Container Camp UK 2018 - Workshop

Building a provider for the Virtual Kubelet

Photo by Hitesh Choudhary on Unsplash

Paul Bouwer
Software Engineer - Microsoft
@pbouwer

Stuart Leeks
Software Engineer - Microsoft
@stuartleeks

the virtual kubelet **masquerades** as a **kubelet** for the purposes of **connecting** kubernetes to other APIs.

virtual kubelet

kubelet

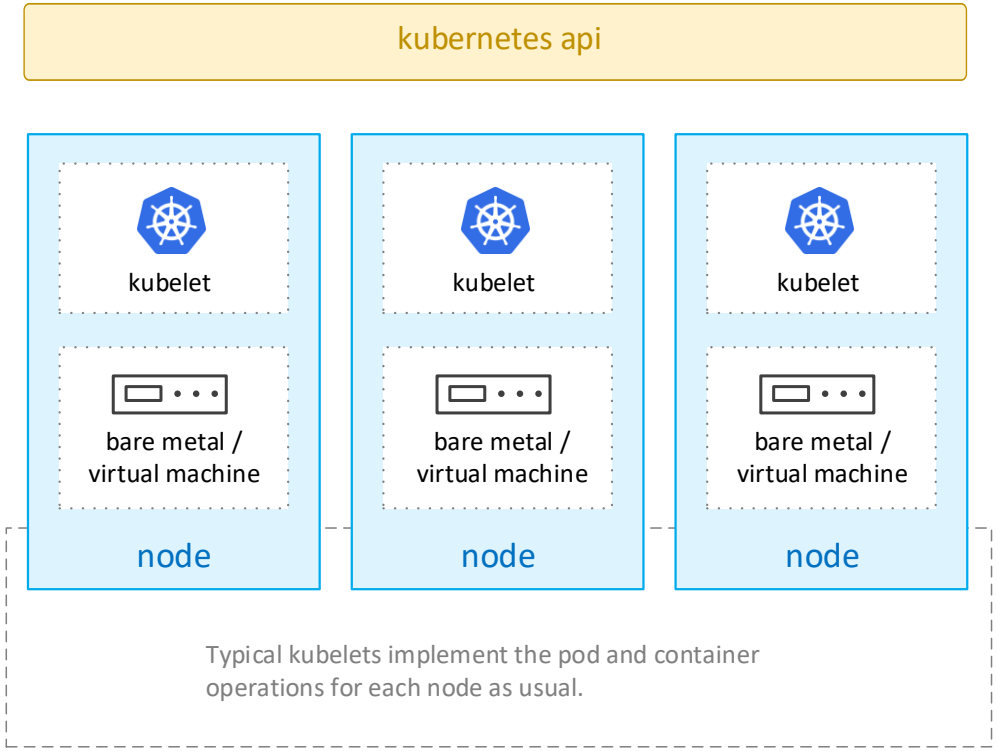
The kubelet is the primary **node agent** that runs on each node.

