# 

|  |  |  |
| --- | --- | --- |
| AD_cl_V_CMYK |  | *AppDynamics-Pro**Monitoring* *Enterprise**Applications*  *AppDynamics 3.4.2 (v1)* |

Welcome to the AppDynamics Technical Training Labs, these labs are design to pose important questions and draw from you a pertinent answer. The practical sessions will use a number of applications in particular BDR Big Deal Retail application .The labs primarily follow the unit and module flow shown below.

# Delivery Units

1. Unit 1: Technology Focus
   1. Fundamentals
   2. Understanding requests, metrics and transactions
   3. Product Walk Through- Acme Online Store
   4. Installation—Proof of Concept
2. Unit 2: Monitoring real time applications
   1. Real Time Monitoring,
   2. Distributed Transaction Tracing
   3. Infrastructure, Back End Performance & End User Monitoring
   4. Alerting & Remediation
   5. Diagnostic Model
3. Unit 3: Troubleshooting Performance Issues
   1. Triaging , Trouble Shooting and Custom Dashboards
   2. Diagnosing
   3. Performance Issues
4. Unit 4: Analysis, Automation and Administration
   1. Analysis
   2. Automation
   3. Controller Administration

Big Deal Retail BDR

Big Deal Retail is an online retail company selling last season’s and older model products. They have a full range of products that is for ever changing depending on what their buyers can purchase bulk from manufactures and large retail giants. They have a large customer base from the general consumer to small independent retailers who purchase from them for onward resell. They have a weekly sales catalogue of specials that is sent out to their 1.5 million online customers. They also do special campaigns and clearance sales which they advertise on the web site and also by email. The nature of their business is that a peak selling period is defined by what is on offer at that point in time. So to some extent they can anticipate increase traffic when the offer is heavily discounted. It is very important to BDR that their online ecommerce application is running very efficiently and that the user experience is of the highest expected. Speed of transactions and management of traffic is paramount as very often customers are competing against each other to make the purchase. It is not good for business if customers feel the system is in some way inhibiting their purchase.

**Tom Cash** who is the IT Director at BDR Big Deal Retail is keen to introduce AppDynamics to BDR having previously had the AppDynamics product deployed to good effect in his previous company ‘Two Cents Worth’.

**Cody** **Barr** who is the Senior Administrator at BDR Big Deal Retail is the main point of contact for application performance related issues. They are considering an on premise solution and Brad’s team will be responsible for this implementation and the ongoing monitoring.

**Penny Wise** is an account manager with AppDynamics concentrating on the retail sector. She has been selling the product for the past three years and was the top sales person last year.

**Chase Carter** is a sales engineer with AppDynamics and works in conjunction with Penny to support the retail market place.

**Bugsy Reader** who is principal professional services engineer at AppDynamics with 10 years’ experience in APM and AppDynamics real time monitoring is the professional services consultant appointed to support Big Deal Retail should they decide to buy.

**Budd Checkov** is an AppDynamics support agent dealing with potential issues from purchasing customers.

**Lab 1.1 Fundamentals**

Following a phone call from Tom Cash to Penny Wise, Tom and Cody are keen to meet with Penny and Chase to discuss a possible purchase of the AppDynamics solution. Chase has decided to give a short presentation of the product features to Cody .Some questions have arisen following the demo can you supply the answers needed in (a) and (b) below:

|  |
| --- |
| 1. Monitoring..Analyzing..Improving are three key stages in success of APM. What are the important elements within each stage that Tom Cash and Cody Barr should be made aware of? 2. Tom Cash has asked Chase ’How will AppDynamics contribute to APM success within their Big Deal Retail Application’. What would you expect his reply to be?   **Lab 1.2 Requests, Metrics and Transactions**  Cody Barr is a little confused regarding the metrics and how they relate to the business transactions can you assist by providing a clear definition in (a).   1. Describe the relationship between requests, metrics and transactions   Tom has asked a direct question about Errors recorded and has asked Chase Carter to describe the errors node on the screen. What would be your interpretation   1. Outline the meaning of the metrics you see below     **Lab 1.3 Product Walk Through Questions: Following his presentation Chase give a short walk through of the product using the ACME demo application. Pay particular attention to the demo and answer the next 3 questions with relevance to the ACME online store.**  The following questions were raised by the BDR team.  Q1 Can you describe the architecture of ACME?  Q2 What is the main difference between a threshold and a baseline  Q3 What’s the metric value of Information Points   1. ACME online store Walk Through -----Sketch the architecture of the Acme online store 2. Differentiate between baselines and thresholds |

1. Information points hold which type of metrics?

BDR Big Deal Retail

Basic Architecture

8-Processor

J2EE

Application

Server Cluster

BDR

J2EE

Application

Load

Balancer

SQL Servers on Amazon EC2

4 Linux Apache

Web Servers

# Lab 1.4 Installation (Proof of Concept)

# Following the meeting to make a purchase with Penny, Tom has decided to purchase but has asked for a proof of concept to be carried out as soon as possible. Chase has arranged a time and date and will go onsite to setup and manage the POC.

# You are to assume the role of Chase Carter and perform the POC as detailed below

# POC setup check:

In order for the POC sessions to run smoothly we need to do some preparation, this will ensure your machine is working correctly.

# Lab POC configuration:

Before we install the relevant agents and controller we need to modify some of the xml files to refer to BDR application, the details are outlined below go ahead and make the necessary changes.

**BDR Big Deal Retail Environment**

Controller3.4.2

* + Installed /AppDynamics/AppDynamics-controller
  + Can be Installed as a service named “controller”
  + Listens on port 8090

Username:, Password: can be set at install time for the controller, please make a note of these values.

AppServer Agent

* + Installed /home/user/AppDynamics/AppServerAgent
  + Contains configuration information for connecting to the controller

Machine Agent

* + Installed /home/user/AppDynamics/MachineAgent
  + Can be Installed as a service named “machineagent”

**Configuration:**

* **AppServerAgent3.4.2 (conf/controller-info.xml)**

<controller-info>

<controller-host>localhost</controller-host>

<controller-port>8090</controller-port>

<application-name>BDR Big Deal Retail</application-name>

<tier-name>1stTier</tier-name>

<node-name>1stTierNode2</node-name>

</controller-info>

* **MachineAgent3.4.2 (conf/controller-info.xml)**

<controller-info>

<controller-host>localhost</controller-host>

<controller-port>8090</controller-port>

<application-name>BDR Big Deal Retail</application-name>

<tier-name>1stTier</tier-name>

<node-name>1stTierNode2</node-name>

</controller-info>

**Start the MySQL**

1. Navigate to c:/appdynamics-controller/db/bin
2. Run mysqld

**Start an agent in standalone mode**

1. Make sure the controller-info.xml to contain correct values shown above.
2. Ensure your PATH points to a java runtime jre version there is one deployed with the controller at c:\appdynamics-controller\jre\bin if you don’t have a version installed on your machine.
3. Run the following batch file: run\_multitiered\_testApp.bat, this will trigger 3 agents. The file is found in C:/testapp

**Start a machine agent**

1. Modify controller-info.xml to contain the correct values for the following:
   1. Host name
   2. Port number
   3. Application name
   4. Tier name
   5. Node name
2. Run the following batch file: machineAgent.bat   
     
   The “-Dmetric.http.listener=true” system property(in the batch file) is not necessary to start the machine agent but we can use it later to show how to pipe in custom metrics. If you wish you can create a .bat file for future use.

