Redbooks
ibm.com/redbooks

IBM Cloud Provisioning and Management for z/OS An Introduction

Keith Winnard
Gary Puchkoff
Hiren Shah

Cloud

z Systems

IBM

Redpaper

Find and read thousands of IBM Redbooks publications

- Search, bookmark, save and organize favorites
- Get personalized notifications of new content
- Link to the latest Redbooks blogs and videos

Get the latest version of the Redbooks Mobile App

iOS **Download Now** Android

Promote your business in an IBM Redbooks publication

Place a Sponsorship Promotion in an IBM® Redbooks® publication, featuring your business or solution with a link to your web site.

Qualified IBM Business Partners may place a full page promotion in the most popular Redbooks publications. Imagine the power of being seen by users who download millions of Redbooks publications each year!

ibm.com/Redbooks
About Redbooks → Business Partner Programs

THIS PAGE INTENTIONALLY LEFT BLANK

Cloud Provisioning and Management Overview

This IBM® Redpaper™ publication introduces you to IBM Cloud Provisioning and Management for z/OS®. The paper was written to coincide with the announcement of IBM Cloud Provisioning and Management for z/OS, and the software is not yet generally available (GA) at the time of this writing. As such, any examples that are shown are subject to change at any time before the software is GA. The goal of this paper is to provide you with an initial overview and more information to support the announcement.

You might be new to clouds, or might have heard of many different variations of what a cloud might be. Therefore, it is useful to begin with establishing an understanding of cloud computing concepts.

The paper describes the provisioning and management of z/OS middleware and explains how these aspects can be offered as a cloud service.

This paper includes the following topics:

Defining and scoping cloud services on z/OS

The approach to provisioning and managing z/OS as cloud services is not focusing on the provisioning of operating system instances, but rather the ability to provision multiple workloads in a single z/OS instance. It takes the responsibility of resource orchestration and mapping resources to be automatically provisioned for middleware instances.

We make the assumption that the z/OS operating system is part of the infrastructure and does not need to be instantiated to deploy a workload in a development, test, or production environment.

Provisioning a new instance of the z/OS operating system in a new LPAR is an important requirement, but is beyond the scope of this Redpaper publication.

Cloud computing

The National Institute of Standards and Technology (NIST) provides the following definition[1] for cloud computing:

> "*Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (for example, networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.*"

Users interact with cloud computing environments with the services that the cloud environment provides. The following examples are services that typically are provided by a cloud:

- Virtual servers
- Database services
- Middleware instances
- Email applications
- Storage

Your organization can use cloud services that are provided by third parties, or you can build your own cloud. If you build your own cloud, you can provide services from your cloud to internal company users, lines of business, selected business partners or customers, or to the world at large.

You might also choose to have a mixture of the two, in which some of the cloud services are internal and accessible through your intranet, and other services are external and accessible by using the Internet.

Cloud services characteristics

Based on the NIST definition, the characteristics and models that constitute a cloud service are shown in Figure 1.

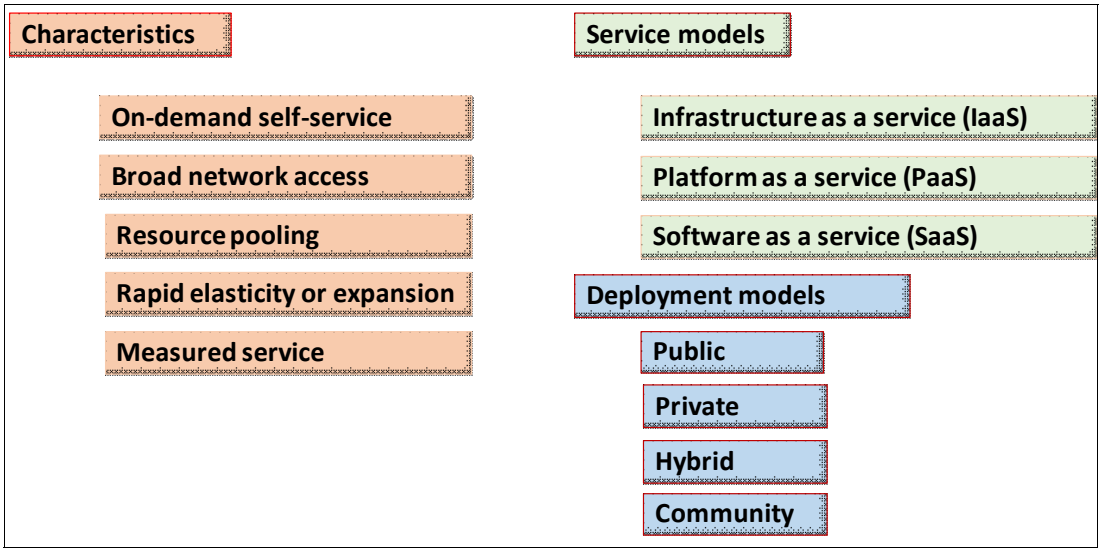

Figure 1 Cloud characteristics and models

[1] For more information, see NIST Special Publication (SP) 800-145, A NIST Definition of Cloud Computing: http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf

For a service to be considered a “cloud service,” it must include the following characteristics:

- Support self-service provisioning
- Be accessible through the Internet or corporate intranet
- Provide resources from a resource pool, without the user needing knowledge of the pool
- Provide simple and fast resource elasticity as users demand changes
- Monitor resources with a dashboard view on cloud health status and potentially support a metering capability, which enables a dynamic charge-back model

To provide these characteristics, the infrastructure that enables the cloud services uses the following key enablers:

- Virtualization

  Allows computing resources to be pooled and allocated on demand. It also enables pay-per-use billing to be implemented.

- Automation

  Enables the elastic use of available resources and workloads to be moved to where resources are available. It also supports provisioning and deprovisioning of service instances to support scalability.

Although these enablers are not part of any formal cloud definition, they are indispensable in delivering the essential cloud service characteristics.

Many traditional IT services are provisioned with the characteristics of a cloud service. For example, as systems programmers you might create a development and testing environment for a particular application project, as shown in Figure 2.

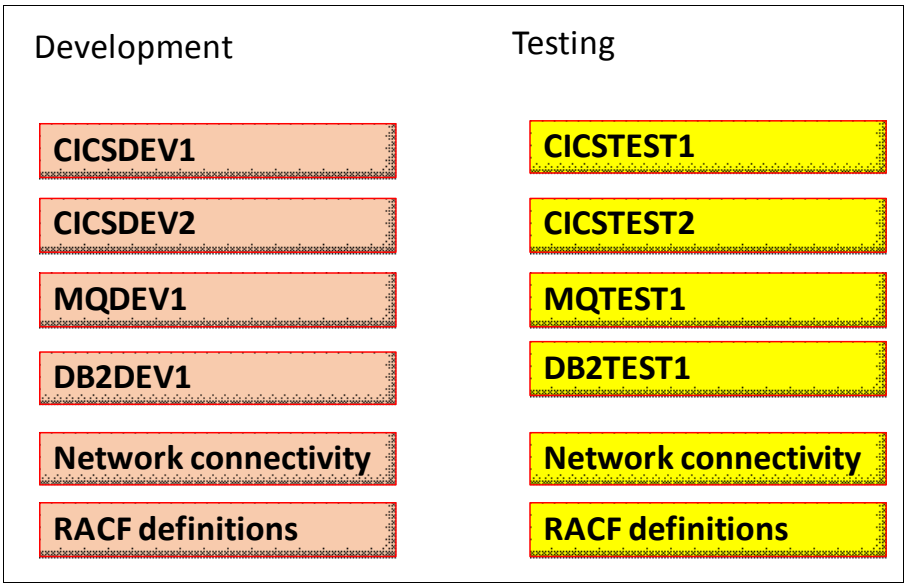

Figure 2 Sample development and test environment

Is this type of provisioning considered a cloud service? The application project leader (user) might submit a work request for these environments to be built. When the systems programmers (provisioners) plan and build the components, they hand these components over to the application project leader. Is this service a type of cloud service? In context of the NIST definition, the answer is no.

How do you know that you are providing or using a cloud service, or when you are using a cloud service?

