

多集群实现修改记录

1./var/lib/ambari-server/resources/views/work/ADMIN_VIEW{2.2.2.0}

(1)scripts子目录下的main.js

先使用<http://tool.oschina.net/codeformat/js>网站的在线JavaScript格式化功能格式化原main.js，然后修改main.js

将182行

```
dashboard: {  
  url: "/dashboard",  
  controller: ["$window",  
  function(e) {  
    e.location.replace("/#/main/dashboard")  
  }],  
  template: ""  
}
```

修改为

```
dashboard: {  
  url: "/dashboard/:id",  
  controller: ["$window", "$routeParams",  
  function(e, f) {  
    e.location.replace("/#/main/dashboard/" + f.id)  
  }],  
  template: ""  
}
```

将216行

```
function d() {  
  o.getStatus().then(function(t) {  
    e.cluster = t,  
    e.isLoaded = !0,  
    t && "INIT" === t.Clusters.provisioning_state && setTimeout(d, 1e3)  
  })["catch"](function(e) {  
    s.error("Cannot load cluster status", e.statusText)  
  })  
}
```

修改为

```
function d() {  
  o.getStatus().then(function(t) {  
    e.cluster = t,  
    e.isLoaded = !0,  
    t && "INIT"  
  })["catch"](function(e) {  
    s.error("Cannot load cluster status", e.statusText)  
  })  
}
```

将323行

```
function c() {  
  t.getStatus().then(function(t) {  
    e.cluster = t,  
    a.allRepos({  
      version: "",  
      cluster: {  
        options: [],  
        current: null  
      }  
    })  
  })  
}
```

```

    },
    {}).then(function(t) {
        e.totalRepos = t.itemTotal
    }),
    t && "INIT" === t.Clusters.provisioning_state && setTimeout(c, 1e3)
))[ "catch" ](function(e) {
    r.error("Cannot load cluster status", e.statusText)
})
}

```

修改为

```

function c() {
    t.getStatus().then(function(t) {
        e.cluster = t,
        a.allRepos({
            version: "",
            cluster: {
                options: [],
                current: null
            }
        }),
        {}).then(function(t) {
            e.totalRepos = t.itemTotal
        }),
        //t && "INIT" === t.Clusters.provisioning_state && setTimeout(c, 1e3)
        t && "INIT"
    ))[ "catch" ](function(e) {
        r.error("Cannot load cluster status", e.statusText)
    })
}

```

349行新增

```

e.viewClusterPermissions = function(index) {
    e.viewWhichClusterPermissions = index;
},

```

353行新增

```

e.oldClusterName = null,
e.changeWhichClusterName = null,

```

将355行

```

e.toggleEditName = function(t) {
    return t && 27 !== t.keyCode ? !1 : (e.editCluster.name = e.cluster.Clusters.cluster_name, void(e.editCluster.editingName =
!e.editCluster.editingName))
}

```

修改为

```

e.toggleEditName = function(p, i) {
    e.changeWhichClusterName = i,
    e.oldClusterName = p;
    return (e.editCluster.name = p, void(e.editCluster.editingName = !e.editCluster.editingName))
}

```

将358行

```

e.saveClusterName = function() {
    var n = e.cluster.Clusters.cluster_name,
    s = e.editCluster.name;
    t.editName(n, s).then(function(t) {

```

```

    e.cluster.Clusters.cluster_name = s,
    r.success("The cluster has been renamed to " + s + ".")
  })["catch"]((function(e) {
    r.error("Cannot rename cluster to " + s, e.data.message)
  })),
  e.toggleEditName()
}

```

修改为

```

e.saveClusterName = function() {
  var n = e.oldClusterName,

  s = e.editCluster.name;
  t.editName(n, s).then(function(t) {
    e.oldClusterName = s,

    r.success("The cluster has been renamed to " + s + "."),
    window.location.reload();
  })["catch"]((function(e) {
    r.error("Cannot rename cluster to " + s, e.data.message)
  })),
  e.toggleEditName();
}

```

将986行

```

n.getAllClusters().then(function(t) {
  t.length > 0 ? (t.forEach(function(t) {
    e.clusters.push(t.Clusters.cluster_name)
  })), e.noClusterAvailable = !1 : e.clusters.push("No Clusters"),
  e.cluster = e.clusters[0]
})