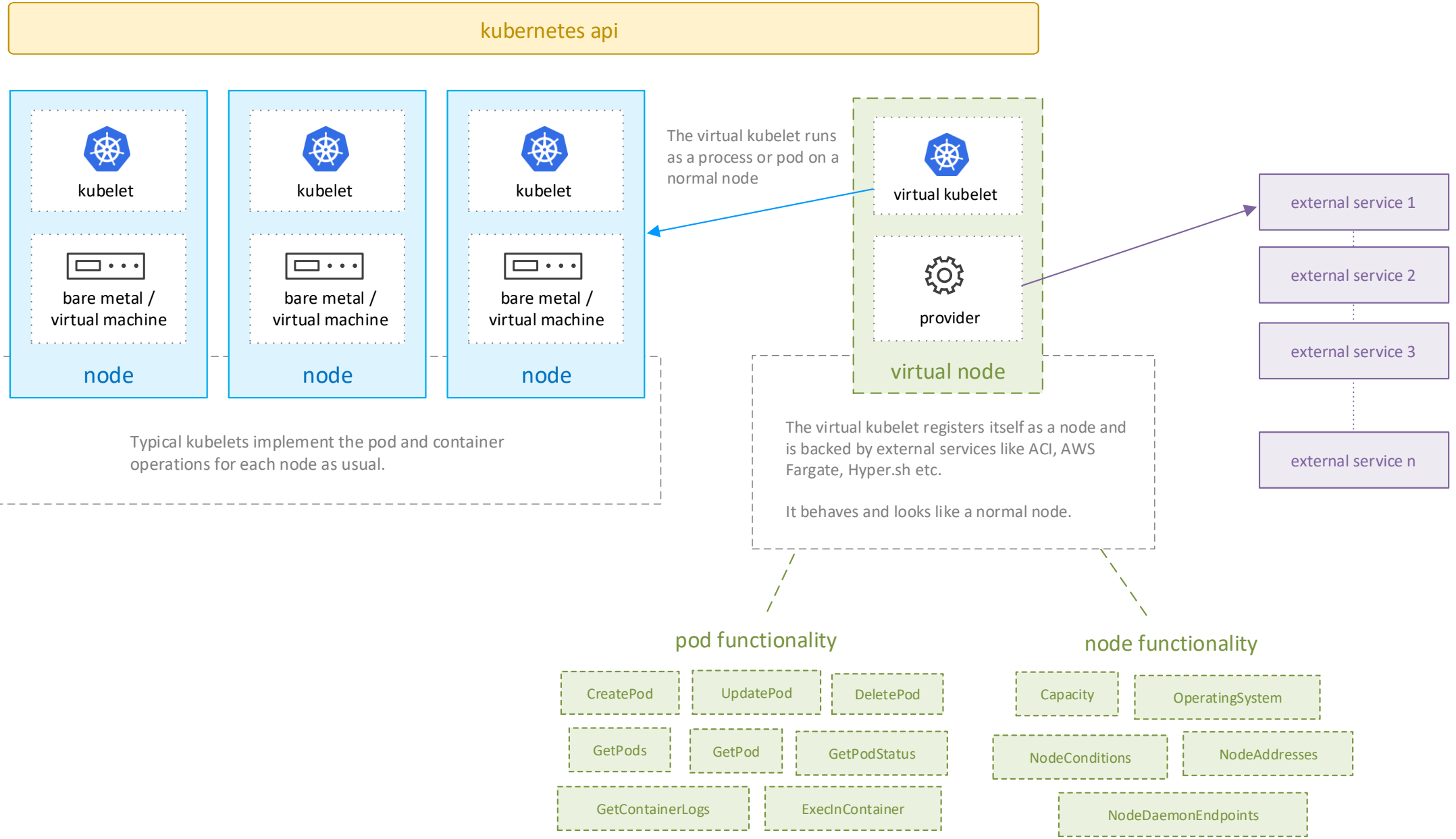
The kubelet works in terms of a **PodSpec**. A PodSpec is a YAML or JSON object that describes a pod. The kubelet takes a set of PodSpecs that are provided through various mechanisms (primarily through the apiserver) and ensures that the containers described in those PodSpecs are **running and healthy**.

The kubelet **doesn't manage** containers which were **not created** by Kubernetes.

https://kubernetes.io/docs/reference/command-line-tools-reference/kubelet/

virtual kubelet


kubernetes api
kubelet
bare metal /
virtual machine
node
kubelet
bare metal /
virtual machine
node
kubelet
bare metal /
virtual machine
node
Typical kubelets implement the pod and container
operations for each node as usual.


kubernetes api
kubelet
bare metal / virtual machine
node
kubelet
bare metal / virtual machine
node
kubelet
bare metal / virtual machine
node
Typical kubelets implement the pod and container operations for each node as usual.
The virtual kubelet runs as a process or pod on a normal node
virtual kubelet
provider
virtual node
The virtual kubelet registers itself as a node and is backed by external services like ACI, AWS Fargate, Hyper.sh etc.
It behaves and looks like a normal node.
external service 1
external service 2
external service 3
external service n
pod functionality
CreatePod
UpdatePod
DeletePod
GetPods
GetPod
GetPodStatus
GetContainerLogs
ExecInContainer
node functionality
Capacity
OperatingSystem
NodeConditions
NodeAddresses
NodeDaemonEndpoints

capabilities

Pod

create pod	deploy a kubernetes pod within the provider
update pod	update a kubernetes pod within the provider
delete pod	delete a kubernetes pod from the provider
get pod	retrieve a pod by name from the provider (can be cached)
get pod status	retrieve the status of a pod by name from the provider
get pods	retrieve a list of all pods running on the provider (can be cached)
get container logs	retrieve the logs of a container by name from the provider
exec in container	execute a command in a container in the pod, copying data between in/out/err and the container's stdin/stdout/stderr

