Welcome to:

Automation and Orchestration

Unit objectives

After completing this unit, you should be able to:

- Learn the concept of automation and orchestration
- Understand key concepts in orchestration
- Gain knowledge on bridging realities, orchestration and programmable infrastructure
- Understand the concept of open source and standards
- Learn about peer perspectives on container orchestration survey
- Gain knowledge on cisco-Cloud-native capabilities and a deeper user experience
- Gain an insight into consideration for containers in production

Automation and orchestration

Figure: Automation and orchestration

Source : https://techture.global/wp-content/uploads/2018/03/Process-Automation-2.png

Key concepts in orchestration

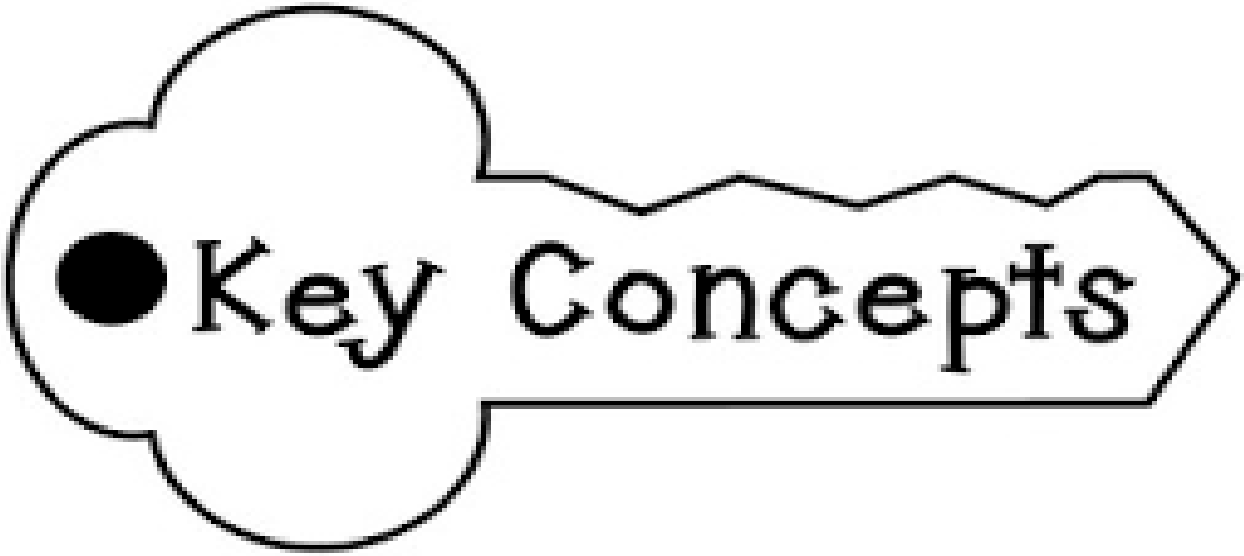

Figure: Key concepts in orchestration

Source: https://www.clipart.email/make-a-clipart/?image=549234

Popular orchestra platforms: Swarm Docker

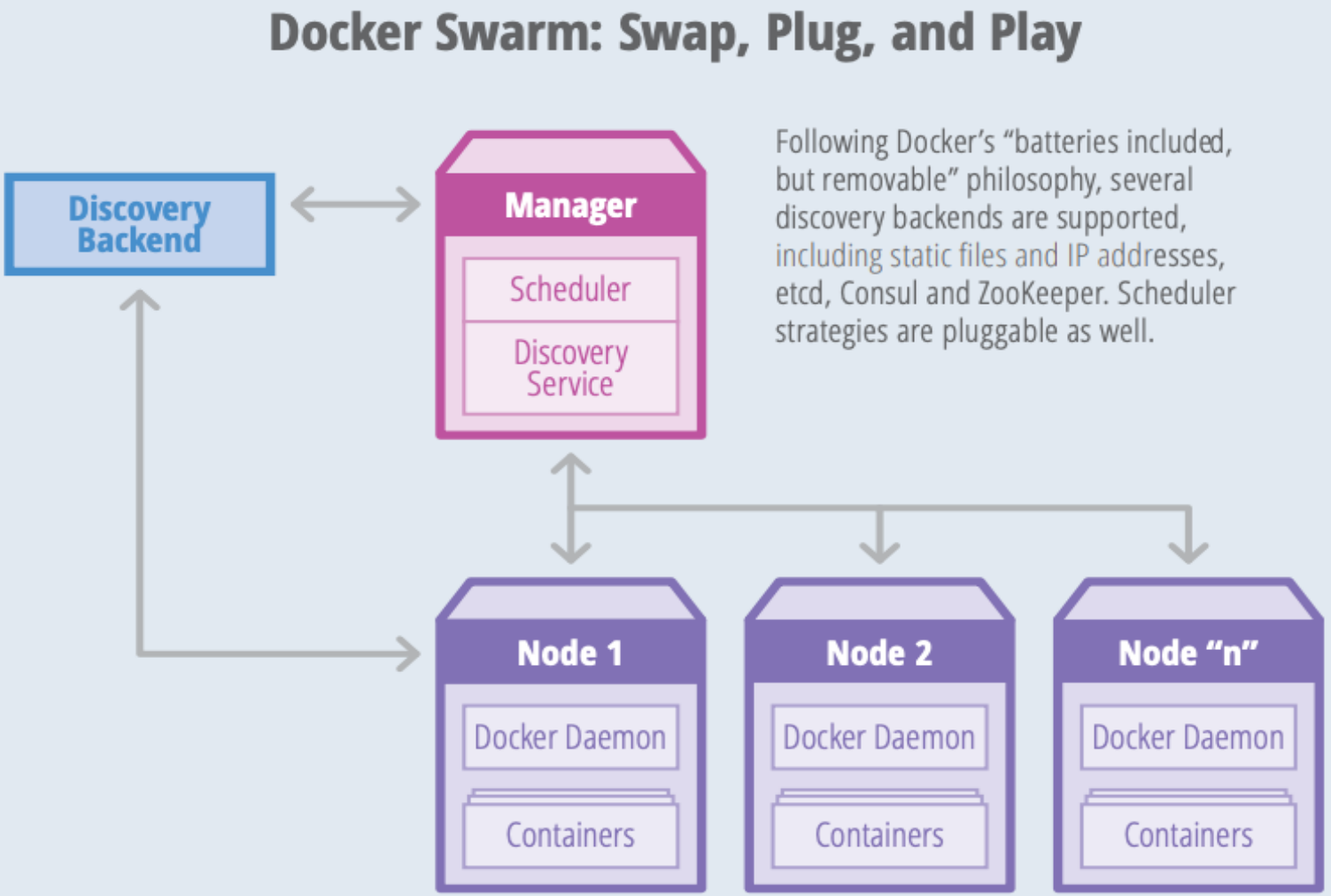

Figure: Popular orchestra platforms: swarm docker

Source: https://pbs.twimg.com/media/Cif2obHWwAA2G9g?format=jpgandname=large

Kubernetes

- The Kubernetes traces their architectural lineage to Google Borg, an inner cluster management scheme that lists more than two billion containers a day.

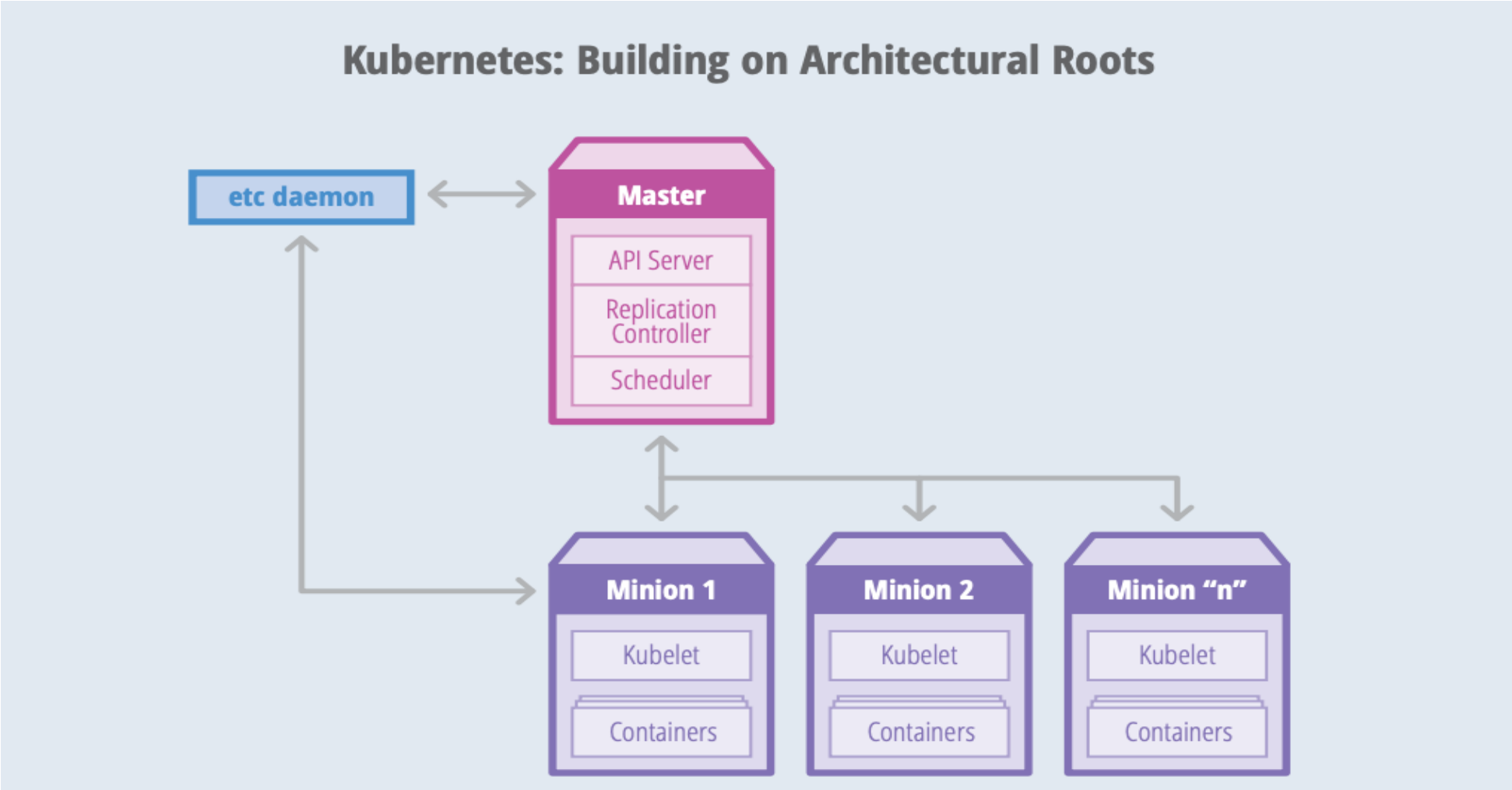

Figure: Kubernetes

Source: https://pbs.twimg.com/media/Cif2obHWwAA2G9g?format=jpgandname=large

Apache Mesos

- With roots in the superior registering world, Mesos supports Hadoop, Spark and more in addition Docker and containers.

