1. Create VPC with CIDR 10.0.0.0/16
2. Create Internet gateway and attach to VPC
3. Create NAT gateway or NAT instance for Private subnet internet
4. Subnets: -
5. **Public Subnets**
6. Create Two Public Subnets in two different AZ (eg ap-southeast-1a and ap-southeast-1b) for public facing traffic Like load balancers or Bastion host
7. CIDR for Public subnets 10.0.1.0/24 and 10.0.2.0/24
8. Create Public Route table and name is public-route
9. Public route table routes are

10.0.0.0/16 local (This is default VPC route)

0.0.0.0/0 IGW all traffic will go to IGW (internet)

1. Associate public subnets (10.0.1.0/24 and 10.0.2.0/24) to public-route
2. Private Subnets: -
3. Create Two private Subnets in two different AZ (eg ap-southeast-1a and ap-southeast-1b)
4. CIDR for Private subnets 10.0.3.0/24 and 10.0.4.0/24
5. Create Private route table and name is private-route
6. Private route table routes are

10.0.0.0/16 local

0.0.0.0/0 NAT-IGW

1. NAT route rule is for internet to private subnet instances.
2. Associate above subnets (10.0.3.0/24 and 10.0.4.0/24) to private-route.
3. DB Subnets: -
4. Create Two DB subnets in two different AZs ( ap-southeast-1a and ap-southeast-1b)
5. CIDR for DB subnets 10.0.5.0/24 and 10.0.6.0/24
6. Create DB Route table and table name is db-route
7. DB Route table routes are
8. 10.0.0.0/16 local

0.0.0.0/0 NAT-IGW (This is optional, If we need internet for software updates and patching) If you don’t want internet just remove this rule. Whenever required just ass this rule)

1. Associate above subnets (10.0.5.0/24 and 10.0.6.0/24) to db-route table.
2. Here My assumption is RDS with Multi AZ. Multi AZ DB subnet group requires two two subnets for HA
3. VPC end Point: -
4. Create VPC end point
5. Select service name as com.amazonaws.ap-southeast-1.s3.
6. Select VPC (10.0.0.0/16)
7. Select all route tables for this VPC (public-route, private-route, db-route)
8. Select full policy or custom policy
9. Create VPC end point.
10. This VPC end point is used for all vpc resources will route internally to all s3 resources.
11. S3 Bucket: -
12. Create s3 bucket
13. Create s3 bucket with policy allow this vpc end point only. So only this vpc resources will access this s3 bucket
14. Ref for s3 bucket policy (<https://aws.amazon.com/premiumsupport/knowledge-center/block-s3-traffic-vpc-ip/>)
15. Bastion Host: -
16. Launch Bastion host or jump host in any public subnet (10.0.1.0/24 or 10.0.2.0/24)
17. Created security group in VPC (10.0.0.0/16). Below are security group rules

Allow ssh (default port 22) any (0.0.0.0/0) ip. (Inbound)

All all (outbound)

Can restrict outbound rules also. Bastion host should require public ip (EIP). Name of security group is bastion-sg. And attach to bastion host instances. (Linux is Port 22 and with Windows is RDP port)

1. Security Groups: -
2. Private subnet SG allow SSH (port 22) from bastion host SG or bastion host private IP (Inbound)
3. Application ports like 80,443 from ELB SG or internal subnets depends on requirements
4. DB-sg allow from application internal subnets for port 3306(mysql) or depends on requirements.

SSH access can be controlled vy using VPN site to site tunnel or VPN client (openvpn or printunl) or Bastion host.

By default IN VPC all subnets can talk to each other (default route rule 10.0.0.0/16 local). Can be controlled by using acls of subnets by deny particular subnet. Depends on business requirements.

Security for PUBLIC cloud: -

L3 Security: - Security groups and ACLS. AWS shield used for protection against for DDOS attacks

L7 Security: - AWS WAF for Sql injection and Cross side Scripting (XSS) , BOTS and etc

KMS: - AWS KMS can be used for encryption and decryption of data

Cloud Front: - It is CDN, But can be used for restricted access for origin (ELB or S3).

For Terraform tool, I am much more comfortable on ansible. I can recreate above scenario by using ansible.