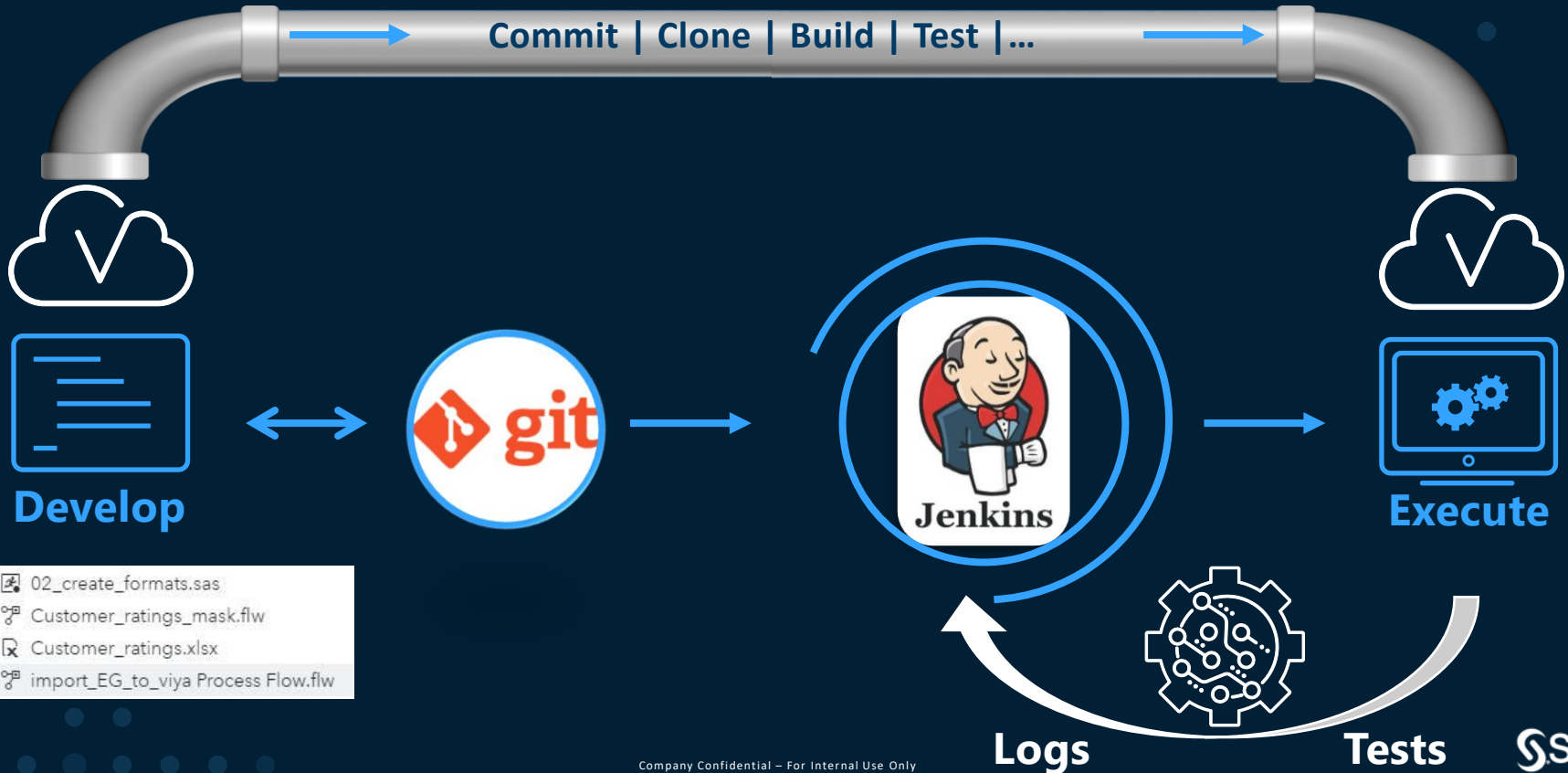


Devops with git and Jenkins



Dataops med devops mot SAS Viya med git og Jenkins

DevOps Applied to SAS Viya Data Management

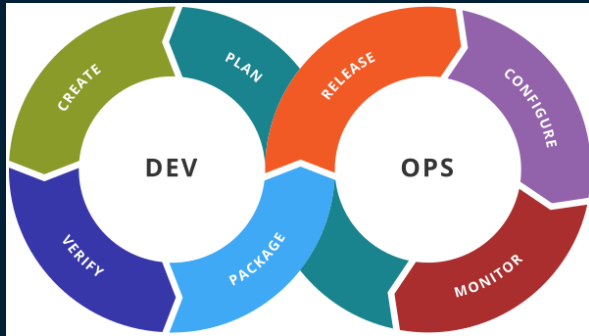
Litt fag og litt demo

DataOps for Analytics ...

