


Course Name	AZ-400: Designing and Implementing Microsoft DevOps Solutions	
About the Course	This course teaches students how to design and implement DevOps processes and practice. This course will teach implementing DevOps practices like CI/CD, infrastructure as code, and automation so that you can manage and implement DevOps services and tools effectively	
Key Skills You Will Learn	Plan and practice DevOps using Azure Boards, Manage builds and packages using Azure Artifacts and containers, Design a license management strategy, Design a strategy for integrating on-premises and cloud resources, Build a secure and quality DevOps setup, Understand site reliability engineering	
Course Pre-Requisite	You should be familiar with Azure administration and development, and have passed either the AZ-104 Azure Administrator Associate or AZ-204 Azure Developer Associate exam	
Target Audience	IT professionals: Aspiring Azure DevOps Engineer Experts, or those preparing for the AZ-400 exam, System administrators, developers, and IT professionals: Wanting to learn how to build DevOps solutions on Microsoft Azure, Professionals with a background in software engineering: With a basic understanding of DevOps, Professionals interested in roles like DevOps engineers, release managers, or automation architect	
Job prospects with this role	DevOps Engineer, Software Engineer, Senior DevOps Engineer, Site Reliability Engineer, Software Release Manager.	
Course Duration	~ 32 Hrs	
Course Customisation	Not applicable	
Certification	READYBELL AZ-400: Designing and Implementing Microsoft DevOps Solutions Certificate	
Mode of Training	Instructor-led 100% Online or 100% Classroom (Salt Lake, Kolkata - India) or hybrid mode (Online + Classroom) as suitable for the learner	
Course Fees	Please contact us	
Refund Policy	Get a 3-hours free trial during which you can cancel at no penalty. After that, we don't give refunds	
Job Assistance	Will assist candidate in securing a suitable job	
Contact	READYBELL SOFTWARE SERVICES PVT. LIMITED AH 12, SALT LAKE SECTOR 2, KOLKATA (INDIA) - 700 091 E-MAIL: contact@readybellssoftware.com PH: +91 - 9147708045/9674552097, +91 - 33-79642872	 Software Services Pvt. Ltd.

CURRICULUM		
Topic	Sub-Topic	Duration (Hrs)
AZ-400: Designing and Implementing Microsoft DevOps Solutions	AZ-400: Development for enterprise DevOps	32 Hrs
	Module 1: Introduction to DevOps	
	Introduction	
	What is DevOps?	
	Explore the DevOps journey	
	Identify transformation teams	
	Define organization structure for agile practices	
	Explore shared goals and define timelines	
	What is Azure DevOps?	
	What is GitHub?	
	Design a license management strategy	
	What is source control?	
	Describe working with Git locally	
	Introduction to Azure Repos	
	Introduction to GitHub	
	Module 2: Plan Agile with GitHub Projects and Azure Boards	
	Introduction	
	Introduction to GitHub Projects and Project boards	
	Introduction to Azure Boards	
	Configure projects and teams in Azure DevOps	
	Link GitHub to Azure Boards	
	Configure GitHub Projects	
	Manage work with GitHub Project boards	
	Customize Project views	
	Collaborate using team discussions	
	Design and implement a strategy for feedback cycles	
	Design and implement source, bug, and quality traceability	
	Agile Plan and Portfolio Management with Azure Boards	
	Module 3: Design and implement branch strategies and workflows	
	Introduction	
	Explore branch workflow types	
	Explore feature branch workflow	
	Explore Git branch model for continuous delivery	
Explore GitHub flow		
Explore fork workflow		
Implement branch merging restrictions		
Version Control with Git in Azure Repos		

