Tutorial: How to Deploy and On-Board an OpenShift Application in DX Application Performance Management

You have decided to containerize applications and use Red Hat OpenShift as the target platform. Now, you'll want to provide the best experience you can for your development and operational teams that are going to be going to production using this tool set. OpenShift provides a proven enterprise-ready and certified Kubernetes distribution that will meet your basic needs out-of-the-box. Red Hat relies on its partner network to provide the best point solutions to add value where customers need it most.

Observability is one of those areas. Broadcom is a Red Hat partner and has gone through the certification process. This ensures its monitoring suite can seamlessly be deployed into an OpenShift cluster and will not introduce any instability to the platform.

Deploying DX Application Performance Management in OpenShift

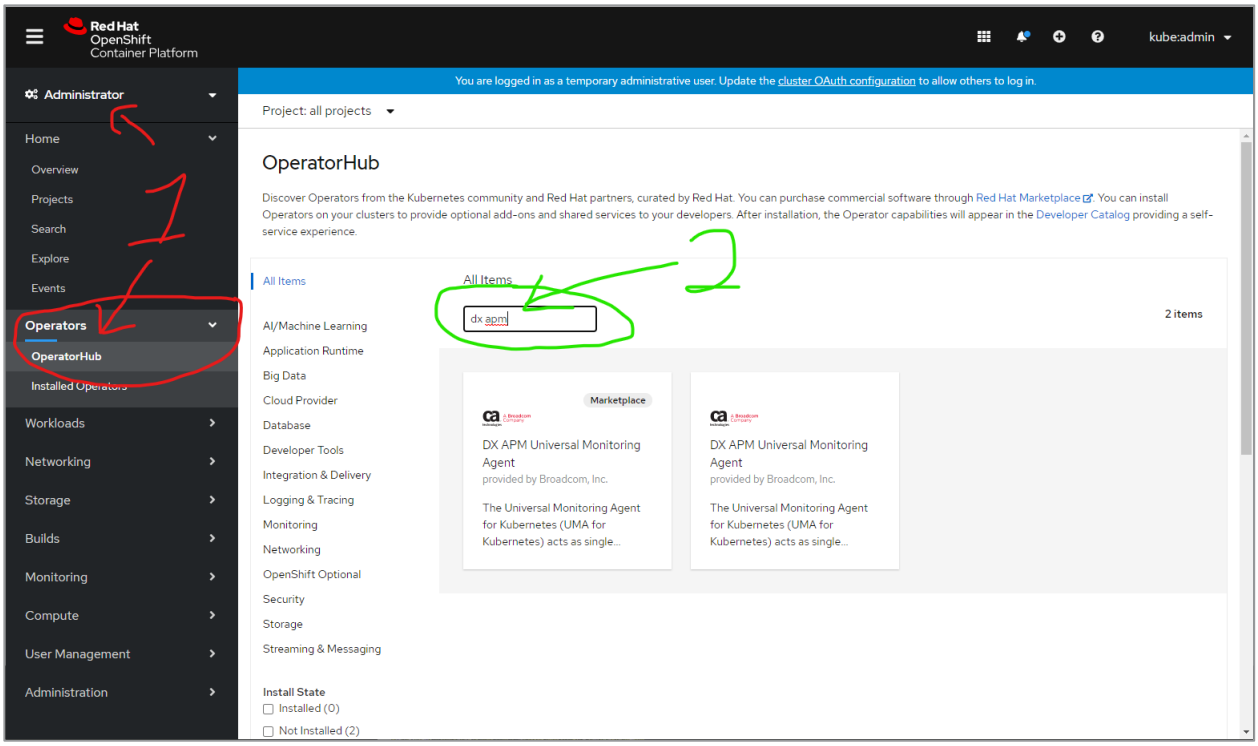
There are two steps to enabling DX monitoring into any environment. The first is to have access to the running DX platform with an account that has permission to add new agents. The second is to deploy those agents on the target systems that will have applications to monitor.

For step one, the easiest option to get started with DX is the starter edition, which is part of the DX APM SaaS offering. This can be with the DX APM Starter Edition, or as part of a larger initiative using starter editions of Broadcom's Enterprise Software products, which involves continuous delivery and both automated functional and performance testing.

The second step is enabling the agents. This can be done in two ways when it comes to OpenShift. The first is to deploy the agent from Operator Hub within the OpenShift administration console. It includes operators certified by Red Hat. Or, enabling the agents can be done by following the instructions found within the DX APM solution. Some companies prefer this method as it is more portable across Kubernetes vendors environments.