You know that you are *providing* or *using* a cloud service when your service exhibits the characteristics that are listed in Figure 1 on page 2 and it is provisioned by using the virtualization and automation enablers.

From a cloud user perspective, it is important that you can perform self-service tasks that relate to the cloud service you are using to quickly provision new service instances. Resources that are elastically sized to meet your changing processing demands also are needed.

In a traditional z/OS environment, many tasks to support the development and support of business applications and the infrastructure require expertise that is in disparate teams. It is a joint effort that requires the skills of different areas to be focused on a common deliverable, such as a new project. In this instance, the skill factor can be high for each task, but the use of well-defined local standards suggests that the tasks are repetitive (with different values passing through them, such as IP addresses). Therefore, it is likely each of the areas feature their own process to deliver their own part of the project. The question becomes: How many of the tasks can be delivered as a cloud service?

To understand the role and scope of a cloud service, consider the questions that are listed in Table 1 and apply them to the project environments that are shown in Figure 2 on page 3. The questions are set from the perspectives of the provisioner and user.

In this instance, you might regard the provisioners as the systems programmers, security administrators, and other infrastructure support roles. The users might be the application development project leader, programmers, testers, or someone from the business. They might even be a member of the infrastructure support.

Table 1 Cloud Services characteristics -related questions

Provisioners	**Users**
How much information did I need to perform the set up tasks that were assigned to me?	How much of each environment did I self-provision?
How is the information given to me?	What information did I need to give to the provisioners?
How long does it take me to set up each component?	How much information did I need to give?
How much validation do I manually complete to ensure that the components are working?	How much information that I give is common to other projects?
Who needed to be involved in setting up the components?	For how long do I have access to the environment?
How many hours are used in meeting with the users to understand what is required?	Will I receive any confirmation of how much resource I used?
How long did it take to set up each component?	After the project is running, how much self-service can I have then? For example, can I add a new user to the project or do I have to request a change?
How many activities were required for each component?	How much resource time was used in providing the information for the project?

Provisioners	Users
How long did it take to set up all the environment?	How much research did I have to complete to supply the infrastructure provisioning and support the team's infrastructure with the information they require to set up the components?
How many activities were required for each component?	What if I want to add IBM WebSphere® Application Server or another IBM CICS® region?
Is there a standard procedure for setting up each element?	Was the environment backed up? Can I reset the environment whenever I want?
How many changes were raised for this environment to be set up?	Do I need to read and understand all of the standards documents to set up my own environment?
How many sign-offs were required for each component before the environment was handed over to the users?	What performance or turnaround can I expect from this environment?
How quickly can I reset or rebuild the environment if there is a major issue?	Where is the information that shows the settings and options for this environment?
How many maintenance activities are set up to maintain the environment?	When I am finished with this environment, can I remove it?
How many IP addresses did I need?	Do I need to set up meetings with the provisioners to discuss changes and refinement?
Does the environment fit in the correct place in terms of performance requirements?	How do I know what I was given?

You might have many more questions to add to this table. The answer to the questions in this table vary from person to person. The aim of asking these questions is to explore whether the provisioning falls into a cloud service because of the characteristics and enablements of the provisioning.

An application development project's lifecycle can be much more than a single development and single test environment. As shown in Figure 3, more environments are added. These environments might include the same requests as the requests that are shown in Figure 2 on page 3, depending on local policies and the scope of project.

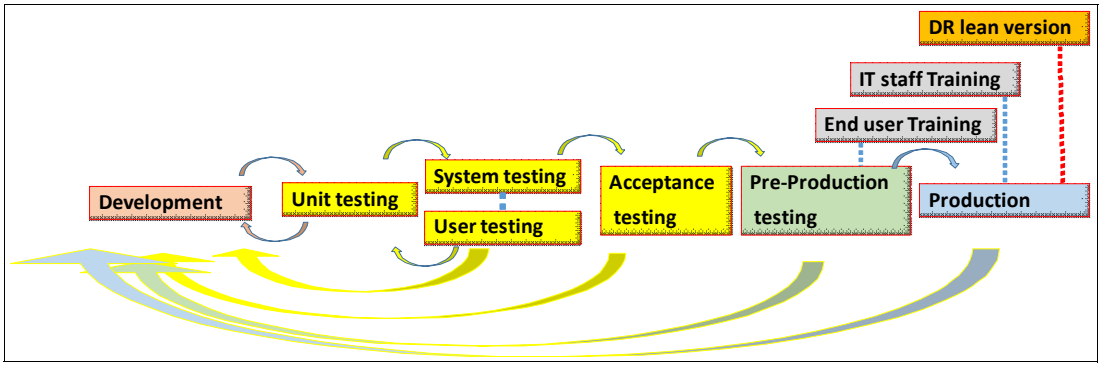

Figure 3 Application development cycle environment sample

Consider again the questions that are listed in Table 1 on page 4 for each of the environments that are shown in Figure 3. When the lifecycle expands, so too does the amount of effort and skills that are needed to produce each environment. As the project progresses through the lifecycle, more people become involved, each with a varying degree of understanding and responsibility.

The people that are involved in building the environments work for different teams within the IT department; therefore, separate work requests are generated, approved, completed, recorded, signed off, and accounted for (such as time sheets and charging).

An effective cloud service can help reduce the level of activities and complexities through the characteristics and enablement factors.

Having examined the characteristics, understanding the cloud service model is the next stage.

Cloud services models

NIST formally describes a standard for grouping cloud services, referring to them as *service models*. These service models are sometimes referred to as *delivery models* because they describe the services that are delivered by the cloud model. The following sections describe the NIST service models.

Infrastructure as a Service

The Infrastructure as a Service (IaaS) model is the simplest for cloud service providers to provision and can include the following elements:

- Compute
- Storage
- Network

Each of these elements is provisioned in an elastic fashion. As an IaaS user, you can deploy and run your chosen software, including operating systems and applications. You do not need to manage or control the underlying cloud infrastructure, but you can control the operating systems, storage, and deployed applications. You might also have limited control over select networking components, such as host firewalls.

Platform as a Service

The Platform as a Service (PaaS) model includes services that are built on IaaS services. They add value to the IaaS services by providing a platform on which the cloud users can provision their own applications, or conduct application development activities. The user does not need to manage the underlying cloud infrastructure (network, storage, operating systems, and so on), but can control configuring the provisioned platform services. The following services are provisioned in PaaS models:

- Middleware
- Application servers
- Database servers
- Portal servers

Software as a Service

The Software as a Service (SaaS) model provides software services that are complete applications that are ready to use. The cloud user connects to the application, which is running at a remote location. The user might not know where the system is located.

The cloud service provider is responsible for managing the cloud infrastructure, the system on which the application is running, and the application. This approach eliminates the need for the users to install and run the application on their own computers, which significantly reduces the need for maintenance and support.

SaaS is sometimes referred to as *applications as a service* because SaaS provides applications as a service rather than only software. SaaS also includes content services (for example, video on demand) and higher value network services (for example, VoIP) than typically encountered in communication service provider scenarios.

A summary of the service models is shown in Figure 4.

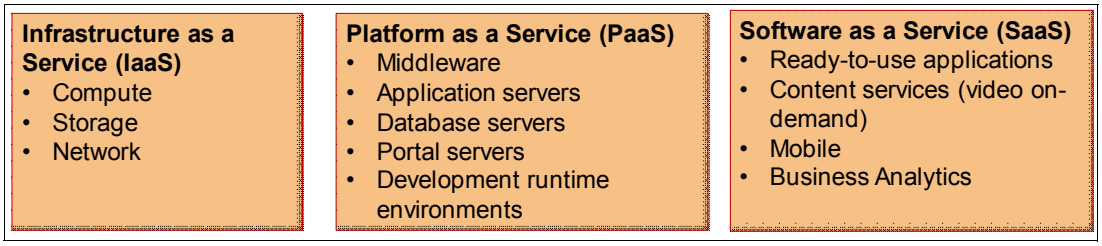

Figure 4 Service models

Cloud delivery models

Cloud delivery models refer to how a cloud solution is used by an organization, where the data is stored, and who operates the cloud solution. Cloud computing supports multiple delivery models that can deliver the needed capabilities in a cloud solution.

