Open Horizon project Agent Working Group Meeting

August 24, 2020

OPEN HORIZON

THE LINUX FOUNDATION

LF EDGE

Meeting Details

August 24, 2020 (@ 11:30am ET/8:30am PT)
Meeting Info: https://zoom.us/j/94980775985

Meeting ID: 949 8077 5985
One tap mobile
+13462487799,,94980775985# US (Houston)
+16699006833,,94980775985# US (San Jose)

Dial by your location
+1 346 248 7799 US (Houston)
+1 669 900 6833 US (San Jose)
+1 253 215 8782 US (Tacoma)
+1 312 626 6799 US (Chicago)
+1 929 205 6099 US (New York)
+1 301 715 8592 US (Germantown)
888 788 0099 US Toll-free
877 853 5247 US Toll-free
Meeting ID: 949 8077 5985
Find your local number: https://zoom.us/u/av0XQgb3W

LF Antitrust Policy Notice

LF Edge meetings involve participation by industry competitors, and it is the intention of the Linux Foundation to conduct all of its activities in accordance with applicable antitrust and competition laws. It is therefore extremely important that attendees adhere to meeting agendas, and be aware of, and not participate in, any activities that are prohibited under applicable US state, federal or foreign antitrust and competition laws.

Examples of types of actions that are prohibited at LF Edge meetings and in connection with Linux Foundation activities are described in the Linux Foundation Antitrust Policy available at http://www.linuxfoundation.org/antitrust-policy. If you have questions about these matters, please contact your company counsel, or if you are a member of the Linux Foundation, feel free to contact Andrew Updegrove of the firm of Gesmer Updegrove LLP, which provides legal counsel to the Linux Foundation.

