

BONUS EXERCISE:

Configure 3 nodes Kubernetes cluster with a master and 2 nodes(node 1,node2) using kubeadm.

Answer: For this there are some steps that we need follow. They are:

- Need to configure Virtual Machine
- Install container runtime on every node like docker
- Install kubeadm in every node
- Need to initialize master server
- Set-up the pod network
- Join worker node to the master node

After going through all the steps as described in the document of Kubernetes, I did the above configuration.

Showing screenshots in the support of this.

```
Usage of /: 2.7% of 38.71GB  Users logged in: 0
Memory usage: 6%          IP address for enp0s3: 10.0.2.15
Swap usage: 0%            IP address for enp0s8: 192.168.56.2

0 updates can be applied immediately.

New release '20.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

[vagrant@kubemaster:~]$ uptime
18:49:08 up 3 min, 1 user, load average: 0.18, 0.18, 0.08
[vagrant@kubemaster:~]$ logout
Connection to 127.0.0.1 closed.
akshaykumar87@USBDGAKSHAYKU1 certified-kubernetes-administrator-course % vagrant ssh kubemaster
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 4.15.0-162-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Thu Nov 11 18:49:47 UTC 2021

System load: 0.08      Processes:            101
Usage of /: 2.7% of 38.71GB  Users logged in:    0
Memory usage: 6%        IP address for enp0s3: 10.0.2.15
Swap usage: 0%          IP address for enp0s8: 192.168.56.3

0 updates can be applied immediately.

New release '20.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

[vagrant@kubemaster:~]$ logout
Connection to 127.0.0.1 closed.
akshaykumar87@USBDGAKSHAYKU1 certified-kubernetes-administrator-course % vagrant ssh kubemaster
The machine with the name 'kubemaster' was not found configured for
this Vagrant environment.
akshaykumar87@USBDGAKSHAYKU1 certified-kubernetes-administrator-course % vagrant ssh kubemaster
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 4.15.0-162-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Thu Nov 11 18:50:16 UTC 2021

System load: 0.08      Processes:            101
Usage of /: 2.7% of 38.71GB  Users logged in:    0
Memory usage: 6%        IP address for enp0s3: 10.0.2.15
Swap usage: 0%          IP address for enp0s8: 192.168.56.4

0 updates can be applied immediately.

New release '20.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

[vagrant@kubemaster:~]$ logout
```

```

=> kubenode01: Configuring and enabling network interfaces...
=> kubenode01: Mounting shared folders...
kubnode01: /vagrant => /Users/akshaykumar87/Desktop/certified-kubernetes-administrator-course
=> kubenode01: Running provisioner: setup-hosts (shell)...
kubnode01: Running: /var/folders/53/wdv3d19x7d18xwczjgvs753w000gp/T/vagrant-shell120211112-1841-131oqd7.sh
=> kubenode01: Running provisioner: setup-dns (shell)...
kubnode01: Running: /var/folders/53/wdv3d19x7d18xwczjgvs753w000gp/T/vagrant-shell120211112-1841-1jmd0bt.sh
=> kubenode02: Importing base box 'ubuntu/bionic64'...
=> kubenode02: Matching MAC address for NAT networking...
=> kubenode02: Setting the name of the VM: kubenode02
=> kubenode02: Fixed port collision for 22 => 2222. Now on port 2201.
=> kubenode02: Clearing any previously set network interfaces...
=> kubenode02: Preparing network interfaces based on configuration...
kubnode02: Adapter 1: nat
kubnode02: Adapter 2: hostonly
=> kubenode02: Forwarding ports...
kubnode02: 22 (guest) => 2222 (host) (adapter 1)
kubnode02: 22 (guest) => 2201 (host) (adapter 1)
=> kubenode02: Running 'pre-boot' VM customizations...
=> kubenode02: Booting VM...
=> kubenode02: Waiting for machine to boot. This may take a few minutes...
kubnode02: SSH address: 127.0.0.1:2201
kubnode02: SSH username: vagrant
kubnode02: SSH auth method: private key
kubnode02: Warning: Connection reset. Retrying...
kubnode02: Warning: Remote connection disconnect. Retrying...
kubnode02:
kubnode02: Vagrant insecure key detected. Vagrant will automatically replace
kubnode02: this with a newly generated keypair for better security.
kubnode02:
kubnode02: Inserting generated public key within guest...
kubnode02: Removing insecure key from the guest if it's present...
kubnode02: Key inserted! Disconnecting and reconnecting using new SSH key...
=> kubenode02: Machine booted and ready!
=> kubenode02: Checking for guest additions in VM...
kubnode02: The guest additions on this VM do not match the installed version of
kubnode02: VirtualBox! In most cases this is fine, but in rare cases it can
kubnode02: prevent things such as shared folders from working properly. If you see
kubnode02: shared folder errors, please make sure the guest additions within the
kubnode02: virtual machine match the version of VirtualBox you have installed on
kubnode02: your host and reload your VM.
kubnode02:
kubnode02: Guest Additions Version: 5.2.42
kubnode02: VirtualBox Version: 6.1
=> kubenode02: Setting hostname...
=> kubenode02: Configuring and enabling network interfaces...
=> kubenode02: Mounting shared folders...
kubnode02: /vagrant => /Users/akshaykumar87/Desktop/certified-kubernetes-administrator-course
=> kubenode02: Running provisioner: setup-hosts (shell)...
kubnode02: Running: /var/folders/53/wdv3d19x7d18xwczjgvs753w000gp/T/vagrant-shell120211112-1841-19r27jt.sh
=> kubenode02: Running provisioner: setup-dns (shell)...
kubnode02: Running: /var/folders/53/wdv3d19x7d18xwczjgvs753w000gp/T/vagrant-shell120211112-1841-7w8qer.sh
akshaykumar87@USBD0AKSHAYKU1: ~$ certied-kubernetes-administrator-course % vagrant status
Current machine states:

kubemaster      running (virtualbox)
kubnode01       running (virtualbox)
kubnode02       running (virtualbox)

This environment represents multiple VMs. The VMs are all listed
above with their current state. For more information about a specific
VM, run 'vagrant status NAME'.
akshaykumar87@USBD0AKSHAYKU1: ~$ certied-kubernetes-administrator-course %