The following cloud delivery models are available:

- Public cloud
- Private cloud
- Hybrid cloud
- Community cloud

These delivery models provide services that are in line with the service models. You can integrate them with IT systems and other clouds.

Public cloud

A public cloud is a solution in which the cloud infrastructure is available to the general public or a large industry group over the Internet. The infrastructure is not owned by the user, but by an organization that provides cloud services. Services can be provided at no cost, as a subscription, or as a pay-as-you-go model.

Private cloud

A private cloud is a solution in which the infrastructure is provisioned for the exclusive use of a single organization. The organization often acts as a cloud service provider to internal business units that obtain all of the benefits of a cloud without having to provision their own infrastructure. By consolidating and centralizing services into a cloud, the organization benefits from centralized service management and economies of scale.

A private cloud provides an organization with some advantages over a public cloud. The organization gains greater control over the resources that make up the cloud. In addition, private clouds are ideal when the type of work that is being done is not practical for a public cloud because of network latency, security, or regulatory concerns.

A private cloud can be owned, managed, and operated by the organization, a third party, or a combination of the two. The private cloud infrastructure is provisioned on the organization's premises, but it can also be hosted in a data center that is owned by a third party.

Hybrid cloud

As the name implies, a hybrid cloud is a combination of various cloud types (public, private, and community). Each cloud in the hybrid mix remains a unique entity, but is bound to the mix by technology that enables data and application portability.

The hybrid approach allows a business to use the scalability and cost-effectiveness of a public cloud without making available applications and data beyond the corporate intranet. A well-constructed hybrid cloud can service secure, mission-critical processes, such as receiving customer payments (a private cloud service) and secondary processes, such as employee payroll processing (a public cloud service).

Community cloud

A community cloud shares the cloud infrastructure across several organizations in support of a specific community that includes common concerns (for example, mission, security requirements, policy, and compliance considerations). The primary goal of a community cloud is to have participating organizations realize the benefits of a public cloud, such as shared infrastructure costs and a pay-as-you-go billing structure, with the added level of privacy, security, and policy compliance that is associated with a private cloud.

The community cloud infrastructure can be provided on-premises or at a third party's data center. It can be managed by the participating organizations or a third party. A summary of the delivery models is shown in Figure 5.

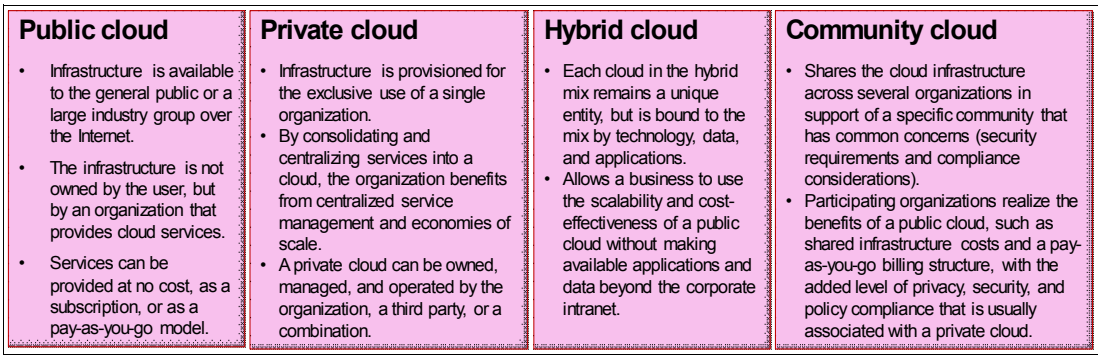

Figure 5 Delivery models

Two types of cloud

Thus far in this Redpaper publication, we focused on environments. Environments hold data that must be stored. Consider that there are two types of cloud services. The first is similar to provisioning environments; the second type is primarily concerned with data storage and how that data can be managed. Figure 6 shows the two types of cloud services.

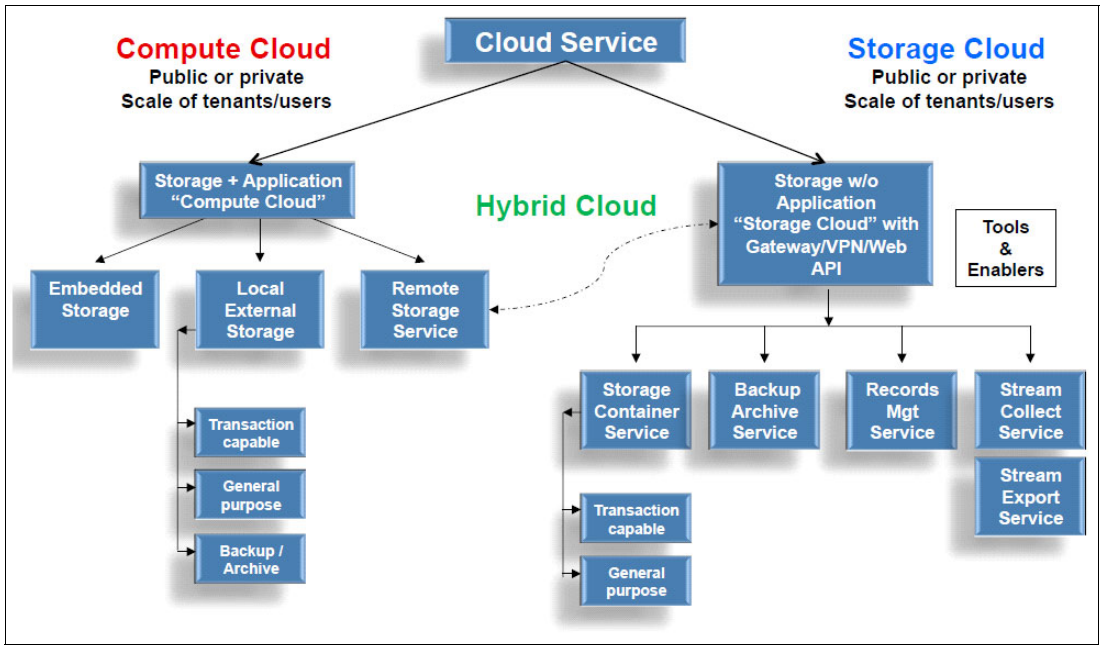

Figure 6 Types of cloud services

The compute cloud provides all of the necessary components to run the applications. The storage cloud holds the data and caters for functions, such as backing up and archiving data.

Cloud provisioning and management service positioning

The provisioning and management areas are within the compute cloud services. The approach for cloud on z/OS is not focusing on the provisioning of operating system instances, but rather the ability to provision multiple workloads in a single z/OS instance.

The targeted area for the cloud enablement is in provisioning middleware and related areas. The focus is within the PaaS and the SaaS, as shown in Figure 7 on page 10.

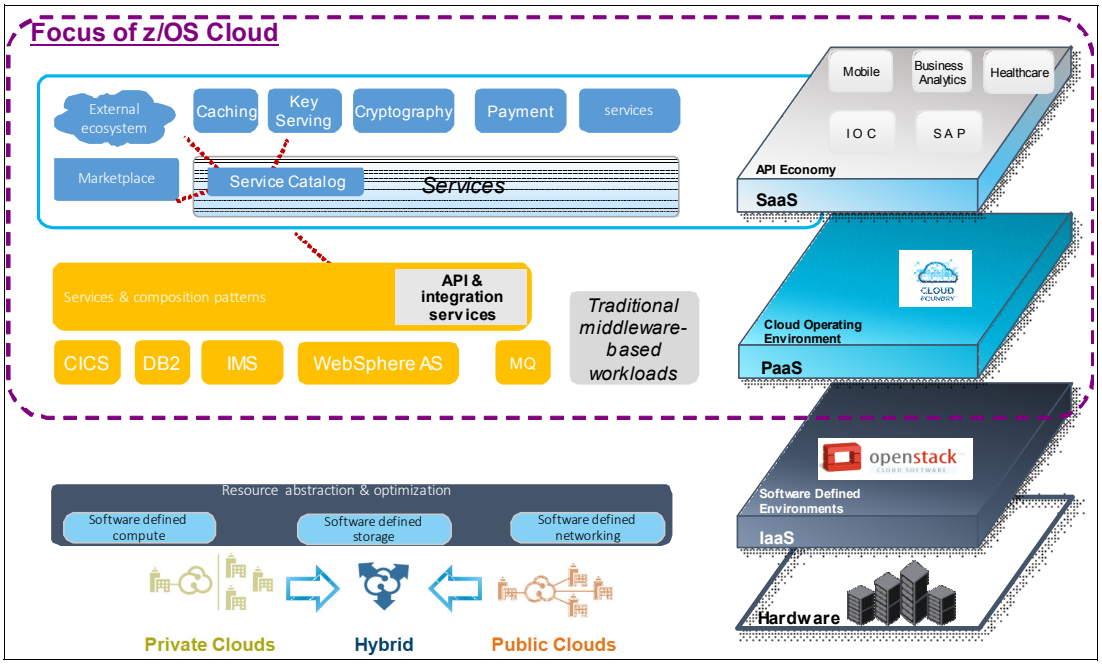

Figure 7 Focus of z/OS cloud enablement

This first step to providing a cloud service is to provision middleware to help quickly and effectively create environments that are in line with the NIST-defined cloud services characteristics and models.

