Red Hat
Enterprise Linux

Declarative Kubernetes Clusters with Cluster API

Joel Speed
Senior Software Engineer

Michael McCune
Principal Software Engineer

Joel Speed
Senior Software Engineer
@JoelASpeed

Michael McCune
Principal Software Engineer
@elmiko@mastodon.technology

Red Hat

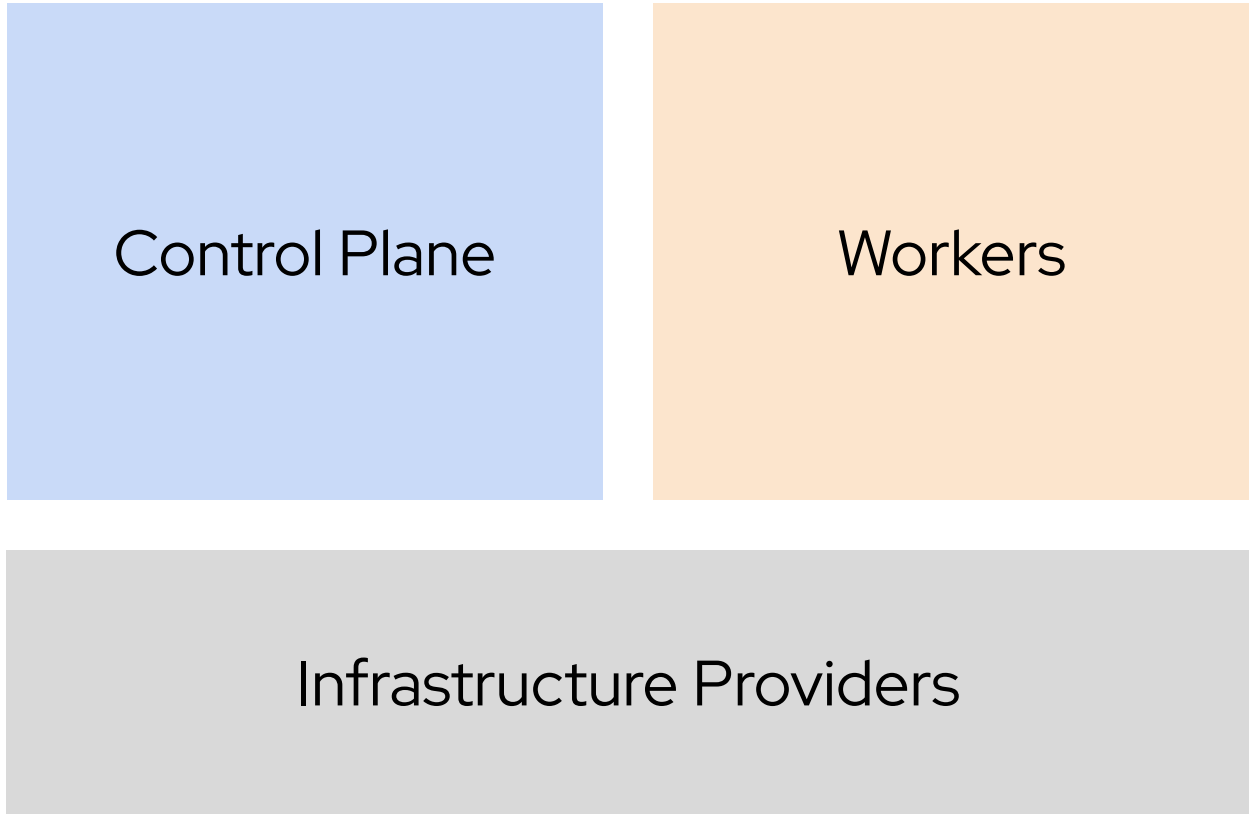
What we will discuss today

- Tech Review
- What is CAPI?
- Why is CAPI?
- CAPI Anatomy
- Demo
- Getting Involved

Red Hat

Tech review: Kubernetes

Red Hat



Red Hat

Control Plane

- kube-apiserver
- etcd
- kube-scheduler
- kube-controller-manager
- cloud-controller-manager

Red Hat

Workers

- kubelet
- kube-proxy
- container runtime

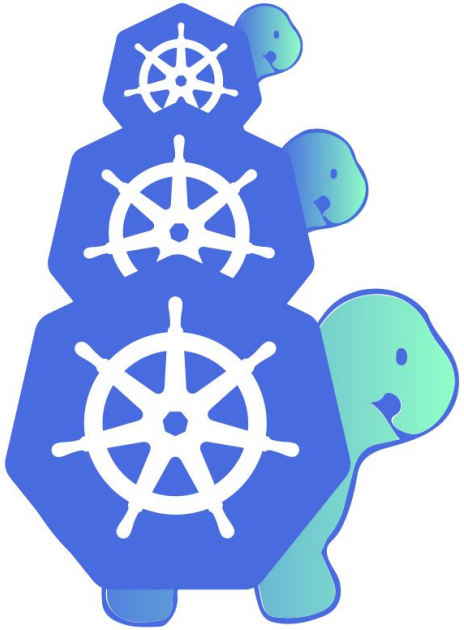
Red Hat

Infrastructure Providers

- bare metal
- virtual machines
- networking
- storage
- containers?
- in-memory?!?

