Cloud Security Best Practices

by Jonathan Marcil
February 2025

Content overview

➔ **Now and then**
A bit of history, a bit of realization.

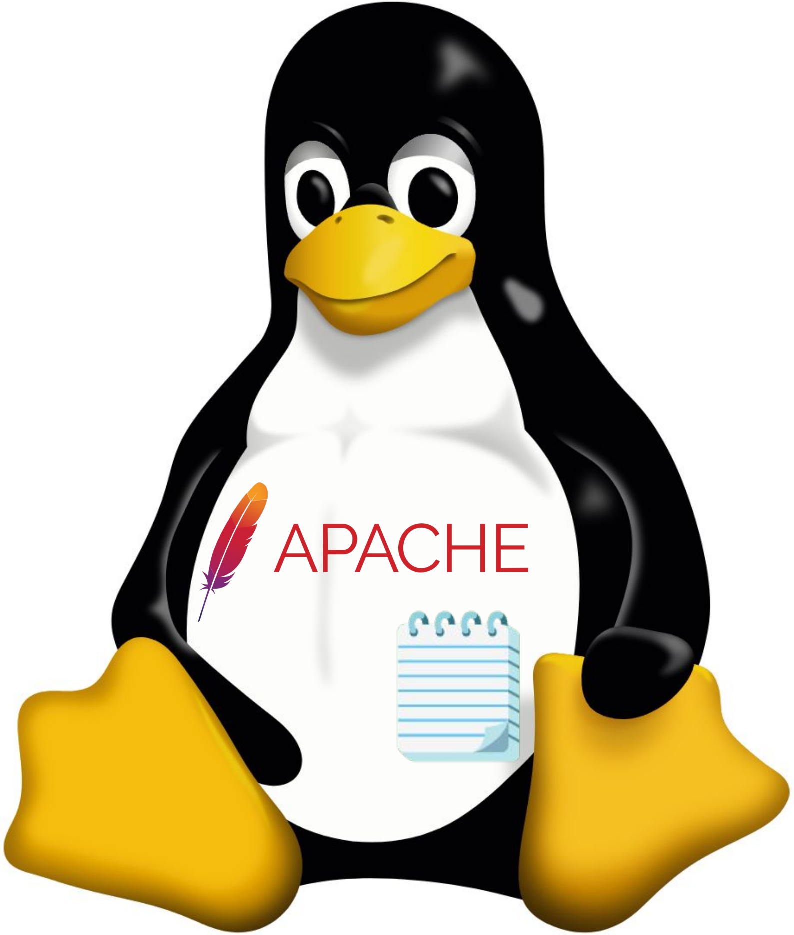
➔ **Security Impact**
How cloud providers are shaping security.

➔ **Patterns**
A security-related selection based on what is needed the most often in my experience.

Boom! How it began.

Not so long ago, in the land of open source…

In an oversimplified fashion…



Install Linux

Install Platform App

Add your code

DONE

Zoom! How it is now.

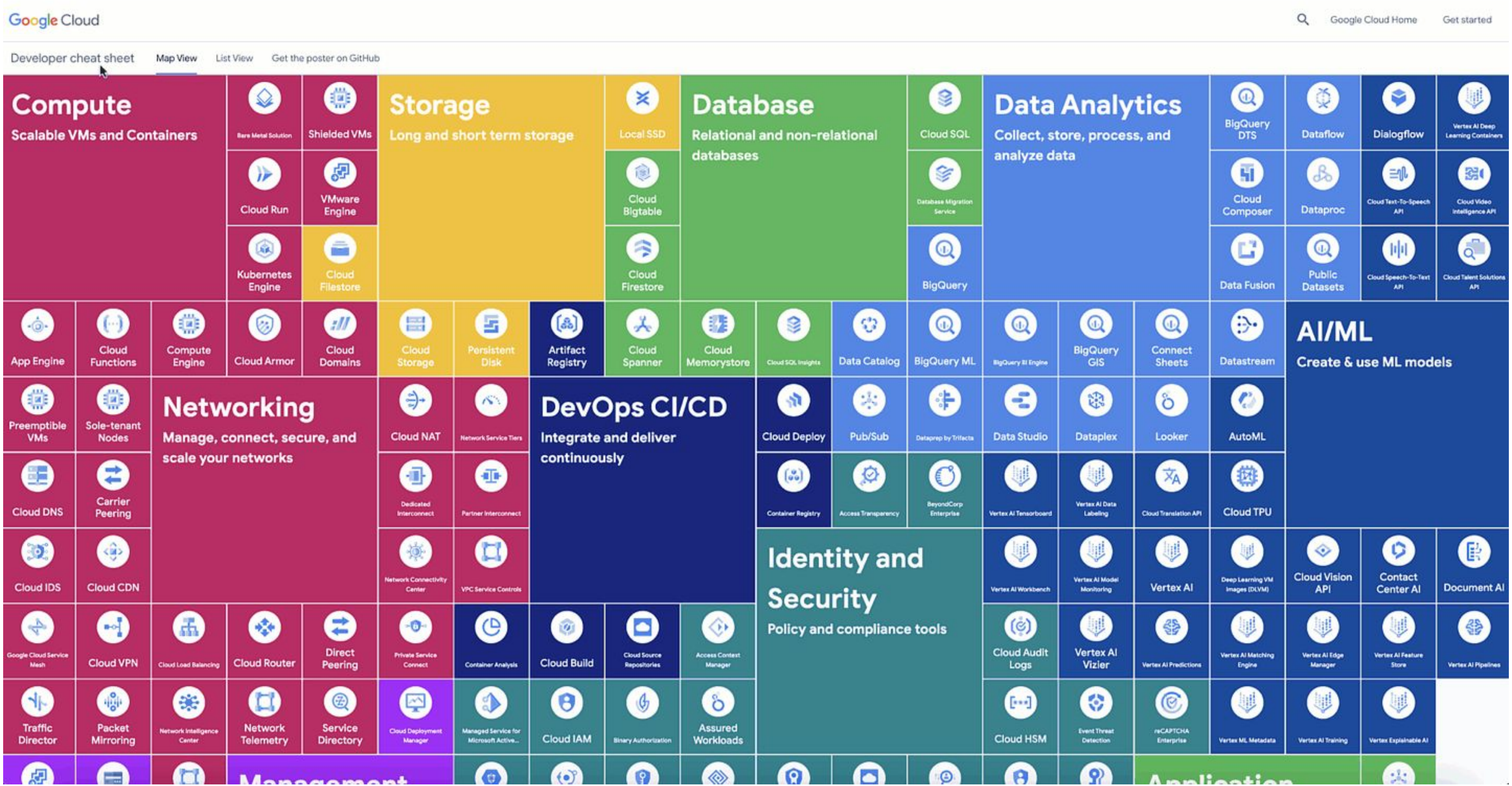
Evolution has come to this…

Select Cloud Provider

420 services (extracted from APIs)

