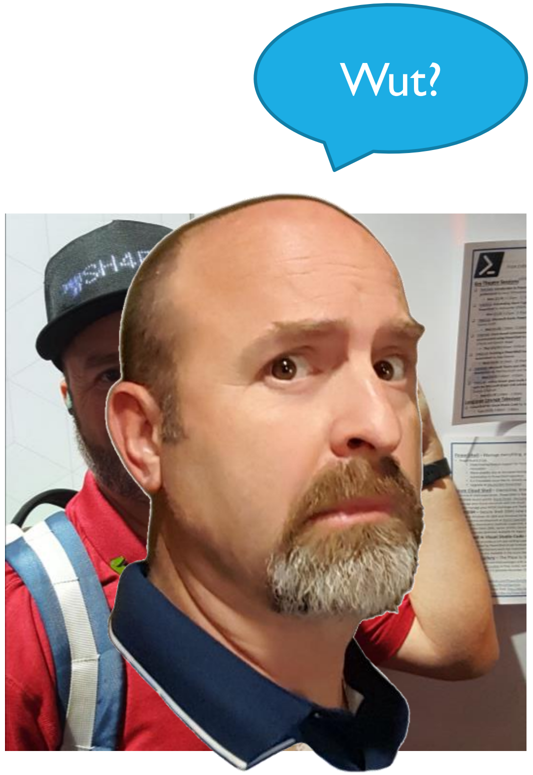
# DON'T DESTROY YOUR DATACENTER WITH AZURE DEVOPS

STEVEN JUDD

# WHO AM I?

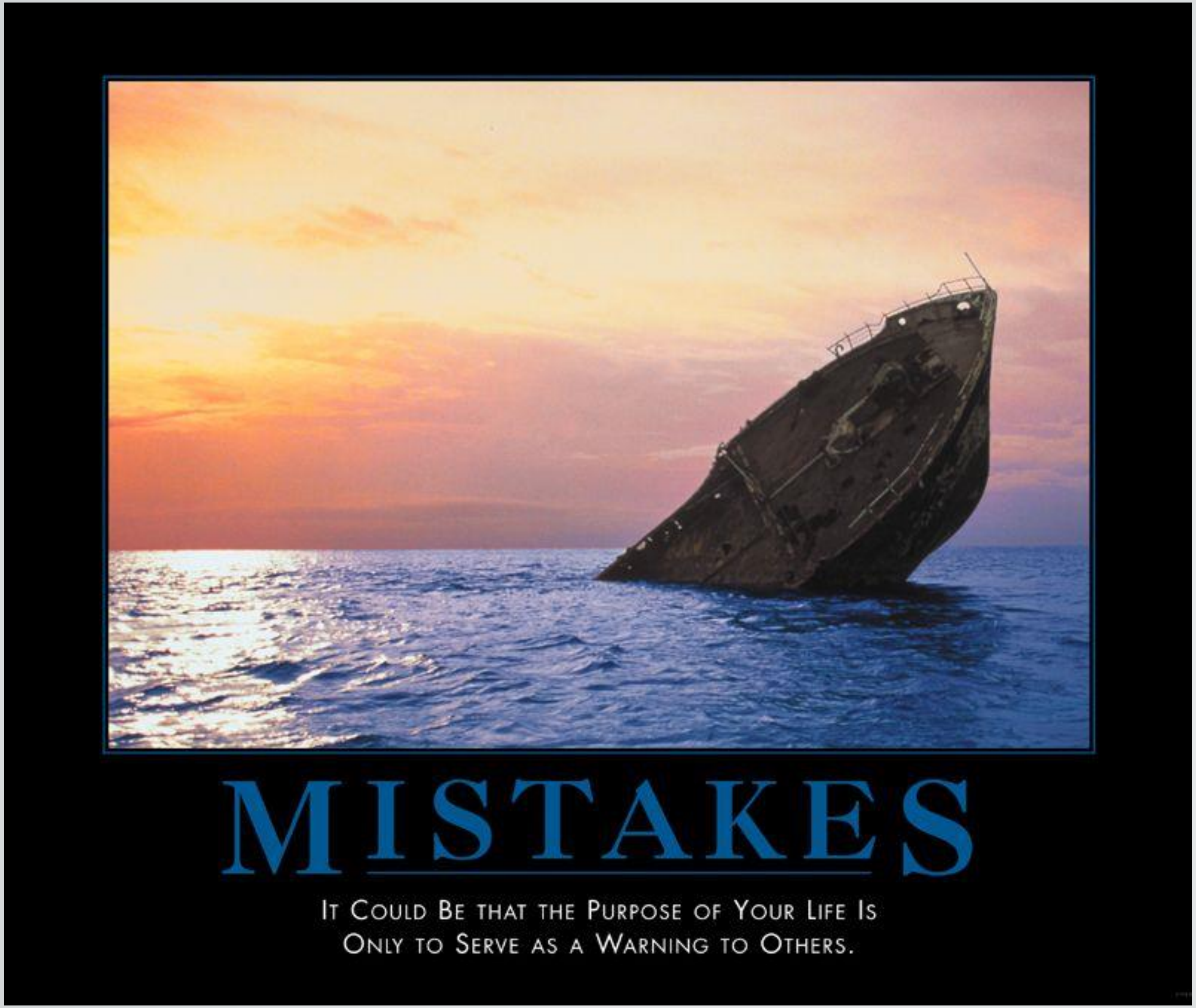
- Steven Judd
  - Multi-year, multi-discipline IT Pro
  - PowerShell enthusiast
  - Dad joke enthusiast
  - Fashion icon
    - Maybe not…



@stevenjudd

# WHY SHOULD YOU CARE?

- We are all professionals, aren't we?
- We trust each other, don't we?
- We know what *that* setting does, right?
- What is the worst that could happen?
  - This is foreshadowing…


MISTAKES
It Could Be that the Purpose of Your Life Is
Only to Serve as a Warning to Others.

https://despair.com/collections/demotivators/products/mistakes?variant=2457302467

# SCOPE FOR DISCUSSION

- Azure DevOps – https://dev.azure.com/<site>
  - Formerly known as https://<site>.visualstudio.com
- Azure Portal – https://portal.azure.com
- GitHub, TFS, GitLab, SVN, and whatever else are out of scope
- AWS, GCP, IBM, Oracle, and whatever else are out of scope
- Don’t Leave! The principles discussed applies to any platform
- Also, very little PowerShell in this presentation

# THE BUSINESS CONTINUITY/ DISASTER RECOVERY TRIPOD

- Protect your data
- Protect your code
- Protect your infrastructure

# WHERE TO WATCH

- Know your users
  - Beware over-privileged user accounts
    - Azure - Global Administrators
    - AzDO - Project Collection Administrators
  - Beware service accounts
    - Especially shared service accounts without password expirations
  - Beware external accounts
    - Isn’t Azure Active Directory helpful?
    - Severely restrict these accounts’ access to your environment

# SECURE AZURE ACTIVE DIRECTORY

- Easy, right? Not many options, right?
- What to focus on
    - Global Administrators
    - 2FA/MFA requirements
    - Password length over complexity/symbols
    - Owner rights
        - Better to give out Contributor rights
        - Even better to limit access to specific Resource Groups

# WHO'S WATCHING WHO?

- Auditing in AzDO
  - Must be backed by Azure Active Directory
    - See previous slides
    - Hopefully, your org already has this setup. If not, setup can be a bit tricky.
  - Only keeps 90 days of events
    - This may be an issue for your organization if you need auditing for longer
  - Connect to a SIEM (if you have one)
    - Allows for alerting based on specific events
- Demo

# BRANCH POLICIES

- Lock down your main branch
  - Who can change the policies
  - Who can merge into main
- This is not the time or place for a branching strategy holy war
  - There isn't a best branching strategy
- What matters is that you are in control of how and when code is merged
- Demo

Trunk based is the only way to go
and GitFlow is for broken people

# PROTECTING THE INFRASTRUCTURE

- Lock your Azure resources
  - Some solutions will not allow you to lock resources
    - Looking at you, Commvault (at least, it used to be this way)
  - Owner and User Access Administrator built-in roles can create and delete management locks
    - Protect your Azure resources with a lock - Azure Resource Manager
- Audit the changes to the resource locks

# EVALUATE YOUR AZDO/AZURE CONNECTIONS

- Connections from AzDO to Azure can be setup in 4 ways
  - Authentication
    - Service Principal (automatic)
    - Service Principal (manual)
    - Managed Identity
    - Publish Profile
- Demo

# EVALUATE YOUR AZDO/AZURE CONNECTIONS, CONT.

- Connection destination has 3 scopes
  - Scope Level
    - Subscription
    - Management Group
    - Machine Learning Workspace
    - The hidden option
- This is where the danger lies…
  - Over-broad destination access can allow for unaccounted access
- Demo

# ROLE PLAYING

- Scene
  - A connection has been made to Azure from a Team Project in AzDO
  - A web site has been setup using a CI/CD pipeline
  - A change in rights to the pipeline
  - A change in the pipeline
  - A change in the code

# RECOMMENDATIONS – A

- Split out your Azure environments
  - Better to lock down your prd environment and have less to watch
  - Allows for greater rights to developers to allow them to create
- Stream Audit Events
  - Setup triggers/alerts for specific actions that are dangerous

# RECOMMENDATIONS – B

- Evaluate Roles in both AzDO and Azure
  - Know who has the ability to do what
  - Know when someone is granted capability
- Communicate
  - Work with your security, identity, and developer team(s)
- Find the right balance
  - There isn't a perfect solution
  - Find the one that works for you

# RECOMMENDATIONS – A + B (YOU C?)

- Split out your Azure environments
  - Better to lock down your prd environment and have less to watch
  - Allows for greater rights to developers to allow them to create
- Stream Audit Events
  - Setup triggers/alerts for specific actions that are dangerous
- Evaluate Roles in both AzDO and Azure
  - Know who has the ability to do what
  - Know when someone is granted capability
- Communicate
  - Work with your security, identity, and developer team(s)
- Find the right balance
  - There isn't a perfect solution
  - Find the one that works for you

# STAY IN TOUCH

- Twitter: @stevenjudd
- LinkedIn: https://www.linkedin.com/in/stevenjudd/
- PowerShell Bridge Channel
  - http://poshcode.org/
  - Discord: https://discord.gg/Ju25cw6 << use this one
  - Slack: https://tinyurl.com/sm3by7m
- Discord: @juddmissile#7741 (does anyone understand the numbers?)
- ICQ: 340799 (I'm not actually on ICQ…)
- http://blog.stevenjudd.com (needs CSS help…)
- https://github.com/stevenjudd