
Cumulus Documentation

Release 1.4.0

Skymill Solutions

Sep 27, 2017

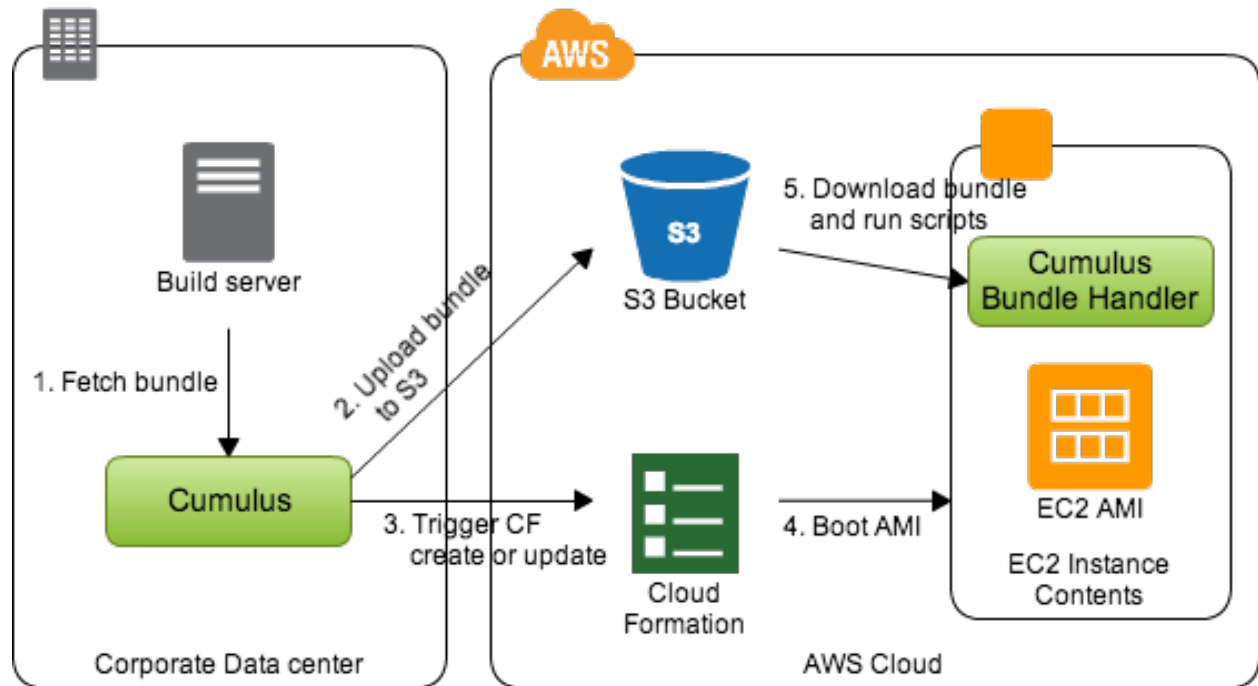
1	Table of contents	3
1.1	Overview	3
1.1.1	Introduction	3
1.1.2	Basic concepts	3
1.1.3	Deployment workflow	3
1.1.4	Supported platforms	4
1.2	Installation and upgrading	5
1.2.1	Requirements	5
1.2.2	Installing cumulus	5
1.2.3	Upgrading cumulus	5
1.2.4	Installing cumulus-bundle-handler	5
1.2.5	Upgrading cumulus-bundle-handler	5
1.3	Cumulus	5
1.3.1	Cumulus configuration	5
1.3.2	Command line options	8
1.3.3	Stack naming	8
1.3.4	Deploying an environment	9
1.3.5	Undeploying (deleting) an environment	9
1.3.6	Note on environment specific configuration	9
1.4	Cumulus Bundle Handler	9
1.4.1	Init scripts	10
1.4.2	Configuration file	10
1.4.3	Logging	11
1.5	CloudFormation template examples	11
1.5.1	Linux host with an Auto Scaling Group	11
1.5.2	Windows instance in a VPC	16
1.6	Cumulus release notes	21
1.6.1	1.4.0	21
1.6.2	1.3.2	21
1.6.3	1.3.1	21
1.6.4	1.3.0	21
1.6.5	1.2.2	21
1.6.6	1.2.1	21
1.6.7	1.2.0	21
1.6.8	1.1.3	22
1.6.9	1.1.2	22

1.6.10	1.1.0	22
1.6.11	1.0.3	22
1.6.12	1.0.2	22
1.6.13	1.0.1	22
1.6.14	1.0.0 (First open source release)	23
1.6.15	0.8.0	23
1.6.16	0.7.0	23
1.6.17	0.6.4	23
1.6.18	0.6.3	24
1.6.19	0.6.2	24
1.6.20	0.6.1	24
1.6.21	0.6.0	24
1.6.22	0.5.0	25
1.6.23	0.4.0	25
1.6.24	0.3.1	25
1.6.25	0.3.0	25
1.6.26	0.2.3	26
1.6.27	0.2.2	26
1.6.28	0.2.1	26
1.6.29	0.2.0	26
1.6.30	0.1.1	26
1.6.31	0.1.0	26
1.7	Cumulus Bundle Handler release notes	26
1.7.1	1.0.5	26
1.7.2	1.0.4	27
1.7.3	1.0.3	27
1.7.4	1.0.2	27
1.7.5	1.0.1	27
1.8	Bug reports and feature requests	27
1.9	License	27

Cumulus is a deployment suite used to deploy and manage environments built with AWS CloudFormation. Cumulus will help you bundle your code and configuration and unpack the bundle to new instances on CloudFormation.

The target for the Cumulus project is to make cloud deployments scriptable, reliable and repeatable. It is of great importance for productivity and product stability that you are able to release often and with as few manual steps as possible.

Cumulus consists of two parts, `cumulus` which is used to manage the software bundling and deployment and the `cumulus-bundle-handler` which handles the software installation on the target servers.



Overview

Introduction

The target for the Cumulus project is to make cloud deployments scriptable, reliable and repeatable. It is of great importance for productivity and product stability that you are able to release often and with as few manual steps as possible.

