Qalius CONSULTING INC.

WHITEPAPER:

Keep Your Healthcare Data Safe – or How to be Like an Orchid

Abstract

Qalius collaborates with Canadian clinicians and researchers to build applications for healthcare use cases on Amazon Web Services (AWS). Data security is a primary consideration because the applications typically process and store personal health information (PHI) that is protected by Canadian legislation. When working with non-technical clinicians and researchers, we use analogies to explain how the application uses AWS services to keep the PHI secure. This whitepaper explains a few concepts that are critical for keeping PHI secure by comparing the PHI to nectar from a special species of orchid from Madagascar. In nature the orchid preserves the nectar for a particular species of moth. In IT the AWS services keep the PHI secure for authorized users of the healthcare application.

Healthcare Applications

Clinicians and researchers in Canada are using state-of-the-art cloud technology to meet the demands of patients, their families, and patient-care teams for intuitive, convenient and secure software applications. Healthcare applications should deliver experiences comparable to the best experiences in the consumer digital economy. The development process needs to get new features to market quickly, often incorporating the latest technologies for data analytics or artificial intelligence. The extra challenge for clinicians and researchers is to meet these expectations for innovation while ensuring that the associated data remains secure.

Applications for healthcare uses cases often contain personal health information (PHI). PHI needs to be protected as it is collected, processed, and stored by the application. Not only is this a moral responsibility, but it is a legal obligation: refer to the panel Regulatory Framework in Canada for more information. Sometimes these obligations can feel like a disincentive to innovation in digital healthcare.

Fortunately, Amazon Web Services (AWS), offers a large catalogue of services that are proven worldwide to deliver critical applications for enterprises and governments. AWS services are standardized, so the features that secure the world's largest government and enterprise applications are available even to relatively small Canadian healthcare customers. We can configure AWS services to deliver healthcare applications with cutting-edge features while keeping PHI secure.

When the Qalius team is discussing healthcare applications with clinicians and researchers in Canada, we often use analogies to explain the security features. The analogies help explain the ways in which we assemble the application from AWS services to ensure the PHI is secure. One of our favourite analogies compares authorized application users to a moth and the personal health information to nectar stored deep inside a very peculiar orchid. If you are considering building applications on AWS that will process personal health information, then stick with us as we use a bit of biology to explain some information technology concepts.

A Special Orchid and A Special Moth

Charles Darwin (1809 – 1882) is famous for proposing his theory of evolution through interpreting his meticulous surveys of plants and animals. He had a particular interest in the natural variation in species of orchids. He conducted controlled experiments to demonstrate that orchids reproduce most effectively if they can exchange pollen with another orchid of the same species, and he understood the role of moths to carry pollen from one orchid to another. The orchid produces nectar that attracts the moth to feed. Pollen sticks to the moth and is transferred to the next orchid as the moth drinks its next meal. The pollination is most efficient when the moth prefers to visit one species of orchid.

In 1862 Darwin received a collection of orchids from Madagascar, including an unusual orchid called Angraecum sesquipedale. This orchid has a nectary that is nearly 30cm down a narrow tube. At that time, there was no known moth that could access the nectar, so the orchid posed a puzzle. If the purpose of the nectar is to attract a moth for feeding, why would the orchid put the nectar where it is inaccessible? Darwin's answer, published in his book on orchids in 1862, was that there must be a moth in Madagascar that is adapted to reach 30cm down into the nectary. In 1903, more than 40 years after Darwin's prediction, the moth Xanthopan morgani praedicta with a 30cm tongue was discovered in Madagascar. In 2004 the moth was captured on video reaching deep into the nectary of Angraecum sesquipedale.

Regulatory Framework in Canada

Healthcare data in Canada is governed by federal and provincial legislation.

