Data Protection and Security with Red Hat

Using Red Hat JBoss Data Virtualization, Red Hat JBoss Fuse, Red Hat JBoss BRMS, Red Hat Storage, and 3scale by Red Hat

Lucy Huh Kerner
Principal Product Technical Marketing Manager - Security, Red Hat
May 3, 2017

redhat.

Security at each layer: Data, Messaging, and Application to Application

redhat.

“You can’t effectively protect your data if you don’t know where it resides.”

Verizon’s 2016 Data Breach Investigations Report[1]

[1][http://www.verizonenterprise.com/resources/reports/rp_dbir-2016-executive-summary_xg_en.pdf]

redhat.

“If the [messaging] traffic is unencrypted, the underlying layer is essentially unencrypted network traffic susceptible to network attacks.”

TechTarget SearchSecurity[2]

[2]http://searchsecurity.techtarget.com/news/1307863/Hannaford-breach-highlights-messaging-system-struggles

redhat.

“API breaches have for some time now plagued startups, and it’s clear these breaches are now affecting established enterprises. APIs offer a new and powerful attack vector for hackers…”

eSecurity Planet[4]

[4]http://www.esecurityplanet.com/network-security/do-apis-pose-a-security-risk.html

redhat.

“A lawsuit claims that data from more than 330,000 accounts were illegally accessed using the Get Transcript API”

The CyberSecurity Place[3]

[3]https://thecybersecurityplace.com/api-security-key-takeaways-from-recent-breaches/

redhat.

Now, story time!

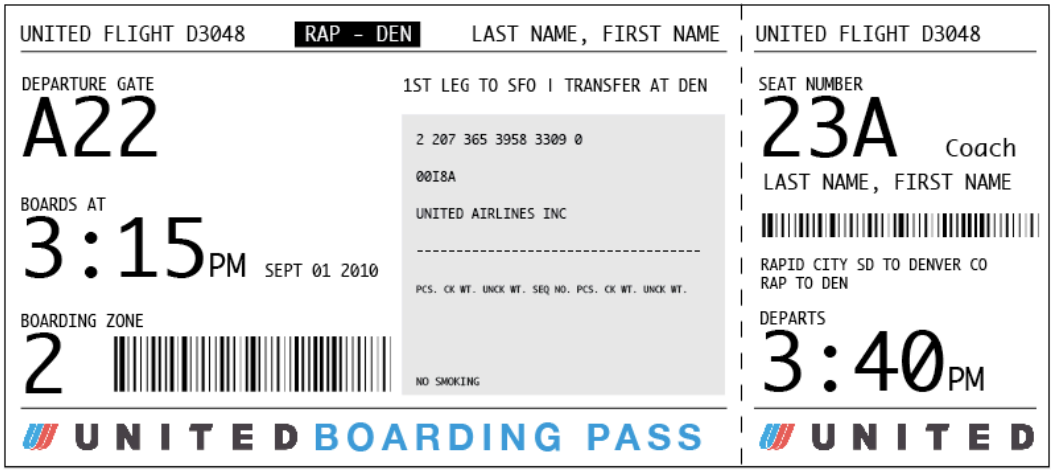
redhat.

A sea of blue…. Recognize this fun place ?

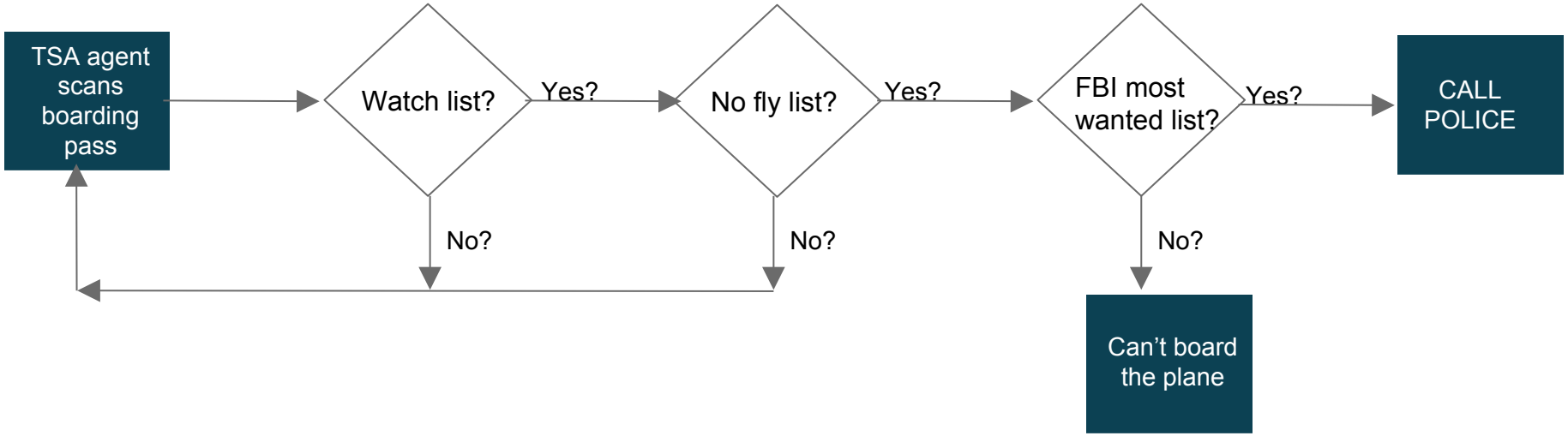
How do we know which passenger is a potential terrorist who should not be allowed on the plane?

redhat.

It all starts with your boarding pass ….



After scanning your boarding pass …



How could I implement this with Red Hat's products?

redhat.

A brief intro to some Red Hat products that can help…

Red Hat JBoss Fuse: Lightweight integration platform for system to system mediation and integration of disparate systems. Also, handles the security between the systems. Service integration.

