

Lidando com os playbooks

Reuna mais módulos e decodifique textos


9 etapas

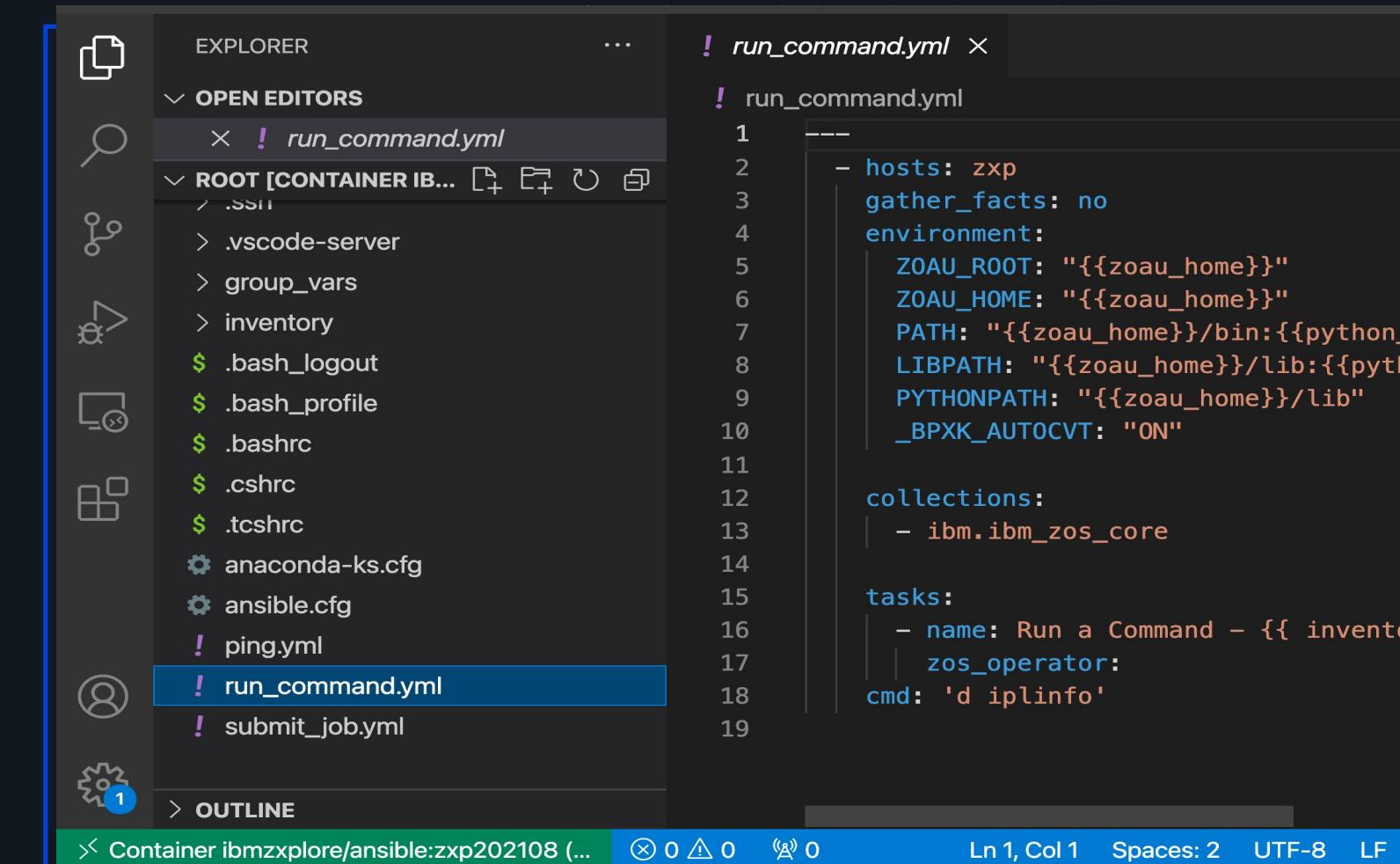
2 horas

O DESAFIO

Com seu ambiente configurado, é possível começar a trabalhar com esses playbooks do Ansible e realmente fazer as coisas acontecerem. Neste desafio, você aprenderá sobre as codificações de caracteres EBCDIC e ASCII, bem como os comandos usados para a conversão entre eles. Em seguida, juntará alguns comandos em um playbook um pouco mais complexo para finalizar este capítulo do que esperamos ter sido uma série emocionante de desafios.

