DuckCorp Infrastructure - Review #519

Review burp role

2017-04-03 14:00 - Pierre-Louis Bonicoli

Status:	In Progress	**Start date:**	2017-04-03
Priority:	Normal		
Assignee:	Marc Dequènes		
Category:	Service :: Backup		
Target version:			
Branch:	master		

Description

The Burp role is available here: https://gitlab.com/pilou-/ansible-role-burp.

Related issues:

Related to DuckCorp Infrastructure - Review #518: Review branch backup	**In Progress**	**2017-04-03**
Related to DuckCorp Infrastructure - Enhancement #497: Change Backup System	**In Progress**	**2017-05-09**

History

#1 - 2017-04-03 14:00 - Pierre-Louis Bonicoli

- Related to Review #518: Review branch backup added

#2 - 2017-04-05 06:49 - Marc Dequènes

- Category set to Service :: Backup

#3 - 2017-04-07 16:34 - Marc Dequènes

- Category deleted (Service :: Backup)

- Status changed from New to In Progress

Let's begin reading this complex role. First, a user point of view without reading the code.

README.md:

- s/roles/roles:/ in the examples
- 'server' or '_server'?
- why use underscore for a variable anyway?
- s/vmail:/ user: vmail/ I guess

So there's 'burp:', 'burp_clients:', but in the "Another client configuration example" there is 'client:' and the code seems to use this entry too, so honestly I'm totally lost.

paths keys are unused? so why have them in the first place?

There are flat keys like 'burp_client_can_delete' and dicts, why such a mix? we should probably clarify what is common/client/server.

I guess you wanted to specify which server to use, but why not make it a parameter for each client? It would allow to split backup data in various location, if you for example wish to split the load or storage need.

#4 - 2017-04-07 17:41 - Pierre-Louis Bonicoli

Marc Dequènes wrote:

> Let's begin reading this complex role. First, a user point of view without reading the code.
>
> *README.md*:
>
> - s/roles/roles:/ in the examples

Done (both branches rebased).

- 'server' or '_server'?

_server

- why use underscore for a variable anyway?

I want the burp server at this level in the configuration structure and I need a way to differentiate this key from the others: other keys are usernames (username can not contain underscore). We can user burp_server instead, what do you think ?

- s/vmail:/ user: vmail/ I guess

No

> So there's 'burp:', 'burp_clients:', but in the "Another client configuration example" there is 'client:' and the code seems to use this entry too, so honestly I'm totally lost.

The client entry in Another client configuration example was wrong (s/client:/burp_client:/).

> paths keys are unused? so why have them in the first place?

It's a way to group some paths together:

```
paths:
  databases:
    - '/var/lib/postgresql/automated_backups'
  homes:
    - '/home/user1'
    - '/home/anotheruser'
```

Besides it allows to define a default configuration in group/all:

```
backups:
  important_data:
    - /home
    - /srv
```

and to add (remember that hash_behaviour is set to merge in ansible.cfg) other paths from the playbook level:

```
vars:
```

```
  backups:
    secondary_data:
      - /var/path/to/data
# backups contains important_data and secondary_data
```

> There are flat keys like 'burp_client_can_delete' and dicts, why such a mix? we should probably clarify what is common/client/server.

burp_client_can_* are role variables: their default values are defined in default/main.yml. They refer to client_can_* options. They are listed in the Server default variables section and are used by the server only. These variables can easily be overridden. burp and burp_client don't have default values.

> I guess you wanted to specify which server to use, but why not make it a parameter for each client? It would allow to split backup data in various location, if you for example wish to split the load or storage need.

In order to split backup data in various location, you need to define various server and because a client is linked to a specific server:

- there is a password for each known client stored in the server configuration
- a client configuration refers to one specific server

you would need to define one client per server.

Moving the currently named _server parameter to a lower level is possible (but I don't see any advantage).

#5 - 2017-04-07 18:03 - Marc Dequènes

'tasks/main.yml': please use YAML-style for apt action.

Btw we should stop using 'apt' and switch to 'package'.

I think we should install rsyslog configuration if rsyslog is installed (which one may decide to remove to only use systemd-journald). In this case the corresponding logrotate should not be installed too. If we agree on this, this is very low priority though.

I think the README should make clear it is in charge of installing the certs and scripts and the path should be the absolute; the example could use the Ansible inventory_dir to make it clear.

Even if the server's homedir should not be created, shouldn't it be specified?

burp_server_conf is unused.

ssl_compression shouldn't be deactivated? I thought I remember SSL compression poses security problems.

