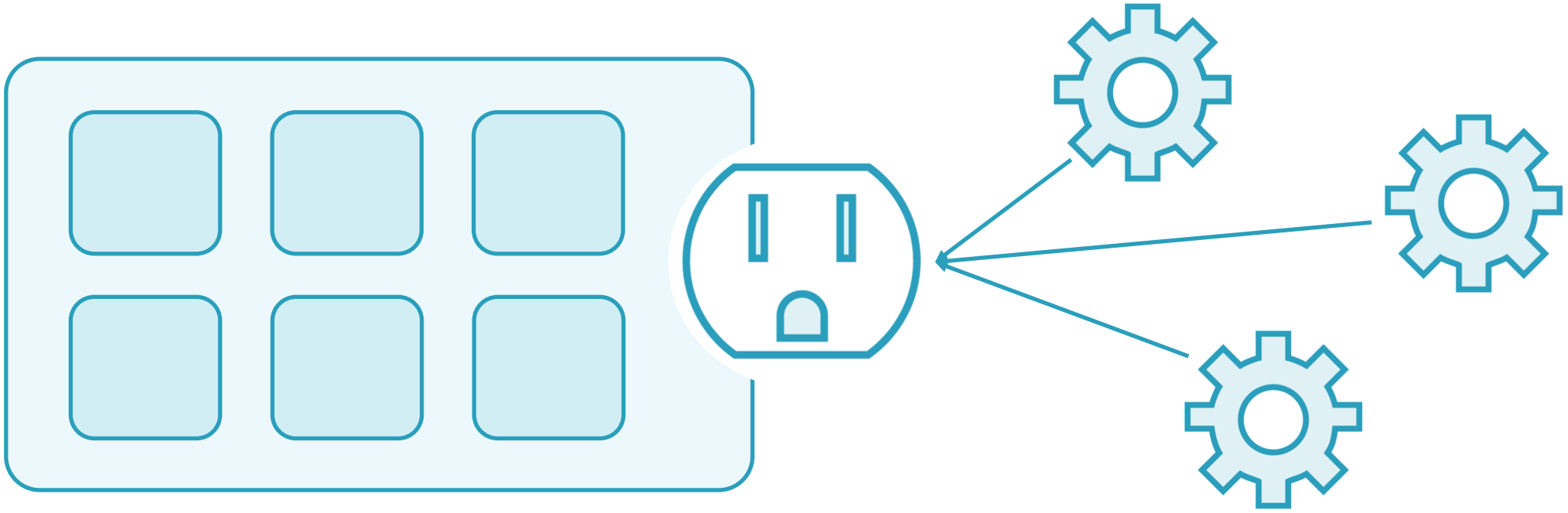
Using and Managing Jenkins Plugins

UNDERSTANDING JENKINS AND THE PLUGIN MODEL

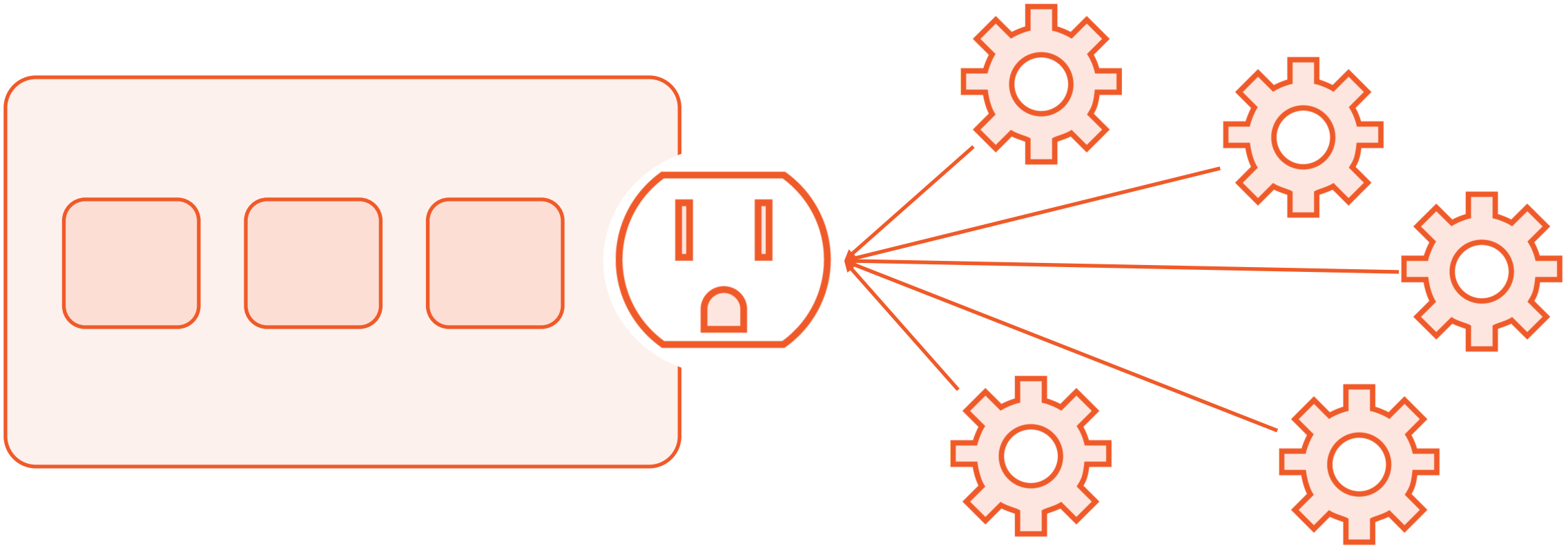
Elton Stoneman
CONSULTANT & TRAINER

@EltonStoneman | blog.sixeyed.com



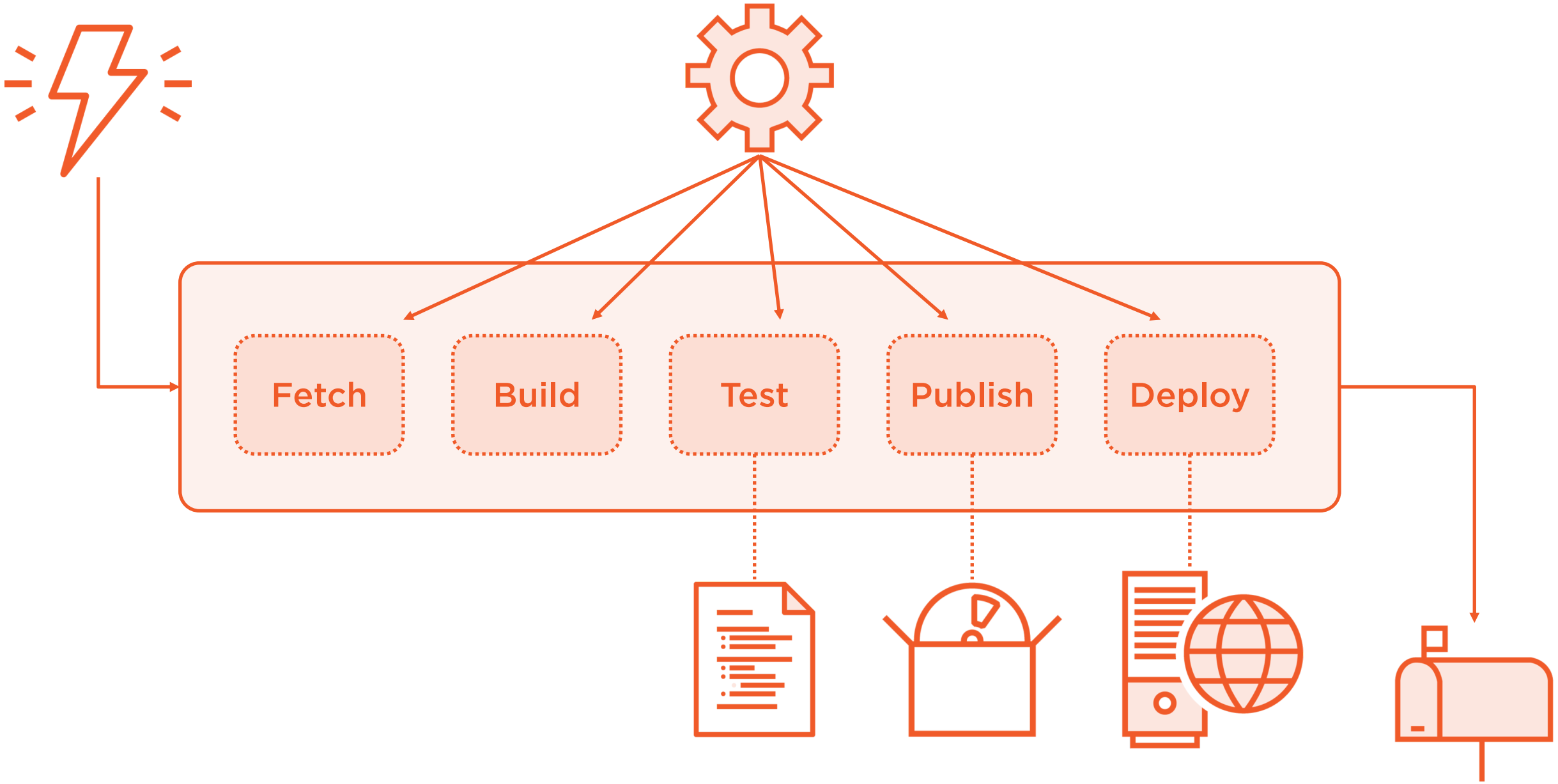
- Text editor
- Filesystem browser
- Terminal
- Job runner

