

## Devcloud

### TD1 :

Téléchargement du GIT

docker-compose build ou -d up

docker-compose run

curl -H(Pour accepter le header) 'accept:json'(Accepte le format json) 'https://ec2.shop?region=us-west-1&filter=t1,t2.micro' | jq -r '.' → affiche au format json la query

```
>>> {
  "Prices": [
    {
      "InstanceType": "t2.micro",
      "Memory": "1 GiB",
      "VCPUS": 1,
      "Storage": "EBS only",
      "Network": "Low to Moderate",
      "Cost": 0.0138,
      "MonthlyPrice": 10.074,
      "SpotPrice": "0.0041"
    },
    {
      "InstanceType": "t1.micro",
      "Memory": "0.613 GiB",
      "VCPUS": 1,
      "Storage": "EBS only",
      "Network": "Very Low",
      "Cost": 0.025,
      "MonthlyPrice": 18.25,
      "SpotPrice": "0.0025"
    }
  ]
}
```

Les données représente les prix des machines cloud.

Voilà les données du m6a.48xlarge :

```
{
  "InstanceType": "m6a.48xlarge",
  "Memory": "768 GiB",
  "VCPUS": 192,
  "Storage": "EBS only",
  "Network": "50000 Megabit",
  "Cost": 8.2944,
  "MonthlyPrice": 6054.911999999999,
  "SpotPrice": "3.4279"
}
```

Voilà les données du t1.micro :

```
{
  "InstanceType": "t1.micro",
  "Memory": "0.613 GiB",
  "VCPUS": 1,
  "Storage": "EBS only",
  "Network": "Very Low",
  "Cost": 0.02,
}
```

## Devcloud

```
"MonthlyPrice": 14.6,  
"SpotPrice": "0.0020"  
}
```

14,6\$ pour tiny et 6054,9\$ pour le m6a.

2/

-r permet de le sortir en JSON string donc sans les quotes.

Pour sortir le json hors prices : jq 'map(.)' medium.json ou jq '.Prices' medium.json

Machine à bas coûts 'spot', vous donne c'est machines uniquement lorsque sont libre. Par contre sont éteinte à n'importe qu'elle moment.

Pour afficher toute les clé dans .Prices :

```
jq -r '.Prices |  
map([.InstanceType, .Memory, .VCPUS, .Storage,.Network, .Cost, .MonthlyPrice, .SpotPrice])'  
medium.json
```

J'affiche au format csv :

```
jq -r '.Prices[] |  
[.InstanceType, .Memory, .VCPUS, .Storage,.Network, .Cost, .MonthlyPrice, .SpotPrice]|@csv'  
medium.json
```

jq -r '.Prices[] | keys ou values|csv' medium.json → affiche les keys.

jq -r '.Prices[0] | keys ou values|csv' medium.json → affiche juste les champs.

jq -r '.Prices[]| join(",")' medium.json → rajoute des ','.

jq -r '.Prices[] | [keys[] as \$k | .[\$k]]|@csv' medium.json → selectionne et mets les champs dans une variable puis les affiche en @csv.