July 18, 2022

Jean-michel Girard

Ticksmith inc.

AWS Costs

GTV Environments

Table of Contents

[Overview 2](#_Toc109048977)

[Instances 3](#_Toc109048978)

[EC2 3](#_Toc109048979)

[Autoscalling 4](#_Toc109048980)

[MQ (AmazonMQ / ActiveMQ) 5](#_Toc109048981)

[Tailles d’instances 5](#_Toc109048982)

[RDS 5](#_Toc109048983)

[Tailles d’instances 5](#_Toc109048984)

[Cloudwatch 6](#_Toc109048985)

[Agent CloudWatch 6](#_Toc109048986)

[Métriques personnalisées 6](#_Toc109048987)

[Métriques dupliquées/calculables 6](#_Toc109048988)

[Sélection des métriques 6](#_Toc109048989)

[Agent configuration 7](#_Toc109048990)

[Autres Recommendations 8](#_Toc109048991)

[AWS Compute Optimizer 8](#_Toc109048992)

[AWS Systems Manager Compliance & Patches 8](#_Toc109048993)

# Overview

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Service (USD) | DEV | UAT | PROD | Total | Recommendations | Est. savings |
| Elastic Compute Cloud | $ 1,243.74 | $ 969.32 | $ 1,470.00 | **$ 3,683.06** | 1. Enable AWS Compute Optimizer in the root account for a minimal duration of 14 days. Past that delay, adjust all device sizes and scaling groups 2. Create dynamic scaling rules to scale with the demand. | **$1000+** |
| CloudWatch | $ 539.58 | $ 248.53 | $ 398.06 | **$ 1,186.17** | 1. Fix CloudWatch agent. Most expansive metrics are:  - disk\_free (42%)  - disk\_used\_percent (42%)  - mem\_used\_percent (5.5%)  - swap\_used\_percent (5%) 2. Configure the namespaces 3. Manage the agent’s config with SSM 4. Reduce granularity when possible | **$500** |
| MQ | $ 207.40 | $ 414.74 | $ 414.75 | **$ 1,036.89** | 1. Use micro instances (single\_instance mode) in development + uat 2. Consider the load in prod. Might be resizable as well | **$760** |
| Relational Database Service | $ 467.09 | $ 246.17 | $ 266.16 | **$ 979.42** | 1. Reduce sizes for Airflow | **$525** |
| Simple Storage Service | $ 276.61 | $ 49.99 | $ 14.82 | **$ 341.42** | 1. Try S3 Intelligent-Tiering storage class (maybe) |  |
| Elastic Load Balancing | $ 81.06 | $ 81.03 | $ 81.04 | **$ 243.13** |  |  |
| Config | $ 130.73 | $ 41.00 | $ 1.45 | **$ 173.18** |  |  |
| ElastiCache | $ 24.48 | $ 24.48 | $ 24.48 | **$ 73.44** |  |  |
| Simple Queue Service | $ 23.13 | $ 19.49 | $ 0.22 | **$ 42.84** |  |  |
| Key Management Service | $ 9.99 | $ 7.47 | $ 7.53 | **$ 24.99** |  |  |
| Data Transfer | $ 10.13 | $ 1.76 | $ 3.24 | **$ 15.13** |  |  |
| Route 53 | $ 0.61 | $ 1.74 | $ 1.98 | **$ 4.33** |  |  |
| S3 Glacier Deep Archive | $ - | $ 1.74 | $ 1.98 | **$ 3.72** |  |  |
| DynamoDB | $ 0.08 | $ - | $ - | **$ 0.08** |  |  |
| Lambda | $ 0.05 | $ - | $ - | **$ 0.05** |  |  |
| Simple Notification Service | $ 0.03 | $ - | $ - | **$ 0.03** |  |  |
| CloudTrail | $ - | $ - | $ - | **$ -** |  |  |
| API Gateway | $ - | $ - | $ - | **$ -** |  |  |
| **Total tax** | **$ 453.60** | **$ 315.34** | **$ 401.87** | **$ 1,170.81** |  | **$ 360+** |
| **Total** | **$ 3,468.31** | **$ 2,422.80** | **$ 3087.58** | **$ 8978.69** |  | **$3145+** |

