PS E:\devops-challenge\terraform\environments\local> **terraform init**

Initializing the backend...

Initializing modules...

- kubernetes in ..\..\modules\kubernetes

- security in ..\..\modules\security

Initializing provider plugins...

- Reusing previous version of hashicorp/kubernetes from the dependency lock file

- Using previously-installed hashicorp/kubernetes v2.38.0

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.

PS E:\devops-challenge\terraform\environments\local> terraform plan -out plan.tfplan

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:

# module.kubernetes.kubernetes\_deployment.app will be created

+ resource "kubernetes\_deployment" "app" {

+ id = (known after apply)

+ wait\_for\_rollout = true

+ metadata {

+ generation = (known after apply)

+ labels = {

+ "app" = "chat-service"

}

+ name = "chat-service"

+ namespace = "dev"

+ resource\_version = (known after apply)

+ uid = (known after apply)

}

+ spec {

+ min\_ready\_seconds = 0

+ paused = false

+ progress\_deadline\_seconds = 600

+ replicas = "1"

+ revision\_history\_limit = 10

+ selector {

+ match\_labels = {

+ "app" = "chat-service"

}

}

+ strategy (known after apply)

+ template {

+ metadata {

+ generation = (known after apply)

+ labels = {

+ "app" = "chat-service"

}

+ name = (known after apply)

+ resource\_version = (known after apply)

+ uid = (known after apply)

}

+ spec {

+ automount\_service\_account\_token = true

+ dns\_policy = "ClusterFirst"

+ enable\_service\_links = true

+ host\_ipc = false

+ host\_network = false

+ host\_pid = false

+ hostname = (known after apply)

+ node\_name = (known after apply)

+ restart\_policy = "Always"

+ scheduler\_name = (known after apply)

+ service\_account\_name = (known after apply)

+ share\_process\_namespace = false

+ termination\_grace\_period\_seconds = 30

+ container {

+ image = "nikhilreddy99/chat-service:1.0.0"

+ image\_pull\_policy = (known after apply)

+ name = "chat-service"

+ stdin = false

+ stdin\_once = false

+ termination\_message\_path = "/dev/termination-log"

+ termination\_message\_policy = (known after apply)

+ tty = false

+ port {

+ container\_port = 80

+ protocol = "TCP"

}

+ resources (known after apply)

+ security\_context {

+ allow\_privilege\_escalation = true

+ privileged = false

+ read\_only\_root\_filesystem = false

+ run\_as\_non\_root = true

+ run\_as\_user = "1000"

}

}

+ image\_pull\_secrets (known after apply)

+ readiness\_gate (known after apply)

}

}

}

}

# module.kubernetes.kubernetes\_namespace.this will be created

+ resource "kubernetes\_namespace" "this" {

+ id = (known after apply)

+ wait\_for\_default\_service\_account = false

+ metadata {

+ generation = (known after apply)

+ labels = {

+ "env" = "dev"

}

+ name = "dev"

+ resource\_version = (known after apply)

+ uid = (known after apply)

}

}

# module.kubernetes.kubernetes\_service.app will be created

+ resource "kubernetes\_service" "app" {

+ id = (known after apply)

+ status = (known after apply)

+ wait\_for\_load\_balancer = true

+ metadata {

+ generation = (known after apply)

+ name = "chat-service-svc"

+ namespace = "dev"

+ 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" = "chat-service"

}

+ session\_affinity = "None"

+ type = "ClusterIP"

+ port {

+ node\_port = (known after apply)

+ port = 80

+ protocol = "TCP"

+ target\_port = "80"

}

+ session\_affinity\_config (known after apply)

}

}

# module.networking.kubernetes\_network\_policy.allow-same-namespace will be created

+ resource "kubernetes\_network\_policy" "allow-same-namespace" {

+ id = (known after apply)

+ metadata {

+ generation = (known after apply)

+ name = "allow-same-namespace"

+ namespace = "dev"

+ resource\_version = (known after apply)

