Beware of peripheral Lambda costs

$$$$$

Serverless: 15% slower and 8x more expensive

Posted: 2019-09-18 Last updated: 2019-09-26

Recently I wanted to try changing the API we have at CardGames.io and try using the Serverless framework. Serverless has been a hot topic in the tech world for the last few years and I ~~was procrastinating~~ wanted to keep my tech skills up to date by trying something new, so I decided to spend a few hours learning about serverless and see if hosting our API that way made sense.

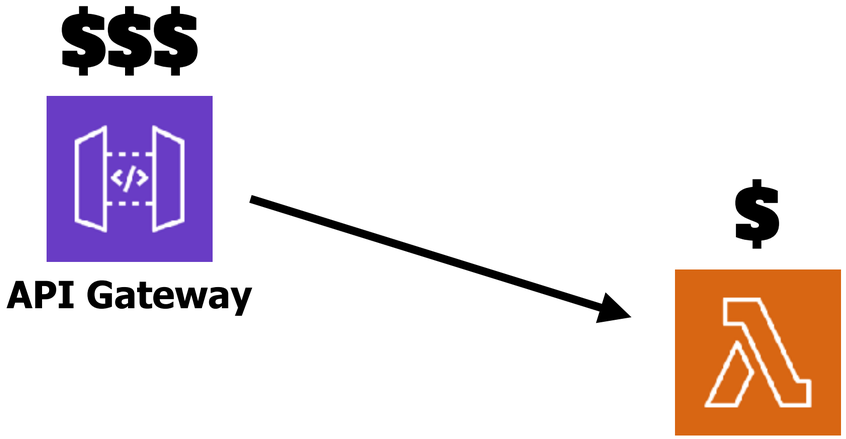
Current setup

CardGames.io is hosted on AWS. We use S3 to store html pages, css, javascript and images. We have an API written in C# which is hosted on Elastic Beanstalk, using Linux servers running .NET Core with Docker. Finally we have a CloudFront CDN running in front of both the static files on S3 and the API. Below is our EC2 bill for August 2019. We have a few other instances, but for the API we use the m1.small instances (yes we should probably be using t2.small) and classic load balancing. When you sum up the parts in red it's **164.21$** for the month, not bad. I even included the whole EBS part in that, mostly because I'm not sure which part of it belongs to other stuff we run, we have a couple of other EC2 things running as well.

https://einaregilsson.com/serverless-15-percent-slower-and-eight-times-more-expensive/

UPDATE 26.9.2019: This post got over a 100.000 views in 3 days and was on the top of both Hacker News and reddit.com/r/programming . I've definitely learnt a few things. I should have used an Application Load Balancer instead of API Gateway in front of Lambda. I should upgrade my ancient m1.small instances to a newer instance type. And I was able to get my Beanstalk deploy time down to a reasonable 30~40 seconds by using the runtime .NET Core docker image instead of the SDK one. Serverless still isn't the right fit for my use case, but it was definitely an interesting experiment!

UPDATE 26.9.2019: This post got over a 100.000 views in 3 days and was on the top of both Hacker News and reddit.com/r/programming . I've definitely learnt a few things. I should have used an Application Load Balancer instead of API Gateway in front of Lambda. I should upgrade my ancient m1.small instances to a newer instance type. And I was able to get my Beanstalk deploy time down to a reasonable 30~40 seconds by using the runtime .NET Core docker image instead of the SDK one. Serverless still isn't the right fit for my use case, but it was definitely an interesting experiment!


$$$
API Gateway
$

API Calls

Number of Requests (per month)	Price (per million)
First 333 million	$3.50
Next 667 million	$2.80
Next 19 billion	$2.38
Over 20 billion	$1.51

Caching

For better performance and faster API execution, you can optionally provision a dedicated cache for each stage of your APIs. After you specify the size of the cache you require, you will be charged the following hourly rates for each stage's cache, without any long-term commitments.

Cache Memory Size (GB)	Price per Hour
0.5	$0.02
1.6	$0.038
6.1	$0.20
13.5	$0.25
28.4	$0.50
58.2	$1.00
118.0	$1.90
237.0	$3.80

https://aws.amazon.com/api-gateway/pricing/

All you need to know about caching for serverless applications

API Gateway, AWS, CloudFront, DynamoDB, Lambda, Programming, Serverless / October 3, 2019

*Check out my new course **Learn you some Lambda best practice for great good!** and learn the best practices for performance, cost, security, resilience, observability and scalability.*

Last week, someone asked me at the AWS User Group in The Hague *"Is caching still relevant for serverless applications?"*

The assumption there is that Lambda auto-scales by traffic, so do we still need to worry about caching? And if so, where and how do we implement caching?

So let's break it down.

Caching is still VERY relevant

Yes, Lambda auto-scales by traffic. But it has limits.

There's the soft limit of 1000 concurrent executions in most regions. Which you can raise via a support ticket. But there's also a hard limit on how quickly you can increase the concurrent executions after the initial 1000. In most regions, that limit is 500 per minute.

Which means, it'll take you 18 minutes to reach a peak throughput of 10k concurrent executions. This is not a problem if your traffic is either very stable or follows the bell curve so there are no sudden spikes.

https://theburningmonk.com/2019/10/all-you-need-to-know-about-caching-for-serverless-applications/

HTTP APIs (Preview)

Pay only for the API calls you make. For HTTP APIs, the API Gateway free tier includes one million API calls per month for up to 12 months.

Region: US East (N. Virginia)

API Calls

Number of Requests (per month)	Price (per million)
First 300 million	$1.00
300+ million	$0.90

*HTTP APIs are metered in 512 KB increments.

70% cheaper!

https://aws.amazon.com/api-gateway/pricing/

Choosing Between HTTP APIs and REST APIs

PDF | Kindle | RSS

HTTP APIs are in beta for Amazon API Gateway and are subject to change.

HTTP APIs are designed for low-latency, cost-effective AWS Lambda proxy and HTTP proxy APIs. HTTP APIs support OIDC and OAuth 2.0 authorization, and come with built-in support for CORS and automatic deployments. Previous-generation REST APIs currently offer more features, and full control over API requests and responses.

