/ Das 7. Ansible Anwendertreffen

Comparing Ansible Development Environments

Agenda

1	2	3	4
Introduction & Motivation	Different Solutions and Demos	Comparison of Development Environments	Closing thoughts

Whoami

Niklas Werker (he/him)

- 📍 Cologne, Germany
- 🏢 SVA System Vertrieb Alexander GmbH
- 🧑🏻‍💻 DevOps: Infrastructure Automation

- 3 years Automotive Engineering Background: Systems Engineering (Embedded Systems)
- 7 years IT Background

- nwerker @ ANSIBLE

About this talk

- Applies to all Development Environment needs but today we'll focus on Ansible
- Hard to decontextualize from Platform Engineering
  - DevEx = Developer Experience
  - Bigger picture & integration possibilities

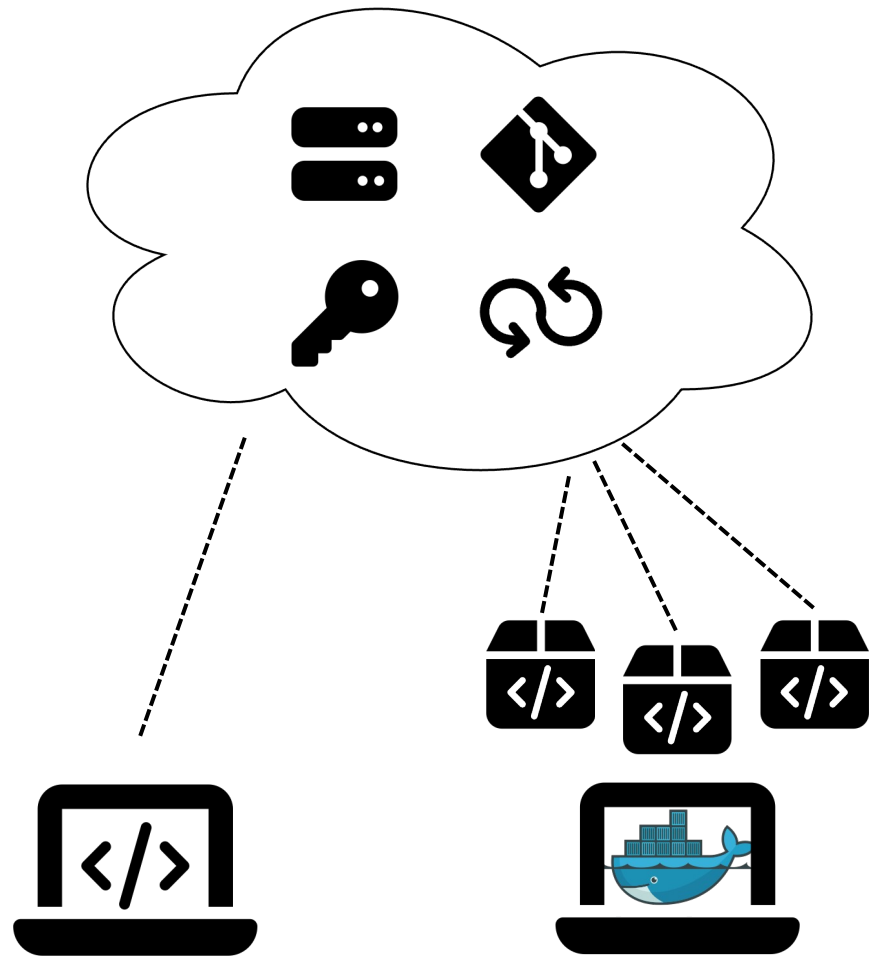- Reason: X Tools trying to solve X Problems
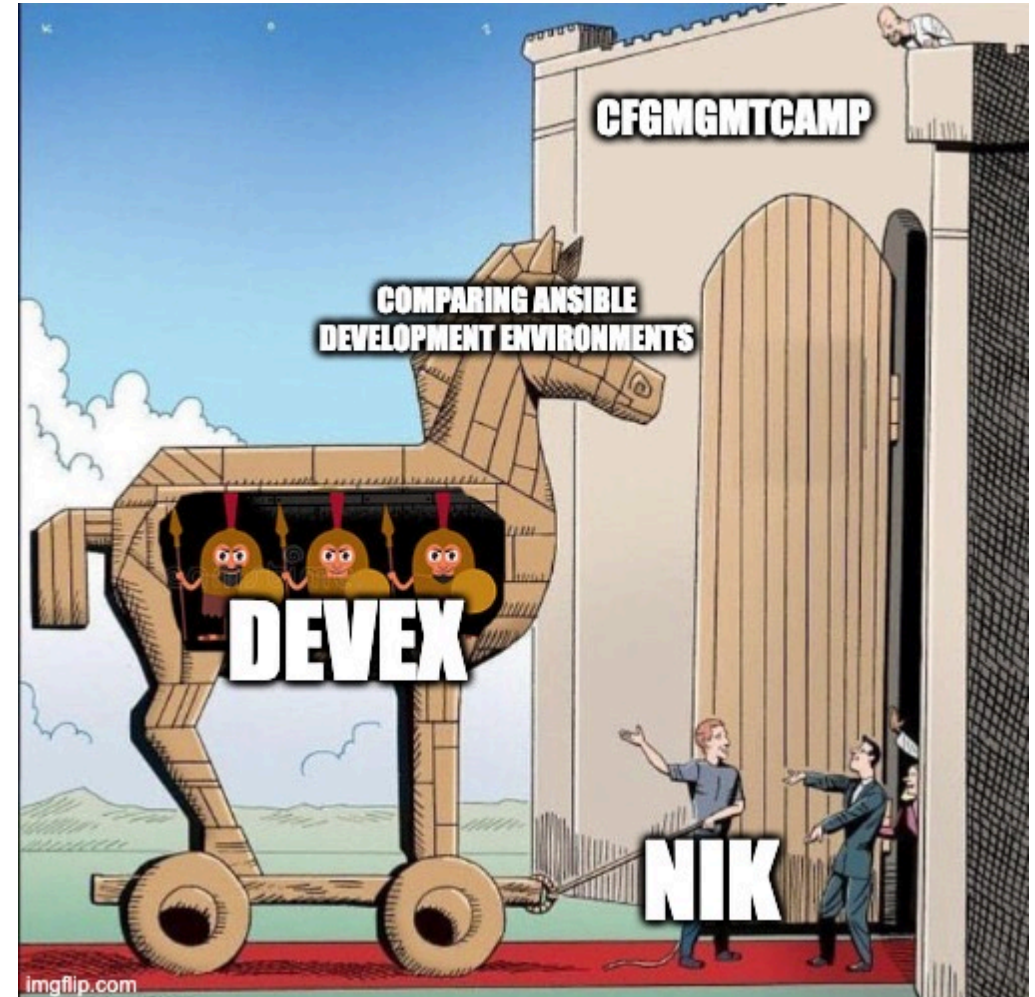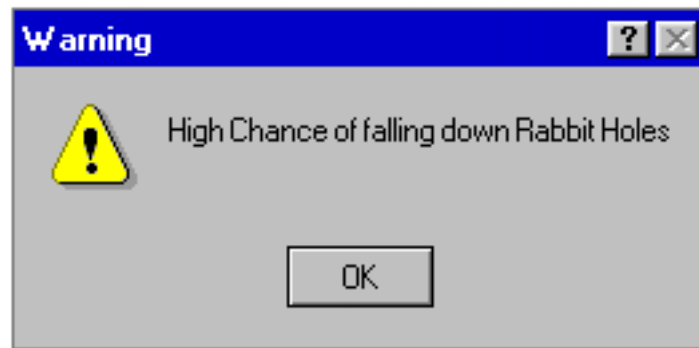
Keep it simple

