

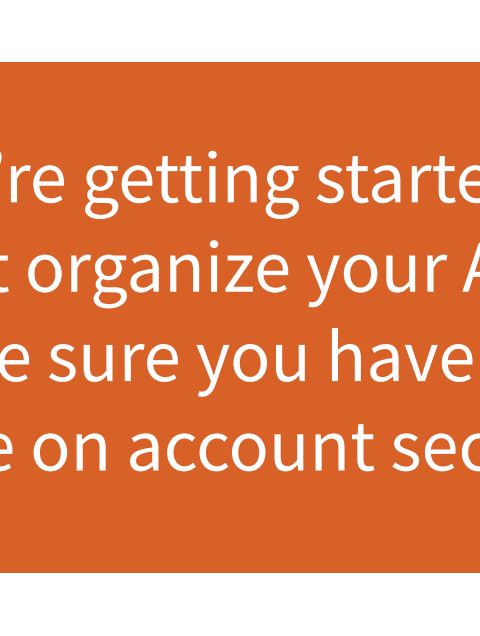
The Top DO'S & DON'TS

For Enterprises on Amazon Web Services (AWS)

Migrating to AWS

+ Building a DevOps Culture and Practice

Limitless Possibilities



But how can you best leverage AWS resources in a reliable and secure manner while deploying rapidly and maintaining cost efficiency?

Foundational Best Practices

When you're getting started on AWS, you must first organize your AWS accounts and make sure you have a strong handle on account security.

DON'T!	DO!
<p>Overload AWS accounts...</p> <p>You may be tempted to add as much as you can to one account, but this can cause complexities such as complex access administration, a larger blast radius, and tricky cost allocation.</p> <p>Create AWS IAM users directly in accounts...</p> <p>For any organization with a substantial number of developers and users, challenges can arise such as: how do you deal with suspending or terminating accounts, key rotation?</p>	<p>Use Organizations API</p> <p>Use a separate toolchain account</p> <p>Use federation</p> <p>Enable AWS CloudTrail</p> <p>Enable Amazon GuardDuty</p> <p>Enable AWS CLI access</p> <p>Enable VPC flow logs</p>

Building Your Infrastructure on AWS

What are some of the key considerations you must take as you organize networks and auto scaling policies within your AWS accounts?

DON'T!	DO!
<p>Use public subnets...</p> <p>Or rather, limit the use of public subnets, and make sure your infrastructure is running in private subnets as much as possible.</p> <p>Launch instances without Auto Scaling Groups (ASG)...</p> <p>You should use ASG for both fleet management and dynamic scaling.</p>	<p>Consider a forward proxy</p> <p>Consider a egress transit VPC</p> <p>Use VPC endpoints</p> <p>Encrypt at rest</p> <p>Encrypt in transit</p> <p>Use target tracking policies</p> <p>Use Amazon EC2 Systems Manager (SSM) parameter store</p>

Taking Advantage of Automation

How can you deploy with consistency and repeatability on AWS? By building automation into your AWS accounts. Automation is key.

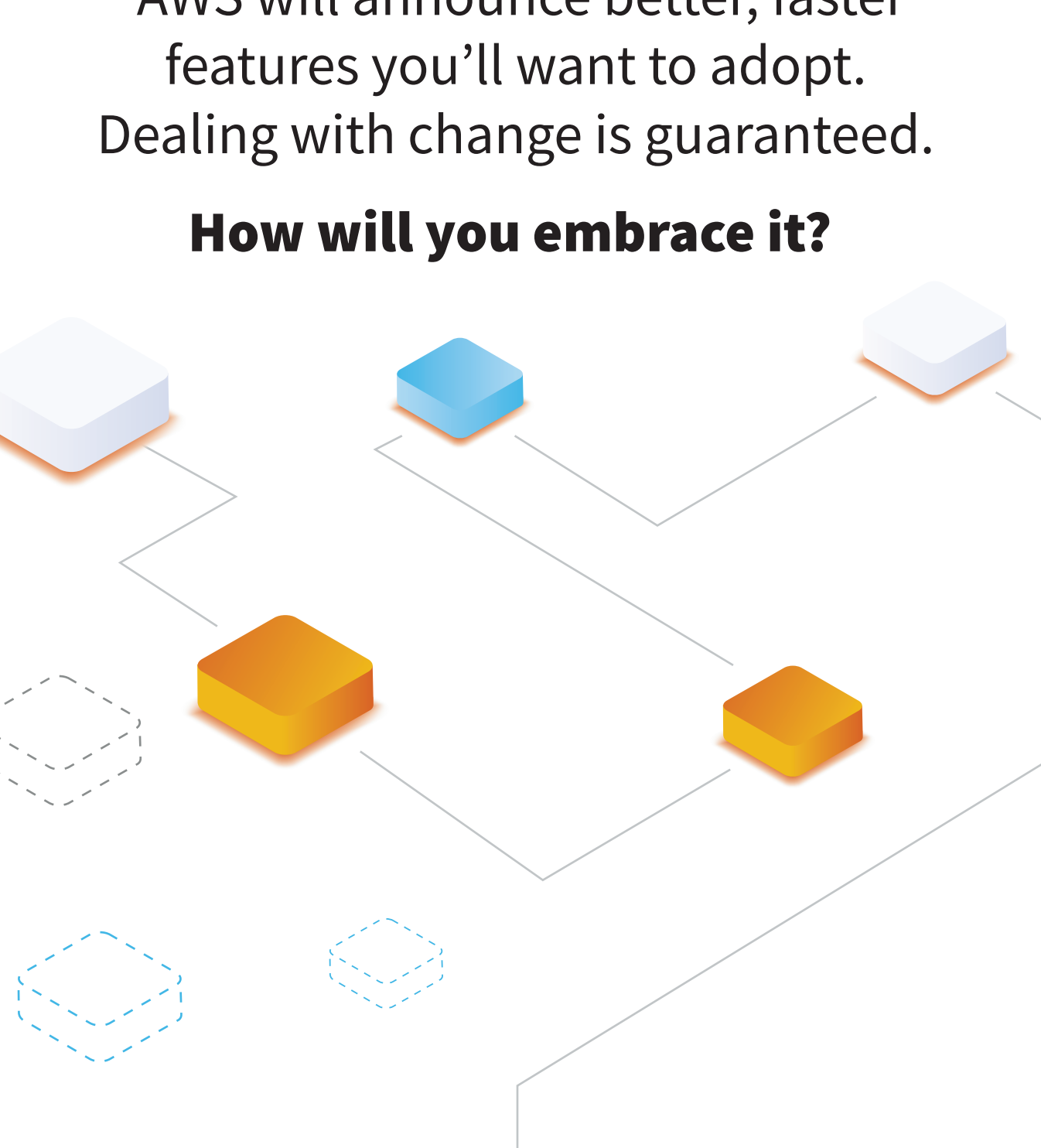
DON'T!	DO!
<p>Manually click the 'Launch Instance' button...</p> <p>Everything you do in your AWS account should leverage code to manage resources.</p> <p>Reinvent automation tools...</p> <p>If the tool has already been built (such as AWS CloudFormation), then let the tool do what it's intended to do to drive consistency.</p> <p>Overload CloudFormation stacks...</p> <p>A single stack that includes everything for an application can present problems for change in the future. Align stacks based on ownership and break down stacks based on the frequency with which stacks change.</p> <p>Go 100% bake or boot for Amazon Machine Images (AMIs)...</p> <p>The question is: How much do you bake into the AMI vs. how much do you configure on the instance after you boot the AMI?</p>	<p>Use Organizations API</p> <p>Use policies in AWS CloudFormation</p> <p>Use VPC endpoints</p> <p>Prefer containers over instances</p> <p>Governance via AWS Service Catalog</p> <p>Access security in pipelines</p> <p>Automate compliance</p>

Most Importantly

Architect for Change...

AWS will announce better, faster features you'll want to adopt. Dealing with change is guaranteed.

How will you embrace it?



Seize past failures as opportunities to improve and make your AWS journey your own.

WATCH WEBINAR