

Ansible & Amazon EC2

Orchestrating NetBSD into the Cloud

pkgsrCon 2013
Berlin, March 23rd 2013
Hubert Feyrer <hubertf@NetBSD.org>

About

- Introducing Ansible
- NetBSD preparation for Ansible
- Milestone 1: Ansible with DB+Web on one local VM
- Milestone 2: Basic Ansible setup for Amazon EC2
... and what is this Amazon EC2 cloud, anyways?!
- Milestone 3: One EC2 instance with both DB+Web
- Milestone 4: Ansible+EC2 with different instances for DB+Web
- Lessons learned
- Future work

Goal

Mozilla Firefox

ec2-184-73-58-92.compute-1.amazonaws.com/webapp/

Showing table hf.names:

id	first	last
1	Donald	Duck
2	Daisy	Duck

Enter new values:

first:

last:

Introducing Ansible

ANSIBLE

HOME

DISCOVER

>DOCS & DOWNLOADS

>GITHUB

>ANSIBLEWORKS

FAQ

RESOURCES



Orchestrate From Above.

Most software does not run on a single machine.

Ansible parallelizes complex multi-tier rollouts across app servers, databases, monitoring servers, and load balancers

Ansible is the easiest way to deploy, manage, and orchestrate computer systems you've ever seen. You can get started in minutes.

Sponsored by

 **AnsibleWorks**

Introducing Ansible

- See also: Puppet, chef
- Describe system state - "playbooks"
- Adjust reality -> "orchestration"
- No daemons left & right
- SSH based
- Copies modules to remote machine and runs them with parameters
- Infrastructure: python, SSH, text files
- Key modules support NetBSD: pkgin, service, ec2

Source: <http://ansible.cc/>

Milestone 1: Ansible with DB+Web on local machine

NetBSD preparation for Ansible

Steps:

1. Setup hardware / VM
2. Do basic NetBSD installation: base, etc
3. From sysinst, install pkgin (yai!)
4. Install ansible binary package
5. Allow root-login via network (yuck!)

Playbook: Basic NetBSD setup

Run:

```
% ansible-playbook \  
-k -i hosts-HF \  
config-netbsd-basic.yml
```

Log: [view](#)

Hosts: [view](#)

Playbook: [view](#)

```
% ansible-playbook -k -i hosts-HF config-netbsd-basic.yml  
SSH password:  
  
PLAY [netbsd] *****  
  
GATHERING FACTS *****  
ok: [10.0.0.181]  
  
TASK: [Install tcsh] *****  
changed: [10.0.0.181]  
  
TASK: [Add user feyrer] *****  
changed: [10.0.0.181]  
  
TASK: [Create ~feyrer/.ssh directory] *****  
changed: [10.0.0.181]  
  
TASK: [Enable ssh login with ssh-key] *****  
changed: [10.0.0.181]  
  
TASK: [Install sudo] *****  
changed: [10.0.0.181]  
  
TASK: [Enable PW-less sudo-access for everyone in group 'wheel'] **  
changed: [10.0.0.181]  
  
TASK: [Disable ssh logins as root] *****  
changed: [10.0.0.181]  
  
NOTIFIED: [restart sshd] *****  
changed: [10.0.0.181]  
  
PLAY RECAP *****  
10.0.0.181 : ok=9 changed=8 unreachable=0
```


Playbook: MySQL database server

Run:

```
% ansible-playbook \  
-i hosts-HF \  
config-netbsd-dbserver.yml
```

Log: [view](#)

Playbook: [view](#)

```
% ansible-playbook -i hosts-HF config-netbsd-dbserver.yml  
  
PLAY [dbservers] *****  
  
GATHERING FACTS *****  
ok: [10.0.0.181]  
  
TASK: [Install mysql] *****  
changed: [10.0.0.181]  
  
TASK: [Install MySQL rc.d script] *****  
changed: [10.0.0.181]  
  
TASK: [Start MySQL service] *****  
changed: [10.0.0.181]  
  
TASK: [Install python-mysqldb (for mysql_user module)] *****  
changed: [10.0.0.181]  
  
TASK: [Setup DB] *****  
changed: [10.0.0.181]  
  
TASK: [Add db-user] *****  
changed: [10.0.0.181]  
  
TASK: [Copy over DB template] *****  
changed: [10.0.0.181]  
  
TASK: [Import DB data] *****  
changed: [10.0.0.181]  
  
PLAY RECAP *****  
10.0.0.181 : ok=9 changed=8 unreach=0
```

Playbook: Apache/PHP server

Run:

```
% ansible-playbook \  
  -i hosts-HF \  
  config-netbsd-webserver.yml
```

Log: [view](#)

Playbook: [view](#)

Webapp: [view](#)

```
TASK: [Enable PHP modules in PHP config file] *****  
changed: [10.0.0.181] => (item={'re': '^extension.*'})  
changed: [10.0.0.181] => (item={'re': '^extension.*'})  
changed: [10.0.0.181] => (item={'re': '^extension.*'})  
changed: [10.0.0.181] => (item={'re': '^extension.*'})  
changed: [10.0.0.181] => (item={'re': '^extension.*'})  
changed: [10.0.0.181] => (item={'re': '^extension.*'})  
changed: [10.0.0.181] => (item={'re': '^extension.*'})  
changed: [10.0.0.181] => (item={'re': '^extension.*'})  
changed: [10.0.0.181] => (item={'re': '^extension.*'})  
changed: [10.0.0.181] => (item={'re': '^extension.*'})  
  
TASK: [Fix Apache access control] *****  
changed: [10.0.0.181]  
  
TASK: [Create directory for webapp] *****  
changed: [10.0.0.181]  
  
TASK: [Deploy example webapp] *****  
changed: [10.0.0.181]  
  
TASK: [Create webapp symlink for easy access] *****  
changed: [10.0.0.181]  
  
NOTIFIED: [restart apache] *****  
changed: [10.0.0.181]  
  
PLAY RECAP *****  
10.0.0.181 : ok=15 changed=14
```

Milestone 2: Basic setup for EC2

Going into the cloud

We have:

- 1 local VM with both DB- and web-server

We want:

- VMs in Amazon EC2 cloud
- 1+ DB-server
- 1+ webservers

NetBSD in Amazon's EC2 cloud

- NetBSD is fully supported in Amazon's Xen-based EC2 cloud
- Steps:
 - Get Amazon AWS account
 - Select Amazon Machine Image (AMI) -> NetBSD
 - Select hardware (CPU/cores, RAM)
 - Create SSH keys
 - Define firewall rules = security groups
 - Start instance
 - Login via SSH

Demo: [AWS Web Console](#)

NetBSD in Amazon's EC2 cloud

top(1) pr0n:

```
Default
load averages: 0.09, 0.08, 0.03; up 0+00:02:41 10:16:30
15 processes: 1 runnable, 12 sleeping, 2 on CPU
CPU0 states: 0.0% user, 0.0% nice, 0.0% system, 0.0% interrupt, 100% idle
CPU1 states: 0.0% user, 0.0% nice, 0.0% system, 0.0% interrupt, 100% idle
CPU2 states: 0.0% user, 0.0% nice, 0.0% system, 0.0% interrupt, 100% idle
CPU3 states: 0.0% user, 0.0% nice, 0.0% system, 0.0% interrupt, 100% idle
CPU4 states: 0.0% user, 0.0% nice, 0.0% system, 0.0% interrupt, 100% idle
CPU5 states: 0.0% user, 0.0% nice, 0.0% system, 0.0% interrupt, 100% idle
CPU6 states: 0.0% user, 0.0% nice, 0.0% system, 0.0% interrupt, 100% idle
CPU7 states: 0.0% user, 0.0% nice, 0.0% system, 0.0% interrupt, 100% idle
Memory: 67M Act, 5800K Exec, 54M File, 66G Free
Swap:

```

PID	USERNAME	PRI	NICE	SIZE	RES	STATE	TIME	WCPU	CPU	COMMAND
0	root	0	0	0K	102M	CPU/7	7:23	0.00%	0.00%	[system]
40	root	43	0	17M	1680K	CPU/0	0:00	0.00%	0.00%	top
594	root	43	0	74M	4904K	RUN/0	0:00	0.00%	0.00%	sshd
797	postfix	85	0	45M	3524K	kqueue/6	0:00	0.00%	0.00%	qmgr
785	postfix	85	0	45M	3456K	kqueue/0	0:00	0.00%	0.00%	pickup
780	root	85	0	45M	3412K	kqueue/0	0:00	0.00%	0.00%	master
435	root	85	0	56M	2380K	select/0	0:00	0.00%	0.00%	sshd
255	root	85	0	23M	1772K	kqueue/0	0:00	0.00%	0.00%	syslogd
816	root	85	0	11M	1372K	pause/0	0:00	0.00%	0.00%	csh
565	root	85	0	13M	1208K	ttyraw/7	0:00	0.00%	0.00%	getty
774	root	85	0	11M	1200K	nanosl/0	0:00	0.00%	0.00%	cron
244	root	85	0	8756K	1048K	select/6	0:00	0.00%	0.00%	dhcpcd
327	root	85	0	13M	948K	kqueue/6	0:00	0.00%	0.00%	powerd
795	root	84	0	15M	1032K	kqueue/7	0:00	0.00%	0.00%	inetd
1	root	83	0	13M	1212K	wait/0	0:00	0.00%	0.00%	init

The euca2ools package

Environment variables:

```
setenv EC2_ACCESS_KEY "AKXXXXXXXXXXXXXXXXXXXX"
setenv EC2_SECRET_KEY "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
setenv EC2_CERT      .../cert-XXX.pem
setenv EC2_PRIVATE_KEY .../pk-XXX.pem
setenv EC2_URL       http://ec2.amazonaws.com
setenv S3_URL        http://s3.amazonaws.com
```

Regions & availability zones:

```
euca-describe-regions
euca-describe-availability-zones
euca-describe-availability-zones --region eu-west-1
```

Change default region:

```
setenv EC2_URL http://ec2.eu-west-1.amazonaws.com
```

AMIs:

```
euca-describe-images --all
```

The euca2ools package, cont'd

SSH Keypairs:

```
euca-add-keypair          -f key-eucaHF.pem eucaHF
euca-describe-keypairs
euca-delete-keypair      eucaHF
```

Instances:

```
euca-run-instances -k eucaHF ami-7fc3c30b
euca-describe-instances
euca-describe-instances i-96a773dc
ssh -i key-eucaHF.pem ec2-54-328-43-220.compute.amazonaws.com -l root
euca-terminate-instances i-96a773dc
```

Links: [Eucalyptus homepage](#)
[Wiki entry: NetBSD Wiki](#)
[Blog entry: hubertf's NetBSD blog](#)

Ansible's ec2 modules & friends

- [ec2:](#) launch instance, wait until up
- [ec2_facts:](#) determine instance facts
- [ec2_vol:](#) create and attach S3 storage

Also:

- [ec2.py](#), [ec2.ini](#)

EC2 preparations for Ansible

1. Setup security (firewall) group:

```
% euca-add-group -d 'Web servers' ec2-webservers
% euca-authorize -P tcp -p 80-80 -s 0.0.0.0/0 ec2-webservers
% euca-authorize -P tcp -p 22-22 -s 0.0.0.0/0 ec2-webservers
% euca-authorize -P icmp -s 0.0.0.0/0 ec2-webservers
%
% euca-describe-groups
GROUP      sg-a854b3c3      749335780469      ec2-webservers      Web servers
PERMISSION  749335780469      ec2-webservers      ALLOWS      tcp      22
PERMISSION  749335780469      ec2-webservers      ALLOWS      tcp      80
PERMISSION  749335780469      ec2-webservers      ALLOWS      icmp     -1
```

2. Load SSH key into agent:

```
% eval `ssh-agent`
Agent pid 9304
% ssh-add -l
The agent has no identities.
% ssh-add ../key-ec2HF.pem
Identity added: ../../euca2ools/key-ec2HF.pem (../../euca2ools/key-ec2HF.pem)
% ssh-add -l
2048 d5:25:19:3d:59:40:35:32:03:f7:c5:83:de:19:b6:d0 ../../euca2ools/key-ec2HF.pem (
```

