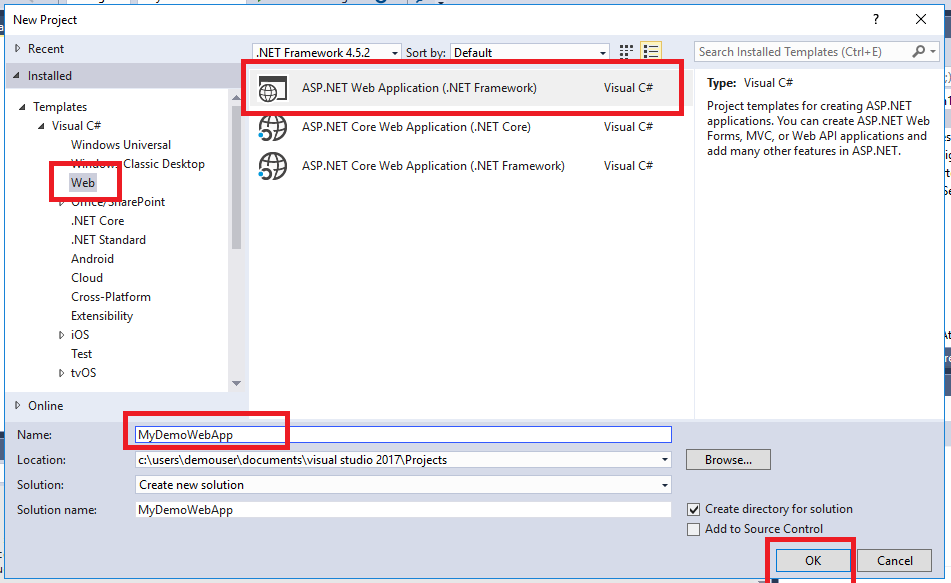
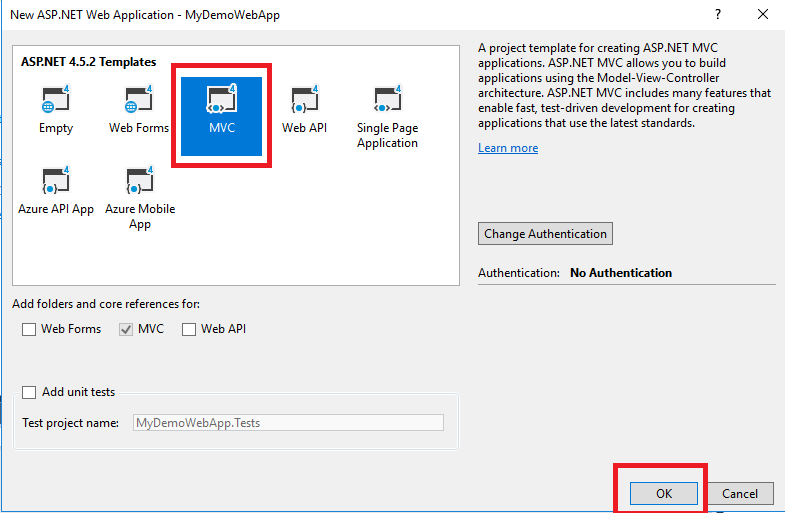
**What is Application Insights?**

Application Insights is an extensible Application Performance Management (APM) service for web developers on multiple platforms. Use it to monitor your live web application. It will automatically detect performance anomalies. It includes powerful analytics tools to help you diagnose issues and to understand what users actually do with your app. It's designed to help you continuously improve performance and usability.