Cumulus consists of two parts, `cumulus` which is used to manage the software bundling and deployment and the `cumulus-bundle-handler` which handles the software installation on the target servers.

Basic concepts

Cumulus is built around three main concepts:

- An **environment** references a whole environment and all it's CloudFormation stacks. It holds together information about the AWS account, which stacks to deploy and in which version.
- A **stack** is simply a CloudFormation stack.
- A **bundle** is a `.zip` file with code and configuration to unpack to instances.

Deployment workflow

Deployments with Cumulus can take many shapes depending on your project needs. But a common pattern can look like the example below.

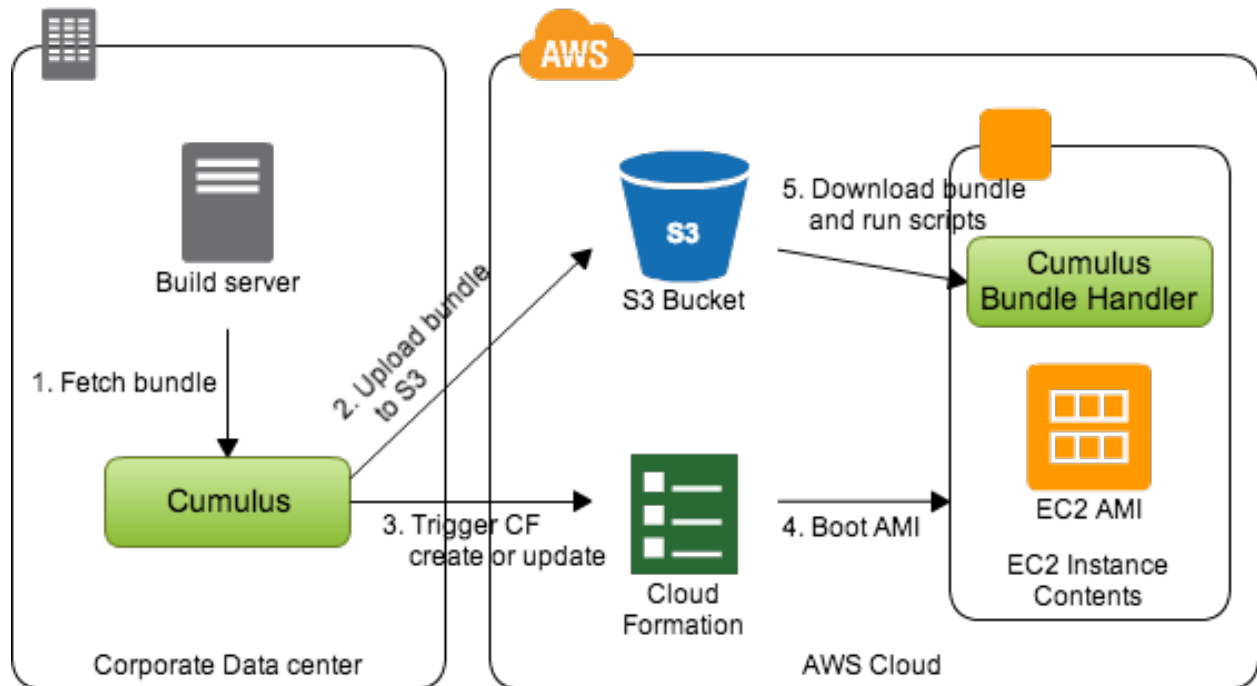
If your build server delivers a package, then Cumulus can use that for deployment. The procedure would be something like this:

1. The build server builds the software
2. The build server places a `.zip` file on the file system

3. `cumulus` picks up the software package - called a **bundle** in Cumulus - and rename it according to the given version and target environment
4. `cumulus` uploads the bundle to AWS S3
5. `cumulus` initiates a AWS CloudFormation CREATE or UPDATE (depending on whether or not the stack exists previously)
6. The EC2 instance has `cumulus-bundle-handler` installed
7. `cumulus-bundle-handler` will download the bundle from S3
8. `cumulus-bundle-handler` will deploy the bundle to the instance
9. `cumulus-bundle-handler` will restart necessary services and run any configured deployment hooks
10. Deployment is now completed

You can also use `cumulus` to build your bundle, if you don't get a pre-packaged version of you software from the build server. `cumulus` can then take a certain path on the file system and convert it to a bundle.

Here's an image of an example workflow. Please note that this is a very simple example with only one server, but there are no limitations in terms of what infrastructure you could set up with Cumulus.



Supported platforms

- Cumulus supports Linux, Mac OS X and Windows
- Cumulus Bundle Handler supports Linux and Windows (and likely Mac OS X, but testing is needed)

The rest of the work is down within the AWS CloudFormation template. Please have a look at our [CloudFormation template examples](#).

Installation and upgrading

Requirements

For now, Cumulus requires Python 2.7.

Installing `cumulus`

You can install Cumulus via PyPI with `pip`.

```
pip install cumulus
```

`cumulus` is now available as a global command on your machine.

Upgrading `cumulus`

Cumulus can be updated using `pip`.

```
pip install -U cumulus
```

Installing `cumulus-bundle-handler`

You can install Cumulus Bundle Handler via PyPI with `pip`.

```
pip install cumulus-bundle-handler
```

`cumulus` is now available as a global command on your machine.

Upgrading `cumulus-bundle-handler`

Cumulus Bundle Handler can be updated using `pip`.

```
pip install -U cumulus-bundle-handler
```

Cumulus

Cumulus (`cumulus`) is used for software bundling and for managing CloudFormation deployments.

Cumulus configuration

Example `cumulus.conf`

All configuration is read from `/etc/cumulus.conf`, `~/.cumulus.conf` and `./cumulus.conf` in order. You can also specify a custom configuration file using `--config`.

Below is a full example configuration:

```
[general]
log-level: info

[environment: stage]
access-key-id: <AWS ACCESS KEY>
secret-access-key: <AWS SECRET KEY>
bucket: se.skymill.bundles
region: eu-west-1
stacks: full
bundles: webserver, database, app
version: 1.0.0-SNAPSHOT
pre-deploy-hook: /path/to/script
post-deploy-hook: echo "Yay" > ~/test.log
stack-name-prefix: myproject
#stack-name-suffix: myproject

[stack: full]
template: /Users/sebastian/tmp/hosts/webserver.json
disable-rollback: true
#timeout-in-minutes: 10
parameters:
    version = 1.1.0,
    test tag = my test value
    key = value
tags:
    project = Example project

[bundle: webserver]
pre-bundle-hook: git clone git://git.example.com/my.git
post-bundle-hook: rm -rf my
paths:
    /Users/sebastian/tmp/hosts/webserver
    /Users/sebastian/tmp/code/wordpress

[bundle: database]
pre-bundle-hook: /path/to/script
paths: /Users/sebastian/tmp/hosts/database
path-rewrites:
    /wordpress -> /var/www/wordpress
    /nginx -> /etc/nginx

[bundle: app]
pre-built-bundle: /Users/sebastian/build/app.zip
```

Section: general

The configuration options here modify the behavior of Cumulus features that are not environment or stack specific.

Option	Type	Required	Comment
log-level	String	No	Log level (one of: debug, info, warning and error)
include	CommaSeparatedList	No	List of config files to include

Section: environment

The following configuration options are available under `[environment: env_name]`. The `env_name` is the identifier for the environment.

Option	Type	Required	Comment
<code>access-key-id</code>	String	Yes	AWS access key
<code>secret-access-key</code>	String	Yes	AWS secret access key
<code>bucket</code>	String	Yes	AWS S3 bucket to store bundles in
<code>region</code>	String	Yes	AWS region name, e.g. <code>us-east-1</code>
<code>stacks</code>	List	Yes	List of stack names to deploy
<code>bundles</code>	List	Yes	List of bundles to build and upload
<code>version</code>	String	Yes	Environment version number
<code>pre-deploy-hook</code>	String	No	Command to execute before deployment
<code>post-deploy-hook</code>	String	No	Command to execute after deployment
<code>stack-name-prefix</code>	String	No	Prepend a prefix to the stack name
<code>stack-name-suffix</code>	String	No	Append a suffix to the stack name

Section: stack

Options for the `[stack: stack_name]` configuration section.

Option	Type	Required	Comment
<code>template</code>	String	Yes	Path to local CloudFormation JSON file
<code>disable-rollback</code>	Boolean	No	Should CloudFormation rollbacks be disabled? Default: <code>false</code>
<code>timeout-in-minutes</code>	Integer	No	Set a CloudFormation creation timeout
<code>parameters</code>	Line sep. string	Yes	Parameters to send to the CloudFormation template. Should be on the form <code>key = value</code> . Each parameter is separated by a new line.
<code>tags</code>	Line sep. string	No	CloudFormation tags to add to the stack

Section: bundle

Options for the `[bundle: bundle_name]` configuration section.

Option	Type	Required	Comment
<code>pre-bundle-script</code>	String	No	Command to execute before bundling
<code>post-bundle-script</code>	String	No	Command to execute after bundling
<code>paths</code>	Line sep. string	Yes	Paths to include in the bundle. Each path should be declared on a new line.
<code>path-rewrite</code>	Line sep. string	No	Replace parts of the paths. Will make a string replace before bundling. Format: <code>/example/path/ -> /</code> (will replace <code>/example/path/</code> will be replaced by <code>/</code>)
<code>pre-build-bundle</code>	String	No	Path to a pre-built bundle. This option will make the <i>paths</i> redundant.

Command line options

Below is a listing of the cumulus command line options.

```
usage: cumulus [-h] [-e ENVIRONMENT] [-s STACKS] [--version VERSION]
              [--parameters PARAMETERS] [--config CONFIG] [--cumulus-version]
              [--force] [--bundle] [--deploy] [--deploy-without-bundling]
              [--redeploy] [--events] [--list] [--outputs]
              [--validate-templates] [--undeploy]

Cumulus cloud management tool

optional arguments:
  -h, --help            show this help message and exit

General options:
  -e ENVIRONMENT, --environment ENVIRONMENT
                        Environment to use
  -s STACKS, --stacks STACKS
                        Comma separated list of stacks to deploy. Default
                        behavior is to deploy all stacks for an environment
  --version VERSION     Environment version number. Overrides the version
                        value from the configuration file
  --parameters PARAMETERS
                        CloudFormation parameters. On the form: stack_name:parameter_name=value,stack_name=parameter_name=value
  --config CONFIG       Path to configuration file. Can be a comma separated
                        list of files.
  --cumulus-version     Print cumulus version number
  --force               Skip any safety questions

Actions:
  --bundle              Build and upload bundles to AWS S3
  --deploy              Bundle and deploy all stacks in the environment
  --deploy-without-bundling
                        Deploy all stacks in the environment, without bundling
                        first
  --redeploy            Undeploy and deploy the stack(s). Implies bundling.
  --events              List events for the stack
  --list                List stacks for each environment
  --outputs             Show output for all stacks
  --validate-templates  Validate all templates for the environment
  --undeploy            Undeploy (delete) all stacks in the environment. Use
                        --force to skip the safety question.
```

Stack naming

CloudFormation stacks must have a unique name. Cumulus will therefore combine the environment name and the stack name from the configuration. The pattern is <environment>-<stack_name>. So, if your environment is called production and your stack is webserver then your CloudFormation stack will be named production-webserver.

You can also optionally add a prefix or suffix to the stack name using the stack-name-prefix and/or stack-name-suffix options.

Deploying an environment

To deploy (create or update) an environment run the following:

```
cumulus --environment production --deploy
```

Note!

When running on Windows, you'll need to invoke Cumulus with `python cumulus`

If you only want to deploy a certain stack, use the `--stacks` option.

Undeploying (deleting) an environment

If you want to remove a whole environment, you'll undeploy it by running:

```
cumulus --environment production --undeploy
```

WARNING! This will delete all resources defined in your CloudFormation template

Note!