# Instances

## EC2

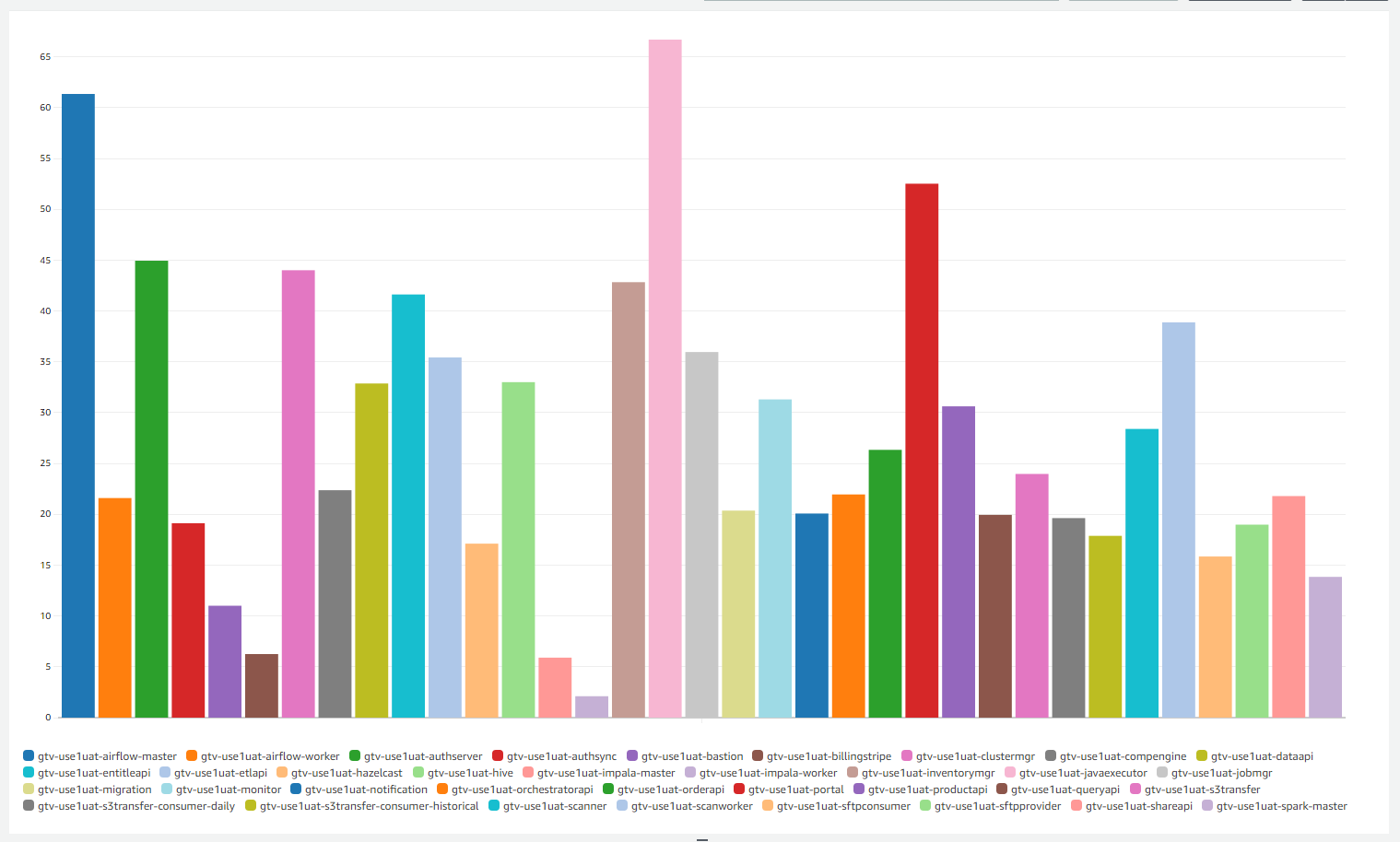
* N’inclus pas les instances RDS & MQ
* Ajustés avec les règles actuelles pour la mise hors-service durant la nuit et la fin-de-semaine

