# Anypoint Monitoring and Auomation

There are various component which helps us to monito mule application

# Visualizer

Visualizer – displays all the APIs deployed in a particular environment

It is useful especially if you have to manage 100s of APIs

It provides a graphical interface to view the application n/w

By default visualizer is not enabled

To Enable: First you need to assign roles

* Goto access management
* Then select Roles and add roles to users
* This enables user to have a visualization roles

There are three types of visualizer

1. Architecture View
2. Troubleshooting View
3. Policies View

## Architecture View

* Displays application network view of your organization
* i.e. we can see all the application in API-led connectivity diagram
* It helps to see how the traffic flow from users to mule applications
* Displays interaction between APIs based on API-led connectivity
* Here,
  + Node – are the applications
  + Edges – are the direction of request
* These nodes and edges are dynamically created based on the real time traffic
* You can define the type of the layer the API belongs to
  + BY DEFAULT: these nodes are not intact with the particular layer
  + Option
    - You can manually link application to a layer
    - Or during the deployment you can add layer name and even add organization name
* This view is created based on the event



## Troubleshooting View

* This allows to view metric like the response time, avg response time, throughput, memory utilization, etc
* Used for trouble shooting
* Eg identify which layers take more response time, memory and CPU
* It also has an option to goto Monitoring, dashboard, and Logging page

## Policies View

* This option allows to view all the policies configured in different APIs
* It is a central place to vie the Policies implemented in your application n/w
* Adv: easy to track and monitors

Use-case of visualizer

* Architecture governance: helps to identify architecture violation
* Dependency management: if a sys api is change the helps to find out which APIs are dependent on the API – change
* Documentation

Deployments

* AVAILABLE IN hybrid, cloudhub and RTF based deployment

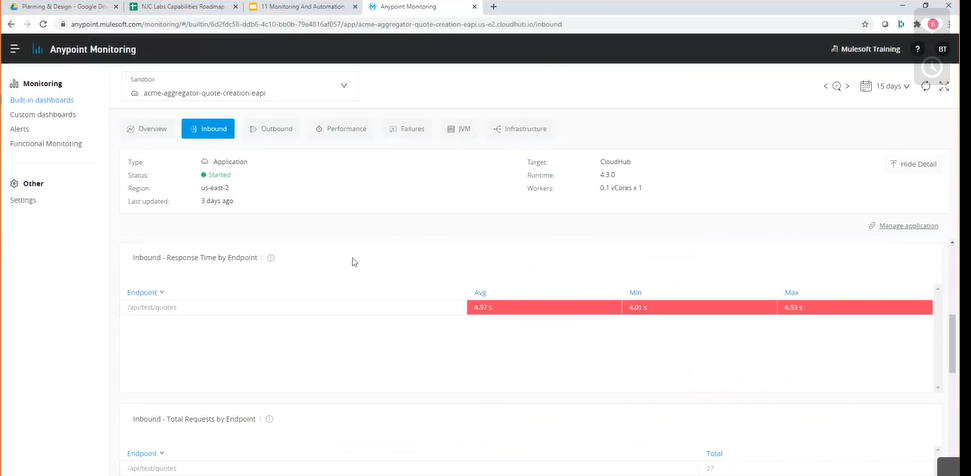
# Anypoint Monitoring

* Used to see the behavior of each application
* Available for all users for free
* Features:
  + Build in dashboards
    - Two option

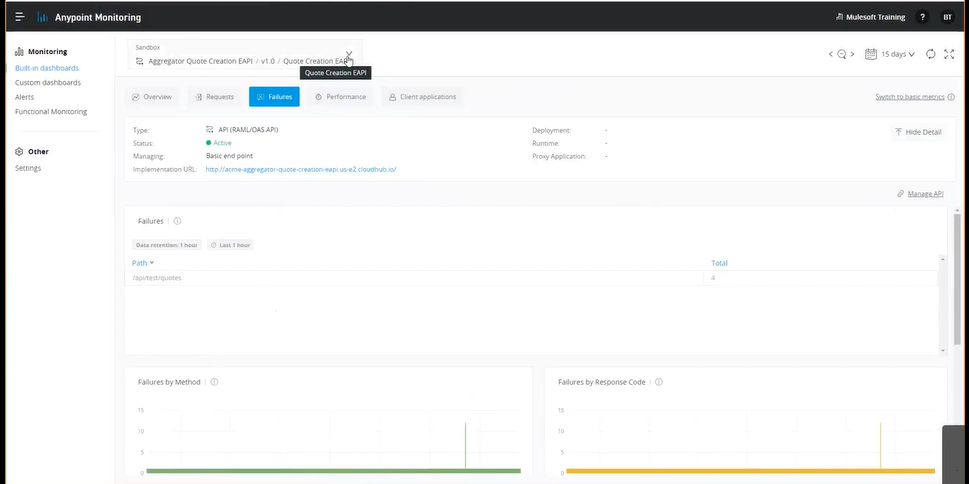
1. Application related
2. API Specific : in addition it shows the instance ID along with api name
   * Create custom dashboards
   * Alert

