https://developer.salesforce.com/docs/atlas.en-[us.sfdx\_dev.meta/sfdx\_dev/](https://developer.salesforce.com/docs/atlas.en-us.sfdx_dev.meta/sfdx_dev/sfdx_dev_scratch_orgs_def_file.htm)[sfdx\_dev\_scratch\_orgs\_def\_file](https://developer.salesforce.com/docs/atlas.en-us.sfdx_dev.meta/sfdx_dev/sfdx_dev_scratch_orgs_def_file.htm)[.htm](https://developer.salesforce.com/docs/atlas.en-us.sfdx_dev.meta/sfdx_dev/sfdx_dev_scratch_orgs_def_file.htm)

[SFDX and How to Convert Package code Using SFDX | Jayakrishna Ganjikunta (wordpress.com)](https://jayakrishnasfdc.wordpress.com/2020/08/15/sfdx-and-how-to-convert-package-code-using-sfdx/)

Ref : <https://wilsonmar.github.io/salesforce-dx/>

<https://www.jitendrazaa.com/blog/salesforce/getting-started-with-salesforce-dx-salesforce-developer-experience/>

[Increase Productivity with Integrated Tools Unit | Salesforce](https://trailhead.salesforce.com/content/learn/modules/sfdx_dev_model/sfdx_dev_model_integrated_tools)

//Deployment Script

sh $sfdx force:source:deploy -x manifest/package.xml -u ${UName\_INT}

Agenda

* Introduction to Salesforce DX
* Creating Scratch Org
* Deploying metadata to scratch org
* Creating Skeleton Workspace
* Running Test classes
* Getting Help
* Using Force.com IDE with Salesforce DX
* Q&A

What is Salesforce DX

Why do we need it?

* New tool to manage deployment Lifecycle
* SCM Driven
* Package based development
* Scratch Org
* CLI & Force.com Support
* Its to move metadata and manage the org.

In Large Enterprise project major challenges.

Maintaining Sandbox

Refreshing Sandbox

Deploying code from sandbox to sandbox or sandbox to production

If you are working on some AppExchange product then that’s all is different challenge.

We can deploy changes using Changeset.;

* But Changeset can be used only in connected environment.
* Once deployment done, Information is lossed until you do manually noting down that this changeset contains this metadata.
* Development using changesets to move code can be troubling for team working on same set of configurations because there is no going back.

To overcome connected environment issue there is another tool i.e. ANT Migration tool.

* Using Ant metadata, we can deploy anywhere to anywhere no dependency on connected environment.
* We always have meta data which can be stored in some source code management like Git/SVN or somewhere else.
* Problem with ANT to get profile you have to get all components.
* To sync 2 sandboxes, 2 sync different orgs.
* You have to understand how ANT works, Ant migration tool kit is nothing but ANT plugin written in Java.
* Need to create package.xml

throughout the software development industry today, there is a movement toward storing **configuration as code**, of keeping metadata out from inside data in the org and into **versioned** code bases separate from the data.

The new “source of truth” for source-driven development is in the VCS (Version Control System) rather than in the production org. This means the configuration of the org exists outside the org. So experimental orgs can be easily created and tested without fear of disruption.

So to overcome these challenges Salesforce came with New tool i.e.

New approach of storing metadata in repository enable time travel control of versions means that all files present at each commit point in time can be restored so now you

**Salesforce DX (Salesforce Developer Experience):**

* Its new concept known as Developer Hub and scratch org.
* Its command Line tool.
* Available from eclipse directly.
* Its source code management driven, which means your metadata will always stay either in GIT, SVN or any SCM.
* Deployment of Salesforce DX is always package driven.
* Does not need any package.xml

Use Case :

Suppose we have team of 40 developers working on enterprise org.

There are 4 teams with 10 developers each.

Its not always that 4 teams will be working on different-different Apex classes as chances of overlap will always be high.

Generally, what we do,

* Create 1 SB for each user or 1 SB for each team
* Teams do their development do its deployment and then there is process to merge codes and other stuffs.
* This works for most of the client as they have defined processes in place.

But SFDX says that

* You must develop in a concept of package.
* Develop one feature and deploy that whole feature as one package.
* Do not mix 2 or 3 development in one package or xml or do the backfilling and see if code merge is happening or not.
* Code merge problem could be here also but we will be deploying on the basis of packages.

Issues with Sandboxes:

* Creation of Sandboxes takes time which could be from 1 hour and sometimes 1 week also for big orgs, depending upon which kind of SB you are trying to create and actual data in SF instance.

You can also create Developer org but that we have to do manually.

Scratch Org:

A **scratch org** is a dedicated, configurable, and short-term Salesforce environments that are quickly spun up within a Dev Hub Org. when starting a new project, a new feature branch, or a feature test.

* It can be created just on one click automatically without any manual intervention.
* All would be created within one minute, so saving in time.
* You can have CI process in place where scratch org will be created nightly, you do your development there or do your POC and destroy the org, once you are able to see that this is how things are working.
* SFDX comes in 2 version
  + SFDX CLI
  + SFDX Force.com
* Scratch orgs are meant for use by individual developers rather than a team sandbox.
* HOW? SFDX keeps track of both changes you make locally as well as any in your scratch org.
* SFDX transforms large source files into smaller files to provide more project flexibility and reduce merge conflicts.
* Any changes made within a scratch org (using point-and-click) needs to be tracked in the Git source to be repeatable.

**Developer Hub Org:**

The Developer Hub org is used to authorize all Salesforce DX users. The Developer Hub (Dev Hub) org is a central location for Salesforce DX because it allows you to create, delete and manage your Salesforce scratch orgs.

Below command will set org as default **Developer Hub Org**and will set its alias as my-devhub-org. It will open Salesforce login page in default browser for OAuth flow, where we need to login to Developer Hub org and authorize Salesforce DX.

sfdx force:auth:web:login --setdefaultdevhubusername --setalias my-devhub

### ****What are scratch Orgs ?****

Salesforce DX can be enabled for any Salesforce instance and they are known as **Developer Hub**. One Developer Hub can have multiple Scratch Orgs. **Scratch Orgs are temporary Salesforce org which can be created quickly and metadata can be deployed from SCM.**These kind of Orgs can be used by developers to perform quick proof of concept or build and test packages. Once package is build and saved back on SCM, scratch org can be destroyed easily.

**Clone Git Repository:**

git clone <https://github.com/forcedotcom/sfdx-simple>

cd sfdx-simple

**Creating Scratch Org**

sfdx force:org:create --setdefaultusername -f config/workspace-scratch-def.json --setalias jitendra2\_scratch

What happens in Background:

Its nodejs application