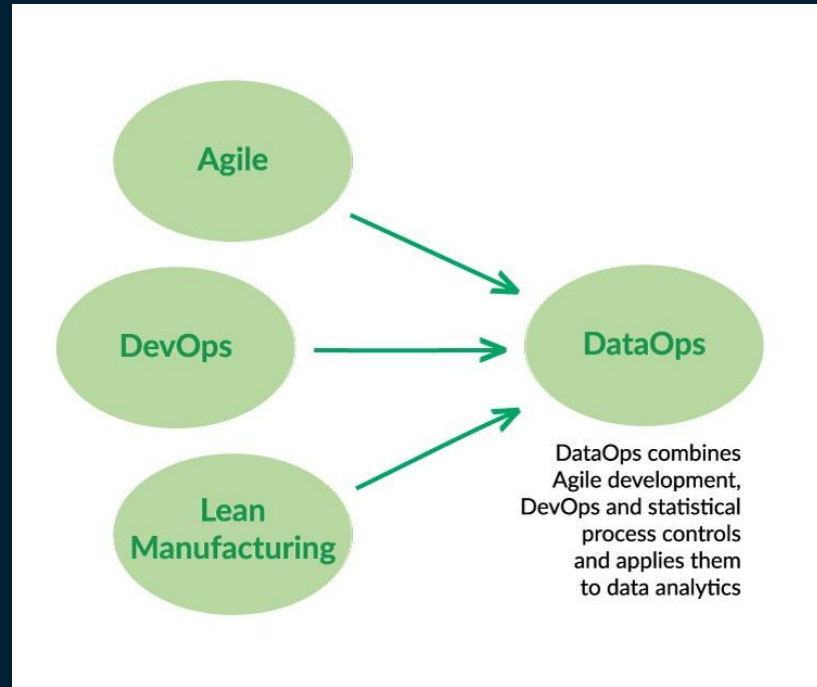
Trusting in the data as we trust in the water we drink..