+ uid = (known after apply)

}

+ spec {

+ policy\_types = [

+ "Ingress",

]

+ ingress {

+ from {

+ namespace\_selector {

}

}

}

+ pod\_selector {

+ match\_labels = {

+ "app" = "chat-service"

}

}

}

}

# module.security.kubernetes\_role.limited will be created

+ resource "kubernetes\_role" "limited" {

+ id = (known after apply)

+ metadata {

+ generation = (known after apply)

+ name = "chat-service-role"

+ namespace = "dev"

+ resource\_version = (known after apply)

+ uid = (known after apply)

}

+ rule {

+ api\_groups = [

+ null,

]

+ resources = [

+ "endpoints",

+ "pods",

+ "services",

]

+ verbs = [

+ "get",

+ "list",

+ "watch",

]

}

}

# module.security.kubernetes\_role\_binding.bind will be created

+ resource "kubernetes\_role\_binding" "bind" {

+ id = (known after apply)

+ metadata {

+ generation = (known after apply)

+ name = "chat-service-rb"

+ namespace = "dev"

+ resource\_version = (known after apply)

+ uid = (known after apply)

}

+ role\_ref {

+ api\_group = "rbac.authorization.k8s.io"

+ kind = "Role"

+ name = "chat-service-role"

}

+ subject {

+ api\_group = (known after apply)

+ kind = "ServiceAccount"

+ name = "chat-service"

+ namespace = "dev"

}

}

# module.security.kubernetes\_service\_account.sa will be created

+ resource "kubernetes\_service\_account" "sa" {

+ automount\_service\_account\_token = true

+ default\_secret\_name = (known after apply)

+ id = (known after apply)

+ metadata {

+ generation = (known after apply)

+ name = "chat-service"

+ namespace = "dev"

+ resource\_version = (known after apply)

}

}

Plan: 7 to add, 0 to change, 0 to destroy.

Changes to Outputs:

+ namespace = "dev"

──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────

Saved the plan to: plan.tfplan

To perform exactly these actions, run the following command to apply:

terraform apply "plan.tfplan"

PS E:\devops-challenge\terraform\environments\local> terraform apply plan.tfplan

module.kubernetes.kubernetes\_namespace.this: Creating...

module.kubernetes.kubernetes\_namespace.this: Creation complete after 0s [id=dev]

module.security.kubernetes\_role.limited: Creating...

module.networking.kubernetes\_network\_policy.allow-same-namespace: Creating...

module.security.kubernetes\_service\_account.sa: Creating...

module.kubernetes.kubernetes\_deployment.app: Creating...

module.security.kubernetes\_role.limited: Creation complete after 1s [id=dev/chat-service-role]

module.networking.kubernetes\_network\_policy.allow-same-namespace: Creation complete after 1s [id=dev/allow-same-namespace]

module.security.kubernetes\_service\_account.sa: Creation complete after 1s [id=dev/chat-service]

module.security.kubernetes\_role\_binding.bind: Creating...

module.kubernetes.kubernetes\_service.app: Creation complete after 1s [id=dev/chat-service-svc]

module.security.kubernetes\_role\_binding.bind: Creation complete after 0s [id=dev/chat-service-rb]

module.kubernetes.kubernetes\_deployment.app: Creation complete after 4s [id=dev/chat-service]

Apply complete! Resources: 7 added, 0 changed, 0 destroyed.

Outputs:

namespace = "dev"

PS E:\devops-challenge\terraform\environments\local> **kubectl -n dev get deployments,services,pods**

NAME READY UP-TO-DATE AVAILABLE AGE

deployment.apps/chat-service 1/1 1 1 3m9s

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

service/chat-service-svc ClusterIP 10.97.216.3 <none> 80/TCP 3m9s

NAME READY STATUS RESTARTS AGE

pod/chat-service-7b6cf6fb59-59gjt 1/1 Running 0 3m8s

PS E:\devops-challenge\terraform\environments\local>