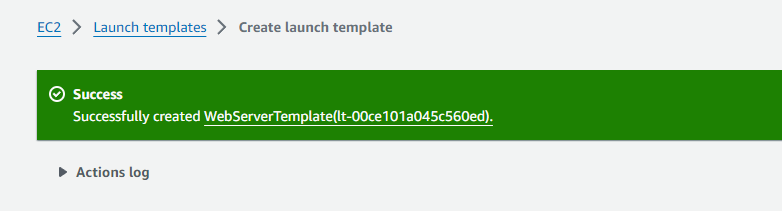
**AWS Hands-On Assignment 04**

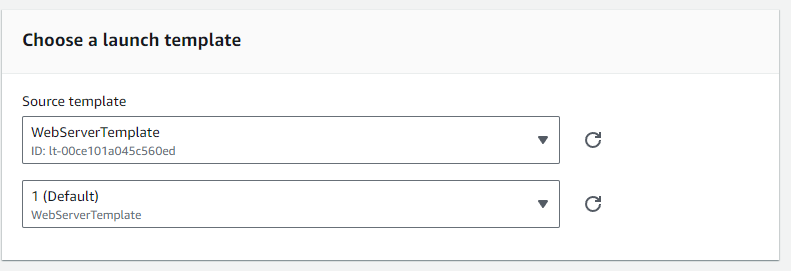
**QUESTION NO: 01**

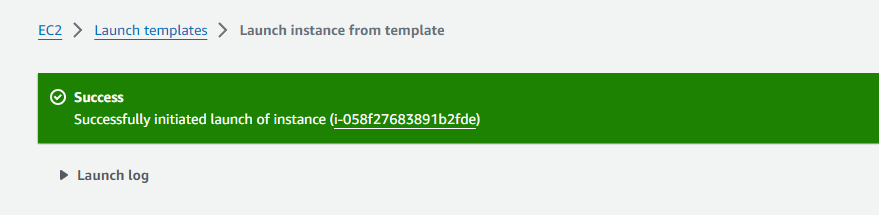
**Console**

1. Create Launch Template on Console:  
   - Navigate to the EC2 dashboard on the AWS Management Console.  
   - Create a launch template named "WebServerTemplate."  
   - Specify configurations such as instance type, key pair, and any additional settings.

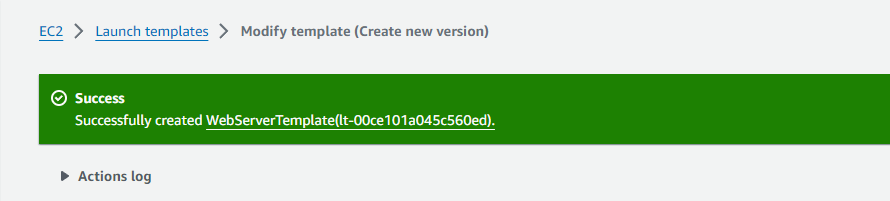


2. Launch Instance Using Launch Template:  
   - Use the launch template "WebServerTemplate" to launch an EC2 instance.  
   - Verify the successful launch of the instance.





3. Modify Launch Template:  
   - Modify the launch template to change the instance type or any other parameter.  
   - Use the modified template to launch another instance.



**CLI**

1. Create Launch Template using AWS CLI:  
   - Use the AWS CLI to create a launch template named "WebServerTemplate" with specified configurations.  
   - Confirm the creation of the launch template.

root@DESKTOP-DA2RDP0:harsh# aws ec2 help

root@DESKTOP-DA2RDP0:harsh# aws ec2 create-launch-template help

root@DESKTOP-DA2RDP0:~# aws ec2 create-launch-template --launch-template-name WebServerTemplate --launch-template-data '{"NetworkInterfaces":[{"AssociatePublicIpAddress":true,"DeviceIndex":0,"SubnetId":"subnet-0be61cf088b9bce21"}],"ImageId":"ami-0005e0cfe09cc9050","InstanceType":"t2.micro","TagSpecifications":[{"ResourceType":"instance","Tags":[{"Key":"purpose","Value":"webserver"}]}]}'