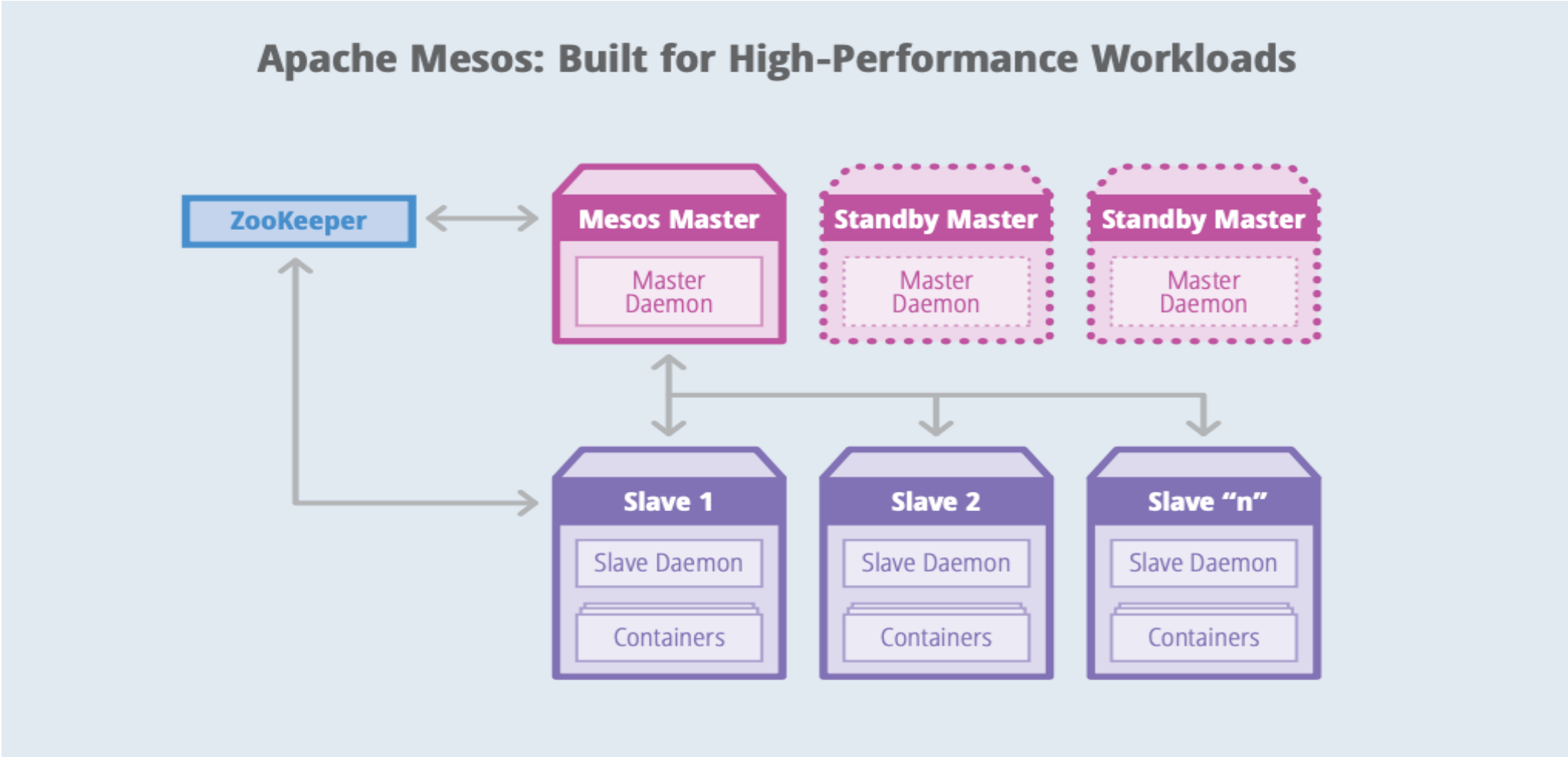

Figure: Apache mesos

Source: https://pbs.twimg.com/media/Cif2obHWwAA2G9g?format=jpgandname=large

Container orchestration survey

Figure: Container orchestration survey

Container adoption

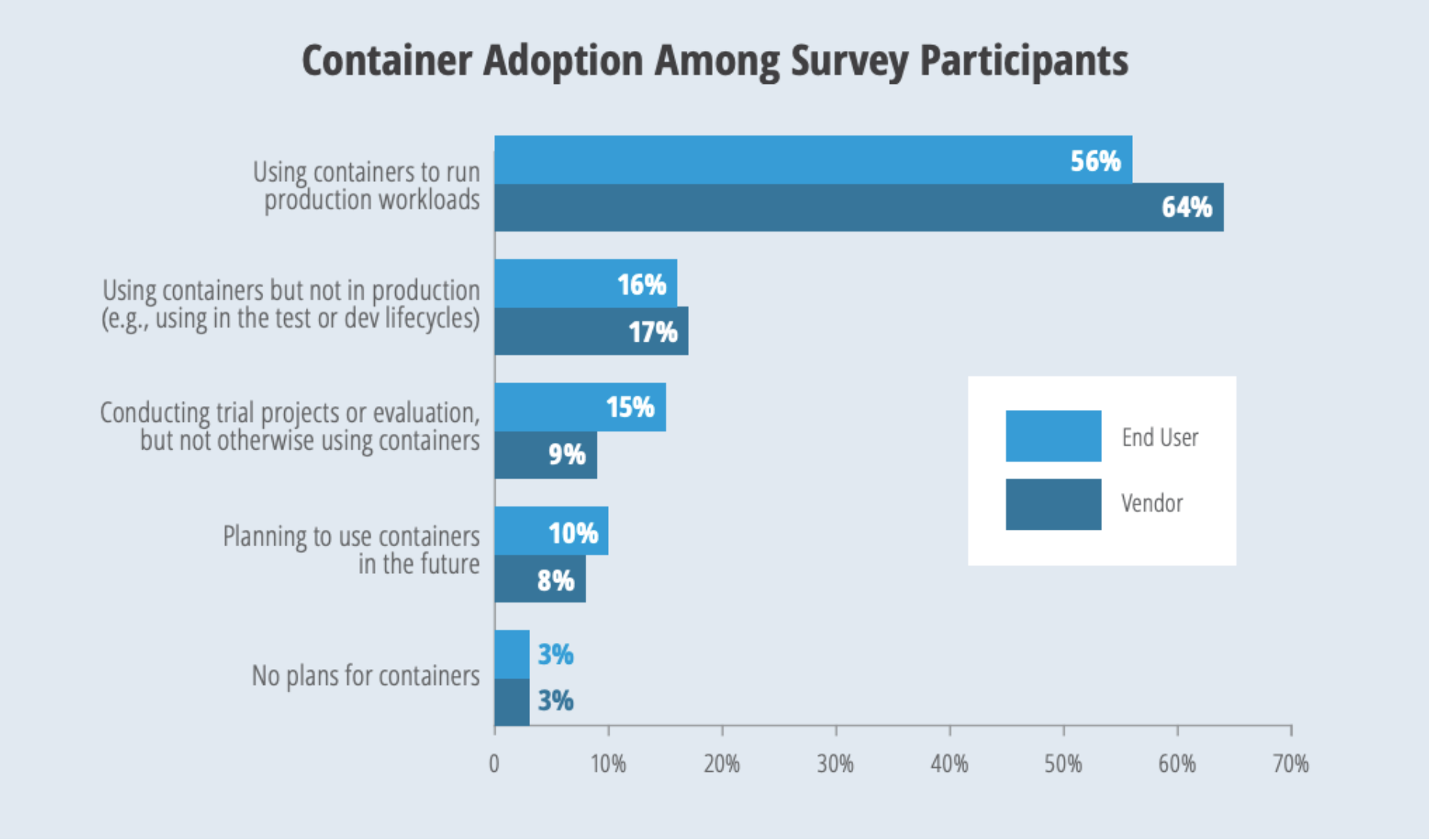

Figure: Some containers are used by 71% of the end-customers reviewed.

http://thenewstack.io/wp-content/uploads/2016/06/Chart_Container-Adoption-Among-Survey-Participants.png

Representation of DevOps pros

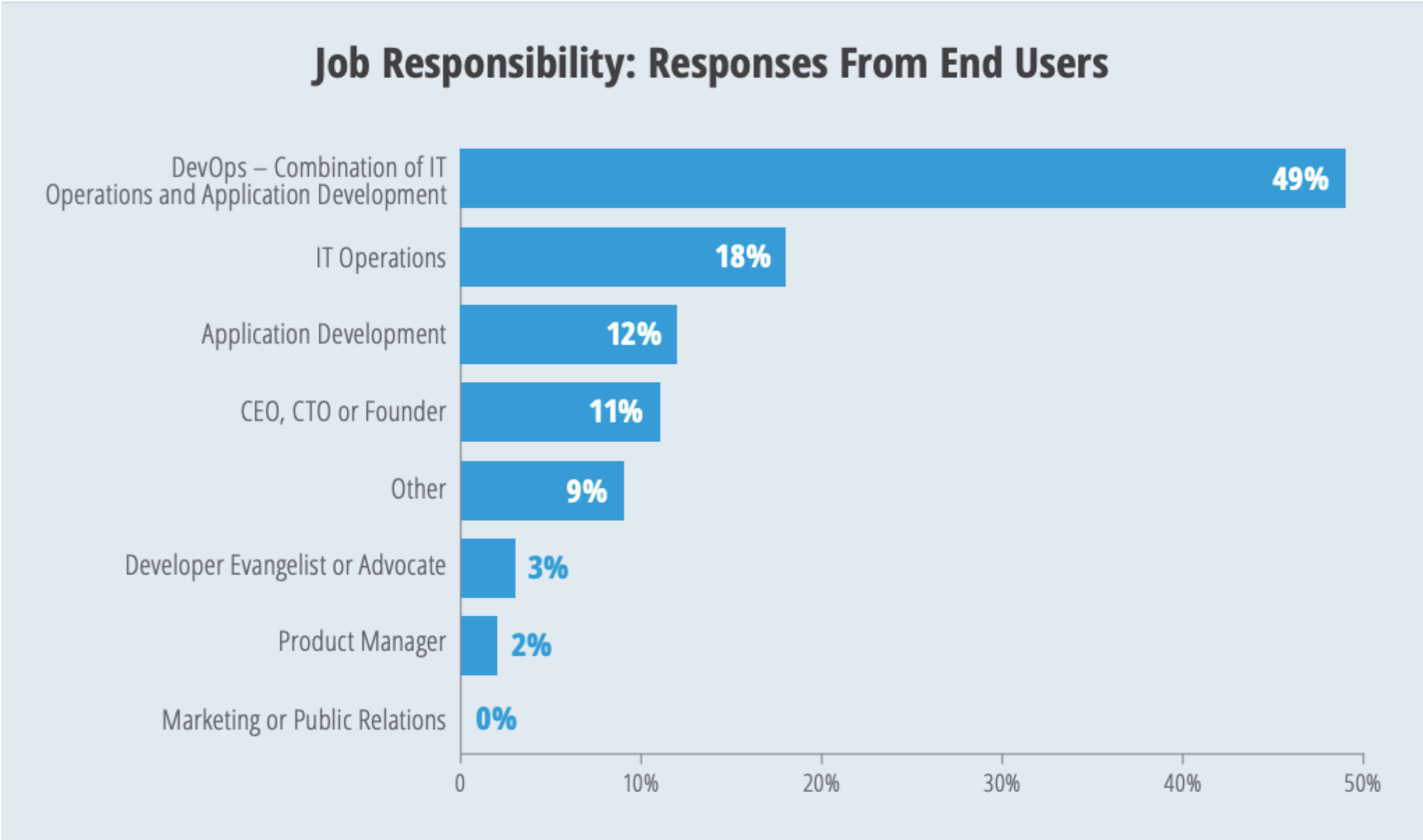

Figure: Representation of DevOps pros.

Source: http://thenewstack.io/wp-content/uploads/2016/06/Chart_Job-Responsibility-End-Users.png

Defining the functionality of container orchestration

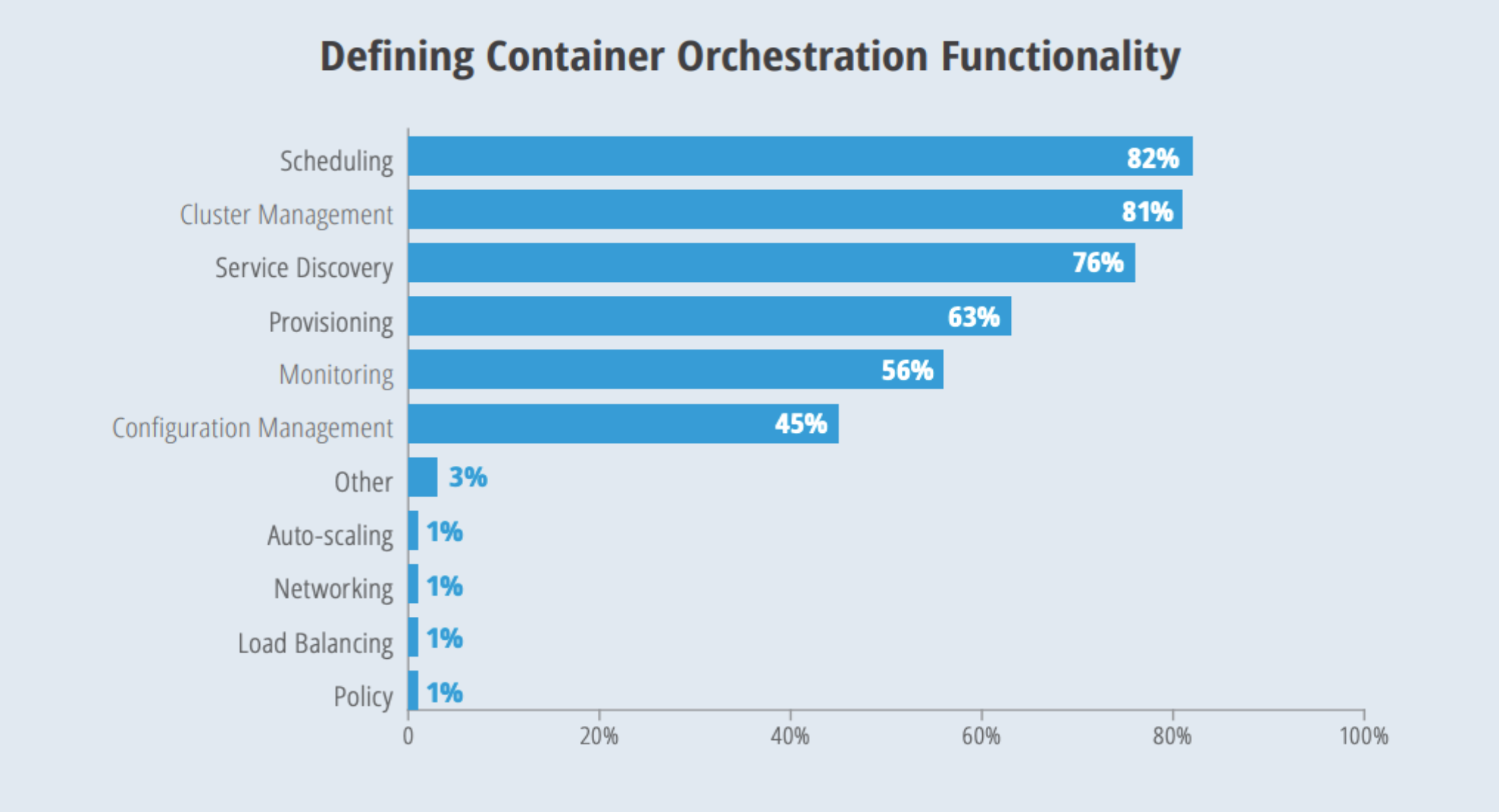

Figure: Product of container orchestration

Source: http://thenewstack.io/wp-content/uploads/2016/06/Chart_Defining-Container-Orchestration-Functionality.png

Response from end users

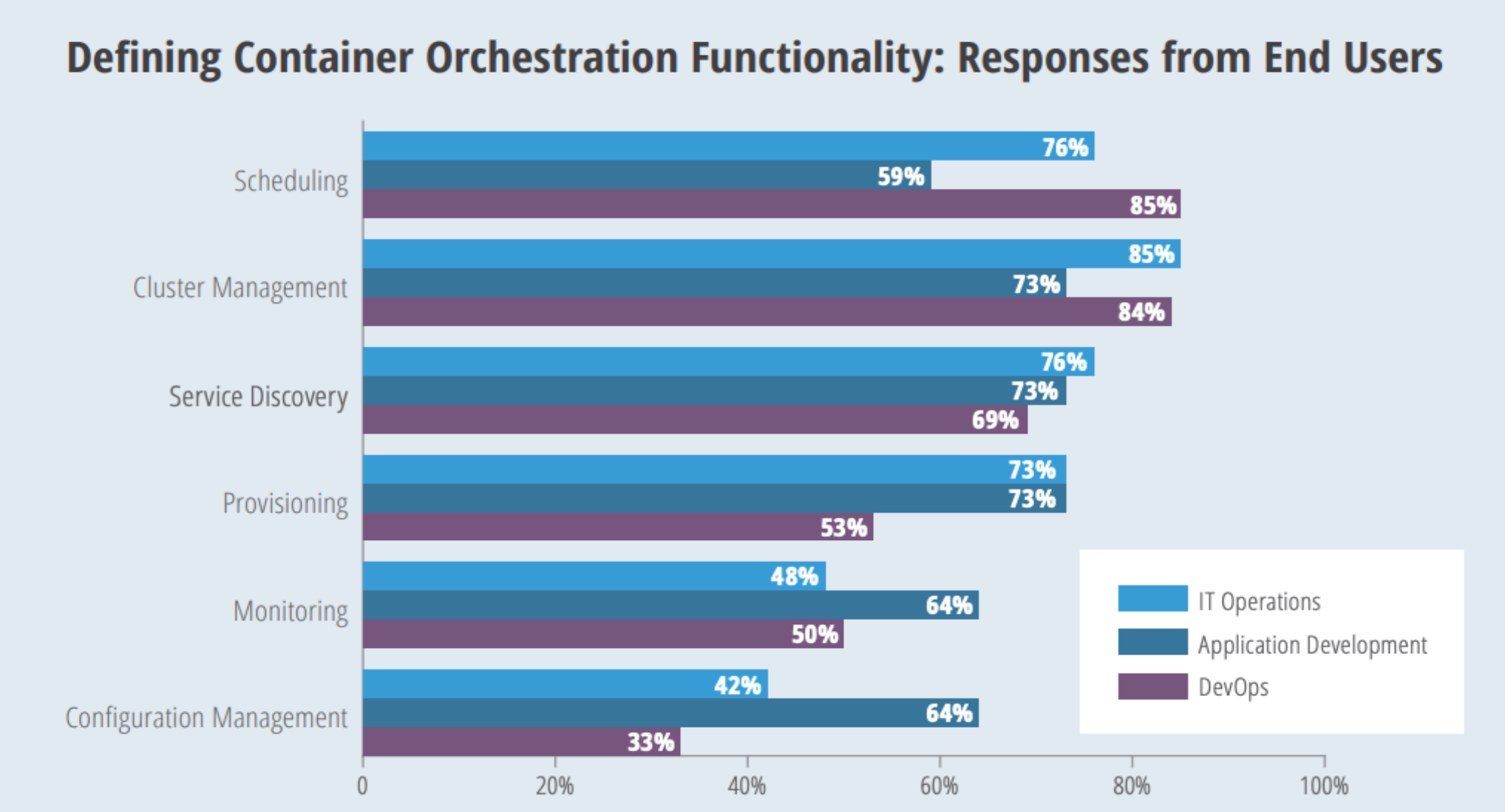

Figure: When app designers believe about container orchestration, scheduling is not top-of-mind.

