Infrastructure Security Whitepaper

TABLE OF CONTENTS

Introduction

Outpatient App is a Digital Health Workflow Automation platform currently implemented for Federal Health entities, including Military Health System (MHS) and VA Health. The system focuses on the logistics of the healthcare process, including coordination, and communication to schedule, track, and report on medical steps and operational processes.

Within clinical environments, the platform has proven it can save on the order of 10,000+ man hours per year per site operation. Improving efficiency for Medical Staff and Patients / Employees saves time, saves money, and improves compliance.

Security Approach

Company Infrastructure is based on PCI-DSS v3.2.1 and NIST framework as described in the AWS Best Practices for NIST and PCI-DSS at:

https://docs.aws.amazon.com/config/latest/developerguide/operational-best-practices-for-nist-800-53_rev_5.html

These Infrastructure controls are further combined with controls from AWS Foundational Best practices and CIS AWS Foundations Benchmark. The final control set subsumes SOC 2, HIPAA, and ISO and NIST standards from an AWS infrastructure perspective.

These controls are achieved by leveraging the DuploCloud platform and services. The details of the approach are described in the following white papers:

Deploy Applications 10x Faster with No-Code/Low-Code DevOps

PCI, HIPAA, and HITRUST Compliance with DuploCloud

DuploCloud is a software platform that provides the infrastructure deployment configurations that enable a compliant infrastructure. Further, the DuploCloud operations team builds, maintains, and monitors the infrastructure so that controls are maintained.

Security posture is monitored and maintained via cloud platform services and tools such as AWS Security Hub, AWS Config and AWS Guard Duty.

Outpatient creates and maintains ownership of the cloud account provisioned in the DuploCloud software. This ensures that Outpatient retains sole root access to the cloud account.

Policy Model

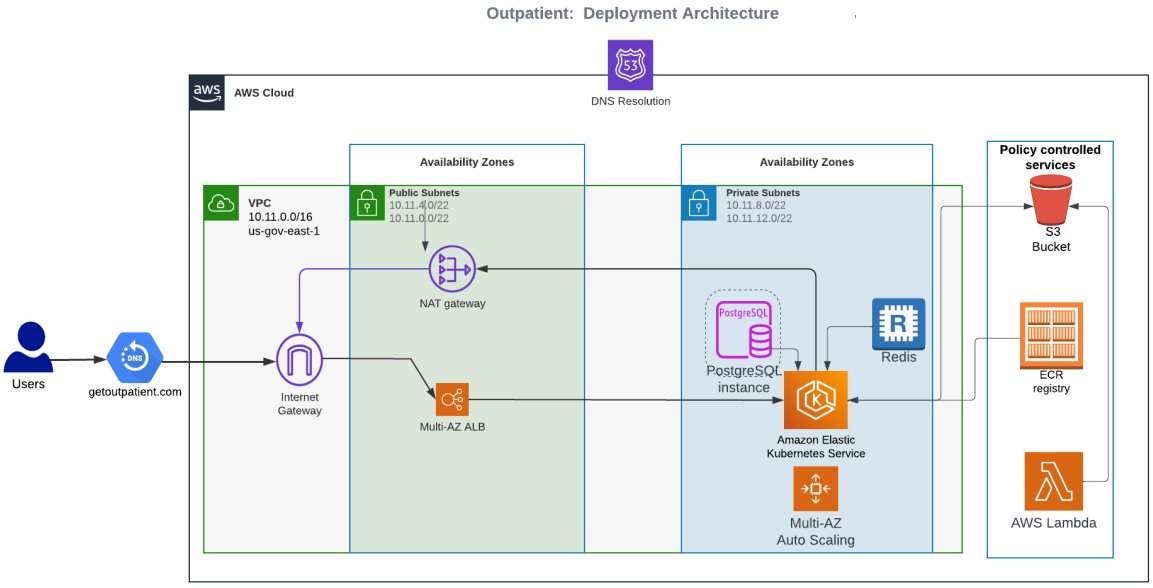
- **Infrastructure.** An infrastructure maps 1:1 with a VPC/VNET and can be in any region. Each infrastructure has a set of subnets spread across multiple availability zones. In AWS GovCloud there is a NAT gateway for private subnets. At Outpatient we have 2 Infrastructures, both in the **us-gov-east-1** Region. One for production and one for non-production.

- **Tenant or Project.** Tenant is the most fundamental construct of the policy model. It represents an application's entire lifecycle. It is:
    - A security boundary i.e., all resources within a tenant have access to each other, but any external access is blocked unless explicitly exposed via an LB, IAM/AD Policy, or SG.
    - A container of resources with each resource implicitly tagged with the tenant name and other labels associated with the tenant. Deleting a tenant deletes all the resources underneath.
    - An access control boundary i.e., each tenant can be accessed by N number of users and each user can access M tenants. The single sign-on access given for a user to a tenant is automatically propagated to provide just-in-time access to the AWS GovCloud resources via the console by the software.
    - Carries all the logs, metrics, and alerts of the application in a single dashboard.
    - Links to the application's code repository for CI/CD, providing a runtime build as a microservice construct such that each tenant can run its own builds in resources in that tenant without worrying about setting up a build system like Jenkins, etc.
    - Part of 1 and only 1 infrastructure. An infrastructure can have multiple tenants.
    - At Outpatient we have a total of 3 tenants. One each for staging, preprod and production.

- **Plan.** This is a logical construct and a container of tenants. It basically has governance policies for the tenants under it. For example, resource usage quota, allowed AMIs, allowed certificates, labels, etc. Each plan can be linked to one and only one infrastructure.
- **User.** This is an individual with a user ID. Each user could have access to one or more tenants/projects.
- **Host.** This is an EC2 instance or VM. This is where your application will run.
- **Service.** Service is where your application code is packaged as a single docker image and running as a set of one or more containers. It is specified as - image-name; replicas; env-variables; vol-mappings, if any. Cloud Automation Platform also allows running applications that are not packaged as Docker images.
- **LB.** A Service can be exposed outside of the tenant\project via an LB and DNS name. LB is defined as - Service name + container-port + External port + Internal-or-internet facing. Optionally, a wild card certificate can be chosen for SSL termination. You can choose to make it internal which will expose it only within your VPC/VNET to other applications.
- **DNS Name.** By default, when a Service is exposed via an LB, Cloud Automation Platform will create a friendly DNS Name. A user can choose to edit this name. The domain name must have been configured in the system by the admin.
- **Docker Host or Fleet Host.** If a host is marked as part of the fleet, then Cloud Automation Platform will use it to deploy containers. If the user needs a host for development purposes such as a test machine, then it would be marked as not part of the pool or fleet.
- **DevOps Controls and SecOps Controls.** We categorize our infrastructure implementation into 2 categories:
    - **DevOps Controls.** These are configurations that are done during provisioning of the infrastructure. Subnets, VPC, Security groups, IAM Roles, Encryption at rest are examples of DevOps Controls.
    - **SecOps Controls.** These are controls that are ongoing. Examples of these are Just-in-time-access, Host Intrusion Detection, CVE etc.

Architecture Design

Following diagram shows the high-level application architecture



DevOps Controls

Network, Security and IAM (Requirement 1)

	PCI DSS Requirements v3.2.1	Cloud Automation Platform Implementation
Requirement 1: Install and maintain a firewall configuration to protect cardholder data		
1.	1.1.4 Requirements for a firewall at each Internet connection and between any demilitarized zone	Infrastructure is split into public and private subnets. Dev, stage, and production are split into different VPCs/VNETs. Cloud Automation Platform introduces a concept of a tenant which is a logical construct above AWS an

	(DMZ) and the Internal network zone	application's entire lifecycle. It is a security boundary implemented by having a unique SG, IAM Role and Instance Profile in a Subnet. By default, no access is allowed into the tenant unless specific ports are exposed via LB.
2.	1.1.5 Description of groups, roles, and responsibilities for management of network components.	Cloud Automation Platform overlays logical constructs of Tenant and infrastructure that represents an application. Within a tenant there are concepts of services. All resources within the tenant are by default labeled in the cloud with the Tenant name. Further the automation allows the user to set any tag at a tenant level and that is automatically propagated to AWS artifacts. The system is always kept in sync with background threads
3.	1.2.1 Restrict inbound and outbound traffic to that which is necessary for the cardholder data environment, and specifically deny all other traffic.	Infrastructure is split into public and private subnets. Dev, stage, and production are split into different VPCs/VNETs. Cloud Automation Platform introduces a concept of a tenant which is a logical construct above AWS and represents an application's entire lifecycle. It is a security boundary implemented by having a unique SG, IAM Role and Instance Profile in a Subnet. By default, no access is allowed into the tenant unless specific ports are exposed via LB.
4.	1.3.1 Implement a DMZ to limit inbound traffic to only system components that provide authorized publicly accessible services, protocols, and ports.	Infrastructure is split into public and private subnets. Dev, stage, and production are split into different VPCs/VNETs. Cloud Automation Platform introduces a concept of a tenant which is a logical construct above AWS and represents an application's entire lifecycle. It is a security boundary implemented by having a unique SG, IAM Role and Instance Profile in a Subnet. By default, no access is allowed into the tenant unless specific ports are exposed via LB.
5.	1.3.2 Limit inbound Internet traffic to IP addresses within the DMZ.	Infrastructure is split into public and private subnets. Dev, stage, and production are split into different VPCs/VNETs. Cloud Automation Platform introduces a concept of a tenant which is a logical construct above AWS and represents an application's entire lifecycle. It is a security boundary implemented by having a unique SG, IAM Role and Instance Profile in a Subnet. By default, no access is allowed into the tenant unless specific ports are exposed via LB.

6.	1.3.4 Do not allow unauthorized outbound traffic from the cardholder data environment to the Internet.	By default, all outbound traffic uses NAT Gateway. We can put in place additional subnet ACLs if needed. Nodes in the private subnets can only go outside only via a NAT Gateway.
7.	1.3.6 Place system components that store cardholder data (such as a database) in an internal network zone, segregated from the DMZ and other untrusted networks.	Infrastructure is split into public and private subnets. Dev, stage, and production are split into different VPCs/VNETs. Cloud Automation Platform introduces a concept of a tenant which is a logical construct above AWS an application's entire lifecycle. It is a security boundary implemented by having a unique SG, IAM Role and Instance Profile in a Subnet. By default, no access is allowed into the tenant unless specific ports are exposed via LB. The application is split into multiple tenants with each tenant having all private resources in a private subnet. An example implementation would be all data stores are in one tenant and frontend UI is in a different tenant
9.	1.5 Ensure that security policies and operational procedures for managing firewalls are documented, in use, and known to all affected parties	Usage of a rules-based approach makes the configuration error free, consistent and documented.

Secret Management (Requirement 2)

	PCI DSS Requirements v3.2.1	Cloud Automation Platform Implementation
Requirement 2: Do not use vendor-supplied defaults for system passwords and other security parameters		
1.	2.1 Always change vendor-supplied defaults and remove or disable unnecessary default accounts before installing a system on the network. This applies to ALL default passwords, including but not limited to those used by operating systems, software that provides security services, application and system accounts, point-of-sale (POS) terminals, Simple Network	Cloud Automation Platform enables user specified password or random password generation options. User access is managed in such a way that all end user access is via single sign on and password less. Even access to AWS console is done by generating a federated console URL that has a validity of less than an hour. The system enables operations with minimal user accounts as most access is JIT

	Management Protocol (SNMP) community strings, etc.)	
2.	2.2.2 Enable only necessary services, protocols, daemons, etc., as required for the function of the system.	By default, no traffic is allowed inside a tenant boundary unless exposed via an LB. Cloud Automation Platform allows automated configuration of desired inter-tenant access w/o users needing to manually write scripts. Further as the env changes dynamically Cloud Automation Platform keys these configs in sync. Cloud Automation Platform also reconciles any orphan resources in the system and cleans them up, this includes docker containers, VMs, LBs, keys, S3 buckets and various other resources
3.	2.2.3 Implement additional security features for any required services, protocols, or daemons that are considered to be insecure. Note: Where SSL/early TLS is used, the requirements in Appendix A2 must be completed.	DC gets certificates from Cert-Manager and automates SSL termination in the LB
4..	2.2.4 Configure system security parameters to prevent misuse.	IAM configuration that implement separation of duties and least privilege, S3 bucket policies. Infrastructure is split into public and private subnets. Dev, stage and production are split into different VPCs/VNETs. Cloud Automation Platform introduces a concept of a tenant which is a logical construct above AWS an application's entire lifecycle. It is a security boundary implemented by having a unique SG, IAM Role and Instance Profile in a Subnet. By default, no access is allowed into the tenant unless specific ports are exposed via LB. The application is split into multiple tenants with each tenant having all private resources in a private subnet. An example implementation would be all data stores are in one tenant and frontend UI is in a different tenant
5.	2.2.5 Remove all unnecessary functionality, such as scripts, drivers, features, subsystems, file systems, and unnecessary web servers.	Cloud Automation Platform reconciles any orphan resources in the system against the user specifications in its database and cleans them up, this includes docker containers, VMs, LBs, keys, S3 buckets and various other resources. All resources specified by the user in the database are tracked and audited every 30 seconds

6.	2.3 Encrypt all non-console administrative access using strong cryptography.	SSL LB and VPN connections are orchestrated. Cloud Automation Platform automates OpenVPN P2S VPN user management by integrating it with user's single sign on i.e., when a user's email is revoked from Cloud Automation Platform portal, it is cleaned up automatically from the VPN server
7..	2.4 Maintain an inventory of system components that are in scope for PCI DSS.	All resources are stored in DB, tracked, and audited. The software has an inventory of resources that can be exported

Encryption and Key Management (Requirement 3)

	PCI DSS Requirements v3.2.1	Cloud Automation Platform Implementation
Requirement 3: Protect stored cardholder data		
1.	3.4.1 If disk encryption is used (rather than file- or column-level database encryption), logical access must be managed separately and independently of native operating system authentication and access control mechanisms (for example, by not using local user account databases or general network login credentials). Decryption keys must not be associated with user accounts. Note: This requirement applies in addition to all other PCI DSS encryption and key-management requirements.	Cloud Automation Platform orchestrates AWS KMSKey Vault keys per tenant to encrypt various AWS resources in that tenant like DBs, S3, Elastic Search, REDIS etc. Access to the keys is granted only to the instance profile w/o any user accounts or keys. By default, Cloud Automation Platform creates a common key per deployment but allows ability to have one key per tenant
2.	3.5.2 Restrict access to cryptographic keys to the fewest number of custodians necessary.	Cloud Automation Platform orchestrates AWS KMS Key Vault keys per tenant to encrypt various AWS resources in that tenant like DBs, S3, Elastic Search, REDIS etc. Access to the keys is granted only to the instance profile w/o any user accounts or keys. By default, Cloud Automation

		Platform creates a common key per deployment but allows ability to have one key per tenant
3.	3.5.3 Store secret and private keys used to encrypt/decrypt cardholder data in one (or more) of the following forms at all times: • Encrypted with a key-encrypting key that is at least as strong as the data encrypting key, and that is stored separately from the data-encrypting key • Within a secure cryptographic device (such as a hardware (host) security module (HSM) or PTS-approved point-of-interaction device) • As at least two full-length key components or key shares, in accordance with an industry accepted method Note: It is not required that public keys be stored in one of these forms.	Cloud Automation Platform orchestrates AWS KMS KeyVault for this and that in turns provides this control that we inherit

Transport Encryption (Requirement 4)

	PCI DSS Requirements v3.2.1	Cloud Automation Platform Implementation
Requirement 4: Encrypt transmission of cardholder data across open, public networks		
1.	4.1 Use strong cryptography and security protocols to safeguard sensitive cardholder data during transmission over open, public networks, including the following: • Only trusted keys and certificates are accepted. • The protocol in use only supports secure versions or configurations. • The encryption strength is appropriate for the encryption methodology in use.	In the secure infrastructure blueprint we adopt Application Load Balancers with HTTPS listeners. HTTP listeners forwarded to HTTPS. The latest cipher is used in the LB automatically by the Cloud Automation Platform software

	Note: Where SSL/early TLS is used, the requirements in Appendix A2 must be completed. Examples of open, public networks include but are not limited to: • The Internet • Wireless technologies, including 802.11 and Bluetooth • Cellular technologies, for example, Global System for Mobile communications (GSM), Code division multiple access (CDMA) • General Packet Radio Service (GPRS) • Satellite communications	

Access Control (Requirements 7 & 8)

