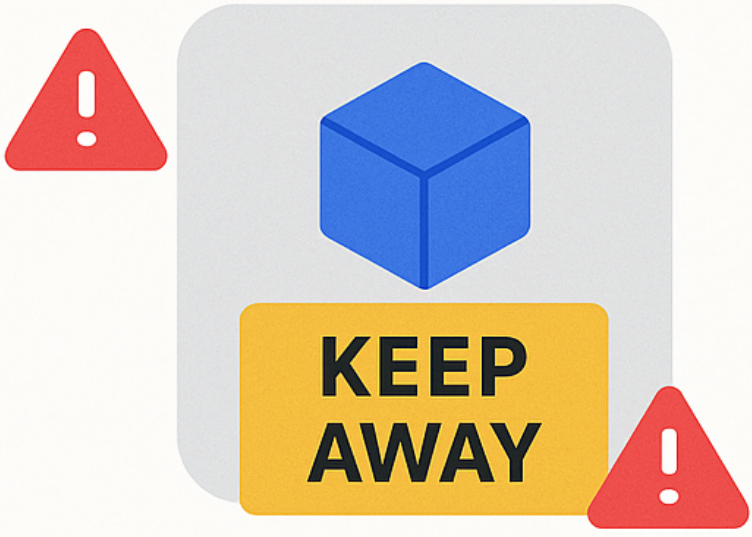
Mastering Kubernetes Taints & Tolerations

A Platform Engineer's Complete Guide

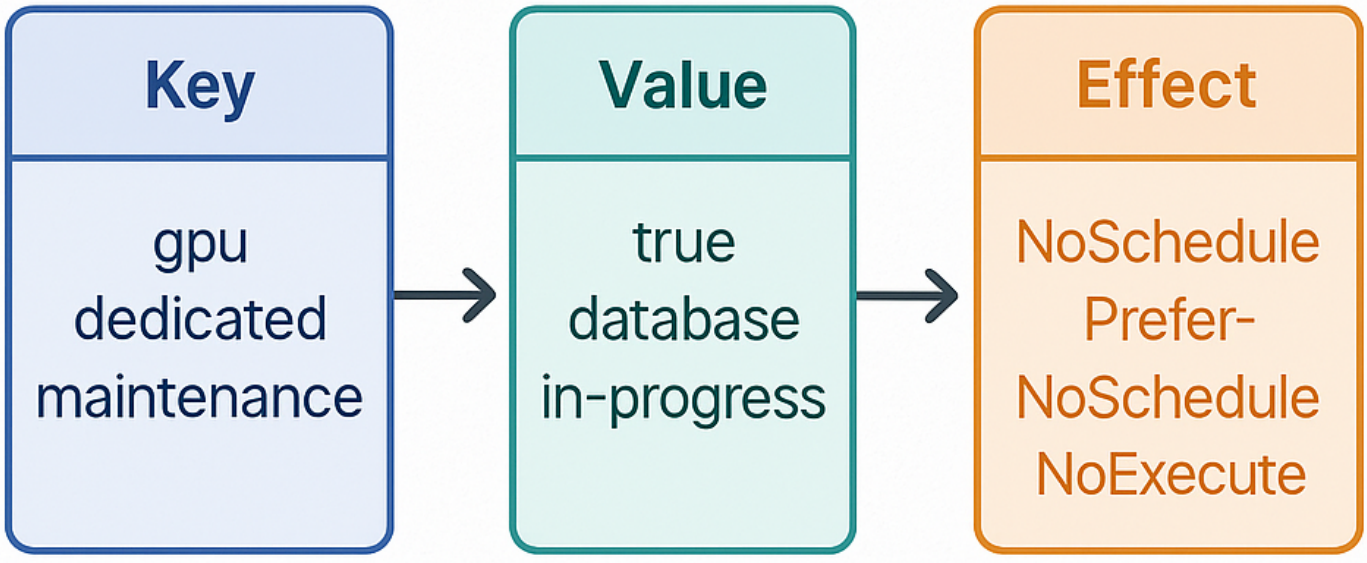
Swipe for insights →

What Are Taints?



