Agenda – Day 6

01 UI Development and Improvements

02 Side effects and defaults

03 Git and Git HUB

04 Introduction to HANA Cloud

05 HDI Containers and Deployment

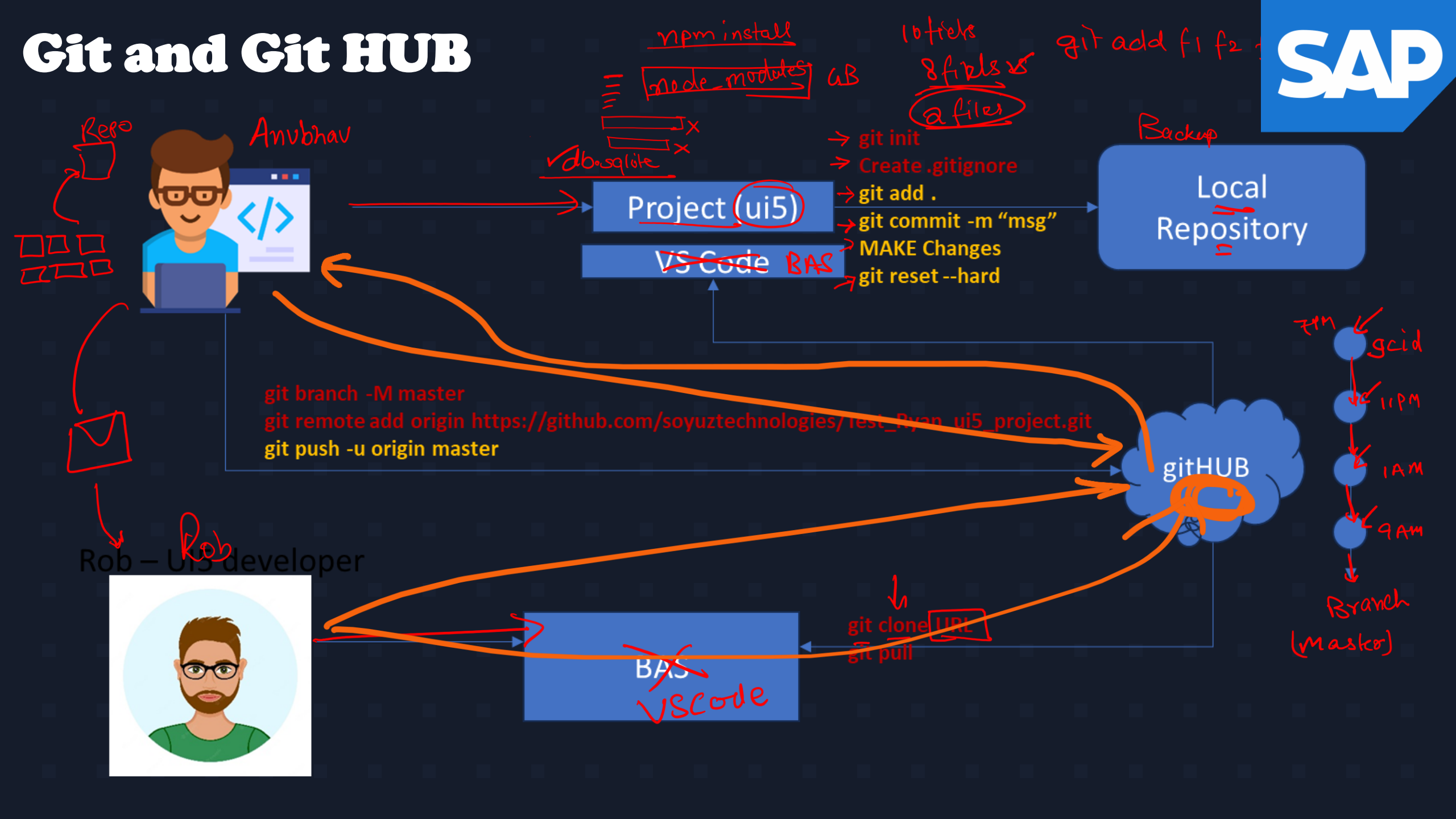
06 Adding App Router

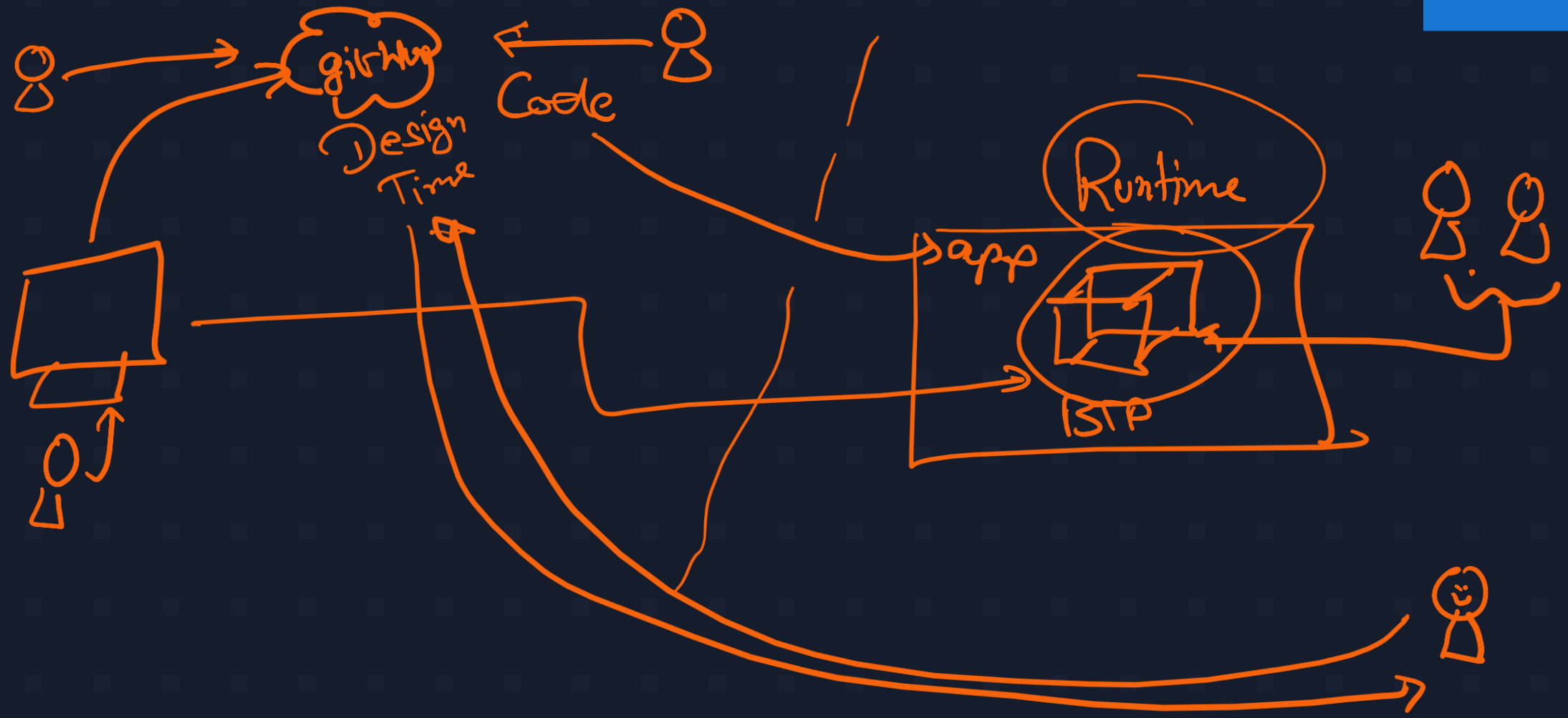
Challenges to share code

- ✓ Till now Anubhav is sharing zip file of project code with all of you, imagine in a team of 40 developers and at end of day if every developer share a zip file, who can merge all code.
- ✓ When someone go on holiday or left team, how can we make sure that we have latest code available for successor to manage it.
- ✓ Anubhav last evening my code was stable, this morning I came to office and made 40-50 changes which has disrupted everything, I pray to god to go back to last night state. Basically, we need version management.
- ✓ In a team, one developer using BAS another using VS code another using WebIDE, how can we easily move code from one dev tool to another
- ✓ When something cause production issues, we want to know which change was made by who last time that caused the issue.
- ✓ How to share our code effectively between multiple team members as best practice from industry. And how to merge conflicts also.

