

INFOSOFT IT SOLUTIONS

Training | Projects | Placements

Revathi Apartments, Ameerpet, 1st Floor, Opposite Annapurna Block, Infosoft It solutions,
Software Training & Development Institute, +91-9059683947|+91-9182540872

Bit Bucket Course Content

Introduction to Bitbucket

1. Overview of version control
 - Definition and importance of version control
 - Historical evolution of version control systems
 - Types of version control systems
 - Role of version control in software development
 - Future trends in version control technology
2. Benefits of using Bitbucket
 - Collaboration features for teams
 - Security features and compliance with standards
 - Integration capabilities with other tools
 - Scalability and reliability of Bitbucket
3. Bitbucket vs other VCS platforms
 - Comparison with GitHub and GitLab
 - Unique features of Bitbucket
 - Performance metrics and user satisfaction

Bitbucket Installation

1. System requirements
 - Hardware requirements for different setups
 - Software prerequisites
 - Network and connectivity needs
 - Scalability considerations
2. Step-by-step installation guide
 - Downloading the installation package

- Installation process walkthrough
 - Common installation errors and solutions
 - Initial configuration settings
 - Verification and troubleshooting tips
3. Initial setup and configuration
 - Setting up user accounts
 - Configuring basic repository settings
 - Security settings and best practices

Navigation of Bitbucket

1. User interface elements
 - Main dashboard components
 - Repository view and features
 - Pull request and issue tracking interfaces
 - Administrative tools and settings
 - Customization options
2. Dashboard features
 - Real-time activity streams
 - Notification and alert systems
 - Repository health indicators
 - User and team management tools
 - Analytics and reporting capabilities
3. Common user tasks
 - Creating and managing repositories
 - Issuing and merging pull requests
 - Managing branches and tags

Introduction to Git

1. Git basics and terminology
 - Fundamental Git commands
 - Understanding commits, branches, and merges
2. The importance of Git in modern VCS
 - Git's impact on distributed development
 - Collaboration enhancements due to Git

- Git in the context of DevOps practices
3. Differences between Git and Mercurial
 - Technical differences in performance and design
 - Tooling and integration options
 - Choosing between Git and Mercurial for projects

Git installation

1. Pre-installation requirements
 - Assessing current system setup
 - Backup and data protection measures
 - Compatibility checks with other software
2. Installing Git on various operating systems
 - Detailed steps for Windows
 - Installation guide for macOS
 - Commands and procedures for Linux distributions
3. Verifying the installation
 - Checking Git version and configuration settings
 - Running basic Git commands to ensure functionality
 - Troubleshooting common post-installation issues
 - Setting up initial user identity (name and email)
 - Configuring default branch and remote settings

Git Commands

1. Basic commands (add, commit, push, pull)
 - Usage and syntax for each command
 - Best practices for committing changes
 - Managing remote repositories with push and pull
2. Branching and merging commands
 - Creating, deleting, and listing branches
 - Merging strategies and resolving conflicts
 - Use of rebase vs. merge
 - Tips for effective branch management
3. Advanced commands for data analysis
 - Using Git log for historical analysis

- Leveraging Git blame for code authorship
- Finding and recovering from mistakes with Git reflog
- Scripting and automation with Git hooks

Git with Bitbucket

1. Integrating Git repositories with Bitbucket
 - Linking local repositories to Bitbucket
 - Synchronization best practices
 - Using webhooks for automation
 - Troubleshooting connectivity issues
2. Managing branches on Bitbucket
 - Online branch creation and deletion
 - Branch comparison and pull requests
 - Setting branch restrictions
 - Strategies for effective online branch management
3. Resolving merge conflicts via Bitbucket
 - Tools and UI features for conflict resolution
 - Step-by-step guide to resolving conflicts
 - Preventive measures to minimize conflicts

Repository Creation

1. Steps to create new repositories
 - Repository setup wizard in Bitbucket
 - Choosing between private and public options
 - Initial commit and push
 - Cloning repositories to local machines
 - Importing existing repositories into Bitbucket
2. Configuring repository settings
 - Managing repository details (name, description)
 - Setting repository privacy and access controls
 - Configuring webhooks and services
 - Managing repository forks and mirrors

3. Best practices for initial setup
 - Defining a clear repository structure
 - Documenting repository guidelines
 - Setting up issue tracking and project boards

Repository permissions

1. Setting up user roles and permissions
 - Defining roles for team members
2. Managing team access
 - Controlling access to specific branches
 - Setting up project-based access controls
 - Using Bitbucket teams for organized access management
3. Best practices for secure repository management
 - Establishing a security protocol for repositories
 - Using encryption and security scans
 - Maintaining compliance with industry security standards
 - Educating team members on security best practices
 - Routine security reviews and updates

Branches

1. Creating and managing branches
 - Step-by-step guide to branch creation
 - Daily management of branches
 - Strategies for temporary vs. long-term branches
 - Pruning and cleaning up old branches
 - Automation tools for branch management
2. Strategies for branch naming and maintenance
 - Standardized naming conventions
 - Branch lifecycle management
 - Best practices for branch merging schedules
 - Tools and scripts for branch maintenance
 - Managing branches in a team environment
3. Merging and its best practices
 - Merge strategies (squash, rebase, merge comm

Branch Permission

1. Configuring branch-level permissions
 - Setting up permissions for protecting branches
 - Tools for automating permission settings
 - Special permissions for release and hotfix branches
 - Emergency access procedures
2. Protecting branches from unauthorized changes
 - Implementing lock-down features
 - Using branch protection rules
 - Ensuring compliance with change management policies
 - Auditing changes for security compliance
 - Setting up notifications for unauthorized access attempts
3. Role-based access controls
 - Detailed role definitions
 - Custom roles for special project needs

Pull Requests

1. Creating, reviewing, and managing pull requests
 - Creating pull requests from branches
 - Guidelines for effective pull request reviews
 - Managing pull request workflows
 - Tools for automating pull request handling
 - Closing and archiving pull requests
2. Best practices for code reviews
 - Structured code review processes
 - Using checklists and templates for reviews
 - Engaging peers in collaborative reviews
3. Automation with pull requests
 - Integrating CI/CD with pull requests
 - Automatic testing and linting setups
 - Setting up rules for auto-merging
 - Using bots for routine tasks within pull requests
 - Notifications and alerts for pull request events

Merging Files

1. Techniques for efficient file merging
 - Merging strategies for different file types
 - Using graphical tools for visual merging
 - Best practices for handling large files
 - Conflict resolution techniques
 - Preserving file integrity during merges
2. Handling merge conflicts
 - Identifying potential conflicts early
 - Step-by-step conflict resolution process
 - Tools and scripts for conflict detection
 - Strategies for avoiding common conflicts
3. Tools and strategies to aid merging
 - Popular merging tools and their features
 - Custom tool integration for specific projects

Repository Clone

1. Cloning repositories for backup and collaboration
 - Understanding the importance of cloning in Git
 - Cloning vs. forking: choosing the right strategy
 - Using clones for backup purposes
2. Best practices for cloning
 - Setting up the environment before cloning
 - Selecting the correct branch to clone
 - Cloning with minimal data for specific tasks
 - Synchronizing clones with the main repository
 - Handling large repositories during cloning
3. Troubleshooting common issues
 - Resolving errors during cloning
 - Managing slow clone operations
 - Dealing with incomplete or corrupt clones
 - Updating clones with changes from the main repository

Git Bash

1. Introduction to Git bash

- Exploring the Git bash interface
- Differences between Git bash and other terminals
- Customizing the Git bash experience
- Key commands for everyday use
- Integrating Git bash with other tools

2. Essential Git bash commands

- Navigating directories and managing files
- Using Git commands in the bash environment

3. Customizing the Git bash environment

- Configuring Git bash settings
- Adding plugins and tools to enhance functionality
- Setting up color schemes and fonts for better readability

Source Tree

1. Overview of Source tree interface

- Navigating the Source tree dashboard
- Understanding the various panels and windows
- Customizing the layout to suit user needs
- Key features and tools available in Source tree
- Integrating Source tree with other Atlassian products

2. Integrating Source tree with Bitbucket

- Connecting Source tree to Bitbucket repositories
- Managing branches and commits from Source tree
- Utilizing pull request features within Source tree

3. Advanced features of Source tree

- Using advanced search and file tracking options
- Applying patches and interactive rebasing
- Handling submodules and large files
- Automating tasks with Source tree actions
- Using Source tree for code review and feedback

Installation of Source Tree

1. System requirements

- Hardware and software prerequisites for Source tree
 - Compatibility with different operating systems
 - Considerations for multi-user setups
 - Preparing your system for optimal performance
2. Step-by-step installation process
 - Downloading and installing Source tree
 - Initial configuration and setup
 - Verifying installation success
 3. Initial configuration for optimal use
 - Configuring general preferences
 - Setting up repository defaults
 - Customizing user interface options
 - Integrating with other tools and services
 - Ensuring security and privacy settings are adequate

User Management

1. Adding, removing, and managing users
 - Defining user roles within Bitbucket
 - Adding new users to the system
 - Removing users and handling data retention
 - Managing user groups and access levels
 - Auditing user activity for compliance and security
2. Role definitions and permissions
 - Creating custom roles for different team functions
 - Assigning permissions based on project requirements
 - Updating roles as project dynamics change
3. Integrating with corporate directories
 - Connecting Bitbucket with LDAP and Active Directory
 - Automating user management through directory services
 - Managing group memberships and roles via directories
 - Troubleshooting common integration issues

Integration with Jira

1. Configuring Jira integration

- Setting up the connection between Bitbucket and Jira
- Mapping repositories to Jira projects
- Configuring automatic issue transitions based on Git actions
- Enhancing traceability between code commits and Jira issues
- Maintaining synchronization between Jira and Bitbucket

2. Tracking issues and tasks

- Using Jira for project management and issue tracking
- Linking commits and branches to Jira issues
- Creating and managing Jira issues directly from Bitbucket
- Best practices for managing project workflows through Jira

3. Enhancing project management with Jira and Bitbucket

- Collaborative features that enhance project transparency
- Reporting tools for performance and progress tracking
- Automating workflow transitions for streamlined operations
- Leveraging Jira queries and dashboards for project insights