

IBM PureApplication System

Database-as-a-Service patterns



© 2012 IBM Corporation

IBM Database Patterns employs pattern types that create and deploy databases in a Database-as-a-Service (DBaaS) cloud environment. This presentation covers the Database-as-a-Service (DBaaS) patterns support found in the IBM PureApplication™ System product.

Table of contents

- Overview
- Database pattern enablement
- Database pattern creation
- Summary

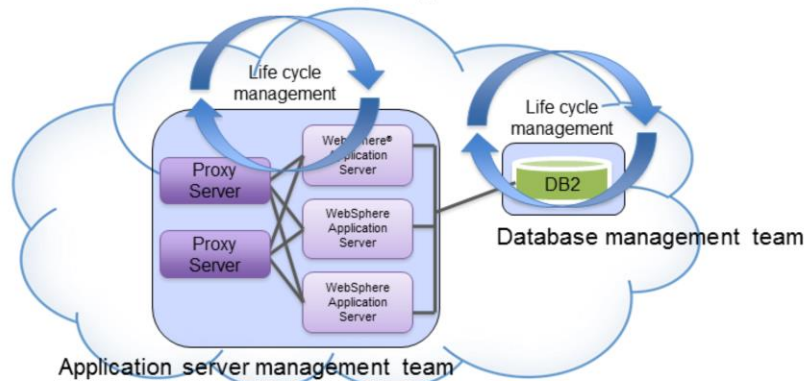
Looking at the agenda, you will see a brief overview of the Database-as-a-Service support in PureApplication Systems, including some terminology and concepts. You will then see some information on enablement and creation of database patterns.

Overview

In this section you will see an overview of the Database-as-a-Service support available in PureApplication Systems.

Overview

- IBM Database Patterns allow you to create and deploy DB2® databases in a Database-as-a-Service (DBaaS) cloud environment
 - You select the database requirements that meet your needs and PureApplication System builds the underlying topology to meet those requirements
- Separate need for databases in many usage scenarios:
 - Separate management team
 - Life cycle independent of application life cycle
 - May need database access across multiple applications



4

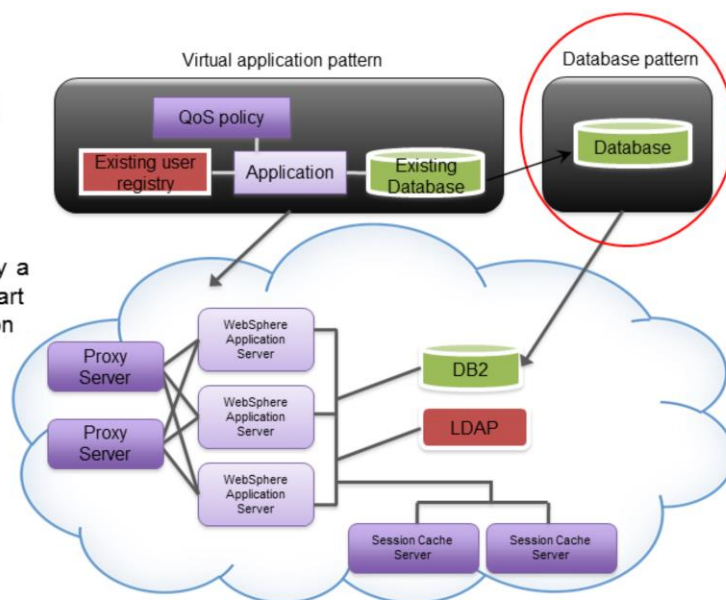
Database-as-a-Service patterns

© 2012 IBM Corporation

In many usage scenarios the database is a distinct entity with its own administrative team and its own life cycle. PureApplication System models this behavior with database patterns and database instances. You give PureApplication System your database requirements in the form of a database pattern and it will quickly build and deploy a DB2 database instance for you.