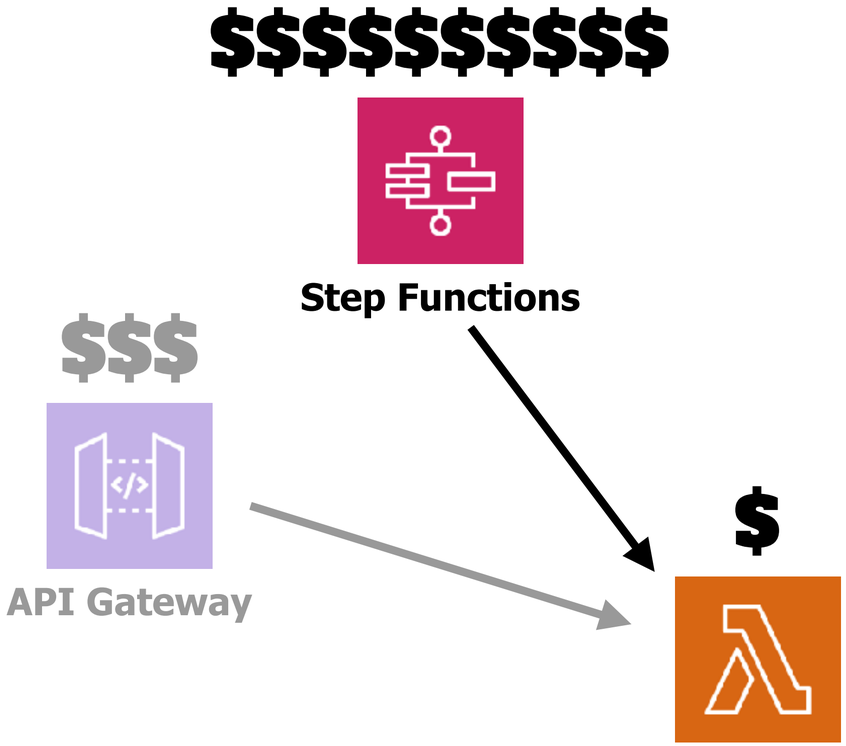
HTTP APIs are available in the following Regions:

- US East (Ohio)
- US East (N. Virginia)
- US West (N. California)
- US West (Oregon)
- Asia Pacific (Sydney)
- Asia Pacific (Tokyo)
- Europe (Frankfurt)
- Europe (Ireland)

The following tables summarize core features that are available in HTTP APIs and REST APIs.

Authorizers	HTTP API	REST API
AWS Lambda		✓
IAM		✓
Amazon Cognito	✓ *	✓
Native OpenID Connect / OAuth 2.0	✓	

https://docs.aws.amazon.com/apigateway/latest/developerguide/http-api-vs-rest.html


$$$$$$$$$$
Step Functions
$$$
API Gateway
$

AWS Step Functions Standard Workflow State transitions pricing

Region: US East (N. Virginia)

$25 per million!!!

Price per 1,000 state transitions
$0.025

https://aws.amazon.com/step-functions/pricing/

AWS Step Functions Express Workflows pricing details

With Step Functions Express Workflows, you pay only for what you use. You are charged based on the number of requests for your workflow and the duration, the time it takes for your workflow to execute.

Step Functions Express Workflows counts a request each time it starts executing a workflow, including tests from the console. You are charged for the total number of requests across all your workflows.

Duration is calculated from the time your workflow begins executing until it completes or otherwise terminates, rounded up to the nearest 100ms, and the amount of memory used in the execution of your workflow, billed in 64-MB chunks.

Memory consumption is based on the size of a workflow definition, the use of map or parallel states, and the execution (payload) data size. Pricing examples 3 and 4 show examples of estimating memory utilization.

Requests

$1.00 per 1M requests

$0.000001 per request

Duration

$0.00001667 per GB-Second ($0.0600 per GB-hour) for the first 1,000 hours GH-hours

$0.00000833 per GB-Second ($0.0300 per GB hour) for the next 4,000 hours GH-hours

$0.00000456 per GB-Second ($0.01642 per GB-hour) beyond that

https://aws.amazon.com/step-functions/pricing/

AWS Step Functions Express Workflows pricing details

With Step Functions Express Workflows, you pay only for what you use. You are charged based on the number of requests for your workflow and the duration, the time it takes for your workflow to execute.

Step Functions Express Workflows counts a ... ss all your workflows.

Duration is calculated from the time your w... in the execution of your workflow, billed in 64...

Memory consumption is based on the size o... of estimating memory utilization.

Requests

$1.00 per 1M requests

$0.000001 per request

Duration

$0.00001667 per GB-Second ($0.0600 per GB-hour) for the first 1,000 hours GH-hours

$0.00000833 per GB-Second ($0.0300 per GB hour) for the next 4,000 hours GH-hours

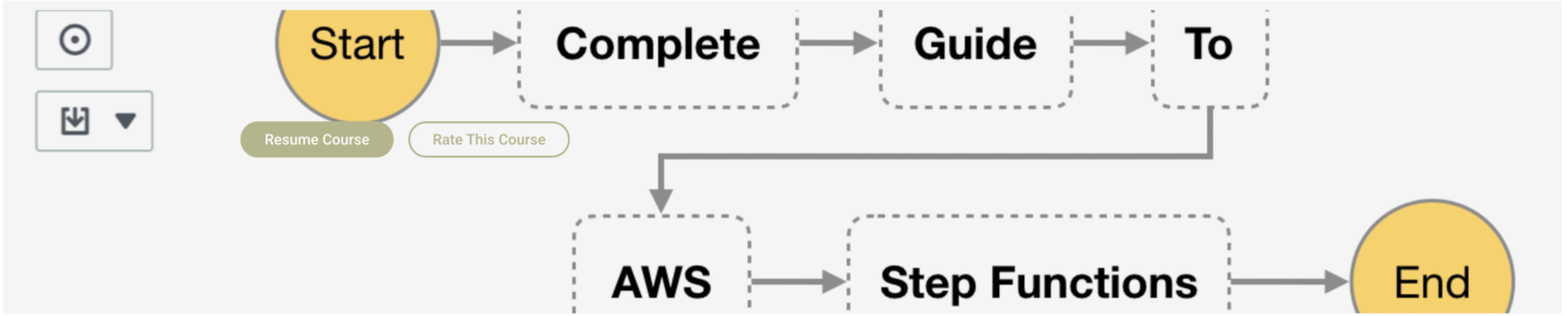
$0.00000456 per GB-Second ($0.01642 per GB-hour) beyond that

AWS Lambda Pricing

Region: US East (N. Virginia)