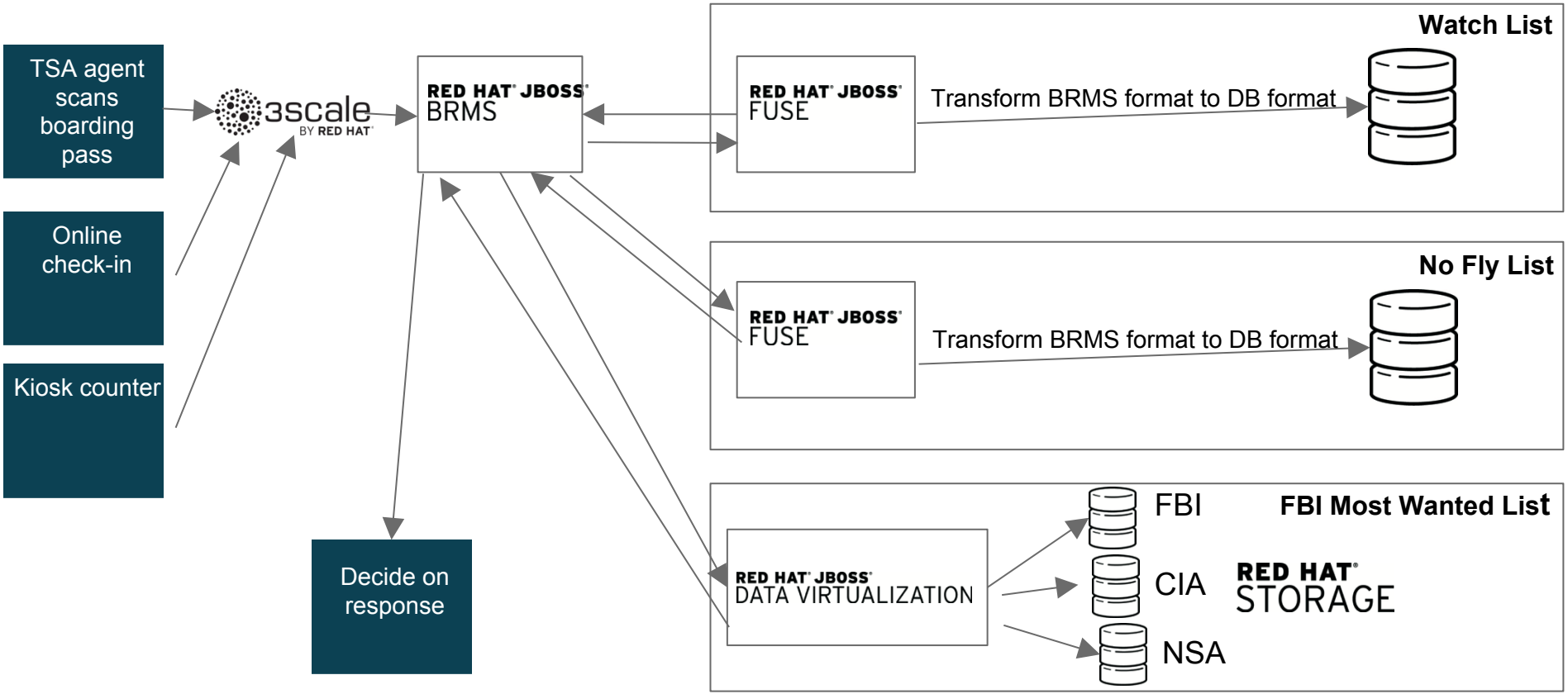
Red Hat JBoss Data Virtualization: Unifies data spread across various disparate sources and makes it available to apps as a single consolidated data source. Data integration.

Red Hat JBoss BRMS(Business Rules Management System): Platform for business rules management and complex event processing; Extract business rules from your java code, allowing for more dynamic and responsive changes to the rules even by business group (vs IT)

Red Hat Storage: Red Hat Gluster: scalable, distributed file system for big, unstructured, and semistructured data and object storage ; Red Hat Ceph Storage: scalable block and object storage

3scale by Red Hat: API management platform

How can I implement this?



Security at each layer with Red Hat: Data, Messaging, and Application to Application

redhat.