Database patterns

- Created and managed independent of any virtual application pattern
- Remote database shown ('Existing Database' component)
 - Also possible to deploy a database pattern as part of the virtual application



5

Database-as-a-Service patterns

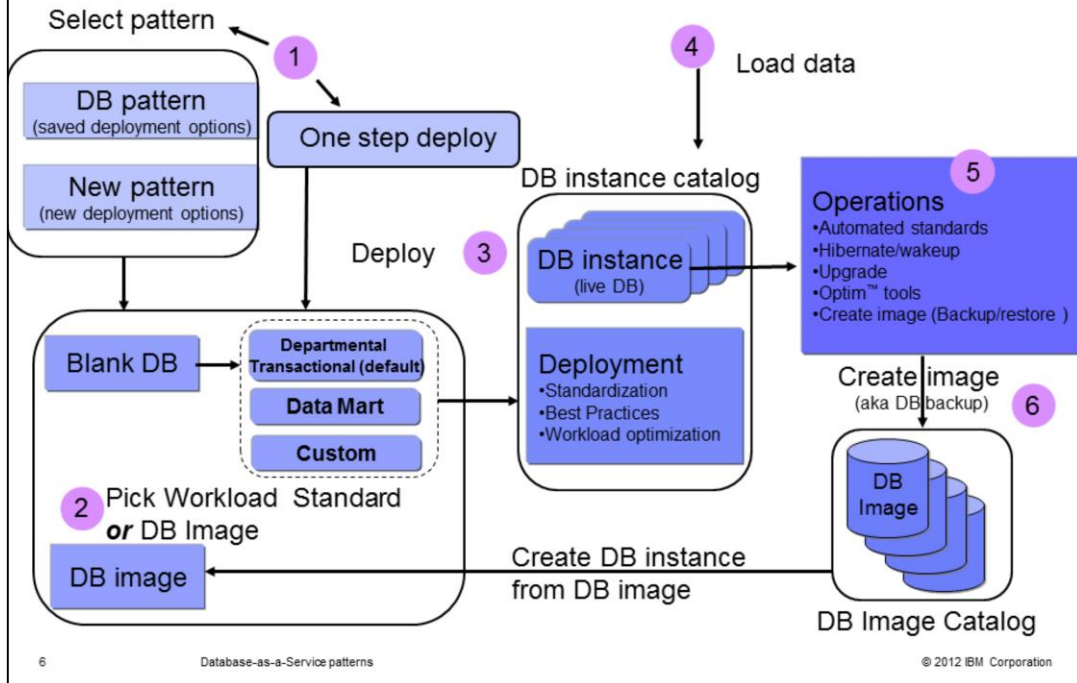
© 2012 IBM Corporation

In most “real world” usage scenarios databases are managed independent of any one application. To more closely model this paradigm, database patterns were introduced. You can create, delete, update, backup and restore databases created using a database pattern. These management activities are independent of your virtual application. Deleting your virtual application has no impact on the deployed database or database pattern.

This slide shows that you can include an existing database as part of your virtual application pattern. This is considered an ‘Existing Database’ component and is not actually deployed as part of your virtual application. The database already exists and is used by the application deployed as part of the virtual application. It can be created as a database instance in PureApplication System, as shown in the red circle labeled as ‘Database Pattern’. It can also be an already existing remote database that was created outside of the PureApplication System environment. In both of these cases, where an “existing database component” is being used, deletion of the database does not affect the state of the virtual application instance, and vice versa.

Note that it is also possible to deploy a database pattern as part of the virtual application. Instead of including an ‘existing database’ component in the virtual application pattern, you can include a database component that becomes a pattern-deployed database service. In the case of a pattern-deployed database service, the database is deployed as part of the virtual application so when the virtual application is deleted, the database is deleted as well.

