



LPI 701-100

LPI DevOps Tools Engineer Certification Questions & Answers

Exam Summary – Syllabus – Questions

701-100

[LPI DevOps Tools Engineer](#)

60 Questions Exam – 500 / 800 Cut Score – Duration of 90 minutes

Table of Contents:

Know Your 701-100 Certification Well:	2
LPI 701-100 DevOps Tools Engineer Certification Details:	2
701-100 Syllabus:.....	3
Software Engineering	3
Container Management	6
Machine Deployment	9
Configuration Management	11
Service Operations	12
LPI 701-100 Sample Questions:.....	15
Study Guide to Crack LPI DevOps Tools Engineer 701- 100 Exam:	18

Know Your 701-100 Certification Well:

The 701-100 is best suitable for candidates who want to gain knowledge in the LPI Open Technology. Before you start your 701-100 preparation you may struggle to get all the crucial DevOps Tools Engineer materials like 701-100 syllabus, sample questions, study guide.

But don't worry the 701-100 PDF is here to help you prepare in a stress free manner.

The PDF is a combination of all your queries like-

- What is in the 701-100 syllabus?
- How many questions are there in the 701-100 exam?
- Which Practice test would help me to pass the 701-100 exam at the first attempt?

Passing the 701-100 exam makes you LPI DevOps Tools Engineer. Having the DevOps Tools Engineer certification opens multiple opportunities for you. You can grab a new job, get a higher salary or simply get recognition within your current organization.

LPI 701-100 DevOps Tools Engineer Certification

Details:

Exam Name	LPI DevOps Tools Engineer
Exam Code	701-100
Exam Price	\$200 (USD)
Duration	90 mins
Number of Questions	60
Passing Score	500 / 800
Schedule Exam	Pearson VUE
Sample Questions	LPI DevOps Tools Engineer Sample Questions
Practice Exam	LPI 701-100 Certification Practice Exam

701-100 Syllabus:

Topic	Details
Software Engineering	
Modern Software Development	<p>Weight: 6</p> <p>Description: Candidates should be able to design software solutions suitable for modern runtime environments. Candidates should understand how services handle data persistence, sessions, status information, transactions, concurrency, security, performance, availability, scaling, load balancing, messaging, monitoring and APIs. Furthermore, candidates should understand the implications of agile and DevOps on software development.</p> <p>Key Knowledge Areas:</p> <ul style="list-style-type: none"> • Understand and design service based applications • Understand common API concepts and standards • Understand aspects of data storage, service status and session handling • Design software to be run in containers • Design software to be deployed to cloud services • Awareness of risks in the migration and integration of monolithic legacy software • Understand common application security risks and ways to mitigate them • Understand the concept of agile software development • Understand the concept of DevOps and its implications to software developers and operators <p>The following is a partial list of the used files, terms and utilities:</p> <ul style="list-style-type: none"> • REST, JSON • Service Orientated Architectures (SOA)

Topic	Details
	<ul style="list-style-type: none"> • Microservices • Immutable servers • Loose coupling • Cross site scripting, SQL injections, verbose error reports, API authentication, consistent enforcement of transport encryption • CORS headers and CSRF tokens • ACID properties and CAP theorem
<p>Standard Components and Platforms for Software</p>	<p>Weight: 2</p> <p>Description: Candidates should understand services offered by common cloud platforms. They should be able to include these services in their application architectures and deployment toolchains and understand the required service configurations. OpenStack service components are used as a reference implementation.</p> <p>Key Knowledge Areas:</p> <ul style="list-style-type: none"> • Features and concepts of object storage • Features and concepts of relational and NoSQL databases • Features and concepts of message brokers and message queues • Features and concepts of big data services • Features and concepts of application runtimes / PaaS • Features and concepts of content delivery networks <p>The following is a partial list of the used files, terms and utilities:</p> <ul style="list-style-type: none"> • OpenStack Swift • OpenStack Trove • OpenStack Zaqr • CloudFoundry

