

Ansible + Rundeck = śpij adminie, śpij

Maciej Lasyk

DevopsKRR meetup #9

Kraków 2016-04-14



Dołącz do projektu Fedora!

<http://fedoraproject.org/en/join-fedora>



**KRAKÓW, POLAND**

**AUGUST 2 - 5**

**FLOCK 2016**

<https://fedoramagazine.org/flock-2016-krakow-poland/>

# Ansible & Rundeck?

- monolityczne instalacje (np. CI)
- a gdyby tak developerzy sami zarządzali CI?
- i do tego są potrzebne narzędzia
- ansible jest prosty
- rundeck jest jeszcze prostszy

# Ansible & Rundeck?

- monolityczne instalacje (np. CI)
- a gdyby tak developerzy sami zarządzali CI?
- i do tego są potrzebne narzędzia
- ansible jest prosty
- rundeck jest jeszcze prostszy

# Ansible & Rundeck?

- monolityczne instalacje (np. CI)
- a gdyby tak developerzy sami zarządzali CI?
- i do tego są potrzebne narzędzia
- ansible jest prosty
- rundeck jest jeszcze prostszy

# Ansible & Rundeck?

- monolityczne instalacje (np. CI)
- a gdyby tak developerzy sami zarządzali CI?
- i do tego są potrzebne narzędzia
- **ansible jest prosty**
- rundeck jest jeszcze prostszy

# Ansible & Rundeck?

- monolityczne instalacje (np. CI)
- a gdyby tak developerzy sami zarządzali CI?
- i do tego są potrzebne narzędzia
- ansible jest prosty
- rundeck jest jeszcze prostszy



# Model współpracy dev ↔ ops



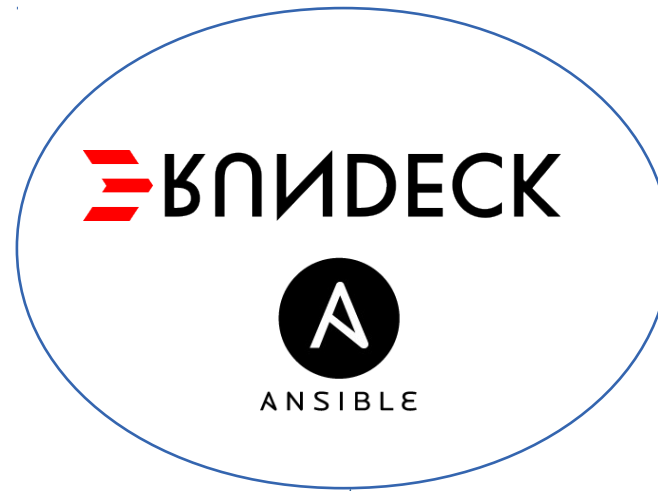
# Model współpracy dev ↔ ops



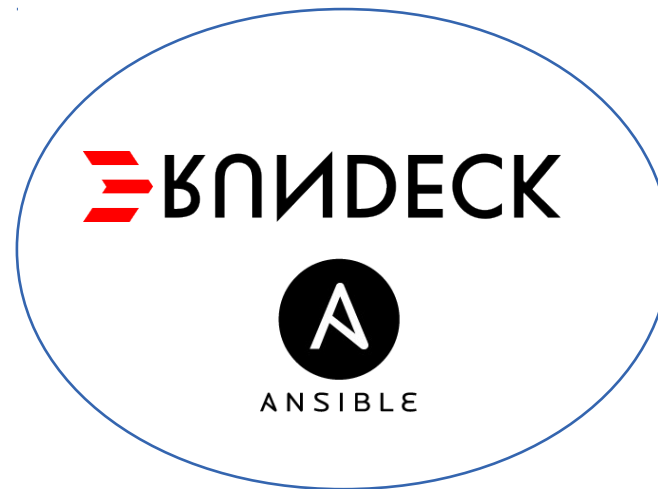
# Model współpracy dev ↔ ops



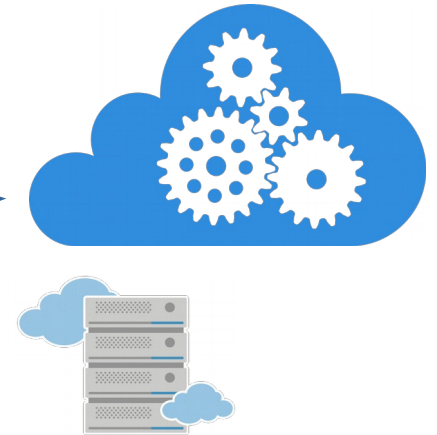
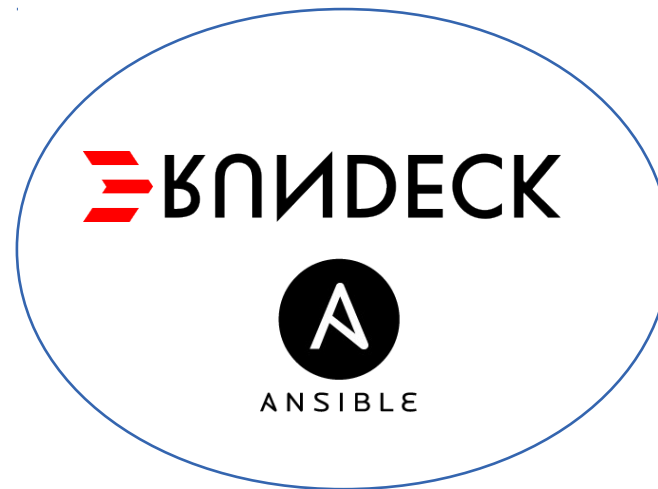
# Model współpracy dev ↔ ops