Deployment and operations flow



This slide shows the overall flow for deploying a database in PureApplication System. To start out you have two options. You can start a deploy from an existing pattern or you can do a one step deploy, bypassing the pattern selection. If using a pattern, you can use a pattern that was saved **or** create a new one here. When creating a pattern you have two deployment options as seen in number two on the slide. You are given these same two options for the one step deploy as well. You can select the type of workload standard to apply to the DB instance to be created **or** you can select to clone an existing DB image that has been backed up to your DB image catalog repository. Tivoli® Storage Manager is used by PureApplication System to store these DB images. If selecting a workload standard from which to create your pattern, you will have three options: Departmental Transactional, Data Mart and Custom. Departmental Transactional is the default. The custom option allows you to add an unlimited number of custom workload standards that define specific tuning requirements that are required for your particular applications or workloads. Once you have your pattern set up as you want, you will deploy the database pattern which causes a DB instance to be deployed as seen in number three on the slide. The deployment process will make many decisions for you to tailor your DB instance. The deployed DB instance will include standardization, best practices and workload optimization that correspond to the workload standard selected. Once deployed, you have a fully functional robust database for your IT environment. Number five on the slide shows some operations that are available to you from the PureApplication System Database Service Console. This includes the backup of your image to the DB Image Catalog as seen in number six. Note that this operation requires Tivoli Storage Manager.

Database pattern enablement

This section covers enablement of the database patterns.

Database pattern enablement

- Licenses for virtual images and pattern types **MUST** be enabled before you can start creating database patterns
 - Accept licenses for base OS virtual images
 - Accept licenses for pattern types and enable
- Configure Tivoli Storage Manager plug-in (for database backups) - optional

Before using database patterns, there is some administrative work that must be done. The virtual image used for database patterns is the base OS virtual image, the 'IBM OS Image for Red Hat Linux® Systems'. You must accept the license for that in order to deploy database patterns. There are also some pattern types that you must accept licenses for and enable. Finally, there are some plug-ins that you might want to configure. If you plan to create database backups, you need to configure the Tivoli Storage Manager plug-in to point to an existing Tivoli Storage Manager installation. You will see these steps on the next few slides.

Enable virtual image licenses for base OS

▪ Catalog > Virtual Images

Needs enabling

Accept license

| Virtual Images | |
|---|--|
| IBM | |
| IBM OS Image for Red Hat Linux Systems | |
| IBM WebSphere Application Server Hypervisor Edition 8.0.0.3 with IMP (no fixes) | |

| IBM OS Image for Red Hat Linux Systems | |
|--|--|
| Description: | IBM OS Image for Red Hat Linux Systems |
| Created on: | Aug 8, 2012 9:58:56 AM |
| Current status: | License not accepted |
| Updated on: | Aug 8, 2012 10:24:40 AM |
| License agreement: | Not accepted [accept...] |
| Hypervisor type: | PureSystems_ESX |
| Operating system: | RedHat Enterprise Linux 64-Bit, version 6.2 (RHEL 6.2 X64) |

To enable the licenses for the base OS virtual image, you need to go to the **Catalog > Virtual Images** menu option in the Workload Console. This slide is showing that the base OS virtual image needs to have its licenses accepted. To do that you need to click the 'Accept license' hyperlink as shown on the right and accept all licenses you are presented with.

Enable pattern types

Pattern Types

- data
- Database Performance Monitoring Pattern 1.0
- Database Performance Monitoring Pattern 1.0.0.0
- IBM Data Mart Pattern 1.1**
- IBM Data Mart Pattern 1.1.0.3
- IBM Database Patterns 1.1**
- IBM Database Patterns 1.1.0.3
- IBM Transactional Database Pattern 1.1**
- IBM Transactional Database Pattern 1.1.0.3

IBM Transactional Database Pattern

Description: IBM Transactional Database Pattern

License Agreement: **Not Accepted** [License...]

Status: **Not Accepted** [Enable All...]

Required: IBM Database Patterns (1.1.0.3 or above)

System Plug-ins: Show me all plug-ins in this pattern type

Dependency

This pattern type has one or more prerequisites that need to be enabled first. In order to change the status of the current one, the listed pattern type(s) need to be enabled in sequence. Press "Enable All" to enable the pattern types. Please be patient, this may take awhile.

| Pattern Type name | Status |
|-----------------------------------|-------------|
| 1 Foundation Pattern Type 2.0.0.3 | Available |
| 2 IBM Database Patterns 1.1.0.3 | Unavailable |

Cloud > Pattern Types

License Agreement: **Accepted** [View...]

