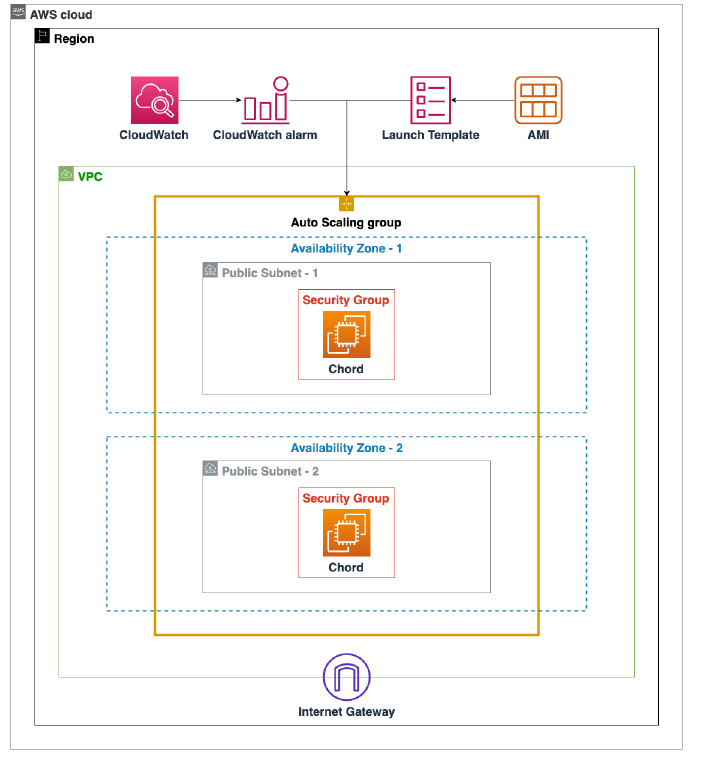
In the first phase,

* run a file server on a single AWS instance without enabling auto-scaling

In the second phase

* distributed file system should automatically scale new EC2 instances running new file servers, based on the disk usage of each instance

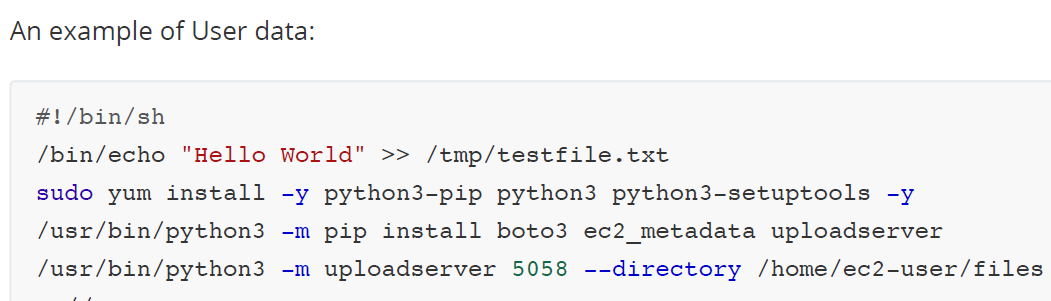
Requirements: 上面兩階段都結束

* 應該就可以把Chord整合進去，整合完後最終會長這樣
  + 

**Phase1: Running the First Node on AWS Cloud**

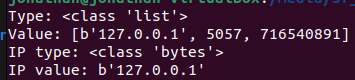
* 先創建IAM
* 再創立Security Group
  + 要修改Inbound Rule
* 創建EC2
  + 創建成功後用SSH連連看，之後安裝東西也是用這個client
* Install各種東西
  + Amazon CloudWatch Agent: 應該是監控log用
  + File Server: python的一個套件
    - 安裝完後也會Run File Server
  + some Additional Requirements: 應該是python會用到的套件
* (Optional) Launch CloudWatch Wizard
  + 不太懂在幹嘛
* Enable CloudWatch Agent
  + 其實就是修改config檔的參數
  + **有寫一個cloudwatch.json要上傳**
* (Optional) CloudWatch Verification
  + 因為CloudWatch的配置生效需要時間，可以用指令確認是否完成
    - sudo yum install telnet -y
    - telnet monitoring.ap-northeast-1.amazonaws.com 443
* Check CloudWatch Metrics
  + 在GUI確認Cloud Metrics
    - Remember to change the metric period to one minute in the tab "Graphed metrics"
    - 應該是要點GUI中的Graphed Metrics去將週期改成1分鐘
* Test your File Server from Local or Another EC2 instance
  + 該裝的都裝了，用SSH Client來執行測是腳本，確認File Server有在正常運行
  + 要上傳的 a.txt 應該要自己寫創建

**Phase 2: Enabling Auto-scaling**

* Create AMI (Amazon Machine Image)
  + 這是用GUI操作
  + 會Based on前面建立的EC2實例
    - 在最後的步驟「Create Image」可以選擇Based on誰
* Launch an Auto-scaling Group (我覺得比較像是寫好Launch Template)
  + 先用GUI到指定地方
  + 然後要修改一些東西，讓他符合我們要自動啟動的服務
    - Ex: User data
      * 
* Create Auto-scaling Group
  + 一樣GUI設定一些東東
  + Scaling policies: you can attach it later (不太懂)
  + 助教建議: 可以寫一個 .py 來做這一步驟，不知道怎麼做
    - recommended to write the create\_auto\_scaling\_group.py script and run it.
    - 
      * ChatGPT 有嘗試寫一個腳本
* Create CloudWatch Alarm and Attach it to Auto-scaling Group
  + 也是GUI設定東東，有兩個要設定
    - Go to EC2 > Auto Scaling Group
    - Go to Cloudwatch > Alarms > Create alarm
  + 也是建議可以寫個 .py 來自動化，感覺好像會常常創建/刪除
    - recommended to write the create\_scaling\_policy.py script and run it.

**Requirements: integrate Chord nodes into the system.**

* 這次Cloud node不會只有File Server (Port 5058)，要在port 5057監聽Chord node
* Cloud node應該只有負責啟動 Chord node (也就是我們用的 ./chord 127.0.0.1 5057這種指令)，並未負責Join node，想Join node 需要:
  + To join the existing Chord ring system you can use
    - 1. AutoScaling.Client.describe\_auto\_scaling\_groups to find IDs of all running instances in the Auto-scaling Group,
    - 2. and then use EC2.Client.describe\_instances to obtain their public IP addresses.
    - 感覺是要寫 python 在 chord node啟動後將node加入ring中，但是不確定要怎麼用join()
      * 之前的join()是這樣:
        + client\_2.call("join", client\_1.call("get\_info"))
        + 這個get info會回傳Node的資料結構，裡面包含IP、Port、ID
        + Node資料結構可以用Python重現?



應該可以，好像單純是 list

Byte類型的IP可以直接 ip = b'127.0.0.1'

* + - * + 這些資訊可以得知?

我們知道Port = 5057

IP 應該可以透過EC2.Client.describe取得

ID呢?自己透過hash去算?

* + - 設定完之後就可以用upload.py、download.py去測試