Intro

SoDA data management software can talk to on-prem and cloud storage. SoDA has two key components in the system: the Conductor and the Agent. The Conductor can be deployed on-prem, in the Customers Cloud, or hosted by CloudSoDA for an additional fee. The Conductor hosts the User Interface (UI), controls the data movement done by agents, and is responsible for all the data tracking and reporting. The SoDA Agent will typically be deployed on a server or VM attached to the file-based storage, but if talking to the Cloud, will need access to the Cloud Storage as well. The SoDA Conductor does include an Internal Agent. Because SoDA Agents interface with storage on-prem and in the Cloud, there are security implications when dealing with data. SoDA is designed with security in mind.

This document explains the data integrity, data transfer, software security, and firewall settings for the SoDA application.

Data Integrity

SoDA moves and copies data. This can be file-to-file or file-to-object. When data moves, it is critical to ensure that data integrity is maintained. SoDA handles this in two distinct ways. When a file-to-file transfer is initiated, a MD5 checksum[(1)] is generated on the source file. The file is then transferred to a temporary file on the target system to avoid unintentionally overwriting a file that might already be there. A MD5 checksum is generated on the target temp file and the checksums are compared. If they match, SoDA renames and updates its attributes on the target file, thus ensuring the transfer is completed and there is no data corruption. If the checksums do not match, the operation is re-attempted up to three times before the file operation is marked as failed. When a file is transferred to an S3 object, the same process is attempted, but the validation is based on the S3 ETag[(2)]mechanism to validate that the operation is successful. AWS or S3 based object storage generates the ETag when the file is uploaded and completed, and SoDA validates the resulting object ETag by calculating it using the source data and the number of parts used in the upload. If they do not match or the upload fails, the transfer is re-attempted up to 40 times with an exponential backoff before the operation is marked as failed. When an object is downloaded, the object's ETag is compared to the local temp files MD5 before SoDA finishes the object to file transfer. This guarantees the files and objects are the exact same and there is no data corruption on transfers.

The validation process for Google Cloud Storage is identical to S3, but it uses a CRC32C hash rather than an ETag. Azure Blob is different, as it does not provide a hash unless it is a single part upload. Therefore, for Azure Blob, SoDA validates the upload by ensuring the correct number of bytes are written to the blob container. At the end of the upload, SoDA sets the MD5 of the object as metadata so it is available for future use.

Data Movement

The SoDA Conductor orchestrates the data movement while the Agent performs the data movement. The Agent can be internal or external to the Conductor. The Internal Agent supports mounting NFS/SMB shares directly. While it is possible for an external agent to mount NFS/SMB storage, it is not supported because it can result in unexpected behavior. Therefore, CloudSoDA recommends that if an external agent needs to access data over NFS/SMB that the end-user directly mounts the storage on the host.

There is no encryption when moving data via the SMB/NFS protocols, as they do not support it by default. SoDA uses the cloud vendor's provided SDK for cloud or object transfers for example AWS uses the SDK for go[(3)]. For the AWS andAzure SDK transfers the data using TLS v1.2[(4)(5)(6)]over the WAN to ensure data is encrypted in flight. The Google SDK uses TLS 1.3. to transfer data into its object storage.[(7)(8)(9)] All data sent agent to agent is encrypted and uses TLS 1.3.

Software Access and Firewalls

SoDA Conductor can be deployed on-prem, in a customer cloud, or cloud-hosted by SoDA. Fig 1 refers to the ports and software packages SoDA uses when talking to file-based storage, Cloud Providers Storage, SoDA Agents, and our managed software control plane.

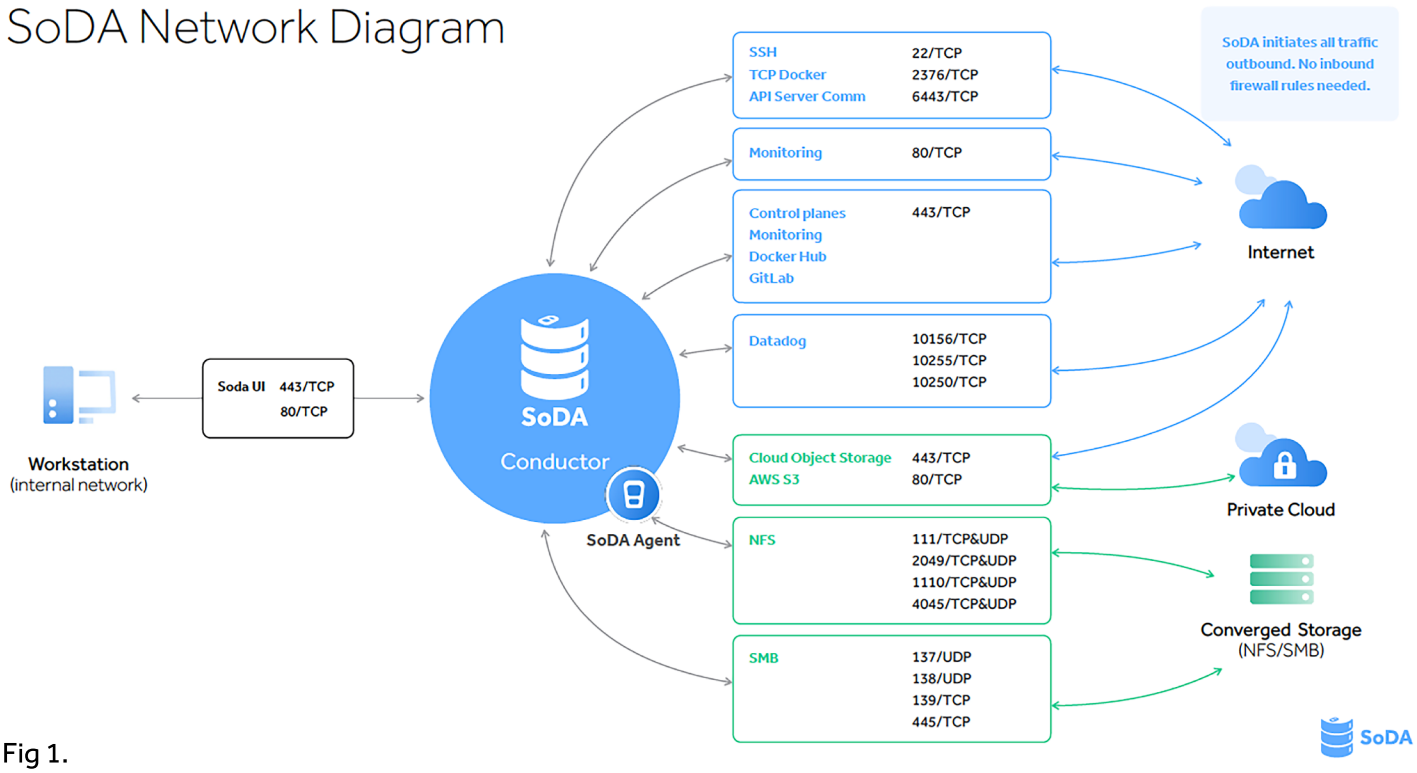

Fig 1.

The SoDA Conductor leverages ports and connections for 4 main tasks:

- Administration: Deploy, upgrade, and monitor the SoDA Conductor and Agents
- Data Movement: File/Object Base Data movement via Agent or SMB/NFS on the Conductor
- SoDA Communication: SoDA Conductor to Agent communication
- Web UI: Administration and monitoring of SoDA

These four main tasks are detailed in their corresponding subsections below.

Administration:

Administration offers many advantages for customers including installation, real-time software patching and updates, and simplified troubleshooting. Collecting information regarding the health of SoDA software and hardware is also required. Absolutely no customer data is sent in the monitoring the health of the SoDA software. Port 80 is used many times for a handshake and then upgraded to SSL (443), to negotiate the secure connection.

Data Movement:

If you use the SoDA Conductor's Internal Agent for NFS/SMB, the SoDA Conductor needs access to all the ports indicated on the diagram to perform file operations.

For Cloud Storage access, SoDA does NOT need any firewall exceptions. SoDA uses ports 80/443 to create a connection to Cloud Storage Targets. Once the connection is made, all data to Cloud Storage Targets is sent encrypted per the above section.encrypted per the above section.

SoDA Communication:

The SoDA Conductor uses agents to move or copy data. The Agent must be able to communicate with the Conductor, which requires outbound UDP Ports 7498/7499 from the Agent. This means the Conductor needs to be able to accept UDP requests on ports 7498/7499, and the Agent needs to be able to reach out to the Conductor on the same ports. If either of the ports is blocked, the Agent will not be able to connect to the Conductor. All data sent between the Conductor and the Agent is encrypted using TLS 1.3. See Fig 2.

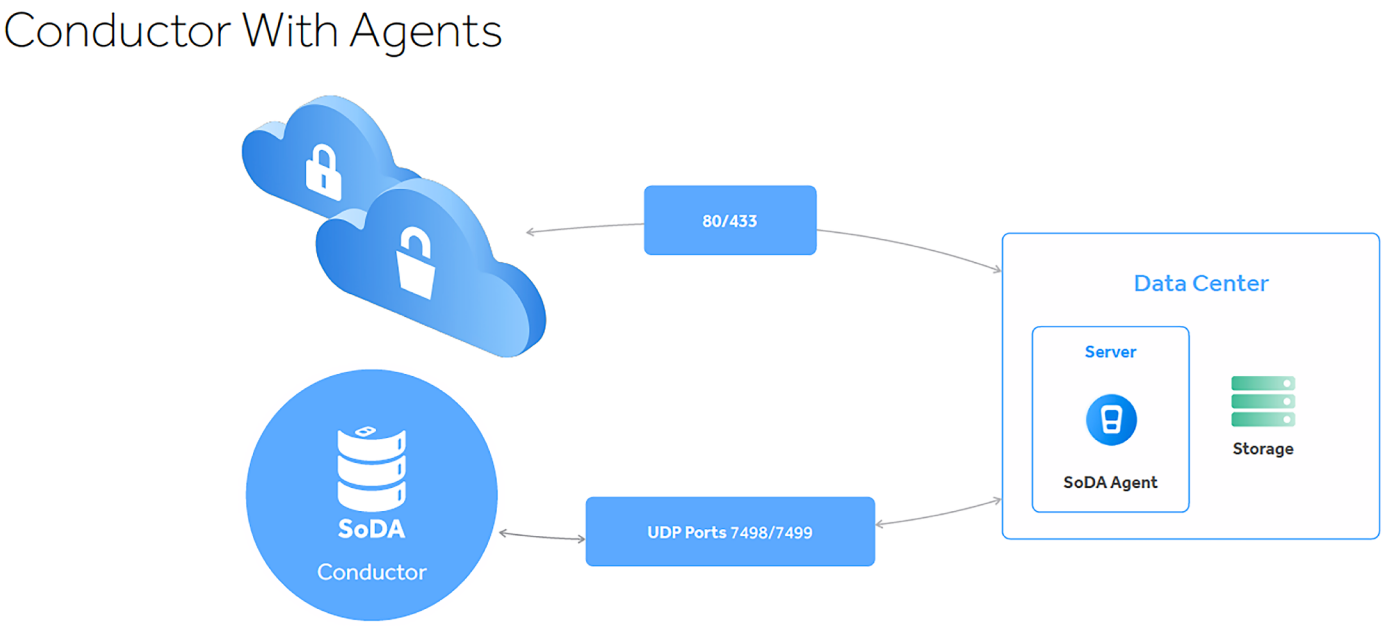

Fig 2.

Web UI:

To use SoDA's web UI, Port 443 must be open to access the SoDA Software. SoDA can be accessed via port 80, but it is upgraded to SSL (443) for the connection.

SoDA Agent

The Agent can be installed on Linux, Windows, or macOS operating systems. The Agent has the potential to access any files that the host operating system can access, including local volumes, Direct Attached Storage, or SMB/NFS mounted storage. For public and private clouds, the Agent can access object storage including S3, GCP Cloud Storage, Azure Blob, and other S3 link storage.

SoDA agent to agent transfer is initiated when the source and target storage are on disparate agents. To facilitate this agent-to-agent transfer, an Agent Mesh is created in which all the Agents attempt to connect with each other. The Agents need the following UDP ports (1024-65535) open (ingress & egress) in order to connect. The Agents establish the mesh by using every network interface possible to connect, including LAN routes, VPN tunnels, and over the WAN using the SoDA control plane. For an Agent to leverage the SoDA control plane, it must be able to access https://controlplane.sna.cloudsoda.io/. Without access to the SoDA control plane, transfers through a NAT or bridging secured private networks are not possible. See Fig 3.

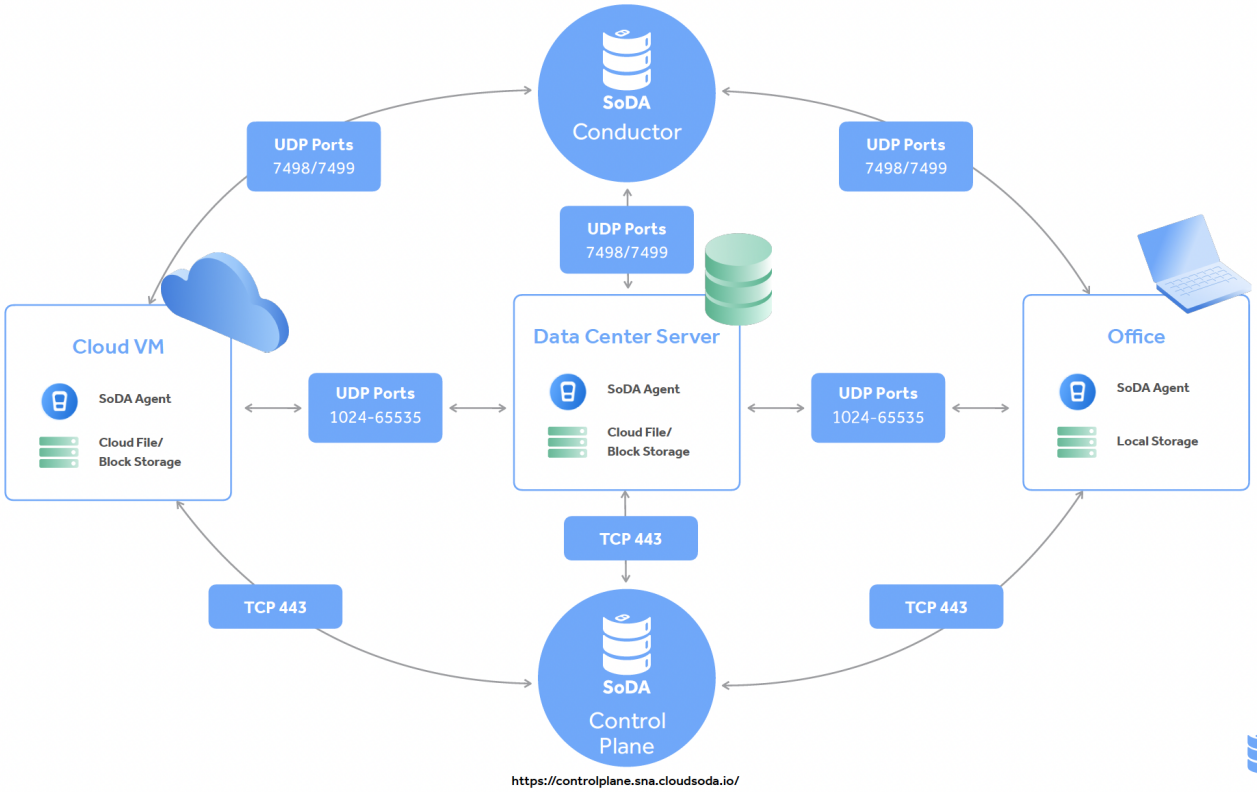

Fig 3.

NOTE: The SoDA Conductor's Internal Agent cannot participate in the agent mesh and therefore, transferring data to another agent is not supported.

References

(1)https://en.wikipedia.org/wiki/MD5

(2)https://aws.amazon.com/premiumsupport/knowledge-center/data-integrity-s3/

(3)https://aws.amazon.com/sdk-for-go/

(4)https://docs.aws.amazon.com/sdk-for-go/v1/developer-guide/security.html

(5)https://github.com/Azure/azure-storage-blob-go

(6)https://docs.microsoft.com/en-us/azure/storage/common/transport-layer-security-configure-client-version?tabs=powershell

(7)https://pkg.go.dev/cloud.google.com/go/storage

(8)https://cloud.google.com/storage/docs/gsutil/addlhelp/SecurityandPrivacyConsiderations#transport-layer-security

(9)https://cloud.google.com/blog/products/networking/tls-1-3-is-now-on-by-default-for-google-cloud-services

(10)https://docs.datadoghq.com/agent/guide/network/?tab=agentv6v7

Appendix A

Ports: TCP 80, 443 for container pulls
gitlab:
registry.gitlab.com
docker hub:
auth.docker.io
index.dockerhub.io
dockerhub.io
quay.io
cdn.quay.io
index.docker.io

Ports: TCP 80,443 for control plane
Rancher:
rnch-prd-usw2-1.cloudsoda.io

Ports: TCP 80,443 for monitoring
Rollbar:
35.184.69.251
35.201.93.97
35.201.81.77

Ports: TCP 443, 10516, 10255, 10250, UDP 123 for monitoring Datadog:
trace.agent.datadoghq.com
process.datadoghq.com
agent-intake.logs.datadoghq.com
agent-http-intake.logs.datadoghq.com
orchestrator.datadoghq.com
app.datadoghq.com
*.agent.datadoghq.com [10]