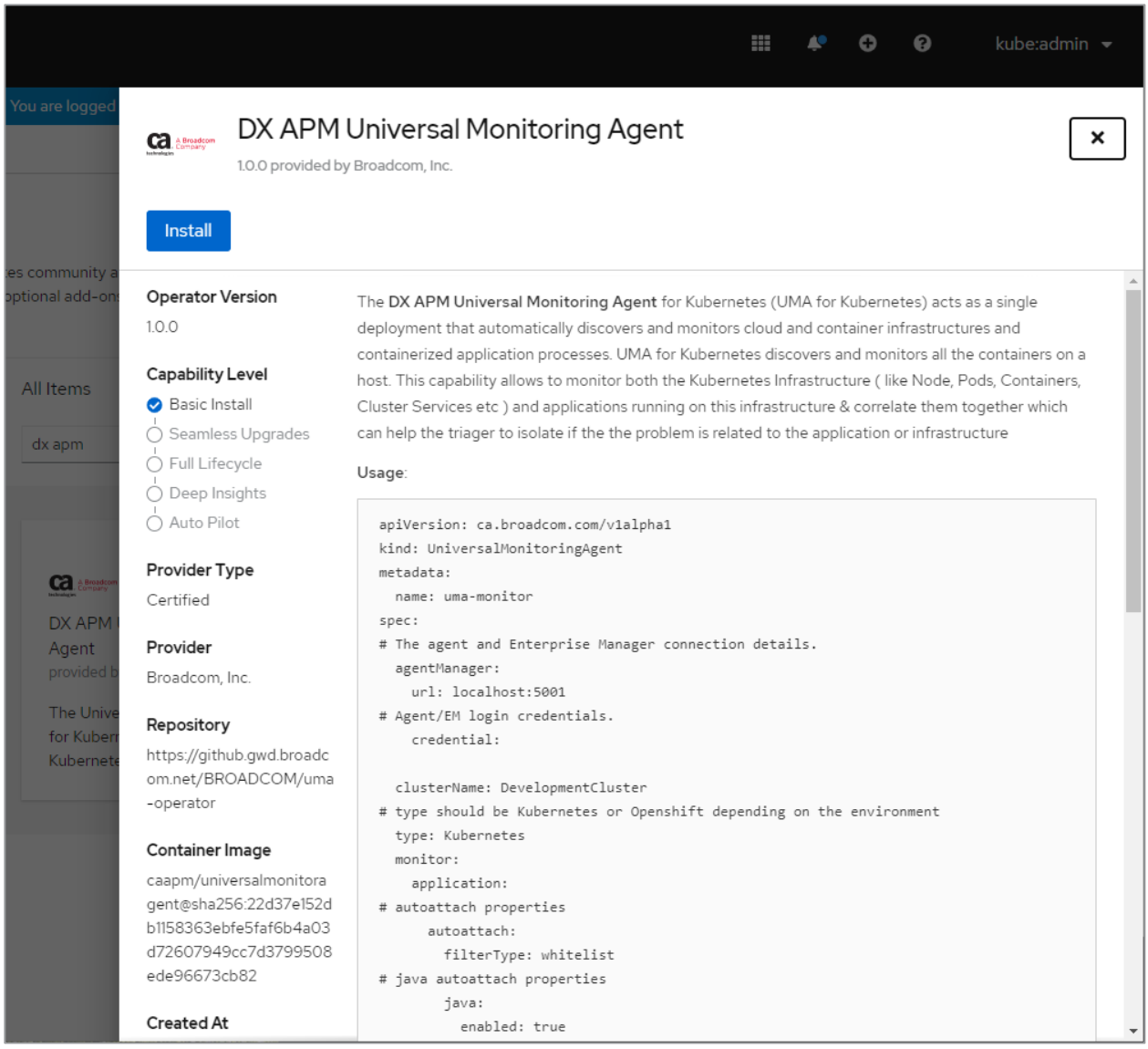
Steps to install an operator are as follows:

First log into the OpenShift console and navigate to "Operator Hub" under the Operators section of the administrator's menu on the left of the screen. Highlighted in red on the following image:

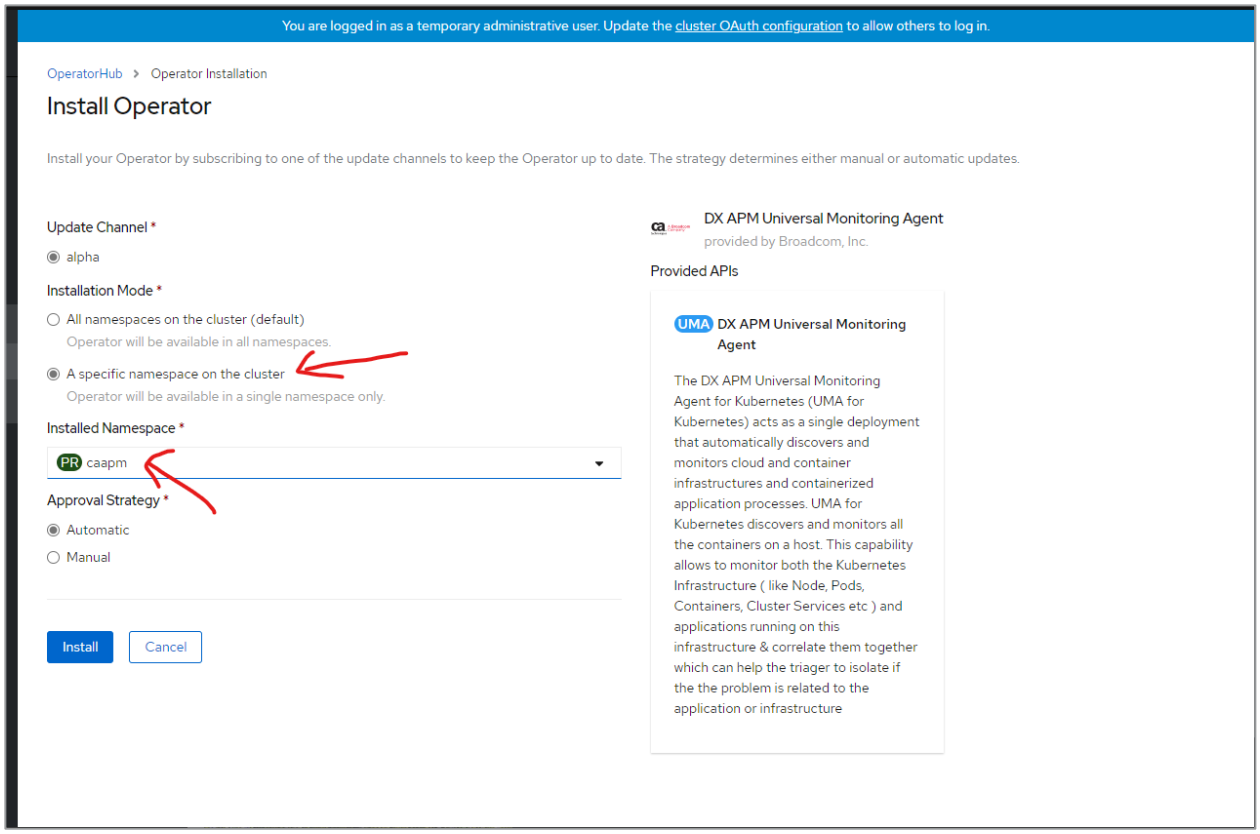


Second, as marked in green in the above image, search for “DX APM” in the Operator Hub and it will return two options. The one marked with Marketplace allows you to purchase a subscription right from the Red Hat Marketplace, the second Operator is if you have an existing subscription or are working with a trial account. That is what we are working with.

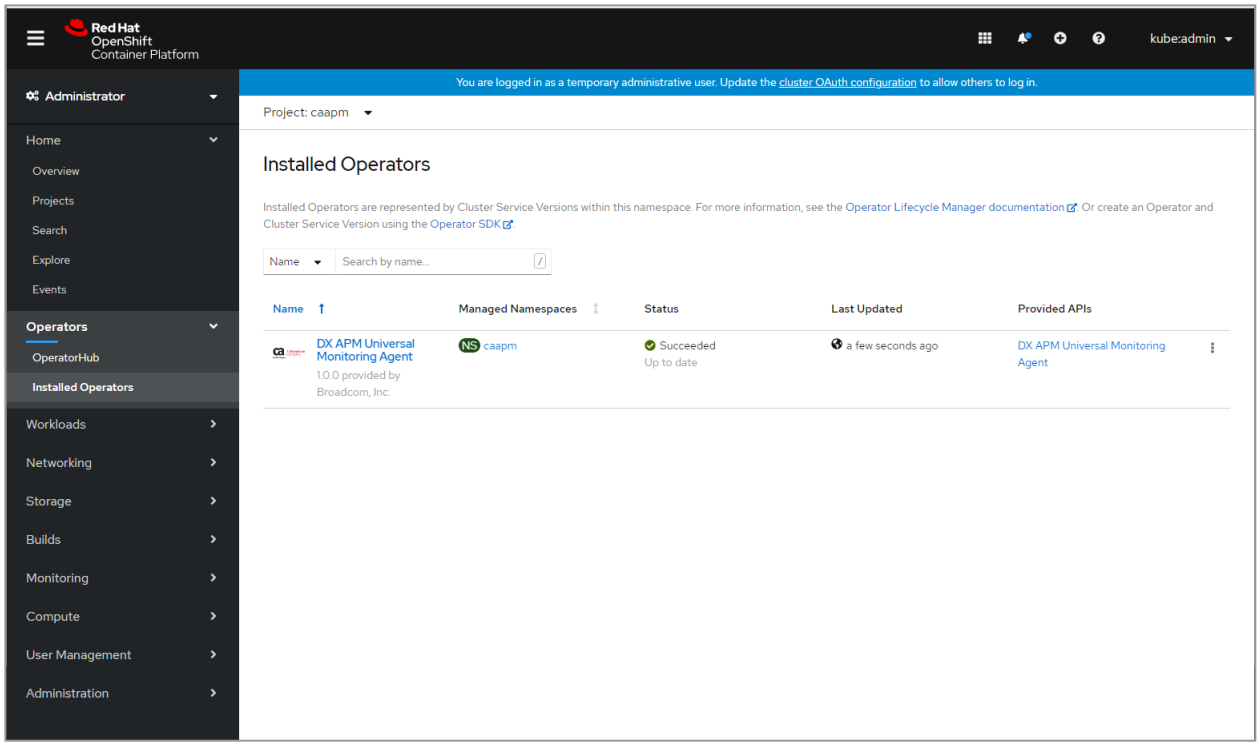
After selecting the DX APM operator, it will give you more details about the product and an install option.