	PCI DSS Requirements v3.2.1	Cloud Automation Platform Implementation
Requirement 7: Restrict access to cardholder data by business need to know		
1.	7.1.1 Define access needs for each role, including: • System components and data resources that each role needs to access for their job function • Level of privilege required (for example, user, administrator, etc.) for accessing resources	Cloud Automation Platform tenant model has access controls built in. This allows access to various tenants based on the user roles. This access control mechanism automatically integrates into the VPN client as well i.e. each user has a static IP in the VPN and based on his tenant access his IP is added to the respective tenant's SG in AWS. Tenant access policies will automatically apply SG or IAM based policy in NSG.
	7.2 Establish an access control system(s) for systems components that restricts access based on a user's need to know and is set to “deny all” unless specifically allowed. This access control system(s) must include the following	User access to AWS console is granted based on tenant permissions and least privilege and a Just in time federated token that expires in less than an hour. Admins have privileged access and read-only user is another role

2.	7.2.1 Coverage of all system components.	AWS resource access is controlled based on IAM role, SG and static VPN client IPs
3.	7.2.3 Default deny-all setting.	This is the default Cloud Automation Platform implementation of Sg and IAM roles in NSG and Managed Identity in Azure
Requirement 8: Identify and authenticate access to system components		
1.	8.1.1 Assign all users a unique ID before allowing them to access system components or cardholder data.	Cloud Automation Platform integrates with our IDP like G Suite and O365 for access to the portal. From there a federated logic in done for AWS resource access
2.	8.1.2 Control addition, deletion, and modification of user IDs, credentials, and other identifier objects.	This is done at infra level in Cloud Automation Platform portal using single sign on
3.	8.1.3 Immediately revoke access for any terminated users.	Cloud Automation Platform integrates with our IDP like G Suite and O365 for access to the portal. The moment the email is disabled all access is revoked. Even if the user has say a private key to a VM even then he cannot connect because VPN will be deprovisioned
4.	8.1.4 Remove/disable inactive user accounts within 90 days.	Cloud Automation Platform integrates with our IDP like G Suite and O365 for access to the portal. The moment the email is disabled all access is revoked. Even if the user has say a private key to a VM even then he cannot connect because VPN will be deprovisioned
5.	8.1.5 Manage IDs used by third parties to access, support, or maintain system components via remote access as follows: • Enabled only during the time period needed and disabled when not in use. • Monitored when in use.	Cloud Automation Platform integrates by calling STS API to provide JIT token and URL
6.	8.1.6 Limit repeated access attempts by locking out the user ID after not more than six attempts.	Cloud Automation Platform integrates with our IDP like G Suite and O365 for access to the portal. When Cloud Automation Platform managed OpenVPN is used it is setup to lock the user out after failed attempts

7.	8.1.7 Set the lockout duration to a minimum of 30 minutes or until an administrator enables the user ID.	Cloud Automation Platform integrates with our IDP like G Suite and O365 for access to the portal. When Cloud Automation Platform managed OpenVPN is used it is setup to lock the user out after failed attempts. In OpenVPN an admin has to unlock the user
8.	8.1.8 If a session has been idle for more than 15 minutes, require the user to re-authenticate to re-activate the terminal or session.	Cloud Automation Platform single sign on has configurable timeout. For AWS resource access we provide JIT access
9.	8.2 In addition to assigning a unique ID, ensure proper user-authentication management for non-consumer users and administrators on all system components by employing at least one of the following methods to authenticate all users: • Something you know, such as a password or passphrase • Something you have, such as a token device or smart card • Something you are, such as a biometric.	Cloud Automation Platform relies on the single sign on / IDP. If the user secures his corporate login using these controls then by virtue of single sign on, this gets implemented in the infrastructure.
10.	8.2.1 Using strong cryptography, render all authentication credentials (such as passwords/phrases) unreadable during transmission and storage on all system components.	Encryption at REST is done via AWS KMSKeyVault and in transit via SSL
11.	8.2.2 Verify user identity before modifying any authentication credential—for example, performing password resets, provisioning new tokens, or generating new keys.	Cloud Automation Platform integrates with our IDP like G Suite and O365 for access to the portal.
12.	8.2.3 Passwords/phrases must meet the following: • Require a minimum length of at least seven characters. • Contain both numeric and alphabetic characters.	Enforced by AWSshould be enforced by our IDP. Cloud Automation Platform integrates with the IDP. The control should be implemented by the organization IDP.

	Alternatively, the passwords/phrases must have complexity and strength at least equivalent to the parameters specified above.	
13.	8.2.4 Change user passwords/passphrases at least every 90 days.	Cloud Automation Platform integrates with our IDP like G Suite and O365 for access to the portal.
14.	8.2.5 Do not allow an individual to submit a new password/phrase that is the same as any of the last four passwords/phrases he or she has used.	Enforced by AWSshould be enforced by our IDP. Cloud Automation Platform integrates with the IDP. The control should be implemented by the organization IDP.
15.	8.2.6 Set passwords/phrases for first time use and upon reset to a unique value for each user and change immediately after the first use.	Enforced by AWSshould be enforced by our IDP. Cloud Automation Platform integrates with the IDP. The control should be implemented by the organization IDP.
16.	8.3 Secure all individual non-console administrative access and all remote access to the CDE using multi-factor authentication. Note: Multi-factor authentication requires that a minimum of two of the three authentication methods (see Requirement 8.2 for descriptions of authentication methods) be used for authentication. Using one factor twice (for example, using two separate passwords) is not considered multi-factor authentication.	Cloud Automation Platform integrates with our IDP like G Suite and O365 for access to the portal. Open VPN has MFA enabled
17.	8.3.1 Incorporate multi-factor authentication for all non-console access into the CDE for personnel with administrative access.	Cloud Automation Platform integrates with our IDP like G Suite and O365 for access to the portal. Open VPN has MFA enabled
18.	8.3.2 Incorporate multi-factor authentication for all remote network access (both user and administrator and including third-	Cloud Automation Platform integrates with our IDP like G Suite and O365 for access to the portal. Open VPN has MFA enabled

	party access for support or maintenance) originating from outside the entity's network.	
19..	8.7 All access to any database containing cardholder data (including access by applications, administrators, and all other users) is restricted as follows: • All user access to, user queries of, and user actions on databases are through programmatic methods. • Only database administrators can directly access or query databases. • Application IDs for database applications can only be used by the applications (and not by individual users or other non-application processes)	The IAM integration with the database makes SQL connections also via Instance Profile. For users, individual JIT access is granted that lasts only 15 mins

SecOps Controls

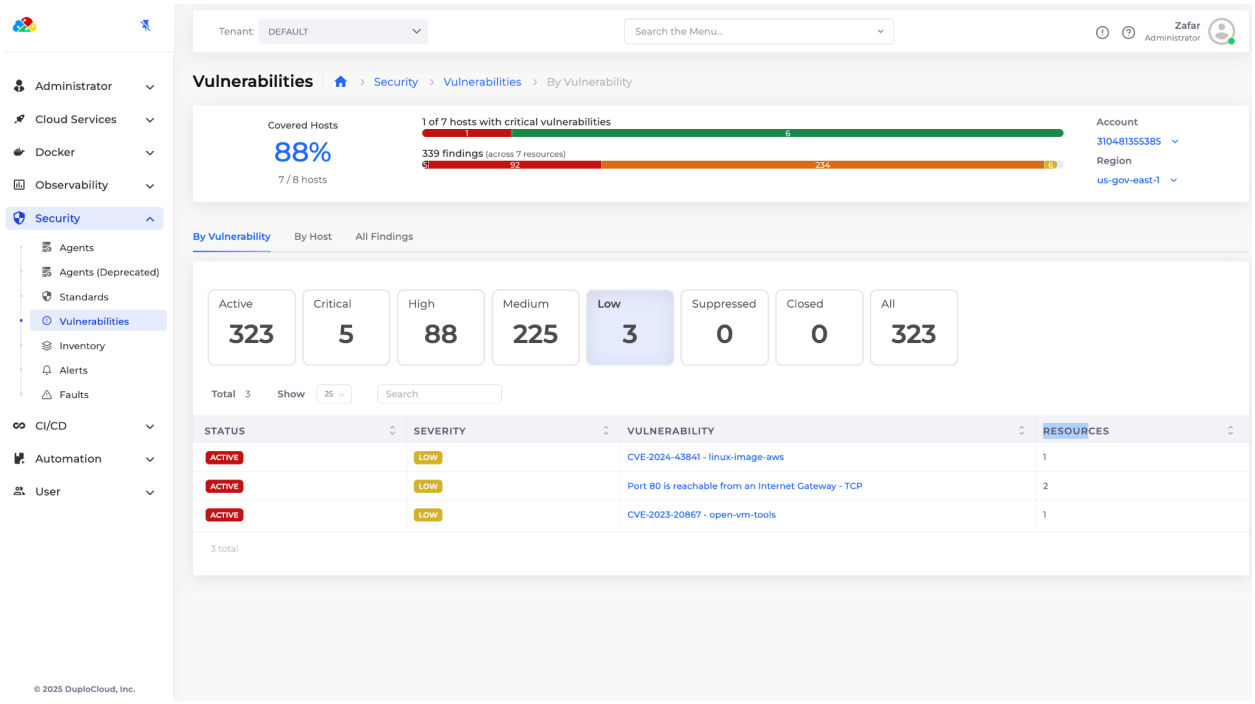
Vulnerability Detection (Requirement 6.1)