```

修改为

```

n.getAllClusters().then(function(t) {
  t.length > 0 ? (t.forEach(function(t) {
    e.clusters.push(t.Clusters.cluster_name)
  })), e.noClusterAvailable = !1 : e.clusters.push("No Clusters"),
  e.cluster = e.clusters[1]
})

```

将2330行

```

getStatus: function() {
  var r = t.defer();
  return e.get(n.baseUrl + "/clusters?fields=Clusters/provisioning_state", {
    mock: "cluster/init.json"
  }).then(function(e, t, n) {
    r.resolve(e.data.items[0])
  })["catch"]((function(e) {
    r.reject(e)
  })),
  r.promise
}

```

修改为

```

getStatus: function() {
  var r = t.defer();
  return e.get(n.baseUrl + "/clusters?fields=Clusters/provisioning_state", {
    mock: "cluster/init.json"
  }).then(function(e, t, n) {

```

```

    r.resolve(e.data.items)
  })["catch"]((function(e) {
    r.reject(e)
  })),
  r.promise
}

```

(2)views子目录下的leftNavbar.html、main.html

leftNavbar.html :

将20行

```
<div class="panel-heading"><span class="glyphicon glyphicon-cloud"></span> Clusters</div>
```

修改为

```
<div class="panel-heading"><span class="glyphicon glyphicon-cloud"></span> 集群</div>
```

将22行

```
<div class="cluster-section" ng-show="cluster">
```

修改为

```
<div class="cluster-section" ng-show="cluster" ng-repeat="singleCluster in cluster">
```

将24行

```
<h5 ng-switch-when="false">{{cluster.Clusters.cluster_name}}
```

修改为

```
<h5>{{singleCluster.Clusters.cluster_name}}
```

将25行

```
<i ng-show="cluster.Clusters.provisioning_state == 'INSTALLED'" ng-click="toggleEditName()" class="glyphicon glyphicon-edit pull-right edit-cluster-name" tooltip="Rename Cluster"></i>
```

修改为

```
<i ng-show="singleCluster.Clusters.provisioning_state == 'INSTALLED'" ng-
click="toggleEditName(singleCluster.Clusters.cluster_name, $index)" class="glyphicon glyphicon-edit pull-right edit-cluster-
name" tooltip="Rename Cluster"></i>
```

将28行

```
<form ng-keyup="toggleEditName($event)" tabindex="1" name="editClusterNameForm" class="editClusterNameForm" ng-
switch-when="true"
```

```
    ng-submit="editCluster.name !== cluster.Clusters.cluster_name && editClusterNameForm.newClusterName.$valid
&& confirmClusterNameChange()">
```

修改为

```
<form tabindex="1" name="editClusterNameForm" class="editClusterNameForm" ng-
show="changeWhichClusterName===$index" ng-submit="editCluster.name !== singleCluster.Clusters.cluster_name &&
editClusterNameForm.newClusterName.$valid && confirmClusterNameChange()">
```

将57行

```
<ul class="nav nav-pills nav-stacked" ng-show="cluster.Clusters.provisioning_state == 'INSTALLED' ">
```

```
  <li ng-class="{active: isActive('clusters.manageAccess')}">
```

```
    <a href="#/clusters/{{cluster.Clusters.cluster_name}}/manageAccess" class="permissions">Permissions</a>
```

```
  </li>
```

```
  <li><a href="#/dashboard" class="gotodashboard">Go to Dashboard</a></li>
```

```
</ul>
```

修改为

```
<ul class="nav nav-pills nav-stacked" ng-show="singleCluster.Clusters.provisioning_state == 'INSTALLED' ">
```

```
  <li ng-class="{active: viewWhichClusterPermissions===$index}" ng-click="viewClusterPermissions($index)">
```

```
    <a href="#/clusters/{{singleCluster.Clusters.cluster_name}}/manageAccess" class="permissions">权限</a>
```

```
  </li>
```

```
  <li><a href="#/dashboard/{{singleCluster.Clusters.cluster_name}}" class="gotodashboard">控制台</a></li>
```

```
</ul>
```

将63行

```
<span class="cluster-installation-progress-label" ng-show="cluster.Clusters.provisioning_state == 'INIT'"><a href="#">Cluster creation in progress...</a></span>
```

修改为

```
<span class="cluster-installation-progress-label" ng-show="singleCluster.Clusters.provisioning_state == 'INIT'"><a href="#">Cluster creation in progress...</a></span>
```