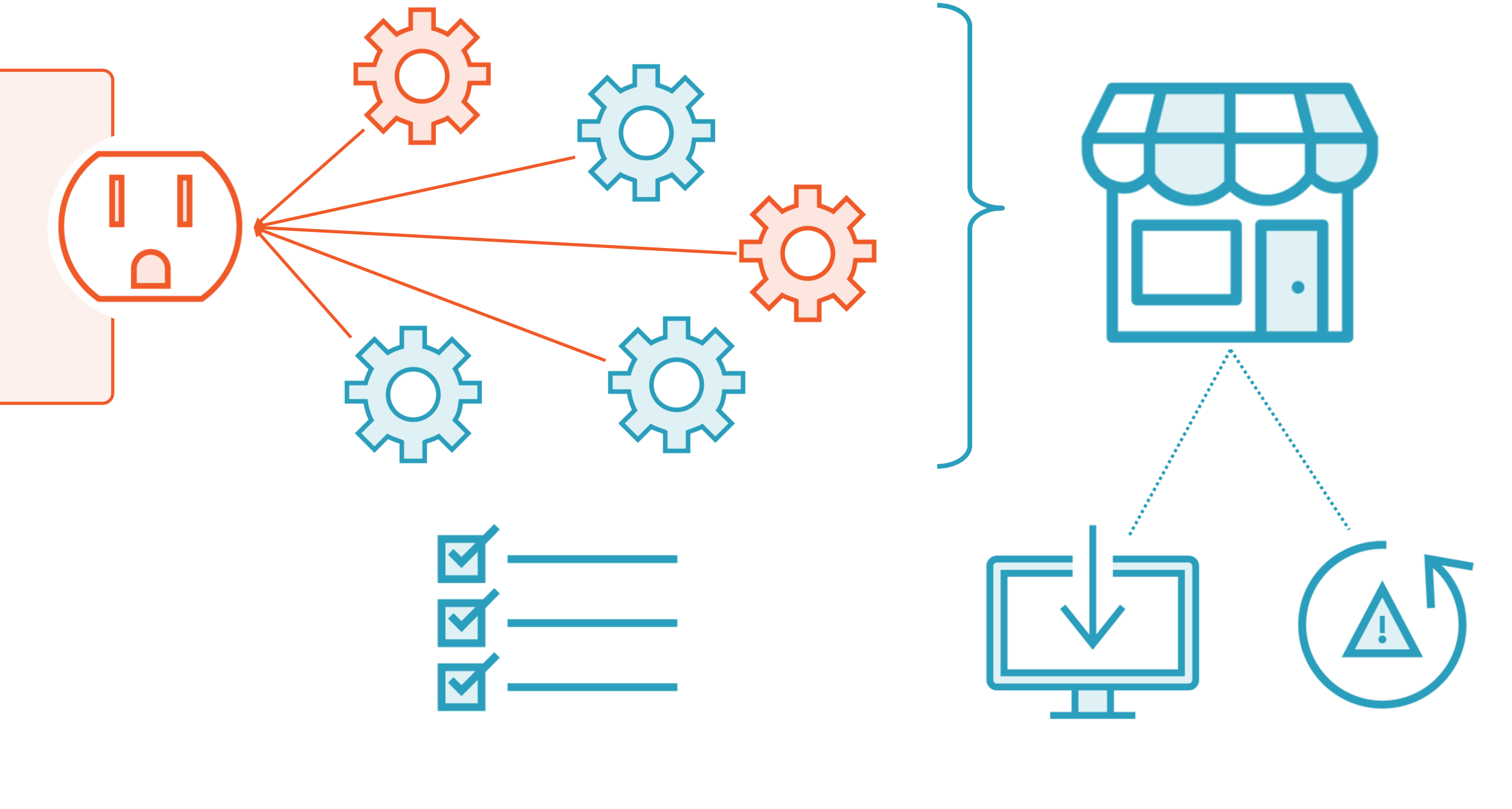
- .NET, Go, Java ...
- Git, Subversion, TFS ...
- Docker, Kubernetes, Azure ...
- Formatting, linting, debugging ...