- Repel pods from specific nodes
- Label-like properties applied to nodes
- Signal to scheduler about node restrictions

Taint Anatomy 🔍



Taint Effects Explained

NoSchedule

Won’t schedule new pods (existing stay)

PreferNoSchedule

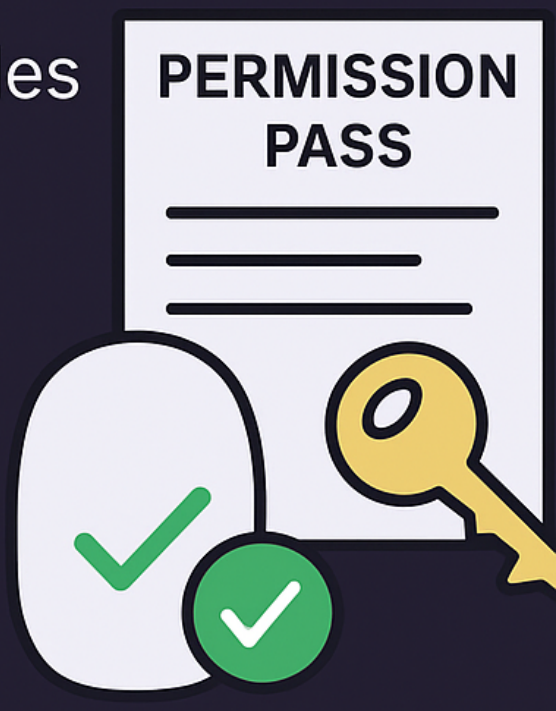
Tries to avoid scheduling (soft rule)

NoExecute

Evicts existing + blocks new pods

What Are Tolerations? ✅

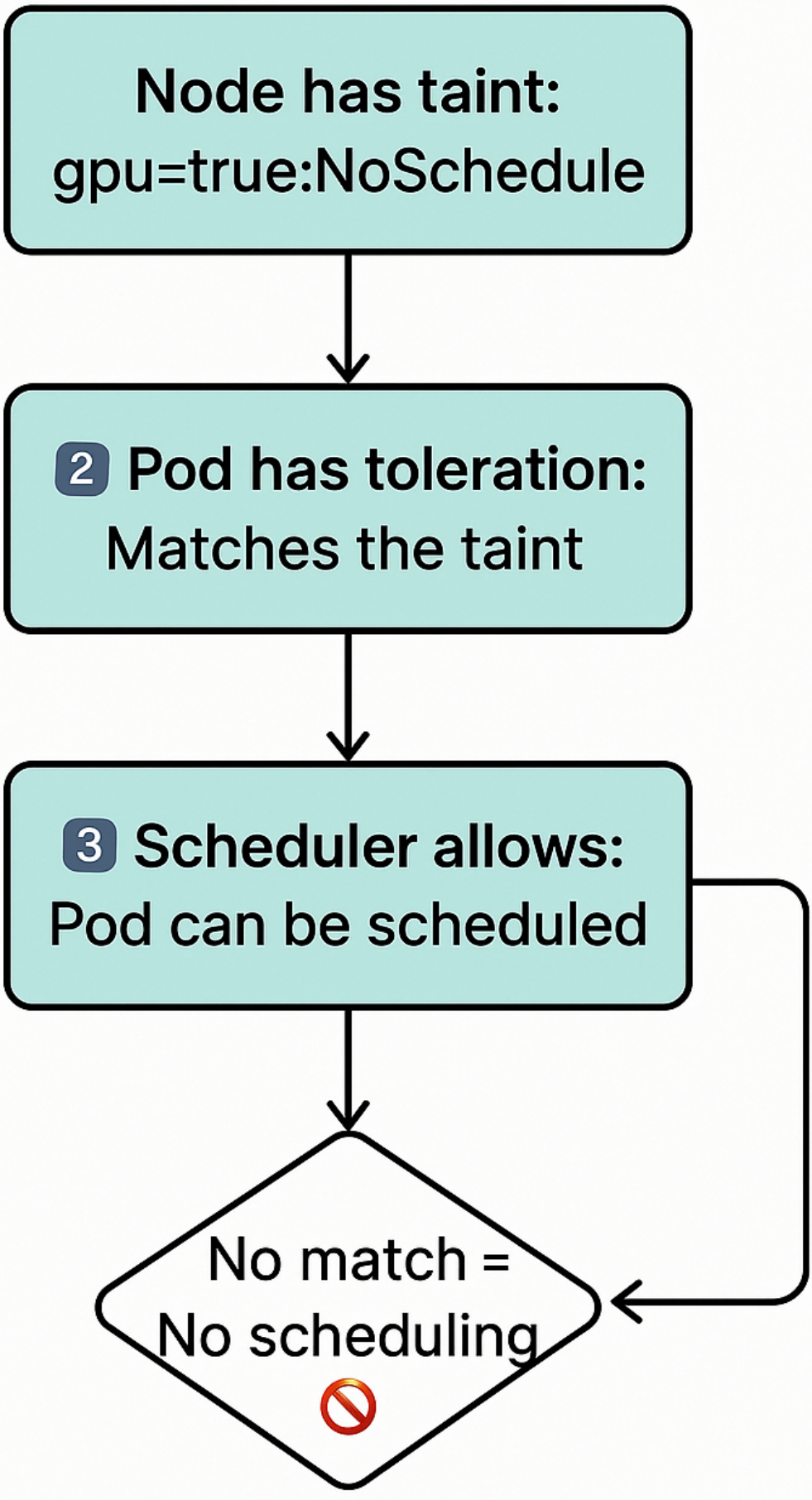
- Allow pods on tainted nodes
- Defined in PodSpec
- Permission slip to bypass taints
- Must match taint properties



