

# CommandLine

## Tools for DevOps

@Jaykul

Press Space for next page →

April 7-10, 2025



SynEdgy



Jane Street



# Thank you Sponsors

---



PATCH MY PC

# Joel "Jaykul" Bennett

## Principal DevOps Engineer

Solving problems with code

15x Microsoft MVP for PowerShell

-  [github.com/Jaykul](https://github.com/Jaykul) and PoshCode
-  [discord.gg/PowerShell](https://discord.gg/PowerShell)
-  [HuddledMasses.org](https://HuddledMasses.org)
-  [@jaykul.powershell.social](https://@jaykul.powershell.social)
-  [@Jaykul@FOSStodon.org](https://@Jaykul@FOSStodon.org)



# A is for age

<https://github.com/FiloSottile/age>

A simple, modern and secure encryption tool with small keys, no config options, and pipeline composability. (See also `rage` and `typage`).

## Tips

Age also supports using SSH Keys for encryption.

Honorable mention to Ansible and A2

## Example

```
age-keygen -o key.txt
Public key: age1ql3z7hjy54pw3hyww5ayyfg7zqgvc7w3j2elw8zmrj2kg5sfn9aqmcac8p
```

```
tar cvz "~/data" | age -r "age1ql3z7hjy54pw3hyww5ayyfg7zqgvc7w3j2elw8zmrj2kg5sfn9aqmcac8p" > data.tar.gz.age
age --decrypt --identity "key.txt" --output "data.tgz" data.tgz.age
```

# B is for bat

<https://github.com/sharkdp/bat>

A `cat` clone with syntax highlighting and git integration. It shows modifications with respect to the git index, can syntax highlight anything from actionscript to zig (including powershell), can show non-printable characters, can display multiple files at once, read pipeline input, pages by default, etc.

## Example

Next time someone asks you to `irm ... | iex`, check it with `bat` first:

```
irm bun.sh/install.ps1 | bat -l ps1
```

# C is for chezmoi

<https://github.com/twpayne/chezmoi>

Chezmoi manages personal configuration files ("dotfiles" like `~/.gitconfig`) and profile scripts across machines. I even use it to manage my WezTerm and Windows Terminal config, and some VSCode settings.

## Example

To set up a new machine with my dotfiles:

```
winget install twpayne.chezmoi
chezmoi init --apply Jaykul
```

To edit a dotfile you have stored in chezmoi:

```
chezmoi edit ~/.gitconfig --apply
```

# C is for curl

The open source `curl` command-line tool is so universal, it's even included in Windows! It's **not** just for HTTP, it supports over two dozen other protocols including FTP, Gopher, IMAP, POP3, SMTP and MQTT, etc.

## Tips

1. You can put default options in `~/.curlrc` and sync them with chezmoi 😊
2. Try [Curl by Example](#).

## Example

```
# To make curl fail when it gets a 4xx or 5xx code
curl --fail http://httpbingo.org/status/503
# To POST json
curl --json ($pwd|select *Path| ConvertTo-Json) http://httpbingo.org/post
# Get someones public SSH keys so you can encrypt a message to them
curl -s https://github.com/jaykul.keys | age -a -R - .\demos\password.txt
```

# D is for dotnet new

<https://github.com/dotnettemplating>

By far the easiest way to create templates for projects.

Take a simple *working* project, add a `template.json` config file, and share via NuGet.

The key is that instead of injecting tokens into your code, you can specify the GUIDs and names that are already in your project and explicitly keep only some parts of the project in the template. This way your templates are based on actual working code, you can use `dotnet new` regardless of the project language.

## Example

```
dotnet new --install Jaykul.PowerShellModule
dotnet new psmodule -n MyModule
```

## See also

- [My PowerShell Template](#)

# E is for earthly

<https://github.com/earthly/earthly>

Earthfile is like a cross between make files and dockerfiles. It can build anything, and the builds are fast, cached, and consistent, running the same whether on a developer laptop or GitHub CI/CD.

```
VERSION 0.8
FROM mcr.microsoft.com/dotnet/sdk:9.0
WORKDIR /work

deps:
  COPY src/*.csproj src/
  RUN dotnet restore src/*.csproj

build:
  FROM +deps
  COPY src src
  RUN dotnet publish --no-restore src/*.csproj -o publish
  SAVE ARTIFACT publish AS LOCAL publish
```

# F is for fzf

<https://github.com/junegunn/fzf>

Think of fzf as a faster, more powerful, text-only version of `Out-GridView` .