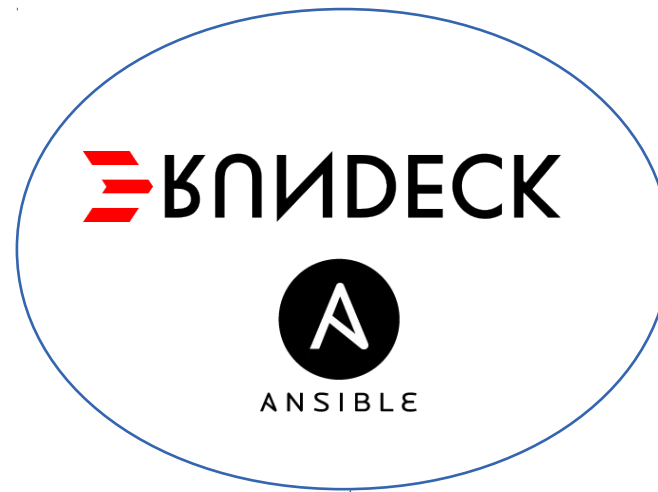
# Model współpracy dev ↔ ops



# Model współpracy dev ↔ ops



# Model współpracy dev ↔ ops



# Model współpracy dev ↔ ops



 KUNDECK

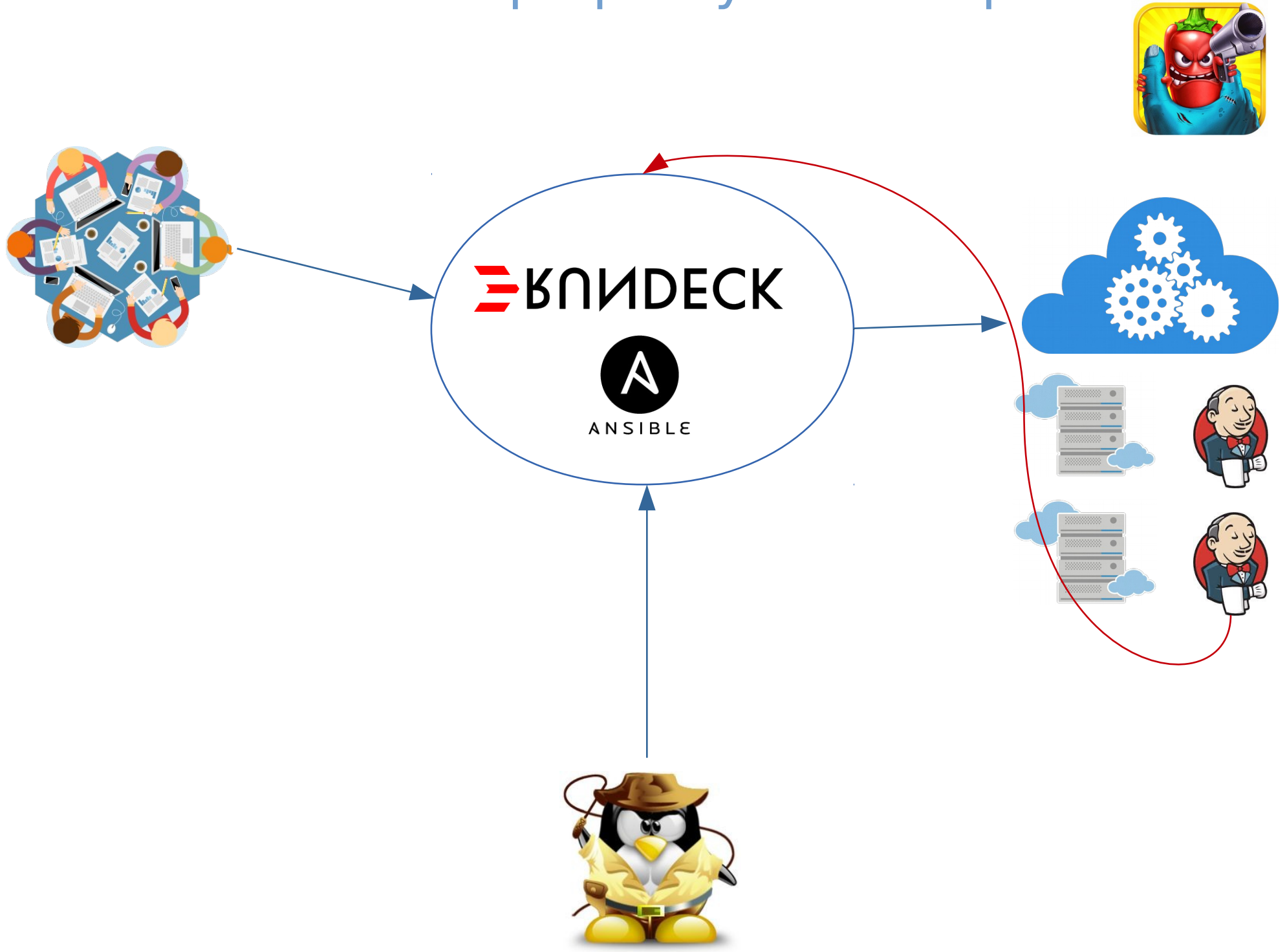


ANSIBLE

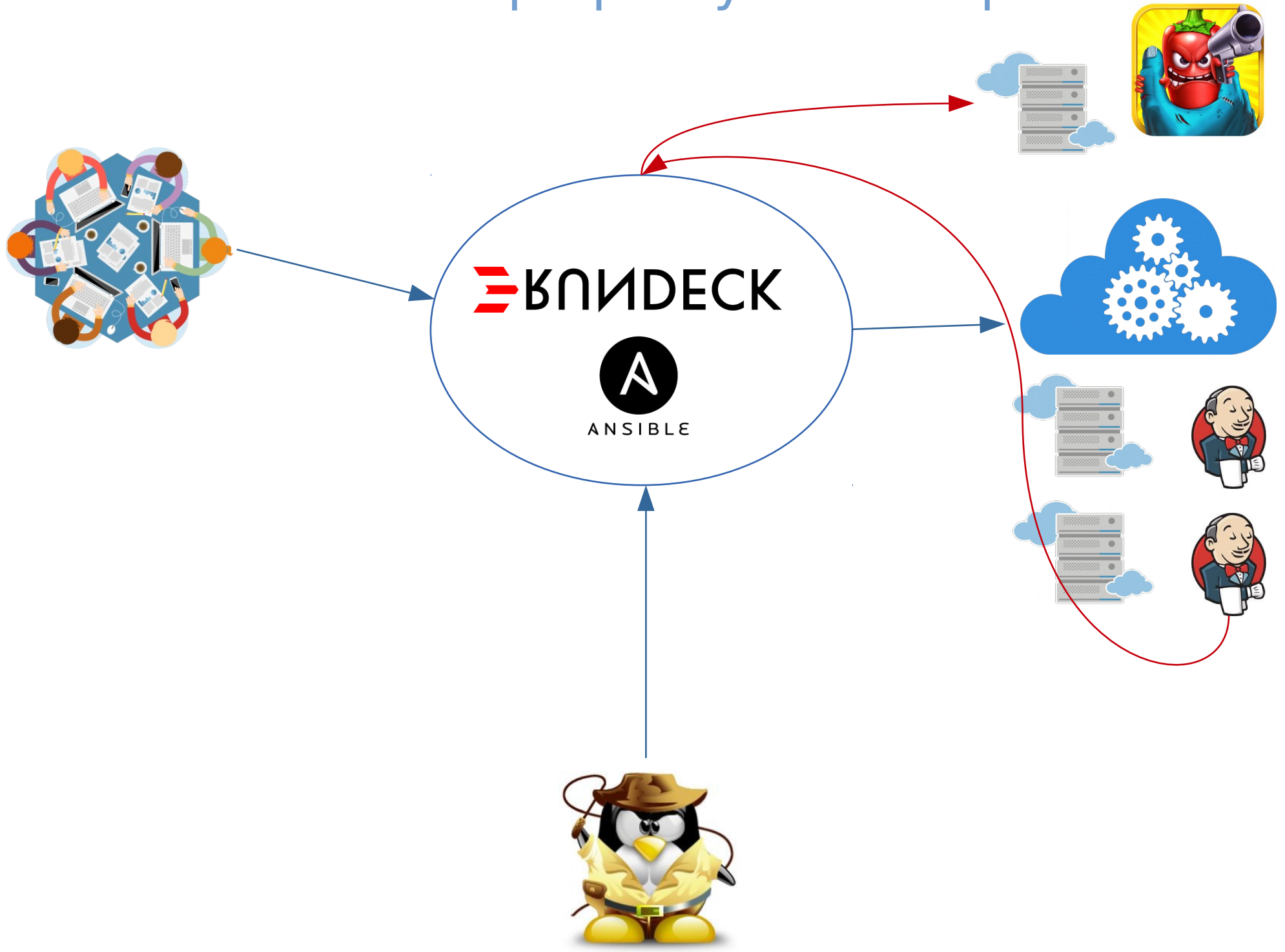




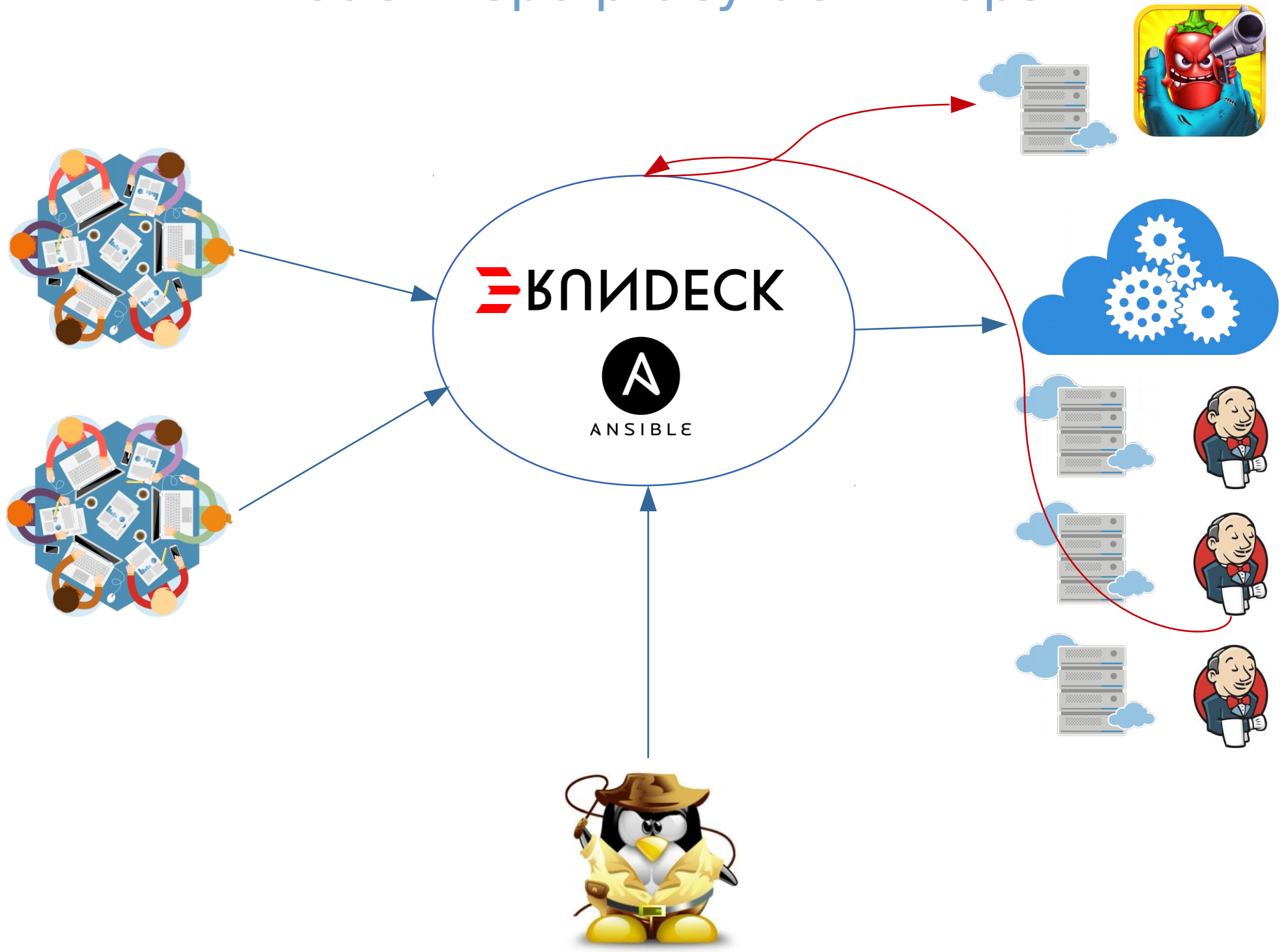
# Model współpracy dev ↔ ops



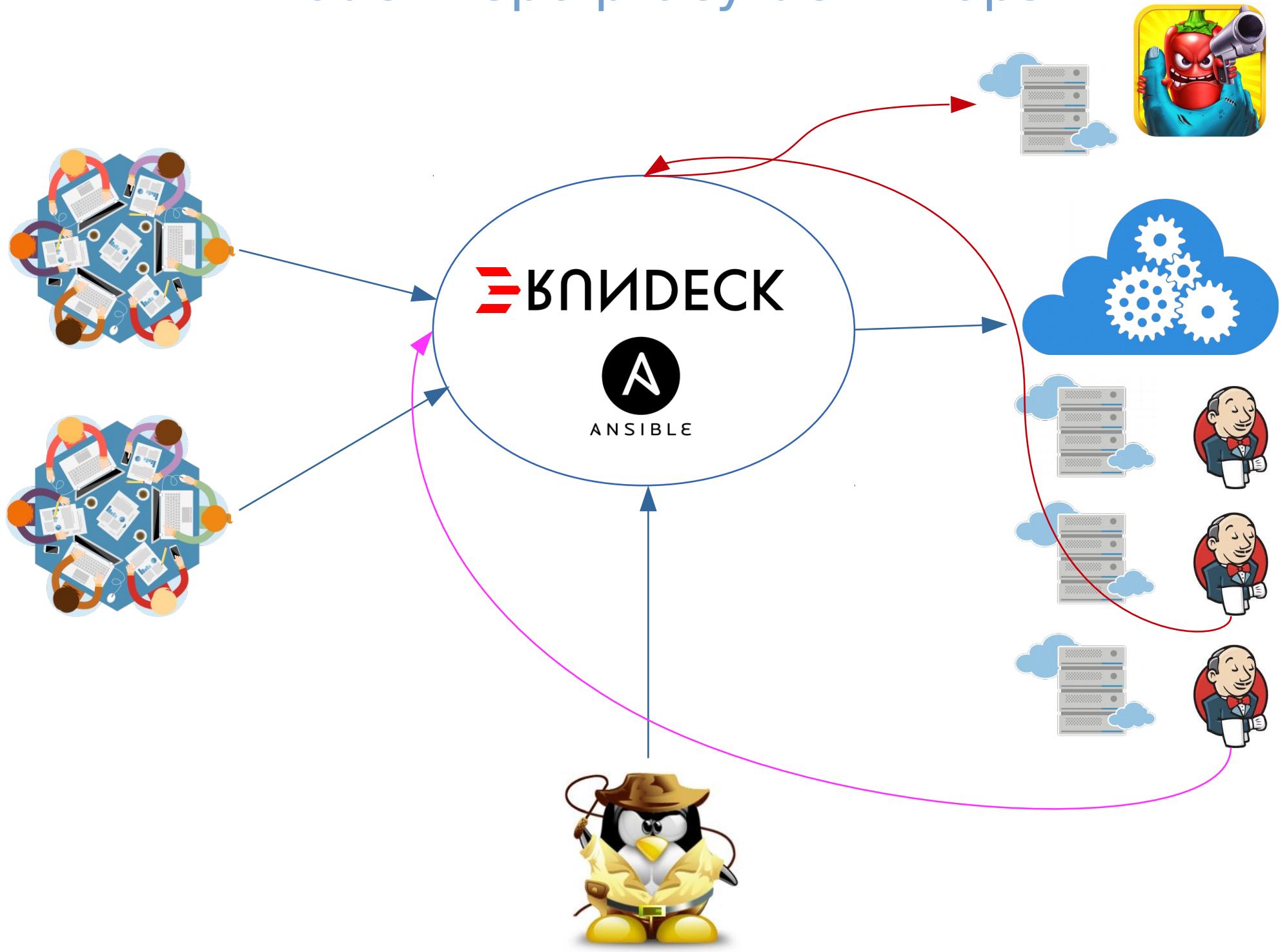
# Model współpracy dev ↔ ops



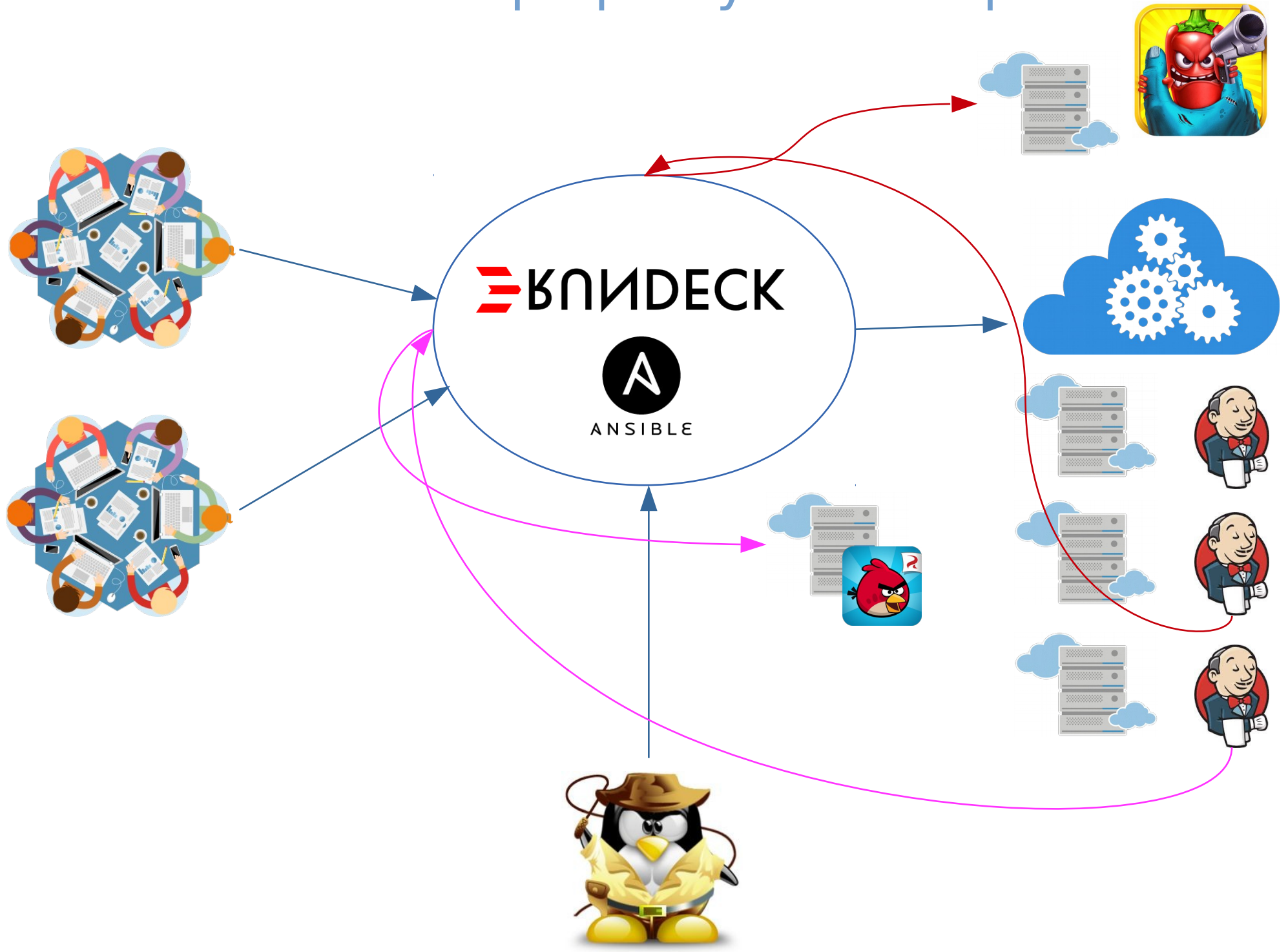
# Model współpracy dev ↔ ops



# Model współpracy dev ↔ ops

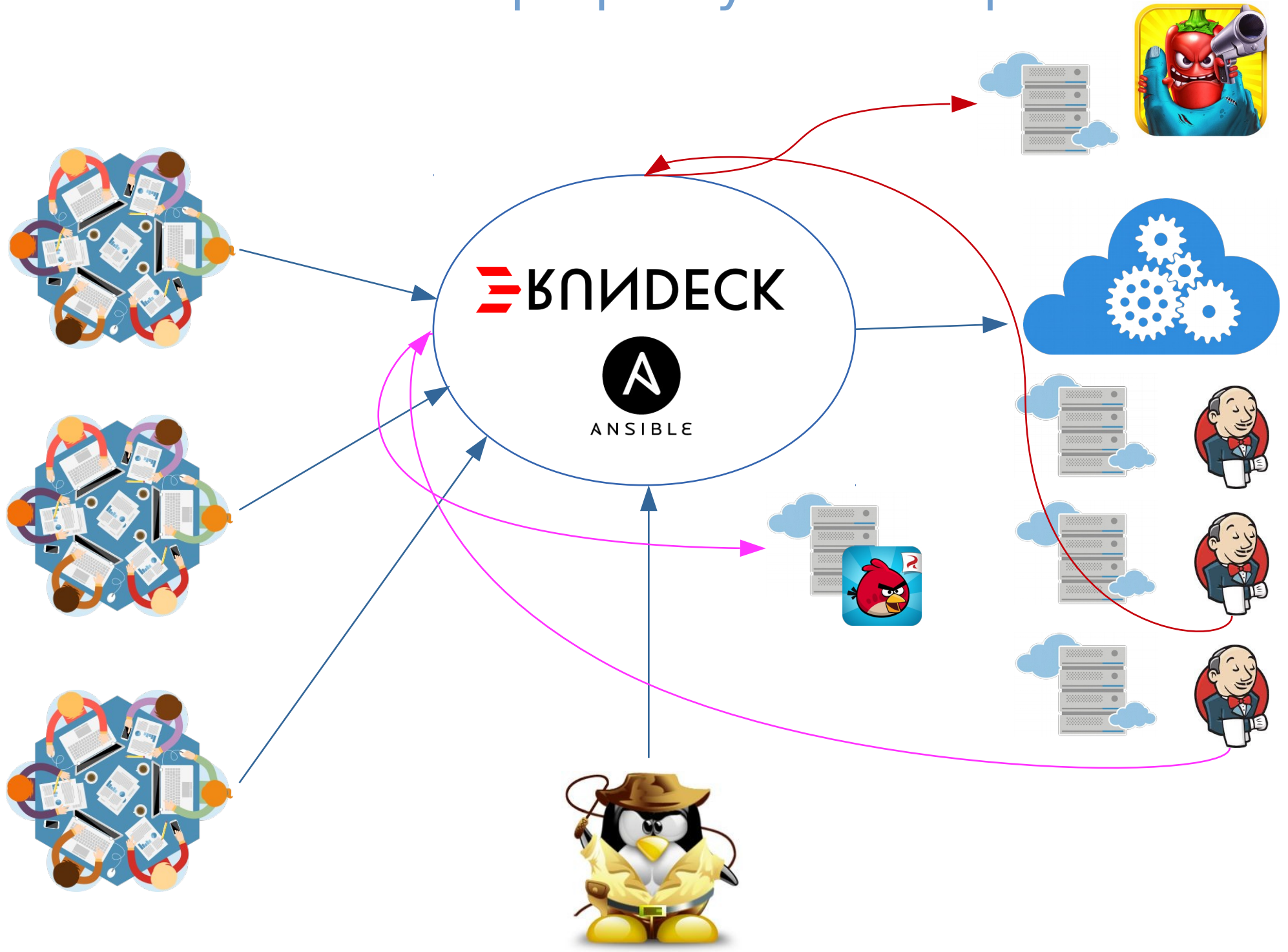


# Model współpracy dev ↔ ops

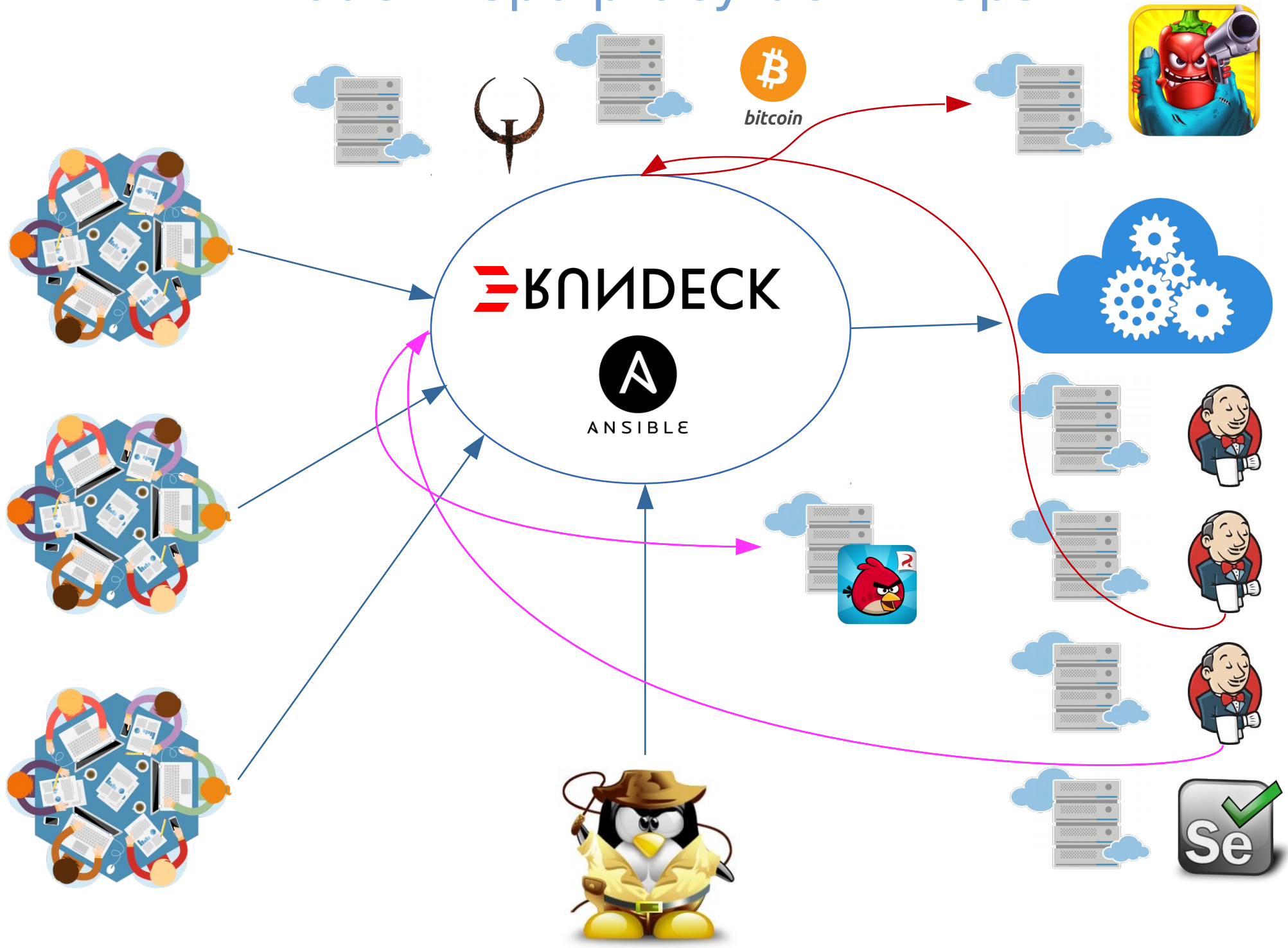


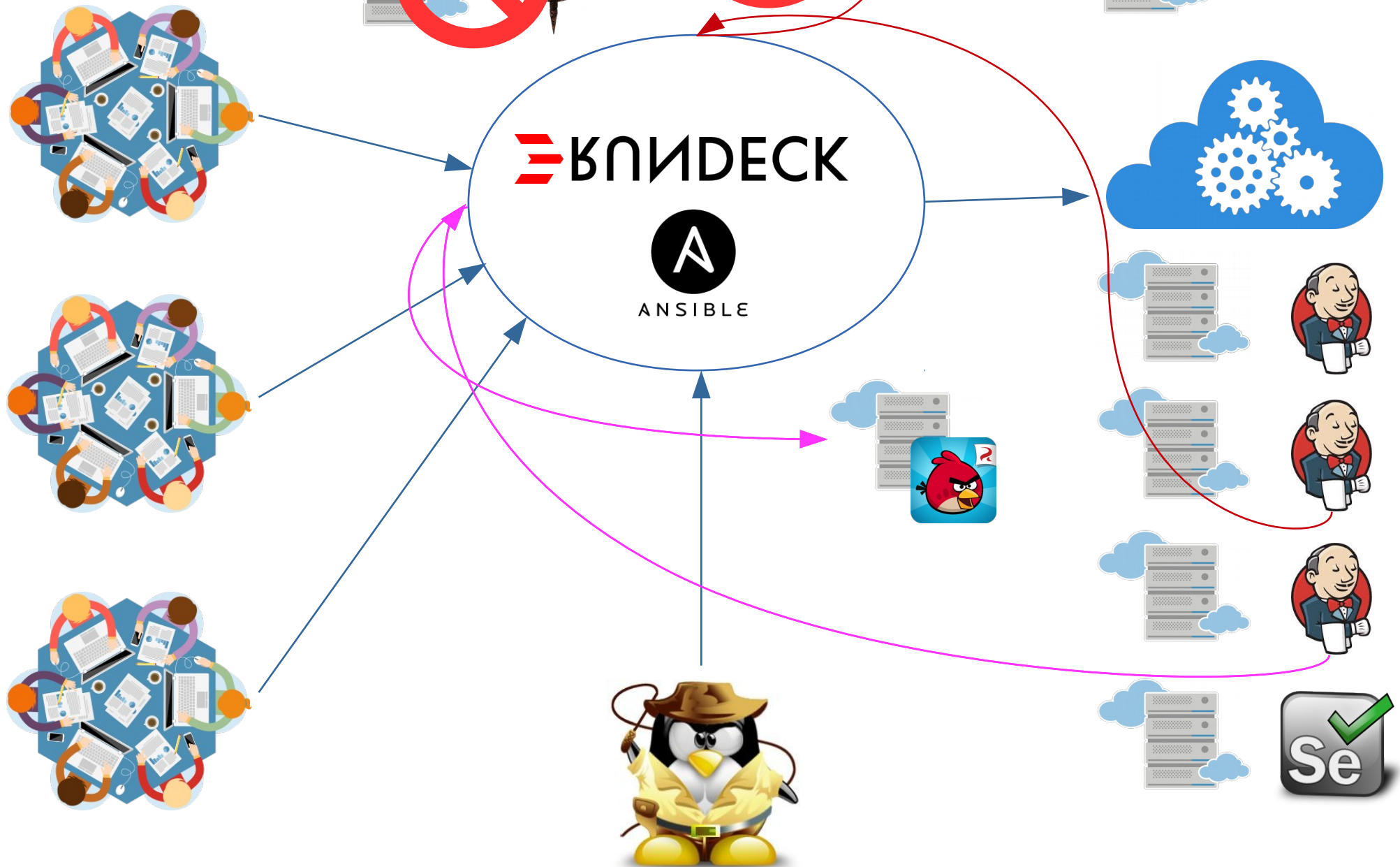


# Model współpracy dev ↔ ops



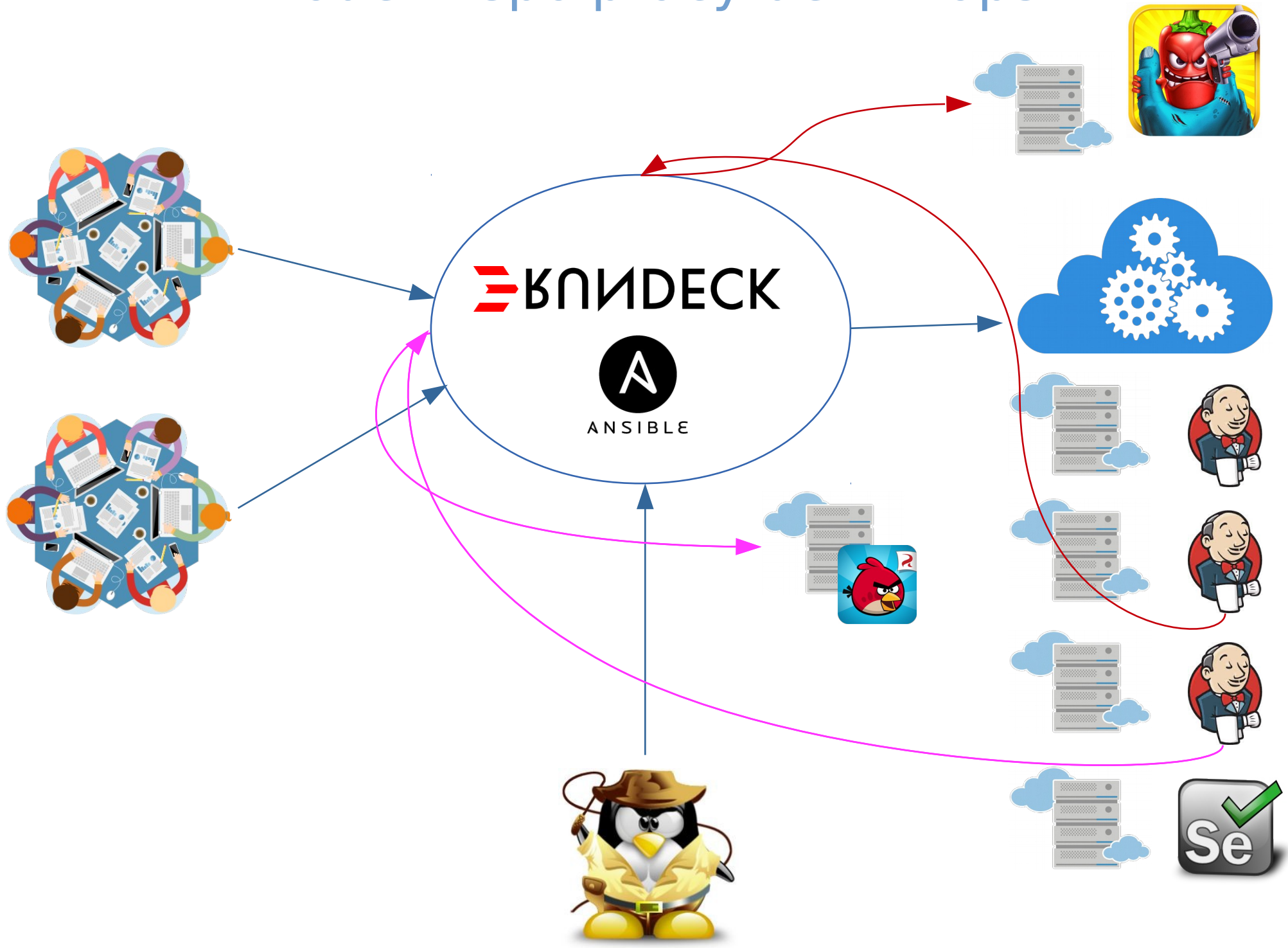
# Model współpracy dev ↔ ops







# Model współpracy dev ↔ ops



# Czym jest Ansible?

- prosty orkiestrator
- tworzymy playbooks
- posiada inventory hostów
- używa ssh
- pod spodem Python
- przykładowy playbook I inventory

# Czym jest Ansible?

- prosty orkiestrator
- tworzymy playbooks
- posiada inventory hostów
- używa ssh
- pod spodem Python
- przykładowy playbook i inventory

# Czym jest Ansible?

- prosty orkiestrator
- tworzymy playbooks
- posiada inventory hostów
- używa ssh
- pod spodem Python
- przykładowy playbook I inventory

# Czym jest Ansible?

- prosty orkiestrator
- tworzymy playbooks
- posiada inventory hostów
- używa ssh**
- pod spodem Python
- przykładowy playbook i inventory

# Czym jest Ansible?

- prosty orkiestrator
- tworzymy playbooks
- posiada inventory hostów
- używa ssh
- **pod spodem Python**
- przykładowy playbook I inventory

# Czym jest Ansible?

- prosty orkiestrator
- tworzymy playbooks
- posiada inventory hostów
- używa ssh
- pod spodem Python
- przykładowy playbook I inventory

# Czym jest Rundeck?

- “Turn your operations procedures into self-service jobs.”
- “Safely give others the control and visibility they need.”
- "...it's the swiss army knife for ops."
- biblioteka procedur
- scheduler
- łatwa integracja z systemami do incydentów
- po CI / buildzie nadchodzi deployment
- GUI & API



# Czym jest Rundeck?

- “Turn your operations procedures into self-service jobs.”
- “Safely give others the control and visibility they need.”
- "...it's the swiss army knife for ops."
- biblioteka procedur
- scheduler
