

Blogs

Told you, we love sharing!

[Home \(/\)](#) / [Blog \(http://www.tothenew.com/blog/\)](http://www.tothenew.com/blog/) / [Technology \(http://www.tothenew.com/blog/category/technology/\)](http://www.tothenew.com/blog/category/technology/) / [DevOps \(http://www.tothenew.com/blog/category/devops/\)](http://www.tothenew.com/blog/category/devops/)

Category

ENTER EMAIL

Subscribe to Our Blog

Chef: Bootstrapping an autoscaled Node using Cloud-Init, Cloud-Config and UserData

DEVOPS ([HTTP://WWW.TOTHENEW.COM/BLOG/CATEGORY/TECHNOLOGY/DEVOPS-TECHNOLOGY/](http://www.tothenew.com/blog/category/technology/devops-technology/))

01 / OCT / 2015 BY [AAKASH GARG \(HTTP://WWW.TOTHENEW.COM/BLOG/AUTHOR/AAKASH/\)](http://www.tothenew.com/blog/author/AAKASH/) [1 COMMENTS](http://www.tothenew.com/blog/chef-bootstrapping-an-autoscaled-node-using-cloud-init-cloud-config-and-userdata/#comments)
([HTTP://WWW.TOTHENEW.COM/BLOG/CHEF-BOOTSTRAPPING-AN-AUTOSCALED-NODE-USING-CLOUD-INIT-CLOUD-CONFIG-AND-USERDATA/#COMMENTS](http://www.tothenew.com/blog/chef-bootstrapping-an-autoscaled-node-using-cloud-init-cloud-config-and-userdata/#comments))

Share this blog

Email

Twitter

Facebook

LinkedIn

Google+

This blog refers to the bootstrapping an autoscaled Node using Cloud-Init, Cloud-Config and UserData and get it available for serving requests. Here, in this blog I am bootstrapping a web server. We will be using a yaml file and a recipe to implement the same.

What is Cloud-Init?

Cloud-Init is the package which is already installed on the cloud environment, it is used to initialize the early stages of the cloud servers. It locates user-data and executes the instructions pre written on it. It understands only two userData formats – shell script and cloud-config directives.

What is Cloud-Config?

It is Cloud-Init script which defines the parameters which are required when the server boots up.

What is UserData?

UserData is a type of metadata that can be passed to an instance to perform different actions including executing scripts and Chef-recipes on boot. It is limited to 16KB.

Chef-Server

It is the configuration management tool written in Ruby and Erlang. It is used to manage and configure organisation's servers.

Bootstrapping

While bootstrapping a node from Chef, user-data on the node will be called by the Cloud-Init program and chef-client will be installed which will register itself on Chef Server (<http://www.tothenew.com/devops-chef-puppet-docker>) and will pull out all the configured recipes for that 'type' of node.

Use Case:

We had a requirement where we wanted an AWS autoscaling (<http://www.tothenew.com/blog/aws-auto-scaling-lifecycle-hooks/>) node to automatically bootstrap and get itself ready for serving web requests via Nginx. Autoscaling node supposed to be a raw machine and will get its packages and services from Chef server via User data and Cloud-Init.

User Data Dissection

```

1  #cloud-config
2
3  # This example assumes the instance is 12.04 (precise)
4  # The default is to install from packages.
5
6  # Key from
7  apt_upgrade: true
8  apt_sources:
9  - source: "deb $RELEASE-0.10 main"
10 key: |
11 -----BEGIN PGP PUBLIC KEY BLOCK-----
12 Version: GnuPG v1.4.9 (GNU/Linux)
13
14 mQGibEppC7QRBADfsOkZU6KZK+YmKw4wev5mjKJEkVGlus+NxW8wItX5sGa6kdUu
15 zIhi7tKfO6WiwjKZq0dzctsJJjtIfr4K4vbiD90jg1iISQQYEQIACQUCSmkLtAIb
16 .....
17 DAAKCRApQKupg++CauISAJ9CxYPOKh0xa1BnVTLLeNUKAHGg2gACeIsbobtaD4ZHG
18 0GLl8EkfA8uhluM=
19 =zKAm
20 -----END PGP PUBLIC KEY BLOCK-----
21
22 #The above will verify that the we are downloading is from trusted source

```

The above will upgrade the package sources list and it will install chef client on the machine.

apt_upgrade :- It will upgrade the package sources.

apt_sources :- Here we specify the package which we are required to be installed.

