

## WEEKLY-2

Name: Ankit Tiwari

Ps.No : 10844395

Q1->Configure a sample apache server using configuration management tool Ansible on your managehost

Ans->

Steps:-

- 1> Create two instance one will be master -node and another will be worker -node

Specifications master-node(ansible-instance):

Redhat

Us-east-1

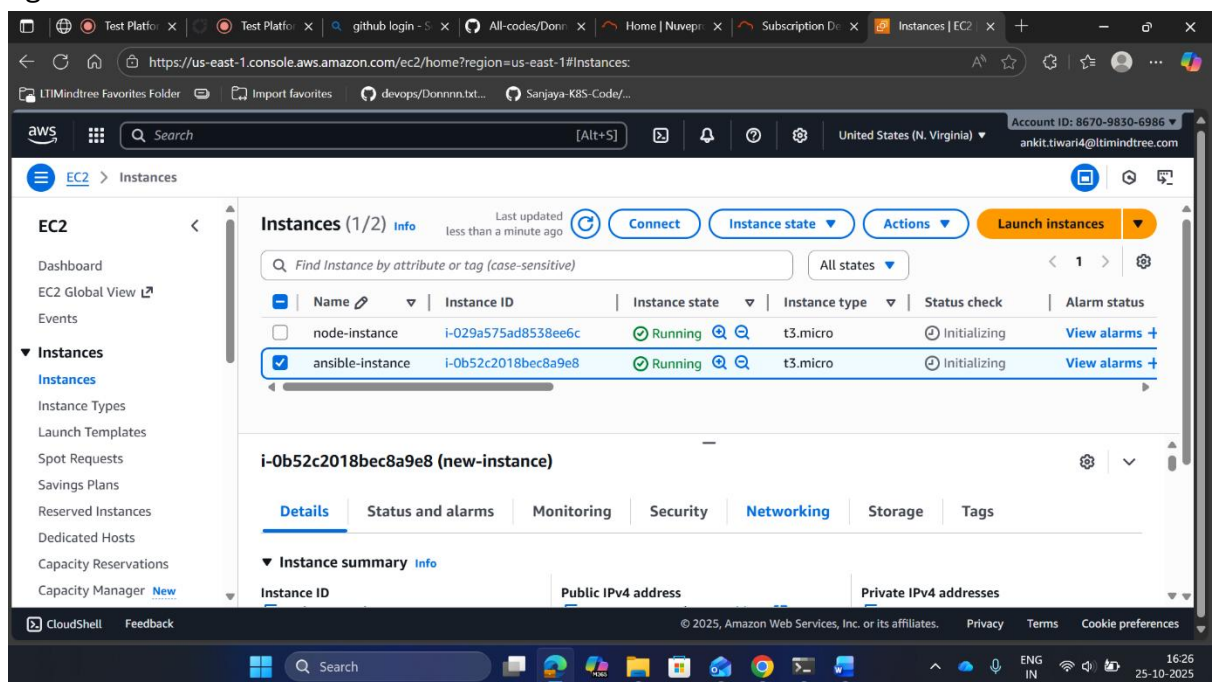
15g GP2

Specifications worker-node(node-instance):

