

GPSTEC302

AWS re:INVENT

Anti-Patterns: Learning through Failure

November 27, 2017

AWS
re:Invent



How does a practice become a "best" practice? How does a pattern become an "anti" pattern? As always, experience is the best teacher. As Partner Solution Architects, we receive a lot of partner feedback on how practices and design patterns work—and occasionally fail to work—in the real world. We use this feedback to inform our recommendations and reference architectures. In this session, we explore a representative set of real-life "failures." We look at what these failures have to teach us about design and how to prioritize remediation of known issues.

Introduction and Definitions

- **Anti-patterns** *lead to best practices*
- **Best practices** are *learned* and often *earned*
- We can learn from the behavior of others

Best Practice Creation—Myth



This work has been released into the [public domain](#) by its author, [Andrew Horne](#) at [English Wikipedia](#). This applies worldwide.

AWS
re:Invent

© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.



Best Practice Creation—Reality



By Sylvain Pedneault - Self-photographed, CC BY-SA 3.0,
<https://commons.wikimedia.org/w/index.php?curid=3616567>

AWS
re:Invent

© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

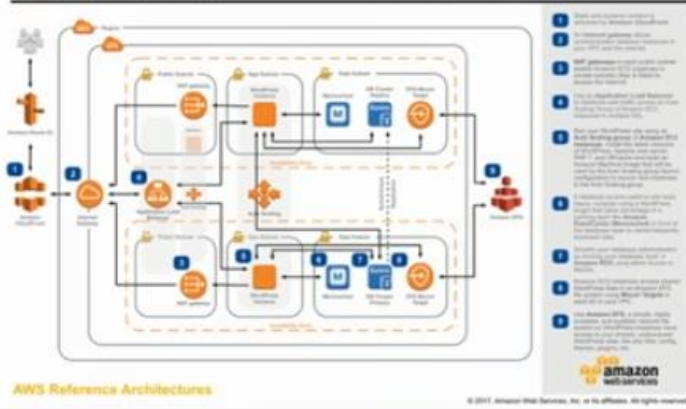


Anti-Pattern: Loss of Control

WordPress Hosting

How to run WordPress on AWS

WordPress is one of the world's most popular web publishing platforms, being used to publish 27% of all websites, from personal blogs to some of the largest news sites. This reference architecture simplifies the complexity of deploying a scalable and highly available WordPress site on AWS.



<https://github.com/aws-labs/aws-refarch-wordpress>

aws re:Invent

© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

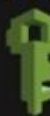
aws



user



temporary security credential



IAM



long-term security credential



mobile device



public repository

aws re:Invent

© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws



temporary security credential



IAM



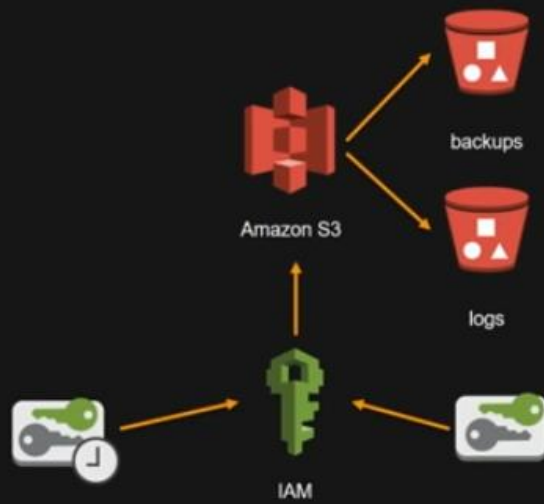
long-term security credential



aws re:Invent

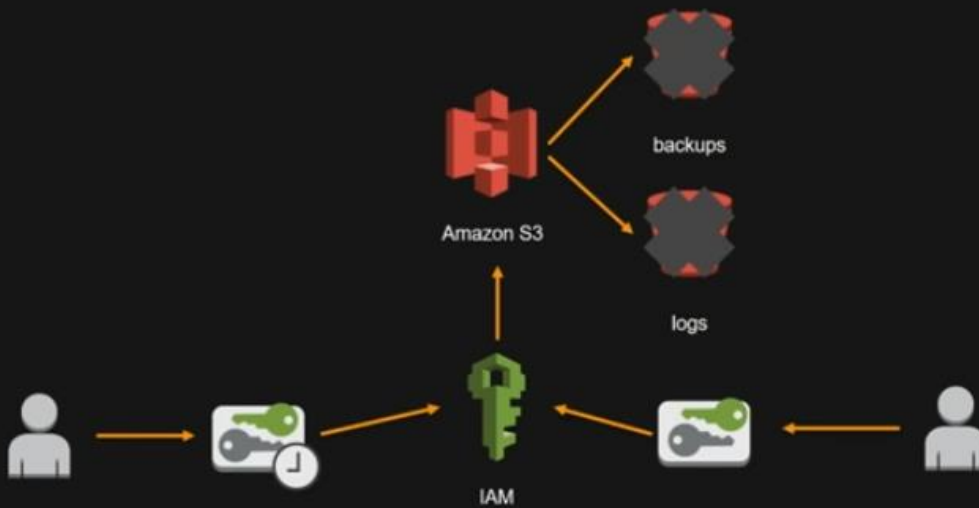
© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws



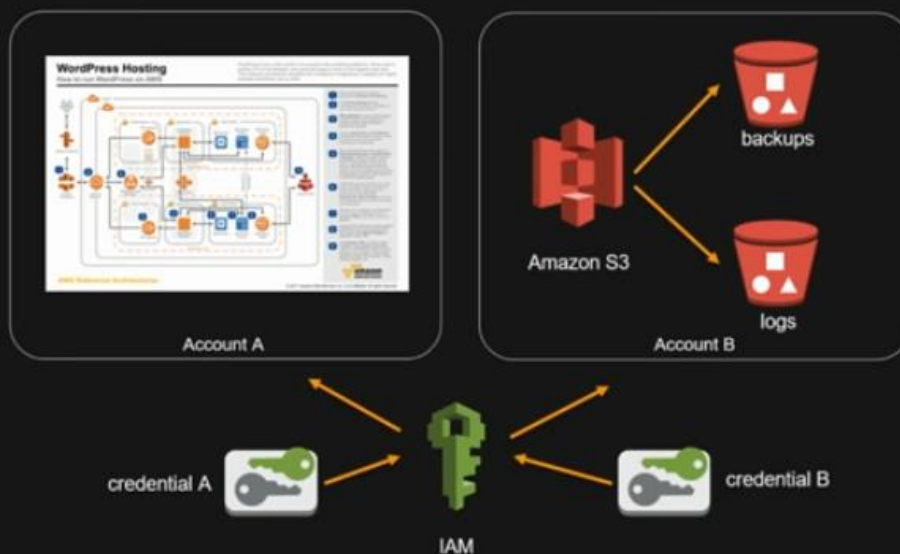
AWS
re:Invent

© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.



