Best Practices for SAP BTP

THE BEST RUN **SAP**

Content

1 Best Practices for SAP BTP

This document helps you plan and set up your landscape and your lifecycle management for running applications on SAP Business Technology Platform (SAP BTP). It contains best practices and recommendations for planning development projects – from setting up the correct organizational structure to creating an account and security model, to developing and operating applications.

Is This Guide for You?

This guide is the starting point for setting up application lifecycle management for your specific use case, business, and IT landscape. It contains recommendations and best practices that give you an overview of what you should consider when planning development projects on SAP BTP. It does not contain specific task descriptions, but does include links to step-by-step instructions when required.

If you're an **architect or a development project lead**, this guide helps you plan your organizational setup and your account and security model. It has an overview of processes, tools, and services that are available for developing and operating applications. Please note that SAP is currently renovating and adding core functionalities to SAP BTP and as part of this process, global accounts are migrated from the existing cloud management tools feature set A to the renovated cloud management tools feature set B. New global accounts will be on feature set B which brings a more flexible account model with so-called **directories** to group and manage subaccounts inside your global account. For more information, see Cloud Management Tools - Feature Set Overview. In this Best Practices guide, you will get recommendations for feature set A and B. They are marked as such, but please ensure that you know if you are on feature set B already.

If you're an **administrator or developer**, this guide helps you define the correct methodologies and tools for your development project.

If you're an **SAP partner**, this guide helps you to set up SAP BTP for developing and running production services for your customers.

> **i** Note
>
> This guide is targeted at customers who want to run and use applications in a production environment. If you're an SAP BTP trial user, you might still find that some information in this guide is useful. Check out the following page for more details about trial accounts: Trial Accounts and Free Tier. Please note that the services available in the trial version differ from the ones in the enterprise version.

How to Use This Guide

If you're **new to SAP BTP**:

- SAP BTP is an integrated offering comprised of four technology portfolios: database and data management, application development and integration, analytics, and intelligent technologies. The platform offers users the ability to turn data into business value, compose end-to-end business processes,

and build and extend SAP applications quickly. It is a vital part of the transition to the Intelligent Enterprise. The application development and integration capabilities of SAP BTP enable you to develop new applications, extend existing ones, and are at the center of integration scenarios. For more information about the Intelligent Enterprise, see https://www.sap.com/products/intelligent-enterprise.html

- Basic Platform Concepts [page 7] – regions, environments, accounts, members, quotas, capabilities and links to in-depth explanations.
- Shared Responsibility Model Between You and SAP [page 19] – your responsibilities and SAP's responsibilities when it comes to application lifecycle management.
- Getting Started Checklist [page 21] – prerequisites and steps to help you get started with your development project.

If you're **familiar with SAP BTP**:

Plan and set up your landscape to manage the lifecycle of your cloud applications.
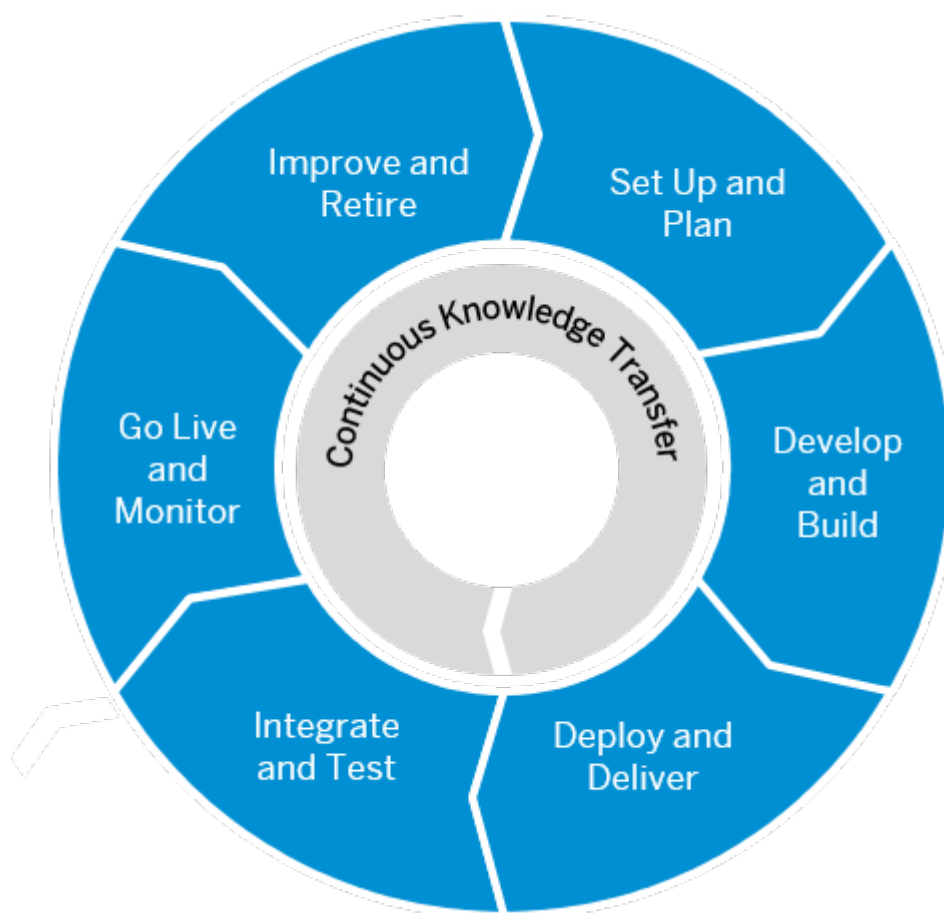
- Improve and Retire [page 101]
- Set Up and Plan [page 23]
- Develop and Build [page 61]
- Deploy and Deliver [page 78]
- Integrate and Test [page 93]
- Go Live and Monitor [page 96]

1. **Set Up and Plan [page 23]** – build teams, set up your account and security model, and create an enrollment process for your development projects.
2. **Develop and Build [page 61]** – find out about the tools and programming languages that are available on SAP BTP, and how to use multitarget applications to efficiently manage dependencies.
3. **Deploy and Deliver [page 78]** – deploy and deliver simple and multitarget applications.
4. **Integrate and Test [page 93]** – test and integrate your application with other solutions.
5. **Go Live and Monitor [page 96]** – learn what's important for going live and monitoring applications, services, and hybrid landscapes.
6. **Improve and Retire [page 101]** – make improvements to your application, perform housekeeping, and learn about what's important to consider when you want to retire it.

Related Resources

The learning jouneys Lifecycle-Management of Applications on SAP BTP and DevOps for Application Development on SAP BTP are extensive, well-structured collections of links to resources such as videos, blog posts, openSAP courses, and additional documentation.

2 Basic Platform Concepts

SAP BTP offers users the ability to turn data into business value, compose end-to-end business processes, and build and extend SAP applications quickly.

The services and solutions of SAP BTP are available on multiple cloud infrastructure providers. The **multi-cloud foundation** supports different environments, such as Cloud Foundry, ABAP, and Kyma, as well as multiple different regions, and a broad choice of programming languages.

The central point of entry to the cloud platform is the **SAP BTP cockpit**, where you can access your accounts and applications and manage all activities associated with them.

The figure below depicts the relationship between a global account, its subaccounts, environments, regions, entitlements, and quotas. It shows the administrative tasks to be considered at the global acccount level as well as at the subaccount level.
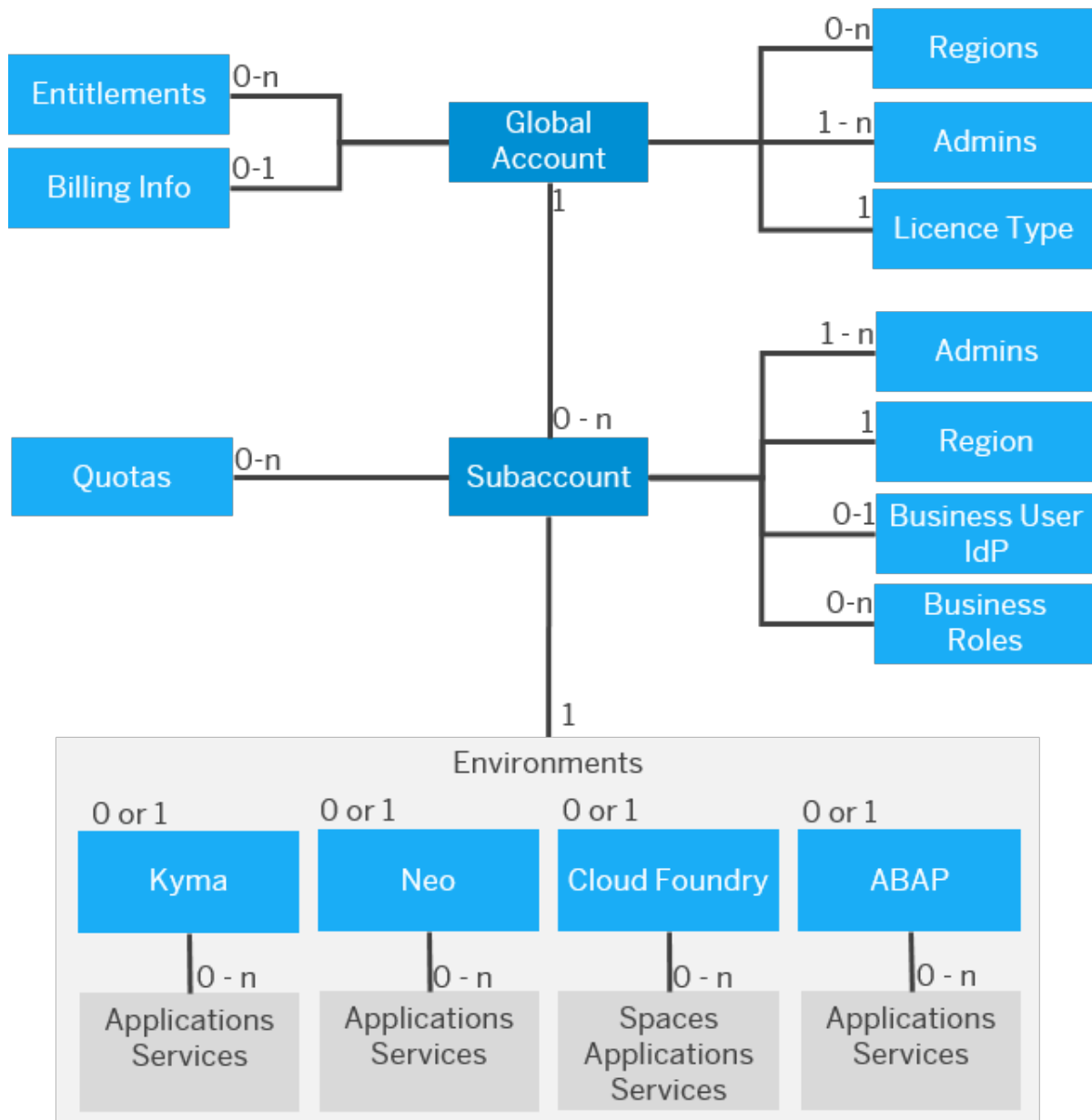
Overview of Global Accounts and Subaccounts (Feature Set A)

If your global account is on cloud management tools, feature set B, the account structure looks different. With feature set B, the new hierarchical element called **directory** is introduced, which is essentially a grouping of subaccounts. Furthermore, subaccounts can have multiple environments.

The figure below depicts the relationship between a global account, its directories, subaccounts, environments, regions, entitlements, and quotas for feature set B.
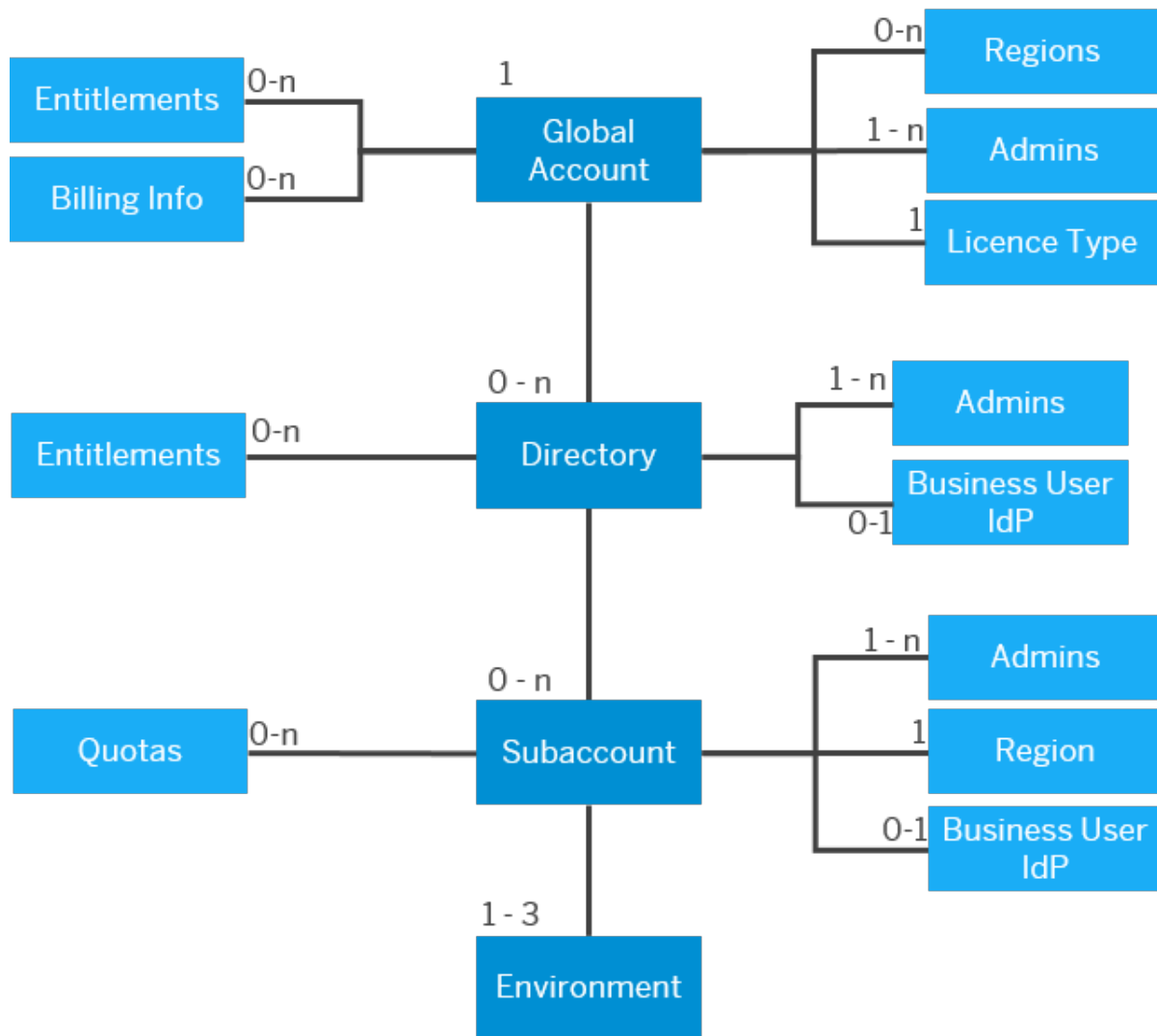
Overview of Global Accounts, Directories, and Subaccounts (Feature Set B)

Level	Administrative Tasks
Global Account	• For each commercial model (licence type), you get a separate global account. • Appoint at least one person as administrator. The administrator is responsible for adding new subaccounts, adding members to a global account, and managing the entitlements. We recommend that you also appoint at least one substitute administrator. If the main administrator leaves the company or is unavailable, it's important that you have someone who's available to take over these tasks. • You purchase entitlements for each global account (according to your commercial model). The administrator of the global account distributes quotas to the individual subaccounts.

Level	Administrative Tasks
Directory (optional, feature set B)	• Directories are only available with feature set B. See Cloud Management Tools — Feature Set Overview. They are groups of subaccounts that you can manage, operate, and analyze together. • Appoint at least one person as administrator. The administrator is responsible for adding new subaccounts, managing members, and managing entitlements. We recommend that you also appoint at least one substitute administrator. If the main administrator leaves the company or is unavailable, it's important that you have someone who's available to take over these tasks.
Subaccount	• Each subaccount runs in exactly one region (data center) and one environment. • Appoint at least one person as administrator. The administrator is responsible for adding new members to the subaccount and assigning their business roles. We recommend that you also appoint at least one substitute administrator. If the main administrator leaves the company or is unavailable, it's important that you have someone who's available to take over these tasks.

For more information, see Account Model [page 13] and Setting Up Your Account Model [page 27].

2.1   Regions

You can deploy applications in different **regions**. Each region represents a geographical location (for example, Europe, US East) where applications, data, or services are hosted.
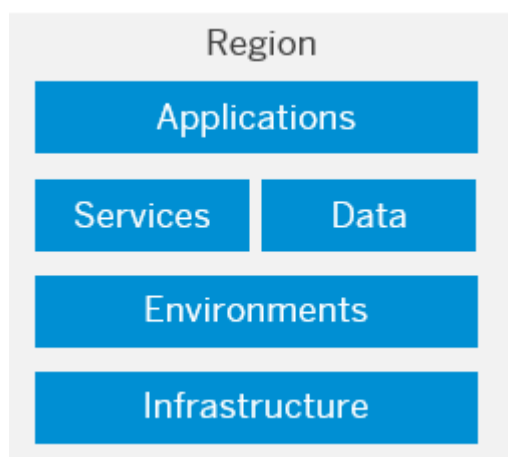
Regions are provided either by SAP or by our Infrastructure-as-a-Service (IaaS) partners Amazon Web Services (AWS), Microsoft Azure, Google Cloud, and Alibaba Cloud. The third-party region providers operate the infrastructure layer of the regions, whereas SAP operates the platform layer and Cloud Foundry.

A region is chosen at the subaccount level. For each subaccount, you select exactly one region (that is one data center).

For more information, see Regions.

Selecting a Region

When deciding on the location of your Platform as a Service (PaaS), consider existing Software as a Service (SaaS) and Infrastructure as a Service (IaaS) and try to locate it close to those or even in the same data center. You can also optimize application performance (response time, latency) by selecting a region that's geographically close to your users. However, the selection of a region is also dependent on many other factors: First, check the availability of specific services in the individual regions. Second, ensure that you comply with security requirements, such as country- or industry-specific data privacy regulations. And third, consider the location of other cloud offerings you're using. You might have to locate your solutions in the same data center.

For a complete overview of the availability of services in the different regions, see Services .

2.2   Environments

Environments constitute the actual platform-as-a-service offering of SAP BTP that allows for the development and administration of business applications. Environments are anchored in SAP BTP on subaccount level.

Each environment comes equipped with specific tools, technologies, and runtimes that you need to build applications. So a multi-environment subaccount is your single address to host a variety of applications and offer diverse development options. One advantage of using different environments in one subaccount is that you only need to manage users, authorizations, and entitlements once per subaccount, and thus grant more flexibility to your developers.
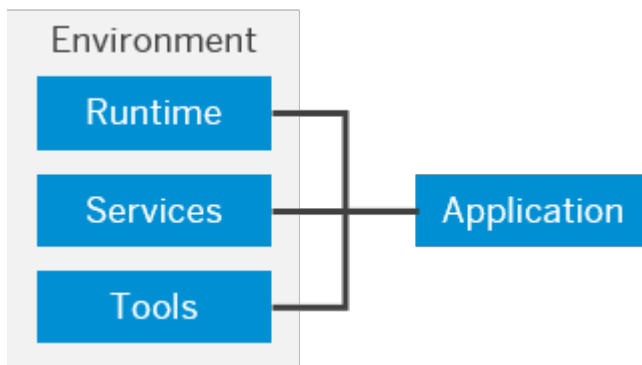
SAP offers the following environments:

- Cloud Foundry Environment
  The Cloud Foundry environment enables you to develop new business applications and business services, supporting multiple runtimes, programming languages, libraries, and services. You can leverage a multitude of buildpacks, including community innovations and self-developed buildpacks.
- ABAP Environment
  Within the Cloud Foundry environment, you can create a new space for ABAP development. This is what we refer to as the ABAP environment. It allows you to create extensions for ABAP-based products, such as SAP S/4HANA Cloud, and develop new cloud applications. You can transform existing ABAP-based custom code or extensions to the cloud.
  To plan and set up your landscape and lifecycle management in the ABAP environment, see ABAP Lifecycle Management.

- **Kyma Environment**

  The Kyma environment is a fully managed Kubernetes runtime based on the open-source project "Kyma" that allows developers to extend SAP solutions with serverless functions and combine them with containerized microservices.
- **Neo Environment**

  The Neo environment lets you develop HTML5, Java, and SAP HANA extended application services (SAP HANA XS) applications. You can also use the UI Development Toolkit for HTML5 (SAPUI5) to develop rich user interfaces for modern web-based business applications.

When you create subaccounts in a global account on feature set B, you can use different environments within the same subaccount: Choose multi-environment subaccounts to use Cloud Foundry, ABAP, and Kyma within the same subaccount. However, inside such a multi-environment subaccount, you cannot use Neo. To use Neo, you create a separate Neo subaccount. Neo cannot be combined with other environments in the same subaccount.

If you consider to create Neo subaccounts, see Migrating from the Neo Environment to the Multi-Cloud Foundation (Cloud Foundry and Kyma) to learn about the Neo environment and the multi-cloud foundation.

Environment Instances

To actually use an environment in a subaccount, you need to **enable** it by creating an instance of that environment. There are several ways to create **environment instances**:

- Using the subaccount overview page in the cockpit: choose Enable.
- Using the Service Marketplace tab in the cockpit: here you get more information, such as the available plans and links to further information.
- Using the btp CLI command `btp create accounts/environment-instance`

Related Information

SAP BTP, Serverless Runtime
Consuming SAP BTP Services from Various Environments

2.3 Commercial Models

SAP BTP offers two different commercial models for enterprise accounts.

- **Consumption-based commercial model:** Your organization receives access to all current and future services that are eligible for this model. You have complete flexibility to turn services on and off and to switch between services as your business requires throughout the duration of your contract. This commercial model is available in two flavors: Cloud Platform Enterprise Agreement (CPEA) and Pay-As-You-Go for SAP BTP.
  For more information, see What Is the Consumption-Based Commercial Model?

- **Subscription-based commercial model:** Your organization subscribes only to the services that you plan to use. You can then use these services at a fixed cost, irrespective of consumption.
For more information, see What Is the Subscription-Based Commercial Model?

For information about service availability, prices, and estimators, see https://www.sap.com/products/extension-suite/pricing.html and https://www.sap.com/products/integration-suite/pricing.html. You can also view the service catalog via the SAP Discovery Center.

> **i Note**
>
> You can use both commercial models, either in separate global accounts or in the same global acccount depending on your business needs. Contact your SAP account executive or sales representative for more information.
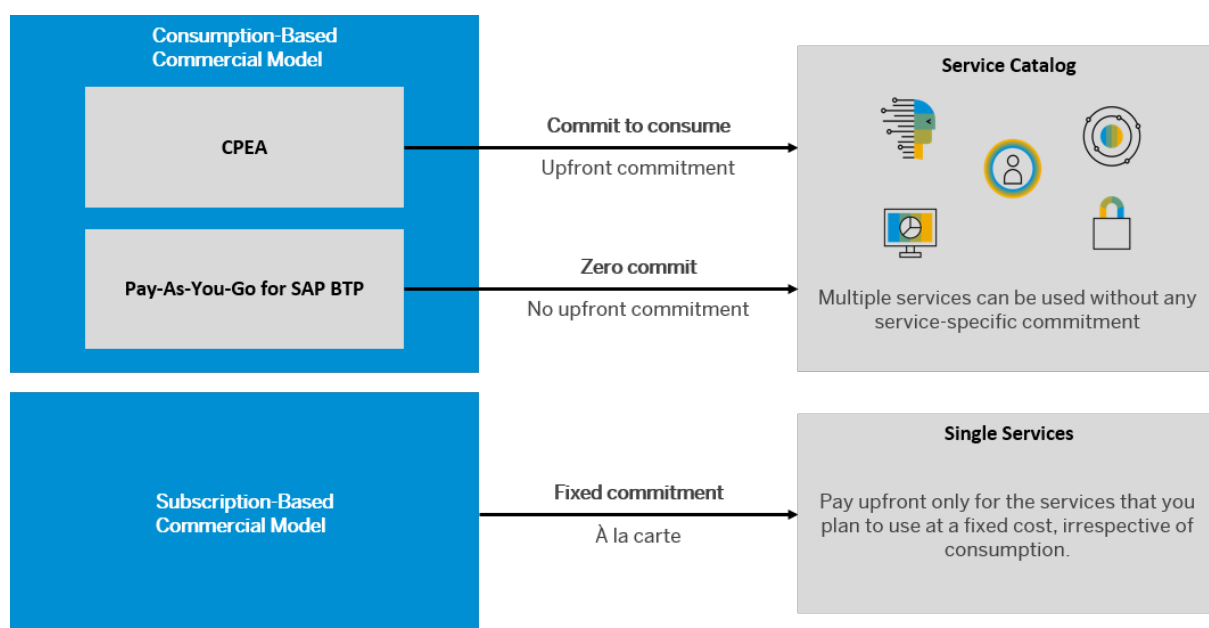
2.4 Account Model

The SAP BTP cockpit is structured according to global accounts, directories, and subaccounts.