Module 4: Collaborate with pull requests in Azure Repos
Introduction
Collaborate with pull requests
Exercise Azure Repos collaborating with pull requests
Module 5: Explore Git hooks
Introduction
Introduction to Git hooks
Implement Git hooks
Module 6: Plan foster inner source
Introduction
Explore foster inner source
Implement the fork workflow
Describe inner source with forks
Module 7: Manage and configure repositories
Introduction
Work with large repositories
Explore monorepo versus multiple repos
Implement a change log
Implement Scalar and cross repo when cross repo sharing
Recover specific data by using Git commands
Purge repository data
Manage releases with GitHub Repos
Automate release notes with GitHub
Create API documentation
Implement automation of Git history documentation
Configure source control repository permissions using GitHub
Configure GitHub tags to organize repositories
Module 8: Identify technical debt
Introduction
Examine code quality
Examine complexity and quality metrics
Introduction to technical debt
Measure and manage technical debt
Introduction to GitHub Advanced Security
Integrate other code quality tools
Plan effective code reviews
AZ-400: Implement CI with Azure Pipelines and GitHub Actions
Module 9: Explore Azure Pipelines
Introduction
Explore the concept of pipelines in DevOps

	Describe Azure Pipelines	
	Understand Azure Pipelines key terms	
	Module 10: Manage Azure Pipeline agents and pools	
	Introduction	
	Choose between Microsoft-hosted versus self-hosted agents	
	Explore job types	
	Introduction to agent pools	
	Explore predefined agent pool	
	Understand typical situations for agent pools	
	Communicate with Azure Pipelines	
	Communicate to deploy to target servers	
	Examine other considerations	
	Describe security of agent pools	
	Configure agent pools and understand pipeline styles	
	Module 11: Describe pipelines and concurrency	
	Introduction	
	Understand parallel jobs	
	Estimate parallel jobs	
	Describe Azure Pipelines and open-source projects	
	Explore Azure Pipelines and Visual Designer	
	Describe Azure Pipelines and YAML	
	Enable Continuous Integration with Azure Pipelines	
	Module 12: Design and implement a pipeline strategy	
	Introduction	
	Configure agent demands	
	Explore multi-configuration and multi-agent	
	Integrate GitHub repos and Azure Pipelines	
	Design and implement a comprehensive testing strategy	
	Implement code coverage and show in the pipeline	
	Implement multi-job builds	
	Explore source control types supported by Azure Pipelines	
	Module 13: Integrate with Azure Pipelines	
	Introduction	
	Describe the anatomy of a pipeline	
	Understand the pipeline structure	
	Detail templates	
	Explore YAML resources	
	Use multiple repositories in your pipeline	
	Migrate a pipeline from classic to YAML in Azure Pipelines	

	Module 14: Introduction to GitHub Actions	
	Introduction	
	What are actions?	
	Explore Actions flow	
	Understand Workflows	
	Describe standard workflow syntax elements	
	Explore Events	
	Explore Jobs	
	Explore Runners	
	Examine release and test an action	
	Module 15: Learn continuous integration with GitHub Actions	
	Introduction	
	Describe continuous integration with actions	
	Examine environment variables	
	Share artifacts between jobs	
	Examine Workflow badges	
	Describe best practices for creating actions	
	Mark releases with Git tags	
	Create encrypted secrets	
	Use secrets in a workflow	
	Implement GitHub Actions for CI/CD	
	Module 16: Design a container build strategy	
	Introduction	
	Examine structure of containers	
	Work with Docker containers	
	Understand Dockerfile core concepts	
	Examine multi-stage dockerfiles	
	Examine considerations for multiple stage builds	
	Explore Azure container-related services	
	Deploy Docker containers to Azure App Service web apps	
	AZ-400: Design and implement a release strategy	
	Module 17: Create a release pipeline	
	Introduction	
	Describe Azure DevOps release pipeline capabilities	
	Explore release pipelines	
	Explore artifact sources	
	Choose the appropriate artifact source	
	Examine considerations for deployment to stages	
	Explore build and release tasks	
	Explore custom build and release tasks	

