|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| What You Want to Change | File | |  | | --- | |  |  |  | | --- | | **Field to Modify** | |
| ServiceNow instance, username, password | terraform.tfvars | servicenow\_instance, servicenow\_username, servicenow\_password |
| Assignment group for CR | terraform.tfvars, modules/servicenow\_cr/main.tf | assignment\_group |
| CR category, risk, impact | main.tf, modules/servicenow\_cr/main.tf | category, risk, impact |
| Requested by (email) | main.tf | requested\_by |
| Terraform state backend | backend.tf | bucket, key, region |
| Jenkins credentials | Jenkinsfile | credentials('servicenow-user'), credentials('servicenow-pass') |
| GitHub Secrets for ServiceNow API | .github/workflows/cr.yml | SNOW\_USER, SNOW\_PASS |

1. **Change ServiceNow Instance, Credentials, and Assignment Group**

📌 **File:** terraform.tfvars

servicenow\_instance = "your-instance.service-now.com"

servicenow\_username = "your-username"

servicenow\_password = "your-password"

assignment\_group = "ITSM Team"

➡️ **Modify these values** if you have a different ServiceNow instance or assignment group.  
❗ **Use AWS Secrets Manager or Vault instead of hardcoding credentials.**

1. **Modify Change Request Fields**

📌 **File:** modules/servicenow\_cr/main.tf

resource "servicenow\_record" "change\_request" {

table = "change\_request"

fields = {

short\_description = var.short\_description

description = var.description

category = var.category # Change this value (e.g., "Infrastructure")

risk = var.risk # Options: "low", "medium", "high"

impact = var.impact # Options: "low", "medium", "high"

assignment\_group = var.assignment\_group

requested\_by = var.requested\_by

state = "new"

}

}

➡️ **Modify default values** for category, risk, and impact if needed.

1. **Change Module Inputs (Modify Values Passed to the CR Module)**

📌**File:** main.tf

module "create\_cr" {

source = "./modules/servicenow\_cr"

short\_description = "Automated CR via Terraform"

description = "Created for deployment automation"

category = "Software" # Change to "Infrastructure", "Security", etc.

risk = "low" # Change to "medium" or "high"

impact = "low" # Change to "medium" or "high"

assignment\_group = var.assignment\_group

requested\_by = "umrao@example.com" # Change to requester's email

}

➡️ **Modify values for category, risk, impact, and requested\_by.**

1. **Modify Remote State Storage (Backend Configuration)**

📌 **File:** backend.tf

terraform {

backend "s3" {

bucket = "terraform-state-bucket" # Change bucket name

key = "servicenow/terraform.tfstate" # Change path if needed

region = "us-east-1" # Change AWS region

}

}

➡️ **Modify values** if using a different AWS region, bucket, or path.

1. **Modify Jenkins Pipeline (Approval Logic & Email Notification)**

📌 **File:** Jenkinsfile

pipeline {

agent any

environment {

TF\_VAR\_servicenow\_username = credentials('servicenow-user') # Change credential ID in Jenkins

TF\_VAR\_servicenow\_password = credentials('servicenow-pass')

}

stages {

stage('Terraform Init') {

steps {

sh 'terraform init'

}

}

stage('Terraform Apply') {

steps {

sh 'terraform apply -auto-approve'

}

}

stage('Check CR Approval') {

steps {

script {

def crState = sh(script: 'terraform output -raw cr\_id', returnStdout: true).trim()

if (crState == 'approved') {

echo "CR Approved! Proceeding with Deployment..."

} else {

echo "CR Not Approved! Stopping Deployment..."

error("Change Request not approved!")

}

}

}

}

}

}

➡️ **Modify credential IDs**, email recipients, or notification settings if needed.

1. **Modify GitHub Actions Workflow (Trigger CR Creation on PRs)**

📌 **File:** .github/workflows/cr.yml

yaml

CopyEdit

name: Create CR on PR

on:

pull\_request:

types: [opened]

jobs:

create-cr:

runs-on: ubuntu-latest

steps:

- name: Checkout Repo

uses: actions/checkout@v3

- name: Setup Terraform

uses: hashicorp/setup-terraform@v2

- name: Terraform Init & Apply

env:

TF\_VAR\_servicenow\_username: ${{ secrets.SNOW\_USER }} # Change secret name

TF\_VAR\_servicenow\_password: ${{ secrets.SNOW\_PASS }}

run: |

terraform init

terraform apply -auto-approve

➡️ **Modify GitHub Secrets (SNOW\_USER, SNOW\_PASS)** or adjust trigger conditions.