Technical Report

IT as a Service Storage Provisioning Using NetApp Workflow Automation, fluid Operations eCloudManager, and Activiti

Stefan Ebener, NetApp

Andreas Eberhart and Irina Parepa, fluid Operations

June 2012| TR-4055

Abstract

This document provides IT administrators and operational staff with information about how NetApp® shared storage integrates with fluid Operations™ eCloudManager™ and Activiti™ automation using NetApp Workflow Automation (WFA). It also demonstrates the simplicity of the product integration, which reduces implementation risk. The document discusses the integration of storage provisioning workflows and their consumption using eCloudManager's flexible service catalog framework and industry-standard workflows shown in the Business Process Modelling Notation (BPMN). The architecture outlined in this document produces a clean separation between administrative roles in a data center and promotes cost efficiency as it adapts the IT infrastructure to future needs.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 Executive Summary

NetApp Workflow Automation (WFA) is a powerful but easy-to-use framework that helps accomplish storage-centric automation tasks.

WFA enables the custom design and automated execution of administrative tasks, such as storage capacity provisioning, system setup, data migration, and storage reclamation. Tasks of this sort can be executed easily by downstream or outsourced administrators with the click of a mouse and without customized scripts.

This document describes how NetApp WFA storage workflows can be leveraged from fluid Operations (fluidOps™) eCloudManager and Activiti. NetApp WFA addresses agility, flexibility, high efficiency, and repeatability for storage layer best practices.

This document:

- Describes the concept and setup of WFA storage workflows and highlights WFA's value for large infrastructures
- Demonstrates how to simplify storage automation in complex and growing environments by using a simple, service-oriented interface
- Describes how eCloudManager interacts with NetApp WFA

2 Introduction

This document discusses the integration of NetApp WFA with eCloudManager data center automation software. The primary focus of this implementation is to provide automated storage provisioning in a highly efficient NetApp shared storage infrastructure.

2.1 Problem with Existing Approaches

The management and operation of today's data centers are changing dramatically. Data centers and IT departments rely on service organizations to unite conflicting requirements and trends. They are expected to offer IT as a service (ITaaS) in a way that is flexible, managed, and secure, but that is also instantaneous and cost efficient. The architecture of the automated software stack must:

- Implement storage practices and storage layouts according to the industry's best-practice recommendations and naming conventions.
- Address the specific needs of a growing and changing storage infrastructure.
- Address the roles of storage administrators and consumers.

These attributes are critical to achieving agility in IT infrastructure and in the automated software stack. A role concept and the separation of storage workflows from other IT-implemented workflows create a cost-efficient and an easily maintained automated solution that can be kept in sync with infrastructure changes. It prevents implementation of storage best practices inside generic frameworks that have different maintenance and update cycles, and therefore cannot be kept in sync with the IT transformation.

A high degree of automation combined with industry-standard best practices creates documented and repeatable end-to-end service delivery. NetApp WFA provides a flexible management layer with self-service interfaces for customers, which means it can function as the central storage provisioning tool for an entire service lifetime—from the instant delivery of service-to-service deprovisioning. Workflow automation enables customized storage workflows to be automated without customized scripts, which can be difficult to maintain.

2.2 Scenario and Use Case

The integration of eCloudManager and WFA is described from the perspective of a large enterprise or service provider. We assume the presence of the eCloudManager IT service management framework. In this context, we describe the use cases for discovering existing NetApp WFA workflows and executing a selected NetApp WFA workflow.

In this scenario there are three roles:

- **Storage administrator.** A storage administrator determines storage-specific workflows using NetApp WFA. He or she defines and implements storage best practices for his or her organization and makes them available to a service requestor through the WFA Web Service Description Language (WSDL) interface or the WFA Web front end. He or she uses the WFA designer studio to specify the workflows that define how to consume storage in the shared storage infrastructure. The storage centricity of WFA maximizes agility in designing storage-centric workflows and adapting them to changes in the IT and storage infrastructure.
- **Service requestor.** A service requestor processes IT infrastructure requests. A service requestor might be a person manually processing tickets or a self-service help desk. The service requestor can select from the NetApp WFA workflow catalog to instantiate the required storage provisioning task. We describe how any service requestor discovers and instantiates WFA workflows from the eCloudManager framework.
- **End user.** An end user uses raw services or preconfigured compositions.

Figure 1 shows the service administrator and service requestor roles and responsibilities.

