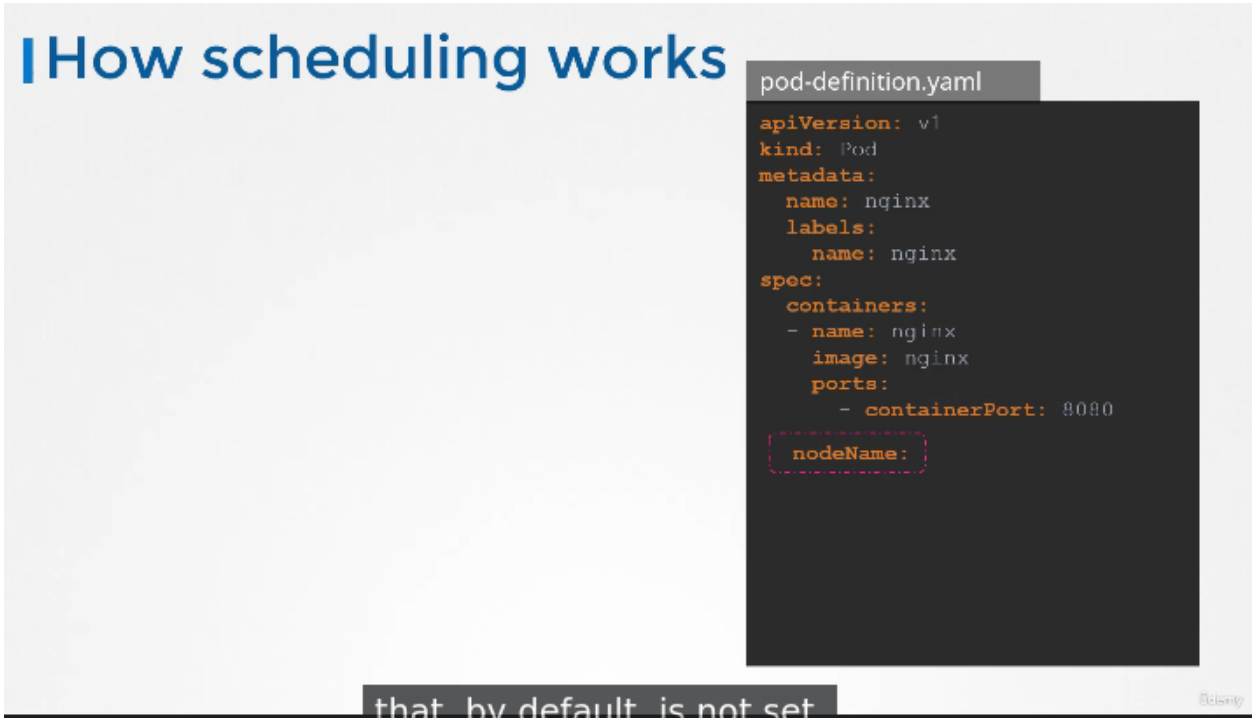
Manual scheduling

Every pod has a field nodeName that is not default set

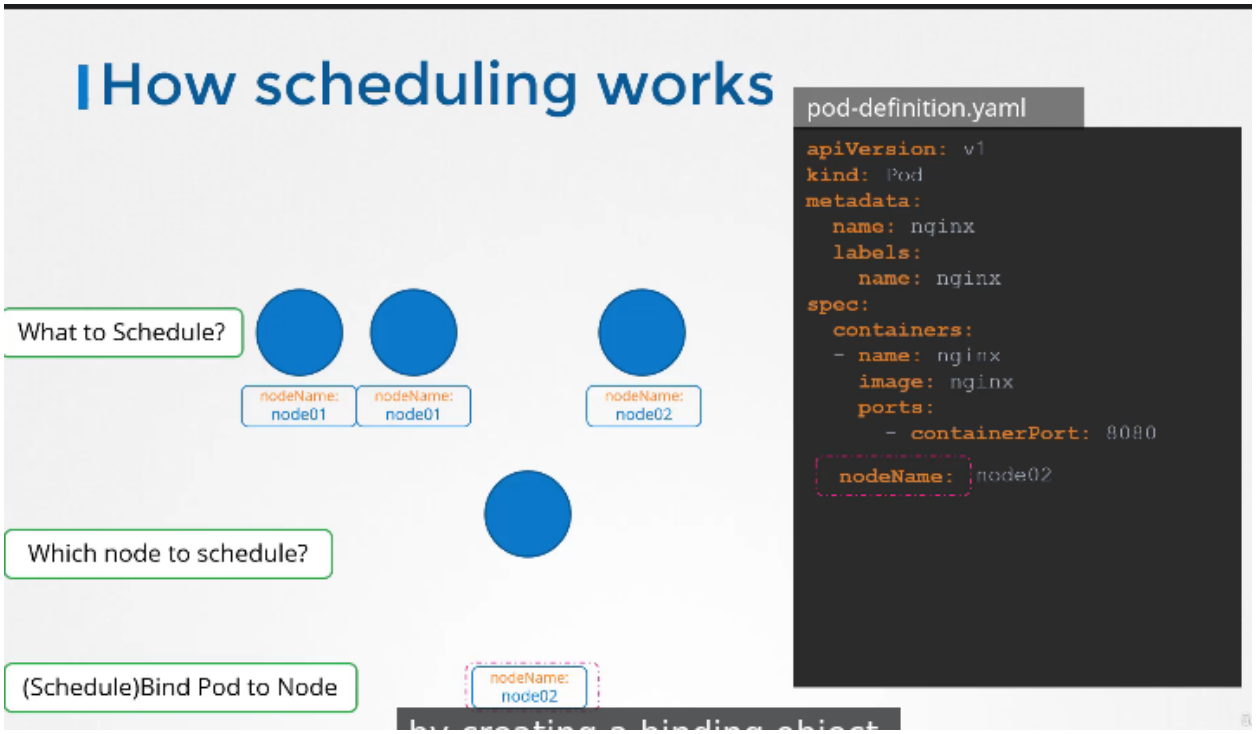


Schedular goes through all pods and scan all pods which not have
Thewse field that is nodeName
And these are the candidate for scheduling

Then it find the right node for the pod ny running scheduling algotirh

Once identified it scheduled the pod on the node
By setting the node name property to the name of the node

