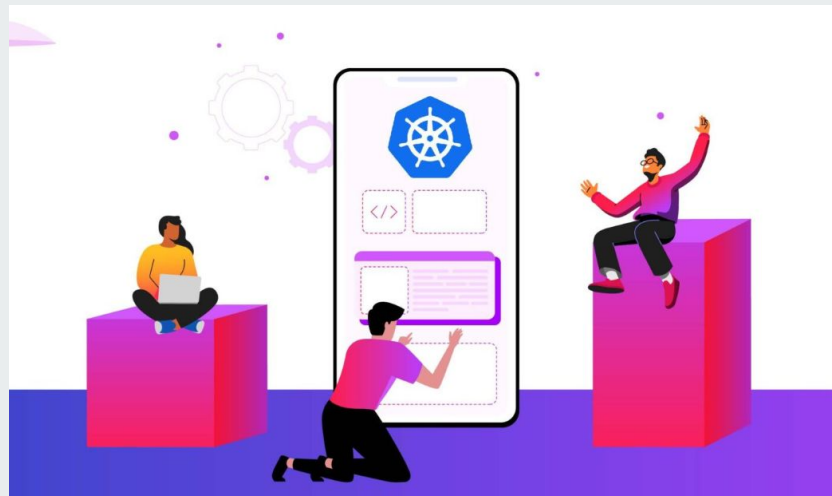


Kubernetes 3

Pods Allocations & Deployments in Kubernetes

caodangsao@gmail.com



Module Target

Kết thúc bài học, học viên cần đạt được các kỹ năng sau:

- Có thể hiểu được cơ chế lập lịch Pod trong Kubernetes.
- Hiểu và triển khai được ứng dụng dưới dạng daemonset, static pod.
- Có thể sử dụng node affinity để cho phép ứng dụng chạy trên node chỉ định.
- Hiểu được cơ chế scaling ứng dụng trong K8s, Replication controller làm gì và replicaset dùng như thế nào.
- Có khả năng triển khai ứng dụng dưới dạng deployments.

Nội dung

1 Pods Allocation in K8s

- Lập lịch Pods
- DaemonSets in K8s
- Static Pods & Node Affinity in K8s

2 Deployments in K8s

- Scaling Application in K8s
- ReplicaSet in K8s
- Deployments in K8s

3 Summary

- Tóm tắt nội dung buổi học



Lab 1: pod scheduling



Lab 2: daemonsets



Lab 3: static pod



Lab 4: node affinity



Lab 5: scaling with replication controller



Lab 6: replicaset



Lab 7: deployments

Section 1:

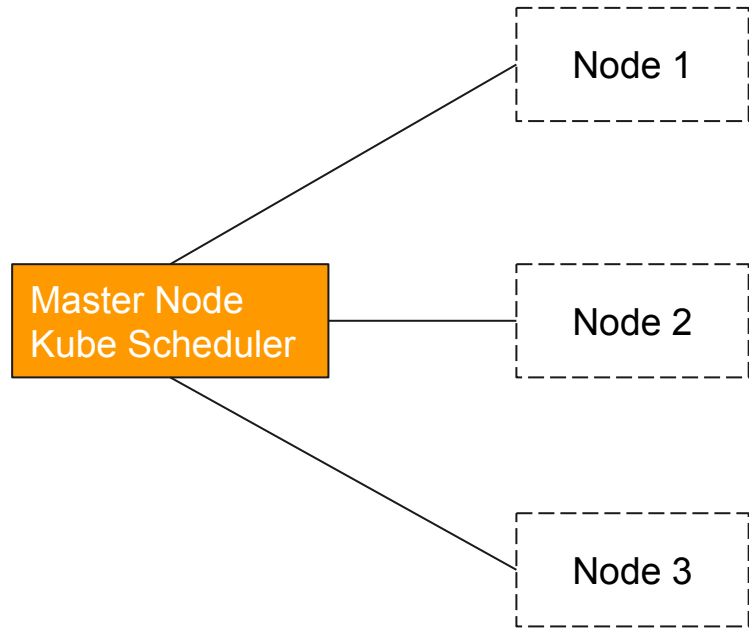
Pods Allocation in K8s

Resource Scheduling

- Scheduling là gì ?
- Quá trình Scheduling
- nodeSelector
- nodeName
- Lab demo

What is Scheduling ?

