Open Horizon
Lightning Talk

Glen Darling (IBM)

#ossummit
What is Open Horizon?

• The most powerful open source project you’ve never heard of!
• Edge fleet management for containerized software and data files
  — on most types of Kubernetes clusters
  — on most stand-alone Linux, using docker or equivalent (with 512MB or +)
  — on most hardware architectures (x86, ARM, others)
  — at massive scale
• Open source, with open governance under the Linux Foundation’s LF-Edge:
  https://www.lfedge.org/projects/openhorizon/
Edge Computing Examples...

- Edge computing is about bringing computation assets (HW, SW) close to the input data and close to where actions must occur.
- Edge use cases are many and varied:
  - Conversational AI
  - Security video
  - Autonomous vehicles
  - Loss prevention
  - Patient monitoring
  - Factory automation
  - Traffic Management
  - and many, many more!
- Here are two examples where Open Horizon is being used today...
Edge Computing Examples....

- Visit https://mas400.com/ for more details
- To watch live (camera video and/or numeric telemetry):
  - https://mas400.com/dashboard#live
Architecture – Quick Snapshot

- Agents are autonomous, firewalled, driven by policies you set
- Nothing ever initiates contact with Agents, Agents always initiate
- Edge node IP addresses are kept private (for security/privacy)
- All code and data files and deployment details are all cryptographically signed
- Highly decentralized, scales extremely well
- Agents continue to function when disconnected from Hub
- A compromised Management Hub cannot take control of Agents!
- All comms (even between internal components) are encrypted
- Agent/AgBot comms have “perfect forward secrecy”
- Model Management System enables independent lifecycles for code and data
- This also enables model update with zero downtime
- Secrets Manager enables Agent to share secrets with containers at runtime
Open Horizon Policies

• Policies are the basis for the autonomy of the Open Horizon Agent
  — Simpler Deployment Patterns can be used instead, but they are less capable
• Policies can be attached to edge nodes, services, and deployments
• Policies always contain properties and/or constraints
• Properties are simply name/value pairs. E.g.:
  — "name": "HasCamera", "value": true
• Some are defined automatically by the Agent (e.g., node attributes)
  — All of the Agent-defined properties have an “openhorizon.” prefix
• Constraints are logical expressions in terms of Properties. E.g:
  — “openhorizon.memory >= 2000”
• The constraint language is rich with many operators
Open Horizon Policies

• Each Agent independently runs constraint resolution based upon:
  — The policies attached to its own node, and
  — Any relevant Deployment Policies (which the AgBots suggest to them), and
  — Any (optional) Service Policies attached to the Service in that deployment, then
  — Matches all those Property values to all of those Constraint expressions, and...
  — Evaluates the result (which is specific for its node only)
  — When the result is "true", the Service will be verified then deployed to its node

• A tool is provided to facilitate modeling and to debug deployments
  — hzn deploycheck ...

• This tool enables you to see the effect of Policy changes ahead of time without actually applying them across your fleet
Open Horizon Usage

- Edge nodes can be registered using their SDO/FDO voucher:
  - `hzn voucher import ...
- Or edge nodes can be manually registered:
  - `hzn register ...
- Then you can publish a Service definition (JSON) for your container
  - `hzn exchange service publish ...
- And publish a deployment pattern (JSON) or deployment policy (JSON):
  - `hzn exchange pattern publish ...
  - `hzn exchange deployment addpolicy ...
- Assuming your Policy Constraints all resolve using your Policy Properties, your Agents will deploy your Service to the appropriate set of your nodes
Open Horizon Usage

• All Policies may contain Properties and/or Constraints, as you wish
• Deployment Policies must also identify the specific Service they will deploy
• How I normally use Policies:
  — Normally my Node Policies contain only Properties (no Constraints)
    I use these to identify any node specifics: its capabilities, its intended role, etc.
  — Normally I omit Service Policies, but developers can use these to specify requirements for their
    Service containers (e.g., memory requirements)
  — Normally my Deployment Policies contain only Constraints (no Properties)
    I use them to orchestrate the Service deployment any way I want
    E.g., I can essentially say:
    “deploy to all smart cameras with Intel Movidius VPUs that are installed for shelf monitoring in the cereal aisles of all stores in Texas”
• When the scale is large (many thousands of nodes) this ”intent” approach is
  much easier to use than the common prescriptive/declarative approach
Want to learn more? Please join my Open Horizon Talk on Thursday.
Open Horizon: Videos, Docs, Example Code…

- Open Horizon playlist on the LF Edge YouTube channel:
  - https://bit.ly/34o9Qn4
  (tech deep dives, flow animations, hands-on demos)
- Open-Horizon documentation:
  - https://open-horizon.github.io
- Open-Horizon GitHub (source code)
  - https://github.com/open-horizon
- Examples ready for Open-Horizon:
  - https://github.com/open-horizon/examples
  - https://github.com/open-horizon-services/
To contact us on the Linux Foundation chat system, first get a Linux Foundation ID (free) from here:

https://identity.linuxfoundation.org/

Then join one of the “open-horizon-...” channels, e.g., the Examples Working Group channel, here:

https://matrix.to/#/#open-horizon-examples:chat.lfx.linuxfoundation.org

Or send me an email:

glendarling@us.ibm.com