



PRIFYSGOL CYMRU
Y Drindod Dewi Sant
UNIVERSITY OF WALES
Trinity Saint David
SWANSEA - ABERTAWE



Microsoft Azure Arc Implementation in an Enterprise Environment

David Corner (1809784)

Dr. Nitheesh Kaliyamurthy

Project submitted as part of the requirements
for the award of BSc Cyber Security and
Advanced Networking

April 2025

Declaration of Originality

I, David Corner declare that I am the sole author of this Project; that all references cited have been consulted; that I have conducted all work of which this is a record, and that the finished work lies within the prescribed word limits.

This work has not previously been accepted as part of any other degree submission.

Signed : D. J. Corner

Date : 27/04/2025

FORM OF CONSENT

I David Corner hereby consent that my Project, submitted in candidature for BSc Cyber Security and Advanced Networking, if successful, may be made available for inter-library loan or photocopying (subject to the law of copyright), and that the title and abstract may be made available to outside organisations.

Signed : D. J. Corner

Date : 27/04/2025

Copyright Acknowledgement

I acknowledge that the copyright of this project report, and any product developed as part of the project, belong to University of Wales Trinity Saint David, Swansea.

Abstract/Synopsis

This project looked at how Microsoft Azure Arc can be used to manage on-premises Windows servers without having to move everything into the cloud. The main goal was to connect four Windows Server 2016 and 2019 machines to Azure Arc, set up monitoring through a Log Analytics workspace, and apply governance policies to keep everything consistent and secure.

During the setup a few problems came up such as network firewall blocks and time sync issues, but these were sorted out without causing major delays. Once the servers were onboarded, policies and alerts were configured successfully, showing that Azure Arc can make hybrid server management much simpler, even when dealing with older infrastructure.

Overall the project showed that Azure Arc is a practical way to extend cloud management tools to on-premises environments. All objectives were met and the environment was running smoothly by the end. Future improvements could include adding more policies, scaling to larger setups and using automation to make management even more efficient.

Glossary

Term	Definition
Azure Arc	A Microsoft service that allows on-premises and multi-cloud resources to be managed through the Azure portal.
Azure Policy	A service in Azure used to enforce governance rules and compliance across resources.
Log Analytics	A monitoring service in Azure that collects and analyzes performance and diagnostic data from resources.
Hybrid Cloud	A computing environment that combines on-premises data centers with public or private cloud services.
Connected Machine Agent	The agent software installed on servers to allow them to be onboarded into Azure Arc.
Governance	The process of setting policies, standards and procedures to manage IT systems effectively.

Table of Contents

Declaration of Originality.....	2
Copyright Acknowledgement	2
Abstract/Synopsis	3
Glossary	4
Chapter 1	8
1.1 Introduction	8
1.2 Aim.....	10
1.3 Objectives	10
1.4 Structure of the Project	11
Chapter 2 - Research & Review of Literature	12
2.1 Introduction	12
2.2 Literature Review Types.....	12
2.2.1 Narrative Literature Review	12
2.2.2 Systematic Literature Review	13
2.3 Search Strategy	13
2.3.1 Keywords	13
2.3.2 Sources and Databases	14
2.3.3 Boolean Search Strategies	14
2.4 Inclusion and Exclusion Criteria	15
2.5 Search Process	17
2.6 Categorisation of Literature	17
2.7 Addressing Each Category	18
2.7.1 Azure Arc Overview and Capabilities.....	18
2.7.2 Onboarding On-Premises Servers.....	18
2.7.3 Policy Enforcement with Azure Policy	19
2.7.4 Monitoring and Alerts through Azure Monitor	20
2.7.5 Security and Compliance in Hybrid Environments	20
2.8 Research Gaps.....	21
2.9 Conclusion	21
Chapter 3 - Research Methodology	22

3.1 Introduction	22
3.2 Research Overview	22
3.2.1 Research Process	22
3.2.2 Research Philosophy	25
3.2.3 Approaches	25
3.2.4 Strategies	25
3.2.5 Choices	26
3.2.6 Time Horizons	26
3.2.7 Techniques and procedures	26
3.2.8 Contingency planning	27
3.2.9 Project Planning	29
3.3 Research Methods	30
3.4 Data Collection	31
3.5 Conclusion	31
Chapter 4 - Design and Implementation	33
4.1 Introduction	33
4.2 Problem formulation	33
4.2.2 Defining the Problem	34
4.2.3 Why it matters	34
4.2.4 Scope and Focus	35
4.2.5 Project Goals	35
4.3 Design	36
4.4 Implementation	36
4.5 Azure Arc Onboarding	39
4.6 Monitoring and Alerts	44
4.6.2 Azure Alerts	46
4.7 Azure Policy Setup	47
4.7.1 Checking the Time Zone	47
4.7.2 Sending Activity Logs to Log Analytics	48
4.8 Challenges Encountered	48
4.9 Results and Validation	49
4.10 Conclusion	50

Chapter 5 - Evaluation and Testing	51
5.1 Introduction	51
5.2 Checking Azure Arc Connections	51
5.3 Checking Monitoring Through Log Analytics	52
5.4 Checking Azure Policy Compliance	54
5.4.1 Time Zone Policy	54
5.4.2 Activity Log Streaming Policy	56
5.4 Azure Monitor Alerts	59
5.5 Summary	61
Chapter 6 - Conclusions and Recommendations	63
6.1 Summary	63
6.2 Conclusion	63
6.3 Recommendations for Further Research	64
Chapter 7 - Reflections	65
7.1 Introduction	65
7.2 Research Aim	65
7.3 Objectives and Reflections	65
7.4 Reflections on Conduct and Progress	66
7.5 Conclusion	67
Bibliography	68

Chapter 1

1.1 Introduction

This chapter provides background information on the research topic, focusing on the implementation of Microsoft Azure Arc in an enterprise environment. The project aims to enhance server management by integrating cloud management technology for on-premises, edge and multi-cloud environments into a centralized solution. Currently, the organisation utilizes VMware for managing its entire on-premises server infrastructure and SolarWinds for event monitoring. A direct migration to a fully cloud-based server environment isn't feasible at this time due to legacy systems that rely on the existing setup. Implementing Azure Arc will enable system administrators to manage and monitor all internal infrastructure over the internet effectively, as if they were fully cloud-hosted. This includes leveraging Microsoft's conditional access and Defender security features from anywhere at any time.

What is Microsoft Azure Arc? It is a hybrid and multi-cloud management platform that extends Azure services and management capabilities to infrastructure beyond the Azure environment. It allows organisations to project their on-premises servers, Kubernetes clusters and other resources into Azure, enabling management of these assets through the Azure Portal as if they were native Azure resources. This unified approach facilitates consistent governance, security and compliance across diverse environments, including on-premises datacenters, edge locations and other cloud platforms [10].

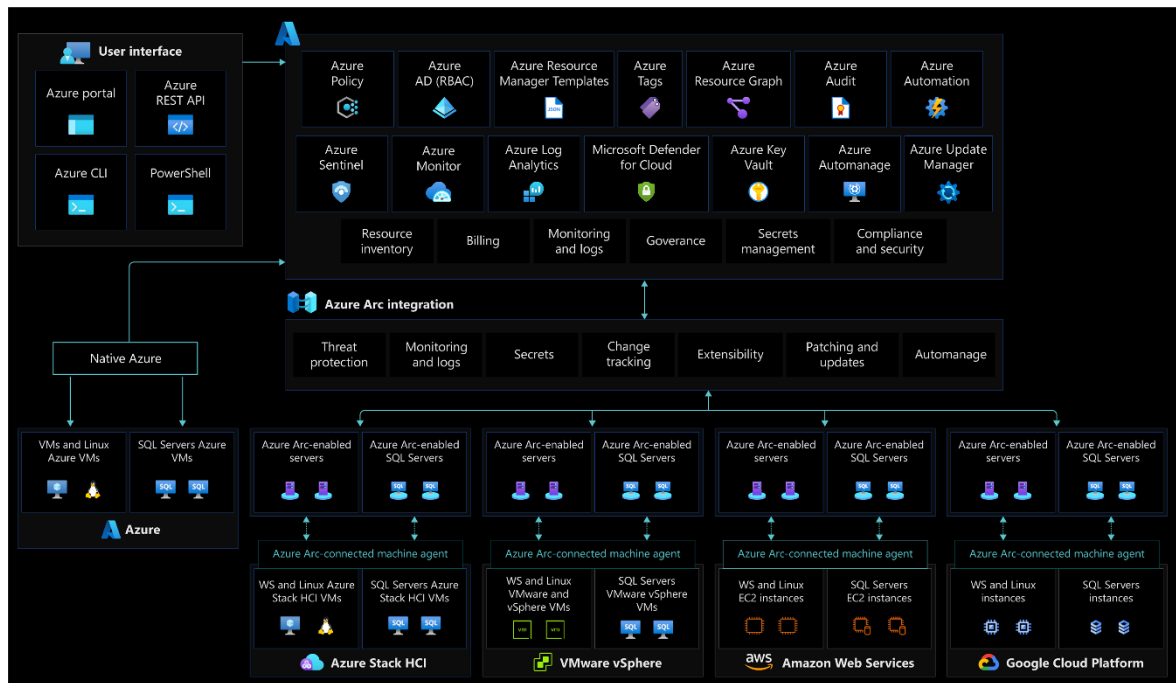


Figure 1 – Azure Arc Architecture (Source: Microsoft Learn, "Monitor Azure Arc-enabled servers with Azure Monitor," Microsoft, Jan. 2024 [14]).

This report also entails reviewing documentation, adopting suitable methodologies, onboarding servers, implementing policies, creating log analytics workspaces and configuring alerts. The literature review will compare methodologies, identify research gaps and outline best practices for Azure Arc implementation. By following these approaches, the project aims to deliver a comprehensive solution, offering insights into the successful implementation of Azure Arc within diverse environments.

1.2 Aim

To enhance the organisation' server infrastructure management by securely implementing Azure Arc, enabling effective real-time analysis, alerting and manageability over the internet anywhere at any time.

1.3 Objectives

This report will consist of the following objectives:

- Carry out a literature review on Microsoft Azure Arc technology
- Select a suitable methodology that will be used for project management and delivery.
- Configure and deploy the Azure Arc connected machine agent, log analytics workspaces, policies and alerting on the required hosts.
- Carry out evaluation and testing to ensure successful implementation.

1.4 Structure of the Project

The project is essentially divided into three parts. Firstly, the theoretical work that underpins the project, which involves reviewing literature and former studies on Azure Arc implementation. This theoretical work will expand the knowledge base and be fundamental to the project's outcome.

The second part details the design and implementation. This section outlines the problem formulation and the design implemented to integrate Azure Arc technology within an enterprise environment. It covers the onboarding of servers, installation of the Arc connected machine agent, creation of baseline policies and setup of log analytics workspaces and dashboards. The software and hardware required to complete this project are also covered here.

Finally, the third part details the findings. This section will analyse the performance, security and management efficiencies achieved through the implementation of Azure Arc. It will include a summary of the previous chapters, a conclusion and outlines of any future work or development.

- Chapter one delivers a background summary of the project, specifying the aim and emphasizing the objectives necessary for a successful outcome.
- Chapter two concentrates on the literature review, comparing systematic and narrative literature review types along with why narrative was chosen, identifying research gaps and outlining best practices for Azure Arc implementation.
- Chapter three details the research approach and design, including the practical steps for onboarding servers, implementing policies, setting up log analytics workspaces and alerts.
- Chapter four outlines the execution of the fieldwork, including the installation of the Arc connected machine agent, creation of baseline policies, log analytics workspaces and alerts.
- Chapter five examines the findings, analysing the performance and management efficiencies achieved through the implementation of Azure Arc.
- Chapter six focuses on the conclusion, summarizing the previous chapters, providing a conclusion and outlining any future work or development.

Chapter 2 - Research & Review of Literature

2.1 Introduction

This chapter explores existing research and technical writings around Microsoft Azure Arc, particularly its use in managing hybrid cloud environments within businesses. The aim is to review key sources critically, picking out patterns, benefits and challenges that directly shaped the project's practical work.

Originally, a systematic literature review was considered. However, Azure Arc is still quite new and there isn't much formal academic material available yet. Because of this, a narrative review approach made more sense, allowing for a broader search across technical documentation, case studies and industry insights.

Hybrid cloud has become a major focus for organisations looking to stay flexible without losing control over governance or security [1]. Azure Arc fits into this evolving landscape by bridging on-premises and cloud environments. Since the technology and its use cases are still developing rapidly, it was important for this research to include both theoretical ideas and real-world deployment experiences.

2.2 Literature Review Types

2.2.1 Narrative Literature Review

At the start of the project, two options for reviewing the literature were considered: a systematic review or a narrative review.

Narrative reviews work particularly well for new areas where formal studies are limited [2]. They let researchers draw on a wider mix of sources, including whitepapers, case studies and expert blogs, which was crucial for a topic like Azure Arc. It also made it easier to spot real-world challenges and patterns across different industries [3]. Given the hands-on nature of this project, the narrative review approach was a better fit than a purely theoretical one.

2.2.2 Systematic Literature Review

Systematic reviews tend to suit more mature research fields where there's already a lot of published academic work [4]. They follow a strict process, using detailed inclusion and exclusion criteria to avoid bias.

While valuable, this method can be quite rigid and isn't always ideal when exploring newer technologies like Azure Arc, where some of the best insights come from industry experts and technical blogs [5]. Considering the early stage of Azure Arc's adoption, sticking to a systematic review would have missed out on key practical knowledge.

2.3 Search Strategy

2.3.1 Keywords

To keep the search targeted, key terms were selected based on the project's focus areas. These included:

- "Azure Arc"
- "Hybrid cloud management"
- "On-premises server onboarding"
- "Azure Policy"
- "Azure Monitor alerts"
- "Governance"
- "Cloud security"

Using a set of keywords like this helped pull in literature that was directly relevant [6].

2.3.2 Sources and Databases

Sources were selected if they discussed Azure Arc onboarding, hybrid governance, monitoring, or cloud security. Searches were run across a range of databases:

- IEEE Xplore
- ACM Digital Library
- ScienceDirect
- SpringerLink
- Google Scholar

Because the academic material around Azure Arc is still catching up, credible industry sources like Microsoft Learn, Gartner reports and respected tech blogs were also brought in [7].

2.3.3 Boolean Search Strategies

Boolean operators made it easier to refine the results, using combinations like:

- "Azure Arc" AND "on-premises management"
- "Azure Arc" AND "Azure Policy" AND "governance"
- "Hybrid cloud" AND "Azure Monitor" AND "alerts" [8]

This helped balance precision with a wide enough scope to capture relevant studies.

2.4 Inclusion and Exclusion Criteria

To make sure the literature collected for this project stayed focused and relevant, a set of inclusion and exclusion rules was applied, that helped steer the search towards material that was both useful for the project's aims and credible enough to rely on.

Inclusion Criteria:

- Sources published from 2019 onwards, covering Azure Arc's introduction and development
- Materials that focused on Azure Arc onboarding, policy management, monitoring, or hybrid cloud security
- Peer-reviewed journal articles, Microsoft official documentation, industry whitepapers and reputable technical blogs

Exclusion Criteria:

- Articles that only talked about general Azure services without linking to Azure Arc
- Non-English language sources
- Material without strong technical backing, like opinion blogs without clear authorship

Since Azure Arc is still quite a new topic, the search followed a narrative review approach, that made it possible to include a broader mix of sources, especially where peer-reviewed academic studies were hard to find.

The search and filtering process is summarised in Figure 2 below:

Step	Description
Initial Search	Around 60 sources identified across databases and technical sites.
Title and Abstract Screening	35 sources excluded based on relevance, date, or scope.
Full Text Screening	7 more excluded due to lack of focus on Azure Arc or hybrid governance.
Final Literature Pool	18 sources selected (1 academic, 17 technical/industry reports).

Figure 2 - Summarisation of literature searches (Author: David Corner).

When searching through databases like IEEE Xplore, ACM Digital Library, ScienceDirect and SpringerLink it quickly became clear that there was only one academic paper directly focused on Azure Arc. This wasn't too surprising given how new the technology currently is. Since there wasn't enough formal research to rely on, the review also heavily drew on grey literature such as Microsoft's own documentation, industry reports and trusted technical blogs, to build a fuller and more practical view. This fitted well with the narrative review approach which is designed to include a broader range of credible sources when academic studies are still limited.

2.5 Search Process

The initial searches produced about results. After screening titles, abstracts and full papers, 11 sources were selected as highly relevant.

This included one peer-reviewed academic paper and ten technical or industry sources. It quickly became clear that formal academic research into Azure Arc remains limited, making technical and industry material essential for building a realistic understanding [10].

2.6 Categorisation of Literature

To make the findings easier to work with, the literature was grouped into five major themes:

- Azure Arc Overview and Capabilities
- Onboarding On-Premises Servers
- Policy Enforcement with Azure Policy
- Monitoring and Alerts through Azure Monitor
- Security and Compliance in Hybrid Cloud Environments [11]

This structure helped keep the review organised and tied closely to the project's technical focus.

2.7 Addressing Each Category

2.7.1 Azure Arc Overview and Capabilities

Aravind [12] describes Azure Arc as a way to connect traditional data centers with the cloud, allowing for Azure-native management across diverse environments.

Krishnan [6] highlights Arc's strength in creating a single governance layer for multi-cloud and on-premises setups but Buyya et al. [13], warn that adding abstraction layers can introduce extra complexity.

Gartner [14] compares Azure Arc to alternatives like AWS Outposts and Google Anthos noting that Arc focuses more on extending governance, whereas others aim to replicate full cloud environments locally.

On the other hand, Arc's model depends heavily on reliable connections to Azure Resource Manager - something that isn't always practical in tightly locked-down networks [15].

2.7.2 Onboarding On-Premises Servers

Mishra et al. [15] offer a clear explanation of the onboarding process using Azure's Connected Machine agent.

Real-world challenges are discussed by Krishnamurthy et al. [9], who point out issues like firewall restrictions, proxy settings and agent authentication failures that aren't always obvious in official documentation.

Microsoft's own guidance [11] admits that network policies and Active Directory complexity can cause onboarding failures.

While the standard onboarding steps [17] are technically correct, many practical issues only emerge during real deployments, suggesting that operational experience plays a big role in successful rollouts.

2.7.3 Policy Enforcement with Azure Policy

Schutten [16] highlights how Azure Policies can extend compliance monitoring to non-Azure resources through Arc.

However, Zhang et al. [7] warn about a "policy drift," where local changes go unnoticed and potentially undermine compliance.

Hill [15] recommends regular compliance scans and automatic remediation policies to keep governance tight.

According to Gartner [14], automated compliance enforcement will likely become standard within a few years, which supports the need for proactive policy management in hybrid environments.

However, Arc's built-in policy sets often need heavy customisation to meet sector-specific standards like PCI DSS and HIPAA [8].

2.7.4 Monitoring and Alerts through Azure Monitor

Microsoft's documentation [17] provides guidance for setting up monitoring of basic metrics like CPU, memory and server health.

However, Erl et al. [13] argue that for more complex hybrid environments, Azure Monitor alone might not provide deep enough observability.

Gartner [14] reports that many organisations are using third-party tools like Datadog, Dynatrace and Splunk alongside Azure Monitor to get more complete visibility.

Krishnamurthy et al. [9] also point out that when servers are geographically distant from Azure regions, there can be delays in telemetry updates and alerts.

2.7.5 Security and Compliance in Hybrid Environments

Microsoft's governance baseline [18] offers strong starting points for securing Arc-connected servers.

However, Kim et al. [8] highlight that hybrid architectures introduce new vulnerabilities - especially around lateral movement between cloud and on-premises assets.

Grozev and Buyya [13] argue that traditional perimeter-based security models need to be replaced by resource-centric, zero-trust strategies to better handle hybrid threats.

Meeting strict compliance standards like GDPR often requires additional policies and custom audits beyond what Microsoft's default templates provide [17].

2.8 Research Gaps

Despite the growing amount of guidance around Azure Arc, several important gaps remain:

- **Onboarding Complexity:** Many guides assume simple network environments and don't address real-world obstacles [9][11].
- **Scaling Policy Management:** How Azure Policy operates at large scale (across hundreds or thousands of resources) hasn't been widely studied [7][15].
- **Hybrid Observability:** There's limited research into integrating Arc with third-party monitoring tools [13][14].
- **Compliance Customisation:** Few sources explain how to tailor Arc's governance templates to strict industry regulations [8][18].

These gaps show the importance of a more practical study and real-world testing to better understand how Azure Arc behaves under complex conditions.

2.9 Conclusion

Overall, the literature reviewed provides important insights into Azure Arc's potential to simplify hybrid governance and monitoring. At the same time, it highlights the gap between theoretical documentation and the reality organisations face when deploying hybrid solutions. The real-world issues around onboarding, scaling policies, monitoring limitations and compliance challenges show that further practical research is needed.

This dissertation aims to address some of these gaps by evaluating Azure Arc's deployment and governance features through hands-on implementation.

Chapter 3 - Research Methodology

3.1 Introduction

This chapter explains how the research for this project was planned and carried out, including the overall design, chosen philosophy, methods of data collection and how each part supported the project's goals. The research focused on implementing Microsoft Azure Arc in an on-premises enterprise setting and aimed to take a well-structured approach while allowing flexibility as new insights came up.

The aim here is to show why the selected research methods made sense for this project. As the work progressed, small changes were made to the design where needed, helping to keep everything aligned with the main objectives and ensuring the methods stayed relevant and effective.

3.2 Research Overview

To make sure the Azure Arc implementation met its intended goals, a carefully thought-out process was followed where this section breaks down how the research was carried out, from the approach and philosophy used to how information was gathered.

3.2.1 Research Process

To help guide the structure of this project, the Research Onion model by Saunders, Lewis and Thornhill was used. This model breaks down the research process into different layers, starting from the wider philosophical approach and narrowing down to the methods used for collecting data. This helped make sure the choices made during the research were well thought out and that each step connected clearly to the overall aims of the project.

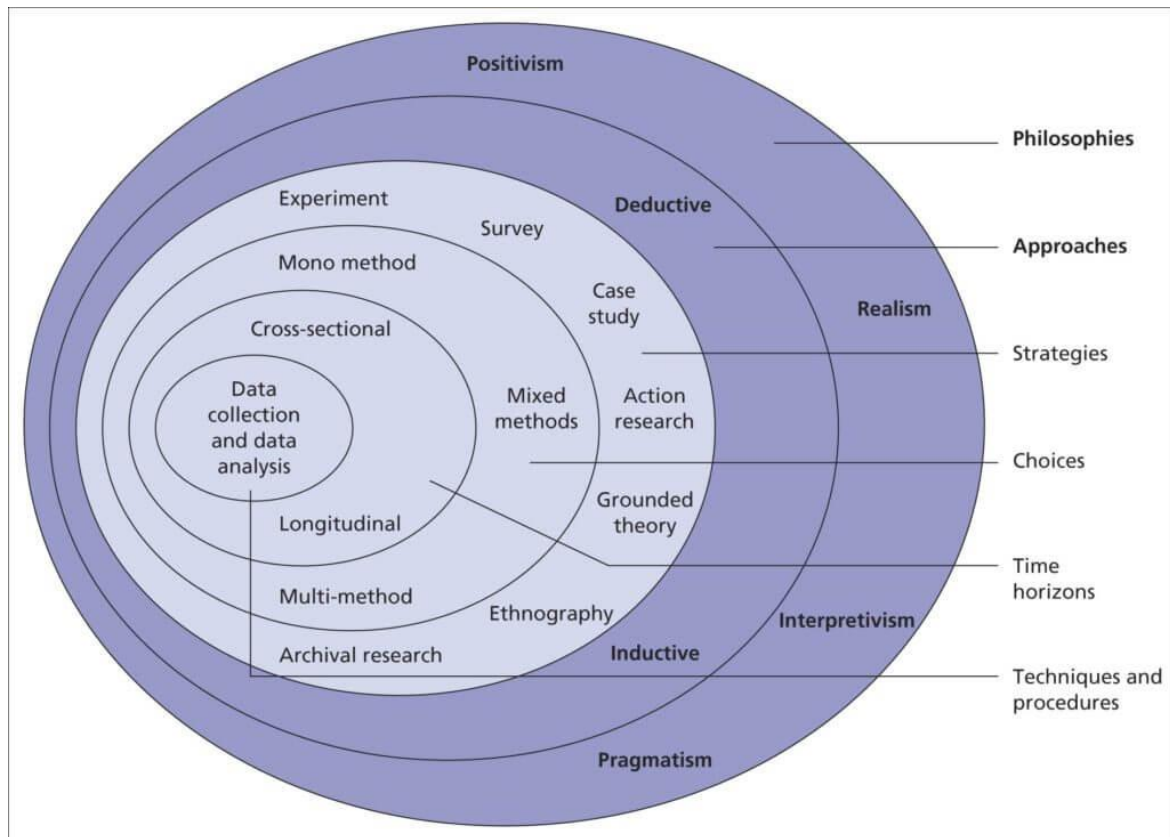


Figure 3 - The Research Onion model (Source: Adapted from M. Saunders, P. Lewis, and A. Thornhill, Research Methods for Business Students, 5th ed., Pearson Education, 2007 [19])

The Research Onion model, developed by Saunders et al., provides a structured framework for designing research methodology [19]. It comprises six layers: research philosophy, research approach, research strategy, methodological choice, time horizon and techniques and procedures. This model guides researchers through the process of developing a coherent and well-structured research design by systematically considering each layer from the outermost to the innermost.

The model was useful in showing how each part of the research fits together. For this project, a few layers were especially important:

- **Philosophy - Positivism**

A positivist view was taken because the focus was on collecting real, measurable data. Since the project looked at things like server performance, alerts and system monitoring, this approach made the most sense as it allowed the results to be based on hard evidence rather than opinions.

- **Approach - Deductive**

A deductive approach was chosen because the research started with clear goals and then tested them during the implementation. This worked well as it allowed ideas from the literature to be applied in practice to see what actually worked.

- **Strategy - Document Review and Hands-On Testing**

The research involved reviewing technical documents and case studies to understand how Azure Arc works and then carrying out the actual implementation in a test setup. This helped connect what was learned in theory with how the technology performs in real scenarios.

- **Techniques - Structured Tasks and Monitoring**

For every task, from setting up the servers to configuring alerts, this was carried out in a planned, step-by-step way. This helped make sure the process was consistent and that the data collected was accurate and useful.

Using the Research Onion gave the project a clear path to follow, but it also allowed room to adapt if needed. It helped make sure each part of the research connected properly to the next and that the overall design stayed on track with what the project aimed to achieve.

3.2.2 Research Philosophy

The positivist philosophy was chosen because it focuses on evidence-based outcomes. Since the project involved testing Azure Arc's ability to monitor and manage on-premises servers, having measurable results was essential.

This approach made it easier to focus on specific data points like CPU usage, heartbeats and alert notifications. Using real-time monitoring tools helped make sure the results were based on actual system performance, which was important for understanding how well Azure Arc worked in a practical setting.

3.2.3 Approaches

A deductive approach will be carried out, this is based on specific hypotheses about the effectiveness of Azure Arc through testing and observations. By starting with the projects aims and objectives the deductive approach allows data validation to be carried out effectively against Azure Arcs features. This consists of server onboarding results and real-time server polling data within log analytics workspaces using KQL queries.

3.2.4 Strategies

The case study strategy was adopted to explore Azure Arc' implementation within a controlled enterprise environment. This study helped provide detailed examination results of the following:

- The enrolment and onboarding procedures for existing servers using the Arc Connected Machine Agent.
- The build and configuration of the log analytics workspaces along with performance monitoring workspaces and alerting.
- Comparisons between traditional SNMP monitoring products such as Solarwinds whilst highlighting the benefits of adopting Azure Arc for remote management.

Using a case study approach on this project, created scope for a detailed analysis of Azure Arc' practical applications whilst keeping the research detail relevant

3.2.5 Choices

The mono-method approach was chosen for this project, as it provides full focus on implementation and data collection through quantitative methods. This method ensures consistency whilst testing Arc' key functionalities in real use cases, that helps streamline the overall research processes.

3.2.6 Time Horizons

This project adopted a cross-sectional approach that evaluates the implementation and performance of Azure Arc within a set timeframe. Whilst keeping focus within the deployment phase during initial implementation, the research captured measurable results, this includes success rates of server onboarding, the initial system performance and additionally the effectiveness of event alerting. This approach highlighted areas of key insights into capabilities within Azure Arc whilst identifying potential room for improvement in other areas.

3.2.7 Techniques and procedures

The implementation process adhered to a clear and logical series of steps that would help ensure that everything ran smoothly and effectively. It started firstly with setting up an Azure Resource Group followed by a log analytics workspace that would be the foundations of where the server data is stored and inspected. Following this, servers were brought aboard by rolling out the connected machine agent to the existing servers so they can effectively be administered through the cloud management dashboard. Once the onboarded phase was successful, event alerts were then set up that would assist with keeping full visibility if in the event something needed critical attention. These alerts were configured using basic email forwarding rules that essentially sends an email to system administrators if a specific event has triggered. Once these notifications were created and functioning effectively, performance data was then pulled from Azure monitoring dashboards to demonstrate server overall health, response times and if any

administration tasks required that were carried out to the server in question remediated the issue.

Each stage in the procedures were recorded, allowing the process to easily be followed and demonstrated into practice if and where needed later. The procedure demonstrates how Azure arc can be implemented effectively to allow secure and efficient server management.

