2024

IoT on AWS

A guide to services and costs

compiuta

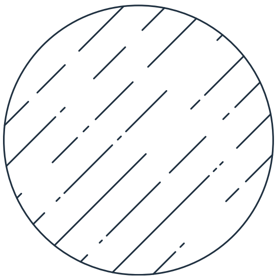
Introduction

This whitepaper describes how to implement a basic - yet complete - IoT infrastructure using AWS services, while also offering a detailed breakdown of all the costs involved.

After reading this document, you'll understand:

- the key components of a modern IoT infrastructure
- how to do some back-of-the-envelope math to estimate costs
- how much each service contributes to the total operational cost and what optimizations are possible

Overview



AWS offers more than 200 services. While this is great, it can also be daunting: how are you supposed to know them all, understand how to piece them together and create the IoT platform your company needs?

This is why we decided to **share our knowledge** and created this guide, with a particular focus on costs. Sometimes even giving a ballpark guess is hard and we're trying to tackle this.

This has no ambition to be an exhaustive resource: treat it more like a primer on how to use AWS services for IoT applications.

How to read this document (and a few caveats)

As always in software, there are many ways to achieve the same objective. We tried to stay clear from "it depends" statements and give you the **simplest path** to follow in order to create an IoT infrastructure.

This whitepaper contains three different sections: **services**, **scenarios** and **cost analysis** - plus a few bonuses. Each section is self-contained, so feel free to skim in whatever order you like.

Throughout this document, you'll find some sections highlighted with the 💡 symbol: these are practical tips based on experience gained over the years. Do yourself a favour: follow them and avoid a few headaches!

Enough with the talking, let's dive into it!

Services

AWS offers a collection of managed services that can be used to create an IoT infrastructure: the main advantage of using these is you **don't need to worry about scalability**.

Here's the minimum set of AWS services you'll need to create a basic IoT solution, coupled with a short summary of the features it provides.

Name	Used to
IoT Core	Maintain device registries, message broker, device connection status, rules engine, device certificates (Amazon Sidewalk)
IoT Remote Device Management	Monitor device fleet, tunneling to a device, bulk device registration
AWS Lambda	Apply rules to incoming messages (message mapper)
DynamoDB	Data storage
S3 - Glacier Deep Archive	Backup archival, disaster recovery archival
Amazon CloudWatch	Cloud monitoring
AWS Key Management Service	Create and control encyption keys (HTTPs, encrypting data in DynamoDB or S3, ...)
Amazon SNS	Simple Notification Service (e-mail, SMS and push notifications)
Amazon API Gateway	Expose data through RESTful APIs and Websockets

As you can see, there are **three** categories of services - respectively dedicated to device **connection**, data **storage** and **auxiliary** tasks.

Data storage deserves a special mention. In IoT applications, this is typically the most critical aspect, both in terms of costs and performances. The following section provides a few pointers on how to achieve an optimal data-storage implementation.

💡Optimizing data storage

IoT scenarios typically involve large amounts of data in timeseries format.

The key insight is this: **not all data is equal**. The most recent values need to be accessed frequently, whereas older data points are rarely consulted.

This maps well to a four tier structure: **hot**, **warm**, **cold** and **archival** storage. As the temperature goes down, performance and cost-per-byte decrease.

> An optimal storage policy involves writing data to a hot storage, then progressively transferring it towards lower temperatures as time goes by.

While the first three tiers are similar to each other, archival deserves a special mention. Think of it as memory that is **cheap to write** and maintain at rest, yet extremely **expensive to read**: it's typically used for storing backups in a cost-efficient fashion.

Let's make this practical with an example involving AWS services. One could for example provision:

- 1 month of *DynamoDB standard* ("hot") storage
- 1 month of *DynamoDB standard-infrequent access* ("warm") storage. Please note that standard-infrequent access storage has the same performances as standard storage - it just costs more for accessing data and less to keep it at rest
- 1 year of *S3 - Glacier Deep Archive* ("archival") storage
- DynamoDB on-demand backup (for hot storage)
- DynamoDB export storage to S3 (for data transfer to Glacier)

Such a configuration would give applications access to the last 2 months of collected data, plus an entire year of data retention. This is explored in more detail in the "Scenarios" section.

