**Ansible template and variable User Guide**

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision** | **Date** | **Author** | **Description** |
| 0 | 25/01/2017 | Michael Mach | Initial guide for interfaces template |
| 1 | 31/01/2017 | Michael Mach | Added link to raw files via github |
|  |  |  |  |
|  |  |  |  |

Full instructions and original files can be found on github link: <https://github.com/mike-mach/make-Junos-templates>

To create a configuration file using templates in Ansible you need:

1. A template file which is a jinja2( .j2) file format (e.g. You can all it “template.j2”)
2. A variable file which is a .yml file format (e.g. You can call it “template\_vars.yml”)
3. A playbook which is a .yaml file format (e.g. You can call it “main\_play.yaml”)

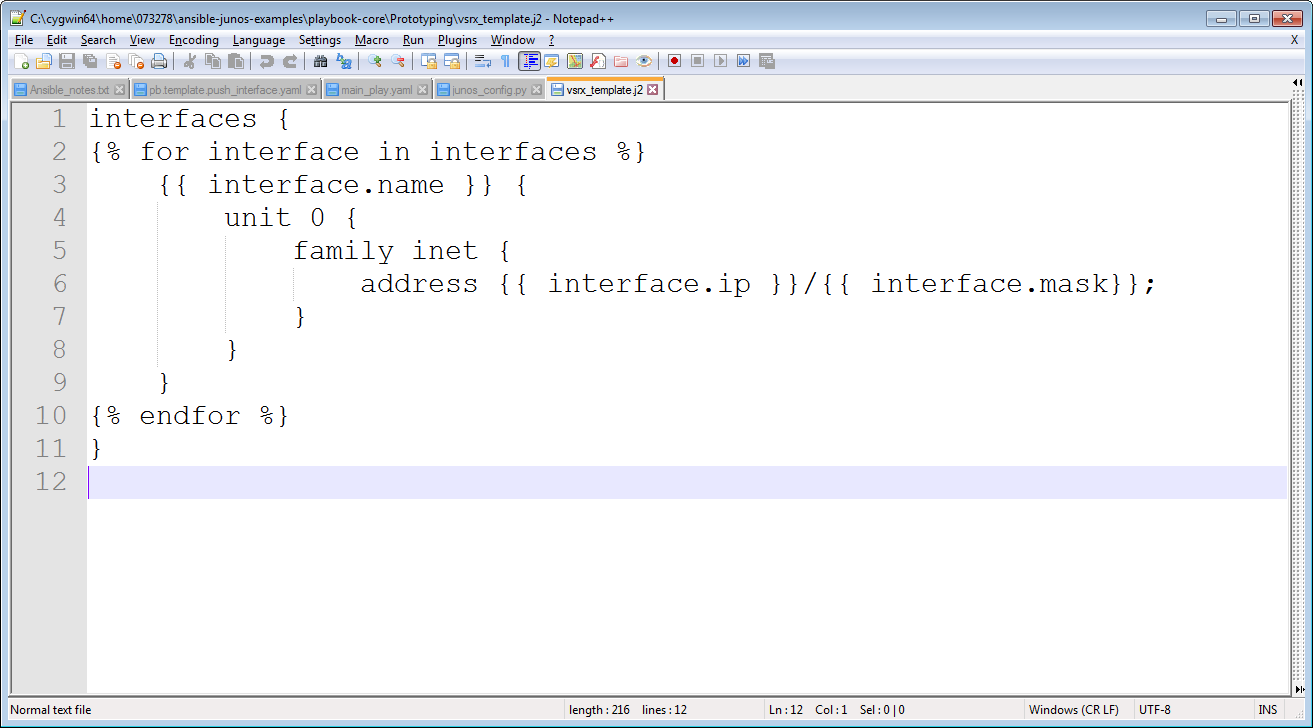
All files can be edited in your preferred text editor, I used Notepad ++

\*Important note: Ansible is space sensitive, make sure indentations and spaces align

