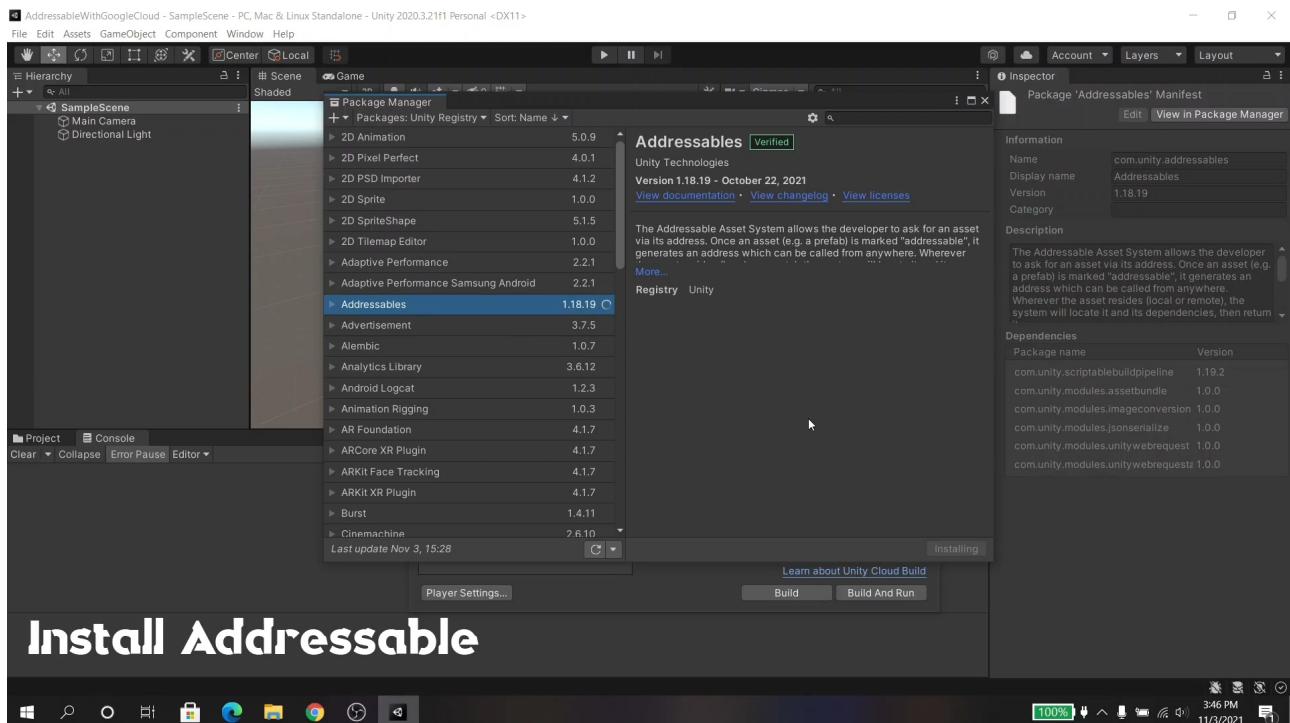


# This tutorial will guide you to set up Addressable Remote content to Private Google Cloud

Note: Only your bundle in private, your catalog will be public

You need to go throw some step:

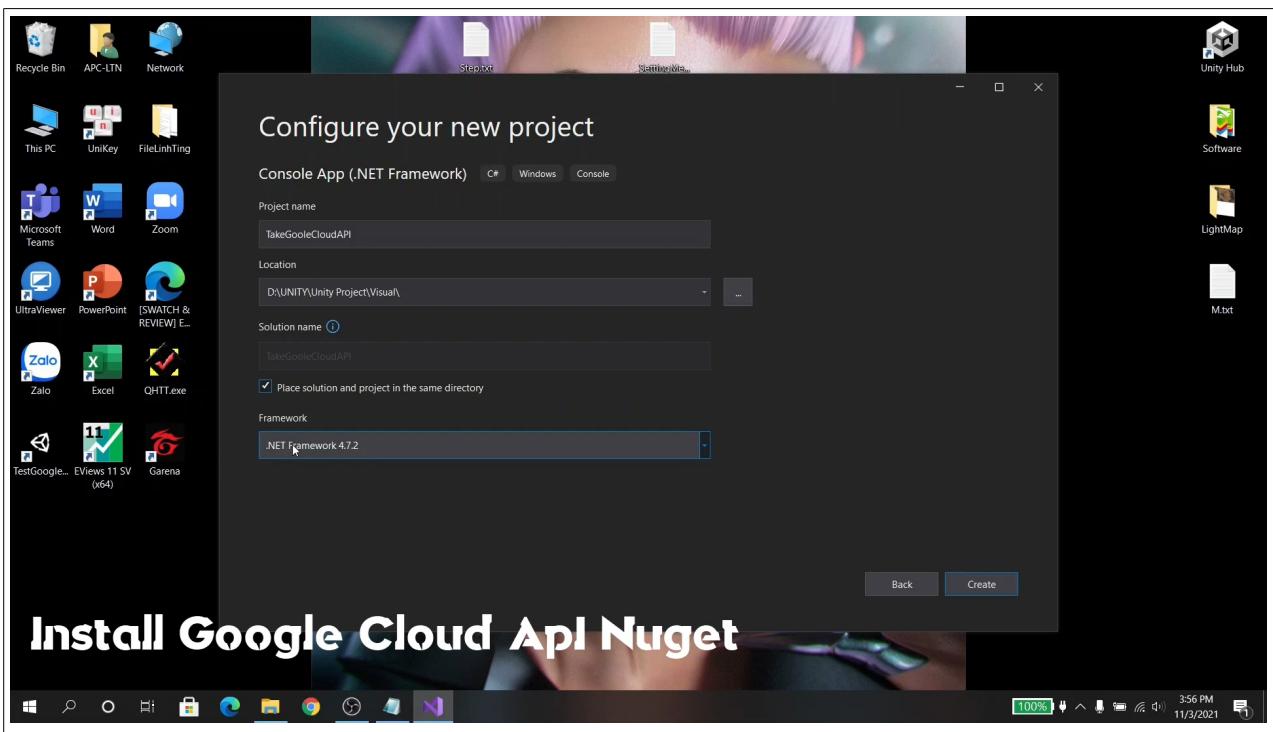
## Step 1: Set up Unity Project and Addressable Package