AWS
re:Invent

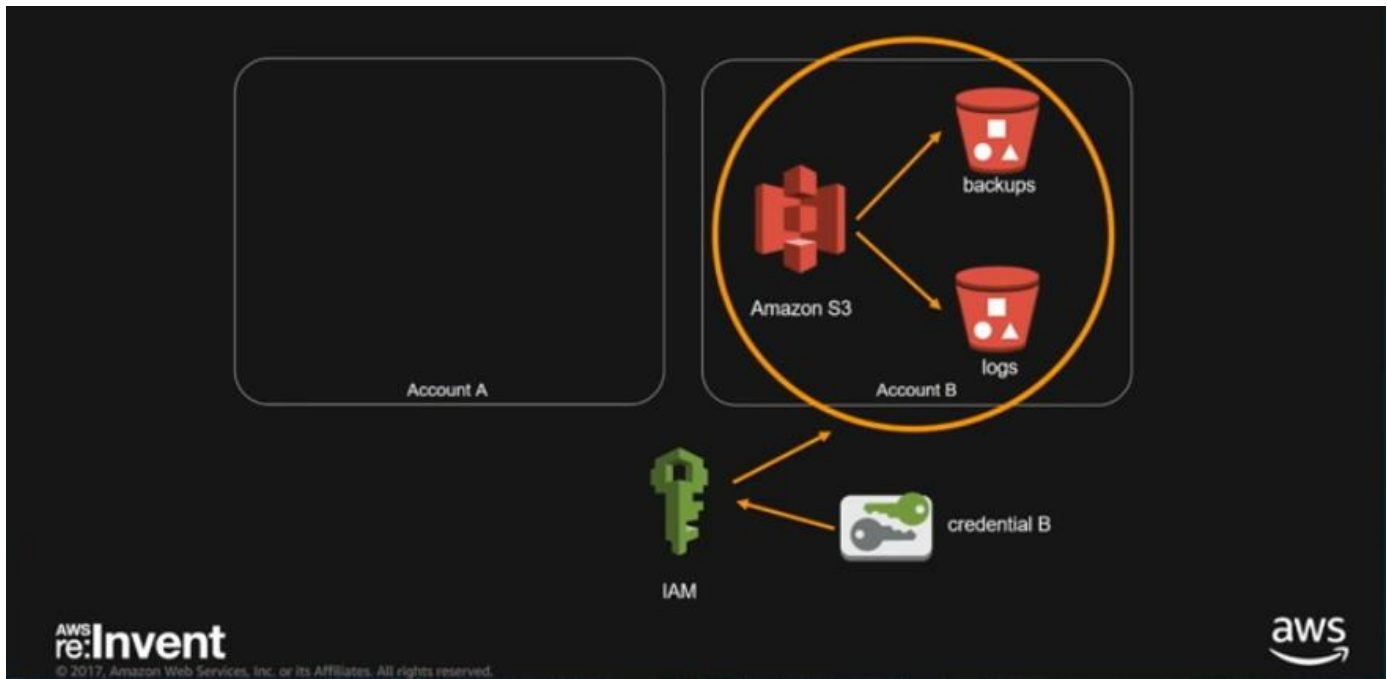
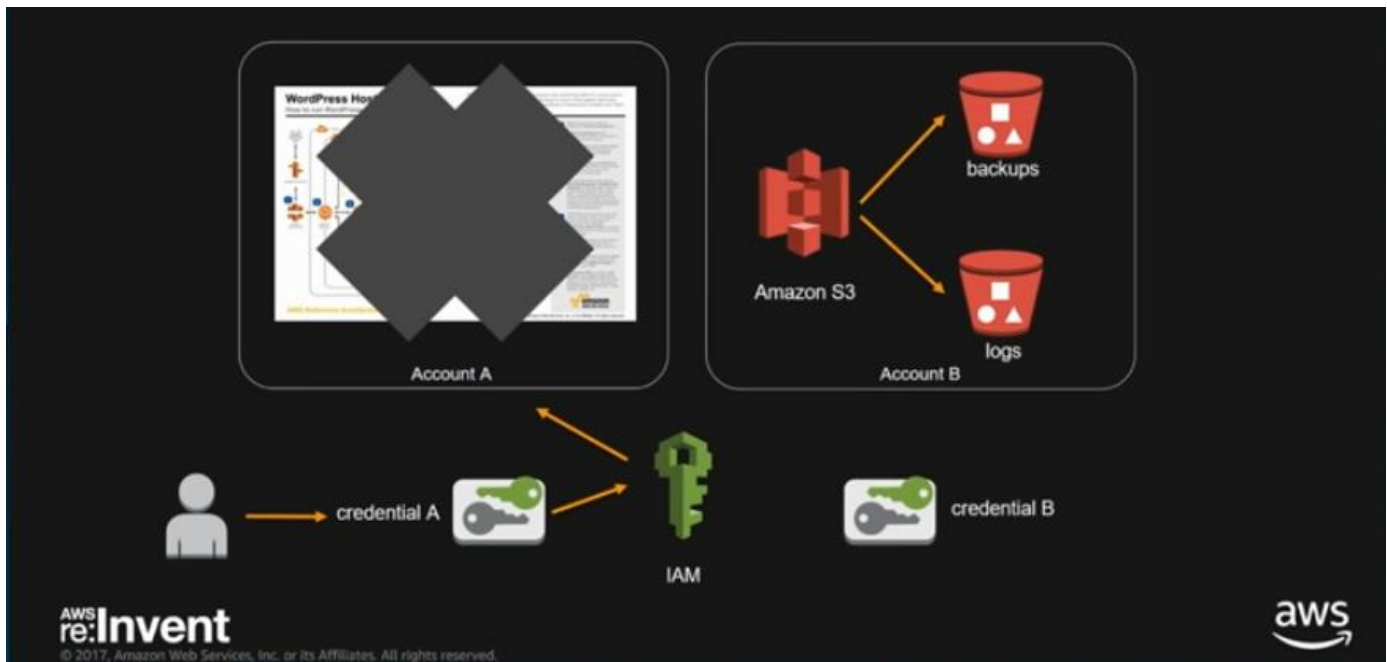
© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