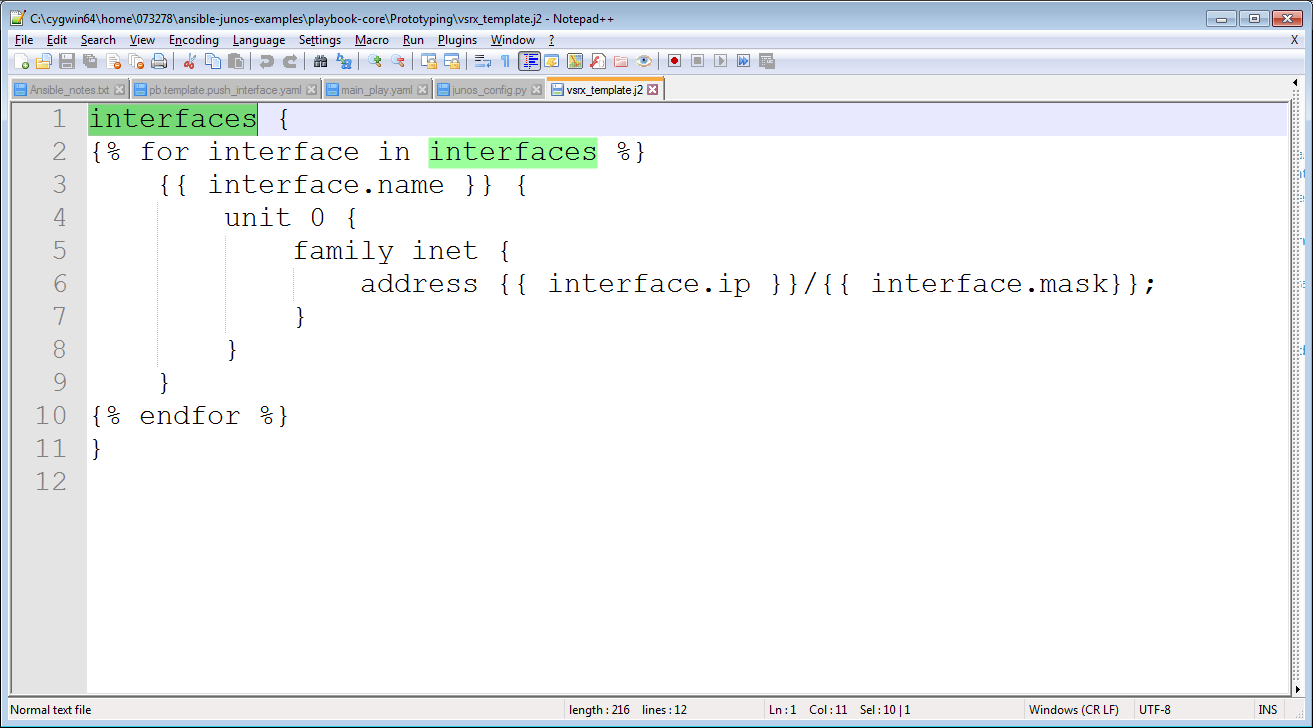
**Template files**

First let’s begin with a template file (jinja2), use an example and explain step by step what each line does