Toleration Structure 📋

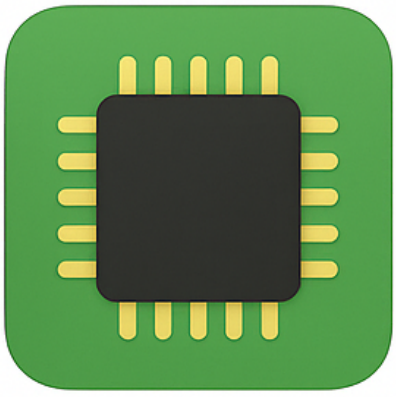
```yaml
tolerations:
 - key: "gpu"
   operator: "Equal"
          # or "Exists
   value: "true
   effect: "NoSchedule
   tolerationSeconds:300
```

The Magic Happens Here ✨



Dedicate GPU Nodes

BEFORE



Pod

Pod

ML/AI Pod

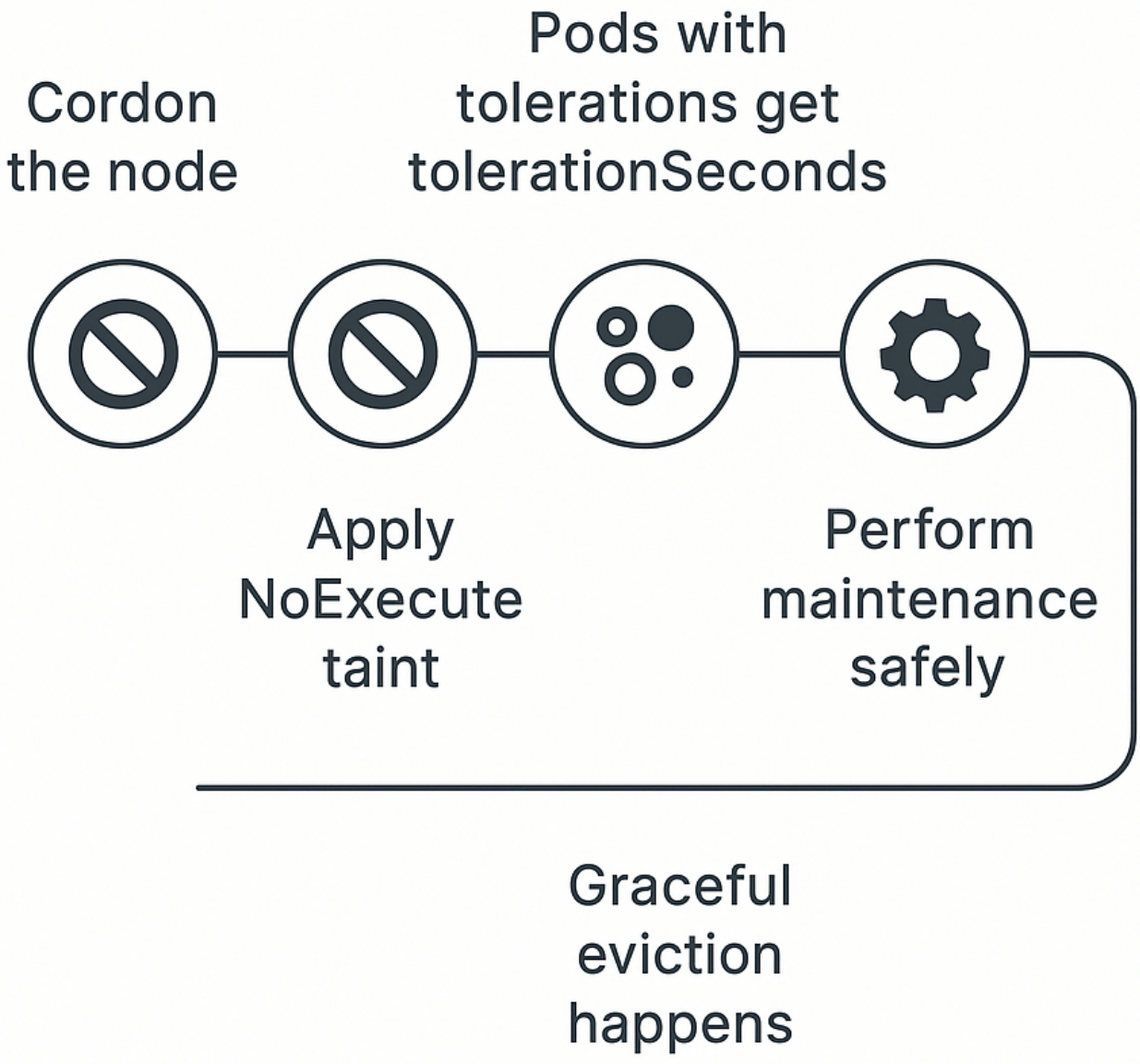
AFTER

Problem:
GPU nodes used by regular pods

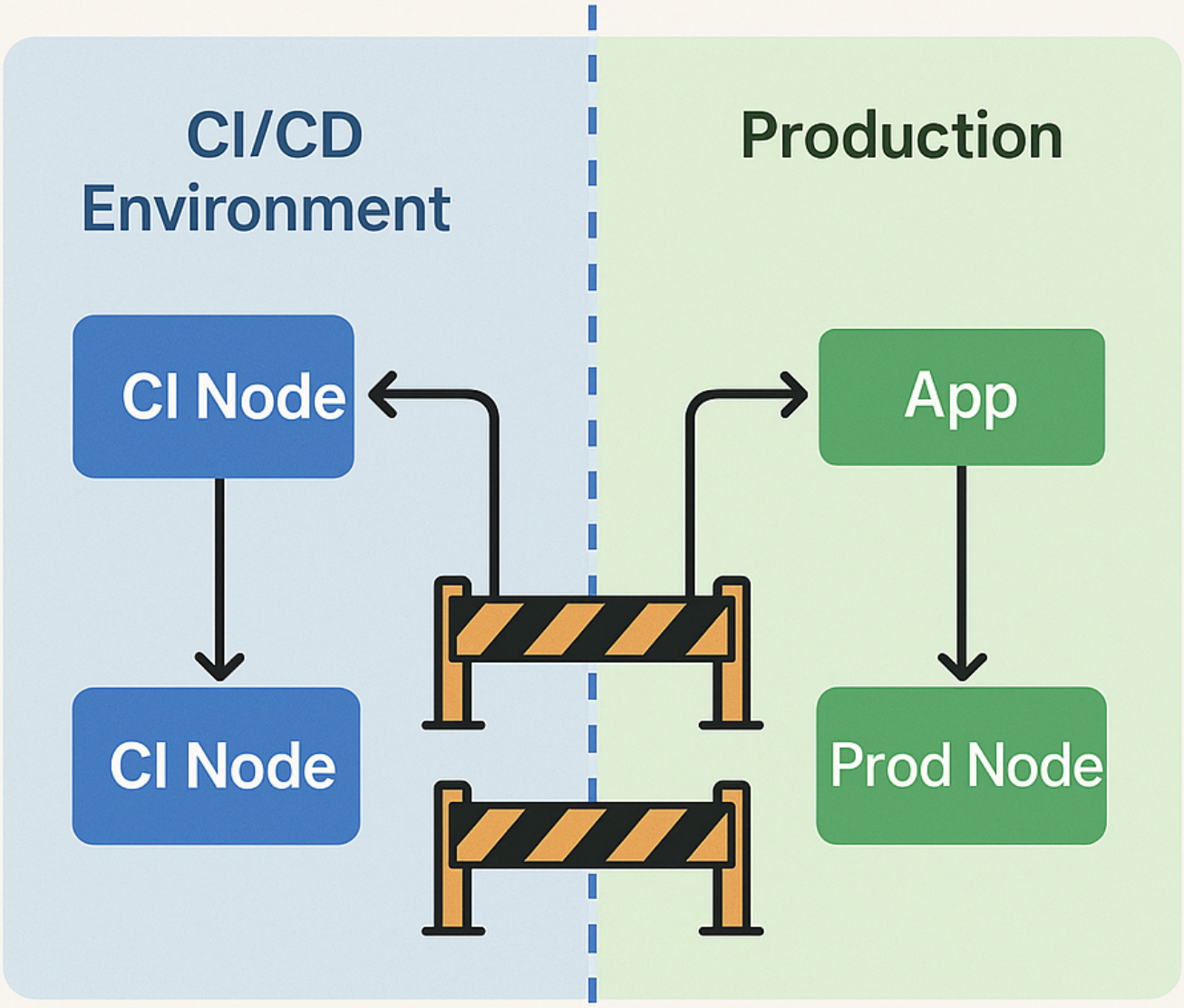
Solution:
Taint GPU nodes: gpu=true:NoSchedule
Add tolerations to ML/AI pods