Select Cloud Provider

Select Cloud Service



Compute

Scalable VMs and Containers

Bare Metal Solution

Shielded VMs

Storage

Long and short term storage

Local SSD

Database

Relational and non-relational databases

Cloud SQL

Data Analytics

Collect, store, process, and analyze data

BigQuery DTS

Dataflow

Dialogflow

Vertex AI Deep Learning Containers

Cloud Run

VMware Engine

Cloud Bigtable

Database Migration Service

Cloud Composer

Dataproc

Cloud Text-To-Speech API

Cloud Video Intelligence API

Kubernetes Engine

Cloud Filestore

Cloud Firestore

BigQuery

Data Fusion

Public Datasets

Cloud Speech-To-Text API

Cloud Talent Solutions API

App Engine

Cloud Functions

Compute Engine

Cloud Armor

Cloud Domains

Cloud Storage

Persistent Disk

Artifact Registry

Cloud Spanner

Cloud Memorystore

Cloud SQL Insights

Data Catalog

BigQuery ML

BigQuery BI Engine

BigQuery GIS

Connect Sheets

Datastream

AI/ML

Create & use ML models

Preemptible VMs

Sole-tenant Nodes

Networking

Manage, connect, secure, and scale your networks

Cloud NAT

Network Service Tiers

DevOps CI/CD

Integrate and deliver continuously

Cloud Deploy

Pub/Sub

Dataprep by Trifacta

Data Studio

Dataplex

Looker

AutoML

Cloud DNS

Carrier Peering

Dedicated Interconnect

Partner Interconnect

Container Registry

Access Transparency

BeyondCorp Enterprise

Vertex AI Tensorboard

Vertex AI Data Labeling

Cloud Translation API

Cloud TPU

Cloud IDS

Cloud CDN

Network Connectivity Center

VPC Service Controls

Identity and Security

Policy and compliance tools

Vertex AI Workbench

Vertex AI Model Monitoring

Vertex AI

Deep Learning VM Images (DLVM)

Cloud Vision API

Contact Center AI

Document AI

Google Cloud Service Mesh

Cloud VPN

Cloud Load Balancing

Cloud Router

Direct Peering

Private Service Connect

Container Analysis

Cloud Build

Cloud Source Repositories

Access Context Manager

Cloud Audit Logs

Vertex AI Vizier

Vertex AI Predictions

Vertex AI Matching Engine

Vertex AI Edge Manager

Vertex AI Feature Store

Vertex AI Pipelines

Traffic Director

Packet Mirroring

Network Intelligence Center

Network Telemetry

Service Directory

Cloud Deployment Manager

Managed Service for Microsoft Active...