Git and Github
Git is a de-central repository management system for managing any kind of source code. And git hub is a central repository where everyone sync their changes (internet).
https://git-scm.com/download/win

Git and Git HUB




Code
Design
Time
Runtime
sapp
ISTP

Introduction to HANA Cloud

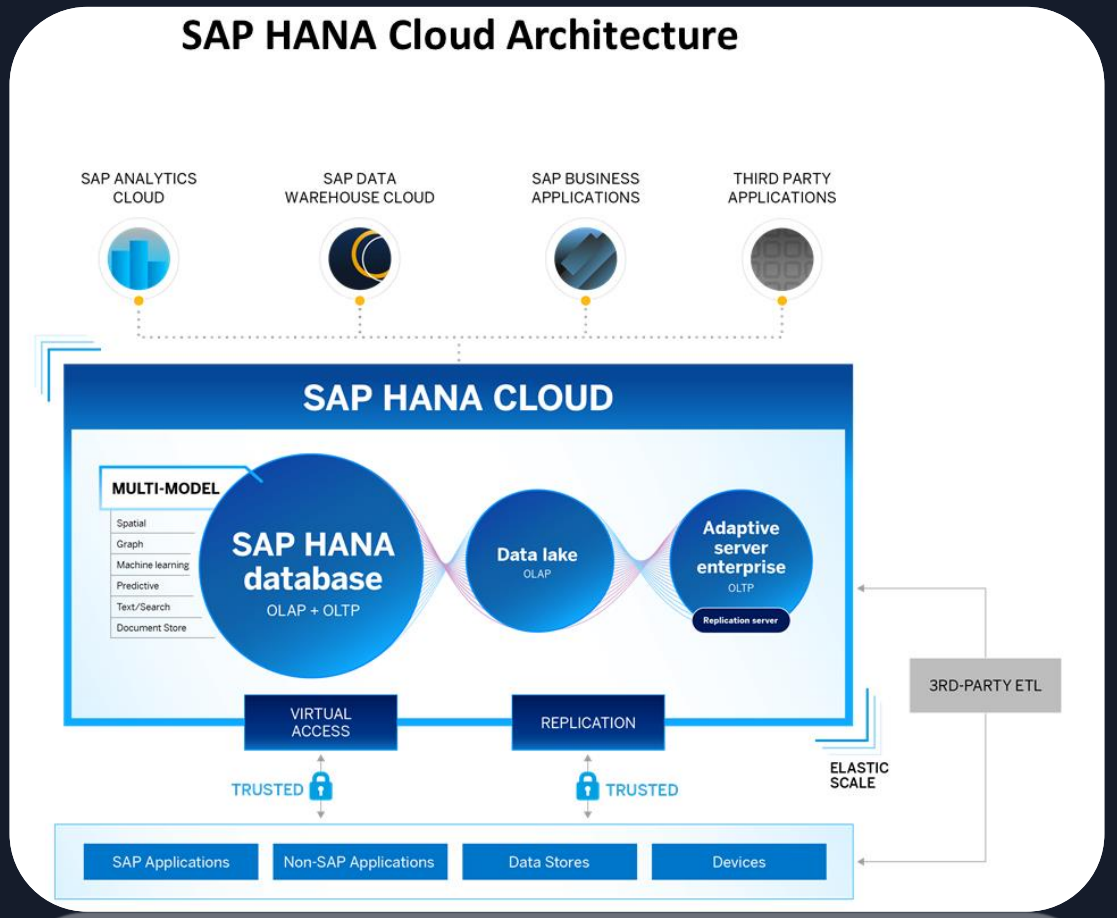
SAP HANA Cloud is a fully managed, in-memory, cloud database as a service (DBaaS). It is the cloud-based data foundation for SAP Business Technology Platform. With SAP HANA Cloud you can create, run, and extend new and existing applications.

SAP HANA Cloud includes a number of software components. The core component is SAP HANA Database, but other components can be added at any time, such as a data lake.

