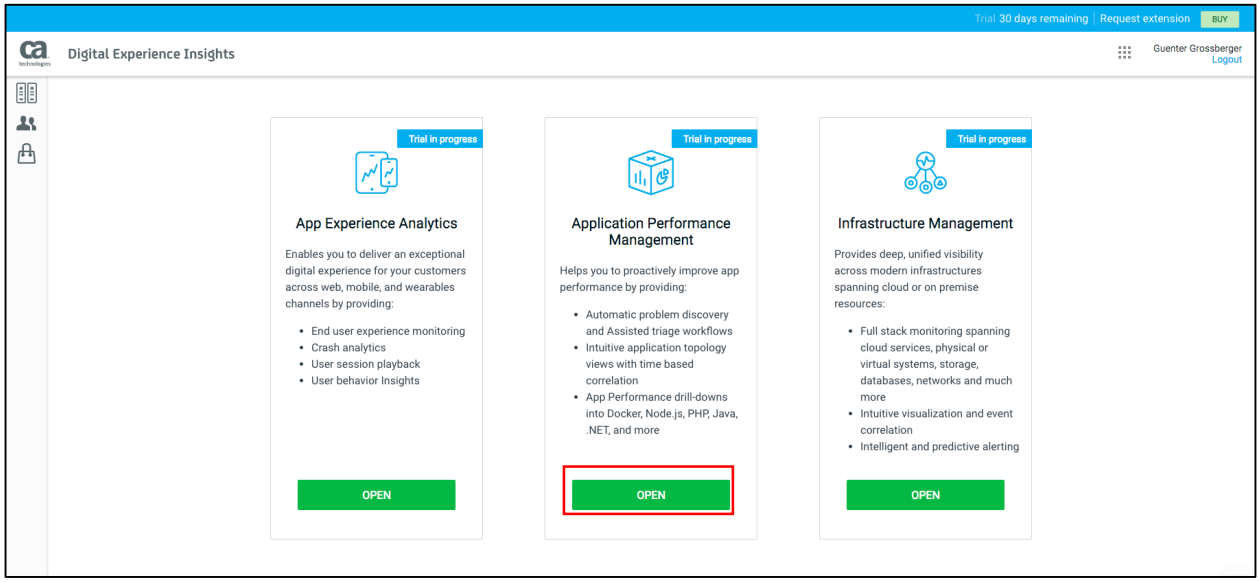
Monitor Modern Applications in the Cloud

Lab Instructions

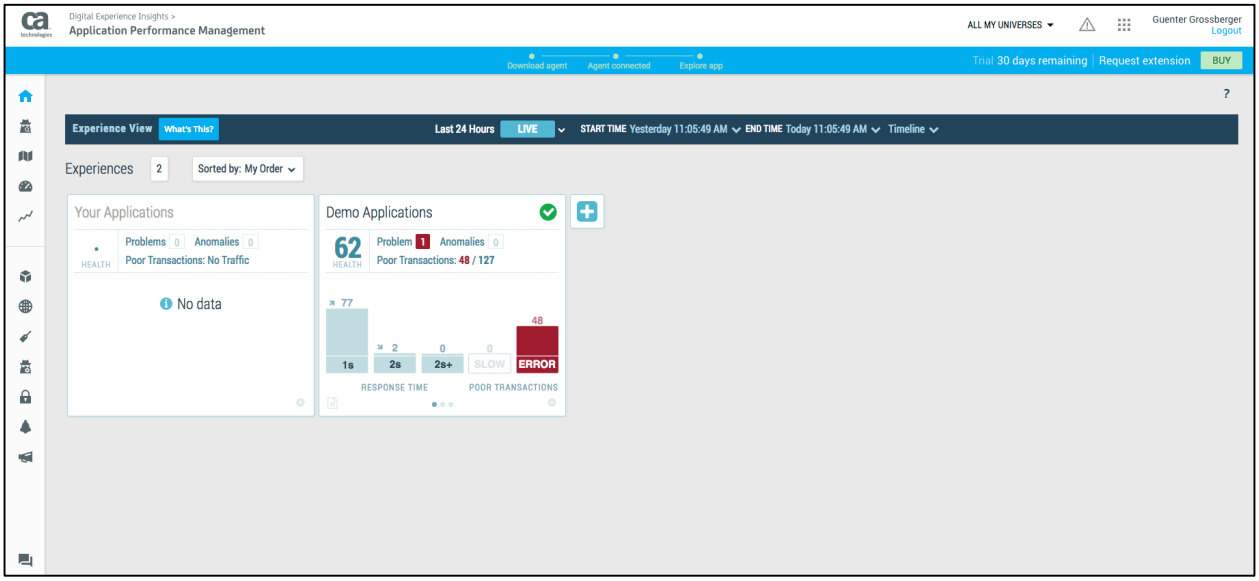
1. Setup CA Digital Experience Insghts (DXI) trial account

If you already have a CA Digital Experience Insghts (DXI) trial account log in with your existing credentials (user your email address as tenant ID if asked) and skip this step.

1. Open https://cloud.ca.com in your browser
2. Click “Click here to register”
3. Click “1. Sign up”
4. Fill out the signup form and click “Get started”. Use your company email address. Public email providers like gmail, yahoo, … are not supported.
5. On the DXI start page click the “Open” button in the Application Performance Management box (middle):