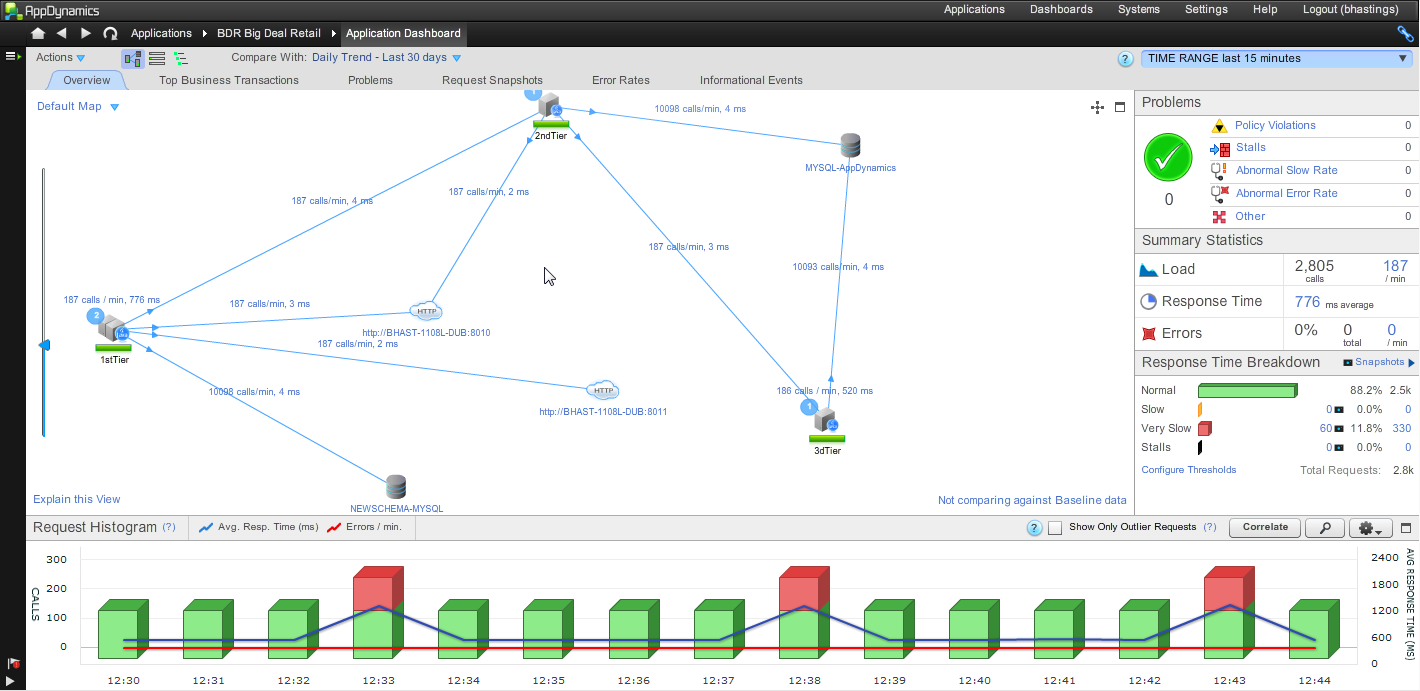
**Start the controller**

1. Navigate to c:/appdynamics-controller/bin
2. Execute the batch file startController.bat
3. You should see the controller database
4. Start a browser
5. Input url http://localhost:8090/controller/

Allow 5 minutes for controller and agents to startup and connect. You should be seeing an application overview screen similar to the one below