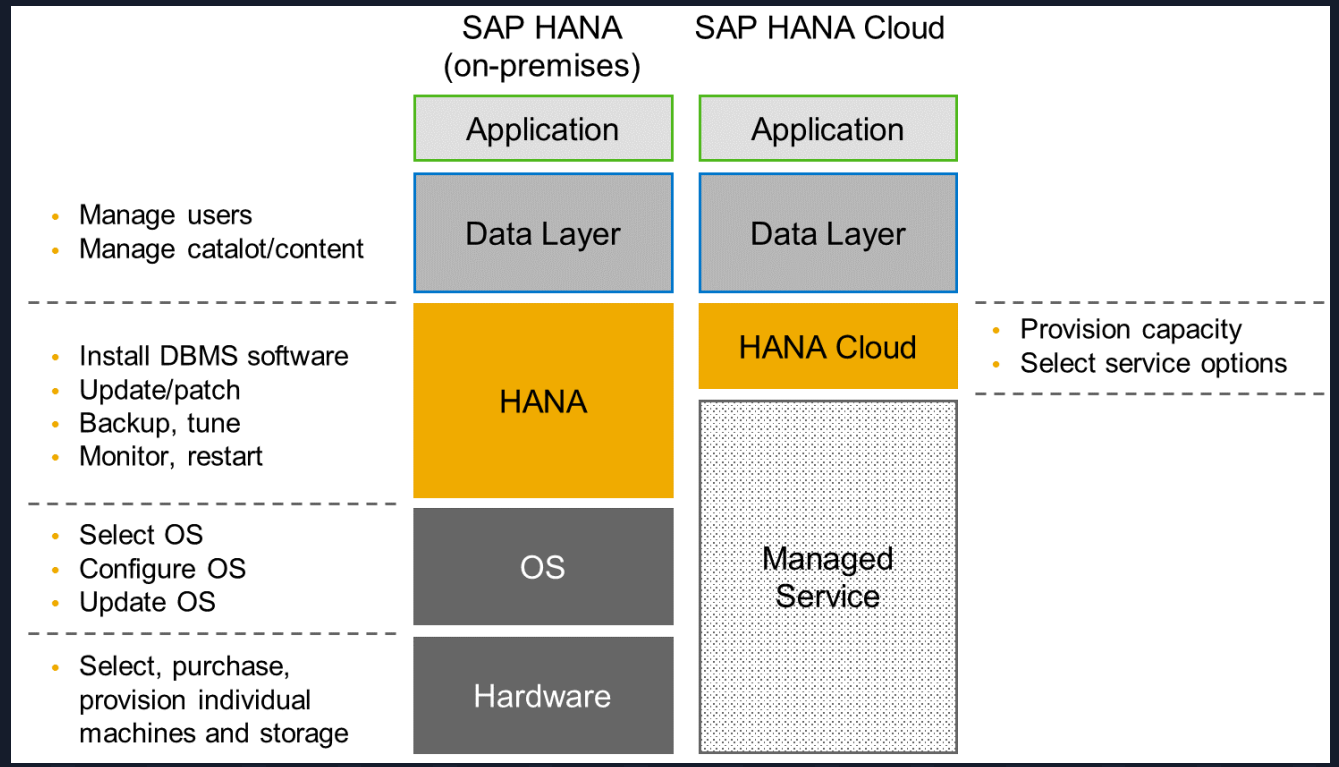
Key Components of SAP HANA Cloud

There are four key components of SAP HANA Cloud:

- **SAP HANA Cloud, SAP HANA Database**
  In-memory database with built-in advanced analytics (spatial, graph, text, etc.)
- **SAP HANA Cloud, data lake**
  Store and query large data sets and most file types
- **SAP HANA Cloud, adaptive server enterprise**
  Support for extreme –performance transactional applications
- **SAP HANA Cloud, adaptive server enterprise replication**
  Bi-directional real time data replication across databases



HANA Cloud v/s On-premise



Feature Differences

SAP HANA Cloud compares very closely with SAP HANA on-premise in terms of features but there are some differences.

