```
jees-macbook:zone2 jeesantony$ ls -ltr
total 64
-rw-r--r--  1 jeesantony  staff   265 Dec  6 10:13 _config.tf
-rw-r--r--  1 jeesantony  staff   234 Dec  6 10:13 _output.tf
-rw-r--r--  1 jeesantony  staff   835 Dec  6 10:13 ec2.tf
-rw-r--r--  1 jeesantony  staff  1160 Dec  6 10:13 eks.tf
drwxr-xr-x  5 jeesantony  staff   160 Dec  6 10:13 modules
-rw-r--r--  1 jeesantony  staff   666 Dec  6 10:13 project.tf
-rw-r--r--  1 jeesantony  staff    46 Dec 17 20:05 _var.tf
-rw-r--r--  1 jeesantony  staff   983 Dec 17 20:05 main.tf
-rw-r--r--  1 jeesantony  staff   805 Dec 18 05:35 _data.tf
jees-macbook:zone2 jeesantony$ terraform init
Initializing the backend...
Initializing modules...
Initializing provider plugins...
- terraform.io/builtin/terraform is built in to Terraform
- Reusing previous version of hashicorp/kubernetes from the
dependency lock file
- Reusing previous version of hashicorp/aws from the dependency
lock file
- Using previously-installed hashicorp/kubernetes v2.24.0
- Using previously-installed hashicorp/aws v5.31.0
Terraform has been successfully initialized!
You may now begin working with Terraform. Try running
"terraform plan" to see
any changes that are required for your infrastructure. All Terraform
commands
should now work.
If you ever set or change modules or backend configuration for
Terraform,
rerun this command to reinitialize your working directory. If you
forget, other
```

```
commands will detect it and remind you to do so if necessary.
jees-macbook:zone2 jeesantony$ terraform apply
data.terraform_remote_state.vpc: Reading...
data.aws_caller_identity.current: Reading...
data.aws_iam_policy.cloudwatch-policy: Reading...
data.aws_ami.amazon_linux_2: Reading...
data.aws_iam_role.eks_node_role: Reading...
data.aws_iam_policy.instance-policy: Reading...
data.aws_iam_role.eks_cluster_role: Reading...
module.project_eks.data.aws_iam_policy_document.eks_assume_role_policy: Reading...
module.project_eks.data.aws_iam_policy_document.eks_node_assume_role_policy: Reading...
module.project_eks.data.aws_iam_policy_document.eks_assume_role_policy: Read complete after 0s [id=3552664922]
module.project_eks.data.aws_iam_policy_document.eks_node_assume_role_policy: Read complete after 0s [id=2851119427]
data.aws_caller_identity.current: Read complete after 1s [id=810716384126]
data.aws_iam_role.eks_node_role: Read complete after 1s [id=app-udacity-eks-node-role]
data.aws_iam_role.eks_cluster_role: Read complete after 1s [id=app-udacity-eks-cluster-role]
data.aws_ami.amazon_linux_2: Read complete after 2s [id=ami-0967795d5c824c5da]
data.terraform_remote_state.vpc: Read complete after 3s
data.aws_iam_policy.cloudwatch-policy: Still reading... [10s elapsed]
data.aws_iam_policy.instance-policy: Still reading... [10s elapsed]
data.aws_iam_policy.cloudwatch-policy: Read complete after 13s
```

```
[id=arn:aws:iam::810716384126:policy/app-udacity-eks-cluster-role-cloudwatch-policy]
data.aws_iam_policy.instance-policy: Read complete after 13s
[id=arn:aws:iam::810716384126:policy/app-udacity-instance-policy]
Terraform used the selected providers to generate the following
execution plan. Resource actions are
indicated with the following symbols:
  + create
 <= read (data resources)
Terraform will perform the following actions:
  # data.aws_eks_cluster.cluster will be read during apply
  # (config refers to values not yet known)
 <= data "aws_eks_cluster" "cluster" {
      + arn                       = (known after apply)
      + certificate_authority     = (known after apply)
      + cluster_id                = (known after apply)
      + created_at                = (known after apply)
      + enabled_cluster_log_types = (known after apply)
      + endpoint                  = (known after apply)
      + id                        = (known after apply)
      + identity                  = (known after apply)
      + kubernetes_network_config = (known after apply)
      + name                      = (known after apply)
      + outpost_config            = (known after apply)
      + platform_version          = (known after apply)
      + role_arn                  = (known after apply)
      + status                    = (known after apply)
      + tags                      = (known after apply)
      + version                   = (known after apply)
      + vpc_config                = (known after apply)
    }
 # data.aws_eks_cluster_auth.cluster will be read during apply
 # (config refers to values not yet known)
<= data "aws_eks_cluster_auth" "cluster" {
    + id    = (known after apply)
    + name  = (known after apply)
    + token = (sensitive value)
  }
 # kubernetes_namespace.udacity will be created
 + resource "kubernetes_namespace" "udacity" {
    + id                              = (known after apply)
    + wait_for_default_service_account = false
    + metadata {
        + generation       = (known after apply)
        + name             = "udacity"
        + resource_version = (known after apply)
        + uid              = (known after apply)
      }
  }
 # kubernetes_service.grafana-external will be created
 + resource "kubernetes_service" "grafana-external" {
    + id                    = (known after apply)
    + status                = (known after apply)
    + wait_for_load_balancer = true
    + metadata {
        + annotations      = {
            +
"service.beta.kubernetes.io/aws-load-balancer-nlb-target-type" =
"ip"
            + "service.beta.kubernetes.io/aws-load-balancer-type"
= "nlb"
          }
        + generation       = (known after apply)
      + name             = "grafana-external"
      + namespace        = "monitoring"
      + resource_version = (known after apply)
      + uid              = (known after apply)
    }
  + spec {
      + allocate_load_balancer_node_ports = true
      + cluster_ip                        = (known after apply)
      + cluster_ips                       = (known after apply)
      + external_traffic_policy           = (known after apply)
      + health_check_node_port            = (known after apply)
      + internal_traffic_policy           = (known after apply)
      + ip_families                       = (known after apply)
      + ip_family_policy                  = (known after apply)
      + publish_not_ready_addresses       = false
      + selector                          = {
          + "app.kubernetes.io/name" = "grafana"
        }
      + session_affinity                  = "None"
      + type                              = "LoadBalancer"
      + port {
          + node_port   = (known after apply)
          + port        = 80
          + protocol    = "TCP"
          + target_port = "3000"
        }
    }
}
# module.project_ec2.aws_instance.ubuntu[0] will be created
+ resource "aws_instance" "ubuntu" {
  + ami                                  = "ami-063d2f012ccad1ebd"
  + arn                                  = (known after apply)
```

```
      + associate_public_ip_address          = (known after apply)
      + availability_zone                    = (known after apply)
      + cpu_core_count                       = (known after apply)
      + cpu_threads_per_core                 = (known after apply)
      + disable_api_stop                     = (known after apply)
      + disable_api_termination              = (known after apply)
      + ebs_optimized                        = (known after apply)
      + get_password_data                    = false
      + host_id                              = (known after apply)
      + host_resource_group_arn              = (known after apply)
      + iam_instance_profile                 = (known after apply)
      + id                                   = (known after apply)
      + instance_initiated_shutdown_behavior = (known after apply)
      + instance_lifecycle                   = (known after apply)
      + instance_state                       = (known after apply)
      + instance_type                        = "t3.micro"
      + ipv6_address_count                   = (known after apply)
      + ipv6_addresses                       = (known after apply)
      + key_name                             = "udacity_west"
      + monitoring                           = (known after apply)
      + outpost_arn                          = (known after apply)
      + password_data                        = (known after apply)
