This doc contains steps to install below appdynamics agents :

1) Servers

2) Clients

3) Databases

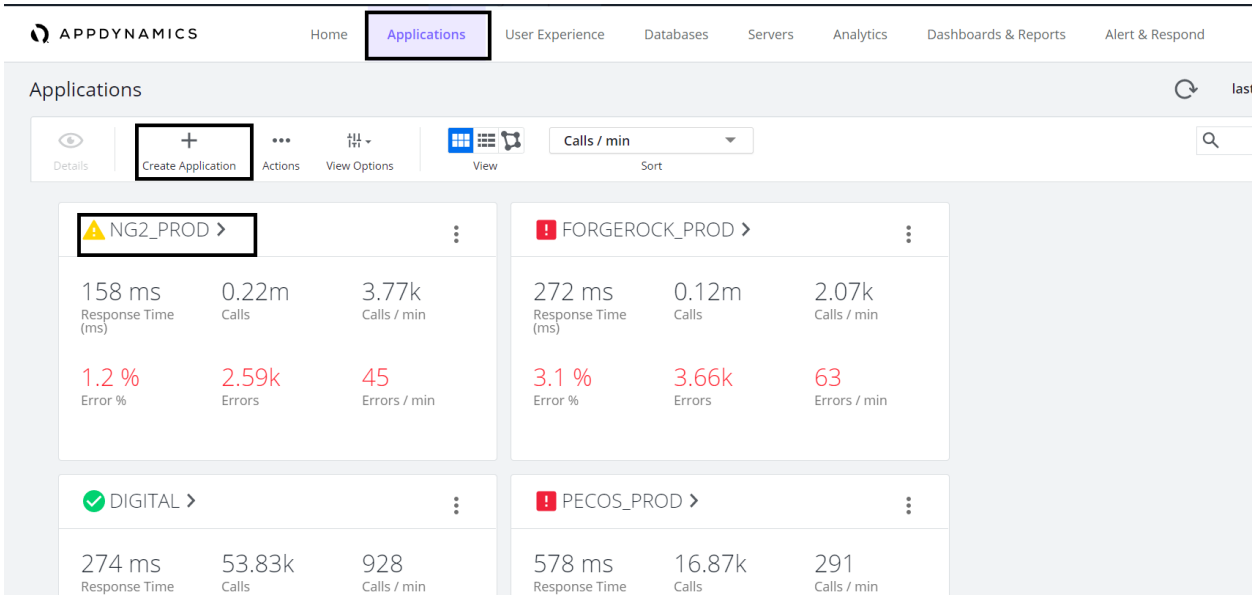
4) Analytics Agent

Servers:

To Install Appserver agent/ Machine agent / Network agent:

1) Before installing appd make sure to provide "**appd_application_name " [for server applications only [Appserver agent]** in the corresponding playbook to push metrics to respective Application in Appdynamics

2) If it needs to create under separate application then create new application as shown in the screenshot and then add the application name in the playbook



3) Run any playbook below to install agents

i) Appdynamics playbook
ii) Each playbook has appd.yml

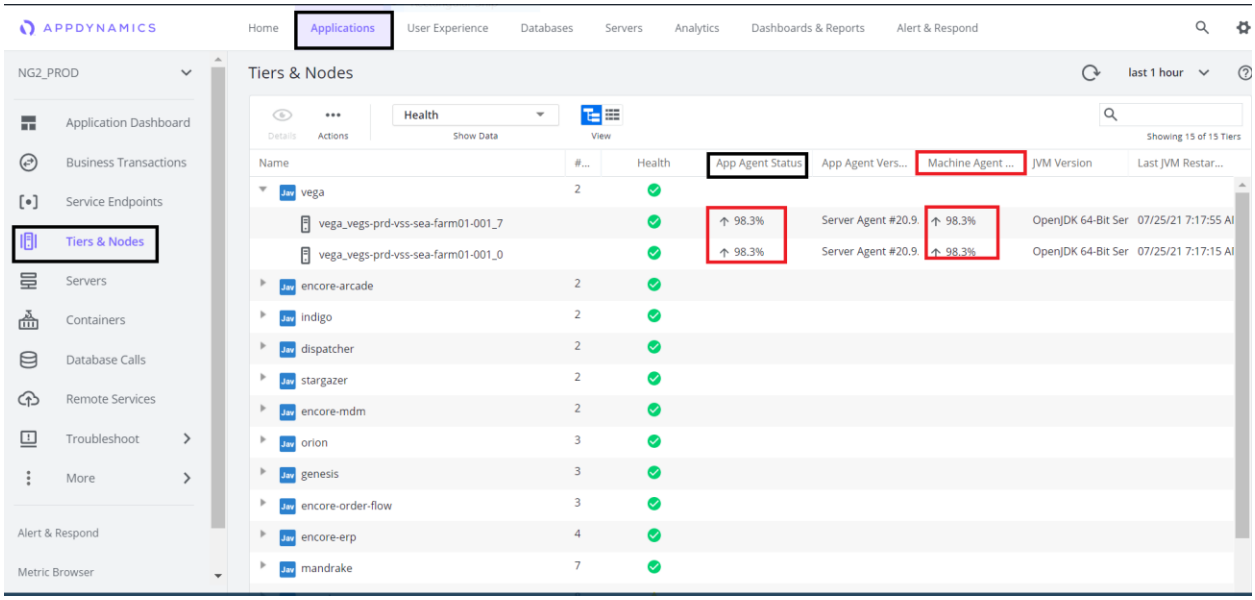
Ref:
/usr/bin/ansible-playbook -i <playbook-dir>/<env>_azure_rm.yml <playbook-dir>/appd.yml --limit <env> --extra-vars ng_server_env=<env> --vault-password-file ~/.vault_pass.txt

