SAP Customer Experience

SAP Commerce Cloud Portal

INTERNAL – SAP and Partners Only

We will learn about:

- Introduction to Cloud Portal
- Code Repository Connection
- Code Repository Structure
- Build & Deployment

The Context

The Cloud Portal enables you to **set up and deploy SAP Commerce Cloud in the public cloud**, which can be managed through its **self-service operations**.

Introduction to Cloud Portal

Cloud Portal Overview

Cloud Portal is a secure, browser-based self-service interface for your SAP Commerce Cloud projects running on SAP Cloud Platform

The Portal allows you to:

- Connect to your code repository
- Build your SAP Commerce Cloud application
- Deploy your builds to provisioned cloud environments
- Configure and secure your SAP Commerce Cloud application

Additional capabilities:

- Monitor performance
- User access management
- Create and restore snapshots
- Extend functionality

Prerequisites & User Management

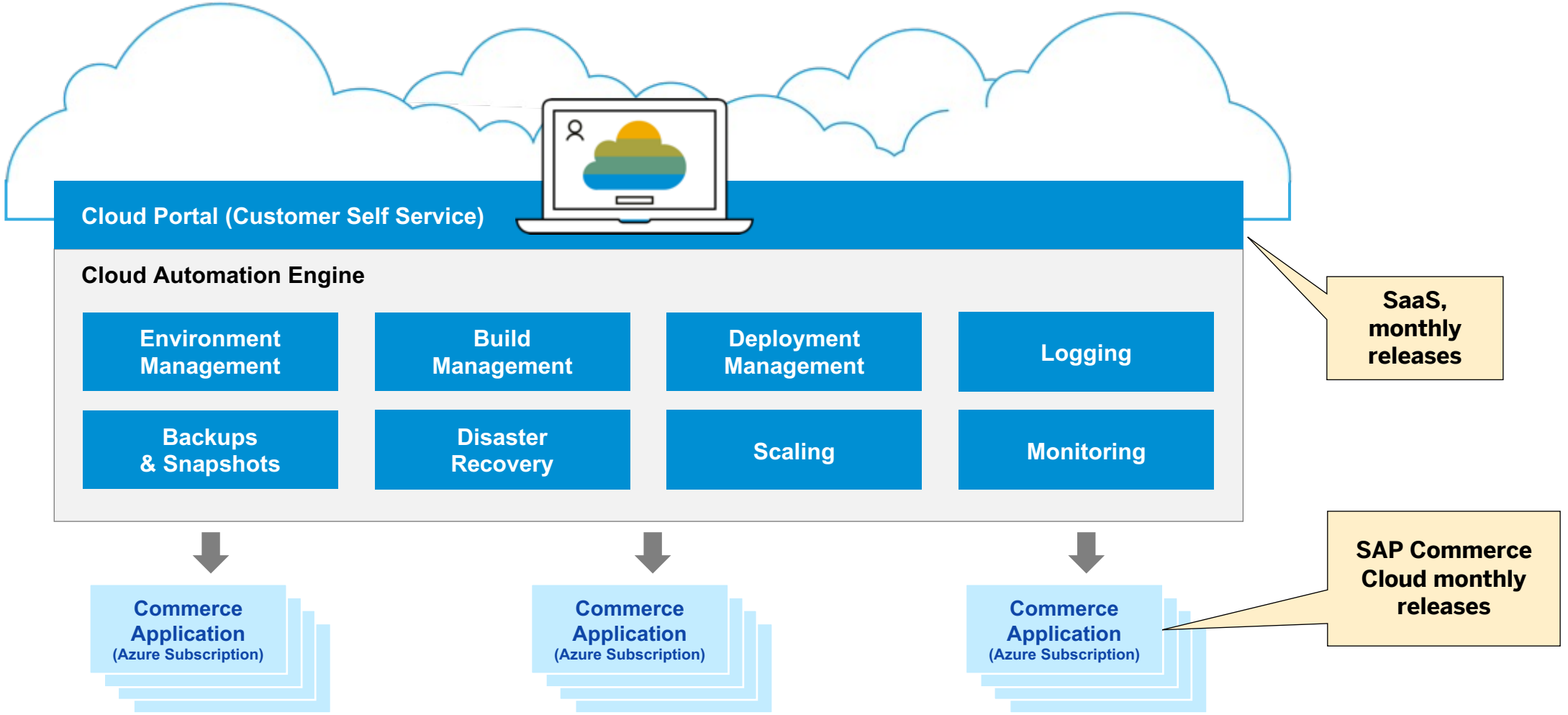
To use Cloud Portal and SAP Commerce Cloud in the public cloud, you will need:

- Internet connection and a supported browser
- Valid SAP account (S-User) with a user ID and a password
- Access to a publicly accessible Git-based code repository (Github, Bitbucket)

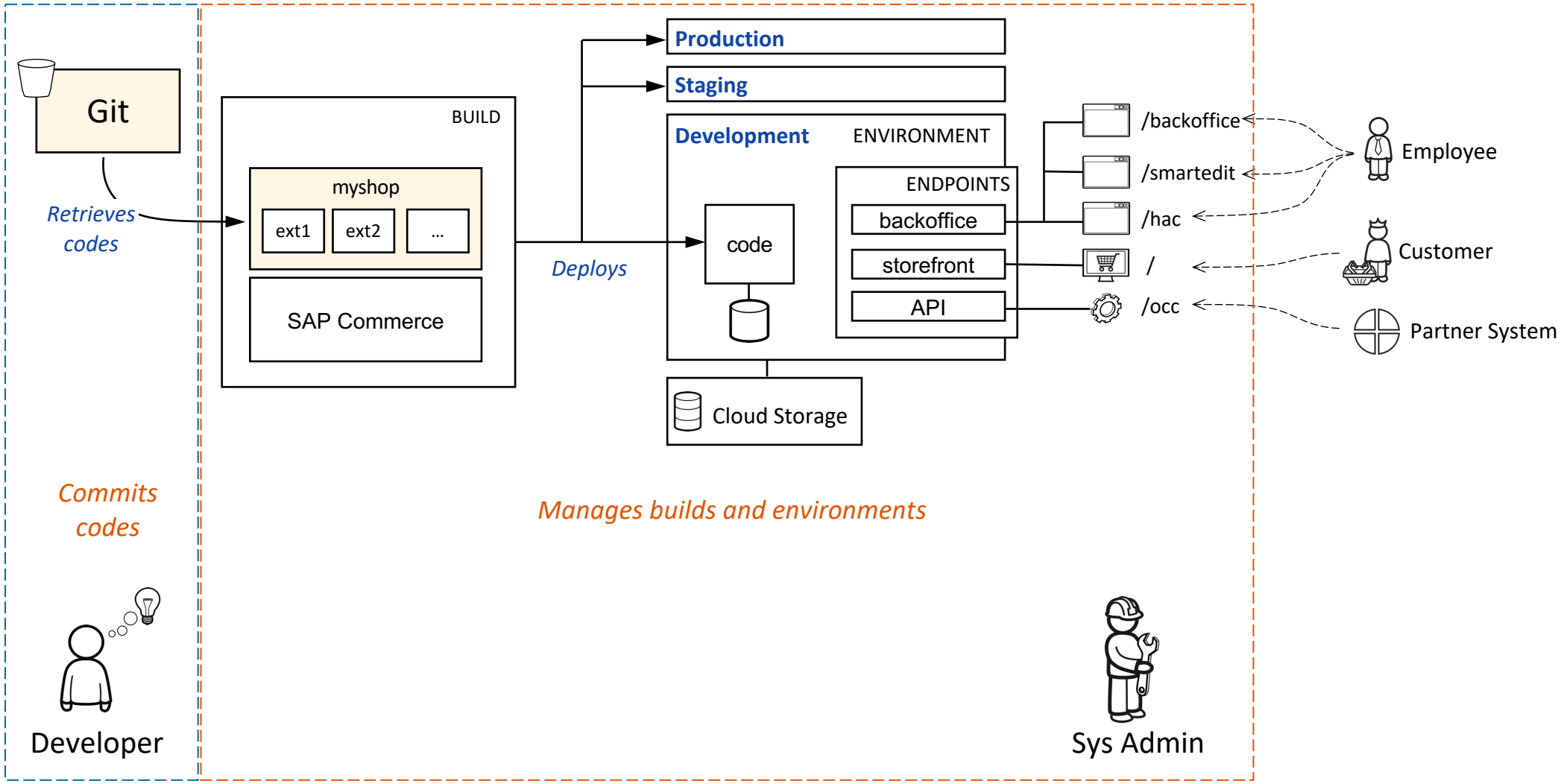
Customer Roles available on Cloud Portal:

- **CUSTOMER_DEVELOPER**: access to all Cloud Portal pages (including Dynatrace APM) and logs, but does not have access to the user management pages.
- **CUSTOMER_SYS_ADMIN**: assign permissions and manages Cloud Portal users, **in addition** to all the access rights of CUSTOMER_DEVELOPER.
- **CUSTOMER_LOG_VIEWER**: access to view only Logs (not including Dynatrace APM)

Architecture



SAP Commerce Cloud Business process



Environment Setup

- To run in the public cloud, a Commerce Cloud application needs an infrastructure of resources, collectively known as an **environment**

- There are 3 different types of environments:

  1. **Development:** used to create, configure and test application builds
  2. **Staging:** designed to simulate production activity
  3. **Production:** used to run live Commerce Cloud applications

- 3 statuses available for an environment:

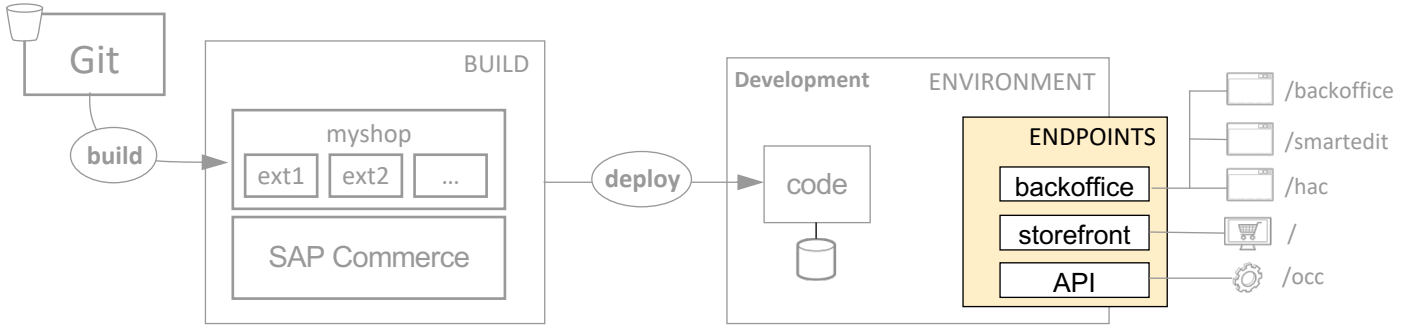
  1. **Provisioning:** created but not ready yet
  2. **Available:** provisioned and ready
  3. **Error:** not successfully provisioned

Environment Configuration

In an environment, you can edit properties to configure:

- **Endpoints:** define web routing and access
- **Cloud storage:** enables files to be uploaded to cloud hot folders or archiving of log files
- **Data backup:** create a back-up of your data and files at any time and provide the capability to restore
- **Deployment Configuration:** lets you add trusted certificates for JVMs, host aliases, and other properties
- **Service Configuration:** provides properties configuration for SAP Commerce Cloud aspects or services, including the generated password for the admin user

Endpoints in Cloud Portal



Endpoints configure your application's web tier

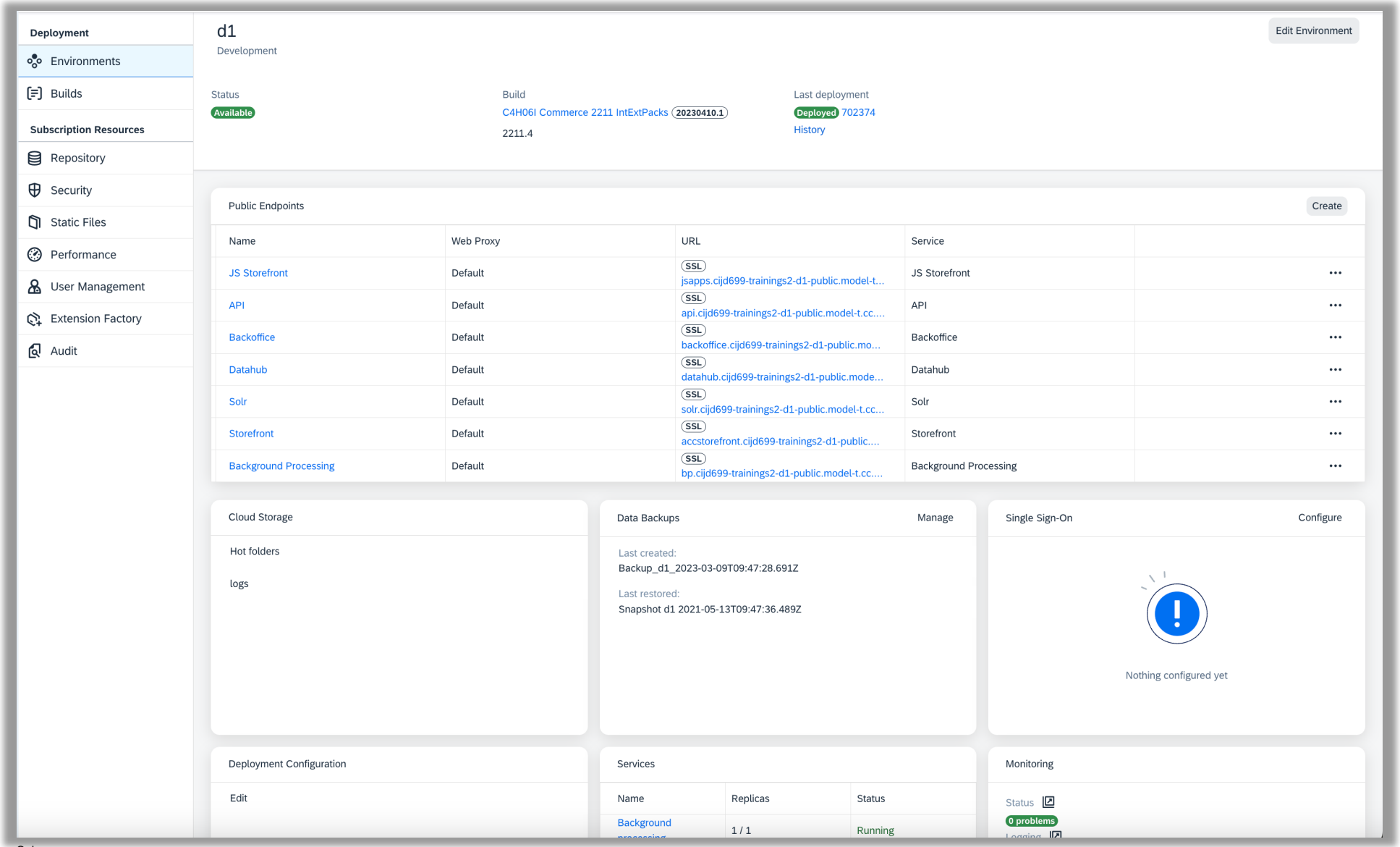
- Control web traffic
- Secure access to Commerce Cloud