Also a way nicer Ctrl+r (thanks [kelleyma49/PSFzf](#) or [domsleee/Posh-Fzf](#))

\$ [ ]

# G is for Git

<https://git-scm.com/>

This one is a no-brainer. Please folks:

1. Put your code in source control
2. Put your infrastructure in code

Honorable mention to `gh`, the GitHub CLI

Create or clone repositories, manage pull requests, and even close issues from your terminal.

# H is for httpie

<https://github.com/httpie/cli>

A colorful modern command-line HTTP client with a more intuitive interface (docs). Designed for testing & debugging APIs, servers & services.

## Examples

```
# Stream in json...
$pwd | Select *Path | ConvertTo-Json | http POST http://httpbingo.org/post

# Type literal json (or forms with -f)
http http://httpbingo.org/post Path=C:\Users\Jaykul ProviderPath=C:\Users\Jaykul

# Download with --download or via redirection
http pie.dev/image/png > image.png
```

## Honorable mention to sharkdp/hexyl

A simple hex viewer that uses a colored output to distinguish different categories of bytes.

# I is for ImageMagick

<https://imagemagick.org/>

So many possibilities for scripting images, from resizing & converting to complex text annotation. Sixels!

```
magick -background '#212021' -fill lightblue -font Jokerman -pointsize 62  
-size 420x -gravity Center caption:'ImageMagick Rules OK?' ``  
sixel:-
```



## Examples

```
magick ./public/images/RateThis.png RateThis.six && get-content RateThis.six  
# convert to sIXEL and output to pipeline  
./public/images/RateThis.png sIXEL:-
```



```
magick -background '#212021' -fill lightblue -font Jokerman -pointsize 62 -size 600x -gravity Center caption:'I Is for I'
```

# J is for jnv

<https://github.com/ynqa/jnv>

The interactive filter editor for jq, still lets you slice and filter, map and transform structured data. You don't *need* jq in PowerShell, but having it can save you time when people's examples use it, and if nothing else, it's a prettier way to display json.

## Examples

```
kubectl get pod opensearch-0 -ojson | jnv
```

```
curl 'https://api.github.com/repos/powershell/powershell/commits?per_page=5'  
| jq '.[] | {message: .commit.message, name: .commit.author.name}'
```

```
curl 'https://api.github.com/repos/powershell/powershell/commits?per_page=5'  
| jnv  
.[] .commit.author
```

# K is for kubeetl k9s

<https://github.com/derailed/k9s>

A terminal based UI to interact with your Kubernetes clusters.

```
Context:    minikube
Cluster:   minikube
User:      minikube
K9s Version: 0.1.6
K8s Version: v1.13.2
CPU:       10%(-)
MEM:       20%(+)
```

```
<?>    Help      <0> all
<ctrl-d> Delete    <1> kube-system
<d>    Describe   <2> default
<e>    Edit
<l>    Logs
<s>    Shell
<v>    View
```



## Pods(all)[12]

NAMESPACE	NAME	READY	STATUS	RESTARTS	CPU	MEM	IP	NODE	QOS	AGE
default	nginx-6988c9989f-wwz6d	1/1	Running	0			172.17.0.6	192.168.64.83	Guaranteed	87s
kube-system	coredns-86c58d9df4-dkhf2	1/1	Running	0	3m	8Mi	172.17.0.3	192.168.64.83	Burstable	17h
kube-system	coredns-86c58d9df4-jt79s	1/1	Running	0	2m(-)	8Mi	172.17.0.2	192.168.64.83	Burstable	17h
kube-system	etcd-minikube	1/1	Running	0	24m(-)	53Mi	192.168.64.83	192.168.64.83	BestEffort	17h
kube-system	kube-addon-manager-minikube	1/1	Running	0	7m(+)	17Mi(-)	192.168.64.83	192.168.64.83	Burstable	17h
kube-system	kube-apiserver-minikube	1/1	Running	0	47m(-)	383Mi	192.168.64.83	192.168.64.83	Burstable	17h
kube-system	kube-controller-manager-minikube	1/1	Running	0	49m(-)	55Mi(+)	192.168.64.83	192.168.64.83	Burstable	17h
kube-system	kube-proxy-pjh2p	1/1	Running	0	4m(-)	10Mi	192.168.64.83	192.168.64.83	BestEffort	17h
kube-system	kube-scheduler-minikube	1/1	Running	0	15m(-)	12Mi	192.168.64.83	192.168.64.83	Burstable	17h
kube-system	kubernetes-dashboard-ccc79bfc9-qgnck	1/1	Running	0	0m	11Mi	172.17.0.4	192.168.64.83	BestEffort	17h
kube-system	metrics-server-6fc4b7bcff-hp6pm	1/1	Running	0	1m	14Mi	172.17.0.5	192.168.64.83	BestEffort	17h
kube-system	storage-provisioner	1/1	Running	0	0m	13Mi	192.168.64.83	192.168.64.83	BestEffort	17h