	Price
Requests	$0.20 per 1M requests
Duration	$0.000016667 for every GB-second

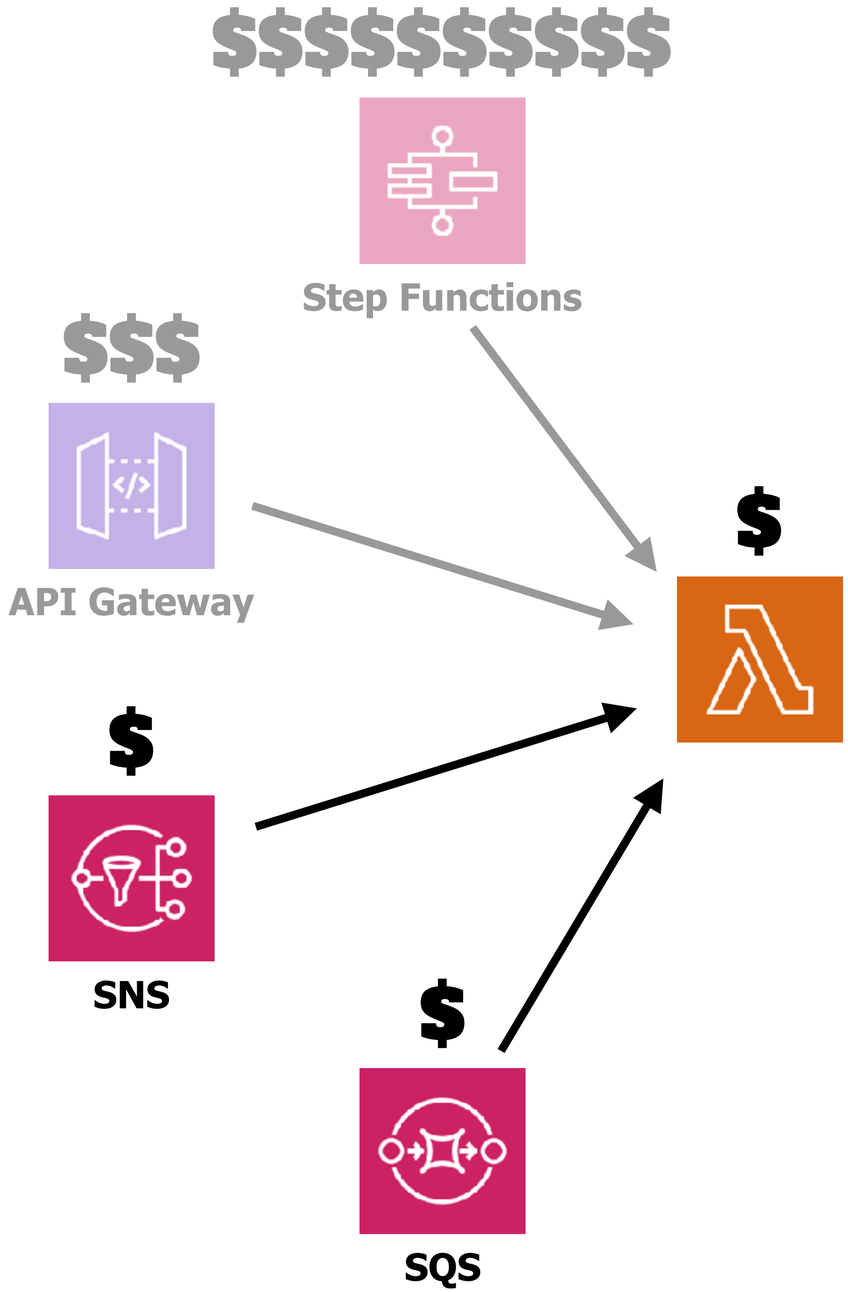
https://aws.amazon.com/step-functions/pricing/