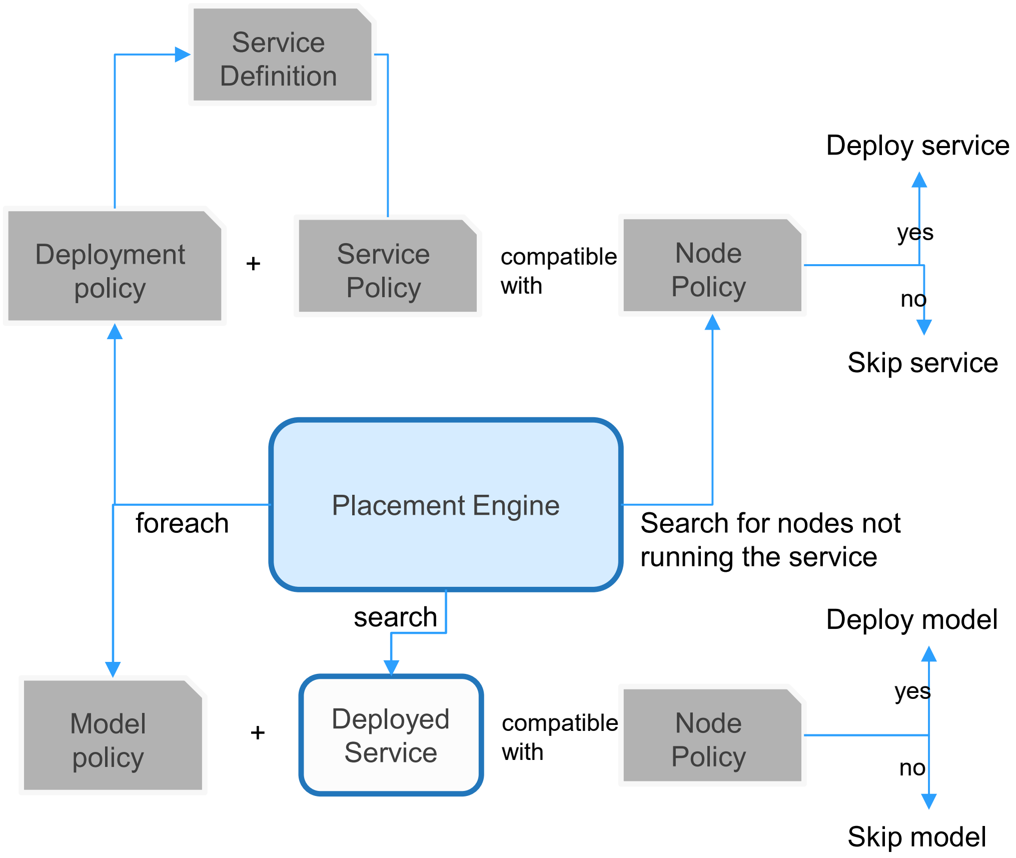
Topics

1. Welcome and Introductions for any new attendees
   A. Reminder of commit signoff
      1. https://wiki.lfedge.org/display/OH/Project+Contributions
   B. Any other administrative matters to address?

2. Deploying services by Policy

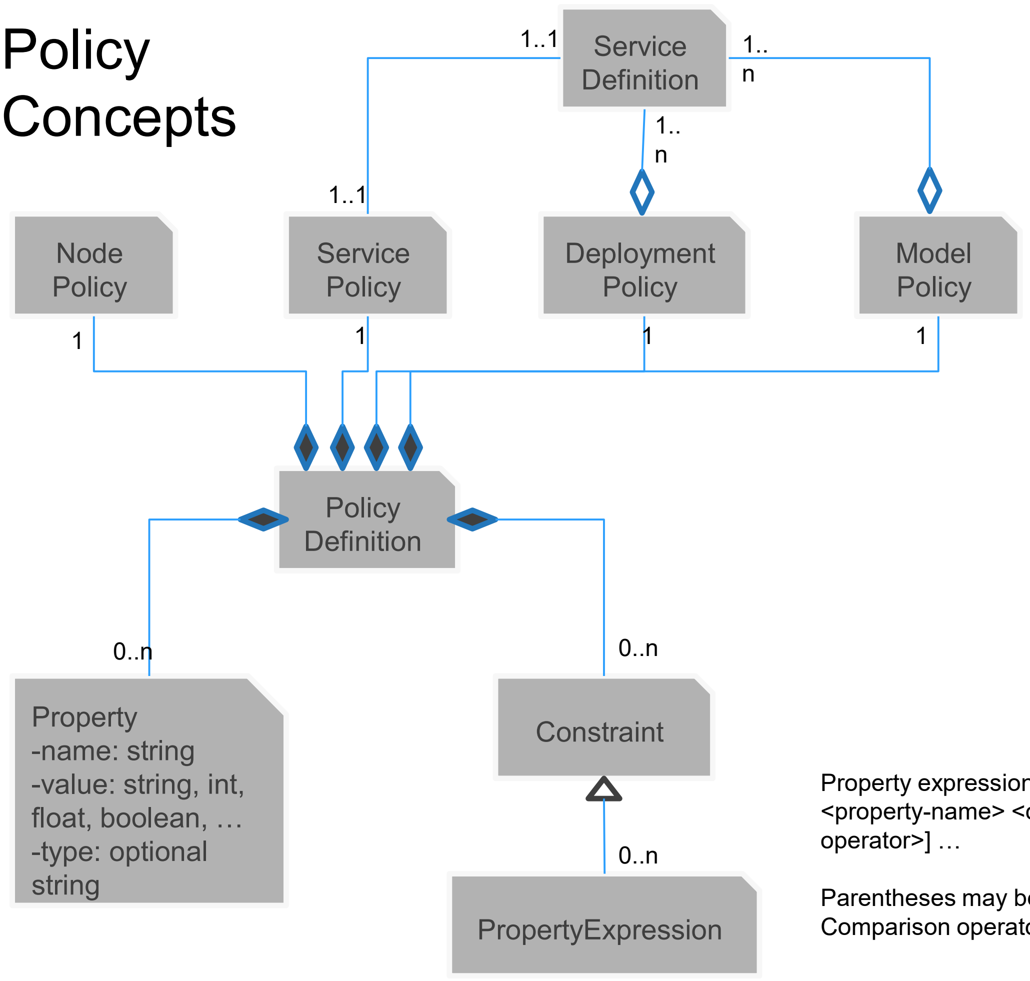
3. Next Meeting

Service Placement



- The placement engine runs in the Agbot.
- Continually searches for nodes that are compatible with existing deployment policies.
- Continually searches for deployed services that have model policy which is compatible with the node where the service is running.
- Deployment policy is logically ‘AND’ed with service policy to form the full set of policy definitions.
- Service policy is optional, specified by the service author.
- Compatibility is bidirectional:
  - the node policy satisfies the combined deployment policy
  - The combined deployment policy satisfies the node policy

