PRJ0041610 - Ansible Automation of Windows Server Privileged Access Provisioning

Project Title	**PRJ0041610 - Ansible Automation of Windows Server Privileged Access Provisioning**
Project Code	PRJ0041610
Owner	Graham Gold
Project Phase	Production Implementation Prep
HLD Status	Approved
Start Date	08 May 2023
End Date	23 Jun 2023

- Document Control
- Authorisation
- Review Sign-Off
- Circulation
- Glossary of Terms
- References
- 1 Strategic Summary
- 2 Introduction
    - 2.1 Purpose of the Document
    - 2.2 Background
    - 2.3 Objectives
- 3 SDLC Stage
- 4 Architecture Dependency Matrix
- 5 Requirements
    - 5.1 Functional Requirements:
    - 5.3 Non Functional Requirements:
- 6 Scope
    - 6.1 In Scope
    - 6.2 Out of Scope
- 7 Assumptions
- 8 Dependencies
- 9 Risks
- 10 Issues
- 11 Design
    - Current Design
    - AD/GPO Structure
    - Current Provisioning Model/Process
    - 11.1 Design Options
        - Enforcement of PAM Desired State on Member Servers - Alternative Approaches
    - Proposal for Proof of Concept
        - Comparison of Options
    - 11.2 Chosen Design
        - Process Flow
        - Infrastructure
        - Migration
    - 11.3 Design Principles

    - 11.4 Application
    - 11.5 Network
    - 11.6 Security
        - Azure DevOps
    - 11.7 Server Deployment
    - 11.8 Data & Privacy
    - 11.9 Logging & Monitoring
    - 11.10 Environment Info
    - 11.11 Storage and Backups
    - 11.12 Resilience
    - 11.13 Business Continuity and Disaster Recovery (BCDR)
- 12 Licensing and Costs
- 13 Ownership
- 14 Operate and Support Model
- 15 Capacity and Performance
- 16 Appendix
    - 16.1 Additional Info

Document Control

Version	Date	Status	Author	Comments
0.1	24 May 2023	Issued	Graham Gold	First draft
0.2	28 May 2023	Updated	Graham Gold	2nd Draft

Authorisation

Mention key personnel who are sponsoring and authorizing this project, along with their signatures and dates.

Name	Role	Signede	Date

Review Sign-Off

State all the key personnel who are going to review and sign-off this design document. This should include architect/consultant/SMEs from the business, Enterprise Services and other functions within M&G. Where a review is performed by Enterprise Services Technical Design Authority (ES-TDA) state ES-TDA.

If design reviews and sign-offs outside ES-TDA are required it is the responsibility of the project design team to ensure such reviews and sign-offs have completed and referenced below before the design is presented to ES-TDA.

Further, it is also the responsibility of the project design team to ensure representation from architecture/consultant/SMEs outside ES-TDA are present in ES-TDA meetings to ensure there is coverage

Name	Role	Signed	Date

ES TDA	*Architectural Governance Body in Enterprise Services*		*Mention date when ES TDA approval chain closed and confirmation sent to project team*
N/A	*Security Consultancy*	*Note: According to Security Consultancy they do not sign documents but publish security findings*	*Mention date when Security Consultancy function communicates the security findings on the project*
Gary Smith	Security Engineering Lead		
Steve Hodge	Security Ops Engineering Lead		
N/A	*Supply Chain Security*		*Mention date when SCS function communicates the findings on the project*
N/A	*Technology Governance Risk Controls*		*Mention date when Risk findings are published*
N/A	*Privacy*		*Mention date when Privacy findings are published*
N/A	*Procurement*		*Mention date when Procurement findings are published*
N/A	*Service Transition*		*Mention date about Service Transition progress needed not needed etc.*
N/A	*CMDB Updates*		*Mention date about CMDB progress needed not needed etc.*

Circulation

State all the personnel either consulted or informed as part of this project

Name	Role
Allan Tuck	Cloud Solution Architect
Mahipal Singh	Senior Cloud Engineer
Rob Slater	Cloud Engineer

Glossary of Terms

Mention all acronyms used in the design, assuming the reader is new to the subject area

Acronym	Definition

References

References

Windows PAM Model: Privileged Access | Windows Member Servers - Enterprise Security - Confluence (valiantys.net)

PAM Provisioning Challenges Workshop Findings: [DRAFT] PAM | Cloud | IaaS | Agile Local Administrator Access and Provisioning - Enterprise Security - Confluence (valiantys.net)

PAM Provisioning - Windows Server - Options Paper and proposed Ansible PoC: [DRAFT] PAM | Cloud | Iaas | Windows Local Administrator Desired State - Enterprise Security - Confluence (valiantys.net)

Windows_PAM Ansible Collection Documentation: Ansible Collections | Windows | Roles | windows_pam - Cloud Platform Services Team - Confluence (valiantys.net)

Ansible Automation Platform intro: Ansible | Ansible Automation Platform (AAP) - Cloud Platform Services Team - Confluence (valiantys.net)

Ansible API Auth: 20. Token-Based Authentication — Automation Controller Administration Guide v4.3 (ansible.com)

Ansible win_group_membership module documentation: https://docs.ansible.com/ansible/2.9/modules/win_group_membership_module.html

Microsoft Guidance on Logic App security: Secure access and data - Azure Logic Apps | Microsoft Learn

Azure API Management Authentication Documentation: https://learn.microsoft.com/en-us/azure/api-management/api-management-howto-protect-backend-with-aad

1 Strategic Summary

Description of the components of the design that comply with MGP IT strategy and TCO objectives.

Table below describes the high level components within the design, and whether they comply with MGP IT strategy or needs to be reviewed as an exception:

Design Component	Strategy Status	Datacenter or Campus Footprint	Comment
Ansible Automation Platform (AAP)	Buy	None - AAP is a SaaS service, with an element of compute infrastructure within our Azure Tenant to interact with managed resources and response to API requests.	Ansible is our strategic automation and configuration management solution for cloud hosted applications and infrastructure.
Azure Sentinel	Buy	None - SaaS SIEM/SOAR solution to automate security event detection and response.	
Azure Logic Apps	Buy	Serverless low-code automation platform (PaaS)	Already used by Sentinel for playbooks to respond to incidents
CyberArk	Buy	Privileged Access Management strategic platform for secure vaulting of privileged credentials, brokering secure sessions to endpoints with session auditing/recording.	

Strategy Status Key:

- *New – Not deployed within the enterprise, or defined as an enterprise standard*
- *Buy – Component of choice for the enterprise, complies with enterprise standard*
- *Hold – Existing enterprise standard, usually legacy that is no longer component of choice for purchase*
- *Sell – No longer enterprise standard; Enterprise is actively looking to move away from*

2 Introduction

2.1 Purpose of the Document

This design addresses challenges with the existing solution used to provision privileged access on Windows domain-joined servers via principals (users and groups) in Active Directory, to allow privileged access.