By creating a binding object

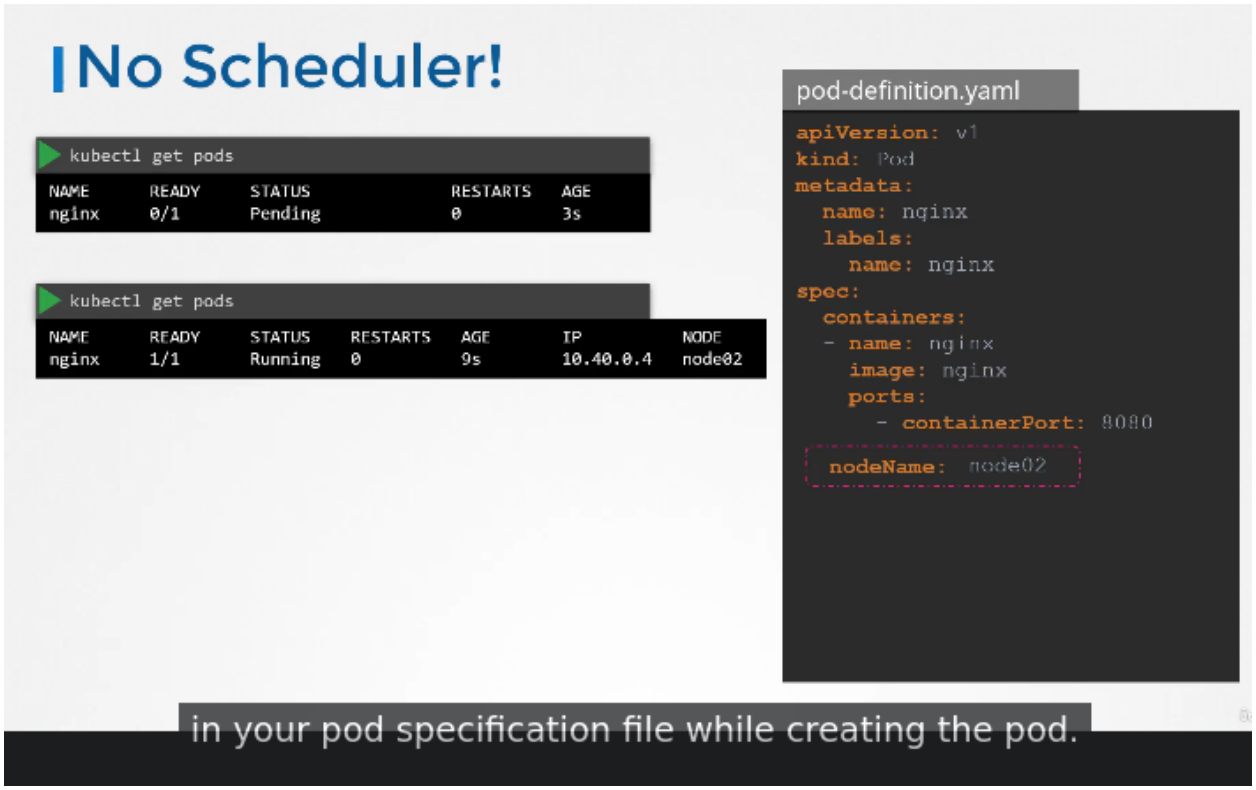


So if there is no scheduler to monitor anf schedule node
Then pod contains to be in a pending state

Then in this case what we can do

We can manually assign pods to nodes yourself

Without the scdeduler the easiest way to schedule the pod is to set the
Node name to the name of the node



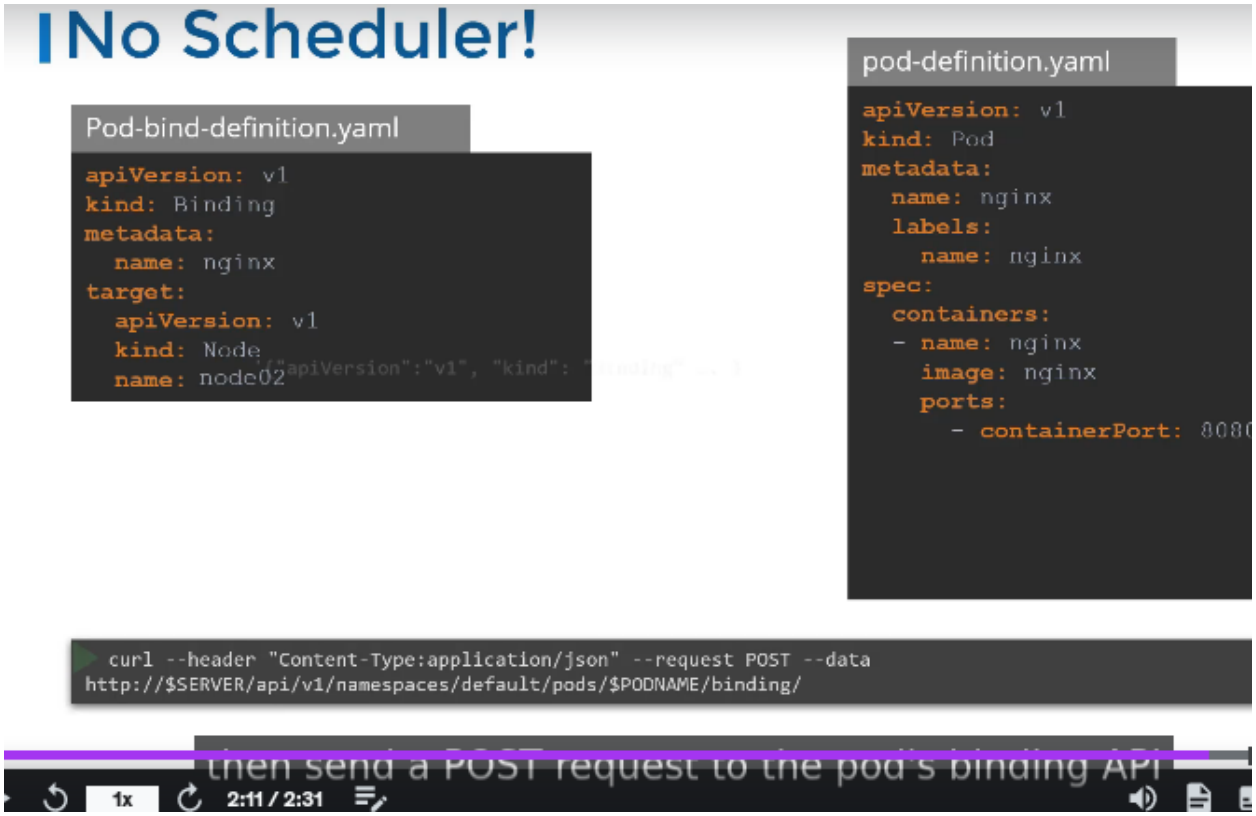
So that pod is assigned to specified node

If pod is already created

K8s does not allow to modify nodeName property so how to assign node to a existing pod

Create a binding object and send a POST request to the pod's binding API
Thus in that way what the actual schedulwer does
In binding object we specified a target node as node of the name
And then send a POST request to the pod's binding API
With the data set to the binding object in a JSON format

apiVersion: v1
Kind: Binding
Metadata:
Name: nginx
Target:
apiVersion: v1
Kind : Node
Name : name-of-node



Send a POST request to POD binding API

```
curl --header "Content-Type:application/json" --request POST --data '{"apiVersion":"v1", "kind": "Binding" …. }
http://$SERVER/api/v1/namespaces/default/pods/$PODNAME/binding/
```

Curl –header “Content-Type:application/json” –request POST —data ‘{json format of binding object”}’ https://$SERVER/api/v1/namesapces/default/pods/4PODNAME/binding/

Kubectl replace –force -f yaml-file.yaml

We can’t move pod from one node to another or system to system we
First delete the pod and then move the pod

Labels and Selector

Labels means group things together and selector fulter things sbased on our needs
Labesla == properties attached to each item

Selector filter thse items based on label
Every object has level

Objec t has levels
And selector fuilter these object based on labels

Selector == condition

Labels:
Key1: value1

2 ways

1 . kubectl get pods **–selector app=App1**

2 . in replica set

Selector:
matchLables
Key1: value1

We can specify many labels

On match rs created successfully

3 . for service

Selector:
Key1: value1

To group and select objects == labels and selector

Annotations

It is used to record other details for informatory purpose
Like name , version , contact details . phone numbers , email ids etc
Buildversion

That may be used for some kind of integration purpose

metadata:
Name:
labels:
annotations:
Buildversion: 1.34

Kubectl get pods –selector env=dev –no-headers | wc -l

Kubectl get all –selector env=dev –no-headers | wc -l
Kubectl get pods –selector env=dev,bu=finance –no-headers | wc -l

Taints and Tolerations

Pod to node relationship

How we can restrict what pods are placed on what nodes

Ex:

Bug approaching a person

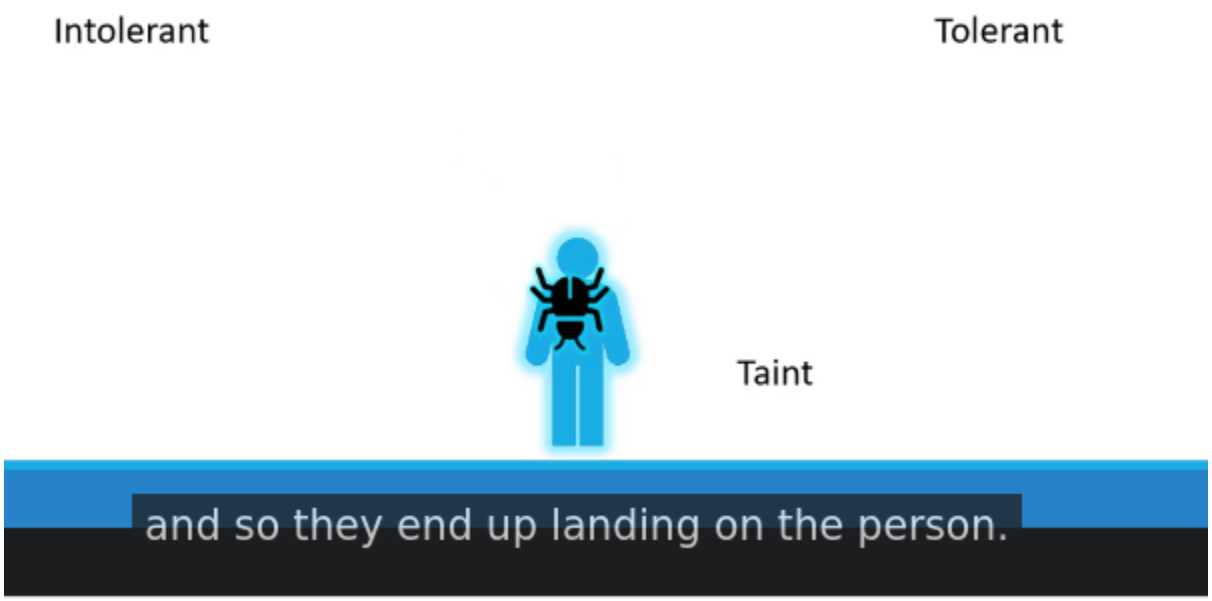
To prevent person from bug we need to spray the person with a repellent spray