Figure 1) Architecture of WFA storage-centric workflows.

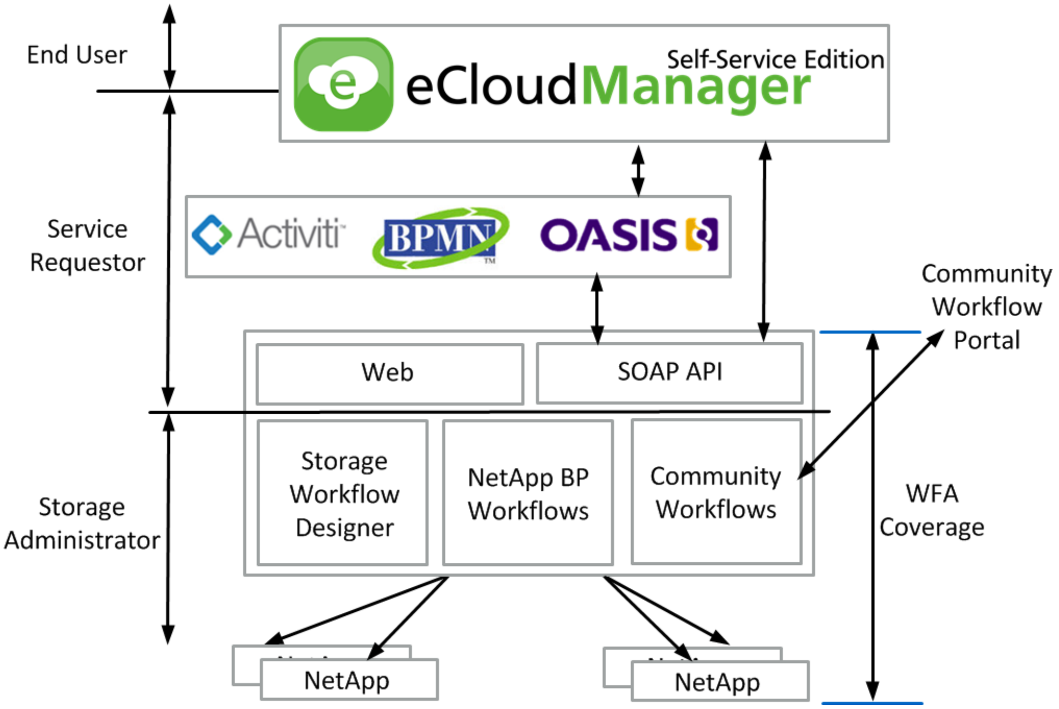


Table 1 summarizes the roles of storage administrators and service requestors.

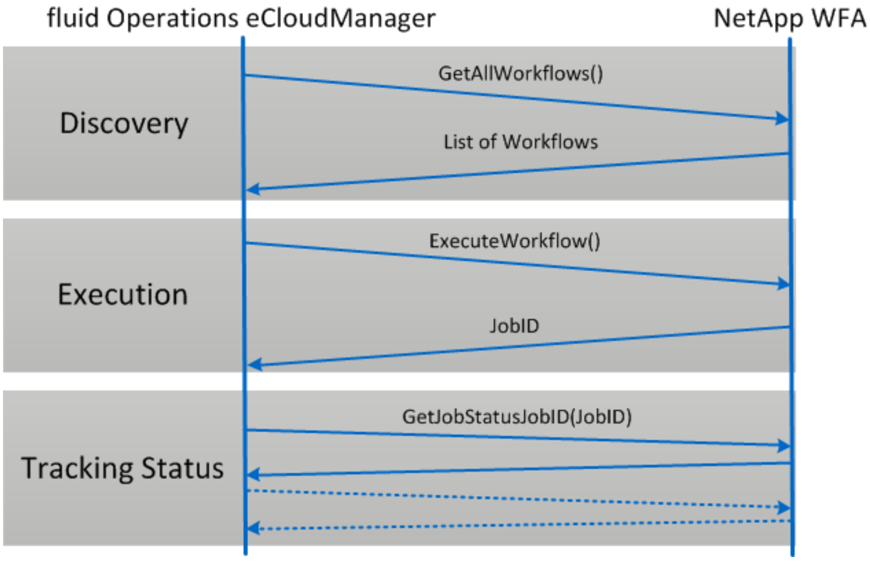
Table 1) Tasks to set up and consume storage-related services.