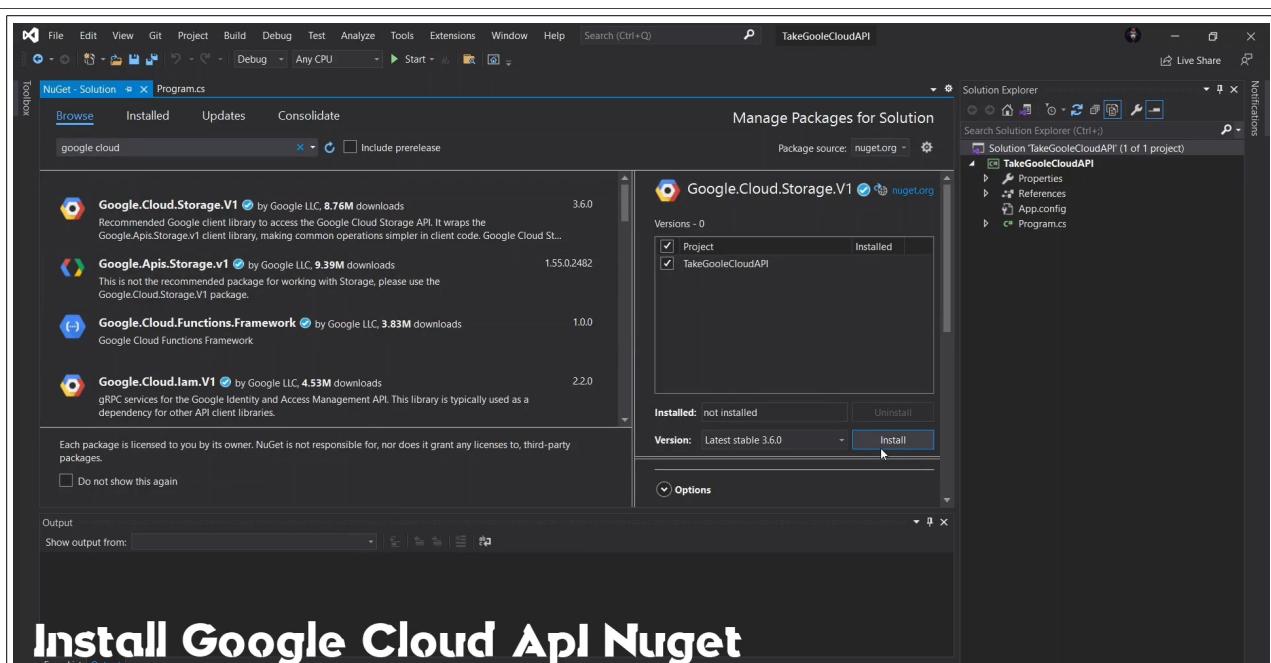
## Step 2: Set up Google Cloud API to Google Cloud (Cause unity not allow install Nuget throw Visual Studio)

\* This base on tutorial: <https://docs.microsoft.com/en-us/visualstudio/gamedev/unity/unity-scripting-upgrade>.

1: Use Visual Studio Create a empty C# project Exam:  
**TakeGoogleCloudAPI**



2: Use Nuget Manager Install `Google.Cloud.Storage.V1` to **TakeGoogleCloudAPI** (Cause `Google.Cloud.Storage.V1` need some dependent package, and dependent package also need other dependent package -> let it auto install all of that)



## Install Google Cloud Api Nugget

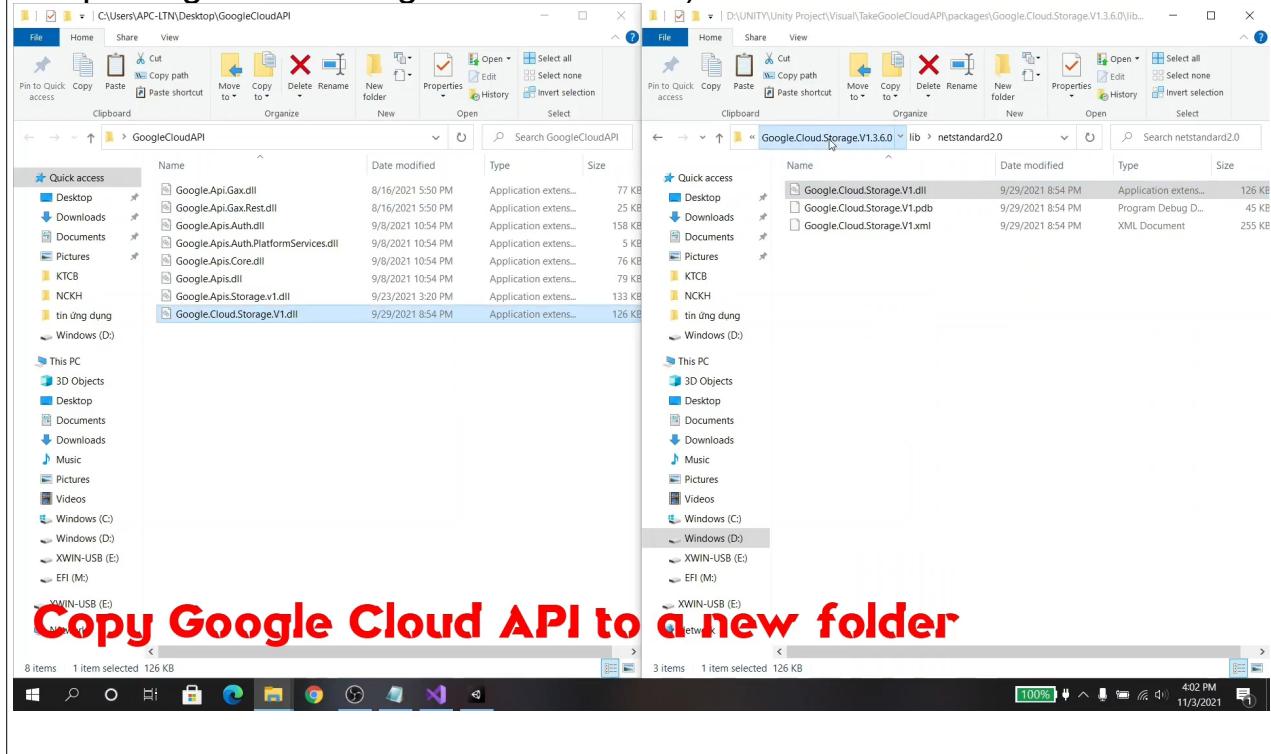
3: In Unity Project, Create folder Plugins inside Assets Folder