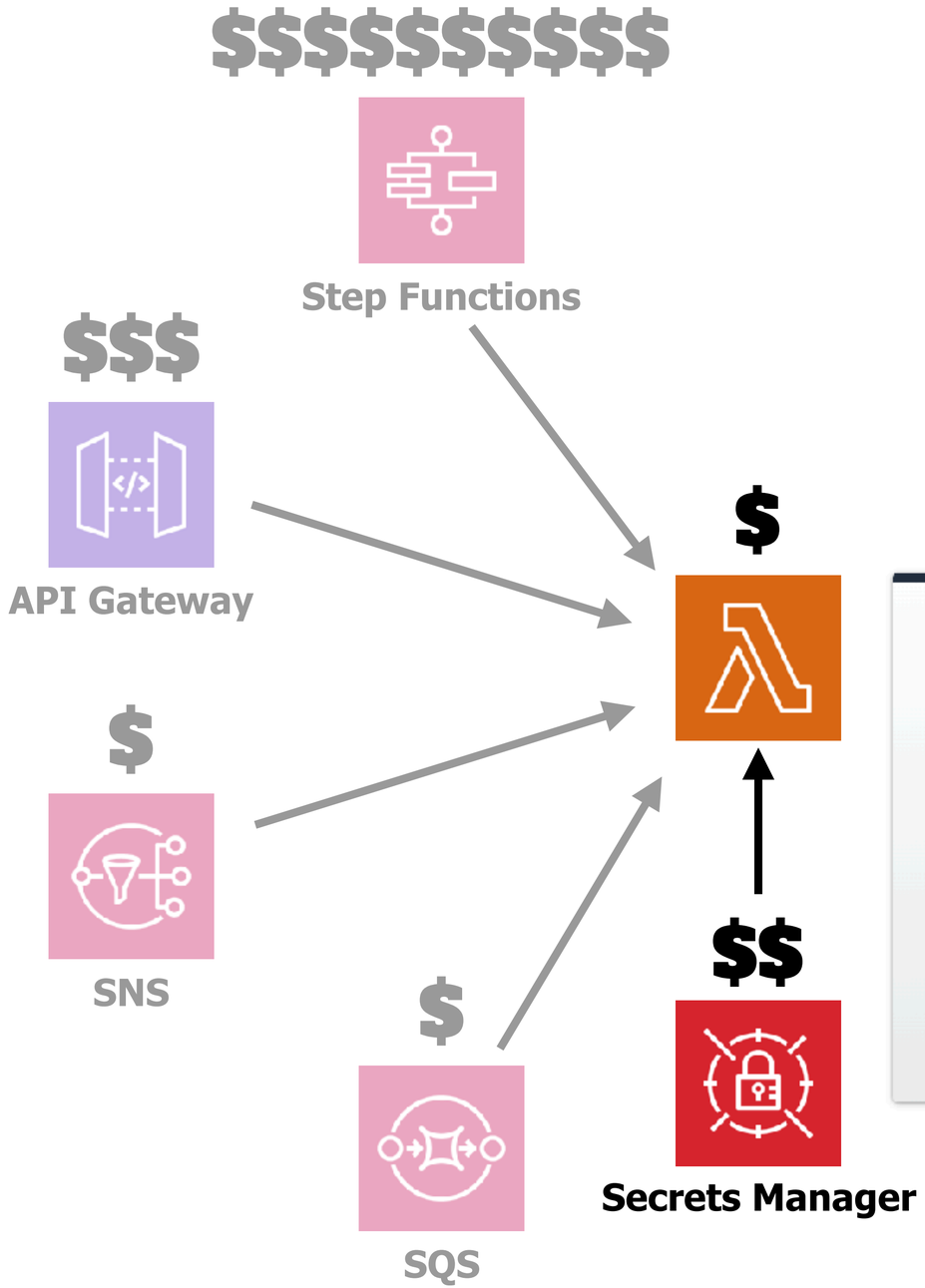


What will I learn?

- What is the Step Functions service and when to use it.
- How to chain Lambda functions together as a workflow.
- How to trigger Step Functions using API Gateway and CloudWatch Events.
- Monitoring and debugging Step Functions.
- How to wait for human input using activities.
- Callback patterns.
- Blue-green deployments..
- Best practices

https://theburningmonk.thinkific.com/courses/complete-guide-to-aws-step-functions


$$$$$$$$$$
Step Functions
$$$
API Gateway
$
$
SNS
$
SQS



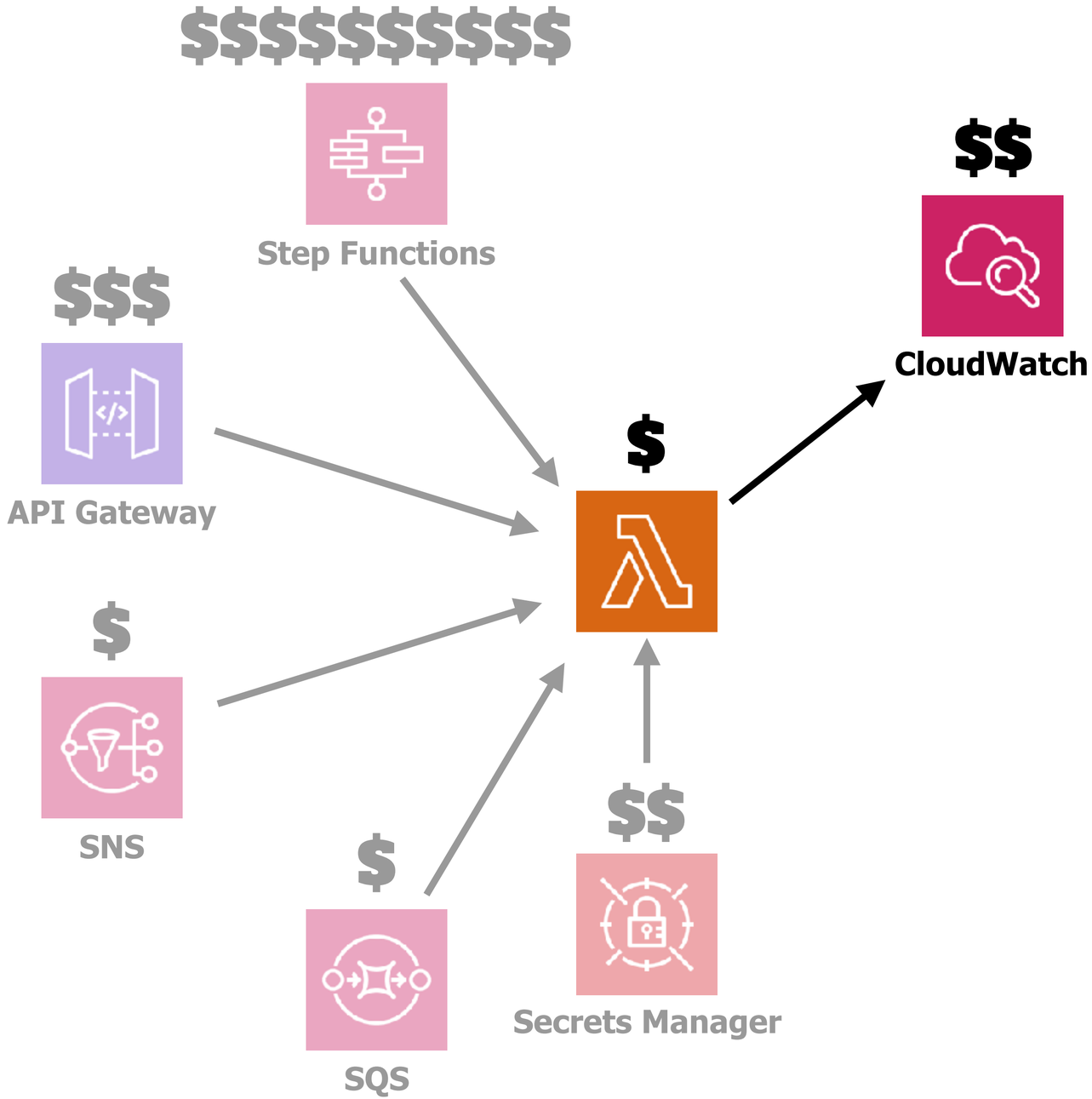
Pricing

PER SECRET PER MONTH

$0.40 per secret per month. For secrets that are stored for less than a month, the price is prorated (based on the number of hours.)

PER 10,000 API CALLS

$0.05 per 10,000 API calls.


$$$$$$$$$$
Step Functions
$$$
API Gateway
$
SNS
$
SQS
$$
Secrets Manager
$
$$
CloudWatch

Logs

All log types. There is no Data Transfer IN charge for any of CloudWatch.

Data Transfer OUT from CloudWatch Logs is priced equivalent to the "Data Transfer OUT from Amazon EC2 To" and "Data Transfer OUT from Amazon EC2 to Internet" tables on the EC2 Pricing Page.