DataOps and DevOps

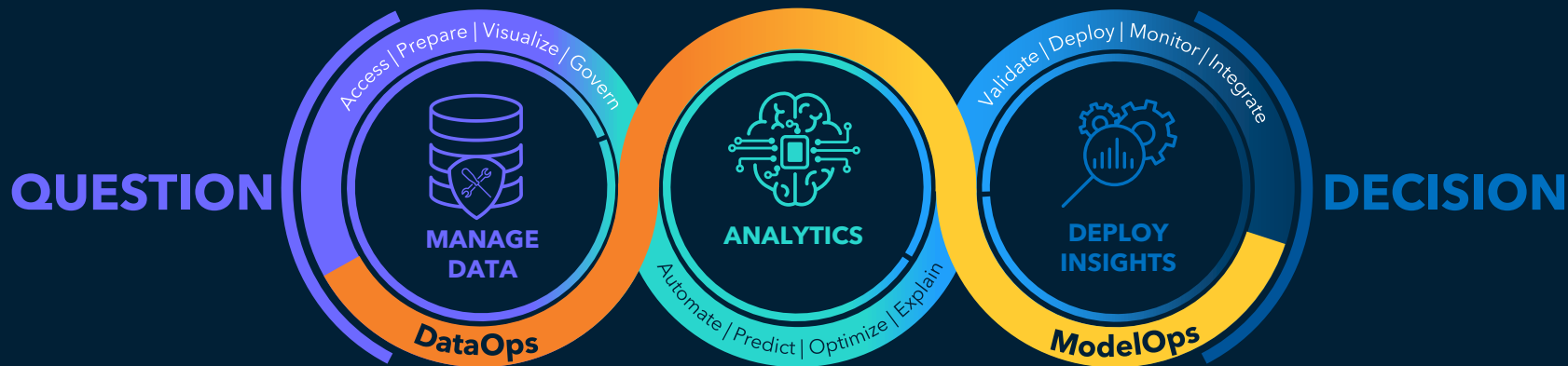


Source:medium.com



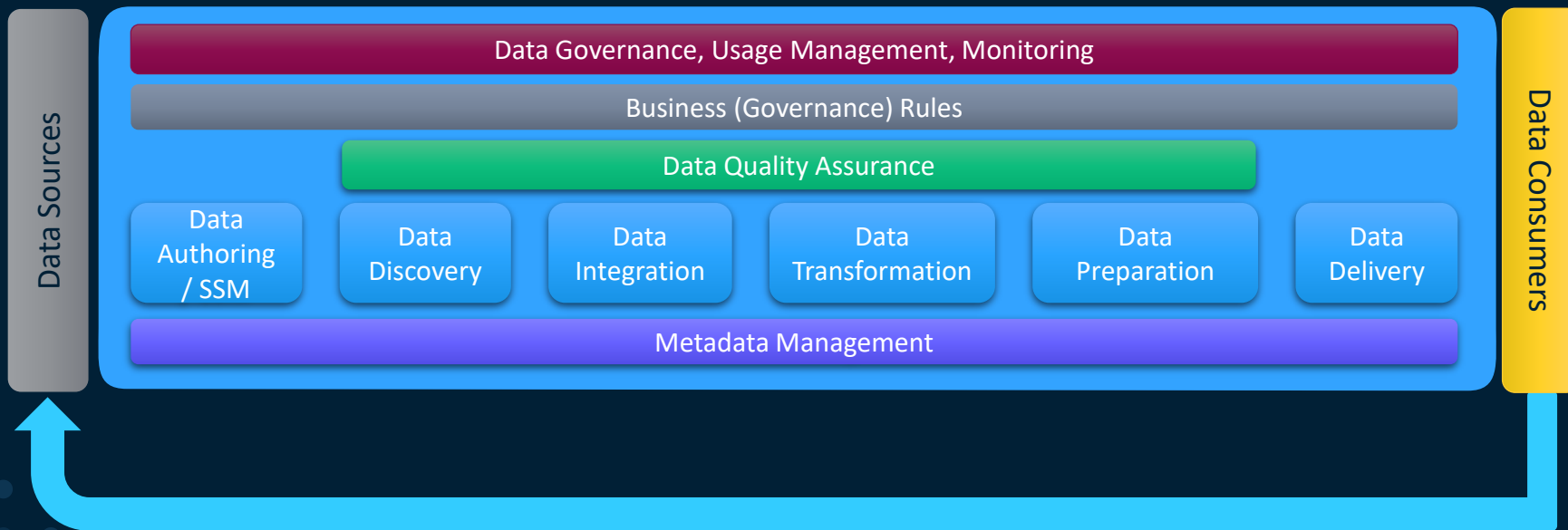
Analytics Life Cycle

Automation, Operationalizing, Industrializing



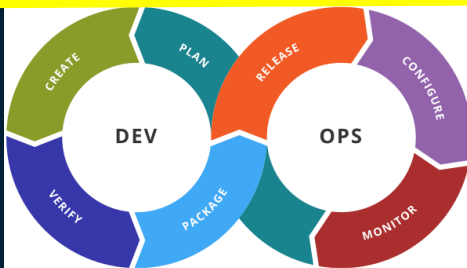
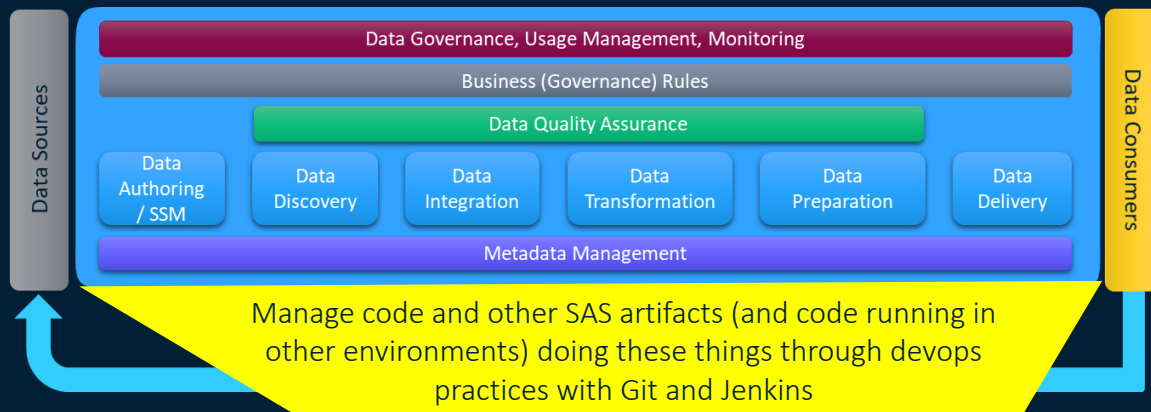
DataOPs – Components

