

# Packaging and Deploying SharePoint Framework Solutions



# Agenda

- Creating the App Catalog Site in SharePoint Online
- Packaging SPFX Solutions for Distribution
- Publishing and Updating SPFX Solutions
- Deploying Webpart Assets in Azure Storage



# Understanding the App Catalog



- App publishing scheme based on App Catalog
  - App packages are published (uploaded) to app catalog
  - Provides better app discovery, installation and upgrade
- You can create one tenant-level app catalog
  - Allow tenant admins to control deployment
  - You must explicitly create tenant-wide app catalog
- You can create app catalog for a site collection
  - Provides site collection owners with control
  - App catalog added to site collection using PowerShell



# Creating the App Catalog Site Collection

- You must create the App Catalog site collection
  - You can create it by hand in SharePoint admin center
  - You can create it with PowerShell script if required

**Create App Catalog Site Collection**

<b>Title</b>	<input type="text" value="App Catalog Site"/>		
<b>Web Site Address</b>	<input type="text" value="https://msd0911.sharepoint.com"/>		
	<input type="text" value="/sites/"/>	<input type="text" value="AppCatalogSite"/>	
<b>Language Selection</b>	Select a language: <input type="text" value="English"/>		
<b>Time Zone</b>	<input type="text" value="(UTC-05:00) Eastern Time (US and Canada)"/>		
<b>Administrator</b>	<input type="text" value="Stu Dent"/>		 
<b>Server Resource Quota</b>	<input type="text" value="0"/>	resources of 5600 resources available	
		<input type="button" value="OK"/>	<input type="button" value="Cancel"/>



# Apps for SharePoint Document Library

- Apps for SharePoint is special document library
  - It's the place where you publish SPFx solutions
  - You upload solution package and enter the related metadata

Apps for SharePoint

Upload completed (1 added) Refresh

New Upload Sync Share More

All Apps Featured Apps Unavailable Apps Find a file SAVE THIS VIEW

✓	📄	Title	Name	App Version	Edit	Product ID	Metadata Language	Default Metadata Language	Modified	Enabled	Valid App Package	Deployed	Tenant Deployed
Product ID : {C6BBEB95-A296-4B4F-A96E-4C72A6177687} (1)													
✓	📄	Global Navigation	global-navigation	1.0.0.0	✎	{C6BBEB95-A296-4B4F-A96E-4C72A6177687}	English - 1033	Yes	A few seconds ago	Yes	Yes	Yes	No
Product ID : {D6C9AA37-2D69-4426-B373-7141797A3536} (1)													
✓	📄	Customer Search	customer-search	1.0.0.0	✎	{D6C9AA37-2D69-4426-B373-7141797A3536}	English - 1033	Yes	17 minutes ago	Yes	Yes	Yes	Yes







**DEMO**

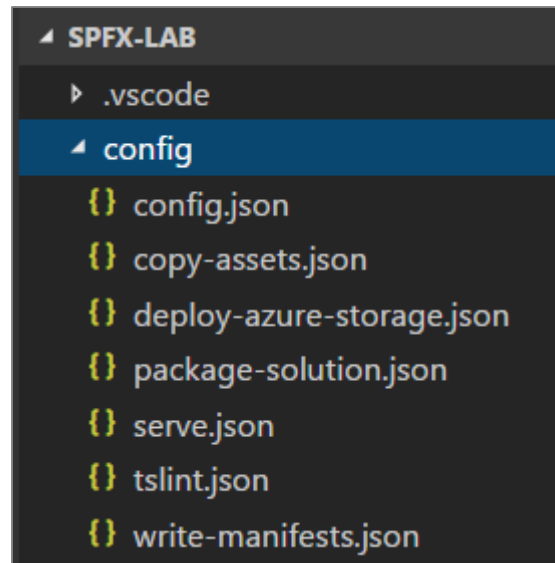
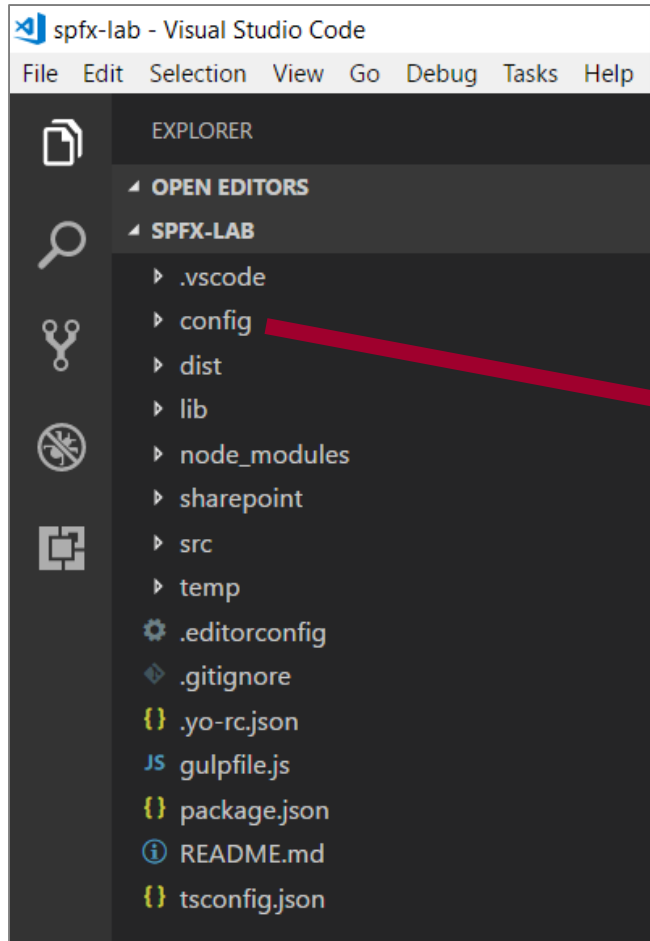
## Creating an App Catalog Site