- łatwa integracja z systemami do incydentów
- po CI / buildzie nadchodzi deployment
- GUI & API

# Czym jest Rundeck?

- “Turn your operations procedures into self-service jobs.”
- “Safely give others the control and visibility they need.”
- "...it's the swiss army knife for ops."
- biblioteka procedur
- scheduler
- łatwa integracja z systemami do incydentów
- po CI / buildzie nadchodzi deployment
- GUI & API

# Czym jest Rundeck?

- “Turn your operations procedures into self-service jobs.”
- “Safely give others the control and visibility they need.”
- "...it's the swiss army knife for ops."
- **biblioteka procedur**
- scheduler
- łatwa integracja z systemami do incydentów
- po CI / buildzie nadchodzi deployment
- GUI & API

# Czym jest Rundeck?

- “Turn your operations procedures into self-service jobs.”
- “Safely give others the control and visibility they need.”
- "...it's the swiss army knife for ops."
- biblioteka procedur
- **scheduler**
- łatwa integracja z systemami do incydentów
- po CI / buildzie nadchodzi deployment
- GUI & API

# Czym jest Rundeck?

- “Turn your operations procedures into self-service jobs.”
- “Safely give others the control and visibility they need.”
- "...it's the swiss army knife for ops."
- biblioteka procedur
- scheduler
- łatwa integracja z systemami do incydentów
- po CI / buildzie nadchodzi deployment
- GUI & API

# Czym jest Rundeck?

- “Turn your operations procedures into self-service jobs.”
- “Safely give others the control and visibility they need.”
- "...it's the swiss army knife for ops."
- biblioteka procedur
- scheduler
- łatwa integracja z systemami do incydentów
- po CI / buildzie nadchodzi deployment
- GUI & API

# Czym jest Rundeck?

- “Turn your operations procedures into self-service jobs.”
- “Safely give others the control and visibility they need.”
- "...it's the swiss army knife for ops."
- biblioteka procedur
- scheduler
- łatwa integracja z systemami do incydentów
- po CI / buildzie nadchodzi deployment
- **GUI & API**

# Czym jest Rundeck?

- stworzony w Javie + Grails + Jetty + Bootstrap
- początki w roku 2010
- Apache license v2.0
- Copyright 2010 DTO Solutions
- “DevOps and Automation Specialists”
- “RunDeck is a command dispatcher with a modern web console. It lets you easily run commands across a set of nodes.”