CloudWatch Container Insights ingests performance events as CloudWatch Logs that automatically create CloudWatch metrics. These performance events are analyzed using CloudWatch Logs Insights queries and are automatically executed as part of some Container Insights automated dashboards (e.g., task/pod, service, node, namespace). Please see the Container Insights documentation for details on the type of performance events ingested and queried.

Collect (Data Ingestion)	$0.50 per GB
Store (Archival)	$0.03 per GB
Analyze (Logs Insights queries)	$0.005 per GB of data scanned

https://aws.amazon.com/cloudwatch/pricing/

Metrics

All custom metrics charges are prorated by the hour and metered only when you send metrics to CloudWatch.

Amazon EC2 Detailed Monitoring pricing is based on the number of custom metrics, with no API charge for sending metrics. The number of metrics sent by an EC2 instance as part of EC2 Detailed Monitoring is dependent on the instance type --- for details, see the Instance Metrics documentation. Typically, EC2 Detailed Monitoring is charged at $2.10 per instance per month (assumes 7 metrics per instance) and goes down to $0.14 per instance at the lowest priced tier. As with all custom metrics, EC2 Detailed Monitoring is prorated by the hour and metered only when the instance sends metrics to CloudWatch.

Tiers	Cost (metric/month)
First 10,000 metrics	$0.30
Next 240,000 metrics	$0.10
Next 750,000 metrics	$0.05
Over 1,000,000 metrics	$0.02

https://aws.amazon.com/cloudwatch/pricing/

Dimension Combinations

CloudWatch treats each unique combination of dimensions as a separate metric, even if the metrics have the same metric name. You can only retrieve statistics using combinations of dimensions that you specifically published. When you retrieve statistics, specify the same values for the namespace, metric name, and dimension parameters that were used when the metrics were created. You can also specify the start and end times for CloudWatch to use for aggregation.

For example, suppose that you publish four distinct metrics named ServerStats in the DataCenterMetric namespace with the following properties:

```
Dimensions: Server=Prod, Domain=Frankfurt, Unit: Count, Timestamp: 2016-10-31T12:30:00Z, Value: 10
Dimensions: Server=Beta, Domain=Frankfurt, Unit: Count, Timestamp: 2016-10-31T12:31:00Z, Value: 115
Dimensions: Server=Prod, Domain=Rio,       Unit: Count, Timestamp: 2016-10-31T12:32:00Z, Value: 95
Dimensions: Server=Beta, Domain=Rio,       Unit: Count, Timestamp: 2016-10-31T12:33:00Z, Value: 97
```

If you publish only those four metrics, you can retrieve statistics for these combinations of dimensions:

- `Server=Prod,Domain=Frankfurt`
- `Server=Prod,Domain=Rio`
- `Server=Beta,Domain=Frankfurt`
- `Server=Beta,Domain=Rio`

https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/cloudwatch_concepts.html

Anomaly Detection is priced at $0.30 per alarm per month for standard resolution alarms and $0.90 per alarm per month for high resolution alarms. When you create an alarm using Anomaly Detection, each alarm will be priced as three standard or high resolution alarm metrics -- one for the evaluated metric, and two for the upper and lower bound of expected behavior.

Region: US East (N. Virginia)

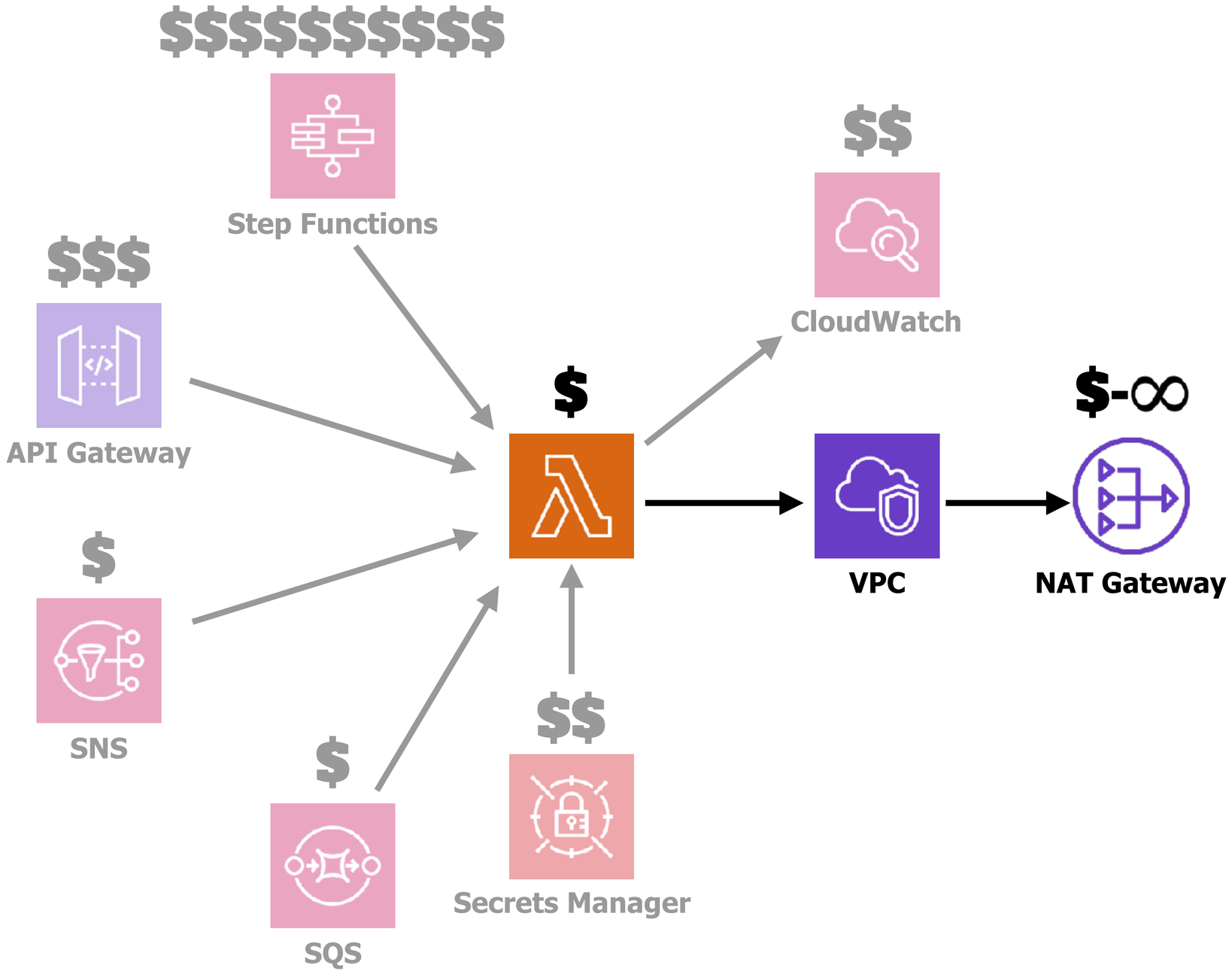
Alarms

Standard Resolution (60 sec)	$0.10 per alarm metric
High Resolution (10 sec)	$0.30 per alarm metric