Global Accounts

A **global account** is the realization of a contract you made with SAP.

A global account is used to manage subaccounts, members, entitlements and quotas. You receive entitlements and quotas to use platform resources per global account and then distribute the entitlements and quotas to the subaccount for actual consumption. There are two types of global accounts: enterprise accounts (paid, unless you only use free tier service plans) and trial accounts (free). The type determines pricing, conditions of

use, resources, available services, scope of the functionality that you can use, and the level of support you can receive.

Global accounts are region- and environment-independent. Within a global account, you manage all of your subaccounts, which in turn are specific to one region.
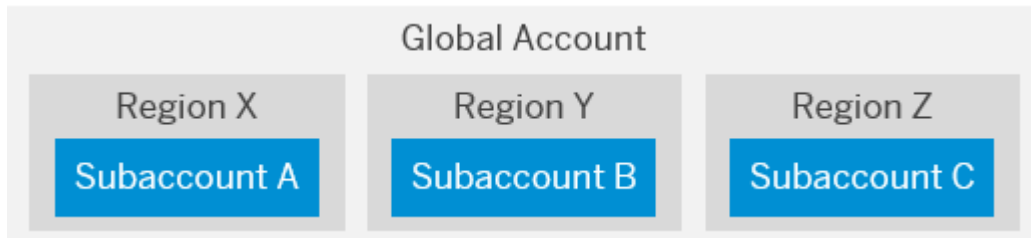
SaaS applications are usually displayed in a separate global account.

> **i Note**
>
> The following features, **directories** and **labels**, are only available if your global account is on feature set B. For more information, see Cloud Management Tools — Feature Set Overview.

Directories

Directories allow you to organize and manage your subaccounts according to your technical and business needs.

A directory can contain directories and subaccounts to create a hierarchy. Using directories to group other directories and subaccounts is optional - you can still create subaccounts directly under your global account.

You can create a hierarchical structure that is 7 levels deep. The highest level of a given path is always the global account and the lowest is a subaccount, which means that you can have up to 5 levels of directories.
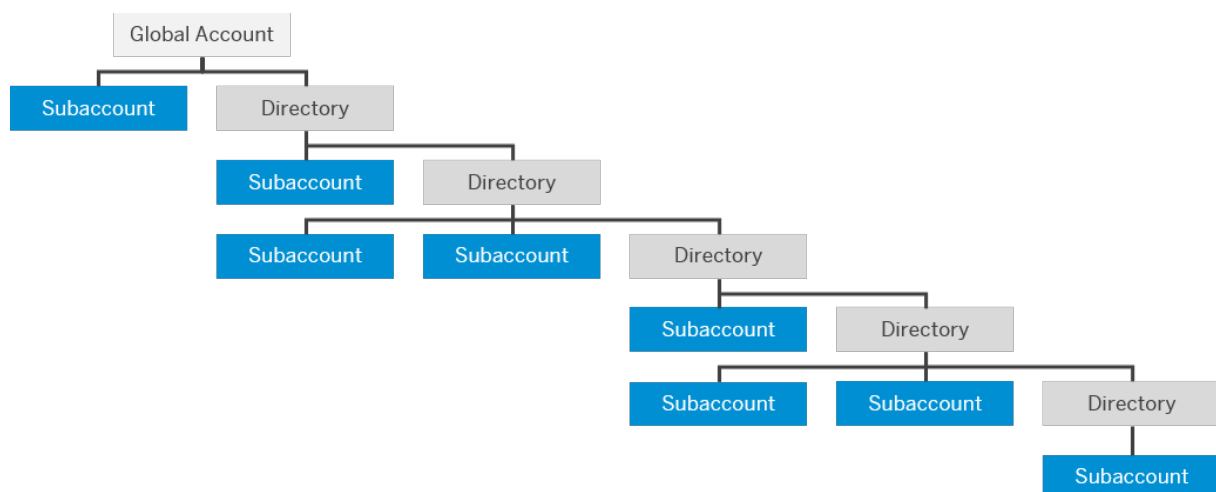
Directories allow you to:

- Group and filter directories and subaccounts

- Monitor usage and costs for contracts that use the consumption-based commercial model

In addition, you can also add the following features to your directories (optional):

- Manage Entitlements: Enables the assignment of a quota for services and applications to the directory from the global account quota for distribution to the directory's subaccounts.
When you assign entitlements to a directory, you express the entitlements and maximum quota that can be distributed across its children subaccounts. You also have the option to choose the auto-assignment of a set amount of quota to all subaccounts created or moved to that directory. Subaccounts that are already in the directory when you select that option will not be auto-assigned quota.
- Manage Authorizations: Enables a custom identity provider and/or authorization management for the directory. For example, it allows certain users to manage directory entitlements. You can only use this feature in combination with the *Manage Entitlements* feature.

Labels

Labels are user-defined words or phrases that you can assign to various entities in SAP BTP to categorize them in your global account, to identify them more easily. They let you organize and filter your directories, subaccounts, instances and subscriptions within your global account.

For more information, see Labels [Feature Set B].

Subaccounts

Subaccounts let you structure a global account according to your organization's and project's requirements with regard to members, authorizations, and entitlements.

A global account can contain one or more subaccounts in which you deploy applications, use services, and manage your subscriptions. Subaccounts in a global account are independent from each other. This is important to consider with respect to security, member management, data management, data migration, integration, and so on, when you plan your landscape and overall architecture.
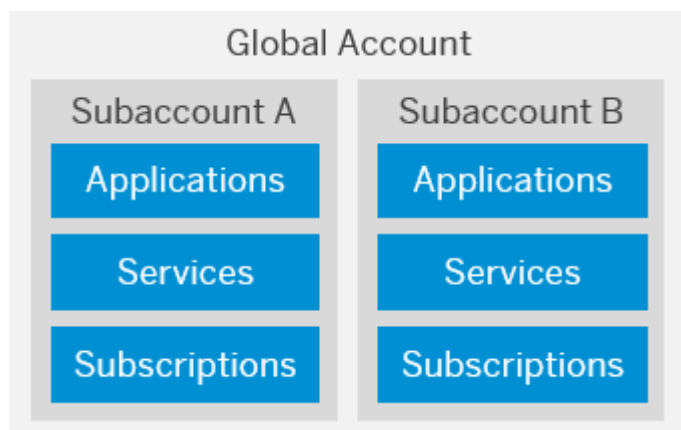
Each subaccount is associated with a region, which is the physical location where applications, data, or services are hosted. The specific region is relevant when you deploy applications and access the SAP BTP

cockpit using the corresponding cockpit URL. The region assigned to your subaccount doesn't have to be directly related to your location. You could be located in the United States, for example, but operate your subaccount in Europe.

The entitlements and quotas that have been purchased for a global account have to be assigned to the individual subaccounts.

Global accounts and subaccounts are completely independent of user accounts. For more information, see User and Member Management.

Relationship between Subaccounts, Orgs, and Spaces

When you enable the Cloud Foundry environment in one of your subaccounts, the system automatically creates a Cloud Foundry org for you. The subaccount and the org have a 1:1 relationship and the same navigation level in the cockpit (even though they may have different names). You can create spaces within that Cloud Foundry org. Spaces let you further break down your account model and use services and functions in the Cloud Foundry environment.
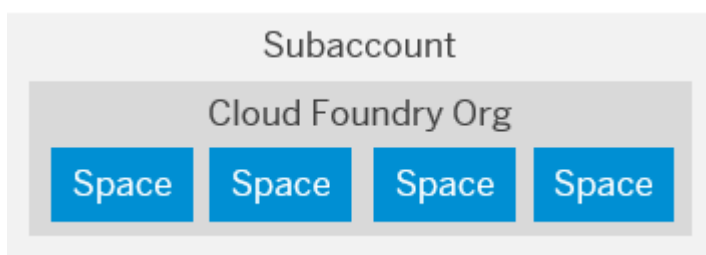
For more information about Cloud Foundry orgs and spaces, see the Cloud Foundry documentation at https://docs.cloudfoundry.org/concepts/roles.html ↗ .

For more information, see

- Account Model
- Managing Global Accounts Using the Cockpit
- Managing Directories Using the Cockpit [Feature Set B]
- Managing Subaccounts Using the Cockpit
- Setting Up Your Account Model [page 27]Setting Up Your Account Model [page 27]

2.5   User and Member Management

On the cloud platform, member management happens at all levels from global account to space, while user management is done for deployed applications.

User accounts enable users to log on to SAP BTP and access subaccounts and use services according to the permissions given to them. We distinguish between two types of users:
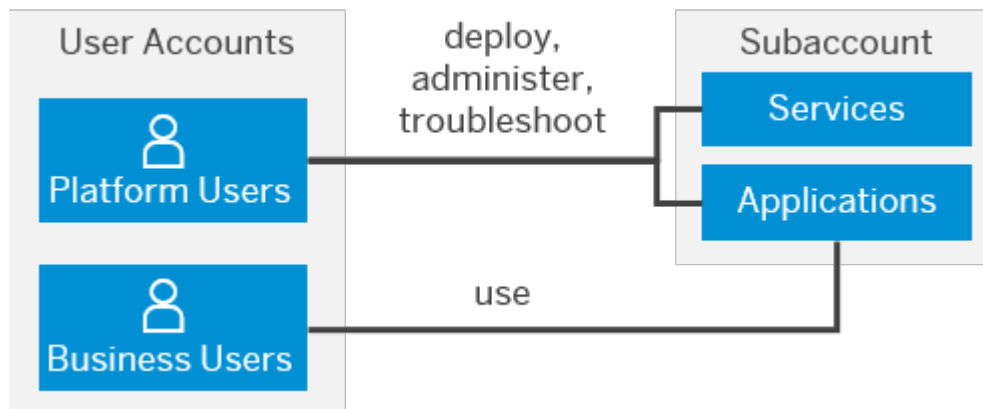
- **Platform users** are usually developers, administrators or operators who deploy, administer, and troubleshoot applications and services on SAP BTP.
- **Business users** use the applications that are deployed to SAP BTP. For example, the end users of your deployed application or users of subscribed apps or services, such as SAP Business Application Studio or SAP Web IDE, are business users.

Member management refers to managing permissions for platform users. A member is a user who is assigned to an SAP BTP global account or subaccount. Administrators can add users to global accounts and subaccounts and assign roles to them as needed. You can use predefined roles, for example the administrator role for managing subaccount members.

User management refers to managing authentication and authorization for your business users. This is only done for your deployed applications.

For more information, see User and Member Management.

2.6 Entitlements and Quotas

When you purchase an enterprise account, you're entitled to use a specific set of resources, such as the amount of memory that can be allocated to your applications.
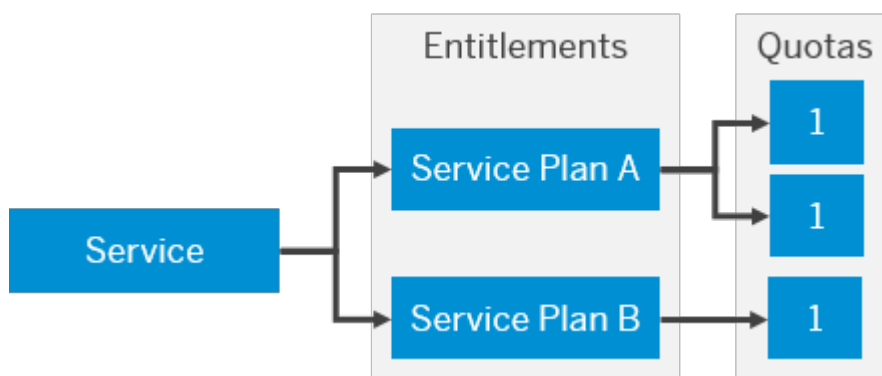
- On SAP BTP, all external dependencies such as databases, messaging systems, files systems, and so on, are **services**. In this context, multitenant applications and environments are considered services.

Each service has one or more **service plans** available. A service plan is the representation of the costs and benefits for a given variant of a particular service. For instance, a database may be configured with various "T-shirt sizes", each of which is a different service plan.

- An **entitlement** is your right to provision and consume a resource. In other words, entitlements are **the service plans** that you're entitled to use.
- A **quota** represents the numeric quantity that defines the maximum allowed consumption of a resource. In other words, **how much** of a service plan you're entitled to use.
- Entitlements are either managed at subaccount or directory level.

For more information, see Entitlements and Quotas and Managing Entitlements and Quotas Using the Cockpit.

2.7 Capabilities and Services

SAP BTP offers fast in-memory processing, sustainable, agile **solutions** and **services** to integrate data and extend applications, and fully embedded analytics and intelligent technologies.

Services enable, facilitate, or accelerate the development of business applications and other platform services on SAP BTP.

For a complete list of services and capabilities, see Services .

For more information, see Solutions and Services.

3 Shared Responsibility Model Between You and SAP

A shared responsibility model applies to SAP BTP: SAP manages the platform, whereas you develop and manage applications.
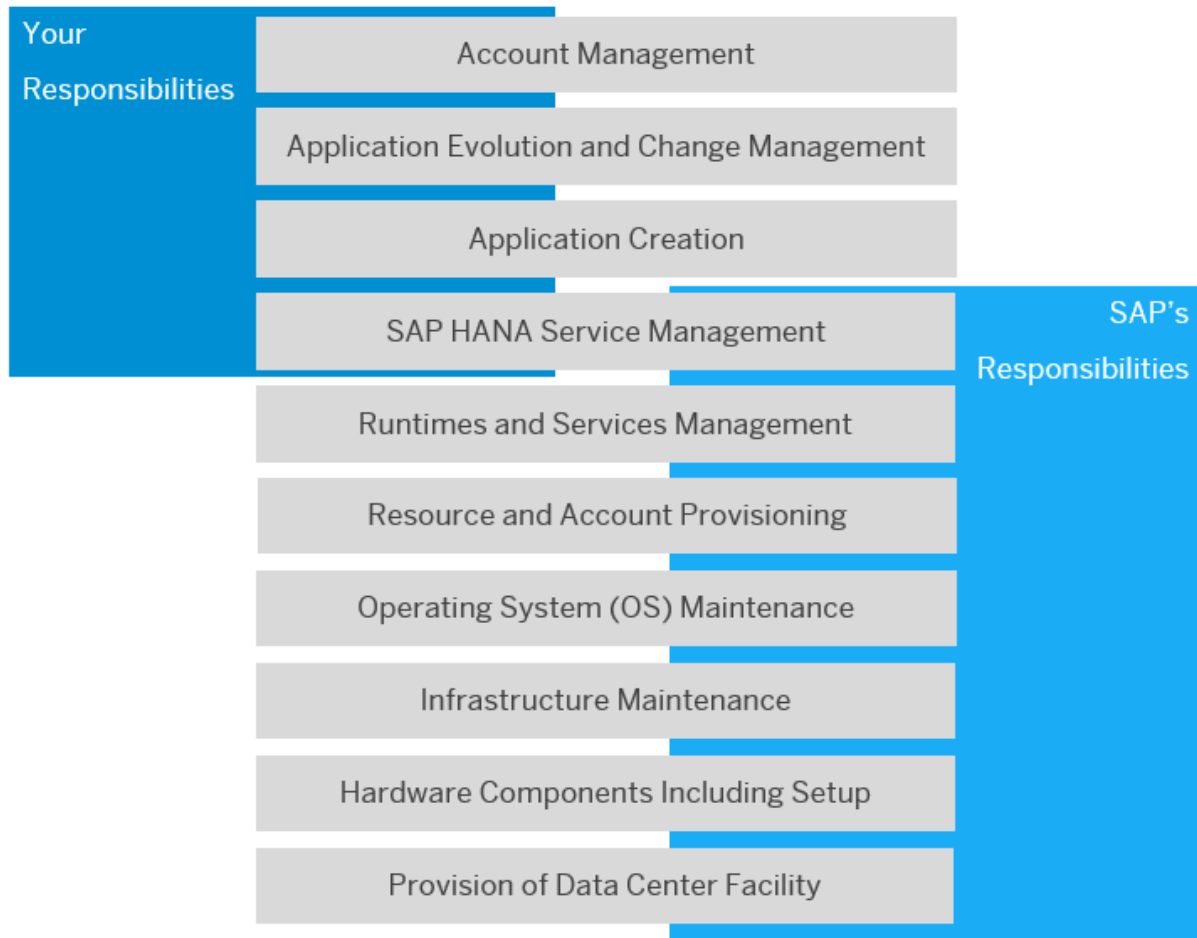
Shared Responsibilities Between You and SAP

SAP's Responsibilities

SAP is responsible for operating the overall infrastructure of SAP BTP, including monitoring, patching, applying software updates, and maintaining the infrastructure and underlying operating systems. SAP is also responsible for technical operations such as monitoring SAP BTP services, providing health check services, managing capacity, performing troubleshooting and housekeeping, implementing regular updates, and managing incidents.

SAP creates your global account and provides you with the resources and services you've purchased.

Finally, SAP is responsible for SAP HANA database platform operations, including hardware configuration management, backup and recovery, space management, security management, and providing SAP HANA data center service point revisions as self-service update options.

Your Responsibilities

As an SAP BTP customer, you must manage your global account and any subaccounts; that is, you are responsible for coming up with an account concept, creating and configuring your subaccounts based on the requirements of your development projects, and for distributing resources and services accordingly.

In addition, it's up to you to develop and operate applications. You are responsible for creating and deploying applications, managing application-specifc role assignments, integrating with existing systems and applications, monitoring and implementing health checks, and performing housekeeping, troubleshooting, and regular updates for your applications that are running on SAP BTP.

If you're using the SAP HANA service, you're required to trigger updates of SAP HANA revisions when applicable, using a self-service from SAP BTP.

For a more granular overview of the responsibilities for operating SAP BTP, see Operating Model in the Cloud Foundry Environment.

4 Getting Started Checklist

If you're new to SAP BTP, this checklist helps you ensure the implementation readiness of your development projects.

Task	Step	More Information and Troubleshooting
Fulfill prerequisites	1. Get familiar with the use cases SAP BTP covers with its service portfolio, understand basic platform concepts and your responsibilities throughout the implementation project.	Learning Journey: SAP BTP Basic Platform Concepts [page 7] Shared Responsibility Model Between You and SAP [page 19]
	2. Identify one or more initial development projects or a pilot project.	Use Cases on SAP BTP
	3. Decide which commercial model (subscription- or consumption-based) best suits your needs and sign contracts for SAP BTP.	Extension Suite and Integration Suite pricing
	4. Review Service Level Agreement.	SAP BTP Service Description Guide SAP Trust Center
	5. Receive onboarding e-mail with a link to the cockpit.	If you don't receive this email, for some reason, or have inadvertently deleted it, simply access your SAP BTP cockpit using the region-specific link Regions. SAP has activated your account prior to sending you this e-mail.
	6. Obtain a customer number (required for support tickets).	Check with your system administrator if you do not know your customer number.
	7. If you use SAP ID Service, create SAP user IDs (S-users) for all technical users who need access to your SAP BTP accounts.	Contact a user administrator in your company. If you don't want to use SAP ID Service, configure your subaccounts to use a different identity provider. i Note SAP cannot assign additional S- or P-user IDs on your company's behalf.

Task	Step	More Information and Troubleshooting
	8. Build a Cloud Administration and a Cloud Development Team (and optionally, a Cloud Center of Excellence).	Building Teams [page 23]
Get started	9. Set up your account model.	Setting Up Your Account Model [page 27]
	10. Set up your security model.	Setting Up Your Security and Compliance Model [page 47]
	11. Create an enrollment process for development projects.	Creating an Onboarding Process for Development Projects [page 25]
Implement	12. Develop, integrate, deploy, and transport your application.	Develop and Build [page 61] Developing with the SAP Cloud Application Programming Model Deploy and Deliver [page 78]
	13. Test and evaluate your application.	Integrate and Test [page 93]
	14. Go live with your application.	Go Live and Monitor [page 96]
	15. Make improvements to your application, or retire it if it's no longer needed.	Improve and Retire [page 101]

5   Set Up and Plan

Before you begin developing your applications, make sure your organizational and landscape setup is appropriate for managing their lifecycles, and consider failover to prevent breakdowns.

Creating a Governance Model [page 23]
> One of the first and most important steps of your journey to the cloud is to establish an appropriate organizational setup and corresponding governance model. A clear and well-thought-out organizational setup makes it easier for your employees to adopt agile processes.

Setting Up Your Account Model [page 27]
> The hierarchical structure between global accounts, directories, and subaccounts lets you define an account model that accurately fits your business and development needs.

Setting Up Your Security and Compliance Model [page 47]
> Applications on SAP BTP are exposed to the Internet and should therefore fulfill the highest possible security requirements to prevent unauthorized access.

Planning Failover on SAP BTP [page 58]
> Use a multi-data center setup and implement automatic failover to ensure the high availability of your applications in case of a data center outage.

5.1   Creating a Governance Model

One of the first and most important steps of your journey to the cloud is to establish an appropriate organizational setup and corresponding governance model. A clear and well-thought-out organizational setup makes it easier for your employees to adopt agile processes.

We also recommend that before you begin your development project, you create an onboarding process and a knowledge transfer process.

For example:

- Define team setups (including IT support roles and responsibilities)
- Create an onboarding process for development projects
- Create a knowledge transfer process for the involved teams
- Define support processes, operations documentation, and involved tools
- Define activities to ramp up resources and implement changes (support processes, tools, documentation)
- Clarify help desk processes and incident and change management

5.1.1 Building Teams

We recommend that you set up two types of teams: **Cloud Development Teams**, who build and operate applications, and a central **Cloud Administration Team / Center of Excellence**, who is responsible for any

account operations, the build infrastructure, for defining central governance and compliance guidelines, and for driving cloud adoption throughout your organization.

5.1.1.1   Cloud Administration Team / Center of Excellence

The Cloud Administration Team defines, sets up, and maintains your cloud landscape.

The responsibility of this team is to operate and ensure a stable and secure cloud landscape, and to enable other developers to build cloud applications. The members of this team are highly qualified experts who have development experience and experience with setting up and running a build infrastructure for continuous deployments and integration scenarios.

This team can support your development teams by providing knowledge and defining guidelines that match your company's quality and security requirements. The Cloud Administration Team should generally not be responsible for the lifecycle management of specific applications; the Cloud Development Team should take responsibility for this task.

The Cloud Administration Team can also operate as a Center of Excellence (CoE), driving cloud adoption, migration, and operation throughout your organization by providing thought leadership and guidance for resolving roadblocks. The CoE is also responsible for identifying, evaluating, and implementing use cases for the SAP BTP.

We recommend that the Cloud Administration Team / Center of Excellence (CoE) create the following documents:

- Onboarding Document [page 25]
- Security Document [page 25]
- Services Catalog [page 26]

5.1.1.2   Cloud Development Teams

The Cloud Development Teams are responsible for developing and operating the applications that run on SAP BTP.

Teams that develop on-premise applications often follow a Build-Run setup. The Build team develops the application, then passes it to the Run team, which operates and maintains it. However, this setup is not optimal in a cloud application development environment.

We recommend that Cloud Development Teams follow a DevOps approach, which means that the team both develops **and** operates applications. The team should maintain applications regularly after Go Live, and fix any issues. For example, if the team develops an SAPUI5 front end, they should verify at least every six months that the UI controls used are still supported by the latest SAPUI5 version. This doesn't require much effort, but does need to be checked to ensure the application continues running properly.

5.1.2 Creating an Onboarding Process for Development Projects

If you're planning on starting multiple development projects on SAP BTP, we recommend that you set up an onboarding process to ensure that new projects are tracked and documented properly, compliant with the security standards and guidelines you define.

Every new application that is managed and tracked by a Cloud Administration Team should have an onboarding process. Owners of new applications should fill out an onboarding document and a security template. Maintaining administrative information about all applications in one central place helps the Cloud Administration Team to keep an overview about all projects, applications, and responsibilities. It also allows the team to inform all project managers and developers about changes and updates to the cloud landscape.

Onboarding Document

An onboarding document ensures that a new application is properly onboarded and documented. Onboarding can happen via e-mail, a ticketing system, or any other tool used in your company that you find suitable. The following provides an example of the information you might want to include:

- Organization or department the application is being developed in
- Application name
- Technical application name (only alphabetic characters, no spaces)
- Business case and application description
- Planned Go-Live date
- Application owners
- Colleagues who need access to the development subaccount (if you've set up a staged development environment)
- Whether the application will be accessible by external users
- End-to-end data flow description, including connections to back-end systems
- Programming languages and technologies that will be used
- URL of the Git repository, if applicable
- Test strategy

After the document is verified, application owners and developers should get access to the development subaccount to start developing. Make sure that all developers understand that on the development subaccount, all applications can be stopped, and code can be deleted by users at any time without notice. We recommend that the Cloud Development Team keeps the relevant code and files in a repository outside of SAP BTP to avoid that applications and artifacts like git repositories (containing code) are deleted in the subaccount by accident.