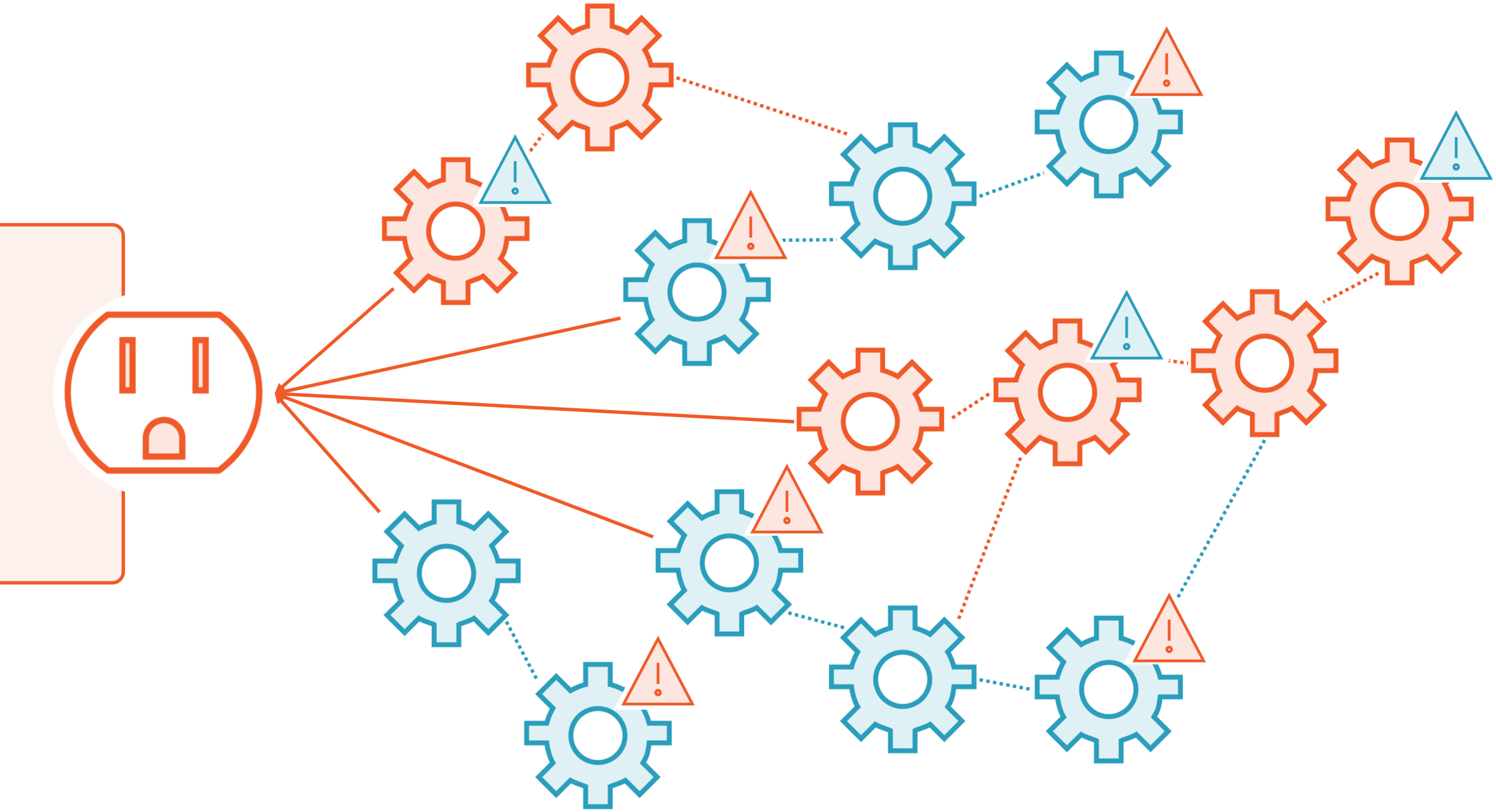


- Web UI
- Job scheduler
- Script executor
- … that's it

- Languages & platforms
- Source control
- Build tools & reporting
- Notifications & publishing


Fetch
Build
Test
Publish
Deploy





Understanding Jenkins and the Plugin Model

Installing and Using Plugins

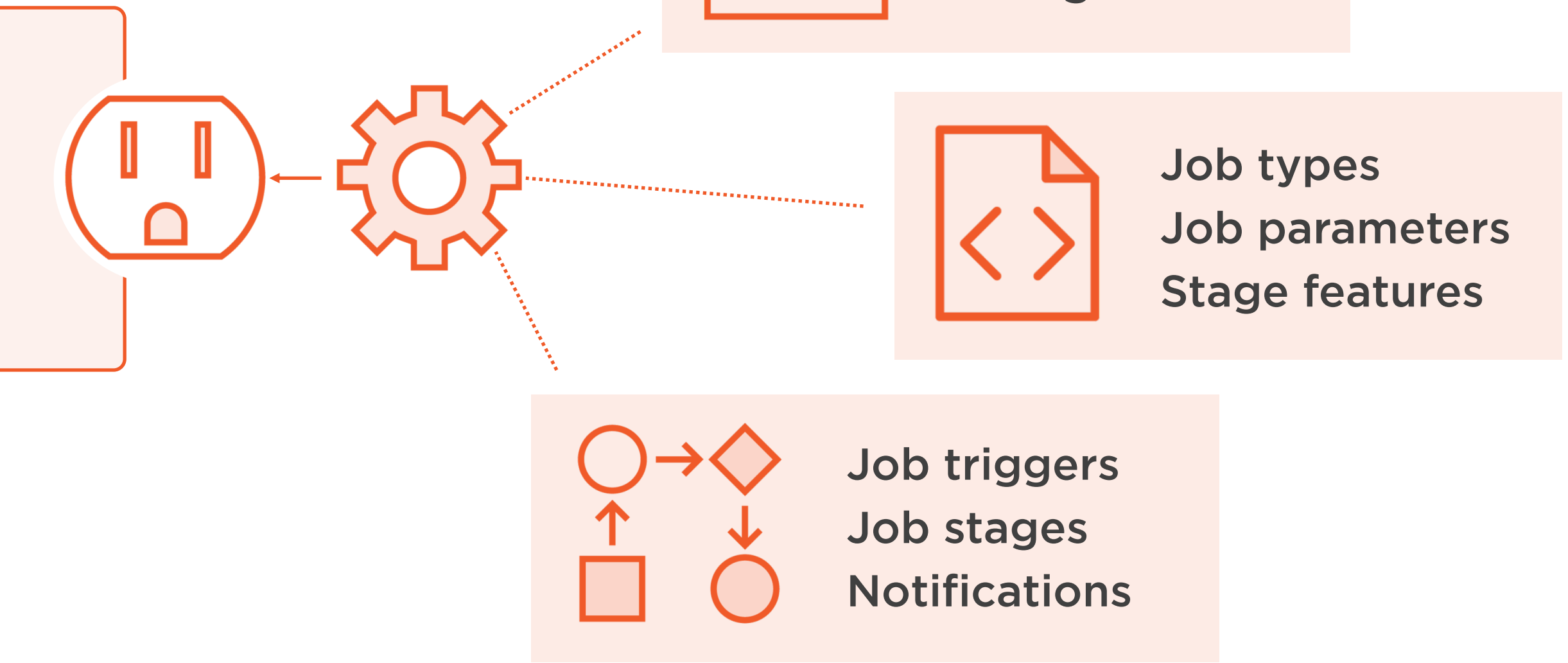
Writing Custom Plugins

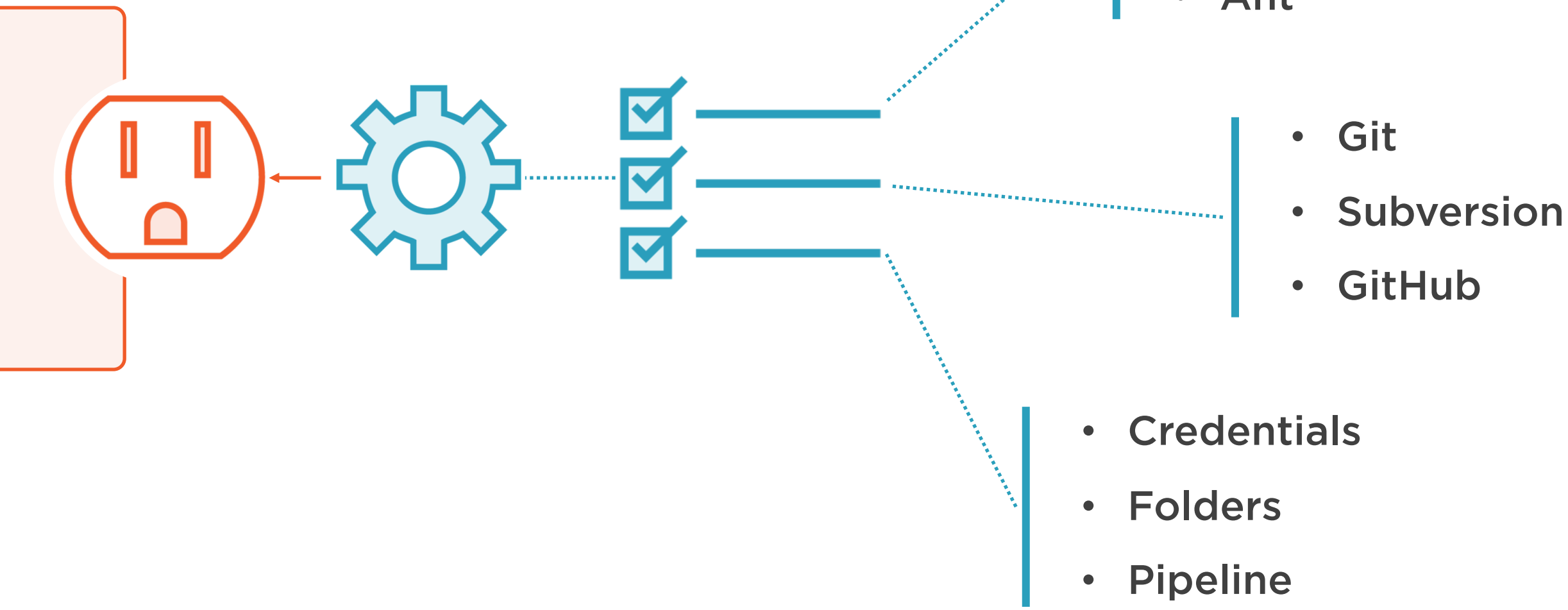
Managing and Upgrading Plugins

Demo

Jenkins - naked

- Fresh install of Jenkins
- No plugins
- Basic capabilities

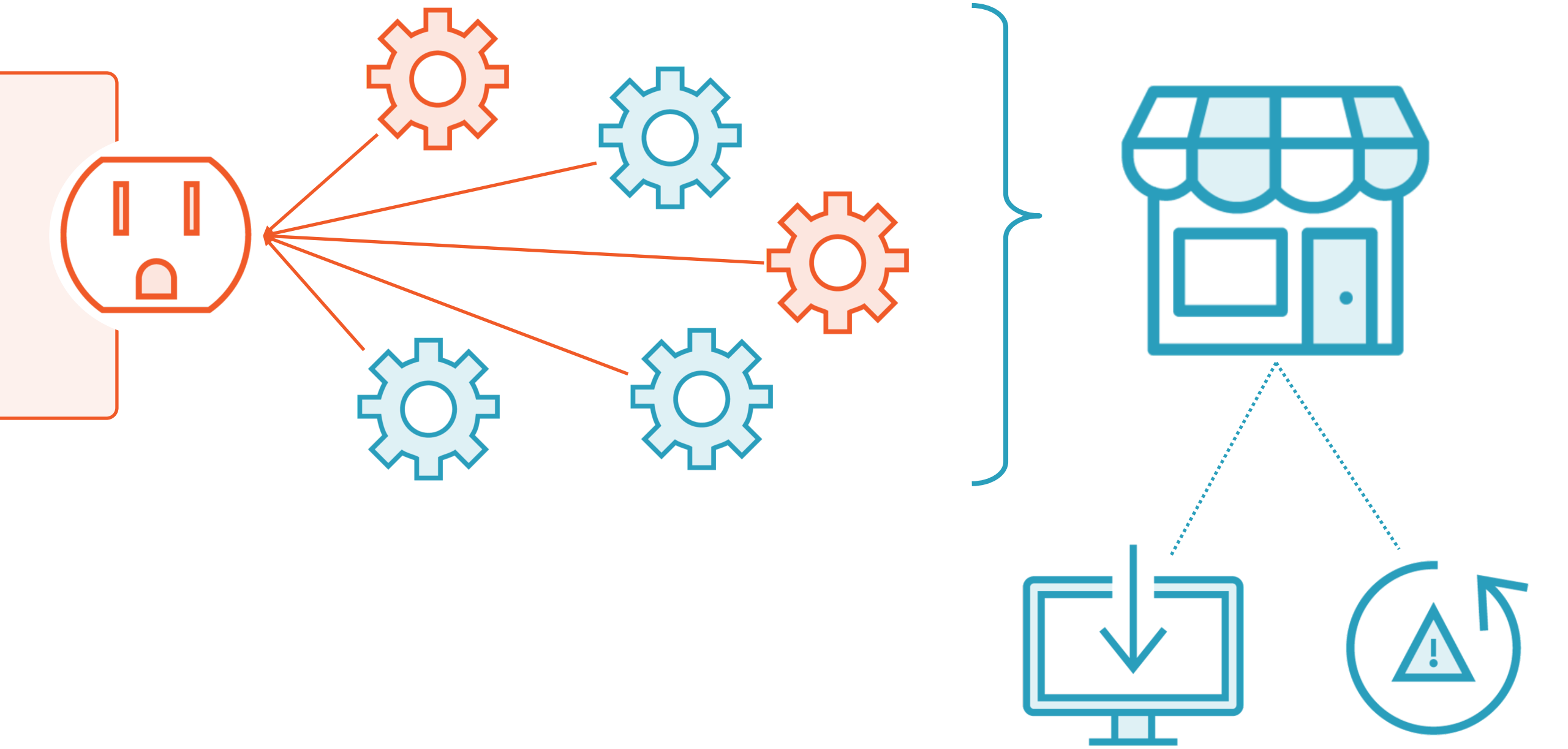

Top-level menu
New resources
Configuration
Job types
Job parameters
Stage features
Job triggers
Job stages
Notifications

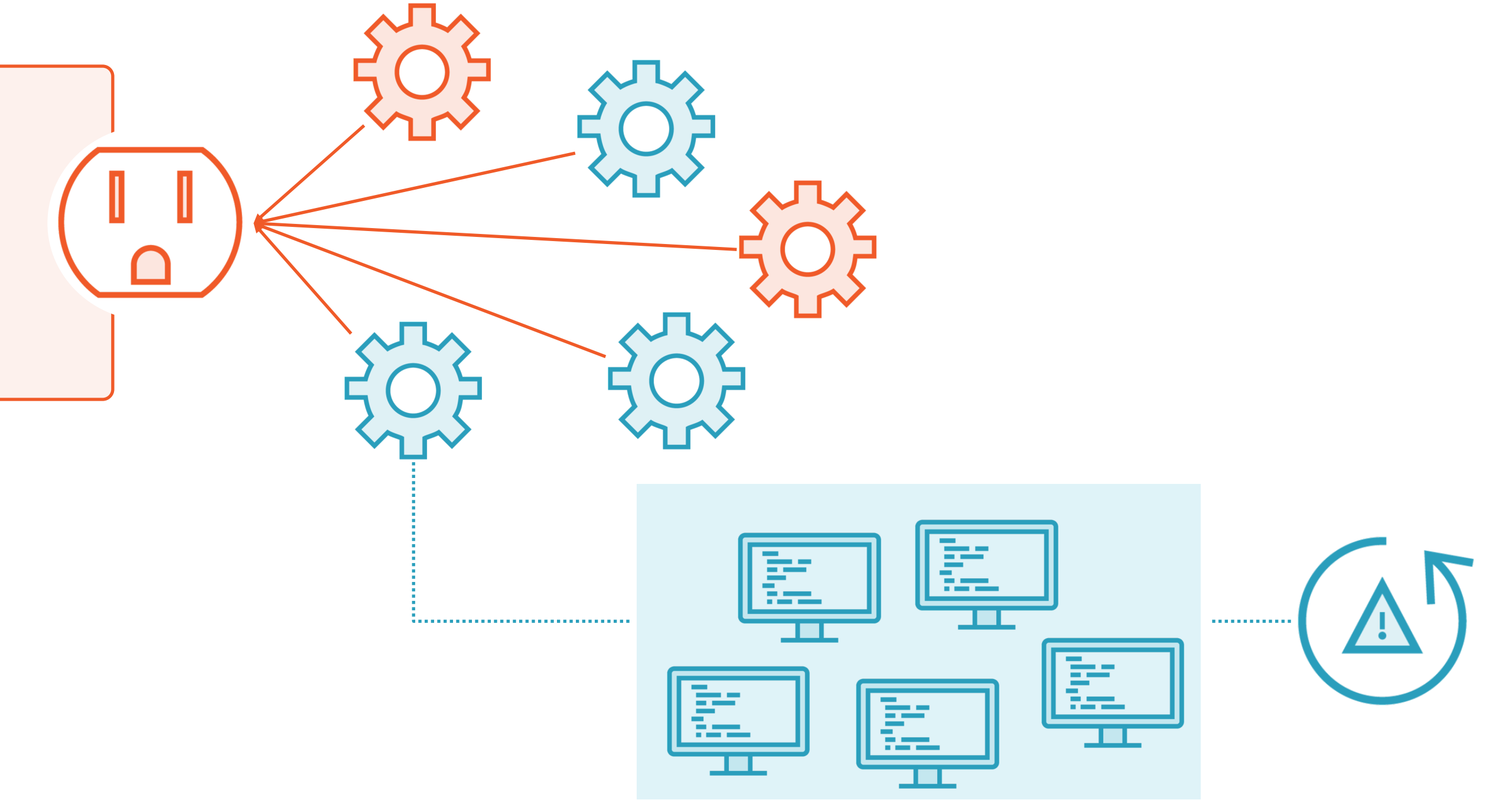

• Maven
• Gradle
• Ant
• Git
• Subversion
• GitHub
• Credentials
• Folders
• Pipeline

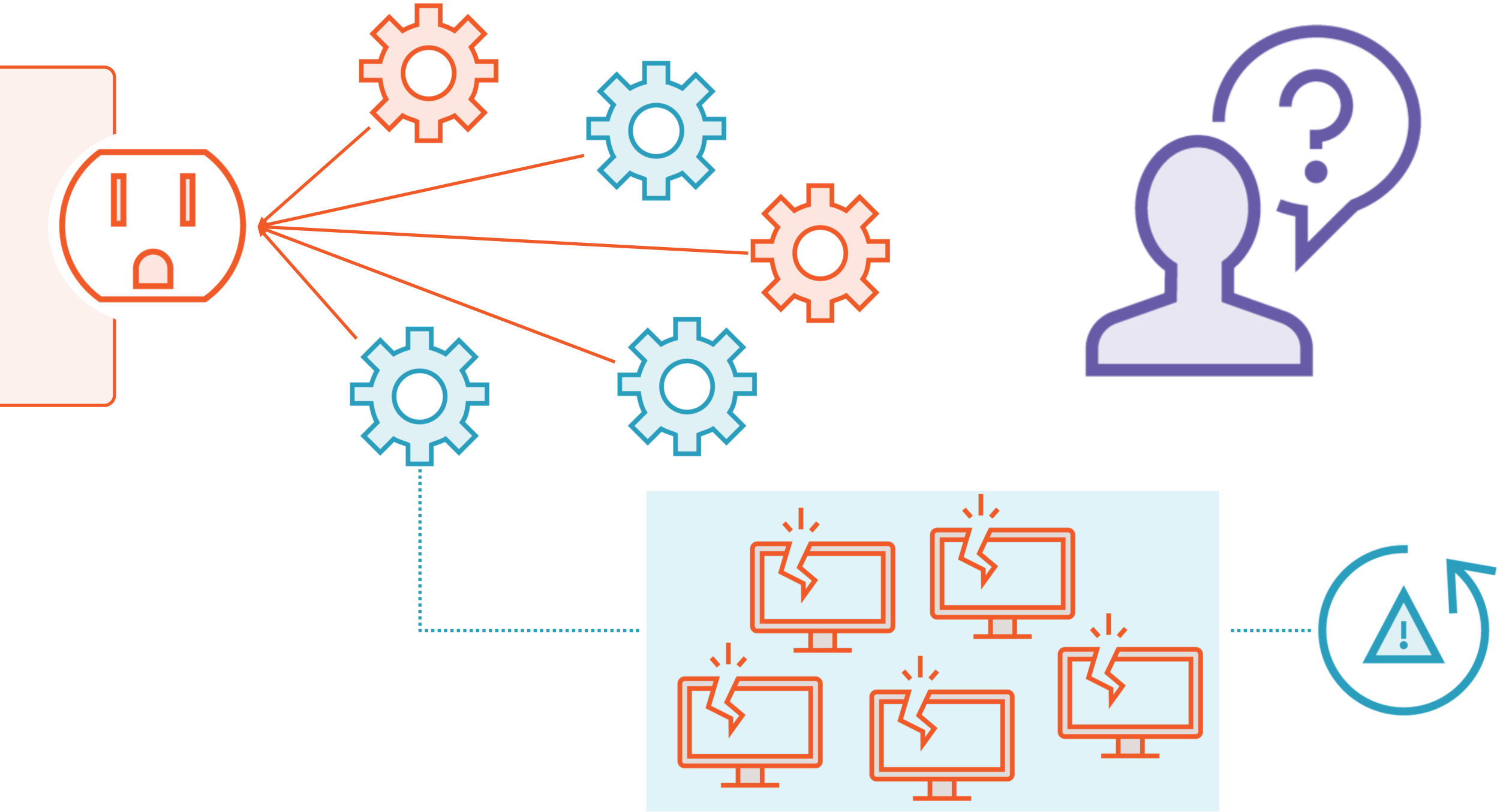
Demo

Jenkins with suggested plugins

- Fresh install of Jenkins
- Install all recommended plugins
- See extension points



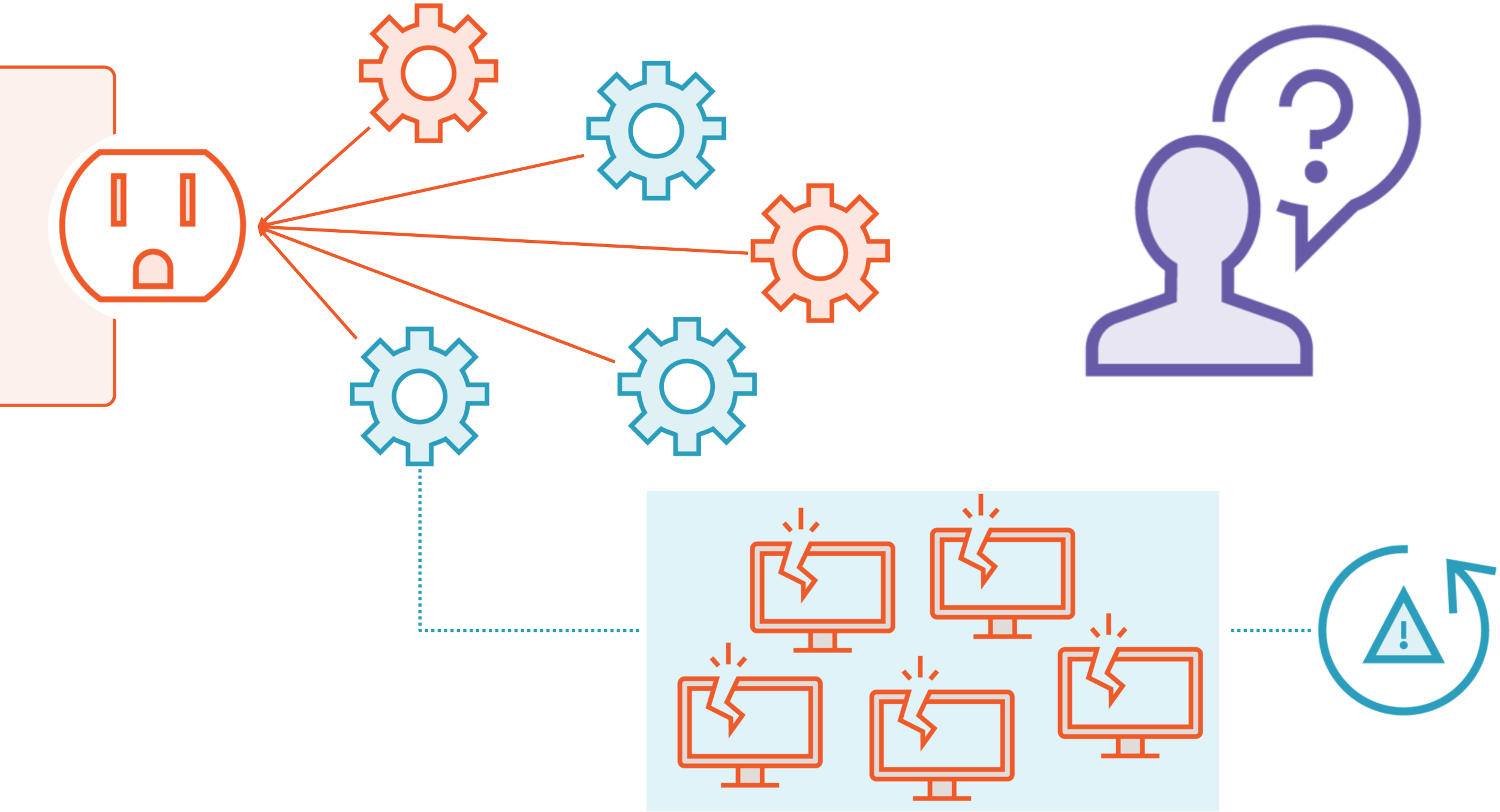


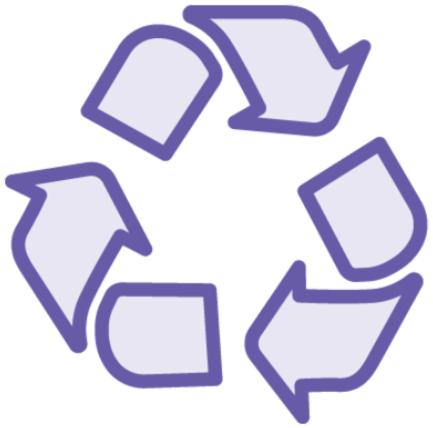


Demo

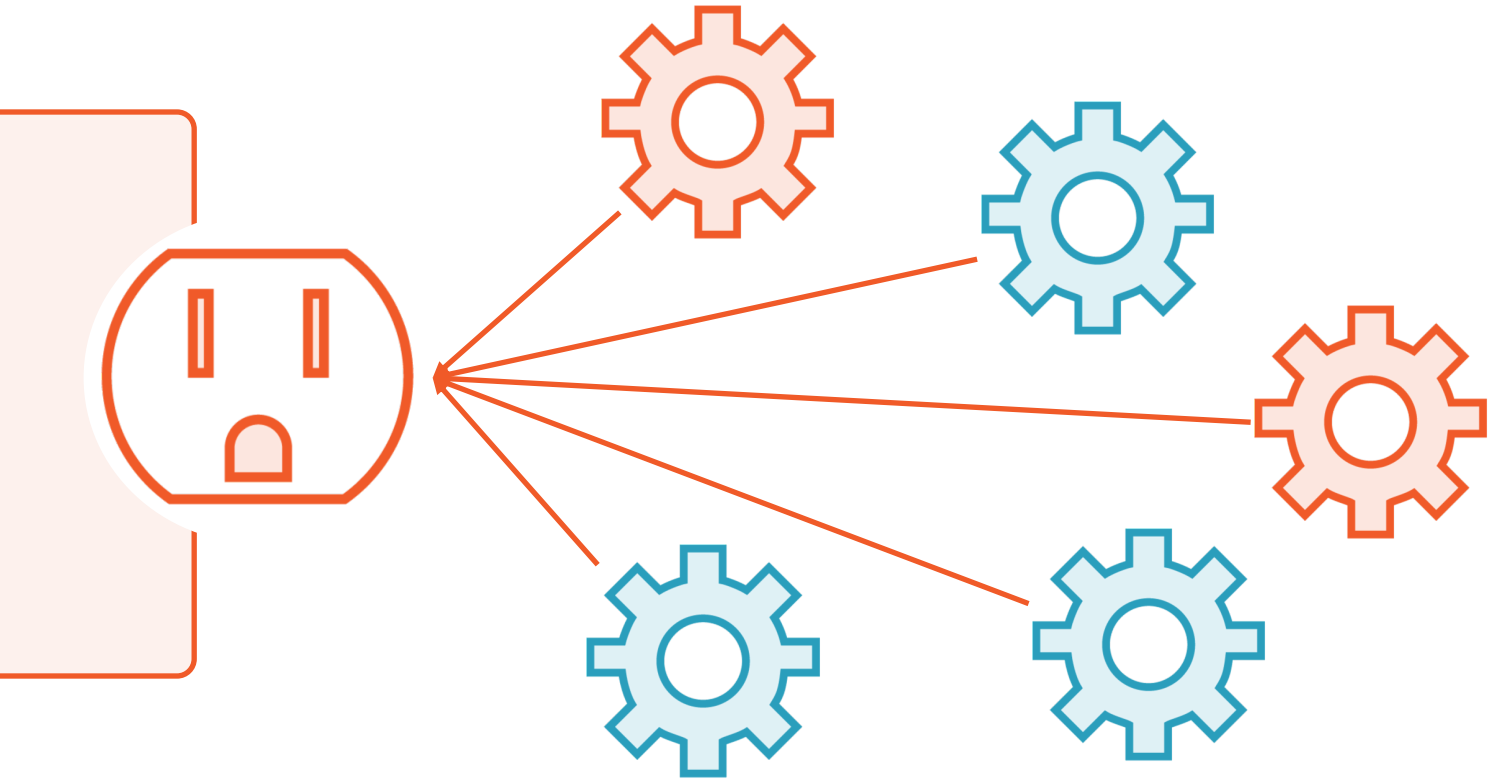
Managing plugin updates

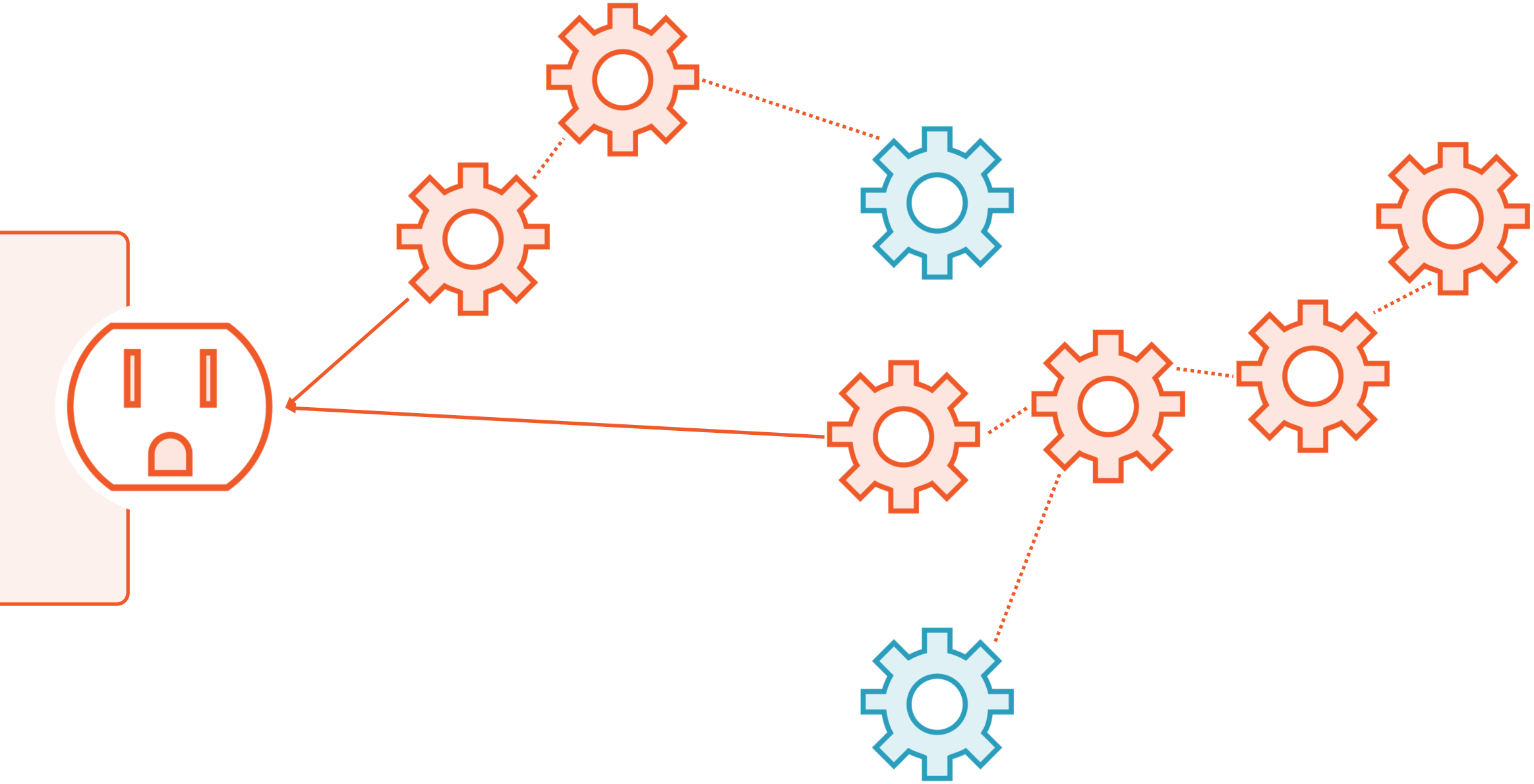
- Existing install of Jenkins
- Outdated plugins
- Notification and management