Topic	Details
Source Code Management	<ul style="list-style-type: none"> • OpenShift <p>Weight: 5</p> <p>Description: Candidates should be able to use Git to manage and share source code. This includes creating and contributing to a repository as well as the usage of tags, branches and remote repositories. Furthermore, the candidate should be able to merge files and resolve merging conflicts.</p> <p>Key Knowledge Areas:</p> <ul style="list-style-type: none"> • Understand Git concepts and repository structure • Manage files within a Git repository • Manage branches and tags • Work with remote repositories and branches as well as submodules • Merge files and branches • Awareness of SVN and CVS, including concepts of centralized and distributed SCM solutions <p>The following is a partial list of the used files, terms and utilities:</p> <ul style="list-style-type: none"> • git • .gitignore
Continuous Integration and Continuous Delivery	<p>Weight: 5</p> <p>Description: Candidates should understand the principles and components of a continuous integration and continuous delivery pipeline. Candidates should be able to implement a CI/CD pipeline using Jenkins, including triggering the CI/CD pipeline, running unit, integration and acceptance tests, packaging software and handling the deployment of tested software artifacts. This objective covers the feature set of Jenkins version 2.0 or later.</p> <p>Key Knowledge Areas:</p> <ul style="list-style-type: none"> • Understand the concepts of Continuous

Topic	Details
	<p>Integration and Continuous Delivery</p> <ul style="list-style-type: none"> • Understand the components of a CI/CD pipeline, including builds, unit, integration and acceptance tests, artifact management, delivery and deployment • Understand deployment best practices • Understand the architecture and features of Jenkins, including Jenkins Plugins, Jenkins API, notifications and distributed builds • Define and run jobs in Jenkins, including parameter handling • Fingerprinting, artifacts and artifact repositories • Understand how Jenkins models continuous delivery pipelines and implement a declarative continuous delivery pipeline in Jenkins • Awareness of possible authentication and authorization models • Understanding of the Pipeline Plugin • Understand the features of important Jenkins modules such as Copy Artifact Plugin, Fingerprint Plugin, Docker Pipeline, Docker Build and Publish plugin, Git Plugin, Credentials Plugin • Awareness of Artifactory and Nexus <p>The following is a partial list of the used files, terms and utilities:</p> <ul style="list-style-type: none"> • Step, Node, Stage • Jenkins DSL • Jenkinsfile • Declarative Pipeline • Blue-green and canary deployment
Container Management	
Container Usage	<p>Weight: 7</p> <p>Description: Candidates should be able to build, share and operate Docker containers. This includes creating</p>

Topic	Details
	<p>Dockerfiles, using a Docker registry, creating and interacting with containers as well as connecting containers to networks and storage volumes. This objective covers the feature set of Docker version 17.06 or later.</p> <p>Key Knowledge Areas:</p> <ul style="list-style-type: none"> • Understand the Docker architecture • Use existing Docker images from a Docker registry • Create Dockerfiles and build images from Dockerfiles • Upload images to a Docker registry • Operate and access Docker containers • Connect container to Docker networks • Use Docker volumes for shared and persistent container storage <p>The following is a partial list of the used files, terms and utilities:</p> <ul style="list-style-type: none"> • docker • Dockerfile • .dockerignore
<p>Container Deployment and Orchestration</p>	<p>Weight: 5</p> <p>Description: Candidates should be able to run and manage multiple containers that work together to provide a service. This includes the orchestration of Docker containers using Docker Compose in conjunction with an existing Docker Swarm cluster as well as using an existing Kubernetes cluster. This objective covers the feature sets of Docker Compose version 1.14 or later, Docker Swarm included in Docker 17.06 or later and Kubernetes 1.6 or later.</p> <p>Key Knowledge Areas:</p> <ul style="list-style-type: none"> • Understand the application model of Docker

