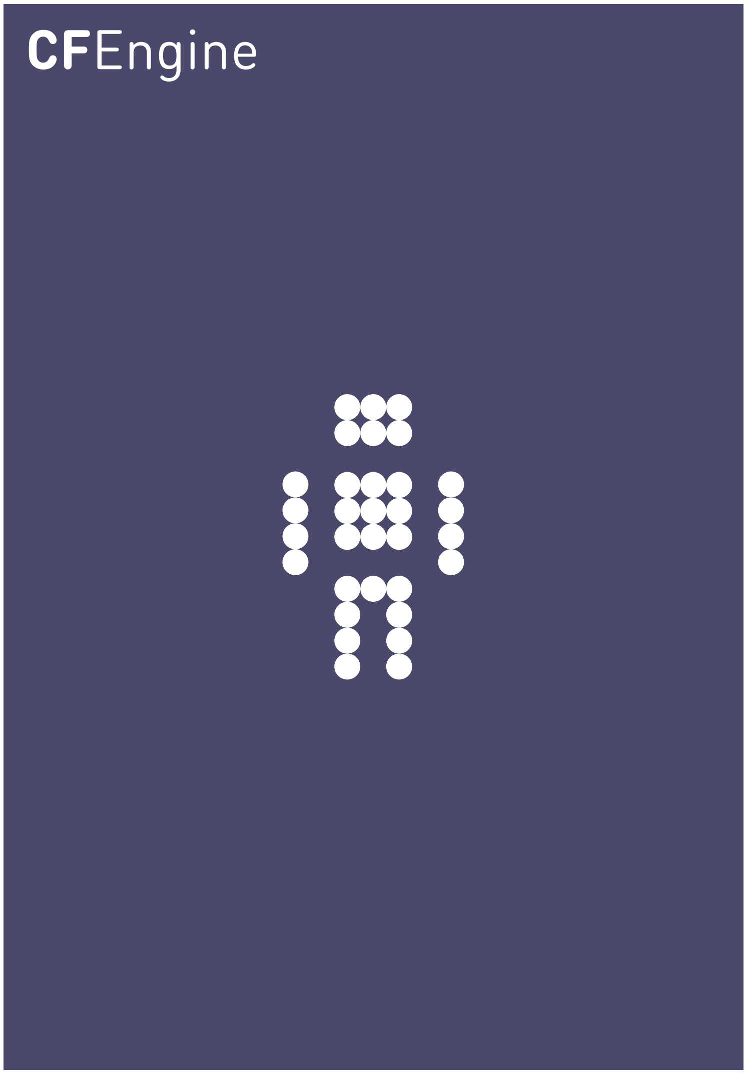


The Cfengine Orion Cloud Pack - EC2

A Cfengine Handbook

Cfengine AS

Get the Cloud under Orion's belt with the Cfengine Orion Cloud Pack.

This document describes the Three Steps you need to bring reliability and efficiency to Managed Services running out of the Amazon Cloud. Set up and tear down machines as you like, and bring instant configuration and compliance, with self-healing to your business.

Table of Contents

Introduction to the CFEngine Orion Cloud Pack

Welcome to the CFEngine Orion Cloud Pack. This version of the package has been designed to work specificallgy with the *Amazon Elastic Compute Cloud*. It allows you to configure a cloud computer or 'platform instance' to run common services in a reproducible and maintainable way and without human intervention.

The CFEngine Cloud Pack is not a tool for performing elastic scaling of services on demand, rather it is a simple repeatable process for speeding up system installation with automatic self-healing, for a consistent and compliant outcome.

CFEngine is uniquely suited to work in the Cloud because it is capable of install systems and maintaining and repairing them with hands-free capabilities that cannot be matched by other software. The CFEngine Orion Cloud Pack may also be used on any other non-Cloud systems that run the base operating-systems discussed here.

You can thus test and learn about CFEngine using disposable Cloud 'instances' and then use your experience on more permanent hardware elsewhere, if you wish. Or, you can simply use the CFEngine Orion Cloud Pack to bring up cloud services and configure them to a desired state on demand.

Some background on EC2

If you are familiar with Amazon Amazon Elastic Compute Cloud (EC2), you will find the instructions here straightforward. You can choose between these alternative paths:

- Do It Yourself: launch your own Amazon Machine Image and install CFEngine on your own, then install the Orion Cloud Pack software and continue. If you choose this option, your CFEngine policies are only examples and are unsupported.
- With our Help: use one of our publicly available images, located in the Amazon storage area, which has CFEngine Community Edition pre-installed for your convenience, then follow the Orion Cloud Pack installation procedure to continue. You will then have access to 30 days of E-mail support from CFEngine, with a maximum of 5 enquires.