# Agenda

- ✓ Creating the App Catalog Site in SharePoint Online
- Packaging SPFX Solutions for Distribution
  - Publishing and Updating SPFX Solutions
  - Deploying Webpart Assets in Azure Storage



# SPFx Project Configuration Files





# package-solution.json

- package-solution.json
  - Contains top-level project properties (`id`, `name`, `version`)
  - `includeClientSideAssets`
  - `skipFeatureDeployment`
  - `zippedPackage`

```
{  
  "$schema": "https://dev.office.com/json-schemas/spfx-build/package-solution.schema.json",  
  "solution": {  
    "name": "SPFX Lab",  
    "id": "aa5d865d-3cfb-4fd6-a4d4-6fc5d21dd9d7",  
    "version": "1.0.0.0",  
    "includeClientSideAssets": true,  
    "skipFeatureDeployment": true  
  },  
  "paths": {  
    "zippedPackage": "solution/spfx-lab.sppkg"  
  }  
}
```

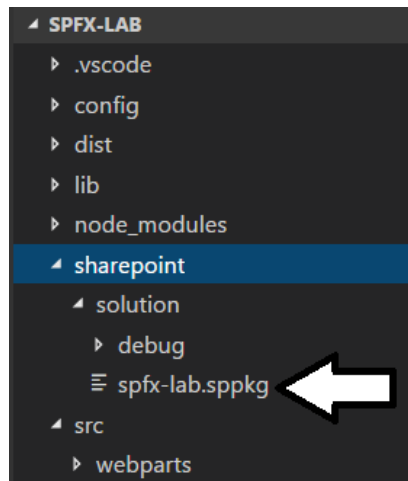


# Building a SPFx Solution

- Done by executing gulp package-solution
  - Can be done with or without **--ship** parameter

```
PS C:\Demos\spfx-lab> gulp package-solution
Build target: DEBUG
[20:02:30] Using gulpfile C:\Demos\spfx-lab\gulpfile.js
[20:02:30] Starting gulp
[20:02:30] Starting 'package-solution'...
[20:02:30] Starting subtask 'configure-sp-build-rig'...
[20:02:30] Finished subtask 'configure-sp-build-rig' after 4.95 ms
[20:02:30] Starting subtask 'package-solution'...
[20:02:30] Warning - [package-solution] This is not a production build (--ship or --production).
```

- Generates zip archive with **.sppkg** extension



# Packaging a SPFx Solution for Distribution

- `gulp bundle --ship`
  - Uses dynamic URL for assets based on tenant CDN settings

```
PS C:\Demos\spfx-lab> gulp bundle --ship
Build target: SHIP
[20:51:24] Using gulpfile C:\Demos\spfx-lab\gulpfile.js
[20:51:24] Starting gulp
[20:51:24] Starting 'bundle'...
[20:51:24] Starting subtask 'configure-sp-build-rig'...
[20:51:24] Finished subtask 'configure-sp-build-rig' after 4.45 ms
```

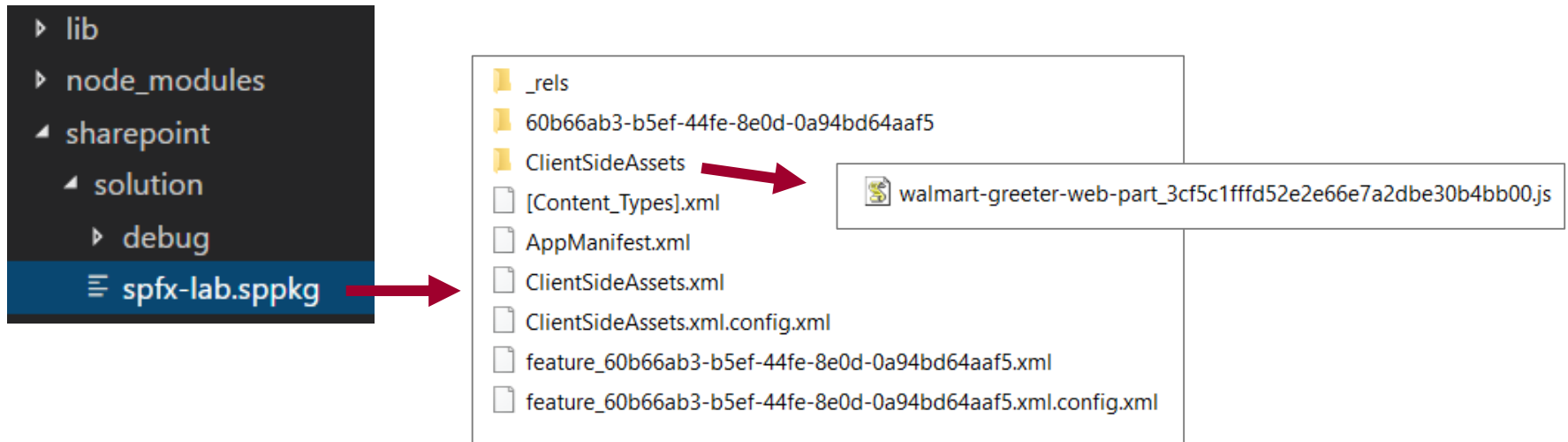
- `gulp package-solution --ship`
  - Uses dynamic URL for assets based on tenant CDN settings

```
PS C:\Demos\spfx-lab> gulp package-solution --ship
Build target: SHIP
[20:53:42] Using gulpfile C:\Demos\spfx-lab\gulpfile.js
[20:53:42] Starting gulp
[20:53:42] Starting 'package-solution'...
[20:53:42] Starting subtask 'configure-sp-build-rig'...
[20:53:42] Finished subtask 'configure-sp-build-rig' after 5.02 ms
[20:53:42] Starting subtask 'package-solution'...
```