{

"LaunchTemplate": {

"LaunchTemplateId": "lt-08a747af0f3aab4f1",

"LaunchTemplateName": "WebServerTemplate",

"CreateTime": "2024-01-18T08:11:27.000Z",

"CreatedBy": "arn:aws:iam::112365962865:root",

"DefaultVersionNumber": 1,

"LatestVersionNumber": 1

}

}

2. Launch Instance Using Launch Template:  
   - Use the AWS CLI to launch an EC2 instance using the "WebServerTemplate."  
   - Confirm the successful launch of the instance.

root@DESKTOP-DA2RDP0:~# aws ec2 run-instances --launch-template LaunchTemplateName=WebServerTemplate

{

"Groups": [],

"Instances": [

{

"AmiLaunchIndex": 0,

"ImageId": "ami-0005e0cfe09cc9050",

"InstanceId": "i-0a522b808d1ca41d8",

"InstanceType": "t2.micro",

"LaunchTime": "2024-01-18T08:11:40.000Z",

"Monitoring": {

"State": "disabled"

},

"Placement": {

"AvailabilityZone": "us-east-1b",

"GroupName": "",

"Tenancy": "default"

},

"PrivateDnsName": "ip-172-31-80-29.ec2.internal",

"PrivateIpAddress": "172.31.80.29",

"ProductCodes": [],

"PublicDnsName": "",

"State": {

"Code": 0,

"Name": "pending"

},

"StateTransitionReason": "",

"SubnetId": "subnet-0be61cf088b9bce21",

"VpcId": "vpc-06961d095a7c3ded1",

"Architecture": "x86\_64",

"BlockDeviceMappings": [],

"ClientToken": "14cb6f96-10a4-420f-9a0b-e8adb2da701b",

"EbsOptimized": false,

"EnaSupport": true,

"Hypervisor": "xen",

"NetworkInterfaces": [

{

"Attachment": {

"AttachTime": "2024-01-18T08:11:40.000Z",

"AttachmentId": "eni-attach-0e5958a74fc4cedb9",

"DeleteOnTermination": true,

"DeviceIndex": 0,

"Status": "attaching",

"NetworkCardIndex": 0

},

"Description": "",

"Groups": [

{

"GroupName": "default",

"GroupId": "sg-0161ea3faea7e18cc"

}

],

"Ipv6Addresses": [],

"MacAddress": "12:ea:40:6c:ef:df",

"NetworkInterfaceId": "eni-0501402659b5062bc",

"OwnerId": "112365962865",

"PrivateDnsName": "ip-172-31-80-29.ec2.internal",

"PrivateIpAddress": "172.31.80.29",

"PrivateIpAddresses": [

{

"Primary": true,

"PrivateDnsName": "ip-172-31-80-29.ec2.internal",

"PrivateIpAddress": "172.31.80.29"

}

],

"SourceDestCheck": true,

"Status": "in-use",

"SubnetId": "subnet-0be61cf088b9bce21",

"VpcId": "vpc-06961d095a7c3ded1",

"InterfaceType": "interface"

}

],

"RootDeviceName": "/dev/xvda",

"RootDeviceType": "ebs",

"SecurityGroups": [

{

"GroupName": "default",

"GroupId": "sg-0161ea3faea7e18cc"

}

],

"SourceDestCheck": true,

"StateReason": {

"Code": "pending",

"Message": "pending"

},

"Tags": [

{

"Key": "purpose",

"Value": "webserver"

},

{

"Key": "aws:ec2launchtemplate:id",

"Value": "lt-08a747af0f3aab4f1"

},

{

"Key": "aws:ec2launchtemplate:version",

"Value": "1"

}

],

"VirtualizationType": "hvm",

"CpuOptions": {

"CoreCount": 1,

"ThreadsPerCore": 1

},

"CapacityReservationSpecification": {

"CapacityReservationPreference": "open"

},

"MetadataOptions": {

"State": "pending",

"HttpTokens": "required",

"HttpPutResponseHopLimit": 2,

"HttpEndpoint": "enabled",

"HttpProtocolIpv6": "disabled",

"InstanceMetadataTags": "disabled"