A final note on DynamoDB: in order to optimize costs, you should also understand what kind of **capacity** is needed for your use case.

Capacity is either *provisioned* or *on-demand*: think of it as paying a fixed amount versus paying per-request. In our experience, **most IoT use cases benefit from a provisioned pricing** model, since the majority of traffic can be forecasted easily - it scales with the number of devices in the field. If you're interested, you can read more here: https://docs.aws.amazon.com/wellarchitected/latest/serverless-applications-lens/capacity.html.

Scenarios

Overview

Creating an IoT infrastructure on AWS and properly evaluating its cost means **understanding how much data** is involved. You should take into account three factors:

- how many devices are connected
- how much data each one of them collects
- how long you want to store data for

If you need to get a proper quote for your use case, your best bet is heading over to the AWS Calculator.

As a rough estimate, keep in mind that costs for connecting more devices or gathering more data scale linearly. This is why, instead of comparing different tiers of devices, the **simulations** contained in this document have been **standardized** to **20.000 devices** and the effect of adding different functionalities has been highlighted.

Configurations

We're going to analyze costs in **six scenarios**. Each one has a unique combination of connectivity, storage and auxiliary services.

	Devices	Monthly transfer rate [messages/device]	Payload [kB]	Lambda processing
Scenario 1	20.000	43.800 (1 msg/min)	**5 kB**	No
Scenario 2	20.000	43.800 (1 msg/min)	**10 kB**	No
Scenario 3	20.000	43.800 (1 msg/min)	**15 kB**	No
Scenario 4	20.000	43.800 (1 msg/min)	5 kB	**No**
Scenario 5	20.000	43.800 (1 msg/min)	10 kB	**Yes**
Scenario 6	20.000	43.800 (1 msg/min)	15 kB	**Yes**

The following table shows how data storage has been configured in each scenario, introducing two new terms. **Data availability** represents how long collected data is available to be consumed via an API (i.e. by web clients, mobile apps, ...). After this period, data isn't lost but can't be easily accessed.

> 💡 If this seems too complicated, just think in terms of charting data. If you want to create visualizations, how far back in time should the oldest data point be? That’s your data availability.

Data retention, on the other hand, is the period after which collected data points are deleted.

	Hot storage [days]	Warm storage [days]	Cold storage [days]	Data availability [days]	Data retention [days]
Scenario 1	30	30	365	**60**	425
Scenario 2	30	30	365	**60**	425
Scenario 3	30	30	365	**60**	425
Scenario 4	180	180	365	**360**	725
Scenario 5	180	180	365	**360**	725
Scenario 6	180	180	365	**360**	725

Finally, some **parameters** have been kept **constant** in all scenarios. These are:

Parameter	Value
Connection protocol	MQTT
Connection duration	Always active (*all-day*)
Location	us-east-2
Lambda processing	If present, applied to all devices, with a minimum memory allocation, no concurrency and 10ms of CPU time for processing.
Amazon SNS	10 HTTP notifications, 10 emails per device per month
Amazon API Gateway	2M REST API, with a 0.5 GB cache. 200k Websocket messages per day (10 commands per device per day)
DynamoDB capacity type	Provisioned

Cost analysis

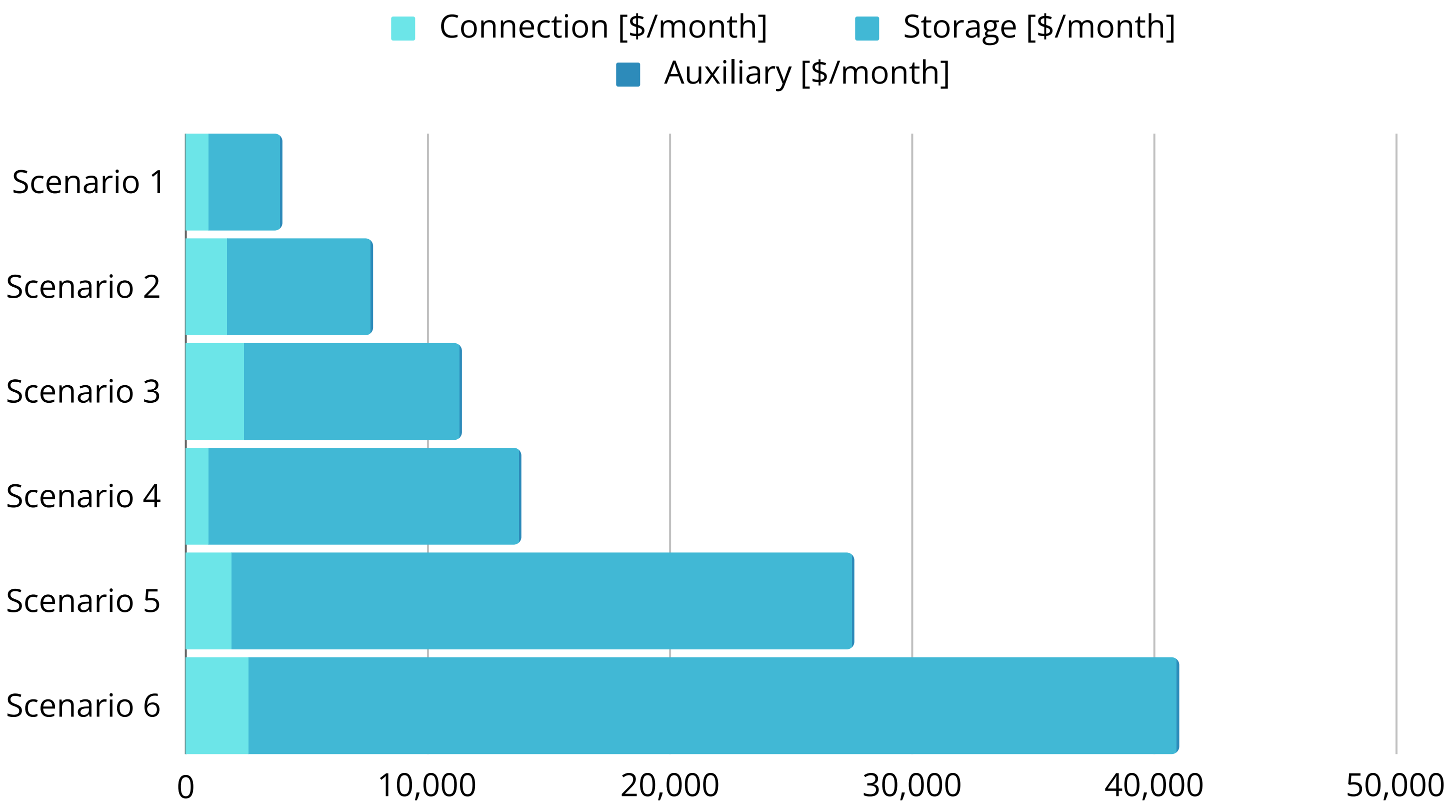
Key assumptions

The following numbers are based on a **single-region** deployment, targeted to the cheapest AWS region we could find (*us-east-2*). As a general rule of thumb, you should consider moving your deployment as close as possible to your customers. If they are distributed across continents, or if you need to ensure higher availability, consider a multi-region deployment too.

All simulations have been performed in six variants - more on that in the “Scenarios” section. Costs have been rounded and, wherever possibile, aggregated for practicality: see the “Additional resources” section for the official estimates obtained through the AWS Pricing Calculator.

