Microservices & Continuous Integration

http://www.bennysbaker.com/poop-emoji-cupcakes/

Makeomatic Team, 13.07.2016.

https://makeomatic.co

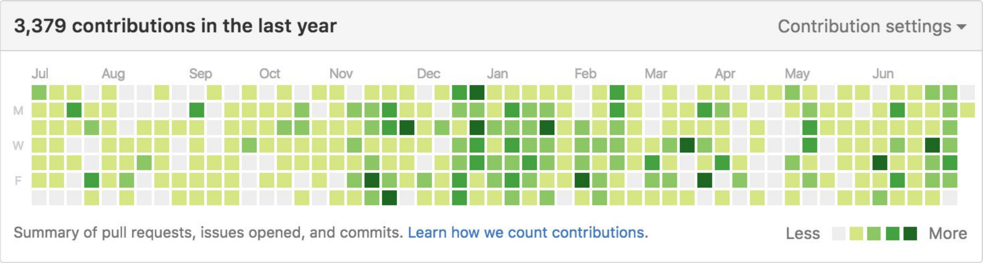
Who Am I

- Vitaly Aminev
- CTO & Founder of Makeomatic
- No CS degree
- DIY enthusiast
- Node.js Evangelist

Follow Right Now:

twitter: @v_aminev
github: @avvs

- Workaholic



3,379 contributions in the last year

Summary of pull requests, issues opened, and commits. Learn how we count contributions.

Typical Project

- Idea & seed $
- MVP
- progressive enhancement
- up to 80% of generic functionality

Common requirements

- Scalability
- Fast iterations
- Time
- Quality

Microservices to the rescue

- Isolated features
- Granular scalability
- Versioning
- Robustness of overall system
- Reusable code
- Composability in “lego” like style
- Easier isolated testing

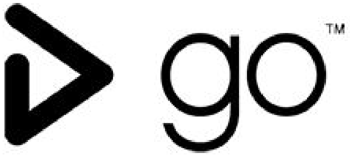
Not a Holy Grail

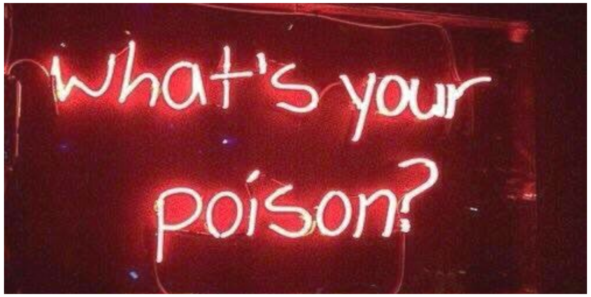
- Complexity of integrational testing
- Coupling
- Versioning
- Configurations
- Monitoring
- Service Discovery
- Orchestration
- Delivery to Environments

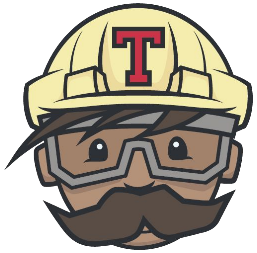
Our Workflow

1. Code → CVS → CI
2. Static Analysis
3. Unit Testing
4. Packaging
5. Integration Testing
6. Storing Artifacts
7. [deploy to staging]
8. [manual QA]
9. Update configurations
10. Shared Tests
11. Release