Source: http://thenewstack.io/wp-content/uploads/2016/06/Chart_Defining-Container-Orchestration-Functionality-End-Users.png

Defining containers as a service function

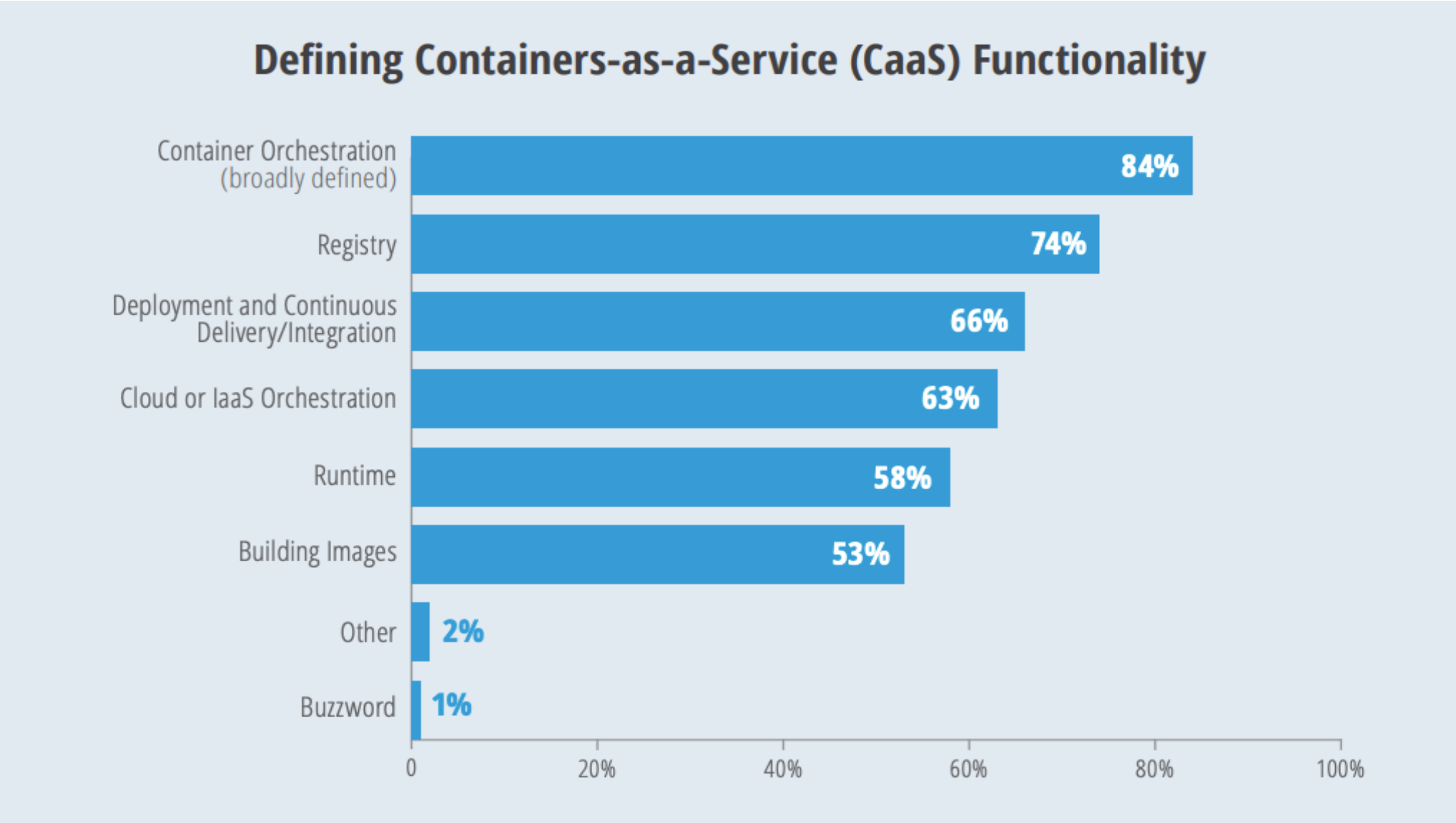

Figure: CaaS most related container orchestration and registers.

Source: http://thenewstack.io/wp-content/uploads/2016/06/Chart_Defining-CaaS-Functionality-1.png

Confusion abounds outside the vendor territory

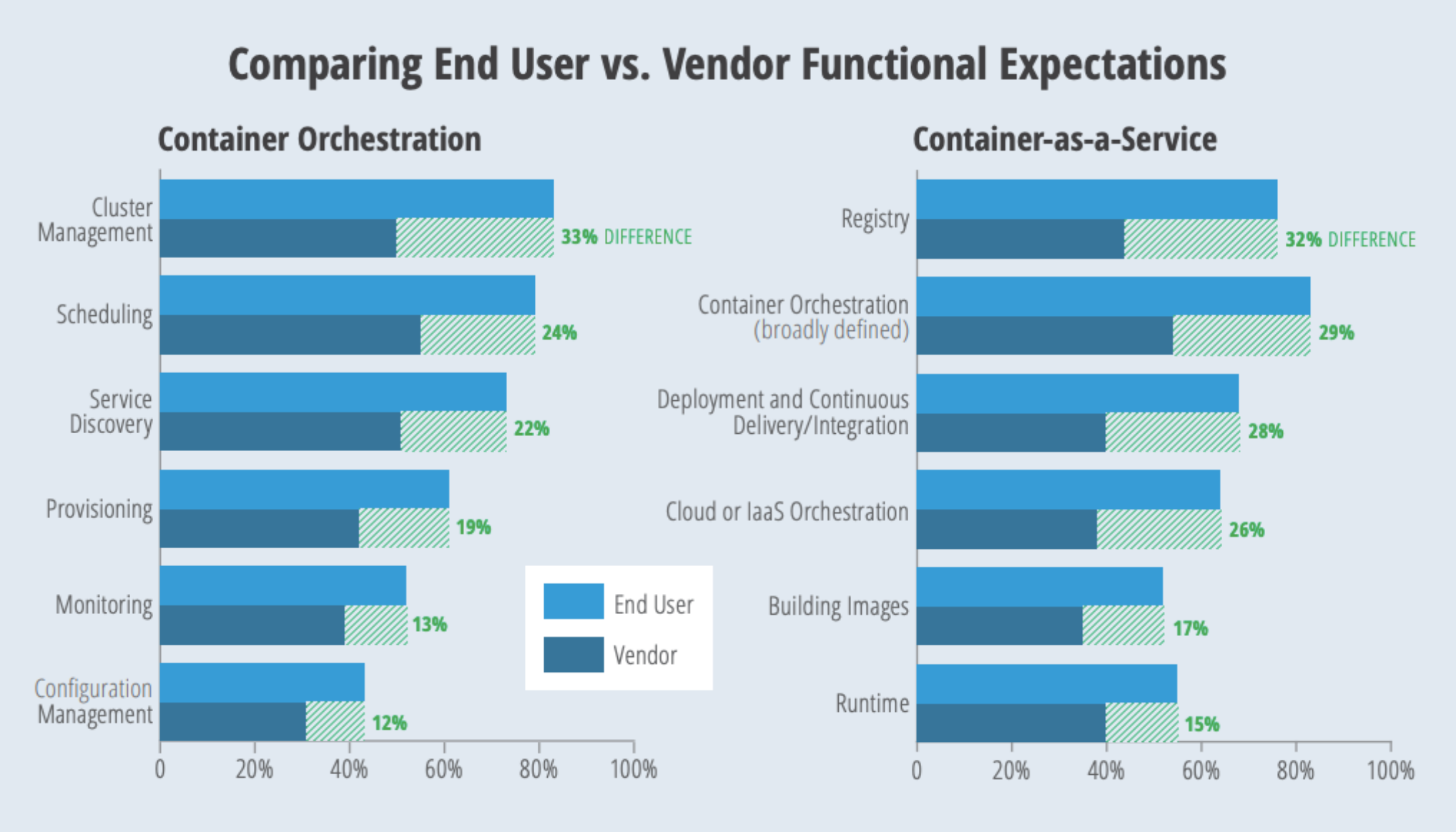

Figure: In characterizing container orchestration and CaaS, the vendors were stingy.

Source: http://thenewstack.io/wp-content/uploads/2016/06/Chart_Comparing-End-User-vs-Vendor-Functional-Expectations.png

Products/services used for container management and orchestration

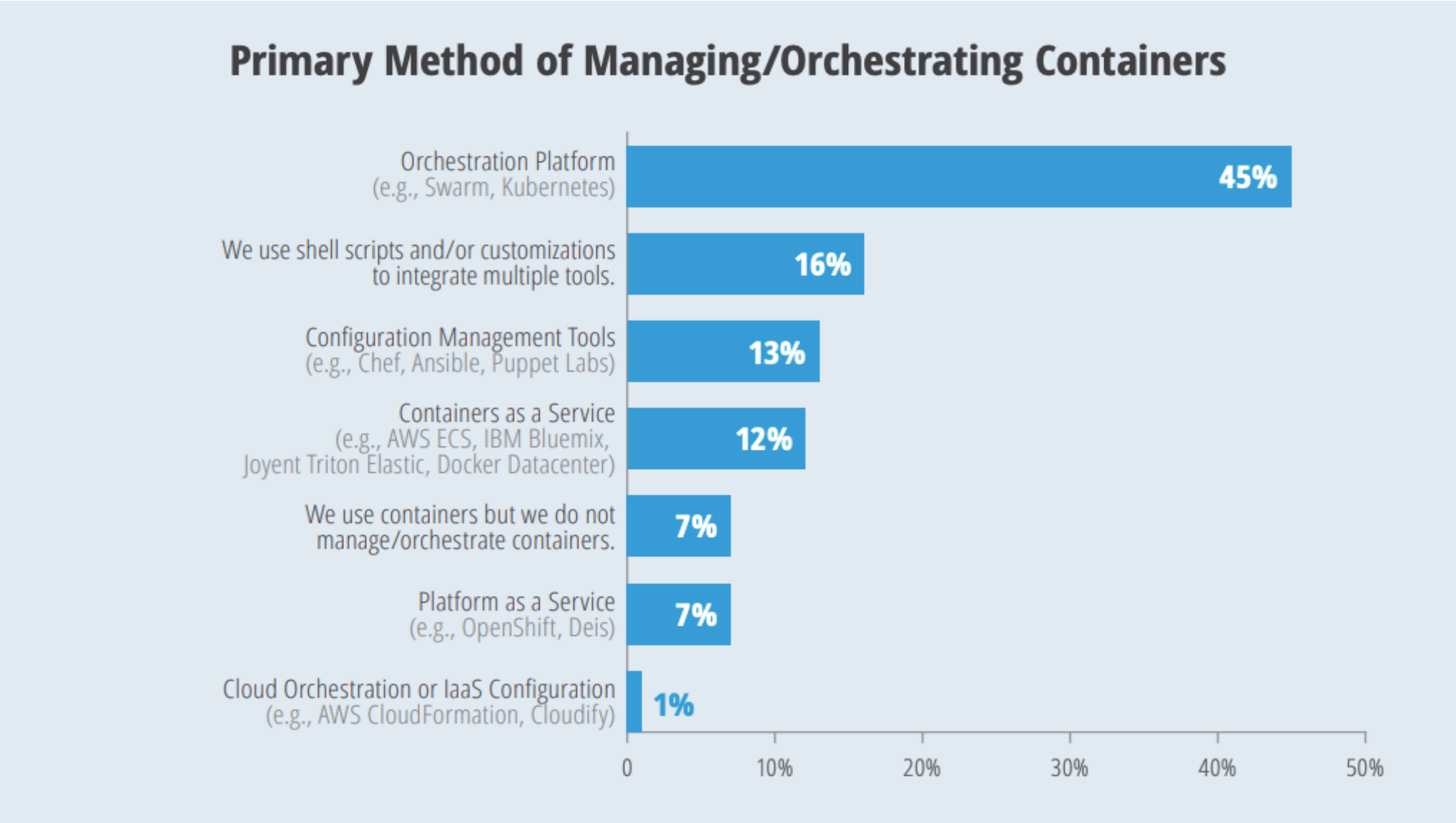

Figure: Container supervision for 4 5% of end customers who use, or test containers is based on platforms focused on orchestrating.

