

Implementing IT Service Intelligence Lab Exercises

Welcome to the Implementing Splunk IT Service Intelligence class. There is one lab exercise for each module of the course. Your instructor will assign you a server and credentials to log on to your server to do your lab work.

Your lab server is a cloud instance running on Amazon Web Services.

You will need a web browser to connect to the server's Splunk web interface.

Lab Typographical Convention

<your student ID#> indicates you should replace this with your student number. When you save your work, you will often be instructed to append your student number to the name of the saved item so the instructor can grade your work.

Lab Credentials

Your instructor will provide the information needed to fill in this table at the beginning of your first lab exercise. You will need this information to log on to the Splunk server running ITSI.

For lab exercises Using-1 through Using-4, you will share one server while working in "user" mode to learn how ITSI works.

In lab exercise Implementing-1 to the end, you will work on your own on a private server to configure a new ITSI system.

Get the information from your instructor for the shared "DEMO" server and enter it here:

| | |
|------------------------|--------------------|
| DEMO Server Address | |
| Splunk Analyst UID/PWD | analyst__ / |



Lab Exercise Using-1: ITSI User Interface

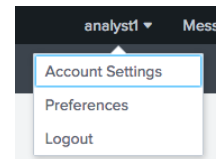
Description

In this exercise, you log in to a demo Service Intelligence server and explore its user interface.

Steps

Task 1: Log on and explore the Service Analyzer and related views.

1. Use your web browser to connect to your ITSI server as an ITSI analyst, using the connection and credential information provided by your instructor. Your Splunk user account is a member of the **itsoa_analyst** role.
2. After you log in, you see the Service Analyzer because ITSI is the default app for your user account.
3. First, set up your user account: click your user name (**analyst1**) and select **Account Settings**. (This is used for grading your lab exercises, so you will receive credit for completing the course.)
4. Set the **Full Name** field to your student ID - first and last name.
For example: 5 - Mary Roberts
5. Click **Save**.
6. Click the **splunk>enterprise** logo to return to the **Service Analyzer** page.
In the Service Analyzer, note that three services are configured: Online Sales, Storefront Web Farm and Support Web Farm. You see all their KPIs, in descending level of alert severity: critical (if any) first, then high, etc.
7. Click the tree view icon  to see the structure of the dependencies between services.
8. Click **Storefront Web Farm**. A side panel opens with details about its KPIs and Critical and High Episodes (if any).
Notice the Storefront Web Farm circle is now outlined and centered in the tree diagram. At the right, notice that its KPIs are displayed.
9. Click **Open all in Deep Dive** to view the listed KPIs in a deep dive.
A new browser tab opens. Mouse over the deep dives. You'll explore deep dives later.
10. Return to the **Service Analyzer** browser tab.
Make sure the Support Web Farm pane is open on the right.
11. Look at the bottom right at Event Analytics pane ("**Critical and High Episodes**" is bold).
12. Click **View All** to view any critical and high episodes.
A new browser tab opens an Alerts and Episodes view.
13. Click an episode from the list at the bottom to see more information and options.
From here, you would click **Acknowledge** an Episode to take ownership and begin working it.
DO NOT acknowledge any Episode at this time.
Only admins can define services, but you can look at how they were defined.
14. In the tree diagram, by default, the selected service is centered. Click  to zoom and center the entire tree.



15. In the KPIs list, click the KPI called **CPU Utilization: %**

An additional panel opens displaying the entities associated with the KPI, which, in this case, are hosts.

16. Click the **Storage Free Space: %** KPI.

Notice it contains the same hosts, but the severity levels for each are specific to this KPI.

17. Click **www1**.

A new browser tab opens. The **Event Data Search** tab shows events associated with the entity. The **Analysis** tab shows metrics associated with the entity's Entity Type. The right side has the Entity Information pane which includes the Services and KPIs associated with the Entity. There may also be a table of any associated notable events.

18. To view specific information about how www1 is performing, from the *Entity Information* panel, click **OS Host Details**.

You will explore **OS Host Details** more later.

There are several tabs you can explore to see what information is available.

19. Close the two most recent browser tabs and return to the **Default Analyzer** browser tab.

Task 2: Filter Service Analyzer views and save new Service Analyzers.

20. Click **Save as...** to save this Service Analyzer, name it **All Services** and keep it private. In the Description field, indicate this saved analyzer is your backup and click **Create**.

21. Click **Service Analyzer > Analyzers**.

Your new Service Analyzer is now listed.

22. Return to the **Default Analyzer**.

Note that it has both a Save and Save as... button.


23. Use the Filter Services field to select only the Online Sales service.

Note that now only the Online Sales service and its KPIs are displayed.


24. Click the Show service dependencies check box.

Notice that the Service Analyzer expanded its display to also include any services on which Online Sales depends and all KPIs associated with those services.

25. Click **Save as...** and name your analyzer <your student ID#> - Online Sales and Dependencies, set Permissions to Shared in App and click Create.

26. Make other changes to your # - Online Sales and Dependencies service analyzer, such as altering the tile size, time range, maximum number of KPIs (settings ) , and selecting Tree view.

27. Click **Save** to make these changes permanent.

28. Use the Standard View  link to toggle between standard and full screen view. (You may want to save some of your Service Analyzers in full screen because it is useful for operations center displays.)

Task 3: Examine potential issues based on severity by drilling down from KPIs to a filtered deep dive.

29. Display your analyzer in **Tile** view.
30. Hover over a KPI tile that is yellow or worse.
Notice a checkmark appears in the tile's upper right corner.
31. Hover and click the checkmarks for any KPIs that are yellow or worse.
32. Click the **Drilldown to Deep Dive** link. A deep dive opens comprised only of lanes for the KPIs you selected. You can see the KPIs changing over time. You'll explore these in detail later.

Task 4: Predict a Service Health Score and explore potential root causes.

33. On the main ITSI menu, click **Alerts and Episodes**. View a few episodes in the table.
34. Click **Dashboards > Predictive Analytics**.
35. Click the **Service** dropdown menu and select **Online Sales**.
36. Under the **Model** dropdown menu, select **LinearRegression**.
37. Click **Cause Analysis**.
Notice you can see the top 5 contributing KPIs and can open them in a Deep Dive for closer examination.

Optional Task: Examine the data underlying IT Service Intelligence

38. Select **Search**.
A Splunk search window appears.
39. Click the **Data Summary** button. Examine the list of hosts, sources and sourcetypes to get an idea of the types of data being indexed. The www* web servers will be the main hosts you work with, along with the access_combined sourcetype, from the web server logs.
40. Run the following search for the **last 60 minutes**:

```
index = itsi_summary
```

This index stores KPI search results. Each of these events represents one KPI at a point in time, depending on the KPI schedule. There are also service health events for each service. Some of these KPIs are also broken out by entity. Examine some of the more relevant fields: alert_level, alert_severity, entity_title, and kpi.
41. Run the following search for the **last 60 minutes**:

```
index = itsi_tracked_alerts
```

This index stores notable events. Examine some of the relevant fields including source (the correlation search name), alert_level, alert_value, entity_title, kpi, owner, status and urgency.

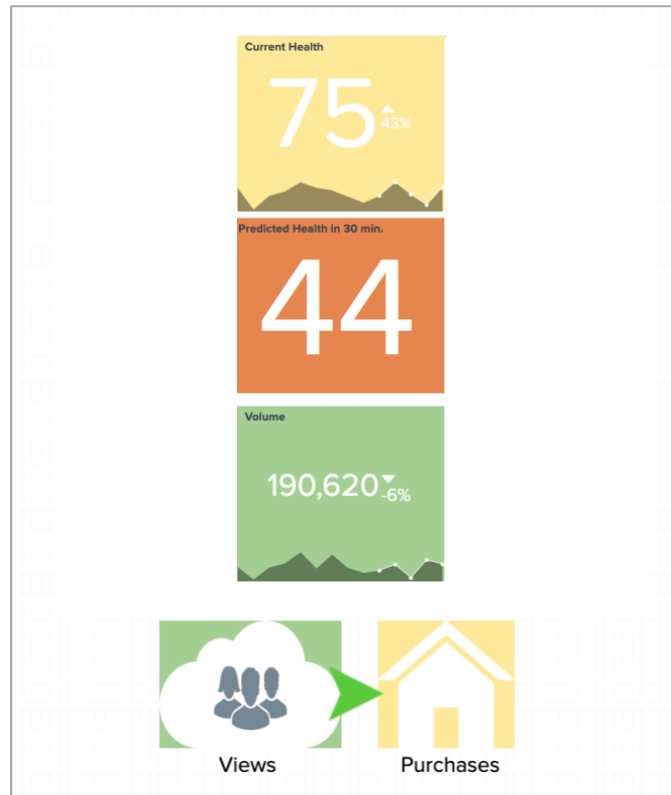
End of Lab Exercise Using-1

Lab Exercise Using-2: Implementing Glass Tables

Description

In this lab exercise, you'll create a glass table to visualize the status of the Online Sales service. The online sales operations team wants to see something like the image below. (This image shows *all* the tasks included in this lab exercise, including the optional tasks. If you choose not to do the optional tasks, some items below will not be present in your glass table).






Use Google Chrome to do Lab Exercise 2.



All of the graphics are available in the glass table editor icon set except the arrow.gif file for the optional exercise, which is provided by your instructor.

| Glass Table Icon Title | KPI / Service Name(s) in ITSI | Notes |
|-----------------------------|------------------------------------|-------------------------------------|
| Current Health | Online Sales ServiceHealthScore | |
| Predicted Health in 30 min. | Ad hoc search | From Predictive Analytics Dashboard |
| Volume | Volume | Volume KPI |
| Views | Views | Views KPI |
| Purchases | Purchases | Purchases KPI |

Task 1: Create glass table with widgets to monitor Online Sales.


1. In the ITSI menu bar, click **Glass Tables** and click **Create Glass Table**.
2. Name it **<your student ID#> - Online Sales**, click **Shared in App**, and click **Create**.
3. In the Actions column for your new glass table, click **Edit > Edit** to open it in edit view.
4. Click the Data icon  and, at the right, click **Online Sales > ServiceHealthScore** then, on the canvas, select its widget, then click the Configuration icon .
 - a. At the right, for Visualization Type, choose Single Value.
 - b. For Title, enter **Current Health**.
 - c. Under Major Value & Trend settings, set Unit Position to After, and Trend Display as Percent
 - d. In the **Sparkline** section, select **Below**.
 - e. Resize the widget until the entire title is visible.
 - f. Under Coloring, set the Dynamic Elements to Background.
 - g. Click the color swatch next to **Static Major Value**, and choose the white swatch.
5. Click the **Save** icon .
6. Click the Data icon  and, at the right, click **Online Sales > Volume** then, on the canvas, select its widget and drag it to its correct location and resize it.
 - a. Under **Visualization Options**, enter the Title as *Volume*
 - b. Under **Major Value & Trend** settings, set **Unit Position** to **After**, and **Trend Display** as **Percent**
 - c. In the **Sparkline** section, select **Below**.
 - d. Select **Trend Display** as **Percent**.
 - e. Under **Coloring**, set the **Dynamic Elements** to **Background**.
 - f. Click the color swatch next to **Static Major Value**, and choose the white swatch.
7. Click the **Save** icon .

Task 2: Add a drilldown.



You can add a Drilldown to create navigation links to saved views, such as a service analyzer, dashboard, or deep dive.