Role	Occurrence	WFA	fluidOps eCloudManager
Storage administrator	One-time	• Define storage workflows that reflect the best practices of customer provisioning • Adapt workflows based on changing demands and infrastructure	
Service requestor	Often (per service request)		Use WFA workflows

The term "workflow" exists both in the context of NetApp WFA and eCloudManager. For NetApp WFA, a workflow relates to a storage-centric workflow. For eCloudManager, a workflow can be translated into the broader context of supporting or implementing IT management processes. WFA workflows denote workflows implemented in the NetApp WFA.

Figure 2 depicts a typical scenario in which WFA workflows are triggered by eCloudManager.

Figure 2) eCloudManager triggers WFA workflow.



- **Discovery.** In this phase the existing WFA workflows are discovered from eCloudManager. A list of defined WFA workflows is returned including short descriptions and parameter definitions.
- **Execution.** A selected WFA workflow is given parameters and executed. The execution can also be scheduled for a point in time using the ScheduleWorkflow method. A job ID is returned that can be used to retrieve additional information about the execution.
- **Tracking Status.** The status of a WFA workflow execution can be retrieved using the job ID.

3 Software Components and Integration

This section describes how to integrate individual software components to implement storage provisioning. eCloudManager automates IT processes to reduce manual activity. eCloudManager orchestrates the functions of multiple technology management frameworks to execute end-to-end workflow. Adapters interface between eCloudManager workflows, IT components, and management frameworks. In a workflow, storage, such as an NFS volume, is provisioned and access credentials are used by a third-party management component that ultimately hands those credentials to the storage consumers.

3.1 NetApp Workflow Automation

The NetApp OnCommand® management software portfolio delivers a suite of monitoring and automation tools used to simplify storage-related tasks, especially in large shared-storage infrastructures.

NetApp WFA is a framework used to automate storage-provisioning tasks. It brings together feature richness and simplicity and can be customized easily to support specific needs and conventions. WFA comes with predefined base building blocks to realize individual provisioning needs.

WFA addresses the following challenges:

- **Manual processes are expensive and error prone.** Most storage management tasks such as provisioning, migration, and decommissioning are performed manually.
- **Long provisioning cycles keep consumers waiting.** In many cases it takes days or weeks to provide storage for an application. This includes selecting the resources and provisioning and updating the internal systems.
- **Errors in configuration can happen even with well-trained personnel.** Mistakes made while performing manual steps can result in outages and delays, wasting time and money.
- **Underutilizing the power of smart storage.** To fully leverage the power of the storage system, it must be optimized for the specific application. Frequently, best practices are not followed and advanced features are not used.
- **The cloud promise is delayed.** Using the cloud requires a high level of automation that cannot rely on manual processes. Many IT groups are being challenged by management to meet the self-service and IT automation goals that drive the business.
- **Total cost of ownership of automation.** Many organizations turn to in-house software engineers or partners to write custom code for automation. These approaches are often expensive and slow to adapt to the changing needs of storage consumers. In most cases, the total cost of ownership of custom software makes it an ineffective solution.
- **Clean storage automation architecture and general-purpose data center orchestration.** Although orchestration solutions can be beneficial for end-to-end automation, they lack a comprehensive storage component to meet customers' process needs. As seen in other domains such as monitoring, an expert storage solution is required to address storage automation requirements.

For a more detailed overview and background information on WFA, refer to the following resources:

- https://communities.netapp.com/servlet/JiveServlet/downloadBody/11606-102-3-30627/WFA_Product_Brief_1.2.pdf
- https://communities.netapp.com/community/interfaces_and_tools/wfa

A standard compliant WSDL interface enables triggering WFA workflows from almost any source and orchestration software. WFA relies on NetApp OnCommand for an up-to-date overview of the storage landscape.

NetApp OnCommand Unified Manager provides a single human interface and an application programming interface (API) for integration with other types of management software. The API uses well-defined XML objects for communication to provide transparency and interoperability. For easy integration, the NetApp Manageability Software Developer's Kit (SDK) provides libraries for all major programming languages that use the NetApp DataFabric® Manager API.

3.2 BPMN and Activiti

IT service management (ITSM) is the discipline of managing IT with a focus on customers' needs and requirements. Naturally, ITSM has a strong business process angle. A major challenge for customers in ITSM is the absence of a standardized business process notation that would allow processes to be exported and imported seamlessly into different engines and workflow designers. The proprietary nature of these systems makes it difficult to find well-trained staff for their respective solutions. This problem is addressed by leveraging the BPMN notation for the Runbook Automation field. BPMN is a well-known business process standard that has been in use in many other domains for some time. This document describes how eCloudManager can be combined with an off-the-shelf BPMN designer and editor to enable an end-to-end ITSM solution.