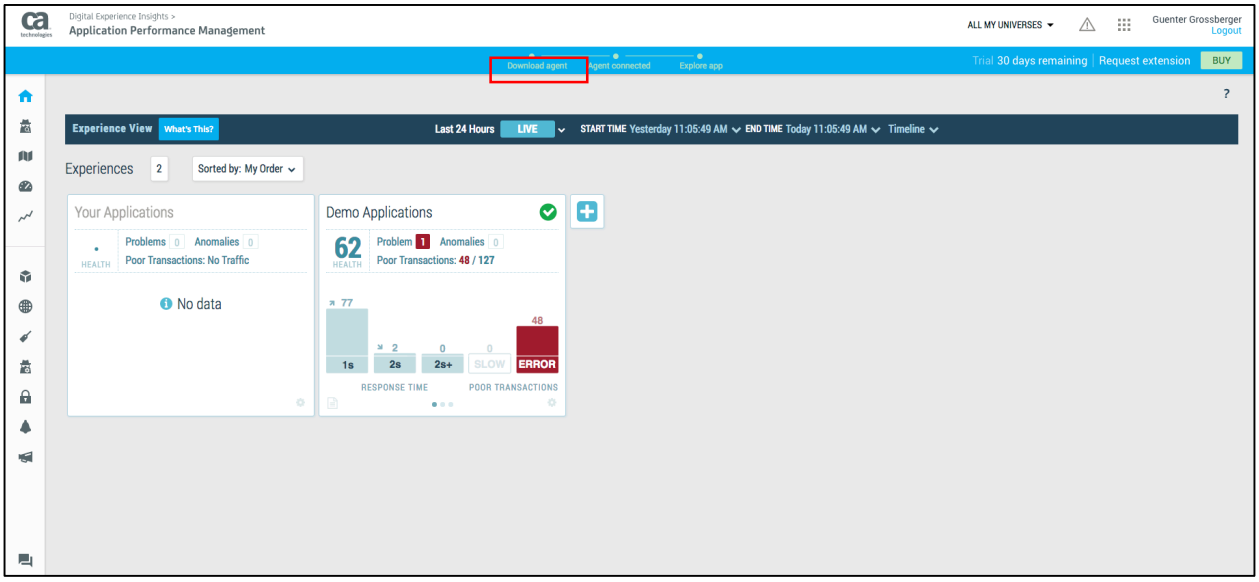


6. You see the CA APM home page (Experience View):

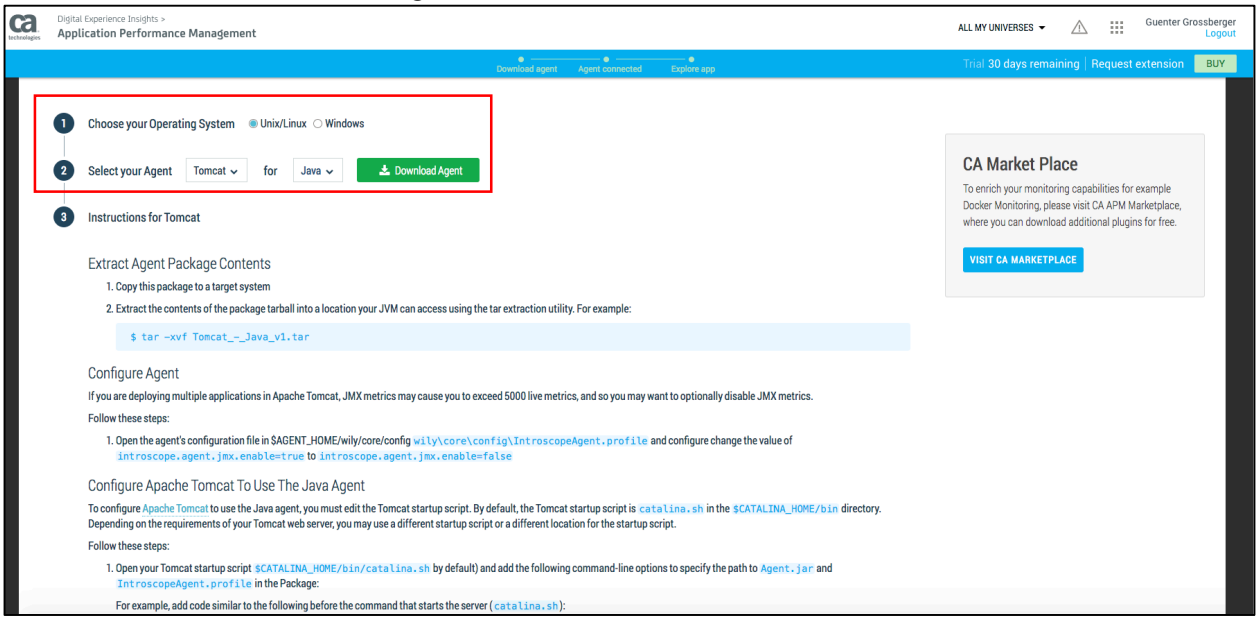


2. Download and install application agents

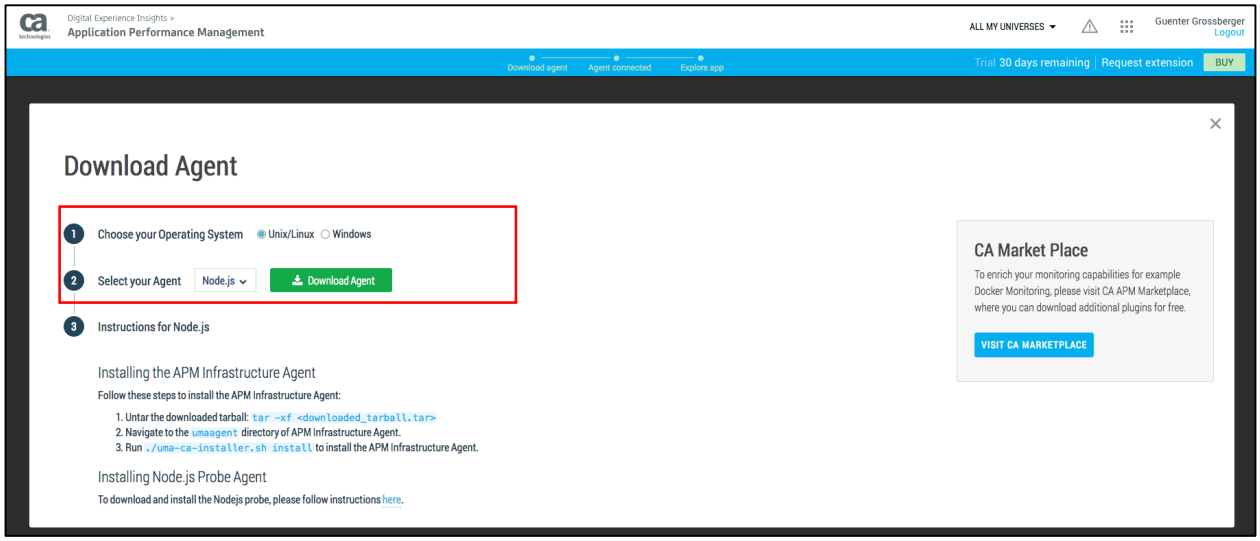
1. Click on “Download agent” in the top blue navigation bar:



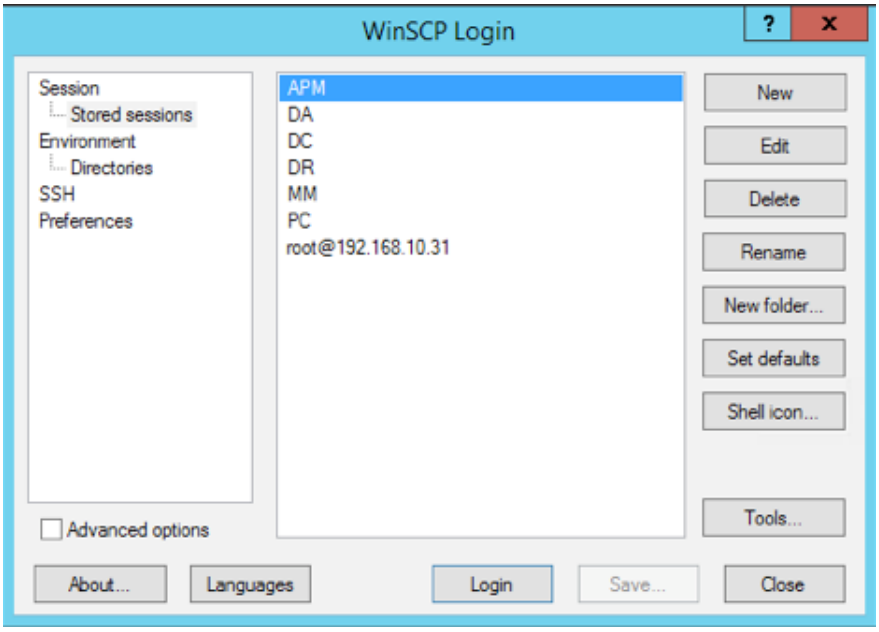
2. Download Java agent: choose operating system “Unix/Linux” and “Tomcat” for “Java”. Click on “Download agent” and save to local disk.



3. Download Node.js agent: choose operating system “Unix/Linux” and “Node.js”. Click on “Download agent” and save to local disk.