Step: open visual studio

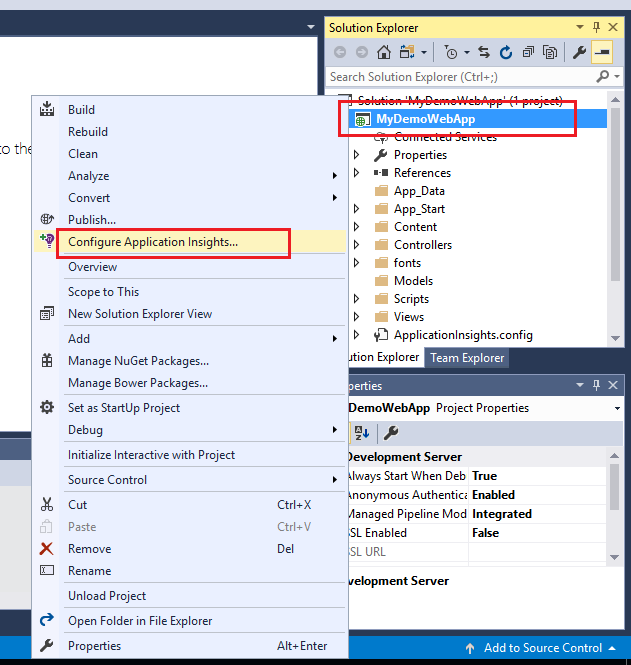
Click on File + New + Project.





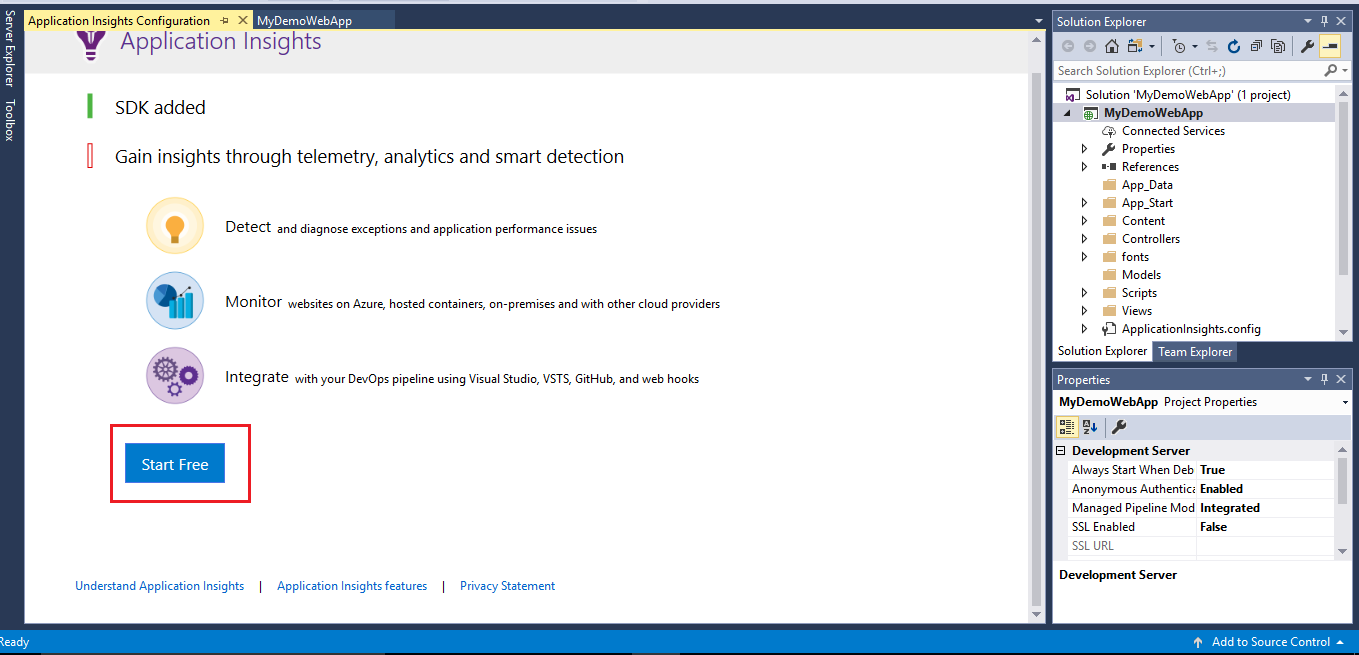
Step:

In the Solution Explorer, Right Click on the project name and select Configure Applicaton Insight.