Each new environment includes standard endpoints for:

- **API**
- **Solr**
- **Storefront**
- **JS Storefront**
- **Backoffice**
- **Logging**
- **Background Processing**

If you create other endpoints, define a corresponding aspect in your manifest file (`manifest.json`)

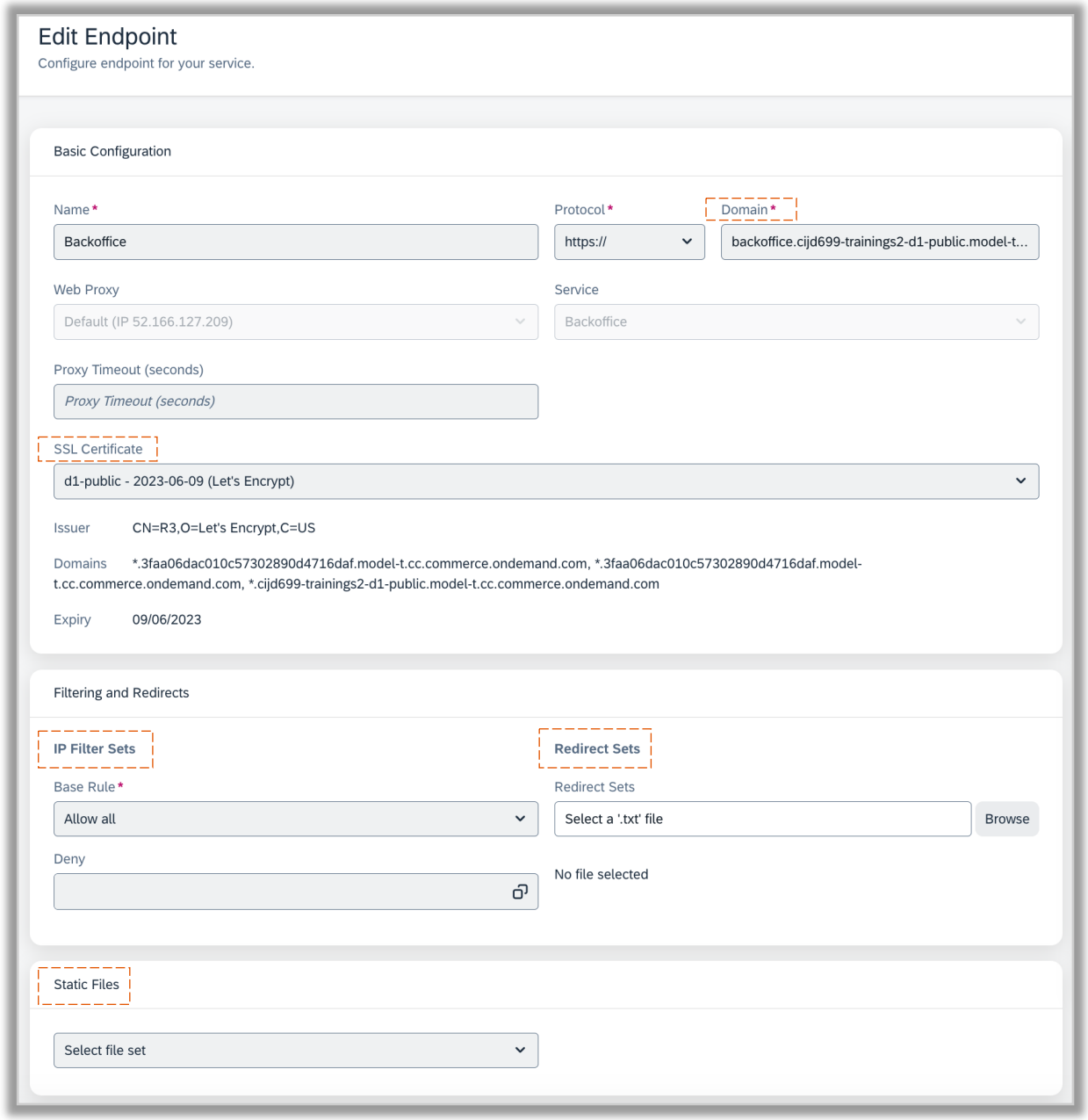
Environment Overview Page in Cloud Portal



Endpoint Configuration

You can configure endpoints in Cloud Portal by editing:

- Domain
- SSL certificate
- IP Filters
- Redirect Sets
- Static Files

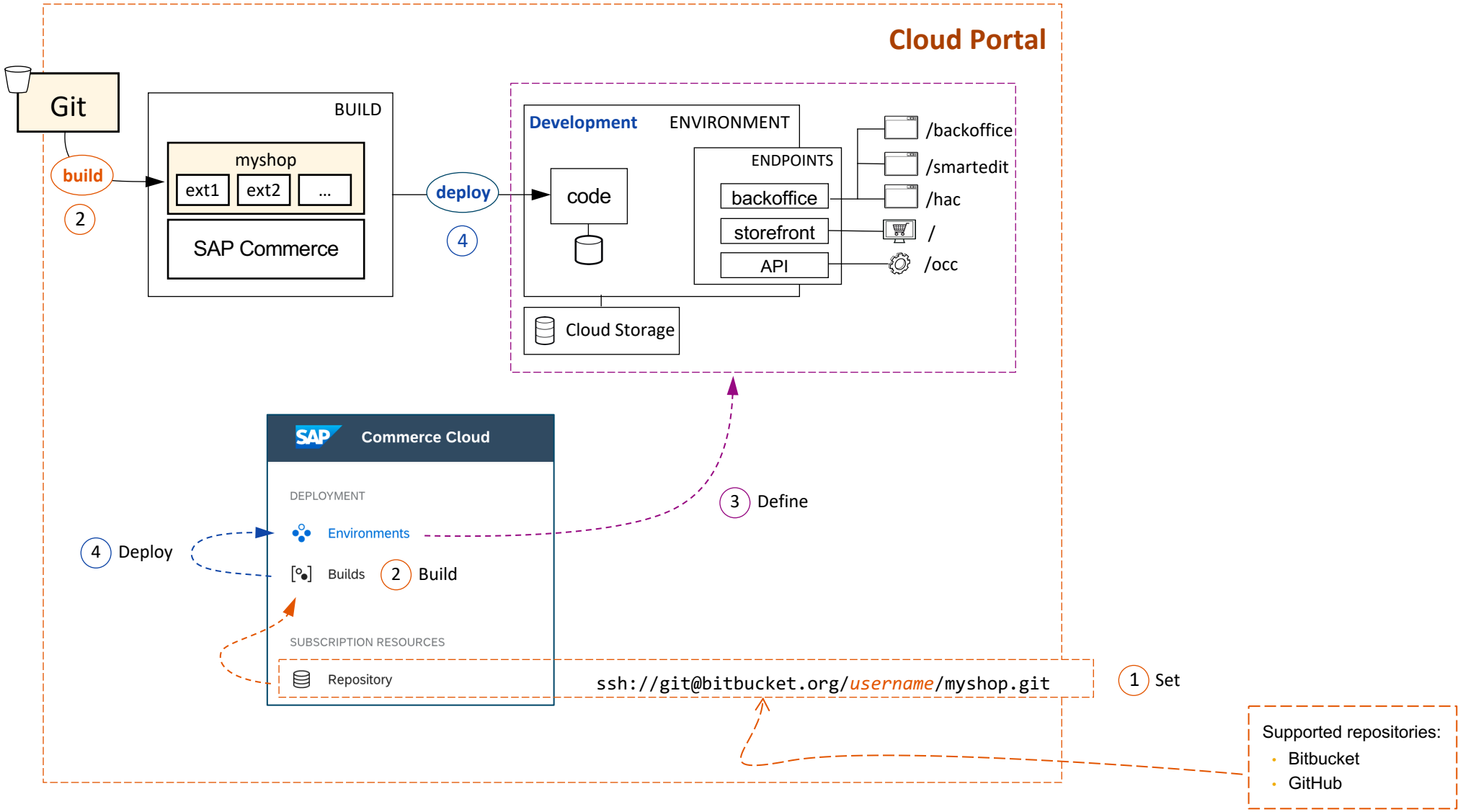


Demo


R100 BIKE
849.40
R300 Bike
373.69
R200 Bike
318.97
R200 BIKe 4s
596.34
M500 BIKE
209.16
eBike E180
75.44
M550 BIKE
74.40
Y200 Bike
62.31
eBike E181
54.06
C900 BIKE
42.47
C950 BIKE
29.90
M525 BIKE
135.61
Y120 Bike
28.38
Horns
19.07
Y240 Bike
43.84
in Thousand USD, %
2,987.26 (+30.59%)
Product Revenue Won Current
in Thousand USD
444.48
Revenue New Products
0.1
0.05
0
-0.05
-0.1
Lost due to price
348
Won due to product
Won due to price
Won due to service
1,657
Top Customers
Dimension
Industry ID
Territory
Sales Unit
Country
Competitor
Measure
in Thousand USD
NA
EMEA
APJ

Code Repository Connection

Cloud Portal

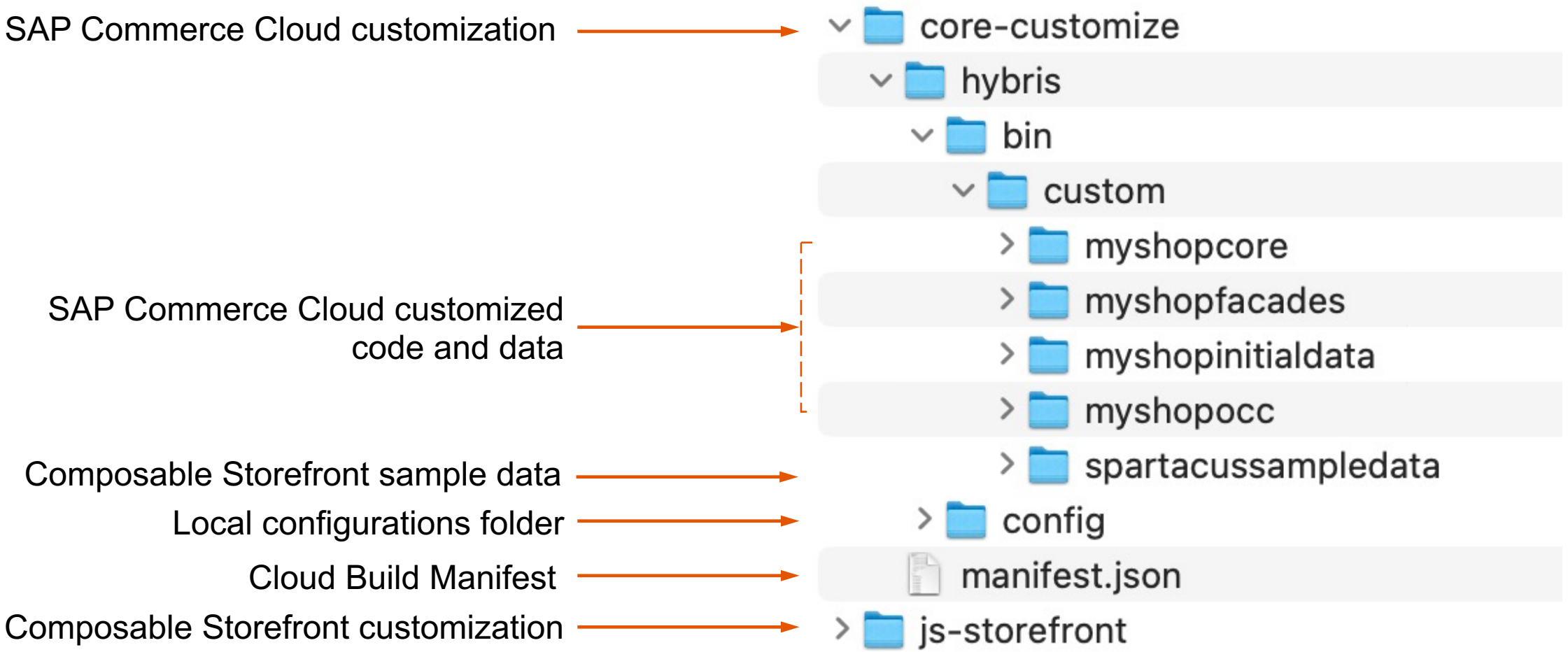


Demo


R100 BIKE
849.40
R300 Bike
373.69
R200 Bike
318.97
R200 BIKe 4s
596.34
M500 BIKE
209.16
eBike E180
75.44
M550 BIKE
74.40
Y200 Bike
62.31
eBike E181
54.06
C900 BIKE
42.47
C950 BIKE
29.90
M525 BIKE
135.61
Y120 Bike
28.38
Horns
19.07
Y240 Bike
43.84
in Thousand USD, %
2,987.26 (+30.59%)
Product Revenue Won Current
in Thousand USD
444.48
Revenue New Products
0.1
0.05
0
-0.05
-0.1
Lost due to price
348
Won due to product
Won due to price
Won due to service
1,657
Top Customers
Dimension
Industry ID
Territory
Sales Unit
Country
Competitor
Measure
in Thousand USD
NA
EMEA
APJ

