

# DevOps for research

Jun Kato

AIST, Arch Inc., Universite Paris-Saclay

#non\_research\_tips #情報科学研究補助技法

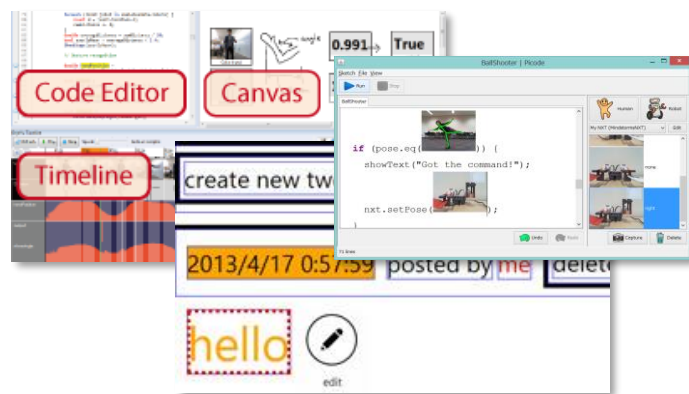
Icons by FontAwesome, licensed under SIL Open Font License 1.1. For more information, visit: <https://fontawesome.com/license/free>

# Links to open-source code in this lecture

- 🔗 **arcatdmz/non-research-tips-gitlab-ci**
  - GitLab CI/CD simple example
  - <https://gitlab.com/arcatdmz/non-research-tips-gitlab-ci>
- 🔗 **arcatdmz/non-research-tips-devcontainer**
  - VS Code DevContainer for consistent development experience
  - Node.js + MySQL + Prisma
  - <https://github.com/arcatdmz/non-research-tips-devcontainer>
- 🔗 **arcatdmz/acm2024-primary-article-template**
  - Editing, linting, and sharing papers
  - Overleaf + GitHub Actions + GitHub Pages + Slack
  - <https://github.com/arcatdmz/acm2024-primary-article-template>
- 🔗 **ChezTachibana/website**
  - Cloudflare Pages + Google Sheets
  - <https://github.com/ChezTachibana/website>
- 🔗 **arcatdmz/arcatdmz.github.io**
  - Single source of truth for multiple views
  - Gulp + Pug + BibTeX etc.
  - <https://github.com/arcatdmz/arcatdmz.github.io>

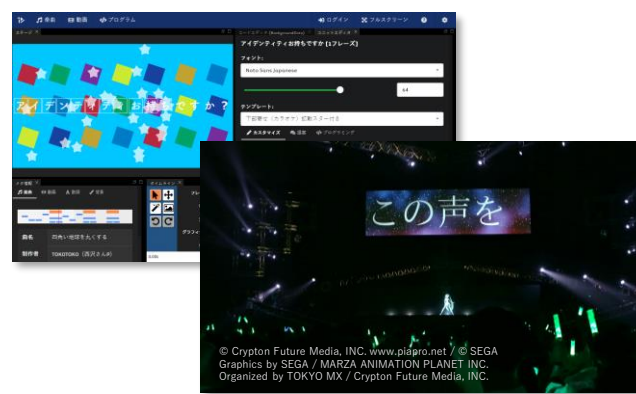
# My research style: a toolsmith researcher

A researcher who builds computational tools for creative people  
(creativity support tools, toolkits, programming environments)



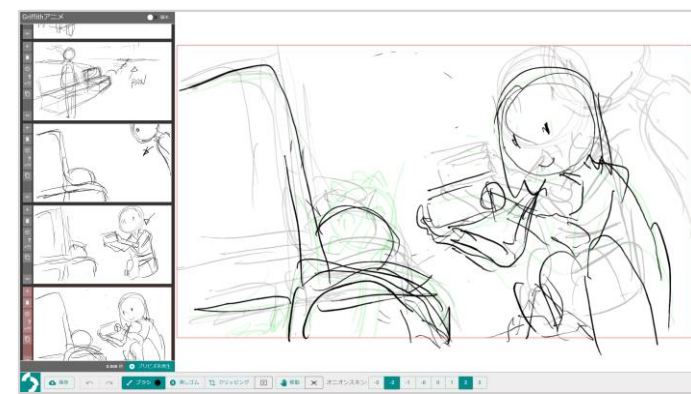
## Programmers

DejaVu [UIST '12], Picode [CHI '13  
Honorable Mention],  
TouchDevelop [PLDI '13]



## Musicians

TextAlive [CHI '15 Honorable Mention],  
Lyric App Framework  
[CHI '23 Honorable Mention]