Source: http://thenewstack.io/wp-content/uploads/2016/06/Chart_Primary-Method-of-Managing-Orchestrating-Containers.png

Container control is reduced by the use of configuration management tools

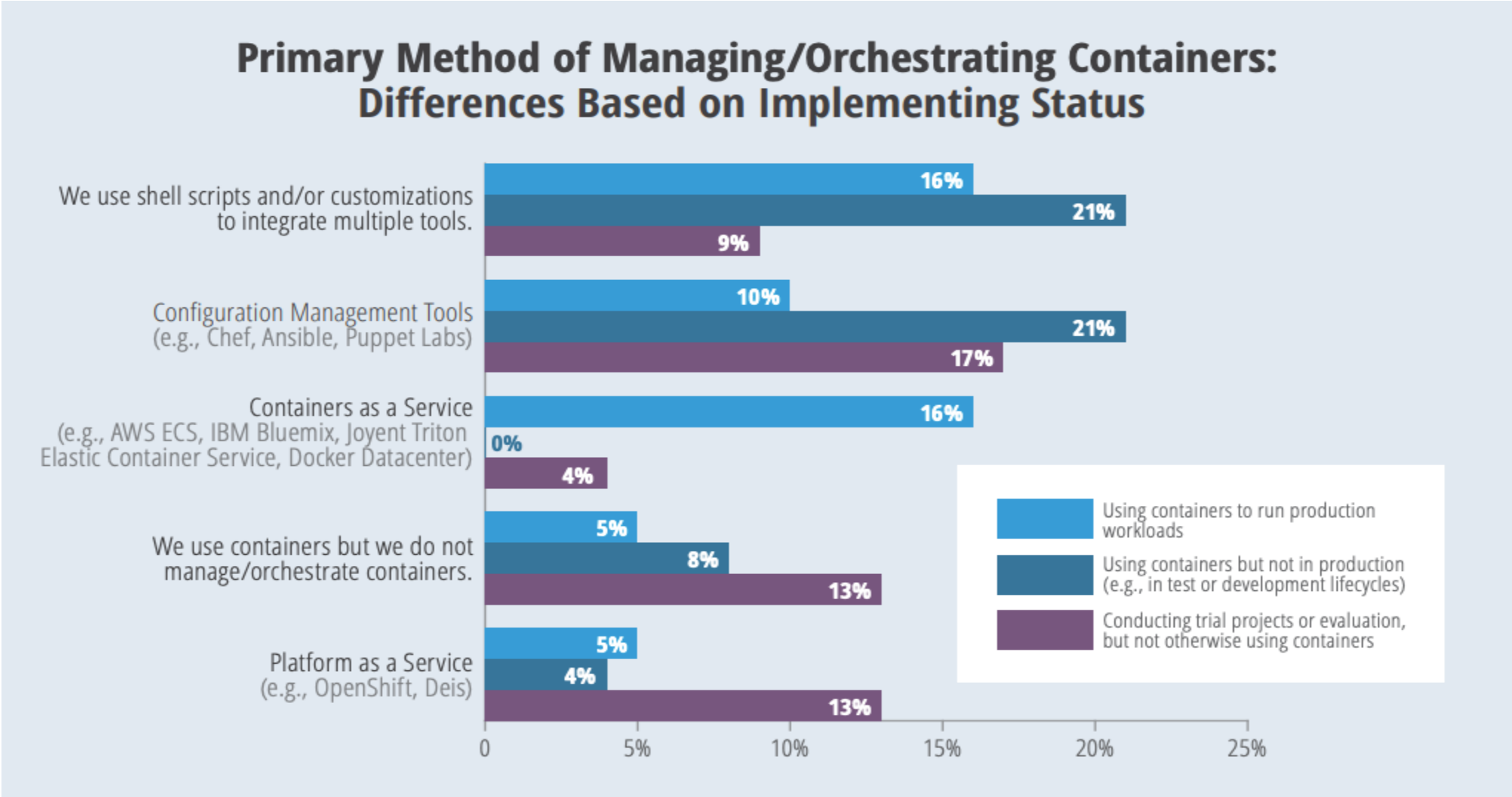

Figure: Container control is reduced using configuration management tools as companies move into production.

Source: http://thenewstack.io/wp-content/uploads/2016/06/Chart_Primary-Method-of-Managing-Orchestrating-Containers-Differences-by-Implementation-Status.png

Orchestration primary method

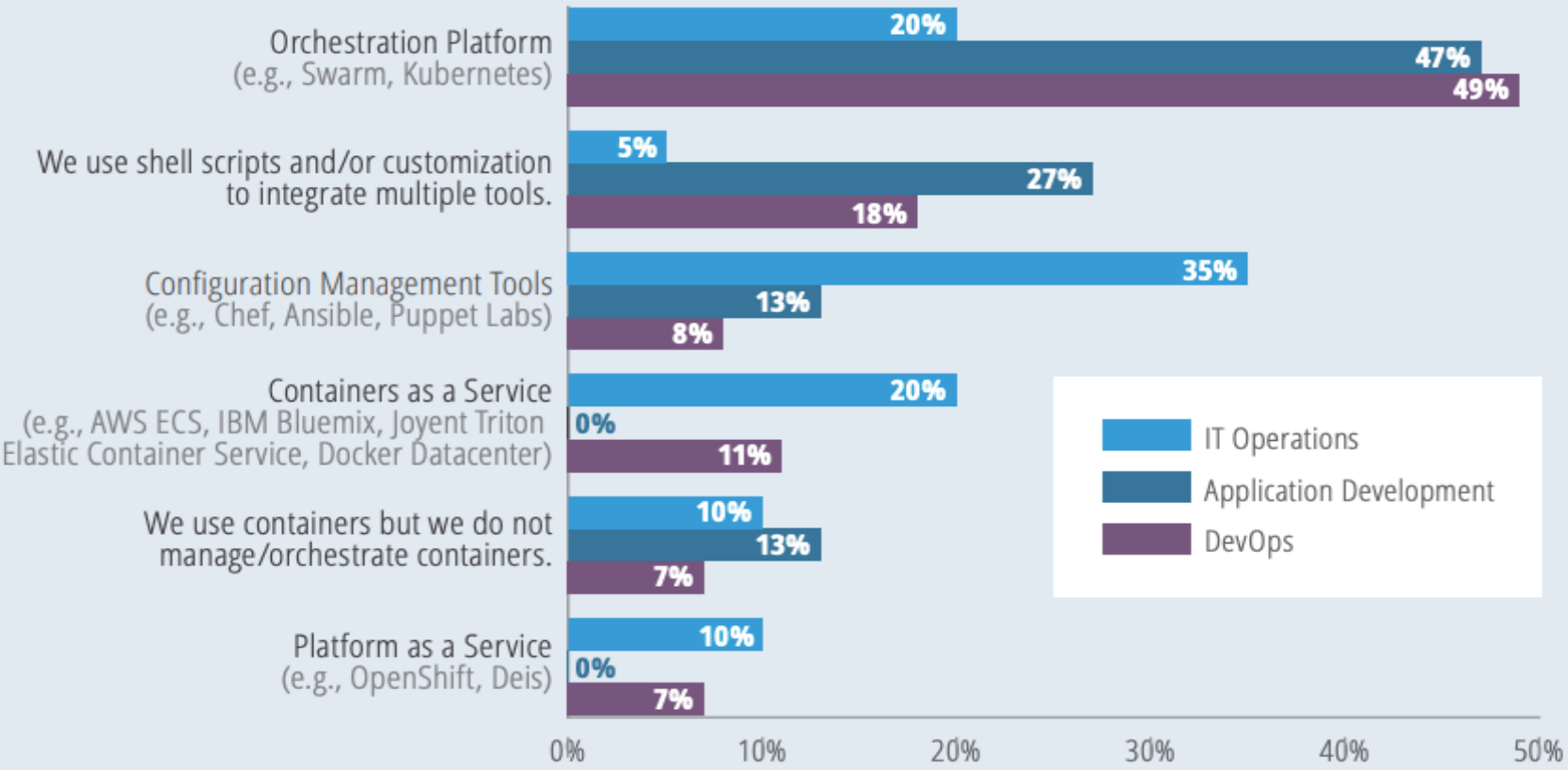

Figure: In its tools for configuring IT operations CaaS and orchestral platforms such as Swarm, Kubernetes and Mesos prevail.

Source: http://thenewstack.io/wp-content/uploads/2016/06/Chart_Primary-Method-of-Managing-Orchestrating-Containers-Differences-Based-on-Job-Roles-v2.png

Top orchestration products

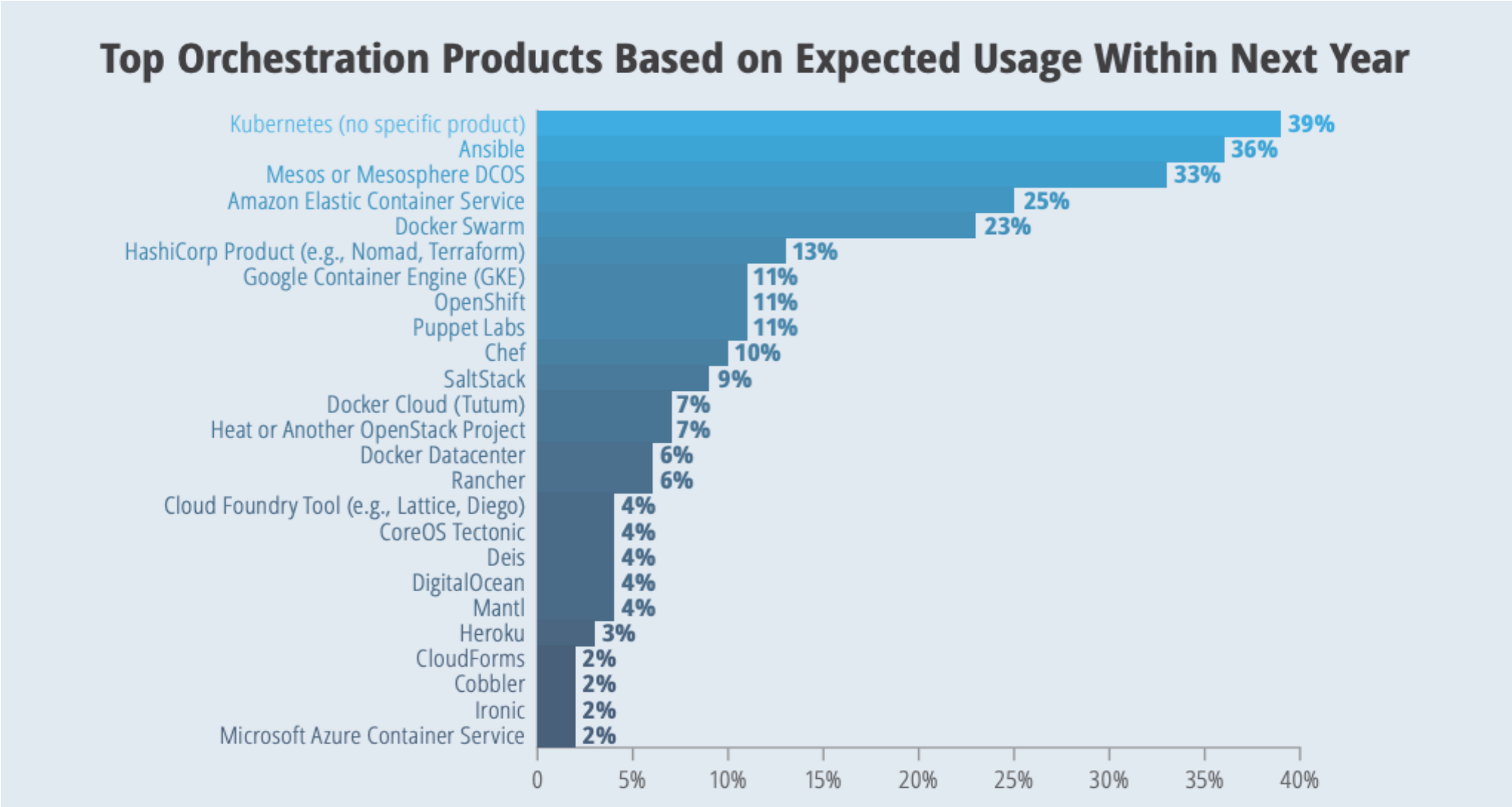

Figure: Container orchestration arrangements for Kubernetes, Ansible, Mesos/Mesosphere, Amazon ECS and Docker Swarm top customers.

Source: http://thenewstack.io/wp-content/uploads/2016/06/Chart_Top-Orchestration-Products-Based-on-Expected-Usage-Within-Next-Year.png

Expected top orchestration products

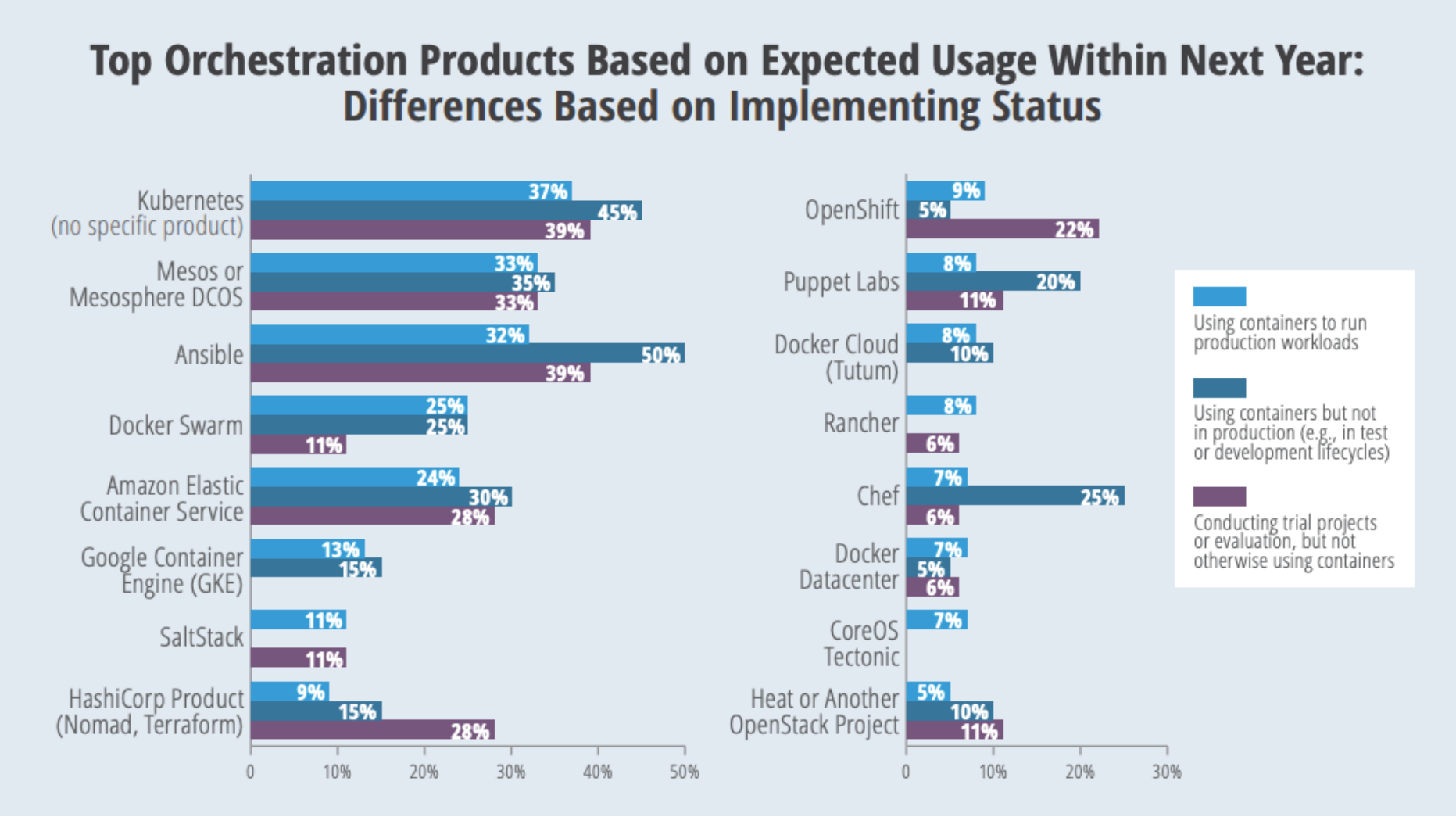

Figure: Among those directing preliminary projects or evaluations, Hashicorp and OpenShift are most often believed of.