# Inside a SPFx Solution Package

- Your .js code is included in ClientSideAssets





# Agenda

- ✓ Creating the App Catalog Site in SharePoint Online
- ✓ Packaging SPFX Solutions for Distribution
- Publishing and Updating SPFX Solutions
- Deploying Webpart Assets in Azure Storage



# Deploying Solution to Office 365 Tenancy

## Apps for SharePoint ⓘ

Upload completed (1 added) [Refresh](#)

[New](#) [Upload](#) [Sync](#) [Share](#) [More](#) ▾

[All Apps](#) [Featured Apps](#) [Unavailable Apps](#) ...  [SAVE THIS VIEW](#)


✓	📄	Title	Name		App Version	Edit	Product ID	Metadata Language	Default Metadata Language	Modified	Enabled	Valid App Package	Deployed	Tenant Deployed
Product ID : {C6BBEB95-A296-4B4F-A96E-4C72A6177687} (1)														
⇒	📄	Global Navigation	global-navigation ✳	...	1.0.0.0	📄	{C6BBEB95-A296-4B4F-A96E-4C72A6177687}	English - 1033	Yes	A few seconds ago	Yes	Yes	Yes	No
Product ID : {D6C9AA37-2D69-4426-B373-7141797A3536} (1)														
	📄	Customer Search	customer-search ✳	...	1.0.0.0	📄	{D6C9AA37-2D69-4426-B373-7141797A3536}	English - 1033	Yes	17 minutes ago	Yes	Yes	Yes	Yes



# includeClientSideAssets = True

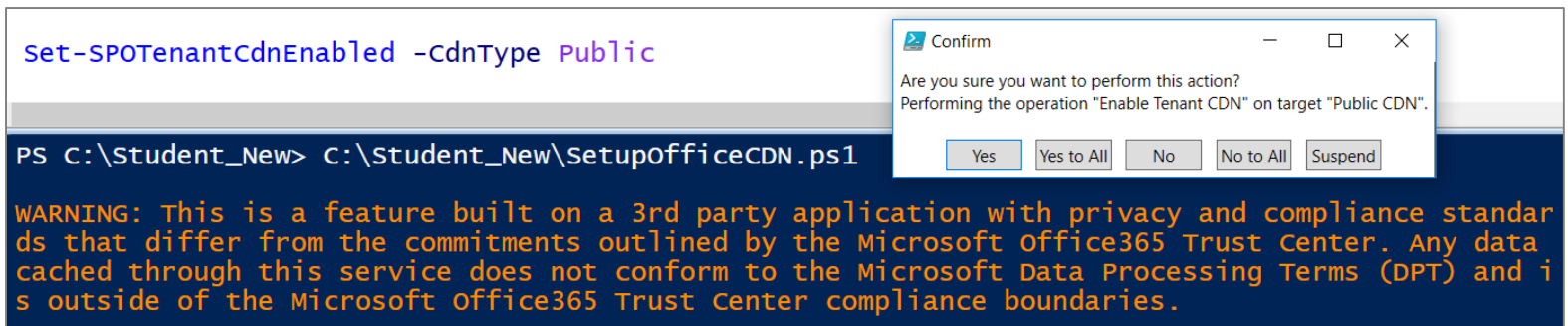
- If Office 365 CDN is enabled...
  - it will be used automatically with default settings.
- If Office 365 CDN is not enabled...
  - assets will be served from app catalog site collection.

```
{ } package-solution.json ●
1  {
2    "$schema": "https://dev.office.com/json-schemas/spfx-build/package-solution.schema.json",
3    "solution": {
4      "name": "SPFX Lab",
5      "id": "aa5d865d-3cfb-4fd6-a4d4-6fc5d21dd9d7",
6      "version": "1.0.0.0",
7      "includeClientSideAssets": true,
8      "skipFeatureDeployment": true
9    },
10   "paths": {
11     "zippedPackage": "solution/spfx-lab.sppkg"
12   }
13 }
```



# Enabling the Office 365 CDN

- Office 365 provides CDN for SPFx solution deployment
  - Office 365 CDN must be enabled using PowerShell



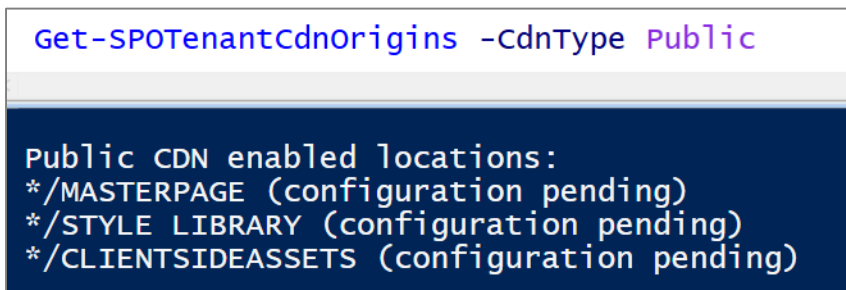
The screenshot shows a PowerShell terminal window with the command `Set-SPOTenantCdnEnabled -cdnType Public` entered. Below the command, a warning message is displayed: "WARNING: This is a feature built on a 3rd party application with privacy and compliance standards that differ from the commitments outlined by the Microsoft Office365 Trust Center. Any data cached through this service does not conform to the Microsoft Data Processing Terms (DPT) and is outside of the Microsoft Office365 Trust Center compliance boundaries." Overlaid on the terminal is a "Confirm" dialog box asking "Are you sure you want to perform this action?" and "Performing the operation 'Enable Tenant CDN' on target 'Public CDN'". The dialog has buttons for "Yes", "Yes to All", "No", "No to All", and "Suspend".

```
Set-SPOTenantCdnEnabled -cdnType Public
```

PS C:\Student\_New> C:\Student\_New\SetupOfficeCDN.ps1

WARNING: This is a feature built on a 3rd party application with privacy and compliance standards that differ from the commitments outlined by the Microsoft Office365 Trust Center. Any data cached through this service does not conform to the Microsoft Data Processing Terms (DPT) and is outside of the Microsoft Office365 Trust Center compliance boundaries.