PIPEDA and PHIPA

Healthcare data in Canada is governed federally by the Personal Information Protection and Electronic Documents Act (PIPEDA). Healthcare data is governed in Ontario by the Personal Health Information Protection Act (PHIPA). The PHIPA standard is deemed "substantially similar" to PIPEDA which means that obligations under PIPEDA are met by adherence to PHIPA.

Obligations

This Canadian legislative framework mandates protecting the privacy of citizens' healthcare data; however, the legislation does not create specific compliance standards that would serve as a checklist of audit requirements. It is left to custodians of personal health information (PHI) to apply judgment whether they are taking adequate steps to meet their obligations.

American Standards

In the United States healthcare data is governed by the Healthcare Insurance Portability and Accountability Act (HIPAA) which includes a security rule that defines standards for the protection of health information. AWS publish a list of AWS services (HIPAA-eligible services) that meet the requirements of the security rule for HIPAA-compliant deployments. It is common to use AWS HIPAA-eligible services for applications that processes Canadian healthcare data, though this isn't a formal requirement..

AWS Advanced Consulting Partners like Qalius help healthcare organizations use the AWS services to meet their obligations to protect PHI.

Expert Configuration

AWS services can be used to architect applications suitable for personal health information; however, it is a customer responsibility to use the AWS services in an appropriate way to meet their legislated obligations. AWS call this the Shared Responsibility Model. AWS Advanced Consulting Partners like Qalius help healthcare organizations use the AWS services to meet their obligations to protect PHI.

The AWS website contains additional information about architecting Canadian healthcare solutions on AWS and the Shared Responsibility Model. See the references below for the links.

Build An Orchid

We approach applications that collect, process, or store personal health information (PHI) in an analogous way. Imagine that the PHI is the nectar and legitimate users of the healthcare application are the special moth. We configure the application to act like the orchid. In nature, the shape of the orchid poses a challenge the wrong moths can't overcome. In healthcare applications, the configuration of the application and the underlying AWS services keeps the wrong people from accessing the PHI.

To keep the PHI secure, we need to prevent users from accessing it through unintended paths. Could the bad guys bypass the application and go to where the data is stored? Block that path by ensuring the only path from the data storage service goes directly to the intended healthcare application. Could they hack into the application server to get a username and password to access the data storage service? Block that path by keeping the credentials to the data storage service away from the application server. In these and countless other ways we plan the architecture of the application and the architecture of the AWS services that host the application so they will only serve the PHI in the specific way we intended. This process is like designing a special orchid for a special moth.

