

Управление производственным процессом разработки программного обеспечения

Infrastructure as a Code (DevOps)

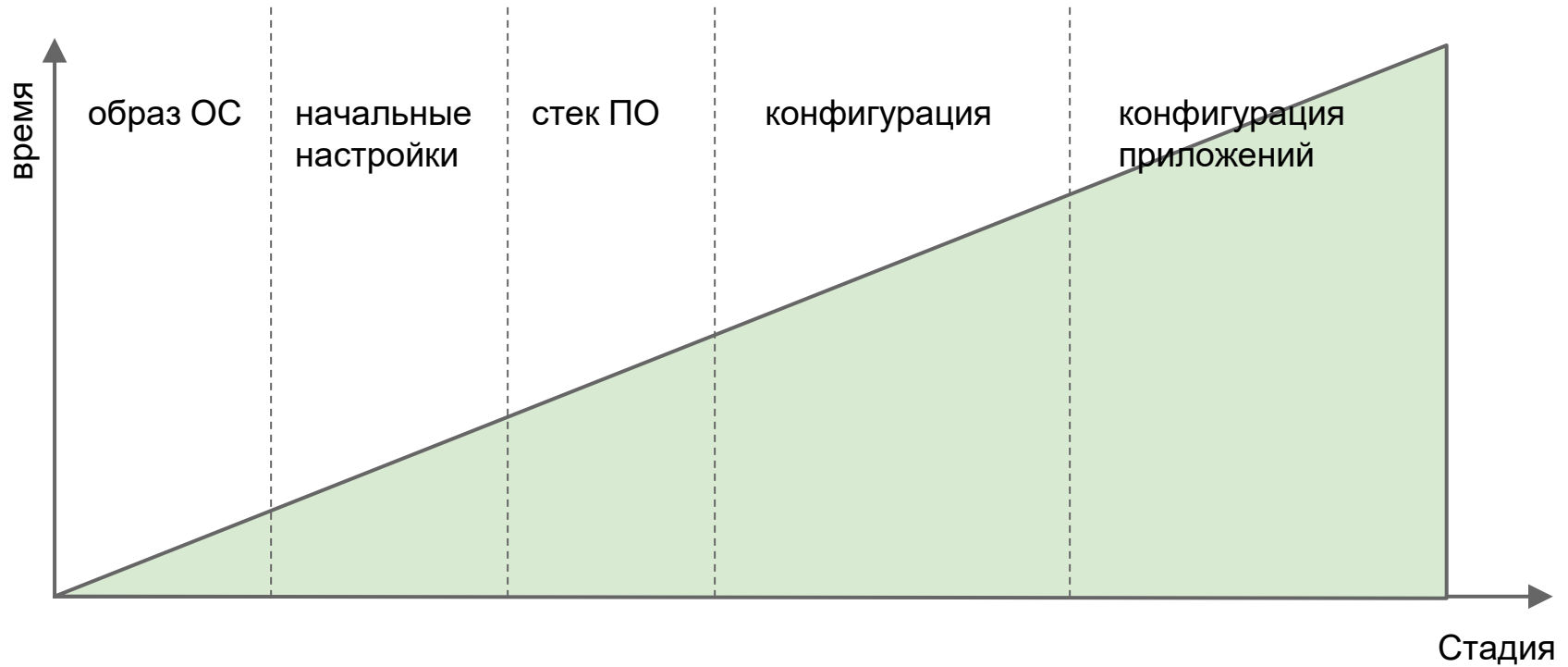
План

- Разработчики и администраторы
- История
- Архитектура
- Виртуализация

Configuration management

- Железо
- Установленное ПО
- Настройки ПО
- Настройки сети
- Мониторинг

Трудозатраты администратора



developers & operators

Dev/Ops

Google trend

● devops
Search term

+ Compare

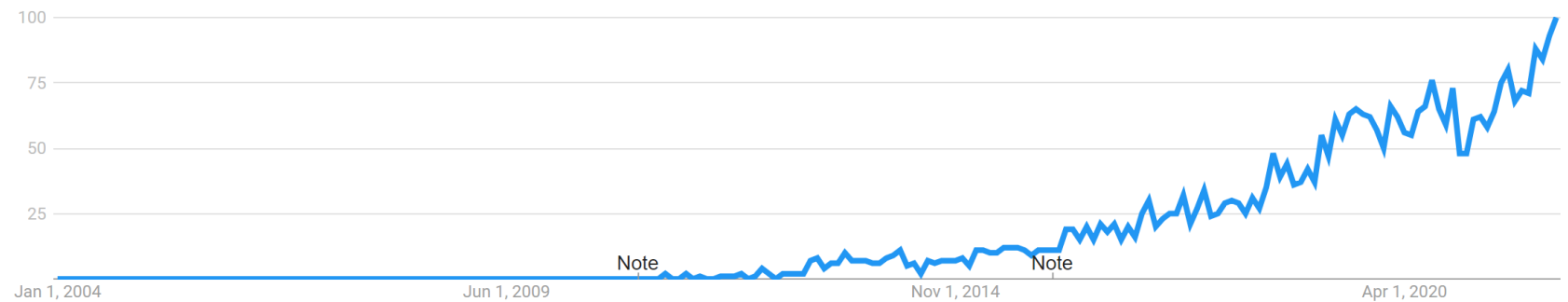
Russia ▼

2004 - present ▼

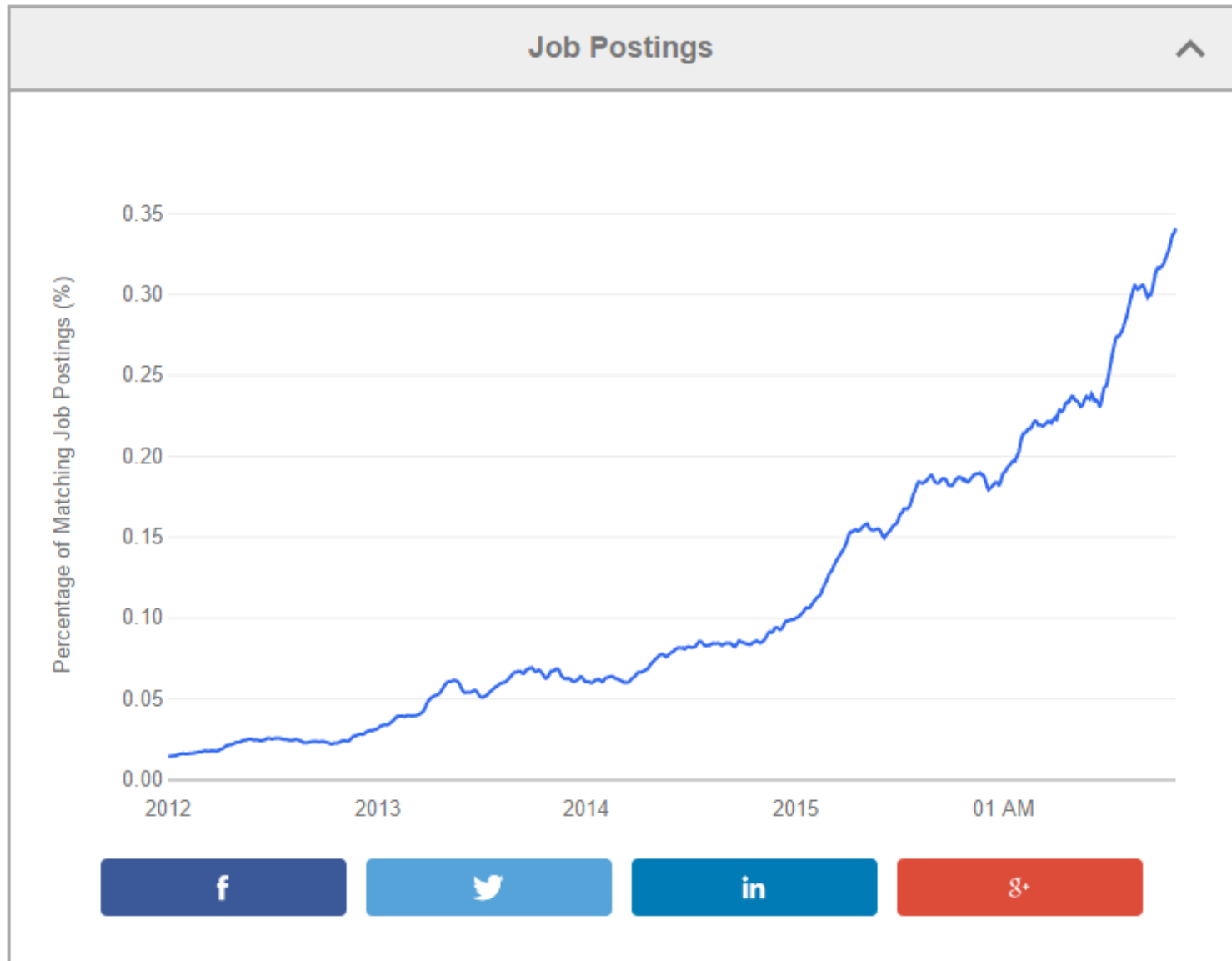
All categories ▼

Web Search ▼

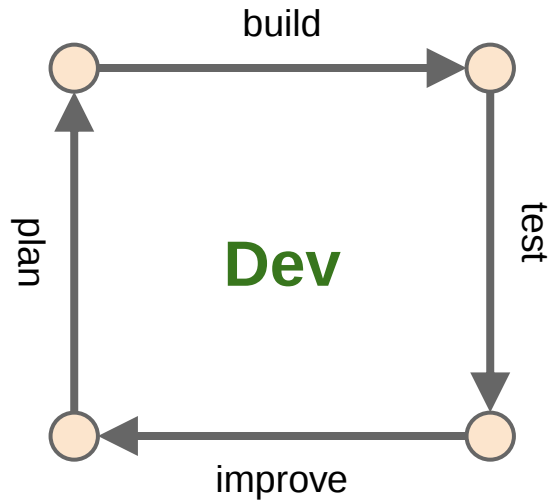
Interest over time ?



Indeed job trend by DevOps keyword



developers & operators



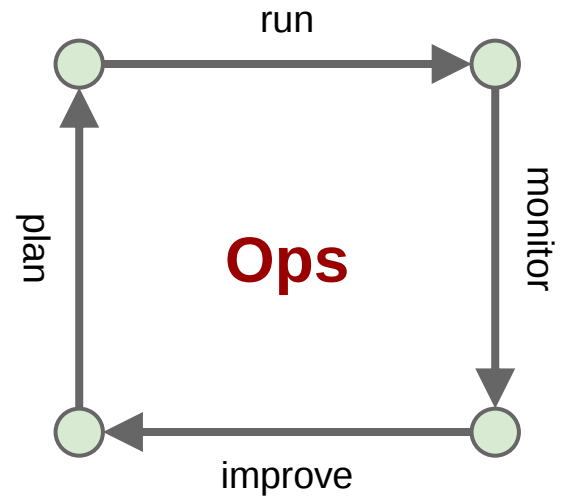
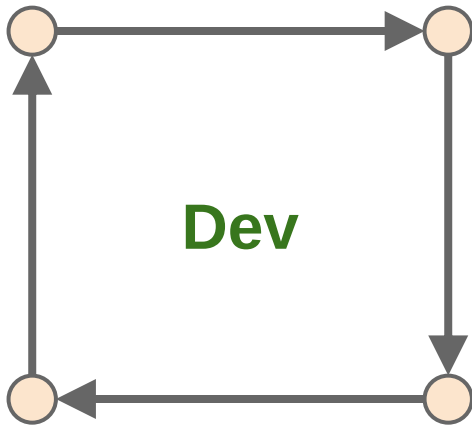
Ops