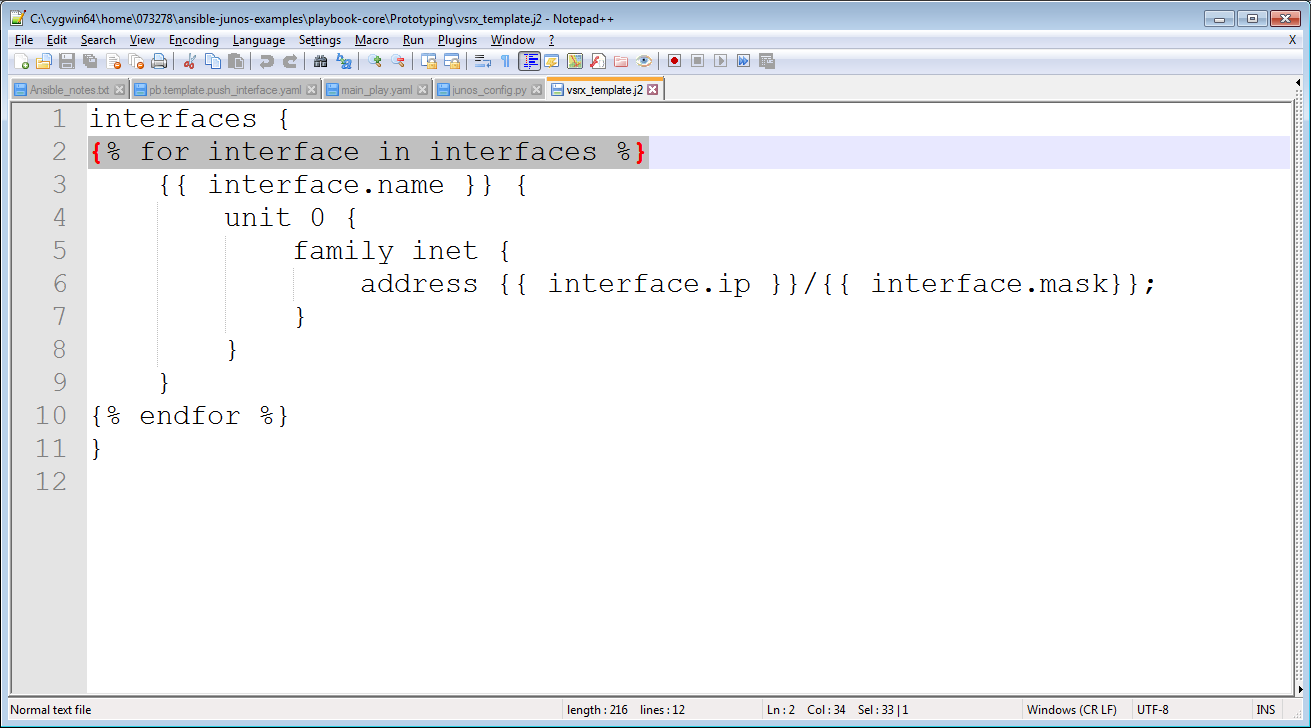
Heres an example of the template I used to create configurations for an interface



So analysing this line by line;



As you can see highlighted in Line 1, ‘interfaces’ indicates the parameter we want to change. Think of it as a heading.



**Line 2** indicates a for loop, which means it looks at each item in the variable file which contains properties of ‘interfaces’.

Note : Notice the ‘interface’ item is not stated in the variable file as it is just an item counter being used in the for loop. The word ‘interface’ can be replaced with any word you like but this name seemed most sensible

**Line 3** looks for the variable ‘name’ in the variable file. The “.” after the item indicates that the item has a property called ‘name’

**Line 4**, when configuring juniper devices they have a structure, in a template file, the tiers of structure are denoted by “{}”, thus line 4 shows the beginning of the ‘unit 0’ structure.

**Line 5**, the follow on structure of ‘unit 0’ is the ‘family inet’ which illustrates that we want to edit the IPv4 configuration.

