

IBM PureApplication System

Database-as-a-Service custom workloads



This presentation covers custom workloads for Database-as-a-Service in IBM PureApplication™ System.

Table of contents

- Overview
- Creating customized workload standards
- Defining customized workload standard package
- Summary

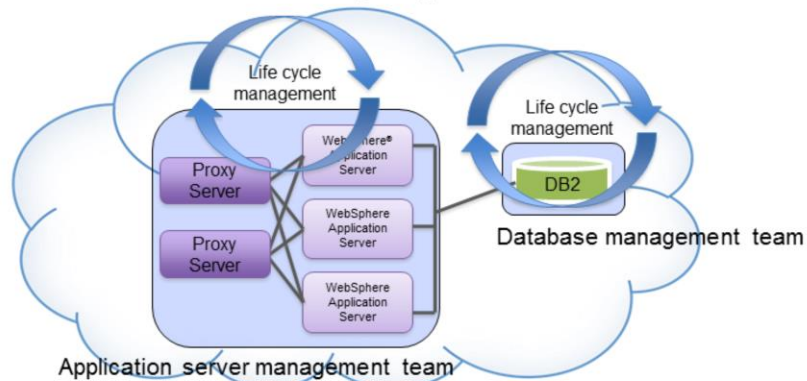
The agenda for this presentation covers a brief overview of Database-as-a-Service and workload standards in IBM PureApplication System, the creation of customized workload standards and the definition of the customized workload standard package.

Overview

This section will cover an overview of Database-as-a-Service and workload standards in IBM PureApplication System.

