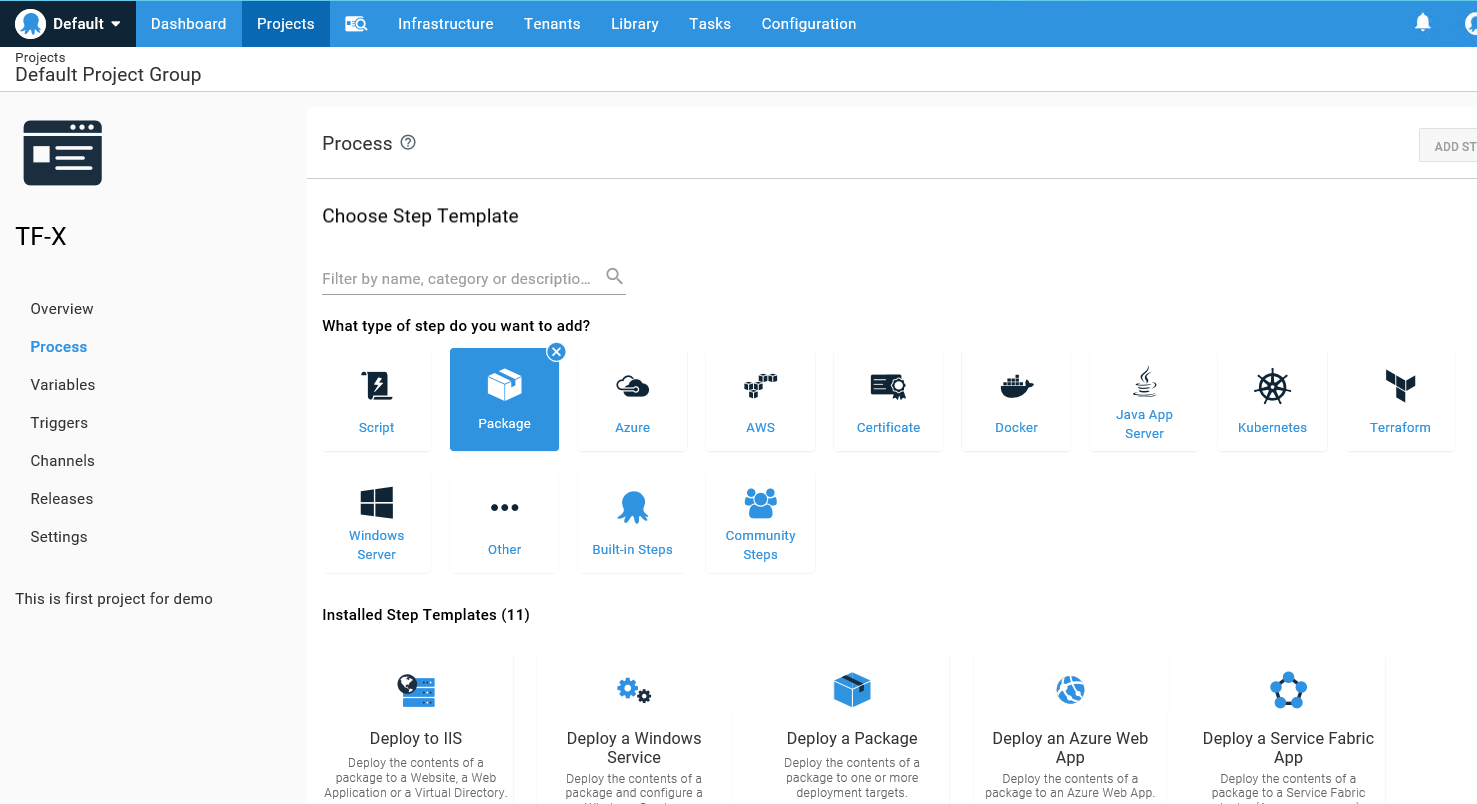
Click **ADD STEP**, and then select the **Run a Package** step.