The **How** and the **What** ... From the Source to the Consumer

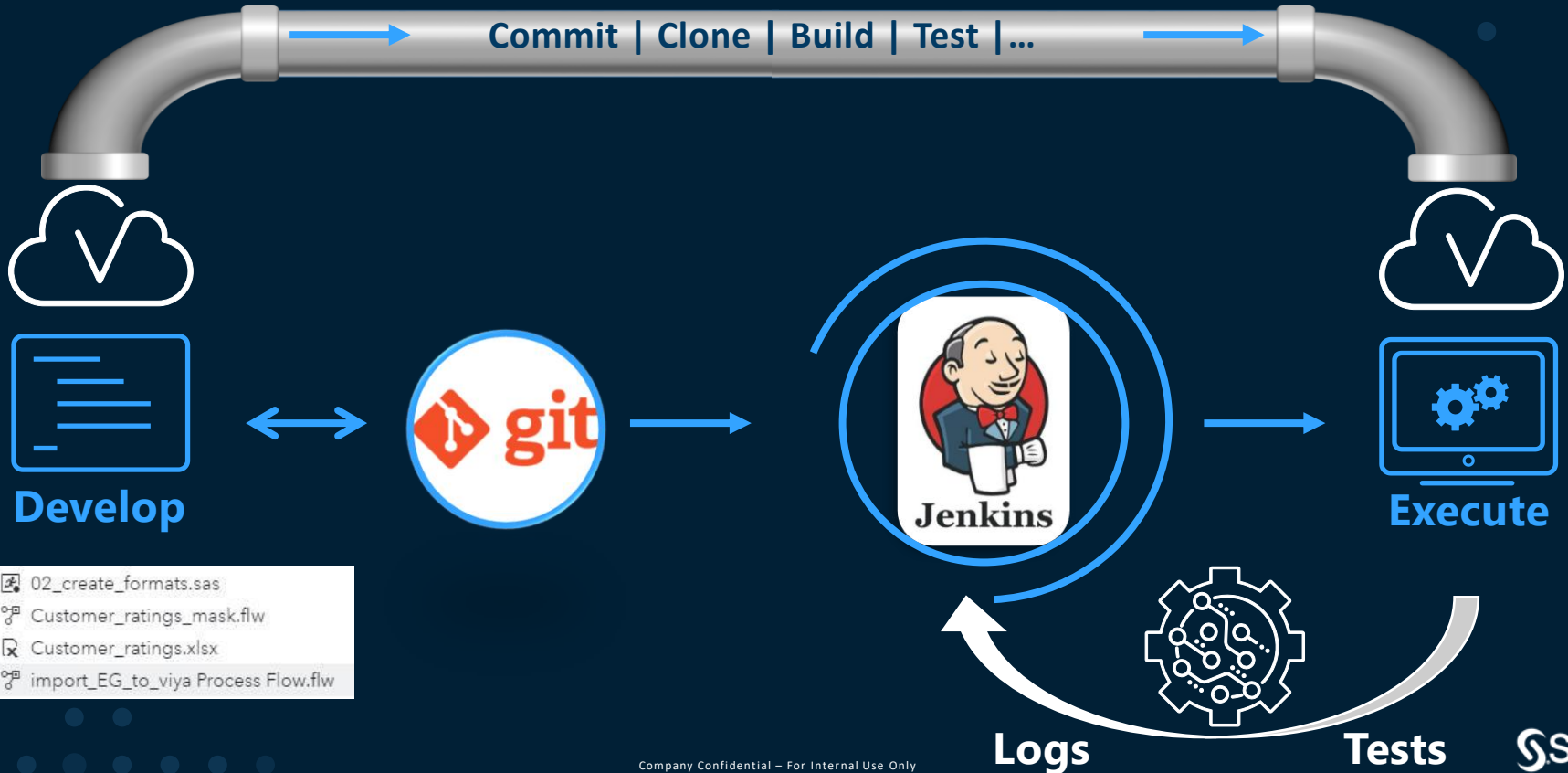


DataOPs – with Devops

Getting your ETL, data prep, data quality and other dataops tasks in operations smooth and frequent through devops practices



Devops with git and Jenkins



Demo time

How does it look? Using Git and Jenkins with SAS Viya

Git

- Configure git in SAS Studio
- Get your code into git – learn to trust git
- Local vs remote repository

Jenkins

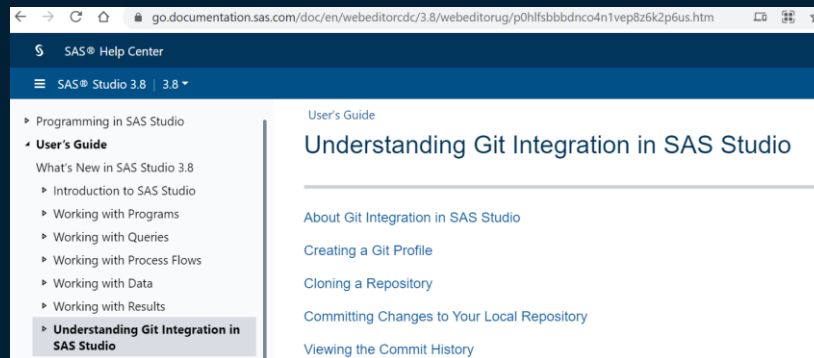
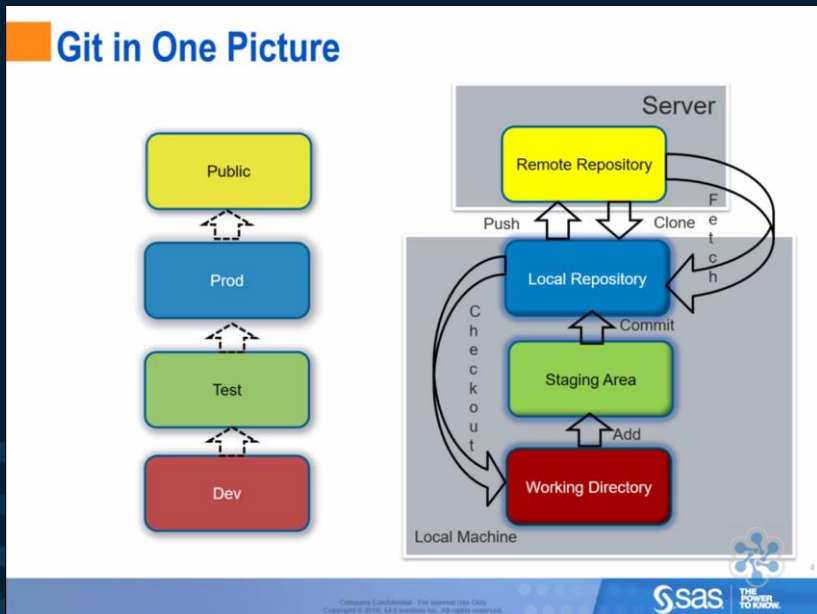
- Ways of automation – freestyle and pipelines
- Integrate with gitlab and SAS Viya
- Run programs and flows from Jenkins
 - (programs as batch, flows as job)
- CI / CD with automated tests