Playbook: Prepare EC2 instance

Run:

```
% ansible-playbook \  
-i hosts-HF \  
config-ec2-prepare1vm.yml
```

Log: [view](#)

Hosts: [view](#)

Playbook: [view](#)

```
% ansible-playbook -i hosts-HF config-ec2-prepare1vm.yml  
  
PLAY [localhost] *****  
  
TASK: [Launch new EC2 instance] *****  
changed: [127.0.0.1]  
  
TASK: [Give the system 30 seconds to boot up] *****  
changed: [127.0.0.1]  
  
TASK: [Get rid of SSH "Are you sure you want to continue connectin  
changed: [127.0.0.1]  
  
TASK: [Fix /usr/bootstrap.sh to run pkgin with -y] *****  
changed: [127.0.0.1] => (item={'cmd': 'install /usr/bootstrap.sh /  
changed: [127.0.0.1] => (item={'cmd': 'chmod +w /usr/bootstrap.sh'  
changed: [127.0.0.1] => (item={'cmd': 'sed "s,bin/pkgin update,bin'  
changed: [127.0.0.1] => (item={'cmd': 'chmod -w /usr/bootstrap.sh'  
  
TASK: [Install pkgin via /usr/bootstrap.sh] *****  
changed: [127.0.0.1] => (item={'cmd': u'env PATH=/usr/sbin:${PATH}'  
  
TASK: [Copy over Ansible binary package] *****  
changed: [127.0.0.1]  
  
TASK: [Install Ansible dependencies] *****  
changed: [127.0.0.1]  
  
TASK: [Install Ansible package (manually)] *****  
changed: [127.0.0.1]  
  
TASK: [Setup lame /usr/bin/python symlink] *****  
changed: [127.0.0.1]  
  
PLAY RECAP *****  
127.0.0.1 : ok=9 changed=9 unreachable=
```

Playbook: Talking to EC2 instances

Run:

```
% env ANSIBLE_HOSTS=./ec2.py \
  ansible-playbook \
  config-ec2-basic.yml
```

Log: [view](#)

Playbook: [view](#)

```
% env ANSIBLE_HOSTS=./ec2.py ansible-playbook con
PLAY [security_group_ec2-webservers] *****

GATHERING FACTS *****
ok: [ec2-50-16-129-62.compute-1.amazonaws.com]
ok: [ec2-54-235-232-227.compute-1.amazonaws.com]
ok: [ec2-23-20-42-71.compute-1.amazonaws.com]
ok: [ec2-23-20-87-176.compute-1.amazonaws.com]
ok: [ec2-54-242-254-237.compute-1.amazonaws.com]
ok: [ec2-54-234-59-5.compute-1.amazonaws.com]

TASK: [ping] *****
ok: [ec2-54-235-232-227.compute-1.amazonaws.com]
ok: [ec2-50-16-129-62.compute-1.amazonaws.com]
ok: [ec2-23-20-87-176.compute-1.amazonaws.com]
ok: [ec2-23-20-42-71.compute-1.amazonaws.com]
ok: [ec2-54-234-59-5.compute-1.amazonaws.com]
ok: [ec2-54-242-254-237.compute-1.amazonaws.com]

PLAY RECAP *****
ec2-23-20-42-71.compute-1.amazonaws.com : ok=2
ec2-23-20-87-176.compute-1.amazonaws.com : ok=2
ec2-50-16-129-62.compute-1.amazonaws.com : ok=2
ec2-54-234-59-5.compute-1.amazonaws.com : ok=2
ec2-54-235-232-227.compute-1.amazonaws.com : ok=2
ec2-54-242-254-237.compute-1.amazonaws.com : ok=2
```

Milestone 3: One EC2 instance with both DB+Web

Playbook: Prepare EC2 instance

Run:

```
% ansible-playbook \  
-i hosts-HF \  
config-ec2-prepare1vm.yml
```

Log: [view](#)

Hosts: [view](#)

Playbook: [view](#)

```
% ansible-playbook -i hosts-HF config-ec2-prepare1vm.yml  
PLAY [localhost] *****  
TASK: [Launch new EC2 instance] *****  
changed: [127.0.0.1]  
TASK: [Give the system 30 seconds to boot up] *****  
changed: [127.0.0.1]  
TASK: [Get rid of SSH "Are you sure you want to continue connecting  
changed: [127.0.0.1]  
TASK: [Fix /usr/bootstrap.sh to run pkgin with -y] *****  
changed: [127.0.0.1] => (item={'cmd': 'install /usr/bootstrap.sh /usr/  
changed: [127.0.0.1] => (item={'cmd': 'chmod +w /usr/bootstrap.sh'})  
changed: [127.0.0.1] => (item={'cmd': 'sed "s,bin/pkgin update,bin/p  
changed: [127.0.0.1] => (item={'cmd': 'chmod -w /usr/bootstrap.sh'})  
TASK: [Install pkgin via /usr/bootstrap.sh] *****  
changed: [127.0.0.1] => (item={'cmd': 'u'env PATH=/usr/sbin:${PATH} /  
TASK: [Copy over Ansible binary package] *****  
changed: [127.0.0.1]  
TASK: [Install Ansible dependencies] *****  
changed: [127.0.0.1]  
TASK: [Install Ansible package (manually)] *****  
changed: [127.0.0.1]  
TASK: [Setup lame /usr/bin/python symlink] *****  
changed: [127.0.0.1]  
PLAY RECAP *****  
127.0.0.1 : ok=9 changed=9 unreachable=0
```

Playbook: Basic setup of EC2 VM

Run:

```
% env \  
  ANSIBLE_HOSTS=./ec2.py \  
  ansible-playbook \  
  config-ec2-basic.yml
```

Log: [view](#)

