

RED HAT  
**SUMMIT**

**LEARN. NETWORK.  
EXPERIENCE OPEN SOURCE.**

June 11-14, 2013  
Boston, MA



RED HAT  
**SUMMIT**

# JON Automation and Scripting

Brian Ashburn  
Senior Solutions Architect,  
Red Hat  
June 13, 2013

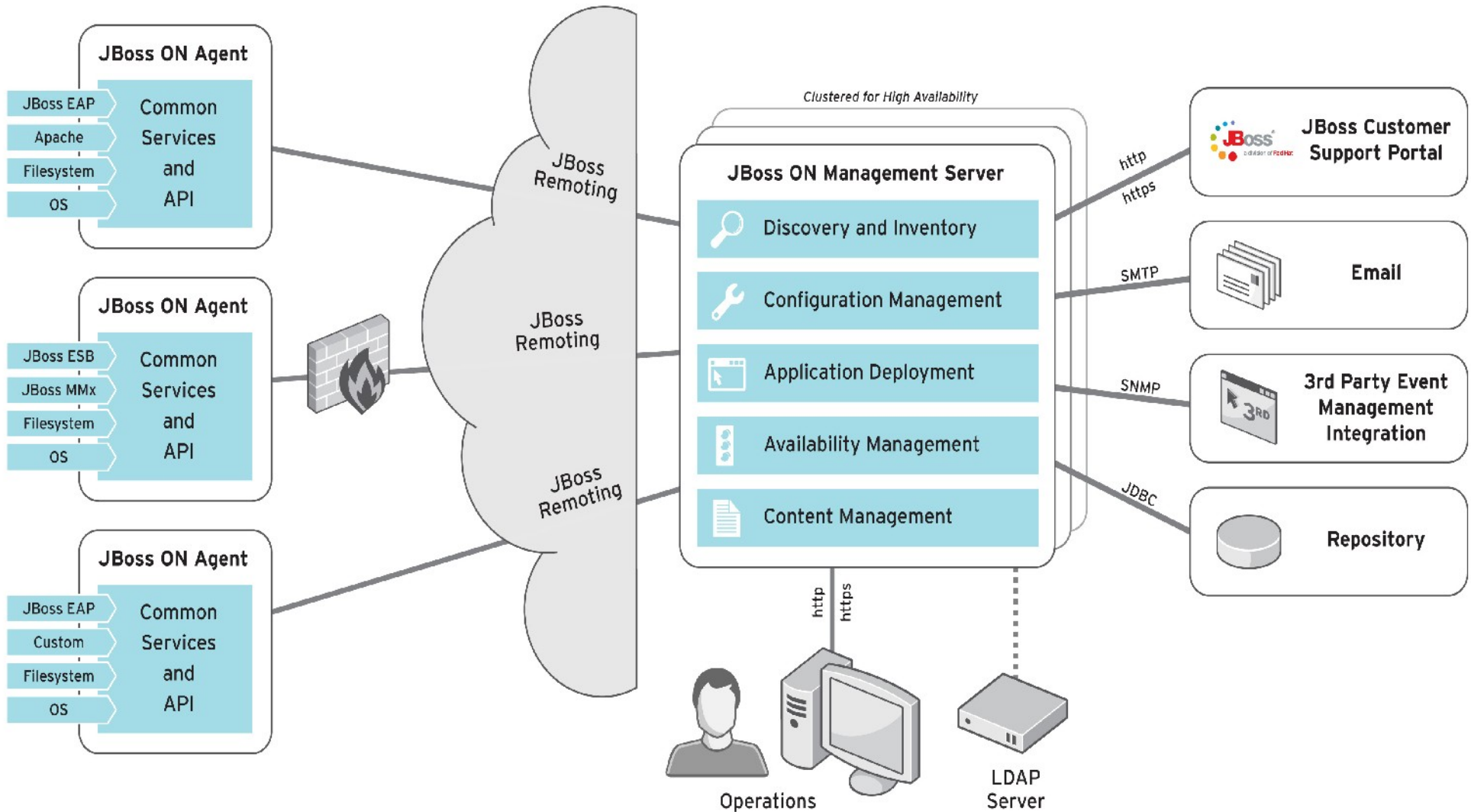
# Who I am

Senior Solutions Architect

3 years with Red Hat

Work with customers in the South East

# JBoss ON - Architecture



# Script Structure

- Find your resources
- Perform operations and configurations
- Check for success
  - Operations and deployments are asynchronous

# Finding your resources

- Resource Criteria
  - Name
  - Parent
  - Plugin or Resource Type
- Resource Group Criteria
  - Name
  - Plugin Type
  - Resource Type

# Remote API

- Resource Manager
  - Finding Resources
- Resource Factory Manager
  - Creating new resources
- Proxy Factory
  - Metrics, Operations, and Configurations
- Bundle Manager
  - Deployment and provisioning

# Sample Script – Add Datasource

```
/**
 * Name: findNamedResource
 * Args:
 *   resourceName: Resource Name or part of name to match
 *
 */
function findNamedResource(resourceName) {
    var resourceType = "JBossAS7";
    var criteria = new ResourceCriteria();
    criteria.addFilterName(resourceName);
    criteria.addFilterResourceTypeName(resourceType);
    return ResourceManager.findResourcesByCriteria(criteria);
}
```



# Sample Script – Add Datasource

```
/**
 * Name: createDatasource
 * Args:
 *   resource: EAP Resource to add datasource to
 *   dsName: Name of the datasource
 *   configs: Hash of configuration parameters for datasource
 */
function createDatasource(resource, dsName, configs) {
  var dssubsys = lookupDatasourceSubsystem(resource);
  var appType = ResourceTypeManager.getResourceTypeByNameAndPlugin("DataSource (Standalone)",
    "JBossAS7");
  return ResourceFactoryManager.createResource(
    dssubsys.id,
    appType.id,
    dsName + "-res",
    dssubsys.pluginConfiguration,
    asConfiguration(configs),
    null);
}
```