Cloud IAM

Binary Authorization

Assured Workloads

Cloud HSM

Event Threat Detection

reCAPTCHA Enterprise

Vertex ML Metadata

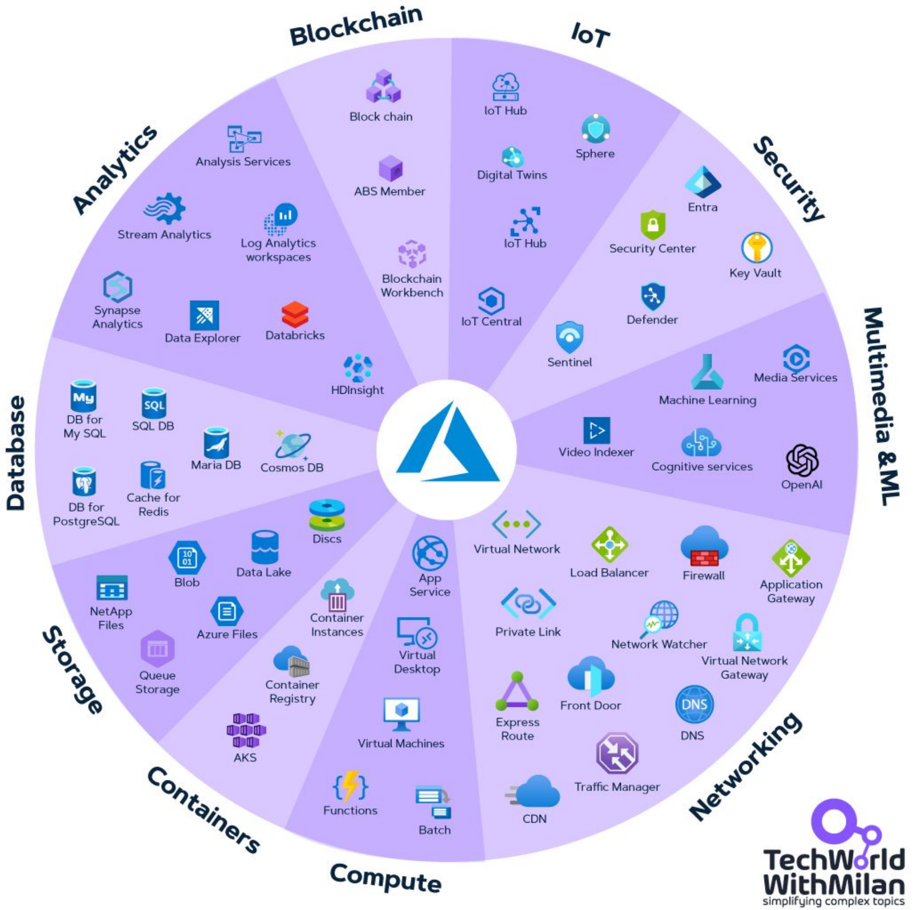
Vertex AI Training

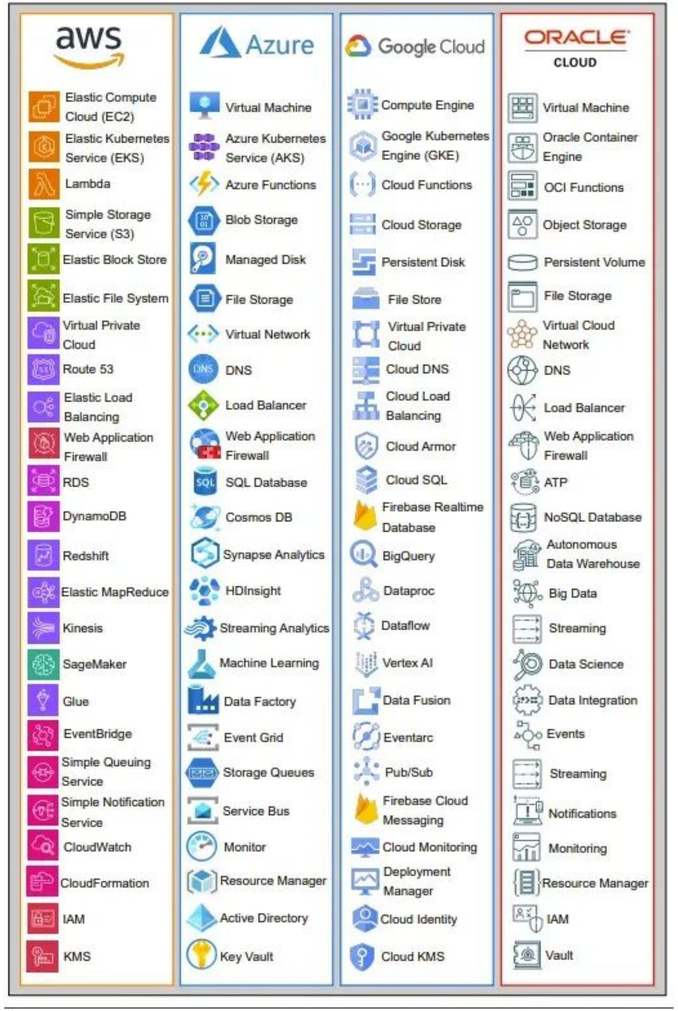
Vertex Explainable AI

Management

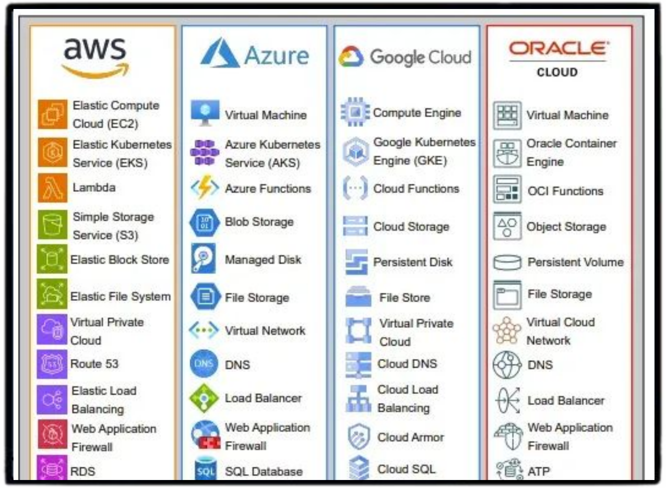
Application

AZURE CLOUD SERVICES CHEAT SHEET





aws	Azure	Google Cloud	ORACLE CLOUD
Elastic Compute Cloud (EC2)	Virtual Machine	Compute Engine	Virtual Machine
Elastic Kubernetes Service (EKS)	Azure Kubernetes Service (AKS)	Google Kubernetes Engine (GKE)	Oracle Container Engine
Lambda	Azure Functions	Cloud Functions	OCI Functions
Simple Storage Service (S3)	Blob Storage	Cloud Storage	Object Storage
Elastic Block Store	Managed Disk	Persistent Disk	Persistent Volume
Elastic File System	File Storage	File Store	File Storage
Virtual Private Cloud	Virtual Network	Virtual Private Cloud	Virtual Cloud Network
Route 53	DNS	Cloud DNS	DNS
Elastic Load Balancing	Load Balancer	Cloud Load Balancing	Load Balancer
Web Application Firewall	Web Application Firewall	Cloud Armor	Web Application Firewall
RDS	SQL Database	Cloud SQL	ATP
DynamoDB	Cosmos DB	Firebase Realtime Database	NoSQL Database
Redshift	Synapse Analytics	BigQuery	Autonomous Data Warehouse
Elastic MapReduce	HDInsight	Dataproc	Big Data
Kinesis	Streaming Analytics	Dataflow	Streaming
SageMaker	Machine Learning	Vertex AI	Data Science
Glue	Data Factory	Data Fusion	Data Integration
EventBridge	Event Grid	Eventarc	Events
Simple Queuing Service	Storage Queues	Pub/Sub	Streaming
Simple Notification Service	Service Bus	Firebase Cloud Messaging	Notifications
CloudWatch	Monitor	Cloud Monitoring	Monitoring
CloudFormation	Resource Manager	Deployment Manager	Resource Manager
IAM	Active Directory	Cloud Identity	IAM
KMS	Key Vault	Cloud KMS	Vault



aws	Azure	Google Cloud	ORACLE CLOUD
Elastic Compute Cloud (EC2)	Virtual Machine	Compute Engine	Virtual Machine
Elastic Kubernetes Service (EKS)	Azure Kubernetes Service (AKS)	Google Kubernetes Engine (GKE)	Oracle Container Engine
Lambda	Azure Functions	Cloud Functions	OCI Functions
Simple Storage Service (S3)	Blob Storage	Cloud Storage	Object Storage
Elastic Block Store	Managed Disk	Persistent Disk	Persistent Volume
Elastic File System	File Storage	File Store	File Storage
Virtual Private Cloud	Virtual Network	Virtual Private Cloud	Virtual Cloud Network
Route 53	DNS	Cloud DNS	DNS
Elastic Load Balancing	Load Balancer	Cloud Load Balancing	Load Balancer
Web Application Firewall	Web Application Firewall	Cloud Armor	Web Application Firewall
RDS	SQL Database	Cloud SQL	ATP

Select Cloud Provider

Select Cloud Service

Setup Cloud Service

Add your code

DONE?

