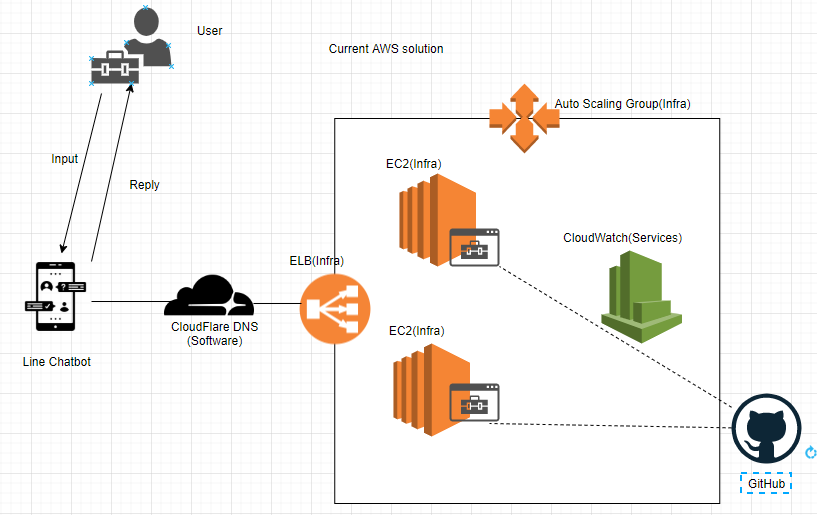
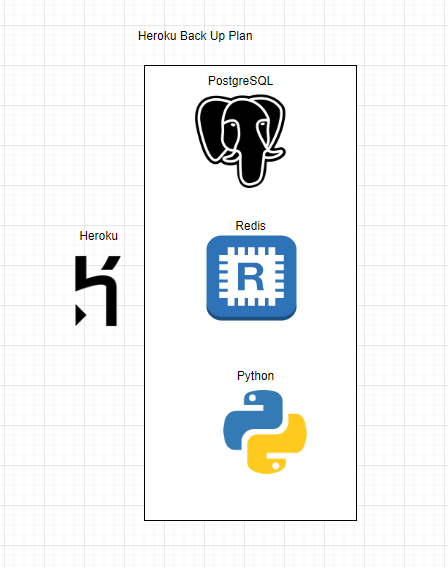
1. How is your project architecture related to the theory taught in the lecture?

Software architecture

1. Git system
   1. Add command
   2. Commit command
   3. Push command

Hardware architecture





The line chatbot is Client-Server Model that partitions tasks or workloads between providers of a service and service requesters.