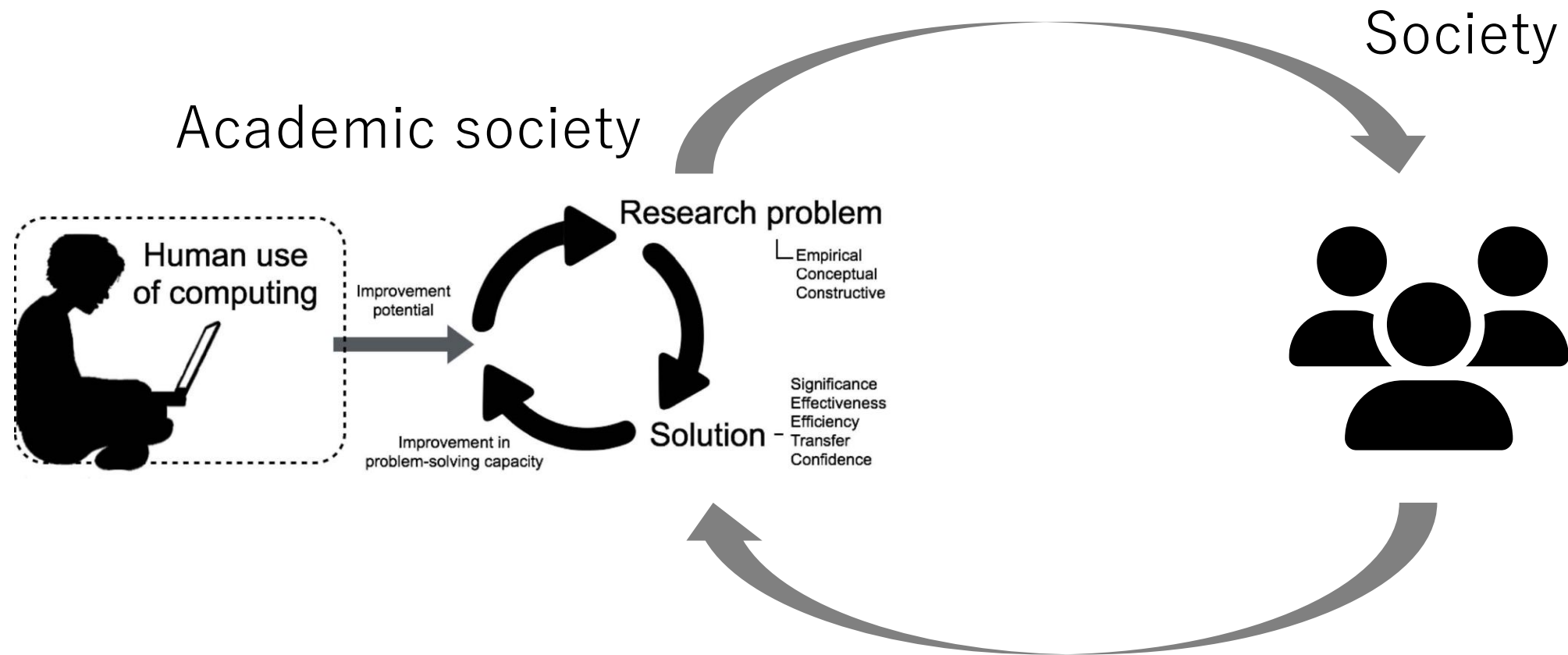


## Anime directors

Griffith [SAS '21 (non-archival),  
ISID '21 & '22 Best Poster Awards,  
CHI '24]

More examples at <https://junkato.jp/projects>

# Last week: research is not a dead-end



“HCI Research as Problem-Solving” Antti Oulasvirta [ACM CHI 2016]

# Let's go against irony of expertise

- 医者
  - Japan: 医者の不養生
- 靴屋
  - France: Les cordonniers sont les plus mal chaussés
  - Germany: Der Schuster hat die schlechtesten Leisten
  - UK: The shoemaker's children go barefoot
- 鍛冶屋
  - Spain: En casa de herrero, cuchillo de palo

# DevOps for research: agenda

- Development environment setup
- Continuous integration/delivery
- Web technologies for development/public relations

# Development environment setup

DevOps for research

# Development environment setup

- Package managers for Windows/macOS
- Two-factor authentication for improved security
- SSH config for custom networking
- DevContainer for consistent development experience



# Package managers for Windows/macOS

- Windows
  - winget [**recommended**]
  - scoop, Chocolatey
- macOS
  - Homebrew [**recommended**]
  - macPorts

# Two-factor authentication for improved security

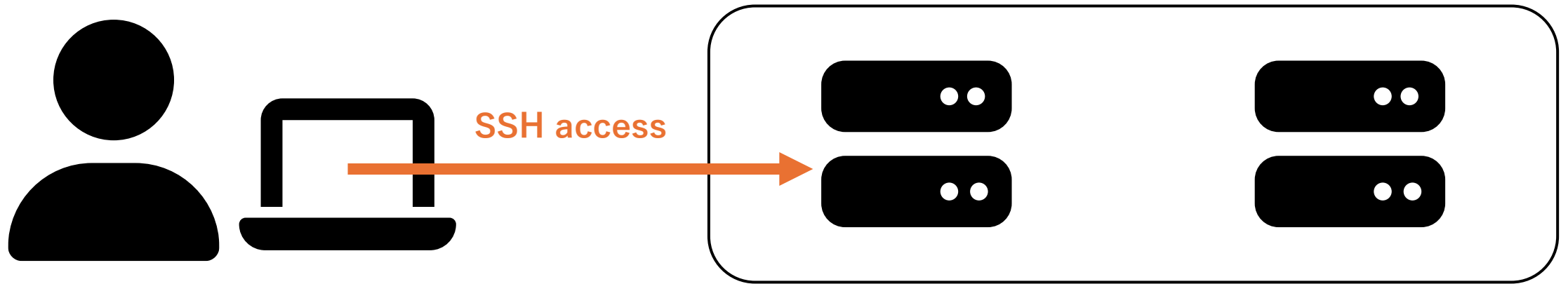
- Microsoft Authenticator
- Authy [**recommended**]
- 1Password

# SSH config

- Cf. Week 1: SSH + server coding
- You can do a lot with SSH, probably more than you'd expect

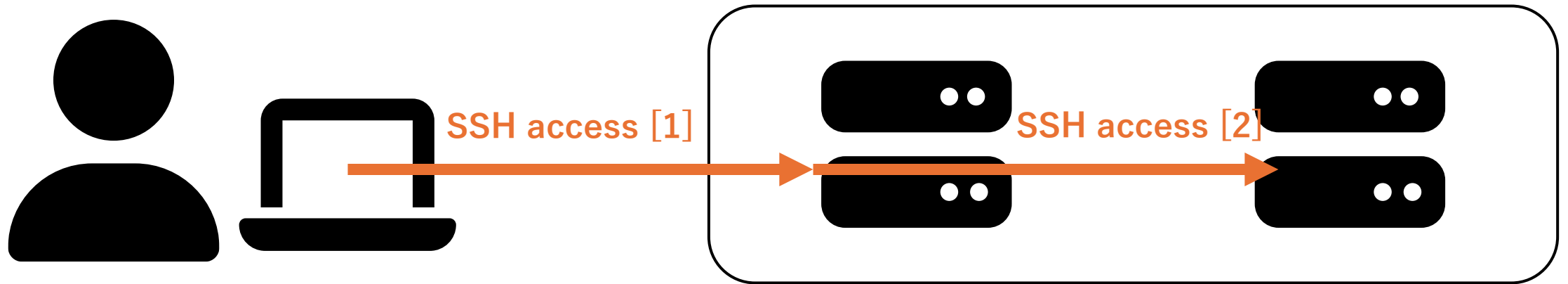
# SSH config case studies 1:

## ProxyJump for jumping hosts



- Suppose you have a server that can only be accessed from within a restricted network that includes another server
- You have SSH keys to access both the bastion host (intermediate server) and the final destination server

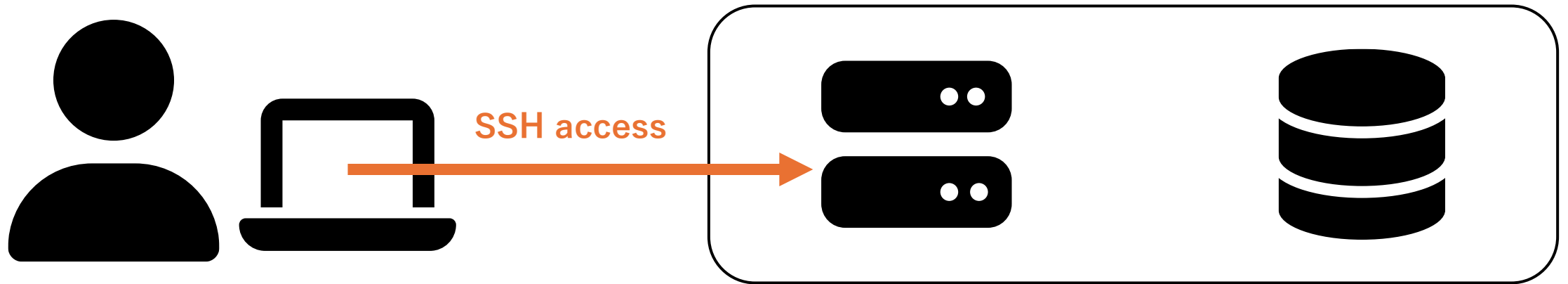
# SSH config case studies 1: **ProxyJump** for jumping hosts



```
1 Host ssh-config-proxy-jump
2   HostName xxx.xxx.xxx.xxx
3   ProxyJump ssh-config-bastion
4 Host ssh-config-bastion
5   HostName final-destination
```

SSH config case studies 2:

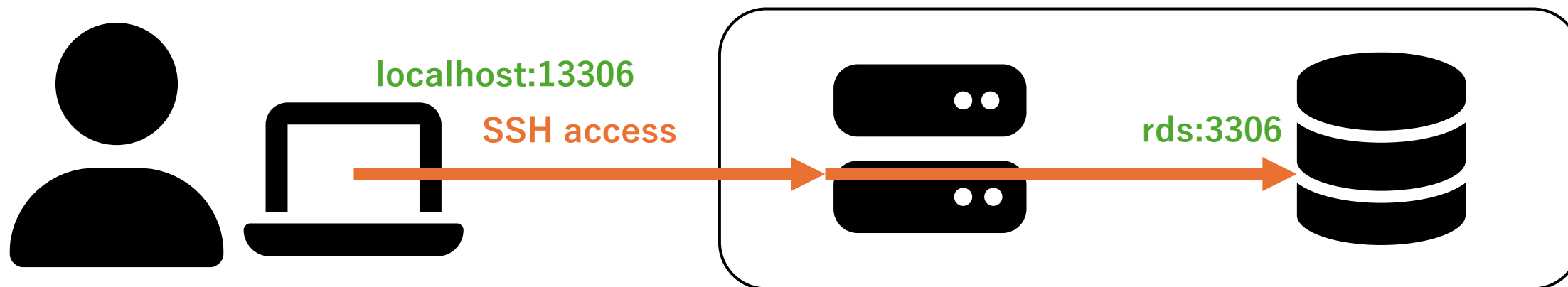
## LocalForward for remote service access



- Suppose you have a database server that can only be accessed from within a restricted network that includes an app server
- You have SSH access to the app server, but not to the database server

SSH config case studies 2:

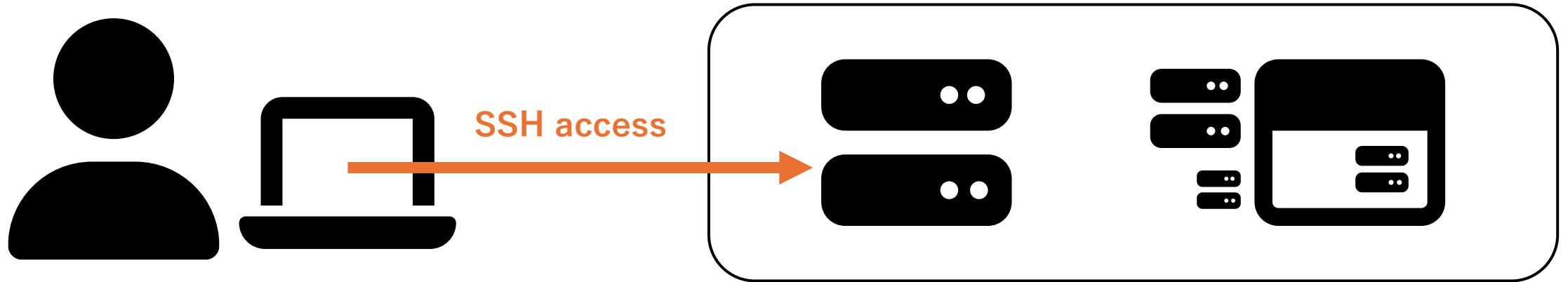
# LocalForward for remote service access



```
1 Host ssh-config-localforward
2   User ec2-user
3   HostName xxx.xxx.xxx.xxx
4   IdentityFile ~/.ssh/xxx.pem
5   LocalForward 13306 rds:3306
```

SSH config case studies 3:

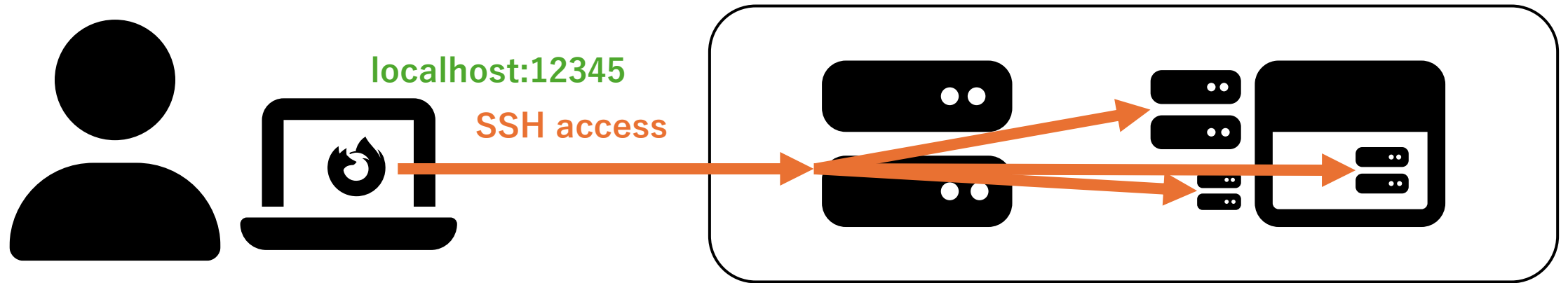
## **DynamicForward** for SOCKS v5 proxy



- Suppose you have a service that can only be accessed from within a restricted network that includes an app server
- You have SSH access to the app server, but the service runs on multiple servers to be accessed via a web browser



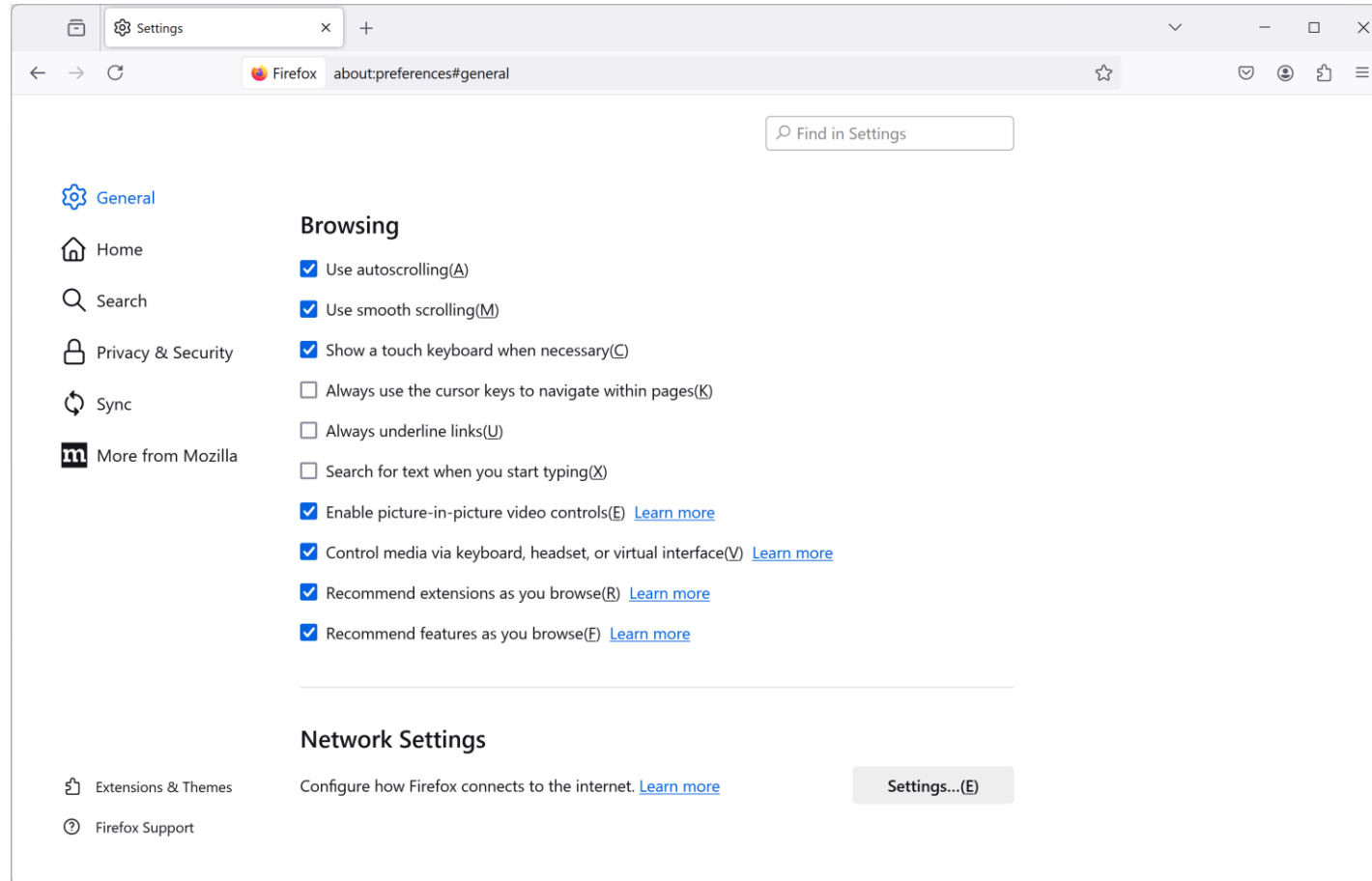
# SSH config case studies 3: **DynamicForward** for SOCKS v5 proxy



```
1 Host ssh-config-dynamicforward
2   User remote-user
3   HostName xxx.xxx.xxx.xxx
4   IdentityFile ~/.ssh/xxx.pem
5   DynamicForward 12345
```