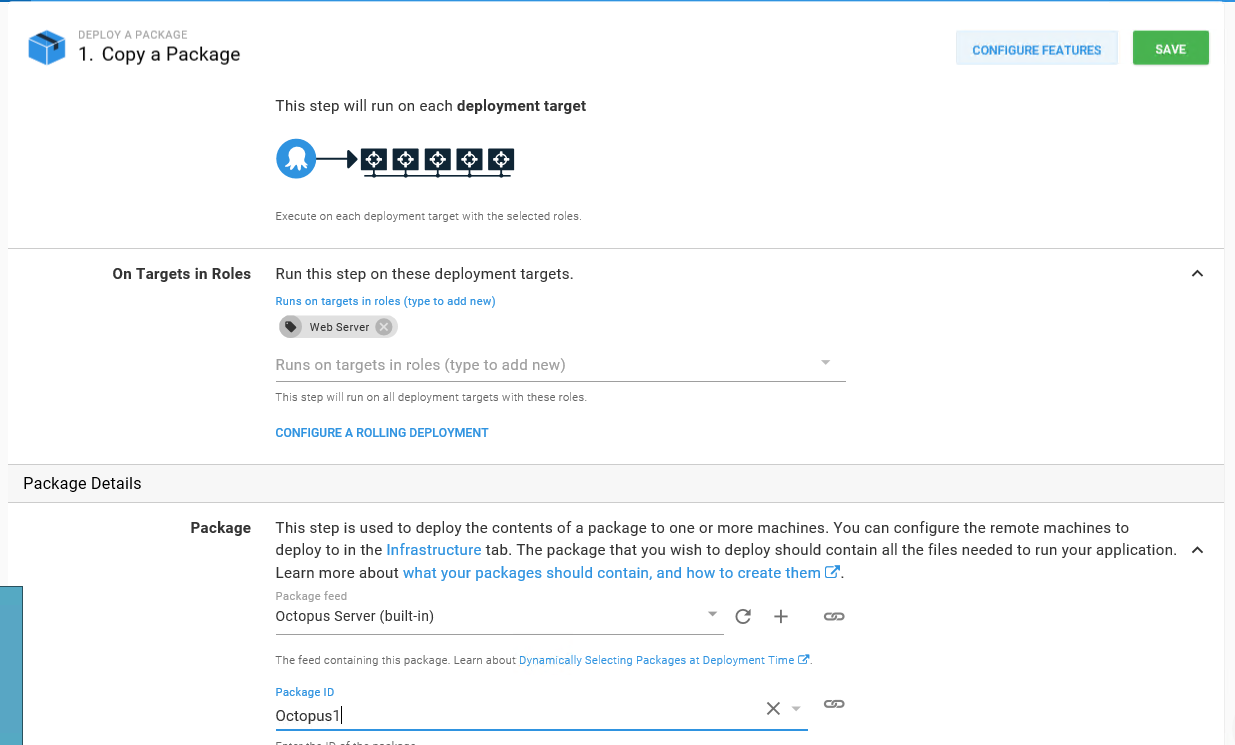
##### List of Steps



### Select Package

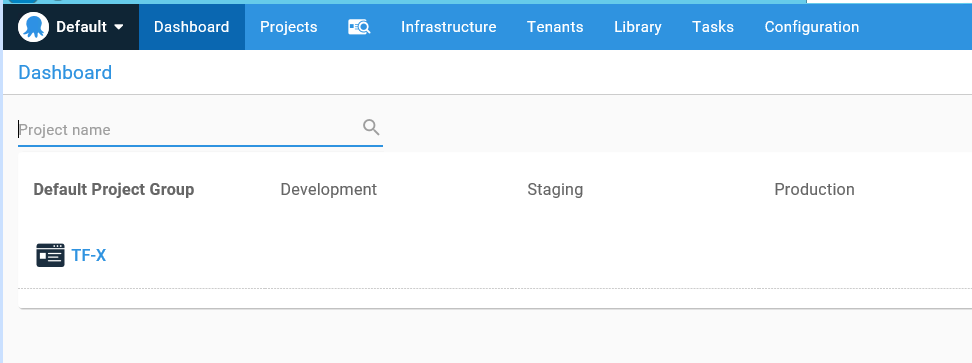


### Select Deploy to Copy a Package



For the execution plan, leave the selection at the default Deployment targets and select the target role server.

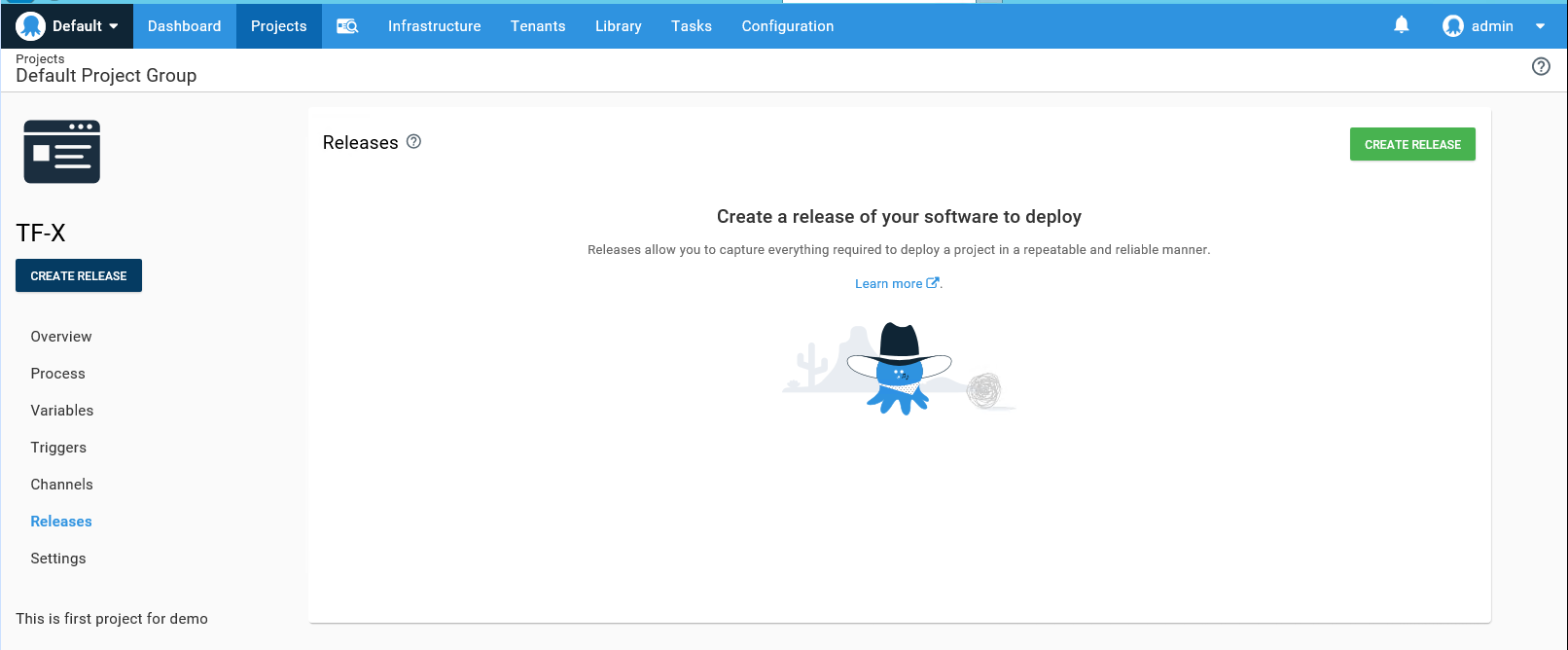
Select Package Feed and Package ID and click **Save**:



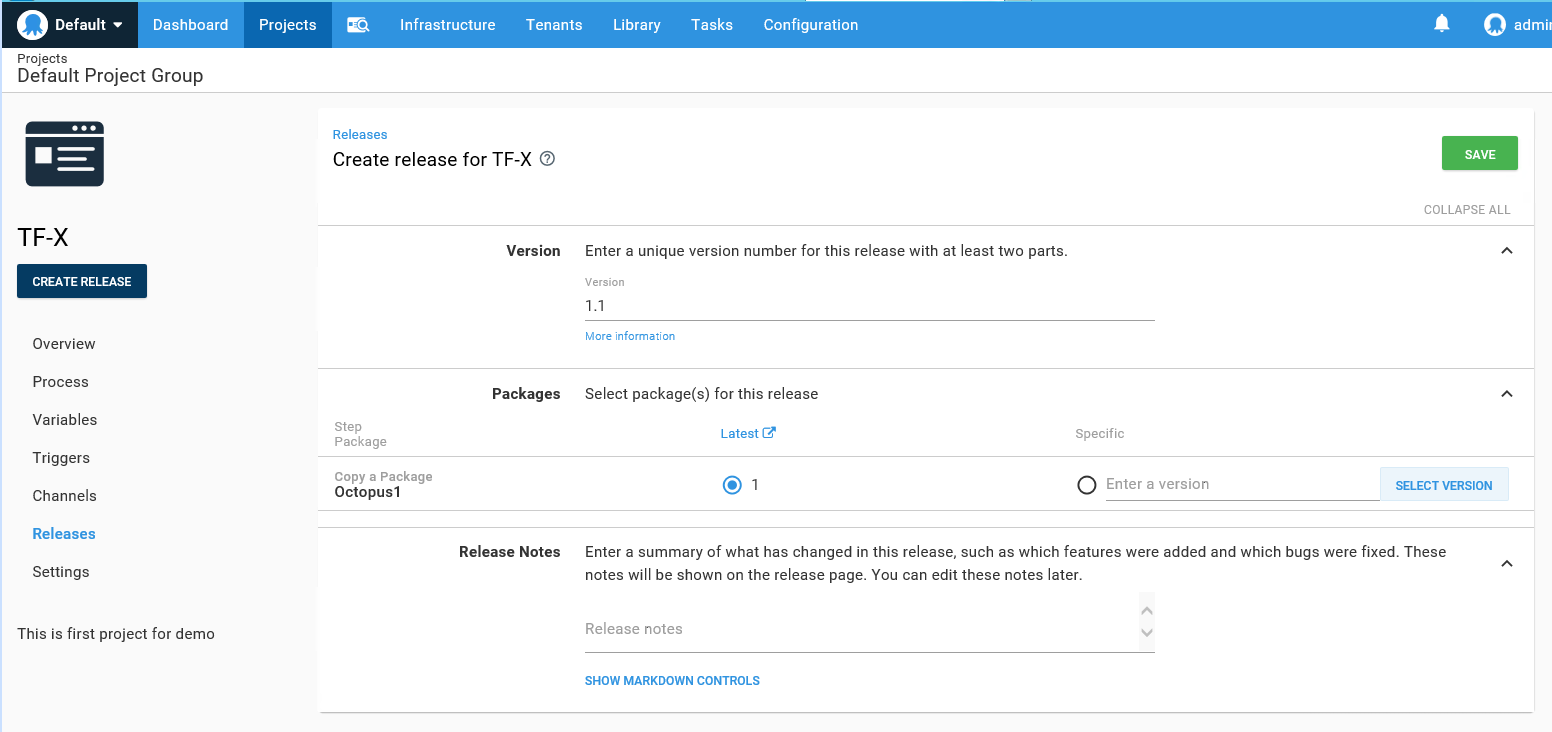
# Deploy your release

From the project's overview page, click **CREATE RELEASE**

## ****Create Release****

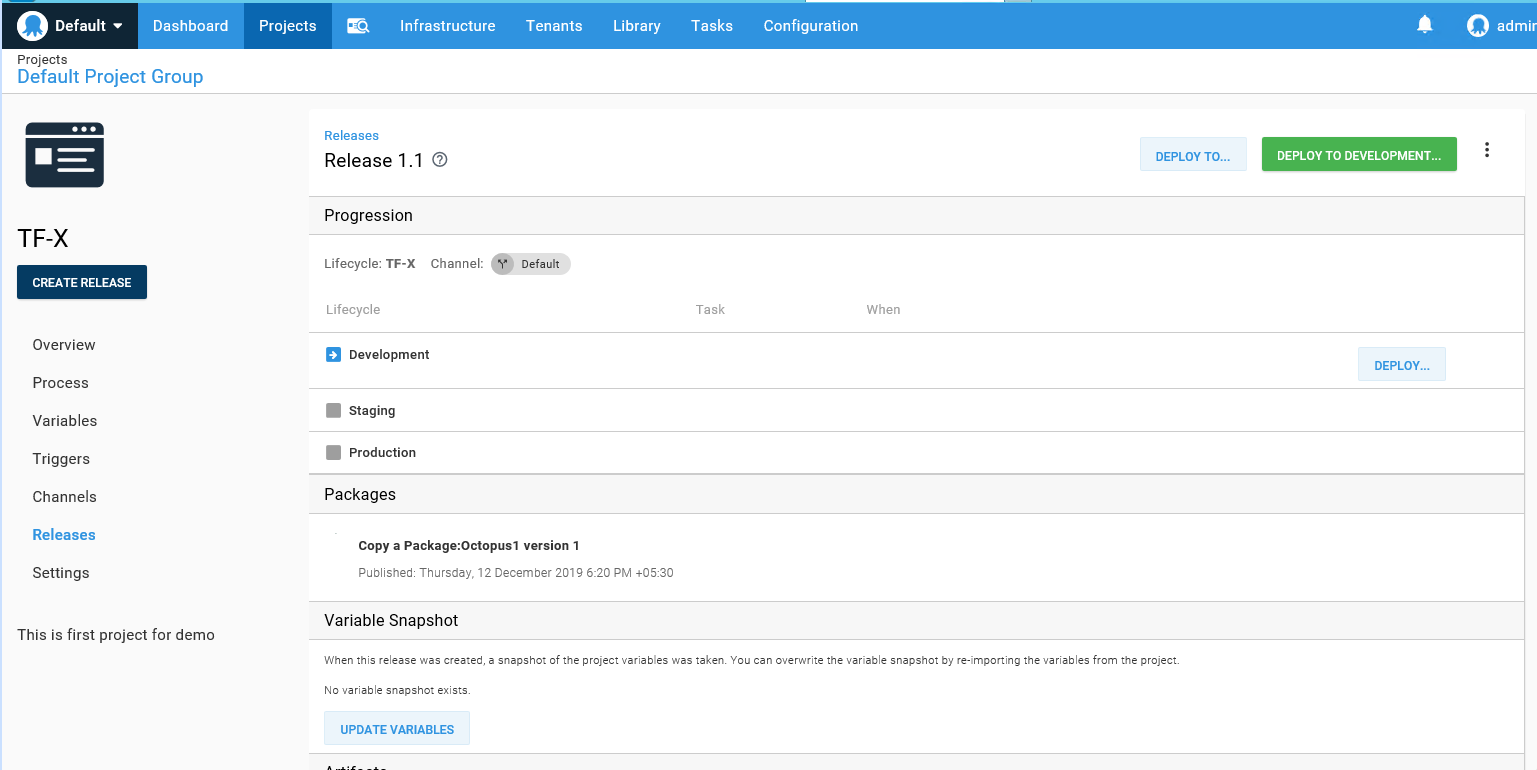


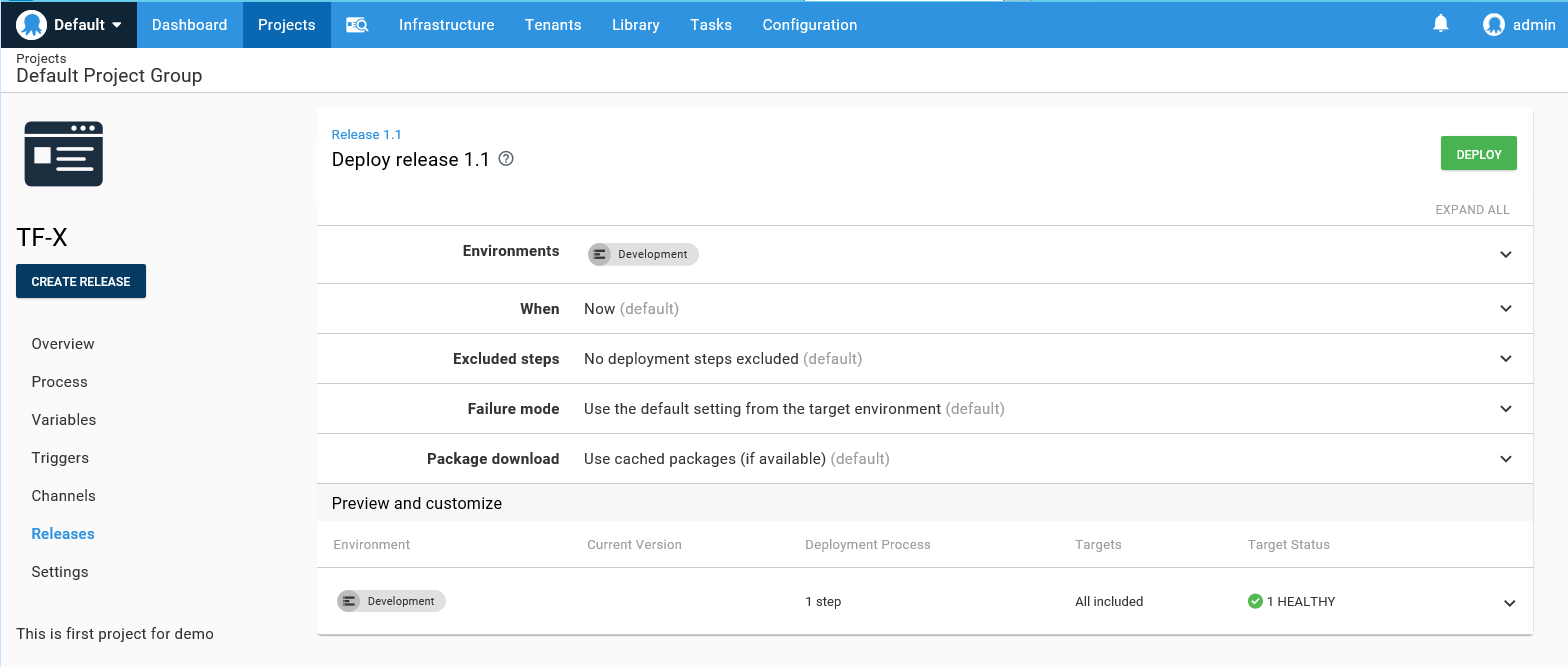