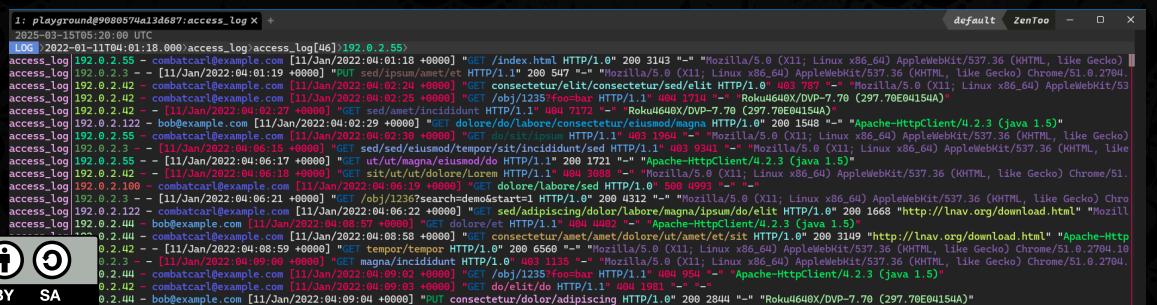
# L is for lnav <https://github.com/tstack/lnav>

Logfile Navigator. A TUI with built-in understanding of syslog, apache, strace, and more, but **very good** with any log files that have time stamps. Supports *navigation by time*, markdown annotations, searching, etc. Try it! There's a 'tutorial1' and a 'playground' you can use to learn what it can do:

```
ssh tutorial1@demo.lnav.org
```