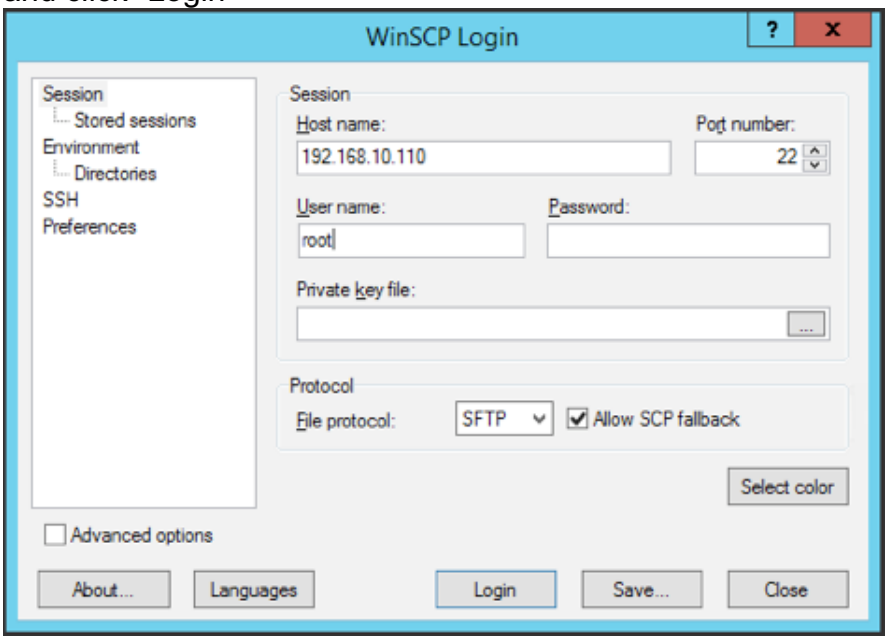


4. Copy agents to Linux VM: open the application “WinSCP” from the task bar and click “New”

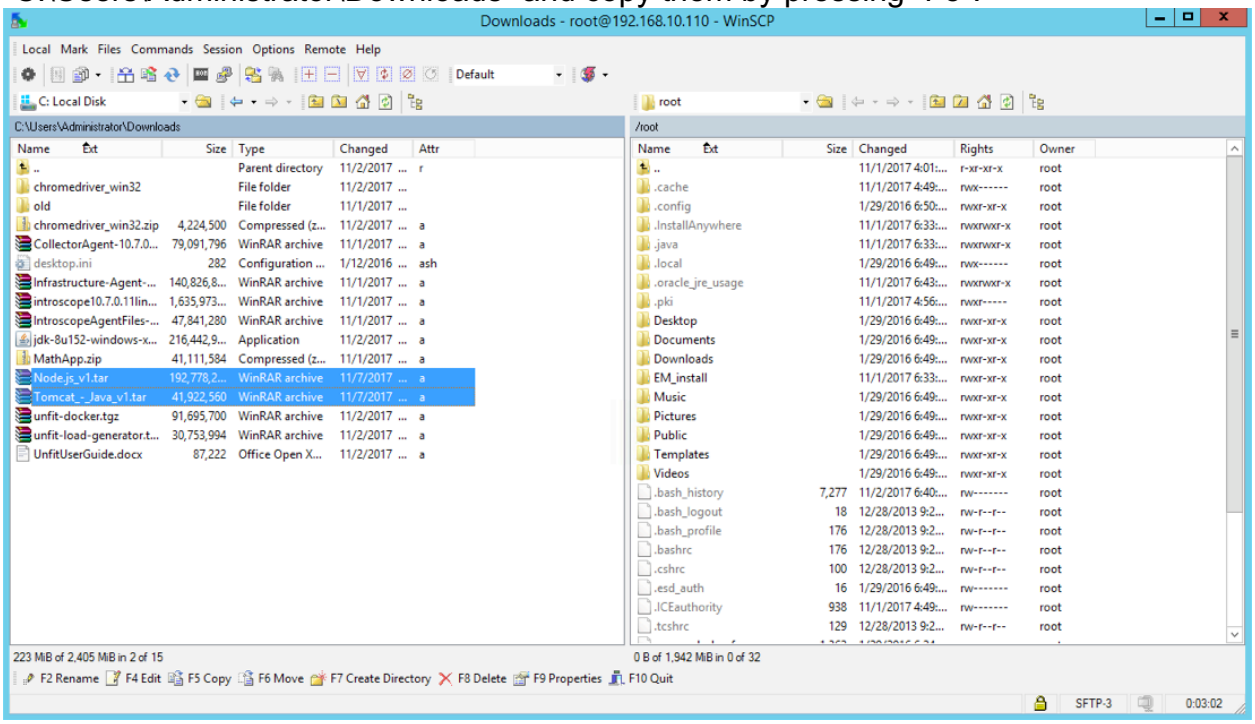


5. Enter:
   - host name “192.168.10.110”
   - user name “root”
   - password “CAdemo123”

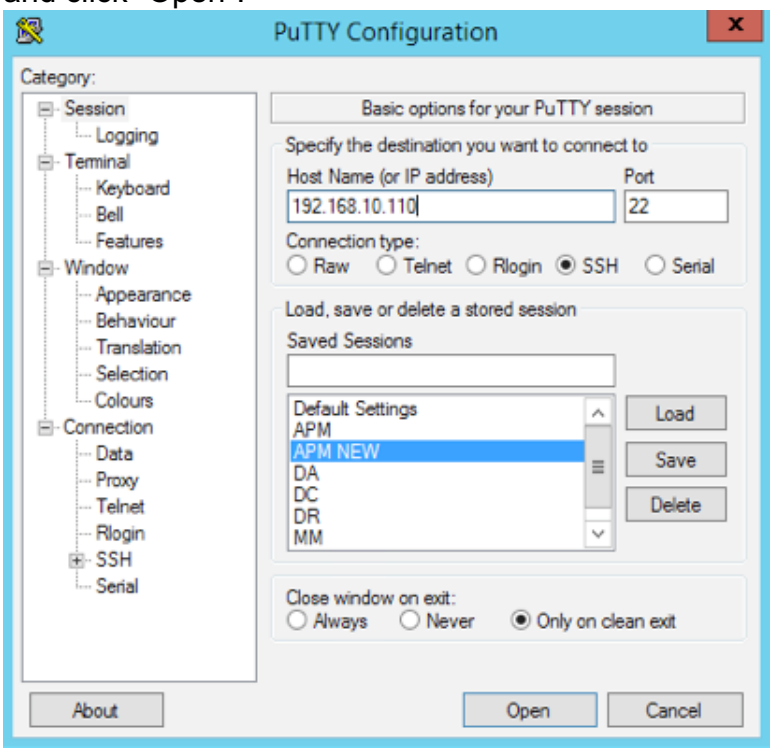
   and click “Login”



6. Select the files “Node.js_v1.tar” and “Tomcat_-_Java_v1.tar” from the diretory “C:\Users\Administrator\Downloads” and copy them by pressing “F5”.