Instead of an empty '_notify.conf' file there is no way to have a 'conf.d' directory?

Shouldn't 'Render BURP clientconfdir configuration' be delegated to the server?

Things like "set client = client.value" are confusing when you're relying on the variable names to help you follow the whole complexe path. Same for backup.

#6 - 2017-04-07 18:05 - Marc Dequènes

- Category set to Service :: Backup

#7 - 2017-04-07 18:47 - Marc Dequènes

Pierre-Louis Bonicoli wrote:

- 'server' or '_server'?

> _server

Well, there's 'server:' in the second example, so I'm confused.

- why use underscore for a variable anyway?

> I want the burp server at this level in the configuration structure and I need a way to differentiate this key from the others: other keys are usernames (username can not contain underscore). We can user burp_server instead, what do you think ?

Maybe.

- s/vmail:/ user: vmail/ I guess

> No

Well 'No' does not help understand one's mistake.
I think I understand now that I read the whole role. But the second example does not have this level of structure, and there is a 'user:' key which seems to be unneeded because it is defined in 'tasks/client_specific_conf.yml': "user: '{{ client.key }}'" (IIUC this time)

> paths keys are unused? so why have them in the first place?

> It's a way to group some paths together:
> and to add (remember that hash_behaviour is set to merge in ansible.cfg) other paths from the playbook level:

ok, nice idea.

> There are flat keys like 'burp_client_can_delete' and dicts, why such a mix? we should probably clarify what is common/client/server.

burp_client_can_* are role variables: their default values are defined in default/main.yml. They refer to client_can_* options. They are listed in the Server default variables section and are used by the server only. These variables can easily be overridden. burp and burp_client don't have default values.

You can have default values as structure too. It's just disconcerting to have a mix.
But it's true if people don't use the same hash_behaviour this may be difficult.

> Moving the currently named _server parameter to a lower level is possible (but I don't see any advantage).

What if you want a client to use a different server then?

#8 - 2017-04-08 00:35 - Pierre-Louis Bonicoli

Marc Dequènes wrote:

> 'tasks/main.yml': please use YAML-style for apt action.

Done

> Btw we should stop using 'apt' and switch to 'package'.

update_cache is not available when using package.

> I think we should install rsyslog configuration if rsyslog is installed (which one may decide to remove to only use systemd-journald). In this case the corresponding logrotate should not be installed too. If we agree on this, this is very low priority though.

Done. But one should not decide that ;)

> I think the README should make clear it is in charge of installing the certs and scripts and the path should be the absolute; the example could use the Ansible inventory_dir to make it clear.

I added one comment in README about certificates installation. I don't understand the second part: path could be absolute or relative.

> Even if the server's homedir should not be created, shouldn't it be specified?

Nor user module neither useradd will complain. What do you think of adding home: /nonexistent ?

> *burp_server_conf* is unused.

Done (removed).

> ssl_compression shouldn't be deactivated? I thought I remember SSL compression poses security problems.

It seems these security problems (CRIME, BREACH) are related to HTTP only.

> Instead of an empty '_notify.conf' file there is no way to have a 'conf.d' directory?

No :-/

> Shouldn't 'Render BURP clientconfdir configuration' be delegated to the server?

It is (delegate_to is at the block level).

> Things like "set client = client.value" are confusing when you're relying on the variable names to help you follow the whole complexe path. Same for backup.

Nothing updated.

#9 - 2017-04-11 11:29 - Marc Dequènes

Pierre-Louis Bonicoli wrote:

> update_cache is not available when using package.

Then I think we should forget about update_cache and it is something to be improved in the package module.

> Nor user module neither useradd will complain. What do you think of adding home: /nonexistent ?

True, but in this case what is chosen? It is related to logins.def settings maybe. I don't really like having something in /home for example, even if unused and even uncreated. /nonexistent is fine.

> It seems these security problems (CRIME, BREACH) are related to HTTP only.

According to WP TLS 1.3 includes the following changes: « Dropping support for many unsecure or obsolete features including compression, renegotiation,… ». So, IIUC we'd better disable compression altogether.

> > Things like "set client = client.value" are confusing when you're relying on the variable names to help you follow the whole complexe path. Same for backup.
>
> Nothing updated.

What does it mean?t could be postponed, but readability of the role is useful too for maintenance. So do you agree on this difficulty or do you think it's useless?

#10 - 2017-04-11 11:40 - Marc Dequènes

I see in 4dc2812 uses ansible_service_mgr, which is future-proof, so this is nice. Nevertheless files are in a subdirectory but includes use underscores. I would vote for the subdirectory, but either is ok.

