Steps to build End-to-End CAPM application with CI/CD setup in SAP.

Tuesday, December 23, 2025 4:58 PM

Prerequisites

1. *A Build Code Subscription with required permissions.*
2. *Cloud foundry runtime with enough space to deploy application.*
3. *A HANA Cloud Subscription with required permissions.*

Initial Process.

1. Open Build code application, click on **create** button. In Objective, select Application card then click on next.
2. Select Full-Stack in Category, click on next. Now select Full-Stack Type Node.JS, then click on next.
3. Enter Project name and description(optional) in Create project details section. Under dev space selection, create or select the existing dev space (here is where the project is created).
4. Click review, and click on create.
5. Once, project is created, click on it. You will redirected to new window where your project is created (Build Code BAS)

Create schema and service cds files in ./db and ./srv folders.

1. Inside the "db/" folder, create a file schema.cds (Which used to define the schema of CAP application, In simple it is known as Domain modeling)

```
namespace uvs;
type OrderStatus : String enum {
    Pending;
    Shipped;
    Delivered;
    Canceled
}
entity Orders {
    key ID : String @title : 'Order ID';
    customerID : String @title : 'Customer ID';
    orderDate : DateTime @title : 'Order Date';
    totalAmount : Decimal(10, 2) @title : 'Total Amount';
    status : OrderStatus;
    items : Composition of many OrderItems on items.order = $self
}
entity OrderItems {
    key ID : String @title : 'Item ID';
    name : String @title : 'Item Name';
    quantity : Integer @title : 'Quantity of Items';
    price : Decimal(10, 2) @title : 'Item Price';
    netPrice : Decimal(10, 2) @title : 'Net Price';
    order : Association to Orders;
}
```

2. Now, Expose the created entities defined in schema.cds file as a service in service.cds file.
3. Here, we can also expose the external services like:
    1. Northwind OData service
    2. SAP Business Accelerator Hub
    3. S4H System (On-premise)
4. Here, we are consuming Northwind OData service. Configured in package.json file.
5. "@" using this we can define the annotations in CAP, Makes the application business wise semantic and conventional.

```
using {uvs} from '../db/schema';
using {northwind} from './external/northwind';
service OrderServices @(
    requires: 'authenticated-user',
    path    : '/OrderManagermentServices'
)
{
    @odata.draft.enabled
    @restrict: [ { grant: 'READ', to: 'Viewer' },
                 { grant: '*', to: 'Admin' }
               ]
    entity Orders     as
        projection on uvs.Orders {
            *,
            customer : Association to Customers
                           on customer.CustomerID = $self.customerID
        };
```

6. You can explore about annotation used at https://cap.cloud.sap/docs/

```
    };

    entity OrderItems as projection on uvs.OrderItems;
    @odata.draft.enabled
    entity ZCustomers as projection on Customers;
    //External Service Exposing
    entity Customers  as
        projection on northwind.Customers {
            CustomerID,
            CompanyName,
            ContactName,
            ContactTitle,
            Address,
            City,
            Region,
            PostalCode,
            Country,
            Phone,
            Fax,
        };
}
```