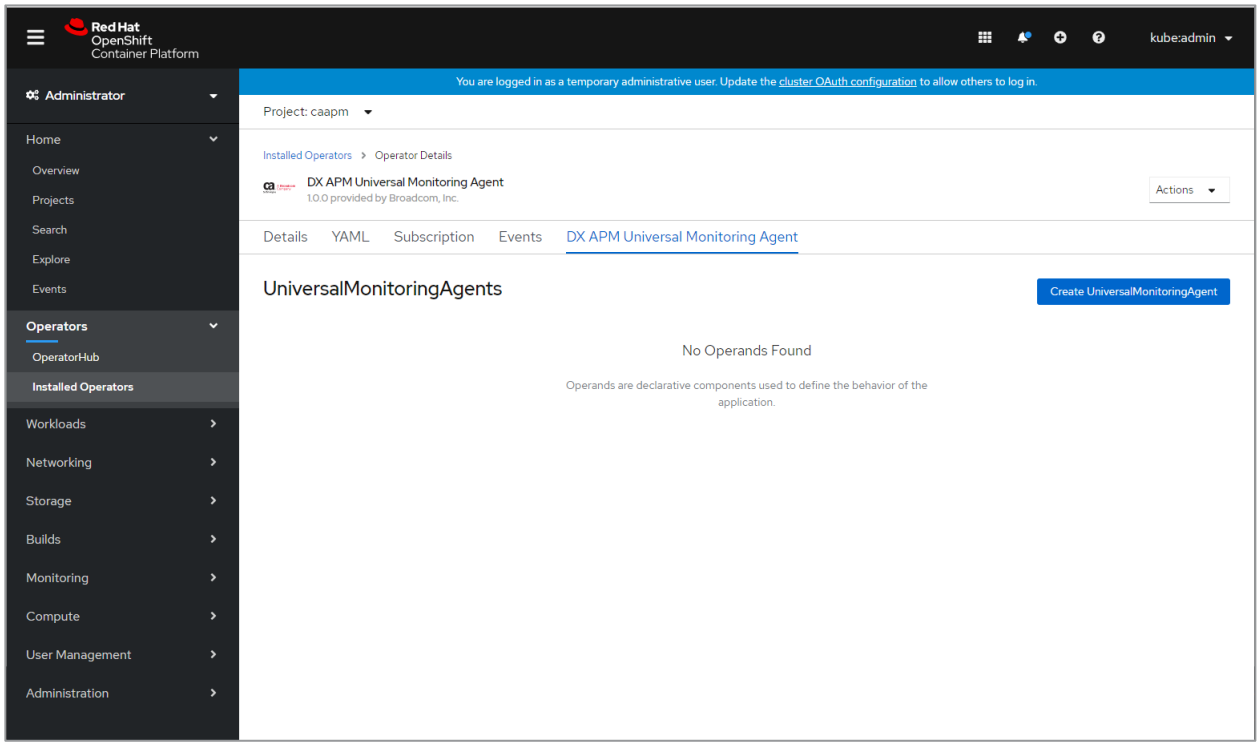
Now that the installation has started, select to use a single namespace. In this case, there was already a “caapm” project created. This can be done from the command line with the command “oc new-project caapm” or in the project section of the web console.



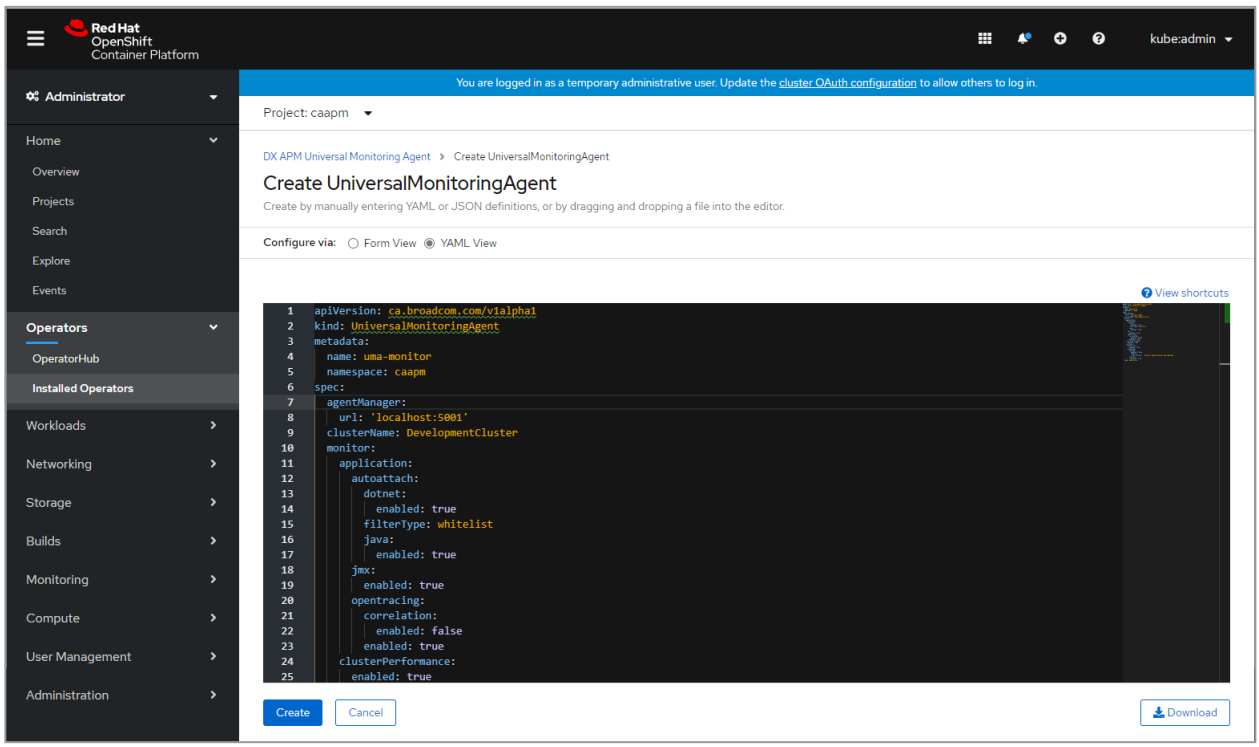
Once install is selected, in a few moments, there will be an installed operator visible under installed operators. If you can not see it then make sure you are on “all projects” or the specific project you used.



