Setting up MemoryDB.

Article: https://medium.com/@jedrzejalchimowicz/amazon-memorydb-for-redis-with-python-a-complete-guide-f06d6030e5aa
We will configure 4 components:

- MemoryDB instance
- EC2 instance - we will connect to MemoryDB from here
- Security group for MemoryDB
- Security group for EC2

Instructions:

1. Connect to the AWS console.
2. Create your EC2 instance. Chose a VPC of your choice, we will place MemoryDB in the same VPC. Make sure you are able to connect to the EC2 and run python scripts on it.
3. Create 2 security groups, let's leave them with no rules for now:
    a. One for the EC2, name it "memDB-test-ec2"
    b. One for the database, name it "memDB-test-db"
4. Configure the "memDB-test-ec2" security group:
    a. Add an output rule: Custom TCP, Port range: 6379 (default for MemoryDB), Destination: Custom, in the next box start typing "sg-" and chose the "memDB-test-db" security group.
5. Configure the "memDB-test-db" security group:
    a. Add input rule: Custom TCP, Port range: 6379 (default for MemoryDB), Destination: Custom, in the next box start typing "sg-" and chose the "memDB-test-ec2" security group.
    b. Add output rule: All traffic, and leave the remaining fields at default
6. Create the MemoryDB database:
    a. Go to: Amazon MemoryDB for Redis -> Clusters -> Create cluster
    b. Select Create new cluster (don't pick Demo)
    c. Set the name to: "memoryDB-test"
    d. Subnet groups: create a subnet group. Chose any subnets that are in the same VPC as your EC2 instance. Ideally they should all be private subnets.
    e. Cluster Settings: leave at default
    f. Press Next
    g. In Selected Security Groups press Manage
    h. Choose the "memoryDB-test-db" security group
    i. In Access control lists (ACL) chose open-access
    j. Leave rest at default
    k. Press Next and Create
    l. Cluster creation can take some time on AWS, usually around 20-30min.
7. That's the end of the setup. Once the cluster is created, we should be able to connect to it from our EC2 machine.