Git Commands

Resources

Introduction to Git

Git guides



- Youtube: Git Integration in SAS Studio and SAS Enterprise Guide
 - <https://www.youtube.com/watch?v=CZwT7kxmBmw>
- <https://www.atlassian.com/git/tutorials/atlassian-git-cheatsheet>
- <https://www.git-tower.com/blog/git-cheat-sheet/>

SAS Studio

Git plug-in

The screenshot displays the SAS Studio interface with the Git plug-in active. The window title is "SAS® Studio - Develop SAS Code". The top menu bar includes "New", "Options", "View", "Open", and "Save All". The left sidebar shows "Git Repositories" with options to "Clone" and "Add", and the current repository is "devops".

The main workspace shows the "devops" repository and the "feature1" branch. The current branch is "feature1", and the last pulled commit is from Nov 11, 2020. The "Push feature1" button is visible.

The commit history table is as follows:

Commit	Message	Author	Date	Commit ID
4664944	✓ feature1 feature 1 developed and tested	geladm	Nov 11, 2020, 1:59:06 AM	4664944
192c7a7	✗ master origin/master correction gitlab-synch ...	Bogdan Teleuca	Nov 10, 2020, 11:19:11 PM	192c7a7
2c05264	redeploy gitlab	Bogdan Teleuca	Nov 10, 2020, 11:07:59 PM	2c05264
670481e	gitlab install and synch DONE!!!	Bogdan Teleuca	Nov 10, 2020, 1:20:47 AM	670481e
cc4e953	casautoin .sas files	Bogdan Teleuca	Nov 10, 2020, 1:06:20 AM	cc4e953
e8dea59	gitlab deploy and synch	Bogdan Teleuca	Nov 10, 2020, 12:51:45 AM	e8dea59

The diff view for the selected commit (4664944) shows the following changes:

```
diff --git a/Data-Management/scripts/900_cleanup.sas b/Data-Management/scripts/900_cleanup.sas index 9c81abb..@@ -28,4 +28,12 @@ droptable casdata="&userid_catcode" quiet;droptable casdata="&userid_customers" quiet;quit;+* Drop sources files;+proc casutil incaslib="&DM_SRC";+deletesource casdata="mailorder.csv" quiet;+deletesource casdata="products.csv" quiet;
```

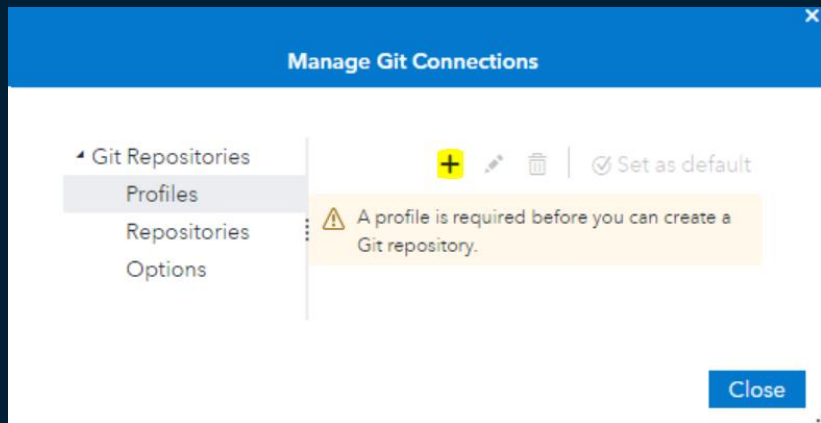
Sample Git Workflow

Steps

- Create a Git profile
- Clone a repository
- Pull the latest changes from the remote repository
- Create a new *feature* branch
- Make changes and test them
- Stage the changes
- Commit the changes
- Merge the *feature* branch back into the master branch
- Push the changes remotely

Sample Git Workflow

Create a Git profile (HTTP clone)



The screenshot shows the 'Edit Profile' dialog box. It contains the following fields and controls:

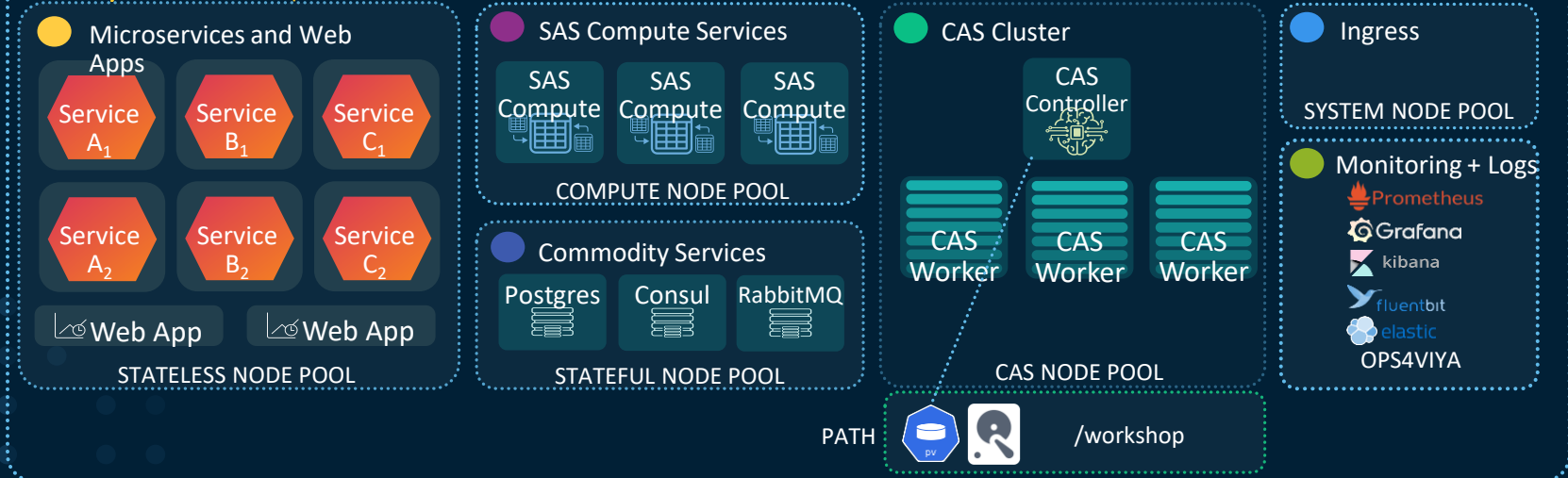
- Profile name:** * Required field, containing the text 'geladm'.
- User name:** * Required field, containing the text 'geladm'.
- Email:** * Required field, containing the text 'geladm@none.sas.com'.
- Public SSH file path:** An empty text field with a folder icon on the right.
- Private SSH file path:** An empty text field with a folder icon on the right.

At the bottom right, there are two buttons: a blue 'OK' button and a grey 'Cancel' button.