Or taint

So that bug is intolerant to the smell so when bug approaches the person the taint throws the bug away

Bug ==1 intolerant to the smell or taint
Or may be
2 tolerant to the smaell

But other bugs which are tolerant to this smell and so that bugs land on the person because taint do not do nothing



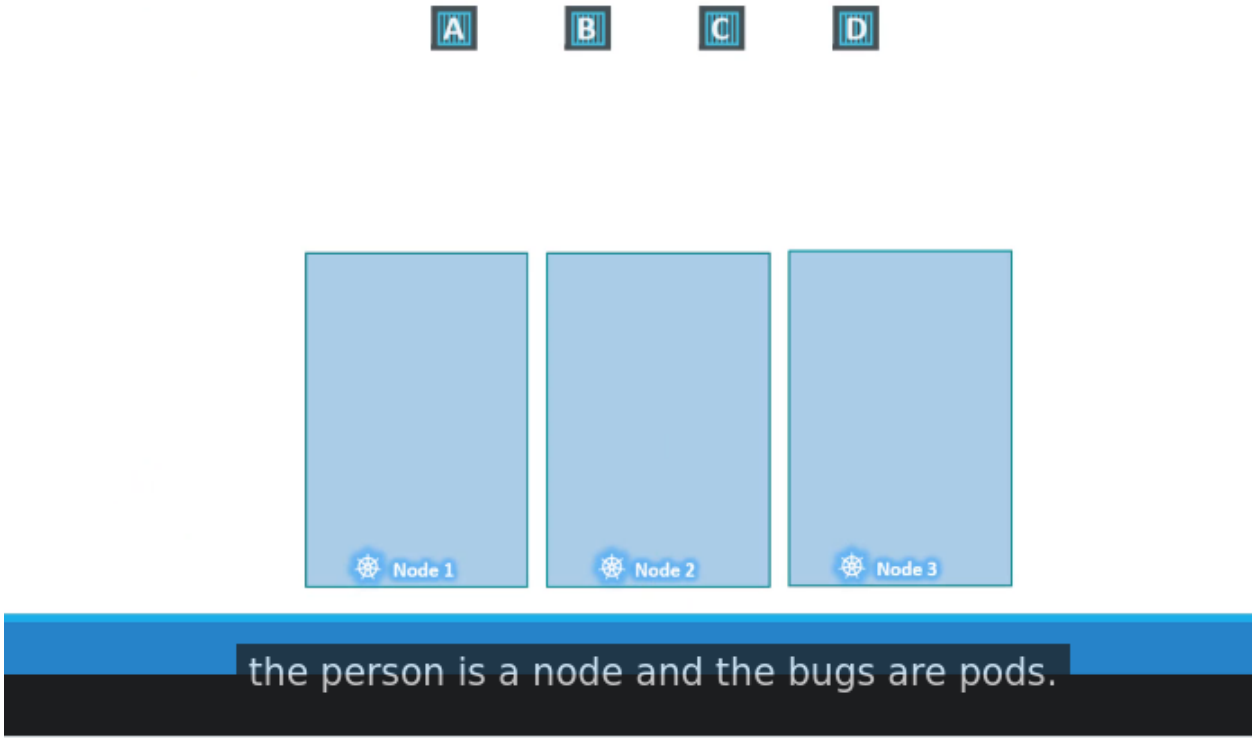
Bug land on the person or not

Taint on the person
And toleration level of the bug

In k8s

Person === node means we apply taint on the node

Bug === pod and we apply toleration on the pod



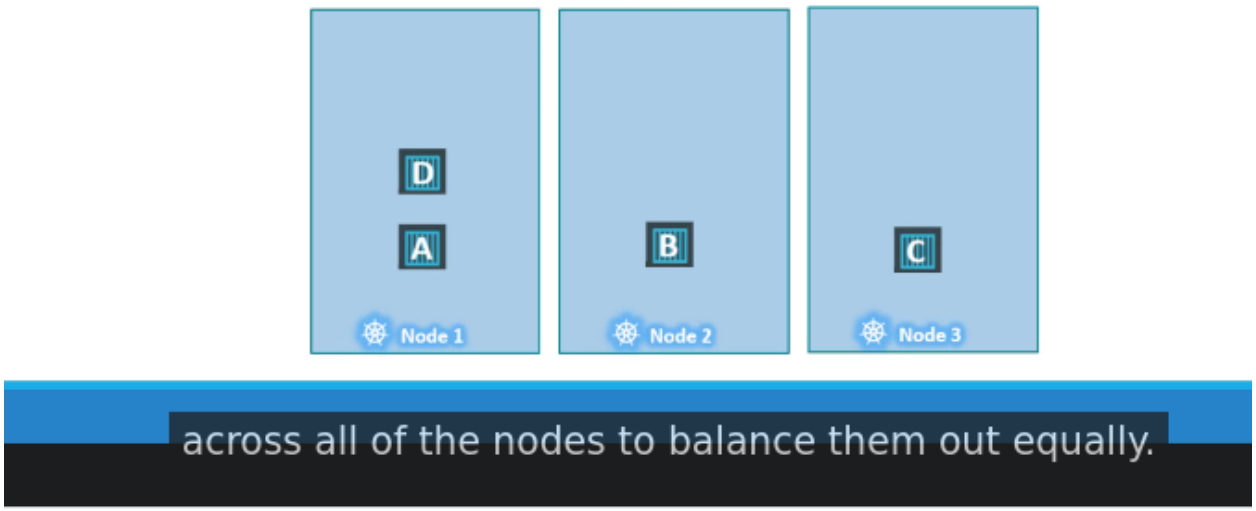
It does not lead any security or intrusion on the cluster

Taints and tolerations are used to set restrictions on
what pods can be scheduled on a pod

Pod == a

When pods are created k8s scheduler tires to place these pod on the avail node

As when there is no restriction then scheduler places the pods across all the nodes to balancer the nodes equally



nscript

But let us assume that there is a particular node that have dedicated resources for a particular application or use case then in this case how we can deploy

So we want to place only those pods on these nodes that belongs these applications

