

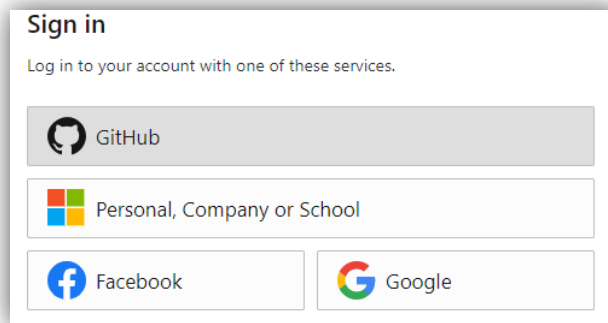
React Native Code Push

STEP-1 [App Center Account Creation]

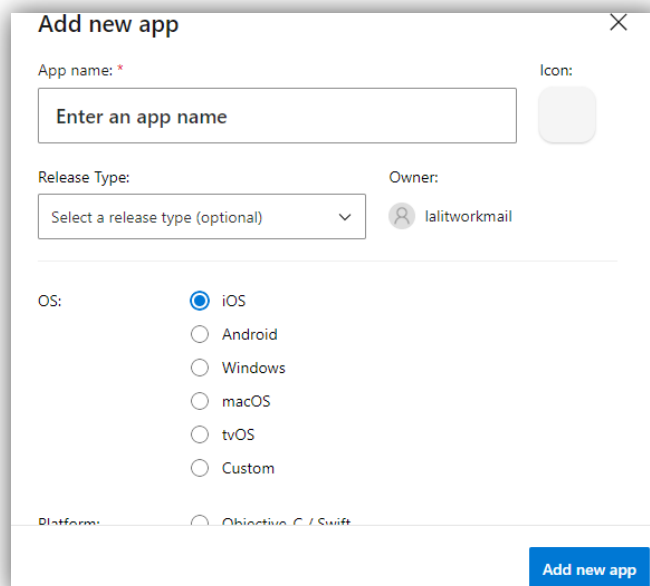
- 1) Open “<https://appcenter.ms/>” this link



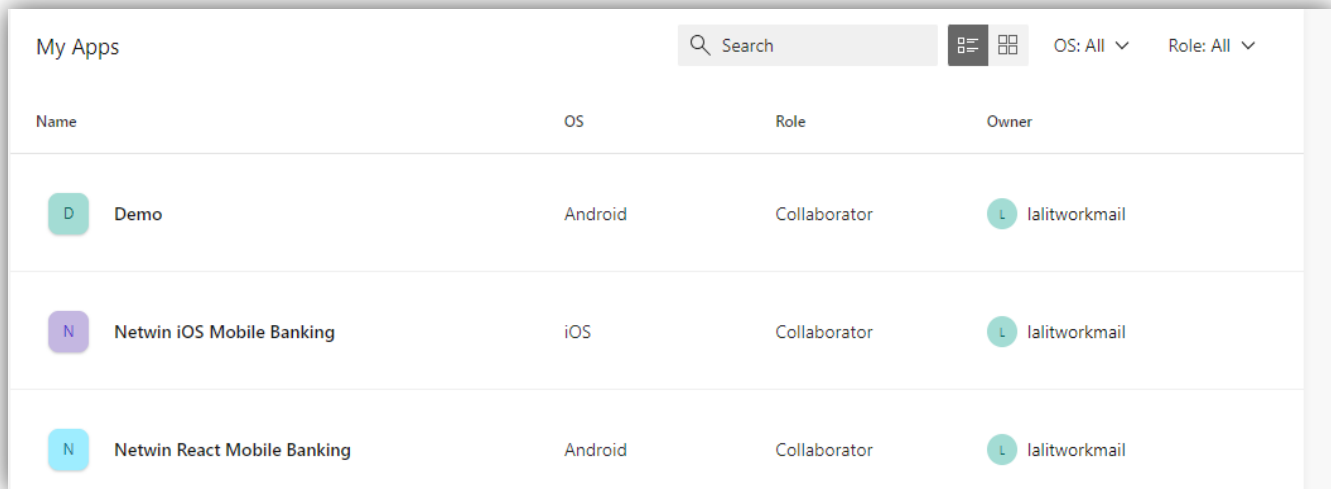
- 2) Log on into it using any of the below option, we used Github for that









