Terraform Cheatsheet

Basic Commands

```
terraform init        Initialize working directory
terraform plan        Show execution plan
terraform apply       Apply the changes
terraform destroy     Destroy managed infrastructure
terraform validate    Validate syntax
terraform fmt         Format code
terraform taint       Mark resource for recreation
terraform state list  List resources in state file
terraform state rm    Remove resource from state
```

File Structure

```
*.tf files (HCL format)
main.tf       - main configuration
variables.tf  - input variables
outputs.tf    - outputs
provider.tf   - provider details
```

Basic Template Example

```
provider "aws" {
  region = "us-east-1"
}
resource "aws_s3_bucket" "my_bucket" {
  bucket = "my-unique-bucket-name"
  acl    = "private"
}
```

Variables

```
variable "region" { default = "us-east-1" }
provider "aws" { region = var.region }
```

Outputs

```
output "bucket_name" { value = aws_s3_bucket.my_bucket.bucket }
```

Data Sources

```
data "aws_ami" "latest_amazon_linux" {
  most_recent = true
  owners = ["amazon"]
  filter { name = "name" values = ["amzn2-ami-hvm-*"] }
}
```

Modules

```
module "s3_module" { source = "./modules/s3" bucket_name = "my-bucket" }
```

Remote State (Backend Example)

```
terraform {
  backend "s3" {
    bucket = "my-tf-state-bucket"
    key    = "state/terraform.tfstate"
    region = "us-east-1"
  }
}
```

Lifecycle Rules

```
resource "aws_instance" "example" {
```

```
  ami           = "ami-12345678"
  instance_type = "t2.micro"
  lifecycle { prevent_destroy = true }
}
```

Provisioners (use carefully)

```
provisioner "local-exec" { command = "echo Hello" }
```

Terraform Tips

- Always use terraform plan before apply
- Use remote state for team collaboration
- Use terraform fmt to keep code clean
- Use modules for reusable code
- Avoid provisioners as much as possible