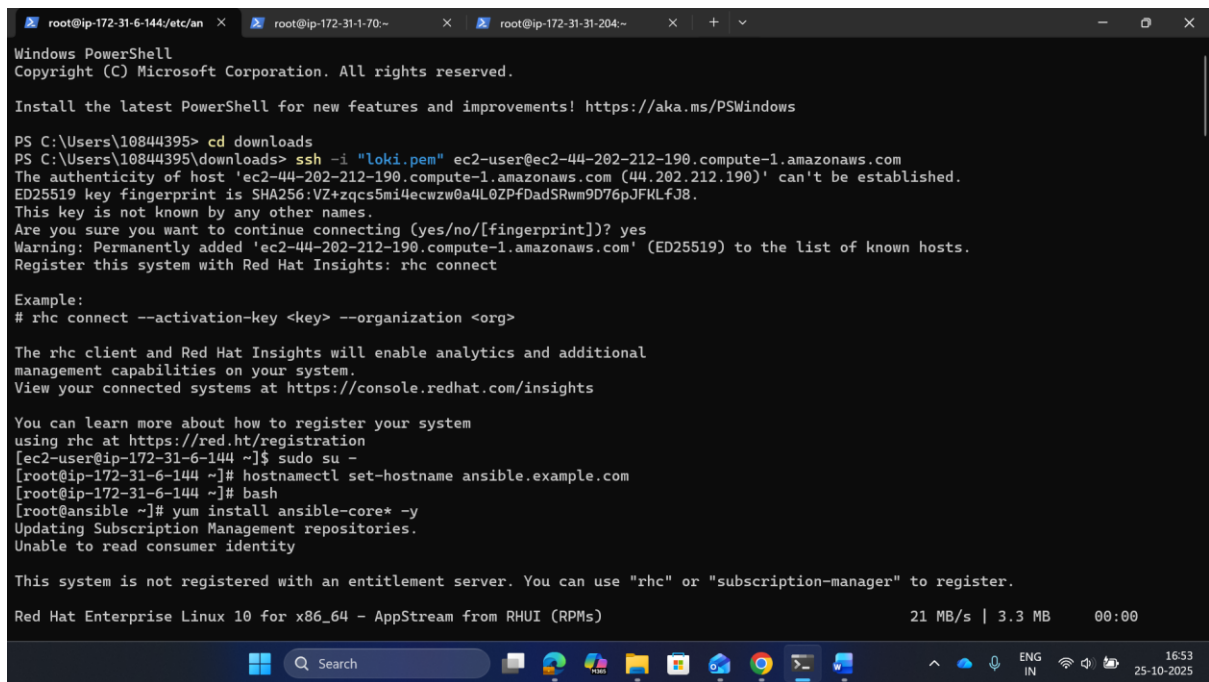
Aws linux

Us-east-1

8g GP2



## ➔ Installing ansible on instance



```
root@ip-172-31-6-144:/etc/an x root@ip-172-31-1-70:~ x root@ip-172-31-31-204:~ x + v
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\10844395> cd downloads
PS C:\Users\10844395\downloads> ssh -i "loki.pem" ec2-user@ec2-44-202-212-190.compute-1.amazonaws.com
The authenticity of host 'ec2-44-202-212-190.compute-1.amazonaws.com (44.202.212.190)' can't be established.
ED25519 key fingerprint is SHA256:VZ+zzqs5mi4ecwz0a4L0ZPfDadSRwm9D76pJFKLfJ8.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-44-202-212-190.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
Register this system with Red Hat Insights: rhc connect

Example:
# rhc connect --activation-key <key> --organization <org>

The rhc client and Red Hat Insights will enable analytics and additional
management capabilities on your system.
View your connected systems at https://console.redhat.com/insights

You can learn more about how to register your system
using rhc at https://red.ht/registration
[ec2-user@ip-172-31-6-144 ~]$ sudo su -
[root@ip-172-31-6-144 ~]# hostnamectl set-hostname ansible.example.com
[root@ip-172-31-6-144 ~]# bash
[root@ansible ~]# yum install ansible-core* -y
Updating Subscription Management repositories.
Unable to read consumer identity

This system is not registered with an entitlement server. You can use "rhc" or "subscription-manager" to register.

Red Hat Enterprise Linux 10 for x86_64 - AppStream from RHUI (RPMs) 21 MB/s | 3.3 MB 00:00
```

## ➔ Editing ansible.cfg file

```
root@ip-172-31-6-144:/etc/ansible$ cat ansible.cfg
# paramiko on older platforms rather than removing it
ssh_args = -o ControlMaster=auto -o ControlPersist=60s

# The path to use for the ControlPath sockets. This defaults to
# "%(directory)s/ansible-ssh-%%h-%%p-%%r", however on some systems with
# very long hostnames or very long path names (caused by long user names or
# deeply nested home directories) this can exceed the character limit on
# file socket names (108 characters for most platforms). In that case, you
# may wish to shorten the string below.
#
# Example:
# control_path = %(directory)s/%%h-%%p-%%r
#control_path = %(directory)s/ansible-ssh-%%h-%%p-%%r

# Enabling pipelining reduces the number of SSH operations required to
# execute a module on the remote server. This can result in a significant
# performance improvement when enabled, however when using "sudo:" you must
# first disable 'requiretty' in /etc/sudoers
#
# By default, this option is disabled to preserve compatibility with
# sudoers configurations that have requiretty (the default on many distros).
#
#pipelining = False

# if True, make ansible use scp if the connection type is ssh
# (default is sftp)
#scp_if_ssh = True

[accelerate]
accelerate_port = 5099
accelerate_timeout = 30
accelerate_connect_timeout = 5.0
"ansible.cfg" 177L, 6733B
```