Люди: agile манифест

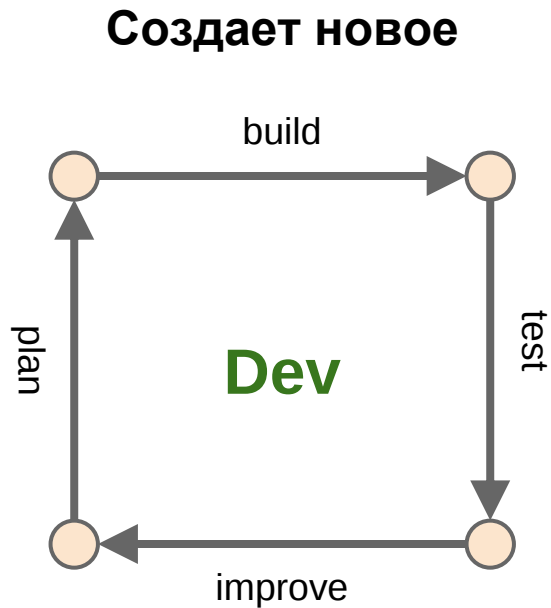
Процессы: XP, Scrum, Kanban

Инструменты: юнит тесты, непрерывная интеграция

developers & operators



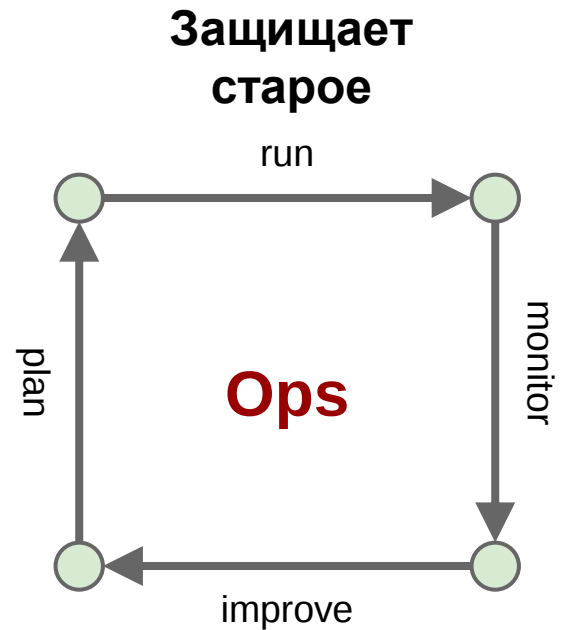
Обратная связь



(мечтатели)

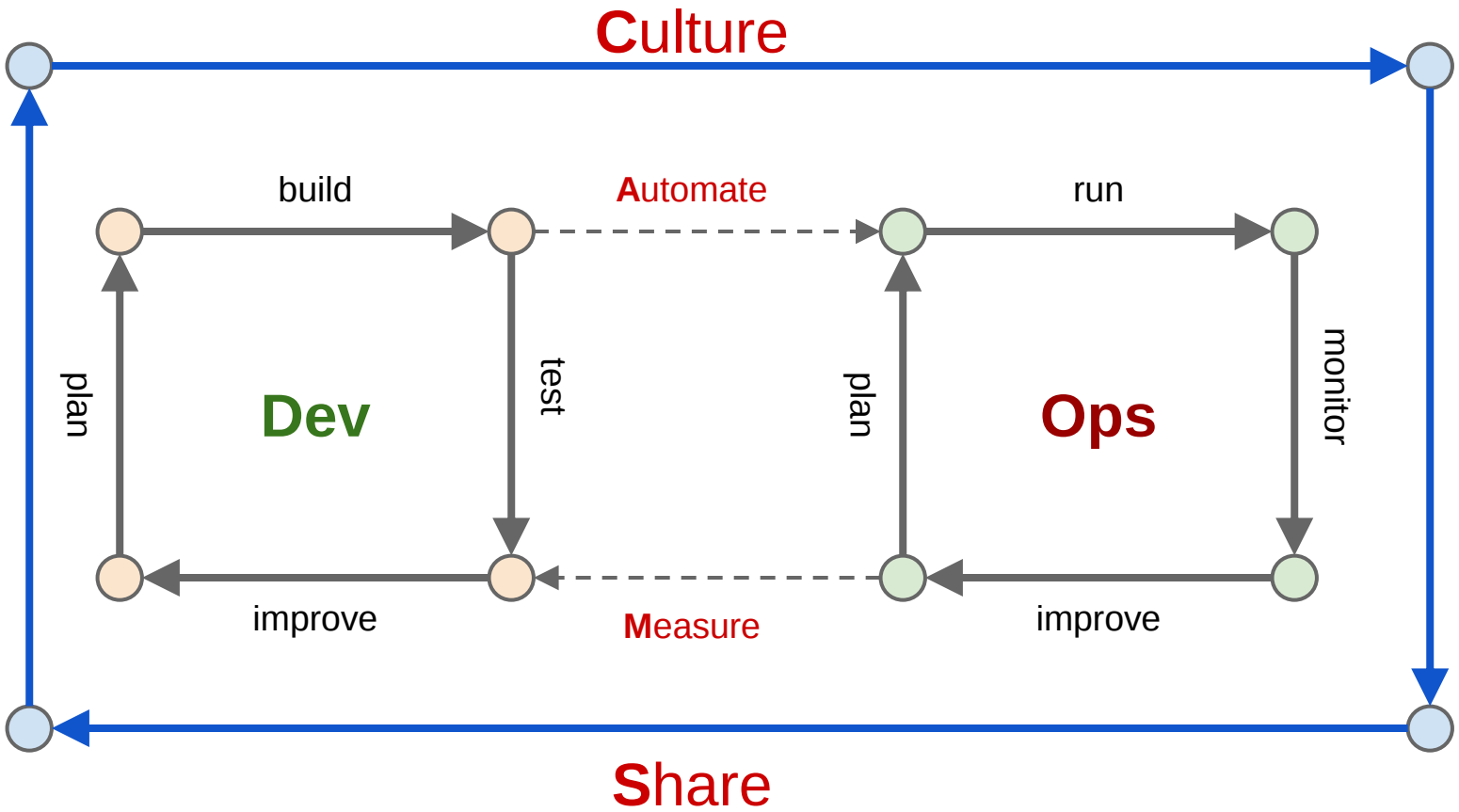
?

(реалисты)



(критики)

CAMS



Culture

Люди и процессы на первом месте. Без культуры попытки все автоматизации провалятся.

Automation

- Инфраструктура как код
- Версионирование
- Переиспользование конфигураций (dev, test, prod)
- Визуализация (с точки зрения управления)

Measure

- Доступ к логам и метрикам
- Расследование инцидентов совместно с разработчиками
- Определение повторяющихся проблем

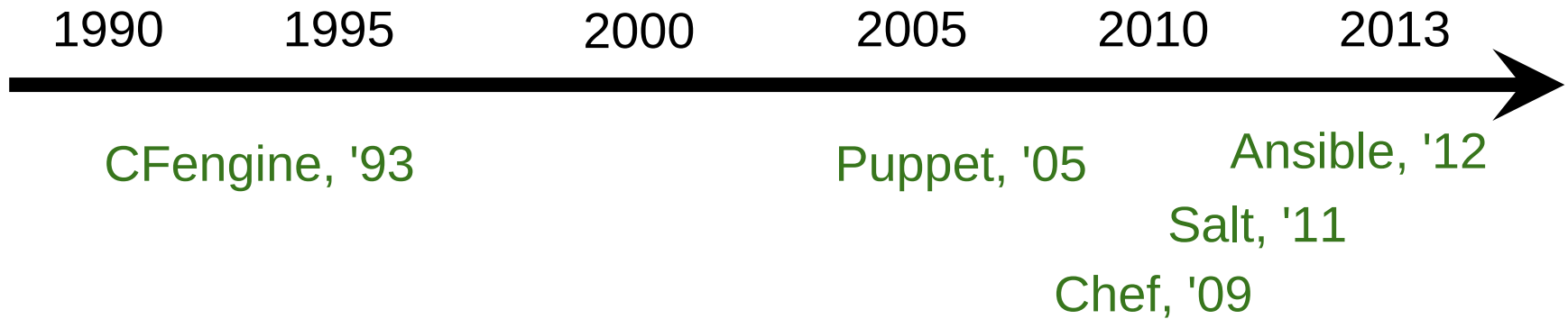
Share

- От разработчиков к операторам (разработчики вовлечены в поддержку рабочего окружения)
- От операторов к разработчикам (обратная связь об ошибках)

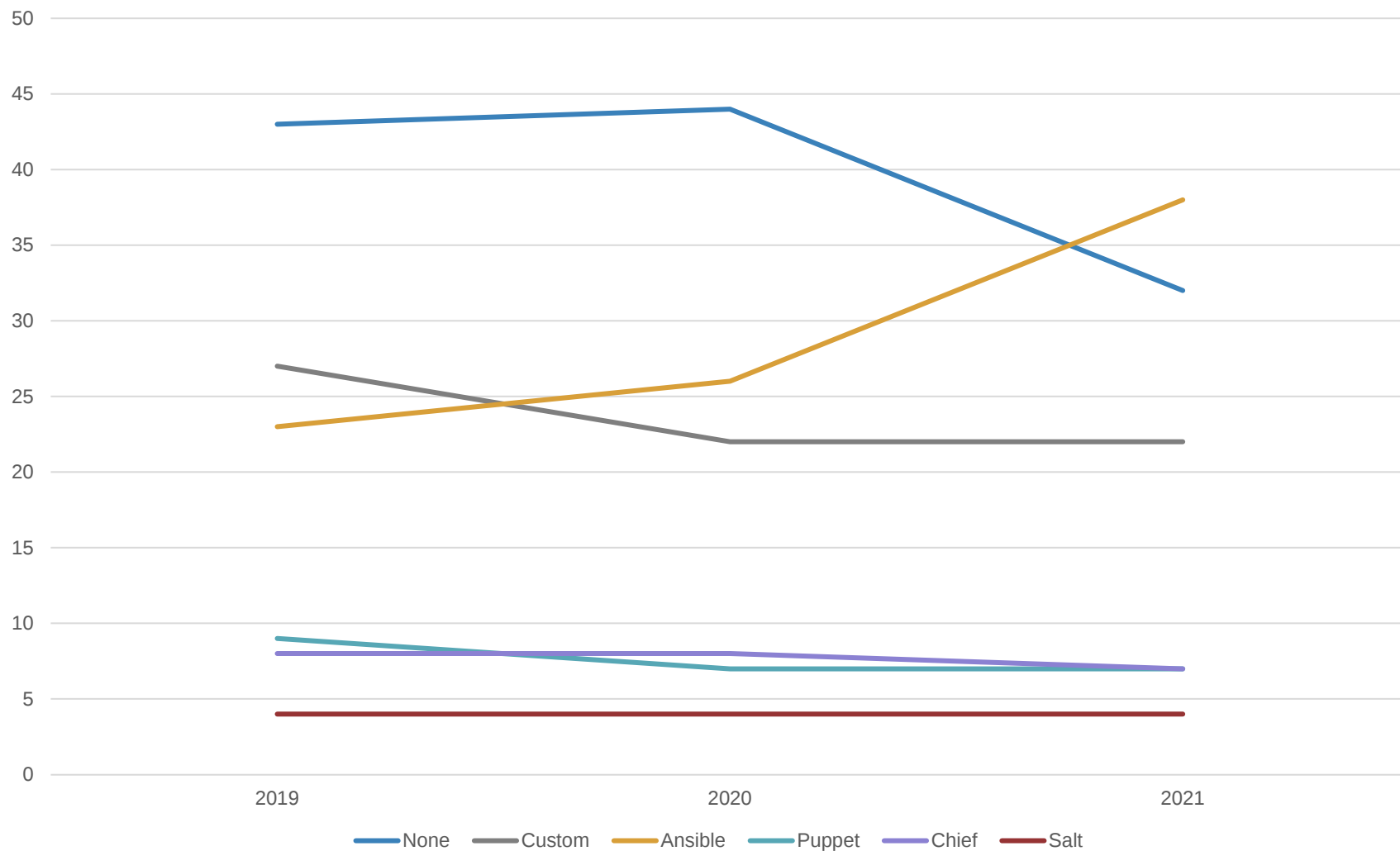
Continuous Integration Maturity Model

Initial	Процессы непредсказуемые, слабо контролируемые, реактивные	VCS, ночные сборки, багтрекер
Managed	Процессы на уровне проектов, реактивные	Build on commit, aut. deploy to dev, unit tests, static code analysis, documentation generation
Defined	Процессы на уровне организации, проактивные	aut. deploy on test, on demand deploy to prod, project metrics
Quantitatively Managed	Процессы измеряются и контролируются	aut. func. tests, aut. deploy to prod, environment metrics, alerts
Optimizing	Фокус на совершенствовании процессов	cont. deploy to prod.

