Hortonworks

Hadoop and Kerberos: The madness beyond the gate

Steve Loughran
stevel@hortonworks.com
@steveloughran
2016

Me: Before Kerberos


After Kerberos

Hortonworks

Leave now if you want to retain your life of naïve innocence


New! Lower price
DUNKIN' DONUTS
DUNKIN' DECAF
DECAF
DUNKIN' DONUTS
DECAFFEINATED
NET WT 12 OZ (340.2 g) COFFEE
McCafé
PREMIUM ROAST
SMOOTH & BALANCED
100% ARABICA BEANS
DECAF
MEDIUM
GROUND
NET WT 12 OZ (340g) 100% ARABICA COFFEE
UPC 88133400048
DUNKIN' DONUTS
DUNKIN' DECAF GROUND
DNKN DECF ORIG BLN GR COF 12 OZ
8.79
11.72 PER POUND
7.99
10.65 PER POUND
New! Lower price
MCCF PREM RST DECAF BAG 12 OZ
6.99
9.32 PER POUND
ORIGINAL BLEND
WHOLE BEAN
WHOLE BEAN
Seattle's Best Coffee
SIGNATURE BLEND №3
Medium & Balanced
NEW!

CHAPTER 2

Securing Distributed Systems

Chapter to Come

```
export HADOOP_USER="root"
```

Hortonworks

Modern Hadoop clusters are locked down through Kerberos

Hortonworks

You cannot hide from Kerberos

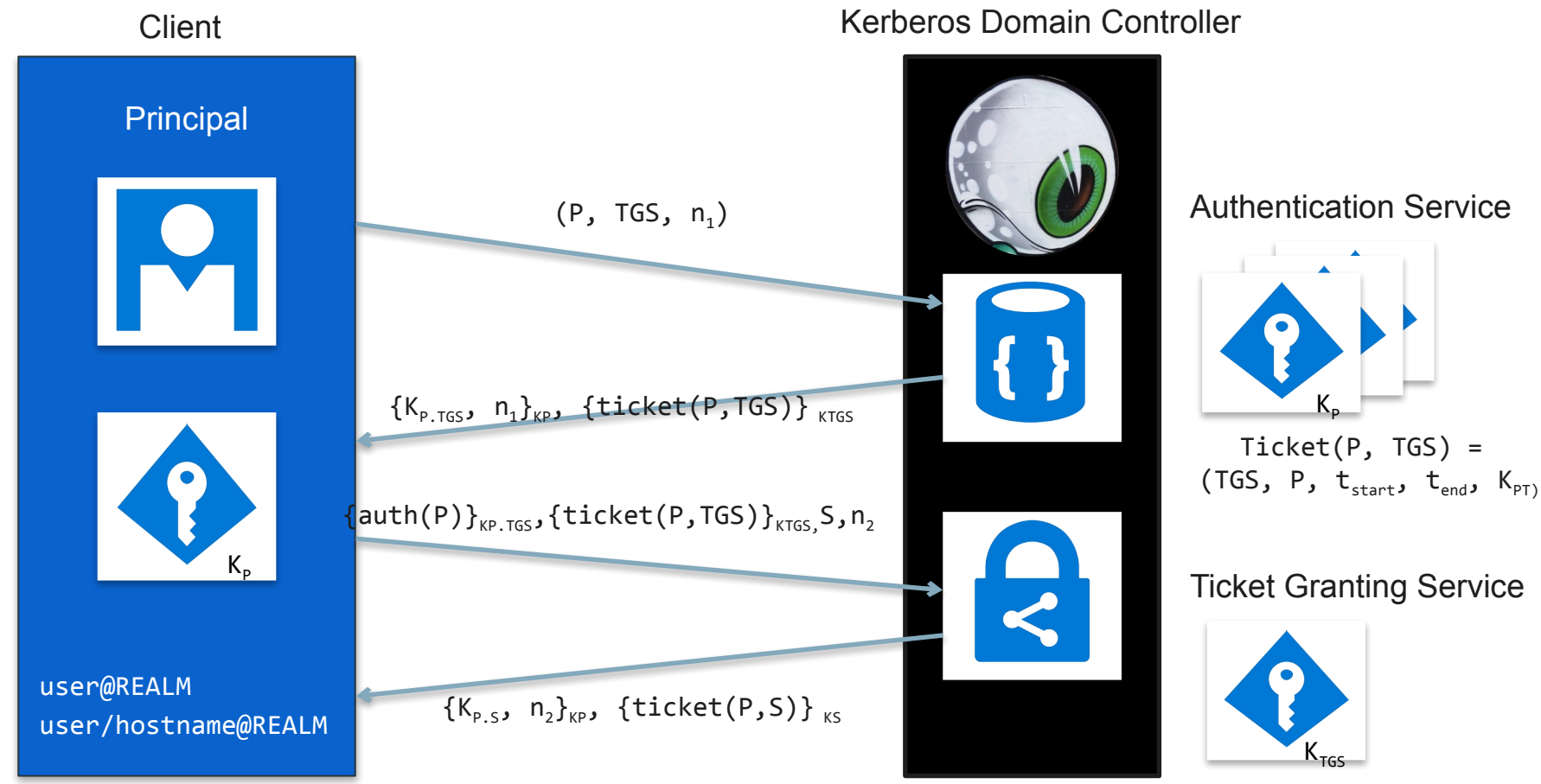
You may choose when Kerberos finds you

Kerberos:
the dog at the gate to hell


This is not a metaphor
Art: Andrés Álvarez Iglesias

HP Lovecraft	Kerberos
Evil lurking in New England	MIT Project Athena
Ancient, inhuman deities	Kerberos Domain Controller
Manuscripts to drive the reader insane	IETF RFC 4120
Entities never spoken of aloud	`UserGroupInformation`
Doomed explorers of darkness	*You*