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

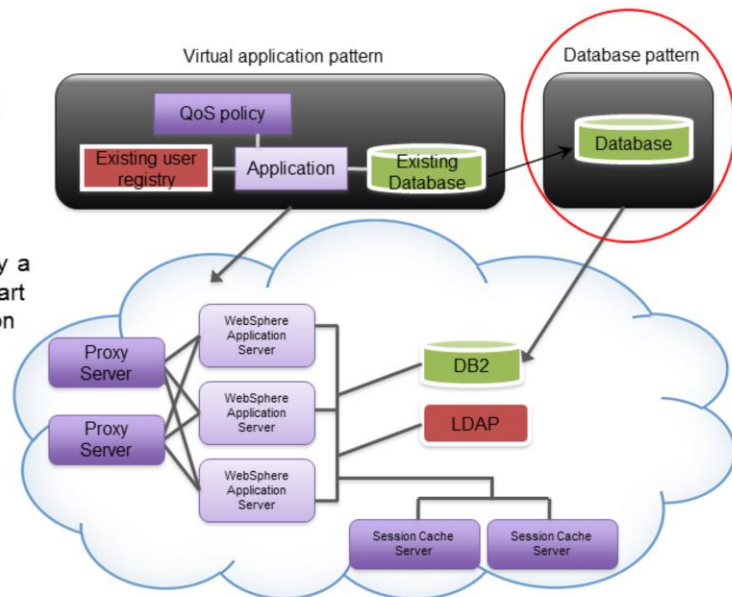
Database-as-a-Service custom workloads

© 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 custom workloads

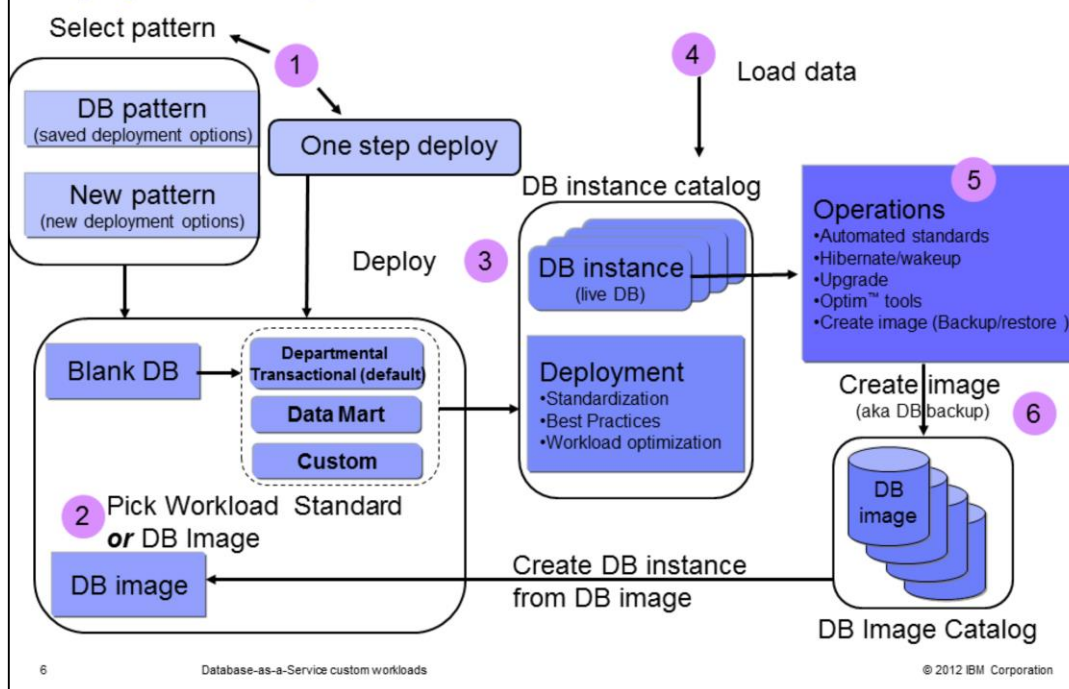
© 2012 IBM Corporation

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.

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

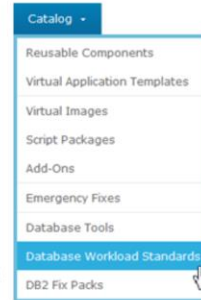
Workload standards are a set of predefined database configurations used as a deployment approach for creating database patterns. When a workload standard is selected, a set of scripts run to tune the operating system and instance configuration, create the database and accompanying objects and load the initial data. There are three types of workload standards available. The first two, Departmental Transactional and Data Mart are shipped with the product. The third type of workload standard, custom, is what the rest of this presentation will focus on. 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.

Creating customized workload standards

This section covers the creation of customized workload standards.

Creating customized workload standard

- From Workload Console
 - **Catalog > Database Workload Standards**
 - "Manage workload resources" permission required



- Two IBM-provided options are shown
 - Cannot be deleted
 - Patterns must be enabled



9

Database-as-a-Service custom workloads

© 2012 IBM Corporation

To create a customized workload standard, there is an option under 'Catalog' which allows you to add your own workload standards. Go to '**Catalog -> Database Workload Standards**'. '**Manage workload resources**' permission is required for this function. When you first go there you will see the two IBM-provided workload standards that ship with PureApplication System: Data Mart and Departmental Transactional. You are not able to delete these workload standards. In order to have them available here though, the pattern for each must be enabled first.

Create new customized workload standard in catalog

Database Workload Standards

Database pattern name ti

Database Workload Standards

Edit options for your customized database workload standard.

- Name:** Custom Workload Standard
- Description:** My custom standard
- Workload type:** Departmental Transactional
- Initial disk size (GB):** 10
- Storage multiplier (User data to physical disk ratio):** 3
- Upload file (.zip):** CustomWorkloadStd.zip

Callouts:

- Data Mart or Departmental Transactional** (points to Workload type)
- Space available when deployed (0-500GB)** (points to Initial disk size)
- Used to determine user data size (1.0-3.0)** (points to Storage multiplier)
- .zip file containing scripts** (points to Upload file)

10 Database-as-a-Service custom workloads © 2012 IBM Corporation

To create a customized workload standard, you need to provide a .zip file containing scripts that will run during the database deployment step. During the pattern definition, you specify your 'Custom Workload Standard' as the workload standard to use. You will see that on the next slide. The 'base' workload type must be specified for your custom workload standard and the choices are 'Data Mart' and 'Departmental Transactional'. In addition to the Workload type, you need to specify the initial disk size for this workload standard and a 'Storage multiplier' which is used to determine the user data size that is needed. The initial disk size can be between 0 and 500GB; the Storage multiplier can be between 1.0 and 3.0.

Create database pattern from custom workload standard

- Custom workload standards available for selection after adding to catalog

Database Pattern

Specify options for your database pattern.

Database pattern name: CustomDB
 Database pattern description: DB from custom workload standard
 Purpose: Production
 Source: Apply a database workload standard

Name	Description
<input type="radio"/> cust_wls_1	for project1
<input checked="" type="radio"/> Custom Workload Standard	My custom standard
<input type="radio"/> DeployerPP_WkldStnd	

Database version: DB2 Version 9.7 for Linux
 Database level: DB2 Version 9.7 Fix Pack 5 for Linux
 Database size (GB): 10

Save Cancel

Workload standard that was added

Specify DB2 v9 or v10

11

Database-as-a-Service custom workloads

© 2012 IBM Corporation

Any custom workload standards defined and added to the Catalog are available when creating your database pattern if 'Apply a database workload standard' is selected. Notice the name here matches what was defined on the previous slide.