      + placement_group                      = (known after apply)
      + placement_partition_number           = (known after apply)
      + primary_network_interface_id         = (known after apply)
      + private_dns                          = (known after apply)
      + private_ip                           = (known after apply)
      + public_dns                           = (known after apply)
      + public_ip                            = (known after apply)
      + secondary_private_ips                = (known after apply)
      + security_groups                      = (known after apply)
      + source_dest_check                    = true
```

```
    + spot_instance_request_id             = (known after apply)
    + subnet_id                            = "subnet-0c3f2d1dcd47bae6d"
    + tags                                 = {
        + "Name" = "Ubuntu-Web"
      }
    + tags_all                             = {
        + "Name"      = "Ubuntu-Web"
        + "Terraform" = "true"
      }
    + tenancy                              = (known after apply)
    + user_data                            = (known after apply)
    + user_data_base64                     = (known after apply)
    + user_data_replace_on_change          = false
    + vpc_security_group_ids               = (known after apply)
  }
# module.project_ec2.aws_instance.ubuntu[1] will be created
+ resource "aws_instance" "ubuntu" {
    + ami                                  = "ami-063d2f012ccad1ebd"
    + arn                                  = (known after apply)
    + associate_public_ip_address          = (known after apply)
    + availability_zone                    = (known after apply)
    + cpu_core_count                       = (known after apply)
    + cpu_threads_per_core                 = (known after apply)
    + disable_api_stop                     = (known after apply)
    + disable_api_termination              = (known after apply)
    + ebs_optimized                        = (known after apply)
    + get_password_data                    = false
    + host_id                              = (known after apply)
    + host_resource_group_arn              = (known after apply)
    + iam_instance_profile                 = (known after apply)
    + id                                   = (known after apply)
    + instance_initiated_shutdown_behavior = (known after apply)
```

```
      + instance_lifecycle                   = (known after apply)
      + instance_state                       = (known after apply)
      + instance_type                        = "t3.micro"
      + ipv6_address_count                   = (known after apply)
      + ipv6_addresses                       = (known after apply)
      + key_name                             = "udacity_west"
      + monitoring                           = (known after apply)
      + outpost_arn                          = (known after apply)
      + password_data                        = (known after apply)
      + placement_group                      = (known after apply)
      + placement_partition_number           = (known after apply)
      + primary_network_interface_id         = (known after apply)
      + private_dns                          = (known after apply)
      + private_ip                           = (known after apply)
      + public_dns                           = (known after apply)
      + public_ip                            = (known after apply)
      + secondary_private_ips                = (known after apply)
      + security_groups                      = (known after apply)
      + source_dest_check                    = true
      + spot_instance_request_id             = (known after apply)
      + subnet_id                            = "subnet-0c3f2d1dcd47bae6d"
      + tags                                 = {
          + "Name" = "Ubuntu-Web"
        }
      + tags_all                             = {
          + "Name"      = "Ubuntu-Web"
          + "Terraform" = "true"
        }
      + tenancy                              = (known after apply)
      + user_data                            = (known after apply)
      + user_data_base64                     = (known after apply)
      + user_data_replace_on_change          = false
      + vpc_security_group_ids               = (known after apply)
    }
  # module.project_ec2.aws_instance.ubuntu[2] will be created
  + resource "aws_instance" "ubuntu" {
      + ami                                  = "ami-063d2f012ccad1ebd"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = (known after apply)
      + availability_zone                    = (known after apply)
      + cpu_core_count                       = (known after apply)
      + cpu_threads_per_core                 = (known after apply)
      + disable_api_stop                     = (known after apply)
      + disable_api_termination              = (known after apply)
      + ebs_optimized                        = (known after apply)
      + get_password_data                    = false
      + host_id                              = (known after apply)
      + host_resource_group_arn              = (known after apply)
      + iam_instance_profile                 = (known after apply)
      + id                                   = (known after apply)
      + instance_initiated_shutdown_behavior = (known after apply)
      + instance_lifecycle                   = (known after apply)
      + instance_state                       = (known after apply)
      + instance_type                        = "t3.micro"
      + ipv6_address_count                   = (known after apply)
      + ipv6_addresses                       = (known after apply)
      + key_name                             = "udacity_west"
      + monitoring                           = (known after apply)
      + outpost_arn                          = (known after apply)
      + password_data                        = (known after apply)
      + placement_group                      = (known after apply)
      + placement_partition_number           = (known after apply)
      + primary_network_interface_id         = (known after apply)
      + private_dns                          = (known after apply)
      + private_ip                           = (known after apply)
      + public_dns                           = (known after apply)
      + public_ip                            = (known after apply)
      + secondary_private_ips                = (known after apply)
      + security_groups                      = (known after apply)
      + source_dest_check                    = true
      + spot_instance_request_id             = (known after apply)
      + subnet_id                            = "subnet-0c3f2d1dcd47bae6d"
      + tags                                 = {
          + "Name" = "Ubuntu-Web"
        }
      + tags_all                             = {
          + "Name"      = "Ubuntu-Web"
          + "Terraform" = "true"
        }
      + tenancy                              = (known after apply)
      + user_data                            = (known after apply)
      + user_data_base64                     = (known after apply)
      + user_data_replace_on_change          = false
      + vpc_security_group_ids               = (known after apply)
    }
  # module.project_ec2.aws_security_group.ec2_sg will be created
  + resource "aws_security_group" "ec2_sg" {
      + arn                    = (known after apply)
      + description            = "Managed by Terraform"
      + egress                 = [
          + {
              + cidr_blocks      = [
                  + "0.0.0.0/0",
                ]
              + description      = ""
              + from_port        = 0
              + ipv6_cidr_blocks = []
              + prefix_list_ids  = []
              + protocol         = "-1"
              + security_groups  = []
              + self             = false
              + to_port          = 0
            },
        ]
      + id                     = (known after apply)
      + ingress                = [
          + {
              + cidr_blocks      = [
                  + "0.0.0.0/0",
                ]
              + description      = "monitoring"
              + from_port        = 9100
              + ipv6_cidr_blocks = []
              + prefix_list_ids  = []
              + protocol         = "tcp"
              + security_groups  = []
