divae
Digital Value Enterprise

Automatisierte Deployments mit ansible

Vorstellung

Thorben Hischke
Senior Architect

- Dipl.-Inf. (FH), seit 2007 bei Netpioneer GmbH / diva-e
- Schwerpunkt CMS (Imperia, Polopoly, FirstSpirit)
- Integration und Kundenzufriedenheit

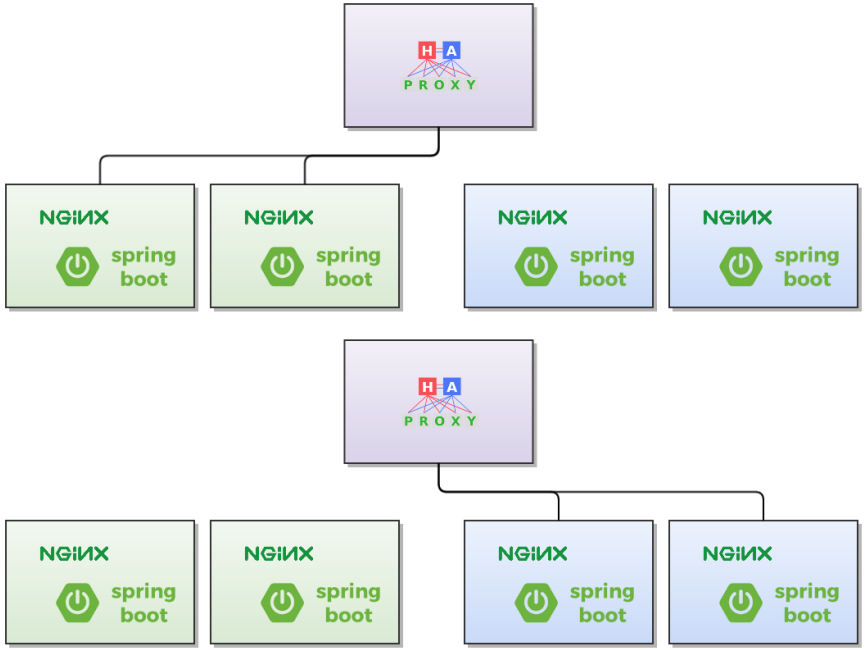
„Ich möchte meine Software qualitätsgesichert per Knopfdruck live bringen. Die Besucher meiner Webseite darf dadurch nicht beeinträchtigt werden.“

„The blue-green deployment approach does this by ensuring you have two production environments, as identical as possible. At any time one of them, let's say blue for the example, is live. As you prepare a new release of your software you do your final stage of testing in the green environment. Once the software is working in the green environment, you switch the router so that all incoming requests go to the green environment - the blue one is now idle.“

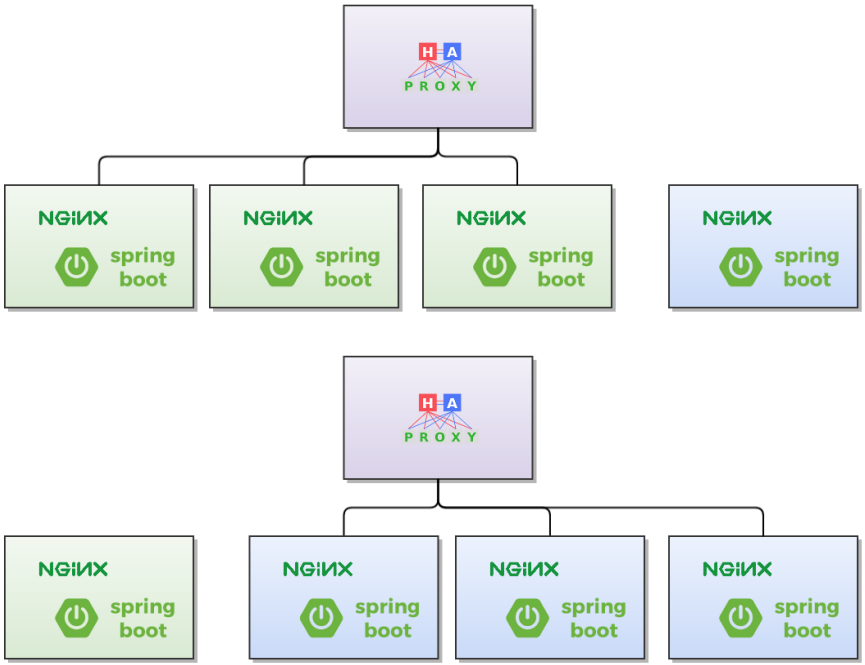
Quelle: *https://martinfowler.com/bliki/BlueGreenDeployment.html*