**Line 6**, shows that we are changing the address properties via the ‘interface.ip’ and ‘interface.mask’ variables.

**Line 7,8,9** are the closing curly brackets for each structure heading and sub-heading.

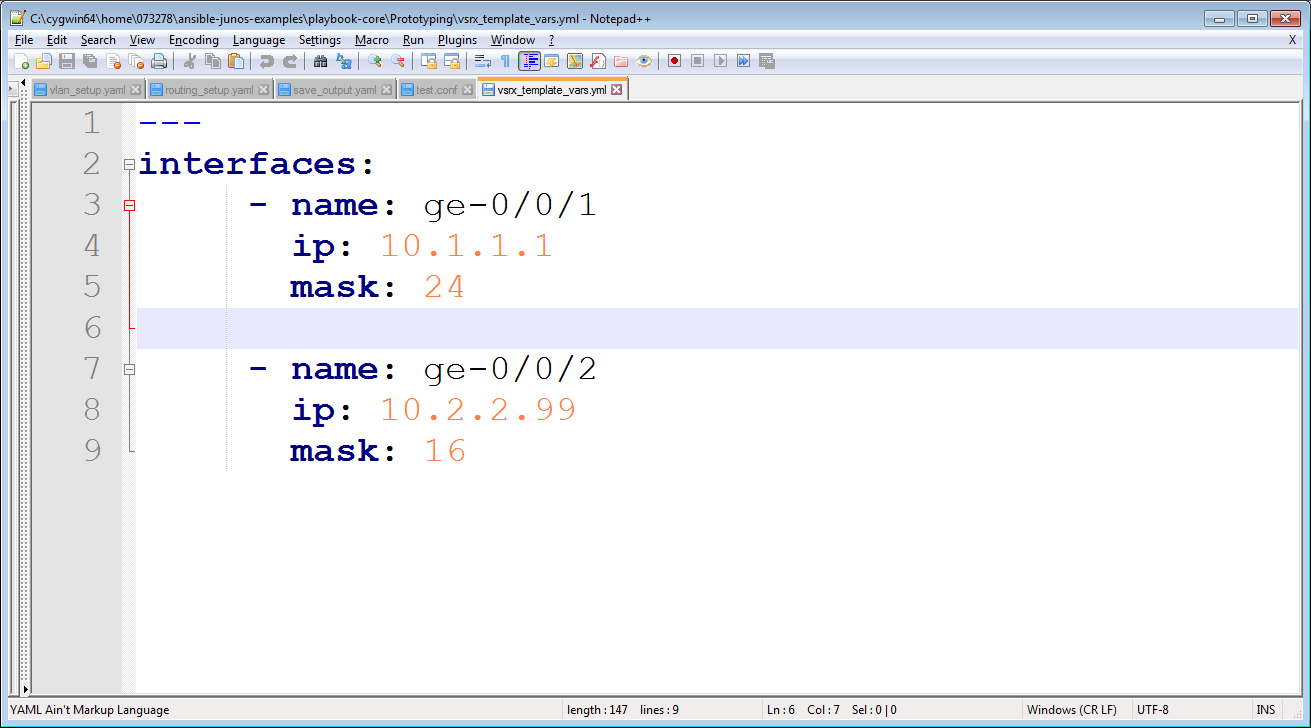
**Line 10** indicates the end of the loop

**Line 11** indicates the end of the ‘interfaces’ heading

There we have it, we can now configure a template for ‘interfaces’ in junos devices. Now let’s have a look at the variable file.

**Variable Files**

Variable files are in the format ‘.yml’ they’re very similar to the playbook format. (Playbooks are mentioned later on)

From the image above you can see the variable file is really simple.