Middleware plays a crucial role in the business application's lifecycle. New business applications (such as those based on mobile devices and IOT) are likely to access middleware components because of the need to access systems of record. They might need their own environments within the development lifecycle; therefore, the appropriate middleware must be provisioned quickly and effectively. Business applications also are changing to use the new technologies and the feeds from them, which raises the demand for more environments.

Middleware is key and must be provisioned in a manner that keeps pace with other provisioning methods and expectations.

z/OS cloud enablement cloud service

The NIST characteristics define the cloud services, but also expect the enablers, service models, and deployment models to be present. Although z/OS is well-positioned for many of the characteristics, what kind of cloud services might be provided?

If you think of yourself as a service provider, you might produce the following deliverables:

- A new user ID
- A new LPAR or an entire sysplex
- Middleware as a service, such as a CICS region or WebSphere Application Server
- An IBM DB2® table or an IBM MQ Queue as a service
- An entire application

You probably already produce these deliverables, but do you deliver them as a cloud service? To determine the answer to this question, it is necessary to revisit the NIST characteristics.

Support self-service provisioning

Traditionally, this area is not where z/OS was accommodating to all users. In the provisioning aspects, you might expect to see a team of infrastructure technicians that can include the following members:

- System programmers
- Storage managers
- Security administrators
- Network administrators
- Operations analysts
- Performance analysts
- Capacity planners
- Database administrators

These technicians build the environments by providing the following skills and standards:

- Operating system support
- Middleware software installation, customization, and roll-out processes
- IP addresses and connectivity configurations
- User IDs and access rules to IT resources and business-related data
- Encrypted data to protect the business
- Workload definition and prioritization within WLM
- SMF data capture, monitoring, analyzing, and reporting
- Automated procedures to help establish and maintain a smooth operational flow
- Application support from a technical infrastructure perspective

These skills are provided perhaps by passing through the following management process points:

- Individual work requests
- Granular change control for each change
- Problem management
- Implementation management
- Application project planning and reviews
- Capacity planning and performance management
- User liaison for training environments
- Service line support for queries and issues

From the service users perspective, you might see the following people who are involved throughout the provisioning process:

- Application development and maintenance:
  - Project managers
  - Project leaders
  - Solution architects
  - Testers
  - Systems analysts
- Business support:
  - IT account managers for LOB
  - Power users for the business
  - Strategy architects
  - Trainers
  - Mobile and other digital development
  - Social media
  - Online business analysts

- Analytics:
  - Business performance
  - Marketing
  - Product analysts
  - Customer behavior specialists
  - Search engine optimizers
  - Online content specialists

This list is not exhaustive because each sector, vertical, and organization has its own specific needs.

The expansion of on-demand services can mean that the users want more from the provisioners. The issue is how the provisioners provide the users with what they want with the following key deliverables:

- Consistent quality
- Rapid provisioning
- Middleware functionality
- Flexibility
- Assured availability levels

The ideal answer is a solution in which a mechanism to provide services (no matter how small or large) is put in place that provides an automated method that is based on standards. The solution must be easy to use and delivers middleware components to an agreed level of specification within a band of flexible options.

The method can encompass various entities from multiple provisioning areas (such as IP addresses and security access) and be classified as a single change.

The process is based on templates that include values that are agreed on by all parties to eliminate the need for repetitive information gathering, reviews, and meetings. The templates are processed and produce the wanted environment with all its components ready for use.

The organization can authorize the personnel to start a process that is deemed appropriate to their needs and roles.

The solution is a series of workflows to create and (where possible) maintain the environments for business development, delivery, and support, as shown in Figure 8.

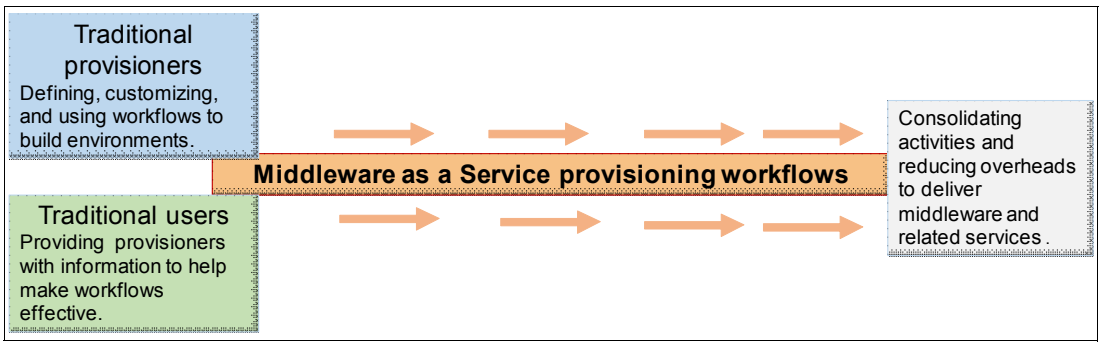

Figure 8 Workflow solution

z/OSMF is the key tool within the solution because it can provide the workflow process to deliver the middleware components. z/OSMF plays a pivotal role in delivering the workflows to build instances of middleware in a consistent, effective, and performant manner.

The z/OSMF GUI must be easily navigated to deliver the workflows. This feature is crucial in establishing the opportunity within z/OS for self-service. The simpler the interface that is used to start a workflow, the more attractive it is for users to feel confident in using it.

A user can be anyone who is given the authority to access the particular required service.

Accessibility through the Internet or corporate intranet

The use of z/OSMF is ideal because it is browser-based and can be accessed by using TCP/IP.

Each organization has its own policies about network access. The access to z/OSMF must be compatible with the standards and practices.

Providing a resource pool

z/OS and the hardware are capable of high virtualization levels and can refine sharing levels across the whole configuration. This feature is a fundamental difference from typical cloud environments; therefore, it has an advantage over more singular environments by creating virtual resources as and when it needs to without the user knowing how it is physically backed.

Resources can be allocated from the resource pool for a project. Resources are returned to the pool after they are not required.

The critical resources are mediated by WLM and other parts of z/OS. The user does not need to know about these mediations or how they are performed.

APIs enhance the users' ability to communicate when requesting information or services.

Providing simple and fast resource elasticity

z/OS can scale from less than 1 GB and fractional MIPS to 4 TB and 112,000 MIPS, then multiplied by up to 32 coupled systems. It can also scale out further to multiple sysplexes.

Monitoring and metered usage

z/OS has a long history and high proficiency in its ability to monitor resources with a dashboard view on cloud health status. SMF also can provide a metering capability, which enables a dynamic charge back model.

SMF and other repositories provide the opportunity to identify resource utilization and to cap usage or resources that is in line with the user's agreement.

Adopting the cloud approach for z/OS

Thus far in this Redpaper publication, we reviewed provisioning services. At the time of this writing, the IBM Cloud Provisioning and Management for z/OS can provide the following middleware:

- IBM WebSphere Application Server for z/OS Liberty Profile V8.5.8 or higher
- IBM MQ V8 or higher
- IBM CICS TS 5.x or higher
- IBM IMS™ V13 or higher
- IBM DB2 V11 or higher

In cloud environments on distributed servers (or even with Linux on IBM z™ Systems), customers provision a virtual machine with an instance of an operating system to run a single workload. To deploy another workload requires another virtual machine with another instance of the operating system.