3.2.8 Contingency planning

To ensure the project remains on track and achieves its objectives a contingency plan has been developed. This plan should solely identify any potential risks that may arise during the implementation of Azure Arc, along with measures to help mitigate these risks and strategies to help monitor and address any unexpected challenges encountered.

Identified Risks

Timeline Delays - Unprecedented delays encountered such as issues with servers being onboarded effectively or the general configuration of resources could impede and create setbacks on the project delivery schedule

Resource constraints - Any limitations related to hardware availability, licensing requirements or insufficient training for the supported users could disrupt the progress of the project.

Technical challenges - Issues with compatibility when integrating Azure Arc with existing infrastructure may create additional obstacles during the implementation phase.

Mitigation Measures

To address the risks identified, the following strategies have been put in place:

- **Effective and Detailed Timelines** - Extra time was allocated for the critical tasks within the project that would help accommodate any unexpected delays and help keep the overall project schedule on track.
- **Personal Training** - Comprehensive training was provided to all stakeholders and users to ensure readiness to help minimise any disruptions.

- **Pilot Testing** - Any changes will be subject to pilot testing to ensure they meet desired expectations before implementing into full production

Monitoring Strategies

Regular monitoring was essential to track the projects progress and to help address any risks proactively:

- **Continuous feedback** - Feedback loops had been created to gather input from users and stakeholders to improve the system throughout its project lifecycle.
- **Frequent Reviews** - Regular reviews of the projects progress were conducted to ensure it continues to align with set goals and schedules.
- **Risk Assessments** - Regular risk assessments were performed to identify emerging risks and devise appropriate solutions to help overcome them.

By incorporating this contingency plan into the project, it helps anticipate any potential challenges that could arise, ensure effective proactive management, help minimize disruptions and help support a successful Azure Arc implementation successfully.

3.2.9 Project Planning

Gantt Chart

The following Gantt chart helps demonstrate a clear breakdown of the timelines of events and phases for the project. Every task is planned out within roughly 2-week intervals that focuses on each key milestone, this includes configuring required resources, onboarding the servers along with configuring monitoring alerts.

The purpose of the Gantt chart is to ensure that each stage is completed in order of dependencies before moving onto the next stage. As an example, onboarding or real-time server monitoring won't effectively operate until the underlying log analytics workspace is configured as there would be no place for the data to reside. By ensuring that all tasks are well structured, not only does it allow the project to remain organised to help hit its objectives without any unexpected delays, but helped stay on top of progression providing any room for adjustments where necessary as demonstrated in my contingency plan.

Phase 1 - Planning (Weeks 1-2)		Week 1					Week 2				
Activity		M	T	W	T	F	M	T	W	T	F
Assess the scope of the project and set goals.											
Look to bring in any stakeholders into the project team where required.											
Assess the stature of the infrastructure by carrying out a full assessment.											
Gather all requirements along with specifying criterion for projects success.											
Look at what resources are required to help deliver the project											
Put together a project plan detailing how the work is going be delivered further											
Build pilot environment for extensive testing for onboarded servers.											
Collate a list of servers that will be targeted as part of pilot testing.											
Plan/schedule training or development time in relation to Azure Arc technology.											

Figure 4 - Gantt chart table for planning phase (Screenshot by David Corner).

Phase 2 - Pre-Production Implementation (Weeks 3-4)		Week 3					Week 4				
Activity		M	T	W	T	F	M	T	W	T	F
Configure the initial compliance and connected machine-based policies											
Roll out the Arc connected machine agent to servers that are part of the pilot list.											
Thoroughly test server management functionality from console.											
Log any issues encountered whilst testing into a raid log where they will be remediated.											
Gather feedback from the infrastructure team on how they are finding managing the pilot servers											
Carry out further evaluations by aggregating security and performance metrics.											
Look to map out implementation to the production environment.											

Figure 5 - Gantt chart table for pre-production implementation phase (Screenshot by David Corner).

Phase 3 - Production Implementation (Weeks 5-8)		Week 5					Week 6					Week 7					Week 8				
Activity		M	T	W	T	F	M	T	W	T	F	M	T	W	T	F	M	T	W	T	F
Raise a high-change control detailing the plan of implementing to production servers																					
Schedule recurring meetings with stakeholders and the infrastructure team																					
Deploy the Arc connected machine agent to production server environment.																					
Gather telemetry on the deployment's progression.																					
Continue to monitor implementation progression.																					
Verify that the connected machine agent has successfully installed on all servers																					
Further testing to ensure nothing has been missed in terms of configuration policies																					
Consult with Cyber Security team to verify if any additional compliance policies are required																					
Update any documentation and guidance on the technologies use and update the project task lists																					
Consideration in providing further training to key users if required.																					

Figure 6 - Gantt chart table for Production implementation phase (Screenshot by David Corner).

3.3 Research Methods

To meet the project's objectives, two main research methods were used:

- **Document Review**

A variety of technical resources were reviewed, including Microsoft documentation, case studies and academic articles. These helped build a better understanding of Azure Arc's features, limitations and how others have used it in similar environments.

- **Hands-On Implementation**

Azure Arc was tested in a lab environment that mirrored enterprise conditions. Tasks included onboarding Windows servers, setting up Log Analytics workspaces, applying policies and configuring alerts. This helped test how well Azure Arc could be used to manage real systems.

Combining both methods allowed the research to stay grounded in proven concepts while also exploring how the technology worked in action.

3.4 Data Collection

Data collection focused on gathering reliable information to support the project's aims. This involved:

- **Testing and Monitoring**

By setting up Azure Arc tools and services, the project captured performance data and alert information. These were used to see how the platform handled real workloads.

- **Literature and Documentation**

Sources like IEEE Xplore, Microsoft Learn, Google Scholar and published case studies were used to gather background information. These materials helped fill in the bigger picture and backed up the findings from practical tests.

3.5 Conclusion

This chapter explained how the research for this project was designed and carried out, step by step. It started by using the Research Onion model to break down the different parts of the methodology—from the bigger picture, like the research philosophy, to the more detailed elements like the tools and procedures used.

A positivist philosophy was chosen because the focus of the project was on collecting clear, measurable data. This made sense considering the technical nature of Azure Arc, where things like server performance and alert data could be tracked and analysed. A deductive approach was used alongside this to allow the research to start with clear objectives and then test them through hands-on work.

The project followed a case study strategy, which helped take a closer look at how Azure Arc could be rolled out in a real enterprise environment. Combining this with a mono-method approach, which focused mainly on quantitative data, meant the project stayed consistent and results were easier to compare and understand.

A cross-sectional time horizon allowed everything to be assessed over a fixed period, particularly during the rollout and early monitoring stages. Setting up the systems such as onboarding servers, configuring alerts and tracking performance was done through a clear and structured process and each step was recorded so it could be followed again if needed.

Planning also included a contingency section to deal with any risks that came up such as delays or technical issues, with backup steps like extra training, timeline buffers and pilot testing. A Gantt chart was used to map out the key stages and make sure everything was done in the right order. This made it easier to stay on track and adjust things if needed along the way.

Altogether the methods used in this project helped make sure the research stayed organised, practical and focused on real outcomes. With a strong foundation now in place, the next chapter will go into the results and what they tell us about using Azure Arc in a real-world setup.

Chapter 4 - Design and Implementation

4.1 Introduction

Creating and integrating that link between Microsoft Azure Arc into an enterprise setup helps overcome quite a leap when looking to bridge traditional on-premises infrastructure with cloud capabilities. As businesses and technology evolves in the near future, a demand for more centralized management, security and operational consistency and scalability becomes absolutely imperative where Azure Arc is an ideal solution to help expand those features and capabilities, especially across diverse types of environment setups - whether it would be on-premises, hybrid-cloud or multi-cloud.

In this chapter the architectural design, onboarding process, alert configurations and overall operational workflows will be demonstrated that lays out and provides a solid understanding on the key principles that shapes the implementation strategy. By tailoring deployment and implementation to align with organisational priorities, this approach will help ensure that efficiency, scalability and overall security postures are maintained throughout.

4.2 Problem formulation

4.2.1 Context and Challenges

Methods on managing hybrid infrastructure with an enterprise setting comes with several obstacles, this is especially the case when existing tools are unable to provide a more streamlined approach in terms of manageability. At present, most governance and policy implementation capabilities and even real time monitoring tend to be scattered across various platforms, leading to potential gaps in security, a lack of consistency across environments and inefficiencies within operational workflows.

With the organisation gradually evolving along with its infrastructure expanding, it's clear that a more unified approach for server management is required. The IT department requires a solution that seamlessly integrates both the on-premises and cloud environment together without adding any further complexity. By harnessing the Azure Arc technology for this type of

management, it helps provide a solid framework to help achieve this but consideration needs to be taken in addressing operational and security concerns whilst ensuring its whole structure carefully fits within existing infrastructure.

4.2.2 Defining the Problem

The current server management method lacks centralized framework which makes it harder to enforce policies effectively, streamline administrative processes and set and maintain security standards as quite a lot of it is currently decentralized. The use of these diverse systems don't have any capabilities to interlink with each other which can lead to a number of inefficiencies and reduced visibility on critical, high availability infrastructure.

This project examines the deployment of Azure Arc as a practical solution to these challenges focusing on the onboarding and enrollment of servers, configuring alerts and policies. The goal would be to determine how Azure Arc helps improve resource along with security and efficiency whilst seamlessly integrating into existing workflows and management solutions.

4.2.3 Why it matters

If operating without a unified management framework, governing and securing server infrastructure can become difficult especially if environments increase in size. Inconsistencies with policies and can create unnecessary overhead making IT management more complex than required.

The project itself takes a more practical approach to evaluate Azure Arc' and where it is and can be applied in the real world, by assessing how it can improve visibility of security and governance along with overall efficiency within an enterprise setting. All insights obtained will help provide a clear and structured pathway for any business looking to modernize their infrastructure and to ensure long-term scalability where needed.

4.2.4 Scope and Focus

This study focused on the practical implementation of Azure Arc within an enterprise setting, this implementation specifically addresses:

- **Server Onboarding** - Simplifying the integration of existing and on-premises resources.
- **Alert Configuration** - Refining out of box monitoring workflows for security and operational oversight to notify or gather telemetry on any events that require action.
- **Policy Enforcement** - ensure that governance consistency across the hybrid environment is met.

Further advanced functionalities such as Kubernetes and automation, fall outside the scope of this project.

4.2.5 Project Goals

This project aims to explore how Azure Arc can optimize IT management within an enterprise setting by improving policy consistency, security enforcement and operational workflows. Key objectives include:

- Developing a streamlined onboarding process tailored to the organisation's infrastructure.
- Enhancing monitoring mechanisms to strengthen security posture.
- Refining governance practices to align with compliance requirements.

By implementing Azure Arc within a controlled enterprise environment, this project will provide actionable insight into hybrid IT optimization, ensuring long-term scalability, security and operational consistency.

4.3 Design

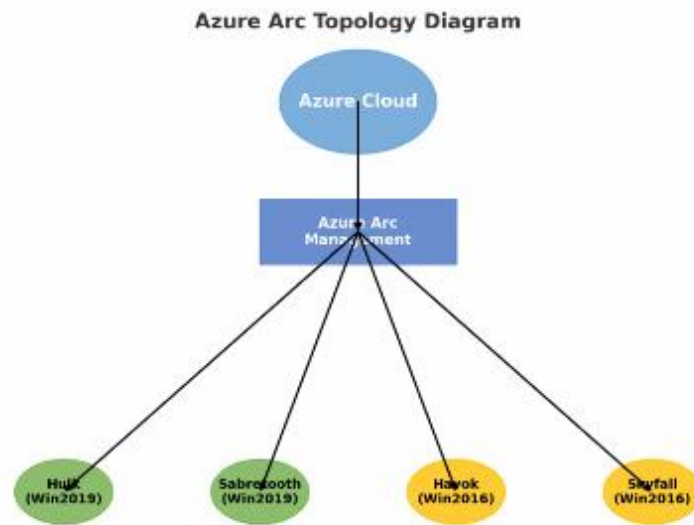


Figure 7 - Azure Arc topology demonstrating architectural setup (Image by David Corner).

4.4 Implementation

4.4.1 Project Objectives

The key goals for this phase of the project include:

- Onboarding non-Azure Windows servers into the Azure environment using Azure Arc.
- Setting up a dedicated Log Analytics workspace to collect and analyse telemetry data.
- Creating Azure Monitor alerts to notify administrators of CPU utilization and server heartbeat conditions.
- Applying Azure Policy to enforce baseline security and configuration settings.
- Validating Azure Arc functionality on both Windows Server 2016 and Windows Server 2019.

Pre-Implementation Preparation

Before onboarding could begin several preparatory steps were completed to ensure compatibility and connectivity with Azure services.

Operating System Compatibility

The project environment consisted of four servers, running either Windows Server 2016, Windows Server 2019 and Windows Server 2022. These diverse versions of operating systems are officially supported by Microsoft for Azure Arc onboarding and fully meet the system requirements for the Azure Connected Machine Agent.

Each server had sufficient system resources available, including at least 2 GB of RAM available, free disk space of up to 100gb each, a 64-bit operating system and PowerShell version 5.1 or later.

Azure Subscription and Configuration

An active Azure subscription was used to deploy the required services. As part of the setup:

- A resource group named 'RG_AzureArc' was created to logically group the Arc-enabled servers.
- A Log Analytics workspace named 'law-arc poc-uks-001' in accordance to applicable naming conventions that demonstrates its intent, services offered and location, was provisioned to store performance metrics and logs for all servers.
- A service principal was configured with the necessary roles (Contributor and Hybrid Resource Administrator) to securely onboard the machines using scripts and automation.
- Azure Arc itself does not incur any charges for server onboarding, but costs can apply for additional features such as logging, policy enforcement and security integration, depending on usage levels.

Network and Firewall Requirements

Azure Arc relies on outbound internet access to communicate with Azure services. No inbound traffic is required. Firewall rules were reviewed and updated to ensure that each server could send traffic over HTTPS (port 443).

In addition, name resolution was confirmed for required Microsoft endpoints. These include:

- **login.microsoftonline.com** - used for authentication
- **management.azure.com** - used for resource management
- **guestconfiguration.azure.com** - used for policy application
- **dc.services.visualstudio.com** - used by Azure Monitor and diagnostics

Where necessary, proxy settings were checked to ensure proper routing.

Required Tools

The following tools were installed on the servers:

- Azure CLI
- Az PowerShell module
- Azure Connected Machine Agent

These tools were required for registering and managing each server through Azure Arc.

Server Inventory

The servers included in this implementation are listed below:

Server Hostname	Operating System	Function	Server Type	IP Address
Hulk	Windows Server 2022	Application Server	Virtual	172.28.1.199
Skyfall	Windows Server 2019	Application Server	Virtual	172.28.2.52
Mystique	Windows Server 2016	Content Delivery Server (with IIS)	Virtual	172.28.0.206
Havok	Windows Server 2019	Content Delivery Server (with IIS)	Virtual	172.28.0.51

Figure 8 - Table showing host server information (Author: David Corner)

This mixture allowed the testing of Azure Arc's functionality across different supported Windows Server versions.

4.5 Azure Arc Onboarding

4.5.1 Agent Installation

Firstly the device needs to be added to Azure Arc in order to commence with real time reporting and provide further analytics and alerting. To do this the 'Add a Machine' option within Azure Arc needs to be selected where it automatically provides a full script to onboard depending on if you require onboarding single or multiple hosts. When going through the onboarding options the 'RG_AzureArc' resource group that was initially created had been selected, along with the required region

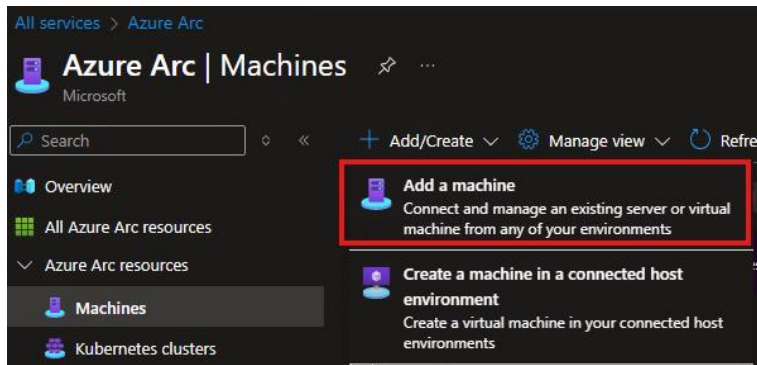


Figure 9 - Add a machine option within Azure Arc console (Screenshot by David Corner).

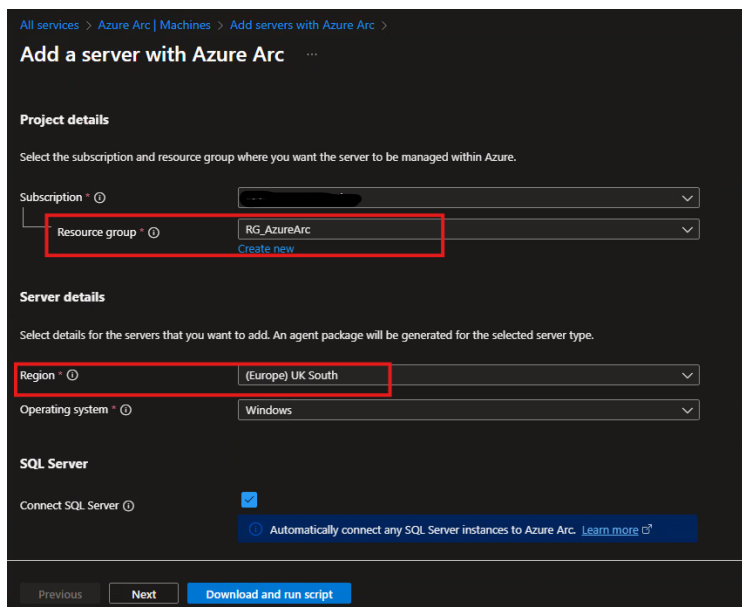
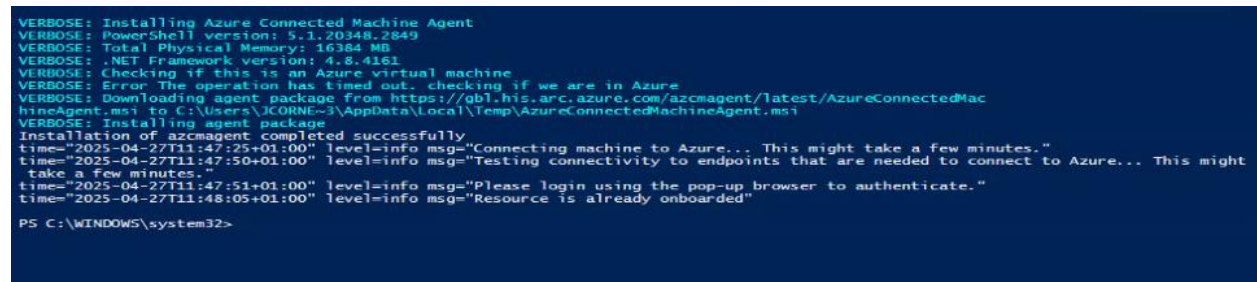


Figure 10 - Required resource group and region selected within the Azure Arc server onboarding configuration (Screenshot by David Corner).

By using the following script that was provided on each of the nominated servers, the Azure Connected Machine Agent was downloaded and installed on each server. This script not only installs the agent but automatically connects the server to the required resource group, cloud tenant and Azure region. This was carried out on each of the four servers so they are connected to Azure Arc services effectively:

```
1 $global:scriptPath = $myInvocation.MyCommand.Definition
2
3 function Restart-AsAdmin {
4     $pushCommand = "powershell"
5     if ($?env:env:PS_SCRIPT_PATH) {
6         $pushCommand = "push"
7     }
8
9     try {
10         Write-Host "This script requires administrator permissions to install the Azure Connected Machine Agent. Attempting to restart script with elevated permissions..."
11         $arguments = "-AddScriptCommand -a '$scriptPath'"
12         Start-Process $pushCommand -Verb RunAs -ArgumentList $arguments
13         exit 0
14     } catch {
15         Write-Host "Failed to elevate permissions. Please run this script as Administrator."
16     }
17 }
18
19 try {
20     if ([Security.Principal.WindowsPrincipal]([Security.Principal.WindowsIdentity]::GetCurrent()).IsInRole([Security.Principal.WindowsBuiltInRole]::Administrator)) {
21         if ($?env:env:PS_SCRIPT_PATH) {
22             Restart-AsAdmin
23         } else {
24             Write-Host "This script requires administrator permissions to install the Azure Connected Machine Agent. Please run this script as Administrator."
25         }
26     }
27
28     $env:SUBSCRIPTION_ID = "UNIQUE SUBSCRIPTION ID HIDDEN"
29     $env:RESOURCE_GROUP = "AZURE RESOURCE GROUP"
30     $env:TENANT_ID = "AZURE TENANT ID HIDDEN"
31     $env:LOCATION = "AZURE LOCATION"
32     $env:AUTH_TYPE = "AZURE AUTH TYPE"
33     $env:CLOUD = "AZURE CLOUD"
34
35     [Net.ServicePointManager]::SecurityProtocol = [Net.ServicePointManager]::SecurityProtocol -bor 3072
36
37     # Download the installation package
38     Invoke-WebRequest -Uri "https://gbl.his.azure.com/azcmagent-windows" -TimeoutSec 30 -OutFile "$env:TEMP\install_windows_azcmagent.ps1"
39
40     # Install the hybrid agent
41     & "$env:TEMP\install_windows_azcmagent.ps1"
42     if ($LASTEXITCODE -ne 0) { exit 1 }
43
44     # Run connect command
45     & "$env:ProgramFiles64\AzureConnectedMachineAgent\azcmagent.exe" connect --resource-group "$env:RESOURCE_GROUP" --tenant-id "$env:TENANT_ID" --location "$env:LOCATION" --subscription-id "$env:SUBSCRIPTION_ID" --cloud "$env:CLOUD" --correlation-id "$env:CORRELATION_ID"
46
47     catch {
48         $logBody = @($SubscriptionId,$env:SUBSCRIPTION_ID,$env:RESOURCE_GROUP,$env:TENANT_ID,$env:LOCATION,$env:CORRELATION_ID,$env:AUTH_TYPE,$env:OPERATION,"onboarding",$messageType,$_.FullyQualifiedExceptionId,$_.Message)
49         Invoke-WebRequest -Uri "https://gbl.his.azure.com/log" -Method "POST" -Body ($logBody | ConvertTo-Json) -OutFile "$env:TEMP\log.json"
50         Write-Host -ForegroundColor Red $_.Exception
51     }
52 }
```

Figure 11 – Azure Arc onboarding Powershell code (Screenshot by David Corner)



```
VERBOSE: Installing Azure Connected Machine Agent
VERBOSE: PowerShell version: 5.1.20348.2849
VERBOSE: Total Physical Memory: 16384 MB
VERBOSE: .NET Framework version: 4.8.4161
VERBOSE: Checking if this is an Azure virtual machine
VERBOSE: Error The operation has timed out. checking if we are in Azure
VERBOSE: Downloading agent package from https://gbl.his.azure.com/azcmagent/latest/AzureConnectedMachineAgent.ms
VERBOSE: Installing agent package
Installation of azcmagent completed successfully
time="2025-04-27T11:47:25+01:00" level=info msg="Connecting machine to Azure... This might take a few minutes."
time="2025-04-27T11:47:50+01:00" level=info msg="Testing connectivity to endpoints that are needed to connect to Azure... This might take a few minutes."
time="2025-04-27T11:47:51+01:00" level=info msg="Please login using the pop-up browser to authenticate."
time="2025-04-27T11:48:05+01:00" level=info msg="Resource is already onboarded"
PS C:\WINDOWS\system32>
```

Figure 12 - Powershell output on host device demonstrating successful Azure Arc server onboard (Screenshot by David Corner).

Once connected, the servers appeared in the Azure portal under Azure Arc - Servers, showing active status and basic telemetry. This demonstrates that real-time monitoring has enabled effectively.

Name ↑	Kind ↑	Arc agent status ↑	Resource group ↑	Subscription ↑	Operating system ↑	Defender extension ↑	Monitoring extens... ↑	Update status ↑
Hulk	VMware	Connected	RG_AzureArc		Windows Server 2022 St...	Not enabled	Installed	Enable periodic assessm...

Figure 13 - Azure Arc Machine list showing HULK successfully connected (Screenshot by David Corner).

[^] Essentials

Resource group : [RG_AzureArc](#)
 Status : Running
 Subscription :
 Subscription ID :
 Agent version : 1.51.03008.2167
 Image : Microsoft Windows Server 2019 (64-bit)
 Virtual machine kind : VMware
 Resource pool/cluster/host : ---
 Datastore : ---

Tags ([edit](#)) : [Add tags](#)

Properties

Capabilities

Virtual machine

Name : Hulk
 Operating system : Windows
 CPU cores : 4
 Memory : 16,384 MB

Networking
 Network interfaces : 1
 fe80:c6a4c8311:716d:5d94

Disks
 Disks : 2
 Total size : 450 GB

Configuration
 Guest management : Enabled (connected)

Extensions

AdminCenter (Microsoft.HybridCompute/machines/extensions)
 DependencyAgentWindows (Microsoft.HybridCompute/machines/extensions)

Figure 14 - Overview page showing telemetry for server HULK (Screenshot by David Corner).

Name ↑	Kind ↑	Arc agent status ↑	Resource group ↑	Subscription ↑	Operating system ↑	Defender extension ↑	Monitoring extens... ↑	Update status ↑
skyfall	VMware	Connected	RG_AzureArc		Windows Server 2019 St...	Not enabled	Installed	Enable periodic assessm...

Figure 15 - Azure Arc Machine list showing SKYFALL successfully connected (Screenshot by David Corner).

[^] Essentials

Resource group : [RG_AzureArc](#)
 Status : Running
 Subscription :
 Subscription ID :
 Agent version : 1.50.02986.2095
 Image : Microsoft Windows Server 2019 (64-bit)
 Virtual machine kind : VMware
 Resource pool/cluster/host : ---
 Datastore : ---

Tags ([edit](#)) : [Add tags](#)

Properties

Capabilities

Virtual machine

Name : skyfall
 Operating system : Windows
 CPU cores : 4
 Memory : 8,192 MB

Networking
 Network interfaces : 1
 fe80:c78c30d4:41f0ee5e

Disks
 Disks : 1
 Total size : 190 GB

Configuration
 Guest management : Enabled (connected)

Extensions

WindowsPatchExtension (Microsoft.HybridCompute/machines/extensions)
 DependencyAgentWindows (Microsoft.HybridCompute/machines/extensions)
 AzureMonitorWindowsAgent (Microsoft.HybridCompute/machines/extensions)

Figure 16 - Overview page showing telemetry for server SKYFALL (Screenshot by David Corner).

Name ↑	Kind ↑	Arc agent status ↑	Resource group ↑	Subscription ↑	Operating system ↑	Defender extension ↑	Monitoring extension ↑	Update status ↑
Mystique		Connected	RG_AzureArc		Windows Server 2016 St...	Not enabled	Installed	Enable periodic assessm...

Figure 17 - Azure Arc Machine list showing MYSTIQUE successfully connected (Screenshot by David Corner).

Essentials	
Resource group (move)	RG_AzureArc
Status	Connected
Location (move)	UK South
Subscription (move)	
Subscription ID	
Agent version	1.50.02986.2095
Computer name	Mystique
FQDN	Mystique.
Operating system	Windows Server 2016 Standard
Operating system version	10.0.14393.7606
Cloud provider	N/A
Manufacturer	HPE
Model	ProLiant DL380 Gen10
Arc gateway	Add Arc gateway
License type	KMS Client, Licensed
Azure benefits	Not activated

Figure 18 - Overview page showing telemetry for server MYSTIQUE (Screenshot by David Corner).

Name ↑	Kind ↑	Arc agent status ↑	Resource group ↑	Subscription ↑	Operating system ↑	Defender extension ↑	Monitoring extension ↑	Update status ↑
Havok	VMware	Connected	RG_AzureArc		Windows Server 2019 St...	Not enabled	Installed	Enable periodic assessm...

Figure 19 - Azure Arc Machine list showing HAVOK successfully connected (Screenshot by David Corner).

Essentials	
Resource group	RG_AzureArc
Status	Running
Subscription	
Subscription ID	
Agent version	1.50.02986.2095
Image	Microsoft Windows Server 2019 (64-bit)
Virtual machine kind	VMware
Resource pool/cluster/host	---
Datastore	---
Tags (edit)	Add tags
Properties Capabilities	
Virtual machine	
Name	Havok
Operating system	Windows
CPU cores	4
Memory	16,384 MB
Networking	
Network interfaces	1
	fe80ca96fadd9a202:39fb
Disks	
Disks	2
Total size	1,114 GB
Configuration	
Guest management	Enabled (connected)
Extensions	
WindowsPatchExtension (Microsoft.HybridCompute/machines/extensions)	
DependencyAgentWindows (Microsoft.HybridCompute/machines/extensions)	
AzureMonitorWindowsAgent (Microsoft.HybridCompute/machines/extensions)	

Figure 20 - Overview page showing telemetry for server HAVOK (Screenshot by David Corner).

4.6 Monitoring and Alerts

4.6.1 Log Analytics Configuration

By default within Azure Arc, each server stores all its real-time telemetry on an individual basis, which can pose quite a challenge when it comes to configuring alert policies and keeping visibility on real-time monitoring. To overcome this a Log Analytics workspace is required that essentially brings all this data all servers and dumps them into this central store where all the monitoring policies are applied to. In addition, having this set up helps provide benefits in the log run as any newly provisioned servers will automatically poll its data here and be part of real-time alerting as enforced by Azure policy.

In order for the above to take place, the Log Analytics workspace 'law-arc poc-uks-001' had been created and configured to collect telemetry such as CPU, memory and heartbeat activity from all Arc-connected servers. This was set up by searching for 'Log Analytics Workspaces' in Azure and selecting the 'Create' option within the header pane where the following setup wizard appears to configure accordingly.

The screenshot shows the 'Create Log Analytics workspace' wizard in the Azure portal. The breadcrumb navigation at the top indicates 'Home > Log Analytics workspaces >'. The main heading is 'Create Log Analytics workspace'. Below this is an information box stating: 'A Log Analytics workspace is the basic management unit of Azure Monitor Logs. There are specific considerations you should take when creating a new Log Analytics workspace. [Learn more](#)'. Below the information box is a descriptive paragraph: 'With Azure Monitor Logs you can easily store, retain, and query data collected from your monitored resources in Azure and other environments for valuable insights. A Log Analytics workspace is the logical storage unit where your log data is collected and stored.' The 'Project details' section instructs to 'Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.' It contains two dropdown menus: 'Subscription' and 'Resource group'. The 'Resource group' dropdown is currently selected and shows 'RG_AzureArc' with a 'Create new' link below it. The 'Instance details' section contains two more dropdown menus: 'Name' (set to 'law-arc poc-uks-001') and 'Region' (set to 'UK South').

Figure 21 - Overview page demonstrating resource group being selected when creating log analytics workspace (Screenshot by David Corner).

Once the workspace had been created, Azure Arc will need to be instructed on what types of performance metrics are needed for real-time monitoring through the Log Analytics Workspace. In order to achieve this, Data collection rules need to be configured and pointed to the types of hardware that requires real-time probing.

To configure this under the new workspace, the 'Agent' options under 'settings' for that Workspace will need to be selected to enable further probing capabilities.

Add data source [X]

Data source | Destination

Select which data source type and the data to collect for your resource(s).

Data source type *

Performance Counters [v]

Choose Basic to enable the collection of performance counters. Choose Custom if you want more control over which performance counters are collected.

None Basic Custom

☒ Performance counter	Sample rate (seconds)
☒ CPU	60
☒ Memory	60
☒ Disk	60
☒ Network	60
☒ System	60
☒ Process	60

Save Next: Destination > Cancel

Figure 22 - Add data source page within data collection rules demonstrating included performance counters (Screenshot by David Corner).

For implementation purposes for the project, the basic option was enabled which should provide sufficient real-time server information, for server analysis and event alerting where required.

4.6.2 Azure Alerts

To provide proactive monitoring, an alert rule was created to detect when processor utilization on the hosts exceeds 80%, using the following KQL query within Azure Monitoring:

```
1 Perf
2 | where ObjectName == "Processor" and CounterName == "% Processor Time" and InstanceName == "_Total"
3 | where TimeGenerated > ago(5m)
4 | summarize AvgCPU = avg(CounterValue), Samples = count() by Computer
5 | where AvgCPU > 80
6 | project TimeGenerated = now(), Computer, AvgCPU, Samples
7 | order by AvgCPU desc
8
```

Figure 23 - KQL query to display and alert hosts that exceed 80% CPU utilization (Screenshot by David Corner).

This query was scheduled to run every five minutes. If the threshold was met, an alert was triggered within the Azure Alert console and an email notification was sent to the system administrator via a configured email action group.

Additionally, to the above a 'heartbeat' alert rule had also been configured to detect if a server unexpectedly goes offline using the following KQL query within Azure Monitoring:

```
1 Heartbeat
2 | where TimeGenerated > ago(5m) // Last 5 minutes, adjust if needed
3 | summarize LastHeartbeat = max(TimeGenerated) by Computer
4 | project Computer, LastHeartbeat
5 | order by LastHeartbeat desc
```

Figure 24 - KQL query to display and alert hosts that miss a heartbeat scan within the last 5 minutes (screenshot by David Corner).

This query is also scheduled to check every five minutes and if this particular event fires, an alert will be triggered within the Azure Alert console along with an email notification sent to the system administrator.

4.7 Azure Policy Setup

As part of bringing the servers into Azure Arc, a couple of Azure Policies were applied to keep everything consistent and improve monitoring. These policies were chosen because they were simple to set up and made sense as they would benefit the environment.

4.7.1 Checking the Time Zone

The first policy used was *'Audit Windows machines that are not set to the specified time zone'*.

This policy helps by checking if each server has the right time zone setting. For this project, it needed to be set to *'(GMT) Dublin, Edinburgh, Lisbon, London'*.

Making sure all servers have the same time zone is important for things like log files and effective change tracking. Once the policy was set up, a check was done to make sure it was active and a quick check over the Azure policy page showed the policy was assigned correctly and reflecting the correct status(s). The results from this are covered later in the Testing and Validation chapter.

4.7.2 Sending Activity Logs to Log Analytics

The second policy used was *'Configure Azure Activity logs to stream to a specified Log Analytics workspace'*.

This one makes sure that Azure Activity Logs, like resource changes and admin actions, are automatically sent to a Log Analytics workspace. For this setup, the logs were sent to 'law-arc poc-001'. After setting it up, the policy showed as assigned and basic checks confirmed it was active. Details on whether the logs were actually arriving in the workspace are explained in the next chapter.

4.8 Challenges Encountered

A few small challenges came up during the setup of Azure Arc and the related services.

One issue was with network connectivity. At the beginning, one of the servers (hulk) could not reach Azure endpoints because outbound HTTPS traffic was blocked by the firewall. This was fixed by updating the firewall settings to allow traffic on port 443, which is needed for the Azure Connected Machine Agent to communicate effectively. For future projects, comparing the networking requirements with current firewall rules should help avoid these delays as server rules can be excluded to communicate directly to those services, that should help alleviate any in depth investigations that in turn can cause delays with implementation.

Another challenge was with time synchronisation. One Windows Server 2016 machine had a system clock that was out of sync by several minutes. Since accurate timekeeping is important for authentication and logging an NTP server was configured to fix the issue.

There were also some minor delays when waiting for policy compliance results and log data to appear in the Azure portal. These delays were expected as these tend to happen from time to time with cloud solutions, but it took some patience for everything to show up properly after the initial setup.

Overall, the challenges were straightforward to resolve and did not cause major disruption to the project timeline.

4.9 Results and Validation

After completing the implementation steps, several checks were carried out to make sure everything was working as expected.

All the four servers appeared in the Azure portal under Azure Arc-enabled servers, showing a connected status and reporting basic telemetry like operating system versions, memory and disk sizes along with activity status.

The installation of the Azure Connected Machine Agent was successful on each machine. Monitoring was set up correctly and performance data such as CPU, memory, disk, process, system and network usage started flowing into the 'law-arc poc-001' Log Analytics workspace since being enabled as data collection rules.

Both Azure Policies were assigned properly. The time zone audit policy was active and the activity log streaming policy was showing as assigned to the resource group.

Validation of the policies, as well as further testing of alerts and monitoring, is discussed in the next chapter (Testing and Validation).

4.10 Conclusion

The project to set up Microsoft Azure Arc for four on-premises Windows servers was completed successfully. It showed that traditional servers can be managed through Azure without needing to move them fully into the cloud. Important tasks like connecting the servers, setting up performance monitoring and applying policies were all put in place and tested. A few issues came up during the setup, but they were solved at ease without causing major delays.

By choosing straightforward but effective Azure Policies, such as checking that time zone settings were correct and making sure activity logs were being recorded, the project improved both the consistency and visibility of the environment. The work done during this phase created a strong base for validating the setup and making further improvements, which are discussed later in the report.

Chapter 5 - Evaluation and Testing

5.1 Introduction

After setting up Azure Arc and getting the servers connected, a few tests were done to make sure everything worked like it was supposed to. This part of the report goes over what was tested, how it was checked and what the results were.

5.2 Checking Azure Arc Connections

The first thing checked was whether the servers showed up properly in Azure Arc.

Each server was looked at in the Azure Portal to see if it was connected and sending back basic details like its name, operating system and connection status.

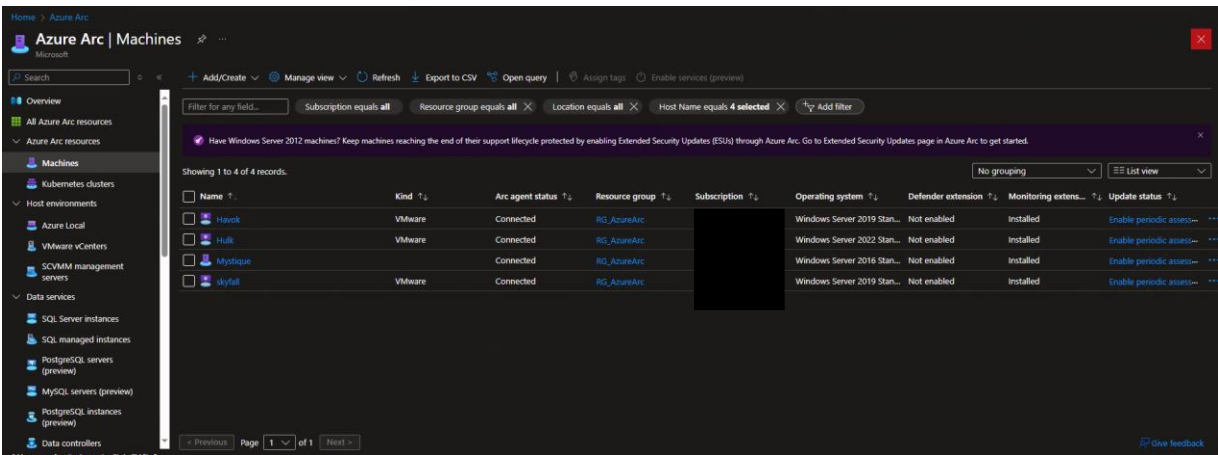


Figure 25 - All servers connected and showing healthy in Azure Arc (Screenshot by David Corner).

As shown in Figure 25, all four servers appeared in the Azure Arc dashboard. Their connection status showed as healthy and the details matched what was expected, which meant the agent installation and registration worked properly along with an accurate representation of its connectivity status.

5.3 Checking Monitoring Through Log Analytics

The next step was to check if performance data and heartbeats were being sent into the Log Analytics workspace (law-arc poc-001).

This was done by running a few basic queries in the workspace and checking for heartbeat entries and CPU/memory usage stats from each server.

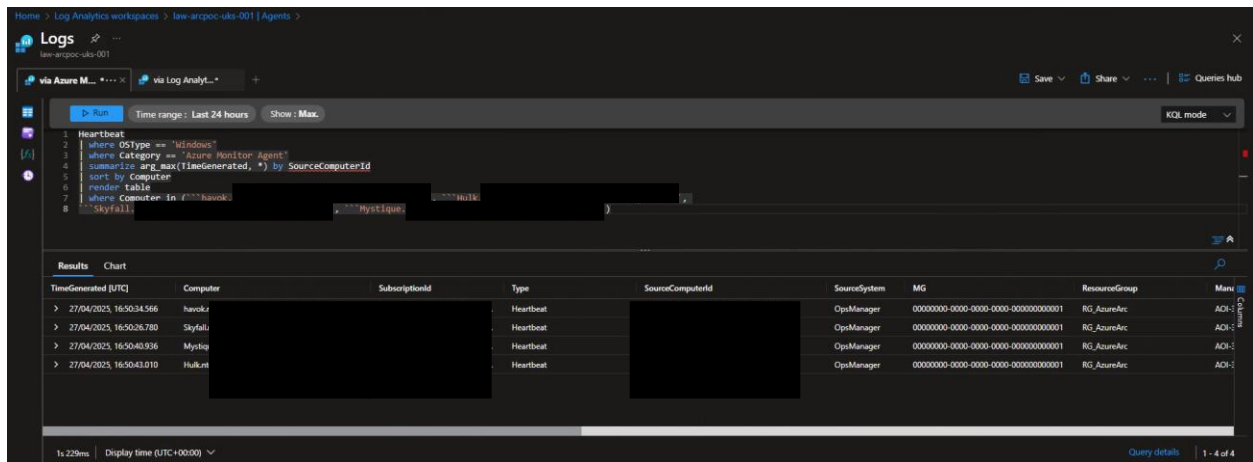


Figure 26 - Heartbeat metrics collected in Log Analytics (Screenshot by David Corner).

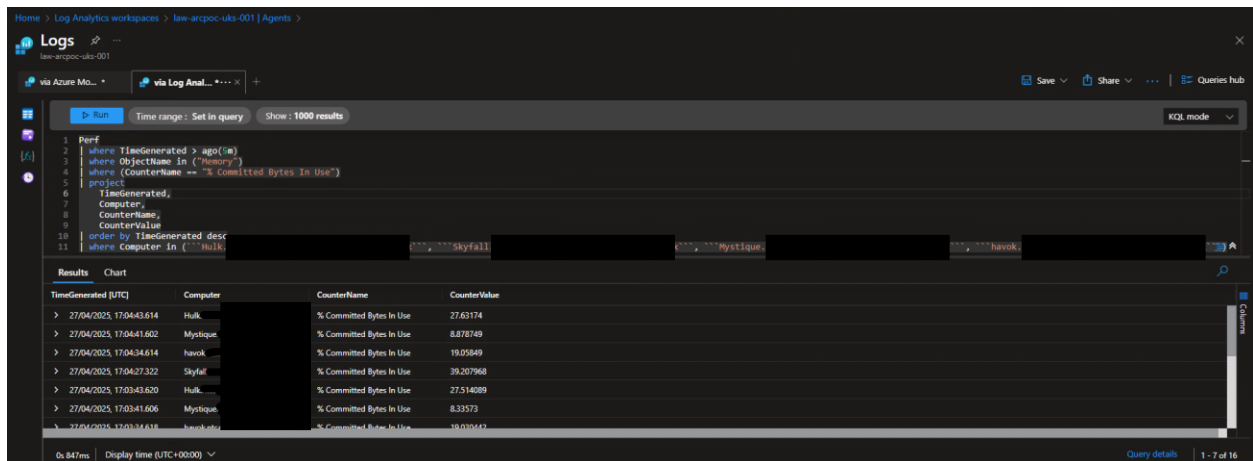


Figure 27 - memory metrics collected in Log Analytics (Screenshot by David Corner).

