Single machine, multiple processes

*At the most basic level, we can run a microservice application as multiple processes on a single machine. Each service listens to a different port and communicates over a loopback interface.*

*​This simple approach has some clear benefits:*

* *Lightweight: there is no overhead as it’s just processes running on a server.*
* *Easy troubleshooting: everything is in the same place, so finding a problem or reverting to a working configuration in case of trouble is very straightforward, if you have continuous delivery in place.*
* *Fixed billing: we know how much we’ll have to pay each month.*

*The approach works best for small applications with only a few microservices.*

*Drawbacks:*

* *No scalability: once you max out the resources of the server, that’s it.*
* *Single point of failure: if the server goes down, the application goes down with it.*
* *Fragile deployment: we need custom deployment and monitoring scripts to ensure that services are installed and running correctly.*
* *No resource limits: any microservice process can consume any amount of CPU or memory, potentially starving other services and leaving the application in a degraded state.*