ANTES DE COMEÇAR

É uma boa não pular direto para o final porque há MUITAS informações sobre mainframe necessárias para fazer basicamente qualquer coisa aqui. Se você chegou até aqui fazendo todos os desafios anteriores, você está no lugar certo. Bem-vindo.

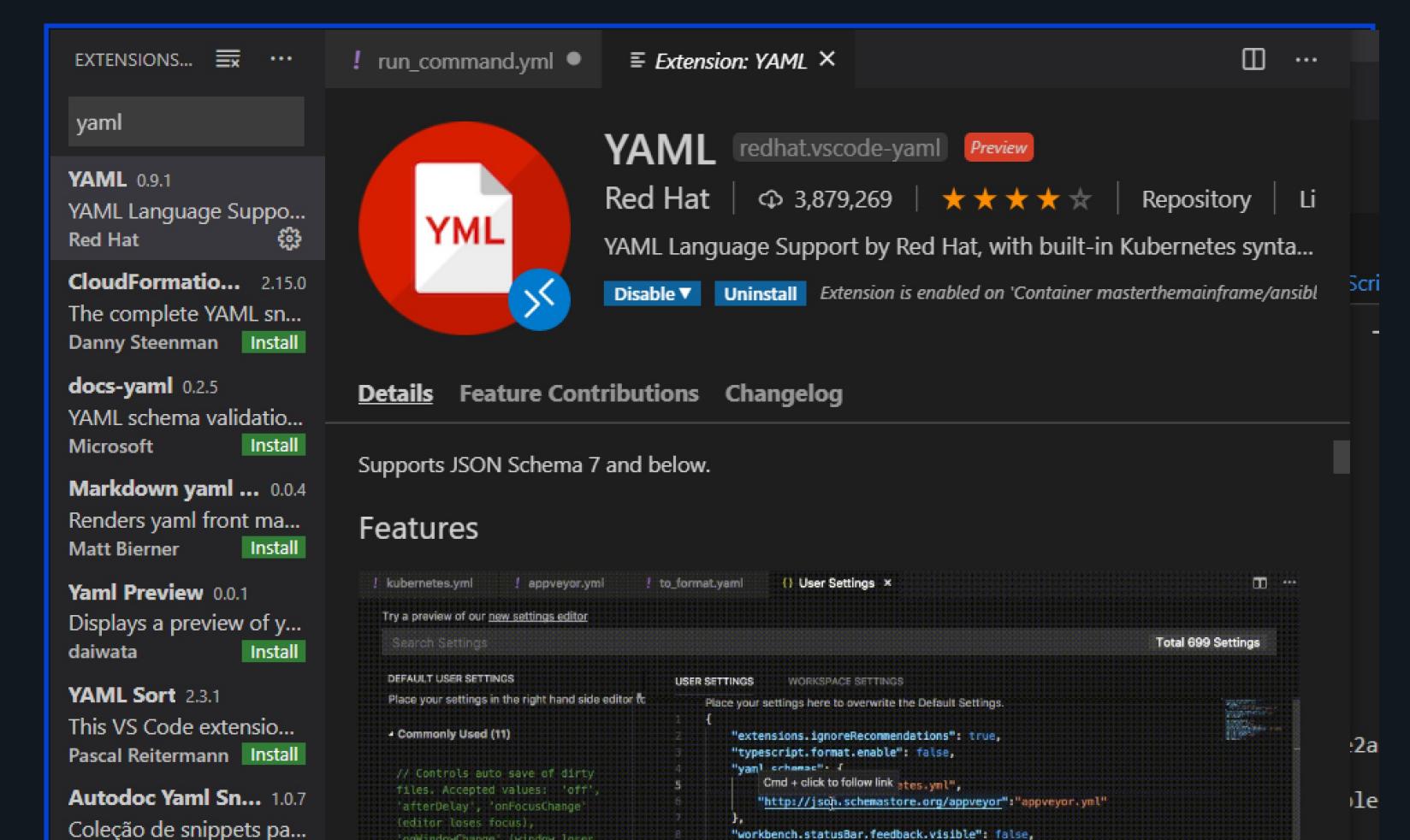
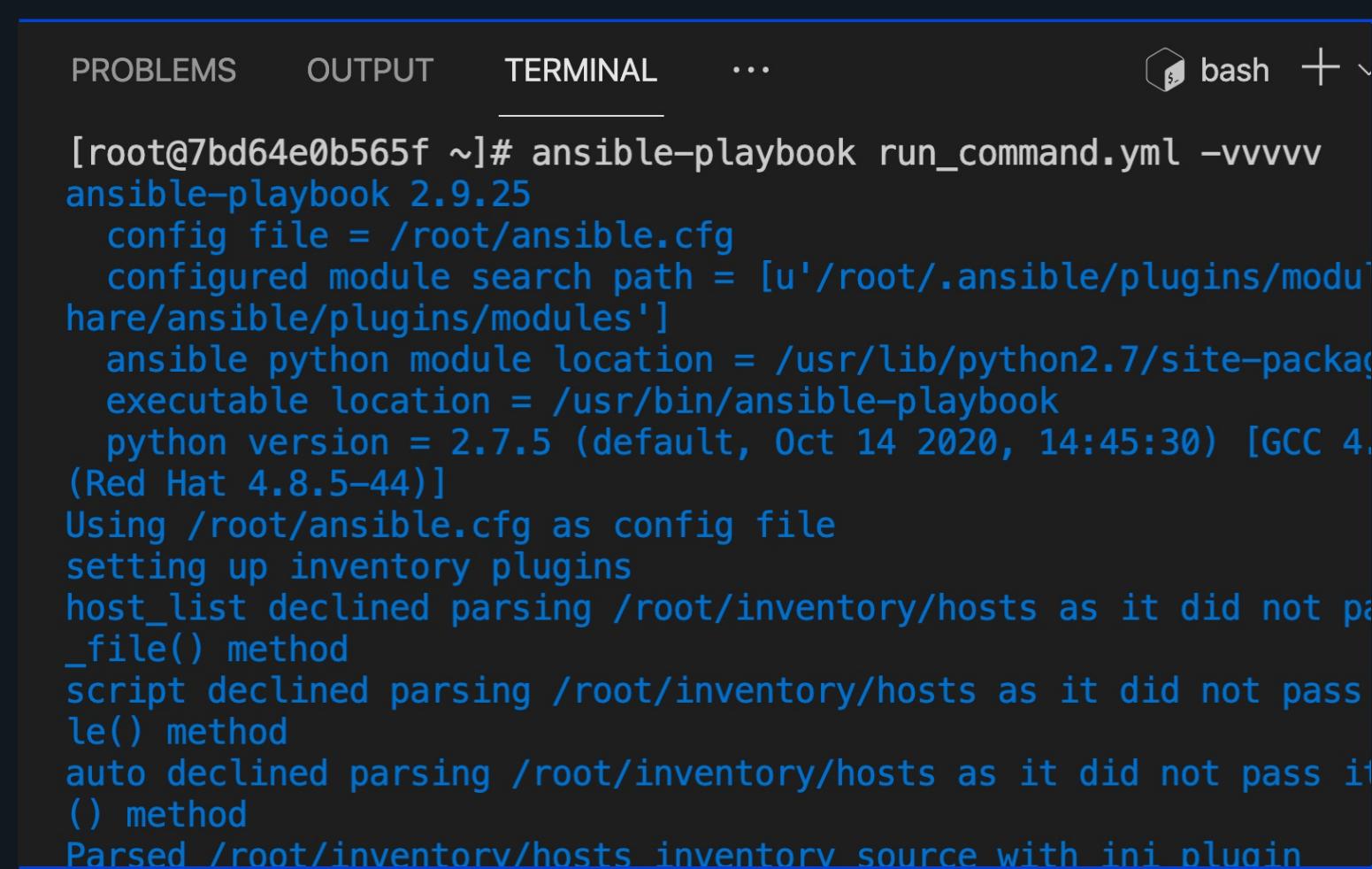


```
EXPLORER
OPEN EDITORS
run_command.yml
ROOT [CONTAINER IB...]
.ssh
.vscode-server
.group_vars
.inventory
.bash_logout
.bash_profile
.bashrc
.cshrc
.tcshrc
.anaconda-ks.cfg
.ansible.cfg
ping.yml
run_command.yml
submit_job.yml

run_command.yml
hosts: zxp
gather_facts: no
environment:
  ZOAU_ROOT: "{{zoau_home}}"
  ZOAU_HOME: "{{zoau_home}}"
  PATH: "{{zoau_home}}/bin:{{python_LIBPATH}}:{{zoau_home}}/lib:{{pyt PYTHONPATH: "{{zoau_home}}/lib"
  _BPXK_AUTOCVT: "ON"

collections:
  - ibm.ibm_zos_core

tasks:
  - name: Run a Command - {{ inventory.zos_operator:
    cmd: 'd iplinfo'
```

```
[root@7bd64e0b565f ~]# ansible-playbook run_command.yml -vvvvv
ansible-playbook 2.9.25
  config file = /root/ansible.cfg
  configured module search path = [u'/root/.ansible/plugins/module_
hare/ansible/plugins/modules']
  ansible python module location = /usr/lib/python2.7/site-pacakage
  executable location = /usr/bin/ansible-playbook
  python version = 2.7.5 (default, Oct 14 2020, 14:45:30) [GCC 4.8.5 (Red Hat 4.8.5-44)]
Using /root/ansible.cfg as config file
setting up inventory plugins
host_list declined parsing /root/inventory/hosts as it did not pass
_file() method
script declined parsing /root/inventory/hosts as it did not pass
le() method
auto declined parsing /root/inventory/hosts as it did not pass
() method
Parsed /root/inventory/hosts inventory source with ini plugin
```