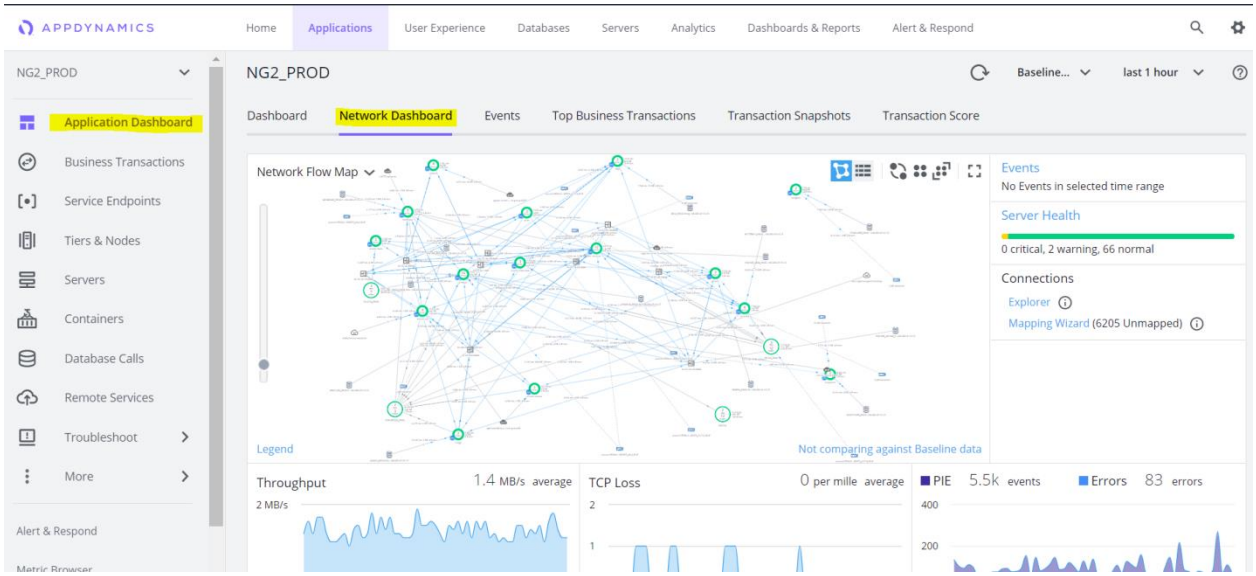
Example:
/usr/bin/ansible-playbook -i ng-crimson-wlp-azure/encoreqa_azure_rm.yml ng-crimson-wlp-azure/appd.yml --limit encoreqa --extra-vars ng_server_env=encoreqa --vault-password-file ~/.vault_pass.txt

4) Once Appdynamics installation done login to appd controller to check the metrics

http://sgdc2apds001t:8090/controller/ **[QA/DEV]**
https://in-apds-cluster1-psg00.asia.essilor.group/controller/ **[PROD]**

5) Go to the application tab and click on the application name
6) On the left side panel you can see Tiers & Nodes click on it
7) If appd installed is done then you can see App Agent Status as well as Machine Agent




APPDYNAMICS
Home
Applications
User Experience
Databases
Servers
Analytics
Dashboards & Reports
Alert & Respond
NG2_PROD
Application Dashboard
Business Transactions
Service Endpoints
Tiers & Nodes
Servers
Containers
Database Calls
Remote Services
Troubleshoot
More
Alert & Respond
Metric Browser
NG2_PROD
Baseline...
last 1 hour
Dashboard
Network Dashboard
Events
Top Business Transactions
Transaction Snapshots
Transaction Score
Network Flow Map
Legend
Not comparing against Baseline data
Events
No Events in selected time range
Server Health
0 critical, 2 warning, 66 normal
Connections
Explorer
Mapping Wizard (6205 Unmapped)
Throughput
1.4 MB/s average
2 MB/s
TCP Loss
0 per mille average
2
1
PIE
5.5k events
Errors
83 errors
400
200

Clients:

1) For clients no need to add app_application_name you can directly installation using playbook command.