Security Document

We recommend you provide a template for the security document that is filled out for every new application, which is then approved by security experts in your company before you start developing. The template should ask for detailed information, such as:

- Name of the application owners
- Scenario description and business need
- User groups, data classification, and confidentiality and integrity requirements
- Compliance with any applicable security policies
- End-to-end data flow
- Data storage
- Overview of all connected cloud and on-premise systems, used protocols, and ports
- Authentication and high-level authorization concept
- Auditing concept, logging, traceability

Services Catalog

If you're going to onboard multiple development projects, we recommend that you set up a services catalog that summarizes all services offered by the Cloud Administration Team, especially when access to the testing and production subaccounts are restricted for developers. This facilitates and speeds up the collaboration between the Cloud Administration and the Cloud Development Teams. Examples of such services include the following:

- Adding or managing destinations
- Creating build plans
- Restarting applications
- Providing read access to the testing or production subaccounts
- Creating database schemas and giving access to a specific schema

All services should be offered as templates in a service catalog that developers can fill out, for example, in a ticketing system.

We recommend that the Cloud Administration Team makes the consumption of services as easy as possible, or even automates it using SAP BTP APIs. The team may want to, for example, write scripts for database schema creation and access management. The Lifecycle Management API lets the Cloud Development Team restart an application in the production subaccount without having access to it and without affecting any other application. For more information, see the Lifecycle Management API .

Related Information

API Documentation

5.1.3 Creating a Knowledge Transfer Process

Carefully plan for sufficient enablement of all stakeholders, and ensure that the knowledge gained throughout your application lifecycle is shared with others.

Before you start your development project, we recommend that you do the following:

- Ensure that the Cloud Administration Team, which should consist of skilled and experienced technology experts, documents and shares their knowledge with existing and new colleagues.
- Set up training and enablement sessions to get everyone on board.
- Create and promote a dedicated communication channel, such as SAP Jam Collaboration, to share lessons learned or provide other developers with guidance and recommendations. For more information, see SAP Jam Collaboration🔗.

5.2 Setting Up Your Account Model

The hierarchical structure between global accounts, directories, and subaccounts lets you define an account model that accurately fits your business and development needs.

Once you've signed your commercial contract with SAP, you'll receive access information and credentials for your global account. Use this account to manage the resources and entitlements for your development projects; it's the realization of your commercial contract with SAP. You'll also receive a bill for all resources used in your global account. If you require multiple global accounts for compliance or other reasons, you'll need to sign a dedicated contract for each one.

> **i Note**
>
> SAP is currently migrating all global accounts from cloud management tools feature set A to the renovated cloud management tools feature set B. We're doing this migration as a phased rollout, which is why cloud management tools feature sets A and B currently coexist. One big change that came with feature set B is a renovated account model: With feature set B, you can use **directories** to group and manage subaccounts inside your global account. For more information, see Cloud Management Tools - Feature Set Overview. Please make sure to understand if your global account is already on feature set B.

Within a global account that is on feature set B, you can create up to five levels of directories to further structure your global account according to your business or technical needs. To actually develop and deploy applications, to use services, and to subscribe to business applications provided by SAP, you need to create subaccounts. Directories are optional, but highly recommended as they offer a way to organize your subaccounts and to manage entitlements or users for an entire group of subaccounts. This means you can distribute the resources that are assigned to your global account across directories, and from there to the subaccounts inside a directory. Directories and subaccounts let you structure a global account according to the requirements of your organization and projects with regards to users, authorizations, and quotas.

Tools For Account Administration

The most important step in account model setup is to work out a concept that meets your business and technical requirements. The account models in this guide are designed to help you accomplish this. However, when it comes to acctually creating directories and subaccounts in your global account, you can choose between different experiences:

- Account Administration in the Cockpit
- Account Administration Using the SAP BTP Command Line Interface (btp CLI) [Feature Set B]

- Account Administration Using APIs

Related Information

5.2.1 Using Subaccounts to Create a Staged Development Environment

The number of subaccounts you create, and for which purpose, depends on your organizational setup and your use case.

> → Recommendation
>
> In general, we recommend that you create at least three subaccounts to set up a staged development environment, including one subaccount each for development, testing, and production.
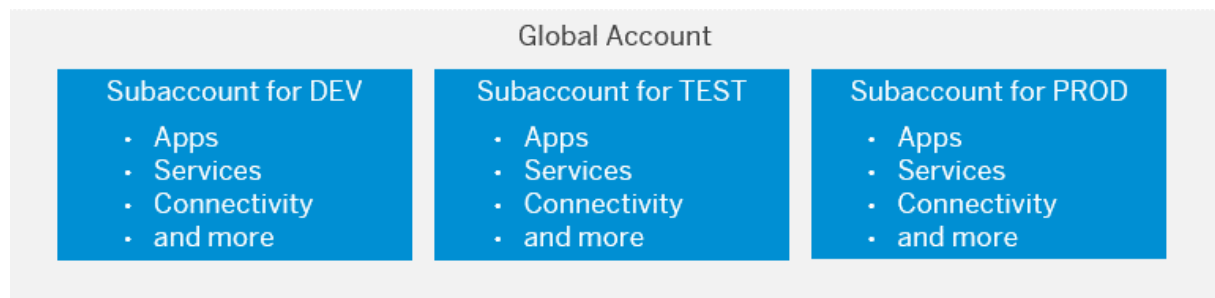
- Development – for development purposes and for testing individual increments in the cloud.
- Testing – for integration testing and testing in production-like environment prior to making it publicly available, to ensure quality delivery. In highly DevOps-driven companies, this subaccount is also used for production applications, as testing occurs in the development subaccount.
- Production – for production applications.

Considerations For Creating Your Account Model

Although we recommend that you use subaccounts to create a staged development environment, you can also create subaccounts to do the following:

- Separate development scenarios and projects to allow for easier configuration, for example, with regard to access restrictions.
- Separate the work of different development teams.
- Restrict access to applications and their administration, for example, by setting up "high-security" subaccounts with restricted access, or creating separate subaccounts to connect with your different back-end systems.
- Share an SAP HANA database in one subaccount with similar projects managed in other subaccounts.

Keep in mind the following considerations when creating different subaccounts:

- Connections to on-premise systems must be made separately for each subaccount, which also means more work for your Cloud Administration Team. However, it might be easier for you to shut down all integration paths for a project if you've maintained them all in one subaccount. This also lets you control which application uses which on-premise connections.
- When choosing a region for your subaccount, consider that the region should be close to your customers' geographic location to reduce network latency. In extension scenarios, choose a region that's close to the systems involved. Also, consider any legal requirements and the load distribution when choosing a region.
- If your S/4HANA tenants need to be segregated for legal or regulatory reasons (e.g. for the segregation of subsidiaries of a company), then you should use different subaccounts for their extensions.
- If the DevOps or operations teams for applications within one subaccount are completely separate, you should consider creating separate subaccounts for them for better maintainability.

For information about creating subaccounts in the SAP BTP cockpit, see Create a Subaccount.

Special Considerations for the Cloud Foundry Environment

When you enable the Cloud Foundry environment in one of your subaccounts, the system automatically creates a Cloud Foundry org for you. The subaccount and the org have a 1:1 relationship and the same navigation level in the cockpit (even though they may have different names). You can create spaces within that Cloud Foundry org. Spaces let you further break down your account model and use services and functions in the Cloud Foundry environment.

You can use both subaccounts and spaces to develop applications and to use services. You must therefore decide, for example, whether to create different subaccounts or spaces within one subaccount to set up a staged development environment. Consider the following:

- Think of a subaccount as a tenant: Data access and data visibility segregation is done on subaccount level, not on application or Cloud Foundry space level. Keep in mind that services that are consumed by every

app within a subaccount will gather messages from all of these services. If those should not be visible across applications, you need to create different subaccounts.

- In general, we recommend that you create different subaccounts for a staged development environment, as shown below. This allows for dedicated user management between the different stages, as well as for dedicated data management in elastic services, such as SAP IoT Application Enablement.
- You can then create a dedicated space for each application, extension, solution, or other project within these subaccounts if you don't need a dedicated user management for these applications and projects.
- You can monitor the consumption of resources in your global account only per subaccount, directory, or Cloud Foundry space. Not per application. To monitor the resources consumed by a specific project, department, or application, create a dedicated subaccount, directory, or space for them.
- Accurate billing is only possible for global accounts. For the consumption-based model, you can calculate costs according to usage, but note that this is only approximate. See Monitoring Usage and Consumption Costs in Your Global Account.
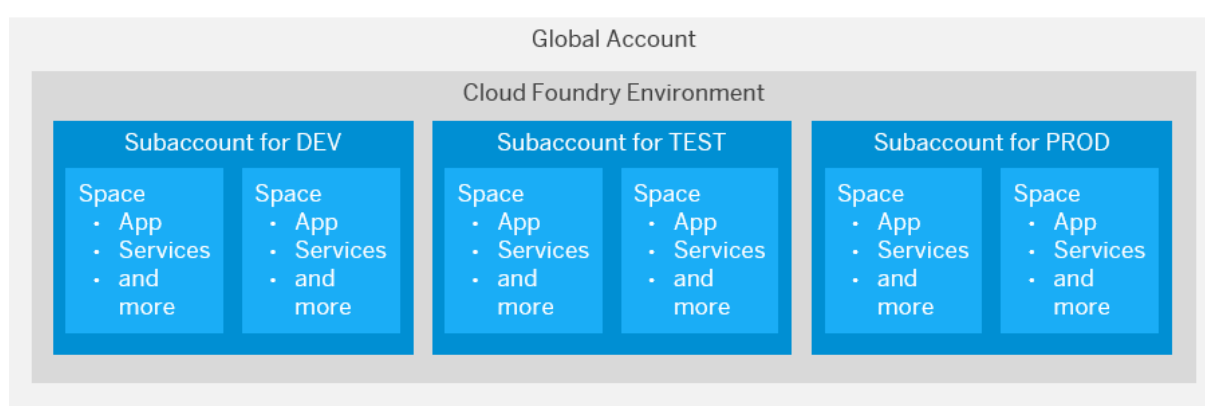
To decide whether to create separate subaccounts or separate spaces within the same subaccount, consider the different configuration possibilities, available for subaccounts and spaces:

Configuration	Subaccount	Space
Configure your own group of business users	Yes	No
Configure a Cloud Connector tunnel	Yes	No
Configure roles and trust	Yes	No
Assign quotas	Yes (mandatory)	Yes

See Create Spaces.

Example Account Models

Consider the following account scenarios for creating your own account model:

- Account Models with Subaccounts [page 31]
- Account Models With Directories and Subaccounts [Feature Set B] [page 40]

5.2.2 Account Models with Subaccounts

Account Models 1 - 5 show ways to structure your global account into subaccounts. Note that all of them are just examples. They are not mutually exclusive and you can adapt them to your own needs.

Once your global account is migrated to cloud management tools, feature set B, you will be able to make use of one more hierarchical level within your global account: While account models 1 - 5 are still possible, with feature set B, you can additionally use directories and custom properties to group subaccounts. For more information, see Cloud Management Tools — Feature Set Overview and Account Models With Directories and Subaccounts [Feature Set B] [page 40].

Related Information

Account Models With Directories and Subaccounts [Feature Set B] [page 40]

5.2.2.1 Account Model 1: Create a Staged Development Environment Combining the Cloud Foundry and the Neo Environment

In this scenario, the IT department of an organization creates a staged development environment by creating three subaccounts in both the Cloud Foundry and the Neo environment, in order to use SAP Web IDE Full-Stack, which only runs in the Neo environment.

> → Tip
>
> Check out SAP Business Application Studio, which is a modern development environment that runs in the Cloud Foundry environment, and which makes using the SAP Web IDE Full-Stack in the Neo environment unnecessary: SAP Business Application Studio.

Dedicated spaces for specific applications or projects are created in each of the subaccounts in the Cloud Foundry environment.
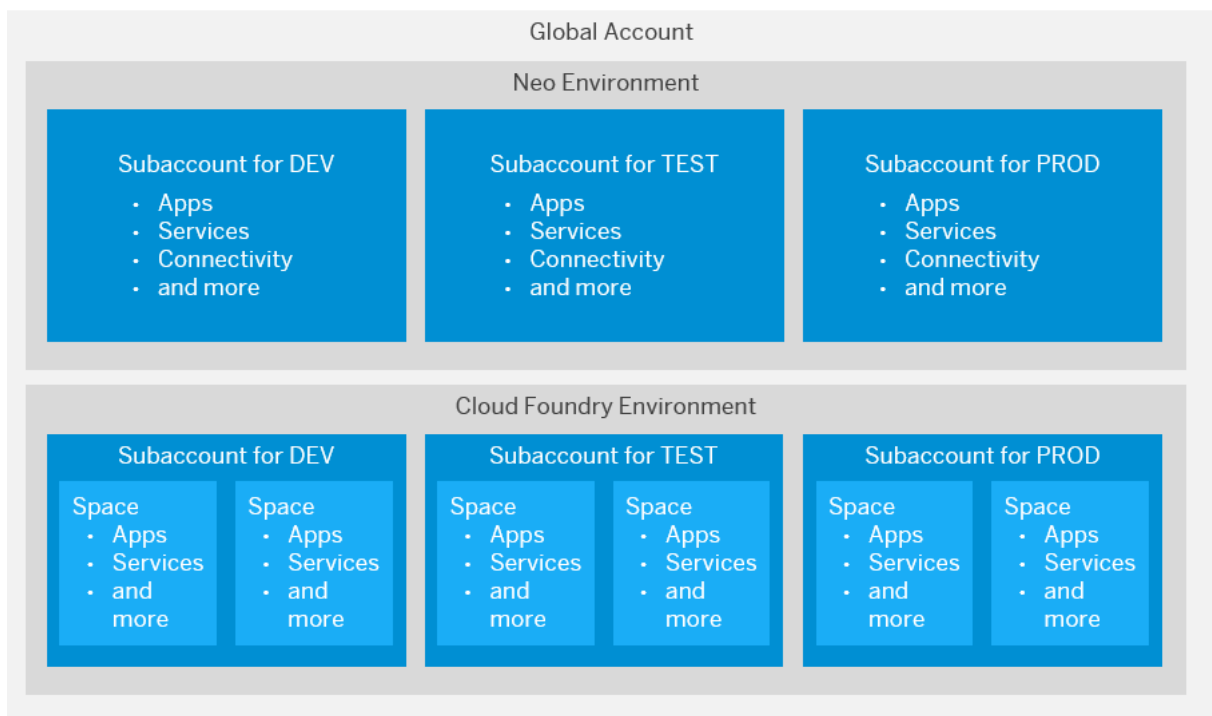
If you need to deploy an application to a space within the Cloud Foundry environment, you can use SAP Web IDE Full-Stack, which runs in a subaccount in the Neo environment. In this case, we recommend that you create three subaccounts in the Cloud Foundry environment to separate the three different development stages. In addition, create one subaccount in the Neo environment that you use for developing using the SAP Web IDE Full-Stack. You can connect that subaccount with your Development and Testing subaccounts in the Cloud Foundry environment by configuring trust and destinations.

> i Note
>
> This setup works only if you use the same identity provider for the subaccounts in the Cloud Foundry and in the Neo environments. For more information, see Principal Propagation from Cloud Foundry to Neo

Related Information

Principal Propagation
Configure Trust
Configure Destinations from the Cockpit
When to Use Which Account Model [page 37]

5.2.2.2 Account Model 2: Use a Subaccount as a Base Template for Development Projects in the Neo Environment

In this scenario, the IT department creates one subaccount that serves as a base template. If new subaccounts for a development project are required, the IT department clones the base template, thereby copying its configuration settings to the new subaccounts.

The owners of the development projects are responsible for application lifecycle management. As shown below, all new subaccounts are cloned from the base template, regardless of whether a project requires three subaccounts for each development stage, or whether only one DEV subaccount is needed (for example, for a proof of concept).

This account setup is especially well suited for organizations that rely heavily on a DevOps methodology with independent development projects.

> **i Note**
>
> This account setup might require a high level of maintenance and governance efforts. For instance, it is important that the responsible teams of each project landscape stay tightly aligned. Since each team has full access to their subaccounts, it is important to define and follow guidelines for all projects, such as security requirements, access guidelines, governance processes, and so on.
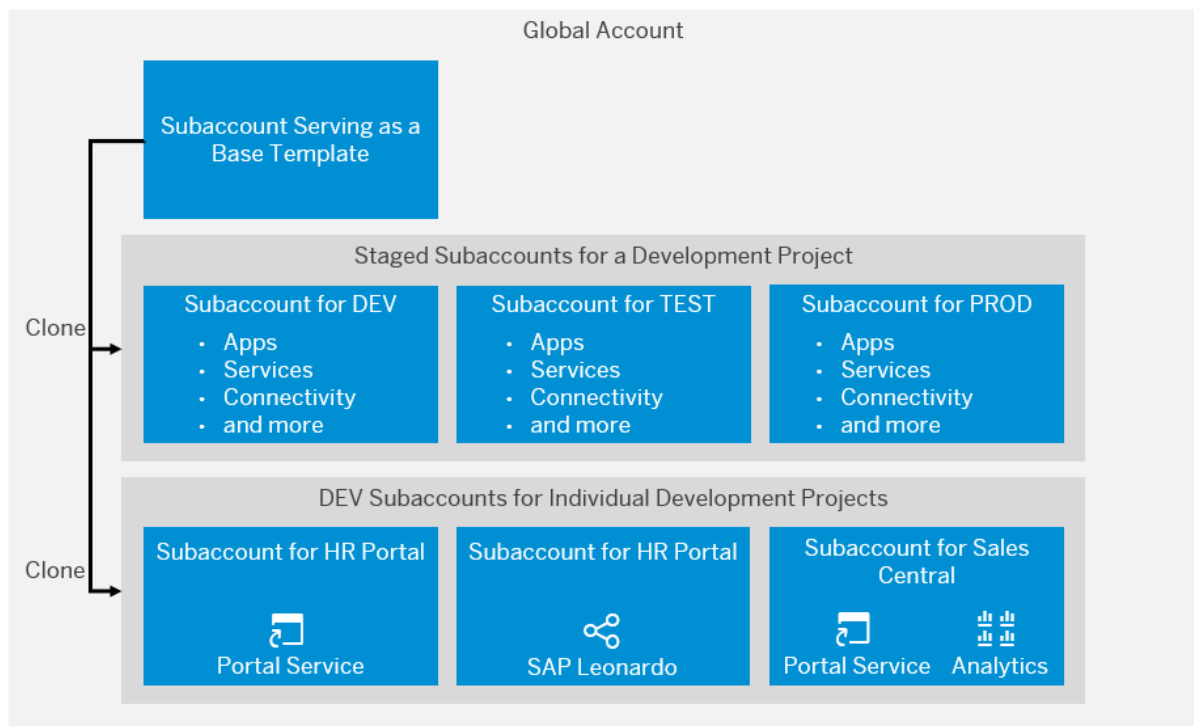
5.2.2.3 Account Model 3: Use Separate Subaccounts for Data and API Management

In this scenario, the IT department of an organization manages separate subaccounts for data isolation and for reuse of API management and connectivity. The department also creates dedicated (staged) subaccounts for development projects.

Access to back-end systems is managed centrally by the IT department via API management in three dedicated subaccounts. Every new development project uses separate subaccounts (one each for Development, Test, and Production) that consumes the APIs that are managed in these subaccounts via destinations. In addition, SAP HANA databases are deployed in two dedicated subaccounts – Development/Testing and Production – and shared across different subaccounts.

The advantages of this model are that it allows you to scale quickly while ensuring that connections to your back-end systems are always established via APIs; it also keeps costs low (due to the sharing of the SAP HANA databases across subaccounts).
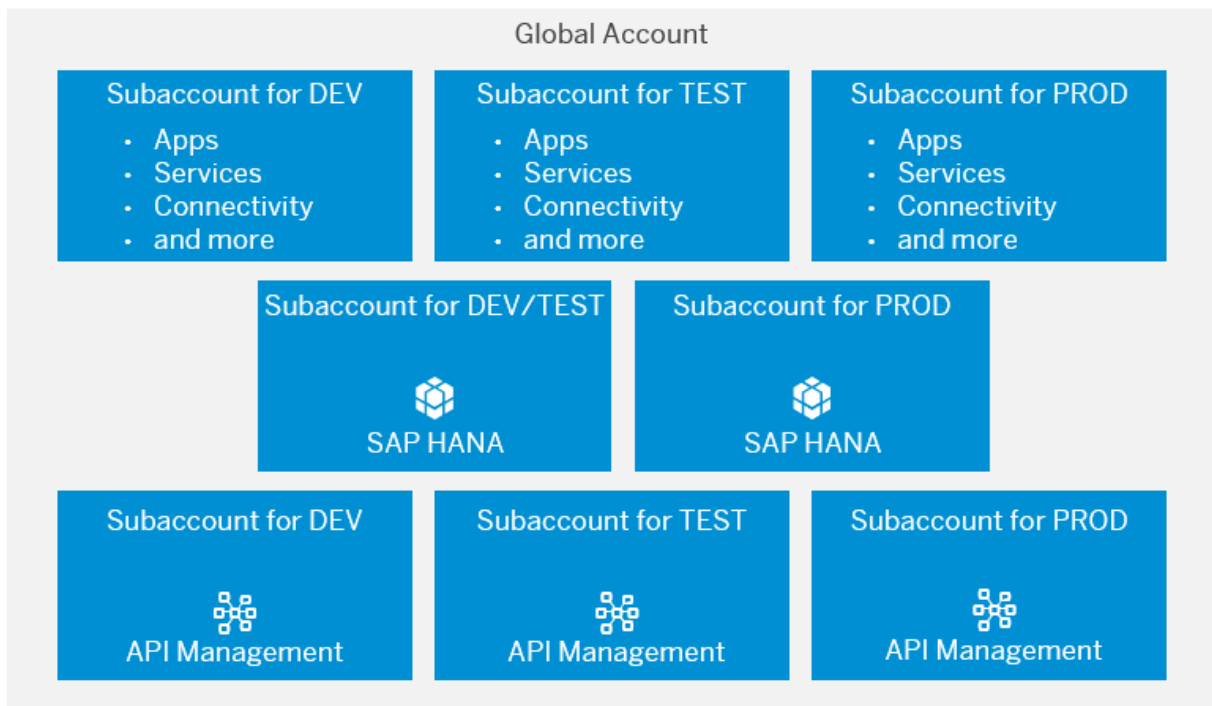
5.2.2.4 Account Model 4: Create a Staged Development Environment Using Continuous Integration / Continuous Delivery

In this account model, a dedicated Development subaccount is created for each development project. Applications that are developed in these subaccounts are consolidated, tested, and published in one single Testing and Production subaccount.

This account model is especially well suited for companies with a focus on continuous integration and continuous delivery, as it lets every development project decide separately about their environment. A central Testing and Production subaccount ensures a high degree of governance, data security, and compliance.
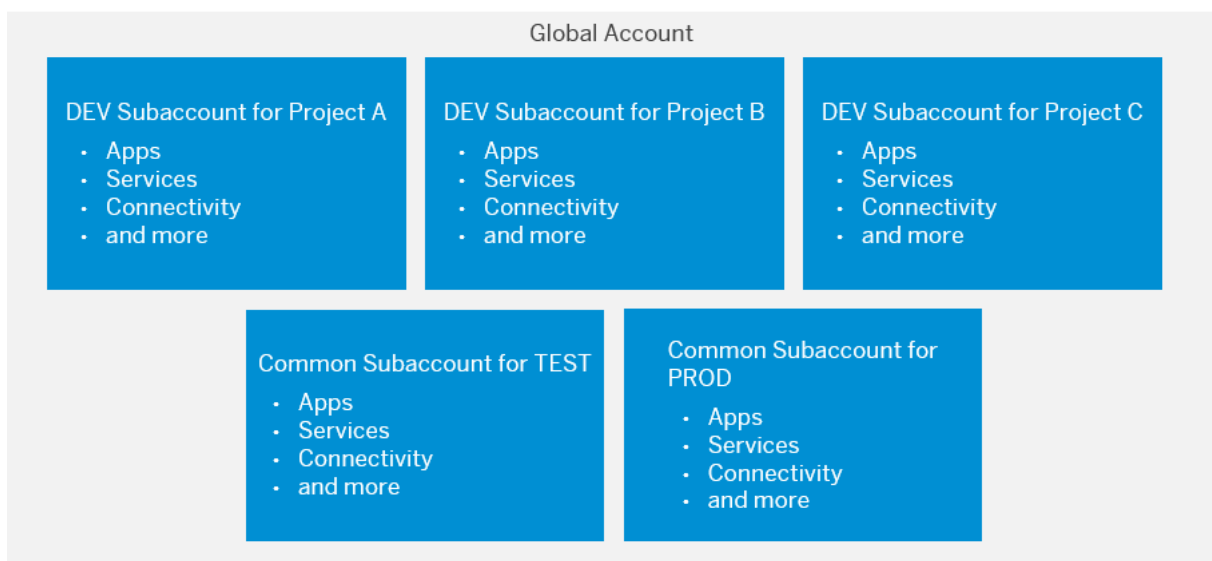
In the Cloud Foundry environment, consider creating separate Development spaces in one subaccount and a dedicated Testing and Production space in a Testing and Production subaccount, respectively.

5.2.2.5 Account Model 5: Create a Staged Development Environment Per Functional Area

In this scenario, the IT department creates separate Dev, Test, and Prod subaccounts for each functional area: In our example, HR, IT, and Sales each get their own subaccounts for development, test, and production.

With a separate account landscape per functional area, you can use multiple instances of the same SaaS subscription (e.g. in case you need to keep data or user access separated). For example, HR subaccounts would use an internal identity provider, whereas for public or external scenarios, an external identity provider could be used.