# Czym jest Rundeck?

- stworzony w Javie + Grails + Jetty + Bootstrap
- **początki w roku 2010**
- Apache license v2.0
- Copyright 2010 DTO Solutions
- “DevOps and Automation Specialists”
- “RunDeck is a command dispatcher with a modern web console. It lets you easily run commands across a set of nodes.”

# Czym jest Rundeck?

- stworzony w Javie + Grails + Jetty + Bootstrap
- początki w roku 2010
- **Apache license v2.0**
- Copyright 2010 DTO Solutions
- “DevOps and Automation Specialists”
- “RunDeck is a command dispatcher with a modern web console. It lets you easily run commands across a set of nodes.”

# Czym jest Rundeck?

- stworzony w Javie + Grails + Jetty + Bootstrap
- początki w roku 2010
- Apache license v2.0
- **Copyright 2010 DTO Solutions**
- “DevOps and Automation Specialists”
- “RunDeck is a command dispatcher with a modern web console. It lets you easily run commands across a set of nodes.”

# Czym jest Rundeck?

- stworzony w Javie + Grails + Jetty + Bootstrap
- początki w roku 2010
- Apache license v2.0
- Copyright 2010 DTO Solutions
- **“DevOps and Automation Specialists”**
- “RunDeck is a command dispatcher with a modern web console. It lets you easily run commands across a set of nodes.”

# Czym jest Rundeck?

- stworzony w Javie + Grails + Jetty + Bootstrap
- początki w roku 2010
- Apache license v2.0
- Copyright 2010 DTO Solutions
- “DevOps and Automation Specialists”
- “RunDeck is a command dispatcher with a modern web console. It lets you easily run commands across a set of nodes.”

# Główne koncepty Rundecka

- projekt (kontener jobów, node'ów i konfiguracji)
- job (sekwencja komend, skryptów)
- workflow (sekwencja jobów)
- nodes (hosty dostępne przez SSH)
- wykonanie (uruchomienia jobów)
- użytkownicy
- uprawnienia
- API

# Główne koncepty Rundecka

- projekt (kontener jobów, node'ów i konfiguracji)
- job (sekwencja komend, skryptów)
- workflow (sekwencja jobów)
- nodes (hosty dostępne przez SSH)
- wykonanie (uruchomienia jobów)
- użytkownicy
- uprawnienia
- API

# Główne koncepty Rundecka

- projekt (kontener jobów, node'ów i konfiguracji)
- job (sekwencja komend, skryptów)
- **workflow (sekwencja jobów)**
- nodes (hosty dostępne przez SSH)
- wykonanie (uruchomienia jobów)
- użytkownicy
- uprawnienia
- API



# Główne koncepty Rundecka

- projekt (kontener jobów, node'ów i konfiguracji)
- job (sekwencja komend, skryptów)
- workflow (sekwencja jobów)
- **nodes (hosty dostępne przez SSH)**
- wykonanie (uruchomienia jobów)
- użytkownicy
- uprawnienia
- API

# Główne koncepty Rundecka

- projekt (kontener jobów, node'ów i konfiguracji)
- job (sekwencja komend, skryptów)
- workflow (sekwencja jobów)
- nodes (hosty dostępne przez SSH)
- **wykonanie (uruchomienia jobów)**
- użytkownicy
- uprawnienia
- API

# Główne koncepty Rundecka

- projekt (kontener jobów, node'ów i konfiguracji)
- job (sekwencja komend, skryptów)
- workflow (sekwencja jobów)
- nodes (hosty dostępne przez SSH)
- wykonanie (uruchomienia jobów)
- **użytkownicy**
- uprawnienia
- API

# Główne koncepty Rundecka

- projekt (kontener jobów, node'ów i konfiguracji)
- job (sekwencja komend, skryptów)
- workflow (sekwencja jobów)
- nodes (hosty dostępne przez SSH)
- wykonanie (uruchomienia jobów)
- użytkownicy
- uprawnienia**
- API

# Główne koncepty Rundecka

- projekt (kontener jobów, node'ów i konfiguracji)
- job (sekwencja komend, skryptów)
- workflow (sekwencja jobów)
- nodes (hosty dostępne przez SSH)
- wykonanie (uruchomienia jobów)
- użytkownicy
- uprawnienia
- **API**

# Wysokopoziomowe zalety Rundecka

- Formalizacja procedur w spójnej formie
- Przejrzysty dashboard
- Logowanie, audyt, widoczność
- Elastyczna granulacja uprawnień
- Interfejs współpracy pomiędzy zespołami

# Wysokopoziomowe zalety Rundecka

- Formalizacja procedur w spójnej formie
- **Przejrzysty dashboard**
- Logowanie, audyt, widoczność
- Elastyczna granulacja uprawnień
- Interfejs współpracy pomiędzy zespołami

# Wysokopoziomowe zalety Rundecka

- Formalizacja procedur w spójnej formie
- Przejrzysty dashboard
- **Logowanie, audyt, widoczność**
- Elastyczna granulacja uprawnień
- Interfejs współpracy pomiędzy zespołami



# Wysokopoziomowe zalety Rundecka

- Formalizacja procedur w spójnej formie
- Przejrzysty dashboard
- Logowanie, audyt, widoczność
- Elastyczna granulacja uprawnień**
- Interfejs współpracy pomiędzy zespołami

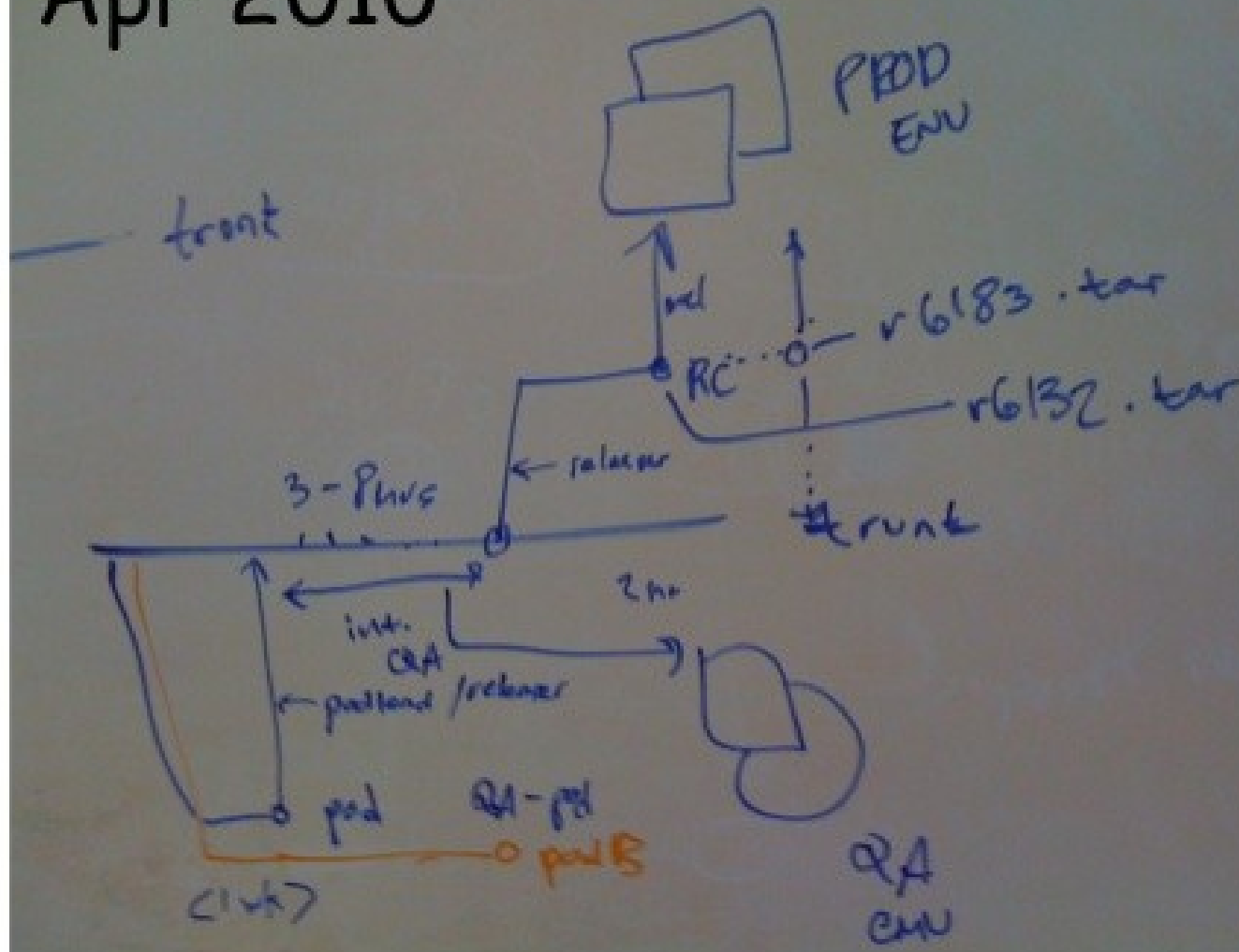
# Wysokopoziomowe zalety Rundecka

- Formalizacja procedur w spójnej formie
- Przejrzysty dashboard
- Logowanie, audyt, widoczność
- Elastyczna granulacja uprawnień
- Interfejs współpracy pomiędzy zespołami

# ≡ Why Rundeck?

## A bit of history

Apr 2010



# Niskopoziomowe zalety Rundecka

→ Spora liczba pluginów i integracji

→ webhooki

→ Jira, IRC

→ Pagerduty, Slack, Hipchat, Redmine

→ Puppet, Salt, Ansible, Chef

→ Nexus, Jenkins

→ AWS EC2, S3

→ Paczki w popularnych dystrybucjach

# Niskopoziomowe zalety Rundecka

→ Spora liczba pluginów i integracji

→ **webhooki**

→ Jira, IRC

→ Pagerduty, Slack, Hipchat, Redmine

→ Puppet, Salt, Ansible, Chef

→ Nexus, Jenkins

→ AWS EC2, S3

→ Paczki w popularnych dystrybucjach

# Niskopoziomowe zalety Rundecka

- Spora liczba pluginów i integracji
  - webhooki
  - Jira, IRC
  - Pagerduty, Slack, Hipchat, Redmine
  - Puppet, Salt, Ansible, Chef
  - Nexus, Jenkins
  - AWS EC2, S3
- Paczki w popularnych dystrybucjach

# Niskopoziomowe zalety Rundecka

- Spora liczba pluginów i integracji
  - webhooki
  - Jira, IRC
  - Pagerduty, Slack, Hipchat, Redmine
  - Puppet, Salt, Ansible, Chef
  - Nexus, Jenkins
  - AWS EC2, S3
- Paczki w popularnych dystrybucjach

# Niskopoziomowe zalety Rundecka

- Spora liczba pluginów i integracji
  - webhooki
  - Jira, IRC
  - Pagerduty, Slack, Hipchat, Redmine
  - **Puppet, Salt, Ansible, Chef**
  - Nexus, Jenkins
  - AWS EC2, S3
- Paczki w popularnych dystrybucjach



# Niskopoziomowe zalety Rundecka

- Spora liczba pluginów i integracji
  - webhooki
  - Jira, IRC
  - Pagerduty, Slack, Hipchat, Redmine
  - Puppet, Salt, Ansible, Chef
  - Nexus, Jenkins**
  - AWS EC2, S3
- Paczki w popularnych dystrybucjach

# Niskopoziomowe zalety Rundecka

- Spora liczba pluginów i integracji
  - webhooki
  - Jira, IRC
  - Pagerduty, Slack, Hipchat, Redmine
  - Puppet, Salt, Ansible, Chef
  - Nexus, Jenkins
  - **AWS EC2, S3**
- Paczki w popularnych dystrybucjach

# Niskopoziomowe zalety Rundecka

- Spora liczba pluginów i integracji
  - webhooki
  - Jira, IRC
  - Pagerduty, Slack, Hipchat, Redmine
  - Puppet, Salt, Ansible, Chef
  - Nexus, Jenkins
  - AWS EC2, S3
- Paczki w popularnych dystrybucjach

# Rundeck vs Jenkins?

- Jenkins is for Development (software builds)
- Rundeck is for Operations (executing operational tasks)
- Both are complementary
- Rundeck has nodes (hosts) and inventory concept
- Rundeck executes jobs / workflows on nodes (inventory)
- Rundeck has ad-hoc commands (you can run those on nodes)
- Rundeck provides you with fine-grained ACLs

# Rundeck vs Jenkins?

- Jenkins is for Development (software builds)
- **Rundeck is for Operations (executing operational tasks)**
- Both are complementary
- Rundeck has nodes (hosts) and inventory concept
- Rundeck executes jobs / workflows on nodes (inventory)
- Rundeck has ad-hoc commands (you can run those on nodes)
- Rundeck provides you with fine-grained ACLs

# Rundeck vs Jenkins?

- Jenkins is for Development (software builds)
- Rundeck is for Operations (executing operational tasks)
- **Both are complementary**
- Rundeck has nodes (hosts) and inventory concept
- Rundeck executes jobs / workflows on nodes (inventory)
- Rundeck has ad-hoc commands (you can run those on nodes)
- Rundeck provides you with fine-grained ACLs

# Rundeck vs Jenkins?

- Jenkins is for Development (software builds)
- Rundeck is for Operations (executing operational tasks)
- Both are complementary
- **Rundeck has nodes (hosts) and inventory concept**
- Rundeck executes jobs / workflows on nodes (inventory)
- Rundeck has ad-hoc commands (you can run those on nodes)
- Rundeck provides you with fine-grained ACLs

# Rundeck vs Jenkins?

- Jenkins is for Development (software builds)
- Rundeck is for Operations (executing operational tasks)
- Both are complementary
- Rundeck has nodes (hosts) and inventory concept
- **Rundeck executes jobs / workflows on nodes (inventory)**
- Rundeck has ad-hoc commands (you can run those on nodes)
- Rundeck provides you with fine-grained ACLs