4: Go to ProjectA/Pakage go inside each folder and choose dll you need .Net2 or .Net4x:

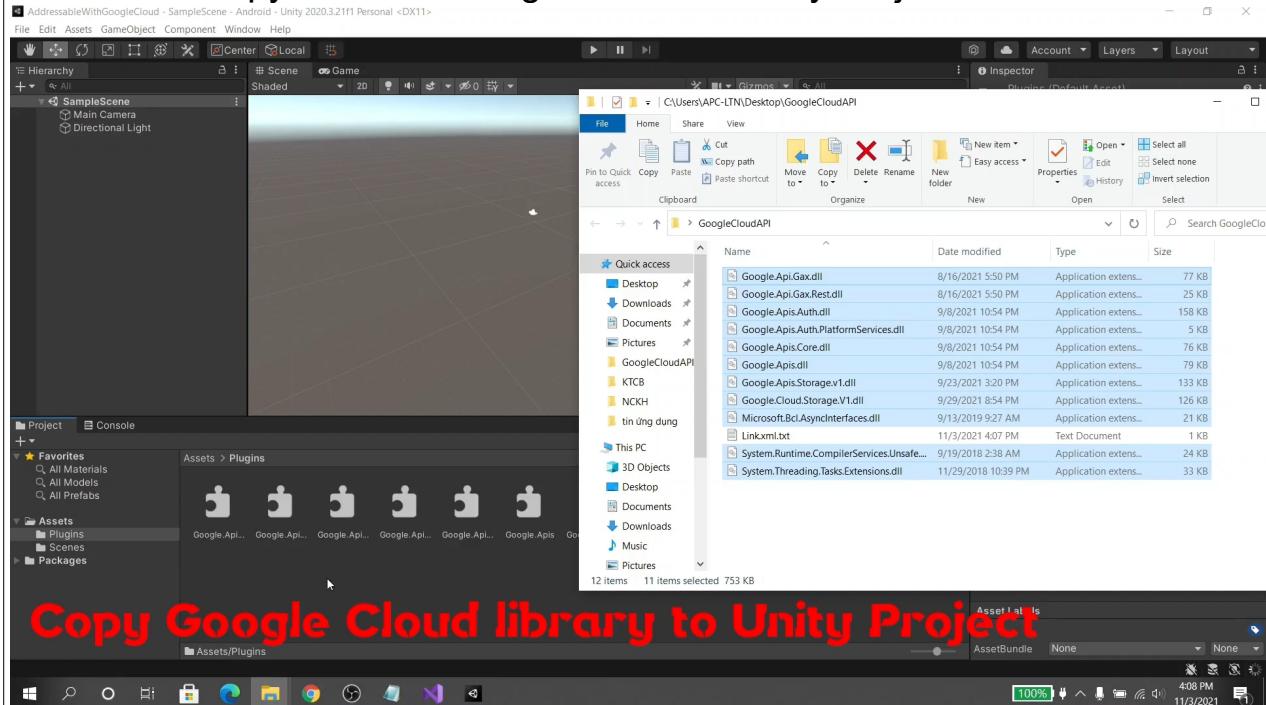
**Exam:** you Use .Net2 in Unity Project

=> Copy lib/netstandard2.0 inside each folder in

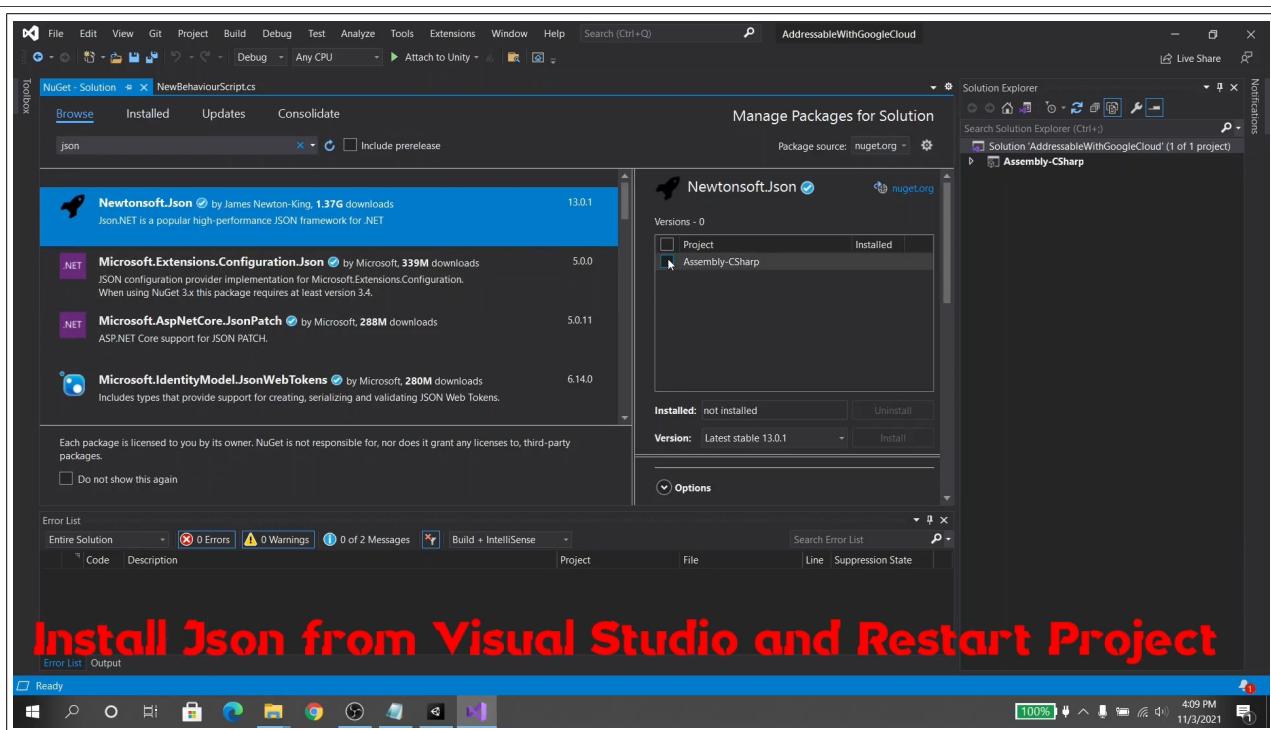
**TakeGoogleCloudAPI/Pakage** (Except Newtonsoft.Json cause Unity has this package even though it's old version)