Package.json

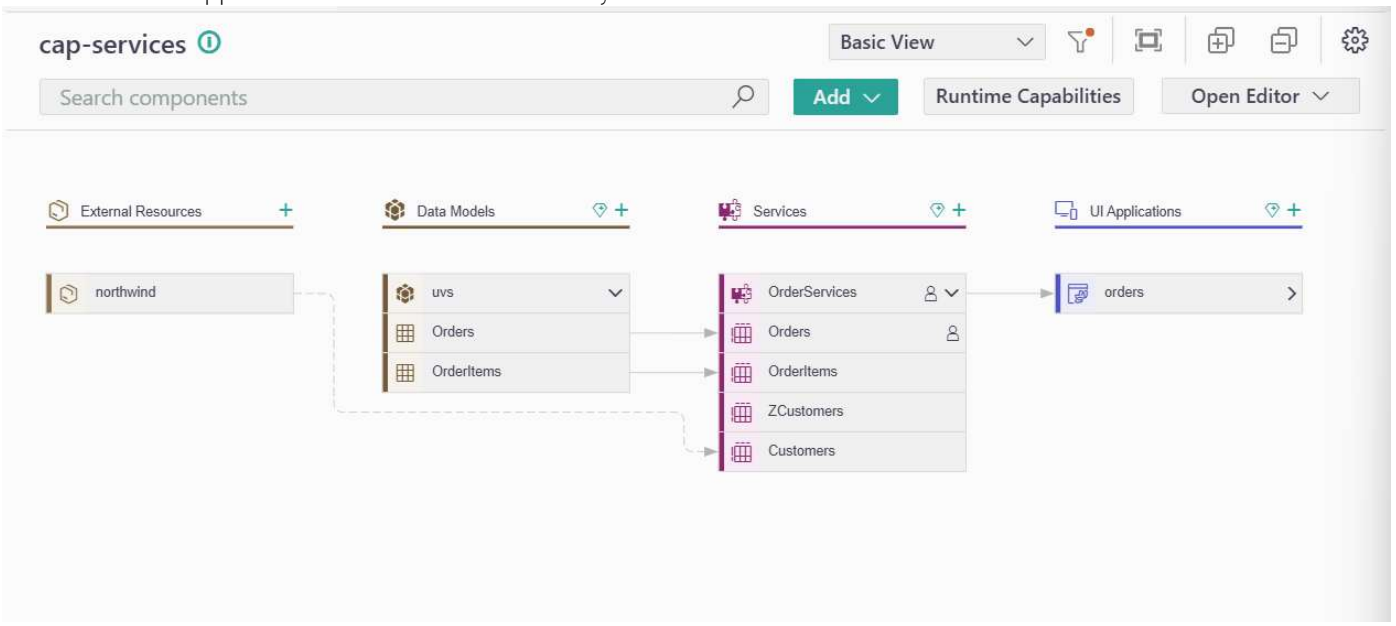
1. Project dependencies
2. CDS run time requirements
    a. Like database, default sqlite in-memory
    b. External services, like northwind service in my case.

```
{
  "name": "cap-services",
  "version": "1.0.0",
  "description": "A simple CAP project.",
  "dependencies": {
    "@sap-cloud-sdk/connectivity": "^4",
    "@sap-cloud-sdk/http-client": "^4",
    "@sap-cloud-sdk/resilience": "^4",
    "@sap/approuter": "^20.8.6",
    "@sap/cds": "^9",
    "express": "^4",
    "@sap/xssec": "^4"
  },
  "devDependencies": {
    "@cap-js/sqlite": "^2",
    "@sap/cds-dk": "^9",
    "cds-plugin-ui5": "^0.13.0"
  },
  "scripts": {
    "start": "cds-serve",
    "watch-orders": "cds watch --open orders/index.html?sap-ui-xx-
viewCache=false --livereload false"
  },
  "private": true,
  "cds": {
    "requires": {
      "northwind": {
        "kind": "odata",
        "model": "srv/external/northwind",
        "credentials": {
          "url": "https://services.odata.org/V4/Northwind/Northwind.svc/"
        }
      },
      "[production]": {
        "auth": "xsuaa"
      }
    }
  },
  "workspaces": [
    "app/*"
  ],
  "sapux": [
    "app/orders"
  ]
}
```

Generate Fiori Application

1. Use keyboard shortcut (ctrl + shift + p) > search for Fiori Application Generator.
2. Select application type as List report
3. Select CAP service (Local OData Service)
4. Name the application.
5. Finish it.
6. For further customizations use the Page Map editor (ctrl + shift + p) add all requirements in UI.

Here is how CAP Application overview looks like in storyboard.



