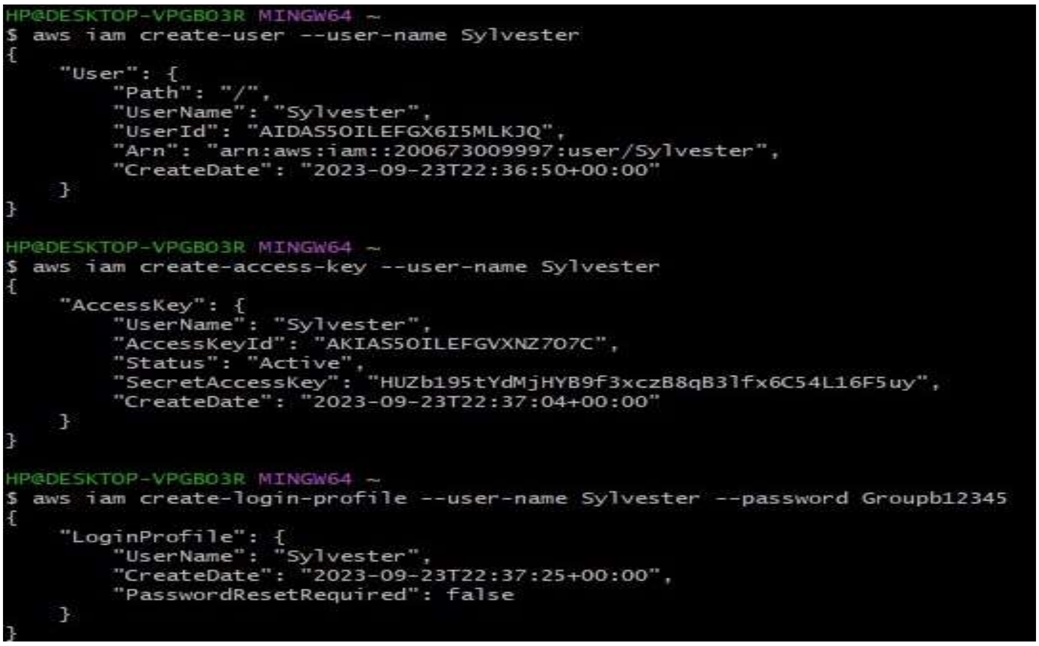
QUIZ: Group leader will create an IAM user for all members of the group, create user and password, Group members Will launch and instance from the IAM user account and group leader will terminate all active instances.



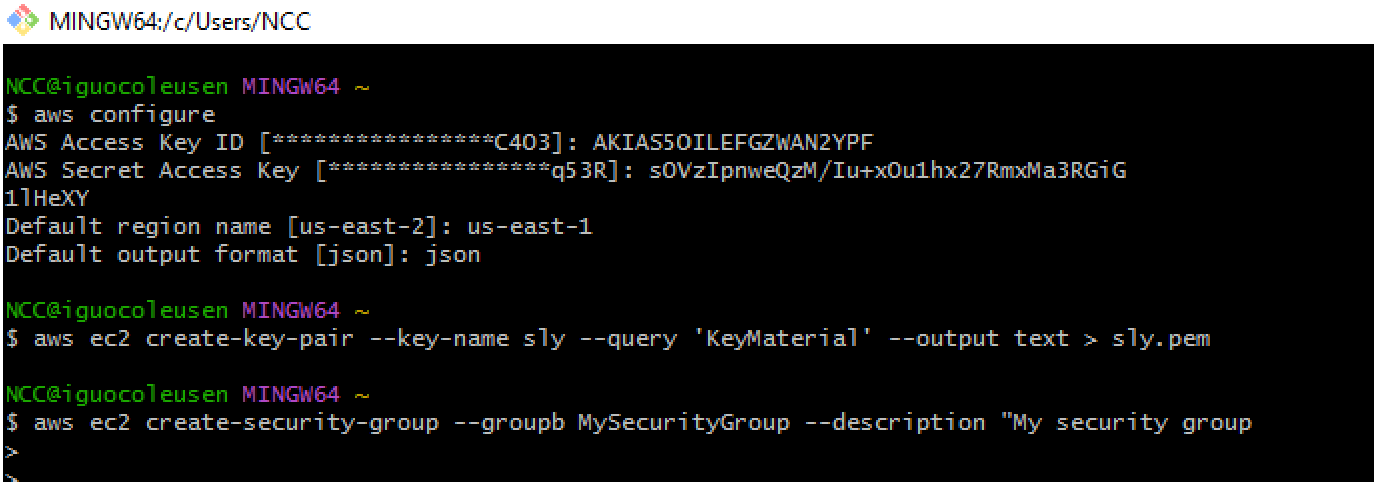
```
HP@DESKTOP-VPGBO3R MINGW64 ~
$ aws iam create-user --user-name Sylvester
{
    "User": {
        "Path": "/",
        "UserName": "Sylvester",
        "UserId": "AIDAS5OILEFGX6I5MLKJQ",
        "Arn": "arn:aws:iam::200673009997:user/Sylvester",
        "CreateDate": "2023-09-23T22:36:50+00:00"
    }
}

HP@DESKTOP-VPGBO3R MINGW64 ~
$ aws iam create-access-key --user-name Sylvester
{
    "AccessKey": {
        "UserName": "Sylvester",
        "AccessKeyId": "AKIAS5OILEFGVXNZ7O7C",
        "Status": "Active",
        "SecretAccessKey": "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",
        "CreateDate": "2023-09-23T22:37:04+00:00"
    }
}

HP@DESKTOP-VPGBO3R MINGW64 ~
$ aws iam create-login-profile --user-name Sylvester --password Groupb12345
{
    "LoginProfile": {
        "UserName": "Sylvester",
        "CreateDate": "2023-09-23T22:37:25+00:00",
        "PasswordResetRequired": false
    }
}
```