# SSH config case studies 2: **DynamicForward** for SOCKS v5 proxy

Firefox browser  
config:  
**Network Settings**

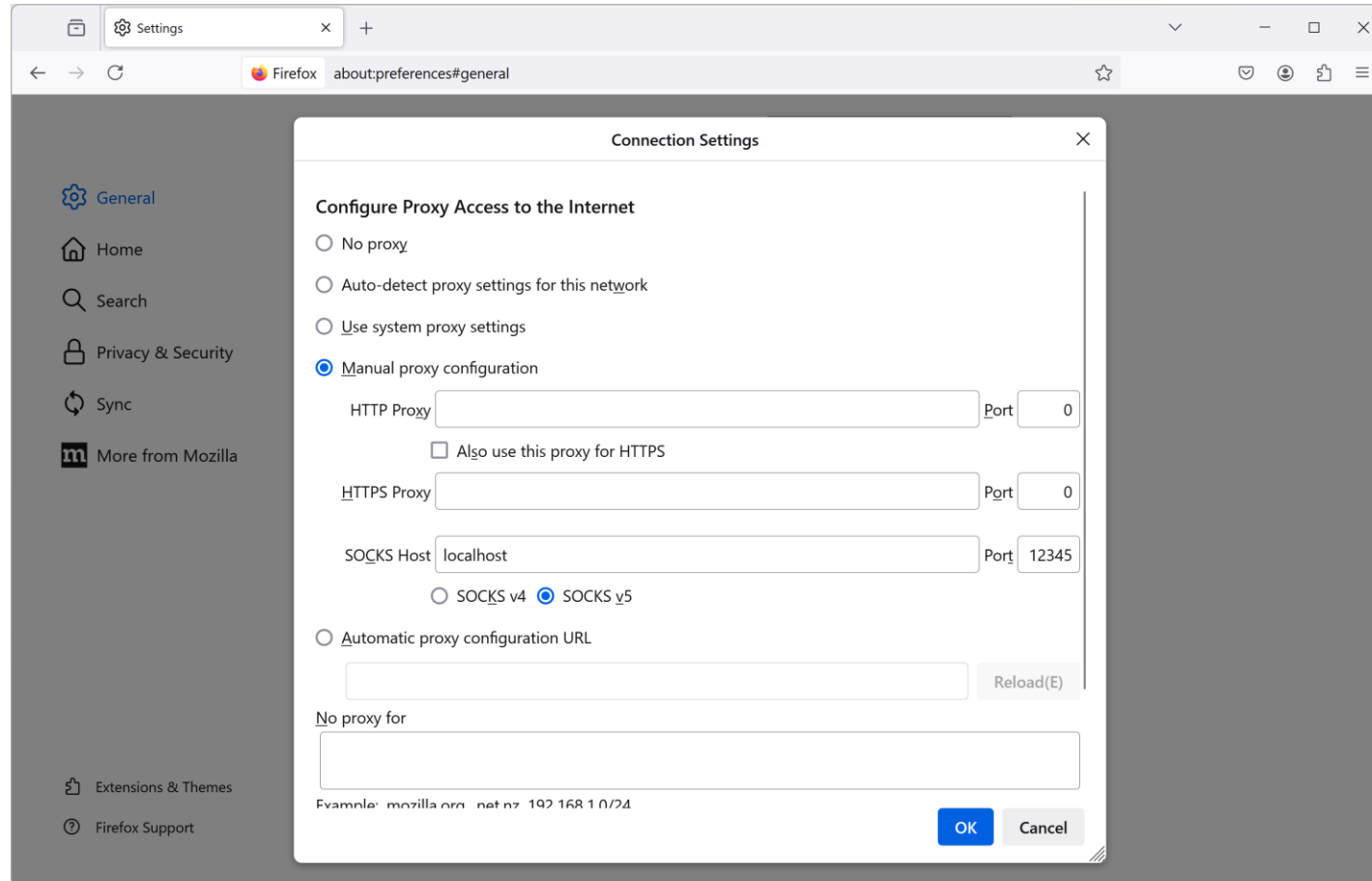


# SSH config case studies 2: **DynamicForward** for SOCKS v5 proxy

Firefox browser  
config:

Network Settings >  
Manual proxy  
configuration >

**SOCKS Host, Port,  
SOCKS v5**



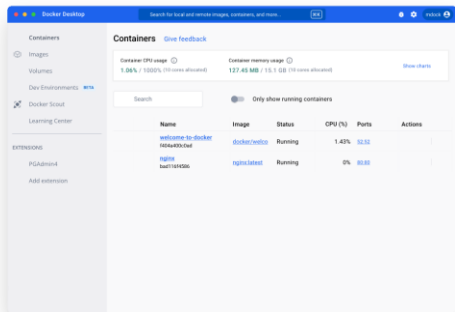
# DevContainer (VS Code feature)

- Prerequisite: Docker Desktop (or Rancher Desktop)
- Useful feature for Windows: WSL2
- DevContainer for consistent development experience

# DevContainer (VS Code feature): Prerequisite

## Docker Desktop

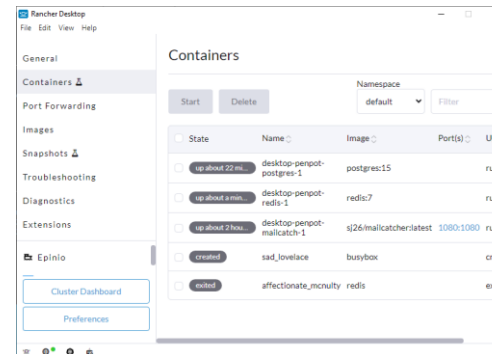
- Free for educational, non-commercial, personal, and small business use
- **docker** and **docker-compose** command support



<https://www.docker.com/wp-content/uploads/2023/08/docker-desktop-hero-v2.svg>

## Rancher Desktop

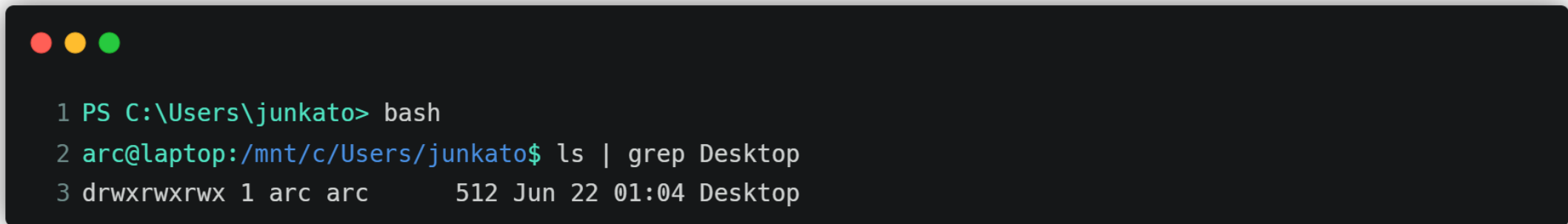
- Completely OSS, free for anyone who agrees the terms of use



