

Heartbeat Server

Background

At some point, services have to be monitored. In this environment one needs a reliable, repeatable way to track of the health of production systems. A simple way to handle this is to send a periodic ping across the network to all of the registered systems, wait for a response from the service, and then record the status of the service.

Task

We would like you to design and code this heartbeat service in Python. The requirements for this task are purposely left vague as we would like to see your design skills in action. We aren't necessarily looking for the highest quality code as we are more so looking for a high quality and well thought out design.

Some common use cases to consider:

1. Web server registers with the heartbeat server but cannot properly handle the heartbeat event (returns an error)
2. Web server is registered with the heartbeat server, loses connection, and then re-registers. We want this to be tracked as one web server that disconnected and reconnected, not two separate web servers.
3. The heartbeat server goes down and when it comes back up it should remember all the servers from before.

Recommendations

We know this may be a daunting task if you do not have extensive web development experience, so we have a few recommendations to make the task easier.

1. Use a web framework – We aren't interested in your ability to write a web server (although bonus points if you do!), so use a common web framework to make your life easier. We recommend Flask, Bottle, or Django (Django is the most complicated of the three, so one of the first two if you don't have much experience with Django.)
2. Make it simple and then add features if you feel necessary – have your heartbeat server send pings every 30 seconds, then extend your solution to allow web servers to can tell the heartbeat server how often to ping them.