Secret key: sOVzIpnweQzM/Iu+xOu1hx27RmxMa3RGiG1lHeXY

Key: AKIAS5OILEFGZWAN2YPF

To configure and establish link and to create key pair

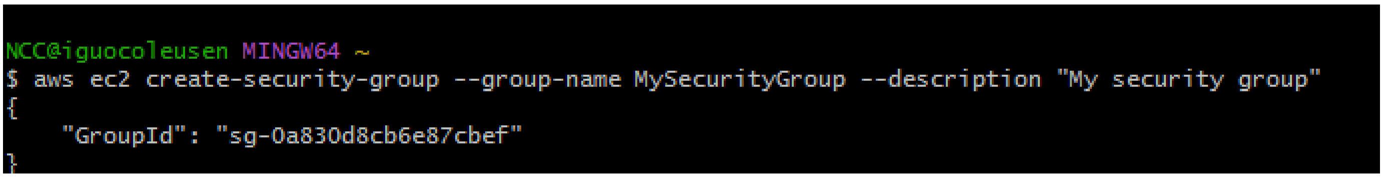
$ aws ec2 create-key-pair --key-name sly --query 'KeyMaterial' --output text > sly.pem



MINGW64:/c/Users/NCC

```
NCC@iguocoleusen MINGW64 ~
$ aws configure
AWS Access Key ID [****************C403]: AKIAS5OILEFGZWAN2YPF
AWS Secret Access Key [****************q53R]: sOVzIpnweQzM/Iu+xOu1hx27RmxMa3RGiG
1lHeXY
Default region name [us-east-2]: us-east-1
Default output format [json]: json

NCC@iguocoleusen MINGW64 ~
$ aws ec2 create-key-pair --key-name sly --query 'KeyMaterial' --output text > sly.pem

NCC@iguocoleusen MINGW64 ~
$ aws ec2 create-security-group --groupb MySecurityGroup --description "My security group
>
>
```

