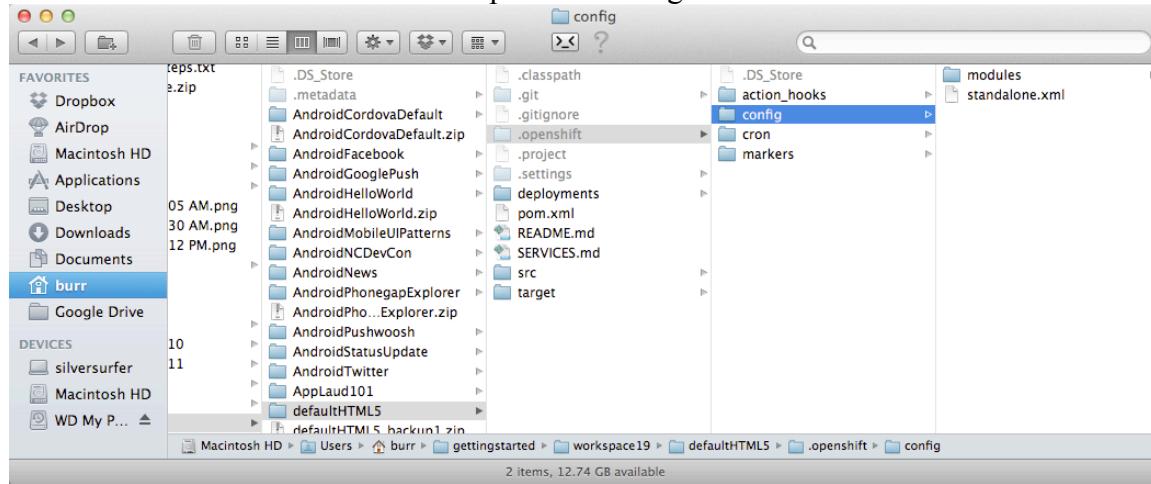


## Setting up Basic Auth on a JBoss on Openshift application

- 1) Turn on the security module in standalone.xml  
standalone.xml can be found under .openshift/config



[https://access.redhat.com/knowledge/docs/en-US/JBoss\\_Enterprise\\_Application\\_Platform/6/html-single/Administration\\_and\\_Configuration\\_Guide/index.html#chap-Securing\\_JBoss\\_Enterprise\\_Application\\_Platform](https://access.redhat.com/knowledge/docs/en-US/JBoss_Enterprise_Application_Platform/6/html-single/Administration_and_Configuration_Guide/index.html#chap-Securing_JBoss_Enterprise_Application_Platform)

Sample Code:

<https://github.com/burrsutter/cordovaendpoint>

Add the login-module for the “ApplicationRealm”

```
<login-module code="UsersRoles" flag="required">
    <module-option name="usersProperties"
        value="${env.OPENSHIFT_DATA_DIR}/application-users.properties"/>
    <module-option name="rolesProperties"
        value="${env.OPENSHIFT_DATA_DIR}/application-roles.properties"/>
</login-module>
```

Note: OPENSHIFT\_DATA\_DIR is an env variable on the openshift host. It points to a persistent storage directory that will be there between restarts of the JBoss server and your cloud instance.

From the command line, use “git commit standalone.xml” and “git push” to make this change happen on your Openshift application.

- 2) SSH into your application

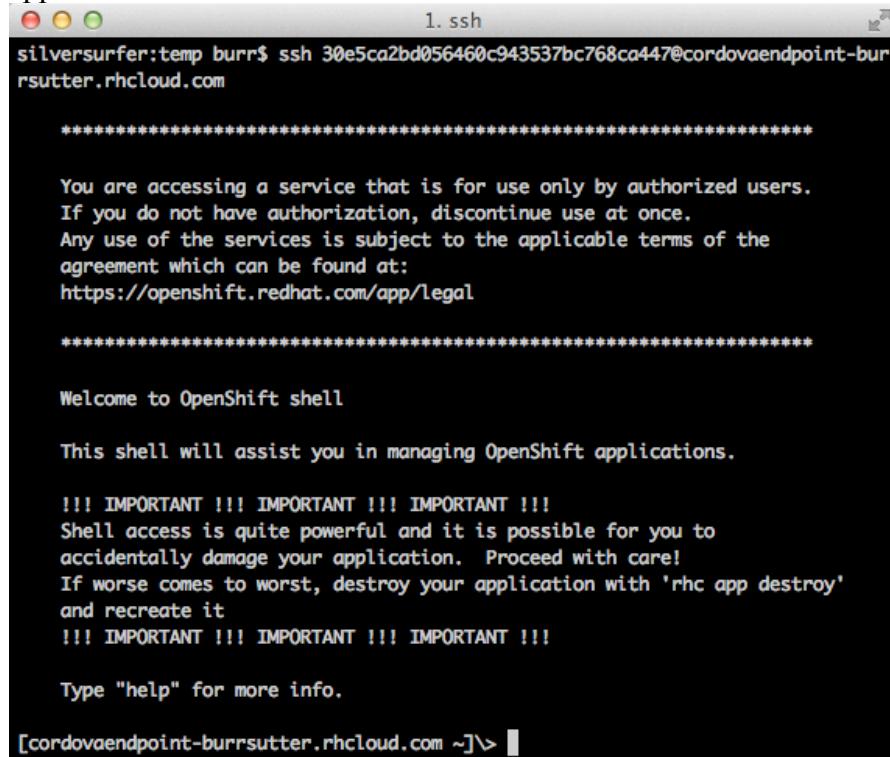
- 2.1) First login to the Openshift Management Console - and select the application needing security.

The screenshot shows the OpenShift Management Console interface. At the top, there's a navigation bar with links for 'Community', 'Developer Center', and a user account. Below the navigation is a header bar with the 'OPENSHIFT | MANAGEMENT CONSOLE' logo, a search icon, and a user icon. The main content area is titled 'All Applications'. It lists three applications: 'Nodemongo2' (status: up), 'Statushandler' (status: up), and 'Cordovaendpoint' (status: up). Each application entry includes a link to its details page. To the right of the application list are two sections: 'OPENSHIFT HELP' and 'POPULAR FAQS'. The 'OPENSHIFT HELP' section contains links to the 'Developer Center', 'OpenShift User Guide', and instructions for 'Installing OpenShift client tools on Mac OS X, Linux, and Windows'. It also includes a link to 'Sync your OpenShift repo with an existing Git repo' and a 'More help' link. The 'POPULAR FAQS' section lists questions like 'How do I start a new Forum discussion?' and 'How do I install the rhc client tools on Windows?'. At the bottom of the main content area is a blue button labeled 'ADD APPLICATION'.

Select the link “Want to log in to your application?” - it will expand to provide you the SSH details.

The screenshot shows the details for the 'Cordovaendpoint' application. The top navigation bar and header are identical to the previous screenshot. The main content area is titled 'MY APPLICATIONS / CORDOVAENDPOINT'. It shows the application name 'Cordovaendpoint' with its URL 'http://cordovaendpoint-burrsutter.rhcloud.com/'. It indicates '1 TOTAL' gear. Below this, there are sections for 'Cartridges', 'Aliases', and 'New to OpenShift?'. The 'Cartridges' section lists a single cartridge: 'JBoss Enterprise Application Pl...' (status: STARTED, gears: 1 SMALL). It also shows the 'GIT REPOSITORY' as 'ssh://30e5ca2bd056460c943537bc768ca447@cordovaendpoint-burrsutter.rhcloud.com'. A note below the repository says 'WANT TO LOG IN TO YOUR APPLICATION?' followed by instructions on how to open an SSH session. There's a text input field containing the command 'ssh 30e5ca2bd056460c943537bc768ca447@cordovaendpoint-burrsutter.rhcloud.com'. At the bottom of the cartridge section is a button labeled 'Enable Jenkins builds'. To the right of the cartridge section are sections for 'ALIASES' (no alias set) and 'NEW TO OPENSHIFT?' (with a link to 'See the getting started tips for this app'). Below these are sections for 'NEED HELP?' (links to 'OpenShift User Guide' and 'Sync your OpenShift repo with an existing Git repo') and a 'Delete this application' button. At the very bottom is a blue 'ADD CARTRIDGE' button.

2.2) Copy the ssh line and pasted it into your SSH client - on Linux and Mac, that is the normal terminal. On Windows, you will need to install & configure a SSH client application.



```
silversurfer:temp burr$ ssh 30e5ca2bd056460c943537bc768ca447@cordovaendpoint-burr.sutter.rhcloud.com

*****
You are accessing a service that is for use only by authorized users.
If you do not have authorization, discontinue use at once.
Any use of the services is subject to the applicable terms of the
agreement which can be found at:
https://openshift.redhat.com/app/legal

*****
Welcome to OpenShift shell

This shell will assist you in managing OpenShift applications.

!!! IMPORTANT !!! IMPORTANT !!! IMPORTANT !!!
Shell access is quite powerful and it is possible for you to
accidentally damage your application. Proceed with care!
If worse comes to worst, destroy your application with 'rhc app destroy'
and recreate it
!!! IMPORTANT !!! IMPORTANT !!! IMPORTANT !!!

Type "help" for more info.

[cordovaendpoint-burr.sutter.rhcloud.com ~]\>
```

2.3) cd jbosseap-6.0/jbosseap-6.0/standalone/configuration/

```
[cordovaendpoint-burr.sutter.rhcloud.com ~]\> ls
app-root cordovaendpoint git jbosseap-6.0
[cordovaendpoint-burr.sutter.rhcloud.com ~]\> cd jbosseap-6.0/jbosseap-6.0/standalone/configuration/
[cordovaendpoint-burr.sutter.rhcloud.com configuration]\> ls
application-roles.properties mgmt-users.properties standalone_xml_history
application-users.properties modules
logging.properties         standalone.xml
[cordovaendpoint-burr.sutter.rhcloud.com configuration]\>
```

2.4) cp application-roles.properties \$OPENSHIFT\_DATA\_DIR

2.5) cp application-users.properties \$OPENSHIFT\_DATA\_DIR

2.6) cd \$OPENSHIFT\_DATA\_DIR

2.7) Edit the roles file for your needs

guest=guest

remove the # which comments this line out

2.8) Edit the users file to add your user

guest=password

3) Add a jboss-web.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- Configure usage of the security domain "other" -->
<jboss-web>
    <security-domain>other</security-domain>
    <disable-audit>true</disable-audit>
</jboss-web>

```

4) Add or update your web.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="http://java.sun.com/xml/ns/javaee"
    xmlns:web="http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
    xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
    http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd" version="3.0">
    <context-param>
        <param-name>resteasy.role.based.security</param-name>
        <param-value>true</param-value>
    </context-param>

    <security-constraint>
        <web-resource-collection>
            <web-resource-name>Resteasy</web-resource-name>
            <url-pattern>/rest/secured/*</url-pattern>
        </web-resource-collection>
        <auth-constraint>
            <role-name>admin</role-name>
            <role-name>guest</role-name>
        </auth-constraint>
    </security-constraint>

    <login-config>
        <auth-method>BASIC</auth-method>
        <realm-name>UsersRoles</realm-name>
    </login-config>

    <security-role>
        <role-name>admin</role-name>
    </security-role>
    <security-role>
        <role-name>guest</role-name>
    </security-role>
</web-app>

```

5) Add RolesAllowed to your REST endpoint

```

@POST
@Consumes("application/json")
@RolesAllowed({"admin", "guest"})
public StatusUpdate create(StatusUpdate entity)
{
    em.joinTransaction();
    entity.setCreate_time(new java.util.Date());

    entity.setVersion(1);
    em.persist(entity);
    return entity;
}

```

6) Client-side JavaScript (Phonegap or mobile web)

```

var endpointURL =
"https://cordovaendpoint-
burrsutter.rhcloud.com/rest/secured/statusupdate";

function make_base_auth(user, password) {
    var tok = user + ':' + password;
    var hash = btoa(tok);
    return "Basic " + hash;
}

$.ajax({
    url: endpointURL,
    contentType: "application/json",
    dataType: "json",
    type: "POST",
    data: statusDataAsJSON,
    beforeSend: function (xhr){
        xhr.setRequestHeader('Authorization',
            make_base_auth("guest", "password"));
    },
    success: function(data) {
        refreshList();
        $("#newStatusForm")[0].reset();
    },
    error: function(error) {
        console.log(error);
    }
}); // ajax

```