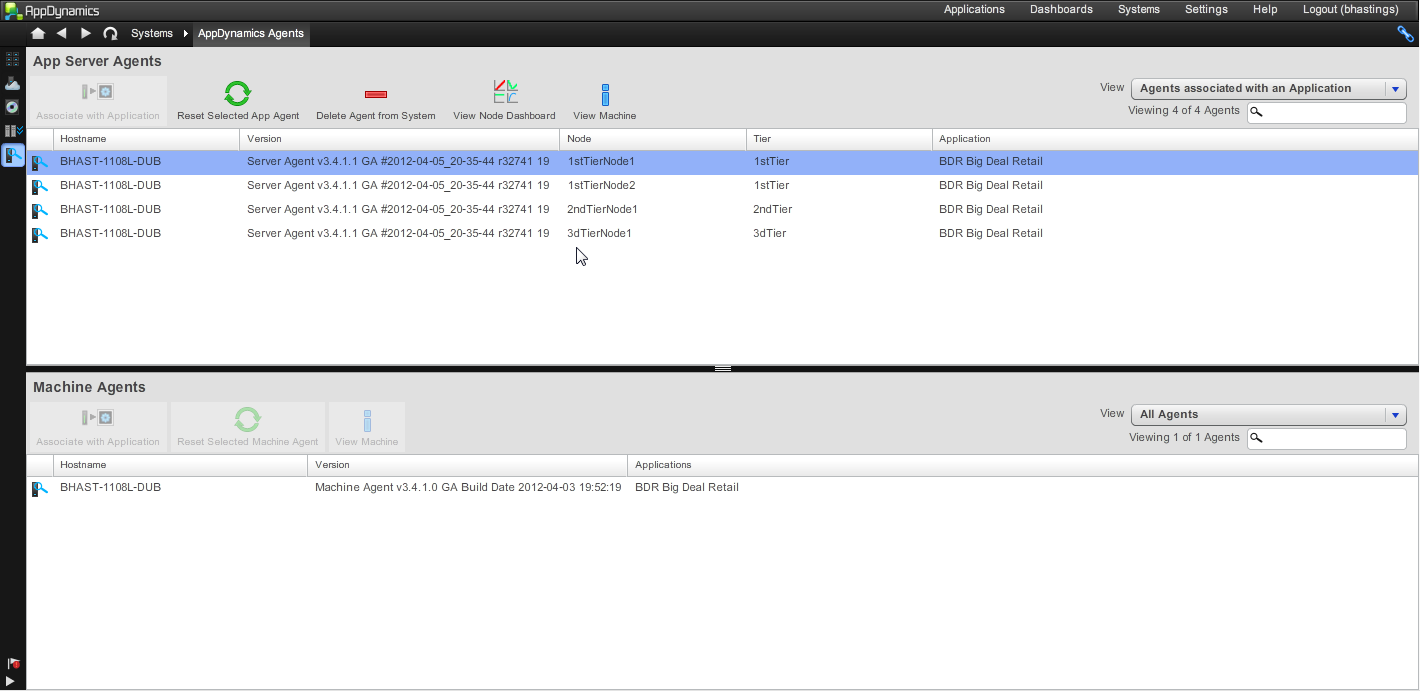
BDR Big Deal Retail

What you should expect to see on startup

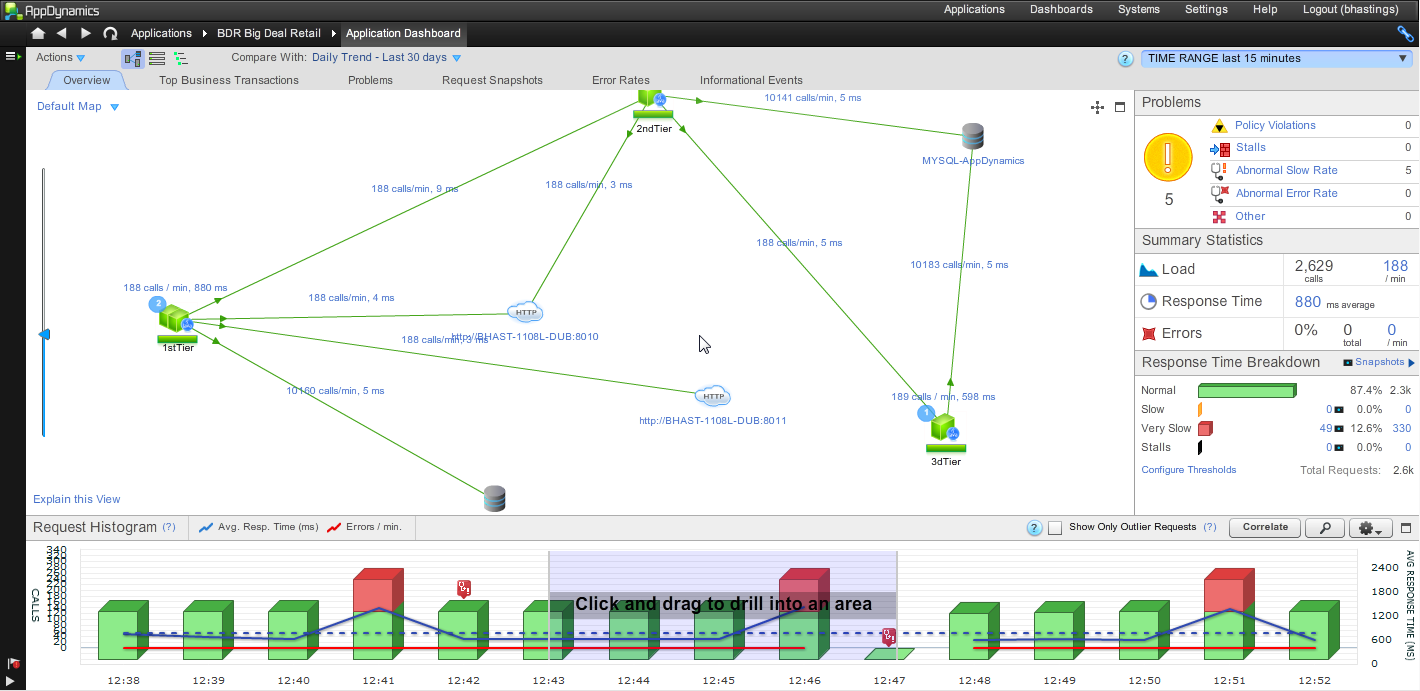


We will begin to test the BDR application so we need to confirm some data within the system.

1. Select the Systems tab and choose AppDynamics agents, confirm that the following agents(see screen below) are present and correct.



1. Under Applications tab select BDR Big Deal Retail/Application Dashboard if you are not seeing any node names, DB names etc. move the slider on the left to increase the view.



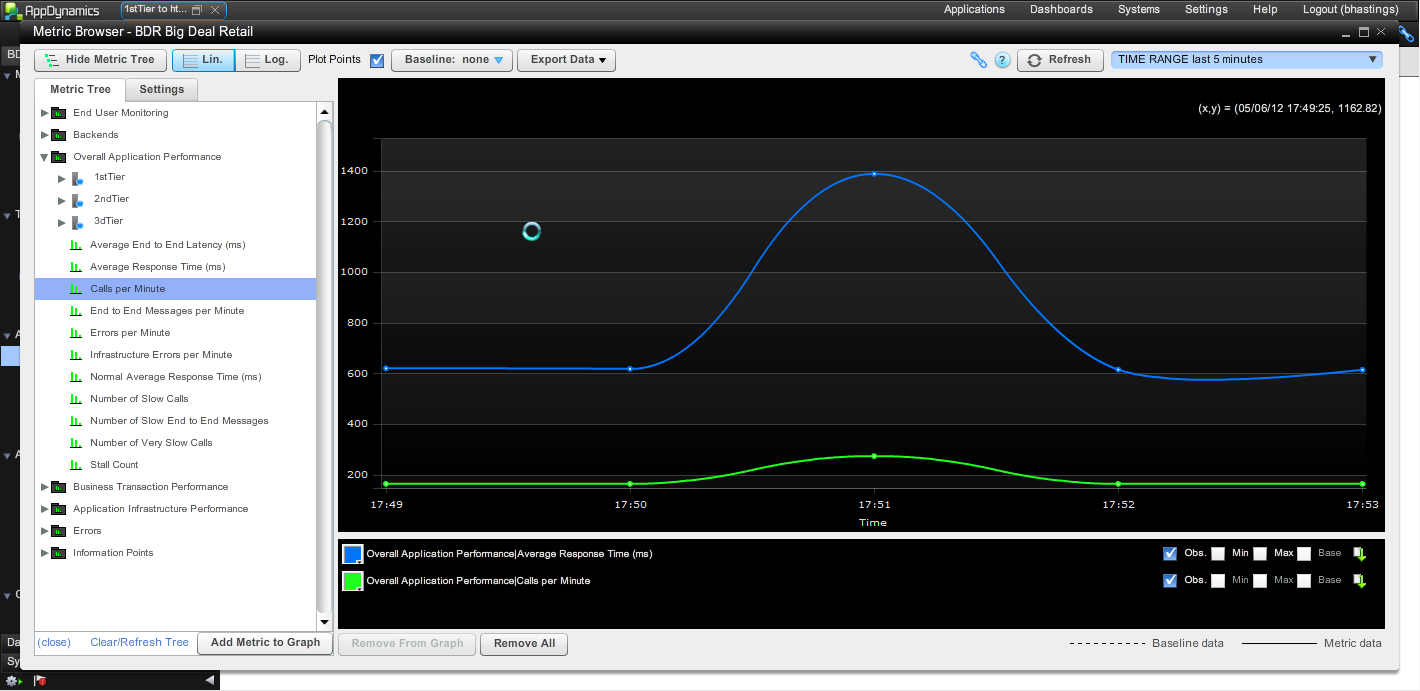
Slider

1. **View the Application dashboard**: Look at the Bar chart and click on any of the red/green bars to get a **summary and response time breakdown.**
2. **Identify the current business transactions** in the BDR application
3. **Identify the Backend** : Record the following Backend metrics
   1. Response time (ms)
   2. Calls
   3. Calls per minute
   4. Errors
4. **Review the policies:** Spend a few minutes noting the policies that are currently set on the BDR application for future use
5. **Overlay Throughput and request metrics:**