## 5: Parse all copied dll in to Plugins folder in Unity Project



6: After that unity will show error about Json Version, then open unity project by Visual Studio then Open Nuget and Install newest Json Package.



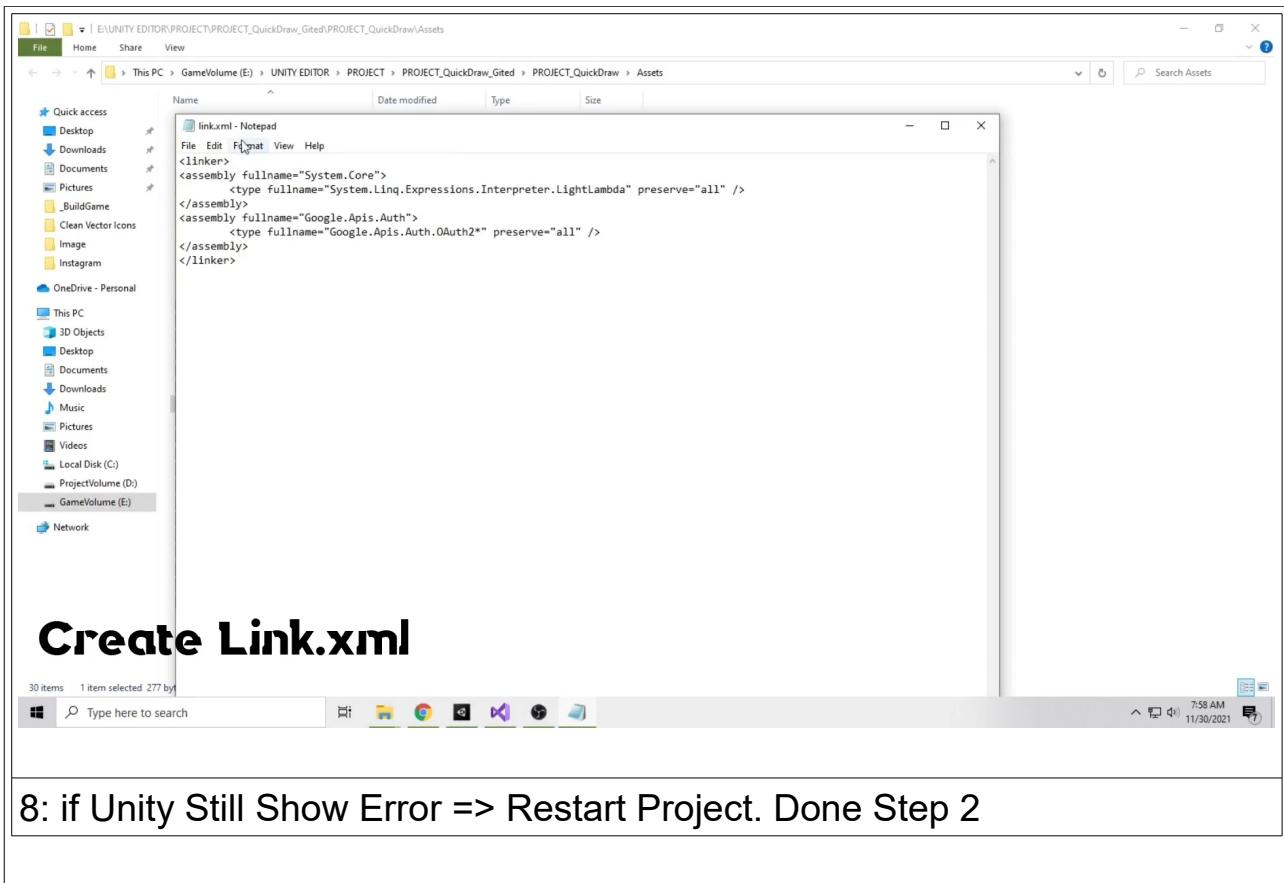
## Install Json from Visual Studio and Restart Project

7: Create a file named link.xml in your Unity project's Assets directory and add the following XML to Assets Folder in Unity project:

```

<linker>
<assembly fullname="System.Core">
    <type fullname="System.Linq.Expressions.Interpreter.LightLambda"
preserve="all" />
</assembly>
<assembly fullname="Google.Apis.Auth">
    <type fullname="Google.Apis.Auth.OAuth2*" preserve="all" />
</assembly>
</linker>

```



### Step 3: Create Google Cloud

1: Create Google Account and Start Google Cloud (If new you has 90 days free trial)

New customers get \$300 in free credits to spend on Google Cloud. All customers get free usage of 20+ products. [See offer details.](#)

Accelerate your transformation with Google Cloud

Build apps faster, make smarter business decisions, and connect people anywhere.

Get started for free Contact sales

Google Cloud Next '21  
It's a wrap!  
Next '21 content now available on demand

Browse catalog

Choose from 100+ on-demand sessions and demos Google Cloud experts.