Note that when you deploy a database from a customized workload standard, you cannot specify Database compatibility mode or upload a Schema File.

Non-custom workload fields differ

- Parameters that are not available for custom workloads

Database Pattern

Specify options for your database pattern.

Database pattern name: CustomDB

Database pattern description: DB from custom workload standard

Purpose: Production

Source: Apply 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: Browse

Advanced options

Save Cancel

12

Database-as-a-Service custom workloads

© 2012 IBM Corporation

This slide shows the Departmental Transactional workload standard selected. The 'Database compatibility mode', 'Schema file' and 'Advanced' options are not available when the 'Custom' workload standard was selected. These options need to be coded in the scripts for the custom workload standard.

Custom workload standard in virtual application builder

The screenshot shows a workflow diagram on the left with an 'application' component pointing to a 'database' component. On the right, the 'Database' configuration panel is visible. A yellow callout box points to the 'Apply a database workload standard' dropdown menu, which is expanded to show three options: 'cust_wis_1 for project1', 'Custom Workload Standard My custom standard', and 'DeployerPP_WildStd'. The 'Custom Workload Standard' option is selected and highlighted in yellow.

Note same pattern options are available in the virtual application builder

13




Database-as-a-Service custom workloads

© 2012 IBM Corporation

This slide is showing the use of a custom workload standard in the virtual application builder. Note that you have the same options available here as you did when building a database pattern from the 'Patterns' menu.

Database workload standards - Operations

- From **Catalog > Database Workload Standards**, select the appropriate workload standard
- Can Open (edit), Delete or Export workload standard
 - Export downloads associated .zip file

Custom Workload Standard	Operations	 Open	 Delete	 Export
Name:	Custom Workload Standard			
Description:	My custom standard			
Workload type:	Departmental OLTP			
Initial disk size (GB):	1			
Storage multiplier (User data to physical disk ratio):	3			
Upload file (.zip):	CustomWorkloadStd.zip			

The operations available for a Database Workload Standard are shown on the slide. When you select a particular workload standard that you have created, you have the option to open it for editing, delete the workload standard or export the workload standard to your workstation.

Database workload standards – Open (edit)

Database Workload Standards

Custom Workload Standard

Open Delete Export

Database Workload Standards

Edit options for your customized database workload standard.

Name: Custom Workload Standard

Description: My custom standard

Workload type: Departmental Transactional

Initial disk size (GB): 10

Storage multiplier (User data to physical disk ratio): 3

Upload file (.zip): CustomWorkloadStd.zip Browse

Save Cancel

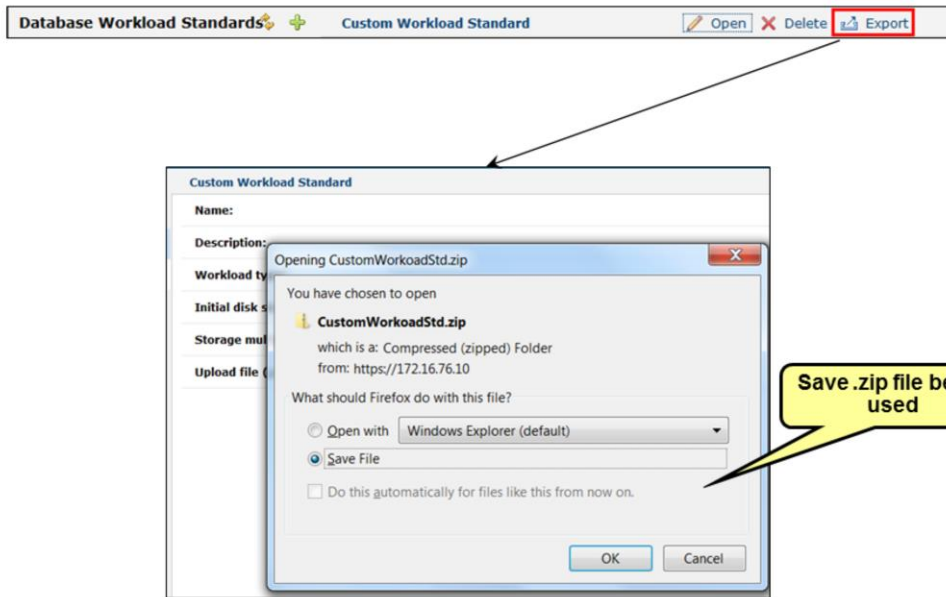
Same panel as when creating pattern

Upload updated .zip file

15 Database-as-a-Service custom workloads © 2012 IBM Corporation

This slide shows the edit option for database workload standards. You see you are shown the same panel as when you created the database workload standard. One of the main reasons for editing the workload standard is to update the .zip file used for deployment. This is possible here as shown on the slide.