1. REVISITE SEU TRABALHO

Além de o Ansible saber que deve procurá-lo na coleção zos_core, também é necessário especificar onde está o ZOAU. Deixar isso de fora fará com que o comando falhe, mas se você chegou até aqui, provavelmente já sabe disso.

2. LIGANDO O VS CODE

É possível que você tenha tido problemas para formatar seu arquivo YAML corretamente e aprendido que a indentação é importante. Há uma extensão YAML da Red Hat que pode detectar erros antes de enviar playbooks, dar maior visibilidade da estrutura do documento e fornecer preenchimento automático, quando suportado. Ela não é necessária, mas pode ser útil.

Na verdade, há extensões para o Ansible e até mesmo para o Ansible especificamente no Z que podem ser interessantes de conferir à medida que você avança. Elas podem reduzir drasticamente o tempo de desenvolvimento ao longo do percurso (desde que não entrem em conflito umas com as outras).

3. FALE COMIGO

Você já quis obter *um pouco mais* de informações sobre o que acontece durante a execução de um playbook? Claro que é bom quando as coisas funcionam sem problemas para você, mas às vezes você quer mais algumas pistas para conseguir resolver um problema. Coloque um -v após seu comando para torná-lo um pouco mais detalhado. Coloque mais alguns v para torná-lo um pouco *mais* detalhado. Use -vvvv para obter o máximo de informações.

Examples

- **name:** Job output with ddname
zos_job_output:
 job_id: "STC02560"
 ddname: "JESMSGLG"
- **name:** JES Job output without ddname
zos_job_output:
 job_id: "STC02560"
- **name:** JES Job output with all ddnames
zos_job_output:
 job_id: "STC*"
 job_name: "*"
 owner: "IBMUSER"
 ddname: "?"