RED HAT SUMMIT

THANK YOU

plus.google.com/+RedHat

facebook.com/redhatinc

linkedin.com/company/red-hat

twitter.com/RedHatNews

youtube.com/user/RedHatVideos

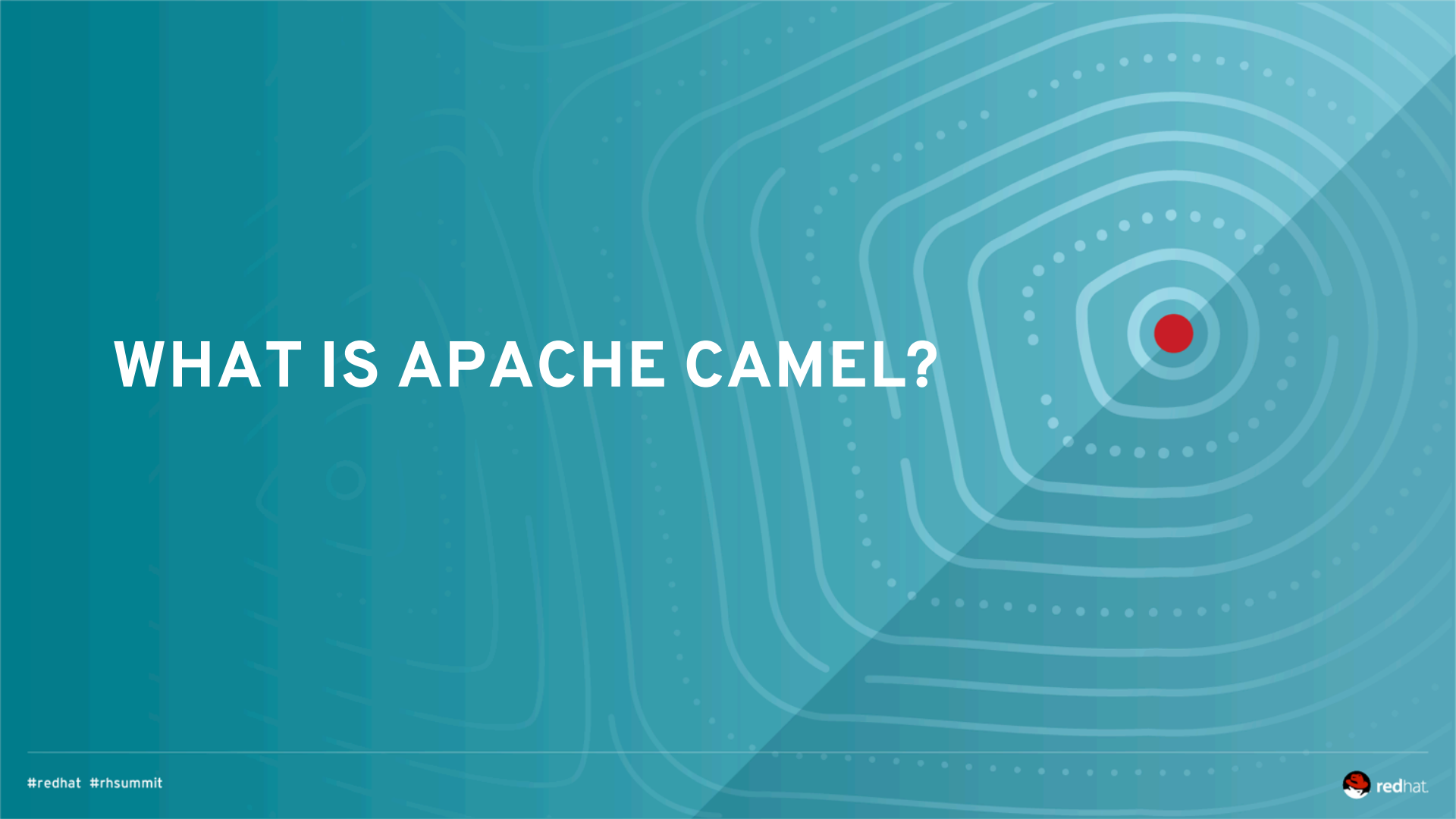
THE EASE OF THE CAMEL REST DSL

USING REST WITH JBOSS EAP AND APACHE KARAF

Mary Cochran
Enterprise Integration Practice Middleware Consultant
May 3, 2017

redhat.

WHAT IS APACHE CAMEL?



redhat.

WHAT IS CAMEL?

- An open source integration framework
- Integration components for smart routing, transformation, mediation, monitoring, etc
- A java based implementation of the Enterprise Integration Patterns

redhat.

ENDPOINT URI

component://specific-part?key=value&key=value

- ftp://john@localhost?password=doe
- jms:queue:MyQueue
- file:input?move=processed&moveFailed=failed

redhat.

JAVA DSL

- Defines a timer which runs every 2 seconds
- Logs at INFO Level the string “Hello World” under the package specified
- Logs the body itself at INFO level

```
public class ExampleRoute extends RouteBuilder {
    @Override
    public void configure() throws Exception {
        from("timer://myTimer?fixedRate=true&period=2000")
            .log(LoggingLevel.INFO, "com.redhat.quickstarts.karaf.route", "Hello World")
            .to("log:HelloWorldLog?level=INFO");
    }
}
```


redhat.

XML Route

- Defines a timer which runs every 2 seconds
- Logs at INFO Level the string “Hello World” under the package specified
- Logs the body itself at INFO level

```
<route>
  <from uri="timer://myTimer?fixedRate=true&amp;period=2000"/>
  <log message="Hello World" loggingLevel="INFO" loggerRef="com.redhat.quickstarts.karaf.route"/>
  <to uri="log:HelloWorldLog?level=INFO"/>
</route>
```


redhat.

APACHE KARAF WITH THE REST DSL

redhat.

KARAF CAMEL CONTEXT

```
<?xml version="1.0" encoding="UTF-8"?>
<blueprint xmlns="http://www.osgi.org/xmlns/blueprint/v1.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:cm="http://aries.apache.org/blueprint/xmlns/blueprint-cm/v1.0.0"
    xmlns:jaxws="http://cxf.apache.org/blueprint/jaxws"
    xmlns:camel="http://camel.apache.org/schema/blueprint"
    xsi:schemaLocation="http://www.osgi.org/xmlns/blueprint/v1.0.0 http://www.osgi.org/xmlns/blueprint/v1.0.0/
    blueprint.xsd http://cxf.apache.org/blueprint/jaxws http://cxf.apache.org/schemas/blueprint/jaxws.xsd
    http://cxf.apache.org/blueprint/core http://cxf.apache.org/schemas/blueprint/core.xsd">
    <!-- The camel context which registers the route -->
    <camel:camelContext id="fusequickstart-restdsl-camel" xmlns="http://camel.apache.org/schema/blueprint">
        <!-- Package Scanning finds the Routes -->
        <packageScan>
            <package>com.redhat.consulting.fusequickstarts.karaf.rest.dsl.route</package>
        </packageScan>
    </camel:camelContext>
</blueprint>
```


redhat.

KARAF CAMEL ROUTE

```
public void configure() throws Exception {
    Note sampleMessage = new Note();
    sampleMessage.setTo("User");
    sampleMessage.setMessage("REST is Awesome");

    restConfiguration().component("restlet").host("localhost").port(8182).bindingMode(RestBindingMode.auto);

    rest("/rest").consumes("application/json").produces("application/json")
        .get("/message").outType(Note.class).to("direct:get")
        .post("/message").type(Note.class).to("direct:post");

    from("direct:get")
        .process(new Processor() {
            public void process(Exchange exchange) throws Exception {
                exchange.getOut().setBody(sampleMessage);}
            }
         )
        .log("Successful GET Request: ${body}");

    from("direct:post")
        .log("Successful POST Request: ${body}")
        .setHeader(Exchange.HTTP_RESPONSE_CODE,simple("201"));
 }
```


redhat.

JBOSS EAP WITH THE REST DSL

redhat.

