Automate application-specific storage provisioning with Rational extensions

Using the IBM Deployment Planning and Automation for the Cloud 2.1.0 accelerator

Sreenivasulu Valmeti  
Snehal Pansare  
Indrajit Poddar  
June 24, 2014

Learn how to model and automate the additional storage needs of an application using IBM Deployment Planning and Automation for the Cloud 2.1.0. This freely downloadable accelerator installs on top of and integrates capabilities in IBM Rational® Software Architect, IBM Rational Automation Framework, and IBM Tivoli® Service Automation Manager. In this article, you can discover how to use this tool to specify custom storage requirements for application components in virtual images through a deployment topology model and use the information to generate deployment automation artifacts that provision the storage elements and attach them to virtual machines in the cloud.

The article "How to reduce deployment time for composite solutions with the integration asset" describes how to use the IBM Deployment Planning and Automation for the Cloud accelerator to generate deployment automation artifacts from application topology models for building and configuring virtual images with middleware and application components. (A topology is a type of model that shows the relationships between IT resources.)

The latest release of the accelerator supports automatically provisioning storage elements prior to application deployment. With this new feature, the accelerator lets software architects specify custom storage requirements for application components in virtual images through a deployment topology model. Later in the process, this information can be used to generate deployment automation artifacts that provision the storage elements and attach them to virtual machines in the cloud.

The accelerator’s integration asset allows architects to use the Rational Software Architect solution modeling tool (RSA) to design application topology and generate workflows for multiple deployment engines such as Tivoli Service Automation Manager (TSAM) and Rational Automation Framework (RAF) — this lets you deliver end-to-end solution deployments.
The generated workflows include, in the same flow:

- TSAM steps for provisioning VMware virtual machines and System p® logical partitions (LPARs), and for installing middleware
- RAF steps for configuring the middleware components and installing applications

V2.1.0 allows customized storage elements in the virtual images. For example, consider a deployment scenario for a Java™ Enterprise Edition web application (an EAR) to the shared virtualized infrastructure. This application requires a web application server and database as its back end, as shown below.

In this scenario, a solution architect named Sally requires that the application server and database server be installed on an additional hard disk — one different from where the host OS is deployed. She wants the two hard disks to be independently allowed to expand as needed.

With the accelerator, she can design the required deployment topology model with a server model unit consisting of a web server model unit (such as WebSphere® Application Server) and a database server model unit (such as IBM DB2®). While designing the deployment topology model, V2.1.0 allows the additional hard disk to be explicitly modeled as an additional storage model unit to be associated with the virtual image model unit. At deployment time, TSAM finds a suitable VMware cluster or Central Electronic Complex host (CEC), which supports System p images to deploy the server with the storage characteristics specified in the storage element.

Through this deployment topology model, deployment engineer Doug can offer a standardized topology model for deploying applications to a shared enterprise computing infrastructure. Using
this standardized topology model, solution architects Sally and Sam can independently deploy two versions of the JavaEE application to the shared infrastructure.

In this manner, Doug, Sally, and Sam can deploy solution assets more efficiently and with fewer errors.

Now that we've provided a scenario-based look at the accelerator's ability to help you model an automate storage provisioning, let's demonstrate some other tasks:

- Installing the accelerator
- Modeling storage extensions in Rational System Architect
- Generating custom TSAM Service Definitions and RAF Projects
- Importing and requesting a custom service in TSAM
- Installing application components and configuring middleware with RAF

Install the integration asset

Deployment Planning and Automation for the Cloud can be downloaded from the Tivoli Integrated Service Management Library. This asset installs on top of the following prerequisite products:

- Rational Software Architect 8.5.5 (RSA) with
  a. Extension for Deployment Planning
  b. Extension for Deployment Automation Planning
  c. (Optionally) IBM Rational Deployment Automation Content Pack for RAFW and WebSphere Application Server
- Rational Asset Manager 7.5.0.1
- Rational Build Forge 7.1.2.2 optionally with Rational Automation Framework for WebSphere 3.0
- Tivoli Service Automation Manager 7.2.4 with Tivoli Provisioning Manager (TPM) The integration asset includes detailed instructions for installing on top of the above prerequisite products.

Model storage extensions in RSA

The integration asset includes detailed instructions for installing on top of the above prerequisite products.

With RSA, solution architects Sally and Sam can create deployment topologies representing a shared development environment. Deployment topologies in RSA model information technology resources and their relationships. Topologies can help in planning and validating deployment scenarios. Topologies include model units and the links between those units. The units represent parts of data domains, such as servers, operating systems, middleware or application components. The links represent different types of relationships between units, such as hosting or dependency. Units contain several elements that represent the behavior and specific features of the corresponding resource in the real world.

