**1st requirement solution:**

data "aws\_ami" "amazon-2" {  
  most\_recent = true  
  
  filter {  
    name = "name"  
    values = ["amzn2-ami-hvm-\*-x86\_64-ebs"]  
  }  
  owners = ["amazon"]  
}

**2nd requirement solution**

resource "aws\_subnet" "subnet-private" {  
  vpc\_id            = [aws\_vpc.vpc.id](http://aws_vpc.vpc.id/)  
  cidr\_block        = var.subnet-cidr-private  
  availability\_zone = "${var.region}c"  
}  
  
resource "aws\_eip" "ip" {  
  vpc      = true  
}  
  
resource "aws\_nat\_gateway" "nat-gateway" {  
  allocation\_id = "${[aws\_eip.ip.id](http://aws_eip.ip.id/)}"  
  subnet\_id     = "${[aws\_subnet.subnet-a.id](http://aws_subnet.subnet-a.id/)}"  
}  
  
resource "aws\_route\_table" "subnet-private-route-table" {  
  vpc\_id = "${[aws\_vpc.vpc.id](http://aws_vpc.vpc.id/)}"  
  
  
  route {  
    cidr\_block = "[0.0.0.0/0](http://0.0.0.0/0)"  
    gateway\_id = "${[aws\_nat\_gateway.nat-gateway.id](http://aws_nat_gateway.nat-gateway.id/)}"  
  }  
}  
  
resource "aws\_route\_table\_association" "subnet-private-route-table-association" {  
  subnet\_id      = [aws\_subnet.subnet-private.id](http://aws_subnet.subnet-private.id/)  
  route\_table\_id = [aws\_route\_table.subnet-private-route-table.id](http://aws_route_table.subnet-private-route-table.id/)  
}

**3rd requirement solution**

resource "aws\_instance" "instance" {  
  ami                         = [data.aws\_ami.amazon-2.id](http://data.aws_ami.amazon-2.id/)  
  instance\_type               = "t2.small"  
  vpc\_security\_group\_ids      = [ [aws\_security\_group.security-group.id](http://aws_security_group.security-group.id/) ]  
  subnet\_id                   = [aws\_subnet.subnet-a.id](http://aws_subnet.subnet-a.id/)  
  associate\_public\_ip\_address = true  
  user\_data                   = <<EOF  
#!/bin/sh  
yum install -y nginx  
service nginx start  
EOF  
}  
  
resource "aws\_instance" "instance2" {  
  ami                         = [data.aws\_ami.amazon-2.id](http://data.aws_ami.amazon-2.id/)  
  instance\_type               = "t2.small"  
  vpc\_security\_group\_ids      = [ [aws\_security\_group.security-group.id](http://aws_security_group.security-group.id/) ]  
  subnet\_id                   = [aws\_subnet.subnet-c.id](http://aws_subnet.subnet-c.id/)  
  associate\_public\_ip\_address = true  
  user\_data                   = <<EOF  
#!/bin/sh  
yum install -y nginx  
service nginx start  
EOF  
}  
  
resource "aws\_elb" "server\_lb" {  
  listener {  
    instance\_port = 80  
    instance\_protocol = "http"  
    lb\_port = 80  
    lb\_protocol = "http"  
  }  
  
  health\_check {  
    healthy\_threshold = 5  
    interval = 10  
    target = "HTTP:80/"  
    timeout = 5  
    unhealthy\_threshold = 3  
  }  
}  
  
resource "aws\_elb\_attachment" "attach\_ec2\_to\_elb" {  
  elb = "${[aws\_elb.server\_lb.id](http://aws_elb.server_lb.id/)}"  
  #instance = "${format("$${aws\_instance.web\_servers.%[d.id](http://d.id/)}",count.index)}"  
  instance = "${[aws\_instance.instance.id](http://aws_instance.instance.id/)}"  
  depends\_on = ["aws\_instance.instance"]  
}  
resource "aws\_elb\_attachment" "attach\_ec2\_to\_elb\_second\_instance" {  
  elb = "${[aws\_elb.server\_lb.id](http://aws_elb.server_lb.id/)}"  
  instance = "${[aws\_instance.instance2.id](http://aws_instance.instance2.id/)}"  
  depends\_on = ["aws\_instance.instance2"]  
}  
  
  
output "aws\_lb" {  
  value = "${aws\_elb.server\_lb.dns\_name}"  
}