AWS Inspector, is a security assessment service that helps improve the security and compliance of applications deployed on AWS. It automatically assesses applications for exposure, vulnerabilities, and deviations from best practices. Here's how AWS Inspector aligns with PCI DSS Requirement 6.1:

1. Automated Security Assessments: AWS Inspector performs automated security assessments of your applications and infrastructure, identifying potential security vulnerabilities.
2. Vulnerability Management: It can detect vulnerabilities in your operating systems, network configurations, and installed applications. This helps in maintaining an updated inventory of vulnerabilities, a key aspect of Requirement 6.1.
3. Risk Ranking: AWS Inspector provides detailed findings with severity levels. These findings can be used to prioritize the remediation of vulnerabilities based on their

risk ranking, aligning with the need to assign a risk ranking to newly discovered vulnerabilities.

4. Continuous Monitoring: By integrating AWS Inspector with other AWS services such as AWS Config and AWS CloudWatch, you can set up continuous monitoring and automatic re-assessment of your resources, ensuring that new vulnerabilities are detected and addressed in a timely manner.



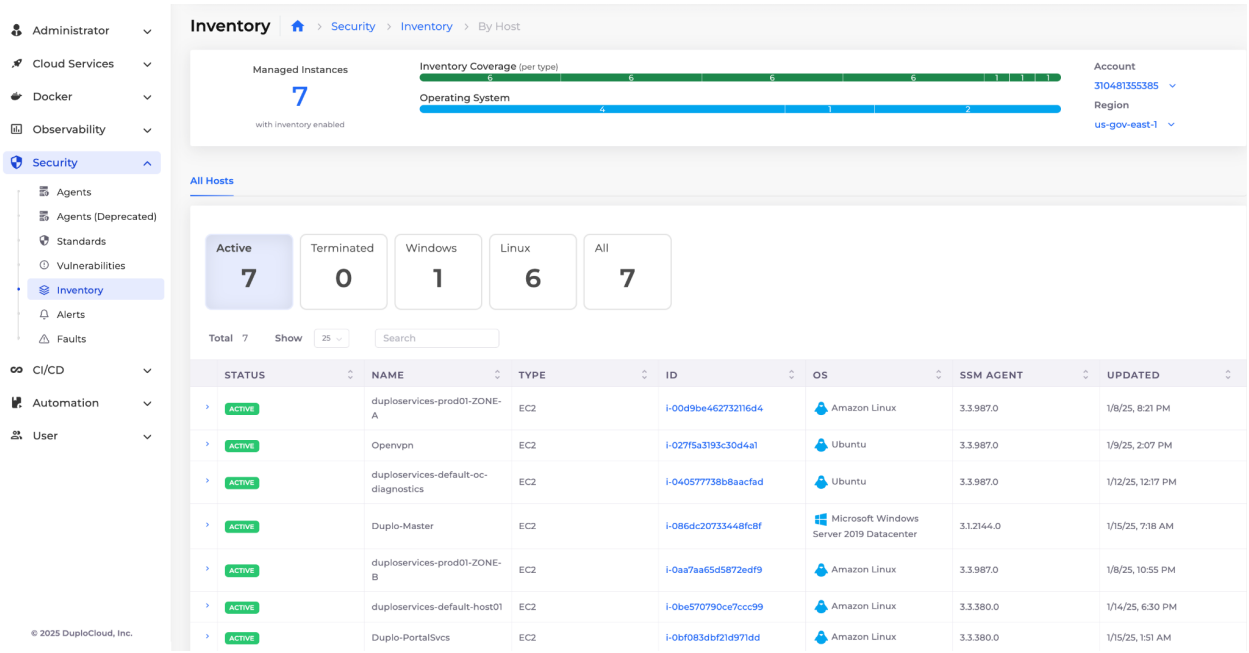
CIS Benchmarks (Requirement 1)

AWS Config is a service that enables you to assess, audit, and evaluate the configurations of your AWS resources. It helps you ensure that your resource configurations comply with best practices and regulatory requirements such as the CIS (Center for Internet Security) Benchmarks.

Here's how AWS Config can be used to meet CIS Benchmarks and thus contribute to PCI Requirement 1:

1. Configuration Recording:
    - AWS Config records the configurations of your AWS resources, enabling you to maintain an inventory of your resource configurations and security settings.

2. CIS Benchmark Rules:
    - AWS Config provides managed rules that map directly to the CIS Benchmarks. These rules automatically evaluate whether your AWS resources comply with the specific security configurations recommended by the CIS.
3. Custom Rules:
    - You can create custom AWS Config rules to address specific security policies or additional controls that are not covered by the managed rules.
4. Continuous Monitoring:
    - AWS Config continuously monitors and records your AWS resource configurations and evaluates them against the CIS Benchmarks. This continuous assessment helps ensure that any deviations from the benchmarks are detected promptly.
5. Compliance Reporting:
    - AWS Config provides compliance reporting that helps you understand the compliance status of your resources in relation to the CIS Benchmarks. This reporting can be used as evidence for compliance audits.

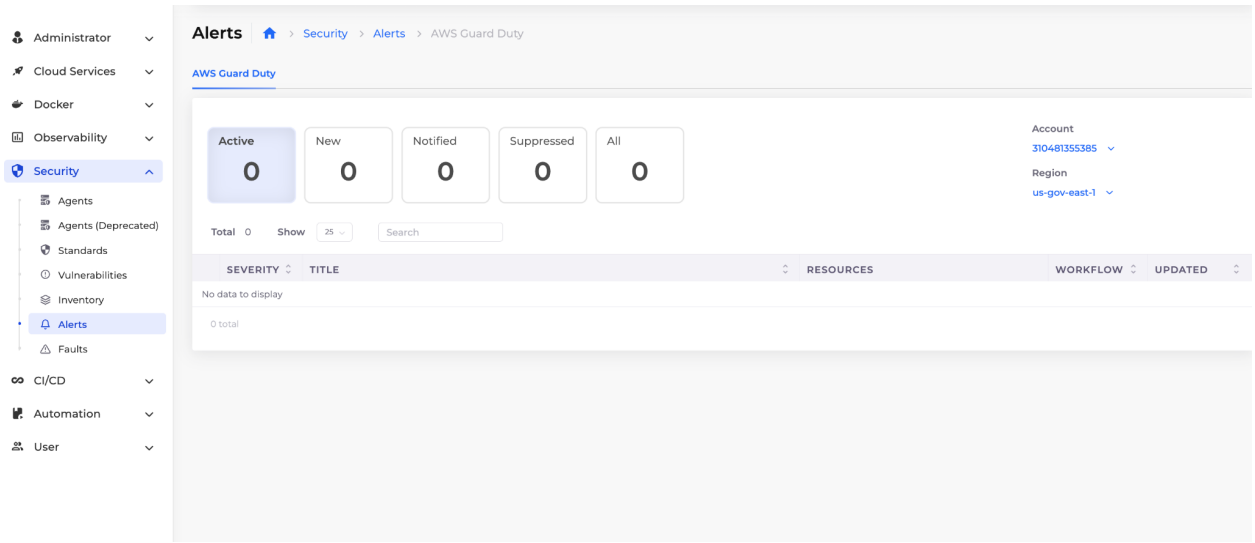


Cloud Vulnerabilities & Intrusion Detection (Requirement 11.4)

Cloud Vulnerabilities are detected with the help of AWS Inspector.

