**Introduction to Salesforce DX**

It can be defined as a new way and accompanying a set of tools, bringing to Salesforce Developers a better and latest development flow. It is a product for the App cloud environment to enhance productivity levels with better control and improved collaboration.

As deployment has always been a difficult part of Salesforce development, Salesforce DX brings you the best and efficient tooling from planning to release phase. It enables organizations to grab digital business opportunities and developers to deliver top-quality applications rapidly.

To briefly summarize my thoughts on what exactly Salesforce DX is, it is important to remember that it is fundamentally a mindset and a philosophy.

A new and advanced philosophy, SFDX includes a series of customized tools and features that enable source-driven development and a new level of custom app development.

**What Salesforce DX provides to its users?**

Salesforce developers are amazing as they think and act practically. Generally, they aren’t just interested in complicated solutions. So, let me tell you what exciting SFDX brings out for you!

* **Version Control System:** Since Salesforce DX stepped in the market, one phrase has been spread widely- “Version control is the source of truth”. You might not understand what I meant to say. It means developers need to maintain the entire history of code changes to work and collaborate effectively. It encourages developers to back up their code changes, keep version history, and manage changes over time.
* **Use of Scratch orgs:** DX introduces the thesis of scratch orgs. It enables you to use a new type of org known as scratch org that is required for setup. Scratch orgs are temporary Salesforce orgs, built from source, used for development and testing, and destroyed when done.

You can use them as a temporary environment to make changes, whenever needed by the developers. It is good because instead of using sandbox with different functionalities on it (which may be difficult to keep track of), DX provides you a feature of having different scratch orgs to work upon a different piece of work independently.

* **Continuous Integration (CI) & Continuous Delivery (CD):** DX possesses the capacity to improve the quality and it makes sure to release new modification to the customers quickly, in a sustainable way
* **Change Management:**It provides developers with more and deeper data insights to change the production and manage it easily.

In a nutshell, using version control on whole code and metadata, you can work in a well-organized way in the development and testing phase. As well as you don’t have to go through the same process of setting up and syncing sandboxes anymore.

Very nice Salesforce! Really, very…. nice!!

In all the above aspects, Salesforce DX will help you. However, just like other CRM, DX also comes up with advantages and disadvantages that we will discuss further.

**Advantages of Salesforce DX**

* It helps improve team collaboration and development.
* It makes the release cycle process more agile and efficient.
* It allows the developers to use any tool to modify the code such as CLI, VIM, Sublime, Atom, etc.
* Facilitates automated testing for your code and enables continuous integration.
* Requires a local development setup for the developers to get hands-on expertise on DX.

**Disadvantages of Salesforce DX**

* As DX is new, some features are still missing and yet to come so users may confront some errors.
* Troubleshooting seems to be a difficult task for system admins as it needs heavy line commands and instructions.
* The learning curve for developers is extremely high to get familiar with the customized platform.

# **Manage Scratch Orgs from Dev Hub**

* <https://help.salesforce.com/articleView?id=sfdx_dev_scratch_orgs_view_lex.htm&type=5>

# **Link a Namespace to a Dev Hub Org**

* <https://help.salesforce.com/articleView?id=sfdx_dev_reg_namespace.htm&type=5>

# **Salesforce DX (Trailheads)**

* <https://trailhead.salesforce.com/content/learn/projects/quick-start-salesforce-dx?trailmix_creator_id=strailhead&trailmix_slug=build-your-developer-career-on-salesforce>
* <https://trailhead.salesforce.com/content/learn/modules/sfdx_app_dev>

**Version Control**

* <https://code.visualstudio.com/docs/editor/versioncontrol>

**Required tools Steps & Link: -**

# Step 1:-Enable Dev Hub in Your Org

<https://help.salesforce.com/articleView?id=sfdx_setup_enable_devhub.htm&type=5>

Step 2:- Install salesforce CLI

<https://developer.salesforce.com/docs/atlas.en-us.sfdx_setup.meta/sfdx_setup/sfdx_setup_install_cli.htm#sfdx_setup_install_cli_windows>

Step 3 :- Install Vs code

<https://trailhead.salesforce.com/en/content/learn/projects/quick-start-lightning-web-components/set-up-visual-studio-code>

Install Salesforce Extension on Vs Code: -

-Salesforce Extension Pack

-Force Code

Step 4 :- Create GitHub account

<https://github.com/>

If already you have account, you can Sign in or you need to create a new account.

**Authenticate with the Dev Hub:**

Step 1: Enable Dev Hub in an org

* Log in as System Administrator to your Developer Edition, trial, or production org (if you’re a customer), or your business org (if you’re an ISV).
* From Setup, enter Dev Hub in the Quick Find box and select **Dev Hub**
* To enable Dev Hub, click **Enable**.

Step 2: Log in to the Dev Hub

* To authorize the Dev Hub, in the command window enter the web login flow. The following command opens the Salesforce login page in the web browser:

sfdx force:auth:web:login -d -a < alias >

* Log in using your Dev Hub org credentials.
* Click **Allow**

**Create a New Scratch Org**

* In a command window, navigate to where you want your project located.
* Create the project:

sfdx force:project:create -n <ProjectName>

* Navigate to Project Directory

Cd <ProjectName>

* Next, use the following command to create the scratch org

sfdx force:org:create -s -f config/project-scratch-def.json -a <username/alias>

**Open the Scratch Org**

In a command window, use the following command to open the scratch org :

sfdx force:org:open -u <username/alias>

**Clone the current code repository to local and push it into scratch org :**

* In Command Prompt window , use the following command to create a unique directory to do your work in

mkdir < directory name>

* Navigate to that Directory

Cd < directory name>

* Next, use this command to clone the app repository:
* Sign In to git Account
* Next open the repository and click on Clone button
* Copy the web URL

git clone < web URL>

* Next, open the directory

Cd <git directory name>

* Create a new branch to do your work in

git checkout <branch name>

* Open the Scratch org

sfdx force:org:open -u <username/alias>

* Push all the local source into the scratch org with this command

sfdx force:source:push -u <username/alias>

* Assign your permission set

sfdx force:user:permset:assign -n <app name> -u <username/alias>

* Load your sample data into the org:

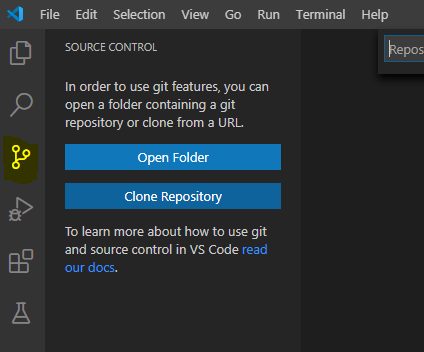
sfdx force:data:tree:import --plan data/sample-data-plan.json

* Open your org:

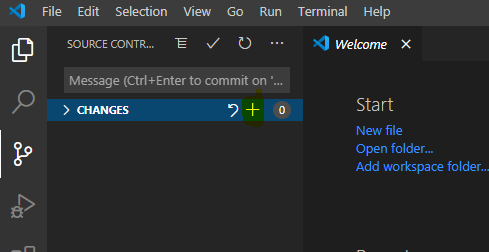
sfdx force:org:open -u <username/alias>

**Push changes from local to GitHub :-**

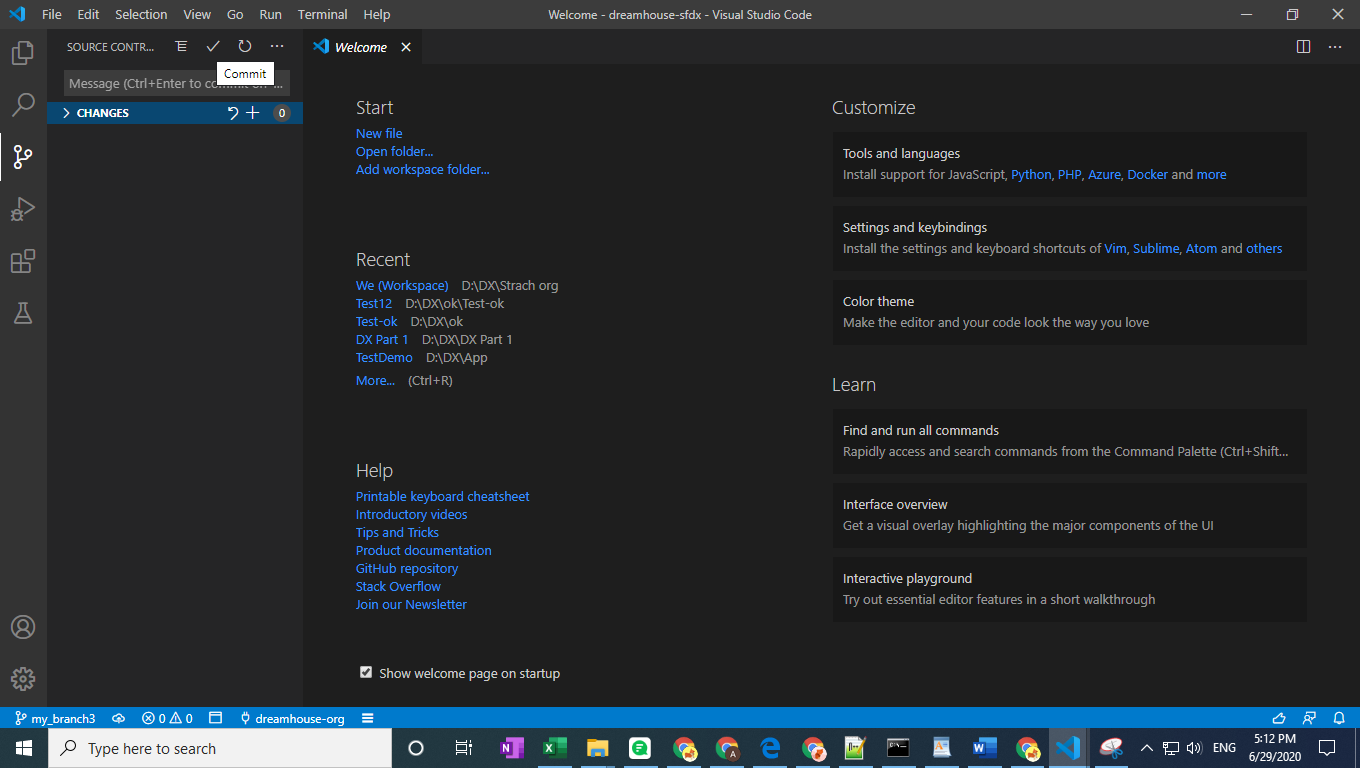
* 1. Click on Open folder, in which changes are done from starch org.
  2. Folder will open, GitHub option, you will see the changes Option.



* 1. In changes Option, click to + option all the changes will come & in message you need to right the text message, it will file name. You can see in the GitHub Repository.



* 1. User need to click on Commit option “Right”



* 1. Refresh the GitHub account in chrome, you will see the metadata is push to GitHub account.

**Make your desired changes: -**

* For E.g. :- We have create a Lightning component we can Push to starch org .
* Please check the below link for steps: -
* <https://trailhead.salesforce.com/en/content/learn/projects/quick-start-lightning-web-components/create-a-hello-world-lightning-web-component>