7. Open the application “Putty” from the task bar, enter host name “192.168.10.110” and click “Open”:



8. Login as user “root” with password “CAdemo123”

9. Type “cd /opt/lab/unfit”

10. Type “cp ~/Tomcat_-_Java_v1.tar agents

11. Type “cp ~/Node.js_v1.tar agents

12. Open the file “.env” in text editor: “vi .env”

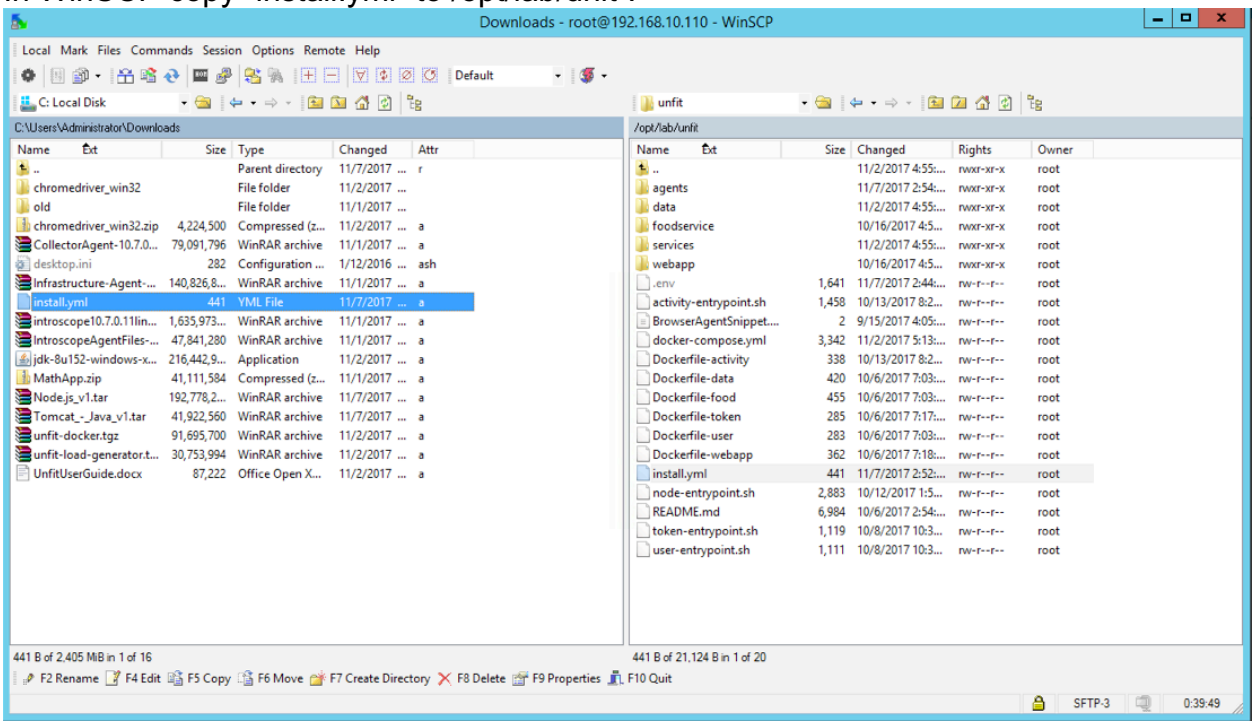
13. Change the following agent properties and comment out the two “EM” properties:
    - JAVA_AGENT=Tomcat_-_Java_v1.tar
    - NODE_AGENT=Node.js_v1.tar
    - #EM_HOST=192.168.10.110
    - #EM_PORT=5001

14. Make sure docker is running by typing “systemctl start docker”

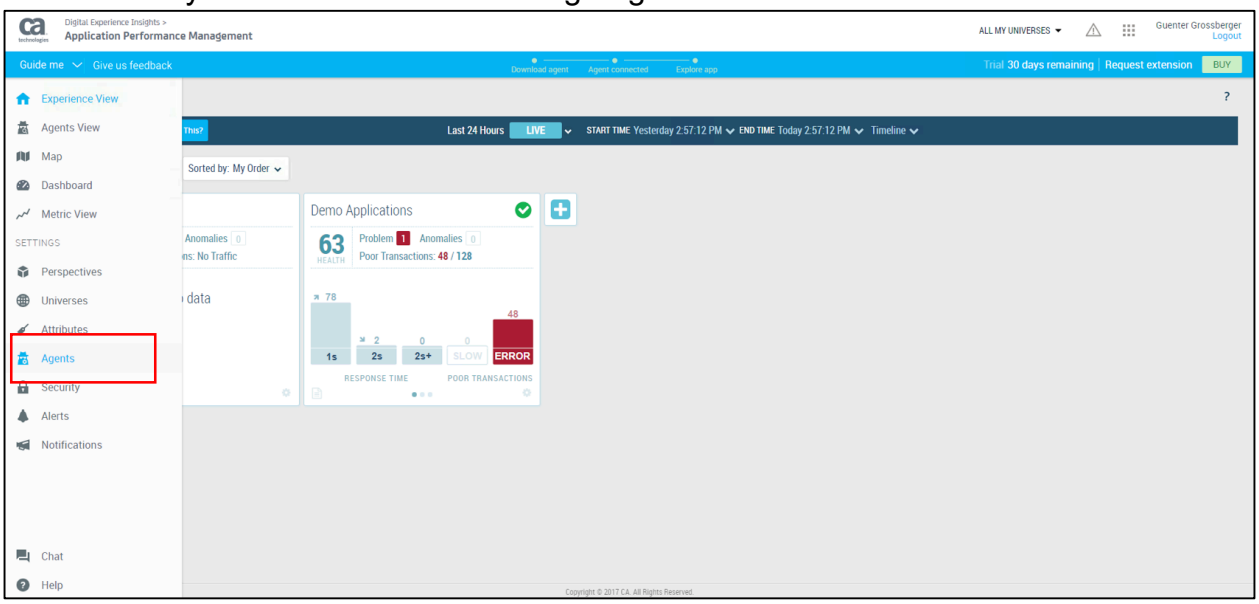
15. Type “docker-compose build” to build the unfit application with the downloaded agents

3. Download and install Docker agent

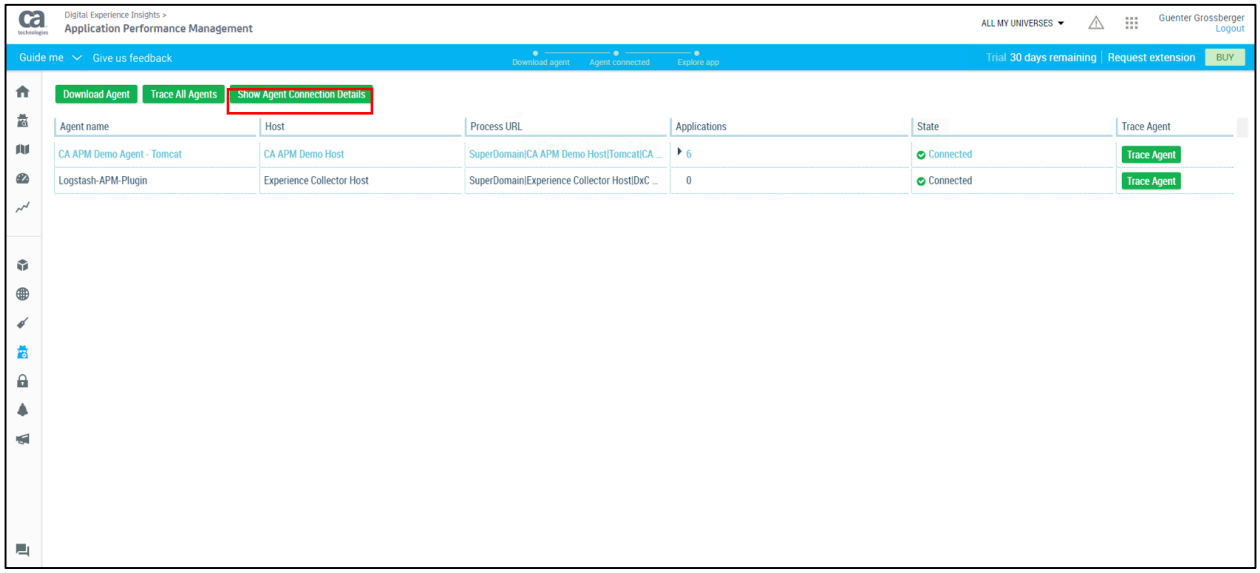
1. In WinSCP copy “install.yml” to /opt/lab/unit”:



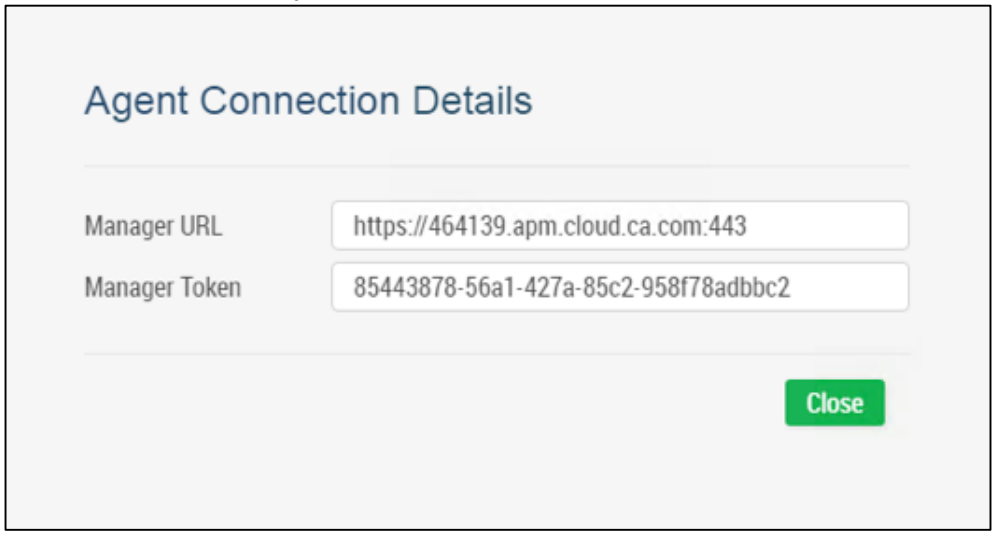
2. In CA DXI in your browser select “Settings/Agents”:



3. Click on “Show Agent Connection Details”:



Leave that window open:



4. Insert **your** values into install.yml with “vi install.yml”:

```
version: "3"
services:
  dockermonitor:
    image : bhaab01/dockermonitor:latest
    environment:
      - agentManager_url_1=https://464139.apm.cloud.ca.com:443
      - agentManager_credential=85443878-56a1-427a-85c2-958f78adbbc2
      - containerflow=enabled
    […]
```

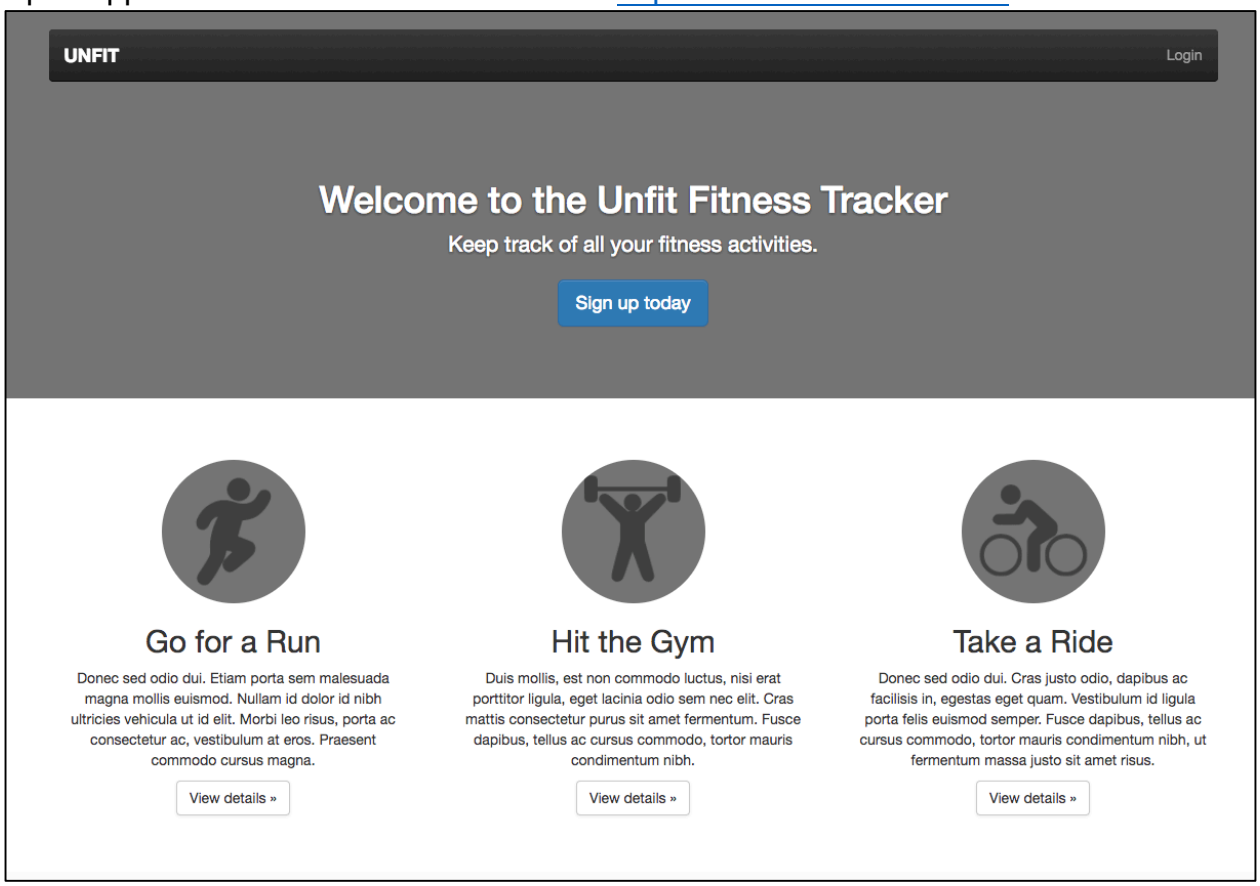
5. Start Docker Monitor with “`docker-compose -f install.yml up –d`

6. Verify that Docker Monitor is running by typing “docker ps”. You should see output like the following with **two containers running** from the image “dockermonitor”, the Docker Monitor and the ContainerFlow agent:

```
docker ps
CONTAINER ID IMAGE                          COMMAND      CREATED       STATUS       PORTS  NAMES
af5bb549ee0e bhaab01/dockermonitor:latest   "bash run.sh"  2 minutes ago Up 2 minutes        dockermonitor
ea9375cbd3a0 bhaab01/dockermonitor:latest   "sh install.sh"  2 minutes ago Up 2 minutes        unfit_dockermonitor_1
```

4. Start unfit application

1. Start unfit application with “docker-compose up –d
2. Open application in browser via shortcut or http://192.168.10.110:8000/



3. Open a windows command prompt from the taskbar
4. Type “cd \apps\unfit-load-generator”
5. Start load generation: type “baseLoad.bat”. Two Chrome windows will open and run load against the unfit application

Section 2:Use CA APM to quickly find the root cause of a performance issue. The Assisted Triage engine looks at application issues, the transactions they affect, and the underlying infrastructure. Utilize CA Docker and infrastructure monitoring for additional insights into the underlying architecture.

Problem Description

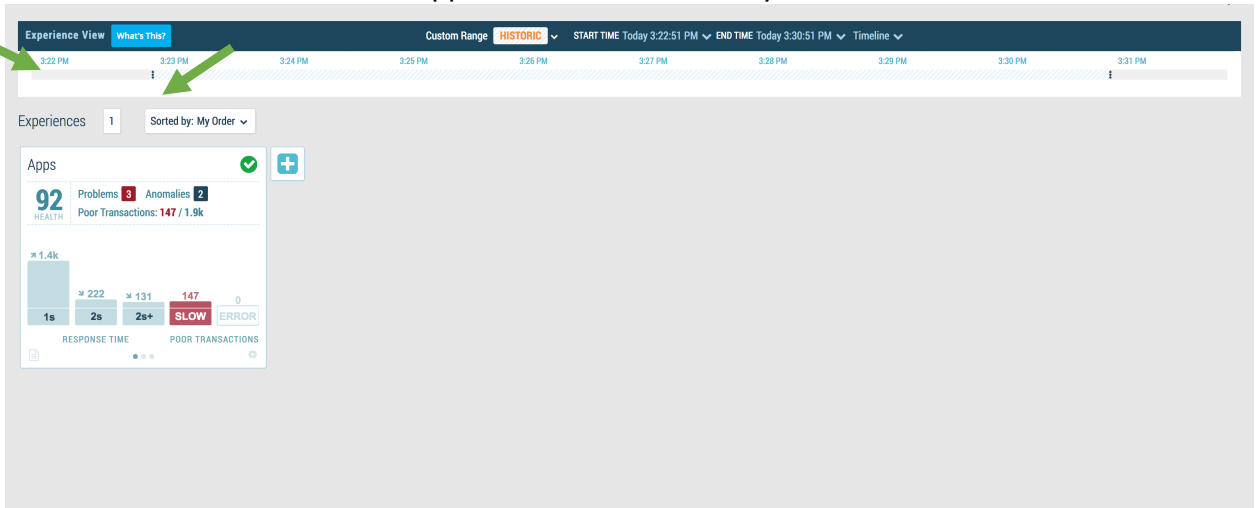
(Example Issue) There has been a steady ramp in traffic since 8 am. At the peak of the traffic, roughly 100,000 concurrent users, the NOC reported that there was a sudden increase in the response time of key transactions, and a decrease in the amount of place order transactions being completed.