	Explore release jobs	
	Understand database deployment task	
	Configure Pipelines as Code with YAML	
	Module 18: Explore release recommendations	
	Introduction	
	Understand the delivery cadence and three types of triggers	
	Explore release approvals	
	Explore release gates	
	Use release gates to protect quality	
	Explore GitOps release strategy and recommendations	
	Control Deployments using Release Gates	
	Module 19: Provision and test environments	
	Introduction	
	Provision and configure target environments	
	Exercise - set up service connections	
	Configure automated integration and functional test automation	
	Understand Shift-left	
	Set up and run availability tests	
	Explore Azure Load Testing	
	Set up and run functional tests	
	Module 20: Manage and modularize tasks and templates	
	Introduction	
	Examine task groups	
	Explore variables in release pipelines	
	Exercise - create and manage variable groups	
	Module 21: Automate inspection of health	
	Introduction	
	Automate inspection of health	
	Explore events and notifications	
	Explore service hooks	
	Configure Azure DevOps notifications	
	Configure GitHub notifications	
	Explore how to measure quality of your release process	
	Examine release notes and documentation	
	Examine considerations for choosing release management tools	
	Explore common release management tools	
	AZ-400: Implement a secure continuous deployment using Azure Pipelines	
	Module 22: Introduction to deployment patterns	
	Introduction	

	Explore microservices architecture	
	Examine classical deployment patterns	
	Understand modern deployment patterns	
	Module 23: Implement blue-green deployment and feature toggles	
	Introduction	
	What is blue-green deployment?	
	Explore deployment slots	
	Introduction to feature toggles	
	Describe feature toggle maintenance	
	Module 24: Implement canary releases and dark launching	
	Introduction	
	Explore canary releases	
	Examine traffic manager	
	Understand dark launching	
	Module 25: Implement A/B testing and progressive exposure deployment	
	Introduction	
	What is A/B testing?	
	Explore CI-CD with deployment rings	
	Module 26: Integrate with identity management systems	
	Introduction	
	Integrate GitHub with single sign-on (SSO)	
	Design and implement permissions and roles in GitHub	
	Design and implement permissions and security groups in Azure DevOps	
	Explore workload identities	
	Implement managed identities	
	Module 27: Manage application configuration data	
	Introduction	
	Rethink application configuration data	
	Explore separation of concerns	
	Understand external configuration store patterns	
	Implement Azure DevOps secure files	
	Introduction to Azure App Configuration	
	Examine Key-value pairs	
	Examine App configuration feature management	
	Integrate Azure Key Vault with Azure Pipelines	
	Manage secrets, tokens and certificates	
	Examine DevOps inner and outer loop	
	Integrate Azure Key Vault with Azure DevOps	
	Enable Dynamic Configuration and Feature Flags	

AZ-400: Manage infrastructure as code using Azure and DSC
Module 28: Explore infrastructure as code and configuration management
Introduction
Explore environment deployment
Examine environment configuration
Understand imperative versus declarative configuration
Understand idempotent configuration
Module 29: Create Azure resources using Azure Resource Manager templates
Introduction
Why use Azure Resource Manager templates?
Explore template components
Manage dependencies
Modularize templates
Manage secrets in templates
Module 30: Create Azure resources by using Azure CLI
Introduction
What is Azure CLI?
Work with Azure CLI
Module 31: Explore Azure Automation with DevOps
Introduction
Create automation accounts
What is a runbook?
Understand automation shared resources
Explore runbook gallery
Examine webhooks
Explore source control integration
Explore PowerShell workflows
Create a workflow
Explore hybrid management
Examine checkpoint and parallel processing
Module 32: Implement Desired State Configuration (DSC)
Introduction
Understand configuration drift
Explore Desired State Configuration (DSC)
Explore Azure Automation State configuration (DSC)
Examine DSC configuration file
Implement DSC and Linux Automation on Azure
Module 33: Implement Bicep
Introduction
What is Bicep?