Status: **Available** [Disable...]

10 Database-as-a-Service patterns © 2012 IBM Corporation

This slide shows the pattern types that you need to enable. This is found in the **Cloud > Pattern Types** menu option in the Workload Console. The three pattern types needed for database patterns are highlighted in the red box on the left and include the Data Mart Pattern, the Database Pattern and the Transactional Database Pattern. On the right you see the steps outlined that need to be completed for each pattern type to be enabled. In step one, you need to click the "License" hyperlink where you are given the opportunity to read and accept this particular license. In step two, you need to enable the pattern. You do this by clicking the "Enable All" hyperlink. Notice the pattern dependencies listed under the dependency section. Those patterns also need to be enabled before using this pattern. In this case the Foundation Pattern and the Database Pattern are required. The Database Pattern has had its license accepted but has not been enabled. The Foundation Pattern is already enabled and is marked "Available" as shown. The "Enable All" hyperlink allows you to enable the Database Pattern when you are enabling the Transactional Database Pattern.

The screenshot displays the 'System Plug-ins' interface in the IBM Workload Console. On the left, a list of plug-ins is shown under the heading 'Configure system plug-ins, if required'. The 'logbackup 2.0.0.5' plug-in is highlighted in yellow, and a blue arrow points to its configuration window. This window includes fields for 'Backup IP:', 'SSH Host key file:', 'SSH key file:', 'User name:', 'Backup root directory:', 'Backup interval:' (set to 1800), 'Backup AIX rsync RPM:', and a checked option for 'Enable log backup on virtual systems:'. A yellow callout box above the configuration window reads 'logbackup: remote sync (rsync) of log files'. Below it, the 'tsm 1.1.0.3' plug-in is also highlighted in yellow, with a yellow callout box above its configuration window that reads 'tsm: For database backups in Tivoli® Storage Manager'. The tsm configuration window includes fields for 'TSM server address:', 'TSM server TCP/IP port:' (set to 1500), 'TSM server administrator user:' (set to admin), 'TSM server administrator password:', and 'Domain for DB2:' (set to DB2DOMAIN). A 'Cloud > System Plug-ins' label is positioned to the right of the logbackup configuration window. The IBM logo is in the top right corner, and the text '© 2012 IBM Corporation' is in the bottom right corner.

If you plan to use Tivoli Storage Manager’s backup functionality to make backups of your deployed databases, you will need to configure the ‘tsm plug-in’. You will find this in the Workload Console in the **Cloud > System Plug-ins** menu under “IBM Database Patterns”. When selected, you will choose the ‘Configure’ option as seen on the top of the slide. You will then provide the information needed for PureApplication Systems to talk to Tivoli Storage Manager such as the address, port and user credential information for the Tivoli Storage Manager. Also shown on the slide is the logbackup plug-in that allows you to save the log files to a remote system. These plug-ins are optional, however.

Database pattern creation

This section covers the creation of database patterns.

Creating new database patterns

- Database patterns
 - Solutions derived from standardized industry best practices
- Database pattern **contains only database**
- Can be created from
 - Workload standard
 - Clone
- From Workload Console
 - **Patterns > Database Patterns**



Database patterns are available to deploy databases within PureApplication System. When deployed, database patterns contain only a database. This allows for the separation of administration from the applications that use it. Two deployment options are available. The first one is pre-defined workload standards which is intended to capture best practices and focuses on particular types of databases. The second option allows you to take an existing database that has been backed up to Tivoli Storage Manager and 'clone it', thus copying it and all its configurations exactly. Within the workload standards option, you will see there are currently two IBM-defined workload standards shipped with the product. While these options allow for many different solutions, it does have limitations. You might have an application that doesn't quite fit into the IBM-defined workload standards. For these situations PureApplication System provides a custom option which allows you to define your own custom workload standards. You will see these workload standards defined on the next slide.

To create a database pattern, you go to **Patterns > Database Patterns** from the Workload Console.