Policy Concepts



In English:

- A policy definition is composed of 0 or more properties, and 0 or more constraints.
- A Property has name, type (optional), and value.
- A Constraint has zero or more property expressions.
- A node, service, deployment and model policy each have 1 policy definition.
- A service definition and a service policy are independent entities with a 1..1 relationship.
- A deployment policy and a model policy are independent entities that each refer to a service definition.
- A deployment policy may refer to more than 1 version of a service to denote service rollback instructions.
- A model policy may refer to more than 1 service definition, for each service the model is going to be deployed with.

Property expression form:
<property-name> <comparison-operator> <property-value> [<boolean-operator>] …

Parentheses may be used to provide precedence order.
Comparison operator depends on property type.

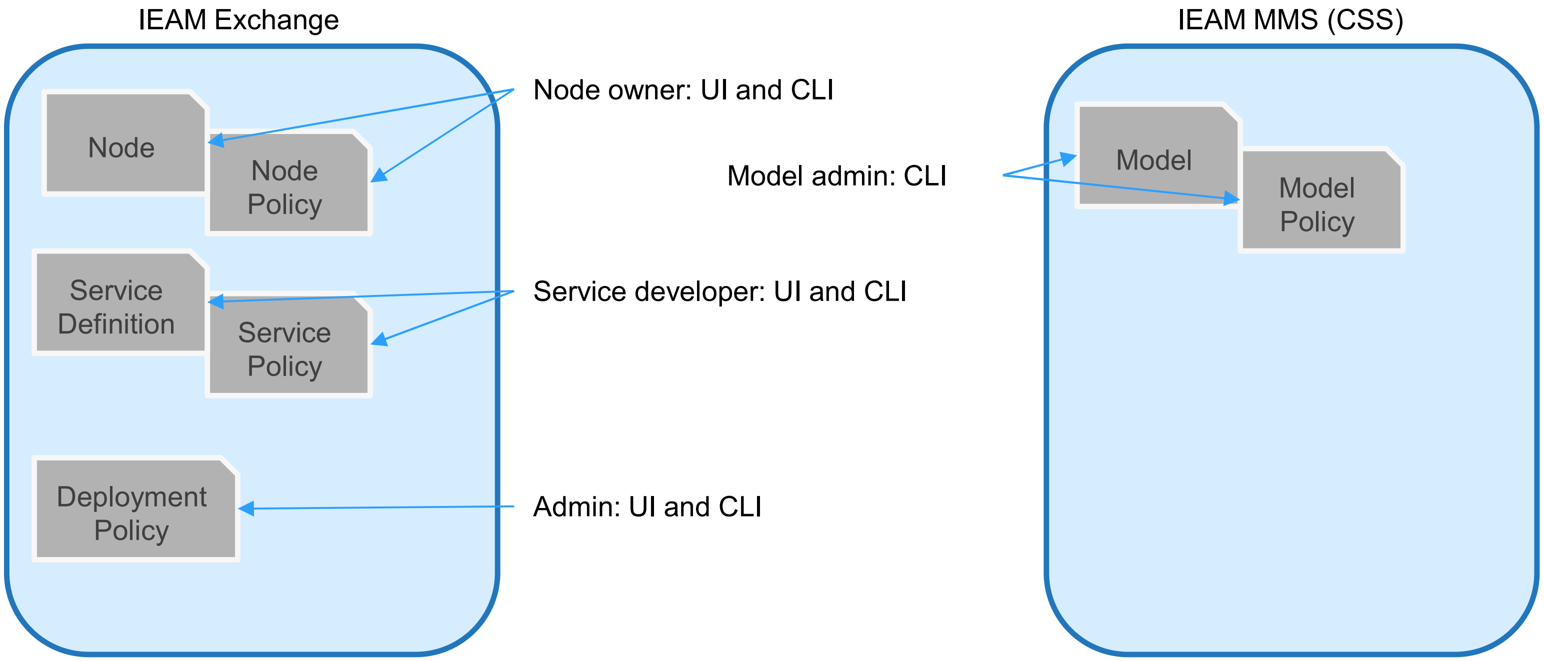
Policy Definition

```
{
  "properties": [   /* A list of name value pairs that describe what they are
attached to. */
    {"name": "robot","value": true}
  ],
  "constraints": [  /* A list of constraint expressions of the form <property
name> <operator> <property value>, separated by boolean operators AND
(&&) or OR (||). */
    "purpose=asset-utilization-monitor OR purpose=assess-paint-quality"
  ]
}
```