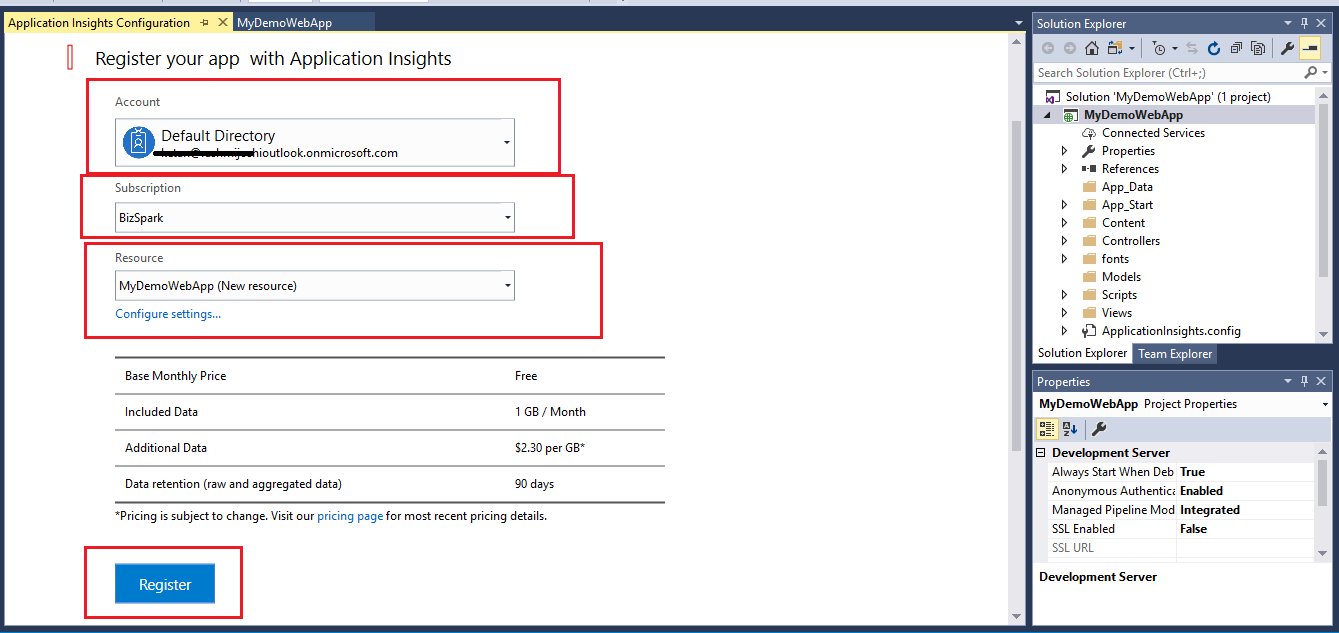
Step:

On the next page click on Start Free button.



Step:

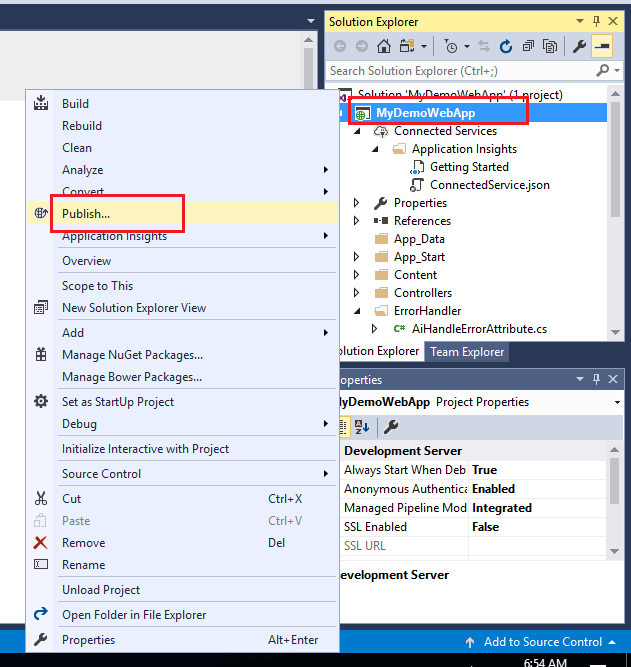
On next screen, Select or enter your azure user Id, subscription and Resource.