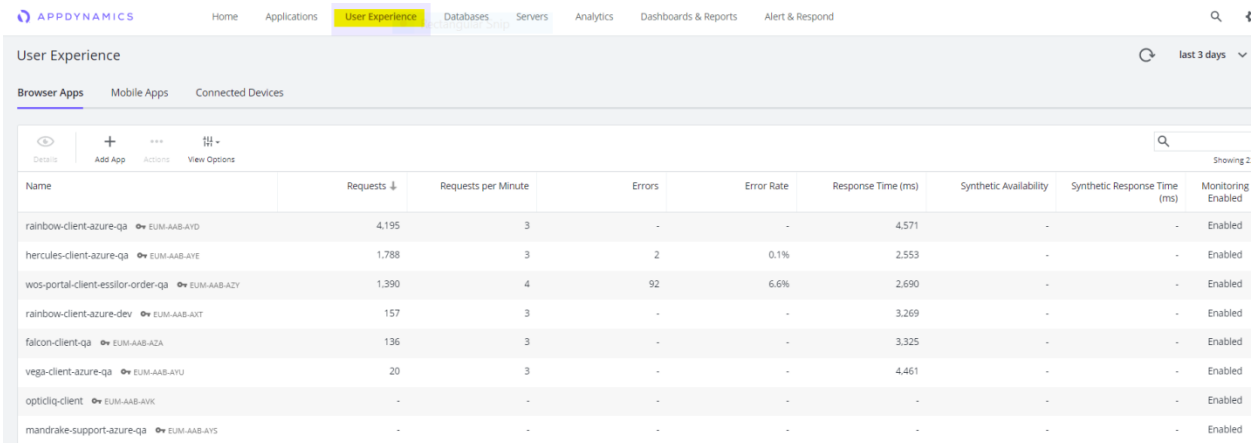
   /usr/bin/ansible-playbook -i <playbook-dir>/<env>_azure_rm.yml <playbook-dir>/appd.yml --limit <env> --extra-vars ng_server_env=<env> --vault-password-file ~/.vault_pass.txt

   Now we need to get app key, adrum.js and code snippet to include in the nginx.conf from controller .

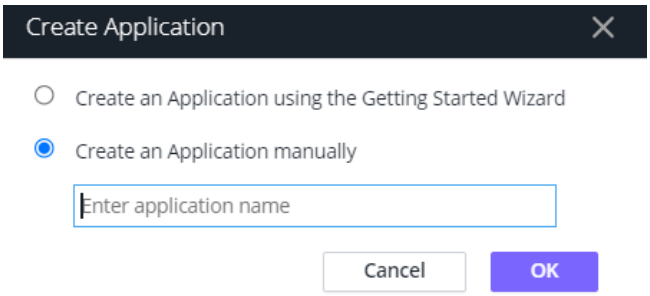
2) Firstly Go to controller →User Experience Tab

   http://sgdc2apds001t:8090/controller/ [QA/DEV]

   https://in-apds-cluster1-psg00.asia.essilor.group/controller/ [PROD]