Enter the version number for the release, select packages, release notes and then click **Save**.



## Deploy a Release

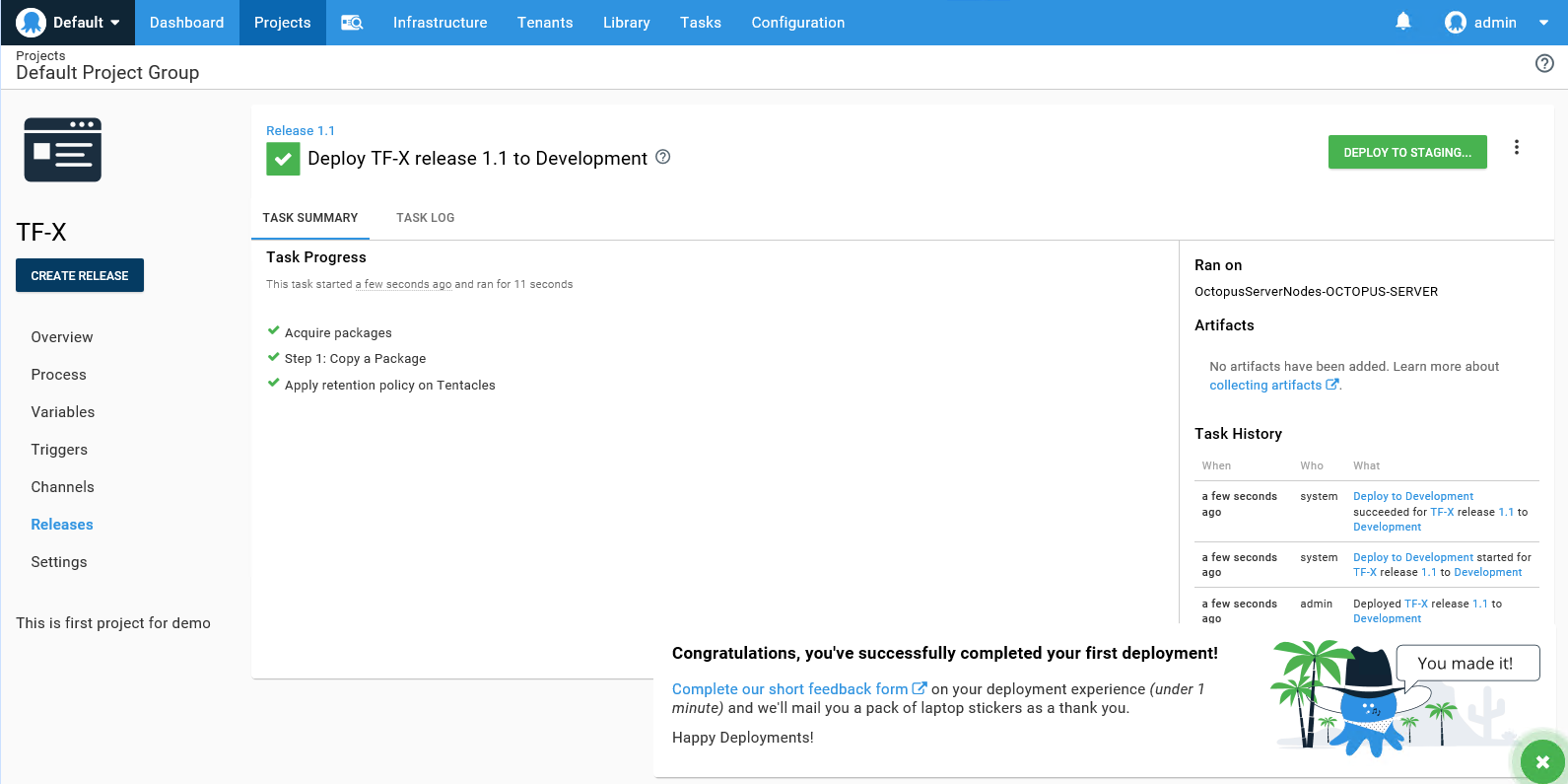
Click **DEPLOY TO DEVELOPMENT**, then click **DEPLOY**.