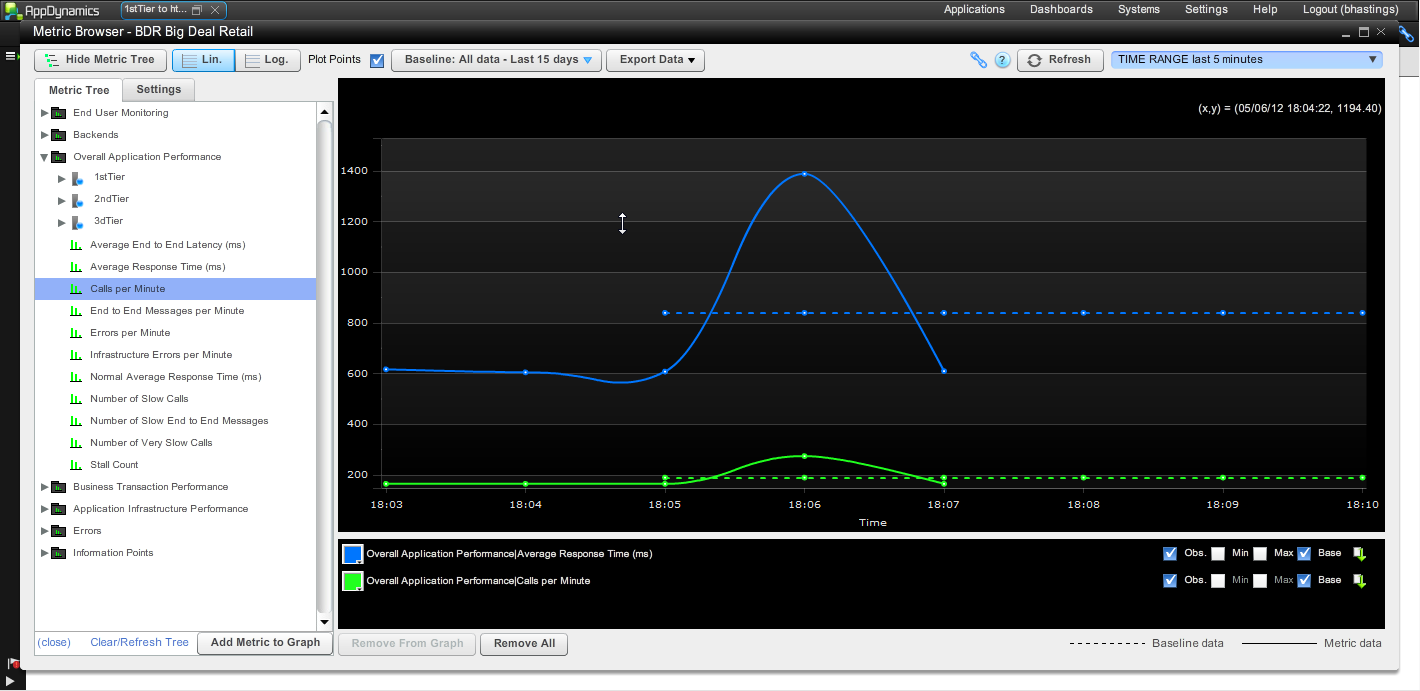
Open the Metric Browser.

Double-click (or drag to the graph) both the Application Performance – Average Response Time and Application Performance – Calls Per Minute metrics.

Graph should show lines for both series see below.



Overlay throughput and requests showing baselines:



**We shall be continuing the BDR POC application in Unit 2 in more detail for now we need a high level view of the application.**

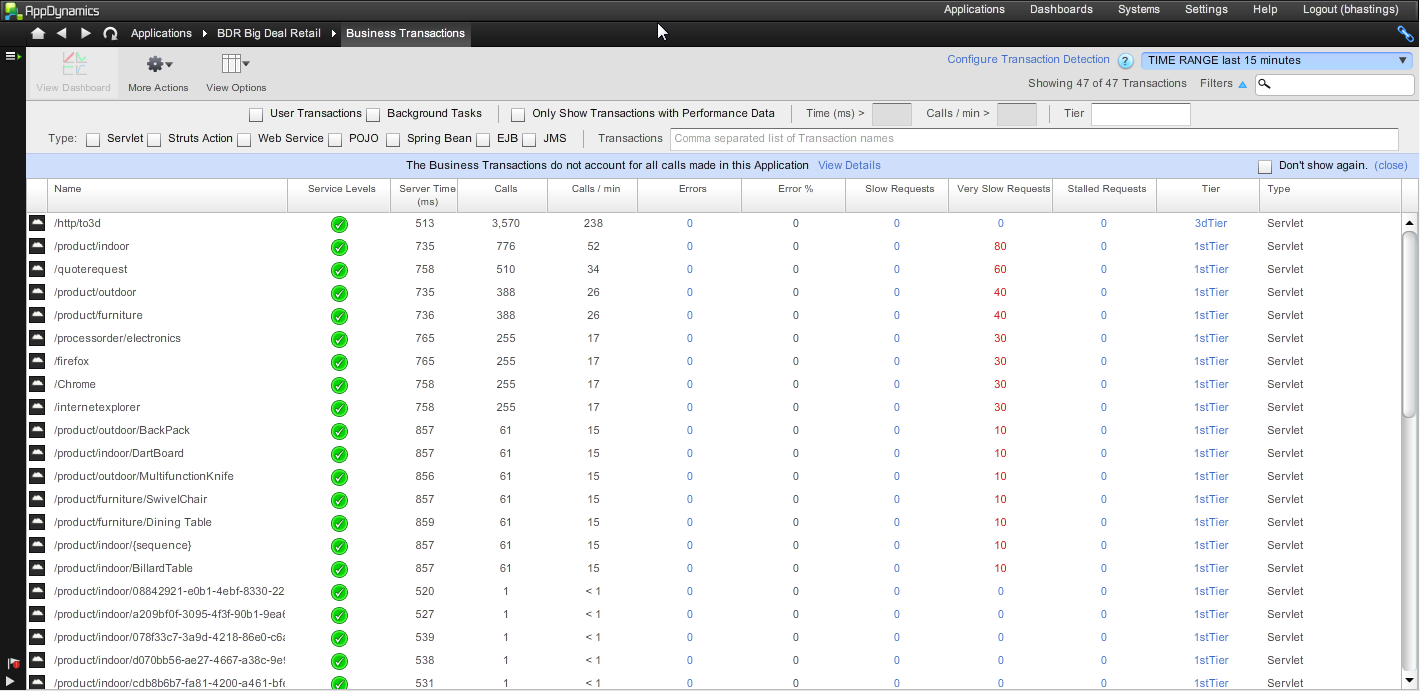
**Unit 2 Labs Monitoring and Diagnostics**