},

"EnclaveOptions": {

"Enabled": false

},

"BootMode": "uefi-preferred",

"PrivateDnsNameOptions": {

"HostnameType": "ip-name",

"EnableResourceNameDnsARecord": false,

"EnableResourceNameDnsAAAARecord": false

}

}

],

"OwnerId": "112365962865",

"ReservationId": "r-0ad510ea33344a726"

}

root@DESKTOP-DA2RDP0:~#

root@DESKTOP-DA2RDP0:~#

3. Modify Launch Template using AWS CLI:  
   - Use the AWS CLI to modify the launch template, e.g., change the instance type.  
   - Use the modified template to launch another instance.

root@DESKTOP-DA2RDP0:~# aws ec2 create-launch-template-version --launch-template-name WebServerTemplate --version-description "my vesion description" --source-version 1 --launch-template-data '{"InstanceType": "t2.medium"}'

{

"LaunchTemplateVersion": {

"LaunchTemplateId": "lt-08a747af0f3aab4f1",

"LaunchTemplateName": "WebServerTemplate",

"VersionNumber": 2,

"VersionDescription": "my vesion description",

"CreateTime": "2024-01-18T08:28:14.000Z",

"CreatedBy": "arn:aws:iam::112365962865:root",

"DefaultVersion": false,

"LaunchTemplateData": {

"NetworkInterfaces": [

{

"AssociatePublicIpAddress": true,

"DeviceIndex": 0,

"SubnetId": "subnet-0be61cf088b9bce21"

}

],

"ImageId": "ami-0005e0cfe09cc9050",

"InstanceType": "t2.medium",

"TagSpecifications": [

{

"ResourceType": "instance",

"Tags": [

{

"Key": "purpose",

"Value": "webserver"

}

]

}

]

}

}

}

root@DESKTOP-DA2RDP0:~# ^C

root@DESKTOP-DA2RDP0:~# aws ec2 run-instances --launch-template LaunchTemplateName=WebServerTemplate,Version=2

{

"Groups": [],

"Instances": [

{

"AmiLaunchIndex": 0,

"ImageId": "ami-0005e0cfe09cc9050",

"InstanceId": "i-01ad4c0f37c71ceb0",

"InstanceType": "t2.medium",

"LaunchTime": "2024-01-18T08:29:23.000Z",

"Monitoring": {

"State": "disabled"

},

"Placement": {

"AvailabilityZone": "us-east-1b",

"GroupName": "",

"Tenancy": "default"

},

"PrivateDnsName": "ip-172-31-86-47.ec2.internal",

"PrivateIpAddress": "172.31.86.47",

"ProductCodes": [],

"PublicDnsName": "",

"State": {

"Code": 0,

"Name": "pending"

},

"StateTransitionReason": "",

"SubnetId": "subnet-0be61cf088b9bce21",

"VpcId": "vpc-06961d095a7c3ded1",

"Architecture": "x86\_64",

"BlockDeviceMappings": [],

"ClientToken": "03dade65-a127-4f25-9d76-b50f320099f2",

"EbsOptimized": false,

"EnaSupport": true,

"Hypervisor": "xen",

"NetworkInterfaces": [

{

"Attachment": {

"AttachTime": "2024-01-18T08:29:23.000Z",

"AttachmentId": "eni-attach-0d20a445edd466aa1",

"DeleteOnTermination": true,

"DeviceIndex": 0,

"Status": "attaching",

"NetworkCardIndex": 0

},

"Description": "",

"Groups": [

{

"GroupName": "default",

"GroupId": "sg-0161ea3faea7e18cc"

}

],

"Ipv6Addresses": [],

"MacAddress": "12:31:3a:46:5e:a7",

"NetworkInterfaceId": "eni-036411396ee11f200",

"OwnerId": "112365962865",

"PrivateDnsName": "ip-172-31-86-47.ec2.internal",

"PrivateIpAddress": "172.31.86.47",

"PrivateIpAddresses": [

{

"Primary": true,

"PrivateDnsName": "ip-172-31-86-47.ec2.internal",

"PrivateIpAddress": "172.31.86.47"