Using this account model, you can distribute the subaccount administration to several teams, which allows for easy scaling as the number of cloud projects grows while still having a manageable amount of maintenance and governance efforts (unlike with account model 2). If possible, consider assigning responsible colleagues to each group of three subaccounts, i.e. to each account landscape.

A SaaS application can only be used once within a subaccount, so if each functional area has their own subaccount (for dev, test, and prod), you won't encounter limitations as to using the same SaaS offering in different projects. Still, in the Cloud Foundry environment, you can separate different projects within a subaccount into different spaces.

This account model can also be combined with the structuring approach of account model 2. In addition to the structuring according to functional areas, you can create separate dev, test, and prod subaccounts for large projects that use different components or services and have a completely separated support process and operations model than your other projects.

In the Cloud Foundry environment, if you use a central Fiori Launchpad, and you want to deploy applications that use it in a different subaccount, use the Launchpad Module. If you want to create applications in the same subaccount as the Fiori launchpad, we recommend to deploy them to the HTML5 repository as described in Developing HTML5 Applications.

Related Information

5.2.2.6   When to Use Which Account Model

Determine which account model with subaccounts is the most appropriate for your needs.

The tables below provide requirements with regards to project and team setup as well as the services and resources involved. You can compare them against your own requirements and see which account model most closely matches. Please remember that the four account models are a result of a best-practice approach. They are not mutually exclusive and you may want to use a combination of approaches, if, for example, you have different projects with different requirements.

Project Setup

Your Require-ments ...	Account Model 1	Account Model 2	Account Model 3	Account Model 4	Account Model 5
You follow the De-vOps methodology and have many au-tonomous projects.		X		(X)	(X)
You need different stages for your project or sce-nario.	X	(X)	(X)	(X)	X
You have a highly centralized project landscape with many projects us-ing the same tech-nologies (for ex-ample, Fiori devel-opment for a Fiori launchpad).	X				X
You want to share more expensive re-sources (HANA databases, Inte-gration service tenants) with mul-tiple projects.	X		X	(X)	(X)

Team Setup

Your Requirements ...	Account Model 1	Account Model 2	Account Model 3	Account Model 4	Account Model 5
You have a large number of developers, with several different project admins responsible for them.		X		X	(X)
You have many different development teams, but only one single production environment for which the teams develop applications.				X	
You have different development teams, each of which works independently, meaning they should not be able to, for example, accidentally break the other team's developments.		X	X	X	X
Your development teams on SAP BTP are coming from different organizational units in different geographic locations and you want to track them separately.		X	X	X	X
You want to ensure that no developer or project admin can assign privileges to themselves for services or applications.		X	X	X	

Your Require-ments ...	Account Model 1	Account Model 2	Account Model 3	Account Model 4	Account Model 5
You have stringent security require-ments. For example, you don't want all developers to have access to all Git re-positories.		X	X	(X)	(X)
You want to allow access to on-prem-ise systems via Cloud Connector only for selected projects, while re-stricting it for the applications of other projects.		X	(X)		(X)

Services and Resources

Your Require-ments ...	Account Model 1	Account Model 2	Account Model 3	Account Model 4	Account Model 5
You want to assign quotas to different projects and tightly control that the projects don't exceed their re-sources.		X	X		(X)
You want to allow the usage of a par-ticular service only for selected projects.		X	(X)		(X)

Your Require-ments ...	Account Model 1	Account Model 2	Account Model 3	Account Model 4	Account Model 5
You want to easily monitor which services are con-sumed by which projects. For ex-ample, you might want to do internal billing based on the resource con-sumption of the in-dividual projects.		X	X	(X)	X
You want to re-strict the exposure of on-premise sys-tems through cen-tral middleware (Integration serv-ice) or an API layer (API Manage-ment).			X		

- X: Account model meets this requirement.
- (X): Account model meets this requirement with restrictions.

5.2.3 Account Models With Directories and Subaccounts [Feature Set B]

With cloud management tools feature set B, we're introducing a more flexible account structure with directories.

> i Note
>
> The content in this section is only relevant for cloud management tools feature set B. For more information, see Cloud Management Tools - Feature Set Overview.

Directories

Apart from structuring your global account into several subaccounts, you can group individual subaccounts into directories to manage, operate, and analyze such groups of subaccounts together. One global account can contain up to five levels of directories, can contain n subaccounts each. All examples of using directories are

based on and only use one level of directories.

Here are a few example use cases where directories help you manage your subaccounts:

- Administrative reasons: Structure your global account according to the responsibilities within your organization. For example, give each subsidiary, department, or LOB their own directory.
- Billing purposes: Structure your global account into directories for accounting purposes.
- Geographical separation: Group subaccounts based on geographical locations to manage different local regulations or to improve network performance for groups that are located together.
- Business scenario: Group subaccounts that belong to the same business scenario or according to other business needs. This gives you the option to control each business solution separately.
- Resource limitations: Use the directory structure to control access to resources, limit usage by generating separate usage and cost reports, or define usage limitations, to give more resources to critical directories, or to enable different monitoring per directory. Or you could structure the subaccounts according to usage limits in different landscapes.
- Technical reasons: Structure directories and subaccounts according to technical limitation and then add labels for virtual grouping or vice versa.

Labels

Directories and subaccounts can be tagged with **labels** for easy organizing and filtering. Also, labels can be attached to other entities in your global account, such as environment instances and service instances. Think of labels as a way of virtually grouping subaccounts or directories that belong together or for which you want to create reportings.

Here are some examples of simple labels:

- Landscape: Dev, Test, Prod
- Department: HR, IT, Finance, Sales
- Cost Center: 000001234, 000001235
- Flagged for deletion
- Important

> → Tip
>
> Directories and labels offer two different ways of structuring subaccounts into groups. We recommend to use directories as the primary structuring mechanism and to use labels to build virtual groups of subaccounts or directories for which you need to create reports.
>
> The below-mentioned account models for feature set B work for all environments. A subaccount with feature set B can either be a multi-environment subaccount where Cloud Foundry, Kyma, ABAP can be enabled, or it can be a Neo environment subaccount. For the account structure, there's no difference. Inside a subaccount, you can further structure your projects depending on the environment, for example by creating Cloud Foundry spaces.

Related Information

5.2.3.1 Account Model 6: Create Directories Per Functional Area

This account model is based on Account Model 5 for feature set A, but it has directories as additional structural elements, i.e. a separate directory for each functional area and one for central services that are used by all.

Within each of those directories, three subaccounts (for development, test, and production) are created. For each directory, the functional area can use their own identity provider and manage their entitlements. Additionally, you can make use of labels, for example for the person responsible, cost center, or other aspects that you need for reportings later on.
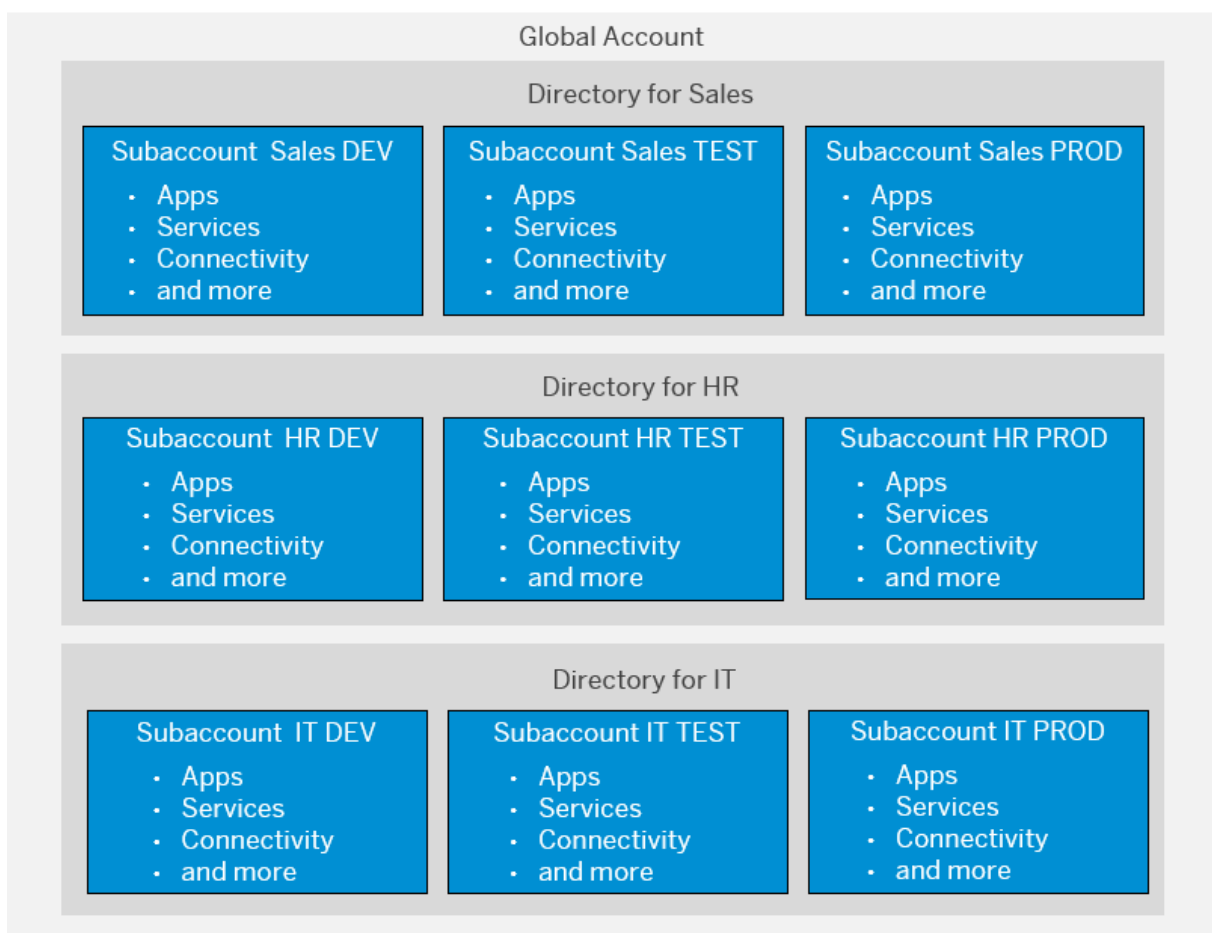
Example of naming conventions for this account model in the Cloud Foundry environment

To ensure that there are no conflicts with special characters, we recommend to stick with lower-case letters and hyphens, and not to use spaces in names. This way, you can name a subaccount, its subdomain, and its Cloud Foundry org exactly the same. Cloud Foundry spaces should be named identically across the three tiers (dev, test, prod).

Directory	mycompany-hr			my-company-sales			my-company-IT		
Subaccounts and subdomains	mycompany-hr-dev	my-company-hr-test	my-company-hr-prod	my-company-sales-dev	my-company-sales-test	my-company-sales-prod	my-company-IT-dev	my-company-IT-test	my-company-IT-prod
Cloud Foundry Orgs (same name as subaccounts)	mycompany-hr-dev	my-company-hr-test	my-company-hr-prod	my-company-sales-dev	my-company-sales-test	my-company-sales-prod	my-company-IT-dev	my-company-IT-test	my-company-IT-prod
Spaces	project-portal activity-recording	project-portal activity-recording	project-portal activity-recording	customer-acquisition sales-support	customer-acquisition sales-support	customer-acquisition sales-support	it-support central-service	it-support central-service	it-support central-service

Related Information

[Account Model 5: Create a Staged Development Environment Per Functional Area [page 35]](#)

5.2.3.2 Account Model 7: Create Directories Per Location

In this account model, you create different directories for geographical areas. Additionally, for example, you can add labels to subaccounts that belong to the same departments in those locations. Alternatively, you could nest additional directories inside these ones.

Global Account

Directory EMEA

Subaccount EMEA DEV	Subaccount EMEA TEST	Subaccount EMEA PROD
• Apps • Services • Connectivity • and more	• Apps • Services • Connectivity • and more	• Apps • Services • Connectivity • and more

Directory AMER

Subaccount AMER DEV	Subaccount AMER TEST	Subaccount AMER PROD
• Apps • Services • Connectivity • and more	• Apps • Services • Connectivity • and more	• Apps • Services • Connectivity • and more

5.2.3.3   Account Model 8: Create Directories Per Subsidiary

In this account model, you create directories for each subsidiary of your company. Additionally, you canadd labels, for example, for cost centers or owners of the individual subaccounts or directories.
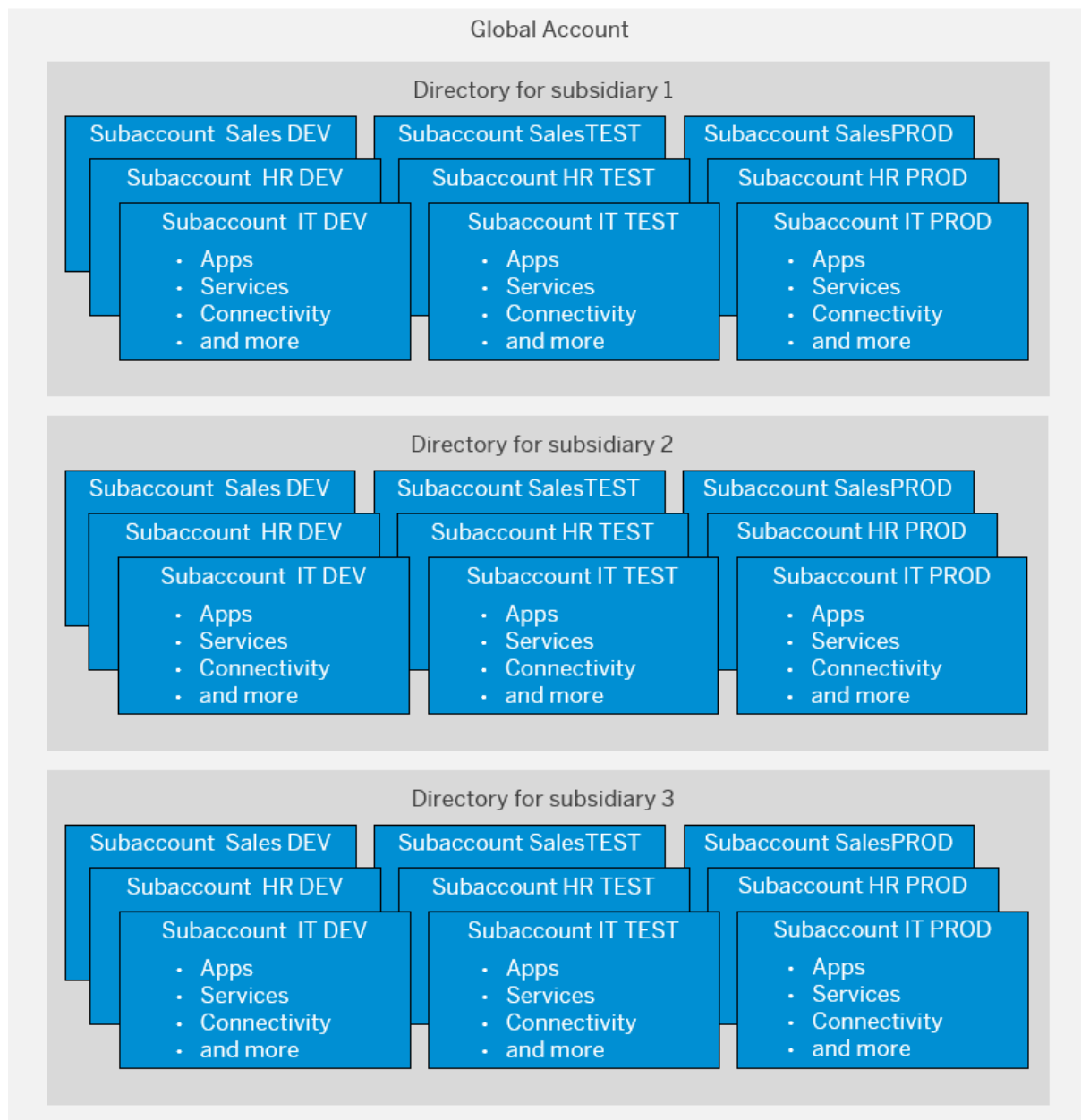
5.2.3.4 Checklist for the Account Model Setup

Independent of the account model you choose, we recommend to go through the following steps and define guidelines for your development teams.

Task	Step
Fulfill prerequisites	1. Evaluate your business and technical needs and define an account model that fits the requirements of your company. Ensure that the account model is suitable for all areas in your company.
	2. Define account hierarchy and guidelines and roll them out to a few pilot project managers to get their feedback.
Define account standards and rules	3. Create a template for new directories. See below.
	4. Define naming conventions, for example: • To ensure that there are no conflicts with special characters, we recommend to stick with lower-case letters and hyphens, and not to use spaces in names. • Give the directory a descriptive name that starts with your company / global account name. • There is a 1:1 relationship between Cloud Foundry org and a subaccount, so you should give them identical names. • Since there is no connection between spaces in differet subaccounts, we recommend to give identical names to the dev, test, and prod versions of projects and spaces.
	5. Define labels and values according to the reports you want to create.
	6. Define rules for quota limitations.

Template for New Directories

We recommend that you create a process for the creation of new directories. Here is an example template for new directories that you can use or adapt to meet your own requirements:

New Directory	
Name	Refer to your naming guidelines.
Owners	We recommend to appoint at least two owners.

New Directory

Description	Describe the developer audience, which LoB or department do they belong to, what types of applications will be developed, which environments should be used, subscriptions that will be used.
Cost center	Define accounting requirements.
Enrollment	Describe how projects can enroll in your directory.

Related Information

Labels [Feature Set B]

5.3 Setting Up Your Security and Compliance Model

Applications on SAP BTP are exposed to the Internet and should therefore fulfill the highest possible security requirements to prevent unauthorized access.

Security Concepts [page 48]
>   The level of security you implement may vary depending on your use case and your company's general security requirements. However, there are a few best practices that we recommend, regardless of your implementation.

Setting Up Authentication [page 49]
>   SAP BTP provides various options for implementing user authentication.

Setting Up Authorization [page 54]
>   Users in an application require different authorizations because they work in different jobs. For more information, see **Authorization Entities**. SAP BTP provides various options for implementing user authorization.

Setting Up Identity Propagation [page 56]
>   SAP BTP provides various options for identity propagation.

Complying with Data Protection and Privacy Requirements [page 57]
>   All companies must respect data protection and privacy rights of individuals. Data protection and privacy most likely influences the architecture and functionalities of your application and should be considered as early as possible in the development process.

Giving Access Rights to Platform Users [page 57]
>   If you've set up a staged development environment using different subaccounts or spaces, such as for development, testing, and production, we recommend that you grant the Cloud Development Team access to development subaccounts and spaces, but that you grant only the Cloud Administration Team access to the testing and production subaccounts or spaces.

5.3.1 Security Concepts

The level of security you implement may vary depending on your use case and your company's general security requirements. However, there are a few best practices that we recommend, regardless of your implementation.

General SAP BTP and Network Security Aspects

The SAP BTP landscape runs in an isolated network that is protected from the outside by firewalls, a DMZ, and communication proxies for all inbound and outbound communication. All user access is protected with transport layer security (TLS). Use SAP BTP Connectivity to enable your SAP BTP applications to access remote services on the Internet or in your on-premise systems. For connecting to Internet services, we recommend that you use APIs with destinations, and for cloud-to-on-premise scenarios, we recommend that you use destinations and the Cloud Connector. If you use destinations, you achieve a separation between the application and the configuration, which means it's easier to make configuration changes. And you can use the destinations to store credentials and certificates.

The Cloud Connector is a lightweight on-premise agent that establishes a tunnel for connecting your cloud applications to on-premise systems. The Cloud Connector acts as a reverse invoke between the on-premise network and SAP BTP. That means you don't need to open in-bound ports in the firewall for external access from the cloud. The Cloud Connector provides a fine-granular access control mechanism, supports multiple protocols, such as RFC and HTTP, and can forward the cloud user identity using principal propagation. It enables users to log on to on-premise systems without logging in again by forwarding their logged-on identity from the cloud. You can configure the Cloud Connector directly from the subaccount in the SAP BTP cockpit.

For more information, see:

- Principal Propagation
- Connectivity
- Cloud Connector

Identity Management and Authorization Management

Restrict access to any endpoint of your application through authentication and authorization. This ensures that only the intended target group, such as your company's employees, can access your application. We recommend that you use SAP BTP's default SAML or OpenID Connect single sign-on protocol for user acccess and principal propagation for back-end access. If only access for technical users is needed, however, principal propagation is not necessary. Once a user is authenticated, single sign-on propagates the credentials to the back-end system without requiring the user to reauthenticate.

Subaccounts get their users from identity providers. Administrators ensure that users can access only their subaccounts by establishing a dedicated trust relationship between the identity providers and the respective subaccounts. The preconfigured default identity provider for SAP BTP is SAP ID Service. SAP ID Service is managed by SAP. You can also connect your own identity provider, which means you have full control over your user base. For developing and administrating your own applications on SAP BTP, we recommend that you use SAP BTP Identity Authentication in your own tenant. Identity Authentication is SAP's cloud solution for identity lifecycle management for SAP BTP applications, and optionally for on-premise applications.

> **→ Recommendation**
>
> We recommend that you use SAP Cloud Identity Services - Identity Authentication as a hub, especially if your business users are stored in multiple corporate identity providers.
>
> For this scenario, connect Identity Authentication as single custom identity provider to SAP BTP. Next use Identity Authentication to integrate your corporate identity providers.
>
> For more information, see Corporate Identity Providers and Configure Conditional Authentication for an Application in What Is Identity Authentication and SAP Cloud Identity Services - Identity Authentication

See the following for more information about security best practices:

- Setting Up Authentication [page 49]
- Setting Up Authorization [page 54]
- Setting Up Identity Propagation [page 56]

Security Considerations for Applications

When building applications, your developers should use the security features of SAP BTP, such as protection from web attacks. We recommend that your developers configure and deploy application-based security artifacts containing authorizations, and administrators assign these authorizations using the cockpit. SAP BTP offers platform roles that help you ensure a segregation of duties, such as between app development and administration.

It's likely that data protection and privacy influence your architecture and the functions of your application, and you should consider any implications as early as possible in your development process. Security monitoring is done with audit logging. SAP BTP writes logs for security-relevant events and the written log files are digitally signed to ensure their integrity.

See the following for more information about security best practices:

- Protection from Web Attacks
- Giving Access Rights to Platform Users [page 57]
- Data Protection and Privacy

5.3.2 Setting Up Authentication

SAP BTP provides various options for implementing user authentication.

Use the decision tree below to determine how to set up authentication. Once you've implemented authentication, you can start implementing user authorization.

> **→ Remember**
>
> There are two types of users on SAP BTP: platform and business. Platform users are usually developers, administrators, or operators who deploy, administer, and troubleshoot applications and services. Business users are those who use the applications that are deployed to SAP BTP.

Setting Up Authentication
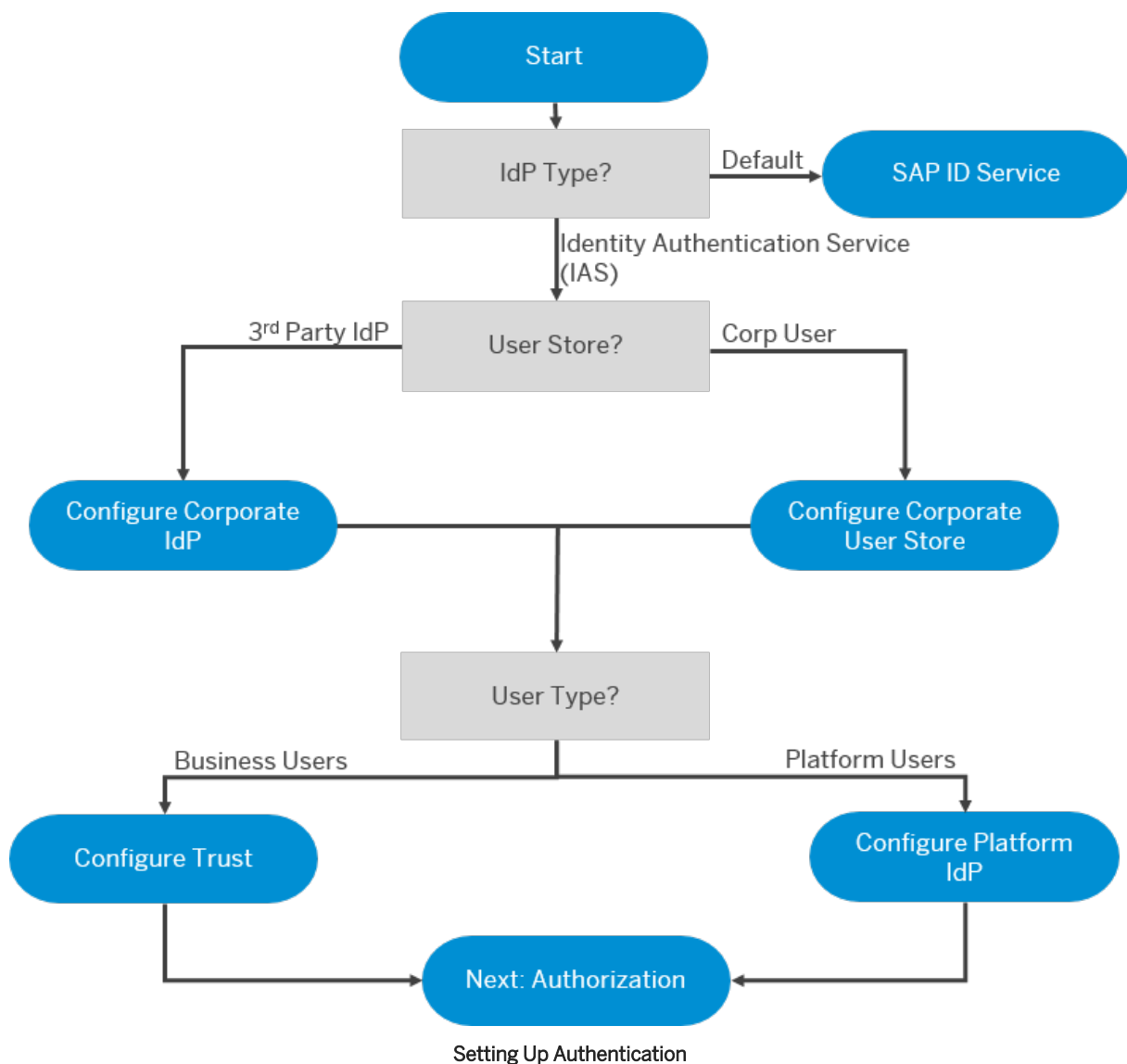
Users can be authenticated by the Identity Authentication service or by the SAP ID Service (default).