- 3) After successful login we get Dashboard from there we have to select Add new Option
- 4) Fill Required Details
- 5) Create Separate App for Both Android & IOS

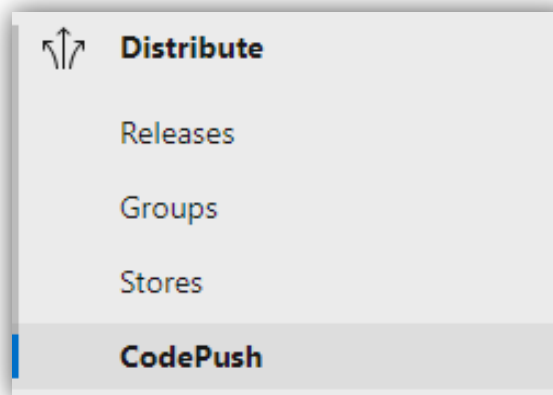


6) Now you can see your Created App listed in dashboard

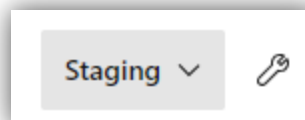


Name	OS	Role	Owner
 Demo	Android	Collaborator	 lalitworkmail
 Netwin iOS Mobile Banking	iOS	Collaborator	 lalitworkmail
 Netwin React Mobile Banking	Android	Collaborator	 lalitworkmail

7) To Generate separate Code Push Key navigate to **CodePush** Tab of **AppCenter**



8) Then click on **Setting Icon** On CodePush Dashboard



9) From there you can create new Deployment Key

Name	Key		
Production	<div>_Ld_FhtaDuQmObM9XJ4CHRNyCjwdOI</div>	<div></div>	<div></div>
Staging	<div>_41A65GzYknLPSUkCgV4sDEIUCRssAv†</div>	<div></div>	<div></div>
Testing	<div>I862qiDPrLTB1jQ1efSd7gfUQ84w2XMxl</div>	<div></div>	<div></div>
<div><div>+</div> New deployment</div>			

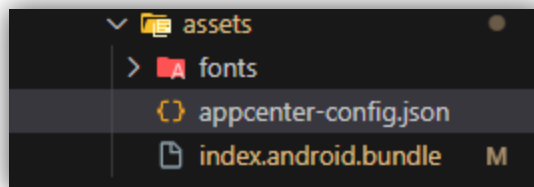
STEP-2 [Android Native Changes]

After that **Overview Tab** will open & there we get Azure Key , & Further process Documentation

- 1) Install below dependency in React app

```
npm install react-native-code-push --f
npm install appcenter appcenter-analytics appcenter-crashes --save-exact
```

- 2) Create a new file with the filename `appcenter-config.json` in `android/app/src/main/assets/` with the following content:



```
{
  "app_secret": "6803654f-102a-4b2e-a98f-4c2519aa6c6a"
}
```

- 3) Modify the app's `res/values/strings.xml` to include the following lines:

```
<string moduleConfig="true" name="CodePushDeploymentKey">DeploymentKey</string>
  <string name="appCenterCrashes_whenToSendCrashes" moduleConfig="true"
translatable="false">DO_NOT_ASK_JAVASCRIPT</string>
  <string name="appCenterAnalytics_whenToEnableAnalytics" moduleConfig="true"
translatable="false">ALWAYS_SEND</string>
```

- 4) Modify the app's `res/Java/packageName/MainApplication.kt` to include the following lines:

```
import com.microsoft.codepush.react.CodePush

override fun getJSBundleFile(): String { return CodePush.getJSBundleFile() }
```

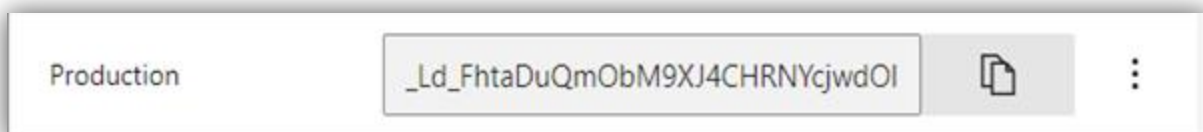
5) Add below lines in **android/settings.gradle**

```
include ':app', ':react-native-code-push'  
project(':react-native-code-push').projectDir = new File(rootProject.projectDir,  
    '../node_modules/react-native-code-push/android/app')
```