8. Open a new ITSI browser tab.
9. Select **Service Analyzer > Default Analyzer** and click the Online Sales tile check box. Click the **Drilldown to Deep Dive** link at the upper right.
10. Copy the URL for the deep dive that opens and close the browser tab.
11. On your glass table canvas, click the **Current Health** widget and click **+ Add Drilldown** (scroll down to bottom of the Configuration panel).
12. From the **On Click** dropdown, select **Link to custom URL** and paste the URL you copied.
13. Select the **Open in new tab** check box and click **Save** (lower right).
14. Save the glass table.
15. At the upper right above the canvas, click the **View** button to preview your glass table to test the drilldown.
16. Click the large number in the **Current Health** widget. The deep dive opens in a new browser tab. Close that browser tab.


Task 3: Add icons and configure them to act as widgets.

17. Click **Edit**, then click  to add the icons: *home*, then *cloud*, then *group*.
18. On the canvas, click to select the group icon. At the right under Configuration and under **Coloring**, click the **Static Icon** box and select grey, then resize the icon so it will fit within the cloud icon.
19. Resize all the icons to match the existing widgets, as illustrated in the sample picture.
You may have to move the layers to overlay the icons to match the example.
20. Drag each icon to its location and **Save** the glass table.
21. Click the cloud icon.
22. At the right under Data Configurations, click **+ Setup Primary Data Source > Online Sales > Views** and the background of the cloud changes color based on the KPI thresholds.
23. Click the home icon.
24. At the right under Data Configurations, click **+ Setup Primary Data Source > Online Sales > Purchases** and the background of the home changes color based on the KPI thresholds.
25. Add and position the text labels under the cloud and home widgets.
Use the text button from the toolbar to create the words “Views” and “Purchases”, then move them under the appropriate graphics.
26. Save the glass table.


Task 4: Add a widget for an ad hoc search from the Predictive Analytics dashboard with a drilldown.

27. In a new browser tab, navigate to: **Dashboards > Predictive Analytics**
28. Click the **Service** dropdown and select **Online Sales**.
29. Under the **Model** dropdown, select **LinearRegression**.
30. Click the  icon.
31. Copy the search string and close the browser tab but leave the Predictive Analytics tab open.
32. Return to editing the glass table browser tab.
33. Click the Data icon  and, at the right, click **+ Create Ad hoc Search**.
34. For **Data Source Name**, type: **PredictedOnlineSalesHealth**
35. In the **Search** field, paste the search string and click **Save**.
36. After the name of the new search appears under + Create Ad hoc Search, click it.
37. On the canvas, click the new widget. In the **Title** field, type: **Predicted Health in 30 min**
38. Set the **Trend Display** and **Sparkline** to Off.
39. Under **Coloring**, set the **Dynamic Elements** to **Background**.
40. Click the color swatch next to **Static Major Value**, and choose the white swatch.
41. Save the glass table.
42. Return to the Predictive Analytics dashboard browser tab, and copy the URL of the browser tab and close the browser tab.
43. Return to the Glass Table editor tab. Select the **Predicted Health in 30 min** widget and click **+ Add Drilldown** at the lower right.
44. In the **On Click** dropdown menu, select **Link to custom URL** and paste the URL you copied.
45. Select the **Open in new tab** check box and click **Save**.
46. Move the **Predicted Health in 30 min** widget to its proper location.
47. Save the glass table.
48. Click **View** to preview your glass table to test the drilldown.
49. Click the **Predicted Health in 30 min** widget. The Predictive Analytics dashboard opens in a new browser tab. Close that browser tab.

(Optional) Task 1: Add an animation.

50. Edit your glass table and click the **Image** icon 
51. In the Configuration panel, drag and drop (or browse to) the arrow.gif file to upload it to your glass table.
52. Be sure the **Preserve Aspect Ratio** button is *not* selected.
53. On the canvas, move the arrow to its correct location and resize it appropriately.
54. Save the glass table.

(Optional) Task 2: Examine the Source icon and consider customizations for future Glass Tables.

55. Click the **Source** icon  and consult the documentation <https://docs.splunk.com/Documentation/ITSI/latest/SI/Inputs>
56. Consider how you might customize future glass tables by adding inputs and tokens.
57. When you're finished, click **Back**.

End of Lab Exercise Using-2

Lab Exercise Using-3: Using and Customizing Deep Dives

Description

In this lab exercise, you'll use a deep dive to diagnose a problem, and also create a custom deep dive.


Scenario: For our case study, you'll walk through the stages of investigating a web server outage. Our Storefront Web Farm service monitors the health of our web servers—**www1**, **www2**, and **www3**. In the sample data, one of these servers has begun to exhibit an issue, which you'll diagnose with ITSI.

Task 1: Examine a potential problem by navigating from Service Analyzer to Deep Dive.

1. Navigate to the default service analyzer and filter to the Storefront Web Farm service. Examine the **Storefront Web Farm** KPIs. The **Storage Free Space** KPI tile should display an exclamation point (!).
By default, you're seeing the **aggregate** for this KPI, which is measuring the storage space available on your Storefront Web Farm server's disk arrays.
2. In the service analyzer, select **KPI Value: Aggregate** and change it to **Maximum Severity**.
Now the KPIs that are measuring individual entities will show the alert level for whichever entity has the worst alert level. Now you should see that storage free space is zero or very low.
3. Click the **Storage Free Space** KPI to open the service and KPI details panels, with the storage free space KPI selected. If prompted, leave the analyzer without saving.
4. In the service details panel, note that the **Storage Free Space** KPI is showing an elevated alert level, and on the KPI details panel, that one entity, **www2**, is in a high alert state. Something's going on with storage availability on this server.
5. From the service details panel, click **Open all in Deep Dive** to open the service's default deep dive in a new browser tab.

Task 2: Facilitate your investigation by creating a custom Deep Dive.

First, you will save your own personal deep dive so you can alter it without affecting other people.

6. From the Deep Dive browser tab, click **Save as**.
7. Name the deep dive **<your student ID#> - Server Info**, select **Shared in App**, and click **Create**.
8. Select the boxes to the left of the lanes for **CPU Utilization** and **Memory Free**, click the **Bulk Actions** dropdown menu, select **Delete**, and confirm it.
9. Change the time range to the **Last 4 hours** (under **Presets**) so you can examine the behavior in more detail.
The alert level for the Storefront Web Farm became worse within the last hour or so. It's unclear from the deep dive which entity(s) is involved. Let's break out the information by server.
10. Click the  icon for the **Storage Free Space** lane and select **Lane Overlay Options**.
11. For Enable Overlays, click **Yes**. Under Selected Entities, make sure **www1**, **www2**, and **www3** are selected and that "static" is selected. Click **Save**.

12. Now the lane shows 3 trend lines for the 3 servers and the default trend line for all three, but it's not easy to see which line maps to each server.
13. Click anywhere within the graph of the **Storage Free Space** lane and select **Add Overlay as Lane +**.
14. Now each server has its own lane and you can see the point in time when **www2**'s storage space dropped to 0.
15. Click **Save**.

Task 3: Gather server information for an effective IT ticket by examining the OS Host Details dashboard for the web servers.

16. Click **Service Analyzer > Default Analyzer**.
17. Click the **Storefront Web Farm** tile and, at the right, click **Storage Free Space %**.
18. In the column even farther to the right, click **www2**.
19. At the right, under Modules, click **OS Host Details**.
 Another browser tab opens. This is the detailed diagnostic for the **www2** server.
20. Since this is a storage-related KPI, if not already selected, click the **Storage** tab.
 You can see all the important details for the server, including filesystems, device IO statistics, storage used, read/write operations, and latency. This information comes from OS logs for performance and system monitoring.
 Some panels at the bottom of the display may be blank, indicating there are no events to display. This tells you that the normal logging events are not being generated.
21. To confirm what Storage Free Space should look like, return to the **Service Analyzer** and repeat the previous steps for **www1**.
22. Switch back to your browser tab displaying the **www2** OS Host Details dashboard, and select the **Splunk Events** tab.
 This shows you all events related to this server. If you navigate through a few pages of the events, you should see some events from the **df** and **iostat** source types that show errors for devices and the RAID bus. You will investigate these in more detail.

Task 4: Examine the underlying events in your deep dive so you can close the ticket. Save the deep dive for use in the future.

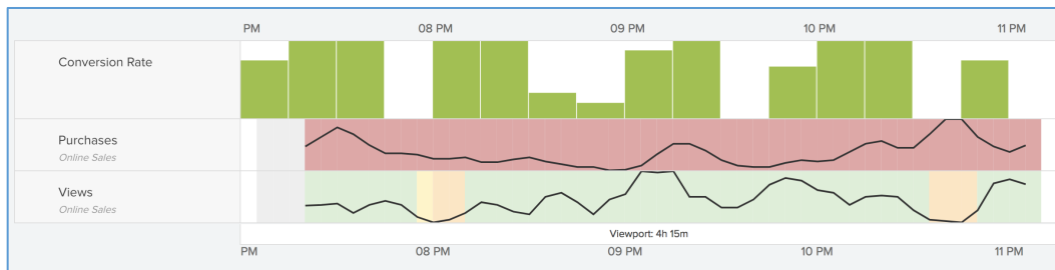
23. Click **Deep Dives** and select your **# - Server Info** deep dive.
You'll add a lane to examine the df and iostat source type errors.
24. Click **+ Add Lane** and select **Add Event Lane** from the dropdown menu and complete these fields:
Title: Storage Error Events
Event Search: index=main host=www* sourcetype=df OR sourcetype=iostat error
25. Click **Create Lane**.
A new event lane is added. You should note darker colored event bars starting near where the **www2** storage failure occurred.
26. **Save** your deep dive.
27. In your new event lane, click one of the darker colored bars to open the events.
You should see some error messages like "RAID BUS ERROR" and "ERROR device not found". You know now that the RAID bus failed at this time on this server.
You could now submit an IT ticket using the server ID, failed device, time of failure. You could also keep this deep dive for the future for examining web server status, or ask your admin to create a Multi-KPI Alert.

Scenario: In this scenario, you build a new Deep Dive and a view-to-purchase conversion rate metric lane.

Task 5: Build a view-to-purchase rate over time metric lane.

28. From the Default Service Analyzer, select the check box on the Online Sales Service, then click **Drilldown to Deep Dive**.
29. Click **Save as** to create custom deep dive named **<your student ID#> - Conversion Rate Deep Dive** and **Shared in App**.
30. Click **Create**.
31. Delete all lanes **except** the Purchases and Views lanes.
32. Add and configure a **Metric** lane as follows:
Title: Conversion Rate
Graph Type: Column
Graph Color: Green
Lane Size: Medium
Search Type: Ad hoc
Search:
sourcetype=access_combined (action=view OR action=purchase)
| timechart span=15m count by action
| eval rate = round(purchase / view * 100)
| eval rate = if(rate > 100, 100 , rate) | fields - purchase,view
33. Click **Create Lane**.
34. Click and drag the **Conversion Rate** lane to the top. Make sure **State Thresholds** are enabled for the **Purchases** and **Views** lanes.
Remember that thresholding is not supported for metric lanes.

35. For Time Range, select **Last 4 hours**. Your deep dive should look something like this:



36. Click **Save**.

37. Close all extra tabs.

End of Lab Exercise Using-3

Lab Exercise Using-4: Working with Episodes

Description

In this exercise, you'll use the Episode Review dashboard to examine and process issues.


Task 1: Examine episodes generated by your correlation searches and KPI alerts.

1. Navigate to **Alerts and Episodes**.

You should see numerous episodes. If not, try refreshing the view.

One episode should be for your Storefront Web Farm, stating that only 3 servers are currently running, and the rest should be warning about a low conversion rate for specific product codes. The conversion rate is the rate at which visitors to the online store are purchasing products. It is based on a comparison of the Purchases and Views KPIs. If there are significantly more views than purchases in a 15-minute period, this type of episode is generated.