Meanwhile...

Cloud providers entered a race of repackaging apps that **YOU** have to select, configure and adapt your code to.

What does that do?

You don't have to think about OS level anymore...

You replace a lot of your code with services and their functionalities...

While concepts remains the same, you are vendor locked into a taxonomy and details...

All of this isn't inherently good or bad...

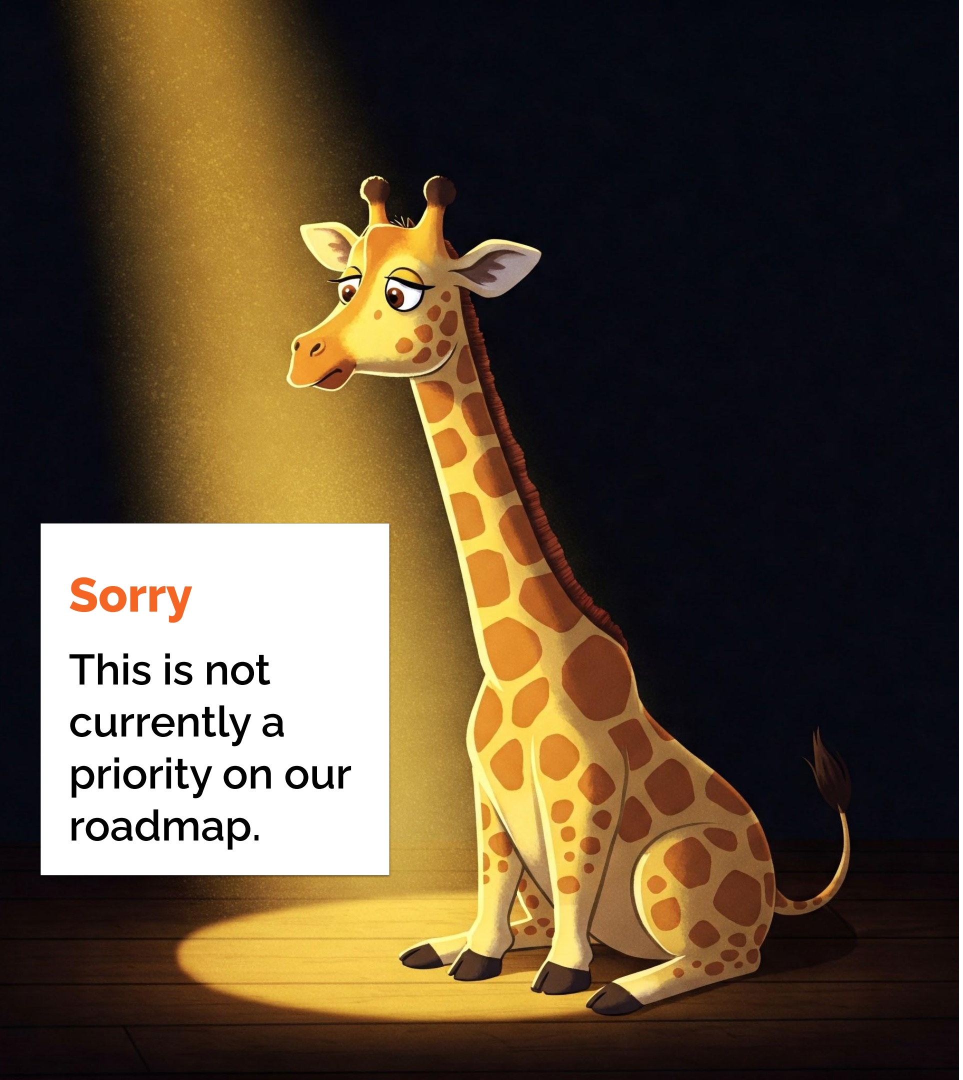
Impact on Security

You don’t have to think about OS level anymore…

Security at the OS level is rock solid, as experienced elite experts are doing the OS configuration, and even updating software for you.

Nothing can go wrong with that?

Until you need a particular update



Sorry

This is not currently a priority on our roadmap.

What if you are using a **software package** and need it updated, and then the provider doesn’t update it quickly?

