ironsphere

Ironsphere Controller

Solution Brief

Table Of Contents

1. Purpose

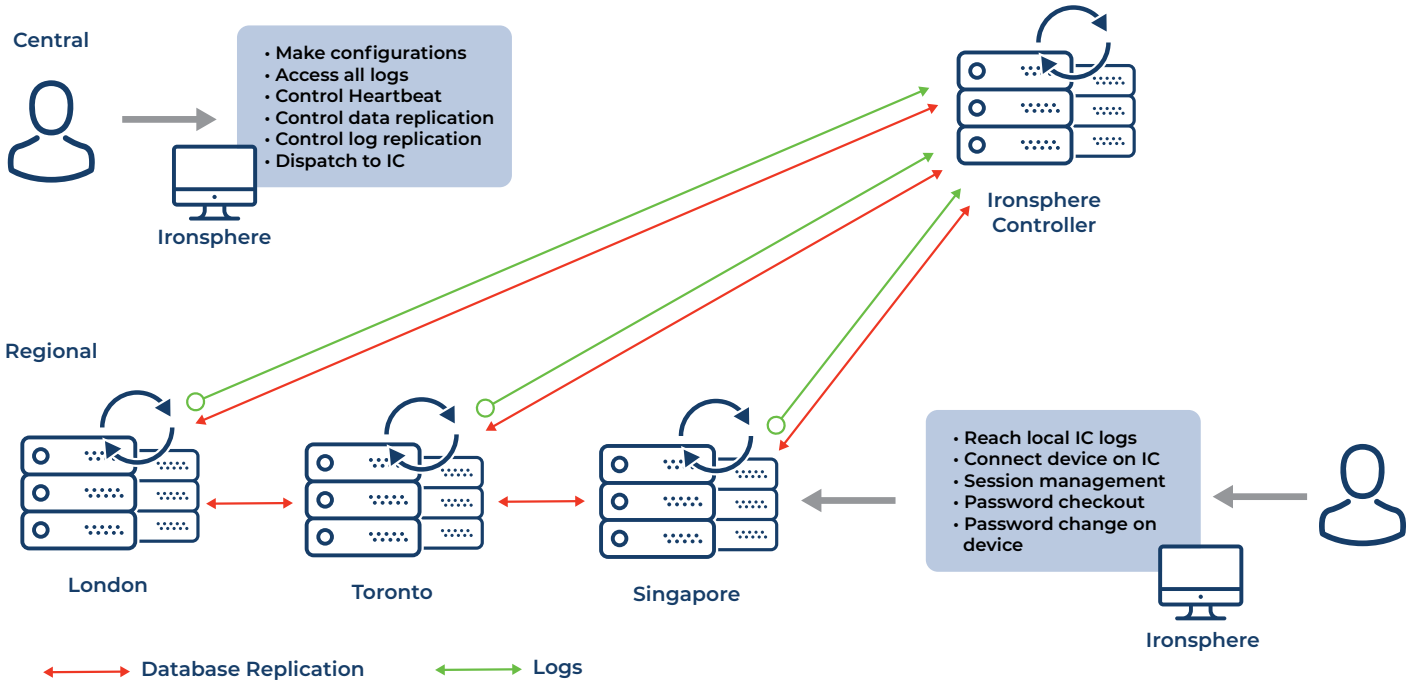
The Ironsphere Controller (IC) is a controller layer within the layered architecture of the Ironsphere software, which sits above the Ironsphere Instances.

The main purpose of the IC is to manage the Ironsphere Instances in a controlled way with a centralized system.

2. High Level Solution

- All configurations are done on the Ironsphere Controller using the GUI (details in Table 1 under the "Functional Solution" section below).
  - o Configuration changes/updates made within the Ironsphere Controller replicate real-time onto the Ironsphere Instances. The latency of the replication is dependent on the network traffic latency between the instances and the Ironsphere Controller.
- Privileged accounts user access and activity logging happens at the regional instance level and the Ironsphere Controller pulls these local logs from the Ironsphere Instances.
  - o The frequency of the log collection can be configured on the Ironsphere Controller.
  - o Local users with the required permission can access these logs through the Ironsphere Instances.
  - o Admin users on the Ironsphere Controller with the required permission can access all the logs.
- The heartbeat functionality located in the Ironsphere Controller Administration Menu is used for monitoring the systems health checks.
  - o A user connected to the Ironsphere Controller can monitor the status of the services running in regional instances.
  - o From this menu it is possible to see progress logs, database (DB) replication status and latencies.

High Level Architecture



3. Functional Solution

Table 1: Role & Responsibilities Matrix of this layered Ironsphere Controller Solution

	Controller Layer	Region Layer
1 Configuration Management	Maker	Implementer
1.1 CRUD Users Management	Maker	Implementer
1.2 CRUD Device Management	Maker	Implementer
1.3 CRUD Policy/Realm Mgmt.	Maker	Implementer
1.4 CRUD Region/Instance Mgmt.	Maker	Implementer
1.5 CRUD License Management	Maker	Implementer
2 Log Management	See All Logs	See Only Local Logs
2.1 Log Creation		Maker
2.2 Log Auditing	See All Logs	See Only Local Logs
3 Password Management	Checker (see all activities)	Maker
3.1 Password Checkout	Checker (see all activities)	Maker
3.2 Password Randomization	Informed	Maker
4 Session Management	Checker (see all activities)	Maker
5 System Monitoring	See heartbeats Replication Status	See Local Services Status

4. Benefits of the Ironsphere Controller Solution

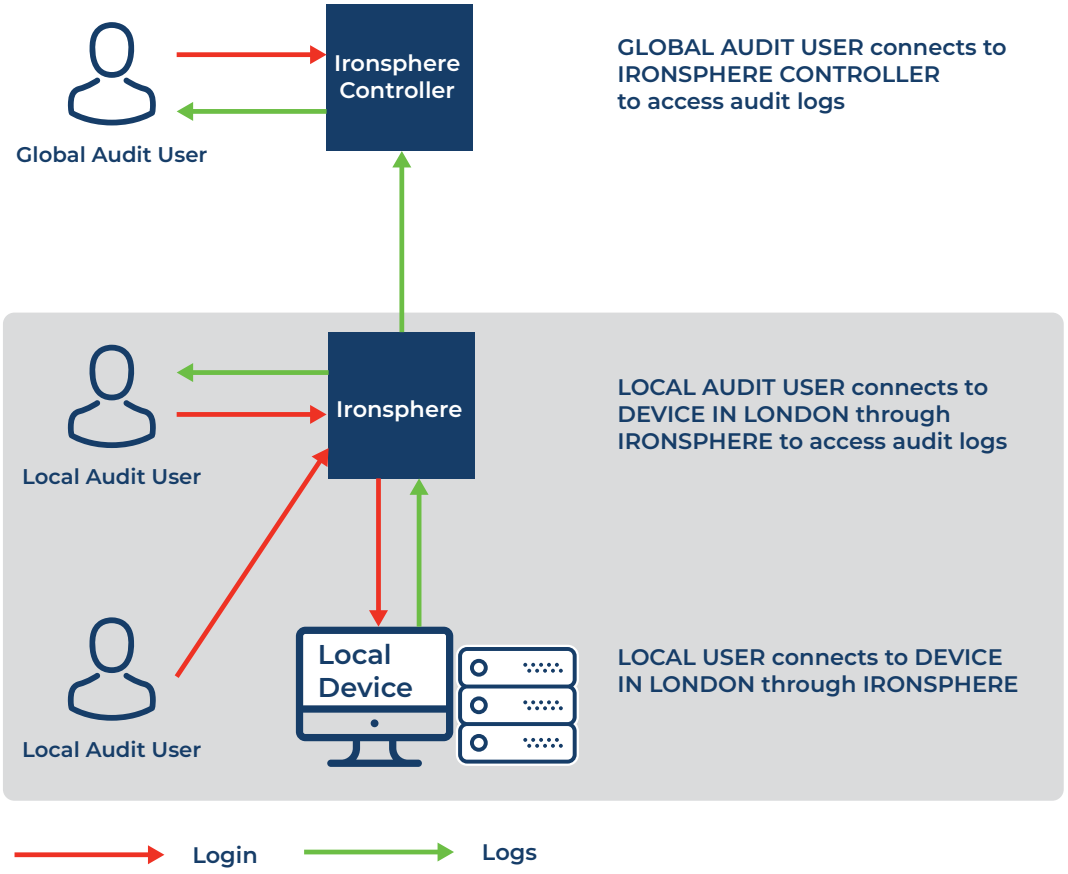
- Single Centralized Management (Fault, Configuration, Performance, Security)
- Federated Accounting Management
- Centralized Controller (Policy, Device, MFA Rights etc.)
- Consolidated and local Log Access
- Single Point of System Health Check
- Simple Management for Distributed Architecture on Different Time zones (On Premise, Cloud, Hybrid)
- Fastest Deployment (Single version of System Configurations & Parameters)
- Easy to Add/Drop the Regions

5. Basic Use Cases

5.1 Regional User Device Access and Log Auditing

Steps:

1. Local User logs in to Ironsphere.
2. Local User connects to Local Device via Ironsphere.
3. While the Local User connects to the Local Device via SSH session, Ironsphere generates session logs.
4. Ironsphere Controller pulls logs from Ironsphere.
5. Local Audit Users, with log access permissions, can inspect the local logs.
6. Global Audit Users access the Local User's session logs through the Ironsphere Controller.

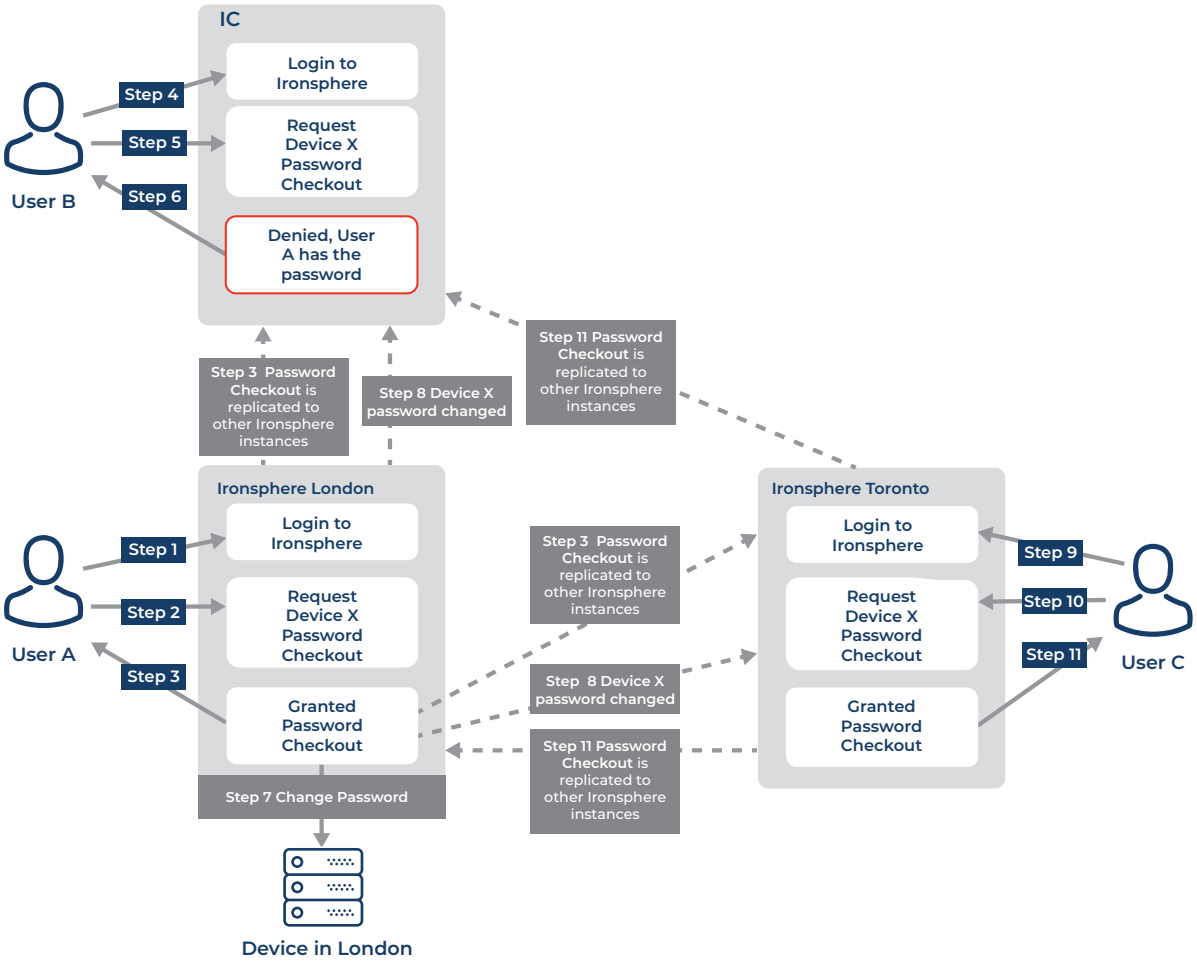


5.2 User Password Checkout

During User A's allotted time, User B wants to check out a password. Once User A completes their work within the specified time window, Ironsphere London changes the password on Device X. At a later time, User C wants to perform a password checkout for Device X on Ironsphere Toronto.

Steps:

1. User A logs in to Ironsphere London.
2. User A makes request for Device X password checkout, for a specific period of time.
3. Ironsphere grants the password to User A and replicates information to other Ironsphere Instances. User A can now connect to the device directly.
4. User B logs in to the Ironsphere Controller
5. User B requests Device X password checkout.
6. The Ironsphere Controller returns a "No" answer to User B's request, because User A has the password.
7. User A completes their work within the specified time window and Ironsphere London changes the password on Device X.
8. Ironsphere London replicates Device X password change to other Ironsphere Instances.
9. User C logs in to Ironsphere London.
10. User C requests a password checkout for Device X (which was previously checked out by User A).
11. Ironsphere London grants User C the new password for that account, as the specified time for User A has ended and Ironsphere subsequently randomized the password.



5.3 Changes on device/user/policy realm configurations on Ironsphere Controller (IC)

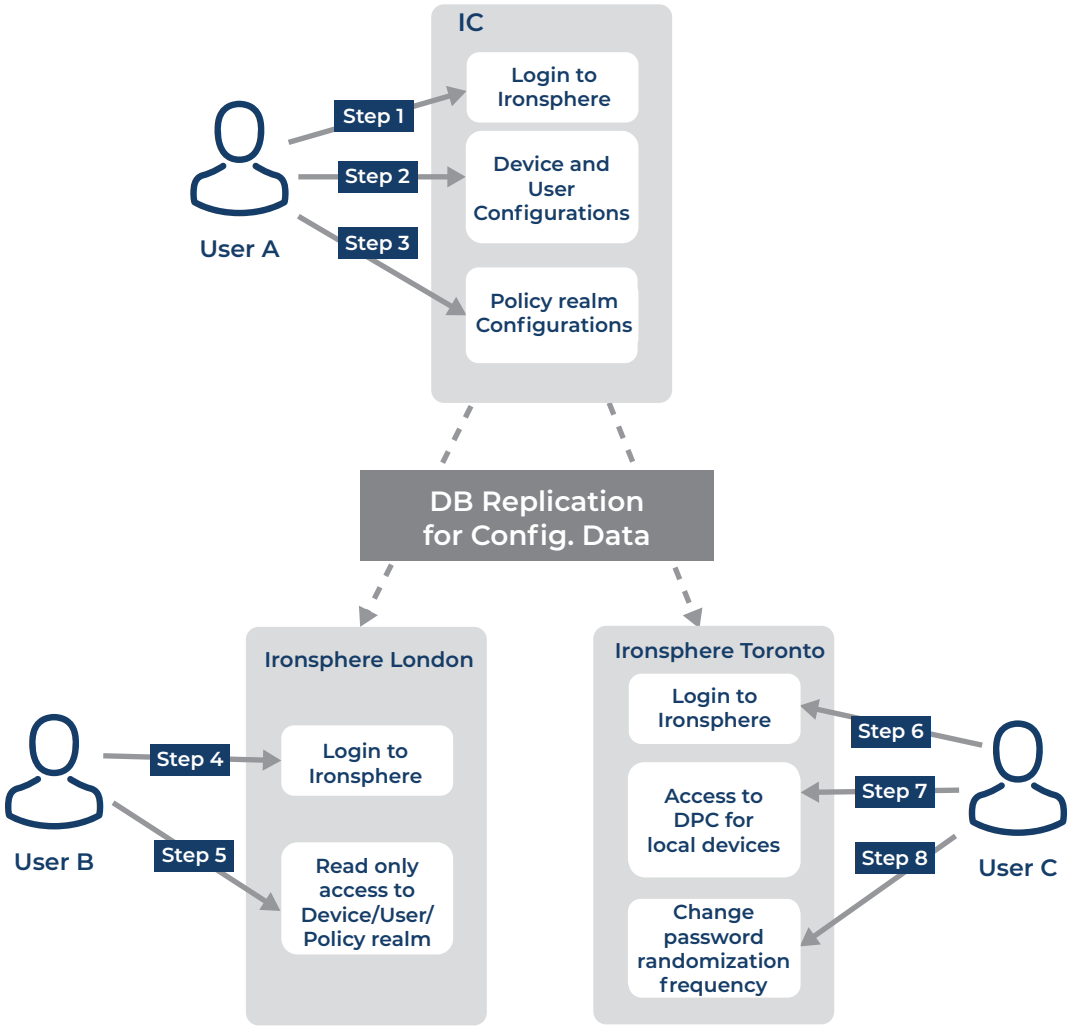
User A has the required permissions to make changes to the device/user/policy realm configurations on IC. The updated data will be pushed through the Ironsphere instances via online replication.

User B has read-only access to the configurations screen (device/user/policy realm).

User C has change access to the local Dynamic Password Controller (DPC) module. User C can change the password randomization frequency on the local DPC for local devices.

Steps

1. User A logs in to the Ironsphere Controller (IC).
2. User A changes the device/user configurations on the IC.
3. User A changes the policy realm configurations on the IC.
4. User B logs in to Ironsphere London.
5. User B sees the updated policies User A has executed and tries to change the policies, which is denied as he/she only has read-only access.
6. User C logs in to Ironsphere Toronto.
7. User C accesses the local DPC module.
8. User C changes the password randomization frequency on the DPC module.



5.4 Password Randomization

The local Ironsphere instances' DPC module is responsible for the local devices' passwords changes. The local Ironsphere instances will replicate the changed passwords information to the other Ironsphere instances and the Ironsphere Controller (IC).

User A only has read-only access to the DPC module configurations on the IC.

Steps of Use Case

1. Ironsphere London randomizes the passwords for its own devices.
2. Ironsphere Toronto randomized the passwords for its own devices.
3. The Ironsphere instances replicate those changes with each other and the IC.
4. User A logs in to the Ironsphere Controller (IC)
5. User A has read-only access to DPC module in the IC