              + self             = false
              + to_port          = 9100
            },
          + {
              + cidr_blocks      = [
                  + "0.0.0.0/0",
                ]
              + description      = "ssh port"
              + from_port        = 22
              + ipv6_cidr_blocks = []
              + prefix_list_ids  = []
          + protocol         = "tcp"
          + security_groups  = []
          + self             = false
          + to_port          = 22
        },
      + {
          + cidr_blocks      = [
              + "0.0.0.0/0",
            ]
          + description      = "web port"
          + from_port        = 80
          + ipv6_cidr_blocks = []
          + prefix_list_ids  = []
          + protocol         = "tcp"
          + security_groups  = []
          + self             = false
          + to_port          = 80
        },
    ]
  + name                   = "ec2_sg"
  + name_prefix            = (known after apply)
  + owner_id               = (known after apply)
  + revoke_rules_on_delete = false
  + tags                   = {
      + "Name" = "ec2_sg"
    }
  + tags_all               = {
      + "Name"      = "ec2_sg"
      + "Terraform" = "true"
    }
  + vpc_id                 = "vpc-0152fd50983383e5a"
}
```

```
# module.project_eks.aws_eks_cluster.cluster will be created
+ resource "aws_eks_cluster" "cluster" {
    + arn                   = (known after apply)
    + certificate_authority = (known after apply)
    + cluster_id            = (known after apply)
    + created_at            = (known after apply)
    + endpoint              = (known after apply)
    + id                    = (known after apply)
    + identity              = (known after apply)
    + name                  = "udacity-cluster"
    + platform_version      = (known after apply)
    + role_arn              =
"arn:aws:iam::810716384126:role/app-udacity-eks-cluster-role"
    + status                = (known after apply)
    + tags_all              = {
        + "Name"      = "udacity"
        + "Terraform" = "true"
      }
    + version               = "1.28"
    + vpc_config {
        + cluster_security_group_id = (known after apply)
        + endpoint_private_access   = false
        + endpoint_public_access    = true
        + public_access_cidrs       = (known after apply)
        + security_group_ids        = (known after apply)
        + subnet_ids                = [
            + "subnet-0843176d9edd531d1",
            + "subnet-0e53b166f0536c356",
          ]
        + vpc_id                    = (known after apply)
      }
  }
```

```
 # module.project_eks.aws_eks_node_group.node will be
created
 + resource "aws_eks_node_group" "node" {
    + ami_type               = (known after apply)
    + arn                    = (known after apply)
    + capacity_type          = (known after apply)
    + cluster_name           = "udacity-cluster"
    + disk_size              = (known after apply)
    + id                     = (known after apply)
    + instance_types         = [
       + "t3.medium",
      ]
    + node_group_name        = "app-udacity-node-group"
    + node_group_name_prefix = (known after apply)
    + node_role_arn          =
"arn:aws:iam::810716384126:role/app-udacity-eks-node-role"
    + release_version        = (known after apply)
    + resources              = (known after apply)
    + status                 = (known after apply)
    + subnet_ids             = [
       + "subnet-0843176d9edd531d1",
       + "subnet-0e53b166f0536c356",
      ]
    + tags                   = {
       + "Name" = "eks-udacity-nodes"
      }
    + tags_all               = {
       + "Name"      = "eks-udacity-nodes"
       + "Terraform" = "true"
      }
    + version                = (known after apply)
    + scaling_config {
          + desired_size = 2
          + max_size     = 2
          + min_size     = 1
        }
    }
  #
module.project_eks.aws_iam_role_policy_attachment.cluster_A
mazonEKSCloudwatchPolicy will be created
  + resource "aws_iam_role_policy_attachment"
"cluster_AmazonEKSCloudwatchPolicy" {
      + id         = (known after apply)
      + policy_arn =
"arn:aws:iam::810716384126:policy/app-udacity-eks-cluster-role-cl
oudwatch-policy"
      + role       = "app-udacity-eks-node-role"
    }
  #
module.project_eks.aws_iam_role_policy_attachment.cluster_A
mazonEKSClusterPolicy will be created
  + resource "aws_iam_role_policy_attachment"
"cluster_AmazonEKSClusterPolicy" {
      + id         = (known after apply)
      + policy_arn =
"arn:aws:iam::aws:policy/AmazonEKSClusterPolicy"
      + role       = "app-udacity-eks-cluster-role"
    }
  #
module.project_eks.aws_iam_role_policy_attachment.cluster_A
mazonEKSServicePolicy will be created
  + resource "aws_iam_role_policy_attachment"
"cluster_AmazonEKSServicePolicy" {
      + id         = (known after apply)
      + policy_arn =
"arn:aws:iam::aws:policy/AmazonEKSServicePolicy"
      + role       = "app-udacity-eks-cluster-role"
    }
  #
module.project_eks.aws_iam_role_policy_attachment.node_Am
azonEC2ContainerRegistryReadOnly will be created
  + resource "aws_iam_role_policy_attachment"
"node_AmazonEC2ContainerRegistryReadOnly" {
      + id         = (known after apply)
      + policy_arn =
"arn:aws:iam::aws:policy/AmazonEC2ContainerRegistryReadOnly
"
      + role       = "app-udacity-eks-node-role"
    }
  #
module.project_eks.aws_iam_role_policy_attachment.node_Am
azonEKSWorkerNodePolicy will be created
  + resource "aws_iam_role_policy_attachment"
"node_AmazonEKSWorkerNodePolicy" {
      + id         = (known after apply)
      + policy_arn =
"arn:aws:iam::aws:policy/AmazonEKSWorkerNodePolicy"
      + role       = "app-udacity-eks-node-role"
    }
  #
module.project_eks.aws_iam_role_policy_attachment.node_Am
azonEKS_CNI_Policy will be created
  + resource "aws_iam_role_policy_attachment"
"node_AmazonEKS_CNI_Policy" {
      + id         = (known after apply)
      + policy_arn =
"arn:aws:iam::aws:policy/AmazonEKS_CNI_Policy"
      + role       = "app-udacity-eks-node-role"
    }
  #
module.project_eks.aws_iam_role_policy_attachment.node_Clo
udWatchAgentServerPolicy will be created
  + resource "aws_iam_role_policy_attachment"
"node_CloudWatchAgentServerPolicy" {
      + id         = (known after apply)
      + policy_arn =
"arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy"
      + role       = "app-udacity-eks-node-role"
    }
  #
module.project_eks.aws_iam_role_policy_attachment.node_har
mony_policy_attachment will be created
  + resource "aws_iam_role_policy_attachment"
"node_harmony_policy_attachment" {
      + id         = (known after apply)
      + policy_arn =
"arn:aws:iam::810716384126:policy/app-udacity-instance-policy"
      + role       = "app-udacity-eks-node-role"
    }
  # module.project_eks.aws_security_group.eks-cluster will be
created
  + resource "aws_security_group" "eks-cluster" {
      + arn                    = (known after apply)
      + description            = "Managed by Terraform"
      + egress                 = [
          + {
              + cidr_blocks      = [
                  + "0.0.0.0/0",
              ]
            + description      = ""
            + from_port        = 0
            + ipv6_cidr_blocks = []
            + prefix_list_ids  = []
            + protocol         = "-1"
            + security_groups  = []
            + self             = false
            + to_port          = 0
          },
      ]
    + id                     = (known after apply)
    + ingress                = [
        + {
            + cidr_blocks      = [
                + "0.0.0.0/0",
              ]
            + description      = ""
            + from_port        = 0
            + ipv6_cidr_blocks = []
            + prefix_list_ids  = []
            + protocol         = "-1"
            + security_groups  = []
            + self             = false
            + to_port          = 0
          },
      ]
    + name                   = "SG-eks-cluster"
    + name_prefix            = (known after apply)
    + owner_id               = (known after apply)
    + revoke_rules_on_delete = false
    + tags_all               = {
          + "Name"      = "udacity"
          + "Terraform" = "true"
        }
      + vpc_id                 = "vpc-0152fd50983383e5a"
    }
Plan: 17 to add, 0 to change, 0 to destroy.