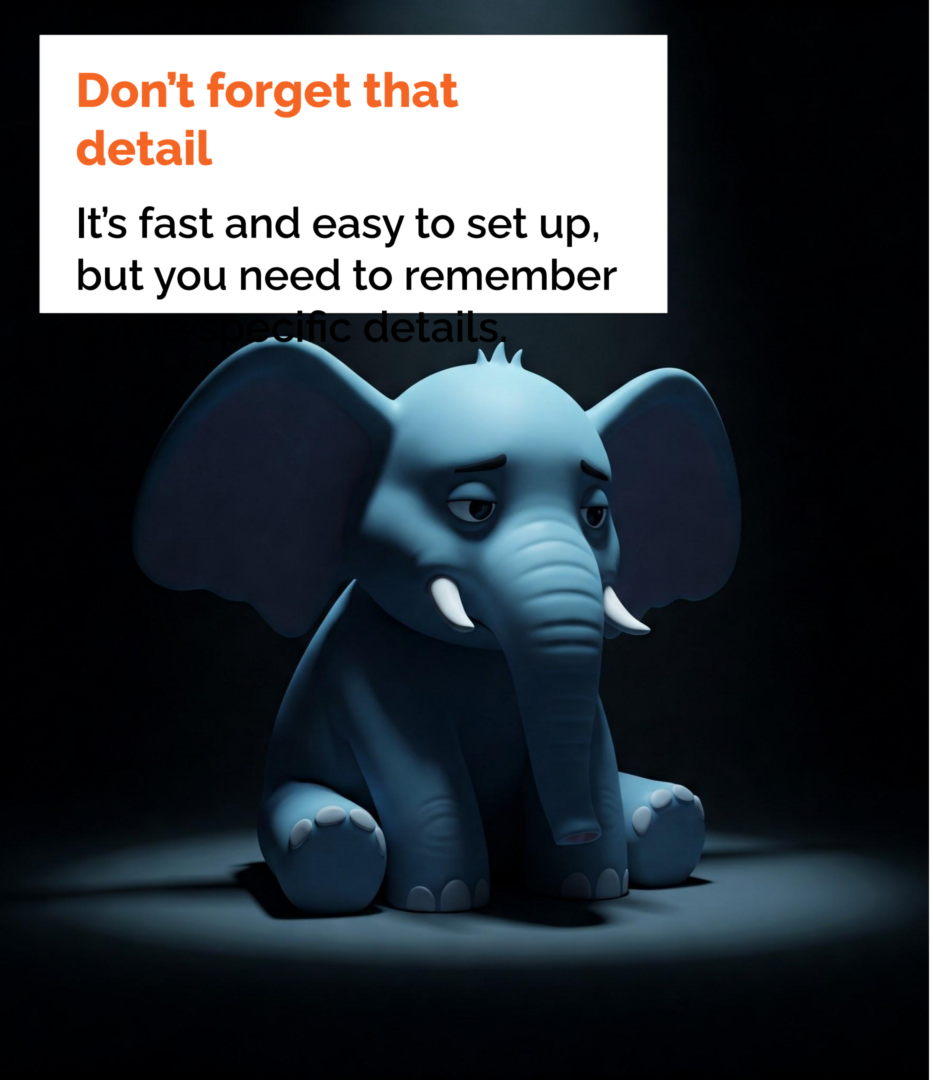
For you, it might a critical vulnerability, but for their total user base it’s not.

Impact on Security

You replace a lot of your code with services and their functionalities...

Implementation of security protocols are secure and robust as tons of paid users share that “code base”, making it like a well maintained library

No cloud provider wants their service to be insecure...



Don’t forget that detail

It’s fast and easy to set up, but you need to remember specific details.

Until you reconfigure for your need

It worked so easily out of the box, but what if you need a **different setup than the default** configuration.

What was easy and required little understanding of the system, now could **rely on details** to not be a misconfiguration that creates a breach.

What does that do?

While concepts remains the same, you are vendor locked into a taxonomy and details...

Secure integration is streamlined, services are (mostly) designed to fit together and security access controls works in an uniform fashion.

While nobody likes to see them self vendor locked, for security, uniformity makes it easier...

Integration supported but..

you need to understand the federation protocol and implementation details specific to all parties.

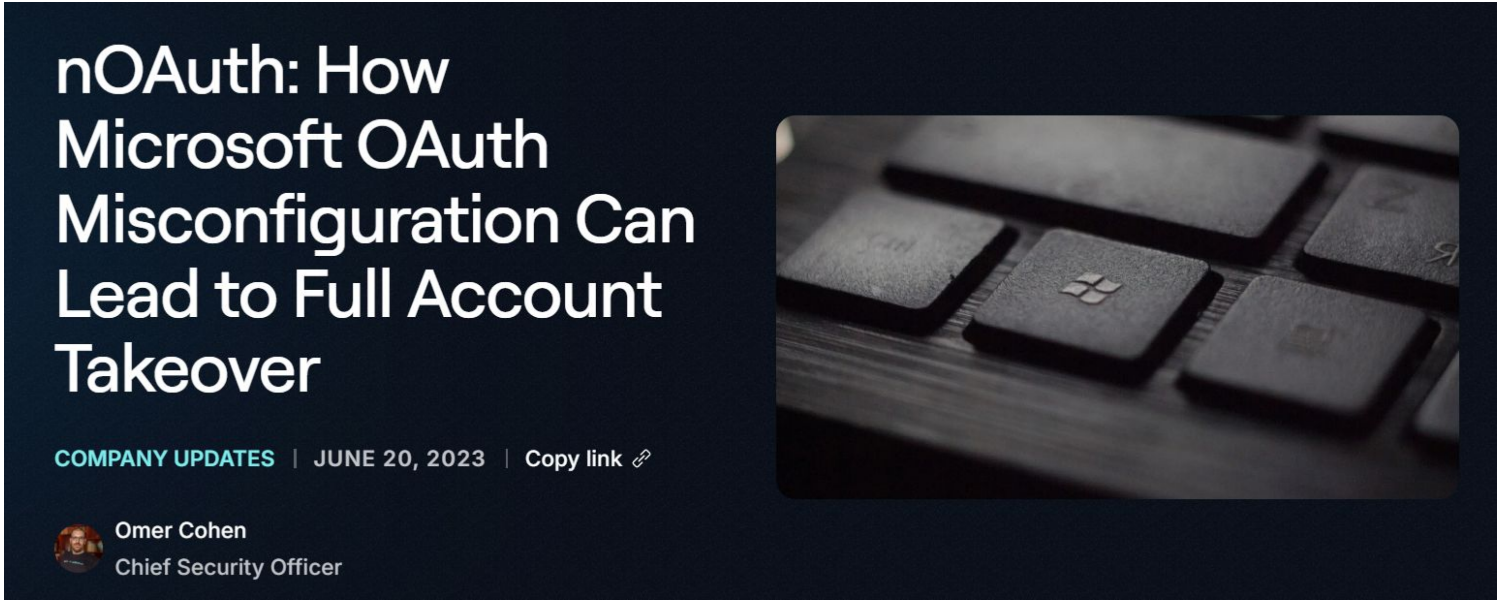
Until you need to integrate with another cloud or system

While the functionality is there, it’s sometimes gated by a higher tier paid price made for large scale enterprises.