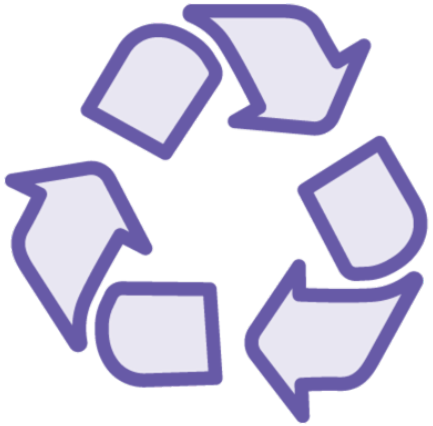


Reduce surface area

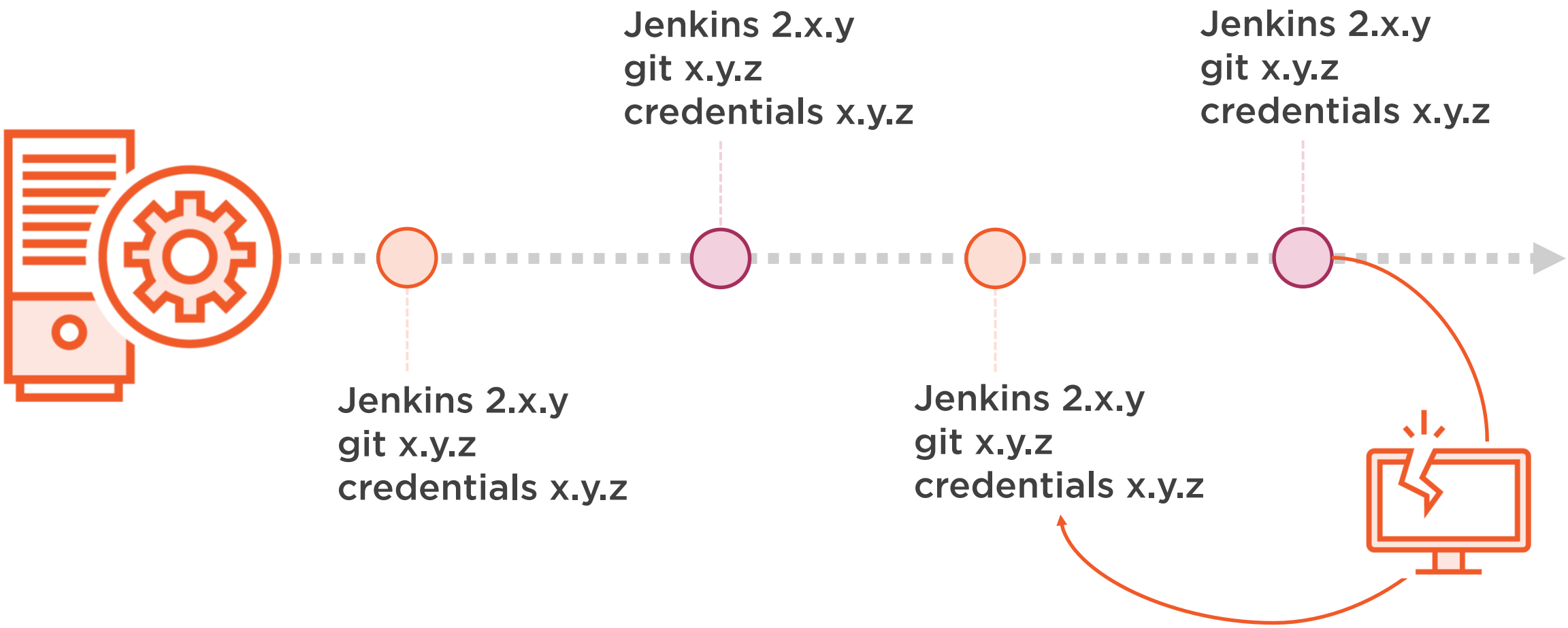




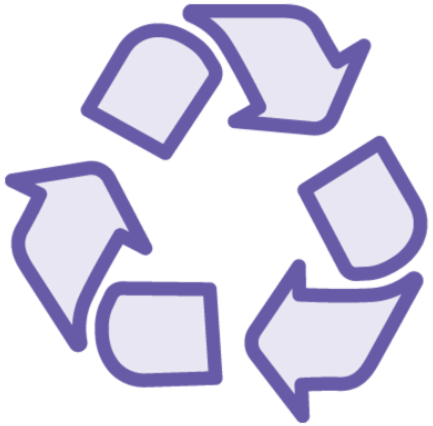
Reduce surface area



Repeatable deployment

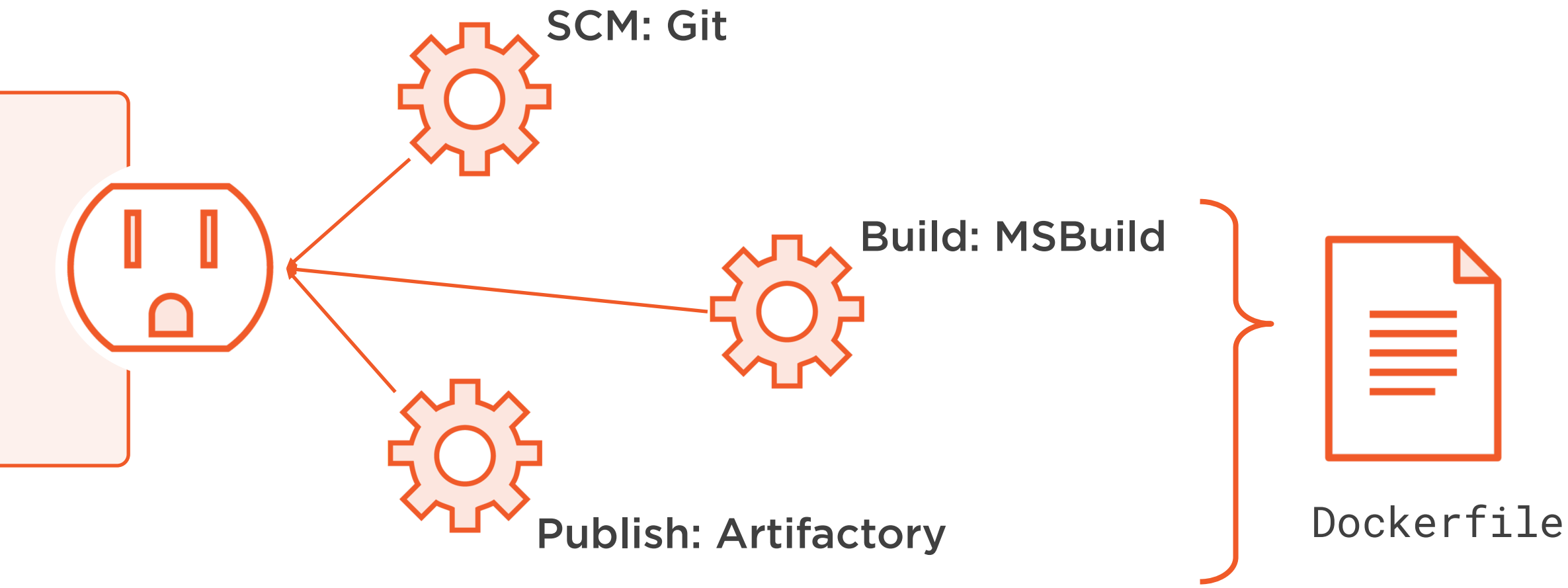

Jenkins 2.x.y
git x.y.z
credentials x.y.z
Jenkins 2.x.y
git x.y.z
credentials x.y.z
Jenkins 2.x.y
git x.y.z
credentials x.y.z
Jenkins 2.x.y
git x.y.z
credentials x.y.z

Reduce surface area



Repeatable deployment

Minimize requirements


SCM: Git
Build: MSBuild
Publish: Artifactory
Dockerfile

Modernizing .NET Framework Apps with Docker

by Elton Stoneman

Docker can help you bring your existing applications into the modern world. This course teaches you how to run full .NET applications in Windows containers, modernize the architecture, and deploy to the cloud.

Start Course | Bookmark | Add to Channel | Download Course

Table of contents | **Description** | Transcript | Exercise files | Discussion | Learning Check | Related Courses

Docker isn't just for greenfield microservices applications, you can take full .NET Framework applications and run them in containers with no code changes. That's a great starting point for modernizing the architecture and moving to the cloud. In this course, Modernizing .NET Framework Apps with Docker, you'll learn how to efficiently run .NET applications and create a more