将69行

```
<li><p class="noclusters">No clusters</p></li>
```

修改为

```
<li><p class="noclusters">暂无</p></li>
```

将74行

```
<a href="#/stackVersions">Versions</a>
```

修改为

```
<a href="#/stackVersions">版本</a>
```

将81行

```
<div class="panel-heading"><span class="glyphicon glyphicon-th"></span> Views</div>
```

修改为

```
<div class="panel-heading"><span class="glyphicon glyphicon-th"></span> 视图</div>
```

将84行

```
<li ng-class="{active: isActive('views.list')}"><link-to route="views.list" class="viewslist-link">Views</link-to></li>
```

修改为

```
<li ng-class="{active: isActive('views.list')}"><link-to route="views.list" class="viewslist-link">视图</link-to></li>
```

将90行

```
<div class="panel-heading"><span class="glyphicon glyphicon-user"></span> User + Group Management</div>
```

修改为

```
<div class="panel-heading"><span class="glyphicon glyphicon-user"></span> 用户 + 组 管理</div>
```

将93行

```
<li ng-class="{active: isActive('users.list')}"><link-to route="users.list" class="userslist-link">Users</link-to></li>
```

修改为

```
<li ng-class="{active: isActive('users.list')}"><link-to route="users.list" class="userslist-link">用户</link-to></li>
```

将94行

```
<li ng-class="{active: isActive('groups.list')}"><link-to route="groups.list" class="groupslist-link">Groups</link-to></li>
```

修改为

```
<li ng-class="{active: isActive('groups.list')}"><link-to route="groups.list" class="groupslist-link">组</link-to></li>
```

main.html :

将20行

```
<h1>Welcome to Apache Ambari</h1>
```

修改为

```
<h1>欢迎使用IDAP Manager</h1>
```

将22行

```
<p ng-hide="cluster">Provision a cluster, manage who can access the cluster, and customize views for Ambari users.</p>
```

修改为

```
<p ng-hide="cluster">提供创建集群，管理访问集群的用户，自定义视图等功能。</p>
```

将23行

```
<p ng-show="cluster">Monitor your cluster resources, manage who can access the cluster, and customize views for Ambari
```

users.</p>

修改为

<p ng-show="cluster">监视群集资源，管理访问群集的用户，以及为LDAP Manager用户自定义视图。</p>

将27行

<h4 class="title">Operate Your Cluster</h4>

修改为

<h4 class="title">操作您的集群</h4>

将28行

<div class="description">Manage the configuration of your cluster and monitor the health of your services</div>

修改为

<div class="description">管理群集的配置和监视服务的健康状况。</div>

将32行

<a ng-show="cluster.Clusters.provisioning_state != 'INSTALLED'" href class="btn btn-primary permission-button" ng-disabled="cluster.Clusters.provisioning_state != 'INSTALLED'">Manage Permissions

修改为

<a ng-show="cluster.Clusters.provisioning_state != 'INSTALLED'" href class="btn btn-primary permission-button" ng-disabled="cluster.Clusters.provisioning_state != 'INSTALLED'">管理权限

将36行

<a ng-show="cluster.Clusters.provisioning_state != 'INSTALLED'" href class="btn btn-primary go-dashboard-button" ng-disabled="cluster.Clusters.provisioning_state != 'INSTALLED'">Go to Dashboard

修改为

<a ng-show="cluster.Clusters.provisioning_state != 'INSTALLED'" href class="btn btn-primary go-dashboard-button" ng-disabled="cluster.Clusters.provisioning_state != 'INSTALLED'">转到仪表板

将41行

<div ng-hide="cluster" class="col-sm-11 thumbnail">

修改为

<div ng-hide="cluster11" class="col-sm-11 thumbnail">

将42行

<h4 class="title">Create a Cluster</h4>

修改为

<h4 class="title">创建集群</h4>

将43行

<div class="description">Use the Install Wizard to select services and configure your cluster</div>

修改为

<div class="description">使用安装向导选择安装的服务，进行集群配置</div>

将45行

<div class="buttons"> Launch Install Wizard
</div>

修改为

<div class="buttons"> 启动安装向导</div>

将50行

<h4 class="title">Manage Users + Groups</h4>

修改为

<h4 class="title">管理用户 + 组</h4>

将51行

<div class="description">Manage the users and groups that can access Ambari</div>

修改为

<div class="description">管理访问LDAP Manager的用户和组</div>

将54行

<link-to route="users.list" class="btn btn-primary userslist-button">Users</link-to>