# Sample Script – Add Datasource

```
function runApp(eapName, dsName, jndiName, driverName, connectionUrl) {
    println("DataSource Name: " + dsName);
    println("JNDI Name: " + jndiName);
    println("Driver Name: " + driverName);
    println("Connection URL: " + connectionUrl);
    var configHash = new Object();
    configHash["jndi-name"] = jndiName;
    configHash["driver-name"] = driverName;
    configHash["connection-url"] = connectionUrl;
    var eapResources = findNamedResource(eapName);
    for(var i = 0; i < eapResources.size(); i++) {
        var eapRes = eapResources.get(i);
        println("Res ID:" + eapRes.id);
        var result = createDatasource(eapRes, dsName, configHash);
        println("Resource created");
        pretty.print(result);
    }
}

if(args.length < 5) {
    println("Usage: test.js eap-name ds-name jndi-name driver-name connection-url");
} else {
    var eapName = args[0];
    var dsName = args[1];
    var jndiName = args[2];
    var driverName = args[3];
    var connectionUrl = args[4];
    runApp(eapName, dsName, jndiName, driverName, connectionUrl);
}
```

# Sample Script – Deploy Bundle

```
var bundleVersion = createBundleVersion(bundleZipFile);
var destination = BundleManager.createBundleDestination(bundleVersion.bundle.id, destinationName, destination
Description, baseDirName, deployDir, group.id);

var deployment = deployBundle(bundleVersion, destination, deploymentConfiguration, null);

if (deployment.status != BundleDeploymentStatus.SUCCESS) {
    throw "Deployment wasn't successful: " + deployment;
}
```

# Sample Script – Deploy Bundle

```
/**
 * A simple function to create a new bundle version from a zip file containing
 * the bundle.
 *
 * @param pathToBundleZipFile the path to the bundle on the local file system
 *
 * @return an instance of BundleVersion class describing what's been created on
 * the RHQ server.
 */
function createBundleVersion(pathToBundleZipFile) {
    var bytes = getFileBytes(pathToBundleZipFile)
    return BundleManager.createBundleVersionViaByteArray(bytes)
}
```

# Sample Script – Deploy Bundle

```
function createBundleDestination(destinationName, description, bundleName, groupName, baseDirName, deployDir) {
    var groupCrit = new ResourceGroupCriteria;
    groupCrit.addFilterName(groupName);
    var groups = ResourceGroupManager.findResourceGroupsByCriteria(groupCrit);

    if (groups.empty) {
        throw "No group called '" + groupName + "' found.";
    }

    var group = groups.get(0);

    var bundleCrit = new BundleCriteria;
    bundleCrit.addFilterName(bundleName);
    var bundles = BundleManager.findBundlesByCriteria(bundleCrit);

    if (bundles.empty) {
        throw "No bundle called '" + bundleName + "' found.";
    }

    var bundle = bundles.get(0);

    return BundleManager.createBundleDestination(bundle.id, destinationName, description, baseDirName, deployDir, group
    .id);
}
```

# Sample Script – Deploy Bundle

```
function deployBundle(destination, bundleVersion, deploymentConfiguration, description, isCleanDeployment) {
    var destinationId = destination;
    if (typeof(destination) == 'object') {
        destinationId = destination.id;
    }

    var bundleVersionId = bundleVersion;
    if (typeof(bundleVersion) == 'object') {
        bundleVersionId = bundleVersion.id;
    }

    var deploymentConfig = deploymentConfiguration;
    if (!(deploymentConfiguration instanceof Configuration)) {
        deploymentConfig = asConfiguration(deploymentConfiguration);
    }

    var deployment = BundleManager.createBundleDeployment(bundleVersionId, destinationId, description, deploymentConfig);

    deployment = BundleManager.scheduleBundleDeployment(deployment.id, isCleanDeployment);

    var crit = new BundleDeploymentCriteria;
    crit.addFilterId(deployment.id);

    while (deployment.status == BundleDeploymentStatus.PENDING || deployment.status == BundleDeploymentStatus.IN_PROGRESS) {
        java.lang.Thread.currentThread().sleep(1000);
        var dps = BundleManager.findBundleDeploymentsByCriteria(crit);
        if (dps.empty) {
            throw "The deployment disappeared while we were waiting for it to complete.";
        }

        deployment = dps.get(0);
    }

    return deployment;
}
```

# Sample Script – Restart All EAP6 on a server

```
var criteria = new ResourceCriteria();
criteria.addFilterParentResourceCategory(ResourceCategory.PLATFORM);
criteria.addFilterParentResourceName("bashburn.redhat.com");
criteria.addFilterResourceTypeName("JBossAS7");
var resources = ResourceManager.findResourcesByCriteria(criteria);
for(var i = 0; i < resources.size(); i++) {
    var eapRes = resources.get(i);
    eapProxy = ProxyFactory.getResource(eapRes.id);
    eapProxy.restart();
}
```

# Alerts and Automation

- Alerts tell you about significant events in your environment.
- They give you the ability to resolve these events.
- Alerts are not just about notification but about action.



# Creating Alerts

- Alerts are created on the resource providing the information.
- From an alert, relative resources can be selected for operations.
- Dampening can be used to control when events are relevant.

# Additional Information

- GitHub: <https://github.com/bashburn/summit-cli>
- Twitter: @bkashburn
- Email: bashburn@redhat.com

# QUESTIONS?