Red Hat

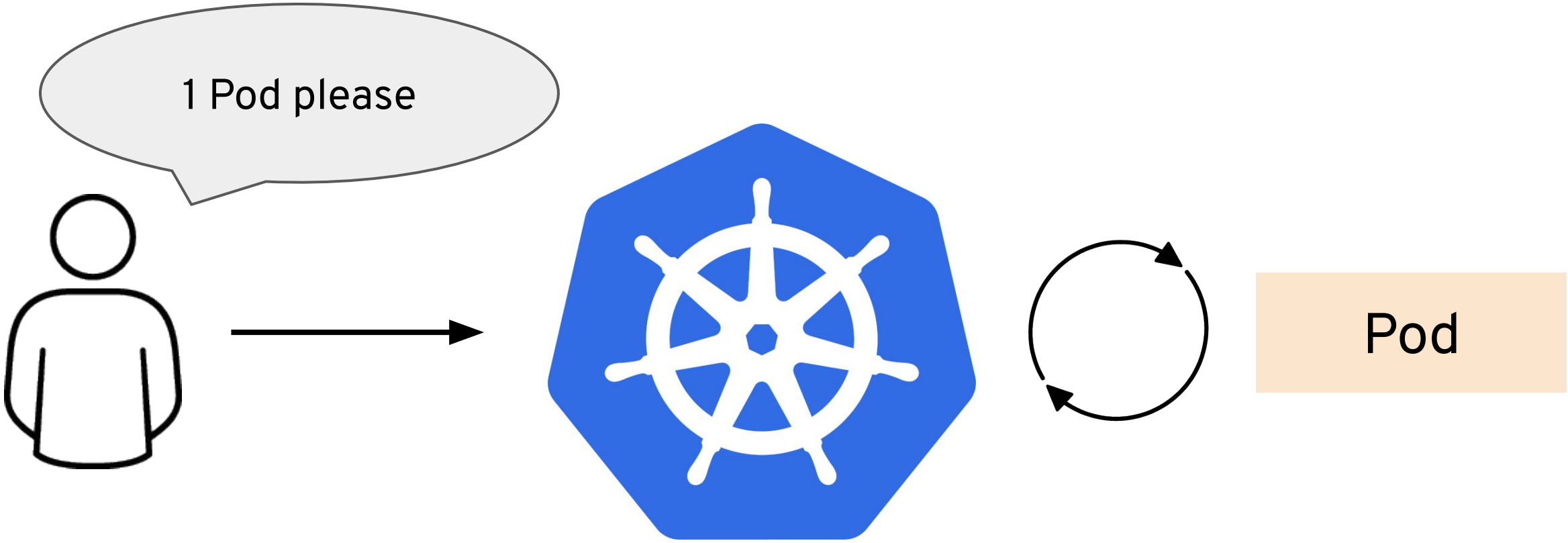
What is CAPI?



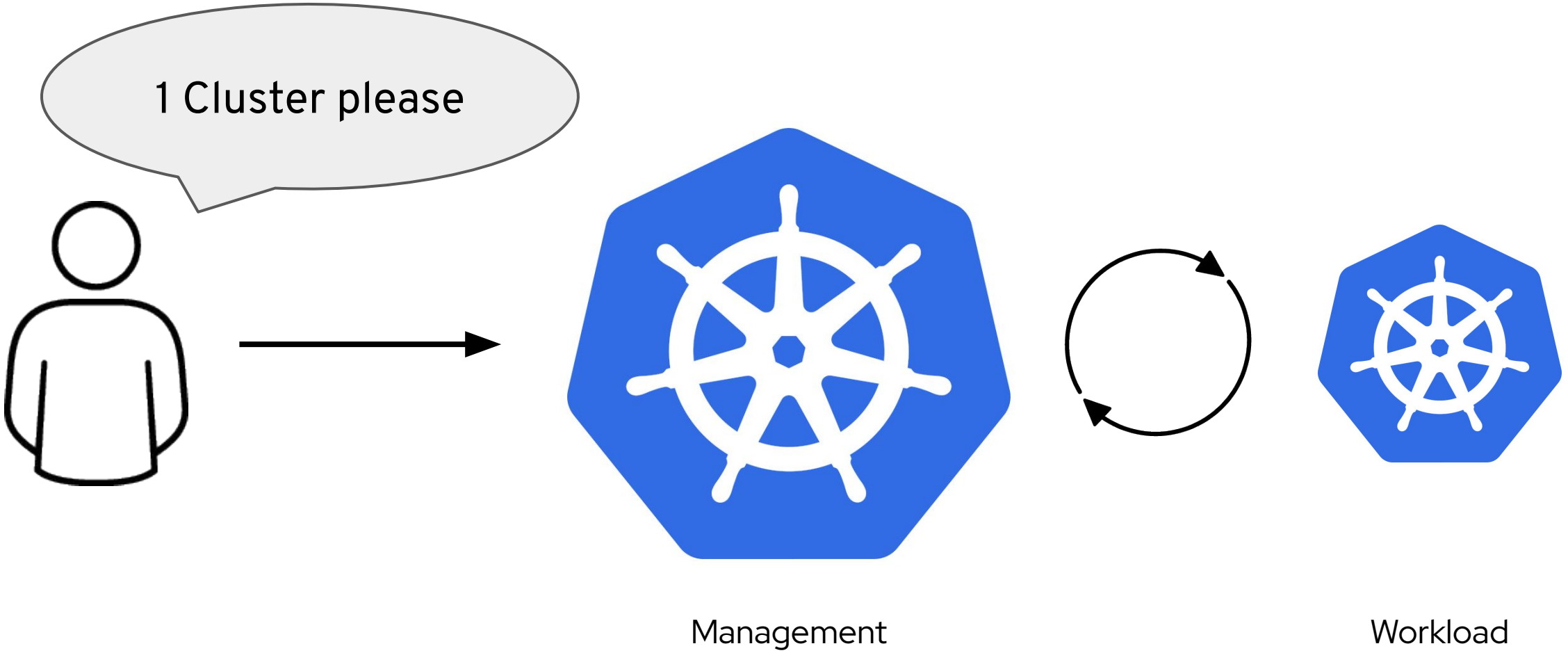
Red Hat

Cluster API is a Kubernetes sub-project focused on providing declarative APIs and tooling to simplify provisioning, upgrading, and operating multiple Kubernetes clusters.

