Plan of Action

Hybrid IT Management Framework

Document Revision History

Project Name: Hybrid IT Management Framework – Plan of Action

Document Status: Final

Document Version	Date	Prepared / Modified by	Reviewed by	Approved by	Section and Text Revised
1.0	20 May 2021	Sien Van Broekhoven	Dries Verschaeve		Full document

Table of Contents

Introduction

This document forms the Plan Of Action for a Hybrid IT Management Framework project. In this document multiple subjects will be described, such as the client organization, the problems this organization faces along with demands they have for a solution, then some explanation about a solution, and a project planning.

The purpose of this document is to provide more insight into why this project is being developed.

The project

This project involves the architecture design for a hybrid IT environment, in which there is central management of systems, and identity management. The eventual solution will be delivered in the form of Infrastructure as Code configuration files, guides, and documentation about this solution. This architecture design and solution will be developed for Hewlett Packard Enterprise.

The purpose of this project is to create a framework that can easily be deployed in multiple client architectures.

Description of the organization

Organization characteristics

Hewlett Packard Enterprise (HPE) is a business to business enterprise technology company. HPE offers solutions across servers, storage, hybrid infrastructure, edge computing, and software and networking equipment, helping companies capitalize on emerging technologies such as AI, IoT and more.

Mission and vision

The mission of Hewlett Packard Enterprise is to help customers use technology to turn ideas into value, and empower them to transform industries, markets, and lives. HPE simplifies Hybrid IT, powers the Intelligent Edge, and provides the expertise to make it all happen.

Core activities

HPE delivers unique, open, and intelligent technology solutions, with a consistent experience across all clouds and edges, to help customers develop new business models, engage in new ways, and increase operational performance.

Organizational structure

Hewlett Packard Enterprise has office locations all over the world. For this segment I will limit the information to the office in Diegem, Belgium.

[censored]

Problem and main- and subquestions

In this segment the problem is described, along with consequences these problems could cause. Some demands are the formulated, with in the end a solution that will be developed.

Problem

Efficiently managing multiple systems, services, and users in a hybrid working environment is not easy. It is hard to keep a clean overview and monitor everything that is happening in the environment. On top of that users might have to log onto different systems with different credentials every time, this can get confusing, and there is no proper overview anymore.

As many GUIs are disappearing from servers, configuring systems via remote desktop services is not always an option. Manually configuring each client is also not a solution as this takes a lot of time.

The “managing” of these systems includes creating new systems, configuring, maintaining, and deleting existing systems.

Possible consequences of the problem:

- Systems are not kept up to date (security vulnerabilities could exist)
- Time is lost with manually configuring GUI-less systems
- There is no proper overview of system states and configurations
- Users write down passwords, which is not secure
- Users are unmotivated to work because they constantly have to log onto different services, there is no centralized server
- Resources are not optimally utilized

Main- and subquestions

The main question is to provide an architecture design that provides centralized identity management and authentication, and also provides centralized management for systems in the environment. It should possible to manage both systems with a GUI and systems without a GUI, through one centralized server. These systems can either be on premises, or they can be virtual clients, so the solution will need to provide support for hybrid environments. This solution should be built with the idea of an existing “on-premise” environment in mind, with the purpose of migrating/expanding to the Azure cloud.

It should be possible to create, manage, configure, and delete clients through a CLI interface or through scripting. Preferably there should be a way to quickly deploy configurations to one/many systems.

Subquestions:

- There should be multiple nodes for failover support, so that downtime is avoided.
- The servers should be configured by making use of the best practices and the Azure Cloud Adoption Framework, this makes sure that efficient methods are used, and configurations are secure.
- Powershell remote scripting should be made possible to be able to quickly enter commands on remote systems.
- Role based access control to servers should be put in place, to make sure that users only have access to the systems they are authorized for. (Principle of least privilege)
- Client credential management and authentication with Microsoft Active Directory. The single sign on feature makes it easier for users to log onto different services, without having to remember different credentials each time.
- Support for monitoring and auditing, to detect possible failures/errors and security threats.
- Password management of server administrator passwords.

