CloudStack 2.1.3 User Interface Customization

August 10, 2010

© 2010 Cloud.com, Inc. All rights reserved.
Proprietary and Confidential Information of Cloud.com, Inc. Do Not Distribute

© 2010 Cloud.com, Inc. All rights reserved. Specifications are subject to change without notice. The Cloud.com logo, Cloud.com, CloudStack, Hypervisor Attached Storage, HAS, Hypervisor Aware Network, HAN, and VMSync are trademarks or registered trademarks of Cloud.com, Inc. All other brands or products are trademarks or registered trademarks of their respective holders.

© 2010 Cloud.com, Inc. All rights reserved.
Proprietary and Confidential Information of Cloud.com, Inc. Do Not Distribute

Contents

© 2010 Cloud.com, Inc. All rights reserved.
Proprietary and Confidential Information of Cloud.com, Inc. Do Not Distribute

1 Introduction

The CloudStack™ User Interface (UI) is a rich AJAX client interface that allows you to manage all aspect of the cloud and is a complete reference implementation of the CloudStack API (for more information on the CloudStack API, see http://open.cloud.com/kb). The CloudStack UI supports three access roles.

- **Root Admin**. Access to all features of the cloud, including both virtual and physical resource management.
- **Domain-Admin**. Access to only the virtual resources of the clouds that belong to the administrator's domain.
- **User**. Access to only the features that allow management of the user's virtual instances, storage, and network.

The following document describes the various methods of user interface customization from simple branding to a complete redesign.

1.1 License

The entire user interface is released under the GNU General Public License v3 or later.

It is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or any later version. This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see http://www.gnu.org/licenses.

1.2 Support

Per our license, you are welcome to modify, add, or reuse any part of the CloudStack UI to suit your needs. However, once modified, Cloud.com™ can no longer support any defects resulting from the customization nor can Cloud.com support any upgrade process to future releases of the CloudStack Management Server.

© 2010 Cloud.com, Inc. All rights reserved.
Proprietary and Confidential Information of Cloud.com, Inc. Do Not Distribute

2 Customization

The CloudStack UI is built entirely on HTML, CSS, Javascript, and uses jQuery 1.4 as the Javascript Library for all AJAX calls, event handling, and animations. You can find the latest jQuery reference API at http://api.jquery.com/. Cloud.com recommends that any changes be made by someone with development experience in the listed above technologies. Cloud.com also recommends using a web development tool such as Firebug for Firefox to help inspect the various elements of the UI for easier modification.

2.1 Getting Started

To get started, log into your CloudStack Management Server and go to the following directory to find all the files and resources that make up the user interface.

/usr/share/cloud/management/webapps/client/

The following table will describe all the major file or files located in the directory used to build the user interface.

File	Description
index.jsp	The main HTML/JSP page. This is where all CSS and Javascripts are specified.
favicon.ico	Default Cloud.com favicon. Replace this to suit your needs.
/images/*.*	Folder that contains all the images used by the CloudStack UI.
/css/main.css	Main CSS file that contains most of the CSS definitions used by the CloudStack UI.
/css/ jquery-ui-1.8.2.custom.css	CSS file used by the jQuery UI library. All default user interface dialog CSS definition are located in this file.
/css/images	Images used by the jQuery UI library.
/css/cloud_custom.css	A list of the most common CSS definitions that are used for simple branding and look and feel changes to the default CloudStack UI.
/jsp/*.jsp	HTML/JSP pages that correspond to each major tab presented in the user interface.
/script/jquery*.js	Javascript libraries used by the CloudStack UI. You should not have to modify any of these files.
/script/date.js	Javascript library used by the CloudStack UI for dealing with date conversion. You should not