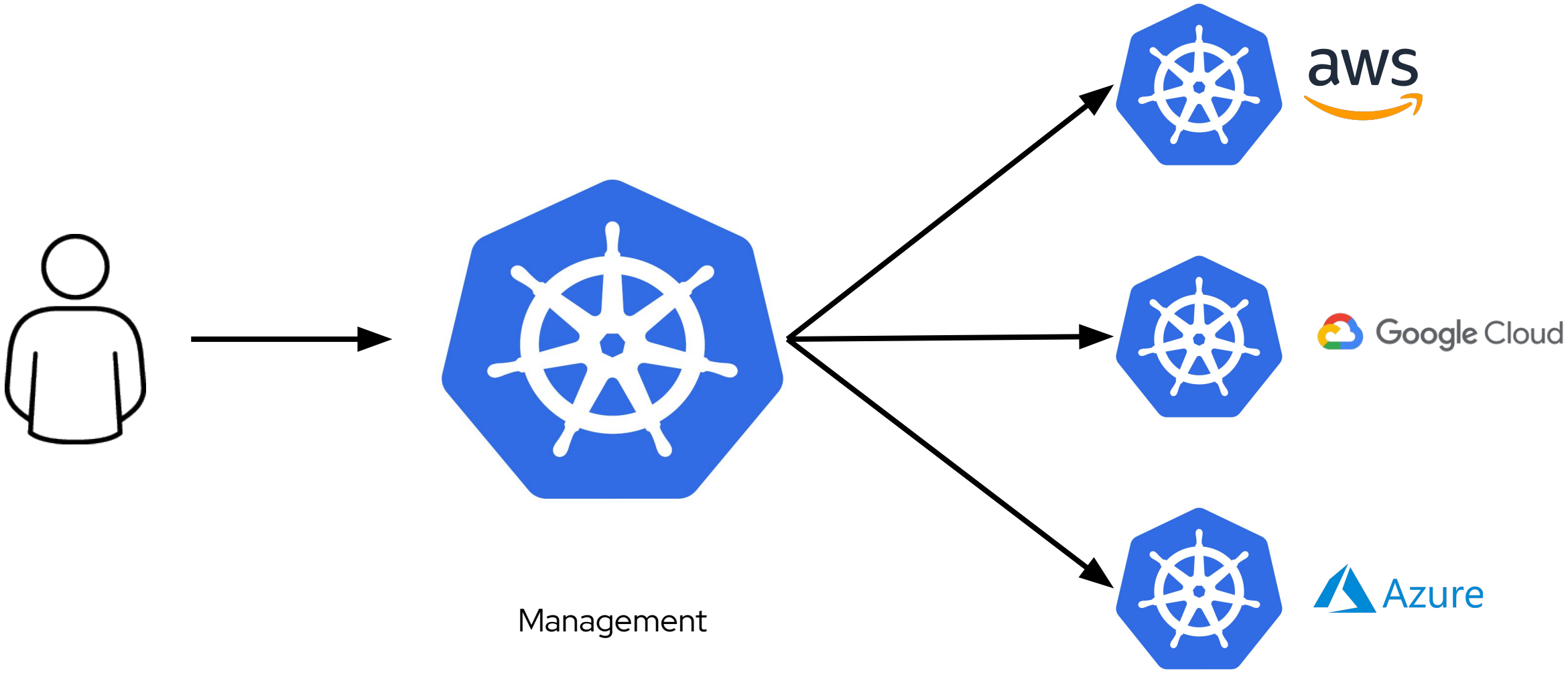
Red Hat



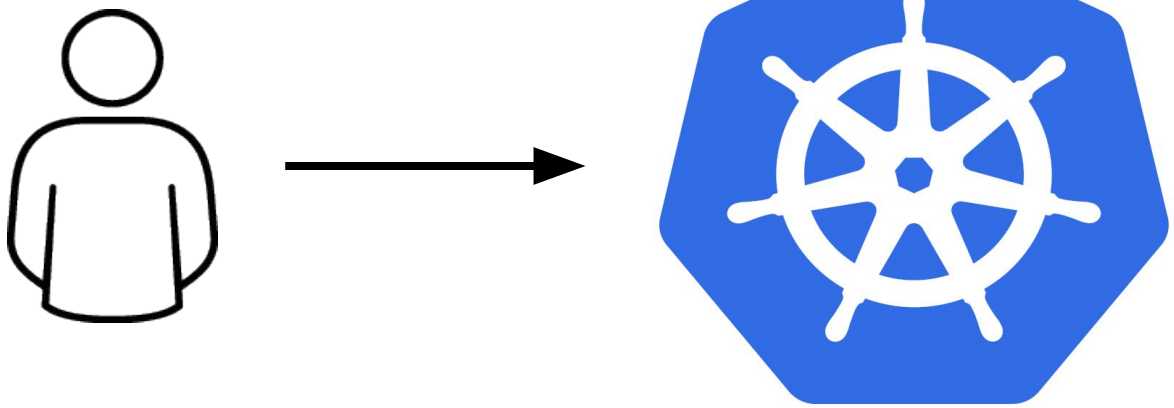
Red Hat



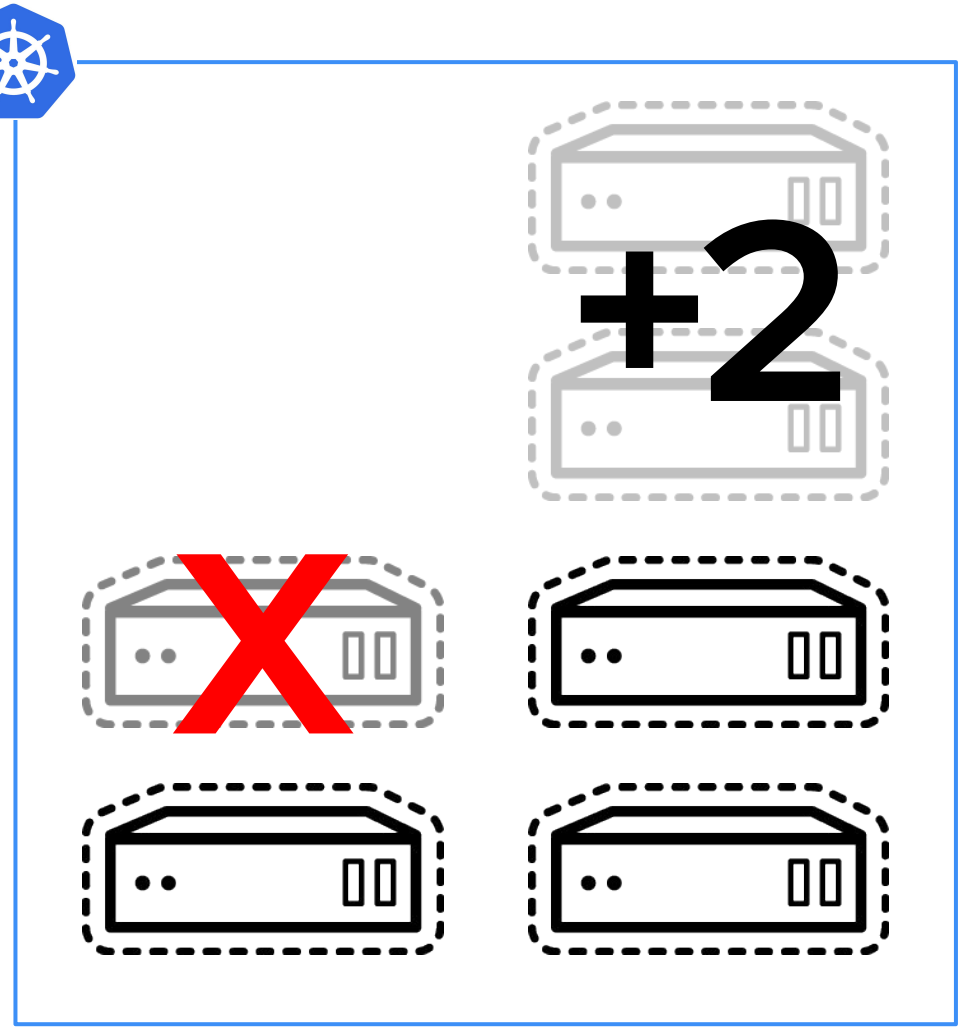
Red Hat



Red Hat



Management



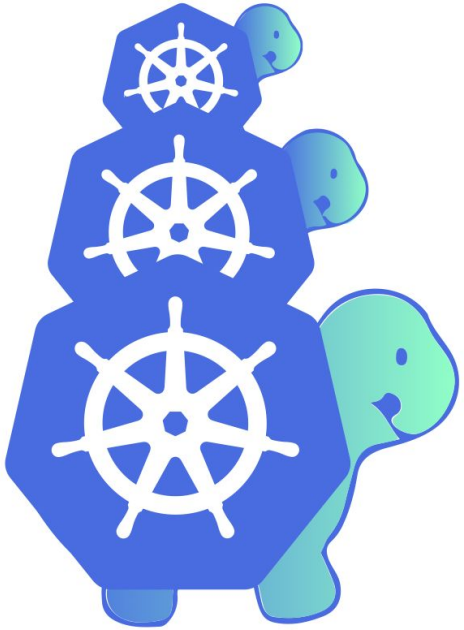
