Security Hardening Guide

Updated for Ignition 8.1

(800) 266-7798
www.inductiveautomation.com

Introduction

Welcome to Inductive Automation's Ignition Security Hardening Guide. Inductive Automation is committed to security and strives to make the product as secure as possible. This document is intended to provide recommendations on how to secure your Ignition installation.

Included in this document are guidelines specifically for the Ignition software, as well as general suggestions regarding the hardware and network where Ignition is installed. The steps provided are recommendations rather than requirements and should be considered in accordance with organizational policies and objectives.

Start with a Good Foundation

Ignition security should be considered within the context of its greater environment. "Defense in Depth" is a strategy that uses overlapping protective mechanisms supporting the ability of defenders to monitor and respond. For example, ISA/IEC 62443 provides a series of OT cybersecurity standards and the "Purdue Model" (ANSI/ISA 95) offers a layered reference model for segmentation.

It is an industry best practice to adhere to a formal cybersecurity framework to align security controls with organizational objectives. The Cybersecurity and Infrastructure Security Agency (CISA) under the Department of Homeland Security (DHS) provides great free cybersecurity resources.

Secure Baseline

The following ten steps offer best practice recommendations for securing Ignition and its environment. These steps cover securing access to the Gateway and devices, user management, application security, auditing, the environment, and keeping Ignition up to date. Security controls should align with organizational goals and policy.

Step 1: Secure Gateway Communication

Enforcing secure communication (HTTPS using an SSL/TLS certificate) is the first and most important step towards securing an Ignition Gateway.

Secure Communication

An Ignition Gateway can be configured to provide modern, end-to-end secure communication using Transport Level Security (TLS) technologies by enabling HTTPS. Secure communication protects all Gateway traffic to Perspective sessions, Vision clients, Ignition Designers, and Ignition web configuration from attackers and eavesdroppers through strong encryption and standard technologies. TLS also helps protect against a class of security vulnerabilities known as "session hijacking" where a threat actor may exploit a valid computer session to gain access to an Ignition system. Examples of "session hijacking" are: Man-in-the-middle attack, cross-site scripting (XSS), or session sniffing.

Terminology note: “Secure Sockets Layer”, (SSL) is the predecessor to the Transport Layer Security (TLS) protocol. SSL is deprecated as a technology. However, the term “SSL” is still widely used to refer to secure communication and “TLS”. For example, modern digital certificates supporting TLS are commonly referred to as “SSL certificates”.

Steps to Enforce Secure Communication

Full instructions to Enable Secure Communication are available online. The broad steps are:

1. Obtain certificate. Secure communication (HTTPS) in Ignition requires an SSL/TLS Certificate installed in the Gateway. It is highly recommended to obtain genuine certificates issued from a reputable Certificate Authority (CA).
2. Install certificate in Ignition Gateway.
3. Force secure redirect. After SSL/TLS is enabled, all Clients, Designers, and web browsers are redirected to the HTTPS port if they try to use the standard HTTP port. By default, the HTTPS port is 8043. Consider the standard https port (443) when customer facing.
4. Consider certificate renewal. First consider organizational standards from IT or cybersecurity. Many SSL/TLS certificates have a cumbersome renewal lifecycle. The renewal process can be simplified and automated using the ACME (Automatic Certificate Management Environment) protocol. ACME is an automated framework for obtaining and renewing certificates. Let’s Encrypt is a “free, automated, and open certificate authority (CA)” that can renew certificates using an ACME server and an Internet connection. See Let’s Encrypt Guide for Ignition.

Disabling Older Cipher Suites

When HTTPS is enabled, cipher suites provide essential information on how to communicate secure data between the Ignition Gateway and the user’s browser. The user’s browser will notify the Ignition server which cipher suites the browser supports, and the most secure cipher suite supported by both will be used for communication. As a security assumption, consider the gateway to be protected at the level of the weakest allowed cipher suite. A threat agent can execute a “downgrade attack” by only offering the chosen suite. Cipher suites can be excluded in the Gateway settings under **Config > Web Server > Excluded Cipher Suites**.

The suites currently considered strong are ever-changing as security researchers discover flaws and create new algorithms. One source for information on the currently accepted list of strong suites is maintained by SSL Labs. SSL Labs provides a free online test if your Ignition server is accessible from the Internet. Their current list of strong cipher suites can be found in their “SSL/TLS Deployment Best Practices” guide under 2.3 User Secure Cipher Suites. In general, allowing older cipher suites helps with compatibility while fewer and stronger cipher suites improve security.

Configuring Strong Headers

Https headers are set to secure default values as described under Gateway Security HTTP Headers and Configuration (KB article). Ignition offers the opportunity to change the values of these headers in the Gateway Configuration (Ignition.conf) file.

Consider enabling HTTP Strict Transport Security (HSTS). HSTS ensures that web browsers connect over https. It is disabled by default in Ignition to allow communication over http. **Enabling HSTS is**

recommended if Ignition is being used over standard http port 80 and https port 443. It is not recommended with default Ignition ports of 8088 and 8043.

1. Enable HSTS per this reference. In **Ignition.conf** file under "Java Additional Parameters" set:
    a. wrapper.java.additional.1=-Dignition.https.sts.maxAge=63072000
        i. Note: 63072000 seconds = 2 years.
    b. wrapper.java.additional.2=-Dignition.https.sts.includeSubDomains=false
    c. wrapper.java.additional.3=-Dignition.https.sts.preload=true
2. Setting "preload=true" means that the gateway url can be registered with the HTST preload list. This will prevent HTTP from ever being used at all and will force HTTPS from the browser itself. To use it, it does require a user to register their site with https://hstspreload.org/ after setting preload=true.

Step 2: Locking the Gateway

The gateway web site is organized into three sections, **Home**, **Status**, and **Configure**. By default, the **Configure** and **Status** sections are protected by the *"Authenticated/Roles/Administrator"* security level. Gateway sections and Designer access can be protected with role based access control.

Details for setting up the Gateway Security Settings.

Step 3: Device, OPC, and MQTT Security

Device connections have historically been made using native device communication protocols. Most PLC manufacturers created their own protocols for communication, and a variety are popular and in heavy use today. Recently, some devices have been released that have OPC UA and MQTT embedded directly in the devices as well. Each category is secured in a different way.

Direct connections from Ignition to OPC UA and MQTT devices are generally the most easily-secured connections, although any connection can be secured, given the right configuration and network security.

OPC UA Communication

OPC UA provides built-in security whether at the server level or embedded directly on a device. One of the ways that this can be done is to encrypt all communication. Different devices/servers support different encryption levels, but when setting up endpoints, be sure to choose the SignAndEncrypt security mode. This ensures all data sent over OPC UA will be encrypted.

Also, when configuring the Ignition OPC Server, trusting remote certificates is required for all secured inbound and outbound connections. Under **OPC UA** > **Security** trusted certificates can be imported and quarantined certificates can be marked as trusted. Some third-party OPC Servers may require additional steps such as manually adding the client certificate.

OPC UA connections also support user authentication. We recommend using a strong password and changing it periodically as defined by IT standards. Ignition's OPC UA server by default, when Ignition is installed, creates a default "opcua-module" user source which contains one user with default credentials. The default usersource allows the Gateway to initially connect as a UA client to its own UA server, but it is recommended that the authentication user source in the OPC UA Server Settings is modified in order to further lockdown the Gateway. After modifying the user source, the OPC UA

Client Connection Settings for the Ignition OPC UA Server loopback connection will need to be modified to match the modified credentials.

MQTT

MQTT includes built-in security features. It is recommended that data transferred between the Publisher and Broker (MQTT Transmission), as well as between the Broker and Subscriber (MQTT Engine), are configured with certificate-based TLS connections. In addition to this encryption, Username/Password Authentication is also supported and can be utilized to protect the data. MQTT also supports Access Control Lists (ACLs) which limit user access based on topic name space. These security measures should be implemented whether the broker is local or hosted in the cloud. For more information on securing MQTT communication, see the Cirrus Link MQTT secure communication instructions.

Native Device Communication

In addition to encryption between Ignition and OPC UA or MQTT devices/servers, communication between Ignition and other devices should also be protected. Since these devices often do not support encryption or certificates, a common practice is to keep them on a separate private OT network. Ignition can provide a layer of separation between the OT/private and the IT/public network to make tags available securely without exposing the devices behind the scenes. Other security options include placing Ignition and devices on a VLAN network with encryption enabled, setting up routing rules on the network or using an edge-of-network computer (such as Ignition Edge on an IPC) to act as a bridge between the device and the network.

We recommend consulting with a network security professional to help identify which option is best for you.

Step 4: Identity and Access Management

When securing an Application you must consider both authentication and authorization. Authentication determines who is logging in, whereas authorization determines their privileges. Ignition has the following tools to help satisfy almost any kind of authentication and authorization strategy required:

- User Identification and Authentication: User Sources and Identity Providers
- Authorization: Role-Based Access Control, Location-Based Access Control through Security Zones
- Security Levels: a hierarchical, inheritance-based access control model which builds off of Roles, Security Zones, and other attribute sources

User Identification and Authentication

Ignition manages users through Identity Providers (IdP). Ignition has a built-in IdP, but can also connect to third-party IdPs such as Okta, Duo, and ADFS via SAML or OpenID Connect.

Ignition Identity Provider

The Ignition IDP supports three main user sources, Internal Authentication, Database Authentication, and Active Directory Authentication.

Internal Authentication

From the Gateway Web Page, users can be managed directly within Ignition. These users are local to the Ignition Gateway where they are defined.

Database Authentication

The Database Authentication type uses an external database instead of storing data inside Ignition. Managing users is done via direct interaction with the database.

Active Directory Authentication

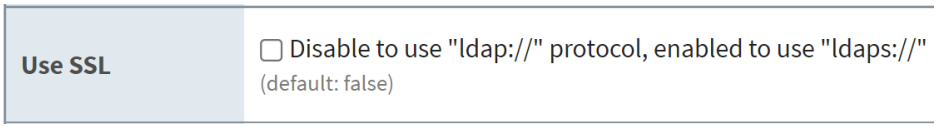
The Active Directory Authentication profile uses Microsoft's Active Directory over LDAP (Lightweight Directory Access Protocol) to store all the users, roles, and more that make up an Authentication profile. Active Directory Groups are used for Ignition's roles and user-role mappings.

While using an Active Directory User Source, administration of users and roles is through Active Directory itself, and not manageable within Ignition. Thus, adding new users to an Active Directory User Source or modifying pre-existing users requires the modifications be made from Active Directory, usually through an AD Administrator.

In Perspective, if a seamless experience is desired with no login prompt, consider using ADFS or another IdP instead of Ignition's internal IdP. See the Third-Party Identity Provider (Perspective) section below.

LDAP Protocol Security

The active directory User Source communicates with a Microsoft Active Directory server through the LDAP protocol. To prevent snooping on authentication, encryption should be implemented. In the advanced options for a new Active Directory User Source, Ignition has a setting to force LDAPS.



Badge Authentication

Another secure method is RFID authentication support for the Ignition Identity Provider allowing you to associate badges with users. Entering your credentials on a screen allows others to see your username and therefore weakens your security. With RFID enabled, users do not have to enter their username and password, and instead must have a physical badge in order to log in to the Session. If your organization makes use of RFID badges, it is recommended that the Badge Authentication Method is enabled and set to default and Badge Secret is also enabled. Badge Secret will require the user to type in their password after scanning their badge. This added layer of security is useful in numerous situations. For example, in the event that a user's badge is stolen and used by another individual, the added layer of a required password would keep the thief from accessing the Session, since a password will still be required for access.

More information on the Badge Authentication Method can be found on our website.

Third-Party Identity Provider

Utilizing an external service allows you to utilize a company's existing IdP infrastructure, leveraging existing investments in web application security. External services also allow features that Ignition doesn't support natively such as multi-factor options like push notifications or biometrics. They generally are easy to tie in if IT has standardized other applications on a particular SSO solution.

Third party IdPs are supported for all major authentication and authorization areas of Ignition, including Perspective Clients, Vision Clients, the Designer, and the Gateway Webpage.

General Authentication Suggestions

User Accounts

To ensure User Account integrity, a strong password policy should be defined including password length and complexity requirements. Establishing a password expiration schedule and quickly removing former user accounts are strongly recommended. When using Ignition's Internal Authentication, these settings can be found under the "Password Policy" section. Generic accounts should be avoided.

Group Access and Disabling Auto Login

Generic logins pose a security risk in any system. If Auto Login is enabled, any user that launches a project is granted basic access. To mitigate this risk, Auto Login should be disabled and each user should have their own unique credentials.

Authorization

Once a user is authenticated, authorization determines what they have access to. Authorization can be determined by a variety of conditions including roles and location. When creating new users, it is best practice to only create users for those who will need access to the environment. Every user credential should be unique to the particular user, avoiding shared or easy to guess usernames and passwords. Credentials should also be unique to the platform and not reused among other platforms. It is also best practice to exercise the Principle of Least Privilege (PoLP), which requires that a user must be able to access only the information and resources that are necessary for the user's purpose. This will benefit the stability and security of the Ignition environment as users are limited to the scope that they have clearance for and possibly have familiarity with. Examples include, preventing unapproved users from causing Gateway and/or project level modifications that can have adverse consequences to the Ignition environment or process.

Role-Based Security

Each user is granted privileges by assigning one or many roles. Roles are not default types but rather created custom during development. Roles can be defined inside Ignition or mapped to Active Directory groups or an IdP's attributes. The level of access granted by a given role is determined in "Step 5: Define Application Security".

Location-Based Security

A Security Zone is a list of Gateways, Computers, or IP addresses that are defined and grouped together. This group now becomes a zone which can have additional policies and restrictions placed on it. While Users and Roles restrict access to specific functions within the gateway, such as making

certain controls read-only for certain users and read/write for others, Security Zones provide this functionality to the Gateway Network, location-based access control in Vision, Perspective, Alarm Notifications and Status, and History Provider and Tag Access. Security Zones allow the application to restrict access based on where the client connects from in addition to a user's role-based privileges. This allows for greater control over the type of information that is passing over the network, improving security and helping to keep different areas of the business separate, while still allowing them to interconnect.

Using Security Zones

Sometimes, in addition to knowing who the user is, it is important to know where they are sending a command from. An operator may have permissions to turn on a machine from an HMI, but if they were to log in to a project on a different Gateway in the network that had remote access to those Tags, it might not be a good idea to let the operator write to those Tags from a remote location where they can't see if the physical machine is clear to run.

This is where Security Zones come in. Security Zones themselves don't define the security, they instead define an area of the Gateway Network, breaking up Gateways and network locations into manageable zones that can then have a security policy set on them. Once there are zones defined, a security policy can be assigned to each zone, and a priority of zones can be set in the event that more than one zone applies in a given situation.

Information on how to set up Security Zones can be found on our website.

Security Levels

Security Levels are a user-defined hierarchy used to define roles to manage access permissions. This authorization system can be used to map roles managed in an Identity Provider (IdP) to Ignition roles. Creating security levels can be used to restrict certain people from being able to access specific functions within the gateway, such as access to certain Perspective sessions, visibility of components in a view, or read/write permissions. This will improve security and allow better management of the information and level of control specific users are granted on the network.

Using Security Levels

When working with the Gateway, it is important that all users are able to get the information or control they need in order for operations to run smoothly. An operator may need to be able to read and write to Tags, and view the status of each plant. However, a manager may need to view the status of the plant and read the Tag values, but should not be able to write to the Tag values. Making sure that each user has the correct permissions is vital to the security of the operation.

With security levels, roles can be defined in order to allow certain users more or less control of the Gateway in flexible ways allowing granular access to specific security zones, projects, and other attributes. These roles can also be used in the Ignition Designer in order to limit a role's viewing access and/or read/write capabilities.

Security Levels are defined in the Gateway and arranged in an inherited tree structure. Each child (nested) level of the tree inherits the security of its parent levels. Consider a powerful permissions model where all Employees can access a screen in read-only mode, but only Plant-Floor Operator

Employees can read / write. Since Plant Floor Operators are Employees, they can do everything all other Employees can do plus the specific things that only Plant Floor Operator Employees can do.

Tip: Complex Security Levels Rules can be created using Ignition's expression language, where a Security Level can be derived from the Authentication / Authorization context. This context can take into account who you are (logged in user identity / profile attributes), where you are (security zones assigned to you), what roles you have, the current time of day (in case users can only log into a system between the hours of 8am-5pm PT, for example), tag state representing the physical system from PLCs, or many other complex factors.

Information on how to set up Security Levels can be found online.

Step 5: Define Application Security

Ignition is a software platform for creating custom applications to suit your needs. These applications could be for HMI (Human Machine Interface), SCADA (Supervisory Control And Data Acquisition), Database Front End and more. Each of the applications require customizable security. Ignition allows for security to be defined at any level from clients and projects down to individual tags.

Vision Client Security

In Clients, security settings can be applied to individual windows or components. While users with different roles may view the same project from the client, the functionality and accessibility can change based on their assigned roles. Generally, higher level access provides full functionality to all contents of a project, and lower level access is restricted to generalized read-only privileges. However, client security settings are flexible enough to accommodate any security requirements.

Information on how to set-up Client Security can be found on our website.

Perspective Session Security

In Sessions, security is managed through Identity Providers (IdP). Identity providers offer a way for users to log into Ignition using credentials that are stored outside of Ignition. Once the identity provider is set up in the Gateway, a security level can be assigned to a Perspective Session, which will grant only the users with the specified security level access to the Session. Generally, the higher level access provides full functionality to all components of the project and lower level sets more restrictions, such as read/write privileges.

Information on how to set up Perspective Session Security can be found on our website.

Perspective Views Security

Added security to a Perspective View allows more granularity of the security in a Perspective Session. An operator, for example, may need to view a Perspective Session but shouldn't be allowed to access a user management View. The operator can be granted access to the Session, but the user management View can be restricted to a higher level role. Adding security levels to both the Perspective Session and Views allows only the roles with proper privileges to be allowed to view or edit content and thus improving the security of the Perspective Session and control of information on a project.

Information on how to set up Perspective Views Security can be found on our website.

Component Scripting Security

Both Vision and Perspective contain role-based component scripting security to ensure that non-privileged users are not allowed to run scripts that can be potentially harmful to operations or manipulate information that a user does not have clearance for.

More information on Vision and Perspective component scripting on security can be found on our website.

Designer Security

When several users are all working on the same project, managing changes to the project can become cumbersome. By default, all users with Designer access can modify, delete, save, and publish all resources available in the Designer. In some situations, it is desirable to limit what each user can do in the Designer. Ignition has several built-in Designer restriction methods to help in these scenarios.

See project security in the Designer documentation for information about protecting Ignition resources.

Tag Security

Tag security is often the best way to configure security for data access. By defining security on a tag, you are applying those rules for that tag across all windows and components that use the specified tag in the project. This is opposed to configuring security on each component that controls the tag.

If a user opens a window that has components that are bound to a tag that the user doesn't have clearance to read or write to, the component will get a forbidden overlay.

You can add read/write security to individual tags through the Designer. Custom Access Rights must be set to set permissions based on Security Levels.

Named Queries

One of Ignition's key features is the ability to easily log, edit and retrieve data from SQL databases. By default, all database interaction is limited to defined queries on the Ignition Gateway, which may be called from clients based on the credentials of the user. These queries can be parameterized to allow for dynamic database interaction while ensuring only relevant data is accessible. It is recommended to only use parameters for individual variables rather than allowing longer SQL chunks to prevent SQL injection.

This feature can be turned off to allow any SQL query to be run directly from an open client. While this can be powerful for adding flexibility to the platform, it also leaves the data potentially exposed. If client-authored queries are enabled, be sure to use SSL and not use auto-login or any shared accounts.

Access to these named queries can be limited using the normal Ignition permission model including roles and security zones.

Step 6: Set up Audit Logging

Audit Profiles allow Ignition to record details about specific events that occurred. Audit Profiles are simple to set up and immediately start to record events. By default, only tag writes, SQL UPDATE, SQL INSERT, and SQL DELETE statements are recorded. This allows you to keep track of which user wrote to which tag, or modified which table. Furthermore, a time-stamp is recorded, so you can easily track the changes, outline, and order of events.

Once some changes have been made to a tag or a database table, Ignition will begin recording.

AUDIT_EVENTS_ID	EVENT_TIMESTAMP	ACTOR	ACTOR_HOST	ACTION	ACTION_TARGET	ACTION_VALUE
1	2016-07-25 17:50:09	admin	IU-WorkStation	tag write	B Tags/B3:1	1.0
2	2016-07-25 17:50:51	admin	IU-WorkStation	tag write	B Tags/B3:1	100.0
3	2016-07-25 17:50:53	admin	IU-WorkStation	tag write	B Tags/B3:1	2.0
4	2016-07-25 17:50:56	admin	IU-WorkStation	tag write	B Tags/B3:1	8.0
5	2016-07-25 17:51:20	admin	IU-WorkStation	query	update audit_events set acto...	4
6	2016-07-25 17:51:51	admin	IU-WorkStation	query	UPDATE audit_events SET `A...	1

More Information regarding Audit Profiles can be found on our website.

Step 7: Protect the Database

Different databases offer different authentication options. We recommend not using a database owner account such as **root** or **sa**. A separate user account with limited privileges should be created for the database connection with the Ignition Gateway.

Most modern databases also support SSL encryption of the connection between Ignition and the database. If the database is running on a different server, SSL can be enabled by following information available for your database's JDBC driver and internal security settings. Refer to the documentation for your database for more information on enabling SSL JDBC connections from Ignition.

Step 8: Locking down the Operating System (OS)

Ignition is a *distributed application development environment* that provides a powerful tool set to interface with physical systems and business applications. In order to properly secure Ignition it is important to understand how Ignition fits within the operating environment.

Ignition Privileged Users

By design, any users with direct or indirect access to modify projects (e.g. Designer, Gateway roles) are able to write Python applications (a.k.a. *programs*) that will be executed by the Ignition Gateway with all privileges granted to the Ignition process.

Ignition Process - System Account and Persistent Connections

Ignition is designed to run 24/7, which includes a persistent operating system process and trusted communication connections. This includes access to devices (e.g. PLC, OPC UA) and databases.

The Ignition Process is executed by a user-specified service account on the Gateway. This means that Ignition is theoretically capable of all operating system actions of that account including privileges on the local host and larger network environment.

The implication is that an Ignition designer (person) is implicitly trusted with (authorized) the sum of all privileges of the Ignition Gateway process, regardless of individual credentials. In order to protect the larger environment, it is important to apply the *Principle of Least Privilege* outside of Ignition. Consider the following tips.

- Do not use a “root” or “domain admin” account as the Ignition service account.
- Minimize privileges of the Ignition computer and service account on the larger network.
- Minimize Ignition permissions on external databases and business systems.
    - API access is often safer than direct access.
- Segment network areas with “zones and conduits” or “accreditation boundaries”.

Finally, many customer implementations use multiple Ignition Gateways. Each gateway has its own Ignition process with associated credentials.

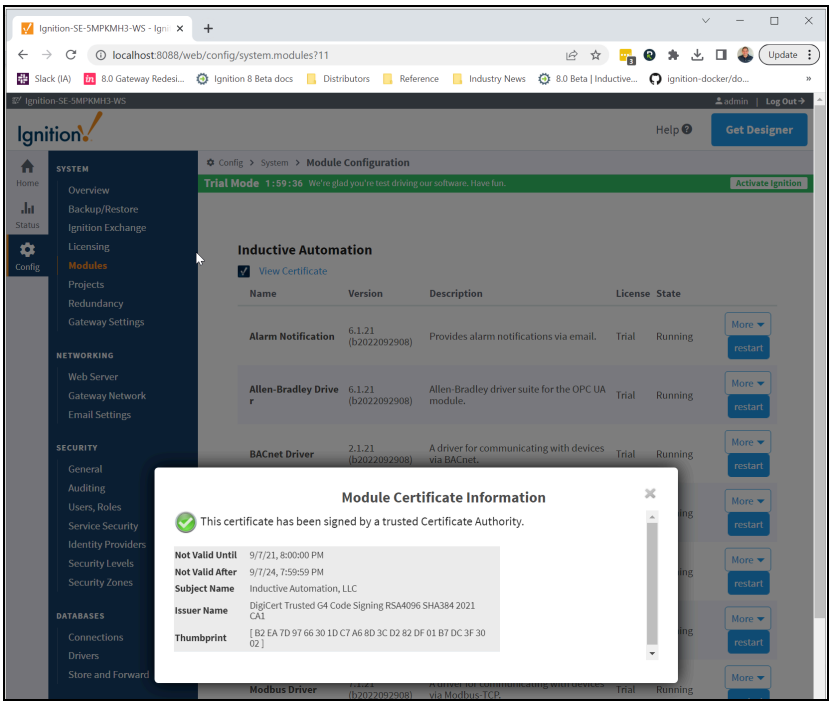
Verify that Ignition Software is Genuine

The Online User Manual provides Inductive Automation certificate information and instructions for verifying that Ignition software is genuine.

Verifying Trusted Code (Modules)

Ignition Modules will not run under normal configuration with self signed, expired or untrusted certificates. Verify the authenticity of Ignition modules in the Gateway web page.

Gateway Configuration Page -> Modules shows modules grouped by certificate (*View Certificate*).



Remove Unnecessary Programs

Each program and dependency library introduce potential entry points for an attacker. Removing unnecessary software limits vulnerabilities. Programs should be run using the minimum credentials required avoiding privileged users when possible. A strong approach to enforcing allowed software is called “application whitelisting”, where only approved applications are allowed to run and all others are disabled. Removing unnecessary programs is always a best practice, even with other security controls in place.

Runtime (Java) Security

Ignition is bundled with a purpose-built Java runtime that is kept up-to-date by staying on a current version of Ignition. Unless required by other applications on an Ignition gateway, it is recommended that externally installed runtimes such as Java and the .NET Frameworks be uninstalled to protect from known and future vulnerabilities. If needed, keep these runtimes up to date.

Patches and Service packs

Keep up-to-date on OS patches and Service Packs. Align with IT and OT policies and procedures.

Remote Services

On Windows, Remote Registry and Windows Remote Management should be disabled.

On Linux and Mac OS, disable root for everything but the ‘physical’ console.

Step 9: Hardening the Environment

Firewalls and ports

Firewalls should be in place to restrict network traffic. Best practice is to close all ports by default and open those that are necessary. Ignition default Gateway ports are listed in the online user manual.

Redundant Servers

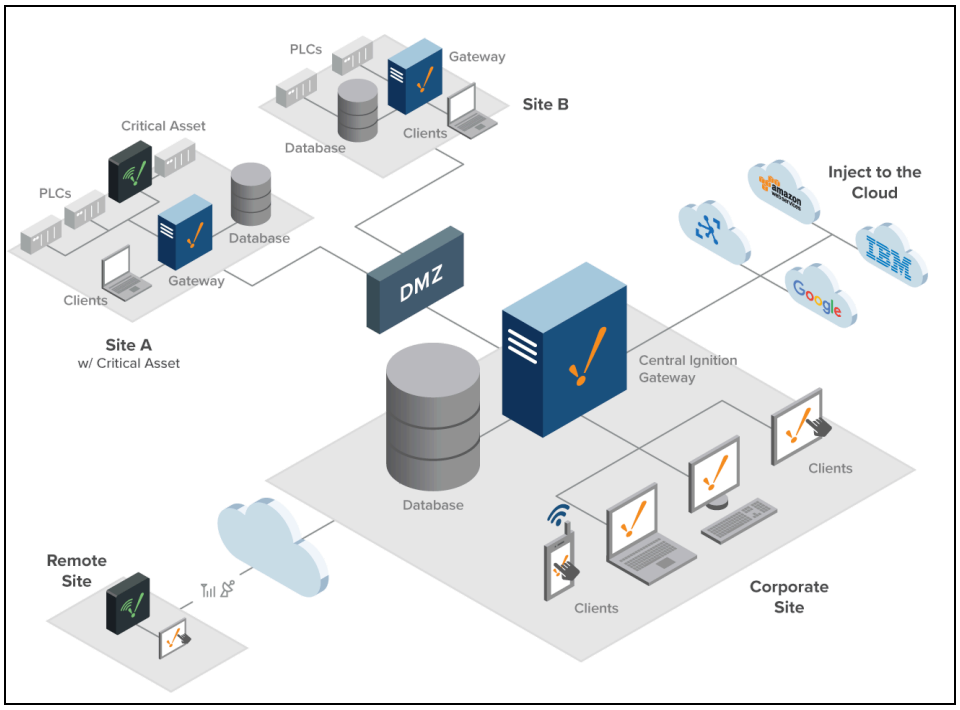
Firewalls must be set up on any server doing redundancy in order to protect the redundancy system from external attacks. For redundancy support, the firewall on the main server should be configured to accept connections on the Gateway Network port from the back-up server IP address. It is recommended to configure that firewall to reject Gateway Network port incoming connections from any other source unless Gateway Network traffic is expected from other Gateways.

DMZ Architecture

A DMZ Architecture (referred to as a “demilitarized zone”) contains a subnetwork that accommodates the organization’s exposed, outward connecting services. In general, it acts as a point of contact between the organization’s internal network and untrusted networks, such as the internet.

The goal of this architecture is to add an extra layer of security to the local area’s network, allowing the local network to access what is exposed in the DMZ and keeping the rest of the network protected behind a firewall.

We recommend consulting with a network security professional to help identify the best network options for your organization, and whether a DMZ is the right choice for your network topology.



Step 10: Stay Informed and Keep Ignition Up-to-date

Inductive Automation recognizes that software security requires constant effort and maintenance. It is important to keep Ignition up to date to protect from known vulnerabilities.

1. Subscribe to Trust Center Updates on the Inductive Automation Security Portal
2. Ignition is released on a 5 week Release Train. Keep Ignition up to date.
3. Read release notes for new versions of Ignition
4. Have an upgrade plan and follow a checklist
5. Keep up with informational resources
    a. Inductive Automation forum and tech advisories
    b. Cybersecurity & Infrastructure Security Agency (CISA) *United States Computer Emergency Readiness Team* (US-CERT) site for advisories and recommendations.

[DRAFT - Not to be included on the web page or PDF download until complete]

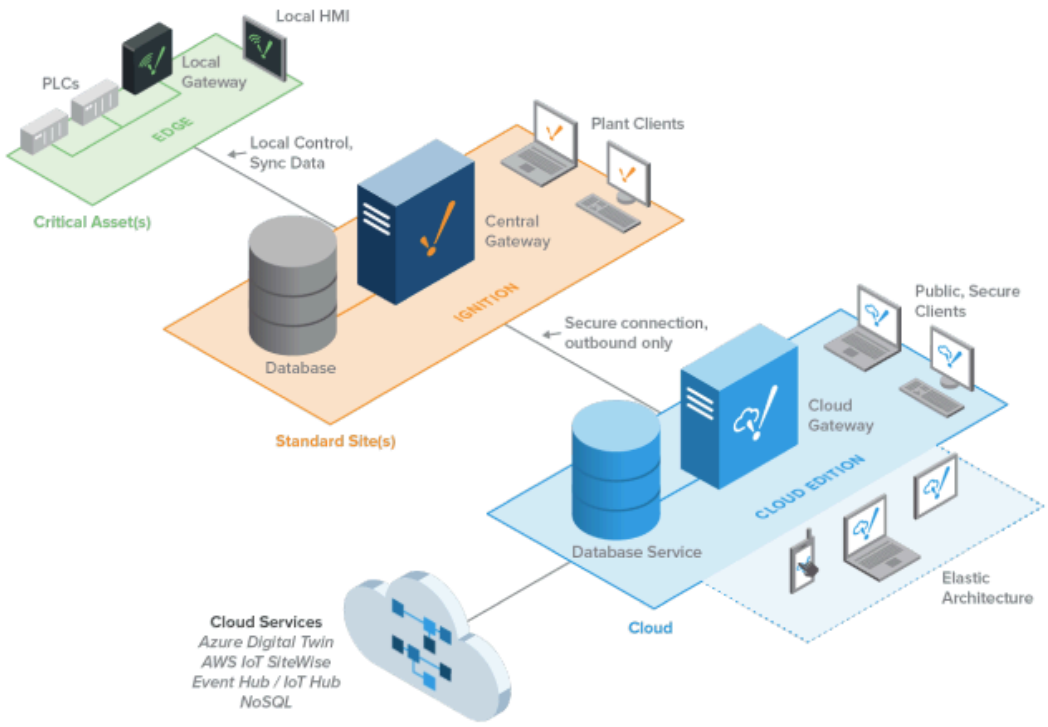
Cybersecurity Best Practices

Network Segmentation

Appendix A - Cloud Edition

Ignition Cloud Edition Architecture / What is Cloud Edition?

Ignition Cloud Edition is a deployment option for running Ignition in the Asset Owner's cloud account that is installed and billed through their Cloud Service Provider (CSP) Marketplace*. When running Ignition in the cloud there are some additional security recommendations, listed below.



*Using Ignition Cloud Edition does not grant Inductive Automation access to data, systems, or services within an asset owner's cloud infrastructure.

Shared Responsibility Model

When running Ignition Cloud Edition, managing security is a shared responsibility between the Cloud Service Provider (CSP), Inductive Automation, and the asset owner.

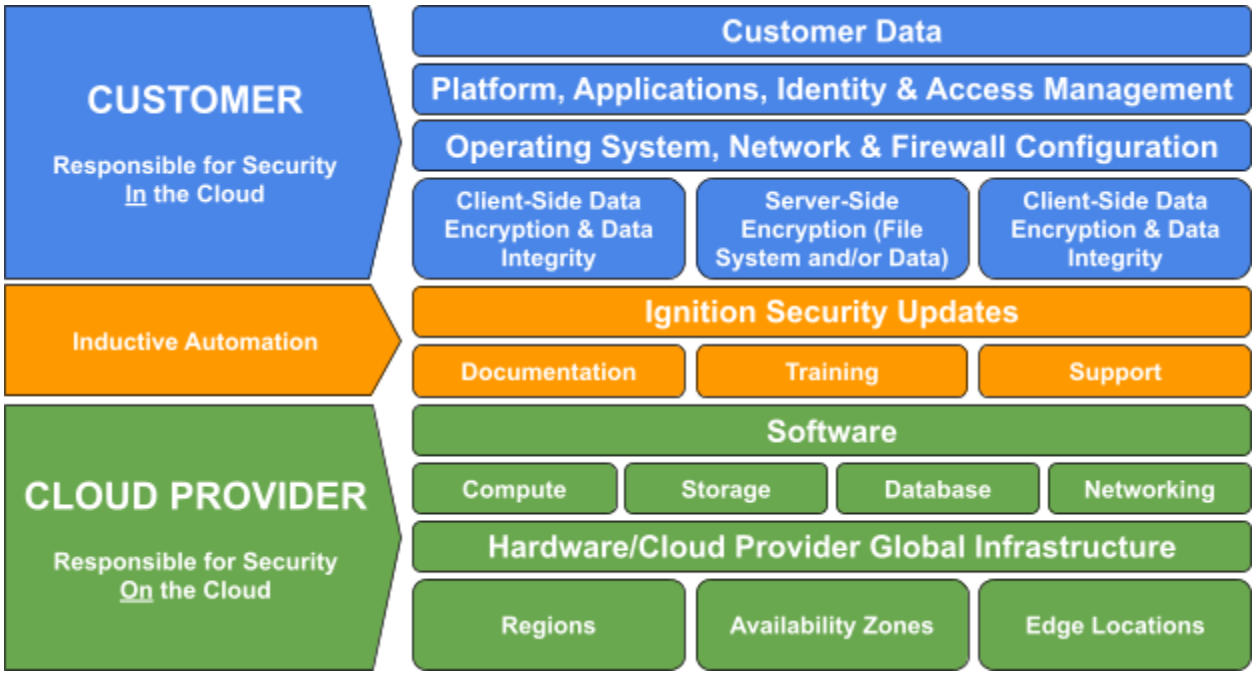
The CSP is responsible for security of the cloud. This means that the CSP protects and secures the infrastructure that runs the services offered by the CSP.

Customers are responsible for security in the cloud. When using any service provided by the CSP, you're responsible for properly configuring the service and your applications, including Ignition, in addition to ensuring that your data is secure.

Inductive Automation is responsible for providing security updates via Ignition software updates. Inductive Automation has a process in place to make sure that customers get the latest security updates through various means, such as nightlies, critical updates, and the release train.

More information regarding the AWS Shared Responsibility Model can be found on the AWS website.

More information regarding the Azure Shared Responsibility Model can be found on the Azure website.



Well-Architected Framework (Security Pillar)

Cloud Service Providers provide a Well-Architected Framework, which describes key concepts, design principles, and architectural best practices for designing and running workloads in the cloud. The Security Pillar of the Well-Architected Framework focuses on protecting information and systems.

Key topics include confidentiality and integrity of data, managing user permissions, and establishing controls to detect security events.

By following the guidance provided by these cloud pillars, you will build a good foundation for security for the Ignition platform.

More information regarding the Security Pillar of the AWS Well-Architected Framework can be found on the AWS website.

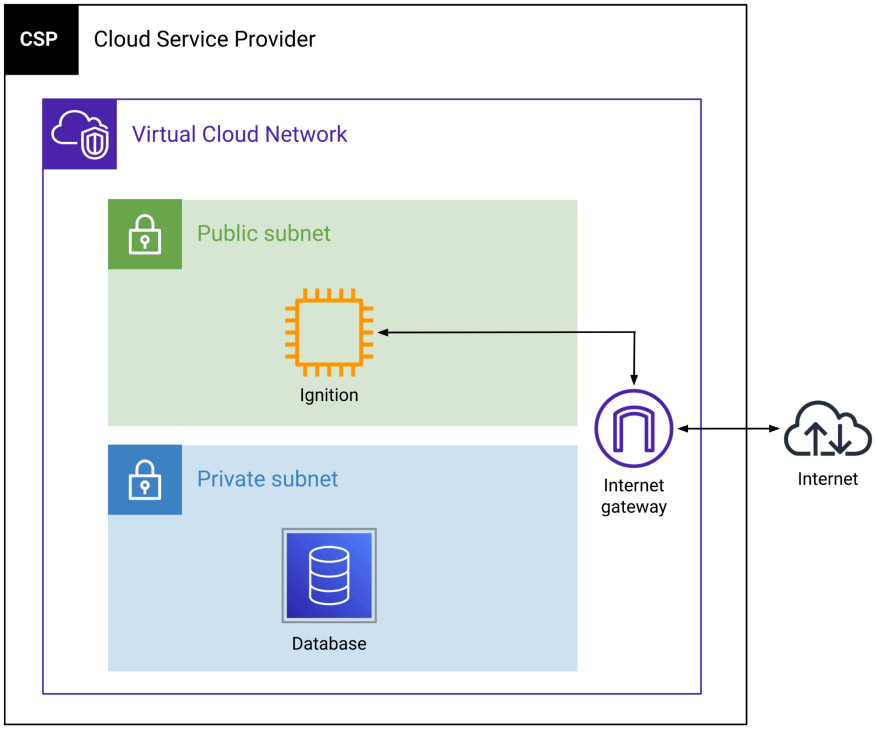
More information regarding the Security Pillar of the Azure Well-Architected Framework can be found on the Azure website.

The following sections highlight Ignition-specific recommendations for increased security within the cloud environment.

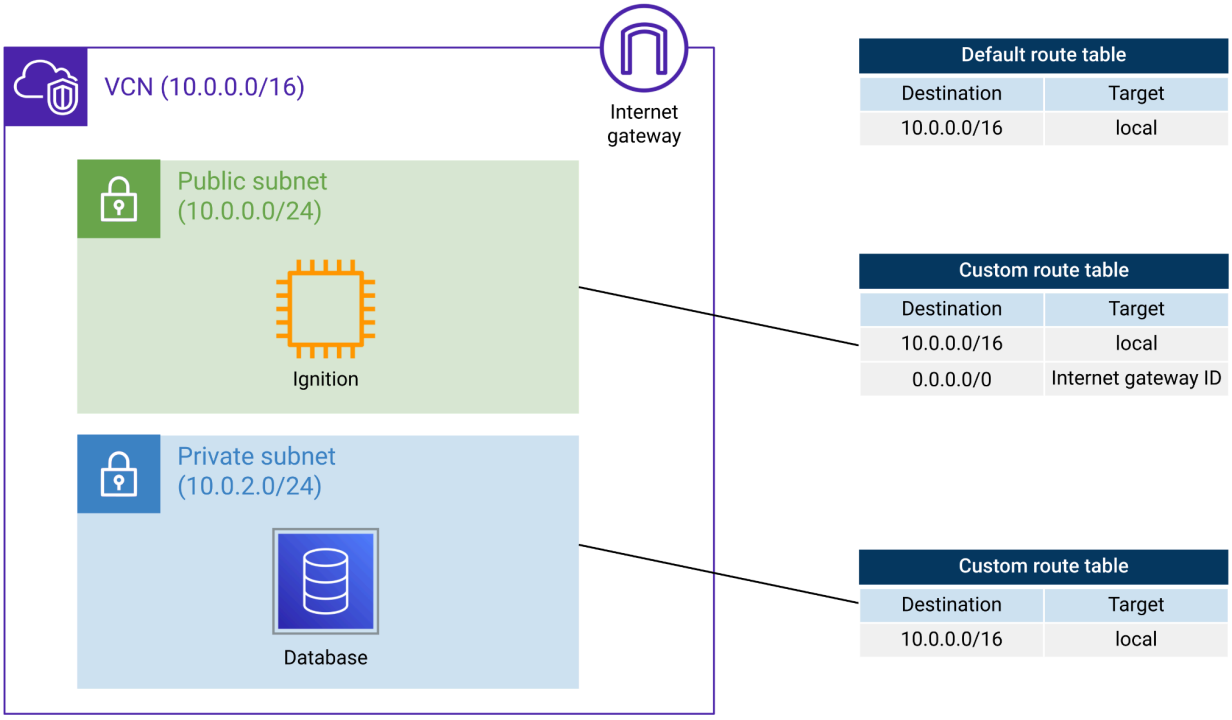
Virtual Cloud Network

Ignition Cloud Edition runs in a virtual compute instance within a virtual cloud network. A virtual cloud network (VCN) is an isolated network that you create in the cloud, similar to a traditional network in a data center.

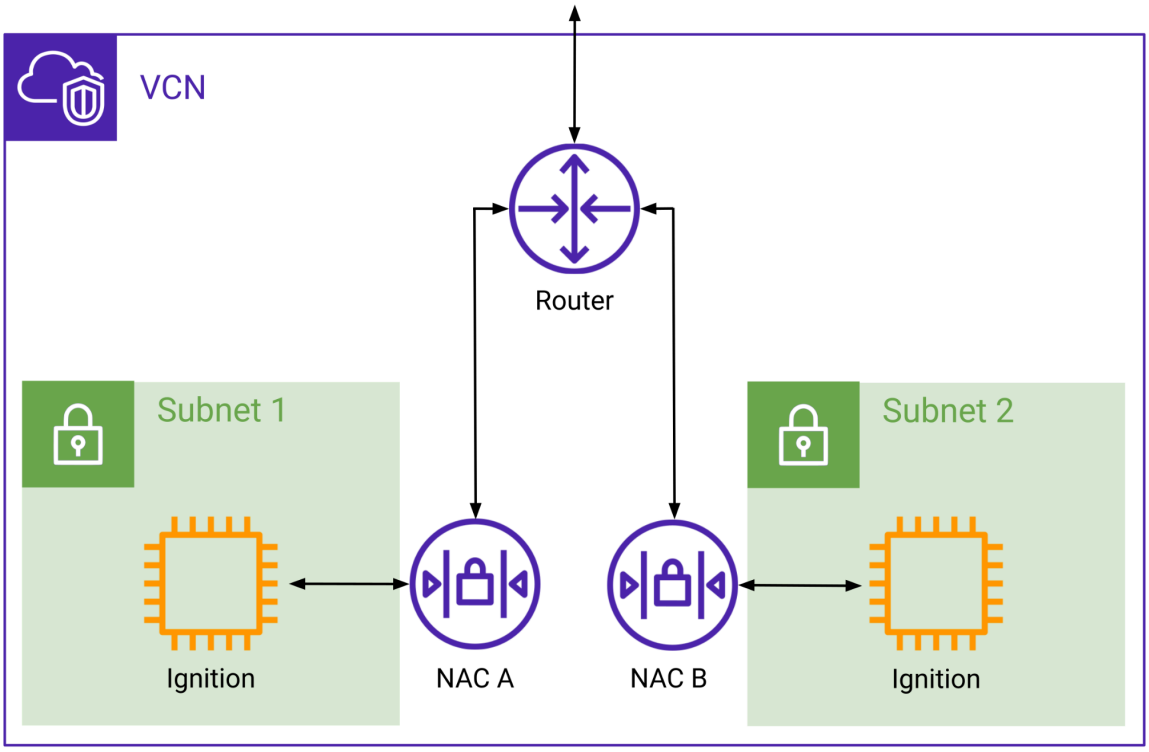
After creating a VCN, you must create subnets inside the network. Use a public subnet for resources that must be connected to the internet and a private subnet for resources that won't be connected to the internet.



When creating a VCN, the CSP creates a default route table for the VCN that is used to determine where network traffic is directed. The default configuration of this route table is to allow traffic between all subnets in the local network. Providing different routes on a per-subnet basis for traffic to access resources outside of the VCN can be accomplished by associating a subnet with a custom route table.



Controlling what kind of traffic is allowed to enter or leave a subnet can be accomplished by creating network access controls (NAC). You can configure NAC by setting up rules that define what you want to filter.



For example, to restrict your network to allow HTTPS traffic and Remote Desktop Protocol (RDP) traffic to your Ignition servers (using the default SSL port to access the Ignition Gateway), the following NAC could be used:

Inbound					
Rule #	Source IP	Protocol	Port	Allow or Deny	Description
100	All IPv4 traffic	TCP	8043	ALLOW	Allows inbound HTTPS from anywhere
130	192.0.2.0/24	TCP	3389	ALLOW	Allows inbound RDP traffic to the Ignition servers from your home network's public IP address range (over the internet gateway)
*	All IPv4 traffic	All	All	DENY	Denies all inbound traffic not already handled by a preceding rule

Outbound					
Rule #	Source IP	Protocol	Port	Allow or Deny	Description
120	0.0.0.0/0	TCP	1025-65535	ALLOW	Allows outbound responses to clients on the internet (serving users visiting the Ignition servers in the subnet)
*	0.0.0.0/0	All	All	DENY	Denies all outbound traffic not already handled by a preceding rule.

In the above NAC, inbound traffic on port 8043 and outbound traffic in the range 1025-65535 are allowed for SSL access to Ignition because SSL connections are initiated on port 8043 and will respond to an ephemeral port.

The next layer of security is for the virtual compute instances. Here, you can create a virtual firewall to control inbound and outbound traffic by defining a set of rules that determine the type of traffic that is allowed to reach or leave the instance.

[do we need more info about this? ^]

To find which ports should be open for your specific environment, please refer to our list of ports above in Section X

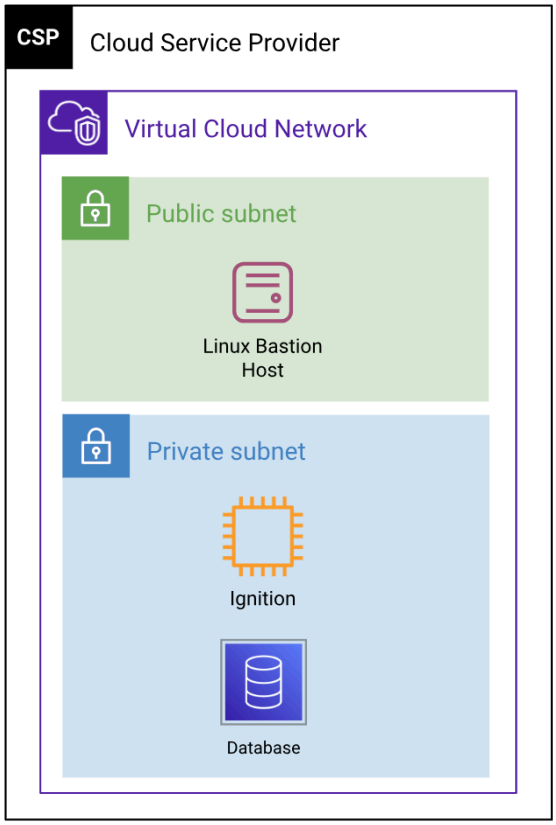
More information regarding configuring subnets, route tables, firewalls at the subnet level (Access Control Lists), and firewalls at the instance level (Security Groups) in AWS can be found on the AWS website.

More information regarding configuring subnets, route tables, and firewalls at the subnet level and instance level (Network Security Groups) in Azure can be found on the Azure website.

Device Connectivity from the Cloud

Cloud edition will not include any device drivers or OPC-UA connections (or Legacy OPC-COM) to prevent users from directly connecting to (and controlling) local devices from the cloud. The intention is to have Cloud edition be a part of a larger architecture, where standard Ignition or Ignition Edge installations **on premises** perform local data acquisition, and pass the data to Ignition Cloud.

Remote Access to Cloud Providers - Linux Bastion Hosts and Client VPNs



To access the resources on your cloud provider from your work network (or even a public network), there are several options for a secure connection. One option is to set up a Linux Bastion Host (also known as a jump box) which is located in a DMZ on your cloud provider. A Linux Bastion Host is hardened against cyberattack, runs only the minimum necessary applications for access, and has logging and monitoring capability to identify any security concerns.

Another option is to use a Client VPN through your cloud provider, which allows remote users to connect to your cloud resources.

With a Bastion Host, it requires another server to be spun up, and requires management like any other server. Additionally, that server is a single point of access (and failure) for your secure remote access. A Client VPN is more elastic, and will not be a bottleneck for a large number of users like a single Bastion Host would be.

For Linux Bastion Hosts on AWS, see this article:

https://aws.amazon.com/solutions/implementations/linux-bastion/

For Client VPN on AWS, see here:

https://aws.amazon.com/vpn/client-vpn/

For Azure Bastion, see here:

https://azure.microsoft.com/en-us/products/azure-bastion

For Azure Client VPN, see here:

https://learn.microsoft.com/en-us/azure/vpn-gateway/openvpn-azure-ad-client

Load balancer redirects (Frontend protection). Config page & Designer.

HTTP vs HTTPS

HTTP (cleartext) should never be used for cloud hosted applications. HTTPS is recommended using TLS 1.2 or newer with genuine certificates. See “Step 1” of this guide for recommendations. AWS and Amazon offer mechanisms for HTTP to HTTPS redirection.

For AWS, you can follow this guide:

https://aws.amazon.com/premiumsupport/knowledge-center/elb-redirect-http-to-https-using-alb/

For Microsoft Azure, you can follow this guide:

https://learn.microsoft.com/en-us/azure/frontdoor/front-door-how-to-redirect-https

Add in Cloud Service Terminology and add this as a downloadable pdf on the SHG page

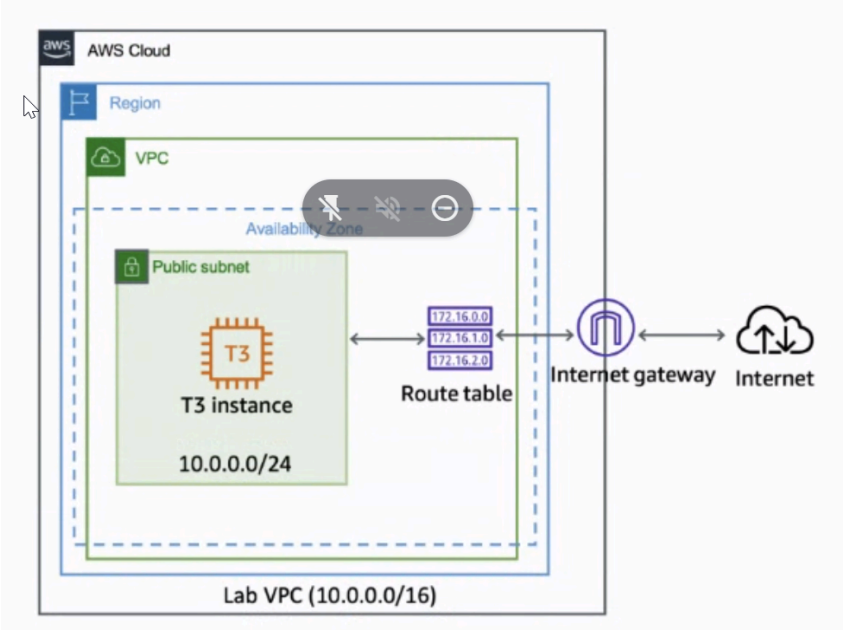
Appendix B Amazon Web Services AWS

IAM users/roles

Protect the root user
https://docs.aws.amazon.com/accounts/latest/reference/best-practices-root-user.html

Application Load Balancers

Single AWS Ignition Gateway with Load Balancer Example Step by step



Appendix C Microsoft Azure

Notes:

- Specific language and terminology for resources
- Azure specific Best Practice links
-