Notice the time selector menu, which defaults to Last 24 hours. You can use this to control the time period for the list of episodes. For instance, if you want to look at the episodes from yesterday, you could click the dropdown menu, click Presets and click Yesterday. For now, leave it set to Last 24 hours.

By default, episodes are sorted in reverse chronological order and are color coded to indicate the severity of each episode. You can sort the episode display by Title, Owner, or other fields with the sort control  on the left of the view above the list of episodes.

2. Next to the **Add Filter** dropdown menu, in the **search** field, type **Low Conversion** and you should now see only the Low Conversion Rate episodes. Expand the time range if necessary.

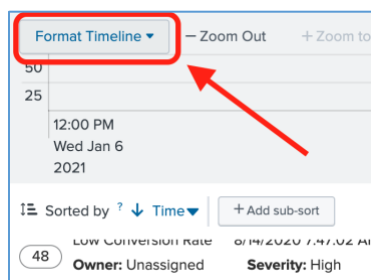
If you need more space on the screen, you can click **Hide Dashboard** to see more episodes and hide the Timeline.

These episodes are being generated by a multi-KPI alert called **Low Conversion Rate**. It looks for instances where the **Views** KPI severity is normal or low, but **Purchase** KPI severity levels are high or critical.

3. Click to display the **View Settings**. You can alter the display options, such as which columns to display, their order, and how to display the episode severity. The Viewing Option Prominent, for example, fills the entire row with the severity color rather than just the edge. Try out some of these settings, but make sure you leave Episode View set to On.

4. Click the **Format Timeline** control. (If you don't see the control, click **Show Dashboard** -if you've previously hidden it- and search the text on the page.)

This control displays a timeline of all episodes in the current time period (defaults to 24 hours). You can use the zoom controls to locate specific episodes by time. Clicking on a timeline column filters the episodes to that unit of time and zooms to that unit. This view can be very useful to determine patterns of episode creation, such as every hour, or at a similar time each day, etc.




Task 2: Take ownership and create a custom view.

All class participants are working with the same set of episodes, so you'll each work with a different episode. Notice each episode description starts with **Product Code: N**. This is an arbitrary value added to the description in your lab environment to give you your own episodes to work. Note that new events are being added to the episodes every minute. This is artificially exaggerated in our lab environment, to make sure we have plenty of episodes to work with and so we can see the effects of changes quickly.


5. Make sure you have filtered to the **Low Conversion** episodes.
6. Locate an episode with a product code value matching your own student ID.
For example, if you are assigned the user **analyst5**, in the Description column at the right, locate an episode associated with **Product Code: 5**.
7. Click the episode.
A details panel appears on the right.
8. At the top of the details panel that just opened, click **Acknowledge**.
This assigns the episode to you and changes its status to **In Progress**. You have now taken ownership of your episode, which is the first step in working the issue.

Task 3: Narrow your view using a filter to see only your episodes and save it.

Let's narrow your view to only your episode.

9. Select **Alerts and Episodes** to clear previously applied filters.
10. Select **Add Filter > Owner** and select your name.
The list of episodes should now be filtered to the episode you acknowledged in the previous task. This could be a useful view—just the episodes you're working.
11. To save this filtered view, click **Save as**.
12. Name your view: **<your student ID#> - Episodes for <your_name>**
13. Leave the default permission set to **Private**.
14. Click **Create**.
Your view name is now displayed at the top of the page.
15. Navigate to the default Service Analyzer and then back to **Alerts and Episodes**.
Note that your saved view is still in effect. This view will now open for you each time you navigate to the Episode Review page, until you change to a different view. (Your most recently accessed view will be the default to appear when you click Episode Review from the ITSI menu.)
16. At the top left, to the left of the title, click **>** to view a dropdown menu from which you can select any other saved view. (The last view you selected or viewed will become default).
You won't do this, but to delete a view quickly, click **>**, mouse over the view name, click , then click Delete View.

Task 4: Work the issue.

17. Click your acknowledged episode to display the episode details.
 Note the severity (High) and status (In Progress).
 You may see a warning that some events in the episode are outside the selected time range or you may not have permissions to view some events.
18. Click the **Activity** tab.
 Notice your actions are being automatically logged for later use.
19. Return to the **Impact** tab.
 The description at the top is generated by the correlation search. You also see the service tile (with a link to its deep dive) and service topology (with a link to tree view service analyzer details) and impacted entities and related tickets. There may be other drilldown options (which may exist as customizations at your site).
20. Click **Analyze in Deep Dive** to see the Online Sales' service health score.
21. Close the deep dive browser tab and return to the **Episode Review** page.
22. To add a comment, select the **Comments** tab, and type: **Investigating low purchase rate**
23. Click **Add comment**.
24. Select the **Events Timeline** tab.
25. In the **Sort for:** dropdown menu, click **Root cause analysis** and hover over the exclamation point icon.
 It looks like the first concern was that purchases were critical while views were normal.
26. Below Root cause analysis and under Event type, you see the "Indicator ..." search. Click the link to that search to view the individual notables. From here you can edit columns or search for more detail. You can also access a similar view by clicking the All Events tab.
27. Maybe you'd like to see how a teammate's insight on these events compares to yours. Click the **Actions** dropdown menu, click **Share episode** and click the  **Copy link** button to copy the url. Now you could send this view to your coworker.
28. Click **Common Fields > all_info**. Here you can examine the frequency of the events you were just viewing.
29. Scroll back up to the top of the page.
 Click **Dashboards > Event Analytics Audit**.
 The Event Analytics Audit dashboard opens, where you can explore data describing how Episodes are being worked.

Optional Task: View Similar Episodes to see how they were worked.

30. Start at the **Alerts and Episodes** browser tab, then select your assigned episode.
31. Click the **Similar Episodes** tab to see episodes similar to the one you've been working. (In this case, you'll see your classmates working on identical episodes.)
32. On this tab, you can adjust the time range and the types of similarities you prefer.
Here you'd be able to see how similar episodes were worked, documented or resolved.
In this case, you are seeing your classmates work on an identical episode.

End of Lab Exercise Using-4

Lab Exercise Implementing-1: Post-installation Check and Initial Configuration

Description

In this exercise, you'll examine an ITSI server that has just been installed and verify that it is ready to configure. You will also begin the process of identifying Splunk events and data models that you'll need to build the customer's services.

From this lab exercise onward, references to your "lab server" will be a new server for your own private use. Get the address for the server from your instructor and record it. The shared server you used for the preceding lab exercises will remain available for reference, but all new work will be on your private server.

| | |
|------------------------|----------------|
| LAB Server Address | |
| Splunk Analyst UID/PWD | admin / |

Steps

Task 1: Log into Splunk as admin and explore the user interface.

1. In your web browser, navigate to your assigned private lab server and log in as admin.
2. Navigate to the **IT Service Intelligence** app.
You may see a system message reporting that no entities could be imported by SAI—this is expected; you can delete this message.
Note the Welcome dialog box—nothing has been done here yet. As soon as you create a service, the Welcome dialog box is disabled.
3. Close the **Getting Started** dialog box. Note that the service analyzer is empty.
4. Select **Service Analyzer > Analyzers**—there are no saved service analyzers yet.
5. Check the **Glass Tables** and **Deep Dives** menus—there are none yet.
6. Select **Search** to open the search view in IT Service Intelligence.
7. Click **Data Summary**. Verify that data is being indexed. Examine some of the most common sourcetypes and hosts.
8. Run the following search over **Last 30 Days**:
`sourcetype=access_combined`
access_combined (web server logs) will be very important. You should find that these sources go back 60 days or more. This will be important for ITSI implementation, as several features require a long baseline of events to analyze.
9. From the **App** menu, select **Manage Apps**.
10. Filter the list for **"ITSI."**
You should see all the apps installed by ITSI (plus one for class lab data generation.) In particular, note the apps starting with "ITSI Module...". These are the ITSI modules that help us model services. You'll begin using them in the next lab exercise.

If you notice any apps reporting newer versions, do not update them. Your lab servers

are updated periodically by Splunk Education staff and then tested to ensure lab instructions still work as written.

11. Navigate to **Settings > Indexes** and sort by the App column, then find the app **SA-IndexCreation**. Note these indexes; they are installed with ITSI.
12. Navigate to **Settings > Data Inputs**.
Note the extensive set of modular inputs associated with ITSI or IT Service Intelligence. None of these require admin interaction, except for the **IT Service Intelligence CSV Import**, which is used to discover and import entities—you'll learn about this later.

Task 2: Configure the Operating System module.

13. Navigate to **Settings > Searches, reports and alerts**.
14. For **App** select **ITSI Module for Operating Systems (DA-ITSI-OS)**. Make sure the Owner is set to **All**. These are the saved searches for the OS module.
15. Note the **DA-ITSI-OS-OS_Hosts_Search** saved search.
This is the saved search the OS Module uses to do entity discovery for servers in your environment.
16. Click **Run** next to the above search. The search executes but returns no results.
17. Note the search begins with the macro ``itsi_os_module_indexes``. If you remove this macro from the search string and re-run the search, it should return results.

The macro that ships with the module looks in specific indexes for events from the **nix** and **windows** add-ons, but these indexes are not guaranteed to exist at any given ITSI installation. You need to have the macro search an additional index.

18. Navigate to **Settings > Advanced search > Search macros**, and select **All** for the app.
19. Filter for the macro name **itsi_os_module_indexes**.
20. Click **itsi_os_module_indexes** to open the macro editor.
21. In the **Definition** field, add " **OR index=main**" to the existing search string. It should now look like this:
(index=windows OR index=perfmon OR index=os OR index=main)
22. Click **Save**.
23. Repeat the steps above to run the **DA-ITSI-OS-OS_Hosts_Search** again and confirm that results are now returned by the saved search. The module will now be able to discover entities in your environment.

End of Lab Exercise Implementing-1

Lab Exercise Implementing-2: Designing Services

Description

You'll work together to define one of the services for Buttercup Games. You will initially focus on defining the service to support the online sales operations team. Then, you'll build some base searches for the data you need in your services.

Task 1: Define the service name and the KPI requirements for the business service.

- Based on statements of the customer sales ops team, define a service and general KPI requirements. From the customer:

"We want a status board that updates every minute and shows the last 15 minutes' overall efficiency of our online sales, broken down by the number of times products has been viewed or bought, and the total volume of web content our customers viewed. We are really focused on the number of items bought—this is a key factor. For online sales, the purchases are the important thing. The purchases KPI should be twice as important as the others. Volume is significant, but less critical than other items."
- What shall we call the new service? **Service Title:** _____
- Now, identify the KPIs based on the customer statement. Remember, that service health scores are created automatically so you don't need to define them here. For each KPI, fill in the following grid, using information from the customer statement:
 - Fill in a quote from the customer statement that establishes the KPI
 - Give the KPI a name
 - Fill in frequency and time span to match their needs
 - On a scale of 1 to 10, identify the importance. Remember that 5 is the default, and 11 means the service health score cannot be better than the KPI's alert severity.
 - For threshold type, enter "low" if lower values are desirable for this KPI; "high" if the KPI is better with higher values, or "mid" if either extreme low or high values could be bad.

| KPI Name | Requirement | Freq. | Time Span | Imp. | Threshold Type |
|----------------------|---|----------|-----------|------|----------------|
| Service Health Score | Overall efficiency | 1 minute | 15 min | -- | -- |
| Views | Number of times a product has been viewed | 1 minute | 15 min | 5 | high |
| | | | | | |
| | | | | | |

- Based on the customer requirements, which if any KPIs need to be split by entity?

Task 2: Design the web farm service.

Use the same approach you used when designing the Online Sales service to plan a new service for the web farm. Do not implement it in ITSI yet—this is just planning.