Objective and final product

A final solution in the form of a hybrid IT environment architecture design. This design should include a way to centrally manage server systems, provide user authentication and identity management. This design needs to be configured by the standards of Azure's Cloud Adoption framework, and meet security standards.

It should be possible to quickly execute common configurations, such as creating new systems/configuring systems/ destroying systems, through Infrastructure as Code. Remote powershell also has to be a possibility to remotely execute configurations on systems.

Essential systems in the environment need to be prepared for possible failures, and thus be configured with failover support in mind. Access to systems needs to be configured with the principle of least privilege in mind.

The final product consists of several aspects:

1. Infrastructure as code files, written in a modular way. This to be able to quickly deploy aspects of- or the entire virtual environment.
2. Installation manual. This manual will provide all necessary information about the way the environment is configured. This document will be created for IT admins.
3. User manual. This manual will provide all necessary information for a user to log onto their workstation, and how they can access services. This document will be created for clients of the environment.
4. Design guide. This document provides a better overview of the decisions that were made in terms of the infrastructure design.

Solution

The following architecture design was created, based on the user stories and requirements.

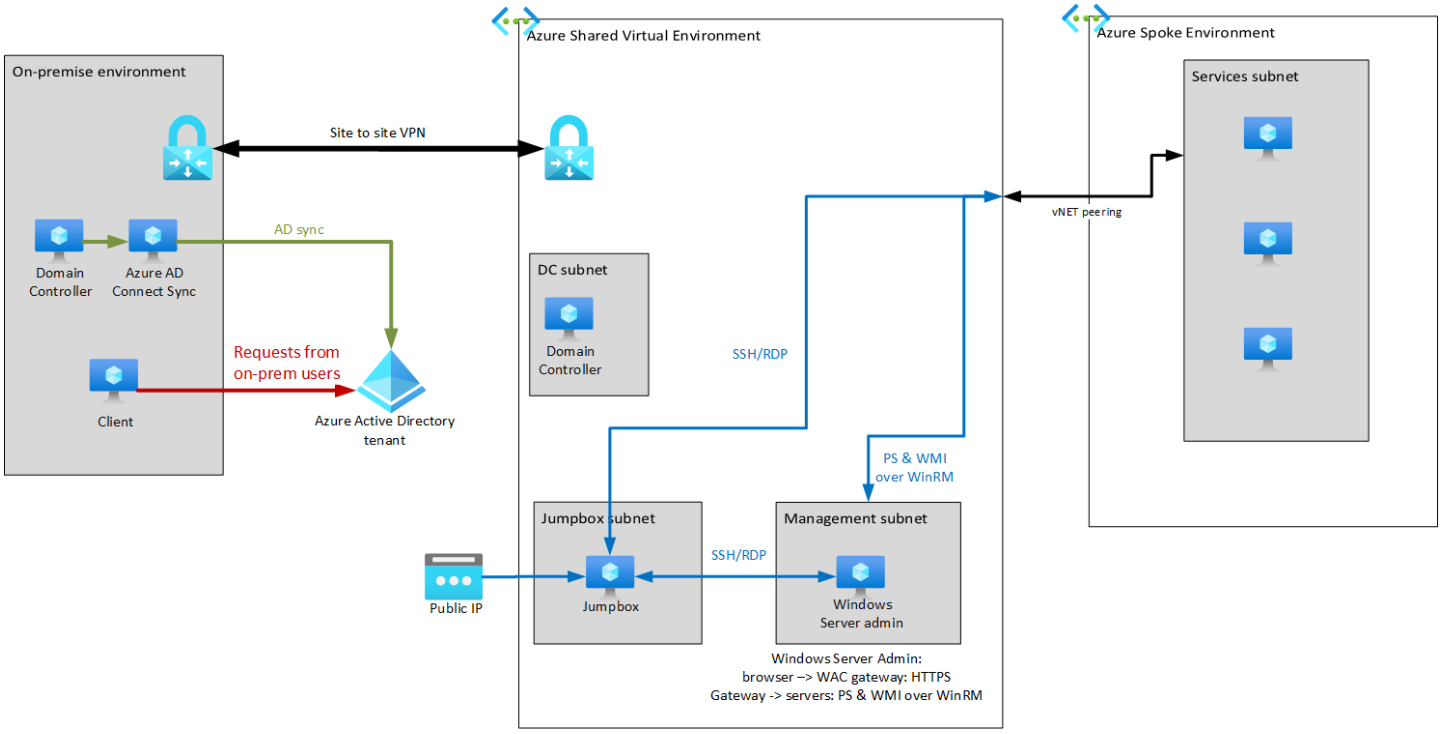

Figure 1: Architecture design

This is a high-level design of how the solution will be configured.

Components

The following components have been researched properly to make sure that they are the best fit to create the solution. The explanation behind why this specific technology is used, can be found in the Design Guide.

This design uses Active Directory (both on-premises, as in the cloud, and Azure AD) to provide centralized identity management and authentication. The second domain controller provides failover support.

Services will be made available through a hub and spoke model. The services are the spokes, and the azure shared virtual environment is the hub.

A windows server which hosts Windows Admin Center will be used as central server to manage all other servers in the environment.

Security measures such as encryption, private IP addresses, network security groups, role-based access control will be configured.

Monitoring and auditing will be done through Azure Monitor, and Azure Alerts for providing alerts. Compliance regulations will be met with Azure Policies, to make sure that best practices are being used.

Password management for local admin accounts of the windows servers will be arranged with Microsoft's Local Administrator Password Solution (LAPS), which stores and protects passwords in Active Directory.

Infrastructure as Code (IaC)

The infrastructure will be deployed, configured, and destroyed with Infrastructure as Code configurations. The technology that will be used for this is HashiCorp Terraform, with Terraform IaC can be written using the declarative HashiCorp Language (HCL).

Powershell scripts can be delivered to systems through Terraform to further configure systems.

Work planning

The planning of the project is divided into two main parts: The research phase and the realization phase.

The research phase was from the beginning of the internship until the end of week four, so until the 26th of April. The realization phase is starting in the final week of the research phase, and ends when the internship ends, so until the 28th of May.

Research steps

Most of the research to find the most fitting technologies for the solution has already been done. A few use cases have been created to test out technologies and gain a better understanding of them.

Along the way new questions will probably arise, and more research will then be done.

Realization steps

The following list shows an overview of what needs to be configured/installed to realize the project.

1. **Base of environment**
   - Resource groups
   - vNETs
   - Subnets
   - NSG
   - NIC
   - VMs (base images)
   - Availability sets
   - VPN gateway
   - vNET peering

This will be configured through Terraform best practices with a modular coding structure

2. Active Directory
   - Install first domain controller + forest
   - Install second domain controller + replication
   - Configure Azure AD
   - Add a local computer
   - Add some users + groups
   - Configure GPOs

3. Windows Admin Center
   - Install Windows Admin Center
   - Add windows servers to WAC

4. Jumphost & Services
   - Configure Azure bastion
   - Configure services
   - Configure connection to services

5. Password Management, security (monitoring)
   - Password management
   - Log analytics
   - Diagnostics storage

- Azure monitor
- Security center

6. Cleanup & Optimization
- Azure policies
- Azure alerts
- AD health monitoring
- Azure advisor
- Cleanup code

7. Extras and finish documentation
- User manual
- Installation manual

Documentation will be written along the way and will be optimized at the end of the project.

Timeframe

The following table is a time estimation of how long these tasks will take, along with starting and ending dates.

Table 1: Estimated planning

Task	Starting date	Ending date	Estimated time needed
Base of environment	23 March	26 April	0.5 week
Active Directory	24 March	9 April	2.5 weeks
Windows Admin Center	5 April	16 April	2 weeks
Jumphost & Services	12 April	23 April	2 weeks
Password Management, security	19 April	30 April	2 weeks
Cleanup & Optimization	26 April	7 Mei	2 weeks
Extras and finish documentation	3 Mei	21 Mei	3 weeks

This planning includes enough time to succesfully complete every task, with some spare time at the end in case of the worst-case scenario where extra time is needed. If all goes according to the planning, this time will be used to implement “nice to haves” and other extras.