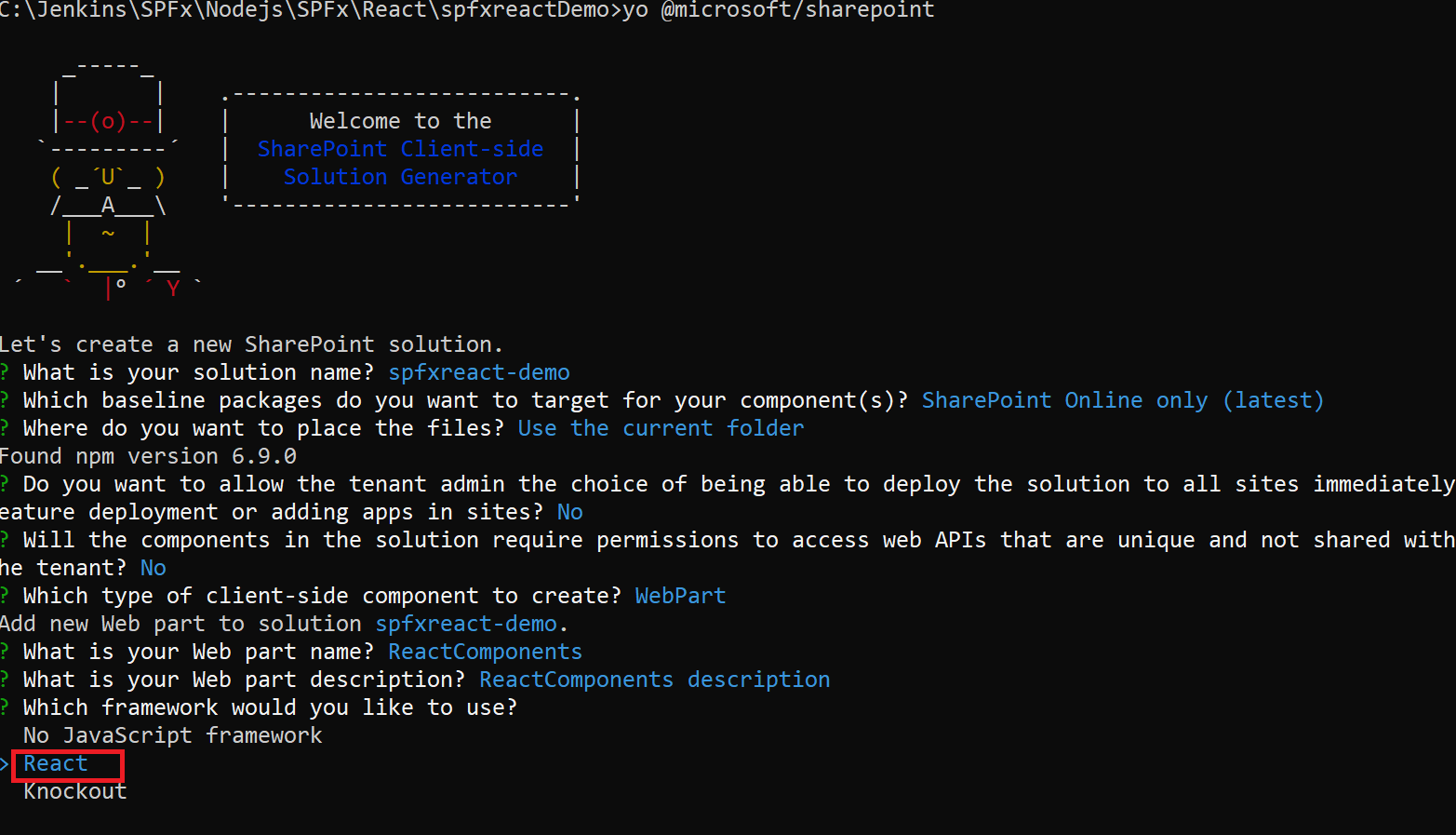
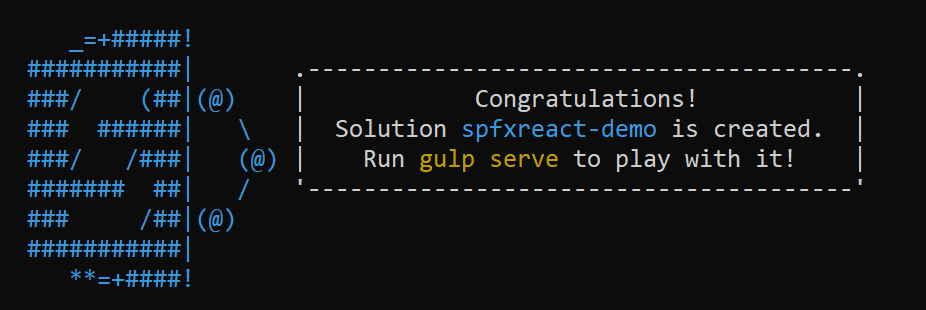
Deploy & Host SPfx Webpart