> → Recommendation
>
> We recommend that you use the Identity Authentication service and also connect any corporate IdPs to it.
>
> > i Note
> >
> > For platform users of feature set B, IdP is not yet available.

The Identity Authentication service can authenticate users stored in the IAS user store, your own corporate user store, or by forwarding to your own corporate identity provider.

Related Information

Set up Platform Identity Provider
Set up Identity Authentication Service for Existing Corporate Users
Trust and Federation with Identity Providers
Identity Authentication🐾

5.3.2.1 Authentication Methods

You can implement the authentication methods that are available on SAP BTP on the frontend of applications.

Authentication Methods on SAP BTP

Authentication Method	Default Options	Description
FORM or SAML2	Trusted SAML 2.0 identity provider with application-to-application SSO	FORM authentication implemented through the Security Assertion Markup Language (SAML) 2.0 protocol. Authentication is delegated to the SAP ID service or a custom identity provider.
BASIC	User name and password	HTTP basic authentication delegated to the SAP ID service or to an on-premise SAP NetWeaver AS Java system. Web browsers prompt users to enter a user name and password. By default, the SAP ID service is used.
CERT	Client certificate	Used for authentication only with client certificate.
BASICCERT	User name and password client certificate	Used for authentication either with a client certificate or with user name and password.
OAUTH	OAuth 2.0 token	Authentication according to the OAuth 2.0 protocol with an OAuth access token.
OpenID-Connect (OIDC)	OIDC token	A simple identity layer on top of the OAuth 2.0 protocol that allows clients to verify the identity of the end user based on the authentication performed by an identity provider, as well as to obtain basic profile information about the user.

Authentication methods depend on buildpacks, and the Cloud Foundry environment offers more than just Java. Please refer to the SAP Authorization and Trust Management Service in the Cloud Foundry environment documentation for more information.

5.3.2.2 Destination Authentication Methods

Destinations are part of SAP BTP Connectivity and are used for communication between a cloud application and a remote system.

Destinations are required to consume a target remote service from an application. They're resolved at runtime based on their symbolic names, resulting in an object that contains customer-specific configuration details, such as the URL of the remote system or service, the authentication type, and the respective credentials.

Destination Authentication Methods on SAP BTP

Authentication Method	Description	Environment	Internet Proxy	On-Premise Proxy
AppToAppSSO	Used for application-to-application communication if the caller needs to propagate its logged-in user. Both applications must be deployed on SAP BTP.	Neo	Yes	No
BasicAuthentication	Used for destinations that refer to a service on the Internet or an on-premise system that requires basic authentication.	Neo and Cloud Foundry	Yes	Yes
ClientCertificateAuthentication	Uses a technical user certificate to perform mutual transport layer security (TLS) authentication.	Neo and Cloud Foundry	Yes	No
NoAuthentication	Used for destinations that refer to a service on the Internet or an on-premise system that doesn't require authentication.	Neo and Cloud Foundry	Yes	Yes
OAuth2SAMLBearerAssertion	Enables applications to use SAML assertions to access OAuth-protected resources.	Neo and Cloud Foundry	Yes	Yes i Note Only for Neo

Authentication Method	Description	Environment	Internet Proxy	On-Premise Proxy
PrincipalPropagation	Forwards the identity of an on-demand user to the Cloud Connector, and from there to the back end of the relevant on-premise system.	Neo and Cloud Foundry	No	Yes
OAuth2ClientCredentials	Used to request an access token from an OAuth authorization server, using the `client_credentials` grant. The retrieved access token is cached and auto-renovated. When a token is about to expire, a new token is created shortly before the expiration of the old one.	Neo and Cloud Foundry	Yes	Yes **i Note** Only for Neo
OAuth2UserTokenExchange	Using this authentication type, the destination service performs all the user token exchange steps automatically, which lets you simplify your application development. This enables you to use a user token that you already have, in order to fetch a token from a different OAuth client, in the context of the same tenant.	Cloud Foundry	Yes	No
OAuth2Password	Provides support for applications to use the OAuth password grant flow for consuming OAuth-protected resources.	Cloud Foundry	Yes	No

Authentication Method	Description	Environment	Internet Proxy	On-Premise Proxy
OAuth2JWTBearer	To allow an application to call another application, pass the user context, and fetch resources, the caller application must pass an access token. In this authorization flow, the initial user token is passed to the OAuth server as input data. This process is performed automatically by the destination service, which helps to simplify application development. This enables you to use a user token that you already have, in order to fetch a token from a different OAuth client, in the context of the same tenant.	Cloud Foundry	Yes	No
SAMLAssertion	Retrieves a generated SAML assertion from the destination service.	Cloud Foundry	Yes	Yes

→ Recommendation

For user token exchange, use the OAuth2JWTBearer authentication method when possible, as OAuth2UserTokenExchange needs a two-step mechanism to achieve the same resolution.

Avoid using BasicAuthentication and OAuth2Password in production environments. However, they work well for test environments.

5.3.3 Setting Up Authorization

Users in an application require different authorizations because they work in different jobs. For more information, see Authorization Entities. SAP BTP provides various options for implementing user authorization.

- Manual assignment by using the command line interface (btp CLI) or the SAP BTP cockpit. Small-sized enterprises are typical use cases for this method.

- Assignment using a custom identity provider (IdP), based on the group attributes used for federation. Large firms, especially in the production phase, are candidates for this option.
  - Technically, with a prerequisite of the passing of the group attribute in the SAML assertion, you can set up federated authorization with platform users configured with a custom IdP too.
- Provisioning by way of external tools like the Identity Provisioning service and the SAP Cloud Identity Access Governance service. Like federation, this is an automated process and maps the roles to the organizations. This method is also typical for large firms, especially in the production phase.
  - With provisioning, you assign individual authorizations to a business role. For example, if you have a user who needs access to different systems, the automation makes sure to assign all the authorizations that are necessary to access all the connected systems.

> ### i Note
>
> Although technically handled differently, federation and provisioning are inherently similar. Doing the assignment in the IdP is simpler to implement but does not scale in the same way that provisioning can. The use of approval workflows and the information that authorizations belong together by way of role definitions are other distinct advantages in using the provisioning option. With setting up authorization in the IdP, this can only be manually achieved.

Use the decision tree below to determine how to set up authorization.

> ### → Remember
>
> There are two types of users on SAP BTP: platform and business. Platform users are usually developers, administrators, or operators who deploy, administer, and troubleshoot applications and services. Business users are those who use the applications that are deployed to SAP BTP.

Setting Up Authorization
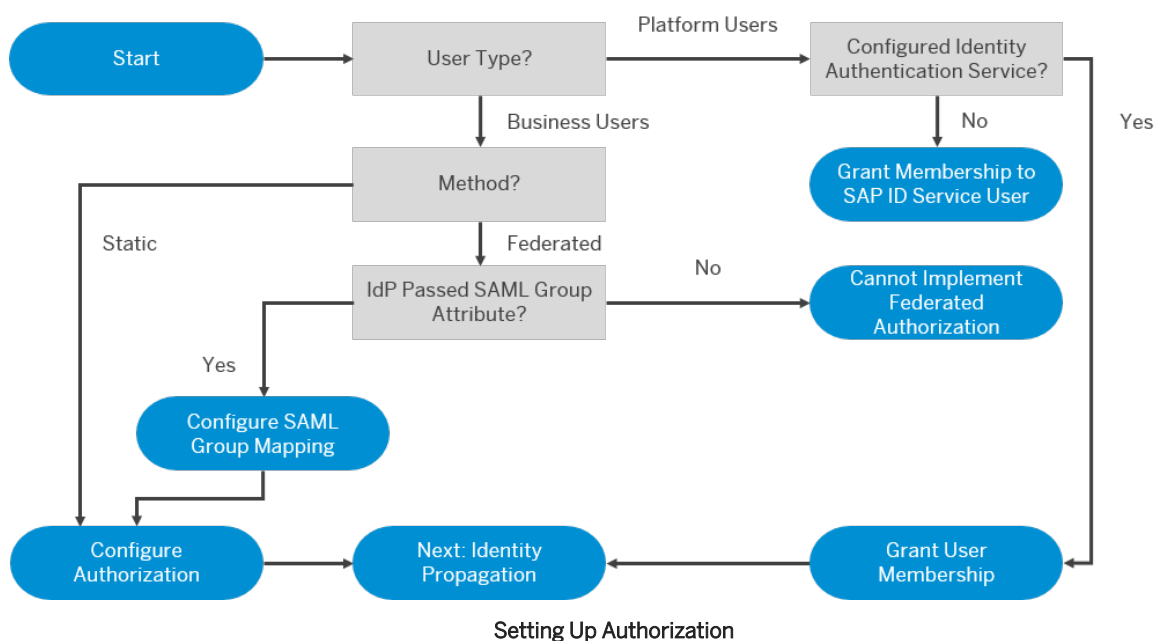
In general, although you can use either identity federation or provisioning in development and testing accounts too, it makes sense to just use the manual authorization configuration.

If you have a lot of users like in widely used production accounts, identity federation or provisioning is typically used.

The federation approach is valid only for business users. Provisioning is partially available for platform users. In Feature Set B, role collections on the subaccount level can be provisioned (for example, for Subaccount Administrator).

In addition to subaccount authorizations (like role collections), you also need to assign roles for the environment, such as the Cloud Foundry User and Member Management.

Related Information

Setup SAML 2.0 for Identity Federation
Setup and Manage Roles for Authorization
Security Administration: Managing Authentication and Authorization
SAP Cloud Identity Services

5.3.4 Setting Up Identity Propagation

SAP BTP provides various options for identity propagation.

The decision trees below for the SAP BTP Neo and Cloud Foundry environments might help you identify an appropriate setup; however, these are recommendations only. Your specific setup highly depends on your use case. In general, we recommend using the PrincipalPropagation authentication type with SAP systems and OAuth2SAMLBearer Assertion with third-party systems.
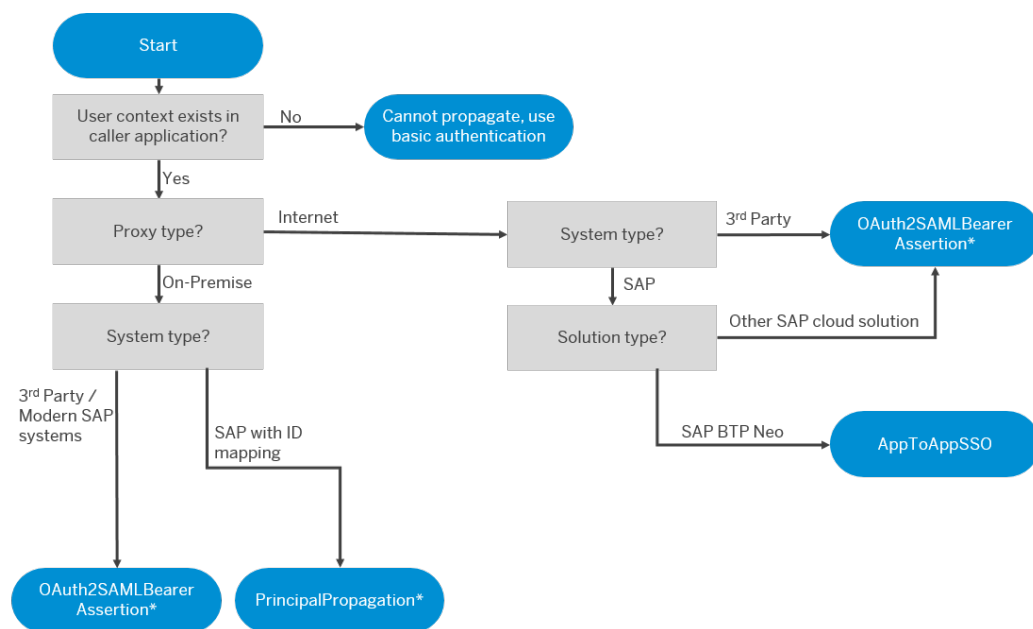
Setting Up Identity Propagation: Neo Environment

Setting Up Identity Propagation: Cloud Foundry Environment

Related Information

Principal Propagation
Principal Propagation via the Cloud Connector
Configuring Principal Propagation to an ABAP System
Application-to-Application SSO Authentication

5.3.5 Complying with Data Protection and Privacy Requirements

All companies must respect data protection and privacy rights of individuals. Data protection and privacy most likely influences the architecture and functionalities of your application and should be considered as early as possible in the development process.

It's important you understand that simply using SAP software, such as SAP BTP, does not make your applications compliant. However, SAP software can help you achieve compliance. You should consult legal experts to determine the requirements for your applications and specific use cases.

> **i Note**
>
> SAP does not provide legal advice in any form. SAP software supports data protection compliance by providing security features and data protection-relevant functions that can help to establish a compliant identity lifecycle management. In many cases, compliance with applicable data protection and privacy laws is not covered by a single product feature, and sometimes, not even within a product. For instance, users' wish to delete their personal data usually requires data deletion in several integrated systems.
>
> Furthermore, this information should not be taken as advice or a recommendation regarding additional features that would be required in specific IT environments. Decisions related to data protection must be made on a case-by-case basis, taking into consideration the given system landscape and the applicable legal requirements.

Related Information

SAP BTP Data Protection and Privacy

5.3.6 Giving Access Rights to Platform Users

If you've set up a staged development environment using different subaccounts or spaces, such as for development, testing, and production, we recommend that you grant the Cloud Development Team access to development subaccounts and spaces, but that you grant only the Cloud Administration Team access to the testing and production subaccounts or spaces.

All developers and stakeholders for a particular development project or application should get access to the Development subaccount or space, if the project or application has been properly enrolled. See Add Security

Administrators to Your Subaccount [Feature Set A] and Add Members to Your Subaccount [Feature Set B]. We recommend that you assign all developers the **Space Developer** role, to enable them to develop, assign roles, and test the application's functionalities. You can also configure custom platform roles. See Manage Custom Platform Roles.

As every small change that's made in a single application (for example, changing a destination) might affect all other applications, only the Cloud Administration Team should have access to the test and prod subaccounts or spaces, and should manage them centrally. If required, developers may be assigned to the **User and Role Auditor** role to get read access. For all changes required in subaccounts or spaces, the Cloud Development Team should reach out to the Cloud Administration Team.

> i Note
>
> To avoid overwhelming the Cloud Administration Team with inquiries, consider automating of tasks and providing a build infrastructure for continuous deployment.

Related Information

User and Member Management

5.4   Planning Failover on SAP BTP

Use a multi-data center setup and implement automatic failover to ensure the high availability of your applications in case of a data center outage.

Together with Implementing Failover [page 83], this chapter helps you implement a basic automated failover for SAPUI5 and HTML5 applications in a multi-data center setup. The failover approach ensures that your business-critical applications continue working in case of an unexpected downtime in an SAP BTP data center or SAP BTP Portal.

> ! Restriction
>
> At the moment, these chapters are only valid for SAPUI5 and HTML5 applications without data persistence or any kind of in-memory caching, whose data is stored in on-premise back-end systems.

What is Failover

Failover is an approach to ensure the high availability of your applications.

Generally, the term denotes the automated process of switching from one server, network, or system to another redundant one in case of an either unexpected or planned downtime. If the primary server, network, or system is unavailable, the traffic is redirected to the secondary one, which takes over the tasks. Therefore, both functionality and availability of your application is maintained.

The following figure illustrates this concept for a mutli-data center setup, as described in this guide:

Failover in Multi-DC Setup
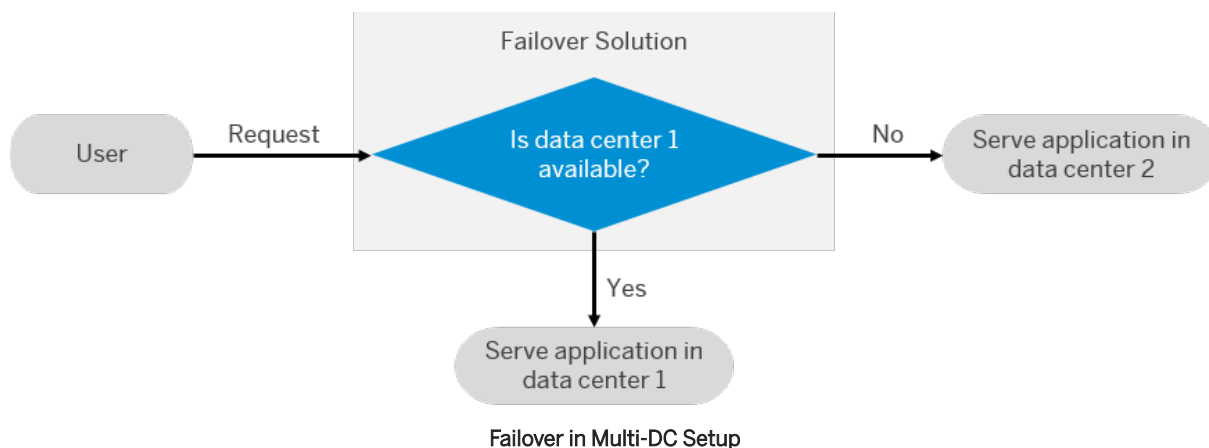
The SAP BTP Failover Guide focuses on four principles to consider when planning a failover:

Deploy Your Application in Two Data Centers

Our guide focuses on a multi-data center setup in which you deploy your application in two different data centers in parallel. The described setup is active/passive, which means that the traffic is generally directed to the application in the primary (active) data center. Only if the active data center experiences a downtime, the traffic is redirected to the application in the secondary (passive) one.

For more information, see Deploy Your Application in Two Data Centers [page 83].

Keep the Two Applications in Sync

To maintain the functionality of your application, make sure that both applications in the primary and the secondary data center are kept in sync. In this guide, we consider three different ways to orchestrate your applications:

- Orchestrate your applications manually. See Synchronize Your Applications Manually [page 84].
- Orchestrate your applications with the help of a continuous integration and delivery pipeline. See Use a Continuous Integration and Delivery Pipeline [page 85].
- Orchestrate your applications through a combination of the Solution Export Wizard and the SAP BTP Transport Management service. See Use the Solution Export Wizard and SAP Cloud Transport Management [page 86].

For more information, see Keep the Two Applications in Sync [page 84].

Define How a Failover Is Detected

To implement an automated failover from one data center to another, you must specify the cases in which the automatic switch has to be performed. You can choose from several options that vary in both their setup effort and flexibility, and use, for example, arbiters and different monitoring tools. In this guide, we describe a basic scenario, in which a user request triggers checks for both error codes and inappropriate response times.

For more information, see Define How a Failover Is Detected [page 88].

Decide on the Failback

Depending on your failover setup, a failback to the primary data center is either required or optional. In our active/passive scenario, the failback is mandatory and performed by the user himself. Additionally, we recommend to visually differentiate between the applications in the primary and the secondary data center so

that the user is constantly reminded of the necessity to switch back as soon as the primary data center is available again.

For more information, see Decide on the Failback [page 89].

Related Information

Develop Resilient Applications [page 73]

6 Develop and Build

SAP BTP offers various tools and programming languages for application development. You might also want to consider to integrate your application with other solutions.

We recommend that you use the SAP Cloud Application Programming Model for your development projects. For more information, see Cloud Application Programming Model [page 63].

Depending on your use case and the skill set of your developers, you can choose the runtime environment of your choice:

Environment options

Cloud Foundry	Kyma	ABAP
Benefits • Simplified developer experience for business application development • Large choice of programming languages • Intuitive "code-to-container" packaging and deployment, managed by the platform • Platform-managed application security patching and updates • Automatic application routing, load balancing, health checks, and multilevel self-healing • Support for CAP – an opinionated business app development framework	• Free choice of programming languages and models (containerized deployments) • Combines microservices and serverless functions • Brings built-in, managed, service mesh • More flexible with Kubernetes • Support for CAP – an opinionated business app development framework	• ABAP programming language • Fast prototyping with ABAP RESTful Programming Model (RAP) • Integrated development lifecycle • Reuse existing on-prem ABAP assets
Shared Benefits • No infrastructure vendor lock-in • Build scalable multitenancy business applications (SaaS) • Out-of-the-box consumption of SAP and hyperscaler services • Built on industry standards and open technology		

Cloud Foundry	Kyma	ABAP
Good For • Managed build-on approach • Enterprise-grade business applications and services • Cloud-native web applications and services • Scalable, microservice-based applications • Small to medium extensions built with CAP/low-code tooling	• Open build-on approach • Cloud-native development of apps and services • Highly scalable, microservice-based applications • Applications built with CAP	• User-centric process extensions • Robust, transactional cloud applications • Migrating and adapting add-ons to the cloud • Reusing existing on-premise ABAP code • Enabling ABAP developers to go to the cloud
Skills • Any major programming languages • SAP Fiori/UI5 and SAP HANA	• Kubernetes knowledge • Docker • NodeJS or Python for serverless functions • Any major programming language • SAP Fiori/UI5 and SAP HANA	• Ability to write modern ABAP code • Core data services • SAP Fiori/UI5 and SAP HANA

We recommend that you create multitarget applications for managing dependencies more easily. For more information, see Using Multitarget Applications to Manage Dependencies [page 71].

For best practices, guidelines and step-by-step instructions, see Development.

If you need information about the Neo environment, please have a look at Development, Neo Environment.

Defining Development Guidelines

To ensure consistency and foster collaboration between developers, we recommend that you define guidelines that include standards for programming principles, code styles, and naming conventions.

SAP provides the following official guidelines:

- SAPUI5 Guidelines
- SAP Fiori Design Guidelines

Programming Models, Programming Languages, Tools, and APIs [page 63]
SAP BTP provides various programming languages and tools for your development project.

Using Multitarget Applications to Manage Dependencies [page 71]
One challenge of moving into the cloud is deploying applications that consist of multiple interconnected components. We recommend that you develop multitarget applications that let you package those components into one bundle, and deploy and manage them all at once.

Add Authentication and Authorization [page 73]
As part of application development, developers create a role template during design time, which consists of scopes and attributes. Once the application is deployed, an administrator can create a role from this role template, which can then be assigned to a role collection.

Develop Resilient Applications [page 73]
Our best practices about resilient application design help you to make your applications running on SAP BTP stable and highly available.

Extending Existing SAP Solutions Using SAP BTP [page 74]

SAP BTP offers services, tools, and capabilities to develop, extend, or integrate business applications in the cloud. You can implement additional workflows or modules on top of existing solutions, which lets you benefit from out-of-the-box security, inherited data access governance, user interface embedding, and so on.

Performing UI, Usability, and Unit Tests [page 76]
> You should always conduct careful testing to ensure that your application is of high quality. Create a release candidate to propagate throughout your landscape.

6.1 Programming Models, Programming Languages, Tools, and APIs

SAP BTP provides various programming languages and tools for your development project.

For an overview about how to develop applications on SAP BTP, see Development.

Cloud Application Programming Model [page 63]
> The Cloud Application Programming Model offers a consistent end-to-end programming model that includes languages, libraries, and APIs that are tailored for full-stack development on SAP BTP.

Programming Languages [page 64]
> SAP BTP supports many different programming languages; the availability of each depends on the development environment you're using.

Continuous Integration and Delivery (CI/CD) [page 64]
> Configure and run predefined continuous integration and delivery (CI/CD) pipelines that automatically build, test, and deploy your code changes to speed up your development and delivery cycles.

Tools [page 67]
> SAP BTP includes many tools to help you develop and manage applications, and connect them to your on-premise systems. The availability of tools can depend on the environment and cloud management tools feature set that you are running on.

APIs [page 70]
> Discover and consume APIs to manage, build, and extend the core capabilities of SAP BTP.

6.1.1 Cloud Application Programming Model

The Cloud Application Programming Model offers a consistent end-to-end programming model that includes languages, libraries, and APIs that are tailored for full-stack development on SAP BTP.

It simplifies the development process by enabling you to create concise and comprehensive data and service models at a conceptual level, which are then used as input for the data, service, and UI layers. The SAP Cloud Application Programming Model is compatible with any development environment, but we recommend SAP Business Application Studio.