Changes to Outputs:
  + account_id  = "810716384126"
  + caller_arn  = "arn:aws:iam::810716384126:user/udacity"
  + caller_user = "AIDA3ZQTWJN7BIQ2VQ5H5"
Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.
  Enter a value: yes
module.project_eks.aws_iam_role_policy_attachment.cluster_AmazonEKSServicePolicy: Creating...
module.project_eks.aws_iam_role_policy_attachment.node_AmazonEKSWorkerNodePolicy: Creating...
module.project_eks.aws_iam_role_policy_attachment.cluster_AmazonEKSCloudwatchPolicy: Creating...
module.project_eks.aws_iam_role_policy_attachment.node_CloudWatchAgentServerPolicy: Creating...
module.project_eks.aws_iam_role_policy_attachment.node_harmony_policy_attachment: Creating...
module.project_eks.aws_iam_role_policy_attachment.node_AmazonEKS_CNI_Policy: Creating...
module.project_eks.aws_iam_role_policy_attachment.cluster_AmazonEKSClusterPolicy: Creating...
module.project_eks.aws_security_group.eks-cluster: Creating...
module.project_eks.aws_iam_role_policy_attachment.node_AmazonEC2ContainerRegistryReadOnly: Creating...
module.project_ec2.aws_security_group.ec2_sg: Creating...
module.project_eks.aws_iam_role_policy_attachment.node_Am
azonEKS_CNI_Policy: Creation complete after 1s
[id=app-udacity-eks-node-role-20231218232100010000000001]
module.project_eks.aws_iam_role_policy_attachment.node_Am
azonEC2ContainerRegistryReadOnly: Creation complete after
1s
[id=app-udacity-eks-node-role-20231218232100024100000003]
module.project_eks.aws_iam_role_policy_attachment.node_Clo
udWatchAgentServerPolicy: Creation complete after 2s
[id=app-udacity-eks-node-role-20231218232100022200000002]
module.project_eks.aws_iam_role_policy_attachment.node_har
mony_policy_attachment: Creation complete after 2s
[id=app-udacity-eks-node-role-20231218232100254900000005]
module.project_eks.aws_iam_role_policy_attachment.node_Am
azonEKSWorkerNodePolicy: Creation complete after 2s
[id=app-udacity-eks-node-role-20231218232100261700000006]
module.project_eks.aws_iam_role_policy_attachment.cluster_A
mazonEKSClusterPolicy: Creation complete after 2s
[id=app-udacity-eks-cluster-role-20231218232100363000000004
]
module.project_eks.aws_iam_role_policy_attachment.cluster_A
mazonEKSServicePolicy: Creation complete after 2s
[id=app-udacity-eks-cluster-role-20231218232100278900000007
]
module.project_eks.aws_iam_role_policy_attachment.cluster_A
mazonEKSCloudwatchPolicy: Creation complete after 2s
[id=app-udacity-eks-node-role-20231218232100308200000008]
module.project_ec2.aws_security_group.ec2_sg: Creation
complete after 5s [id=sg-0e83f20d15da5deeb]
module.project_ec2.aws_instance.ubuntu[2]: Creating...
module.project_ec2.aws_instance.ubuntu[1]: Creating...
module.project_ec2.aws_instance.ubuntu[0]: Creating...
module.project_eks.aws_security_group.eks-cluster: Creation complete after 6s [id=sg-0fbd98dbd1b98d607]
module.project_eks.aws_eks_cluster.cluster: Creating...
module.project_ec2.aws_instance.ubuntu[2]: Still creating... [10s elapsed]
module.project_ec2.aws_instance.ubuntu[0]: Still creating... [10s elapsed]
module.project_ec2.aws_instance.ubuntu[1]: Still creating... [10s elapsed]
module.project_eks.aws_eks_cluster.cluster: Still creating... [10s elapsed]
module.project_ec2.aws_instance.ubuntu[2]: Creation complete after 15s [id=i-0c069009e1eb582ef]
module.project_ec2.aws_instance.ubuntu[1]: Creation complete after 15s [id=i-00d8670964b04a40d]
module.project_eks.aws_eks_cluster.cluster: Still creating... [20s elapsed]
module.project_eks.aws_eks_cluster.cluster: Still creating... [30s elapsed]
module.project_eks.aws_eks_cluster.cluster: Still creating... [40s elapsed]
module.project_eks.aws_eks_cluster.cluster: Still creating... [50s elapsed]
module.project_eks.aws_eks_cluster.cluster: Still creating... [1m0s elapsed]
module.project_eks.aws_eks_cluster.cluster: Still creating... [1m10s elapsed]
module.project_eks.aws_eks_cluster.cluster: Still creating... [1m20s elapsed]
module.project_eks.aws_eks_cluster.cluster: Still creating... [1m30s elapsed]
module.project_eks.aws_eks_cluster.cluster: Still creating...
```

```
[1m40s elapsed]
module.project_eks.aws_eks_cluster.cluster: Still creating...
[1m50s elapsed]
module.project_eks.aws_eks_cluster.cluster: Still creating...
[2m0s elapsed]
module.project_eks.aws_eks_cluster.cluster: Still creating...
[2m10s elapsed]
module.project_eks.aws_eks_cluster.cluster: Still creating...
[2m20s elapsed]
module.project_eks.aws_eks_cluster.cluster: Still creating...
[2m30s elapsed]
module.project_eks.aws_eks_cluster.cluster: Still creating...
[2m40s elapsed]
module.project_eks.aws_eks_cluster.cluster: Still creating...
[2m50s elapsed]
module.project_eks.aws_eks_cluster.cluster: Still creating...
[3m0s elapsed]
module.project_eks.aws_eks_cluster.cluster: Still creating...
[3m10s elapsed]
module.project_eks.aws_eks_cluster.cluster: Still creating...
[3m20s elapsed]
module.project_eks.aws_eks_cluster.cluster: Still creating...
[3m30s elapsed]
module.project_eks.aws_eks_cluster.cluster: Still creating...
[3m40s elapsed]
module.project_eks.aws_eks_cluster.cluster: Still creating...
[3m50s elapsed]
module.project_eks.aws_eks_cluster.cluster: Still creating...
[4m0s elapsed]
module.project_eks.aws_eks_cluster.cluster: Still creating...
[4m10s elapsed]
module.project_eks.aws_eks_cluster.cluster: Still creating...
```

```
[4m20s elapsed]
module.project_eks.aws_eks_cluster.cluster: Still creating...
[4m30s elapsed]
module.project_eks.aws_eks_cluster.cluster: Still creating...
[4m40s elapsed]
module.project_eks.aws_eks_cluster.cluster: Still creating...
[4m50s elapsed]
module.project_eks.aws_eks_cluster.cluster: Still creating...
[5m0s elapsed]
module.project_eks.aws_eks_cluster.cluster: Still creating...
[5m10s elapsed]
module.project_eks.aws_eks_cluster.cluster: Still creating...