#11 - 2017-04-11 12:22 - Pierre-Louis Bonicoli

Marc Dequènes wrote:

> Pierre-Louis Bonicoli wrote:
>
> - 'server' or '_server'?
>
> _server
>
> Well, there's 'server:' in the second example, so I'm confused.

Second example fixed.

- why use underscore for a variable anyway?

> I want the burp server at this level in the configuration structure and I need a way to differentiate this key from the others: other keys are usernames (username can not contain underscore). We can user burp_server instead, what do you think ?
>
> Maybe.

Nothing done.

- s/vmail:/ user: vmail/ I guess

> No
>
> Well 'No' does not help understand one's mistake.
> I think I understand now that I read the whole role. But the second example does not have this level of structure, and there is a 'user:' key which seems to be unneeded because it is defined in 'tasks/client_specific_conf.yml': "user: '{{ client.key }}'" (IIUC this time)

Second example fixed.

> paths keys are unused? so why have them in the first place?
>
> It's a way to group some paths together:
> and to add (remember that hash_behaviour is set to merge in ansible.cfg) other paths from the playbook level:

ok, nice idea.

> There are flat keys like 'burp_client_can_delete' and dicts, why such a mix? we should probably clarify what is common/client/server.

burp_client_can_* are role variables: their default values are defined in default/main.yml. They refer to client_can_* options. They are listed in the Server default variables section and are used by the server only. These variables can easily be overridden. burp and burp_client don't have default values.

You can have default values as structure too. It's just disconcerting to have a mix.
But it's true if people don't use the same hash_behaviour this may be difficult.

Note that using hash_behaviour=merge isn't recommended.

> Moving the currently named _server parameter to a lower level is possible (but I don't see any advantage).

What if you want a client to use a different server then?

You need two clients (one client per server).

#12 - 2017-04-12 18:41 - Marc Dequènes

Pierre-Louis Bonicoli wrote:

> Note that using hash_behaviour=merge isn't recommended.

You're right, but it was a desired behavior in previous softwares (with hiera for example) and I was never able to see any reasoning explaining why it is bad.
Being able to auto-merge based on group_vars and host_vars seem a very important feature to me, and we're already using it for many things.
Having a Jinja filter is useful but you cannot plan all the useful merges in your playbook.

Thanks for all the explanations. When you have time I think we can deploy and test the DB backup. Also mere files are also useful, like mails which are critical, and we do not have to wait for a perfect and complete PR to do that (as currently there is no competition).

#13 - 2017-05-09 16:06 - Pierre-Louis Bonicoli

- Related to Enhancement #497: Change Backup System added

#14 - 2017-05-10 18:38 - Marc Dequènes

Certs are installed with a special group but mode 0400 so this group has in fact no access to the files :-/
(just borrowed you code and…)

#15 - 2017-07-16 10:04 - Marc Dequènes

So a new review based on master/406a38a.

It's not an easy role so please do not get angry by my silly questions.

Many things are similar but the logic on the playbook side changed a lot. I honestly had a hard time following the logic. I was a bit astonished to see the role is like giving up on handling the client config file (at least in our case). So I looked at the difference between duckcorp_ccd.conf and the default config file and could not see the cause of this change. For example the list of keeps: I don't see why backup.keeps is not used anymore, as we already use inventory_hostname in the parameters defined in the playbook. Is this related to Ansible bugs or limitations when templating?

Other point in the template (and divergence):

- the config block seems an unused feature, can this be useful with a different configuration?
- client_lockdir could be the same I guess
- the role could be slightly more opinionated and decide to set the compression parameter as I think most people do not care changing this
- nobackup could be a new parameter in the burp dict or made a convention of the role

Why is there a need for files/paths in with_first_found in server.yml? (why not do like in https://gitlab.com/osas/ansible-role-entropy/blob/master/tasks/main.yml#L3) (I don't really like ../)

We could use content instead of an empty lockfile template.

#16 - 2017-08-26 07:39 - Marc Dequènes

The current rule creating the lockfile overwrites an existing file, thus destroying the lock is the backup is running at the same time.

#17 - 2017-08-26 08:28 - Marc Dequènes

Also Burp unlinks the lockfile at the end of the backup, which means the next one will never occur if not run as root…
Not sure how to solve this problem as we don't want to allow anyone to write files in the lockfiles directory.

