

# Part 1: Application Builds 101

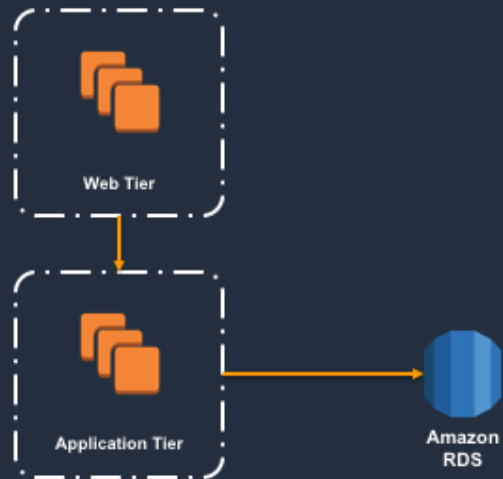
Day 1

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark



## Architecture Overview

- 3-tier web application.
- Focus will be on application deployment.



© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark.



Oversimplified view – in reality there would be VPCs, AZs,  
Focus will be on web applications

## The traditional approach

- ...what is the traditional approach?



© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark



Image source: <https://02varvara.wordpress.com/2014/01/19/images-from-the-kiev-riot-its-time-to-put-these-feral-rebels-down/>

## It's not all doom and gloom

- Build and packaging tools have come a long way.
- Dependency hell is becoming easier to manage.
- A lot of this knowledge is transferrable within the industry.

## Demo application

- Taylor has too many cats!
- If only there was a way of keeping track of them...

What are Linux engineers most afraid of?

## Demo application



© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark.



## Demo application

- .NET Core 2.1 Web API
- ~150 lines
- Stores data using SQLite

```
11 [Route("api/[controller]")]
12 public class CatsController : Controller
13 {
14     // GET api/Cats
15     [HttpGet]
16     public IEnumerable<Cat> Get()
17     {
18         using (CatDb db = new CatDb())
19         {
20             return db.Cats.ToList();
21         }
22     }
23
24     // GET api/Cats/5
25     [HttpGet("{id}")]
26     public Cat Get(int id)
27     {
28         using (CatDb db = new CatDb())
29         {
30             return db.Cats.First(t => t.Id == id);
31         }
32     }
33
34     // POST api/Cats
35     [HttpPost]
36     public void Post([FromBody]JObject value)
37     {
38         Cat posted = value.ToObject<Cat>();
39         using (CatDb db = new CatDb())
40         {
41             db.Cats.Add(posted);
42         }
43     }
44 }
```



# Demo

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark



Demo with Postman

## Activity 2

- Create a CloudFormation stack from the `infrastructure.json` template in the `ap-southeast-2` region.
  - <https://s3-ap-southeast-2.amazonaws.com/syd-deployment-training/infrastructure.json>

# Section 1 Review

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark



## Part 2: Continuous Integration

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark



## What is Continuous Integration?

- CI entails both culture and tools.
- Developers commit small changes regularly.
- Tests are run as part of the automated build process, identifying issues sooner.



© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark



CI is integrating, continuously!

CI is a way of thinking. Small changes merged sooner reduce the effort involved in resolving merge conflicts.

## Why CI?

- Smaller changes are generally easier to merge.
- Breaking changes are detected sooner, and are easier to strip out.
- Developers spend less time performing manual unit testing.
- Incremental updates can be delivered faster and more frequently to customers.