2.2 Background

Briefly describe the requirements that necessitated the need for executing this project. It is also a good place to give AS-IS state of logical/physical/application architecture. Importantly, brevity of expression is key in ensuring the above is succinctly defined for the benefit of a reader.

At present, in order to gain access to Windows domain joined servers as an administrator, the following must be in place:

Type	Description	Who	How	Elapsed Time to Provision once requirements finalised	In-Scope Of This Solution?
Active Directory	Generic privileged AD user accounts (non-personal) - often referred to as GENPAM accounts due to the Naming Standard	Lansing Exit Platform Capability Engineering Team (Soon to be self-service by LEDE workstream SMEs)	AD Automation Pipeline	0.5 Days	No
Active Directory	AD ACC_PAM_Role_<*team/app*> group that contains the above accounts, for a given team that have a support function for the application/server	Lansing Exit Platform Capability Engineering Team (Soon to be self-service by LEDE workstream SMEs	AD Automation Pipeline		
Active Directory	AD ACC_ServerAdministrators_<*team/app*> group which contains the role group and must be present in the Administrators group of the in-scope servers for the accounts to be able to login	Lansing Exit Platform Capability Engineering Team (Soon to be self-service by LEDE workstream SMEs	AD Automation Pipeline		
CyberArk	CyberArk safe containing the GENPAM accounts	MG Security Inf Sup MSS	ServiceNow Task, complete word form, team then run script to create safe and AD groups for access, then request ARC roles through ARC)	Up to 2 weeks (but usually 3-5 days with escalation)	No
Active Directory	2x AD Groups for access to the safe (as an Owner or a User)	MG Security Inf Sup MSS	ServiceNow Task, complete word form, team then run script to create safe and AD groups for access, then request ARC roles through ARC)		
ARC	2x ARC Roles to grant access to the respective group(s)	ARC Team	ServiceNow Task, complete word form, team then run script to create safe and AD groups for access, then request ARC roles through ARC)		
Active Directory	1x AD Machine group (SRV_COMP_SRV_CORE_CYB_<*team/app*>) per AD domain that the in-scope servers are added to apply the GPO COMP_SRV_CORE_CYB_<*team/app*> to the server(s)	Lansing Exit Platform Capability Engineering Team (Soon to be self-service by LEDE workstream SMEs	AD Automation Pipeline	0.5 Days	Yes

Active Directory	1x GPO (COMP_SRV_CORE_CYB_<*team/app*>) per domain to apply an agreed set of Service Accounts and groups to the local Administrators group of the servers the GPO applies to.	LEDE workstream SMEs	NetIQ Group Policy Administrator (GPA) (manual GUI driven process with 2-day Change Lead Time)	2-3 Days	Yes

The focus of this solution is to reduce the mean-time to deliver for granting administrative access to Windows Servers - specifically the reliance on Group Policy Objects as the mechanism to do this.

At present, a VM can be built in around 18 minutes, but it would take at least 3 days (due to ITIL requirements around the Change Management process) to provision access to app teams on the VM via GPO (longer if they don't have GENPAM accounts, CyberArk safe etc already in place).

Due to the decision to shift-left and allow application teams administrative access to their own servers, this would lead to significant sprawl of GPOs (potentially 1 GPO per server in the worst case scenario) and is therefore unsustainable from a number of perspectives:

- Lead time to create and deploy GPOs (resource constraint + ITIL change management lead times)
- GPO Processing time (each GPO needs to be evaluated by each domain joined server when it starts and every 90-120 minutes thereafter)
- Resource impact on Member servers, domain controllers, network bandwidth due to significant increase in processing demands
- Risk of failure of GPO processing due to the above, which could have service and security impacts.

2.3 Objectives

1. Utilise Infrastructure-as-code/Automation first.
2. Utilise self-service principles to allow developers and cloud engineers to provision access, more quickly, reduce delay between VM build completion and access being available.
3. Maintain appropriate governance/compliance controls and auditing of privileged access grants.
4. Utilize strategic tooling where possible.
5. Minimise bespoke code - buy before build.

3 SDLC Stage

N/A

4 Architecture Dependency Matrix

Describe the key architectural decisions or building blocks this design is dependent on. For example: if an application/vendor was chosen by the business stakeholders and if this design is associated with realising that application then state the application as a dependency with associated commentary.

Business Aligned Architect	Designation/Function (Ex: Application/Security)	Review/Approval Required	Comments

5 Requirements

Describe in summary, all the functional and non-functional requirements this project design will fulfil.

5.1 Functional Requirements:

Re f	Requirement Summary
FR1	Infrastructure Requirements - PaaS/SaaS only
FR2	Business Requirement - Enable timely self-service provisioning of administrative access to Windows domain joined Virtual Machines
FR3	Authentication and Authorisation levels - Ensure all authentication involved in all aspects of the solution adhere to PAM Controls, Access Management Controls and use strong authentication.
FR4	Internal and External Interfaces - Azure DevOps is primary interface for users/consumers of the solution. Integration to Sentinel is also configured to allow remediation if an administrative user were to manually amend the administrators group.
FR5	Reporting Requirements - VM Build process already reports on status of Ansible Role installation at VM Build time - new role would be part of the same process and reporting inside of Azure DevOps when building a VM.

Note : Add more as per Project scope and demand

5.3 Non Functional Requirements:

Ref	Requirement Summary
NFR1	Scalability – *Must be able to scale to meet demands - met by use of AAP and ADO.*
NFR2	Availability– *Must be highly available - VM Build process must not be impacted by this solution. Met by use of AAP and ADO*
NFR3	Performance – VM admin access must be provisioned in near real-time - within 5-10 minutes of VM being built.
NFR4	Capacity – *No capacity concerns due to use of SaaS/PaaS and CI/CD pipelines*
NFR5	Reliability – *Solution will use AAP, ADO which are both highly reliable strategic platforms.*
NFR6	Resiliency – *As per the resilience of AAP and ADO - highly resilient SaaS/PaaS*

Note : Add more as per Project scope and demand in terms of SLA,

6 Scope

6.1 In Scope

The scope of this design is specifically around the provision of Administrative access to Windows domain joined servers, via membership of the Administrators local group and replacing the current GPO mechanism for this provisioning.

6.2 Out of Scope

This design does not cover any other aspect of PAM provisioning (creation of users/groups in AD, creation/update of CyberArc Safes or ARC Roles).

This expressly covers the access to the VM for existing AD groups and service accounts,

This design does not cover on-premises servers - however, as a design principle, on-premise servers could adopt the same solution (see later caveats on this).

7 Assumptions

Describe the assumptions this project design made.

Ref	Assumption	Comments

8 Dependencies

Describe the dependencies on the external systems here.

Ref	Dependency	Comments	Status

9 Risks

Describe all the known risks here

Ref	Risk	Likelihood	Impact	Mitigation

10 Issues

Describe all the known issues here.