Code Repository Structure

Git Repository Structure



Manifest.json File - Overview

Application builds are initiated through the Cloud Portal, API or CLI

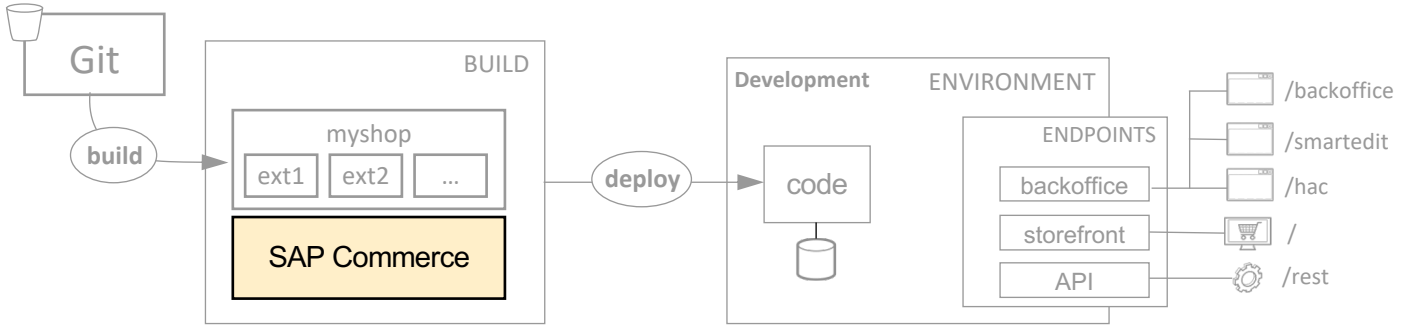
SAP Commerce Cloud relies on the **manifest.json** file to configure the build structure. This file keeps information for:

- Application version
- Extensions
- Properties
- Aspects
- Tests/webTests

manifest.json

```
{
   "commerceSuiteVersion":”2211",
   "extensions": [ … ],
   "useConfig": { … },
    ...
   "aspects": [ … ],
   "tests": { … },
   "webtests": { … }
}
```

Build Manifest Components – Application Version



The application version specifies the exact SAP Commerce version which should be downloaded. It's a mandatory component for the build.

You can use the latest patch version automatically, use a specific patch, or delay automatic patch upgrades.

Recommended practice: use the latest patch of the current version

manifest.json

```
{
    "commerceSuiteVersion": "2211",
    ...
}
```

Specify a particular patch explicitly:

manifest.json

```
{
    "commerceSuiteVersion": "2211.1",
    ...
}
```

Delay automatic patch upgrades:

manifest.json

```
{
    "commerceSuiteVersion" : [
        {
            "major" : "2211",
            "keepPriorPatchCount" : "1"
        }
    ]
    …
}
```

Build Manifest Components – SAP Commerce Configuration Reuse

We recommend that you configure your manifest.json file to point to a common config folder in your repository

This ensures you align your local development environment with what is deployed to Commerce Cloud

The folder can contain configuration for:

- Solr
- Properties
- Extensions
- Languages

manifest.json

```
"useConfig": {
  "properties":
  [ { "location": "/config/local.properties" },
    {..}
  ],
  "extensions": {
     "location": "/config/localextensions.xml",
     "exclude": [
        "backoffice"
     ]
  },
  "solr": { "location": "/config/solr" },
  "languages" : { "location": "_LANGUAGES_" }
}
```

Build Manifest Components – Explicit Extension Use

Instead of (or in **addition** to) using the extensions listed in localextensions.xml, you may also explicitly name the extensions you want to use:

- Custom extensions from your Git repository
- SAP-managed extensions implemented to run SAP Commerce in the public cloud

SAP Commerce extensions do not need to be listed – the ones needed for your build are downloaded automatically.

Examples of explicitly naming extensions:

In addition to using localextensions.xml

manifest.json

```
"extensions": [
   "azurecloudhotfolder"
]
```

Instead of using localextensions.xml

manifest.json

```
"extensions": [
   "myshopcore",
   "myshopfacades",
   "myshopocc",
   "spartacussampledata",
   etc…
]
```

Build Manifest Components – Properties

To define your properties in the manifest.json file, you need to specify the following attributes:

- `key`: the property name or id (mandatory attribute)
- `value`: the value of the property (mandatory)
- `persona`: specifies the environment type for the property
- `secret`: a secret property value which will be excluded from the build logs

We recommend adding the following properties to prevent the SAP Commerce database from overloading:

- `auditing.enabled=false`
- `default.session.timeout=360`

Persona is an optional tag limiting a property value to a specific deployment type (development, staging, or production)

manifest.json

```
"properties": [
   {
      "key": "your.property.1",
      "value": "your.value.1",
      "persona": "development"
   }
]
```

Build Manifest Components – Language Packs

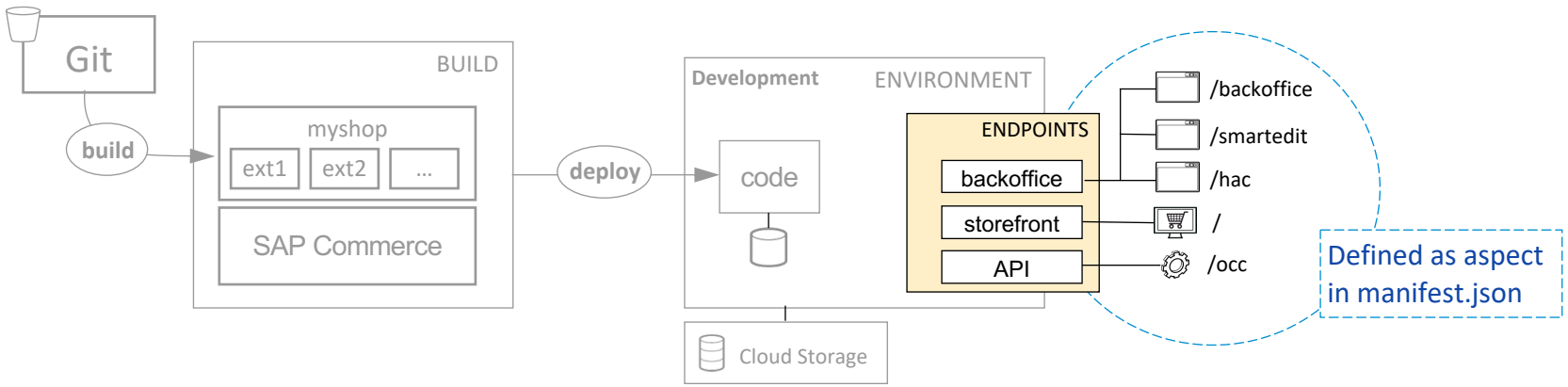
To deploy language packs into Commerce Cloud:

- Create a folder called `_Languages_` in the root of your code repository
- Add the language pack zip files to this folder
- Add the following property into your manifest.json file, listing the languages that you want to include:

manifest.json

```
"properties": [
   {
      "key": "lang.packs",
      "value": "en,de,fr,es,…",
   }
]
```

Build Manifest Components – Aspects



The aspects are composed of:

- **`properties`** and
- **`webapps`**: all the available web applications for the aspect, including the `contextPath` for accessing the application

The build process defines available aspects and provides automatic configuration for the following:

- **`accstorefront`**: configures the storefront
- **`backoffice`**: configures Backoffice
- **`backgroundProcessing`**: performs tasks in the background
- **`admin`**: executes ant admin tasks, like `ant initialize` and `ant updatesystem`
- **`api:`** configures OCC web services

Aspect Example in Manifest

manifest.json

```
"aspects": [
  {
    "name": "backoffice",
    "properties": [
      define properties here
    ],
    "webapps": [
      {
        "name": "backoffice",
        "contextPath": "/backoffice",
      }
      {
        "name": "hac",
        "contextPath": "/hac"
      }
      define more web apps here
    ]
  },
```

manifest.json (continued)

```
  {
   "name": "accstorefront",
      "properties": [
        {
          "key": "storefrontContextRoot",
          "value": ""
        }
        more properties…
      ],
      "webapps": [
        {
          "name": "myshopstorefront",
          "contextPath": "",
        }
        {
          "name": "mediaweb",
          "contextPath": "/medias"
        },
        more web apps…
      ]
    },
```

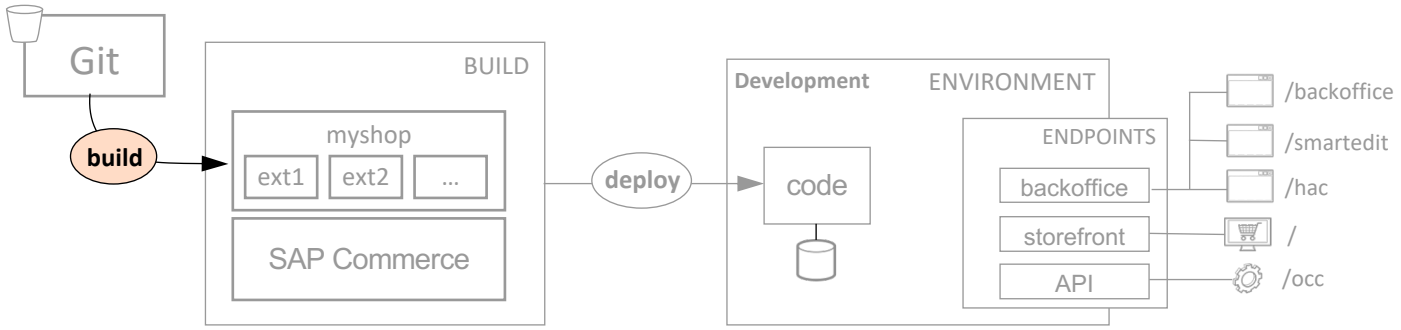
manifest.json file reference

In the product documentation, you can find the manifest.json components reference from here:

Code Repository Setup and Build Guidelines on //help.sap.com

Build & Deployment

Build Overview



A **Build** is a collective term for all the artifacts (security-hardened Docker images) generated by the **Builder** according to the **application definition**.

The Builder creates customized Builds using codes taken from the customer code repository

Customizations must be integrated into the Commerce Suite as new Docker images

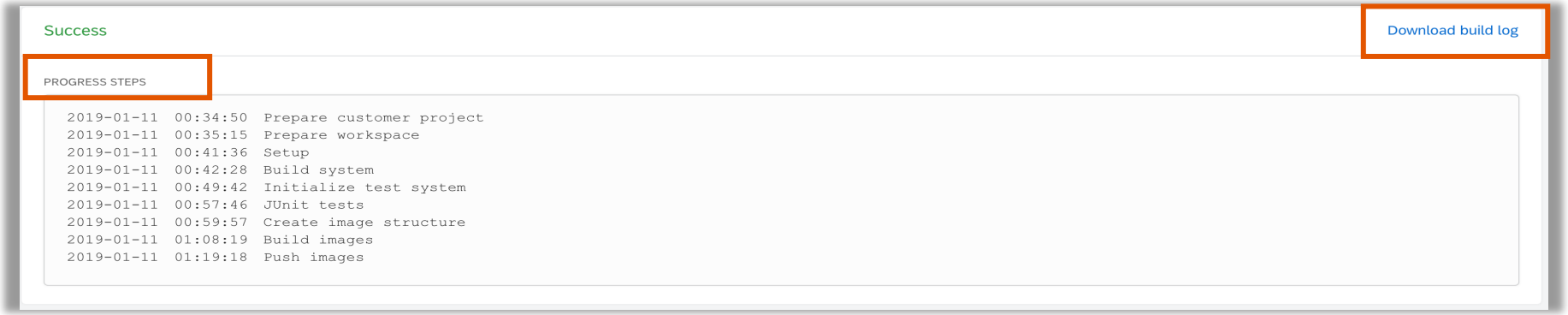
The Build Process

The process of creating an application build is initiated via the Cloud Portal, API or CLI