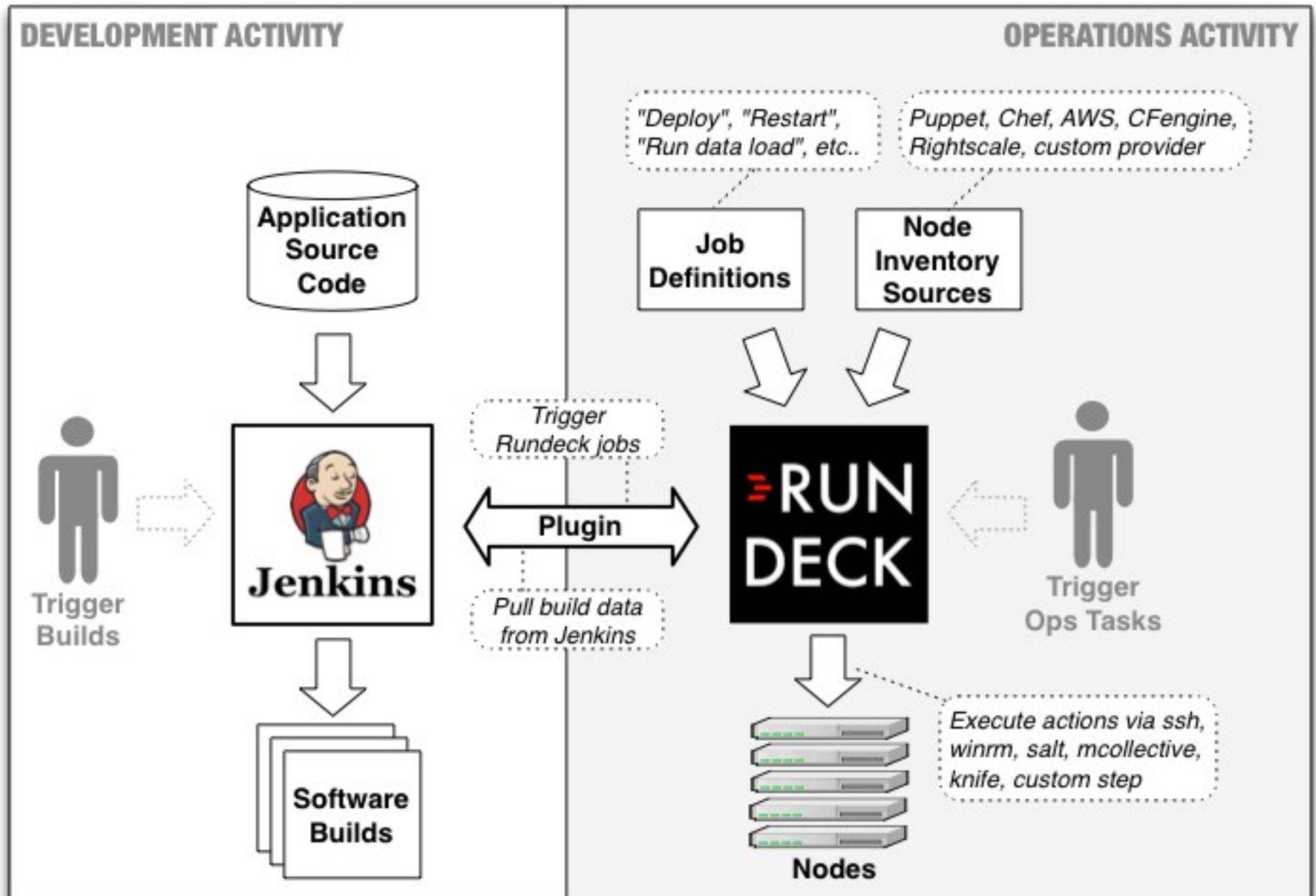
# Rundeck vs Jenkins?

- Jenkins is for Development (software builds)
- Rundeck is for Operations (executing operational tasks)
- Both are complementary
- Rundeck has nodes (hosts) and inventory concept
- Rundeck executes jobs / workflows on nodes (inventory)
- **Rundeck has ad-hoc commands (you can run those on nodes)**
- Rundeck provides you with fine-grained ACLs

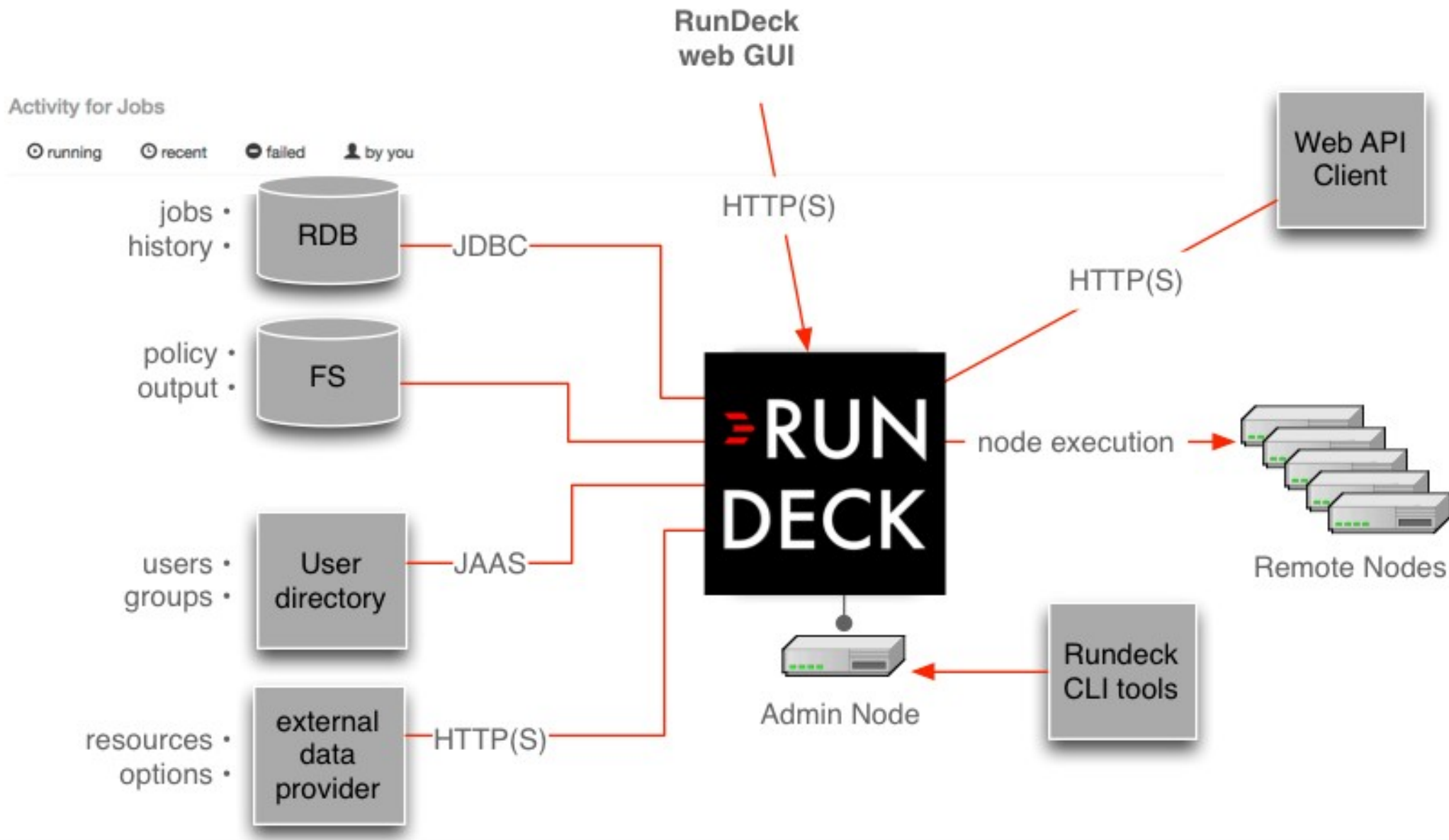
# Rundeck vs Jenkins?

- Jenkins is for Development (software builds)
- Rundeck is for Operations (executing operational tasks)
- Both are complementary
- Rundeck has nodes (hosts) and inventory concept
- Rundeck executes jobs / workflows on nodes (inventory)
- Rundeck has ad-hoc commands (you can run those on nodes)
- Rundeck provides you with fine-grained ACLs

# Rundeck and Jenkins



# Architektura Rundecka



And now  
for something  
completely different...



systemd



**SYSTEM D**  
Live a Better Life!



**systemd**  
**REFUGEES**

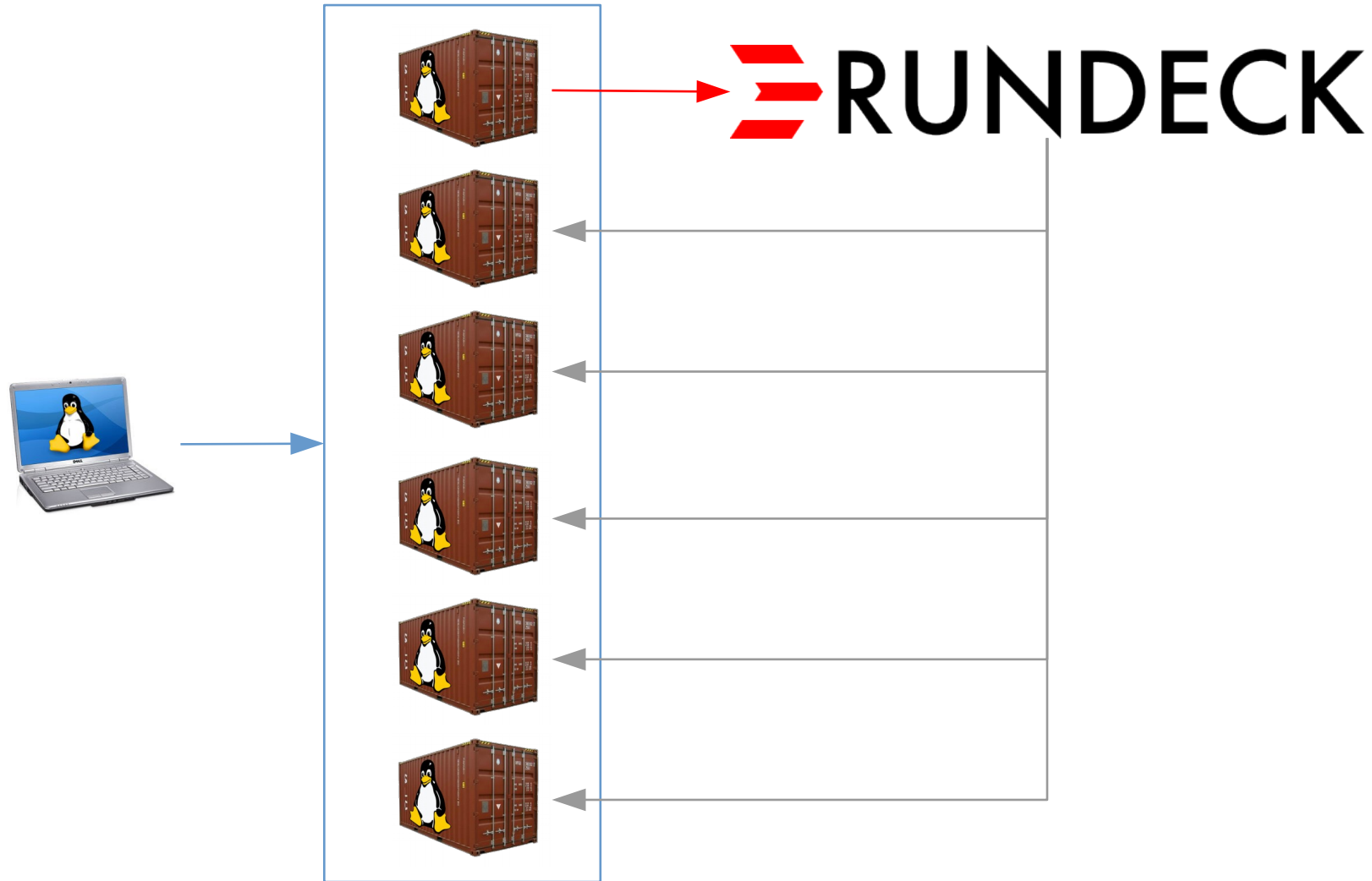


Bring your family to a BSD of your choice

**WELCOME**



# Lab for demo



# Rundeck: przegląd

→ konfiguracja ogólna

→ projekt

→ job

→ nodes

→ command

# Rundeck: przegląd

→ konfiguracja ogólna

→ **projekt**

→ job

→ nodes

→ command

# Rundeck: przegląd

→ konfiguracja ogólna

→ projekt

**→ job**

→ nodes

→ command

# Rundeck: przegląd

- konfiguracja ogólna
- projekt
- job
- **nodes**
- command

# Rundeck: przegląd

- konfiguracja ogólna
- projekt
- job
- nodes
- **command**

# Rundeck adhoc

demo (bez ansible'a)

# Rundeck & Ansible

- Use Rundeck only as GUI, API and scheduler
- Bring Rundeck inventory to common with Ansible
- Remember that Rundeck is stupid
  - Create workspace for jobs
  - Clean up this workspace automatically
  - Only run ansible-playbooks
- Good practices (in my opinion)?
  - Common provisioning core jobs
  - Common “playbook-runner” job
  - Integrate Rundeck & Ansible debug



# Rundeck & Ansible

- Use Rundeck only as GUI, API and scheduler
- **Bring Rundeck inventory to common with Ansible**
- Remember that Rundeck is stupid
  - Create workspace for jobs
  - Clean up this workspace automatically
  - Only run ansible-playbooks
- Good practices (in my opinion)?
  - Common provisioning core jobs
  - Common “playbook-runner” job
  - Integrate Rundeck & Ansible debug

# Rundeck & Ansible

- Use Rundeck only as GUI, API and scheduler
- Bring Rundeck inventory to common with Ansible
- **Remember that Rundeck is stupid**
  - Create workspace for jobs
  - Clean up this workspace automatically
  - Only run ansible-playbooks
- Good practices (in my opinion)?
  - Common provisioning core jobs
  - Common “playbook-runner” job
  - Integrate Rundeck & Ansible debug

# Rundeck & Ansible

- Use Rundeck only as GUI, API and scheduler
- Bring Rundeck inventory to common with Ansible
- Remember that Rundeck is stupid
  - **Create workspace for jobs**
  - Clean up this workspace automatically
  - Only run ansible-playbooks
- Good practices (in my opinion)?
  - Common provisioning core jobs
  - Common “playbook-runner” job
  - Integrate Rundeck & Ansible debug

# Rundeck & Ansible

- Use Rundeck only as GUI, API and scheduler
- Bring Rundeck inventory to common with Ansible
- Remember that Rundeck is stupid
  - Create workspace for jobs
  - **Clean up this workspace automatically**
  - Only run ansible-playbooks
- Good practices (in my opinion)?
  - Common provisioning core jobs
  - Common “playbook-runner” job
  - Integrate Rundeck & Ansible debug