Kerberos is the gateway



$auth(P)_{KP.TGS} = \{P, time)\}_{KP.TGS}$

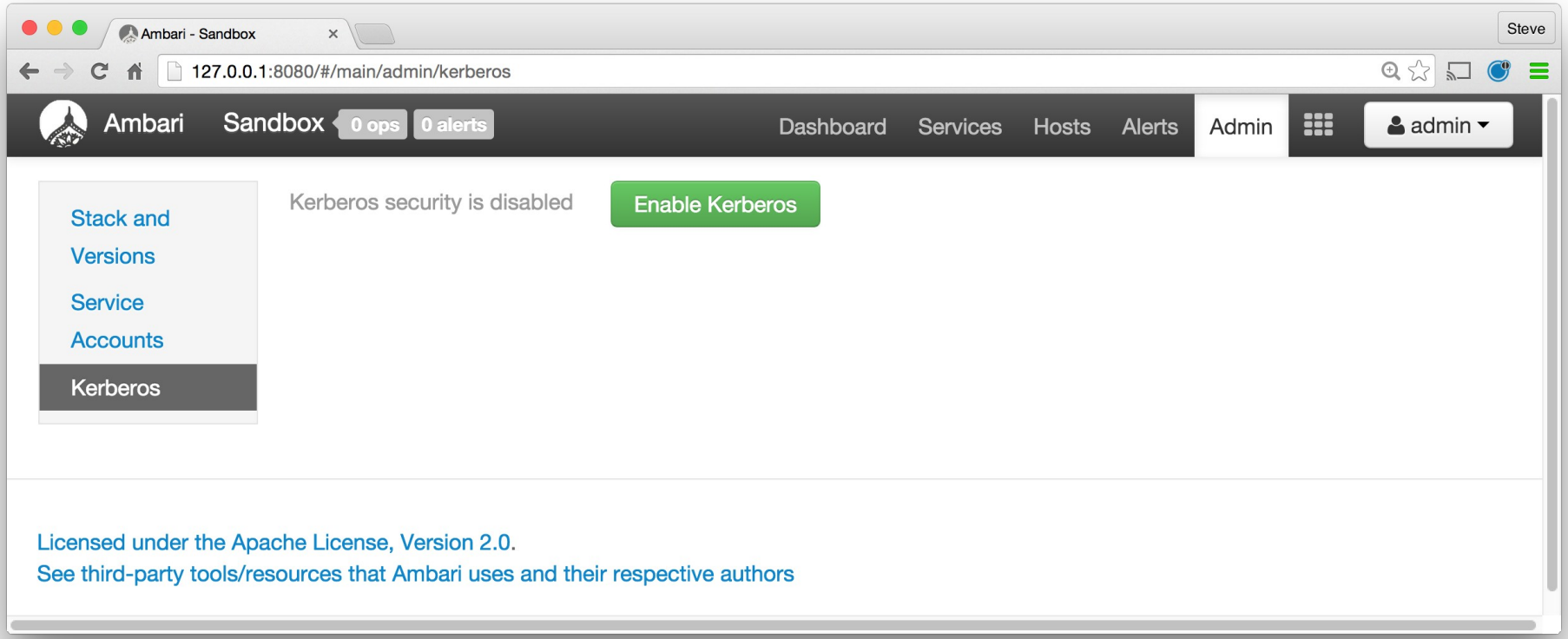
Every service is a principal

```
alice@REALM
bob@REALM

oozie/ooziehost@REALM
namenode/nn1@REALM

hdfs/_HOST@REALM
  hdfs/r04s12@REALM
  hdfs/r04s13@REALM
yarn/_HOST@REALM
  yarn/r04s12@REALM
HTTP/_HOST@REALM
```