Environment Overview



SAS Viya sse namespace



Automation Server

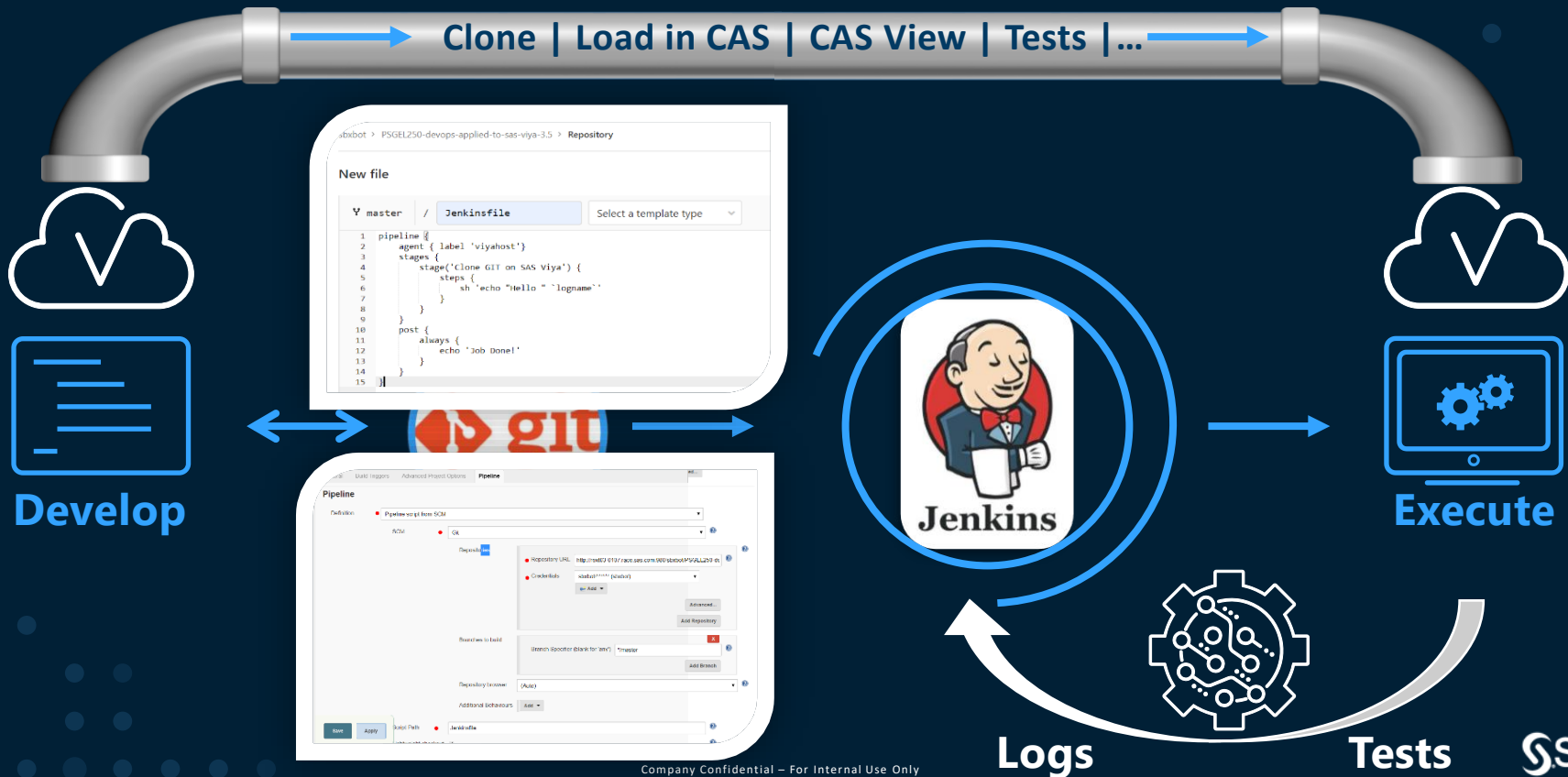
Jenkins

jenkins.io



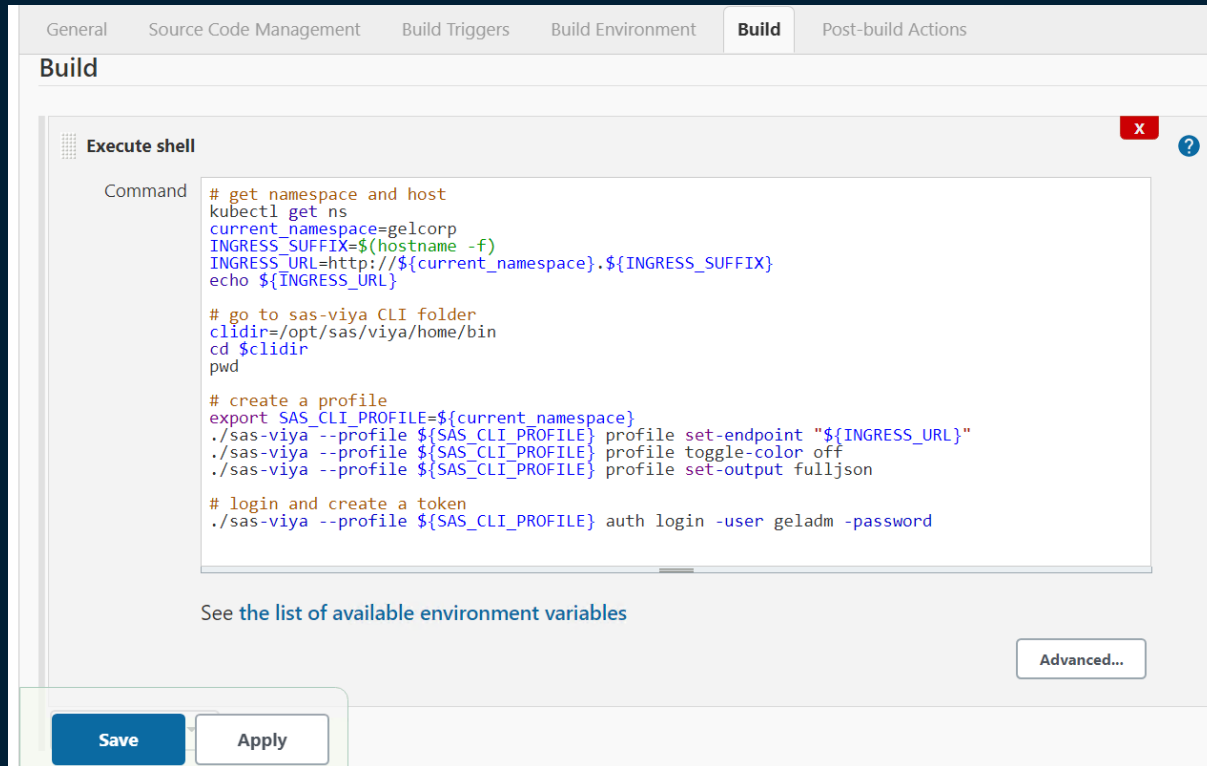
- Open-source, free automation software
- Used to automate software building, testing, deployment and delivery
- Typically associated with software development CI/CD
- Can be used to automate many facets of SAS Viya
- Extensible with numerous available plug-ins
- Fast deployment available via Docker image (or k8s)
- Good documentation
- Version control integration