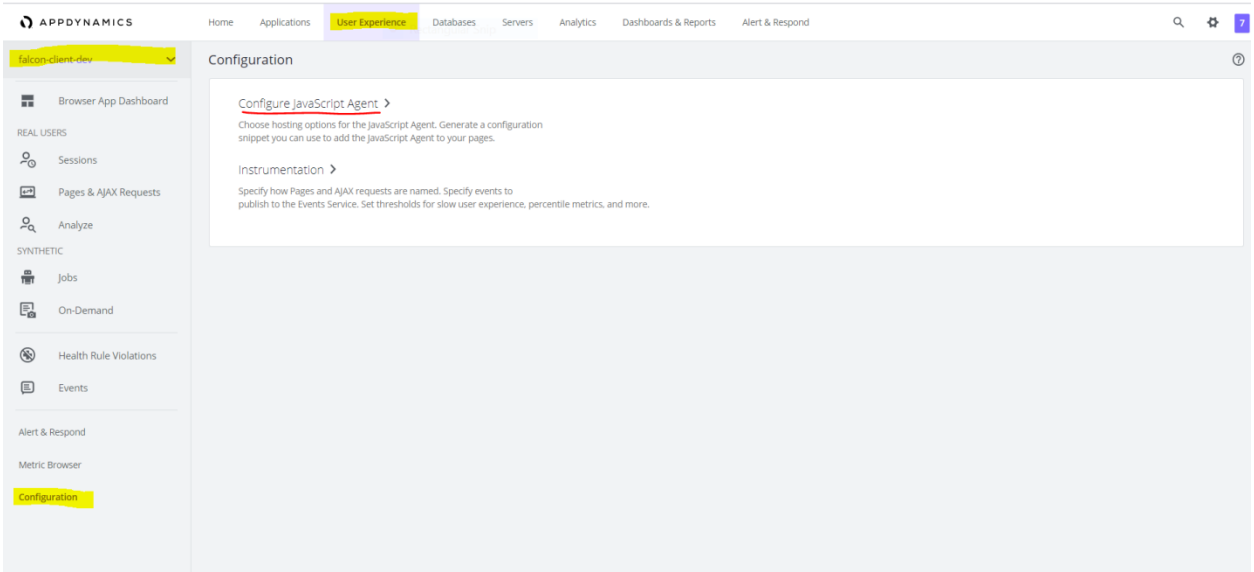


3) Click on Add app + → Diagloue box appears to create app name -- > choose to create app manually and enter app name



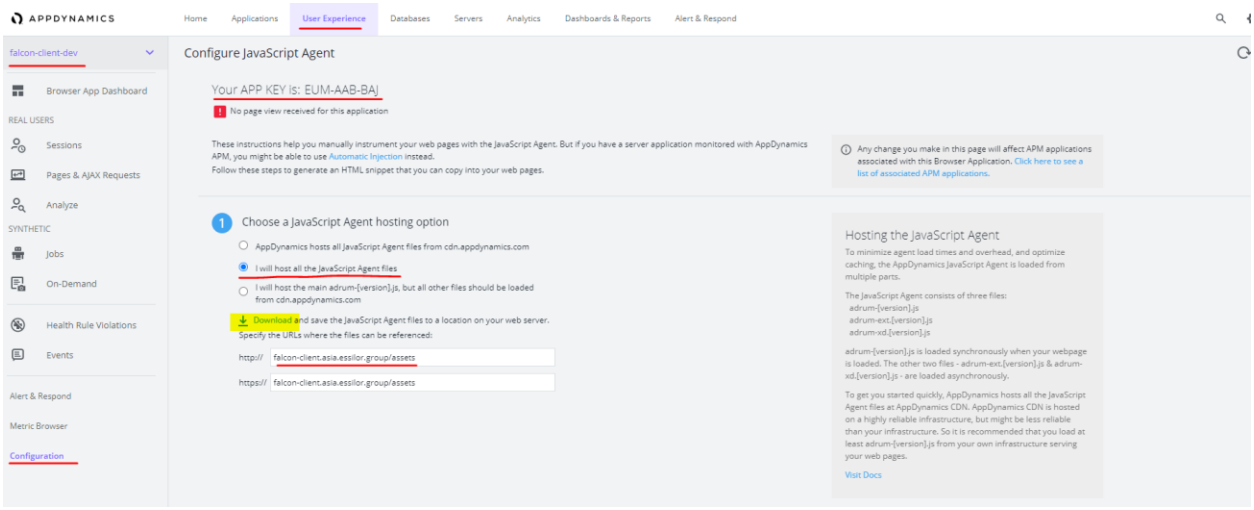
4) After creating application name , on the left side panel click on Configuration -- > Configure JavaScript Agent

Ref: below screen:



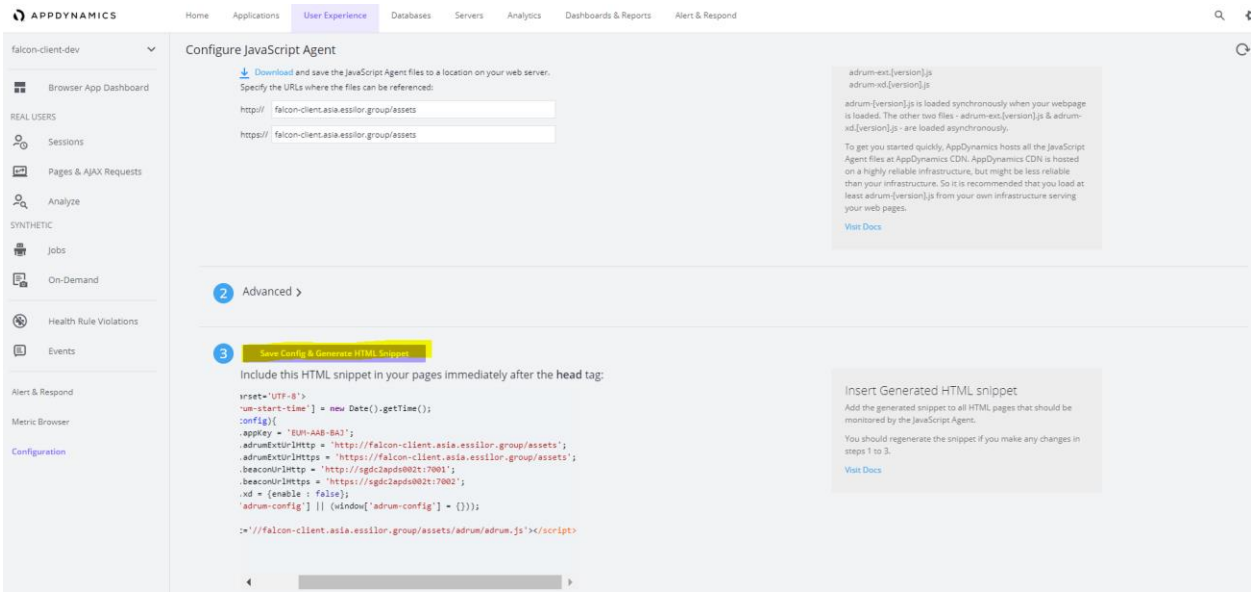
5) Now you can view your app key on next page. As next step choose a JavaScript Agent hosting option -- > I will host all Java Script Agent Files

6) Then add http url of the app of assests location



7) After adding url you can download adrum.js file from the controller in the same page Or else we can take adrum file from any one of the client playbooks.

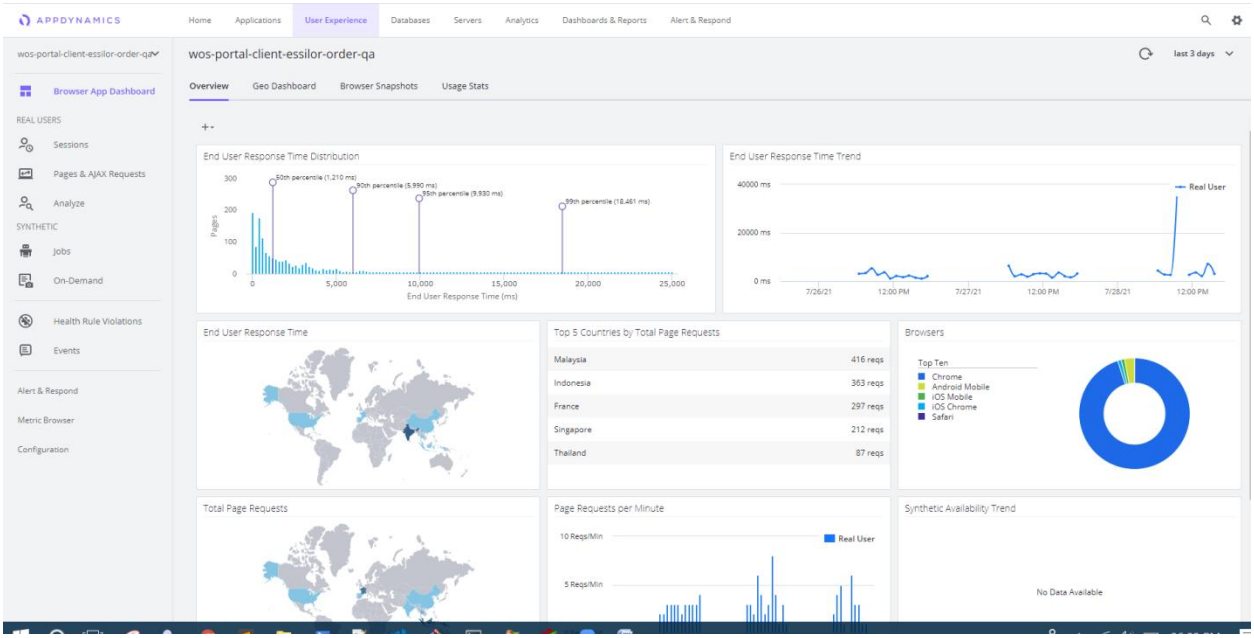
8) Finally Click on Save Config and Generate HTML Snippet. You will see code snippet



9) Finally go to client playbooks and edit nginx conf to add following code snippet(Refer existing client playbooks) Change appkey and http url values for existing code.

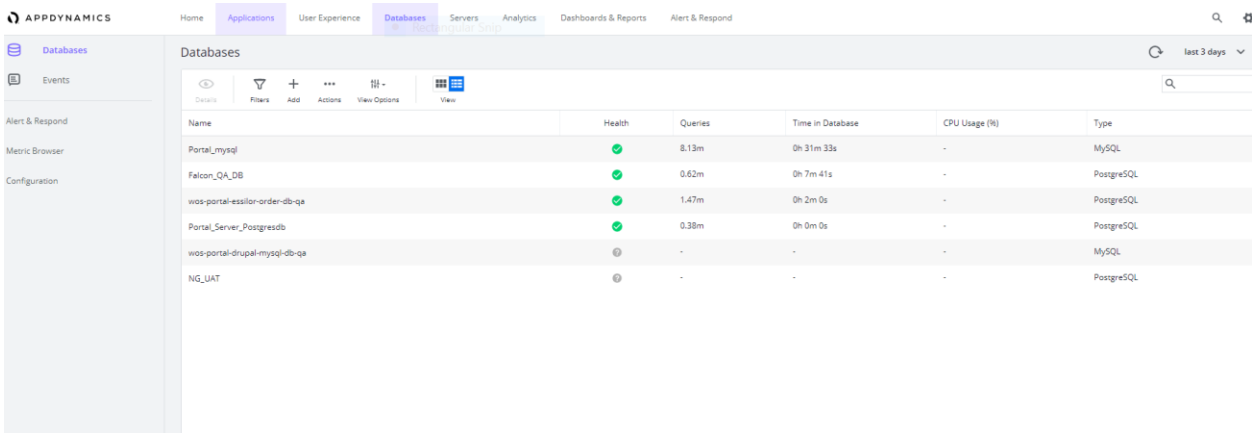
```
<script charset='UTF-8'>
window['adrum-start-time'] = new Date().getTime();
(function(config){
   config.appKey = 'EUM-AAB-BAJ';
   config.adrumExtUrlHttp = 'http://falcon-client.asia.essilor.group/assets';
   config.adrumExtUrlHttps = 'https://falcon-client.asia.essilor.group/assets';
   config.beaconUrlHttp = 'http://sgdc2apds002t:7001';
   config.beaconUrlHttps = 'https://sgdc2apds002t:7002';
   config.xd = {enable : false};
})(window['adrum-config'] || (window['adrum-config'] = {}));
</script>
<script src='//falcon-client.asia.essilor.group/assets/adrum/adrum.js'></script>
```

10) After deploying latest changes go to controller to view end user metrics from client apps.