- Enabling CDN creates `*/CLIENTSIDEASSETS` origin



The screenshot shows a PowerShell terminal window with the command `Get-SPOTenantCdnOrigins -cdnType Public` entered. The output lists the enabled locations: "Public CDN enabled locations:", "\*/MASTERPAGE (configuration pending)", "\*/STYLE LIBRARY (configuration pending)", and "\*/CLIENTSIDEASSETS (configuration pending)".

```
Get-SPOTenantCdnOrigins -cdnType Public
```

Public CDN enabled locations:  
\*/MASTERPAGE (configuration pending)  
\*/STYLE LIBRARY (configuration pending)  
\*/CLIENTSIDEASSETS (configuration pending)

- CDN supports these file type extensions  
CSS, EOT, GIF, ICO, JPEG, JPG, JS, MAP, PNG, SVG, TTF, WOFF





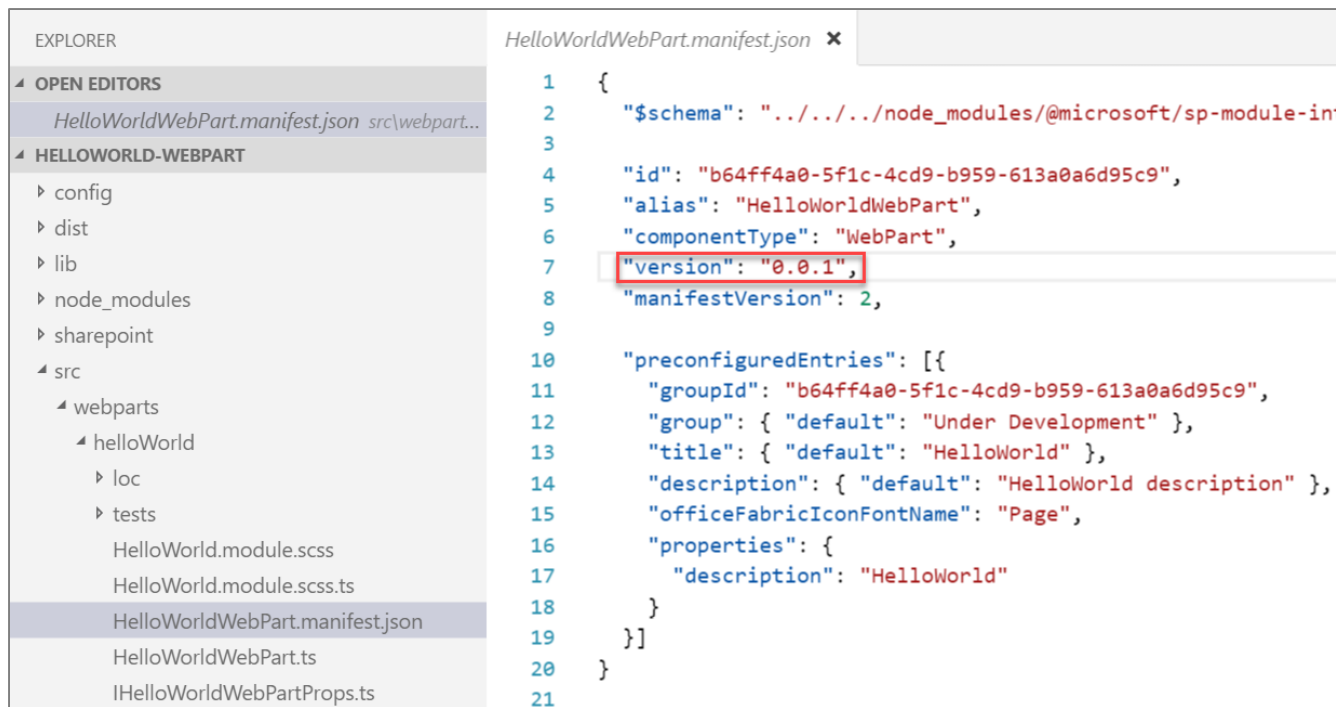
# Update Webparts and Increment Version

Update the code or configuration for your web part

Increment the version in the <web part name>.manifest.json file

This sets the version for the web part

Multiple web parts in the same solution may be versioned independently



```
1  {
2    "$schema": "../../node_modules/@microsoft/sp-module-int
3
4    "id": "b64ff4a0-5f1c-4cd9-b959-613a0a6d95c9",
5    "alias": "HelloWorldWebPart",
6    "componentType": "WebPart",
7    "version": "0.0.1",
8    "manifestVersion": 2,
9
10   "preconfiguredEntries": [{
11     "groupId": "b64ff4a0-5f1c-4cd9-b959-613a0a6d95c9",
12     "group": { "default": "Under Development" },
13     "title": { "default": "HelloWorld" },
14     "description": { "default": "HelloWorld description" },
15     "officeFabricIconFontName": "Page",
16     "properties": {
17       "description": "HelloWorld"
18     }
19   }]
20 }
21
```