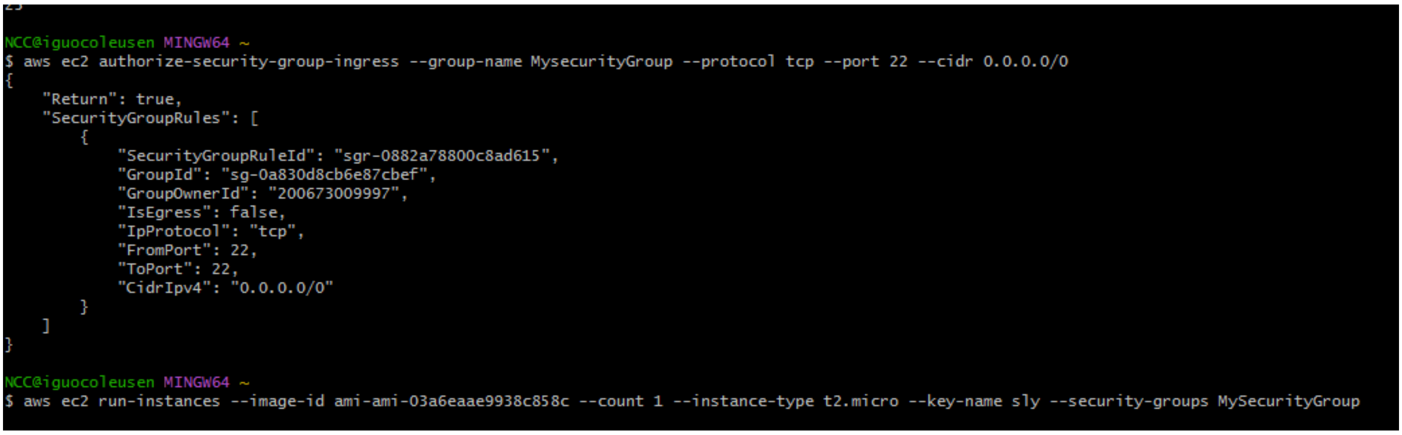
to link security group



```
NCC@iguocoleusen MINGW64 ~
$ aws ec2 create-security-group --group-name MySecurityGroup --description "My security group"
{
    "GroupId": "sg-0a830d8cb6e87cbef"
}
```

to access security group

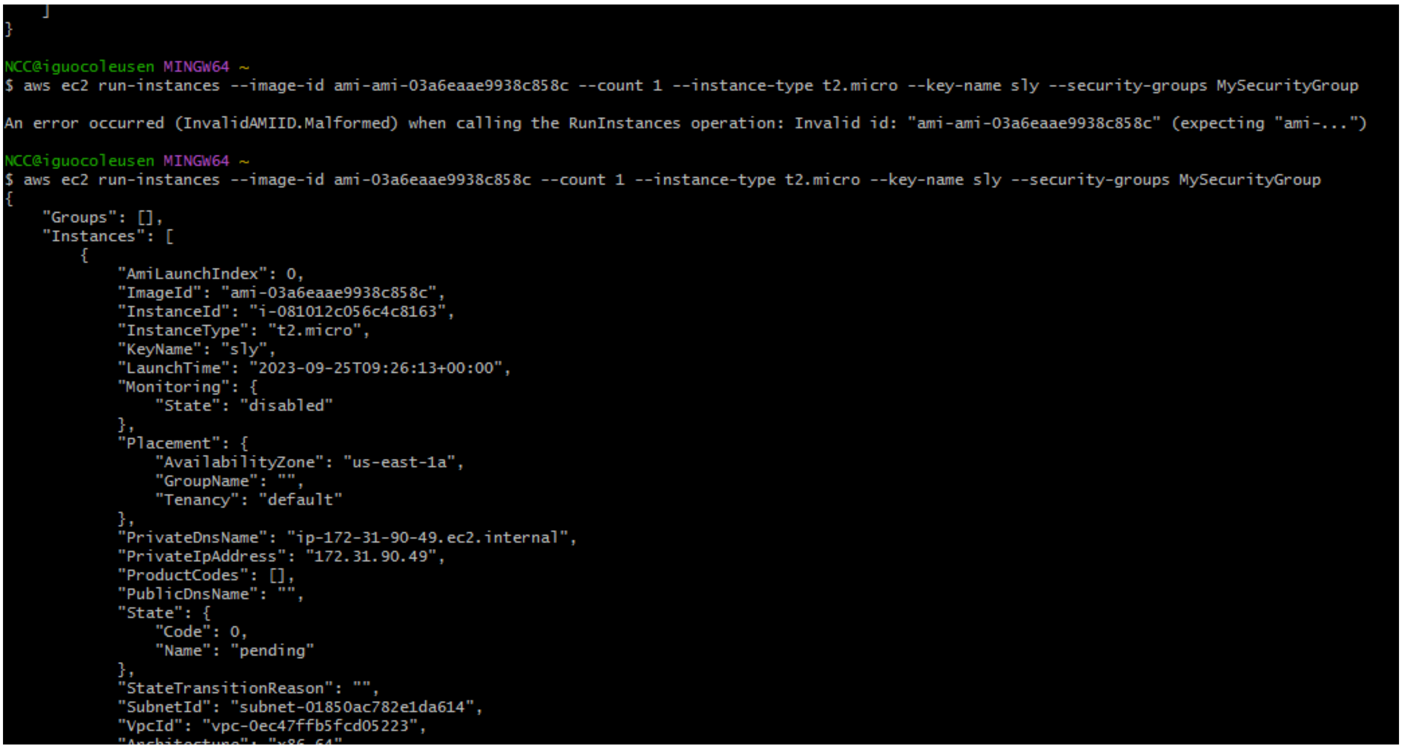
$ aws ec2 authorize-security-group-ingress --group-name MysecurityGroup --protocol tcp --port 22 --cidr 0.0.0.0/0