modern architecture utilizing Docker. First, you'll discover how to package and run .NET apps in Docker containers on Windows. Then, you'll explore how to evolve the application architecture by breaking features out into separate containers. Finally, you'll delve into taking your modernized app to production on Azure. By the end of the course, you'll understand how Docker works on Windows and what Docker can do for your existing .NET landscape. Software required: Docker.

Course author

Elton Stoneman

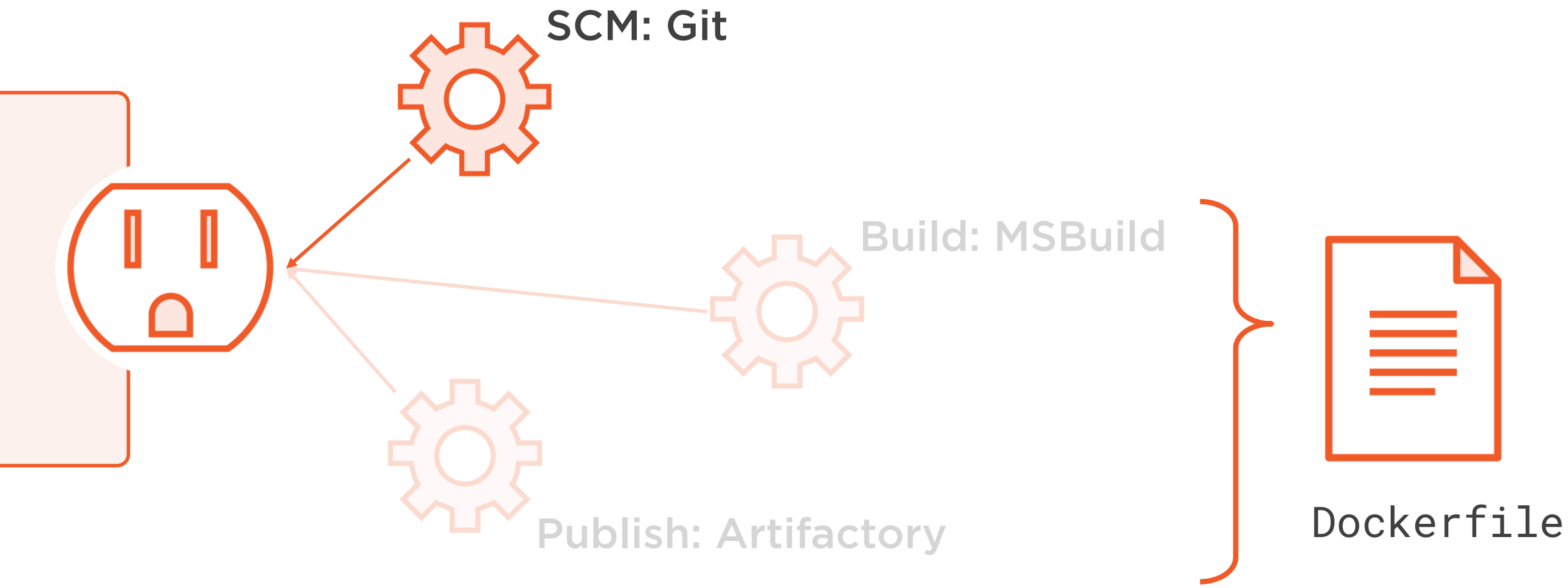
Elton is a 10-time Microsoft MVP, author, trainer and speaker. He spent most of his career as a consultant working in Microsoft technologies, architecting and delivering complex solutions for...

Course info

Level	Intermediate
Rating	★★★★½ (70)
My rating	★★★★★
Duration	3h 42m
Released	28 Dec 2017

Share course

https://is.gd/izukep


SCM: Git
Build: MSBuild
Publish: Artifactory
Dockerfile

Demo

Jenkins with minimal plugins

- Fresh install of Jenkins
- A curated plugin list
- Exploring new features

Summary

Jenkins is an automation server

- Core features define & run jobs
- All useful functionality in plugins

Plugins extend the whole stack

- Composite Web UI
- Job definition and runtime

Dependencies can sprawl

- Minimize plugin requirements
- Reduce upgrade risk

Up Next:
Installing and Using Plugins