# Update Version in package-solution.json

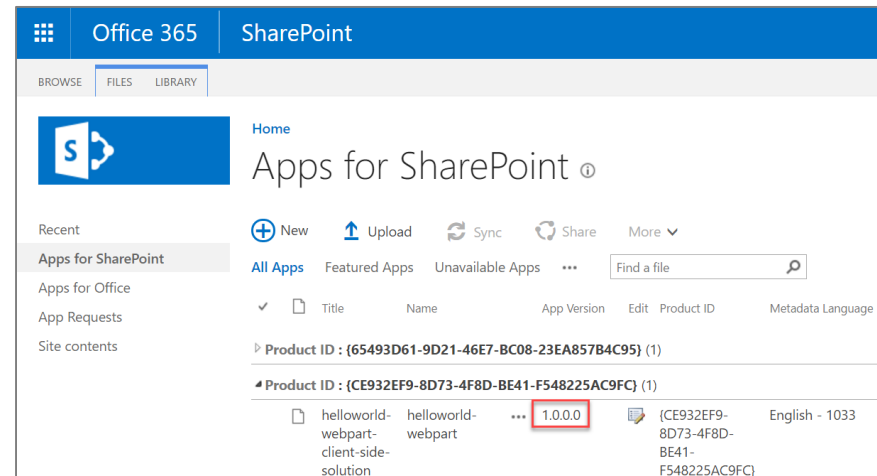
This sets the version for the .sppkg Add-in  
This version is displayed in the app catalog



The image shows a Visual Studio Code window with the Explorer sidebar on the left and the package-solution.json file open in the editor. The Explorer sidebar shows the project structure with folders like .vscode, config, and files like package-solution.json. The package-solution.json file contains the following JSON content:

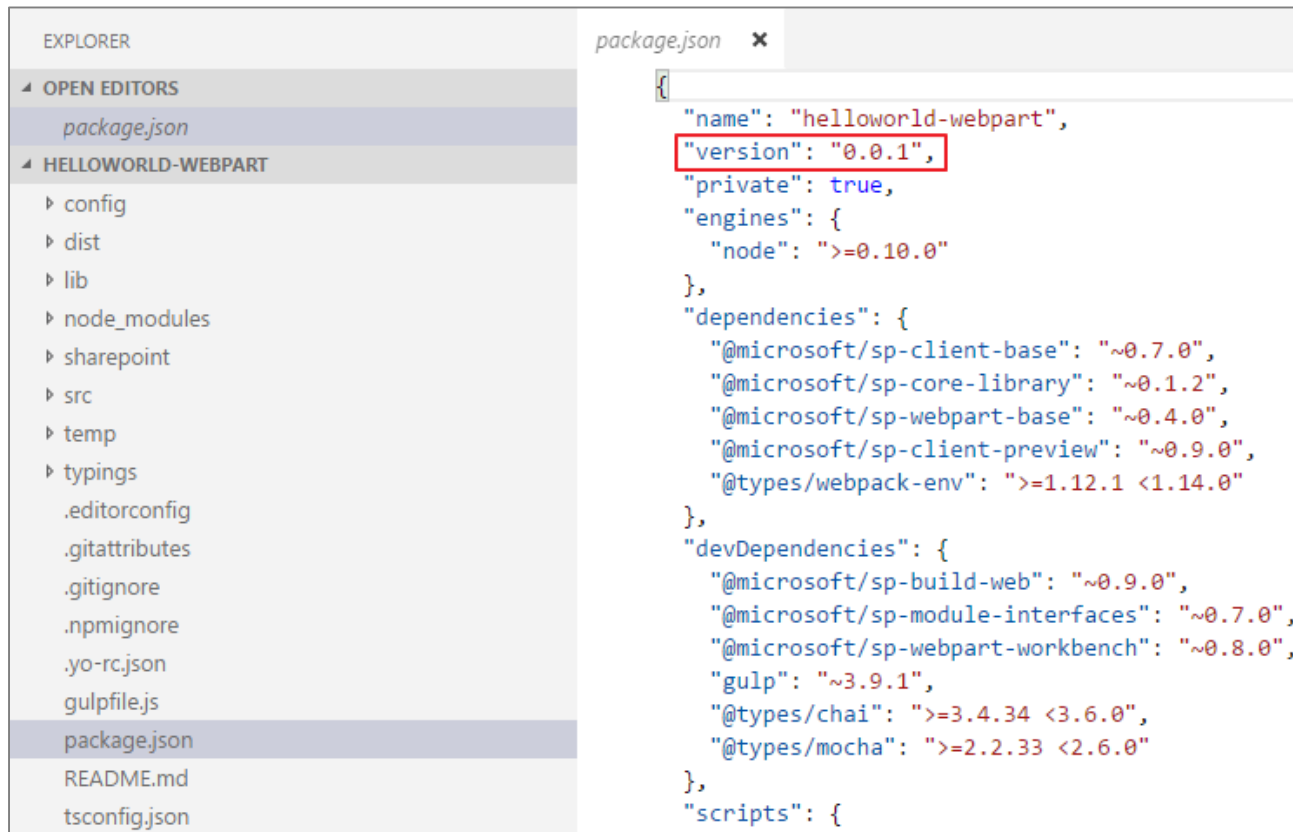
```
{
  "solution": {
    "name": "helloworld-webpart-client-side-solution",
    "id": "bd5dbf97-9507-44f6-9e49-a5547f30e5ec",
    "version": "1.0.0.0"
  },
  "paths": {
    "zippedPackage": "solution/helloworld-webpart.sppkg"
  }
}
```

The "version" field is highlighted with a red box.