➔ Adding the worker machine on master-node

```
root@ip-172-31-6-144:/etc/ansible$ cat hosts
## 192.168.1.110

# If you have multiple hosts following a pattern, you can specify
# them like this:
## www[001:006].example.com

# You can also use ranges for multiple hosts:
## db-[99:101]-node.example.com

# Ex 3: A collection of database servers in the 'dbservers' group:
## [dbservers]
##
## db01.intranet.mydomain.net
## db02.intranet.mydomain.net
## 10.25.1.56
## 10.25.1.57

# Ex4: Multiple hosts arranged into groups such as 'Debian' and 'openSUSE':
## [Debian]
## alpha.example.org
## beta.example.org

## [openSUSE]
## green.example.com
## blue.example.com
[dev-server]
172.31.1.70
"hosts" 55L, 1199B
```

➔ Checking if the worker machine is reachable or not



## ➔ executing the playbook

```
root@ip-172-31-6-144:/etc/ans x root@ip-172-31-1-70:~ x root@ip-172-31-31-204:~ x + v
[root@ansible ansible]# vim hosts
[root@ansible ansible]# ansible all --list-hosts
[WARNING]: Invalid characters were found in group names but not replaced, use -vvvv to see details
hosts (1):
  172.31.1.70
[root@ansible ansible]# vim apache.yaml
[root@ansible ansible]# ansible-playbook apache.yaml
[WARNING]: Invalid characters were found in group names but not replaced, use -vvvv to see details
[DEPRECATION WARNING]: The 'smart' option for connections is deprecated. Set the connection plugin directly instead. This feature
will be removed in version 2.20. Deprecation warnings can be disabled by setting deprecation_warnings=False in ansible.cfg.

PLAY [tasks] *****

TASK [Gathering Facts] *****
The authenticity of host '172.31.1.70 (172.31.1.70)' can't be established.
ED25519 key fingerprint is SHA256:ytRG3HxHjbU7PCbH8/046qsyrJt5YeyU3ut9sVwTpgU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
[WARNING]: Platform linux on host 172.31.1.70 is using the discovered Python interpreter at /usr/bin/python3.9, but future
installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-
core/2.16/reference_appendices/interpreter_discovery.html for more information.
ok: [172.31.1.70]

TASK [Install httpd] *****
changed: [172.31.1.70]

TASK [Start and enable httpd service] *****
changed: [172.31.1.70]

PLAY RECAP *****
172.31.1.70 : ok=3 changed=2 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

[root@ansible ansible]#
```

## ➔ installing python on node-machine

```
root@ip-172-31-6-144:/etc/ans x root@ip-172-31-1-70:~ x root@ip-172-31-31-204:~ x + v
-rw----- 1 root root 552 Oct 25 10:56 authorized_keys
[root@worker-node .ssh]# vim authorized_keys
[root@worker-node .ssh]# cd
[root@worker-node ~]# yum install python -y
Amazon Linux 2023 Kernel Livepatch repository                256 kB/s | 26 kB    00:00
Dependencies resolved.
=====
Package                                Architecture      Version           Repository          Size
=====
Installing:
python-unversioned-command              noarch            3.9.23-1.amzn2023.0.3  amazonlinux         11 k
=====
Transaction Summary
=====
Install 1 Package

Total download size: 11 k
Installed size: 23
Downloading Packages:
python-unversioned-command-3.9.23-1.amzn2023.0.3.noarch.rpm 323 kB/s | 11 kB    00:00
-----
Total
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing      : python-unversioned-command-3.9.23-1.amzn2023.0.3.noarch 1/1
  Installing     : python-unversioned-command-3.9.23-1.amzn2023.0.3.noarch 1/1
  Running scriptlet: python-unversioned-command-3.9.23-1.amzn2023.0.3.noarch 1/1
  Verifying      : python-unversioned-command-3.9.23-1.amzn2023.0.3.noarch 1/1
Installed:

```

➔ successfully apache server is hosted and accessible on port: 80

```
root@ip-172-31-6-144/etc/ans x root@ip-172-31-1-70:~ x root@ip-172-31-31-204:~ x + v
Running transaction
  Preparing      :                                1/1
  Installing     : python-unversioned-command-3.9.23-1.amzn2023.0.3.noarch 1/1
  Running scriptlet: python-unversioned-command-3.9.23-1.amzn2023.0.3.noarch 1/1
  Verifying      : python-unversioned-command-3.9.23-1.amzn2023.0.3.noarch 1/1

Installed:
  python-unversioned-command-3.9.23-1.amzn2023.0.3.noarch

Complete!
[root@worker-node ~]# rpmquery httpd
httpd-2.4.65-1.amzn2023.0.1.x86_64
[root@worker-node ~]# systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
   Active: active (running) since Sat 2025-10-25 11:16:38 UTC; 36s ago
     Docs: man:httpd.service(8)
   Main PID: 26417 (httpd)
    Status: "Total requests: 0; Idle/Busy workers 100/0; Requests/sec: 0; Bytes served/sec: 0 B/sec"
     Tasks: 177 (Limit: 1053)
    Memory: 13.3M
       CPU: 92ms
    CGroup: /system.slice/httpd.service
            └─26417 /usr/sbin/httpd -DFOREGROUND
              └─26418 /usr/sbin/httpd -DFOREGROUND
                └─26419 /usr/sbin/httpd -DFOREGROUND
                  └─26420 /usr/sbin/httpd -DFOREGROUND
                    └─26421 /usr/sbin/httpd -DFOREGROUND

Oct 25 11:16:38 worker-node.example.com systemd[1]: Starting httpd.service - The Apache HTTP Server...
Oct 25 11:16:38 worker-node.example.com systemd[1]: Started httpd.service - The Apache HTTP Server.
Oct 25 11:16:38 worker-node.example.com httpd[26417]: Server configured, listening on: port 80
[root@worker-node ~]#
```

Q2->Deploy a Ngnix application on your kubernetes cluster. and it should be available across of the cluster on port NO. 80.

Ans->

Steps:-

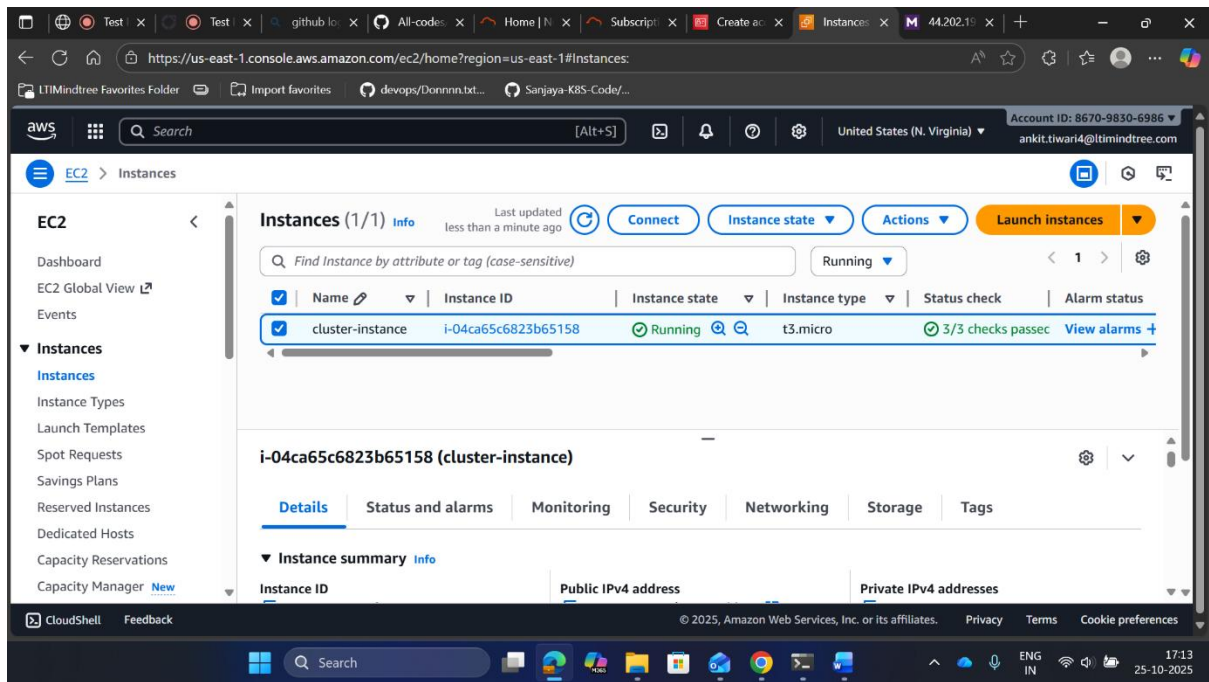
➔ create an ec2 instance (cluster-instance)

