Creating the second Node Group

Create a second managed nodegroup using SPOT instances

```
1
cd ~/environment/tfekscode/extra/nodeg2
```

Initialize Terraform:

```
1
terraform init
```

Validate the Terraform code:

```
1
terraform validate
```

Plan the deployment:

```
1
terraform plan -out tfplan
data.aws_ssm_parameter.allnodes-sg: Reading...
data.aws_ssm_parameter.cicd-cidr: Reading...
....
data.aws_caller_identity.current: Read complete after 0s [id=440018911661]
data.aws_ssm_parameter.eksami: Read complete after 0s [id=/aws/service/eks/optimized-ami/1.24/amazon-linux-2/recommended/image_id]


Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  + create


Terraform will perform the following actions:


  # aws_eks_node_group.ng2 will be created
  + resource "aws_eks_node_group" "ng2" {
      + ami_type             = (known after apply)
```

```
+ arn                  = (known after apply)
+ capacity_type        = "SPOT"
+ cluster_name         = "mycluster1"
+ disk_size            = 0
+ id                   = (known after apply)
+ instance_types       = [
    + "m5.large",
    + "m5a.large",
    + "m5d.large",
    + "m5ad.large",
  ]
+ labels               = {
    + "eks/cluster-name"   = "mycluster1"
    + "eks/nodegroup-name" = "ng2-mycluster1"
  }
+ node_group_name       = "ng2-mycluster1"
+ node_group_name_prefix = (known after apply)
+ node_role_arn         = (sensitive value)
+ release_version       = (known after apply)
+ resources            = (known after apply)
+ status               = (known after apply)
+ subnet_ids           = (sensitive value)
+ tags                 = {
    + "eks/cluster-name"   = "mycluster1"
    + "eks/nodegroup-name" = "ng2-mycluster1"
    + "eks/nodegroup-type" = "managed"
    + "eksnet"           = "net-main"
  }
+ tags_all             = {
    + "eks/cluster-name"   = "mycluster1"
    + "eks/nodegroup-name" = "ng2-mycluster1"
    + "eks/nodegroup-type" = "managed"
    + "eksnet"           = "net-main"
  }
+ version              = (known after apply)
```

```
    + launch_template {
        + id      = (known after apply)
        + name    = "at-lt-mycluster1-ng2"
        + version = "1"
      }

    + scaling_config {
        + desired_size = 2
        + max_size     = 3
        + min_size     = 1
      }

    + timeouts {}
  }

  # aws_launch_template.lt-ng2 will be created
  + resource "aws_launch_template" "lt-ng2" {
    + arn                    = (known after apply)

....

    + tag_specifications {
        + resource_type = "instance"
        + tags          = {
            + "Name" = "mycluster1-ng2"
          }
      }
  }

Plan: 2 to add, 0 to change, 0 to destroy.

Changes to Outputs:
  + config-map-aws-auth = "local.config-map-aws-auth"
  + kubeconfig          = "local.kubeconfig"
```

Saved the plan to: tfplan

To perform exactly these acti

You can see from the plan the following resources will be created

- A Launch template
- A NodeGroup using the launch template above

Build the environment:

1
terraform apply tfplan
aws_launch_template.lt-ng2: Creating...
aws_launch_template.lt-ng2: Creation complete after 0s [id=lt-000589a487be493c7]
aws_eks_node_group.ng2: Creating...
aws_eks_node_group.ng2: Still creating... [10s elapsed]
aws_eks_node_group.ng2: Still creating... [20s elapsed]
aws_eks_node_group.ng2: Still creating... [30s elapsed]
aws_eks_node_group.ng2: Still creating... [40s elapsed]
aws_eks_node_group.ng2: Still creating... [50s elapsed]
aws_eks_node_group.ng2: Still creating... [1m0s elapsed]
aws_eks_node_group.ng2: Still creating... [1m10s elapsed]
aws_eks_node_group.ng2: Still creating... [1m20s elapsed]
aws_eks_node_group.ng2: Still creating... [1m30s elapsed]
aws_eks_node_group.ng2: Creation complete after 1m38s [id=mycluster1:ng2-mycluster1]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.

Outputs:

config-map-aws-auth = "local.config-map-aws-auth"

kubeconfig = "local.kubeconfig"

We should now have 4 kubernetes worker nodes

1

kubectl get nodes

```
NAME                                          STATUS   ROLES    AGE    VERSION
ip-10-0-1-19.eu-west-1.compute.internal    Ready   <none>   145m   v1.24.11-eks-a59e1f0
ip-10-0-1-248.eu-west-1.compute.internal   Ready   <none>   2m23s   v1.24.11-eks-a59e1f0
ip-10-0-2-153.eu-west-1.compute.internal   Ready   <none>   146m   v1.24.11-eks-a59e1f0
ip-10-0-2-235.eu-west-1.compute.internal   Ready   <none>   2m23s   v1.24.11-eks-a59e1f0
```