- Take what you need, leave the rest
- Your mileage may vary
- The Idea is not new:
  - .dotfiles
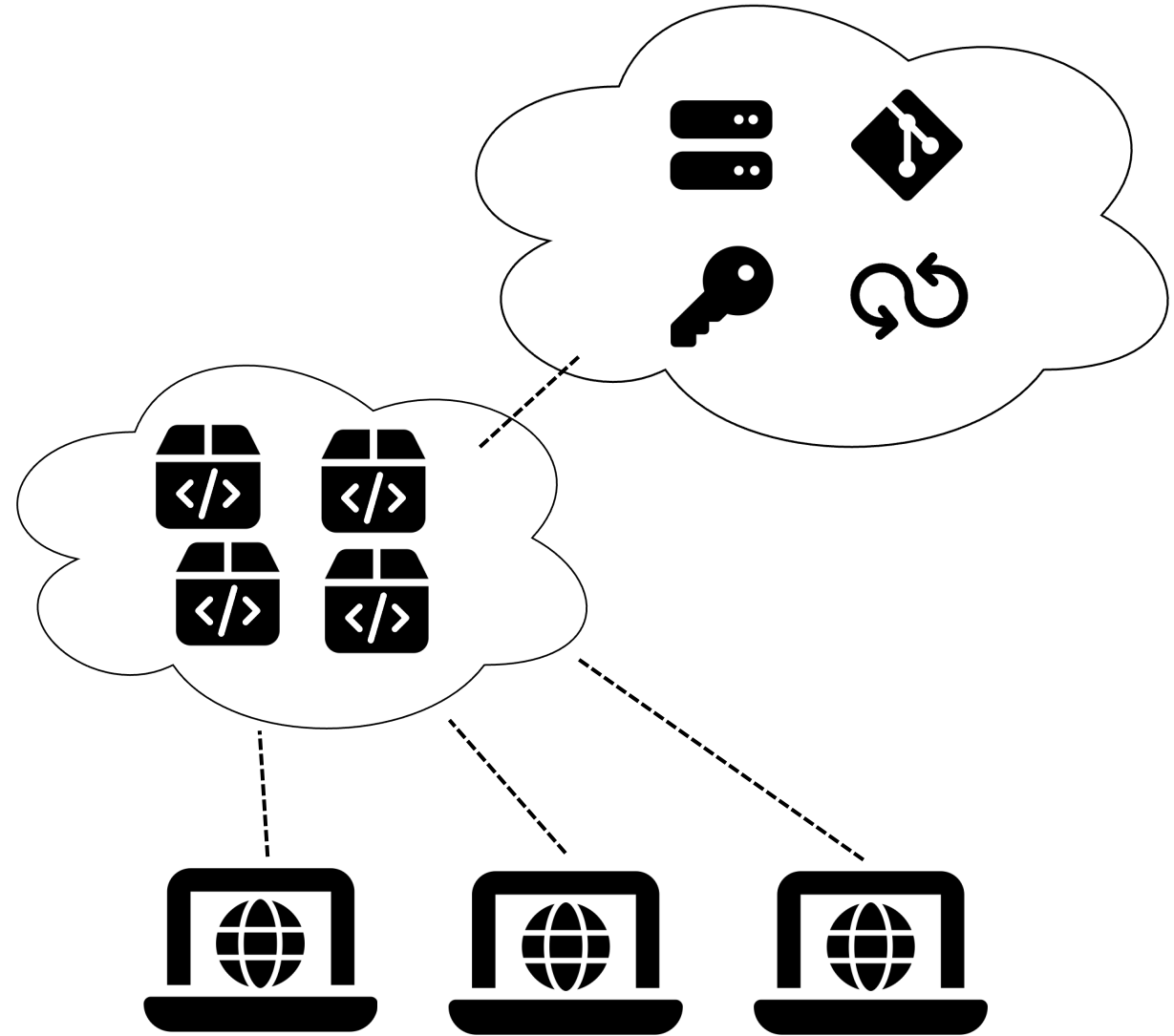 - Coder Server
  - etc.
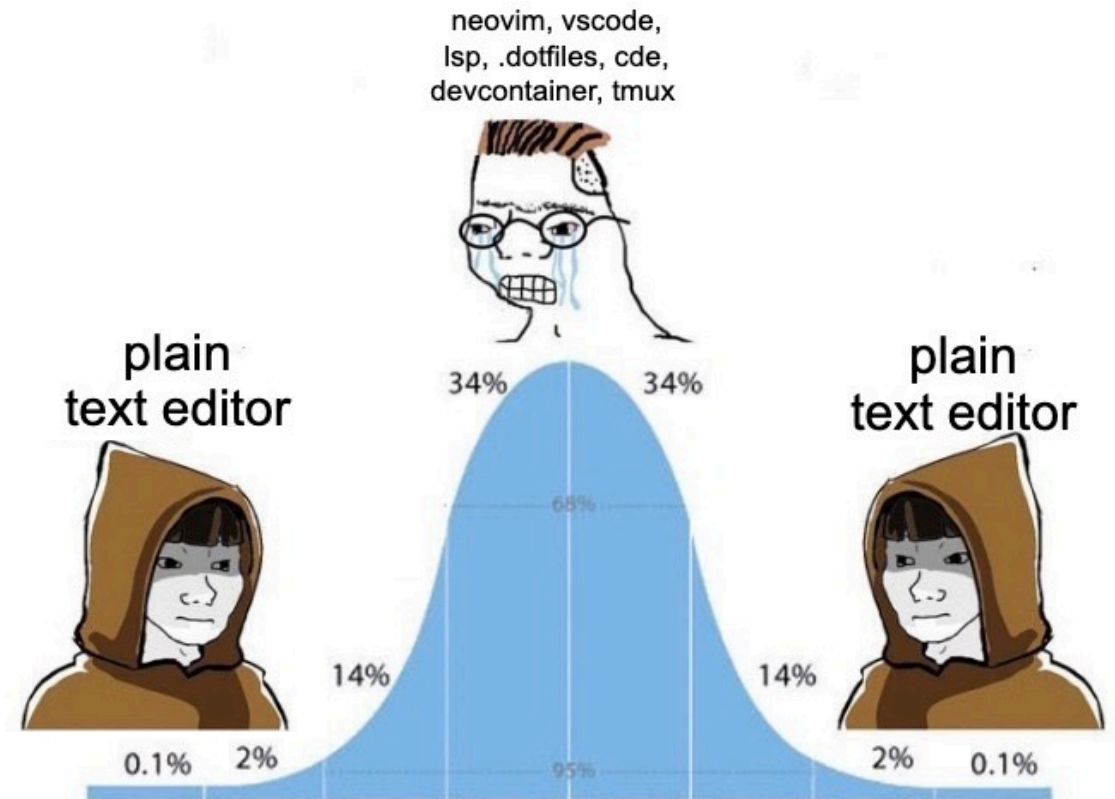
Conceptual