When running on Windows, you'll need to invoke Cumulus with `python cumulus`

Note on environment specific configuration

Cumulus supports environment specific configuration, if you are using `cumulus` to create your bundles. This is useful if you have one `httpd.conf` for production purposes and another for testing. To have files that should only be included in specific environments, prefix them with `__cumulus-environment__filename`.

So for example: `__cumulus-production__nginx.conf` is the `nginx.conf` for the production environment.

Cumulus Bundle Handler

The Cumulus Bundle Handle is a Python script that should reside on each server in the environment. The script is responsible for

- Downloading and extracting the correct bundles for the host
- Running pre and post deployment scripts on the host, e.g. to restart relevant services and trigger various deployment hooks

The bundles are generated via the `cumulus` command (or in your build server) and uploaded to S3. Cumulus Bundle Handler will then download the bundle when the script is triggered (usually by a CloudFormation `create` or `update`).

Init scripts

The Cumulus Bundle Handler supports scripts to be executed:

- Before bundle extraction (good for stopping services etc)
- After bundle extraction (good for starting services)
- Both before and after extraction (typically cleaning jobs)

All init script should reside in `/etc/cumulus-init.d` on Linux systems and in `C:\cumulus\init.d` on Windows systems and must be executable.

- Scripts starting with `K` (capital K) are executed *before* the bundle is extracted
- Scripts starting with `S` (capital S) are executed *after* the bundle is extracted
- Scripts starting with anything else than `S` or `K` are executed both before and after the bundle is extracted

Configuration file

The configuration file for Cumulus Bundle Handler should reside on your EC2 instances under `/etc/cumulus/metadata.conf` on Linux systems and under `C:\cumulus\conf\metadata.conf` on Windows systems. It recommended to serve it to that location using CloudFormation [AWS::CloudFormation::Init](#).

metadata.conf example

The Cumulus Bundle Handler relies on a configuration file called `metadata.conf`. Here's an example configuration file.

```
[metadata]
log-level: INFO
access-key-id: <AWS_ACCESS_KEY>
secret-access-key: <AWS_SECRET_KEY>
region: eu-west-1
bundle-bucket: com.example.bundles
environment: stage
bundle-types: webserver
bundle-extraction-paths:
    generic -> /etc/example
    webserver -> /
version: 1.0.0-SNAPSHOT
```

Configuration options for `metadata.conf`

Option	Type	Required	Comment
<code>access-key-id</code>	String	Yes	AWS access key
<code>secret-access-key</code>	String	Yes	AWS secret access key
<code>region</code>	String	Yes	AWS region name, e.g. <code>us-east-1</code>
<code>bucket</code>	String	Yes	AWS S3 bucket to fetch bundles from
<code>environment</code>	String	Yes	Environment name
<code>version</code>	String	Yes	Environment version to apply
<code>bundle-types</code>	List	Yes	Bundle names to apply to this host
<code>bundle-extract-dir</code>	Non-path line sep. list	No	Decide in which parent directory a bundle shall be extracted. Default is <code>/</code> on Linux and Mac OS X and <code>C:</code> on Windows systems. Expecting absolute paths
<code>log-level</code>	String	No	Log level for the bundle handler

Logging

Cumulus Bundle Handler will log to `/var/log/cumulus-bundle-handler.log` on Linux systems and to `C:\cumulus\logs\cumulus-bundle-handler.log` on Windows systems.

This log file can be really helpful when trying to debug your deployments.

CloudFormation template examples

We are utilizing `cfn-init` to populate objects on the target instances. You will need to ensure that the [CFN helper scripts](#) are installed on the servers.

Also, you will need to have Python 2.7 as well as `cumulus-bundle-handler` on all servers.

If you are running on a Windows server, please make sure that the `UserData` is read on boot. You should take a look at [Bootstrapping AWS CloudFormation Windows Stacks](#) and [Configuring a Windows Instance Using the EC2Config Service](#).

Linux host with an Auto Scaling Group

Here's an example CloudFormation JSON document for a webserver in an Auto Scaling Group with Cumulus configured.

```
{
  "Description" : "Webservers for Cumulus test",
  "Parameters" : {
    "KeyName" : {
      "Description" : "AWS key to use",
      "Type" : "String",
      "Default": "cumulus-prod"
    },
    "InstanceType" : {
      "Description" : "EC2 instance type",
      "Type" : "String",

```

```

    "Default" : "t1.micro",
    "AllowedValues" : [ "t1.micro", "m1.small", "m1.medium", "m1.large", "m1.xlarge",
↪ "m2.xlarge", "m2.2xlarge", "m2.4xlarge", "c1.medium", "c1.xlarge", "cc1.4xlarge", "cc2.
↪ 8xlarge", "cgl.4xlarge"],
    "ConstraintDescription" : "must be a valid EC2 instance type."
  },

  "CumulusEnvironment": {
    "Description" : "Cumulus environment name",
    "Type": "String"
  },

  "CumulusBundleBucket": {
    "Description" : "Cumulus bundle bucket name",
    "Type": "String"
  },

  "CumulusVersion": {
    "Description" : "Version of the software",
    "Type": "String"
  }
},

"Mappings" : {
  "AWSInstanceType2Arch" : {
    "t1.micro" : { "Arch" : "64" }
  },

  "AWSRegionArch2AMI" : {
    "eu-west-1" : { "32" : "NOT_YET_SUPPORTED", "64" : "ami-db595faf", "64HVM"
↪ : "NOT_YET_SUPPORTED" }
  }
},

"Resources" : {

  "WebServerLaunchConfiguration" : {
    "Type": "AWS::AutoScaling::LaunchConfiguration",
    "Metadata" : {
      "AWS::CloudFormation::Init" : {
        "configSets" : {
          "cumulus": [ "fileConfig", "commandConfig" ]
        },
        "fileConfig" : {
          "files" : {
            "/etc/cumulus/metadata.conf" : {
              "content" : { "Fn::Join" : [ "", [
                "[metadata]\n",
                "access-key-id: ", { "Ref" : "WebServerKeys" }, "\n",
                "secret-access-key: ", { "Fn::GetAtt" : [ "WebServerKeys",
↪ "SecretAccessKey" ] }, "\n",
                "region: ", { "Ref" : "AWS::Region" }, "\n",
                "bundle-bucket: ", { "Ref" : "CumulusBundleBucket" }, "\n",
                "environment: ", { "Ref" : "CumulusEnvironment" }, "\n",
                "bundle-types: generic, webserver\n",
                "bundle-extraction-paths:\n",
                "    generic -> /etc/example\n"
              ] ] }
            }
          }
        }
      }
    }
  }
}