key :- Key present here will verify the key that the package is being installed from trusted source.

```

1  chef:
2  run_list:
3  - "recipe[nginx]"
4
5  # Valid values are 'gems' and 'packages' and 'omnibus'
6  install_type: "packages"
7
8  # Boolean: run 'install_type' code even if chef-client
9  # appears already installed.
10 force_install: false
11
12 # Chef settings
13 server_url: "https://ip-X-X-X-X.ec2.internal/organizations/organisation-name"
14
15 # Node Name
16 # Defaults to the instance-id if not present
17 node_name: "#{Socket.gethostname}-test"
18 # Default validation name is organisation-validator
19 validation_name: "organisation-validator"
20 validation_key: |
21 -----BEGIN RSA PRIVATE KEY-----
22 MIIIEowIBAAKCAQEAw75CltGJ1oYf980m1B9soHbDVK/6l1dhgzd86aw8GjFpqfz8
23 Q9kXZ3wrLOxJ64+vIF1qouMzgtogCayZ00t62Mn CZodgPCzcCHY0CZKOKx2V1dDW
24 PAt1veQkXODDLilFiGfTlwBud+01JLoDa5Nxt6y9INL21IGx3sXi54Z7Df98ZS2z
25 5K5IzLjRsJmh2NhwG/fB7xRYXaImHrjua7c309rH411Pgo1xcQb9xETU4Wa0yl/C
26 .....
27 QEO7WQKBgGKenUL/CmBmehY6EBfsYyq1Pbwn/f3n1D0w61ef+zBD0KVn078dZozq
28 Wd2pd4EHxp4+v1NEGCF47YVfiO+D9Lk3V4ic+1kVfYCsefygdr5i1C4zUV0Gcso
29 QcbwgH/doZL7MfiofwHP8MupY410UQnnV2it72Ml+bJf9Q7ezjsX
30 -----END RSA PRIVATE KEY-----
31
32 #The above key is organisation-validator.pem which is present on the chef server

```

The above code will be executed after chef-client has installed

run_list :- Here we specify the recipes which needs to be executed after Chef has booted.

install_type :- It specifies which type of installation you want which can be 'gems', 'packages' and 'omnibus'.

server_url :- Here we specify the Chef server URL.

node_name :- Here we specify the name of the node which you want to be registered as.

validation_name :- Here we give the name organisation validator.

validation_key :- Here we specify the organisation validator key to authenticate the node with Chef server.

```

1  # if install_type is 'omnibus', change the url to download
2  omnibus_url: "https://www.opscode.com/chef/install.sh"
3
4  # Capture all subprocess output into a logfile
5  # Useful for troubleshooting cloud-init issues
6  output: {all: '| tee -a /var/log/cloud-init-output.log'}

```

omnibus_url :- Here we specify the URL from which we can download packages if install type omnibus is selected.

output :- It specifies where you want to store the logs of the above events.

runcmd :- Here we can write shell commands or scripts which will be executed on the machine.

The above yaml script can be downloaded from here (<https://github.com/aakash21/UserData/blob/master/userdata.yaml>)

Troubleshooting

Logs can be checked at cloud-init-output.log and client.log.

cloud-init-output.log

This file would contain all the logs of the events occurred when the userdata.yaml is executed. This has specified in the userdata.yaml itself.

```
output: {all: '| tee -a /var/log/cloud-init-output.log'}
```

client.log

/var/log/chef/client.log

This file contain the logs after Chef has installed and it tries to connect to Chef server. We can check here whether our node has connected to our Chef server.

What this UserData will do?

- It will first update the package sources.
- Then it will download and install the chef-client on that node from the defined repository.
- For initial handshake and registration, it will use organization validator key and will generate a RSA Private key (client.pem) for further communication.
- After making connection with the Chef server, it will get its packages as configured on Chef server and execute the recipe which is passed as run_list.
- Once chef-client run is completed, Server is ready !!

Conclusion

We can bootstrap our servers automatically with the pre defined cookbooks without logging into that server instance or into chef-workstation to manually bootstrap it.