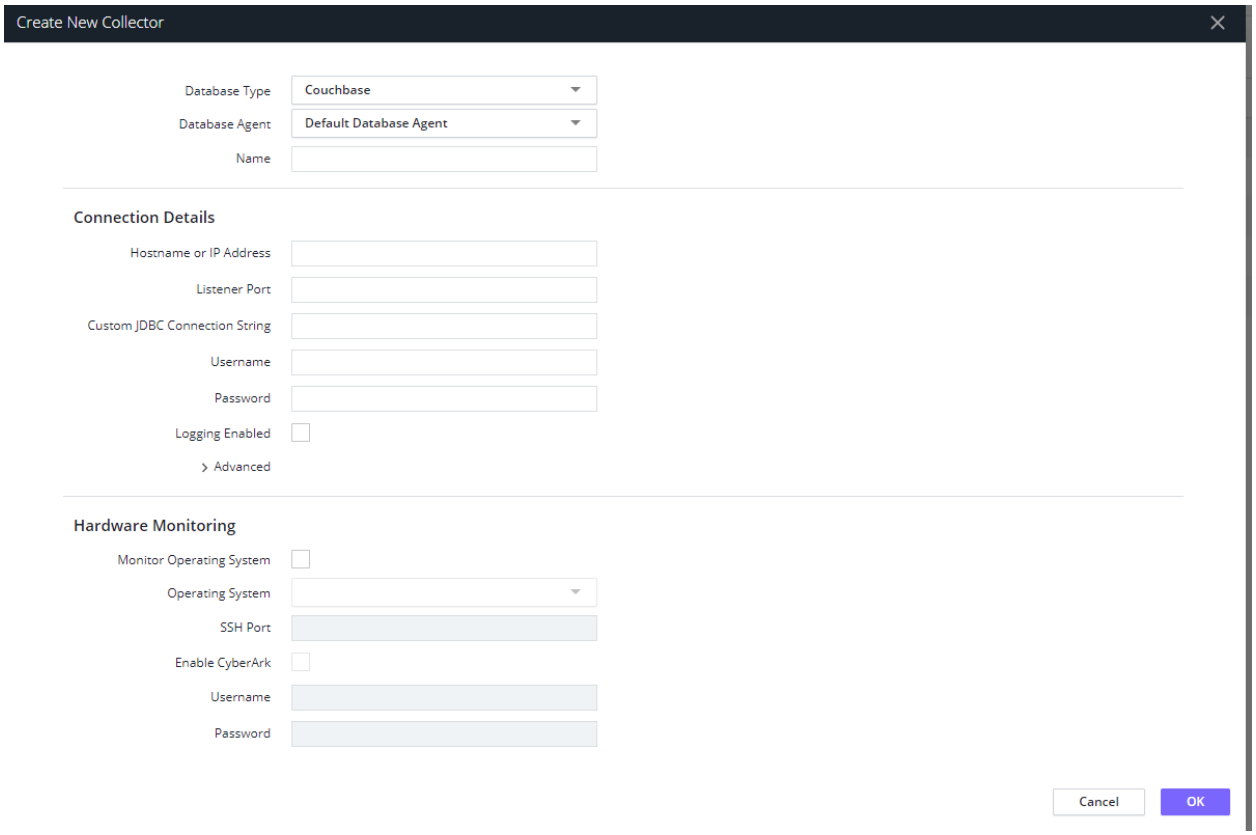


DATABASES:

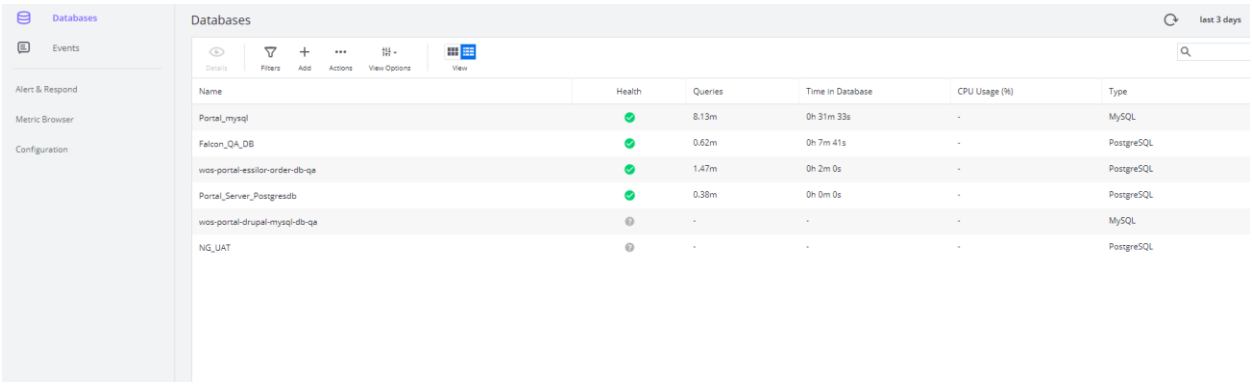
1) To add databases agent -- > Go to Controller -- > Databases Tab -- > click on Add