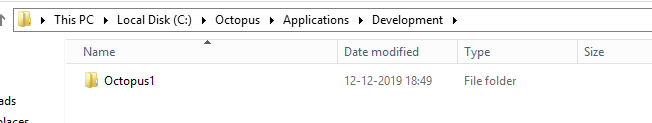


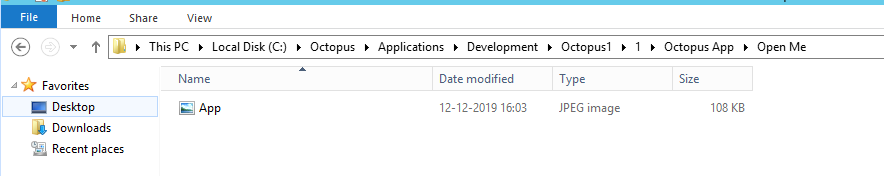


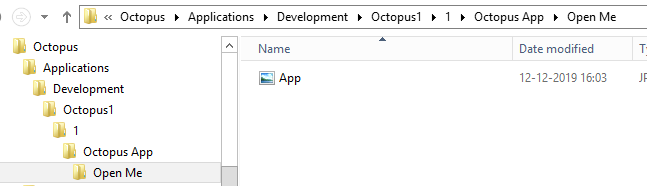
This will deploy the release. In the task summary, you'll see the release was deployed to your Test environment, and the Package is deployed on the deployment target, in this case Web Server.

## Demo 1: Deploy a Package to a Server





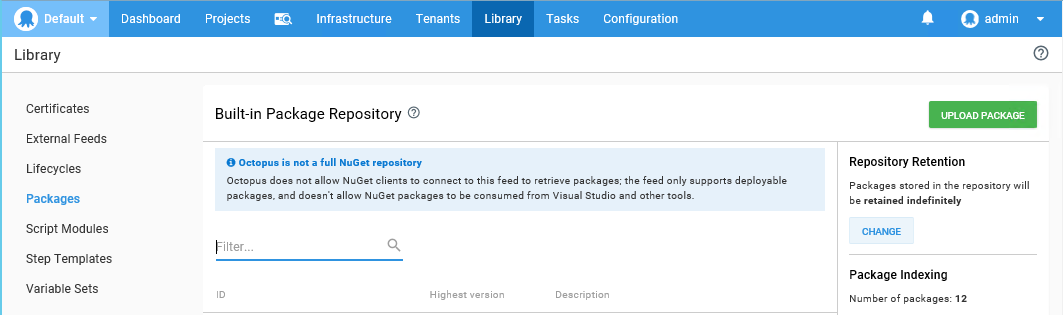




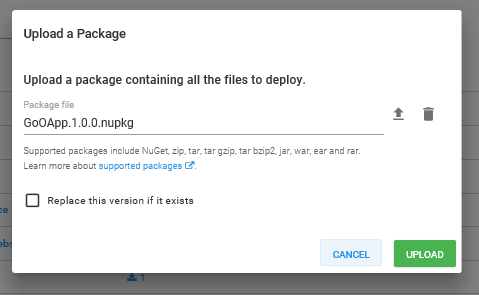
## Demo 2: Deploying a .Net application to windows Server

### Upload a Package into Repository:-

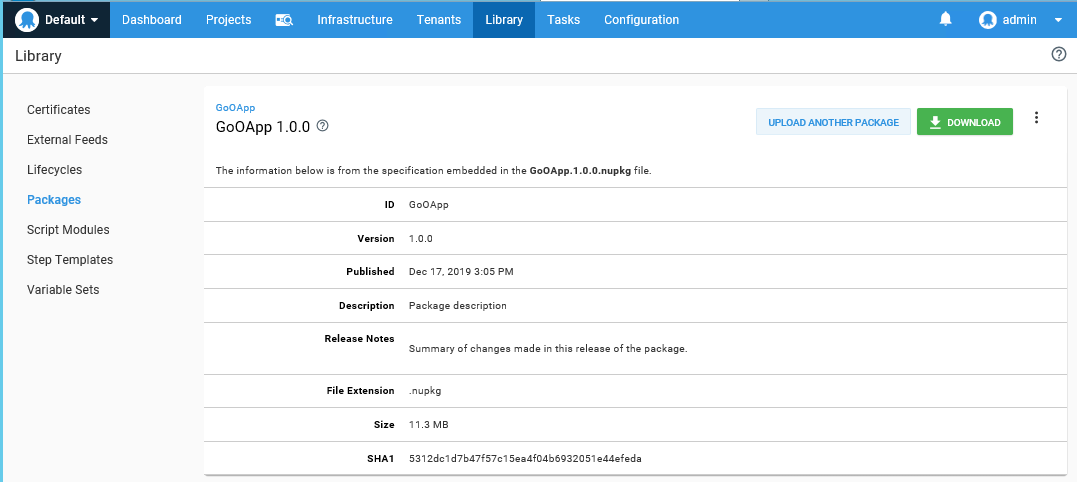
Click on **UPLOAD PACKAGE** and select the desired package to deploy