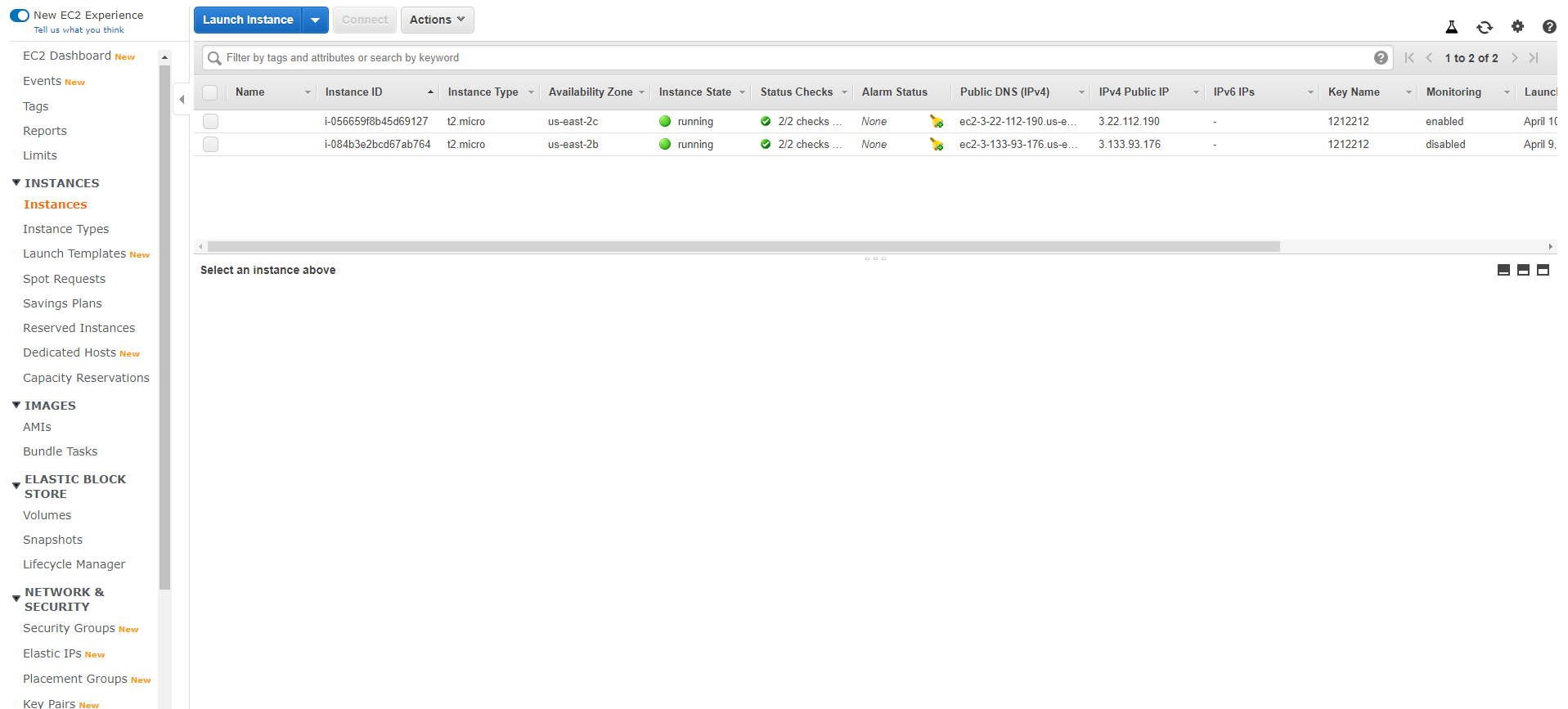
For communication paradigms, the line chatbot is Indirect Communication via a third party. It is space uncoupling as the senders do not need to know who they are sending to and it is time uncoupling as the senders and receivers do not need to exist at the same time. There are message queues that producer processes can send messages to a specified queue, and consumer processes can receive messages from the queue.

For the architectural patterns, it is three-tier architecture. There is a one-to-one mapping from logical elements to physical elements. Each tier has a well-defined role. Tier 1 is mobile device and personal computer for user view and control the line chatbot. Tier 2 is application server for the line chatbot application logic. Tier 3 is database server for database manager. The application is the line messenger itself. In mobile version of line chatbot, Apple IOS and Google Android are used as operating system; while in desktop version of line chatbot, Microsoft Windows and Apple macOS are used as operating system.

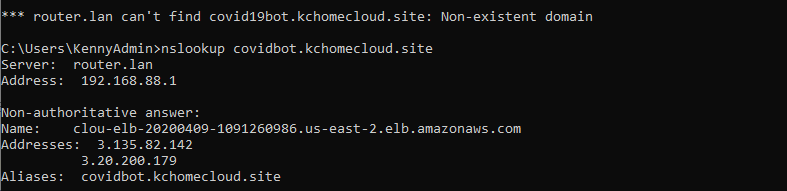
It is an asynchronous distributed system as there is no bounds on process execution speeds, message transmission delays and clock drift rates. Omission failure may occur when a process or communication channel fails to perform pre-defined line chatbot actions. Process Omission Failure exist when a process is halted and does not execute any further steps of its program. Other processes cannot detect the failure, but only indicate that the suspicious process is not responding. Communication Omission Failure exist if it does not transport a message from outgoing message buffer to incoming message buffer. This is known as dropping messages which is generally caused by a lack of buffer space at the receiver or at an intervening gateway, or by a network transmission error.

