K8

Prerequisites

- Docker
- Minikube with kubectl
- (optional) NewRelic Kubernetes Integration to demonstrate overall kube metrics
- nri-flex
- Clone this repository

Configuring Sample Tomcat application with Jolokia

- Using the dockerfile in ***./tomcat*** directory create a docker image

  ```
  docker build -t <dockerid>/java-app:1.0 .
  ```

- Push the image to the repository

  ```
  docker push  <dockerid>/java-app:2.0
  ```

- Update the **tomcat-sample.yaml** to update the image that was created in previous step

  ```
  image: <dockerid>/java-app:2.0
  ```

- Check ./tomcat/tomcat-sample.yaml file to validate flexDiscoveryTomcatJolokia is passed as environment variable to be compliant with V1 Container Discovery feature

  ```
  env:
  - name: flexDiscoveryTomcatJolokia
      value: "t=tomcat,c=jolokia,tt=img,tm=contains"
  ```

Configuring nri-flex for K8

- Download latest nri-flex release *version 0.8.3-pre is already included in this repository*
- Using V1 Container Discovery
  - Create a folder **flexContainerDiscovery** in the root folder
  - Create a new configuration file **jolokia.yml** to use with Container Discovery. In this example we are reading jolokia metrics

```
name: jolokiaFlex
global:
    base_url: http://${auto:ip}:${auto:port}/jolokia/read/
apis:
- event_type: jolokiaMemorySample  ## http://192.168.64.6:31448/jolokia/read/java.lang:type=Me
    url: java.lang:type=Memory
- event_type: jolokiaCatalinaThreadpool  ##  http://192.168.64.6:31448/jolokia/read/Catalina:n
    url: Catalina:name=*,type=ThreadPool
- event_type: jolokiaCatalinaServlet  ## http://192.168.64.6:31448/jolokia/read/Catalina:J2EEA
    url: Catalina:J2EEApplication=*,J2EEServer=*,WebModule=*,j2eeType=Servlet,name=*
```

- Edit the docker file to ADD flexContainerDiscovery folder to be added in arget docker image
  Sample Dockerfile

~~~~
    ```
    FROM newrelic/infrastructure:1.7.1

    # Note if using for Kubernetes you could alternatively set these options in your

    # define license key as below, or copy a newrelic-infra.yml over
    # refer to here for more info: https://hub.docker.com/r/newrelic/infrastructure/
    # ENV NRIA_LICENSE_KEY=1234567890abcdefghijklmnopqrstuvwxyz1234

    # disable the newrelic infrastructure agent from performing any additional monito
    # using forwarder mode will only make it responsible for executing integrations
    ENV NRIA_IS_FORWARD_ONLY true

    # Enable Container Discovery for all orchestrators (for Fargate see below option)
    ENV CONTAINER_DISCOVERY true

    # Enable Container Discovery for Fargate
    # ENV FARGATE true

    # Allow environment variables to be passed through to flex
    # https://docs.newrelic.com/docs/infrastructure/install-configure-manage-infrastru
    ENV NRIA_PASSTHROUGH_ENVIRONMENT="CONTAINER_DISCOVERY,GIT_SERVICE,GIT_REPO,GIT_TO

    # create some needed default directories for flex
    RUN mkdir -p /var/db/newrelic-infra/custom-integrations/flexConfigs/
    RUN mkdir -p /var/db/newrelic-infra/custom-integrations/flexContainerDiscovery/

    # if using container discovery configs uncomment this section
    # https://github.com/newrelic/nri-flex/wiki/Service-Discovery
    ADD flexConfigs /var/db/newrelic-infra/custom-integrations/flexConfigs/
    ADD flexContainerDiscovery /var/db/newrelic-infra/custom-integrations/flexContaine


    # copy config/definition/binary over
    COPY ./nri-flex-config.yml /etc/newrelic-infra/integrations.d/
    COPY ./nri-flex-definition.yml /var/db/newrelic-infra/custom-integrations/nri-fle
    COPY ./nri-flex /var/db/newrelic-infra/custom-integrations/nri-flex

    # eg adding netcat if extra packages are needed
    # RUN apk add --update netcat-openbsd && rm -rf /var/cache/apk/*


    ```
~~~~

- Create docker image

  ```
  docker build -t <dockerid>/nriflex:1.0 .
  ```

- Push the docker image to docker repository

```
docker push <dockerid>/nriflex:1.0
```

- Update ***./nri-flex-k8s.yml*** file to update ***NRIA_LICENSE_KEY*** with your newrelic license key and the docker ***image*** created in last step

```
containers:
- name: nri-flex
    image: "<dockerID>/nriflex:1.6"
    imagePullPolicy: Always
    env:
    - name: NRIA_LICENSE_KEY
    value: "YOUR LICENSE KEY"
```

Deploying Sample app and nri-flex

- Start minikube

```
$ minikube start
😄  minikube v1.5.2 on Darwin 10.15.1
✨  Automatically selected the 'hyperkit' driver (alternates: [virtualbox vmwarefusion])
🔥  Creating hyperkit VM (CPUs=2, Memory=2000MB, Disk=20000MB) ...
🐳  Preparing Kubernetes v1.16.2 on Docker '18.09.9' ...
🚜  Pulling images ...
🚀  Launching Kubernetes ...
⌛  Waiting for: apiserver
🏄  Done! kubectl is now configured to use "minikube"
```

- Start sample app on cluster

```
$ kubectl create -f tomcat/tomcat-sample.yaml
deployment.apps/tomcat created
service/tomcat-service created
```

- Check if the app is up and running

```
$ minikube service list
|----------------------|---------------------------|---------------------------|-----|
|      NAMESPACE       |           NAME            |        TARGET PORT        | URL |
|----------------------|---------------------------|---------------------------|-----|
| default              | kubernetes                | No node port              |
| default              | tomcat-service            | http://192.168.64.7:31003 |
| kube-system          | kube-dns                  | No node port              |
| kubernetes-dashboard | dashboard-metrics-scraper | No node port              |
| kubernetes-dashboard | kubernetes-dashboard      | No node port              |
|----------------------|---------------------------|---------------------------|-----|

$ curl http://192.168.64.7:31003/jolokia/version

{"request":{"type":"version"},"value":{"agent":"1.6.2","protocol":"7.2","config":{"listenForHttpSe
```

- Start nri-flex daemon

```
$ kubectl create -f nri-flex-k8s.yml
daemonset.apps/nri-flex created
```

- Check Insights Data Explorer for availability of Custom Events: ***jolokiaMemorySample, jolokiaCatalinaThreadpool,jolokiaCatalinaServlet***