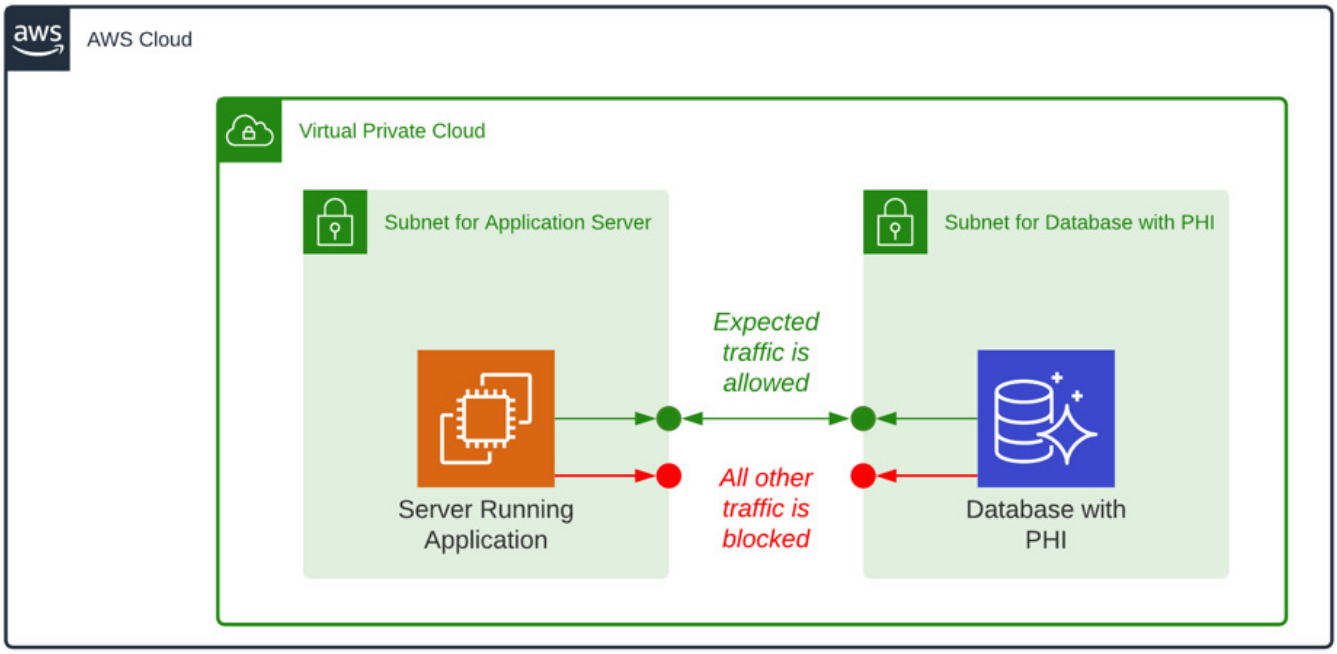
An analogy inspired by Charles Darwin. Darwin understood that the orchid Angraecum sesquipedale co-evolved with a moth with a 30cm tongue (later discovered and named Xanthopan morgani praedicta) that can reach deep into the orchid for nectar. The healthcare application and the underlying AWS services are configured to ensure that only authorized users can access the personal health information.

In the paragraphs that follow we explain a few of the ways we use AWS services to restrict PHI to the intended application. We will refer to the analogy as we go.

Firewalls

In AWS there is a service called Amazon Virtual Private Cloud (VPC) that performs essential functions to keep your data secure. Amazon VPC isolates a small portion of the AWS network and dedicates it to your application. We further subdivide that small portion to create a network environment, called a subnet, for each component of the application. For example, your application might have a server that processes and interprets PHI and a database that stores PHI. The server goes into one subnet, the database goes into another subnet.

The critical feature is that we restrict the traffic between subnets so that only the authorized application traffic patterns are permitted. The subnet containing the database only accepts requests from the server hosting your application. The subnet containing the database is only able to send data to the server hosting your application, and it has no path to the Internet. The result is something like the diagram below.