# Rundeck & Ansible

- Use Rundeck only as GUI, API and scheduler
- Bring Rundeck inventory to common with Ansible
- Remember that Rundeck is stupid
  - Create workspace for jobs
  - Clean up this workspace automatically
  - **Only run ansible-playbooks**
- Good practices (in my opinion)?
  - Common provisioning core jobs
  - Common “playbook-runner” job
  - Integrate Rundeck & Ansible debug

# Rundeck & Ansible

- Use Rundeck only as GUI, API and scheduler
- Bring Rundeck inventory to common with Ansible
- Remember that Rundeck is stupid
  - Create workspace for jobs
  - Clean up this workspace automatically
  - Only run ansible-playbooks
- **Good practices (in my opinion)?**
  - Common provisioning core jobs
  - Common “playbook-runner” job
  - Integrate Rundeck & Ansible debug

# Rundeck & Ansible

- Use Rundeck only as GUI, API and scheduler
- Bring Rundeck inventory to common with Ansible
- Remember that Rundeck is stupid
  - Create workspace for jobs
  - Clean up this workspace automatically
  - Only run ansible-playbooks
- Good practices (in my opinion)?
  - **Common provisioning core jobs**
  - Common “playbook-runner” job
  - Integrate Rundeck & Ansible debug

# Rundeck & Ansible

- Use Rundeck only as GUI, API and scheduler
- Bring Rundeck inventory to common with Ansible
- Remember that Rundeck is stupid
  - Create workspace for jobs
  - Clean up this workspace automatically
  - Only run ansible-playbooks
- Good practices (in my opinion)?
  - Common provisioning core jobs
  - **Common “playbook-runner” job**
  - Integrate Rundeck & Ansible debug



# Rundeck & Ansible

- Use Rundeck only as GUI, API and scheduler
- Bring Rundeck inventory to common with Ansible
- Remember that Rundeck is stupid
  - Create workspace for jobs
  - Clean up this workspace automatically
  - Only run ansible-playbooks
- Good practices (in my opinion)?
  - Common provisioning core jobs
  - Common “playbook-runner” job
  - Integrate Rundeck & Ansible debug

# Rundeck & Ansible



run job on localhost node

this job actually invokes ansible-playbook  
on chosen hosts

# Rundeck & Ansible

demo

# Rundeck & Ansible plugin

I'm new to both Rundeck and Ansible so I expect there to be room for improvements.

Only basic features have been implemented in this first pass, so I can play around with both tools.

Liking it very much so far! :)

<https://github.com/Batix/rundeck-ansible-plugin>

# Rundeck API

→ nieoficjalny wrapper pythonowy:

<https://github.com/marklap/rundeckrun>

→ rundeck\_api\_helper.py

→ apitoken.policy vs JSESSIONID

# Rundeck API

→ nieoficjalny wrapper pythonowy:

<https://github.com/marklap/rundeckrun>

→ `rundeck_api_helper.py`

→ `apitoken.policy` vs `JSESSIONID`

# Rundeck API

→ nieoficjalny wrapper pythonowy:

<https://github.com/marklap/rundeckrun>

→ `rundeck_api_helper.py`

→ `apitoken.policy` vs `JSESSIONID`

# To jak z tą współpracą DEV ↔ OPS?

(aka Ocado - use case)





# To jak z tą współpracą DEV ↔ OPS?

(aka Ocado - use case)



# To jak z tą współpracą DEV ↔ OPS?

(aka Ocado – use case)



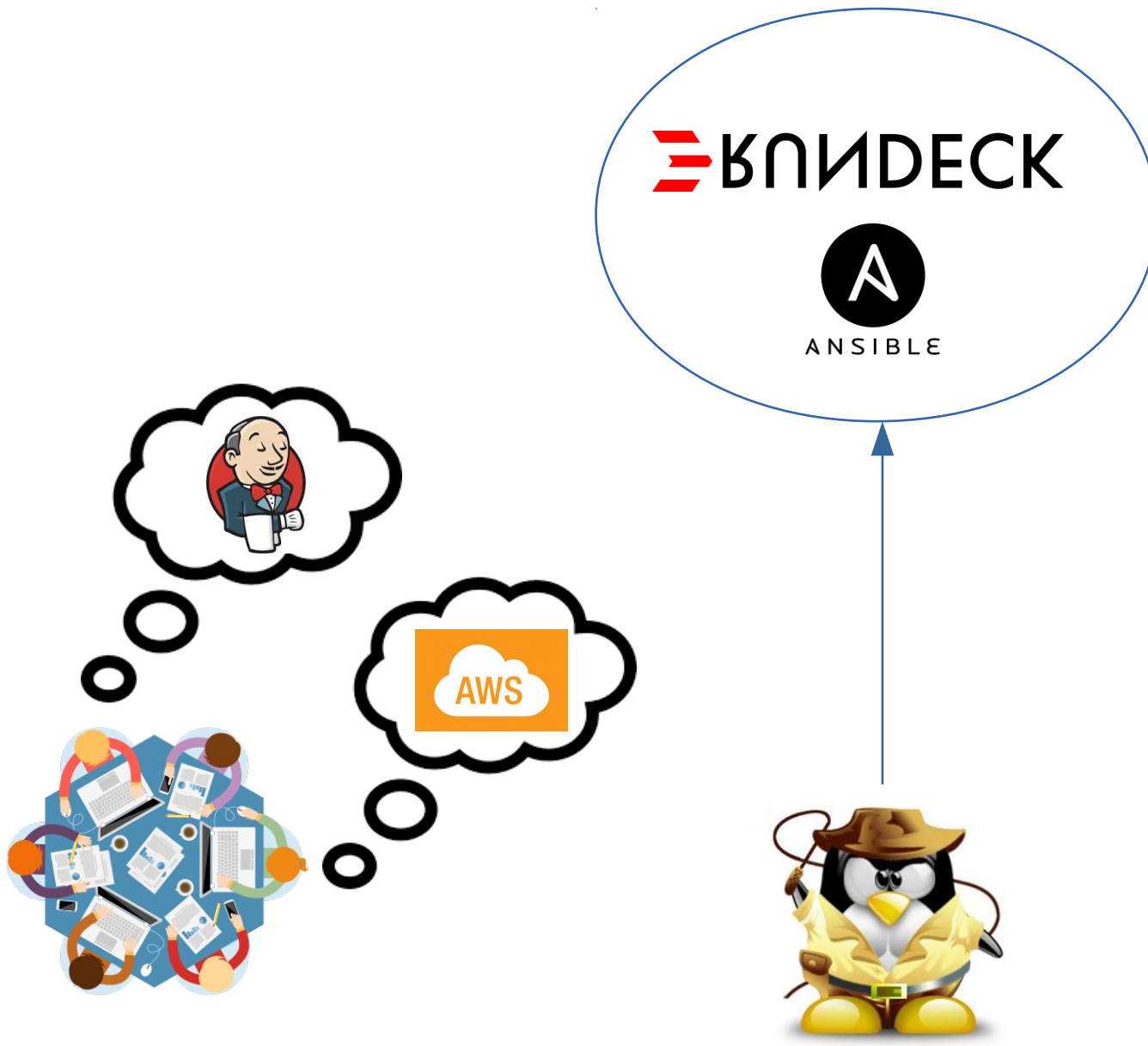
# To jak z tą współpracą DEV ↔ OPS?

(aka Ocado - use case)



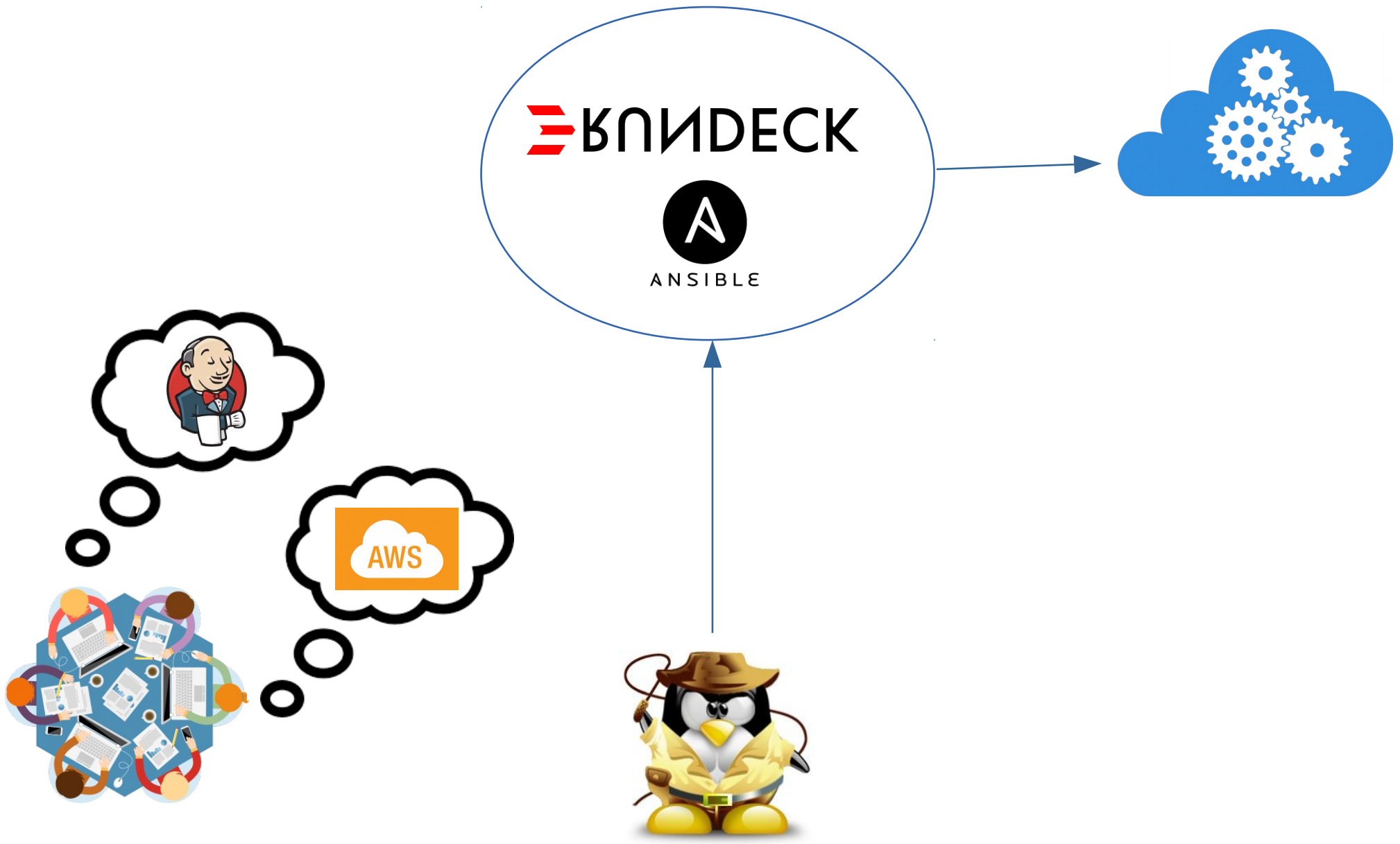
# To jak z tą współpracą DEV ↔ OPS?

(aka Ocado – use case)



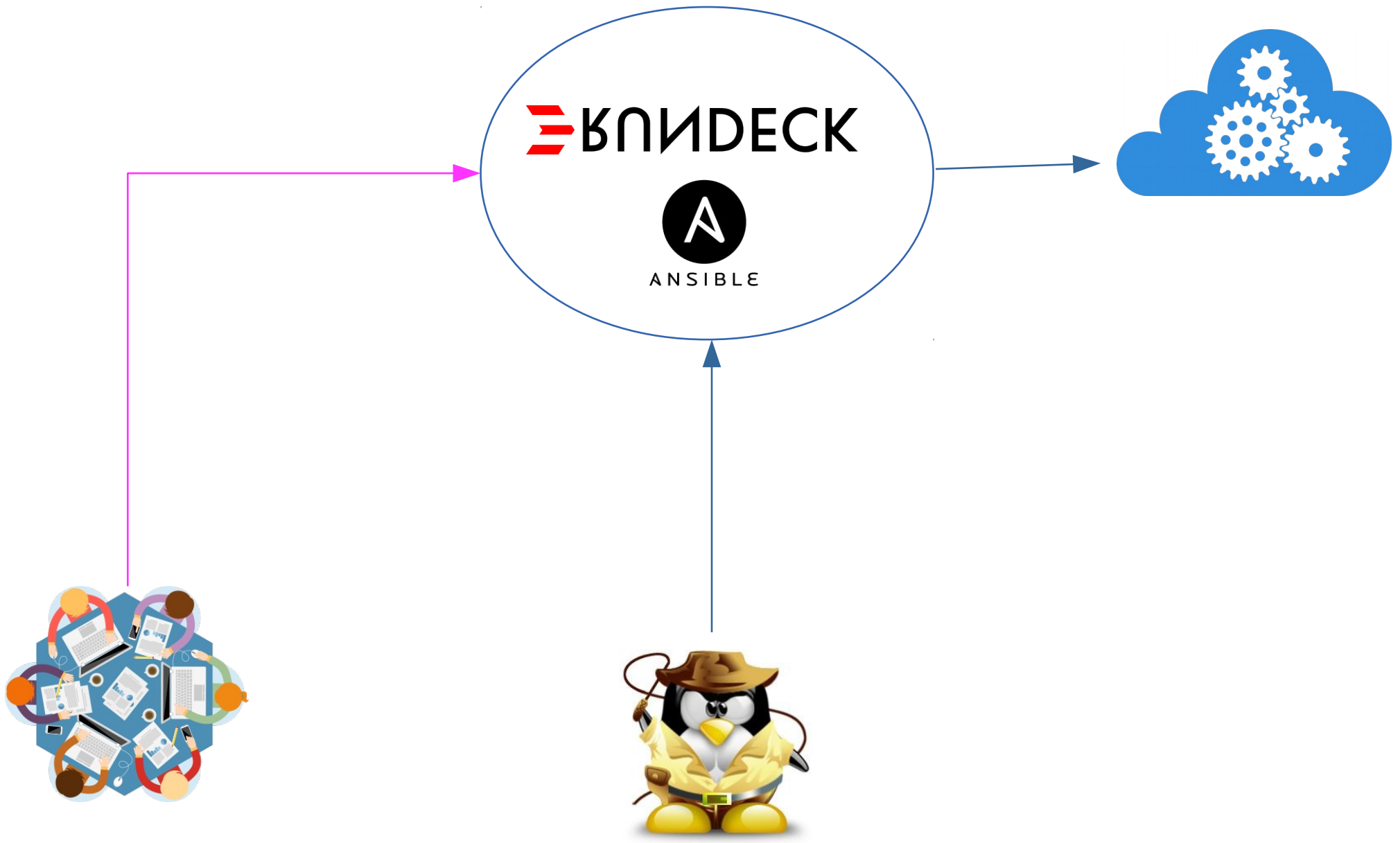
# To jak z tą współpracą DEV ↔ OPS?