specifications:

aws-linux

us-east-1a

12 GP2



➔ Now connecting the instance with the terminal

➔ Running the script to create the cluster and nodegroup for nginx hosting



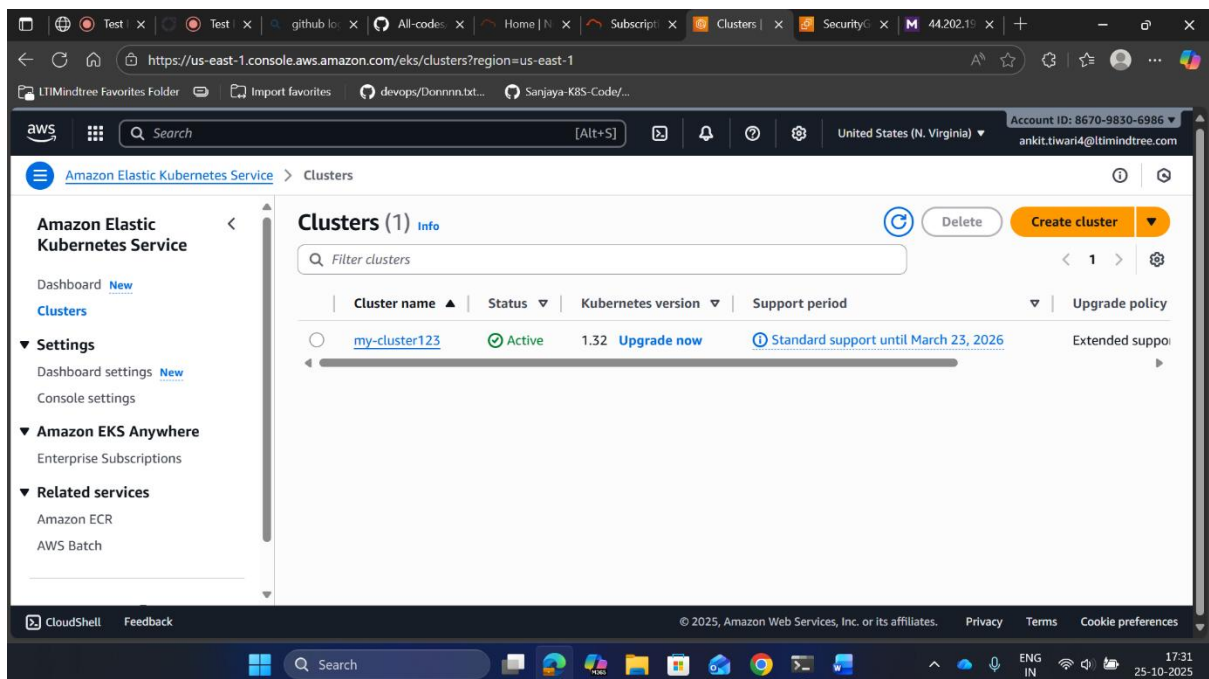






```
root@ip-172-31-204:~ x root@ip-172-31-39-3:~ x + v
2025-10-25 11:53:05 [I] will create a CloudFormation stack for each of 1 managed nodegroups in cluster "my-cluster123"
2025-10-25 11:53:05 [I]
2 sequential tasks: { fix cluster compatibility, 1 task: { 1 task: { create managed nodegroup "my-node-group" } } }
}
2025-10-25 11:53:05 [I] checking cluster stack for missing resources
2025-10-25 11:53:05 [I] cluster stack has all required resources
2025-10-25 11:53:05 [I] building managed nodegroup stack "eksctl-my-cluster123-nodegroup-my-node-group"
2025-10-25 11:53:06 [I] deploying stack "eksctl-my-cluster123-nodegroup-my-node-group"
2025-10-25 11:53:06 [I] waiting for CloudFormation stack "eksctl-my-cluster123-nodegroup-my-node-group"
2025-10-25 11:53:36 [I] waiting for CloudFormation stack "eksctl-my-cluster123-nodegroup-my-node-group"
2025-10-25 11:54:20 [I] waiting for CloudFormation stack "eksctl-my-cluster123-nodegroup-my-node-group"
2025-10-25 11:56:08 [I] waiting for CloudFormation stack "eksctl-my-cluster123-nodegroup-my-node-group"
2025-10-25 11:56:08 [I] no tasks
2025-10-25 11:56:08 [✓] created 0 nodegroup(s) in cluster "my-cluster123"
2025-10-25 11:56:08 [I] nodegroup "my-node-group" has 3 node(s)
2025-10-25 11:56:08 [I] node "ip-172-31-10-121.ec2.internal" is ready
2025-10-25 11:56:08 [I] node "ip-172-31-35-135.ec2.internal" is ready
2025-10-25 11:56:08 [I] node "ip-172-31-35-82.ec2.internal" is ready
2025-10-25 11:56:08 [I] waiting for at least 2 node(s) to become ready in "my-node-group"
2025-10-25 11:56:08 [I] nodegroup "my-node-group" has 3 node(s)
2025-10-25 11:56:08 [I] node "ip-172-31-10-121.ec2.internal" is ready