Policy Definition notes

- Properties are defined by the user at the time of use.
- Names are arbitrary to allow the user to define any naming scheme they desire.
- Node, deployment, and model policy can be changed at any time and the placement engine will react.
- Service policy can be changed at any time, but it is not a best practice to do so. If a service policy needs to change then it should be changed with a corresponding change to service version.
- Node policy has built-in properties, prefixed with openhorizon:
  - allowPrivileged, arch, cpu, hardwareId, memory
  - allowPrivileged is how a node opts into running privileged containers, it is the only built-in property that is read/write. All others are readonly.
- Service policy has builtin properties, prefixed with openhorizon:
  - allowPrivileged, arch, name, org, url, version
  - allowPrivileged in introspected from the associated service definition.
- All policy definitions are stored in the exchange (DB).

Policy Authoring



IEAM Exchange

Node

Node Policy

Service Definition

Service Policy

Deployment Policy

Node owner: UI and CLI

Service developer: UI and CLI

Admin: UI and CLI

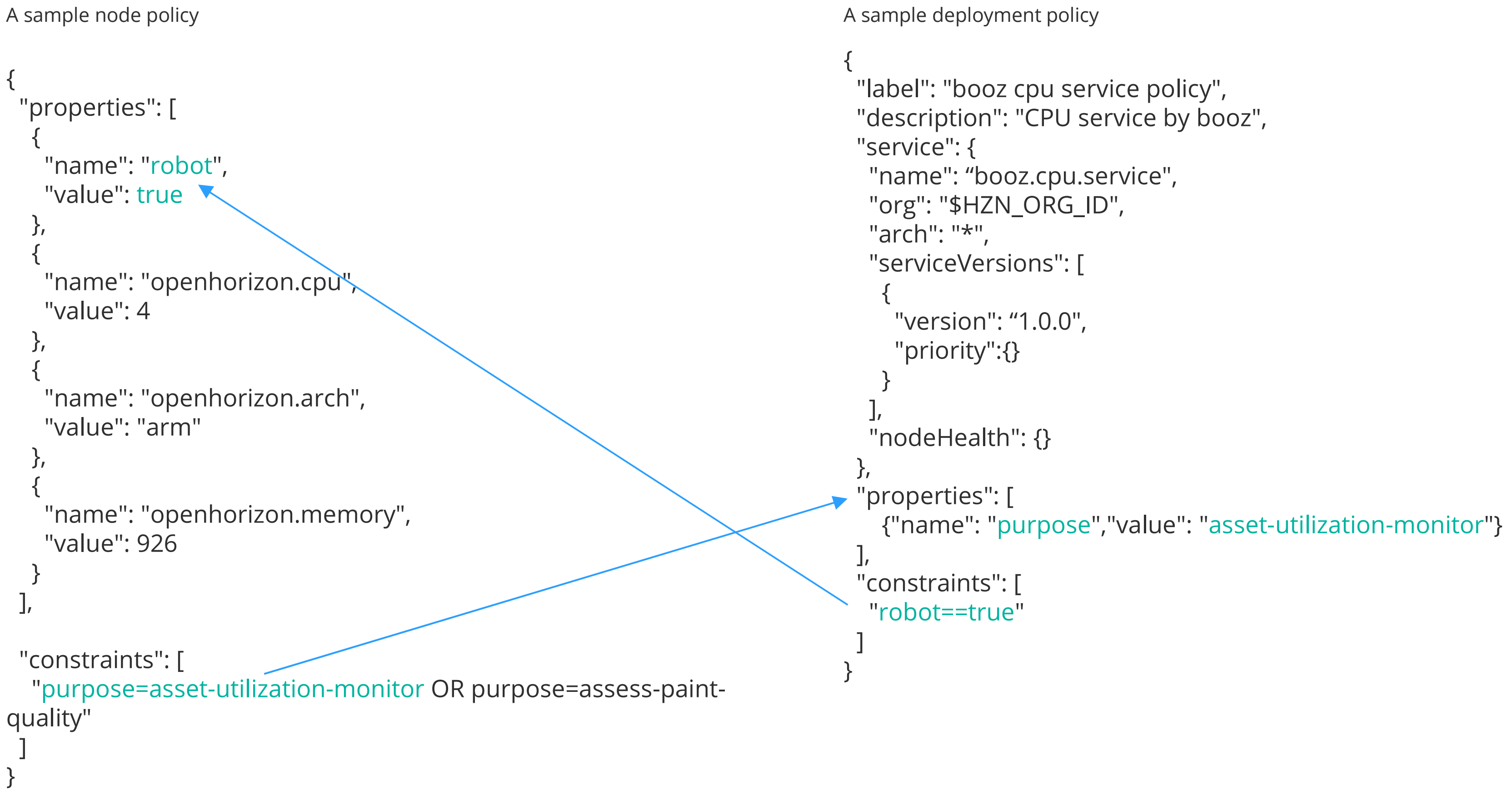
Model admin: CLI

IEAM MMS (CSS)

Model

Model Policy

Policy compatibility example - bidirectional



A sample node policy

```
{
  "properties": [
    {
      "name": "robot",
      "value": true
    },
    {
      "name": "openhorizon.cpu",
      "value": 4
    },
    {
      "name": "openhorizon.arch",
      "value": "arm"
    },
    {
      "name": "openhorizon.memory",
      "value": 926
    }
  ],

  "constraints": [
