Ex 1 - Tachyon v5.1 - Using - Using Guaranteed State

Exercise Overview:

Using Guaranteed State

Guaranteed State allows us to check our environment against a desired configuration and remediate any devices that do not meet that desired configuration. We use a combination of Policies and Rules to accomplish this with Tachyon. Behind the scenes fragments come into play, if you need to create your own then you should take the Tachyon v5.1 - Using - Creating Instructions and Fragments Using TIMS and the Tachyon v5.1 - Advanced courses.

The Guaranteed State application

The Guaranteed State Administrator and Guaranteed State Viewer roles in Tachyon have been created in Tachyon, if you are not aware of how to do this then you should consider taking the Tachyon v5.1 - Configure and Install course.

Exploring Guaranteed State Application

In this task we will look at the different nodes in the Guaranteed State Application as both the Guaranteed State Administrator and the Guaranteed State Viewer

1ETRNW72

1. Log onto **1ETRNW72** as **1ETRN\Manager1**.
2. Open the Tachyon Portal and launch **Guaranteed State**.
   The Guaranteed State Administrator can also go directly into Guaranteed State https://tachyon.1etrn.local/tachyon/app/#/guaranteedstate, which is especially useful if they have no other role in Tachyon.
3. Navigate to **Overview** and look at the different tiles. We will visit these again later after we have deployed some policies. They will all be blank until we have assigned and deployed our policies.
4. Notice the drop-down at the top defaults to **All Policies**, we can also select the different policies that were imported via the Product Pack Deployment Tool.
5. Navigate to **AdministrationPolicies**. Notice we have policies listed. This was from the import.
6. Select the **Windows Client Health** policy and click the **Assign Mgmt Grp** button on the right-hand side of the screen.
7. On the **Assign Management Groups** popup, start to type **All Dev** in the **Search for Management Groups** field.
8. Click the *All Devices* Management Group when it resolves. Click **Save**
9. Select the **MEMCM Client Health** and assign it to All Devices
10. We have now assigned policies
11. Once you have assigned the Policies to your Management Group you must now click **Deploy**
12. Click **Yes** on the *Are you sure* box
13. Navigate to *Administration - Rule*s node. Notice the rules that are imported when the Product Pack Deployment Tool is run
14. Click **View Details** on the *Check free disk space (Check)*. In the pop-up notice this is a *Check Rule* that will query the device to see that the *system disk has at least 5 Gb of free space*. Notice the **Trigger** – this query will run if the registry values in HKLM\Software\1E\Tachyon\Triggers\CheckDisk changes. Notice the **Pre-Condition** – it will only run on Windows devices
15. *View Details* on some of the rules to get an understanding of what they do
    We will log into Guaranteed State as our account that is only a Guaranteed State Viewer, once we have some data to see the differences in the access permissions.

Viewing the Results of a Policy

Now that we have our policies assigned to a management group and deployed, we can look at the results of that policy.

1ETRNW72

16. Still logged in as **1ETRN\Manager1**
17. Navigate to *Overview*
18. Notice the top drop-down on the far right – We have *All Policies* Selected. We could select a specific policy to view the results for that policy only. Leave it at all policies for now
19. On the top row of tiles there is a Tile – *Device State* – Hover over the  icon. Read the information about this tile
20. Hover over each slice of the pie chart to view the actual number of devices in each state. We may have some devices in Unknown state until they finish evaluating the policies that we have just deployed

