

- **Version Control Systems**
  - Advanced Git concepts
  - Common branching strategies and when is better to use each one depending on the project situation and team configuration
  - Git best practices
    - Merging versus Rebasing - Advantages and disadvantages
    - Keeping a clean history
      - Stashing
      - Commit comments
    - Semantic versioning

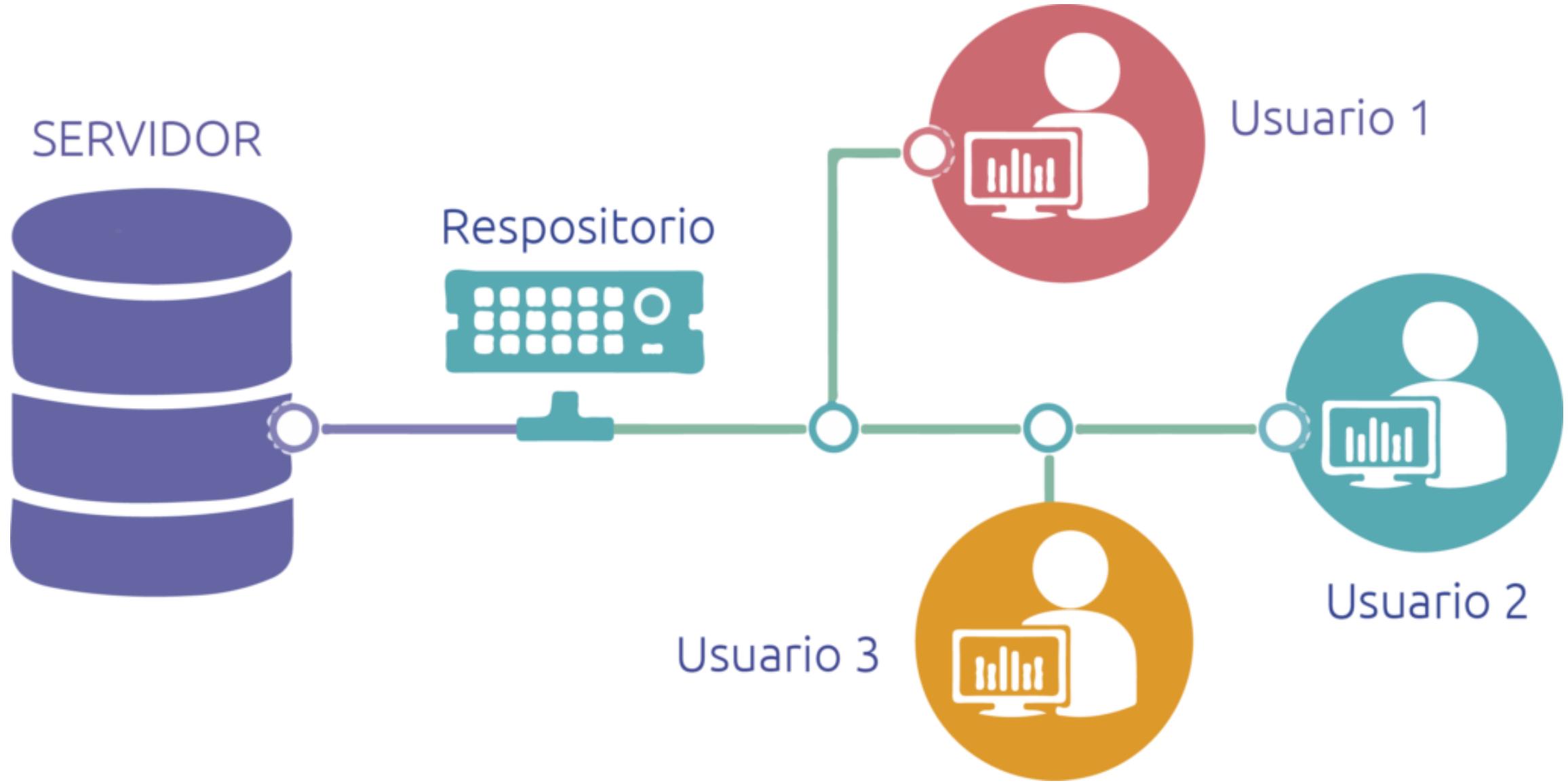
- **Continuous Integration (CI)**
  - Principles of continuous integration
  - Creating a build pipeline for a dotnet solution using yaml configuration files tracked in the repository
  - Unit test integration
  - Pipeline triggers

- **AWS Fundamentals**
  - Introduction to AWS (not detailed, most of the students will have this knowledge)
  - Basic AWS for deploying a dotnet solution
    - EC2 instances
    - S3
    - Lambdas
    - AWS Secret Manager
    - Cloud Watch

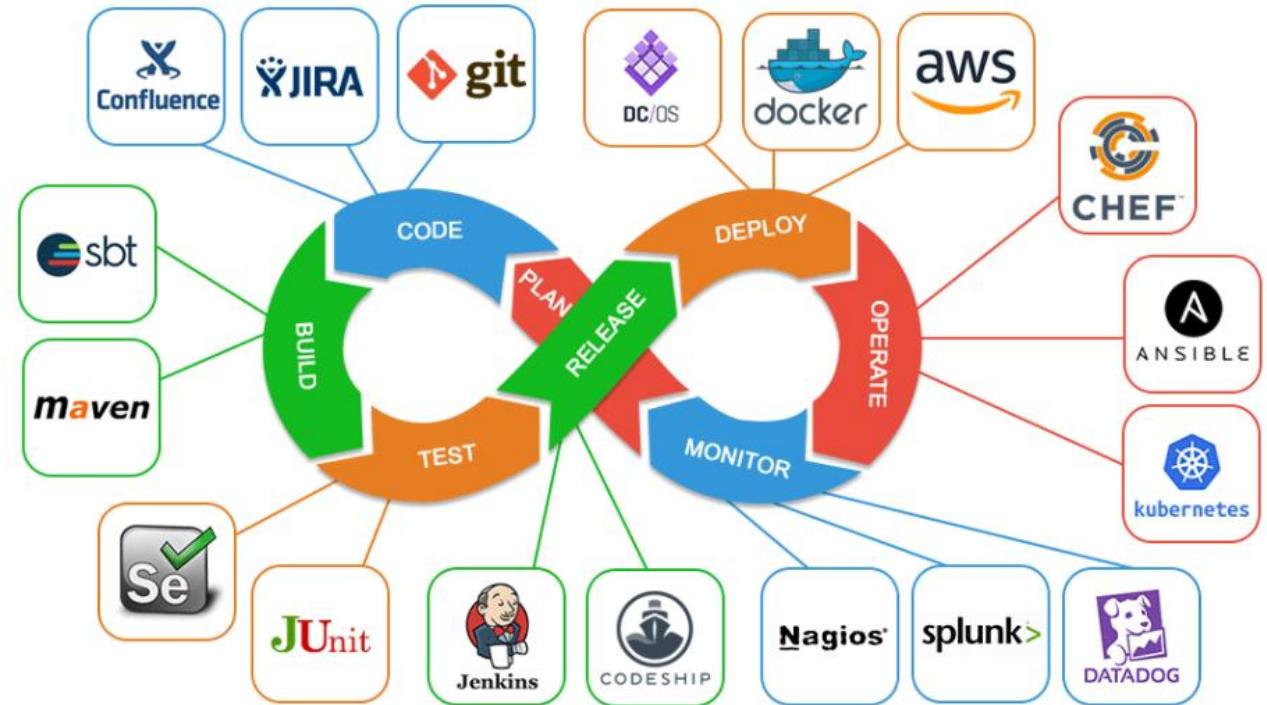
- **Continuous Delivery (CD)**
  - Principles of continuous delivery and deployment
  - Creating a Release pipeline using Jenkins or Github Actions
    - Using the artifacts generated from the build pipeline
    - Deploy into different environment
    - Approvals and gatekeepers
    - HealthChecks
    - Blue/Green deployments
    - Canary deployments
- **Introduction of Infrastructure as Code (IaC)**
  - Basics of Infrastructure as Code
  - Introduction to Terraform

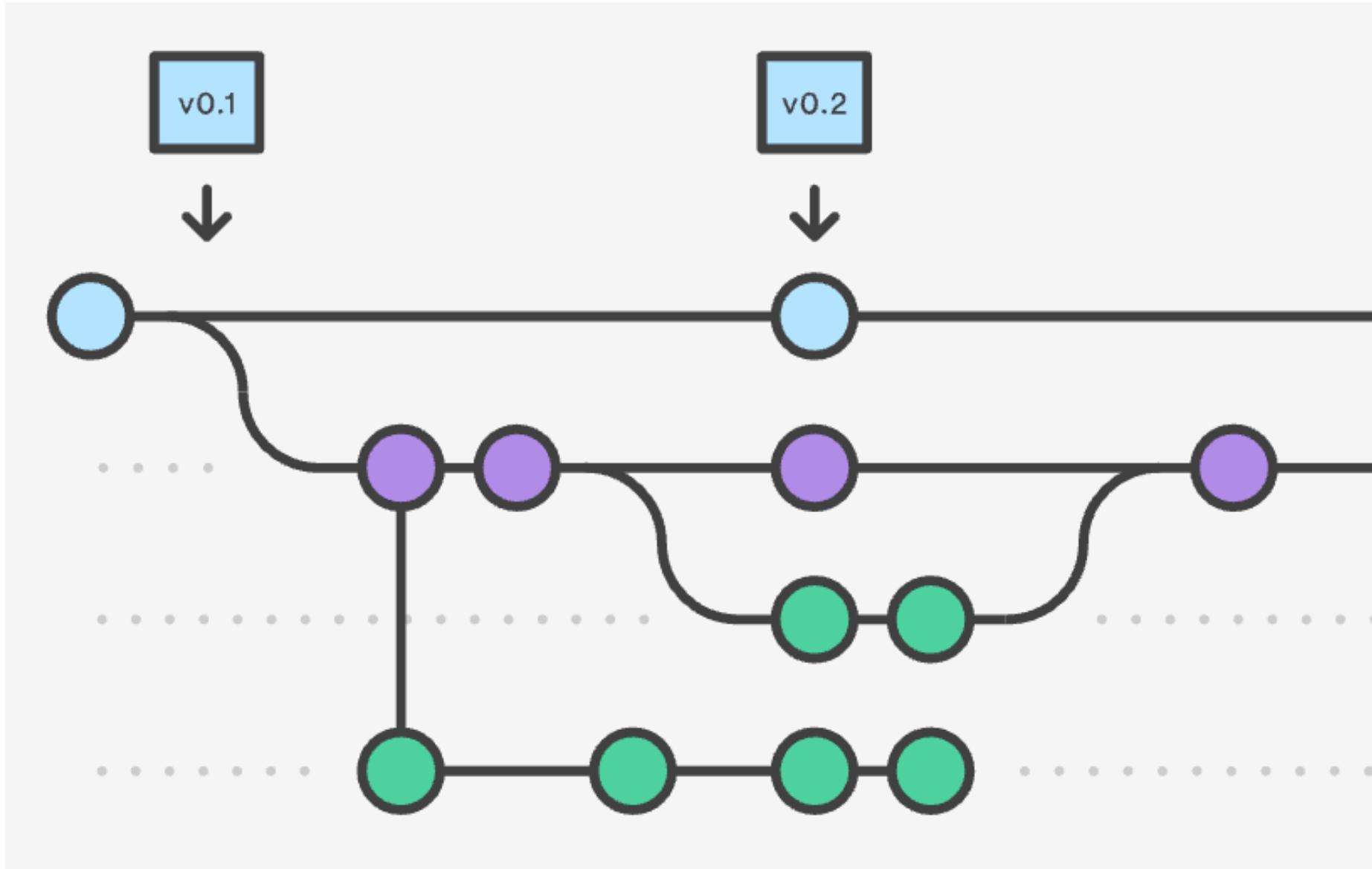


# INTRODUCE YOURSELF



# DEV OPS





# Local Computer

Checkout

File

Version Database

Version 3

Version 2

Version 1

Computer A

File

Computer B

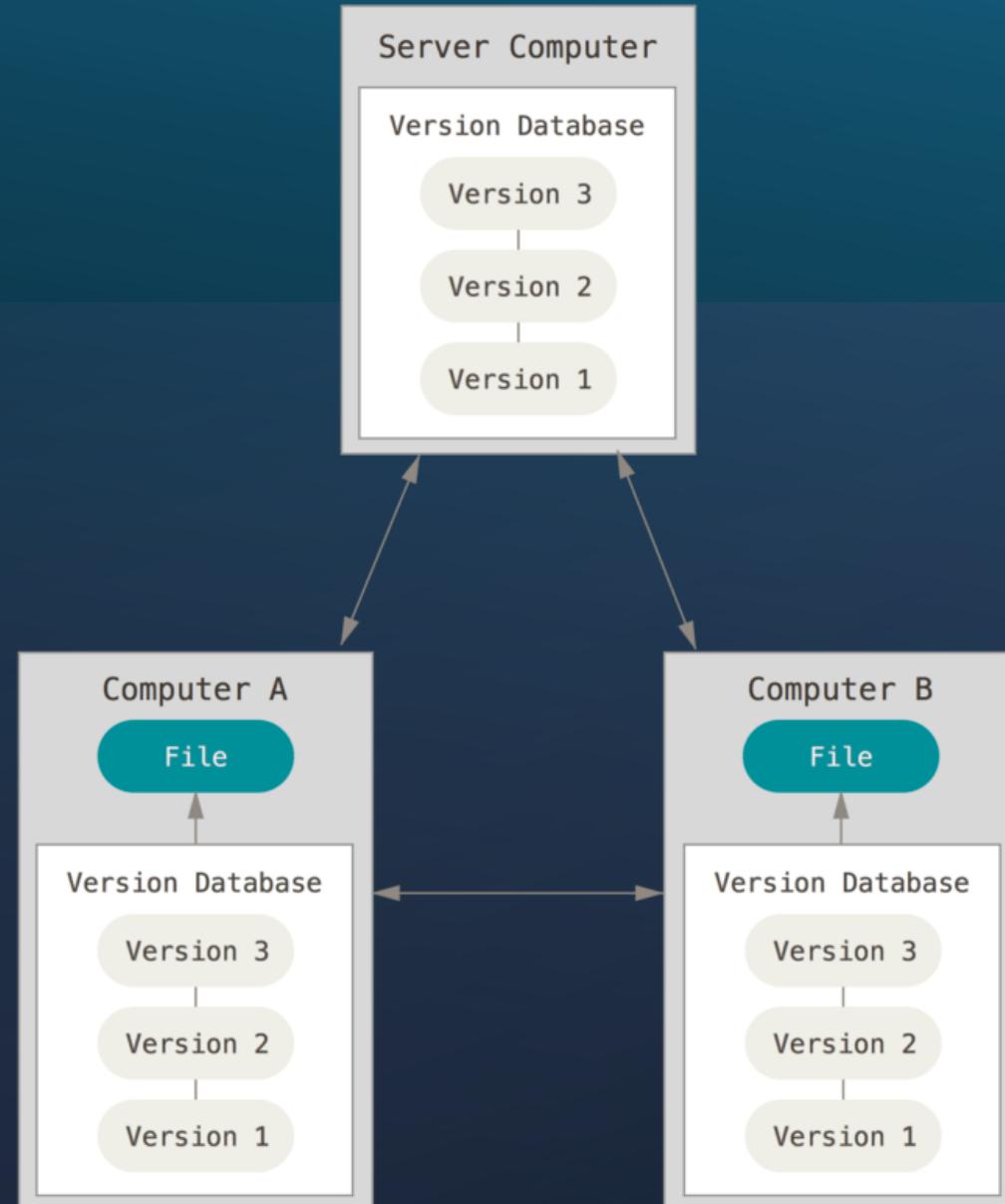
Central VCS Server

Version Database

Version 3

Version 2

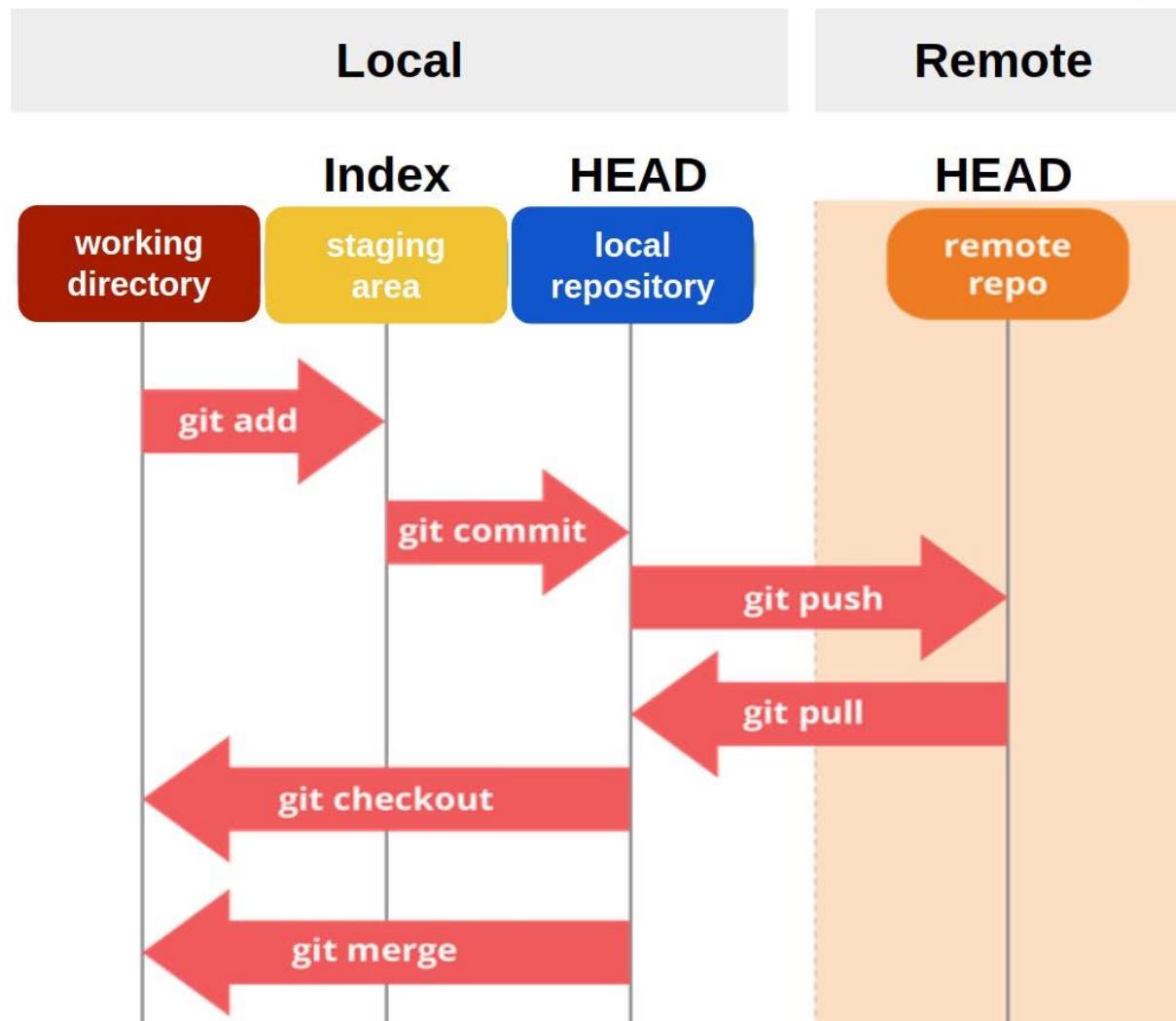
Version 1

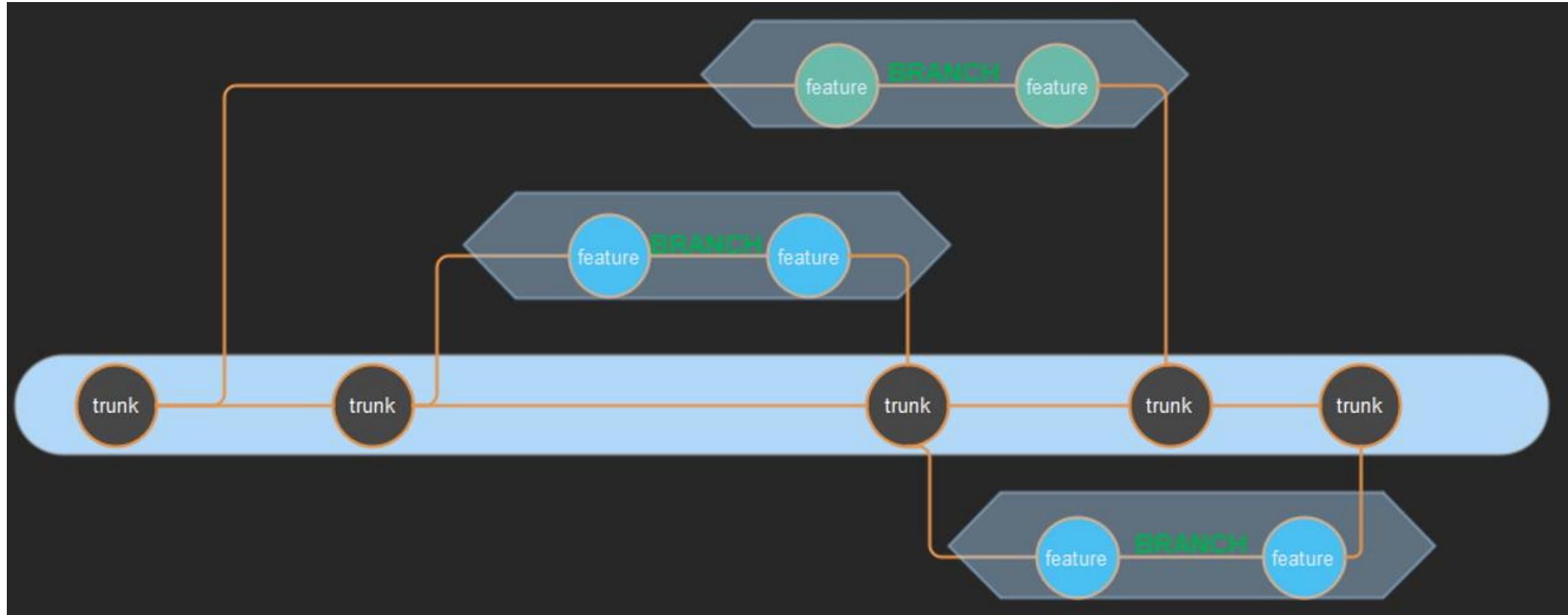




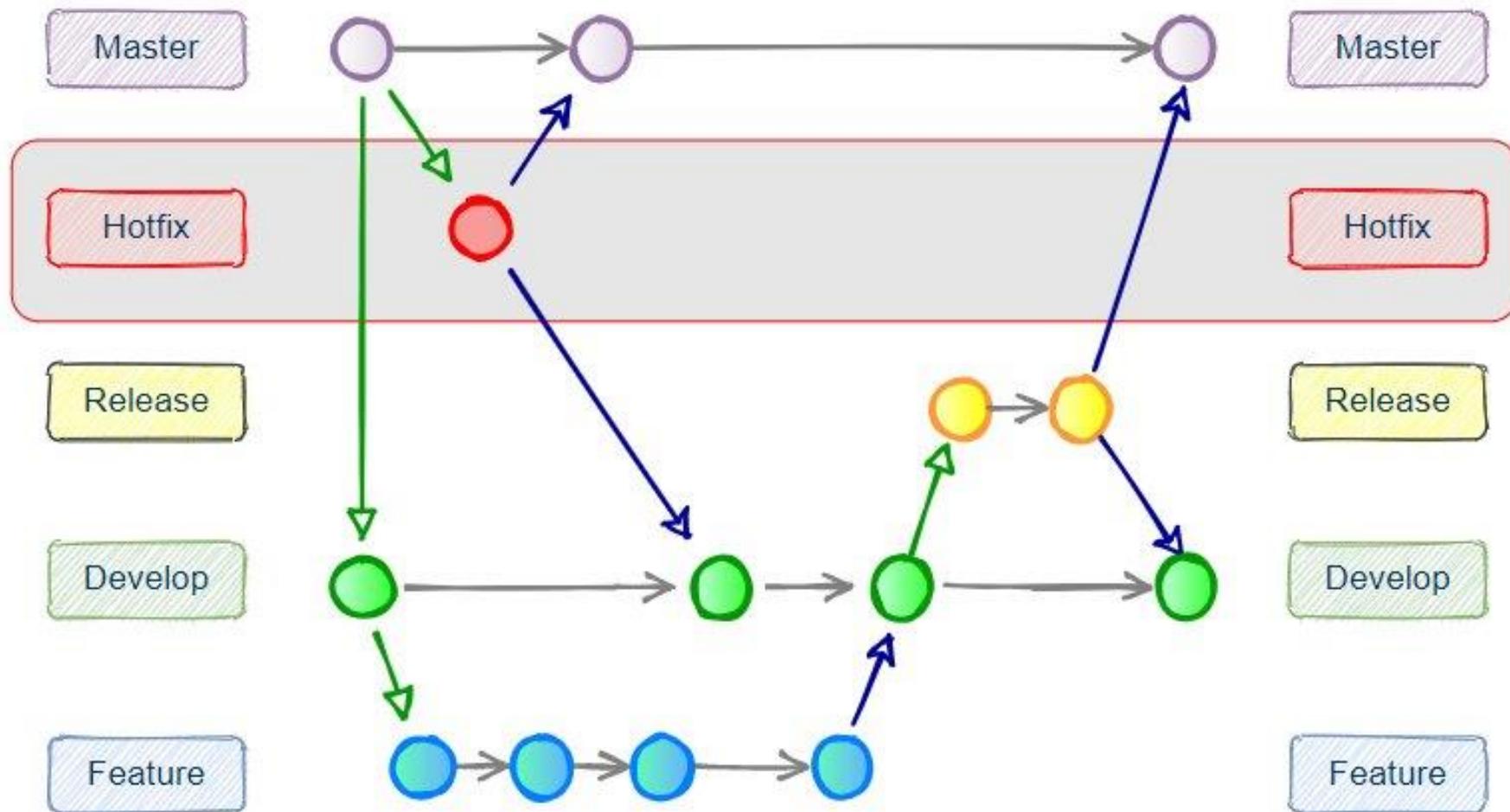


git





# Gitflow Workflow Diagram



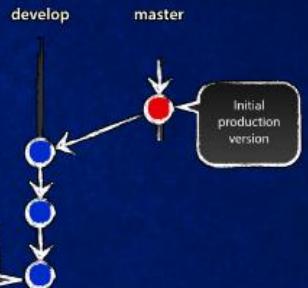
# DEVELOP

## Git

```
# Start develop branch  
git checkout -b develop master
```

## Git Flow

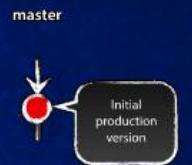
```
# Initialize flow  
git flow init
```



# MASTER

## Git

```
# New repo  
git init  
  
# Clone repo  
git clone git@server.name:repo.git
```



# FEATURE/\*

## Git

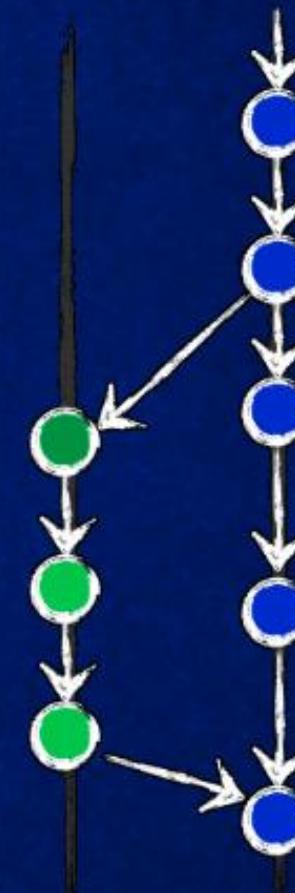
```
# Creación  
git checkout -b feature/lorem-ipsum develop  
  
# Finalización  
git checkout develop  
git merge --no-ff feature/lorem-ipsum  
git branch -d feature/lorem-ipsum  
git push origin develop
```

## Git Flow

```
# Creación  
git flow feature start lorem-ipsum  
  
# Finalización  
git flow feature finish lorem-ipsum
```

feature  
branches

develop



# RELEASE/VERSION

# Git

```
# Creación  
git checkout -b release/v2.3.0 develop
```

```
# Finalización  
git checkout master  
git merge --no-ff release/v2.3.0  
git tag -a v2.3.0
```

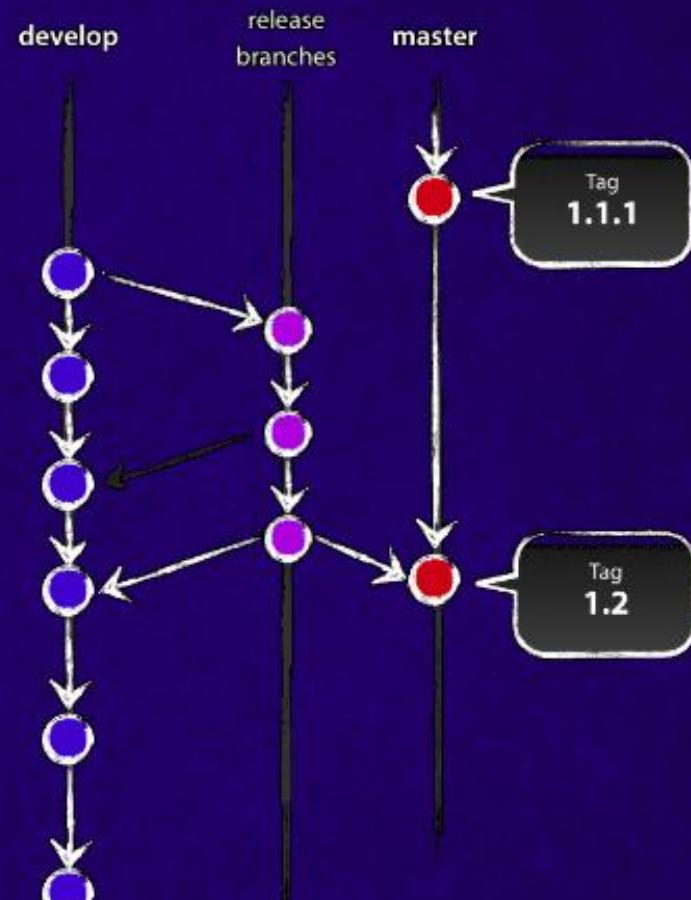
```
git checkout develop  
git merge --no-ff release/v2.3.0
```

```
git branch -d release/v2.3.0
```

## Git Flow

```
# Creación  
git flow release start v2.3.0
```

```
# Finalización  
git flow release finish v2.3.0
```



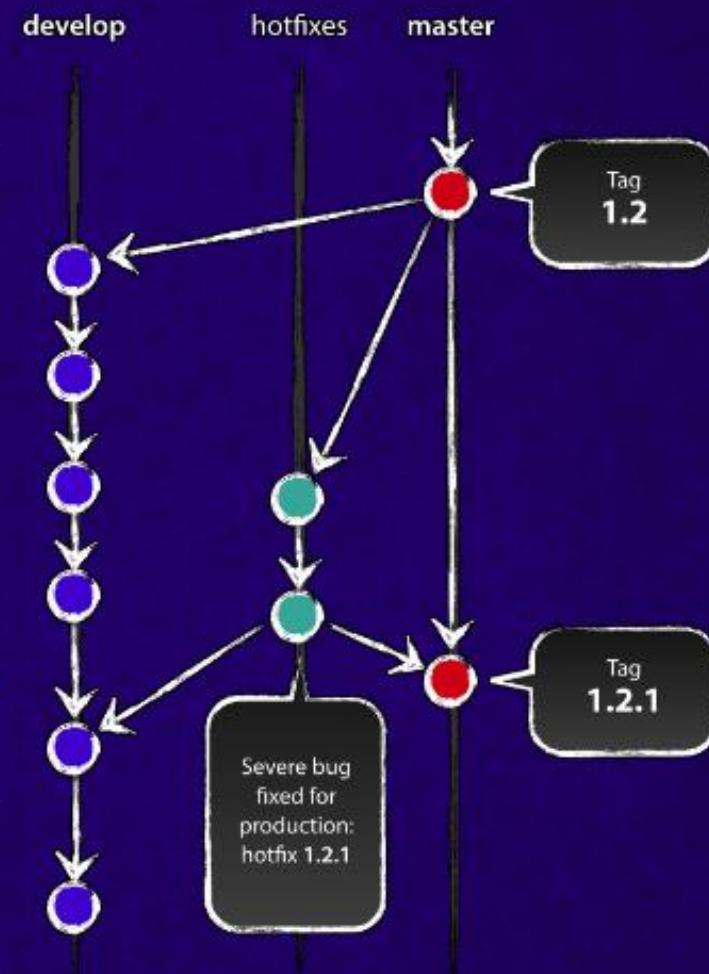
# HOTFIX/«VERSION»

## Git

```
# Creación  
git checkout -b hotfix/v2.3.7 master  
  
# Finalización  
git checkout master  
git merge --no-ff hotfix/v2.3.7  
git tag -a v2.3.7  
  
git checkout develop  
git merge --no-ff hotfix/v2.3.7  
  
git branch -d hotfix/v2.3.7
```

## Git Flow

```
# Creación  
git flow hotfix start v2.3.7  
  
# Finalización  
git flow hotfix finish v2.3.7
```



# EXERCISE 1

# Advanced Git Concepts

**Rebasing:** Rewrites history to create a linear commit timeline.

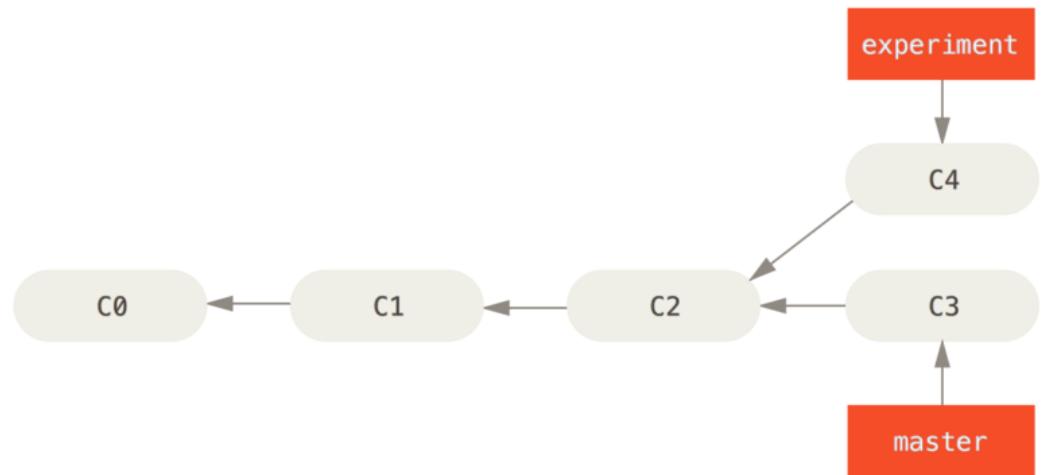
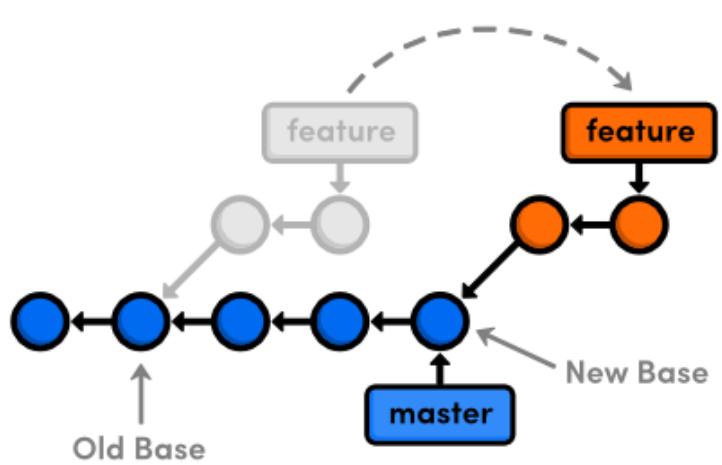
**Cherry-pick:** Applies specific commits from one branch to another.

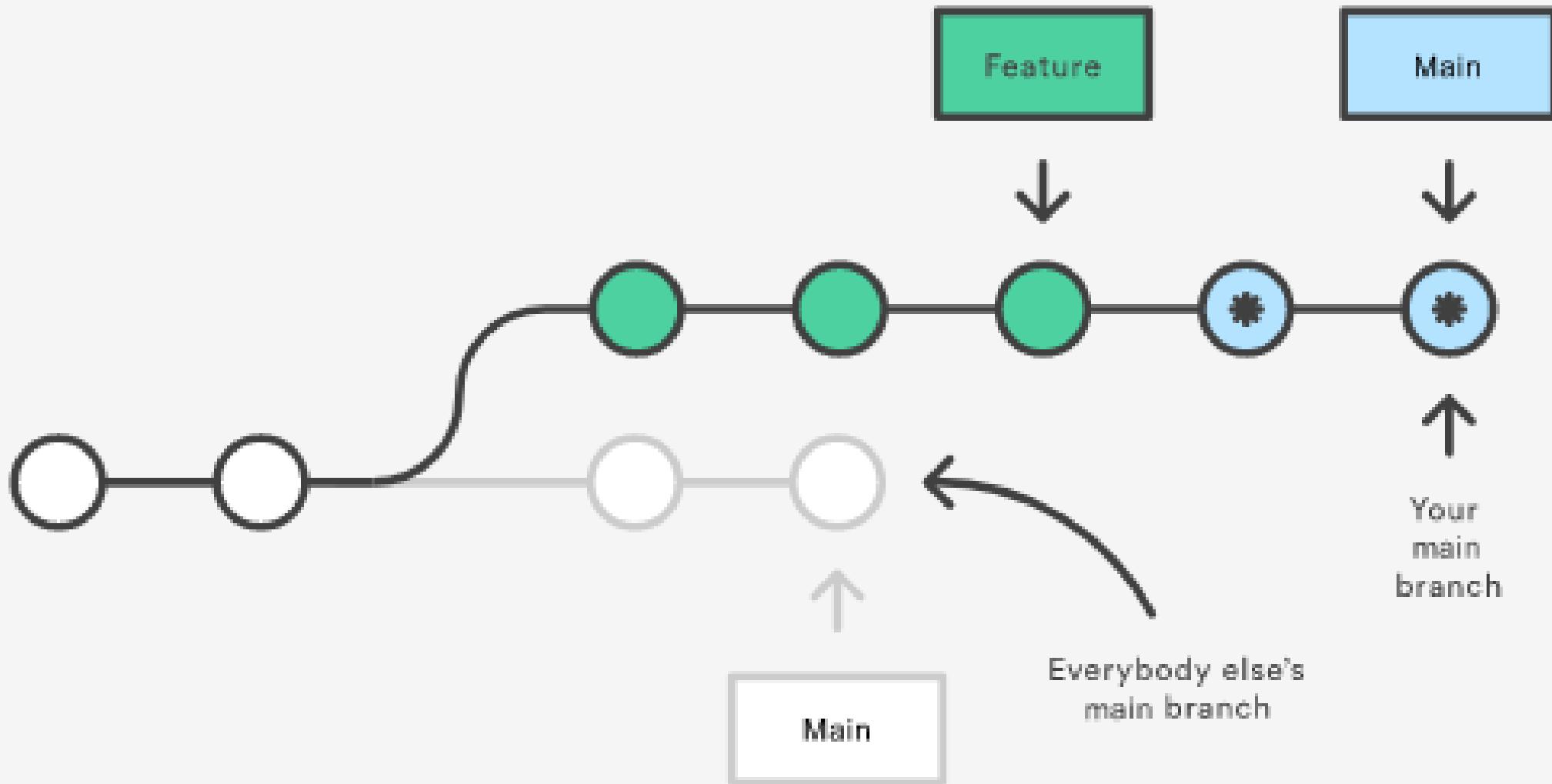
**Stashing:** Temporarily saves changes without committing.

**Hooks:** Custom scripts triggered by Git events (e.g., pre-commit).

**Submodules:** Includes one repository as a subdirectory of another.

# REBASE



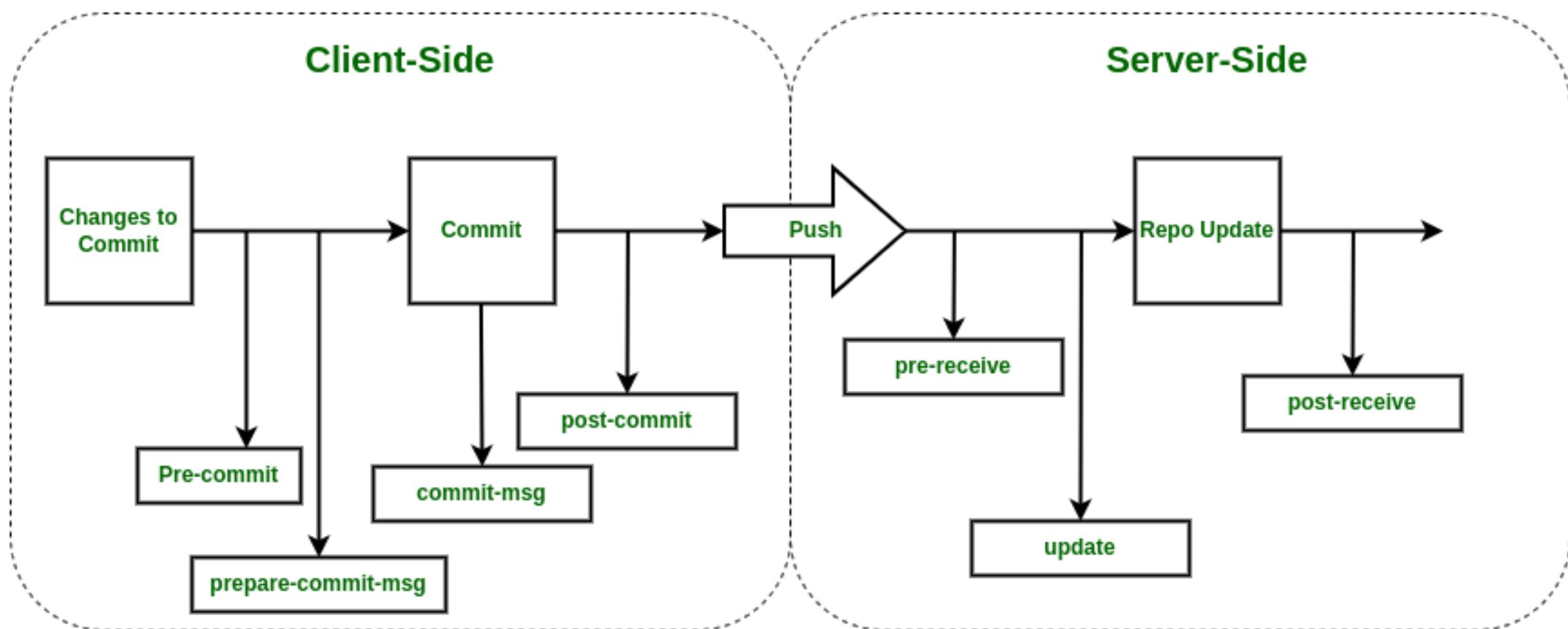


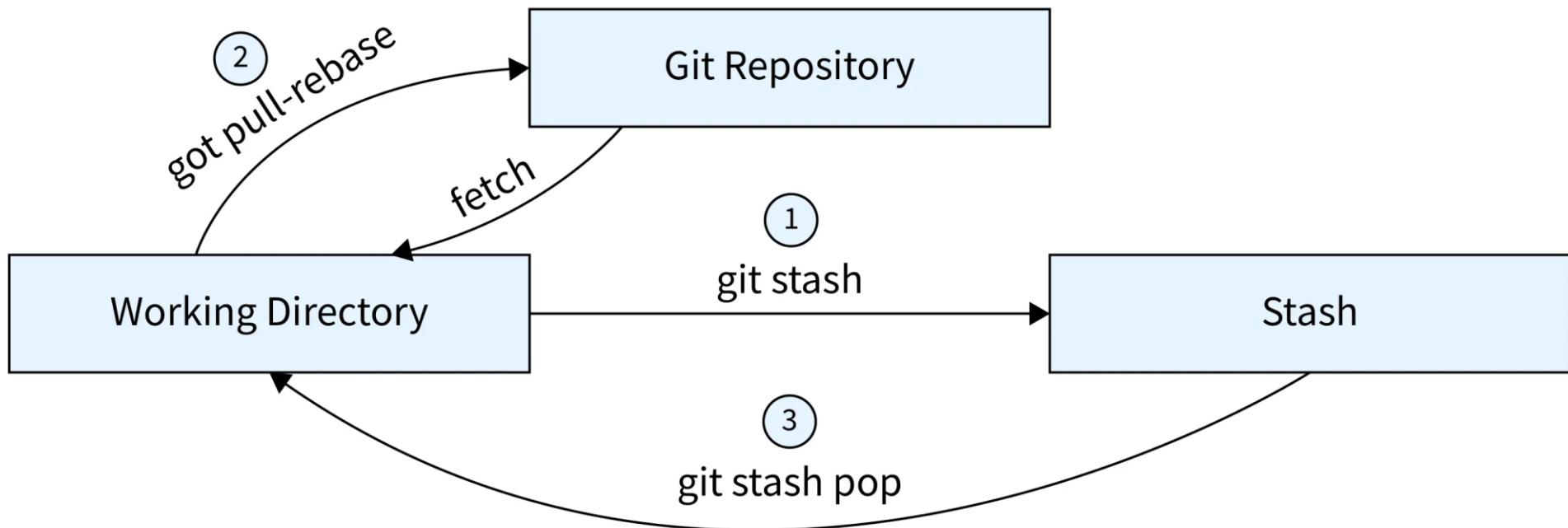
DEMO



# HOOKS

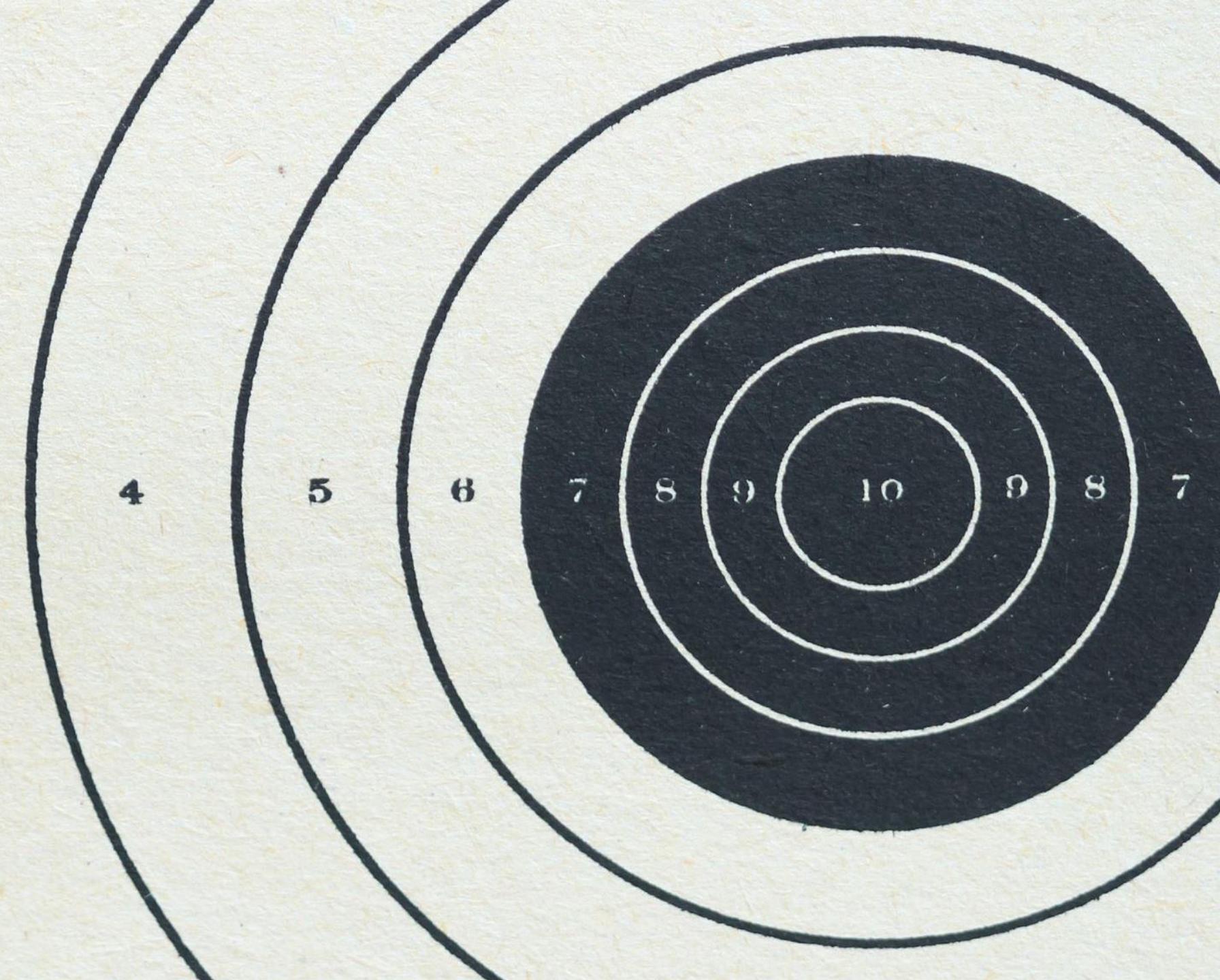


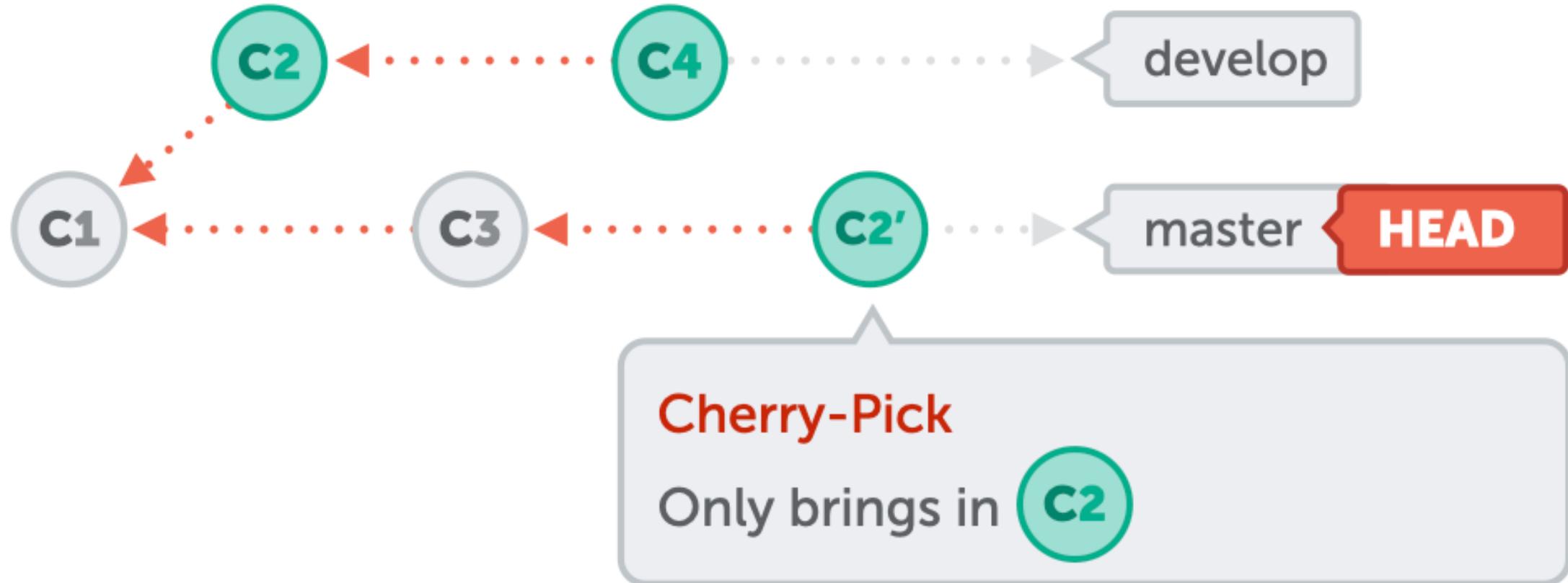




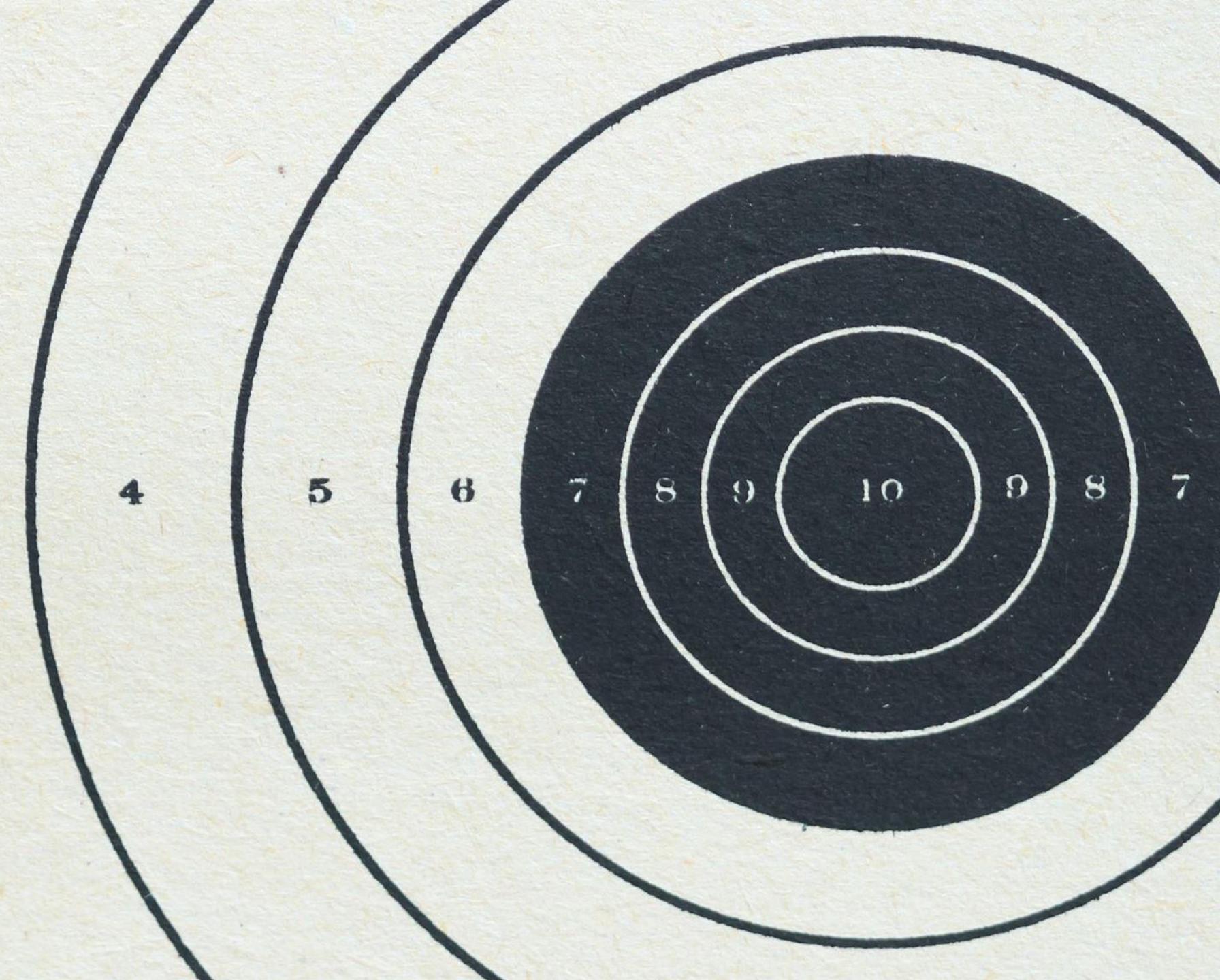
SCALER  
*Topics*

DEMO

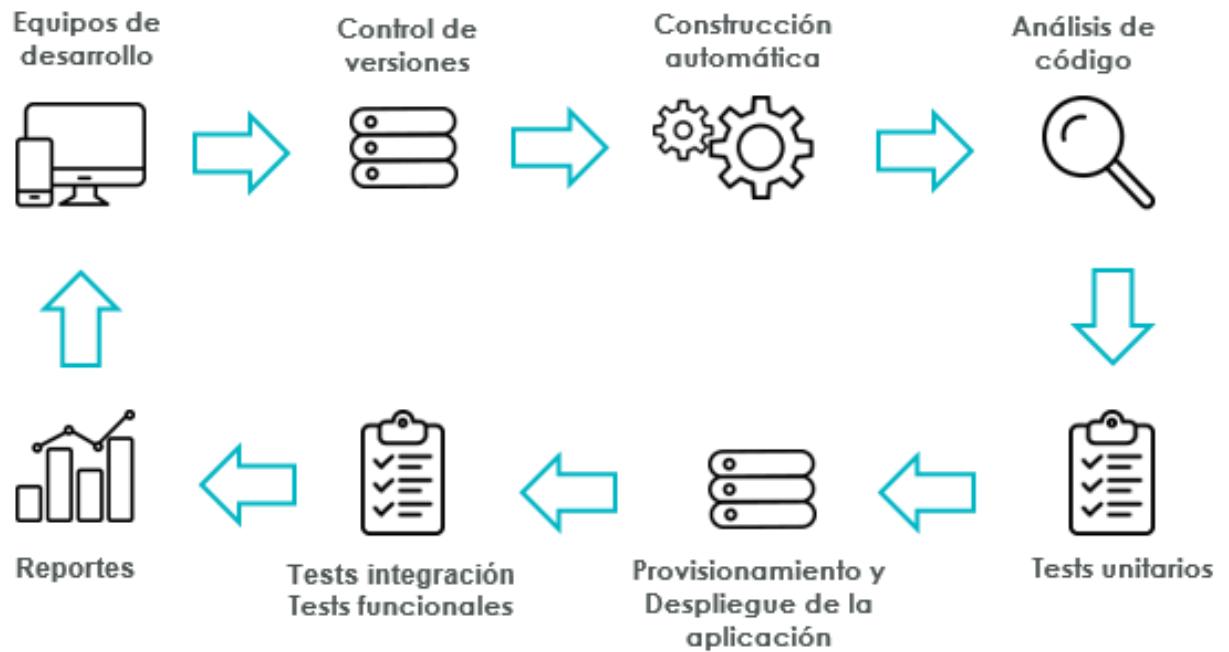


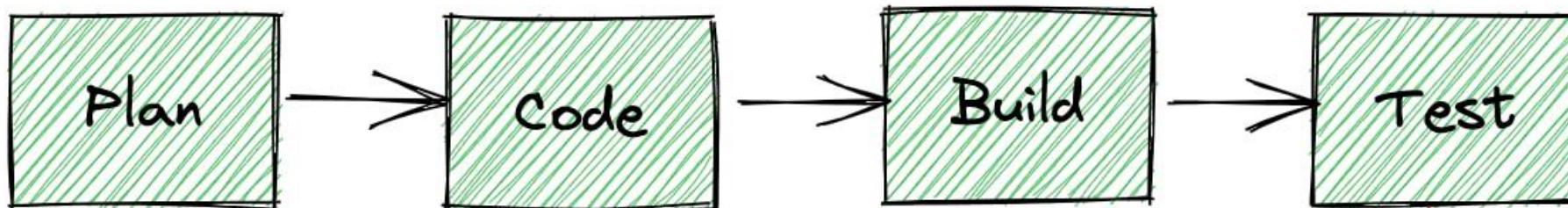


DEMO



# EXERCISE 2

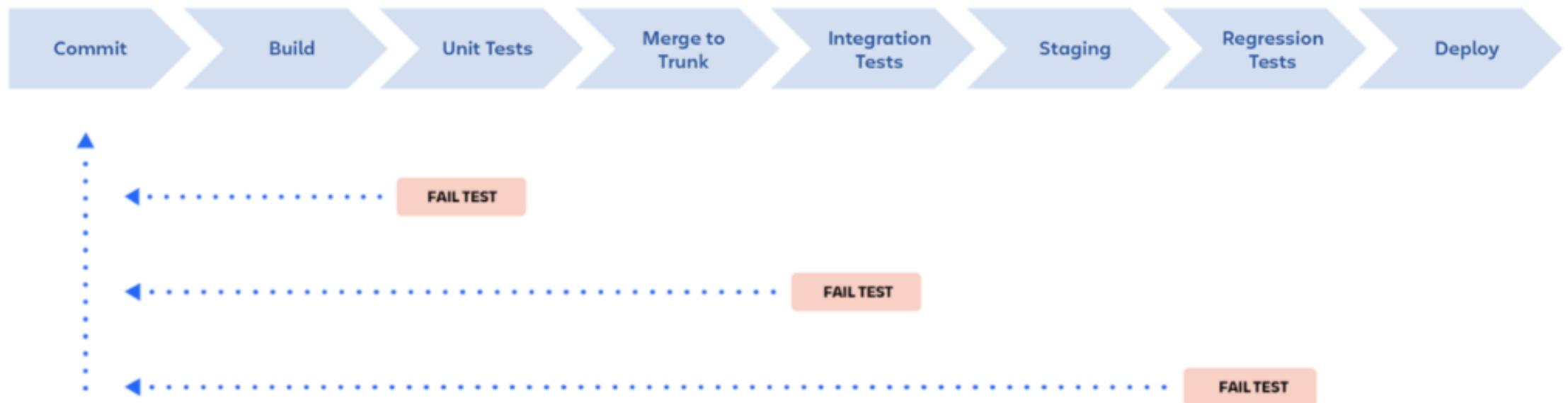


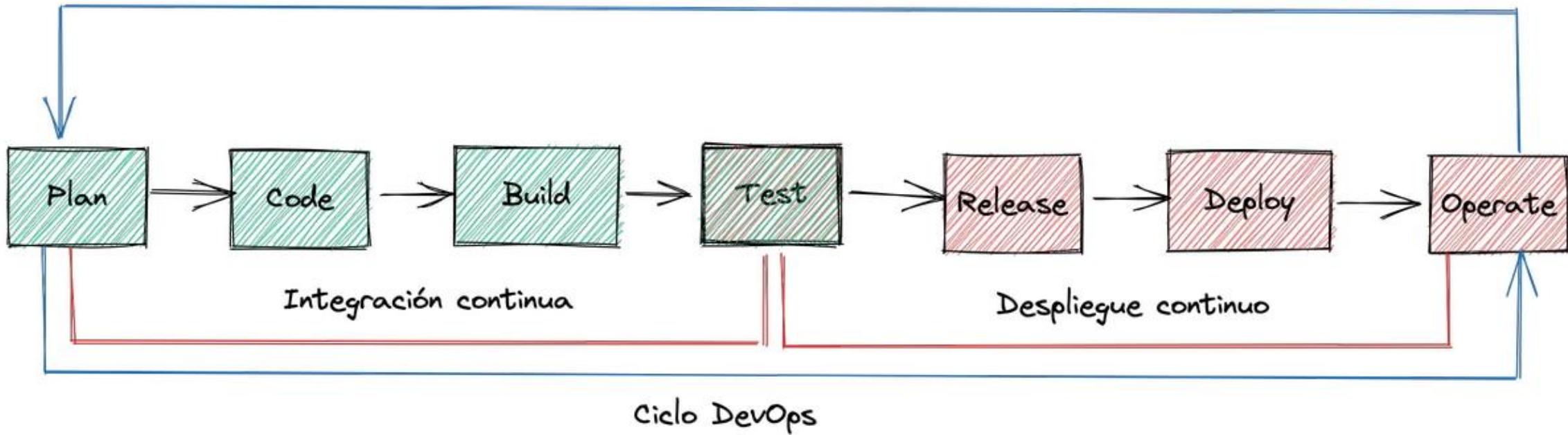


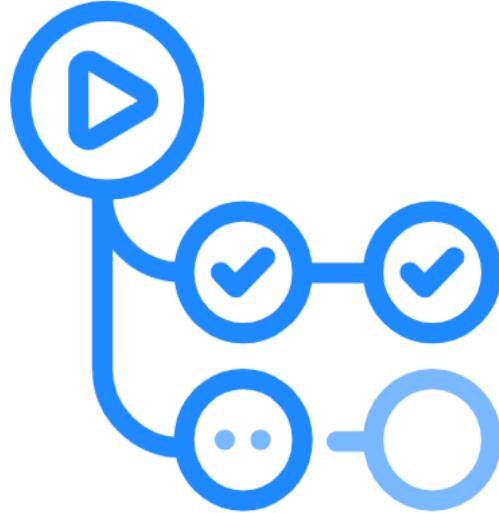
Integración continua



# PIPELINE







GitHub Actions  
Github actions

Event

Runner 1

Runner 2

Job 1

Step 1: Run action

Step 2: Run script

Step 3: Run script

Step 4: Run action

Job 2

Step 1: Run action

Step 2: Run script

Step 3: Run script

```
name: Nombre-del-Workflow

on:
  evento:
    - evento-trigger

jobs:
  nombre-del-trabajo:
    runs-on: tipo-de-maquina
    steps:
      - nombre-del-paso:
          run: comando-a-ejecutar
```

Workflows

New workflow

All workflows

How to manually trigger a work...

## How to manually trigger a workflow

workflow:"How to manually trigger a workflow"

0 results

Event ▾

Status ▾

Branch ▾

Actor ▾

This workflow has a `workflow_dispatch` event trigger.

Run workflow ▾

Use workflow from

Branch: main ▾

Your GitHub username ▾

Why are you running this workflow  
manually? ▾

I am running tests before implementing an aut...

Run workflow



No results matched your se

Try broadening your search filters

YAML

```
name: Greeting on variable day

on:
  workflow_dispatch

env:
  DAY_OF_WEEK: Monday

jobs:
  greeting_job:
    runs-on: ubuntu-latest
    env:
      Greeting: Hello
    steps:
      - name: "Say Hello Mona it's Monday"
        run: echo "$Greeting $First_Name. Today is $DAY_OF_WEEK!"
        env:
          First_Name: Mona
```

# Actions secrets and variables

New repository secret

Secrets and variables allow you to manage reusable configuration data. Secrets are **encrypted** and are used for sensitive data. [Learn more about encrypted secrets](#). Variables are shown as plain text and are used for **non-sensitive** data. [Learn more about variables](#).

Anyone with collaborator access to this repository can use these secrets and variables for actions. They are not passed to workflows that are triggered by a pull request from a fork.

Secrets

Variables

# Actions

An entirely new way to automate your development workflow.

18381 results filtered by

Actions



## Actions



### Setup Java JDK

By actions

Set up a specific version of the Java JDK and add the command-line tools to the PATH

1.1k stars



### Setup .NET Core SDK

By actions

Used to build and publish .NET source. Set up a specific version of the .NET and authentication to private NuGet repository

747 stars



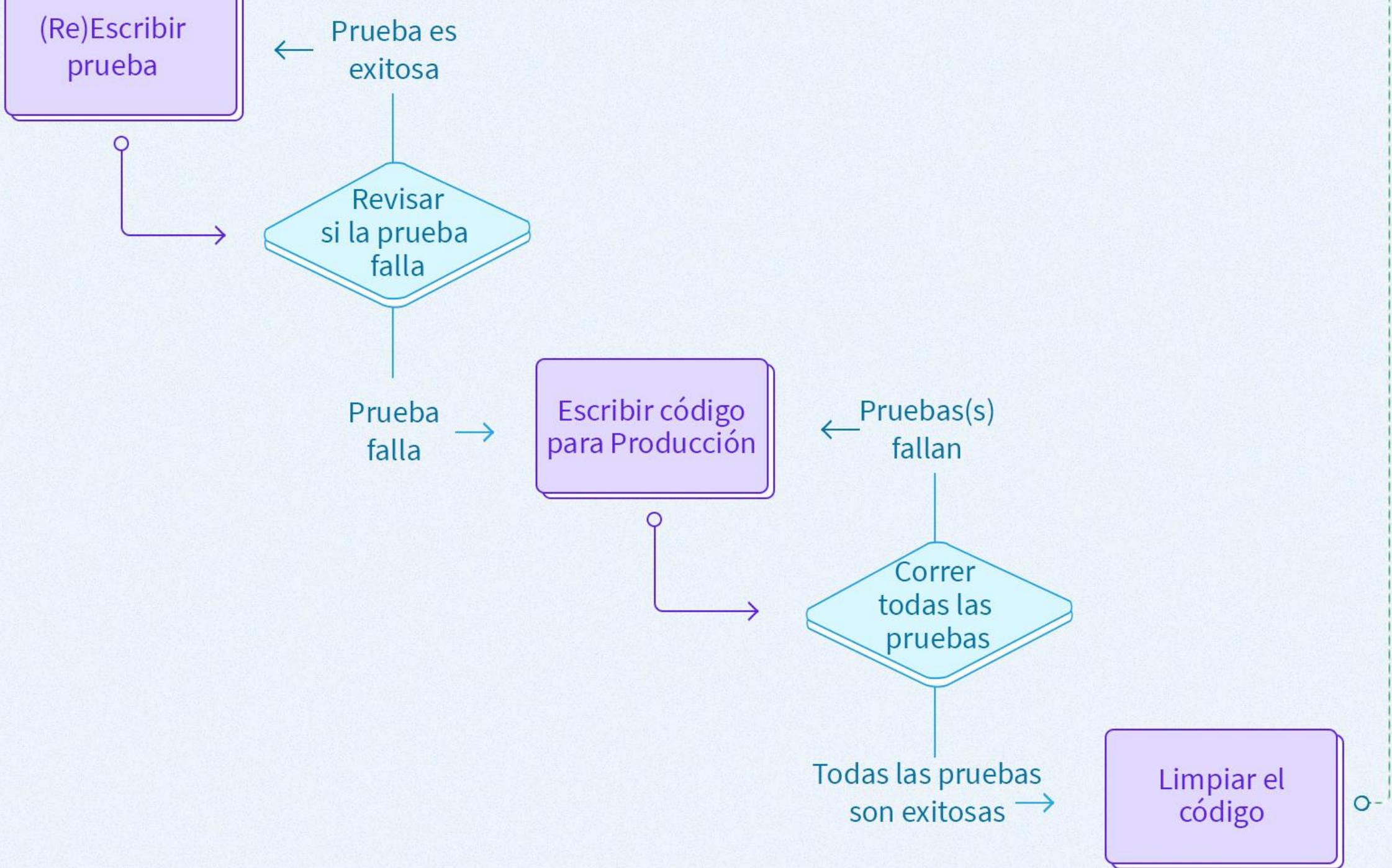


vs.

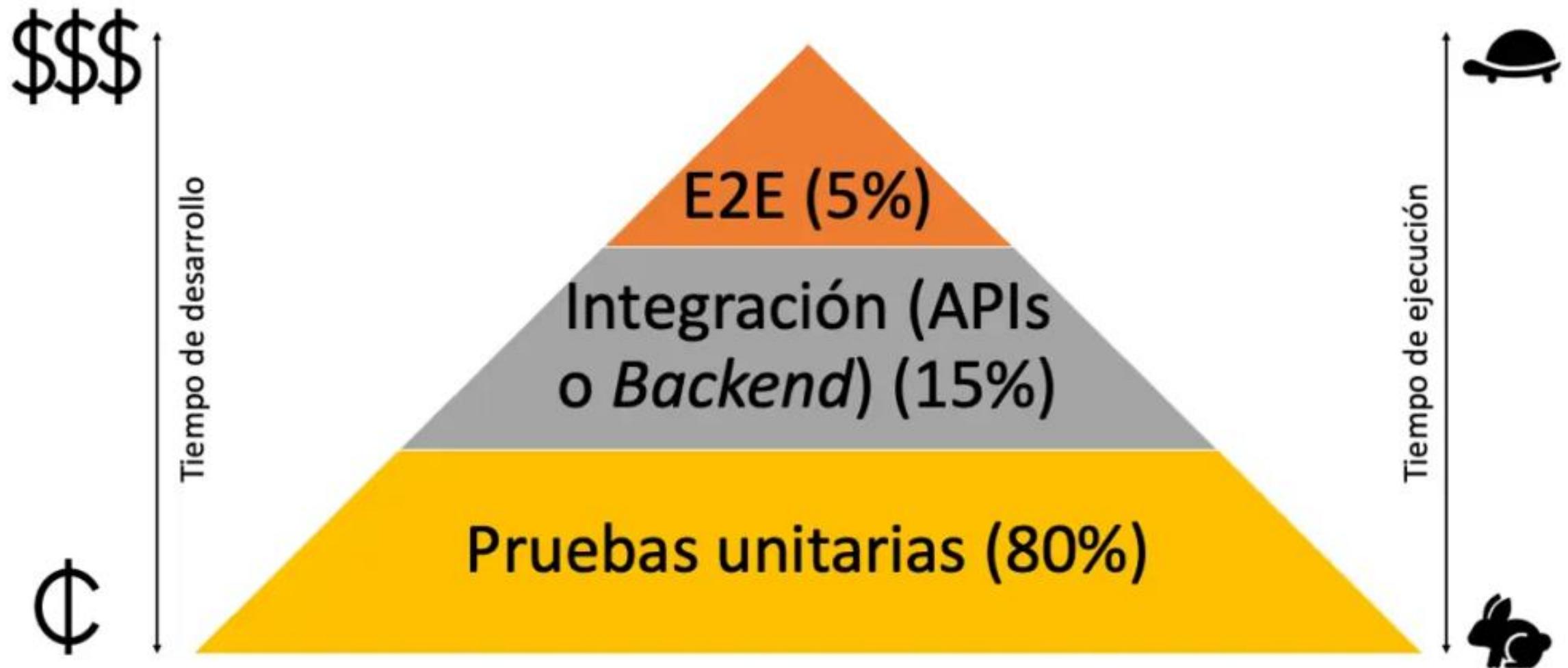


Pruebas  
dinámicas

Pruebas  
estáticas



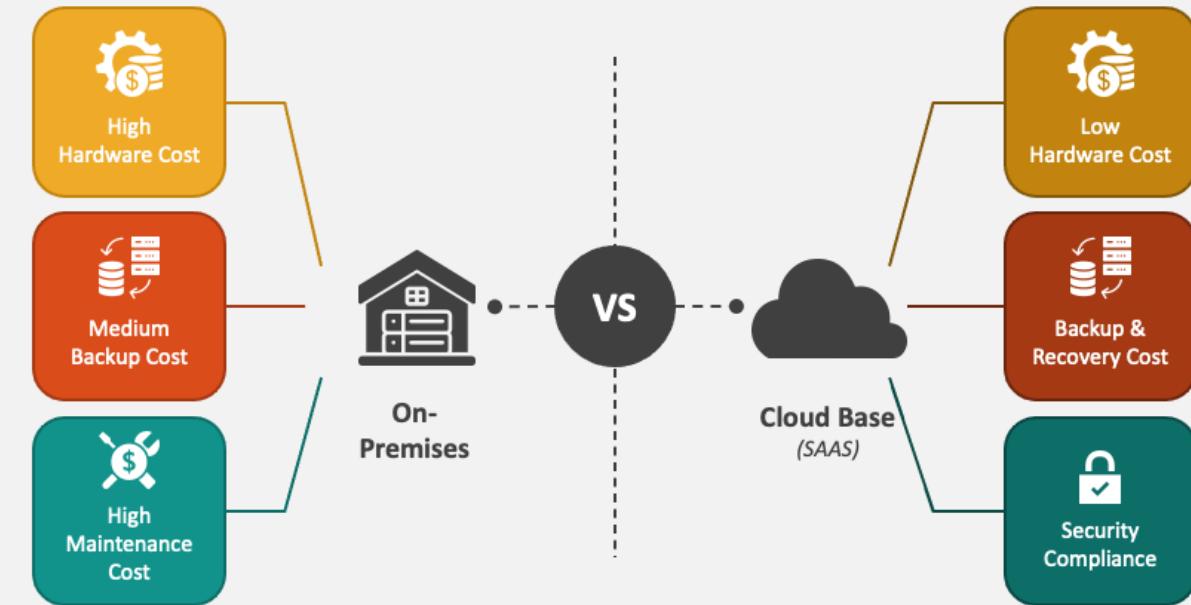




# EXERCISE 3

# CLOUD COMPUTING VS ON-PREMISE

## ON-PREMISE VS CLOUD



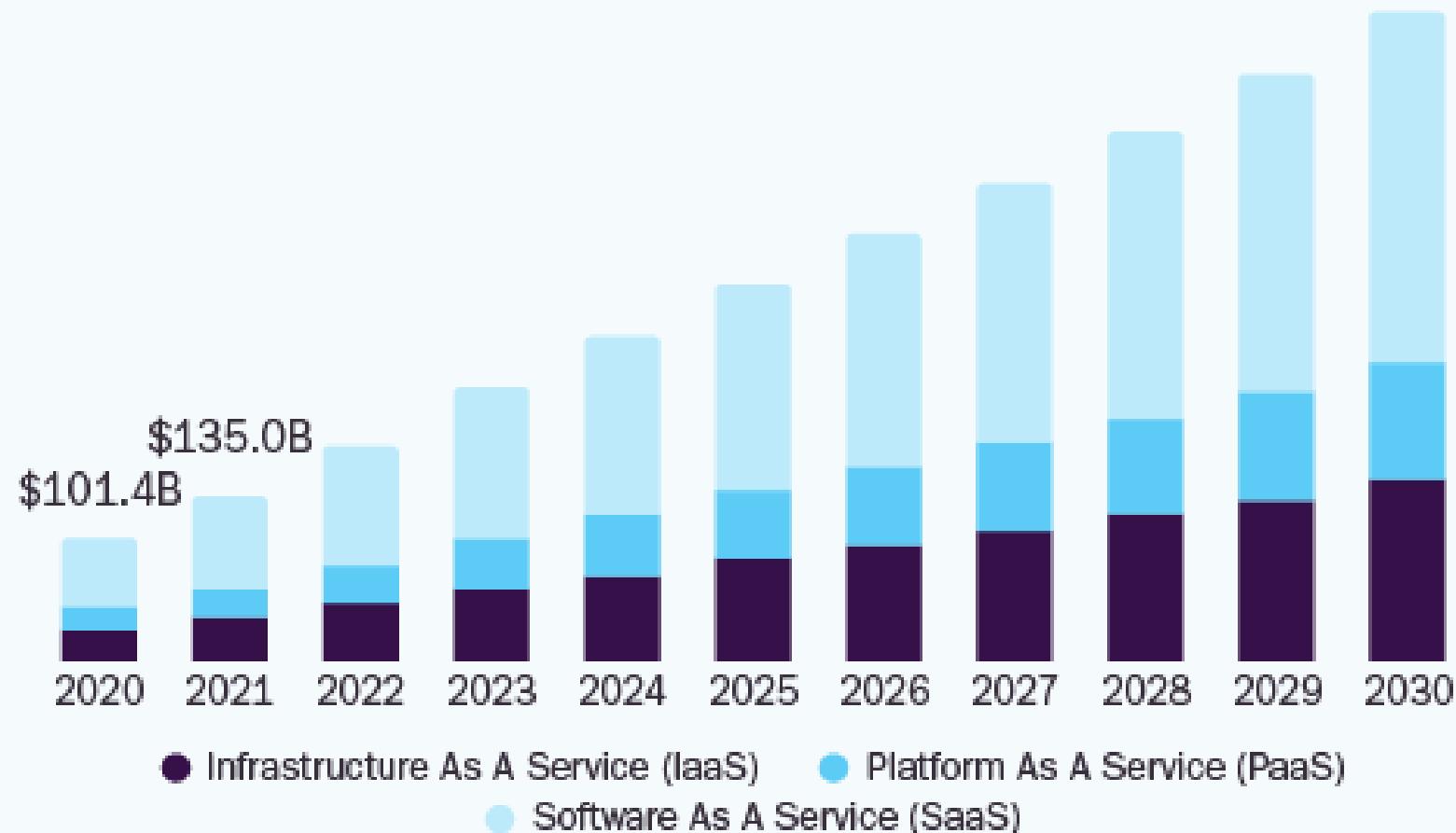


# U.S. Cloud Computing Market

size, by service, 2020 - 2030 (USD Billion)



GRAND VIEW RESEARCH



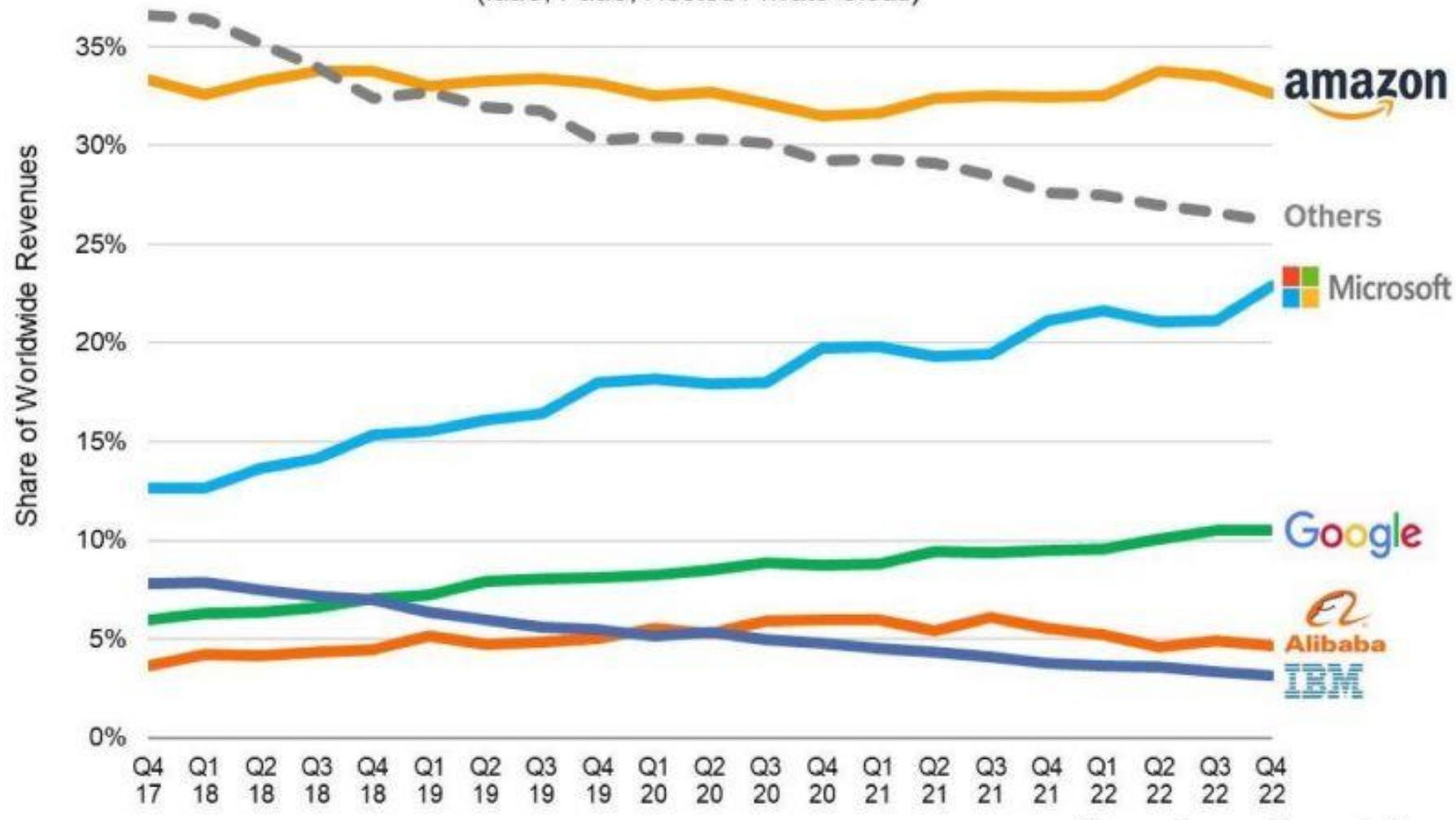
**13.1%**

U.S. Market CAGR,  
2023 - 2030

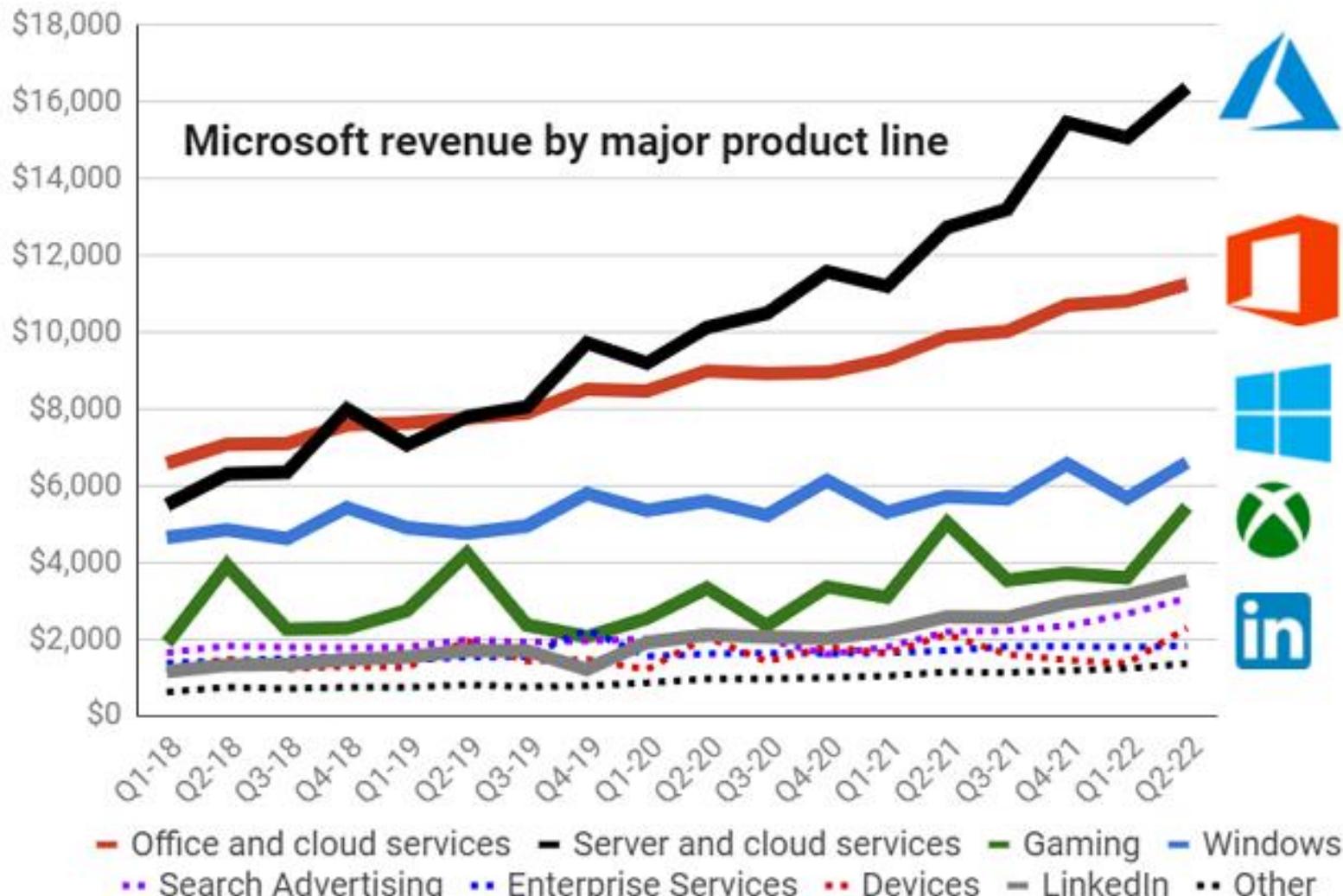
Source:  
[www.grandviewresearch.com](http://www.grandviewresearch.com)

## Cloud Provider Market Share Trend

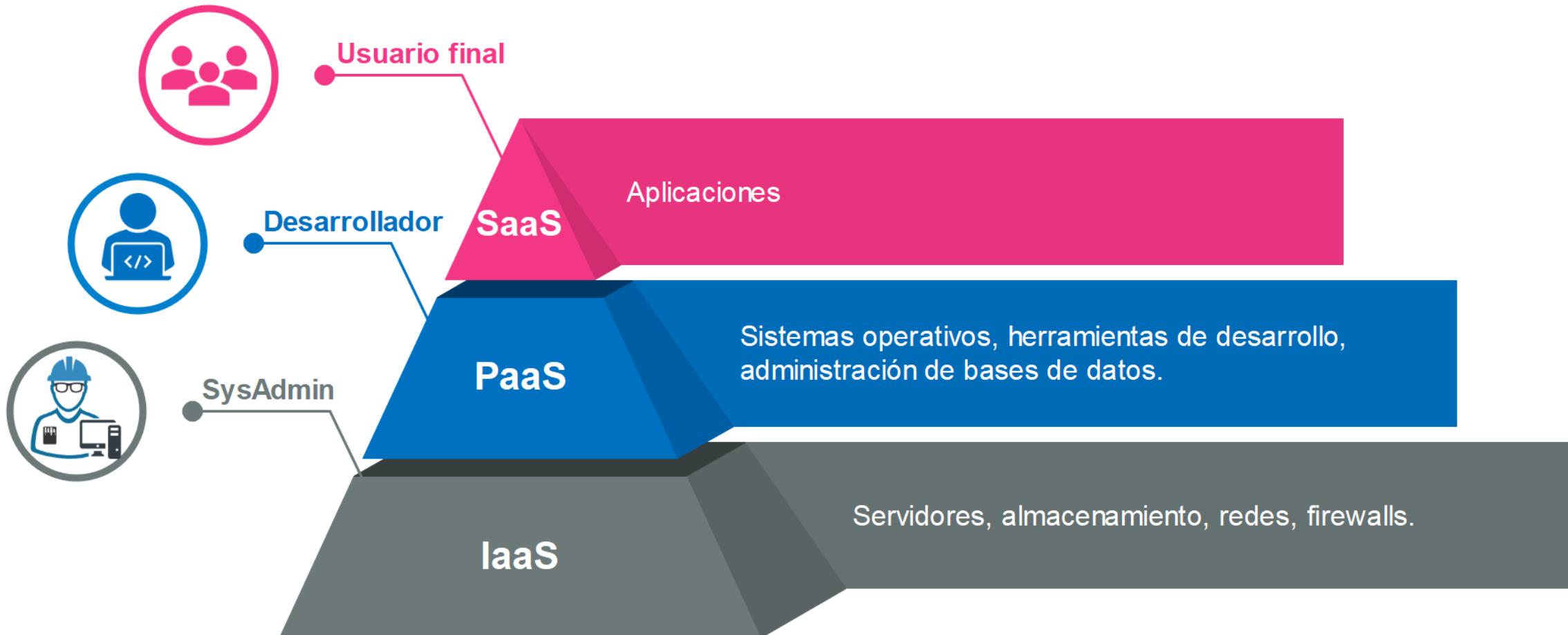
(IaaS, PaaS, Hosted Private Cloud)

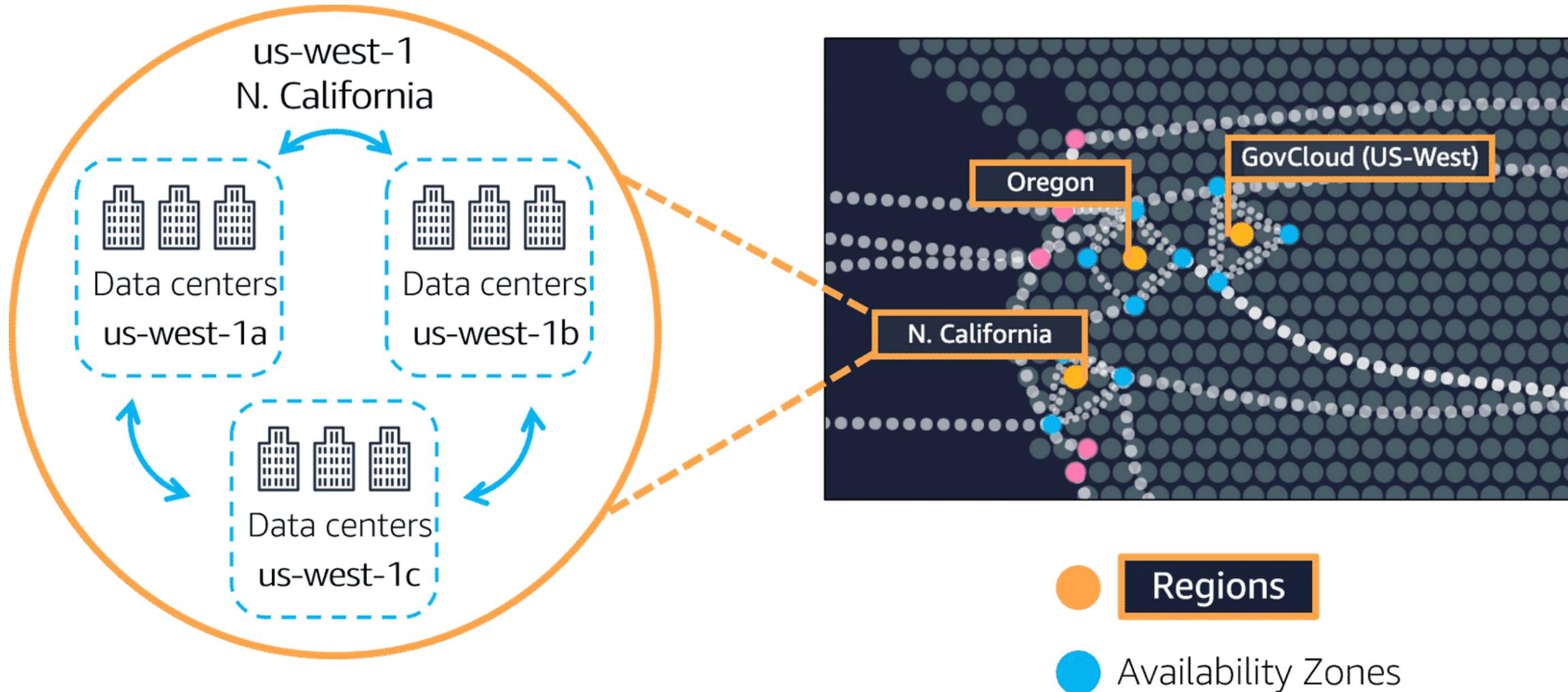


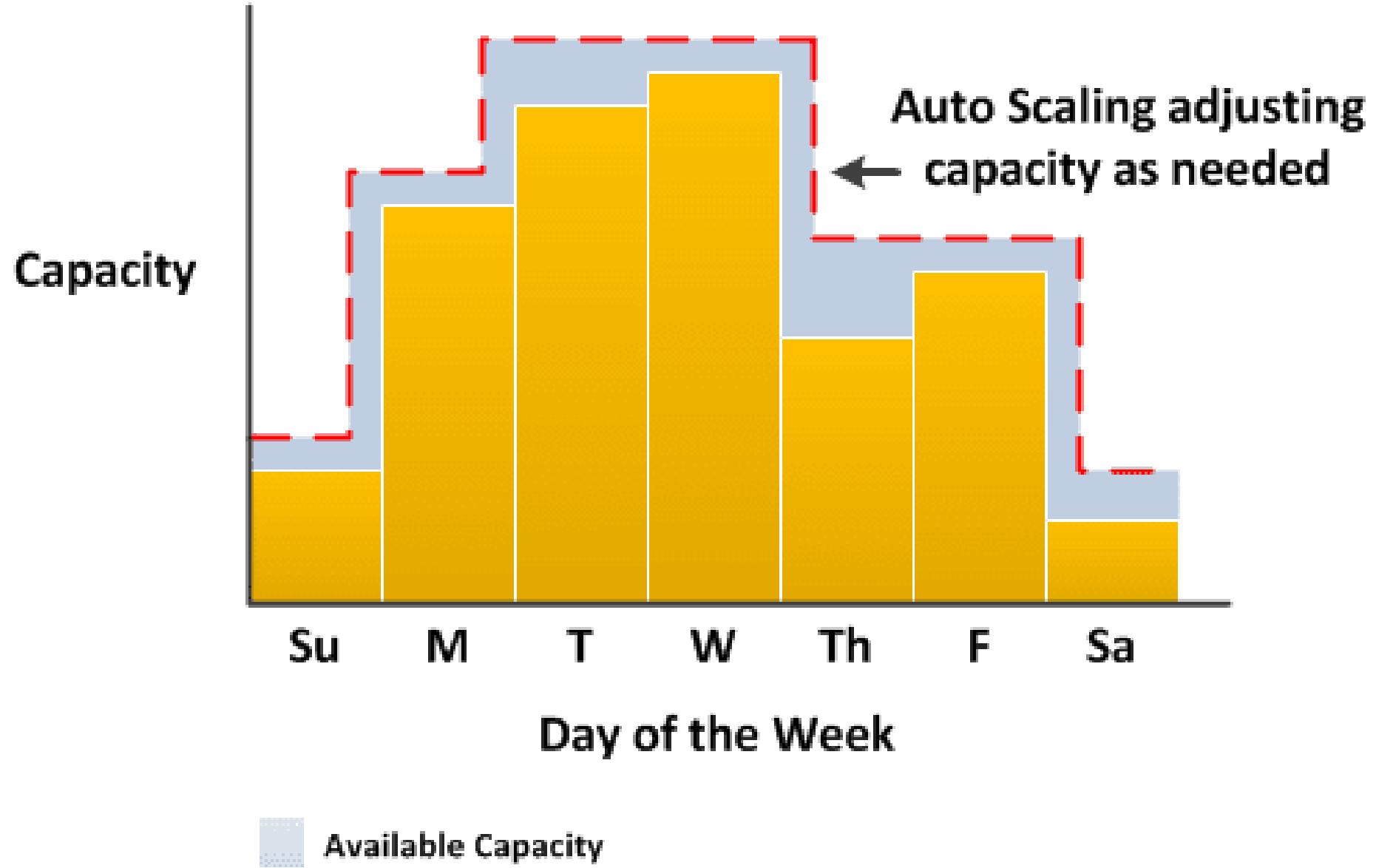
Source: Synergy Research Group



Source: Microsoft 10K and 10Q filings, in millions per fiscal quarter GeekWire

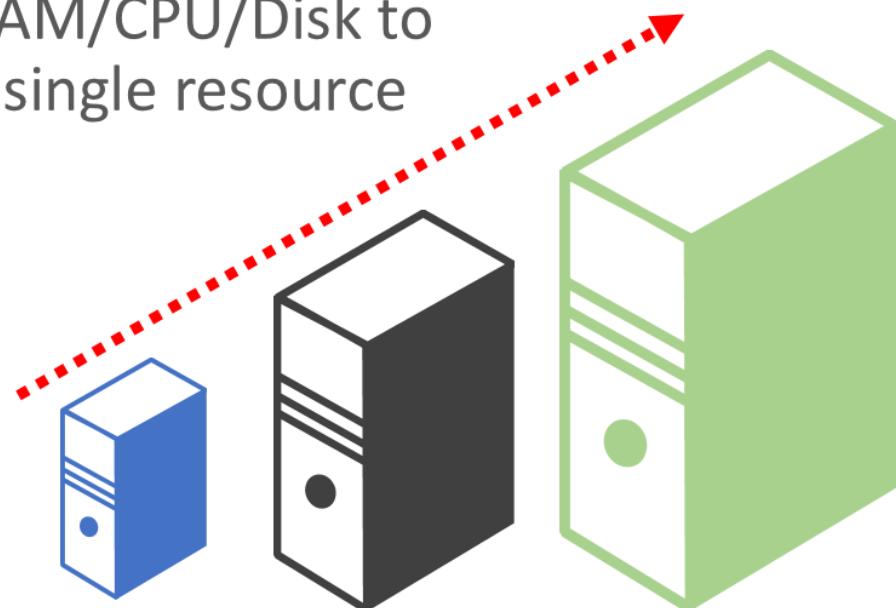






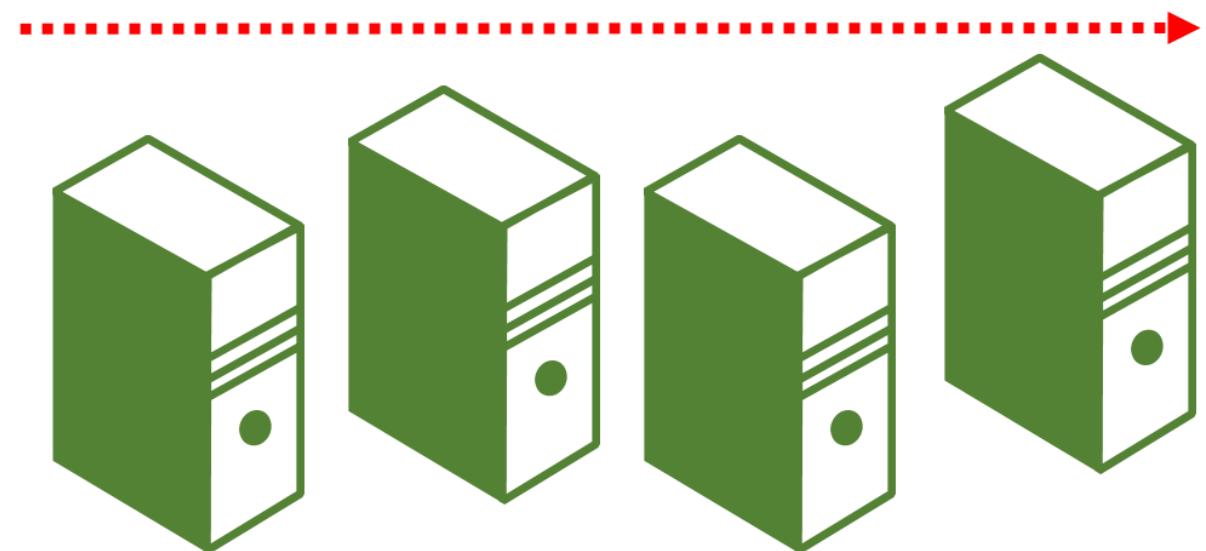
## Scale Up (vertical scaling)

Increase capacity by adding RAM/CPU/Disk to a single resource



## Scale Out (horizontal scaling)

Increase capacity by adding resources





Microsoft

chat.openai.com

ChatGPT





Services ▾

Search results for 'ec2'



# AWS N

AWS services

▼ Recently



▼ All services



Services (6)

Features (34)

Documentation (149,513)

Marketplace (967)

## Services

[See all 6 results ▶](#)



EC2

Virtual Servers in the Cloud



EC2 Image Builder

A managed service to automate build, customize and deploy OS images



AWS Compute Optimizer

Recommend optimal AWS Compute resources for your workloads



AWS Firewall Manager

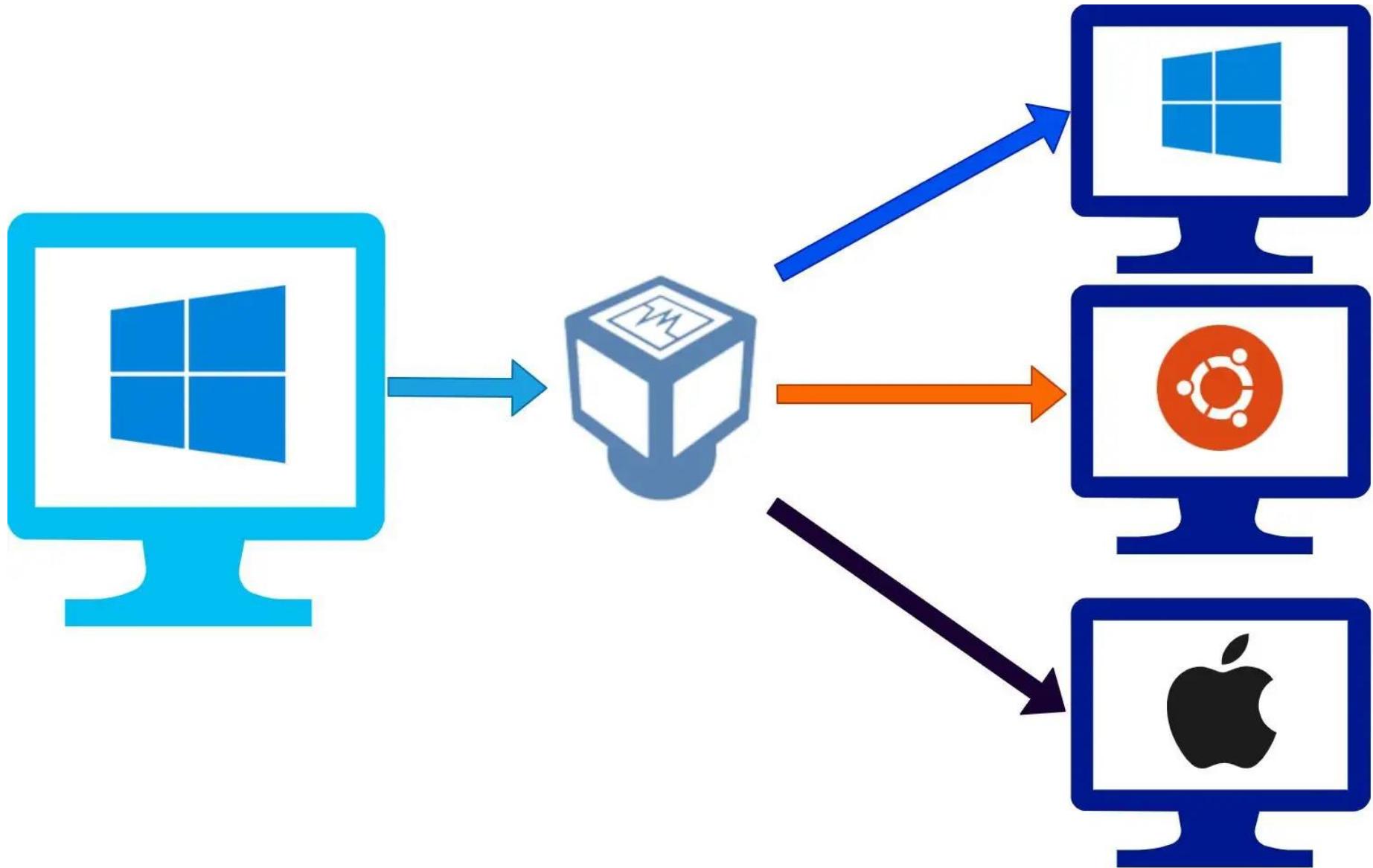
Central management of firewall rules



James (aws)

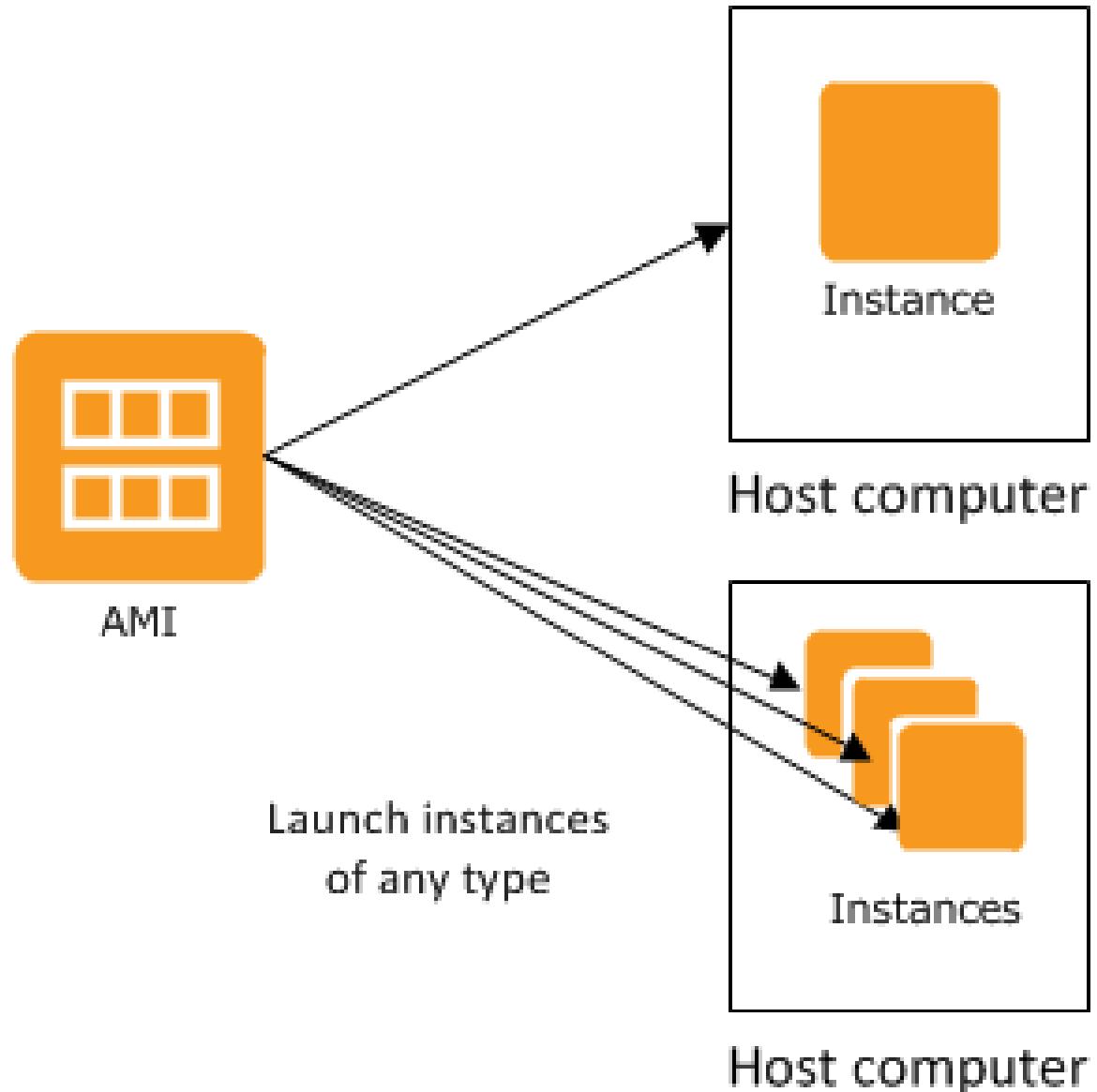
⌘2

```
~ $ aws dynamodb wizard new-table
Enter the name of the table: MyTable
Enter the name of the primary key: pk
Primary key type
Add a sort key?
Select the read/write capacity mode.
Select Server-side encryption settings for your DynamoDB table
  DEFAULT
  KMS - AWS managed CMK
> KMS - Customer managed CMK
```

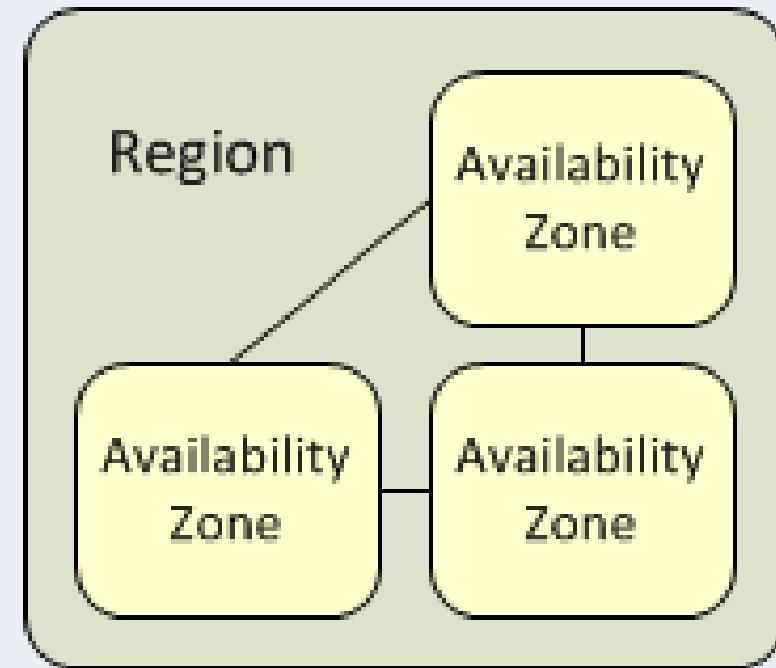
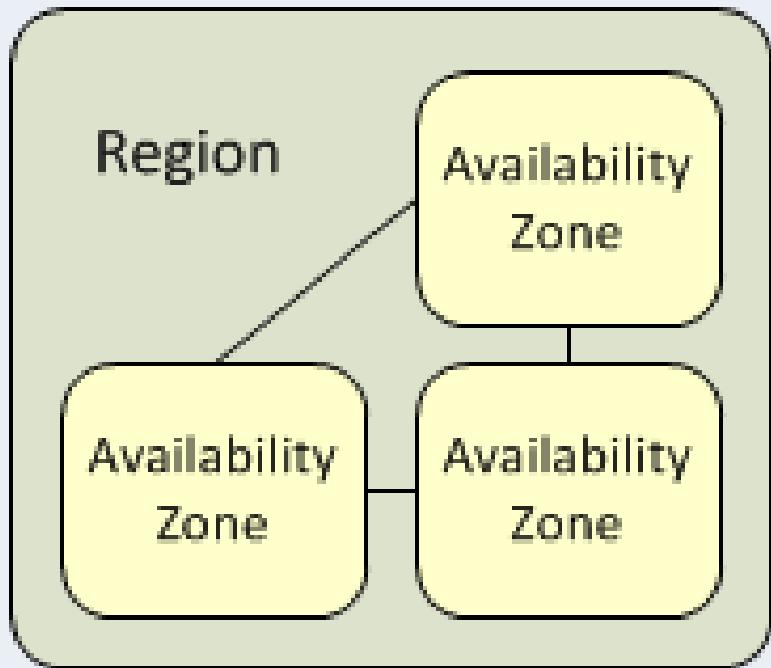


EC2





# Amazon Web Services

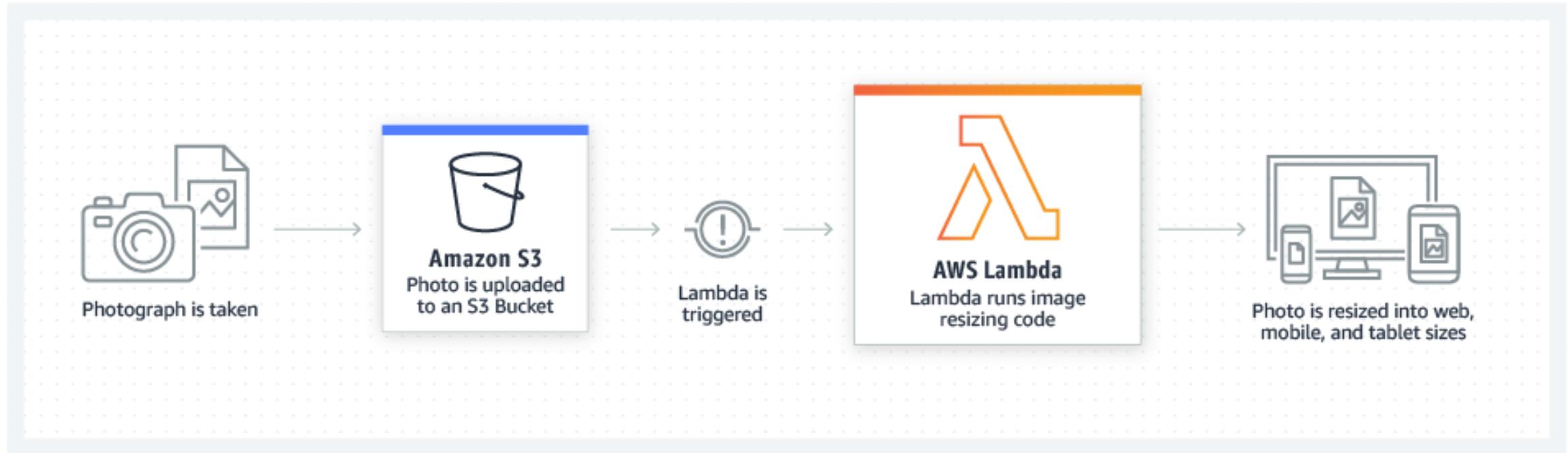




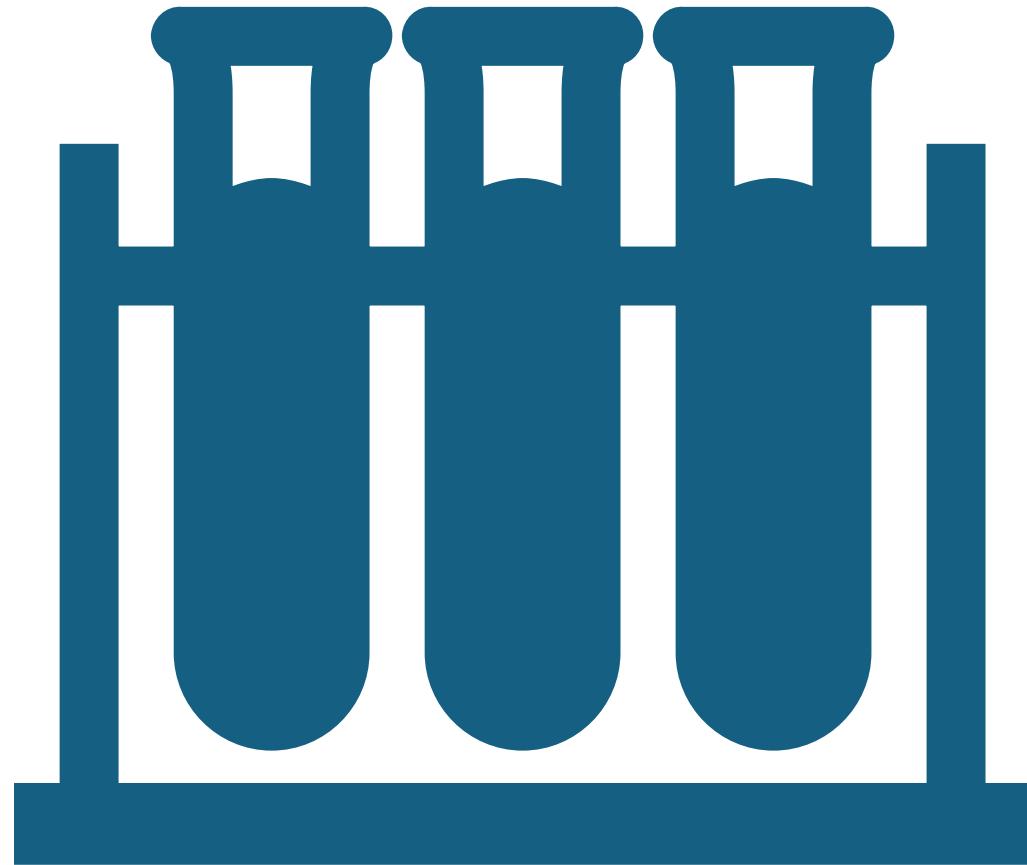
# AWS LAMBDA

- Automatic provisioning and scaling: Lambda automatically handles the infrastructure needed to run your code, scaling up or down depending on the number of events or requests.
- Code monitoring and logging: It manages the monitoring of your code execution and logs activities for you, integrating with AWS CloudWatch.
- Pay only for what you use: You are not charged when your code is not running. Billing is based solely on the actual compute time your code consumes, measured in milliseconds.



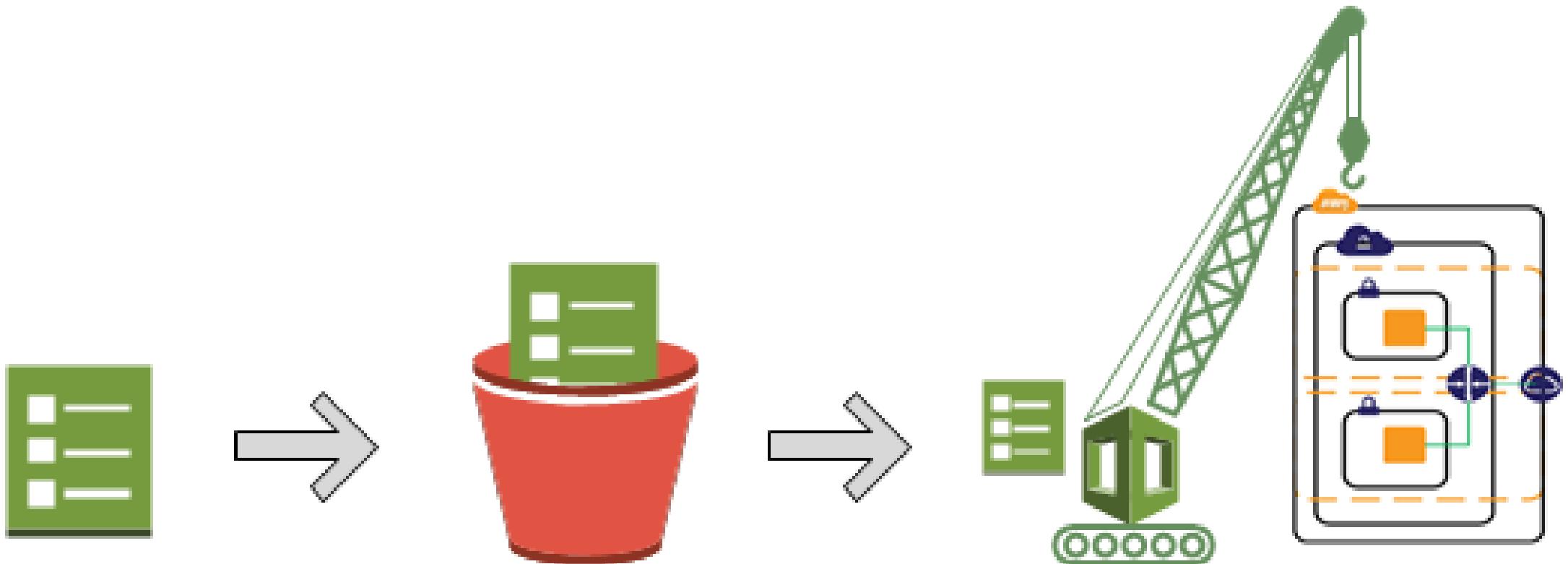


## EXERCISE 4





# CLOUDFORMATION



1 Create or use an existing template

2 Save locally or in S3 bucket

3 Use AWS CloudFormation to create a stack based on your template. It constructs and configures your stack resources.

File 'newtemplate'

Resource types

- ▶ ApiGateway
- ▶ ApplicationAutoScaling
- ▶ Athena
- ▶ AutoScaling
- ▶ Batch
- ▶ CertificateManager
- ▶ Cloud9
- ▶ CloudFormation
- ▶ CloudFront
- ▶ CloudTrail
- ▶ CloudWatch
- ▶ CodeCommit
- ▶ CodeDeploy
- ▶ CodePipeline
- ▶ Cognito
- ▶ Config
- ▶ DMS
- ▶ DataPipeline

To start building your template, drag resources from the Resources pane.

new.template 

Choose template language: 

```
1 - {  
2   "AWSTemplateFormatVersion": "2010-09-09"  
3 }
```

### Example in JSON

```
"Parameters" : {  
    "InstanceTypeParameter" : {  
        "Type" : "String",  
        "Default" : "t2.micro",  
        "AllowedValues" : ["t2.micro", "m1.small", "m1.large"],  
        "Description" : "Enter t2.micro, m1.small, or m1.large. Default is  
        t2.micro."  
    }  
}
```

InstanceTypeParameter has a  
default value of t2.micro

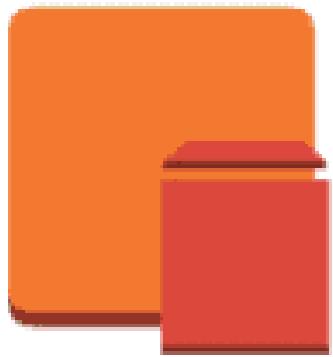
# EXERCISE 5





Amazon EFS

File

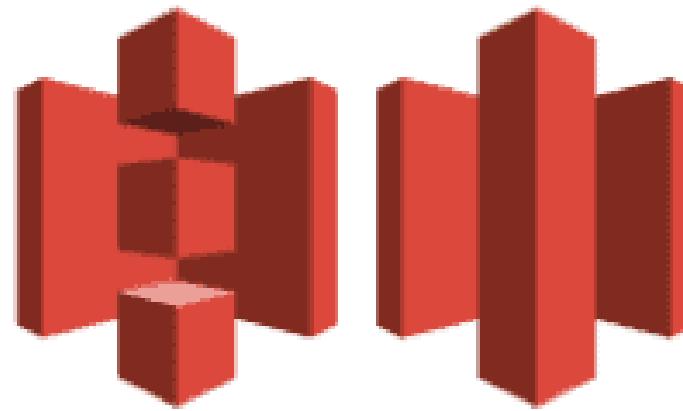


Amazon EBS

Block



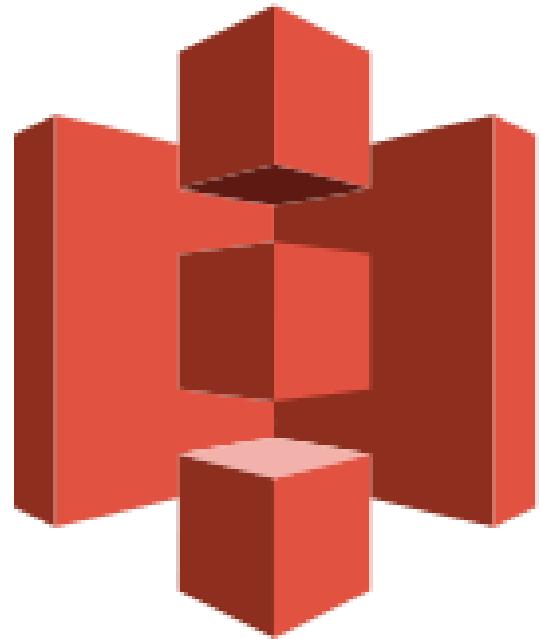
Amazon EC2  
Instance Store



Amazon S3

Object

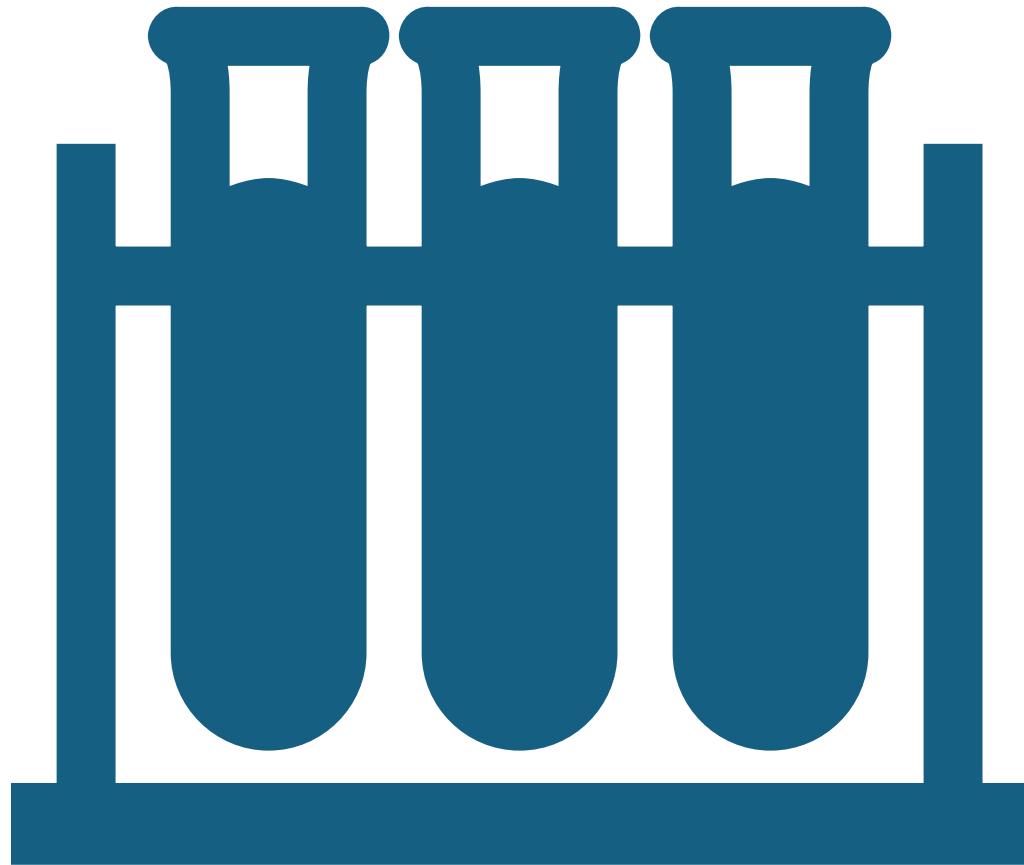
Amazon Glacier



amazon  
S3

					
S3 Standard	S3 Intelligent-Tiering	S3 Standard-IA	S3 One Zone-IA	S3 Glacier	S3 Glacier Deep Archive
<i><b>Access frequency</b></i>					<i><b>Archive</b></i>
<p><b>Frequent</b></p> <ul style="list-style-type: none"> <li>• Active, frequently accessed data</li> <li>• Milliseconds access</li> <li>• <math>\geq 3</math> AZ</li> <li>• \$0.0210/GB</li> </ul>	<ul style="list-style-type: none"> <li>• Data with changing access patterns</li> <li>• Milliseconds access</li> <li>• <math>\geq 3</math> AZ</li> <li>• \$0.0210 to \$0.0125/GB</li> <li>• Monitoring fee per object</li> <li>• Min storage duration</li> </ul>	<ul style="list-style-type: none"> <li>• Infrequently accessed data</li> <li>• Milliseconds access</li> <li>• <math>\geq 3</math> AZ</li> <li>• \$0.0125/GB</li> <li>• Retrieval fee per GB</li> <li>• Min storage duration</li> <li>• Min object size</li> </ul>	<ul style="list-style-type: none"> <li>• Re-creatable, less accessed data</li> <li>• Milliseconds access</li> <li>• 1 AZ</li> <li>• \$0.0100/GB</li> <li>• Retrieval fee per GB</li> <li>• Min storage duration</li> <li>• Min object size</li> </ul>	<ul style="list-style-type: none"> <li>• Archive data</li> <li>• Select minutes or hours</li> <li>• <math>\geq 3</math> AZ</li> <li>• \$0.0040/GB</li> <li>• Retrieval fee per GB</li> <li>• Min storage duration</li> </ul>	<ul style="list-style-type: none"> <li>• Long-term archive-data</li> <li>• Select hours</li> <li>• <math>\geq 3</math> AZ</li> <li>• \$0.00099/GB</li> <li>• Retrieval fee per GB</li> <li>• Min storage duration</li> </ul>

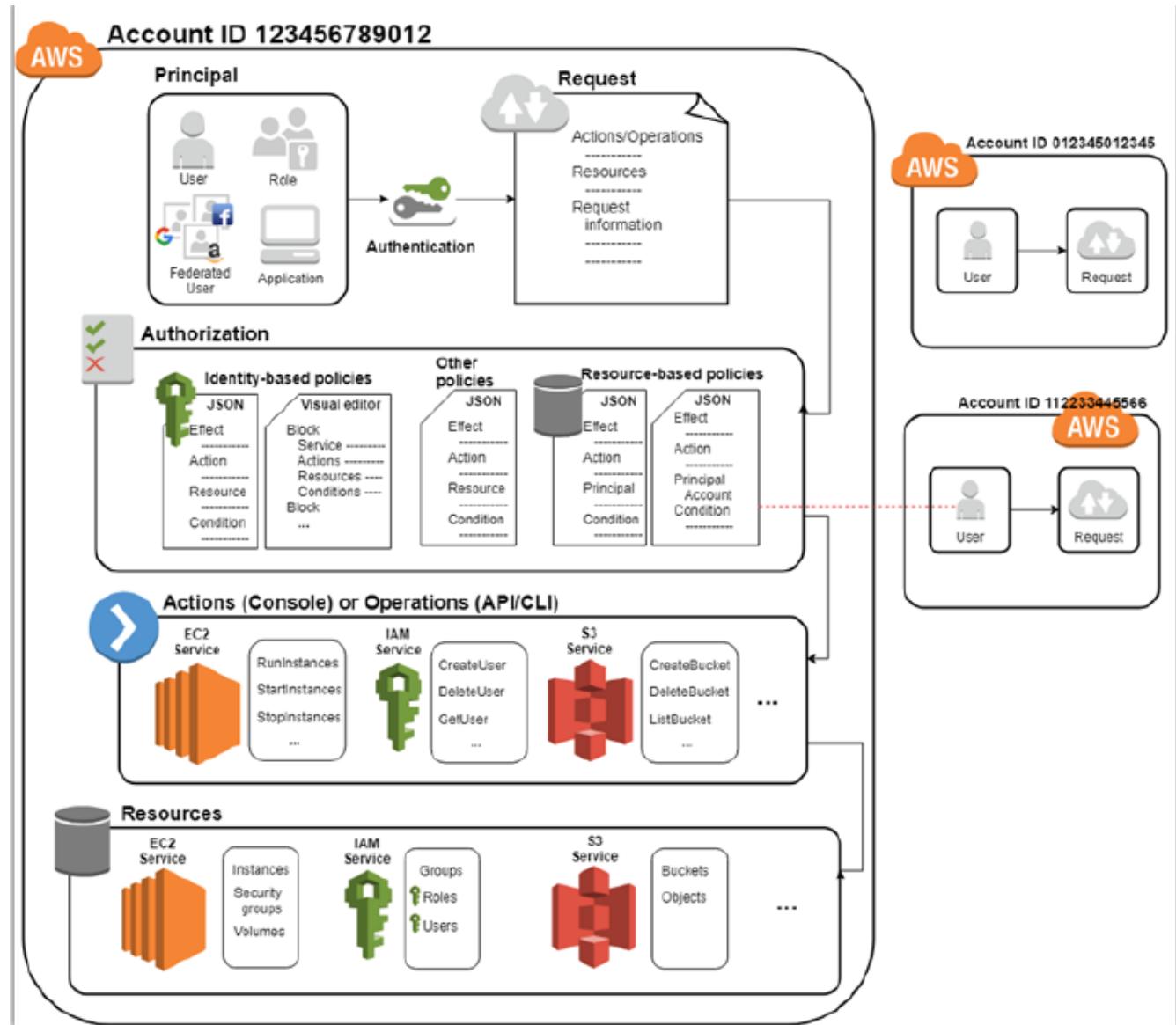
## EXERCISE 6



# IAM

- **Users:** A user is a person who needs access to your AWS services. You can grant permissions to a user to control what they can and cannot access.
- **Groups:** A group is a collection of users who share the same permissions. This is often used to simplify management.
- **Roles:** A role is used to define a set of permissions and who can use them. However, it is not directly assigned to a person or service. Instead, a service or person can assume the role when needed. Roles are temporary and therefore provide greater security compared to granting permanent permissions through groups or users.





# Generated policy

1

2

3

## Customize permissions

Review the following policy template. You must specify resources for actions that support resource-level permissions to continue creating the policy.

```
1  [{}  
2      "Version": "2012-10-17",  
3      "Statement": [  
4          {  
5              "Effect": "Allow",  
6              "Action": [  
7                  "access-analyzer:ValidatePolicy",  
8                  "iam:GetAccountPasswordPolicy",  
9                  "iam:GetAccountSummary",  
10                 "iam>ListAccountAliases",  
11                 "iam>ListGroups",  
12                 "iam>ListPolicies",  
13                 "iam>ListRoles",  
14                 "iam>ListUsers"  
15             ],  
16             "Resource": "*"  
17         },  
18         {  
19             "Effect": "Allow",  
20             "Action": [  
21                 "iam:GetRole",  
22                 "iam>ListAttachedRolePolicies",  
23                 "iam>ListInstanceProfilesForRole",  
24                 "iam>ListRolePolicies",  
25                 "iam>ListRoleTags"  
26             ],  
27             "Resource": "arn:aws:iam:${Account}:role/${RoleNameWithPath}"  
28         },  
29         {  
30             "Effect": "Allow",  
31             "Action": [  
32                 "iam GetUser",  
33                 "iam>ListAccessKeys",  
34                 "iam>ListAttachedUserPolicies",  
35                 "iam>ListGroupsForUser",  
36                 "iam>ListUserTags"  
37             ],  
38             "Resource": "arn:aws:iam:${Account}:user/${UserNameWithPath}"  
39         }]
```

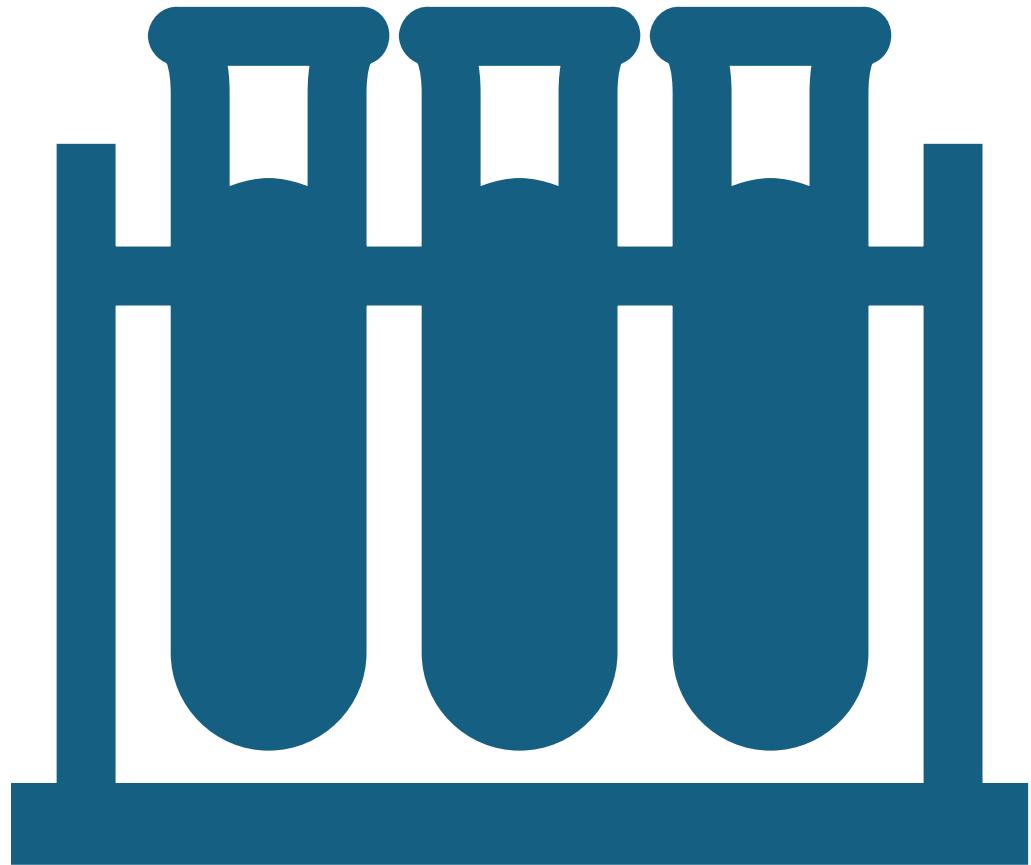
### Edit statement

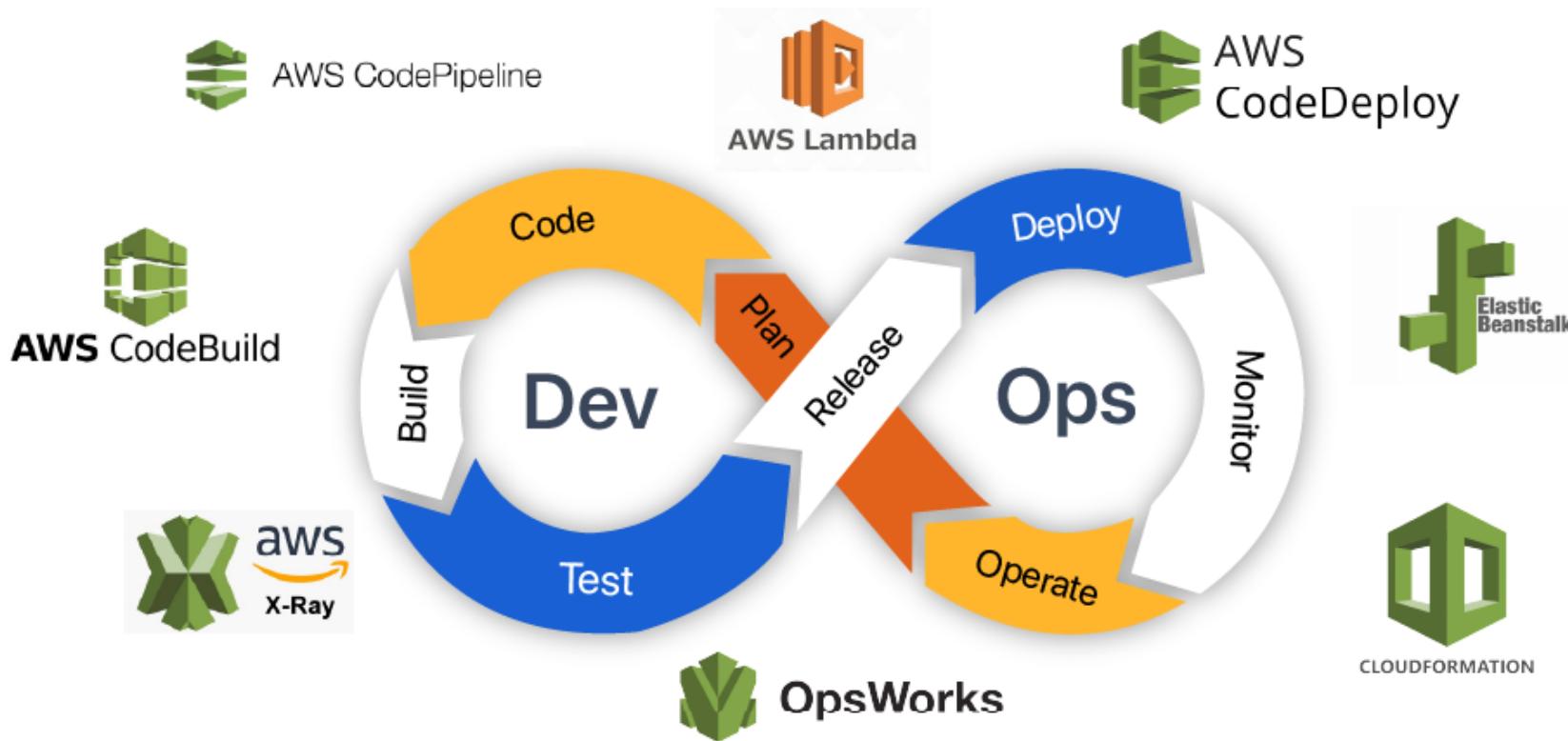
#### Select a statement

Select an existing statement in the policy or add a new statement.

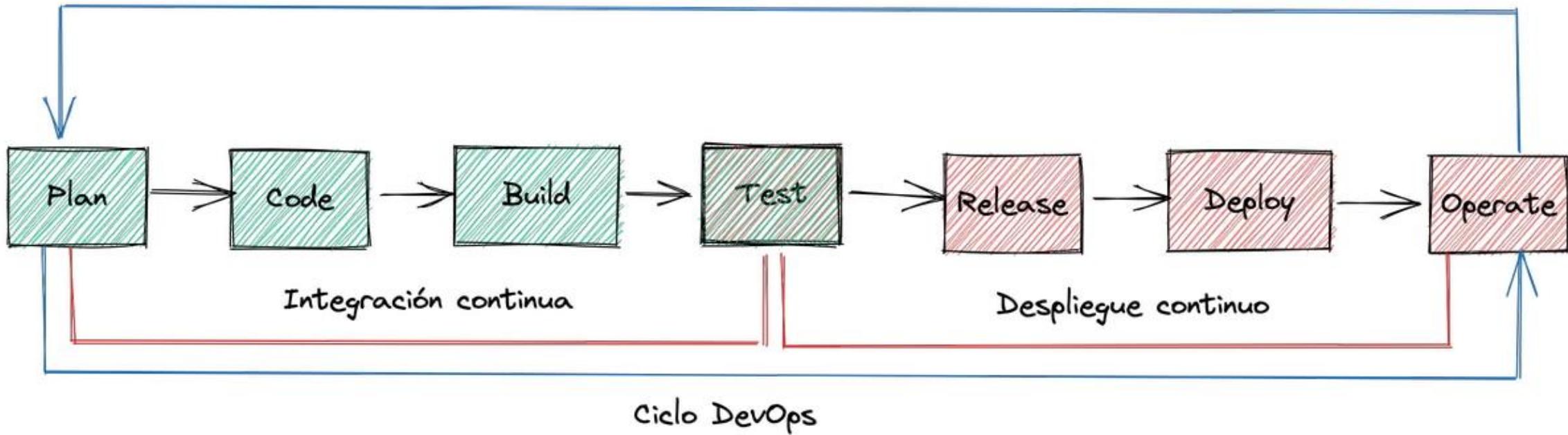
[+ Add new statement](#)

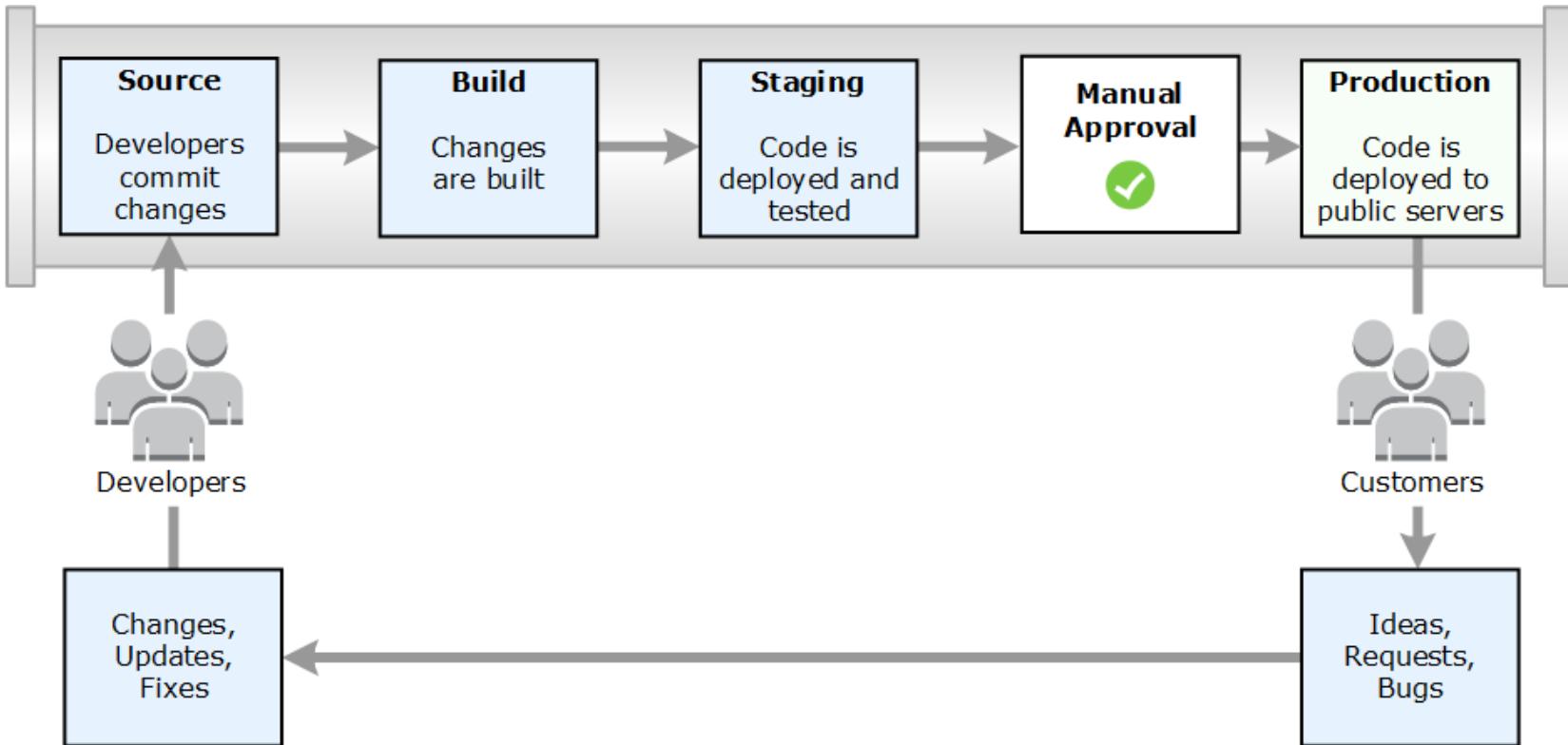
## EXERCISE 7

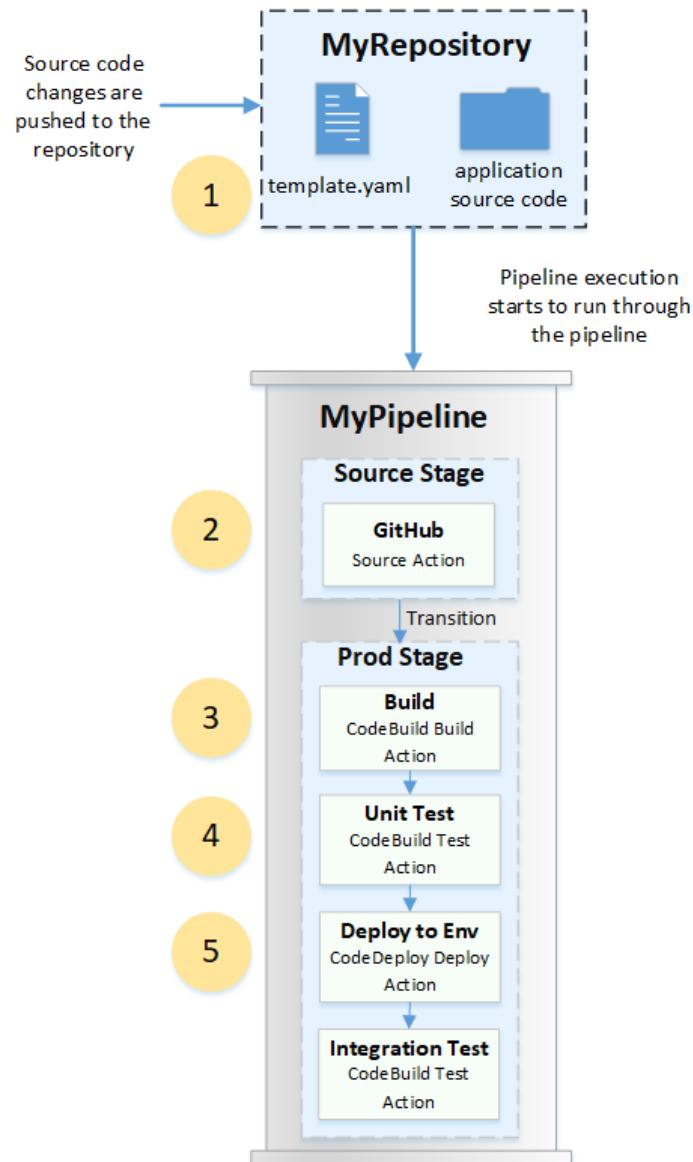


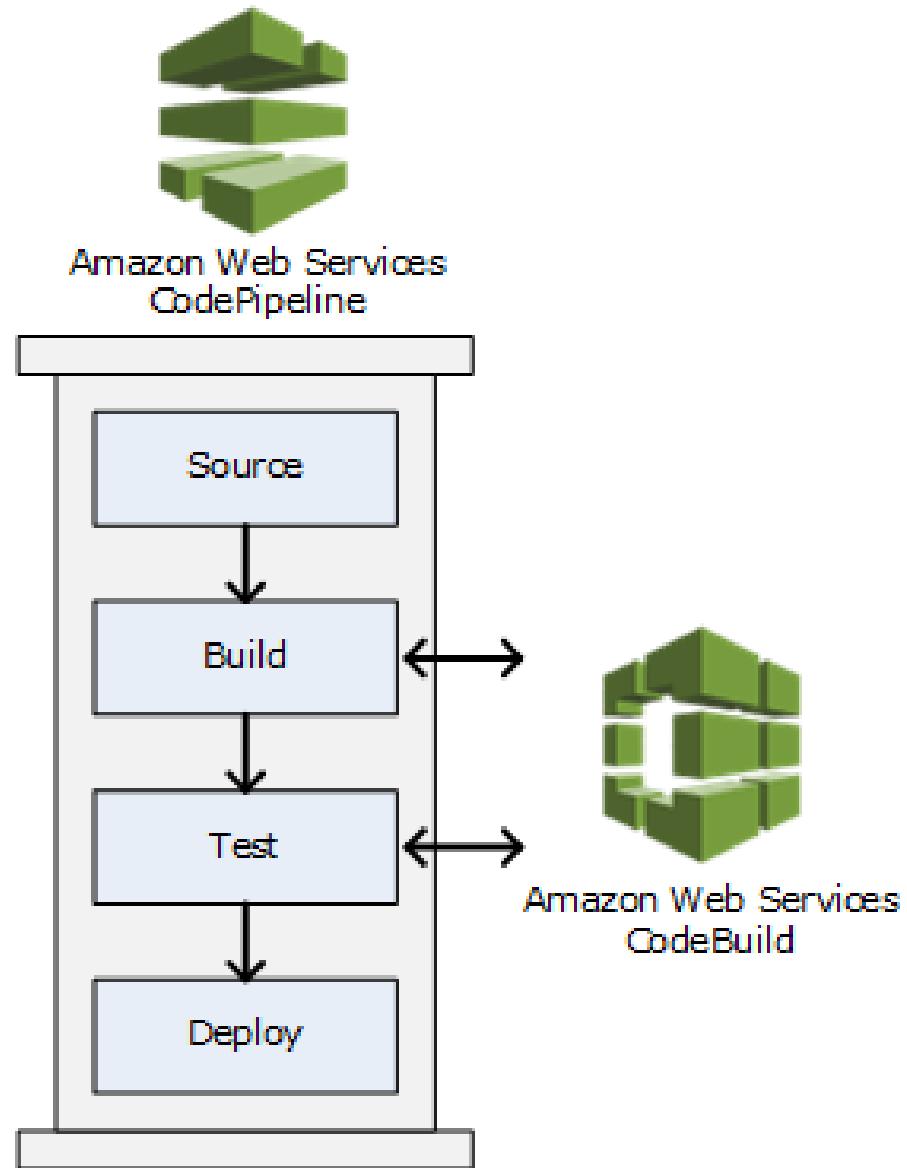


# Top 8 AWS DevOps Tools and Use Cases







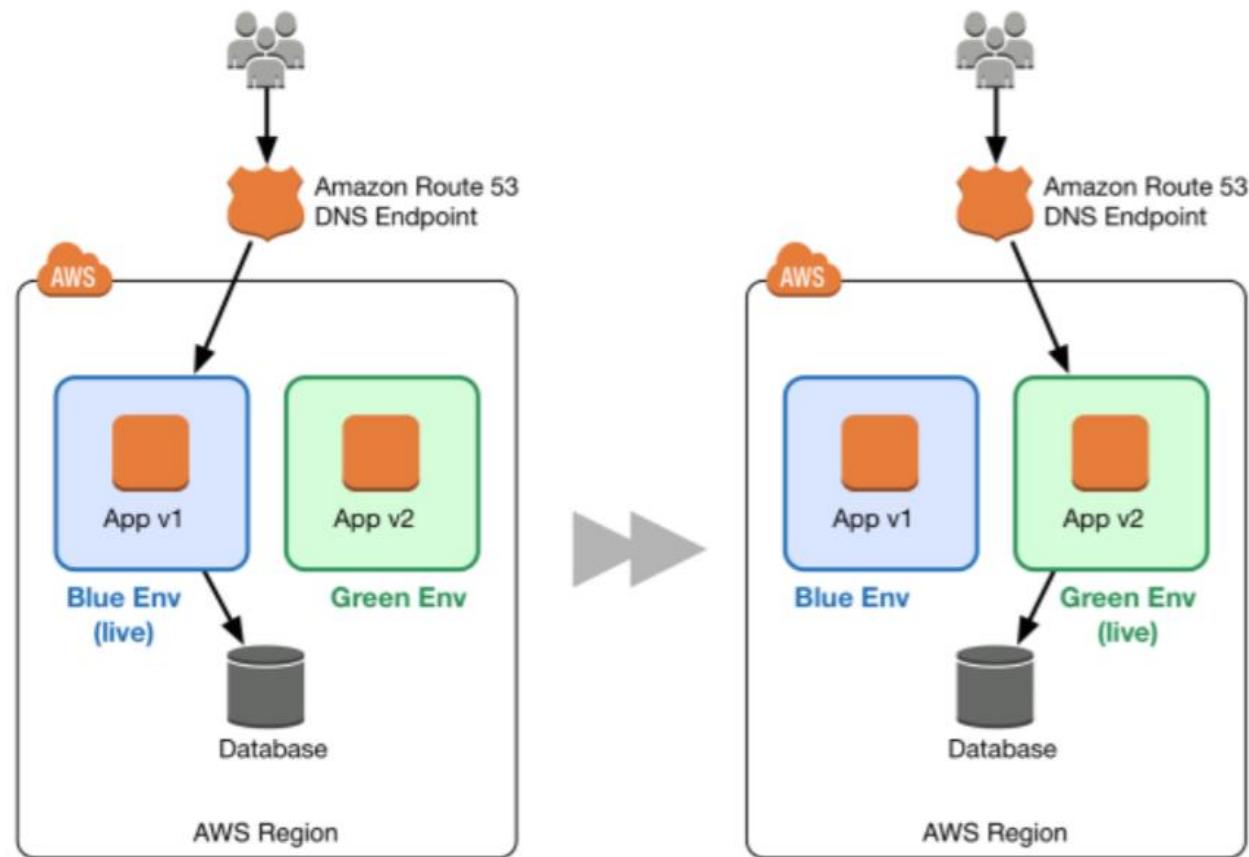


# DEMO

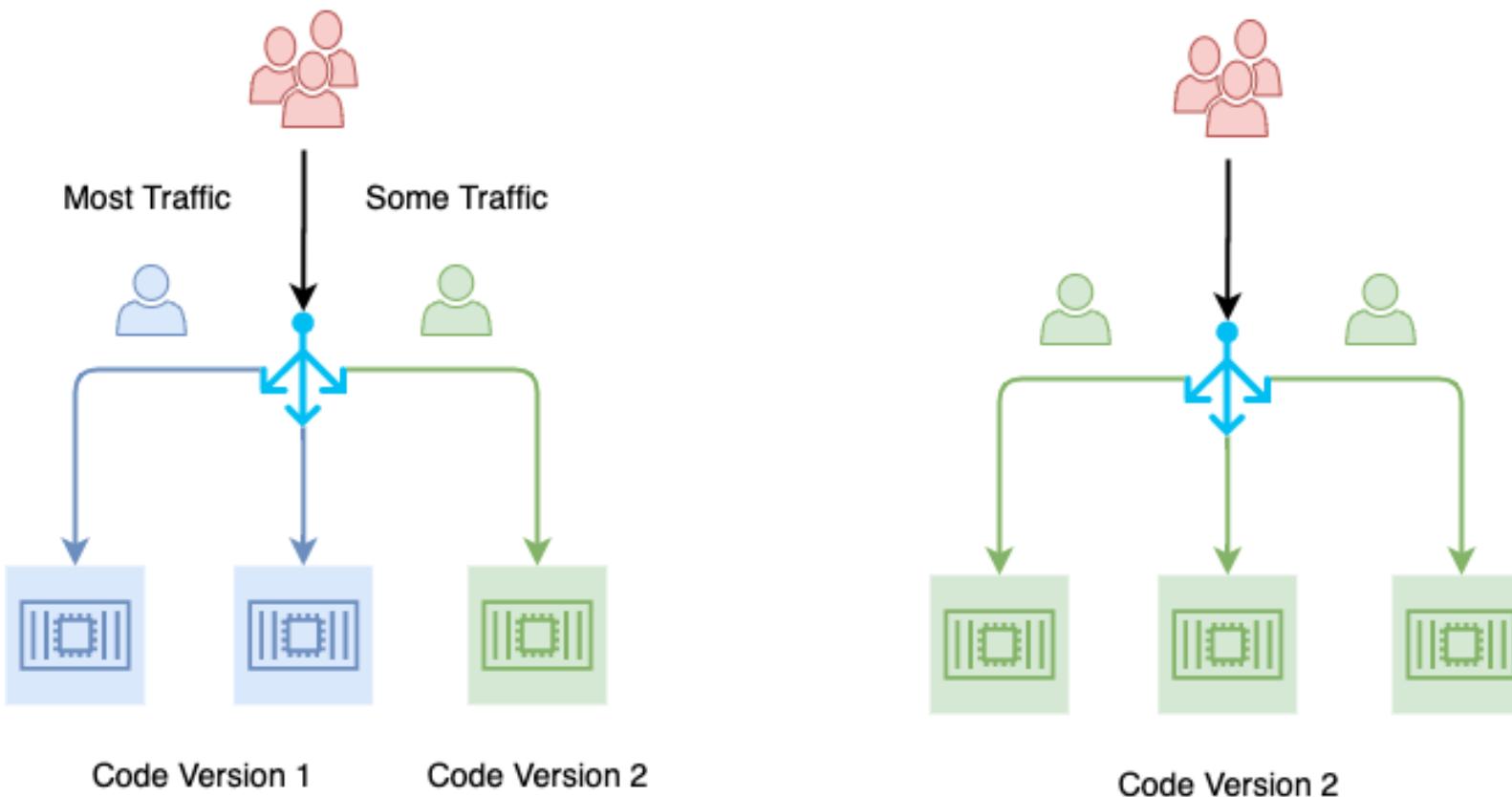


# EXERCISE 8



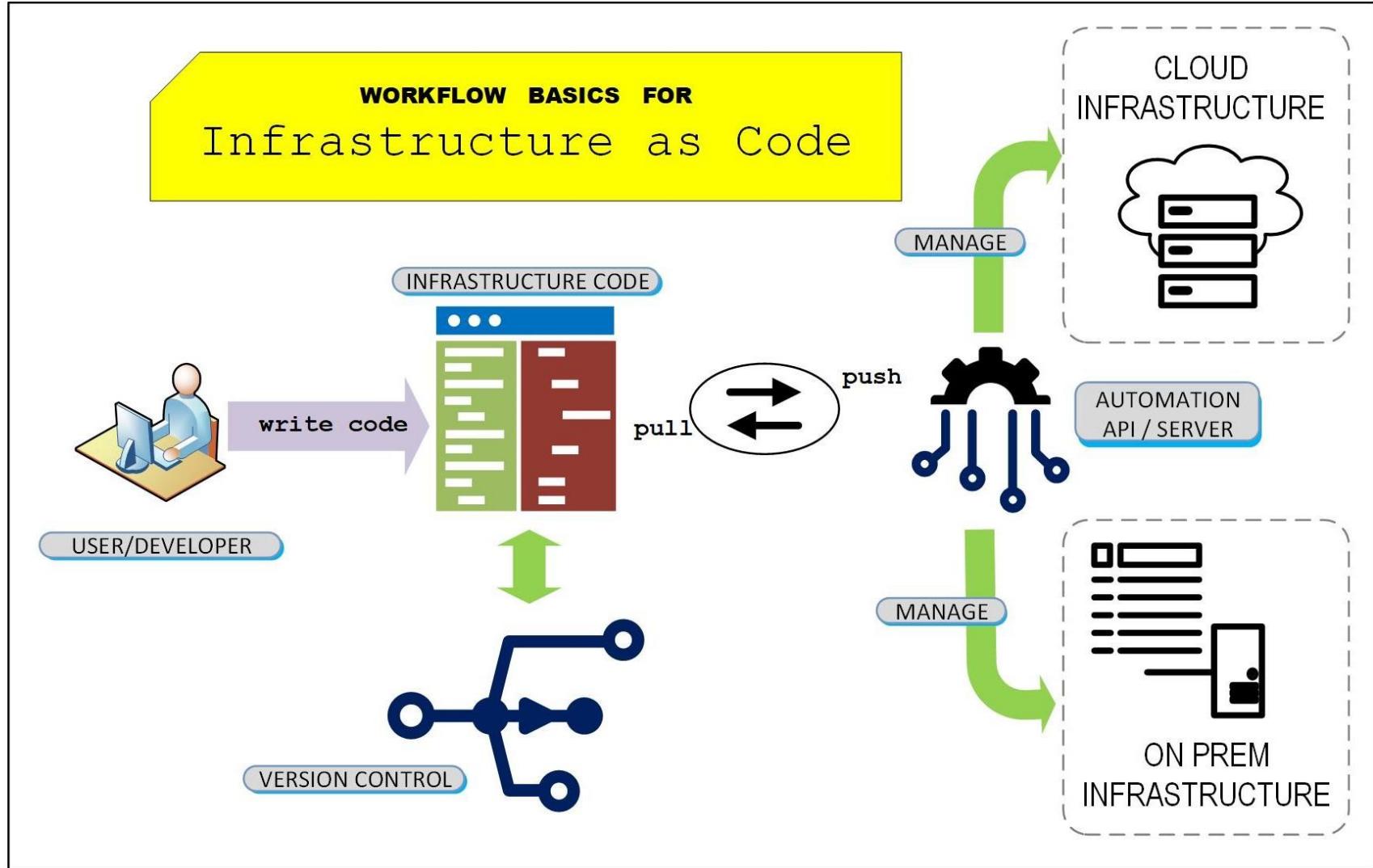


# Canary Deployment



# EXERCISE 9



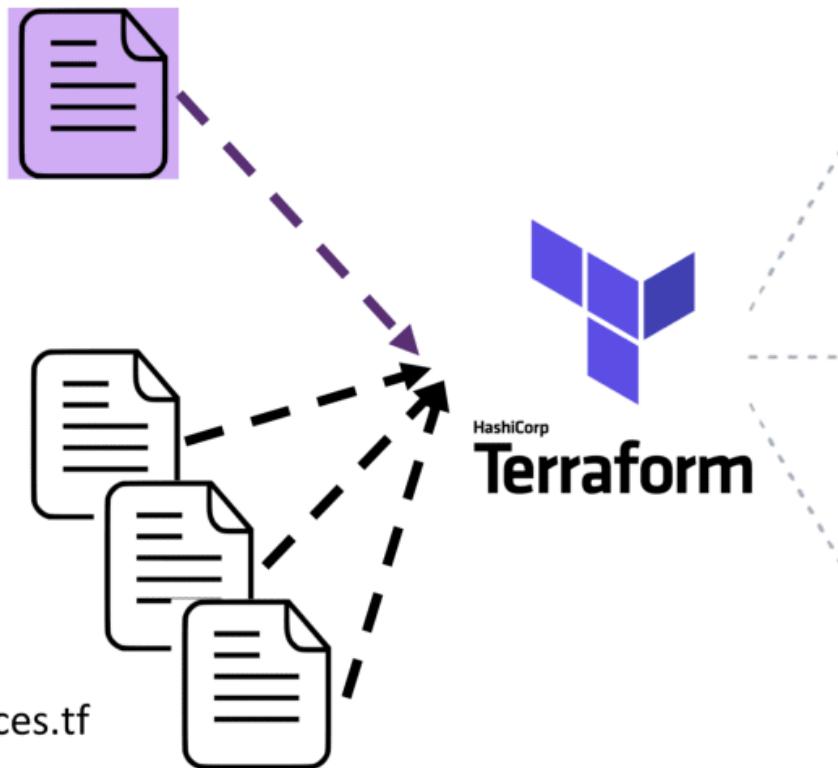


`terraform.state`

`terraform.tfvars`

`terraform-provider.tf`

`terraform- instances.tf`



# DEMO



# EXERCISE 10