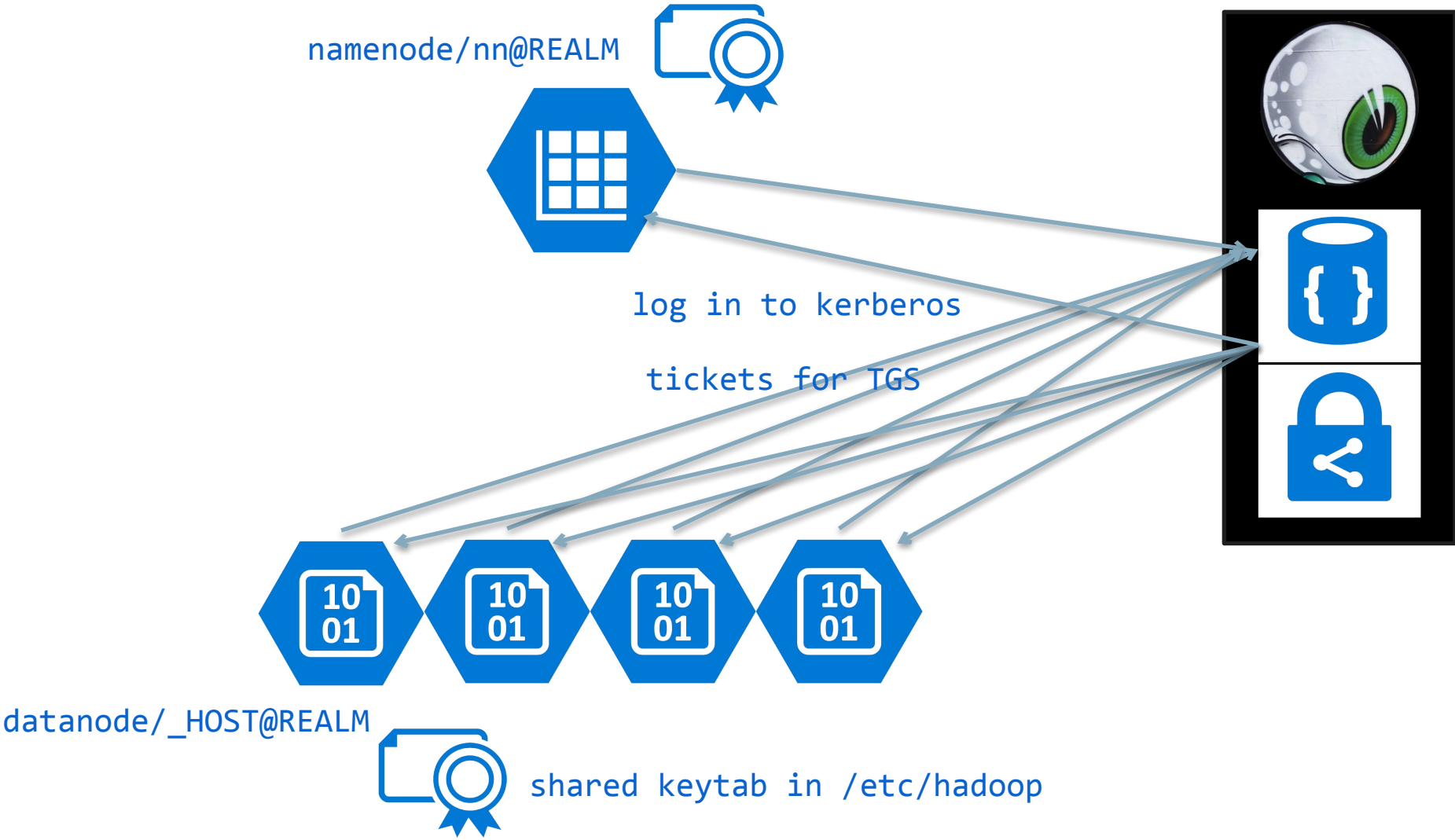
short names:

```
alice
bob
oozie
namenode
hdfs
yarn
HTTP
```