[5m20s elapsed]
module.project_eks.aws_eks_cluster.cluster: Still creating...
[5m30s elapsed]
module.project_eks.aws_eks_cluster.cluster: Still creating...
[5m40s elapsed]
module.project_eks.aws_eks_cluster.cluster: Still creating...
[5m50s elapsed]
module.project_eks.aws_eks_cluster.cluster: Still creating...
[6m0s elapsed]
module.project_eks.aws_eks_cluster.cluster: Creation complete
after 6m0s [id=udacity-cluster]
data.aws_eks_cluster_auth.cluster: Reading...
data.aws_eks_cluster.cluster: Reading...
data.aws_eks_cluster_auth.cluster: Read complete after 0s
[id=udacity-cluster]
module.project_eks.aws_eks_node_group.node: Creating...
data.aws_eks_cluster.cluster: Read complete after 1s
[id=udacity-cluster]
module.project_eks.aws_eks_node_group.node: Still creating...
[10s elapsed]
```

```
module.project_eks.aws_eks_node_group.node: Still creating...
[20s elapsed]
module.project_eks.aws_eks_node_group.node: Still creating...
[30s elapsed]
module.project_eks.aws_eks_node_group.node: Still creating...
[40s elapsed]
module.project_eks.aws_eks_node_group.node: Still creating...
[50s elapsed]
module.project_eks.aws_eks_node_group.node: Still creating...
[1m0s elapsed]
module.project_eks.aws_eks_node_group.node: Still creating...
[1m10s elapsed]
module.project_eks.aws_eks_node_group.node: Still creating...
[1m20s elapsed]
module.project_eks.aws_eks_node_group.node: Still creating...
[1m30s elapsed]
module.project_eks.aws_eks_node_group.node: Still creating...
[1m40s elapsed]
module.project_eks.aws_eks_node_group.node: Still creating...
[1m50s elapsed]
module.project_eks.aws_eks_node_group.node: Still creating...
[2m0s elapsed]
module.project_eks.aws_eks_node_group.node: Still creating...
[2m10s elapsed]
module.project_eks.aws_eks_node_group.node: Creation
complete after 2m14s
[id=udacity-cluster:app-udacity-node-group]
kubernetes_namespace.udacity: Creating...
kubernetes_service.grafana-external: Creating...
kubernetes_namespace.udacity: Creation complete after 3s
[id=udacity]
```

```
│ Error: namespaces "monitoring" not found
│
│   with kubernetes_service.grafana-external,
│   on project.tf line 10, in resource "kubernetes_service"
"grafana-external":
│   10:   resource "kubernetes_service" "grafana-external" {
│
╵

╷
│ Error: waiting for EC2 Instance (i-056945cbc45004bf8)
create: unexpected state 'shutting-down', wanted target
'running'. last error: Client.UserInitiatedShutdown: User
initiated shutdown
│
│   with module.project_ec2.aws_instance.ubuntu[0],
│   on modules/ec2/ec2.tf line 10, in resource "aws_instance"
"ubuntu":
│   10: resource "aws_instance" "ubuntu" {
│
╵

jees-macbook:zone2 jeesantony$ aws eks --region us-west-1
update-kubeconfig --name udacity-cluster
Added new context
arn:aws:eks:us-west-1:810716384126:cluster/udacity-cluster to
/Users/jeesantony/.kube/config
jees-macbook:zone2 jeesantony$ kubectl config use-context
arn:aws:eks:us-west-1:810716384126:cluster/udacity-cluster
Switched to context
"arn:aws:eks:us-west-1:810716384126:cluster/udacity-cluster".
jees-macbook:zone2 jeesantony$ kubectl create namespace
monitoring
namespace/monitoring created
```

```
jees-macbook:zone2 jeesantony$ terraform init
Initializing the backend...
Initializing modules...
Initializing provider plugins...
- terraform.io/builtin/terraform is built in to Terraform
- Reusing previous version of hashicorp/aws from the dependency
lock file
- Reusing previous version of hashicorp/kubernetes from the
dependency lock file
- Using previously-installed hashicorp/aws v5.31.0
- Using previously-installed hashicorp/kubernetes v2.24.0
Terraform has been successfully initialized!
You may now begin working with Terraform. Try running
"terraform plan" to see
any changes that are required for your infrastructure. All Terraform
commands
should now work.
If you ever set or change modules or backend configuration for
Terraform,
rerun this command to reinitialize your working directory. If you
forget, other
commands will detect it and remind you to do so if necessary.
jees-macbook:zone2 jeesantony$ terraform apply
data.terraform_remote_state.vpc: Reading...
data.aws_iam_policy.cloudwatch-policy: Reading...
data.aws_ami.amazon_linux_2: Reading...
module.project_eks.data.aws_iam_policy_document.eks_node_
assume_role_policy: Reading...
data.aws_iam_role.eks_node_role: Reading...
module.project_eks.aws_iam_role_policy_attachment.node_Am
azonEKSWorkerNodePolicy: Refreshing state...
[id=app-udacity-eks-node-role-20231218232100261700000006]
```

```
data.aws_iam_role.eks_cluster_role: Reading...
module.project_eks.data.aws_iam_policy_document.eks_assume_role_policy: Reading...
module.project_eks.aws_iam_role_policy_attachment.node_AmazonEC2ContainerRegistryReadOnly: Refreshing state... [id=app-udacity-eks-node-role-20231218232100024100000003]
data.aws_caller_identity.current: Reading...
module.project_eks.data.aws_iam_policy_document.eks_node_assume_role_policy: Read complete after 0s [id=2851119427]
module.project_eks.data.aws_iam_policy_document.eks_assume_role_policy: Read complete after 0s [id=3552664922]
module.project_eks.aws_iam_role_policy_attachment.cluster_AmazonEKSServicePolicy: Refreshing state... [id=app-udacity-eks-cluster-role-20231218232100278900000007]
module.project_eks.aws_iam_role_policy_attachment.cluster_AmazonEKSClusterPolicy: Refreshing state... [id=app-udacity-eks-cluster-role-20231218232100036300000004]
data.aws_caller_identity.current: Read complete after 0s [id=810716384126]
module.project_eks.aws_iam_role_policy_attachment.node_CloudWatchAgentServerPolicy: Refreshing state... [id=app-udacity-eks-node-role-20231218232100022200000002]
data.aws_iam_policy.instance-policy: Reading...
module.project_eks.aws_iam_role_policy_attachment.node_AmazonEKS_CNI_Policy: Refreshing state... [id=app-udacity-eks-node-role-20231218232100010000000001]
data.aws_iam_role.eks_node_role: Read complete after 1s [id=app-udacity-eks-node-role]
data.aws_iam_role.eks_cluster_role: Read complete after 1s [id=app-udacity-eks-cluster-role]
data.aws_ami.amazon_linux_2: Read complete after 1s
[id=ami-0967795d5c824c5da]
data.terraform_remote_state.vpc: Read complete after 4s
module.project_eks.aws_security_group.eks-cluster: Refreshing
state... [id=sg-0fbd98dbd1b98d607]
module.project_ec2.aws_security_group.ec2_sg: Refreshing
state... [id=sg-0e83f20d15da5deeb]
module.project_eks.aws_eks_cluster.cluster: Refreshing state...
[id=udacity-cluster]
module.project_ec2.aws_instance.ubuntu[0]: Refreshing state...
[id=i-056945cbc45004bf8]
module.project_ec2.aws_instance.ubuntu[1]: Refreshing state...
[id=i-00d8670964b04a40d]
module.project_ec2.aws_instance.ubuntu[2]: Refreshing state...
[id=i-0c069009e1eb582ef]
data.aws_eks_cluster.cluster: Reading...
data.aws_eks_cluster_auth.cluster: Reading...
data.aws_eks_cluster_auth.cluster: Read complete after 0s
[id=udacity-cluster]
data.aws_eks_cluster.cluster: Read complete after 1s
[id=udacity-cluster]
data.aws_iam_policy.cloudwatch-policy: Read complete after 8s
[id=arn:aws:iam::810716384126:policy/app-udacity-eks-cluster-
role-cloudwatch-policy]
module.project_eks.aws_iam_role_policy_attachment.cluster_A
mazonEKSCloudwatchPolicy: Refreshing state...
[id=app-udacity-eks-node-role-20231218232100308200000008]
data.aws_iam_policy.instance-policy: Read complete after 10s
[id=arn:aws:iam::810716384126:policy/app-udacity-instance-po
licy]
module.project_eks.aws_iam_role_policy_attachment.node_har
mony_policy_attachment: Refreshing state...
```

```
[id=app-udacity-eks-node-role-20231218232100254900000005]
module.project_eks.aws_eks_node_group.node: Refreshing
state... [id=udacity-cluster:app-udacity-node-group]
kubernetes_namespace.udacity: Refreshing state... [id=udacity]
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:
  # kubernetes_service.grafana-external will be created
  + resource "kubernetes_service" "grafana-external" {
      + id                     = (known after apply)
      + status                 = (known after apply)
      + wait_for_load_balancer = true
      + metadata {
          + annotations      = {
              +
"service.beta.kubernetes.io/aws-load-balancer-nlb-target-type" =
"ip"
              + "service.beta.kubernetes.io/aws-load-balancer-type"
= "nlb"
            }
          + generation       = (known after apply)
          + name             = "grafana-external"
          + namespace        = "monitoring"
          + resource_version = (known after apply)
          + uid              = (known after apply)
        }
      + spec {
          + allocate_load_balancer_node_ports = true
          + cluster_ip                        = (known after apply)
          + cluster_ips                       = (known after apply)
          + external_traffic_policy           = (known after apply)
          + health_check_node_port            = (known after apply)
          + internal_traffic_policy           = (known after apply)
          + ip_families                       = (known after apply)
          + ip_family_policy                  = (known after apply)
          + publish_not_ready_addresses       = false
          + selector                          = {
              + "app.kubernetes.io/name" = "grafana"
            }
          + session_affinity                  = "None"
          + type                              = "LoadBalancer"
          + port {
              + node_port   = (known after apply)
              + port        = 80
              + protocol    = "TCP"
              + target_port = "3000"
            }
        }
    }
  # module.project_ec2.aws_instance.ubuntu[0] will be created
  + resource "aws_instance" "ubuntu" {
      + ami                                  = "ami-063d2f012ccad1ebd"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = (known after apply)
      + availability_zone                    = (known after apply)
      + cpu_core_count                       = (known after apply)
      + cpu_threads_per_core                 = (known after apply)
      + disable_api_stop                     = (known after apply)
      + disable_api_termination              = (known after apply)
      + ebs_optimized                        = (known after apply)
      + get_password_data                    = false
      + host_id                              = (known after apply)
      + host_resource_group_arn              = (known after apply)
      + iam_instance_profile                 = (known after apply)
      + id                                   = (known after apply)
      + instance_initiated_shutdown_behavior = (known after apply)
      + instance_lifecycle                   = (known after apply)
      + instance_state                       = (known after apply)
      + instance_type                        = "t3.micro"
      + ipv6_address_count                   = (known after apply)
      + ipv6_addresses                       = (known after apply)
      + key_name                             = "udacity_west"
      + monitoring                           = (known after apply)
      + outpost_arn                          = (known after apply)
      + password_data                        = (known after apply)
      + placement_group                      = (known after apply)
      + placement_partition_number           = (known after apply)
      + primary_network_interface_id         = (known after apply)
      + private_dns                          = (known after apply)
      + private_ip                           = (known after apply)
      + public_dns                           = (known after apply)
      + public_ip                            = (known after apply)
      + secondary_private_ips                = (known after apply)
      + security_groups                      = (known after apply)
      + source_dest_check                    = true
      + spot_instance_request_id             = (known after apply)
      + subnet_id                            = "subnet-0c3f2d1dcd47bae6d"
      + tags                                 = {
          + "Name" = "Ubuntu-Web"
        }
      + tags_all                             = {
          + "Name"      = "Ubuntu-Web"
          + "Terraform" = "true"
        }
      + tenancy                              = (known after apply)
      + user_data                            = (known after apply)
      + user_data_base64                     = (known after apply)
      + user_data_replace_on_change          = false
      + vpc_security_group_ids               = [
          + "sg-0e83f20d15da5deeb",
        ]
    }
Plan: 2 to add, 0 to change, 0 to destroy.
Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.
  Enter a value: yes
kubernetes_service.grafana-external: Creating...
module.project_ec2.aws_instance.ubuntu[0]: Creating...
kubernetes_service.grafana-external: Still creating... [10s
elapsed]
module.project_ec2.aws_instance.ubuntu[0]: Still creating...