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Environnements | Count |  | $ / h | day | week | 4 weeks |
| **DEV** | **TOTAL** | **38** |  | **$2.1642** | **$34.63** | **$173.14** | **$692.54** |
| c5.large | **2** |  | $0.1860 | $2.98 | $14.88 | $59.52 |
| m5.large | **1** |  | $0.1070 | $1.71 | $8.56 | $34.24 |
| m6i.large | **1** |  | $0.0960 | $1.54 | $7.68 | $30.72 |
| r5.xlarge | **1** |  | $0.2760 | $4.42 | $22.08 | $88.32 |
| t3.large | **2** |  | $0.1856 | $2.97 | $14.85 | $59.39 |
| t3.medium | **23** |  | $0.9744 | $15.59 | $77.95 | $311.81 |
| t3.micro | **2** |  | $0.0232 | $0.37 | $1.86 | $7.42 |
| t3.small | **4** |  | $0.0928 | $1.48 | $7.42 | $29.70 |
| t3.xlarge | **1** |  | $0.1856 | $2.97 | $14.85 | $59.39 |
| t3a.medium | **1** |  | $0.0376 | $0.60 | $3.01 | $12.03 |
| **PROD** | **TOTAL** | **34** |  | **$1.9806** | **$47.53** | **$332.74** | **$1,330.96** |
| c5.large | 2 |  | $0.1860 | $4.46 | $31.25 | $124.99 |
| m5.large | 2 |  | $0.3210 | $7.70 | $53.93 | $215.71 |
| r5.xlarge | 1 |  | $0.2760 | $6.62 | $46.37 | $185.47 |
| t3.large | 2 |  | $0.1856 | $4.45 | $31.18 | $124.72 |
| t3.medium | 20 |  | $0.8352 | $20.04 | $140.31 | $561.25 |
| t3.micro | 2 |  | $0.0232 | $0.56 | $3.90 | $15.59 |
| t3.small | 4 |  | $0.1160 | $2.78 | $19.49 | $77.95 |
| t3a.medium | 1 |  | $0.0376 | $0.90 | $6.32 | $25.27 |
| **UAT** | **TOTAL** | **37** |  | **$2.1288** | **$34.06** | **$170.30** | **$681.22** |
| c5.large | 2 |  | $0.1860 | $2.98 | $14.88 | $59.52 |
| m5.large | 2 |  | $0.2140 | $3.42 | $17.12 | $68.48 |
| r5.xlarge | 1 |  | $0.2760 | $4.42 | $22.08 | $88.32 |
| t3.large | 2 |  | $0.1856 | $2.97 | $14.85 | $59.39 |
| t3.medium | 22 |  | $0.9280 | $14.85 | $74.24 | $296.96 |
| t3.micro | 2 |  | $0.0232 | $0.37 | $1.86 | $7.42 |
| t3.small | 4 |  | $0.0928 | $1.48 | $7.42 | $29.70 |
| t3.xlarge | 1 |  | $0.1856 | $2.97 | $14.85 | $59.39 |
| t3a.medium | 1 |  | $0.0376 | $0.60 | $3.01 | $12.03 |
|  | **Grand Total** | **109** |  | **$6.2736** | **$100.38** | **$501.89** | **$2,007.55** |

[Graphical user interface, website

Description automatically generated](https://aws.amazon.com/compute-optimizer/)

## Autoscalling



1 Avg. consumption in UAT

35%

1. Les tailles d’instances pourraient à peu-près toutes être réduites (de moitié ou +).
2. Les groupes n’ont pas de règles dynamiques.
3. Les groupes en dev/uat ferment la fin-de-semaine et la nuit. (Bon)

## MQ (AmazonMQ / ActiveMQ)

### Tailles d’instances

Les environnements de dev et uat pourraient utiliser les instances *mq.t3.micro* spécialement créées pour les environnements de dev. Les instances *mq.t3.micro* pourraient possiblement être amenées en prod pour certain tenants (à étudier).

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Env | Name | Instance type | Qty | Hourly | Daily | Weekly | Monthly |
| DEV | gtv-use1dev-amazonmq | mq.m5.large | 1 | $ 0.58 | $ 13.82 | $ 96.77 | $ 414.72 |
| UAT | gtv-use1uat-amazonmq | mq.m5.large | 1 | $ 0.58 | $ 13.82 | $ 96.77 | $ 414.72 |
| PROD | gtv-use1prod-amazonmq | mq.m5.large | 1 | $ 0.58 | $ 13.82 | $ 96.77 | $ 414.72 |

Économie par “mq.*t3.micro”* : 289$/instance/mois (90%).

*Les instances actuelles sont littéralement inutilisées.*

## RDS

1. Les tailles d’instances pour « AirFlow » pourraient être largement réduites.
2. Les pools de connections seraient à vérifier pour les applications GTV (base 200+ connections)

### Tailles d’instances

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Env | Name | Instance size | Qty | Hourly | Daily | Weekly | Monthly |
| DEV | gtv-use1dev-airflow-db | db.r5.large | 1 | $ 0.2900 | $ 6.96 | $ 48.72 | $ 208.80 |
| DEV | gtv-use1dev-database-57 | db.t3.small | 1 | $ 0.0410 | $ 0.98 | $ 6.89 | $ 29.52 |
| DEV | gtv-use1dev-database-57 | db.t3.small | 1 | $ 0.0410 | $ 0.98 | $ 6.89 | $ 29.52 |
| UAT | gtv-use1uat-airflow-db | db.r5.large | 1 | $ 0.2900 | $ 6.96 | $ 48.72 | $ 208.80 |
| UAT | gtv-use1uat-database-57 | db.t3.small | 1 | $ 0.0410 | $ 0.98 | $ 6.89 | $ 29.52 |
| PROD | gtv-use1prod-airflow-db | db.r5.large | 1 | $ 0.2900 | $ 6.96 | $ 48.72 | $ 208.80 |
| PROD | gtv-use1prod-database-57 | db.t3.small | 1 | $ 0.0410 | $ 0.98 | $ 6.89 | $ 29.52 |

# Cloudwatch

## Agent CloudWatch

### Métriques personnalisées

1. Approx. 50% des coûts associés proviennent des coûts de métriques personnalisées (0.30$ / métrique / instance) générées par l’agent CloudWatch.
2. De ces 50%, **4 métriques représentent près de 95% à elles seule.**
   * *disk\_free (42%)*
   * *disk\_used\_percent (42%)*
   * *mem\_used\_percent (5.5%)*
   * *swap\_used\_percent (5%)*

2 Métriques produites par CWAgent en dev

### Métriques dupliquées/calculables

Certaines métriques sont supposées être calculées dans les queries CloudWatch (min, max, avg, etc…). Elles ont effectivement la même valeur.

Example

Connections / connections\_min / connections\_max

Note

L’utilisation de « [namespaces](https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/cloudwatch_concepts.html" \l "Namespace) » clarifierait énormément la navigation dans CloudWatch.

Graphical user interface, text, application, email

Description automatically generated

### Sélection des métriques

Beaucoup de métriques ne sont pas nécessaires ou sont le résultat d’un sélecteur étoile \*.

### Agent configuration

{

"agent": {

"debug": false,

"logfile": "/var/log/cloudwatch.log",

"metrics\_collection\_interval": 60

},

"logs": {

"log\_stream\_name": "{local\_hostname}",

"logs\_collected": {

"files": {

"collect\_list": [

{

"file\_path": "/var/log/cloud-init.log",

"log\_group\_name": "var-log-cloudinit",

"timestamp\_format": "%Y-%m-%d %H:%M:%S,%f"

},

**. . .**

]

}

}

},

"metrics": {

"aggregation\_dimensions": [

["AutoScalingGroupName"],

["InstanceId","InstanceType"],

[]

],

"append\_dimensions": {

"AutoScalingGroupName": "${aws:AutoScalingGroupName}",

"InstanceId": "${aws:InstanceId}"

},

"metrics\_collected": {

"disk": {

"measurement": [

"disk\_free",

"disk\_used\_percent"

]

},

"mem": {

"measurement": [

"mem\_used\_percent"

]

},

"statsd": {},

"swap": {

"measurement": [

"swap\_used\_percent"

]

}

},

"namespace": "CWAgent"

}

}

Il est possible d’utiliser Cloud-init et les metadatas AWS pour utiliser l’index d’instance plutôt que l’ID. Ce n’est pas parfait, mais réduit dramatiquement les couts

Eg : ‘${scalling-group}-[0,1,2,3,…]’

L’utilité de « disk\_free » serait à considérer étant donné les coûts

Les disques devraient être spécifiées

Eg. "resources":[ "/", "/tmp" ],

Les fichiers pourraient être séparés et chacun donnés un interval et un namespace complet

Eg : “CWAgent/Disks”

# Autres Recommendations

## AWS Compute Optimizer

Graphical user interface, application

Description automatically generated[Voir sur AWS](https://aws.amazon.com/compute-optimizer/)

*AWS Compute optimizer* générera des rapports pour optimizer les groupes d’autoscalling et les tailles d’instances.

Il est recommandé de laisser rouler 14 jours avant de faire les modifications

## AWS Systems Manager Compliance & Patches

[Voir sur AWS](https://aws.amazon.com/systems-manager/features/" \l "Compliance)

Plusieurs agents et autres programmes sont passé date. AWS system manager permet d’administrer et d’automatiser les updates en plus d’offrir la possibilité d’administrer les configurations à distance sans causer de ‘downtime’.

Graphical user interface, text, application

Description automatically generated