修改为

<link-to route="users.list" class="btn btn-primary userslist-button">用户</link-to>

将55行

<link-to route="groups.list" class="btn btn-primary groupslist-button">Groups</link-to>

修改为

<link-to route="groups.list" class="btn btn-primary groupslist-button">组</link-to>

将61行

<h4 class="title">Deploy Views</h4>

修改为

<h4 class="title">部署视图</h4>

将62行

<div class="description">Create view instances and grant permissions</div>

修改为

<div class="description">创建视图实例并授权</div>

将64行

<div ng-class="{active: isActive('views.list')}" class="buttons"><link-to route="views.list" class="btn btn-primary viewslist-button">Views</link-to></div>

修改为

<div ng-class="{active: isActive('views.list')}" class="buttons"><link-to route="views.list" class="btn btn-primary viewslist-button">视图</link-to></div>

(3)styles子目录下的main.css

将207行

height: 620px;

修改为

height: auto;

将1273行

.pull-up {
margin-top: -2px;

}

删除

2./usr/lib/ambari-server/web/javascripts目录下的app.js

将1496行

```
loadClusterNameSuccessCallback: function (data) {  
  if (data.items && data.items.length > 0) {  
    App.set('clusterName', data.items[0].Clusters.cluster_name);  
    App.set('currentStackVersion', data.items[0].Clusters.version);  
    this.set('isClusterNameLoaded', true);  
    App.set('isKerberosEnabled', data.items[0].Clusters.security_type === 'KERBEROS');  
  }  
}
```

```
}
```

修改为

```
loadClusterNameSuccessCallback: function (data) {
    var url = window.location.href;
    var endPos = url.lastIndexOf("/");
    var startPos = url.lastIndexOf("/", endPos - 1);
    var selectedClusterName = url.substring(startPos + 1, endPos);
    if (data.items && data.items.length > 0) {
        if (selectedClusterName == 'null') {
            window.location.href = window.location.href.replace('null', data.items[0].Clusters.cluster_name);
            App.set('clusterName', data.items[0].Clusters.cluster_name);
            this.set('isClusterNameLoaded', true);
            App.set('isKerberosEnabled', data.items[0].Clusters.security_type === 'KERBEROS');
        } else if (selectedClusterName == 'undefined') {
            window.location.href = window.location.href.replace('undefined', data.items[0].Clusters.cluster_name);
            App.set('clusterName', data.items[0].Clusters.cluster_name);
            App.set('currentStackVersion', data.items[0].Clusters.version);
            this.set('isClusterNameLoaded', true);
            App.set('isKerberosEnabled', data.items[0].Clusters.security_type === 'KERBEROS');
        } else {
            var isChangeClusterOper = false;
            var index = 0;
            for (var i = 0; i < data.items.length; i++){
                var currentClusterName = data.items[i].Clusters.cluster_name;
                if (currentClusterName == selectedClusterName) {
                    isChangeClusterOper = true;
                    index = i;
                    break;
                }
            }
        }
        if (isChangeClusterOper) {
            App.set('clusterName', data.items[index].Clusters.cluster_name);
            App.set('currentStackVersion', data.items[index].Clusters.version);
            this.set('isClusterNameLoaded', true);
            App.set('isKerberosEnabled', data.items[index].Clusters.security_type === 'KERBEROS');
        } else {
            App.set('clusterName', data.items[0].Clusters.cluster_name);
            App.set('currentStackVersion', data.items[0].Clusters.version);
            this.set('isClusterNameLoaded', true);
            App.set('isKerberosEnabled', data.items[0].Clusters.security_type === 'KERBEROS');
        }
    }
}
```

将2370行

```
preLoadHosts: function (callback) {
    var preLoadKeys = this.get('hostsPreLoadKeys');

    if (this.get('queryParams.Hosts').length > 0 && this.get('queryParams.Hosts').filter(function (param) {
        return (preLoadKeys.contains(param.key));
    }, this).length > 0) {
        this.getHostByHostComponents(callback);
        return true;
    }
    return false;
}
```



```
}
```

修改为

```
preLoadHosts: function (callback) {
  var preLoadKeys = this.get('hostsPreLoadKeys');

  if (this.get('queryParams.Hosts').length > 0 && this.get('queryParams.Hosts').filter(function (param) {
    return (preLoadKeys.contains(param.key));
  }, this).length > 0) {
    this.getHostByHostComponents(callback);
    //return true;
  }
  return false;
}
```