(aka Ocado – use case)



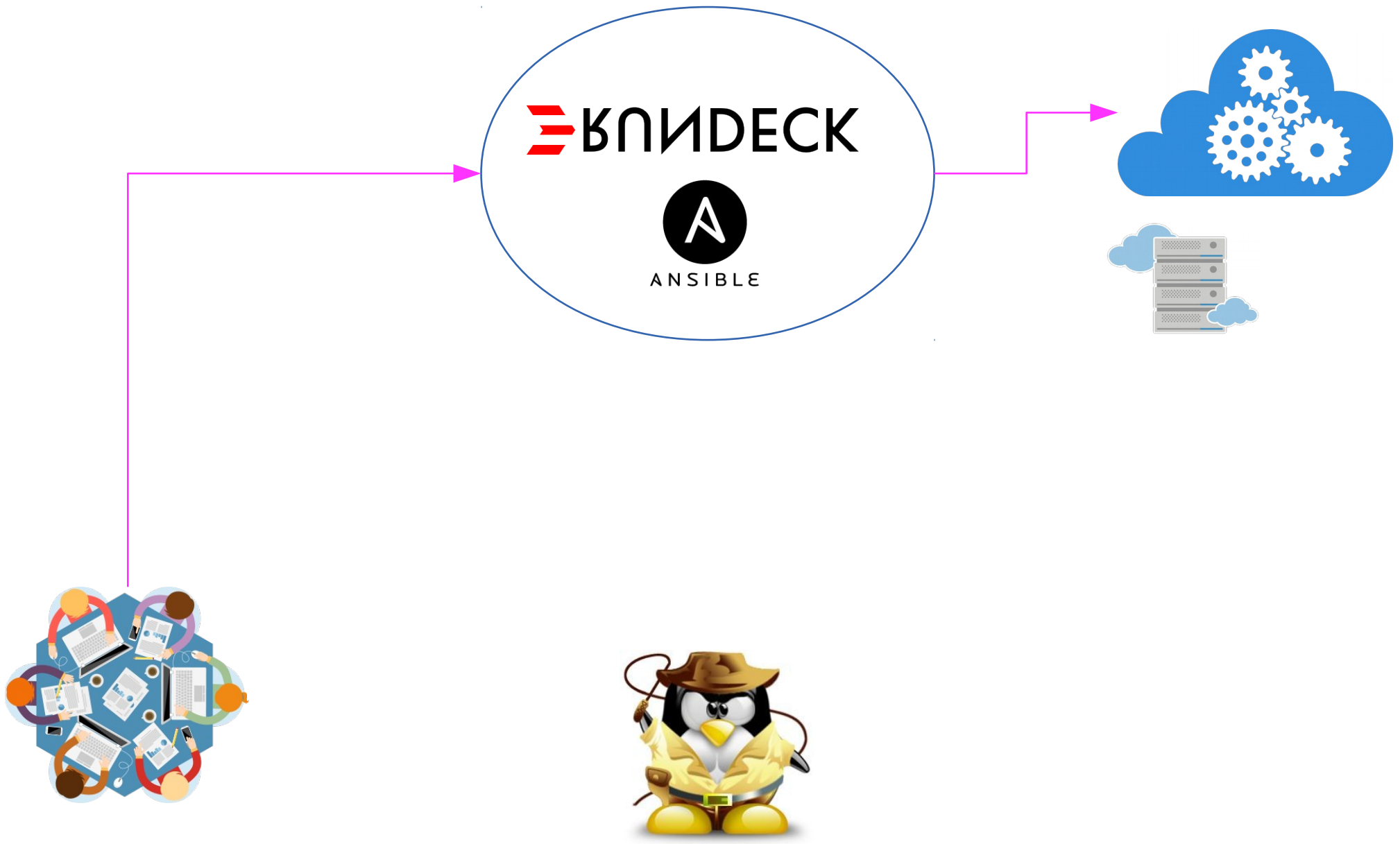
# To jak z tą współpracą DEV ↔ OPS?

(aka Ocado - use case)



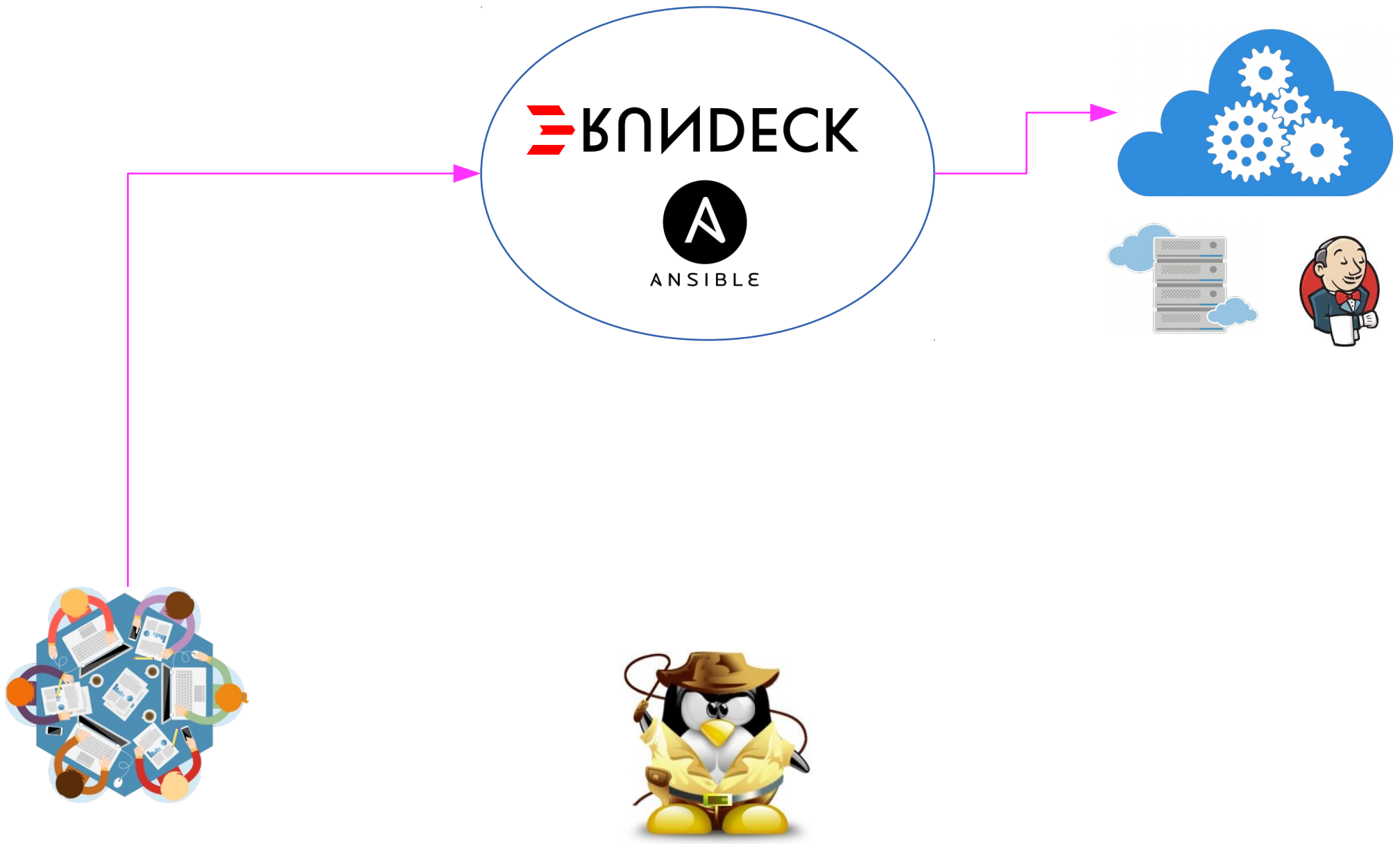
# To jak z tą współpracą DEV ↔ OPS?

(aka Ocado - use case)



# To jak z tą współpracą DEV ↔ OPS?

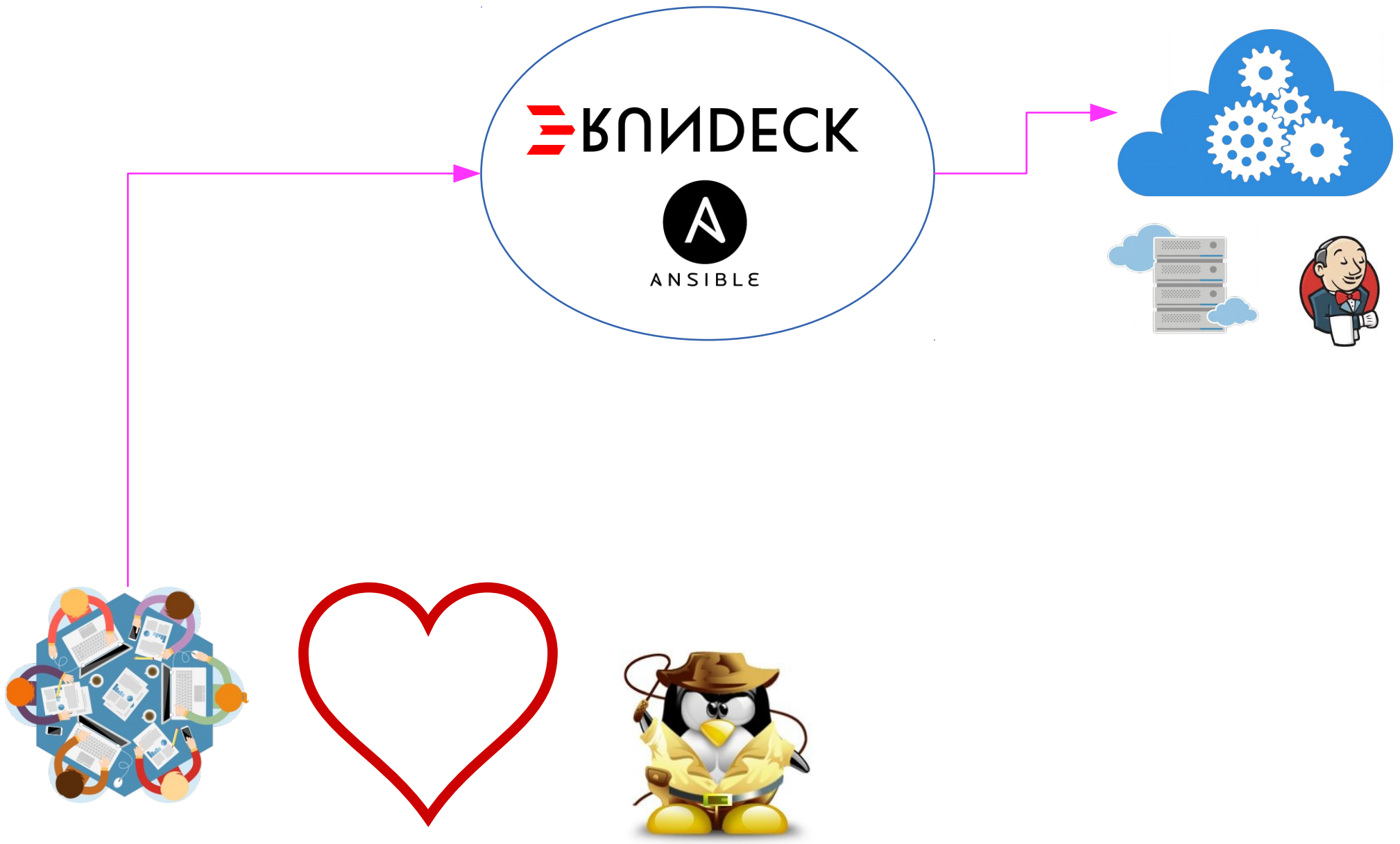
(aka Ocado – use case)





# To jak z tą współpracą DEV ↔ OPS?

(aka Ocado – use case)



# To jak z tą współpracą DEV ↔ OPS?

## (aka Ocado – use case)

Jobs (11) Filter ▶ Expand All Collapse All

### ▼ 3rd\_party

- ▶ [anvil ec2 3rdparty provisioner forgetest3rdpartyapp](#) ▼ Provisioning job for test app forgetest3rdpartyapp; fully templated - don't edit it here! Use Git / Forge / Orchestration repo!
- ▶ [anvil ec2 3rdparty provisioner gocd cluster](#) ▼ Provisioning job for GOCD cluster
- ▶ [anvil ec2 3rdparty provisioner jenkins cluster](#) ▼ Provisioning job for Jenkins cluster
- ▶ [anvil ec2 3rdparty provisioner selenium cluster](#) ▼ Provisioning job for Selenium cluster
- ▶ [anvil ec2 3rdparty terminator gocd cluster](#) ▼ Job for terminating GOCD cluster
- ▶ [anvil ec2 3rdparty terminator jenkins cluster](#) ▼ Job for terminating Jenkins cluster

### ▼ ec2\_core

- ▶ [anvil ec2\\_core playbook runner](#) ▼ This job is used to run playbooks choosen in params; fully templated - don't edit it here! Use Git / Forge / Orchestration repo!
- ▶ [anvil ec2\\_core provisioner](#) ▼ EC2 core provisioning job; fully templated - don't edit it here! Use Git / Forge / Orchestration repo!
- ▶ [anvil ec2\\_core terminator](#) ▼ EC2 core terminating job; fully templated - don't edit it here! Use Git / Forge / Orchestration repo!

### ▼ ec2\_maintenance

- ▶ [anvil ebs remove volumes](#) ▼ Use this job in order to obliterate EBS volumes; fully templated - don't edit it here! Use Git / Forge / Orchestration repo!

