Cloud Computing Architecture

Infrastructure As Code

SWINBURNE UNIVERSITY OF TECHNOLOGY

Infrastructure As Code

This presentation:

- Why Automate Infrastructure
- Infrastructure as Code
- Infrastructure as Code on AWS

ClickOps

- Click Operations
- Several clicks in the AWS Management Console to deploy infrastructure
- Manual process

Another option: Launch an EC2 instance with the AWS Command Line Interface

- EC2 instances can also be created programmatically.

AWS Command Line Interface (AWS CLI)

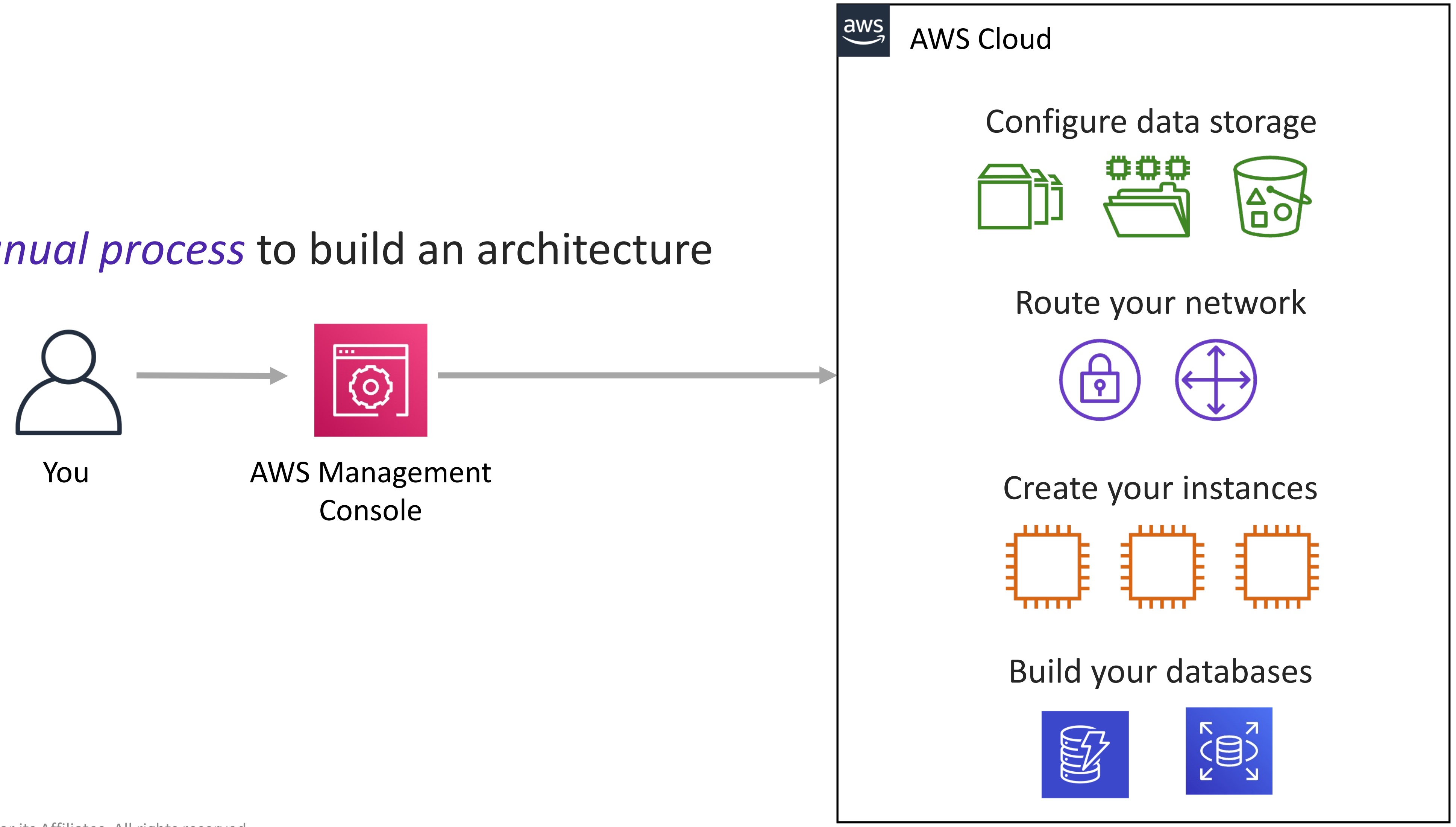
- This example shows how simple the command can be.
  - This command assumes that the key pair and security group already exist.
  - More options could be specified. See the AWS CLI Command Reference for details.

Example command:

```
aws ec2 run-instances \
--image-id ami-1a2b3c4d \
--count 1 \
--instance-type c3.large \
--key-name MyKeyPair \
--security-groups MySecurityGroup \
--region us-east-1
```

Without automation

Long *manual process* to build an architecture



Risks from manual processes

Does not support repeatability at scale

- How will you replicate deployments to multiple Regions?

No version control

- How will you roll back the production environment to a prior version?

Lack of audit trails

- How will you ensure compliance? How will you track changes to configuration details at the resource level?

Inconsistent data management

- For example, how will you ensure matching configurations across multiple Amazon Elastic Compute Cloud (Amazon EC2) instances?

Complying with AWS Well-Architected Framework principles

- Operational excellence design principles
  - Perform operations as code
  - Make frequent, small, reversible changes

- Reliability pillar design principles
  - Manage change in automation

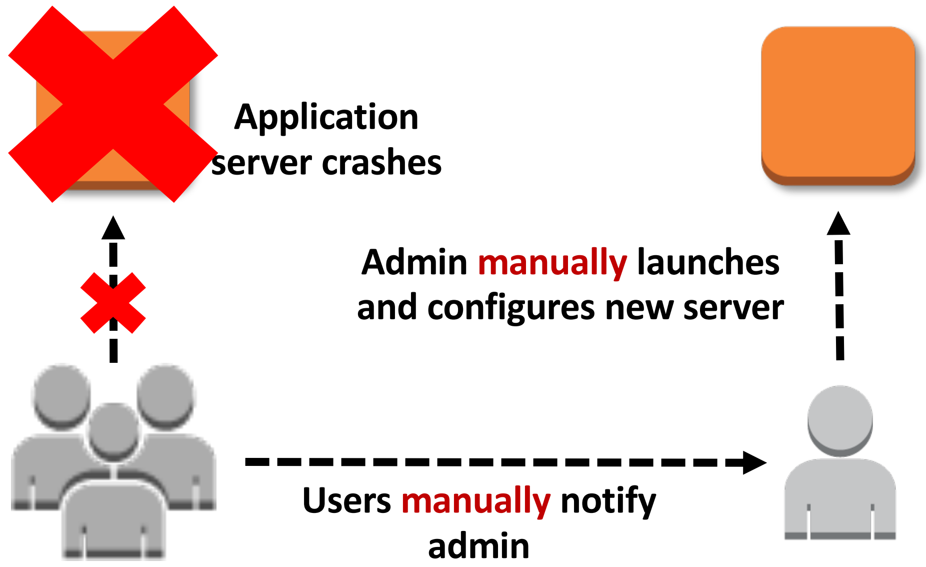
Creating and maintaining AWS resources and deployments by following a manual approach does not enable you to meet these guidelines.

Best Practice: Automate Your Environment

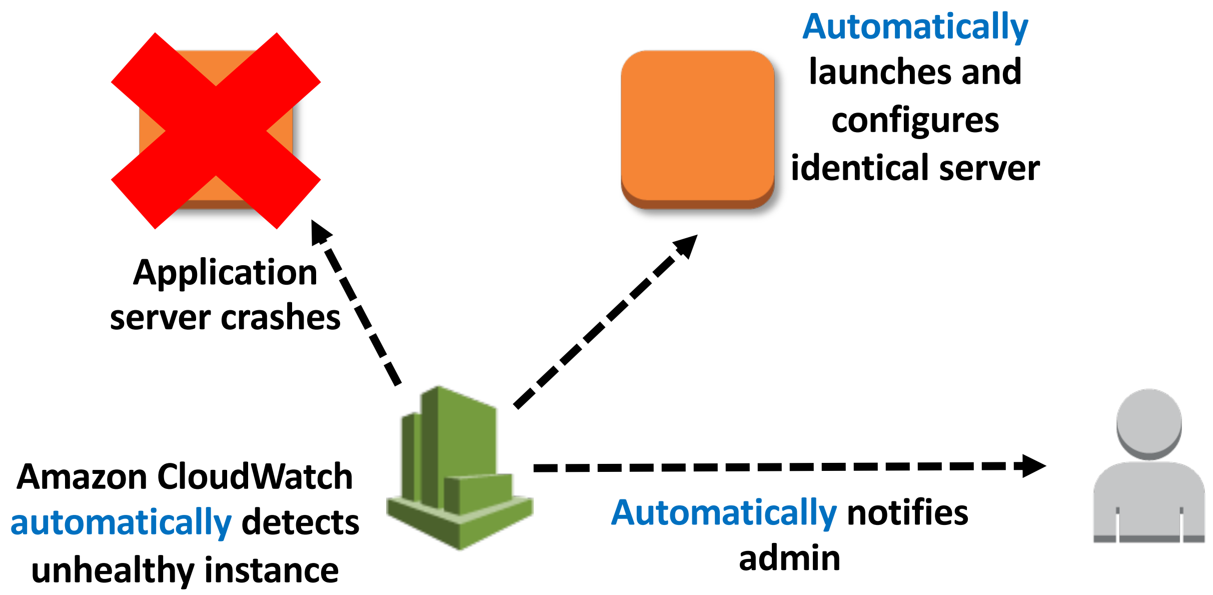
Where possible, automate the provisioning, termination, and configuration of resources.

Improve your system's **stability** and **consistency**, as well as the **efficiency** of your organization, by removing manual processes.

Anti-pattern


Application server crashes
Admin manually launches and configures new server
Users manually notify admin

Best practice


Automatically launches and configures identical server
Application server crashes
Amazon CloudWatch automatically detects unhealthy instance
Automatically notifies admin

Best Practice: Use Disposable Resources

Take advantage of the dynamically provisioned nature of cloud computing.

Treat servers and other components like **temporary resources**.

Anti-pattern

- Over time, different servers end up in different configurations.
- Resources run when not needed.
- Hardcoded IP addresses prevent flexibility.
- Difficult/inconvenient to test new updates on hardware that's in use.

Best practice

- Automate deployment of new resources with identical configurations.
- Terminate resources not in use.
- Switch to new IP addresses automatically.
- Test updates on new resources, and then replace old resources with updated ones.

What Does Infrastructure as Code Mean?

Automating your infrastructure:

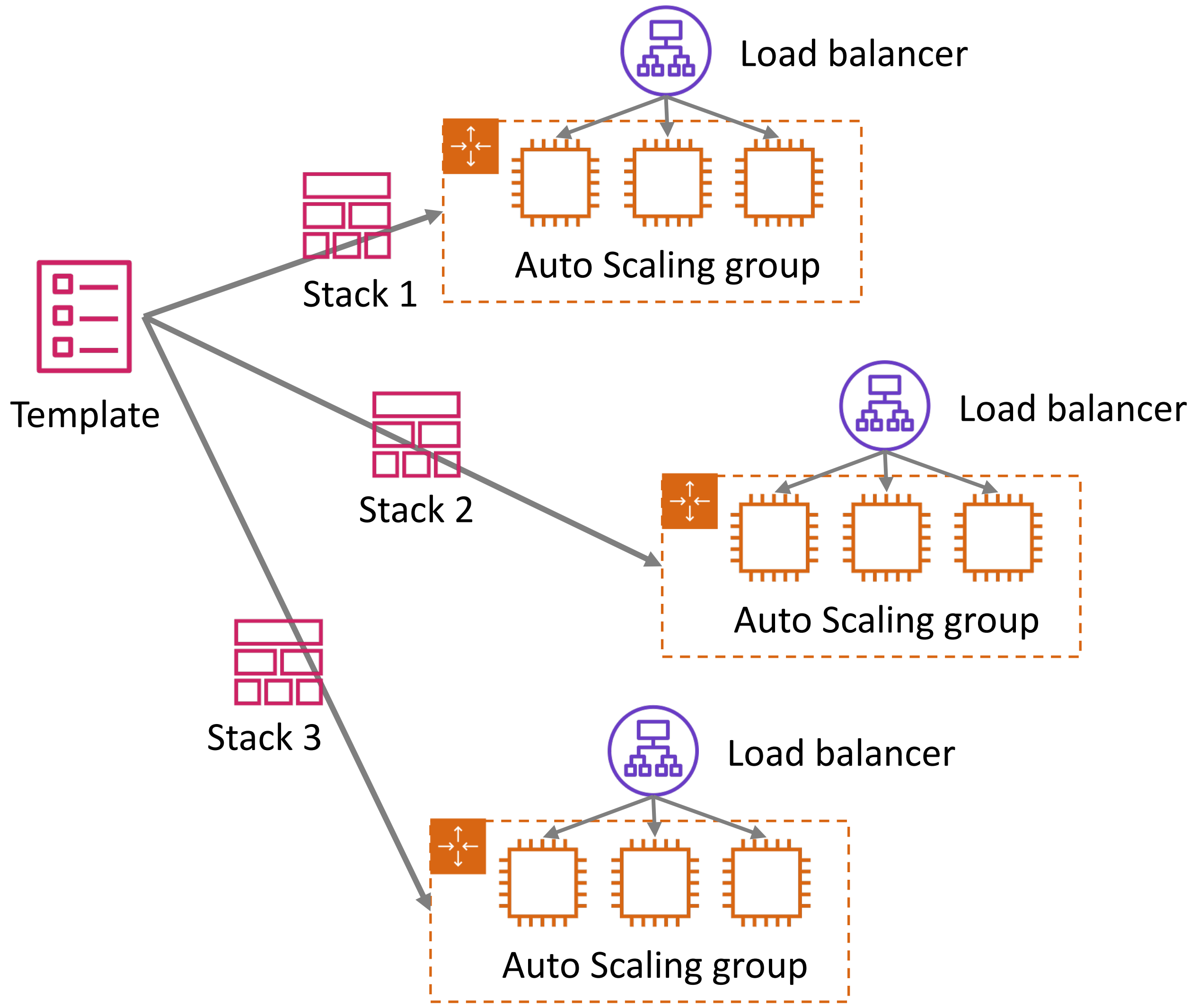
Define your infrastructure as **code**, not as bundles of hardware components.

Process of applying techniques, practices, and tools from **software development** to create **reusable**, **maintainable**, **extensible,** and **testable** infrastructure.

More Benefits of IaC

- AWS is much more reluctant to break automation
- Breaking changes are made much more regularly to the AWS Management Console than to automation
- Doing it manually using ClickOps – can see need to use new way

Infrastructure as code: Benefits



Reduce multiple matching environments

- Rapid deployment of complex environments
- Provides configuration consistency
- Simple clean up when wanted (deleting the stack deletes the resources created)
- Easy to propagate a change to all stacks
  - Modify the template, run update stack on all stacks

Benefits

- Reusability
- Repeatability
- Maintainability

Infrastructure as Code on AWS - CloudFormation

SWINBURNE UNIVERSITY OF TECHNOLOGY

AWS CloudFormation: Infrastructure as Code

Allows you to **launch, configure, and connect AWS resources** with JavaScript Object Notation (JSON) or YAML-formatted templates

Template

- JSON or YAML-formatted file describing the resources to be created
- Treat it as source code: put it in your repository

AWS CloudFormation Engine

- AWS service component
- Interprets AWS CloudFormation template into stacks of AWS resources

Stack

- A collection of resources created by AWS CloudFormation
- Tracked and reviewable in the AWS Management Console
- Cross stack references

Ways to Work with AWS CloudFormation Templates

- Simple JSON or YAML text editor
- CloudFormation Designer
  - Is available via the AWS Management Console.
  - Lets you drag and drop resources onto a design area to automatically generate a JSON-formatted or YAML-formatted CloudFormation template.
  - Edit the properties of the JSON or YAML template on the same page.
  - Open and edit existing CloudFormation templates using the CloudFormation Designer tool.

More ways to Generate CloudFormation

SWINBURNE UNIVERSITY OF TECHNOLOGY

CDK for CloudFormation

- Cloud Development Kit
- Use a programming language (e.g. Typescript, Python, Java, .NET)
- Programmatically generate your CloudFormation templates
- Can be very short amount fo c

Lecture References

SWINBURNE UNIVERSITY OF TECHNOLOGY

Recommend Viewing

Swinburne Lecture – High Level Overview

AWS Academy – Deeper dive

ACA Module 10