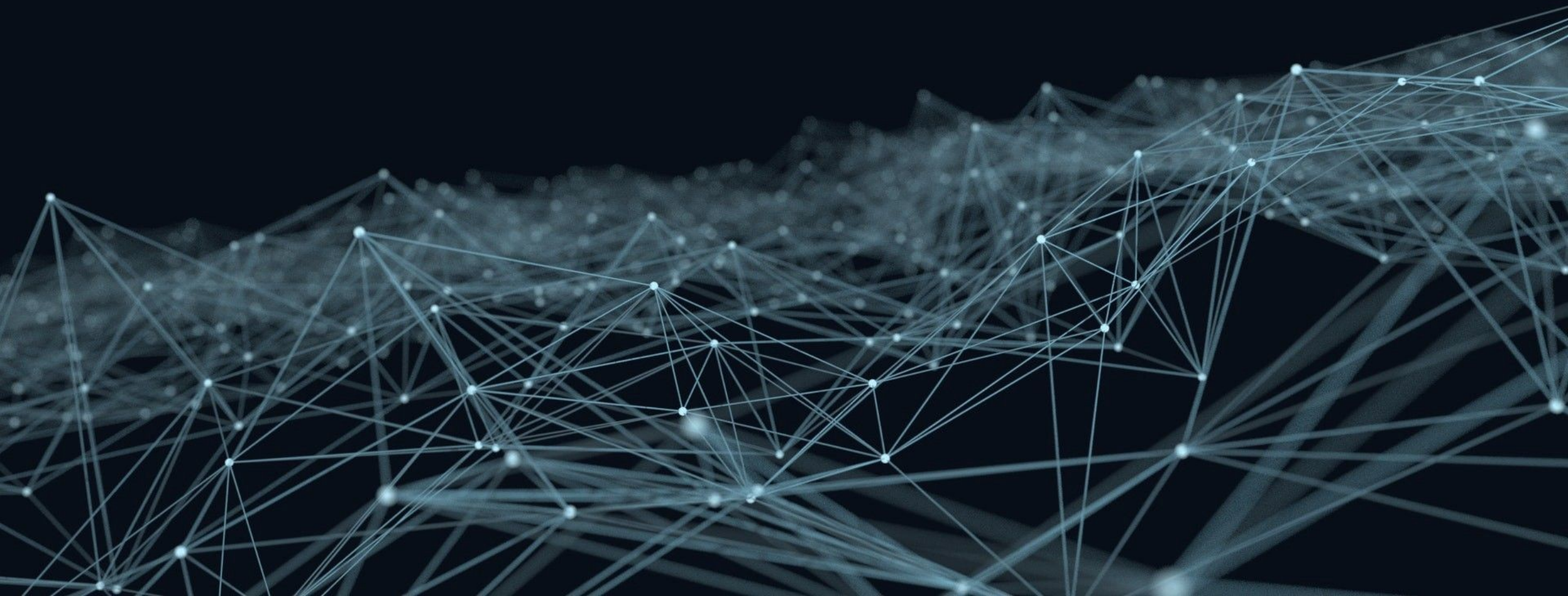
# Introduction to Jenkins

## Module 7: Remote API and Automation

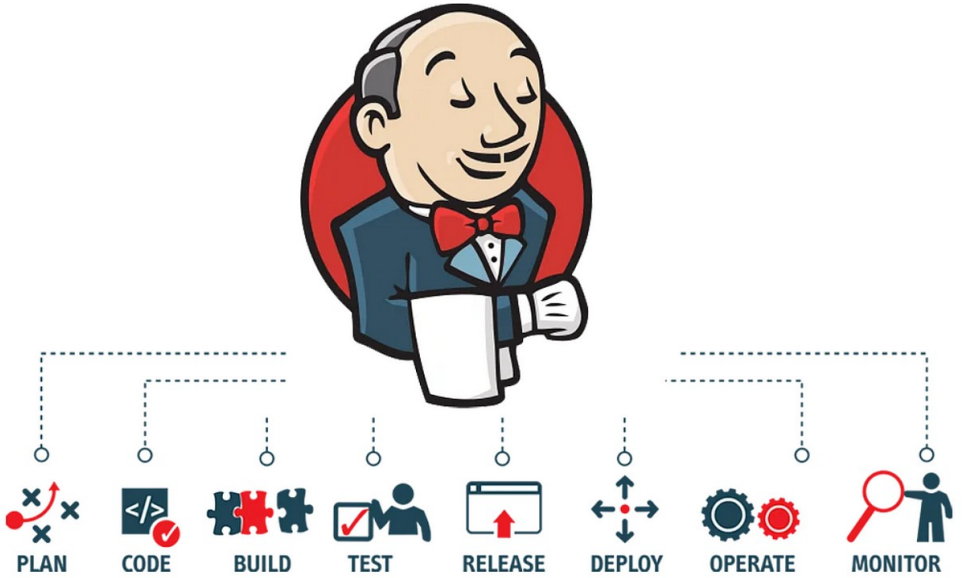


# Topics

- REST API overview and authentication
- Triggering builds from scripts
- Querying jobs and logs
- Python and curl examples

# Jenkins Remote API

- Jenkins exposes a REST-style HTTP API
  - Allows external systems and scripts to interact with Jenkins programmatically.
- Using the API, clients can
  - Trigger builds
  - Check job and build status
  - Retrieve console output
  - Integrate Jenkins with other tools and systems
- Almost every object in Jenkins
  - Has a URL
  - Can return structured data (JSON or XML)
  - Adding /api/json to a Jenkins URL typically returns JSON data.