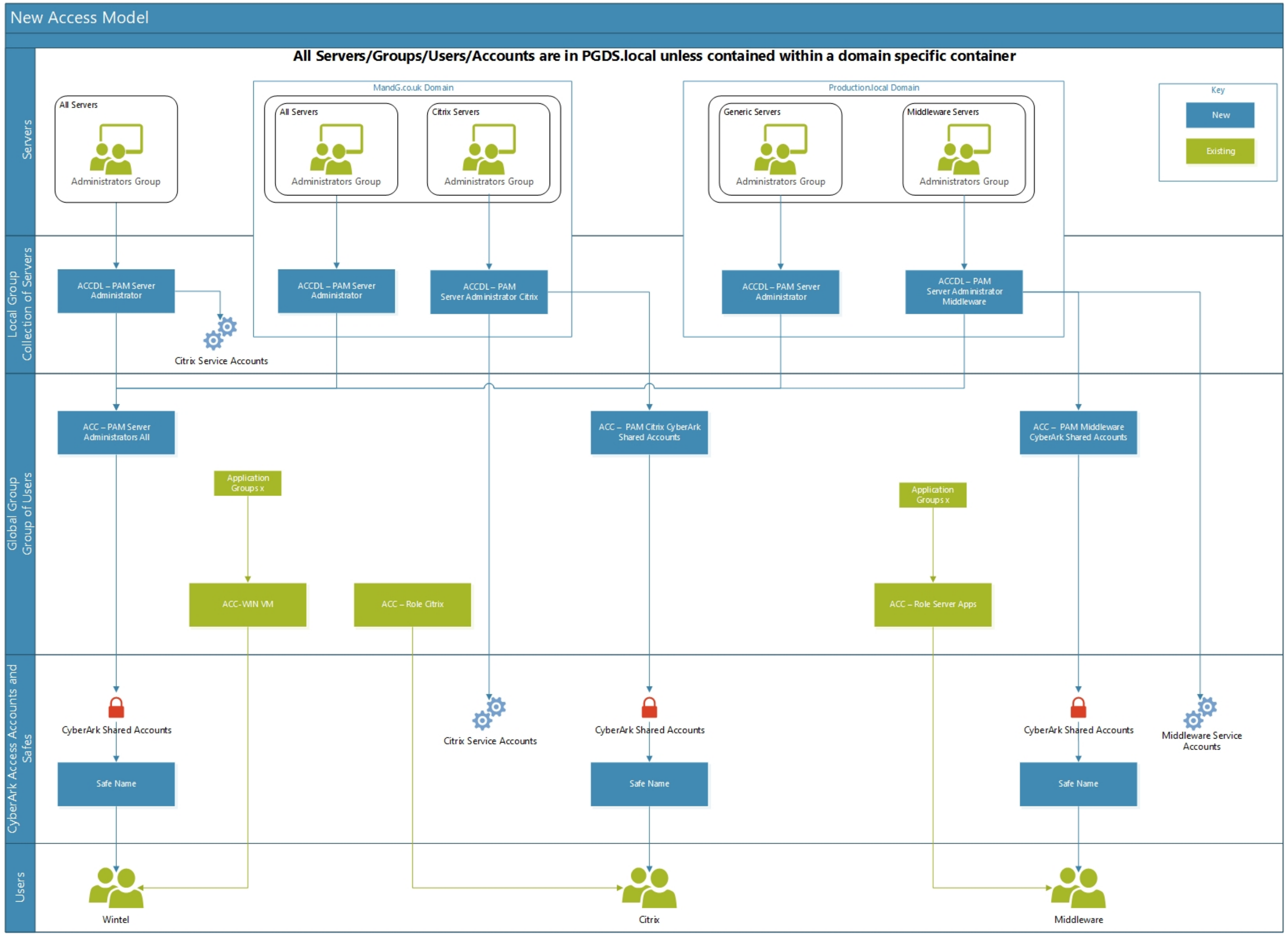
Ref	Issue	Comments

11 Design

Current Design

AD/GPO Structure

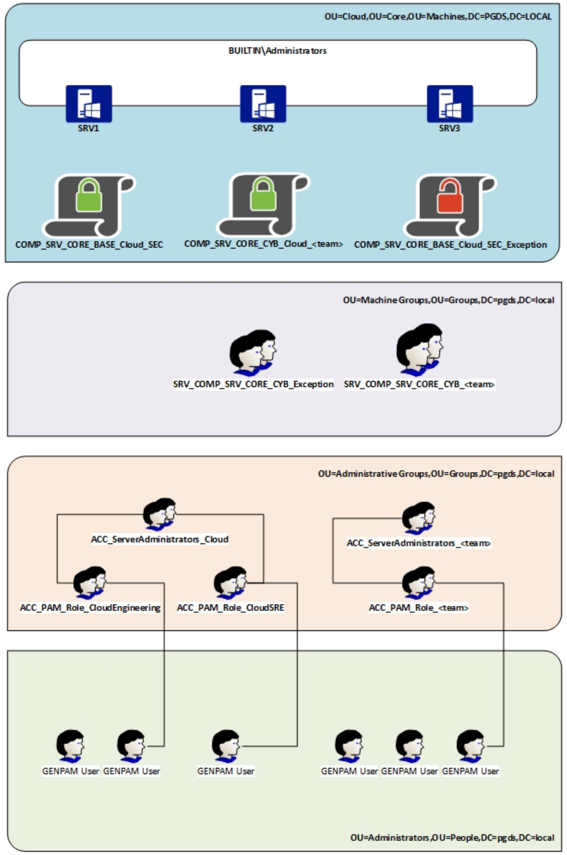
The current model/design, from an Active Directory Structure perspective is as below:



Current Provisioning Model/Process

How this works in practice is as below:

PAM Access Provisioning Model For Cloud (Current)


OU=Cloud,OU=Core,OU=Machines,DC=PGDS,DC=LOCAL
BUILTIN\Administrators
SRV1
SRV2
SRV3
COMP_SRV_CORE_BASE_Cloud_SEC
COMP_SRV_CORE_CYB_Cloud_<team>
COMP_SRV_CORE_BASE_Cloud_SEC_Exception
OU=Machine Groups,OU=Groups,DC=pgds,DC=local
SRV_COMP_SRV_CORE_CYB_Exception
SRV_COMP_SRV_CORE_CYB_<team>
OU=Administrative Groups,OU=Groups,DC=pgds,DC=local
ACC_ServerAdministrators_Cloud
ACC_ServerAdministrators_<team>
ACC_PAM_Role_CloudEngineering
ACC_PAM_Role_CloudSRE
ACC_PAM_Role_<team>
GENPAM User
GENPAM User
GENPAM User
GENPAM User
GENPAM User
GENPAM User
OU=Administrators,OU=People,DC=pgds,DC=local

Provisioning and Enforcement Process (Cloud)

1. **Server Build (all domain joined servers):**
   a. Join to Domain in Cloud OU
   b. Group Policy applied based on:
      - GPOs Inherited from parent OUs (e.g. SCB GPO)
      - GPOs in the servers' OU (applied based on link order)
      - Delegation (AD Machine Groups to conditionally apply or exclude GPOs from servers in groups)
   c. Default PAM Access Provisioning for all servers:
      - PGDS\ACC_ServerAdministrators_Cloud and CyberArk Discovery service account added to BUILTIN\Administrators on all servers and **will not permit any other members of Administrators.**
      - This group has as members: ACC_PAM_Role_CloudEngineering and ACC_PAM_Role_CloudSRE, Service Accounts required on all servers for management tooling.
      - These groups contain the GENPAMWNxxxPnn accounts that are vaulted in CyberArk
      - RDP Access to server facilitated via CyberArk PSM (session recording & session isolation)
      - Password retrieval possible from CyberArk with additional approval workflow

2. **At build time or as a later addition, if another team requires access:**
   a. Add server object in AD into the appropriate SRV_COMP_SRV_CORE_CYB_<team> group linked to COMP_SRV_CORE_CYB_Cloud_<team> GPO so that the GPO is applied
   b. GPO adds the PGDS\ACC_ServerAdministrators_<team> group to BUILTIN\Administrators group on server in addition to the groups and accounts added by the base GPO.
   c. **Administrators group membership is again restricted membership.**
   d. Access is then available as before via CyberArk for the CPS teams and the app team.

3. **In the scenario where a single server or subset of servers belonging to a team requires service accounts to be a member of the BUILTIN\Administrators group:**
   a. The server object in AD is added into the COMP_SRV_CORE_BASE_Cloud_SEC_Exception machine group
   b. This adds the same accounts and groups as the base policy, but instead of being a Restricted Group, this is only a preference that ensure the base set of members are always applied:
      Other groups and users can be added and will persist rather than being removed by Group Policy
   c. This is the most sustainable approach without creating custom GPOs for individual servers or groups of servers.

What happens when Group Policy Refreshes?

COMP_SRV_CORE_BASE_Cloud_SEC
Any groups/users in Administrators but not in the GPO are removed

COMP_SRV_CORE_BASE_Cloud_<Team>
Any groups/users in Administrators but not in the GPO are removed

COMP_SRV_CORE_BASE_Cloud_SEC_Exception
If the groups/users in the GPO are missing they are added back in. All other members are left as-is.

When does Group Policy Refresh?

1. When the server joins the domain
2. When the server starts/restarts
3. Every 90-120 minutes (configurable) from when the server last refreshed Group Policy
4. When a user logs on (for User policies – PAM GPOs are Computer Policies)
5. Can be forced by a logged on administrator using a command line tool.

11.1 Design Options

Enforcement of PAM Desired State on Member Servers - Alternative Approaches

Option 1 - Group Policy Automation/Enhancement

As the current GPO process is both largely manual, and only performed by 2 members of Enterprise Security (Security Engineering), automation could be explored as follows:

1. Use the NetIQ Group Policy Adminstrator tools' PowerShell cmdlets to automate the manual GUI driven process used in GPA today - Automating GPA Operations with PowerShell Cmdlets - Group Policy Administrator User Guide (netiq.com)
2. Seek to codify GPOs as JSON templates and use ADO to manage GPO as code, and create pipeline to execute powershell scripts on GPA server to perform GPO changes
3. Investigate possibility of reducing the Group Policy refresh time to invoke the refresh more frequently (this isn't an automation deliverable so much as an attempt to strengthen the control by reducing the drift window)

Option 2 - Powershell Desired State Configuration (DSC)

Policy Based processes such as Group Policy Objects will, by their nature, always have an element of lag and a window within which the configuration can drift from the desired state, which presents both service and security risks.

It would be possible to use Powershell Desired State Configuration (DSC) to set the local group membership on servers.

This can be delivered in a number of ways (via GPO, via logon scripts, via Configuration Management tooling).

This has the advantage of not having a refresh window by default (though a pull model is possible, it has been discounted at this stage as it requires additional infrastructure and operational overhead) - however drift would still be possible as an admin can interfere with the DSC process itself.

In addition, it requires significant effort to configure and manage, and there is no natural owner for that overhead today.

Option 3 - Ansible Automation Platform (Configuration Management/Automation Tooling)

M&G are already in the process of deploying the Ansible Automation Platform (AAP) automation product to centrally manage configuration of applications and resources in Azure.

AAP has a win_group_membership module that can set the membership of local groups on Windows servers.

Normally, AAP playbooks for a given application (which can include Infrastructure-as-a-Service (Iaas) VMs) would only run when the application is deployed, which covers the build scenario.

However, the configuration can then drift as by default there is no further run of the playbook for that application unless triggered again (build, manual trigger).

AAP does support scheduling of job templates (and their associated playbooks) on a desired frequency.

However, to run this against all playbooks even as frequently as GPO refreshes today would add significant complexity to the AAP configurations and, more importantly, place significant additional workload on AAP and also Network bandwidth.

GPO is a pull process - the server polls AD every 90-120 minutes to ask if there are any changes to group policy and apply them. This means that the load on the domain controllers and the network is distributed - not all computers are doing a GP update at the same time.

AAP schedules are very much a push process, and would mean AAP pushing out config to every server it manages the configuration for every time the schedule runs - a significant overhead in terms of CPU, Memory and Network Bandwidth.

Therefore a solution is required to allow AAP to enforce desired state in a more sustainable, less resource intensive, less disruptive manner. (See proposal below for detail on how this is achieved)

Option 4 - Azure Virtual Machine Administrator RBAC role with PIM

Shaun O'connor has explored the use of Azure PIM as an alternative to CyberArk and has also highlighted the relatively new capability of using the Virtual Machine Administrator RBAC role in Azure.

This would entail a user who has that role for a VM, Resource Group etc using PIM to elevate to that role.

Doing so would give them local administrator access and enable RDP access at the NIC/NSG level for the duration of the PIM elevation period.

Proposal for Proof of Concept

Comparison of Options

Option	Pros	Cons
1	Increased Automation Adopting IaC and DevOps strategic approaches Possibility to reduce reliance on a single person Possibility to reduce drift window	Requires significant effort to investigate and develop Single resource for development Decreasing the GP refresh window may have an adverse impact on network bandwidth and domain controllers, potentially degrading service for all users/servers/applications in that domain
2	DSC alleviates the need for an Exception group therefore drift is less likely - administrator group membership is still enforced rather than just a preference. DSC doesn't use a polling mechanism by default (though a pull mode is available - which has been discounted to keep the operational overhead as small as possible), so is less demanding on the network.	DSC still requires a management framework, and an owner and resources to operate DSC can be tampered with, therefore elimination of drift from desired state is not something that can be fully relied upon.
3	Increased Automation Adopting IaC and DevOps strategic approaches Possibility to reduce reliance on a single person Possibility to reduce drift window Adopting strategic automation tooling for cloud, which will have it's own owners and resources	Scheduling playbooks is not advised by the vendor, especially for this use case, due to resource overheads on AAP itself, as well as member servers and network bandwidth (but there are solutions to work around this to achieve the desired outcome)
4	Simplified and automatable provisioning Uses native features Adopts a Just-In-Time approach - no standing privilege on endpoints	New feature, perhaps not mature enough in terms of features Exposed to the PIM tailgating risk discussed elsewhere. Takes privileged access outside of the PAM controls - e.g direct access, not via PSM, so no session isolation and no session recording. Requires RDP to be enabled from all clients to the Azure address space, otherwise access is not possible (could perhaps restrict to an admin subnet, but would require adoption of Privileged Access Workstation (PAW) model or other bastion host (of which CyberArk PSM is already a bastion host). Using this mechanism joins the VM to Azure AD - not something we are ready for in our extensive hybrid estate. https://docs.microsoft.com/en-us/azure/active-directory/devices/howto-vm-sign-in-azure-ad-windows

Based on the above analysis, the recommendation was to take forward a PoC of option 3, AAP.