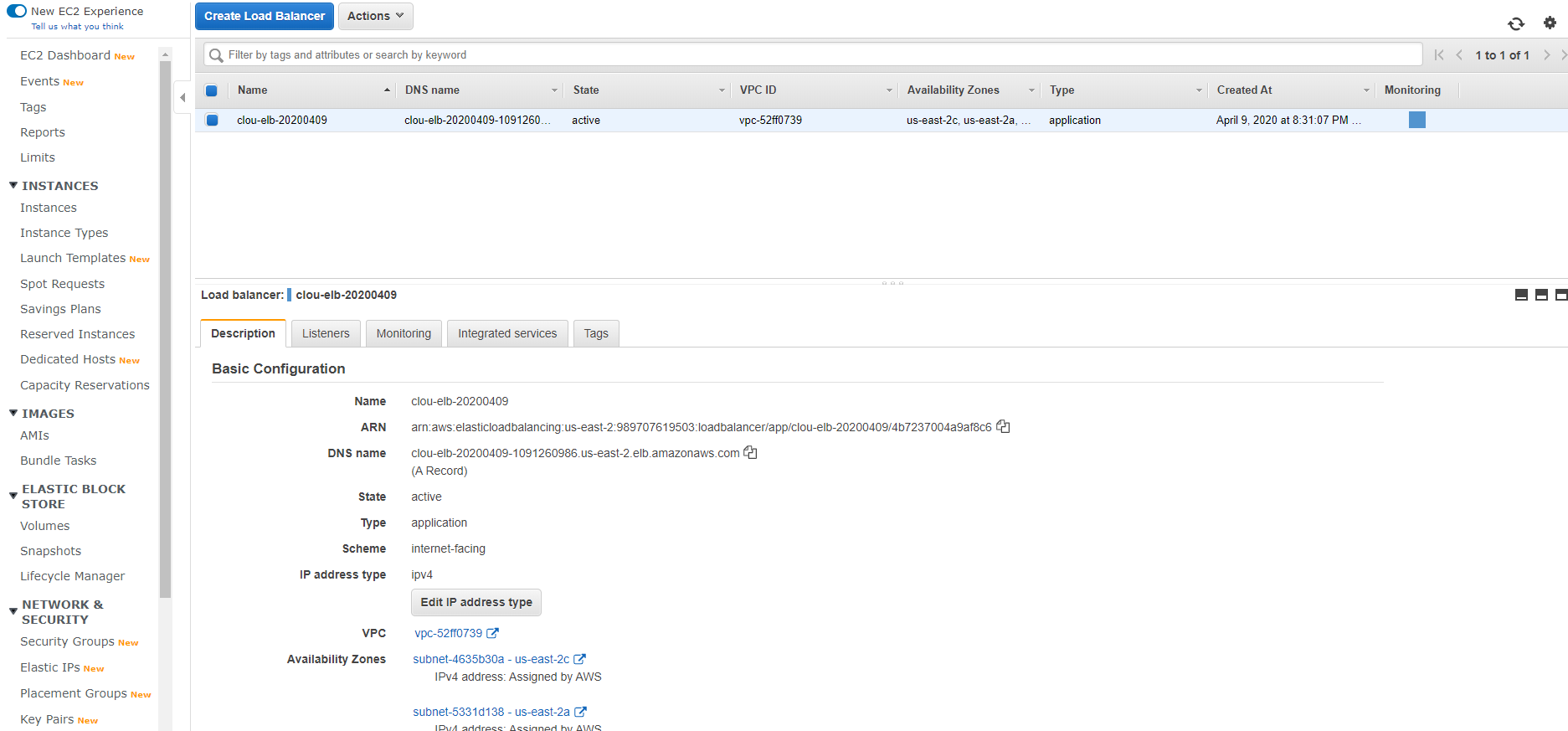
2) Can you demonstrate, with some screen cap, how to increase capacity of your chat bot service?

