Ansible to Conditionally Prompt for a Variable?

Asked 5 years, 1 month ago Active 10 months ago Viewed 24k times



31

I would like to be able to prompt for my super secure password variable if it is not already in the environment variables. (I'm thinking that I might not want to put the definition into .bash profile or one of the other spots.)



This is not working. It always prompts me.

```
\star
7
```

```
vars:
  THISUSER: "{{ lookup('env','LOGNAME') }}"
  SSHPWD: "{{ lookup('env','MY PWD') }}"
vars_prompt:
  - name: "release_version"
   prompt: "Product release version"
   default: "1.0"
   when: SSHPWD == null
```

NOTE: I'm on a Mac, but I'd like for any solutions to be platform-independent.

variables ansible



asked Aug 23 '14 at 21:48





@AnnTheAgile When you hardcode the above variable (SSHPWD) to some dummy / actual value, did the above piece of code work for you, i.e. for NOT prompting you at all? My understanding is, it'll alway prompt you no matter whether SSHPWD was set to null or was set to a valid value (if hardcoded to some value instead of using/depending upon an ENV variable). - Arun Sangal Jan 2 '17 at 13:43 /

6 Answers



According to the replies from the devs and a quick test I've done with the latest version, the vars prompt is run before "GATHERING FACTS". This means that the env var SSHPWD is always null at the time of your check with when .



34

Unfortunately it seems there is no way of allowing the vars_prompt statement at task level.



Michael DeHaan's reasoning for this is that allowing prompts at the task-level would open up the

By using our site, you acknowledge that you have read and understand our Cookie Policy, Privacy Policy, and our Terms of Service.

There's been a decided emphasis in automation in Ansible and asking questions at task level is not something we really want to do.

However, you can still ask vars_prompt questions at play level and use those variables throughout tasks. You just can't ask questions in roles.

And really, that's what I would like to enforce -- if a lot of Galaxy roles start asking questions, I can see that being annoying:)





1 __ good explanation for the ordering of GATHERING FACTS vs vars_prompt part. – Arun Sangal Jan 2 '17 at 11:45



I might be late to the party but a quick way to avoid <code>vars_prompt</code> is to disable the <code>interactive</code> mode by doing that simple trick:

28



This add no control over which vars_prompt to avoid but coupled with default: "my_default" it can be used in a script.

Full example here:

```
---
- hosts: localhost
  vars_prompt:
    - prompt: Enter blah value
    - default: "{{ my_blah }}"
    - name: blah

echo -n | ansible-playbook -e my_blah=blih site.yaml
```

EDIT:

I've found that using the pause module and the prompt argument was doing what I wanted:

```
---
- pause:
    prompt: "Sudo password for localhost "
when: ( env == 'local' ) and
    ( inventory_hostname == "localhost" ) and
    ( hostvars["localhost"]["ansible become password"] is not defined )
```

By using our site, you acknowledge that you have read and understand our Cookie Policy, Privacy Policy, and our Terms of Service.



1 Cool trick. This even works for skipping multiple prompts – vikas027 Oct 4 '16 at 8:21



4

This is indeed not possible by default in Ansible. I understand the reasoning behind not allowing it, yet I think it *can* be appropriate in some contexts. I've been writing an AWS EC2 deploy script, using the blue/green deploy system, and at some point in the role I need to ask the user if a rollback needs to be done if something has gone awry. As said, there is no way to do this (conditionally and/or non-fugly).

So I wrote a very simple Ansible (2.x) action plugin, based on the pause action from the standard library. It a bit spartan in that it only accepts a single key press, but it might be of use. You can find it in a <u>Github gist here</u>. You need to copy the whole Gist file to the <u>action plugins</u> directory of your playbook directory. See the documentation in the file.

edited Mar 21 '16 at 17:34

answered Mar 21 '16 at 16:12

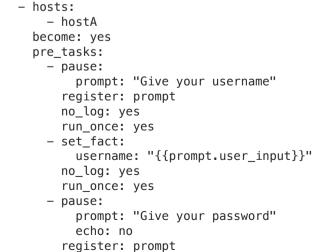


- Pleae include the relevant part of your link in your answer. GitHub depositories may change at any time.
- Thanks. zx485 Mar 21 '16 at 16:36
- The whole file is the relevant part. It's a module than needs to be included wholly to add extra functionality to Ansible. I added a bit more documentation to the anwer. If you mean 'relevant git
- commit': if I find a bug tomorrow I want future visitors to see the newer version. Confiks Mar 21 '16 at 17:46



Based on tehmoon's answer with some modifications I did it that way:

2



no_log: yes
run once: ves

By using our site, you acknowledge that you have read and understand our Cookie Policy, Privacy Policy, and our Terms of Service.

```
tags: [my_role_using_user_pass]
roles:
    - role: my_role_using_user_pass
```

answered Oct 30 '18 at 16:58

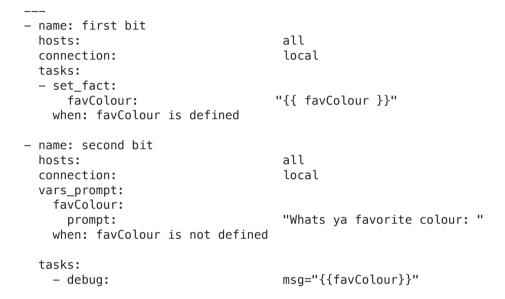




This works for me (2.3) .. do two bits in the one file. This allows me to consruct a tmp vars file when running the playbook via jenkins.. but also allow prompting on the command line

2

And you get to do it with only the one var used



answered Apr 18 '18 at 8:19





As can be seen in the <u>source code</u>, the when keyword isn't implemented for vars_prompt (and in fact never was). The same was mentioned in this Github <u>comment</u>.



The only way in which <code>vars_prompt</code> is currently conditional is that it only prompts when the variable (defined in <code>name</code>) is already defined (using the command-line <code>extra_vars</code> argument).

answered Jul 20 '17 at 12:43