**Line 1** uses 3 dashes to note the start of the file

**Line 2** defines the heading of the structure, the colon is used to assign variables to the heading

**Line 3** uses a “-“ to indicate a list of variables. We want to make changes to the name variable, ‘ge-0/0/1’

**Line 4** then goes onto the ‘ip’ variable which is being set to 10.1.1.1

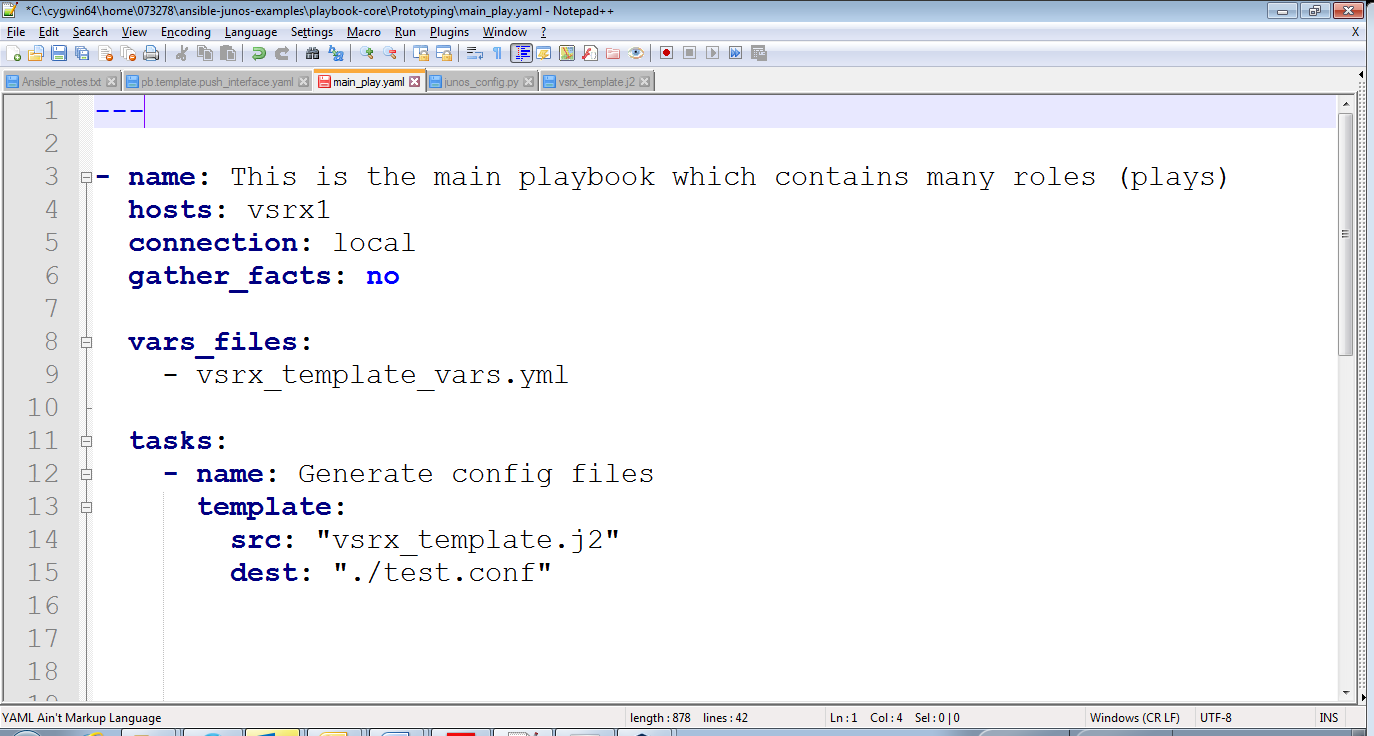
**Line 5** shows the subnet mask variable being assigned as 24

**Line 7,8,9** indicated as another item in the interfaces list with the ‘-‘ and the name of the interface that needs to be configured. These few lines follow the same structure as 3,4,5

Note: as you can see we’ve listed the interface we want to change which is ‘ge-0/0/1’ in this case and we’ve given the interface and ip address of ‘10.1.1.1’ and a subnet mask of ‘24’. You can easily add more interface variables by adding to the list with a “-“ and the individual configuration you want on the device

**Playbooks**

Finally onto playbooks, they’re relatively easy to understand you just need to practice! (.yaml) file



**Line 1**, the ‘---‘ indicates the beginning of the playbook

**Line 3**, the ‘name:’ key is used to describe your play, you should add a brief descriptions so users are aware what the playbook is going to do

**Line 4**, the ‘host:’ key is to tell Ansible which hosts you want to target from your inventory file (this file is just a text file with all your host IP’s or domain names)

**Line 5**, the ‘connection:’ key is to let Ansible know what modules to use,“local” indicates you want to use modules on your local machine. (Don’t worry about this and leave it as local)

**Line 6**, “gather\_facts:” is to collect detailed and a long list of information from the remote host, its unnecessary, so this is set to ‘no’

**Line 8 and 9** ‘vars\_files’ is a key used define the variable file(s) you want to use. Notice the ‘-‘, indicating that variable files can be listed

**Line 11** ‘tasks’ this is the body of your playbook and instructs what you want completed

**Line 12** similar to line 3 where the ‘-‘ indicates the list of task you want completed, a short description of what the task does should be added to the ‘name:’ key

**Line 13** the ‘template’ key is to let Ansible know where to find the template and where to store the conf file to.

**Line 14** The ‘src’ key is there to tell Ansible where to look for the template file, src is short for source. **Line 15** The ‘dest’ key is to tell Ansible where to store the combined template and variable file and what to call that file.