Playbook: [view](#)

```
% env ANSIBLE_HOSTS=./ec2.py ansible-playbook config-ec2-b  
PLAY [security_group_ec2-webservers] *****  
TASK: [ping] *****  
ok: [ec2-107-22-69-112.compute-1.amazonaws.com]  
TASK: [Install tcsh] *****  
changed: [ec2-107-22-69-112.compute-1.amazonaws.com]  
TASK: [Add user feyrer] *****  
changed: [ec2-107-22-69-112.compute-1.amazonaws.com]  
TASK: [Create ~feyrer/.ssh directory] *****  
changed: [ec2-107-22-69-112.compute-1.amazonaws.com]  
TASK: [Enable ssh login with ssh-key] *****  
changed: [ec2-107-22-69-112.compute-1.amazonaws.com]  
TASK: [Install sudo] *****  
changed: [ec2-107-22-69-112.compute-1.amazonaws.com]  
TASK: [Enable PW-less sudo-access for everyone in group 'w  
changed: [ec2-107-22-69-112.compute-1.amazonaws.com]  
TASK: [Disable ssh logins as root] *****  
ok: [ec2-107-22-69-112.compute-1.amazonaws.com]  
PLAY RECAP *****  
ec2-107-22-69-112.compute-1.amazonaws.com : ok=8    change
```

Playbook: Setup database server

Run:

```
% env \  
  ANSIBLE_HOSTS=./ec2.py \  
  ansible-playbook \  
  config-ec2-dbserver.yml
```

Log: [view](#)

Playbook: [view](#)

```
% env ANSIBLE_HOSTS=./ec2.py ansible-playbook config-ec2-dbserver.yml  
PLAY [security_group_ec2-webservers] *****  
TASK: [Install mysql] *****  
changed: [ec2-107-22-69-112.compute-1.amazonaws.com]  
TASK: [Install MySQL rc.d script] *****  
changed: [ec2-107-22-69-112.compute-1.amazonaws.com]  
TASK: [Start MySQL service] *****  
changed: [ec2-107-22-69-112.compute-1.amazonaws.com]  
TASK: [Install python-mysqldb (for mysql_user module)] *****  
changed: [ec2-107-22-69-112.compute-1.amazonaws.com]  
TASK: [Setup DB] *****  
changed: [ec2-107-22-69-112.compute-1.amazonaws.com]  
TASK: [Add db-user] *****  
changed: [ec2-107-22-69-112.compute-1.amazonaws.com]  
TASK: [Copy over DB template] *****  
changed: [ec2-107-22-69-112.compute-1.amazonaws.com]  
TASK: [Import DB data] *****  
changed: [ec2-107-22-69-112.compute-1.amazonaws.com]  
PLAY RECAP *****  
ec2-107-22-69-112.compute-1.amazonaws.com : ok=8    changed=0
```


Playbook: Setup web server

Run:

```
% env \  
  ANSIBLE_HOSTS=./ec2.py \  
  ansible-playbook \  
  config-ec2-webserver.yml
```

Log: [view](#)

Playbook: [view](#)

```
TASK: [Enable PHP modules in PHP config file] ****  
changed: [ec2-107-22-69-112.compute-1.amazonaws.co  
changed: [ec2-107-22-69-112.compute-1.amazonaws.co  
changed: [ec2-107-22-69-112.compute-1.amazonaws.co  
changed: [ec2-107-22-69-112.compute-1.amazonaws.co  
changed: [ec2-107-22-69-112.compute-1.amazonaws.co  
changed: [ec2-107-22-69-112.compute-1.amazonaws.co  
changed: [ec2-107-22-69-112.compute-1.amazonaws.co  
changed: [ec2-107-22-69-112.compute-1.amazonaws.co  
changed: [ec2-107-22-69-112.compute-1.amazonaws.co  
changed: [ec2-107-22-69-112.compute-1.amazonaws.co  
  
TASK: [Create directory for webapp] *****  
changed: [ec2-107-22-69-112.compute-1.amazonaws.co  
  
TASK: [Deploy example webapp] *****  
changed: [ec2-107-22-69-112.compute-1.amazonaws.co  
  
TASK: [Create webapp symlink for easy access] ****  
changed: [ec2-107-22-69-112.compute-1.amazonaws.co  
  
NOTIFIED: [restart apache] *****  
changed: [ec2-107-22-69-112.compute-1.amazonaws.co  
  
PLAY RECAP *****  
ec2-107-22-69-112.compute-1.amazonaws.com : ok=14
```

Milestone 4: Ansible and EC2 with separate DB + Web

Playbook: Prepare VMs for Web+DB

Run:

```
% ansible-playbook \  
  -i hosts-HF \  
  config-ec2-prepare-db+web-vm.yml
```

Log: [view](#)

Hosts: [view](#)

Playbook for both VMs: [view](#)

Playbook for 1 VM: [view](#)

```
TASK: [ec2-dbservers | Launch new EC2 instance] **  
changed: [127.0.0.1]  
  
TASK: [ec2-dbservers | Give the system 30 seconds  
changed: [127.0.0.1]  
  
TASK: [ec2-dbservers | Get rid of SSH "Are you sur  
changed: [127.0.0.1]  
  
TASK: [ec2-dbservers | Fix /usr/bootstrap.sh to ru  
changed: [127.0.0.1] => (item={'cmd': 'install /us  
changed: [127.0.0.1] => (item={'cmd': 'chmod +w /u  
changed: [127.0.0.1] => (item={'cmd': 'sed "s,bin/  
changed: [127.0.0.1] => (item={'cmd': 'chmod -w /u  
  
TASK: [ec2-dbservers | Install pkgin via /usr/boot  
changed: [127.0.0.1] => (item={'cmd': u'env PATH=  
  
TASK: [ec2-dbservers | Copy over Ansible binary pa  
changed: [127.0.0.1]  
  
TASK: [ec2-dbservers | Install Ansible dependencie  
changed: [127.0.0.1]  
  
TASK: [ec2-dbservers | Install Ansible package (ma  
changed: [127.0.0.1]  
  
TASK: [ec2-dbservers | Setup lame /usr/bin/python  
changed: [127.0.0.1]  
  
PLAY RECAP *****  
127.0.0.1 : ok=18 changed=1
```