For more information, see Developing with the SAP Cloud Application Programming Model and SAP Business Application Studio.

6.1.2 Programming Languages

SAP BTP supports many different programming languages; the availability of each depends on the development environment you're using.

Environment	More Information
Cloud Foundry environnment	See Development in the Cloud Foundry Environment
ABAP environment	See Development in the ABAP Environment
Kyma environment	See Development in the Kyma Environment
Neo environment	See Development, Neo Environment.

Related Information

SAP BTP, Serverless Runtime - Development

6.1.3 Continuous Integration and Delivery (CI/CD)

Configure and run predefined continuous integration and delivery (CI/CD) pipelines that automatically build, test, and deploy your code changes to speed up your development and delivery cycles.

> i Note
>
> For links to all SAP solutions for CI/CD, blog posts, presentations, and tutorials, have a look at our Continuous Integration and Delivery by SAP overview.

Continuous integration (CI) describes a software development process, in which various team members integrate their contributions frequently into a single main line. Before each integration, the changes are verified through builds and automated testing. Thereby, you can detect errors as quickly as possible and prevent integration problems before completing the development.

The **continuous delivery (CD)** concept expands on the one of continuous integration. It adds the aspect that any change that has successfully passed the tests is immediately ready to be deployed to production, both from a technical and a qualitative point of view.

The following graphic shows the basic flow for continuous integration and delivery:
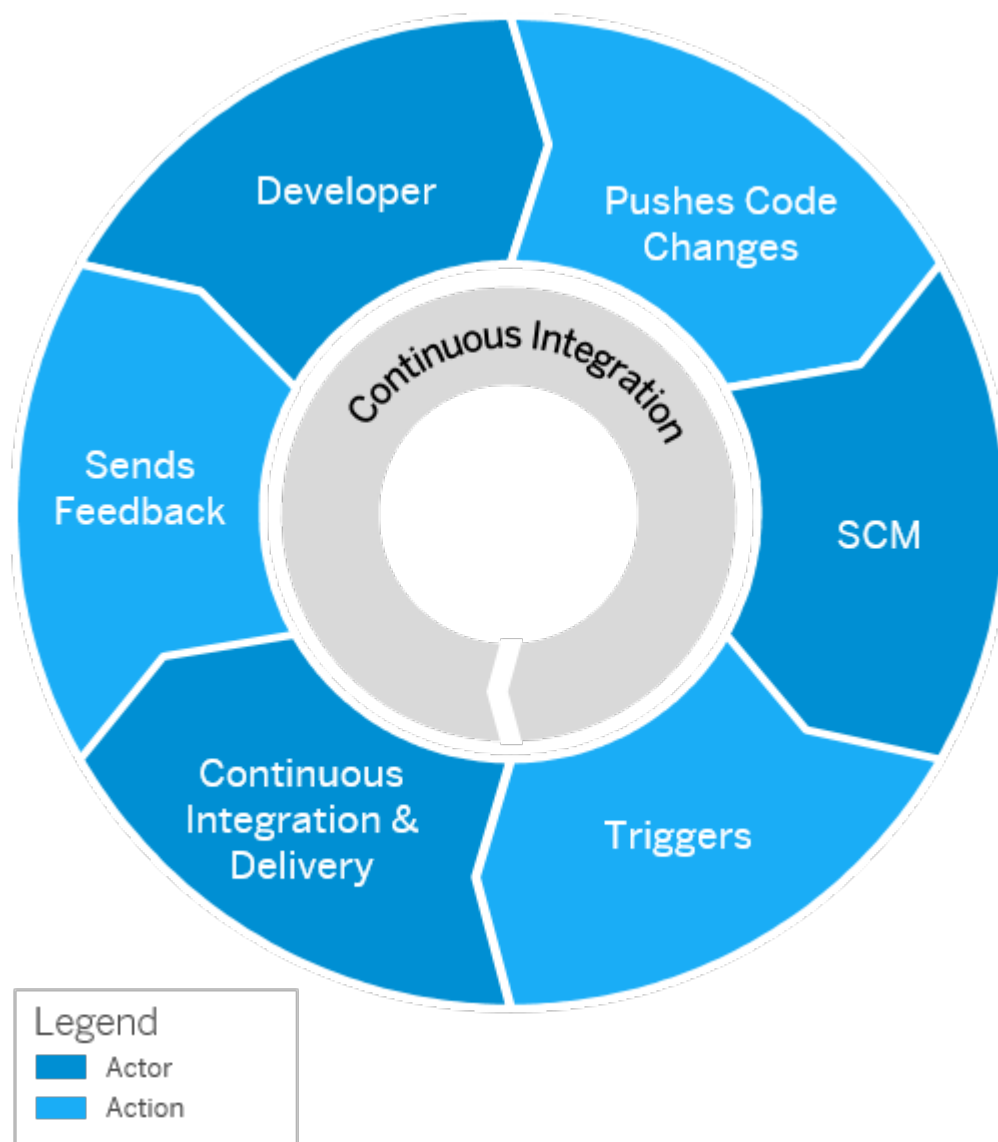
Continuous Integration Basic Flow

For more information about the continuous integration and continuous delivery concepts, see What Are Continuous Integration and Continuous Delivery?.

Use

Depending on your use case, you can choose between different CI/CD pipelines to help you implement continuous integration and delivery in your software development.

SAP Continuous Integration and Delivery lets you configure and run predefined pipelines for the development of the following applications:

- SAP Cloud Application Programming Model

Configure a CI/CD pipeline for the development of applications that follow the SAP Cloud Application Programming Model in the Cloud Foundry environment.

- SAP Fiori in the Cloud Foundry Environment
  Configure a CI/CD pipeline for the development of SAPUI5/SAP Fiori applications in the Cloud Foundry environment.

- SAP Fiori in the Neo environment
  Configure a CI/CD pipeline for the development of SAPUI5/SAP Fiori applications in the Neo environment.

- SAP Integration Suite Artifacts
  Configure a CI/CD pipeline for the development of SAP Cloud Integration artifacts in the Cloud Foundry environment.

- Container-Based Applications
  Configure a CI/CD pipeline for the development of container-based applications.

To learn more about the CI/CD pipelines supported by SAP Continuous Integration and Delivery and the stages each pipeline can comprise, see Supported Pipelines.

Get Started with CI/CD

SAP Continuous Integration and Delivery provides an easy, UI-guided way to set up the service and configure and run your pipelines, without hosting your own Jenkins instance.

To set up SAP Continuous Integration and Delivery:

1. Enable the service in the SAP BTP cockpit.
2. Assign either the *Administrator* or *Developer* role to your user.
3. Enable the API usage to connect SAP Continuous Integration and Delivery to other services, if necessary.

To configure SAP Continuous Integration and Delivery:

> **i Note**
>
> Only administrators of SAP Continuous Integration and Delivery can configure the service.

1. Configure credentials for connecting SAP Continuous Integration and Delivery to other services (for example, GitHub, GitLab or Bitbucket Server to clone your sources, and SAP BTP to deploy your built application).
2. Add your repository.

Now you can create and modify your CI/CD jobs and monitor their outcome. If you want to automate your builds, you can configure a webhook between your repository and the service. You can create and modify timed triggers for your jobs, if necessary.

For more information, see Initial Setup and Configuration, or follow the tutorial Configure and Run a Predefined SAP Continuous Integration and Delivery (CI/CD) Pipeline .

Learn and Get Certified

Depending on your learning goals and level of expertise, you can choose from the following offerings:

- **Continuous Integration and Continuous Delivery on Learning Journey**
  Our learning journey is a visual guide that helps you complete the learning path for CI/CD on SAP BTP. It makes you understand the practices and principles of continuous integration and delivery, provides the opportunity to gather hands-on experience, and helps you improve your skills in CI/CD.
  For more information about Learning Journeys, see Jump-start your Learning.
- **Continuous Integration and Delivery Introduction Guide**
  The CI/CD Introduction Guide provides you with basic knowledge for setting up and implementing continuous integration and delivery processes. It gives an overview of the concepts and principles of CI/CD, explains both procedures and their relation, and helps you plan your own CI/CD process.

6.1.4 Tools

SAP BTP includes many tools to help you develop and manage applications, and connect them to your on-premise systems. The availability of tools can depend on the environment and cloud management tools feature set that you are running on.

Tools Available for SAP BTP Multi-Cloud Foundation

Tool	Description
Account Administration in the Cockpit	The SAP BTP cockpit is the web-based administration interface of SAP BTP and provides access to a number of functions for configuring and managing applications, services, and subaccounts. Use the cockpit to manage resources, services, security, monitor application metrics, and perform actions on cloud applications.
Cloud Connector	The Cloud Connector serves as the link between on-demand applications in SAP BTP. This is the browser-based and existing on-premise systems. You can control the resources available for the cloud applications in those systems.
Command Line Interface for Cloud Foundry	The Cloud Foundry command line interface enables you to work with the Cloud Foundry environment to deploy and manage your applications.
SAP BTP Command Line Interface [Feature Set B]	The SAP BTP command line interface (btp CLI) is the command line tool for convenient account management, such as managing global accounts, directories, subaccounts, entitlements, environment instances, multitenant application subscriptions, and users and their authorizations.

Tool	Description
SAP BTP SDK for iOS	The SAP BTP SDK for iOS is based on the Apple Swift programming language for developing apps in the Xcode IDE and includes well-defined layers (SDK frameworks, components, and platform services) that simplify development of enterprise-ready mobile native iOS apps. The SDK is tightly integrated with SAP Mobile Services for Development and Operations.
SAP BTP SDK for Android	The SAP BTP SDK for Android provides development tools for creating native Android mobile applications that use SAP Mobile Services. The SDK is based on the Java programming language and is built on top of Google's Android SDK.
SAP Cloud SDK	SAP Cloud SDK provides a layer of abstractions for features of SAP BTP such as logging, multitenancy, and connectivity. It also includes project templates for different execution environments and implementations.
Eclipse Tool for the Cloud Foundry Environment ↗	The Eclipse plug-in for the Cloud Foundry environment is a Java-based toolkit for Eclipse IDE that enables you to develop and deploy Java and Spring applications in the Cloud Foundry environment from Eclipse or Spring Tool Suite, as well as perform operations such as logging, managing user roles, creating connectivity destinations, and so on.
SAP Web IDE	With SAP Web IDE Full-Stack, you can easily develop, test, build, deploy, and extend role-based, consumer-grade apps for business users. Create applications rapidly and deliver an outstanding user experience. You can extend or build SAP Fiori apps, create SaaS solutions, extend SAP S/4HANA cloud services, develop hybrid mobile apps, and build IoT apps for SAP Leonardo, using the UI development toolkit for HTML5, the SAP HANA toolset, and Java programming language and technologies.
SAP Business Application Studio	SAP Business Application Studio is the next generation of SAP Web IDE - Develop, debug, test, and deploy SAP business applications.
Service-Specific Tools	The services that run on SAP BTP can come with service-specific tools. For an overview of the services and their tools, see the SAP Discovery Center ↗.

Tools Available in the Neo Environment

Tool	Description
SAP BTP Cockpit	The SAP BTP cockpit is the web-based administration interface of SAP BTP and provides access to a number of functions for configuring and managing applications, services, and subaccounts. Use the cockpit to manage resources, services, security, monitor application metrics, and perform actions on cloud applications.
Account Administration Using the SAP BTP Command Line Interface (btp CLI) [Feature Set B]	The SAP BTP command line interface (btp CLI) is the command-line tool for convenient account management, such as managing global accounts, directories, and subaccounts.
SAP Web IDE	SAP Web IDE is a cloud-based meeting space where multiple application developers can work together from a common Web interface — connecting to the same shared repository with virtually no setup required. SAP Web IDE allows you to prototype, develop, package, deploy, and extend SAPUI5 applications.
Maven Plugin	The Maven plugin supports you in using Maven to develop Java applications to the Neo environment. It allows you to conveniently call the console client and its commands from the Maven environment.
Cloud Connector	The Cloud Connector serves as the link between on-demand applications in the Neo environment. This is the browser-based and existing on-premise systems. You can control the resources available for the cloud applications in those systems.
SAP BTP SDK for Neo Environment	The SAP BTP SDK for the Neo environment contains everything you need to work with the Neo environment , including a local server runtime and a set of command line tools.
SAP BTP SDK for iOS	The SAP BTP SDK for iOS is based on the Apple Swift programming language for developing apps in the Xcode IDE and includes well-defined layers (SDK frameworks, components, and platform services) that simplify development of enterprise-ready mobile native iOS apps. The SDK is tightly integrated with SAP Mobile Services for Development and Operations.
SAP JVM Tools in Eclipse for the Neo Environment	SAP JVM Tools for Eclipse is an Eclipse plugin that allows you to debug and profile applications which run on SAP JVM.
Console Client for the Neo Environment	The console client for the Neo environment enables development, deployment and configuration of an application outside the Eclipse IDE as well as continuous integration and automation tasks.

6.1.5 APIs

Discover and consume APIs to manage, build, and extend the core capabilities of SAP BTP.

An Application Programming Interface or API is an **interface provided by an application for interacting with other applications**. APIs specify how software programs are able to exchange information with each other, even if designed and run by different organizations. They facilitate interaction by selectively exposing certain functionality, allowing different apps, websites, and devices to communicate effectively with each other. More importantly, APIs allow businesses to reach beyond regular business channels and share data, content, and services directly to both B2B (business to business) and B2C (business to consumer) clients, making UI development easy.

API Offerings on SAP BTP

SAP BTP enables you to consume APIs and publish your own ones through the following offerings:

Offering	Description
SAP BTP on the SAP API Business Hub	The SAP API Business Hub provides you with one central repository for browsing and accessing APIs from SAP and select partners. Test APIs and try out mock data in your systems. It is also the official place where REST and OData REST API references are published.
SAP BTP API Management	API Management allows you to build, manage, publish, and monetize your own APIs within one secure and scalable environment.
SDKs	The software development kits (SDKs) available for SAP BTP offer APIs to, for example, accelerate enterprise app development. For an overview on the available SDKs, see Tools.

Related Information

SAP API Style Guide

6.2 Using Multitarget Applications to Manage Dependencies

One challenge of moving into the cloud is deploying applications that consist of multiple interconnected components. We recommend that you develop multitarget applications that let you package those components into one bundle, and deploy and manage them all at once.

Cloud applications often come with a lot of heterogeneity, which is one of the key strengths of cloud development, allowing for agility, resilience, and the rapid development of new features. However, it also increases the complexity of cloud applications, which:

- Usually consist of multiple interdependent software modules
- Are written in different programming languages using multiple development tools
- Might involve different products
- May be deployed to multiple target runtimes

A combined lifecycle lets you deploy all parts together, automatically, and in the right order, and manage the configuration of the complete solution. You can achieve such a combined lifecycle by developing multitarget applications. Each multitarget application has the following characteristics:

- One archive file that includes all modules and a description of the dependencies
- Can be delivered, transported, linked to SAP software components, and deployed
- The process can be automated in a continuous integration pipeline

The multitarget application archive contains all required application types and configurations, as well as a deployment descriptor file. It is intended to be used as a generic artifact that can be deployed and managed on several SAP BTP subaccounts. For example, you can use one multitarget application archive in your development subaccount and reuse it in your production subaccounts.

As all interdependencies are part of the archive file, it's easy to pass multitarget applications from development to operations. All required information for deployment is provided during the development process. Due to the benefits provided by applying the multitarget application approach, it is also part of the SAP Cloud Application Programming Model.
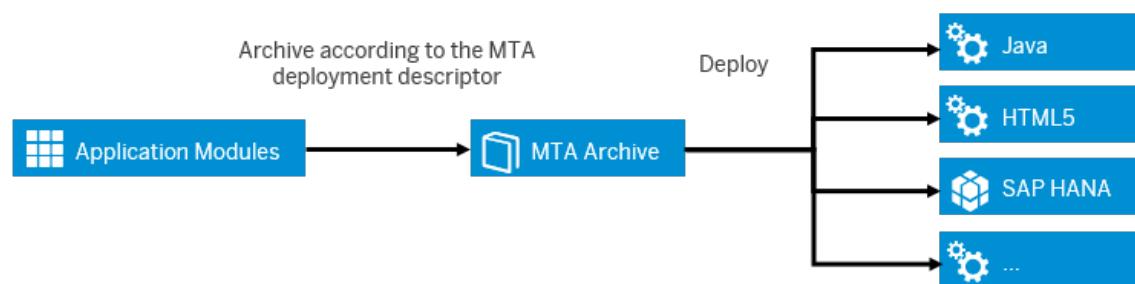
> ### → Recommendation
>
> The approach isn't mandatory for applications that are running on SAP BTP – you can also develop without applying it. Without the multitarget application approach, you'll need to manually deploy your application artifacts, for example by triggering the deployment from SAP Web IDE or manually uploading artifacts via SAP BTP cockpit.
>
> We recommend that you use multitarget applications in the following cases:

- You're developing a business application composed of several different parts – apps, services, content, and others – that you want to manage as a single unit.
- Your business application has dependencies to external resources, such as backing services (database, messaging, and so on), APIs, and configurations from other applications.
- Your business application has a certain default configuration, for example memory, disk, number of individual app instances, environment variables, service plans, and others.

For more conceptual information about multitarget applications and detailed step-by-step instructions, see Multitarget Applications in the Cloud Foundry Environment.

There are several options to create multitarget application archives:

- If you use SAP Web IDE Full-Stack, you can use multitarget application templates for Cloud Foundry applications, where the descriptor file is maintained automatically, for example, whenever you add a new module in the SAP Web IDE.
- If you have development modules from other sources, you can use the multitarget application archive builder, a Java-based command-line tool that builds modules and packages them into a deployable multitarget application archive, together with a deployment descriptor. It is available for download from SAP Development Tools (see https://tools.hana.ondemand.com/#cloud).
- If you have solutions already deployed in the Neo environment, you can use the solution export wizard, as offered in the Solutions view of the SAP BTP cockpit. The wizard lets you select existing components (considering interdependencies automatically) and offers to download the multitarget application archive (not supported for archives comprising Java modules) or descriptor files, or to integrate the changes into transport management processes (see Deploy and Deliver [page 78]). Applications, configuration content, such as destinations, roles, and security groups are supported. For more information, see Exporting Solutions on SAP Help Portal.

6.2.1 Establishing a Provider/Subscriber Scenario Using Multitenancy

The provider/subscriber approach is one of the important scenarios that are supported with multitarget application.

SAP BTP allows you to develop and run multitenant (tenant-aware) applications. These applications, which must apply the multitarget application approach, run on a shared compute unit that can be used by multiple consumers (tenants). Each consumer accesses the application through a dedicated URL.

For example, SAP partners or central IT organizations can build and run multitenant apps in their provider accounts, deploy them using the multitarget application concept as providers, then deliver them as services to their customers or line of business subsidiaries in corresponding subscriber accounts via subscriptions.

One of the advantages of this approach is that the application lifecycle and resource management is centrally managed. A single deployment can serve several subscribing subaccounts, which eases onboarding of new customers and decreases the number of code versions to manage. This also eases the consumption, as the actual customers of the applications do not have to deploy solutions and manage corresponding resources.

For more information, see the following:

- Multitenancy Roles
- Developing Multitenant Applications in the Neo Environment

- Developing Multitenant Applications in the Cloud Foundry Environment

6.3 Add Authentication and Authorization

As part of application development, developers create a role template during design time, which consists of scopes and attributes. Once the application is deployed, an administrator can create a role from this role template, which can then be assigned to a role collection.

For more information about the relevant security aspects and how to set those up, see Adding Authentication and Authorization.

6.4 Develop Resilient Applications

Our best practices about resilient application design help you to make your applications running on SAP BTP stable and highly available.

There are many different principles and patterns you can use to make your software resilient. It is, however, not always easy to find the combination that best fits your applications. The *Developing Resilient Apps on SAP BTP* guide gives an overview of the various options you have when developing applications and detailed information about the individual patterns you can use. Have a look at examples for applied resilience principles and find out how to develop your own resilient apps on SAP BTP: **Developing Resilient Apps on SAP BTP**.

Developing Resilient Applications in the Cloud Foundry Environment

In the Cloud Foundry environment, you can make use of the availability zones (AZ): Benefit from the high availability mechanisms in Cloud Foundry by setting up your applications with multiple instances. You do not have to do anything explicitly to distribute the instances across the different AZs - this is handled by the platform.

Setting up your application with multiple instances might have an impact on your applications' capability for handling requests: In case of an AZ failure in a 3-AZ-deployment, at least one third of the application instances are affected and unavailable until Cloud Foundry can reschedule these instances on a healthy virtual machine.
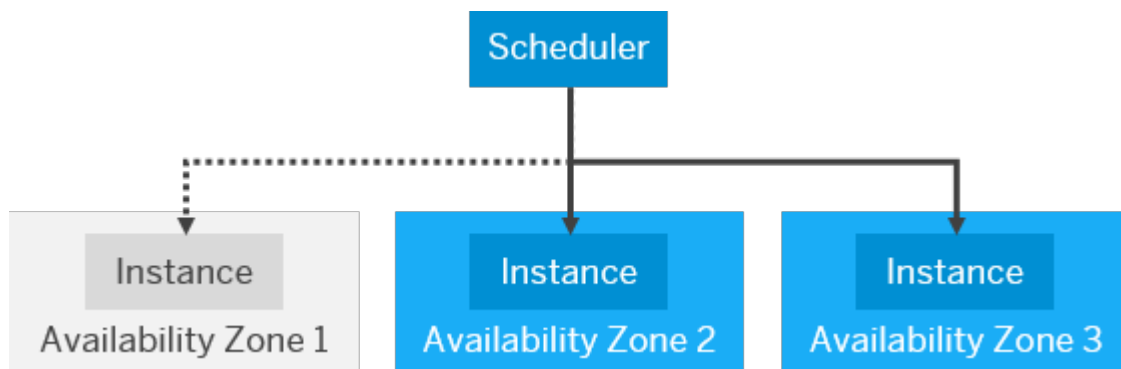
Distribution During an Availability Zone Failure

During that period, the remaining instances in the healthy AZs have to carry the request load.

For more information about availability zones, see Availability Zones in the Cloud Foundry Environment.

See also:

- Resilience, High Availability, and Disaster Recovery (Cloud Foundry, ABAP and Kyma Environments)
- Resilience, High Availability, and Disaster Recovery (Neo Environment)

6.5   Extending Existing SAP Solutions Using SAP BTP

SAP BTP offers services, tools, and capabilities to develop, extend, or integrate business applications in the cloud. You can implement additional workflows or modules on top of existing solutions, which lets you benefit from out-of-the-box security, inherited data access governance, user interface embedding, and so on.

Overview

All standard SAP solutions are offered with customizing capabilities. Additionally, customers often have their own requirements for innovative or industry-specific extensions and the extension capability of SAP BTP can help them build, deploy, and operate their new functionalities easily and securely.

You can use SAP BTP to extend standard SAP solutions without disrupting their performance and core processes. When building extension applications, you can also benefit from the automation of the integration between and the SAP BTPextended SAP solutions.

Extensions with Automated Configurations

SAP BTP provides a standard way for extending SAP solutions and developing event-driven extensions and applications. This framework includes:

- Simplified, standardized and unified extensibility and configuration for the SAP solutions.
- Central catalog per customer for all connected SAP systems where data such as APIs, events, credentials and other is stored. You can benefit from business services and actionable events across end-to-end business processes.

If you have to group the systems of different SAP solutions in the same business case, you can set up the connectivity between all these systems and SAP BTP in a single formation.

The following SAP solutions currently support the automated configurations:

- SAP S/4HANA Cloud
- SAP SuccessFactors
- SAP Cloud for Customer
- SAP Commerce Cloud
- SAP Field Service Management

Extensions with Manual Configurations

If the automated integration with SAP BTP does not support your scenario, you can configure the integration manually. Using manual configurations, you can extend SAP solutions on SAP BTP, such as:

- SAP S/4HANA Cloud
- SAP SuccessFactors
- SAP Cloud for Customer

Use Cases

The extension use cases include but are not limited to:

- Extending the user interface of your SAP cloud solution.
  You can complement an existing SAP solution with a new or improved SAP Fiori user interface without adding any major logic or data.
  You can also provide necessary information and tools on mobile devices.
  For example:
    - Building completely new user interfaces that can be seamlessly integrated in the SAP SuccessFactors Employee Central home page.
    - Adding new functionality to SAP SuccessFactors by connecting it to other SAP solutions, such as SAP Commerce Cloud, SAP Cloud for Customer, SAP Field Service Management, SAP S/4HANA Cloud.
    - Building completely new Fiori-based user interfaces that can be seamlessly integrated in the SAP Fiori launchpad.
- Enhancing the functionality of your SAP cloud solution by connecting it with other SAP solutions.
  For example:
    - Connecting SAP S/4HANA Cloud with other SAP solutions, such as SAP Commerce Cloud, SAP Cloud for Customer, SAP Field Service Management, SAP SuccessFactors.
    - Connecting SAP SuccessFactors by connecting it to other SAP solutions, such as SAP Commerce Cloud, SAP Cloud for Customer, SAP Field Service Management, SAP S/4HANA Cloud.
- Enhancing core business processes with analytics and machine learning capabilities.
- Extend data insights by consolidating and combining data in one central place.
- Building an extensions ecosystem, by creating and operating SaaS applications and selling them to multiple customers.
  For example, as an SAP Partner, you can build a multi-tenant SaaS extension application and provide it to your customers via:
    - SAP App Center

○ SAP BTP cockpit

For more use cases, see https://www.sap.com/products/cloud-platform/use-cases.html?sort=title_asc🌩.