- Use this input from the IT team for requirements:

“The most important thing for us is our website health. How many errors are being generated? Errors in the storefront web app cause huge problems for us. Also, what is the average usage of CPUs, memory and disk space per machine in the web farm? We’d like to see this update every minute and show the last 15 minutes’ data. And we want to be alerted if the number of servers in the web farm falls below our service level.”

- First, define the web farm service name: **Service Title:** _____
- Next, document the KPIs for this service, using the information you gathered during the initial data audit (in lab exercise 1):

| KPI Name | Requirement | Freq. | Time Span | Imp. | Threshold Type |
|----------------------|-------------------------------------|----------|-----------|------|----------------|
| Service Health Score | Website health | 1 minute | 15 min | -- | -- |
| Errors | How many errors are being generated | 1 minute | 15 min | 11 | low |
| | | | | | |
| | | | | | |
| | | | | | |

- Based on the customer requirements, which if any KPIs need to be split by entity?

End of Lab Exercise Implementing-2

Lab Exercise Implementing-3: Data Audit and Base Searches

Description

From the previous lab exercise, we know the services we're building and the KPIs required, along with some of their settings, like scheduling, importance and threshold type. Now we want to complete the service implementation plan by filling in the source of the data each KPI will display: What events are we looking for, which fields in the events are relevant, and how will we summarize the data (count, sum, average, etc.).

Task 1: Identify useful kinds of events.

Based on discussions with the customer (from the slides), you know they use an Apache-based web storefront that logs NCSA web events to the **access_combined** source type.

| System | Related Source Types or Modules |
|--|---|
| NCSA logs showing customer interaction with the website, as well as error events | access_combined |
| Server performance metrics, including CPU, memory and disk space data | Operating Systems module, CPU , Memory and Storage KPI groups |

And their statement of requirements:

"We want a status board that updates every minute and shows the last 15 minutes' overall efficiency of our online sales, broken down by the number of times a product has been viewed or bought, and the total volume of web content our customers viewed."

Use the following breakdown of customer expectations to map out the source types and fields you should focus on.

Note that you do not need to fill in your answers in the tables below—you can use your own scratch paper or text editor file to do that.

Online sales operations requirements:

| Statement | Source Type & Field / Module & KPI |
|--|---|
| "Overall efficiency of our online sales..." | Service health score |
| "Number of times a product has been viewed..." | access_combined field = action, value = view |
| "Number of times a product has been bought..." | |
| "Total volume of web content..." | |

IT team requirements:

| Statement | Source Type & Field / Module & KPI |
|---|--|
| "How healthy is the website..." | Service health score |
| "...are errors being generated..." | |
| "...what is the usage of CPU..." | Operating System module CPU Utilization: % template |
| "...what is the usage of memory..." | |
| "...what is the usage of disk space..." | |

1. Fill in the empty cells in the tables above.

- Use Splunk search commands to examine the available fields in the source types and identify the field(s) that contain(s) the information needed to fulfill the stated requirement.
- Use the documentation at docs.splunk.com/Documentation/ITSI/latest/IModules/OSModuleKPIsandthresholds to determine which KPIs contained in the OS Host module map to requirements.
- For these module-based KPIs, test run the corresponding module-sourced base search to verify each base search returns events.

(Hint: Select **Configuration > KPI Base Searches**, and look for the base searches from the **DA-ITSI-OS** module that correspond to your KPI titles. Open them and use the **Run Search** link to test run the base search in a new window. If events are found, the base search is working.)

Task 2: Build base searches.

Now that you know what kind of KPIs you'll be creating, you should consider using base searches to improve performance. In the real world, base searches improve performance by producing multiple values for KPIs with one search.

In this case, you won't be realizing true performance improvement, because you're only creating three KPIs, and each will need to be in its own use case because each uses a different set of source events.

Each base search needs to be defined with a single source (discrete set of events), but you can then define multiple metrics, or results, to gather from the set of events.

In this project, the source events all come from the same data source (sourcetype=access_combined), but require three different filters:

- Volume of all events (no filter)
- View events (action=view)
- Purchase events (action=purchase)

In this task, you only create one metric in each base search—so you won't realize any performance improvement. However, you are preparing for the creation of additional KPIs in the

© 2021 Splunk Inc. All rights reserved. Implementing Splunk IT Service Intelligence 4.9

future. For each new KPI, based on the same selection of events, you can add more metrics, which will not increase the total number of searches for the Splunk scheduler to manage.

In addition, you can revise the base searches as needed, and the changes are automatically propagated to all dependent KPIs immediately.

You'll only be building base searches for the online sales service, since the web farm service KPIs will depend mainly on module KPIs.

2. Make sure you are logged in to your private lab server as **admin**.
3. Navigate to the IT Service Intelligence app.
4. Select **Configuration > KPI Base Searches**.

Note there are numerous pre-built base searches—these are defined in the default modules that ship with ITSI. Note the naming conventions—the module name is the prefix. You should use a naming convention so you can find your base searches easily.

5. Click **Create KPI Base Search** and enter the title: **BCG: All Web Traffic**
6. Click **Create**.
7. In the Search field, replace the asterisk with the following search:
sourcetype=access_combined
8. Click **Run Search** to make sure you get results.
9. Configure the following parameters:

KPI Search Schedule: Every minute
Calculation Window: Last 15 minutes
Monitoring Lag (in seconds): 30
Split by Entity: No
Filter to Entities in Service: No

10. Click **Add Metric** and configure the following parameters:

Title: total_bytes
Threshold Field: bytes
Service/Aggregate Calculation: Sum
Fill Data Gaps with: Null values
Threshold for Null values: Unknown

11. Click **Add**.

12. Click **Save**.

At this point, you're done with this base search for lab exercise purposes. You don't have any other metrics you need for this set of events.

In the real world, you might also add metrics for the total count of events, distinct count of sessions, average of duration, etc.

All of these would be generated each time the base search runs, and would be available for use in future KPIs, without increasing scheduler load as much as individual searches for each KPI would.

13. Navigate back to the KPI Base Searches configuration page.
14. Create the base search for the Views KPI, using the following parameters:

Title: BCG: Web Views
Search: sourcetype=access_combined action=view
KPI Search Schedule: Every minute
Calculation Window: Last 15 minutes
Monitoring Lag (in seconds): 30
Split by Entity: No

Filter to Entities in Service: No

Metric parameters:

Title: view_count
Threshold Field: _time
Service/Aggregate Calculation: Count
Fill Data Gaps with: Null values
Threshold level for Null values: Unknown

15. Create the base search for the Purchases KPI using the following parameters:

Title: BCG: Web Purchases
Search: sourcetype=access_combined action=purchase
KPI Search Schedule: Every minute
Calculation Window: Last 15 minutes
Monitoring Lag (in seconds): 30
Split by Entity: No
Filter to Entities in Service: No

Metric parameters:

Title: purchase_count
Threshold Field: _time
Service/Aggregate Calculation: Count
Fill Data Gaps with: Null values
Threshold level for Null values: Unknown

Consolidating base searches

You might be asking, "Why build three base searches, each with only one metric, when the search strings are so similar?" And you'd be right. Given the small scale of your lab environment, these are not very realistic. You might be trying to think of a way to consolidate the three base searches into a single base search that uses three different metrics.

You could create one base search with the search string:

```
sourcetype = access_combined | eval is_{action} = 1
```

This search generates additional fields in each event based on the value of the action field. If **action=view**, then **is_view=1** is created. Now, you can create three metrics: **sum(bytes)**, **sum(is_view)**, and **sum(is_purchase)**. This will scale better, and you can produce individual KPIs for the Purchase and View actions, using a single base search!

End of Lab Exercise Implementing-3

Lab Exercise Implementing-4: Implementing the Business Service

Description

Using the information from the previous lab exercises, implement the Online Sales service for Buttercup Games.

Task 1: Create the Online Sales service.

1. Navigate to the IT Service Intelligence app.
2. Select **Configuration > Services** and click **Create Service > Create Service**.
3. Complete the dialog as follows:
 - Title:** Online Sales
 - Description:** Monitor online sales
 - Team:** Global
 - Select **Manually add service content**
4. Click **Create**.
 - You won't be adding entities to this service at this time.
5. Select the **KPIs** tab.
 - You'll be adding the three KPIs from your plan. You'll start by creating the **Views** KPI.
6. Click **New** and select **Generic KPI**.
 - The 7-step KPI dialog box appears.
7. In step 1, set the title of the KPI: **Views**, with the description: **Online Sales views by customers**
8. In step 2, click **Base Search**, and select the **BCG: Web Views** base search.
9. Select the **view_count** metric.
10. Click **Finish** to skip the remaining steps (you will configure the KPI's thresholds and backfill options in subsequent lab exercises).
11. Click **Save**, then **Save and Enable** to save the current service configuration and enable the new service.
 - Note that in some cases you might want to leave a new or partially completed service disabled, but you will enable this one now to save time and so you can begin to immediately generate KPI events for testing.
 - Congratulations, you have created your first KPI!
 - You can now clone the **Views** KPI to create the **Purchases** KPI because it is almost identical.
12. On the KPIs tab, click **Clone**, select the **Online Sales** service, and the **Views** KPI.
13. Click **Clone**.
 - The **Views (copy)** KPI is created.

14. Click this KPI and make the following changes:
 Change the title to **Purchases** and click **Done**, then change the description to "**Online Sales purchases by customers**" and click **Done**.
 Under **Search and Calculate**, click **Edit (for Source)** and change the selected base search to **BCG: Web Purchases** and the metric to **purchase_count**.
15. Click **Finish** and click **Save** to save the new KPI.
16. Create the **Volume** KPI now, using the same process you used for the **Views** KPI, except as follows:
Title: Volume
Description: Online Sales web traffic volume
KPI Source: Base Search
Base Search: BCG: All Web Traffic
Metric: total_bytes
Enable backfill: On, for 14 days
 Do not set any thresholds now.
17. Click **Finish**.
18. **Save** the service. The Volume KPI backfill starts.
 Note, you are backfilling the Volume KPI now because you will be configuring adaptive time thresholds for that KPI in the next lab exercise.
 You are waiting to backfill the Purchases and Views KPIs until after you configure static thresholds. You need historical KPI alert and severity values before you backfill the service health score, which is used by the predictive analytics feature. At production sites, there may be enough time between KPI creation and later analytical work that backfill might not be needed.
 Also, if you ever need to "re-backfill" a KPI, you can do so by first cloning it, then deleting the old KPI and re-naming the clone to replace the original. The "new" version of the KPI can now be backfilled again.

Task 2: Configure service health importance.

Now you'll configure the service health score importance for each of our KPIs.

19. Select the **Settings** tab of the service editor.
20. Use the sliders to set the importance for each KPI as previously identified (lab exercise Implementing-2).
Important: Do **not** backfill the service health score at this time.
21. Click **Save**.
22. Try using the Simulated Severity controls to test different alert level combinations and see what the effect is on the service health. This is a simulation only—it has no effect on the service.

Task 3: Create a Service Analyzer.

By default, there is one global service analyzer, which is displayed as the default page for the ITSI app. However, if you click the **Service Analyzers** menu, none are listed.

23. Select **Service Analyzer > Default Analyzer**.

The default ITSI service analyzer showing your new Online Sales service is displayed. (it sometimes takes a few minutes for a new service to show.)

Click **Save as...** and enter **Buttercup Games Services** for the service analyzer title.

24. Select **Shared in App** for permissions.

25. Click **Create**.

26. Select **Service Analyzer > Analyzers**.

Note your new service analyzer is displayed here. In the future, you can make changes to this service analyzer, such as filtering which services are displayed or how many KPIs are displayed.

27. Select **Edit > Edit Permissions** and note that you can configure which roles have access to this analyzer.

Along with other Service Intelligence UI views like glass tables and deep dives, you can use roles-based permissions to configure tools based on user needs.