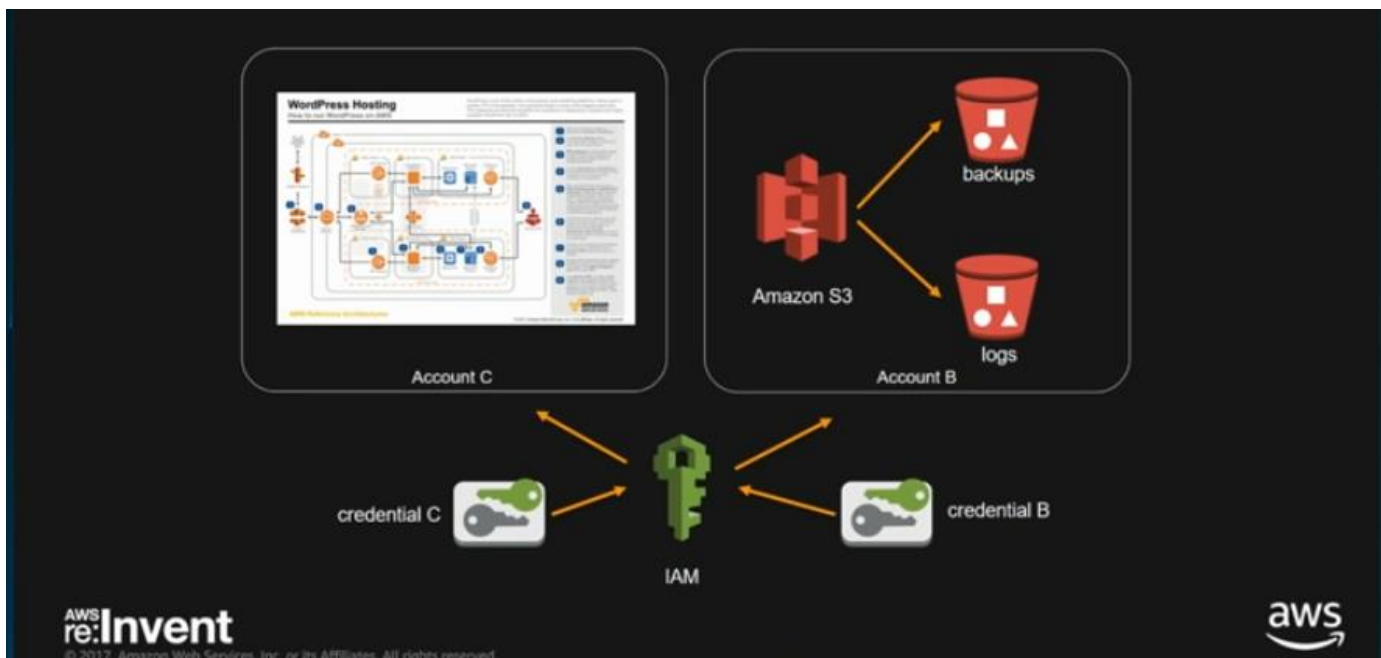


AWS
re:Invent

© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.







Anti-Pattern: Loss of Control

Anti-pattern: **Poor IAM Access Key controls**

Best practices:

1. Lock away your AWS account root user access keys
2. Create individual IAM users
3. Enable MFA for privileged users
4. Never automate with privileged credentials
5. Rotate credentials regularly
6. Audit for compliance
7. Establish separate administrative domains

..and regularly review access policies with an AWS Solutions Architect!

More AWS IAM Best Practices

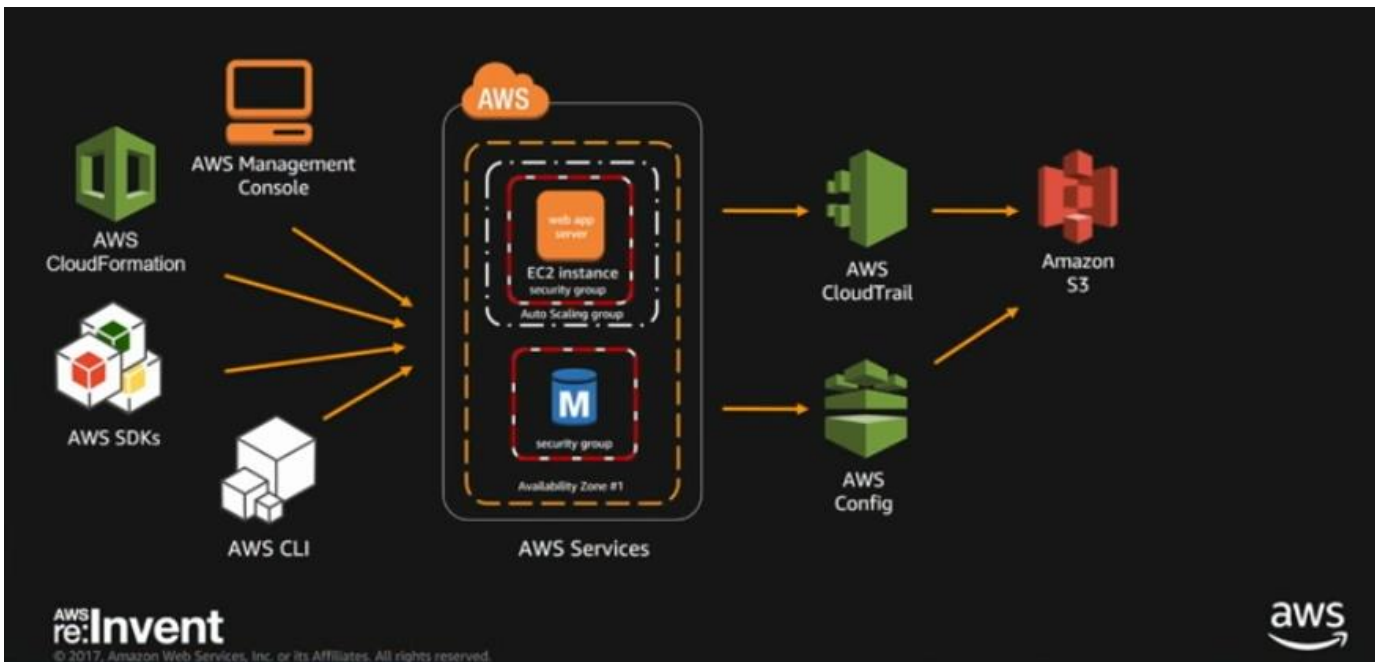
<http://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html>

AWS
re:Invent

© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

Anti-Pattern: Control Gaps



```
{ "Records":
  [ {
    "eventVersion": "1.0",
    "userIdentity": {
      "type": "IAMUser",
      "principalId": "EX_PRINCIPAL_ID",
      "arn": "arn:aws:iam::123456789012:user/Alice",
      "accessKeyId": "EXAMPLE_KEY_ID",
      "accountId": "123456789012",
      "userName": "Alice"
    },
    "eventTime": "2014-03-06T21:22:54Z",
    "eventSource": "ec2.amazonaws.com",
    "eventName": "StartInstances",
    "awsRegion": "us-east-2",
    "sourceIPAddress": "205.251.233.176"
  }
  ...
}
```

AWS CloudTrail is awesome!

AWS CloudTrail gives us logs of everything that goes on in our account

```
"resourceType": "AWS::EC2::Instance",
"resourceCreationTime": "2014-02-26T22:56:35.000Z",
"tags": { "Name": "integ-test-1", "exampleName": "examplevalue" },
"relationships":
  [ {
    "resourceId": "vol-ce676ccc", "resourceType": "AWS::EC2::Volume", "name": "Attached Volume" },
    {
      "resourceId": "vol-ef0e06ed", "resourceType": "AWS::EC2::Volume", "name": "Attached volume", "direction": "OUT" },
    {
      "resourceId": "subnet-47b4cf2c", "resourceType": "AWS::EC2::SUBNET", "name": "Is contained in subnet",
      "direction": "IN" }
  ]
...
```

AWS Config is awesome!

