Presenters

Cindy Xing  
Huawei  
@cindyxing

Dejan Bosanac  
Red Hat  
@dejanb

Preston Holmes  
Google  
@ptone

Steve Wong  
VMware  
@cantbewong
Agenda

• Intro
• Edge workloads
• Workload challenges
• Q&A
Some Use Case Examples

- ROBO
- Smart Cities
- Remote Sensors
- Machine Learning
- Video Surveillance
- Gaming
- IoT
- Autonomous Vehicles
- MEC
- Infrastructure edge
- Industrial IoT (IIoT)

Plus many more
### Edge/IoT vs Cloud Data Center

**Kubernetes has limitations today..but can we fix this?**

<table>
<thead>
<tr>
<th>Similar</th>
<th>Different</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manage many nodes with compute, storage, networking</td>
<td>Resource constrained (small node counts, low power CPUs, low resources)</td>
</tr>
<tr>
<td>Want to containerize apps and services</td>
<td>Special network requirements (protocols, topologies)</td>
</tr>
<tr>
<td>Want standardized APIs, tools</td>
<td>Challenges of unattended, and disconnected operation</td>
</tr>
<tr>
<td>Want security features (encryption, authentication)</td>
<td></td>
</tr>
</tbody>
</table>
How does Kubernetes interact with the Edge?

1. Edge workloads that run ON Kubernetes
2. Edge challenges mitigated BY Kubernetes
3. Edge capabilities not easily serviceable by Kubernetes
Edge workloads - Why?

- Data ingestion and processing
  - Protocol conversion
  - Data preprocessing
- Reliability and availability
  - Buffer and batch
  - Caching
- Latency
  - Edge functions
  - Compute offloading
  - Machine learning
Protocol conversion

• Network level
  • Converting non-IP protocols to TCP/IP based ones
    • Modbus in IIoT
    • Bluetooth in consumer IoT
  • Usually converting to some widely used messaging protocol
    • MQTT
    • AMQP
    • HTTP

● Kubernetes supports "device plugins"
● Taints and tolerances can be used for scheduling to appropriate nodes
● New concepts for easier access to interfaces
  ○ https://www.networkservicemesh.io/
Data preprocessing

- Convert data to general structured messages
- Normalize data structure
  - Vorto, LWM2M
- Data analytics
  - Send only relevant data
  - Combine multiple sources
- Add metadata
  - Location
  - Identity
  - Security

Generic Kubernetes workloads
Needs to be properly containerized and orchestrated on the Edge nodes
Reliability and high availability

- Buffer and batch
  - Store and forward
  - Brokers on Edge nodes
- Caching
  - Local databases on Edge nodes
  - Sync data with the cloud and other Edge nodes

Edge Clusters may have limited storage volumes to hold data until it can be uploaded.
Latency: Functions

• React locally on sensor or scheduled events

• Possible CNCF projects collaboration
  • Cloud Events - https://cloudevents.io/
  • Knative - https://github.com/knative/
Latency

- Compute offload
  - Schedule resource intensive tasks on the dedicated hardware on the Edge
  - Example AR/VR renderings
- Machine learning
  - Cloud trained models - executed on the Edge
  - Edge specific training (environment and data policies)

Taints and tolerances can be used for scheduling to appropriate nodes (e.g. GPU availability)
Workload Challenges
Workload challenges

- Limited node resources
  - Workload prioritization
  - Alternative Kubelet implementations (Virtual Kubelet or KubeEdge)
- Unreliable/Limited network
  - Traffic shaping
  - KubeEdge Architecture (EdgeController and EdgeBus)
Workload prioritization - Why

- Limited number of nodes on the Edge
- No autoscaling
- Workloads with wide range of priorities
- Adds more emphasis on prioritization
Kubernetes prioritization toolkit

Prioritization
- Ranking of priority classes
- Input to pre-emption logic
- Applied to a pod, but acted on by node
- Different from resource based eviction

Quality of Service
- Three levels
  - Guaranteed
  - Burstable
  - Best Effort
- These are implicit from pod spec
- Is NOT considered for preemption
- IS considered in the case of eviction
- preemption != eviction
Traffic shaping - Why

- Managing bandwidth
- Network capacity can be limited
- Different workloads should have different network policies
- Related to "Workload Prioritization"
Traffic shaping