Red Hat

Beyond the basics

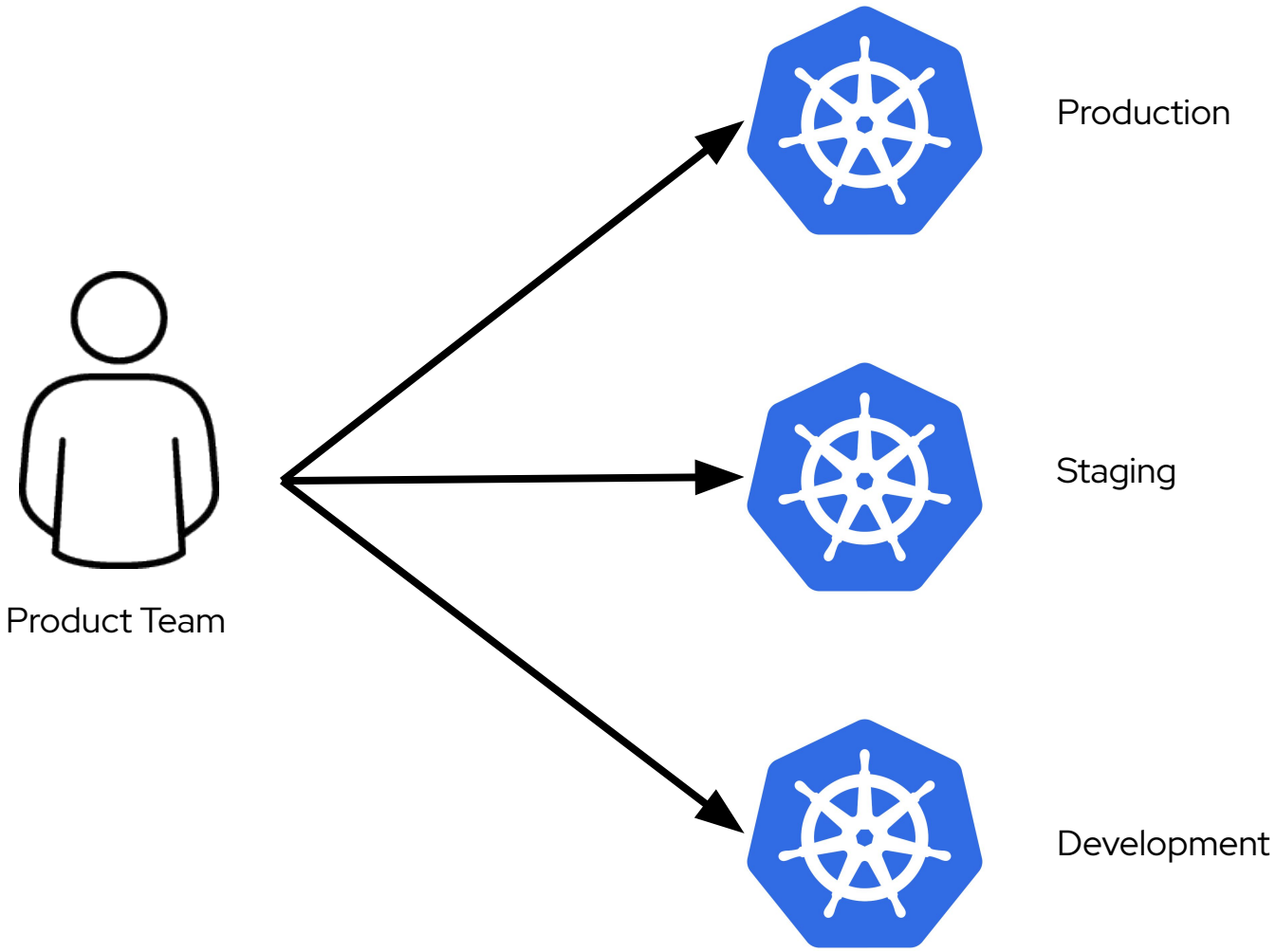
- Control plane management
- Machine health checking
- Cluster autoscaling
- Bootstrapping
- Spot instance usage
- 8 cloud providers