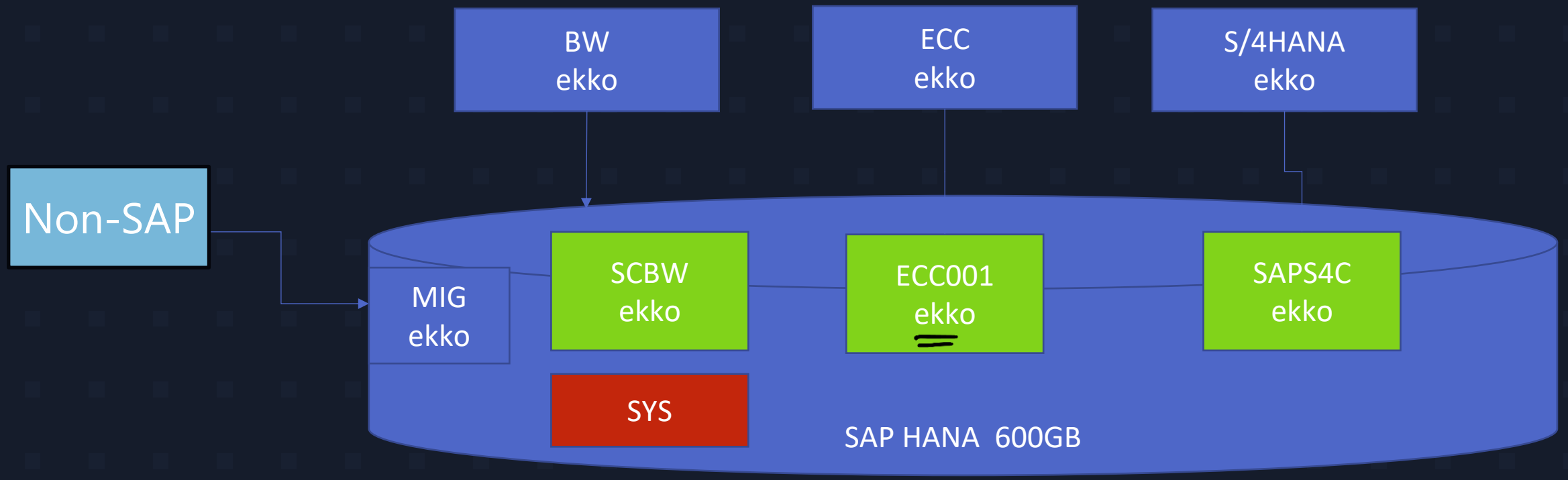
Feature	SAP HANA Cloud	SAP HANA on-premise
Row / column in-memory database	✓	✓
Data modeling	✓ *	✓
Data Replication	✓	✓
Data Virtualization	✓	✓
Data Cleansing (SDQ)	✗	✓
Text Analysis and Mining	✗	✓
XS Classic and XS Advanced	✗	✓
Multi-tenancy	✗	✓
SAP Data Warehousing Foundation	✗	✓
SAP Streaming Analytics	✗	✓
Data Lake	✓	✗

* scope of modeling includes:
- Calculation views
- Graph
- Text Search
- Predictive
- Spatial

What is Schema in HANA

Schema – Is a mandatory database object of database which stores other database objects.

- It's a logical separation of database objects.
- It is home of all the runtime object
- Security



HDI Container

An HDI Container is also a **schema**. But it is managed differently from a normal schema.

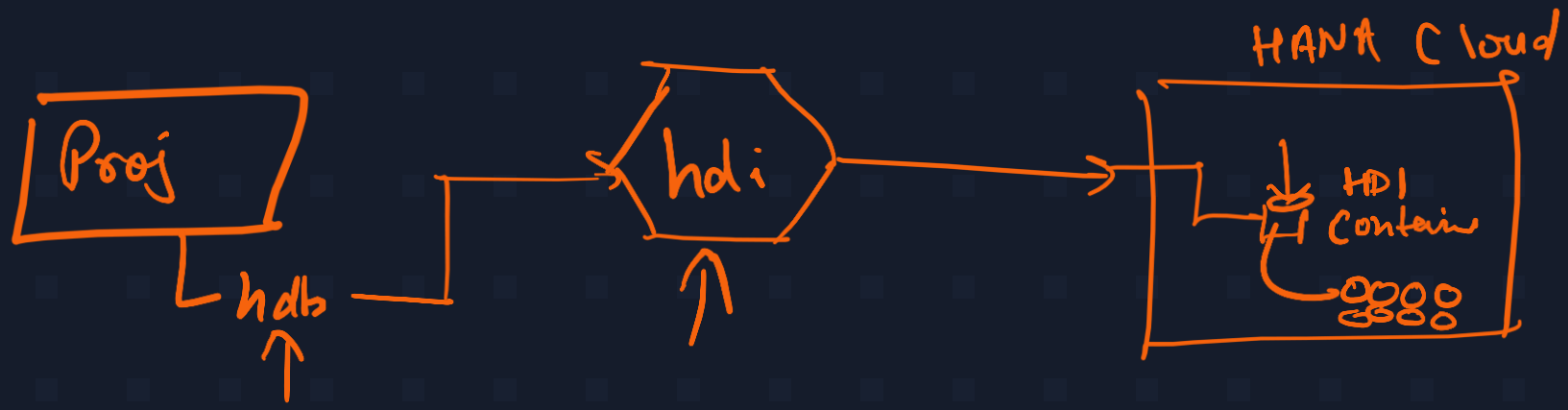
Schema -- How its created?