- Scheduling một quá trình gán các Pods cho các Node để Kubectl có thể chạy chúng.
- **Kube-Scheduler**: Nó là một thành phần chạy trên Master node, làm nhiệm vụ quyết định Pods được gán cho nodes nào.



Scheduling Process

- Kube-Scheduler **lựa chọn node phù hợp cho Pods**
- Yêu cầu cấp phát tài nguyên vào Tài nguyên node sẵn sàng
- Cấu hình giống như nhãn của Node
- nodeSelector, Affinity và Anti-Affinity

nodeSelector

- nodeSelector khai báo trong Pod Spec để giới hạn Nodes mà Pod có thể triển khai lên.
- NodeSelector sử dụng Labels để lựa chọn Node phù hợp.

```
apiVersion: v1
kind: Pod
metadata:
  name: cassandra
spec:
  containers:
    - name: cassandra
      image: cassandra
      nodeSelector:
        disktype: ssd
```


nodeName

- User có thể bỏ scheduling và gán Pods cho một Node cụ thể sử dụng Node Name.
- NodeName thường không phải lựa chọn tốt để dùng scheduling Pods bởi vì nó có giới hạn.

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx
spec:
  containers:
    - name: nginx
      image: nginx
      nodeName: kubeNode-01
```

Labs



Lab 1: pod scheduling

DaemonSets in K8s

- DaemonSets là gì?
- Cơ chế hoạt động DaemonSets
- DaemonSets và Scheduling
- Labs

DaemonSet

- Tự động chạy các bản copy của Pod trên tất cả các node.
- DaemonSet chạy bản copy của Pod trên node mới như chúng đã được thêm vào cụm.
- DaemonSet phù hợp với các trường hợp như Monitoring, Log Collection, Proxy Configuration, v.v.

Node 1

Node 2

Node 3

Scheduling & DaemonSets

- DaemonSets tuân theo quy tắc lập lịch thông thường bằng node labels, taints và tolerations.
- Nếu Pods không được lập lịch bình thường trên một node, thì daemonset cũng sẽ không tạo bản copy của Pod trên node đó.

Labs



Lab 2: daemonsets

Static Pods

- Static Pods là gì?
- Mirror Pod
- Labs

DaemonSet

- Static Pods được quản lý trực tiếp bởi Kubelet trên các Nodes Kubernetes.
- K8s API Server không giám sát Static Pods.
- Kubelet theo dõi trạng thái của các Static Pods này và restart nếu nó lỗi.
- Kubelet tự động tạo Static Pods từ tệp YAML nằm ở đường dẫn manifest chỉ định trên Node.

Mirror Pod

- Kubelet sẽ tạo Mirror Pod cho mỗi Static Pod.
- Mirror pods cho phép user giám sát Static Pods thông qua K8s APIs.
- User không thể thay đổi hay cập nhật Static Pods thông qua Mirror Pods.

Labs



Lab 3: static pod

Node Affinity

- NodeAffinity là gì?
- Ví dụ
- Labs

Node Affinity

- Node Affinity là phiên bản được nâng cấp hơn so với nodeSelector.
- Node Affinity được sử dụng để phân phối Pods trên các node worker.
- Không lập lịch Pod trên các node thỏa mãn Node Anti-Affinity.
- Anti-Affinity ngược lại với khái niệm Affinity và NodeSelector.

Node Affinity

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx-nodeselector
spec:
  containers:
    - name: nginx
      image: nginx
  nodeSelector:
    disktype: ssd
```

```
...
spec:
  containers:
    - name: nginx
      image: nginx
  affinity:
    nodeAffinity:
      requiredDuringSchedulingIgnoredDuringExecution:
        nodeSelectorTerms:
          - matchExpressions:
              - key: disktype
                operator: In
                values:
                  - ssd
```