To run application	Command > cds w --profile hybrid cds w (short for cds watch) --profile selects from the package.json > cds > requires > "hybrid" configurations
Add the xsuaa	Configure the xs-security.json file

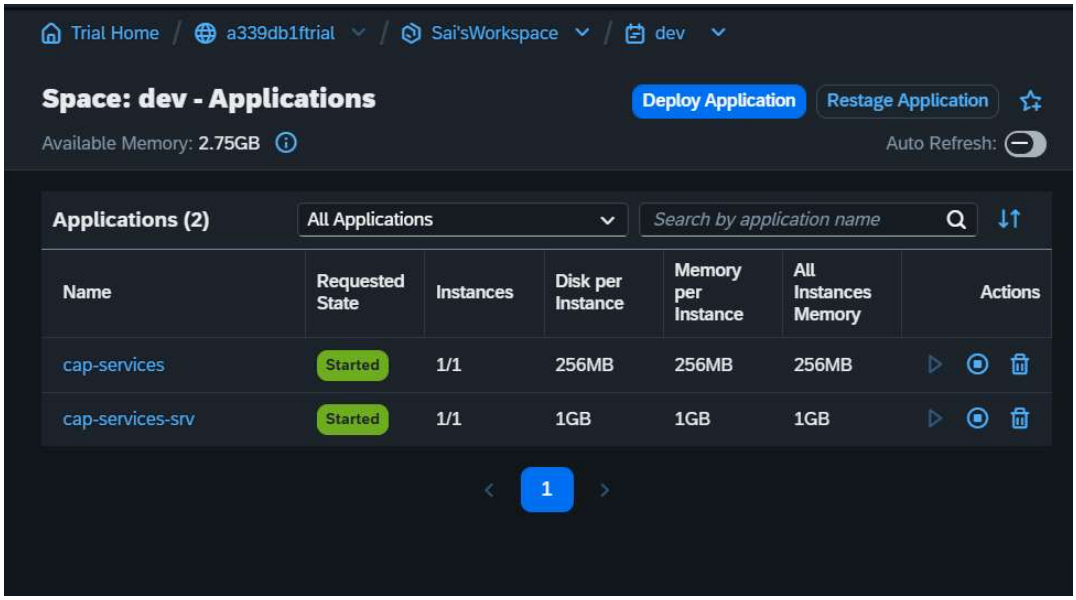
```
{
  "scopes": [
    {
      "name": "$XSAPPNAME.Viewer",
      "description": "Viewer"
    },
    {
      "name": "$XSAPPNAME.Admin",
      "description": "Admin"
    }
  ],
  "attributes": [],
  "role-templates": [
    {
      "name": "Viewer",
      "description": "Orders Adminstrator role",
      "scope-references": [
        "$XSAPPNAME.Viewer"
      ],
      "attribute-references": []
    },
    {
      "name": "Admin",
      "description": "Orders Viewer role",
      "scope-references": [
        "$XSAPPNAME.Admin"
      ],
      "attribute-references": []
    }
  ],
```

```
  "authorities": [
    "$ACCEPT_GRANTED_AUTHORITIES"
  ],
  "oauth2-configuration": {
    "token-validity": 900,
    "refresh-token-validity": 1800,
    "redirect-uris": [
      "https://*.cfapps.us10-001.hana.ondemand.com/login/callback"
    ],
    "autoapprove": "false"
  },
  "xsenableasyncservice": true
}
```