So we can do

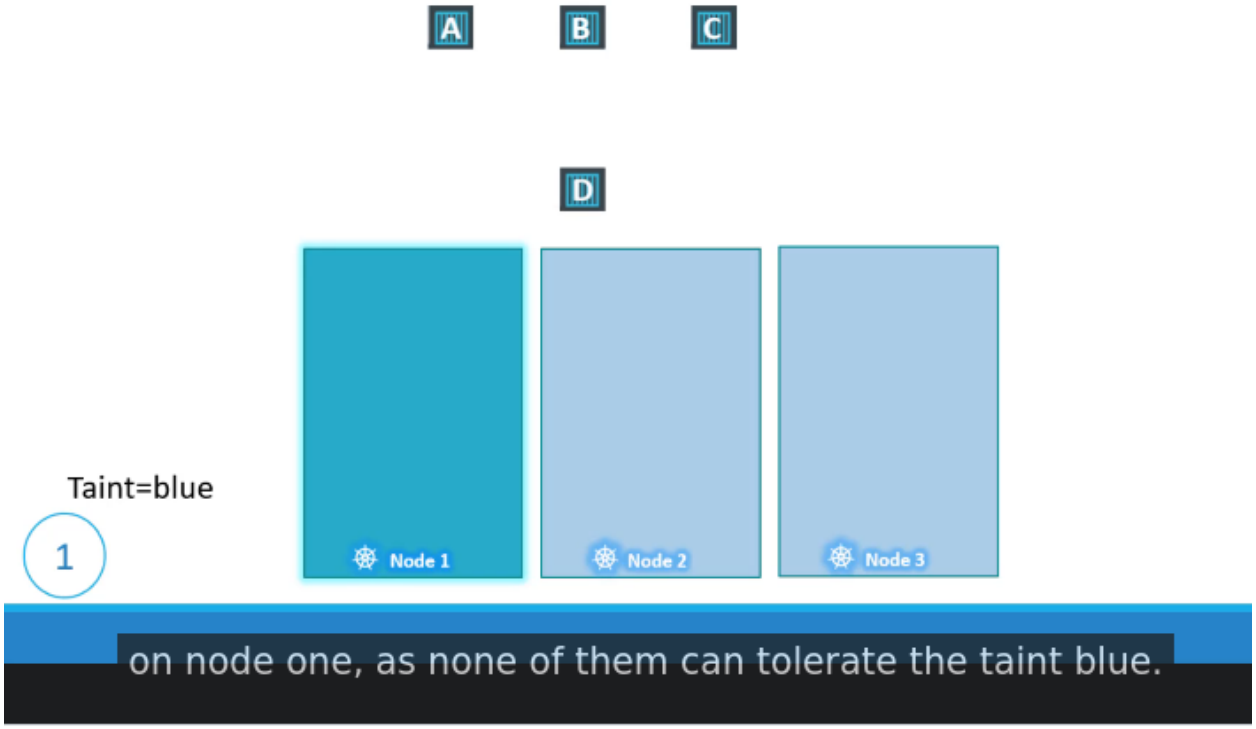
1, first prevent all pods from being placed on the node by placing a taint on the node
Means apply taint on the node

Taint and tolerant == field or property of object

Taint=blue

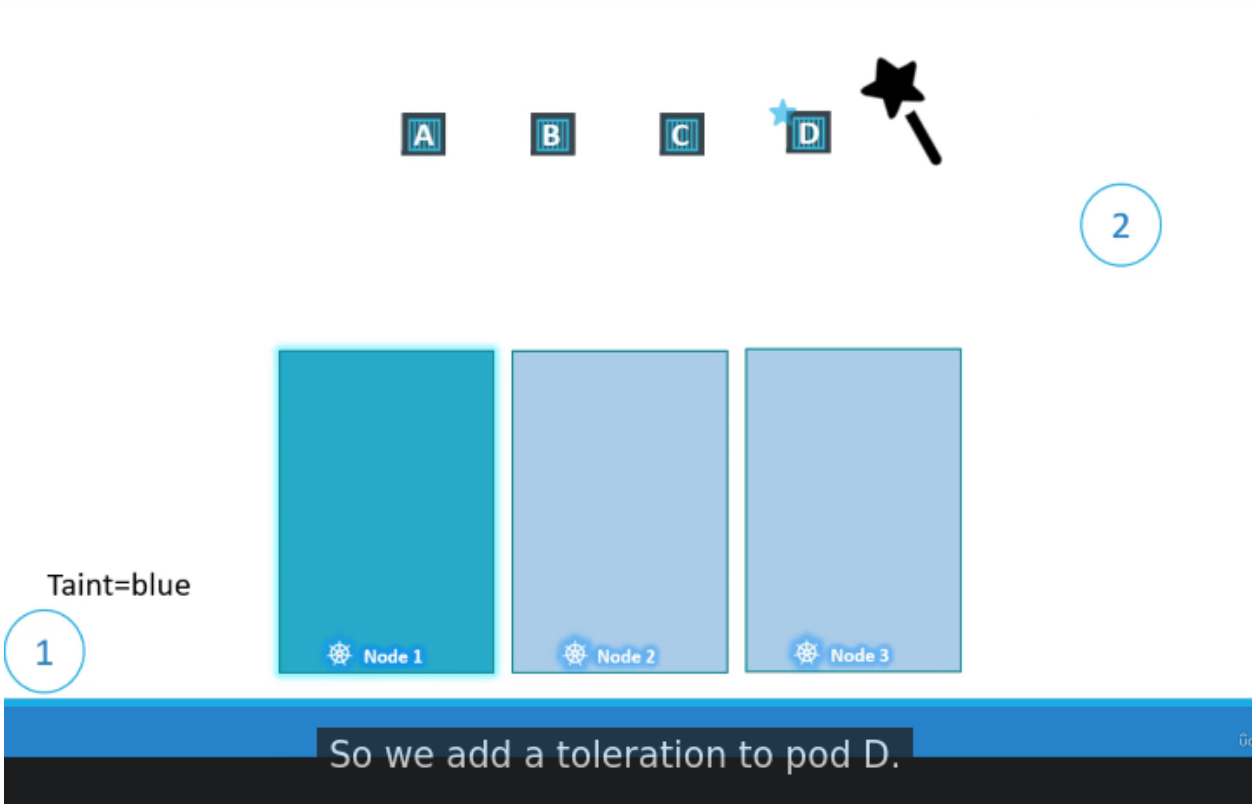
And by default pods have no tolerance means 0 tolerant level

Means none of the pods can tolerate any taint
So none of the pods can be placed on the taint node without having toleration power

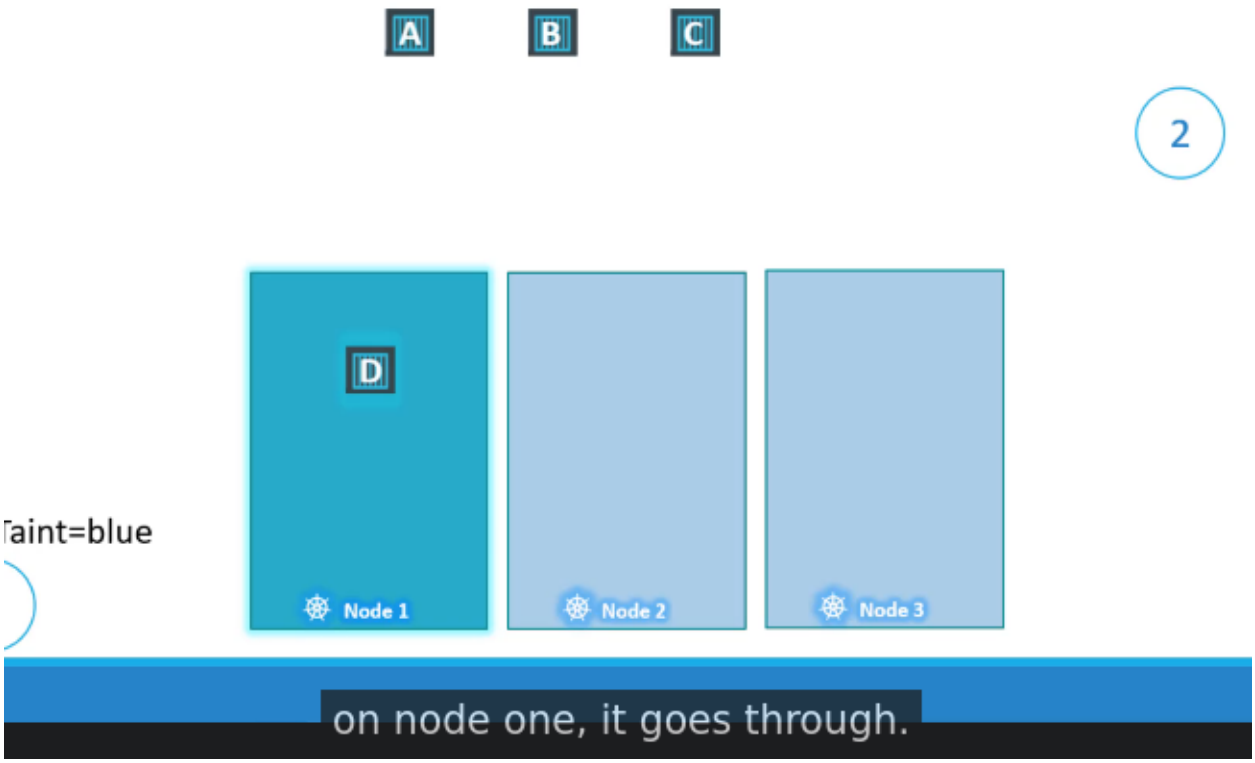


So to place node on the taint node we need to enable certain podsa
To be placed on this node
For this we need to specify which pods are tolerate to this taint nodepo
Pod = a,b,c,d and we want only pod D to placed on the taint node1