BPMN solutions typically include a workflow designer, a task management dashboard, and a workflow execution engine. With Activiti, the workflow designer is embedded in the Eclipse IDE to provide convenient and powerful editing on both a visual designer level and an XML level. BPMN allows both programmatic and human interactions to be represented. Code can be seamlessly integrated through Service Tasks. Human input is encapsulated in User Tasks. The eCloudManager API is called directly from Activiti Service Tasks that are part of higher-level workflows. The workflow-to-API binding can be done in the full-fledged Eclipse IDE together with the eCloudManager Client SDK. Workflows can be deployed right out of the workflow designer into staging or production environments.

3.3 eCloudManager Self-Service Edition

In addition to business processes, self-service is another important part of ITSM. The eCloudManager self-service solution features a flexible service catalog and billing component that can be used to expose BPMN processes or traditional scripts to nontechnical end users. The eCloudManager Self-Service Edition allows internal and external users to directly provision the infrastructure resources and enterprise applications they need with the SLAs they require, while making sure that the governance, billing, and metering of the services stay within the defined scope.

3.4 Assumption

We assume that relevant WFA workflows exist to demonstrate how discovery, execution, and tracking status work. Creating WFA workflows is beyond the of scope of this document. For more information about creating WFA workflows, refer to NetApp WFA Communities.

3.5 Integration: Getting Started with a Java Client

This section gives a technical overview of eCloudManager and WFA integration. WSDL is used to interact between the components. Within this context, a subset of the WSDL capabilities of WFA is used. For more information about the WSDL API capabilities of WFA, refer to the NetApp WFA Web Services Primer.

WFA supports using and monitoring its workflows through Web services. Web services are supported using two common WSDL SOAP bindings:

- Document/literal encoding, which is used most often by Java® applications

```
http://<WFA_IP address>/wfa-ws/WorkflowService_doc?wsdl
```

- RPC encoding, which is used most often by .Net, Visual C#, and Windows® PowerShell™

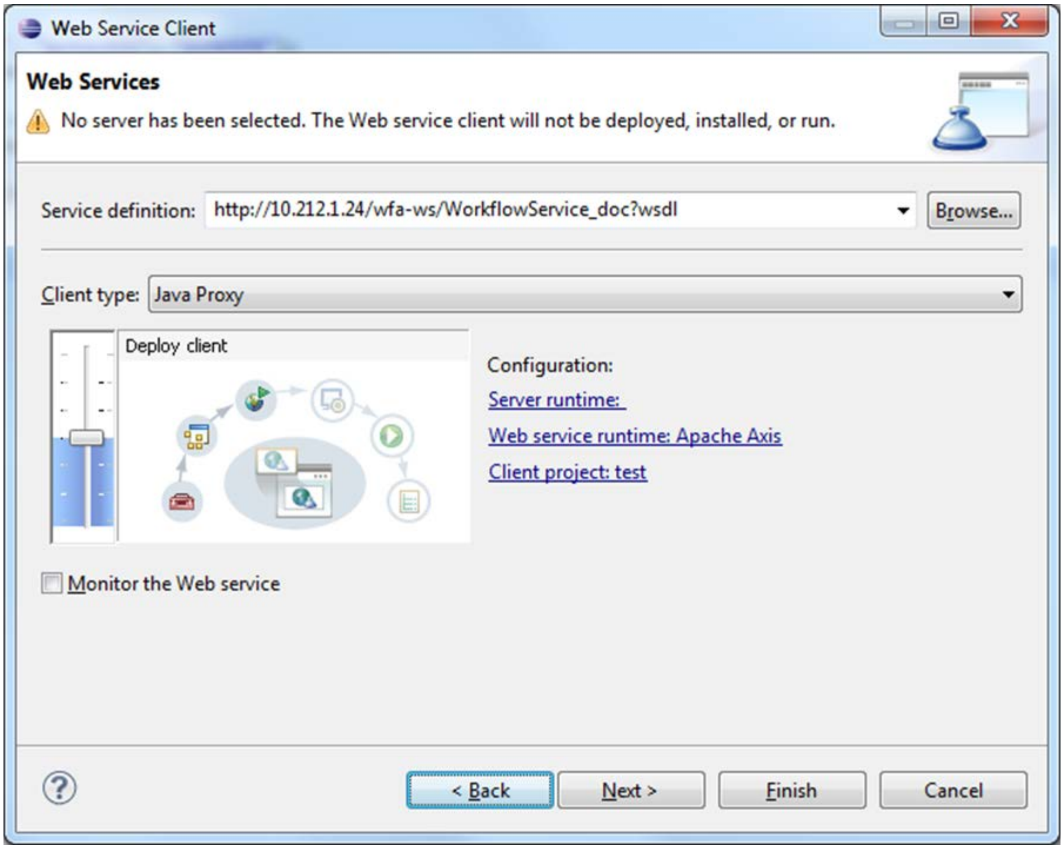
```
http://<WFA_IP address>/wfa-ws/WorkflowService_rpc?wsdl
```