Source: http://thenewstack.io/wp-content/uploads/2016/06/Chart_Top-Orchestration-Products-Based-on-Expected-Usage-Within-Next-Year-Differences.png

Service discovery tools

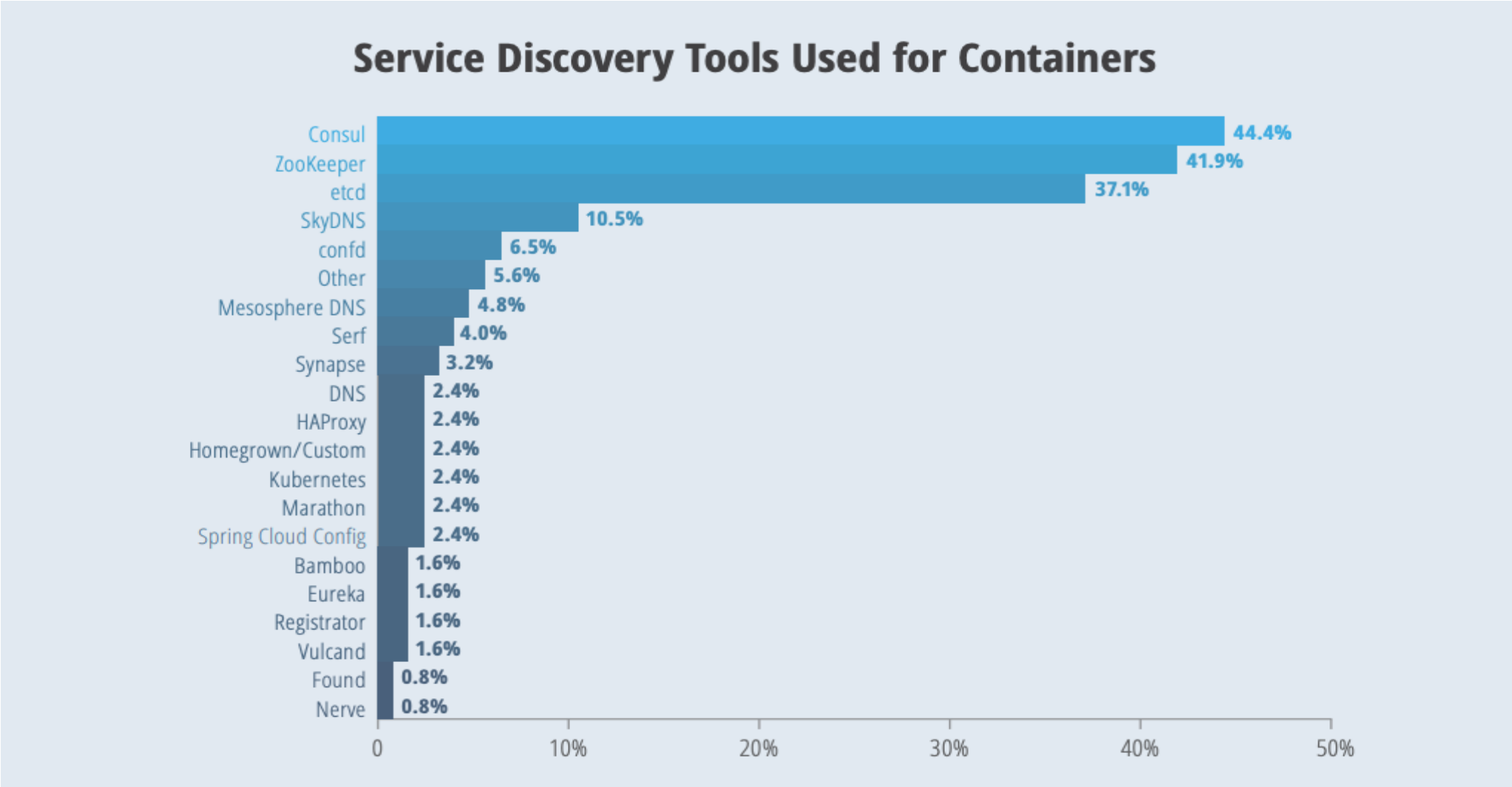

Figure: Discover services of Consul, zookeeper and others which are used frequently used than other tools

Source: http://thenewstack.io/wp-content/uploads/2016/06/Chart_Service-Discovery-Tools-Used-for-Containers.png

Planning tools

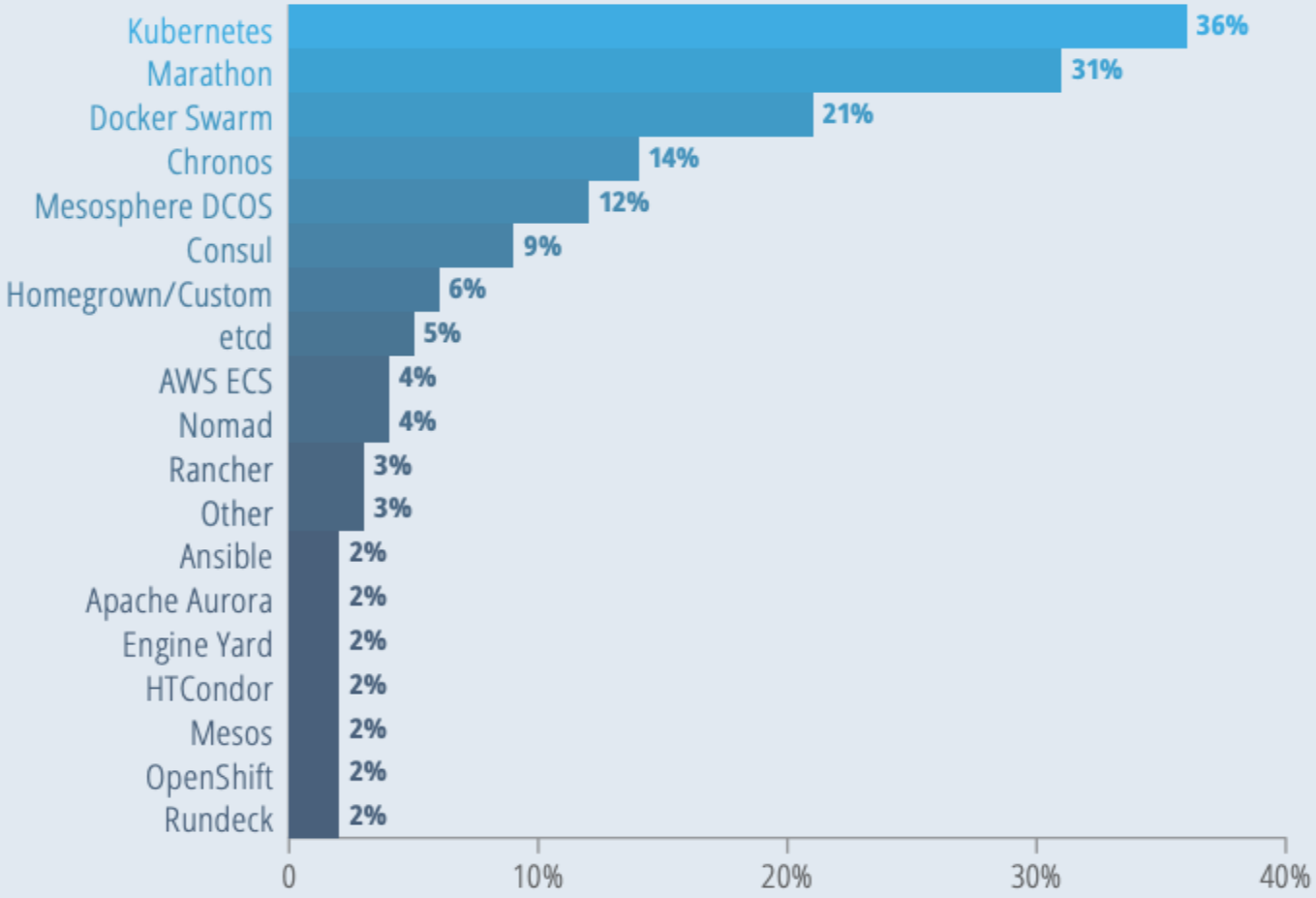

Figure: Generally planned for open sources tools like Kubernetes, Marathon and Swarm.

Source: http://thenewstack.io/wp-content/uploads/2016/06/Chart_Scheduling-Tools-Used-for-Containers.png

Cluster management

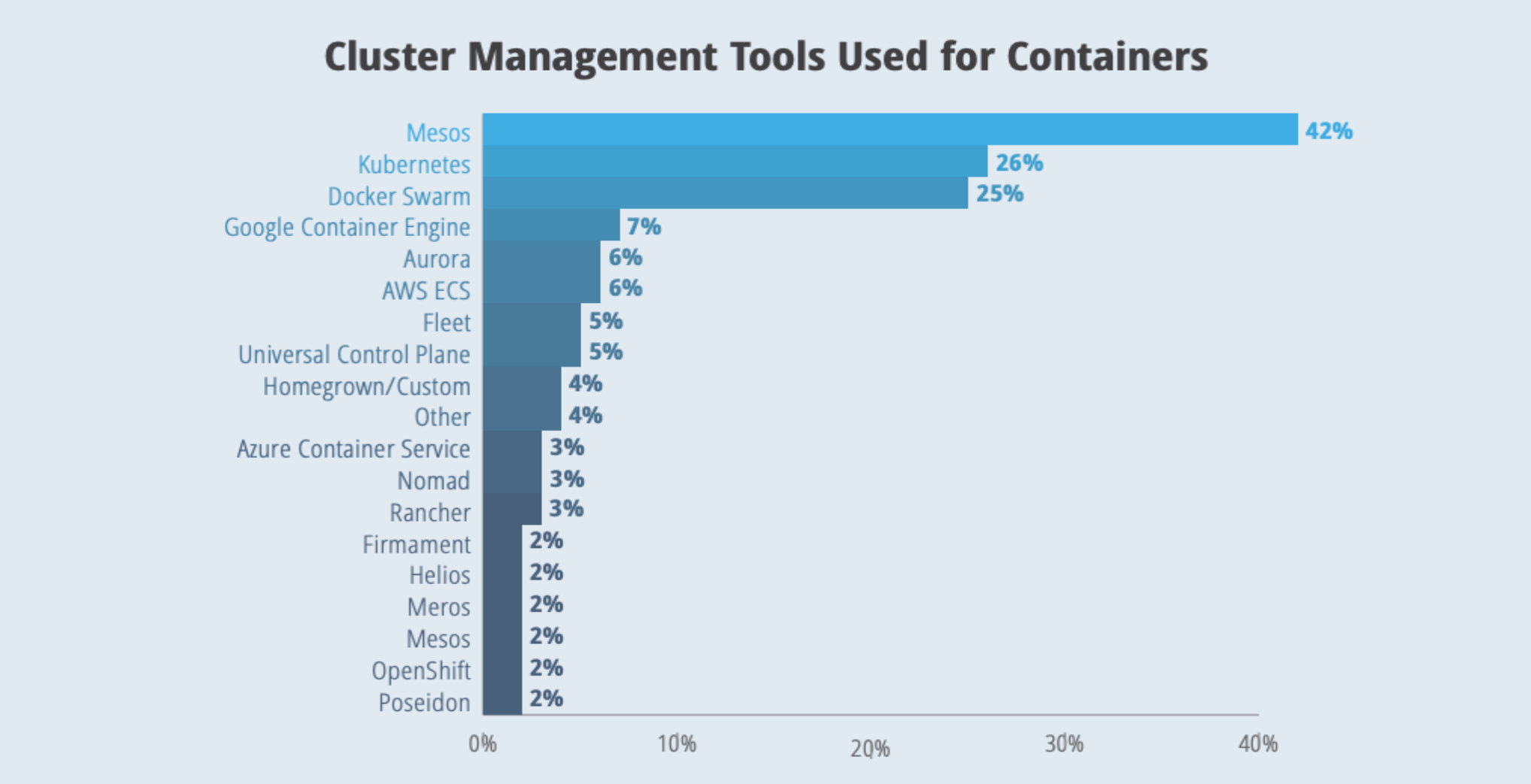

Figure: Cluster management is a tripartite battle between Kubernetes, Swarm and Mésos, taking the trend of evaluating the Mesospheric environment into account.

Source: http://thenewstack.io/wp-content/uploads/2016/06/Chart_Cluster-Management-Tools-Used-for-Containers-rev-10-17-16.png

Acceptances

- Container Orchestration Means:
  - It is usually acknowledged that scheduling, cluster management and service discovery are a piece of container orchestration. Nonetheless, over part of participants also saw provisioning and checking as a function of orchestration.
  - Docker's Swarm is viewed by enormous numbers for those who want to use Docker Cloud and Docker Datacentre as the hidden technology.
  - While research remains to be done, countless Mesos and Kubernetes customers understand what the basic technology uses.
  - Commonly used strategy remains container orchestration platforms. However, when taking a gander at explicit offerings, clients are well on the way to state that they are using Kubernetes which, by themselves, is not an item.

What is Kubernetes?