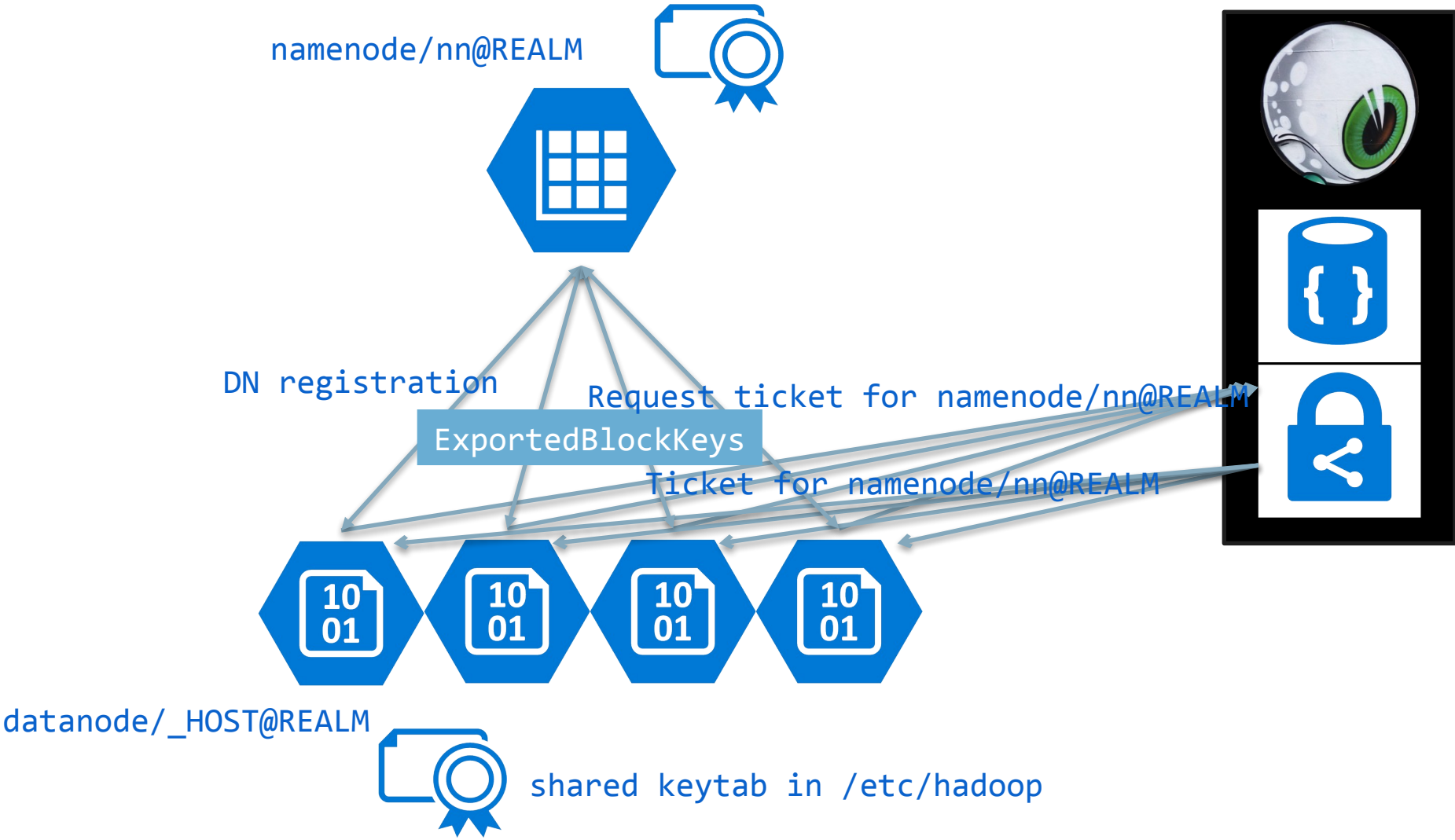
Entering the darkness



HDFS Bootstrap: Kerberos Login



HDFS Bootstrap: DNs register with NN



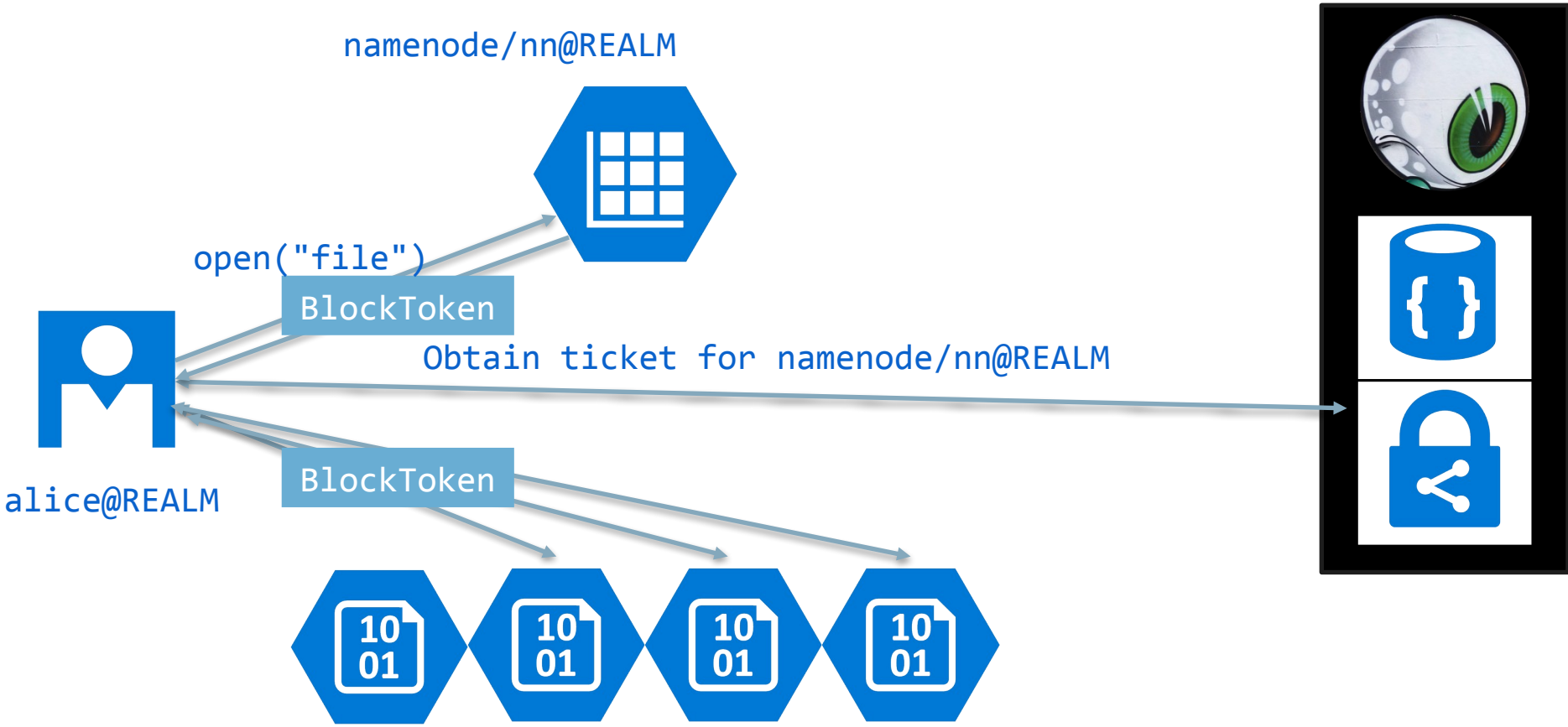
Hadoop Tokens

Hadoop Tokens

- Issued and tracked by individual services (HDFS, WebHDFS, Timeline Server, YARN RM, …)
- Grant some form of access: *Block tokens*, *Delegation Tokens*
- Can be forwarded
- Renewable via service APIs (RPC, HTTP)
- Revocable in server via service APIs