4. USE ESTES EXEMPLOS

Já dissemos que exemplos são uma boa documentação, lembra? Isso definitivamente se aplica aqui. Na coleção principal do z/OS há exemplos que você provavelmente usará. Na maioria dos casos, é possível apenas copiá-los/colá-los da página em seu código e depois ajustá-los para atender às suas necessidades específicas.

É possível encontrar os exemplos em:
[https://ibm.github.io/z_ansible_collections_doc\(ibm_zos_core/docs/ansible_content.html](https://ibm.github.io/z_ansible_collections_doc(ibm_zos_core/docs/ansible_content.html)

CONTE-ME MAIS SOBRE A CODIFICAÇÃO DE CARACTERES

Que bom que você perguntou. Imagine um único bit de memória, que pode ser 1 ou 0, ligado ou desligado. Com ele, é possível representar dois números, Zero ou Um. Coloque outro bit ao lado dele e agora você dobrará a quantidade de números que podem ser representados para 4. Inclua outro e você terá 8, depois 16 e assim por diante. Você se lembra da conversa sobre hexadecimais na Parte 1? É por isso que vai de 0 a 15.

Com letras e caracteres, as coisas ficam muito mais confusas. Queremos permitir um grande conjunto de caracteres que seja eficiente. Considere que certos caracteres só aparecem em determinados idiomas falados e escritos e outros raramente são vistos fora de um computador. A codificação de caracteres é simplesmente uma tentativa de representar um conjunto de caracteres eletronicamente. Algumas codificações favorecem a eficiência, outras a flexibilidade. Não tome como certa a codificação do texto que você está usando.

start

The final state desired for specified data set.

If `state=absent` and the data set does not exist on the managed node, no action taken, module completes successfully with `changed=False`.

If `state=absent` and the data set does exist on the managed node, remove the data set, module completes successfully with `changed=True`.

If `state=absent` and `volumes` is provided, and the data set is not found in the catalog, the module attempts to perform catalog using supplied `name` and `volumes`. If the attempt to catalog the data set catalog is successful, then the data set is removed. Module completes successfully with `changed=True`.

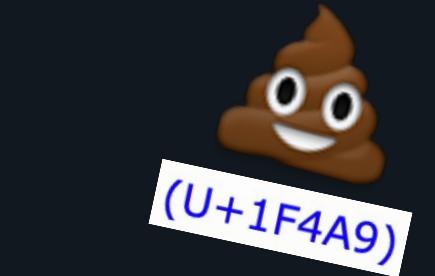
If `state=absent` and `volumes` is provided, and the data set is not found in the catalog, the module attempts to perform catalog using supplied `name` and `volumes`. If the attempt to catalog the data set catalog fails, then no action is taken. Module completes successfully with `changed=False`.

If `state=present` and the data set does not exist on the managed node, create and catalog the data set. module completes successfully with `changed=True`.

5. SUGESTÕES DE SINTAXE

A filosofia de design para tarefas do Ansible é descrever o que precisa ser feito da forma mais sucinta possível, sem confusão. Analisando o módulo `zos_data_set`, é possível ver que a principal diferença entre uma criação e uma exclusão é o parâmetro "estado", que é definido como *Presente* para criação e *Ausente* para exclusão. Você basicamente escreve o que descreveria como um resultado ideal dessa etapa. Você também encontrará frases e abordagens semelhantes em outros módulos que não são do z/OS.

EBCDIC Code		8-Bit ASCII Code		7-Bit ASCII Code	
Graphic	Hex	Graphic	Hex	Graphic	Hex
\$	4A	[5B	[5B
!	5A]	5D]	5D
[AD	(n/a)	D5	SUB	1A
]	BD	(n/a)	E5	SUB	1A
			21	!	21



6. A CODIFICAÇÃO É IMPORTANTE

Há muito tempo, a IBM criou uma maneira de representar caracteres na memória do computador chamada EBCDIC (IBM-1047). Outro padrão surgiu mais tarde chamado ASCII (também conhecido como ISO8859-1). Ambos são igualmente capazes, mas se você está vendendo textos em um mainframe, há uma boa chance de ser EBCDIC e se você está vendendo textos da Internet ou em um aplicativo, provavelmente é ASCII ou UNICODE.