# Update the version in the package.json file

- Change the version every time the package changes



The screenshot shows the Visual Studio Code interface. On the left, the Explorer sidebar displays the file structure of a project named 'HELLOWORLD-WEBPART'. The 'package.json' file is selected and highlighted. The main editor area on the right shows the contents of 'package.json'. The 'version' field, which is currently '0.0.1', is highlighted with a red rectangular box. The JSON content is as follows:

```
{
  "name": "helloworld-webpart",
  "version": "0.0.1",
  "private": true,
  "engines": {
    "node": ">=0.10.0"
  },
  "dependencies": {
    "@microsoft/sp-client-base": "~0.7.0",
    "@microsoft/sp-core-library": "~0.1.2",
    "@microsoft/sp-webpart-base": "~0.4.0",
    "@microsoft/sp-client-preview": "~0.9.0",
    "@types/webpack-env": ">=1.12.1 <1.14.0"
  },
  "devDependencies": {
    "@microsoft/sp-build-web": "~0.9.0",
    "@microsoft/sp-module-interfaces": "~0.7.0",
    "@microsoft/sp-webpart-workbench": "~0.8.0",
    "gulp": "~3.9.1",
    "@types/chai": ">=3.4.34 <3.6.0",
    "@types/mocha": ">=2.2.33 <2.6.0"
  },
  "scripts": {
```



# Agenda

- ✓ Creating the App Catalog Site in SharePoint Online
- ✓ Packaging SPFX Solutions for Distribution
- ✓ Publishing and Updating SPFX Solutions
- Deploying Webpart Assets in Azure Storage





# Deploying Client-side Assets to a CDN

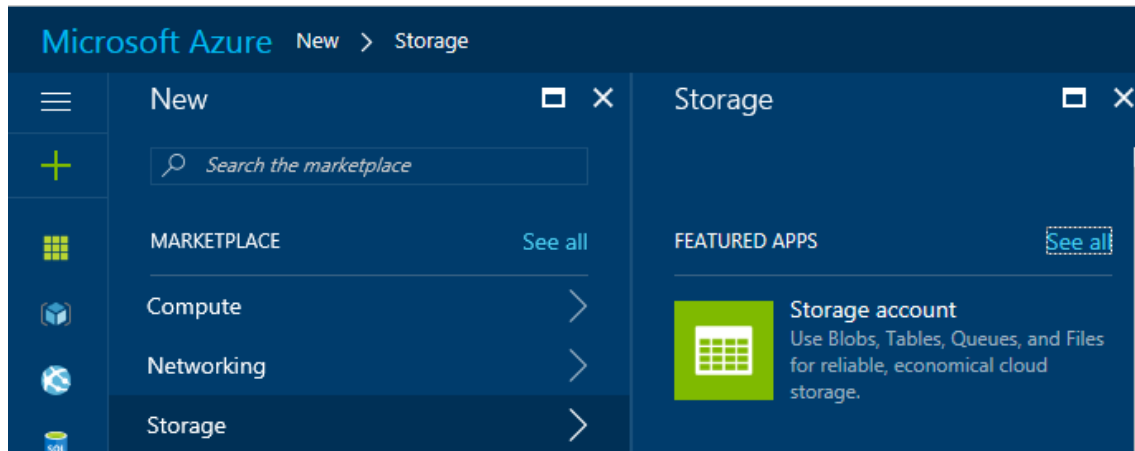
- What happens when **includeClientSideAssets** is false?
  - You must deploy the web part assets to a CDN location
  - You must update the `cdnBasePath` in the `write-manifests.json` file

```
{ } package-solution.json •
1  {
2    "$schema": "https://developer.microsoft.com/json-schemas/spfx-build/package-solution.schema.json",
3    "solution": {
4      "name": "spfx-lab",
5      "id": "ab2ff8d9-ddbc-430f-adab-20011ff4e840",
6      "version": "1.0.0.0",
7      "includeClientSideAssets": false,
8      "skipFeatureDeployment": true
9    },
10   "paths": {
11     "zippedPackage": "solution/spfx-lab.sppkg"
12   }
13 }
```



# Create Azure Storage Account

- Go to the Azure Management Portal
- Create a new Storage Account

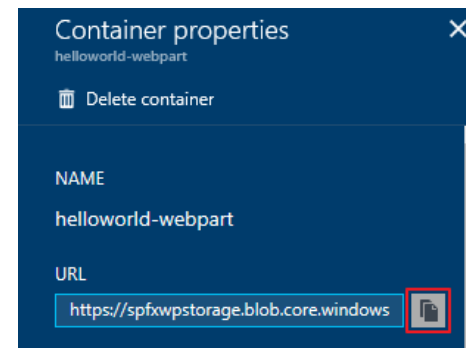
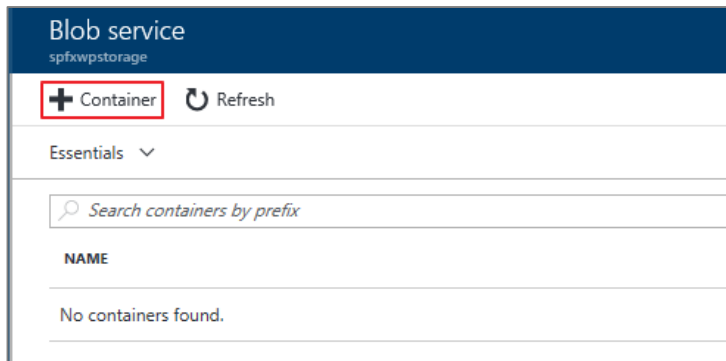
The screenshot displays the 'Create storage account' configuration form. At the top, it states: 'The cost of your storage account depends on the usage and the options you choose below. [Learn more](#)'. The form includes several fields and options:

- Name:** 'spfxwpstorage' (with a green checkmark).
- Deployment model:** 'Resource manager' (selected) and 'Classic'.
- Account kind:** 'General purpose' (dropdown).
- Performance:** 'Standard' (selected) and 'Premium'.
- Replication:** 'Read-access geo-redundant storage (RA...)' (dropdown).
- Storage service encryption:** 'Disabled' (selected) and 'Enabled'.
- Subscription:** A dropdown menu.
- Resource group:** 'Create new' (selected) and 'Use existing'. Below it, 'spfxwprg' is listed with a green checkmark.
- Location:** 'West US' (dropdown).