```

```

...nistrator-course -- vagrant@kubemaster: ~ -- ssh - vagrant ssh kubemaster
root@kubemaster:~# logout
vagrant@kubemaster:~$ mkdir -p $HOME/.kube
vagrant@kubemaster:~$ sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
vagrant@kubemaster:~$ sudo chown $(id -u):$(id -g) $HOME/.kube/config
vagrant@kubemaster:~$ kubectl get status
error: the server doesn't have a resource type "status"
vagrant@kubemaster:~$ kubectl get nodes
NAME                STATUS    ROLES    AGE   VERSION
kubemaster          NotReady control-plane,master   70s   v1.22.3
vagrant@kubemaster:~$ kubectl apply -f "https://cloud.weave.works/k8s/net?k8s-version=$(kubectl version | base64 | tr -d '\n')"
Command 'base64' not found, did you mean:
  command 'base64' from deb coreutils

Try: apt install <deb name>

^[[Error: unable to read URL "https://cloud.weave.works/k8s/net?k8s-version=", server reported 400 Bad Request, status code=400
vagrant@kubemaster:~$ kubectl apply -f "https://cloud.weave.works/k8s/net?k8s-version=$(kubectl version | base64 | tr -d '\n')"
serviceaccount/weave-net created
clusterrole.rbac.authorization.k8s.io/weave-net created
clusterrolebinding.rbac.authorization.k8s.io/weave-net created
role.rbac.authorization.k8s.io/weave-net created
rolebinding.rbac.authorization.k8s.io/weave-net created
daemonset.apps/weave-net created
vagrant@kubemaster:~$ kubectl get nodes
NAME                STATUS    ROLES    AGE   VERSION
kubemaster          NotReady control-plane,master   11m   v1.22.3
vagrant@kubemaster:~$ kubectl get nodes
NAME                STATUS    ROLES    AGE   VERSION
kubemaster          Ready     control-plane,master   13m   v1.22.3
kubnode01            Ready     <none>    117s   v1.22.3
kubnode02            NotReady <none>    34s   v1.22.3
vagrant@kubemaster:~$ kubectl get nodes
NAME                STATUS    ROLES    AGE   VERSION
kubemaster          Ready     control-plane,master   13m   v1.22.3
kubnode01            Ready     <none>    2m3s   v1.22.3
kubnode02            NotReady <none>    40s   v1.22.3
vagrant@kubemaster:~$ kubectl get nodes
NAME                STATUS    ROLES    AGE   VERSION
kubemaster          Ready     control-plane,master   14m   v1.22.3
kubnode01            Ready     <none>    2m8s   v1.22.3
kubnode02            Ready     <none>    45s   v1.22.3
vagrant@kubemaster:~$