**Lab 2.1 Dealing with Business Transaction Explosion**

The POC is going very well but Chase has noticed a potential for Business Transaction explosion. This will be useful for him to demonstrate the versatility of the product by creating custom match rules to provide more granular monitoring of Business Transactions. Follow the steps below to demonstrate the explosion and correct it.

**Lab 2.1 (a) Create the Business Transaction explosion**

1. Go to Configuration/ Instrumentation/Transaction Detection
2. Select 1stTeir / Servlet/Configure Naming for Tier1. Change from using the first 2 segments to first 3.
3. Save the changes
4. Continue running load through the application.
5. BT Explosion should occur fairly quickly as shown below

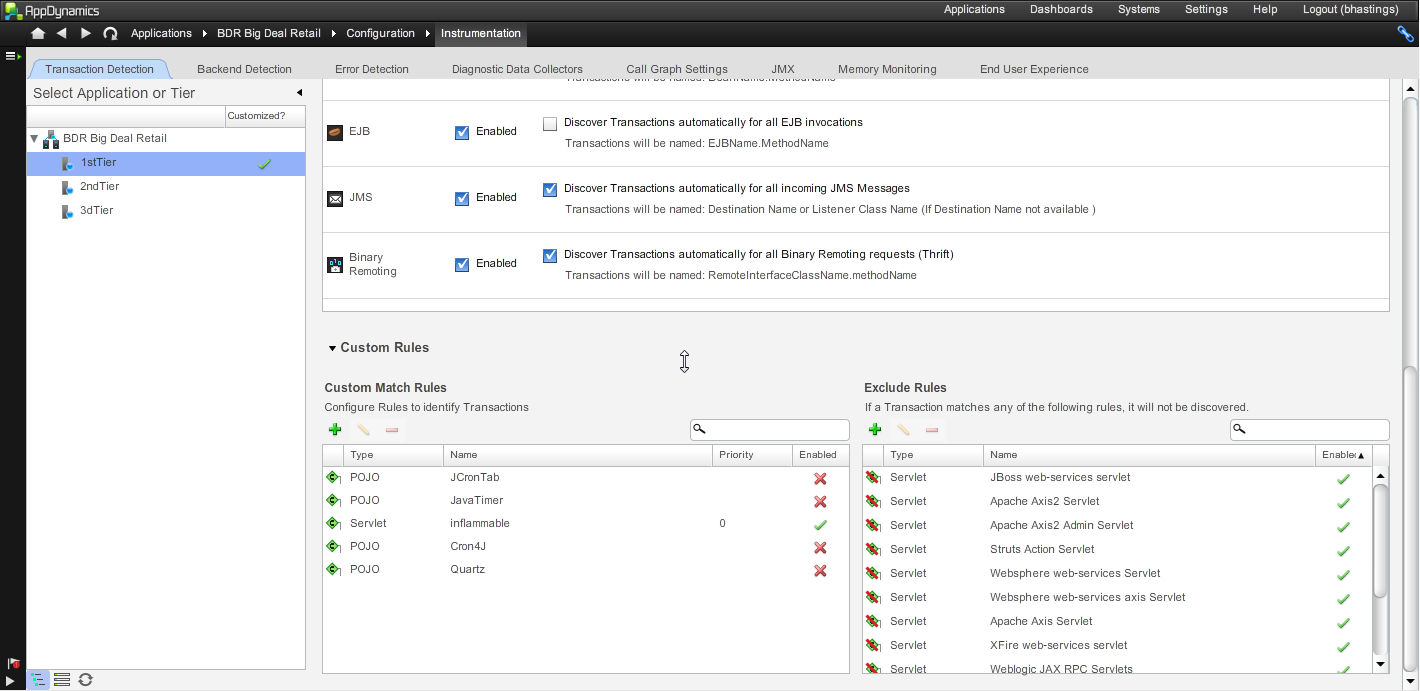


**Lab2.1 (b) Business Transaction Custom Match Rule -- URI**

1. In the 1stTierconfig.xml add an entry as follows:

**<url method="GET" value="/inflammable/indoor/{GUID}"/>**

1. Create a new Servlet match rule for tier 1
2. Specify that the URI should contain “inflammable”
3. Eventually all the URIs containing inflammable should be detected as this new Business Transaction.



Inflammable/indoor transaction is now visible

**Lab2.1 (c) Business Transaction Custom Match Rule -- URI**

1. In the 1stTierconfig.xml add an entry as follows:
2. <url method="GET" value="/product/outdoor/MultifunctionKnife">
3. <request-data>
4. <parameter name="color" value="Purple"/>
5. <parameter name="category" value="Utility"/>
6. </request-data>
7. </url>
8. Create a new Servlet match rule for tier 1
9. Select configure naming
10. **Reset the first part back to 2 from 3 (as set in part(a)**
11. Checkbox New transactions dynamically using part of the request
12. Specify that the Parameter name as color
13. Save
14. Eventually all the URIs containing Purple should be detected as this new Business Transaction.

**Lab 2.2****Creating a Custom Exit Point**

Transactions consist of entry and exit points and can be defined as such, so we can assign an exit point for example to processOrder which would delegate it to appear as a backend as part of the transaction flow. As the final stage of the POC Chase is setting up the processOrder exit point highlighting the effectiveness of this feature. Being able to determine how you see a business transaction is key to representing the system with APM

Follow the steps below to set up such an exit point.

1. In the left navigation panel, click **Configure -> Instrumentation**.

2. Click the **Backend Detection** tab.

3. Select the application or tier for which you are configuring the custom exit point.

4. Scroll down to Custom Exit Points.

5. Click **+**.

6. In the Create Custom Exit Point window, click the **Identification** tab if it is not selected.

7. Enter a name for the exit point. This is the name that identifies the backend.

8. Select the type of backend from the Type drop-down menu or check Use Custom if the type is not listed.

9. Configure the class (com.appdynamics.testappserver.spring.SpringBean4)

with method name (processOrder) that identify the custom exit point.