Node Affinity

- `requiredDuringSchedulingIgnoredDuringExecution` :
- Phải đáp ứng điều kiện lúc tạo Pods.
 - Còn được gọi là Hard Affinity.
 - `IgnoredDuringExecution` - Pods sẽ vẫn chạy nếu các nhãn trên một node thay đổi và các quy tắc về mối quan hệ không còn được đáp ứng.

Node Affinity

- `preferredDuringSchedulingIgnoredDuringExecution`:
- Lựa chọn node đáp ứng điều kiện nhưng không bắt buộc.
 - Còn được gọi là Soft Affinity.
 - `IgnoredDuringExecution` - Pods sẽ vẫn chạy nếu các nhãn trên một node thay đổi và các quy tắc về mối quan hệ không còn được đáp ứng.

Node Anti-Affinity

```
...
spec:
  containers:
    - name: nginx
      image: nginx
  affinity:
    nodeAffinity:
      requiredDuringSchedulingIgnoredDuringExecution:
        nodeSelectorTerms:
          - matchExpressions:
              - key: disktype
                operator: In
              values:
                - ssd
```

```
...
spec:
  containers:
    - name: nginx
      image: nginx
  affinity:
    nodeAffinity:
      requiredDuringSchedulingIgnoredDuringExecution:
        nodeSelectorTerms:
          - matchExpressions:
              - key: disktype
                operator: NotIn
              values:
                - ssd
```


Labs



Lab 4: node affinity



Section 2:

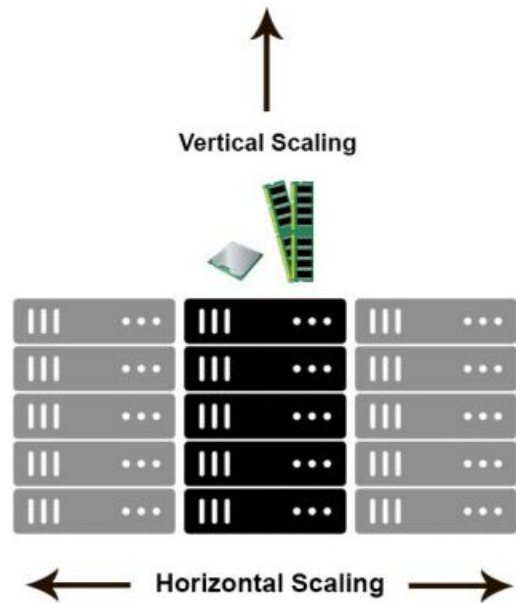
Deployments in K8s

Application Scaling

- App Scaling là gì?
- Stateless Application
- Statefull Application
- Labs

App Scaling

- **Application scalability** là khả năng của một ứng dụng mở rộng kịp thời để có thể xử lý hiệu quả ngày càng nhiều yêu cầu mỗi phút (Requests Per Minute - RPM).
- Trong Kubernetes, user có thể scale Pods.
- Pods có thể scale Vertically hoặc Horizontally.



Stateless Application

- **Stateless** app là chương trình ứng dụng không lưu dữ liệu client để phục vụ cho lần tiếp theo khi client đó kết nối đến.
- Các Stateless application có thể được scale **Horizontally**.
- Pods mới có thể được tạo cho Stateless Application.

Stateful Application

- **Statefull** app là chương trình ứng dụng lưu dữ liệu client để phục vụ cho lần tiếp theo khi client đó kết nối đến. Dữ liệu được lưu gọi là trạng thái của ứng dụng.
- Dịch vụ Cơ sở dữ liệu (database) là stateful service, chúng có hệ thống tệp không thể chia thành nhiều Instance.
- Stateful Application có thể scale **Vertically**.