28. Modify the permissions for your new service analyzer. Configure it so only **admin** users can modify it, but all users can access it.

29. Open the **Buttercup Games Services** service analyzer.

Notice that all the tiles are showing a status of green (normal). This is because you have not configured thresholds yet—you'll do this in the next lab exercise.

Also notice your Volume KPI has a full sparkline, but Views and Purchases do not. This is the result of backfilling the Volume KPI earlier.

End of Lab Exercise Implementing-4

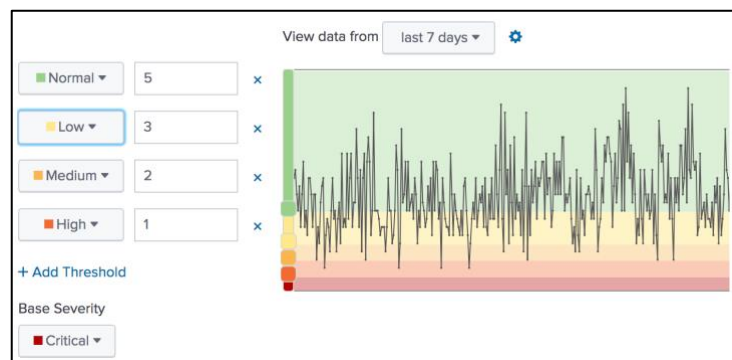
Lab Exercise Implementing-5: Thresholds and Time Policies

Description

In this lab exercise, you'll configure thresholds for the Online Sales service.

Task 1: Configure static aggregate thresholds.

1. Select **Configuration > Services** and open the **Online Sales** service.
2. From the **KPIs** tab, select the **Purchases** KPI.
3. Expand the **Thresholding** tab.
4. Use the **View data from ...** link to examine the pattern of the source events over the last 7 days.
5. Use the **Add Threshold** link to define five thresholds as per the image below. Adjust the sliders for the thresholds until most of the data for the last 7 days is normal, but approximately the lower 10% make up the highest alert levels. The exact values of the threshold alert levels do not matter for this exercise, but set high to 1, so that critical only occurs if there are zero purchases during the 15 minute period. The following screenshot is an example—your actual thresholds will likely be different.



Example thresholds for Purchases and Views

6. Click **Save**.
7. Expand the **Search and Calculate** tab.
8. For **Backfill**, click **Edit**.
9. Enable backfill and select **last 14 days**.
10. Click **Finish**.
11. Repeat the above steps to configure static thresholds for the **Views** KPI. Make sure to backfill for 14 days.
12. Click **Save** to save the service.

Task 2: Configure adaptive thresholds.

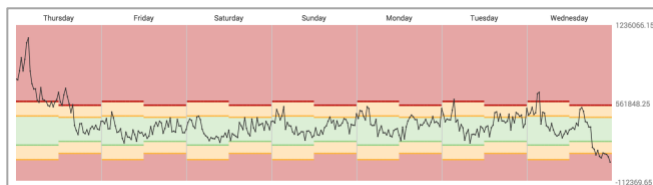
13. Make sure the backfill job for Volume is complete—you should see a system message at the top of the Splunk Web page.
14. Select the **Volume** KPI and expand the **Thresholding** tab.
15. Enable both **Time Policies** and **Adaptive Thresholding**.
16. Set the **Training window** to **14 days**.

The **Preview Aggregate Thresholds** display appears and shows the alert values for this KPI over the last 7 days. This graph shows the KPI values that were backfilled.

17. Hover your mouse over the preview panel to assess the data ranges. Check the highest and lowest values.
18. Expand the **Configure Thresholds for Time Policies** tab. Note that everything is "normal" right now—you have not added any time policies yet.
19. Scroll up to the top of the **Thresholding** section, click the **Select a thresholding template** dropdown menu and select the **AM, PM (adaptive/stddev)** policy. You'll be prompted to apply the template and discard existing threshold settings—click **Apply**.
20. Expand the **Configure Thresholds for Time Policies** tab and you'll see there are now two additional time policies in the left panel—*Everyday, 12AM-12PM* and *Everyday, 12PM-12AM*. But the adaptive thresholds are not set—it's using a generic standard deviation pattern at the moment.
21. Click **Apply Adaptive Thresholding**.

Note: You might need to scroll down inside the **Aggregate Threshold Values** pane to see the button.

The adaptive threshold dataset is now applied. You should see something like this in the **Preview Aggregate Thresholds** panel:



22. Click **Save** to make sure your changes are saved.
23. Navigate to your saved Service Analyzer (**Buttercup Games Services**).
24. Click the **Online Sales** service. A service panel opens on the right.
25. Mouse over each of your three KPIs to see the thresholds you set.

Task 3: Inspect the `itsi_summary` index.


First, you can check to make sure your KPIs are being generated in `itsi_summary`.

26. Open a search window and search for all events in the **`itsi_summary`** index over the **Last 7 days**. Because you backfilled all the KPIs, you should see thousands of events.
27. Examine the values for the **`kpi`** field. These are the titles for your KPIs, plus the service health score.
28. Examine the values for the **`alert_value`** field. These are the actual values the KPI searches detected.
29. Examine the values for the **`alert_severity`** field. These are the labels for the state of the KPI. Note that the service name is not shown—the service is identified by an ID (the “serviced” field) instead of a name.

Task 4: Create a custom threshold template.

To extend your understanding of thresholding, see what you need to do to create your own special threshold template. You'll create a template that applies the adaptive standard deviation policy Monday through Friday but exempts the weekends.

Before you do this task, you should make sure you have not altered your timezone settings in the administrator's account preferences. It should be set to Default System Timezone. This setting keeps your view synchronized with the indexer time (UTC) to avoid confusion while setting time periods.

30. Open the Online Sales service in the service editor (**Configuration > Services**).
31. In the thresholding tab of the **Volume** KPI page, select **Set Custom Thresholds**.
32. Expand the **Configure Thresholds for Time Policies** tab.
33. For **Everyday, 12AM-12PM**, click the more options menu , then **Edit** to open the **Update Time Policy** editor.
34. Update the **Title** to: Weekdays, 12AM-12PM
35. Uncheck **Su** and **Sa**, then click **Done**.
36. Repeat the above steps for the **Everyday, 12PM-12AM** policy.
 At this point, you've removed the weekend days from the policy list. Now the **Default** policy will apply to KPIs during that time interval. You could go on to customize the default policy, but instead you'll add a policy for the weekend.
37. Click **+ Add Time Policy**.
38. Enter the title **Weekend**.
39. Set **Duration** to **24 hours**.
40. For **Repeat**, check only **Su** and **Sa**
41. Click **Add**.
42. For the new **Weekend** policy, set **Base Severity** to **Info**.
 This KPI will no longer affect the service health score during the weekend.
43. Scroll up to the top of the Thresholding section, click the **Save as template** link, and title the template **BCG: adaptive/standard deviation, no weekends**, then click **Save** to save the new template.
44. **Save** the service definition.
45. Navigate to **Configuration > KPI Threshold Templates** and type "BCG" into the filter.
 Note your template is now saved for future use.
46. Open your threshold template.
 Note that you can edit it here. If you make changes to the threshold template, it affects all KPIs using that template.

End of Lab Exercise Implementing-5

Lab Exercise Implementing-6: Entities and Modules

Description

In this lab exercise, you'll configure pseudo entities for the Online Sales service, add entities, create a new technical service using module KPIs, apply entity types entities, and use the Entity Health page.

Task 1: Break online sales purchases down by product category.

If you look at the `access_combined` source type fields, you'll note a `categoryId` field that identifies the type of product involved in a transaction. The sales team would like to be able to monitor sales per category, so you will need to alter the base search for purchases to split by entity, where the `categoryId` field identifies the entity. You'll use the categories as an abstract entity. For this use case, you don't need to formally import entities—a simple entity split does the job.

1. Navigate to **Configuration > KPI Base Searches** and select the **BCG: Web Purchases** base search.
2. Change **Split by Entity** to **Yes**.
3. Change the **Entity Split Field** to `categoryId`.
4. Edit the **purchase_count** metric.
5. Set **Entity Calculation** to **Count** and **Service/Aggregate Calculation** to **Sum**.
 This setting counts the purchase events for each category, then sums the category counts as the aggregate value of the KPI.
6. Click **Done**, then **Save**.
7. A warning message displays; click **Save** to complete the save operation.
8. Navigate to **Configuration > Services** and edit the **Online Sales** service.
9. Select the **Purchases** KPI and expand the **Thresholding** section.
10. Click **Per-Entity Thresholds**. A new threshold map displays.
11. The base severity should be set to Normal. Change it to **Info**. You won't be changing the alert severity per category.
12. **Save** the service.
13. Navigate to the default service analyzer and click the **Purchases** KPI tile.
 You should now see a list of entities for the KPI with a severity of Info.

Task 2: Create entities for the Web Farm service.

Now you'll create the service for the Web Farm, using KPIs provided by the OS Hosts module. In preparation, you'll create the entities for the web farm servers.

14. Navigate to **Configuration > Entities**.
15. Select **Create Entity > Import from Search**.
16. Select **Modules > ITSI Module for Operating Systems > OS Hosts Import**.

This search is loaded:

```
| savedsearch DA-ITSI-OS-OS_Hosts_Search
```

The **DA-ITSI-OS-OS_Hosts_Search** saved search retrieves host names and additional attributes from multiple sources and references `oshost` objects to find events with

version or *vendor_product* values. These usually come from **Splunk_TA_nix** or **Splunk_TA_windows**, so make sure those are configured when you install on prem.

17. Click the  button to run this search.

Note that several columns of information are returned for each server, and that all servers in the environment have been returned. In practice, you will usually want to finetune this.

18. Add the following to the existing search and re-run:

```
| search host=www* OR host=supp*
```

Now you should only see your webserver hosts—the *www** servers are for the public storefront website, and the *supp** servers are for your support site.

19. If the search doesn't find 8 entities, increase the time range and re-run the search.
20. Click **Next** to display the **Select Columns** page. In the **Import Column As** column, make sure **Entity Title** is selected for the **host** field, make sure **Entity Information Field** is selected for all other fields that have sample values. Set the remaining fields (*cpu_cores* and *memory*) to **Do Not Import**.

Use **Update Existing Entities** which is the default action for Conflict Resolution. This is usually the best option. It adds new entities, but for existing entities it only adds new field values but does not alter any existing fields.

21. In the Preview section below, click **Entities to be Imported** to see the entity titles and information that will be imported.
22. Click **Import**.

The new entities are created. The confirmation page is displayed. Note that this search can now be saved as a recurring import so you can repeat it in the future, including on a schedule to automatically add new entities that are created in your environment. You can also edit the search to finetune it to select specific hosts, and you can add more searches later to find different types of hosts.

23. Click **View All Entities**, scroll through the list of entities and note that the entities for the web servers have been added.

The entity title serves as the first alias for the entity, and you will use the alias within the entity rules later.

Task 3: Create the Web Farm service.

Based on your design, the IT team wants a Web Farm service to monitor the health of the web servers. For this service, you'll save time by using the OS Hosts Monitoring module, which automatically creates the KPIs for you. Then, you'll add the generic KPI that the service design calls for—the Errors KPI.

24. Navigate to **Configuration > Services** and click **Create Service > Create Service**.

Title: Web Farm

Description: Monitor web site servers

25. From the list of prebuilt KPIs, select the **OS Hosts Monitoring** category on the left.

26. Make sure only these three KPIs are checked:

- CPU Utilization: %
- Memory Free: %
- Storage Free Space: %

These are the best "fits" in the list of pre-built KPIs for the KPIs in your plan.