Continuous integration

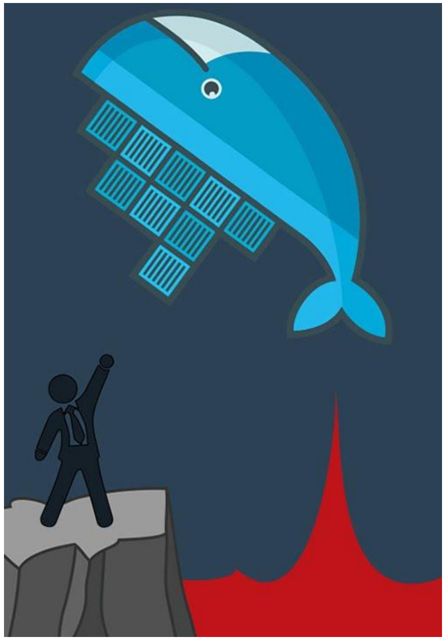






Tests

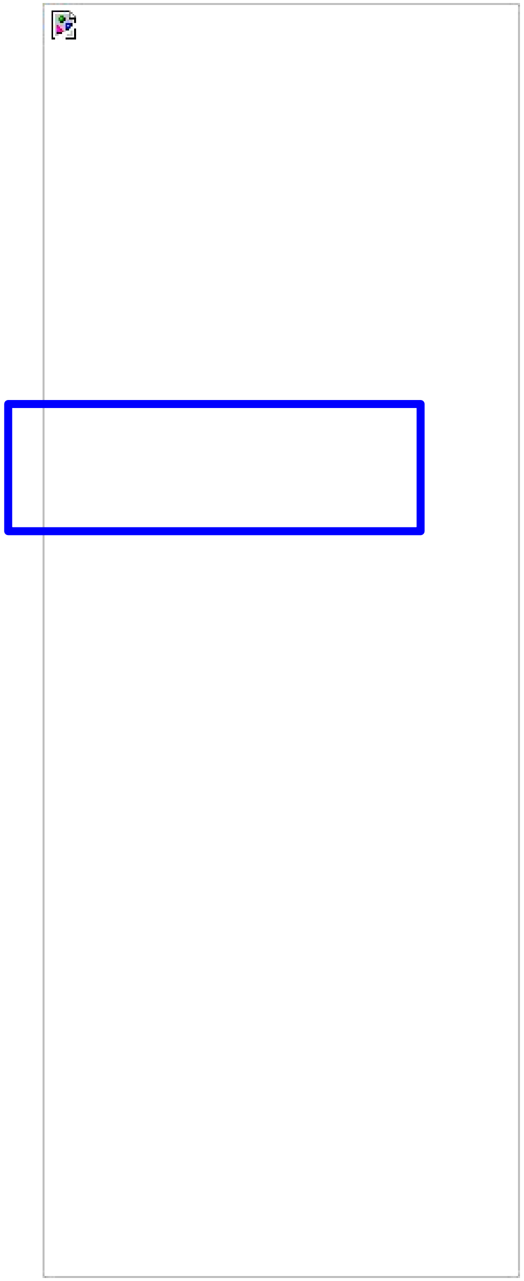
- **Static**: eslint, XO
- **Unit, Integration, Acceptance**
- **Code Review**



Packaging

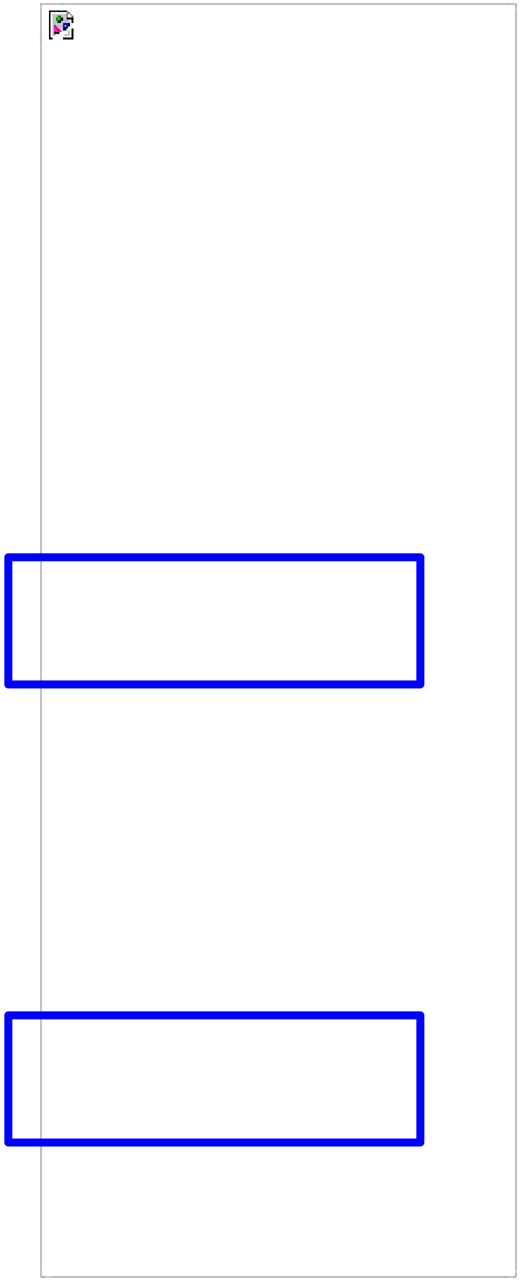
- Build production-ready docker or any other image.
- Upload to docker registry: public for OSS and private for IP.
- In case of other types of images use S3-like providers for storage

Alternative: Packer by Hashicorp as a higher-level packaging API



Provisioning

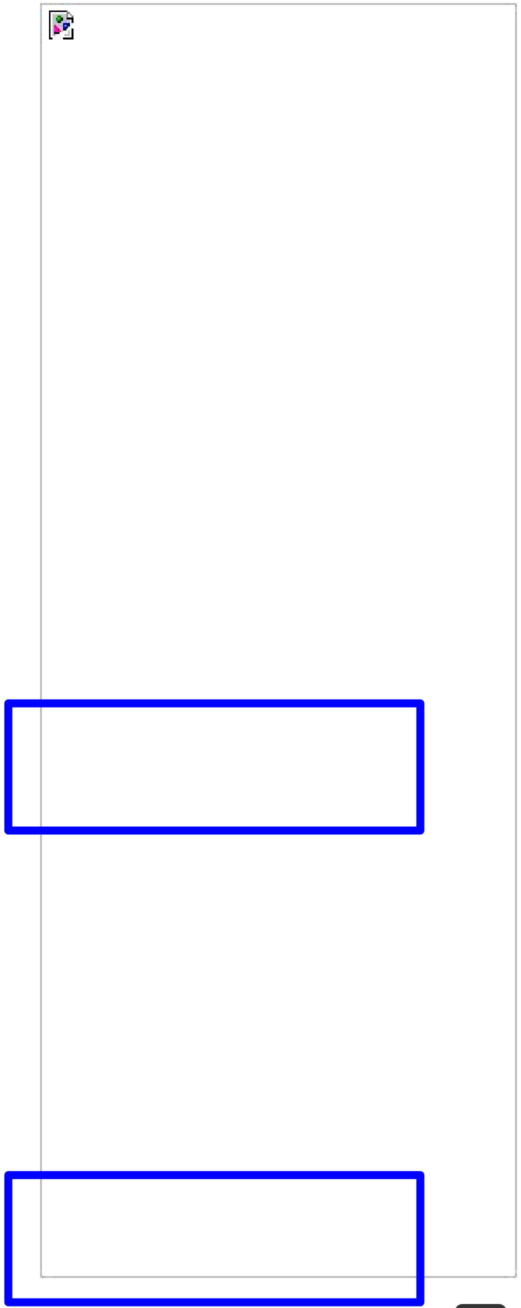
1. Terraform: infrastructure as code
2. Docker-machine: do it manually
3. Cloud-config for CoreOS & other cloud-based systems
4. Chef / Ansible / Puppet for configuring host machine



Deployment

1. Hubot (chat-ops)
2. Chef / Puppet / Ansible recipes
3. Cloud-config for CoreOS & other cloud-based systems
4. Env-specific configurations

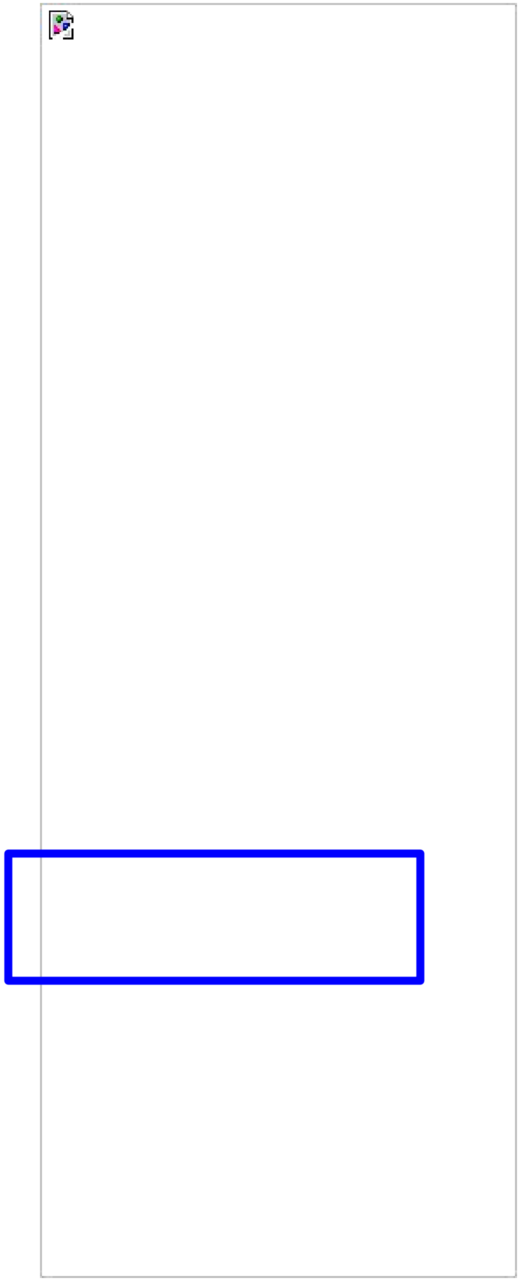
```
$ hubot env staging.ms-users.version 3.5.7
$ hubot converge staging
$ hubot promote staging production
$ hubot converge production
```



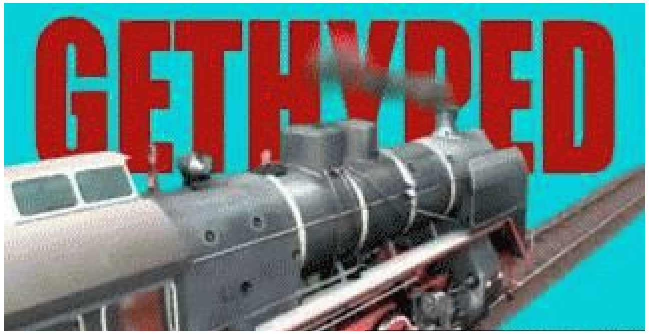
QA

Check your releases.

Do smoke tests if possible



Pros of our approach



Infrastructure as code

Easy to combine, compose & rollback

Work concurrency

“Lego” blocks for future projects

“Mostly” tested

Problems yet remain

Multiple environment configurations

Drastic differences between dev & prod

Peer Dependencies

Server Affinity

Discovery

Orchestration

Thank you

We are hiring

Interesting projects

Constant growth

Cross disciplinary

Ready to study ourselves

v@makeomatic.ru

Problems yet remain

Multiple environment configurations

Drastic differences between dev & prod

Peer Dependencies

Server Affinity

Discovery

Orchestration