In the context of the document, literal encoding is used.

Note: To attach to the service endpoint, the IP address or hostname of the WFA instance must be known. A Web service adapter on eCloudManager also needs to be present.

Both Activiti and eCloudManager allow callouts to Java classes to be defined that can connect to the WFA SOAP service. To illustrate this approach, the Eclipse designer with the Web tools platform enabled is used as a starting point. Creating a new Web Service Client project launches a software wizard in which the WSDL URL can be entered.

Figure 3) Wizard for creating a new Web Service Client.



⚠ No server has been selected. The Web service client will not be deployed, installed, or run.

Service definition: http://10.212.1.24/wfa-ws/WorkflowService_doc?wsdl

Client type: Java Proxy

Configuration:
Server runtime:
Web service runtime: Apache Axis
Client project: test

□ Monitor the Web service

As a result of the WSDL proxy object creation, the respective Java stubs for the WFA service are created. The WFA service can be invoked by entering the following client code.

```
import org.apache.axis.client.Stub;

import com.netapp.wfa.ws.Workflow;
import com.netapp.wfa.ws.WorkflowServiceLocator;
import com.netapp.wfa.ws.WorkflowService_doc;

public class WFAClient
{
        public static void main(String[] args) throws Exception
        {
                WorkflowServiceLocator loc = new WorkflowServiceLocator();
                WorkflowService_doc ser = loc.getWorkflowService_docPort();
                ((Stub)ser).setUsername( "admin" );
                ((Stub)ser).setPassword( "admin" );
                for ( Workflow w : ser.getAllWorkflows() )
                        System.out.println( w );
        }
}
```

Note: The default username and password of the WFA service are both `admin`. Use the username and password that are applicable for your environment. NetApp does not recommend using the default credentials in production.

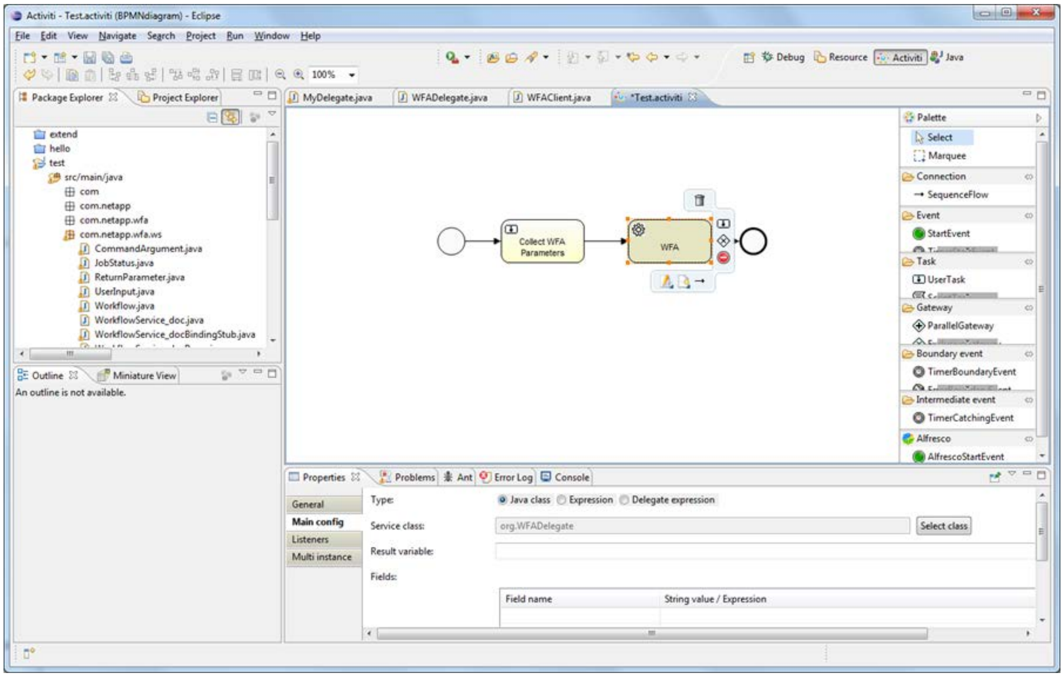
3.6 Calling a WFA Storage Workflow from Activiti

