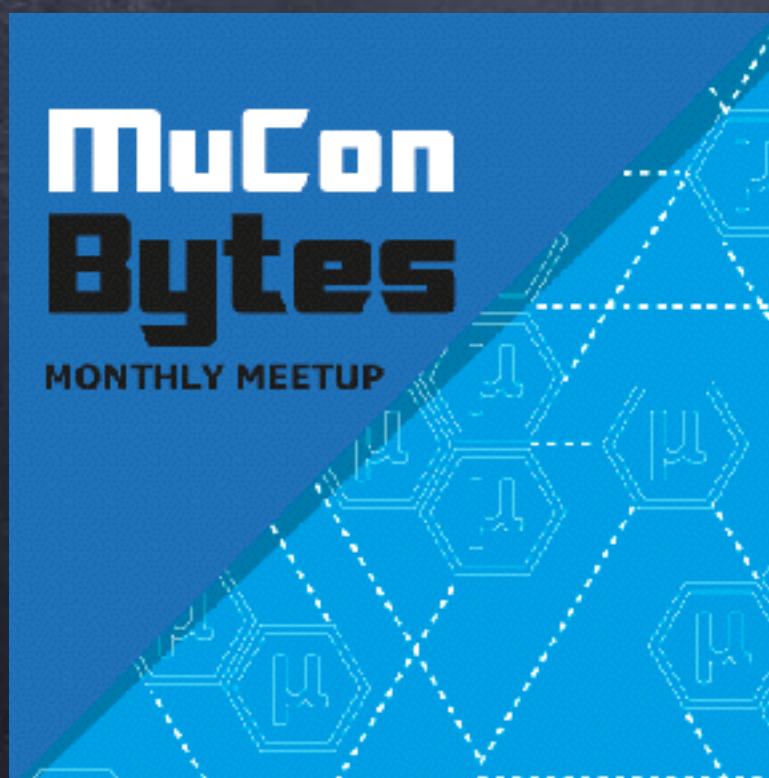
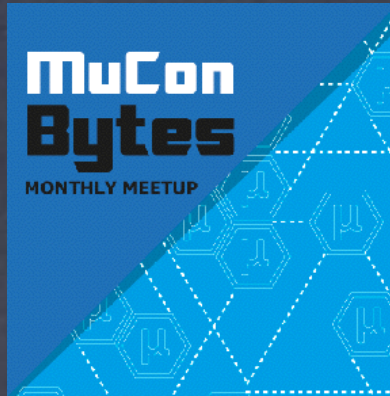