Resources

The following resources are available for your extension scenarios. The links to various documents guide you through the configuration tasks that you need to perform to enable the SAP BTP for developing extension applications for your existing SAP solutions, and the learning journeys are collections of links to additional resources.

- SAP S/4HANA Cloud
  ○ Extending SAP S/4HANA Cloud in the Cloud Foundry and Kyma Environment (SAP Help Portal)
  ○ Extend SAP S/4HANA Cloud on SAP Business Technology Platform🌩 (Mission in Discovery Center)
  ○ Build an Event-Driven Geomarketing Extension for SAP S/4HANA Cloud🌩 (Mission in Discovery Center)
  ○ SAP Extensibility Explorer for SAP S/4HANA Cloud
  ○ Tutorials about the integration of SAP BTP ABAP Environment with SAP S/4HANA Cloud🌩
  ○ SAP S/4HANA Cloud Extensions on SAP BTP (Learning Journey on SAP Help Portal)
  ○ Extending SAP S/4HANA Cloud in the Cloud Foundry Environment Manually (SAP Help Portal)
- SAP Customer Experience (CX) products
  ○ Extending SAP Customer Experience Products in the Kyma Environment (SAP Help Portal)
  ○ Create Extensions in the Kyma Environment Using Functions (SAP Help Portal)
  ○ Extending SAP Cloud for Customer in the Cloud Foundry Environment Manually (SAP Help Portal)
- SAP SuccessFactors
  ○ Extending SAP SuccessFactors in the Cloud Foundry and Kyma Environment (SAP Help Portal)
  ○ Extend SAP SuccessFactors on SAP BTP 🌩 (Mission in Discovery Center)
  ○ Build Resilient Cloud Native Applications 🌩 (Mission in Discovery Center)
  ○ SAP SuccessFactors Extensions on SAP BTP (Learning Journey on SAP Help Portal)

Related Information

Extensions

6.6 Performing UI, Usability, and Unit Tests

You should always conduct careful testing to ensure that your application is of high quality. Create a release candidate to propagate throughout your landscape.

For example, consider testing the user interface, usability, and performance, as well as running general unit tests. Developers should use unit tests to ensure that software modules behave as expected. Unit tests are the

smallest tests, and they detect issues fast. Taking good care of unit tests usually leads to better maintainable and understandable code. If you use the Continuous Integration process, you should consider automated tests as part of the pipeline.

SAPUI5

SAPUI5 supports QUnit testing. We recommend that you write unit tests as small as possible, ensuring that you only test one single thing at a time. This will not only make debugging easier, but also maintaining the tests in the long run.

For an overview of available testing options for SAPUI5 developments, see Testing and Performance Measurement.

For SAPUI5 and SAP Fiori testing, please see the demokit for SAP UI5: https://sapui5.hana.ondemand.com/#/topic.

Testing in SAP Web IDE

SAP Web IDE empowers developers to test and evaluate the functionality and performance of their apps:

- Use the application preview to test functionality, design and performance before deploying it. There are different configuration options available, such as SAP UI5 runtime version, URL parameters. See Running Applications in Development Mode.
- Run your application using a client mock server. See Run Applications with Mock Data.
- Execute unit tests for SAPUI5 (Qunit) and Java (JUnit). See Developing Application Tests and Test a Java Module.
- Execute SAPUI5 integration tests based on OPA5. See Performing Integration Tests [page 94].

7 Deploy and Deliver

After you've developed your application, you need to deploy it and deliver it to your TEST and PROD environments. By deploying it to two different data centers in parallel, you can implement failover.

7.1 Deploying Applications

You can leverage different deployment tools and methods, depending on the application type and your requirements.

7.1.1 Deploying Simple Applications

If your application consists of only one module, there are different native options for deploying it, depending on the programming language you're using.

Java

You can use the SAP BTP cockpit, the console client (Neo), or the command-line tool (Cloud Foundry) to deploy a Java application. You have several options to choose from with regards to runtimes, Java Virtual Machine versions, and more.

For large-scale development setups, we recommend that you connect your own development infrastructure.

HTML5

You can deploy HTML5 applications from within the SAP Web IDE. In addition:

- In the Neo environment, you can export and import your application via the SAP BTP cockpit.
- In the Cloud Foundry environment, you can deploy your application using the Cloud Foundry command-line interface.

Node.js

In the Cloud Foundry environment, you can deploy Node.js applications from within SAP Web IDE, or use the Cloud Foundry command-line interface.

SAP HANA Extended Application Services, Classic Model (SAP HANA XS)

In the Neo environment, you can create a delivery unit as a .tgz file that contains all the artifacts for applications that are based on SAP HANA XS. You can import and export the application using SAP HANA administration tools, and open the application using the SAP BTP cockpit.

Please be aware that SAP HANA Extended Application Services, Classic Model has been deprecated – for more information, see 0002465027 .

SAP HANA Extended Application Services, Advanced Model (SAP HANA XSA)

In the Cloud Foundry environment, you can deploy applications based on SAP HANA XSA using the SAP HANA Deployment Infrastructure (HDI) from within SAP Web IDE. You can also deploy your application using the Cloud Foundry command-line interface.

Bring Your Own Buildpack

In the Cloud Foundry environment, you can use your own buildpack to create applications. The deployment of such an application depends on the buildpack-specific development and deployment infrastructure.

7.1.2 Deploying Multitarget Applications

There are two options for deploying MTAs: Manually or managed.

- **Manually:** You can archive all components of your application into one package that includes the deployment descriptor. You can then manually trigger the deployment of the solution using the SAP BTP cockpit or the Console Client. The actual deployment of the MTA files is then performed automatically by the SAP BTP deployment infrastructure, considering all interdependencies specified in the MTA deployment descriptor that is part of the MTA archive. By assigning roles and encapsulating the Console Client in a script, you can achieve a somewhat controlled release mechanism.
- **Managed:** If you prefer a managed approach, you can deploy MTAs as part of a CI/CD approach or implement transport or change management processes for your SAP BTP apps, using enhanced Change and Transport (CTS+) for example. For more information, see Transporting Multitarget Applications with CTS+.

Related Information

Deploying to the Cloud Foundry Environment

7.2 Delivering Applications

There are several options for propagating developed applications from one subaccount to another, for example, from your Development to your Testing and Production subaccounts.

Delivery Option (Tool Support and Automation Decreasing from Top to Bottom)	Development content (such as Java, HTML5, SAP HANA XSA)	SAP Integration packages	Other app-specific content	SAP HANA XS	Portal Site
CI/CD	recommended	n/a	n/a	recommended for SAP HANA XSA	partly possible
Transport Management service with the option to integrate into Change Management (ChaRM or QGM of SAP Solution Manager)	recommended, but only possible as MTA	recommended, but only possible as MTA	recommended	recommended	recommended, but only possible as MTA
CTS+ with the option to integrate into Change Management (ChaRM or QGM of SAP Solution Manager)	recommended, but only possible as MTA	recommended, but only possible as MTA	n/a	recommended	recommended, but only possible as MTA
SAP HANA Application Lifecycle Management	n/a	n/a	n/a	recommended for SAP HANA XSC	n/a
Manually	possible	possible	recommended, but with app-specific procedures	possible	possible

- Choose your SAP solution for continuous integration and delivery.
  At the moment, SAP offers three solutions for applying CI/CD in your software development. Depending on your expertise in CI/CD, the level of flexibility you wish, and the infrastructure you bring (or don't bring) along, choose one of the following options:

SAP Solutions for Continuous Integration and Delivery

CI/CD Solution	Required Expertise in CI/CD	Level of Flexibility	Infrastructure
SAP Continuous Integration and Delivery	Low	Low	SAP Continuous Integration and Delivery is ready to use and doesn't need a separate CI/CD infrastructure.
Project "Piper" 🔗	Medium	Medium	Project "Piper" is an open-source project that provides preconfigured Jenkins pipelines, which you can use in your own Jenkins infrastructure and adapt according to your needs, if necessary. Project "Piper" requires Jenkins as underlying CI/CD infrastructure. If you don't have a Jenkins instance, yet, use the Cx Server to bootstrap a preconfigured one. See Getting Started with Project "Piper".
Continuous Integration and Delivery Best Practices Guide	High	High	The CI/CD Best Practices guide requires an existing infrastructure on any CI/CD stack.

For more information, see SAP Solutions for Continuous Integration and Delivery and Which SAP Solution for CI/CD Meets Your Needs?.

- If you want to have more control, especially when propagating changes towards your production environment, consider the following transport management options:
  - For cloud transports, also in hybrid landscapes: the **SAP Cloud Transport Management** service, which allows you to transport content archives, delivery units of SAP HANA XS classic model, and application-specific content, such as SAP BTP Integration content, between different subaccounts, which might also reside in different global accounts, spread across different regions. This service might be especially suitable if you don't have an ABAP-based landscape, don't want to involve on-premise ABAP systems in your scenario, or have to handle SAP BTP application-specific content, independent of whether it is provided as multitarget application files or in other formats.
    For more information, see SAP Transport Management Service.
  - For ABAP-centric or hybrid scenarios, the **enhanced Change and Transport System (CTS+)**, running in an on-premise ABAP system. Besides handling the transport of on-premise systems (such as of ABAP and Java artifacts), CTS+ also lets you transport SAP BTP artifacts from one subaccount to another. However, this is only supported if the artifacts are packaged as multitarget application archives. Please note that no other SAP BTP content format is planned to be supported by CTS+.
    For more information, see Transporting Multitarget Applications with CTS+.

○ Both CTS+ and SAP BTP Transport Management allow an integration into change management approaches around SAP Solution Manager. You can either run CTS+ centrally on SAP Solution Manager and integrate it with change management tools, such as Change Request Management (ChaRM) or Quality Gate Management – or you can integrate SAP BTP Transport Management service into these tools running on SAP Solution Manager (minimum version: SAP Solution Manager 7.2 SP10). It is even possible to use CTS+ and SAP BTP Transport Management service in parallel within ChaRM or QGM.

With this approach, you can set up synchronized transports in hybrid scenarios (involving both on-premise ABAP content and SAP BTP content). For more information, see Interplay of SAP Transport Management, CTS+ and ChaRM in hybrid landscapes.

Combine CI/CD with transport management (either in the form of SAP BTP Transport Management service or CTS+), to gain more control over the delivery of release candidates towards your production environment and at the same time benefit from the agility CI/CD brings to your development. For more information, see How to integrate SAP Transport Management into your CI/CD pipeline. It is also possible to combine CI/CD, CTS+, Transport Management and SAP Solution Manager Change Management.

- **Deploy applications manually** from your IDE to different subaccounts. Import or export your application using the SAP BTP cockpit, or use the Console Client for the Neo environment or the Cloud Foundry command line interface to deploy your application.

> **i Note**
>
> You can transport portal settings and configurations only by manually redeploying your application in each subaccount.

- In the Neo environment, consider to use the Solution Export Wizard that you can start in SAP BTP cockpit to model solutions by selecting comprised components (with an automated resolution of interdependencies) and to either export the outcome (as multitarget application file) or to directly handle the modeled solution via the change management options outlined above. For more information, see Solution Export Wizard - Ease Modeling and Export of Solutions as MTA.
- (HTML5 applications only) **Synchronize Git repositories** and create build scripts that let you transfer commits from the repository of the source subaccount to the repository of the target subaccount. After the synchronization, you must still deploy the application manually using the SAP BTP cockpit.
- (SAP HANA XS applications only) Use the **SAP HANA XS Application Lifecycle Management (HALM)** to create transport routes between two SAP HANA systems, then transport delivery units between those systems. For more information, see SAP HANA Application Lifecycle Management.

Related Information

Blog on Transport Approaches in SAP BTP

7.3   Implementing Failover

Execute the following tasks to implement a basic failover setup:

1. Deploy Your Application in Two Data Centers [page 83]
    Deploy your application in two data centers in parallel so that in case of an issue, you can switch from one to the other.
2. Keep the Two Applications in Sync [page 84]
    Synchronize your applications in both data centers to maintain their functionality in case of a downtime.
3. Define How a Failover Is Detected [page 88]
    Define in which cases the automatic failover from one data center to the other is triggered.
4. Decide on the Failback [page 89]
    In the setup of your failover scenario, define whether a failback is needed and how it is performed.

Related Information

Develop Resilient Applications [page 73]

7.3.1 Deploy Your Application in Two Data Centers

Deploy your application in two data centers in parallel so that in case of an issue, you can switch from one to the other.

To implement failover in an active/passive setup, deploy your application in two data centers in parallel. Among these two data centers, define the primary (active) one to which the traffic is generally directed and the secondary (passive) data center that takes over in case of a downtime.

Depending on your type of global account on SAP BTP and the environment you're using, you can choose from different regions to deploy your applications. To deploy an application in subaccounts of multiple regions, execute the deployment separately for each subaccount. For more information about regions and a complete list of data centers from which you can choose, see Regions.

> → Recommendation
>
> To optimize the performance of your application, which is, for example, response time and latency, we recommend that you deploy it in two data centers of the region your target audience and your back-end system are located in. If you are located in Europe, choose, for example, the data centers in Frankfurt and Amsterdam.

> **i Note**
>
> If you want to deploy your application in two data centers that are not located in the same region, keep in mind that there might be issues concerning legal data processing restrictions. For more information, see Data Protection and Privacy.

Parent topic: Implementing Failover [page 83]

Next: Keep the Two Applications in Sync [page 84]

7.3.2 Keep the Two Applications in Sync

Synchronize your applications in both data centers to maintain their functionality in case of a downtime.

Make sure that you transfer changes on one of your applications to the one in the other data center, as well, so that its functionality is not interrupted in case of a failover.

> **→ Recommendation**
>
> Keeping your two applications in sync does not necessarily mean that both applications need to be identical. If you want to offer only a reduced set of functions in your secondary application, we recommend to visually mark it as your backup. In this way, your users are constantly reminded that they are using the backup version and should switch back to the primary one once it is available again.

There are three different ways to synchronize your applications in the primary and the secondary data center:

- Orchestrate your applications manually. See Synchronize Your Applications Manually [page 84].
- Orchestrate your applications with the help of a continuous integration and delivery pipeline. See Use a Continuous Integration and Delivery Pipeline [page 85].
- Orchestrate your applications through a combination of the Solution Export Wizard and the SAP Cloud Transport Management service. See Use the Solution Export Wizard and SAP Cloud Transport Management [page 86].

Parent topic: Implementing Failover [page 83]

Previous: Deploy Your Application in Two Data Centers [page 83]

Next: Define How a Failover Is Detected [page 88]

7.3.2.1   Synchronize Your Applications Manually

You can orchestrate your applications manually by duplicating your modifications in both data centers, for example, by mirroring from Git repositories.

The following figure illustrates this process:
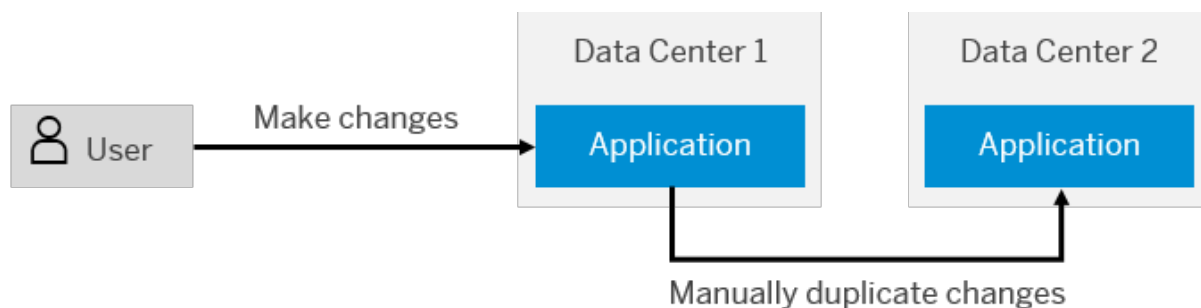
Manual Synchronization

With this approach, you can decide yourself which of your changes to transport from one data center to the other. Therefore, synchronizing your applications manually allows you to maintain two non-identical applications, for example, if you want to visually differentiate between the UIs in your primary and the backup version.
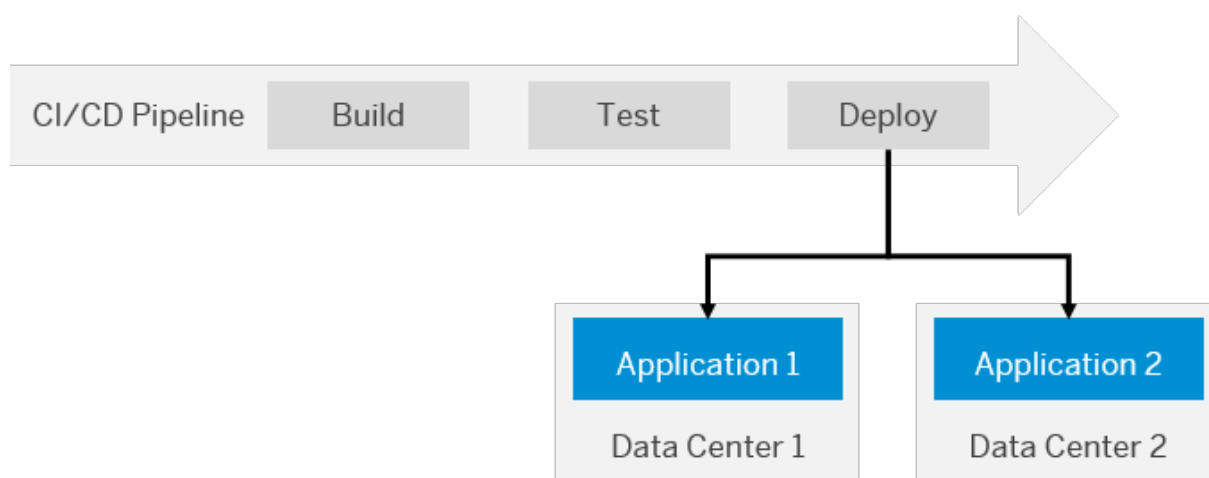
> → Recommendation
>
> With regard to the relatively high effort needed for this approach, we recommend it only if you do not plan regular updates or changes to your application.

7.3.2.2 Use a Continuous Integration and Delivery Pipeline

You can synchronize the deployment of your applications in two different data centers by using a continuous integration and delivery (CI/CD) pipeline. Configure it for multi-deployment so that in the final stage of your CI/CD pipeline, your changes are automatically deployed to two subaccounts in parallel. In our case, these subaccounts are hosted in different data centers. For more information about subaccounts, see Managing Subaccounts.

The following figure illustrates this procedure:

Synchronized Deployment Through a CI/CD Pipeline

> **→ Recommendation**
>
> Use a pre-configured pipeline of project "Piper" and adapt it for multi-deployment to your two subaccounts in different data centers. For more information, see project "Piper" ↗ .

Related Information

[Continuous Integration (CI) and Delivery (CD) Overview](#)
[Overview of SAP Offerings for CI and CD](#)
[Blog - Continuous Delivery with Jenkins Pipelines](#)↗

7.3.2.3   Use the Solution Export Wizard and SAP Cloud Transport Management

You can orchestrate your applications through a combination of the solution export wizard and the SAP Cloud Transport Management service.

Prerequisites

- You have deployed the applications you want to synchronize in two different subaccounts in the Neo environment. See [Create Subaccounts Using the Cockpit](#).
- You have a subaccount in the Cloud Foundry environment. See [Create Subaccounts Using the Cockpit](#).
- You are subscribed and have access to the SAP Cloud Transport Management service. See [SAP Cloud Transport Management](#)↗ and [Initial Setup](#).
- In SAP Cloud Transport Management, you have defined a transport route from your development system to your failover system. See [Configuring the Landscape](#).

Context

In your subaccount in the Neo environment hosted in your primary data center, use the solution export wizard to export the changes you have made to your application to the SAP Cloud Transport Management service. From the Transport Management service, which is hosted in your subaccount in the Cloud Foundry environment, transport your changes to the backup application in your subaccount in the Neo environment in the secondary data center.

The following figure illustrates this procedure:

Synchronization Through Solution Export Wizard and Transport Management
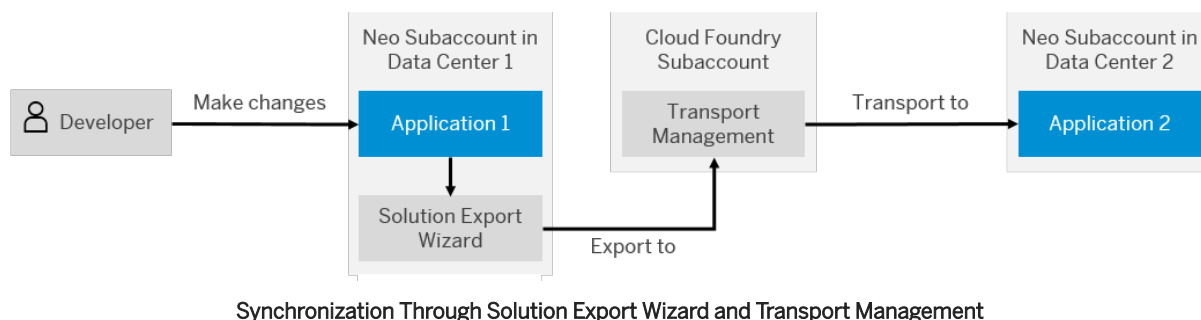
> **i Note**
>
> You can also use the solution export wizard independently from SAP Cloud Transport Management by manually uploading your exported changes to the target account.

Procedure

1. In the cockpit, navigate to your subaccount in the Neo environment, in which your HTML5 application is deployed. See Navigate to Global Accounts and Subaccounts in the Cockpit.

2. From the navigation pane, choose ▣ *Solutions*.

3. To access the solution export wizard, choose ▣ *Export*. For more information, see Exporting Solutions.

4. In the *Select Components* section of the *Solution Export Wizard* dialog, select the components of your HTML5 application you have changed.

   > **i Note**
   >
   > The solution export wizard can automatically resolve dependencies, for example, by selecting corresponding roles.

5. In the *Export options* section, select *Transport Management Service* to hand over your MTA archive to SAP Cloud Transport Management. For more information about MTA archives, see Multitarget Applications.

6. In the cockpit, navigate to your subaccount in the Cloud Foundry environment. See Navigate to Global Accounts and Subaccounts in the Cockpit.

7. From the navigation pane, choose ▣ *Subscriptions*.

8. On the ▣ *Transport Management Service* tile, choose *Go to Application*.

9. From the navigation pane, choose ⤴ *Transport Nodes* and select the target node.

10. In the *IMPORT QUEUE* tab, select the MTA archive you have exported with the help of the solution export wizard.

11. Choose *Import*.

Related Information

[Blog - Solution Export Wizard – ease modeling and export of solutions as MTA](#)☁

[Blog - The new cloud-based Transport Management Service](#)☁

[Blog - SAP Cloud Transport Management service generally available](#)☁

7.3.3 Define How a Failover Is Detected

Define in which cases the automatic failover from one data center to the other is triggered.

You can choose from different test categories to decide when the automatic switch from your primary to the backup application is triggered. Either define these test categories manually in your code or use rule-based performance solutions such as Akamai ION ⬈ . At each user request, you could, for example, check for a combination of a predefined maximum read timeout and set HTTP status codes. If one of these two checks responds an unwanted behavior of your application, the failover is triggered.

The following figure illustrates this procedure:

Failover Detection
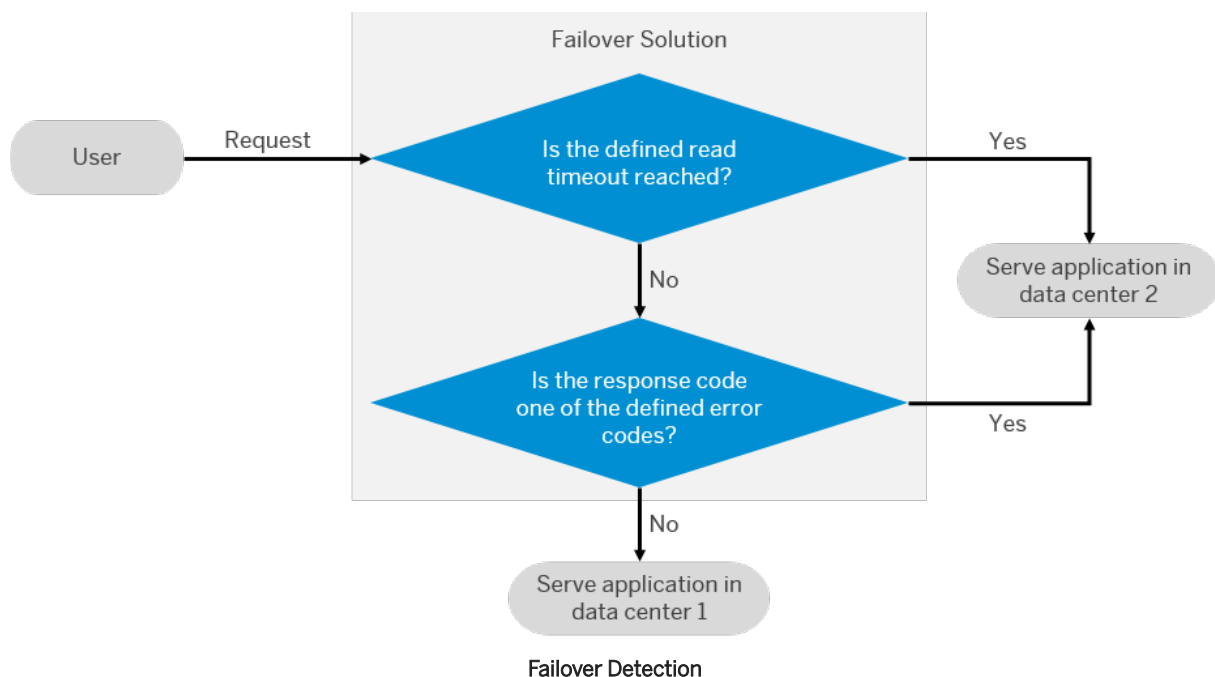
> ⸭ Example
>
> As an example, you could define 25 seconds as maximum read timeout and set a number of server error codes (5xx) to be checked. At each first HTTP request towards the URL of your application, both checks are triggered. All following HTTP requests are ignored to avoid an unnecessary failover if, for example, only an image resource is missing. If one of the two checks fails, the user is automatically redirected to an HTML down page, which must not be cached. This down page provides the link to your failover application. Strictly speaking, the failover itself is therefore manually performed by the user of your application.