Playbook: Basic setup of EC2 VMs

Run:

```
% env \  
  ANSIBLE_HOSTS=./ec2.py \  
  ansible-playbook \  
  config-ec2-basic.yml
```

Log: [view](#)

Playbook: [view](#)

```
% env ANSIBLE_HOSTS=./ec2.py ansible-playbook config-ec2-basic.yml  
  
PLAY [security_group_ec2-webservers;security_group_ec2-dbservers] ***  
  
TASK: [ping] *****  
ok: [ec2-54-235-44-118.compute-1.amazonaws.com]  
ok: [ec2-54-234-139-151.compute-1.amazonaws.com]  
  
TASK: [Install tcsh] *****  
changed: [ec2-54-235-44-118.compute-1.amazonaws.com]  
changed: [ec2-54-234-139-151.compute-1.amazonaws.com]  
  
TASK: [Add user feyrer] *****  
changed: [ec2-54-234-139-151.compute-1.amazonaws.com]  
changed: [ec2-54-235-44-118.compute-1.amazonaws.com]  
  
TASK: [Create ~feyrer/.ssh directory] *****  
changed: [ec2-54-235-44-118.compute-1.amazonaws.com]  
changed: [ec2-54-234-139-151.compute-1.amazonaws.com]  
  
TASK: [Enable ssh login with ssh-key] *****  
changed: [ec2-54-235-44-118.compute-1.amazonaws.com]  
changed: [ec2-54-234-139-151.compute-1.amazonaws.com]  
  
TASK: [Install sudo] *****  
changed: [ec2-54-235-44-118.compute-1.amazonaws.com]  
changed: [ec2-54-234-139-151.compute-1.amazonaws.com]  
  
TASK: [Enable PW-less sudo-access for everyone in group 'wheel'] ****  
changed: [ec2-54-234-139-151.compute-1.amazonaws.com]  
changed: [ec2-54-235-44-118.compute-1.amazonaws.com]  
  
TASK: [Disable ssh logins as root] *****  
ok: [ec2-54-235-44-118.compute-1.amazonaws.com]  
ok: [ec2-54-234-139-151.compute-1.amazonaws.com]  
  
PLAY RECAP *****  
ec2-54-234-139-151.compute-1.amazonaws.com : ok=8    changed=6    unr  
ec2-54-235-44-118.compute-1.amazonaws.com : ok=8    changed=6    unre
```

Playbook: Setup database server VM

Run:

```
% env \  
  ANSIBLE_HOSTS=./ec2.py \  
  ansible-playbook \  
  config-ec2-dbserver.yml
```

Log: [view](#)

Playbook: [view](#)

```
% env ANSIBLE_HOSTS=./ec2.py ansible-playbook config-ec2-d  
  
PLAY [security_group_ec2-dbservers] *****  
  
TASK: [Install mysql] *****  
changed: [ec2-54-235-44-118.compute-1.amazonaws.com]  
  
TASK: [Install MySQL rc.d script] *****  
changed: [ec2-54-235-44-118.compute-1.amazonaws.com]  
  
TASK: [Start MySQL service] *****  
changed: [ec2-54-235-44-118.compute-1.amazonaws.com]  
  
TASK: [Install python-mysqldb (for mysql_user module)] ***  
changed: [ec2-54-235-44-118.compute-1.amazonaws.com]  
  
TASK: [Setup DB] *****  
changed: [ec2-54-235-44-118.compute-1.amazonaws.com]  
  
TASK: [Add db-user] *****  
changed: [ec2-54-235-44-118.compute-1.amazonaws.com]  
  
TASK: [Copy over DB template] *****  
changed: [ec2-54-235-44-118.compute-1.amazonaws.com]  
  
TASK: [Import DB data] *****  
changed: [ec2-54-235-44-118.compute-1.amazonaws.com]  
  
PLAY RECAP *****  
ec2-54-235-44-118.compute-1.amazonaws.com : ok=8    change
```

Playbook: Setup web server VM

Run:

```
% env \  
  ANSIBLE_HOSTS=./ec2.py \  
  ansible-playbook \  
  config-ec2-webserver.yml
```

Log: [view](#)

Playbook: [view](#)

```
TASK: [Enable PHP modules in PHP config file] ***  
changed: [ec2-54-234-139-151.compute-1.amazonaws.  
changed: [ec2-54-234-139-151.compute-1.amazonaws.  
changed: [ec2-54-234-139-151.compute-1.amazonaws.  
changed: [ec2-54-234-139-151.compute-1.amazonaws.  
changed: [ec2-54-234-139-151.compute-1.amazonaws.  
changed: [ec2-54-234-139-151.compute-1.amazonaws.  
changed: [ec2-54-234-139-151.compute-1.amazonaws.  
changed: [ec2-54-234-139-151.compute-1.amazonaws.  
changed: [ec2-54-234-139-151.compute-1.amazonaws.  
changed: [ec2-54-234-139-151.compute-1.amazonaws.  
  
TASK: [Create directory for webapp] *****  
changed: [ec2-54-234-139-151.compute-1.amazonaws.  
  
TASK: [Deploy example webapp] *****  
changed: [ec2-54-234-139-151.compute-1.amazonaws.  
  
TASK: [Create webapp symlink for easy access] ***  
changed: [ec2-54-234-139-151.compute-1.amazonaws.  
  
NOTIFIED: [restart apache] *****  
changed: [ec2-54-234-139-151.compute-1.amazonaws.  
  
PLAY RECAP *****  
ec2-54-234-139-151.compute-1.amazonaws.com : ok=1
```