EAP WEB.XML

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<web-app version="2.4" xmlns="http://java.sun.com/xml/ns/j2ee"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
                        http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">
    <listener>
        <listener-class>org.jboss.weld.environment.servlet.Listener</listener-class>
    </listener>
    <!-- Camel Servlet -->
    <servlet>
        <servlet-name>CamelServlet</servlet-name>
        <servlet-class>org.apache.camel.component.servlet.CamelHttpTransportServlet</servlet-class>
        <load-on-startup>1</load-on-startup>
    </servlet>
    <!-- Camel Servlet Mapping -->
    <servlet-mapping>
        <servlet-name>CamelServlet</servlet-name>
        <url-pattern>/rest/*</url-pattern>
    </servlet-mapping>
</web-app>
```


redhat.

EAP CAMEL ROUTE

```
@ContextName("rest-dsl")
public class RestRoutes extends RouteBuilder {

    @Override
    public void configure() throws Exception {
        restConfiguration().component("servlet").bindingMode(RestBindingMode.json);

        rest("/user")
                .get("/{id}").produces("application/json").to("direct:getUser")
                .post().consumes("application/json").type(User.class).to("direct:createUser");

        from("direct:getUser")
                .log("GET /user/${header.id} request received!")
                .setBody(simple("${header.id}"))
                .to("bean:usersBean?method=getUser");