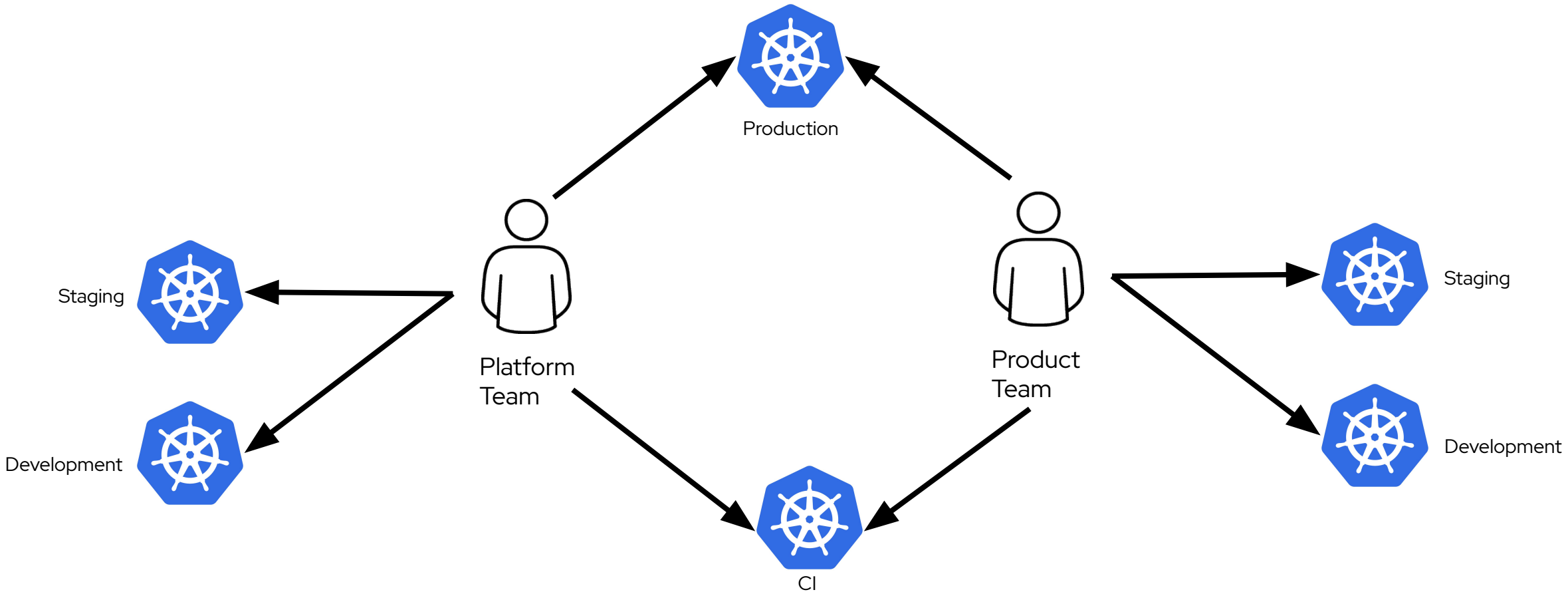
Red Hat

Why do we need CAPI?