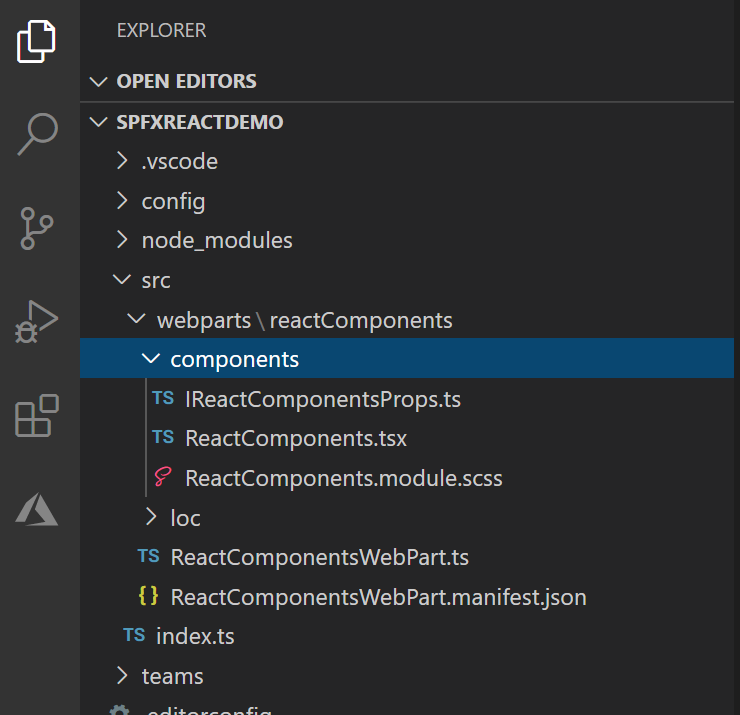
1. Create a folder “spfxreactDemo” under your favourite folder
2. Move to the folder “spfxreactDemo”
3. Create a webpart same as previous example, and select the framework as React



1. Enter to create a Webpart



1. Open the solution in Visual Studio Code
2. Type **Code .**



1. Add required functionality.
2. Test the webpart

gulp serve

If you are doing deployment first time in this tenant, then follow **Lab 2 - Create the App Catalog site collection.docx** to create or verify app Catalog available or not.

Then Enable Public CDN to deploy the webpart in office 365 app catalog

REF for planning - <https://docs.microsoft.com/en-us/office365/enterprise/use-office-365-cdn-with-spo>

**Office 365 Content Delivery Network (CDN)**

Microsoft introduced Office 365 CDN capability, which is used to host your assets directly from your Office 365 Tenant and it can be accessed by SharePoint sites and add-ins with the help of CDN URLs.

**One-time activity \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

Download SharePoint Online Management Shell - <https://www.microsoft.com/en-in/download/details.aspx?id=35588>

* Steps involved to enable Public CDN
  + Open SharePoint Online Management Shell as administrator
  + Run the command given below in the Management Shell to connect to your SharePoint Online Tenant
    - Connect-SPOService -Url <https://tenantname-admin.sharepoint.com>
  + Run the command given below to enable CDN in your Tenant.
    - Set-SPOTenantCdnEnabled -CdnType Public
  + Create SPSite and Create a new document library in one of your SharePoint Online sites within your Tenant. I have created a folder , which will be added as new CDN origin.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Review solution settings**

1. Go to your webpart
2. Open **package-solution.json** from the **config** folder.
3. The **package-solution.json** file defines the package metadata as shown in the following code:

{

"$schema": "https://developer.microsoft.com/json-schemas/spfx-build/package-solution.schema.json",

"solution": {

"name": "helloword-webpart-client-side-solution",

"id": "3c1af394-bbf0-473c-bb7d-0798f0587cb7",

"version": "1.0.0.0",

"includeClientSideAssets": true,

"isDomainIsolated": false

},

"paths": {

"zippedPackage": "solution/helloword-webpart.sppkg"

}

}

The default value for the includeClientSideAssets is true, which means that static assets are packaged automatically in the **\*.sppkg** files, and you don't need to separately host your assets from an external system.

Do not change this setting for this exercise, so that assets are automatically hosted when solution is deployed to your tenant.

