

ბიზნესისა და ტექნოლოგიების უნივერსიტეტი

# ვერსიონირება და უწყვეტი ინტეგრაცია

თემა მეექვსე და მეშვიდე:  
**Jenkins** ინსტალაცია, **Plugin-ები** და  
**Dashboard-ი**

## Jenkins

Jenkins-ი არის open source, Java-ზე დაფუძნებული ავტომატიზაციის სერვერი, რომელიც ფართოდ გამოიყენება პროგრამული უზრუნველყოფის განვითარების პროცესში, კერძოდ (CI/CD) უზრუნველსაყოფად. მისი მიზანია პროგრამული კოდის ცვლილებების ავტომატური აგება (build), ტესტირება და განთავსება.

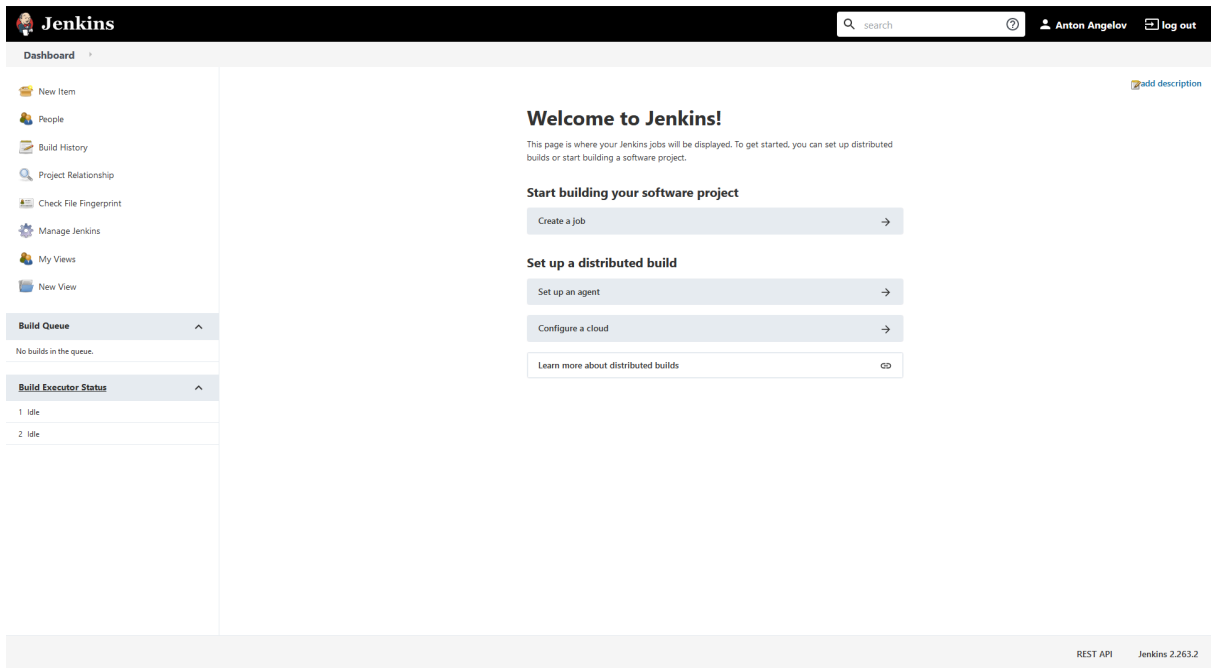
Jenkins-ს აქვს მდიდარი პლაგინების სისტემა, რაც საშუალებას იძლევა მარტივად ინტეგრირდეს სხვა ინსტრუმენტებთან, როგორებიცაა Git, Docker, Kubernetes, Maven, Puppet და სხვა. Jenkins-ის მეშვეობით შესაძლებელია პროექტების ავტომატური Build-ი სხვადასხვა ენაზე (Java, Python, Node.js და სხვა), ტესტების ავტომატური გაშვება, არტეფაქტების შენახვა და განლაგება, პროცესების მონიტორინგი და რეპორტების გენერირება. გარდა ამისა, Jenkins-ს აქვს მოქნილი Pipeline მხარდაჭერა, რომელიც საშუალებას გაძლევთ აღწეროთ სრული CI/CD პროცესის ეტაპები კოდით (Jenkinsfile-ის მეშვეობით), რაც აუმჯობესებს გამჭვირვალობას და ვერსიების მართვას.

იმისათვის რომ დააყენოთ Jenkins-ი Ubuntu-ზე :

- **sudo wget -O /etc/apt/keyrings/jenkins-keyring.asc  
https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key**
- **echo "deb [signed-by=/etc/apt/keyrings/jenkins-keyring.asc]  
https://pkg.jenkins.io/debian-stable binary/ | sudo tee  
/etc/apt/sources.list.d/jenkins.list > /dev/null**
- **sudo apt-get update**
- **sudo apt-get install jenkins**
- **sudo systemctl start jenkins**
- **sudo systemctl enable jenkins**

ამ ბრძანებების შემდეგ ბრაუზერში უნდა ჩანდეს <http://localhost:8080> რის შემდეგაც გაიხსნება jenkins-ის ავტორიზაციის ფანჯარა. ნაგულისხმევი მნიშვნელობით შექმნილია მხოლოდ admin მომხმარებელი რომლითაც უნდა შეხვიდეთ jenkins-ზე, admin მომხმარებლის პაროლი შეგიძლიათ ნახოთ **/var/lib/jenkins/secrets/initialAdminPassword**

