GIT - VERSION ONTROL

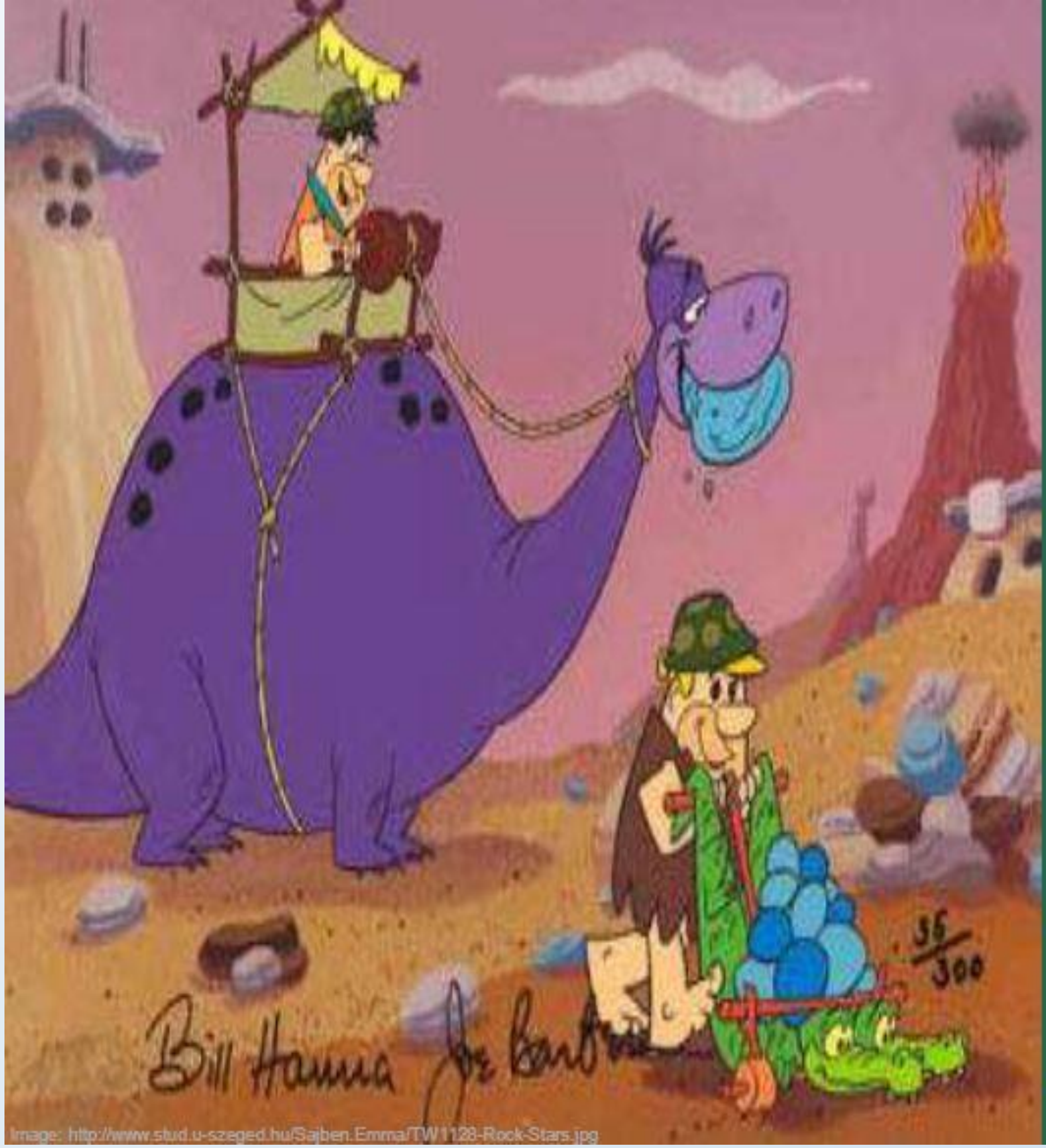


Image: http://www.stud.u-szeged.hu/Sajben.Emma/TW1128-Rock-Stars.jpg

Before Version Control

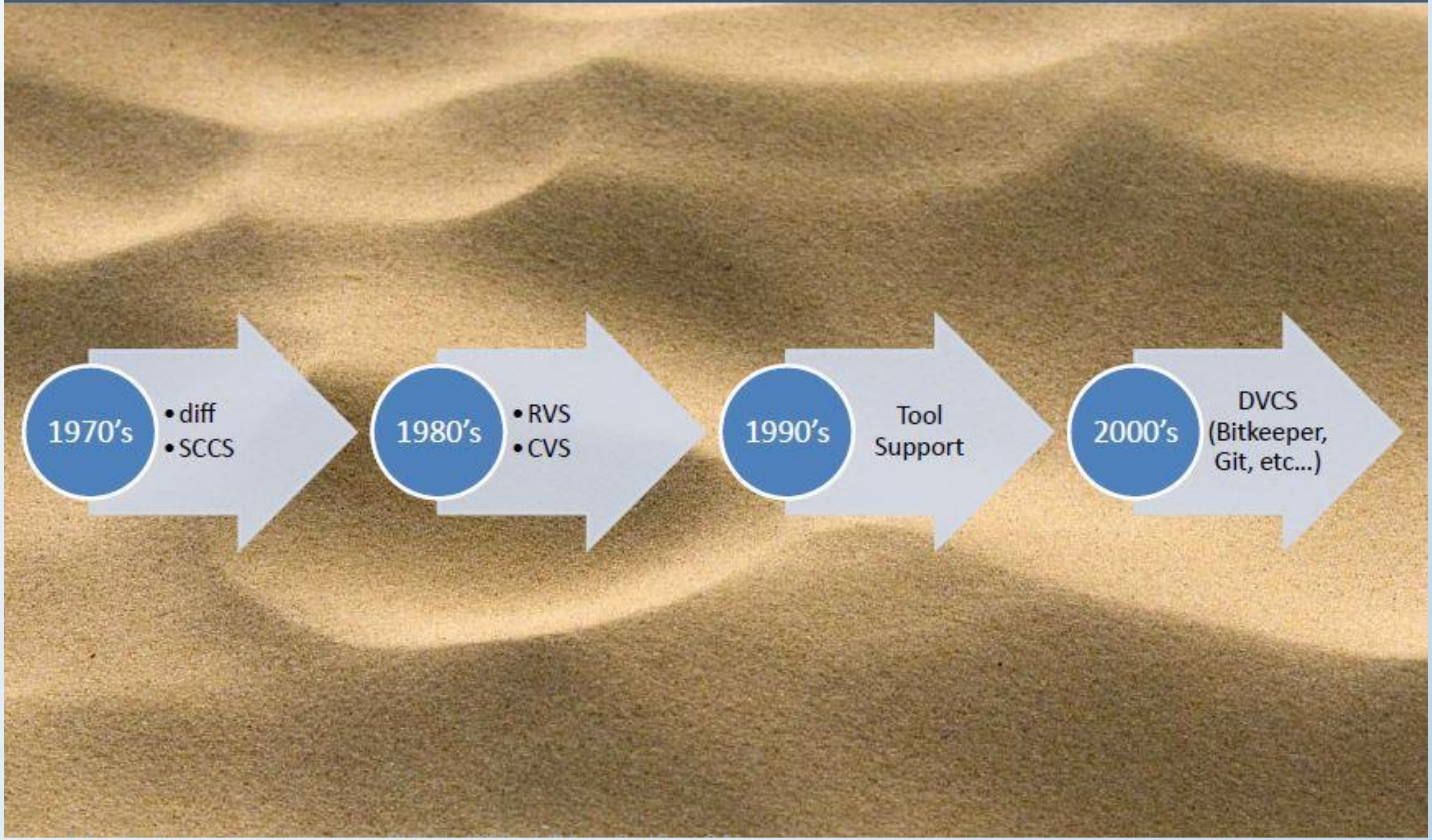
- ✓ File renaming
  - Code.001
