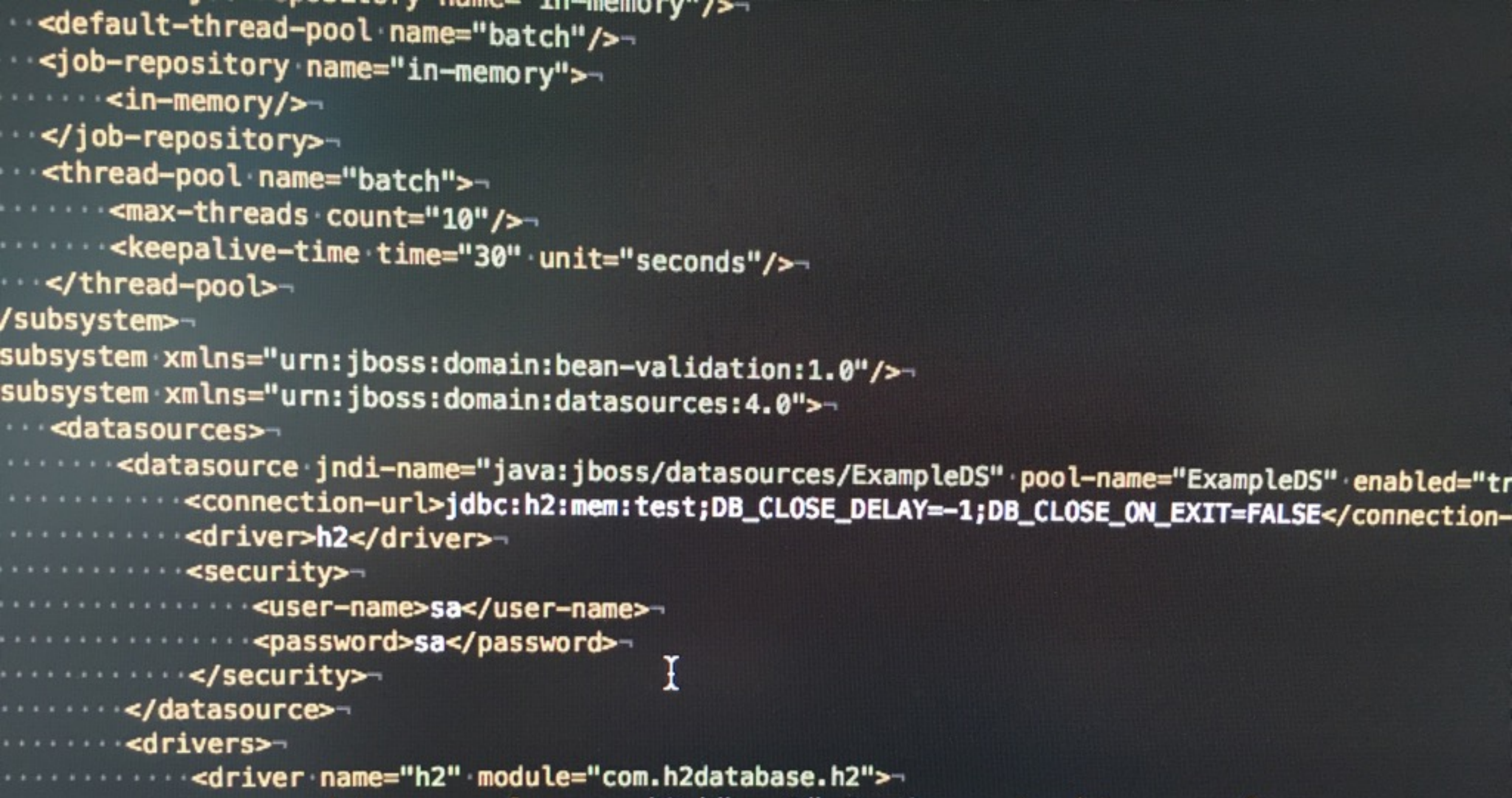
Managing Secrets at Scale

Mark Paluch

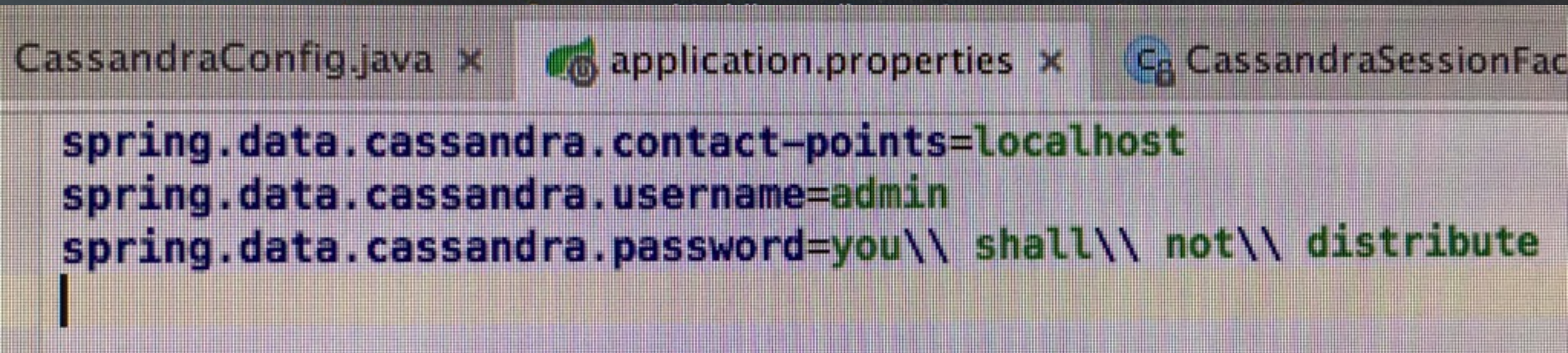
@mp911de

github.com/mp911de

paluch.biz



```
    <job-repository name="in-memory"/>
  <default-thread-pool name="batch"/>
  <job-repository name="in-memory">
      <in-memory/>
  </job-repository>
  <thread-pool name="batch">
      <max-threads count="10"/>
      <keepalive-time time="30" unit="seconds"/>
  </thread-pool>
/subsystem>
subsystem xmlns="urn:jboss:domain:bean-validation:1.0"/>
subsystem xmlns="urn:jboss:domain:datasources:4.0">
  <datasources>
      <datasource jndi-name="java:jboss/datasources/ExampleDS" pool-name="ExampleDS" enabled="tr
          <connection-url>jdbc:h2:mem:test;DB_CLOSE_DELAY=-1;DB_CLOSE_ON_EXIT=FALSE</connection-
          <driver>h2</driver>
          <security>
              <user-name>sa</user-name>
              <password>sa</password>
          </security>
      </datasource>
      <drivers>
          <driver name="h2" module="com.h2database.h2">
```



```
spring.data.cassandra.contact-points=localhost
spring.data.cassandra.username=admin
spring.data.cassandra.password=you\\ shall\\ not\\ distribute
```

TomEE

```
<Resource id="MySQL Database" type="DataSource">
    UserName    test

    xMH5uM1V9vQzVUv5LG7YLA==
    Password    xMH5uM1V9vQzVUv5LG7YLA==
    PasswordCipher Static3DES
</Resource>
```