Node

capacity	return a resource list with the capacity constraints of the provider
node conditions	return a list of conditions (Ready, OutOfDisk, etc), which is polled periodically to update the node status within kubernetes
node addresses	return a list of addresses for the node status within kubernetes
node daemon endpoints	return NodeDaemonEndpoints for the node status within kubernetes
operating system	return the operating system the provider is for

get it

virtual-kubelet/virtual-kubelet

Search or jump to... / Pull requests Issues Marketplace Explore

virtual-kubelet / virtual-kubelet

Watch 90 | Star 1,125 | Fork 168

Code | Issues 74 | Pull requests 11 | Projects 0 | Wiki | Insights

Branch: master ▾ | virtual-kubelet / README.md | Find file | Copy path

yaron2 spaces fix — ebdb925 26 days ago

16 contributors

238 lines (158 sloc) | 9.83 KB | Raw | Blame | History

Virtual Kubelet

Virtual Kubelet is an open source Kubernetes kubelet implementation that masquerades as a kubelet for the purposes of connecting Kubernetes to other APIs. This allows the nodes to be backed by other services like ACI, AWS Fargate, Hyper.sh, IoT Edge etc. The primary scenario for VK is enabling the extension of the Kubernetes API into serverless container platforms like ACI, Fargate, and Hyper.sh, though we are open to others. However, it should be noted that VK is explicitly not intended to be an alternative to Kubernetes federation.

Virtual Kubelet features a pluggable architecture and direct use of Kubernetes primitives, making it much easier to build on.

We invite the Kubernetes ecosystem to join us in empowering developers to build upon our base. Join our slack channel named, virtual-kubelet, within the Kubernetes slack group.

Please note this software is experimental and should not be used for anything resembling a production workload.

The best description is "Kubernetes API on top, programmable back."

Table of Contents

- How It Works
- Usage
- Providers
  - Azure Container Instances Provider
  - Azure Batch GPU Provider
  - AWS Fargate Provider
  - Hyper.sh Provider
  - Service Fabric Mesh Provider
  - Adding a New Provider via the Provider Interface
- Testing
  - Testing the Azure Provider Client