Classic Approach

Local Development
Environment

Remote / Cloud Development Environment

But why?!

- Reproducible
- Standardization
- Governance: Development Environment as Code
- Faster Onboarding on new Projects
- Lifecycle Challenges:
  - Align Dev Dependencies with Team and Infrastructure
- Security

- Develop on prod-like infrastructure / dependencies
  - e.g. Containerized Network Appliances
  - e.g. against systems like HashiCorp Vault, Netbox

Anatomy of a Ansible Development Environment

Ansible Content Developer

- Integrated Development Environment (IDE)
- Ansible Development Tools (ADT)
  - Including: ansible-core, ansible-lint, molecule etc.
- Ansible Language Server (LSP)
  - Syntax highlighting, validation, linting, auto-completion, doc reference
- Collections (for collection specific syntax highlighting & linting)
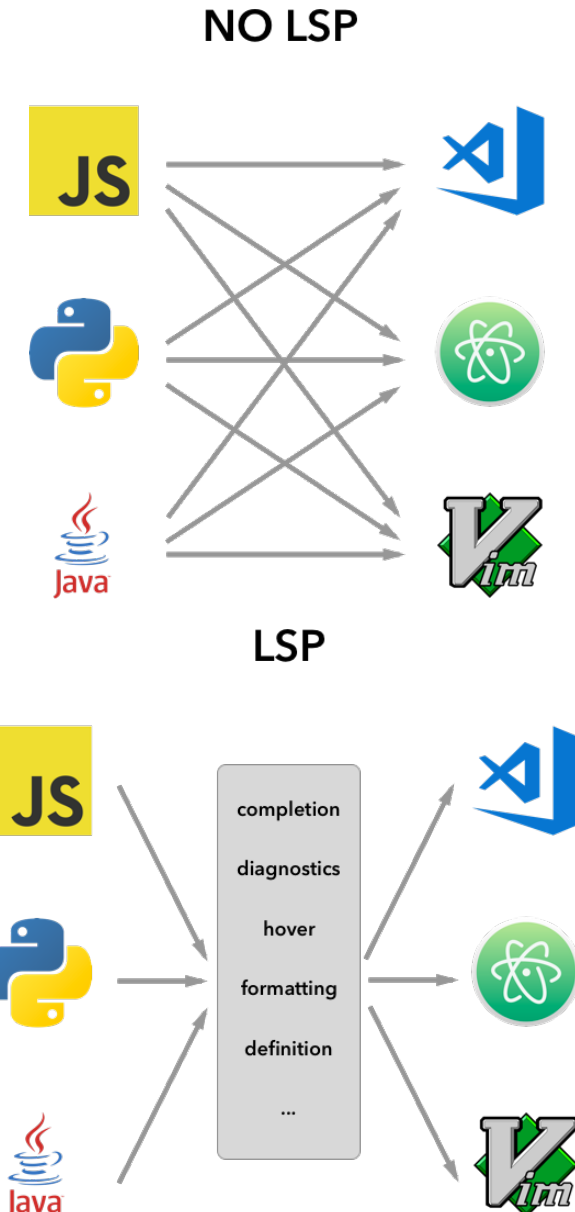
Optional:

Test Driven Development Capabilities

Collection Dependencies (python, bindep etc.)