Database workload standards – Export



16

Database-as-a-Service custom workloads

© 2012 IBM Corporation

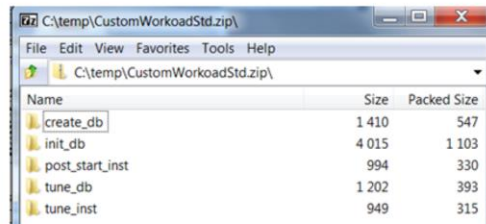
This slide shows the export option for database workload standards. When 'export' is specified, you are given the opportunity to open or save the .zip file being used to your workstation. You can then make any needed changes and update the workload standard. You can also use it as a base for a new workload standard that you need to create. Note that not all parameters are exported; the export only exports the .zip file.

Customized workload standard package definition

This section will talk about defining a custom workload standard package.

Script package for use with custom workload standards

- Packaged as a .zip file
 - Name should be one to eighteen alphanumeric characters in length
 - Cannot begin with a number or an underscore
- Consists of shell scripts that will perform the configuration changes required
 - Scripts run as the OS user **db2inst1**
- Contains following possible first-level subdirectories:
 - tune_inst
 - **create_db (required)**
 - tune_db
 - init_db
 - post_start_inst
- Each subdirectory has an entry script that can invoke additional scripts under it when called



The script package that is used to deploy a database pattern using a custom workload standard is packaged as a .zip file. The name of the .zip package should be one to eighteen alphanumeric characters in length and cannot begin with a number or an underscore. The .zip file consists of shell scripts that will run in order to configure the database as required. The scripts are run as the db2inst1 user. These scripts are contained in the possible first-level subdirectories listed on the slide. Only the 'create_db' subdirectory is required. Each of these directories must contain an entry script that is invoked and that script can then call additional scripts under the same directory. You will see more information on each of these directories on the next few slides.

Entry scripts

- Entry scripts are named as follows and invoked in this order:

- 1.tune_inst.sh
- 2.create_db.sh (mandatory)**
- 3.tune_db.sh
- 4.init_db.sh
- 5.post_start_inst.sh

– All scripts return '0' if successful

The entry script names match the directory names in which they live. They are run in the order shown on the slide. Each script returns '0' if successful; a non-zero code is returned if there is a problem. The only required script and directory is create_db.sh. It is invoked to create the database after the DB2 instance is created.

create_db.sh

- **create_db.sh** -- invoked to create the database after the DB2 instance is created

- Parameters:

- instance name (always "db2inst1" and cannot be changed)
- database name

```
inst_name=$1
db_name=$2
```

```
db2 "CREATE DATABASE ${db_name} ON /home/${inst_name} USING
CODESET UTF-8 TERRITORY US COLLATE USING SYSTEM PAGESIZE
${pagesize} K dft_extent_sz 16 AUTOCONFIGURE APPLY NONE"
```

```
rc=$?
if [[ ${rc} -ne 0 ]] ; then
    echo "Failed to create database."
    exit ${rc}
fi
```

Again, create_db is invoked to create the database after the DB2 instance is created. It is passed the instance and database name as parameters. This is shown in the example on the slide. The example then shows the default 'CREATE DATABASE' command for the Departmental Transactional workload standard. The return code should then be returned.

Pre-create_db script

- **tune_inst.sh** – invoked to tune the DB2 instance
 - Parameters:
 - instance name
 - Use this format:
 - db2 "update dbm cfg using MAXAGENTS 10"
 - db2 "update dbm cfg using NUM_POOLAGENTS 8"
 - db2 "update dbm cfg using NUM_INITAGENTS 2"

There is one script that you can include to run before the creation of the database. The `tune_inst.sh` script is invoked to tune the DB2 instance that is deployed by PureApplication System. You can update the database manager configuration here. Examples of the command format are shown on the slide. The `tune_inst` shell script is passed the instance name as a parameter.

