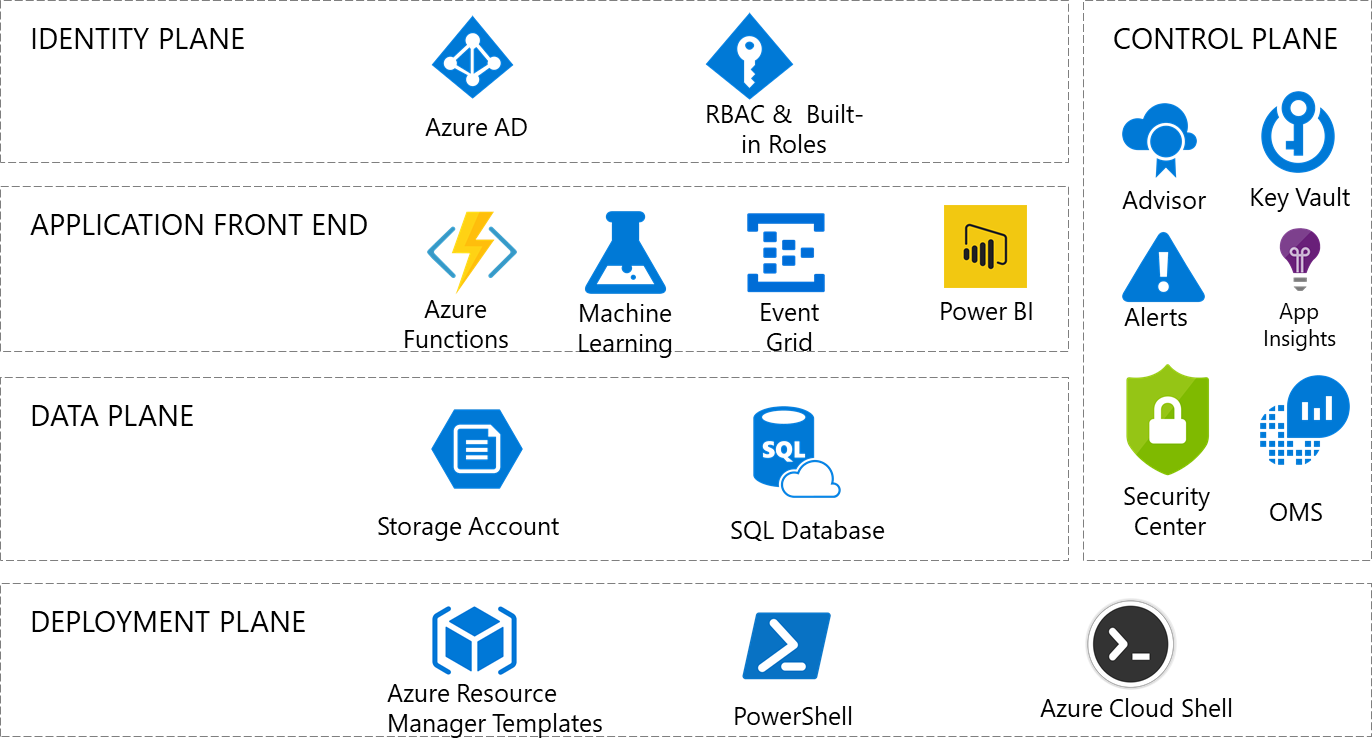
Azure Blueprint for Healthcare

# Overview

**The Azure Blueprint program offers turn-key compliance solutions and support, tailored to the needs of industry verticals, that accelerate cloud adoption and utilization for customers with regulated or restricted data. These resources enable customers to deploy secure and compliant applications in less time, and at a lower cost, when compared to on premises solutions or competitive cloud platforms.**

The Azure Blueprint for Healthcare provides tools and guidance for deploying a secure, HIPAA-compliant platform-as-a-service (PaaS) environment for storing and working with medical records in a secure, compliant multi-tier cloud environment deployed as an end-to-end solution. It showcases a common reference architecture and is designed to simplify adoption of Microsoft Azure. This foundational architecture illustrates an end-to-end solution to meet the needs of organizations seeking a cloud-based approach to reducing the burden and cost of deployment.



This foundational architecture meets the requirements of the Health Insurance Portability and Accountability Act (HIPAA) for protecting the confidentiality of medical records and other patient data. It can accept and output data in Fast Healthcare Interoperability Resources (FHIR) format, a worldwide standard for exchanging healthcare information electronically. Customers can use Azure Machine Learning to take advantage of powerful business intelligence tools and analytics. As an example of the kind of analysis Azure Machine Learning can facilitate, the blueprint includes scripts and tools for predicting the length of a patient’s stay in a hospital facility.

This blueprint is intended to serve as a foundation for customers to adjust to their specific requirements. It is designed to be secure and compliant when deployed; however, customers are responsible for configuring roles correctly and implementing any modifications in a secure and compliant way. Please note the following:

* This foundational architecture provides a baseline to help customers use Microsoft Azure in a HIPAA-compliant manner.
* Although the blueprint is designed to be compliant with HIPAA (through NIST SP 800-66) and HITRUST (through the Common Security Framework – CSF), it should not be considered compliant until certified by an external auditor per HIPAA and HITRUST certification requirements.
* Customers are responsible for conducting appropriate security and compliance reviews of any solution built using this foundational architecture.

For a quick overview of how this solution works, watch this video explaining and demonstrating its deployment.

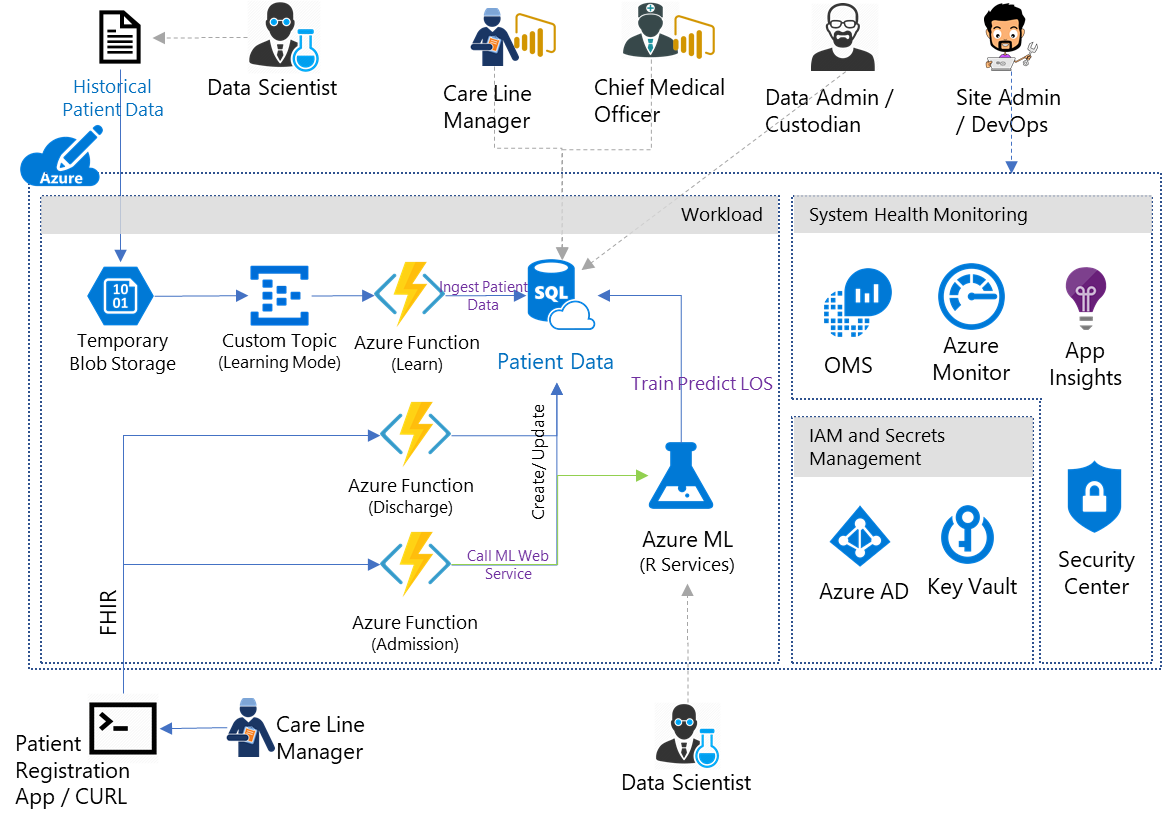
# Solution components

The foundational architecture is comprised of the following components:

* **Architectural diagram.** The diagram shows the reference architecture used for the blueprint and use case scenario.
* **Deployment templates**. In this deployment, [Azure Resource Manager templates](https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-group-overview#template-deployment) are used to automatically deploy the components of the architecture into Microsoft Azure by specifying configuration parameters during setup.
* **Automated deployment scripts**. These scripts help deploy the end-to-end solution. The scripts consist of:
* A module installation and [global administrator](https://docs.microsoft.com/en-us/azure/active-directory/active-directory-assign-admin-roles-azure-portal) setup script is used to install and verify that required PowerShell modules and global administrator roles are configured correctly.
* An installation PowerShell script is used to deploy the end-to-end solution, provided via a .zip file and a .bacpac file that contain a pre-built demo web application with SQL database sample content. The source code for this solution is available for review [here](https://github.com/Microsoft/azure-sql-security-sample).
* **Threat model.** A comprehensive threat model is provided in .tm7 format for use with the [Microsoft Threat Modeling Tool](https://www.microsoft.com/en-us/download/details.aspx?id=49168), showing the components of the solution, the data flows between them, and the trust boundaries. The model can help customers understand the points of potential risk in the system infrastructure when developing machine learning components or other modifications.
* **Customer responsibility matrix.** A Microsoft Excel workbook lists the relevant HITRUST requirements and explains how Microsoft and the customer are responsible for meeting each one.
* **Healthcare compliance review.** […]

[pricing information]

# Architectural diagram



# Roles

The blueprint defines two roles for administrative users (operators), and three roles for users in hospital management and patient care. A sixth role is defined for an auditor to evaluate compliance with HIPAA and other regulations. Azure Role-based Access Control (RBAC) enqbles precisely focused access management for each user of the solution through built-in and custom roles. See [Get started with Role-Based Access Control in the Azure portal](https://docs.microsoft.com/azure/active-directory/role-based-access-control-what-is) and [Built-in roles for Azure role-based access control](https://docs.microsoft.com/azure/active-directory/role-based-access-built-in-roles) for detailed information about RBAC, roles, and permissions.

## Site Administrator

The site administrator is responsible for the customer’s Azure subscription. They control the overall ecosystem, but have no access to patient records.

* Default role assignments: [Owner](https://docs.microsoft.com/azure/active-directory/role-based-access-built-in-roles#owner)
* Custom role assignments: N/A
* Scope: Subscription

## Database Analyst

The database analyst administers the SQL Server instance and database. They have no access to patient records.

* Built-in role assignments: [SQL DB Contributor](https://docs.microsoft.com/azure/active-directory/role-based-access-built-in-roles#sql-db-contributor), [SQL Server Contributor](https://docs.microsoft.com/azure/active-directory/role-based-access-built-in-roles#sql-server-contributor)
* Custom role assignments: N/A
* Scope: ResourceGroup

## Data Scientist

The data scientist operates the Azure Machine Learning service. They can import, export, and manage data, and run reports. The data scientist has access to patient data, but does not have administrative privileges.

* Built-in role assignments: [Storage Account Contributor](https://docs.microsoft.com/azure/active-directory/role-based-access-built-in-roles#storage-account-contributor)
* Custom role assignments: N/A
* Scope: ResourceGroup

## Chief Medical Information Officer (CMIO)

The CMIO straddles the divide between informatics/technology and healthcare professionals in a healthcare organization. Their duties typically include using analytics to determine if resources are being allocated appropriately within the organization.

* Built-in role assignments: ???
* Custom role assignments: N/A
* Scope: Subscription

## Care Line Manager

The care line manager is directly involved with the care of patients. This role requires monitoring the status of individual patients as well as ensuring that staff is available to meet the specific care requirements of their patients. The care line manager is responsible for adding and updating patient records.

* Built-in role assignments: ???
* Custom role assignments: N/A
* Scope: ResourceGroup

## Auditor

The auditor evaluates the solution for compliance. They have no direct access to the network.

* Built-in role assignments: [Reader](https://docs.microsoft.com/azure/active-directory/role-based-access-built-in-roles#reader)
* Custom role assignments: N/A
* Scope: Subscription

[matrix of roles and permissions?]

# Use case

The use case included with this blueprint illustrates how the Azure Blueprint for Healthcare can be used to enable machine learning and analytics on healthcare data in the cloud. Contosoclinic is a small hospital network with three locations in the United States. Each location has 50 beds and is geared toward short- and mid-term care. The hospital network administrators want to use Azure Machine Learning to better predict the length of a patient’s stay at the time of admittance, in order to increase operational workload efficiency and enhance the quality of care it can provide.

## Predicting length of stay

The use case solution uses Azure Machine Learning to predict a newly admitted patient’s length of stay by comparing the medical details taken at patient intake to aggregated historical data from previous patients. The blueprint includes a large set of anonymized medical records to demonstrate the training and predictive capabilities of the solution. In a production deployment, customers would use their own records to train the solution for more accurate predictions reflecting the unique details of their environment, facilities, and patients.

## Users and roles

**Site Administrator – Alex**

*Email: Alex\_SiteAdmin*

Alex’s job is to evaluate technologies that can reduce the burden of managing an on-premise network and reduce costs for management. Alex has been evaluating Azure for some time, but has struggled to configure the services that he needs to meet Han’s compliance requirements. Alex has just selected the Azure Blueprint for Healthcare to deploy a compliance-ready healthcare solution.

**Data Scientist – Debra**

*Email: Debra\_DataScientist*

Debra is in charge of using and creating models that analyze medical records to provide insights into patient care. Debra uses SQL and the R statistical programming language to create her models.

**Database Analyst – Danny**

*Email: Danny\_DBAnalyst*

Danny is the main contact for anything regarding the Microsoft SQL Server that stores all the patient data for Contosoclinic. Danny is an experienced SQL Server administrator who has recently become familiar with Azure SQL Database.

**Chief Medical Information Officer – Caroline**

*Email:*

Caroline needs to be able to determine which facilities are being overtaxed and, specifically, what resources at those facilities may need to be bolstered to realign such resources with demand. Caroline will use the predictions from the length-of-stay (LOS) solution to determine if resources are being allocated appropriately in the hospital network. For example, using the dashboard provided in this solution, Caroline will be able to determine which facilities are not discharging patients at the rate that they are coming in. Using this knowledge, she can then make recommendations to transfer or re-route incoming patients to other facilities when necessary.

**Care Line Manager – Chris**

*Email: Chris\_CareLineManager*

As the individual directly responsible for managing patient discharges at one of the Contosoclinic hospitals, Chris will use the predictions generated by the LOS solution to ensure that adequate staff are available to provide care to patients while they are staying in the facility.

**Auditor – Han**

*Email: Han\_Auditor*

Han is a certified auditor who has experience auditing for ISO, SOC, and HiTrust. He was hired to review Contosoclinc’s network. Han will interview the staff and work with them to ensure that the blueprint and LOS solution meet HiTrust and HIPAA business associate agreement (BAA) requirements.

# Operations and security configuration

This section details the default configurations and security measures built into the Azure Blueprint for Healthcare.

## Azure Active Directory and role-based access control (RBAC)

**Authentication:**

* [Azure Active Directory (Azure AD)](https://azure.microsoft.com/services/active-directory/) is the Microsoft's multi-tenant cloud-based directory and identity management service. All users for the solution were created in Azure Active Directory, including users accessing the SQL Database.
* Authentication to the application is performed using Azure AD. For more information, see [Integrating applications with Azure Active Directory](https://docs.microsoft.com/en-us/azure/active-directory/develop/active-directory-integrating-applications).
* [Azure Active Directory Identity Protection](https://docs.microsoft.com/en-us/azure/active-directory/active-directory-identityprotection) detects potential vulnerabilities affecting your organization’s identities, configures automated responses to detected suspicious actions related to your organization’s identities, and investigates suspicious incidents and takes appropriate action to resolve them.
* [Azure Role-based Access Control (RBAC)](https://docs.microsoft.com/en-us/azure/active-directory/role-based-access-control-configure) enables precisely focused access management for Azure. Subscription access is limited to the subscription administrator, and Azure Key Vault access is limited to the site administrator. Strong passwords (12 characters minimum with at least one letter, number, and special character) are required.
* Multi-factor authentication is supported if the -enableMFA switch is set during deployment.
* Passwords expire after 60 days if the -enableADDomainPasswordPolicy switch is set during deployment.

**Roles:**

* The solution makes use of [built-in roles](https://docs.microsoft.com/en-us/azure/active-directory/role-based-access-built-in-roles) to manage access to resources.
* All users are assigned specific built-in roles by default.

## Event Grid

The solution supports Azure Event Grid, a single service for managing routing of all events from any source to any destination, providing:

* [Security and authentication](https://docs.microsoft.com/en-us/azure/event-grid/security-authentication)
* [Role-based access control](https://docs.microsoft.com/en-us/azure/event-grid/security-authentication#management-access-control) for various management operations such as listing event subscriptions, creating new ones, and generating keys
* Auditing

## SQL Database and Server

* [Transparent Data Encryption (TDE)](https://docs.microsoft.com/en-us/sql/relational-databases/security/encryption/transparent-data-encryption-azure-sql) provides real-time encryption and decryption of data stored in the Azure SQL Database, using a key stored in Azure Key Vault.
* [SQL Vulnerability Assessment](https://docs.microsoft.com/azure/sql-database/sql-vulnerability-assessment) is an easy to configure tool that can discover, track, and remediate potential database vulnerabilities.
* [SQL Database Threat Detection](https://docs.microsoft.com/en-us/azure/sql-database/sql-database-threat-detection) is enabled.
* [SQL Database Auditing](https://docs.microsoft.com/en-us/azure/sql-database/sql-database-auditing) is enabled.
* [SQL Database metrics and diagnostics logging](https://docs.microsoft.com/en-us/azure/sql-database/sql-database-metrics-diag-logging) is enabled.
* [Server- and database-level firewall rules](https://docs.microsoft.com/en-us/azure/sql-database/sql-database-firewall-configure) have been tightened.
* [Always Encrypted columns](https://docs.microsoft.com/en-us/azure/sql-database/sql-database-always-encrypted-azure-key-vault) are used to protect patient first, middle, and last names. Additionally, the database column encryption also uses Azure AD to authenticate the application to Azure SQL Database.
* Access to SQL Database and SQL Server is configured according to the principle of least privilege.
* Only required IP addresses are allowed access through the SQL firewall.

## Storage accounts

* [Data in motion is transferred using TLS/SSL only](https://docs.microsoft.com/en-us/azure/storage/common/storage-require-secure-transfer?toc=%2Fazure%2Fstorage%2Fblobs%2Ftoc.json).
* Anonymous access is not allowed for containers.
* Alert rules are configured for tracking anonymous activity.
* HTTPS is required for accessing storage account resources.
* Authentication request data is logged and monitored.
* Data in Blob storage is encrypted at rest.

## Machine Learning

* [Logging is enabled](https://docs.microsoft.com/en-us/azure/machine-learning/studio/web-services-logging) for Machine Learning web services.

## Azure Security Center

## Data classification and FHIR

All sensitive data in the solution is tagged as electronic protected health information (ePHI), as follows:

* dataProfile => “ePHI”
* owner => *<Side Admin UPN>*
* environment => “Pilot”
* department => “Global Ecosystem”
* tier => API | Application | DataStore | Operations

The solution implements the following FHIR resources for data ingestion and output:

* [Condition](https://www.hl7.org/fhir/condition.html)
* [Encounter](https://www.hl7.org/fhir/encounter.html)
* [Observation](https://www.hl7.org/fhir/observation.html)
* [Patient](https://www.hl7.org/fhir/patient.html)

Customers can extend the solution to add support for additional resources as needed.

## Azure Key Vault

* Data stored in Key Vault includes:
  + Application insight key
  + Patient Data Storage Access key
  + Patient connection string
  + Patient data table name
  + Azure ML Web Service Endpoint
  + Azure ML Service API Key
* Advanced access policies are configured on a need basis
* Key Vault access policies are defined with minimum required permissions to keys and secrets
* All keys and secrets in Key Vault have expiration dates
* All keys in Key Vault are protected by HSM [Key Type = HSM Protected Key]
* All users/identities are granted minimum required permissions using Role Based Access Control (RBAC)
* Applications do not share a Key Vault unless they trust each other and they need access to the same secrets at runtime
* Diagnostics logs for Key Vault are enabled with a retention period of at least 365 days.
* Permitted cryptographic operations for keys are restricted to the ones actually required

## OMS

* Workspace is enabled for Security Center
* Workspace is enabled for Workload Monitoring
* Workload Monitoring is enabled for:
  + Identity and Access
  + Security and Audit
  + Azure SQL DB Analytics
  + [Azure WebApp Analytics](https://docs.microsoft.com/en-us/azure/log-analytics/log-analytics-azure-web-apps-analytics) Solution
  + Key Vault Analytics
  + Change Tracking
* [Application Insights Connector (Preview)](https://docs.microsoft.com/en-us/azure/log-analytics/log-analytics-app-insights-connector) is enabled
* [Activity log analytics](https://docs.microsoft.com/en-us/azure/log-analytics/log-analytics-activity) is enabled
* The [Office 365 solution for Operations Management Suite](https://docs.microsoft.com/en-us/azure/operations-management-suite/oms-solution-office-365?toc=%2Fazure%2Flog-analytics%2Ftoc.json) is enabled

[…]

# Compliance

[Link to Customer Responsibility Matrix xls]

# Deployment guide

The components for deploying the solution can be downloaded from the Azure Blueprint for Healthcare repository on Github. For integration with PowerShell it is necessary to have Git for Windows installed, which is available at <https://git-scm.com/download/win>.

## Download solution from remote repository

[instructions for downloading solution, when final]

**Note:** You must create a personal access token (preferred) or enable alternate authentication credentials in order to allow command line git access to your Visual Studio repository. Visit <https://globalecosystem.visualstudio.com/_details/security/tokens> to create a personal access token or <https://globalecosystem.visualstudio.com/_details/security/altcreds> to enable alternate authentication credentials.

## Deploy the solution

1. Once the repository has been cloned, change the working directory to **Deployment**:

cd .\CHSE%20Healthcare%20Blueprint\Deployment\

1. Run **deploy.ps1** with the **installModules** switch to install all necessary modules:

.\deploy.ps1 -installModules

1. Once the modules are installed, run **deploy.ps1** again to deploy the solution. For detailed usage instructions, see [deploy.ps1 man page].

The script will ask you to supply a value for the **globalAdminPassword** parameter; enter the password for the administrative account you are using for the **-globalAdminUsername** parameter. The script then deploys the solution, which may take some time.

### [deploy.ps1 man page]

**deploy.ps1**

Deploys or manages the Azure Solution for Healthcare.

**Example 1: Installing required modules**

.\deploy.ps1 -installModules

This command will validate or install any missing PowerShell modules that the solution requires.

**Example 2: Deploying the solution in a test environment**

.\deploy.ps1 -deploymentPrefix *<prefix>*

-tenantId *<tenant-id>*

-tenantDomain *<tenant-domain>*

-subscriptionId *<subscription-id>*

-globalAdminUsername *<username>*

-deploymentPassword *<password>*

This command deploys the solution and sets a single common password for all solution users, for testing purposes.

**Example 3: Uninstall the solution**

.\deploy.ps1 -deploymentPrefix *<deployment-prefix>*

-tenantId <*tenant-id*>

-subscriptionId <*subscription-id*>

-tenantDomain <*tenant-domain*>

-globalAdminUsername <*username*>

-clearDeployment

Uninstalls the solution, removing all resource groups, service principles, AD applications, and AD users.

**Parameters**

All parameters are optional, although at least one parameter must be present to successfully run the script.

-clearDeployment

Uninstalls the solution, removing all resource groups, service principles, AD applications, and AD users.

-deploymentPassword *<password>*

If this parameter is set, all of the passwords used within the solution will be set to the supplied value. The deploymentPassword parameter is intended for testing purposes, and should not be used in a production environment. If this parameter is not used, the script will generate and display 15-character strong passwords for every role.

-deploymentPrefix *<prefix>*

A string of 1 to 5 alphanumeric characters that will be used to create distinct resource group names. If you run the script multiple times, you must choose a different prefix each time to avoid conflicts with other resources.

-enableADDomainPasswordPolicy

Include this switch when deploying the solution to set the password policy to 60 days at the Domain level.

-enableMFA

Include this switch when deploying the solution to enable multi-factor authentication for solution users.

-installModules

Installs and updates all necessary PowerShell modules. Run the script with this switch before performing the main deployment to ensure that all necessary modules are present.

-globalAdminUsername *<username>*

Typically, this is the username of the user performing the deployment. Use a valid Azure Active Directory OrgID username (example: “alex@contosohealthcare.onmicrosoft.com”) rather than a Microsoft or corporate account name (example: “alex@contoso.com”).

-subscriptionId *<subscription-id>*

For the subscription ID, sign in to the [Subscriptions view in the Azure portal](https://portal.azure.com/#blade/Microsoft_Azure_Billing/SubscriptionsBlade). The subscription you are using should be listed in the table, along with its associated subscription ID, a GUID.

-tenantDomain *<tenant-domain>*

The tenant domain is the default directory name listed under your account name in the dashboard followed by “.onmicrosoft.com” (example: “contosohealthcare.onmicrosoft.com”).

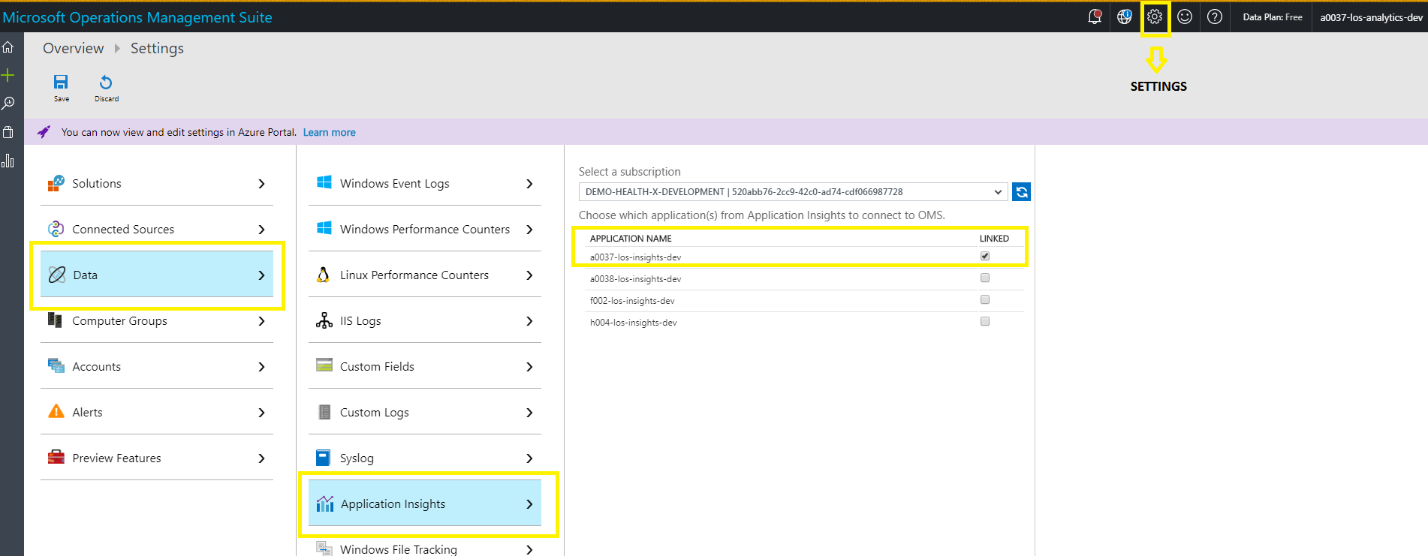
-tenantId *<tenant-id>*

To find the Azure tenant ID, click **Azure Active Directory** in the dashboard sidebar, and under **Manage**, click **Properties**. The tenant ID is a GUID in the box labeled **Directory ID**.

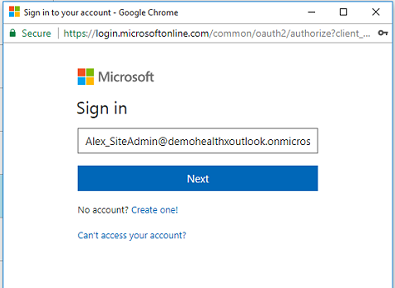
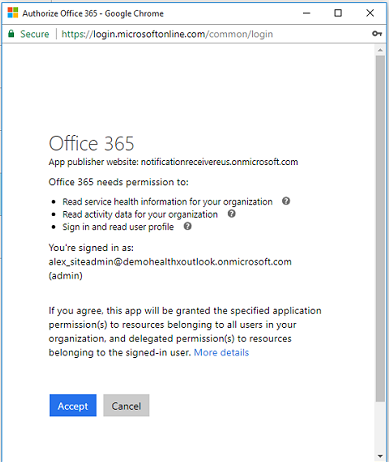
## Integrate Application Insights and Office 365 to Log Analytics

Use the [Microsoft Operations Management Suite](https://docs.microsoft.com/en-us/azure/operations-management-suite/operations-management-suite-overview) portal to integrate [Application Insights](https://azure.microsoft.com/en-us/services/application-insights/) and Office 365 into the [Log Analytics](https://azure.microsoft.com/en-us/services/log-analytics/) tool.

1. In the portal, click the Settings icon at the top of the page. Click **Data** on the left, and then click **Application Insights**.
2. Under **Select a subscription**, select a subscription that has Application Insights resources, and then under **Application Name**, select one or more applications.
3. Click **Save**.



1. To configure Office 365, click **Connected Sources** on the left, and then click **Office 365**.
2. Click **Connect Office 365**. When prompted, log in with the Alex\_SiteAdmin account and password. (If you supplied a value for the **deploymentPassword** parameter when deploying the solution, use that value.) You will be prompted to authorize Office 365. Click **Accept** to continue.

1. While the solution is preparing, the status will be reported as “Performing Assessment” on the Office 365 tile in the OMS portal. It may take several hours for the operation to complete.

## Grant permissions in Azure Active Directory

1. In the Azure portal, click **Azure Active Directory** in the sidebar.
2. Click **App registrations**.
3. Click *<deployment-prefix>* **Azure HIPAA LOS Sample**.
4. Click **Required permissions**.
5. Click **Grant Permissions** at top. You will be asked if you want to grant permissions for all accounts in the current directory. Click **Yes**.

## Uninstall solution and clean up deployment resources

To uninstall the solution, run the deployment script again with the following parameters:

.\deploy.ps1 -deploymentPrefix *<deployment-prefix>*

-tenantId <*tenant-id*>

-subscriptionId <*subscription-id*>

-tenantDomain <*tenant-domain*>

-globalAdminUsername <*username*>

-clearDeployment

The script will ask you to supply a value for the **globalAdminPassword** parameter; enter the password for the administrative account you are using for the **-globalAdminUsername** parameter. The script then removes all resource groups, service principles, AD applications, and AD users.

# Deploying and running the demo

[intro]

## Bulk data ingestion (Database Analyst role)

After deploying the Azure Blueprint for Healthcare, the first step in running the length-of-stay (LOS) demonstration project is for Danny, the database analyst, to input the sample historical patient data into the solution. This data is provided in a large .csv file in the CHSE Healthcare Blueprint\Deployment\trainingdata directory containing approximately 100,000 real, anonymized medical records.

To input the data, navigate to the **CHSE Healthcare Blueprint\Deployment\scripts\demoscripts\** directory in PowerShell and run the following script:

.\HealthcareDemo.ps1 -deploymentPrefix *<deployment-prefix>* -Operation Ingestion

where *<deployment-prefix>* is the prefix you selected when installing the solution.

To confirm that the data has been successfully ingested, log into the Azure portal using the Danny\_DBAnalyst account. Click **SQL databases** in the sidebar, and then click **patientdb**. Click **Data explorer (preview)** to open an interface for executing SQL queries. (If you are logged in to the Azure portal using any other account, you will have to enter the credentials for Danny\_DBAnalyst when prompted.) To check the number of rows in the PatientData table, run the following query:

SELECT COUNT (\*)

FROM [dbo].[PatientData]

It should return a result of 100,000 rows, representing several years’ worth of historical patient data. To check the integrity of the data, run the following query:

SELECT TOP 100 \*

FROM [dbo].[PatientData]

This displays the first 100 rows from the table. Note that the first, middle, and last names of each patient are encrypted at rest in the database.

## Patient admission (Care Line Manager role)

Chris, the care line manager, adds newly admitted patients to the database by uploading their information in FHIR format. These records are stored in ten individual .json files, one for each patient, in the CHSE Healthcare Blueprint\Deployment\scripts\demoscripts\admit directory.

To upload these records, navigate to the **CHSE Healthcare Blueprint\Deployment\scripts\demoscripts\** directory in PowerShell and run the following script:

.\HealthcareDemo.ps1 -deploymentPrefix *<deployment-prefix>* -Operation BulkPatientAdmission

A browser window will appear asking for credentials. Log in using the credentials for the Chris\_CareLineManager account. The PowerShell window will show the progress of the upload. To confirm the addition of the new records, log into the Azure dashboard using the Danny\_DBAnalyst account. In the SQL query window, run the following query:

SELECT COUNT (\*)

FROM [dbo].[PatientData]

It should report 100,010 records, 10 more than after bulk ingestion.

## Patient discharge (Care Line Manager role)

Chris, the care line manager, is also responsible for keeping track of patient discharges by uploading discharges in FHIR format. As with the admission records, the discharge records are stored in individual .json files, in the CHSE Healthcare Blueprint\Deployment\scripts\demoscripts\discharge directory.

To upload these records, navigate to the **CHSE Healthcare Blueprint\Deployment\scripts\demoscripts\** directory in PowerShell and run the following script:

.\HealthcareDemo.ps1 -deploymentPrefix *<deployment-prefix>* -Operation BulkPatientDischarge

A browser window will appear asking for credentials. Log in using the credentials for the Chris\_CareLineManager account. The PowerShell window will show the progress of the upload. To confirm the discharges, log into the Azure dashboard using the Danny\_DBAnalyst account. In the SQL query window, run the following query:

SELECT TOP 20 \*

FROM [dbo].[PatientData]

ORDER BY EId DESC

It should show that the most recently admitted patients have all been discharged.

## Machine learning (Data Scientist role)

Distinguish between bulk import for ML training and admissions/discharge later. Import format will depend on what their own ML is expecting

## Data visualization (Chief Medical Information Officer and Care Line Manager roles)

[refer to pbix received from Frank]

# FAQs

[Are we having these?]