Policy

- Deals with what traffic is allowed
- Applied via Network Plugin
- Creates NetworkPolicy resource
- Based on 'cluster-external' IPs
- Based on SRC/DST and port
  - src/dst can be specified several ways
  - May be subject to cluster environment

```yaml
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: test-network-policy
  namespace: default
spec:
podSelector:
  matchLabels:
    role: db
policyTypes:
- Ingress
- Egress
ingress:
- from:
  - ipBlock:
    cidr: 172.17.0.0/16
    except:
      - 172.17.1.0/24
  - namespaceSelector:
    matchLabels:
      project: myproject
    podSelector:
      matchLabels:
        role: frontend
  ports:
    - protocol: TCP
      port: 6379
egress:
- to:
  - ipBlock:
    cidr: 10.0.0.0/24
  ports:
    - protocol: TCP
      port: 5978
```
Traffic shaping

Bandwidth

- Effected by a set of layers
- Does not manage bandwidth cluster wide

Pod: bandwidth annotations

CNI: Bandwidth Plugin

`tc` (Traffic Control)

Linux: Network Namespace
Traffic shaping

**Bandwidth: CNI**

- Can be enabled as a plugin without specific limits

  ```json
  {
    "type": "bandwidth",
    "capabilities": {"bandwidth": true}
  }
  ```

- Can be chained to a specific network interface and limit interface bandwidth use

  ```json
  {
    "cniVersion": "0.3.1",
    "name": "mynet",
    "plugins": [
      {
        "type": "ptp",
        "ipMasq": true,
        "mtu": 512,
        "ipam": {
          "type": "host-local",
          "subnet": "10.0.0.0/24"
        },
        "dns": {
          "nameservers": ["10.1.0.1"]
        }
      },
      {
        "name": "slowdown",
        "type": "bandwidth",
        "ingressRate": 123,
        "ingressBurst": 456,
        "egressRate": 123,
        "egressBurst": 456
      }
    ]
  }
  ```
Traffic shaping

Bandwidth: Pod Spec

```json
{
    "kind": "Pod",
    "metadata": {
        "name": "iperf-slow",
        "annotations": {
            "kubernetes.io/ingress-bandwidth": "10M",
            "kubernetes.io/egress-bandwidth": "10M"
        }
    }
}
```
Distance from mission accomplished

- Large scale of remote edge nodes and devices
  - We are talking about hundreds of thousands, millions or more
  - Remote management for devops, sre, etc.
  - Multi-tenancy @cloud
  - Cloud/edge native
- Unreliable/Limited network
  - Network bandwidth and topology
  - Edge autonomy
- Heterogeneous hardware config.
- Limited node resources
  - Kubelet
  - Container runtime
- Diversified device connection & protocol
  - pub/sub
  - Device state
KubeEdge

- An opensource project contributed/started by Huawei: github.com/kubeedge
- KubeEdge allows customers (devops, SRE, etc.) to manage edge nodes, deploy/orchestrated/monitor apps, etc. the same way as in the cloud
- KubeEdge contains components running at cloud and edge. Currently the edge part is opensourced and cloud will be very soon
KubeEdge supports
- multi-tenancy, large scale of distributed edge nodes/apps.,
- lightweight agent, container native,
- duplex/multiplex cloud/edge network connection,
- device twin, mqtt/http device & edge connection
- Edge side autonomy
KubeEdge - Roadmap

Release 1.0

KubeEdge will provide the fundamental infrastructure and basic functionalities for IOT/Edge workload. This includes:

- K8s Application deployment through kubectl from Cloud to Edge node(s)
- K8s configmap, secret deployment through kubectl from Cloud to Edge node(s) and their applications in Pod
- Bi-directional and multiplex network communication between Cloud and edge nodes
- K8s Pod and Node status querying with kubectl at Cloud with data collected/reported from Edge
- Edge node autonomy when its getting offline and recover post reconnection to Cloud
- Device twin and MQTT protocol for IOT devices talking to Edge node

Release 2.0 and Future

- Build service mesh with KubeEdge and Istio
- Enable function as a service at Edge
- Support more types of device protocols to Edge node such as AMQP, BlueTooth, ZigBee, etc.
- Evaluate and enable super large scale of Edge clusters with thousands of Edge nodes and millions of devices
- Enable intelligent scheduling of apps. to large scale of Edge nodes
- etc.
Questions?
How to get involved in the Working Group, learn more...

Regular Work Group Meeting: Fridays at 8:00am Pacific (bi-weekly)
  • Meeting notes and agenda

Link to join the group
  • https://groups.google.com/forum/#!forum/kubernetes-wg-iot-edge

Link to join Slack
  • https://kubernetes.slack.com/messages/wg-iot-edge

White Paper
    • Workloads being considered
    • Technical challenges
    • Available architectural solutions