7.3.4 Decide on the Failback

In the setup of your failover scenario, define whether a failback is needed and how it is performed.

Depending on the setup of your failover scenario, a failback is either optional or mandatory:

- If you use an **active/active setup**, your applications in both data centers must be identical and provide the exact same functionality. In this setup, both data centers are of equal value. Therefore, a failback to the primary data center is not needed but performed automatically the next time a failover is triggered.
- If you use an **active/passive setup** and your applications in both data centers slightly differ from each other, a failback to the primary data center is mandatory to provide the full functionality of your application.

> → Recommendation
>
> In a basic scenario, you can hand over the failback to the users of your application, who perform the failback automatically when later trying to access the primary version of your application again. In this scenario, visually differentiate between your primary and secondary application so that they are constantly reminded to switch back to the primary one once they have completed their transaction. Unlike an automatic failback as soon as the primary data center is available again, this approach ensures that your users are not interrupted while completing a transaction. Instead, they can decide on their own when to switch back to the primary application.

7.4 Configuring Backup

Backup and recovery of data stored in the following services are performed by SAP. For other services, you can follow our best practices to back up your configurations.

7.4.1 Data Backups Managed by SAP

SAP handles the backup and recovery of service data. However, there are some exceptions.

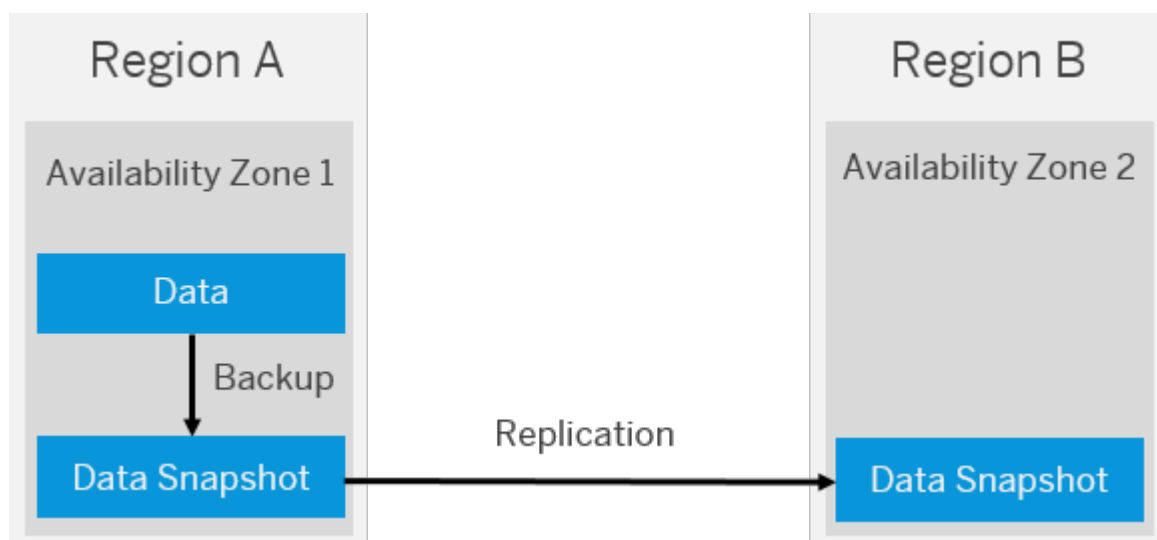### Services with Automated Data Backups

Data stored in the following services is automatically backed up by SAP:

- **SAP HANA Cloud**: SAP HANA Cloud instances are continually backed up to safeguard your database and ensure that it can be recovered. For more information, see Backup and Recovery.
- **PostgreSQL on SAP BTP, Hyperscaler Option**: SAP keeps backups of your PostgreSQL database instances for a retention period of 14 days. You can restore a database instance to any point in time within the backup retention period by creating a new database instance. For more information, see Restore Database Instance to a Specified Time.
- **SAP BTP, Kyma environment**: The Kyma environment relies on managed Kubernetes clusters for periodic backups of Kubernetes objects. For more information, see Kyma Environment Backup.

Data stored using these services and components is fully backed up on a regular basis. For services running in the SAP BTP, Cloud Foundry environment, the backed-up data is copied to additional availability zones within the same region. Each availability zone represents a physically separate location with its own power supply, network, and cooling. For more information, see Regions.
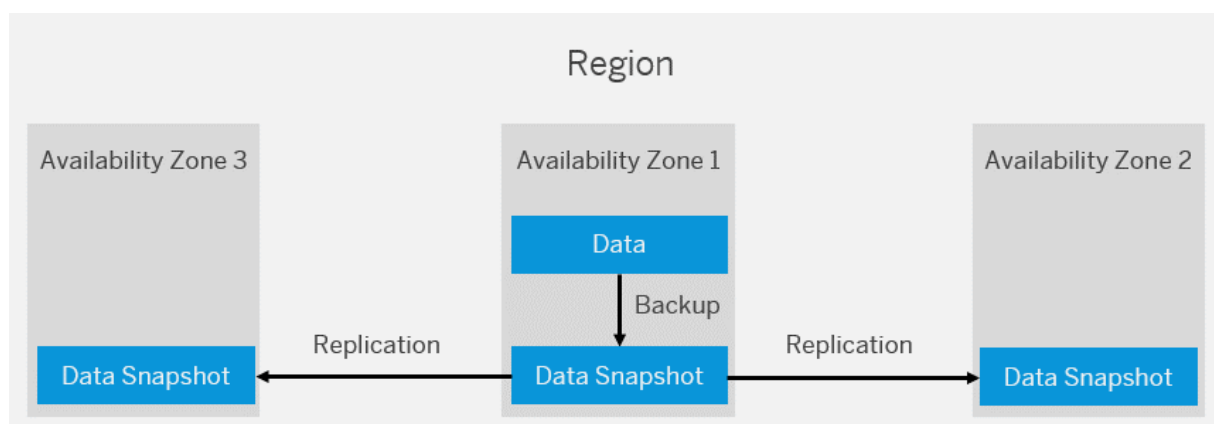
For services running in the SAP BTP, Neo environment, the backed-up data is copied to an availability zone in another region. For more information, see Regions in the Neo Environment.

Services Without Automated Data Backups

The following services don't currently provide any backup and restore features:

- **Redis on SAP BTP, hyperscaler option**: This service doesn't support persistence.
- **Object Store on SAP BTP**: There's no backup and restore feature provided by the Object Store service. However, there are additional features that could help in backup and restore in certain scenarios:
  - Object Versioning allows you to recover from accidental deletion and modification of objects because the hyperscalers keep older versions in case the objects are modified or deleted.
  - Object Expiration Rules helps with the automatic deletion of older versions.
  - The **prevent deletion** feature helps preventing your blob stores from accidental deletion. For more information, see:
    - Prevent Accidental Deletion of AWS S3 Buckets
    - Preventing Accidental Deletion of Azure Containers

7.4.2 Service Configurations Managed by Customers

SAP doesn't manage backups of service configurations. However, you can back up your service-specific configuration yourself. This backup allows you to restore your configuration in case it's deleted by accident.

Services with User-Specific Configurations

Different configurations can be backed up, depending on your specific service. Here are some examples:

- **SAP Connectivity service**
  - You can back up your Cloud Connector configuration. For more information, see Configuration Backup.

- You can back up your destination configuration. For more information, see Export Destinations.
- You can back up your X.509 certificates. For more information, see Set up Trust Between Systems.
- **Process Visibility capability within the SAP Workflow Management service**: You can back up your visibility scenarios. For more information, see Export a Scenario.
- **SAP Authorization and Trust Management service**: You can back up your configurations with the SAP Authorization and Trust Management service API package available on the SAP API Business Hub. The configurations include:
  - Roles. You can back up role configurations with the Authorization API.
  - Role collections and role collection assignments. You can back up configurations for role collections and role collection assignments with the System for Cross-domain Identity Management (SCIM) API.
  - Identity provider configurations. You can back up identity provider configurations with the Identity Provider Management API.
- **SAP Cloud Integration**: You can back up keystore entries. For more information, see Backing Up Keystore Entries.
- **SAP Alert Notification service for SAP BTP**: You can back up your subscriptions, conditions, and actions. For more information, see Exporting or Importing Configurations.
- **Identity Provisioning service**: You can back up your systems. For more information, see Export and Import Systems.

If your service has configurations that can be backed up, you can find more details in the **Administration** section of your service documentation. This section provides links to detailed instructions on how to back up and restore your configurations.

Services Without User-Specific Configurations

There's no backup for user-specific configurations available for the following services:

- **SAP HTML5 Application Repository service for SAP BTP**
- **Identity Authentication service**
- **SAP Credential Store service**
- **Workflow capability within the SAP Workflow Management service**
- **SAP Audit Log service**
- **SAP Application Logging service for SAP BTP**

8 Integrate and Test

Test and integrate your application with other solutions.

8.1 Integrating Your Application

SAP BTP offers various options for integrating your cloud application with other cloud and on-premise solutions.

Cloud Connector

If you want an application running on SAP BTP to access data from your on-premise back-end system, we recommend that you use the Cloud Connector. It's an on-premise piece of software that you'll need to install in your on-premise landscape, within the firewall. Once it's configured and paired with your SAP BTP subaccount, a secure tunnel is established between SAP BTP (and all the services and applications that run on it) and the Cloud Connector. Communication between SAP BTP and the back-end system is routed via the Cloud Connector through a secure SSL tunnel.

We recommend that you use the Cloud Connector if your use case requires any of the following:

- Point-to-point connectivity and programmatic integration
- Connectivity with cloud applications from on-premise systems, without mediation, such as for mapping, routing, or security
- Data replication from on-premise databases or Business Intelligence tools to SAP HANA databases running in the cloud
- On-premise system integration using SAP Cloud Integration

For more information, see Cloud Connector.

SAP Cloud Integration

SAP Cloud Integration (Cloud Integration) supports end-to-end process integration across cloud and on-premise applications (cloud to cloud and cloud to on-premise integration). It provides the following features:

- Core runtime for processing, transforming, and routing of messages to be exchanged between your systems
- Out-of-the-box connectivity support (IDoc, SFTP, SOAP/HTTPS, SAP SuccessFactors, OData, HTTPS)
- Security features such as content encryption and certificate-based communication

Cloud Integration offers full flexibility for the manner in which messages are exchanged between your systems. We recommend that you use the Cloud Connector if you require any of the following (or if you don't require on-premise-based middleware, such as SAP Process Orchestration):

- Compliance scenarios, such as e-invoicing and payroll
- Graphical model of integration flow
- More than a few systems to be integrated, and you'd like to managed integrations centrally, rather than having to manually code applications when connections change
- Individual messages to be processed during integration
- Integration with third-party solutions

For more information, see SAP Cloud Integration.

Cloud Integration Automation Service

If you're planning on integrating your SAP BTP application into a hybrid landscape, you can also leverage the Cloud Integration Automation Service that is offered for selected integration scenarios. This service provides a guided workflow that uses customer-specific system information, reusable configuration settings between tasks, and automated execution capabilities, thereby reducing the manual work for your scenario integration. For more information, see Cloud Integration Automation Service.

Related Information

Blog Post: SAP Cloud Integration: When to use which tool
Blog Post: Using SAP Analytics Cloud Connector with SAP Cloud Integration
Blog Post: Cloud Integration Automation Service - What is it?
Blog Post: Cloud Integration Automation Service: Step-by-Step

8.2   Performing Integration Tests

In contrast to unit tests that are performed locally on your development subaccount without any external dependencies in the develop and build phase, integration tests target the interplay of a complete system, spanning several single components and units potentially spread throughout a hybrid landscape. Integration tests verify that all building blocks work together, meet specified requirements, and fulfill the targeted business case. They should therefore then be executed on your test infrastructure or in your test subaccount after the integration into an existing landscape has taken place. This ensures that the interplay with backend functionalities is verified. In a CI/CD setup, note that such integration tests are part of the pipeline.

Integration Tests for SAPUI5

One Page Acceptance Tests (OPA5) is an API for SAPUI5 controls. It hides asynchronicity and eases access to SAPUI5 elements. This makes OPA especially helpful for testing user interactions, integration with SAPUI5, navigation, and data binding. See Integration Testing with One Page Acceptance Tests (OPA5).

9   Go Live and Monitor

Once you've deployed, transported, tested, and integrated your application, you can go live.

After you've released an application, you have to ensure that it's provided with the right performance and availability by monitoring its various parts – such as platform services, application parts, and databases. SAP BTP offers various native tools for monitoring and operating your application, optionally complemented by third-party offerings, in case you need deep monitoring of cloud-native applications.

For hybrid scenarios across the SAP portfolio, or if you already have an operations process in place, you can also integrate operation aspects of SAP BTP into strategic operation platforms (such as SAP Solution Manager, SAP Focused Run of SAP Solution Manager, and SAP Cloud ALM).

9.1   Going Live

Once you've tested your application successfully and ensured that you are compliant with any applicable security guidelines and compliance regulations, you can go live with your application.

If you've set up a staged development environment, we recommend that your Cloud Administration Team defines specific timeframes during which the Cloud Development Team is allowed to go live with an application. You can forbid go-live activities during times that a stable landscape is particularly crucial (for example, at the end of each quarter), and allow deployments to your production subaccount only in case of emergencies.

Consider embedding applications with internal end users in the SAP Fiori launchpad using SAP BTP Portal before you go live. This enables your internal target audience to access all applications in one central place and thus improves the app's usability. For more information, see SAP Cloud Portal Service ➤ .

Manage Authentication and Authorization

Ensure that the business users of your application are being provided by using SAP ID service or configuring the trust relationship to an external identity provider. They should also get the right authorization by using application-based authorization artifacts provided by the developers. This allows the administrators to create roles, build role collections, and assign these collections to business users and user groups.

For more information, see Security Administration: Managing Authentication and Authorization .

Web Acceleration

If your application's end users are widely spread across different countries or even continents, you can improve your application's load performance by using a Content Delivery Network (CDN). For example, you can use common CDN providers such as Akami or Cloudflare.

If you include SAP's SAPUI5 library directly from *.hana.ondemand.com*, the SAPUI5 resources are automatically delivered via a CDN. See Variant for Bootstrapping from Content Delivery Network.

Consider the following when using a CDN:

- Secure transport: Make sure any access via HTTP will get redirected to a HTTPS connection before loading any content.
- Block access based on location: If blocking access from a certain country is required, the client's location data can be used. Keep in mind that this does not guarantee a full blocking, since the client's location data can be changed, or the client could access from another country via a virtual private network (VPN).
- Content compression: You can compress your content with gzip before it gets delivered to the client. This improves the performance, especially for clients with a slow connection.
- Content caching: In addition to client caching, you can make the CDN provider cache your server's content, so that following requests of the same resources by other clients will be delivered faster. While this can additionally improve the performance, you need to keep in mind the following:
  - Make sure that you only cache static content. We recommend that you exclude certain files (for example, for UI5 apps, exclude the neo-app.json file) or paths (for example, a route to OData services), to make sure dynamic content is never cached.
  - You should configure your CDN provider so that it respects the `Cache-Control` and `Expires` header of your server.
  - You should not cache any dynamic header, such as the `X-CSRF-Token` header, that is used against cross-site request forgery (CSRF).
- See SAP Note 2943781 for information about using CDN for on-premise systems.

9.2 Monitoring Applications and Services

There are various options you can use to monitor applications and services on SAP BTP, provided natively by the platform itself. Which of them you might use depends on the SAP BTP environments you work in.

Monitoring Applications in the Cloud Foundry Environment

In the Cloud Foundry environment, you can use the SAP Application Logging service for SAP BTP to access logs from your applications. You can also run the Cloud Foundry CLI command `cf logs` to show the recent logs for an application. For more information, see the Cloud Foundry documentation at https://docs.cloudfoundry.org/devguide/deploy-apps/streaming-logs.html .

> i Note
>
> Container metrics in syslog drains are not currently logged by SAP BTP, but you can use the SAP Application Logging service to access container metrics of your applications.

You can also configure health checks that continually monitor the status of a running application. For more information, see https://docs.cloudfoundry.org/devguide/deploy-apps/healthchecks.html .

Monitoring Applications in the Neo Environment

In the Neo environment, you can use the cockpit to monitor applications:

Monitoring Applications Using the Cockpit in the Neo Environment

Application Type	Monitoring Options	More Information
Java applications	• Application monitoring • Application performance statistics (Beta) • Custom JMX checks • Monitoring APIs	Managing Deployed Applications
SAP HANA XS applications	• Health statistics for SAP HANA database instances • Application availability checks • Email notifications	SAP HANA: Application Operations
HTML5 applications	Log viewer collecting error messages	Using Logs in the Cockpit for HTML5 Applications

You can also connect your Java applications to a **Dynatrace SaaS monitoring environment**. For more information, see Agent Activation for Dynatrace.

Monitoring Platform and Service Availability

You can follow the availability of the platform at SAP Trust Center . You can check:

- the availability by service on the *SAP BTP* tile of the *Cloud Status* tab page;
- the availability by region on the *Data Center* tab page.

To get notifications for updates and downtimes, subscribe at the Cloud System Notification Subscriptions application. Create a subscription by specifying Cloud Product, Cloud Service, and Notification Type. For more information, see Cloud System Notification Subscriptions User Guide .

In addition, you can get a personalized, at-a-glance view of additional SAP BTP offerings with SAP Cloud Availability Center , such as SAP BTP Integration.

There are additional monitoring tools and options available for certain SAP BTP services, such as the SAP BTP Identity Authentication service. Please refer to the respective service documentation for more details: Solutions and Services.

Alerting

In addition to the general notifications mentioned above, Alert Notification for SAP BTP service lets you know instantly whenever there's an issue with your specific cloud application or its dependencies, no matter if your

apps are running in the Neo or Cloud Foundry environment. You can subscribe to events coming from used SAP BTP services, from hyperscalers, from third-party tools such as Dynatrace, or you can create custom alerts for your apps. Furthermore, the notifications arrive via any channel of alert management you select, for example, email, some sort of corporate chat, a ticketing system, or even SAP Solution Manager, Focused Run for SAP Solution Manager , or and SAP Cloud ALM . For more information, see SAP Alert Notification Service.

9.3 Monitoring a Hybrid Landscape

If you've integrated SAP BTP with your on-premise SAP landscape, you can use SAP Solution Manager, Focused Run for SAP Solution Manager, and, for more and more options and scenarios also SAP Cloud ALM to monitor and operate that hybrid landscape.

For example, the table below lists monitoring options offered by SAP Solution Manager.

Monitoring Option	Description	More Information
Integration Monitoring	Ensure reliable data exchange between your SAP on-premise system and SAP BTP. Includes process integration monitoring, interface and connection monitoring, and message flow monitoring.	Integration Monitoring
User Experience Monitoring	Simulate the behavior of users who access central servers at different locations to run business processes. Lets you monitor the availability of the systems, and connection performance, from the end-user perspective, in real time.	User Experience Monitoring
Trace Analysis	Trace the performance of SAP BTP applications. You can: - Trace requests to applications that are deployed on SAP BTP or running on your on-premise system - Trace application performance data, such as application and UI response times, and database calls - Trace request paths through inbound and outbound request points	Trace Analysis End-to-End Trace Analysis (Neo environment)

Monitoring Option	Description	More Information
Exception Management	Forward business-critical exceptions from SAP BTP to on-premise operations and: • Centrally retrieve and handle all single, cross-components, process flow-driven, and multistep exceptions. • Correlate, analyze, and process exceptions, and use predefined guided procedures to quickly handle errors.	Exception Management

See also SAP Cloud ALM for operations .

Related Information

Blog: Brief Overview of Hybrid Supportability Options for SAP BTP

10   Improve and Retire

Once your application has gone live, we recommend that you continually ensure its quality. If you determine that you no longer need it, you should retire it.

10.1   Improving and Maintaining Your Application

Once an application has gone live, the Cloud Development Team is responsible for housekeeping and ongoing improvements.

The team should:

- Regularly verify and test the application to avoid issues caused by library and platform updates, such as a new SAPUI5 version
- Ensure that the application remains compliant with all applicable standards and guidelines
- Fix bugs in a timely manner, and ensure that the user experience remains of high quality and is improved and adapted over time
- Listen to user feedback
- Set up alerting that informs you automatically about issues with your application, for example by using SAP Alert Notification Service
- Identify recurring operation tasks and automate them, such as by using SAP Automation Pilot. This services provides catalogs of automated technical operation tasks for handling the lifecycle of apps running on SAP BTP, such as for alert remediation and recurring DevOps tasks.
- Adapt/extend your application, as required:
  - For deploying new application versions in the Cloud Foundry environment, consider using a blue-green deployment, in which you deploy a new application version in parallel to the previous version to test it and to route to the new version only after successful testing and without downtime – for more information, see Multitarget Application Commands for the Cloud Foundry Environment on SAP Help Portal.
  - In the Cloud Foundry environment, consider using the Feature Flag service that lets you enable or disable specific features in a production solution, for testing purposes, without having to redeploy or restart the solution.

> **→ Recommendation**
>
> To stay up to date, we recommend that you regularly check the Release Notes and the SAP Community.

10.2 Retiring Your Application

If your application is no longer needed, the Cloud Administration and the Cloud Development Teams should ensure that it is properly retired, and that all data retention requirements are met.

10.3 Migrating from Neo to Multi-Cloud

SAP is integrating more closely with hyperscalers, such as Amazon Web Services, Google Cloud Platforrm, and Microsoft Azure. If you are currently running secnarios in our Neo enviroment, please have a look at our migration guide for moving from Neo to the multi-cloud foundation.

Migrating from the Neo Environment to the Multi-Cloud Foundation (Cloud Foundry and Kyma)

Important Disclaimers and Legal Information

Hyperlinks

Some links are classified by an icon and/or a mouseover text. These links provide additional information.
About the icons:

- Links with the icon 🡕 : You are entering a Web site that is not hosted by SAP. By using such links, you agree (unless expressly stated otherwise in your agreements with SAP) to this:

    - The content of the linked-to site is not SAP documentation. You may not infer any product claims against SAP based on this information.
    - SAP does not agree or disagree with the content on the linked-to site, nor does SAP warrant the availability and correctness. SAP shall not be liable for any damages caused by the use of such content unless damages have been caused by SAP's gross negligence or willful misconduct.

- Links with the icon 🖎: You are leaving the documentation for that particular SAP product or service and are entering a SAP-hosted Web site. By using such links, you agree that (unless expressly stated otherwise in your agreements with SAP) you may not infer any product claims against SAP based on this information.

Videos Hosted on External Platforms

Some videos may point to third-party video hosting platforms. SAP cannot guarantee the future availability of videos stored on these platforms. Furthermore, any advertisements or other content hosted on these platforms (for example, suggested videos or by navigating to other videos hosted on the same site), are not within the control or responsibility of SAP.

Beta and Other Experimental Features

Experimental features are not part of the officially delivered scope that SAP guarantees for future releases. This means that experimental features may be changed by SAP at any time for any reason without notice. Experimental features are not for productive use. You may not demonstrate, test, examine, evaluate or otherwise use the experimental features in a live operating environment or with data that has not been sufficiently backed up.
The purpose of experimental features is to get feedback early on, allowing customers and partners to influence the future product accordingly. By providing your feedback (e.g. in the SAP Community), you accept that intellectual property rights of the contributions or derivative works shall remain the exclusive property of SAP.

Example Code

Any software coding and/or code snippets are examples. They are not for productive use. The example code is only intended to better explain and visualize the syntax and phrasing rules. SAP does not warrant the correctness and completeness of the example code. SAP shall not be liable for errors or damages caused by the use of example code unless damages have been caused by SAP's gross negligence or willful misconduct.

Bias-Free Language

SAP supports a culture of diversity and inclusion. Whenever possible, we use unbiased language in our documentation to refer to people of all cultures, ethnicities, genders, and abilities.

THE BEST RUN **SAP**