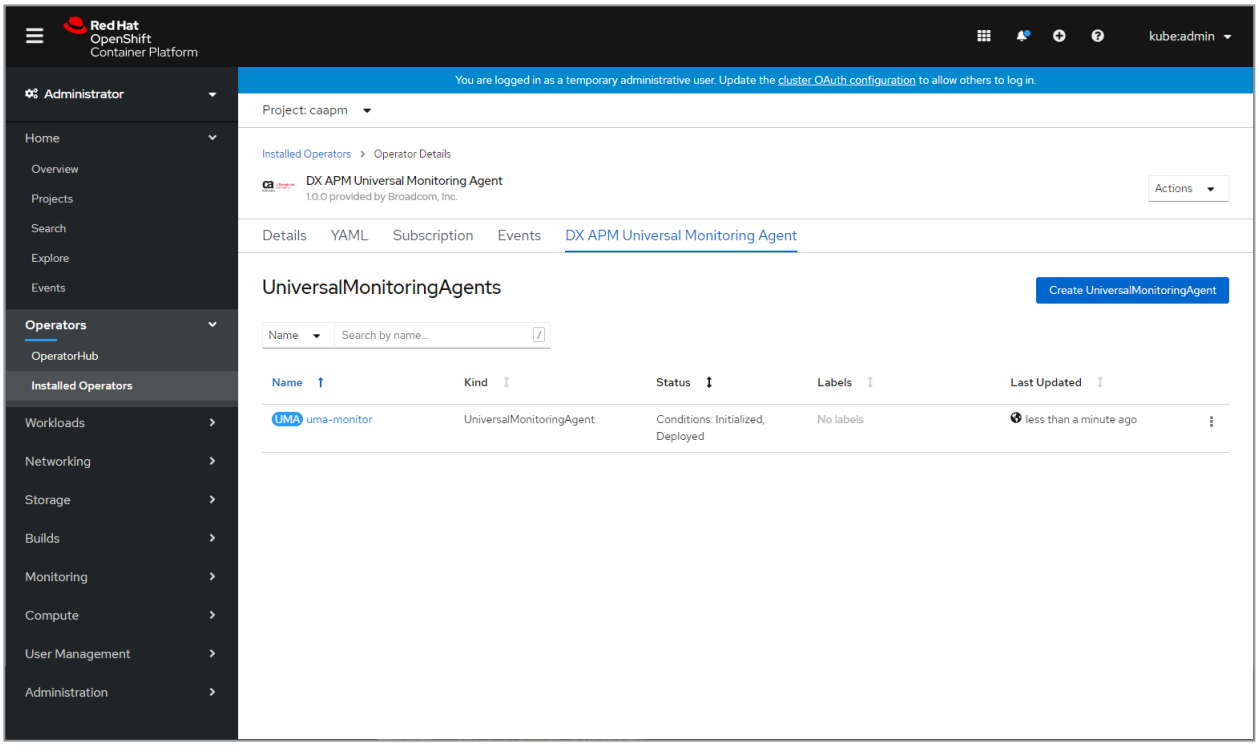
Entering the configuration for the operator brings up a screen with multiple tabs. On this tab select the Agents tab, so we can create an agent to monitor the cluster.



When you create an agent, you can choose two views, a form view or a YAML view. The YAML view is required to configure the address of the DX APM server and to add the required credentials. After those fields are added, select create, and an agent will be up and running almost instantly.



This is how the agent display looks when it is being created and deployed across all nodes as a DaemonSet, so a single instance is running on every node



Now that the agent is installed in the OpenShift cluster. The benefits of this are DX APM gets a holistic look at everything – from the performance of the underpinning infrastructure to be able to monitor any instance of the applications it is monitoring as it scales up and down across the cluster.

Verify Monitoring is Enabled within the OpenShift Cluster

The best and most reliable way to verify that monitoring is to check within the OpenShift cluster for the processes to be running. This can be done in three steps.

First is to select the correct name space, the one that was used as part of the deployment.

```
# oc describe deploy tix-web-deploy
Name:                   tix-web-deploy
Namespace:              tixchange
CreationTimestamp:      Mon, 28 Sep 2020 12:40:01 +0000
Labels:                 app=tixchange-web
                        env=BCP
                        tier=frontend
Annotations:            deployment.kubernetes.io/revision=1
```

```
Selector:               app=tixchange-web,tier=frontend
...
Pod Template:
  Labels:       app=tixchange-web
                env=BCP
                tier=frontend
  Annotations:  ca.broadcom.com/autoattach.enabled=true
                ca.broadcom.com/ca.apm.monitoring.enabled=true
                ver=1.0
  Containers:
   tix-web:
...
    Environment:
      CA_APM_MONITORING_ENABLED:  true
```

The second step is to list all the nodes, so we can verify the deployment is running on every node.

```
# oc describe deploy tix-web-deploy
Name:                   tix-web-deploy
Namespace:              tixchange
CreationTimestamp:      Mon, 28 Sep 2020 12:40:01 +0000
Labels:                 app=tixchange-web
                        env=BCP
                        tier=frontend
Annotations:            deployment.kubernetes.io/revision=1
Selector:               app=tixchange-web,tier=frontend
...
Pod Template:
  Labels:       app=tixchange-web
                env=BCP
                tier=frontend
```