Figure 3 shows a sample topology model representing a VMware virtual image with a WebSphere Application Server, an OS, database server (DB2), and a JavaEE standard EAR installed in it.
It is possible to associate an application component with a URL to a Rational Asset Manager asset from **Properties > Artifacts > Add** button.

Using such URLs and versions, Sally and Sam can associate different topology models with their own versions of an application EAR for deployment.

To design the storage elements, Sally adds the VMware Virtual Disk Def template from the Virtualization drawer of the Palette to the topology.
The VMware Virtual Disk Def unit will be added to the topology. Sally sets this unit's installState as "to be installed" and creates a hosting link to the VMware Virtual Image unit.

On the VMware Virtual Disk Def unit, Sally adds the detail configuration to the following capability properties as per the storage needs of the application:

1. Additional file system types:
   - For a Linux® image, Sally specifies one of the file system type ext2/ext3/None
   - For a Windows® image, Sally specifies one of the file system type NTFS/FAT32/None

2. Additional mount points:
   - A directory/drive where additional storage will be mounted
   - For Linux, Sally specifies a directory name
   - For Windows, Sally specifies a drive letter

   Note: Do not use standard mount points like /root, /bin, /opt, etc. This will be the directory where the application server (WebSphere Application Server) and database server (DB2) will be installed.

3. Additional storage pool names
   Sally specifies Cloud Storage Pool name as defined in TSAM. For example, Figure 7 shows the Name property in the Cloud Storage Pool Definition tab in TSAM. She specifies this name for each of the hard disk that she wants to add separated by a commas.

4. Additional storage unit sizes in GBs
   Specify GB size for the additional storage; size must not be in fractions. For example, to enter two additional hard disks with the following specifications:
• Storage 1 details:
  File System Type: ext2
  Mount Point: mnt0
  Storage Pool Name: AD1
  Storage Unit size: 1

• Storage 2 details:
  File System Type: ext3
  Mount Point: mnt1
  Storage Pool Name: AD2
  Storage Unit size: 2

Sally enters the values in the RSA Virtual Disk Def model unit.

**Figure 8. Capturing the storage need details in storage disk unit**

After a topology model is developed completely in RSA, it can include model units representing database components and application resources, as well as one or more virtual images. Figure 9 shows a completed sample topology model corresponding to the sample scenario. Sally and Sam can now consult with Doug to specify deployment parameters, such as virtual image ID and passwords for the shared deployment environment. Once built, such topologies can be standardized for use in multiple deployments.
Automate application-specific storage provisioning with Rational extensions

Figure 9. Completed topology with WebSphere Application Server, DB2, application and storage resources in a VMware virtual image

Generate a custom TSAM service definition and RAF project

Sally and Sam can generate TSAM Service Definitions and RAFW projects from these RSA topologies. TSAM service definitions are stored in Cloud Service Archives. Generated RAFW projects are directly imported into RAFW. This is done in a two-step process. In the first step, Sally or Sam generates a Rational automation workflow model from the topology model. In the second step, they use the Publish topology menu for the generated automation workflow model from the first step in RSA to generate the Cloud Service Archive and the RAFW project.

Figure 10. Generate a TSAM Cloud Service Archive and RAFW project from RSA
An RSA automation workflow model is an ordered list of automation signature models related to a topology model. An automation signature model is a template for an automation task that runs on an IT system through a deployment engine.

For example, a task could be for provisioning a virtual server using TSAM. Automation signatures for automation tasks in TSAM, RAFW, Build Forge, and Apache Ant are available out of the box in RSA and by installing the sample project archive included in the integration asset.

Automation signature models connect names of automation tasks in different deployment engines, and their input and output parameters to different model units in the topology model. When an automation workflow is generated, all such parameters have values propagated from the topology.

One master automation signature can be applied to different topologies to pass in different attribute values to different generated automation workflows. Figure 11 shows an example of such automation signature that connects the automation task of provisioning virtual servers using TSAM as the deployment engine.

**Figure 11. Example of the automation signature for deploying virtual servers with TSAM**

This signature also maps the input parameters for this task, such as the imageID, to a similar attribute in the topology.

Each automation signature has a strong relation with topology model units and a deployment engine. New automation signatures for new model units or deployment engines or tasks can be added through new RSA projects. The order of steps in the generated automation workflow is derived automatically based on relationships between units in initial automation topology. This order can be changed and new steps can be added or removed or modified in the generated automation workflow model.

Sally or Sam can see the results of the generation steps in the "Publish report" view in RSA.
Figure 12. Results of publisher work

Governing application asset lifecycle using Rational Asset Manager can help manage application assets involved in software delivery. In this section, Sally and Sam create a new parameter in the generated Cloud Service Archive to specify a different version of the RAM asset storing their own application EAR file.

In the RSA automation workflow model, there are global parameters that can provide input to any automated task in the generated automation workflows. These parameters are automatically transferred to a custom service in TSAM as mandatory parameters that a service requester must specify before requesting the service.