If Microsoft 365 CDN is enabled, it's used automatically with default settings. If Microsoft 365 CDN isn't enabled, assets are served from the app catalog site collection. This means that if you leave the includeClientSideAssets setting true, your solution assets are automatically hosted in the tenant.

## Prepare web part assets to deploy

1. Compile the code

gulp clean

gulp build

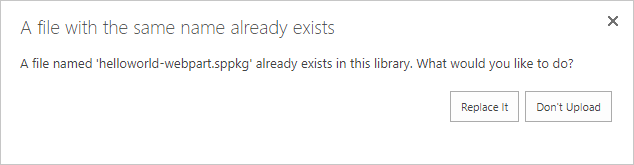
1. Execute the following task to bundle your solution. This executes a release build of your project by using a dynamic label as the host URL for your assets. This URL is automatically updated based on your tenant CDN settings.

gulp bundle --ship

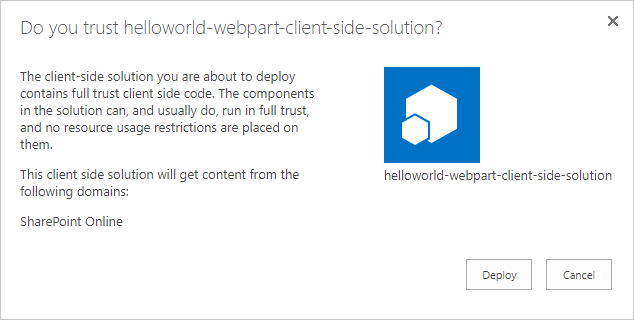
1. Execute the following task to package your solution. This creates an updated **spfxreactDemo-webpart.sppkg** package on the **sharepoint/solution** folder.

gulp package-solution --ship

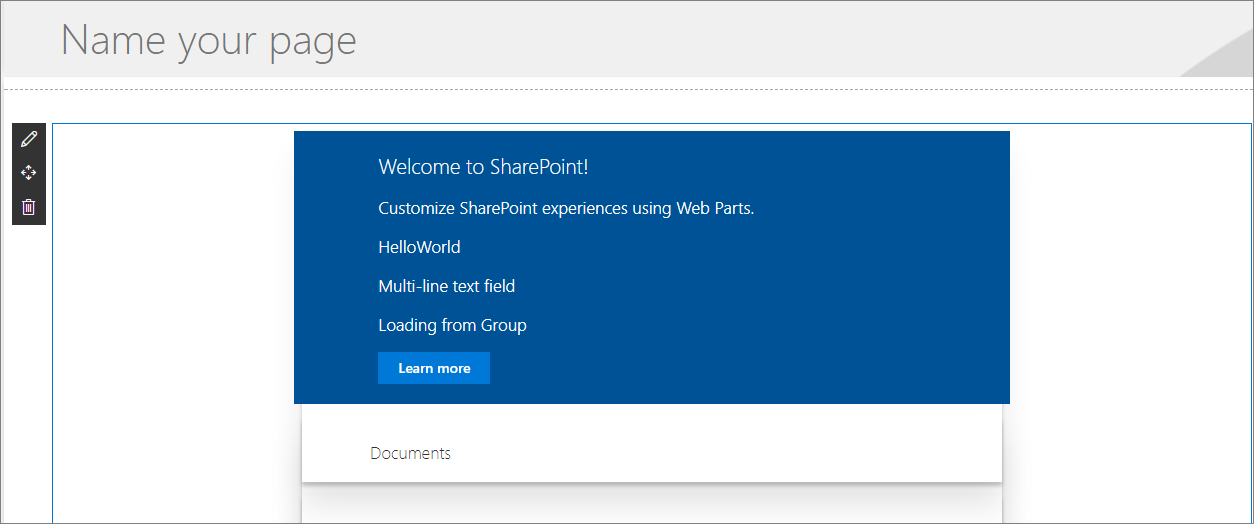
1. Upload or drag and drop the newly created client-side solution package to the app catalog in your tenant.
2. Because you already deployed the package, you're prompted as to whether to replace the existing package. Select **Replace It** if you are deploying second time.



1. Notice how the **domain** list in the prompt says *SharePoint Online*. This is because the content is either served from the Microsoft 365 CDN or from the app catalog, depending on the tenant settings. Select **Deploy**.



1. Open the site where you previously installed the **spfxreactDemo-webpart-client-side-solution** or installed the solution to a new site.
2. After the solution has been installed, select **Add an app** from the *gear* menu, and select **spfxreactDemo** from the modern page web part picker to add your custom web part to page.
3. Notice how the web part is rendered even though you're not running the local web server.



1. Save changes on the page with the web part.