The risk is when you are forced to stitch together a security solution for cross-cloud integration.

Don’t use email as primary key

In some clouds, email are mutable or unverified.



nOAuth: How Microsoft OAuth Misconfiguration Can Lead to Full Account Takeover

COMPANY UPDATES | JUNE 20, 2023 | Copy link

Omer Cohen
Chief Security Officer

Microsoft Guidance

https://learn.microsoft.com/en-us/entra/identity-platform/migrate-off-email-claim-authorization 👉

How do I know if my application is impacted?

Microsoft recommends reviewing application source code and determining whether the following patterns are present:

- **A mutable claim, such as `email`, is used for the purposes of uniquely identifying a user**
- A mutable claim, such as `email` is used for the purposes of authorizing a user's access to resources

These patterns are considered insecure, as users without a provisioned mailbox can have any email address set for their Mail (Primary SMTP) attribute. **This attribute is not guaranteed to come from a verified email address.** When an email claim with an unverified domain owner is used for authorization, any user without a provisioned mailbox has the potential to gain unauthorized access by changing their Mail attribute to impersonate another user.

An email is considered to be domain-owner verified if:

- The domain belongs to the tenant where the user account resides, and the tenant admin has done verification of the domain
- The email is from a Microsoft Account (MSA)
- The email is from a Google account
- The email was used for authentication using the one-time passcode (OTP) flow

It should also be noted that Facebook and SAML/WS-Fed accounts don't have verified domains.

This risk of unauthorized access has only been found in multi-tenant apps, as a user from one tenant could escalate their privileges to access resources from another tenant through modification of their Mail attribute.

Pause for dramatic effect

And maybe take a sip of water?

Security friendly cloud architecture patterns

These are a selection of some patterns
I have concrete experience with.

HTTPS Load Balancer

HTTPS Load Balancer

For any HTTP URL you want to expose.

Plain text ports shouldn't be associated with a **public IP address**.

TLS/SSL certificate management is handled by the cloud provider.

Variations exists for other protocols.

Hostname checking

SNI (multiple hosts on same IP) can serve the wrong hostname for your app, which can act unexpectedly.

$$$

In some cloud, choosing to manage your own SSL provider is easy, but might cost significant per month fees.

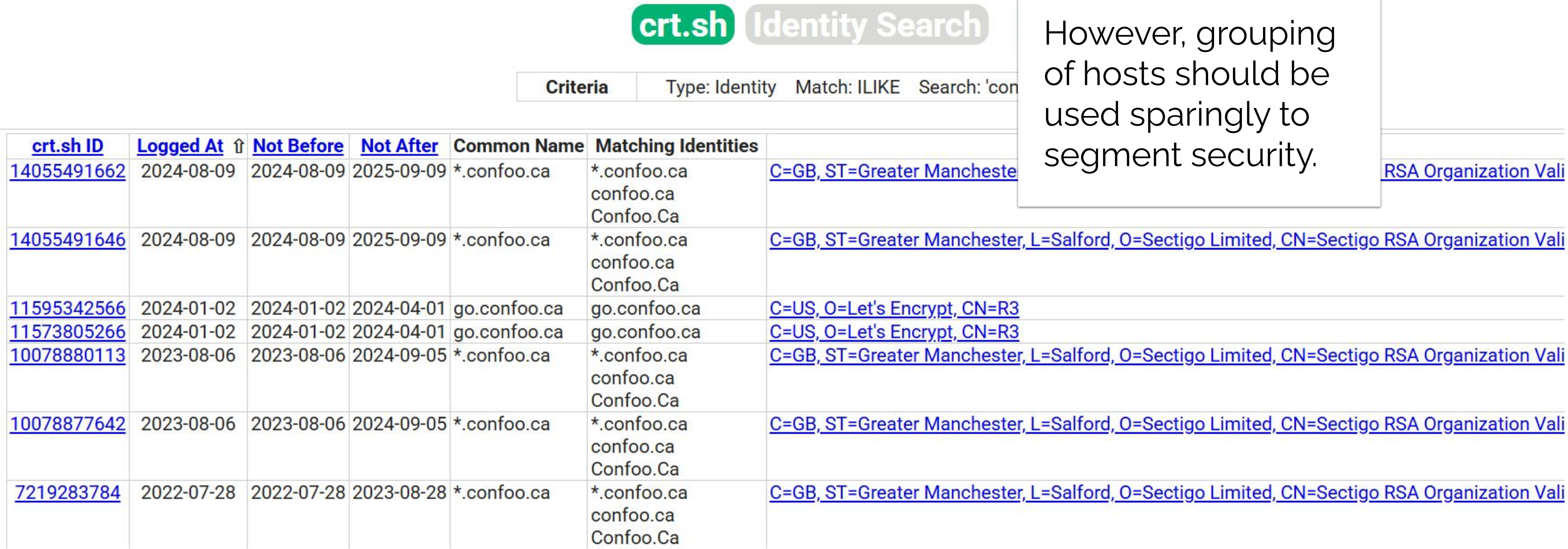
crt.sh

When registering SSL certificate, transparency is a feature that expose all entries to the world.

Good job Confoo!