## CI Tools



**Jenkins**

**Hudson**



 **circleci**



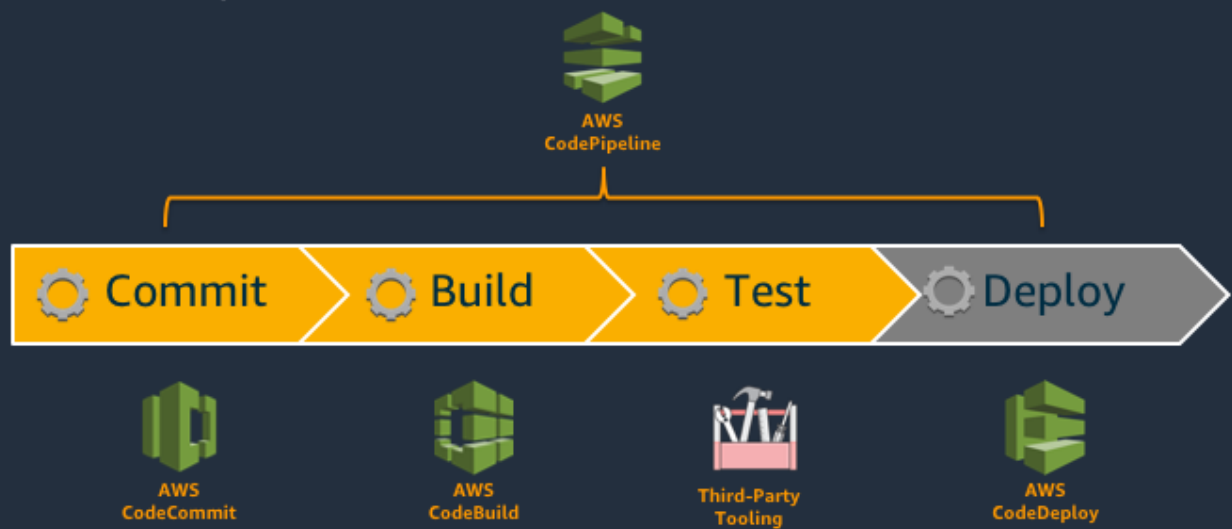
**Bamboo**

**TC**

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark.



## AWS Developer Tools



© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark.





## CI in 3 easy steps

1. Put code somewhere accessible.
2. Implement builds and tests
3. Automate!

## AWS CodeCommit

AWS CodeCommit is a **fully-managed source control** service that makes it easy for companies to host secure and highly scalable **private Git repositories**.

### Key Benefits:

- **Secure**
- **Highly available**
- **Scalable**
- **Faster development lifecycle**



© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark



**Secure** –AWS CodeCommit automatically encrypts your files in transit (HTTPS/SSH) and at rest (via KMS using customer-specific keys). AWS CodeCommit is integrated with AWS Identity and Access Management (IAM), allowing you to assign user-specific permissions to your repositories.

**Highly Available** – AWS CodeCommit is built on highly scalable, redundant, and durable AWS services such as Amazon S3 and Amazon DynamoDB.

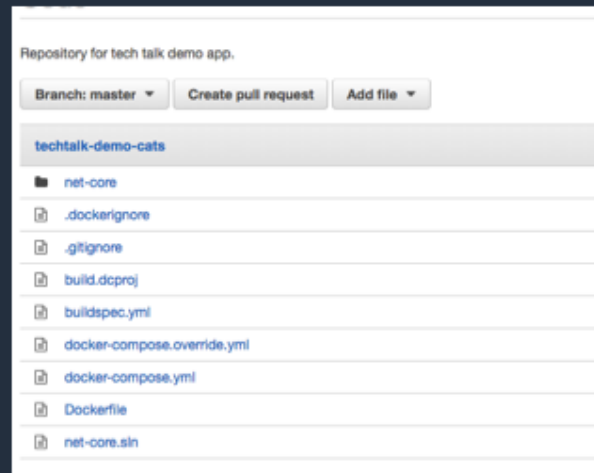
**Scalable** - AWS CodeCommit allows you store any number of files and there are no repository size limits.

**Faster Development Lifecycle** - AWS CodeCommit keeps your repositories close to your build, staging, and production environments in the AWS cloud. This allows you to increase the speed and frequency of your development lifecycle.

## CodeCommit Demo + Activity 3

1. Create a CodeCommit repository.
2. Configure an IAM user with Git SSH/HTTPS credentials.
3. Push application to repository.

Check the repository in the CodeCommit console. The application code should be visible there.



© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark



Instructions for configuring Git credentials for an IAM user can be found here:

<https://docs.aws.amazon.com/codecommit/latest/userguide/setting-up-gc.html>

Use a graphical Git client (e.g. Sourcetree, Visual Studio), or use Git on the command line.

<https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>

## CI in 3 easy steps

1. Put code somewhere accessible.
2. **Implement builds and tests**
3. Automate!

## AWS CodeBuild

AWS CodeBuild is a **fully managed build service** that compiles source code, runs tests, and produces software packages that are ready to deploy.

### Key benefits:

- **Fully managed build service**
- **Continuous scaling**
- **Pay as you go**
- **Extensible**
- **Secure**



© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark



**Fully Managed Build Service** - AWS CodeBuild eliminates the need to set up, patch, update, and manage your own build servers and software. There is no software to install or manage.

**Continuous Scaling** - AWS CodeBuild scales automatically to meet your build volume. It immediately processes each build you submit and can run separate builds concurrently, which means your builds are not left waiting in a queue.

**Pay as You Go** - With AWS CodeBuild, you are charged based on the number of minutes it takes to complete your build. This means you no longer have to worry about paying for idle build server capacity.

**Extensible** - You can bring your own build tools and programming runtimes to use with AWS CodeBuild by creating customized build environments in addition to the prepackaged build tools and runtimes supported by CodeBuild.

**Secure** - CodeBuild integrates with IAM and KMS to authenticate to AWS services, and encrypt build artifacts.

## CodeBuild Demo + Activity 4

1. Create a CodeBuild project.
2. Image: Ubuntu / .NET Core 2.1
3. Buildspec: buildspec-dotnet.yml
4. Allow CodeBuild to create the service role to access CodeCommit.

Test your build project by starting a build.

Build ARN	arn:aws:codebuild:ap-southeast-2:358529147491:build/techtalk-5-83db-1201946d4a8a
Build project	techtalk-demo-cats
Source provider	AWS CodePipeline
Repository	
Start time	Jun 25, 2018 8:41:03 AM UTC
End time	Jun 25, 2018 8:43:44 AM UTC
Status	Succeeded
Initiator	techtalk-demo-cats

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark



Source: CodeCommit

Image: Ubuntu / .NET Core / version 2.1

Buildspec: buildspec-dotnet.yml

Artifacts:

Type: S3

Name: /

Namespace ID: Build ID

Bucket Name: <S3 bucket created by CloudFormation>

## CI in 3 easy steps

1. Put code somewhere accessible.
2. Implement builds and tests
3. **Automate!**

# AWS CodePipeline

AWS CodePipeline is a **continuous integration** and **continuous delivery** service for fast and reliable application and infrastructure updates.

## Key benefits:

- Fully managed
- AWS integrations
- Plugins
- Declarative templates
- Parallel executions
- Access control



© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark



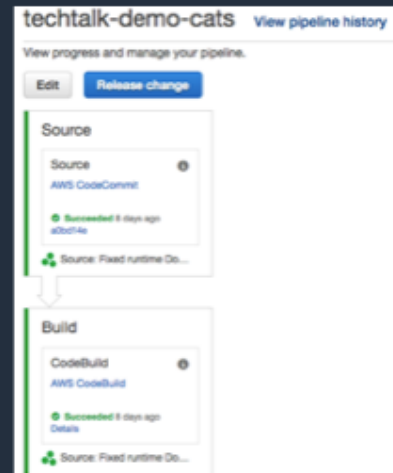
CodeCommit/S3/CodeBuild/CodeDeploy/ECS/OpsWorks

Plugins for GitHub, CloudBees, Jenkins, TeamCity, BlazeMeter, Storm Runner, Ghost Inspector, Novovola, Runscope, XebiaLabs



## CodePipeline Demo + Activity 5

1. Create a new pipeline.
2. Source: CodeCommit
3. Build: CodeBuild
  - Image: Ubuntu / Docker / <latest>
  - Specify environment variables:
    - AWS\_ACCOUNT\_ID
    - IMAGE\_REPO\_NAME
4. Deployment: No Deployment
5. Add the AmazonEC2ContainerRegistryPower User policy to the CodeBuild role.



© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark



Most parameters as per the previous build project, except artifacts aren't required.

AWS\_ACCOUNT\_ID can be found in the support center. IMAGE\_REPO\_NAME is the value of the ecsRepo output of the CloudFormation stack that was created previously.

After defining the CodeBuild project, you will need to add the ECR managed policy to the role created for the CodeBuild project. You can do this through the IAM console.

## Putting it all together

- Commit and push a change to CodeCommit.
- The pipeline should automatically trigger, resulting in CodeBuild executing a build.
- If all goes well, the pipeline execution should succeed.

# Section 2 Review

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark