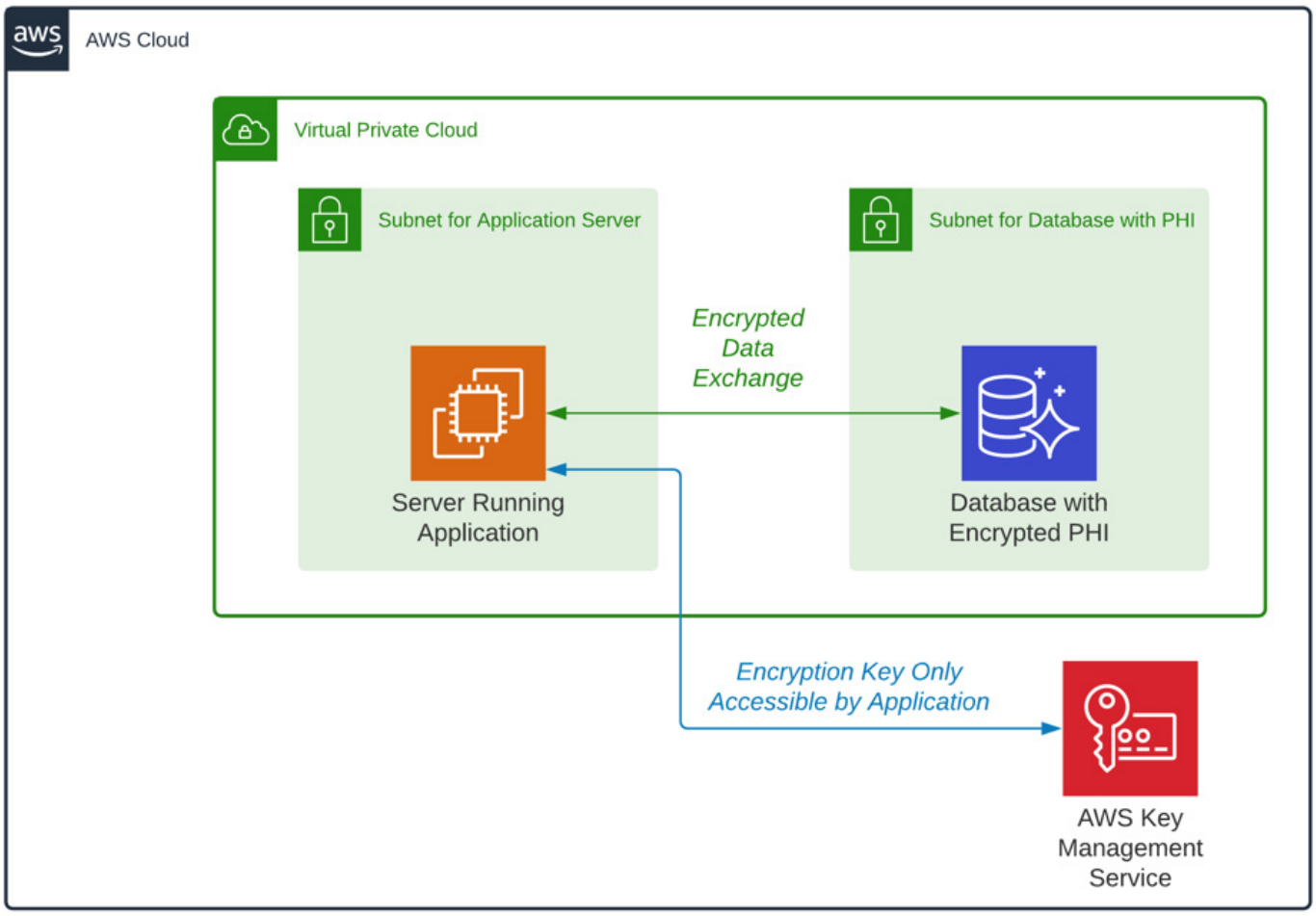
Network Environment: Amazon Virtual Private Cloud (VPC) creates an isolated portion of the AWS network. Within the VPC portions of the network called subnets are defined. The application server and the database with the PHI are installed in the network subnets. The subnets are configured so that only expected application traffic patterns are allowed.

This configuration of Amazon VPC is something like the long nectary on the orchid. To access the PHI (nectar) one needs to follow the special path through the Amazon VPC (the orchid). The networking is configured to restrict access to the intended users via the authorized application (moth).

Encryption

Sensitive data are encrypted using a complex digital key. The key is a string of letters and numbers that are input to a mathematical algorithm to scramble data into an unreadable state. The same key or another (depending on the specific algorithm) is used to unscramble the data. We always encrypt PHI so it is unusable to anybody who doesn't have the key.

There is an AWS service called AWS Key Management Service (KMS) that is a fantastic tool for keeping the key safe. The key is stored in AWS KMS and access to the key is restricted so that only the application running on the correct server can access the key. The application can decrypt the data but if an attacker were to access the database, they would see only encrypted data. The customer team, the Qalius team, and AWS employees are not able to access the key and therefore none of us is able accidentally or maliciously to see the PHI. The result is something like the diagram below.



Encryption Key Management: The application encrypts data before sending it to the database using an encryption key that is managed in the AWS Key Management Service (KMS). AWS KMS restricts access to the key so that only the application running on the expected server can access the key to decrypt the data.