Configuration Management



Usage in % of interviewed developers

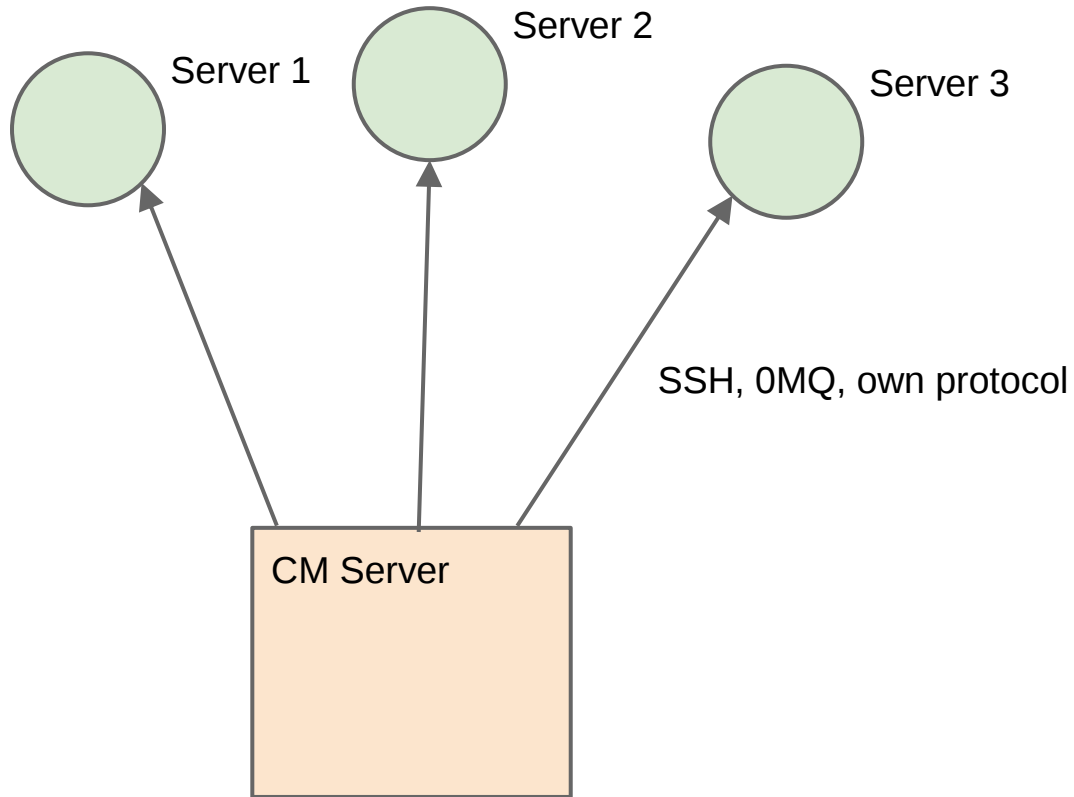


Источник: <https://www.jetbrains.com/lp/devecosystem-2021/devops/>

Общие черты

- push/pull модель
- DSL, часто на ruby
- Переиспользуемые рецепты (recipe, playbooks)
- Декларативный стиль (мы описываем что хотим получить в итоге, а как этого достичь решает инструмент)

push



Chef-solo, Ansible

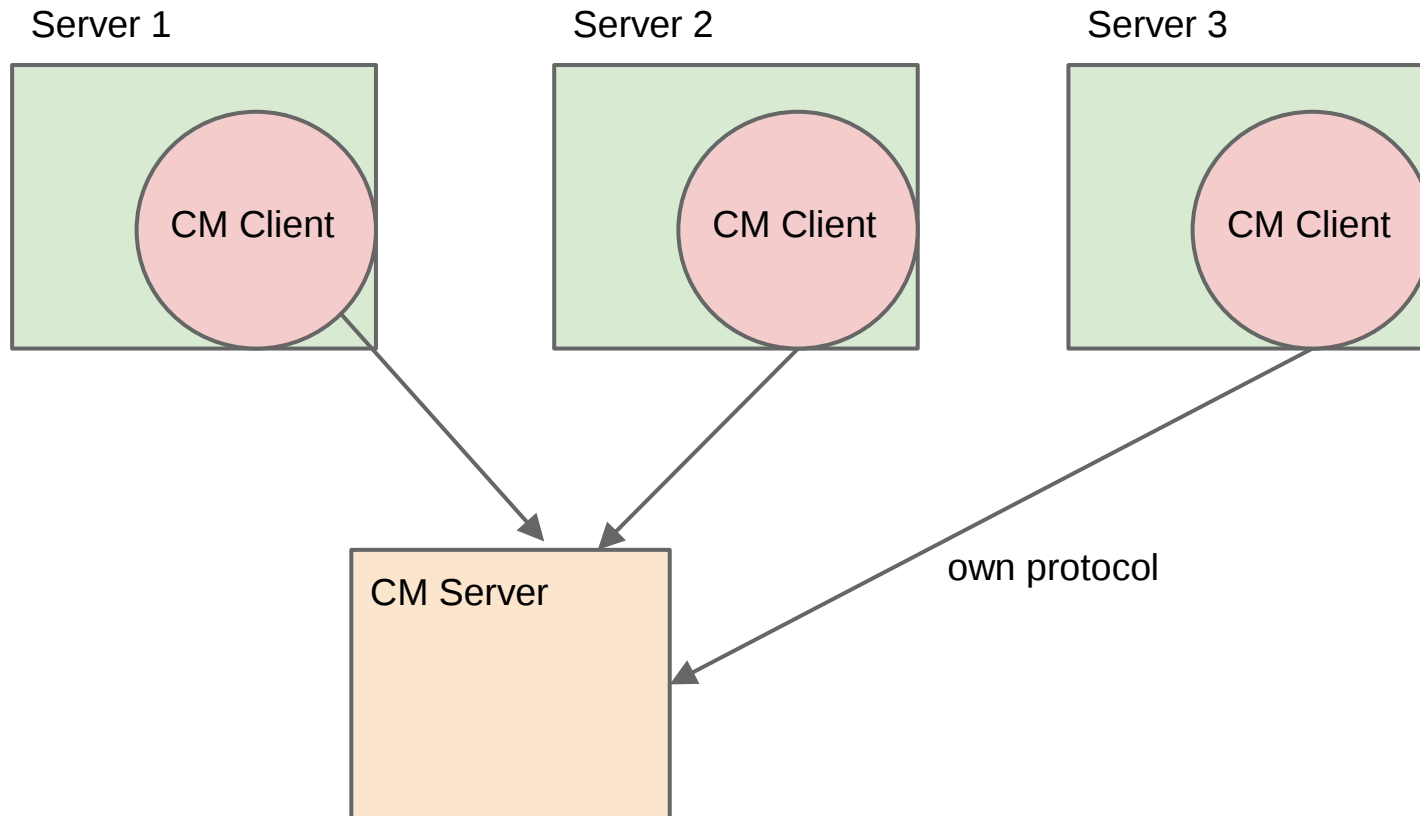
Достоинства Push модели

- Не требует установки дополнительного ПО на серверах
- Более привычно для пользователей Linux

Недостатки Push модели

- Зависит от качества соединения
- Требуется sudo
- Требуется доступа на все управляемые сервера

pull



Chef, Puppet, Salt, Ansible, CF engine

Достоинства Pull модели

- Централизованное управление
- Горизонтальное масштабирование
- Не требует дополнительных пользователей на сервере

Недостатки Pull модели

- Требуется первоначальной установки клиента
- Централизованное управление
- Непрозрачно для пользователей

Ansible facts

- Новый (первый релиз 2012)
- написан на Python
- Простой (годится даже для разработчиков)
- Push based
- Работает через SSH
- Использует YAML в качестве языка рецептов
- one-man software

Ansible example

setup-user-env.yml

```
- hosts: ubuntu
  sudo: yes
  tasks:
    - name: Upload .bashrc file with colorful prompt
      action: copy src=files/bash/.bashrc dest=~/.bashrc
    - apt: pkg=vim
    - apt: pkg=iotop
    - apt: pkg=htop
    - apt: pkg=curl
```

hosts.list

```
[ubuntu]
dev001.example.com
dev002.example.com
```

```
#ansible-playbook --inventory-file=hosts.list setup-user-env.yml
```

Ansible example

```
- hosts: '{{ hosts }}'
  user: ansible
  become: true
  max_fail_percentage: 10

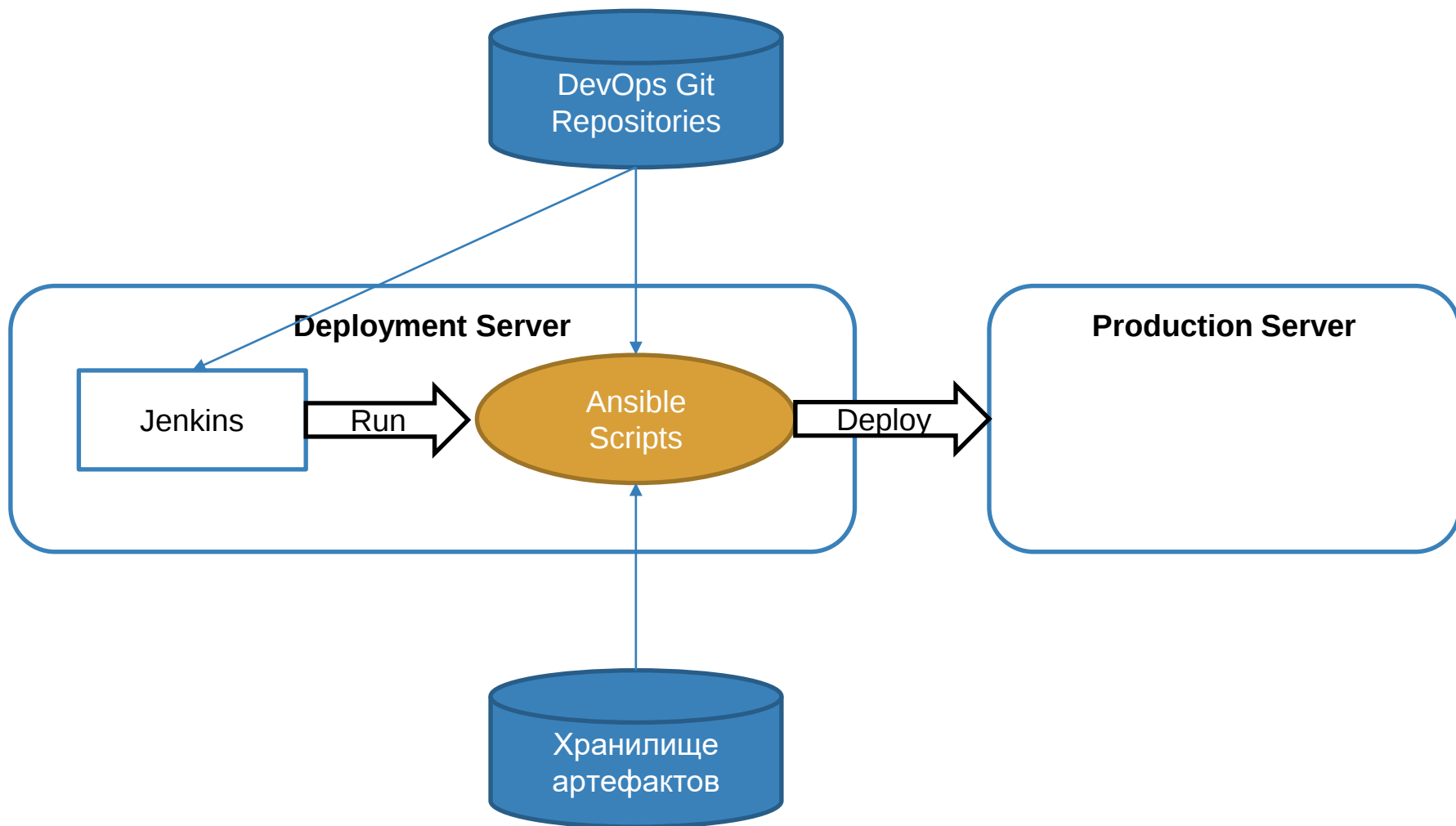
vars:
  frontendOnly: false
  serialNumber: "{{ serialNumberSet | default(2) }}"
  javaHome: "/usr/java/jdk-15.0.2"

serial: '{{ serialNumber }}'

roles:
  - { role: node-exporter }
  - { role: aerospike-bidder }

tasks:
  - name: Prepare directory /usr/java
    file:
      path: '/usr/java'
      state: directory
      owner: root
      group: root
      mode: 0755

  - name: Get OpenJDK
    get_url:
      url: 'https://personartrb-files.s3.amazonaws.com/openjdk-{{ java_version | default("15.0.2")
    }}_linux-{{ java_arch | default("x64") }}_bin.tar.gz'
      dest: '/tmp/java.tar.gz'
      mode: 0444
      force: yes
```