Amazon GuardDuty is a threat detection service that continuously monitors for malicious activity and unauthorized behavior to protect your AWS accounts and workloads. It uses machine learning, anomaly detection, and integrated threat intelligence to identify and prioritize potential threats.

- Threat Detection: GuardDuty analyzes continuous streams of metadata generated from your AWS environment, including VPC Flow Logs, AWS CloudTrail event logs, and DNS logs, to detect threats.
- Actionable Alerts: It generates detailed security findings that include information on affected resources and recommended remediation steps.
- Ease of Use: GuardDuty is easy to enable and doesn't require deploying or managing additional infrastructure.



Network Intrusion Detection (Requirement 11.4)

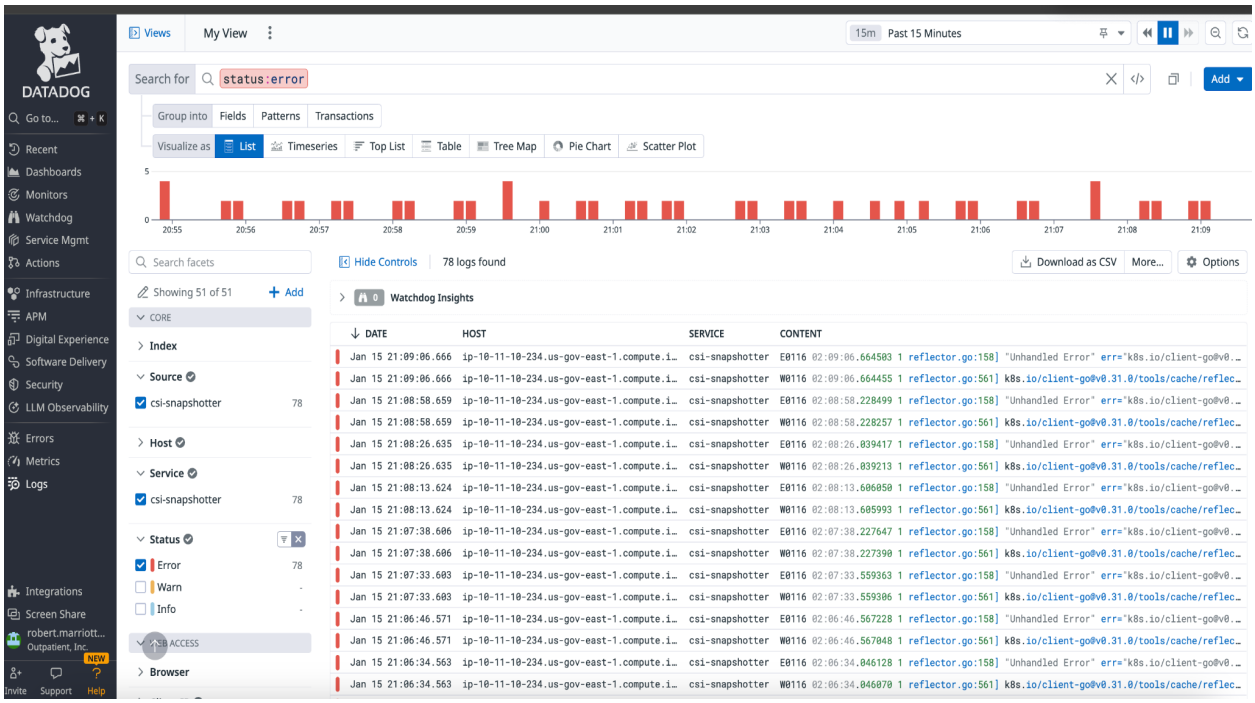
Automation-Platform uses AWS Guard duty for NIDS. Please see details of Guard Duty as NIDS in this AWS document

https://aws.amazon.com/blogs/security/new-third-party-test-compares-amazon-guardduty-to-network-intrusion-detection-systems

Alert Notifications

Automation-Platform extends integration with DataDog, all faults could be alerted real time to the appropriate on-call process.

Logging in Datadog



Incident Management

All Incidents are captured as Issues in GitHub, assigned and owner and all progress is captured in the tickets till resolution and closure.

Control-by-Control PCI Implementation Detail

	PCI DSS Requirements v3.2.1	Cloud Automation Platform Implementation
1.	1.1.4 Requirements for a firewall at each Internet connection and between any demilitarized zone (DMZ) and the Internal network zone	Infrastructure is split into public and private subnets. Dev, stage, and production are split into different VPCs/VNETs. Cloud Automation Platform introduces a concept of a tenant which is a logical construct above AWS an application's entire lifecycle. It is a security boundary implemented by having a unique SG, IAM Role and Instance Profile in Subnet. By default, no access is allowed into the tenant unless specific ports are exposed via LB.
2.	1.1.5 Description of groups, roles, and responsibilities for management of network components.	Cloud Automation Platform overlays logical constructs of Tenant and infrastructure that represents an application. Within a tenant there are concepts of services. All resources within the tenant are by default labeled in the cloud with the Tenant name. Further the automation allows the user to set any tag at a tenant level and that is automatically propagated to AWS artifacts. The system is always kept in sync with background threads
3.	1.2.1 Restrict inbound and outbound traffic to that which is necessary for the cardholder data environment, and specifically deny all other traffic.	Infrastructure is split into public and private subnets. Dev, stage and production are split into different VPCs/VNETs. Cloud Automation Platform introduces a concept of a tenant which is a logical construct above AWS an application's entire lifecycle. It is a security boundary implemented by having a unique SG, IAM Role and Instance Profile in Subnet. By default, no access is allowed into the tenant unless specific ports are exposed via LB.
4.	1.3.1 Implement a DMZ to limit inbound traffic to only system components that provide authorized publicly accessible services, protocols, and ports.	Infrastructure is split into public and private subnets. Dev, stage, and production are split into different VPCs/VNETs. Cloud Automation Platform introduces a concept of a tenant which is a logical construct above AWS an application's entire lifecycle. It is a security boundary implemented by having a unique SG, IAM Role and

		Instance Profile in Subnet. By default, no access is allowed into the tenant unless specific ports are exposed via ELB.
5.	1.3.2 Limit inbound Internet traffic to IP addresses within the DMZ.	Infrastructure is split into public and private subnets. Dev, stage and production are split into different VPCs/VNETs. Cloud Automation Platform introduces a concept of a tenant which is a logical construct above AWS an application's entire lifecycle. It is a security boundary implemented by having a unique SG, IAM Role and Instance Profile in Subnet. By default, no access is allowed into the tenant unless specific ports are exposed via ELB.
6.	1.3.4 Do not allow unauthorized outbound traffic from the cardholder data environment to the Internet.	By default, all outbound traffic uses NAT Gateway. We can put in place additional subnet ACLs if needed. Nodes in the private subnets can only go outside only via a NAT Gateway.
7.	1.3.6 Place system components that store cardholder data (such as a database) in an internal network zone, segregated from the DMZ and other untrusted networks.	Infrastructure is split into public and private subnets. Dev, stage, and production are split into different VPCs/VNETs. Cloud Automation Platform introduces a concept of a tenant which is a logical construct above AWS an application's entire lifecycle. It is a security boundary implemented by having a unique SG, IAM Role and Instance Profile in a Subnet. By default, no access is allowed into the tenant unless specific ports are exposed via LB. The application is split into multiple tenants with each tenant having all private resources in a private subnet. An example implementation would be all data stores are in one tenant and frontend UI is in a different tenant
8.	1.3.7 Do not disclose private IP addresses and routing information to unauthorized parties.	Use Private subnets and private R53 hosted zones
9.	1.5 Ensure that security policies and operational procedures for managing firewalls are documented, in use, and known to all affected parties	Usage of a rules-based approach makes the configuration error free, consistent and documented.
10.	2.1 Always change vendor-supplied defaults and remove or disable unnecessary default accounts before installing a system on the network.	Cloud Automation Platform enables user specified password or random password generation options. User access is managed in such a way that all end user access is via single sign on and password less. Even access to AWSconsole is done by generating a federated console