2025-10-25 11:56:08 [I] node "ip-172-31-35-135.ec2.internal" is ready
2025-10-25 11:56:08 [I] node "ip-172-31-35-82.ec2.internal" is ready
2025-10-25 11:56:08 [✓] created 1 managed nodegroup(s) in cluster "my-cluster123"
2025-10-25 11:56:08 [I] checking security group configuration for all nodegroups
2025-10-25 11:56:08 [I] all nodegroups have up-to-date cloudformation templates
[root@instance-cluster ~]# vim deploy.yaml
[root@instance-cluster ~]# kubectl apply -f deploy.yaml
pod/devops created
[root@instance-cluster ~]# kubectl get pods
NAME      READY   STATUS    RESTARTS   AGE
devops    1/1     Running   0           11s
[root@instance-cluster ~]#
```

➔ Cluster got created



➔ As you can see we have successfully created the pods on port:80

Test x

Test x

github x

All codes x

Home | N x

Subscrip x

Instances x

Security x

44.202.1 x

+ x

https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Instances: 

Account ID: 8670-9830-6986 ankit.tiwari4@ttimindtree.com

LTIMindtree Favorites Folder Import favorites devops/Donnnn.txt... Sanjaya-K8S-Code/...

aws Search [Alt+S] United States (N. Virginia)

EC2 > Instances

EC2

Dashboard

EC2 Global View

Events

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Capacity Reservations

Capacity Manager

Instances (6) Info

Connect

Instance state

Actions

Launch instances

Find Instance by attribute or tag (case-sensitive)

All states

1

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status
<input type="checkbox"/>	my-cluster123...	i-09a91614c371da6e0	Running	t3.small	3/3 checks passed	View alarms +
<input type="checkbox"/>	my-cluster123...	i-0dd15f916c102ab98	Running	t3.small	3/3 checks passed	View alarms +
<input type="checkbox"/>	cluster-instance	i-04ca65c6823b65158	Running	t3.micro	3/3 checks passed	View alarms +

Select an instance

CloudShell Feedback

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Search

17:32 25-10-2025