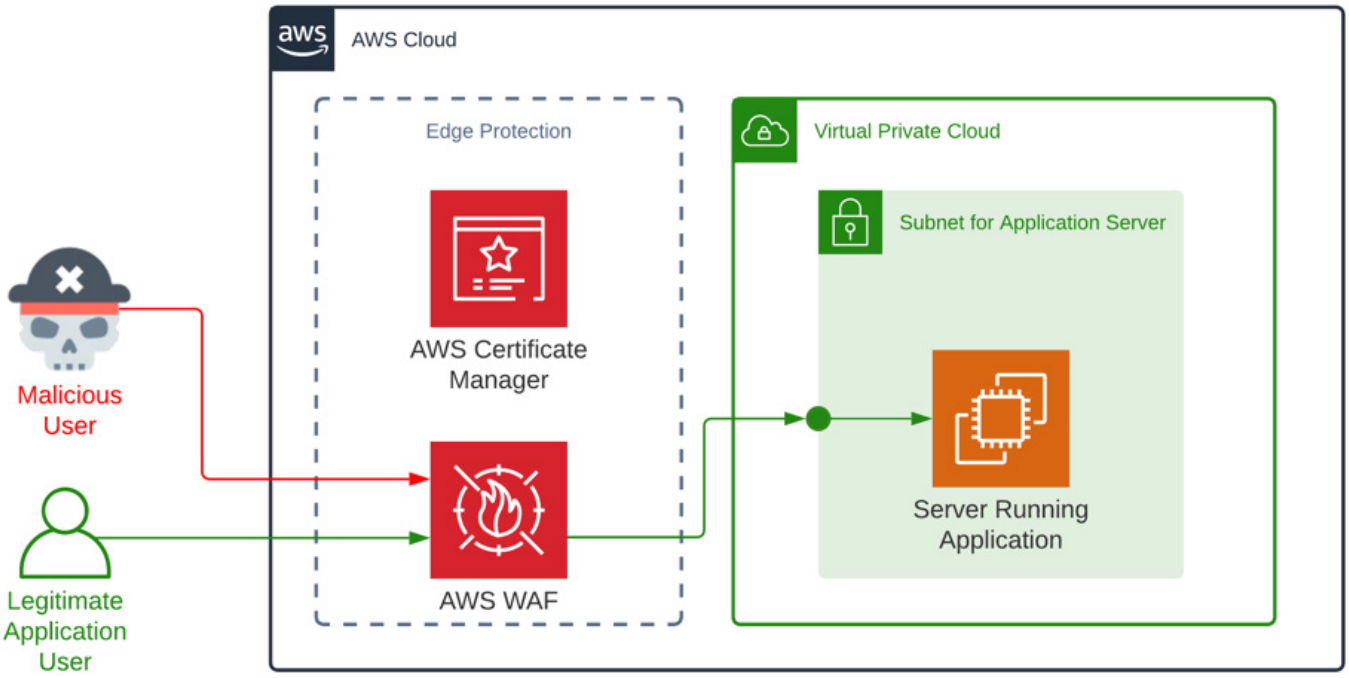
This configuration of AWS KMS is something like the long tongue on the moth. To decrypt the PHI (drink the nectar) one needs to have access to the key (have the right tongue).

Edge Protection

When a browser such as Chrome connects to a secure web application, the browser and the application establish a connection for exchanging encrypted information. The secure connection is established with the aid of a certificate that proves the authenticity of the application. The certificates are renewed to help ensure no imposter applications can pretend to be your application. Additional protection is achieved by reviewing the traffic incoming to the application to reject traffic that appears to be malicious. The traffic should be rejected before it gets to your

application server. Together these are examples of ways to keep malicious actors away from your application and data: protection at the edge of your infrastructure.

