# Ansible IIS代码更新

# (配置说明)

1. 系统环境

|  |  |  |  |
| --- | --- | --- | --- |
| 系统 | 版本 | 角色 | 依赖 |
| Windows | server 2008 r2+ | 被控制端 | IIS 7.0 |
| WinRM |
| Powershell 3.0+ |
| cwRsync client |
| Linux | CentOS 6.0+ | 控制端 | Ansible 2.0+ |
| Ubuntu 12.04+ | pywinrm |
| Fedora | Rsync daemon |
| Opensuse | Subversion |

1. role结构

|  |  |
| --- | --- |
| /data/ansible | 根路径 |
| ├── playbook | Playbook目录 |
| │ ├── update.yml | 主playbook文件 |
| │   └── var\_files |  |
| │ └── token.yml | 字典文件 |
| └── win/iis\_update | Role名称 |
| ├── tasks |  |
| │   ├── main.yml | 主tasks文件 |
| │   ├── facts.yml | 信息处理 |
| │   ├── local\_update.yml | 本地执行更新 |
| │   └── rsync.yml | 推送rsync脚本和执行 |
| ├── templates |  |
| │   ├── password.txt | Password模板 |
| │ ├── rsync.bat | Rsync脚本 |
| └── defaults |  |
| └── main.yml | 默认参数 |

1. 配置文件说明
2. update.yml [执行文件]

|  |
| --- |
| ---  - hosts: "{{ hosts }}"  gather\_facts: false  roles:  - { role: iis\_update, tags: iis } |

1. taks/main.yml [主任务文件]

|  |
| --- |
| ---  # 信息处理  - include: facts.yml  # 执行更新前停止应用池  - name: Windows | Stop {{ site\_path }} application pool  win\_iis\_webapppool: name={{ pool }} attributes='' state=stopped  # 本地代码拉取  - include: local\_update.yml  # 推送和执行脚本  - include: rsync.yml  # 启动应用池  - name: Windows | Start {{ site\_path }} application pool  win\_iis\_webapppool: name={{ pool }} attributes='' state=started  # 代码打包  - include: packaging.yml |

1. tasks/facts.yml [用于获取配置信息和参数处理]

|  |
| --- |
| ---  - block:  # 获取站点信息  - name: Windows | Get {{ site }} infomation  win\_iis\_website: name={{ site }}  register: site\_info  # 初始化变量(参数处理)  - set\_fact:  physicalpath: "{{ site\_info.site.PhysicalPath }}"  pool: "{{ site\_info.site.ApplicationPool }}"  site\_path: "{{ site }}"    when: (app is undefined) # 只有当{{ app }} 未定义才允许该block  - block:  - name: Windows | Get {{ site }}/{{ app }} infomation  win\_iis\_webapplication: name={{ app }} site={{ site }}  register: site\_info  - set\_fact:  # 注册 {{ physicalpath }} 变量  physicalpath: "{{ site\_info.application.PhysicalPath }}"  # 注册{{ pool }}变量  pool:"{{ site\_info.application.ApplicationPool }}"  # 注册{{ site\_path }}变量  site\_path: "{{ site }}/{{ app }}"  when: (app is defined) # 只有当{{ app }}定义了才允许该block  - block:  - name: SVN | Get {{ site\_path }} SVN version  local\_action: |  shell svn info {{ svn\_info.url }}/{{ site\_path }} --username {{ svn\_info.user }} --password {{ svn\_info.pass }} --no-auth-cache --trust-server-cert | grep -Po "(?<=Revision:\s)\d+"  register: revision\_latest  # 本地命令，获取版本号  - name: Local | Get svn infomation  local\_action: |  shell if [ ! -d {{ svn\_temp }} ]  then  mkdir -p {{ svn\_temp }}  fi  svn info {{ svn\_temp }}/{{ site\_path }} --username {{ svn\_info.user }} --password {{ svn\_info.pass }} --no-auth-cache --trust-server-cert | awk '/Revision/{print $2}'  register: revision\_old  changed\_when: true  # debug打印版本信息  - set\_fact:  revision: "{{ revision\_latest.stdout | int }}"  when: (revision is undefined)  - debug:  msg: |  "{{ site\_path }}" update to r{{ revision }}  - debug:  msg: |  "{{ site\_path }}" old version is {{ revision\_old.stdout | int }}  when: (revision\_old.stderr == "") and (revision|int != revision\_old.stdout|int)  run\_once: true # 该block只执行一次 |

1. tasks/local\_update.yml [本地svn执行更新]

|  |
| --- |
| ---  - block:  # 本地命令，代码检出，并作版本判断（jinja2语法）  - name: Local | 项目 "{{ site\_path }}" 检出  local\_action: |  shell \  {% if revision\_old.stdout == "" %}  svn co {{ svn\_info.url }}/{{ site\_path }} {{ svn\_temp }}/{{ site\_path }} --username {{ svn\_info.user }} --password {{ svn\_info.pass }} --force --no-auth-cache --non-interactive --trust-server-cert  {% else %}  svn up -r {{ revision }} {{ svn\_temp }}/{{ site\_path }} --username {{ svn\_info.user }} --password {{ svn\_info.pass }} --force --no-auth-cache --non-interactive --trust-server-cert  {% endif %}  > {{ rsync.log\_path }} # 每次执行提前清除rsync log  run\_once: true # 该block只执行一次 |