https://www.flickr.com/photos/dahlstroms/4188244058

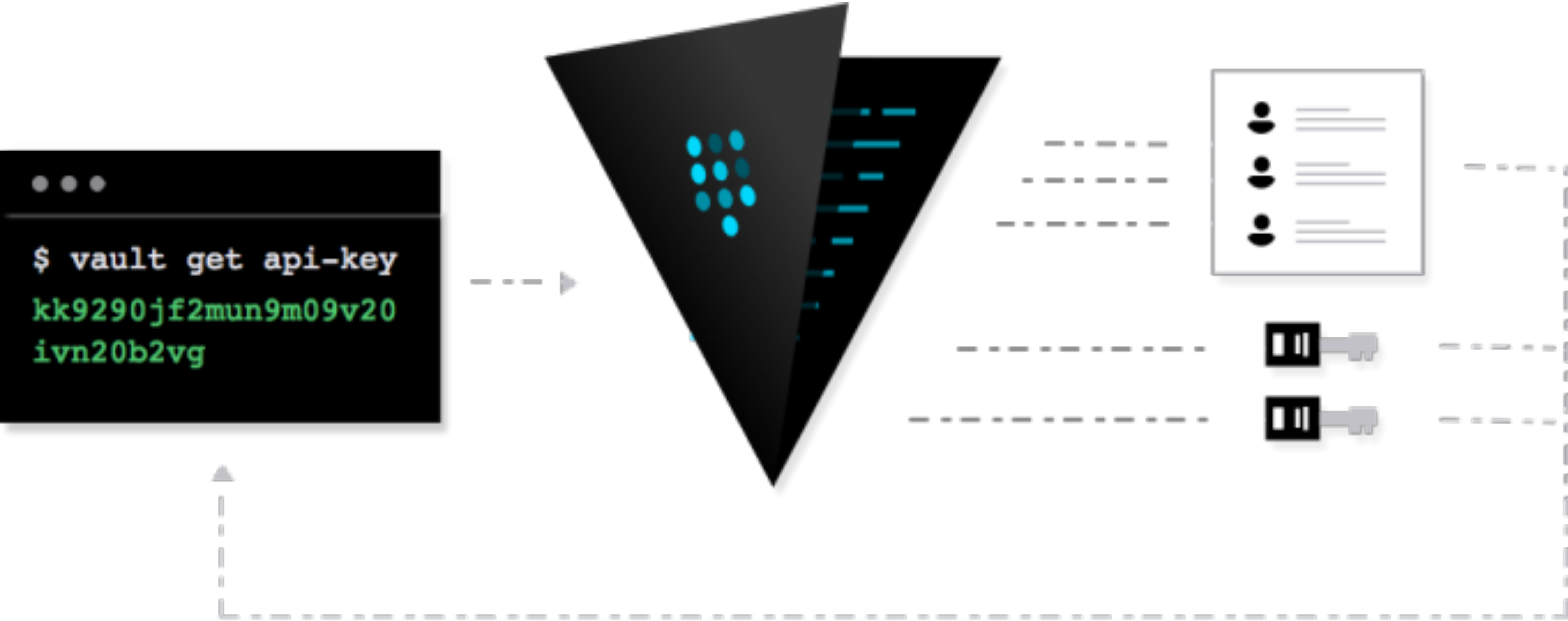
Change all the Passwords

https://www.flickr.com/photos/nateone/5456129071

Keeping secrets secret

- Limit distribution
- Access control
- Encrypted
- Key rotation
- Locking access

A tool for managing secrets.



```
$ vault get api-key
kk9290jf2mun9m09v20
ivn20b2vg
```

Vault Project

- Secure storages
- Sealing/Unsealing
- Multiple authentication mechanisms
- Multiple secret backends
- ACL/policies
- HA
- HTTP API

Vault Project: Editions

Community

- Secret storage
- Tokens and access control policies
- Dynamic secrets with leasing and revocation
- Key rolling
- Audit logs