microsoft/virtual-kubelet

Search

Explore Help Sign up Sign in

PUBLIC REPOSITORY

microsoft/virtual-kubelet ☆

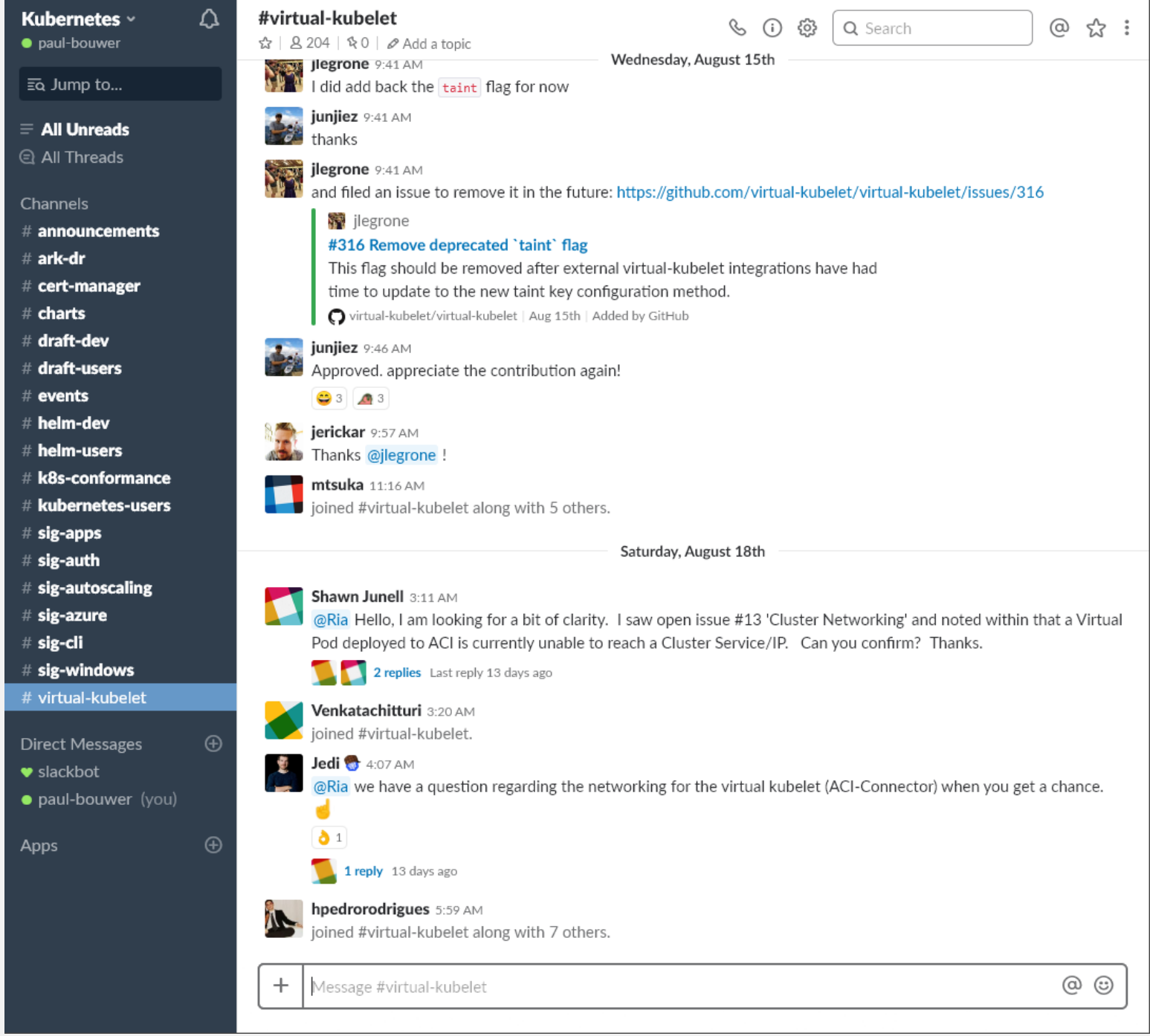
Last pushed: 5 hours ago

Repo Info Tags

Tag Name	Compressed Size	Last Updated
0.5.2	24 MB	5 hours ago
latest	18 MB	12 days ago
0.5.1	18 MB	12 days ago
0.5.0	18 MB	12 days ago
0.4.1	18 MB	a month ago
0.4	17 MB	2 months ago
0.3.3	17 MB	2 months ago
0.3.2	17 MB	3 months ago
0.3.1	17 MB	3 months ago
0.3	12 MB	3 months ago
0.2-rc-2	11 MB	4 months ago
0.2-rc-1	11 MB	4 months ago
0.2-beta-12	10 MB	4 months ago
0.2-beta-11	10 MB	4 months ago
0.2-beta-10	10 MB	5 months ago
0.2-beta-9	10 MB	6 months ago
0.2-beta-8	10 MB	6 months ago

get (grok) it

kubernetes.slack.com
virtual-kubelet



configure

USAGE

```
Usage:
  virtual-kubelet [flags]
  virtual-kubelet [command]

Available Commands:
  help        Help about any command
  version     Show the version of the program

Flags:
  -h, --help                      help for virtual-kubelet
      --kubeconfig string         config file (default "$HOME/.kube/config")
      --namespace string          kubernetes namespace (default "all")
      --nodename string           kubernetes node name (default "virtual-kubelet")
      --os string                 operating system (Linux/Windows) (default "Linux")
      --provider string           cloud provider
      --provider-config string    cloud provider configuration file
      --taint string              apply taint to node, making scheduling explicit

Use "virtual-kubelet [command] --help" for more information about a command.
```

NODE TAINTS

Defaults:

```
taint: virtual-kubelet.io/provider
value: <providername>
effect: NoSchedule
```

Customise:

```
taint: <taint flag value> # --taint flag on virtual kubelet binary
effect: NoSchedule
```

or

```
taint: $VKUBELET_TAINT_KEY
value: $VKUBELET_TAINT_VALUE
effect: $VKUBELET_TAINT_EFFECT # NoSchedule, NoExecute, PreferNoSchedule
```

NODE LABELS

```
type: virtual-kubelet
kubernetes.io/role: agent
beta.kubernetes.io/os: <configured node os>
kubernetes.io/hostname: <configured node name>
alpha.service-controller.kubernetes.io/exclude-balancer: true
```

providers

Platform Providers

aws	AWS Fargate
azure	Azure Container Instances (ACI)
azurebatch	Azure Batch
huawei	Huawei Cloud Container Instance (CCI)
hypersh	Hyper.sh Serverless Container Platform
sfmesh	Azure Service Fabric Mesh
vic	vSphere Integrated Containers
web	Http bridge