# Build in Dashboards

## Application related

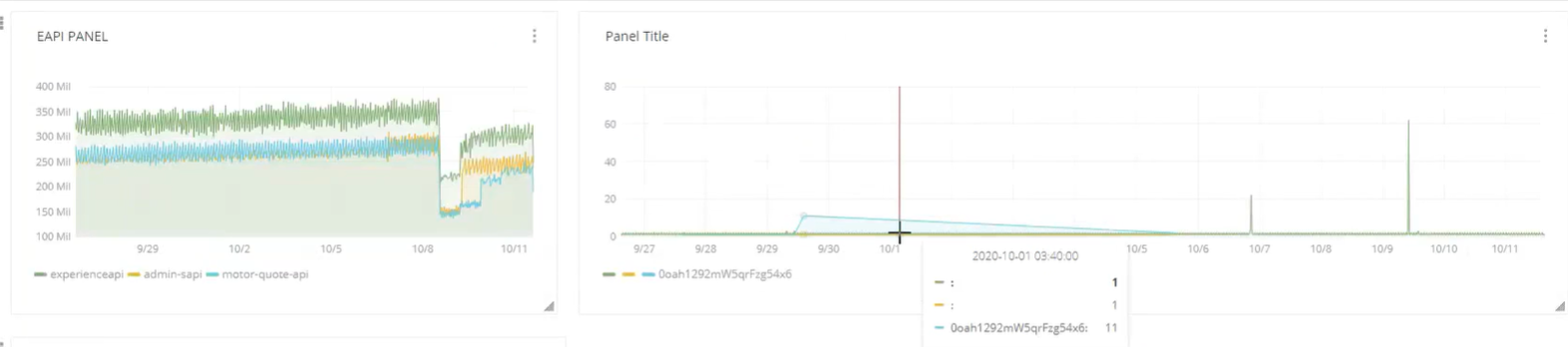
* Is based on **application perspective**
* Features: Overview, inbound, outbound, performance, failure, JVM, Infrastructure
* Overview you can see
  + Total response time
  + CPU utilization
  + Mule message its handling
* Inbound: show no incoming request
* Outbound: shows the application outbound response
* Performance: performance related metric like mule messages, avg response time
* Failure: different types of failures
* JVM: JVM related details like garbage collection count, garbage collection time, Class load/ unload time
* Infrastructure: CPU utilization, Heap utilization, Total system memory, JVM thread count
* 
* Figure 2 application specific

## API Specific

* Features: Overview, Request, Failure, Performance, Client application
* This have information based on **request perspective**
* Request: Resource you are requesting, request size, response time
* Failure: display failure by response code, failure by method, failure by response size
* Performance: avg, min and max time taken for a resource to respond
* Client application: number of applications that are trying to invoke a API
* 
* Figure 1 API specific view

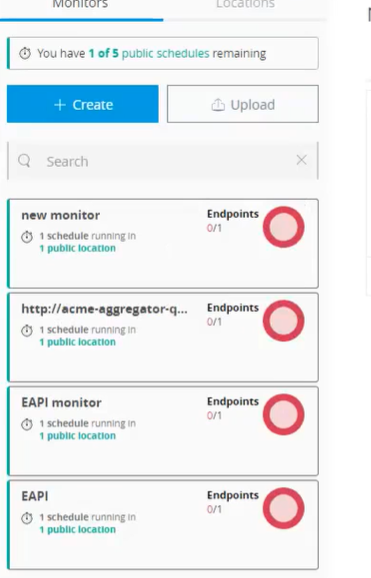
## Custom Dashboards

* We **can create custom dashboard**
* You can see details about your project, APIs or application n/w in one place
* Can add multiple panels
* What you can do
  + Display memory utilization of experience, process and system
  + No of request each API is handling
  + **Select type of graph to display, change graph size**
  + **For higher subscription like titanium:** you can generate alerts based on graph, change color scheme. Add multiple queries to get information about multiple APIs using advance component
  + Configure the panel
    - Basic mode: Application view or API view
      * Select API, version of API, environment