        from("direct:createUser")
                .log("POST /user/ request received!")
                .to("bean:usersBean?method=createUser");
    }
}
```


redhat.

CONVERTING A TIBCO BUSINESSWORKS APPLICATION TO APACHE CAMEL

Ashwin Karpe, Integration Practice Lead
Sundar Rajendran, Architect - Integration

Thursday, May 4, 11:30 AM - 12:15 PM

Located at the Consulting Discovery Zone at the Services Showcase in the Partner

Pavilion

redhat.

RED HAT SUMMIT

THANK YOU

plus.google.com/+RedHat

facebook.com/redhatinc

linkedin.com/company/red-hat

twitter.com/RedHatNews

youtube.com/user/RedHatVideos

HA DEPLOYMENTS WITH JBOSS FUSE

HIGH-AVAILABILITY WITH JBOSS FUSE/A-MQ

Josh Reagan
Middleware Integration Specialist
May 3, 2017

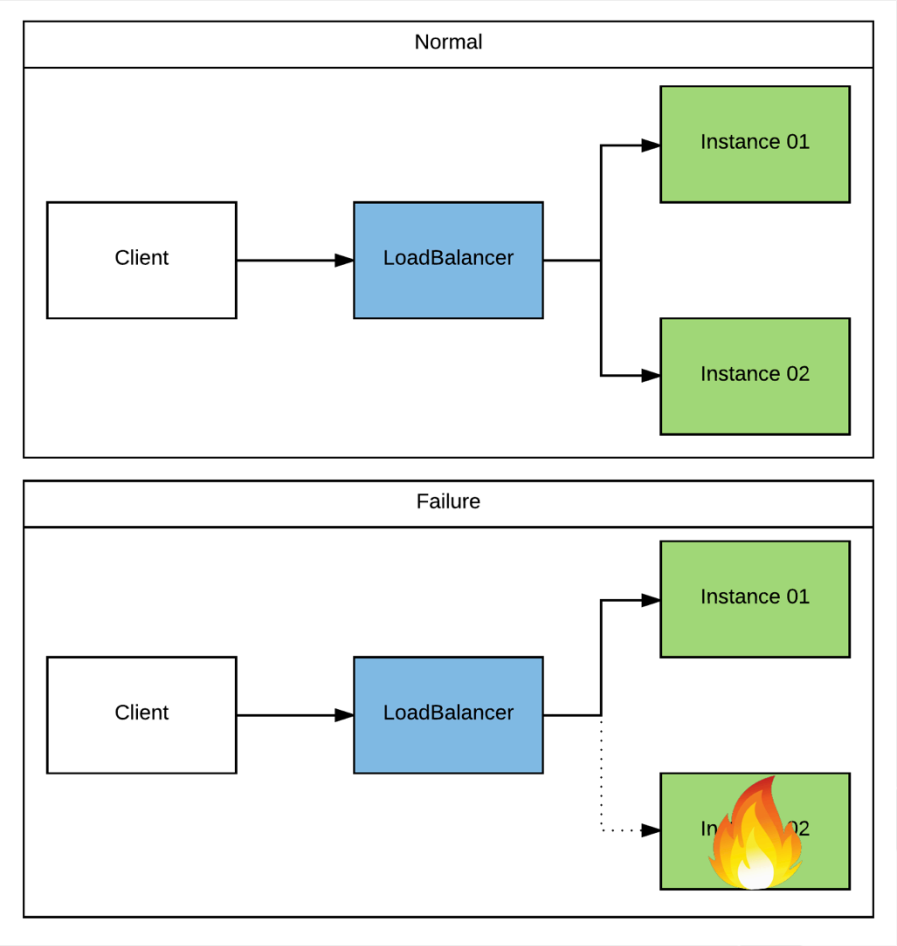
redhat.

HTTP - SOAP/REST

