Oracle Weblogic on Kubernetes

Rationale

<table>
<thead>
<tr>
<th>CERN IT-DB group:</th>
<th>WebLogic Domains as a Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMs provisioning: Openstack IaaS</td>
<td>From... towards...</td>
</tr>
<tr>
<td>Software deployment: CERN RPMs</td>
<td>K8s cluster provisioning: Openstack Magnum PaaS</td>
</tr>
<tr>
<td>Configuration management: Puppet</td>
<td>GitLab + Docker images + Kubernetes</td>
</tr>
<tr>
<td>Service configuration definition: JSON</td>
<td>Continues integration and deployment, configuration and orchestration:</td>
</tr>
<tr>
<td>Configuration deployment: JYTHON</td>
<td>Service configuration definition: JSON</td>
</tr>
</tbody>
</table>

Why WLS on Kubernetes?

- Recoverable
- Faster delivery
- Repeatable
- Plug-able
- Industry standard
- Portable
- Version controlled
- Easier patching

Architecture

- Developers (users of WebLogic domains) upload applications via Web interface to dedicated location in storage system
- Protection and security of external resources:
  - Applications deployed in WebLogic use dedicated Oracle databases
  - Databases reject any connection from unknown hosts
  - List of IP's of nodes running a given K8s cluster are kept in network sets
  - Firewall rules for database services are defined based on the contents of the network sets
- Isolation and security of deployed applications:
  - K8s Clusters dedicated to different categories of applications
  - WebLogic domains, deployed in individual namespaces
- High available front-end:
  - HAProxy + Pacemaker + Corosync
  - Canary deployment: roll out new release to a subset of users

Deployment model

- Streamlined production of images with GitLab:
  - Changes are moderated, controlled and driven by code using merge requests
  - Automated build of development, staging and production images
  - Maintenance of several versions in parallel
  - Storage in the GitLab's integrated image registry

- CLI tools for WebLogic domains deployment:
  - Set of scripted tools to interface with the infrastructure
  - Consume from external sources, information like configurations of WebLogic domains and front-end endpoints in JSON format
  - Generate description of K8s objects and connect to appropriate Magnum clusters to create/destroy/modify

Challenges

- Cloud provider independent
- Portable deployment
- Pre-configured WebLogic domains included in the Docker images
- Persist specific WebLogic domain configuration in Kubernetes objects
- WebLogic to download deployed application from WebService at start-up time
- Cultural change of containers ecosystem
- No access to particular hosts
- Logging
- Debugging

References

- Antonio Nappi (September 2017, CERN Openlab Open Day) [http://cern.ch/go/WW7B]
- WebLogic on Docker images [http://cern.ch/go/8Spk]
- GitLab CI [https://about.gitlab.com/features/gitlab-ci-cd/]

Future plans

- Evaluate Helm [https://helm.sh/] and GitLab Auto DevOps [https://docs.gitlab.com/ee/topics/autodevops/] for operations and deployment
- Evaluate Traefik [https://traefik.io/] as alternative to HAProxy ingress
- Kubernetes RBAC Authorization [http://cern.ch/go/L96] to delegate operator role to application developers