An Amazon Machine Image or AMI is a pre-configured server designed to be launched and available on demand. Each AMI template has a unique number of the form `ami-XXXXX`. Advanced users may also create their own images customized with applications, data and configuration settings as desired. The AMIs are the *baseline* or starting point for Cloud Computing.

Prerequisites for setting up Orion

To get started in the Cloud, you will need an Amazon Web services account and some familiarity with either the web interface or command line tools[1].

> You should know something about configuration of *security groups*. For a start-to-finish guide to launching an AMI and setting up the CFEngine Orion Cloud Pack see Appendix A.

We have published the following AMIs on EC2 storage for your convenience:

	us-east	us-west
Ubuntu Server 9.10 32	ami-XXXXXX	ami-XXXXXX
Ubuntu Server 9.10 64	ami-XXXXXX	ami-XXXXXX
Centos 5.4 32	ami-XXXXXX	ami-XXXXXX
Centos 5.4 64	ami-XXXXXX	ami-XXXXXX

You should subscribe to one or more of these. The following sections assume that you have already acquired such an instance.

Three steps to the cloud

To use the CFEngine Orion Cloud Pack, there are just three steps:

1. **UNPACK**: Copy the CFEngine Orion Cloud Pack to your EC2 instance and unpack the files in '`/var/cfengine/masterfiles`'.
2. **CUSTOMIZE**: Edit the policy promise examples for your purposes. In particular, look at the master file '`promises.cf`', comment out or uncomment the promises you want. The default promises construct a PHP enabled webserver. You should also place the IP address of the policy server in the '`policy_server.dat`' file, like this:

   ```
   echo "IP-address" > /var/cfengine/policy_server.dat
   ```

   This tells CFEngine where to look for the cloud pack. If you are running on a single test machine, you can make this localhost (127.0.0.1); if you are running several machines, make it the first machine where you installed the cloud-pack.

   Fill in the `policy_server` variable with the appropriate IP address in '`update.cf`' and '`promises.cf`'. This guide assumes that the single instance your hvae created will be its own policy server. However when setting up multiple instances one or more may be promoted' to be policy servers.
3. **ACTIVATE**: Run `cf-agent -f /var/cfengine/masterfiles/failsafe.cf'` to start CFEngine management.

You now have a running 'instance', with services configured and maintained by CFEngine.

[1] This booklet is not meant as an introduction to using the Amazon EC2. See the Appendices for references to Cloud Resources.

How do I know that CFEngine is maintaining the system?

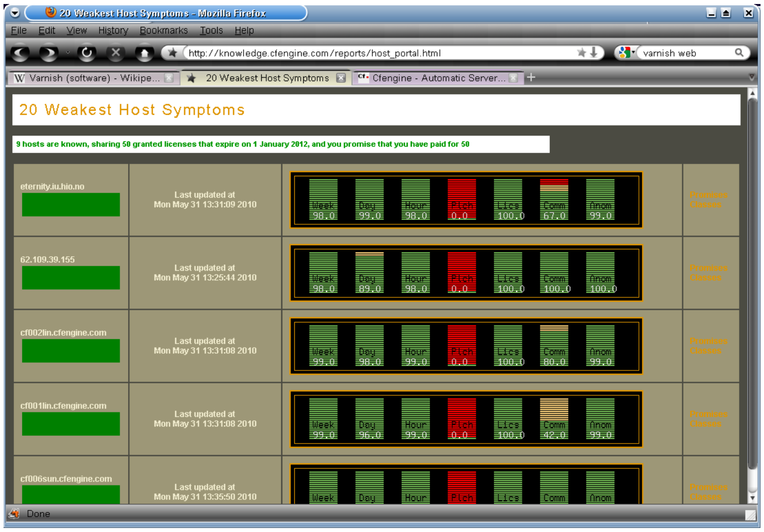
Execute ‘ps aux | grep cf-’ to see cf-agent, cf-serverd and cf-execd in the process table. To test self repair try stopping the web server.

Additional benefits for users running CFEngine Nova

If you have access to a commercial edition of the CFEngine software, such as CFEngine Nova, you will have a number of immediate benefits to simplify Cloud Management.