redhat.

HTTP - STATELESS

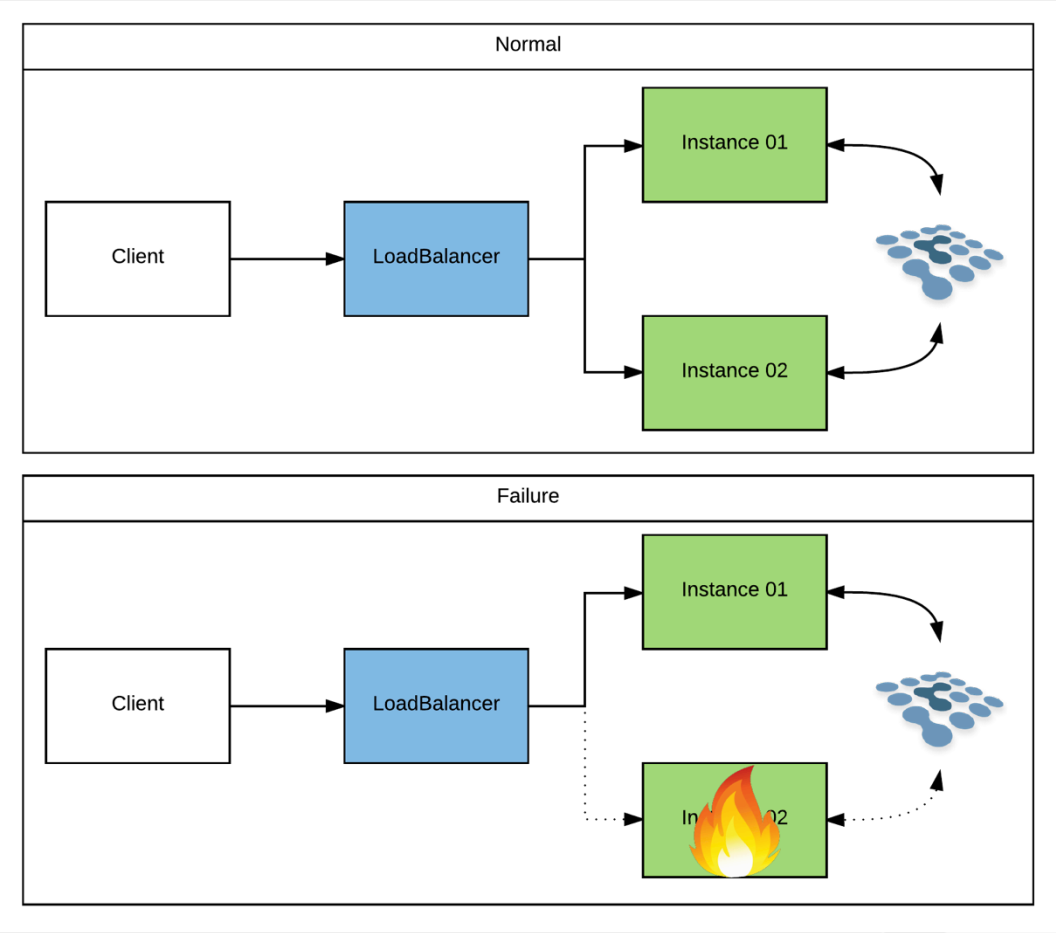
- Simple load balancer
  - Apache HTTPD, Layer7, f5, ...
- Multiple DNS A records



redhat.

HTTP - STATEFUL

- Most JavaEE app servers have this already built-in
  - JBoss EAP uses Infinispan
- Karaf can use Jetty and supply a custom session management implementation
  - Could use one based on Infinispan



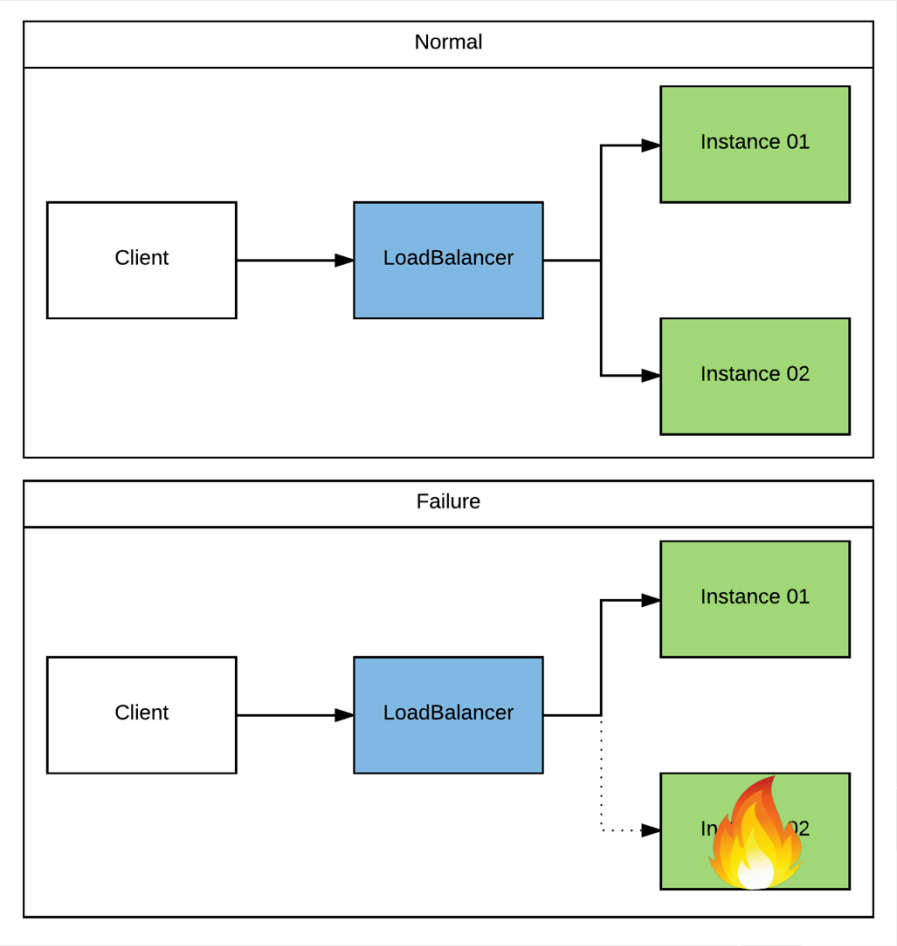
redhat.

TCP - HL7/MLLP

redhat.

HL7/MLLP

- Protocol is stateless
- Can use any TCP capable load balancer
  - Must do sticky load balancing
  - nginx works well

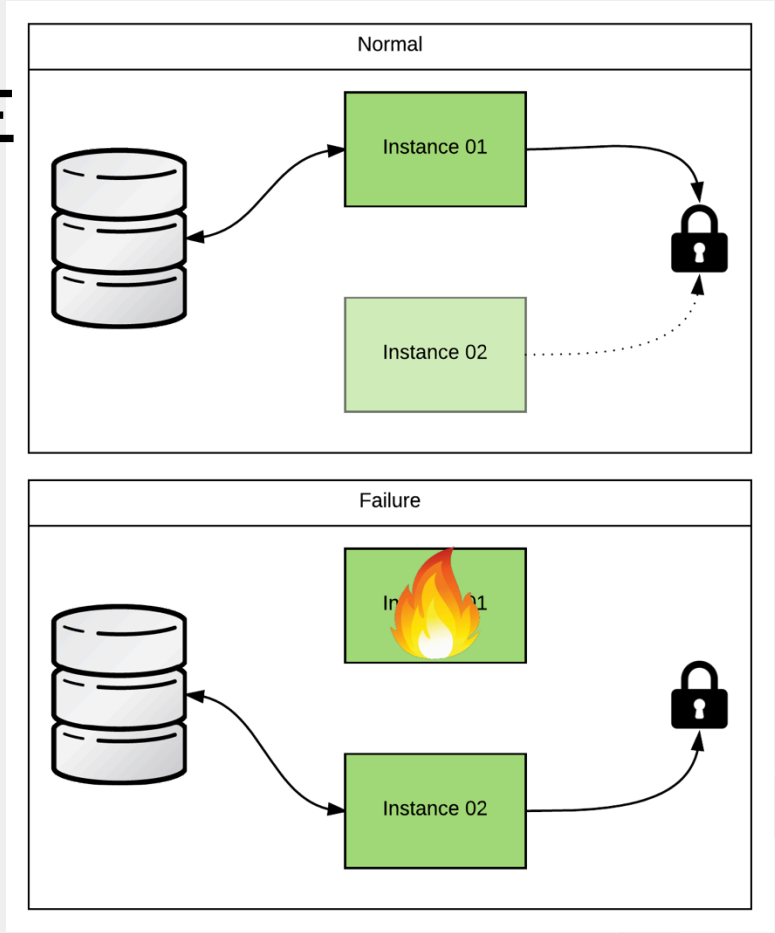


FILE/FTP

redhat.

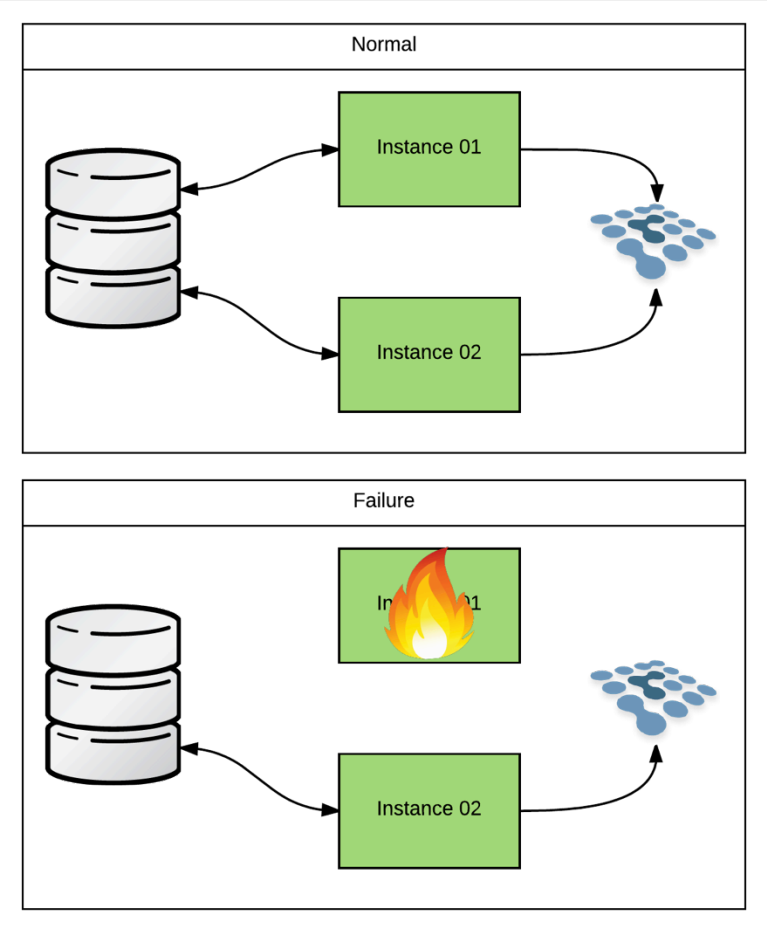
FILE/FTP - ACTIVE/PASSIVE

- Useful for batch data that come in as very few files (usually once a day)
- Need a lock for leader election