However, in the context of z/OS, this methodology can be viewed from a different perspective by running under z/OS. With z/OS, multiple disparate workloads can be run with different service levels for those hosted workloads with isolation or multi-tenancy on the same virtual machine, as shown in Figure 9. (An LPAR that includes z/OS as its operating system can be considered a virtual machine.)

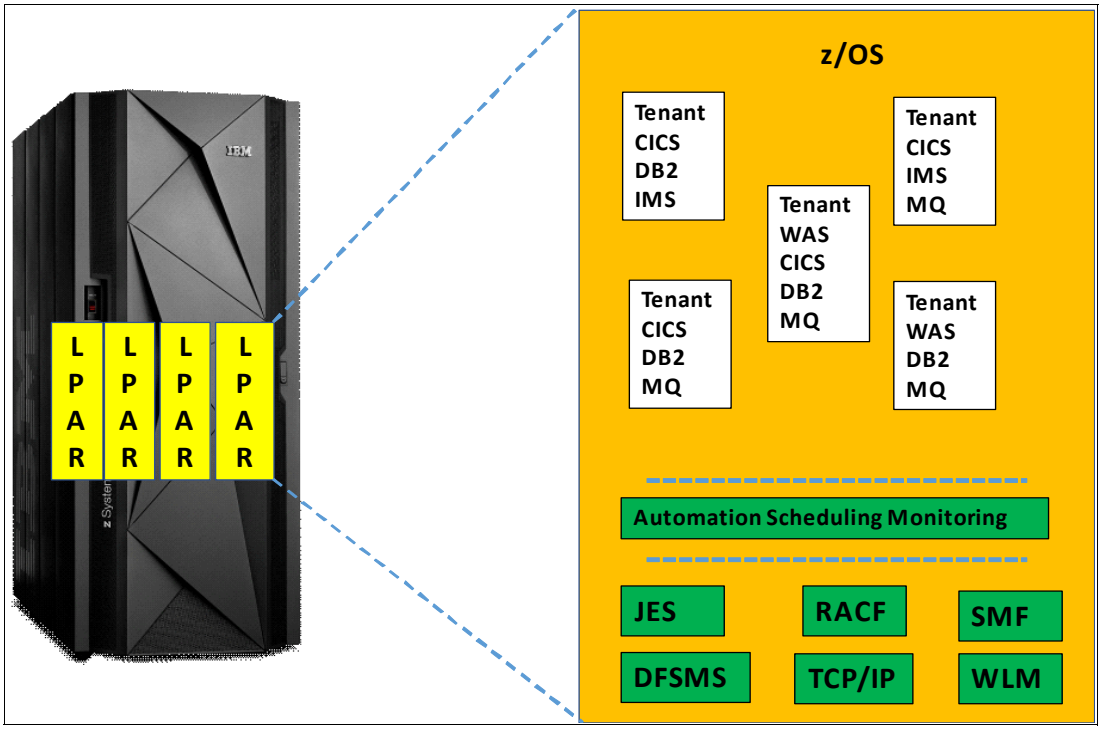

Figure 9 z/OS multi-tenant services

The design of z/OS (in particular the virtualization layers) provides you with the capability to deploy single instances of critical functions, such as security (IBM RACF®) and to control all of the security for multi-tenants from this single instance without compromising the isolation levels of any single tenant. Key address spaces, such as WLM, TCP/IP, JES2, and DFSMS SMF, can run in the same manner; that is, one instance that controls multi-tenants. WLM can prioritize at the tenant level and within the tenant level to provide scalable resource control at multiple levels.

Understanding the terms

It is important to understand the following terms to help clarify how provisioning and management works:

- Resources

  The following key resources are featured within the cloud provisioning tasks:

  - Domain

    Domains define the management scope for tenants, services, and resource pools. A domain consists of a z/OS system or set of z/OS systems in a sysplex. A z/OS system can be in a single domain or in multiple domains that are managed by a single z/OSMF.

- Resource pool

  These pools define the scope of shared z/OS resources within a cloud domain that includes multiple tenants.

- Tenant

  Tenants define the resource sharing scope; for example, a line of businesses or a class of users. A tenant consists of a user or group of users that were contracted to use specified services and pooled z/OS resources that are associated with the services in a domain.

- Roles

  The following key roles are featured in the cloud provisioning tasks:

  - Consumer

    A consumer is a user who is given access to the services and resource pools for a specific tenant. A consumer can provision a service instance and manage the lifecycle of a service instance.

  - Domain administrator

    This type of administrator is a user who manages a domain. The domain administrator is responsible for defining services, tenants, and resource deployment pools for the domain and managing the relationship across tenants, services, and resource pools.

  - Landlord

    A landlord defines the high-level cloud scope and the associated system resources for the cloud.

  - Resource pool administrator

    This user is responsible for managing a resource pool.

- Objects

  The following basic objects are featured in the Cloud Provisioning tasks:

  - Instance or software services instance

    This instances is the software that is provisioned by using templates.

  - Template or software services template

    This template consists of workflows and associated actions and variables that can be used to provision z/OS middleware.

Consumer scenario

Imagine that you are in the role of a consumer. You might be a project leader with a project to run and want some software instances provisioned. Where do you start?

After you are set up and can access z/OSMF from your browser, complete the following steps to start the provisioning process:

1. Browse to the marketplace, as shown in Figure 10 on page 16. As you can see, five instances are available.

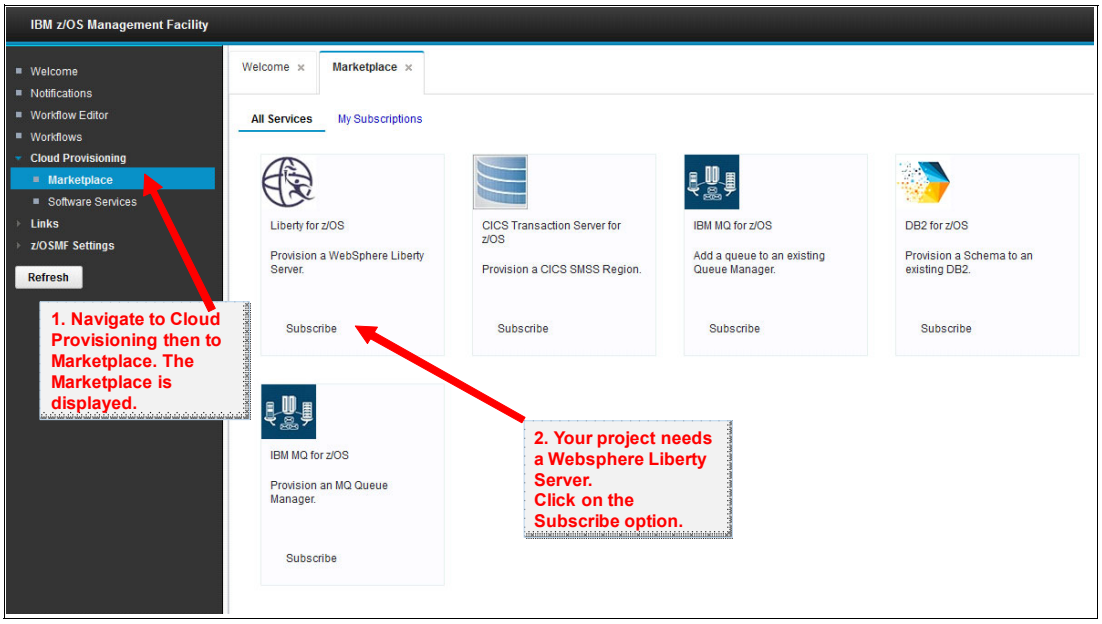

Figure 10 Marketplace instances

2. As an application project leader, you entered the marketplace as a consumer. Your project must add a WebSphere Liberty Server. Assume that a DB2 instance was provisioned. Click the **Subscribe** option to select the WebSphere Liberty Service, as shown in Figure 10.
3. The Subscribe to Liberty for z/OS window opens and in this scenario for the Tenant, you select the **Default** option from the drop-down menu and enter `Provision WLP` in the Subscription details field, as shown in Figure 11. Click **OK**.