Language Server Protocol (LSP)

- Development started 2016
- Involved:
  - Microsoft
  - Codenvy / Red Hat

- Todays standard in Language Intelligence Tooling

- Features: Syntax highlighting, validation, linting integration, auto-completion, doc reference etc.

NO LSP

LSP

Development Containers (devcontainer)

- Specification / Standard

- Declarative definition of
  - Software Dependencies (Containerfile, Compose, Image etc.)
  - IDE Dependencies (Extensions & Settings)
  - Runtime Dependencies

- A lot of different tooling:
  - Integration in Visual Studio (+Code), Jetbrains IDEs, CLI etc.
  - Tools Building on Spec: DevPod, GitPod Flex etc.

- SaaS Solution from Microsoft: GitHub Codespaces

```
0 {
1    "name": "ansible-dev-container-docker",
2    "image": "ghcr.io/ansible/community-ansible-dev-tools:latest",
3    "containerUser": "root",
4    "runArgs": [
5      "--privileged",
6      "--device",
7      "/dev/fuse",
8      "--hostname=ansible-dev-container"
9    ],
10   "updateRemoteUserUID": true,
11   "customizations": {
12     "vscode": {
13       "extensions": ["redhat.ansible"]
14     }
15   }
16 }
```

A more detailed view

- VS Code Devcontainer takes care about the container lifecycle
- Create one or more containers
- Editor uses SSH to connect do Development Container
- SSH Agent and Git Credential Helper are getting forwarded

- Advanced features like:
  - Templates (to bootstrap projects)
  - Features (share devcontainer code)

pull
run
rebuild

ssh agent forwarding

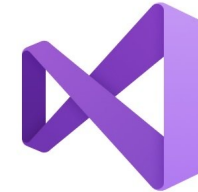ssh into container

SSH
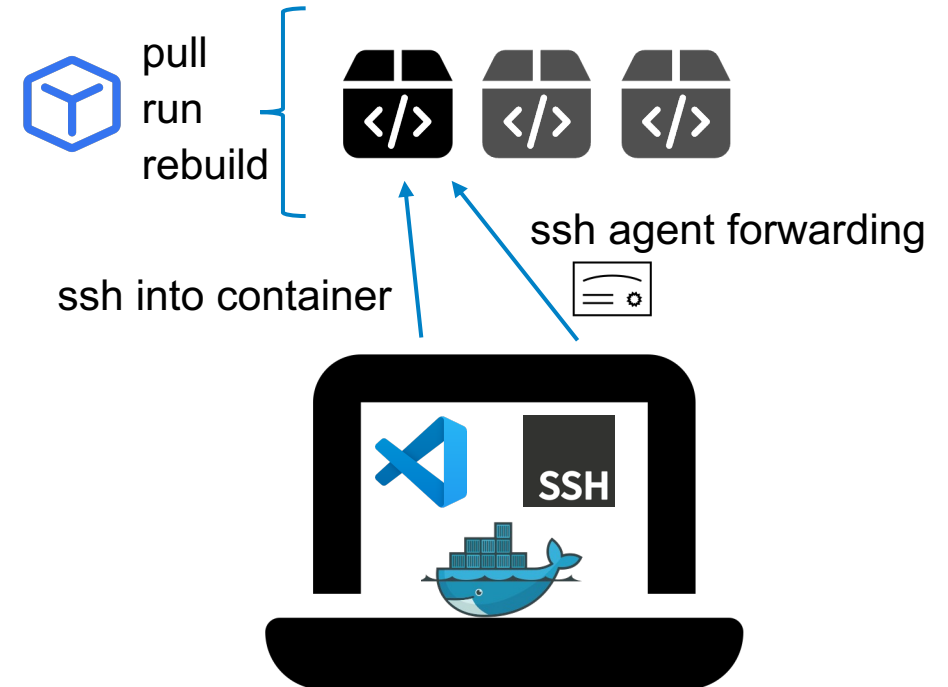
OpenSource Controversy

- All presentations must be about truly open source software.

Visual Studio Code → Install packages → Visual Studio Marketplace

OSS Alternatives → Install packages → openVSX Registry

- VSCode: Built on OSS, proprietary Microsoft license and telemetry
- Licenses and agreements permits certain usage
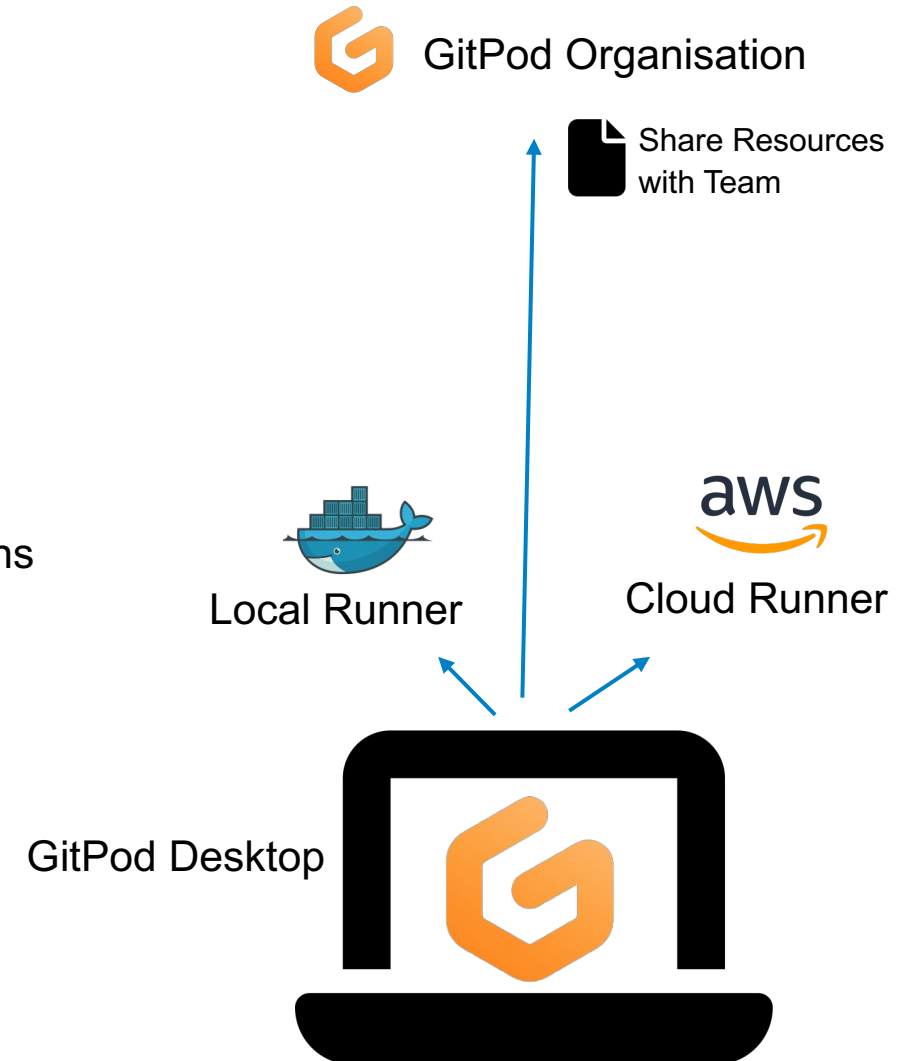- Remote Management and Development Extensions not available OSS

DevPod

- Desktop Client for managing Development Environments anywhere

- Supports
  - devcontainer Spec
  - Terraform provisioning

- Combining local and remote dev environments and different editors

- Enterprise Version in development: DevPod .Pro

- .dotfiles Integration

- Community Codium Plugin, to allow "VSCode like" feel with DevPod

integrated
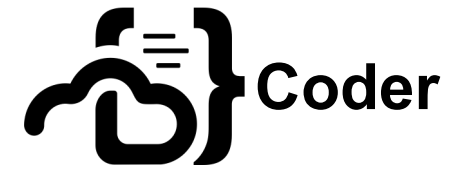providers

custom
terraform
providers
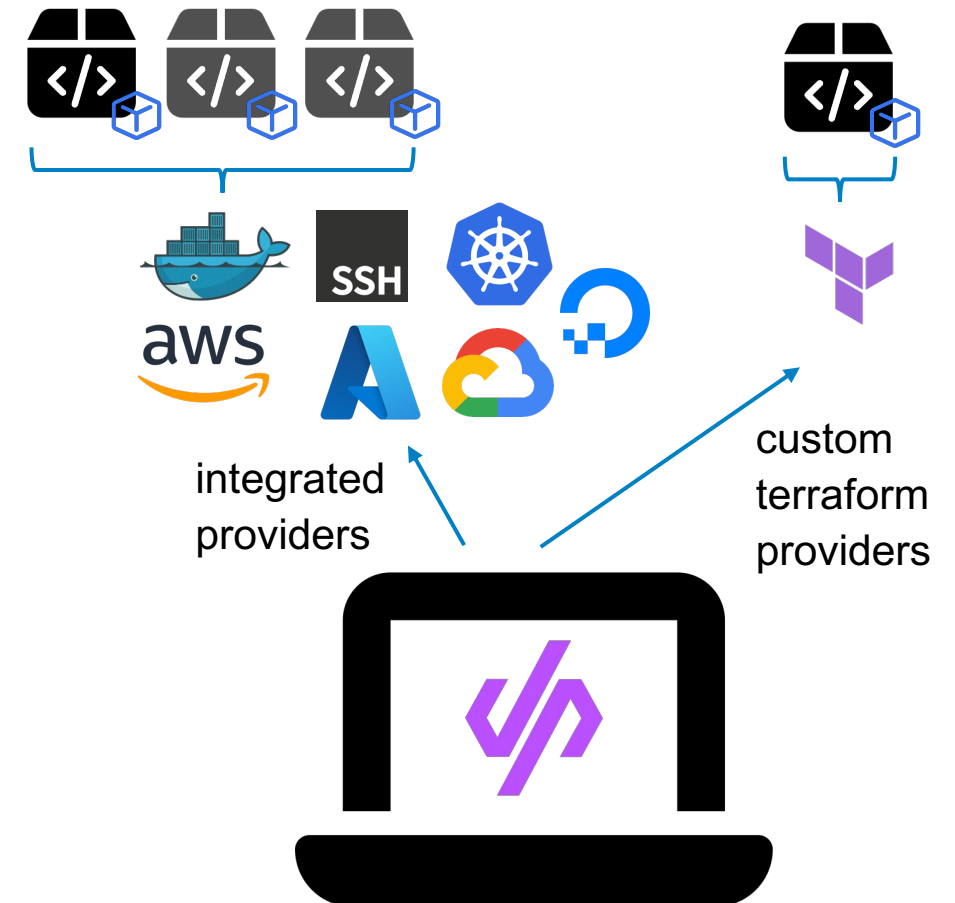
GitPod

2 different products:
- GitPod Classic (EOL 04/25)
  - SaaS Including alternative Spec to
- GitPod Free, Flex (early access), Enterprise
  - Implementing Devcontainers
  - Local Environment & Self Hosted in Cloud
  - Management Plane on GitPod infrastructure

- Early Access of GitPod Flex came with some flaws
  - e.g. Performance / Rosetta Virtualization and missing VSCode Extensions

- Adds Features like "Automations" & sharing Projects with teams
  - More to come

GitPod Organisation

Share Resources with Team

Local Runner

aws
Cloud Runner

GitPod Desktop

Coder

- Makers of "code-server"
- Selfhost management plane and provisioning of CDEs

- Terraform driven templates
  - VMs, Pods, Containers etc.

- Coder Application Single Go Binary (Server + Client) + PSQL

- Consume Workspaces via. IDE, Web IDE or Web SSH Terminal

- Enterprise:
  - HA / Scalability
  - Advanced Networking and Security Features
  - Lifecycle Features

- Integration into Secret Management (HashiCorp Vault) or Artifactories

create

consume
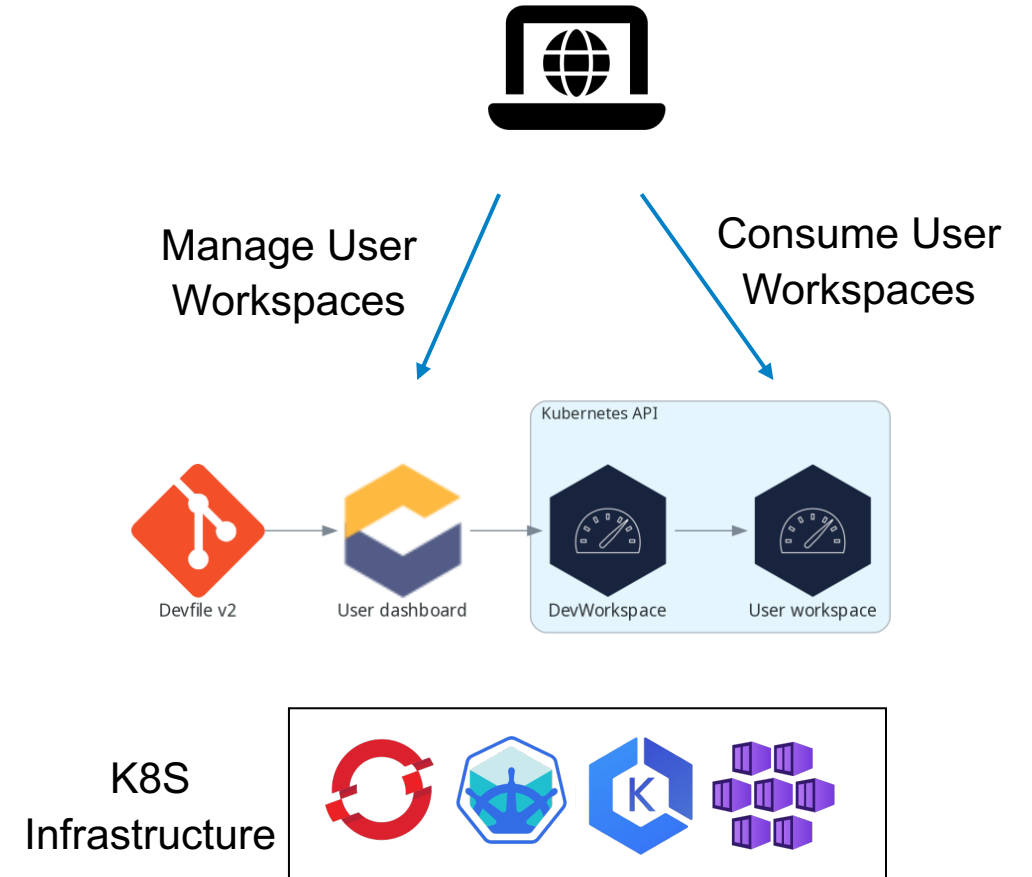
Templates

Workspaces

Coder Server
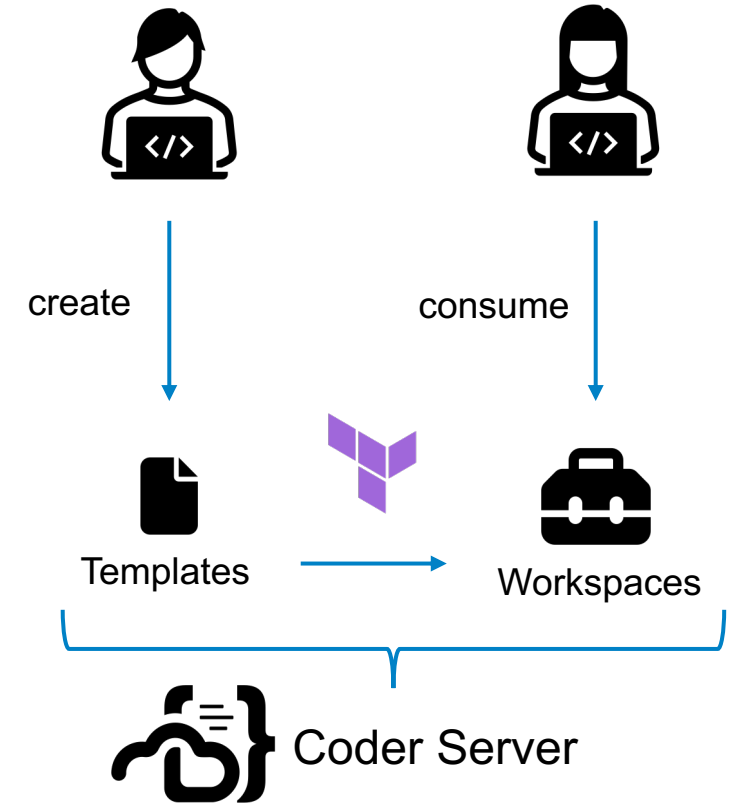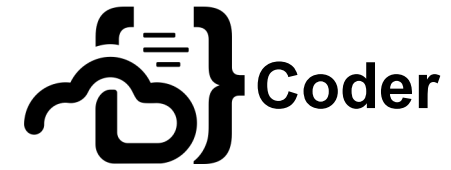
Coder

- Devcontainer Support via.: Coder Envbuilder
  - Abstraction of Devcontainer Lifecycle Stages
  - Builds Devcontainer and Image to Coder Specific Image

- No integration to automatically install ide extensions
  - Can be installed in Client IDE or via. Coder Script in Coder Server Workspace Instance

- Traditionally Coder Templates are completely defined within Terraform and "Coder Agent Scripts"

```
# Install Red Hat Ansible Extension
SERVICE_URL=https://open-vsx.org/vscode/gallery \
  ITEM_URL=https://open-vsx.org/vscode/item \
  /tmp/code-server/bin/code-server \
  --install-extension redhat.ansible
```

Eclipse Che

- Kubernetes based Cloud Development Environments

- Based on Devfile Spec

- User consumes IDE, Project and Kubernetes Namespace

- Tailored to Cloud Native Development
  - Well suited for Test Driven Development (e.g. Molecule)

- Enterprise Version: Red Hat OpenShift Devspaces
  - fka: CodeReady Workspaces / Codenvy
  - Red Hat Developer Sandbox Account (Testing)

Manage User Workspaces

Consume User Workspaces

Devfile v2 → User dashboard → Kubernetes API: DevWorkspace → User workspace

K8S Infrastructure

Common Enterprise Features

- Scalability and High Availability
- Pre Build Environments
- Scheduled Downtimes and Availability
- Lifecycle: Enforce certain template Versions
- Scheduled Uptime and Availability
- Metrics and Telemetry
- RBAC
- Onboard Contractors
- Cost Control
- Governance
- Support / SLAs etc.
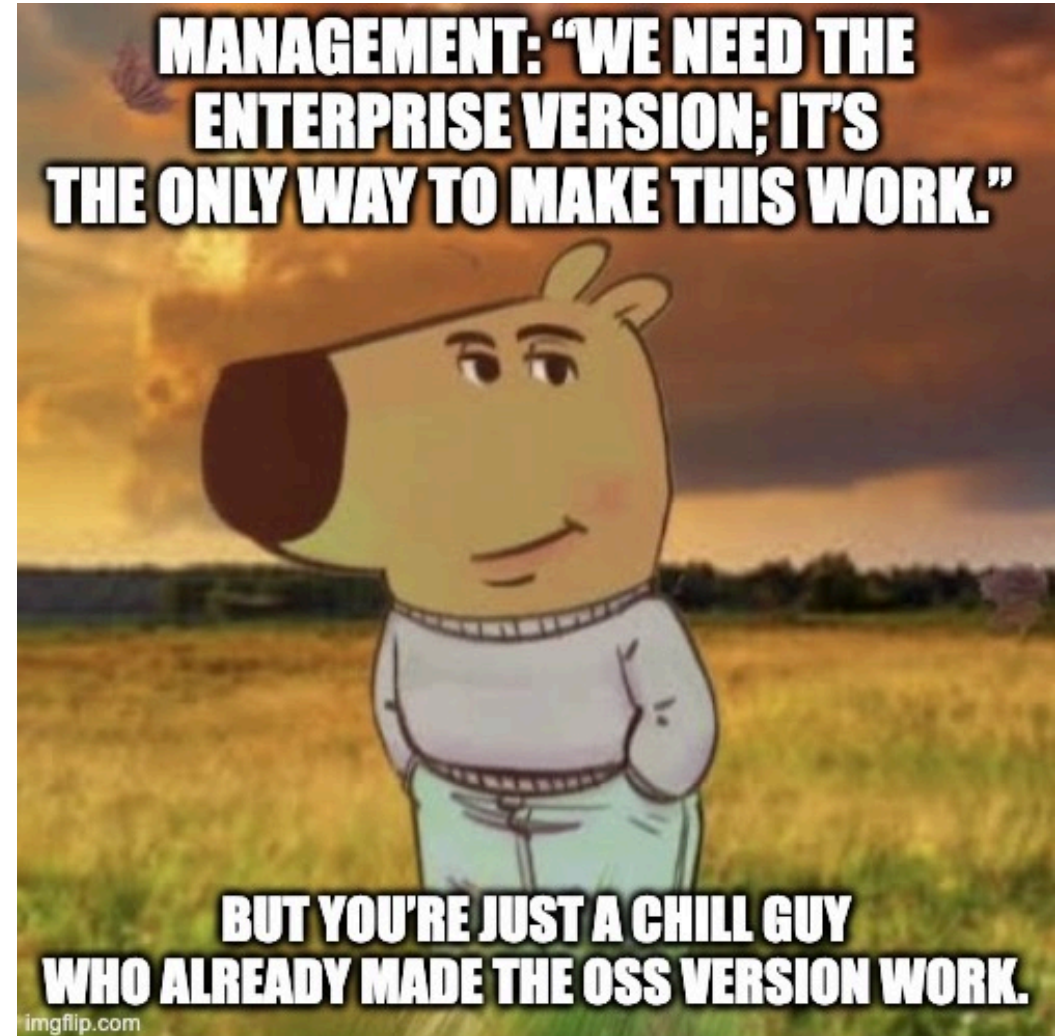
Entry points to interact with Development Environments
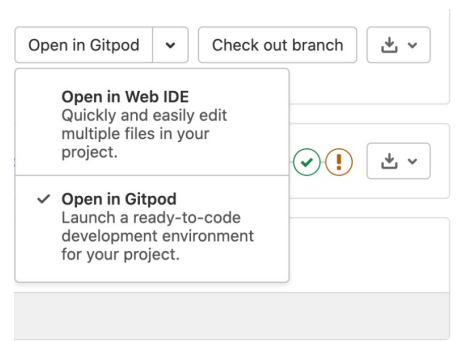
IDE