AWS Config gives us a point-in-time snapshot of the inventory of the assets in our accounts

```

AWSTemplateFormatVersion:2010-09-09
Resources:
  SGBase:
    Type: 'AWS::EC2::SecurityGroup'
    Properties:
      GroupDescription: Whitelist Security Group
      SecurityGroupIngress:
        - IpProtocol: tcp
          CidrIp: 167.55.180.10/0 ←
          FromPort: '22'
          ToPort: '22'

```

What's wrong with this picture?

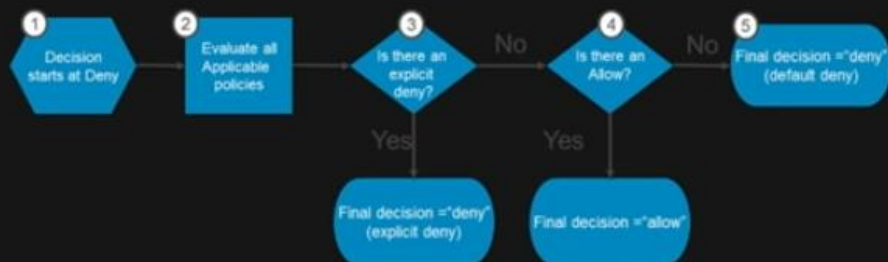
```

AWSTemplateFormatVersion:2010-09-09
Resources:
  SGBase:
    Type: 'AWS::EC2::SecurityGroup'
    Properties:
      GroupDescription: Whitelist Security Group
      SecurityGroupIngress:
        - IpProtocol: any
          CidrIp: 167.55.180.10/32
          FromPort: '3388'
          ToPort: '3390'

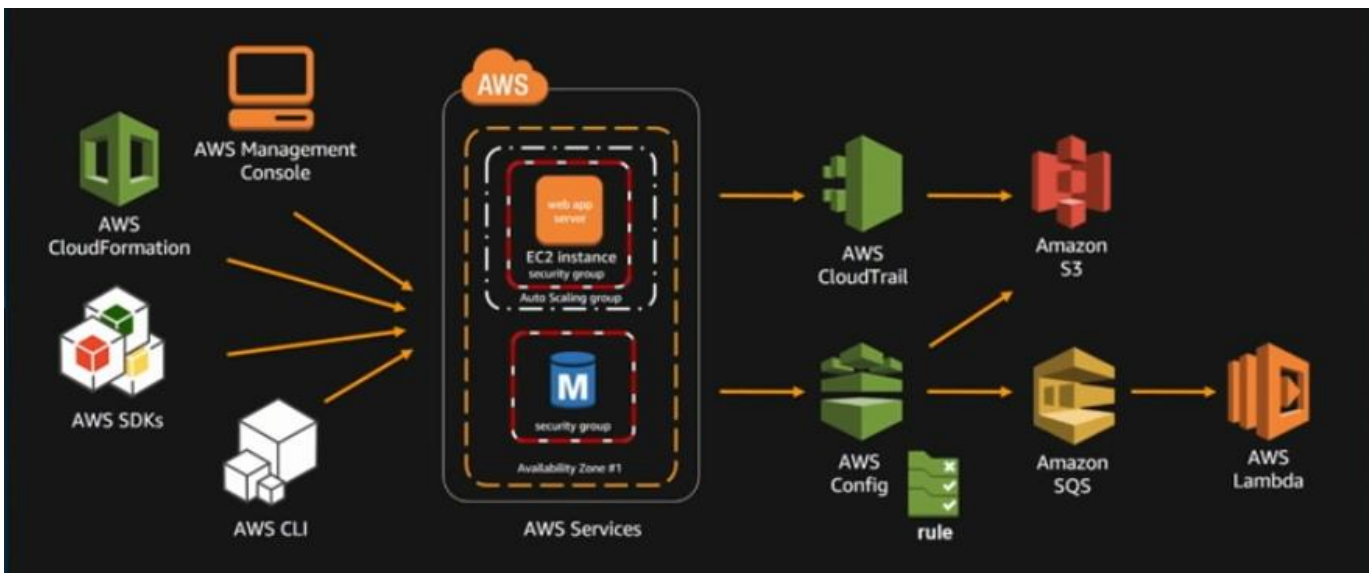
```

What's wrong with this picture?

This is a bad rule to allow access that escapes automation detection but will be easily detected by a human



Amazon S3 authorization process



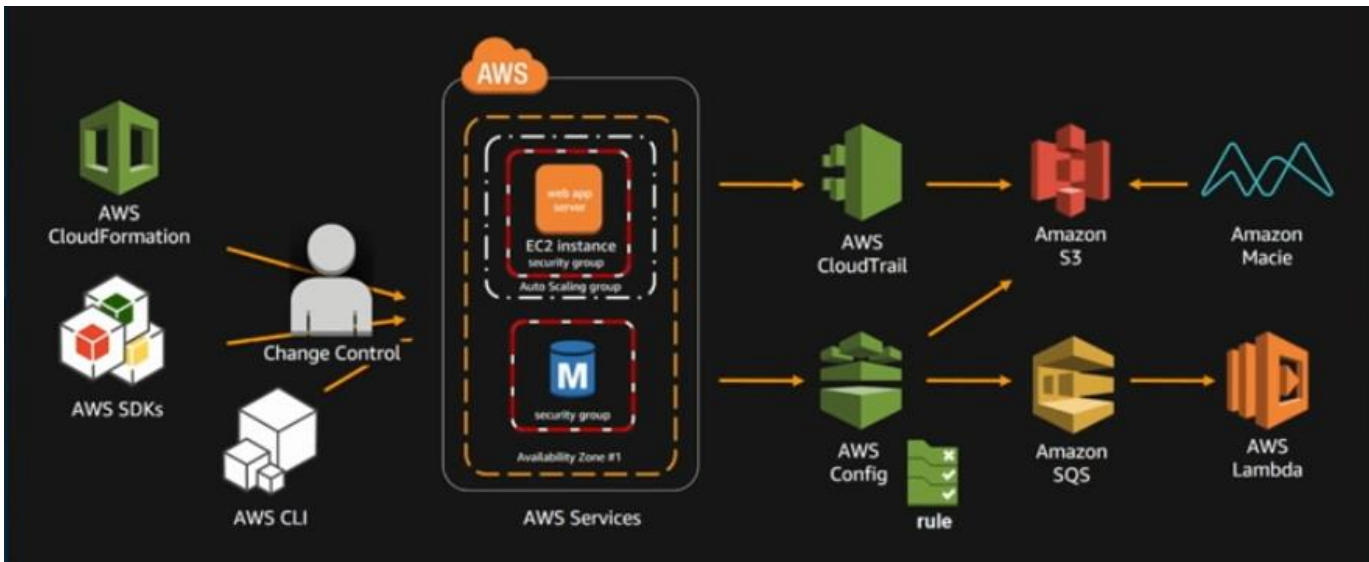
Use the AWS Managed Rules service to help close some of the control gaps in your infrastructure, also use services like SNS and Lambda to auto-correct the control gaps.