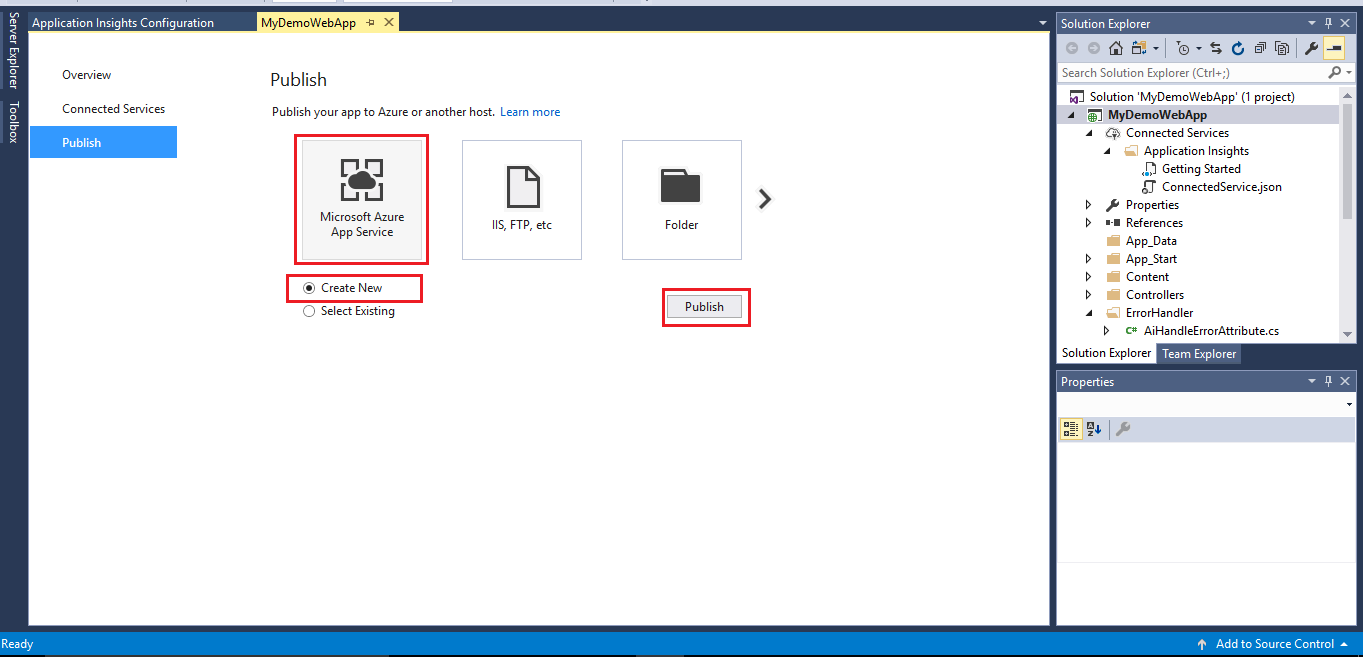
On the next screen click on OK button.

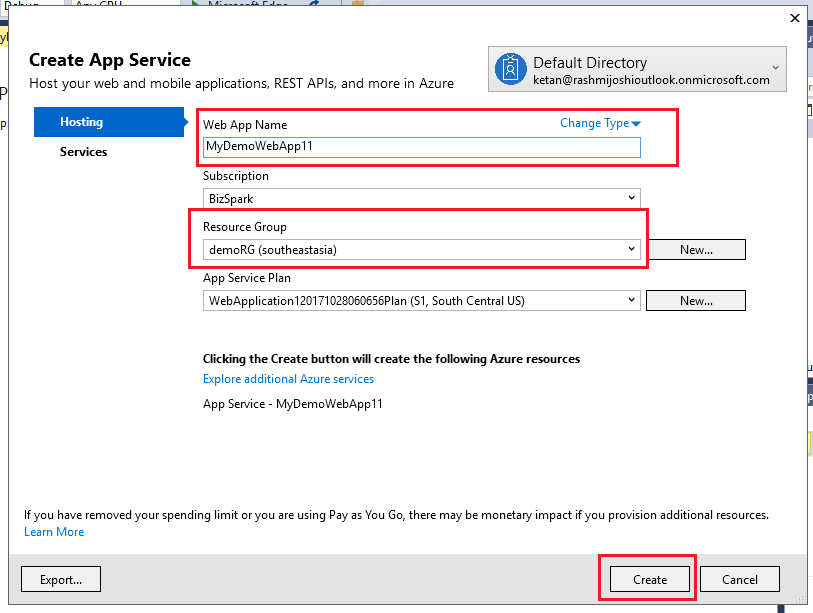
Step:

Now in the solution explorer, Right Click on Project Name and select Publish.



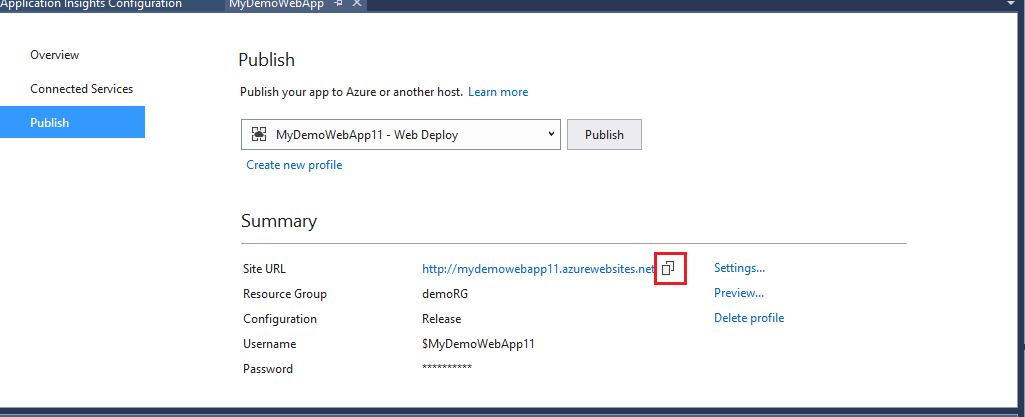
On the next page, Select Microsoft Azure App Service, Create New and then click on Publish button.





Step:

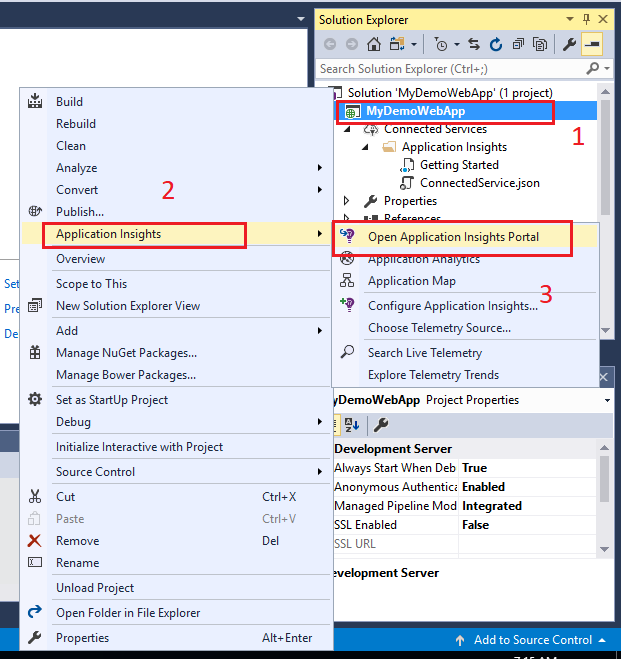
After completion of publish, copy the url of the web app and note down on notepad.



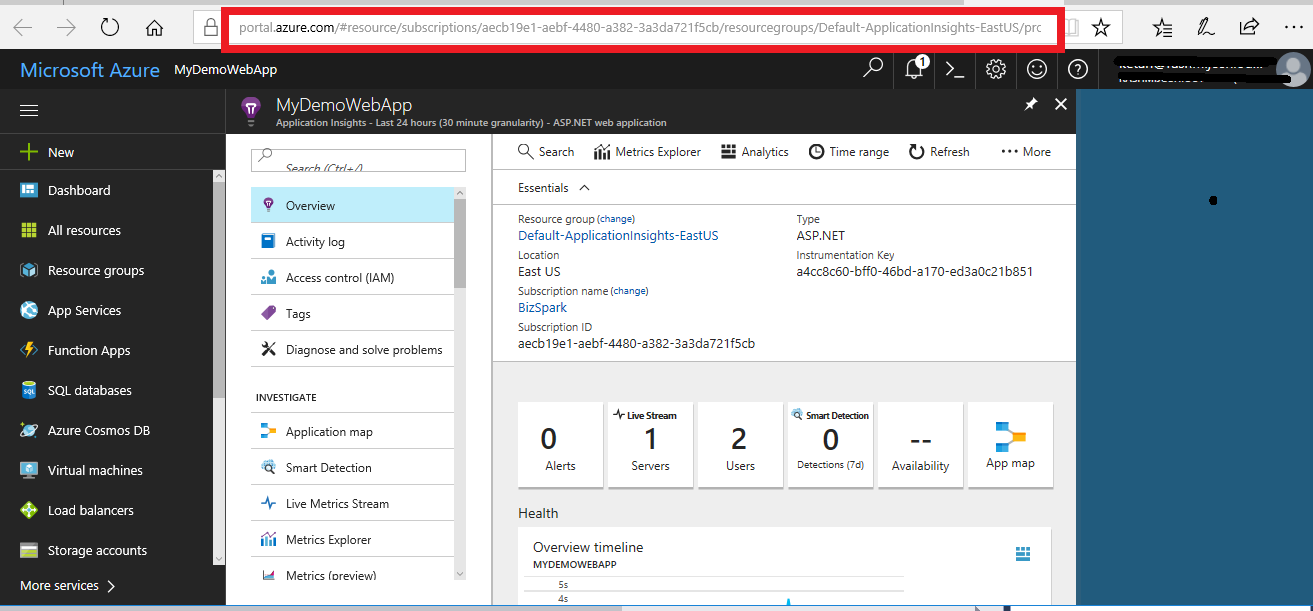
Step:

In the solution explorer, right click on project name then select Applicatio insight then select open

application insight portal.



When browser opens, Select and copy the url and note down on notepad. This is your application insight url for your web app.

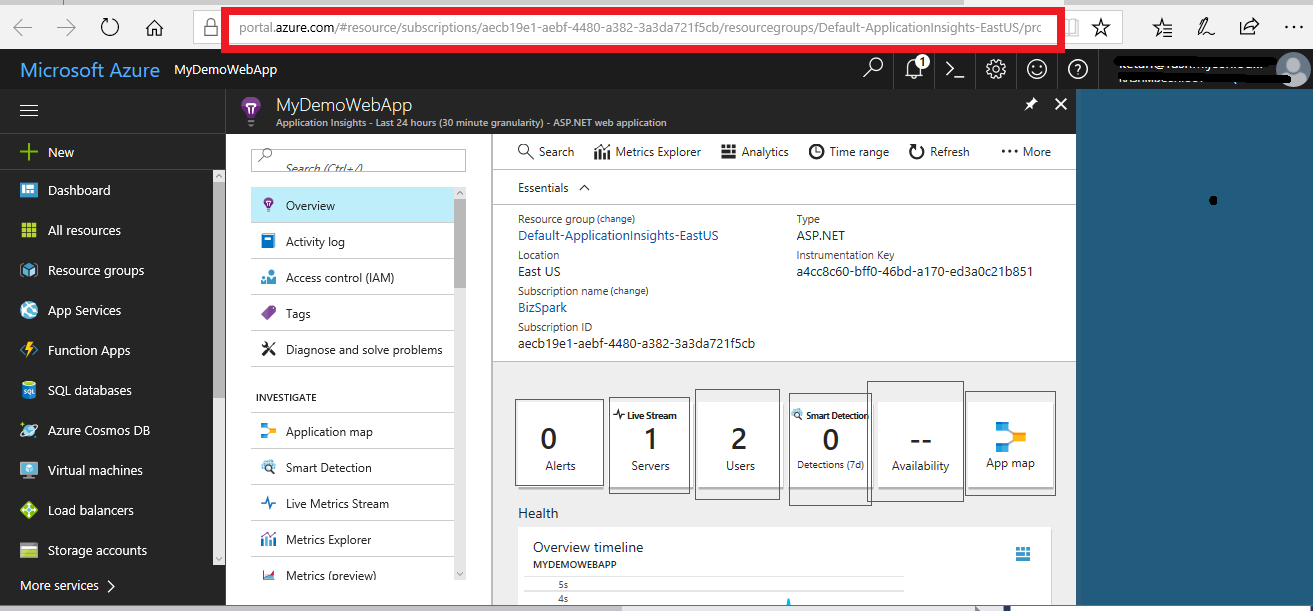


Step:

Open any web browser and open your web app using the web app url(copied on notepad).

Step:

Now in the web browser , enter url of your application insight(copied on notepad).



**Alert** : Application insight sends you an alert when something is wrong. You can define your own customized alerts.

**Live Stream**: Probe the beating heart of your live, in-production web application by using Live Metrics Stream from Application Insights. Select and filter metrics and performance counters to watch in real time, without any disturbance to your service.

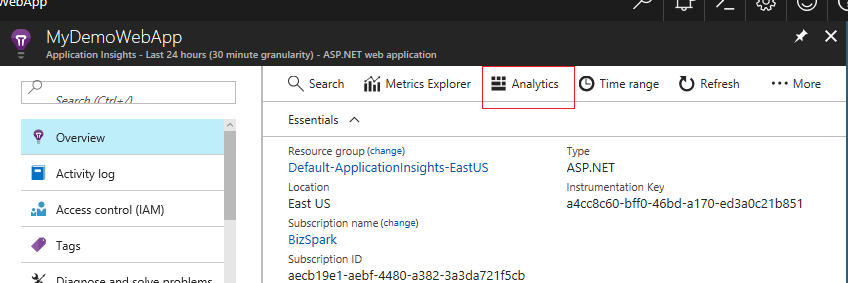
**Users** : you can monitor user and session activities.

**Smart Detection**: Smart Detection automatically warns you of potential performance problems in your web application. It performs proactive analysis of the telemetry that your app sends to Application Insights. If there is a sudden rise in failure rates, or abnormal patterns in client or server performance, you get an alert. This feature needs no configuration. It operates if your application sends enough telemetry.

**Availability**: You can check availability of your web app from different regions. You can add test and application insight will gives the availability checking.

**Applicaton** **Map**: Application Map is a visual layout of the dependency relationships of your application components. Each component shows KPIs such as load, performance, failures, and alerts, to help you discover any component causing a performance issue or failure. You can click through from any component to more detailed diagnostics, such as Application Insights events.

**Step**: Click on Analytics button, will open analytics blade. On this blade, you can analyse the application data using different queries.

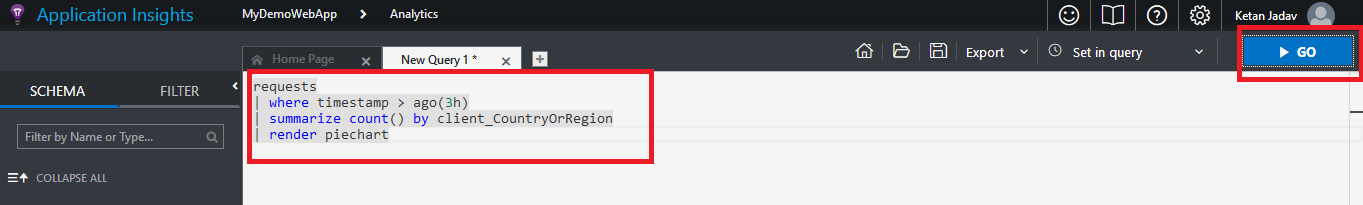


Analytics is the powerful search and query tool of Application Insights. Analytics is a web tool so no setup is required. If you've already configured Application Insights for one of your apps then you can analyze your app's data by opening Analytics from your app's overview blade.

You can query and analyse your application health and performance.

A typical query starts with a table name followed by a series of operators separated by |. For example, let's find out how many requests our app received from different countries, during the last 3 hours:

|  |
| --- |
| requests  | where timestamp > ago(3h)  | summarize count() by client\_CountryOrRegion  | render piechart |



More Queries –

requests | top 10 by timestamp desc

requests | sort by timestamp desc | take 10

requests

| summarize event\_count=sum(itemCount)

by bin(timestamp, 1h)

requests

| summarize count\_=sum(itemCount), avg(duration)

by bin(timestamp, 1h), client\_StateOrProvince, client\_City

| order by timestamp asc, client\_StateOrProvince, client\_City