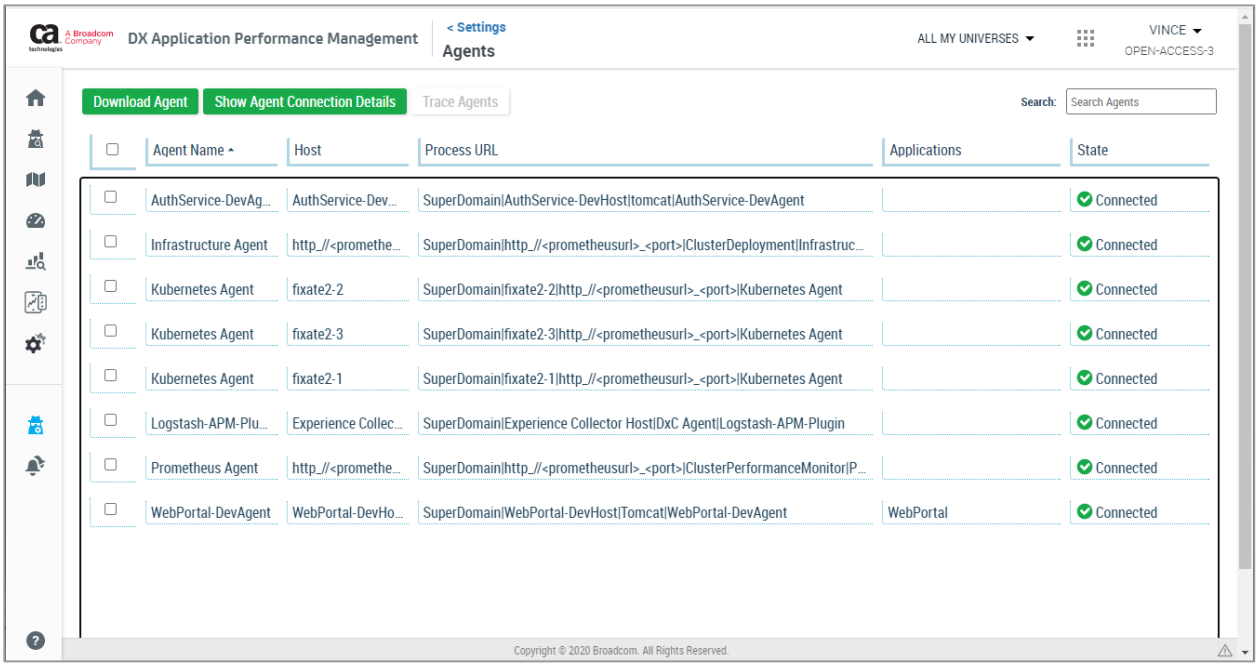
```
 Annotations:  ca.broadcom.com/autoattach.enabled=true
               ca.broadcom.com/ca.apm.monitoring.enabled=true
               ver=1.0
 Containers:
  tix-web:
...
   Environment:
     CA_APM_MONITORING_ENABLED:  true
```

And finally to get a list of all pods running to see there is an app-container-monitor on every node so it can inspect all running processes. In addition, there will be a cluster, clusterinfo, and prometheus pods running which gather metrics from the various available endpoints within OpenShift that expose metrics that should be captured and reported by DX APM.

```
# oc get pods \
  -o custom-columns=NAME:.metadata.name,STATUS:.status.phase,NODE:.spec.nodeName
NAME                                        STATUS    NODE
app-container-monitor-f2sr9                 Running   fixate2-2
app-container-monitor-hs22m                 Running   fixate2-1
app-container-monitor-nlph2                 Running   fixate2-3
cluster-performance-prometheus-657474dd45-qkgt4   Running   fixate2-3
clusterinfo-8f4bfc9d7-r4ws4                 Running   fixate2-3
container-monitor-676968b99c-txlvl          Running   fixate2-2
```

As we can see each node has the appropriate pods running so it is operational.

Once you know monitoring is running in the cluster, it is time to verify logging is registered and being sent into the DX Application Performance Management server through the web interface. After logging into DX APM go to the Agents view on the left side of the menu. There should be several agents listed and connected.



The agents involved are Kubernetes agents for every node in the OpenShift cluster. These gather metrics related to pods, namespaces, containers, and metrics around CPU and memory usage. The full list is extensive and available in the documentation.

Next are Infrastructure Agents and Prometheus agents, which can query all exposed metrics on anything they are told to watch. These metrics often go above and beyond what Kubernetes exposes on its own, as applications can expose custom metrics that Prometheus will autodiscover. OpenShift contains an embedded Prometheus instance just to support this capability; and DX's Kubernetes connector automatically connects to that instance.

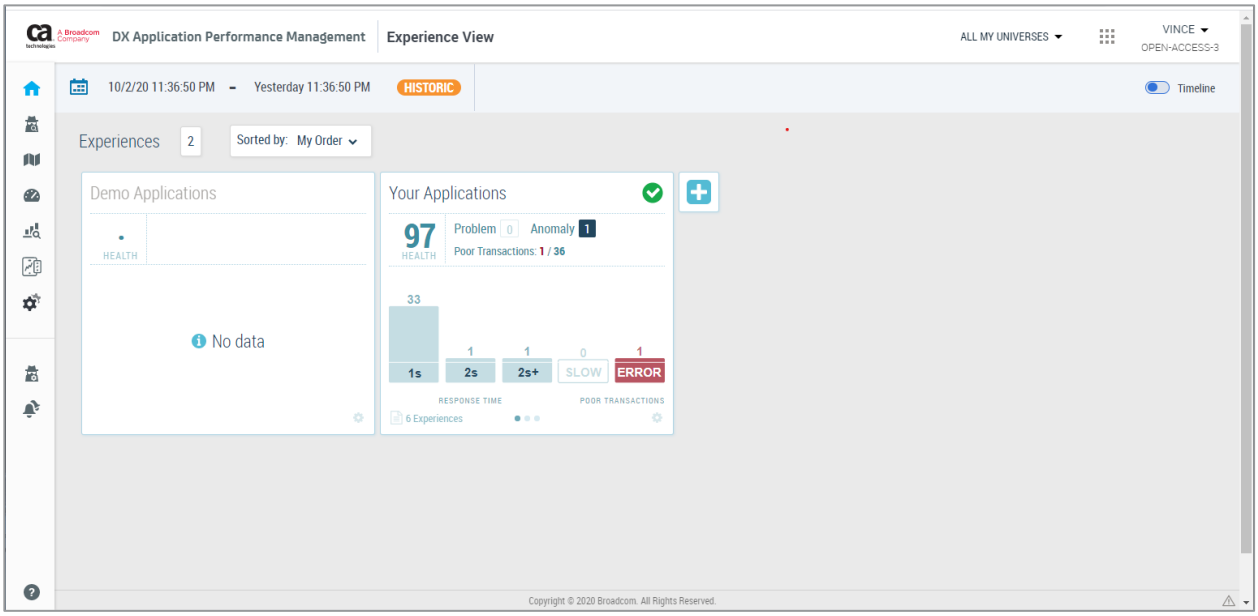
Last is the LogStash-APM-Plugin, which gathers logs from a remote end-point – like nodes in a Kubernetes cluster. It can consolidate them within DX APM to add insight through the AIOps capabilities of the platform.

