SAP ECCà HANAàCloud

ECC:-

RICEFW –

* Report(Custom Output not covered by standard) Ex:- Sales Report with Custom Field – Classic ALV/Interactive ALV/Web-dynpro/Module pool
* Interface(IDoc/ALE, SOAP, REST, RFC, TCP) for integrating SAP to SAP and Non-sap systems
* Conversion(One time migration program to convert legacy data into sap internal data) – LSMW/BDC
* Enhancement:- Enhance Standard logic to meet custom requirements
* Form:- Customer invoice/tax invoice – Sap script/smartforms/adope forms
* Workflow:- For automated approval / rejection of any request/flow

HANA:-

* Moving to Column base table, advance compression of data, in-memory database, support both OLAP/OLTP, code push down techniques
* HANA can be used as database under ECC as application later known as “Suite on HANA”
* If HANA itself is application layer as well database its called S4HANA can be used as standalone outside SAP in custom apps or for analytical reporting
* ECC can run on any db like HANA, oracle, SQL, DB2, etc but S4HANA runs on only HANA database.

Cloud:-

* BTP is main tool to manage everything right from integration, admin, user maintenance, roles, custom app, deployment, launpad,etc
* Business can have public cloud only or private cloud with on-premises or hybrid.
* Ex:- You’re a start up company and want to grow in sap cloud. You have less resource and infrastructure at your end. You let SAP manage it for you. This is public cloud, where you use standard app or extension on public cloud(Multi-tenant) managed by SAP.
* Ex:- You’re a big company already in SAP for long. Currently you have on-premise ecc /HANA system. But now you want to migrate to cloud but not in one go. So, first company buys public cloud connect with on-premises system. Keep critical finance/employee salary details on on-premises. Other things like employee leave, employee on-boarding you push it to public cloud so everything happens online. Critical data entered online will be push to on-premise and not stored on public cloud. Slowly, you add private cloud version where infrastructure is still on sap side but tenant is specific to customer. Customer has control over cloud system.

You have a kitchen where you use to cook and run your business. You still want to cook your dishes which are your speciality in market, as it requires special vessel and recipe. You still cook it in your kitchen(on-premise). But now you want to keep fast food like pizza/sandwich to be ordered online on swiggy/Zomato from other kitchen(Public cloud). You might want to serve your special pasta from your kitchen(on-premise) along with garlic bread from swiggy(public cloud).

IDOC/ALE/SOAP/REST/RFC

* IDOC:- Asynchronous data transfer between sap-sap and non-sap system, specially for large volume like orders, master data,etc.
* ALE(Application Link Enabling) When you want to distribute data from 1 system to multiple sap system. List of material sits in one sap system which should be distributed to various different system based on location. It sits on Idoc with distribution profile.
* SOAP:- Synchronous or asyn, specifically for heavy payload. It only supports xml which is heavier to process
* REST/ODATA:- Mostly synchronous(Can be asyn using batch), uses json and light weight xml, specifically for Mobile / UI apps
* RFC:- SAP to SAP Function call , not recommended for non-sap to sap

BAS vs Eclipse

* BAS(Business Application Studio) is Cloud IDE used for front-end development or side-by-side development like CAP based ODATA service or Front-end UI development.
* For ABAP object like RAP or Backend CDS, we still need ABAP IDE like Eclipse/HANA Studio to develop ABAP Object.
* So, to create RAP object like CDS, BDEF, BI, SDEF, SB à Eclipse. Create dev space in BTP sub-account and then create abap environment instance to generate Service key to be used in ECLIPSE to develop RAP.
* Now, RAP runs on abap environment, it cannot be directly deployed on Cloud foundry or kyma. But any Fiori or CAP running on CF or Kyma can consume this RAP exposed odata. These Fiori app can be used by user using BTP launchpad service.

CF / KYMA / Neo:-

* Neo:- Its legacy runtime environment before CF for legacy app. Less flexible and slow. Will be obsolete in few years.
* CF: More flexible and used for most of Fiori app
* Kyma:- more advance as it works on kubernete theory(Containerized). Different micro-service are deployed inside Containers and consume together as one app. This require Kubernete knowledge. Advanced and more flexible.

RAP VS CAP

* RESTful Application programming:- Runs on ABAP Environment, specially when your moving from Legacy SAP environment toward Cloud where you have mostly ABAP resource in your project. More connected with custom logic/ sap process. Eclipse for development
* CAP(Cloud Application Programming) : Developed on java/nodejs, when you have multiple different system apart from sap where you have team with knowledge on front-end languages and you want to create more advanced Front-end application. Loose engagement with sap process.

BAS for development

BTP à Cloud Connector à On-premises

1. Download SCC from HANA-Ondemand site
2. Login with default ID password. You can access it via https://localhost:8443
3. Create sub-account on BTP
4. Add that sub-account in Cloud Connector under Cloud to Cloud option.
5. Once added, you will see cloud connector in BTP Sub-accountàUnder connectivityàCloud Connector
6. Add on-premise system in CC under Cloud->On-premise option. Keep host, port, virtual host which BTP should see, etc.
7. Once back-end system added and connected in CCà define access control like which object should be exposed ex:- ODATA, RFC, etc
8. Create Destination in BTP using Virtual host/port defined in CC, authentication details. Then check connection in same screen.
9. When RAP or backend ODATA is consumed in Fiori app deployed on CF/Kyma, ODATA will be called via this destination.

BADI >> Explicit Enhancement >> Implicit Enhancement >> Customer Exit >> Modification directly in standard code

* BADI:- Most preferrable option as of date for extensibility because of OO, Multi implementation and upgrade safe
* Explicit Enhancement:- Second best
* Implicit Enhancement:- Only at specific locations like start/end of FM, include, Methodsà Upgrade safe but can end up with mixing of logic
* Customer Exità Available in ECC and single implementation. Not available in S/4
* Modification: worst case scenario

AMDP Procedure VS Function

Both are implemented as Class – Method. Difference it with use and its consumption

* Procedure:- It has Importing and exporting parameters. Specifically for Multi-step functionality. Like first you want to check if sales order exist or not. If exist then sum all quantity and net value. Also get line item details. All these we can do in single amdp procedure method. But this cannot be consumed in CDS directly as it can have multiple table in output. It can be called via Class-method in report.
* Function: - Importing and single return parameter of type Table. When you need only item detail for particular sales order, then use functions as it can be directly consumed in CDS.