6) To Create Separate CodePush for Different ENVs like(we can add our own or rename though) Development , Testing , Debug need to add below lines in **android/app/build.gradle** File .

```
buildTypes {  
    debug {  
        signingConfig signingConfigs.debug  
        proguardFiles getDefaultProguardFile("proguard-android.txt"), "proguard-rules.pro"  
    }  
    release {  
        signingConfig signingConfigs.debug  
        minifyEnabled enableProguardInReleaseBuilds  
        proguardFiles getDefaultProguardFile("proguard-android.txt"), "proguard-rules.pro"  
        resValue "string", "CodePushDeploymentKey", '_Ld_FhtaDuQmObM9XJ4CHRNycjwd0l0x7N8Kq' // key for  
particular env  
    }  
    releaseDevelopment {  
        matchingFallbacks = ['release']  
        signingConfig signingConfigs.debug  
        minifyEnabled enableProguardInReleaseBuilds  
        proguardFiles getDefaultProguardFile("proguard-android.txt"), "proguard-rules.pro"  
        resValue "string", "CodePushDeploymentKey", '_41A65GzYknLPSUKCgV4sDElUCRssAvNhytLS'  
    }  
    releaseTesting {  
        matchingFallbacks = ['release']  
        signingConfig signingConfigs.debug  
        minifyEnabled enableProguardInReleaseBuilds  
        proguardFiles getDefaultProguardFile("proguard-android.txt"), "proguard-rules.pro"  
        resValue "string", "CodePushDeploymentKey", 'l862qiDPPrLTB1jQ1efSd7gfUQ84w2XMxHKp1_'  
    }  
}  
apply from: "../node_modules/react-native-code-push/android/codepush.gradle"
```

7) Key Used in this Generated from AppCenter Portal



STEP-3 [Generation of Private/Public.pem]

- 1) To Generate of .Pem file need have installed “OpenSSL ” in your System [Click here to download](#)
- 2) Then Open bin folder of that Installed software in CMD as **System Administration**

```
C:\Program Files\OpenSSL-Win64>cd bin
```

- 3) Then execute this command to generate Private .pem File “openssl genrsa -out private.pem”

```
C:\Program Files\OpenSSL-Win64\bin>openssl genrsa -out private.pem
```

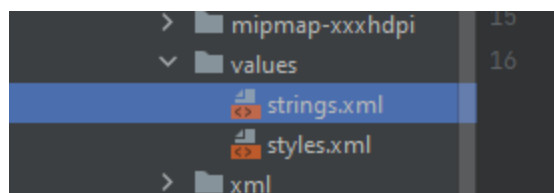
- 4) Then execute this command to generate Public .PemFile “openssl rsa -pubout -in private.pem -out public.pem”

```
C:\Program Files\OpenSSL-Win64\bin>openssl rsa -pubout -in private.pem -out public.pem  
writing RSA key
```

- 5) Here you can see two file were created

p_test.dll	30-01-2024 18:28	Application exten...	39 KB
padlock.dll	30-01-2024 18:28	Application exten...	64 KB
private	24-05-2024 14:36	PEM File	2 KB
progs	30-01-2024 18:28	Perl Source File	8 KB
public	24-05-2024 15:03	PEM File	1 KB
tsget	30-01-2024 18:28	Perl Source File	7 KB

- 6) Now Add Public Key to Apps Strings.xml file [ANDROID]



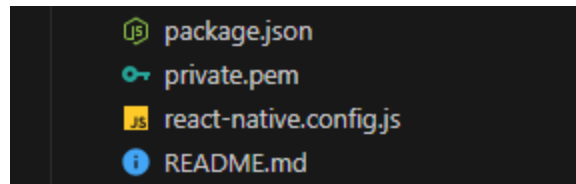
Add a `CodePushPublicKey` string item to

`/path_to_your_app/android/app/src/main/res/values/strings.xml`. It may look like this:

XML Copy

```
<resources>
  <string name="app_name">my_app</string>
  <string name="CodePushPublicKey">-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAtPSR9lkGzZ4FR01xF+ZA
P6jJ8+Xi5L601BPN4QESoRVsrJM08roOCVrs4qoYqYJy30f2cQWvNBEh8ti3FhHu
tiuLFpNdfzM4DjAw0Ti5h0TftixqVBXTJPYpSjDh7K6tUvp9MV015q/Ps3se1vud
M1/X6g54lIX/QoEXTdMgR+SKXvIUIC13T7GkDHT6Z4RlwXkWk0mf2tGguRcEBL6j
ww7w/3g0kWILz7nNPtXyDhIB9WLH7MKSJWdVCZm+cAqabUfpCFo7sHiyHLnUxcVY
OTw3sz9ceaci7z2r8SZdsfjyjIDJrq69ewtvKVUpredy9HtyALtNuLjDITahdh8A
zwIDAQAB
-----END PUBLIC KEY-----</string>
</resources>
```

- 7) Add `Private.pem` file to Root Of Your React Native Project



STEP-4 [ReactNative File Changes]

In App.js File

```
import React, { Component } from 'react';
import { View, ActivityIndicator } from 'react-native';
import Navigation from './src/Navigation';
import CodePush from 'react-native-code-push';

import { _getAsyncStorage, _saveAsyncStorage } from './src/common/util';
import { connect, mapStateToProps, mapDispatchToProps } from './src/App';

var updateDialogOptions = {
  updateTitle: 'Latest App Update Available. Do you want to Install?',
  optionalIgnoreButtonLabel: 'No',
  optionalInstallButtonLabel: 'Yes',
};

const codePushOptions = {
  installMode: CodePush.InstallMode.IMMEDIATE,
  deploymentKey: '_41A65GzYknLPSUkCgV4sDElUCRssAvNhytLS',
  checkFrequency: CodePush.CheckFrequency.ON_APP_START,
  updateDialog: updateDialogOptions,
};

class App_ extends Component {
  constructor(props) {
    super(props);
    this.state = {
      isPackageDownloading: false,
      isUpdateReady: false,
      appUptoDate: false,
    };
  }

  componentDidMount() {
    this.checkForUpdate();
  }

  checkForUpdate = async () => {
    this.setState({ isPackageDownloading: true });

    try {
      await CodePush.checkForUpdate(codePushOptions?.deploymentKey).then((update) => {
        if (!update) {
          console.log("The app is up to date!");
          this.setState({ isPackageDownloading: false, isUpdateReady: false, appUptoDate: true });
        } else {
          CodePush.sync(
            {},

```



```

status => {

  switch (status) {
    case CodePush.SyncStatus.UP_TO_DATE:
      this.setState({ appUptoDate: true });
      break;
    case CodePush.SyncStatus.DOWNLOADING_PACKAGE:
      this.setState({ isPackageDownloading: true });
      break;
    case CodePush.SyncStatus.INSTALLING_UPDATE:
      break;
    case CodePush.SyncStatus.UPDATE_INSTALLED:
      this.setState({ isPackageDownloading: false, isUpdateReady: true });
      break;
  }
},
// Progress callback
progress => {
  // Update your loading indicator based on progress.percent
}
);
}
})
} catch (error) {
  // Handle deployment not found error
  this.setState({ isPackageDownloading: false });
  this.setState({ deploymentNotFound: true });
}
};

render() {
  const { isPackageDownloading, isUpdateReady, deploymentNotFound, appUptoDate } =
this.state;
  if (isPackageDownloading) {
    // Show a loading indicator while the package is downloading
    return (
      <View style={{ flex: 1, justifyContent: 'center', alignItems: 'center' }}>

        <ActivityIndicator size="large" color="#0000ff" />
      </View>
    );
  } else if (isUpdateReady) {
    // If an update is ready, display a message to restart the app
    return (
      <View style={{ flex: 1, justifyContent: 'center', alignItems: 'center' }}>
        <Text>Update is ready, please restart the app.</Text>
      </View>
    );
  } else if (appUptoDate) {
    {console.log('continue to app', appUptoDate)}
  }
}