```

```

        "    webserver -> /\n",
        "version: ", { "Ref" : "CumulusVersion" }, "\n"
    ]}},
    "mode" : "000644",
    "owner" : "root",
    "group" : "root"
},

"/etc/cfn/cfn-credentials" : {
    "content" : { "Fn::Join" : [ "", [
        "AWSAccessKeyId=", { "Ref" : "WebServerKeys" }, "\n",
        "AWSSecretKey=", { "Fn::GetAtt" : ["WebServerKeys", "SecretAccessKey
↪ " ]}], "\n"
    ]}},
    "mode" : "000400",
    "owner" : "root",
    "group" : "root"
},

"/etc/cfn/cfn-hup.conf" : {
    "content" : { "Fn::Join" : [ "", [
        "[main]\n",
        "stack=", { "Ref" : "AWS::StackName" }, "\n",
        "credential-file=/etc/cfn/cfn-credentials\n",
        "region=", { "Ref" : "AWS::Region" }, "\n",
        "interval=1\n"
    ]}},
    "mode" : "000400",
    "owner" : "root",
    "group" : "root"
},

"/etc/cfn/hooks.d/cfn-auto-reloader.conf" : {
    "content": { "Fn::Join" : [ "", [
        "[cfn-auto-reloader-hook]\n",
        "triggers=post.update\n",
        "path=Resources.WebServerLaunchConfiguration.Metadata.
↪ AWS::CloudFormation::Init\n",
        "action=/usr/local/bin/cfn-init -c cumulus -s ",
        { "Ref" : "AWS::StackName" }, " -r WebServerLaunchConfiguration
↪ ",
        " --credential-file /etc/cfn/cfn-credentials ",
        " --region ", { "Ref" : "AWS::Region" }, "\n",
        "runas=root\n"
    ]}}
    }
},

"commandConfig" : {
    "commands" : {
        "cumulus_bundle_handler" : {
            "command" : "/usr/local/bin/cumulus_bundle_handler.py",
            "ignoreErrors" : "false"
        }
    }
}
}

```

```

    },

    "Properties": {
        "ImageId" : {
            "Fn::FindInMap" : [
                "AWSRegionArch2AMI",
                { "Ref" : "AWS::Region" },
                { "Fn::FindInMap" : [
                    "AWSInstanceType2Arch",
                    { "Ref" : "InstanceType" },
                    "Arch"
                ] }
            ]
        },
        "InstanceType" : { "Ref" : "InstanceType" },
        "SecurityGroups" : [ { "Ref" : "WebServerSecurityGroup" } ],
        "KeyName" : { "Ref" : "KeyName" },
        "UserData" : { "Fn::Base64" : { "Fn::Join" : [ "", [
            "#!/bin/bash -v\n",

            "# Install cfn bootstrapping tools\n",
            "apt-get update\n",
            "apt-get -y install python-setuptools python-pip\n",
            "easy_install https://s3.amazonaws.com/cloudformation-examples/aws-cfn-
↳bootstrap-latest.tar.gz\n",

            "# Helper function\n",
            "function error_exit\n",
            "{\n",
            "  /usr/local/bin/cfn-signal -e 1 -r \"$1\" ", { "Ref" : "WaitHandle" }, "
↳'\n",

            "  exit 1\n",
            "}\n",

            "# Make sure we have the latest cumulus-bundle-handler\n",
            "pip install --upgrade cumulus-bundle-handler || error_exit 'Failed_
↳upgrading cumulus-bundle-handler to the latest version'\n",

            "# Install software\n",
            "/usr/local/bin/cfn-init -v -c cumulus -s ", { "Ref" : "AWS::StackName" },
↳" -r WebServerLaunchConfiguration ",
            "  --access-key ", { "Ref" : "WebServerKeys" },
            "  --secret-key ", { "Fn::GetAtt": ["WebServerKeys", "SecretAccessKey"] },
            "  --region ", { "Ref" : "AWS::Region" }, " >> /var/log/cfn-init.log ||
↳error_exit 'Failed to run cfn-init'\n",

            "# Start up the cfn-hup daemon to listen for changes to the Web Server_
↳metadata\n",
            "/usr/local/bin/cfn-hup || error_exit 'Failed to start cfn-hup'\n",

            "# All is well so signal success\n",
            "/usr/local/bin/cfn-signal -e 0 -r \"Webserver setup complete\" ", { "Ref"
↳: "WaitHandle" }, "\n"

        ] ] ] }
    }
  },
}

```

```

"WebServerAutoScalingGroup": {
  "Type": "AWS::AutoScaling::AutoScalingGroup",
  "Version": "2009-05-15",
  "Properties": {
    "AvailabilityZones": { "Fn::GetAZs": "" },
    "LaunchConfigurationName": { "Ref": "WebServerLaunchConfiguration" },
    "MinSize": "1",
    "MaxSize": "1",
    "Tags" : [{
      "Key" : "Name",
      "Value" : { "Fn::Join" : [ "-", [ { "Ref" : "AWS::StackName" }, "webserver
↪" ] ] },
      "PropagateAtLaunch" : "true"
    }]
  }
},

"WebServerUser" : {
  "Type" : "AWS::IAM::User",
  "Properties" : {
    "Path": "/",
    "Policies": [
      {
        "PolicyName": "cloudformation",
        "PolicyDocument": { "Statement": [{
          "Effect": "Allow",
          "Action": [
            "cloudformation:DescribeStackResource",
            "s3:*"
          ],
          "Resource": "*"
        }] }
      ]
    ]
  }
},

"WebServerKeys" : {
  "Type" : "AWS::IAM::AccessKey",
  "Properties" : {
    "UserName" : { "Ref": "WebServerUser" }
  }
},

"WaitHandle" : {
  "Type" : "AWS::CloudFormation::WaitConditionHandle"
},

"WaitCondition" : {
  "Type" : "AWS::CloudFormation::WaitCondition",
  "DependsOn" : "WebServerAutoScalingGroup",
  "Properties" : {
    "Handle" : { "Ref" : "WaitHandle" },
    "Timeout" : "600"
  }
},

"WebServerSecurityGroup" : {