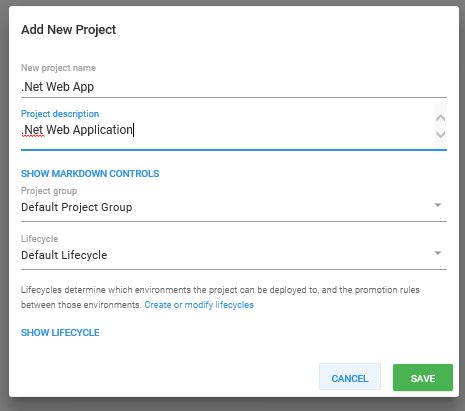
my case: GoOApp.1.0.0.nupkg



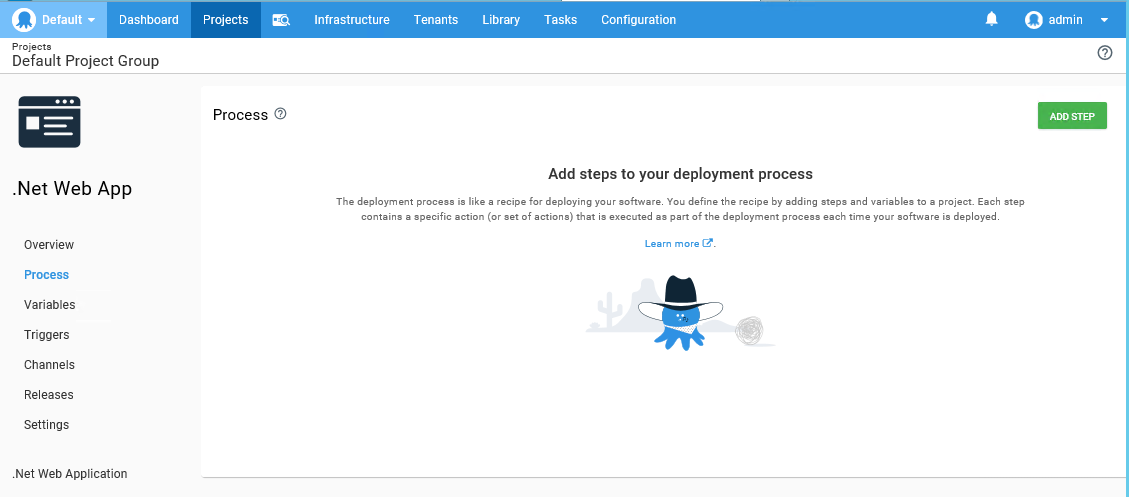
Click **UPLOAD**.