Topic	Details
	<p>Compose</p> <ul style="list-style-type: none"> • Create and run Docker Compose Files (version 3 or later) • Understand the architecture and functionality of Docker Swarm mode • Run containers in a Docker Swarm, including the definition of services, stacks and the usage of secrets • Understand the architecture and application model Kubernetes • Define and manage a container-based application for Kubernetes, including the definition of Deployments, Services, ReplicaSets and Pods <p>The following is a partial list of the used files, terms and utilities:</p> <ul style="list-style-type: none"> • docker-compose • docker • kubectl
Container Infrastructure	<p>Weight: 4</p> <p>Description: Candidates should be able to set up a runtime environment for containers. This includes running containers on a local workstation as well as setting up a dedicated container host. Furthermore, candidates should be aware of other container infrastructures, storage, networking and container specific security aspects. This objective covers the feature set of Docker version 17.06 or later and Docker Machine 0.12 or later.</p> <p>Key Knowledge Areas:</p> <ul style="list-style-type: none"> • Use Docker Machine to setup a Docker host • Understand Docker networking concepts, including overlay networks • Create and manage Docker networks • Understand Docker storage concepts

Topic	Details
	<ul style="list-style-type: none"> • Create and manage Docker volumes • Awareness of Flocker and flannel • Understand the concepts of service discovery • Basic feature knowledge of CoreOS Container Linux, rkt and etcd • Understand security risks of container virtualization and container images and how to mitigate them <p>The following is a partial list of the used files, terms and utilities:</p> <ul style="list-style-type: none"> • docker-machine
<p>Machine Deployment</p>	
<p>Virtual Machine Deployment</p>	<p>Weight: 4</p> <p>Description: Candidates should be able to automate the deployment of a virtual machine with an operating system and a specific set of configuration files and software.</p> <p>Key Knowledge Areas:</p> <ul style="list-style-type: none"> • Understand Vagrant architecture and concepts, including storage and networking • Retrieve and use boxes from Atlas • Create and run Vagrantfiles • Access Vagrant virtual machines • Share and synchronize folder between a Vagrant virtual machine and the host system • Understand Vagrant provisioning, including File, Shell, Ansible and Docker • Understand multi-machine setup <p>The following is a partial list of the used files, terms and utilities:</p> <ul style="list-style-type: none"> • vagrant

Topic	Details
	<ul style="list-style-type: none"> • Vagrantfile
Cloud Deployment	<p>Weight: 2</p> <p>Description: Candidates should be able to configure IaaS cloud instances and adjust them to match their available hardware resources, specifically, disk space and volumes. Additionally, candidates should be able to configure instances to allow secure SSH logins and prepare the instances to be ready for a configuration management tool such as Ansible.</p> <p>Key Knowledge Areas:</p> <ul style="list-style-type: none"> • Understanding the features and concepts of cloud-init, including user-data and initializing and configuring cloud-init • Use cloud-init to create, resize and mount file systems, configure user accounts, including login credentials such as SSH keys and install software packages from the distribution's repository • Understand the features and implications of IaaS clouds and virtualization for a computing instance, such as snapshotting, pausing, cloning and resource limits.
System Image Creation	<p>Weight: 2</p> <p>Description: Candidates should be able to create images for containers, virtual machines and IaaS cloud instances.</p> <p>Key Knowledge Areas:</p> <ul style="list-style-type: none"> • Understand the functionality and features of Packer • Create and maintain template files • Build images from template files using different builders <p>The following is a partial list of the used files, terms and utilities:</p>