10. Click **Save**.

**Lab 2.3: Infrastructure monitoring-Creating a new JMX metric**

1. Open the MBean Browser(App infrastructure/1stTierNode1/jmx)
2. Find the MBean we’re interested in (doesn’t exist yet, so let’s just look at the Information under com.appdynamics.testappserver domain)
3. Turn on Live Graph to see it updating itself
4. Click on “Create Metric” to create a new metric for it

**Lab 2.4 Create an Information Point**

The AppDynamics product provides additional features that allow for calculated metrics to be determined and classified as information points. Chase is going to set up an Information Point the tracks the average and total orders placed. Using the instructions below create the Information Point.

1. Navigate to Information Points and click ‘New’ OR right-click on a method in a snapshot.
2. Create IP with the following:
   1. Class Name: com.appdynamics.testappserver.spring.SpringBean4
   2. Method Name: processOrder
3. Define the following Custom Metrics:
   1. Collect data from parameter 1 and use toString. AVG
4. IP will be created that tracks avg. order amount.

**Lab 2.5 : Set up policy on slow transactions**

AppDynamics allows for custom policies to be set up this can assist in the monitoring of the system allowing for full customization in the control of how we should react to the metrics. To demonstrate this point you(Cody) have been asked to take this action

1. Go to Policy Browser
2. Create a BT policy and set up a rule
3. Select custom from policy type, evaluation time 1 mins wait time 1
4. Select next
5. Chose business transaction /product/outdoor.blue
6. Select next
7. Setup condition First box is a name for condition add testoutdoor
8. Select the metic (Average Response Time (ART)) and specific value say 1ms
9. Save
10. This should quickly trigger an incident
11. Inspect the incident to concur with the outcome

**Unit 3 Labs Triage and Trouble Shooting**

Following a successful POC BDR have purchased and installed AppDynamics across their BDR Big Deal Retail platform. They are experiencing a number of stalls, slow transactions, and potential memory issues. Cody Barr is setting up the system to try and troubleshoot the problems. Assume the role of Cody Barr and try and isolate and if possible correct the issues.

**Lab 3.1 Create a Custom Troubleshooting Dashboard**

The Big Deal Retail IT team can see the value in creating a custom dashboard to assist in trouble shooting performance issues. This is a standard practice in APM troubleshooting. However the metrics need to be available to give direction to the underlying problem. In preparation use the Information Point created previously

1. Create graph of average order amount with metric label of same metric.
2. Add calls/min. vs. ART graph.
3. Add status light tied to custom BT policy.
4. Feel free to add other widgets to the dashboard
5. Add any hardware metrics that may be significant
6. Export to PNG for inclusion in a report

**Lab 3.2(a) Start a diagnostic session**

Cody understands that all the metrics required may not be immediately available and so additional monitoring may need to be configured to assist in the trouble shooting. Running a diagnostic session is a good starting point to get an overall health check for business transactions. Sessions collect snapshots both scheduled and on demand so we don’t collect for every request just a sample

1. Select Business Transactions from the BDR Application List.
2. Right click on 1 or more transactions and select “Start Diagnostic Session”
3. Should see request snapshots with a description of “On-Demand Snapshot Collection”

They can be useful for drill down when doing further diagnosis etc.

**Lab 3.2(b) What is the root cause of any slow requests you may see**

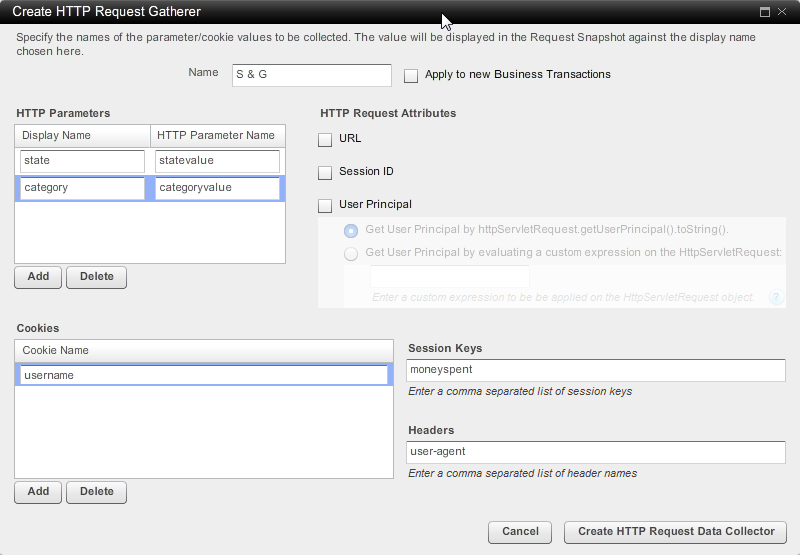
**Cody has established a methodology for drill down which he has outlined below, follow his directions and record the outcomes.**

* Process
  + Troubleshoot 🡪 All Problems
  + Pick a stalled or abnormal slow request (filter if you need to)
  + Click “Troubleshoot”
  + Find the “hot” tier and click on “Drill Down”
  + Review the call graph
  + Identify longest running component

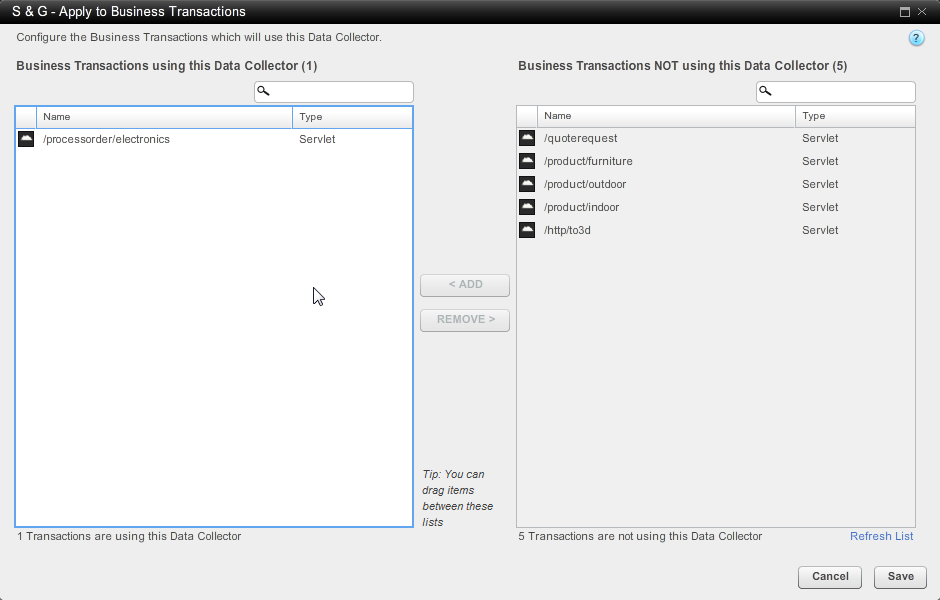
**Lab 3.3: Set up HTTP Data Collectors for Electronics business transaction**