```

```
    // If no update is available or downloading, render the main app content
    return <Navigation />;
  }
}

export default connect(
  mapStateToProps,
  mapDispatchToProps,
)(CodePush(codePushOptions)(App_));
```

STEP-5 [STEPS TO FOLLOW WHILE CODE PUSH] [Click Here For Refi](#)

1) Update deployment Key of your Project/Bank

```
const codePushOptions = {
  installMode: CodePush.InstallMode.IMMEDIATE,
  deploymentKey: '_41A65GzYknLPSUkCgV4sDElUCRssAvNhytLS',
  checkFrequency: CodePush.CheckFrequency.ON_APP_START,
  updateDialog: updateDialogOptions,
};
```

2) Then Execute Code Push Code

```
npx appcenter codepush release-react -a netwinAddon/Netwin-React-Mobile-Banking -d Staging -k private.pem -m --description "DEPOSIT SCREEN CHANGES BY DEVELOPER"
```

3) If Above code get Error Then Need to Login App Center Again From VSCode

```
npx appcenter login
```

4) After executing above code It will redirect to App Center Portal and you get Authentication code



5) Then Enter the Authentication code

```
PS C:\NetwinNew\mb_git> npx appcenter login
Opening your browser...
? Visit https://appcenter.ms/cli-login?hostname=Lalit and enter the code:
? Access code from browser: 745ae91a504c47e0d7b9078d33bcdcd710f3f2da
Logged in as netwinAddon
```

Command Syntax BreakDown

```
npx appcenter codepush release-react -a <appName> -d <deploymentName> [options]
```

Parameters and Options

- **-a, --app <appName>**
 - Description: Specifies the application name in the format `ownerName/appName`.
 - Example: `netwinAddon/Netwin-React-Mobile-Banking`
 - Required: Yes
- **-d, --deployment-name <deploymentName>**
 - Description: Specifies the deployment where the update should be released. Common deployment names are `Staging` and `Production`.
 - Example: `Staging`
 - Required: Yes
- **-k, --private-key-path <privateKeyPath>**
 - Description: Specifies the path to a private key file used to sign the update. This is optional and used for enhanced security.
 - Example: `private.pem`
 - Required: No
- **-m, --mandatory**
 - Description: Marks the release as mandatory. If an update is mandatory, users must install it.
 - Required: No
 - Flag: This is a boolean flag. When included, it sets the release as mandatory.
- **--description <description>**
 - Description: Provides a description of the release. This is helpful for keeping track of changes in each release.
 - Example: `"Testing the codepush update for xyz bugfix"`
 - Required: No

Full Example Command

```
npx appcenter codepush release-react -a netwinAddon/Netwin-React-Mobile-Banking -d Staging -k private.pem -m --description "Hemant banking homescreen recharge slider"
```

Detailed Breakdown

- **npx appcenter codepush release-react:** This is the base command used to release a React Native update via CodePush.

- **-a netwinAddon/Netwin-React-Mobile-Banking**: Specifies the application name. Here, **netwinAddon** is the owner and **Netwin-React-Mobile-Banking** is the app name.
- **-d Staging**: Specifies that the release should be deployed to the **Staging** deployment.
- **-k private.pem**: Uses **private.pem** as the private key file for signing the update.
- **-m**: Marks this release as mandatory.
- **--description "Banking home screen changes for recharge slider"**: Provides a description for this release, detailing the changes or new features included.

Get Code push Code Directly From App Center

You can get Release code directly from Appcenter/Project/Distribute/CodePush Option

No updates released yet for Production.

Release an update to Production with the command:

```
$ appcenter codepush release-react -a netwinAddon/Netwin-React-Mobile-Banking -d Production
```

Refi Link :- [“https://medium.com/@ritika100898/implementing-codepush-for-over-the-air-updates-in-react-native-535e750ec308”](https://medium.com/@ritika100898/implementing-codepush-for-over-the-air-updates-in-react-native-535e750ec308)

PemKey Refi Link :- [“https://learn.microsoft.com/en-us/appcenter/distribution/codepush/rn-get-started”](https://learn.microsoft.com/en-us/appcenter/distribution/codepush/rn-get-started)