Full Stack Visibility of an Application

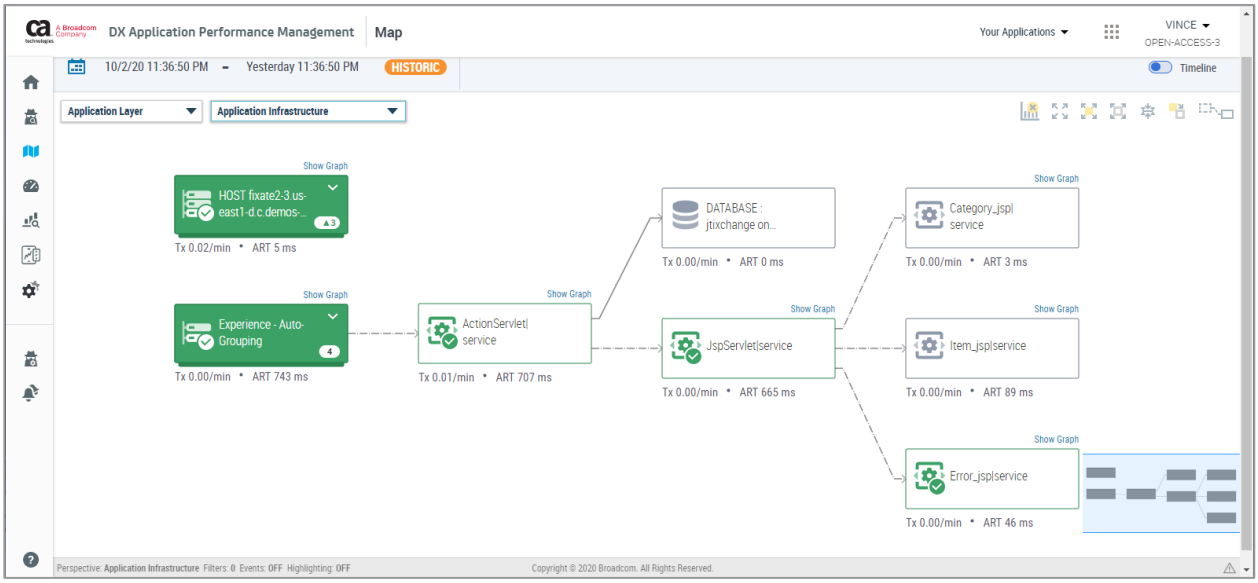
Now that we know monitoring is enabled end to end, we deploy a Java application to show how DX APM detects it, and how it is automatically monitored. By default, any application that sets the environment variable "CA_APM_MONITORING_ENABLED" to "true" will be whitelisted (i.e., approved and permitted) and monitored by the APM suite.

```
# oc describe deploy tix-web-deploy
Name:                   tix-web-deploy
Namespace:              tixchange
CreationTimestamp:      Mon, 28 Sep 2020 12:40:01 +0000
Labels:                 app=tixchange-web
                        env=BCP
                        tier=frontend
Annotations:            deployment.kubernetes.io/revision=1
Selector:               app=tixchange-web,tier=frontend
...
Pod Template:
  Labels:       app=tixchange-web
                env=BCP
                tier=frontend
  Annotations:  ca.broadcom.com/autoattach.enabled=true
                ca.broadcom.com/ca.apm.monitoring.enabled=true
                ver=1.0
  Containers:
   tix-web:
...
    Environment:
      CA_APM_MONITORING_ENABLED:  true
```