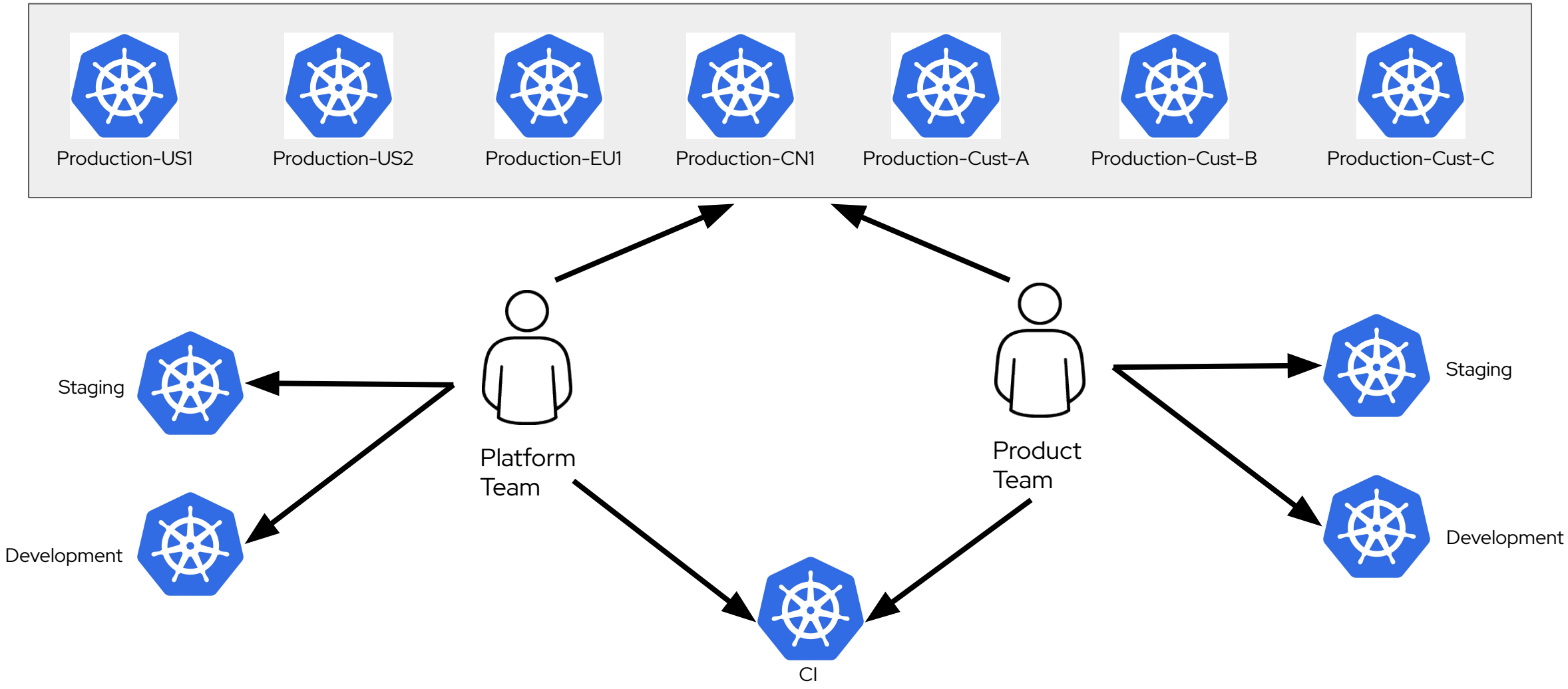


Red Hat


Production
Staging
Development
Product Team


Production
Staging
Development
Platform
Team
Product
Team
Staging
Development
CI

Why do we need CAPI?



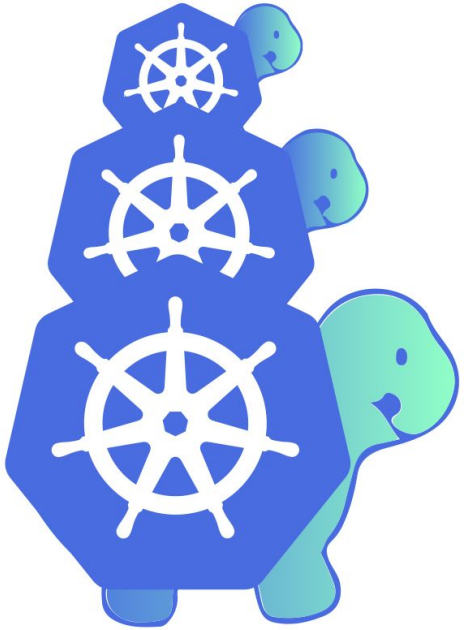
Red Hat

How does CAPI help?

- Centralises Management
- Automated Provisioning
- Automated Remediation
- Automated Upgrades (for workers)

Red Hat

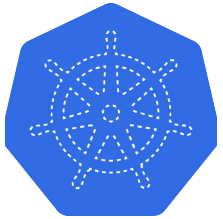
CAPI Anatomy



Red Hat

Core CAPI resources

Optional subheading

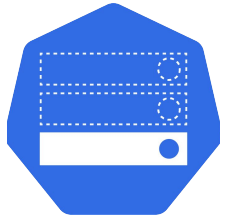


Cluster

Acts as a parent for other resources.

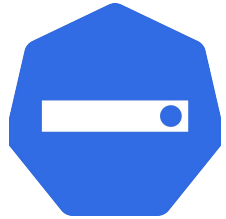
MachineDeployment

Creates MachineSets and updates them as needed.



MachineSet

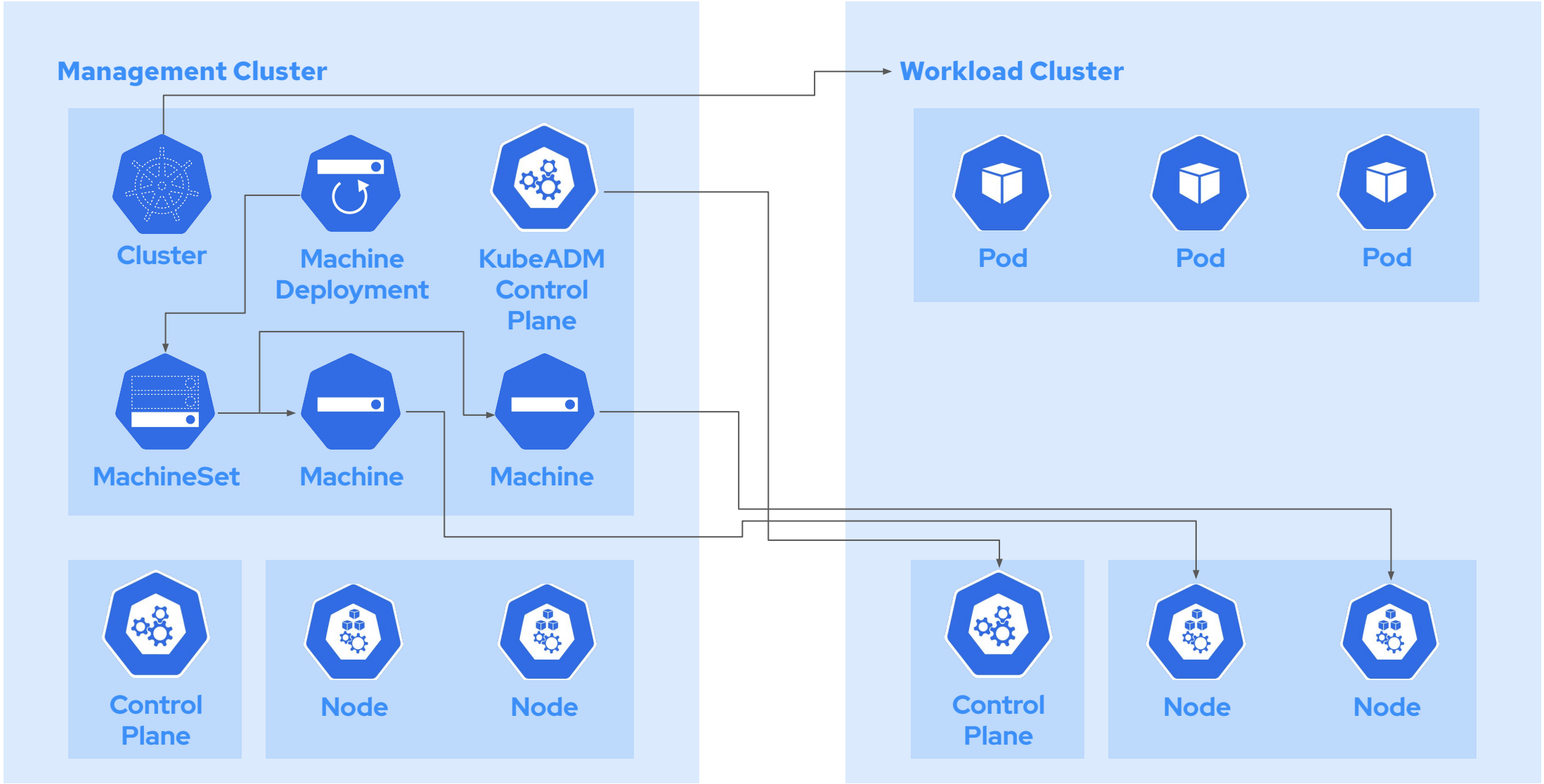
Ensures the desired number of Machines.