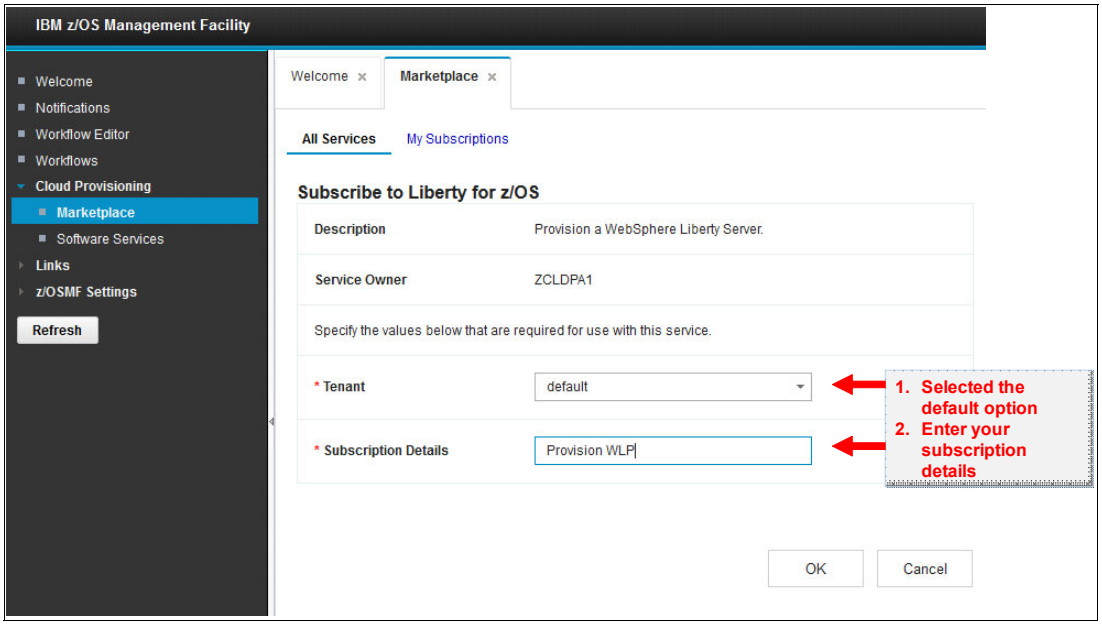

Figure 11 Subscribing to Liberty for z/OS

The window that shown in Figure 12 opens. You see a message in which the progress of your subscription to Liberty for z/OS is shown.

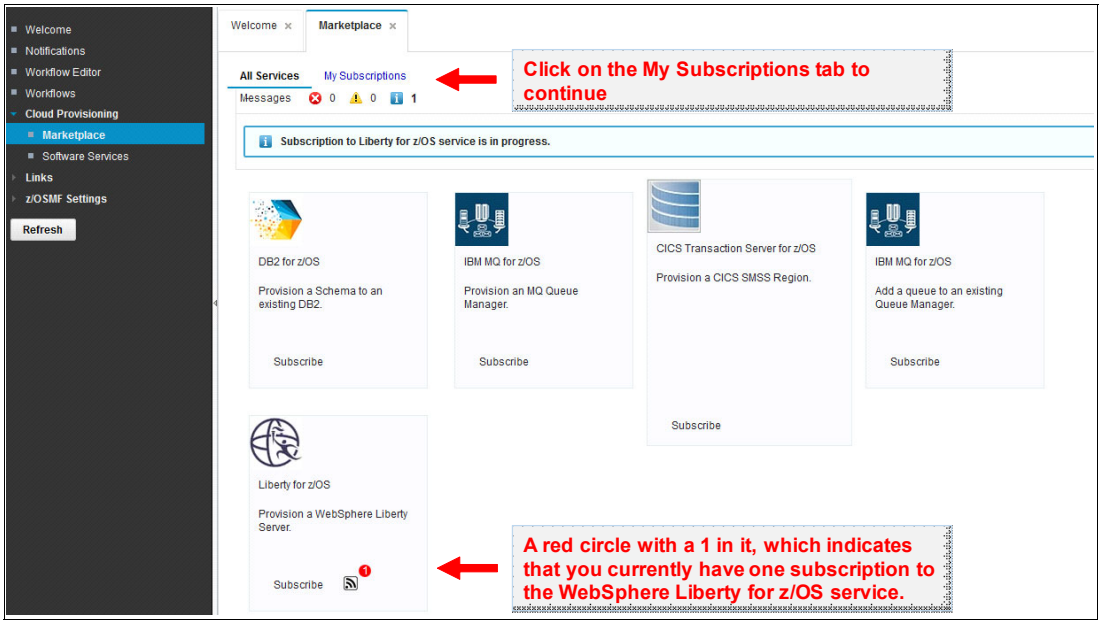

Figure 12 Subscription in progress

4. Click the **My Subscriptions** tab to continue. The provisioned status is shown. Click the refresh button until the State changes to “provisioned”, as shown in Figure 13.

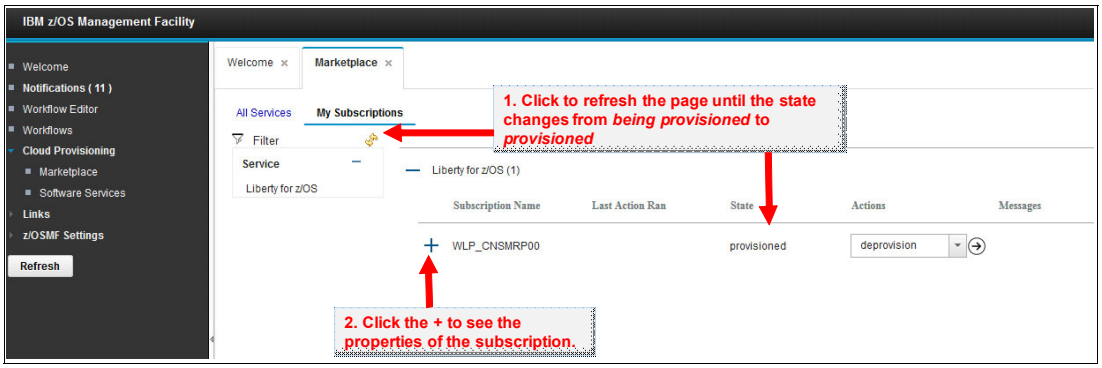

Figure 13 Provisioning completed

5. After your server is provisioned, you must know specific information about it. Click **+** to display the properties of your newly provisioned server, as shown in Figure 14.

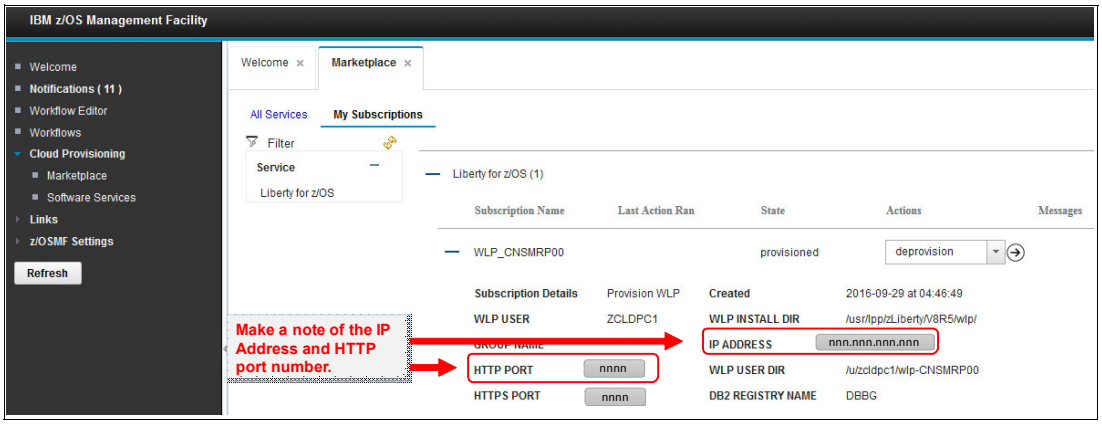

Figure 14 Your provisioned server properties

6. Record the IP address and the HTTP port number. You are now going to test your server to determine whether it is working. (The IP address and port numbers are obscured in our scenario.) Your own numbers are displayed in your instance.
7. You can verify that your WebSphere Liberty server was provisioned by entering the IP address and port number from the service properties in the address bar of your browser. Use the `http://ip_address:port` format.