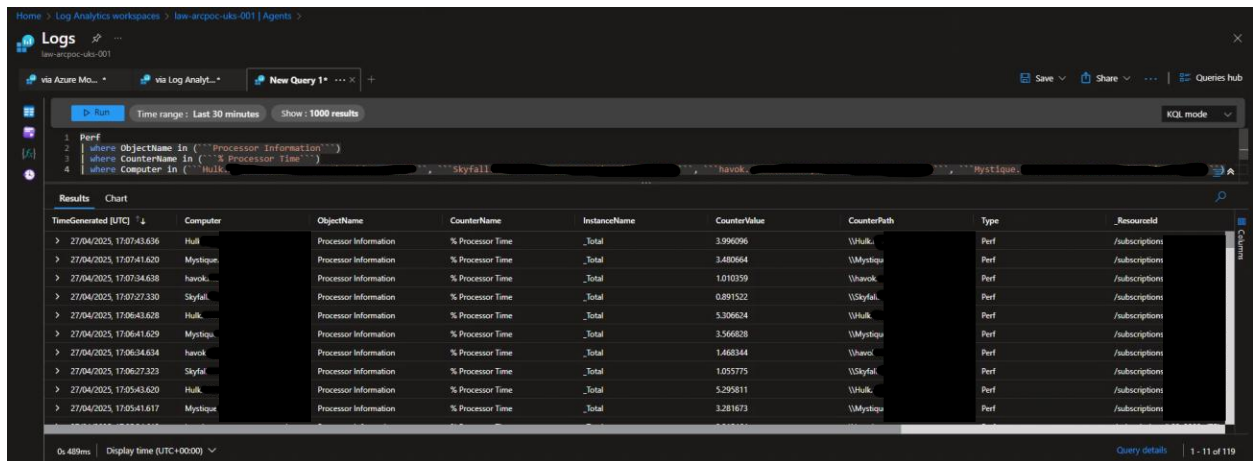


Figure 28 - Performance metrics collected in Log Analytics (Screenshot by David Corner).

By referring to the above figures, this should effectively demonstrate that the monitoring side of the setup was working properly.

5.4 Checking Azure Policy Compliance

Two policies were applied earlier and both were checked to make sure they were doing what they were supposed to.

5.4.1 Time Zone Policy

The first policy was set to check that all servers were using the right time zone (GMT London).

After letting Azure Policy scan the resources, the compliance report showed that all four servers were compliant.

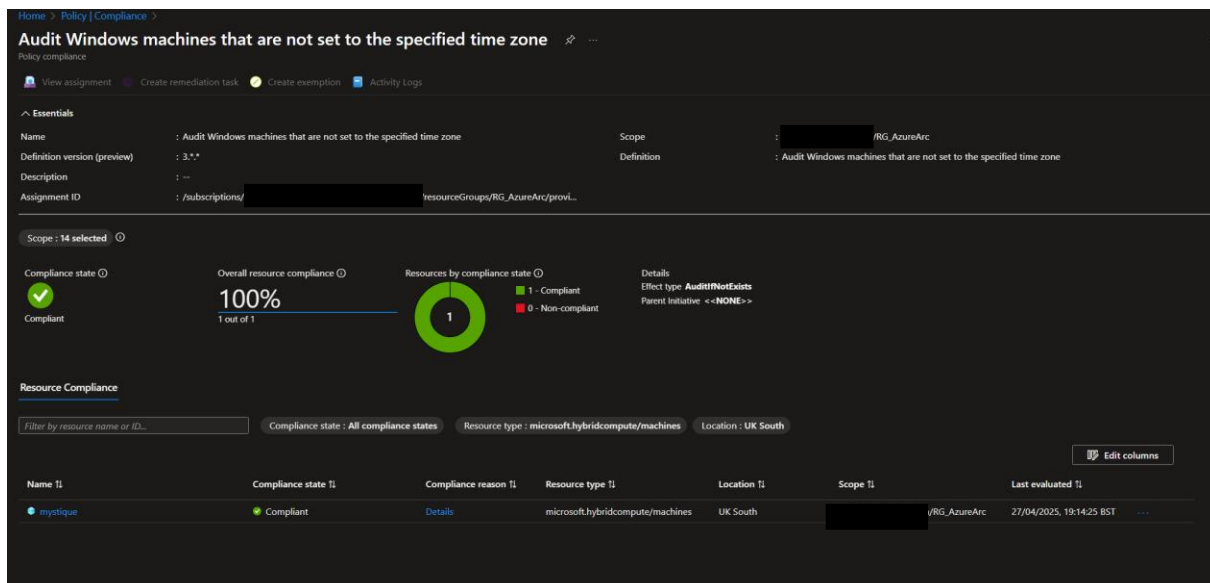


Figure 29 - Time zone policy showing as compliant on MYSTIQUE (Screenshot by David Corner).

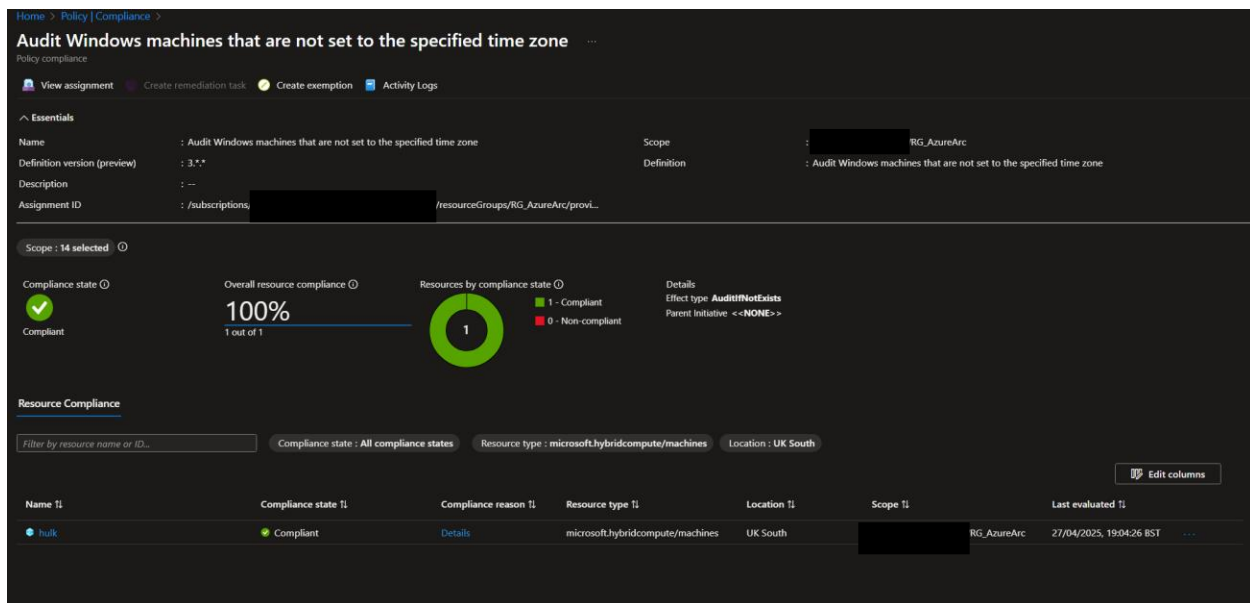


Figure 30 - Time zone policy showing as compliant on HULK (Screenshot by David Corner).

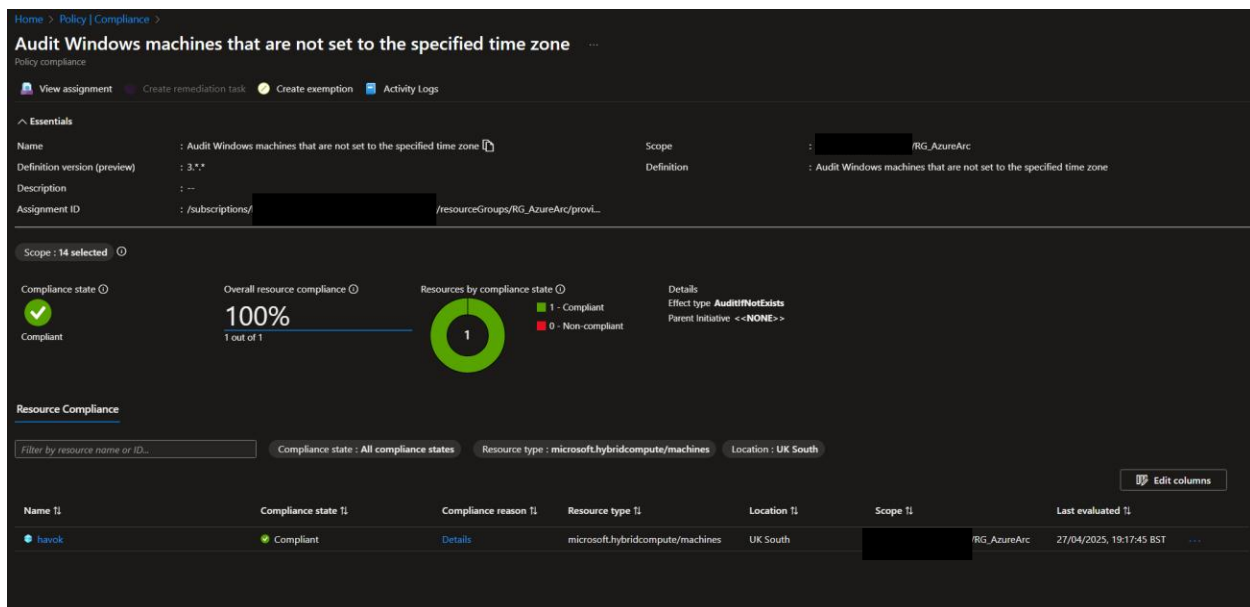


Figure 31 - Time zone policy showing as compliant on HAVOK (Screenshot by David Corner).

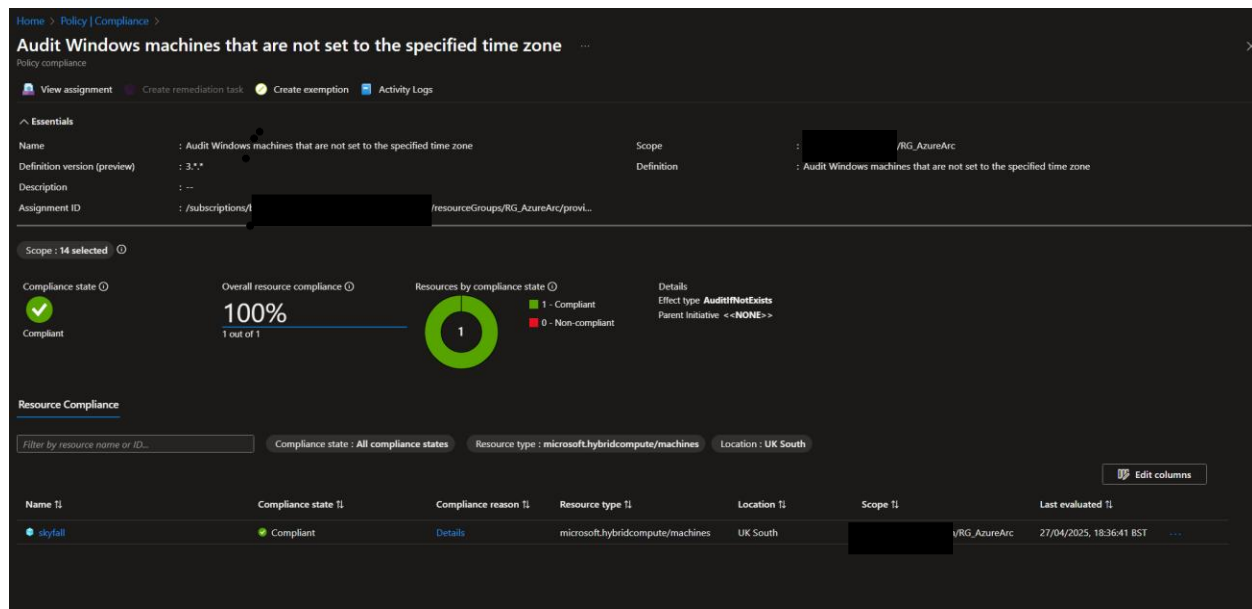


Figure 32 - Time zone policy showing as compliant on SKYFALL (Screenshot by David Corner).

All servers passed the check without any problems, which meant their clocks were set up properly.

5.4.2 Activity Log Streaming Policy

The second policy was about making sure Azure Activity Logs were being sent to the workspace. This was checked by opening Log Analytics and looking at the logs collected. There were entries for things like resource group changes and policy assignments, showing that the logs were coming through.

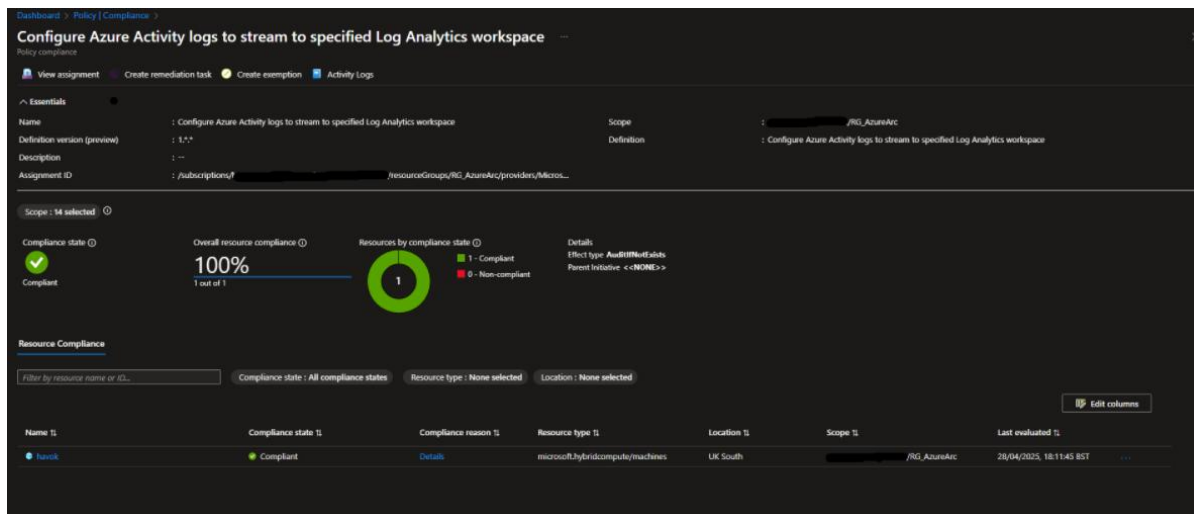


Figure 33 - 'Configure Azure Activity logs to stream to a specified Log Analytics workspace' Azure policy showing as compliant against server HAVOK (Screenshot by David Corner).