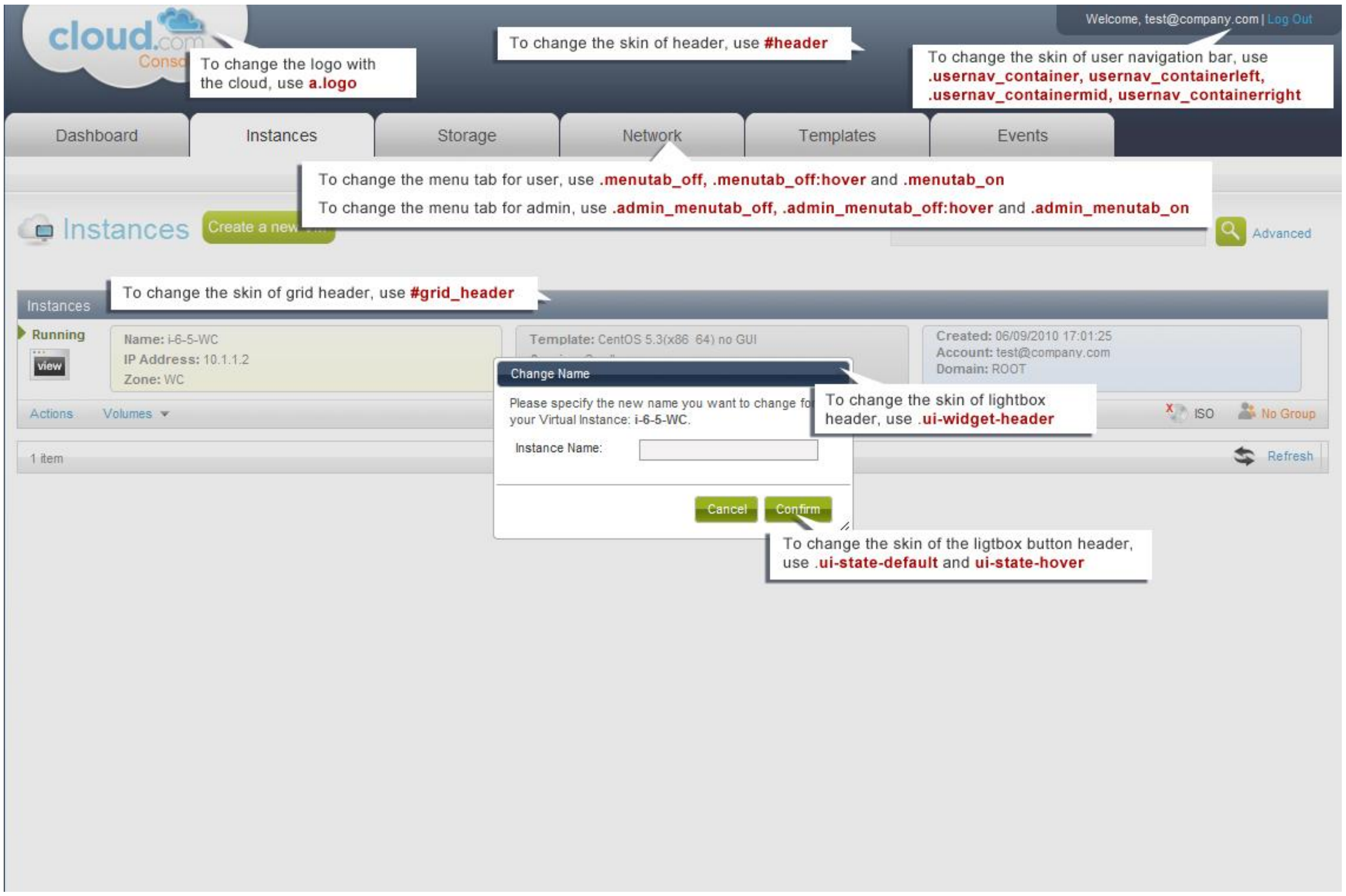
© 2010 Cloud.com, Inc. All rights reserved.
Proprietary and Confidential Information of Cloud.com, Inc. Do Not Distribute

	have to modify this file.
/script/cloud.core.callback.js	Javascript file that you can modify if you wish to integrate the CloudStack UI as a single sign-on solution with your existing website/portal.
/script/cloud.core.js	Javascript file that contains the common functions used by the CloudStack UI.
/script/cloud.core.init.js	Javascript file that contains the default initialization logic for the CloudStack UI. This is also the location where you need to specify the default API URL for AJAX calls if you decide to change the default URL.
/script/cloud.core.*.js	Javascript files that correspond to each major tab presented in the CloudStack UI.

2.2 Simple Branding

Simple branding is defined as the replacement of the header logo and all major color schemes in the CloudStack UI to match your company's logo and colors. This includes modification of the color of the header, tabs, grid header, and all dialogs. To make these changes, the only file you need to modify is /css/cloud_custom.css. You can also replace the favicon.ico and /images/cloud_logo.gif to replace the default Cloud.com images. The diagram below identifies some of the major CSS definitions which are typically modified in simple branding.