21. Click on the slice of the pie for Non-compliant. Notice how it builds our filter for us and takes us to the devices report for the ones that are State Non-compliant
22. Click on View History. Scroll down to see which check(s) failed. Click close
23. Go back to Overview. Look at the next Tile – *Device State per Criticality Level*. Hover over the icon to read the information about the tile. This tile shows a bar chart of the state of our devices but based on the Criticality of the Device that we have assigned
    Device criticality is set in the *Explorer* Application via an Instruction. Tachyon allows us to assign Criticality Levels to devices – much the same way that Microsoft assigns classifications to security update to set the importance level of that update. You might label your Domain Controllers as Critical while a File Server might be set as Low. We can use this to report on status in Guaranteed State. In the Explorer Application we can use this attribute to define coverage for an instruction.
24. The *Last Seen* tile shows the time the devices last checked in with the Tachyon Server. All seven of our devices should show in the current column. If they do not, drill into the chart and make sure the device is powered on. When you click on one of the bars notice that it creates a filter for you and takes you to the results of the filter
25. Click the **back** button or navigate back to *Overview*
26. Look at the *Rule Effectiveness* tile which shows rules that are being used in the environment. Hover over the icon and read the details
    This is a very good tile to use in production. When we have rules or policies in our environment (think of Group Policy or Config Man Compliance Settings) it is nearly impossible to determine if specific policies or baselines are still needed so after time you end up with hundreds, if not thousands of them and no way to tell if you really need them. This tile will show you the polices that are disabled and ones that are not deployed. That is good information to have but the real value here are the Ineffective category. These are the policies that have not changed their compliance position changed in the last 30 days. No devices that are assigned that policy have had any movement either they are still Compliant or still Non-compliant. The policy may not be needed anymore or might not be remediating the issue correctly. The last category – Effective shows all the policies that have had devices that have changed their position in the last 30 days – so they either moved from Compliant to Non-compliant or from Non-compliant to Compliant.
27. Look at the two tiles – *Rule Status* and *Rule Remediations Last 7 Days*
28. Navigate to Administration – Polices. Notice the number of rules in each policy. Guaranteed State calls the device Non-compliant for that policy if any of the rules are not compliant
29. Navigate to Administration – Rules and scroll to find where the Rule type is Fix, and notice the Red in the enabled field. The Tachyon Product Pack Deployment Tool imports in the Fix rules but makes them disabled. Only the check rules are enabled. This ensures that changes to devices will not occur if the environment needs to follow a change control process. We will not enable the fix rules for Nomad in our lab as we do not have Nomad installed. We will though enable the other rules and if you have Nomad in your environment you will want to enable them in your environment if you want the Nomad items corrected.
30. **Select** the disabled rules that do not include the word Nomad (you may need to scroll to see them all), Once selected the Enable Selection button becomes enabled **click** this.
31. **Click** Yes on the Enable Selection pop up once you have read the warning message
32. Now that rules are enabled they need to be deployed, navigate to Policies and note the message that Pending policy changes to be deployed. **Cli ck** Deploy and **click** Yes
33. Navigate to Reports – Policies. Here you can see the details on the devices and how many are Compliant or Non-compliant
    Notice that we did not assign or deploy the Nomad Client Health policy as we do not have Nomad installed in this lab

Working with Policies, Rules, and Triggers

Policies that contain one or more rules are assigned to Management Groups and deployed. There are two primary types of rules – Check Rules and Fix Rules. Check Rules query the device for a specific state and Fix Rules are used to remediate that device based on the state an Administrator wants to maintain. In this task we will build a Registry Policy. We see applications that need specific settings to function properly. This policy is an example of how to monitor applications to make sure they have the proper configurations or even change the configurations when necessary.

Create A Registry Rule

1ETRNW72

34. In *Guaranteed State*, navigate to *Administration – Rules*
35. Click **New Rule** in the far right. In the *Name* field type in **My Application Registry**
36. In the *Description* window type in **Rule to ensure the My Application Registry is set to 7**
37. We will not set a Pre-Condition check on this rule
    A Precondition check would be appropriate for this one in the real world as we could set this to only run if the application in question was installed.
38. Look at the Precondition check options that are available but leave without setting one since in our lab we do not have an application install to go with our fake registry key
39. Scroll to **Triggers**
40. Start typing when a regi*s*. Select **When a Registry Key Changes** from the options shown
    Triggers tell the Tachyon Agent when to run the check rule – so when to perform that query. Notice that we can do this check on a timed interval, but we can also do this check each time something changes on the devices (Service state changes, file changes, registry key changes, event log entry, process start up, security event, etc). The interval check should rarely be used as this is just an arbitrary cycle and will use processing power on the device to run the query even if it is not needed. The value of Guaranteed State is the ability to look for the specific change and only run your rule at that time – if my policy says a service should be started it is a waste of the device to run that check every hour when I could set my trigger to only run if the state of that specific service changes.
41. We need to add our Registry Key for this task
42. Leave *Guaranteed State* as it is and Click **Start button** – in the *Search* Type in **Regedit**
43. Open *Regedit* and Navigate to **HKEY_Local_Machine\Software**
44. Right Click on **Software** and choose **New - Key**. Name the key **MyApplication**
45. Right Click on **MyApplication** and Create a **New DWord Value**
46. Rename *New Value #1* to **MySettings**. Notice our value is set to 0 – we will leave it like that and let *Guaranteed State* fix it for us
47. Minimize Regedit. Back in *Guaranteed State* – In the *Hive* box select **HKLM** (if not already selected)
48. In the *Subkey* box type in **Software\MyApplication**
49. In the *Include Subkeys* box choose **true** (if not already selected)