```

Last screenshot showing the all three nodes are in the ready status.

1. Deploy a nginx deployment with name web1 and custom index.html. Express it to nodePort of 30080 in web namespace for this scenario.

Answer:

For accomplishing this need to use **congiMap**. This is used to add custom content to the nginx deployment.

Here, made 3 yaml files for this. Showing below one by one.

Conf.yaml file:

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: index-config
  namespace: default
data:
  index.html: |
    <html>
    <h1>Welcome Linker</h1>
    </br>
    <h1>This is the custom index file.</h1>
    </html>
```

Created **myapp.yaml** file for the deployment of nginx.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  namespace: default
spec:
  selector:
    matchLabels:
      app: nginx
```

```
replicas: 2
template:
  metadata:
    labels:
      app: nginx
  spec:
    containers:
      - name: nginx
        image: nginx:latest
        ports:
          - containerPort: 80
        volumeMounts:
          - name: nginx-index-file
            mountPath: /usr/share/nginx/html/
    volumes:
      - name: nginx-index-file
        configMap:
          name: index-config
```

Lastly created service file as servi.yaml:

```
apiVersion: v1
kind: Service
metadata:
  name: nginx-service
  namespace: default
spec:
  selector:
    app: nginx
  type: NodePort
  ports:
    - port: 80
      nodePort: 30080
      targetPort: 80
```

2.. Create a wordpress deployment with mysql as a database and expose it to nodeport 30081. Use “volume” namespace for this scenario and login to wordpress with predefined credentials.

Answer: For accomplishing the above we need to do some important steps. Where we need PV, PVC and secrets for storing the credentials.

Creating a kustomization.yaml file where it contains a Secret generator, Mysql resource config, wordpress resource config.

Following is the command for adding Secret generator:

```
Cat <<EOF >./kustomization.yaml
secretGenerator:
  - name: mysql-pass
    literals:
  - password=akshay01
EOF
```

After this need to add the resource config for mysql and wordpress.

Uploaded the same in my git hub account.

Now downloading the mysql deployment configuration file
And wordpress config file.

After this need to add these in kustomization.yaml file.

And then can apply it using the command:

We need to then apply it. So for this we can use command:

Kubectl apply -k ./

And then running the command

minikube service wordpress --url can get the url and we can check this as well.