At the bottom, there is a 'Pin to dashboard' checkbox and a 'Create' button. A link for 'Automation options' is also present.

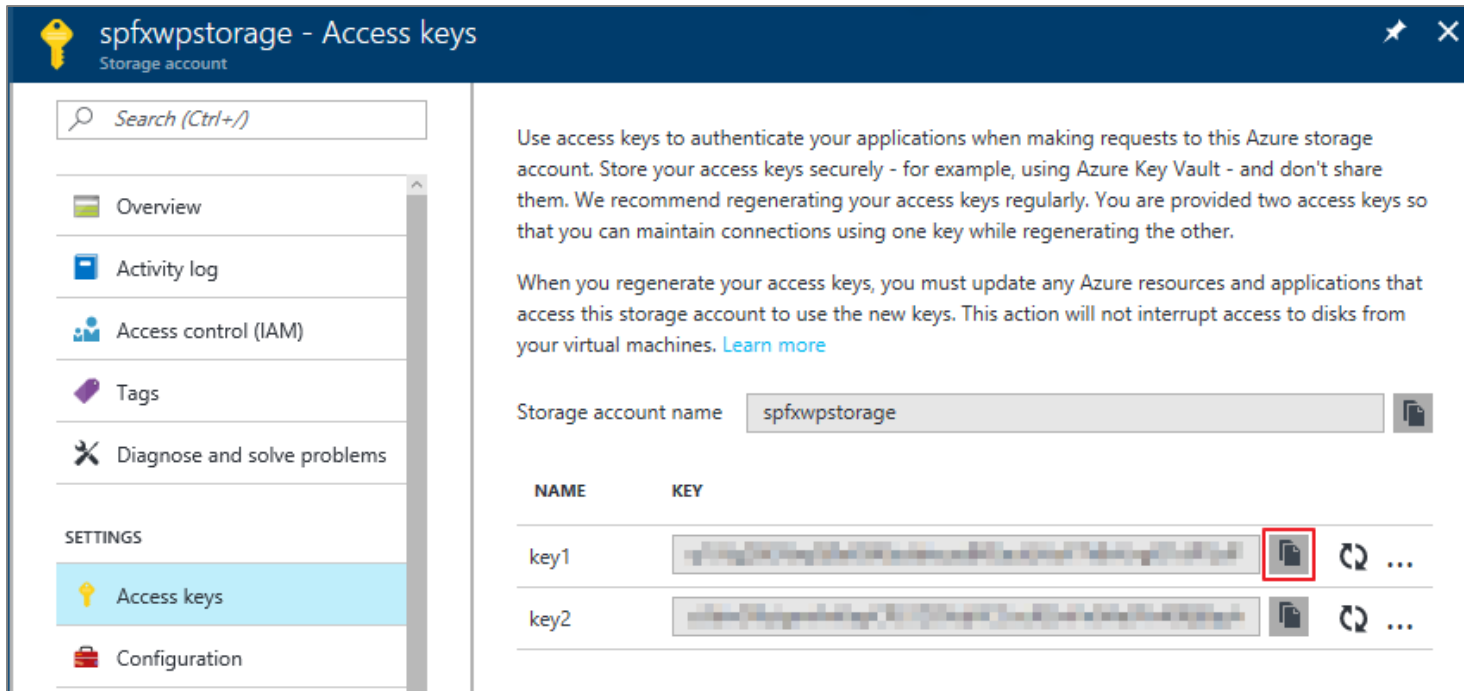
# Create Blob Container in Storage Account

- Create a Blob Container and copy the URL



# Obtain Storage Account Access Key

- The Storage Account Access Key is used to automate deployments to the Azure Storage CDN, copy one of the keys



The screenshot shows the 'Access keys' page for the storage account 'spfxwpstorage'. The left sidebar contains navigation links: Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, and a SETTINGS section with 'Access keys' (highlighted) and 'Configuration'. The main content area includes instructions on using access keys and a table of the two keys. The first key, 'key1', has its value highlighted with a red box, and a copy icon is visible next to it.

spfxwpstorage - Access keys  
Storage account

Search (Ctrl+/)





Overview  
Activity log  
Access control (IAM)  
Tags  
Diagnose and solve problems

SETTINGS  
Access keys  
Configuration

Use access keys to authenticate your applications when making requests to this Azure storage account. Store your access keys securely - for example, using Azure Key Vault - and don't share them. We recommend regenerating your access keys regularly. You are provided two access keys so that you can maintain connections using one key while regenerating the other.