	This applies to ALL default passwords, including but not limited to those used by operating systems, software that provides security services, application and system accounts, point-of-sale (POS) terminals, Simple Network Management Protocol (SNMP) community strings, etc.)	URL that has a validity of less than an hour. The system enables operations with minimal user accounts as most access is JIT
12.	2.2.2 Enable only necessary services, protocols, daemons, etc., as required for the function of the system.	By default, no traffic is allowed inside a tenant boundary unless exposed via an LB. Cloud Automation Platform allows automated configuration of desired inter-tenant access w/o users needing to manually write scripts. Further as the env changes dynamically Cloud Automation Platform keys these configs in sync. Cloud Automation Platform also reconciles any orphan resources in the system and cleans them up, this includes docker containers, VMs, LBs, keys, S3 buckets and various other resources
13.	2.2.3 Implement additional security features for any required services, protocols, or daemons that are insecure. Note: Where SSL/early TLS is used, the requirements in Appendix A2 must be completed.	DC gets certificates from Cert-Manager and automates SSL termination in the LB
14.	2.2.4 Configure system security parameters to prevent misuse.	IAM configuration and policies that implement separation of duties and least privilege, S3 bucket policies. Infrastructure is split into public and private subnets. Dev, stage, and production are split into different VPCs/VNETs. Cloud Automation Platform introduces a concept of a tenant which is a logical construct above AWS an application's entire lifecycle. It is a security boundary implemented by having a unique SG, IAM Role and Instance Profile in a Subnet. By default, no access is allowed into the tenant unless specific ports are exposed via LB. The application is split into multiple tenants with each tenant having all private resources in a private subnet. An example implementation would be all data stores are in one tenant and frontend UI is in a different tenant

15.	2.2.5 Remove all unnecessary functionality, such as scripts, drivers, features, subsystems, file systems, and unnecessary web servers.	Cloud Automation Platform reconciles any orphan resources in the system against the user specifications in its database and cleans them up, this includes docker containers, VMs, LBs, keys, S3 buckets and various other resources. All resources specified by the user in the database are tracked and audited every 30 seconds
16.	2.3 Encrypt all non-console administrative access using strong cryptography.	SSL LB and VPN connections are orchestrated. Cloud Automation Platform automates OpenVPN P2S VPN user management by integrating it with user's single sign on i.e., when a user's email is revoked from Cloud Automation Platform portal, it is cleaned up automatically from the VPN server
17.	2.4 Maintain an inventory of system components that are in scope for PCI DSS.	All resources are stored in DB, tracked, and audited. The software has an inventory of resources that can be exported
Requirement 3: Protect stored cardholder data		
18.	3.4.1 If disk encryption is used (rather than file- or column-level database encryption), logical access must be managed separately and independently of native operating system authentication and access control mechanisms (for example, by not using local user account databases or general network login credentials). Decryption keys must not be associated with user accounts. Note: This requirement applies in addition to all other PCI DSS encryption and key-management requirements.	Cloud Automation Platform orchestrates AWS KMSKey Vault keys per tenant to encrypt various AWS resources in that tenant like DBs, S3, Elastic Search, REDIS etc. Access to the keys is granted only to the instance profile w/o any user accounts or keys. By default, Cloud Automation Platform creates a common key per deployment but allows ability to have one key per tenant
19.	3.5.2 Restrict access to cryptographic keys to the fewest number of custodians necessary.	Cloud Automation Platform orchestrates AWS KMSKey Vault keys per tenant to encrypt various AWS resources in that tenant like DBs, S3, Elastic Search, REDIS etc. Access to the keys is granted only to the instance profile w/o any user accounts or keys. By default, Cloud Automation Platform creates a common key per deployment but allows ability to have one key per tenant

20.	3.5.3 Store secret and private keys used to encrypt/decrypt cardholder data in one (or more) of the following forms at all times: • Encrypted with a key-encrypting key that is at least as strong as the data encrypting key, and that is stored separately from the data-encrypting key • Within a secure cryptographic device (such as a hardware (host) security module (HSM) or PTS-approved point-of-interaction device) • As at least two full-length key components or key shares, in accordance with an industry accepted method Note: It is not required that public keys be stored in one of these forms.	Cloud Automation Platform orchestrates AWS KMSKey Vault for this and that in turns provides this control that we inherit
Requirement 4: Encrypt transmission of cardholder data across open, public networks		
21.	4.1 Use strong cryptography and security protocols to safeguard sensitive cardholder data during transmission over open, public networks, including the following: • Only trusted keys and certificates are accepted. • The protocol in use only supports secure versions or configurations. • The encryption strength is appropriate for the encryption methodology in use. Note: Where SSL/early TLS is used, the requirements in Appendix A2 must be completed. Examples of open, public networks include but are not limited to: • The Internet • Wireless technologies, including 802.11 and Bluetooth	In the secure infrastructure blueprint we adopt Application Load Balancers with HTTPS listeners. HTTP listeners forwarded to HTTPS. The latest cipher is used in the LB automatically by the Cloud Automation Platform software

	• Cellular technologies, for example, Global System for Mobile communications (GSM), Code division multiple access (CDMA) • General Packet Radio Service (GPRS) • Satellite communications	
Requirement 5: Protect all systems against malware and regularly update anti-virus software or programs		
23.	5.1.2 For systems considered to be not commonly affected by malicious software, perform periodic evaluations to identify and evaluate evolving malware threats in order to confirm whether such systems continue to not require anti-virus software.	Cloud Automation Platform agent modules can be enabled
25	5.3 Ensure that anti-virus mechanisms are actively running and cannot be disabled or altered by users, unless specifically authorized by management on a case-by-case basis for a limited time period. Note: Anti-virus solutions may be temporarily disabled only if there is legitimate technical need, as authorized by management on a case-by-case basis. If anti-virus protection needs to be disabled for a specific purpose, it must be formally authorized. Additional security measures may also need to be implemented for the period of time during which anti-virus protection is not active.	Cloud Automation Platform agent modules do thousand raise an alert if a service is not running
Requirement 6: Develop and maintain secure systems and applications		
26.	6.3.2 Review custom code prior to release to production or customers in order to identify any potential coding vulnerability (using either	Cloud Automation Platform's CI/CD offering provides an out-of-box integration with SonarQube that can be integrated into the pipeline to scan the code.

	manual or automated processes) to include at least the following: • Code changes are reviewed by individuals other than the originating code author, and by individuals knowledgeable about code-review techniques and secure coding practices. • Code reviews ensure code is developed according to secure coding guidelines • Appropriate corrections are implemented prior to release. • Code-review results are reviewed and approved by management prior to release Note: This requirement for code reviews applies to all custom code (both internal and public-facing), as part of the system development life cycle. Code reviews can be conducted by knowledgeable internal personnel or third parties. Public-facing web applications are also subject to additional controls, to address ongoing threats and vulnerabilities after implementation, as defined at PCI DSS Requirement 6.6.	
Requirement 7: Restrict access to cardholder data by business need to know		
27.	7.1.1 Define access needs for each role, including: • System components and data resources that each role needs to access for their job function • Level of privilege required (for example, user, administrator, etc.) for accessing resources	Cloud Automation Platform tenant model has access controls built in. This allows access to various tenant based on the user roles. This access control mechanism automatically integrates into the VPN client as well i.e. each user has a static IP in the VPN and based on his tenant access his IP is added to the respective tenant's SG in NSG in Azure. Tenant access policies will automatically apply SG or IAM based policy in NSG.
28.	7.2 Establish an access control system(s) for system components that restricts access based on a user's need to know and is set to	User access to AWSconsole is granted based on tenant permissions and least privilege and a Just in time federated token that expires in less than an hour. Admins have privileged access and read-only user is another role

	"deny all" unless specifically allowed. This access control system(s) must include the following	
29.	7.2.1 Coverage of all system components.	AWS resource access is controlled based on IAM role, SG and static VPN client Ips.
30.	7.2.3 Default deny-all setting.	This is the default Cloud Automation Platform implementation of Sg and IAM roles in NSG and Managed Identity in Azure
Requirement 8: Identify and authenticate access to system components		
31.	8.1.1 Assign all users a unique ID before allowing them to access system components or cardholder data.	Cloud Automation Platform integrates with our IDP like G Suite and O365 for access to the portal. From there a federated logic in done for AWS resource access
32.	8.1.2 Control addition, deletion, and modification of user IDs, credentials, and other identifier objects.	This is done at infra level in Cloud Automation Platform portal using single sign on
33.	8.1.3 Immediately revoke access for any terminated users.	Cloud Automation Platform integrates with our IDP like G Suite and O365 for access to the portal. The moment the email is disabled all access is revoked. Even if the user has say a private key to a VM even then he cannot connect because VPN will be deprovisioned
34.	8.1.4 Remove/disable inactive user accounts within 90 days.	Cloud Automation Platform integrates with our IDP like G Suite and O365 for access to the portal. The moment the email is disabled all access is revoke. Even if the user has say a private key to a VM even then he cannot connect because VPN will be deprovisioned
35.	8.1.5 Manage IDs used by third parties to access, support, or maintain system components via remote access as follows: • Enabled only during the time period needed and disabled when not in use. • Monitored when in use.	Cloud Automation Platform integrates by calling STS API to provide JIT token and URL