27. Click **Create**.

The new service is created and you are taken to the service edit window, with the **Entities** tab active.

The Entity Rules form for the service appears. This form controls which entities are part of this service. By default, there are two rules ANDed together: **host does not match blank**, AND **itsi_role matches operating_system_host**.

28. Change the first rule to: **host – matches – www*** and press **[tab]** to cause the rule to update.

29. Delete the second rule regarding "Info."

Only the three **www*** servers should appear in the list below.

30. Click **Save**, then **Save and Enable**.

31. On the **KPIs** tab, select the **Memory Free: %** KPI and examine the **Search and Calculate** settings. Note the KPI is sourced from a KPI base search and is split by host (all the entities in the service, the three web servers) and are also filtered to only the entities in this service.

32. Click **Edit** for the **Source** section.

Note that this KPI is set to use the **Performance.Memory** base search from the **DA-ITSI-OS** module.

33. Click the **Edit Base Search** link and inspect the base search in a new browser tab.

Note that the search expression is:

```
(`itsi_os_module_indexes` tag=oshost tag=performance tag=memory)
```

This search uses these tags to identify the events that correspond to the **Memory** object in the **Host_OS** data model.

34. Close the browser tab to return to the service editor and click **Cancel** to close the KPI editor.

35. To add the **Errors** KPI, click **New**, scroll down to the **Templates** section near the end, and select **Count Based Ad Hoc KPI**.

36. Update the following parameters for Steps 1-3:

Title: Errors

Description: Errors in web farm.

Search: sourcetype=access_combined status>=400

Threshold Field: _time

Split by Entity: Yes

Filter to Entities in Service: Yes

The **host** field is automatically added to the **Entity Filter Field**. This is the field in the event that you will split by. This generates KPIs for each server in the service—all three "www*" servers in this case.

37. On step 3, click the **Generated Search** link to expand the search window and examine the search generated so far. This search becomes the KPI saved search when you're done.

38. On step 4, configure the Calculation Options as follows:

KPI Search Schedule: Every minute

Entity Calculation: Count

Service/Aggregate Calculation: Sum

Calculation Window: Last 15 minutes

Fill Data Gaps with: Null values

Threshold level for Null values: Unknown

The explanation should read:

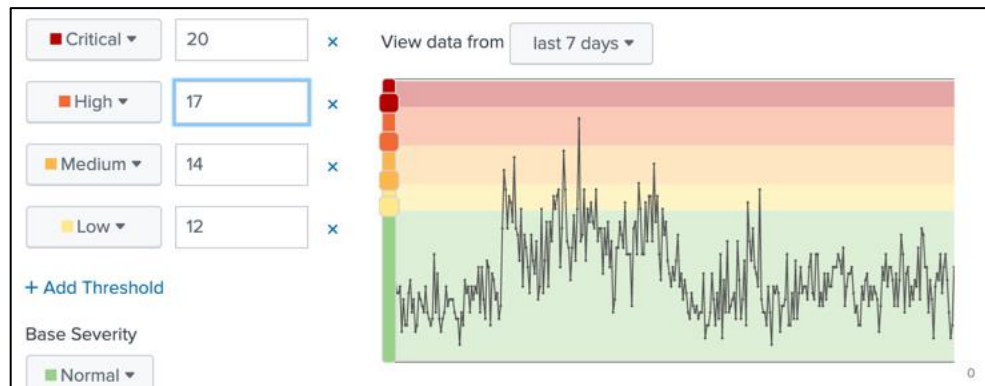
"Every minute take the count of _time for each entity as the entity value then take the sum of all entity values as the service/aggregate value all over the last 15 minutes. Fill gaps in data with Null values and use an unknown threshold level for them."

39. On step 5, the **Unit** will show up next to the KPI value in numerous places. Clear the default value "ct." Leave the **monitoring lag** at 30 (default).

40. On step 6, backfill the KPI for the **last 14 days**.

On step 7, the threshold configuration appears. Note that it has arranged a "low normal" threshold pattern automatically.

41. Set the Aggregate Thresholds as follows (your sample data may be different):



42. Click **Finish**, then **Save**.

43. Click the **Settings** tab and configure the **Errors** KPI as a minimum health score KPI (importance of 11).

44. Click **Save**.

45. Open the default service analyzer. The new Web Farm service should be displayed.

Task 4: Backfill the module KPIs.

You've already configured backfill for the Errors KPI, but the other 3 imported from the OS Hosts module do not have backfill enabled. You'll want that historic baseline in a later lab exercise, so you'll backfill those KPIs now.

46. Navigate to **Configuration > Services** and edit the **Web Farm** service.
47. Select the **KPIs** tab and select the **CPU Utilization** KPI.
48. Open the **Search and Calculate** section and click **Edit** next to **Backfill**.
49. Enable backfill for **14 days**.
50. Click **Finish**.
51. Repeat the above steps to enable backfill for the **Memory Free** and **Storage Free Space** KPIs.
52. Click **Save**. The backfill jobs begin, and messages are posted when the backfills are complete.

Task 5: Work with entity types and the Entity Health page.

You did not select entity types for the www* and supp* servers during import. This will restrict the information shown in the entity health display.


53. Navigate to **Configuration > Entities** and click **View Health** for one of the www entities. Recent events for this entity are displayed, but if you select the **Analytics** tab, no analyses are available.
54. Return to the Entity configuration page and filter for all www* entities.
55. Click one of the www* entities to open its **Edit** display.
56. In **Entity Types**, enter "*nix" and save the entity.
57. Repeat the above steps for the other www* entities.
58. Examine one of the entity's health display again.
Note that there is now an additional dashboard tab.
59. On the **Analytics** tab, select some data items from the **Events** category (left side panel), such as **cpu**, **ps**, **df**, etc.
60. Return to **Configuration > Entities** and click the **Entity types** tab.
The Entity types you see can't be changed, but you can make your own.
61. Click **Create Entity Type** and name the type "Custom *nix".
You can make an external URL navigation shortcut associated with this Entity type. You'd use the URL of a dashboard, or even some API endpoint.
In this lab exercise, you will use a placeholder URL to the documentation for creating custom entity types.
62. In the Navigations section, name the navigation *Custom Entity Doc*, and use the URL: <https://docs.splunk.com/Documentation/ITSI/latest/Entity/EntityType>
Make sure to read the docs on this feature because it does a lot more than just add a URL.
63. Click **Save Navigation**.
64. In the Splunk dashboards section, attach an existing dashboard to the entity view. Select as a placeholder **ITSI Health Check** from the Dashboard name menu.

65. Click **Add**, and then **Save**.

If needed, you can select your custom Entity types, edit them, or delete them.

66. Return to **Configuration > Entities** to edit one of the www* servers and add “Custom *nix” in the Entity type field and click **Save**.

67. Click **View Health** for the entity you just modified.

68. In the upper right corner of the page, click the Open entity information panel icon  to see and test your navigation suggestion.

End of Lab Exercise Implementing-6

Lab Exercise Implementing-7: Templates and Dependencies

Description

In this exercise, you will first address the known association between Online Sales and the Web Farm service. Then, address a new requirement: The customer support team has learned about ITSI and wants you to implement a service for them!

Task 1: Make the Online Sales service dependent upon the Web Farm service.

1. Navigate to **Configuration > Services** and edit the **Online Sales** service.
2. Select the **Service Dependencies** tab and click **Add dependencies**.
3. Select the **Web Farm** service, and then the **ServiceHealthScore**.
4. Click **Done**.
5. Select the **Settings** tab and scroll to the bottom of the page to see the Dependent KPIs.
6. Confirm that the Web Farm's service health score is now a dependency of the Online Sales service with an importance of 11.
7. Click **Save**.

Task 2: Create a new service template.

The IT support team has a customer support portal running on a set of servers. They mainly want the same level of monitoring as the web IT team. In fact, the new Support service would be a copy of the Web Farm service but using different entities for the web servers. Looking ahead, you can see that there are other web resources that would likely also be added in the future, so this is a good time to implement a Web Farm service template.

First, you'll rename the existing Web Farm service to make it more identifiable.

8. Navigate to **Configuration > Services** and click **Web Farm** to edit the service.
9. Change the name of the service to: **Storefront Web Farm**
10. Click **Done**, then **Save**.
11. Navigate back to **Configuration > Services**.
12. For the **Storefront Web Farm** service, click **Edit > Create Service Template**.
13. Enter the title: **Web Farm Service Template**
14. Click **Create**.

You have created the template, and the template editor UI is opened. You can see that it contains the KPIs from the Storefront Web Farm service.

15. Select the **Linked Services** tab.

Notice the **Storefront Web Farm** service is a linked service. Any future changes you make to the **Web Farm Service Template** affects the Storefront Web Farm and any other linked service(s).

16. Select the **Entities** tab; note the entity rule from the Storefront Web Farm is replicated here. You don't want this template to select **www*** hostnames for all possible web farm services, so change the **matches www*** rule to **matches a value to be defined in the service**.
17. Click **Save**, and on **Save Service Template**, note first that you have options to overwrite linked services configurations (although you will leave these off) and also to schedule the changes for later (but you'll push it now to save time.)

18. Click **Save**.
19. Open the **KPIs** tab.
20. Select the **Errors** KPI and expand the **Search and Calculate** section.
Notice the KPI is now using a new base search and metric, which was automatically created when you created the service template.

Task 3: Create a new service using a service template.

Now you'll create a new service using the Web Farm Service Template.

21. Navigate to **Configuration > Services** and select **Create Service > Create Service**.
22. On the Create Service dialog box, enter the title **Support Web Farm**, and select **Link service to a service template**.
23. For **Link to template**, select **Web Farm Service Template**.
Note the list of entity rules, KPIs and other linked services.
24. Scroll down to **Settings** and check **Enable 7 days of backfill for all service KPIs**.
25. Click **Create** to create the new service.
The service editor UI is opened with the **Entities** tab selected. Note the rule has the blank value for matching host aliases and a yellow warning icon indicating the missing value.
26. Enter **supp*** for the **matches** value and press **[TAB]** to rerun the entity rule preview.
You should see a set of matching host names for **supp*** hosts in the preview list at the bottom.
27. Click **Save**, then **Save and Enable**.
Your new service is up and running! In a few minutes, you can check the default service analyzer to confirm the service is active and see the initial values for its KPIs.

End of Lab Exercise Implementing-7

Lab Exercise Implementing-8: Anomaly Detection and Predictive Analytics

Description

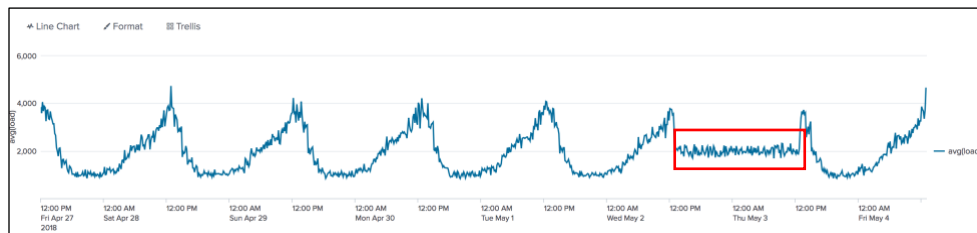
In this lab exercise, you'll add a new KPI to the Support Web Farm and enable anomaly detection. You'll also create some predictive analytics models for analysts to use on the Predictive Analytics dashboard.

Task 1: Add the ticketing load metric.

The support team has an additional metric to track: the load on their ticketing system. They want to be alerted if the load behaves "abnormally," so you'll add this KPI to the service and enable anomaly detection.

First, evaluate the source events. Talking to the support team, you learn that the required sourcetype is **support**, and in the support events, a field named **load** contains a value expressing the total load on the ticketing system.