1. Open another windows command prompt from the taskbar
2. Type “cd \apps\unfit-load-generator”
3. Start load generation: type “dbtcLoad.bat”. Two Chrome windows will open and run load against the unfit application. If there is no dbtcLoad.bat save the baseLoad.bat as dbtcLoad.bat and replace “baseLoad.xml” with “dbtcLoad.xml” and save again.

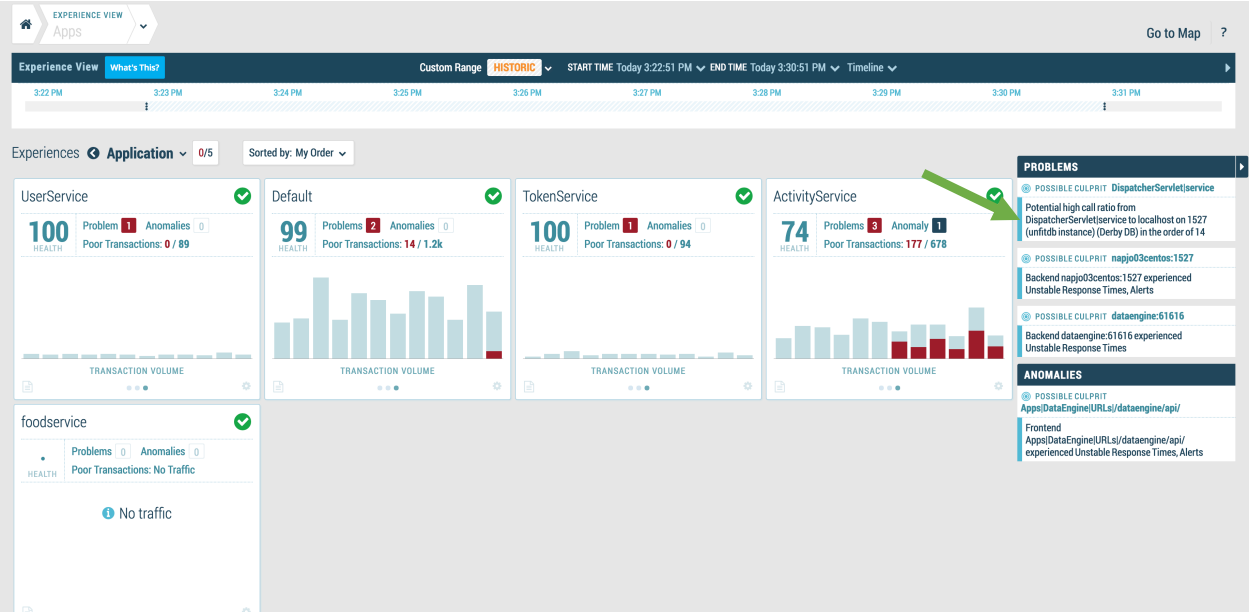
Wait a few minutes for the problem to develop.

Open Chrome browser and goto the APM Homepage. Click on Experience View. Note that your view may be different than the screenshot below

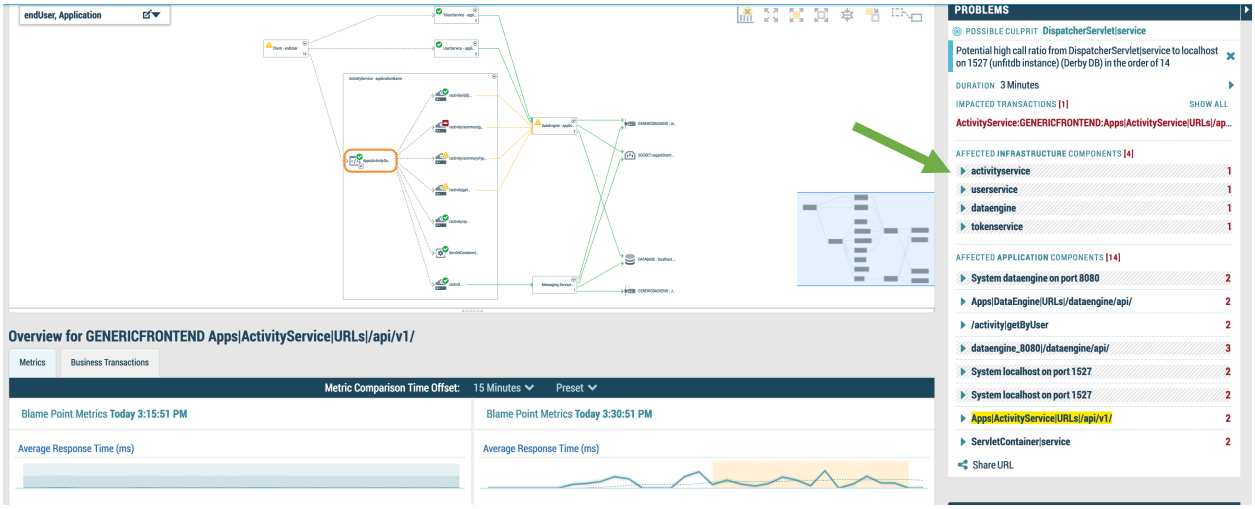
Here we see that there have been 147 slow transactions, creating 3 separate problems for the end users. Drill in to see which applications are affected by this issue.



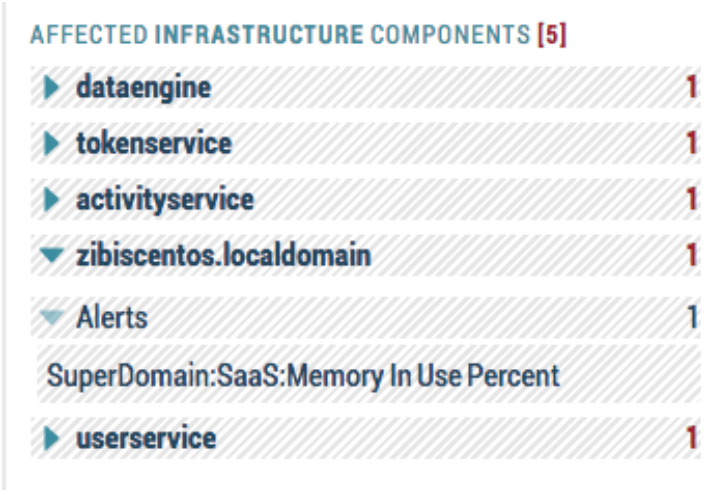
Let's check out the high call ratio issue listed on the right hand 'Problems' panel. Click on the 'Potential high call ratio' issue on the right.