36.	8.1.6 Limit repeated access attempts by locking out the user ID after not more than six attempts.	Cloud Automation Platform integrates with our IDP like G Suite and O365 for access to the portal. When Cloud Automation Platform managed OpenVPN is used it is setup to lock the user out after failed attempts
37.	8.1.7 Set the lockout duration to a minimum of 30 minutes or until an administrator enables the user ID.	Cloud Automation Platform integrates with our IDP like G Suite and O365 for access to the portal. When Cloud Automation Platform managed OpenVPN is used it is setup to lock the user out after failed attempts. In Open VPN an admin must unlock the user
38.	8.1.8 If a session has been idle for more than 15 minutes, require the user to re-authenticate to re-activate the terminal or session.	Cloud Automation Platform single sign on has configurable timeout. For AWS resource access we provide JIT access
39.	8.2 In addition to assigning a unique ID, ensure proper user-authentication management for non-consumer users and administrators on all system components by employing at least one of the following methods to authenticate all users: • Something you know, such as a password or passphrase • Something you have, such as a token device or smart card • Something you are, such as a biometric.	Cloud Automation Platform relies on our single sign on / IDP. If the user secures his corporate login using these controls, then by virtue of single sign on, this get implemented in the infrastructure.
40.	8.2.1 Using strong cryptography, render all authentication credentials (such as passwords/phrases) unreadable during transmission and storage on all system components.	Encryption at REST is done via AWS KMSKeyVault and in transit via SSL
41.	8.2.2 Verify user identity before modifying any authentication credential—for example, performing password resets, provisioning new tokens, or generating new keys.	Cloud Automation Platform integrates with our IDP like G Suite and O365 for access to the portal.

42.	8.2.3 Passwords/phrases must meet the following: • Require a minimum length of at least seven characters. • Contain both numeric and alphabetic characters. Alternatively, the passwords/phrases must have complexity and strength at least equivalent to the parameters specified above.	Enforced by AWS through our IDP. Cloud Automation Platform integrates with the IDP. The control should be implemented by the organization IDP.
43.	8.2.4 Change user passwords/passphrases at least every 90 days.	Cloud Automation Platform integrates with our IDP like G Suite and O365 for access to the portal.
44.	8.2.5 Do not allow an individual to submit a new password/phrase that is the same as any of the last four passwords/phrases he or she has used.	Enforced by AWS through our IDP. Cloud Automation Platform integrates with the IDP. The control should be implemented by the organization IDP.
45.	8.2.6 Set passwords/phrases for first time use and upon reset to a unique value for each user and change immediately after the first use.	Enforced by AWS through our IDP. Cloud Automation Platform integrates with the IDP. The control should be implemented by the organization IDP,
46.	8.3 Secure all individual non-console administrative access and all remote access to the CDE using multi-factor authentication. Note: Multi-factor authentication requires that a minimum of two of the three authentication methods (see Requirement 8.2 for descriptions of authentication methods) be used for authentication. Using one factor twice (for example, using two separate passwords) is not considered multi-factor authentication.	Cloud Automation Platform integrates with our IDP like G Suite and O365 for access to the portal. Open VPN has MFA enabled
47.	8.3.1 Incorporate multi-factor authentication for all non-console	Cloud Automation Platform integrates with our IDP like G Suite and O365 for access to the portal. Open VPN has MFA enabled

	access into the CDE for personnel with administrative access.	
48.	8.3.2 Incorporate multi-factor authentication for all remote network access (both user and administrator and including third-party access for support or maintenance) originating from outside the entity's network.	Cloud Automation Platform integrates with our IDP like G Suite and O365 for access to the portal. Open VPN has MFA enabled
49.	8.7 All access to any database containing cardholder data (including access by applications, administrators, and all other users) is restricted as follows: • All user access to, user queries of, and user actions on databases are through programmatic methods. • Only database administrators have the ability to directly access or query databases. • Application IDs for database applications can only be used by the applications (and not by individual users or other non-application processes)	The IAM integration with database makes SQL connections also via Instance Profile. For users, individual JIT access is granted that lasts only 15 mins
Requirement 10: Track and monitor all access to network resources and cardholder data		
50.	10.2.6 Initialization, stopping, or pausing of the audit logs	AWS IAM policies prevent start/stop of AWS CloudTrail, S3 bucket policies protect access to log data, alerts are sent if AWS CloudTrail is disabled, AWS Config rule provides monitoring of AWS CloudTrail enabled
	10.3 Record at least the following audit trail entries for all system components for each event:	
51.	10.4 Using time-synchronization technology, synchronize all critical system clocks and times and ensure that the following is implemented for acquiring, distributing, and storing time.	All instances launched in VPC are synced with NTP. User data is injected for time sync

	Note: One example of time synchronization technology is Network Time Protocol (NTP).	
52.	10.4.1 Critical systems have the correct and consistent time.	All instances launched in VPC are synced with NTP using user data that is implicitly added. All log data has timestamp provided by NTP
53.	10.4.3 Time settings are received from industry-accepted time sources.	All instances launched in VPC are synced with AWS NTP servers which in turn obtain time from NTP.org
54.	10.5.1 Limit viewing of audit trails to those with a job-related need.	Audit trails views access are part of the Cloud Automation Platform Access controls
55.	10.5.3 Promptly back up audit trail files to a centralized log server or media that is difficult to alter.	This is done automatically by Cloud Automation Platform
56..	10.6.1 Review the following at least daily: • All security events • Logs of all system components that store, process, or transmit CHD and/or SAD • Logs of all critical system components • Logs of all servers and system components that perform security functions (for example, firewalls, intrusion-detection systems/intrusion-prevention systems (IDS/IPS), authentication servers, e-commerce redirection servers, etc.).	Done by Cloud Automation Platform SOC Team
Requirement 11: Regularly test security systems and processes		
57.	11.2 Run internal and external network vulnerability scans at least quarterly and after any significant change in the network (such as new system component installations, changes in network topology, firewall rule modifications, product upgrades).	Offered as part of Cloud Automation Platform SOC

	Note: Multiple scan reports can be combined for the quarterly scan process to show that all systems were scanned and all applicable vulnerabilities have been addressed. Additional documentation may be required to verify non-remediated vulnerabilities are in the process of being addressed. For initial PCI DSS compliance, it is not required that four quarters of passing scans be completed if the assessor verifies 1) the most recent scan result was a passing scan, 2) the entity has documented policies and procedures requiring quarterly scanning, and 3) vulnerabilities noted in the scan results have been corrected as shown in a re-scan(s). For subsequent years after the initial PCI DSS review, four quarters of passing scans must have occurred.	
58.	11.2.1 Perform quarterly internal vulnerability scans. Address vulnerabilities and perform rescans to verify all “high risk” vulnerabilities are resolved in accordance with the entity’s vulnerability ranking (per Requirement 6.1). Scans must be performed by qualified personnel.	Offered as part of Cloud Automation Platform SOC
59.	11.3.3 Exploitable vulnerabilities found during penetration testing are corrected and testing is repeated to verify the corrections.	Cloud Automation Platform enables WAF rules to mitigate many of these vulnerabilities if the application change is less viable
60.	11.4 Use intrusion-detection and/or intrusion-prevention techniques to detect and/or prevent intrusions into the network. Monitor all traffic at the perimeter of the cardholder data environment as	Cloud Automation Platform orchestrates AWS Guard Duty for NIDS and Ossec for HIDS

	well as at critical points in the cardholder data environment, and alert personnel to suspected compromises. Keep all intrusion-detection and prevention engines, baselines, and signatures up to date.	
61.	11.5.1 Implement a process to respond to any alerts generated by the change detection solution.	Cloud Automation Platform SOC team will receive the email and operate as per the defined and approved Incident management solution

Control-by-Control HIPAA Implementation Detail

	HIPAA Regulation Text	Cloud Automation Platform Implementation
1.	§164.306(a) Covered entities and business associates must do the following: (1) Ensure the confidentiality, integrity, and availability of all electronic protected health information the covered entity or business associate creates, receives, maintains, or transmits. (2) Protect against any reasonably anticipated threats or hazards to the security or integrity of such information. (3) Protect against any reasonably anticipated uses or disclosures of such information that are not permitted or required under subpart E of this part; and (4) Ensure compliance with this subpart by its workforce.	For data at rest Cloud Automation Platform orchestrates KMS keys per tenant to encrypt various AWS resource in that tenant like RDS DBs, S3, Elastic Search, REDIS etc. For data in transit Cloud Automation Platform fetches the certificates from cert manager and all the requests can be made through TLS.
2.	§164.308(a) A covered entity or business associate must in accordance with §164.306: (1)(i) Implement policies and procedures to prevent, detect, contain, and correct security violations.	Usage of a rules-based approach makes the configuration error free, consistent, and documented. In addition, Cloud Automation Platform also provides audit trails for any change in the system.