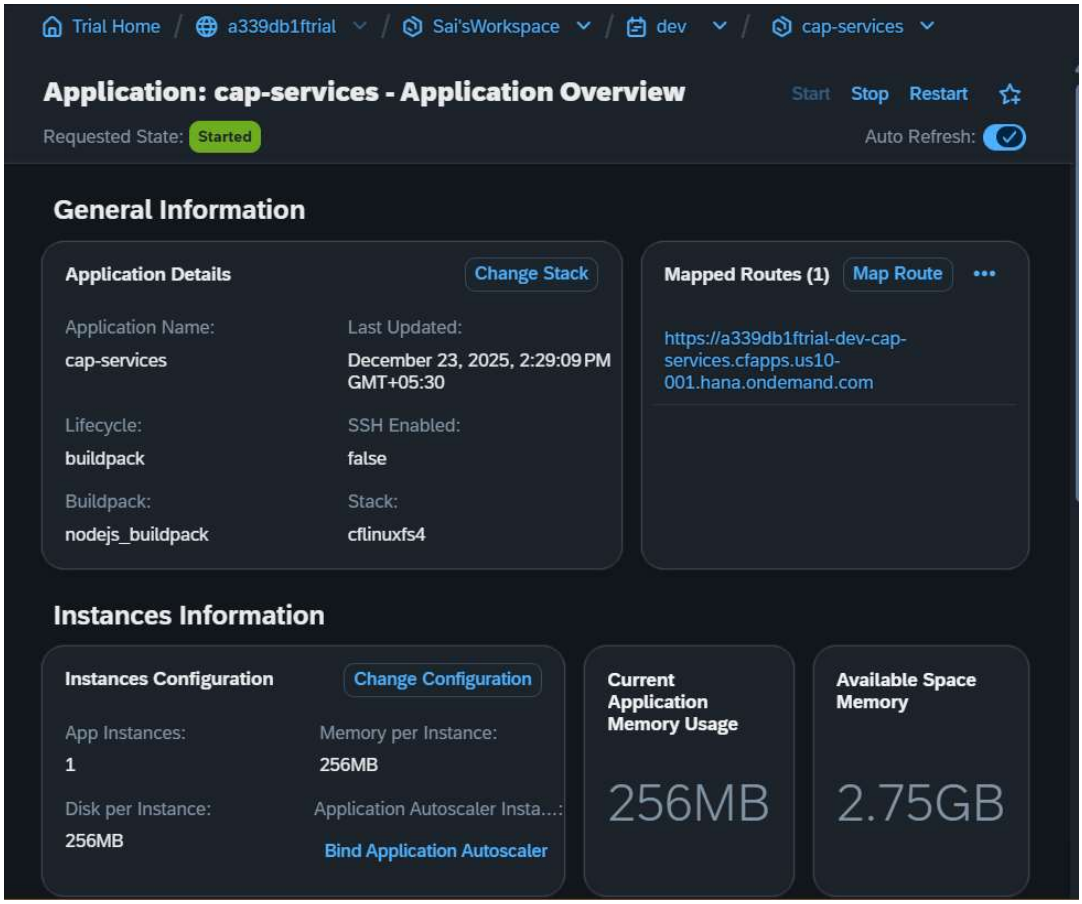
Deployment Process

1. Approuter	Create a approuter config. Files like xs-app.json, package.json with required configurations and route setup. Xs-app.json `{ "welcomeFile": "orders/webapp/", "authenticationMethod": "route", "sessionTimeout": 100, "pluginMetadataEndpoint": "/metadata", "routes": [{ "source": "^/orders/webapp/(.*)", "target": "$1", "localDir": "orders/webapp/", "authenticationType": "xsuaa" }, { "source": "^/(.*)$", "target": "$1", "destination": "srv-api", "csrfProtection": true }] }`
2. Create a YML file	In terminal> cds add mta This will add the mta.yml (Multi Target Application config. file)
3. Build the application	Use command > cds build --production --> this generates the production deployable files or simply right click on mta.yml file, select Build MTA Project
4. Deploy the MTAR file (Project file)	Simply right click on file inside the mta_archives/*.mtar and select Deploy MTA Archive. Now put your hands up, until application deploys. Unless you haven't face issue with : 1. cf login 2. Hana instance isn't started. 3. node modules aren't updated before building and deploying. 4. node modules aren't there in the application file, Space issue in cf etc. for other error download the log and check in OPERATION log file where the deployment failed.