#18 - 2017-08-26 08:41 - Marc Dequènes

Another bug when getting the TZ info, fixed by this patch (I'm not on GH):

```
--- a/tasks/client_specific_conf.yml
+++ b/tasks/client_specific_conf.yml
@@ -121,11 +121,15 @@
       gather_subset: '!all'
     delegate_facts: True

+  - name: 'Get backup TZ'
+    set_fact:
+      backup_timer: "{{ hostvars[inventory_hostname]['timer_' + backup.key] }}"
+
   - name: 'Check timezone of the server'
     assert:
-      that: "{{ hostvars[burp_clients._server]['ansible_date_time']['tz'] == backup.value.timer.TZ }}"
-      msg: "Server timezone is {{ hostvars[burp_clients._server]['ansible_date_time']['tz'] }}, expected time
zone: {{ backup.value.timer.TZ }}"
-    when: backup.value.timer.TZ is defined
+      that: "{{ hostvars[burp_clients._server]['ansible_date_time']['tz'] == backup_timer.TZ }}"
+      msg: "Server timezone is {{ hostvars[burp_clients._server]['ansible_date_time']['tz'] }}, expected time
zone: {{ backup_timer.TZ }}"
+    when: backup_timer.TZ is defined

   - name: 'Create BURP spool directory'
     file:
```

#19 - 2017-08-26 09:45 - Marc Dequènes

Ok, so this patch is not needed, the keep and cron parameters in each backups needed to be setup (in the DC file backup). Sorry for the noise.

#20 - 2017-08-26 15:28 - Marc Dequènes

The TZ is repeated for each cron config while this is a host-wide parameter. All these information are all in the same file, so I think it would be more practical to specify it only once.

#21 - 2017-08-27 01:53 - Marc Dequènes

It seems long backup can easily be interrupted, and I don't see any retry feature:

```
2017-08-26 18:40:58: burp[18740] main socket: Got SSL_ERROR_SYSCALL
2017-08-26 18:40:58: burp[18740] main socket: SSL write problem in asfd_do_write_ssl: 5 - 104=Connection reset
 by peer
```

So I wonder why working_dir_recovery_method is set to delete instead of resume.

Also after this problem when I relaunched the backup manually I got:

```
2017-08-27 01:53:56: burp[29996] main socket: unexpected command in asfd_simple_loop(), called from maybe_chec
k_timer(): e:problem with lock file on server
2017-08-27 01:53:56: burp[29996] error in backup
```

It seems /var/local/backups/burp//main/toushirou.duckcorp.org-root-important.lock/toushirou.duckcorp.org-root-important/lockfile stayed around and Burp is unable to cope. Removing it does not solve anything. The file is recreated and the same error is thrown.

```
drwx------ 2 burp-main burp-main 4096 Aug 26 23:54 /var/local/backups/burp//main/toushirou.duckcorp.org-root-i
mportant.lock/toushirou.duckcorp.org-root-important
-rw------- 1 burp-main burp-main 0 Aug 26 23:54 /var/local/backups/burp//main/toushirou.duckcorp.org-root-impo
rtant.lock/toushirou.duckcorp.org-root-important/lockfile
```

[…]

So after investigation the real problem is we have a fs full (and Burp is unable to tell it properly :-/).

#22 - 2017-08-28 11:18 - Marc Dequènes

There is an unclosed quote at the end of templates/cron preventing the crontab to run properly.

#23 - 2017-08-29 01:01 - Pierre-Louis Bonicoli

Marc Dequènes wrote:

> The current rule creating the lockfile overwrites an existing file, thus destroying the lock is the backup is running at the same time.

Lock files are managed by burp, these files should not be created by role (done: 98be9adfd4c530d85daf4e7762b0499fbb275e7c).

> Also Burp unlinks the lockfile at the end of the backup, which means the next one will never occur if not run as root…
> Not sure how to solve this problem as we don't want to allow anyone to write files in the lockfiles directory.

Instead of using one directory containing every lockfiles, one directory per user (for example: one for root, one for postgres) should be created (done: 7ce1a13cae6a8f7ea160c1573f50a6b1931c0412).

#24 - 2017-08-29 01:05 - Pierre-Louis Bonicoli

Marc Dequènes wrote:

> There is an unclosed quote at the end of templates/cron preventing the crontab to run properly.

Quote removed (7e639a82d365d6086cac3c1d4ce510b7f592f709)

#25 - 2017-08-29 04:45 - Marc Dequènes

Thanks.

I think remarks in #14 and #15 were not reviewed, could you have a look to? (no urgency)