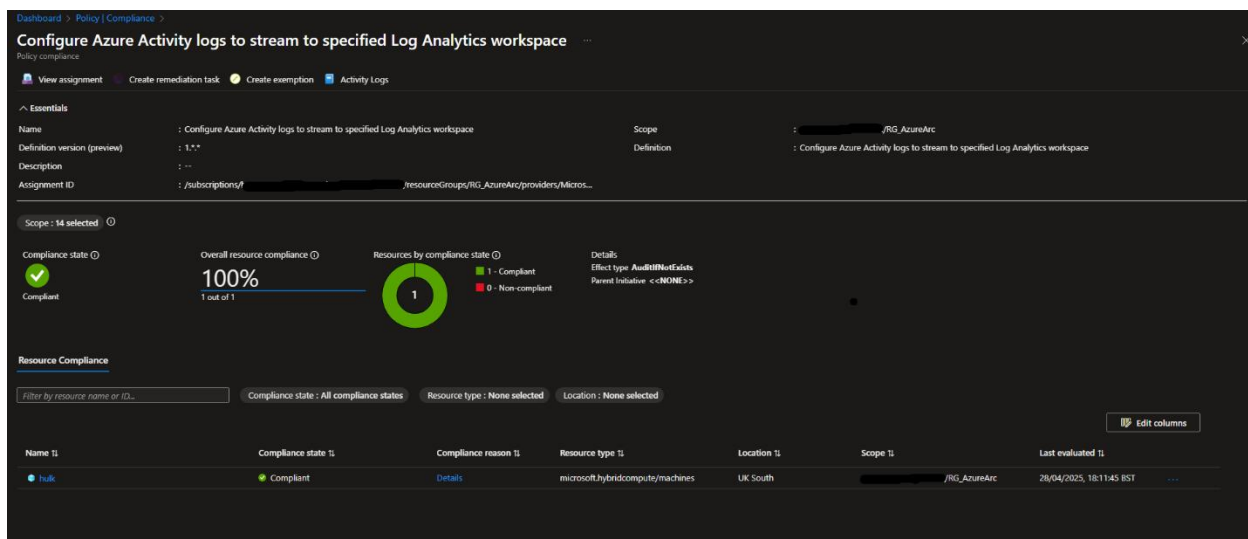


Figure 34 - 'Configure Azure Activity logs to stream to a specified Log Analytics workspace' Azure policy showing as compliant against server HULK (Screenshot by David Corner).

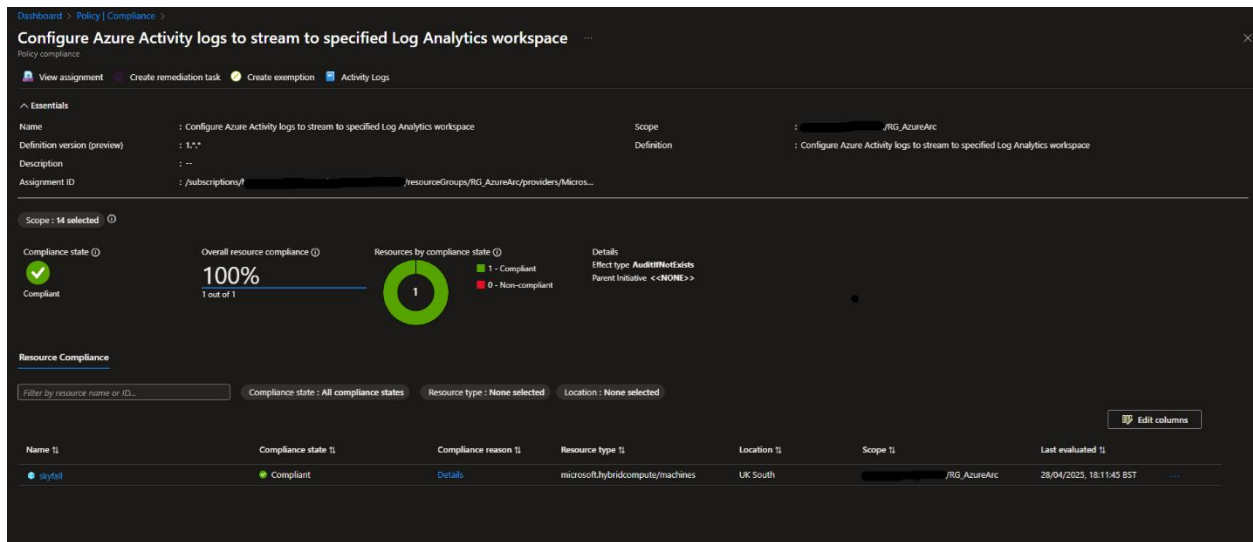


Figure 35 - ‘Configure Azure Activity logs to stream to a specified Log Analytics workspace’ Azure policy showing as compliant against server SKYFALL (Screenshot by David Corner).

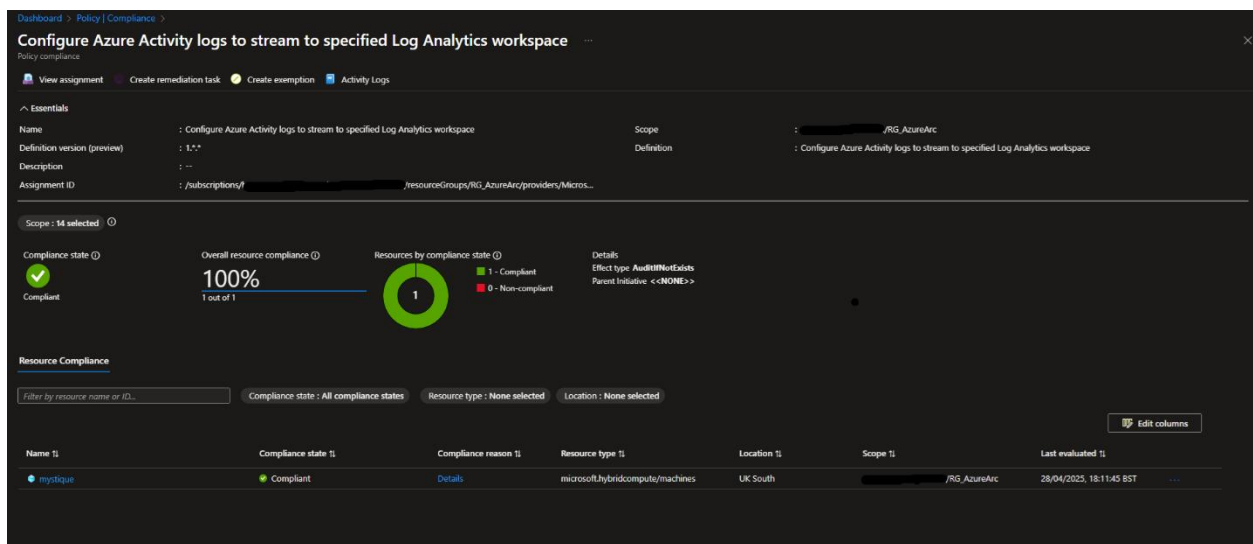


Figure 36 - ‘Configure Azure Activity logs to stream to a specified Log Analytics workspace’ Azure policy showing as compliant against server MYSTIQUE (Screenshot by David Corner).

5.4 Azure Monitor Alerts

As part of the project objectives, Azure Monitor alerts were configured to track CPU, memory utilization levels and server heartbeat conditions for the onboarded Windows Servers. These alerts were designed to notify administrators if CPU usage exceeded a defined threshold or if a server heartbeat was not received, indicating a potential server issue.

The alert rules were set up within the Log Analytics workspace (law-arc poc-001) and were assigned action groups for notification purposes. During the validation phase, the alert rule configurations were reviewed to ensure they were active and targeted at the correct resources. Although no real alerts were triggered during the testing period due to stable server conditions, the alert rules appeared correctly configured and active within the Azure Monitor portal.

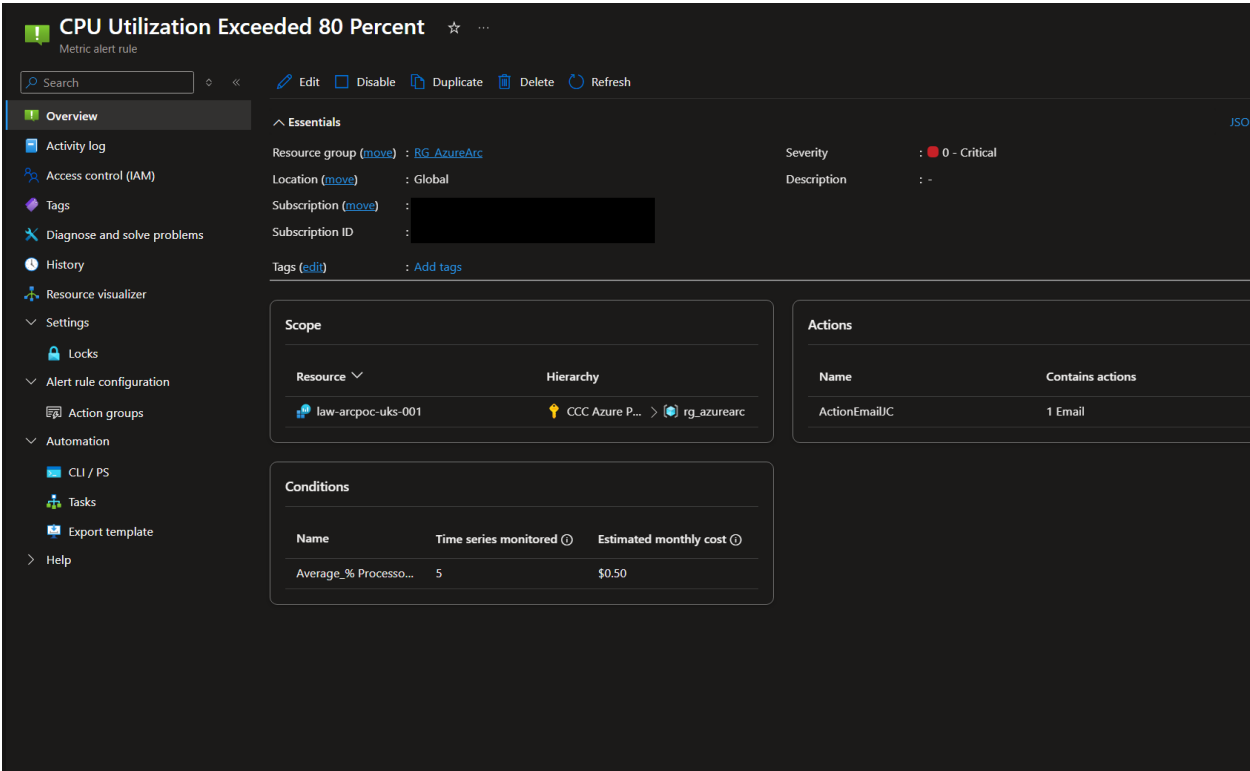


Figure 37 - Configured Azure Monitor alert overview page for CPU utilization (Screenshot by David Corner).

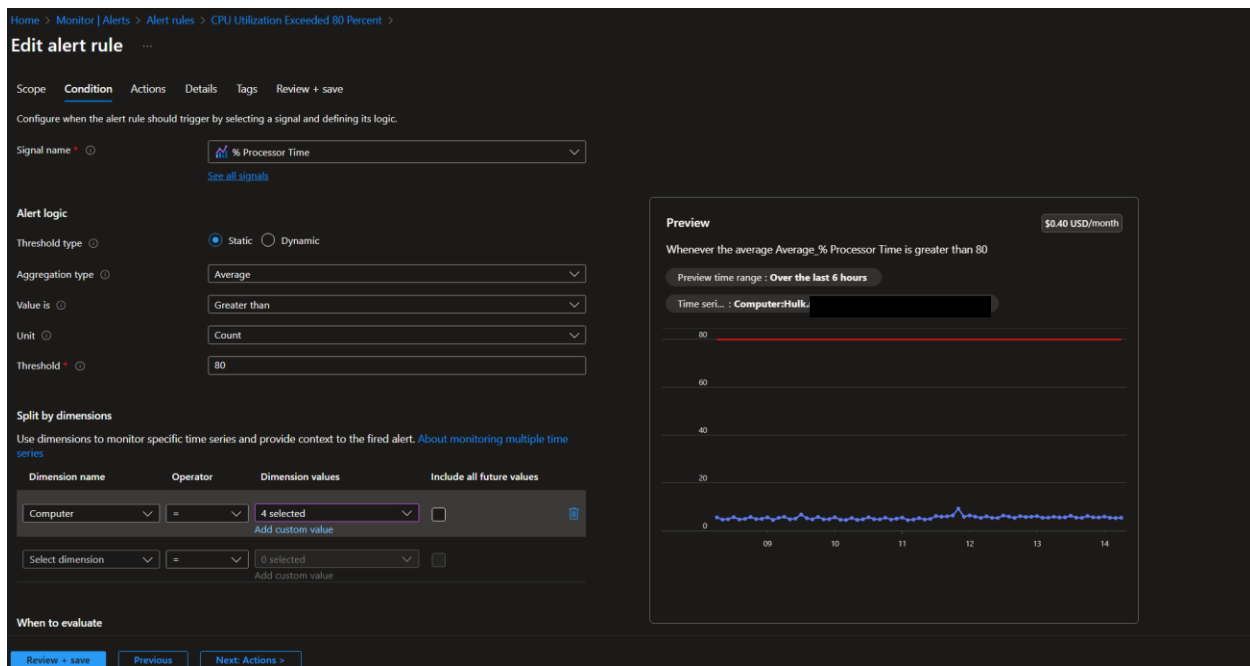


Figure 38 - Configured Azure Monitor alert rule conditions for CPU utilization (Screenshot by David Corner).