Platform Providers

aws	AWS Fargate
azure	Azure Container Instances (ACI)
azurebatch	Azure Batch
huawei	Huawei Cloud Container Instance (CCI)
hypersh	Hyper.sh Serverless Container Platform
sfmesh	Azure Service Fabric Mesh
vic	vSphere Integrated Containers
web	Http bridge

Testing and Prototyping Providers

cri	CRI-based container runtime
mock	Mock virtual kubelet provider for tests

implementation

GO INTERFACE

```go
// Provider contains the methods required to implement a
// virtual-kubelet provider.
type Provider interface {

 ...

}
```

GO INTERFACE / POD METHODS

```
// CreatePod takes a Kubernetes Pod and deploys it within the provider.
CreatePod(pod *v1.Pod) error

// UpdatePod takes a Kubernetes Pod and updates it within the provider.
UpdatePod(pod *v1.Pod) error

// DeletePod takes a Kubernetes Pod and deletes it from the provider.
DeletePod(pod *v1.Pod) error

// GetPod retrieves a pod by name from the provider (can be cached).
GetPod(namespace, name string) (*v1.Pod, error)

// GetPodStatus retrieves the status of a pod by name from the provider.
GetPodStatus(namespace, name string) (*v1.PodStatus, error)

// GetPods retrieves a list of all pods running on the provider (can be cached).
GetPods() ([]*v1.Pod, error)
```

GO INTERFACE / POD METHODS

```
// GetContainerLogs retrieves the logs of a container by name from the provider.
GetContainerLogs(namespace, podName, containerName string, tail int) (string, error)

// ExecInContainer executes a command in a container in the pod, copying data
// between in/out/err and the container's stdin/stdout/stderr.
ExecInContainer(name string, uid types.UID, container string, cmd []string, in
        io.Reader, out, err io.WriteCloser, tty bool, resize <-chan
        remotecommand.TerminalSize, timeout time.Duration) error
```

GO INTERFACE / NODE METHODS

```
// Capacity returns a resource list with the capacity constraints of the provider.
Capacity() v1.ResourceList

// NodeConditions returns a list of conditions (Ready, OutOfDisk, etc), which is
// polled periodically to update the node status within Kubernetes.
NodeConditions() []v1.NodeCondition

// NodeAddresses returns a list of addresses for the node status
// within Kubernetes.
NodeAddresses() []v1.NodeAddress

// NodeDaemonEndpoints returns NodeDaemonEndpoints for the node status
// within Kubernetes.
NodeDaemonEndpoints() *v1.NodeDaemonEndpoints

// OperatingSystem returns the operating system the provider is for.
OperatingSystem() string
```

WEB PROVIDER / POD METHODS

```
// Create a new pod.
POST /createPod

// Update an existing pod.
PUT /updatePod

// Delete an existing pod.
DELETE /deletePod

// Given a pod namespace and name, return the pod.
GET /getPod ? namespace, name

// Given a pod namespace and name, return the pod's status.
GET /getPodStatus

// Return list of existing pods.
GET /getPods

// Given the namespace, pod name and container name, return "tail" log lines.
GET /getContainerLogs ? namespace, podName, containerName, tail
```

WEB PROVIDER / NODE METHODS

```
// Return node capacity values.
GET /capacity

// Return list of node conditions (Ready, OutOfDisk etc).
GET /nodeConditions

// Return a list of addresses for the node status.
GET /nodeAddresses
```

requirements

kubernetes cluster

RBAC

Enabled, with ClusterAdminRole

deployment yaml

TOLERATIONS

```
tolerations:
- key: virtual-kubelet.io/provider
  value: azure
  effect: NoSchedule
```

```
tolerations:
- key: virtual-kubelet.io/provider
  value: aws
  effect: NoSchedule
```

```
tolerations:
- key: azure.com/aci
  effect: NoSchedule
```

NODE SELECTOR

```
nodeSelector:
  kubernetes.io/hostname: virtual-kubelet-linux-aci


nodeName: virtual-kubelet-linux-web
```

RESOURCE LIMITS

```
containers:
  - name: hello-kubernetes
    image: paulbouwer/hello-kubernetes:1.5
    resources:
      requests:
        memory: 1G
        cpu: 1
```

installation

helm chart

HELM INSTALL

```
helm install "charts/virtual-kubelet" --name "mine" \
  --namespace "virtual-kubelet" \
  --set "provider=<provider>" \
  --set "nodeName=virtual-kubelet-mine"
```