Web Portal

Browser Extention

Internal Developer Portal

CLI

Desktop Client

| Open in Gitpod ▾ | Check out branch | ⤓ ▾ |

Open in Web IDE
Quickly and easily edit
multiple files in your
project.

✓ **Open in Gitpod**
Launch a ready-to-code
development environment
for your project.

Merge Request

To get started, simply click the button below...

◆ Developer Workspace
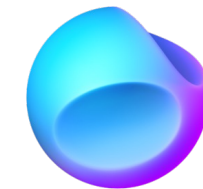
Markdown Badges
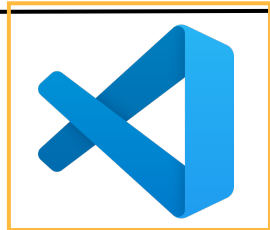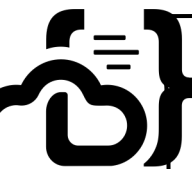
Comparison of supported IDEs
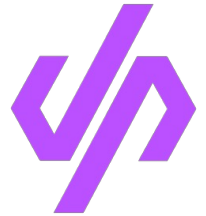

VS Code   JetBrains   Cool Kids   Data Science

Infrastructure point of view

Development Containers: IDE Dependent, Environments on OCI Container Runtime

DevPod: Local Client, Environments Everywhere

GitPod: Management Plane in Cloud, Environments Locally and in Cloud
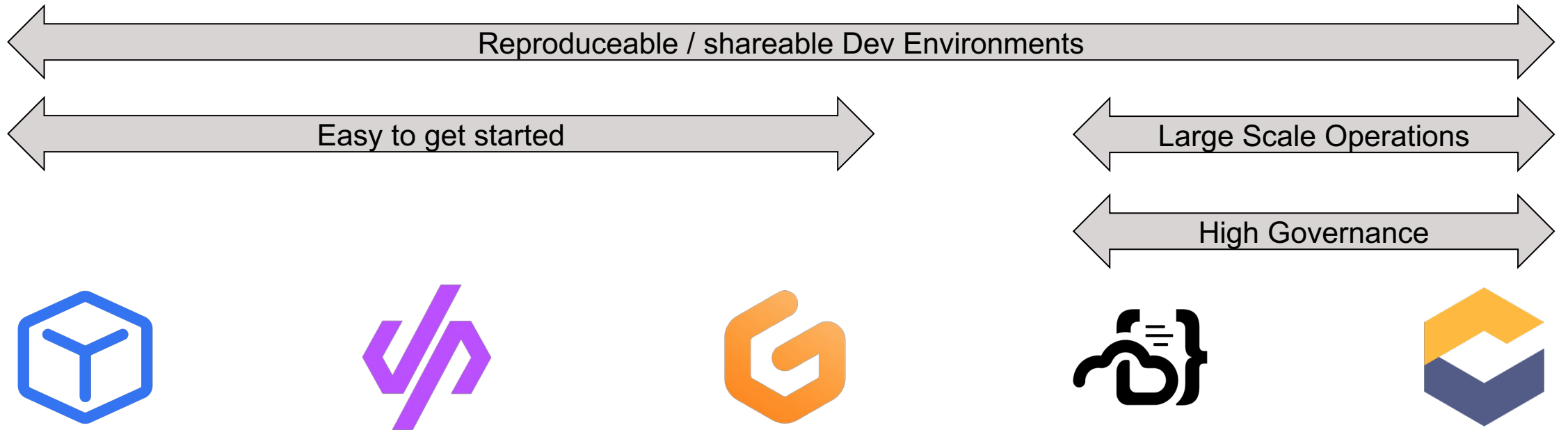
Coder: Coderserver Everywhere, Environments Everywhere

Eclipse Che: Both K8S

Sweet Spots

Where do we go from here?!

- Development Containers as a new standard defining Dependencies
  - For instance in a lot of OSS Communities & Repositories

- Ansible Dev Tools for the Go-To Repository / Entrypoint for Tools

- A lot of movement and innovation in terms of different CDE Tools + new IDEs (Zed, Cursor, Fleed etc.)

- Chance to deliver AI Coding Assistants for whole enterprises

- Uncertain: Central Development Environments in a world with faster and faster compute power on mobile computers

Don't miss out on:

Today:

- 14:25 / Ganesh B Nalawade: [Streamlining the Ansible creator experience with the new and improved Ansible Development tools](#)
- 14:50 / Sorin Sbarnea: [Beyond copy-paste: Using Ansible Development Tools for Robust Automation Content](#)

Tuesday:

- 16:50 / Sorin Sbarnea: [From Manual Testing to Continuous Validation: Taking the Quality of Ansible Content to the Next Level](#)

Wednesday:

- 12:15 / Ansible Contributor Summit: [Ansible Development Tools workshop](#)