Key to note is that, in order to address the issue with scheduling of playbooks, the PoC adopted the use of the Ansible AWX API to trigger the execution of a playbook for a specific server if a change to it's local administrators group is detected by SOC/CSIRT - thereby reducing the frequency, scope and impact of any non Build related playbook runs, whilst allowing certainty that desired state is maintained.

This requires the Sentinel Playbook to run a Logic App to connect to the Ansible API to run the playbook that will reinstate the correct local Administrators membership.

This has been verified by RedHat as a viable and accepted use case and they have verified that an event driven approach such as this is preferred over scheduling of regular playbook runs.

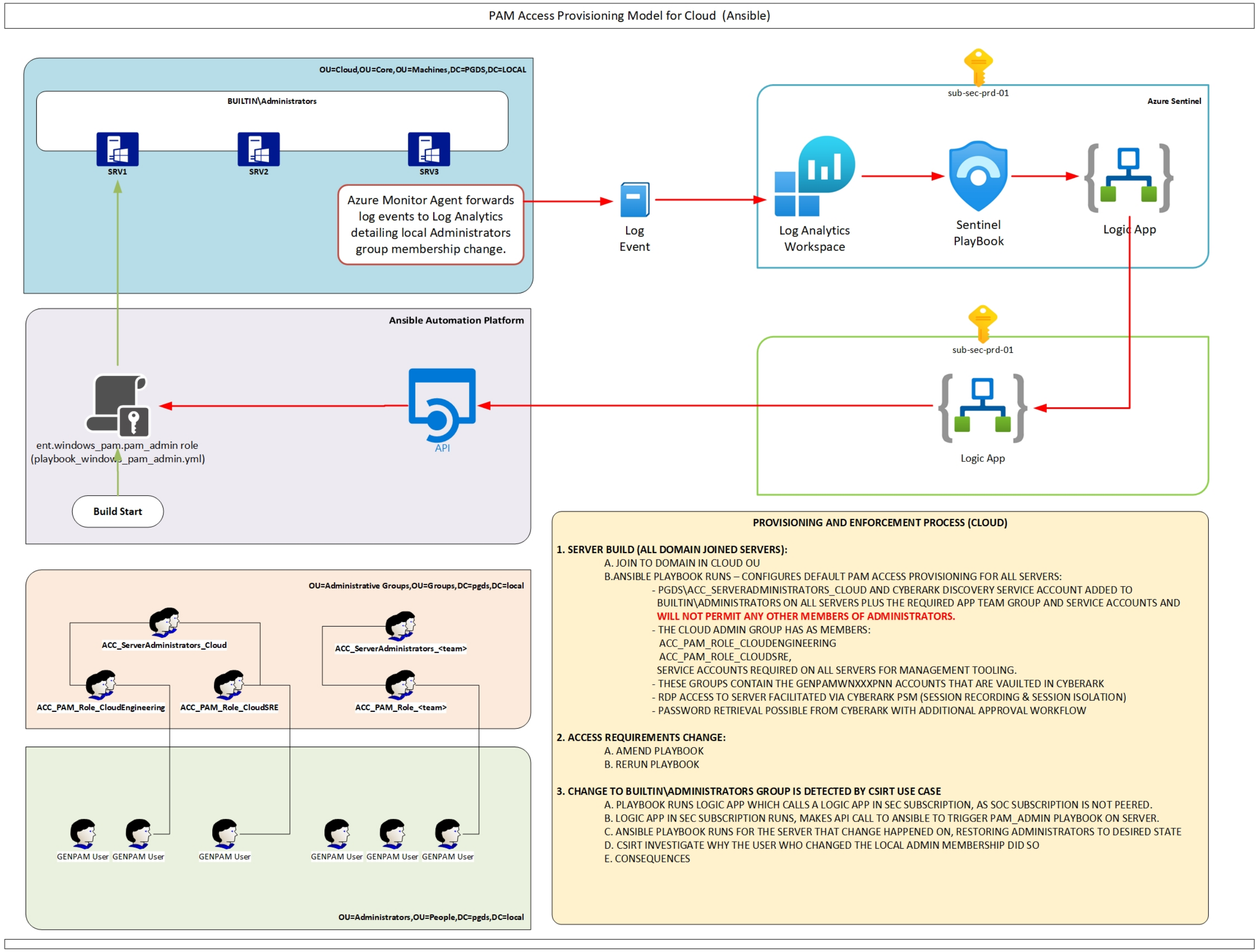
As the playbook source code is maintained in a git repo (in Azure DevOps), all changes to local administrators group access are therefore source controlled, with appropriate approval workflows.

11.2 Chosen Design

Option 3 above was chosen and has been built out successfully as a proof of concept using AAP non-production environment, some non-production VMs in sub-en-nonprod-01, production Azure Sentinel, Logic App in sub-soc-nonprod-01 and a Standard logic app, with VNET integration, in sub-en-rnd-01.

The reason for the logic app double-hop is that the soc subscriptions have no networking, but the api call needs to be able to reach the AAP controller, which is hosted in our Azure tenant, in one of our VNETs - there is no other way to access the controller SaaS endpoint - access is tightly controlled.

Process Flow


PAM Access Provisioning Model for Cloud (Ansible)

OU=Cloud,OU=Core,OU=Machines,DC=PGDS,DC=LOCAL
BUILTIN\Administrators
SRV1
SRV2
SRV3
Log Event
sub-sec-prd-01
Azure Sentinel
Log Analytics Workspace
Sentinel PlayBook
Logic App
Ansible Automation Platform
ent.windows_pam.pam_admin role (playbook_windows_pam_admin.yml)
API
Build Start
sub-sec-prd-01
Logic App
OU=Administrative Groups,OU=Groups,DC=pgds,DC=local
ACC_ServerAdministrators_Cloud
ACC_ServerAdministrators_<team>
ACC_PAM_Role_CloudEngineering
ACC_PAM_Role_CloudSRE
ACC_PAM_Role_<team>
GENPAM User
GENPAM User
GENPAM User
GENPAM User
GENPAM User
GENPAM User
OU=Administrators,OU=People,DC=pgds,DC=local

Azure Monitor Agent forwards log events to Log Analytics detailing local Administrators group membership change.