# The Recipe for DevOps

Robert Firek

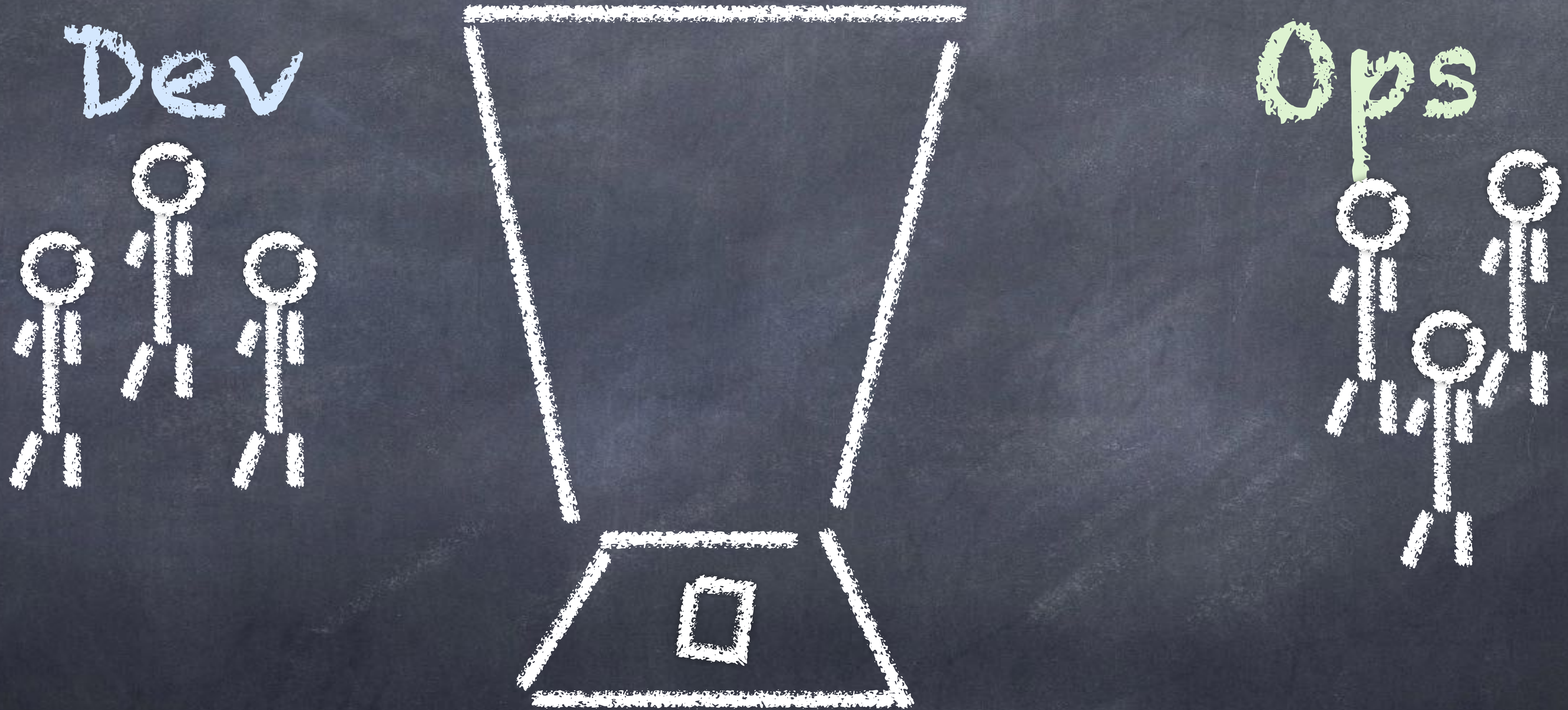






# What is DevOps?







# DevOps







What is  
YOUR DEFINITION  
of DevOps?





"I don't know"

"You are the expert"

"I don't care"





Software engineering practice that  
aims at UNIFYING software  
DEVELOPMENT (Dev)  
and  
SOFTWARE OPERATION (Ops).





DevOps movement is to strongly  
ADVOCATE  
AUTOMATION and MONITORING  
(...),  
from integration, testing, releasing  
to deployment and infrastructure  
management.





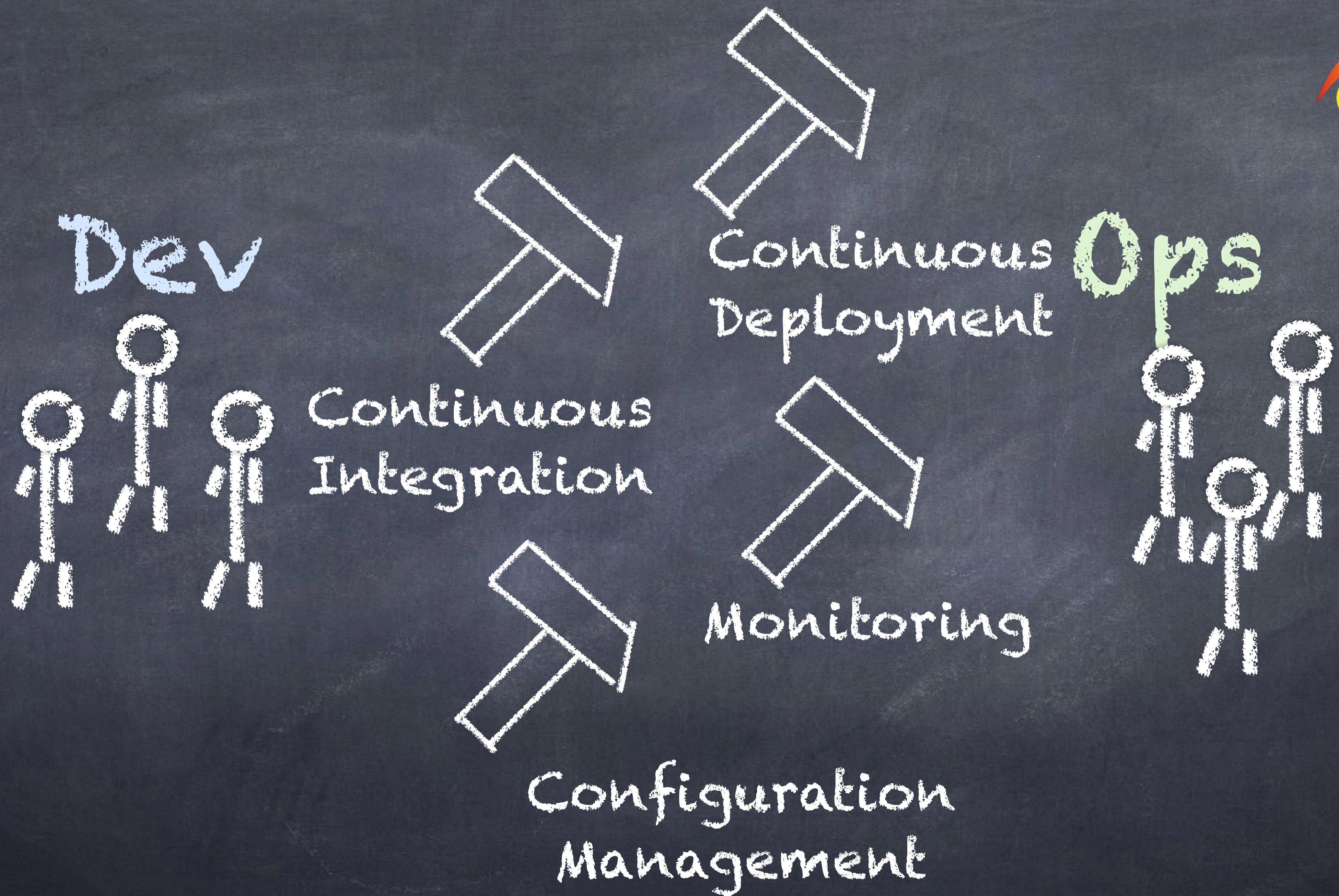
DevOps aims at  
SHORTER DEVELOPMENT CYCLES,  
INCREASE DEPLOYMENT FREQUENCY,  
more DEPENDABLE RELEASES,  
in close  
ALIGNMENT WITH BUSINESS  
OBJECTIVES



# DevOps










Dev



DevOps

Ops







Do I really  
know  
what is DevOps?





What do  
we want to achieve  
by introducing  
DevOps?





Understand Process

Automate Process

Frequent Deployments

Scale my application





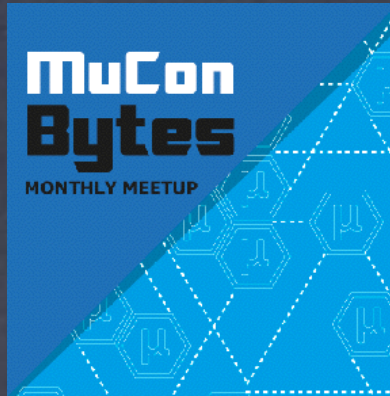
What does  
DevOps bring to  
our project?





What is the most  
important for  
our company?





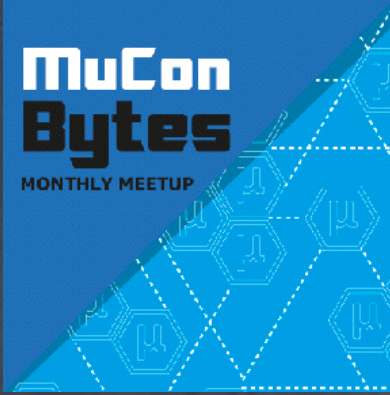
# Business Value Software





How does  
our software  
deliver  
our business value?





# Agile





How do we  
deliver  
our software to  
our customers?





# Code Executable Infrastructure Deployment



Where  
is  
our application?





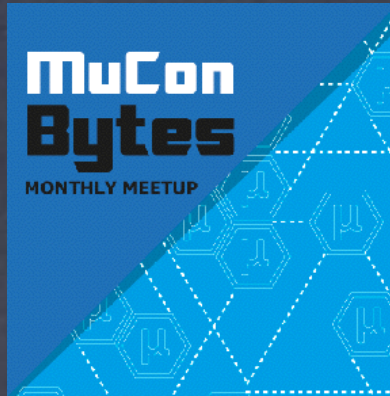
Why do we need to  
know that?





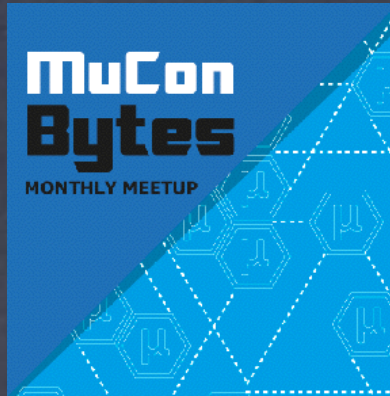
Who is  
responsible  
for  
our infrastructure?





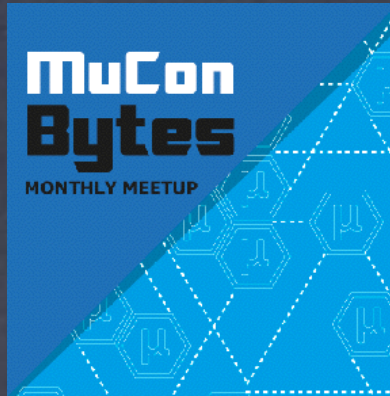
Nobody  
Everyone





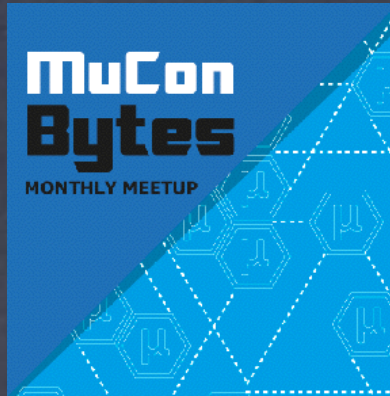
# Personal Administrator





# Ticket-driven Infrastructure





# Infrastructure as Code





What  
kind of resources  
do we need for  
our software?





How do we put  
our application  
on the machine?





How can we  
control  
our deployment?





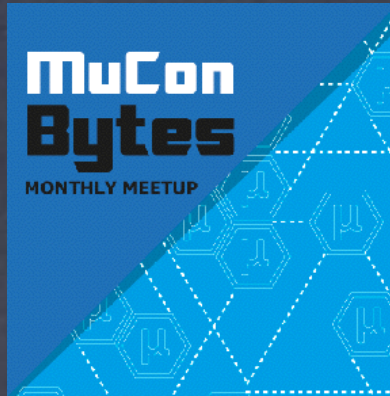
Manually  
Improve  
Automate





Why is it so  
important?





# Feedback





How do we know  
that our application  
is running?



How do we know  
that our application  
is failing?





Do we know  
how to fix  
our application?





Do we have  
enough data to  
fix  
our application?





How should we  
react on failures?



What do we do  
after the failure?





How do we prevent  
the problem from  
happening again?





How do we  
protect  
our application?





When should we  
introduce  
all these practices?









How the hell  
should we  
do all of that?!



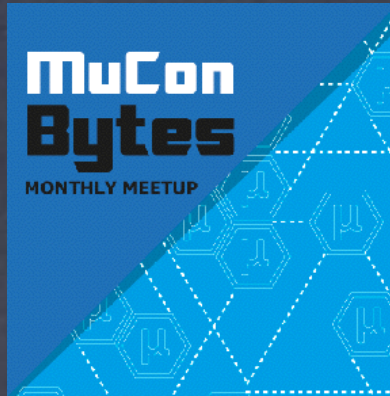
Who will  
allow us  
to do  
all these things?!





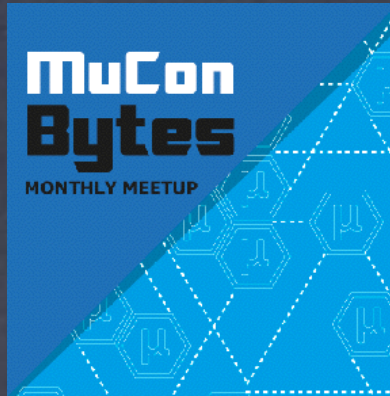
Where is  
the recipe  
for  
DevOps?





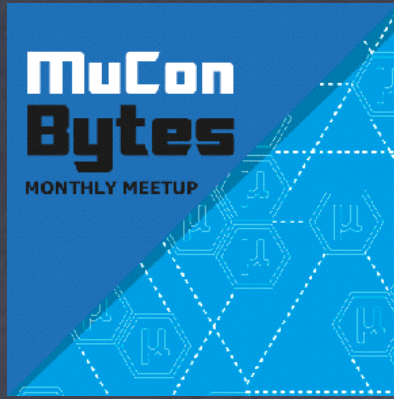
THERE IS  
NO  
RECIPE.





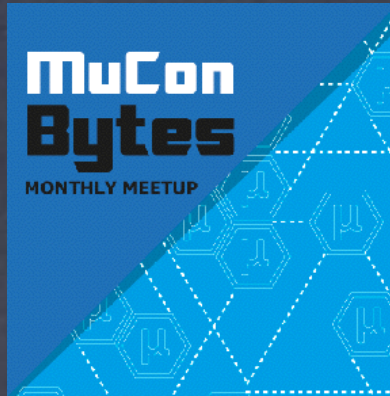
THERE IS  
NO  
ONE RECIPE.





Thank you





# Questions.