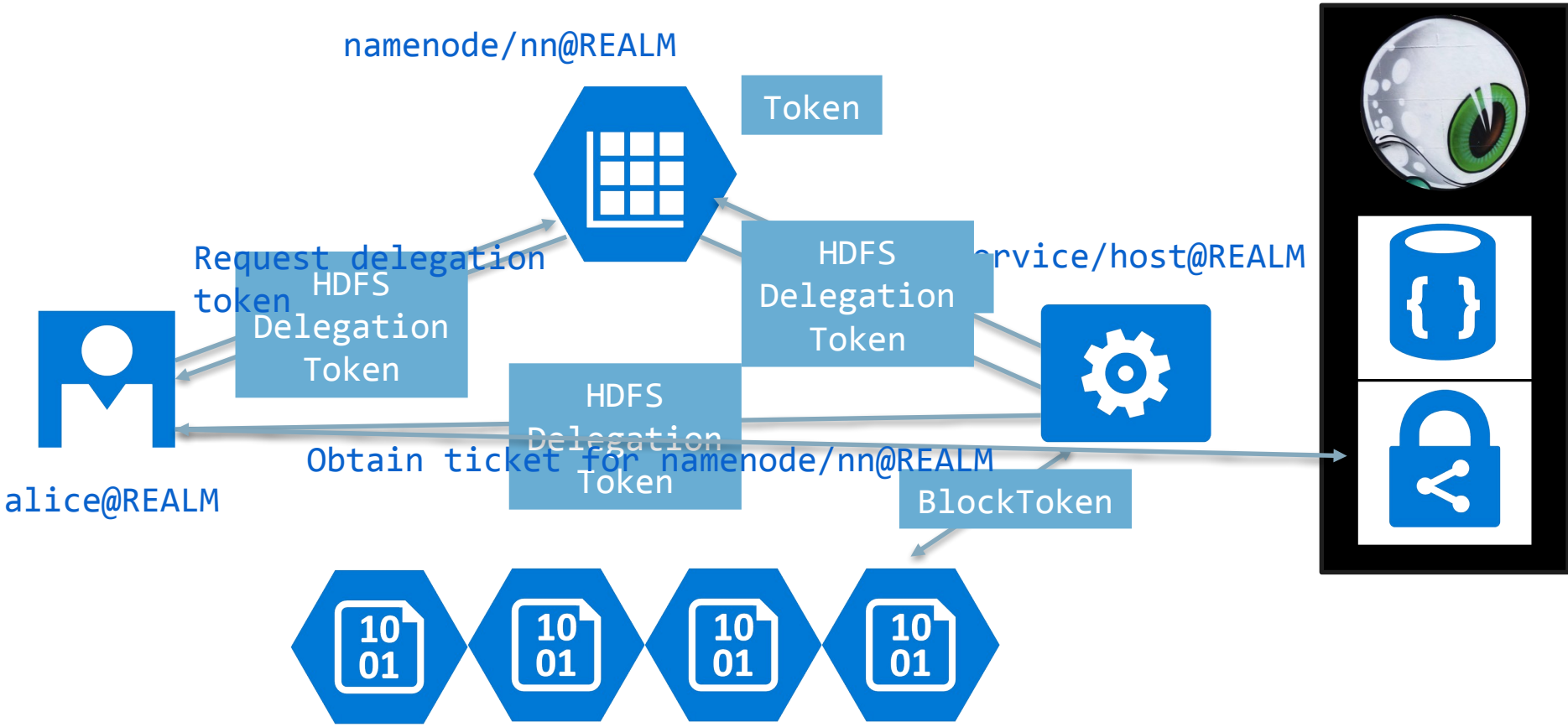
read: O'Malley 2009, *Hadoop Security Architecture*