Post-create_db scripts

- **tune_db.sh** – tune the database
 - Parameters:
 - instance name
 - db_name
 - appuser
 - appuser password
 - appdba
 - appdba password
 - Invoked to create tablespaces or grant privileges to appuser/appdba after the database is created
 - NOTE: DBaaS uses /home/db2inst1/ as the database path so all new tablespaces must be created here
- **init_db.sh** – initialize the database
 - Parameters:
 - instance name
 - db_name
 - appuser
 - appuser password
 - appdba
 - appdba password
 - Create database objects such as schema, tables, views, procedures, and functions
 - Invoked to create database objects or load data into database

After the database is created with the create_db shell script, the tune_db script is called where you can create tablespaces and grant privileges to the appuser/appdba user IDs. Note that DBaaS uses /home/db2inst1/ as the database path so all new tablespaces must be created here. The appuser and appdba user IDs and passwords are passed in as parameters to this script.

Finally, the init_db script is called to allow you to initialize the database. Here you can create database objects and load data. You can call other files or scripts from this script if needed.

Start DB2-related processes

- **post_start_inst.sh** – invoked to start DB2-related processes after the DB2 instance starts
 - Parameters:
 - instance name
 - db_name
 - Begin processes such as **start db2 Text Search** after each start of the db2 instance
 - `db2ts "START FOR TEXT"`
 - Runs each time the db2 instance is restarted

You can include a `post_start_inst` shell script to start db2-related processes after the db2 instance starts. An example of this is the `'db2ts "START FOR TEXT"'` command. Note that this script is automatically invoked every time the db2 instance is restarted as well.

Results found in IWD Agent logs

Log Viewer [database-db2.11344982545099 (172.17.110.201)]

Refresh Download All
Monitor end of log file

GO

- database-db2.11344982545099
 - DB2
 - DeploymentInlet
 - IWD Agent
 - /oconfig
 - ./debug/output
 - ./ITM/logs
 - ./logs/database-db2.11344982545099
 - ./logs/database-db2.11344982545099.AGENT
 - ./logs/database-db2.11344982545099.DB2
 - console.log
 - trace.log**
 - ./logs/database-db2.11344982545099.MAINTENANCE
 - ./logs/database-db2.11344982545099.MONITORING
 - ./logs/database-db2.11344982545099.OPMDB2
 - ./logs/database-db2.11344982545099.OQWTD82
 - ./logs/database-db2.11344982545099.SSH
 - ./logs/install
 - OS

```

/opt/IBM/maestro/agent/usr/servers/database-db2.11344982545099/logs/database-db2.11344982545099.DB2/trace.log
[08/14/2012 22:29:20] INFO: Updating DB2 instance configuration...
[08/14/2012 22:29:20] INFO: Updating DB2 instance configuration...
UPDATE DBM CFG USING SYSADM_GROUP g_sysadm SYSCTRL_GROUP g_sysctl SYSMANT_GROUP g_sysmnt SYSMON_GROUP g_sysmon
DB20000I The UPDATE DATABASE MANAGER CONFIGURATION command completed
successfully.
08/14/2012 22:29:23 0 0 SQL1064N DB2STOP processing was successful.
SQL1064N DB2STOP processing was successful.
08/14/2012 22:29:25 0 0 SQL1063N DB2START processing was successful.
SQL1063N DB2START processing was successful.
[08/14/2012 22:29:25] End: /opt/IBM/maestro/agent/usr/servers/database-db2.11344982545099/scripts/DB2/shell/post_create_db.sh
[08/14/2012] DB2/install.py 140644782761728 pid=30646 DEBUG Start tune_db.sh deploy99
inst_name: db2inst1
db_name: deploy99
appuser: appuser
appuserPW: 18V1sCNuWZk1cSSL
appdba: appdba
appdbaPW: dd6y5zGzQ7kCAs
End tune_db.sh deploy99
[08/14/2012] DB2/install.py 140644782761728 pid=30646 DEBUG Start init_db.sh deploy99
inst_name: db2inst1
db_name: deploy99

```

24

Database-as-a-Service custom workloads

© 2012 IBM Corporation

To see the results of the scripts' execution, you can look in the DB2 IWD Agent logs. Any failures can be found there. In this case, the tune_db.sh output the parameters passed into the script.

IPASv1_VApp_DBaaS_CustomWorkloads.ppt

Page 24 of 27

Summary

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

Summary

- Creating customized workload standards
 - Added to the catalog
- Customized workload standard package definition
 - Packaged as a .zip file
 - create_db only required script

This presentation showed you how to create custom workload standards that can then be used to deploy your databases. Your custom workload standards are added to the catalog for use by your database patterns. To create a customized workload standard, you need to create a .zip file with a set of predetermined directories that each contain an 'entry script'. The only required directory and script is create_db, the one to create your database. The others allow you to further tune your database and start additional services, if needed.

Trademarks, disclaimer, and copyright information

IBM, the IBM logo, ibm.com, DB2, 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>

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. 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.