Using wildcards to hide hosts

However, grouping of hosts should be used sparingly to segment security.



crt.sh Identity Search

Criteria	Type: Identity Match: ILIKE Search: 'con

crt.sh ID	Logged At ⇧	Not Before	Not After	Common Name	Matching Identities	
14055491662	2024-08-09	2024-08-09	2025-09-09	*.confoo.ca	*.confoo.ca confoo.ca Confoo.Ca	C=GB, ST=Greater Mancheste RSA Organization Vali
14055491646	2024-08-09	2024-08-09	2025-09-09	*.confoo.ca	*.confoo.ca confoo.ca Confoo.Ca	C=GB, ST=Greater Manchester, L=Salford, O=Sectigo Limited, CN=Sectigo RSA Organization Vali
11595342566	2024-01-02	2024-01-02	2024-04-01	go.confoo.ca	go.confoo.ca	C=US, O=Let's Encrypt, CN=R3
11573805266	2024-01-02	2024-01-02	2024-04-01	go.confoo.ca	go.confoo.ca	C=US, O=Let's Encrypt, CN=R3
10078880113	2023-08-06	2023-08-06	2024-09-05	*.confoo.ca	*.confoo.ca confoo.ca Confoo.Ca	C=GB, ST=Greater Manchester, L=Salford, O=Sectigo Limited, CN=Sectigo RSA Organization Vali
10078877642	2023-08-06	2023-08-06	2024-09-05	*.confoo.ca	*.confoo.ca confoo.ca Confoo.Ca	C=GB, ST=Greater Manchester, L=Salford, O=Sectigo Limited, CN=Sectigo RSA Organization Vali
7219283784	2022-07-28	2022-07-28	2023-08-28	*.confoo.ca	*.confoo.ca confoo.ca Confoo.Ca	C=GB, ST=Greater Manchester, L=Salford, O=Sectigo Limited, CN=Sectigo RSA Organization Vali

👉 https://crt.sh/

Authorization Proxy

Authorization Proxy

Often done with the Load Balancer, “simple” to enable with defaults.

Adds a **layer of authorization** that can be connected to authentication managed by the cloud provider instead of your app.

Most likely will use something like OpenID Connect (OIDC) and OAuth2.

Authorization Proxy: headers

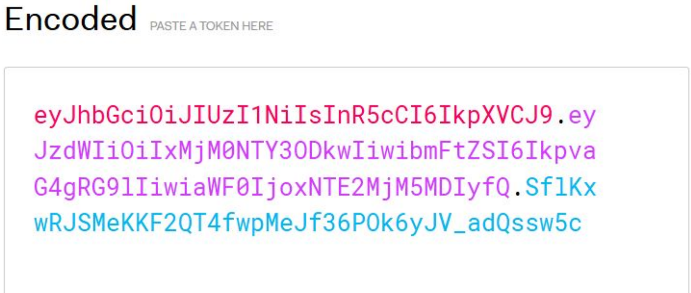
Varies by cloud provider, and are added inline in every request by the proxy.

Your app receives them as request headers.

Headers Example (Azure)

```
{
    'Disguised-Host': 'jonathan-test-headers.azurewebsites.net',
    'Host': 'jonathan-test-headers.azurewebsites.net',
    'X-Appservice-Proto': 'https',
    'X-Client-Ip': '107.159.175.56',
    'X-Client-Port': '56344',
    'X-Forwarded-For': '107.159.175.56:56344',
    'X-Forwarded-Proto': 'https',
    'X-Forwarded-Tlsversion': '1.3',
    'X-Ms-Client-Principal': 'eyJhdXR[...]xlIn0=',
    'X-Ms-Client-Principal-Id': '9db[...]3',
    'X-Ms-Client-Principal-Idp': 'aad',
    'X-Ms-Client-Principal-Name': 'Jonathan Marcil',
    'X-Ms-Token-Aad-Access-Token': 'eyJ0eXAi[...]L2QQ',
    'X-Ms-Token-Aad-Expires-On': '2025-02-19T16:19:49.7332612Z',
    'X-Ms-Token-Aad-Id-Token': 'eyJ0eX[...]X85TA',
    'X-Ms-Token-Aad-Refresh-Token': '1.ASkA[...]0O-B',
    'X-Original-Url': '/headers',
    'X-Site-Deployment-Id': 'jonathan-test-headers'
}
```

Authorization Proxy: tokens

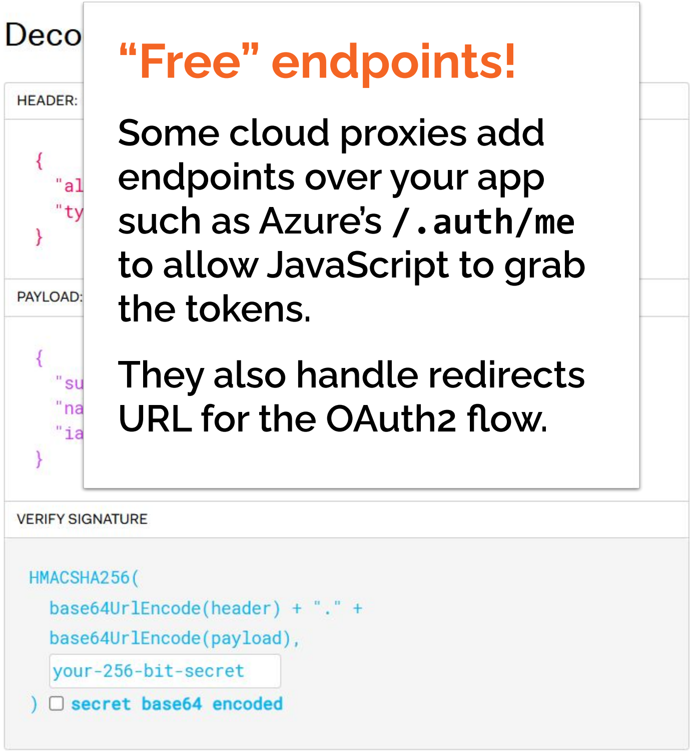


Encoded PASTE A TOKEN HERE

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.ey
JzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6Ikpva
G4gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ.SflKx
wRJSMeKKF2QT4fwpMeJf36POk6yJV_adQssw5c
```

HTTP Headers sent to your app

Librairies exist to handle them in your application code, but under the hood it’s JWT tokens.



Deco

HEADER:

```
{
  "al
  "ty
}
```

PAYLOAD:

```
{
  "su
  "na
  "ia
}
```

VERIFY SIGNATURE

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  your-256-bit-secret
) □ secret base64 encoded
```

“Free” endpoints!

Some cloud proxies add endpoints over your app such as Azure’s `/.auth/me` to allow JavaScript to grab the tokens.

They also handle redirects URL for the OAuth2 flow.