O mundo da codificação de caracteres é uma loucura, e há muitos vídeos e artigos escritos sobre o assunto, se você estiver procurando por um material de leitura mais fácil. Lembre-se de que os conjuntos de dados no Z são EBCDIC por padrão e que é preciso converter ou traduzir ASCII para a leitura.

å??À€! ?Â €ß?Í ÍÁ€ËÍÄÄÁËËÍ‰` €ÍËÁÀ€/>€ >ËÑÂ%Á€ø%/` Â?? , €È?€Ä?>ÎÁÈÈeeeeeeeeeeeeeeee
ëäññ Á>Ã?ÀÁÀ€ÈÁÌÈ€È?€áâäàñä €/>À€` ?Í€ÀÑÀ€ÑÈ€Ñ>€/€Ï` €ÈÇ/È€Ä/>€ÂÁ€ÉÍÑÄ,%` €€€€
ÄÇ/>ÅÁÀ€È?€Ä?>ÎÁÈÈ€/>` €?ÈÇÁÊ€À/È/€ËÁÈÈ€` ?Í€Ä?_Á€/ÄÊ?ËË €eeeeeeeeeeeeeeeeeeeeeeee
ee
íÁ ÈÁ€/%_?ËÈ€È?€ÈÇÁ€Á>À€?Ã€ +ëâ €ÂÍÈ€À?> È€%ÁÈ€ÍÈ€È?ø€` ?Í€ÃÈ?_€ÁÌø%?ËÑ>Å €€€€
ß?Í€øÈ?Â/Â‰`€>?ÈÑÄÁÀ€èë |€Ä?__/>À€_?ÀÍ%ÁË €/>À€ÑÄ€` ?Í€Ï/>È€È?€ÈÍ>€í+ñì€eeeeeee
Ä?__/>ÀË `?Í Ä/> À? ÈÇ/È ÍËÑ>Å ÈÇÁ ÈÁÄÍ‰/È ä?__/>À _?ÀÍ%Á ÈÑ>ÄÁ
/_Ñ>ÃÈ/_Á€ÑË €Ñ>€ÈÇÑË€ÈÁÅ/ÈÀ €;ÍËÈ€/>?ÈÇÁÈ€ø%/` ÈÄ?È_€ÈÇ/È€Ä/>€ÂÁ€Á/ËÑ‰` €€€€€€
/ÍÈ?/_ÈÁÀ€ÈÇÊ?ÍÁç€_?ÀÁÈ>€È?‰€ È€Á/À€Íø€?>€ÈÇ/È€ÇÁÈÁ €ÑÈ€ _ÑÄÇÈ €Ä?_Á€Ñ>€€€€
Ç/>À`€Ä?È€ÈÇÁ‰/` ÈÈ€ÈÈÁø €ee
ÇÈÈøÈ À?ÄË />ÈÑÂ%Á Ä?__/>ÈÑÂ%Á %/ÈÁÈÈ _?ÀÍ%ÁË Ä?__/>À^_?ÀÍ%Á ÇÈ_%
ee
è?€Ñ>ÑËÇÈÇÑË€Äç/%‰Á>ÅÁ €` ?Í€>ÀÁÀ€È?€ ` ?Í€ÁÌÁËÈÀ€ÑÈ €ÄÈÁ/ÈÀÀ€/>?ÈÇÁÈ€€€€
ø%/` Â?? , €èÇÑË€?>À€ÍÑ‰% €ee
è/, Á ÈÇÑË ÈÁÌÈ
(?ÍÁ ÑÈ ?ÍÁÈ È? `?ÍÈ Ç?_Á ÀÑÈÄÈ?È` : :ììììì />ÈÀ ?ÍÈ
ñ>ÈÁÈÈ / _ÁËÈ/ÅÁ ?È / ÈÍ?ÈÁ ?È ;ÍËÈ È?_ÀÈÇÑ>Å `?Í ï/>È È? È/
Â?ÈÈ?_€?Ã€ÈÇÁ€Ñ‰Á €. ÅÁø€ÑÈ€Ä‰Á/>€ ÍÁ€À?ÈÁ/À€ÈÇÁËÁÈÇÑ>ÅË €/>À€ÂÁ€ÉÍÈÁ€€€€€€
È?€Ñ>Ä‰ÍÀÁ€` ?ÍÈ€ÍÈÁÈÑÀ€Ñ>€ÈC/È€ ÁËÈ/ÅÁ€Ë? ÁÏCÁÈÁ €eeeeeeeeeeeeeeeeeeeeeeeeeeee

