

## Actionscript description and usage

Bluedata platform provides “Action Script” capability allowing authorized users to run ad hoc commands/scripts on compute nodes, anytime after a cluster is created. Actions can be submitted as a CLI commands from the WebUI, uploading a script with a set of commands, or submit a REST API with a json payload.

**Use Cases** - Actionscript use cases can be several. Some specific ones include:

1. Ability to run bash commands on one or many nodes of a cluster, in parallel. Example:
  - a. `yum -y install <python_package>`
  - b. `mkdir /usr/lib/<dir_name>`
2. Ability to view common log files on all nodes of a cluster
  - a. `tail -f /var/log/bluedata/guestconfig.log`
3. Execute commands on a specific node of the cluster instead of all nodes
4. Ability to submit hadoop,spark,python/R jobs from Bluedata UI
5. View the output of the commands from WebUI
6. Invoke commands using REST API

Sample action scripts are available at this [git](#) repository.

**Usage** - Action scripts can be run by clicking on Tenant -> Specific Cluster -> Action Script(s) -> Create. Enter a name, type commands or upload a script file, select a node group, and submit action as shown below. Generally there is one node group based on the image type. You may have more than one nodegroup if there are gateway nodes, and other BI tools. For example: One CDH cluster with master/workers is considered as one node group. Each additional edge/gateway is its own nodegroup. BI/ETL tools such as Paxata, Datameer will be in their own nodegroup. You can execute commands on all or selected nodes of specific nodegroups.

**Input Commands** - Input commands can be individual CLI commands or a script file. Example command to list files under vagent directory.

## Create Action Scripts

This feature assumes that the user has appropriate privileges to run scripts on the cluster.

Name ?	<input type="text" value="CheckVagentLogs"/>
Script source ?	<input type="text" value="Type action commands"/>
Command ?	<input type="text" value="ls -la /opt/bluedata/vagent"/>
Arguments ?	<input type="text"/>
Node Group ?	<input type="text" value="Spark 2.1.1 with Notebooks"/>
Run as a root ?	<input type="checkbox"/>

Cancel

Submit

**Generated logs** - Command runs on all nodes of the selected cluster and you can view multiple outputs, on each node, as shown below.

### Logs

#### bluedata-10.bdlocal's logs

```
total 20
drwxr-xr-x. 5 bluedata bluedata 4096 Jun  8 15:37 .
drwxr-xr-x. 5 bluedata bluedata 4096 Jun  8 15:37 ..
drwxr-x---. 3 bluedata bluedata 4096 Jun  8 15:37 guestconfig
drwxr-x---. 3 bluedata bluedata 4096 Jun  8 15:37 postconfig
drwxrwxr-x. 5 bluedata bluedata 4096 Jun  8 15:37 vagent
```

#### bluedata-9.bdlocal's logs

```
total 20
drwxr-xr-x. 5 bluedata bluedata 4096 Jun  8 15:37 .
drwxr-xr-x. 5 bluedata bluedata 4096 Jun  8 15:37 ..
drwxr-x---. 3 bluedata bluedata 4096 Jun  8 15:37 guestconfig
drwxr-x---. 3 bluedata bluedata 4096 Jun  8 15:37 postconfig
drwxrwxr-x. 5 bluedata bluedata 4096 Jun  8 15:37 vagent
```

#### bluedata-5.bdlocal's logs

```
total 20
drwxr-xr-x. 5 bluedata bluedata 4096 Jun  6 16:30 .
drwxr-xr-x. 5 bluedata bluedata 4096 Jun  6 16:30 ..
drwxr-x---. 3 bluedata bluedata 4096 Jun  6 16:30 guestconfig
drwxr-x---. 3 bluedata bluedata 4096 Jun  6 16:30 postconfig
drwxrwxr-x. 5 bluedata bluedata 4096 Jun  6 16:30 vagent
```

Close

**Node Selection in scripts** - Commands generally run on all nodes of the selected cluster. Scripts can specify subset of nodes for command execution.

Node identified by node role - This runs on the “role controller” only of a given cluster

```
#!/bin/bash

export node=`bdvcli --get node.role_id`

if [[ $node == "controller" ]]; then

    /usr/lib/spark/spark-2.1.1-bin-hadoop2.7/bin/spark-submit --class
org.apache.spark.examples.SparkPi --master spark://bluedata-4.bdlocal:7077
/usr/lib/spark/spark-2.1.1-bin-hadoop2.7/examples/jars/spark-examples_2.11-2.1.1.jar
100
fi
```

Node identified by creation sequence - This script runs on the first node created during cluster creation

```
#!/bin/bash

export node=`bdmacro node --get_self_index`

if [[ $node == 0 ]]; then

    /usr/lib/spark/spark-2.1.1-bin-hadoop2.6/bin/spark-submit --class
org.apache.spark.examples.SparkPi --master
mesos://zk://10.39.250.6:2181,10.39.250.7:2181,10.39.250.10:2181/mesos
/usr/lib/spark/spark-2.1.1-bin-hadoop2.6/examples/jars/spark-examples_2.11-2.1.1.jar
100
fi
```

**API usage** - Action scripts can also be submitted using APIs. Please refer to API samples and docs for details

**Security** - Action scripts follow authorization framework already in place for Tenant and container usage. Only authorized users can access Tenants. Within each Tenant, only users with admin privileges can run ‘sudo’ commands on clusters. For others, a user must exist or be created to allow a user action. Standard linux security for user will be enforced to run action commands.