  - CodeNov1.xml
- ✓ Directories
  - \Nov1Code
- ✓ Zip files
  - Nov1Code.zip
- ✓ Nothing at all

Image: http://yabadabadoo.files.wordpress.com/2008/03/puzzled-fred.jpg

We have moved on…

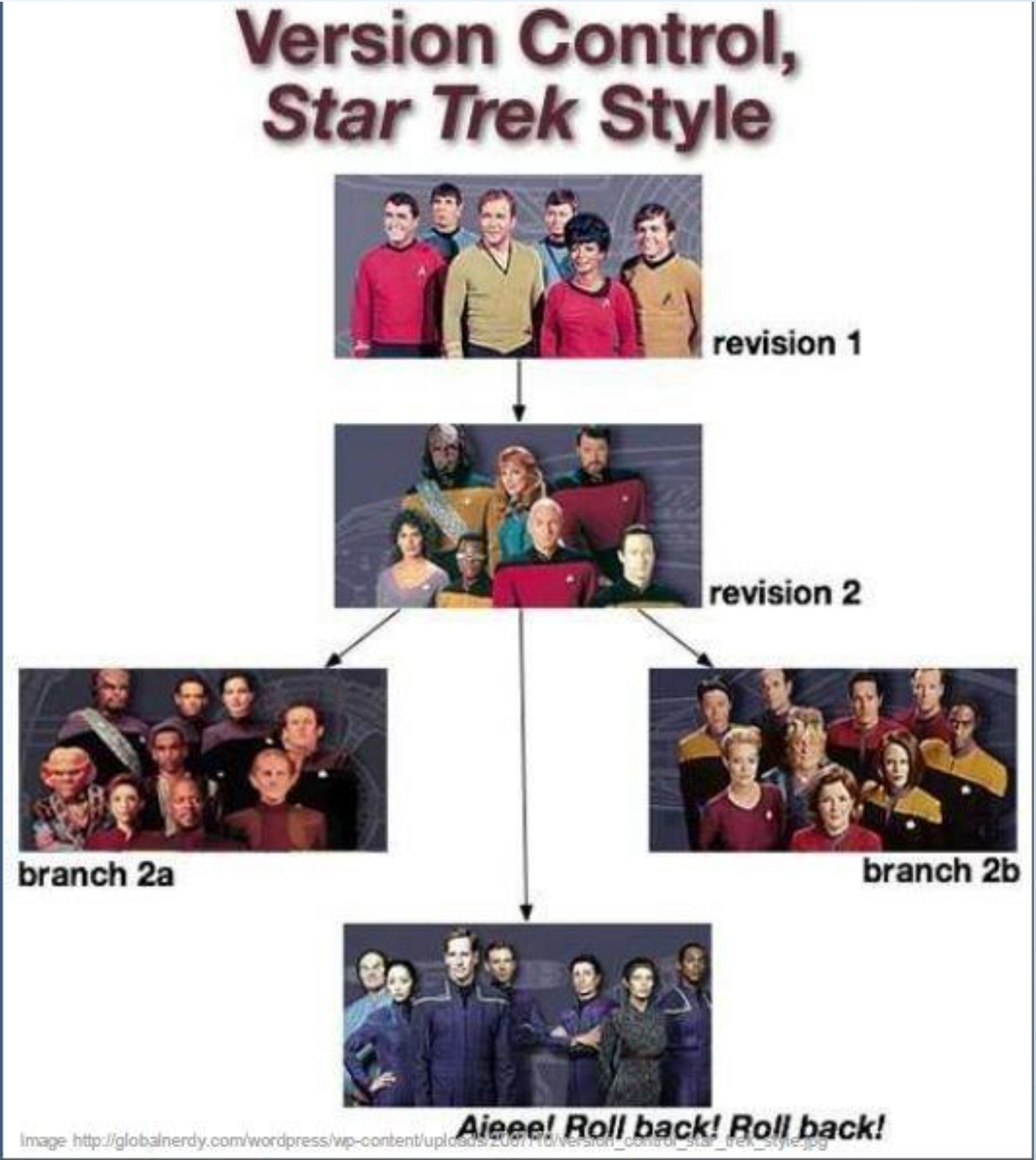
haven’t we?

A Brief History

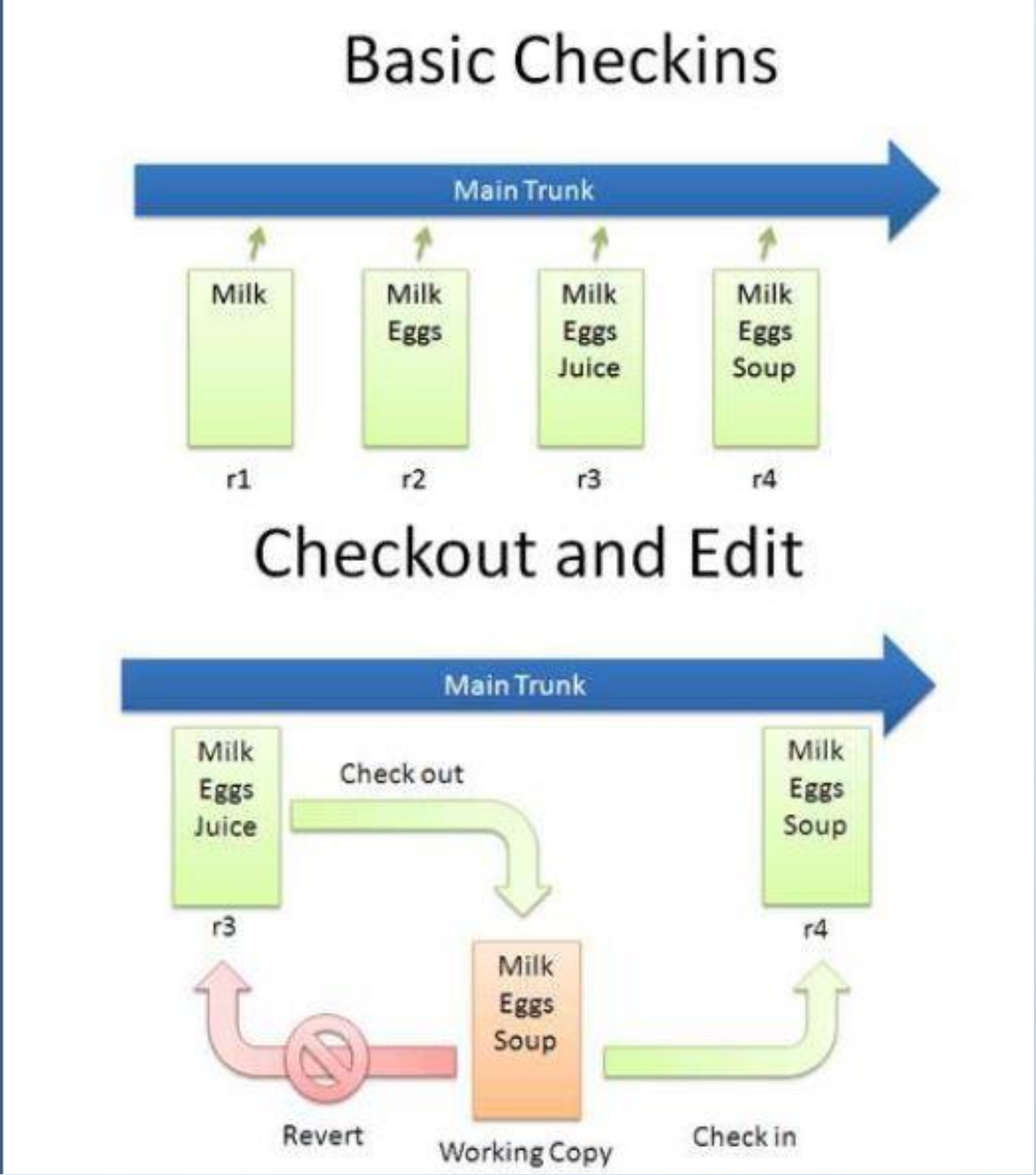


Version Control is…

- ✓ Backup & restore
- ✓ Synchronisation
- ✓ Undo
- ✓ Track changes
- ✓ Sandbox / spike
- ✓ Branch / merge
- ✓ Not just for code


Version Control, *Star Trek Style*
revision 1
revision 2
branch 2a
branch 2b
Aieee! Roll back! Roll back!
Image http://globalnerdy.com/wordpress/wp-content/uploads/2007/10/version_control_star_trek_style.jpg

Check-in & Check-out



Branching & Merging

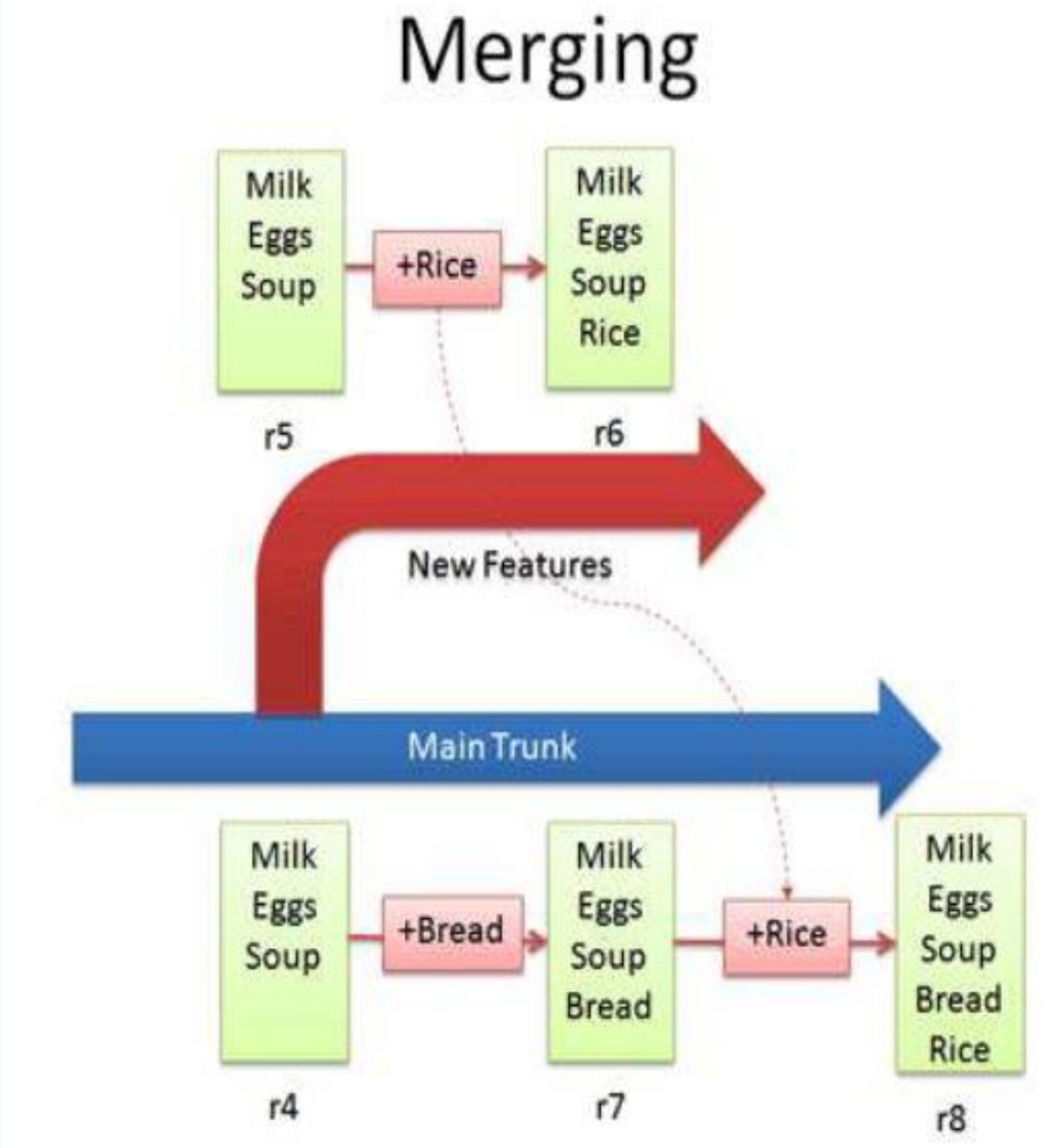


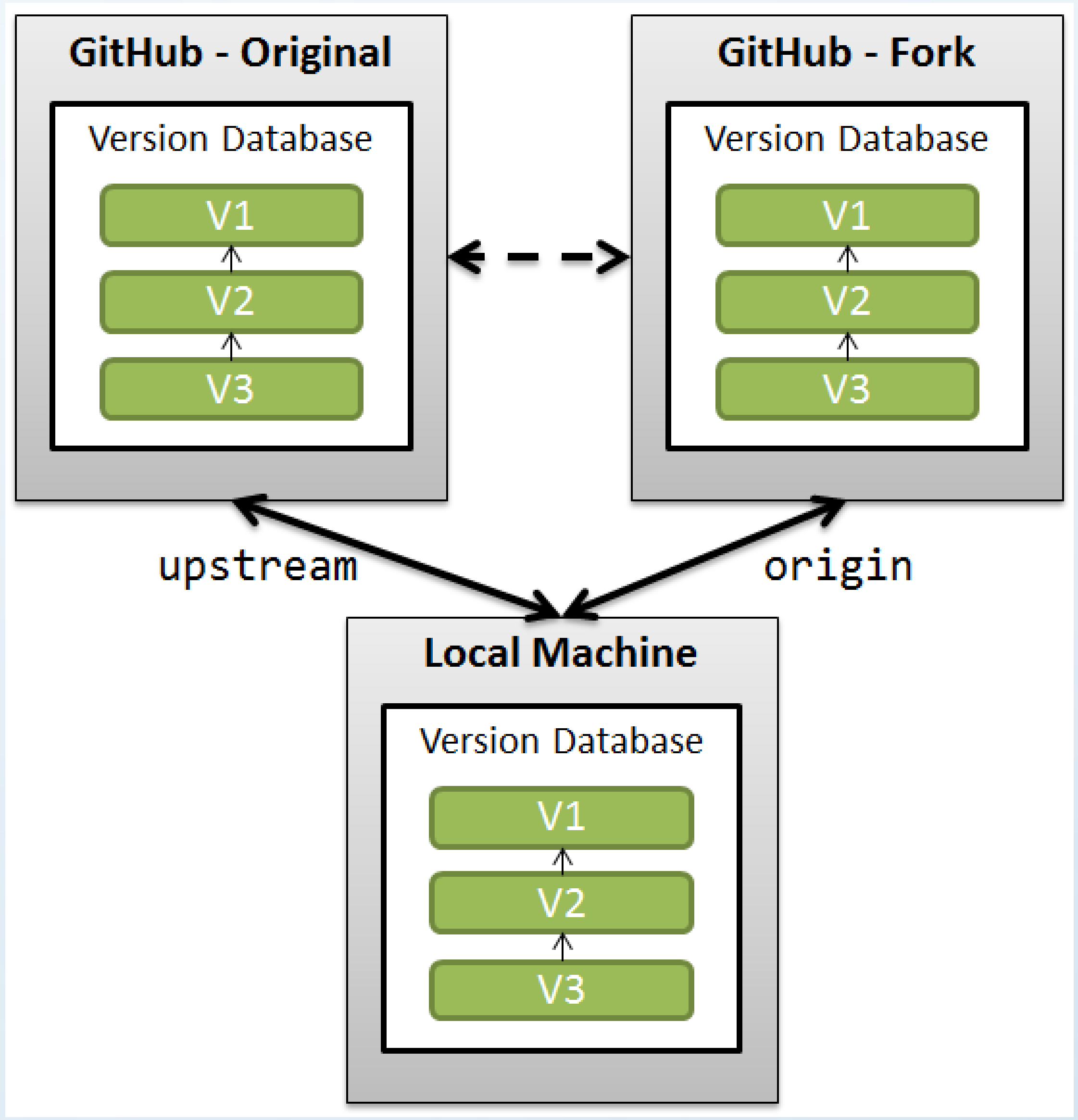
Image http://www.improvizations.com/Portals/42614/images//best-

Version Control Best Practice

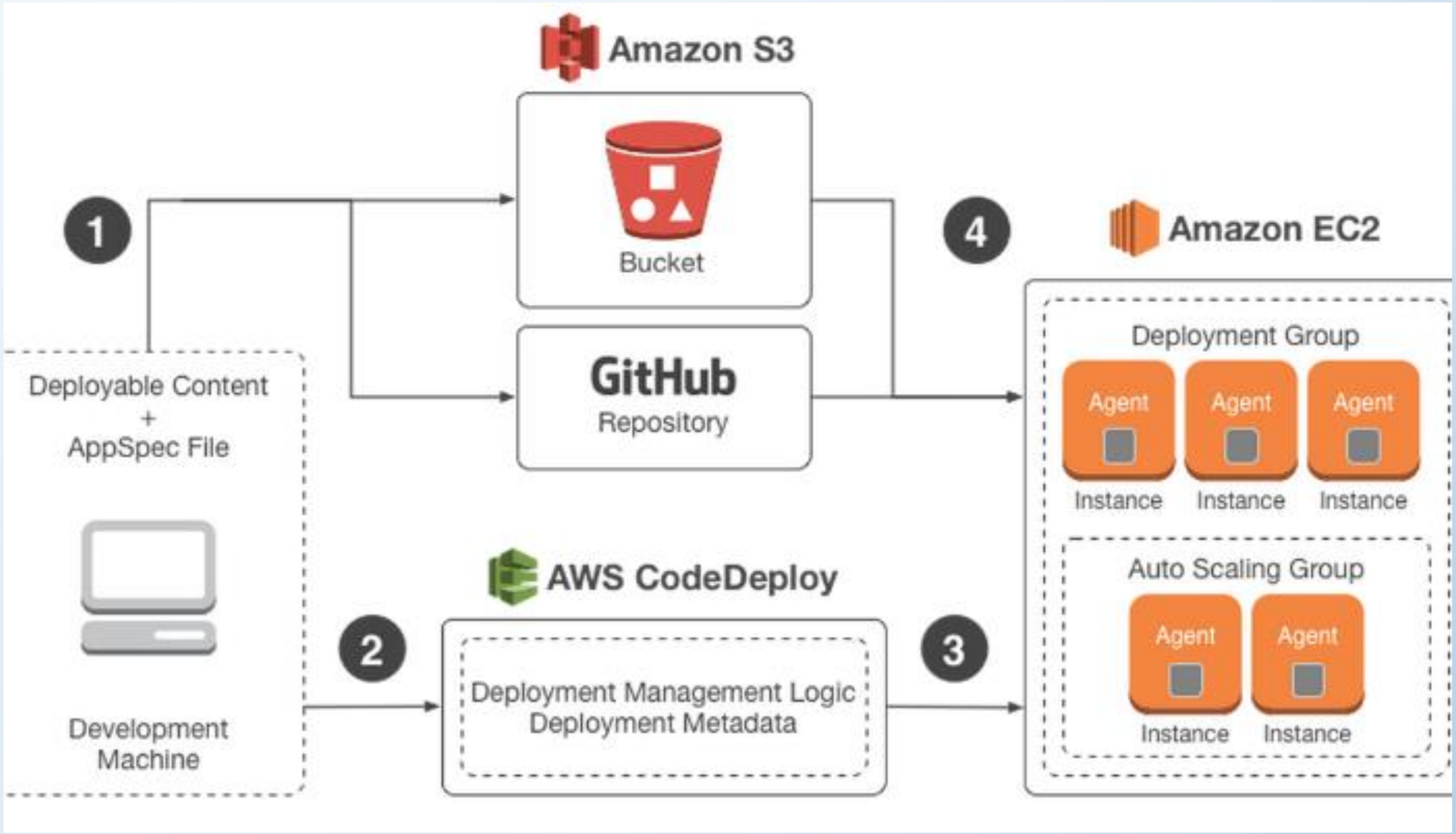
- ✓ Use good comments
- ✓ Commit often
- ✓ Single vs multi project repos
- ✓ Branch/tag when appropriate
- ✓ Binaries/large files
- ✓ Respect the trunk
- ✓ Get to know your command line & client

GITHUB

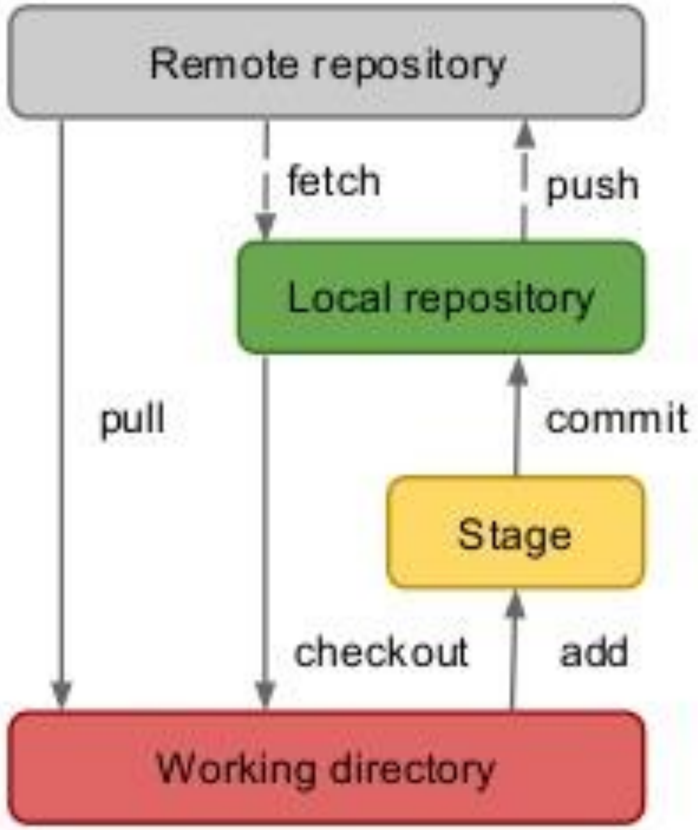
GITHUB


GitHub - Original
Version Database
V1
V2
V3
GitHub - Fork
Version Database
V1
V2
V3
upstream
origin
Local Machine
Version Database
V1
V2
V3

