

Solution guide: Implementing the Azure blueprint for healthcare



Introduction

Healthcare organizations are realizing that AI (Artificial Intelligence) and ML (machine learning) can be valuable tools for many parts of their business, from improving patient outcomes to streamlining daily operations. Often, healthcare organizations do not have the technology staff to implement AI/ML systems. To improve this situation and get AI/ML solutions running on Azure quickly, Microsoft created the [Azure healthcare AI blueprint](#). Using the blueprint, we show how to get started with AI/ML quickly in a safe, compliant, secure, and reliable way.

The health blueprint for AI bootstraps AI/ML into your organization using Azure. This article describes installing the blueprint, its components, and how to use it to run an AI/ML experiment that predicts a patient's length of stay.

Benefits

The blueprint was created to give healthcare organizations guidance and a quick start on proper [PaaS](#) (Platform as a Service) architectures to support AI/ML in highly regulated healthcare environments, including ensuring the system upholds HIPAA and HITRUST compliance requirements.

Technology staff in healthcare organizations often have little time for new projects, especially those in which they must learn a new and complex technology. The blueprint can help technical staff become familiar with Azure and several of its services quickly, saving the cost of a learning curve. Technical staff can learn from the blueprint as a reference implementation after it is installed and use that knowledge to extend its capabilities or create a new AI/ML solution patterned after the blueprint.

The blueprint gets your organization up and running with new AI/ML capabilities—quickly. With AI and ML in place, technical staff are ready to run AI/ML experiments using data collected through various sources. For instance, data may already exist on previous instances of sepsis and many of the accompanying variables that were tracked for individual patients with the condition. Using this data in an anonymized format, technical staff can look for indicators of potential sepsis in patients and help change operational procedures to better avoid the condition.

The blueprint provides the data and example code for learning how to predict a patient's length of stay. This is a sample use case that can be used for learning about the components of the AI/ML solution.

Platform or infrastructure as a service

Microsoft Azure offers both PaaS and SaaS offerings and choosing the right one for your needs differs per use case. The blueprint is designed to use PaaS services that solve for predicting a patient's length of stay in hospital. The Azure healthcare AI blueprint provides everything needed to instantiate a secure and compliant AI/ML solution pre-configured for healthcare organizations. The PaaS model used by this blueprint installs and configures the blueprint as a complete solution.

PaaS option

Using a PaaS services model results in reduced Total Cost of Ownership (TCO) because there is no hardware to manage. The organization doesn't need to buy and maintain hardware or VMs. The blueprint uses PaaS services exclusively.

This reduces the cost of maintaining an on-premises solution and frees technical staff to focus on strategic initiatives instead of infrastructure. It can also move paying for computing and storage from capital expense budgets to operational expense budgets. The costs of running this blueprint scenario are driven by usage of the services plus the costs of data storage.

IaaS option

Although the blueprint and this article focus on the PaaS implementation, there is an [open source extension](#) to the blueprint which allows using it in an infrastructure as a service (IaaS) environments.

In an IaaS hosting model, customers pay for uptime of Azure hosted VMs and their processing power. IaaS gives a higher level of control since the customer is managing their own VMs, but typically at increased costs as VMs are charged for uptime versus usage. Further, the customer is responsible for maintaining the VMs by applying patches, guarding against malware and so on.

The IaaS model is beyond the scope of this article, which focuses on a PaaS deployment of the blueprint.

The healthcare AI/ML blueprint

The blueprint creates a starting point for using this technology in a healthcare context. When the blueprint is installed to Azure, all resources, services and several user accounts are created to support the AI/ML scenario with appropriate actors, permissions, and services.

The blueprint includes an AI/ML experiment to predict a patient’s length of stay, which can help in forecasting staffing, bed counts, and other logistics. The package includes installation scripts, example code, test data, security and privacy support and more.

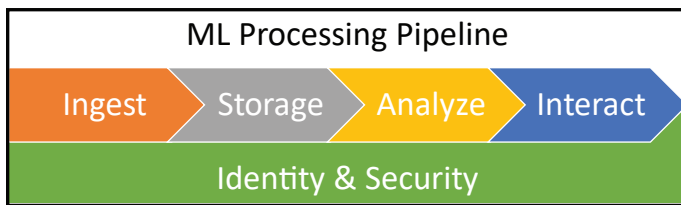
Blueprint technical resources

The resources below are all found in this [GitHub repository](#).

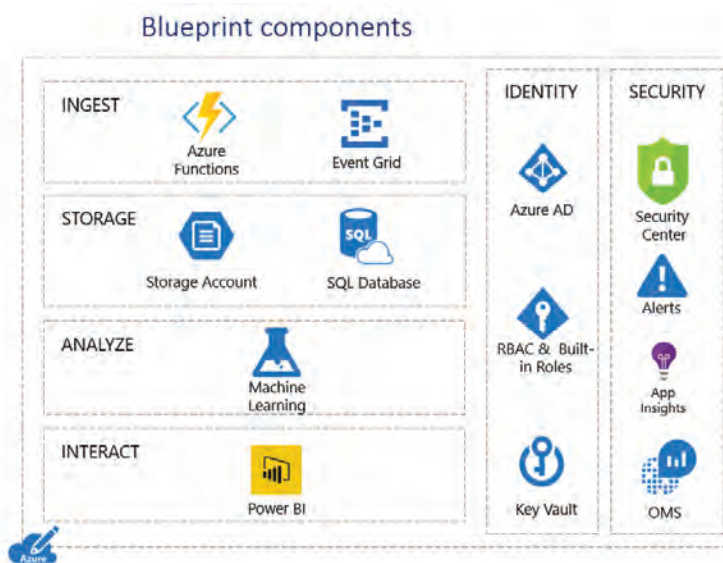
Primary resources are:

1. [PowerShell scripts](#) for deployment, configuration, and other tasks.
2. [Detailed instructions for installation](#) which include how to use the install script.
3. [A comprehensive FAQ](#).

Cross cutting concerns for this model include identity and security, both of which are especially important when dealing with patient data. The components of the ML pipeline are shown in this graphic.



The graphic below shows the Azure products that are installed. Each resource or service provides a component of the AI/ML processing solution, including the cross-cutting concerns of identity and security.



Implementing a new system in a regulated healthcare environment is complex. For example, ensuring all aspects of the system are HIPAA compliant and HITRUST certifiable takes more than developing a lightweight solution. The blueprint installs identification and resource permissions to help with these complexities.

The blueprint also provides additional scripts and data used to simulate and study the results of admitting or discharging patients. These scripts allow staff to immediately begin to learn how to implement AI and ML using the solution in a safe, isolated scenario.

Additional blueprint resources

The blueprint provides exceptional guidance and instructions for technical staff and also includes artifacts to help create a fully functional installation. These other artifacts include:

1. A [threat model](#) for use with the [Microsoft Threat Modeling Tool](#). This threat model shows components of the solution, the data flows between them, and the trust boundaries. The tool can be used for threat modeling by those looking to extend the base blueprint or for learning about the system architecture from a security perspective.
2. The [HITRUST customer responsibility matrix](#), in an Excel workbook. This shows what you (the customer) must provide versus what Microsoft provides for each requirement in the matrix. More information about this responsibility matrix is included in this article in the Security and Compliance > Blueprint responsibility matrix section of this document.
3. The [HITRUST health data and AI review](#) whitepaper examines the blueprint through the lens of requirements to be met for HITRUST certification.
4. The [HIPAA health data and AI review whitepaper](#) reviews the architecture with HIPAA regulations in mind.

These resources are [here on GitHub](#).

Installing the blueprint

There is little time investment to get up and running with this blueprint solution. A bit of PowerShell scripting knowledge is recommended, but step by step instructions are available to help guide the installation so technologists will be successful deploying this blueprint regardless of their scripting skills.

Technical staff can expect to install the blueprint with little experience using Azure in 30 minutes to an hour.

The installation script

The blueprint provides [exceptional guidance and instructions for installation](#). It also provides scripting for install and uninstall of the blueprint services and resources. Calling the PowerShell deployment script is simple. Before the blueprint is installed, certain data must be collected and used as arguments to the `deploy.ps1` script as show below.

```
.\deploy.ps1 -deploymentPrefix '<prefix> `
    -tenantId <tenant id> `                # also known as the AAD directory
    -tenantDomain <tenant domain> `
    -subscriptionId <subscription id> `
    -globalAdminUsername <user id> `      # ID from your AAD account
    -deploymentPassword <universal password> ` # applied to all new users and service accounts
    -appInsightsPlan 1                    # we want app insights set up
```

The installation environment

Important! Do not install the blueprint from a machine outside of Azure. The install is much more likely to succeed if you create a clean Windows 10 (or other Windows VM) in Azure and run the install scripts from there. This technique uses a cloud-based VM to mitigate latency and help to create a smooth installation.

During installation, the script calls out to other packages to load and use. When installing from a VM in Azure, the lag between the installation machine and the target resources will be much lower. However, some of the scripting packages downloaded are still vulnerable to latency as script packages live outside the Azure environment—which may lead to time-out failures.

Install failure! (don't panic)

The installer downloads some external packages during installation. Sometimes, a script resource request will time out due to lag between the install machine and the package. When this happens, you have two choices:

1. Run the install script again with no changes. The installer checks for already allocated resources and installs only those needed. While this technique can work, there is a risk the install script will try to allocate resources already in place. This can cause an error and the installation will fail.
2. You still run the **deploy.ps1** script, but pass different arguments for uninstalling the blueprint services.

```
.\deploy.ps1 -clearDeploymentPrefix <prefix> `
             -tenantId <value> `
             -subscriptionId <value> `
             -tenantDomain <value> `
             -globalAdminUsername <value> `
             -clearDeployment
```

After the uninstall is done, change the prefix in the install script and try installing again. The latency issue may not occur again. If the installation fails while downloading script packages, run the uninstaller script and then the installer again.

After running the uninstall script, the following will be gone.

- Users installed by the installer script
- The resource groups and their respective services are gone, including data storage
- The application registered with AAD (Azure Active Directory)

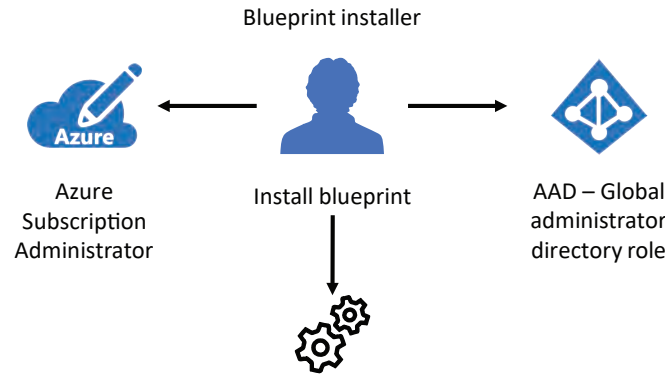
Note the Key Vault is held as a "soft delete" and while it isn't seen in the portal, it doesn't get deallocated for 30 days. This enables reconstituting the Key Vault if needed. To learn more about the implications of this and how to handle it, see the [Technical Issues > Key Vault](#) section of this article.

Reinstall after an uninstall

If there is a need to reinstall the blueprint after an uninstall, you must change the prefix in the next deployment as the uninstalled Key Vault will cause an error if you do not change the prefix. More about this is covered below, in the **Technical Issues: Key Vault** section of this article.

Required administrator roles

The person installing the blueprint must be in the [Global Administrator role](#) in the AAD. The installing account must also be an [Azure subscription administrator](#) for the subscription being used. If the person doing the install is not in both of these roles, the install will fail.



Further, the install is not designed to work with MSDN subscriptions due to the tight integration with AAD. A standard Azure account must be used. If needed, [get a free trial](#) with credit to spend for installing the blueprint solution and running its demos.

Adding other resources

The Azure blueprint installation doesn't include more services than those needed to implement the AI/ML use case. However, more resources or services can be added to the Azure environment, making it a good test bed for additional initiatives, or a starting point for a production system. For instance, one might add other PaaS services or IaaS resources in the same subscription and AAD.

New resources, like [Cosmos DB](#) or a new [Azure Functions](#), may be added to the solution as more Azure capabilities are needed. When adding new resources or services, ensure they are configured to meet security and privacy policies to remain compliant with regulations and policy.

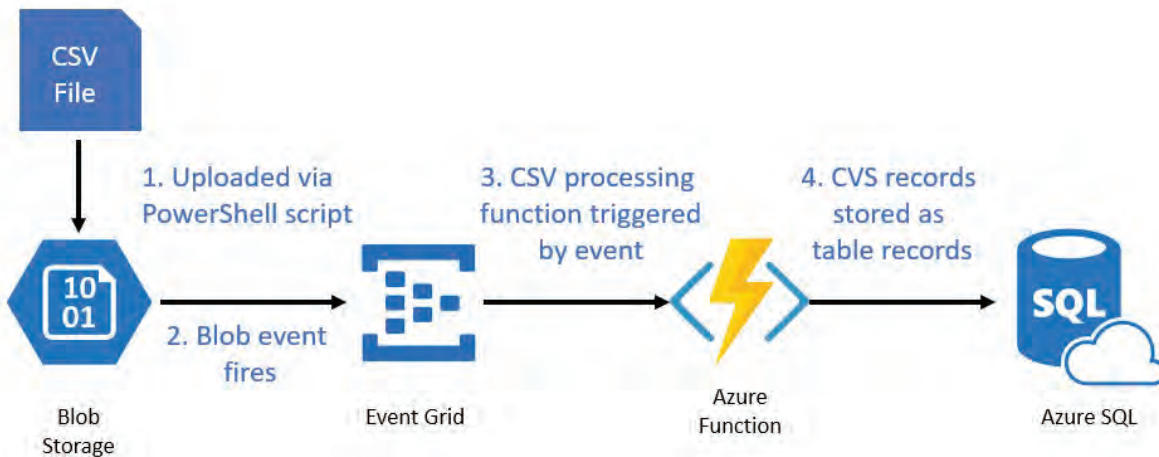
New resources and services may be created with [Azure REST APIs](#), [Azure PowerShell scripting](#), or by using the [Azure Portal](#).

Using machine learning with the blueprint

The blueprint was built to demonstrate an ML scenario with a regression algorithm used in a model to predict [a patient's length of stay](#). This is a common prediction for healthcare providers to run as it helps in scheduling staffing and other operational decisions. Further, anomalies can be detected over time when an average length of stay for a given condition rises or declines.

Ingesting training data

With the blueprint installed and all services working properly, the data to be analyzed can be ingested. 100,000 patient records are [available for ingest and](#) working with the model. Ingesting patient records is the first step in the using [Azure Machine Learning Studio](#) to run the patient length of stay experiment as shown below.



The blueprint includes an experiment and the necessary data to run an ML job in Machine Learning Studio (MLS). The example uses a trained model in an experiment to forecast patient length of stay based on many variables.

In this demonstration environment, the data ingested into the Azure SQL database is free from any defects or missing data elements. This data is clean. Often, unclean data is ingested and must be “cleaned” before it can be used for feeding a machine learning training algorithm, or before using the data in a ML job. Missing data or incorrect values in the data will negatively impact results of the ML analysis.

Azure Machine Learning Studio

Many healthcare organizations don’t have the technical staff to focus on ML projects. This often means valuable data is left unused or expensive consultants are brought in to create ML solutions.

AI/ML experts as well as those learning about AI/ML can use Azure Machine Learning Studio to design experiments. MLS is a web-based design environment used to create ML experiments. With MLS, you can create, train, evaluate, and score models, saving precious time when using different tools to develop models.

MLS offers a complete toolset for ML workloads. This means people new to ML can get a jumpstart using the tool and produce results faster than with other ML tools. That lets your IT staff focus on providing value elsewhere and without bringing in a ML specialist. This capability in your own healthcare organization means various hypotheses can be tested and the resulting data analyzed for actionable insights, like patient interventionism offers [pre-written modules](#) to be used on a drag and drop canvas, visually composing end-to-end data-science workflows as experiments. There are [pre-written modules](#) that encapsulate specific algorithms such as decision trees, decision forests, clustering, time series, anomaly detection and others.

Custom modules can be added to any experiment. These are written in the [R language](#) or in [Python](#). This allows using pre-built modules as well as custom logic to create a more sophisticated experiment.

MLS enables [creating and using learning models](#), as well as providing a set of pre-designed experiments for use in common applications. Additionally, new experiments can be added to MLS without changing any of the blueprint’s resources.

To save time, visit the [Azure AI Gallery](#) to find ready-to-use ML solutions for specific industries, including healthcare. For example, the gallery includes solutions and experiments for breast cancer detection and heart disease prediction.

Security and compliance

Security and compliance are two of the most important things to be mindful of when creating, installing or managing software systems in a healthcare environment. The investment made in adopting a software system can be undercut by not meeting required security policies and certifications.

Although this article and the healthcare blueprint focus on technical security, other types of security are also important including physical security and administrative security. These security topics are beyond the scope of this article, which focuses on the blueprint's technical security.

Principle of least privilege

The blueprint installs named users with roles to support and limit their access needs to resources in the solution. This model is known as the "principle of least privilege," an approach to resource access in system design. The principle states that service and user accounts should have access to only those systems and services needed for a legitimate purpose.

This security model ensures the system's compliance with HIPAA and HITRUST requirements, removing risk to the organization.

Defense in depth

System designs using multiple abstraction layers of security controls are using defense in depth. Defense in depth provides security redundancy at multiple levels. It means you are not dependent on a single layer of defense. It ensures that user and service accounts have appropriate access to resources, services and data. Azure provides security and monitoring resources at every level of system architecture to provide defense in depth for the entire landscape of technologies.

In a software system, like the one installed by the blueprint, a user may login but not have permission to a specific resource. This example of defense in depth is provided by RBAC (Role Based Access Control) and AAD, supporting the principle of least privilege.

Two-factor authentication is also a form of technical defense in depth and may be optionally included when the blueprint is installed.

Azure Key Vault

The [Azure Key Vault](#) service is a container—vault—used to store secrets, certificates, and other data used by applications. These include database strings, rest endpoint URLs, API keys, and other things developers don't want to hard-code into an application or distribute in a .config file.

Additionally, vaults are accessible by application service identities or other accounts in with AAD permissions. This allows secrets to be accessed at runtime by applications needing a vault's contents.

Keys stored in a vault may be encrypted or signed, and key usage can be monitored for any security concerns.

If a Key Vault is deleted, it is not immediately purged from Azure. Implications of this are covered in the [Technical Issues > Key Vault](#) section of this article.

Application Insights

Healthcare organizations often have mission and life-critical systems that must be reliable and resilient. Anomalies or disruptions in service must be detected and corrected as soon as possible. [Application Insights](#) is an Application Performance Management (APM) technology that monitors applications and sends alerts when something goes wrong. It monitors applications at runtime for errors or application anomalies. It is designed to work with multiple programming languages and provides a rich set of capabilities to help ensure applications are healthy and running smoothly.

For example, an application may have a memory leak. Application Insights can help find and diagnose issues like this through the rich reporting and KPIs it monitors. Application Insights is a robust APM service for application developers.

This [interactive demo](#) shows key features and capabilities of Application Insights, including a comprehensive monitoring dashboard which can be used by technologists in the health organization to monitor application state and health.

Azure Security Center

Real time security and KPI monitoring is a necessity in mission critical applications. [Azure Security Center](#) (ASC) helps ensure your Azure resources are secure and protected. ASC is a security management and advanced threat protection service. It can be used to apply security policies across your workloads, limit your exposure to threats and detect and respond to attacks.

Security Center standard provides the following services.

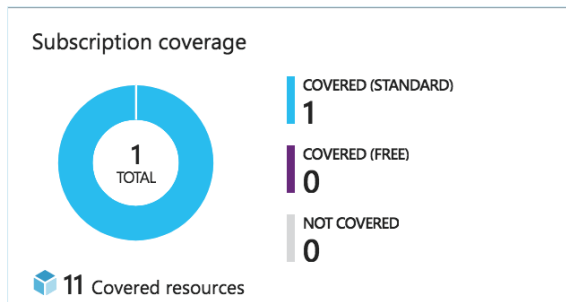
- **Hybrid security** – Get a unified view of security across all your on-premises and cloud workloads. This is especially helpful in hybrid cloud networks used by healthcare organizations with Azure.
- **Advanced threat detection** – ASC uses advanced analytics and the [Microsoft Intelligent Security Graph](#) to get an edge over evolving cyber-attacks and mitigate them right away.
- **Access and application controls** - Block malware and other unwanted applications by applying whitelisting recommendations for your specific workloads and powered by machine learning.

In the context of the Health AI blueprint, ASC analyzes the system components and provides a dashboard showing vulnerabilities in services and resources in the subscription. Distinct dashboard elements provide visibility into a solution's concerns as follows.

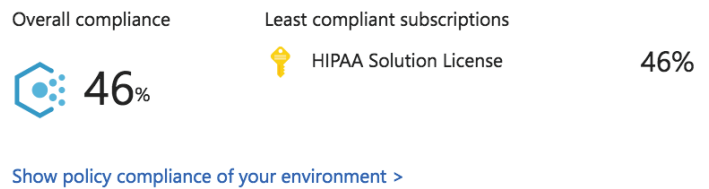
- Policy and compliance
- Resource security hygiene
- Threat protection

Below is an example dashboard identifying 13 suggestions for improving system threat vulnerabilities. It also shows a mere 46% compliance with HIPAA and policy.

Policy & compliance

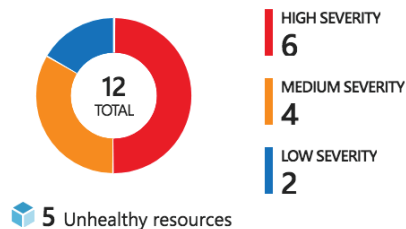


Policy compliance

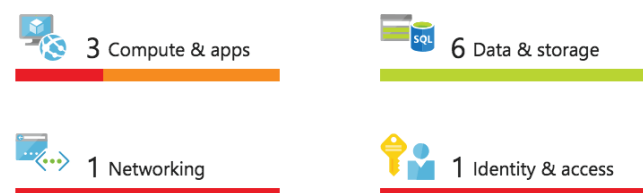


Resource security hygiene

Recommendations















Resource health monitoring



Drilling into the high severity security problems shows what resources are affected and the remediation needed for each resource, as shown below.

Many hours can be spent by IT staff trying to manually secure all resources and networks. With ASC to identify vulnerabilities in a given system, time can be spent in other strategic pursuits. For many of the vulnerabilities identified, ASC can automatically apply the remediating action and secure the resource without an administrator having to dig deeply into the problem.

DESCRIPTION	RESOURCE	SEVERITY
Enable MFA for accounts with owner permissions on your subscription (Preview)	 1 subscription	 High
Endpoint Protection not installed on Azure VMs	 1 virtual machine	 High
Apply disk encryption	 1 virtual machine	 High
Add a Next Generation Firewall	 ds-vm-02-ip	 High
Enable Network Security Groups on subnets	 default	 High
Apply a Just-In-Time network access control	 ds-vm-02	 High

ACS does even more through its threat detection and alerting capabilities. Use ACS to monitor networks, machines, and cloud services for incoming attacks and post-breach activity to keep your environment secure.

ASC automatically collects, analyzes, and integrates security information and logs from a variety of Azure resources. The ML capabilities in ASC allow it to detect threats manual approaches would not reveal. A list of prioritized security alerts is shown in ASC along with the information needed to quickly investigate the problem along with recommendations for how to remediate an attack.

RBAC security

[Role Based Access Control](#) (RBAC) provides or denies access to protected resources, sometimes with specific rights per resource. This ensures only appropriate users can access their designated system components.

For example, a database administrator may have access to a database containing encrypted patient data whereas a health care provider may only have access to appropriate patient's records through the application that displays them. This is typically an Electronic Medical Record or Electronic Health Record system. The nurse has no need to access the databases and the database administrator has no need to see a patient's health record data.

To enable this, RBAC is part of Azure security and enables precisely focused access management for Azure resources. Fine-grained settings for each user enable security and systems administrators to be very exact in the rights they give each user.

Blueprint responsibility matrix

The [HITRUST customer responsibility matrix](#) is an Excel document that supports customers implementing and documenting security controls for systems built on Azure. The workbook lists the relevant HITRUST requirements and explains how Microsoft and the customer are responsible for meeting each one.

Understanding the shared responsibility for implementing security controls in a cloud environment is essential for customers building systems on Azure. Implementing a specific security control may be the responsibility of Microsoft, the responsibility of customers, or a shared responsibility between Microsoft and customers. Different cloud implementations affect the way responsibilities are shared between Microsoft and customers.

See the responsibilities table below for examples.

Azure is responsible for implementation, management, and monitoring of information protection program methods and mechanisms in relation to its service provision environment.

The customer is responsible for implementation, configuration, management, and monitoring of information protection program methods and mechanisms for customer-controlled assets used to access and consume Azure services.

Azure is responsible for implementation, configuration, management, and monitoring of account management methods and mechanisms in relation to its service provision environment.

The customer is also responsible for account management of deployed Azure virtual machine instances and resident application components.

These are only two examples of the many responsibilities to be considered when deploying cloud systems. The HITRUST customer responsibility matrix is designed to support an organization's HITRUST compliance with an Azure system implementation.

Customization

It is common to customize the blueprint after it is installed. Reasons and techniques to customize the environment vary.

The blueprint may be customized before installation by modifying the install scripts. While this is possible, it is advisable to create independent PowerShell scripts to run after the initial install is complete. New services may also be added to the system through the portal once the initial installation has taken place.

Customizations may include any or more of the following.

- Adding new experiments to Machine Learning Studio
- Adding additional unrelated services to environment
- Modifying data ingestion and the ML experiment output to use a different data source than the Azure SQL patientdb database
- Providing production data to the ML experiment
- Cleaning any proprietary data being ingested to match that needed by the experiment

Customizing the installation is no different than working with any Azure solution. Services or resources may be added or removed, providing new capabilities. When customizing the blueprint, take care not to alter the overall ML pipeline to ensure the implementation continues working.

Technical Issues

The following issues can cause the blueprint installation to fail or to install in an undesirable configuration.

Key Vault

Key vaults are unique when deleting an Azure resource. Vaults are kept by Azure for recovery purposes. Accordingly, a different prefix must be passed into the install script each time the install script is run, or the install will fail due to a collision with the old vault name. Key vaults, and all other resources, are named using the prefix you provide to the install script.

A Key Vault created by the installation script is retained as a "soft delete" for 30 days. Although not currently accessible through the portal, soft deleted [Key Vaults are manageable from PowerShell](#), and may even be deleted manually.

Azure Active Directory

It is strongly recommended that you install the blueprint in an empty AAD rather than into a production system. Create a new AAD instance and use its tenant id during installs to avoid adding blueprint accounts to your live AAD instance.

Technologies presented

- Learn more about the [Azure Health Data and AI blueprint](#).
- Download, clone or fork the [GitHub repo here](#).
- [Machine Learning Studio](#) is the workspace and tool data scientists use to create Machine Learning experiments. It allows using built-in algorithms, special purpose widgets, and Python and R scripts.
- Secrets, certificates, and other private data is held in [Azure Key Vault](#).
- The scripting language [PowerShell](#) is instrumental to setting up the blueprint, although needed commands are presented in the installation instructions.
- [Azure AI Gallery](#) provides a recipe box of AI/ML solutions useful for customers by their industry. There are several solutions published by data scientists along with other experts for healthcare.
- [Azure Security Center](#) provides insights into your application's behavior, vulnerabilities, and mitigation techniques.
- The [Microsoft Threat Modeling Tool](#) is used to plan and predict threats to your system environment. It is needed to review the threat model included with the blueprint.

Next steps

The [Azure Health Data AI blueprint](#) is a complete ML solution and can be used as a learning tool for technologists to better understand Azure and how to ensure systems conform to healthcare regulatory requirements. It can also be used as a starting point for a production system using Azure Machine Learning Studio as the focal point.

[Download the blueprint](#) to get started with your implementation in hours, not days or weeks. If you have problems with your install or questions about blueprint, [visit the FAQ page](#).

Download the supporting collateral to gain a better understanding of the blueprint implementation beyond the installation and ML experiment. This collateral includes the following.

1. [HITRUST customer responsibility matrix](#)
2. [The comprehensive threat model](#)
3. [HITRUST health data and AI review whitepaper](#)
4. [HIPAA health data and AI review](#)

© 2018 Microsoft Corporation. All rights reserved.

This document is provided "as is." Information and views expressed in this document, including URL and other internet website references, may change without notice. You bear the risk of using it.

This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.