Step-1: Create Virtual Machine

1. Go to portal.azure.com and create virtual machines.
2. On the homepage select create resource 🡪 create 2 “Windows Server” virtual machines.
3. While creating the VMs you need to provide few data like resource group, VM name, region, availability set, specifying fault domain, size, user credential for the VM, enabling ports and inbound rules, disk allocation etc.
4. Similar way, create 2 Linux VMs.

Step-2: Create Load-Balancers

1. On the azure homepage, select create resource and search for “Load balancer” and create the Load balancer network resource. While creating provide few data like resource group, type of the load balancer, VNET and subnet details.
2. Create 2 different load balancers. One public and one internal.
3. Configure the load balancer backend pool with the virtual machine from the availability set.

Step-3: Create SQL Database

1. Similar way, create the SQL Database as a resource under the same resource group and provide necessary details like server configuration details, login parameters, collation etc.
2. Create 2 SQL databases. One will be the primary and the other will be for secondary(failover) in such a way that both the SQL database resource will be in different region which later on will be helpful in case of failover.

Step-4: Configure Automatic Failover

1. The primary database remains as a writable whereas the secondary one is readable until unless it’s failover.
2. From Geo replication, create a failover group and define your primary and secondary databases.
3. Enable the automatic failover and set grace period as well.

Step-5: Configure Load balancer for public and internal traffic

1. Install IIS webservers on the windows VMs created in step-1.
2. Set up health probe for the backend pools such that it listens the port configured in the health probe with respect to the VM. Enable the port in the VMs as well such that it should allow traffic through the same configured port.
3. Add the load balancer rule for the public or front-end servers which will act as the public load balancer.
4. Similar way, configure the health probe and load balancer rule for the backend server traffic which will act as the internal load balancer.

So once any application is deployed it will act as a 3-tier architecture-based environment. This is a basic set up. There can be many improvement areas where we can have Front end and backend servers with different subnets or some sort of fire wall restriction and backup and recovery of web and application servers etc.