Enterprise

- HSM
- 24x7x365 Phone and Email Support

Demo: Start and initialize Vault

Generic secret backend

- Store arbitrary secrets
- Hierarchical paths
- JSON data structures

Demo: Storing/Loading generic secrets

Secret backends

- AWS
- Cassandra
- Consul
- MySQL/MSSSQL/PostgreSQL
- PKI

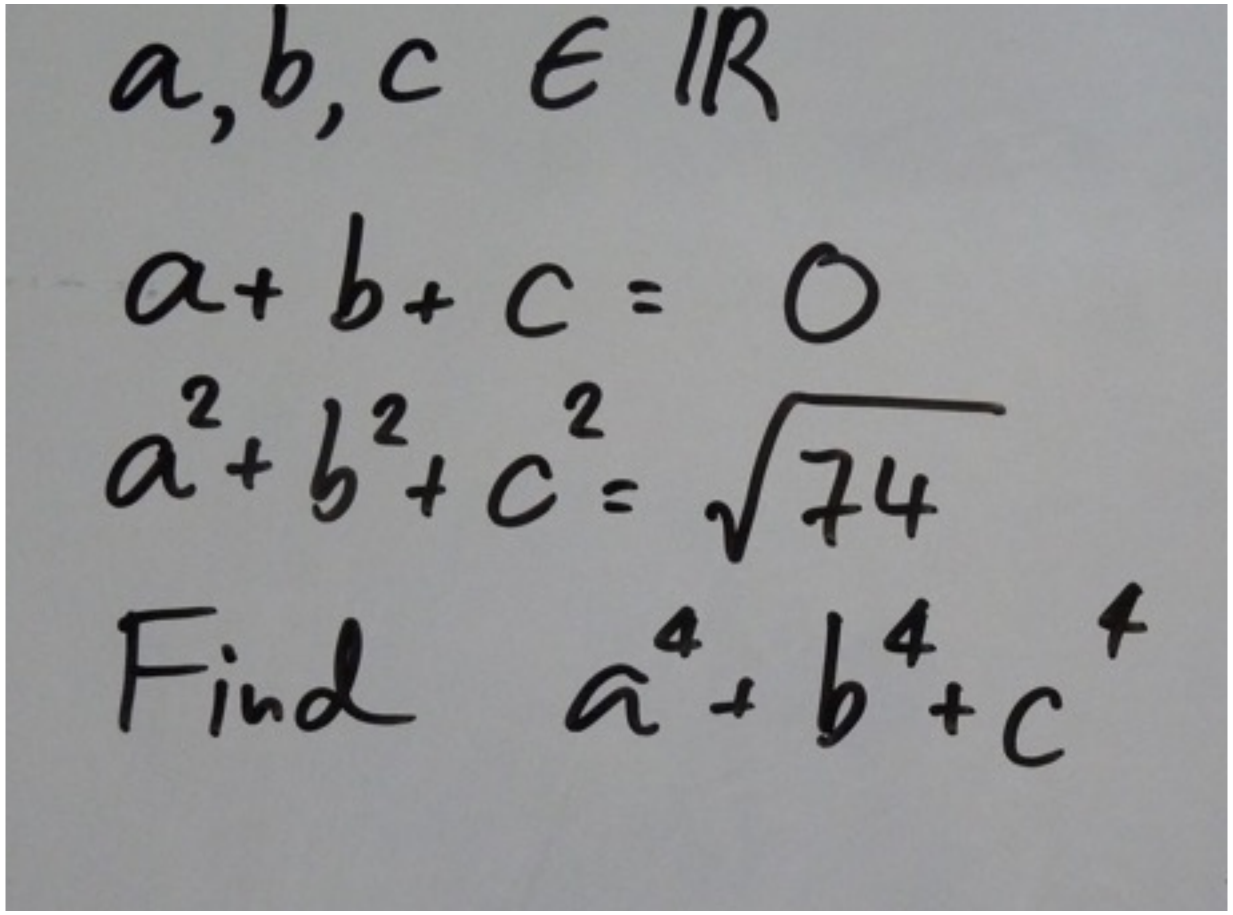
Keeping secrets secret

- Limit distribution ✅
- Access control
- Encrypted ✅
- Key rotation ✅
- Locking access

https://www.flickr.com/photos/kristencavanaugh/10710047746

Authentication methods

- Token
- Username/password
- LDAP
- GitHub Token
- MFA
- TLS Certificates
- App ID



$a,b,c \in \mathbb{R}$

$a+b+c = 0$

$a^2+b^2+c^2 = \sqrt{74}$

Find $a^4+b^4+c^4$

Two secure components

App Id

- Create unique AppId's (UUID, ...), map to policies
- Store AppId's in config management system
- Out-of-band process to map AppId to UserId
- New service: Knows AppId and determines UserId

Keeping secrets secret

- Limit distribution ✅
- Access control ✅
- Encrypted ✅
- Key rotation ✅
- Locking access ✅

Demo: Spring Cloud Vault Config

Operation hints

- Use SSL
- Use SSL
- Keep unseal keys secret
- Operate it HA

Key takeaways

- Vault is a secure storage service
- Versatile secrets
- Multiple authentication methods
- HTTP API
- Spring Cloud Vault Integration in the works

Q&A

System.exit(0);

Resources

- https://www.vaultproject.io/
- https://github.com/spencergibb/spring-cloud-vault-config/
- https://github.com/mp911de/spring-cloud-vault-config-samples