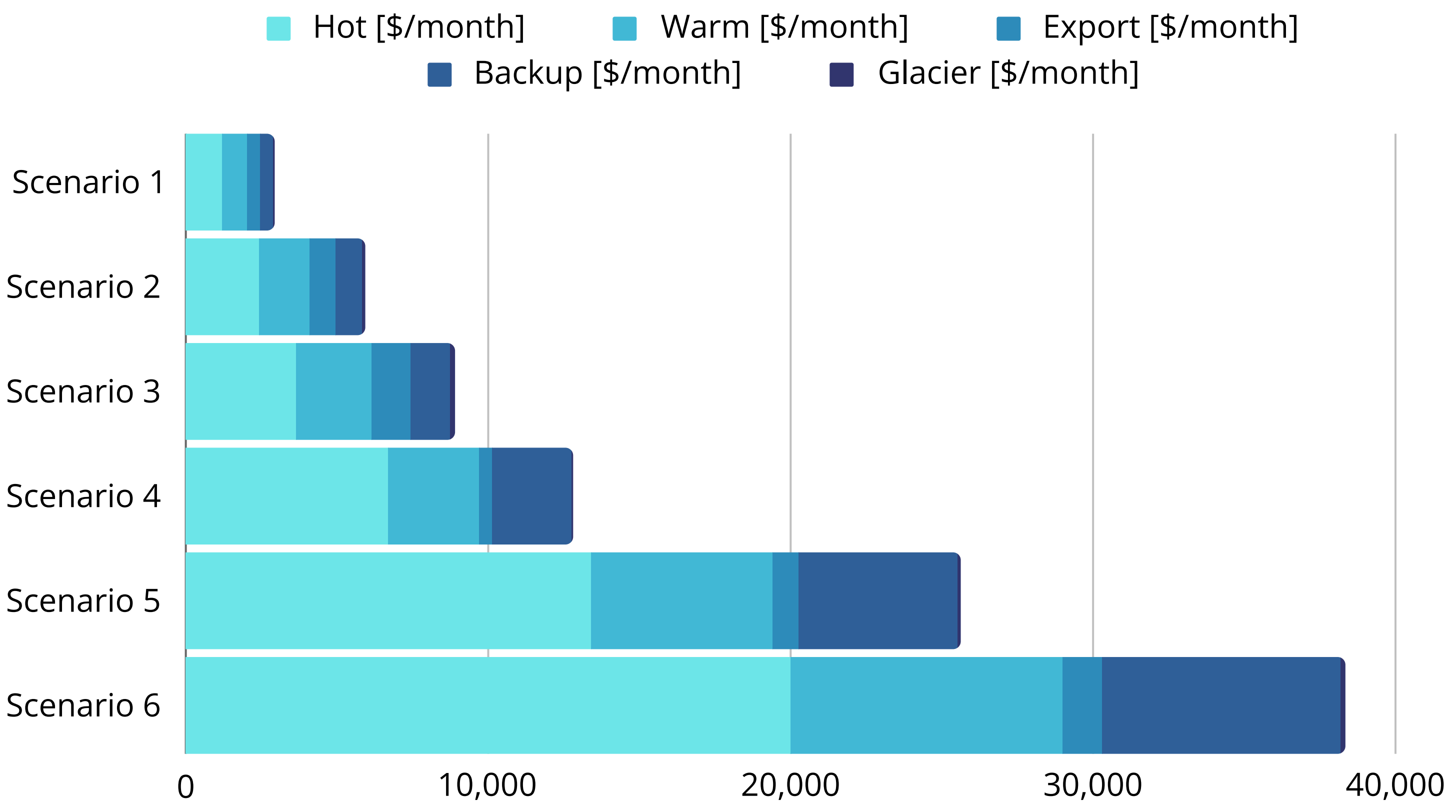
Analysis

Let’s start with a breakdown of costs per **category**:



Variable costs: category breakdown.