1. tasks/rsync.yml [推送脚本和执行]

|  |
| --- |
| ---  # 创建临时目录  - name: Windows | Create script temp  win\_file: path='d:/script\_temp' state=directory  # 推送脚本和密码文件  - name: Windows | Push rsync script  win\_template: src={{ item }} dest=d:/script\_temp/  with\_items:  - rsync.bat  - password.txt  # 脚本执行  - name: Windows | Run rsync script  raw: d:/script\_temp/rsync.bat  # 移除临时目录  - name: Windows | Remove script temp  win\_file: path='d:/script\_temp' state=absent |

1. tasks/packaging.yml[用于更新的代码打包]

|  |
| --- |
| ---  - block:  - local\_action: setup  - set\_fact:  dtime: "{{ ansible\_date\_time.date }}"  ftime: "{{ ansible\_date\_time.iso8601\_basic\_short }}"  - set\_fact:  package\_name: "{{ site\_path | replace('/','.') }}\_r{{ revision }}\_{{ ftime }}.zip"  # 根据rsync log的内容打包更新文件至指定位置  - name: Local | "{{ site\_path }}" 代码打包  local\_action: |  shell \  if [ -n "`grep send {{ rsync.log\_path }}`" ];then  [ -d {{ package\_path}}/{{ site }}/{{ dtime }} ] || \  mkdir -p {{ package\_path}}/{{ site }}/{{ dtime }} && \  cd {{ svn\_temp }} && \  /usr/bin/zip {{ package\_name }} `awk '/send/{print $10}' {{ rsync.log\_path }}` && \  mv {{ package\_name }} {{ package\_path }}/{{ site }}/{{ dtime }}/  else  echo 1  fi  register: package\_state  - debug: msg="{{ site\_path }} package is {{ package\_path }}/{{ site }}/{{ dtime }}/{{ package\_name }}"  when: (package\_state.stdout|int != 1)  run\_once: true |

1. templates/rsync.bat [rsync脚本模板]

|  |
| --- |
| @echo off  set RSYNC\_HOME=C:\Program Files (x86)\cwRsync\bin  cd %RSYNC\_HOME%  # 执行同步  rsync.exe -arPO --delete --force --bwlimit=40000 --exclude "\*.log" --exclude ".svn" --password-file=/cygdrive/D/script\_temp/password.txt --port 873 {{ rsync.user }}@{{ rsync.ip }}::update\_temp/{{ site\_path }}/ /cygdrive/d/iisroot/{{ site\_path }}  # 使用icacls修改目录权限  icacls {{ physicalpath }} /t /c /inheritance:e /grant:r SYSTEM:(oi)(ci)f administrators:(oi)(ci)f IIS\_IUSRS:(oi)(ci)f Users:(oi)(ci)(RD,RC,W) |

1. defaults/main.yml [存放一些信息，如：svn]

|  |
| --- |
| ---  # svn infomation  svn\_info:  url: https://svn.abc.com/svn  user: abc  pass: xxxxx  # local  svn\_temp: /data/svn\_temp  rsync\_temp: /data/rsync\_temp  package\_path: /data/package  # rsync infomation  rsync:  module: update\_temp  ip: 192.168.11.60  user: awx  pass: JNDhdMo9AlFcnjPb1484qe  log\_path: /data/logs/rsync/rsync\_server.log |

1. /etc/rsyncd.conf [rsyncd配置文件]

|  |
| --- |
| # 同步的日志，用tower的话要配置属主为awx,该日志还用于代码打包  log file= /data/logs/rsync/rsync\_server.log  syslog facility = daemon  transfer logging = yes  port=873  [update\_temp]  comment = public archive  # 目录权限应和认证用户一致  path = /data/svn\_temp  use chroot = no  lock file = /var/lock/rsyncd  read only = no  list = yes  uid = awx  gid = awx  hosts allow = 192.168.0.0/8  #incoming chmod = Du=rwx,g=rwx,o=rx,Fu=rw,g=rw,o=r  #outgoing chmod = Du=rwx,g=rwx,o=rx,Fu=rw,g=rw,o=r  # awx用户是为了ansible tower执行，不用tower其他用户随意  auth users = awx  # 存放账号密码文件，权限400  secrets file = /etc/rsyncd\_server.secrets  strict modes = yes  ignore errors = yes  ignore nonreadable = yes  transfer logging = yes  log format = %t: host %h (%a) %o %f (%l bytes). Total %b bytes.  timeout = 600  refuse options = checksum dry-run  dont compress = \*.gz \*.tgz \*.zip \*.z \*.rpm \*.deb \*.iso \*.bz2 \*.tbz |

1. Templates/password.txt[密码文件模板]

|  |
| --- |
| {{ rsync\_info.password }} |