2 EC2



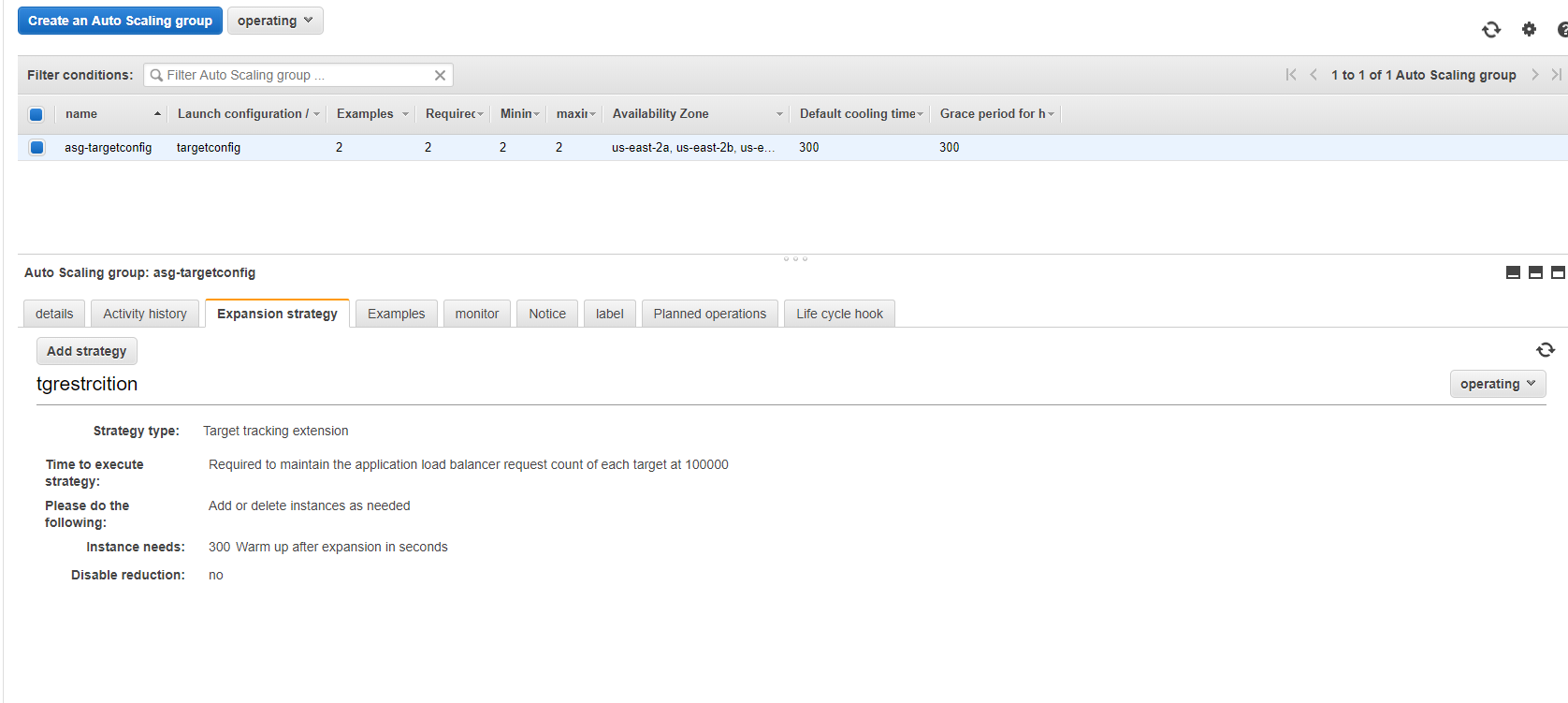
Elastic Load Balancer(Proof)





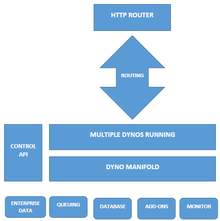
Autoscaling





3) Can you identify if you bot is one of the example of PaaS, IaaS, SaaS? Explain your answer.

(Before) Heroku is a cloud platform as a service (PaaS) supporting several programming languages



(After) IaaS. All application server, elb, autoscaling(failover) do in AWS.