As you can see, **storage** is the **major contributor** to the total variable cost of the implementation. If we break down storage costs even more, here’s what we get:

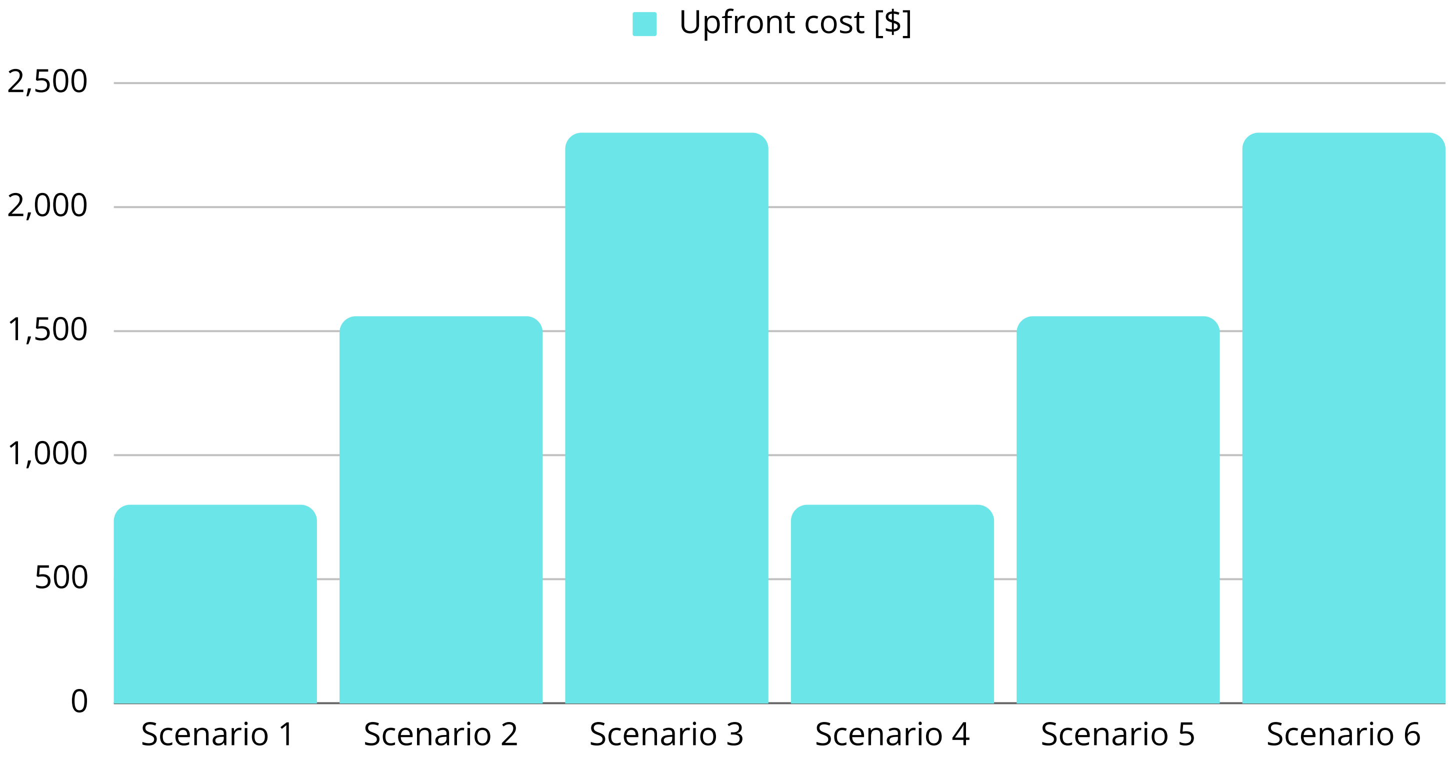


Storage costs breakdown.