Playbook: Connect Web-VM & DB-VM

Run:

```
% env \
ANSIBLE_
ansible
config-
% env ANSIBLE_HOSTS=./ec2.py ansible-playbook config-ec2-conne
PLAY [security_group_ec2-webservers;security_group_ec2-dbserve
TASK: [Collect EC2 host information] *****
ok: [ec2-54-234-139-151.compute-1.amazonaws.com]
ok: [ec2-54-235-44-118.compute-1.amazonaws.com]
TASK: [Prepare connection-script in /tmp/do-connect-vm.sh] **
changed: [ec2-54-234-139-151.compute-1.amazonaws.com]
changed: [ec2-54-235-44-118.compute-1.amazonaws.com]
TASK: [Run connection-script] *****
changed: [ec2-54-234-139-151.compute-1.amazonaws.com]
changed: [ec2-54-235-44-118.compute-1.amazonaws.com]
PLAY RECAP *****
ec2-54-234-139-151.compute-1.amazonaws.com : ok=3      changed=2
ec2-54-235-44-118.compute-1.amazonaws.com : ok=3      changed=2
```

Log: [view](#)

Playbook

Script: [view](#)

Lessons learned

1. NetBSD goes well into the Cloud
2. ... with some Amazon glitches at times
3. Ansible is useful with NetBSD & EC2
4. ... and is somewhat in motion
5. Using Ansible with or without Amazon EC2 looks useful:
 - reliable
 - repeatable
 - human readable
 - extensible
 - free, with commercial support

Future work

Steps:

1. Move to >1 DB/webserver - loadbalancing
(And while there, go from "push" to "pull"?)
2. Document!

Summary

- Introducing Ansible
- NetBSD preparation for Ansible
- Milestone 1: Ansible with DB+Web on one local VM
- Milestone 2: Basic Ansible setup for Amazon EC2
... and what is this Amazon EC2 cloud, anyways?!
- Milestone 3: One EC2 instance with both DB+Web
- Milestone 4: Ansible+EC2 with different instances for DB+Web
- Lessons learned
- Future work


Thanks!

Questions?

Hubert Feyrer <hubertf@NetBSD.org>

Backup

AWS - Overview

 **Services** ▾ **Edit** ▾ Hubert Feyrer ▾ Global ▾ Help ▾

Welcome

The AWS Management Console provides a graphical interface to Amazon Web Services. Learn more about how to use our services to meet your needs, or get started by selecting a service.


[Getting started guides](#)

[Reference architectures](#)

[Free Usage Tier](#)






Set Start Page

[Console Home](#) ▾





 **AWS Marketplace**
Find & buy software, launch with 1-Click and pay by the hour.

Amazon Web Services





Compute & Networking

-  **Direct Connect**
Dedicated Network Connection to AWS
-  **EC2**
Virtual Servers in the Cloud
-  **Elastic MapReduce**
Managed Hadoop Framework
-  **Route 53**
Scalable Domain Name System
-  **VPC**
Isolated Cloud Resources







Storage & Content Delivery

-  **CloudFront**
Global Content Delivery Network
-  **Glacier**
Archive Storage in the Cloud
-  **S3**
Scalable Storage in the Cloud
-  **Storage Gateway**
Integrates On-Premises IT Environments with Cloud Storage







Database

-  **DynamoDB**
Predictable and Scalable NoSQL Data Store
-  **ElastiCache**
In-Memory Cache
-  **RDS**
Managed Relational Database Service
-  **Redshift** **NEW**
Managed Petabyte-Scale Data Warehouse Service

Deployment & Management

-  **CloudFormation**
Templated AWS Resource Creation
-  **CloudWatch**
Resource and Application Monitoring
-  **Data Pipeline**
Orchestration for Data-Driven Workflows
-  **Elastic Beanstalk**
AWS Application Container
-  **IAM**
Secure AWS Access Control
-  **OpsWorks** **NEW**
DevOps Application Management Service

App Services

-  **CloudSearch**
Managed Search Service
-  **Elastic Transcoder** **NEW**
Easy-to-use Scalable Media Transcoding
-  **SES**
Email Sending Service
-  **SNS**
Push Notification Service
-  **SQS**
Message Queue Service
-  **SWF**
Workflow Service for Coordinating Application Components

Announcements

- [Amazon DynamoDB Reduces Prices](#)
- [AWS Free Usage Tier Now Includes Amazon ElastiCache](#)
- [Announcing New Lower Pricing for Amazon EC2 Reserved Instances](#)
- [More...](#)

Service Health [Edit](#)

Click [Edit](#) to add at least one service and at least one region to monitor.

[Service Health Dashboard](#)

AWS - Community AMI for NetBSD

Request Instances Wizard

Cancel 

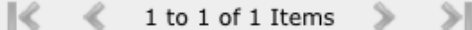
 **CHOOSE AN AMI** INSTANCE DETAILS CREATE KEY PAIR CONFIGURE FIREWALL REVIEW




Choose an Amazon Machine Image (AMI) from one of the tabbed lists below by clicking its **Select** button.

Quick Start

My AMIs

Community AMIs

Viewing: 64-bit  1 to 1 of 1 Items

AMI ID	Root Device	Manifest	Platform	
 ami-9df0cee9	ebs	983624114127/NetBSD-5.1	 Other Linux	Select 

AWS - Select CPU & Memory

Request Instances Wizard Cancel

CHOOSE AN AMI **INSTANCE DETAILS** CREATE KEY PAIR CONFIGURE FIREWALL REVIEW

Provide the details for your instance(s). You may also decide whether you want to launch your instances as "on-demand" or "spot" instances.

Number of Instances: **Instance Type:**

Launch as an EBS-Optimized Instance (if applicable):

Launch Instances

EC2 Instances let you provision virtual servers in the cloud. They are commonly large fixed capacity instances.