1. Manually create it in BTP
2. Application Managed HDI Container (You as developer ONLY work with App, App will talk to HANA to create and manage this schema to store/read data internally)

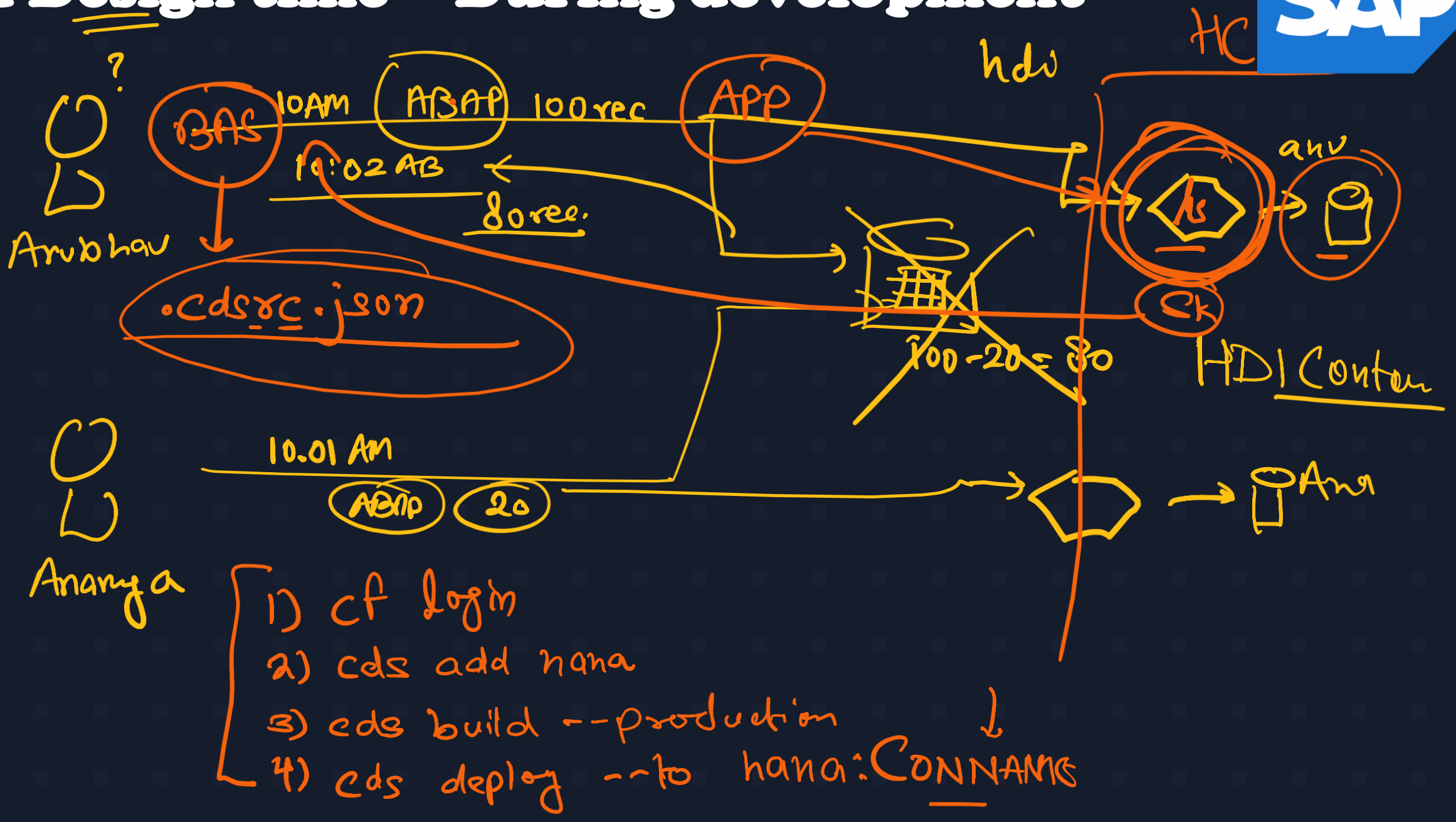
You cant access HDI Container using HANA Studio (Deprecated tool), We can use new SAP HANA Database Explorer, SAP HANA WebIDE for OP or BAS to access HDI Container.

HDI -- Who creates it?

There is a new service in SAP HANA called SAP HDI (HANA Deployer Infrastructure) which is just a microservice responsible to managing the container. HDI Deployer is a component which is used to talk to this service from our app.