    "purpose=asset-utilization-monitor OR purpose=assess-paint-quality"
  ]
}
```

A sample deployment policy

```
{
  "label": "booz cpu service policy",
  "description": "CPU service by booz",
  "service": {
    "name": “booz.cpu.service",
    "org": "$HZN_ORG_ID",
    "arch": "*",
    "serviceVersions": [
      {
        "version": “1.0.0",
        "priority":{}
      }
    ],
    "nodeHealth": {}
  },
  "properties": [
      {"name": "purpose","value": "asset-utilization-monitor"}
  ],
  "constraints": [
    "robot==true"
  ]
}
```

Deployment Checker – Verify policy compatibility

- Suppose you have a node policy in a file that you want to apply to a set of devices when you register them. You also have a deployment/business policy that is already in the exchange. You want to use that deployment policy to deploy a service to those nodes. The question; is the node policy compatible with the deployment policy? Or stated another way, will IEAM deploy the service to these nodes or not?

```
# hzn deploycheck policy --node-pol node.policy.json -b booz.top-service.policy
{
  "compatible": false,
  "reason": {
    "mycluster/booz.top-service_1.0.1_amd64": "Policy Incompatible: Compatibility Error: Node properties do not satisfy constraint requirements. The required properties (robot==true, openhorizon.allowPrivileged==true) AND service-tree-test=true were not found in the available properties robot=true, model=100, single-service-test=true, service-tree-test=false"
  }
}
```

Next Meeting

THE LINUX FOUNDATION

LF EDGE

Next Meeting

- Monday Sept 7 is a US Holiday, Labor Day.
- Next Meeting: Monday, Sept 21 @ 8:30am PT/11:30am ET

Thank You

THE LINUX FOUNDATION

LF EDGE