Result:
GPU reserved for intended workloads

Graceful Node Maintenance



Isolate CI/CD Workloads 🔄



Why: Unpredictable resource usage

How: Taint CI nodes: workload=cicd: NoSchedule

CI runners have matching tolerations

Benefit: Production workloads stay safe

Essentiial Commands 💻

Þ Apply taint

```
kubectl taint ndes node-01
key=value:Effect
```

Þ Remove taint

```
kubectl taint ndes node-01
key=value:Effect-
```

Þ View taints

```
kubectl describe node node-01
| grep Taints
```

NoExecute + tolerationSeconds ⏱️

When NoExecute taint is added:

Pods without tolerations → Immediate eviction

Pods with tolerations → Stay indefinitely

With tolerationSeconds → Graceful period

@jzmath

Control Plane Protection👑

Default taint:

```
node-role.kubernetes.io/control-plane:NoShedule
```

Purpose: Reserve resources for cluster management

Exception: System pods have tolerations

Best Practices ✅

- Clear naming conventions
- NoSchedule for dedication
- NoExecute for eviction
- Combine with node affinity
- Manage as code (GitOps)
- Regular auditing

Avoid These Mistakes ⚠️

- ❌ Broad tolerations everywhere
- ❌ Forgetting tolerationSeconds
- ❌ Mismatched key/value pairs
- ❌ No documentation
- ❌ Testing only in production

Troubleshooting 101 🔍

Pod stuck in Pending?

- Check node taints
- Verify pod tolerations
- Review scheduler events
- Confirm resource availability

Advanced Patterns 🛠️

- Empty key + Exists operator
- Empty effect matching
- Multiple taint combinations
- Custom node conditions
- Webhook-based tainting

Production Example 🏭

```
# GPU Node Taint
kubectl taint nodes gpu-node-01
 nvidia.com/gpu=tesla-v100:
NoSchedule

# ML Pod Toleration
tolerations:
  key: "nvidia-com/gpu"
  value: "tesla-v100"
  effect: "NoSchedule
```

Key Takeaways 📝

1. Taints repel, tolerations allow
2. Three effects: NoSchedule, PreferNoSchedule, NoExecute
3. Perfect for: Dedication, maintenance, isolation
4. Combine with other scheduling features
5. Test thoroughly before production

Want to Learn More?🚀

📖 **Full guide with YAML examples:**
https://medium.com/@salwan.mohamed/a-comprehensive-guide-to-kubernetes-taints-and-tolerations-58af8659e92a

Questions?
Drop them below!

Comment

Repost if this
helped you

Repost

Tag a colleague
who needs this

Tag

@SalwanMohamed