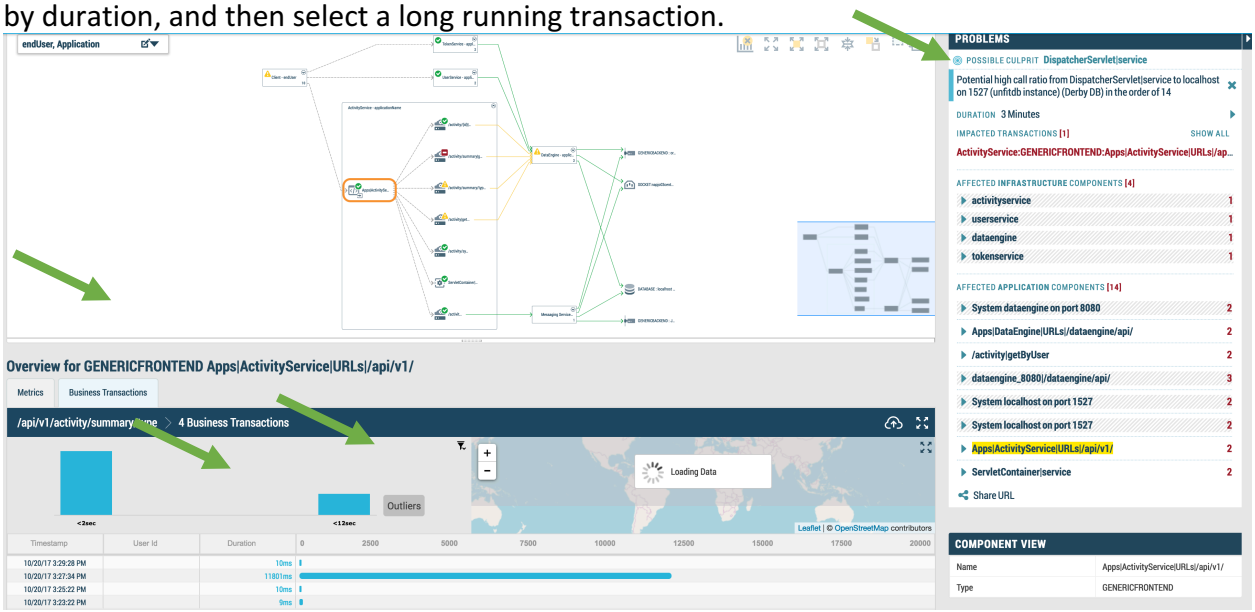
Here, we see all the components involved in this problem. Notice that the affected end User Transaction is highlighted in red. Also note that *infrastructure components are included in the analysis of this issue.* Click on one of the infrastructure components under "Affected Infrastructure Components"



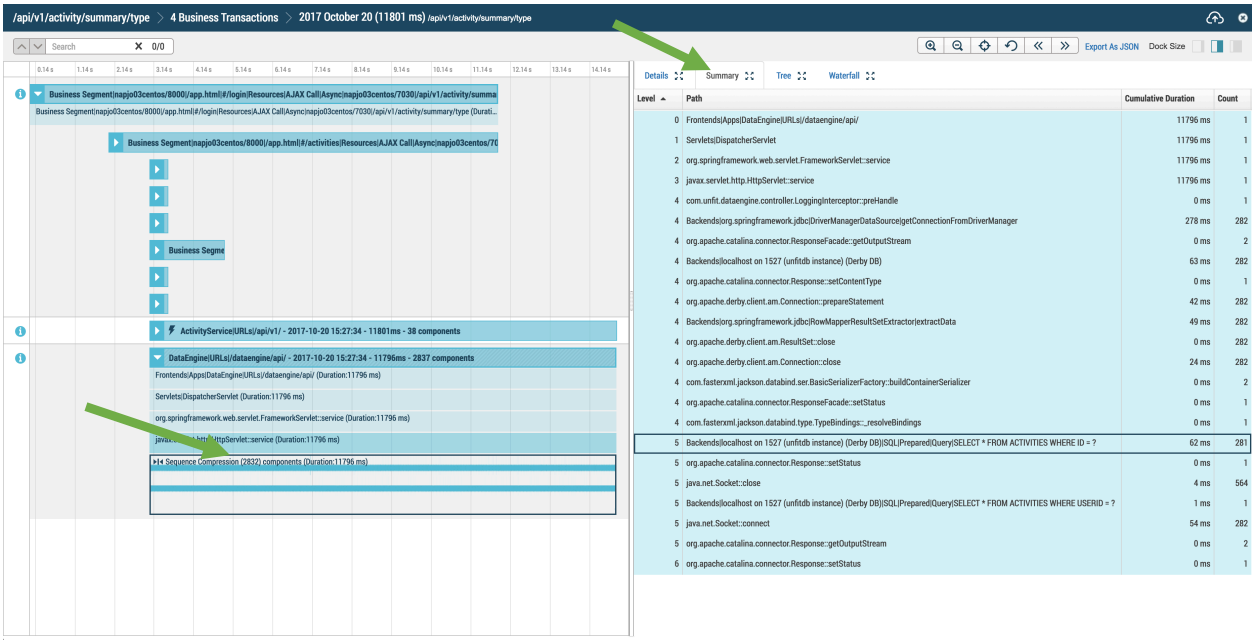
Note that we are monitoring host level metrics including memory, cpu, network and storage. The UIM integration provides even more infrastructure visibility. In this case, this issue has caused a spike in the available memory used.



Now, let's click on the "Impacted Transaction" to see what the impacted transactions look like. Once you have clicked the "Impacted transactions, click on the Business Transaction Tab, sort by duration, and then select a long running transaction.

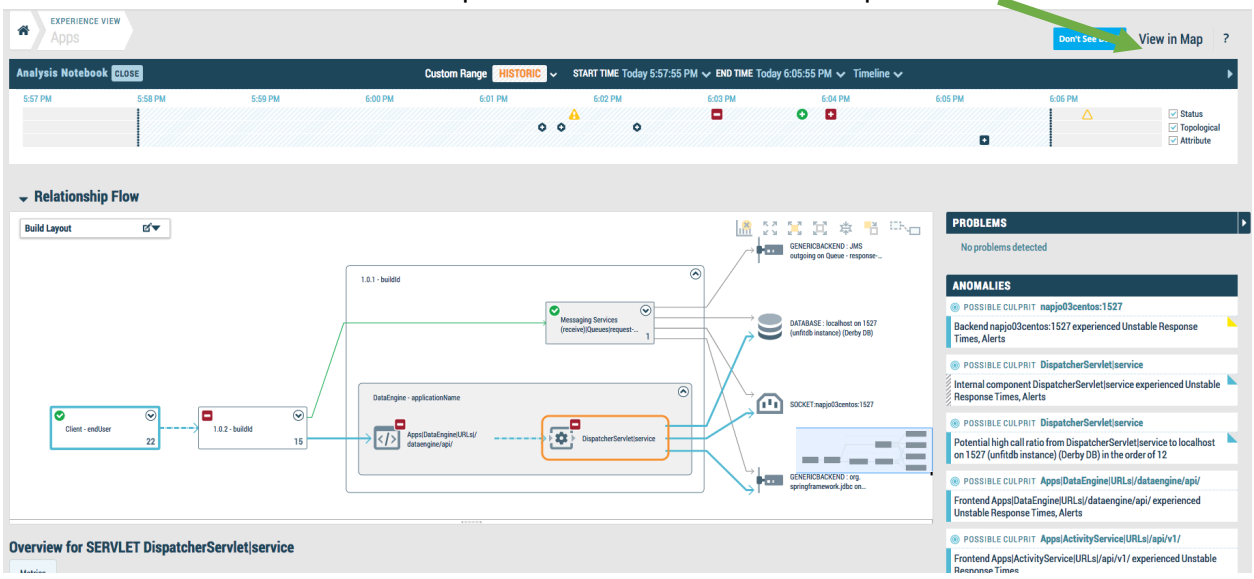


Here, we see an individual transaction that had a high number of calls (2832) to a backend database. This churn is what has caused the issue in the dataengine component. Click on the Summary and tree views for different perspectives of call data.