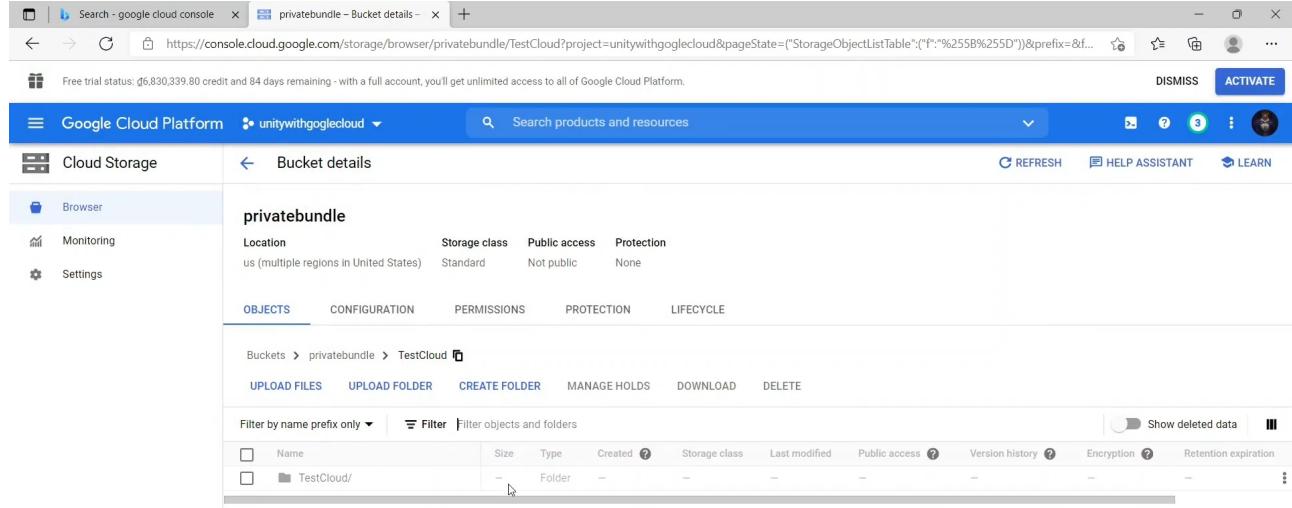
Create Google Account and Start Google Cloud

2: Create Store private Budget for bundle and a folder same name with remote load path on Unity

Exam:

In Google Cloud: Create Budget **PrivateBundle**, and folder **TestCloud** inside.

In Unity Project, set Remote Load Path = **TestCloud**



The screenshot shows the Google Cloud Platform interface for a bucket named "privatebundle". Under the "OBJECTS" tab, there is a single entry for a folder named "TestCloud/". The "REMOTE LOAD PATH" field for this folder is set to "TestCloud".



The screenshot shows the Unity Editor with the "Addressables" window open. In the "Addressables Profiles" section, a profile named "Default" is selected. The "BuildTarget" dropdown is set to "ServerData/[BuildTarget]". The "LocalBuildPath" dropdown is set to "[UnityEngine.AddressableAssets.Addressables.BuildPath]/[BuildTarget]". The "LocalLoadPath" dropdown is set to "[UnityEngine.AddressableAssets.Addressables.RuntimePath]/[BuildTarget]". The "RemoteBuildPath" dropdown is set to "ServerData/[BuildTarget]". The "RemoteLoadPath" dropdown is set to "TestCloud/".

3: Create Store public budget for catalog

Created by XwinStudio

## Exam: In Google Cloud: Create Budget PublicCatalog.

The screenshot shows the Google Cloud Platform Storage browser interface. On the left, there's a sidebar with 'Cloud Storage' selected, followed by 'Browser', '+ CREATE BUCKET', 'DELETE', and 'REFRESH'. A search bar at the top right says 'Search products and resources'. Below the sidebar is a table titled 'Filter Filter buckets' with columns: Name, Created, Location type, Location, Default storage class, Last modified, and Public access. Two buckets are listed: 'privatebundle' and 'publiccatalog'. The 'publiccatalog' bucket is highlighted with a blue border. At the bottom of the browser, there's a note about release notes and a link to 'Create Store public budget for catalog'. The taskbar at the bottom shows various icons and the date/time as 11/3/2021 4:37 PM.

## 4: Create Server, Create Key, DownloadKey

The screenshot shows the Google Cloud Platform IAM & Admin service accounts creation page. The left sidebar has 'API & Services' selected, followed by 'Dashboard', 'Library', 'Credentials' (which is highlighted), 'OAuth consent screen', 'Domain verification', and 'Page usage agreements'. The main area is titled 'Credentials' with options: '+ CREATE CREDENTIALS' and 'DELETE'. A dropdown menu is open under 'Create credentials to access' with 'API key' selected. Other options include 'OAuth client ID' (disabled) and 'Service account'. Below this is a section for 'API Keys' with a 'Name' field and a 'Help me choose' link. To the right, there are sections for 'OAuth 2.0 Client IDs' and 'Service Accounts', both currently empty. A 'CONFIGURE CONSENT SCREEN' button is visible. At the bottom, there's a note about creating Google Certinal and Key. The taskbar at the bottom shows various icons and the date/time as 11/3/2021 4:35 PM.