}

],

"SourceDestCheck": true,

"Status": "in-use",

"SubnetId": "subnet-0be61cf088b9bce21",

"VpcId": "vpc-06961d095a7c3ded1",

"InterfaceType": "interface"

}

],

"RootDeviceName": "/dev/xvda",

"RootDeviceType": "ebs",

"SecurityGroups": [

{

"GroupName": "default",

"GroupId": "sg-0161ea3faea7e18cc"

}

],

"SourceDestCheck": true,

"StateReason": {

"Code": "pending",

"Message": "pending"

},

"Tags": [

{

"Key": "aws:ec2launchtemplate:id",

"Value": "lt-08a747af0f3aab4f1"

},

{

"Key": "purpose",

"Value": "webserver"

},

{

"Key": "aws:ec2launchtemplate:version",

"Value": "2"

}

],

"VirtualizationType": "hvm",

"CpuOptions": {

"CoreCount": 2,

"ThreadsPerCore": 1

},

"CapacityReservationSpecification": {

"CapacityReservationPreference": "open"

},

"MetadataOptions": {

"State": "pending",

"HttpTokens": "required",

"HttpPutResponseHopLimit": 2,

"HttpEndpoint": "enabled",

"HttpProtocolIpv6": "disabled",

"InstanceMetadataTags": "disabled"

},

"EnclaveOptions": {

"Enabled": false

},

"BootMode": "uefi-preferred",

"PrivateDnsNameOptions": {

"HostnameType": "ip-name",

"EnableResourceNameDnsARecord": false,

"EnableResourceNameDnsAAAARecord": false

}

}

],

"OwnerId": "112365962865",

"ReservationId": "r-0cd9cd36124ba01b3"

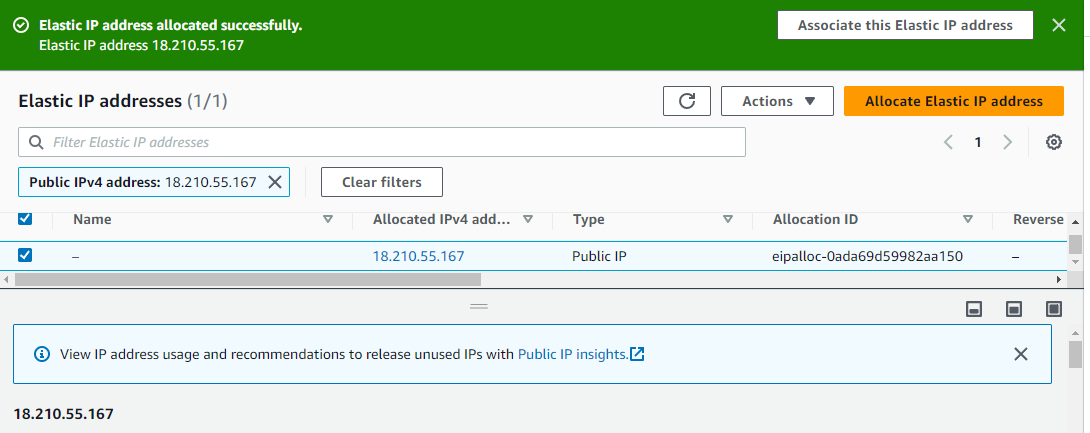
}

root@DESKTOP-DA2RDP0:~#

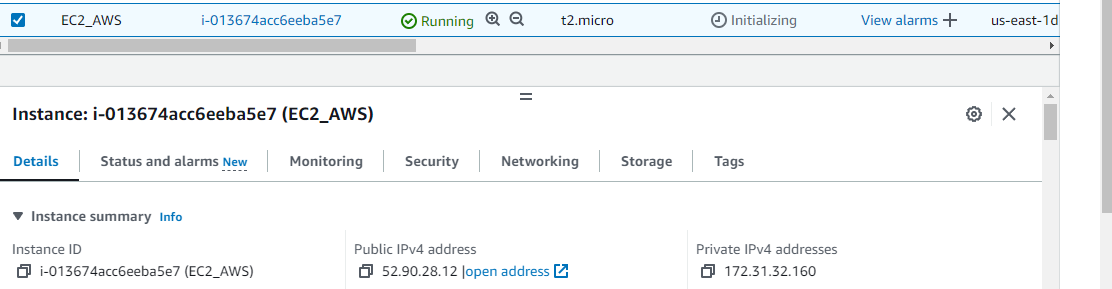
**QUESTION NO: 02**

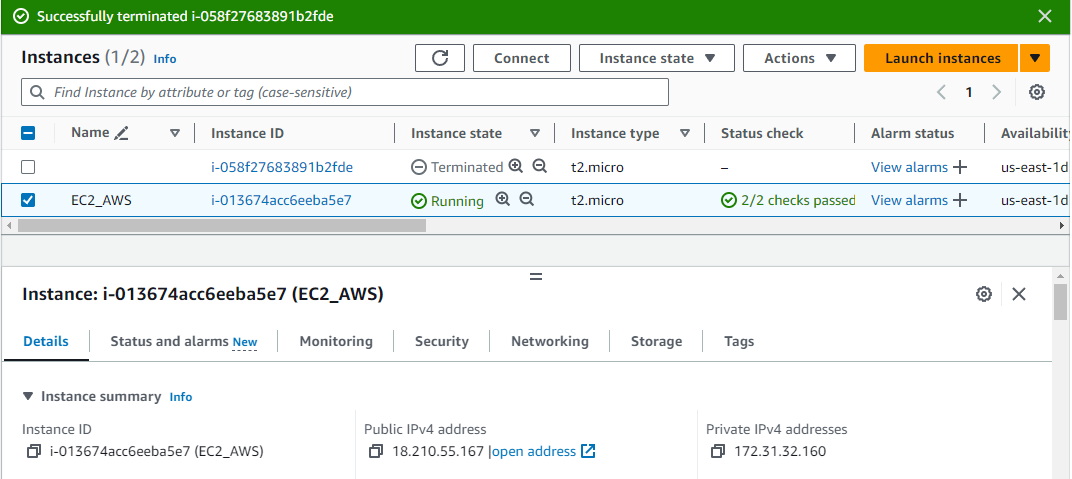
**Console**

1. Allocate Elastic IP and Associate:  
   - Using the AWS Management Console, allocate an Elastic IP address.  
   - Associate the Elastic IP with an existing running EC2 instance.

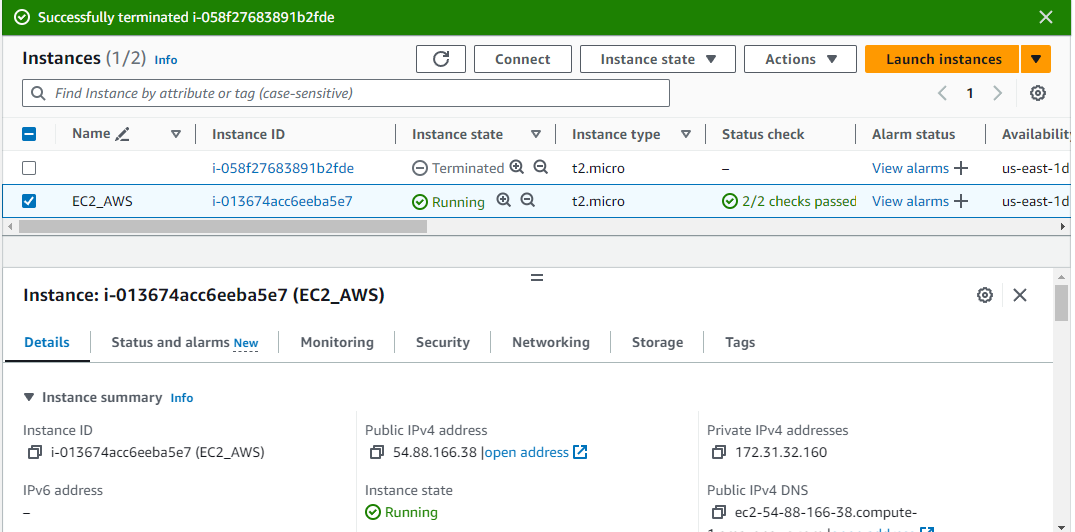


2. Verify Elastic IP Functionality:  
   - Confirm the functionality of the Elastic IP by accessing the associated EC2 instance.  
   - Document any observations or considerations related to Elastic IP usage.





3. Swap Elastic IPs:  
   - Allocate another Elastic IP and swap it with the original Elastic IP.  
   - Document the steps taken and verify the new Elastic IP functionality.



**CLI**

1. Allocate Elastic IP and Associate using AWS CLI:  
   - Use the AWS CLI to allocate an Elastic IP address.  
   - Associate the Elastic IP with an existing running EC2 instance.

root@DESKTOP-DA2RDP0:~# aws ec2 allocate-address --domain vpc

{

"PublicIp": "52.6.131.144",

"AllocationId": "eipalloc-0670f1c2ec7fbb2d5",

"PublicIpv4Pool": "amazon",

"NetworkBorderGroup": "us-east-1",

"Domain": "vpc"

}

root@DESKTOP-DA2RDP0:~#

root@DESKTOP-DA2RDP0:~# aws ec2 associate-address --allocation-id eipalloc-0670f1c2ec7fbb2d5 --instance-id i-01ad4c0f37c71ceb0

{

"AssociationId": "eipassoc-0c790e14eb4588cd8"

}

root@DESKTOP-DA2RDP0:~#

2. Verify Elastic IP Functionality using AWS CLI:  
   - Use the AWS CLI to confirm the functionality of the Elastic IP by accessing the associated EC2 instance.  
   - Document any observations or considerations related to Elastic IP usage.

99.128

root@DESKTOP-DA2RDP0:~# aws ec2 describe-instances --instance-ids i-01ad4c0f37c71ceb0 --query 'Reservations[\*].Instances[\*].PublicIpAddress' --output json

[

[

"52.6.131.144"

]

]

root@DESKTOP-DA2RDP0:~#

3. Swap Elastic IPs using AWS CLI:  
   - Use the AWS CLI to allocate another Elastic IP.

root@DESKTOP-DA2RDP0:~# aws ec2 allocate-address --domain vpc --output json

{

"PublicIp": "52.0.83.67",

"AllocationId": "eipalloc-099555d7d51fa2601",

"PublicIpv4Pool": "amazon",

"NetworkBorderGroup": "us-east-1",

"Domain": "vpc"

}

root@DESKTOP-DA2RDP0:~# aws ec2 describe-addresses

{

"Addresses": [

{

"PublicIp": "34.195.211.112",

"AllocationId": "eipalloc-05a364eac49ae3828",

"Domain": "vpc",

"PublicIpv4Pool": "amazon",

"NetworkBorderGroup": "us-east-1"

},

{

"PublicIp": "52.0.83.67",

"AllocationId": "eipalloc-099555d7d51fa2601",

"Domain": "vpc",

"PublicIpv4Pool": "amazon",

"NetworkBorderGroup": "us-east-1"

},

{

"InstanceId": "i-01ad4c0f37c71ceb0",

"PublicIp": "52.6.131.144",

"AllocationId": "eipalloc-0670f1c2ec7fbb2d5",

"AssociationId": "eipassoc-0c790e14eb4588cd8",

"Domain": "vpc",

"NetworkInterfaceId": "eni-036411396ee11f200",

"NetworkInterfaceOwnerId": "112365962865",

"PrivateIpAddress": "172.31.86.47",

"PublicIpv4Pool": "amazon",

"NetworkBorderGroup": "us-east-1"

}

]

}

root@DESKTOP-DA2RDP0:~# aws ec2 disassociate-address --association-id eipassoc-0c790e14eb4588cd8

   - Swap the newly allocated Elastic IP with the original one.

root@DESKTOP-DA2RDP0:~# aws ec2 associate-address --allocation-id eipalloc-0670f1c2ec7fbb2d5 --instance-id i-0a522b808d1ca41d8

{

"AssociationId": "eipassoc-0a132761a3a74b16a"

}

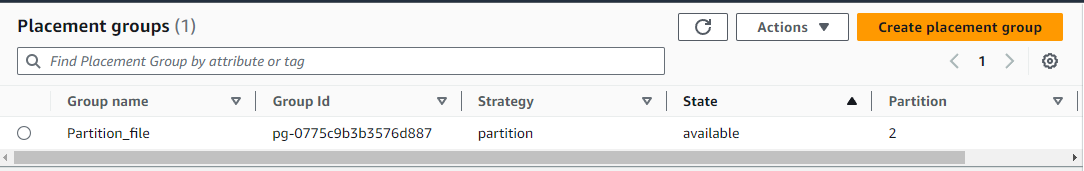
root@DESKTOP-DA2RDP0:~#

  - Document the steps taken and verify the new Elastic IP functionality.

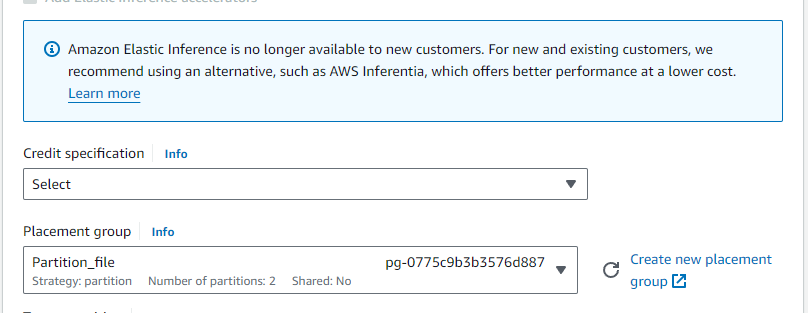
**QUESTION NO: 03**

**Console**

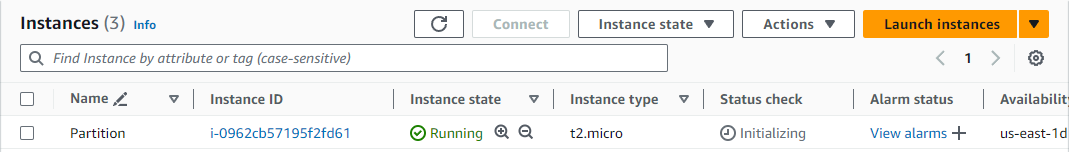
1. Create Partition Placement Group:  
   - Using the AWS Management Console, create a "Partition" placement group.  
   - Ensure it is associated with a specific region.

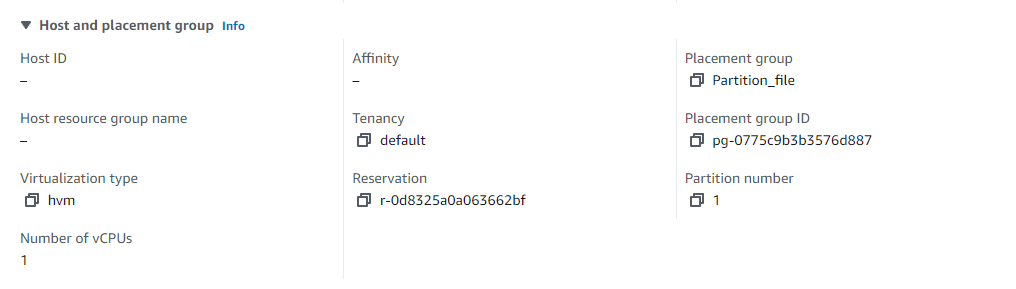


2. Launch Instances into Partition Placement Group:  
   - Launch multiple EC2 instances into the created "Partition" placement group with distinct partition numbers.  
   - Confirm that instances are distributed across partitions.