50. This will tell the Tachyon Agent to monitor this Registry Key for changes – if it changes it will then run our rule to check to make sure the device is still configured properly
51. Navigate to the **Check** section – and in the *Check* textbox start typing Registry and select Check that registry key Hive\Subkey\Name has ValueType value of "Value"
52. In the *Hive* box select **HKLM**
53. In the *Subkey* box type in **Software\MyApplication**
54. In the *Name* box type in **MySettings**
55. In the *Value* box type in **7**
56. In the *Value Type* box select **Reg_DWord**
57. Click **Save** in the Rules page
58. Navigate to *Administration – Policies*. Click on **New Policy** in the far right
59. In the *New Policy* page, *Name* field type in **My Application Registry**
60. In the *Description* field type in **My Application Registry**
61. In the *All Rules* pane select our **MyApplication Registry** rule click the **>>** to move it over to *Assigned Rules* pane
62. Click **Save**. Notice we have a pop-up reminding us that our Policy has not been deployed
63. We will assign and deploy that now. Select our **My Application Registry** policy click **Assign Mgmt Grp**
64. Start typing *Win* and once suggested select the ***All Win7 Lab Workstations*** Management Group
    We could repeat this to select multiple management groups
65. Click **Save**
66. Click **Deploy**. Click **Yes** on the *Are you sure* pop-up
67. Navigate back to the *Overview* Page. Click **Refresh**
68. Select the *My Application Registry* policy from the drop-down to view our tiles for only that policy
69. Notice our device state is Non-compliant for all three devices
70. Drill into the Device Status to see the individual devices
71. Click View History on each one to see the results for each machine
72. This is the expected result because we deployed to our three Windows 7 devices – two of them do not have the registry key and 1ETRNW72 has the key but with a 0 value

Watch Guaranteed State in Action

In this task we are going to work with our MyApplication Registry setting again but this time we are going to create the value if it does not exist. If it is set incorrectly, set it correctly. Again, this could help us in production with the applications in our environment that need specific settings to function correctly, this Policy will ensure that our application will function correctly even if something modifies the setting.

1ETRNW72

73. In *Guaranteed State*, navigate to *Administration - Rules*
74. Select the *My Application Registry* Rule click on **Edit**
75. Note the *Name* field is read only
76. Note the *Description* field is updateable, do not add anything
77. Leave the *Precondition* field alone
78. Leave the *Triggers* field as it was
    In addition to running based on the trigger that is configured – in our case when our MyApplication key changes - rules will also run when initially deployed to devices.
79. Leave the Check as was
80. In the *Fix (optional)* section start typing registry and **Select** the Set registry key Hive\Subkey\Name to ValueType value of "Value"
81. In *Hive* select **HKLM** (if not already selected)
82. In the *Subkey* field type in **Software\MyApplication**
83. In the *name* field type in **MySettings**
84. In the *Value* field type in **7**
85. In the *ValueType* field select **Reg_DWord**
86. Click **Save**
87. Navigate to *Administration – Policies* click on **Deploy** at the top. Click **Yes** on the *pop-up*
88. Navigate back to *Overview* and Click **Refresh**
89. Once the devices have checked in you should see it change to all Green in device state with My Application Registry selected for the overview
90. Open regedit to check that the value has changed. You may need to refresh if you left it minimized. Log into the other Windows 7 devices to check them also
91. When you have completed checking the other machines – return to **1ETRNW72** and change the registry value to **20**. Right click on **MySettings** an d choose **Modify** – type in **20.** Click **ok**
92. Hit **refresh** and you will see Tachyon almost instantly change it back to 7
93. Go back into *Guaranteed State*
94. Navigate to *Overview – Policy Rule Effectiveness* – Click on the *Effective* Gear Icon
95. It will take you to the *Rules* page with our **MyApplication Registry** rule filtered
96. Click on the **MyApplication Registry** Rule. It will take you to our *Devices* page that is filtered for our deployed rule. Click on **View History** for **1ET RNW72**. Notice the status entries as you changed it to 20 Hex which is 32 Decimal