BDR sell a lot of older models in electronics from TVs to utility appliances it’s important for the IT team to be able to identify metric data related orders within this business transaction. They have asked Chase to set this up within the POC. Follow the steps below to achieve this monitoring

1. Navigate to Applications -> BDR->Instrumentation
2. Select ->Diagnostic Data Collectors.
3. Click the Add button
4. Create a new HTTP Request Data Collector called S&G to collect the following:
   1. Two HTTP parameters: state and category
   2. One cookie: username
   3. One session key: moneyspent
   4. One header: user-agent



1. Click **Create http request data collector**. This is then applied to the /processorder/electronics transaction.



1. Snapshots collected after the creation of this collector should contain the specified request data.

**Lab 3.4 Business Transaction Custom Match Rule – session object**

Big Deal Retail needs to track requests from particular browsers, so you are asked to setup three browsers firefox , internetexplorer and chrome as the 3 most used browsers for tracking. This can be done by creating 3 separate transactions identifying each. Shown below is the entry for firefox add this plus the other two entries IE and Chrome

1. In **1stTier1config.xml** file add the following in the appropriate place, see the firefox entry:

<url method="GET" value="/firefox">

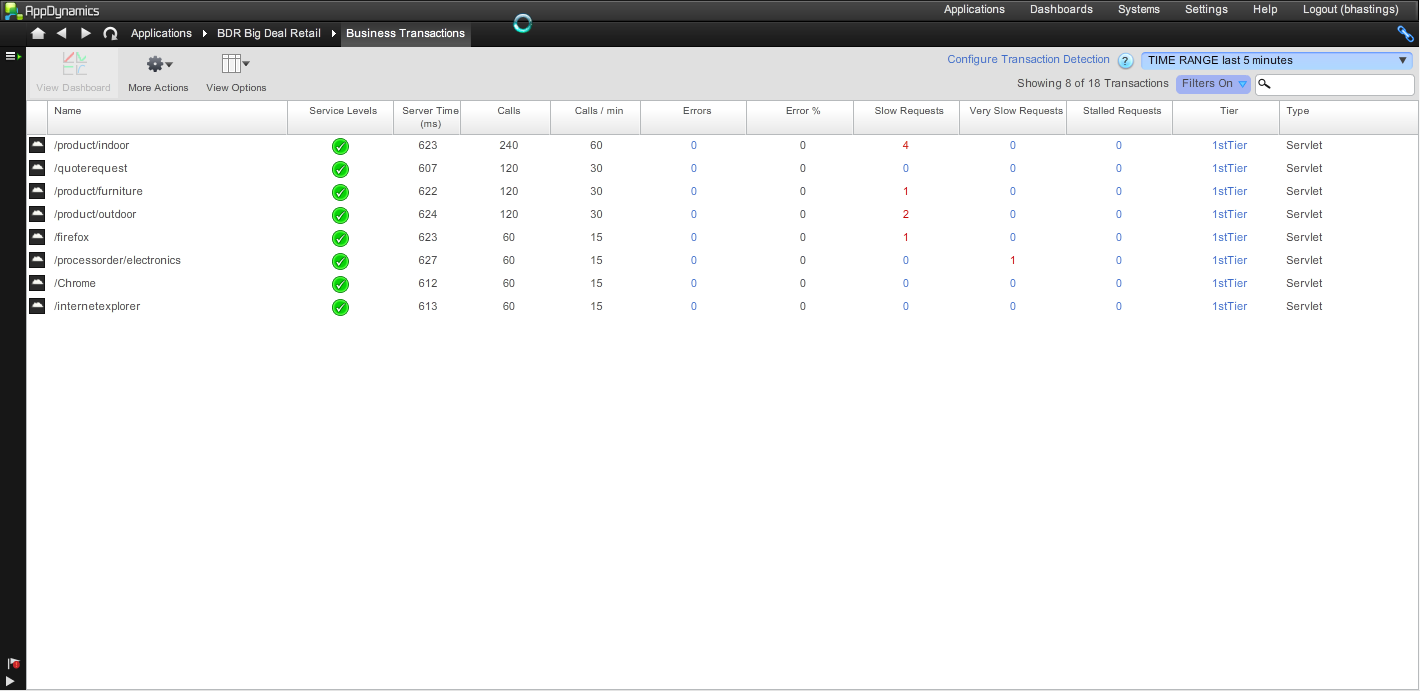
<request-data>

<header name="user-agent" value="firefox"/>

</request-data>

</url>

1. Add the two further custom match configs which contain a user-agent header value of, internet explorer and chrome similar to above.
2. Eventually all three should appear as Business Transactions see below.



**Lab 3.5 Trouble Shooting issues**

**Analyze Business Transaction execution / root cause analysis to find slow code, slow sql, and exceptions**

1. Navigate to Applications--BDR🡪Application dashboard
2. Find a spike in the ART graph (blue). Note the time range and peak value
3. Click on “Request Snapshots”.
4. Select custom from the time range and enter the noted time range or use the peak ART as execution time threshold.
5. Find the tier making SQL calls, drill into snapshot and find slowest statements.

**Lab 3.6: Preparing The New Environment Setup**

**Setup: note in preparation for the final exercise in the trouble shooting unit you will have downloaded a number of files. These files will be used to generate a different environment for the next exercise.**

1. **Make sure the 3 AppserverAgent windows are closed; there is no need to close down the machine agent or controller.**
2. **Make copies of the (1stTierconfig.xml, 2ndTierconfig.xml, 3rdTierconfig.xml) from c:testapp directory to the copies files folder.**
3. **Copy the files from the troubleshoot folder into c:\testapp and overwrite the files that exist there.**
4. **Run the run\_multitiered\_testApp.bat**
5. **There is no need to run the machineagent.bat its running.**

**We are now ready to begin the final exercise of Unit 3**

**Lab 3.6:** **New Environment Performance Issues**

Big Deal Retail has deployed a new version of their application overnight. This contains a number of enhancements and additions to their environment. This deployment has created a number of performance issues which Cody Barr has been notified about from support. He has decided to trouble shoot the issues and implement any additional monitoring to assist in the discovery of the issues. Tom Cash has asked for a detailed report on the issues and the steps employed to solve them

Assume the role of Cody Barr and carryout the trouble shooting on the system identifying, diagnosing the issues you see finding the reason. Record the steps you are taking using the methodology supplied in the presentations. Export any significant screenshots or data required for the report to Tom Cash.

**Lab Unit4:**

An important part of Cody’s team is to manage the appDynamics software in production. This will involve making the most of all the tools the product provides and ensuring that administrative procedures are setup to optimize performance. Cody is doing some in house training with his IT staff and is demonstrating some of the main administration components of the software. The Metric Browser is a useful tool for investigating metrics so Cody wants his team to understand its use.

