1 The Normality of being a member

You are not alone

- Share the workload
- Share ideas
- Share the responsibility
- Guidance and mentoring
- Specialisation
- Polyphony
- Bus Factor > 1
2 Joining a startup
as a sole system engineer

2.1 You are alone

- You will do most of the work
- You will talk to yourself
- You are responsible
- You are both a generalist and a specialist
- Bus Factor $\approx 1$
2 Joining a startup
as a sole system engineer

2.2 The bright side

- Time to grow up
- Take decisions and responsibility
- Resourcefulness and creativity
- Fewer blind spots
2 Joining a startup as a sole system engineer

2.3 Significance of your role

- Know why you are here
- Set the standards
- Bring your experience
- Stand between devs and infrastructure
- Find answers and solutions
- Design and advise for the future
3 Planning and building
your infrastructure

3.1 Observe and listen

- Learn about the product and its future
- Understand components dependencies
- Learn the current deployment process
- Gather information
- Document the current state
- Document manual steps
- Detect bad habits
3 Planning and Building
your infrastructure

3.2 The big picture

- Configuration Management*/Orchestration
- Provisioning environments
- Monitoring & Alerting
- Logging & Metrics
- Update Management
- Viable backups
- Security

* where available
3 Planning and building
your infrastructure

3.3 Take small steps

- Prioritise
- Start with familiar and basic tools
- Avoid complex solutions (save it for later)
- Develop tools
- Iterate
3 Planning and building
your infrastructure

3.4 Creating processes and rules

- Decide how things should be done
- Automate as much as possible
- Find reusable and clear solutions
- **Be consistent** (e.g. in naming)
- **Be consistent** (i.e. be consistent)
- Hide complex procedures
- Revise when you should
3 Planning and building
your infrastructure

3.5 Research, deploy, break

☐ Choose wisely what to research, and when
☐ Understand the company’s size and needs
☐ Learn to let go
☐ Be patient
☐ Failures will happen, and you’ll fix them
☐ Think ahead and design for tomorrow
3 Planning and building
your infrastructure

3.6 Document

- Use a work tracking tool
- Try to plan your week
- Have a *shared* wiki (FAQs, Runbooks etc)
- Write readable code
4 Common pitfalls

Mind the gap

- Always saying yes
- Always saying no
- Over-engineering
- Failure to decode information
- Underestimation
- Assumptions
5 Working in a development team

You are in the same team

- Manage interruptions
- Be approachable and compromise
- Share your knowledge and train
- Communicate processes and rules
- Don’t be arrogant
6 Becoming a team

- Recount your experiences
- Delegate
- Train
- Let the team evolve
- Don’t micromanage

Share your Legos
7 Summing up

- Make short iterations
- Let your solutions mature
- Revise
- Your role is to save time
- Always say “we”
- Embrace failures and mistakes
- Don’t be arrogant
THANK YOU
Links

Tools

- Ansible: https://scotch.io/tutorials/getting-started-with-ansible
- Jenkins: https://www.sumologic.com/blog/devops/jenkins-developing-ci-pipeline/
- Jenkins: https://www.novoda.com/blog/new-jenkins-pipelines/
- Fabric: https://tinyurl.com/q86mer3
- Munin: http://blog.bbv.ch/2013/02/16/linux-server-monitoring-with-munin/
Links

Images