HDFS IO: Block Tokens

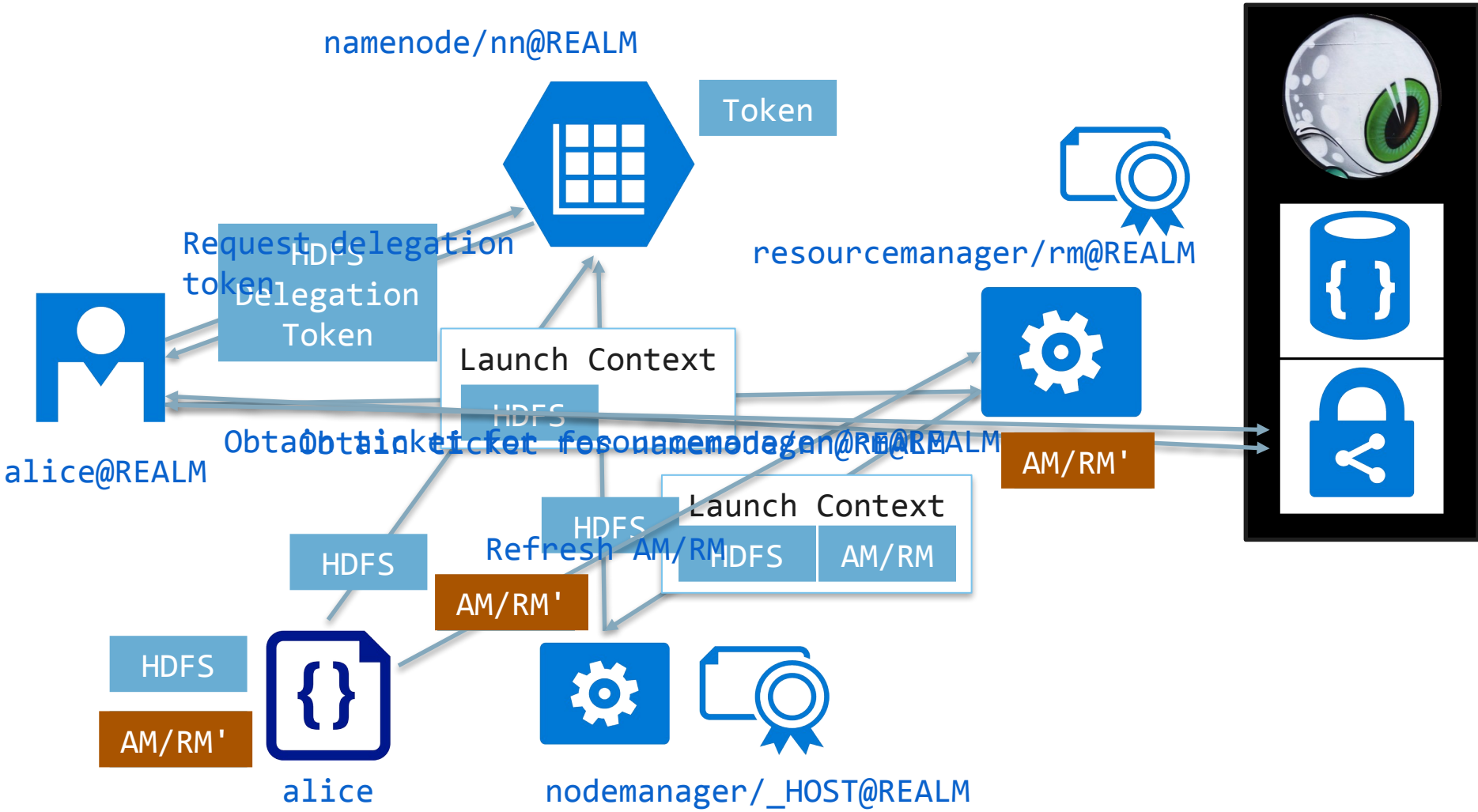


BlockToken: userId, (BlockPoolId, BlockId), keyId, expiryDate, access-modes

Delegation Tokens delegate access



YARN Applications



That which must not be named: *UGI*

```
if(!UserGroupInformation.isSecurityEnabled()) {
  stayInALifeOfNaiveInnocence();
} else {
  sufferTheEternalPainOfKerberos();
}

UserGroupInformation.checkTGTAndReloginFromKeytab();

UserGroupInformation.getLoginUser()   // principal logged in as
UserGroupInformation.getCurrentUser() // principal acting as
```

UGI.doAs()

```
UserGroupInformation bob =
  UserGroupInformation.createProxyUser("bob",
   UserGroupInformation.getLoginUser());
```

```
FileSystem userFS = bob.doAs(
  new PrivilegedExceptionAction<FileSystem>() {
    public FileSystem run() throws Exception {
      return FileSystem.get(FileSystem.getDefaultUri(), conf);
    }
  });
```

Services

- RPC authentication via annotations & metadata in JAR
- YARN Web UIs: rely on RM proxy for authentication
- Authentication != Authorization
- Add audit logs on service endpoints
- YARN services: come up with a token refresh strategy: keytab everywhere; keytab in AM; update from client

Hadoop RPC

```
@KerberosInfo(serverPrincipal = "my.kerberos.principal")
public interface MyRpc extends VersionedProtocol { … }

public class MyRpcPolicyProvider extends PolicyProvider {
  public Service[] getServices() {
    return  new Service[] {
     new Service("my.protocol.acl", MyRpc.class)
    };
  }
}
public class MyRpcSecurityInfo extends SecurityInfo { … }

META-INF/services/org.apache.hadoop.security.SecurityInfo
   org.example.rpc.MyRpcSecurityInfo
```

IPC Server: get the current user identity

```
Messages.KillResponse killContainer(Messages.KillRequest request) {

UserGroupInformation callerUGI;

try {
  callerUGI = UserGroupInformation.getCurrentUser();
} catch (IOException ie) {
  LOG.info("Error getting UGI ", ie);
  AuditLogger.authFail("E_UNKNOWN", "killContainer",
    "Error getting UGI", ie);
  throw RPCUtil.getRemoteException(ie);
}

…
```

