

Golang backend Task

A little overview.

What is the task.

To create a file handling server that take name and size,

Get consensus based on raft and update the entities to redis database.

Logging with hclog.

Degree of Freedom:

- Update only if they are new or modified.

- If same entity tell the user it is duplicate.

Packages or Libraries used.

Golang Echo for api , routing , context.

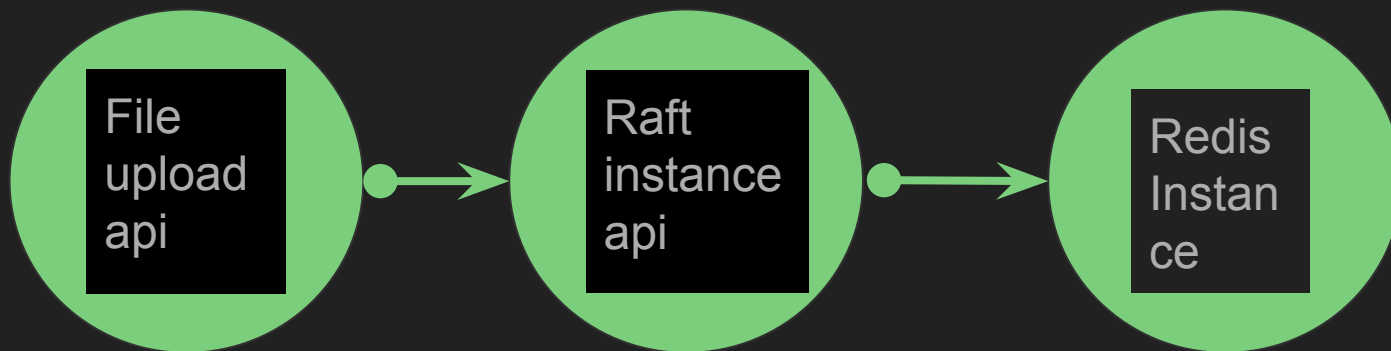
<https://github.com/shaj13/raft> as Raft implementation. (hcraft and etcd raft also available , but chosen for its easy to use feature.

Redis stack docker image.

Hclog package.

Other small helper packages.

Flow



Fileserver:
127.0.0.1:8000
User interacts

Raftinstances:
127.0.0.1:9090 / 91
/92
Fileserver intracts

127.0.0.1:6379
Localhost:8081 / web ui
Raft leader interacts

Execution

Docker compose up // redis-stack

raftinstance> ./raftinstance -state_dir=\$TMPDIR/1 -raft :8080 -api :9090 //for single instance.

./raftinstance -state_dir \$TMPDIR/3 -raft :8082 -api :9092 -join :8080 / to add new instance after running one node.

filesarva> ./filesarva //for file upload api.

For building from code.

It is normal golang build process.

Navigate to folders filesarva and raftinstance run following:

Go mod tidy

Go build .

Architectures

Tried to follow Hex architecture as close as possible for both raftinstances and filesarva , but principalities might not be completely implemented.

Hclog is passed to context , so that other internal methods can use for logging data , without creating new instances every time.

Hclog will write the logs to the file in disk , for further processing when required.

Please see demo video for execution and outputs.

Feel free to reachout.

Mallampati.Naveenbhargav@gmail.com

+918008513742

Thank you.

Naveenbhargav Mallampati.