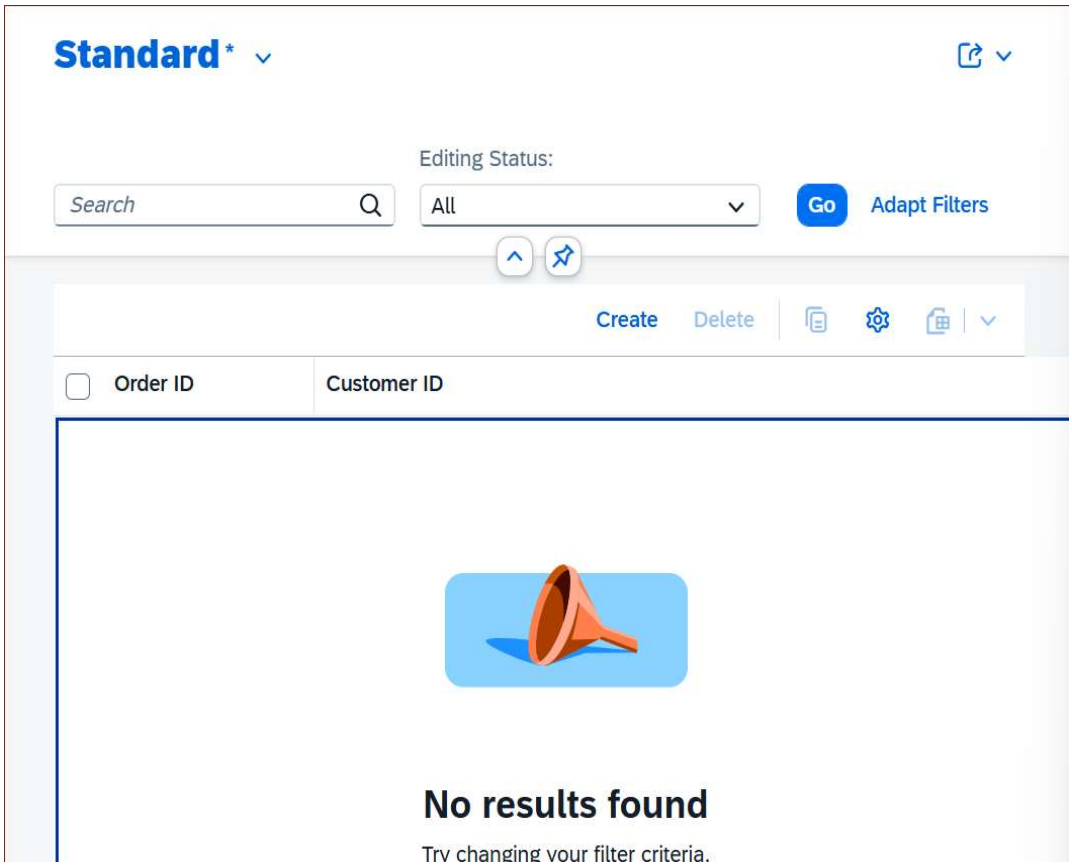
Go the BTP cockpit, inside your subaccount>cloud foundry> spaces > check for the application> click it and you can see the url over there like below.
Click to access the application with xsuaa security.
a.


Trial Home / a339db1ftrial / Sai'sWorkspace / dev
Space: dev - Applications
Deploy Application
Restage Application
Available Memory: 2.75GB
Auto Refresh:
Applications (2)
All Applications
Search by application name
Name
Requested State
Instances
Disk per Instance
Memory per Instance
All Instances Memory
Actions
cap-services
Started
1/1
256MB
256MB
256MB
cap-services-srv
Started
1/1
1GB
1GB
1GB
1

b.


Trial Home / a339db1ftrial / Sai'sWorkspace / dev / cap-services
Application: cap-services - Application Overview
Start
Stop
Restart
Requested State: Started
Auto Refresh:
General Information
Application Details
Change Stack
Application Name:
cap-services
Last Updated:
December 23, 2025, 2:29:09 PM GMT+05:30
Lifecycle:
buildpack
SSH Enabled:
false
Buildpack:
nodejs_buildpack
Stack:
cflinuxfs4
Mapped Routes (1)
Map Route
https://a339db1ftrial-dev-cap-services.cfapps.us10-001.hana.ondemand.com
Instances Information
Instances Configuration
Change Configuration
App Instances:
1
Memory per Instance:
256MB
Disk per Instance:
256MB
Application Autoscaler Insta...:
Bind Application Autoscaler
Current Application Memory Usage
256MB
Available Space Memory
2.75GB

c.



CI/CD Setup.

For CI/CD follow the below blog from sap developers, in detailed steps.
> https://developers.sap.com/tutorials/set-up-cicd..html

Here are the steps.

- Create a repository
- Initialize a repository in SAP Business Application Studio
- Enable SAP Continuous Integration and Delivery Service
- Access SAP Continuous Integration and Delivery Service
- Add Credentials
- Add a CI/CD job
- Create a GitHub webhook
- Test your job