Scaling in K8s

- **ReplicationController** được sử dụng để quản lý việc Scaling ứng dụng.
- ReplicationController đảm bảo đúng số lượng bản sao của Pod đang chạy mọi lúc.
- ReplicationController đảm bảo rằng một pod hoặc một nhóm các pod luôn hoạt động và khả dụng.

Labs



Lab 5: scaling with replication controller

ReplicaSet in K8s

- ReplicaSet là gì?
- ReplicaSet vs ReplicationController
- ReplicaSet vs Bare Pods
- Labs

Replica-Set

- **ReplicaSet** phiên bản nâng cao của ReplicationController.
- Giống như ReplicationController, mục đích của ReplicaSet là duy trì một tập bản sao Pod chạy ổn định tại bất kỳ thời điểm nào.
- Sự khác biệt chính giữa ReplicaSet và ReplicationController là hỗ trợ selector.
- **Label Selectors** được sử dụng để xác định một tập hợp các đối tượng trong Kubernetes.

Replica-Set

- ReplicaSets cho phép sử dụng label selector theo một list.
- Các toán tử In, NotIn, Exists được sử dụng để match các nhãn đối tượng của K8s.

Replica-Set vs Controller

```
....  
spec:  
  replicas: 3  
  selector:  
    app: alipne-box  
  template:  
    metadata:
```

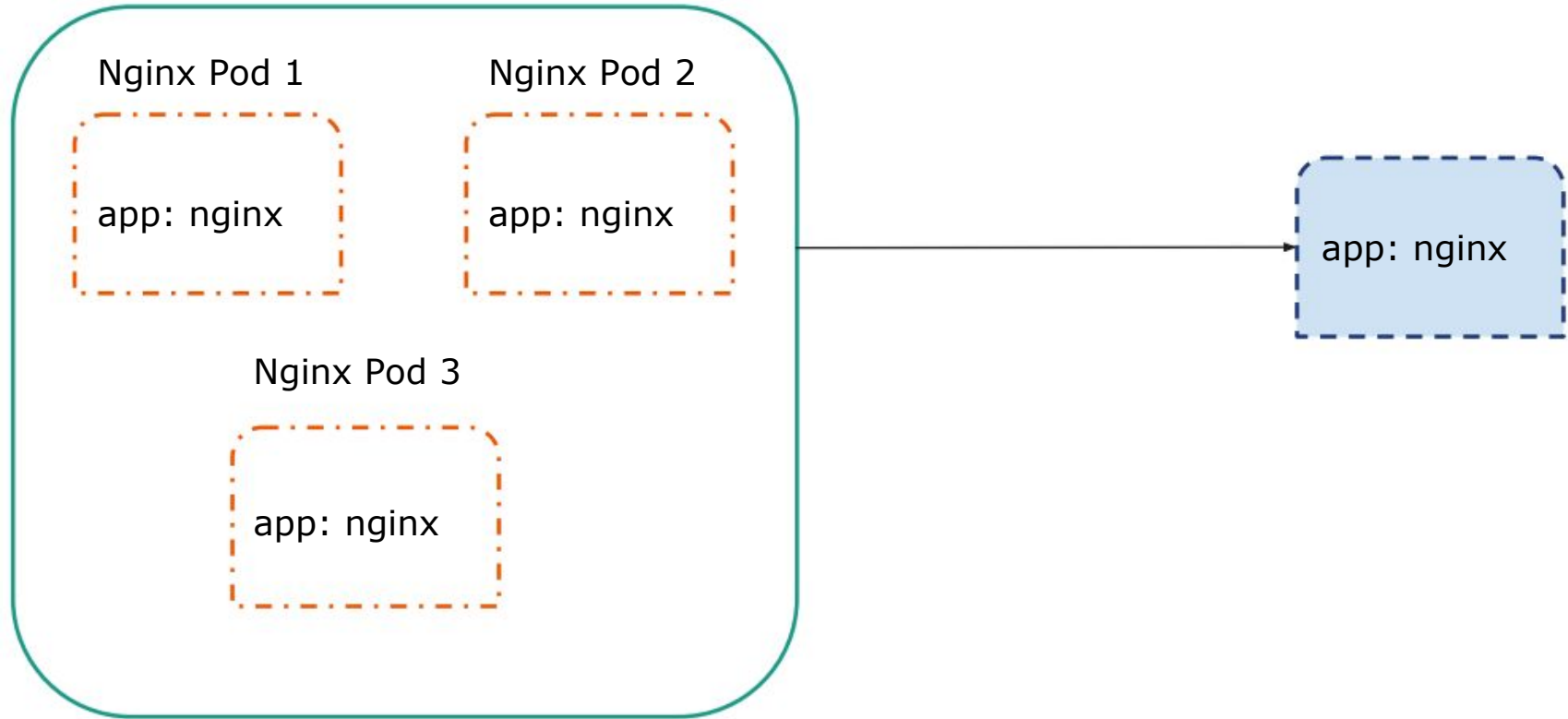
```
...  
spec:  
  replicas: 3  
  selector:  
    matchExpressions:  
      - {key: app, operator: In, values: [example, example, rs]}  
      - {key: teir, operator: NotIn, values: [production]}  
  template:  
    metadata:
```