Dashboards

Dashboard	$3.00 per dashboard per month

https://aws.amazon.com/cloudwatch/pricing/


$$$$$$$$$$
Step Functions
$$$
API Gateway
$
SNS
$
SQS
$$
Secrets Manager
$$
CloudWatch
$
$-∞
VPC
NAT Gateway

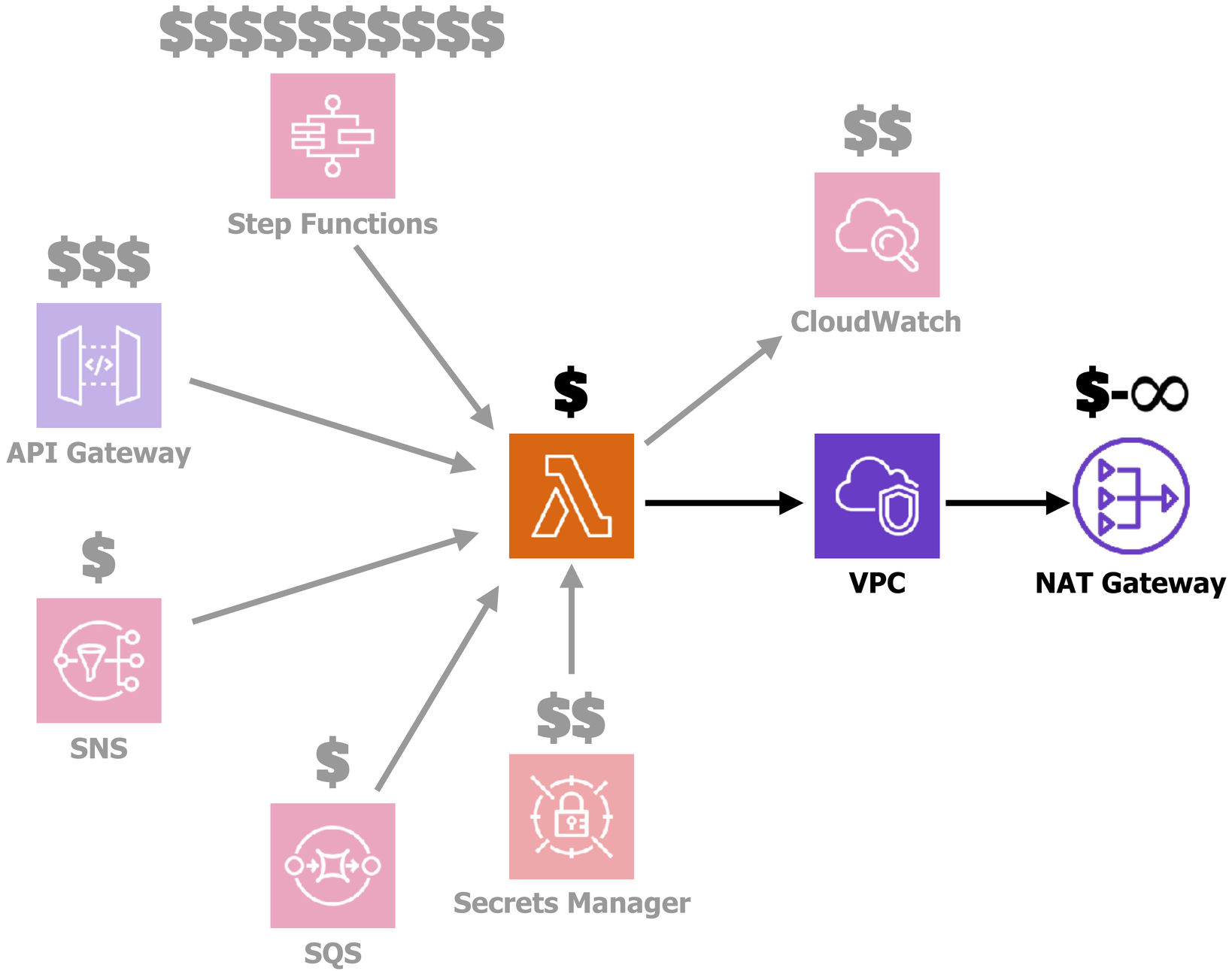
NAT Gateway Pricing

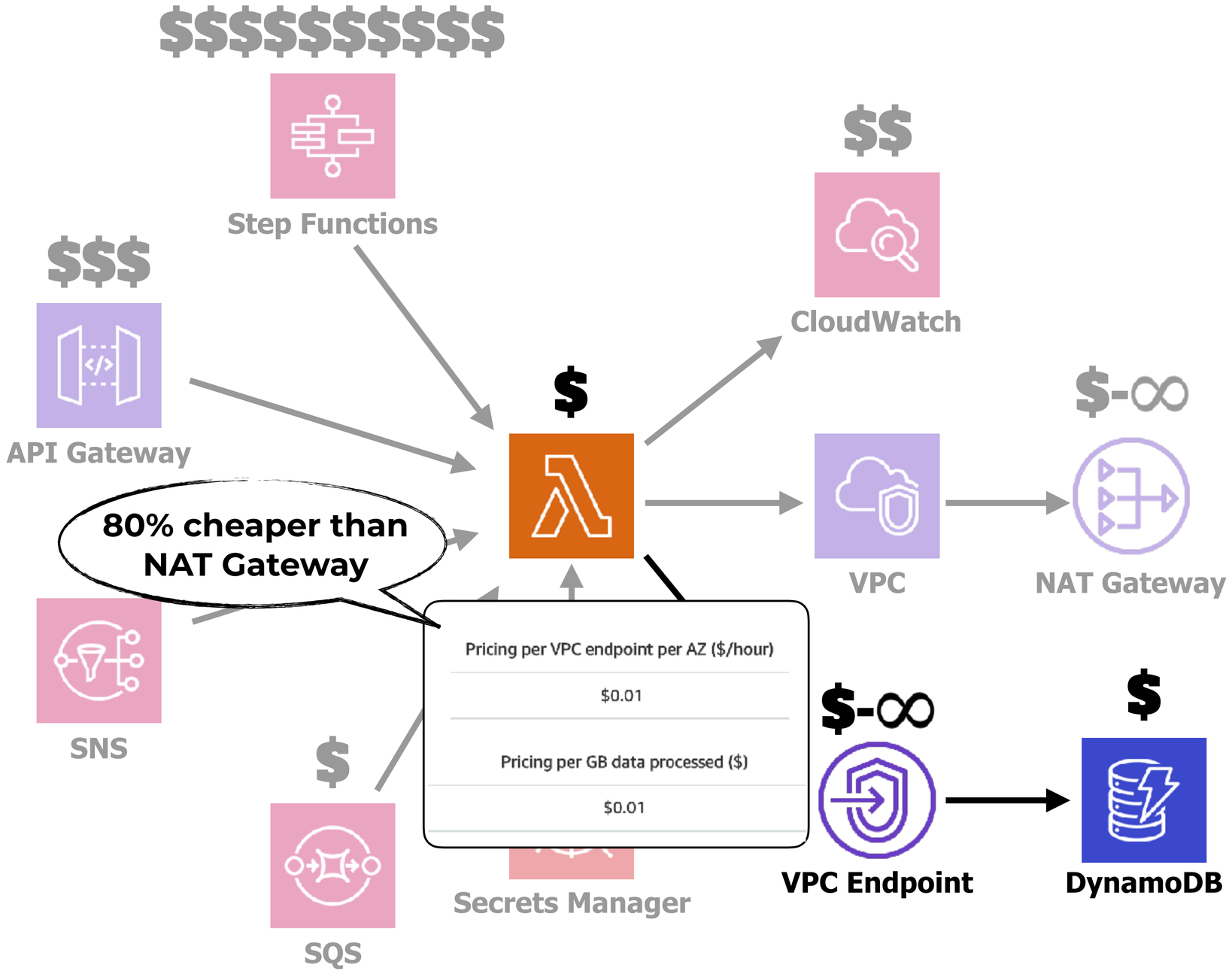
If you choose to create a NAT gateway in your VPC, you are charged for each "NAT Gateway-hour" that your NAT gateway is provisioned and available. Data processing charges apply