In support of this showing the screenshot as well:

```
Last login: Wed Nov 10 21:12:33 on ttys000
akshaykumar87@USBGOAKSHAYKU1 ~ % kubectl apply -k ./
error: unable to find one of 'kustomization.yaml', 'kustomization.yml' or 'Kustomization' in directory '/Users/akshaykumar87'
akshaykumar87@USBGOAKSHAYKU1 ~ % ls
Desktop      Downloads    Movies       Parallels    Public       kubectl.sha256  minikube-darwin-arm64
Documents    Library      Music        Pictures     Vagrantfile   kuber
akshaykumar87@USBGOAKSHAYKU1 ~ % cd kuber
akshaykumar87@USBGOAKSHAYKU1 kuber % kubectl apply -k ./
secret/mysql-pass-6999bkmk7d created
service/wordpress created
service/wordpress-mysql created
persistentvolumeclaim/mysql-pv-claim created
persistentvolumeclaim/wp-pv-claim created
deployment.apps/wordpress created
deployment.apps/wordpress-mysql created
akshaykumar87@USBGOAKSHAYKU1 kuber % kubectl get secrets
NAME                                TYPE                                DATA  AGE
default-token-6lgjj                kubernetes.io/service-account-token  3      21h
mysql-pass-6999bkmk7d              Opaque                               1      14s
akshaykumar87@USBGOAKSHAYKU1 kuber % kubectl get pvc
NAME                                STATUS  VOLUME                                     CAPACITY  ACCESS MODES  STORAGECLASS  AGE
mysql-pv-claim                     Bound   pvc-6983ae57-a79a-4239-9b2d-2dc82b44935d  20Gi      RWO            standard      32s
wp-pv-claim                        Bound   pvc-7a4c8d7d-3b95-4d48-8008-f3b972311af9  20Gi      RWO            standard      32s
akshaykumar87@USBGOAKSHAYKU1 kuber % kubectl get pods
NAME                                READY  STATUS    RESTARTS  AGE
my-deployment-8b45d68b4-gmbdx       1/1    Running   0          110m
my-deployment-8b45d68b4-mfdwf       1/1    Running   0          110m
my-deployment-8b45d68b4-q4lw7       1/1    Running   0          110m
nginx-deployment-c7c8cb6d4-5nhp5    1/1    Running   0          98m
nginx-deployment-c7c8cb6d4-grs2c    1/1    Running   0          98m
replica-set-2bhjx                   1/1    Running   0          110m
replica-set-9htwq                   1/1    Running   0          110m
replica-set-g82n7                   1/1    Running   0          110m
wordpress-644958c448-cjf8x          0/1    ContainerCreating  0          50s
wordpress-mysql-67456c9c78-xws2w    1/1    Running   0          50s
akshaykumar87@USBGOAKSHAYKU1 kuber % kubectl get services wordpress
NAME      TYPE        CLUSTER-IP    EXTERNAL-IP    PORT(S)    AGE
wordpress LoadBalancer 10.99.171.76   <pending>      80:30231/TCP  62s
akshaykumar87@USBGOAKSHAYKU1 kuber % minikube service wordpress --url
🔗 Starting tunnel for service wordpress.
|-----|
| NAMESPACE | NAME | TARGET PORT | URL |
|-----|
| default | wordpress | | http://127.0.0.1:64140 |
|-----|
http://127.0.0.1:64140
! Because you are using a Docker driver on darwin, the terminal needs to be open to run it.
```

Showing the creation of all above files and the url of wordpress.

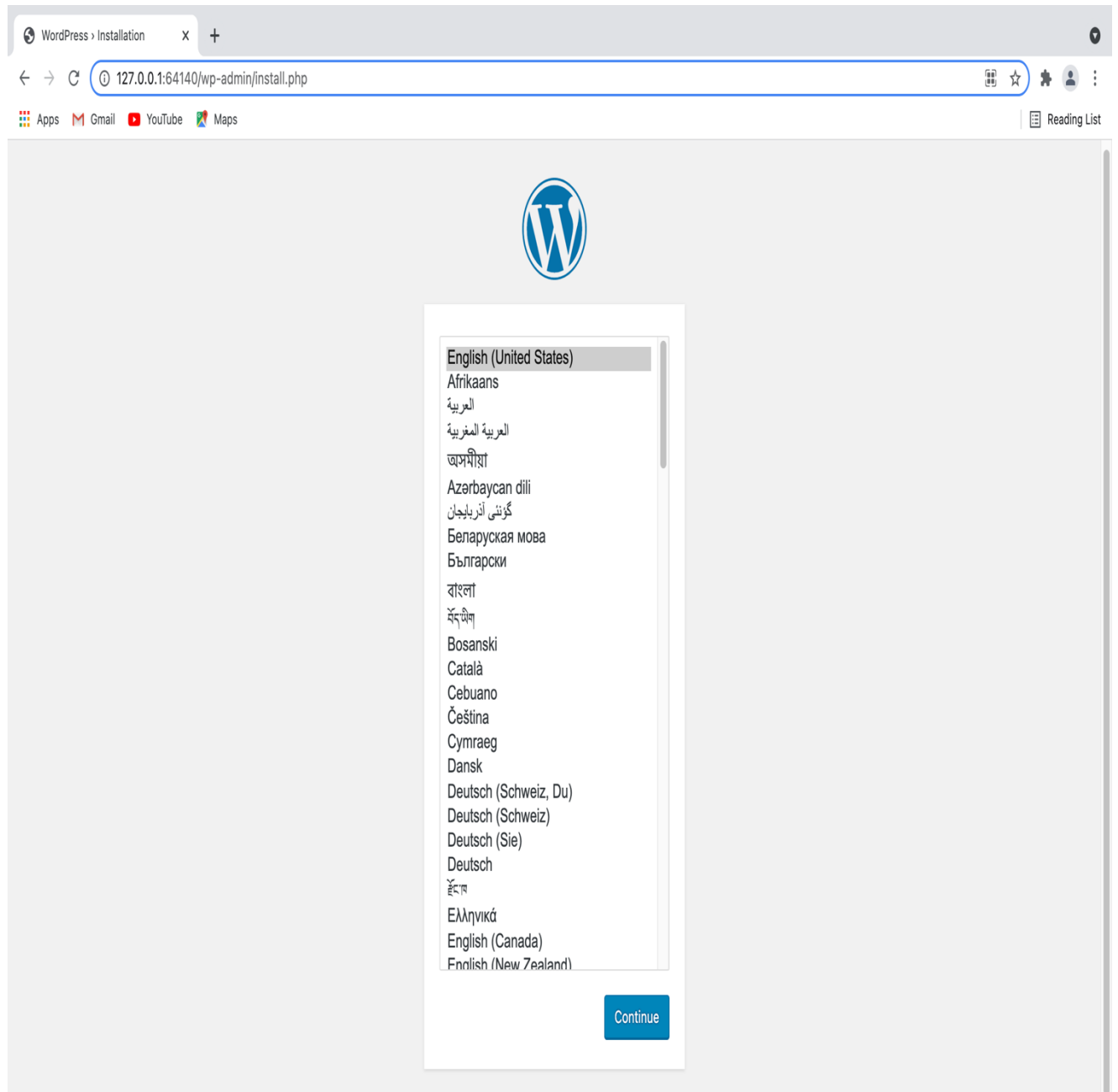


Figure showing the accessing part of wordpress.

3. Create a pod named sidecar with 2 containers named main and sidecar respectively. Fetch the logs from app container to sidecar container. Use nginx for main and busybox:1.28 image for sidecar container. Use namespace sidecar for this scenario.

Answer:

For accomplish this it can be done in many ways. Here, I;m doing the communication using sharing the same volume.

For this made a myapp.yaml file where the sidecar container add the current date to the volume and when the user request it from the main then it read and return the same from the volume where it is mounted.