```

```
"Type" : "AWS::EC2::SecurityGroup",
"Properties" : {
  "GroupDescription" : "Enable HTTP access via port 80/443 and SSH access",
  "SecurityGroupIngress" : [
    {"IpProtocol" : "tcp", "FromPort" : "80", "ToPort" : "80", "CidrIp" : "0.0.
↪0.0/0"},
    {"IpProtocol" : "tcp", "FromPort" : "443", "ToPort" : "443", "CidrIp" : "0.
↪0.0.0/0"},
    {"IpProtocol" : "tcp", "FromPort" : "22", "ToPort" : "22", "CidrIp" : "0.0.
↪0.0/0"},
    {"IpProtocol" : "icmp", "FromPort" : "-1", "ToPort" : "-1", "CidrIp" : "0.0.
↪0.0/0"}
  ]
}
}
```

Windows instance in a VPC

Below is an example of a Windows instance in a VPC.

```
{
  "Description" : "Example with Windows instance and VPC",
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Parameters" : {
    "InstanceType" : {
      "Description" : "Instance type to use",
      "Type" : "String",
      "AllowedValues" : [ "t1.micro", "m1.small", "m1.medium", "m1.large", "m1.xlarge",
↪"m2.xlarge", "m2.2xlarge", "m2.4xlarge", "c1.medium", "c1.xlarge", "cc1.4xlarge", "cc2.
↪8xlarge", "cg1.4xlarge"],
      "ConstraintDescription" : "must be a valid EC2 instance type."
    },
    "CumulusEnvironment": {
      "Description" : "Cumulus environment name",
      "Type": "String"
    },
    "CumulusBundleBucket": {
      "Description" : "Cumulus bundle bucket name",
      "Type": "String"
    },
    "CumulusVersion": {
      "Description" : "Version of the software",
      "Type": "String"
    }
  },
  "Mappings" : {
```

```

"AWSInstanceType2Arch" : {
  "m1.small"      : { "Arch" : "64" },
  "m1.medium"     : { "Arch" : "64" },
  "m2.xlarge"     : { "Arch" : "64" },
  "m2.2xlarge"    : { "Arch" : "64" },
  "m2.4xlarge"    : { "Arch" : "64" },
  "m3.medium"     : { "Arch" : "64" },
  "m3.large"      : { "Arch" : "64" },
  "m3.xlarge"     : { "Arch" : "64" },
  "m3.2xlarge"    : { "Arch" : "64" },
  "m1.medium"     : { "Arch" : "64" }
},

"AWSRegionArch2AMI": {
  "eu-west-1": {
    "32" : "NOT_YET_SUPPORTED",
    "64" : "ami-12345678",
    "64HVM" : "NOT_YET_SUPPORTED"
  }
}
},

"Resources" : {

  "WebServer" : {
    "Type" : "AWS::EC2::Instance",
    "Properties" : {
      "ImageId" : {
        "Fn::FindInMap" : [
          "AWSRegionArch2AMI",
          { "Ref" : "AWS::Region" },
          { "Fn::FindInMap" : [ "AWSInstanceType2Arch", { "Ref" : "InstanceType" },
↪ "Arch" ] }
        ]
      },
      "KeyName": "sebdah-test",
      "InstanceType" : { "Ref" : "InstanceType" },
      "NetworkInterfaces" : [{
        "GroupSet" : [{ "Ref" : "WebServerSecurityGroup" }],
        "AssociatePublicIpAddress" : "true",
        "DeviceIndex" : "0",
        "DeleteOnTermination" : "true",
        "SubnetId" : "subnet-12345678"
      }],
      "Tags" : [
        { "Key": "Name", "Value" : { "Ref" : "AWS::StackName" } },
        { "Key": "Project", "Value" : { "Ref" : "Project" } }
      ],
      "UserData" : { "Fn::Base64" : { "Fn::Join" : [ "", [
        "<powershell>\n",
        "pip install -U cumulus-bundle-handler\n",
        "cfn-init.exe -v -c cumulus ",
        "-s ", { "Ref" : "AWS::StackName" },
        "-r WebServer ",
        "--access-key ", { "Ref" : "WebServerKeys" },