PROVISIONING AND ENFORCEMENT PROCESS (CLOUD)

1. SERVER BUILD (ALL DOMAIN JOINED SERVERS):

A. JOIN TO DOMAIN IN CLOUD OU

B.ANSIBLE PLAYBOOK RUNS – CONFIGURES DEFAULT PAM ACCESS PROVISIONING FOR ALL SERVERS:

- PGDS\ACC_SERVERADMINISTRATORS_CLOUD AND CYBERARK DISCOVERY SERVICE ACCOUNT ADDED TO BUILTIN\ADMINISTRATORS ON ALL SERVERS PLUS THE REQUIRED APP TEAM GROUP AND SERVICE ACCOUNTS AND **WILL NOT PERMIT ANY OTHER MEMBERS OF ADMINISTRATORS.**
- THE CLOUD ADMIN GROUP HAS AS MEMBERS:
  ACC_PAM_ROLE_CLOUDENGINEERING
  ACC_PAM_ROLE_CLOUDSRE,
  SERVICE ACCOUNTS REQUIRED ON ALL SERVERS FOR MANAGEMENT TOOLING.
- THESE GROUPS CONTAIN THE GENPAMWNXXXPNN ACCOUNTS THAT ARE VAULTED IN CYBERARK
- RDP ACCESS TO SERVER FACILITATED VIA CYBERARK PSM (SESSION RECORDING & SESSION ISOLATION)
- PASSWORD RETRIEVAL POSSIBLE FROM CYBERARK WITH ADDITIONAL APPROVAL WORKFLOW

2. ACCESS REQUIREMENTS CHANGE:

A. AMEND PLAYBOOK

B. RERUN PLAYBOOK

3. CHANGE TO BUILTIN\ADMINISTRATORS GROUP IS DETECTED BY CSIRT USE CASE

A. PLAYBOOK RUNS LOGIC APP WHICH CALLS A LOGIC APP IN SEC SUBSCRIPTION, AS SOC SUBSCRIPTION IS NOT PEERED.

B. LOGIC APP IN SEC SUBSCRIPTION RUNS, MAKES API CALL TO ANSIBLE TO TRIGGER PAM_ADMIN PLAYBOOK ON SERVER.

C. ANSIBLE PLAYBOOK RUNS FOR THE SERVER THAT CHANGE HAPPENED ON, RESTORING ADMINISTRATORS TO DESIRED STATE

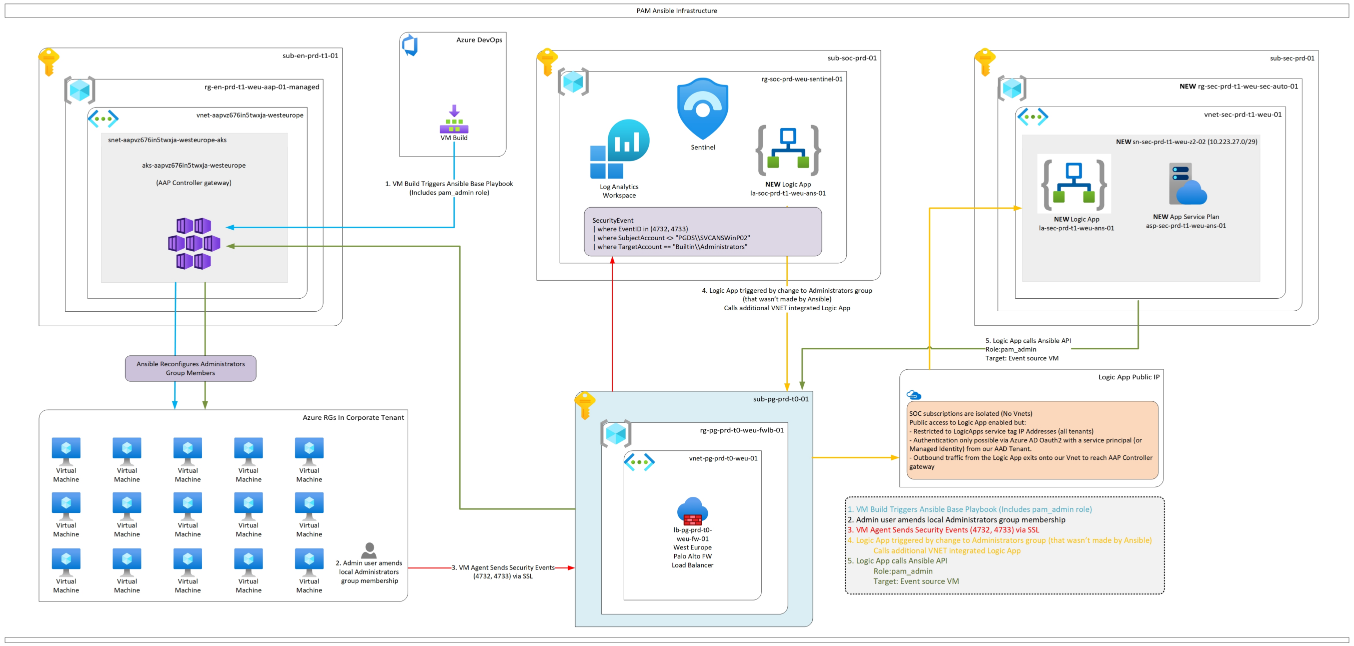
D. CSIRT INVESTIGATE WHY THE USER WHO CHANGED THE LOCAL ADMIN MEMBERSHIP DID SO

E. CONSEQUENCES

Infrastructure

Full AAP infrastructure: Ansible | Ansible Automation Platform (AAP) - Cloud Platform Services Team - Confluence (valiantys.net)

Infrastructure for this solution:


PAM Ansible Infrastructure
sub-en-prd-t1-01
rg-en-prd-t1-weu-aap-01-managed
vnet-aapvz676in5twxja-westeurope
snet-aapvz676in5twxja-westeurope-aks
aks-aapvz676in5twxja-westeurope
(AAP Controller gateway)
Azure DevOps
VM Build
1. VM Build Triggers Ansible Base Playbook
(Includes pam_admin role)
sub-soc-prd-01
rg-soc-prd-weu-sentinel-01
Sentinel
Log Analytics Workspace
NEW Logic App
la-soc-prd-t1-weu-ans-01
4. Logic App triggered by change to Administrators group
(that wasn't made by Ansible)
Calls additional VNET integrated Logic App
sub-sec-prd-01
NEW rg-sec-prd-t1-weu-sec-auto-01
vnet-sec-prd-t1-weu-01
NEW sn-sec-prd-t1-weu-z2-02 (10.223.27.0/29)
NEW Logic App
la-sec-prd-t1-weu-ans-01
NEW App Service Plan
asp-sec-prd-t1-weu-ans-01
5. Logic App calls Ansible API
Role:pam_admin
Target: Event source VM
Logic App Public IP
Ansible Reconfigures Administrators Group Members
Azure RGs In Corporate Tenant
Virtual Machine
2. Admin user amends local Administrators group membership
3. VM Agent Sends Security Events
(4732, 4733) via SSL
sub-pg-prd-t0-01
rg-pg-prd-t0-weu-fwlb-01
vnet-pg-prd-t0-weu-01
lb-pg-prd-t0-weu-fw-01
West Europe
Palo Alto FW
Load Balancer

```
SecurityEvent
| where EventID in (4732, 4733)
| where SubjectAccount <> "PGDS\\SVCANSWinP02"
| where TargetAccount == "Builtin\\Administrators"
```