Topic	Details
	<ul style="list-style-type: none"> • packer
<p>Configuration Management</p>	
<p>Ansible</p>	<p>Weight: 8</p> <p>Description: Candidates should be able to use Ansible to ensure a target server is in a specific state regarding its configuration and installed software. This objective covers the feature set of Ansible version 2.2 or later.</p> <p>Key Knowledge Areas:</p> <ul style="list-style-type: none"> • Understand the principles of automated system configuration and software installation • Create and maintain inventory files • Understand how Ansible interacts with remote systems • Manage SSH login credentials for Ansible, including using unprivileged login accounts • Create, maintain and run Ansible playbooks, including tasks, handlers, conditionals, loops and registers • Set and use variables • Maintain secrets using Ansible vaults • Write Jinja2 templates, including using common filters, loops and conditionals • Understand and use Ansible roles and install Ansible roles from Ansible Galaxy • Understand and use important Ansible tasks, including file, copy, template, ini_file, lineinfile, patch, replace, user, group, command, shell, service, systemd, cron, apt, debconf, yum, git, and debug • Awareness of dynamic inventory • Awareness of Ansibles features for non-Linux systems • Awareness of Ansible containers

Topic	Details
	<p>The following is a partial list of the used files, terms and utilities:</p> <ul style="list-style-type: none"> • ansible.cfg • ansible-playbook • ansible-vault • ansible-galaxy • ansible-doc
Other Configuration Management Tools	<p>Weight: 2</p> <p>Description: Candidates should understand the main features and principles of important configuration management tools other than Ansible.</p> <p>Key Knowledge Areas:</p> <ul style="list-style-type: none"> • Basic feature and architecture knowledge of Puppet. • Basic feature and architecture knowledge of Chef. <p>The following is a partial list of the used files, terms and utilities:</p> <ul style="list-style-type: none"> • Manifest, Class, Recipe, Cookbook • puppet • chef • chef-solo • chef-client • chef-server-ctl • knife
Service Operations	
IT Operations and Monitoring	<p>Weight: 4</p> <p>Description: Candidates should understand how IT infrastructure is involved in delivering a service. This includes knowledge about the major goals of IT operations, understanding functional and nonfunctional properties of an IT services and ways to monitor and</p>

Topic	Details
	<p>measure them using Prometheus. Furthermore candidates should understand major security risks in IT infrastructure. This objective covers the feature set of Prometheus 1.7 or later.</p> <p>Key Knowledge Areas:</p> <ul style="list-style-type: none"> • Understand goals of IT operations and service provisioning, including nonfunctional properties such as availability, latency, responsiveness • Understand and identify metrics and indicators to monitor and measure the technical functionality of a service • Understand and identify metrics and indicators to monitor and measure the logical functionality of a service • Understand the architecture of Prometheus, including Exporters, Pushgateway, Alertmanager and Grafana • Monitor containers and microservices using Prometheus • Understand the principles of IT attacks against IT infrastructure • Understand the principles of the most important ways to protect IT infrastructure • Understand core IT infrastructure components and their role in deployment <p>The following is a partial list of the used files, terms and utilities:</p> <ul style="list-style-type: none"> • Prometheus, Node exporter, Pushgateway, Altermanager, Grafana • Service exploits, brute force attacks, and denial of service attacks • Security updates, packet filtering and application gateways • Virtualization hosts, DNS and load balancers

Topic	Details
Log Management and Analysis	<p>Weight: 4</p> <p>Description: Candidates should understand the role of log files in operations and troubleshooting. They should be able to set up centralized logging infrastructure based on Logstash to collect and normalize log data. Furthermore, candidates should understand how Elasticsearch and Kibana help to store and access log data.</p> <p>Key Knowledge Areas:</p> <ul style="list-style-type: none"> • Understand how application and system logging works • Understand the architecture and functionality of Logstash, including the lifecycle of a log message and Logstash plugins • Understand the architecture and functionality of Elasticsearch and Kibana in the context of log data management (Elastic Stack) • Configure Logstash to collect, normalize, transform and ship log data • Configure syslog and Filebeat to send log data to Logstash • Configure Logstash to send email alerts • Understand application support for log management <p>The following is a partial list of the used files, terms and utilities:</p> <ul style="list-style-type: none"> • logstash • input, filter, output • grok filter • Log files, metrics • syslog.conf • /etc/logstash/logstash.yml • /etc/filebeat/filebeat.yml

LPI 701-100 Sample Questions:

Question: 1

When working with remote repositories in Git, what is a typical action?