Machine

Creates a virtual Machine.

Red Hat


Management Cluster
Cluster
Machine Deployment
KubeADM Control Plane
MachineSet
Machine
Machine
Control Plane
Node
Node
Workload Cluster
Pod
Pod
Pod
Control Plane
Node
Node

Red Hat

CAPI Controllers

Cluster API Controllers

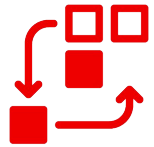
Set of controllers responsible for core Cluster API responsibilities.

Provider Controllers

Responsible for interacting with cloud providers to manage VM instances and supporting services.

Bootstrap Provider

Responsible for providing configuration for VMs to enable them to join the cluster.



Control Plane Controller

Responsible for creating and managing a control-plane, eg configuring etcd.

Red Hat

```yaml
apiVersion: cluster.x-k8s.io/v1alpha3
kind: Cluster
metadata:
  name: capi-demo-1
spec:
  clusterNetwork:
    pods:
      cidrBlocks:
      - 192.168.0.0/16
  controlPlaneRef:
    apiVersion: controlplane.cluster.x-k8s.io/v1alpha3
    kind: KubeadmControlPlane
    name: capi-demo-1-control-plane
  infrastructureRef:
    apiVersion: infrastructure.cluster.x-k8s.io/v1alpha3
    kind: AWSCluster
    name: capi-demo-1
```

Cluster

Red Hat

```yaml
apiVersion: infrastructure.cluster.x-k8s.io/v1alpha3
kind: AWSCluster
metadata:
  name: capi-demo-1
spec:
  region: us-east-1
  sshKeyName: capi-demo-ssh-key
```

InfrastructureCluster

Red Hat

```
apiVersion: controlplane.cluster.x-k8s.io/v1alpha3
kind: KubeadmControlPlane
metadata:
  name: capi-demo-1-control-plane
spec:
  infrastructureTemplate:
    apiVersion: infrastructure.cluster.x-k8s.io/v1alpha3
    kind: AWSMachineTemplate
    name: capi-demo-1-control-plane
  kubeadmConfigSpec:
    clusterConfiguration:
      ...
    initConfiguration:
      ...
    joinConfiguration:
      ...
  replicas: 3
  version: 1.18.2
```

ControlPlane

Red Hat

```yaml
apiVersion: infrastructure.cluster.x-k8s.io/v1alpha3
kind: AWSMachineTemplate
metadata:
  name: capi-demo-1-control-plane
spec:
  template:
    spec:
      iamInstanceProfile: control-plane.cluster-api-provider-aws.sigs.k8s.io
      instanceType: t3.large
      sshKeyName: capi-demo-ssh-key
```

MachineTemplate

Red Hat

```yaml
apiVersion: cluster.x-k8s.io/v1alpha3
kind: MachineDeployment
metadata:
  name: capi-demo-1-md-0
spec:
  clusterName: capi-demo-1
  replicas: 0
  template:
    spec:
      bootstrap:
        configRef:
          apiVersion: bootstrap.cluster.x-k8s.io/v1alpha3
          kind: KubeadmConfigTemplate
          name: capi-demo-1-md-0
      clusterName: capi-demo-1
      infrastructureRef:
        apiVersion: infrastructure.cluster.x-k8s.io/v1alpha3
        kind: AWSMachineTemplate
        name: capi-demo-1-md-0
      version: 1.18.2
```

