Task Automation Tools Cheatsheet

A comprehensive cheat sheet covering essential task automation tools, their features, syntax, and use cases, enabling efficient workflow automation.

Core Concepts & Tools Overview

Introduction to Task Automation

Task automation involves using software or scripts to perform repetitive tasks automatically, reducing manual effort and increasing efficiency.

Key benefits include reduced errors, faster processing times, and freeing up human resources for more complex tasks.

Common use cases include data processing, system administration, software deployment, and report generation.

Popular Task Automation Tools

Ansible	An open-source automation tool for configuration management, application deployment, and task automation. Uses YAML for playbooks.
Chef	An automation platform that transforms infrastructure into code. Uses Ruby DSL for defining infrastructure.
Puppet	An open-source configuration management tool that automates the provisioning and management of servers. Uses a declarative language.
Cron	A time-based job scheduler in Unix-like operating systems. Used to schedule tasks to run automatically at specific times.
Jenkins	An open-source automation server used for continuous integration and continuous delivery (CI/CD).
Task Scheduler (Windows)	A component of Microsoft Windows that allows users to schedule the launch of programs or scripts at pre-defined times or after specific intervals.

Ansible Essentials

Ansible Playbooks

Playbooks are YAML files that define the tasks to be executed on managed nodes.

They describe the desired state of a system and ensure that the system reaches that state.

A basic playbook structure includes hosts, tasks, and modules.

Example:

```yaml
---
- hosts: webservers
  tasks:
    - name: Install Apache
      apt:
        name: apache2
        state: present
```

Common Ansible Modules

`apt`	Manages apt packages on Debian/Ubuntu systems.
`yum`	Manages yum packages on Red Hat/CentOS systems.
`file`	Manages files and directories.
`template`	Deploys files from Jinja2 templates.
`service`	Manages services.
`copy`	Copies files to remote locations.

Ansible Commands

`ansible-playbook playbook.yml` - Executes an Ansible playbook.

`ansible-inventory --list` - Lists the inventory.

`ansible all -m ping` - Pings all hosts in the inventory.

`ansible-vault encrypt secrets.yml` - Encrypts a YAML file.

Cron Scheduling

Cron Syntax

Cron jobs are defined in a crontab file, with each line representing a job.

The syntax is: `minute hour day month weekday command`

- - - - - command to be executed

| | | | |
| | | | +—— Day of the week (0 - 6) (Sunday=0)
| | | +——— Month (1 - 12)
| | +———— Day of the month (1 - 31)
| +————— Hour (0 - 23)
+—————— Minute (0 - 59)

Cron Examples

`0 0 * * * /path/to/script.sh`	Runs `/path/to/script.sh` every day at midnight.
`0 * * * * /path/to/script.sh`	Runs `/path/to/script.sh` every hour at the beginning of the hour.
`0 9 * * 1-5 /path/to/script.sh`	Runs `/path/to/script.sh` every weekday (Monday-Friday) at 9:00 AM.
`*/5 * * * * /path/to/script.sh`	Runs `/path/to/script.sh` every 5 minutes.
`0 0 1 * * /path/to/script.sh`	Runs `/path/to/script.sh` on the first day of every month at midnight.
`@reboot /path/to/script.sh`	Runs `/path/to/script.sh` after every system reboot.

Cron Management

`crontab -e` - Opens the crontab file in an editor to add, modify, or delete cron jobs.

`crontab -l` - Lists the current cron jobs.

`crontab -r` - Removes the current crontab file (use with caution).

Ensure the script has execute permissions: `chmod +x /path/to/script.sh`

Windows Task Scheduler

Task Scheduler Interface

The Task Scheduler is a Windows component that allows you to automate tasks by scheduling them to run at specific times or when certain events occur.

Key components include Triggers, Actions, and Conditions.

Creating Tasks

Basic Task	Use the Basic Task wizard for simple tasks.
Create Task	Use the Create Task dialog for more advanced options and configurations.
Triggers	Define when the task should start (e.g., on a schedule, at startup, on an event).
Actions	Specify what the task should do (e.g., start a program, send an email).
Conditions	Define conditions that must be met for the task to run (e.g., idle time, power state).
Settings	Configure additional settings such as allowing the task to run on demand, stopping the task if it runs longer than a specified time, etc.

Task Scheduler Commands

`schtasks /create /tn "MyTask" /tr "C:\path\to\script.exe" /sc DAILY /st 09:00` - Creates a scheduled task named ‘MyTask’ to run daily at 9:00 AM.

`schtasks /run /tn "MyTask"` - Runs the scheduled task named ‘MyTask’ immediately.

`schtasks /delete /tn "MyTask" /f` - Deletes the scheduled task named ‘MyTask’ without confirmation.

`schtasks /query /tn "MyTask"` - Displays detailed information about the scheduled task named ‘MyTask’.

`schtasks /change /tn "MyTask" /disable` - Disables the scheduled task named ‘MyTask’.