👉 https://jwt.io/

Authorization Proxy: JWT

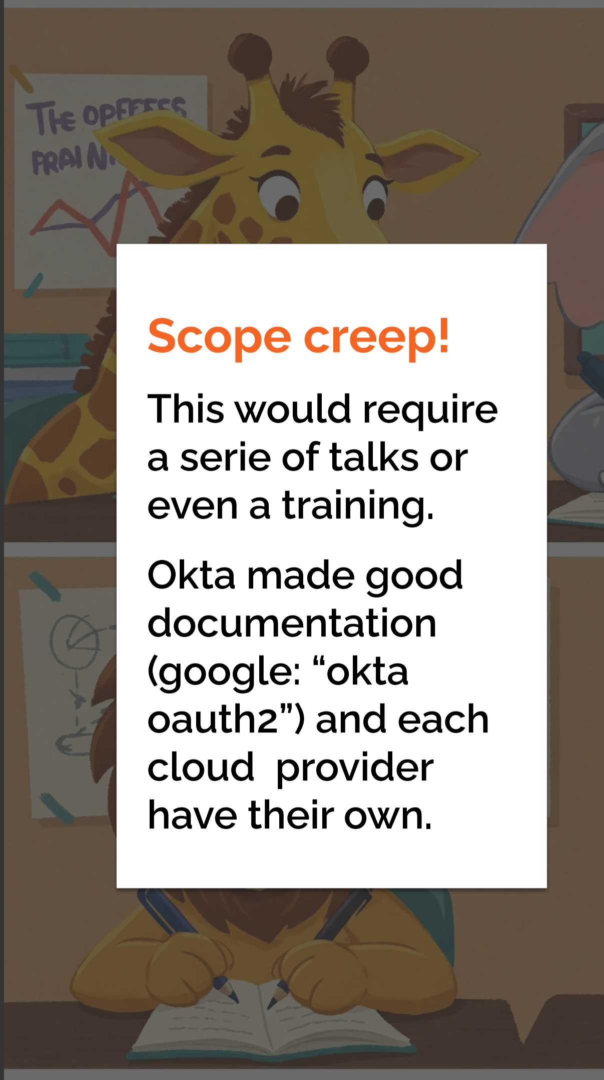
JSON Web Tokens contains claims that gives information about the logged in user

They are cryptographically signed to ensure authenticity

ID Token Example (Google)

```
{
  "iss": "https://accounts.google.com",
  "aud": "32555350559.apps.googleusercontent.com",
  "sub": "111260650121185072906",
  "hd": "google.com",
  "email": "user@example.com",
  "email_verified": "true",
  "at_hash": "_LLKKivfvfme9eoQ3WcMIg",
  "iat": "1650053185",
  "exp": "1650056785",
  "alg": "RS256",
  "kid": "f1338ca26835863f671403941738a7b49e740fc0",
  "typ": "JWT"
}
```

Understanding JWT, OAuth2 and OIDC can be counter-intuitive but rewarding when creating solutions



Scope creep!

This would require a serie of talks or even a training.

Okta made good documentation (google: “okta oauth2”) and each cloud provider have their own.

Authorization Proxy

Essentially you can trust the cloud provider to handle protocol integration.

If your application require no authorization logic, you're done.

However if you need to handle users, you have to be careful with your trust model.

Trusting headers

For high security needs, only trust signed headers.

If you trust other header, make sure they are safe (`X-Client-IP` and not `X-Forwarded-For`).

Zero trust

Your application shouldn't do anything unless authorization goes thought.

Trusting tokens

In your code, rely on libraries and make sure you are checking the signature of tokens using a hardcoded validation type.

Trusting claims

Use unique cloud identifier to identify a user and not their email as they can be unverified or changed.

envoy

Do it yourself

Envoy proxy is used in many big cloud providers to provide load balancing and handle the authorization layer.

Some providers will even admit that the solution is based on Envoy.

www.envoyproxy.io

Developer Access Tunnel

Dev access tunnel

You have a service, could be a database or caching system that developers on their local machine need access to.

Some cloud providers might just offer to expose the service port with a public IP, and then it's up to you to add restrictions.

A better alternative would be to provide a TLS-secured and authenticated path from their machine to the cloud.

Jumpbox

Create a SSH only instance in the same VPC network than the target service.

IP restrict this one!

No-jumpbox

If you're lucky your cloud will give you a facility to SSH and forward ports without the need to spin your own jumpbox.

IAM

Major clouds will manage the access using their native IAM if you ssh using their command line tools.

Free logging

The tunnel gives you service access logs by having the SSH connection and/or IAM check loggable.

Dev access tunnel: jumpbox

Plain old SSH port forwarding

```
ssh user@cloud-instance.provider.com -L 1234:10.1.1.4:5432


-L [local port]:[service ip]:[service port]
```

Inside the SSH instance

```
psql -h 10.1.1.4 -p 5432
```

≡

On the local machine

```
psql -h localhost -p 1234
```

GitHub Deploys Without Keys

GitHub Actions OIDC

You want to deploy into your cloud using GitHub Actions.

Instead of using shared secrets, you can authorize GitHub repos to deploy.

Harder to configure, easier to handle security as you rely on GitHub claims.

Validate org by org id

When you configure your cloud, make sure you validate the `repository_owner_id` and not just the repo name (not unique) or org name (can change over time).

Official GitHub Documentation

Cloud Logging

Encryption at REST

Flash round

Some small patterns for quick wins

- ➔ **Use Cloud Logging**
  You might need to enable and configure it, but you can get **write-only logs** that are useful for high security level requirements.
- ➔ **Encryption at rest**
  This is basically free with cloud storage; no need to worry about someone stealing your hard-drive.

At rest limits

Doesn’t protect access to your data from any of the applications that has access to storage.

Use IAM with the level you need.

Cloud Security Cheat Sheet

Maps the patterns into AWS, Azure, GCP

jonathanmarcil.ca/cloud-security

THANKS!

Slides and links on:

about.jonathanmarcil.ca 👉

Special thanks to:

IVADO Labs

Confoo 2025

Camile

ConFoo.CA
DEVELOPER CONFERENCE

JONATHAN MARCIL
INTERNATIONAL