**⚠ Note:** Windows builds currently only available on ci-build output, but just unzip and run.

## Logs



```
1: playground@9880574a134687:access_log x +
2025-03-15T05:20:00 UTC
LOG | 2022-01-11T04:01:18.800 access_log.access_log[46] 192.0.2.55
access_log 192.0.2.55 - combatcar@example.com [11/Jan/2022:04:01:18 +0000] "GET /index.html HTTP/1.0" 200 3143 "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/51.0.2704.106 Safari/537.36"
access_log 192.0.2.3 - - [11/Jan/2022:04:01:19 +0000] "PUT sed/ ipsum/ amet/ et HTTP/1.1" 200 547 "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/51.0.2704.106 Safari/537.36"
access_log 192.0.2.3 - - [11/Jan/2022:04:02:24 +0000] "GET /constuctetor/ do/ /connected/ sed/ HTTP/1.1" 200 485 "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/51.0.2704.106 Safari/537.36"
access_log 192.0.2.42 - - [11/Jan/2022:04:02:25 +0000] "GET /obj/1235/ FooBar/ HTTP/1.1" 404 1710 "-" "Redukt/0.0.27-7.78 (297.70E04154A)"
access_log 192.0.2.42 - - [11/Jan/2022:04:02:27 +0000] "GET /sed/ incidunt/ HTTP/1.1" 404 1710 "-" "Redukt/0.0.27-7.78 (297.70E04154A)"
access_log 192.0.2.122 - bob@example.com [11/Jan/2022:04:02:29 +0000] "GET dolore/ do/ labore/ consectetur/ eiusmod/ magna/ HTTP/1.1" 200 1548 "-" "Apache-HttpClient/4.2.3 (java 1.5)"
access_log 192.0.2.55 - combatcar@example.com [11/Jan/2022:04:02:38 +0000] "GET /sit/ ut/ ipsum/ HTTP/1.1" 403 1964 "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/51.0.2704.106 Safari/537.36"
access_log 192.0.2.3 - - [11/Jan/2022:04:06:15 +0000] "GET sed/ sed/ eiusmod/ tempor/ sit/ incidunt/ sed/ HTTP/1.1" 403 9341 "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/51.0.2704.106 Safari/537.36"
access_log 192.0.2.55 - - [11/Jan/2022:04:06:17 +0000] "GET ut/ ut/ magna/ eiusmod/ do/ HTTP/1.1" 200 1721 "-" "Apache-HttpClient/4.2.3 (java 1.5)"
access_log 192.0.2.42 - - [11/Jan/2022:04:06:18 +0000] "GET ut/ ut/ dolore/ /Lorem/ HTTP/1.1" 404 3088 "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/51.0.2704.106 Safari/537.36"
access_log 192.0.2.109 - combatcar@example.com [11/Jan/2022:04:06:20 +0000] "GET dolore/ labore/ sed/ HTTP/1.0" 509 993 "-" "Apache-HttpClient/4.2.3 (java 1.5)"
access_log 192.0.2.122 - combatcar@example.com [11/Jan/2022:04:06:21 +0000] "GET /obj/1235/ search/ demonstrate/ HTTP/1.0" 200 4312 "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/51.0.2704.106 Safari/537.36"
access_log 192.0.2.42 - bob@example.com [11/Jan/2022:04:06:22 +0000] "GET /do/ /adipiscing/ /eiusmod/ /tempore/ /incidunt/ /dolore/ /ut/ /amet/ /et/ /sit/ HTTP/1.0" 200 1668 "http://lnav.org/download.html" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/51.0.2704.106 Safari/537.36"
access_log 192.0.2.44 - combatcar@example.com [11/Jan/2022:04:06:57 +0000] "GET magna/ incidunt/ /tempore/ /incidunt/ /dolore/ /ut/ /amet/ /et/ /sit/ HTTP/1.0" 404 1002 "-" "Apache-HttpClient/4.2.3 (java 1.5)"
access_log 192.0.2.44 - combatcar@example.com [11/Jan/2022:04:09:02 +0000] "GET magna/ incidunt/ /tempore/ /incidunt/ /dolore/ /ut/ /amet/ /et/ /sit/ HTTP/1.0" 404 954 "-" "Apache-HttpClient/4.2.3 (java 1.5)"
access_log 192.0.2.44 - bob@example.com [11/Jan/2022:04:09:03 +0000] "GET /do/ /elit/ /dolore/ /HTTP/1.1" 404 1981 "-" "Apache-HttpClient/4.2.3 (java 1.5)"
access_log 192.0.2.42 - - [11/Jan/2022:04:10:56 +0000] "PUT /consectetur/ /dolor/ /adipiscing/ /HTTP/1.0" 200 2844 "-" "Roku460X/DVP-7.78 (297.70E04154A)"
```

## Markdown

### An h1 header

Paragraphs are separated by a blank line.

2nd paragraph. *Italic*, **bold**, and `monospace`. Itemized lists look like:

- this one
- this one



BY SA

access\_log 192.0.2.42 - - [11/Jan/2022:04:10:56 +0000] "PUT /consectetur/ /dolor/ /adipiscing/ /HTTP/1.0" 200 2844 "-" "Roku460X/DVP-7.78 (297.70E04154A)"

# M is for Miller

<https://github.com/johnkerl/miller>

Miller is for stream-processing tabular data. It has a simple syntax for filtering, transforming, and aggregating data. It even handles a few formats that PowerShell does not, like pretty-printed tabular data and delimited key-value pair data, and can read compressed files.

## Examples

```
# Flatten json to tabular data ...
curl 'https://api.github.com/repos/powershell/powershell/commits?per_page=5' -s
| mlr --json --ocsv flatten then cut -f "sha,author.login"
# Convert (and clean up) TSV data
helm list -A | mlr --itsv --opprint clean-whitespace then cut -x -f STATUS,CHART
helm list -A | mlr --itsv --opprint clean-whitespace
```

# N is for nvm

<https://nmap.org/>

Node Version Manager. Using node to support lots of developer teams without NVM is ...  
difficult

```
nvm list  
nvm use newest
```

# O is for Ov

<https://github.com/noborus/ov>

## Let's talk about pagers

- **More** is a pager. It outputs one page at a time so you can read it all.
- **Less** does more. We can go *back*, search, *filter*, use colors and your mouse. `less -R --mouse`
- **Moar** does less. Adds incremental search, syntax highlighting.
- **Ov** supports tabular data, multiple highlights, configurable keys, named modes...

## Examples

```
$Env:LESS = '-R --mouse'  
$Env:PAGER = 'ov'  
# Viewing tab-delimited columnar data that's too wide for the screen  
helm list -A | ov --view-mode posix  
# Highlighting ERROR and WARNING lines, etc.  
multi-color "ERROR,WARN,INFO,DEBUG,not,^.{24}" access.log  
ting columns to find the index of an error  
ov --jump-target 3 --ruler --line-number --pattern "O is for Ov" ./slides.md'
```



# P is for Pandoc

<https://github.com/jgm/pandoc>

If you need to convert text files from one markup format into another, pandoc is your swiss-army knife.