- **Knowledge and monitoring**

  Without any further configuration or third party software, you can monitor your cloud instances with CFEngine Nova's reporting features. You can add special logging and see performance trend analysis, integrated into your Nova Knowledge Map.


20 Weakest Host Symptoms - Mozilla Firefox
File
Edit
View
History
Bookmarks
Tools
Help
http://knowledge.cfengine.com/reports/host_portal.html
varnish web
Varnish (software) - Wikipe...
20 Weakest Host Symptoms
Cfengine - Automatic Server...
Done

20 Weakest Host Symptoms

9 hosts are known, sharing 50 granted licenses that expire on 1 January 2012, and you promise that you have paid for 50

eternity.iu.hio.no	Last updated at Mon May 31 13:31:09 2010	Week 98.0	Day 99.0	Hour 98.0	Plch 0.0	Lics 100.0	Comm 67.0	Anom 99.0	Promises Classes
62.109.39.155	Last updated at Mon May 31 13:25:44 2010	Week 98.0	Day 89.0	Hour 98.0	Plch 0.0	Lics 100.0	Comm 100.0	Anom 100.0	Promises Classes
cf002lin.cfengine.com	Last updated at Mon May 31 13:31:08 2010	Week 99.0	Day 98.0	Hour 99.0	Plch 0.0	Lics 100.0	Comm 80.0	Anom 99.0	Promises Classes
cf001lin.cfengine.com	Last updated at Mon May 31 13:31:08 2010	Week 99.0	Day 96.0	Hour 99.0	Plch 0.0	Lics 100.0	Comm 42.0	Anom 99.0	Promises Classes
cf006sun.cfengine.com	Last updated at Mon May 31 13:35:50 2010	Week	Day	Hour	Plch	Lics	Comm	Anom	Promises Classes

- **Compliance reporting**

  Not only will you see the automatically integrated policy documentation but you will be able to trace the behaviour and utilization of your systems over the lifetime of the virtual instance.

- **Set up databases using `database` promsises**

  Most users working in the cloud are running some kind of Model View Controller web framework in which a database (like MySQL or PostgreSQL) features importantly. With CFEngine Nova, you can also manage the creation of databases for the setting up the data services and applications inside PHP, Java and other frameworks.

The Cloud Pack Contents

The files in the Cloud Pack fall into 3 categories: essential set up files for getting CFEngine running, examples of pro-active maintenance and examples of basic machine installation and setup.

Essential Files	**Description**
‘promises.cf’	Main configuration file.
‘update.cf’	Update configuration.
‘failsafe.cf’	Falisafe configuration.
‘cfengine_stdlib.cf’	CFEngine standard library.

Maintenance examples	**Description**
‘change_mgt.cf’	Implement security tripwire on files/directories.
‘ensure_ownership.cf’	Home directory ownership and permission maintenance.
‘fix_broken_software.cf’	Package installation and permission correction.
‘garbage_collection.cf’	Log rotation and removal.
‘harden_xinetd.cf’	Disable xinetd services specified.
‘iptables.cf’	Secure system with sysctl.conf and iptables.
‘name_resolution.cf’	Edit /etc/resolv.conf to the specified DNS servers

System setup examples	**Description**
‘c_cpp_env.cf’	Set up C programming environment.
‘db_mysql.cf’	Install and run MySQL
‘db_postgresql.cf’	Install and run PostgreSQL
‘db_sqllite.cf’	Install and run SQLlite
‘jboss_server.cf’	Prepare JAVA environment and run JBOSS.
‘nagios.cf’	Setup NAGIOS monitoring node.
‘nginx_perlcgi.cf’	Setup NGINX webserver perlCGI.
‘nginx_phpcgi.cf’	Setup NGINX webserver phpCGI.
‘ntp.cf’	Setup NTP server and clients.
‘perl_env.cf’	PERL programming language install.
‘php_webserver.cf’	Setup a PHP webserver.
‘python_env.cf’	PYTHON programming install.
‘ruby_env.cf’	Setup ruby on rails environment.
‘sshd_conf.cf’	Ensure sshd config is correct.
‘tomcat_server.cf’	Setup a tomcat server.
‘varnish.cf’	Set up Varnish web accelerator

Orion Support

Your CFEngine Orion Cloud Pack comes with 30 days of email support to help you get started (`cloudsupport@cfengine.com`). Support applies to the single individual who is recorded as the purchaser of the software, unless otherwise agreed with CFEngine.