7. CONVERTA E TRANSFORME

Confira ZXP.PUBLIC.INPUT (COM ANSIBLE). Não é o Dothraki. Isso é o que acontece quando você tenta abrir textos codificados em ASCII em um ambiente EBCDIC.

Felizmente, há um módulo inteiro para lidar com a transformação de codificação, `zos_encode`.

Monte um novo playbook do Ansible para convertê-lo de ASCII para EBCDIC e coloque a saída em algum lugar em seu conjunto de dados SOURCE. Em seguida, abra-o e vá para a Etapa 8.

Good job! You've successfully used an Ansible playbook to convert ASCII-encoded text to EBCDIC, and you did it in a way that can be quickly changed to convert any other data sets you come across.

which starts in the end of 1902. You don't get as soon as the books come up, you already know the stories yourself, and if you want to run with it, you can do it or leave it. They are "the great books" in our library, so, in very English and popular characters there are the classics, including Shakespear and so on. The "new" is always good to ready for the "old" story.

There's a new, though it's not yet published, called "The Great War," which is the history of the war, and it's a good book, and it's a good number of pages. That's the old.

I have this "book"

It has 12 parts, and each book contains 12 chapters, and

is about a message, or a desire, or just words that you want to say, not the names of the cities, towns, or cities and the names of the villages and the names of the mountains and the names of the rivers.



8. SIGA ESTAS ETAPAS

As instruções para finalizar este desafio estão nesse texto e podem ser lidas depois que você traduzi-lo. Leia-as com atenção e comece a criar outro playbook do Ansible. Este será um pouco mais complicado e usará um pouco de lógica, mas sabemos que você está pronto para o desafio agora.

Para concluir, emita CHKAANS2 em ZXP.PUBLIC.JCL e faça acontecer.

9. NÃO PARE AQUI

Você deve estar pensando rápido agora, e isso é uma coisa boa. Nós realmente não precisamos dizer o que você deve fazer agora, pois você provavelmente tem uma lista de coisas que *quer* testar. É possível automatizar um ASCII do REXX. Também pode ser possível encontrar uma maneira melhor de executar um dos verificadores de algoritmo Luhn que envolva menos etapas manuais. Agora que terminou os principais desafios da Parte 3, é hora de nos mostrar o que você realmente pode fazer.

BOM TRABALHO! VAMOS RECAPITULAR

Como é possível ver, integrar o z/OS ao Ansible não é tanto sobre as coisas novas que podem ser feitas, mas sobre quantas tarefas que normalmente requerem muitas etapas manuais ou profundo conhecimento institucional podem ser simplificadas ou automatizadas. Se você tivesse começado daqui, provavelmente ficaria absolutamente sem saída depois de ver sua primeira mensagem de erro e não entenderia o que está acontecendo nos bastidores. Agora que você conhece os dois lados, realmente não há limite para o que você pode fazer.

EM SEQUÊNCIA...

Se você guardou esses desafios do Ansible para o final, acabou com todos os desafios detalhados.

Parabéns! Tudo o que resta (além de uma dança rápida da vitória) é ler sobre o Concurso de Estudantes atual e começar a escrever seu discurso de aceitação do Primeiro Prêmio.

(mantenha-se sempre otimista)