Showing the myapp.yaml file for the same.

```
apiVersion: v1
kind: Pod
metadata:
  name: sidecar
spec:
  volumes:
  - name: html
    emptyDir: {}
  containers:
  - name: main
    image: nginx
    volumeMounts:
    - name: html
      mountPath: /usr/share/nginx/html
  - name: sidecar
    image: busybox:1.28
    volumeMounts:
    - name: html
      mountPath: /html
    command: ["/bin/sh", "-c"]
    args:
      - while true; do
        date >> /html/index.html;
        sleep 1;
      done
```


Here showing the output:

```
...nistrator-course -- vagrant@kubemaster: ~ -- ssh • vagrant ssh kubemaster    ...nistrator-course -- root@kubenode01: ~ -- ssh • vagrant ssh kubemaster    ...nistrator-course -- root@kubenode02: ~ -- ssh • vagrant ssh kubemaster02    +
vagrant@kubemaster:~$ clear
vagrant@kubemaster:~$ ls
vagrant@kubemaster:~$ vi first.yaml
vagrant@kubemaster:~$ vi first.yaml
vagrant@kubemaster:~$ kubectl exec sidecar -c main -- /bin/cat /usr/share/nginx/html/index.html
Error from server (NotFound): pods "sidecar" not found
vagrant@kubemaster:~$ vi first.yaml
vagrant@kubemaster:~$ kubectl apply -f first.yaml
pod/sidecar created
vagrant@kubemaster:~$ kubectl exec sidecar -c main -- /bin/cat /usr/share/nginx/html/index.html
error: unable to upgrade connection: container not found ("main")
vagrant@kubemaster:~$ vi first.yaml
vagrant@kubemaster:~$ kubectl exec sidecar -c sidecar -- /bin/cat /html/index.html
Thu Nov 11 21:10:13 UTC 2021
Thu Nov 11 21:10:14 UTC 2021
Thu Nov 11 21:10:15 UTC 2021
Thu Nov 11 21:10:16 UTC 2021
Thu Nov 11 21:10:17 UTC 2021
Thu Nov 11 21:10:18 UTC 2021
Thu Nov 11 21:10:19 UTC 2021
Thu Nov 11 21:10:20 UTC 2021
Thu Nov 11 21:10:21 UTC 2021
Thu Nov 11 21:10:22 UTC 2021
Thu Nov 11 21:10:23 UTC 2021
Thu Nov 11 21:10:24 UTC 2021
Thu Nov 11 21:10:25 UTC 2021
Thu Nov 11 21:10:26 UTC 2021
Thu Nov 11 21:10:27 UTC 2021
Thu Nov 11 21:10:28 UTC 2021
Thu Nov 11 21:10:29 UTC 2021
Thu Nov 11 21:10:30 UTC 2021
Thu Nov 11 21:10:31 UTC 2021
Thu Nov 11 21:10:32 UTC 2021
Thu Nov 11 21:10:33 UTC 2021
Thu Nov 11 21:10:34 UTC 2021
Thu Nov 11 21:10:35 UTC 2021
Thu Nov 11 21:10:36 UTC 2021
Thu Nov 11 21:10:37 UTC 2021
Thu Nov 11 21:10:38 UTC 2021
Thu Nov 11 21:10:39 UTC 2021
Thu Nov 11 21:10:40 UTC 2021
Thu Nov 11 21:10:41 UTC 2021
Thu Nov 11 21:10:42 UTC 2021
Thu Nov 11 21:10:43 UTC 2021
Thu Nov 11 21:10:44 UTC 2021
Thu Nov 11 21:10:45 UTC 2021
Thu Nov 11 21:10:46 UTC 2021
Thu Nov 11 21:10:47 UTC 2021
Thu Nov 11 21:10:48 UTC 2021
Thu Nov 11 21:10:49 UTC 2021
Thu Nov 11 21:10:50 UTC 2021
Thu Nov 11 21:10:51 UTC 2021
Thu Nov 11 21:10:52 UTC 2021
Thu Nov 11 21:10:53 UTC 2021
Thu Nov 11 21:10:54 UTC 2021
Thu Nov 11 21:10:55 UTC 2021
Thu Nov 11 21:10:56 UTC 2021
Thu Nov 11 21:10:57 UTC 2021
Thu Nov 11 21:10:58 UTC 2021
Thu Nov 11 21:10:59 UTC 2021
```