When you regenerate your access keys, you must update any Azure resources and applications that access this storage account to use the new keys. This action will not interrupt access to disks from your virtual machines. [Learn more](#)

Storage account name: spfxwpstorage

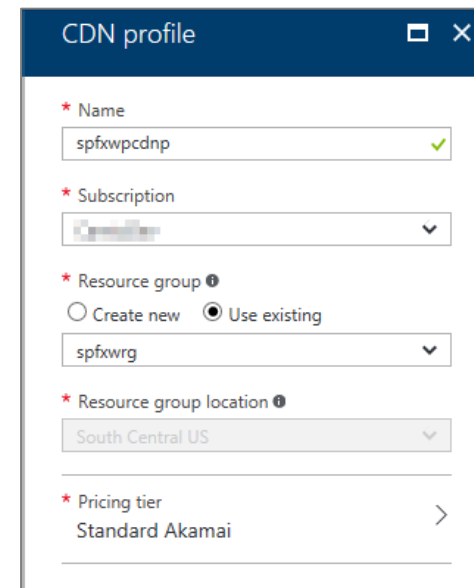
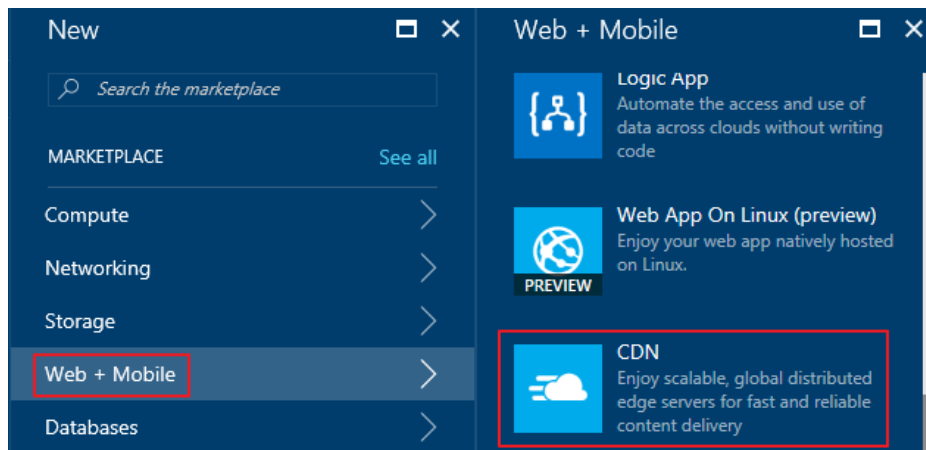
NAME	KEY
key1	[Key Value]   ...
key2	[Key Value]   ...





# Create CDN Profile

- In the Azure Management Portal, create a new CDN Profile



# Create CDN Endpoint

- In the CDN Profile, create a new CDN endpoint

### Add an endpoint

Allows configuring content delivery behavior an...

\* Name  
spfxwpcdne ✓  
.azureedge.net

\* Origin type  
Storage ▼

\* Origin hostname ⓘ  
spfxwpstorage.blob.core.windows.net ▼

Origin path ⓘ  
/Path

Origin host header ⓘ  
spfxwpstorage.blob.core.windows.net ✓

Protocol ⓘ	Origin port ⓘ
<input checked="" type="checkbox"/> HTTP	80
<input checked="" type="checkbox"/> HTTPS	443



# Configure Webpart to Deploy Assets to CDN

- Update the account, container, and accessKey in the deploy-azure-storage.json file
  - account – Storage account name
  - container – Name of the container you wish to use for the web part
  - accessKey – Storage Account Access Key

```
{  
  "workingDir": "./temp/deploy/",  
  "account": "spfxwpstorage",  
  "container": "helloworld-webpart",  
  "accessKey": "hL2503 ... "  
}
```



# Configure the web part to use the CDN

- Create CDN base path  
`https://<Storage Account Name>.blob.core.windows.net/<Container Name>`
- Update the **cdnBasePath** in the **write-manifests.json** file

```
{ write-manifests.json •  
{  
  "$schema": "https://developer.microsoft.com/json-schemas/spfx-build/write-manifests.schema.json",  
  "cdnBasePath": "https://spfxwpstorage.blob.core.windows.net/helloworld-webpart"  
}
```



# Deploy web part assets to Azure

- Use the deploy-azure-storage gulp task to deploy the assets to the Azure Storage Account

```
> gulp deploy-azure-storage
```

```
> gulp deploy-azure-storage
Build target: DEBUG
[11:56:24] Using gulpfile C:\SPFx\helloworld-webpart\gulpfile.js
[11:56:24] Starting gulp
[11:56:24] Starting 'deploy-azure-storage'...
[11:56:24] Starting subtask 'deploy-azure-storage'...
[11:56:24] [deploy-azure-storage] Uploading files '**/*.*' from directory './temp/deploy/' to Azure
[11:56:25] [deploy-azure-storage] Created container: helloworld-webpart
[11:56:25] [deploy-azure-storage] Uploading [3] files...
[11:56:26] [deploy-azure-storage] Uploaded file: helloworld-webpart-helloworldstrings_en-us_536e65149b0acf4d52c0043073b9fc59.js
[11:56:26] [deploy-azure-storage] Uploaded file: hello-world.bundle_b8a80975dedeb31de300b580fab61182.js
[11:56:26] [deploy-azure-storage] Uploaded file: dd331d09-a9cd-448d-a687-7e43060191e2.json
[11:56:26] [deploy-azure-storage] Upload complete!
[11:56:26] [deploy-azure-storage] Access your files at: https://spfxwpstorage.blob.core.windows.net/helloworld-webpart
[11:56:26] Finished subtask 'deploy-azure-storage' after 1.92 s
[11:56:26] Finished 'deploy-azure-storage' after 1.93 s
[11:56:26] =====[ Finished ]=====
[11:56:27] Project helloworld-webpart version: 0.0.1
[11:56:27] Build tools version: 2.4.0
[11:56:27] Node version: v6.10.0
[11:56:27] Total duration: 5.94 s
```



# Summary

- ✓ Creating the App Catalog Site in SharePoint Online
- ✓ Packaging SPFX Solutions for Distribution
- ✓ Publishing and Updating SPFX Solutions
- ✓ Deploying Webpart Assets in Azure Storage