```

```

        --secret-key ", { "Fn::GetAtt": ["WebServerKeys", "SecretAccessKey"] },
        --region ", { "Ref" : "AWS::Region" }, "\n",

        "cfn-signal.exe -e $LASTEXITCODE ", { "Fn::Base64" : { "Ref" : "WaitHandle"
↪}}, "\n",

        "</powershell>"

    ]]]}
},
"Metadata" : {

    "AWS::CloudFormation::Init" : {
        "configSets" : {
            "cumulus": [ "fileConfig", "commandConfig", "serviceConfig" ]
        },

        "fileConfig" : {
            "files" : {
                "c:\\cumulus\\conf\\metadata.conf" : {
                    "content" : { "Fn::Join" : [ "", [
                        "[metadata]\n",
                        "access-key-id: ", { "Ref" : "WebServerKeys" }, "\n",
                        "secret-access-key: ", { "Fn::GetAtt": ["WebServerKeys",
↪"SecretAccessKey"] }, "\n",
                        "region: ", { "Ref" : "AWS::Region" }, "\n",
                        "bundle-bucket: ", { "Ref" : "CumulusBundleBucket" }, "\n",
                        "environment: ", { "Ref" : "CumulusEnvironment" }, "\n",
                        "bundle-types: web\n",
                        "bundle-extraction-paths:\n",
                        "    web -> c:\\InetPub\\wwwroot\n",
                        "version: ", { "Ref" : "CumulusVersion" }, "\n"
                    ] ] },
                    "mode" : "000644",
                    "owner" : "root",
                    "group" : "root"
                },

                "c:\\cfn\\cfn-credentials" : {
                    "content" : { "Fn::Join" : [ "", [
                        "AWSAccessKeyId=", { "Ref" : "WebServerKeys" }, "\n",
                        "AWSSecretKey=", { "Fn::GetAtt": ["WebServerKeys", "SecretAccessKey
↪"] }, "\n"
                    ] ] },
                    "mode" : "000400",
                    "owner" : "root",
                    "group" : "root"
                },

                "c:\\cfn\\cfn-hup.conf" : {
                    "content" : { "Fn::Join" : [ "", [
                        "[main]\n",
                        "stack=", { "Ref" : "AWS::StackName" }, "\n",
                        "credential-file=c:\\cfn\\cfn-credentials\n",
                        "region=", { "Ref" : "AWS::Region" }, "\n",
                        "interval=1\n"
                    ] ] },
                    "mode" : "000400",

```

```

        "owner"      : "root",
        "group"     : "root"
    },

    "c:\\cfn\\hooks.d\\cfn-auto-reloader.conf" : {
        "content": { "Fn::Join" : [ "", [
            "[cfn-auto-reloader-hook]\\n",
            "triggers=post.update\\n",
            "path=Resources.WebServer.Metadata.AWS::CloudFormation::Init\\n",
            "action=cfn-init.exe -c cumulus -s ",
            { "Ref" : "AWS::StackName" }, " -r WebServer ",
            " --credential-file c:\\cfn\\cfn-credentials ",
            " --region ", { "Ref" : "AWS::Region" }, "\\n"
        ] ] }
    }
},

"commandConfig" : {
    "commands" : {
        "cumulus-bundle-handler" : {
            "command" : "python cumulus-bundle-handler",
            "ignoreErrors" : false
        },
        "RecycleAppPool" : {
            "command" : "C:\\windows\\System32\\inetsrv\\appcmd.exe recycle_
↪apppool DefaultAppPool",
            "ignoreErrors" : false
        }
    }
},

"serviceConfig" : {
    "services" : {
        "windows" : {
            "cfn-hup" : {
                "enabled" : "true",
                "ensureRunning" : "true",
                "files" : ["c:\\cfn\\cfn-hup.conf", "c:\\cfn\\hooks.d\\cfn-auto-
↪reloader.conf"]
            }
        }
    }
}
},

"WebServerKeys" : {
    "Type" : "AWS::IAM::AccessKey",
    "Properties" : {
        "UserName" : { "Ref" : "WebServerUser" }
    }
},

"WebServerUser" : {
    "Type" : "AWS::IAM::User",
    "Properties" : {

```

```
"Path" : "/",
"Policies" : [
  {
    "PolicyName" : "cloudformation",
    "PolicyDocument" : { "Statement": [{
      "Effect" : "Allow",
      "Action" : [
        "cloudformation:DescribeStackResource",
        "s3:*"
      ],
      "Resource" : "*"
    }]}
  }
],
},
{
  "WaitHandle" : {
    "Type" : "AWS::CloudFormation::WaitConditionHandle"
  },
  "WaitCondition" : {
    "Type" : "AWS::CloudFormation::WaitCondition",
    "DependsOn" : "WebServer",
    "Properties" : {
      "Handle" : { "Ref" : "WaitHandle" },
      "Timeout" : "3600"
    }
  },
  "WebServerSecurityGroup" : {
    "Type": "AWS::EC2::SecurityGroup",
    "Properties" : {
      "VpcId": "vpc-12345678",
      "GroupDescription": "Allow all traffic",
      "SecurityGroupIngress": [
        {
          "IpProtocol": "tcp",
          "FromPort": "0",
          "ToPort": "65535",
          "CidrIp": "0.0.0.0/0"
        }
      ],
      "SecurityGroupEgress": [
        {
          "IpProtocol": "tcp",
          "FromPort": "0",
          "ToPort": "65535",
          "CidrIp": "0.0.0.0/0"
        }
      ],
      "Tags" : [
        { "Key": "Name",      "Value" : { "Ref" : "AWS::StackName" } }
      ]
    }
  }
}
```

Cumulus release notes

1.4.0

Release date: 2014-04-17

- Add support for `--redploy` #134
- Fix event order when using `--deploy` and `--undeploy` in combination #135

1.3.2

Release date: 2014-04-14

- Follow symlinks when creating bundles #133

1.3.1

Release date: 2014-04-07

- Only directories allowed in bundle paths #132

1.3.0

Release date: 2014-03-17

- New `include` statement for config file inheritance #128
- Support for multiple config files in `--config` #129
- Print configuration errors prettier #125
- Catch error when template does not exist #122

1.2.2

Release date: 2014-03-14

- Bugfix: Error in pattern matching when bundling #123

1.2.1

Release date: 2014-03-12

- Bug fix: Cumulus misses stacks if the stack history is very long #121

1.2.0

Release date: 2014-03-11

- Support both `/` and `in` bundle paths and path rewrites #120
- Make INFO the default log level in Cumulus #119

1.1.3

Release date: 2014-03-10

- Bug fix: Exception when parsing parameters containing “=” #118

1.1.2

Release date: 2014-03-10

- Bug fix: Fixed ugly output of CloudFormation outputs

>>>>>>> hotfix/cumulus-1.1.2 1.1.1 —

Release date: 2014-03-10

- Bug fix: Stacks are added without the environment context #117

1.1.0

Release date: 2014-03-04

- Cumulus is now comparing the md5 checksums after uploads to ensure file integrity (#115)
- CloudFormation output is now shown after template deployment and if you issue the `--outputs` command (#114)
- Cumulus will only upload bundles to S3 if it does not exist or if the md5 checksum is updated (#99)
- Bug fix: `-cumulus-version` is broken #116