Automation Server



Automation Server

Jenkins Project



The screenshot shows the Jenkins 'Build' configuration page. The 'Execute shell' section is active, displaying a series of shell commands for setting up the environment and running the application. The commands are as follows:

```
# get namespace and host
kubectl get ns
current_namespace=gelcorp
INGRESS_SUFFIX=$(hostname -f)
INGRESS_URL=http://${current_namespace}.${INGRESS_SUFFIX}
echo ${INGRESS_URL}

# go to sas-viya CLI folder
clidir=/opt/sas/viya/home/bin
cd $clidir
pwd

# create a profile
export SAS_CLI_PROFILE=${current_namespace}
./sas-viya --profile ${SAS_CLI_PROFILE} profile set-endpoint "${INGRESS_URL}"
./sas-viya --profile ${SAS_CLI_PROFILE} profile toggle-color off
./sas-viya --profile ${SAS_CLI_PROFILE} profile set-output fulljson

# login and create a token
./sas-viya --profile ${SAS_CLI_PROFILE} auth login -user geladm -password
```

Below the command field, there is a link: [See the list of available environment variables](#). At the bottom right of the configuration area is an 'Advanced...' button. At the bottom of the page are 'Save' and 'Apply' buttons.

Automation Server

Jenkins Pipeline and Jenkinsfile

The screenshot shows the Jenkins Pipeline console for a job named 'geladm:PSGEL2xx-devops-applied-to-sas-viya-202yz 1'. The pipeline is in a 'Completed' state with a green checkmark. The console output shows the following steps:

```
1 the recommended git tool is: none
2 using credential: e0f98c3124-sacs-9118-75e805913f6
3 Cloning the remote git repository
4 Cloning repository http://reft00-0232.racc.sas.com/908/geladm/psg2xx-devops-applied-to-sas-viya-202yz.1.git
5 > git init /tmp/workspace/geladm-PSGEL2xx-devops-applied-to-sas-viya-202yz.1 # timeout=10
6 Fetching upstream changes from http://reft00-0232.racc.sas.com/908/geladm/psg2xx-devops-applied-to-sas-viya-202yz.1.git
7 > git -version # timeout=10
8 > git -version # 'git version 2.24.1'
9 using git_SSHPS65 to set remoteURL: geladm@reft00
10 > git fetch --tags --force --progress --http://reft00-0232.racc.sas.com/908/geladm/psg2xx-devops-applied-to-sas-viya-202yz.1.git --refs/heads/*:refs/remotes/origin/* # timeout=10
11 Avoid second fetch
12 Checking out revision eac7218c3486d1ee68c8c599693f795ca20b (refs/remotes/origin/master)
13 > git config remote.origin.url http://reft00-0232.racc.sas.com/908/geladm/psg2xx-devops-applied-to-sas-viya-202yz.1.git # timeout=10
14 > git config --add remote.origin.fetch refs/heads/*:refs/remotes/origin/* # timeout=10
15 > git rev-parse refs/remotes/origin/master^{commit} # timeout=10
16 > git config core.sparsecheckout # timeout=10
17 > git checkout -f eac7218c3486d1ee68c8c599693f795ca20b # timeout=10
18 Commit message: "First Jenkinsfile"
19 First time build. Skipping changelog.

20 #!/bin/bash -xe echo "Execution user:" `logname` echo "Job name: geladm-PSGEL2xx-devops-applied-to-sas-viya-202yz" echo "Workspace name: /tmp/workspace/geladm-PSGEL2xx-devops-applied-to-s..." -- Shell Script <1s
21 ++ logname
22 + echo "Execution user: cloud-user"
23 Execution user: cloud-user
24 + echo "Job name: geladm-PSGEL2xx-devops-applied-to-sas-viya-202yz"
25 Job name: geladm-PSGEL2xx-devops-applied-to-sas-viya-202yz
26 + echo "workspace name: /tmp/workspace/geladm-PSGEL2xx-devops-applied-to-sas-viya-202yz"
27 workspace name: /tmp/workspace/geladm-PSGEL2xx-devops-applied-to-sas-viya-202yz

28 First Pipeline! -- Print Message <1s
29 First Pipeline!
```

```
pipeline {
  agent { label 'sasnode01' }
  stages {
    stage('Environment variables') {
      steps {
        sh """
        #!/bin/bash -xe
        echo "Execution user: "`logname`
        echo "Job name: ${env.JOB_NAME}"
        echo "Workspace name: ${env.WORKSPACE}"
        """
      }
    }
  }
  post {
    always {
      echo 'First Pipeline!'
    }
  }
}
```

Summary: Devops with SAS Viya, git and jenkins

devops practices – through Git and Jenkins

Git

- Configure git in SAS Studio
- Get your code into git – learn to trust git

Jenkins

- freestyle and pipelines
- Integrate with gitlab and SAS Viya
- Run programs and flows from Jenkins
- CI / CD with automated tests