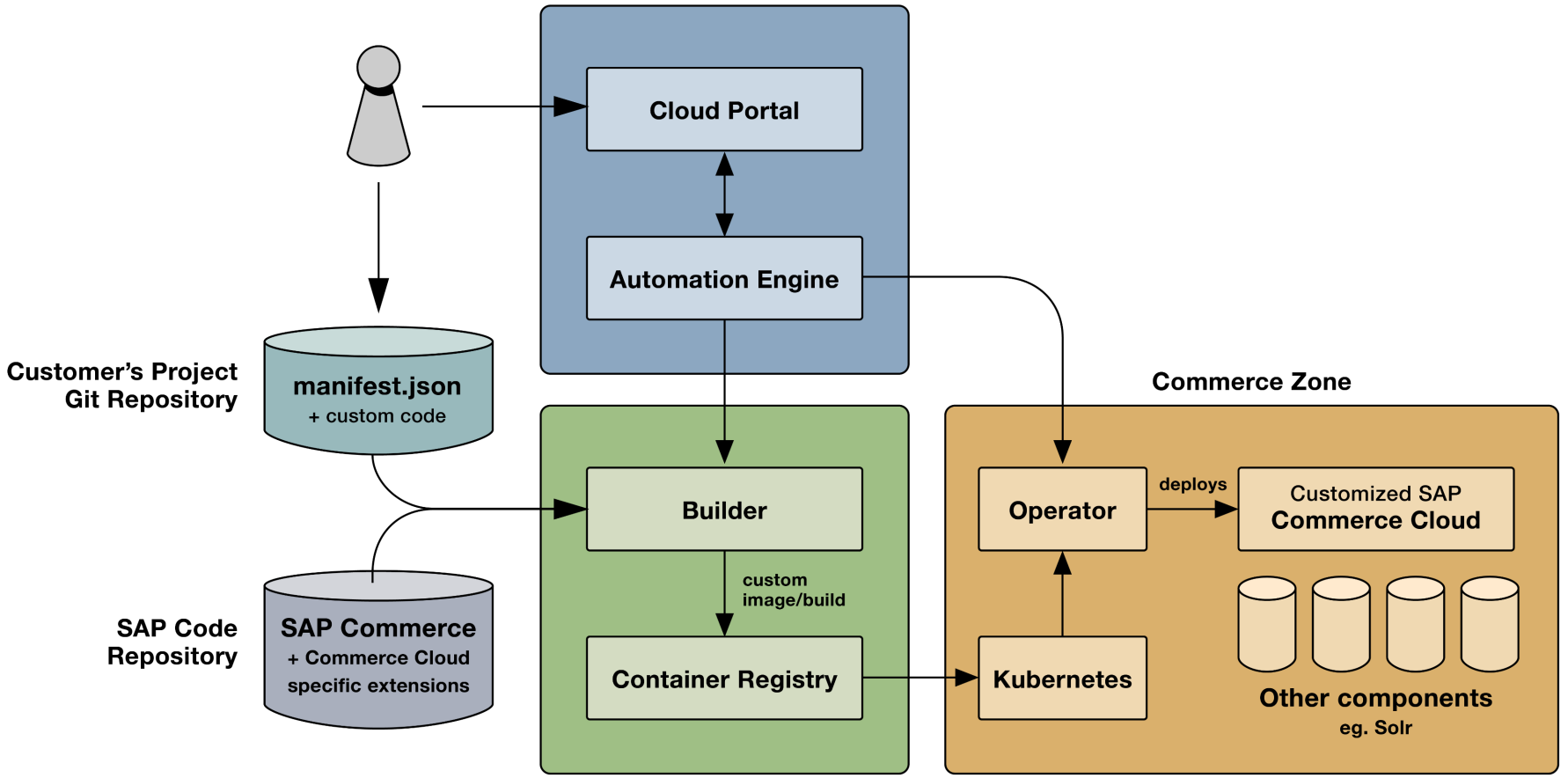
The Build process has 4 different statuses:

- **Scheduled:** the Build process waits to start
- **Building:** the Build process starts
- **Success:** the Build has finished successfully and is ready for deployment
- **Error:** The Build failed with errors

You can check or download the whole Build logs from the specific build in Cloud Portal



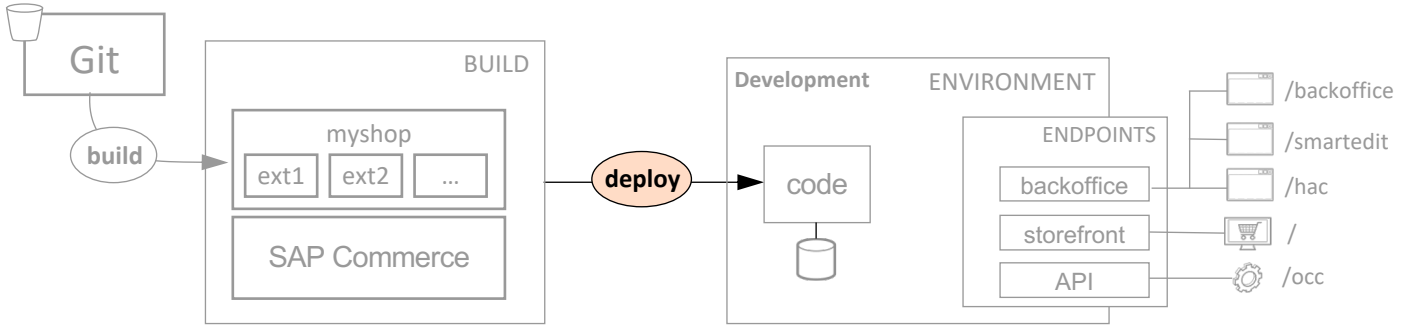
The Build Process Diagram

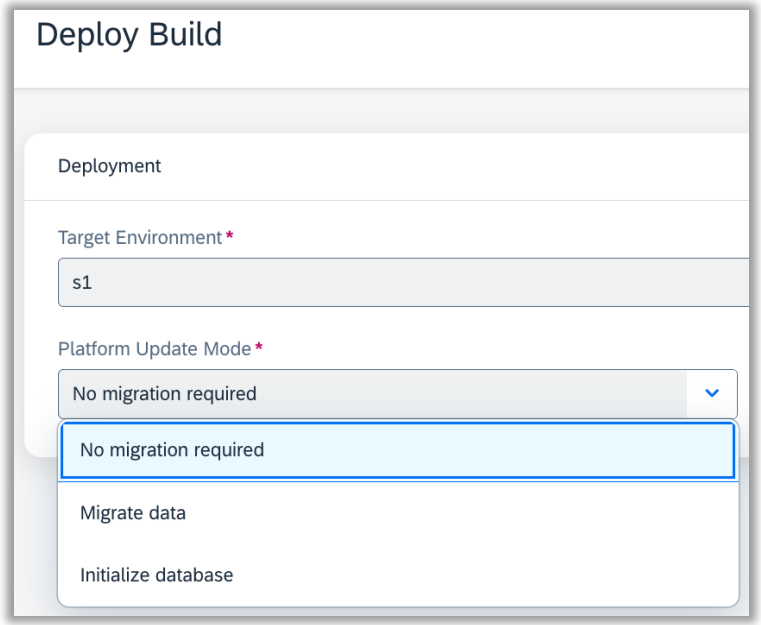


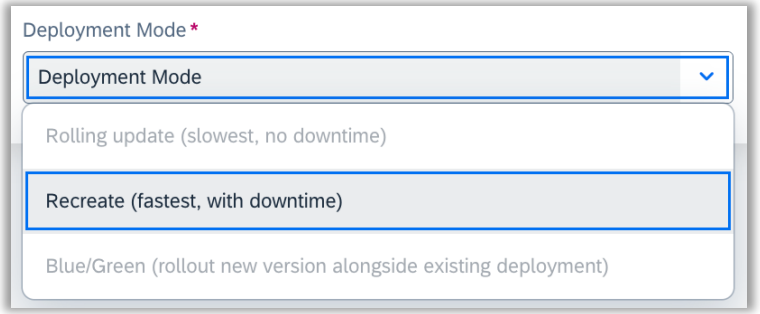
Demo


R100 BIKE
849.40
R300 Bike
373.69
R200 Bike
318.97
R200 BIKe 4s
596.34
M500 BIKE
209.16
eBike E180
75.44
M550 BIKE
74.40
Y200 Bike
62.31
eBike E181
54.06
C900 BIKE
42.47
C950 BIKE
29.90
M525 BIKE
135.61
Y120 Bike
28.38
Horns
19.07
Y240 Bike
43.84
in Thousand USD, %
2,987.26 (+30.59%)
Product Revenue Won Current
in Thousand USD
444.48
Revenue New Products
0.1
0.05
0
-0.05
-0.1
Lost due to price
348
Won due to product
Won due to price
Won due to service
1,657
Top Customers
Dimension
Industry ID
Territory
Sales Unit
Country
Competitor
Measure
in Thousand USD
NA
EMEA
APJ

Deployment in Cloud Portal







To deploy your Build:

1. Go to Cloud Portal > *Builds* > *Deploy to Environment*
2. Select the target environment
3. Select *appropriate data migration mode & deployment mode*.