## 3. Functional Monitoring

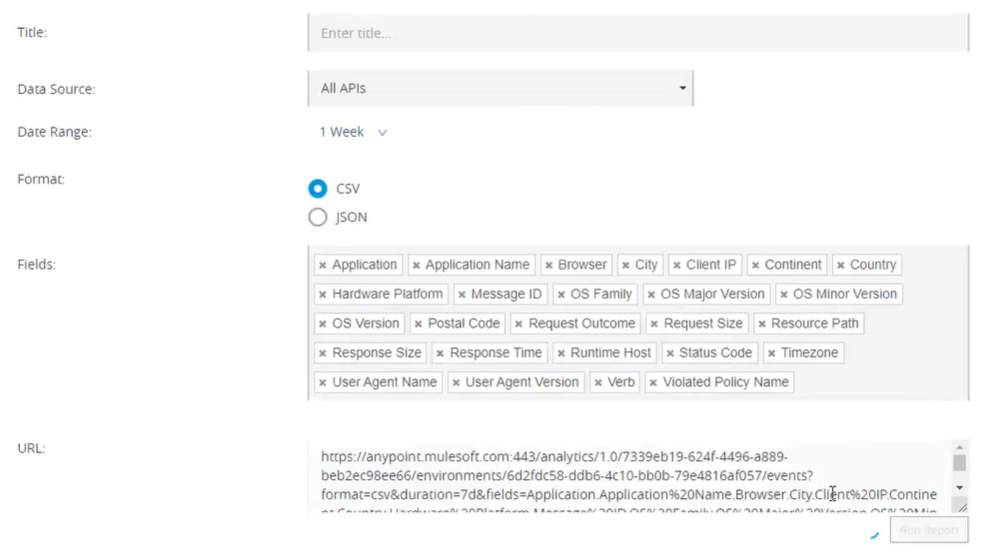
* + It helps to see what is going on in your APIs
  + It is a graphical representation about the health of APIS
  + Every API we create should have a health endpoint. It is a resource -> /resource. If it active it means URL is functioning
  + If the API in unhealthy it means there is API failure. This failure can be notified via alerts
  + WE can upload bad CLI scripts



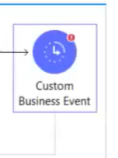
* + Anypoint bat (**Blackbox Automated Testing**) CLI
* It provides a playground to write and execute the test cases
* API call is made with in the scripts and result is shown in the playground itself
* Link: <https://anypoint.mulesoft.com/apitesting/playground/>
* You can upload those scripts in API monitoring and schedule a time for the script to execute. It can be every one hour, each day
  + TEST DRIVEN DEVELOPMENT
* It is a development approach
* In this first the test cases are written first and development is based on those test cases
  + BEHAVIOUR DRIVEN DEVELOPMENT
* It is a agile method
* First the business scenarios are created by the business owners written in plain english
* Developer matches the secenario and create test cases
* It ensures that all the business scnarios are used
* Example project: UI based project like portal app, website

## Alerts

# API Analytics

* It another ways of monitoring API
* It is found in API manager
* Need to be purchased
* It consist of custom dashboard, manage charts, manage reports
* You can **view request by date, location, application, from where API are executes**
* In this you have option to create a **custom dashboard**
* **First** you need to create charts and **add to the custom** dashboard
* charts are based on the parameters request size, no of request, response, response size
* **Manage Report** give information about how the APIs are performing
* **Like: when the APIs are called, location of requester: county, continent**
* 
* **Platform API can be used to get the same detail**

## Insights

* If you enable it
* It will show all information about the event
* Like: what component has error, time it was used, error details
* It used to **trace the application**
* It **should not be enabled in production** because it generated lots of event messages
* Cause
  + Application may be slow
  + OH is associated with it
  + Use high band width
* Used for short interval like root cause analysis .. API is not working in particular time of the day
* Business data component:
  + U need to adda business event component in studio.
  + Add custom information about the business
  + 

# Alerts

* This is used **to make the proactive decisions** in production
* All alerts work by sending Email
* Production can have different kinds of support
  + **Reactive support** 
    - When business team support application they can encounter different issues.
    - Wait for someone to raise the issue in the portal and the fix the issue
  + **Proactive Support**
    - Platform helps to identify the issue
    - Example: High CPU utilization send an alert message

Ways to configure the Alerts

1. Runtime Manager
2. API Manager
3. Application Alerts

## Runtime Manager

* In this it has
  + Above 80% CPU usage
  + Create application level alerts
  + You can select server type: deployment cloudhub or hybrid based, servers
  + Based on the server type chosen the configuration also changes

## API Manager

* Alerts are based on parameters like: Request count, Request Code, Policy validation, Response Time
* These are API level alerts
* Example: if high request are recived more than expected then alert the developer so they can check the reason of high traffic

## Application Alerts

* It is seen in runtime manger -> Alarm
* In studio, we have cloud hub connector,
* **Create notification is used to generate alert with in the platform**
* **Only works for cloud hub**
* This Create notification
* When you send a notification from your application to a runtime manager in cloud hub
* Such application can push this notification to the control plane and display as alter
* **Only support the application deployed to cloud hub**
* The alerts generated by cloud hub can be cached by using the method **Custom Application Notification**

When should developer goes for any Alert method

This alert happens from the control plane.

# Automation-

* There are two ways to perform automation

1. Basic automation
2. Advance Automation

## Basic automation

* Every mule application have mule maven plugin
* **Configuration depends on the type of deployment**
* For cloud hub you need to provide information like: --- in configuration
  + User Id and password of the anypoint platform
  + Region to deploy
  + Worker size and number of workers
* Most project are done using MAven

## Advance Automation

* Suppose you have configured all you APIs and policy sandbox. When youhave completed the testing of the application in dev envi, then we need to send it to production environment
* Instead of deploying all the APIs one by one… You can **save time by deploying** it in one go
* API Manager API – It is a platform API
  + It allows to automate many things
* Anypoint CLI
  + It is a non -JMS based API
  + It should have