IPC Server: Authorize

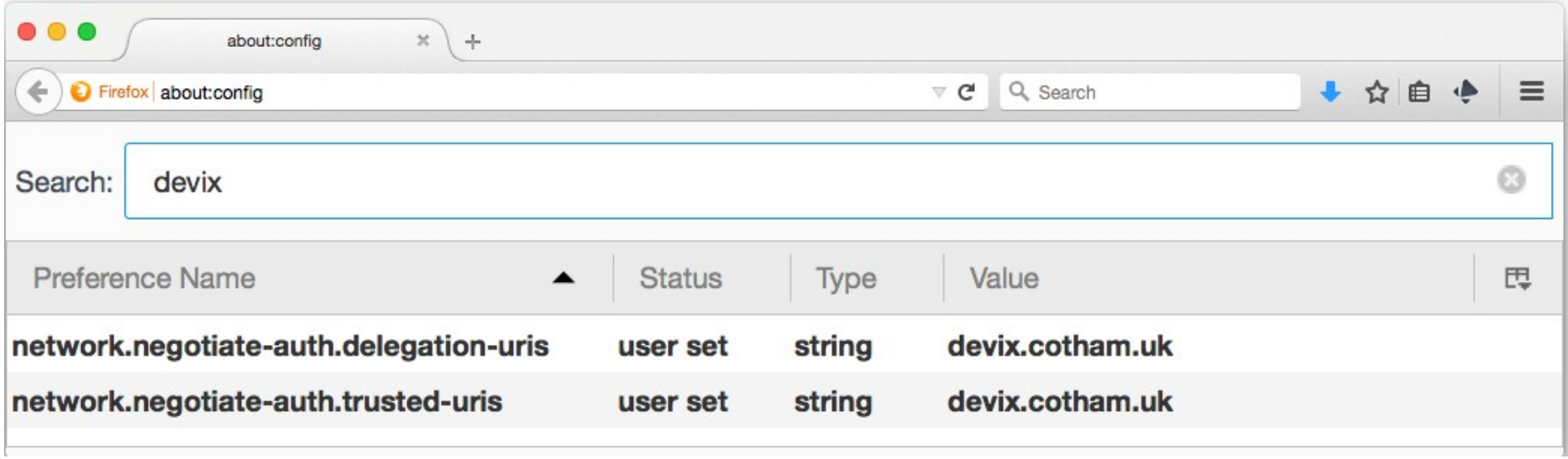
```
String user = callerUGI.getShortUserName();

if (!checkAccess(callerUGI, MODIFY)) {
  AuditLog.unauth(user,
    KILL_CONTAINER_REQUEST, callerUGI,
    "User doesn't have permissions to " + MODIFY);
  throw RPCUtil.getRemoteException(
    new AccessControlException(
      + user + " lacks access "
      + MODIFY_APP.name()));
}

AuditLog.authorized(user, KILL_CONTAINER_REQUEST)
```

SASL: RFC4422

SASL is conceptually a framework that provides an abstraction layer between protocols and mechanisms as illustrated in the following diagram.

```
          SMTP     LDAP     XMPP    Other protocols ...
            \        |        |        /
             \       |        |       /
            SASL abstraction layer
             /       |        |       \
            /        |        |        \
     EXTERNAL    GSSAPI   PLAIN    Other mechanisms ...
```

REST: SPNEGO (+ Delegation tokens)



- Jersey + java.net
- httpclient? *"if lucky it'll work"*

HADOOP-11825: *Move timeline client Jersey+Kerberos+UGI support into a public implementation*

Testing

```
krb-72h.conf  krb-shortlived.conf  krb-shortlived.conf\~  krb5.conf
root@devix /etc# cat krb-shortlived.conf
[logging]
 default = FILE:/var/log/krb5libs.log
 kdc = FILE:/var/log/krb5kdc.log
 admin_server = FILE:/var/log/kadmind.log

[libdefaults]
 default_realm = COTHAM
 dns_lookup_realm = false
 dns_lookup_kdc = false
 ticket_lifetime = 1h
 renew_lifetime = 2h
 forwardable = true

[realms]
 COTHAM = {
   kdc = devix
   admin_server = devix
 }

#[domain_realm]
# .example.com = EXAMPLE.COM
# example.com = EXAMPLE.COM
root@devix /etc#
```

Error messages to fear

Failure unspecified at GSS-API level (Checksum failed)
No valid credentials provided (Failed to find any Kerberos tgt)
Server not found in Kerberos database
Clock skew too great
Principal not found
No valid credentials provided (Illegal key size)

Art: Andrés Álvarez Iglesias

System Properties for debugging

```
-Dsun.security.krb5.debug=true
-Dsun.security.spnego.debug=true

export HADOOP_JAAS_DEBUG=true
```