[10s elapsed]
kubernetes_service.grafana-external: Still creating... [20s
elapsed]
kubernetes_service.grafana-external: Still creating... [30s
elapsed]
kubernetes_service.grafana-external: Still creating... [40s
elapsed]
kubernetes_service.grafana-external: Still creating... [50s
elapsed]
kubernetes_service.grafana-external: Still creating... [1m0s
elapsed]
kubernetes_service.grafana-external: Still creating... [1m10s
elapsed]
kubernetes_service.grafana-external: Still creating... [1m20s
```

```
elapsed]
kubernetes_service.grafana-external: Still creating... [1m30s
elapsed]
kubernetes_service.grafana-external: Still creating... [1m40s
elapsed]
kubernetes_service.grafana-external: Still creating... [1m50s
elapsed]
kubernetes_service.grafana-external: Still creating... [2m0s
elapsed]
kubernetes_service.grafana-external: Still creating... [2m10s
elapsed]
kubernetes_service.grafana-external: Still creating... [2m20s
elapsed]
kubernetes_service.grafana-external: Still creating... [2m30s
elapsed]
kubernetes_service.grafana-external: Still creating... [2m40s
elapsed]
kubernetes_service.grafana-external: Still creating... [2m50s
elapsed]
kubernetes_service.grafana-external: Still creating... [3m0s
elapsed]
kubernetes_service.grafana-external: Still creating... [3m10s
elapsed]
kubernetes_service.grafana-external: Still creating... [3m20s
elapsed]
kubernetes_service.grafana-external: Still creating... [3m30s
elapsed]
kubernetes_service.grafana-external: Still creating... [3m40s
elapsed]
kubernetes_service.grafana-external: Still creating... [3m50s
elapsed]
kubernetes_service.grafana-external: Still creating... [4m0s
```

```
elapsed]
kubernetes_service.grafana-external: Still creating... [4m10s
elapsed]
kubernetes_service.grafana-external: Still creating... [4m20s
elapsed]
kubernetes_service.grafana-external: Still creating... [4m30s
elapsed]
kubernetes_service.grafana-external: Still creating... [4m40s
elapsed]
kubernetes_service.grafana-external: Still creating... [4m50s
elapsed]
kubernetes_service.grafana-external: Still creating... [5m0s
elapsed]
kubernetes_service.grafana-external: Still creating... [5m10s
elapsed]
kubernetes_service.grafana-external: Still creating... [5m20s
elapsed]
kubernetes_service.grafana-external: Still creating... [5m30s
elapsed]
kubernetes_service.grafana-external: Still creating... [5m40s
elapsed]
kubernetes_service.grafana-external: Still creating... [5m50s
elapsed]
kubernetes_service.grafana-external: Still creating... [6m0s
elapsed]
kubernetes_service.grafana-external: Still creating... [6m10s
elapsed]
kubernetes_service.grafana-external: Still creating... [6m20s
elapsed]
kubernetes_service.grafana-external: Still creating... [6m30s
elapsed]
kubernetes_service.grafana-external: Still creating... [6m40s
```

```
elapsed]
kubernetes_service.grafana-external: Still creating... [6m50s
elapsed]
kubernetes_service.grafana-external: Still creating... [7m0s
elapsed]
kubernetes_service.grafana-external: Still creating... [7m10s
elapsed]
kubernetes_service.grafana-external: Still creating... [7m20s
elapsed]
kubernetes_service.grafana-external: Still creating... [7m30s
elapsed]
kubernetes_service.grafana-external: Still creating... [7m40s
elapsed]
kubernetes_service.grafana-external: Still creating... [7m50s
elapsed]
kubernetes_service.grafana-external: Still creating... [8m0s
elapsed]
kubernetes_service.grafana-external: Still creating... [8m10s
elapsed]
kubernetes_service.grafana-external: Still creating... [8m20s
elapsed]
kubernetes_service.grafana-external: Still creating... [8m30s
elapsed]
kubernetes_service.grafana-external: Still creating... [8m40s
elapsed]
kubernetes_service.grafana-external: Still creating... [8m50s
elapsed]
kubernetes_service.grafana-external: Still creating... [9m0s
elapsed]
kubernetes_service.grafana-external: Still creating... [9m10s
elapsed]
kubernetes_service.grafana-external: Still creating... [9m20s
```

```
elapsed]
kubernetes_service.grafana-external: Still creating... [9m30s
elapsed]
kubernetes_service.grafana-external: Still creating... [9m40s
elapsed]
kubernetes_service.grafana-external: Still creating... [9m50s
elapsed]
╷
│ Error: client rate limiter Wait returned an error: context
deadline exceeded
│
│   with kubernetes_service.grafana-external,
│   on project.tf line 10, in resource "kubernetes_service"
"grafana-external":
│   10:   resource "kubernetes_service" "grafana-external" {
│
╵

╷
│ Error: waiting for EC2 Instance (i-015e233ace87f88cb)
create: unexpected state 'shutting-down', wanted target
'running'. last error: Client.UserInitiatedShutdown: User
initiated shutdown
│
│   with module.project_ec2.aws_instance.ubuntu[0],
│   on modules/ec2/ec2.tf line 10, in resource "aws_instance"
"ubuntu":
│   10: resource "aws_instance" "ubuntu" {
│
╵

jees-macbook:zone2 jeesantony$ terraform init
Initializing the backend...
Initializing modules...
Initializing provider plugins...
- terraform.io/builtin/terraform is built in to Terraform
- Reusing previous version of hashicorp/kubernetes from the dependency lock file
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/kubernetes v2.24.0
- Using previously-installed hashicorp/aws v5.31.0
Terraform has been successfully initialized!
You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.
If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
jees-macbook:zone2 jeesantony$ terraform apply
data.terraform_remote_state.vpc: Reading...
module.project_eks.aws_iam_role_policy_attachment.node_AmazonEC2ContainerRegistryReadOnly: Refreshing state... [id=app-udacity-eks-node-role-20231218232100024100000003]
data.aws_iam_role.eks_cluster_role: Reading...
module.project_eks.data.aws_iam_policy_document.eks_node_assume_role_policy: Reading...
data.aws_iam_role.eks_node_role: Reading...
module.project_eks.data.aws_iam_policy_document.eks_assume_role_policy: Reading...
data.aws_iam_policy.instance-policy: Reading...
data.aws_iam_policy.cloudwatch-policy: Reading...
data.aws_caller_identity.current: Reading...
data.aws_ami.amazon_linux_2: Reading...
module.project_eks.data.aws_iam_policy_document.eks_node_
assume_role_policy: Read complete after 0s [id=2851119427]
module.project_eks.data.aws_iam_policy_document.eks_assum
e_role_policy: Read complete after 0s [id=3552664922]
module.project_eks.aws_iam_role_policy_attachment.cluster_A
mazonEKSClusterPolicy: Refreshing state...
[id=app-udacity-eks-cluster-role-20231218232100036300000004
]
module.project_eks.aws_iam_role_policy_attachment.node_Clo
udWatchAgentServerPolicy: Refreshing state...
[id=app-udacity-eks-node-role-20231218232100022200000002]
data.aws_caller_identity.current: Read complete after 0s
[id=810716384126]
module.project_eks.aws_iam_role_policy_attachment.cluster_A
mazonEKSServicePolicy: Refreshing state...
[id=app-udacity-eks-cluster-role-20231218232100278900000007
]
data.aws_ami.amazon_linux_2: Read complete after 1s
[id=ami-0967795d5c824c5da]
module.project_eks.aws_iam_role_policy_attachment.node_Am
azonEKSWorkerNodePolicy: Refreshing state...
[id=app-udacity-eks-node-role-20231218232100261700000006]
data.aws_iam_role.eks_node_role: Read complete after 2s
[id=app-udacity-eks-node-role]
module.project_eks.aws_iam_role_policy_attachment.node_Am
azonEKS_CNI_Policy: Refreshing state...
[id=app-udacity-eks-node-role-20231218232100010000000001]
data.aws_iam_role.eks_cluster_role: Read complete after 2s
[id=app-udacity-eks-cluster-role]
data.terraform_remote_state.vpc: Read complete after 4s
```

```
module.project_eks.aws_security_group.eks-cluster: Refreshing
state... [id=sg-0fbd98dbd1b98d607]
module.project_ec2.aws_security_group.ec2_sg: Refreshing
state... [id=sg-0e83f20d15da5deeb]
module.project_eks.aws_eks_cluster.cluster: Refreshing state...
[id=udacity-cluster]
module.project_ec2.aws_instance.ubuntu[0]: Refreshing state...
[id=i-015e233ace87f88cb]
module.project_ec2.aws_instance.ubuntu[1]: Refreshing state...
[id=i-00d8670964b04a40d]
module.project_ec2.aws_instance.ubuntu[2]: Refreshing state...