1.0.3

Release date: 2014-02-28

- Fixed licensing. Removed old references to proprietary

1.0.2

Release date: 2014-02-24

- Ensure removed backslashes in Windows rewrites #113

1.0.1

Release date: 2014-02-24

- Update default `cumulus.conf` paths for Windows #112
- Minor fixes

1.0.0 (First open source release)

Release date: 2014-02-20

- Write event status reason in terminal output #110
- Make it possible to force undeployment #105
- Break out Cumulus Bundle Handler to it's own module #90
- Bug fix: Catch missing pre-built bundles cleanly #109
- Bug fix: Proper error message when CBH can't find the config #101
- Bug fix: Old update events are shown when new updates are performed #79

0.8.0

Release date: 2014-01-31

This release is the first Cumulus release to support Windows. Windows is supported both as client system and a target system.

- Support for running Cumulus on a Windows client #80
- Support using pre-bundled software #82
- Create clean error when command line options are missing #85
- Windows support in Cumulus Bundle Handler #83
- Custom extraction path in Cumulus Bundle Handler #84
- Support zip, tar.gz and tar.bz2 in Cumulus Bundle Handler #88

0.7.0

Release date: 2014-01-28

- Support deployment of certain stacks only #70
- Add support for stack creation timeouts #76
- Ensure stack deletion order #74
- Support CloudFormation stack tags #78
- Update for all stacks fail if one stack fails #73
- Log level config in CBH #64
- Ugly error when trying to deploy unconfigured environment #71
- Stack deletion events are not handled properly #72
- Catch ctrl-c interruptions cleanly #75

0.6.4

Release date: 2014-01-21

- Fix odd syntax in parameters option #69

0.6.3

Release date: 2014-01-20

- It is not possible to run `--deploy` with a `cumulus.conf` without bundles #67
- Minor fix: Enhanced event log output

0.6.2

Release date: 2013-01-20

- Exclude all other configuration files if `--config` is set #68

0.6.1

Release date: 2013-12-02

- All `cumulus-init.d` scripts run both before and after bundle deploy #66
- Widen output formatting for Logical ID #65

0.6.0

Release date: 2013-11-29

Major features:

- Global `cumulus` command and documentation generation #56
- Support multiple bundle types on hosts #52
- Support CloudFormation templates served from S3 #58
- Cumulus bundle handler should support both start and kill scripts in `init.d` #49
- Generate Python docs with `autodoc` #59
- Added Sphinx documentation #48
- Set CF parameters on command line #61
- Log level is now configurable #63

Minor improvements:

- Stop writing to target dir, use tempfile instead #62
- Harmonize CBH option names #53
- Restructured project folders #54
- Bundle Cumulus in a Python egg #55
- Remove docs from README #57
- Read versions from one place #60
- Bug: paths should be `n` separated, not comma separated #51

0.5.0

Release date: 2013-10-28

- Clean up host on bundle update #38
- Cumulus bundle handler should use Python logging #40
- Get rid of Cumulus metadata.conf and make the bundle handler self-contained #41
- Remove __name__ from logging output #42
- Filter events when creating/updating/deleting stacks #43
- Add function for listing stack events on command line #45
- Enhance status output when waiting for stack change to complete #46

0.4.0

Release date: 2013-10-25

- Path prefix in bundles #36

0.3.1

Release date: 2013-10-24

- Error handling stack delete status #34
- Running `–deploy` on existing stack fails #35
- Initial stack creation fails when using `–deploy-without-bundling` #33
- Bundle type missing in Cumulus metadata #37

0.3.0

Release date: 2013-10-11

- Write hooks for Cumulus deployments #26
- Wait until stack is done updating/creating #20
- Specify config file location as input parameter #30
- Set environment version as input parameter #28
- Make it possible to environment prefix whole directories #10
- Create shortcut for both bundling and deploying #27
- Ask before delete when running `–undeploy` #24
- Ensure that boto is available for cumulus bundle handler #25
- Remove skymill reference from JSON template #23
- Remove unnecessary stack name in metadata #22
- Remove unnecessary bundle-type in metadata #21

0.2.3

Release date: 2013-09-26

- Symbolic links should be dereferenced in bundles #19
- Current directory is added to bundle #18

0.2.2

Release date: 2013-09-25

- Mismatch in metadata and `cumulus_bundle_handler.py` #16
- Various bug fixes in the bundle handler system

0.2.1

Release date: 2013-09-25

- Cumulus CF namespace conflicts with some rules #15

0.2.0

Release date: 2013-09-24

- Custom parameters in CloudFormation #14
- Expand ~ in config template & bundle paths #12
- Read the bucket name from configuration in CF template #11
- Exception when building non-configured bundle #13

0.1.1

Release date: 2013-09-23

- Prefixes for prefixed files is not removed in bundle #9

0.1.0

Release date: 2013-09-23

Initial release with some basic functions and concepts.

- Basic bundling and stack management features implemented

Cumulus Bundle Handler release notes

1.0.5

Release date: 2016-05-17

- Fixed script execution ordering issue, now properly honors numbering (S01-script1, S05-script2, S30-script3)

1.0.4

Release date: 2014-03-18

- Fixed potential issue with path joins on Windows
- Fixed wrong permission on extracted directories

1.0.3

Release date: 2014-03-14

- Bugfix: [CBH does not preserve the file permissions on extraction #124](#)

1.0.2

Release date: 2014-02-28

- Fixed licensing. Removed old references to proprietary

1.0.1

Release date: 2014-02-21

- Bugfix: [Bundle extraction paths are not determined properly #111](#)

Bug reports and feature requests

If you find any bugs, need help or have any feature requests, please feel free to submit an issue on the projects [GitHub issues page](#).

Pull requests are always very welcome :)!

License

APACHE LICENSE 2.0 Copyright 2013-2014 Sebastian Dahlgren & Skymill Solutions

Licensed under the Apache License, Version 2.0 (the “License”); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.