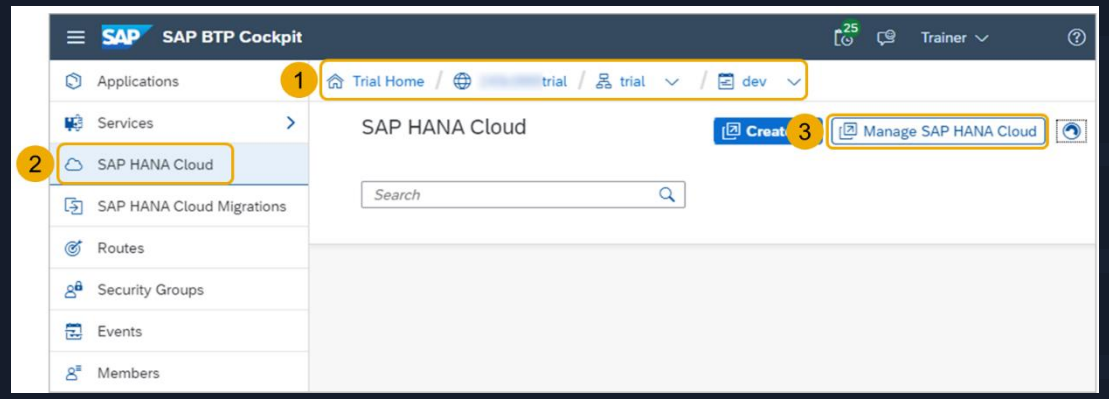


HDI Design time – During development

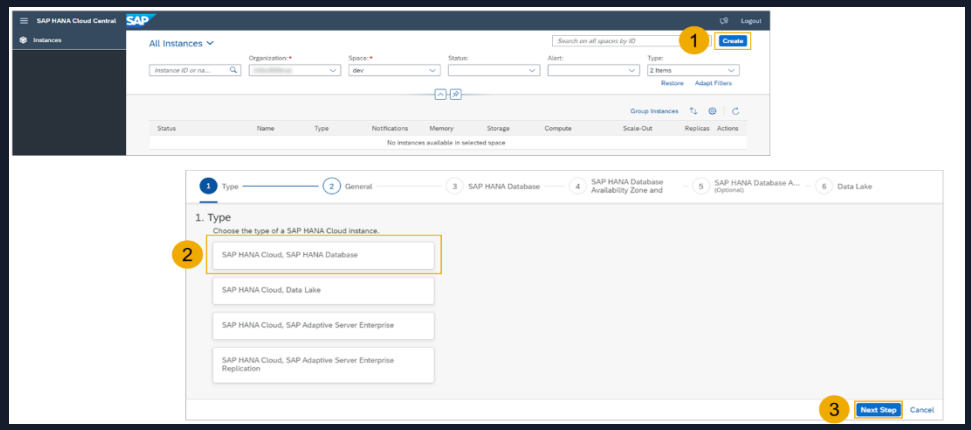


Hands on - Create HANA Cloud Instance

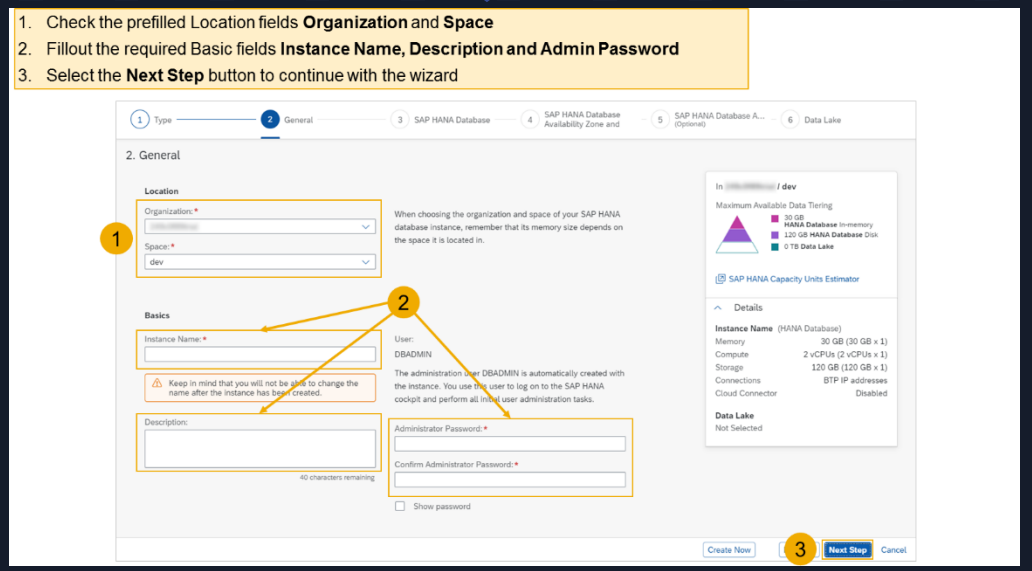
1. Login to your trial and navigate to your **dev** space
2. Select **SAP HANA Cloud**
3. Select the **Manage SAP HANA Cloud** button to start the SAP HANA Cloud Central