A/B – Deployment – Variante 1

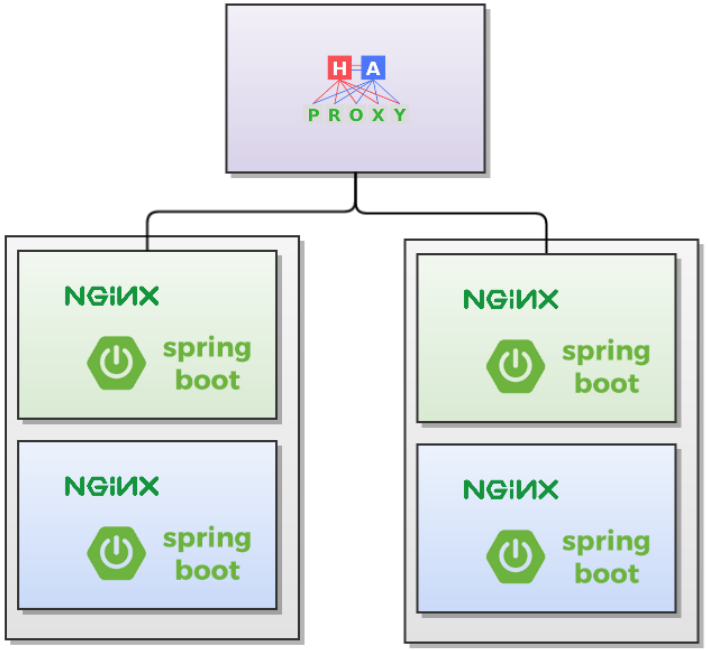
Digital Value Enterprise



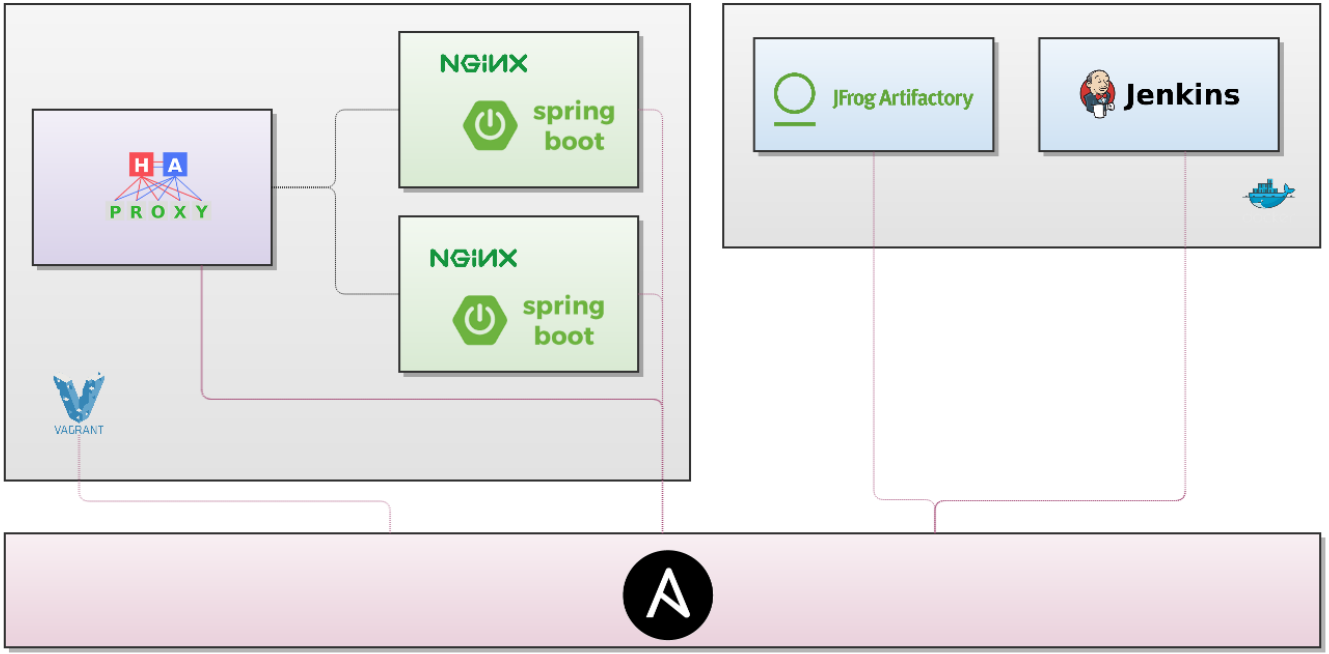
A/B – Deployment – Variante 2



A/B – Deployment – Variante 3



Architektur - Skizze



Warum Provisionierungs-Werkzeuge?

1. Konfiguration wird beschrieben ☛„Infrastructure-as-Code“
   - lesbar
   - versionierbar
2. Reproduzierbar
   - Server befinden sich in einem identischem Zustand
   - kein „auf dem dritten Server war die Config noch nicht angepasst“