## Examples

```
# Output ANSI escape sequences for a markdown file
pandoc -s README.md -t ansi
# Convert a markdown file to a Jupyter notebook
pandoc -s README.md -o README.ipynb
# Convert a markdown file to Textile (we use Jira)
pandoc -s README.md -o README.textile
```

Did you know you can use pandoc as a lua interpreter? Just run `pandoc --lua` ...`

# Q is for qrc

<https://github.com/fumiyas/qrc>

QRC is a helpful little command-line tool for generating QR codes *inside* your terminal.

## Examples

```
qrc "https://survey.sogolytics.com/survey/form?k=RQsQSQTQTsRUsPsPsP&lang=0"  
qrc "MAILTO:Jaykul@HuddledMasses.org"  
qrc "WIFI:S:guestwifi;T:WPA;P:we are not allowed to share that;;"
```

# R is for Ripgrep

<https://github.com/BurntSushi/ripgrep>

Ripgrep *recursively* searches directories for files and then for text matching a regex pattern. Respects .gitignore, and has switches to search specific file types, and can do globbing.

## Examples

```
rg -tps "Write-Host"  
rg -tyaml "dev" --iglob "**/*prod/*"
```

# S is for SOPS

<https://github.com/getSops/sops>

SOPS is a tool for managing secrets in configuration files, encrypting with Age, PGP, Azure Key Vault, AWS KMS or GCP KMS.

Demonstrating it's full use is practically a session in itself, here's an example of encrypting a Kubernetes secrets file:

```
$key = age-keygen -y $Env:AppData\sops\age\keys.txt
sops -e -i --age $key --encrypted-regex '^(\data|stringData)$' .\secret.yaml
Get-Content '.\secret.yaml'
```

And decrypting it:

```
sops ./secret.yaml
```



# V is for viu

<https://github.com/atanunq/viu>

A fast image viewer for the terminal.

Really I should mention trackd/sixel but it's a native PowerShell module 😊

## Examples

```
curl https://media3.giphy.com/media/v1.Y2lkPTc5MGI3NjExdjdncljazQyYmlyM2gyMwVpeTE3YTB5M2Ixaw5kMWFwOW84djh2dCZlcD12MV9pI  
| viu --once -
```

```
< ----->  
Get-ChildItem .\Icons\weather\ -Exclude *.txt  
| Get-Random -Count 5  
| ConvertTo-Sixel
```

# W is for Wush

<https://github.com/coder/wush>

A wireguard powered file-transfer tool. Also available in your browser at [wush.dev](https://wush.dev)

```
wush serve  
wush cp 1gb.txt
```

Honorable mention to WezTerm (and Windows Terminal)

And Winget, and Wsl...

# X is XQ

<https://github.com/sibprogrammer/xq>

It's a little like jq, but for XML and with XPath or CSS selectors instead

```
Get-Content unformatted.xml | xq  
Get-Content unformatted.xml | xq -x /user/@status  
Get-Content unformatted.xml | xq -q "body > p"
```

# Y is for yq

<https://github.com/mikefarah/yq>

yq is a command-line structured data processor, like `jq` but with support for yaml, json, xml, properties, csv, tsv and toml. It doesn't support quite everything that jq does, but it's faster...

## Examples

```
# Check the image versions in a deployment
yq '.spec.template.spec.containers[].image' ./deployment.yaml
# Update in place, the image for the first container
yq -i '.spec.template.spec.containers[0].image = "ghcr.io/stefanprodan/podinfo:6.8.0"' ./deployment.yaml
```

# Z is for zoxide

<https://github.com/ajeetdsouza/zoxide>

A cross-platform, cross-shell, smarter `cd` command. It tracks your most used directories and lets you jump to them with a few keystrokes.

It uses fzf to fuzzy match folders if you want, and supports integration to PowerShell, and importing your "database" from ZLocation (an *awesome* PowerShell module that basically does the same thing, with better PowerShell support, but obviously without support for running in other shells).

## Examples

```
z some/long/path/to/a/directory
z # will go to ~
z direc # will return to "some/long/path/to/a/directory"
```

# THANK YOU!

Feedback is a  
gift. Please  
review this  
session!