SO we need to add a toleration power to the pod D so that pod D is tolerate the node node 1



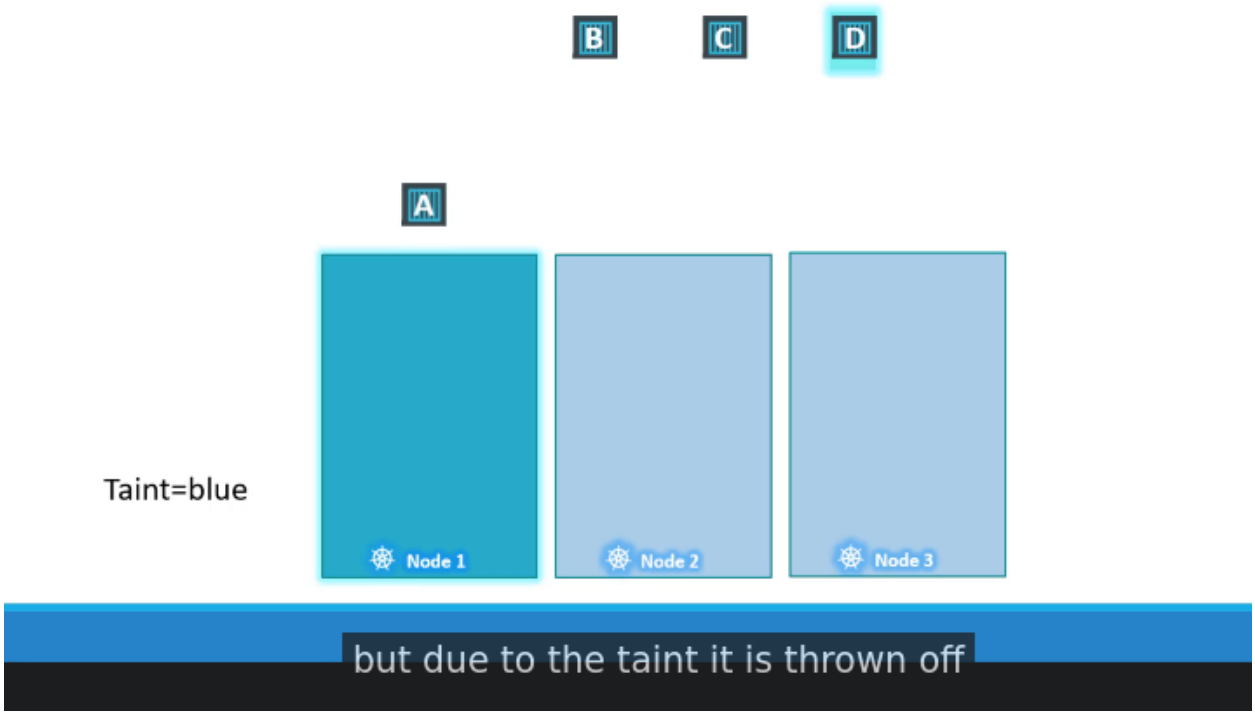
And so when scheduler tries to put pod D on node 1 it will now placed on node 1

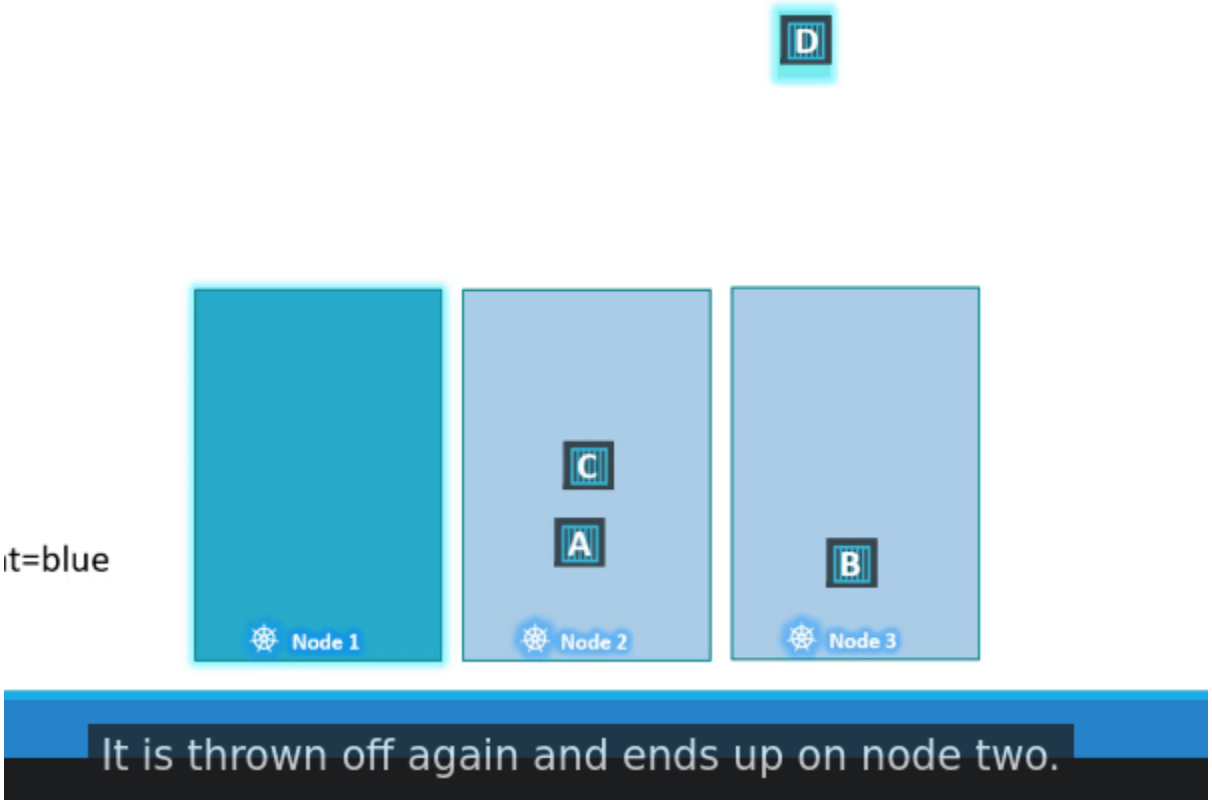


Node 1 only accept those pods which can tolerate taint blue

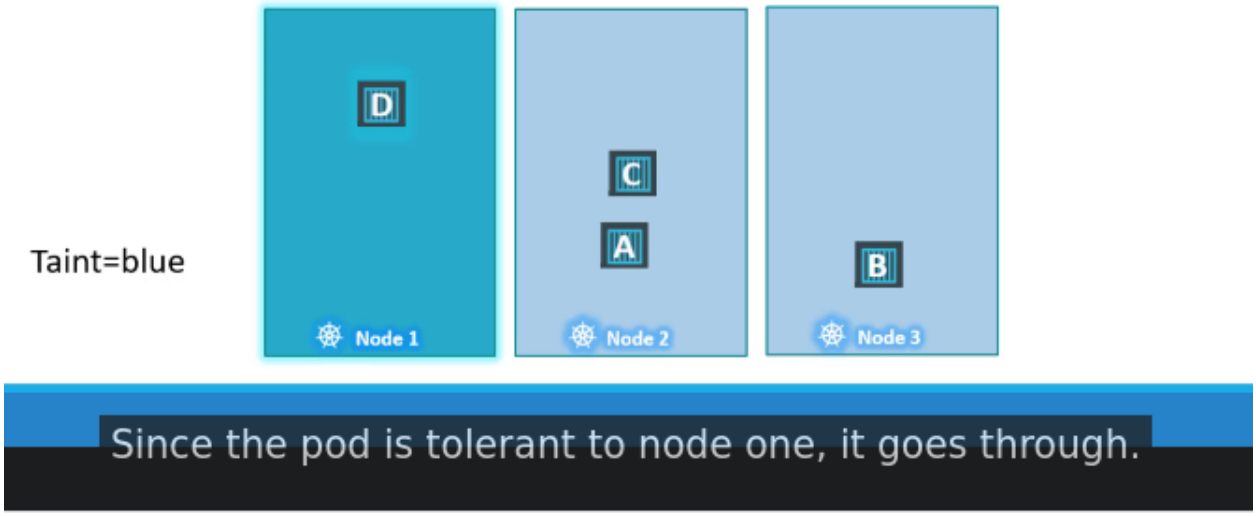
1 . scheduler tries to put pod A on node 1 but due to taint it will not placed
And thrown off and then pod A goes to node 2 and node 2 has no taint then pod A will placed on node 2
Same with pod B node 1 thrown off pod B and then it will go to node 3

Same with C


B
C
D
A
Taint=blue
Node 1
Node 2
Node 3
but due to the taint it is thrown off


D
t=blue
C
A
B
Node 1
Node 2
Node 3
It is thrown off again and ends up on node two.

2 . now pod D comes since the pod is tolerate to the taint node 1 , scheduler put the pod on node 1



Taint = node
Toleration = pod

Pod which have power to tolerate taint on taint node can be placed on that node

By default pod have no toleration

Create

Kubectl taint node node-name key=value:taint-effect

Taint-effect

The taint-effect defines what would happen if a pod do not tolerate the taint

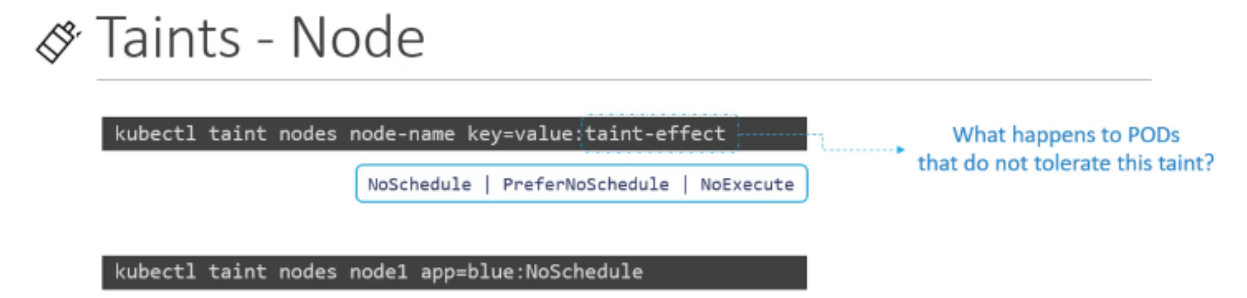
There are 3 taint-effect

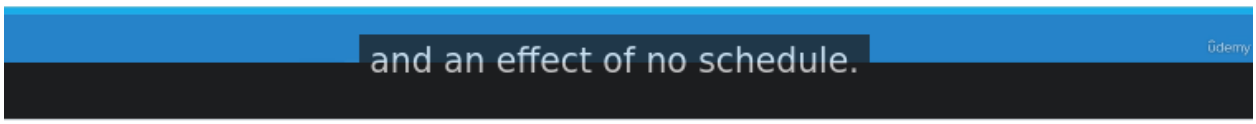
1. NoSchedule :
   Means pod will not be scheduled on the node

2 PreferNoSchedule == means system will try to avoid a pod on the node but
That is not guaranteed

3 . NoExecute ==
Means that new pods will not be scheduled on the node and existing pods on the node , if any will be evicted if they do not tolerate the taint
नए पॉड्स को नोड पर शेड्यूल नहीं किया जाएगा और नोड पर मौजूदा पॉड्स, यदि कोई हो तो बेदखल कर दिया जाएगा
नए पॉड्स को नोड पर शेड्यूल नहीं किया जाएगा और नोड पर मौजूदा पॉड्स, अगर कोई दागी को बर्दाश्त नहीं करता है तो उसे बेदखल कर दिया जाएगा

These pods may have been scheduled on the node before the taint was applied to the node.
ये पॉड्स रहे होंगे
दागी को नोड पर लागू करने से पहले नोड पर निर्धारित किया गया था।





Kubectl craete taint node1 app=blue:NoSchedule

TOLERATIONS ARE ADDED TO THE PODS

How to add toleration to a pod

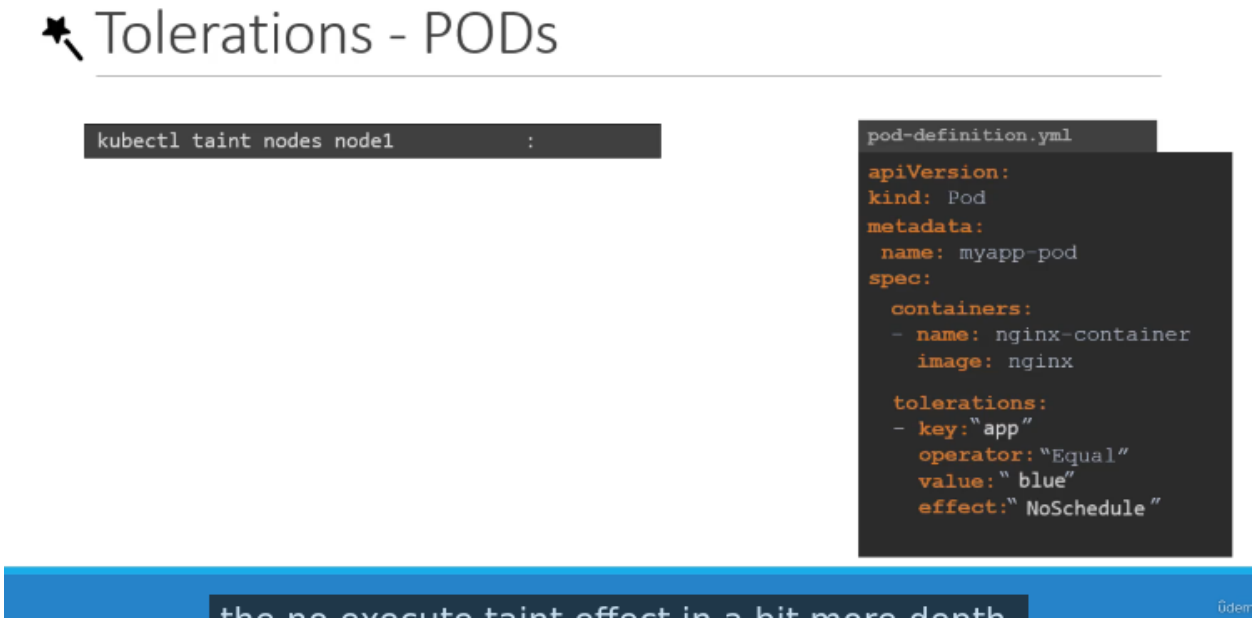
First open pod definiton file
And under spec: section of pod-definiton file

Add a sectiions called **tolerations**
And
Move the same values while craeting the taint in double quotes

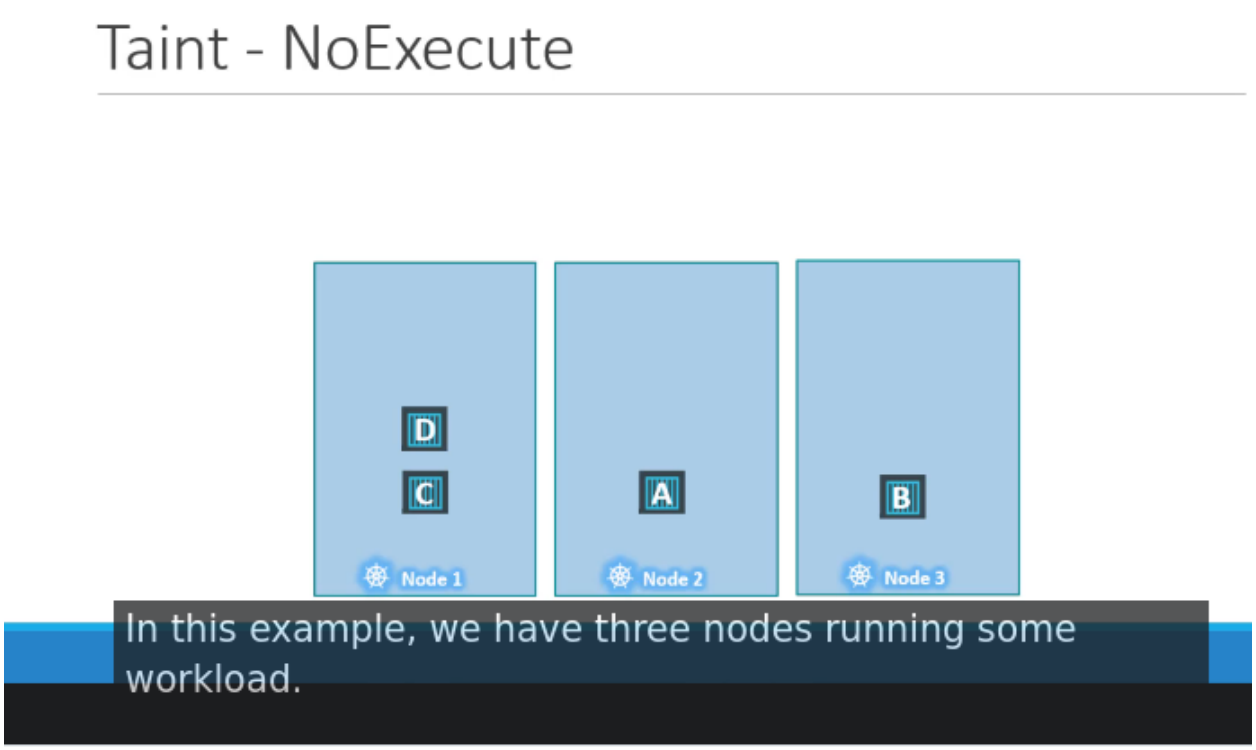
Spec:
  Tolerations:
    - Key: “app”
      Operator: “Equal”
      Value: “blue”
      Effect: “NoSchedule”

So now when we craete pod or updated with the new tolerations

Two things may happen
1 . Not scheduled on nodes (if first time craeting:
2 . evicting from the existing nodes .



NoExecute taint-effect

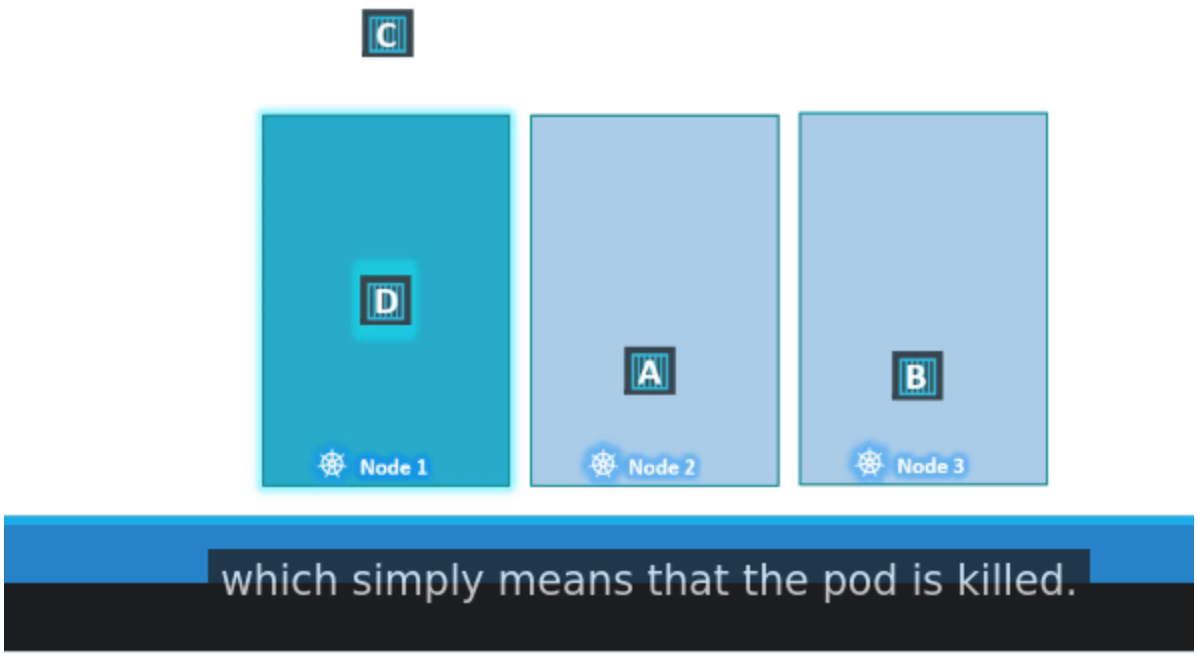


anscript

We do not have any taints and tolerations at this point

Now we taint node1 and tolerate some pods which belong to node 1

D = tolerate
And tainti-effect = NoExecute
And when applied pod c is evicted fro node 1 and it simply means kill



Taints and tolerations are meant to spoecify nodes to accept certain pods only

In this case node 1 only accept pod 1

But tolerate pod can be placed on the other node also

It only allows node to accept certain types of pods

But if we want to place pod on only specific node we can achieve through node affinity

Master nodes on the cluster

It is just another node
That has all the capabilituies of hosting a pod and runs all the management software

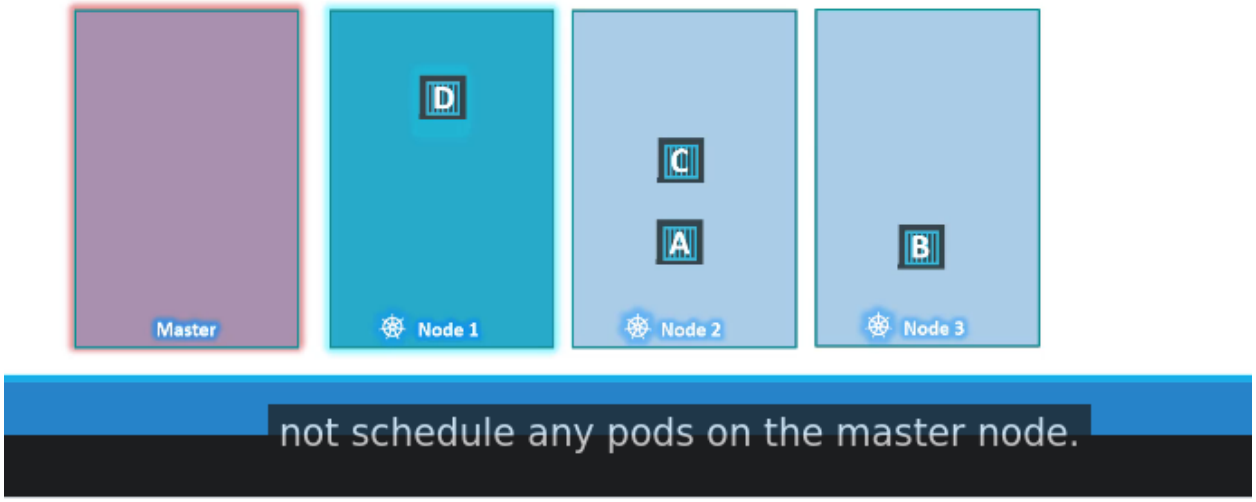
Scheduler does not schedule any pods on the master node

When the k8s cluster is first set up
A taint is set on the master node automatically that prevents any pods from being scheduled on this node

We can see this and also modify this behaviour
But best practice is to not deploy application workloads on a master server

Kubectl describe node kubemaster | grep Taint
To delete taint
kubectl taint node controlplane node-role.kubernetes.io/control-plane:NoSchedule-

```
kubectl describe node kubemaster | grep Taint
Taints:             node-role.kubernetes.io/master:NoSchedule
```



Nodes from accesoting certain pods

Node 1 only accept pod D
But it does not gurantee pod D will always be placed on node one

Id we want that pod should go to a certain pod not any random pod == wecan achieve through node affinity

Kubectl taint node node-name key=value:NoScheduler

```
status: {}

controlplane ~ ➜  kubectl run bee --image=nginx --dry-run=client -o yaml > bee.
yaml

controlplane ~ ➜
```

```
See 'kubectl taint -h' for help and examples

controlplane ~ ✖ kubectl taint node controlplane node-role.kubernetes.io/contro
l-plane:NoSchedule-
node/controlplane untainted

controlplane ~ ➜
```

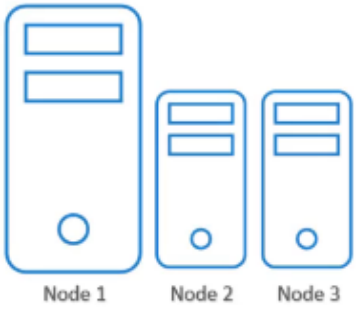
```
Use "kubectl options" for a list of global command-line options (applies t
 commands).

controlplane ~ ➜  kubectl taint nodes node01 spray=mortein:NoSchedule
node/node01 tainted
```

Node Selector

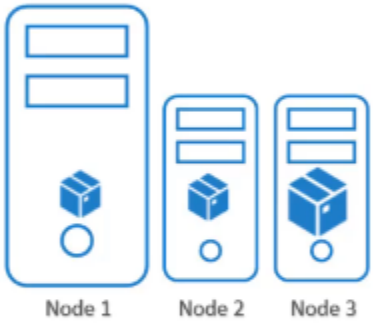
3 node cluster and 2 are smaller nodes with lower hardware resources one is larger node
With higher resources

And you have different kinds of workload sin your cluster

For ex ;
Data processing workloads that require higher horsepower to the larger node
Because that have the resources that will not go out of resources
In case the job demand extra resources
But by default setup any pod can go to any nodes



You have a three node cluster of which two are smaller nodes



may very well end up on nodes two or three,

In this case pod C which require large resources may be placed in node c which have less resources
So to avoid this we use node selector

We set limitation on the pod so that they only run on particular nodes
Two ways to do

1. Node Selector == SIMPLE AND easier method

For this we look into the pod definition file created earlier

User spec section specify the field nideSelector and attribute size as large

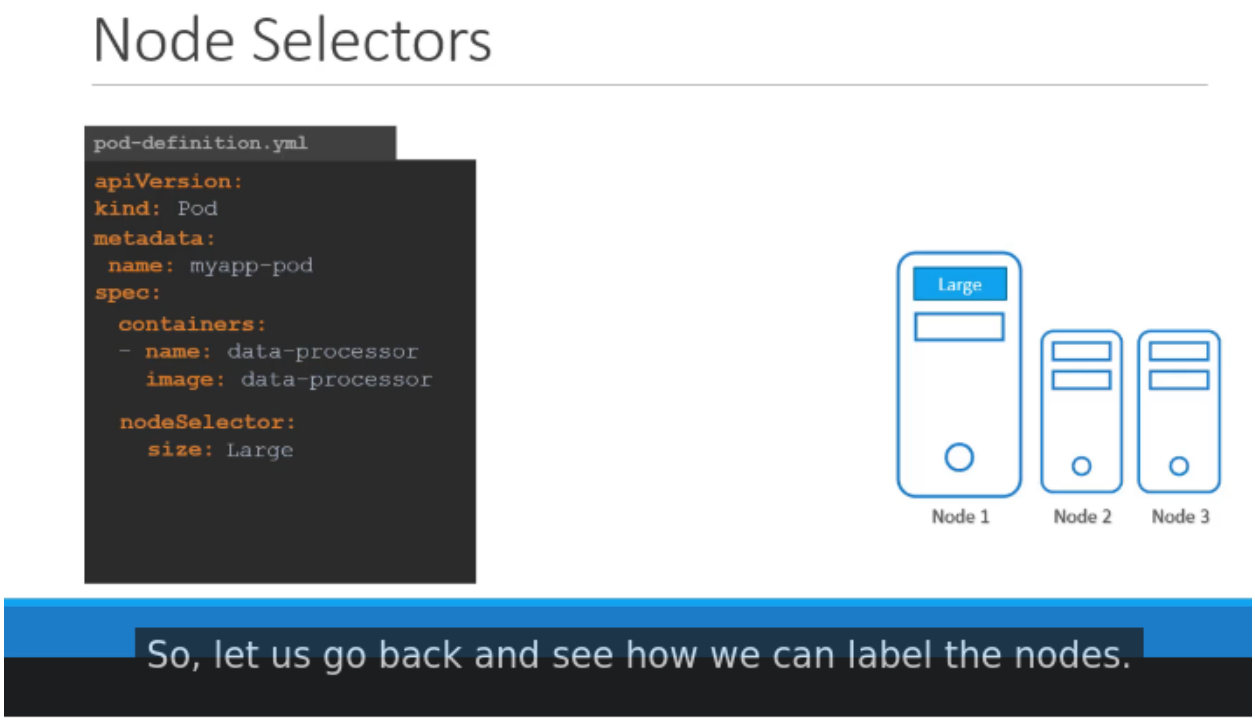
Spec:
    Container:
    nodeSelector:
        Size: Large

But from where size Large comes and how k8s know which has Large Node

Basically the key value size and large are label key value pair assigned to nodes

Scheduler uses these labels
To match and identify the right node to place the pods on

For this we first label our nodes to use this one before creating this pod



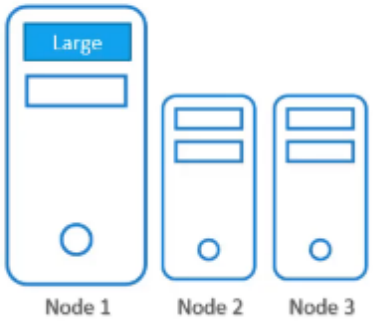
How we can the labels the nodes

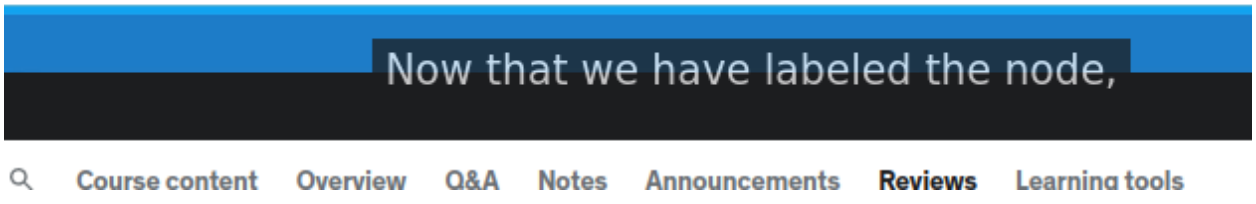
Kubectl label nodes node-name label-key=label-value
Kubectl label nodes node-1 size=Large

Now we can create pod
With node selector as size Large
‘

Label Nodes

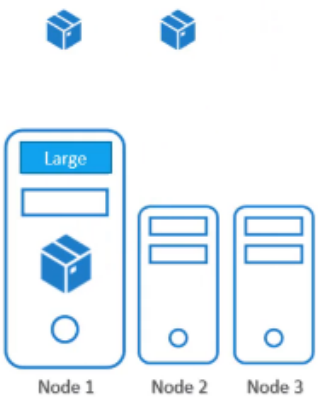
```
kubectl label nodes <node-name> <label-key>=<label-value>
kubectl label nodes node-1 size=Large
```





Now run command

Node Selector



pod-definition.yml

```yaml
apiVersion:
kind: Pod
metadata:
 name: myapp-pod
spec:
  containers:
  - name: data-processor
    image: data-processor

  nodeSelector:
    size: Large
```

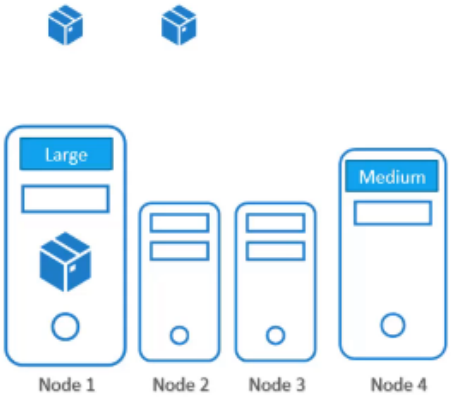
```
kubectl create -f pod-definition.yml
```

it is placed on node one as desired.

Course content | Overview | Q&A | Notes | Announcements | **Reviews** | Learning tools

We use single label and selector to achieve our goal
We can’t use mix of LABELS

Node Selector - Limitations



- Large OR Medium?
- NOT Small

You cannot achieve this using node selectors.