For example, Sally and Sam will create new global parameter: DevTeamAssetVersion and associate it with the AssetVersion parameter of the download deployable assets automation task.

Figure 13. Rational automation workflow with new created DevTeamAssetVersion global parameter
Figure 14. Mapping of the new created global parameter to existing parameter of the automated task

After this customization, Sally and Sam can generate the Cloud Service Archive and the RAFW project as already shown.

**Import a custom service in TSAM**

TSAM enables requesting, deployment, monitoring, and management of cloud computing services with support for approvals and processes. When the accelerator's TSAM extension module (tsam_csd_pmp7.2.4.zip) is installed, a new **Import Service Definition** button gets added to the toolbar in the TSAM administrator console in the Service Definitions view.

Figure 15. Import Service Definition button on Service Definitions Application toolbar
This button allows deployment engineer Doug to import the Cloud Service Archive files generated from RSA (as described previously) into TSAM. During the import process, multiple TSAM artifacts like service definitions, management plans and offerings, etc. for managing new services are be created automatically.

To import the Cloud Service Archive, Doug specifies the full path to the Cloud Service Archive in a local directory, as well as the category and the catalog. The category specifies the TSAM self-service UI category, under which the new service will appear. By default, this should be specified as "Custom Services." The catalog specifies catalogs to which new offerings will be added. The selection of catalogs to which the offerings will be added controls the user groups (such as cloud administrators, service requesters that will be able to see and select the new offerings in the TSAM self-service UI).

**Figure 16. Specify category and catalog for new service created after import**

![Figure 16](image)

Upon importing the Cloud Service Archive, a new custom Service Definition along with Management plans defined are created with the status as approved. New revision of the same service definition are created if same the same deployment topology is imported with changes. Offerings created for each management plan in Figure 17 appear on the TSAM self-service UI and can be used requesting this service.
Figure 17. New service definition created for Cloud Service Archive

Request a custom service in TSAM

Figure 18 shows offerings created in the TSAM self-service UI for requesting the generated service definition. The provisioning service, for example `Provisioning_SingleTierDeploymentTSAMForVMware_Workflow`, can be used to deploy the application into a VMware virtual server in a shared cloud environment as a service. Similarly, the de-provision service (`Deprovisioning_SingleTierDeploymentTSAMForVMware_Workflow`) can be requested to remove the deployed virtual server from the shared cloud environment.

Figure 18. Management operations to perform on cloud service created as a result of import

Sally can design the generated workflow model in RSA to include a global parameter so service requesters can specify the version of the application asset in RAM. This attribute appears in the self-service UI as a mandatory input parameter.

Figure 19. Customized attribute as an input during Requesting a New Cloud Service instance
When request to provision a service is initiated, TSAM triggers automatically generated workflows from RSA, which provisions a virtual server, attaches additional hard disks to the virtual server, installs WebSphere Application Server on the first mapped hard disk, installs DB2 on the second mapped hard disk, then invokes the RAFW project, which configures WebSphere Application Server and DB2, and installs the JEE application on top of them. This way, the service requestor gets a running application into production with a single click.

Now Sally and Sam can request their own versions of an asset under development to be deployed through the TSAM self-service UI. With more customizations to the generated workflow model in RSA, other parameters can be specified, such as the prerequisite OS version, and user IDs and passwords, etc. Doug can also standardize other parameters in the deployment topology such as the versions of WebSphere Application Server and DB2.

Install application components and configure middleware with RAF

Using the accelerator’s integration asset, Sally can generate a customized automation workflow for installing an application EAR file onto virtual partitions deployed from TSAM. Figure 20 shows an RAFW project generated from the RSA topology model described in Figure 8.

Figure 20. Generated RAFW project for installing a WebSphere and DB2 application eTierDeployment topology model

During the execution of the TSAM provisioning service generated by the accelerator (as shown here), the last task in the management plan invokes the generated RAFW or BF project. The TSAM task passes in parameters required for application installation and middleware configuration from the topology model (for example, WebSphere Application Server administrator password) or from the output of previous tasks for provisioning the virtual servers or installing the middleware (for example, hostname of the virtual machine deployed to host the WebSphere Application Server). Figure 21 shows an example environment with the parameters passed into a RAFW project from TSAM.
Figure 21. Environment variables for the generated RAFW project

Doug can now monitor the status of the BF job in the RAF console.

Figure 22. RAF job results reflecting the status of the executed automation for installing apps and configuring middleware

Conclusion

In this article, we have described how to model and automate the additional storage needs of application by using the IBM Deployment Planning and Automation for the Cloud accelerator.

© Copyright IBM Corporation 2014
Trademarks
(www.ibm.com/developerworks/ibm/trademarks/)