An execution plan has been generated and is shown below.

Resource actions are indicated with the following symbols:

+ create

Terraform will perform the following actions:

# aws\_eip.project-elastic-ip will be created

+ resource "aws\_eip" "project-elastic-ip" {

+ allocation\_id = (known after apply)

+ associate\_with\_private\_ip = "10.0.1.50"

+ association\_id = (known after apply)

+ carrier\_ip = (known after apply)

+ customer\_owned\_ip = (known after apply)

+ domain = (known after apply)

+ id = (known after apply)

+ instance = (known after apply)

+ network\_border\_group = (known after apply)

+ network\_interface = (known after apply)

+ private\_dns = (known after apply)

+ private\_ip = (known after apply)

+ public\_dns = (known after apply)

+ public\_ip = (known after apply)

+ public\_ipv4\_pool = (known after apply)

+ vpc = true

}

# aws\_instance.project-instance will be created

+ resource "aws\_instance" "project-instance" {

+ ami = "ami-0b84c6433cdbe5c3e"

+ arn = (known after apply)

+ associate\_public\_ip\_address = (known after apply)

+ availability\_zone = "ap-south-1a"

+ cpu\_core\_count = (known after apply)

+ cpu\_threads\_per\_core = (known after apply)

+ get\_password\_data = false

+ host\_id = (known after apply)

+ id = (known after apply)

+ instance\_state = (known after apply)

+ instance\_type = "t2.micro"

+ ipv6\_address\_count = (known after apply)

+ ipv6\_addresses = (known after apply)

+ key\_name = "devops"

+ outpost\_arn = (known after apply)

+ password\_data = (known after apply)

+ placement\_group = (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)

+ subnet\_id = (known after apply)

+ tags = {

+ "Name" = "project-instance"

}

+ tenancy = (known after apply)

+ vpc\_security\_group\_ids = (known after apply)

+ ebs\_block\_device {

+ delete\_on\_termination = (known after apply)

+ device\_name = (known after apply)

+ encrypted = (known after apply)

+ iops = (known after apply)

+ kms\_key\_id = (known after apply)

+ snapshot\_id = (known after apply)

+ tags = (known after apply)

+ throughput = (known after apply)

+ volume\_id = (known after apply)

+ volume\_size = (known after apply)

+ volume\_type = (known after apply)

}

+ enclave\_options {

+ enabled = (known after apply)

}

+ ephemeral\_block\_device {

+ device\_name = (known after apply)

+ no\_device = (known after apply)

+ virtual\_name = (known after apply)

}

+ metadata\_options {

+ http\_endpoint = (known after apply)

+ http\_put\_response\_hop\_limit = (known after apply)

+ http\_tokens = (known after apply)

}

+ network\_interface {

+ delete\_on\_termination = false

+ device\_index = 0

+ network\_interface\_id = (known after apply)

}

+ root\_block\_device {

+ delete\_on\_termination = (known after apply)

+ device\_name = (known after apply)

+ encrypted = (known after apply)

+ iops = (known after apply)

+ kms\_key\_id = (known after apply)

+ tags = (known after apply)

+ throughput = (known after apply)

+ volume\_id = (known after apply)

+ volume\_size = (known after apply)

+ volume\_type = (known after apply)

}

}

# aws\_internet\_gateway.project-gateway will be created

+ resource "aws\_internet\_gateway" "project-gateway" {

+ arn = (known after apply)

+ id = (known after apply)

+ owner\_id = (known after apply)

+ vpc\_id = (known after apply)

}

# aws\_network\_interface.project-nic will be created

+ resource "aws\_network\_interface" "project-nic" {

+ id = (known after apply)

+ ipv6\_address\_count = (known after apply)

+ ipv6\_addresses = (known after apply)

+ mac\_address = (known after apply)

+ outpost\_arn = (known after apply)

+ private\_dns\_name = (known after apply)

+ private\_ip = (known after apply)

+ private\_ips = [

+ "10.0.1.50",

]

+ private\_ips\_count = (known after apply)

+ security\_groups = (known after apply)

+ source\_dest\_check = true

+ subnet\_id = (known after apply)

+ attachment {

+ attachment\_id = (known after apply)

+ device\_index = (known after apply)

+ instance = (known after apply)

}

}

# aws\_route\_table.project-route-table will be created

+ resource "aws\_route\_table" "project-route-table" {

+ arn = (known after apply)

+ id = (known after apply)

+ owner\_id = (known after apply)

+ propagating\_vgws = (known after apply)

+ route = [

+ {

+ carrier\_gateway\_id = ""

+ cidr\_block = ""

+ destination\_prefix\_list\_id = ""

+ egress\_only\_gateway\_id = ""

+ gateway\_id = (known after apply)

+ instance\_id = ""

+ ipv6\_cidr\_block = "::/0"

+ local\_gateway\_id = ""

+ nat\_gateway\_id = ""

+ network\_interface\_id = ""

+ transit\_gateway\_id = ""

+ vpc\_endpoint\_id = ""

+ vpc\_peering\_connection\_id = ""

},

+ {

+ carrier\_gateway\_id = ""

+ cidr\_block = "0.0.0.0/0"

+ destination\_prefix\_list\_id = ""

+ egress\_only\_gateway\_id = ""

+ gateway\_id = (known after apply)

+ instance\_id = ""

+ ipv6\_cidr\_block = ""

+ local\_gateway\_id = ""

+ nat\_gateway\_id = ""

+ network\_interface\_id = ""

+ transit\_gateway\_id = ""

+ vpc\_endpoint\_id = ""

+ vpc\_peering\_connection\_id = ""

},

]

+ tags = {

+ "Name" = "project-route-table"

}

+ vpc\_id = (known after apply)

}

# aws\_route\_table\_association.project-association will be created

+ resource "aws\_route\_table\_association" "project-association" {

+ id = (known after apply)

+ route\_table\_id = (known after apply)

+ subnet\_id = (known after apply)

}

# aws\_security\_group.project-allowance will be created

+ resource "aws\_security\_group" "project-allowance" {

+ arn = (known after apply)

+ description = "Allow Web inbound traffic"

+ 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 = "HTTP"

+ from\_port = 80

+ ipv6\_cidr\_blocks = []

+ prefix\_list\_ids = []

+ protocol = "tcp"

+ security\_groups = []

+ self = false

+ to\_port = 80

},

+ {

+ cidr\_blocks = [

+ "0.0.0.0/0",

]

+ description = "SSH"

+ from\_port = 22

+ ipv6\_cidr\_blocks = []

+ prefix\_list\_ids = []

+ protocol = "tcp"

+ security\_groups = []

+ self = false

+ to\_port = 22

},

]

+ name = "project\_allowance\_traffic"

+ name\_prefix = (known after apply)

+ owner\_id = (known after apply)

+ revoke\_rules\_on\_delete = false

+ tags = {

+ "Name" = "project-allowance"

}

+ vpc\_id = (known after apply)

}

# aws\_subnet.project-subnet will be created

+ resource "aws\_subnet" "project-subnet" {

+ arn = (known after apply)

+ assign\_ipv6\_address\_on\_creation = false

+ availability\_zone = "ap-south-1a"

+ availability\_zone\_id = (known after apply)

+ cidr\_block = "10.0.1.0/24"

+ id = (known after apply)

+ ipv6\_cidr\_block\_association\_id = (known after apply)

+ map\_public\_ip\_on\_launch = false

+ owner\_id = (known after apply)

+ tags = {

+ "Name" = "project-subnet"

}

+ tags\_all = {

+ "Name" = "project-subnet"

}

+ vpc\_id = (known after apply)

}

# aws\_vpc.project-vpc will be created

+ resource "aws\_vpc" "project-vpc" {

+ arn = (known after apply)

+ assign\_generated\_ipv6\_cidr\_block = false

+ cidr\_block = "10.0.0.0/16"

+ default\_network\_acl\_id = (known after apply)

+ default\_route\_table\_id = (known after apply)

+ default\_security\_group\_id = (known after apply)

+ dhcp\_options\_id = (known after apply)

+ enable\_classiclink = (known after apply)

+ enable\_classiclink\_dns\_support = (known after apply)

+ enable\_dns\_hostnames = (known after apply)

+ enable\_dns\_support = true

+ id = (known after apply)

+ instance\_tenancy = "default"

+ ipv6\_association\_id = (known after apply)

+ ipv6\_cidr\_block = (known after apply)

+ main\_route\_table\_id = (known after apply)

+ owner\_id = (known after apply)

+ tags = {

+ "Name" = "project-vpc"

}

+ tags\_all = {

+ "Name" = "project-vpc"

}

}

Plan: 9 to add, 0 to change, 0 to destroy.

Changes to Outputs:

+ public\_dns = (known after apply)

+ server\_ip = (known after apply)

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

aws\_vpc.project-vpc: Creating...

aws\_vpc.project-vpc: Creation complete after 2s [id=vpc-06ceced028cc9d87d]

aws\_subnet.project-subnet: Creating...

aws\_internet\_gateway.project-gateway: Creating...

aws\_security\_group.project-allowance: Creating...

aws\_subnet.project-subnet: Creation complete after 2s [id=subnet-0f7dbd339d1b07efa]

aws\_internet\_gateway.project-gateway: Creation complete after 3s [id=igw-056675f2cb764dae7]

aws\_route\_table.project-route-table: Creating...

aws\_route\_table.project-route-table: Creation complete after 1s [id=rtb-0a4b556ead99deceb]

aws\_route\_table\_association.project-association: Creating...

aws\_route\_table\_association.project-association: Creation complete after 0s [id=rtbassoc-09f1116184669779c]

aws\_security\_group.project-allowance: Creation complete after 5s [id=sg-04c532d4c075d5d94]

aws\_network\_interface.project-nic: Creating...

aws\_network\_interface.project-nic: Still creating... [10s elapsed]

aws\_network\_interface.project-nic: Still creating... [20s elapsed]

aws\_network\_interface.project-nic: Still creating... [30s elapsed]

aws\_network\_interface.project-nic: Creation complete after 30s [id=eni-09cd0e1bbda57e225]

aws\_eip.project-elastic-ip: Creating...

aws\_instance.project-instance: Creating...

aws\_eip.project-elastic-ip: Creation complete after 1s [id=eipalloc-0618ee4fb343e7b5f]

aws\_instance.project-instance: Still creating... [10s elapsed]

aws\_instance.project-instance: Still creating... [20s elapsed]

aws\_instance.project-instance: Provisioning with 'remote-exec'...

aws\_instance.project-instance (remote-exec): Connecting to remote host via SSH...

aws\_instance.project-instance (remote-exec): Host: 65.2.91.114

aws\_instance.project-instance (remote-exec): User: ubuntu

aws\_instance.project-instance (remote-exec): Password: false

aws\_instance.project-instance (remote-exec): Private key: true

aws\_instance.project-instance (remote-exec): Certificate: false

aws\_instance.project-instance (remote-exec): SSH Agent: false

aws\_instance.project-instance (remote-exec): Checking Host Key: false

aws\_instance.project-instance (remote-exec): Target Platform: unix

aws\_instance.project-instance: Still creating... [30s elapsed]

aws\_instance.project-instance (remote-exec): Connecting to remote host via SSH...

aws\_instance.project-instance (remote-exec): Host: 65.2.91.114

aws\_instance.project-instance (remote-exec): User: ubuntu

aws\_instance.project-instance (remote-exec): Password: false

aws\_instance.project-instance (remote-exec): Private key: true

aws\_instance.project-instance (remote-exec): Certificate: false

aws\_instance.project-instance (remote-exec): SSH Agent: false

aws\_instance.project-instance (remote-exec): Checking Host Key: false

aws\_instance.project-instance (remote-exec): Target Platform: unix

aws\_instance.project-instance (remote-exec): Connected!

aws\_instance.project-instance: Still creating... [40s elapsed]

aws\_instance.project-instance (remote-exec): Wait until SSH is ready

aws\_instance.project-instance: Provisioning with 'local-exec'...

aws\_instance.project-instance (local-exec): Executing: ["/bin/sh" "-c" "ansible-playbook -i 65.2.91.114, --private-key ~/devops.pem nginx.yaml"]

aws\_instance.project-instance (local-exec): [WARNING] Ansible is being run in a world writable directory (/mnt/d/Desktop/project/devops-cloud), ignoring it as an ansible.cfg source. For more information see https://docs.ansible.com/ansible/devel/reference\_appendices/config.html#cfg-in-world-writable-dir

aws\_instance.project-instance (local-exec): PLAY [Install Nginx] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

aws\_instance.project-instance (local-exec): TASK [Gathering Facts] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

The authenticity of host '65.2.91.114 (65.2.91.114)' can't be established.

ECDSA key fingerprint is SHA256:hZU/eBcv2oQ6Vt7117QAahZQ8faOiqIpdfQvb3P7Oh0.

Are you sure you want to continue connecting (yes/no)? yes

aws\_instance.project-instance (local-exec): ok: [65.2.91.114]

aws\_instance.project-instance (local-exec): TASK [nginx : Update and upgrade apt packages] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

aws\_instance.project-instance: Still creating... [50s elapsed]

aws\_instance.project-instance: Still creating... [1m0s elapsed]

aws\_instance.project-instance: Still creating... [1m10s elapsed]

aws\_instance.project-instance: Still creating... [1m20s elapsed]

aws\_instance.project-instance: Still creating... [1m30s elapsed]

aws\_instance.project-instance (local-exec): changed: [65.2.91.114]

aws\_instance.project-instance (local-exec): TASK [nginx : Ensure Nginx is at the latest version] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

aws\_instance.project-instance: Still creating... [1m40s elapsed]

aws\_instance.project-instance (local-exec): changed: [65.2.91.114]

aws\_instance.project-instance (local-exec): TASK [nginx : Make sure Nginx is running] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

aws\_instance.project-instance (local-exec): ok: [65.2.91.114]

aws\_instance.project-instance (local-exec): TASK [copy nginx config file and restart] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

aws\_instance.project-instance (local-exec): changed: [65.2.91.114]

aws\_instance.project-instance (local-exec): TASK [nginx : create symlink] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

aws\_instance.project-instance (local-exec): changed: [65.2.91.114]

aws\_instance.project-instance (local-exec): TASK [nginx : copy website] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

aws\_instance.project-instance (local-exec): changed: [65.2.91.114]

aws\_instance.project-instance (local-exec): TASK [restart nginx] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

aws\_instance.project-instance (local-exec): changed: [65.2.91.114]

aws\_instance.project-instance (local-exec): PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

aws\_instance.project-instance (local-exec): 65.2.91.114 : ok=8 changed=6 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

aws\_instance.project-instance: Creation complete after 1m49s [id=i-050a9c1876e084a29]

Apply complete! Resources: 9 added, 0 changed, 0 destroyed.

Outputs:

public\_dns = ""

server\_ip = "65.2.91.114"