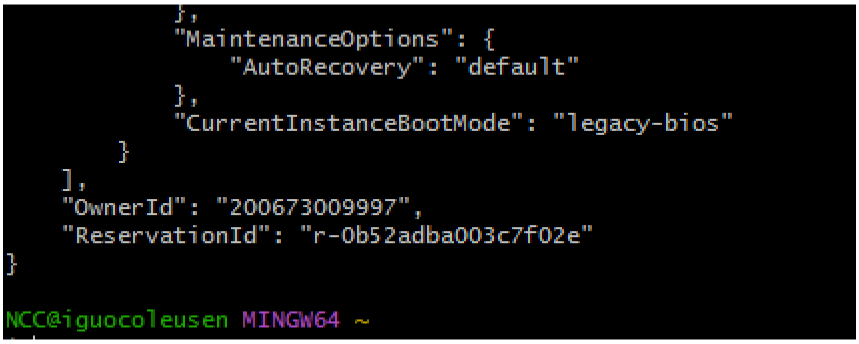
```
NCC@iguocoleusen MINGW64 ~
$ aws ec2 authorize-security-group-ingress --group-name MysecurityGroup --protocol tcp --port 22 --cidr 0.0.0.0/0
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-0882a78800c8ad615",
            "GroupId": "sg-0a830d8cb6e87cbef",
            "GroupOwnerId": "200673009997",
            "IsEgress": false,
            "IpProtocol": "tcp",
            "FromPort": 22,
            "ToPort": 22,
            "CidrIpv4": "0.0.0.0/0"
        }
    ]
}

NCC@iguocoleusen MINGW64 ~
$ aws ec2 run-instances --image-id ami-ami-03a6eaae9938c858c --count 1 --instance-type t2.micro --key-name sly --security-groups MySecurityGroup
```