BPMN differentiates between service and user tasks. A user task appears in the BPMN task dashboard. From there, the user can delegate tasks and provide inputs that are represented in BPMN variables. Service tasks can invoke scripts and Java delegates. Our delegate invokes WFA via SOAP. Therefore, the following code is developed in Eclipse. We simply combine the JavaDelegate interface with the SOAP client created in section 3.5. The credentials are managed securely through the password store that is part of the eCloudManager SDK.

```
package org;

import org.activiti.engine.delegate.DelegateExecution;
import org.activiti.engine.delegate.JavaDelegate;
import org.apache.axis.client.Stub;

import com.fluidops.coremgmt.backend.PwdSafe;
import com.netapp.wfa.ws.WorkflowServiceLocator;
import com.netapp.wfa.ws.WorkflowService_doc;

public class WFADelegate implements JavaDelegate
{
        public void execute(DelegateExecution execution) throws Exception
        {
                // lookup service
                WorkflowServiceLocator loc = new WorkflowServiceLocator();
                WorkflowService_doc ser = loc.getWorkflowService_docPort();

                // http basic authentication
                String pwd = PwdSafe.getPwd("wfa");
                ((Stub)ser).setUsername( "admin" );
                ((Stub)ser).setPassword( pwd );

                // get workflow ID & parameters
                Integer workflowId = (Integer) execution.getVariable( "input" );
                String[] parameters = (String[]) execution.getVariable( "parameters" );

                // go
                int jobId = ser.executeWorkflow(workflowId, parameters);

                // return jobId to activiti
                execution.setVariable( "jobId", jobId );
        }
}
```

Next, the WFA invocation is integrated into the actual business process. This is done using the Activiti designer to reference the delegate with a Java service task.

Figure 4) Activiti Designer.



Type: ● Java class ○ Expression ○ Delegate expression

Service class: org.WFADelegate

Result variable:

Fields:

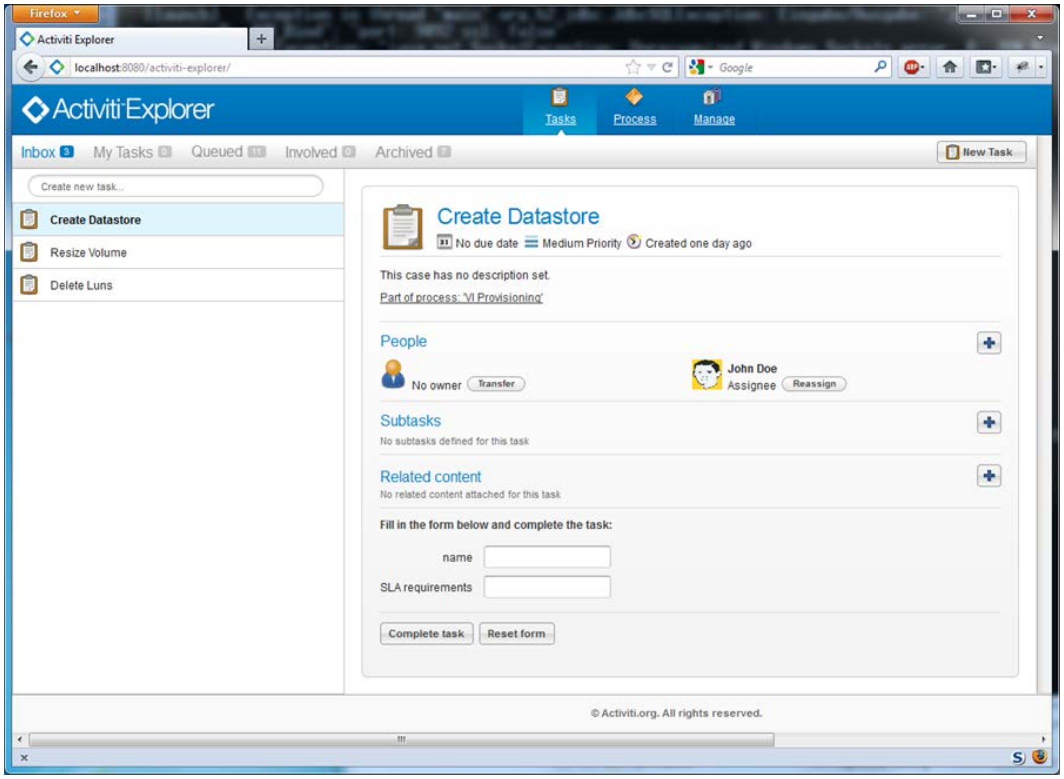
Field name	String value / Expression

To deploy the workflow into the BPMN engine, complete the following steps.

1. On the Eclipse project, right-click and select Create Deployment Artifacts.
2. This creates a `bar` and a `jar` file. However, the `jar` file lacks the manifest in META-INF. Generate the manifest using the Eclipse export wizard on the source folder that contains the Activiti delegates.
3. Copy the jar file to the J2EE application server to the same location as the SDK jar.
4. Click Administrate > Processes > Upload to upload the bar file using the Activiti management user interface.

This process can be run using the Activiti Explorer shown in Figure 5.

Figure 5) Activiti Explorer.



Create Datastore

No due date Medium Priority Created one day ago

This case has no description set.

Part of process: 'VI Provisioning'

No subtasks defined for this task

No related content attached for this task

Fill in the form below and complete the task:

1. Select the business process.
2. Select Create Instance.
3. Enter form values and watch `System.out.println()` on the Tomcat console.

3.7 Integrating WFA in eCloudManager Self-Service Edition

The eCloudManager Self-Service Edition (SES) offers flexible management of service catalogs that can include the Activiti workflow designed in section 3.6. Alternatively, WFA can be called directly from the eCloudManager Self-Service Edition. The development process is similar. By implementing commands such as `CatalogItem`, `CreateOperation` and `LifecycleOperation`, a relationship between eCloudManager's billing and user management and the WFA client is established. Furthermore, the customer can customize and define the interface used to enter the parameters required for the WFA workflow.

To deploy the extension, complete the following steps:

1. Compile and add the code (shown in section 3.6) to the eCloudManager classpath.
2. Activate Service Catalog Extensions on the eCloudManager CLI.

```
> setConfigOption  -key sesCatalogExtensions -value WFA
> setConfigOption  -key activePortalTabs -value
"Dashboard,Instances,Golden_Images,Golden_Landscapes,Account,Usage,Service Catalog"
```

3. Start eCloudManager.
4. Log into the Self-Service Portal and select the service catalog shown in Figure 6.

3.8 Tracking Status: Understand the Results of the Execution

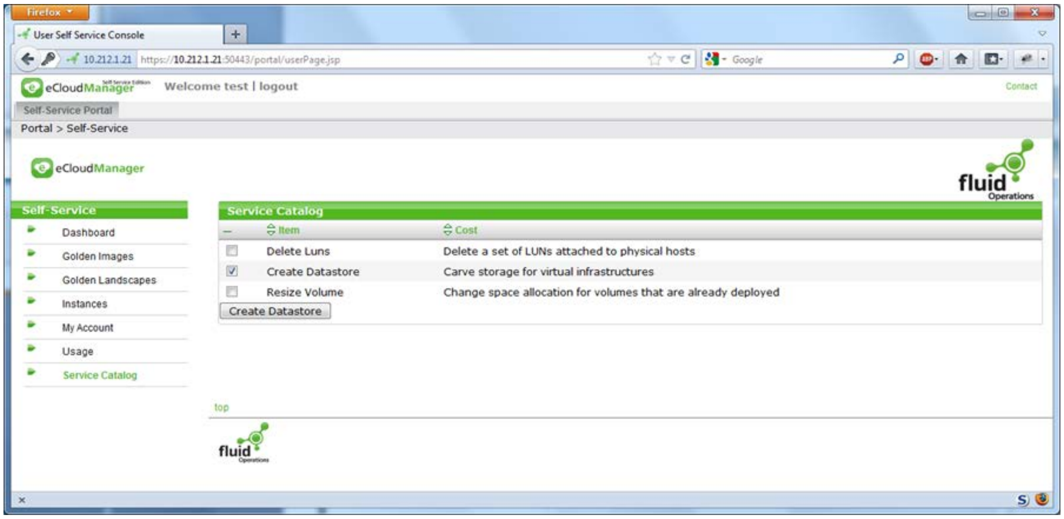
WFA performs actions asynchronously. This means that the ExecuteWorkflow method does not wait for the flow to complete. Instead a `jobId` is returned that can be used by clients to track the execution state and to retrieve job results. The following code can be invoked either in regular intervals or by user request. It uses the `jobId` obtained from the executeWorkflow call to query the job status, get potential error messages, and view results (for example, the path to a new volume).

```
JobStatus status = ser.getJobStatus(jobId);
if ( "done".equals( status.getJobStatus() ) )
{
        ReturnParameter[] results = ser.getReturnParameters(jobId, parameters);
        for ( ReturnParameter result : results )
                System.out.println( result.getName() + "=" + result.getValue() );
}
else if ( "error".equals( status.getJobStatus() ) )
        throw new Exception( "job " + jobId + " finished with an error: " +
status.getErrorMessage() );
```