Salt facts

- Сравнительно новый (2011)
- Написан на Python
- Push based
- Для рецептов используется YAML
- Простой



Chef facts

- Зрелый (2009)
- написан на Ruby + Erlang
- Для рецептов используется DSL на Ruby
- pull/push based
- Hosted chef (платный)
- Отличная библиотека рецептов
- Огромное сообщество
- Сложный

Puppet facts

- Старый и зрелый (2005)
- Написан на Ruby
- DSL на Ruby для рецептов
- Огромная библиотека модулей
- Push based
- Есть коммерческая поддержка
- Сложный

Amazon Web Services (AWS)

Проблема:

Большие проекты требуют сложного окружения

Решение:

Виртуальные машины в облаке

Дополнительно:

- Несколько дата-центров
- *RDS – Relational Database Service*
- *S3 – Simple Storage Service*
- *Redshift* - data warehouse service
- *Batch* – пакетная обработка данных
- и многое другое ...

Amazon Web Services (AWS)

**Services** ^ Resource Groups v

History

EC2

Console Home

Cost Explorer

Billing

CloudWatch

IAM

Find a service by name or feature (for example, EC2, S3 or VM, storage).

Group A-Z

**Compute**

EC2

Lightsail 

Elastic Container Service

Lambda

Batch

Elastic Beanstalk

**Storage**

S3

EFS

Glacier

Storage Gateway

**Database**

RDS

DynamoDB

ElastiCache

Amazon Redshift

**Migration**

AWS Migration Hub

Application Discovery Service

Database Migration Service

Server Migration Service

Snowball

**Networking & Content Delivery**

**Developer Tools**

CodeStar

CodeCommit

CodeBuild

CodeDeploy

CodePipeline

Cloud9

X-Ray

**Management Tools**

CloudWatch

CloudFormation

CloudTrail

Config

OpsWorks

Service Catalog

Systems Manager

Trusted Advisor

Managed Services

**Media Services**

Elastic Transcoder

MediaConvert

MediaLive

MediaPackage

MediaStore

MediaTailor

**Machine Learning**

**Analytics**

Athena

EMR

CloudSearch

Elasticsearch Service

Kinesis

Kinesis Video Streams

QuickSight 

Data Pipeline

AWS Glue

**Security, Identity & Compliance**

IAM

Cognito

GuardDuty

Inspector

Amazon Macie 

AWS Single Sign-On

Certificate Manager

CloudHSM

Directory Service

WAF & Shield

Artifact

**Mobile Services**

Mobile Hub

AWS AppSync

Device Farm

Mobile Analytics

**Customer Engagement**

Amazon Connect

Pinpoint

Simple Email Service

**Business Productivity**

Alexa for Business

Amazon Chime 

WorkDocs

WorkMail

**Desktop & App Streaming**

WorkSpaces

AppStream 2.0

**Internet Of Things**

AWS IoT

IoT Device Management

Amazon FreeRTOS

AWS Greengrass

**Game Development**

Amazon GameLift

close

Configurations

AWS Management Console

The screenshot shows the AWS Management Console interface. At the top, there's a navigation bar with the AWS logo, 'Services' dropdown, a search bar, and user information. The left sidebar contains navigation links for 'New EC2 Experience', 'EC2 Dashboard', 'Events', 'Tags', 'Limits', 'Instances', 'Instance Types', 'Launch Templates', 'Spot Requests', 'Savings Plans', 'Reserved Instances', 'Dedicated Hosts', 'Capacity Reservations', 'Images', 'AMIs', 'Elastic Block Store', 'Volumes', 'Snapshots', 'Lifecycle Manager', 'Network & Security', 'Security Groups', 'Elastic IPs', 'Placement Groups', and 'Key Pairs'.

The main content area is titled 'Instances (1/44) Info'. It includes a search bar for filtering instances and a filter button set to 'Instance state: running'. Below this is a table of instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
prod-jp-aerospike-config	i-0a7938141a7f68cb6	Running	c5.large	2/2 checks passed	No alarms	ap-northeast-1a
qa-jp-persona-aerospike-config	i-0081ca5bae2f58cc6	Running	m5a.large	2/2 checks passed	No alarms	ap-northeast-1c
prod-jp-prometheus	i-0bb26459757e7754a	Running	t3a.large	2/2 checks passed	No alarms	ap-northeast-1a
qa-jp-aerospike-cluster1	i-028064377c81be57f	Running	t3a.small	2/2 checks passed	No alarms	ap-northeast-1a
prod-jp-aerospike-cluster-NEW1	i-063105ff49cb8a8a9	Running	r5d.xlarge	2/2 checks passed	No alarms	ap-northeast-1a
prod-jp-aerospike-cluster-NEW3	i-062f09827133bf91f	Running	r5d.xlarge	2/2 checks passed	No alarms	ap-northeast-1a
prod-jp-aerospike-cluster-NEW2	i-058b3add5e7d33e15	Running	r5d.xlarge	2/2 checks passed	No alarms	ap-northeast-1a
prod-jp-aerospike-cluster-NEW4	i-0f540512e0b139bbb	Running	r5d.xlarge	2/2 checks passed	No alarms	ap-northeast-1a

The instance 'prod-jp-aerospike-cluster-NEW4' is selected. A context menu is open over it, showing actions: 'Launch instances', 'Launch instance from template', 'Connect', 'Stop instance', 'Start instance', 'Reboot instance', 'Hibernate instance', 'Terminate instance', 'Instance settings', 'Networking', 'Security', 'Image and templates', and 'Monitor and troubleshoot'.

The details panel for the selected instance shows the following information:

- Instance summary** Info
- Instance details** Info
- Platform**: Linux/UNIX (Inferred)
- AMI ID**: ami-08...
- AMI name**: prod-jp...
- AMI location**: ...
- Launch time**: ...
- Monitoring**: disabled
- Termination protection**: Enabled
- Lifecycle**: ...

At the bottom of the console, there's a footer with 'Feedback', 'English (US)', and copyright information: '© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.' along with links to 'Privacy Policy', 'Terms of Use', and 'Cookie preferences'.

AWS New Instance Launch

aws

Services ▾

Q

Search for services, features, marketplace products, and docs

[Alt+S]

 aden @ ▾ Tokyo ▾ Support ▾

1. Choose AMI

2. Choose Instance Type

3. Configure Instance

4. Add Storage

5. Add Tags

6. Configure Security Group

7. Review

Step 2: Choose an Instance Type

Currently selected: t2.micro (- ECUs, 1 vCPUs, 2.5 GHz, -, 1 GiB memory, EBS only)

	Family ▾	Type ▾	vCPUs ⓘ ▾	Memory (GiB) ▾	Instance Storage (GB) ⓘ ▾	EBS-Optimized Available ⓘ ▾	Network Performance ⓘ ▾	IPv6 Support ⓘ ▾
☐	t2	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
☒	t2	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes
☐	t2	t2.small	1	2	EBS only	-	Low to Moderate	Yes
☐	t2	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
☐	t2	t2.large	2	8	EBS only	-	Low to Moderate	Yes
☐	t2	t2.xlarge	4	16	EBS only	-	Moderate	Yes
☐	t2	t2.2xlarge	8	32	EBS only	-	Moderate	Yes
☐	t3	t3.nano	2	0.5	EBS only	Yes	Up to 5 Gigabit	Yes
☐	t3	t3.micro	2	1	EBS only	Yes	Up to 5 Gigabit	Yes
☐	t3	t3.small	2	2	EBS only	Yes	Up to 5 Gigabit	Yes
☐	t3	t3.medium	2	4	EBS only	Yes	Up to 5 Gigabit	Yes
☐	t3	t3.large	2	8	EBS only	Yes	Up to 5 Gigabit	Yes
☐	t3	t3.xlarge	4	16	EBS only	Yes	Up to 5 Gigabit	Yes
☐	t3	t3.2xlarge	8	32	EBS only	Yes	Up to 5 Gigabit	Yes
☐	t3a	t3a.nano	2	0.5	EBS only	Yes	Up to 5 Gigabit	Yes

Cancel

Previous

Review and Launch

Next: Configure Instance Details

Feedback

English (US) ▾

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Privacy Policy

Terms of Use

Cookie preferences

AMI – Amazon Machine Images

The screenshot shows the AWS Management Console interface for Amazon Machine Images (AMIs). The top navigation bar includes the AWS logo, 'Services', 'Resource Groups', and user information. The left sidebar contains a navigation menu with categories like INSTANCES, IMAGES, ELASTIC BLOCK STORE, NETWORK & SECURITY, LOAD BALANCING, and AUTO SCALING. The main content area displays a list of AMIs, with 'Test Ubuntu' selected. A context menu is open over the selected AMI, showing options like Launch, Spot Request, Deregister, Register New AMI, Copy AMI, Modify Image Permissions, Add/Edit Tags, and Modify Boot Volume Setting. The details section for the selected AMI shows the following information:

Property	Value
AMI ID	ami-0c8a1576
Owner	164067767295
Status	available
Creation date	November 30, 2017 at 2:43:10 AM UTC+7
Architecture	x86_64
Virtualization type	paravirtual
AMI Name	Test Ubuntu
Source	164067767295/Test Ubuntu
State Reason	-
Platform	Ubuntu
Image Type	machine
Description	-

AWS Marketplace

aws marketplace

AMI & SaaS



Hello, asteriosoft

View Categories

Sell in AWS Marketplace

Amazon Web Services Home

Help

Categories

All Categories

Software Infrastructure (2851)

Developer Tools (740)

Business Software (1894)

Filters

Operating System

+ All Windows

+ All Linux/Unix

Software Pricing Plans

☐ Free (625)

☐ Hourly (1971)

☐ Monthly (96)

☐ Annual (1172)

☐ Bring Your Own License (450)

[Show more](#)

Support

☐ Product Support Connection (103)

Software Free Trial

☐ Free Trial (788)

Delivery Method

☐ Amazon Machine Image (3159)

☐ CloudFormation Stack (236)

☐ SaaS (792)

Average Rating

★★★★★ & up (348)

★★★★★ & up (426)

★★★★★ & up (470)

★★★★★ & up (528)

Architecture

☐ 32-bit (17)

☐ 64-bit (3142)

Region

☐ US East (N. Virginia) (3138)

☐ US East (Ohio) (2549)

All Categories (3951 results) showing 1 - 10

1 2 3 4 5 ... 395



CentOS 7 (x86_64) - with Updates HVM

★★★★★ (49) | Version 1708_11 | Sold by [Centos.org](#)

This is the Official CentOS 7 x86_64 HVM image that has been built with a minimal profile, suitable for use in HVM instance types only. The image contains just enough packages...

Linux/Unix, CentOS 7 - 64-bit Amazon Machine Image (AMI)



WordPress Certified by Bitnami

★★★★★ (21) | Version 4.9-0 on Ubuntu 14.04 | Sold by [Bitnami](#)

Wordpress powers over 25% of all websites on the internet, making it the world's most popular blogging and content management platform. It is free and open source software...

Linux/Unix, Ubuntu 14.04 - 64-bit Amazon Machine Image (AMI)



CentOS 6 (x86_64) - with Updates HVM

★★★★★ (33) | Version 1704 | Sold by [Centos.org](#)

This is the Official CentOS 6 x86_64 HVM image that has been built with a minimal profile. The image contains just enough packages to run within AWS, bring up an SSH Server...

Linux/Unix, CentOS 6 - 64-bit Amazon Machine Image (AMI)



CentOS 6.5 (x86_64) - Release Media

★★★★★ (55) | Version 6.5 - 2013-12-01 | Sold by [CentOS.org](#)

This is the Official CentOS 6.5 x86_64 image that has been built with a minimal profile. The image contains just enough packages to run within AWS, bring up an SSH Server...

Linux/Unix, CentOS 6.5 - 64-bit Amazon Machine Image (AMI)



Debian GNU/Linux 8 (Jessie)

★★★★★ (86) | Version 8.7 | Sold by [Debian](#)

Debian is a computer operating system composed of software packages released as free and open source software primarily under the GNU General Public License along with other...

Linux/Unix, Debian 8.6+1 - 64-bit Amazon Machine Image (AMI)



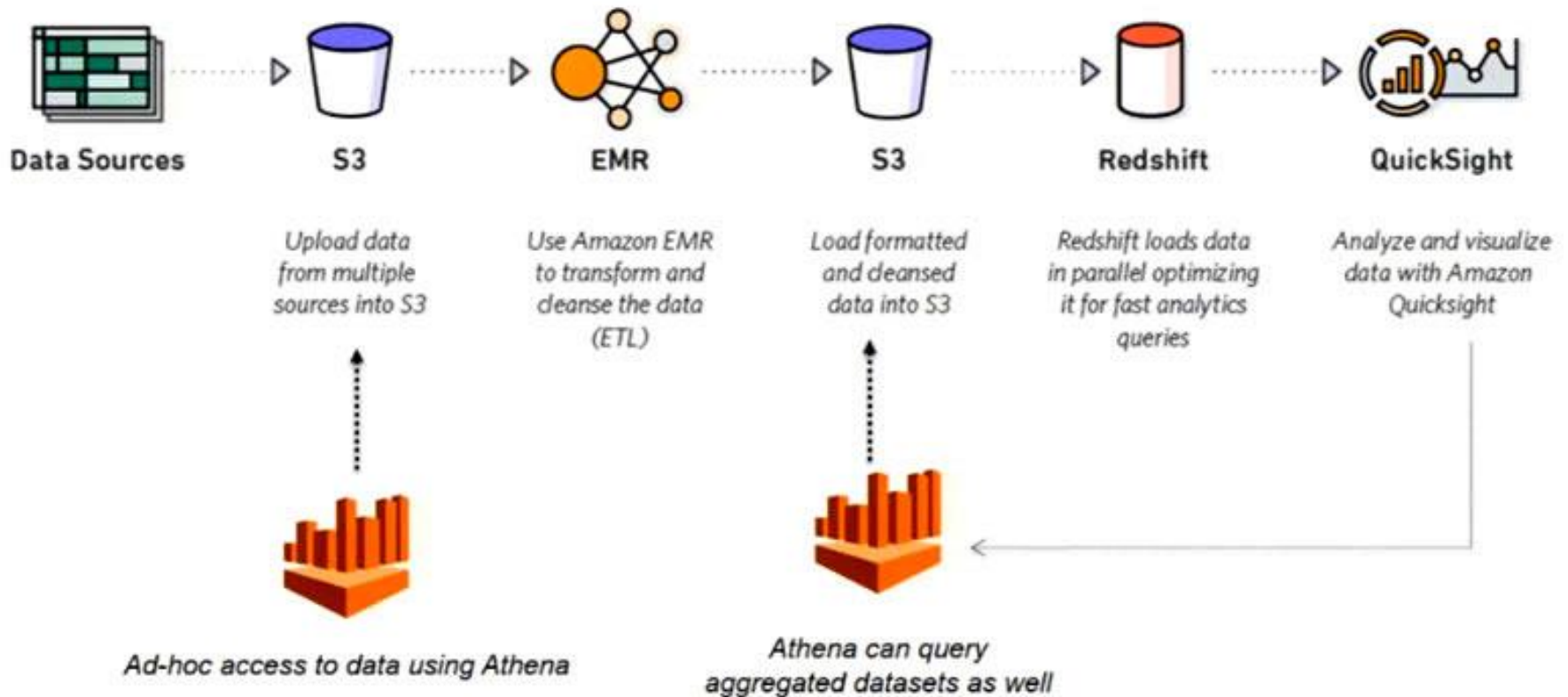
OpenVPN Access Server

★★★★★ (30) | Version 2.1.9 | Sold by [OpenVPN Technologies, Inc.](#)

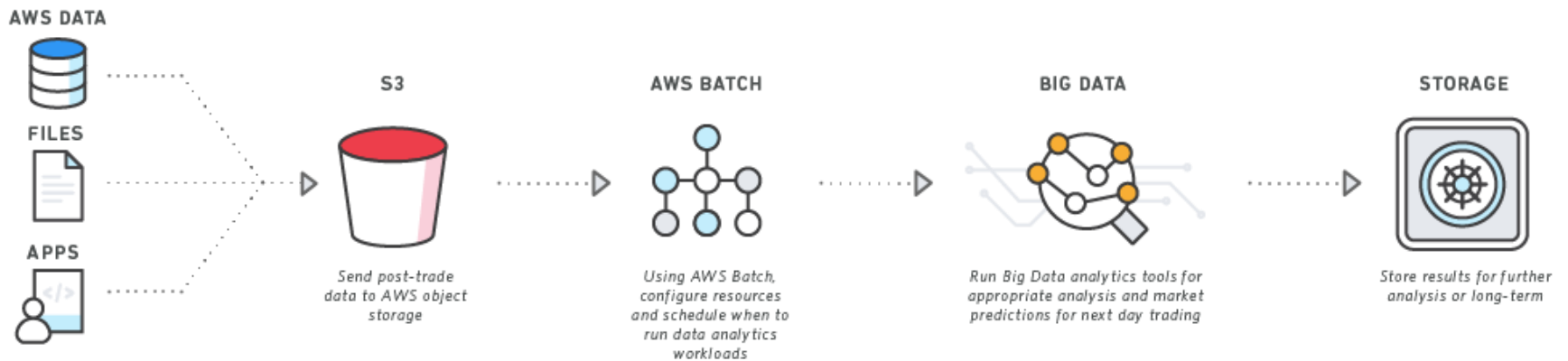
OpenVPN Access Server is a full featured SSL VPN software solution that integrates OpenVPN

AWS Athena & AWS Redshift

A Sample Pipeline



AWS Batch



Управление Spots



Elastigroup

Dashboard

Spot Analyzer

Elastigroups

Idle Resources

Master

Ocean

Cloud Clusters

On Prem Clusters

Elastigroup

Use Cases

General

Compute

Scaling

Review

USE CASES

Select a use case template to create your Elastigroup

GENERAL

Auto Scaling Group

Load Balancer

Elastic Beanstalk

Ops Works

Route 53

CI/CD

CodeDeploy

Chef

STATEFUL

Stateful

KUBERNETES

Kubernetes
Join Existing Cluster

EKS
Create New Cluster

EKS
Join Existing EKS

KOPS

BIG DATA AND DB

EMR

CONTAINERS & MICRO SERVICES

ECS
Create New ECS

ECS
Join Existing ECS

Docker Swarm

Rancher

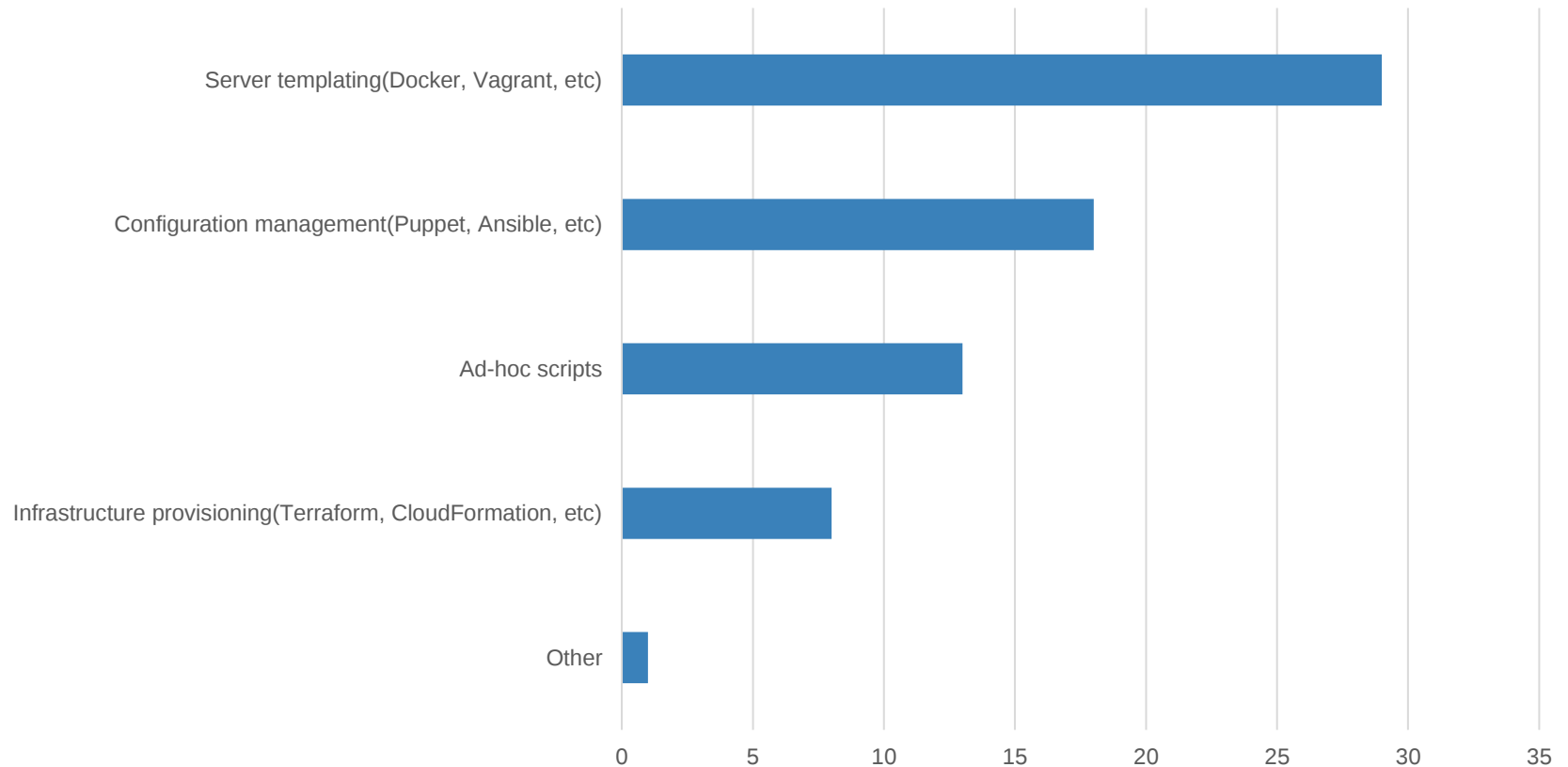
Nomad

Mesosphere

Empty Template

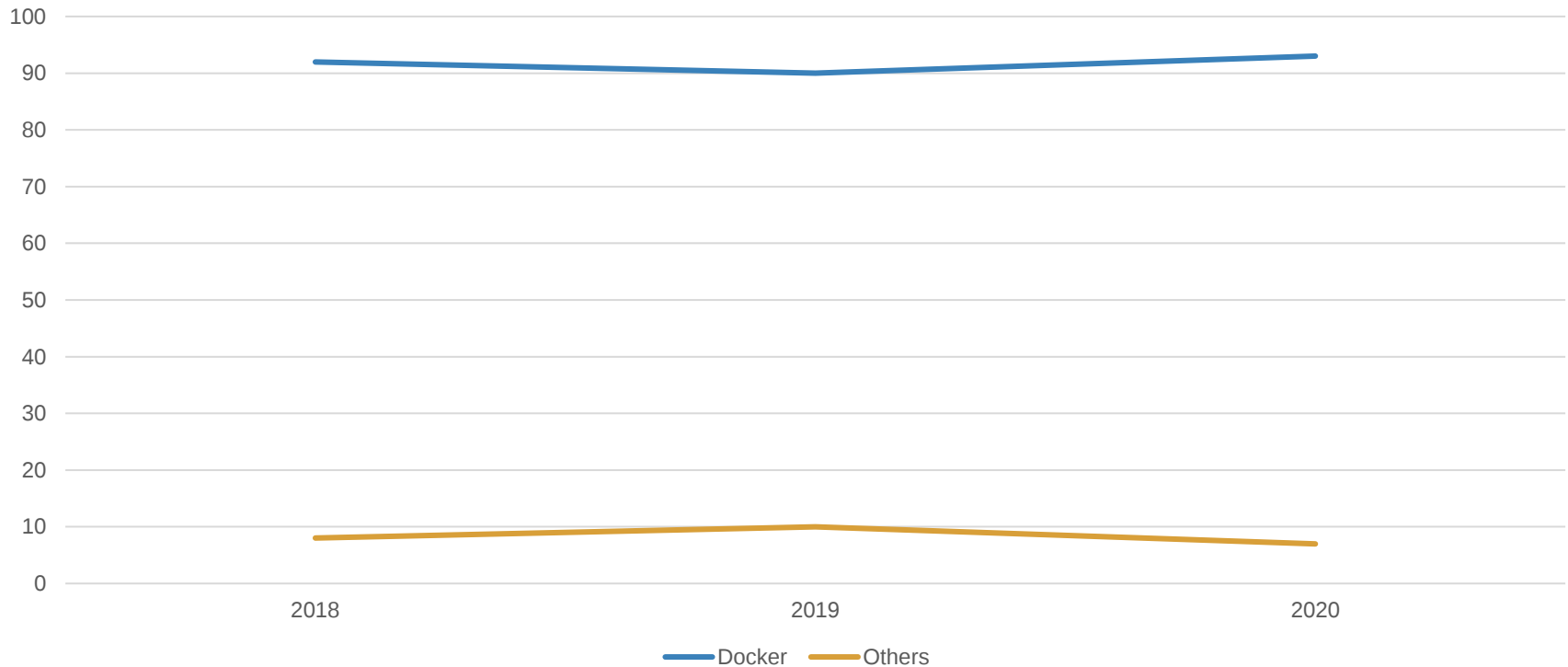
Infrastructure-As-a-Code tools

Usage in % of interviewed developers, 2018



Источник: <https://www.jetbrains.com/research/devecosystem-2018/devops/>

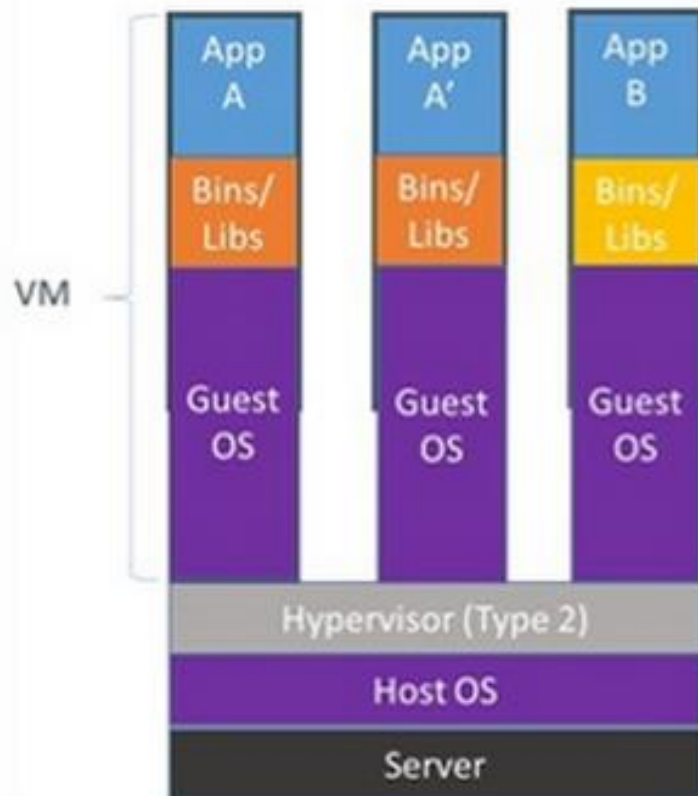
Server templating tools usage (%)



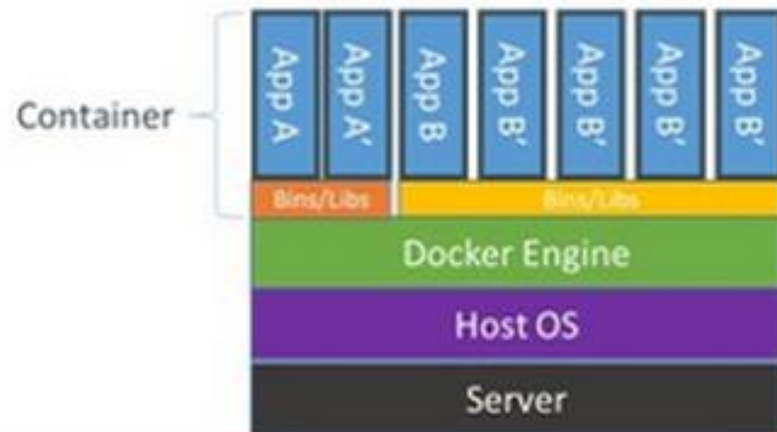
Источник: <https://www.jetbrains.com/research/devecosystem-2018/devops/>

Docker

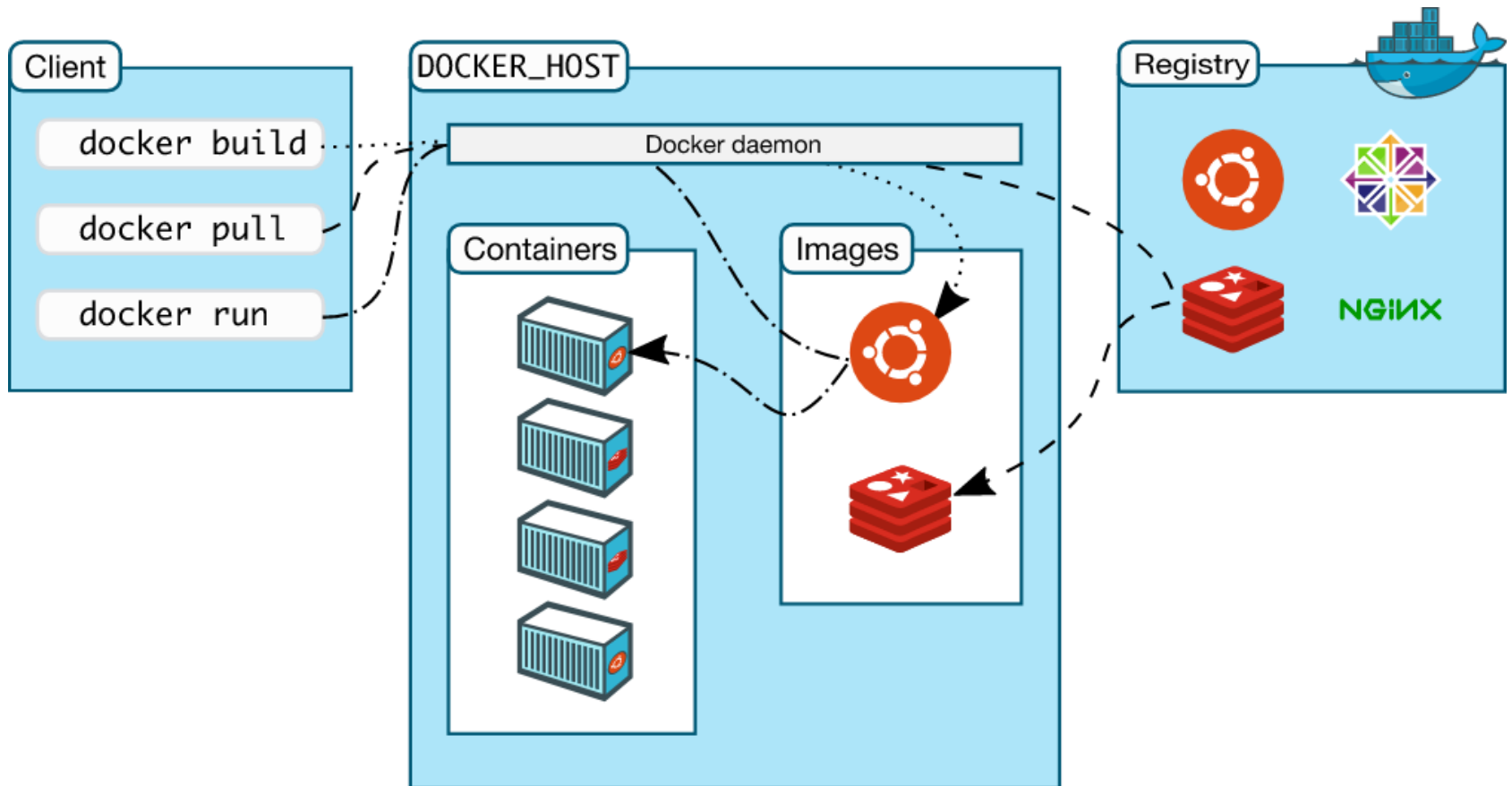
Containers vs. VMs



Containers are isolated, but share OS and, where appropriate, bins/libraries



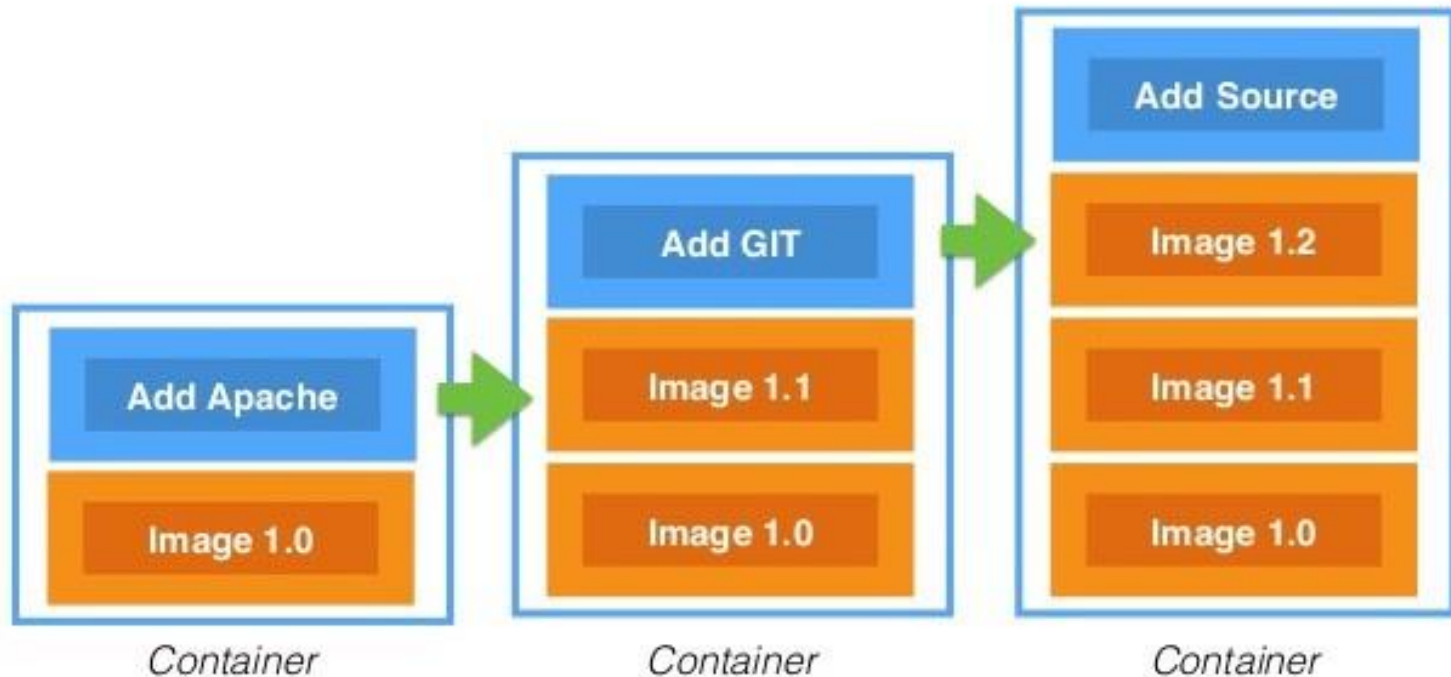
Docker



Docker Layers

AuFS
Layered Filesystem

2019



Example: Run Aerospike image with custom version/config/port

```
docker run -d -v C:\Projects\docker-  
files\aerospike:/opt/aerospike/etc --name  
aerospike-jarsoft -p 3000:3000 -p 3001:3001 -p  
3002:3002 aerospike:4.8.0.2 asd --foreground -  
-config-file /opt/aerospike/etc/aerospike-  
jarsoft.conf
```

Example: Redis Cluster

FROM redis:5.0.6

MAINTAINER aden@jarsoft.ru

COPY redis0.conf /etc/redis0.conf

COPY redis1.conf /etc/redis1.conf

COPY redis2.conf /etc/redis2.conf

COPY redis3.conf /etc/redis3.conf

COPY redis4.conf /etc/redis4.conf

COPY redis5.conf /etc/redis5.conf

RUN mkdir -p /var/log/redis

RUN chmod 0777 /var/log/redis

RUN mkdir -p /var/lib/redis

RUN chmod 0777 /var/lib/redis

COPY redis-cluster.sh /etc/redis-cluster.sh

RUN chmod +x /etc/redis-cluster.sh

CMD /etc/redis-cluster.sh

EXPOSE 7000

EXPOSE 7001

EXPOSE 7002

EXPOSE 7003

EXPOSE 7004

EXPOSE 7005

```
..  
buildRedisCluster.cmd  
Dockerfile  
redis-cluster.sh  
redis0.conf  
redis1.conf  
redis2.conf  
redis3.conf  
redis4.conf  
redis5.conf  
runRedisCluster.cmd
```

Example: Redis Cluster

```
#!/bin/sh
```

```
echo "start cluster"
```

```
nohup redis-server /etc/redis0.conf > /var/log/redis/redis0.log &  
nohup redis-server /etc/redis1.conf > /var/log/redis/redis1.log &  
nohup redis-server /etc/redis2.conf > /var/log/redis/redis2.log &  
nohup redis-server /etc/redis3.conf > /var/log/redis/redis3.log &  
nohup redis-server /etc/redis4.conf > /var/log/redis/redis4.log &  
nohup redis-server /etc/redis5.conf > /var/log/redis/redis5.log &
```

```
sleep 3
```

```
tail -10 /var/log/redis/redis0.log  
tail -10 /var/log/redis/redis1.log  
tail -10 /var/log/redis/redis2.log  
tail -10 /var/log/redis/redis3.log  
tail -10 /var/log/redis/redis4.log  
tail -10 /var/log/redis/redis5.log
```

```
if [ -f "/etc/redis-init.txt" ]; then  
    echo 'cluster is already init'
```

```
else  
    echo 'yes' | redis-cli --cluster create 127.0.0.1:7000 127.0.0.1:7001 127.0.0.1:7002 127.0.0.1:7003  
127.0.0.1:7004 127.0.0.1:7005 --cluster-replicas 1  
    touch /etc/redis-init.txt  
fi
```

```
tail -f /var/log/redis/redis0.log
```

```
..  
buildRedisCluster.cmd  
Dockerfile  
redis-cluster.sh  
redis0.conf  
redis1.conf  
redis2.conf  
redis3.conf  
redis4.conf  
redis5.conf  
runRedisCluster.cmd
```

Example: Redis Cluster

redis0.conf

port 7000
cluster-enabled yes
cluster-config-file /etc/nodes0.conf
cluster-node-timeout 5000
appendonly yes

redis1.conf

port 7001
cluster-enabled yes
cluster-config-file nodes1.conf
cluster-node-timeout 5000
appendonly yes

redis2.conf

port 7002
cluster-enabled yes
cluster-config-file nodes2.conf
cluster-node-timeout 5000
appendonly yes

```
..  
buildRedisCluster.cmd  
Dockerfile  
redis-cluster.sh  
redis0.conf  
redis1.conf  
redis2.conf  
redis3.conf  
redis4.conf  
redis5.conf  
runRedisCluster.cmd
```

Example: Redis Cluster

buildRedisCluster.cmd

`docker build --tag redis-cluster-jarsoft .`

```
..  
buildRedisCluster.cmd  
Dockerfile  
redis-cluster.sh  
redis0.conf  
redis1.conf  
redis2.conf  
redis3.conf  
redis4.conf  
redis5.conf  
runRedisCluster.cmd
```

buildRedisCluster.cmd

`start docker run -it -p 7000:7000 -p 7001:7001 -p 7002:7002 -p 7003:7003 -p 7004:7004 -p 7005:7005 redis-cluster-jarsoft`

Example: Druid & Kafka & Registry

FROM openjdk:8

ENV DRUID_VERSION 0.19.0

Get Druid

RUN mkdir -p /tmp \

&& cd /tmp/ \

&& curl -fsLS "http://mirror.linux-ia64.org/apache/druid/0.19.0/apache-druid-\${DRUID_VERSION}-bin.tar.gz" | tar xvz \

&& mv apache-druid-\${DRUID_VERSION} /opt/druid

RUN sed -i 's/druid\.extensions\.loadList\.*/druid.extensions.loadList=["druid-hdfs-storage", "druid-kafka-indexing-service", "druid-datasketches", "druid-avro-extensions", "druid-parquet-extensions", "druid-kafka-indexing-service"]]/' /opt/druid/conf/druid/single-server/nano-quickstart/_common/common.runtime.properties

RUN sed -i 's/druid\.worker\.capacity\.*/druid\.worker\.capacity=10/' /opt/druid/conf/druid/single-server/nano-quickstart/middleManager/runtime.properties

RUN mkdir -p /tmp \

&& cd /tmp/ \

&& curl -fsLS "http://packages.confluent.io/archive/5.5/confluent-5.5.1-2.12.tar.gz" | tar xvz \

&& mv confluent-5.5.1 /opt/confluent

RUN sed -i 's/listeners=\.*/listeners=http://0.0.0.0:9081/' /opt/confluent/etc/schema-registry/schema-registry.properties

```
RUN sed -i 's/#advertised.listeners.*/advertised.listeners=PLAINTEXT://localhost:9092/'  
/opt/confluent/etc/kafka/server.properties
```

```
RUN echo "auto.create.topics.enable=true" >> /opt/confluent/etc/kafka/server.properties
```

```
COPY druid-kafka.sh /opt/druid-kafka.sh
```

```
RUN chmod 0755 /opt/druid-kafka.sh
```

```
# Expose ports:
```

```
# - 8888: Druid router
```

```
# - 8081: Druid coordinator/overlord
```

```
# - 8082: Druid broker
```

```
# - 8083: Druid historical
```

```
# - 2181 2888 3888: ZooKeeper
```

```
# - 8090: Kafka supervisor
```

```
# - 9092: Kafka
```

```
# - 9081: Schema-registry (moved from default 8081)
```

```
EXPOSE 8888
```

```
EXPOSE 8081
```

```
EXPOSE 8082
```

```
EXPOSE 8083
```

```
EXPOSE 2181 2888 3888
```

```
EXPOSE 8090
```

```
EXPOSE 9092
```

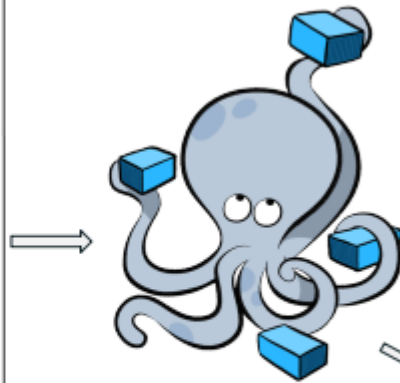
```
EXPOSE 9081
```

```
ENTRYPOINT /opt/druid-kafka.sh
```

Docker Compose

```
docker-compose.yml

version: "3.7"
services:
  db:
    image: mysql:8.0.19
    restart: always
    environment:
      - MYSQL_DATABASE=example
      - MYSQL_ROOT_PASSWORD=password
  app:
    build: app
    restart: always
  web:
    build: web
    restart: always
    ports:
      - 80:80
```



Dockerfile

Dockerfile

mysql

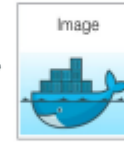


run

db



app



run

app



web



run

web



Docker Compose

```
/usr/local/bin/docker-compose up --abort-on-container-exit --quiet-pull
```

docker-compose.yml

```
version: '2.1'
```

```
services:
```

```
  config:
```

```
    container_name: config
```

```
    image: registry.jarsoft.ru/qa/config:qa-latest
```

```
    ports:
```

```
      - "8010:8010"
```

```
    healthcheck:
```

```
      test: "curl -f http://config:8010/ui/qa"
```

```
  mysql:
```

```
    container_name: mysql
```

```
    image: registry.jarsoft.ru/qa/mysql:5.7
```

```
    ports:
```

```
      - "3306:3306"
```

```
    volumes:
```

```
      - "/data/mysql/data:/var/lib/mysql"
```

```
    healthcheck:
```

```
      test: "/usr/bin/mysql --user=root --password=root --execute \"USE atomik;\""
```

aerospike:

container_name: aerospike

image: aerospike:4.3.0.7

logging:

driver: none

ports:

- "3000:3000"

command: ["asd", "--foreground", "--config-file", "/opt/aerospike/etc/aerospike.conf"]

volumes:

- "/data/aerospike/data:/opt/aerospike/data"

- "/data/aerospike/etc:/opt/aerospike/etc"

- "/data/aerospike/log:/var/log/aerospike"

healthcheck:

test: ["CMD", "asinfo", "-v", "status"]

ui:

container_name: ui

image: registry.jarsoft.ru/qa/ui:qa-latest

ports:

- "8080:8080"

healthcheck:

test: "curl -f http://ui:8080/build-info"

retries: 10

depends_on:

mysql:

condition: service_healthy

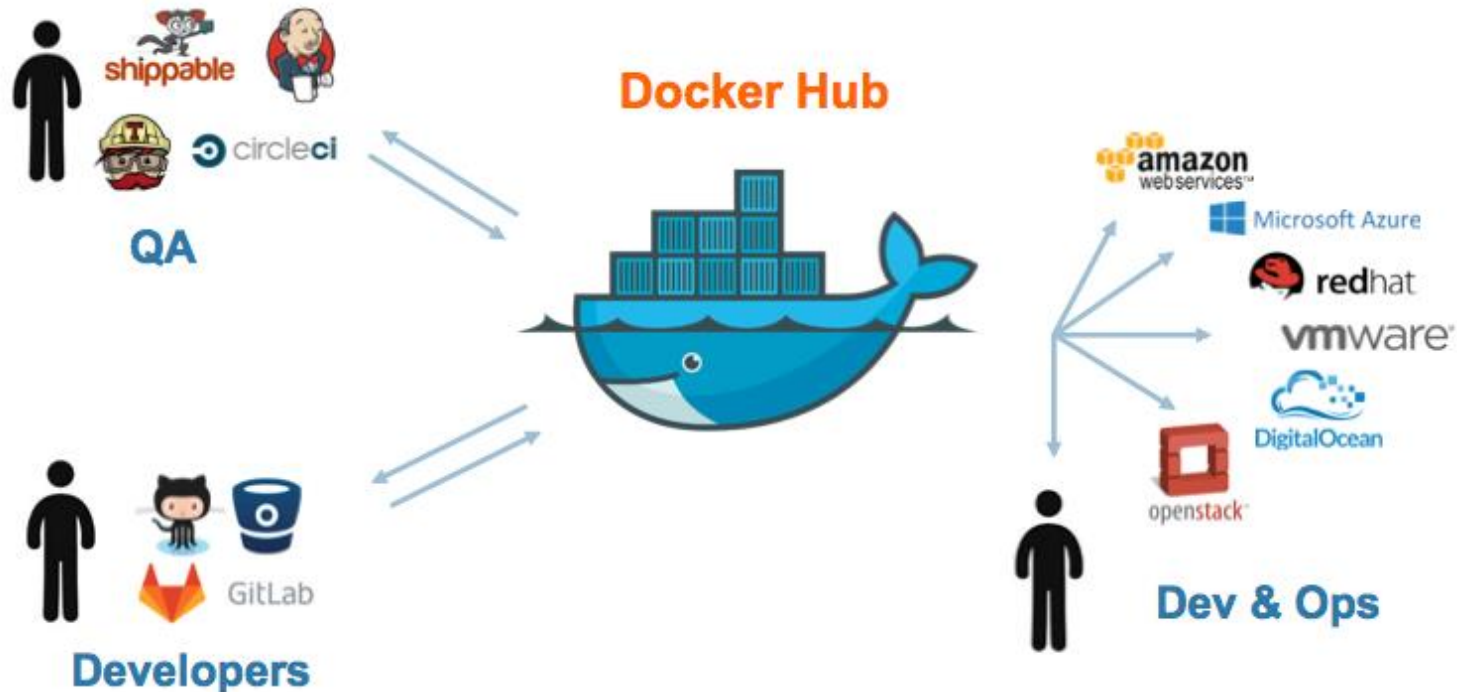
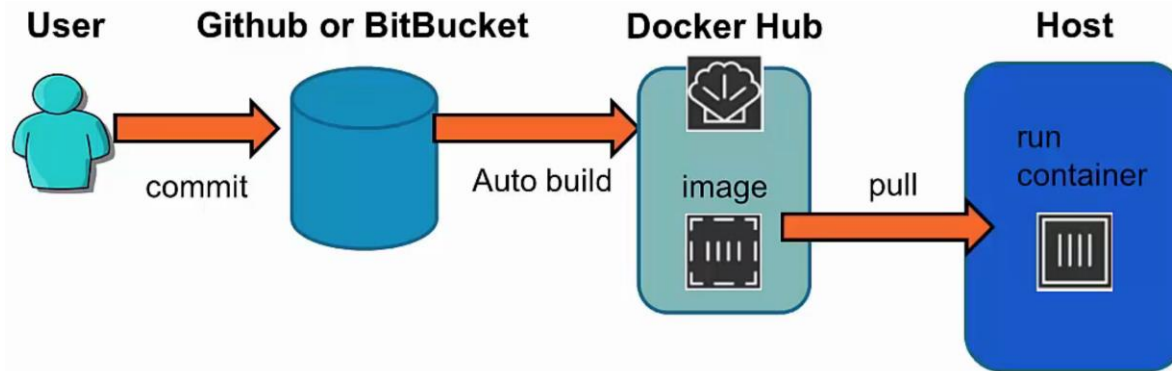
aerospike:

condition: service_healthy

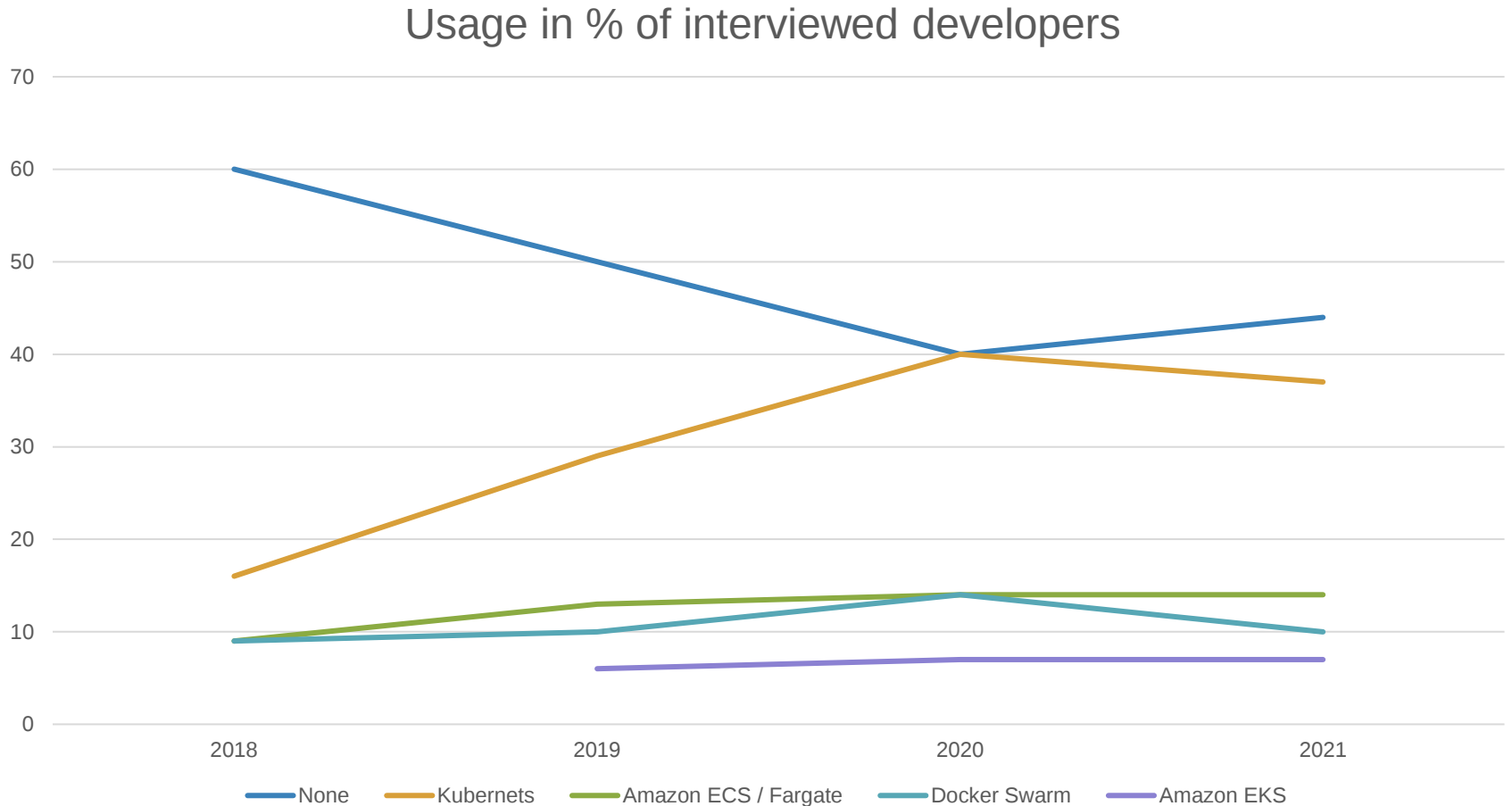
config:

condition: service_healthy

Docker Hub



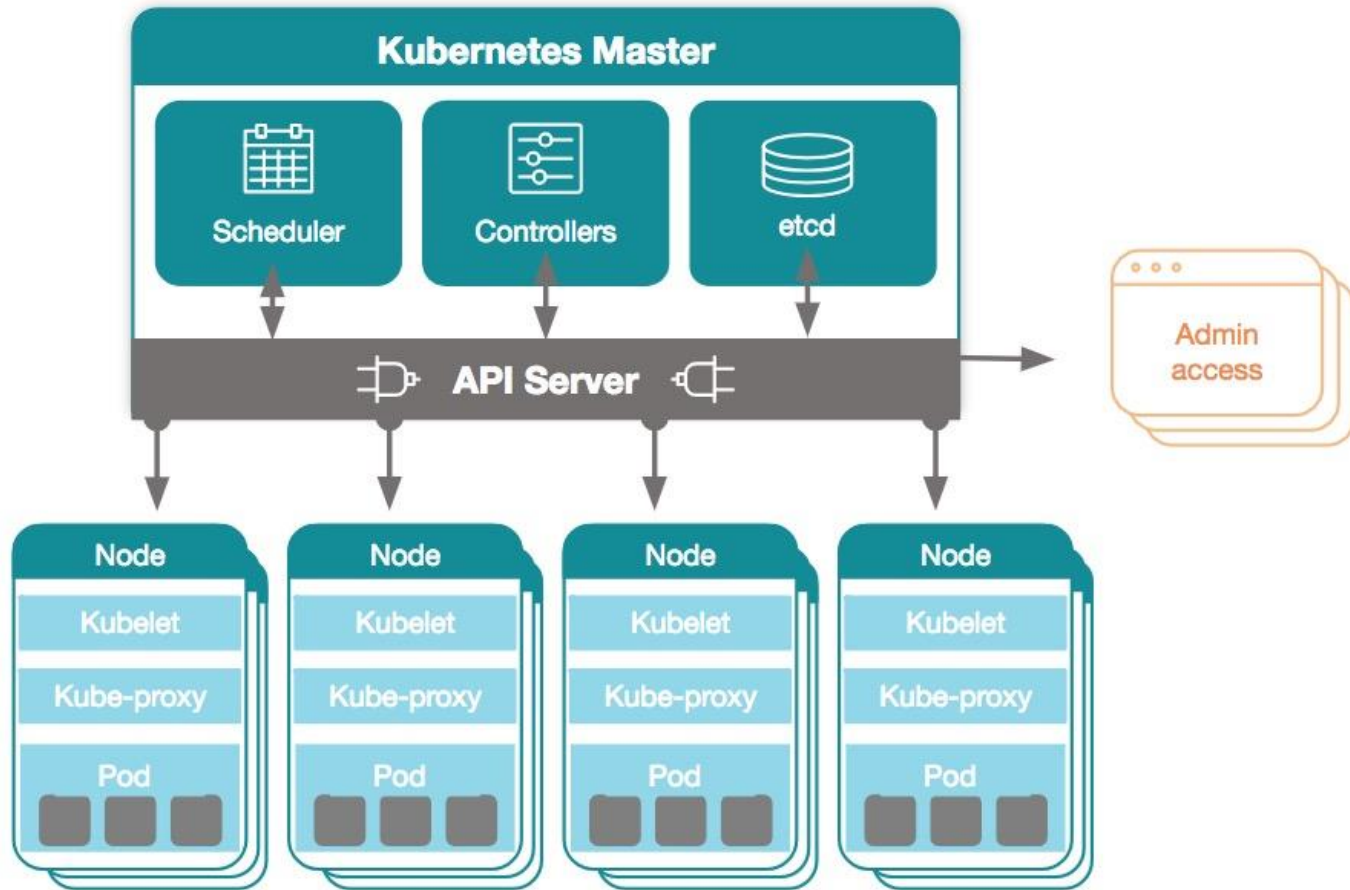
Container orchestration services



Источник: <https://www.jetbrains.com/idea/devecosystem-2021/devops/>

Kubernetes

Kubernetes Architecture



Kubernetes IDE

The screenshot displays the Kubernetes IDE interface, which is a web-based tool for managing Kubernetes clusters. The interface is divided into several sections:

- Left Sidebar:** Contains navigation icons and a list of resources. The 'Workloads' section is expanded, showing 'Overview', 'Pods', 'Deployments', 'DaemonSets', 'StatefulSets', 'Jobs', and 'CronJobs'. The 'Pods' section is selected.
- Top Bar:** Shows the current context as 'NK-Karbon-Cluster-context' and the active view as 'Pods'.
- Pods List:** A table showing a list of pods. The 'cassandra-0' pod is selected, and its details are shown in the right pane.
- Pod Details (cassandra-0):** A detailed view of the selected pod, including its name, namespace, labels, and status. The pod is running on the node 'karbon-nk-karbon-cluster-cd5492-k8s-worker-0'.
- Metrics:** A graph showing the pod's resource usage (CPU, Memory, Network, Filesystem) over time. The graph shows that the pod is using approximately 0.3 CPU and 0.3 GB of memory.
- Terminal:** A terminal window at the bottom showing logs for the 'cassandra-0' pod. The logs indicate that the pod is running successfully and has received streaming plans.

The interface is designed to provide a comprehensive view of the Kubernetes cluster, allowing users to monitor and manage their workloads effectively.

Рекомендуемая литература

- <http://www.opscode.com/chef/>
- <http://saltstack.com/>
- <http://ansible.cc>
- <http://www.cfengine.com/>
- <http://www.puppetlabs.com/>
- <http://vagrantup.com/>