  - Can use Master component if using Fabric
  - Can create a singleton RoutePolicy



FILE/FTP - ACTIVE/ACTIVE

- Useful for data that comes as a ton of separate files
- Need a common in-progress repo
    - Can use Infinispan, DB, ...



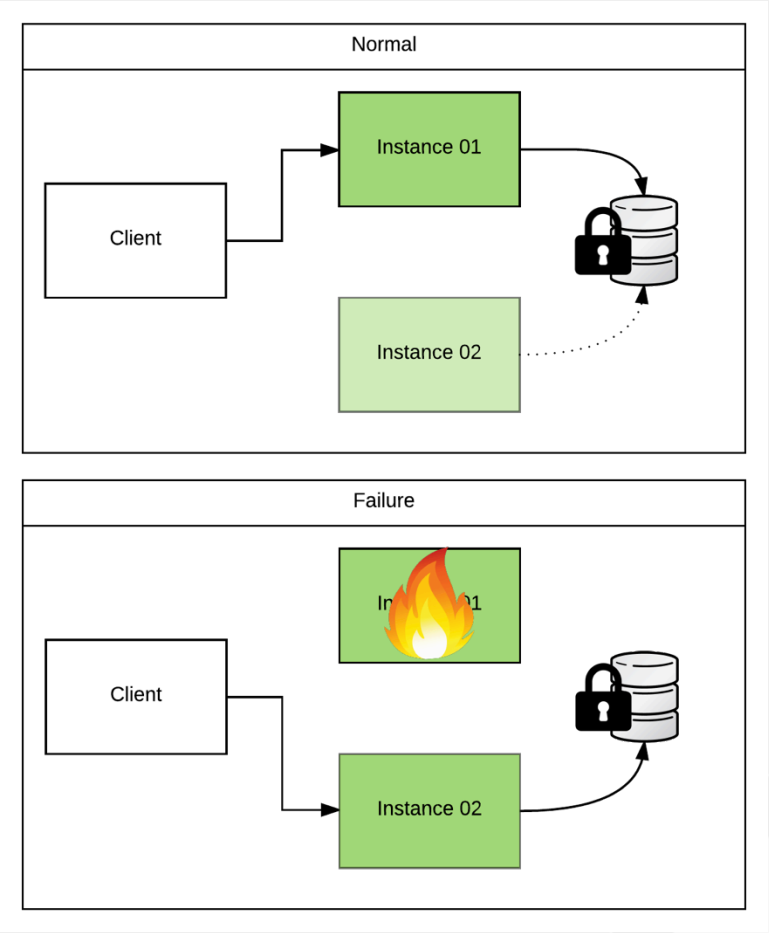
redhat.

JMS - ActiveMQ

redhat.

JMS - CLIENT

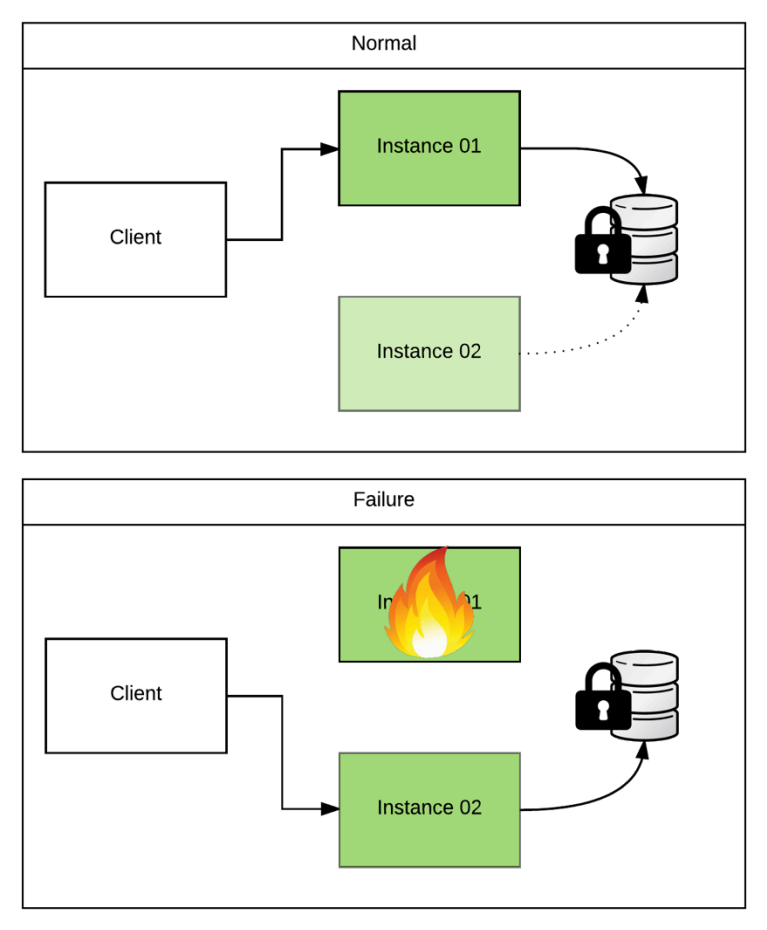
- Can just use the Failover transport
- Can failover to a slave instance, or another active master instance



redhat.

JMS - BROKER

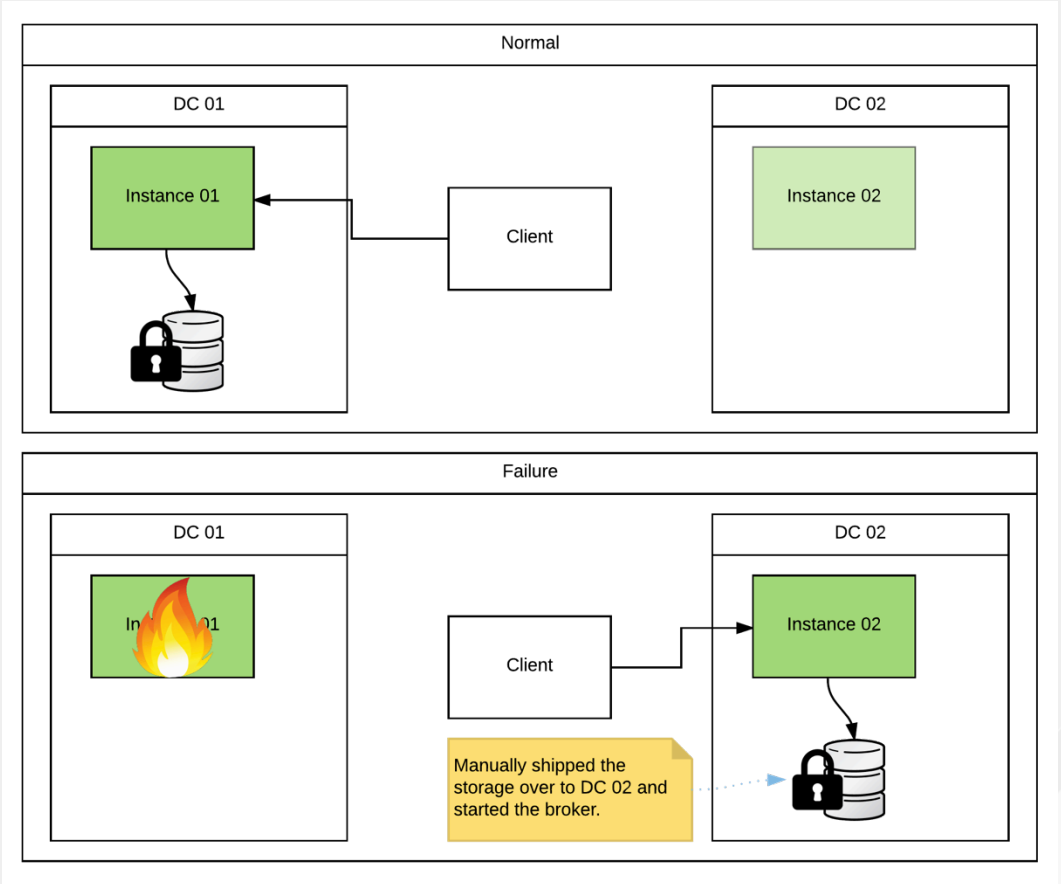
- Master/Slave is simplest
- Must have shared storage
    - Backups/replication (if any) must be fully synchronous
- Must live within a datacenter



redhat.

JMS - BROKER

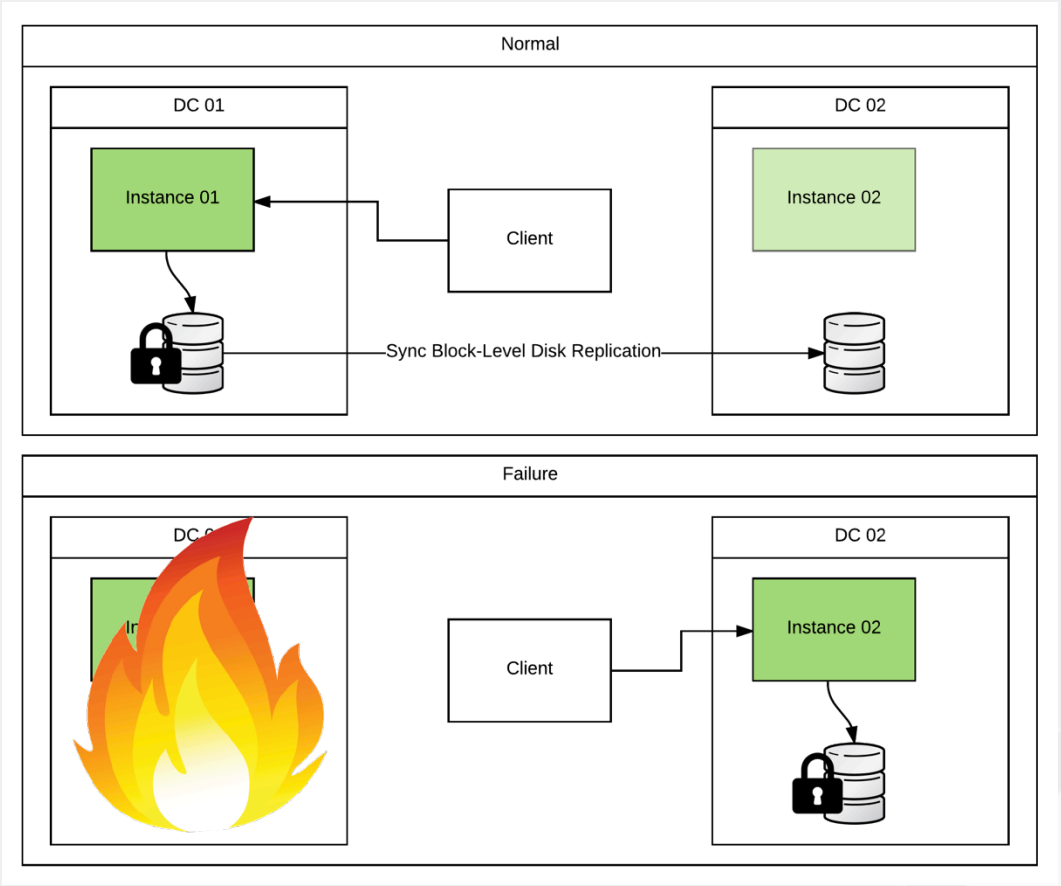
- Can wait for downed system to come back online