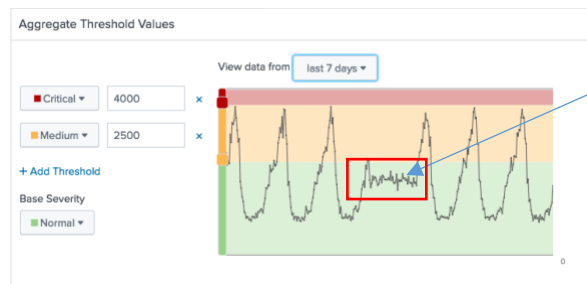
1. Run this search over the **Last 7 days** to examine the events:
`sourcetype=support | timechart span=10m avg(load)`
2. Viewing the results as a line chart visualization, you should see something like this:



Since these events have a regular pattern, it's a very good candidate for anomaly analysis. If the KPI deviates from this expected pattern, you'll get an alert. However, note the area in the red box (this should be in your data too—about 1 day before the start of your class). This looks odd; a day where there was no regular high-low cycle. The load level is not too high nor too low, so it wouldn't be caught by simple thresholding. But it is anomalous—it indicates something is interfering with your measurement of the ticketing system. You should be on the lookout for it in the future.

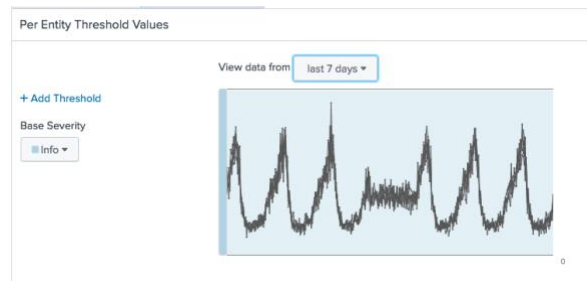
3. Run a search over all time on the **anomaly_detection** index. Note that it is empty—no anomaly detection is configured.
4. Edit the **Support Web Farm** service, and add a new generic KPI with the following settings (accept default values for all other parameters):

Title: Load
Search: sourcetype=support
Threshold field: load
Split by Entity: Yes
Filter to Entities in Service: Yes
KPI Search Schedule: Every minute
Backfill: enabled
Backfill period: Last 30 days (to provide a long baseline for analysis)
Aggregate Thresholds:



Notice the anomaly event is in the normal severity range, so it would not be detected by threshold monitoring.

Per Entity Thresholds > Base Severity: Info



5. Click **Finish** to add the new KPI and then click **Save** to update the service.

The backfill will now start and will take a few minutes to finish. When it completes, proceed to the next task. The backfill must complete to provide the necessary baseline for anomaly detection.

Task 2: Configure anomaly detection.

6. In the service editor, open the **Anomaly Detection** section of the **Load** KPI.
7. For **Analysis Time Window**, select **Last 30 days**, and leave the algorithm sensitivity at 8.
Note: If you receive a timeout error, reduce the time window to Last 7 days.
8. Click **Analyze KPI Data**.

The AD system audits the existing KPI data in **itsi_summary**, looking to see if the **alert_values** for the **Load** KPI indicate a usable statistical model for the algorithms used for anomaly detection. Both trending and entity cohesion are tested. Since this KPI has more than 3 entities, it is a candidate for both entity cohesion analysis and trend analysis.

The analysis result should return with a recommendation to enable anomaly detection for both the trending and cohesion models.

If you get no recommendation (it can happen), you can still enable both AD algorithms.

9. Expand the analysis breakdown and note the anomalies the algorithm found in the past.

This indicates a departure from the normal 24 hour pattern for this metric. If this happens again in the future, AD will alert with a notable event.

10. Enable the **Trending AD** and **Cohesive AD** algorithms.

11. Leave the sensitivity sliders set to 8.

This is a good general setting for AD; in practice, you can adjust this up or down as needed to deal with false positives or false negatives.

12. **Save** the service.

13. Wait a few moments, then run this search over the **Last 24 hours**:

```
index=anomaly_detection
```

You should see events created here beginning from when you configured AD on the Load KPI. Two sources should be present: `itsi_mad_context` and `itsi_mad_cohesive_context`. The `alert` field is **true** for detected anomalies.

You might see notable events created, if any new anomalies are detected.

Task 3: Configure predictive analytics.

14. Navigate to **Apps > Manage Apps** and filter for **toolkit**. The **Machine Learning Toolkit** is installed and enabled.
15. Filter for **scientific**. The **Python for Scientific Computing** app is installed and enabled.
16. Navigate back to the IT Service Intelligence app.
17. Navigate to **Dashboards > Predictive Analytics**.
18. Select the **Online Sales** service.


Note there are no predictive models available yet for Online Sales. You'll need to add a predictive model for this service. Because predictive analytics requires a historic baseline of service health score data as an input, you'll first need to backfill the service health score of Online Sales. Before you do that, you should also backfill the service health score for the Storefront Web Farm service, because Online Sales depends on that service.

19. Navigate to **Configuration > Services** and edit the **Storefront Web Farm** service.
20. On the **Settings** tab, enable the **Backfill** option. Select **14 days** for the backfill period and **Save** the service configuration.
21. Wait for a few minutes until a system message appears to confirm the service health score backfill is complete.
22. Repeat the previous steps to backfill the **Online Sales** service health score. After the backfill for the Online Sales service health score is complete, proceed with the rest of the lab.
23. In the Online Sales service configuration, select the **Predictive Analytics** tab.

Note the two top graphs, one showing values for the service health score and KPI values over time, the other showing the distribution of service health score values from 0 to 100. It may be difficult to see all the KPI and health score values—the scaling on the chart may leave all but the **Volume** KPI out of view at the bottom of the chart.

24. Leave **Time Period** set to **Last 14 days** and **Split for Training/Test** at **70/30**.
25. Use the **Algorithm Type** and **Algorithm** selectors to select each of the 4 model types one at a time, and then use the **Train** button to train a model for each of the 4 algorithms.
26. After training all four models, scroll down to the **Test a Model** section.
27. In the **Regression Models** panel, click the name of each model to see its metrics.

Note that one of the models has a check mark in the **Recommended** column. This model shows the best predictive behavior.

28. **Save** the service.
29. From the **Regression Models** panel, select the recommended model.
30. Scroll down to the **Predicted Worst Case Service Health Score** panel and click the **Create Correlation Search**  icon.

31. Accept all default values and click **Create**, then **OK**. You will now be notified with an alert in **Alerts and Episodes** indicating the Online Sales service is predicted to hit or drop below a threshold within 30 minutes.

32. Navigate to **Dashboards > Predictive Analytics**.
33. Select the **Online Sales** service.
34. In the **Model** selector, note the models are now available for analysts. Select the **Recommended** model.
35. The 30-minute prediction for the health score is shown.

Optional Task: Validate predicted health scores.

After about 30 minutes you may have some predictive analytic notable events generated, depending on your sample data, but it may take an arbitrary amount of time for the conditions that would generate the events to come to pass. Later during the course, you can take these steps to check.

36. Navigate to **Alerts and Episodes**, and search for episodes generated by predictive analytics.
37. If an episode appears, click the episode and select the **Common Fields** tab.
38. Examine the current value(s) of the **next30m_worst_hs** field. The predicted health score value(s) is/are lower than the threshold value defined in the notable event title.
39. Navigate to the Online Sales service's default deep dive, select Last 60 minutes for the time range. Do the actual health scores match the predicted health scores?

End of Lab Exercise Implementing-8

Lab Exercise Implementing-9: Service Access Control

Description

In this lab exercise, you'll create a Sales Operations team to restrict the Online Sales service to only the sales team.

Task 1: Restrict the Online Sales service to the Sales Operations team.

1. Navigate to **Settings > Roles**.
2. Create a new role as follows:
 - Role name:** salesops_admin
 - Inheritance:** itoa_team_admin
 - Default app** (on the Resources tab): itsi
3. Click **Create**.
4. Create a new role as follows:
 - Role name:** salesops_analyst
 - Inheritance:** itoa_analyst
 - Default app** (on the Resources tab): itsi
5. Click **Create**.
6. Navigate to **Settings > Users**.
7. Edit **analyst1**. This will be your test Sales Ops analyst.
8. Add **salesops_analyst** to the list of selected roles.
9. Click **Save**.
 - A matching password must be entered in the Set password and Confirm password fields to save the role change.
10. Navigate to the IT Service Intelligence app, and select **Configuration > Teams**.
11. Click **Create Team** and configure as follows:
 - Title:** sales_ops
 - Give **salesops_admin** read and write privileges
 - Give **salesops_analyst** read privilege
12. Click **Create**.
13. Navigate to **Configuration > Services** and, for the **Online Sales** service, select **Edit > Edit Team**.
14. Change the team setting to **sales_ops**.
15. Click **Save**.
 - Now only members of the two **salesops** roles can access the service, and only the **salesops_admin** role (or **itoa_admin**) can modify the service.
16. Test this by starting a new browser session as **analyst1** (either logout/login again or use a second private/incognito browser window. Your instructor will supply passwords).
 - You should see the **Online Sales**, **Storefront Web Farm**, and **Support Web Farm** services in your Service Analyzer.
17. Log in as **analyst2**.
 - You should only see the **Storefront Web Farm** and **Support Web Farm** services, as well as their KPIs, but not the **Online Sales** service. If you were to create a new glass table, you would not see any KPIs from the **Online Sales** service.

End of Lab Exercise Implementing-9

Lab Exercise Implementing-10: Backup, Content Packs and Maintenance Mode

Description

In this lab exercise, you'll modify the default daily backup, manually back up the KV store, download the archive to your workstation to examine its contents, upload a content pack, and put the Web Farm service into maintenance mode.

Task 1: Modify the daily backup schedule and initiate an immediate backup of the ITSI KV store.

1. Make sure you're logged in to Splunk as **admin**.
2. Navigate to **Configuration > Backup/Restore** in the ITSI app.
Note there is one job, **Default Scheduled Backup**. The last backup start and end timestamps are shown, along with the status, **Scheduled Daily**.
3. For the default job, select **Edit** dropdown menu.
Note that you can download the most recent scheduled backup, and also restore from the most recent backup. You can also disable the scheduled backup or edit its settings.
4. Select **Edit > Edit**.
You can alter the scheduled backup name and description, and also change the schedule to be daily or weekly.
5. Change the schedule to **Weekly**, on **Friday** at **23:00**, and click **Save**.
The scheduled backup will now run at the end of each week.
6. To run an ad hoc backup, click **Create Job > Create Backup Job**.
7. Name the job: **Back It Up**, and leave all other options enabled.
8. Click **Create**.
9. Refresh the web page until the Status changes to **Completed** for the **Back It Up** job.
10. Select **Edit > Download Backup** to download a zip file of the backup.
Note: If the **Download Backup** option is not active in the Edit menu, reload the web page.
11. Expand the zip file on your workstation.
You'll see a set of JSON files storing the ITSI configuration for the server as well as a **conf** folder containing ITSI configuration files.
Note: You could now use this backup package to perform a restore on a different ITSI server.

Task 2: Upload a Content Pack.

12. Open the documentation page at:
docs.splunk.com/Documentation/ITSICP/current/Config/ConfigShared
13. Under **Install the content pack on an on-premises instance**, use the link to download `BACKUP_CP-SHARED-INFRA-1.3.1.zip` to your local workstation. This is a partial backup containing the shared IT infrastructure content pack.
14. In ITSI, navigate to **Configuration > Backup/Restore**.
15. Select **Create Job > Create Restore Job**.
16. Enter **Install Shared IT Infrastructure Content Pack** for the **Name**.
17. For **Backup File**, browse to the zip file you downloaded previously.
18. Click **Create**.

