Cloud Architectures
Q: What are cloud architectures?

Three Primary Design Patterns for Cloud Architectures:
[1]:

[2]:

[3]:

Monolithic Software Architecture:
Characteristics:

A (Small) Monolithic Example: Illinois Open Source Queue
Endpoint: https://queue.illinois.edu/
Source Code: https://github.com/illinois/queue

Microservice Software Architecture:
Characteristics:

Microservice Example:
PiggyMetrics by Alexander Lukyanchikov
GitHub: https://github.com/sqshq/PiggyMetrics
via: https://github.com/davidetaibi/Microservices_Project_List

Serverless Software Architecture:
Characteristics:

Serverless Examples:

```
provider:
  name: aws
  runtime: nodejs4.3
  region: us-east-1

functions:
  tweetPet:
    handler: handlers/tweetPet.tweetPet
    description: Tweets Adoptable Pets on a Schedule
    memorySize: 512
    timeout: 10
    events:
      - schedule: rate(6 hours)
```

adoptable-pet-bot/serverless.yml
https://github.com/lynnaloo/adoptable-pet-bot/blob/master/serverless.yml
**emojibot/serverless.yaml**

```yaml
provider:
  name: aws
  region: eu-west-1
  stage: dev
  runtime: nodejs14.x
  memorySize: 128
  lambdaHashingVersion: 20201221
  environment:
  - BOT_ACCESS_TOKEN: ${env:BOT_ACCESS_TOKEN}

functions:
  event:
    handler: src/handler.event
    events:
      - http: POST /event
  explain:
    handler: src/handler.explain
    events:
      - http: POST /explain
```

**serverless-image-labeller/serverless.yml**

```yaml
provider:
  name: aws
  logs:
    restApi: true
  runtime: python3.7
  stage: dev
  region: us-east-1

functions:
  labelOnS3Upload:
    handler: handlers/S3UploadHandler.labelOnS3Upload
    events:
      - s3:
          bucket: ${self:provider.environment.SERVERLESS_IMAGE/labelling.BUCKET}
          event: s3:ObjectCreated:*
          existing: true

  getImagesByLabel:
    handler: handlers/getImagesByLabelHandler.getImagesByLabel
    events:
      - http:
          path: getImagesByLabel
          method: post
          cors: true
```

<table>
<thead>
<tr>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Monolithic Software Architectures</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Microservice Software Architecture</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Serverless Software Architecture</strong></td>
<td></td>
</tr>
</tbody>
</table>

**Remember:** These architectures are design patterns and software projects may have influences from multiple design patterns:
- A microservice architecture may evolve into pieces of software that take on a monolithic architecture.
- A microservice architecture may make some calls to serverless components.
- ...etc...