Create a Service Rule

In this task we are going to create a policy that checks the state of the Remote Registry Service and starts the service if it is stopped. We are going to assign this to our All Devices management group and deploy it but use a pre-condition to make it only run on Windows 10.

1ETRNW72

97. In *Guaranteed State*, navigate to *Administration – Rules*
98. Click the *New Rule* button
99. In the *Name* field type in **RemoteRegistry Service**
100. In the *Description* type in **RemoteRegistry Service Started**

101. In the *Precondition* field start typing Operating system and select Run if operating system is OsText
102. In the OSText field that appears select **Windows 10**
103. In the *Triggers* field start typing When the state and select **When the State of the named Windows Service changes**
104. In the S*erviceName* field type in **RemoteRegistry**
     Note that we could select multiple triggers by click in the + sign and adding additional triggers.
105. In the *Check* field start typing Check the Service and select **Check the ServiceName service is in State state**
106. In the *ServiceName* field type in **RemoteRegistry**
     Make sure you type in RemoteRegistry for the service name without a space or your rule will fail.
107. In the *State* field select **Running** (if not already selected)
108. In the *Fix* field start typing Service and choose **Request service "serviceName" to action**
109. In the *ServiceName* field type in **RemoteRegistry**
110. In the *Action* field select **Start**
111. Click **Save**
112. Navigate to *Administration – Policies* and Create a Remote Registry Policy that contains our Remote Registry Rule. **Save** the policy
113. Assign to our All Devices Management Group and then **deploy**
114. Navigate to *Overview* – select our **Remote Registry** Policy
115. Hit *refresh* until you begin to see results
116. Look at the state of the devices
     Note that both Windows 10 devices fail initially but the fix rule starts the service, all other devices show as Not applicable, this is because we set a Pre condition of Windows 10

1ETRNW102

117. Open *file explorer* and look at **c:\ProgramData\1E\Client\1E.Client.log**. Notice how the policy is downloaded, checked against our certificates, and then processed by the agent, but failed and then started the RemoteRegistry

Guaranteed State as a Viewer

In this task we will look at Guaranteed State as a Viewer

1ETRNW101

118. Log into **1ETRNW101** as **1ETRN\User**
119. Launch the *Tachyon Portal*
120. Switch App to *Guaranteed State*
121. Navigate to *Overview*. Look at the tiles for each Policy that we have configured
122. Drill into the tiles
123. Navigate to *Administration – Rules*. Notice that this user can see all the rules but has no ability to create or deploy them
124. Notice that within *Rules* you can drill into the **View Details**
     In this lab we worked with Rules- both *Check* rules that just run a query for device state, and *Fix* rules that act to remediate the device if the check rule fails. We also looked at triggers, these are what tell the Tachyon Agent when to run a specific rule. We used Precondition checks to target our rules at only applicable devices (i.e. a Policy to fix a registry key for a specific application should only be needed to run on devices that have that application installed). We also looked at the Guaranteed State application as our user that is only a viewer.

Lab Summary

In this lab we looked at the Guaranteed State Application. We explored the application and then deployed polices to devices and observed the results when a device did not meet a condition. We then saw how the Guaranteed State Application reported on our device status in near real-time.