AWS Microservice ML Deployment

# Microservices

1. Make EC2 Ubuntu instance (use existing key-pair is fine).
2. Set up security:
   1. Go to AWS UI, running compute instance
   2. Edit inbound rules
   3. All traffic, anywhere for both IP4 & IP6
3. Use 2 separate instances of PuTTY to:
   1. Access the machine on the server, and

sudo apt-get update

sudo apt install golang-go

git clone https://github.com/palomasoftware/MSDS434\_Golang\_MicroserviceIntroduction.git

vi intro.go

go run intro.go

* 1. Run the client to that machine

Apt curl -X GET HTTP://ec2-34-222-53-60.us-west-2.compute.amazonaws.com:8080/TEST (change to the VM DNS you’re running)

1. Now go back to access the machine on the server PuTTY instance and:

sudo apt install golang-github-gorilla-mux-d

go mod init gorilla/mux

cat go.mod

go mod tidy

go get github.com/gorilla/mux

vi intro\_two.go

go run intro\_two.go

ls

cat intro.go

ls

cat intro\_two.go

ls

cd go

ls

cd ..

ls

vi intro.go

vi intro\_two.go

history

1. Go to other client and:

curl <http://ec2-34-222-53-60.us-west-2.compute.amazonaws.com:8080/resources/123>

curl <http://ec2-34-222-53-60.us-west-2.compute.amazonaws.com:8080/resources/456/values>

# Microservices: Reading & Writing JSON to Data Source

1. Make T2 Ubuntu instance (use existing key-pair is fine).
2. Connect to instance via PuTTY
3. Clone repo: <https://github.com/palomasoftware/MSDS434_ModuleSeven_parttwo_GolangMicroserviceJson.git>
4. Set up security:
   1. Go to AWS UI, running compute instance
   2. Edit inbound rules
   3. All traffic, anywhere
5. Command: go run main.go
6. Go to EC2 instance, grab DNS, paste “:9090” at the end of the DNS, put in browser to see it running
7. Download Postman
8. Put the DNS in Postman along with a single record formatted JSON-style:
   1. You should change the existing record so you can have evidence of it changing when you check the browser again (once you make the POST request)
   2. 