# Exercise 6: Deploying a SharePoint Framework solution

## Task 1: Create your project

1. From the PowerShell command prompt, change to the C:/LabFiles/SharePoint directory by executing the following command: cd c:/LabFiles/SharePoint
2. Make a new directory for your SharePoint project files by executing the following command: md DeploymentDemo
3. Navigate to the newly created SharePoint directory by executing the following command: cd DeploymentDemo
4. Run the SharePoint Yeoman generator by executing the following command: yo @microsoft/sharepoint
5. Use the following to complete the prompt that is displayed:
   * **What is your solution name?**: DeploymentDemo
   * **Which baseline packages do you want to target for your component(s)?**: SharePoint Online only (latest)
   * **Where do you want to place the files?**: Use the current folder
   * **Do you want to allow the tenant admin the choice of being able to deploy the solution to all sites immediately without running any feature deployment or adding apps in sites?**: No
   * **Will the components in the solution require permissions to access web APIs that are unique and not shared with other components in the tenant?**: No
   * **Which type of client-side component to create?**: WebPart
   * **What is your Web part name?**: Deployment Demo
   * **What is your Web part description?**: Deployment Demo description
   * **Which framework would you like to use?**: No JavaScript framework
6. After provisioning the folders required for the project, the generator will install all the dependency packages using NPM.

## Task 2: Creating a deployment package for the project

1. When NPM completes downloading all dependencies, build the project by running the following command on the command line from the root of the project: gulp build
2. Create a production bundle of the project by running the following command on the command line from the root of the project: gulp bundle –ship
3. Create a deployment package of the project by running the following command on the command line from the root of the project: gulp package-solution –ship

## Task 3: Deploying the package to a SharePoint site

1. In a browser, navigate to your SharePoint tenant’s App Catalog site.
2. Select the **Apps for SharePoint** link in the left navigation pane.
3. Drag the package created in the previous steps, located in the project’s **./sharepoint/solution/deployment-demo.sppkg**, into the **Apps for SharePoint** library.
4. SharePoint will launch a dialog box asking if you want to trust the package.
5. Select **Deploy**.

## Task 4: Installing the SharePoint package in a site collection

1. Navigate to an existing site collection, or create a new one.
2. Select **Site Contents** from the left navigation pane.
3. From the **New** menu, select **App**.
4. Locate the solution you previously deployed and select it.

SharePoint will start to install the application. At first it will appear dimmed, but after a few moments you should see it listed.

## Task 5: Adding the web part to a page

1. Navigate to a SharePoint page.
2. Put it in edit mode by selecting the **Edit** button in the upper-right portion of the content area on the page.
3. Select the web part icon button to open the list of available web parts.
4. Select the expand icon, a diagonal line with two arrows in the upper-right corner, to expand the web part toolbox.
5. Scroll to the bottom; locate and select the **Deployment Demo** web part.

## Task 6: Examining the deployed web part files

1. Once the page loads, open the browser’s developer tools and navigate to the **Sources** tab.
2. Refresh the page and examine where the JavaScript bundle is being hosted.

* **Note**: If you have not enabled the Office 365 CDN then the bundle will be hosted from a document library named ClientSideAssets in the app catalog site. If you have enabled the Office 365 CDN then the bundle will be automatically hosted from the CDN. ## Review

In this exercise, you learned how to package and deploy your SharePoint Framework solutions.