SOC subscriptions are isolated (No Vnets)
Public access to Logic App enabled but:
- Restricted to LogicApps service tag IP Addresses (all tenants)
- Authentication only possible via Azure AD Oauth2 with a service principal (or Managed Identity) from our AAD Tenant.
- Outbound traffic from the Logic App exits onto our Vnet to reach AAP Controller gateway

1. VM Build Triggers Ansible Base Playbook (Includes pam_admin role)
2. Admin user amends local Administrators group membership
3. VM Agent Sends Security Events (4732, 4733) via SSL
4. Logic App triggered by change to Administrators group (that wasn't made by Ansible)
   Calls additional VNET integrated Logic App
5. Logic App calls Ansible API
   Role:pam_admin
   Target: Event source VM

In the POC, it was established that because the SOC subscriptions are isolated (they have no networking), it would not be possible to call the Ansible API directly from a logic app that resides within the SOC subscriptions as it would come from a public Azure IP address, and would not be permitted through the Azure hosted Palo Alto firewalls into our network, and then through to the AAP controller subnet.

To address this, a number of options were considered:

Option	Description	Pros	Cons
1	Enable Public Access to AAP SaaS instance		Insecure
2	Enable VNET Peering for SOC subscriptions	Enables other future integrations from Sentinel to other on-premise resources and resources with private IP addresses within our tenant	Cost Time to Implement
3	Use Logic App in SOC to trigger a Logic App in SEC subscription that is VNET Integrated	Low cost Time to implement	Slightly more effort to secure

During POC, general agreement within Cloud Engineering, Cloud DG&A and Enterprise Security was to opt for option 3 to enable the POC to complete and to allow faster delivery of the overall solution, which will deliver significant reductions in time to deliver and access VMs as part of application migrations.

Option 2 requires further conversations with Security Operations to cover off operational, cost and security implications of adding VNETs to SOC subscriptions.

Option 3 had some further sub-options that were considered:

Option	Description	Pros	Cons	List Price
3a	Standard (ie not Consumption) VNET Integrated Logic App with App Service Plan	Relatively inexpensive	Apps only accessible publicly via enabling Public Access and then adding mitigating controls such as Service Tag restriction and strong authentication (AAD OAuth2) Requires an empty subnet dedicated to the ASP, though can use small subnets (/29, for example).	For **Production Workflow Standard 1** SKU (WS1) (Linux option may be ~50% cheaper - requires view from SecOps engineering teams as to whether Windows features needed, otherwise, use Linux. Also consider premium v3 plans which allow for reserved instances (1 and 3 year plans) for further savings of ~43%. Monthly Charges £124.313 GBP (PAYG) Price Per Call Standard Connector £0.000101 Enterprise Connector £0.000802 (Number of calls may be different than number of action executions) Data retention: £0.10 GB/month
3b	VNET Integrated App Service Environment to host Standard Logic App - Public IP Enabled for Logic App	Use of an ASE promotes economies of scale - adding additional Logic Apps, Function Apps, Web Apps to the ASE is more cost effective than accumulating single use resources. ASEs can be built with smaller SKUs and can be configured for scale out /up as needed. Can move to other plans, including reserved instance plans, without major redesign - logic app, function app, app service code is portable.	More expensive than a single standalone logic app and App Service Plan. Apps only accessible publicly via enabling Public Access and then adding mitigating controls such as Service Tag restriction and strong authentication (AAD OAuth2) Requires an empty subnet dedicated to the ASE, though can use small subnets (/29, for example).	Monthly Charges £333.708/month (PAYG) **DOES NOT** include Logic App/Function App execution costs (Linux option may be ~50% cheaper - requires view from SecOps engineering teams as to whether Windows features needed, otherwise, use Linux. Also consider premium v3 plans which allow for reserved instances (1 and 3 year plans) for further savings of ~43%.
3c	VNET Integrated App Service Environment to host Standard Logic App - fronted by an API Management (APIM) instance	Use of APIM enables straightforward use of Managed Identities to authenticate to APIs behind it from Azure resources (still AAD OAuth2 but the service principle is assigned to the resource, and is fully managed by Microsoft (fully passwordless authentication). With use of APIM, the logic app can be setup to restrict access to only the APIM Instance IP, with the public facing endpoint being an APIM frontend IP.	Most expensive of these sub-options - requires at least Premium SKU for VNET integration. Strategic tooling for API management is Apigee Edge. Apigee Edge retirement has been announced for 2025 - review is underway to consider replacements, which may include Apigee X, Azure API Management and other solutions - therefore opting for APIM before such a review concludes may lead to later rework to migrate from APIM to another solution.	Monthly Charges £2,241.70/month (PAYG) Unit cost of incremental units (>1) of the same service instance charged at 50% of the first unit purchased. **DOES NOT** include App Service Plan and Logic App /Function App execution costs

Option 3a was selected for the POC and is being recommended for the production implementation due to:

- Minimal cost.
- Minimal complexity.
- Time to implement.
- Unknown future strategic API Management tooling decision.
- Unknown direction for security automation to leverage costlier but more capable options (ASE, APIM).
- Ease of migration to Azure alternatives later if required.
- Security of 3a using network restrictions that block all public traffic unless it is from Azure, and is from LogicApps service tagged IPs, and only allowing authentication from an AAD service principle in our corporate AAD tenant, is deemed secure enough given the data classification (only data being sent is server hostnames, AD users/groups, all traffic is encrypted https).