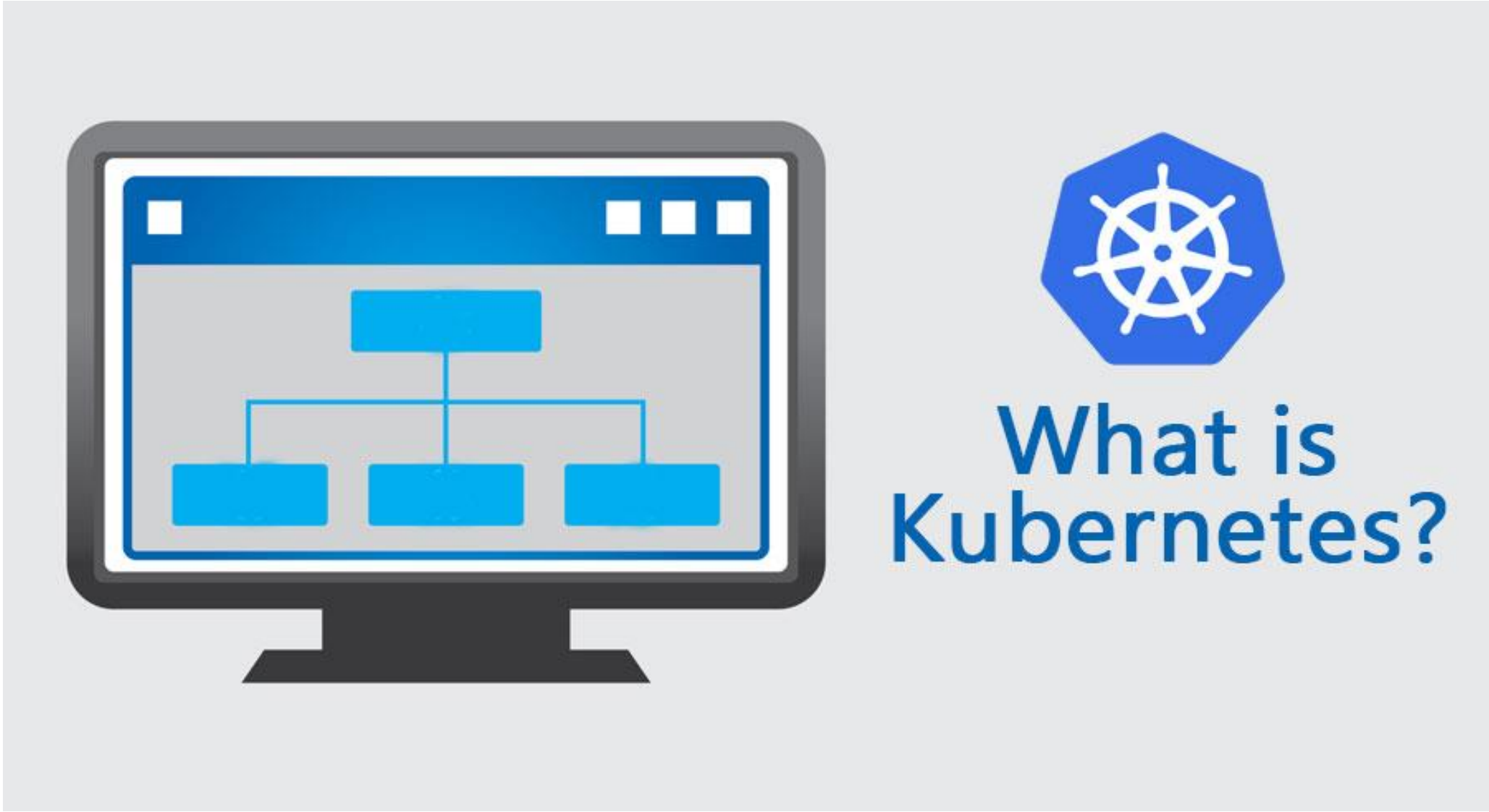

Figure: Kubernetes

Source: https://cdn.educba.com/academy/wp-content/uploads/2019/05/What-is-Kubernetes.jpg

Clusters and architecture

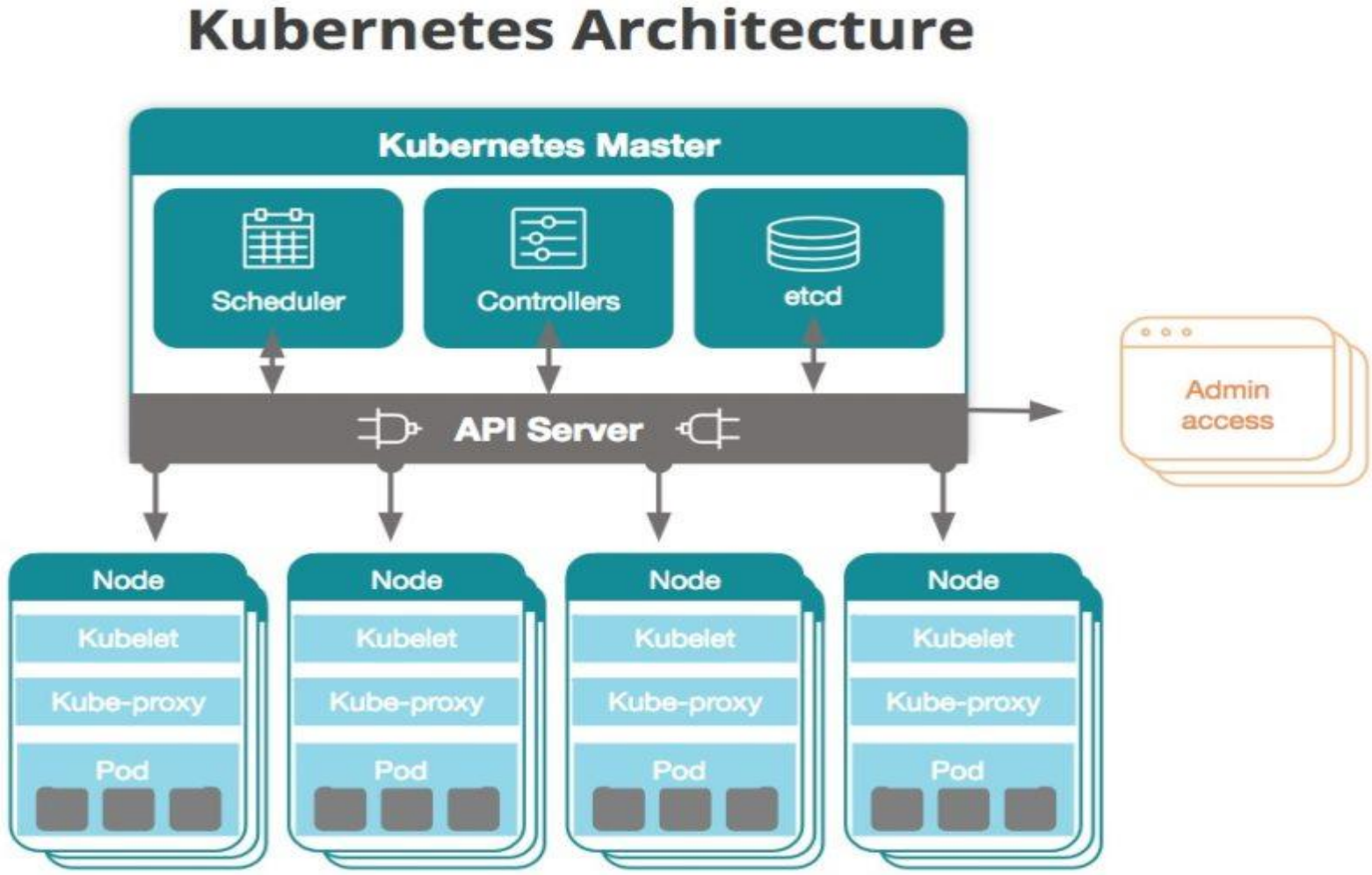

Figure: Clusters and architecture

Source: https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRQXqVPOtQLTzQg0kIdXuLika6jbvJaJ8se7I3kDhlzsg-Ngweqands

Docker file instructions: CMD

Figure: Docker file instructions : CMD

Source: https://www.tigera.io/wp-content/uploads/2019/10/IBM-Cloud-Kubernetes-Service-clouds.png

Benefits

Figure: Benefits

Source: https://dzone.com/storage/temp/12575486-benifites-of-kubernetes-microservices-architecture.png

Kubernetes and DevOps

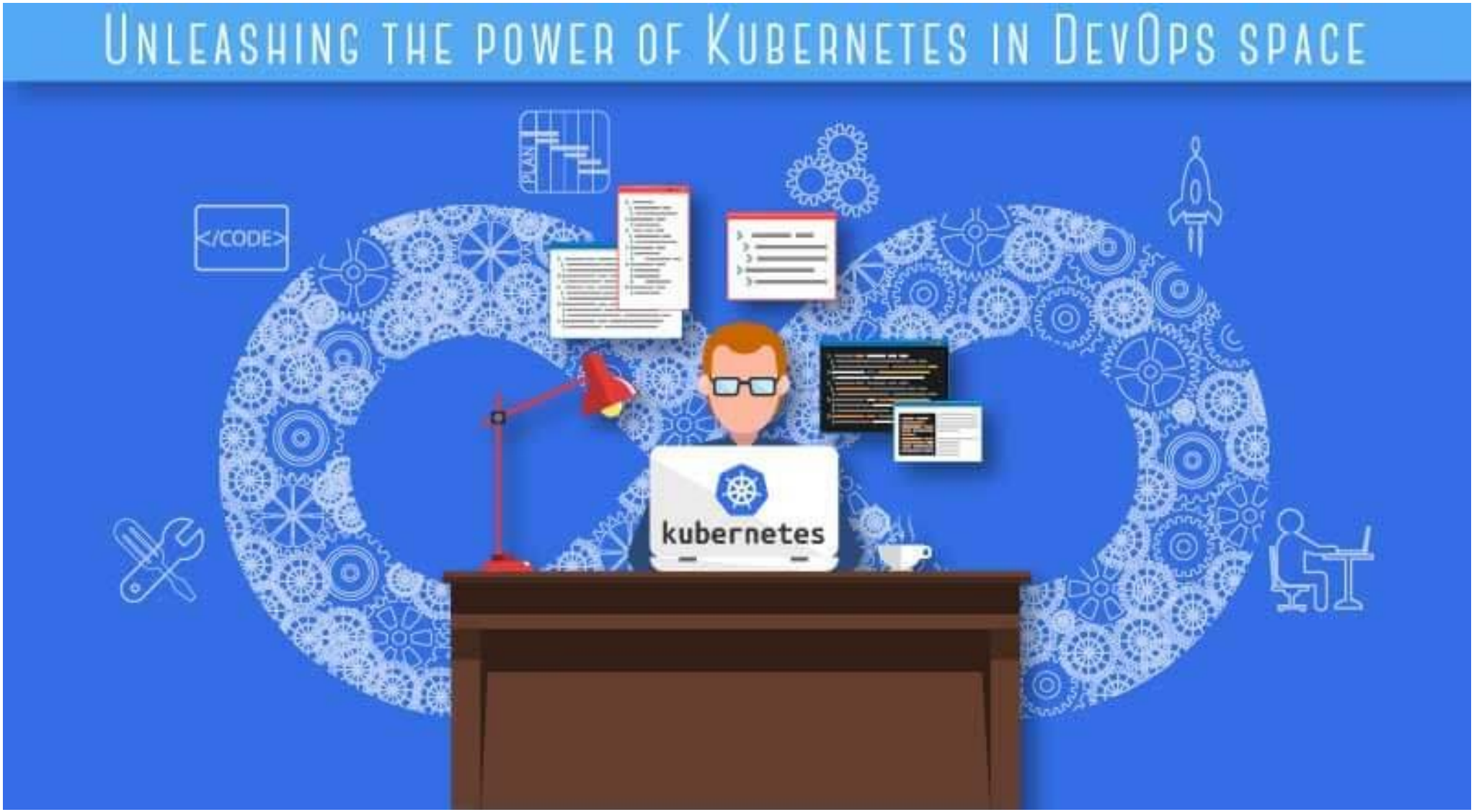

Figure: Kubernetes and DevOps

Source : https://cdn.spec-india.com/wp-content/uploads/2018/10/DevOps-and-Kubernetes.jpg

Kubernetes vs Docker

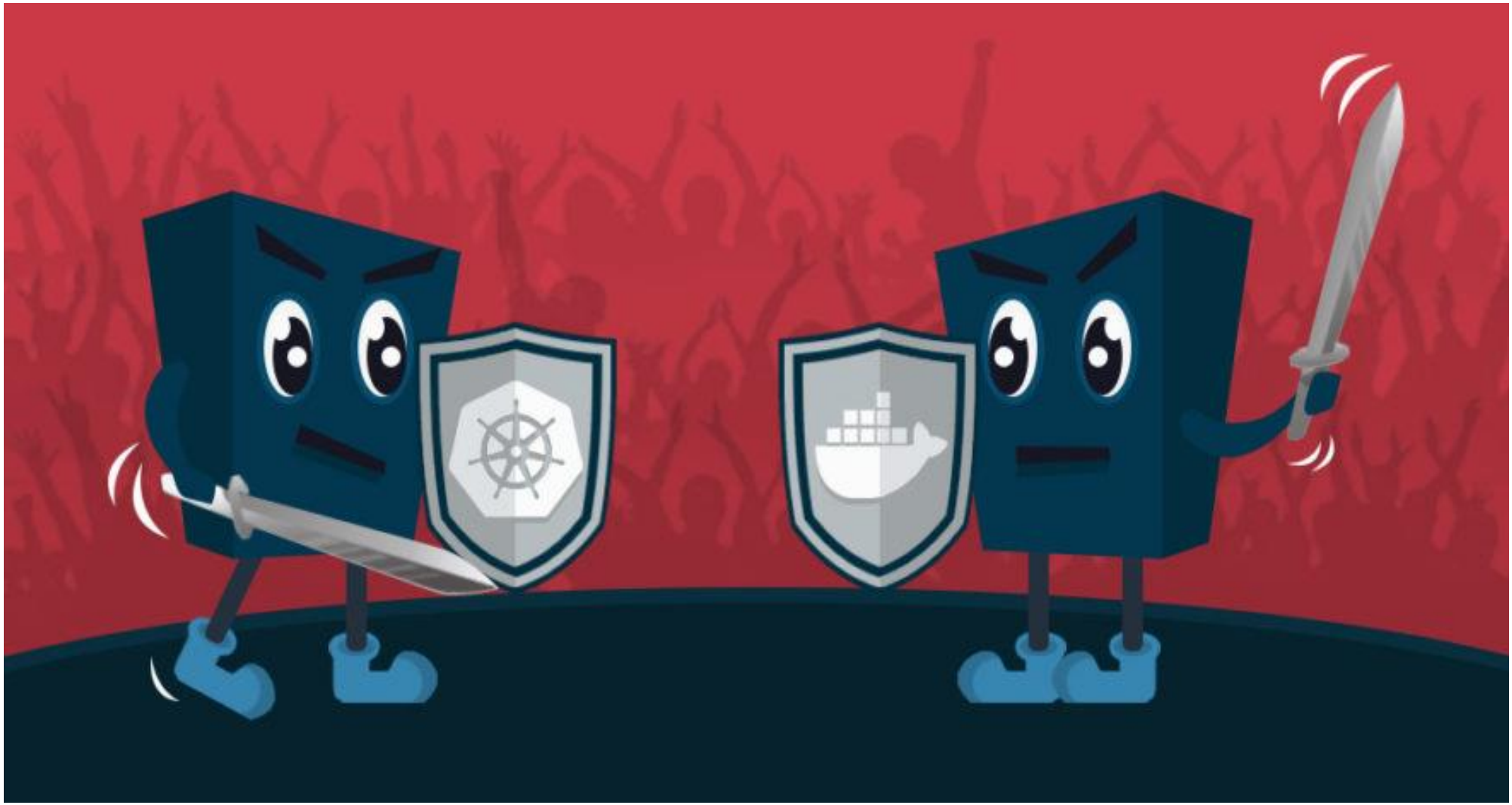

Figure: Kubernetes vs Docker

Source: https://stackify.com/wp-content/uploads/2019/05/kubernetes-vs-docker-881x441.jpg

