- https://github.com/lisong/code-push-demo-app
- https://reactnative.dev/docs/environment-setup
- https://docs.microsoft.com/en-us/appcenter/distribution/codepush/
- https://docs.microsoft.com/en-us/appcenter/distribution/codepush/rn-get-started#android-setup
- https://docs.microsoft.com/en-us/appcenter/distribution/codepush/cli#code-signing
- Tutorials consulted and followed a lot: https://www.youtube.com/watch?v=jEEPlqIcICM&list=PLucG_ap4OxzhOGhcO4pJxm-ojSV5LMHWi&index=1
- https://appcenter.ms
- https://www.giacomocerquone.com/blog/microsoft-codepush-integration-in-react-native-0.60
- https://github.com/cmcaboy/react-native-code-push

Github link to my repository(The code for Android application only):
https://github.com/froheel/react-native-codepush-example

Simple Steps to follow:
1) Make sure the react native project is properly working.
2) Ensure you have the correct react native version installed that is supported by codepush
3) Open the official documentation of react native with code push :https://docs.microsoft.com/en-us/appcenter/distribution/codepush/
4) Create a visual studio app center account. I linked it with my github.
5) Download the app center cli using shell/cmd/powershell.
6) Login using shell/cmd/powershell with your app center credentials and verify that you are logged in.
7) Create an application (ios and android separate) for your react native project. I created only the android named Myapp.
8) .Follow the official documentation of codePush for react native and follow the steps according to the versions you currently have installed for codePush and react-native.CodePushify your application! Make sure you make 2 folders of your react native app - one for android and one for ios and codePushify them separately.
9) Release a build with a key (I used the production key for my android app) on Appcenter. Add this key in the react-native project too, wherever required.
10) Make sure you only make changes to the JS,html,css etc. Do not make major changes in your application. Consult the documentation for the components that are supported by codePush and which changes are reflected.
11) Update your application and save the changes using codepush. I used codePush to update the Production app with your <username>/<applicationname>: appcenter codepush release-react -a froheel/Myapp -d Production
12) Once this is done you can now go and see in the app that the codePush version has been updated. You did not need to deploy the build again. You just made changes using codepush and it automatically reflected in the build. You can use a group of

people to which you can distribute your app or optionally connect with appstore/playstore. TA-DA ! You are now done. Congratulations!!!

- All these details are provided based on my experience only. Your best guide would be the official documentation of codePush. Please follow that. I too have followed it too.
  https://docs.microsoft.com/en-us/appcenter/distribution/codepush/
- I found this playlist to be very useful:
  https://www.youtube.com/watch?v=jEEPlqIcICM&list=PLucG_ap4OxzhOGhcO4pJxm-ojSV5LMHWi&index=1