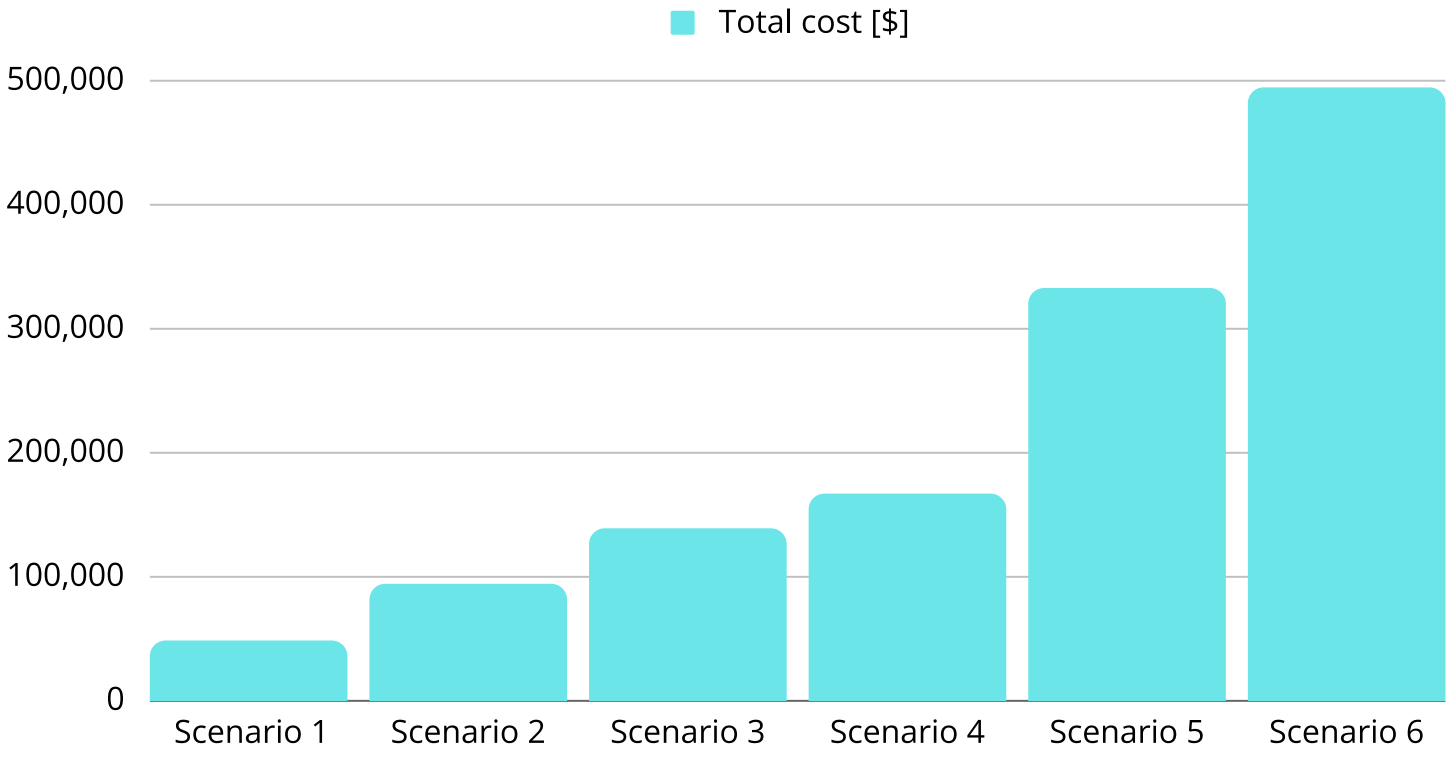
Hot storage is the biggest component in all scenarios.

> 💡 Besides reducing the availability window, an efficient strategy to minimize costs is storing aggregated or processed metrics, instead of raw data.

Additional, albeit limited, contributors to the total cost of an IoT infrastructure on AWS are **upfront costs**.



By adding variable and upfront costs, we can come up with an estimate of the **total cost for the first year** of operations:



As you can see from the chart above, the **cost** for an IoT infrastructure on AWS is **extremely variable**, even if the number of devices is held constant: there's a **10x difference** between the highest and lowest estimate (from $50.000 to $500.000), just for services and compute.

Where to go from here

If you made it this far, congratulations! Hopefully now you have a deeper understanding on what's needed to build an IoT platform and how much it could cost.

Now, the bad news: **this is just half of the puzzle**. To get a usable IoT solution, you're still missing:

- a way of piecing together all the aforementioned services
- customer facing applications (web apps, mobile apps, ...)
- users and permissions management (don't overlook this!)
- support
- additional services (depending on your needs: reports, rules engine, payments systems integration, ...)