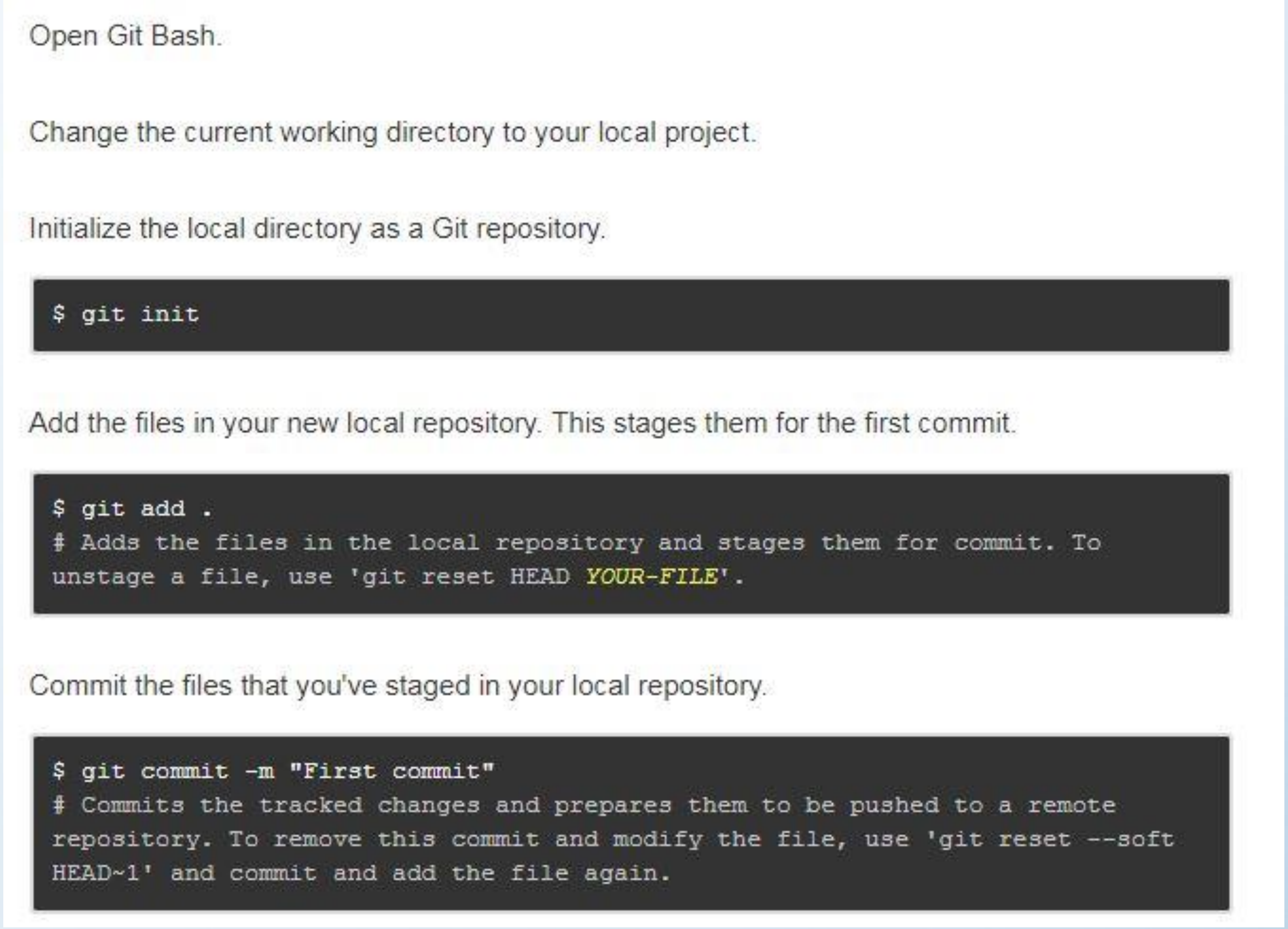
GIT + IAAS


Amazon S3
Bucket
GitHub
Repository
1
4
Amazon EC2
Deployable Content
+
AppSpec File
Development Machine
Deployment Group
Agent
Instance
Agent
Instance
Agent
Instance
AWS CodeDeploy
2
3
Deployment Management Logic
Deployment Metadata
Auto Scaling Group
Agent
Instance
Agent
Instance

Understanding of workflow



- Obtain a repository
  - *git init* or *git clone*
- Make some changes
- Stage your changes
  - *git add*
- Commit changes to the local repository
  - *git commit -m “My message”*
- Push changes to remote
  - *git push remotename remotebranch*



Open Git Bash.

Change the current working directory to your local project.

Initialize the local directory as a Git repository.

```
$ git init
```

Add the files in your new local repository. This stages them for the first commit.

```
$ git add .
# Adds the files in the local repository and stages them for commit. To
unstage a file, use 'git reset HEAD YOUR-FILE'.
```

Commit the files that you've staged in your local repository.

```
$ git commit -m "First commit"
# Commits the tracked changes and prepares them to be pushed to a remote
repository. To remove this commit and modify the file, use 'git reset --soft
HEAD~1' and commit and add the file again.
```

At the top of your GitHub repository's Quick Setup page, click 📋 to copy the remote repository URL.

In the Command prompt, add the URL for the remote repository where your local repository will be pushed.

```
$ git remote add origin remote repository URL
# Sets the new remote
$ git remote -v
# Verifies the new remote URL
```

Push the changes in your local repository to GitHub.

```
$ git push origin master
# Pushes the changes in your local repository up to the remote repository you
specified as the origin
```

Platform As A Service (PAAS)

What is PAAS?

- **P**latform **a**s **a** **S**ervice (**PaaS**) is a class of Cloud Computing services which allows its users to develop, run, and manage applications without worrying about the underlying infrastructure.
- With **PaaS**, users can simply focus on building their applications, which is a great help to developers.
- We can either use **PaaS** services offered by different cloud computing providers like **Amazon**, **Google**, **Azure**, etc., or deploy it on-premise, using software like **Cloud Foundry**, **Heroku**, **Deis**, etc
- **PaaS** can be deployed on top of **IaaS**, or, independently on VMs, bare-metal, and Containers.

CLOUD FOUNDRY

What is Cloud Foundry?