© 2010 Cloud.com, Inc. All rights reserved.
Proprietary and Confidential Information of Cloud.com, Inc. Do Not Distribute



To change the logo with the cloud, use **a.logo**

To change the skin of header, use **#header**

To change the skin of user navigation bar, use **.usernav_container, usernav_containerleft, .usernav_containermid, usernav_containerright**

To change the menu tab for user, use **.menutab_off, .menutab_off:hover** and **.menutab_on**

To change the menu tab for admin, use **.admin_menutab_off, .admin_menutab_off:hover** and **.admin_menutab_on**

To change the skin of grid header, use **#grid_header**

Name: i-6-5-WC
IP Address: 10.1.1.2
Zone: WC

Template: CentOS 5.3(x86_64) no GUI

Created: 06/09/2010 17:01:25
Account: test@company.com
Domain: ROOT

Please specify the new name you want to change fo your Virtual Instance: i-6-5-WC.

Instance Name:

To change the skin of lightbox header, use **.ui-widget-header**

To change the skin of the ligtbox button header, use **.ui-state-default** and **ui-state-hover**

2.3 Advanced Customizations

The following describes various customizations that can be applied to the user interface.

2.3.1 Changing the API URL

The default host URL on a new installation of the CloudStack Management Service is http://<server>:<port>/client. Please refer to http://tomcat.apache.org/tomcat-5.5-doc/index.html for documentation on how to change default host URL.

An API URL may need to be changed for the following reasons.

- The default configuration of the Tomcat engine has been changed to your desired URL.
- You have a load balancer or proxy server that is fronting a public host URL that is different from what is currently configured as the default API URL.

If at any point the public API URL is different than what is configured by default, you will need to reconfigure your setup by changing the API path and modifying the cloud.core.init.js file. Use the following sections to make these changes.

© 2010 Cloud.com, Inc. All rights reserved.
Proprietary and Confidential Information of Cloud.com, Inc. Do Not Distribute

2.3.1.1 To Edit the API Path

By default, the API URL as configured in the user interface is "client/api" which is relative to the default host DNS/IP. If you would like to change API path, use the following steps.

1. Edit the following file.

```
/usr/share/cloud/management/webapps/client/WEB-INF/web.xml
```

2. Within the file, find the following XML tag:

```
<servlet-mapping>
         <servlet-name>apiServlet</servlet-name>
         <url-pattern>/api/*</url-pattern>
</servlet-mapping>
```

3. Change the <url-pattern> tag to the desired API URL.

Once you have changed the API path, proceed to the next section for steps to modify the cloud.core.init.js file.

2.3.1.2 To Modify the cloud.core.init.js File

If the default API URL has changed from "client/api", use the following steps to modify the cloud.core.init.js file.

1. Edit the following file.

```
/usr/share/cloud/management/webapps/client/scripts/cloud.core.init.js
```

2. Find the following jQuery definition in the file:

```
$.ajaxSetup({
      url: "/client/api",
      dataType: "json",
      cache: false,
      error: function(XMLHttpResponse) {
            handleError(XMLHttpResponse);
      },
      beforeSend: function(XMLHttpRequest) {
            if (g_mySession == $.cookie("JSESSIONID")) {
                  return true;
            } else {
                  $("#dialog_session_expired").dialog("open");
                  return false;
            }
      }
});
```

3. Modify the URL option (this option is in bold in the above example) to your desired URL.

Once this has been modified, all subsequent AJAX calls will be made to the new URL. You may need to refresh the browser to update any cached Javascript files for the new settings to take place.

© 2010 Cloud.com, Inc. All rights reserved.
Proprietary and Confidential Information of Cloud.com, Inc. Do Not Distribute

2.3.2 Removing the Test Provisioning Tool

The default user interface also has a test provisioning tool link located on the top right of the screen once you have logged in as a ROOT admin. It is there to simulate user and domain creation. However, once you have fully integrated your user database with the CloudStack Management, that link should be disabled.

To disable it, use the following steps.

1. Edit the following file.

```
/usr/share/cloud/management/webapps/client/index.jsp
```

2. Comment out the following HTML tag in the file.

```
<div class="title_testlinks" id="launch_test" style="display:none">Launch Test
Provisioning Tool</div>
```

2.4 Single Sign-on Integration

The user interface is created entirely using the session-based CloudStack API. Once a user account has successfully logged in, a JSESSIONID cookie is sent back as part of the authorization process that can be used until the session has expired on the server. As a result, there are multiple ways that single sign-on can be integrated. Two of these methods are discussed in detail in the following sections. Please feel free to email support@cloud.com if you wish to discuss single sign-on integration that is unique to your environment if the below scenarios do not apply to you.

2.4.1 Integrating the CloudStack User Interface as an iframe

Perhaps the easiest solution of all, you can iframe the entire user interface and make sure the JSESSIONID cookie is properly set for the domain of the CloudStack UI after a successfully login. Once you get the login response, you will need to make sure the following cookies are also set (using jQuery as an example to set the cookies). You can also refer to cloud.core.callback.js for more information.

- JSESSIONID – cookie returned from the login request
- $.cookie('username', json.loginresponse.username);
- $.cookie('role', json.loginresponse.type);
- $.cookie('networktype', json.loginresponse.networktype);
- $.cookie('hypervisortype', json.loginresponse.hypervisortype);
- $.cookie('domainid', '1'); //e.g. domainid of ROOT domain is 1
- $.cookie('account', json.loginresponse.account);
- $.cookie('directattachnetworkgroupsenabled', json.loginresponse.directattachnetworkgroupsenabled);
- $.cookie('directattacheduntaggedenabled', json.loginresponse.directattacheduntaggedenabled);

Beginning with CloudStack 2.1, the user's password is no longer required for login attempts. Instead, you can now use a shared secret key between CloudStack and any outside application to sign the login request for a more secured single sign-on. The actual process of signing is very similar to the process described at http://cloud.com/community/security-0 under the "API/Secret key security section" with the following exceptions:

- You do not need to pass in the API Key
- The four parameters that must be passed in for the login command are domainId, username, timestamp, and signature.

A sample login request:

© 2010 Cloud.com, Inc. All rights reserved.
Proprietary and Confidential Information of Cloud.com, Inc. Do Not Distribute

Error! Hyperlink reference not valid.>

To get the single sign-on secret key, you must retrieve it from the CloudStack database under the configuration table for the key "security.singlesignon.key". Copy this key to the application you wish to integrate CloudStack with and follow the above instructions to sign the login command.

The timestamp parameter is simply the current system time in milliseconds. There is also a fault tolerance configurable value in the configuration table, "security.singlesignon.tolerance.millis" that can be changed to suit your preference. The timestamp passed in as part of the login request needs to be within the CloudStack Management Server time plus the fault tolerance time.

2.4.1.1 Client-side Login

To use this method of integration, you will need to send the CloudStack user account credentials to the client browser and make sure that, before the user has been directed to the iframed CloudStack user interface, the login API command has been executed successfully. A successful login will automatically set the browser cookie with the valid JSESSIONID.

Note: This method assumes that both the portal/web server and Management Server are on the same domain. The CloudStack interface will not allow itself to be iframed from another domain for security reasons. Please contact Support if you wish to have this removed to understand security implications.

Use the following steps to create a client-side single sign-on solution.

1. Open the following file.

   ```
   /usr/share/cloud/management/webapps/client/scripts/cloud.core.callbacks.js
   ```

   Within this file is a callback method, onLogoutCallback(), that is called every time the user interface detects the session has timed out for the user and requires a refresh of the use credentials. By default, if you return true, it will display the CloudStack login dialog.
2. Redirect the user interface to your own login dialog in order to implement a single sign-on solution.

There is also an example within this file that indicates a way of retrieving the user credentials from hidden fields in the HTML page, executing the login API command, and setting the cookies before the CloudStack UI takes over.

2.4.1.2 Server-side Login

To use this method of integration, you will need to execute the login API command on the server side and setting the cookie as part of the response before showing the iframed user interface. The following shows a typical use case:

1. User logs into portal.
2. User clicks on a "cloud management" link to view the iframed CloudStack user interface.
3. Your portal server receives this call and performs a server side login to the Management server with the user's credentials.
4. When successful, the response of that API call will include a JSESSIONID cookie.
5. You will read that cookie and set the cookie as part of your response to the user which will also include the HTML containing the iframed user interface.
6. You will also need to set all the cookies as described in section 2.4.1 from the login response.

© 2010 Cloud.com, Inc. All rights reserved.
Proprietary and Confidential Information of Cloud.com, Inc. Do Not Distribute

7. Make sure you edit the onLogoutCallback() method in the file "/usr/share/cloud/management/webapps/client/scripts/cloud.core.callbacks.js" to redirect the user to your login dialog when the session times out.

2.4.2 Redirecting to the CloudStack User Interface

The steps for redirecting a user to the CloudStack user interface are very similar to Section 2.4.1.1 Client-side Login. You must be sure to pass in the user credentials as part of the redirection via the querystring.

1. Modify the following file.

   ```
   /usr/share/cloud/management/webapps/client/scripts/cloud.core.callbacks.js
   ```

2. Change the script above to receive query parameters and make a login call with the correct user credentials.

© 2010 Cloud.com, Inc. All rights reserved.
Proprietary and Confidential Information of Cloud.com, Inc. Do Not Distribute