Kubernetes and IBM

Figure:Kubernetes and IBM

Source: https://d33wubrfki0l68.cloudfront.net/817bfdd83a524fed7342e77a26df18c87266b8f4/3da7c/images/docs/components-of-kubernetes.png

Kubernetes architecture

Kubernetes Architecture

Etcd - a highly-available key value store which K8s uses for persistent storage of all of its REST API objects

API Server – Kubernetes API server

Controller manager – Daemon that runs controllers (background threads that handle routine tasks). Includes Node Controller, Replication Controller (ReplicaSet), Endpoints Controller, Service Account * Token Controllers)

Scheduler – schedules pods in worker nodes

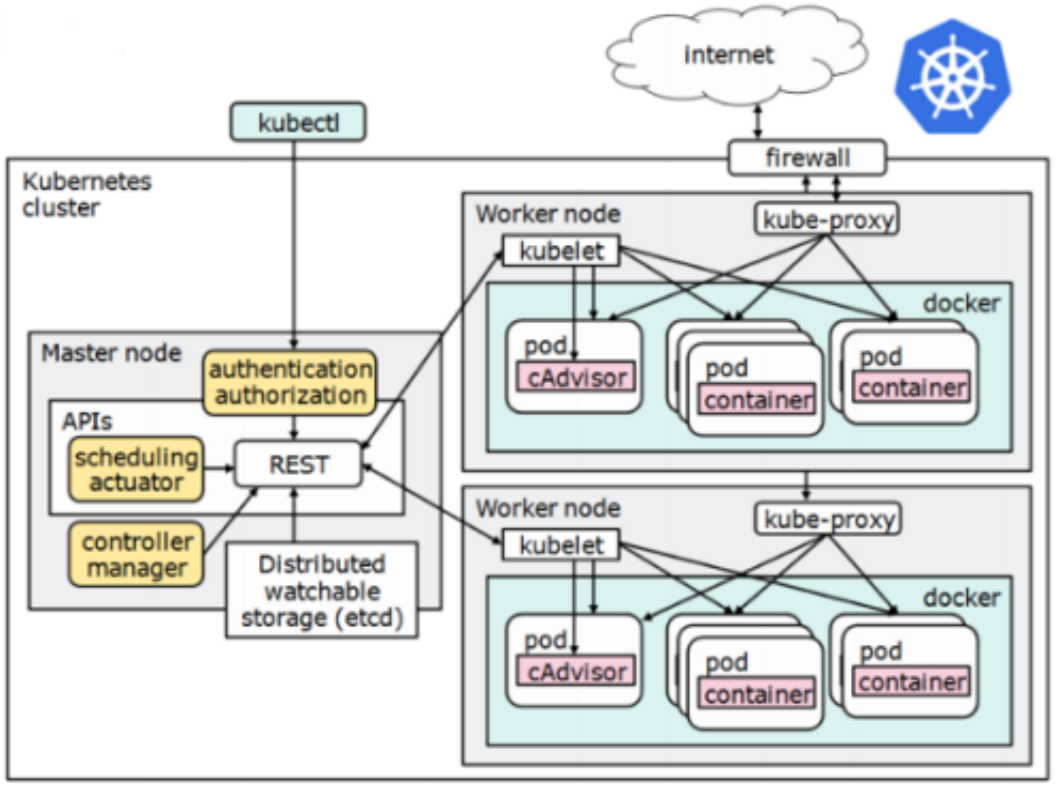

Figure:Kubernetes architecture

Source: https://d33wubrfki0l68.cloudfront.net/817bfdd83a524fed7342e77a26df18c87266b8f4/3da7c/images/docs/components-of-kubernetes.png

Decentralized approach

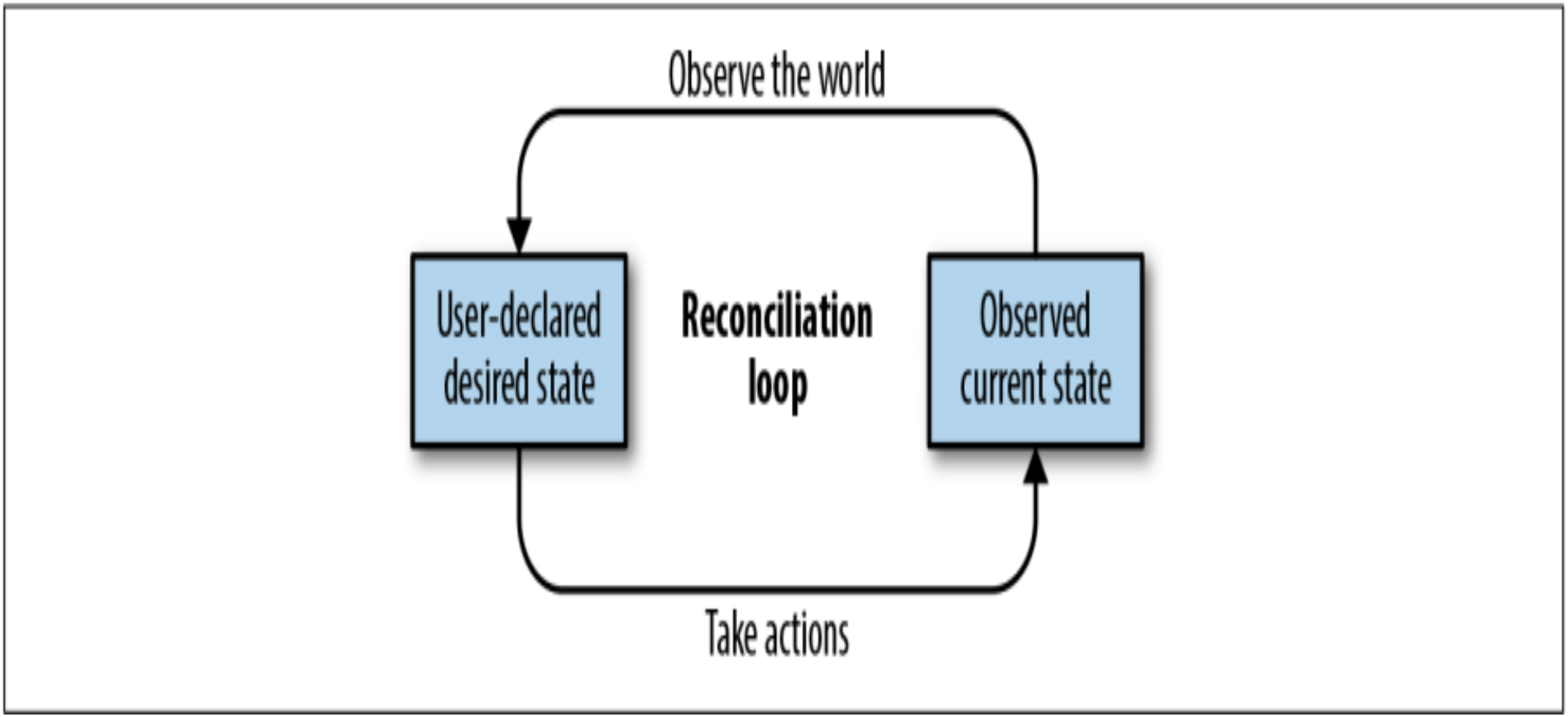

Figure: A delineation of conventional compromise circle

Dynamic grouping

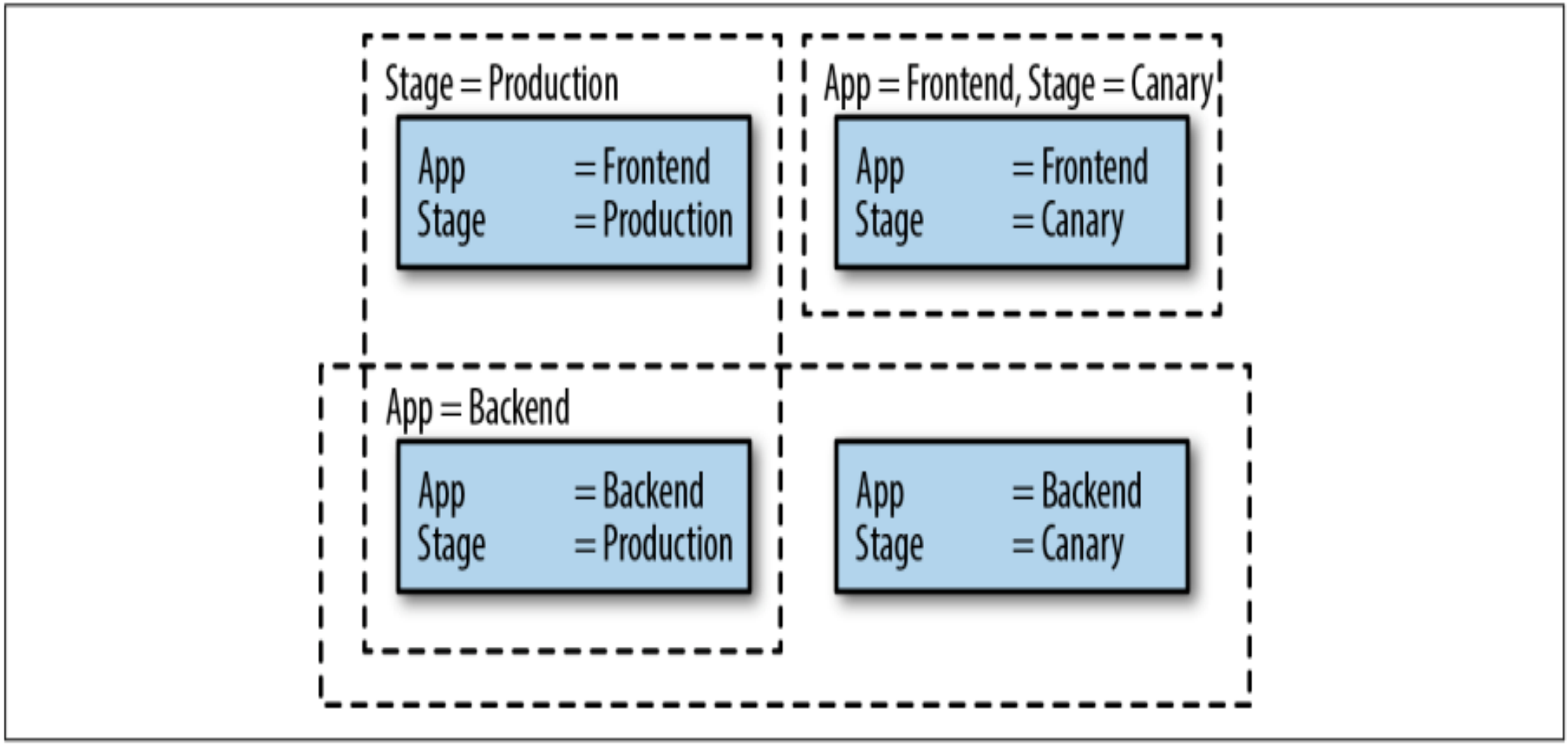

Figure: Instances of marks and name choice

Kubernetes structure

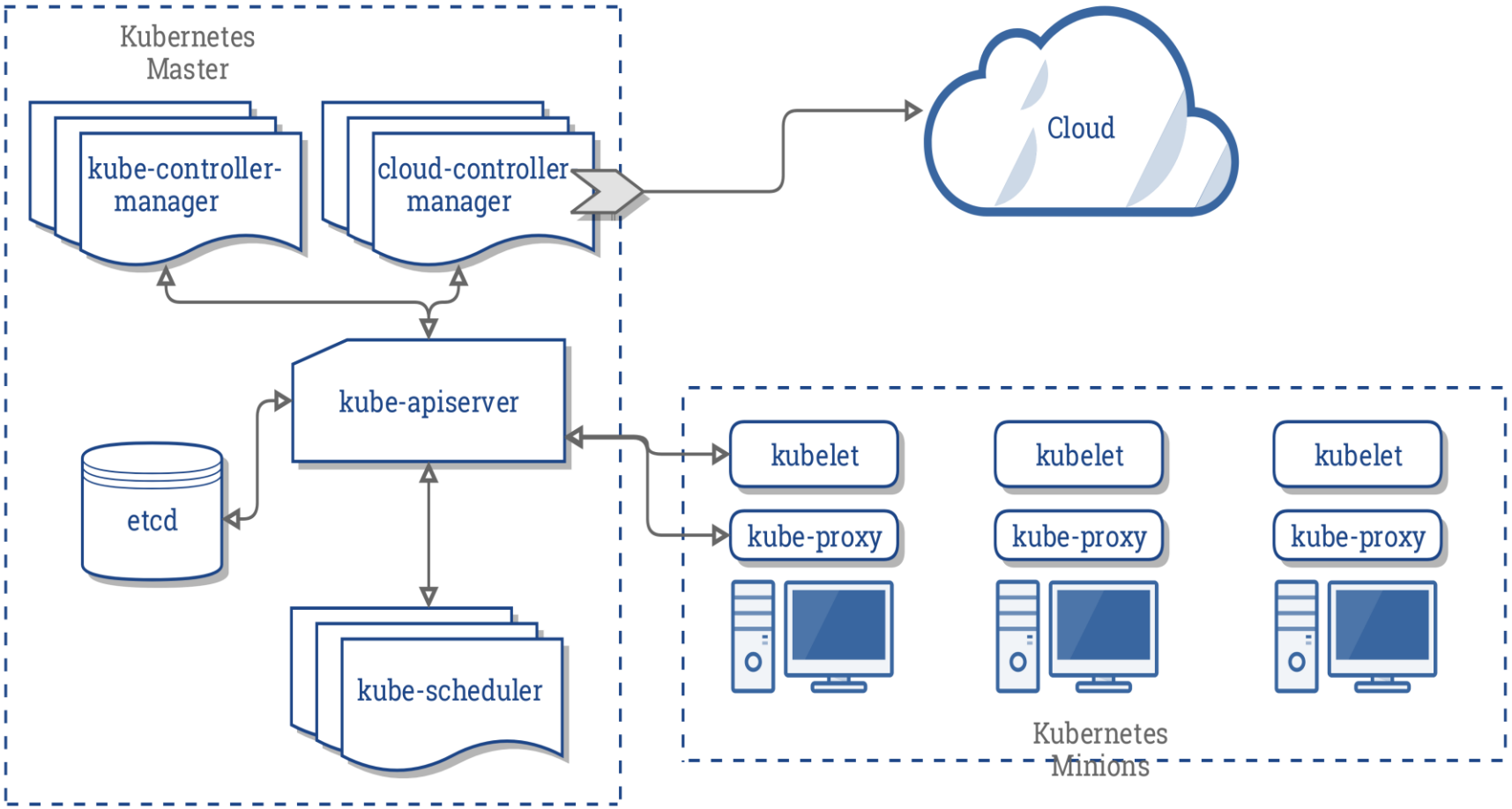

Figure: Kubernetes structure

Source: https://d33wubrfki0l68.cloudfront.net/817bfdd83a524fed7342e77a26df18c87266b8f4/3da7c/images/docs/components-of-kubernetes.png