Launch into:

Request Spot Instances

Type	CPU Units	CPU Cores	Memory
T1 Micro (t1.micro) ★ Free tier eligible	Up to 2 ECUs	1 Core	613 MiB
M1 Small (m1.small)	1 ECU	1 Core	1.7 GiB
M1 Medium (m1.medium)	2 ECUs	1 Core	3.7 GiB
M1 Large (m1.large)	4 ECUs	2 Cores	7.5 GiB
M1 Extra Large (m1.xlarge)	8 ECUs	4 Cores	15 GiB
M3 Extra Large (m3.xlarge)	13 ECUs	4 Cores	15 GiB
M3 Double Extra Large (m3.2xlarge)	26 ECUs	8 Cores	30 GiB
M2 High-Memory Extra Large (m2.xlarge)	6.5 ECUs	2 Cores	17.1 GiB
M2 High-Memory Double Extra Large (m2.2xlarge)	13 ECUs	4 Cores	34.2 GiB
M2 High-Memory Quadruple Extra Large (m2.4xlarge)	26 ECUs	8 Cores	68.4 GiB
C1 High-CPU Medium (c1.medium)	5 ECUs	2 Cores	1.7 GiB
C1 High-CPU Extra Large (c1.xlarge)	20 ECUs	8 Cores	7 GiB
High Storage Eight Extra Large (hs1.8xlarge)	35 ECUs	16 Cores	117 GiB

[< Back](#) [Continue](#)

AWS - Create SSH key

Request Instances Wizard Cancel


CHOOSE AN AMI INSTANCE DETAILS **CREATE KEY PAIR** CONFIGURE FIREWALL REVIEW


Public/private key pairs allow you to securely connect to your instance after it launches. To create a key pair, enter a name and click **Create & Download your Key Pair**. You will then be prompted to save the private key to your computer. Note, you only need to generate a key pair once - not each time you want to deploy an Amazon EC2 instance.

Choose from your existing Key Pairs

Create a new Key Pair

1. Enter a name for your key pair:* (e.g., jdoekey)

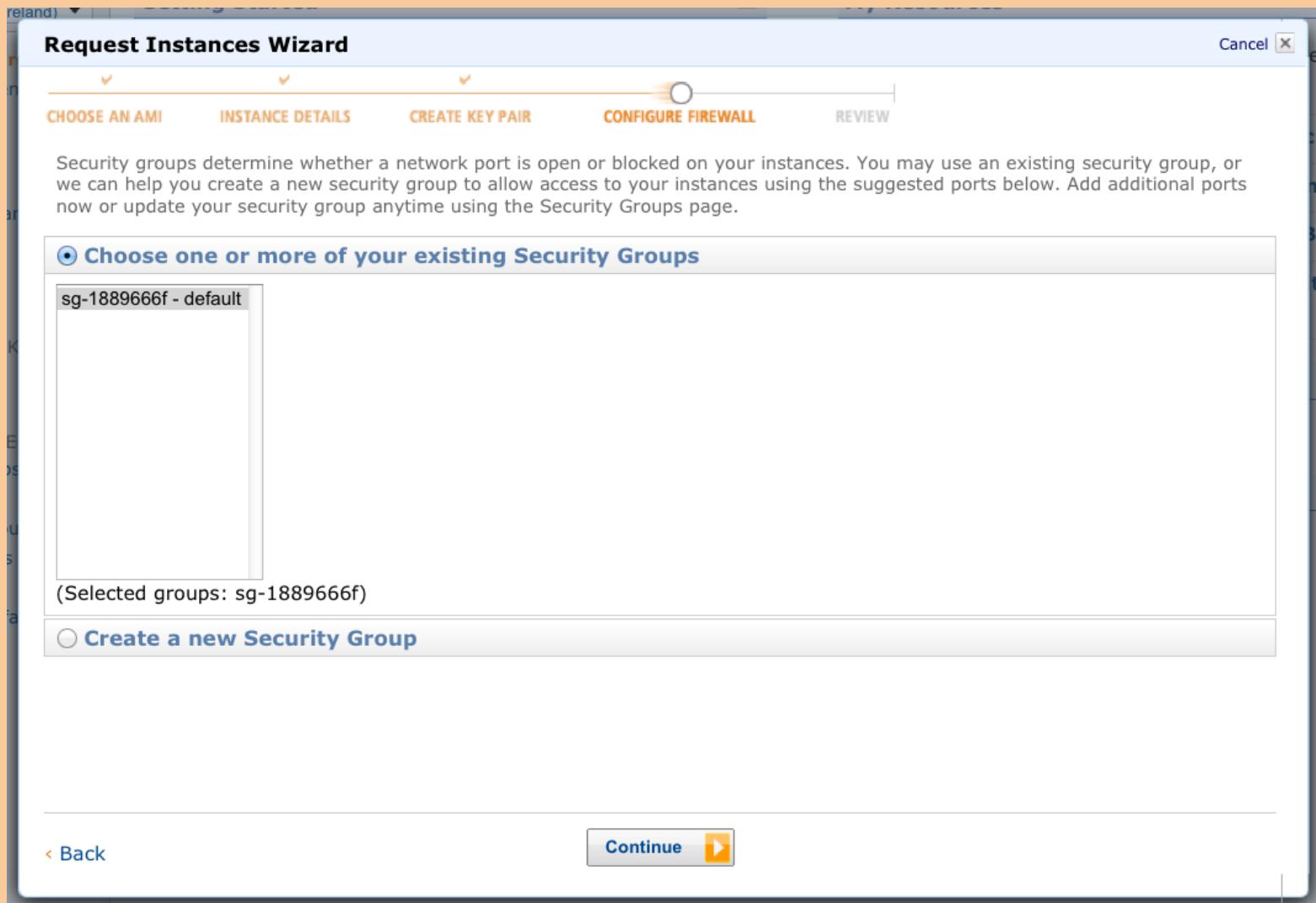
2. Click to create your key pair:* 

 Save this file in a place you will remember. You can use this key pair to launch other instances in the future or visit the Key Pairs page to create or manage existing ones.

Proceed without a Key Pair

< Back Continue >

AWS - Firewall rules



Request Instances Wizard Cancel

CHOOSE AN AMI **INSTANCE DETAILS** **CREATE KEY PAIR** **CONFIGURE FIREWALL** REVIEW

Security groups determine whether a network port is open or blocked on your instances. You may use an existing security group, or we can help you create a new security group to allow access to your instances using the suggested ports below. Add additional ports now or update your security group anytime using the Security Groups page.