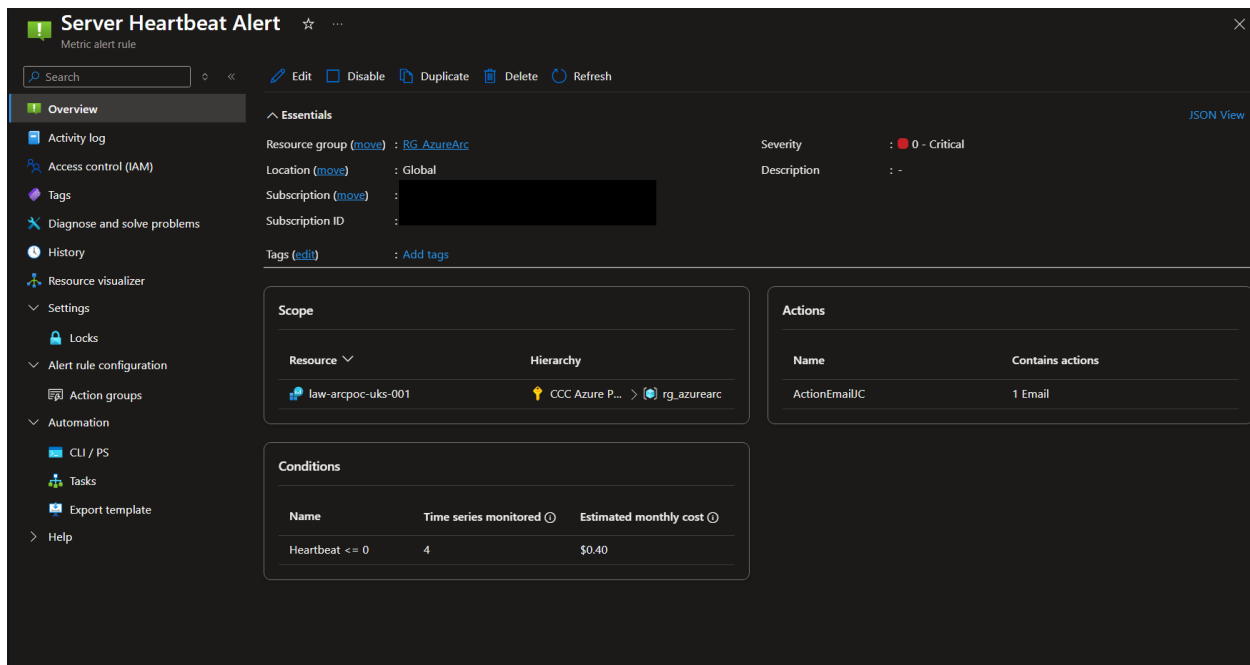


Figure 39 - Configured Azure Monitor alert rule overview page for heartbeats (Screenshot by David Corner).

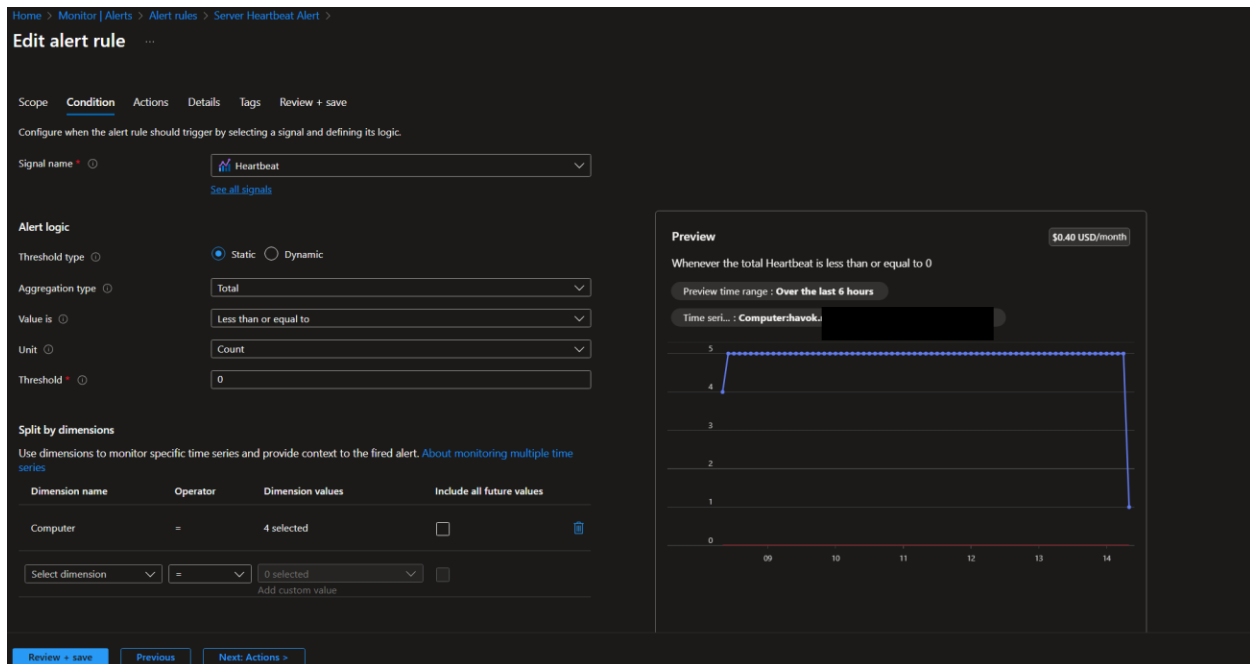


Figure 40 - Configured Azure Monitor alert rule conditions for heartbeats (Screenshot by David Corner).

5.5 Summary

The testing and validation phase confirmed that all key aspects of the Azure Arc implementation were working as intended.

Each server was successfully onboarded into Azure Arc and appeared as connected resources within the Azure Portal. Monitoring was configured through a dedicated Log Analytics workspace 'law-arc poc-001' with heartbeat and performance data successfully collected from all servers.

Azure Policies were applied and evaluated, with compliance results confirming that servers met the required configurations, including correct time zone settings and activity log streaming.

Azure Monitor alerts were also configured to track CPU utilization levels and server heartbeat status. Although no active alerts were triggered during the testing period, the alert rules were reviewed and confirmed to be active and correctly targeting the appropriate resources.

Overall, the implementation achieved the core monitoring and governance objectives, providing a clear demonstration of how Azure Arc can be used to manage and monitor on-premise servers through a cloud-based platform.

Chapter 6- Conclusions and Recommendations

6.1 Summary

This project looked at how Microsoft Azure Arc could be used to manage on-premise Windows servers without needing to move them to the cloud.

Four servers, running Windows Server 2016 and 2019, were onboarded into Azure and linked to a Log Analytics workspace called law-arcpoc-001. Two Azure Policies were also set up: one to check the servers' time zones and another to stream activity logs into the workspace.

After the setup, several checks were carried out. The servers connected properly, monitoring data was being collected and the policies showed that everything was set up the right way. Overall the project showed that using Azure Arc in a mixed environment is a practical and reliable solution for server manageability.

6.2 Conclusion

Bringing Azure Arc into the on-premise setup made managing the servers much simpler. Instead of using different tools for different servers everything could be viewed and managed in one place through the Azure Portal.

Setting up the servers was straightforward once the network and firewall settings were sorted out. Monitoring through Log Analytics worked well and having the Azure Policies in place added another layer of control.

The project showed that it's possible to apply cloud style management to older systems without needing to fully migrate them, making it easier for businesses to modernise at their own pace.

While the setup was successful, there's still plenty of room for adding more policies, alerts and automations in the future to make the environment even stronger.

6.3 Recommendations for Further Research

If this project were to be taken further, a few ideas could be explored:

- Managing larger environments - It would be useful to see how Azure Arc performs when managing a bigger number of servers and if anything changes when scaling up.
- Adding more policies - Looking into more built-in or custom Azure Policies would help cover areas like security updates, software installations, or backup settings.
- Automating fixes - Researching ways to automatically fix problems (for example, through Azure Automation when a server becomes non-compliant) could save even more time.
- Using advanced monitoring tools - It might be interesting to use features like Azure Monitor Workbooks or machine learning to predict server issues before they happen.
- Cost vs benefit - Doing a proper study on the running costs compared to the benefits could help businesses understand the real value of using Azure Arc for hybrid management.

Chapter 7- Reflections

7.1 Introduction

This chapter shows how well the aim and objectives for the project were achieved. It also discusses the challenges faced, how they were handled and how the project progressed overall. These reflections are linked back to the original objectives set at the start of the project.

7.2 Research Aim

The aim of this project was to explore the implementation of Microsoft Azure Arc for managing on-premise Windows servers, using Azure-based tools for monitoring, compliance and operational management without migrating the servers to the cloud.

7.3 Objectives and Reflections

Objective 1: Establish an understanding of Azure Arc's capabilities for hybrid cloud management.

This objective was achieved. Background research in the Literature Review chapter helped build a clear understanding of how Azure Arc fits into hybrid environments. The knowledge gained was directly applied during the setup and configuration of the Arc environment in the Implementation chapter.

Objective 2: Successfully connect on-premise servers to Azure Arc.

This objective was achieved. All four servers were connected to Azure Arc following the installation of the Connected Machine Agent. The servers appeared as connected resources in the Azure portal and monitoring through Log Analytics confirmed that the onboarding process was successful, as detailed in the Testing and Validation chapter.

Objective 3: Configure monitoring and basic policies through Azure Arc.

This objective was also achieved. A dedicated Log Analytics workspace (law-arcpoc-001) was created to collect system data from the servers. Two Azure Policies were applied to audit the servers' time zone settings and ensure activity logs were streamed to the workspace. Both policies showed successful compliance during testing.

7.4 Reflections on Conduct and Progress

The project overall went smoothly, but there were a few small challenges. One early problem was network-related: a firewall rule initially blocked outbound HTTPS traffic, which prevented the servers from connecting to Azure Arc. This was fixed quickly by adjusting firewall settings and spotting the affected server was mistakenly included in an implicit deny rule. Another minor issue was a server that had an incorrect system time due to being en-US out of box, which caused connection problems. This was solved by setting up proper NTP time syncing to en-GB, where in the end the time zone Azure Policy later implemented showed it as compliant.

Most problems were identified in the risk analysis section and minimise strategies, such as planning extra time for network troubleshooting proved effective. There were no major unexpected delays and the project stayed close to the original timeline.

One point for improvement would be to allow more time for policy evaluation and Azure system refreshes, as policy compliance results took longer to appear than expected. This could have been factored more accurately into the planning stages.

Overall, the project was conducted in an organised way and progress was steady from the initial setup through to testing and validation.

7.5 Conclusion

The implementation of Microsoft Azure Arc for four on-premises Windows servers was completed successfully. The project showed how traditional infrastructure can be brought into Azure's management platform without having to move anything fully into the cloud. Key steps like server onboarding, monitoring and applying policies were set up and working. A few problems came up during the setup, like firewall issues and time synchronisation, but these were sorted out without causing major setbacks.

Choosing simple but meaningful Azure Policies like checking time zone settings and making sure activity logs were collected, helped keep the environment more consistent and improved visibility across the servers. Monitoring and alerts were also put in place using Azure Monitor and the machines were sending back telemetry and triggering alerts based on the rules set.

Along the way, the project picked up on a few of the real-world challenges that had been highlighted in the earlier literature review like onboarding not always going as smoothly as Microsoft's guides suggest and monitoring needing extra work to fully cover everything properly. Even though this was a small-scale setup, it reflected some of the bigger challenges that larger organisations would likely face.

There were some limits to the project, mainly because it only covered four servers in a controlled environment. In a larger or live enterprise setup, things like policy drift, deeper security compliance and more complex monitoring would need to be planned out much more carefully.

If the project was expanded in future, it would make sense to roll out Azure Arc across multiple sites or regions, add more automation for policy management and look at plugging in third-party monitoring tools to improve visibility even further.

Overall, the project proved that Azure Arc is a strong option for organisations wanting to manage on-premises and cloud servers through a single platform, without needing to fully migrate everything to Azure.

Bibliography

- [1] R. G. Aravind, "Optimizing Hybrid Cloud Architectures with Azure Arc in the Era of Multi-Cloud," *International Journal of Multidisciplinary Research in Science, Engineering and Technology Management*, vol. 12, no. 3, pp. 770-774, Mar. 2025. [Online]. Available: <https://philarchive.org/rec/ARAOHC>
- [2] T. Greenhalgh, S. Thorne, and K. Malterud, "Time to challenge the spurious hierarchy of systematic over narrative reviews?," *European Journal of Clinical Investigation*, vol. 48, no. 6, p. e12931, 2018.
- [3] S. Krishnan, "Managing Hybrid Cloud Infrastructures with Azure Arc," *IEEE Cloud Computing*, vol. 8, no. 1, pp. 27-34, 2021.
- [4] D. Moher, A. Liberati, J. Tetzlaff, and D. G. Altman, "Preferred reporting items for systematic reviews and meta-analyses: The PRISMA statement," *PLoS Medicine*, vol. 6, no. 7, p. e1000097, 2009.
- [5] B. Kitchenham and S. Charters, "Guidelines for performing Systematic Literature Reviews in Software Engineering," *EBSE Technical Report*, 2007.
- [6] M. Krishnan, "Hybrid cloud management trends in 2023," *IEEE Internet Computing*, vol. 27, no. 1, pp. 41-49, 2023.
- [7] Y. Zhang, C. Xu, and J. Lin, "Cloud governance and security in hybrid environments," *International Journal of Information Management*, vol. 52, p. 102085, 2020.
- [8] H. Kim, S. Park, and J. Lee, "Security challenges in hybrid cloud computing environments," *IEEE Access*, vol. 8, pp. 122-136, 2020.
- [9] S. Krishnamurthy, V. Sitaraman, and T. Bapat, "Performance considerations in hybrid cloud architectures," *IEEE Transactions on Cloud Computing*, 2022.

- [10] Microsoft Azure Blog, "Extending Azure Management to On-Premises with Azure Arc," 2024. [Online]. Available: <https://azure.microsoft.com/en-us/blog/extending-azure-management-to-on-premises-with-azure-arc>
- [11] Gartner, "Hybrid Cloud Trends and Best Practices 2024," Gartner Research, 2024. [Online]. Available: <https://www.gartner.com>
- [12] R. Buyya, R. Ranjan, and R. N. Calheiros, "InterCloud: Utility-Oriented Federation of Cloud Computing Environments for Scaling of Application Services," *Proceedings of the 10th International Conference on Algorithms and Architectures for Parallel Processing*, 2020.
- [13] T. Erl, R. Puttini, and Z. Mahmood, *Cloud Computing: Concepts, Technology & Architecture*, Pearson Education, 2013.
- [14] Microsoft Learn, "Monitor Azure Arc-enabled servers with Azure Monitor," Microsoft, Jan. 2024. [Online]. Available: <https://learn.microsoft.com/en-us/azure/azure-monitor/agents/azure-monitor-agent-manage>
- [15] M. Mishra, S. Patel, and R. Sharma, "Implementing Hybrid Cloud Solutions Using Azure Arc to Integrate On-Premises and Cloud Environments," *ResearchGate*, Feb. 2025. [Online]. Available: <https://www.researchgate.net/publication/388757925>
- [16] R. Schutten, "Implementing Hybrid Governance and Compliance with Azure Policy and Azure Arc," *Schutten.Cloud*, Aug. 2023. [Online]. Available: <https://schutten.cloud/post/implementing-hybrid-governance-and-compliance-with-azure-policy-and-azure-arc/>
- [17] Microsoft Learn, "Governance, Security, and Compliance Baseline for Azure Arc-Enabled Servers," Microsoft, Jan. 2024. [Online]. Available: <https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/scenarios/hybrid/arc-enabled-servers/eslz-security-governance-and-compliance>
- [18] P. Grozev and R. Buyya, "Multi-Cloud Provisioning and Load Distribution for Three-Tier Applications," *ACM Transactions on Autonomous and Adaptive Systems*, vol. 9, no. 3, 2014.

[19] M. Saunders, P. Lewis, and A. Thornhill, Research Methods for Business Students, 5th ed., Harlow, England: Pearson Education, 2007.