**Lab 4.1 Explore the Metrics Browser**

* Review the metrics you see
* Compare a couple metrics, such as the average response time, against a baseline
* Look at that response time against 1-2 standard deviations from the baseline

As part of the tools demo scalability analysis is important and can be presented in production, there is a requirement for significant load on the system to generate interesting graphs, however it is useful to understand the possible outcomes.

**Lab 4.2 Scalability analysis**

We don’t have enough load to make these graphs look interesting (you need to generate significant load in order to see performance degrade substantially), but this is a good set of tools for you to explore

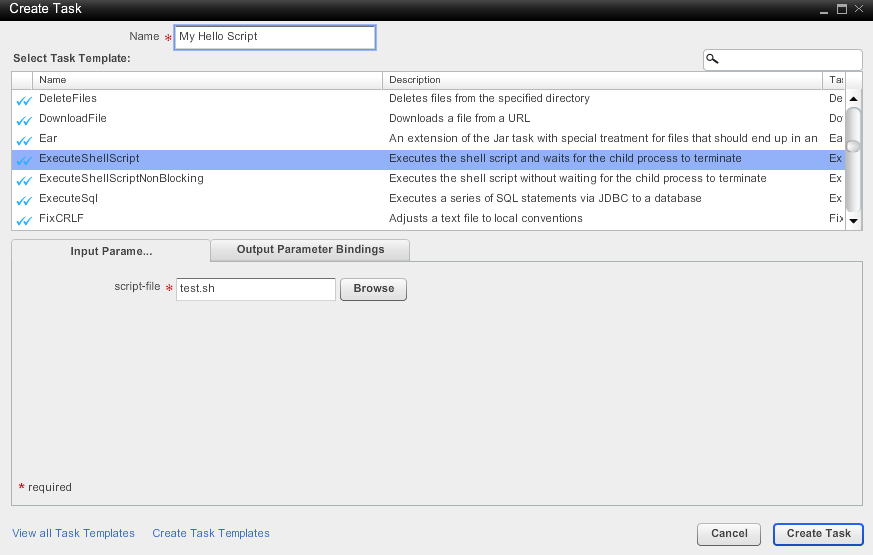
Cody’s team is well versed on script writing and implementing script based admin features. The appdynamics software makes provision for delivering scripts in the form of tasks and work flows. Practice executing scripts via tasks and work flows set out below.

**Lab 4.3(a) : Execute a Shell Script**

Select Automation 🡪 Tasks 🡪 New

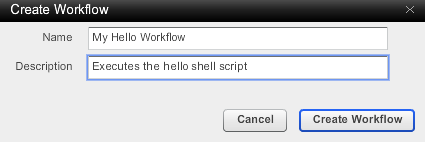
Choose “Execute Shell Script”

Test Script: example(any one will do)

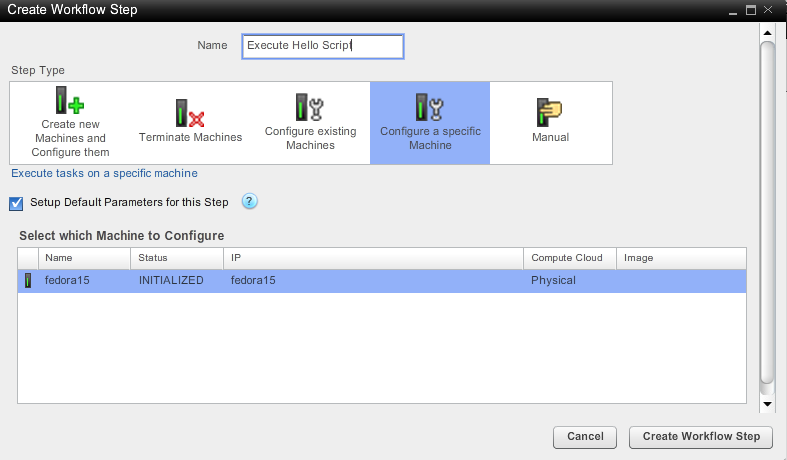


**Lab 4.3(b) Creating a Work Flow**

Select Automate 🡪 Workflows 🡪 New

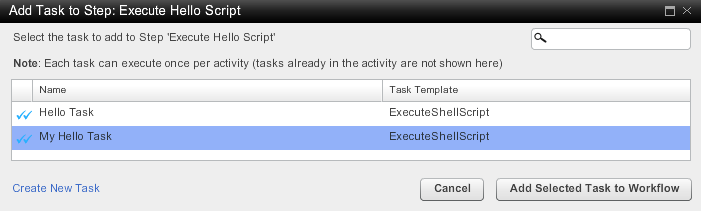


* Add a new step: Configure a Specific Machine and choose our Virtual Machine

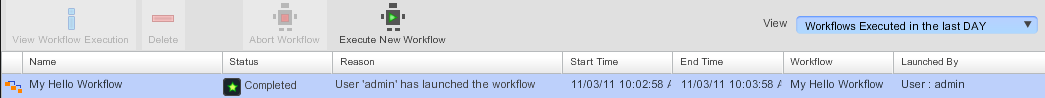


Add a new task to execute the script





**Test the Workflow**



**Trigger the workflow as an action on the policy violation you setup in unit 3.**

**Complete Lab 4.4 Resetting & Disabling Agents**

Following a support call Budd Checkov to Cody Budd has asked him to systematically reset the agents. This will allow the business transaction count to be reset. Follow the directions below

Monitor 🡪 Application Infrastructure 🡪 1stTierNode1: “Agents” tab and apply reset.

**Repeat for 2ndTierNode1 and 3rdTierNode1**

Following the reset you are asked to disable each agent and check by inspection what metrics are still being received.

**Additional Labs**

**Lab Additional: Adding a Custom Metric to a Machine Agent**

Cody’s first step is to setup some custom metrics in the machineAgent.There are two ways to do this. The simplest is to pipe the metrics into the machine agent through an http listener. The more secure way is to create a machine agent monitor that pipes the metrics into the machine agent through stdout. Cody decides to use the more secure method.

Setting up an Http Listener

1. Refer to the machine agent readme.txt.
2. Refer to the hardware metrics monitor embedded in the machine agent at MachineAgent/monitors/HardwareMonitor
3. Create your own monitor in the “monitors” directory giving it a new sub directory of “monitors”.
4. Copy over the monitor.xml file from the HardwareMonitor into the new monitors sub directory and edit the monitor.xml to suite the new monitor.
5. Create OS specific monitor scripts for your new monitor.
6. Start up the machine agent with the http listener enabled:  
   **nohup java -Dmetric.http.listener=true -jar machineagent.jar &**

Appendix A: ACME Online System: