Ironstar Security

This document provides detailed information about how Ironstar systems are provisioned and maintained, and the security, data integrity, and reliability procedures and controls employed by Ironstar to help protect customer environments as well as Ironstar's internal operations.

Our procedures and platform are constantly evolving and being improved to keep pace with new threats and industry best practices. The information in this document is subject to change without notice. The latest copy of this document is available at https://ironstar.io/ironstar-security.pdf

At no time should this document be considered to establish a binding agreement between Ironstar and any third party. Ironstar customers receive security commitments and service warranties (including commitments to the controls in this document) solely through a Subscription Agreement, an Enterprise Services Agreement, or other agreement entered into between Ironstar and you.

Contents

Ironstar Subscription Types

Ironstar operates a homogenous infrastructure services platform (more commonly referred to as “the platform”) which is used to operate many different “subscription types”. These subscription types benefit from universal security, reliability, and performance controls while differing in their support, service level, and higher-end capabilities.

Each Subscription is made up of one or more Environments, including just one Production Environment. Within a Subscription Type, different Environments will have different capabilities, service level commitments, and support response times.

Ironstar Free
An invite-only platform designed for registered non-profit organisations, Ironstar Free operates on the same platform as all other types and as such benefits from many of the controls discussed in this document. However, Ironstar Free provides no contractual guarantees around performance, security, or reliability and is considered a “best effort” platform.

Ironstar Legacy
Some existing Ironstar customers have environments that are orchestrated in part by legacy systems. These systems provide much of the same controls as the newer platform types, but may differ in subtle ways. Where Ironstar Legacy environments differ from the standard platform, those differences will be highlighted in this document.

Ironstar Core
Suitable for small sites with no complex security or performance requirements, the Ironstar Core platform provides industry-leading levels of security and performance at minimal cost. Ironstar Core and Ironstar Advanced are functionally very similar, but differ in terms of the support and service level commitments that Ironstar makes.

Ironstar Advanced
Medium-size organisations with important production workloads can leverage the Ironstar Advanced platform to receive guaranteed commitments from Ironstar as to the availability, security, and performance of their environments.

Ironstar Enterprise
Ideal for large organisations hosting mission-critical sites, Ironstar Enterprise is one of the most capable and robust PHP managed hosting platforms in the world. This platform type is suitable for teams and organisations with very specific security requirements.

Ironstar Enterprise - Dedicated Cluster
A variant of the Ironstar Enterprise platform, a Dedicated Cluster enables customers to provision their own cluster in their own AWS accounts, which Ironstar then maintains and deploys a licensed copy of our controller software to. Customers receive infrastructure invoices directly from AWS and Ironstar provides a fixed yearly management fee to operate, support, and maintain the cluster(s).

Comparison of Ironstar Subscription Types

	Free	Core	Advanced	Enterprise
Isolated Environments	Yes	Yes	Yes	Yes
Web Servers	1	1	1	2+
Database Servers	1	1	1	2
Named SSH Users	Yes	Yes	Yes	Yes
High Availability	Yes	Yes	Yes	-
Fault Tolerance	-	-	-	Yes
Load Balancer	Shared	Shared	Shared	Dedicated
Content Delivery Network	No	No	No	Option
Disaster Recovery	No	No	No	Yes
Log Retention	7 Days	7 Days	30 Days	90+ Days
Backup Schedule*	On-Demand Only	Daily	Daily	Up to Hourly
Backup Retention*	7 Days	7 Days (Production)	30 Days (Production)	6 Months (Production)
Drupal Multisite	No	No	No	Option
Automated SSL Certificate	Yes	Yes	Yes	Yes
Custom SSL Certificates	No	Yes	Yes	Yes
Security Compliance Auditing Assistance	No	No	No	Option
Support Types	Community	Ticket	Ticket/Phone	Ticket/Phone
Technical Account Manager	No	No	No	Yes
Encryption at Rest	Yes	Yes	Yes	Yes
Encryption in Transit	Yes	Yes	Yes	Yes
Infrastructure SLA	-	-	99.5%	99.99%
Support SLA	-	-	4 Business Hour Sev-1	1 Hour Sev-1

* Backup retention and schedules are more complex than expressed here, but these values serve as a guide. Very complex schedules and retention periods can be set for Enterprise customers

Environment Infrastructure

Each Ironstar Subscription is made up of one or more Environments. An Environment hosts integrated components like web servers, database engines, storage systems, load balancers, and more, which all work together to provide a single instance of your Application.

In this section, we'll explain the different components in each Environment and how they connect to each other.

With the exception of load balancers, all instances are entirely private and isolated only to your Subscription. There is absolutely no mixing of instances or access between environments and subscriptions.

Instances

When we talk about an "Instance", we refer to a single copy of a specific component responsible for delivering a service. In practice, Instances are most commonly Kubernetes Pods running one or more Docker containers, but we use the term Instance to make these concepts easier to understand for readers who are not familiar with Kubernetes or Docker.

The "Manager" Instance
The Manager Instance (known as the "Admin" instance in Ironstar Legacy) provides centralised control and orchestration for an Environment's deployments and scheduled tasks. Each Environment has exactly one Manager which provides these services:

- SSH Access for authorised users, behind the SSH Proxy Instance
- Scheduled task (cron) execution
- Centralised log collection, where other Instance types forward their logs to the Manager and it saves them to `/app/logs` and makes them available to SSH users
- Deployment orchestration. The manager downloads new releases, unpacks them, runs any custom deploy hooks, every time an instruction is sent to roll out or roll back a release.

With each deploy, the code in the Manager instance is updated but the instance itself is not replaced.

The "Application" Instance
The Application Instance hosts your application's web and application servers. The web server is Nginx, while the application server will be PHP, NodeJS, or a similar service based on your Subscription type. Drupal sites, for example, use the PHP-FPM server behind Nginx.

In the Application Instance, the Nginx container is treated as untrusted since it's generally open to the Internet. Nginx runs as "nobody" in Linux, and has read-only access to the entire filesystem, with the exception of it's own log files. Generally speaking, Nginx delivers static content from the disk anonymously if it can access it, and if not it passes the request to the application process.

Because Nginx will deliver any content it finds on the disk, it's very important that you never make private content accessible in the following locations:

- Under the `/app/storage/public` (or Drupal "files") path

- Under the `/app/site/{public root}` path, where `{document root}` is the public root you specify for your subscription, such as “web” or “docroot”

The application process (such as PHP-FPM) runs your actual application and receives all requests for content that Nginx couldn’t find on the disk. This process has access to read and write files on the following paths:

- `/app/storage/private`
- `/app/storage/public`

It can also write to the application log files, but not to any other location.

The “Database” Instance(s)

For non-static content, your application most likely needs to talk to a relational database management system such as MariaDB/MySQL in order to perform complex, transactional tasks. The Database Instance(s) host these databases.

For Ironstar Enterprise Subscriptions, there are two Database Instances that run in an active/passive configuration with asynchronous replication from the Primary to Secondary server. In addition, two Database Proxy Instances broker connections from the Application and Manager instances to the Database Instances, ensuring that the current Primary is receiving write-only traffic.

Enterprise customers with very busy databases or with a need to perform complex reports can upgrade their subscription to ensure that the Secondary Database Instance is available as a read-only replica.

For non-Enterprise Subscriptions, a single Database Instance is provided in a single availability zone. Both instance types are backed by very fast SSD-style disks with single-digit millisecond-access to the filesystem for read and write operations.

The “Cache” Instance(s)

When a request for content is received in your environment, it first passes through the Cache instance(s). These instances are simple services that look for relay content from the visitor to your application.

The cache server saves a local copy of any content from your Application Instance, provided the following conditions are met:

- The request must be a HTTP GET or HEAD request to be eligible for caching
- The content must not belong to an authenticated user session. The presence of any “Authorization” header or PHP Session cookie in the request will prevent any caching of the response content.
- The content must provide a “Cache-Control” header. If this header is present, the cache server will obey this setting and cache content only as long as instructed.
- For Drupal sites, the request must not in any of the paths that Drupal recommends not be cached, including:
    - /healthz
    - /admin
    - /user
    - /users
    - /feed

- /info
- /flag
- /ajax
- /ahah
- /status.php
- /install.php
- /update.php

Additional Services
Most Subscriptions will make use of secondary services such as Redis, Apache Solr or Memcache. These services all exist in their own dedicated Instance types that are configured according to their specific needs.

In the case of Apache Solr, the Solr core is also mounted on the `/app/storage/solr` path in the Manager so that SSH users have access to view and modify the Solr core on disk.

In all cases, a firewall limits traffic between instance types only to what is strictly required for that service to function. For example, the Application and Manager instances can talk to the Database Instances via the MySQL port (3306), but the Solr, Redis, and Memcache instances have no such access.

Load Balancers

Load balancers are a special class of Instance which exist on the “edge” of an Environment and respond to incoming public requests for the site.

Ironstar Enterprise customers run their own dedicated load balancers across both availability zones, with non-Enterprise customers generally use shared load balancers providing access for multiple Ironstar customers on different domain names.

While there is no security difference between dedicated and shared load balancing, customers wishing to perform load testing will need to upgrade to a dedicated load balancer.

Ironstar Common Infrastructure

Ironstar makes application hosting Environments available to customers from a variety of secure data centre facilities across the world. All of these Environments are built on top of a set of universal low-level components, such as security and network services, orchestration software, and the powerful Ironstar API.

Cloud Infrastructure Providers

Today, Ironstar services are operated from various geographic regions in the Amazon Web Services (AWS) platform. Over time, we plan to add other cloud infrastructure providers such as Google and Microsoft to this list so that customers can choose the underlying provider that best fits their needs. Your Subscription will never be moved from one provider to another without at least 30 days prior notice before the end of your current subscription term, or 180 days prior for customers on yearly subscriptions.

Each provider's geographic region provides at least two physically and logically isolated "availability zones". Each zone is made up of one or more secure data centres.

Customer data **never** leaves the region that your Subscription belongs to. For example, a Subscription created in the Sydney region will only ever have it's data stored in the Sydney region. There is no system in place that allows any customer, Ironstar automated system, or Ironstar team member to inadvertently copy customer data from one region to another using Ironstar's platform and tools, and as such customers can have confidence that data sovereignty requirements are always met.

Metadata relating to services such as DNS entries, log files, and configuration elements may be stored in different regions. For example, the Ironstar API operates from a single region and is used to store and provision certain data in each Environment, such as customer-defined environment variables. These variables are stored encrypted inside the Ironstar API, and as such this information may be stored in a different region to your Subscription.

Kubernetes Clusters

Kubernetes is an open-source infrastructure orchestration platform that Ironstar deploys across multiple availability zones in each region. Each cluster is highly scalable and survivable with no single point of failure. Clusters are provisioned across two or more availability zones.

Clusters are provisioned by the types of Environments they can host. This means that non-production environments are only ever run in non-production clusters (designated as "np" clusters), while critical-workload ("cw") clusters only ever run "production" or "live" Environments. In addition, we run internal development clusters for our own testing purposes in independent and isolated networks.

These cluster roles determine how new changes are deployed. New control-plane software and security patches are first deployed to non-production environments where they are run for 4 weeks

before being ‘promoted’ to critical-workload clusters. This timeline may be accelerated for the release of security features with a Risk Rating of “Medium” or higher.

Storage subsystems

Each environment utilises four distinct storage classes which are treated and protected differently based on the types of customer data they are intended to house.

Ephemeral Data
Whenever your Application is deployed to an Ironstar Environment, your code is unpacked and made read-only on a temporary (ephemeral) disk. Each time a new deployment is performed or your application restarted (such as when we perform scheduled maintenance), the contents of this disk are destroyed and replaced with the new deploy package. It is not possible to write your own data to ephemeral storage.

Ephemeral storage is mounted at the path `/app/site`.

There is no permanent data loss with ephemeral storage, as it only ever contains the code in your deployment package and is read-only to all users. Resources like uploaded content from your users, run-time-compiled content (such as dynamic CSS/JS), and other dynamic objects are stored in Persistent or Database Storage systems which are non-volatile and replicated.

Using read-only ephemeral storage for your code has numerous benefits. The disks used for ephemeral storage are incredibly fast, so access speeds are increased significantly. Similarly, because the code deployed to ephemeral storage is read-only and replaced with each deploy, there’s no chance of attackers being able to manipulate your deployed code at run-time.

Each Environment automatically receives 1GB of Ephemeral Storage for each 1GB of Persistent Storage that you purchase. This allocation is duplicated for any replicas of your Application Instance. For example, an Environment with 10GB of Persistent Storage will receive 10GB of Ephemeral Storage for each replica: If the Environment has 2 Application instances and one Manager instance, it includes 20GB of total Ephemeral storage (10GB each).

If you need more Ephemeral Storage, you can purchase more Persistent Storage. You will need to re-deploy the Environment to have the Ephemeral Storage increase take effect.

Ephemeral storage is not replicated between instances. For example during a deployment the content in /app/site on your SSH Manager instance may be different to the content in your web replicas. This ensures that canary-deployments are possible with no mixing of code between old and new web servers.

Persistent Storage
Each Environment uses Persistent Storage to save content that your users upload or that is dynamically generated by your application. This includes for example Drupal Public and Private Files or Laravel File Storage.

Each Environment’s Persistent Storage volume is broken up into distinct “Paths”. These paths have unique security and access permissions and may be backed up differently or not at all.

Persistent Storage Paths are defined as follows:

- `/app/storage/public` for content that is accessible to all visitors to your Application
- `/app/storage/private` for content that is only accessible to your application server
- `/app/storage/temp` for temporary content such as buffering uploaded files
- `/app/logs` for centralised logs from all of your Environment's components

Unlike ephemeral storage, your Application Instance (ie: PHP FPM) and SSH Manager Instance are able to write content to the persistent disk and that disk is shared between both instance types.

Your web server (Nginx) can only write to the `/app/storage/temp` which it uses for buffering uploads. Nginx has no access to `/app/storage/private` at all, and read-only access to `/app/storage/public`.

Persistent storage is shared between all of your instances in an Environment. For example, if you use SSH to upload a file to `/app/storage/public` on your Manager Instance, it will be immediately visible on all of your Application Instances.

Drupal Environments automatically provision a symbolic link from the Drupal "files" directory (eg `/app/site/web/sites/default/files`) to the public storage path (eg `/app/storage/public/default`)

Content stored in the public storage path is automatically cached by the web server for 24 hours. You can modify this by changing the `static_cache` rules in your `.ironstar/config.yml` file.

In each Environment, the persistent disk is replicated across at least two physically isolated and geographically distributed availability zones. This provides exceptional data resilience and integrity.

The Persistent Storage disk is also used in the control plane for storing system content which is generally very small (<100MB) as well as persistent SSH user home directories. This means that any files that your SSH users save in their home directories count towards your Persistent Storage usage. Content stored by the control plane does not count towards your Persistent Storage usage.

Persistent Storage is also used to store data from the Redis and Solr services, which counts towards your total Persistent Storage Usage. Redis, Solr, and user home directories are not backed up.

The Persistent Storage filesystem is mounted with the Linux `noexec` flag set. This prevents direct execution of binaries and scripts on Persistent Storage. As an example, if an attacker was somehow able to upload a php file to the Drupal Public Files path with an execute flag set, they would not be able to call that file remotely.

It is possible to bypass this restriction by passing the file to another binary which is executable such as calling the 'php' executable directly. Ironstar Enterprise customers running Drupal can upgrade to a "Hardened Application Instance" which removes access to common executables like `bash` and `php`, mitigating this attack vector.

Ironstar recommends that maximum Persistent Storage usage be limited to 250GiB and less than 10,000 objects per directory. Larger volumes take longer to back up and as such to restore, and you should factor this into any Disaster Recovery Planning exercises that you undertake.

Database Storage
Relational databases like MariaDB require very high performing storage systems in order to quickly read and write database content from disk. For this reason, we use SSD-or-better disks in a single availability zone. These disks are replicated within the availability zone they are hosted in, ensuring availability levels beyond 99.99%.

Database storage files are not directly accessible from your application, and instead may only be accessed using the database service over the network.

Backup Storage
Persistent and Database Storage systems both use a similar backup mechanism to automatically perform a full backup and save it to a redundant and highly secure distributed storage system. Backups are never retained on the same infrastructure hosting the Environment, and are automatically distributed across at least three availability zones.

Each Environment is backed up at least once per day between 10PM and 2AM local time to the region that Environment is hosted in. The specific time of the backup within this window is calculated randomly when the Environment is created. Ironstar Enterprise customers may upgrade their service to include more frequent backups of Production, up to once every 5 minutes.

Default backup retention and aging periods differ by Subscription type. Ironstar Enterprise customers can upgrade to custom retention schedules with up to 7 years of retained backups.

Each Path in the Persistent Storage system is backed up separately, allowing for granular restores. Backups are performed in sequence, with Paths being backed up before Databases. This means that the start and end time of each individual backup component may be different.

Customers may perform backups of any Environment on-demand. By default, only 10 of these manual backups for any single environment can be in place at one time.

Infrastructure Networks

Ironstar operates several network systems to provide communication between the Internet and customer environments, and between our management control plane and other systems.

Cluster Networking

Each Kubernetes Cluster runs it’s own dedicated network with no connectivity between clusters. Environments can still talk to other Environments via public Internet access if both environments permit access in their respective network policies.

Inter-network access (such as from the Management Network to the Master Nodes Network) is only ever performed using pre-created secure network links and this traffic never traverses the Internet. Within each cluster, several purpose-built networks are defined:

DMZ
The DMZ network allows public access to the Application instances from the Internet. This is how your visitors obtain access to your site.

The DMZ only allows incoming port 80 (HTTP) and port 443 (HTTPS) connections. Connections on port 80 are automatically routed to the same hostname on port 443 via HTTPS. Non-TLS encrypted connections to Ironstar environments are strictly prohibited.

Ironstar Legacy customers also obtain SSH access to their environments via the DMZ, and can supply a list of IP addresses that are allowed to connect via SSH.

When a connection is allowed through the DMZ (ie, is not firewalled or subject to DDoS or other attack filtering), it is passed to the Worker Nodes network.

Worker Nodes Network
The Worker Nodes host customer Environments and run on private IP addresses. Access to these worker nodes is only possible through the DMZ or via the SSH Proxy Network, and only via HTTPS or SSH.

Master Nodes Network
The Master Nodes Network hosts the Kubernetes “master” servers which run on private IP addresses. These servers run management software that is responsible for orchestrating the various workloads hosted within the Kubernetes Cluster.

The Master Nodes Network is not able to be accessed from the Internet and only accepts connections for specific services from the Management and Worker Nodes Networks via TLS-encrypted connections.

SSH Proxy Network
With the exception of Ironstar Legacy, each Subscription has its own dedicated SSH Proxy instance. This instance provides inbound filtering and tunneling of SSH connections to any Environment inside that subscription.

The SSH Proxy Network allows for incoming SSH connections from the Internet, and customers may provide a list of specific IPv4 addresses that can access it via SSH. The SSH network is heavily restricted and no outbound connections are possible except in response to permitted incoming connections.

Management Network
The Management Network provides authorised and authenticated Ironstar engineering and support team members with secure access to administer and maintain the platform. The Management Network is hosted independently of any individual cluster and only permits access to the services necessary for the routine maintenance of the platform.

Access to the Management Network is achieved through three-factor authentication where permitted Ironstar staff members must authenticate with a password, private encryption key, and provide a unique cryptographic token.

API Network

The Ironstar API is hosted in it's own network with strictly controlled access to Kubernetes clusters. The API only responds to incoming HTTPS connections from the Internet.

Shared Security Model

While Ironstar is responsible for provisioning a secure and reliable infrastructure platform for your Applications, we are not responsible for the security of the Applications themselves. Together, you and Ironstar share responsibility for the overall security platform with key areas of responsibility. This section identifies those areas of responsibility for the avoidance of any doubt.

The most effective way to consider this division of responsibilities is that Ironstar and its partners are responsible for the security "of" the platform, while you (and your affiliates or partners) are responsible for the security of your application "in" the platform.

Ironstar Security Responsibilities

Ironstar is solely responsible for ensuring that:

- Public access to your Environments is only possible via encrypted HTTPS
- Where configured by you, outbound access from your environment is limited only to the destination IP addresses or networks that you provide, and to our management control plane and storage systems as is necessary to operate the platform.
- SSH access is only possible to the usernames and access keys (and, if configured, MFA tokens) that you have granted access to your Environment
- Where configured by you, inbound access to your SSH manager instance or application instance(s) is only possible from the IP addresses or networks that you provide
- Your code base when deployed is read-only
- Your "public" files are read-only by your Web Server (nginx)
- Your "public" files are read-write by your Application Servers (eg PHP)
- Your "private" files are read-write by your Application Server
- Your "log" files are read-only by all SSH users except Ironstar administrators
- Database access is only possible over secure TLS connections from your Application Server or SSH Manager Instance
- Database connections are only possible using mutual TLS authentication and encryption
- Connections to Redis or Solr are encrypted. (Memcache encryption is not supported)
- Persistent and Database storage is always encrypted using AES256 (or better)
- Performing security and maintenance patches to all Ironstar-managed components in a reasonable timeframe based on the associated security risk assessment
- Ensuring that our systems are generally developed and maintained in accordance with industry best practices and ISO 27001 and PCI DSS standards.
- Where Ironstar engages a third party to perform certain services, we must ensure that the partner complies with these requirements.

Customer Security Responsibilities

At a high level, you are responsible for your Application. Ironstar does not provide software development services to any customer and makes no representations that we can or will provide maintenance services for your Application.

Specifically, you are solely responsible for ensuring:

- That your Application receives all appropriate security and maintenance releases to ensure it remains secure from any common or published vulnerabilities
- That your team operates according to best practices and the security framework that you designate
- Provisioning and maintaining user accounts for your access to your Application
- Your users protect and secure their access to your application and the Ironstar Environments and API
- Compliance with any security frameworks or industry standards to which you are legally bound, such as PCI DSS compliance.
- Training your team in security best-practices and your internal policies and controls

In addition, Ironstar is not responsible for any action taken by you or your application's visitors (whether authorised by you or not). Further, Ironstar is not responsible for the security of the platform where you have made changes that circumvent Ironstar's efforts to secure the platform.

Access Control Systems

Ironstar API

The Ironstar API provides orchestration of your Environment(s), Backups, and other key components. It is accessible via *https://api.ironstar.io*

You may nominate any user at any time to have access to your Subscription and Environments via the Ironstar API. These users must authenticate with an email address and password. In addition, you can require that users with access to your Subscription also authenticate using a time-based cryptographic token (MFA) before being granted access the API.

When a user receives access to the API, they are issued an authentication token that can be used to authorise subsequent actions. These tokens expire after 2 weeks and can be renewed at any time up to 2 weeks before expiry without needing to re-login.

Role-based Access Control (RBAC)

API users can be provided one of three access levels for your Subscription:

- Read-only
- Read-write
- Administrator

Read-only users have access to view Subscription and Environment information, but can not modify this information. They can also open and view support tickets. Read-only access is generally suitable for users who require access to support and billing information.

Read-write users have access to modify Subscription and Environment configuration, including performing deployments, modifying settings, and copying data between Environments.

Administrator users have all the same privileges as **Write** users, but also have access to add or remove Users from the Subscription, or to terminate the Subscription.

Task-based Access Control (Service Accounts)

Internal systems such as the backup agent or SSH Proxy servers interact with the Ironstar API in very specific ways. Each of these systems receives its own unique authentication token when provisioned by the Ironstar API. These keys are unique to each of these services for each Environment or Subscription.

These keys allow automated systems to make requests to the API to receive new data (for example, the SSH Proxy instance asking for a list of allowed users). These keys are rotated periodically, and are only granted access to the specific data required for that system to perform its core functionality.

SSH Access

Each Ironstar Environment has a dedicated SSH instance (the “Manager Instance”) that provides developers and administrative users, as well as Ironstar support team members, with access to the Application’s file system, database, and other tools.

For Ironstar Legacy, SSH access is provided by a load balancer in the DMZ network. Customers can specify individual networks that are allowed to access SSH (provided they have a valid username and SSH key). For all other platform types, access is provided by an SSH Proxy server for each Subscription.

For non-Legacy environments, in order to be granted SSH access, each user must connect to the SSH Proxy using:

- Their unique username, which is generated by the Ironstar API
- Their personal SSH key which matches the one provided to the Ironstar API
- Be connecting from an authorised network (if specified)
- Present a secure cryptographic token (MFA) at least once every 12 hours (if required by the Subscription configuration)

With the exception of Ironstar Free customers, each Subscription has its own unique SSH Proxy instance. The SSH proxy is responsible for “tunneling” connections to each Environment’s SSH Manager instance.

The Ironstar API can not prevent users from sharing a username and SSH key, but does prevent the same SSH key from being used in multiple individual accounts.

Once authenticated via SSH, Ironstar can not control or monitor what users do within the system. They will be able to view and modify files in Persistent Storage, connect to the Database and other services, and interact with the Application including by gaining administrative access. As such, SSH access should only be provided to fully-trusted users and developers.

Load Balancer Access Controls

Each Environment is accessible via either a shared or dedicated load balancer. Customers can configure a single username and password (HTTP “Basic Auth”) to provide minimal access control to an Environment. This is generally very useful for non-production environments to prevent them being accessed publicly.

Each load balancer for an Environment can have only a single username and password of this type, and as such the password does not uniquely identify any individual user. You should use security controls within your Application to authenticate users.

Inbound Network Filtering (firewall)

Customers on the Ironstar Advanced or Ironstar Enterprise platforms may supply a list of authorised networks (IPv4) that are granted access to connect to each Environment either via SSH or HTTPS. If you provide any authorised networks, all other networks are refused access.

Alternatively, for HTTPS traffic you can specify a list of one or more networks (IPv4) that are denied access. Networks not in this list are granted access.

Ironstar Enterprise customers with the CDN upgrade can add geographic filtering to block all IP addresses from specific countries, or to allow IP addresses only from a single country. In addition, Enterprise customers can request that only their CDN be granted access to the Environment, making the origin web servers completely inaccessible except via the CDN.

Outbound Network Filtering (firewall)

By default, your Application and Manager instances can initiate new connections to the Internet. This ensures that external services like payment gateways and mail services are reachable.

Customers on Ironstar Advanced or Ironstar Enterprise can supply a list of allowed networks that can be accessed from their Environments. If this list is provided, then all other networks are blocked. This is an incredibly effective way to ensure data security as it significantly limits the ability of any successful intrusion to export stolen data to a remote system.

Application and Manager instances can always respond to incoming requests that are allowed through the firewall.

Ironstar Internal Systems

Ironstar’s corporate network is disconnected from all customer and management networks. Non-support and non-engineering Ironstar team members receive no access to customer environments. When Ironstar team members are on leave or have tendered their resignation, their access to systems is suspended.

Where Ironstar uses third-party providers for certain of our internal systems (such as email hosting or billing services), we use only those systems that comply with our security policies and best-practices.

Compliance and Industry Security Controls

Our platform and services are built to comply with the following industry security standards and controls:

- Payment Card Industry Data Security Standard (PCI DSS)
- Health Insurance Portability and Accountability Act (HIPAA)
- EU General Data Protection Regulation (GDPR)
- The (Australian) Privacy Act 1988
- ISO 27001

Ironstar Enterprise customers may request assistance in executing their own audits and certification process for one or more of the above standards. Charges for this work will apply and a quote can be provided based on the requirements.

A copy of the “Ironstar and PCI DSS” whitepaper can be provided upon request. This document explains each of the PCI DSS requirements and how Ironstar Enterprise customers can leverage their Ironstar platform to achieve compliance.

Data Sovereignty Controls

Many of our customers have very specific requirements about where their data is stored and transmitted. For this reason, each Subscription is assigned to an Ironstar geographic region and can not be moved to another region for any reason. Customers wishing to move their Subscription to another region must register a new Subscription in the new region and manually migrate their data over.

Certain data is naturally “region-less”, or needs to be stored inside our API which may not be in the same region as the Subscription’s region. The following table identifies these data types and where they are stored:

Data Type	Region
Code packages and Git repositories uploaded to Ironstar systems	Subscription’s Region
Database, Persistent, and Ephemeral Storage	Subscription’s Region
Backup Storage	Subscription’s Region
Domain Name System (DNS) data	Global
Email send through the Ironstar system	Global
Ironstar API-hosted Data, such as subscription information, custom environment variables, SSL keys, and other information uploaded to the Ironstar API*	Sydney Region (au1) Ohio Region (us1) Frankfurt Region (eu1)

* The Ironstar API is primarily hosted from Sydney, but with replication and failover supported to the US and EU.

Security Testing and Review Procedures

Documentation Review and Training

Ironstar security and process documentation is reviewed by the Ironstar Engineering Team every 6 months to ensure they are keeping up with industry standards and best practices.

Following each review process, Ironstar team members undergo training in the updated procedures. Following this training, team members must undergo a validation process with their managers on the aspects of the Ironstar security framework that relates to their role.

Penetration Testing

Public-facing IP addresses used by Ironstar for the Ironstar API and other management systems undergo automated security scanning every 3 months.

Customers may request Ironstar's assistance in performing their own penetration tests of their Environment(s). Such tests must be scheduled in advance and the emergency contact details of the tester(s) must be provided to Ironstar before commencing.

Disaster Recovery and Reliability Controls

High Availability

"High Availability" refers to the ability of an individual system to recover from failure and be restarted on replacement infrastructure. Ironstar clusters are always provisioned with at least N+1 capacity, meaning that at least 1 replacement Worker Node is available to automatically recover lost capacity if another single Worker Node fails.

Because High Availability systems are deployed as single units with no online spares, recovery from failure in these components can take several minutes (generally between 5 to 20 minutes, depending on the component).

For all Subscription types, certain services are only provisioned in a High Availability model. These include:

- Apache Solr
- Redis
- SSH Access
- Scheduled (cron) Tasks
- Log collection and forwarding (during an outage, logs are retained and transmitted later when the "Manager" instance returns to service after approximately 5 minutes)

For Ironstar Enterprise customers, the Application and Database instances are provisioned in a Fault Tolerant arrangement.

Fault Tolerance

“Fault Tolerance” refers to the ability of an individual system to sustain failure with extremely minimal or no interruption. This is achieved by ensuring that at least two copies of the affected component are online at all times, with N+1 capacity ensuring that the loss of any one component can be sustained even during high traffic.

For Ironstar Enterprise customers, the Application and Database servers are deployed in a Fault Tolerant arrangement. All Application services are “active” and equally share incoming traffic.

Database servers for Enterprise customers are provisioned in an “active/passive” arrangement, with asynchronous replication of all traffic to both database servers. Monitoring will alert the Ironstar support team if replication lag falls to more than 5 minutes at any time.

Memcache is provisioned as two replicas across two availability zones. Customer applications are responsible for maintaining proper failover for Memcache and duplicating data across both services. Memcache for PHP does not support TLS encryption, and as such we strongly recommend the use of Redis.

Disaster Recovery

Disaster Recovery refers to the ability for an Environment or Ironstar control system (such as the Ironstar API) to survive the partial or total loss of a single availability zone in any region.

The nature of our infrastructure is that it is already replicated across multiple availability zones. As such, any single Kubernetes cluster can survive the loss of a single zone without any intervention.

Ironstar maintains a Disaster Recovery Plan for our engineering and support team members to use whenever such a disaster occurs. This instructs the team on what to do to ensure clusters are ready for a disaster, and what to do if the automated recovery processes are failing for any reason. In addition, communications protocols are defined to ensure customers receive timely and relevant information during the disaster.

While the cluster infrastructure is fault tolerant, Environments not on the Ironstar Enterprise platform have their databases stored within a single availability zone. As such, recovery from a disaster affecting the zone hosting the Environment’s database will require rebuilding of the database in a replacement zone. There is no SLA provided for recoveries of this type, however recovery for Enterprise customers is within less than 5 minutes with no more than 5 minutes data loss.

Survivability Across Regions

For customers requiring protection from total region loss, custom solutions are available. These involve replicating backups from one region to another and providing standby infrastructure in the target region that can be recovered to in the event of the loss of the primary region.

The cost and complexity of such services depend heavily on the Recovery Time Objective (RTO) and Recovery Point Objective (RPO) of the customer, and are only available to customers on the Ironstar Enterprise - Dedicated Cluster platform.

Backups

Customers can recover their data from any backup in the Ironstar system. Backups from one environment can be restored to any other environment at any time. Backups are stored in a distributed system across 3 or more availability zones, and are encrypted.

Backups are taken at least once every day at a predictable schedule. Ironstar Enterprise customers can request custom schedules and more frequent backups, up to once every 5 minutes.

Performance Controls

Scaling for Ironstar Clusters

Each cluster is provisioned to scale automatically to meet the demand of new Environments and increased workload from customers. This ensures that during busy periods, our systems will automatically add capacity to the underlying platform.

Reductions in capacity are performed manually during the Maintenance Window. For Ironstar Enterprise customers, there is no downtime from down-scaling operations. For all other Subscription types, there may be brief downtime between 5 and 10 minutes as work is rescheduled.

Within a cluster, all workloads specify a minimum and maximum resource commitment. The maximum commitment is the amount of RAM and CPU that you purchased for your Environment. For Ironstar Enterprise customers, there is no minimum commitment, as you are guaranteed access to the purchased resources, while all other Subscription types share some of the resource limit with other customers. In practice, this is rarely an issue and if maximum capacity is required by multiple customers on the same node at the same time, the cluster will be automatically scaled out.

Scaling for Customer Environments

Ironstar Enterprise customers can purchase automated horizontal scaling for their production Environment. This solution provides a minimum and maximum number of replicas for the Application Instance, with the system automatically creating additional replicas up to the maximum as needed to ensure CPU and memory usage remains stable.

Ironstar Advanced customers can choose the number of Application Instance replicas they wish to run as part of their Subscription agreement. If immediate increases in capacity are required, they can contact the support team to request an immediate increase.

All customers can increase their resource capacity by opening a support request and agreeing to the temporary increase in fees without needing to execute a new services agreement.

Load Testing

Ironstar Enterprise customers can schedule load testing of their Production environment with at least 7 days notice. The source IP address(es), contact details for the tester(s), and the time of the test must be provided in advance.

All other subscription types do not allow load testing. Customers who perform unauthorised testing may have their Subscription disabled until the tests are confirmed to have stopped.

Physical Security Controls

Physical data centre access is provided by our Cloud Infrastructure Providers, namely AWS. The following information is taken from the AWS Security Whitepaper which provides significant detail about how AWS' platform is provisioned and secured.

Fire Detection and Suppression

Automatic fire detection and suppression equipment has been installed to reduce risk. The fire detection system utilises smoke detection sensors in all data center environments, mechanical and electrical infrastructure spaces, chiller rooms and generator equipment rooms. These areas are protected by either wet-pipe, double-interlocked pre-action, or gaseous sprinkler systems.

Power

The data center electrical power systems are designed to be fully redundant and maintainable without impact to operations, 24 hours a day, and seven days a week. Uninterruptible Power Supply (UPS) units provide back-up power in the event of an electrical failure for critical and essential loads in the facility. Data centers use generators to provide back-up power for the entire facility.

Climate and Temperature

Climate control is required to maintain a constant operating temperature for servers and other hardware, which prevents overheating and reduces the possibility of service outages. Data centers are conditioned to maintain atmospheric conditions at optimal levels. Personnel and systems monitor and control temperature and humidity at appropriate levels.

Management

AWS monitors electrical, mechanical, and life support systems and equipment so that any issues are immediately identified. Preventative maintenance is performed to maintain the continued operability of equipment.

Storage Device Decommissioning

When a storage device has reached the end of its useful life, AWS procedures include a decommissioning process that is designed to prevent customer data from being exposed to

unauthorized individuals. AWS uses the techniques detailed in NIST 800-88 (“Guidelines for Media Sanitization”) as part of the decommissioning process.

Network Security Controls

Ironstar deploy software-defined networking with automated reconciliation of protected Environment network firewall rules.

Cluster-level Firewall Rules

Each cluster is protected by a range of firewall rules and network segmentation that prohibits unauthorised traffic flows between cluster systems and from external sources. This first level of firewall rule allows incoming access more broadly and is further restricted per-environment

The cluster-level rules are as follows:

	Source Target	Destination Target	Allowed Port(s)
Incoming HTTPS for Application Instances	From Internet	Load Balancers	443 (HTTPS)
Incoming HTTP for Application Instances (automatically redirected to HTTPS)	From Internet	Load Balancers	80 (HTTP)
Incoming SSH for Legacy SSH Service	From Authorised Partner Networks only*	Legacy SSH Load Balancer	22 (SSH)
Incoming SSH for Manager Instances	From SSH Proxy Servers	SSH Manager	22 (SSH)
Incoming SSH for SSH Proxy Server	From Customer Specified Networks	SSH Proxy Instance	22, 443 (both SSH)
Internal Customer Services Traffic	Worker Nodes	Worker Nodes	443 (HTTPS) 6789 (Redis) 11211 (Memcached) 8983 (Solr) 3306 (MySQL)
Internal Management Traffic	Nodes	Nodes	443 (HTTPS)
All Outbound Traffic	Worker Nodes	Internet	All

All other traffic flows not listed above are implicitly denied. For example, SSH Proxy Servers can not connect to the Internet, nor can Worker Nodes be directly accessed from the Internet (they have non-routable private addresses).

* Legacy SSH load balancing is provided in partnership with a pre-approved selection of digital agency partners. Only these partner’s networks have access to this system.

Environment-level Firewall Rules

In addition to the more broadly defined cluster-level rules, each Environment has its own set of rules that enforce specific access controls for that environment. By default, these are as follows:

	Source Target	Destination Target	Allowed Port(s)
Incoming HTTPS for Application Instances	From Load Balancers	Application Instance	443 (HTTPS)
Incoming SSH for Legacy SSH Service	Legacy SSH Load Balancer	SSH Manager	22 (SSH)
Incoming SSH for Manager Instances	From Subscription's Dedicated SSH Proxy	SSH Manager	Environment-specific SSH port
All Outbound Traffic	SSH Manager Application Instance	Internet	All
Intra-environment Redis	Application Instance Manager Instance	Redis Instance	6789 (Redis)
Intra-environment Memcache	Application Instance Manager Instance	Memcache Instance	11211 (Memcached)
Intra-environment Apache Solr	Application Instance Manager Instance	Solr Instance	8983 (Solr)
Intra-environment MySQL	Application Instance Manager Instance Backup Service	Mariadb Instance	3306 (MySQL)

Ironstar Enterprise and Advanced customers can further improve this network firewall by providing a list of approved outbound networks for the SSH Manager and Application instances. This will prevent all outbound traffic from these instances except to the destination networks provided, and to Ironstar management and storage systems necessary to operate the platform.

As with cluster-level firewall rules, any permissions not explicitly allowed here are implicitly denied. For example, the MariaDB/MySQL servers can not talk to the Internet or any other system: they can only respond to incoming connections from approved instances.

Software Defined Networking

Cluster-level firewall rules are defined as code by Ironstar engineers, and as such are subject to our standard Software Development Lifecycle controls, including peer-review and routine auditing.

Environment-level firewall rules are configured automatically by our Environment control software. These firewall rules are enforced every time an Environment configuration changes, and routinely every 10 hours.

Ironstar Software Security

All Ironstar infrastructure is software-defined. This includes infrastructure provisioning scripts, controller software which runs in-cluster and responds to API events, and the Ironstar API itself.

We develop this software accordingly to industry best-practices and maintain a rigid Software Development Lifecycle (SDLC) with the following controls:

- Software is developed and tested only in isolated non-production environments.
- Production data and databases are not used in software development and testing
- Automated tests are used to verify the stability and reliability of software before release
- Source code access is restricted to Ironstar team members that actively require it
- Software is run in secure “distroless” environments with no access to shell utilities
- Developers receive instruction in defensive development practices and threat mitigation
- All code is peer-reviewed and audited before being accepted for release
- All releases have comprehensive production verification testing and roll-back plans
- With the exception of the Ironstar API, all Ironstar software is hosted in private networks with strict firewall rules and no access to the Internet

Security Controls for Ironstar Staff Members

Before commencing employment, all Ironstar team members undergo a vetting process to ensure their suitability to work in high-risk environments and should be granted access to sensitive customer information.

At least once every 12 months, each team member must also perform a criminal history check relative to their current place of residence (eg, Australian residents must undergo a Australian National Police Check).

Ironstar team members' use of Ironstar systems and networks is monitored and routinely audited to ensure compliance with our standard and processes. Team members who repeatedly fail to follow published processes are subject to disciplinary action and possible termination of employment. Team members who are found to be wilfully misusing equipment or compromising system security are subject to immediate termination without notice and referred to the relevant authorities.

Access to Ironstar management systems is only permitted from Ironstar-provided devices. Team members may not use their own devices to access Ironstar systems.

Change Management

Non-customer systems are subject to Ironstar's Change Management Process. This process ensures that changes to our systems and processes are subject to rigorous scrutiny and validation before receiving approval to be implemented.

Before being implemented, any Change Request must first satisfy the following requirements:

- Have a justifiable need explaining why the change is required
- Provide implementation, verification, and roll-back documentation for the change
- Be assigned both a Change Implementer and Change Validator, which cannot be the same person

Once this information is available, our internal Change Advisory Board (CAB) will receive the Change Request and if approved, assign it to the next scheduled Maintenance Window (unless the implementation target requests otherwise). The Change Implementer then adds it to the scheduled notification system and the affected customer(s) will be notified at least 2 weeks prior to the implementation of that change.

Following the change implementation, the Change Validator is responsible for documenting any changes to the system and communicating those changes to any relevant team members and customer(s).

In certain circumstances, Change Requests may receive automated approval if they satisfy the following criteria:

- Are classified as Very Low Risk and do not impact running Critical Workload Environments
- Have a pre-defined Implementation Plan because they are a Routine Change
- Have an automated Change Verification process
- Have an automated Rollback process

In addition, Emergency Change Requests can be approved by any two members of the CAB if the full CAB is unable to be convened within 2 hours. Emergency Change Requests may be implemented meeting the full Change Request requirements in exceptional circumstances, such as when the change is necessary to protect customer data.

Customer systems such as Environment configuration and network controls are not subject to Ironstar's change management process, and Ironstar do not interface with our customers internal change control processes except under certain, pre-arranged circumstances.

Software Patching and Maintenance

The Ironstar platform is made up of integrations between several individual third-party systems, such as OpenSSH, Nginx, PHP, Kubernetes, MariaDB, and more. To manage these systems, we ensure that we maintain the most up-to-date software releases for these systems as possible, while maintaining a defensive posture against possibly unstable code in new releases.

Monitoring for Component Updates

Every third-party component in the Ironstar platform is catalogued and tracked. For each of these, our engineering team is responsible for following these components and any updates to those components, as well as published advisories about security or other issues with the component.

Classifying Software Updates

When the engineering team identifies a new version of a third-party component, they will assign a Risk rating to the release. This classification is based on the risk to the affected systems if the change is not implemented, and this may be a security risk or a performance or other risk (such as potential system instability in production).

The Schedule associated with each Risk level defines how quickly that release must be made available in all Critical Workload Environments, and includes a period of validation in Non-Production Environments and testing in internal non-customer test environments.

Criteria	Risk	Schedule (Days)
Routine software release with no known vulnerabilities	**Low**	90 Days
Addresses minor security risks with no published exploits or with mitigations in place by Ironstar that render it ineffective	**Medium**	60 Days
Addresses significant security risks that may impact certain customers with improper application configurations, where a workaround by Ironstar is not effective or not possible	**High**	14 Days
Addresses a major security risk that will impact customers with published exploit code available publicly and no workaround in place	**Critical**	48 Hours

Software Maintenance Plan

After testing, Low and Medium Risk releases are grouped and applied to non-production Environments in the next available Maintenance Window. After 30 days, they are considered ready for Critical Workload Environments and are included in the next Maintenance Window.

High and Critical Urgency releases are not grouped and applied to non-production clusters immediately. After 7 days for High Urgency, or 24 hours for Critical Urgency, they are promoted into Critical Workload Environments.

Data Security and Integrity

Multiple systems and processes ensure continuous protection and security of data you and your visitors upload to your Ironstar Environment. These controls are generally “above and beyond” what is standard for the class of hosting that Ironstar provides, and satisfies some of the most rigid security and integrity standards and frameworks.

Data Isolation

On all Ironstar platforms, each Environment has its own independent data plane. There is no sharing of data systems between Subscriptions or even between Environments for the same Subscription. The only exception to this is the storage of backups, where backups are stored together at the Subscription-level, which enables restore of backups from one Environment into another. Ironstar never mixes data storage between customers, and they exist as completely independent storage systems.

Encryption

Ironstar storage systems are all encrypted at rest using at least the industry-standard AES-256 algorithm. The key used to encrypt these volumes is unique to the Kubernetes Cluster the data resides in. Backups are also stored encrypted, and the encryption key used here is unique to the backup system and not shared with any Kubernetes clusters or other systems.

When being accessed by a running Environment, traffic between the Worker Node running the Environment and the underlying storage system is encrypted using TLSv1.2 or better. Access policies enforce the use of this encryption, and it is not possible to mount these storage systems without encryption.

Backups

Backup systems run “out of band” at Ironstar to ensure that backups have minimal impact on running Environments. When a backup is performed, the contents of the backup are first streamed to a staging volume before being compressed and uploaded to remote storage.

Each Subscription has its own independent backup storage pool that is also AES256-encrypted at rest and requires TLS encryption to be accessed. This storage pool can be considered an “offline” pool for regulatory purposes because it is detached from your Environments.

You can perform a restore of any individual backup file to any environment at any time, but can also configure protection to prevent accidental restoration to any environment. This protection is enabled for your “Production” environment by default. You can also synchronise data between your environments as a shortcut to individual backup and restore operations.

Be advised that when moving data between environments, Ironstar does not perform any sanitisation of that data, which may result in your copying personally identifiable information from production to non-production systems.

Monitoring

Ironstar performs internal monitoring of more than 700 unique metrics. This includes standard data points like CPU, memory, and disk usage, as well as more complex and application-specific data points like FPM request volumes, MySQL replication delay, load balancer requests per second, and much, much more.

Each Environment is automatically tracked across these metrics and those metrics are retained in our system for at least 30 days. We use this information to perform capacity analysis, post-incident retrospection, and to trigger alarms to our 24x7 support team when anomalies or problems are detected.

This system is also capable of predicting issues, such as when disk usage is growing fast enough to lead to exhaustion within 12 hours, even if disk usage is currently less than the 80% threshold for an alert to be raised.

Currently, customer access to monitoring systems is not possible, but we do plan to release access to a subset of these metrics in the near future.

Application Readiness Test

One of the most useful monitoring metrics available is the application readiness test. This test monitors a single URL of a customer's application and, if it detects a failure, will restart the affected Application Instance. By default, this endpoint monitors the web server system itself, but customers can change this URL to be any path inside their site that they prefer. Most customers configure this to hit a specific health-check URL that confirms connectivity to their database, external payment gateways, and more.

This same test endpoint is also used when a new version of the Application is deployed. Each new instance of the application must pass this check before it is brought online. If all endpoints fail to come online within 15 minutes (configurable), the deployment is automatically rolled-back. This system ensures that deployments which fail do not cause downtime.

Logging

Log data is an essential part of our platform and enables both our engineers and our customers to review, analyse, and resolve application problems.

Ironstar Internal Logging

Log forwarding agents run alongside internal systems and collect log data as it is emitted and then forward those logs, along with additional metadata, into a secure centralised log storage system. Access to this log storage system is configured so that only Ironstar engineering and support team members can view logs, and no users can delete logs without requesting elevated access.

Customer Application Logging

Log forwarding agents run inside each customer environment to collect logs from customer systems, including:

- The cache server “access” and “error” logs
- The nginx web server “access” and “error” logs
- The PHP FPM “access”, “error”, and “slow” logs
- Deployment and roll-back logs
- Application-specific log files for Drupal, Laravel, and NodeJS.

All of these logs are streamed to the SSH Manager instance, where they are written to disk and made available to SSH users read-only at the `/app/logs` Path.

Currently, the logs Path is backed up every day along with all other Paths. In the future, this will be changed with logs being exported to a centralised storage system and made available

The period of time that logs are retained is determined by the Subscription type. Ironstar Enterprise customers can request custom log retention schedules and the installation of additional log forwarding agents such as Splunk, Datadog, or Sumologic with export to their own systems at additional cost.

Glossary of terms

“Application” means any software that you install into any system provided by Ironstar, such as a Drupal Content Management System, Laravel, Wordpress, NodeJS, or other similar software.

“Environment” means an isolated platform for you to run a single instance of your Application. Environments are provisioned into two classes: “non-production” and “critical workload (production)”. There is no degradation or limitation of security controls between these environment classes.

“High Availability” refers to the design practice of having standby infrastructure available or rapidly-deployable as replacement capacity of failed components. Systems that are “Highly Available” are therefore automatically recovered but may experience downtime (generally between 10 to 30 minutes) while replacement systems are brought online and provisioned.

“Fault Tolerance” refers to the design practice of having multiple systems online at one time with sufficient free capacity to handle the maximum expected utilisation with no or very minimal (< 5 minutes) downtime in the event of individual component or datacentre failure.

“Application instance” is a single copy of your application running with a Web Server (Nginx) and an Application Server (such as NodeJS or PHP FPM). The web server delivers static content and passes application requests back to the Application Server. Environments can have one or more Application instance replicas, which allow for fault tolerance and higher capacities.

“Manager instance” is a single utility that provides secure SSH access, deployment orchestration, log collection and forwarding, and scheduled task (cron) execution for an Environment.

“Maintenance Window” is the time during which scheduled maintenance is performed. Scheduled maintenance may occur every Wednesday between the hours of 0200 and 0500 in that region’s timezone.

“Worker Node” refers to a Kubernetes Cluster Node that is actively running one or more customer Environments.