[https://suse-rancher-media.s3.amazonaws.com/desktop/v1.14/ui-main/Windows\\_Containers.png](https://suse-rancher-media.s3.amazonaws.com/desktop/v1.14/ui-main/Windows_Containers.png)

# DevContainer (VS Code feature): Useful Windows feature: WSL2

- Windows Subsystem for Linux 2
- To install and start using bash on Ubuntu:
  - `wsl --install` in the administrator privileged console
  - Restart system
  - `wsl --set-default-version 2` in the admin console
  - `bash` in console

A terminal window with a dark background and three colored window control buttons (red, yellow, green) in the top-left corner. The terminal shows three lines of text: 1. A Windows command prompt prompt 'PS C:\Users\junkato>' followed by the command 'bash'. 2. A Linux shell prompt 'arc@laptop:/mnt/c/Users/junkato\$' followed by the command 'ls | grep Desktop'. 3. The output of the command: 'drwxrwxrwx 1 arc arc 512 Jun 22 01:04 Desktop'.

```
1 PS C:\Users\junkato> bash
2 arc@laptop:/mnt/c/Users/junkato$ ls | grep Desktop
3 drwxrwxrwx 1 arc arc 512 Jun 22 01:04 Desktop
```

# DevContainer (VS Code feature) for consistent development experience

- Use the same runtime (e.g., Node.js)
- Use the same package manager (e.g., npm)
- Use the same backend server (e.g., MySQL)
- Use the same VS Code extensions, editor config, etc.

...

Want a consistent development experience across machines?

**DevContainer has you covered!**

# DevContainer (VS Code feature) for consistent development experience

```
1 {
2   "name": "Example dev container",
3   "dockerComposeFile": ["/docker-compose.yml"],
4   "service": "node",
5   "workspaceFolder": "/workspace",
6   "customizations": { "vscode": {
7     "extensions": [
8       "esbenp.prettier-vscode",
9       "EditorConfig.EditorConfig",
10    ]
11  } },
12  "forwardPorts": [3000, "mysql:3306", "prisma:5555"],
13 }
```

Example [devcontainer.json](#)





# DevContainer (VS Code feature) for consistent development experience



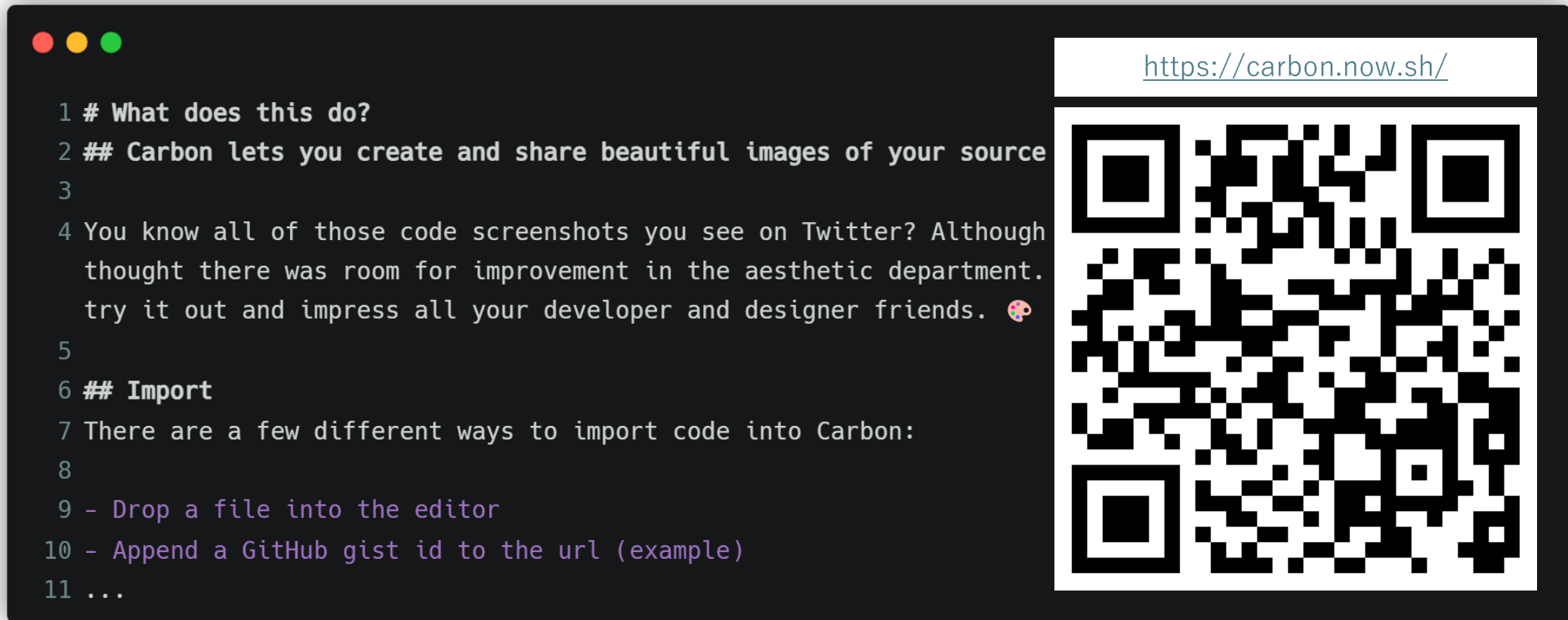
```
1 version: "3.7"
2 services:
3   mysql:
4     image: mysql:8
5     healthcheck: { test: ["CMD", "mysqladmin ping -h localhost"] }
6   prisma:
7     # mysql client
8   node:
9     build:
10      context: ../
11      dockerfile: ../.devcontainer/Dockerfile
12      command: sleep infinity
13      # more config entries omitted
```

Example [docker-compose.yml](#)




By the way...

# Share beautiful images of your code



```
1 # What does this do?
2 ## Carbon lets you create and share beautiful images of your source
3
4 You know all of those code screenshots you see on Twitter? Although
  thought there was room for improvement in the aesthetic department.
  try it out and impress all your developer and designer friends. 🎨
5
6 ## Import
7 There are a few different ways to import code into Carbon:
8
9 - Drop a file into the editor
10 - Append a GitHub gist id to the url (example)
11 ...
```

<https://carbon.now.sh/>



# Continuous integration/ continuous delivery

DevOps for research

# Continuous integration/continuous delivery

- GitHub or GitLab?
- Self-hosted runners
- CI/CD use cases 1: linting papers
- CI/CD use cases 2: running heavy workloads on clouds
- CI/CD use cases 3: ChatGPT integration
- CI/CD use cases 4: publishing websites

# GitHub or GitLab?

from CI/CD perspective

## GitHub Actions

- Not much difference for basic usage
- writes config under `~/.github/workflows/`
- **Pros:** Has a vast library of community-defined “Actions” useful for various purposes
- **Cons:** You’re vendor-locked!

## GitLab CI/CD

- writes config in `~/.gitlab-ci.yml`
- **Pros:** Can be hosted entirely on your own servers without worrying much about resource limits
- **Cons:** Maintenance costs

# GitLab CI/CD basic example

```
.gitlab-ci.yml
```

```
1 image: docker:dind
2
3 before_script:
4   - docker login -u $CI_REGISTRY_USER -p $CI_REGISTRY_PASSWORD $CI_REGISTRY
5
6 stages:
7   - build
8   - deploy
9
10 variables:
11   IMAGE_PATH: ${CI_REGISTRY}/${CI_PROJECT_NAMESPACE}/${CI_PROJECT_NAME}
12   CONTAINER_NAME: ${CI_PROJECT_NAME}-${CI_JOB_NAME}
13 # more config entries omitted
```

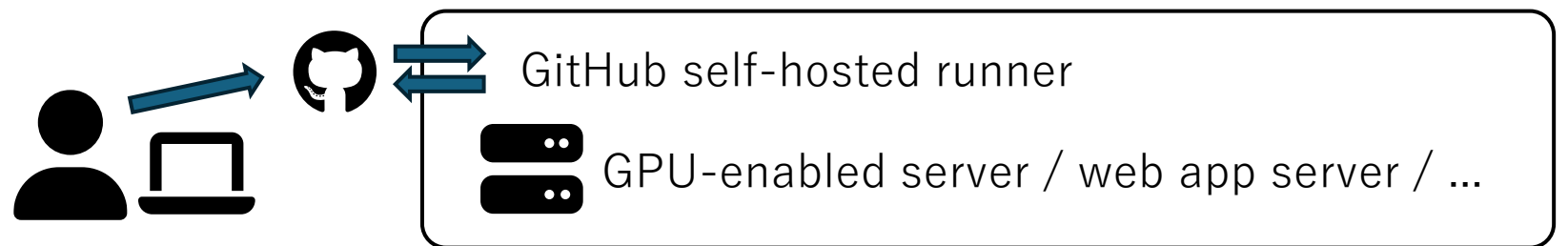
Example [.gitlab-ci.yml](#)



# Self-hosted runners

## Basics and usages

- GitHub Actions and GitLab CI/CD both support self-hosted runners – CI/CD workers that run on machines under your control, such as **modern NAS devices** and various servers
- These are especially useful if
  - you have **a handy machine (or some special machines)** and want to avoid being charged for running an automated process
  - you have **a deployment target on cloud computing platforms** and want to securely update running services on each push



# CI/CD case studies 1: Editing, linting, and sharing papers

Overleaf + GitHub Actions + GitHub Pages + Slack

- Overleaf has a good GitHub integration by default
- GitHub Actions can automatically process pushed TeX files
- GitHub Actions can upload the end result to GitHub Pages
- GitHub Actions can post a message to Slack upon completion

<https://github.com/arcatdmz/acm2024-primary-article-template>



**Let's look at the actual code on GitHub:**



# CI/CD case studies 2: Running heavy workloads on clouds

## Amazon EC2 + GitHub + GitHub Actions

- Automated self-hosted runner management with GitHub personal access token of **repo** scope
- Automated EC2 instance management with AWS access keys with permissions for **ec2:\*** actions
- On-demand tests/experiments with EC2 instance!  
<https://github.com/machulav/ec2-github-runner>

# CI/CD case studies 3: ChatGPT integration

GitHub + GitHub Actions + OpenAI Platforms API

For each GitHub comment,

- look for harmful text ([moderations](https://github.com/takahiroanno2024/election2024/blob/main/.github/workflows/comment-review.yml#L89-L96) endpoint)
- look for URLs and check if they look harmful

# CI/CD case studies 4: Publishing web services

**To be covered in the next section!**

# Web technologies for development/public relations

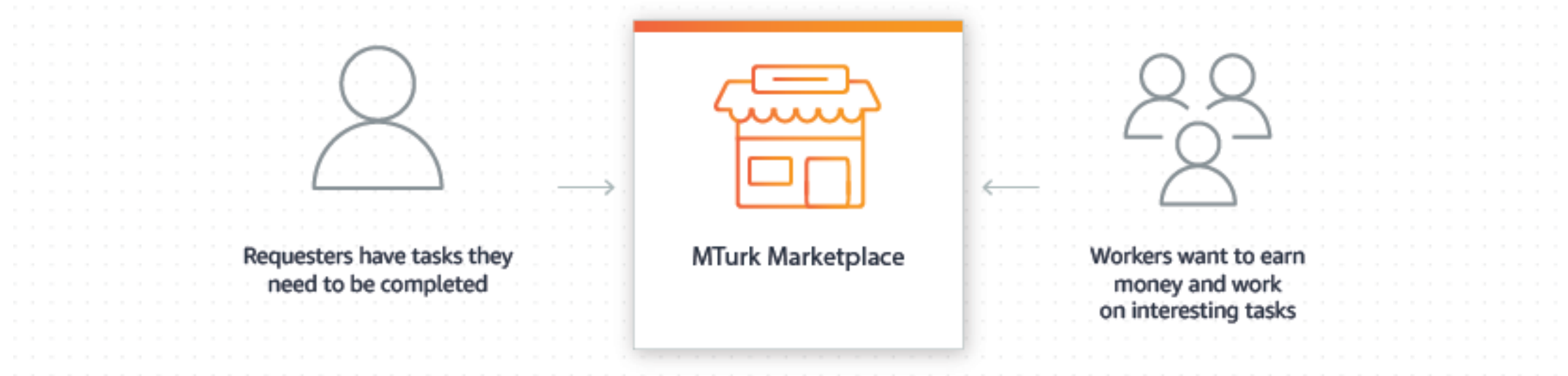
DevOps for research

# Web technologies for development/public relations

- Crowdsourcing
- npx + Prettier for code beautification
- Infrastructure for web deployments
- Languages and frameworks
- Tips for making informative portfolio pages

# Crowdsourcing

<https://www.mturk.com/>



- Amazon Mechanical Turk, Yahoo! Crowdsourcing cf. [acarin website](#)
- Crowdsourcing and LLMs (search [llm modeling human behaviors to replace crowdsourcing](#) or visit [HCOMP website](#))

# npx + Prettier for code beautification

- `npx` comes with Node.js by default and Prettier is a well-known OSS code formatter
- `npx prettier --write {your files}` is useful, even if you don't write JavaScript (e.g, HTML and Markdown files; more languages if combined with plugins)
- `npx http-server {your directory}` is also useful
- See also: `live-server`, `localtunnel`, `cpx`, `cross-env`, ...

**Tell me your favorite npm package!**

# Infrastructure for web deployments

- For static sites: GitHub Pages and Cloudflare Pages
- For a bit more complex sites: Netlify and Vercel
- For backends: PlanetScale and... Google Sheets!
- For professional-minded developers: AWS, GCP, Azure



# Web technologies case studies 1: Cloudflare Pages + Google Sheets

- Google Forms → Google Sheets
- Google Sheets → Cloudflare Pages

<https://github.com/ChezTachibana/website>



**Let's look at the actual code on GitHub:**

# Languages and frameworks

- HTML, Pug, Markdown
- React (JSX/TSX), MDX, and more!
- Responsive design for free with CSS frameworks
- Pretty icons with FontAwesome and alternatives

# Languages and frameworks

## HTML, simplified

### Pug

- **html**
  - **body**
    - **h1#mytitle** Main title
    - **p** This is the main paragraph
    - **ol.myclass**
      - **li** This is the first item
      - **li** second item
      - **li** third item
    - **p** You can add
      - **a(href="#mytitle")** links
      - | pretty easily!

### Markdown

- **#** Main title **{#mytitle}**
- This is the main paragraph
  - **1.** This is the first item
  - **2.** second item
  - **3.** third item
- You can add **[links](#mytitle)** pretty easily!

# Languages and frameworks

## HTML, evolved

### React (\*.jsx)

```
// Every user interface component is a function
export function MyButton = () => {
  const handleClick = useCallback(() => { ... }, []);
  return <Button onClick={handleClick}
  className="my-button">My button!</Button>
}
```

### MDX

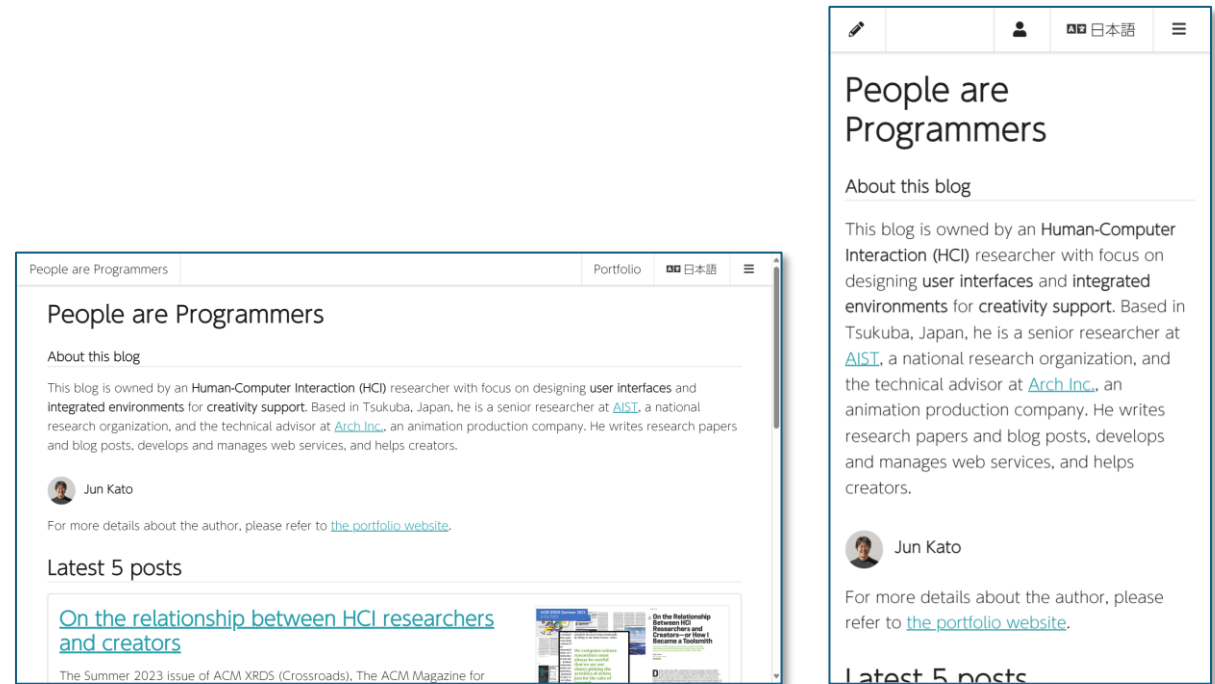
```
import { MyButton } from
"../src/MyButton";
```

This is the main paragraph  
with a custom `<MyButton />`.

# Languages and frameworks

## CSS frameworks for responsive design

- Bootstrap
- Semantic UI / Fomantic UI
- and more!



Search for “CSS frameworks”

# Tips for making informative portfolio pages

- Stick to static sites, allow easy copy and paste
- Show the story, not only the papers
- Single source of truth, multiple views for multiple purposes

# Web technologies case studies 2: Portfolio web page

<https://github.com/arcatdmz/arcatdmz.github.io>



**Let's look at the actual code on GitHub:**

# DevOps for research: agenda

- Development environment setup
- Continuous integration/delivery
- Web technologies for development/public relations



# DevOps for research