3. Automatisierbar

Provisionierungs-Werkzeuge

1. Push-Prinzip, Agent wird nicht benötigt
2. nur SSH und Python werden benötigt
3. leichter Einstieg, steile Leinkurve
4. umfangreiche Module, Rollen sind wiederverwendbar

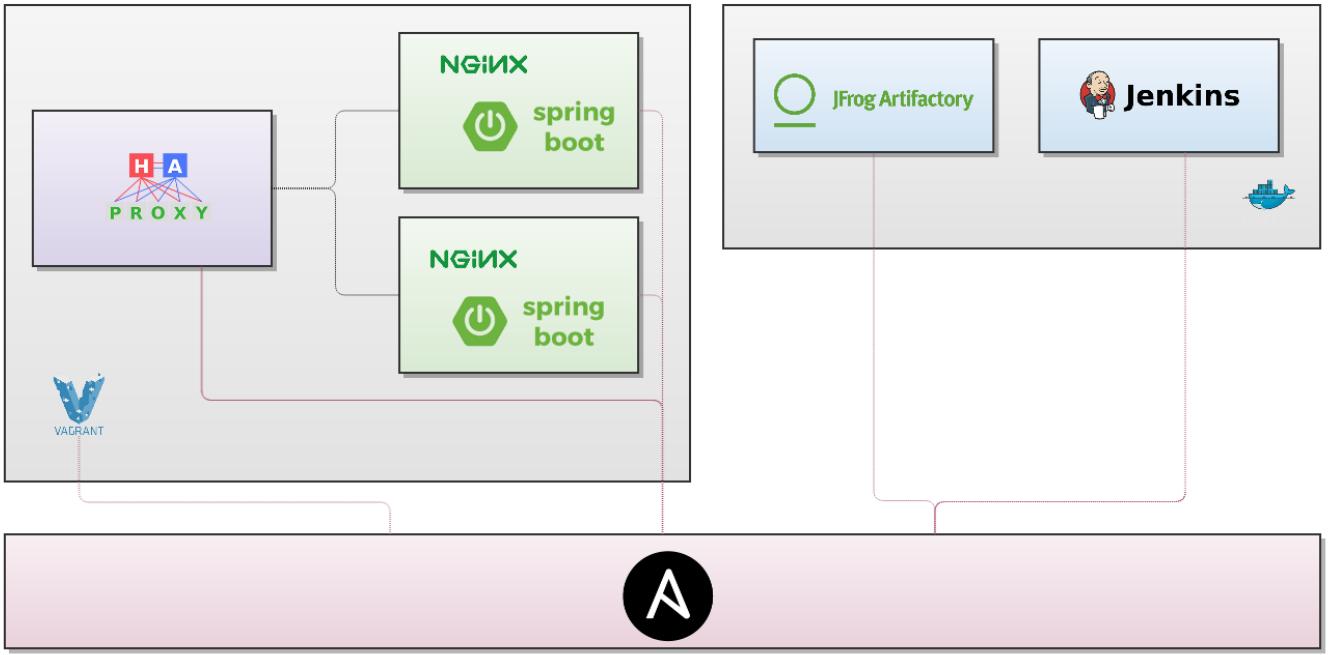
Ansible - Contra

1. YAML
2. Editor-Unterstützung
3. flexible Programmierung („*Wo habe ich diese #X?!-Variable definiert?*“)
4. Keine Abstraktion über OS-Grenzen hinweg

Ein simples Beispiel


```
1     - hosts: webservers
2       vars:
3         http_port: 80
4         max_clients: 200
5       remote_user: root
6       tasks:
7       - name: ensure apache is at the latest version
8         yum: name=httpd state=latest
9       - name: write the apache config file
10        template: src=/srv/httpd.j2 dest=/etc/httpd.conf
11        notify:
12        - restart apache
13      - name: ensure apache is running (and enable it at boot)
14        service: name=httpd state=started enabled=yes
15      handlers:
16        - name: restart apache
17          service: name=httpd state=restarted
18
```

Architektur - Skizze



Inventory


```
1     [stage:children]
2     app
3     haproxy
4
5     [stage:vars]
6
7     [app]
8     app1.local.dev
9     app2.local.dev
10
11    [haproxy]
12    haproxy.local.dev
13
```

Playbooks - playbook-vagrant-app.yml


```
1     ---
2     ## usage: ansible-playbook -i <inventory> playbook-vagrant-app.yml
3
4     - hosts:
5         - app
6       become: true
7       roles:
8         - common
9         - common-app
10        - java
11        - nginx
12
```

Roles - common


```
1       - name: update apt cache
2         apt: update_cache=yes
3
4       - name: install vim
5         apt: name=vim state=installed
6
7       - name: install curl
8         apt: name=curl state=installed
9
10      - name: create install directory
11        file: path="/root/INSTALL" state=directory owner=root group=root mode=0755
12
13      - name: set authorized key in alternate location
14        authorized_key:
15          user: root
16          state: present
17          key: "{{ lookup('file', '/Users/thischke/.ssh/id_rsa_entwicklertag.pub') }}"
18          manage_dir: True
19
```

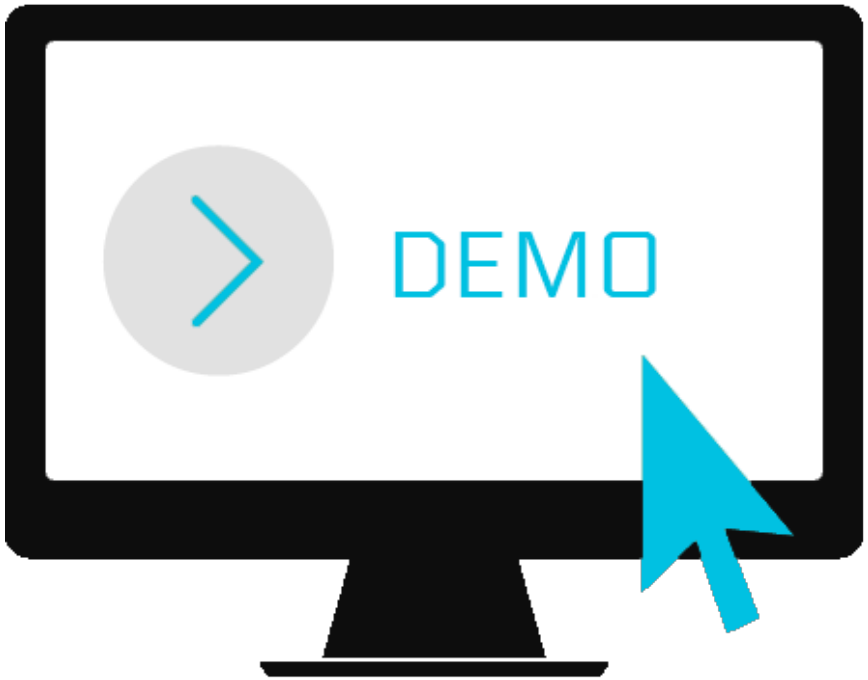
Templates – nginx.conf

Digital Value Enterprise

```
# file /etc/nginx/nginx.conf
# {{ ansible_managed }}

user {{ nginx.user }};
worker_processes {{ nginx.worker.processes }};
pid /var/run/nginx.pid;

events {
    worker_connections  {{ nginx.worker.connections }};
    multi_accept on;
    use epoll;
}

http {
    include       /etc/nginx/mime.types;
    default_type  application/octet-stream;
    server_names_hash_bucket_size 64;

    log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
                      '$status $body_bytes_sent "$http_referer" '
                      '"$http_user_agent" "$http_x_forwarded_for"';
    ...
}
```

Playbooks ausführen

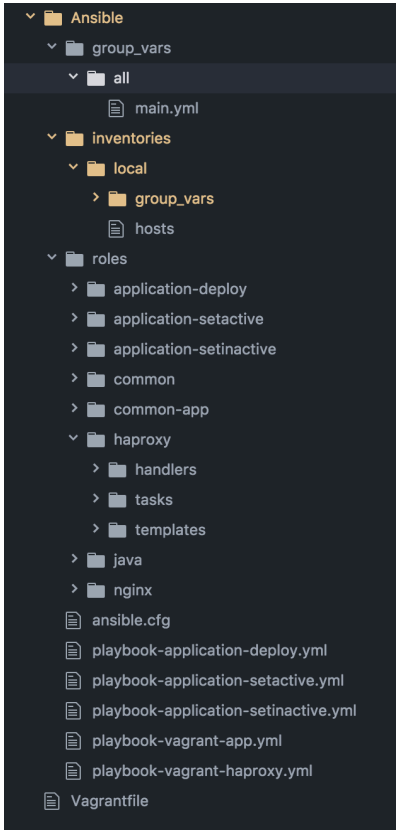

```
1     # deploy application
2     ansible-playbook -i inventories/local playbook-application-deploy.yml
3     ansible-playbook -i inventories/local playbook-application-setactive.yml -l app1.local.dev
4
5
6
7     # limitiert die Ausführung auf den angebenen Host
8     ansible -l|--limit host
9
10     # zeigt an, auf welchen Hosts das Playbook ausgeführt werden wird, führt aber nichts aus
11    ansible --list-hosts
12
13    # dry-run
14    ansible-playbook foo.yml --check
15
16
```


DEMO

Ansible – best practice - Projektstruktur

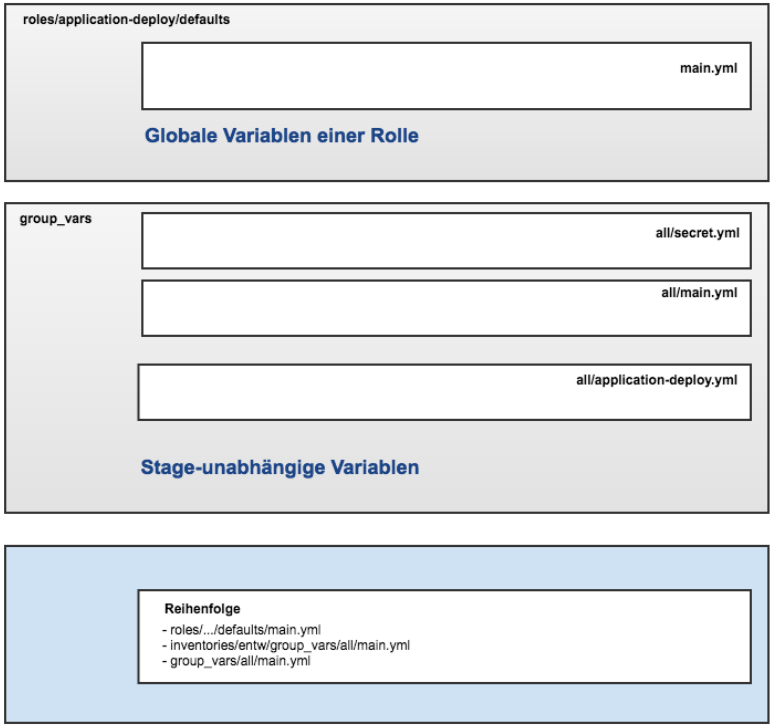
Digital Value Enterprise

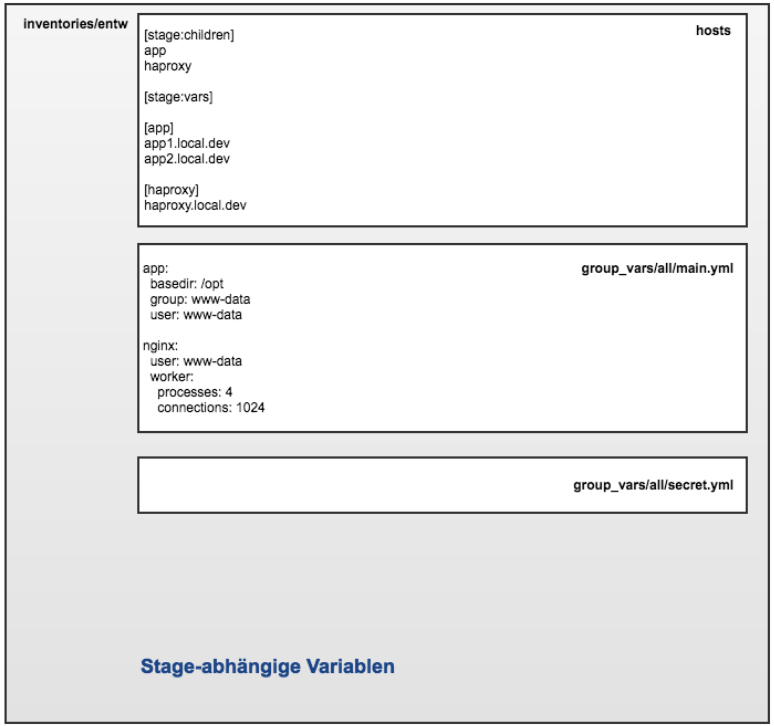
Ansible bietet eine grosse Flexibilität in der Art, wie man sein Projekt strukturiert. Eine gemeinsame Struktur über die Projekte hinweg erleichtert dem Team den Alltag ungemein.



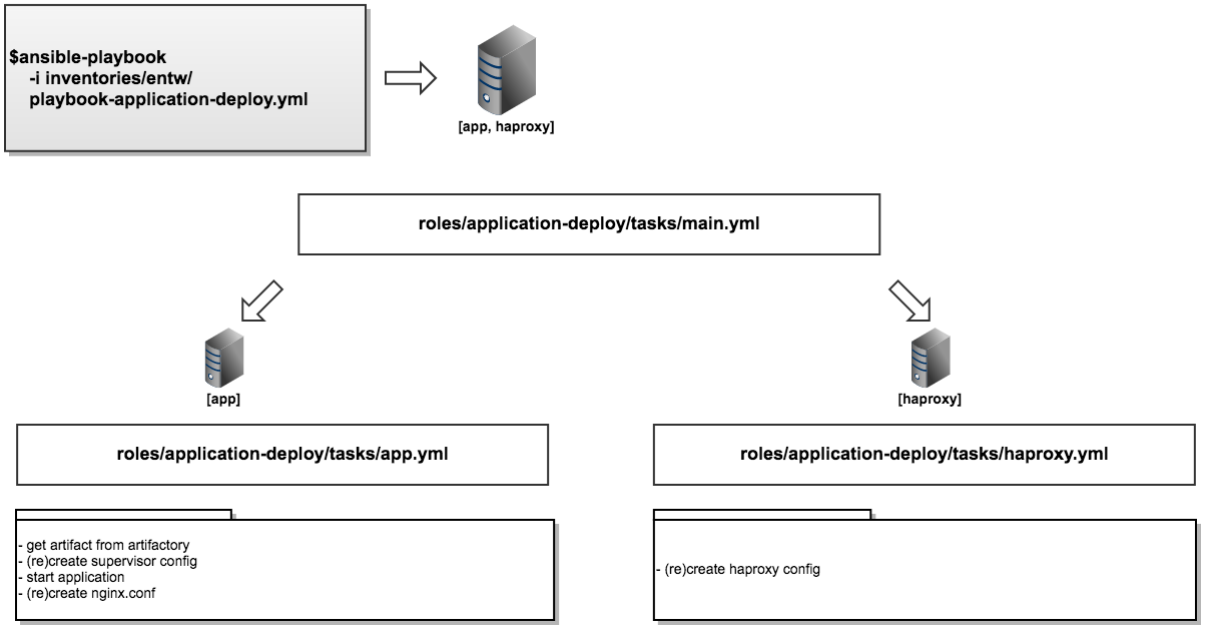
Ansible – best practice - Variablen

Digital Value Enterprise

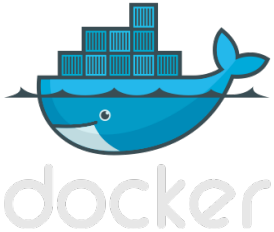




Ansible – best practice - Dokumentation



Alles gut jetzt, oder ... ?!?

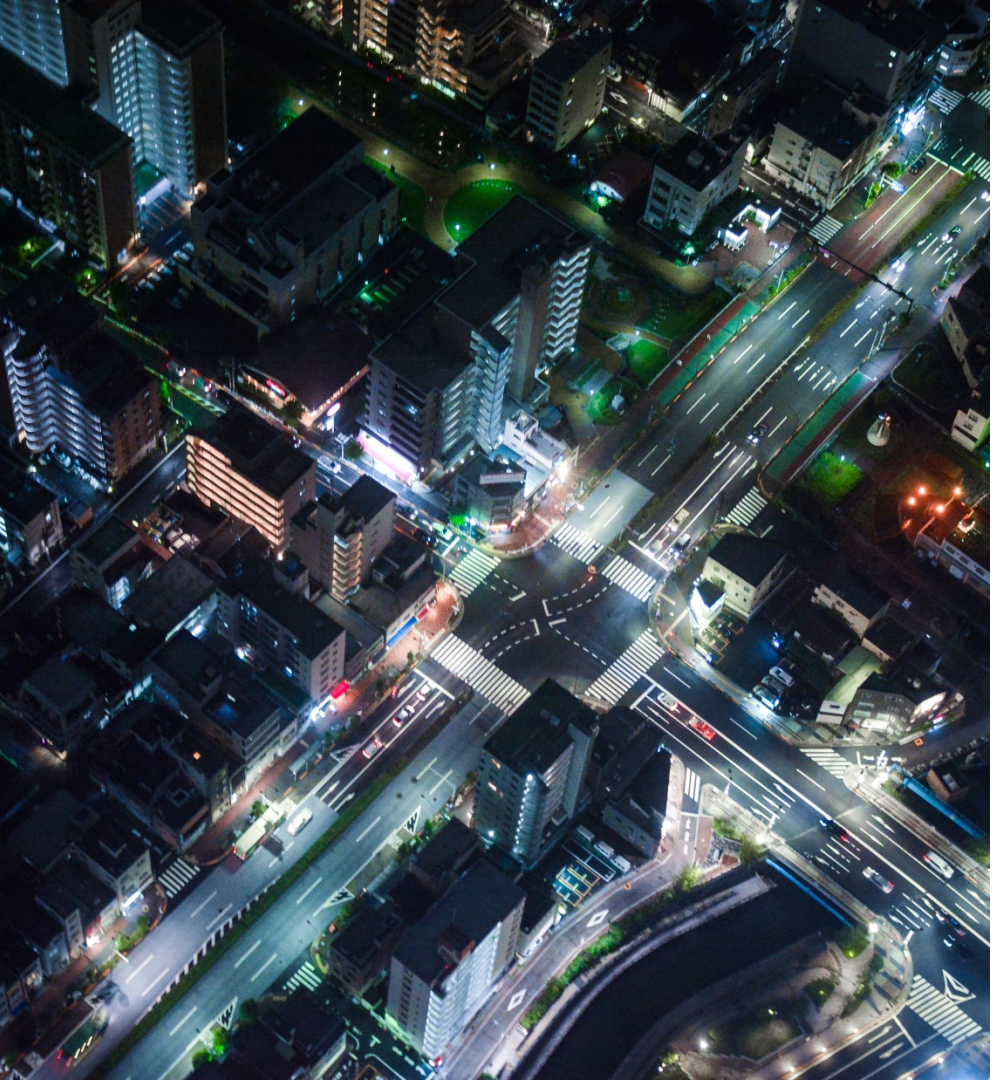


Quellen

https://github.com/thischke/Entwicklertag2017

@thischke

https://www.slideshare.net/thischke/continuous-delivery-with-ansible



Vielen Dank für Ihre Aufmerksamkeit.

See You Next Time

Adresse
diva-e Digital Value Enterprise GmbH
Office Karlsruhe

Ihr Kontakt
Thorben Hischke
Senior Architect

T +49 721 920 60 740
M +49 151 528 78 942
thorben.hischke@diva-e.com
www.diva-e.com

diva-e
Digital Value Enterprise

diva-e. You can’t buy it. You can’t make it.
And you sure can’t fake it.