- Cloud Foundry is an Open Source platform as a service (**PaaS**) that provides a choice of clouds, developer frameworks, and application services.
- It can be deployed on-premise or on **IaaS**, like **AWS**, **vSphere**, or **OpenStack**.
- There are many commercial CF cloud providers as well, like **HPE Helion Cloud Foundry**, **IBM Bluemix**, **Pivotal Cloud Foundry**, etc.

Cloud Foundry Subsystems

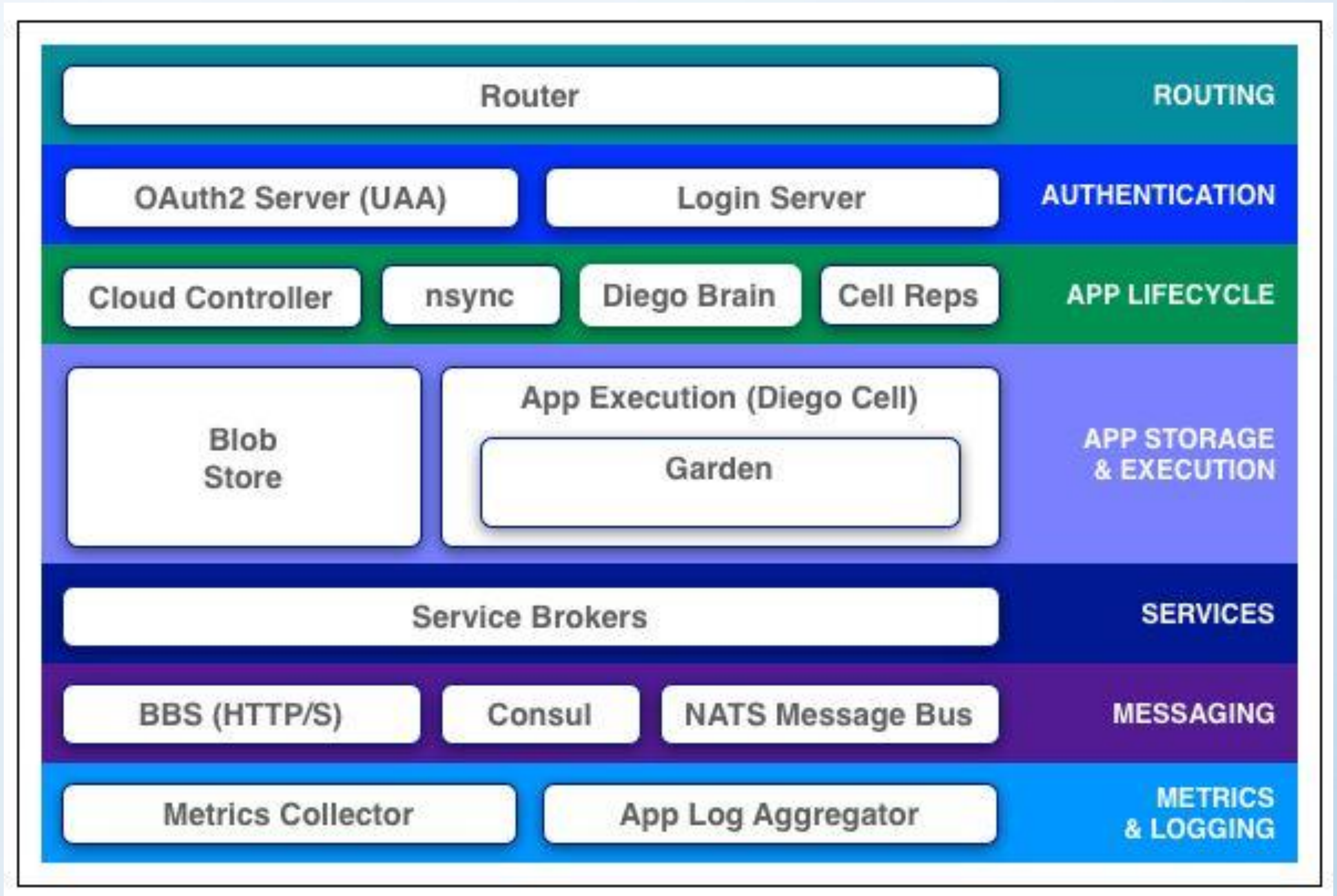
Three major subsystems:

- **Bosh** - **CF** runs on top of VMs from existing **IaaS** like **AWS**, **vSphere**, or **OpenStack**.
- **Cloud Controller** - It runs the applications and other processes on provisioned VMs. It also manages the life-cycle and demand of applications.
- **(Go) Router** - It routes the incoming traffic to the Cloud Controller or the application.

Build-packs

- Provide the framework and runtime support for the applications.
- Build-packs for following languages –
    - Java
    - Python
    - Ruby
    - .Net
    - PHP

Cloud Foundry Components



CF - DEPLOYMENT

- Create cloud group.
- Create Space to deploy the applications.
- Use following commands to deploy apps:
  - cf login -a api.run.pivotal.io
  - cf push <app_name>
  - cf logs <app_name> --recent
  - cf scale <app_name> –i <numer_of_instance>

- Procfile
- requirements.txt
- test.py

FLASK - PYTHON

```
from flask import Flask
import os

app = Flask(__name__)

port = int(os.getenv("PORT"))

@app.route('/')
def hello_world():
        return 'Welcome'

if __name__ == '__main__':
        app.run(host='0.0.0.0',port=port)
```