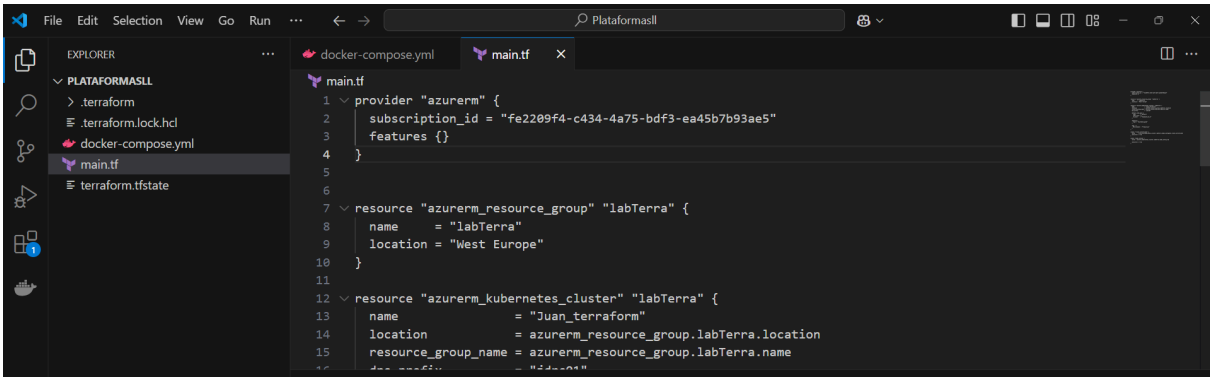
Terraform Workshop

Fisrt of all we need to create an achive to explain everything from our cluster



There are some steps that we need to know before doing the cluster into azure

- We need to have an account with suscription or have credits into azure
- We need to have installed Terraform
- We need to have installed Azure CLI

So whe we want to deploy the cluster we use the next comand lines and specific things like:

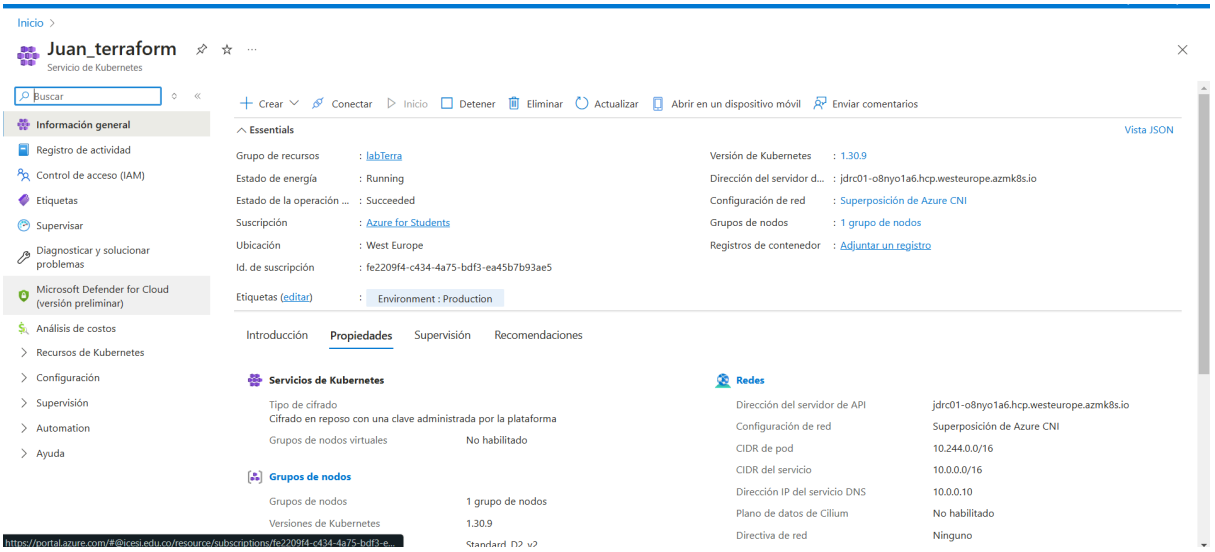
In the start of the document put the provider and the suscription id from our account

```
provider "azurerm" {
  subscription_id = "fe2209f4-c434-4a75-bdf3-ea45b7b93ae5"
  features {}
}
```

We use this commands to deploy the cluster

```
terraform init
terraform validate
terraform plan
terraform apply
```

Here we verify that our service is on azure



Now we switch the context

```
juand@Huawei-Laptop:~/terraform$ kubectl config use-context Juan_terraform
Switched to context "Juan_terraform".
```

Now we create the nginx pod

```
juand@Huawei-Laptop:~/terraform$ kubectl run nginx-pod --image=nginx --restart=Never
}pod/nginx-pod created
juand@Huawei-Laptop:~/terraform$ kubectl get pods
NAME        READY   STATUS    RESTARTS   AGE
nginx-pod   1/1     Running   0          9s
```

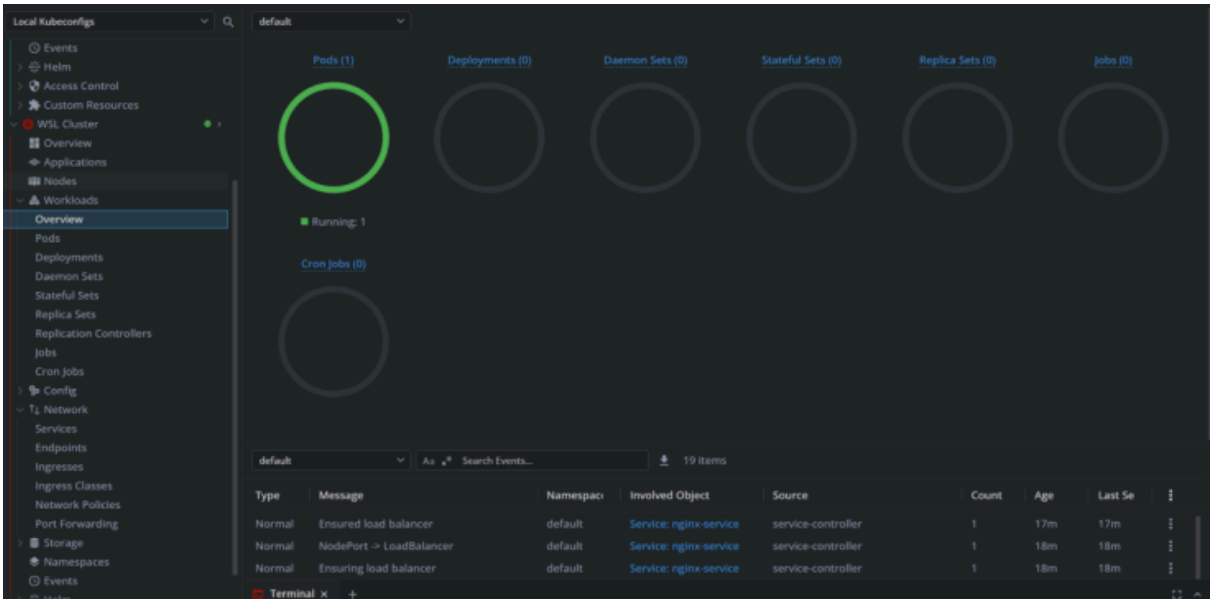
Now we change the type to obtain a port to search in http

```
juand@Huawei-Laptop:~/terraform$ kubectl get svc
NAME         TYPE        CLUSTER-IP    EXTERNAL-IP   PORT(S)   AGE
kubernetes   ClusterIP   10.0.0.1      <none>        443/TCP   100m
nginx-pod    ClusterIP   10.0.93.203   <none>        80/TCP    114s
juand@Huawei-Laptop:~/terraform$ kubectl describe svc nginx-pod
Name:                     nginx-pod
Namespace:                default
Labels:                   run=nginx-pod
Annotations:              <none>
Selector:                 run=nginx-pod
Type:                     ClusterIP
IP Family Policy:         SingleStack
IP Families:              IPv4
IP:                       10.0.93.203
IPs:                      10.0.93.203
Port:                     <unset>  80/TCP
TargetPort:               80/TCP
Endpoints:                10.244.0.136:80
Session Affinity:         None
Internal Traffic Policy:  Cluster
Events:                   <none>
juand@Huawei-Laptop:~/terraform$ kubectl get svc nginx-pod -o=jsonpath='{.spec.ports[0].nodePort}'
juand@Huawei-Laptop:~/terraform$ kubectl patch svc nginx-pod -p '{"spec": {"type": "NodePort"}}'
service/nginx-pod patched
juand@Huawei-Laptop:~/terraform$ kubectl get svc nginx-pod
NAME        TYPE       CLUSTER-IP    EXTERNAL-IP   PORT(S)        AGE
nginx-pod   NodePort   10.0.93.203   <none>        80:30303/TCP   3m10s
juand@Huawei-Laptop:~/terraform$ kubectl get svc nginx-pod -o=jsonpath='{.spec.ports[0].nodePort}'
30303juand@Huawei-Laptop:~/terraform$ minikube ip
```

```
uand@Huawei-Laptop:~/terraform$ curl http://localhost:8080
{
uand@Huawei-Laptop:~/terraform$ kubectl get svc nginx-pod -o=jsonpath='{.spec.ports[0].nodePort}'
0303juand@Huawei-Laptop:~/terrafkubectl proxy --address=0.0.0.0 --port=8001 --accept-hosts='.*'='.*'
tarting to serve on [::]:8001
```

Here we:

- Switch kubectl to cluster context
- create nginx pod
- create service and on this amount the nginx so that port 80 is seen
- download lenz in window


Local Kubeconfigs
default
Events
Helm
Access Control
Custom Resources
WSL Cluster
Overview
Applications
Nodes
Workloads
Overview
Pods
Deployments
Daemon Sets
Stateful Sets
Replica Sets
Replication Controllers
Jobs
Cron Jobs
Config
Network
Services
Endpoints
Ingresses
Ingress Classes
Network Policies
Port Forwarding
Storage
Namespaces
Events
Pods (1)
Deployments (0)
Daemon Sets (0)
Stateful Sets (0)
Replica Sets (0)
Jobs (0)
Running: 1
Cron Jobs (0)
default
Search Events...
19 items
Type
Message
Namespac
Involved Object
Source
Count
Age
Last Se
Normal
Ensured load balancer
default
Service: nginx-service
service-controller
1
17m
17m
Normal
NodePort -> LoadBalancer
default
Service: nginx-service
service-controller
1
18m
18m
Normal
Ensuring load balancer
default
Service: nginx-service
service-controller
1
18m
18m
Terminal