3.	§164.308(a)(1)(ii)(A) Conduct an accurate and thorough assessment of the potential risks and vulnerabilities to the confidentiality, integrity, and availability of electronic protected health information held by the covered entity or business associate.	Inherited from AWS.
4.	§164.308(a)(1)(ii)(B) Implement security measures sufficient to reduce risks and vulnerabilities to a reasonable and appropriate level to comply with §164.306(a).	Usage of a rules-based approach makes the configuration error free, consistent, and documented.
6.	§164.308(a)(3)(i) Implement policies and procedures to ensure that all members of its workforce have appropriate access to electronic protected health information, as provided under paragraph (a)(4) of this section, and to prevent those workforce members who do not have access under paragraph (a)(4) of this section from obtaining access to electronic protected health information.	Cloud Automation Platform introduces a concept of a tenant which is a logical construct above AWS and represents an application's entire lifecycle. It is a security boundary implemented by having a unique SG, IAM Role and Instance Profile per tenant. By default, no access is allowed into the tenant unless specific ports are exposed via ELB.
8.	§164.308(a)(5)(ii)(D) Procedures for creating, changing, and safeguarding passwords.	Cloud Automation Platform enables user specified password or random password generation options. User access is managed in such a way that all end user access is via single sign on and password less. Even access to AWS console is done by generating a federated console URL that has a validity of less than an hour. The system enables operations with minimal user accounts as most access is JIT.
11.	§164.308(a)(7)(i) Establish (and implement as needed) policies and procedures for responding to an emergency or other occurrence (As one illustrative example, fire, vandalism, system failure, and natural disaster) that damages systems that contain electronic protected health information.	Duplo infrastructure is created with 2 or more availability zones. With this alternate storage and processing capability that dynamically provides transfer and resumption of system operation in times of failure.

12.	§164.308(a)(7)(ii)(A) Establish and implement procedures to create and maintain retrievable exact copies of electronic protected health information.	Inherited from AWS.
13.	§164.308(a)(7)(ii)(B) Establish (and implement as needed) procedures to restore any loss of data.	Cloud Automation Platform includes DR and BCP. This includes data backups for services like S3, EBS and RDS. The automation supports multi-regions with the platform that can be deployed in different regions as per the BCP needs. The MTTR is minimized and is typically less than an hour by virtue of the automation.
14.	§164.308(a)(7)(ii)(C) Establish (and implement as needed) procedures to enable continuation of critical business processes for protection of the security of electronic protected health information while operating in emergency mode.	Duplo infrastructure is created with 2 or more availability zones. With this alternate storage and processing capability that dynamically provides transfer and resumption of system operation in times of failure.
15.	§164.310(a)(2)(i) Establish (and implement as needed) procedures that allow facility access in support of restoration of lost data under the disaster recovery plan and emergency mode operations plan in the event of an emergency.	Duplo infrastructure is created with 2 or more availability zones. With this alternate storage and processing capability that dynamically provides transfer and resumption of system operation in times of failure.
16.	§164.310(a)(2)(iii) Implement procedures to control and validate a person's access to facilities based on their role or function, including visitor control, and control of access to software programs for testing and revision.	Cloud Automation Platform's single sign on functionality over various cloud system accesses enable a Just in time and secure access to software systems. Cloud Automation Platform enables user specified password or random password generation options. User access is managed in such a way that all end user access is via single sign on and password less. Even access to AWS console is done by generating a federated console URL that has a validity of less than an hour.
17.	§164.310(d)(2)(iv) Create a retrievable, exact copy of electronic protected health information, when needed, before movement of equipment.	Inherited from AWS.

18.	§164.312(a)(1) Implement technical policies and procedures for electronic information systems that maintain electronic protected health information to allow access only to those persons or software programs that have been granted access rights as specified in §164.308(a)(4).	Cloud Automation Platform tenant model has access controls built in. This allows access to various tenant based on the user roles. This access control mechanism automatically integrates into the VPN client as well i.e., each user has a static IP in the VPN and based on his tenant access his IP is added to the respective tenant's SG. Tenant access policies will automatically apply SG or IAM based policy based on the resource type.
19.	§164.312(a)(2)(i) Assign a unique name and/or number for identifying and tracking user identity.	Cloud Automation Platform integrates with our IDP like G Suite and O365 for access to the portal. From there a federated logic in done for AWS resource access.
20.	§164.312(a)(2)(ii) Establish (and implement as needed) procedures for obtaining necessary electronic protected health information during an emergency.	Duplo infrastructure is created with 2 or more availability zones. With this alternate storage and processing capability that dynamically provides transfer and resumption of system operation in times of failure.
21.	§164.312(a)(2)(iii) Implement electronic procedures that terminate an electronic session after a predetermined time of inactivity.	Inherited from AWS.
22.	§164.312(a)(2)(iv) Implement a mechanism to encrypt and decrypt electronic protected health information.	Cloud Automation Platform orchestrates KMS keys per tenant to encrypt various AWS resources in that tenant like RDS DBs, S3, Elastic Search, REDIS etc. Access to the KMS keys is granted only to the instance profile w/o any user accounts or keys. By default, Cloud Automation Platform creates a common KMS key per deployment but allows the ability to have one key per tenant.
24.	§164.312(c)(1) Implement policies and procedures to protect electronic protected health information from improper alteration or destruction.	Cloud Automation Platform introduces a concept of a tenant which is a logical construct above AWS and represents an application's entire lifecycle. It is a security boundary implemented by having a unique SG, IAM Role and Instance Profile per tenant. By default, no access is allowed into the tenant unless specific ports are exposed via ELB.
25.	§164.312(c)(2) Implement electronic mechanisms to corroborate that electronic protected health information has	Cloud Automation Platform introduces a concept of a tenant which is a logical construct above AWS and represents an application's entire lifecycle. It is a security boundary implemented by having a unique SG, IAM Role

	not been altered or destroyed in an unauthorized manner.	and Instance Profile per tenant. By default, no access is allowed into the tenant unless specific ports are exposed via ELB.
26.	§164.312(e)(1) Implement technical security measures to guard against unauthorized access to electronic protected health information that is being transmitted over an electronic communications network.	DC gets certificates from Cert-Manager and automates SSL termination in the ELB. In addition, TLS/SSH ports are enforced in the security groups by the Cloud Automation Platform.
27.	§164.312(e)(2)(i) Implement security measures to ensure that electronically transmitted electronic protected health information is not improperly modified without detection until disposed of.	Cloud Automation Platform by default orchestrates appropriate services like Encryption at rest and transit to protect data integrity.
28.	§164.312(e)(2)(ii) Implement a mechanism to encrypt electronic protected health information whenever deemed appropriate.	Cloud Automation Platform orchestrates KMS keys per tenant to encrypt various AWS resources in that tenant like RDS DBs, S3, Elastic Search, REDIS etc. Access to the KMS keys is granted only to the instance profile w/o any user accounts or keys. By default, Cloud Automation Platform creates a common KMS key per deployment but allows the ability to have one key per tenant.

Dynamic Application Security Testing

ZAP is run periodically to security penetration test our application against OWASP vulnerabilities. OWASP ZAP is an open-source web application security scanner. It is intended to be used by both those new to application security as well as professional penetration testers. We configured ZAP and proxied all the URLs of the Web-Application to ZAP, which then scans and attacks the URLs and generates reports.