The restore job begins to run. This can take several minutes. When it is done, you'll see a system message confirming the installation.
19. Navigate to **Service Analyzer > Analyzers** and click on **IT Infrastructure**.

You'll see the infrastructure model as a set of dependent services. It may take a few minutes for the KPI searches to run and set the service health scores.
20. Navigate to **Glass Tables**. You'll see a new glass table named **IT Infrastructure Health**.
21. Navigate to **Configuration > Services** and note that many new services have been installed.

These are all part of the Shared IT Infrastructure model. They each contain one KPI named **Heartbeat**, that currently always returns a normal alert severity. To use the model, you'd add the required KPIs to each service according to the monitoring needs in your organization. You could also rename, add or disable services as necessary. The content pack is just a starting point.

Task 3: Put the Storefront Web Farm service into maintenance mode and review the ITSI Health Check and Event Analytics Audit dashboards.

22. Navigate to **Configuration > Maintenance Windows**.
23. Click **Create Maintenance Window**.
24. Configure the parameters as follows:
 - Title:** Update storefront web servers
 - Start Time:** leave the default (current time)
 - Duration:** 4 hours
 - Objects:** Services
25. Click **Next**.
26. Select the **Storefront Web Farm** service (page 2 of services) and click **Create**.
27. Click **Back to Maintenance Windows**.
28. Click **Update storefront web servers**.
29. To see a list of KPIs impacted by this maintenance window, select the **Impacted KPIs** tab. Keep refreshing the page until the KPIs display.
30. Navigate to the default service analyzer.
31. Use the Filter Services field to display only the **Storefront Web Farm** service.
32. Select tile view.
 - If the **Storefront Web Farm** service and its KPIs have not already changed to grey, refresh the page until they do.
 - Great news, the storefront web server update has completed early!
33. Navigate back to **Configuration > Maintenance Windows**.
34. For the **Update storefront web servers** maintenance window, select **Edit > End Now**.
35. Click **End Now**.
36. Navigate to **Dashboards > ITSI Health Check**.
37. Review the dashboard panels.
38. Navigate to **Dashboards > Event Analytics Monitoring**.
39. Review some of the dashboard panels.
40. Navigate to the default service analyzer to confirm the Storefront Web Farm service has resumed normal operation.
 - It will take a few minutes for the **Storefront Web Farm** service and its KPIs to return to normal.

End of Lab Exercise Implementing-10

Lab Exercise Implementing-11: Implementing Correlation Searches and Multi-KPI Alerts

Description

The customer has asked for the following notifications:

- The IT team wants to be alerted if the number of available web servers falls below a certain level.
- The sales operations team wants to watch the conversion rate. This is roughly computed as the ratio of page views to purchases. High page views correlated with low purchases indicates there's something wrong with the marketing or messaging on the web site. This should be a high severity notable event.
- The anomaly detection you configured for the support ticketing system load KPI might generate notable events.

Task 1: Create the web server alert for the IT team.

You will configure the correlation search to find any errors in a one-hour interval for testing purposes.

1. Select **Configuration > Correlation Searches**.
2. Click **Create New Search > Create Correlation Search** and define the Search Properties, Association, and Schedule as follows (accept all other defaults):

Search Name: Storefront Web Farm Service Level

Description: Too few servers are running to support the web site

Search Type: Ad hoc

Search:

```
sourcetype=access_combined host=www*
| eval service_level = 4
| stats dc(host) as num_servers, last(service_level) as
service_level
| where num_servers < service_level
```

This search is intentionally designed to raise alerts immediately, since there are only three "www" servers in the storefront web farm. But for now, you'll use this for testing to ensure notable events are generated.

Time range: Last 60 minutes

Service: Storefront Web Farm

Schedule Type: Basic

Run Every: minute

3. Notable Events settings (accept all other defaults):

Notable Event Title: There are only %num_servers% servers running in the storefront web farm

Notable Event Description: Minimum service level is %service_level% servers

Severity: High

Drilldown Search Name: Examine events

Drilldown Search: sourcetype=access_combined

Drilldown earliest offset: Last 60 minutes

Drilldown latest offset: Next minute

4. Click **Save**, and **OK**.

5. The correlation search begins to run. After a few minutes, you can check **Alerts and Episodes**. You should have an episode for this correlation search. You may also have an episode for the predicted health of the Online Sales service, from the previous lab exercise.

Task 2: Create the multi-KPI alert for the online sales team.

A multi-KPI alert is well suited for the online sales operations team's needs. They want to raise an alert when purchases are low but views are high.

6. Select **Configuration > Correlation Searches** and select **Create New Search > Create Multi-KPI Alert**.
7. Click **Status over time** in the upper right corner.
8. Select **Last 60 minutes** for the search time range.
9. Select the **Online Sales** service in the Services panel.
The service's KPIs appear in the KPIs in selected services panel.
10. Click the **+Add** link for the **Purchases** KPI to add it to the **Selected KPIs** panel.
The Triggers for Correlation Search dialog box displays.
11. Set the following trigger conditions, removing others if set:
Critical $\geq 1\%$
High $\geq 50\%$ (make sure all others are unchecked)
12. Click **Apply**.
13. Repeat the above steps for the **Views** KPI, using these trigger values:
Normal $\geq 1\%$
Low $\geq 25\%$ (make sure all others are unchecked)
To summarize: "Over the last 60 minutes, if Views are good (alert_level = normal or low), but Purchases are bad (alert_level = critical or high), raise an alert."
14. Click **Save**. The **Create Correlation Search** dialog box appears. Configure as follows:
Search Name: Conversion rate
Notable Event Title: Low conversion rate
Notable Event Description: %event_description% (auto-populates the description field)
Schedule Type: Basic
Run Every: minute (for testing)
Severity: Medium
15. Click **Save**.
Now to test this correlation search, you'll need to set up a test condition. Right now your data sources from the website logs are running in normal mode, so views and purchases are not showing a large difference, so the conversion rate correlation will probably not generate a notable event very often. To test it, you'll temporarily alter the base search.
16. Edit your **BCG: Web Purchases** base search.
17. Modify the **search** field:
`sourcetype=access_combined action=purchase | eval test_value = 0`
18. Modify the **purchase_count** metric:
 - **Threshold Field:** test_value
 - **Entity Calculation:** Average
 - **Service/Aggregate Calculation:** Average

19. Click **Done**, then **Save**.

This change causes the **purchase_count** to always be zero. You can use this to test conversion rate failure. In production, it's a good idea to create different base searches for test mode vs. live data. Note that it's also easy to clone a KPI or an entire service and use a new testing base search to test it, then delete the cloned objects when testing is complete.

20. Navigate to **Alerts and Episodes**.

An episode for the correlation search you created earlier should appear. It could take a few minutes for the Multi-KPI alert to generate an episode. You might also see episodes for the Online Sale's health score predictive analytics alert and/or the anomaly detection alert you configured in the previous lab exercise.

End of Lab Exercise Implementing-11

Lab Exercise Implementing-12: Aggregation Policies

Description

In this lab exercise, you'll work with aggregation policies to achieve some of the customer's requirements:

- The IT team wants to be alerted when errors happen in the storefront web farm.
- They work on these errors differently depending on error type, so they want to be able to group by type of error in the Alerts and Episodes display.
- Errors during customer purchase activity are critical, so the severity for these notables needs to be increased.

Task 1: Add a new correlation search for storefront web farm errors.

1. Create a new correlation search with the following settings, leaving other settings at the default:
 - Search Name:** Storefront Web Farm Error
 - Search:** sourcetype=access_combined action=* status >= 400 host=www*
 - Time Range:** Last 60 minutes
 - Service:** Storefront Web Farm
 - Entity Lookup Field:** host
 - Note:** After you enter this value, the search field is appended with additional qualifiers to change the host field to entity_title, and suppress this correlation search for entities in maintenance mode.
 - Run Every:** minute
 - Notable Event Title:** Storefront Web Farm Error on %entity_title%
 - Notable Event Description:** Error %orig_status% occurred on server %entity_title% during %action% process
 - Note:** The Status field becomes orig_status in the notable event, because status is used for notable event state.
 - Severity:** Medium
 - Drilldown Search Name:** Storefront Web Farm Events
 - Drilldown Search:** sourcetype=access_combined
2. **Save** the correlation search.
3. Navigate to the **Alerts and Episodes** page.
4. Refresh the page until a Storefront Web Farm Error episode appears.
5. Click **Add Filter** and select **Status**, then **New**.
6. In the search box, type: **Storefront Web Farm Error**
7. Save this as the **Storefront Web Farm Errors** view. You can use this view to focus on new notable events.
8. Select the **Storefront Web Farm Error** episode when it appears.
9. Note the list of impacted entities on the **Impact** tab. Notice the latest error code for each entity is displayed.
10. Click the **Common Fields** tab and expand the list of values for **orig_status**. These are the status codes being returned. There are a range of 4xx and 5xx error codes. Different error types are triaged differently by the IT team, so they'd like to be able to group the notables by error type.

Task 2: Create an aggregation policy to group notables from the web farm error correlation by error type.

11. Navigate to **Configuration > Notable Event Aggregation Policies**.
12. Click **Create Notable Event Aggregation Policy**.
13. Under **Include the events if**, click **+ Add Rule (OR)**.
14. In the field name (Select...), type: **source**
15. Make sure **matches** is selected as the operator, in the right-hand field, type: **Storefront Web Farm Error**.
16. Use the **Preview results** button to update the preview panel.
17. In the **Split events by field** section, remove the default value (host) and type **orig_status**.
18. Update the preview. It should now show one row per value of **orig_status**.
19. In the **Break episode** section, remove the default (If an event occurs for which severity = Normal).
20. Click **+ Add Breaking Condition (OR)**.
21. From the **If** dropdown menu, select **the number of events in this episode is**.
22. In the **count limit** field, type: **10**
23. Update the preview. It should now show no more than 10 events in any group.
24. Click **Episode information** to expand the section.
25. For **Episode Title**, select **Static value** and type: **Storefront Web Farm: Error %orig_status%**
26. Update the preview.
27. Click **Next** to display the Action Rules page. You'll change this later. For now, click **Next** to display the **Policy Info** page.
28. In the **Policy Title** field, type: **Storefront Web Farm Errors by Type**
29. Make sure **Status** is **Enabled**, and click **Next**.
Your policy is created and saved.
30. Wait a few moments, then navigate to the **Alerts and Episodes** page.
31. Update the search field to: **Storefront Web Farm: Error**
32. Click **Save** to update the Storefront Web Farm Errors view.
Pre-existing notable events will not be re-grouped, but new Storefront Web Farm Error notable events should now start being grouped by error type. You might need to refresh the page a few times.

Task 3: Define an automated action to escalate notable severity for errors in storefront purchase workflow.

The customer has one additional requirement: Errors that occur during the purchase workflow are urgent and should be escalated to critical severity.

33. Navigate back to the **Notable Event Aggregation Policies** configuration page.

34. Click **Storefront Web Farm Errors by Type** to edit the policy.

35. Select the **Action Rules** tab.

36. Click **+ Add Rule**.

37. From the **If** dropdown menu, select **the following event occurs**.

38. From the **Select...** dropdown menu, select the **action** field.

39. Tab to the **Matches** value field and type: **purchase**

This rule triggers now for any notable event where the **action** field is **purchase**.

40. In the **Then** pane, make sure **change severity to** is selected (but have a look at other options in that list) and change the severity value from Info to **Critical**.

41. **Save** the edited policy.

42. Navigate back to **Alerts and Episodes**.

Any new notable events for storefront web farm errors that happen during a purchase activity will now be set to critical severity. It may take some time before an example of this scenario is generated.

End of Lab Exercise Implementing-12