to run the instance

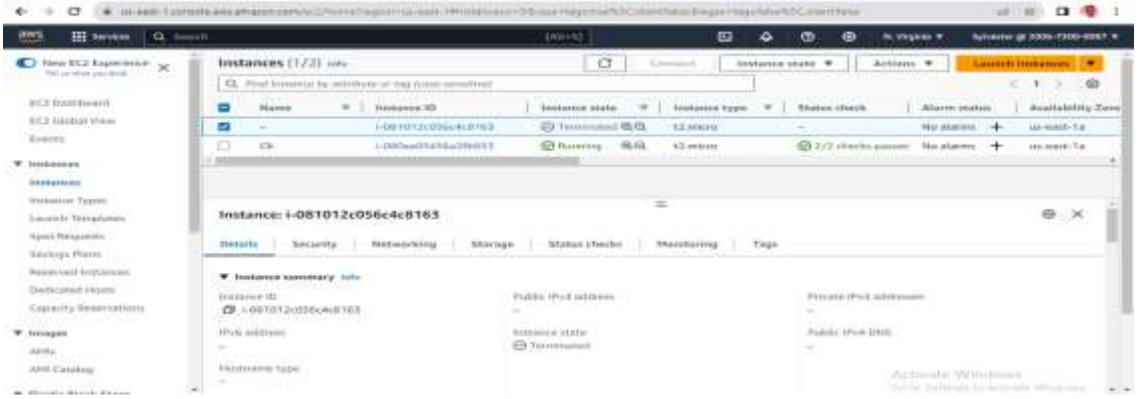
$ aws ec2 run-instances --image-id ami-03a6eaae9938c858c --count 1 --instance-type t2. micro --key-name sly --security-groups MySecurityGroup



```
    ]
}

NCC@iguocoleusen MINGW64 ~
$ aws ec2 run-instances --image-id ami-ami-03a6eaae9938c858c --count 1 --instance-type t2.micro --key-name sly --security-groups MySecurityGroup

An error occurred (InvalidAMIID.Malformed) when calling the RunInstances operation: Invalid id: "ami-ami-03a6eaae9938c858c" (expecting "ami-...")

NCC@iguocoleusen MINGW64 ~
$ aws ec2 run-instances --image-id ami-03a6eaae9938c858c --count 1 --instance-type t2.micro --key-name sly --security-groups MySecurityGroup
{
    "Groups": [],
    "Instances": [
        {
            "AmiLaunchIndex": 0,
            "ImageId": "ami-03a6eaae9938c858c",
            "InstanceId": "i-081012c056c4c8163",
            "InstanceType": "t2.micro",
            "KeyName": "sly",
            "LaunchTime": "2023-09-25T09:26:13+00:00",
            "Monitoring": {
                "State": "disabled"
            },
            "Placement": {
                "AvailabilityZone": "us-east-1a",
                "GroupName": "",
                "Tenancy": "default"
            },
            "PrivateDnsName": "ip-172-31-90-49.ec2.internal",
            "PrivateIpAddress": "172.31.90.49",
            "ProductCodes": [],
            "PublicDnsName": "",
            "State": {
                "Code": 0,
                "Name": "pending"
            },
            "StateTransitionReason": "",
            "SubnetId": "subnet-01850ac782e1da614",
            "VpcId": "vpc-0ec47ffb5fcd05223",
            "Architecture": "x86_64",
```



```
            },
            "MaintenanceOptions": {
                "AutoRecovery": "default"
            },
            "CurrentInstanceBootMode": "legacy-bios"
        }
    ],
    "OwnerId": "200673009997",
    "ReservationId": "r-0b52adba003c7f02e"
}

NCC@iguocoleusen MINGW64 ~
```



Instances (1/2) Info

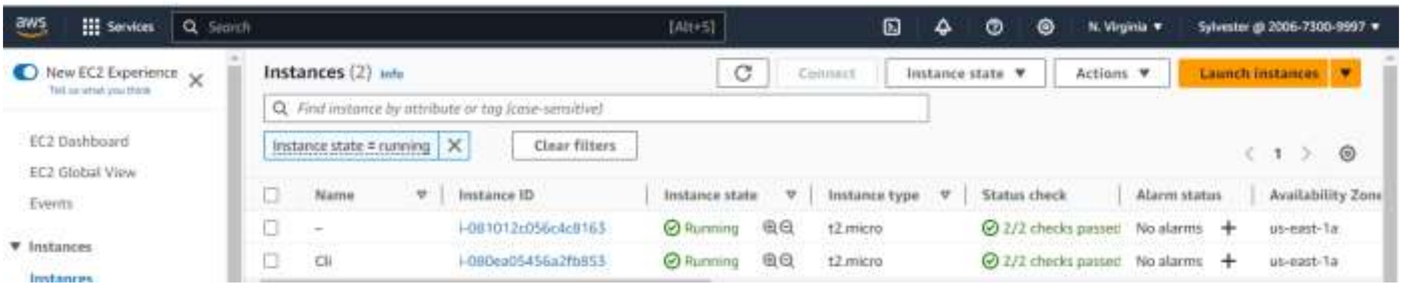
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
–	i-081012c056c4c8163	Terminated	t2.micro	–	No alarms	us-east-1a
Cli	i-080ea05456a2fb853	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a