Docker Container Data

Now that we have quickly discovered the root cause of the issue, let's take a look at some of the other data collected about this problem. Click on 'View in Map' to see the full visualization.



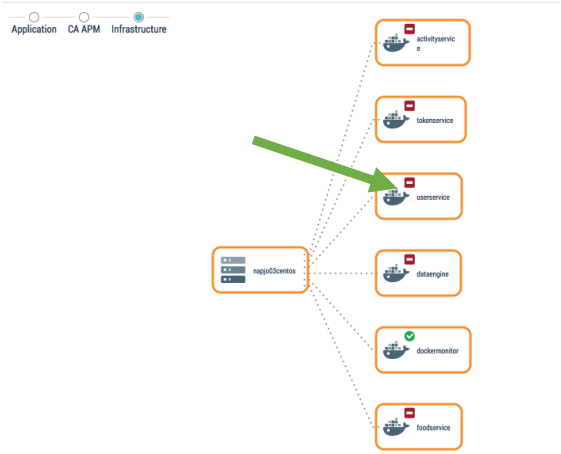
Note that there are three levels in this map:

Application: Application dependencies and relationships

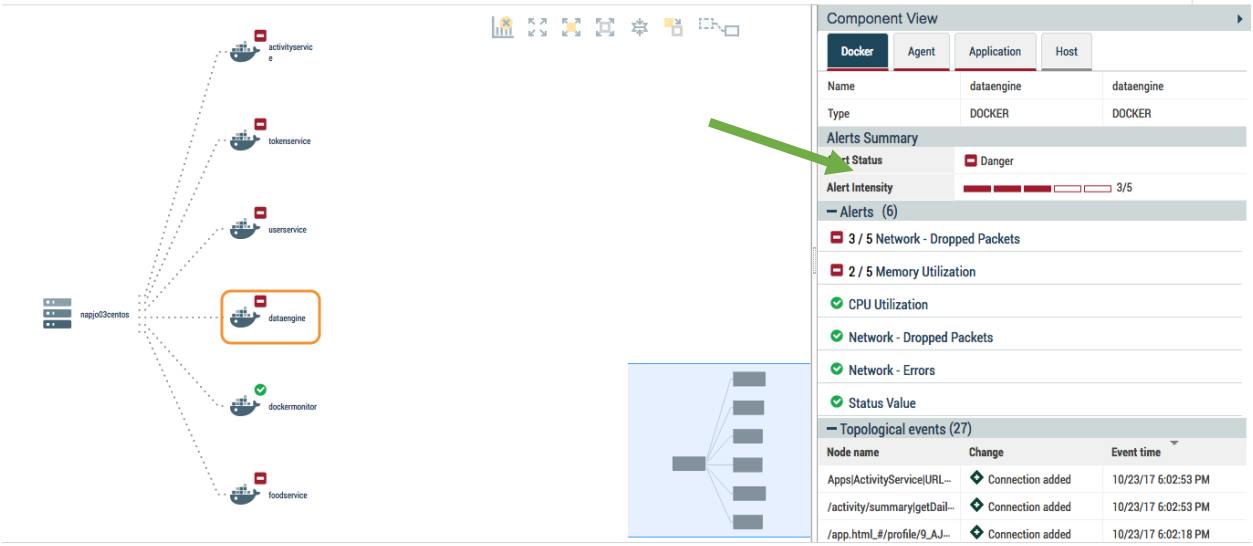
CA APM: Visualization of the environment defined by the CA EM and Agents

Infrastructure: Network infrastructure of your environment

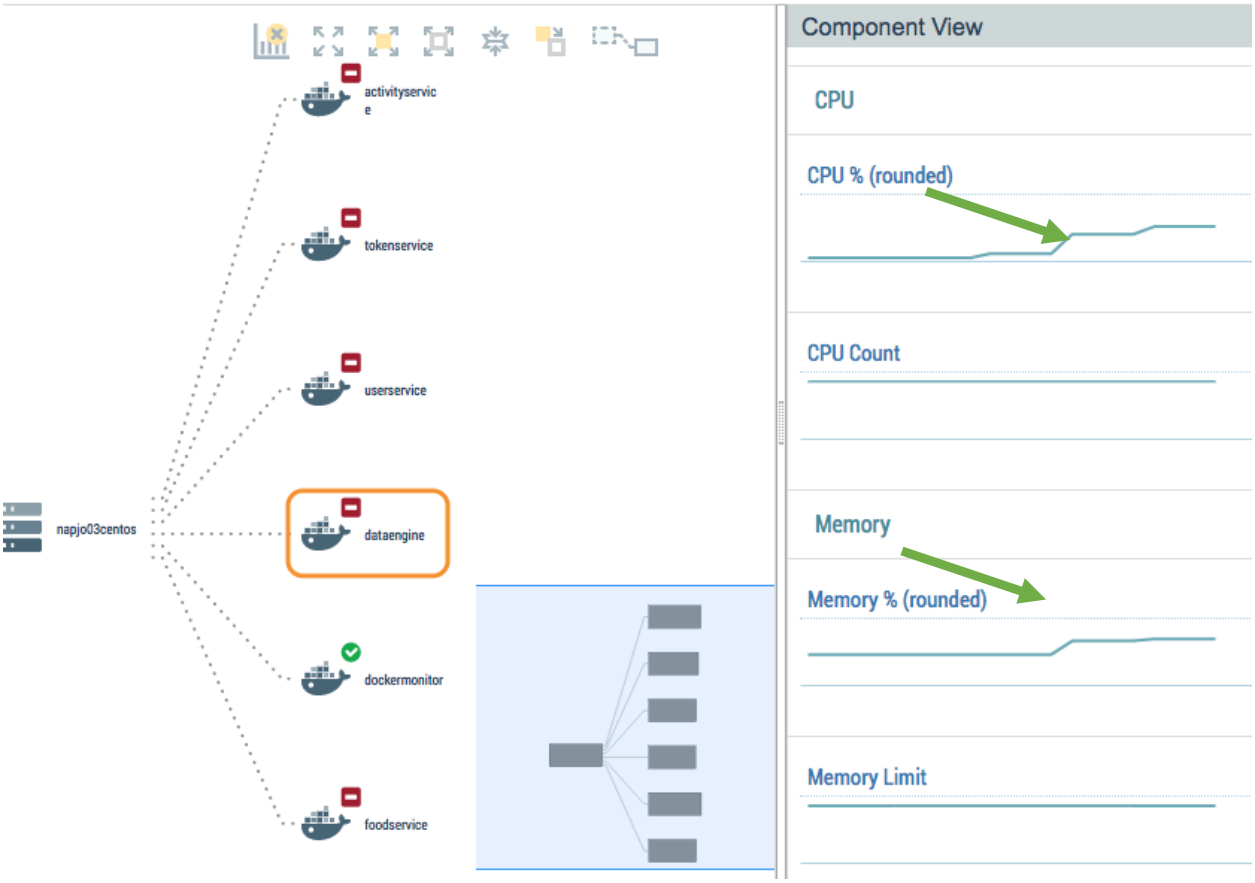
We see in this view that the problem has propagated to multiple containers, but since we are interested in the dataengine, let's start there. Click on the dataengine.



We notice that there are 4 tabs, each relating to a different level of monitoring within the application: Docker, Agent, Application and Host. Clicking on the Docker monitor, we see that two alerts – CPU and Memory have been triggered.



Scrolling down, we can see the metrics that triggered those alerts, and the attributes automatically collected by the Docker monitor. This information, along with the host level metrics are included in the analysis conducted by Assisted Triage.



Note that in addition to individual container health metrics, we also collect host level metrics such as CPU, Memory, Storage, and Network Metrics: