Making the Journey to FedRAMP

Cisco Umbrella, Altinity, and ClickHouse-based Analytics

Pauline Yeung
Robert Hodges

Altinity

cisco SECURE

Let’s make some introductions

Robert Hodges

Database geek with 30+ years on DBMS systems. Day job: CEO at Altinity

Pauline Yeung

Software Engineer at Cisco Data Sec Dev Ops

What is FedRAMP?

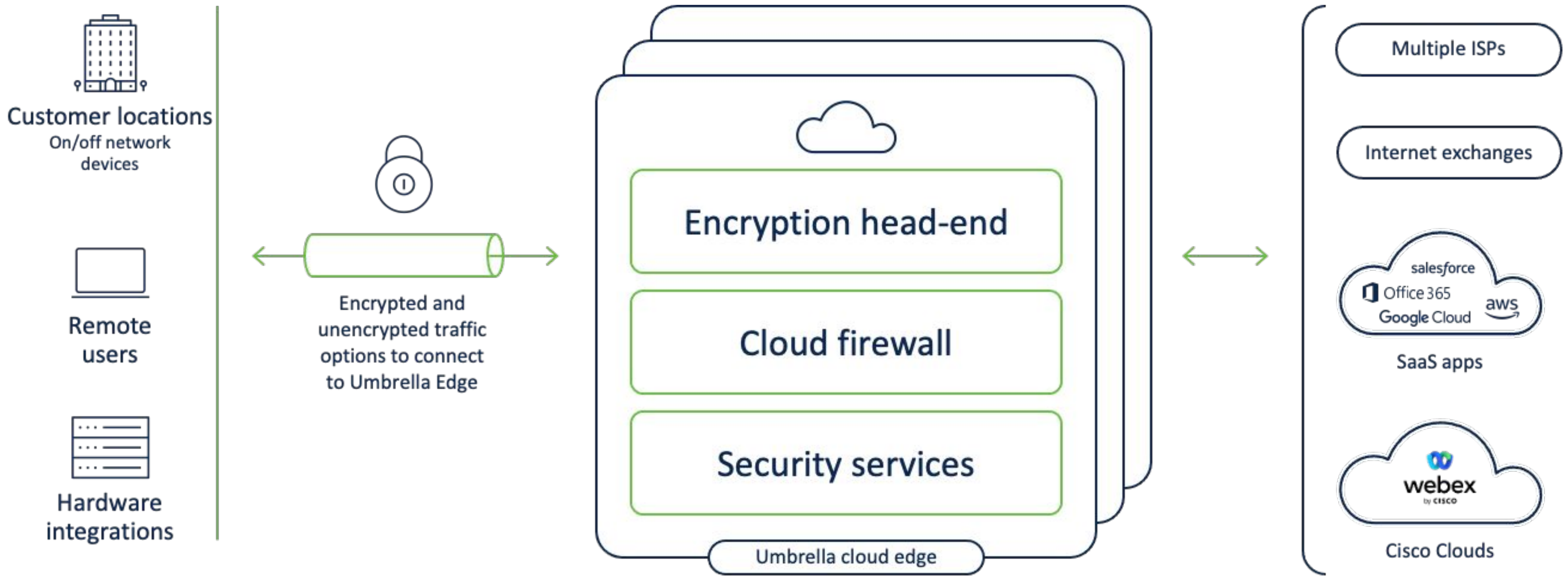
- A security compliance program for United States government systems
- Is a standard requirement for doing business with the US government
- There are multiple levels of compliance
    - High
    - Moderate
    - Low
- FedRAMP-compliant systems commonly run in GovCloud or similar secure cloud environments

What is Cisco Umbrella and How Does it Use ClickHouse?

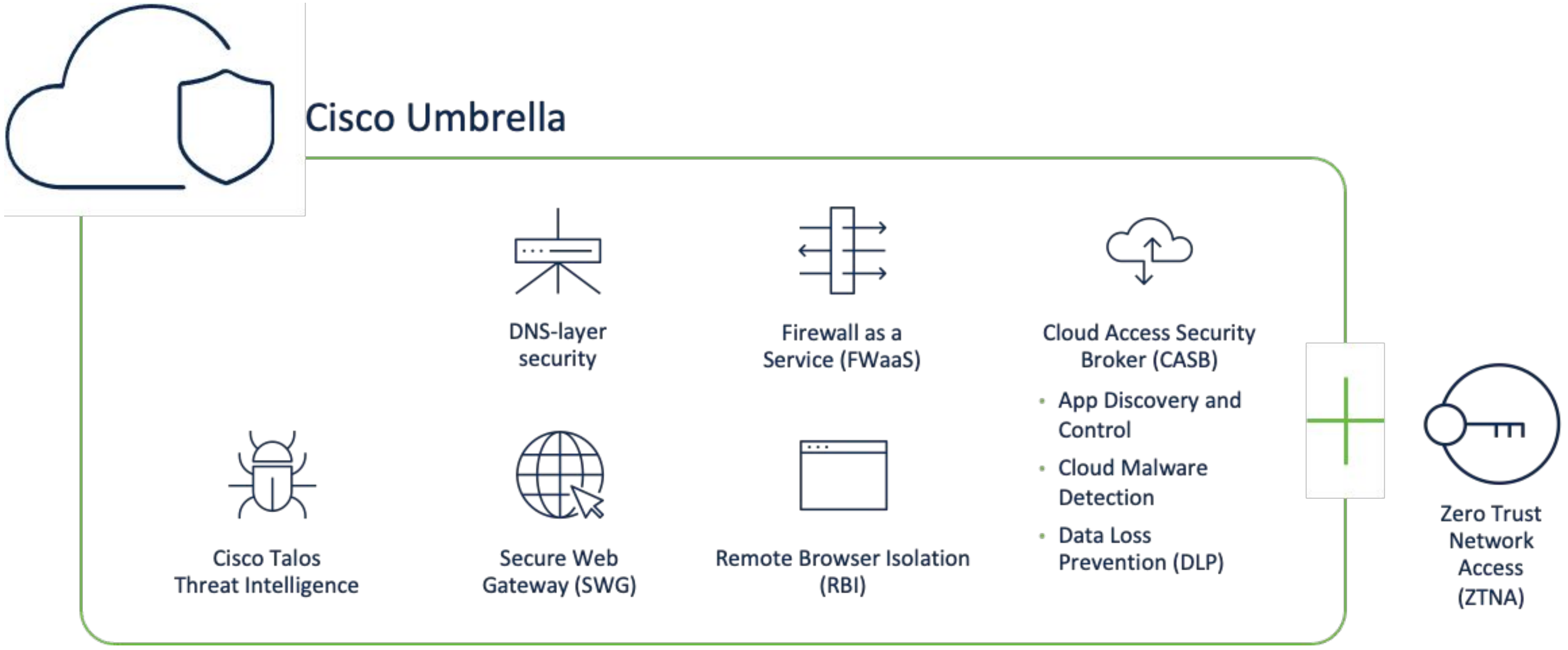
- Secure Services in Cisco Umbrella log activities to Clickhouse
- Logs are used for
  - Security Report
  - Activity Search
  - Threat Intelligence

Altinity

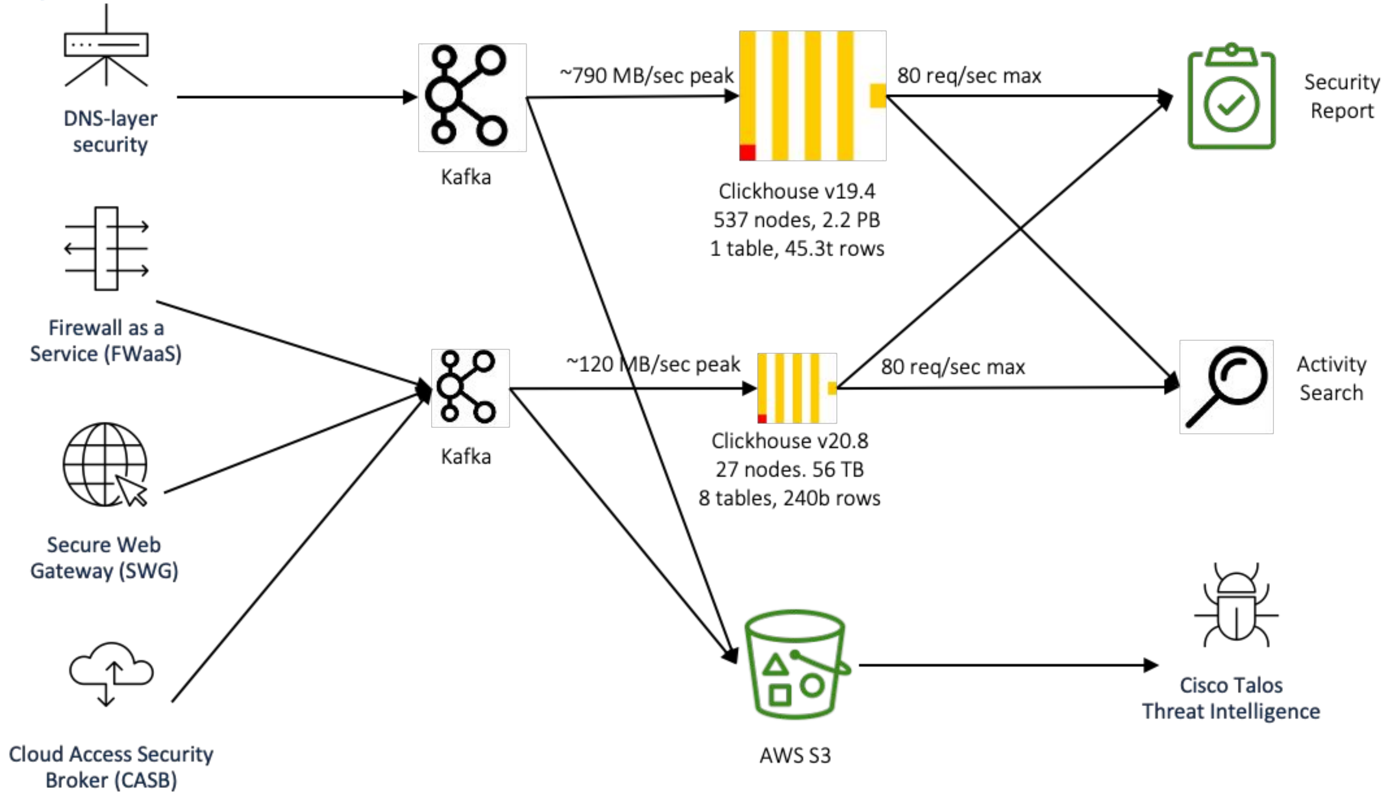
Cisco Umbrella Cloud Architecture



Cisco Umbrella Secure Service Edge (SSE)


Cisco Umbrella
DNS-layer security
Firewall as a Service (FWaaS)
Cloud Access Security Broker (CASB)
App Discovery and Control
Cloud Malware Detection
Data Loss Prevention (DLP)
Cisco Talos Threat Intelligence
Secure Web Gateway (SWG)
Remote Browser Isolation (RBI)
Zero Trust Network Access (ZTNA)

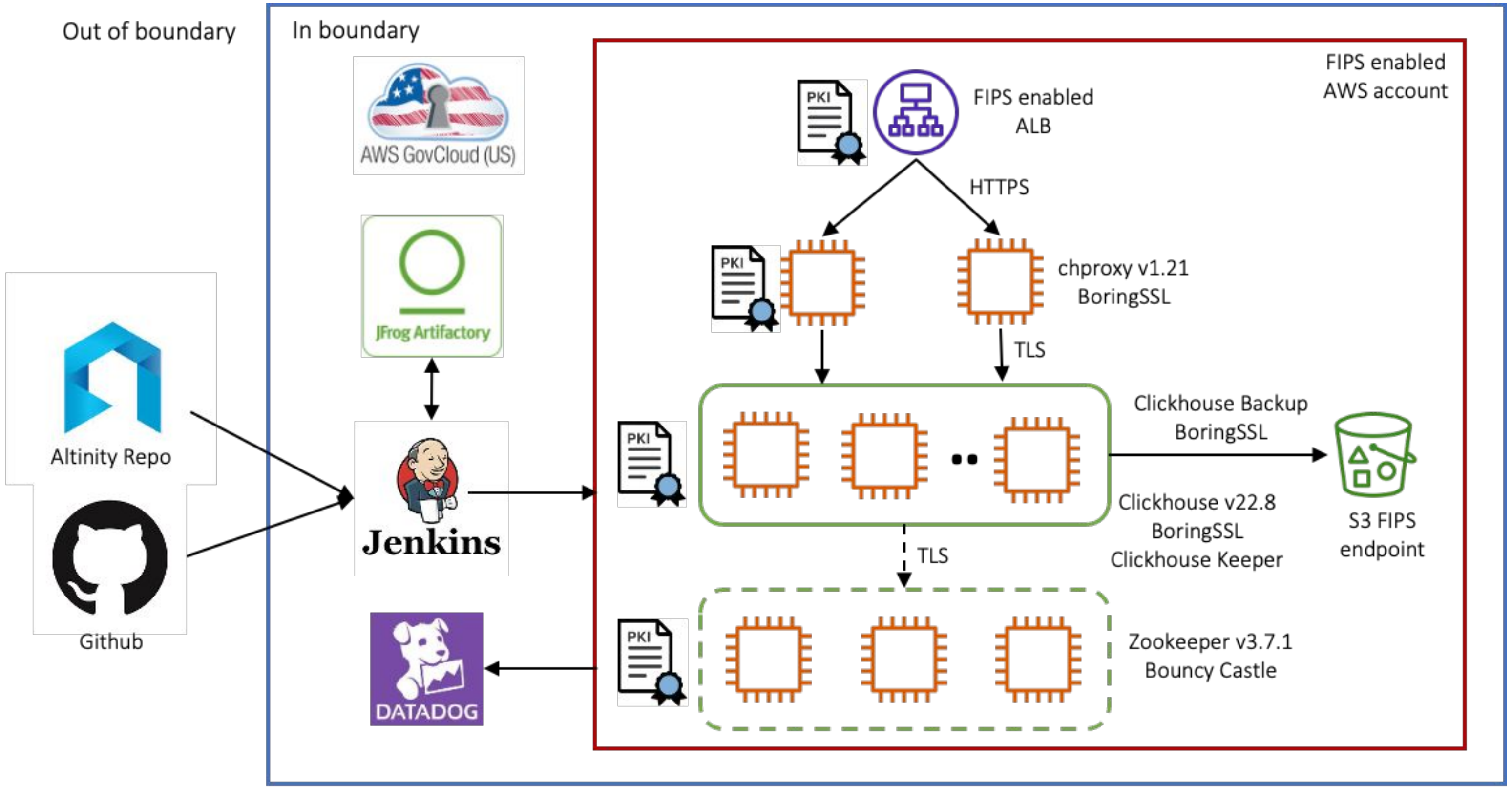
SSE logs to Clickhouse



What is FIPS 140-2?

- "FIPS" = "Federal Information Processing Standards"
- FIPS 140-2 = Standard for cryptography in US government systems
  - And also several other countries like Canada and Japan
- FIPS certification is a long process
  - Has to be repeated when you change the software
- Most applications try to be FIPS-compatible instead
  - Do all the steps to prepare for certification but don't certify

FedRAMP Clickhouse Cluster



FIPS Compliance

- Govcloud AWS account default to FIPS disabled, has to request AWS Support to enable FIPS.
- Create ALB with tag “alb-fips-enabled”.
- chproxy is compiled with BoringSSL for Go.
- Clickhouse is compiled with BoringSSL.
- Clickhouse backup is complied with BoringSSL.
- Zookeeper loads Bouncy Castle jar.
- Point to S3 FIPS endpoint by setting AWS_USE_FIPS_ENDPOINT = true.

Introducing FIPS-compatible ClickHouse

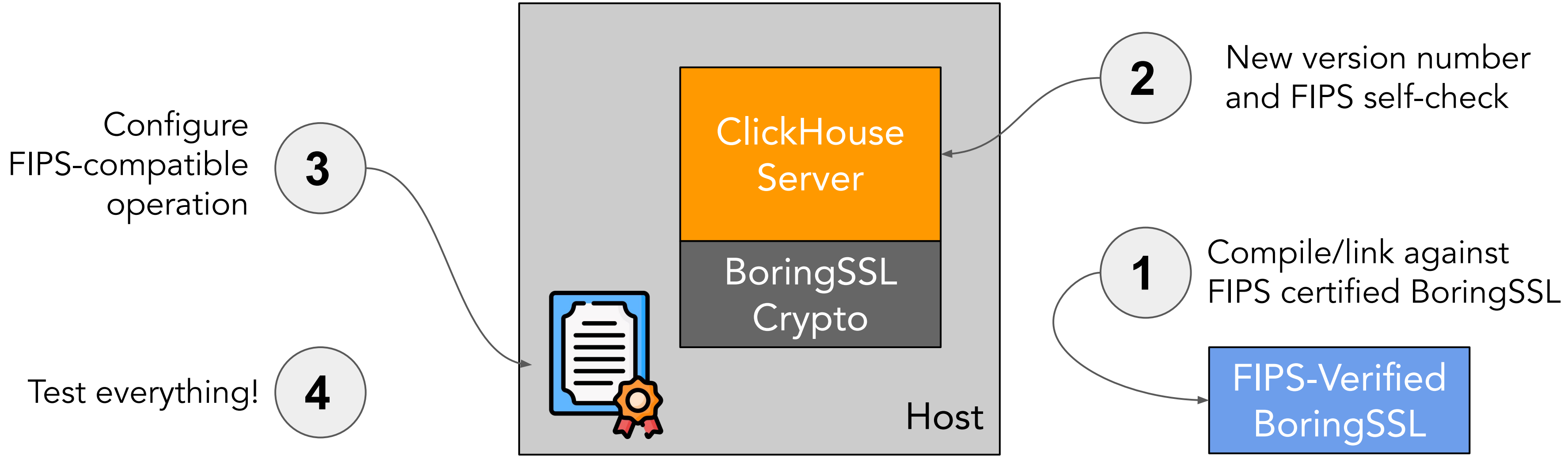
What are Altinity Stable Builds for ClickHouse?

Altinity Stable Builds are open source builds of ClickHouse for enterprise users

- Based on Long-Term Support releases of ClickHouse
- Plus selected bug fixes and features (including patches for CVEs)
- Vetted thoroughly for production use
- Supported for three years (rather than one)
- 100% open source
- Strive for full compatibility with upstream ClickHouse

https://docs.altinity.com/altinitystablebuilds/

How to make apps like ClickHouse "FIPS-compatible?"


Configure FIPS-compatible operation
3
Test everything!
4
ClickHouse Server
BoringSSL Crypto
Host
2
New version number and FIPS self-check
1
Compile/link against FIPS certified BoringSSL
FIPS-Verified BoringSSL

Introducing FIPS-compatible Altinity Stable builds

Altinity Stable Builds include FIPS-compatible versions starting from version 22.8.

- Managed in a separate branch
- Identical to mainline ClickHouse except for FIPS features
  - Self-check and software version
  - Extensions for ClickHouse Keeper crypto
- Use BoringSSL source code that passed certification on June 29, 2022
- Use same procedure for building
  - BoringCrypto FIPS 140-2 Non-Proprietary Security Policy
- Crypto behavior verified using Altinity test suite

How are FIPS-compatible Altinity Stable Builds tested?

Altinity Stable Builds must pass a large suite of tests including:

- ClickHouse unit and integration tests (in Altinity ClickHouse repo)
- Altinity regression tests (in Altinity clickhouse-regression repo)
- Code scans on containers (Snyk & Scout)

Crypto behavior is complex and "delicate"

- Altinity ssl_server test – Tests crypto between applications and ClickHouse
- Altinity ssl_keeper test – Tests crypto for ClickHouse clusters

How can you get FIPS-compatible Altinity Stable Builds?

Get pre-built binaries from Altinity Stable repo!

- https://builds.altinity.cloud/
  - (Altinity Stable FIPS-Compatible Build section)

Build it yourself!

- Checkout Altinity's FIPS compatible ClickHouse version
  - https://github.com/altinity/clickHouse/tree/releases/22.8.15-fips
- Build it as usual (cmake .. && cmake --build . --target all)
  - There is a `FIPS_CLICKHOUSE` configuration parameter but in that version it is set to ON, so no changes to build process are required.

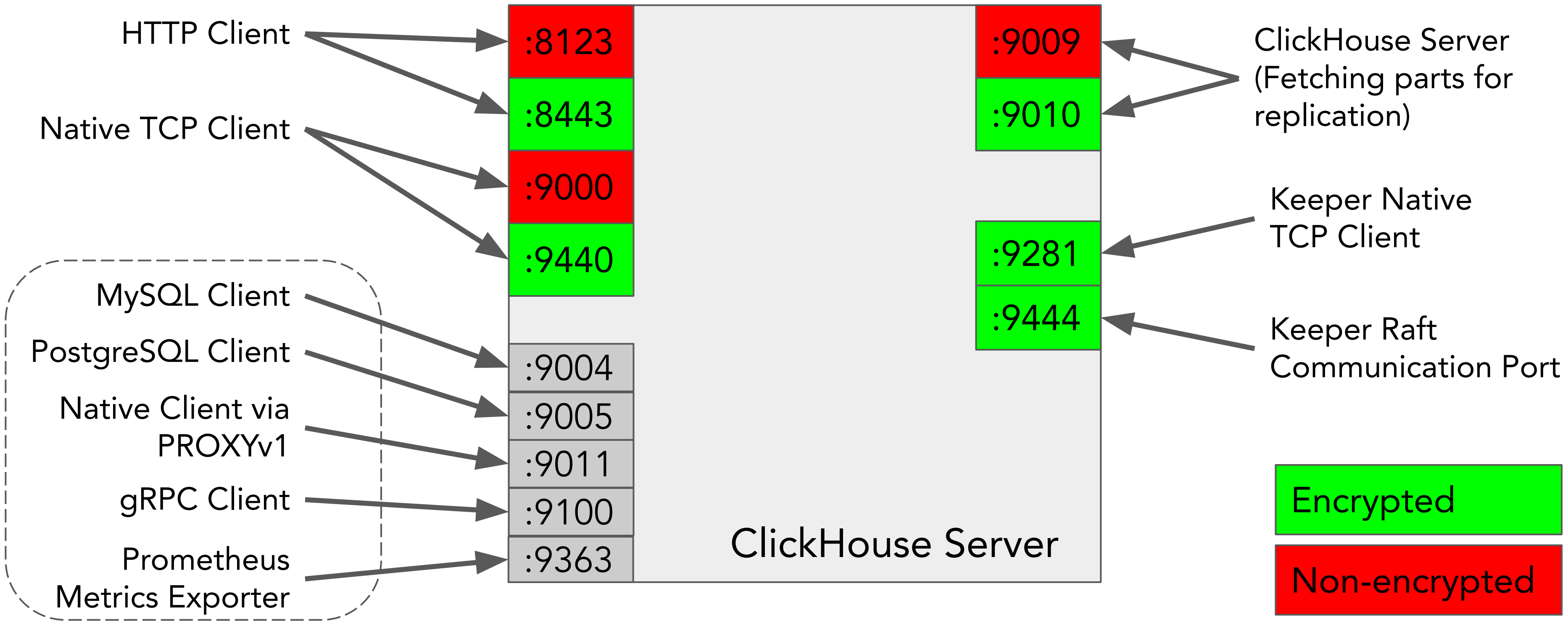
How do you configure FIPS-compatible operation?

Follow the docs for FIPS-compatible Altinity Stable Builds

1. Shut off all non-FIPs ports.
2. Add fips.xml configuration file.
   a. Sets TLS version.
   b. Specifies allowed ciphers.
3. Start server and verify successful self-test on startup.

```
$ grep 'FIPS mode' /var/log/clickhouse-server/clickhouse-server.log
2023.05.28 18:19:03.064038 [ 1 ] {} <Information> Application:
Starting in FIPS mode, KAT test result: 1
```

Know your ClickHouse ports!

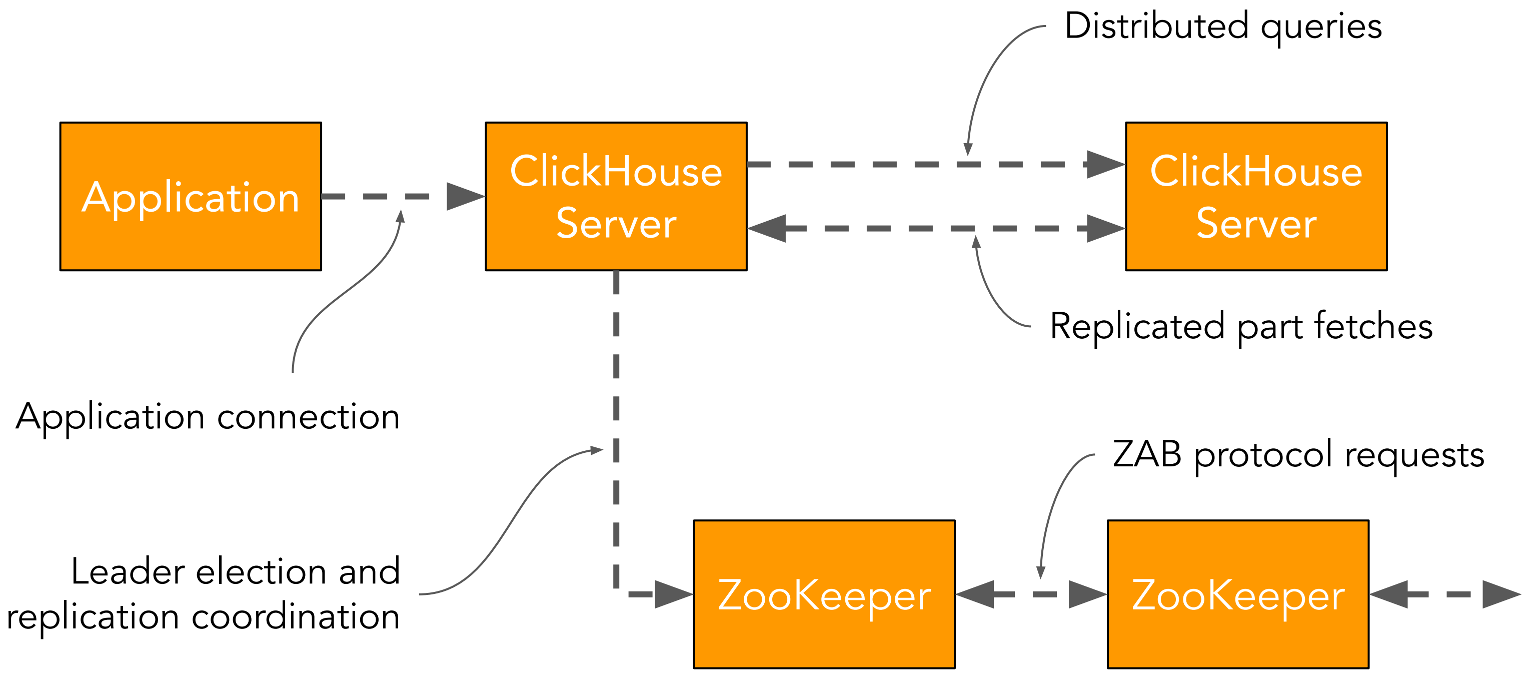


What does the fips.xml file look like?

```
<clickhouse>
  <http_port remove="true"/>
  <https_port>8443</https_port>
  <tcp_port remove="true"/>
  <tcp_port_secure>9440</tcp_port_secure>
  <openSSL>
    <server>
      <certificateFile>${CERT_PATH}/server.crt</certificateFile>
      <privateKeyFile>${CERT_PATH}/server.key</privateKeyFile>
      <dhParamsFile>${CERT_PATH}/dh_params.pem</dhParamsFile>
      <cipherList>ECDHE-RSA-AES128-GCM-SHA256:...:AES256-GCM-SHA384</cipherList>
      <loadDefaultCAFile>true</loadDefaultCAFile>
      <cacheSessions>true</cacheSessions>
      <preferServerCiphers>true</preferServerCiphers>
      <requireTLSv1_2>true</requireTLSv1_2>
      <disableProtocols>sslv2,sslv3,tlsv1,tlsv1_1,tlsv1_3</disableProtocols>
      <verificationMode>relaxed</verificationMode>
    </server>
```

Enabling FIPS-compatible ClickHouse clusters

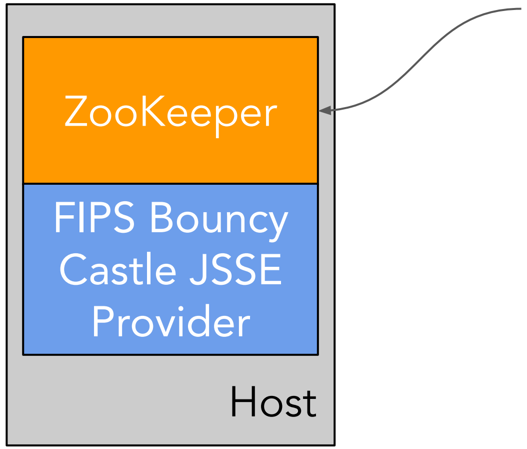
ClickHouse clusters add complexity and attack surfaces



ClickHouse clusters depend on centralized coordination

Should we use ZooKeeper or ClickHouse Keeper?

Pros and cons of making ZooKeeper FIPS-compatible



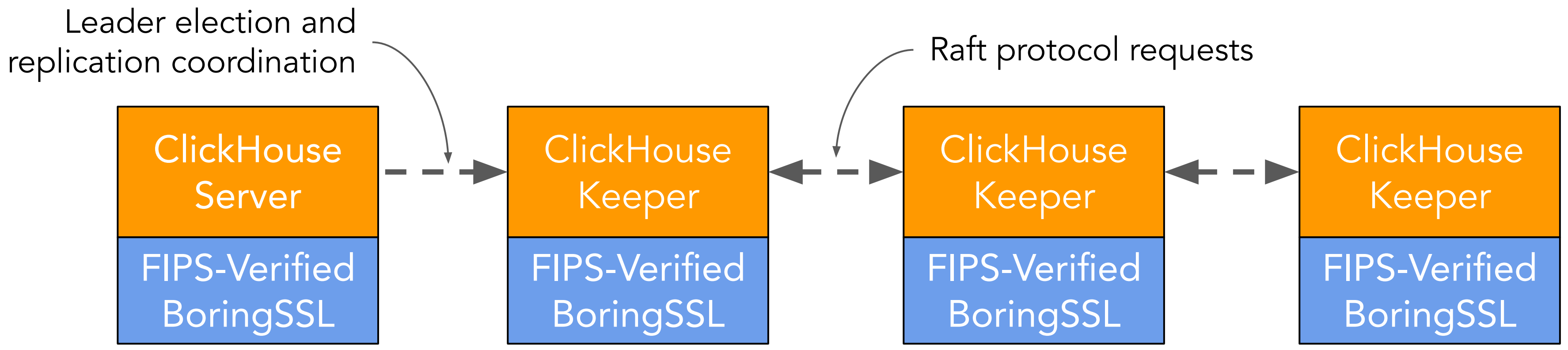
Build ZooKeeper with FIPS-compatible Bouncy Castle libraries

✔ It works!

❌ Hard to maintain
❌ ClickHouse Keeper is the future

https://www.bouncycastle.org/fips-java/

We decided to switch to ClickHouse Keeper



Key changes:

- Update NuRaft library to use same SSL context as ClickHouse
- Test it very carefully!

FIPS-compatible Keeper is on the way

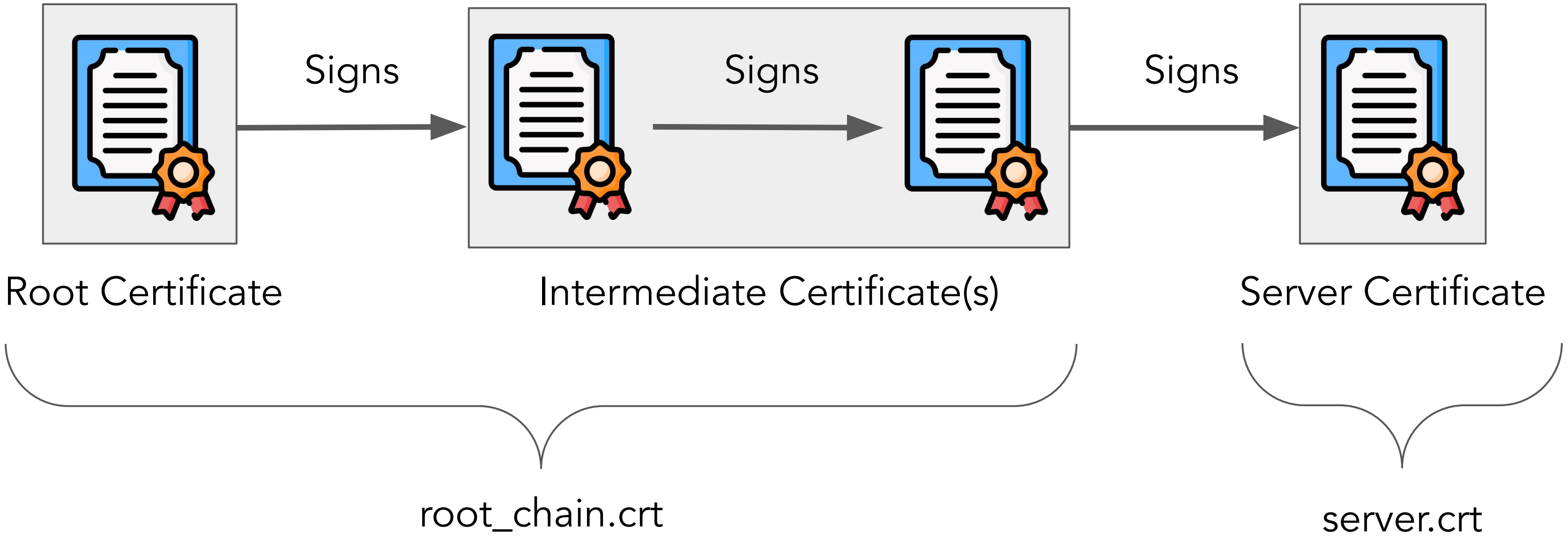
Altinity will support ClickHouse Keeper as part of FIPS-compatible Altinity Stable 23.3.

Release date: Late June 2023

Additional FedRAMP Challenges

- Cisco FedRAMP certificates
- Fully hardening ClickHouse

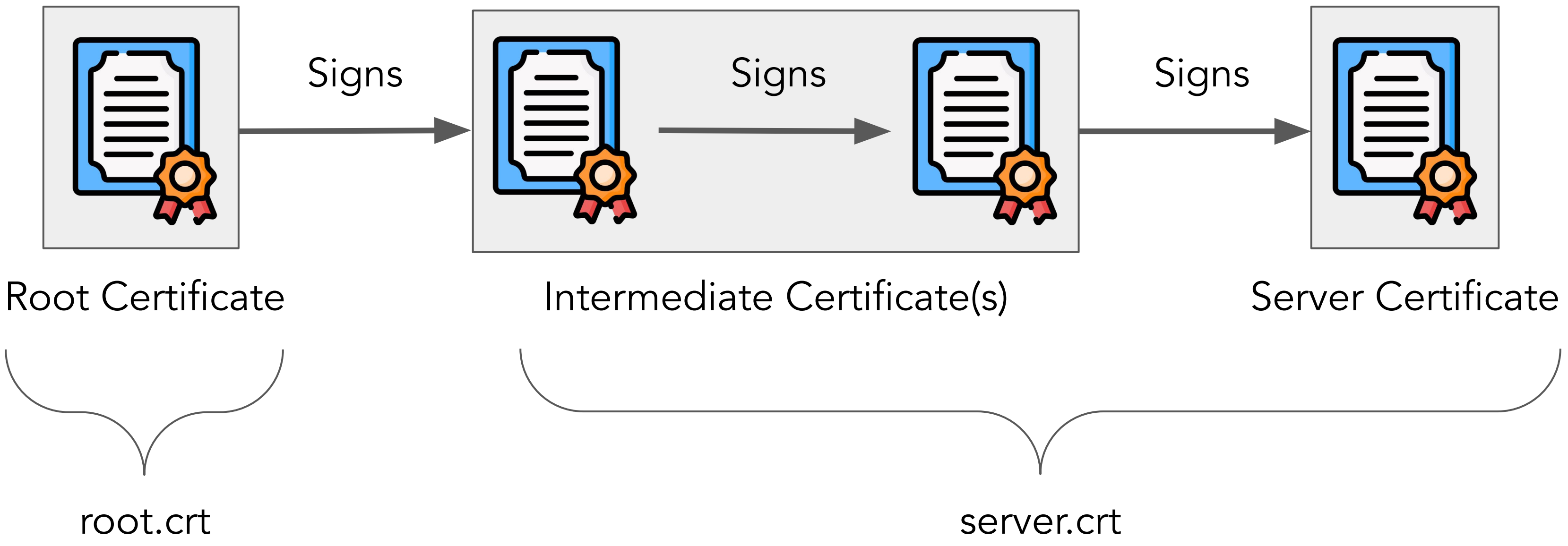
Certificate chains require special treatment in ClickHouse



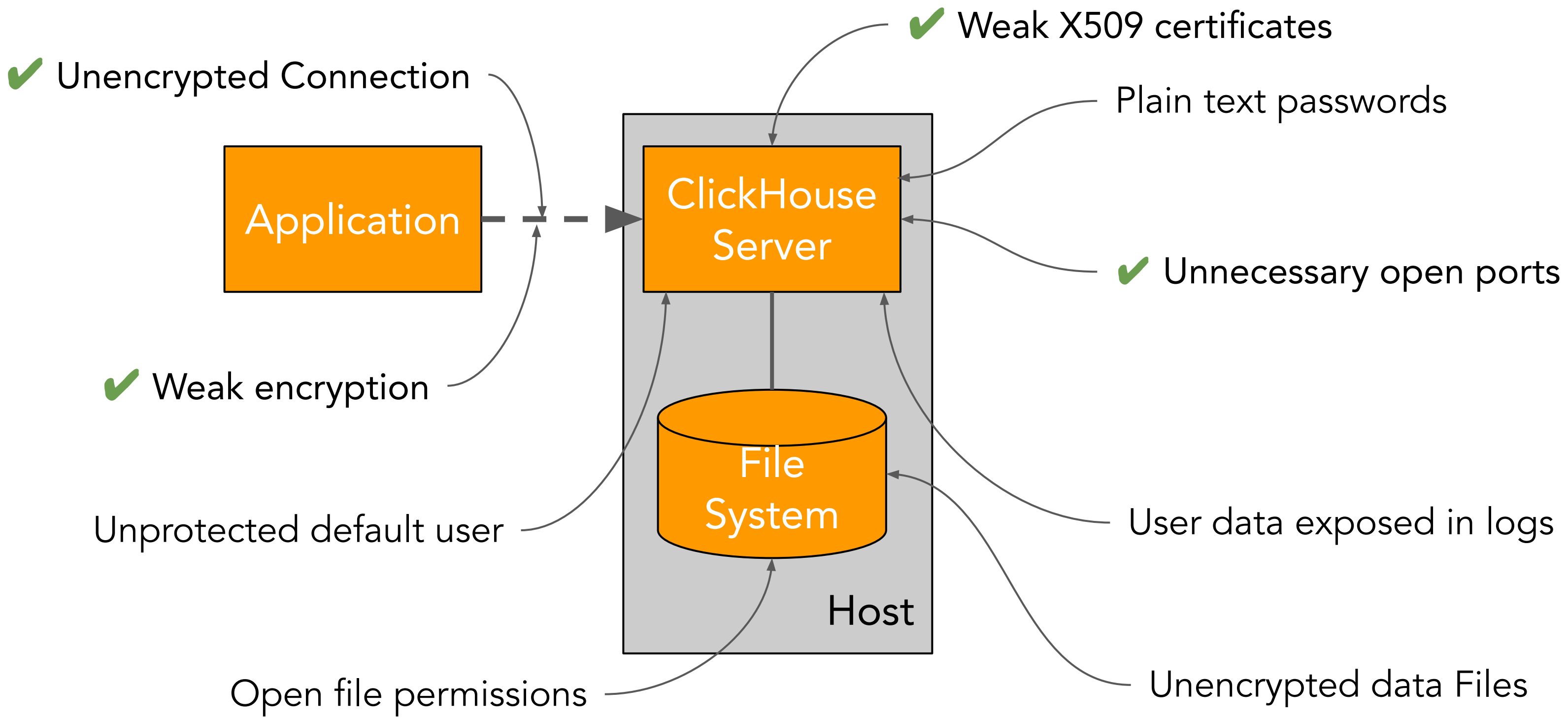
What does the fips.xml file look like?

```
<clickhouse>
  <https_port>8443</https_port>
  <tcp_port_secure>9440</tcp_port_secure>
  <openSSL>
    <server>
      <caConfig>${CERT_PATH}/root_chain.crt</caConfig>
      <certificateFile>${CERT_PATH}/server.crt</certificateFile>
      <privateKeyFile>${CERT_PATH}/server.key</privateKeyFile>
      <dhParamsFile>${CERT_PATH}/dh_params.pem</dhParamsFile>
      <cipherList>ECDHE-RSA-AES128-GCM-SHA256:...:AES256-GCM-SHA384</cipherList>
      <loadDefaultCAFile>false</loadDefaultCAFile>
      <cacheSessions>true</cacheSessions>
      <preferServerCiphers>true</preferServerCiphers>
      <requireTLSv1_2>true</requireTLSv1_2>
      <disableProtocols>sslv2,sslv3,tlsv1,tlsv1_1,tlsv1_3</disableProtocols>
      <verificationMode>relaxed</verificationMode>
    </server>
    . . .
```

Some cases require downloading the intermediate certs!



Thinking holistically about ClickHouse hardening



Setting up an operational system

Cisco’s Deployment and Configuration

- Use terraform to deploy the clickhouse cluster in EC2 in a FIPS enabled ubuntu 20.04.
- Use 3 different ansible playbooks to configure zookeeper, then clickhouse, then chproxy.
- Ansible code to install clickhouse from Altinity FIPS repo.

```
- name: add altinity apt repo GPG key
  apt_key:
    data: "{{ lookup('file', 'altinity-apt-repo.asc') }}"
    state: present
```

Ansible to install clickhouse

```
- name: add clickhouse repo
  apt_repository:
    repo: "deb https://builds.altinity.cloud/fips-apt-repo stable main"
    state: present
    update_cache: yes

- name: install clickhouse packages
  apt:
    pkg:
    - clickhouse-common-static={{ clickhouse_version }}
    - clickhouse-client={{ clickhouse_version }}
    - clickhouse-server={{ clickhouse_version }}
    state: present
    update_cache: yes
    install_recommends: yes
```

Address Security

- Filters to redact clickhouse password when running Ansible from Jenkins.
- Only allow chproxy and other clickhouse nodes to access TLS port 9440 using AWS Security Group.
- Non TLS port 9000 can only be accessed locally, used by DataDog agent.
- Only error logs are sent to DataDog for centralized logging.
- Encryption at rest.