HADOOP-12649
Fix Kerberos

slider-funtest — fish /Users/stevel/Projects/Hortonworks/Projects/slider/slider-funtest — fish — 193×59

fish /Users/stevel/Projects/Hortonworks/Projec... | fish /Users/stevel/Projects/Hortonworks/Projec... | fish /Users/stevel/Projects/Hortonworks/Projec... | fish /Users/stevel/Projects/Hortonworks/Projec... | fish /Users/stevel/Java/Apache/slider-dist — fish ...

```
== Examining keytab /Users/stevel/Projects/Hortonworks/Projects/clusterconfigs/clusters/devix/keytabs/dn.service.keytab ==

keytab entry count: 1
    dn/devix.cotham.uk@COTHAM
-----
Using keytab /Users/stevel/Projects/Hortonworks/Projects/clusterconfigs/clusters/devix/keytabs/dn.service.keytab principal dn/devix.cotham.uk@COTHAM
2016-01-17 19:21:41,280 [main] DEBUG security.Groups (getUserToGroupsMappingService(301)) -  Creating new Groups object
2016-01-17 19:21:41,286 [main] DEBUG security.JniBasedUnixGroupsMappingWithFallback (<init>(45)) - Group mapping impl=org.apache.hadoop.security.ShellBasedUnixGroupsMapping
2016-01-17 19:21:41,450 [main] DEBUG security.Groups (<init>(112)) - Group mapping impl=org.apache.hadoop.security.JniBasedUnixGroupsMappingWithFallback; cacheTimeout=300000; warningDeltaMs=500
0
Debug is  true storeKey true useTicketCache false useKeyTab true doNotPrompt true ticketCache is null isInitiator true KeyTab is /Users/stevel/Projects/Hortonworks/Projects/clusterconfigs/clust
ers/devix/keytabs/dn.service.keytab refreshKrb5Config is true principal is dn/devix.cotham.uk@COTHAM tryFirstPass is false useFirstPass is false storePass is false clearPass is false
Refreshing Kerberos configuration
principal is dn/devix.cotham.uk@COTHAM
Will use keytab
2016-01-17 19:21:41,652 [main] DEBUG security.UserGroupInformation (login(221)) - hadoop login
Commit Succeeded

2016-01-17 19:21:41,654 [main] DEBUG security.UserGroupInformation (commit(156)) - hadoop login commit
2016-01-17 19:21:41,655 [main] DEBUG security.UserGroupInformation (commit(170)) - using kerberos user: dn/devix.cotham.uk@COTHAM
2016-01-17 19:21:41,655 [main] DEBUG security.UserGroupInformation (commit(192)) - Using user: "dn/devix.cotham.uk@COTHAM" with name dn/devix.cotham.uk@COTHAM


2016-01-17 19:21:41,656 [main] DEBUG security.UserGroupInformation (commit(202)) - User entry: "dn/devix.cotham.uk@COTHAM"
== dn/devix.cotham.uk@COTHAM ==

UGI instance = dn/devix.cotham.uk@COTHAM (auth:KERBEROS)
Has kerberos credentials: true
Authentication method: KERBEROS
Real Authentication method: KERBEROS


== Group names ==

2016-01-17 19:21:41,685 [main] WARN  security.ShellBasedUnixGroupsMapping (getUnixGroups(87)) - got exception trying to get groups for user hdfs: id: hdfs: no such user


2016-01-17 19:21:41,689 [main] WARN  security.UserGroupInformation (getGroupNames(1521)) - No groups available for user hdfs

== Credentials ==



== Secret keys ==

(none)


== Token Count: 0 ==



== Attempting to log in from keytab again ==

2016-01-17 19:21:41,702 [main] DEBUG security.UserGroupInformation (getTGT(857)) - Found tgt Ticket (hex) =
0000: 61 82 01 3C 30 82 01 38   A0 03 02 01 05 A1 08 1B  a..<0..8........
0010: 06 43 4F 54 48 41 4D A2   1B 30 19 A0 03 02 01 02  .COTHAM..0......
```


gitbook.com/@steveloughran
HADOOP-12649

Topics ~~Avoided~~ Not Covered

- Trying to use HTTPS in a YARN application
- Trying to use Full REST in a YARN application
- Group management
- HADOOP_PROXY_USER
- Oozie

Art: Andrés Álvarez Iglesias

Zookeeper

- SASL to negotiate security:

```
System.setProperty("zookeeper.sasl.client", "true");
```

- Zookeeper needs JAAS
- Default permissions: wide open
- Permissions are not transitive down the tree

```
List<ACL> perms = new ArrayList<>();
if (UserGroupInformation.isSecurityEnabled()) {
  perms(new ACL(ZooDefs.Perms.ALL, ZooDefs.Ids.AUTH_IDS));
  perms.add(new ACL(ZooDefs.Perms.READ,ZooDefs.Ids.ANYONE_ID_UNSAFE));
} else {
  perms.add(new ACL(ZooDefs.Perms.ALL, ZooDefs.Ids.ANYONE_ID_UNSAFE));
}
zk.createPath(path, null, perms, CreateMode.PERSISTENT);
```

JAAS

- Java Authentication and Authorization Service
- Core Kerberos classes and types (`Principal`)
- Text files to configure
  - Different for different JVMs
  - Need to double escape \ for windows paths
- UGI handles setting up a JAAS context & logging in
- Zookeeper needs JAAS

Glossary

- KDC: Kerberos Domain Controller
- TGT/"krbtgt" Ticket Granting Ticket
- Simple Authentication and Security Layer (SASL)
- GSSAPI Generic Security Service Application Program Interface (RFC-2743+ others)
- JAAS: Java Authentication and Authorization Service
- SPNEGO: Simple and Protected GSSAPI Negotiation Mechanism