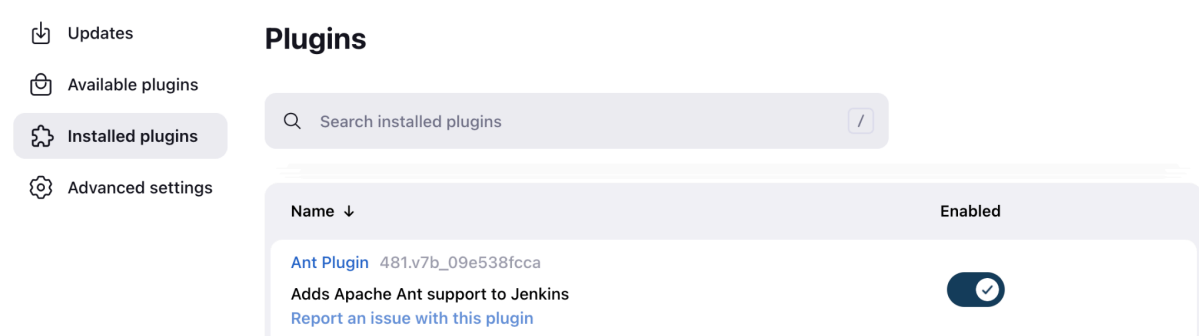
დალოგინების შემდეგ Jenkins-ი შემოგთავაზებთ დაყენების ტიპის არჩევას: დაყენება ნაგულისხმევი პლაგინებით ან არჩევით. შემდეგ ახალი მომხმარებლის დამატების ფანჯარა გაიხსნება რომლის გამოტოვებაც იქნება შესაძლებელი თუმცა admin მომხმარებლით მუშაობა არ არის რეკომენდირებული.



Jenkins-ზე შესვლის შემდეგ, მუშაობის დაწყებამდე საჭირო იქნება: მომხმარებლების დამატება, ნვდომების შეზღუდვა და უსაფრთხოების უზრუნველყოფა რისი გაკეთებაც შეგიძლიათ **Manage jenkins** გრაფიდან.

## Jenkins Plugin-ები

Jenkins-ის ერთ-ერთი უპირატესობა, სხვა CI/CD ხელსაწყოებთან შედარებით არის **plugin-ები**. ამჟამად 2000 plugin-ზე მეტი არსებობს რომლის გამოყენებაც ყველა მსურველს შეუძლია. Plugin-ების მოძებნა, მათი დეტალური გარჩევა, ვერსიების და დამოკიდებულებების ნახვა შესაძლებელია <https://plugins.jenkins.io/> ოფიციალური საიტიდან. Plugin-ის არჩევისას ყურადღება უნდა გაამახვილოთ იმას თუ რამდენი მომხმარებელი იყენებს ამა თუ იმ plugin-ს.



რიგ შემთხვევაში საჭიროა კონკრეტული plugin-ის ძველი ვერსიის დაყენება რისი საშუალებაც უშუალოდ jenkins-ის dashboard-იდან არ არის შესაძლებელი, ამისთვის საჭიროა plugin-ის ძველი ვერსიის გადმოწერა ზემოთ ნახსენები საიტიდან და მისი ატვირთვა jenkins-ზე advanced გრაფიდან. გასათვალისწინებელია რომ, jenkins-ის plugin-ების უმეტესობას აქვს სხვადასხვა დამოკიდებულებები რომლებიც plugin-თან ერთად ყენდება, თუმცა plugin-ის ნაშლისას დამოკიდებულებები არ იშლება. Jenkins-ის პლაგინური არქიტექტურა საშუალებას გაძლევთ შექმნათ პერსონალიზებული და ძლიერ ავტომატიზებული სისტემებიაც.

Jenkins-ის მთავარ გვერდზე თქვენ ნახავთ Job-ების შექმნის გრაფას ან თვითონ ჯობებს, Jenkins-ის ჯობი არის სამუშაო ერთეული, რომელიც განსაზღვრავს მოქმედებათა მიმდევრობას. Jenkins ჯობებს ხშირად პროექტებსაც უწოდებენ. თითოეული ჯობი შეიძლება იყოს დამოუკიდებელი ან დამოკიდებული, გამოიყენოს სხვა ჯობები, პარამეტრები, აგენტები და სკრიპტები.

Jenkins-ის შემოთავაზებულ პლაგინებთან ერთად დაყენების შემთხვევაში, რამოდენიმე ტიპის Job-ის შექმნა იქნება შესაძლებელი:

- **Freestyle Project** - მარტივი და კლასიკური ჯობია, რომელიც უზრუნველყოფს ვიზუალურ ინტერფეისს build-ისთვის.
- **Pipeline Job** - საშუალებას გაძლევთ დაწეროთ build პროცესი როგორც კოდი Groovy DSL-ის მეშვეობით.
- **Multibranch Pipeline** - ავტომატურად ქმნის pipeline-ებს თითოეული Git ბრანჩისთვის, რაც ამარტივებს მრავალბრენჩიან პროექტებზე მუშაობას.
- **Folder** - გამოიყენება ჯობების ორგანიზებისთვის ჯგუფებად, რათა უკეთ მართოთ დიდი რაოდენობის პროექტები.
- **Matrix Project** - გაუშვებს ჯობს სხვადასხვა პარამეტრის კომბინაციით (მაგალითად: JDK-ს სხვადასხვა ვერსიებით, ან სხვადასხვა ოპერაციული სისტემებით).