A pair of AWS services are a great help with edge protection. We use AWS Certificate Manager to ensure the web application has up-to-date certificates for establishing encrypted connections. We AWS Web Application Firewall (WAF) to identify and block suspicious traffic. AWS WAF is available with rules defined by AWS to protect against common vulnerabilities. Additional rules can be added to suit the specific application use case, or in response to specific malicious activity. The result is something like the diagram below.



Encryption Key Management: The application encrypts data before sending it to the database using an encryption key that is managed in the AWS Key Management Service (KMS). AWS KMS restricts access to the key so that only the application running on the expected server can access the key to decrypt the data.

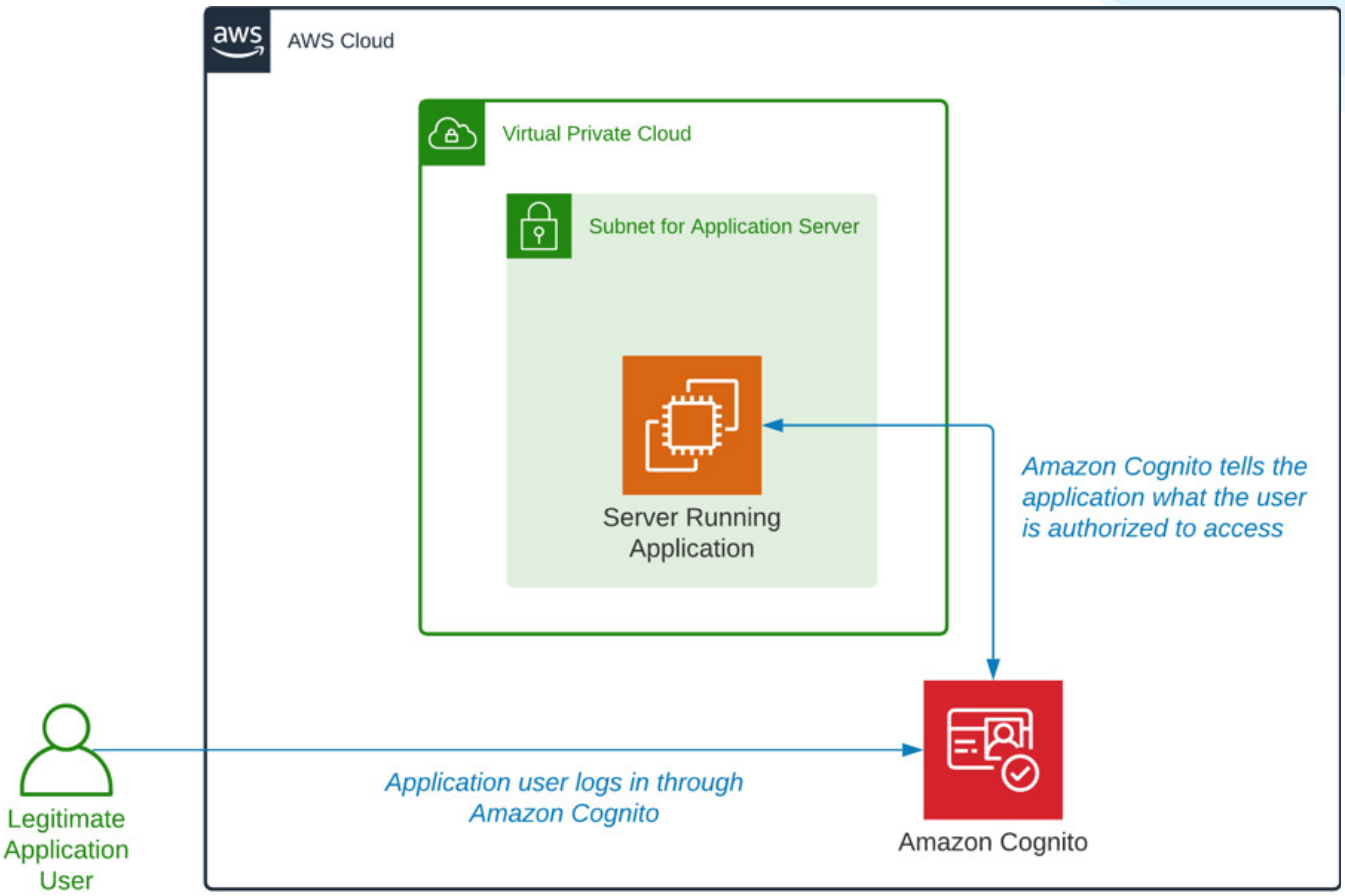
We can think of AWS Certificate Manager and AWS WAF as limiting the ways the application server and the PHI can be accessed. To access the PHI (nectar) one needs to make a secure encrypted connection and send safe traffic (be the moth that can reach into the nectary).