将40460

```
// delete any existing clusters to start from a clean slate
// before creating a new cluster in install wizard
// TODO: modify for multi-cluster support
this.getExistingClusterNames().complete(function () {
  var clusterNames = self.get('clusterNames');
  if (self.get('content.controllerName') == 'installerController' && (!App.get('testMode'))) && clusterNames.length) {
    self.deleteClusters(clusterNames);
  } else {
    self.startDeploy();
  }
})
```

修改为

```
// delete any existing clusters to start from a clean slate
// before creating a new cluster in install wizard
// TODO: modify for multi-cluster support
this.getExistingClusterNames().complete(function () {
  /**
  var clusterNames = self.get('clusterNames');
  if (self.get('content.controllerName') == 'installerController' && (!App.get('testMode'))) && clusterNames.length) {
    self.deleteClusters(clusterNames);
  } else {
    self.startDeploy();
  }
  */
  self.startDeploy();
})
```

将78933行

```
/**
 * Redirect function on sign off request.
 *
 * @param {$.Event} [context=undefined] - triggered event context
 */
logoffRedirect: function(context) {
  if (App.router.get('clusterController.isLoaded')) {
    window.location.reload();
  } else {
    this.transitionTo('login', context);
  }
}
```

修改为

```

/**
 * Redirect function on sign off request.
 *
 * @param {$.Event} [context=undefined] - triggered event context
 */
logoffRedirect: function(context) {
  /**
   if (App.router.get('clusterController.isLoaded')) {
     window.location.reload();
   } else {
     this.transitionTo('login', context);
   }
  */
  this.transitionTo('login', context);
}

```

将82369行

```

dashboard: Em.Route.extend({
  route: '/dashboard',
  .....
})

```

修改为

```

dashboard: Em.Route.extend({
  route: '/dashboard/:id',
  .....
})

```

将195667行

```

/**
 * Set visibility-status for widgets
 */
setOnLoadVisibleWidgets: function () {
  var self = this;
  if (App.get('testMode')) {
    this.translateToReal(this.get('initPrefObject'));
  } else {
    // called when first load/refresh/jump back page
    this.getUserPref(this.get('persistKey')).complete(function () {
      self.setOnLoadVisibleWidgetsComplete.apply(self);
    });
  }
}

```

修改为

```

/**
 * Set visibility-status for widgets
 */
setOnLoadVisibleWidgets: function () {
  var self = this;
  if (App.get('testMode')) {
    this.translateToReal(this.get('initPrefObject'));
  } else {
    // called when first load/refresh/jump back page
    this.getUserPref(this.get('persistKey')).complete(function () {
      self.setOnLoadVisibleWidgetsComplete.apply(self);
    });
  }
}

```

```
    window.location.href = window.location.href.replace('undefined', App.get('clusterName'));
}
```

将202569行

```
goToSection: function (event) {
  if (event.context === 'hosts') {
    App.router.set('mainHostController.showFilterConditionsFirstLoad', false);
  } else if (event.context === 'views') {
    App.router.route('views');
    return;
  } else if (event.context === 'alerts') {
    App.router.set('mainAlertDefinitionsController.showFilterConditionsFirstLoad', false);
  }
  App.router.route('main/' + event.context);
}
```

修改为

```
goToSection: function (event) {
  if (event.context === 'hosts') {
    App.router.set('mainHostController.showFilterConditionsFirstLoad', false);
  } else if (event.context === 'views') {
    App.router.route('views');
    return;
  } else if (event.context === 'alerts') {
    App.router.set('mainAlertDefinitionsController.showFilterConditionsFirstLoad', false);
  }
  if (event.context === 'dashboard') {
    App.router.route('main/' + event.context + '/' + App.get('clusterName'));
  } else {
    App.router.route('main/' + event.context);
  }
}
```

将81770行

```
if (currentClusterStatus) {
  if (self.get('installerStatuses').contains(currentClusterStatus.clusterState)) {
    self.redirectToInstaller(router, currentClusterStatus, true);
  }
  else {
    router.transitionTo('main.dashboard.index');
  }
}
```

修改为

```
if (currentClusterStatus) {
  if (self.get('installerStatuses').contains("CLUSTER_NOT_CREATED_1")) {
    self.redirectToInstaller(router, currentClusterStatus, true);
  } else {
    router.transitionTo('main.dashboard.index');
  }
}
```