Workload standard deployment

- **Workload standard:**
 - Specified as part of the database pattern
 - Set of predefined database configurations used to deploy a database
 - A set of scripts run to tune the operating system and instance configuration, create the database and accompanying objects and load the initial data
- **Three workload standards available:**
 - Departmental Transactional standard – predefined
 - Optimized for transactional applications
 - Data Mart standard – predefined
 - Used for data warehousing
 - Optimized for reporting applications
 - Custom standard – defined by you
 - Allows you to introduce required tuning and corporate standards

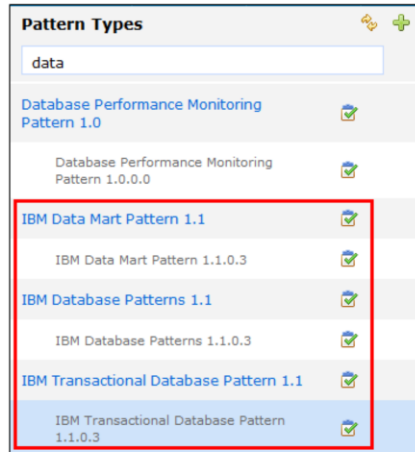
Workload standards are a set of predefined database configurations used as a deployment approach for creating database patterns. When a workload standard is selected, a set of scripts run to tune the operating system and instance configuration, create the database and accompanying objects and load the initial data. There are two workload standards shipped with PureApplication System. The Departmental Transactional standard is primarily used for online transaction processing and is optimized for transactional applications.

The Data Mart standard is primarily used for data warehousing and is optimized for reporting applications.

A third option, the custom standards, allow you to define your own workload standard when you need to customize the deployed database to meet some required tuning or corporate standards.

Associated pattern types

- IBM database workload patterns
 - Transactional Database Pattern
 - Data Mart Pattern



In order to support these workload standards, two database workload patterns are available. They are the 'Transactional Database Pattern' and the 'Data Mart Pattern'. These patterns need to have licenses accepted and be enabled before use.

The Transactional Database Pattern provides a set of capabilities that are essential to the provisioning and management of transactional database infrastructure for data centric applications

The Data Mart Pattern is similar to the Transactional Database Pattern, but deploys data marts including configuring the unique memory and disk requirements for this type of workload. The Data Mart Pattern includes data compression for improved performance in a shared disk environment typical to cloud data mart deployment.

Clone support

- Define a database pattern using an existing database image
 - Uses existing image as a model for creating the database pattern
 - Existing image has been previously created
 - Existing image can be manually created or automatically created by PureApplication System
 - Recommend using one that has been manually created
- Virtual machine is created with the same resource settings as the original
 - Uses same licenses and configuration

Cloning is a provisioning approach that uses an existing database image as a model for creating database patterns. When an image is selected, the metadata stored during backup is retrieved. A new virtual machine is created with the same resource settings. The DB2 Restore command creates a new database with the same license and configurations. This cloned database then sits on the newly created virtual machine.

You should use manually created images in preference to automatically created backups for this task. You can manually create a database image with the **Create a Database Image** function in the Database Service Console.

Tivoli Storage Manager must be configured to use this function.

Creating new database from default workload standards

Database Patterns

Database pattern name

Database Pattern

Specify options for your database pattern.

Database pattern name: Test DB pattern

Database pattern description:

Purpose: Production

Source: a default database workload standard

| Name | Description |
|---|---|
| <input checked="" type="radio"/> Departmental Transactional | For databases primarily used for online transaction processing (OLTP). The database will be optimized for transactional applications. |
| <input type="radio"/> Data Mart | For databases primarily used for data warehousing. The database will be optimized for reporting applications. |

Database version: DB2 Version 9.7 for Linux

Database level: DB2 Version 9.7 Fix Pack 5 for Lir

Database size (GB): 30

Database compatibility mode: DB2 (Default)