# API URL Structure

- Using the class lab URL
  - Jenkins root
    - *http://localhost:8080/api/json*
  - Returns metadata about the Jenkins server, such as:
    - *Jenkins version*
    - *Server description*
    - *Executor information*
    - *List of top-level jobs*
    - *Primary view*
    - *System mode (NORMAL / EXCLUSIVE)*
    - *URLs to other API endpoints*
  - This is read-only information
    - *Used by dashboards, scripts, and discovery tools.*


```
{
  "_class": "hudson.model.Hudson",
  "mode": "NORMAL",
  "nodeDescription": "the master Jenkins node",
  "numExecutors": 2,
  "description": null,
  "jobs": [
    {
      "_class": "org.jenkinsci.plugins.workflow.job.WorkflowJob",
      "name": "build-app",
      "url": "http://localhost:8080/job/build-app/"
    },
    {
      "_class": "org.jenkinsci.plugins.workflow.job.WorkflowJob",
      "name": "test-app",
      "url": "http://localhost:8080/job/test-app/"
    }
  ],
  "primaryView": {
    "name": "all",
    "url": "http://localhost:8080/"
  },
  "url": "http://localhost:8080/"
}
```


# Job API URL Structure

- Using the class lab url
  - Job metadata:
    - *http://localhost:8080/job/<job-name>/api/json*
  - Returns metadata about a job
    - *Job identity (name, URL, type)*
    - *Whether the job is enabled or buildable*
    - *Overall job status (based on last run)*
    - *Build history (list of builds)*
    - *Last build*
    - *Last successful build*
    - *Last failed build*
    - *Next build number*
    - *Queue status*


```
{
  "_class": "org.jenkinsci.plugins.workflow.job.WorkflowJob",
  "name": "build-app",
  "displayName": "build-app",
  "url": "http://localhost:8080/job/build-app/",
  "description": "Builds and tests the application",
  "buildable": true,
  "inQueue": false,
  "color": "blue",
  "lastBuild": {
    "number": 42,
    "url": "http://localhost:8080/job/build-app/42/"
  },
  "lastSuccessfulBuild": {
    "number": 41,
    "url": "http://localhost:8080/job/build-app/41/"
  },
  "nextBuildNumber": 43,
  "builds": [
    {
      "number": 42,
      "url": "http://localhost:8080/job/build-app/42/"
    },
    {
      "number": 41,
      "url": "http://localhost:8080/job/build-app/41/"
    }
  ]
}
```

# Build API URL Structure

- Using the class lab url
  - Build metadata:
    - *http://localhost:8080/job/<job-name>/<build-number>/api/json*
  - Returns metadata about a build
    - *Build number and ID*
    - *Build result (SUCCESS, FAILURE, etc.)*
    - *Whether the build is currently running*
    - *Start time and duration*
    - *Agent used*
    - *Build parameters*
    - *Source control changes*


```
{
  "_class": "org.jenkinsci.plugins.workflow.job.WorkflowRun",
  "id": "42",
  "number": 42,
  "result": "SUCCESS",
  "building": false,
  "timestamp": 1707000000000,
  "duration": 15324,
  "estimatedDuration": 16000,
  "url": "http://localhost:8080/job/build-app/42/",
  "executor": null,
  "builtOn": "",
  "changeSet": {},
  "actions": []
}
```


# Triggering a Build

- The POST command is used to trigger a build for a job
  - This requires the use of an API token for authentication
  - Returns a “201 Created” on success
  - The “build” endpoint is used for builds without parameters
  - The example below triggers a new build for
    - *Job “Demo1”*
    - *User “Rod”*
    - *API Token “11fbc4d3b17ad053c3276ff199f25ff773”*

```
curl -X POST \
  -u username:API_TOKEN \
  http://localhost:8080/job/Demo1/build
```

curl -X POST -u Rod:11fbc4d3b17ad053c3276ff199f25ff773  http://localhost:8080/job/Demo1/build

# Triggering a Build

- If the build is parameterized, these are passed in the body as shown in the example on the right
  - Note that the API endpoint is “buildWithParameters”

```
curl -X POST \
  -u Rod:API_TOKEN \
  http://localhost:8080/job/Demo1/buildWithParameters \
  --data ENV=dev \
  --data VERSION=1.2.0
```

# Last Build Shortcut

- Get the most recent build without knowing the build number
  - GET /job/<job-name>/lastBuild/api/json

```
curl -u Rod:API_TOKEN \
  http://localhost:8080/job/Demo1/lastBuild/api/json
```

# Console Logs

- Retrieve raw console output for a build.
  - GET /job/<job-name>/<build-number>/consoleText

http://localhost:8080/job/Demo1/15/consoleText

```
Started by user Rod
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /home/rod/.jenkins/workspace/Demo1
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Hello)
[Pipeline] echo
Hello World
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

# Questions