4 Conclusion

This document describes how to integrate eCloudManager and NetApp WFA. It describes the clean architecture, the technical interaction, and the typical usage pattern for consuming NetApp WFA workflows. The examples used in this document outline the individual steps for setting up the integration and for consuming a selected WFA workflow that was triggered through eCloudManager. Moreover, this document demonstrates that realizing this integration in customer environments requires minimum time and effort and reduces implementation risks.

Figure 6) eCloudManager service catalog.



–	Item	Cost
□	Delete Luns	Delete a set of LUNs attached to physical hosts
☑	Create Datastore	Carve storage for virtual infrastructures
□	Resize Volume	Change space allocation for volumes that are already deployed

References

This document cites the following references:

- Activiti BPM Platform
  http://activiti.org/
- eCloudManager Product Suite
  www.fluidops.com/ecloudmanager/

- “NetApp WFA Web Services Primer”
  https://communities.netapp.com/docs/DOC-12977
- NetApp WFA Communities
  https://communities.netapp.com/community/interfaces_and_tools/wfa
- “NetApp Workflow Automation Product Brief”
  https://communities.netapp.com/servlet/JiveServlet/downloadBody/11606-102-3-30627/WFA_Product_Brief_1.2.pdf
- “Operations Manager Install and Upgrade Guide”
  http://now.netapp.com/NOW/knowledge/docs/DFM_win/rel40/pdfs/upgrade.pdf
- “Operations Manager Administration Guide”
  http://now.netapp.com/NOW/knowledge/docs/DFM_win/rel40/pdfs/opsmgr.pdf
- “Provisioning Manager and Protection Manager Guide to Common Workflows for Administrators”
  http://now.netapp.com/NOW/knowledge/docs/DFM_win/rel40/pdfs/workflow.pdf

Refer to the Interoperability Matrix Tool (IMT) on the NetApp Support site to validate that the exact product and feature versions described in this document are supported for your specific environment. The NetApp IMT defines the product components and versions that can be used to construct configurations that are supported by NetApp. Specific results depend on each customer's installation in accordance with published specifications.

NetApp provides no representations or warranties regarding the accuracy, reliability, or serviceability of any information or recommendations provided in this publication, or with respect to any results that may be obtained by the use of the information or observance of any recommendations provided herein. The information in this document is distributed AS IS, and the use of this information or the implementation of any recommendations or techniques herein is a customer’s responsibility and depends on the customer’s ability to evaluate and integrate them into the customer’s operational environment. This document and the information contained herein may be used solely in connection with the NetApp products discussed in this document.

Go further, faster®

NetApp®
www.netapp.com

© 2012 NetApp, Inc. All rights reserved. No portions of this document may be reproduced without prior written consent of NetApp, Inc. Specifications are subject to change without notice. NetApp, the NetApp logo, Go further, faster, DataFabric, and OnCommand, and OnCommand Unified Manager are trademarks or registered trademarks of NetApp, Inc. in the United States and/or other countries. Activiti is a trademark of Activiti. eCloudManager, fluid Operations, and fluidOps are trademarks of fluid Operations. Java is a registered trademark of Oracle Corporation. Windows is a registered trademark and PowerShell is a trademark of Microsoft Corporation. All other brands or products are trademarks or registered trademarks of their respective holders and should be treated as such. TR-4055-0612