Migration

To migrate to the new solution for all cloud VMs in all domains the process is:

1. Deploy additional prod Azure resources (Logic Apps, Sentinel rules, KeyVault, subnet).
2. Extract Administrators group settings from existing PAM GPOs in all Cloud OUs in all domains (scripted via PowerShell)
3. Extract membership of all relevant machine groups linked to the above GPOs, in all domains
4. For Exception group membership, audit the admins on those servers, for all appropriate access, incorporate in the new ansible pam_role_ groups, communicate any admin removals to server owners.
5. Define the pam_role_ lists in the pam_admin role variable definition file, each list matching an existing policy definition and list of admins
6. Commit the above change, merge into main branch, run automation pipeline, approve the update in the Ansible Automation Hub
7. Establish the required Resource Graph queries to match the same VMs as are members of the associated machine groups
8. Update the prod, nonprod and rnd Dynamic Inventories files in the app configuration repo, commit, merge into main branch, run the automation pipeline, resync the inventories.
9. Enable Sentinel monitoring
10. Amend the PAM base GPOs for Cloud OUs so that they no longer enforce local administrator membership and only apply a base set of accounts, if they are missing (one of which is the service account used by Ansible to connect to the VM over WinRM).
11. Remove VMs from machine groups for existing cloud OU PAM GPOs
12. Remove/unlink remaining cloud OU PAM GPOs and related machine groups, leaving only the COMP_SRV_CORE_CYB_Cloud_BASE_SEC GPO in each domains Cloud OU.
13. Update PAM GPO documentation on Confluence.

11.3 Design Principles

Describe the design principles that influenced the design.

Design Decision	Rationale
Use Ansible	Strategic configuration management tooling. Can use OOTB vendor supported plugins rather than custom code. Minimises operation cost and effort - no need to build out e.g. DSC infrastructure. Adheres to cost and complexity requirements and also SaaS first.
Don't add VNETs and peering to SOC subscriptions	SOC subscriptions were designed as isolated subscriptions for a reason. Decision made here for this solution doesn't preclude a later decision to add VNETs and peering. Doing so would allow some simplification to this solution by removing the double-hop (instead use just 1 logic app, in SOC subscription) with minor changes to achieve that.
Use Standard VNET Integrated Logic app with App Service Plan	Less expensive than alternatives, whilst not accruing technical debt.
Focus solely on cloud VMs	Largest pain point in terms of change runway is Lansing Exit - changes to local administrators on on-premise servers is fairly static.

Ensure that solution is extensible to on-premise servers and also multi-cloud if so desired in future	Rather than implement LDAP connectors to allow Ansible inventory to discover on-premise servers, decision made to stick with Azure Resource Graph as inventory source. When Cortex XDR to Microsoft Defender for Endpoint (MDE) migration completes, all on-premise servers will be Hybrid Joined (MDE requires it). At that point, they can be surfaced through Azure Arc if required, and therefore visible in the Azure Resource Graph (with a provider type of *Microsoft. HybridCompute/machines* rather than *Microsoft.Compute/virtualmachines*). Also - due to Lansing Exit, plus Cloud-Only objectives, on-premise server estate will shrink significantly, with any that need to remain on-premise expected to be managed as a cloud server would, which Arc enables. Use of Resource Graph for inventory, alongside Arc, also provides capability to extend this solution to VMs in other Cloud Service Providers if desired at a later date.

11.4 Application

N/A - no application is being developed/implemented.

11.5 Network

Additional Subnets:

1x new /29 subnet proposed in vnet-sec-prd-t1-weu-01 (10.223.27.0/29)

1x Azure Hosted Palo Alto firewall rulebase change to permit SSL from the above new subnet to:

Name	Address range	Virtual network	Service
snet-aapxpxb5efguu5ni-westeurope-aks	10.223.43.0/26	vnet-aapxpxb5efguu5ni-westeurope	SSL

11.6 Security

Describe all the security design considerations here covering:

- Certificates: N/A
- Keys: AAP API Key (Client ID & Secret) - to be stored in new Azure KeyVault in rg-sec-prd-t1-weu-sec-auto-01
- Firewalls: Azure Hosted Palo Alto (see 11.5)
- API: Ansible Automation Platform - plus HTTP trigger for Logic App
- Whitelisting: N/A
- SSO: N/A
- Integration Parameters: N/A
- AD Group based ACL on Firewall: N/A
- IPS (Internet Traffic): N/A
- Traps: N/A
- AV Exclusions: N/A

If not applicable say N/A

Azure DevOps

Organisation	Project	Repository /Pipeline Name	Purpose	Approvers Groups	Minimum Approval	AD Group	ARC Role
mgpru	Ansible	repo collection_windows_pam	Contains pam role - vars file must be updated to add new pam role groups and associated administrators group members	Cloud Platform Security Approvers (Cloud Engineering & Cloud SRE) Cloud Enterprise Security Approvers (Security Engineering & Identity Specialists)	1 Member of each group must approve for merge into main branch	TBC	TBC
mgpru	Ansible	repo playbook_windows_pam	Contains playbook for pam role	Cloud Platform Security Approvers (Cloud Engineering & Cloud SRE) Cloud Enterprise Security Approvers (Security Engineering & Identity Specialists)	1 Member of each group must approve for merge into main branch	TBC	TBC
mgpru	Ansible	repo aap_iac_configure_as_code	Contains AAP config - needs updated to update dynamic inventories for new/amended conditional inventory groups for new pam roles to target correct hosts	Cloud Platform Security Approvers (Cloud Engineering & Cloud SRE) Cloud Enterprise Security Approvers (Security Engineering & Identity Specialists)	1 Member of each group must approve for merge into main branch	TBC	TBC
mgpru	Ansible	pipeline aap_iac_configure_as_code	Updates AAP configuration when main branch of repo is updated	Cloud Platform Security Approvers (Cloud Engineering & Cloud SRE) Cloud Enterprise Security Approvers (Security Engineering & Identity Specialists)	1 Member of each group must approve pipeline run	TBC	TBC
mgpru	Ansible	pipeline aap_AutomationHub_CI	Updates collections and playbooks configuration when main branch of repo is updated	Cloud Platform Security Approvers (Cloud Engineering & Cloud SRE) Cloud Enterprise Security Approvers (Security Engineering & Identity Specialists)	1 Member of each group must approve pipeline run	TBC	TBC

11.7 Server Deployment

N/A

11.8 Data & Privacy

N/A - no storage of sensitive data. Authentication secrets vaulted in Azure KeyVault, with appropriate access control and network access controls

11.9 Logging & Monitoring

N/A

11.10 Environment Info

N/A - Only prod environment given the nature of the solution.

11.11 Storage and Backups

N/A - All components are SaaS/PaaS

11.12 Resilience

N/A - All components are SaaS/PaaS

11.13 Business Continuity and Disaster Recovery (BCDR)

N/A - All components are SaaS/PaaS

12 Licensing and Costs

See infrastructure section for costs.

13 Ownership

State the following details. These details are used to onboard applications to Application Portfolio Management (APM) tool in ServiceNow by Enterprise Architecture team.

- *Business Owner: Enterprise Security & Privacy - Security Operations*
- *Technology Product Owner (TPO): Katy Hinchcliffe*

If not applicable say N/A and state reasons why it is not applicable..

14 Operate and Support Model

All components supported by Cloud Platform Services except Sentinel and Logic App in SOC subscription - supported by Enterprise Security & Privacy - Security Operations

15 Capacity and Performance

N/A

16 Appendix

Acronym	Definition

16.1 Additional Info

Add any relevant info that helps a reader understand the design