   The welcome window that is shown in Figure 15 opens.

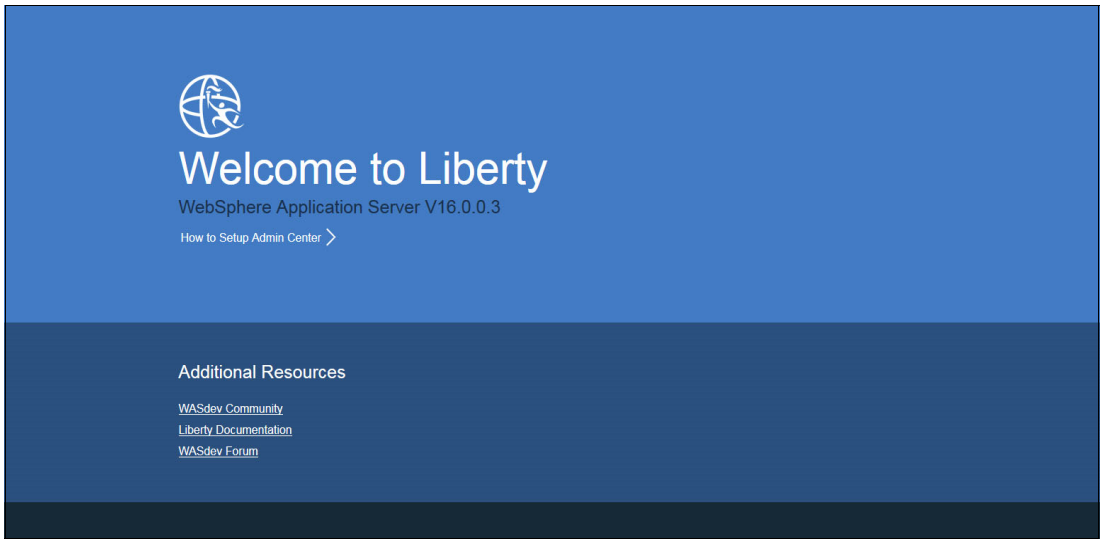

Figure 15 Welcome to Liberty window

In this scenario, a sample application is available and can be accessed by adding the name after the port, as shown in the following example:

```
http://ip_address:port/name_of_sample_app
```

The sample application that is used is `CloudTestJDBC`**.** The application tests the DB2 bind for the WebSphere. The results are shown in Figure 16 on page 19.

Servlet for validating JDBC with **IBM z/OSMF Cloud Provisioning**

This servlet dynamically creates a table, populates it, selects from it, then drops it

Connection to DB2 successfully achieved

Creation of table successful, will now populate ...

Table populated ... will now select and display results

```
Table Record = MyCity0, M0
Table Record = MyCity1, M1
Table Record = MyCity2, M2
Table Record = MyCity3, M3
Table Record = MyCity4, M4
Table Record = MyCity5, M5
Table Record = MyCity6, M6
Table Record = MyCity7, M7
Table Record = MyCity8, M8
Table Record = MyCity9, M9
```

End of records

Database table successfully dropped

Connection to DB2 closed

Figure 16 Sample application results

The server is now available for your project.

After the developers on your project complete their work, the Liberty server is no longer required and can be deprovisioned. The resources can be returned to the resource pool and made available for use.

Complete the following steps to deprovision the server:

1. Click the **My Subscriptions** tab.
2. Under the Actions column, select the **deprovision** option and click the arrow, as shown in Figure 17.

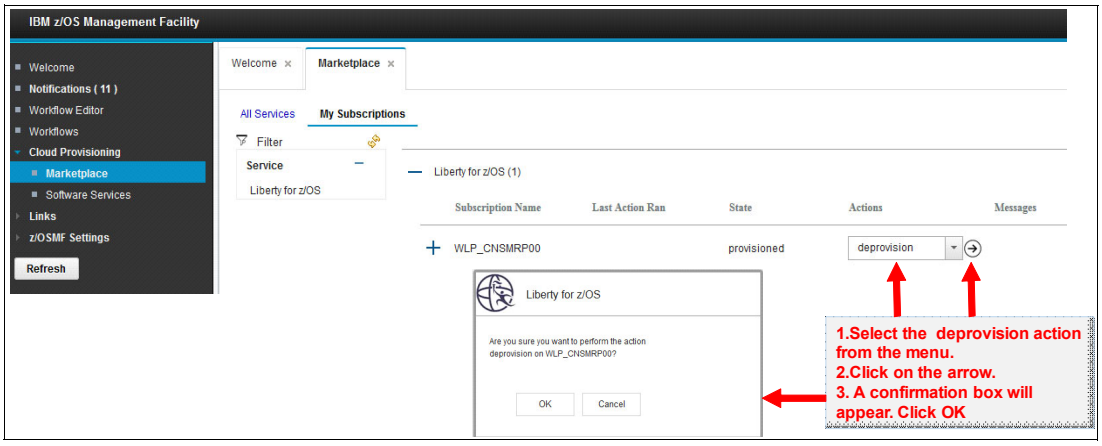

Figure 17 Selecting the deprovision action

3. A confirmation message opens. Click **OK** to proceed with the deprovisioning process. When completed, you see the window that is shown in Figure 18.

Figure 18 Deprovisioning the server

Your Liberty server is successfully deprovisioned. Your project has nothing else that is provisioned and is approaching its close-down phase.

4. Remove your subscription by selecting **Remove** from the Actions menu, as shown in Figure 19.

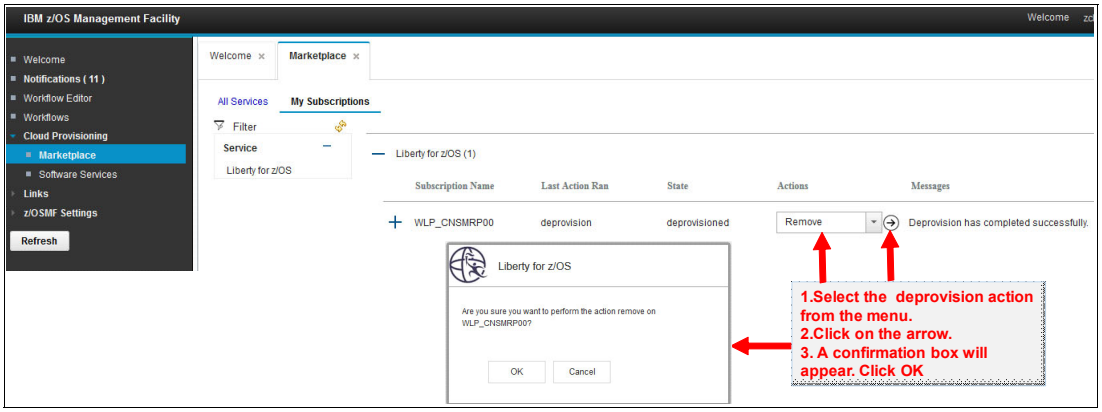

Figure 19 Removing your subscription

5. A confirmation message opens. Click **OK**. The window that is shown in Figure 20 opens, in which it is confirmed that no active subscriptions are available.

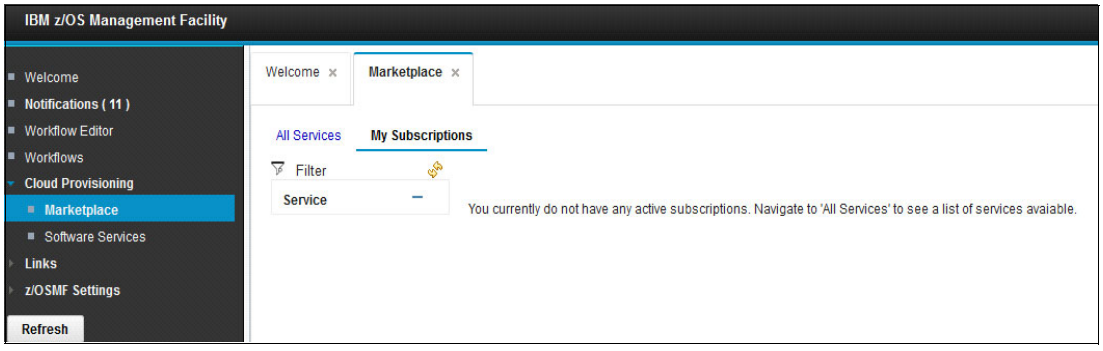

Figure 20 Confirmation that you have no active subscriptions

In this scenario, the consumer (project leader) provisioned a Liberty server for the project and then deprovisioned it. There are “behind-the-scenes” tasks, such as IP address allocation, security, and other infrastructure processes. This type of information is available within the processes, but the consumer was unaware of all these details. Instead, these details were the responsibility of other people, such as the resource pool administrators. The information was added by using templates, so instances can be provisioned seamlessly from the consumer’s perspective.

Architectural view

Several components work together to provide z/OS cloud enablement for the middleware initially. The architecture is shown in Figure 21.

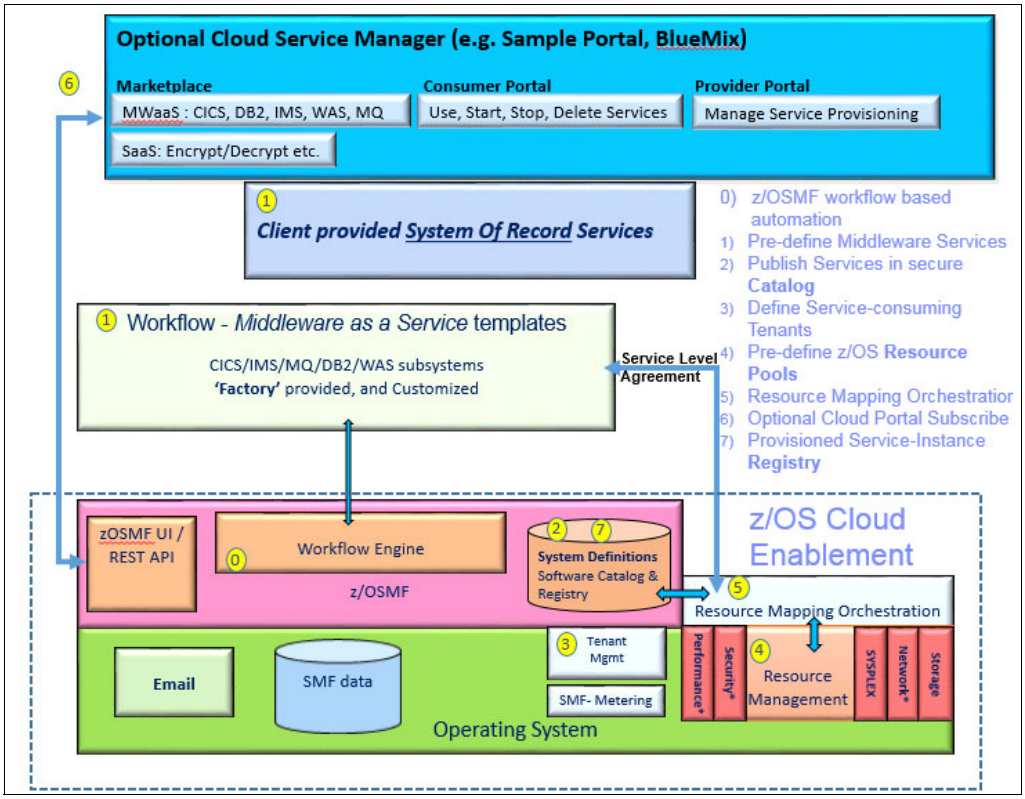

Figure 21 Architectural overview

The architecture consists of the following components:

- SMF data

  The provisioning occurs within the Systems of Record; that is, under z/OS, and collecting the SMF data allows you to integrate the following components with your business processes:

  - Audit requirements
  - Project performance (provisioning time)
  - Charge back for resource consumption
  - Analysis for future project requirements

- Tenant management

  This management defines the tenants who use the services, which allows degrees of sharing and isolation.

- Email

  Self-service provisioning is done by using email or is used as a notification option.

- Systems definition registry

  Provides a view in z/OSMF of the registry and definitions that are used for a specific system. It consists of the following components:

  - Software configuration catalog

    The repository of offered middleware services that includes tenant independent service template definitions. A single template can be available to multiple consumer groups.

  - Software instance registry

    Keeps record of provisioned middleware instances and their properties.

- Resource mapping services

  Maps from which tenants can access services from a specific resource pool. Workflows might require resources from the operating system, such as a user ID, port/VIPA, file system, SMF record (for auditing), or an entry in the workload management policy to set dispatching priority.

- Workflow engine

  Processes manual and automated actions (steps) with dependency tracking to complete a configuration activity.

- z/OS Management Facility

  Browser-based user interface in z/OS that provides administrator and user tasks and is based on WebSphere Liberty.

- Workflows

  Each subsystem provides a set of guided jobs, commands, or other activity that configures the subsystem.

- Cloud services and applications

  Business services (SaaS) that are built by using the subsystems and infrastructure. The marketplace shows the offerings to which you can subscribe.

A sample portal is available to support building a marketplace for services that are published in the z/OSMF catalog. It provides a consumer portal for a centric view of provisioned services and interacts with the z/OSMF software catalog and instance registry that uses REST APIs provided by z/OSMF. The sample portal can be customized to meet your requirements.

The administrator's role is more complex than the consumer's role and includes many tasks, such as maintenance and creating templates. Although the full complexity is beyond the scope of this Redpaper publication, it will be covered in future publications when IBM Cloud Provisioning and Management for z/OS is available.

Authors

This paper was produced by a team of specialists from around the world working at the International Technical Support Organization, Poughkeepsie Center.

Keith Winnard is the z/OS Project Leader at the International Technical Support Organization, Poughkeepsie Center. He writes extensively and is keen to engage with customers to understand what they want from IBM Redbooks® publications. Before joining the ITSO in 2014, Keith worked for clients and Business Partners in the UK and Europe in various technical and account management roles. He is experienced with blending and integrating new technologies into the traditional use of mainframes.

Gary Puchkoff leads the z/OS content selection and prioritization team in the IBM System z® group. His areas of expertise are in JES, SDSF, z/OSMF, and related work.

Hiren Shah is in the z/OS Development group in which he leads the z/OS Cloud Enablement development team. He is the architect and lead designer of IBM Cloud Provisioning and Management for z/OS. His areas of expertise include z/OSMF, WLM, System Logger, and related work.

Thanks to the following people for their contributions to this project:

Marna Walle (Senior Technical Staff Member, IBM) for reviewing the paper and suggestions about topic coverage.

Kevin Evans (Executive Project Manager, IBM) for overseeing access to laboratory systems for provisioning scenarios.

Vaughn C Page (IBM z Systems™ Software Engineer, IBM) for contributing reference materials.

Kara M. Todd (Manager, z/OS Cloud Enablement, IBM) for high-level coverage requirements.

Larry Coyne (Redbooks Project Leader, IBM) for reference materials about cloud definitions.

Jose Gilberto Biondo Jr (Infrastructure Architect, IBM) for contributing on the topic of infrastructure matters.

Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author —all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Stay connected to IBM Redbooks

- Find us on Facebook:

  http://www.facebook.com/IBMRedbooks

- Follow us on Twitter:

  http://twitter.com/ibmredbooks

- Look for us on LinkedIn:

  http://www.linkedin.com/groups?home=&gid=2130806

- Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

  https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm

- Stay current on recent Redbooks publications with RSS Feeds:

  http://www.redbooks.ibm.com/rss.html

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, MD-NC119, Armonk, NY 10504-1785, US

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Trademarks

REDP-5416-00

ISBN 0738455644

Printed in U.S.A.