[id=i-0c069009e1eb582ef]
data.aws_eks_cluster_auth.cluster: Reading...
data.aws_eks_cluster.cluster: Reading...
data.aws_eks_cluster_auth.cluster: Read complete after 0s
[id=udacity-cluster]
data.aws_eks_cluster.cluster: Read complete after 1s
[id=udacity-cluster]
data.aws_iam_policy.cloudwatch-policy: Still reading... [10s
elapsed]
data.aws_iam_policy.instance-policy: Still reading... [10s
elapsed]
data.aws_iam_policy.cloudwatch-policy: Read complete after
11s
[id=arn:aws:iam::810716384126:policy/app-udacity-eks-cluster-
role-cloudwatch-policy]
module.project_eks.aws_iam_role_policy_attachment.cluster_A
mazonEKSCloudwatchPolicy: Refreshing state...
[id=app-udacity-eks-node-role-20231218232100308200000008]
data.aws_iam_policy.instance-policy: Read complete after 13s
[id=arn:aws:iam::810716384126:policy/app-udacity-instance-po
licy]
```

```
module.project_eks.aws_iam_role_policy_attachment.node_harmony_policy_attachment: Refreshing state...
[id=app-udacity-eks-node-role-20231218232100254900000005]
module.project_eks.aws_eks_node_group.node: Refreshing
state... [id=udacity-cluster:app-udacity-node-group]
kubernetes_namespace.udacity: Refreshing state... [id=udacity]
kubernetes_service.grafana-external: Refreshing state...
[id=monitoring/grafana-external]
Terraform used the selected providers to generate the following
execution plan. Resource actions are
indicated with the following symbols:
  + create
-/+ destroy and then create replacement
Terraform will perform the following actions:
  # kubernetes_service.grafana-external is tainted, so must be
replaced
-/+ resource "kubernetes_service" "grafana-external" {
      ~ id                     = "monitoring/grafana-external" -> (known
after apply)
      ~ status                 = [
          - {
              - load_balancer = [
                  - {
                      - ingress = []
                    },
                ]
            },
        ] -> (known after apply)
        # (1 unchanged attribute hidden)
      ~ metadata {
          ~ generation       = 0 -> (known after apply)
          - labels           = {} -> null
```

```
            name             = "grafana-external"
          ~ resource_version = "1820" -> (known after apply)
          ~ uid              = "fe8b16c2-0587-458c-95bf-180eba1b49a1"
-> (known after apply)
            # (2 unchanged attributes hidden)
        }
      ~ spec {
          ~ cluster_ip                        = "172.20.70.121" -> (known after
apply)
          ~ cluster_ips                       = [
              - "172.20.70.121",
            ] -> (known after apply)
          - external_ips                      = [] -> null
          ~ external_traffic_policy           = "Cluster" -> (known after
apply)
          ~ health_check_node_port            = 0 -> (known after apply)
          ~ internal_traffic_policy           = "Cluster" -> (known after
apply)
          ~ ip_families                       = [
              - "IPv4",
            ] -> (known after apply)
          ~ ip_family_policy                  = "SingleStack" -> (known
after apply)
          - load_balancer_source_ranges       = [] -> null
            # (5 unchanged attributes hidden)
          ~ port {
              ~ node_port   = 31708 -> (known after apply)
                # (3 unchanged attributes hidden)
            }
        }
    }
  # module.project_ec2.aws_instance.ubuntu[0] will be created
```

```
+ resource "aws_instance" "ubuntu" {
    + ami                                  = "ami-063d2f012ccad1ebd"
    + arn                                  = (known after apply)
    + associate_public_ip_address          = (known after apply)
    + availability_zone                    = (known after apply)
    + cpu_core_count                       = (known after apply)
    + cpu_threads_per_core                 = (known after apply)
    + disable_api_stop                     = (known after apply)
    + disable_api_termination              = (known after apply)
    + ebs_optimized                        = (known after apply)
    + get_password_data                    = false
    + host_id                              = (known after apply)
    + host_resource_group_arn              = (known after apply)
    + iam_instance_profile                 = (known after apply)
    + id                                   = (known after apply)
    + instance_initiated_shutdown_behavior = (known after apply)
    + instance_lifecycle                   = (known after apply)
    + instance_state                       = (known after apply)
    + instance_type                        = "t3.micro"
    + ipv6_address_count                   = (known after apply)
    + ipv6_addresses                       = (known after apply)
    + key_name                             = "udacity_west"
    + monitoring                           = (known after apply)
    + outpost_arn                          = (known after apply)
    + password_data                        = (known after apply)
    + placement_group                      = (known after apply)
    + placement_partition_number           = (known after apply)
    + primary_network_interface_id         = (known after apply)
    + private_dns                          = (known after apply)
    + private_ip                           = (known after apply)
    + public_dns                           = (known after apply)
    + public_ip                            = (known after apply)
```

```
      + secondary_private_ips                = (known after apply)
      + security_groups                      = (known after apply)
      + source_dest_check                    = true
      + spot_instance_request_id             = (known after apply)
      + subnet_id                            = "subnet-0c3f2d1dcd47bae6d"
      + tags                                 = {
          + "Name" = "Ubuntu-Web"
        }
      + tags_all                             = {
          + "Name"      = "Ubuntu-Web"
          + "Terraform" = "true"
        }
      + tenancy                              = (known after apply)
      + user_data                            = (known after apply)
      + user_data_base64                     = (known after apply)
      + user_data_replace_on_change          = false
      + vpc_security_group_ids               = [
          + "sg-0e83f20d15da5deeb",
        ]
    }
Plan: 2 to add, 0 to change, 1 to destroy.
Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.
  Enter a value: yes
kubernetes_service.grafana-external: Destroying...
[id=monitoring/grafana-external]
module.project_ec2.aws_instance.ubuntu[0]: Creating...
kubernetes_service.grafana-external: Destruction complete
after 2s
kubernetes_service.grafana-external: Creating...
module.project_ec2.aws_instance.ubuntu[0]: Still creating...
```

```
[10s elapsed]
kubernetes_service.grafana-external: Still creating... [10s
elapsed]
module.project_ec2.aws_instance.ubuntu[0]: Creation complete
after 17s [id=i-07037dce3b86e1fac]
kubernetes_service.grafana-external: Still creating... [20s
elapsed]
kubernetes_service.grafana-external: Still creating... [30s
elapsed]
kubernetes_service.grafana-external: Still creating... [40s
elapsed]
kubernetes_service.grafana-external: Still creating... [50s
elapsed]
kubernetes_service.grafana-external: Still creating... [1m0s
elapsed]
kubernetes_service.grafana-external: Still creating... [1m10s
elapsed]
kubernetes_service.grafana-external: Still creating... [1m20s
elapsed]
kubernetes_service.grafana-external: Creation complete after
1m21s [id=monitoring/grafana-external]
Apply complete! Resources: 2 added, 0 changed, 1 destroyed.
Outputs:
account_id = "810716384126"
caller_arn = "arn:aws:iam::810716384126:user/udacity"
caller_user = "AIDA3ZQTWJN7BIQ2VQ5H5"
jees-macbook:zone2 jeesantony$
```