Essential characteristics for manageability

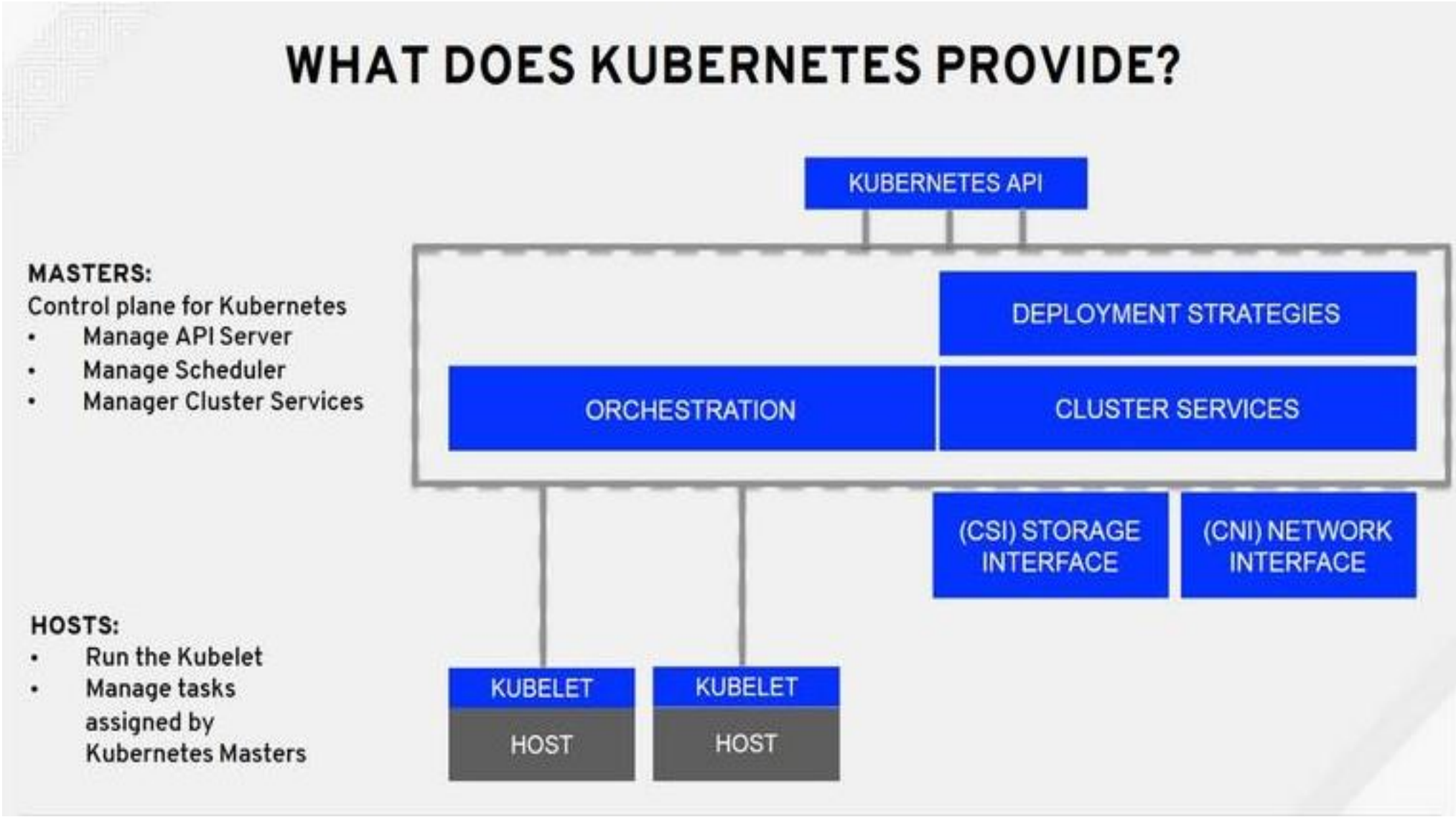

Figure: Essential characteristics for manageability

Source: https://www.oreilly.com/library/view/managing-kubernetes/9781492033905/assets/mgk8_0401.png

API discovery and translation


```
$ curl localhost:8001/api
{
  "kind": "APIVersions",
  "versions": [
    "v1"
  ],
  "serverAddressByClientCIDRs": [
    {
      "clientCIDR": "0.0.0.0/0",
      "serverAddress": "10.0.0.1:6443"
    }
  ]
}
```

```
{
  "name": "pods/attach",
  "singularName": "",
  "namespaced": true,
  "kind": "Pod",
  "verbs": []
}
```

Life of a request

- We shall separate the processing of one request for the API server in order to better comprehend what the API server is doing for each of these various demands.

- Authentication:
  - Authentication, that creates the identity connected with this request, is the first phase in application processing.

- RBAC/Authorization:
  - After an application's identity has been determined by the API server, it passes on to authorizing it. Every implementation in Kubernetes application follows a conventional model of RBAC.

- Admission control:
  - Upon authentication and authorisation of the request, the application passes to an admission check.

- Validation:
  - Validation of the requests occurs after entry control but can also be done as a portion of admission checks, especially for inner webhook-based validations.

Specialized requests

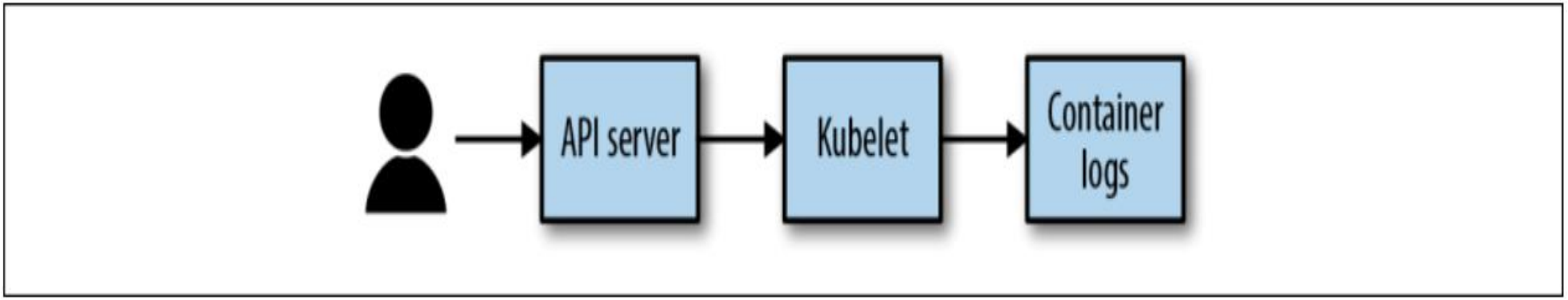

Figure: The fundamental flow of an HTTP container log request

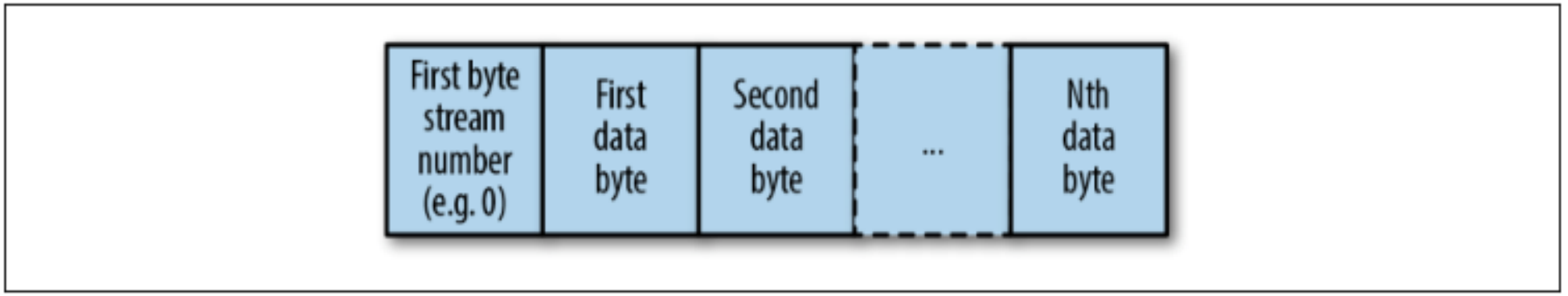

Figure: An example of multi-channel framework for the Kubernetes Web Socket

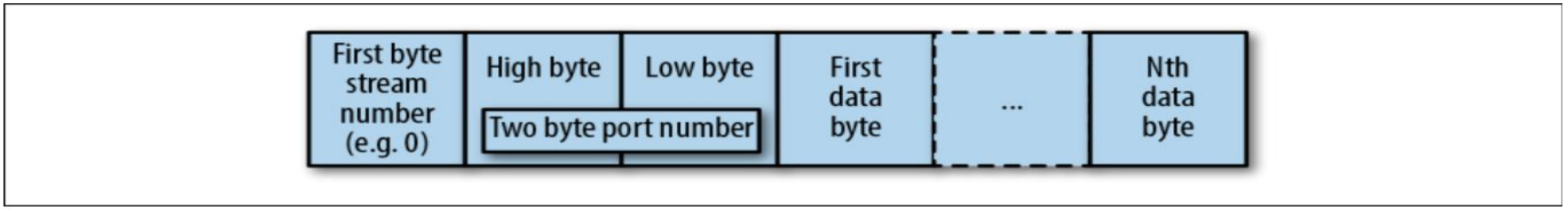

Figure: An instance of the web-based port forwarding information framework

Watch operations

Figure: Watch operations

Source : https://opensource.com/sites/default/files/styles/image-full-size/public/lead-images/data_metrics_analytics_desktop_laptop.png?itok=9QXd7AUr

API server internals


```
for crd in AllCustomResourceDefinitions:
    if !RegisteredPath(crd):
       registerPath

for path in AllRegisteredPaths:
    if !CustomResourceExists(path):
       markPathInvalid(path)
       delete custom resource data
       delete path
```

```
I0803 19:59:19.929302   1 trace.go:76] Trace[1449222206]:
 "Create /api/v1/namespaces/default/events" (started: 2018-08-03
 19:59:19.001777279 +0000 UTC m=+25.386403121) (total time: 927.484579ms):
Trace[1449222206]: [927.401927ms] [927.279642ms] Object stored in database
I0803 19:59:20.402215 1 controller.go:537] quota admission added evaluator
 for: { namespaces}
```

```
schedule(pod): string
    nodes := getAllHealthyNodes()
    viableNodes := []
    for node in nodes:
                        for predicate in predicates:
                            if predicate(node, pod):
                viableNodes.append(node)

    scoredNodes := PriorityQueue<score, Node[]> priorities :=
    GetPriorityFunctions() for node in viableNodes:
        score = CalculateCombinedPriority(node, pod, priorities)
        scoredNodes[score].push(node)

    bestScore := scoredNodes.top().score
    selectedNodes := []
          while scoredNodes.top().score == bestScore:
            selectedNodes.append(scoredNodes.pop())

    node := selectAtRandom(selectedNodes)
    return node.Name
```

Conflicts


```
kind: Pod
...
spec:
  affinity:
    nodeAffinity:
      requiredDuringSchedulingIgnoredDuringExecution:
        nodeSelectorTerms:
        - matchExpressions:
          # foo == A or B - key: foo
            operator: In values:
            - A
            - B
...
```

Checkpoint (1 of 2)

Multiple choice questions:

1. The goal of Docker Swarm _____.
   a) Use a similar Docker API that works with the core Docker engine.
   b) Use SOAP connector to connect with Docker.
   c) Use application to run container inside the physical server.
   d) Non of the above.

2. The architecture of Kubernetes relies on a _____.
   a) Just master server.
   b) Master server with various components.
   c) API endpoint.
   d) Non of the above.

3. Apache MESOS is manager of the cluster that makes running functions:
   a) On a pool of data centre.
   b) On a pool of common servers more predictable.
   c) On a pool of array volumes.
   d) Non of the above.

Checkpoint solutions (1 of 2)

Multiple choice questions:

1. The goal of Docker Swarm _____.
   - **a) Use a similar Docker API that works with the core Docker engine .**
   - b) Use SOAP connector to connect with Docker.
   - c) Use application to run container inside the physical server.
   - d) Non of the above.

2. The architecture of Kubernetes relies on a _____.
   - a) Just master server.
   - **b) Master server with various components.**
   - c) API endpoint.
   - d) Non of the above.

3. Apache MESOS is manager of the cluster that makes running functions:
   - a) On a pool of data centre.
   - **b) On a pool of common servers more predictable.**
   - c) On a pool of array volumes.
   - d) Non of the above.

Checkpoint (2 of 2)

Fill in the blanks:

1. Red Hat customer choose _____ as a way to integrate other Red Hat contributions in its configuration management.
2. Docker Swarm Kubernetes and Marathon were frequently used tools for _____ planning.
3. _____ is a open source container orchestration tool.
4. IBM cloud pack utilizes _____ and IBM technologies to allow clients to generate cloud applications.

True or False:

1. DevOps combines IT operations and hardware deployment processes. True/False
2. Kubernetes and Mesosphere are orchestration tools. True/False
3. Amazon elastic cloud service is a VM handling service used on a virtual machine cluster for VM monitoring. True/False

Checkpoint solutions (2 of 2)

Fill in the blanks:

1. Red Hat customer choose **Ansible** as a way to integrate other Red Hat contributions in its configuration management.
2. Docker Swarm Kubernetes and Marathon were frequently used tools for **container** planning.
3. **Kubernetes** is a open source container orchestration tool.
4. IBM cloud pack utilizes **Docker** and IBM technologies to allow clients to generate cloud applications.

True or False:

1. DevOps combines IT operations and hardware deployment processes. **False**
2. Kubernetes and Mesosphere are orchestration tools. **True**
3. Amazon elastic cloud service is a VM handling service used on a virtual machine cluster for VM monitoring. **False**

Question bank

Two mark questions:

1. Define Apache Mesos.
2. What is kubernets?
3. Define container as a services.
4. Define planning.

Four mark questions:

1. Explain security.
2. Explain advantages of managed kubernets.
3. Explain types of request.
4. Explain implicit or dynamic groping.

Eight mark questions:

1. Explain in detail head node components.
2. Explain key components in orchestration.

Unit summary

Having completing this unit, you should be able to:

- Learn the concept of automation and orchestration
- Understand key concepts in orchestration
- Gain knowledge on bridging realities, orchestration and programmable infrastructure
- Understand the concept of open source and standards
- Learn about peer perspectives on container orchestration survey
- Gain knowledge on cisco-cloud-native capabilities and a deeper user experience
- Gain an insight into consideration for containers in production