This is why we created Connhex: if you want to take a look at it, just visit www.connhex.com .

What if you could keep all the advantages of having **your own IoT** infrastructure without any of the hassle involved in building one?

What you're looking for is Connhex, our IoT suite.

www.connhex.com

About Compiuta

👋 Hi there! We are Compiuta, an Italian sofware company focused on building products at the intersection between **Industrial IoT** and **AI** - our greatest hit so far is Connhex.

As you can probably tell from this document, we have spent quite a lot of time creating IoT solutions: if you'd like to chat about it, just contact us at **info@compiuta.com**.

Additional resources

Pricing estimates

Pricing estimates were obtained through the AWS Pricing Calculator. Additional savings through dedicated contracts might be possible for high traffic volumes.

Scenario 1

Scenario 2

Scenario 3

Scenario 4

Scenario 5

Scenario 6

Costs per-service

In the “Cost Analysis” section, monthly costs have been grouped by category. The following table presents a detailed cost breakdown in a per-service fashion:

Service	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5	Scenario 6
IoT Core (MQTT only)	$950,00	$1.700,00	$2.400,00	$950,00	$1.700,00	$2.400,00
IoT Device Management	$5,00	$5,00	$5,00	$5,00	$5,00	$5,00
Aws Lambda (no concurrency)	$-	$-	$-	$-	$185,00	$185,00
Total (Connection)	**$955,00**	**$1.705,00**	**$2.405,00**	**$955,00**	**$1.890,00**	**$2.590,00**
Dynamo DB (hot storage)	$1.200,00	$2.430,00	$3.650,00	$6.700,00	$13.400,00	$20.000,00
Dynamo DB (warm storage)	$830,00	$1.660,00	$2.493,00	$3.000,00	$6.000,00	$9.000,00
Dynamo DB - Export storage to S3	$430,00	$870,00	$1.300,00	$430,00	$870,00	$1.300,00
Dynamo DB - On demand backup	$430,00	$870,00	$1.300,00	$2.628,00	$5.250,00	$7.884,00
S3 - Glacier Deep Archive	$50,00	$105,00	$160,00	$50,00	$105,00	$160,00
Total (Storage)	**$2.940,00**	**$5.935,00**	**$8.903,00**	**$12.808,00**	**$25.625,00**	**$38.344,00**
CloudWatch	$50,00	$50,00	$50,00	$50,00	$50,00	$50,00
Key Management System	$10,00	$10,00	$10,00	$10,00	$10,00	$10,00
SNS	$5,00	$5,00	$5,00	$5,00	$5,00	$5,00
Amazon API Gateway	$30,00	$30,00	$30,00	$30,00	$30,00	$30,00
Total (Auxiliary)	**$95,00**	**$95,00**	**$95,00**	**$95,00**	**$95,00**	**$95,00**
Total	**$3.990,00**	**$7.735,00**	**$11.403,00**	**$13.858,00**	**$27.610,00**	**$41.029,00**

Legal notice

This document and its contents are the exclusive property of Compiuta S.r.l.

All rights are reserved. No part of this document may be reproduced, distributed, or transmitted in any form or by any means, including photocopying, recording, or other electronic or mechanical methods, without the prior written permission of the copyright owner, except in the case of brief quotations with clear attribution and certain other noncommercial uses permitted by copyright law.

The content of this document is provided "as is" without warranty of any kind, either express or implied, including but not limited to the implied warranties of merchantability, fitness for a particular purpose, or non-infringement. Compiuta S.r.l. assumes no responsibility for errors or omissions in the content or other documents which are referenced by or linked to this document. In no event shall Compiuta S.r.l. be liable for any special, incidental, indirect, or consequential damages of any kind, or any damages whatsoever, including, without limitation, those resulting from loss of use, data, or profits, whether or not advised of the possibility of damage, and on any theory of liability arising out of or in connection with the use or performance of this information.