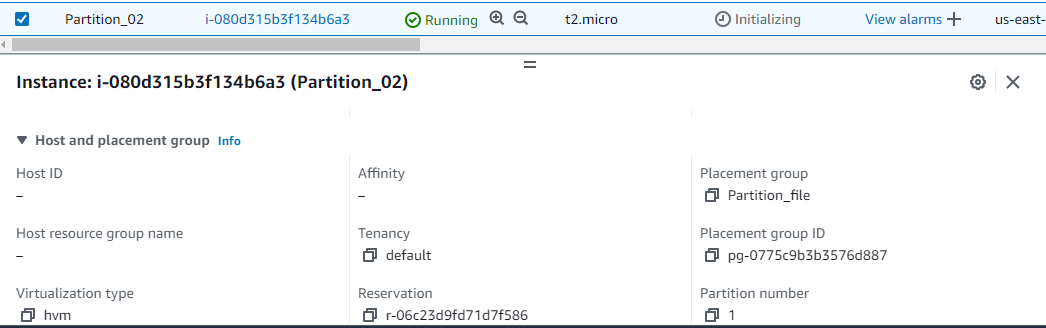
Creating New Instance of Partition



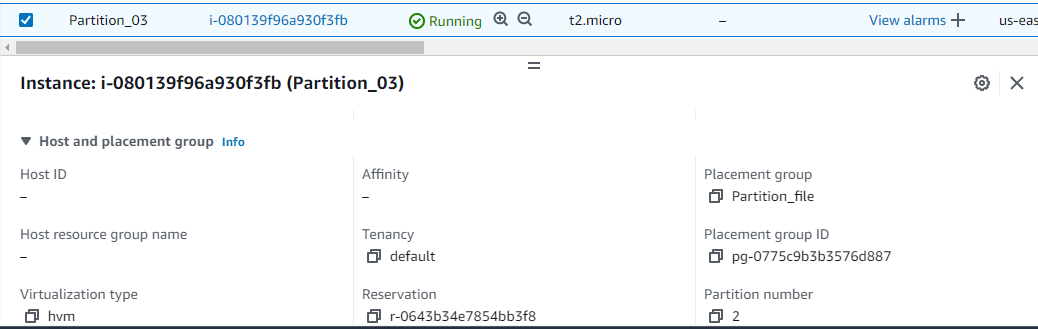




Created new partition\_02 instance

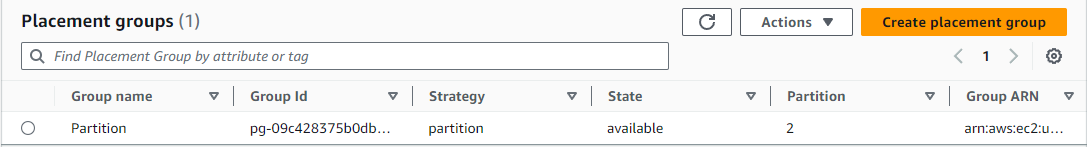


Created new partition\_03 instance

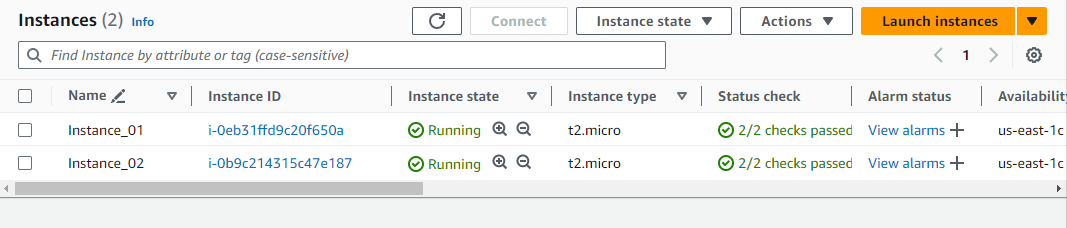


3. Test Isolation:  
   - Use the console to observe the network and resource isolation between instances in different partitions.  
   - Verify that instances in one partition do not share the underlying hardware with instances in other partitions.

Create placement groups == Partition



Creating 2 instances and add partition 1 for Instance\_01 2 for Instance\_02.



 root@DESKTOP-DA2RDP0:DOCKER\_TASK# aws ec2 describe-instances --filters Name=placement-group-name,Values=Partition

"Placement": {

"AvailabilityZone": "us-east-1c",

"GroupName": "Partition",

"PartitionNumber": 1,

"Tenancy": "default"

},

"Placement": {

"AvailabilityZone": "us-east-1c",

"GroupName": "Partition",

"PartitionNumber": 2,

"Tenancy": "default"

},