Instance: i-081012c056c4c8163

Instance summary

Instance ID: i-081012c056c4c8163

Instance state: Terminated



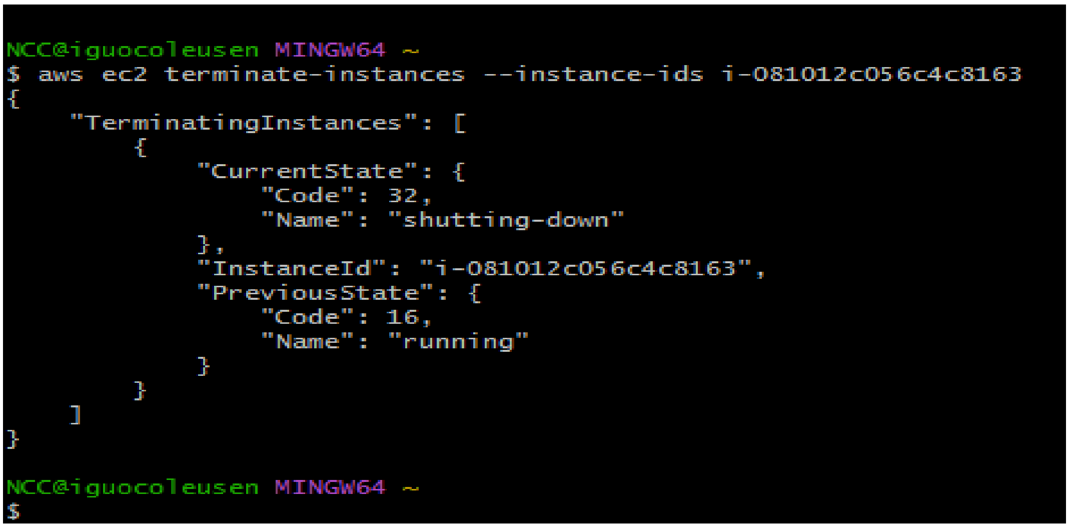
Instances (2) Info

Instance state = running

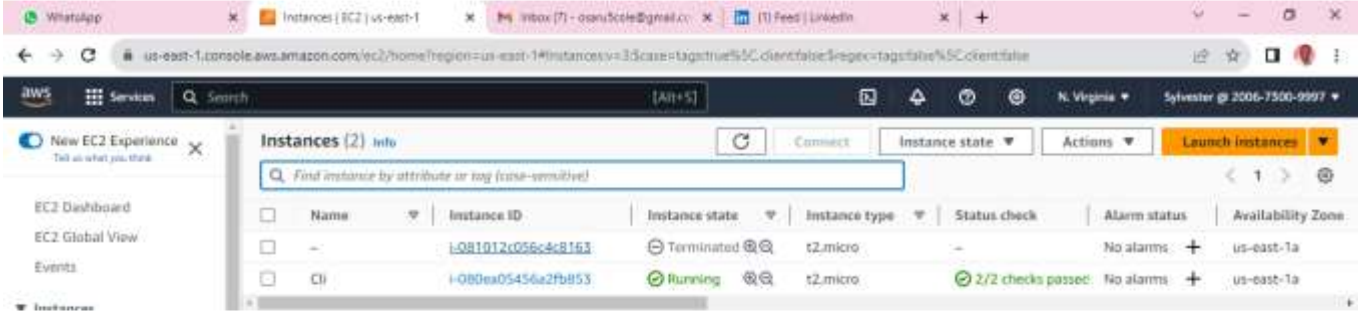
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
–	i-081012c056c4c8163	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a
Cli	i-080ea05456a2fb853	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a

to terminate the instance

$ aws ec2 terminate-instances --instance-ids i-081012c056c4c8163



```
NCC@iguocoleusen MINGW64 ~
$ aws ec2 terminate-instances --instance-ids i-081012c056c4c8163
{
    "TerminatingInstances": [
        {
            "CurrentState": {
                "Code": 32,
                "Name": "shutting-down"
            },
            "InstanceId": "i-081012c056c4c8163",
            "PreviousState": {
                "Code": 16,
                "Name": "running"
            }
        }
    ]
}

NCC@iguocoleusen MINGW64 ~
$
```



us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Instances:v=3;$case=tags:true%5C,client:false;$regex=tags:false%5C,client:false

Instances (2) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
–	i-081012c056c4c8163	Terminated	t2.micro	–	No alarms	us-east-1a
Cli	i-080ea05456a2fb853	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a