for each Gigabyte processed through the NAT gateway regardless of the traffic's source or destination. Each partial NAT Gateway-hour consumed is billed as a full hour. You also incur standard AWS data transfer charges for all data transferred via the NAT gateway. If you no longer wish to be charged for a NAT gateway, simply delete your NAT gateway using the AWS Management Console, commandline interface, or API.

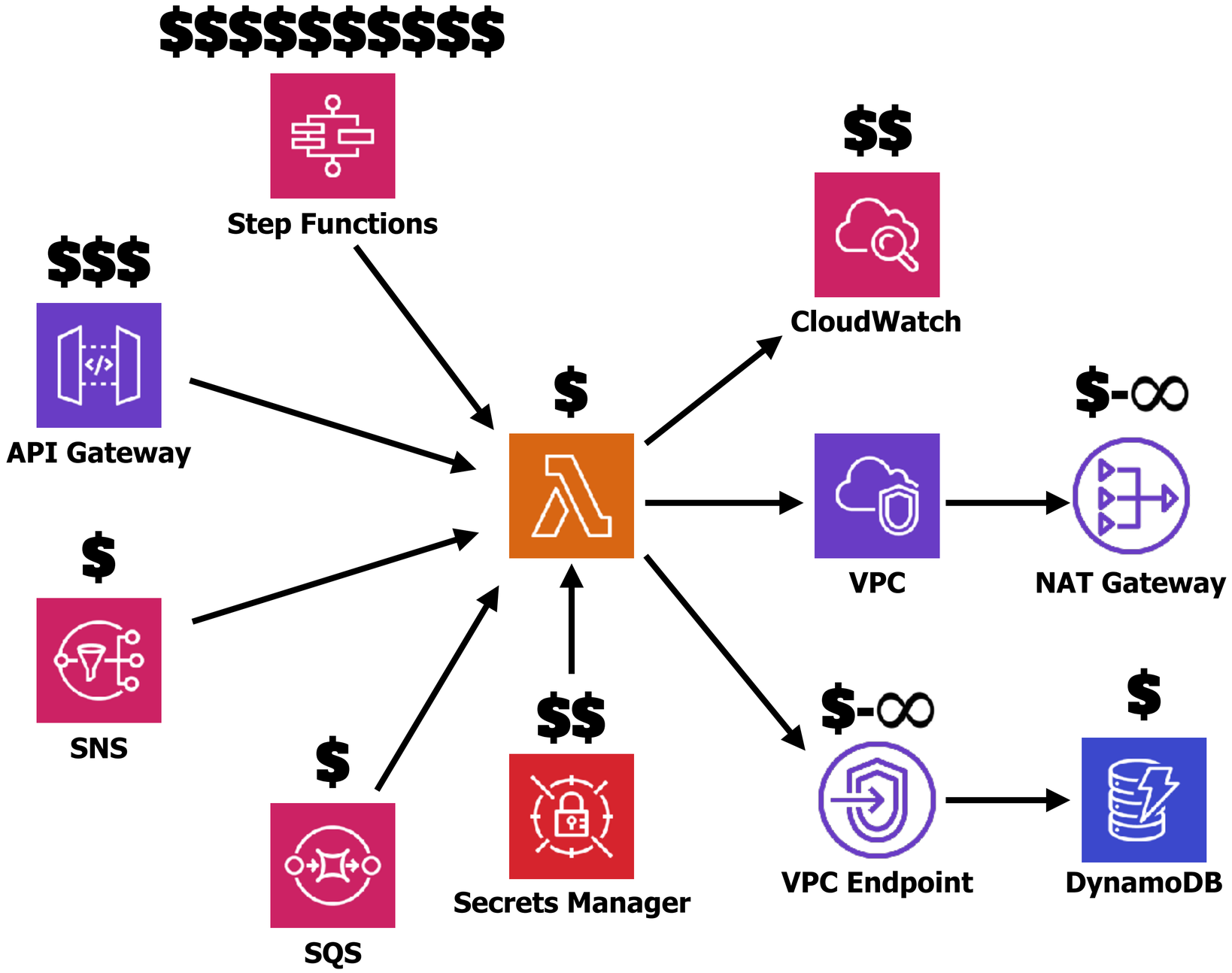
Region: US East (N. Virginia)