Schema file: artifacts/InfoDB.sql

Save Cancel

17 Database-as-a-Service patterns © 2012 IBM Corporation

To create a new database, you will click the green plus sign in the **Patterns > Database Patterns** menu. To create a database from the shipped workload standards, you will set the “Source” field to “**Apply a default database workload standard**”. This tells PureApplication System the characteristics of the database you will provision. When that ‘source’ is selected, you have the option of choosing from the two pre-loaded workload standards. The first choice, Departmental Transactional, is the transactional standard for online transaction processing. The second choice here, Data Mart, can be used for a database that is used primarily for data warehousing. When your source is “**Apply a default database workload standard,**” you can specify a schema file to format the database that is created; this is not a required parameter however. If specified, the schema file is used to format the database that is created. Another piece of information that is asked for when building your database patterns is “Purpose” which is set to ‘Production’ since you have an unlimited set of DB2 licenses in PureApplication System. This is normally used to determine which license should be applied to the DB2 instance to be provisioned. The database size will dictate the size of the virtual machine that is created. You also have the option to select the version of DB2 to use for your deployment and whether you want to use the Oracle compatibility mode.

Creating new database from custom workload standards

- When 'Source' is set to 'Apply a database workload standard':
 - List of available custom workload standards

18
Database-as-a-Service patterns
© 2012 IBM Corporation

To create a database from a 'custom' workload standard in "Database Patterns," you will set the "Source" field to "**Apply a database workload standard**". When selected, you are given a list of non-default 'Database Workload Standards' that have been created and added to the catalog. Another presentation goes into details about actually creating custom workload standards. Notice that you are not given the option of providing a schema file here. The definition of the database is taken care of in the workload standard definition.

Creating new database from clone

- When 'Clone' is chosen as source, a list of available images is displayed

Database Pattern

Specify options for your database pattern.

Database pattern name:

Database pattern description:

Purpose:

Source:

| Image Name | Database Name | OS |
|--|---------------|-------|
| <input checked="" type="radio"/> System-created image 'oltpdb' | oltpdb | Linux |
| <input type="radio"/> System-created image 'dmdb' | dmdb | Linux |
| <input type="radio"/> System-created image 'martdb3' | martdb3 | Linux |
| <input type="radio"/> myimage1 | martdb3 | Linux |

19 Database-as-a-Service patterns © 2012 IBM Corporation

When building your database pattern, the other choice for 'Source' is '**Clone from database image**'. If that is selected, you are presented with a list of images to choose from. This is shown by the red box on the slide. In this case, there are four images available to choose from.

Notice that no other fields are required when using cloning.

Create new database in virtual application builder

- Also available when deploying a database as part of a virtual application

The screenshot displays the configuration interface for a new database in the virtual application builder. On the left, a diagram shows an 'Enterprise Application' connected to a 'Database'. The main configuration panel on the right includes the following fields:

- Name:** Database
- Database name:** mydb
- Database description:** (empty)
- Purpose:** Production
- Source:**
 - Apply a default database workload standard:

| Name | Description |
|----------------------------|---|
| Departmental Transactional | For databases primarily used for online transaction processing (OLTP). The database will be optimized for transactional applications. |
| Data Mart | For databases primarily used for data warehousing. The database will be optimized for reporting applications. |
- Database version:** DB2 Version 9.7 for Linux
- Database level:** DB2 Version 9.7 Fix Pack 5 for Linux
- Database size (GB):** 30
- Database compatibility mode:** DB2 (Default)
- Schema file:** artifacts\enup_db.sql (with Edit and Delete buttons)

20

Database-as-a-Service patterns

© 2012 IBM Corporation

You can also deploy a database as part of a virtual application. In this case, the database is configured in the virtual application builder as seen on the slide. The same options are available as you saw on the previous slides. This is showing a database whose 'Source' is one of the default workload standards.

Database patterns - Operations

- From **Workload Console > Pattern > Database Patterns**, select the appropriate database pattern
- Can Deploy, Open (edit) or Delete database pattern
- Can give others access to the selected database pattern

The screenshot displays the details for a database pattern named "Test DB pattern". At the top right, there is an "Operations" menu with three options: "Deploy", "Open", and "Delete". Below this, the pattern's details are listed:

- Database Pattern ID:** a-0f2af245-2b51-455e-9730-e7a6f36254a4
- Created by:** deploy18
- Updated by:** deploy18
- Access granted to:** dbaasGroup [read] [remove]
Add more...
- Database Pattern Description:** Test DB
- Purpose:** production
- Source:** Apply a default database workload standard

A callout box labeled "Grant access to others" points to the "Access granted to" section.

