Pros and cons of using Kubernetes as a development platform
Who we are

Software Engineer
Eclipse Che committer
workspaces.openshift.com

Software Engineer
Eclipse Che Lead
CNCF Ambassador
Developer Efficiency
“For Phoenix, it takes us **three or four weeks** for new developers to get builds running on their machine...”

Anyone, anytime, can contribute to a project **without installing or configuring** software.
Pros

Using Kubernetes as a CDE Platform
Eclipse Che Architecture Workspace
Cons

The issues we faced building Eclipse Che on top of Kubernetes
Running a CDE in a Kubernetes Pod
Running a CDE in a Kubernetes Pod

- Permission Denied
- OOM Kill
- CPU Throttling
- Immutable
- docker build Not Working
- docker run Not Working
- Pulling Images Time
### Running a CDE in a Kubernetes Pod

<table>
<thead>
<tr>
<th>Issue</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Permission Denied</td>
<td>Specify correct Memory Limit</td>
</tr>
<tr>
<td>OOM Kill</td>
<td>Specify correct CPU Limit (or don’t at all!)</td>
</tr>
<tr>
<td>CPU Throttling</td>
<td>Install packages at build time. Use of a universal developer image.</td>
</tr>
<tr>
<td>Immutable</td>
<td>Use Rootless build and the right Pods Security Context</td>
</tr>
<tr>
<td>docker build Not Working</td>
<td>kubedock</td>
</tr>
<tr>
<td>docker run Not Working</td>
<td>Image Pre-Pulling</td>
</tr>
</tbody>
</table>

**Build images with $HOME R/W for arbitrary unprivileged user.**

**Specify correct Memory Limit.**

**Specify correct CPU Limit (or don’t at all!).**

**Install packages at build time. Use of a universal developer image.**

**Use Rootless build and the right Pods Security Context.**

**kubedock**
Persistence Volumes: Network Attached Storage
Persistence Volumes: Network Attached Storage

- Write Errors
- Exclusive Access Mode
- Mount Timeout
- Quotas
Persistence Volumes: Network Attached Storage

- **Write Errors**
- **Exclusive Access Mode**
- **Mount Timeout**
- **Quotas**

**Use Block Storage Only**

**One Persistent Volume per workspace**

**Ephemeral / Asynchronous Storage (experimental feature)**

**Fail Fast / Explicit Error Message**
Networking: running behind a reverse proxy
Networking: running behind a reverse proxy

- URL Rewrite Support
- TLS Certificate
- Ingress Readiness
- Ingress Quota
Networking: running behind a reverse proxy

URL Rewrite Support

TLS Certificate

Ingress Readiness

Ingress Quota

Backend services have an “external” and an “internal” URL

Use one unique external domain

Use a Threshold of number of success to consider a service available

Delete Ingresses at workspace stop
New Opportunities
Leveraging the latest Kubernetes features for CDE purposes
Conclusion

Is it worth it?
<table>
<thead>
<tr>
<th>Cons</th>
<th>Pros</th>
</tr>
</thead>
<tbody>
<tr>
<td>Starting a container-based CDEs is not faster than VM-based CDEs</td>
<td>CDEs are immutable and software defined</td>
</tr>
<tr>
<td>Developers are not used to code and build in security hardened</td>
<td>Administrators can effectively enforce development environments</td>
</tr>
<tr>
<td>development environments. Some things won’t work.</td>
<td>enterprise policies</td>
</tr>
<tr>
<td>CDE customization is not straightforward</td>
<td>When Kubernetes is the application target platform, the CDE makes it</td>
</tr>
<tr>
<td>Current IDE technologies are not designed to run in the cloud</td>
<td>easy to test it and debug it there</td>
</tr>
<tr>
<td></td>
<td>Kubernetes is evolving rapidly with plenty of new features and</td>
</tr>
<tr>
<td></td>
<td>opportunities at every release</td>
</tr>
</tbody>
</table>
Thank you!

Join the conversation:

@EclipseCon | #EclipseCon