Terraform AWS Resource Arguments Cheat Sheet

aws_vpc

```
resource "aws_vpc" "main" {
  cidr_block           = "10.0.0.0/16"
  enable_dns_support   = true
  enable_dns_hostnames = true
  instance_tenancy     = "default"
  tags                 = { Name = "main-vpc" }
}
```

aws_subnet

```
resource "aws_subnet" "public" {
  vpc_id                  = aws_vpc.main.id
  cidr_block              = "10.0.1.0/24"
  availability_zone       = "us-east-1a"
  map_public_ip_on_launch = true
  tags                    = { Name = "public-subnet" }
}
```

aws_instance

```
resource "aws_instance" "web" {
  ami                         = "ami-0abc1234567890"
  instance_type               = "t3.micro"
  subnet_id                   = aws_subnet.public.id
  vpc_security_group_ids      = [aws_security_group.web_sg.id]
  key_name                    = "my-key"
  associate_public_ip_address = true
  tags                        = { Name = "web-server" }
}
```

aws_lb

```
resource "aws_lb" "app_alb" {
  name               = "app-alb"
  internal           = false
  load_balancer_type = "application"
  subnets            = [aws_subnet.public.id]
  security_groups    = [aws_security_group.lb_sg.id]
  tags               = { Name = "app-alb" }
}
```

aws_eks_cluster

```
resource "aws_eks_cluster" "eks" {
```

```
  name     = "eks-cluster"
  role_arn = aws_iam_role.eks_role.arn
  vpc_config {
    subnet_ids = [aws_subnet.private1.id, aws_subnet.private2.id]
  }
}
```

aws_ecr_repository

```
resource "aws_ecr_repository" "repo" {
  name                 = "my-app"
  image_tag_mutability = "MUTABLE"
  image_scanning_configuration {
    scan_on_push = true
  }
}
```

aws_cloudwatch_log_group

```
resource "aws_cloudwatch_log_group" "log_group" {
  name              = "/aws/my-app"
  retention_in_days = 7
  tags              = { Name = "my-log-group" }
}
```

aws_ecs_service

```
resource "aws_ecs_service" "app" {
  name            = "app-service"
  cluster         = aws_ecs_cluster.main.id
  task_definition = aws_ecs_task_definition.app.arn
  desired_count   = 2
  launch_type     = "FARGATE"
  network_configuration {
    subnets          = [aws_subnet.private1.id]
    assign_public_ip = true
    security_groups  = [aws_security_group.web_sg.id]
  }
  load_balancer {
    target_group_arn = aws_lb_target_group.tg.arn
    container_name   = "app"
    container_port   = 80
  }
}
```