Authentication

No matter how secure the hosting infrastructure, the application needs a robust method to authenticate the application users. Then it needs to authorize them to access application features that are suitable for their role. Without a robust method of authentication and authorization, a legitimate user could accidentally be exposed to private data of other users. Even worse, an attacker who is pretending to be an authorized user could gain access to the PHI.

In AWS there is a service called Amazon Cognito that helps application developers implement robust authentication and authorization. At Qalius we use Amazon Cognito for features like password complexity rules and ready-to-use workflows for password reset. We use Amazon Cognito to enforce multi-factor authentication where the user needs a second way to prove their identity besides their password. Amazon Cognito can detect when a user is using a password

that was compromised on another service and force the user to change their password to keep the application and its data secure. The passwords are stored in the Amazon Cognito service, so they are out of reach of the application developers or an attacker who somehow gained access to the server hosting the application.



Authentication and Authorization: Amazon Cognito manages user authentication with advanced features. Amazon Cognito tells the application what role the user is authorized to take in the application. Security is improved by separating the user passwords from the application.

What Next

The paragraphs above have provided a few examples how the application and the supporting AWS services are designed together so the PHI can only be accessed the way it is intended. The database can only share data with the approved application. Only the approved application can decrypt the data. Suspicious traffic is blocked before it reaches the application. Passwords are managed in a separate standardized service. Every element of the solution is designed to ensure that unauthorized users are unable to see the PHI.

Every element of the solution is designed to ensure that unauthorized users are unable to see the PHI.

No analogy is perfect, but Darwin's orchid and moth give us a way to think about the essential concept. We want our application and the supporting configuration of AWS services to be like the orchid with a long nectary that keeps the nectar safe for the very special moth.

References

Refer to the sources below for more information.

BOOK:
On the various contrivances by which British and foreign orchids are fertilised by insects, and on the good effects of intercrossing

1862

https://www.google.ca/books/edition/On_the_Various_Contrivances_by_Which_Bri/mpMUAAAAYAAJ?hl=en&gbpv=0

VIDEO:
Madagascar Star Orchid and Hawk Moth

March 31, 2017

https://www.youtube.com/watch?v=Q8GdkMBluis

AWS WEBSITE:
Personal Health Information Protection Act (Ontario)

https://aws.amazon.com/compliance/phipa/

AWS WEBSITE:
Shared Responsibility Model

https://aws.amazon.com/compliance/shared-responsibility-model/

About Qalius

Specialists in Web Applications for AWS

At Qalius (sounds like "kale-eee-us") we are specialists in web applications that run on Amazon Web Services (AWS). We build applications to your specifications, and we update the applications you already have to take advantage of high-availability and high-performance architectures on AWS.

Custom Applications

Qalius builds web applications to your requirements in conventional LAMP or cloud-native serverless architectures.

Installation in AWS

Qalius uses infrastructure-as-code techniques to provision AWS services for application performance and data security.

Ongoing Support

Qalius delivers 99.99% application availability by configuring AWS platform automations to monitor and maintain the solution.

Tell Us What You Think

Write to us at wellarchitected@qalius.com with your feedback.