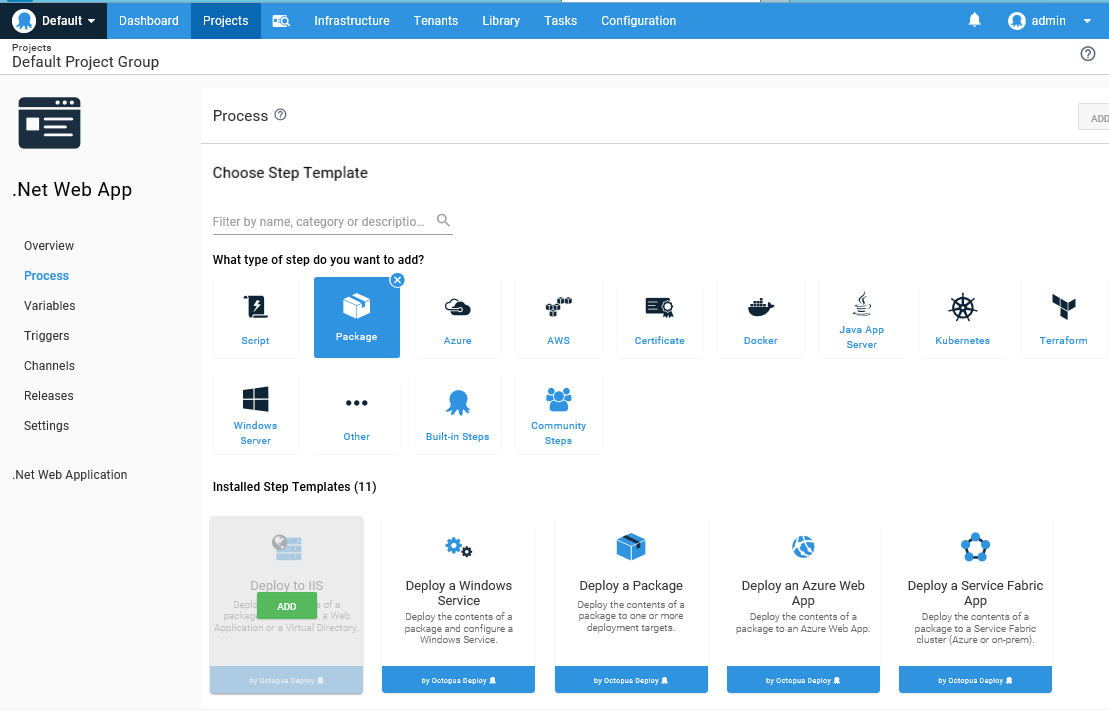
### Go to Projects> Add New Project



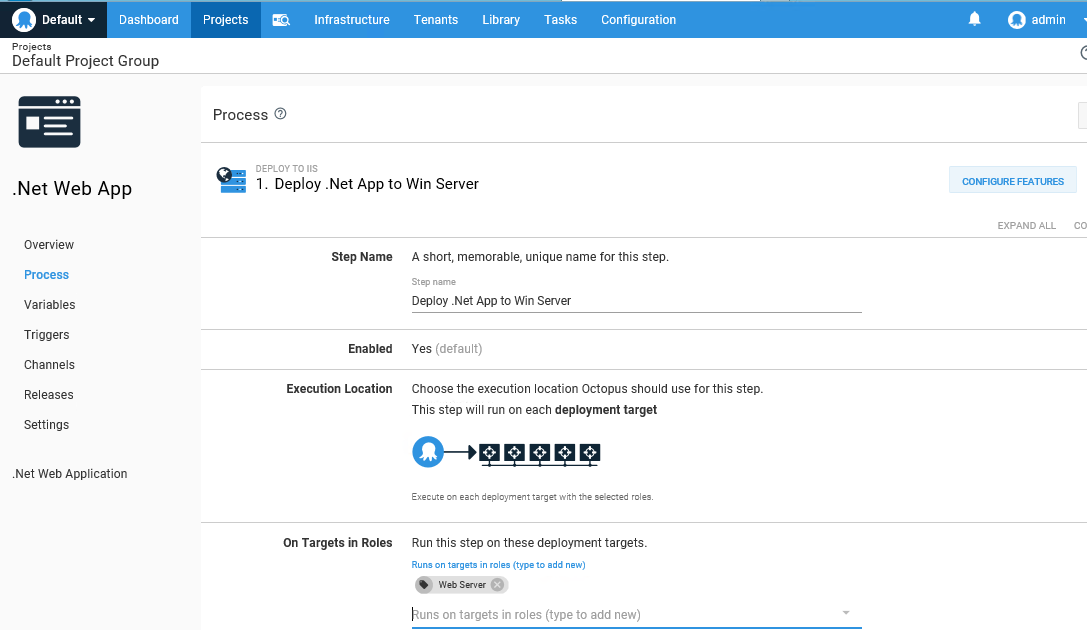
Go to Process> **ADD STEP**

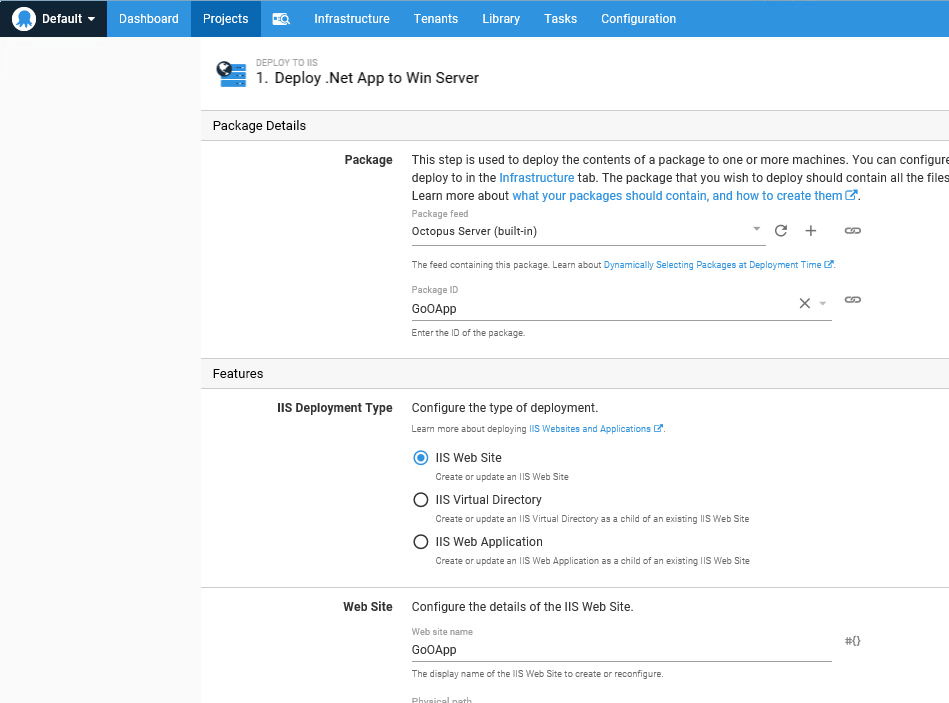


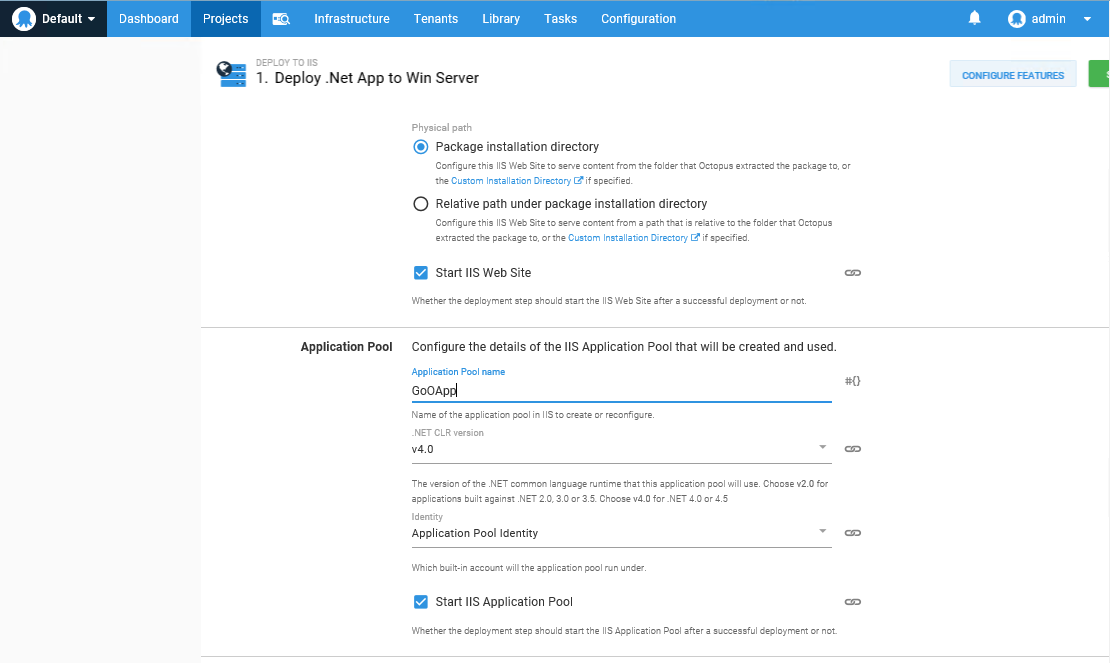
Select Package> **Deploy to IIS**



Fill required details about the project and save

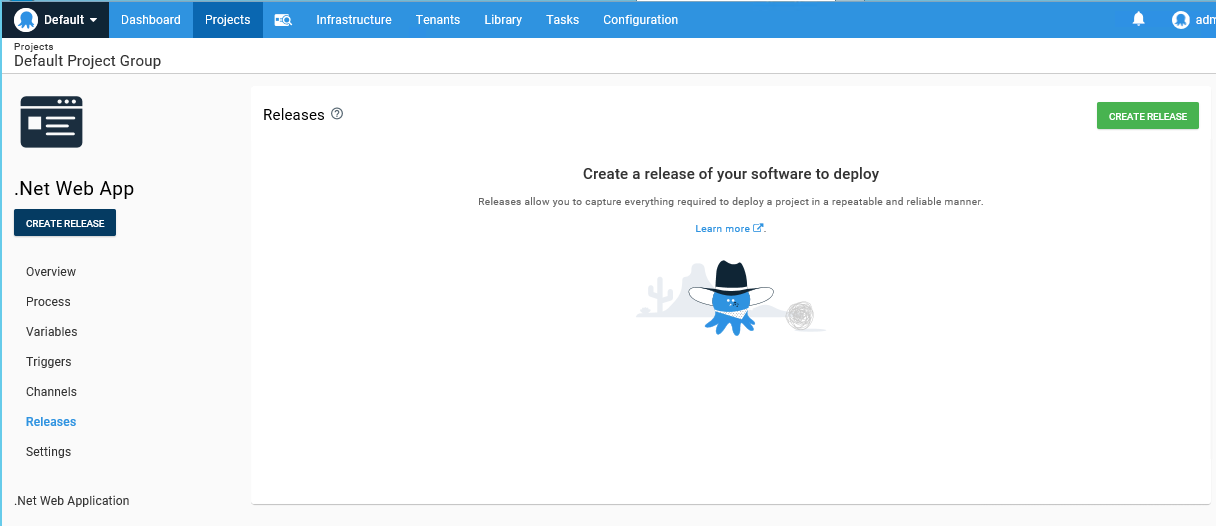




