OPITZ CONSULTING

ADHOC COMMANDS

AD-HOC COMMANDS - BENEFITS

- One-time tasks for execution on the inventory hosts

```
ansible webservers -a "/sbin/reboot" (reboot of all web servers)
```

- For simple use cases / one-off actions
  - Reboot
  - Copy individual files to various hosts
- All modules can be executed ad-hoc
- Disadvantages
  - Execution not traceable, config of the module not persistent
  - No complex logic

AD-HOC COMMANDS - SOME USE CASES

- Local testing of modules
- Query the system status of the hosts
  - Environment variables
  - Disk Usage
  - Logfiles
  - Versions of software packages
  - ...
- Restart services / hosts
- Better not: Change the system configuration
  - Use playbooks for this

COMMAND LINE CALL ANSIBLE

```
ansible webservers (host or group name)
        -i inventory (path to the inventory)
        -m yum (module name, default: command)
        -a "name=ntp state=installed" (module arguments, key=value)
        -b (become, sudo user)
```

- Examples

```
ansible all -m ping (Check SSH connection to all hosts)

ansible node1 -m setup (Determine all facts of host node1)

ansible all -a "df -h" (Check disk usage on all hosts)

ansible all -b -m user -a "name=mike" (Create user mike on all hosts)
```

SOME EXAMPLE MODULES

Module name Parameters

command: hostname # any command line commands

shell: somescript.sh >> somelog.txt # with shell environment

user: name=james groups=developers # Create user

package: name=nginx state=present # Install package

service: name=nginx state=started # Start service

copy: src=/httpd.conf dest=/etc/httpd/httpd.conf # Copy file from control node

unarchive: src=assets.tgz dest=/var/www/ # Copy file from control node and unzip

OBTAIN INFORMATION ABOUT MODULES - RECAP

- https://docs.ansible.com → Module Index

- The ansible-doc command

`ansible-doc yum` (Everything about the yum module)

`ansible-doc -s yum` (snippet for the playbook)

`ansible-doc -l` (list of all modules)