Commerce Cloud APIs

With a valid API token, you can use a REST client to call SAP Commerce Cloud APIs. You can generate API token in the Cloud Portal:

- Build APIs allow you to test builds locally or on your continuous integration pipeline. When testing is complete, you can trigger a build
- Deployment APIs allow you to trigger a deployment outside of the Cloud Portal
- ServiceProperties APIs allow you to get/put all Service Properties for/into a SAP Commerce Cloud environment
- Databackup APIs allow you to manage data backup and restore.

You can use these APIs to:

- Affect cloud builds or deployments, as well as track the progress and status
- Write custom scripts to automate cloud build and deployment processes
- Send HTTP requests to Commerce Cloud API endpoints using client libraries

Command Line Interface

The CLI tool is packaged as a ZIP file, downloadable from SAP Software Downloads (search for CX COMM CLOUD MAN CLI).

Use the CLI tool to initiate build and deploy actions in Commerce Cloud outside of Cloud Portal.

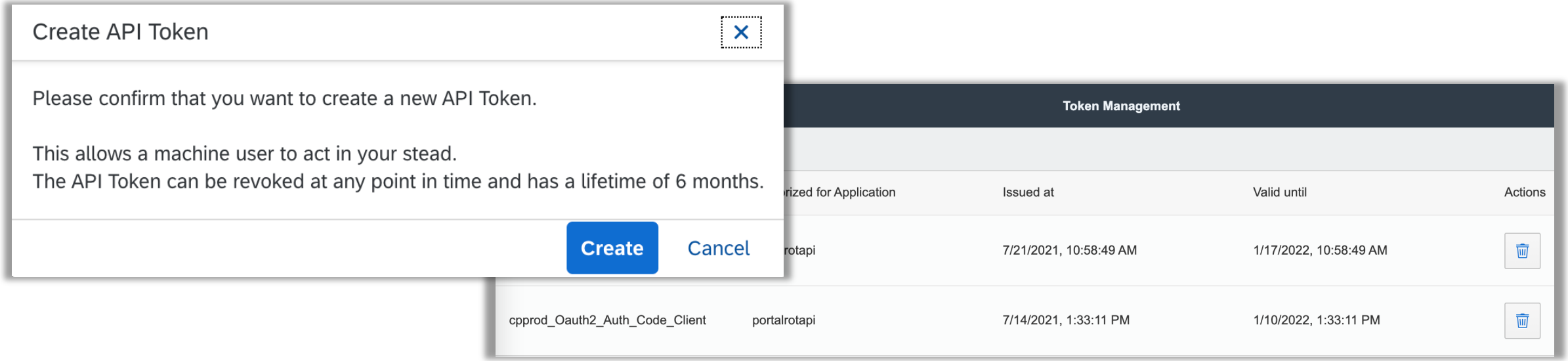
The following actions from a command line can be performed:

- Trigger a build
- Find a list of available builds
- Find the details for a specific build
- Download build logs
- Trigger a deployment
- Find a list of deployments
- Find the details for a specific deployment
- Get the options for canceling a deployment and cancel a deployment if necessary

Commerce Cloud API Token

API tokens contain the user credentials that grant you access to Commerce Cloud APIs or CLI tools.

- Find the Token Management page at the top-right of the cloud portal page, under Account | API Token.
- API tokens can be generated in the Cloud Portal
- To access a list of existing tokens, navigate to the Token Management page.
- With a valid token, you can use a REST client to call Commerce Cloud APIs or the CLI tools.
- To revoke a token, access the Token Management page and click Revoke on the token row.



Demo


R100 BIKE
849.40
R300 Bike
373.69
R200 Bike
318.97
R200 BIKe 4s
596.34
M500 BIKE
209.16
eBike E180
75.44
M550 BIKE
74.40
Y200 Bike
62.31
eBike E181
54.06
C900 BIKE
42.47
C950 BIKE
29.90
M525 BIKE
135.61
Y120 Bike
28.38
Horns
19.07
Y240 Bike
43.84
in Thousand USD, %
2,987.26 (+30.59%)
Product Revenue Won Current
in Thousand USD
444.48
Revenue New Products
0.1
0.05
0
-0.05
-0.1
Lost due to price
348
Won due to product
Won due to price
Won due to service
1,657
Top Customers
Dimension
Industry ID
Territory
Sales Unit
Country
Competitor
Measure
in Thousand USD
NA
EMEA
APJ

Key Points

1. Cloud Portal enables you to build, deploy and configure your SAP Commerce Cloud in the public cloud.
2. Developers can commit their custom SAP Commerce Cloud codes to a **Git-based** repository and connect it to the Cloud Portal.
3. The build process is driven by a customizable **manifest.json** file.
4. The build and deployment processes can be handled manually in Cloud Portal or via the Commerce Cloud API and CLI tools.
5. **Cloud Hot Folder** is covered in the Learning Hub live sessions.

"SAP Commerce Cloud – Additional Technical Essentials"

References

- SAP Commerce Cloud documentation:
- https://help.sap.com/docs/SAP_COMMERCE_CLOUD_PUBLIC_CLOUD

- Commerce Cloud Repository:
- https://help.sap.com/viewer/1be46286b36a4aa48205be5a96240672/latest/en-US/a3d5658925f1430f81bbd894a6abd72e.html

- Build Manifest Components :
- https://help.sap.com/viewer/1be46286b36a4aa48205be5a96240672/latest/en-US/2be55790d99e4a1dad4caa7a1fc1738f.html

- SAP Commerce Cloud API documentation:
- https://help.sap.com/viewer/452dcbb0e00f47e88a69cdaeb87a925d/latest/en-US/66abfe678b55457fab235ce8039dda71.html

- Cloud Portal documentation:
- https://help.sap.com/docs/SAP_COMMERCE_CLOUD_PUBLIC_CLOUD/0fa6bcf4736c46f78c248512391eb467/2bb13fbce5d145ec89b09ea5d2feef25.html

- Cloud Automation Components:
- https://help.sap.com/viewer/20125f0eca6340dba918bda360e3cdfa/latest/en-US/dbafcc89e99c427c83cd8bbfdd7ac790.html

Thank you.