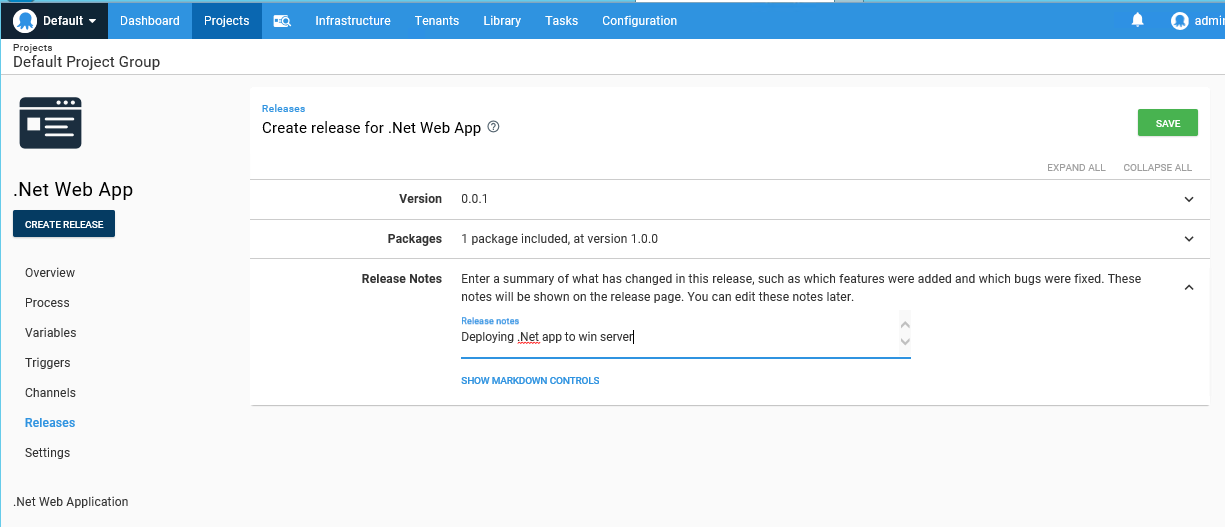




### Go to Release> CREATE RELEASE



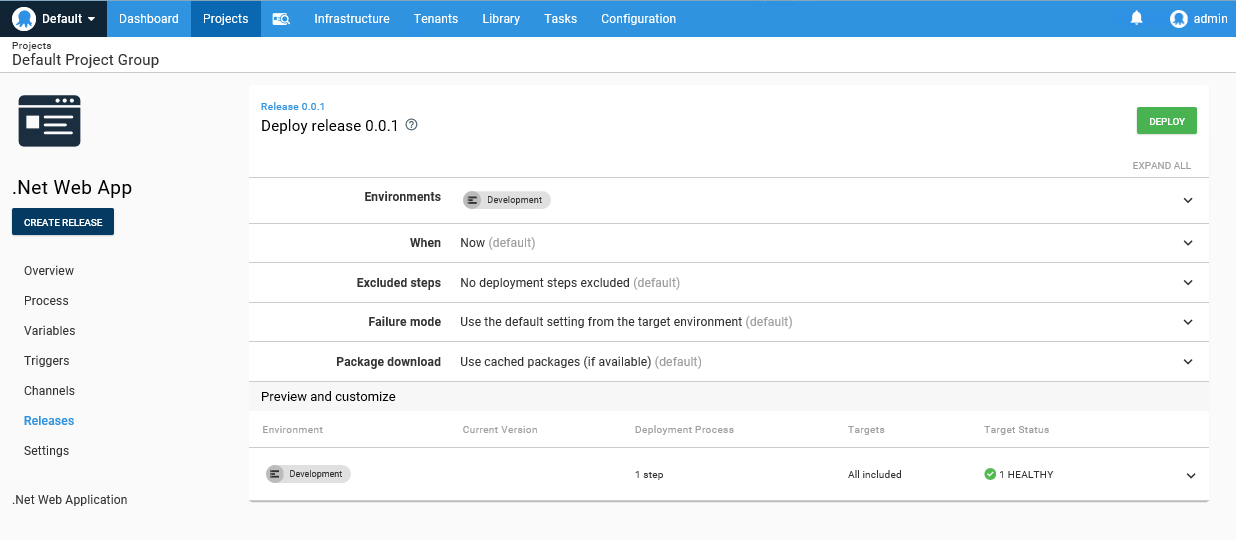
Select the version and **SAVE**.



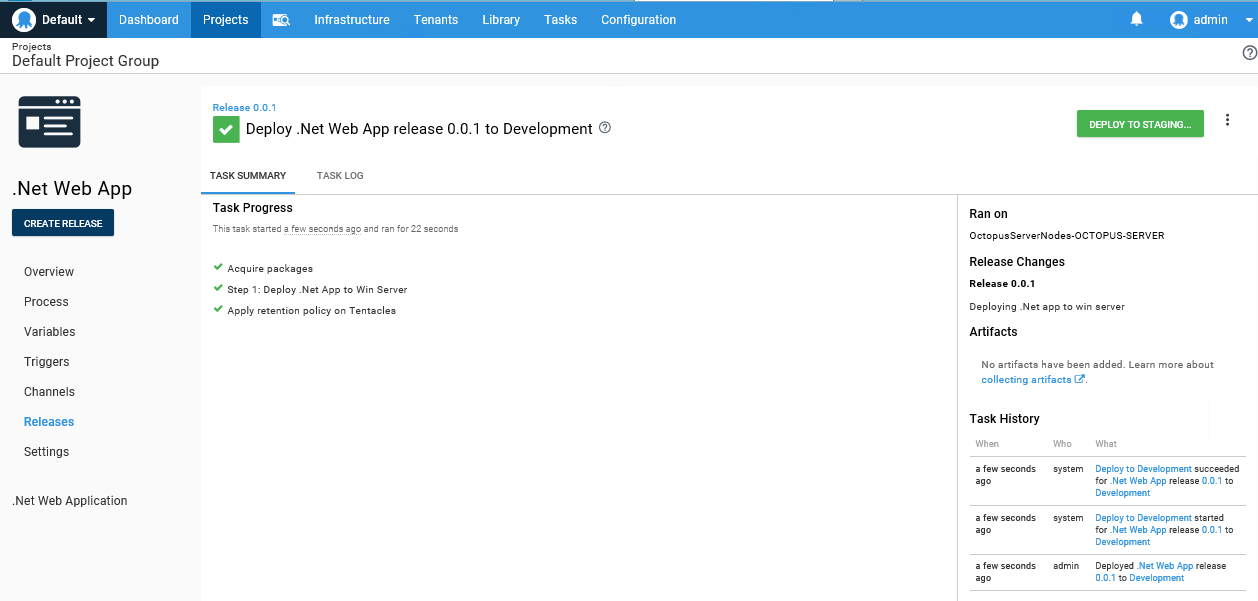
Click on **DEPLOY TO DEVELOPMENT**



### Click on DEPLOY



Deployment status will be shown once it is done. **Task Progress**



### Successfully Deployed .Net Application to windows server.