On receipt of a query, our engineers will respond withing 48 hours on business days for a maximum period of 30 days from the date of purchase of the Orion Cloud Pack. Existing users of CFEngine Nova will receive support transparently under their existing support arrangements.

Support queries may cover issues connected with the use of CFEngine Orion Cloud Pack, but not with the use of your Cloud provider. Support does not include the development of new solutions or other consulting. Users are free to seek Professional Services from CFEngine for such matters.

Orion License

The CFEngine Orion Cloud Pack is not Free or Open Source Software. It is provided under a perpetual license as part of the CFEngine Cloud Pack (hereby called The Software). The Software may be used within a single Internet Domain. If no Internet Domain is registered, it may be used within a single legal organization possessing a maximum of 1024 computers, or by a single individual with up to 250 computers. Multiple licenses may be purchased, as needed.

The Licensee may modify, adapt and create derivative works based upon the Software, for use within its organisation and for sharing between other consecutive licensees. However, the Licensee shall not reproduce, distribute, resell, rent, lease or disclose the Software in any manner or form to any other third party not holding a license for the Software.

The Licensee may not transfer any of its rights under this agreement without the prior and express written consent of CFEngine.

CFEngine does not transfer any copyrights or other intellectual property rights relating to the Software to the Licensee. Such rights are protected by intellectual property legislation in the United States, Europe and other jurisdictions and by international treaty provisions. CFEngine and its suppliers retain all rights in the Software that are not expressly granted to the Licensee through this license.

The Licensee is not allowed to remove, alter or destroy any proprietary, trademark or copyright markings or notices placed upon or contained within the Software.

To the maximum extent permitted by law, CFEngine disclaims any warranty for the Software. The Software, any services and any related documentation are provided on an "as is" basis without warranty of any kind, whether express or implied, including, but not limited to, implied warranties of merchantability, fitness for a particular purpose or non-infringement. Hereunder the parties acknowledges that CFEngine does not warrant for the performance of any data centre on which the Software runs, or the absence of any errors in the Software, and that any such errors does not constitute a contractual defect.

The liability of the parties in contract, tort (including negligence) or otherwise shall for all incidents during the entire term of 30 days from the date of purchase be limited to a half of the fees paid for a perpetual license. CFEngine or its suppliers shall not be liable for any special, incidental, indirect or consequential damages whatsoever (including, without limitation, damages for loss of business profits, lost savings, business interruption, loss of business information, personal injury, loss of privacy, loss of goodwill or any other financial loss) arising out of the use of or inability to use the Software, even if advised of the possibility of such damages.

For third-party software that is integrated with or used by CFEngine, the current terms of the relevant third party software supplier shall apply.

Cultural References to Orion and Cloud

Orion, the hunter from Greek mythology, was taken by renaissance mythologist Natalis Comes to be an allegory for an approaching storm cloud.

Better-known today, Orion is the name of one of the most famous and studied astronomical constellations in the night sky. It contains the three bright stars of Orion's belt (the three steps to the cloud) and the *Orion M42 Nebula* (a gigantic dust cloud): http://en.wikipedia.org/wiki/Orion_Nebula beneath its belt. Now with the CFEngine Orion Cloud Pack, you too can get the Cloud under your belt.

The Orion Nebula is, in fact, part of a much larger cloud or nebula in the constellation of Orion (http://en.wikipedia.org/wiki/Orion_Molecular_Cloud_Complex). We chose the name Orion for our Cloud Pack (apart from the obvious puns) because we believe that Cloud Computing is only a stepping stone towards what we term *Molecular Computing*, in which many independent platforms and services bond together to form *network patterns*. These patterned systems act like molecules with new properties, bonded together by *promises*. This view follows naturally from a description of computing using Promise Theory, replacing traditional monolithic and hierarchical systems with a more natural form of engineering (https://cfengine.com/inside/deepBackground).

Hubble Image: http://apod.gsfc.nasa.gov/apod/ap051013.html

Appendix A EC2 Cloud command line setup

This Appendix details the creation of an EC2 instance, i.e. the pre-requisites for CFEngine installation, using a command-line approach. You may also use the web interface.

For greater depth and explanation of all the commands and options see the getting started guide:

http://docs.amazonwebservices.com/AWSEC2/latest/GettingStartedGuide/.

1. Create an amazon web services account and sign up for Amazon Elastic Compute Cloud (EC2) at http://aws.amazon.com/ec2/.
2. Login and go to the access identifiers page.
3. Create new X.509 certificate.
4. Download the private key file and the X.509 certificate.
5. Download the EC2 api tools:

   http://developer.amazonwebservices.com/connect/entry.jspa
?categoryID=88&externalID=351
6. Extract the tools into a suitable directory: $dir.
7. Setup EC2 control environment:
   - Install Java
   - `export EC2_HOME=~/$dir`
   - `export PATH=$PATH:$EC2_HOME/bin`
   - `export EC2_PRIVATE_KEY=pk-KEYNAME.pem`
   - `export EC2_CERT=cert-KEYNAME.pem`
   - `export JAVA_HOME=/path/to/java/`
8. Create keys for accessing your instances:
   - `cd $dir`
   - `ec2-add-keypair keyname` where 'keyname' is users choice. Save output to a file:
   - `vi $dir/keyname`
   - paste in key and save.
   - `sudo chmod 600 keyname`
9. Choose an Amazon Machine Image (AMI) to launch: e.g. '`ami-6d0be204`', an Ubuntu 9.10 server with CFEngine community pre-installed (small instance in us-east region).
10. **Warning billing begins after the next command**. Run the instance:
    - `ec2-run-instances ami-6d0be204 -k keyname`

      Note is it possible to launch an instance without specifying a key but it will not be accessible via ssh if you do not create one. The selected public AMI is an Ubuntu 9.10 i368 server with CFEngine community installed.
11. Get status of your instance:
    - `ec2-describe-instances`

      This will reveal the external URL to your instance similar to:

      `ec2-xxx-xxx-xxx-xxx.compute-1.amazonaws.com`

12. Allow ssh:
    - `ec2-authorize default -p 22`

      Note this will allow ssh access to all instances in the default security group from anywhere on port 22.
    - Allow http traffic: `ec2-authorize default -p 80`
13. Access via ssh:
    - `ssh -i keyname ubuntu@ec2-xxx-xxx-xxx-xxx.compute-1.amazonaws.com`

Appendix B EC2 Cloud CFEngine Configuration

You will need `root` access to the Amazon instance.

1. Copy the CFEngine Orion Cloud Pack to your instance and unpack it:

   ```
   scp -i keyname cloud_pack.tar ubuntu@ec2-xxx-xxx-xxx-xxx.compute-1.amazonaws.com
   mv cloud_pack.tar /var/cfengine/masterfiles
   cd /var/cfengine/masterfiles
   tar xvf cloud_pack.tar
   ```

2. Set policy server ip address: e.g.

   `ifconfig eth0 ...`

3. Copy this IP address to the policy server variable in the '`update.cf`' and '`promises.cf`' files.
4. `cf-agent -f /var/cfengine/masterfiles/failsafe.cf`

Note billing continues as long as instances are running. To terminate an instance:

- `ec2-terminate-instances instance_number` The instance number (i-xxxxxxxx) is revealed by running:
- `ec2-describe-instances`

Appendix C Integrating the Cloud Pack Futher

The Orion Cloud Pack is delivered in a form that makes it easy to choose the services. As you move forward, or want to integrate these methods into a larger framework, you can choose to alter the way in which you 'call up' these methods.

In the Cloud Pack, all bundles are listed in the `bundlesequence`, making them simple to compare and comment out, but we may also re-bundle them in a single bundle like this:

```
body common control
{
bundlesequence => { "update", "main" };
}
```

To re-bundle, we create a single agent bundle (e.g. called 'main') and call the bundles as method promises. An advantage of this is to make it easier to classify different parts of your configuration in a single location. For instance, you might want two groups of servers supporting different platforms, one supporting Ruby and one supporting PHP:

```
bundle agent main
{
methods:

 group1::
   "environment 1" usebundle => ruby_on_rails;

 group2::
   "environment 2" usebundle => php_apache;

 any::
   "common" usebundle => fix_broken_software;
   "common" usebundle => garbage_collection;
   "common" usebundle => harden_xinetd;
}
```

This approach allows a great control over the execution of the bundles, with additional transaction controls, but cannot be considered superior to the simple list used in the Cloud Pack. Ultimately this is a question of style.

Appendix D Ubuntu quirks

The Ubuntu operating systems does not have a root account you can log onto directly. You first log in as the '`ubuntu`' user, and then type '`sudo su`' to become root.

CFEngine®