RBAC

Service Account (virtual-kubelet)
Cluster Role Binding (cluster-admin)

Config

Secret (api-server certs, provider secrets)

Virtual Kubelet

Deployment (virtual kubelet)

binary

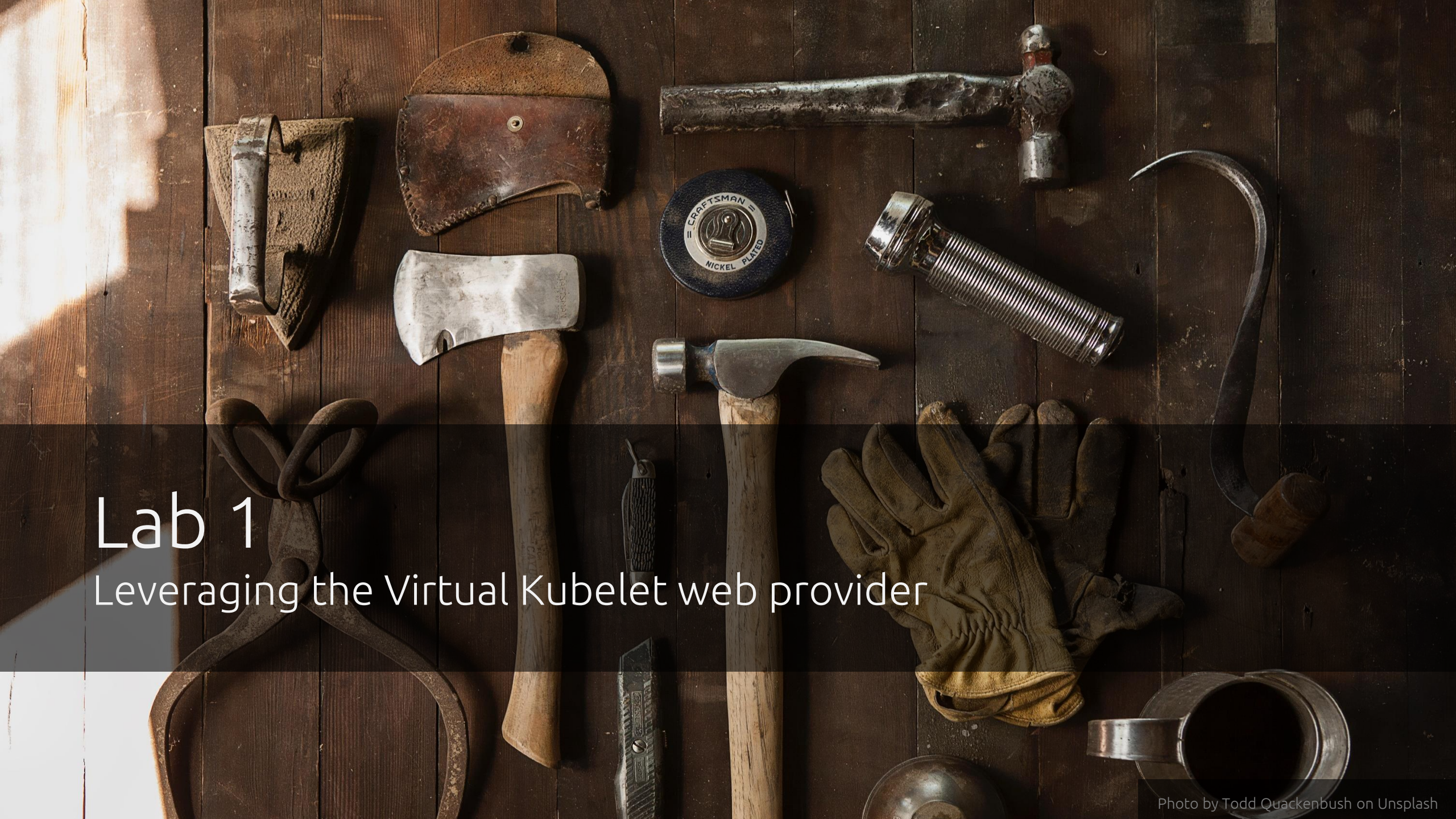
BINARY

```
virtual-kubelet --nodename "virtual-kubelet-mine" \
  --os "Linux" \
  --provider "<provider>"
```