    - Data is not lost unless you're dealing with a catastrophic failure
- Can manually migrate the data to a warm site



redhat.

JMS - BROKER

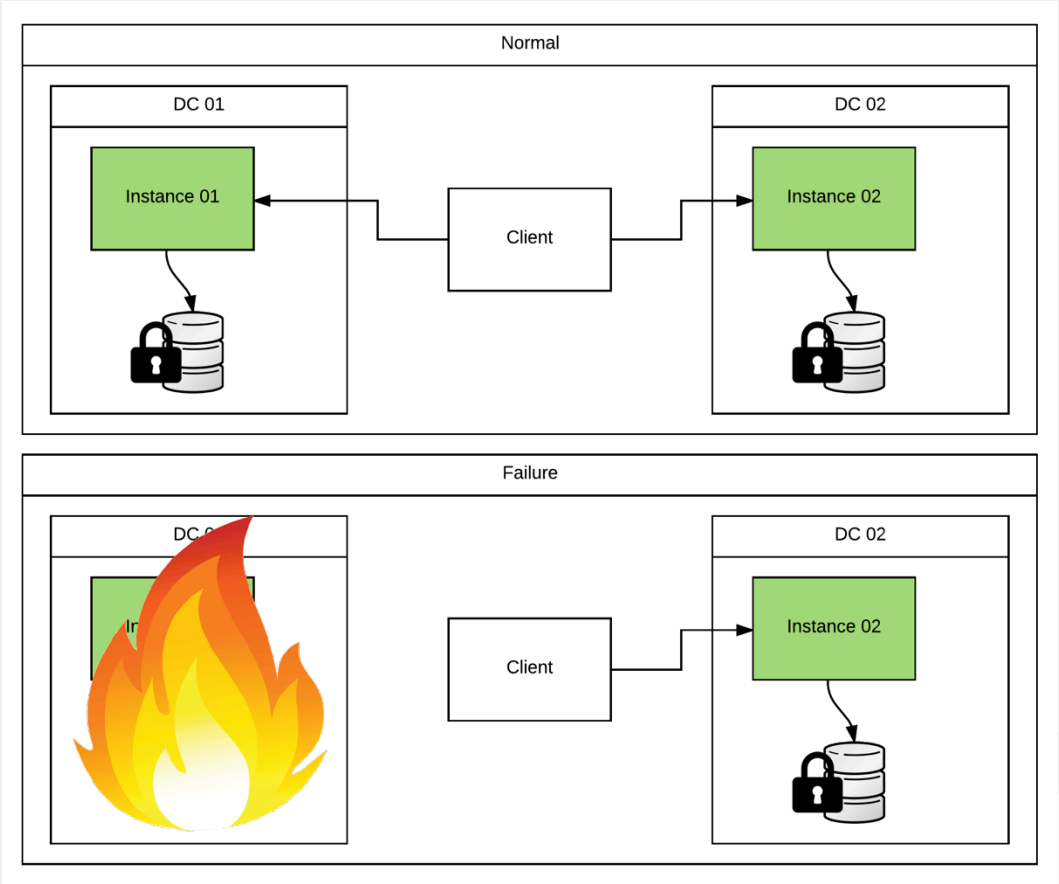
- Can use synchronous, block-level disk replication to warm site
- Must still perform manual failover since locks don't replicate
- Backup site is now the system of record



redhat.

JMS - BROKER

- Client fanout/multicast
- Duplicate data
  - Idempotent Consumer
  - Process and deal with duplicates in your application



redhat.

MORE INFORMATION

http://blog.joshdreagan.com/2016/07/28/ha_deployments_with_fuse/

redhat.

RED HAT SUMMIT

THANK YOU

plus.google.com/+RedHat

facebook.com/redhatinc

linkedin.com/company/red-hat

twitter.com/RedHatNews

youtube.com/user/RedHatVideos

LEARN. NETWORK.
EXPERIENCE
OPEN SOURCE.

redhat.