

# Lab: Intro to Application Services

## Application integration

### Step functions

Make it easy to orchestrate bunch of microservice in a particular order. You define your application visually as series of step and then you can deploy it. Say you only want microservice B to run after microservice A is run, you can do that.

### Simple WorkFlow Service

Similar to step function, coordinate multiple component for an application. However, use step function is better and easier rather than SWF

### Simple notification service

Hub-sub service. Create topic and user can subscribe to that topic, when you publish a message to the topic, users who are subscribed to the topic will be notified with the message you have published.

Also be used for push notification for mobile devices.

### Simple Queue Service

Fully managed message queueing service, make it easy to decouple application from demand. You can store demand (in form of requests) inside Simple Queue Service, and then forward those request to the servers.

There are two types of queues. Standard queues and FIFO queues. Standard queues is best effort delivery, meaning messages inside queue might be lost, but it will be at-least once delivery. The ordering aren't guaranteed because of message lost.

FIFO queue on the other hand guarantees that each message will only be processed once, and the ordering preserves. First in, then you're first out.

## Process decoupling example

So let's say we got bunch of servers that is processing say requests. And it is backed by a auto scaling service. Without a queue, the request have chances of overwhelming the instances, because auto scaling might not be able to keep up with the sudden spike in demand since it will take 5/10 minutes to spin up an instance. While the surge in demand could period of less than a

minute.

**So request will be dropped if we don't do something.** How do we solve this? We can store those requests into a queue, so that if the servers are overwhelmed the requests will not be dropped. Instead, a CloudWatch metric is set up to watch the queue, if the request are increasing then it will trigger an auto scaling group to spin up more instances, and then the queue can finally forward those request to the free instances. Now when the demand decreases, the instances that are created can be terminated.

# Customer engagement services

## Amazon connect

Drag-and-drop graphical user interface. Allow you to define customer interaction without any coding at all.

## Amazon pinpoint

Sent notification and email for targeted marketing campaign.

## Simple email service (SES)

Cloud based bulk email sending service. Mass email sending.

---

Revision #1

Created 2023-01-12 19:23:10 UTC by Tamarine

Updated 2023-02-06 18:24:02 UTC by Tamarine