	Install Bicep	
	Exercise - Create Bicep templates	
	Understand Bicep file structure and syntax	
	Exercise - Deploy a Bicep file from Azure Pipelines	
	Exercise - Deploy a Bicep file from GitHub workflows	
	Deployments using Azure Bicep templates	
	AZ-400: Design and implement a dependency management strategy	
	Module 34: Explore package dependencies	
	Introduction	
	What is dependency management?	
	Describe elements of a dependency management strategy	
	Identify dependencies	
	Understand source and package componentization	
	Decompose your system	
	Scan your codebase for dependencies	
	Module 35: Understand package management	
	Introduction	
	Explore packages	
	Understand package feeds	
	Explore package feed managers	
	Explore common public package sources	
	Explore self-hosted and SaaS based package sources	
	Consume packages	
	Introduction to Azure Artifacts	
	Publish packages	
	Package management with Azure Artifacts	
	Module 36: Migrate consolidate and secure artifacts	
	Introduction	
	Identify existing artifact repositories	
	Migrate and integrating artifact repositories	
	Secure access to package feeds	
	Examine roles	
	Examine permissions	
	Examine authentication	
	Module 37: Implement a versioning strategy	
	Introduction	
	Understand versioning of artifacts	
	Explore semantic versioning	
	Examine release views	
	Promote packages	

Explore best practices for versioning
Module 38: Introduction to GitHub Packages
Introduction
Publish packages
Install a package
Delete and restore a package
Explore package access control and visibility
AZ-400: Implement continuous feedback
Module 39: Implement tools to track usage and flow
Introduction
Understand the inner loop
Introduction to continuous monitoring
Explore Azure Monitor and Log Analytics
Examine Kusto Query Language (KQL)
Explore Application Insights
Implement Application Insights
Design and implement metrics and queries
Monitor application performance with Application Insights
Module 40: Develop monitor and status dashboards
Introduction
Configure monitoring in GitHub
Explore Azure Dashboards
Explore Azure Monitor workbooks
Explore Power BI
Build your own custom application
Monitor pipeline health, including failure rate, duration, and flaky tests
Optimize a pipeline for cost, time, performance, and reliability
Optimize pipeline concurrency for performance and cost
Module 41: Share knowledge within teams
Introduction
Share acquired knowledge within development teams
Introduction to Azure DevOps project wikis
Integrate GitHub and Azure DevOps with Microsoft Teams
Share team knowledge using Azure Project Wiki
Module 42: Design processes to automate application analytics
Introduction
Explore rapid responses and augmented search
Integrate telemetry
Examine monitoring tools and technologies
Explore IT Service Management Connector

Module 43: Manage alerts, blameless retrospectives and a just culture
Introduction
Examine when to get a notification
Explore how to fix it
Explore smart detection notifications
Improve performance
Understand server response time degradation
Reduce meaningless and non-actionable alerts
Examine blameless retrospective
Develop a just culture
AZ-400: Implement security and validate code bases for compliance
Module 44: Introduction to Secure DevOps
Introduction
Describe SQL injection attack
Understand DevSecOps
Explore Secure DevOps Pipeline
Explore key validation points
Explore continuous security validation
Understand threat modeling
Explore CodeQL in GitHub
Module 45: Implement open-source software
Introduction
Explore how software is built
What is open-source software?
Explore corporate concerns with open-source software components
Introduction to open-source licenses
Explore common open-source licenses
Examine license implications and ratings
Module 46: Software Composition Analysis
Introduction
Inspect and validate code bases for compliance
Explore software composition analysis (SCA)
Implement GitHub Dependabot alerts and security updates
Integrate software composition analysis checks into pipelines
Examine tools for assess package security and license rate
Automate container scanning, including container images
Interpret alerts from scanner tools
Module 47: Security Monitoring and Governance
Introduction
Implement pipeline security

	Explore Microsoft Defender for Cloud	
	Examine Microsoft Defender for Cloud usage scenarios	
	Explore Azure Policy	
	Understand policies	
	Explore initiatives	
	Explore resource locks	
	Understand Microsoft Defender for Identity	
	Integrate GitHub Advanced Security with Microsoft Defender for Cloud	
	Configure GitHub Advanced Security for GitHub and Azure DevOps	
To register for this course please e-mail/call us		