Demo

The Virtual Kubelet web provider in action

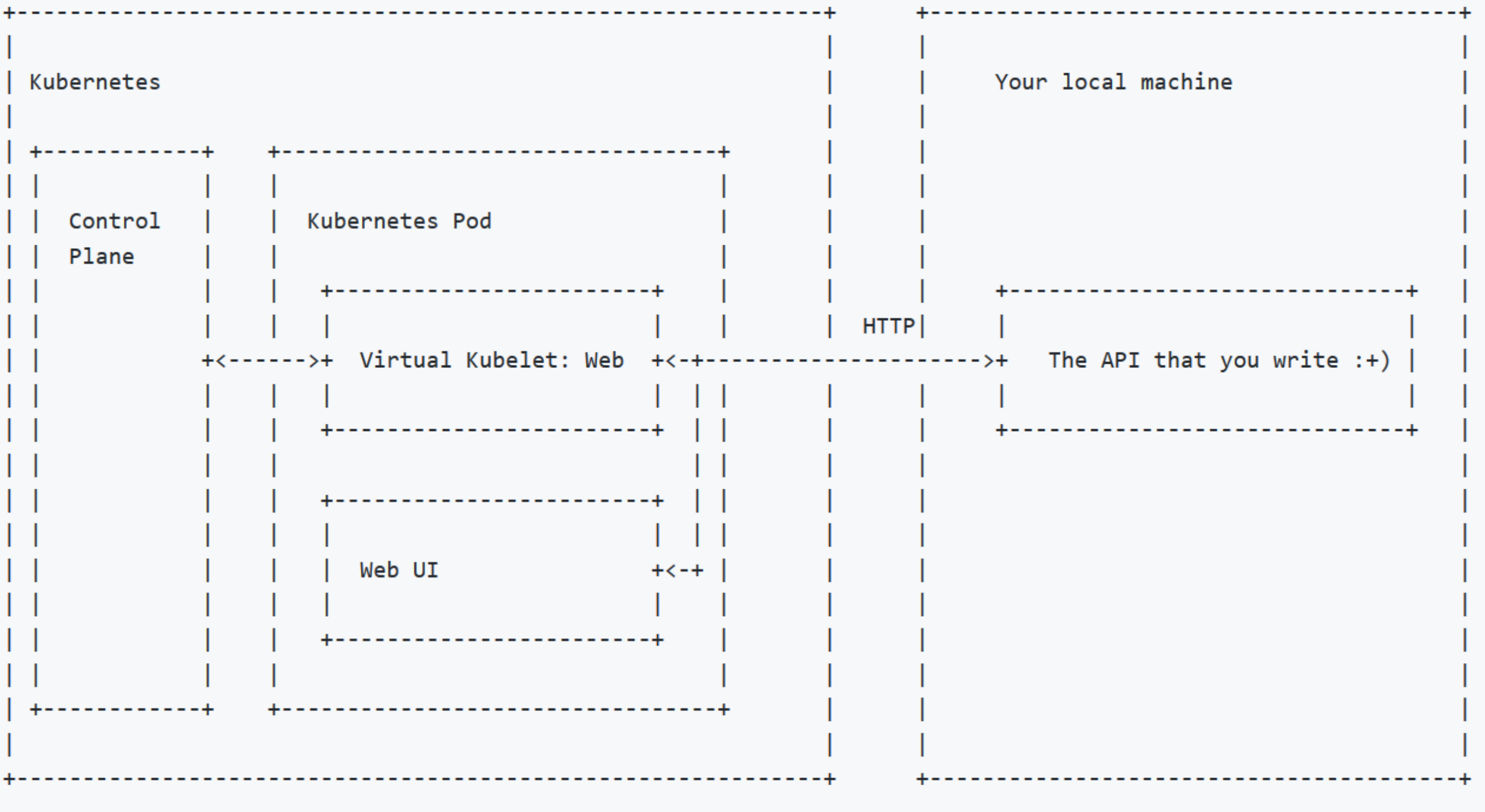
Photo by Samuel Zeller on Unsplash



Lab 1

Leveraging the Virtual Kubelet web provider

Photo by Todd Quackenbush on Unsplash


Kubernetes
Control Plane
Kubernetes Pod
Virtual Kubelet: Web
Web UI
HTTP
Your local machine
The API that you write :+)

GitHub

https://github.com/stuartleeks/virtual-kubelet-workshop-building-a-provider

wrap up