Choose one or more of your existing Security Groups

sg-1889666f - default

(Selected groups: sg-1889666f)

Create a new Security Group

[< Back](#) [Continue](#)

AWS - Instance running

AWS Management Console **Amazon EC2** Hubert Feyrer | Help

[Elastic Beanstalk](#) [S3](#) **Amazon EC2** [VPC](#) [CloudWatch](#) [Elastic MapReduce](#) [CloudFront](#) [CloudFormation](#) [RDS](#) [ElastiCache](#) [SQS](#) [IAM](#) [SNS](#) [SES](#) [Route 53](#) [DynamoDB](#) [More...](#)

Navigation

Region: EU West (Ireland)

- EC2 Dashboard
- Scheduled Events
- INSTANCES
 - Instances**
 - Spot Requests
 - Reserved Instances
- IMAGES
 - AMIs
 - Bundle Tasks
- ELASTIC BLOCK STORE
 - Volumes
 - Snapshots
- NETWORK & SECURITY
 - Security Groups
 - Elastic IPs
 - Placement Groups
 - Load Balancers
 - Key Pairs
 - Network Interfaces

My Instances

[Launch Instance](#) [Instance Actions](#) [Show/Hide](#) [Refresh](#) [Help](#)

Viewing: All Instances All Instance Types 1 to 1 of 1 Instances

	Name	Instance	AMI ID	Root Device	Type	State	Status Checks	Monitoring	Security Groups
<input type="checkbox"/>	<i>empty</i>	i-880340c1	ami-9df0cee9	ebs	t1.micro	● running	initializing...	basic	default

AWS - Xen console

System Log: i-90a6a3d9

Cancel X

```
Your identification has been saved in /etc/ssh/ssh_host_rsa_key.
Your public key has been saved in /etc/ssh/ssh_host_rsa_key.pub.
The key fingerprint is:
5c:99:51:42:63:f1:ef:6e:0a:a2:9f:1f:ab:1c:18:dd root@ip-10-226-115-71
The key's randomart image is:
+--[ RSA 1024]-----+
  |      .Bo.      |
  |       .B       |
  |        +       |
  |     o o .      |
  |    .S E .      |
  |     o          |
  |    . o o .     |
  |   o + + . .   |
  |  ..=oo .o.   |
+-----+
Starting sshd.
cprng 442129d1: WARNING insufficient entropy at creation.

cprng 4350d0: WARNING insufficient entropy at creation.

ec2: Setting EC2 SSH key pair: nb6key
ec2: Setting EC2 hostname: ip-10-226-115-71.eu-west-1.compute.internal
ec2: #####
ec2: -----BEGIN SSH HOST KEY FINGERPRINTS-----
ec2: 1024 92:af:9f:e9:46:db:7a:84:fa:7c:3a:7c:bl:23:ce:bf root@ip-10-226-115-71 (DSA)
ec2: 521 91:77:0d:d8:58:2a:e4:9d:83:fe:97:90:88:al:b5:a6 root@ip-10-226-115-71 (ECDSA)
ec2: 1024 5c:99:51:42:63:f1:ef:6e:0a:a2:9f:1f:ab:1c:18:dd root@ip-10-226-115-71 (RSA)
ec2: -----END SSH HOST KEY FINGERPRINTS-----
ec2: #####
Starting local daemons:.
Updating motd.
Starting powerd.
cprng 45197d0: WARNING insufficient entropy at creation.

postfix: rebuilding /etc/mail/aliases (missing /etc/mail/aliases.db)
Starting inetd.
Starting cron.
The following components reported failures:
  rcorder /etc/rc.d/swap2
See /var/run/rc.log for more information.
Sat Mar 24 23:43:32 UTC 2012

NetBSD/amd64 (ip-10-226-115-71.eu-west-1.compute.internal) (console)

login: cprng 4356d0: WARNING insufficient entropy at creation.

cprng 5281d1: WARNING insufficient entropy at creation.

cprng 5282d2: WARNING insufficient entropy at creation.

cprng 8901d1: WARNING insufficient entropy at creation.
```

Close

AWS - SSH login - help

Connect to an instance Cancel

Instance: i-90a6a3d9

▼ **Connect with a standalone SSH Client**

To access your instance:

1. Open an SSH client.
2. Locate your private key file (nb6key.pem). The wizard automatically detects the key you used to launch the instance.
3. Your key file must not be publicly viewable for SSH to work. Use this command if needed:
`chmod 400 nb6key.pem`
4. Connect to your instance using its Public DNS.
[ec2-176-34-219-219.eu-west-1.compute.amazonaws.com].

Example

Enter the following command line:

```
ssh -i nb6key.pem root@ec2-176-34-219-219.eu-west
```

[Connect from a Windows client using PuTTY](#)

▶ [Connect from your browser using the MindTerm SSH Client \(Java Required\)](#)

Close

AWS - SSH login

```
Default

promise% chmod 400 nb6key.pem
promise% ls -la nb6key.pem
-rw----- 1 feyrer  staff  1692 Mar 25 00:41 nb6key.pem
promise% ssh -i nb6key.pem root@ec2-176-34-219-219.eu-west-1.compute.amazonaws.com
NetBSD 6.0_BETA (EC2) #2: Fri Feb 24 13:32:44 PST 2012
Welcome to NetBSD - Amazon EC2 image!

This system is running a snapshot of a stable branch of the NetBSD
operating system, adapted for running on the Amazon EC2 infrastructure.

The environment is very similar to one provided within a typical Xen domU
installation. It contains a small, autonomous environment (including a
compiler toolchain) that you can run to build your own system.

The file system is lightly populated so you have plenty of space to play with.
Should you need a src or pkgsrc tree, please use the "bootstrap" script found
under /usr to download them. You can also use the script to set up
binary packages using "pkgin":

    /usr/bootstrap.sh [src|pkgsrc|binpkg]

This AMI sends email to the maintainer on first boot, to help get
an idea of what is in use at any given time.

You are encouraged to test this image as thoroughly as possible. Should you
encounter any problem, please report it back to the development team using the
send-pr(1) utility (requires a working MTA). If yours is not properly set up,
use the web interface at: http://www.NetBSD.org/support/send-pr.html

Thank you for helping us test and improve NetBSD's quality!
Terminal type is vt220.
We recommend that you create a non-root account and use su(1) for root access.
ip-10-226-115-71#
```