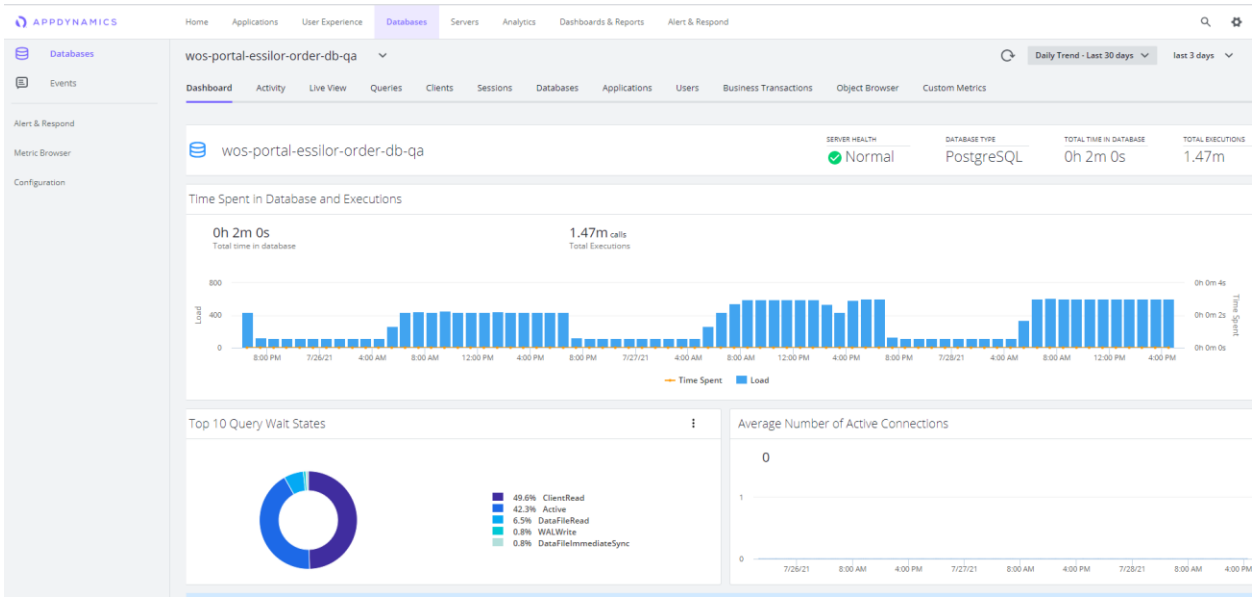


2) Enter Database Type, Name, Hostname, Listener Port, UserName and Password



3) Upon successfull addition you can view metrics by clicking on database name





Analytics Agent :

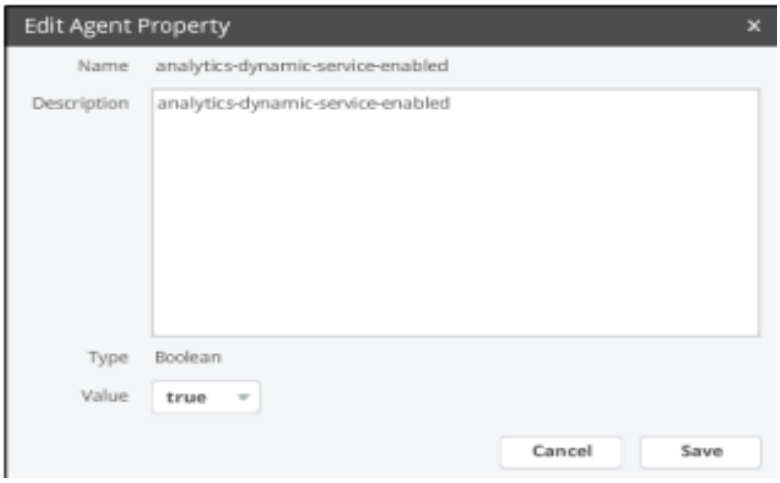
Installation & Configuration:

Download and install either the Standalone Machine Agent or the Analytics Agent (standalone binary) on each machine.

Enable Analytics Agent as Standalone Machine Agent Extension

The Analytics Agent is implemented as an extension to the Standalone Machine Agent (and runs as a machine agent monitor). In environments with the Standalone Machine Agent already running, you enable and run the Analytics Agent as an extension. Use the following steps:

1. Add analytics-dynamic-service-enabled property in app server agent configuration for analytics agent below 4.3 version. Path: Go to **Tiers & Nodes –> Agents -> App Server agents -> Configure** -> + (create property)



2. On the host running the Standalone Machine Agent, use a text editor to open **/monitors/analytics-agent/monitor.xml**

Set the enabled tag to true as follows, saving the file when you are finished:

```
<monitor>
    <name>AppDynamics Analytics Agent</name>
    <type>managed</type>
    <!-- Enabling this requires JRE 7 or higher  -->
    <enabled>true</enabled>
```

3. Configure connectivity from the analytics-agent to the Events Service by editing the analyticsagent.properties file:
   /monitors/analytics-agent/conf/analytics-agent.properties

4. In the analytics-agent.properties file, edit the following fields
   - **http.event.endpoint=(Event Service IP:Port)**
   - **http.event.accountName=(Global Account Name)**
   - **http.event.accessKey = (Controller's Access Key)**

   5. Save and close the file.

   6. If the machine-agent is already running at this point, it needs to be restarted to pick up the changes in the configuration.

To start and stop the Analytics Agent without starting the machine agent (Linux):

/monitors/analytics-agent/bin/analytics-agent.sh

To start and stop the Analytics Agent (Windows):

You can use native windows services menu to start/stop the service or you can do it directly from command line using the following two commands.

To start the agent service from the command line: **bin\analytics-agent.exe service-start**

To stop the agent service for the command line: **bin\analytics-agent.exe service-stop**

To uninstall the Windows service:

Run the .exe file with the uninstall command as follows:

bin\analytics-agent.exe service-uninstall.