

Terraform DevOps Interview Cheat Sheet

1. Provider Block

```
provider "aws" {  
    region = var.aws_region  
}
```

2. Variables

```
variable "aws_region" { default = "us-east-1" }  
variable "ami_id"      { default = "ami-0c02fb55956c7d316" }  
variable "instance_type" { default = "t2.micro" }  
variable "key_name"     { default = "devops-key" }
```

3. Security Group (Allow SSH)

```
resource "aws_security_group" "allow_ssh" {  
    name           = "allow_ssh"  
    description    = "Allow SSH inbound traffic"  
  
    ingress {  
        description = "SSH from anywhere (dev only)"  
        from_port   = 22  
        to_port     = 22  
        protocol    = "tcp"  
        cidr_blocks = ["0.0.0.0/0"]  
    }  
  
    egress {  
        from_port   = 0  
        to_port     = 0  
        protocol    = "-1"  
        cidr_blocks = ["0.0.0.0/0"]  
    }  
  
    tags = {  
        Name           = "allow_ssh"  
        Environment    = "dev"  
    }  
}
```

```
}  
}
```

4. EC2 Instance

```
resource "aws_instance" "ec2" {  
  ami                = var.ami_id  
  instance_type      = var.instance_type  
  key_name           = var.key_name  
  vpc_security_group_ids = [aws_security_group.allow_ssh.id]  
  
  tags = {  
    Name      = "devops-instance"  
    Environment = "dev"  
  }  
}
```

5. Outputs

```
output "instance_ip" {  
  value = aws_instance.ec2.public_ip  
}
```

6. terraform.tfvars Example

```
aws_region      = "us-east-1"  
ami_id          = "ami-0c02fb55956c7d316"  
instance_type   = "t2.micro"  
key_name        = "devops-key"
```

7. .gitignore Essentials

```
*.tfstate  
*.tfstate.*  
.terraform/  
terraform.tfvars  
.terraform.lock.hcl
```

8. Core CLI Commands

Command	Purpose
<code>terraform init</code>	Initialize working dir
<code>terraform plan</code>	Show execution plan
<code>terraform apply</code>	Apply changes
<code>terraform destroy</code>	Tear down infrastructure
<code>terraform fmt</code>	Format configuration files
<code>terraform validate</code>	Validate configuration

9. Production Notes

- **Tags** for cost, owner, env.
- Use **remote state** (S3 + DynamoDB) in teams.
- Restrict SSH CIDRs in prod.
- Modules for reusable patterns.
- State locking prevents concurrent changes.

DevOps Troubleshooting & Interview Journal

Day 1-3: Common Errors & Fixes

Git & GitHub

- **Error:** `remote origin already exists` \ **Fix:** Use `git remote remove origin` or check `git remote -v` to debug existing remotes.

Terraform

- **Error:** Terraform asking for variable values manually. \ **Fix:** Double-check `terraform.tfvars` and default values. Also ensure variable names match exactly.
- **Error:** Invalid target when destroying resources \ **Fix:** Target name should match `terraform state list`. Example: `terraform destroy -target=aws_instance.resource_name`

AWS CLI

- **Issue:** Terraform can't identify AWS account. \ **Fix:** Run `aws configure` in terminal. Provide **Access Key** and **Secret Key** from IAM user with proper permissions.

SSH Access to EC2

- **Error:** `ssh: connect to host ... port 22: Connection timed out` \ **Fix:** Missing port 22 in security group's inbound rule.
 - **Best Practice:** Modify security group **through Terraform** instead of console to ensure IaC consistency.
-

We'll keep updating this with each new day's challenges, interview-worthy stories, and pro tips