Bare Pods & ReplicaSet

- Khi đã tạo các Bare Pods, các Bare Pods không nên có nhãn match với selector của một trong các Replicaset của bạn.
- ReplicaSet không giới hạn các Pods được chỉ định trong template của nó - nó có thể yêu cầu các Pods khác có Labels thỏa mãn.

Bare Pods & ReplicaSet

ReplicaSet



Labs



Lab 6: replicaset

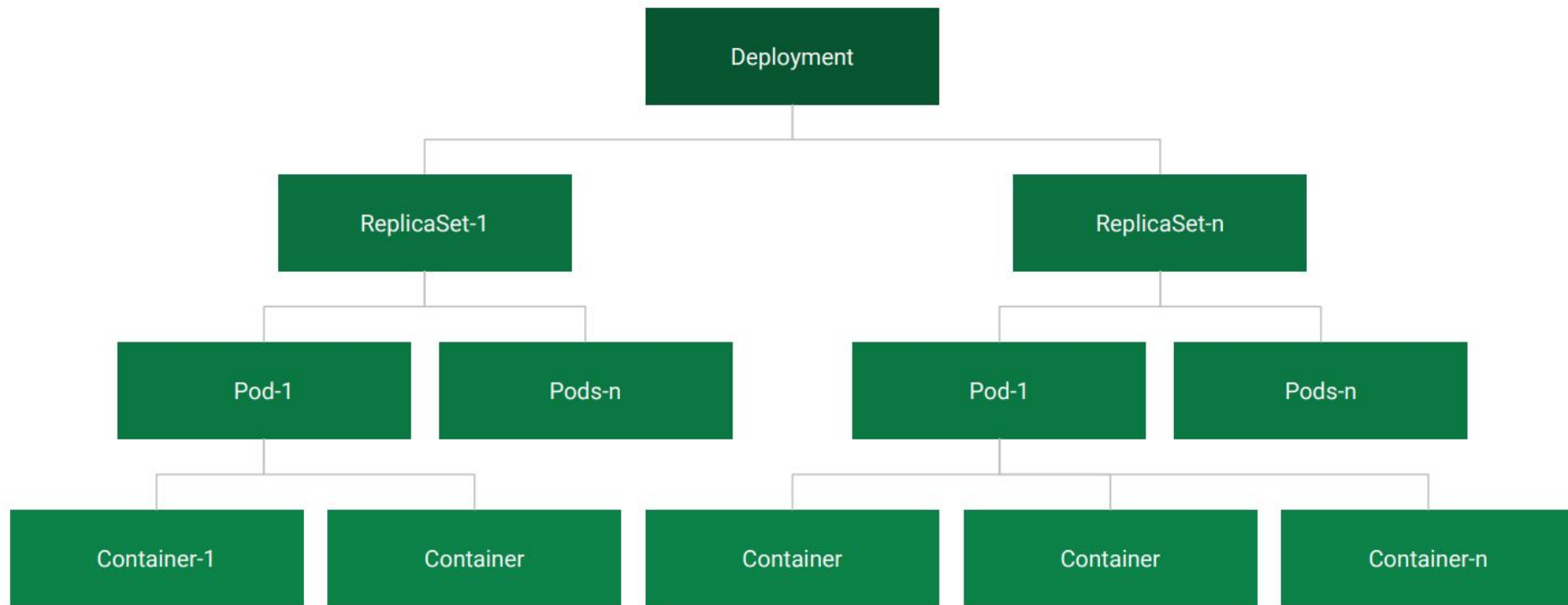
Deployments in K8s

- Deployments là gì?
- Desired state
- Sử dụng Deployments
- Labs

Deployments

- **Deployment** là đối tượng ở mức cao hơn ReplicaSet.
- Deployment là **desired State** của ReplicaSet.
- Deployment điều khiển cả Pods và ReplicaSets theo thông tin đã khai báo.

Deployments

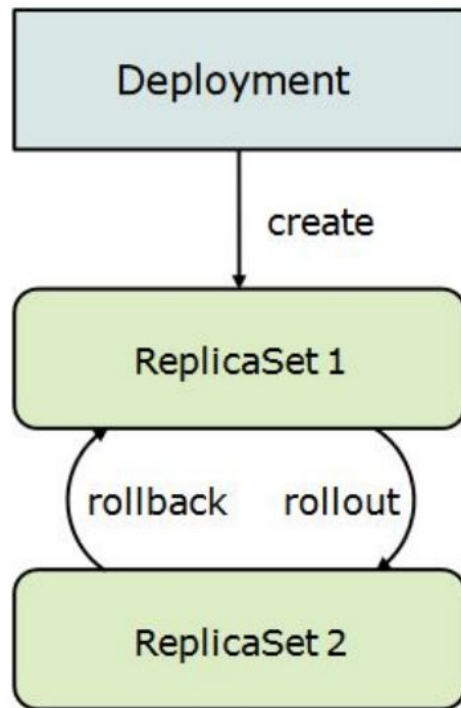


Use Case of Deployment

- **Create Deployment**: Triển khai các Pods ứng dụng.
- **Update Deployment**: Thay thế phiên bản mới của ứng dụng có kiểm soát.
- **Rolling Upgrade**: Nâng cấp ứng dụng không bị Downtime.
- **Rollback**: Phục hồi nâng cấp trong trường hợp bản nâng cấp không ổn định. Sửa đổi trạng thái của Deployment.

Use Case of Deployment

- **Pause/Resume Deployment:** Triển khai một tỷ lệ phần trăm nhất định.
- **Deployment RollOut Upgrade and Rollback Update.**



Labs



Lab 7: deployments



Section 3:

Summary

Mục tiêu

Kết thúc bài học, học viên cần đạt được các kỹ năng sau:

- Có thể hiểu được cơ chế lập lịch Pod trong Kubernetes.
- Hiểu và triển khai được ứng dụng dưới dạng daemonset, static pod.
- Có thể sử dụng node affinity để cho phép ứng dụng chạy trên node chỉ định.
- Hiểu được cơ chế scaling ứng dụng trong K8s, Replication controller làm gì và replicaset dùng như thế nào.
- Có khả năng triển khai ứng dụng dưới dạng deployments.

Tài liệu tham khảo

Thank you