### ▼ unit\_tests

- ▶ [anvil ec2\\_ut provisioner](#) ▼ Unit test for EC2 provisioning job; fully templated - don't edit it here! Use Git / Forge / Orchestration repo! 🕒 in 9h44m

# To jak z tą współpracą DEV ↔ OPS?

(aka Ocado – use case)

 KUBERNETES



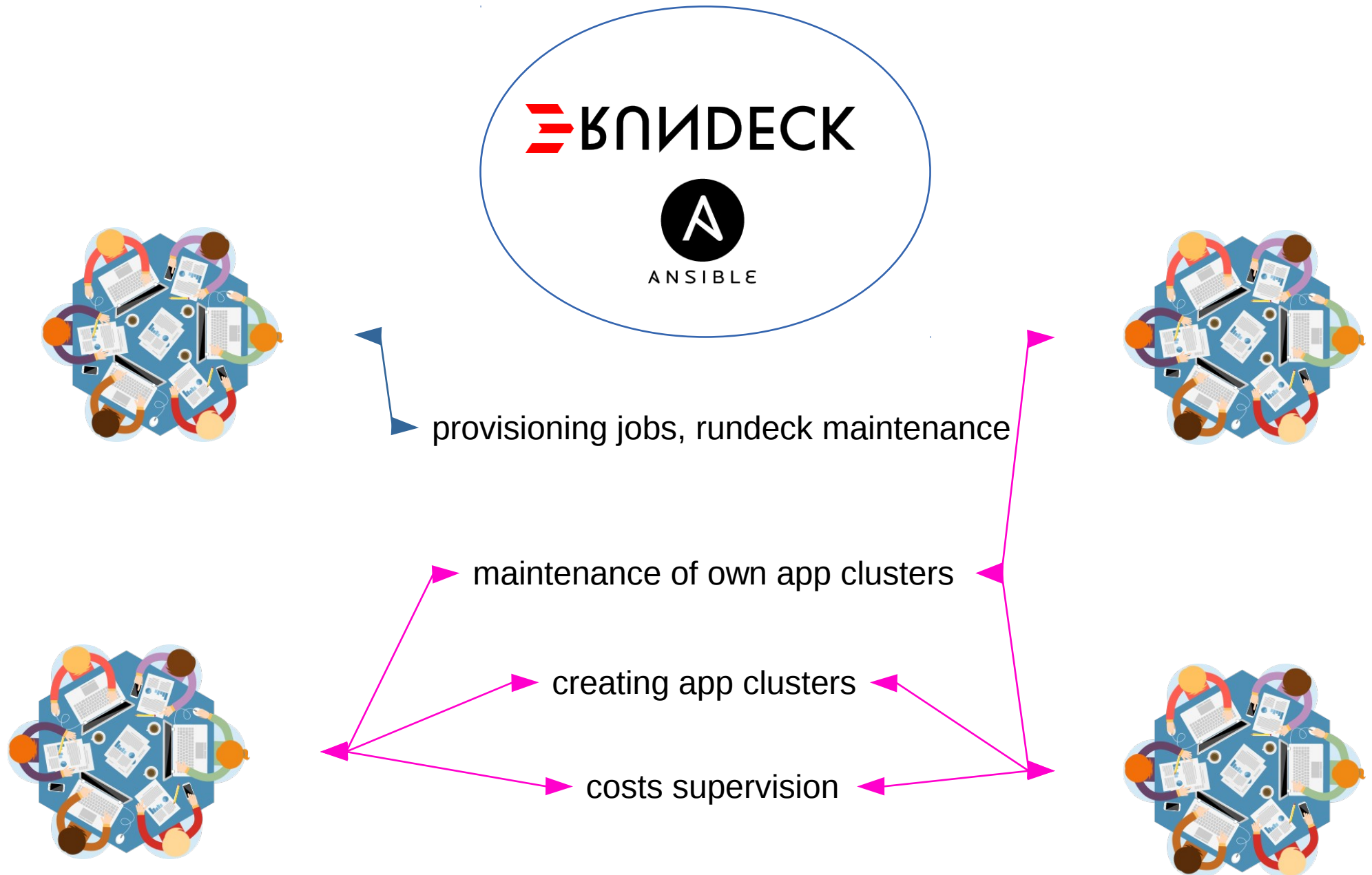
ANSIBLE

provisioning jobs, rundeck maintenance



# To jak z tą współpracą DEV ↔ OPS?

(aka Ocado – use case)



# To jak z tą współpracą DEV ↔ OPS?

Dobre praktyki utrzymania Rundecka w środowisku dev – ops?

- **\*\*nie\*\*** edytujcie jobów na codzień w UI
- joby definiujcie w repo i importujcie (np. przez wrapper API)
- dzielcie się między sobą swoimi jobami / template'ami
- trzymajcie logikę w Ansible'u (Puppecie / cokolwiek) – nie w definicjach jobów rundeckowych
- testujcie automatycznie wykonanie swoich jobów
- przekręcajcie często całą instancję Rundecka (powtarzalność)

# To jak z tą współpracą DEV ↔ OPS?

Dobre praktyki utrzymania Rundecka w środowisku dev – ops?

- **\*\*nie\*\*** edytujcie jobów na codzień w UI
- joby definiujcie w repo i importujcie (np. przez wrapper API)
- dzielcie się między sobą swoimi jobami / template'ami
- trzymajcie logikę w Ansible'u (Puppecie / cokolwiek) – nie w definicjach jobów rundeckowych
- testujcie automatycznie wykonanie swoich jobów
- przekręcajcie często całą instancję Rundecka (powtarzalność)

# To jak z tą współpracą DEV ↔ OPS?

Dobre praktyki utrzymania Rundecka w środowisku dev – ops?

- **\*\*nie\*\*** edytujcie jobów na codzień w UI
- joby definiujcie w repo i importujcie (np. przez wrapper API)
- dzielcie się między sobą swoimi jobami / template'ami
- trzymajcie logikę w Ansible'u (Puppecie / cokolwiek) – nie w definicjach jobów rundeckowych
- testujcie automatycznie wykonanie swoich jobów
- przekręcajcie często całą instancję Rundecka (powtarzalność)

# To jak z tą współpracą DEV ↔ OPS?

Dobre praktyki utrzymania Rundecka w środowisku dev – ops?

- **\*\*nie\*\*** edytujcie jobów na codzień w UI
- joby definiujcie w repo i importujcie (np. przez wrapper API)
- dzielcie się między sobą swoimi jobami / template'ami
- trzymajcie logikę w Ansible'u (Puppecie / cokolwiek) – nie w definicjach jobów rundeckowych
- testujcie automatycznie wykonanie swoich jobów
- przekręcajcie często całą instancję Rundecka (powtarzalność)



# To jak z tą współpracą DEV ↔ OPS?

Dobre praktyki utrzymania Rundecka w środowisku dev – ops?

- **\*\*nie\*\*** edytujcie jobów na codzień w UI
- joby definiujcie w repo i importujcie (np. przez wrapper API)
- dzielcie się między sobą swoimi jobami / template'ami
- trzymajcie logikę w Ansible'u (Puppecie / cokolwiek) – nie w definicjach jobów rundeckowych
- **testujcie automatycznie wykonanie swoich jobów**
- przekręcajcie często całą instancję Rundecka (powtarzalność)

# To jak z tą współpracą DEV ↔ OPS?

Dobre praktyki utrzymania Rundecka w środowisku dev – ops?

- **\*\*nie\*\*** edytujcie jobów na codzień w UI
- joby definiujcie w repo i importujcie (np. przez wrapper API)
- dzielcie się między sobą swoimi jobami / template'ami
- trzymajcie logikę w Ansible'u (Puppecie / cokolwiek) – nie w definicjach jobów rundeckowych
- testujcie automatycznie wykonanie swoich jobów
- przekręcajcie często całą instancję Rundecka (powtarzalność)

# To jak z tą współpracą DEV ↔ OPS?

Rundeck per zespół? Czemu nie

 RUNDECK ←



→  RUNDECK



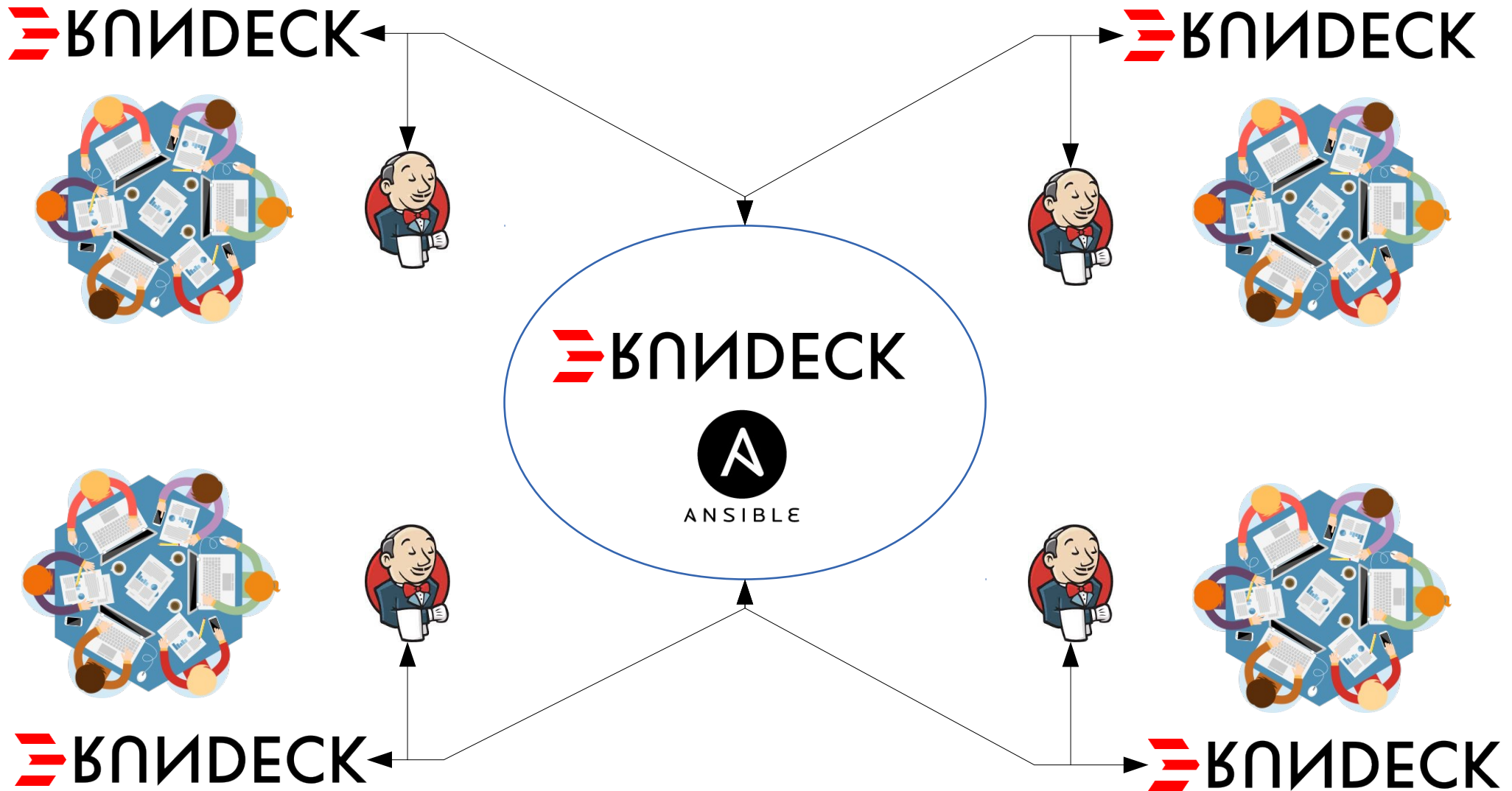
 RUNDECK ←



→  RUNDECK

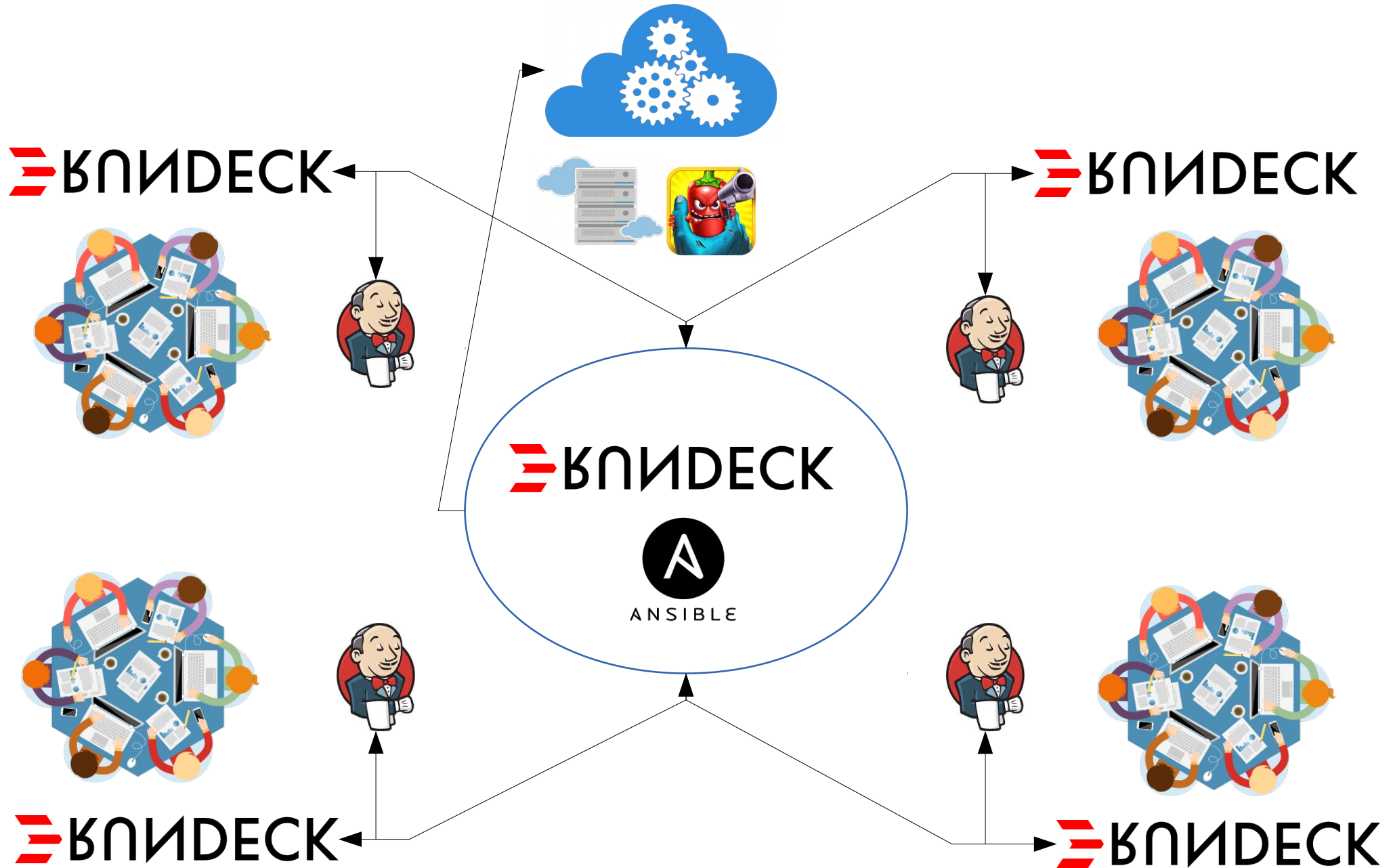
# To jak z tą współpracą DEV ↔ OPS?

Rundeck per zespół? Czemu nie



# To jak z tą współpracą DEV ↔ OPS?

Rundeck per zespół? Czemu nie



Jak chciałem to podsumować?

Jak chciałem to podsumować?



Dzięki :)

Ansible + Rundeck = śpij adminie, śpij

Maciej Lasyk

DevopsKRR meetup #9

Kraków 2016-04-14