21

Database-as-a-Service patterns

© 2012 IBM Corporation

The operations available for a database pattern are shown on the slide. When you select a particular database pattern that you have created, you have the option to deploy an instance of it into your cloud, open it for editing or delete the database pattern. You can also grant others access to the selected database pattern.

Database patterns – Open (edit)

The screenshot shows a toolbar for a database pattern named "Test DB pattern". The toolbar contains three buttons: "Deploy", "Open" (highlighted with a red box), and "Delete". An arrow points from the "Open" button to a larger panel titled "Database Pattern" with the subtitle "Edit options for your database pattern." A yellow callout bubble points to this panel with the text "Same panel as when creating pattern".

The "Edit options for your database pattern" panel includes the following fields and options:

- Database pattern name: Test DB pattern
- Database pattern description: Test DB
- Purpose: Production
- Source: Apply a default database workload standard
- Name**
 - Departmental Transactional
 - Data Mart
- Description**
 - For databases primarily used for online transaction processing (OLTP). The database will be optimized for transactional applications.
 - For databases primarily used for data warehousing. The database will be optimized for reporting applications.
- Database version: DB2 Version 9.7 for Linux
- Database level: DB2 Version 9.7 Fix Pack 5 for Lin
- Database size (GB): 30
- Database compatibility mode: DB2 (Default)
- Schema file: artifacts/setup_db.sql (with Browse and Delete buttons)
- Advanced options (checkbox)
- Save and Cancel buttons at the bottom.

22

Database-as-a-Service patterns

© 2012 IBM Corporation

This slide shows the edit option for database patterns. You see you are shown the same panel as when you created the pattern.

Database patterns - Deploy

Deployment of a database pattern creates a database instance

Test DB pattern Deploy Open Delete

Deploy Database from Database Pattern x

Name:

Filter by IP type: IPv4 IPv6

Filter by profile type:

Profile:

Priority:

Cloud group:

IP group:

Advanced

Database name (1-8 characters long)

The deploy panel for database patterns is identical to the panel you see when deploying a virtual application. The name, however, will become the name of your deployed database and must be between one to eight characters long. You select the environment profile for the deployment target from the drop-down menu if you have permission to more than one environment profile. The environment profile selected will then determine what cloud groups and IP groups can be selected. In this case, you have permissions to more than one environment profile but that environment profile is limited to one cloud group and IP group which is why they are both not available for selection. You can also add or create an SSH key here to allow you to login to the virtual machine once it has been deployed. An SSH key can be added later from the Database Service Console if needed.

Any database pattern that is part of a virtual application pattern is deployed as part of the virtual application deployment. You can see an example of that in the virtual application administration education module.

Summary

This section gives a summary of what was covered in this presentation.

Summary

- Real life database usage scenarios modeled in PureApplication System with Database Patterns
 - Database managed independent of virtual application
- Database also available as part of virtual applications
 - Life cycle is tied to the virtual application

To align with real world database usage scenarios, databases can be managed independent of any virtual application. This is modeled in PureApplication System with Database Patterns. This is only one available database management scenario. You can also create a database belonging to your virtual application. In this case the database's life cycle will follow your virtual application's life cycle. In this presentation you saw how to enable the database patterns and how to create them. To see information on the management of deployed database patterns, see the DBaaS Administration presentation.

Trademarks, disclaimer, and copyright information

IBM, the IBM logo, ibm.com, DB2, PureApplication, and Tivoli are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of other IBM trademarks is available on the web at "[Copyright and trademark information](http://www.ibm.com/legal/copytrade.shtml)" at <http://www.ibm.com/legal/copytrade.shtml>

Linux is a registered trademark of Linus Torvalds in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS OR SOFTWARE.

© Copyright International Business Machines Corporation 2012. All rights reserved.