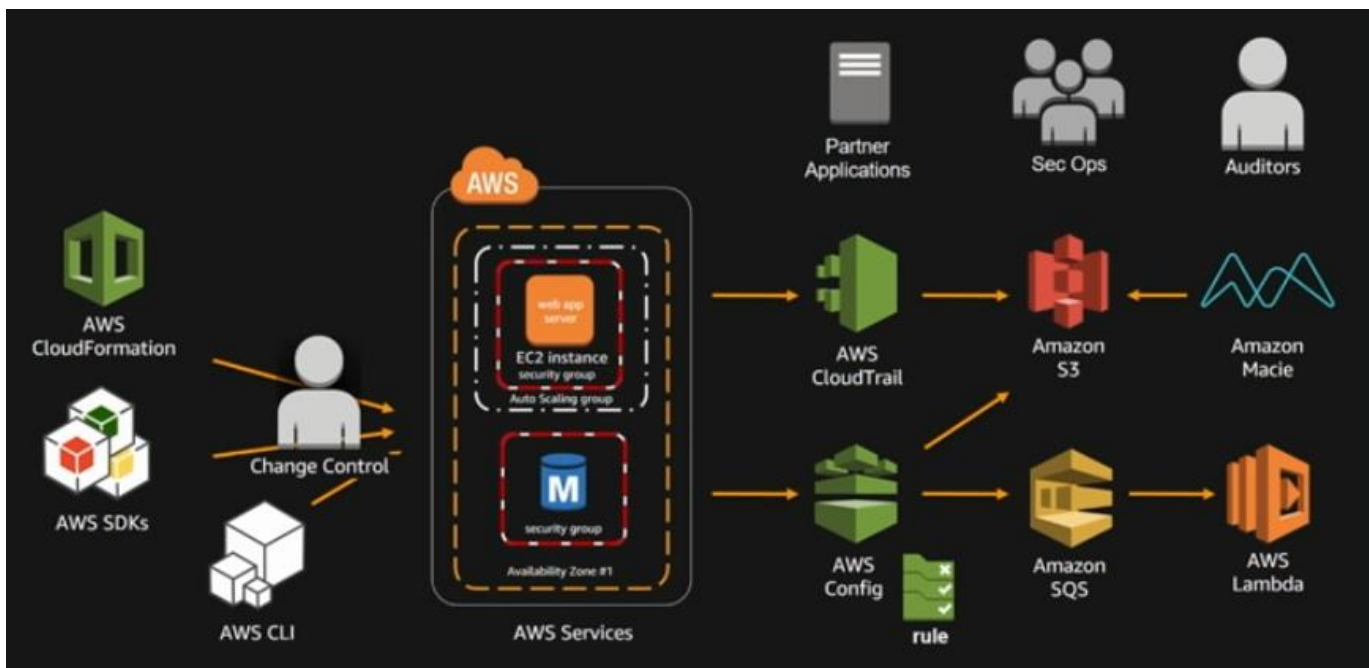
You can also use a service like Amazon Macie



Also consider using change control by using a human to look at the changes before deploying code changes



Maybe don't use the AWS Management Console for managing changes manually, instead use automation



Anti-Pattern: Automated Control Gaps

Anti-pattern: **Reliance on incomplete controls automation**

Best practices:

1. Use managed rules
2. Inject canary events to test controls
3. Use external tests and tools for validation
4. Audit to verify compliance
5. Add manual checkpoints prior to pushing changes
6. Automate everything, but mind the gaps!

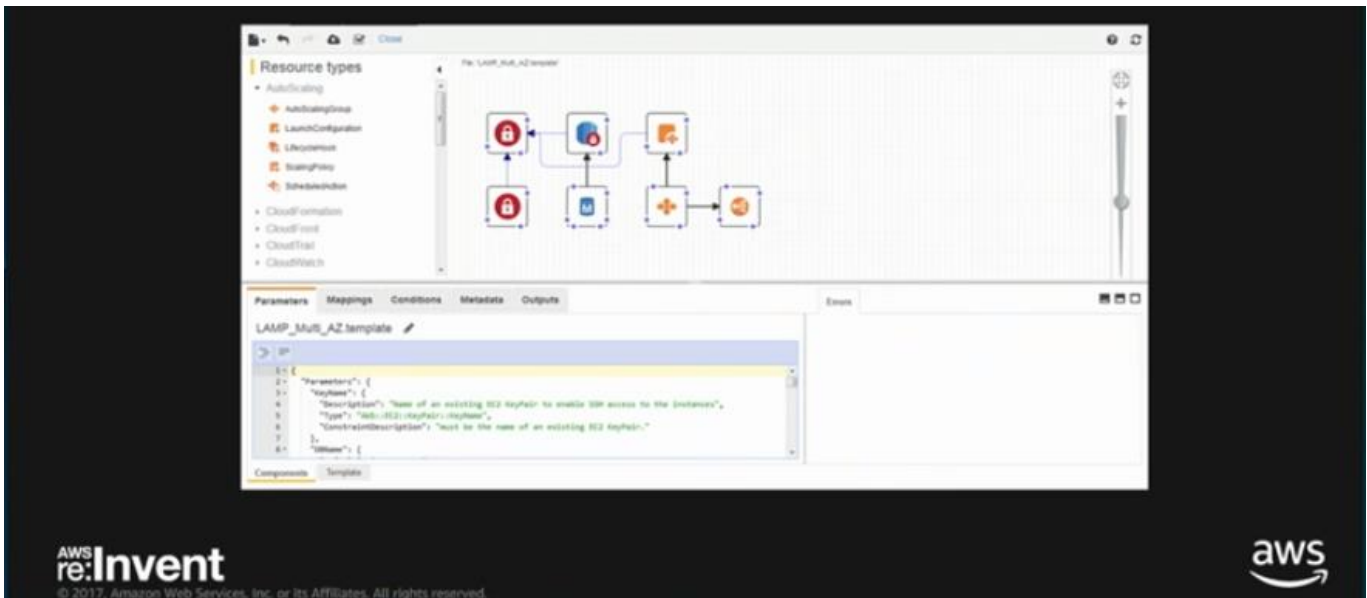
More on AWS Config Managed Rules

<https://aws.amazon.com/blogs/aws/aws-config-update-new-managed-rules-to-secure-s3-buckets/>

More on Automating Governance on AWS

https://www.youtube.com/watch?v=9q0u_05WBig

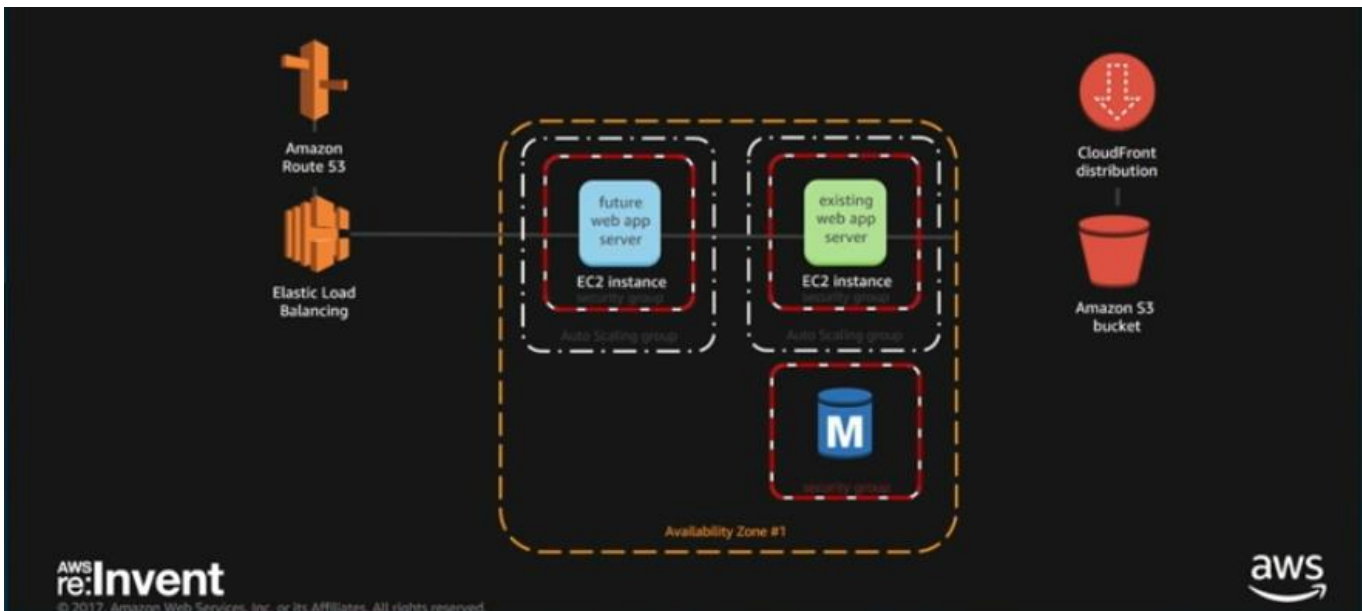
Anti-Pattern: Automating Outages



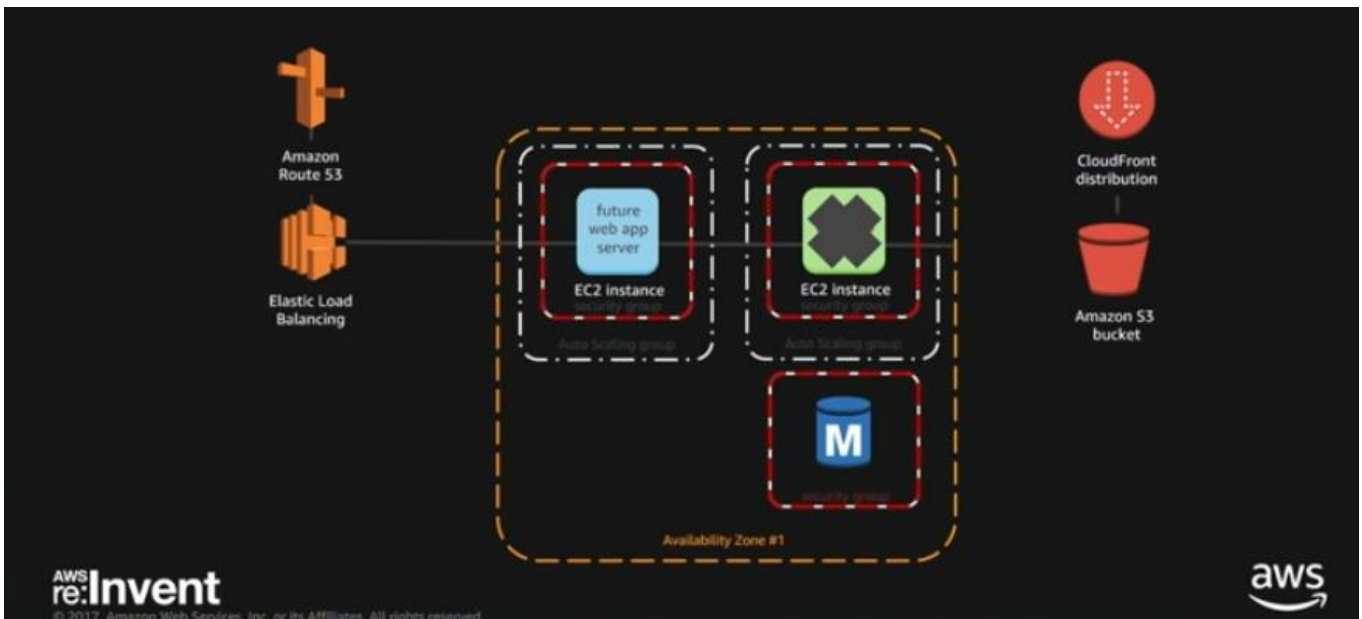
We can easily automate our deployments by using CloudFormation or similar tooling like Terraform, Puppet, etc.



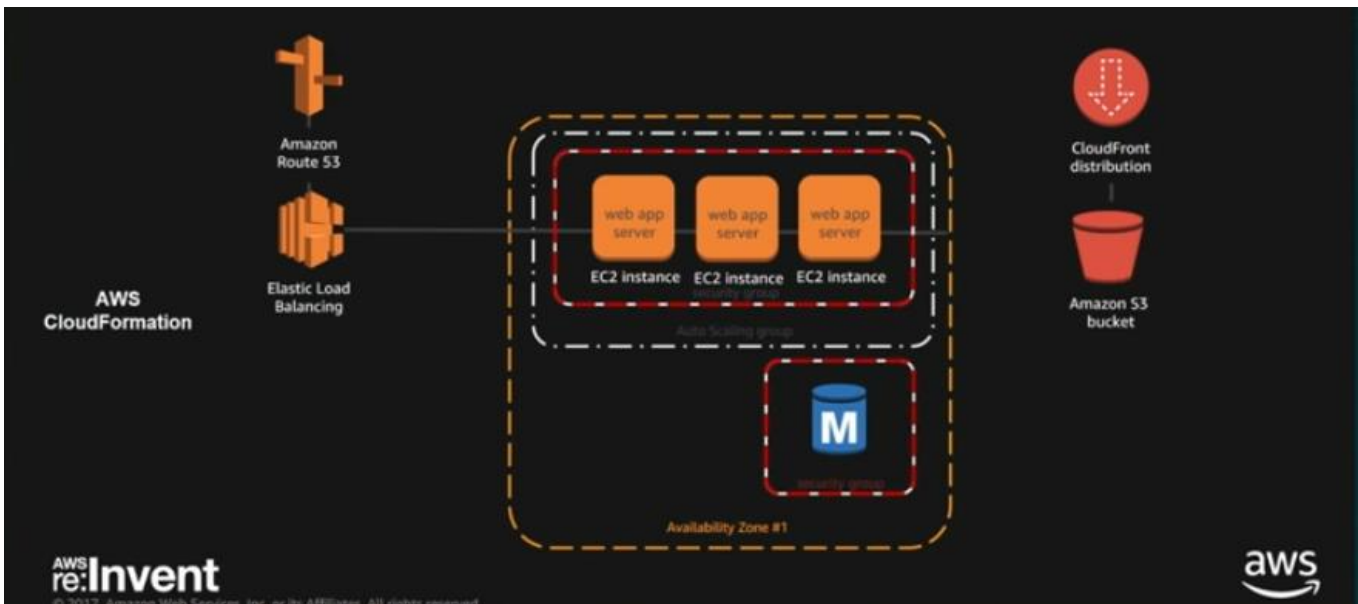
We can scale in or scale out automatically using auto-scaling groups using AWS native services as above.



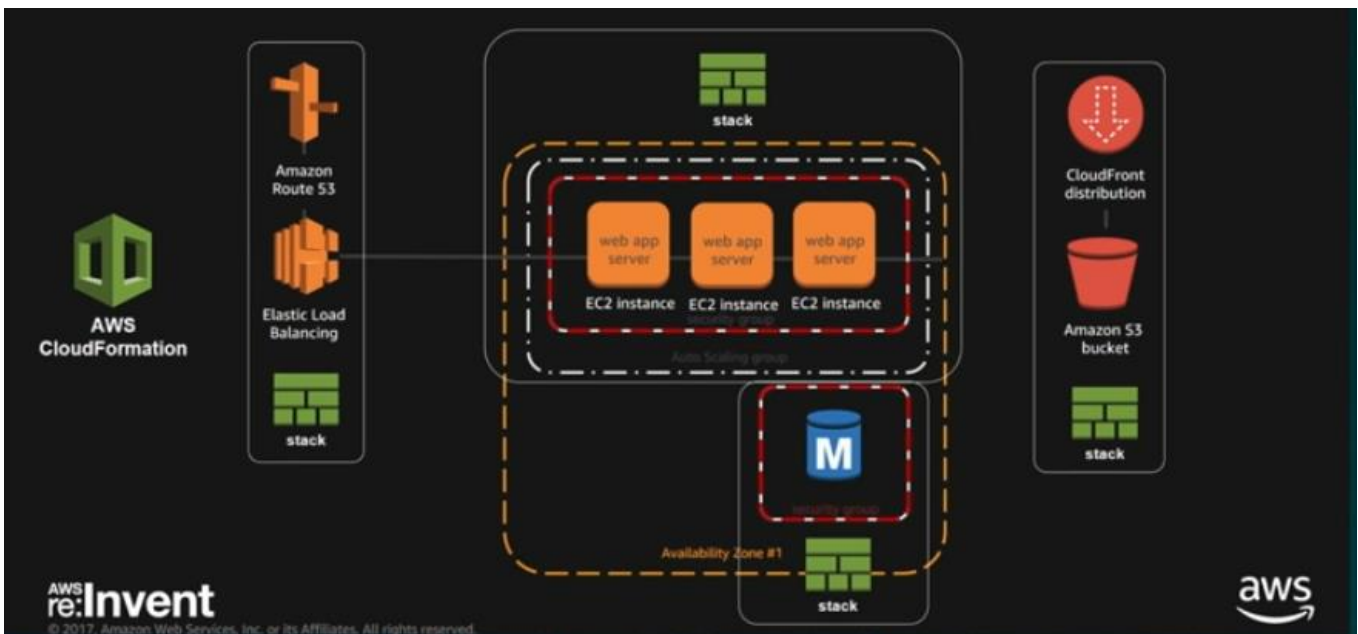
We can start to use automation and blue green deployments as above,



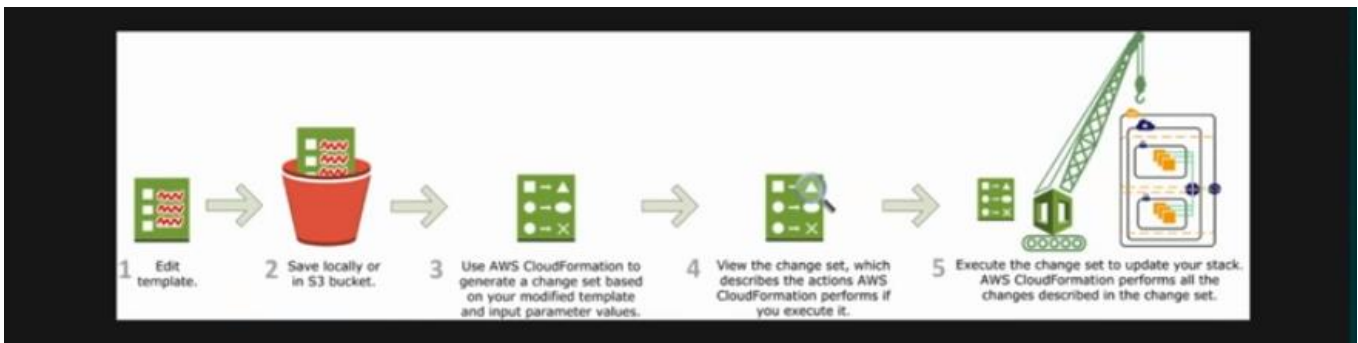
You then flip the traffic to the new stack



We can use auto-scaling group to scale our infrastructure in an automated fashion



We should be decoupling the infrastructure into stacks that are based on the responsibility of each stack application and the requirements. We have a web-server stack, database stack, CloudFront distribution stack,



This is how we can use CF to create different stacks



Anti-Pattern: Automating Outages

Anti-pattern: Incomplete Automation and Testing

Best practices:

1. Decouple stateful and stateless infrastructure management automation
2. Limit interactive access to infrastructure
3. Define and enforce tagging policy
4. Implement blue/green and rolling upgrades
5. Test infrastructure automation in non-production environments
6. Administrative domains!

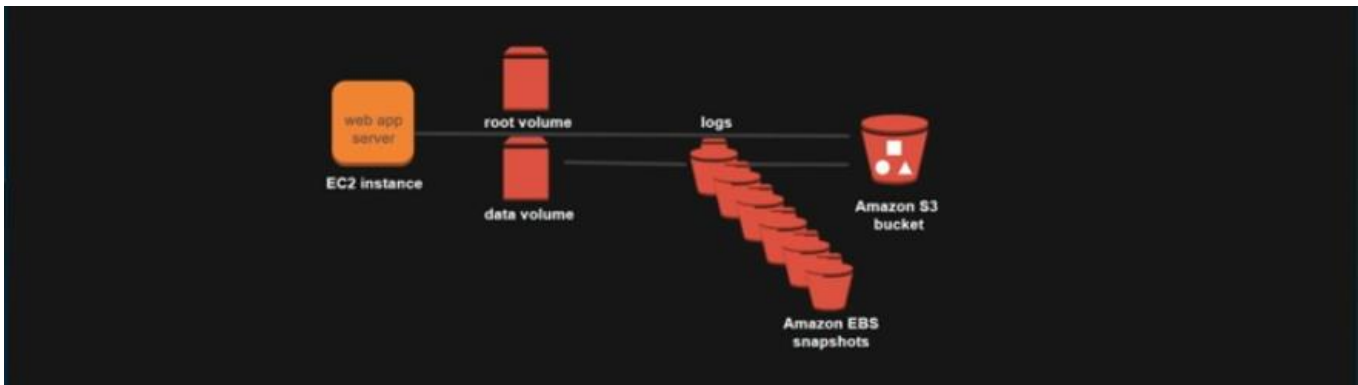
More AWS Infrastructure Automation Best Practices

<http://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/best-practices.html>

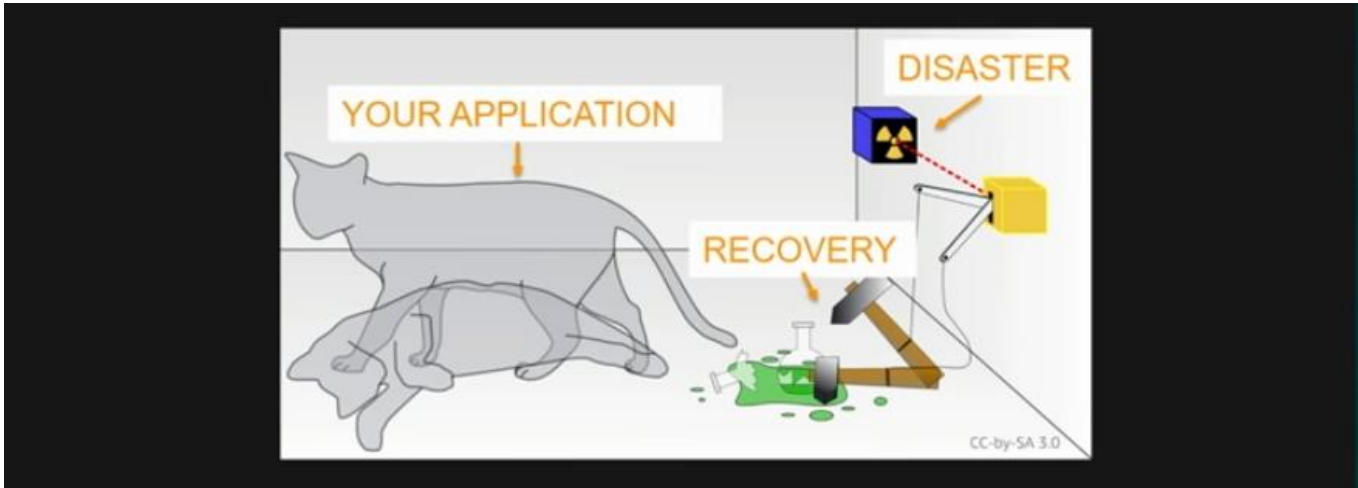
Anti-Pattern: Schrödinger's Backup



We have a sample infrastructure shown above that we want keep backed up using EBS snapshots



We then create a bunch of snapshots over time for possible recovery in an outage disaster



You need to test your backups quickly to make sure you are really covered when you need to use the backups. A test could be to make sure your backups are getting bigger in storage from yesterday to today? are you trimming older backups like 2 weeks ago? are your backups 0 KB in size? You don't need to redeploy your backups

Anti-Pattern: Schrödinger's Backup

Anti-pattern: **No Regular Recovery Testing**

Best practices:

1. Automate backups
2. Use services that include native backup features
3. Automate recovery testing
4. Alert on failure
5. Replication is not a backup

More AWS Backup and Recovery Best Practices

https://d0.awsstatic.com/whitepapers/Backup_and_Recovery_Approaches_Using_AWS.pdf

Establishing Best Practices

Establishing Best Practices

It's a journey...

1. Identify Best Practices
 - Learn from mistakes, and, ideally, the mistakes of others
 - Use FAQs, troubleshooting guides, and Backup and Recovery steps BEFORE deployment
2. Test Your Assumptions
 - Schedule Trial Restores and DR Exercises
 - War game scenarios
3. Reassess Frequently
 - Follow blogs or the What's New page for new features and announcements
 - Schedule periodic architecture reviews with AWS Solutions Architects

An Introduction to AWS Trusted Advisor



Security Partner Solutions

Protect your data with cloud-powered security

Infrastructure Security Access & Control Logging & Monitoring Configuration & Vulnerability Analysis Data Protection Cloud Security Consulting Partners

API Partners offer hundreds of industry-leading products that are equivalent, identical to, or integrate with existing controls in your on-premises environments. These products complement the existing AWS services to enable you to deploy a comprehensive security architecture and a more seamless experience across your cloud and on-premises environments.

Find below the products and solutions pre-qualified by the AWS Partner Competency Program to support you in multiple areas including: infrastructure security, policy management, identity management, security monitoring, vulnerability management, data protection, and consulting services.

<https://aws.amazon.com/security/partner-solutions/>

The screenshot shows the AWS Well-Architected framework page. The top navigation bar includes links for Menu, aws, Contact Sales, Products, Solutions, Pricing, and More. On the right, there are links for English, My Account, and a Sign In to the Console button. The left sidebar lists various AWS resources, with 'AWS Well-Architected' highlighted. The main content area is titled 'AWS Well-Architected' and includes a brief description of the framework. Below this, there are four columns, each with an icon and a title: 'Build and deploy faster' (lightning bolt icon), 'Lower or mitigate risks' (scales icon), 'Make informed decisions' (lightbulb icon), and 'Learn AWS best practices' (thumbs up icon). Each column contains a short paragraph of text.

Menu aws Contact Sales Products Solutions Pricing More English My Account Sign In to the Console

RECENT AWS

AWS Well-Architected

RECENT LINKS

- AWS Well-Architected
- AWS Economics Center
- Security & Compliance
- AWS Products & Services
- AWS Solutions
- Case Studies

Manage Your Resources

Sign In to the Console

AWS Well-Architected

The Well-Architected framework has been developed to help cloud architects build the most secure, high-performing, resilient, and efficient infrastructure possible for their applications. This framework provides a consistent approach for customers and partners to evaluate architectures, and provides guidance to help implement designs that will scale with your application needs over time.



Build and deploy faster

Stop guessing capacity needs, test systems at scale, and use automation to make experimentation easier by building cloud-native architectures.



Lower or mitigate risks

Understand where you have risks in your architecture, and address them before your applications are put into production.



Make informed decisions

Determine how architectural decisions and/or trade-offs might impact the performance and availability of your applications and business outcomes.



Learn AWS best practices

Access training and whitepapers that provide guidance based on what we have learned through reviewing thousands of customer architectures on AWS.

<https://aws.amazon.com/architecture/well-architected/>

GPSTEC302

AWS
re:Invent

THANK YOU!

AWS
re:Invent

© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