Conclusion

What have we learned about ClickHouse and FedRAMP

- Make everything FIPS compliant: Linux distro, S3 endpoint, applications, ClickHouse, …, the universe
- ZooKeeper is not a long-term solution for FIPS-compliant ClickHouse
- Ansible stands up hardened services consistently and quickly
- Test everything - crypto is complex and delicate
  - There are many more ways to configure incorrectly than correctly
- Documentation and configuration guidelines are essential to success

Background information

- Altinity security documentation – https://docs.altinity.com
  - FIPS-Compatible Altinity Stable Builds
- Altinity Blog – https://altinity.com/blog
- Altinity YouTube Channel
  - Fortress ClickHouse video

Thank you! Questions?

Which BoringSSL do we use?

- There are multiple FIPS-certified BoringSSL versions
- Most recent is certified on June 29, 2022
- ClickHouse uses a more recent version of BoringSSL
  - So we have to downgrade it.

What did we change in ClickHouse to make FIPS work?

FIPS-compatible Altinity Stable builds are identical to mainline ClickHouse except:

- Build system changes (described above) + some extra to allow Docker-in-Docker for CI/CD
- BoringSSL API changes (pretty trivial ones)
  - For example different header file locations & const values
- More logging (to highlight that CH is starting in FIPS mode)
- Includes changes to NuRaft implementation (link to PR to NuRaft)
- Update to the ClickHouse Keeper to support full range of SSL configuration options available to ClickHouse

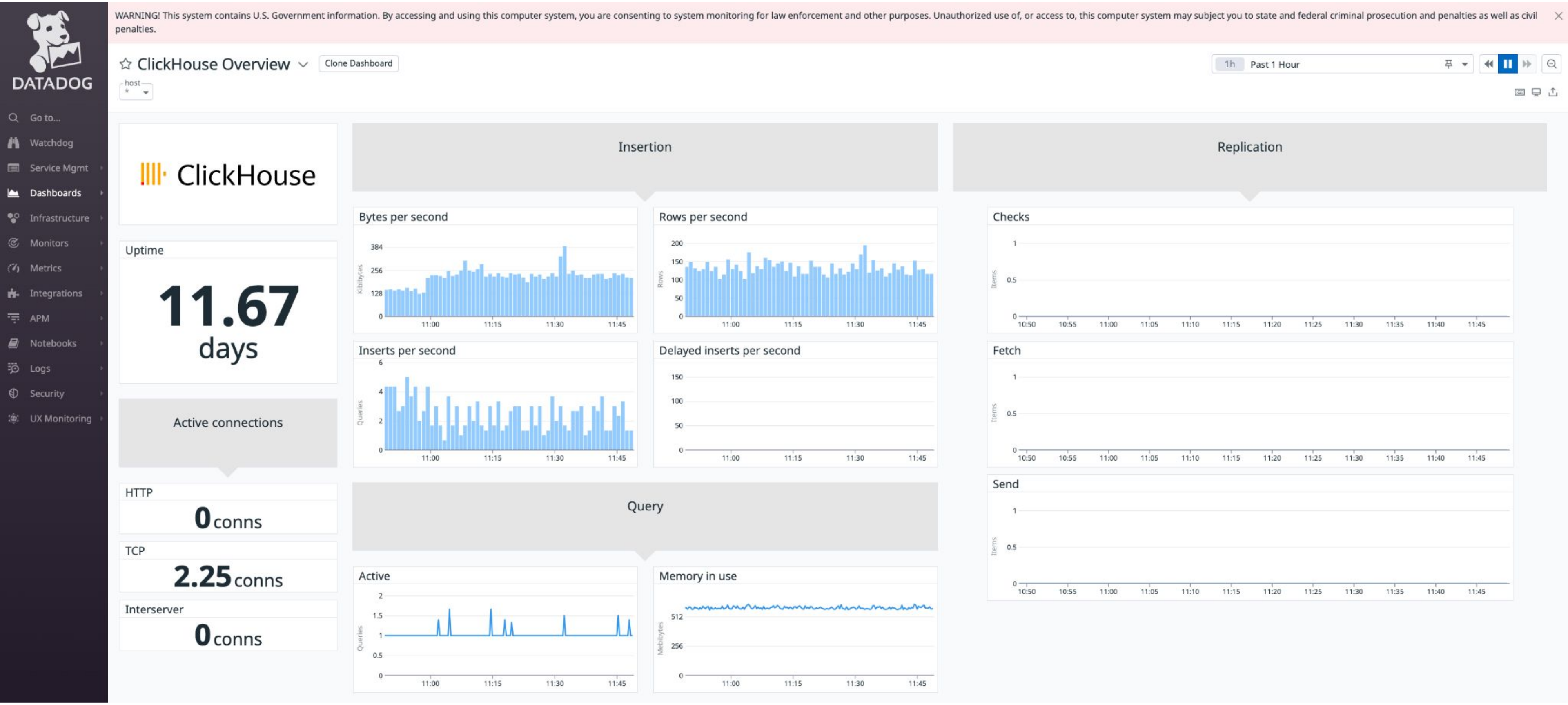
How do we build BoringSSL?

FIPS-certification is not just WHAT but HOW you build the software

- Must follow the [BoringCrypto FIPS 140-2 Non-Proprietary Security Policy]
- Run build process with docker using Golang Dockerfile
- Copy out the headers and binaries from the docker container to the host filesystem
- Modify ClickHouse build dependencies so that it uses headers and binaries extracted in step above.
- Continue to build ClickHouse

Migrate from Graphite/Grafana to DataDog

- Straight forward configurations in Clickhouse and DataDog agent.



WARNING! This system contains U.S. Government information. By accessing and using this computer system, you are consenting to system monitoring for law enforcement and other purposes. Unauthorized use of, or access to, this computer system may subject you to state and federal criminal prosecution and penalties as well as civil penalties.

ClickHouse Overview

Uptime: 11.67 days

Active connections

HTTP: 0 conns

TCP: 2.25 conns

Interserver: 0 conns