

Package ‘AWS.tools’

August 11, 2014

Title R package to use Amazon Web Services

Version 0.0.5

Maintainer Whit Armstrong <armstrong.whit@gmail.com>

Author Whit Armstrong <armstrong.whit@gmail.com>

Description R package for Amazon Web Services EC2/S3

License GPL-3

Depends XML

SystemRequirements ec2-api-tools, s3cmd

URL <http://github.com/armstrtw/AWS.tools>

BugReports <http://github.com/armstrtw/AWS.tools/issues>

Repository CRAN

Date/Publication 2012-01-03 07:46:48

NeedsCompilation no

R topics documented:

ec2din	2
ec2stop.instances	3
ec2stop.reservation	3
s3.cmds	4
startCluster	5
stopCluster	6
Index	8

`ec2din`*Describe instances.*

Description

List and describe your instances The INSTANCE parameter is the instance ID(s) to describe. If unspecified all your instances will be returned.

Usage

```
ec2din(instance = NULL, filters = NULL, verbose = FALSE)
```

Arguments

<code>instance</code>	the instanceID to describe
<code>filters</code>	filters to apply to the command
<code>verbose</code>	show the command to be executed

Value

a dataframe enumerating the details of your instances

Author(s)

Whit Armstrong

References

<http://docs.amazonwebservices.com/AWSEC2/latest/CommandLineReference>

Examples

```
## Not run:
library(aws.tools)

c1 <- startCluster(ami="ami-a531fccc",
                  key="my-ec2-key",
                  instance.count=10,
                  instance.type="t1.micro",
                  verbose=FALSE)

ec2din()
stopCluster(c1)

## End(Not run)
```

ec2stop.instances *Stop selected running instances.*

Description

Stop selected running instances. The INSTANCE parameter is an instance ID to stop.

Usage

```
ec2stop.instances(instance.ids)
ec2terminate.instances(instance.ids)
```

Arguments

instance.ids the instances to stop

Value

a vector describing the termination status of the instances.

Author(s)

Whit Armstrong

References

<http://docs.amazonwebservices.com/AWSEC2/latest/CommandLineReference>

ec2stop.reservation *Stop or terminate all the instances in a reservation.*

Description

Stop or terminate all the instances in a reservation.

Usage

```
ec2stop.reservation(reservation.id)
ec2terminate.reservation(reservation.id)
```

Arguments

reservation.id the reservation.id to be stopped.

Value

a dataframe containing the termination status of the instances.

Author(s)

Whit Armstrong

References

<http://docs.amazonwebservices.com/AWSEC2/latest/CommandLineReference>

s3.cmds

Run S3 commands from R.

Description

Run most of the commands available in s3cmd.

Usage

```
s3.mb(bucket, bucket.location = "US", verbose = FALSE,  
      debug = FALSE)
```

```
s3.rb(bucket, recursive = FALSE, force = FALSE,  
      bucket.location = "US", verbose = FALSE, debug = FALSE)
```

```
s3.ls(bucket = NULL, bucket.location = "US",  
      human.readable.sizes = TRUE,  
      list.md5 = FALSE, verbose = FALSE, debug = FALSE)
```

```
s3.la(bucket.location = "US", human.readable.sizes = TRUE,  
      list.md5 = FALSE, verbose = FALSE, debug = FALSE)
```

```
s3.put(x, bucket, bucket.location = "US", verbose = FALSE,  
      debug = FALSE, encrypt = FALSE)
```

```
s3.get(bucket, bucket.location = "US", verbose = FALSE,  
      debug = FALSE)
```

```
s3.del(bucket, bucket.location = "US",  
      human.readable.sizes = TRUE, list.md5 = FALSE,  
      verbose = FALSE, debug = FALSE)
```

```
s3.info(bucket, bucket.location = "US",  
      human.readable.sizes = TRUE, list.md5 = FALSE,  
      verbose = FALSE, debug = FALSE)
```

```
s3.du(bucket, bucket.location = "US",  
      human.readable.sizes = TRUE, list.md5 = FALSE,  
      verbose = FALSE, debug = FALSE)
```

Arguments

x	an R object to upload.
bucket	S3 bucket
bucket.location	Specify datacentre where to create the bucket. Possible values are US (default) or EU.
human.readable.sizes	Print sizes in human readable form.
list.md5	Include MD5 sums in bucket listings (only for ls command).
verbose	Enable verbose output.
debug	Enable debug output.
recursive	Recursive upload, download or removal. When used with del it can remove all the files in a bucket.
force	Force overwrite and other dangerous operations. Can be used to remove a non-empty buckets with <code>s3cmd rb -force s3://bkt</code>
encrypt	Use GPG encryption to protect stored objects from unauthorized access. See http://s3tools.org/s3cmd-public for details about encryption.

Value

Return value depends on command used. with the exception of get, most commands return as text the reply from the call to the underlying s3cmd.

Author(s)

Whit Armstrong

References

<http://s3tools.org/s3cmd>

startCluster	<i>Start an EC2 cluster.</i>
--------------	------------------------------

Description

Start a group of instances which serve as a compute cluster.

Usage

```
startCluster(ami,
             key,
             instance.count,
             instance.type,
             verbose = FALSE)
```

Arguments

ami	machine-id
key	securtiy key
instance.count	number of instances requested
instance.type	type of instance requested
verbose	print verbose output

Value

a cluster object encompassing the reservationID and instanceID's

Author(s)

Whit Armstrong

References

<http://docs.amazonwebservices.com/AWSEC2/latest/CommandLineReference>

Examples

```
## Not run:
library(aws.tools)
cl <- startCluster(ami="ami-a531fccc",
                  key="my-ec2-key",
                  instance.count=10,
                  instance.type="t1.micro",
                  verbose=FALSE)

stopCluster(cl)

## End(Not run)
```

stopCluster

Stop or terminate instances in a cluster.

Description

Stop or terminate the machine instances in a cluster object.

Usage

```
stopCluster(cluster)
terminateCluster(cluster)
```

Arguments

cluster	a cluster object describing the reservationID and machine instances composing the cluster
---------	---

Value

a dataframe listing the termination status of the instances in the cluster.

Author(s)

Whit Armstrong

References

<http://docs.amazonwebservices.com/AWSEC2/latest/CommandLineReference>

See Also

[startCluster](#)

Examples

```
## Not run:
library(aws.tools)

cl <- startCluster(ami="ami-a531fccc",
                  key="my-ec2-key",
                  instance.count=10,
                  instance.type="t1.micro",
                  verbose=FALSE)

stopCluster(cl)

## End(Not run)
```

Index

*Topic **utilities**

- ec2din, 2
- ec2stop.instances, 3
- ec2stop.reservation, 3
- startCluster, 5
- stopCluster, 6

- ec2din, 2
- ec2stop.instances, 3
- ec2stop.reservation, 3
- ec2terminate.instances
 - (ec2stop.instances), 3
- ec2terminate.reservation
 - (ec2stop.reservation), 3

- s3.cmds, 4
- s3.del (s3.cmds), 4
- s3.du (s3.cmds), 4
- s3.get (s3.cmds), 4
- s3.info (s3.cmds), 4
- s3.la (s3.cmds), 4
- s3.ls (s3.cmds), 4
- s3.mb (s3.cmds), 4
- s3.put (s3.cmds), 4
- s3.rb (s3.cmds), 4
- startCluster, 5, 7
- stopCluster, 6

- terminateCluster (stopCluster), 6