Price per NAT gateway ($/hour)	Price per GB data processed ($)
$0.045	$0.045

https://aws.amazon.com/vpc/pricing/


$$$$$$$$$$
Step Functions
$$$
API Gateway
$
SNS
$
SQS
$$
Secrets Manager
$$
CloudWatch
$
$-∞
VPC
NAT Gateway

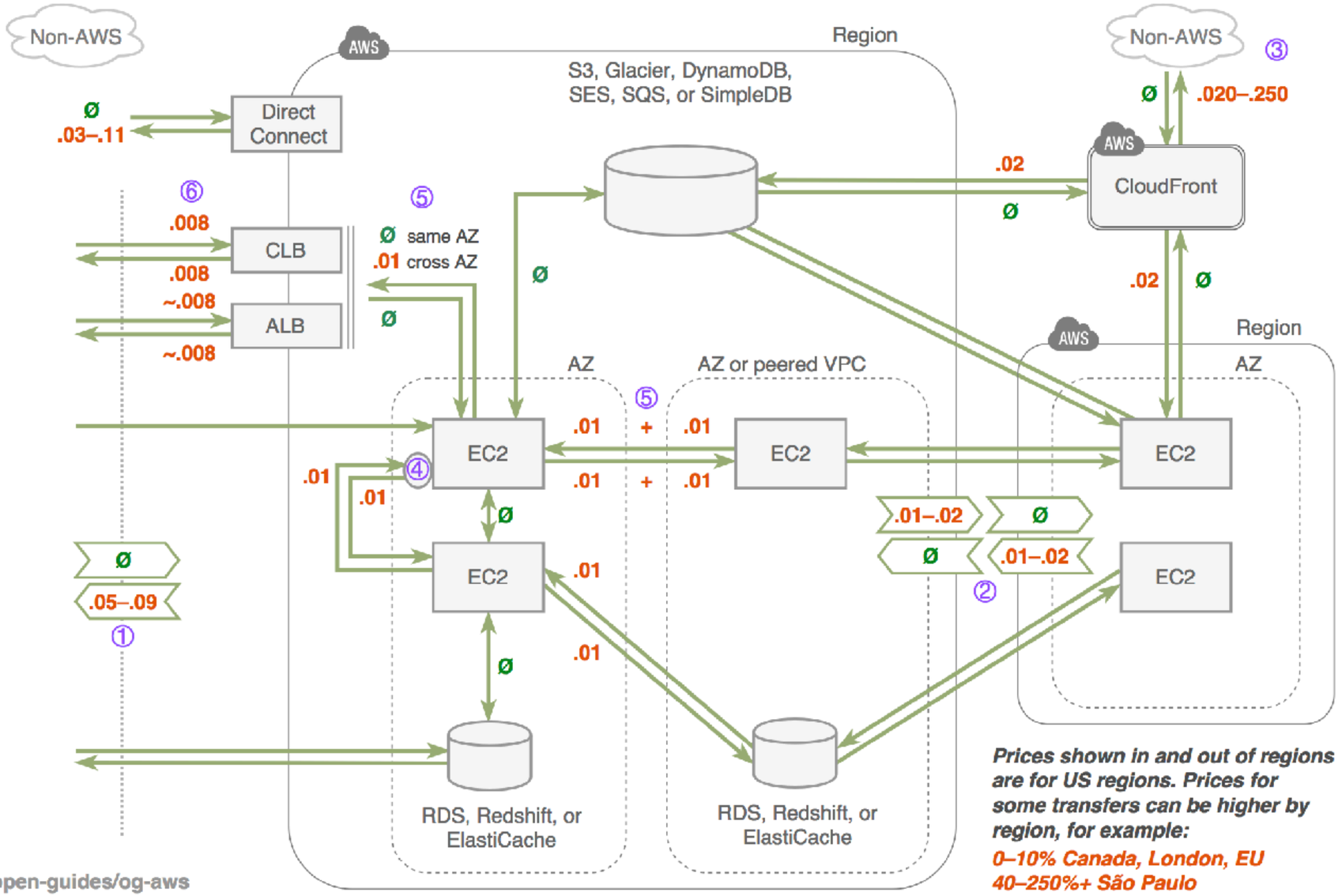

$$$$$$$$$$
Step Functions
$$
CloudWatch
$$$
API Gateway
$
$-∞
80% cheaper than NAT Gateway
VPC
NAT Gateway
SNS
Pricing per VPC endpoint per AZ ($/hour)
$0.01
Pricing per GB data processed ($)
$0.01
$-∞
$
$
SQS
Secrets Manager
VPC Endpoint
DynamoDB


$$$$$$$$$$
Step Functions
$$$
API Gateway
$
SNS
$
SQS
$$
Secrets Manager
$
$$
CloudWatch
$-∞
VPC
NAT Gateway
$-∞
VPC Endpoint
$
DynamoDB

AWS DATA TRANSFER COSTS

Numbers are data transfer in $/GB. Transaction and hourly prices are not shown. See notes.

- Ø Free. Inbound traffic is mostly free—you pay on the way out. Some but not all internal traffic is free.
- ① Direct outbound data starts at $.09/GB for <10TB, and discounts with volume. First 1GB free.
- ② Region-to-region traffic is $.02/GB when it exits a region for indicated services except between us-east-1 and us-east-2, where it's $.01/GB.
- ③ Outbound CloudFront prices are highly variable by geography and regional edge cache and start at $.085/GB in US/Canada.
- ④ Internal traffic via public or elastic IPs incurs additional fees in both directions.
- ⑤ Cross-AZ EC2 traffic within a region costs as much as region-to-region! ELB-EC2 traffic is free except outbound crossing AZs.
- ⑥ Elastic Load Balancing: Classic LB is priced per GB. Application LB costs are in LCUs, not $/GB.



Credits and latest version: **github.com/open-guides/og-aws**
Last update: 2017-08-14

https://raw.githubusercontent.com/open-guides/og-aws/master/figures/aws-data-transfer-costs.png