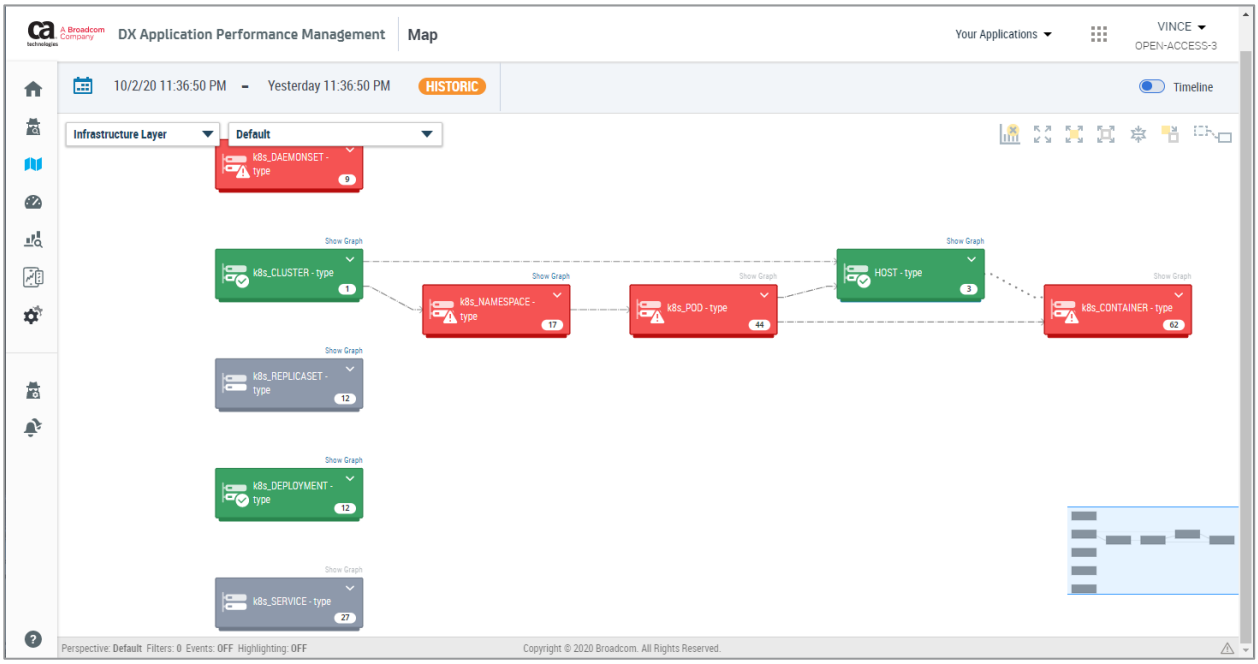
Start with the Experience View screen in DX APM. It shows any applications that you have configured.



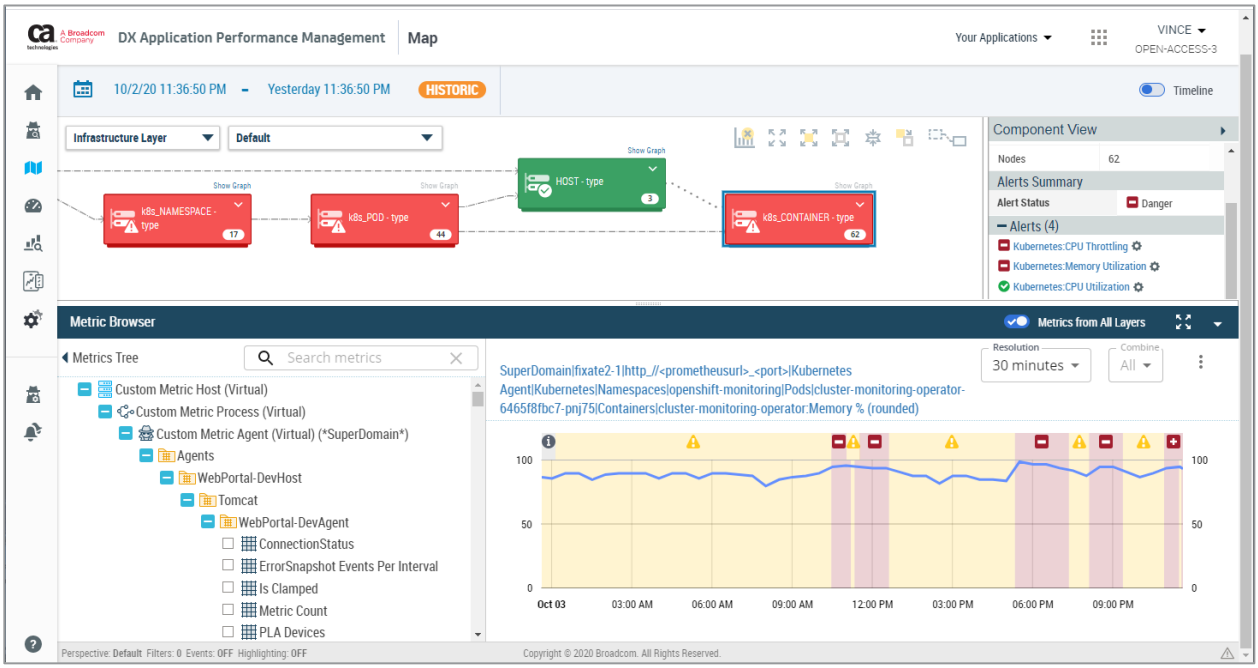
Next, go to the Map section on the left and use the Application Layer view to see a more detailed view of the auto-discovered components of the application. This shows all the auto-discovered components of the application that have been called, plus external calls to a separate container running a Database and how they are related to one another.



You can also switch to an infrastructure view, by selecting the ‘Infrastructure Layer’ from the drop-down. This view shows how the application is deployed within the OpenShift cluster. It includes all the ReplicaSets, Deployments, and Namespaces involved on top of the pods and individual containers. In this specific case, green indicates no errors detected; red shows components that have seen an error in the time window that is being reviewed.



It is often best to start as far to the right as possible with the transaction entry point. This way, you can get to the deepest error first, as it is often the source of the problem. In this case, the error on the container shows it's hitting both memory and CPU limits. The graph shows how close to the limit it has been. Knowing this will allow the development or operations team to make a slight adjustment to the deployment's container specification, and to resolve the problem long before it causes an outage and directly impacts customers.



Next Steps

Learn more about DX APM at www.broadcom.com/apm or check out our documentation page for more in-depth information on how to configure and operate the solution.