Search - google cloud console

Create service account - IAM &

https://console.cloud.google.com/iam-admin/serviceaccounts/create?previousPage=%2Fapis%2Fcredentials%3Fproject%3Dunitywithgooglecloud&project=unitywithgooglecloud

DISMISS ACTIVATE

Google Cloud Platform unitywithgooglecloud Search products and resources HIDE INFO PANEL

IAM & Admin Create service account

- IAM
- Identity & Organization
- Policy Troubleshooter
- Policy Analyzer
- Organization Policies
- Service Accounts**
- Workload Identity Federation
- Labels
- Tags
- Settings
- Privacy & Security
- Manage Resources
- Release Notes

Service account details

Grant this service account access to project (optional)

Grant users access to this service account (optional)

Service account users role

Service account admins role

DONE CANCEL

Permissions

Show inherited permissions

Filter Enter property name or value

Role / Principal ↑ Inheritance

Owner (1)

Viewer (4)

Create Google Certinal and Key

Search - google cloud console

testunity - IAM & Admin - unity

https://console.cloud.google.com/iam-admin/serviceaccounts/details/103063647254493507978;edit=true;keys?project=unitywithgooglecloud

DISMISS ACTIVATE

Google Cloud Platform unitywithgooglecloud Search products and resources

IAM & Admin testunity

- IAM
- Identity & Organization
- Policy Troubleshooter
- Policy Analyzer
- Organization Policies
- Service Accounts**
- Workload Identity Federation
- Labels
- Tags
- Settings
- Privacy & Security
- Manage Resources
- Release Notes

DETAILS PERMISSIONS KEYS METRICS LOGS

Keys

Create private key for "testunity"

Add a new key pair or upload a public key.

Block service account key creation. Learn more about setting organization policy.

ADD KEY ▾

Type Status Key Ke

Key type

JSON Recommended

P12 For backward compatibility with code using the P12 format

CANCEL CREATE

Create Google Certinal and Key

## Step 4: Set up Scene Address

1: Create one scene to include in build and othe scene on cloud

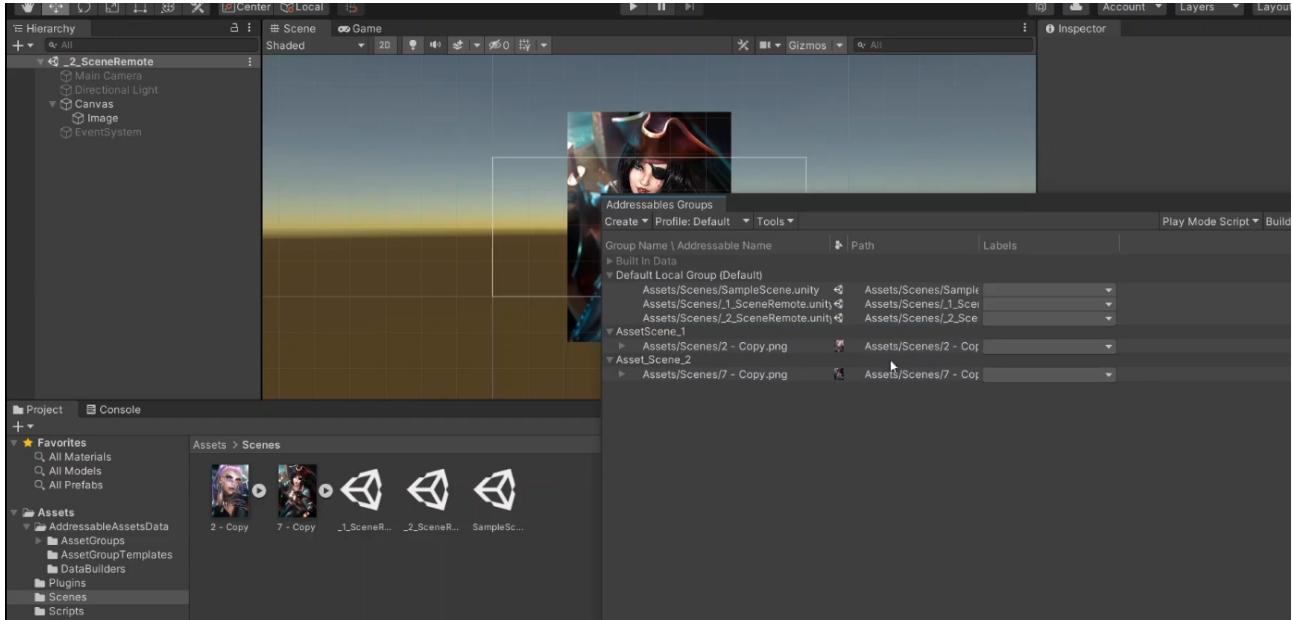
Exam:

Create **SampleScene** to include in build

Created by XwinStudio

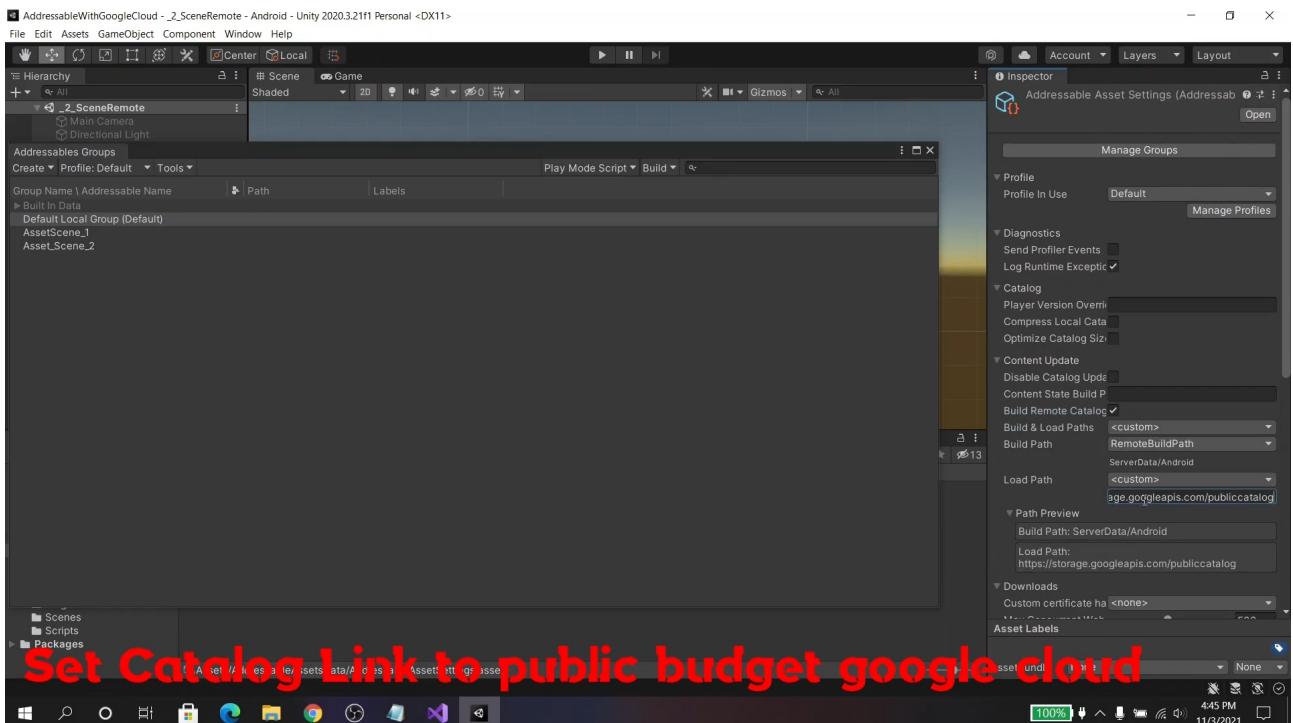
Create **\_1\_SceneRemote** to upload on Cloud, also set up some texture (Exam: **TextureA**) in **\_1\_SceneRemote**.

2: Tick **\_1\_SceneRemote**, **TextureA** is Addressable.



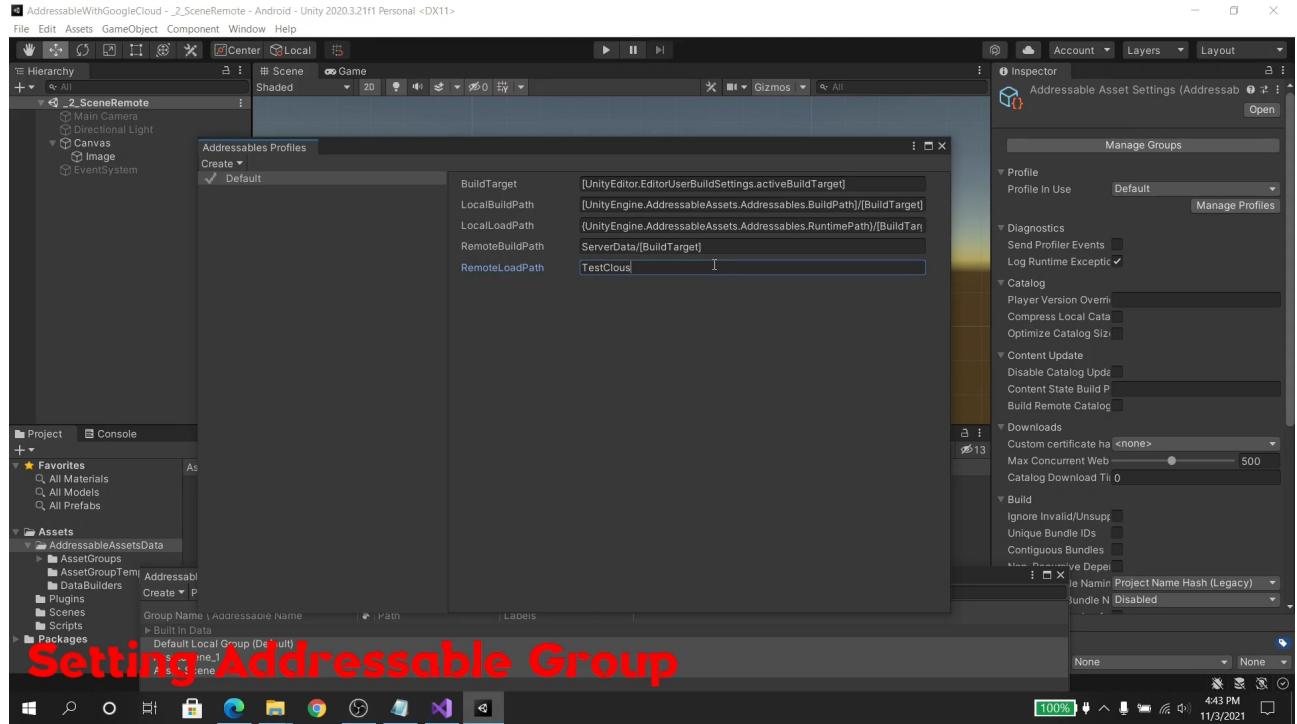
3: On Addressable Group, Add 2 more Group(PackedAsset).

Exam: Add 2 more Group: **AssetScene\_1**, **Asset\_Scene\_2**

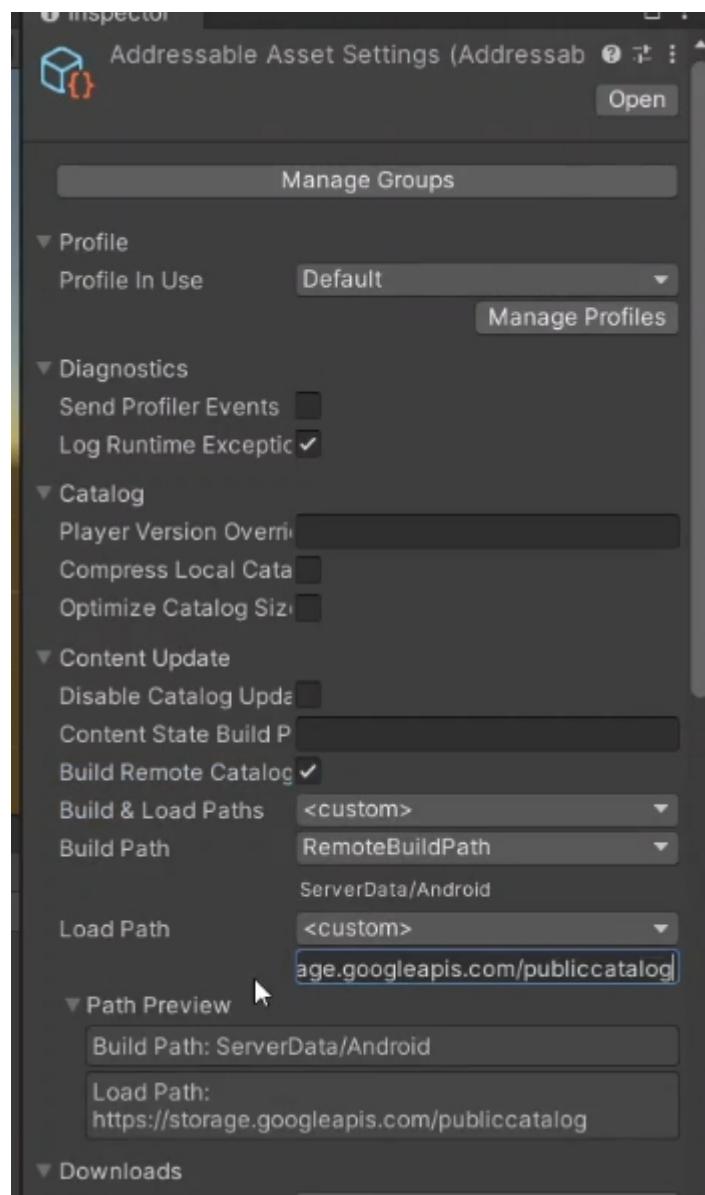


4: Set all group **default**, **AssetScene\_1**, **Asset\_Scene\_2** load path and build path to Remote Load Path and Remove BuildPath

## 5: Set Remote Load Path to "TestCloud" (step3)



6: In Addressable Assets Setting, tick Build Remote Catalog and set Build path = Remote Build Path, Load path = path to you Google Cloud Budget PublicCatalog.



## Step 4: Make Some Script

You need three script:

1: Script to load address scene

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.AddressableAssets;
using UnityEngine.SceneManagement;
using TMPro;
using UnityEngine.ResourceManagement.AsyncOperations;
public class S_LoadAddressable : MonoBehaviour
{
    public AssetReference m_Scene;
    public TextMeshProUGUI m_Text_Alert;
    public TextMeshProUGUI m_Text_Process;
```

```

// Update is called once per frame
void Update()
{
    if(Input.GetKeyDown(KeyCode.F))
    {
        m_Text_Alert.text = "Start Load";
        StartCoroutine(LoadAssets());
    }
}

private IEnumerator LoadAssets()
{
    var isDone = false;
    //Addressable will Load m_Scene
    var download = Addressables.LoadSceneAsync(m_Scene, LoadSceneMode.Additive);

    download.Completed += Func_Complete;
    download.Completed += (operation) =>
    {
        isDone = true;
        m_Text_Process.text = "Done";
    };

    //Show process Download
    while (!isDone)
    {
        m_Text_Process.text = download.PercentComplete.ToString();
        yield return 0f;
    }

    yield return new WaitUntil(() => isDone);
}

//After Download Complete
public void
Func_Complete(AsyncOperationHandle<UnityEngine.ResourceManagement.ResourceProviders.SceneInstance> obj)
{
    if(obj.Status == AsyncOperationStatus.Succeeded)
    {
        m_Text_Alert.text = "Load Scene Succeeded";
    }
}
}

```

## 2: Script to Override link to your private budget

```

using UnityEngine;
using UnityEngine.AddressableAssets;
using UnityEngine.ResourceManagement.ResourceLocations;

public class S_SignURL : MonoBehaviour
{
    public S_GenerateV4SignedReadUrl m_SingURL;
}

```

```

void Start()
{
    //Need To Fix All Bundle path in this func
    Addressables.ResourceManager.InternalIdTransformFunc += TransformFuncContent;
}

string TransformFuncContent(IResourceLocation location)
{
    //Get the url you want to use to point to your current server
    string _OldContentURL = null;// == TestCloud....bundle (begin with Remote Load path
on Addressable Asset Setitng)
    string _currentUrlToUse = null;// == https//..

    if (location.InternalId.StartsWith("TestCloud"))
    {
        _OldContentURL = location.InternalId;
        Debug.Log("Use Signed URL = " + location.InternalId);
        //Sign your Url to Google Cloud
        _currentUrlToUse = m_SingURL.Func_GetSignedURL(location.InternalId);
        Debug.LogWarning("location.InternalId.Replace = " +
location.InternalId.Replace(location.InternalId, _currentUrlToUse));
        return location.InternalId.Replace(location.InternalId, _currentUrlToUse);
    }
    Debug.LogWarning("location.InternalId = " + location.InternalId);
    return location.InternalId;
}

```

### 3: Script get Signed Google Cloud Url

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using Google.Cloud.Storage.V1;
using System;
using System.Net.Http;
using TMPro;
using System.IO;

public class S_GenerateV4SignedReadUrl : MonoBehaviour
{
    //This get signed Url to bundle file on google cloud
    public string m_BucketName;
    [SerializeField]
    private TextAsset _Key = null;
    public string Func_GetSignedURL(string _ObjectPathOnCloud)
    {
        string _SignURL = GenerateV4SignedReadUrl(
            m_BucketName, _ObjectPathOnCloud);
        return _SignURL;
    }

    public string GenerateV4SignedReadUrl(
        string bucketName = "your-unique-bucket-name",
        string objectName = "your-object-name")
    {
        //Cause FromServiceAccountPath need a path to file key but can't access file in
        android
            //=>create new json file from Text asset key
    }
}

```

```
        string _path = Application.persistentDataPath + "/Credenthal.json";
        File.WriteAllText(_path, _Key.ToString());

        UrlSigner urlSigner = UrlSigner.FromServiceAccountPath(_path);
        // V4 is the default signing version.
        string url = urlSigner.Sign(bucketName, objectName, TimeSpan.FromSeconds(50),
HttpMethod.Get);
        //Clean File Credenthal.json
        File.WriteAllText(_path, "");
        return url;
    }
}
```

## *Step5: Buiding Game*

1: Build Assaddressable and Upload to Google Cloud

2: Build Game and Test

For Detail Watch Video

<https://youtu.be/2EfqWDOLh6Q>