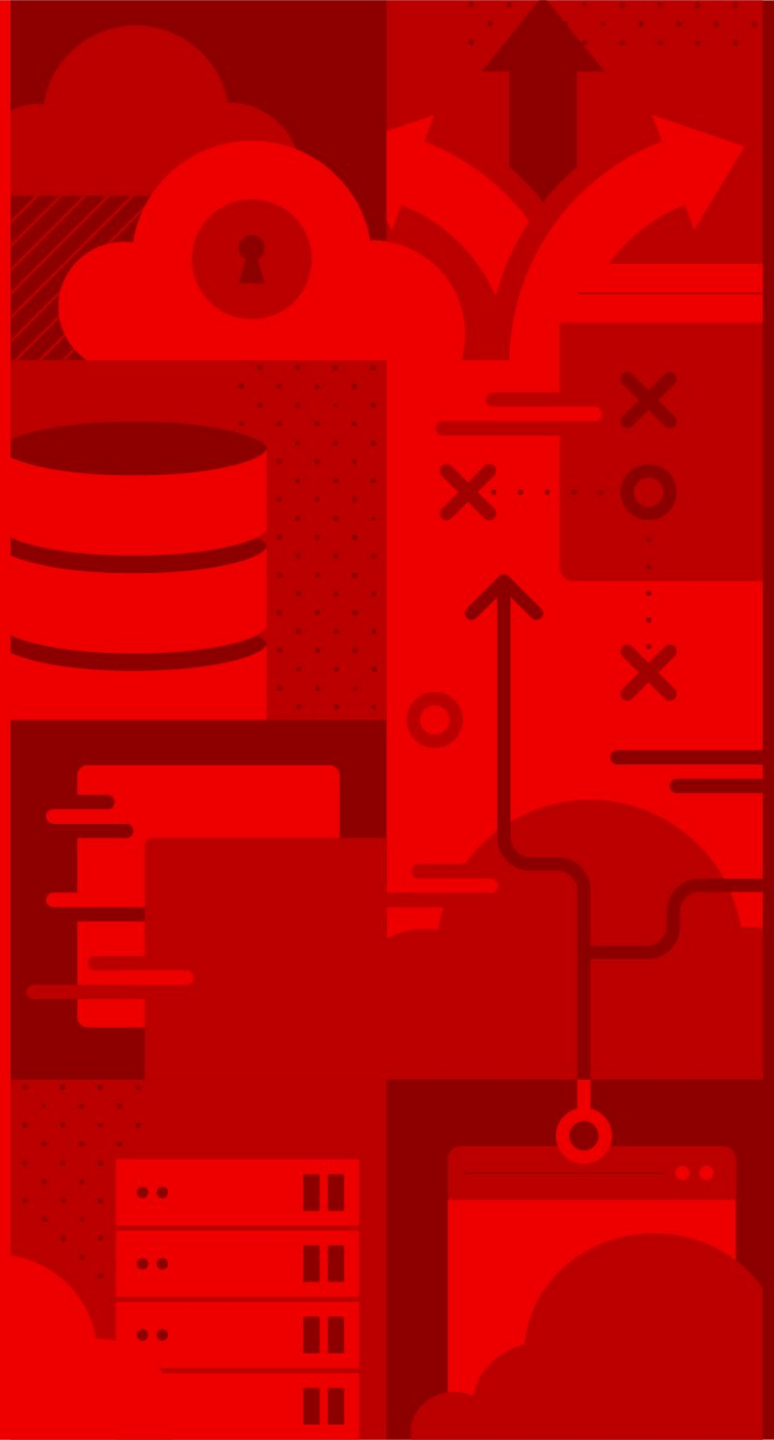
MachineDeployment

Red Hat

```
apiVersion: bootstrap.cluster.x-k8s.io/v1alpha3
kind: KubeadmConfigTemplate
metadata:
  name: capi-demo-1-md-0
spec:
  template:
    spec:
      joinConfiguration:
        nodeRegistration:
          kubeletExtraArgs:
            cloud-provider: aws
          name: '{{ ds.meta_data.local_hostname }}'
```

BootstrapConfigTemplate

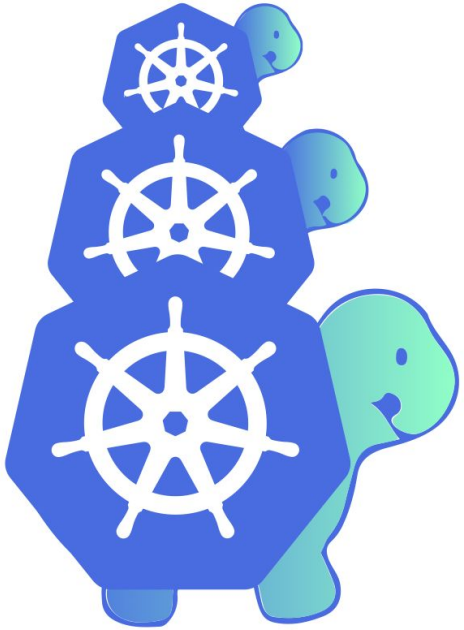
Red Hat



CAPI In Action (Demo)

Red Hat

How to get involved?



Red Hat

Read the book, try it out!

https://cluster-api.sigs.k8s.io/

see the Quick Start section to launch your own clusters, with or without a cloud provider.

Red Hat

Discuss your issues!

#cluster-api

kubernetes.slack.com

see the Cluster API book for links to the group mailing list

Red Hat

Attend a meeting!

Wednesdays @ 17:00 UTC

https://zoom.us/j/861487554

see the Cluster API book for recordings, notes, and updated information

Red Hat

Propose a change!

on github

kubernetes-sigs/cluster-api*

https://github.com/kubernetes-sigs/cluster-api

Red Hat

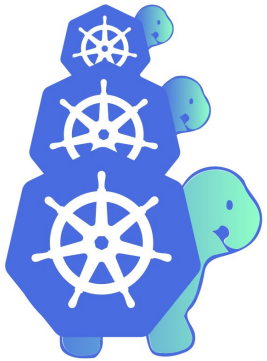
CAPI Roadmap

- Beta API version
- Increased testing
- CLI tooling improvements (clusterctl)
- Kubernetes API refactoring
- Bootstrap failure detection
- Pluggable load balancers

Red Hat

Thank You

Cluster API Book
https://cluster-api.sigs.k8s.io



Cluster API Project
https://github.com/kubernetes-sigs/cluster-api

Stay in touch

Joel Speed
@JoelASpeed

Michael McCune
@elmiko@mastodon.technology

Red Hat