- a) Discarding local changes
- b) Pulling changes from the remote repository
- c) Restricting branch creation
- d) Using SVN commands

Answer: b

Question: 2

What can cloud-init do in terms of file systems?

- a) Create, resize, and mount file systems
- b) Only delete file systems
- c) Disallow the creation of new file systems
- d) Restrict access to file systems

Answer: a

Question: 3

Which feature of Prometheus is used to push metrics from jobs that cannot be scraped?

- a) Pushgateway
- b) Grafana
- c) Alertmanager
- d) Node exporter

Answer: a

Question: 4

How does Ansible interact with remote systems?

- a) Through direct database connections
- b) Primarily using SSH
- c) Via encrypted email messages
- d) Using FTP transfers

Answer: b

Question: 5

In DevOps practices, what is a primary implication for software developers and operators?

- a) Working in isolated teams
- b) Increased collaboration and integration
- c) Reduced need for automation
- d) Focus solely on development, not operations

Answer: b

Question: 6

What is a typical feature of Platform as a Service (PaaS) offerings like CloudFoundry and OpenShift?

- a) Complete control over the underlying infrastructure
- b) Integrated development and deployment environment
- c) Limited language and framework support
- d) High requirement for system administration

Answer: b

Question: 7

What feature of Packer allows it to work with different platforms?

- a) Single-builder functionality
- b) Limited template options
- c) Multiple builders in template files
- d) Fixed output formats

Answer: c

Question: 8

In Docker, what is the purpose of docker-compose?

- a) To build images from Dockerfiles
- b) To run single-container applications
- c) To define and run multi-container Docker applications
- d) To manage Docker volumes

Answer: c

Question: 9

Which of the following are key aspects of designing service-based applications for modern runtime environments?

- a) Implementing monolithic structures
- b) Using stateful components
- c) Ignoring API standards
- d) Handling data persistence and session management

Answer: d

Question: 10

In a Jenkins pipeline, what is a 'Stage'?

- a) A step where the pipeline pauses for manual input
- b) A tool for visualizing test results
- c) A method to revert changes
- d) A distinct phase of the pipeline process

Answer: d

Study Guide to Crack LPI DevOps Tools Engineer 701-100 Exam:

- Getting details of the 701-100 syllabus, is the first step of a study plan. This pdf is going to be of ultimate help. Completion of the syllabus is must to pass the 701-100 exam.
- Making a schedule is vital. A structured method of preparation leads to success. A candidate must plan his schedule and follow it rigorously to attain success.
- Joining the LPI provided training for 701-100 exam could be of much help. If there is specific training for the exam, you can discover it from the link above.
- Read from the 701-100 sample questions to gain your idea about the actual exam questions. In this PDF useful sample questions are provided to make your exam preparation easy.
- Practicing on 701-100 practice tests is must. Continuous practice will make you an expert in all syllabus areas.

Reliable Online Practice Test for 701-100 Certification

Make EduSum.com your best friend during your LPI DevOps Tools Engineer - 701 exam preparation. We provide authentic practice tests for the 701-100 exam. Experts design these online practice tests, so we can offer you an exclusive experience of taking the actual 701-100 exam. We guarantee you 100% success in your first exam attempt if you continue practicing regularly. Don't bother if you don't get 100% marks in initial practice exam attempts. Just utilize the result section to know your strengths and weaknesses and prepare according to that until you get 100% with our practice tests. Our evaluation makes you confident, and you can score high in the 701-100 exam.

Start Online practice of 701-100 Exam by visiting URL

<https://www.edusum.com/lpi/701-100-lpi-devops-tools-engineer-701>