Tag -

FOUND THIS USEFUL? SHARE IT

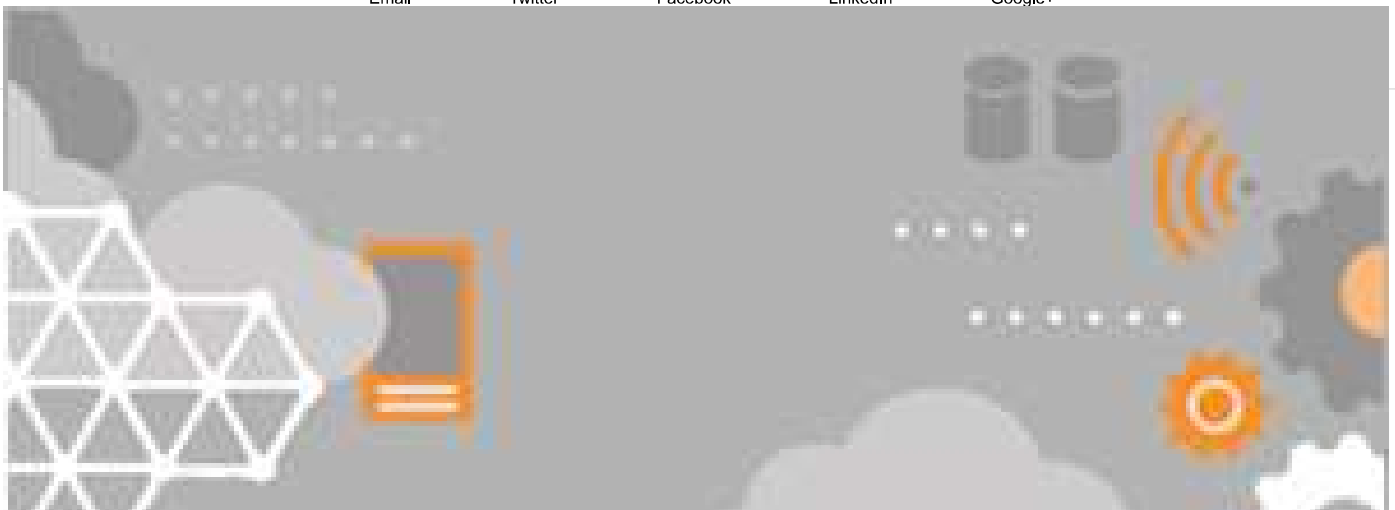
Email

Twitter

Facebook

LinkedIn

Google+





Ebook

4 Effective Tools for Docker Monitoring

COMMENTS (1: "CHEF: BOOTSTRAPPING AN AUTOSCALED NODE USING CLOUD-INIT, CLOUD-CONFIG AND USERDATA")

Download Ebook (<http://insights.tothenew.com/devops-tools-to-monitor-docker>)



Narendra -October 27, 2016 at 4:44 pm ([http://www.tothenew.com/blog/chef-](http://www.tothenew.com/blog/chef-bootstrapping-an-autoscaled-node-using-cloud-init-cloud-config-and-userdata/comment-page-1/#comment-172163)

[bootstrapping-an-autoscaled-node-using-cloud-init-cloud-config-and-userdata/comment-page-1/#comment-172163](http://www.tothenew.com/blog/chef-bootstrapping-an-autoscaled-node-using-cloud-init-cloud-config-and-userdata/comment-page-1/#comment-172163))

Hi Garg,

It looks very helpful for me to get started bootstrapping the ec2 instances automatically. I've come across a problem, where I've to pass unique name to the name="" parameter to identify my components in chef-server. am using terraform currently to provision the infrastructure. I am not able to achieve this scenario, it would be great if you can help me on this. Thanks very much in advance

Reply (/Blog/Chef-Bootstrapping-An-Autoscaled-Node-Using-Cloud-Init-Cloud-Config-And-Userdata/?Replyto=172163#Respond)

LEAVE A COMMENT -

Name *

Email *

Comment

Submit



by

Aakash Garg (<http://www.tothenew.com/blog/author/Aakash/>)**YOU MAY ALSO LIKE**

Register chef-client in AWS Autoscaling (<http://www.tothenew.com/blog/register-chef-client-in-aws-autoscaling/>)

Installing LAMP stack through Chef (<http://www.tothenew.com/blog/installing-lamp-stack-through-chef/>)

Chef recipe to deploy Grails war on a new production server (<http://www.tothenew.com/blog/chef-recipe-to-deploy-grails-war-on-a-new-production-server/>)



FOLLOW US ON

(<https://www.facebook.com/tothenew/>) (<https://twitter.com/tothenew/>) (<https://www.linkedin.com/company/tothenew/>)



Subscribe to our Blog

Get latest articles straight to your inbox

[Who We Are](#)

+

[What We Do](#)

+

[Knowledge](#)

+

[Contact Us](#)

+

[Connect With Us](#)