Container native development

with RHOAR, Node.js, and OpenShift service mesh

Philip Hayes
Architect, Red Hat Emerging Technologies Practice
Thurs 9th May, 2019
Let's take a look at a simple Node.js application architecture
Can we take this Node.js app and deploy it on OpenShift?

Will it work ok?

Is there any problem doing this?
Your application on OpenShift
Do I need to learn about creating Dockerfiles?

How do I communicate things like resource requirements to the platform?

Is there any way to tell if my application is running ok?
How do I accept inbound traffic?

Can I continue to work with my existing IDE or do I need to use new tools?

I really need to focus on the application logic, I don't need any extra overhead.
What happens if the application detects a system fault, how does the application relay this to the orchestration layer?
Customers are looking for guidance from Red Hat Consulting. How can we help developers to adopt container native development practices?
How can your application be a better OpenShift citizen?

- Readiness / liveness probes
- Service discovery
- Deployment configuration
- Graceful shutdown
- Use of config maps
- Tracing
Container Native Development

We define container native development as conforming to the following key tenets:

- Single concern principle
- Self containment principle
- High observability
- Lifecycle conformance
- Runtime confinement
- Process disposability
- Image immutability

Check out the "PRINCIPLES OF CONTAINER-BASED APPLICATION DESIGN" white paper by Bilgin Ibryam (link at end)
Container native Node.js demonstration

- Red Hat Service Mesh setup using the Istio Operator
- Kiali installation
- RHOAR - Node.js configuration
- Node.js boilerplate code for tracing
- Node.js - OCP communication
- Node.js - Swagger definition
- Deploying our app to openshift from our favorite IDE
- Viewing results using the Istio control plane
We are going to use the following technologies for this demonstration:

- RHOAR
- OpenShift Service Mesh
- Kiali
Demonstration
Red Hat Service Mesh installation

- Install the OpenShift Service Mesh: [https://docs.openshift.com/container-platform/3.10/servicemesh-install/servicemesh-install.html#installing-service-mesh](https://docs.openshift.com/container-platform/3.10/servicemesh-install/servicemesh-install.html#installing-service-mesh)
- Or... use istiooc [https://github.com/Maistra/origin/releases](https://github.com/Maistra/origin/releases)
- Installing Istio Operator
- Deploying Istio Control Plane
- Install Kiali
Red Hat Service Mesh installation

- Deploying the Istio Control plane
  - https://github.com/Maistra/openshift-ansible/tree/maistra-0.1.0-ocp-3.1.0-istio-1.0.0/istio

- Installing Kiali
  - https://github.com/kiali/kiali
# Post-installation status

<table>
<thead>
<tr>
<th>Name</th>
<th>Status</th>
<th>Containers Ready</th>
<th>Container Records</th>
<th>Age</th>
</tr>
</thead>
<tbody>
<tr>
<td>kiali/577f150a0f-nginx</td>
<td>Running</td>
<td>1/1</td>
<td>0</td>
<td>2 hours</td>
</tr>
<tr>
<td>jaeger-agent/pado7</td>
<td>Running</td>
<td>1/1</td>
<td>0</td>
<td>2 hours</td>
</tr>
<tr>
<td>jaeger-query/774c9b975b-2122b</td>
<td>Running</td>
<td>1/1</td>
<td>1</td>
<td>2 hours</td>
</tr>
<tr>
<td>jaeger-collector/88857a8d481b81</td>
<td>Running</td>
<td>1/1</td>
<td>12</td>
<td>2 hours</td>
</tr>
<tr>
<td>elasticsearch/c3</td>
<td>Running</td>
<td>1/1</td>
<td>9</td>
<td>2 hours</td>
</tr>
<tr>
<td>grafana/de5c5a477-plt7</td>
<td>Running</td>
<td>1/1</td>
<td>0</td>
<td>2 hours</td>
</tr>
<tr>
<td>influxdb-injector/46c56a87-4p7a</td>
<td>Running</td>
<td>1/1</td>
<td>0</td>
<td>2 hours</td>
</tr>
<tr>
<td>influxdb-pino-7d586c78-5g6b</td>
<td>Running</td>
<td>2/2</td>
<td>0</td>
<td>2 hours</td>
</tr>
<tr>
<td>influxdb-prometheus/7fa4ab6dd-p0b7</td>
<td>Running</td>
<td>1/1</td>
<td>0</td>
<td>2 hours</td>
</tr>
<tr>
<td>influxdb-reconciliation/7573d8c5-apzc</td>
<td>Running</td>
<td>2/2</td>
<td>0</td>
<td>2 hours</td>
</tr>
<tr>
<td>influxdb-protocol-d6e5d569-d9a9</td>
<td>Running</td>
<td>1/1</td>
<td>0</td>
<td>2 hours</td>
</tr>
<tr>
<td>prometheus-84ebd8d9-kmfx</td>
<td>Running</td>
<td>1/1</td>
<td>0</td>
<td>2 hours</td>
</tr>
<tr>
<td>prometheus-c704464-kd9z</td>
<td>Running</td>
<td>2/2</td>
<td>0</td>
<td>2 hours</td>
</tr>
<tr>
<td>ingressgateway-56d9f9eddc-e704</td>
<td>Running</td>
<td>1/1</td>
<td>0</td>
<td>2 hours</td>
</tr>
<tr>
<td>ingressgateway-7c4686bb-gl7b</td>
<td>Running</td>
<td>1/1</td>
<td>2</td>
<td>2 hours</td>
</tr>
<tr>
<td>ingressgateway-7967e678-v0e8f</td>
<td>Running</td>
<td>1/1</td>
<td>0</td>
<td>2 hours</td>
</tr>
<tr>
<td>openshift-namespace-influx-7meppt</td>
<td>Completed</td>
<td>0/1</td>
<td>0</td>
<td>2 hours</td>
</tr>
</tbody>
</table>
Viewing results Kiali, Grafana, Jaeger
RHOAR Node.js configuration

- Nodeshift
  https://www.npmjs.com/package/nodeshift
- Istio Sidecar annotation
- Health endpoints
- Limits
- Port types
- Config Maps
- Labels
Resources

https://developers.redhat.com/products/rhoar/overview/
https://learn.openshift.com/servicemesh/

phayes@redhat.com
THANK YOU

linkedin.com/company/Red-Hat
youtube.com/user/RedHatVideos
facebook.com/RedHatInc
twitter.com/RedHat
THANK YOU

linkedin.com/company/Red-Hat
youtube.com/user/RedHatVideos
facebook.com/RedHatInc
twitter.com/RedHat