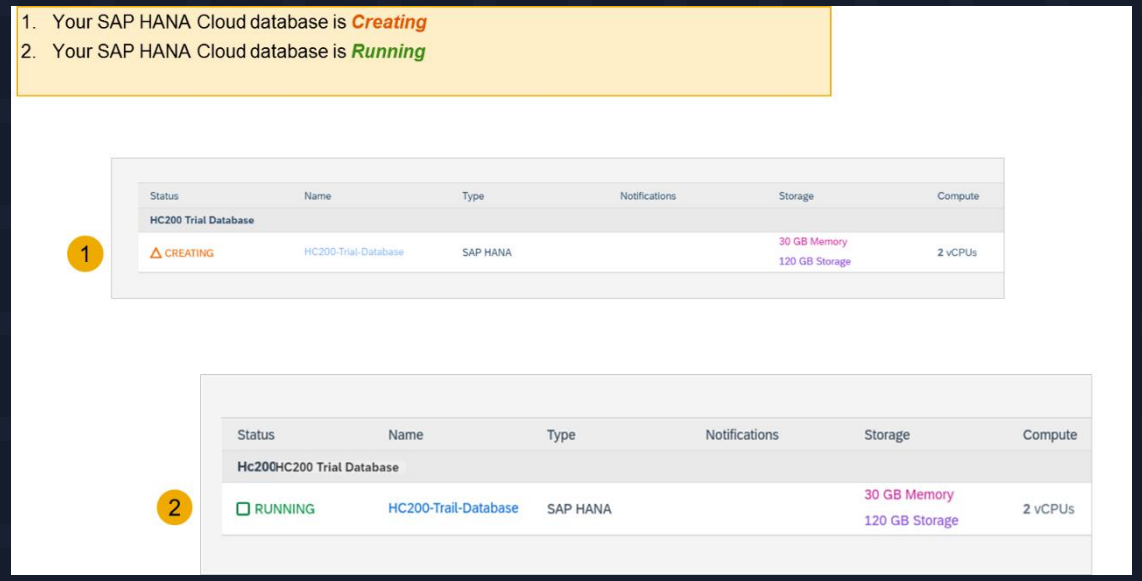
1. Select the **Create** button to start the wizard
2. Select the SAP HANA Cloud instance Type to create
3. Select the **Next Step** button to continue



1. Check the prefilled Location fields **Organization** and **Space**
2. Fillout the required Basic fields **Instance Name, Description and Admin Password**
3. Select the **Next Step** button to continue with the wizard



1. Your SAP HANA Cloud database is ***Creating***
2. Your SAP HANA Cloud database is ***Running***



Hands on – HANA Cloud Integration

Pre-requisites

1. Check the version of CDS (which must be > 2.0) **cds –v**
2. Start a HANA cloud instance in SAP BTP

Main steps

1. Add hana configuration to our app, which tell cap framework that the default db is now hana – **cds add hana**
2. Add the hana specific deployment format name to our project under cds section as below inside package.json file

```
"hana":{
      "deploy-format":"hdbtable"
    }
```

3. Since the first step added the hdb node module, we need to install it. Run **npm install**
4. We need to perform a build, which will create all the hana specific files which will be deployed to HANA cloud. **cds build --production**
5. We need to login to cloud foundry from BAS – **cf login**
6. Finally we need to deploy the DB and everything to SAP HANA Cloud – **cds deploy --to hana:batman**
7. We fixed issues related to excel format of HANA and changed the size of field bankid for employee, redo the build and deploy
8. If deployment worked, a new file **cdsrc-private.json** gets created automatically, this file contain the information about which container in SAP BTP HANA Cloud to connect to. And the private key is stored in this file.
9. Provide the credentials which will be used to connect database, start using **cds watch --profile hybrid**

```
"credentials": {
          "database": "batman-key"
        }
```