



Linux
Professional
Institute

Exam Preparation Checklist

Linux Professional Institute
DevOps Tools Engineer

目次

はじめに	3
チェックリストの使い方	4
1. 701 ソフトウェアエンジニアリング	5
2. 702 コンテナ管理	8
3. 703 マシンデプロイメント	10
4. 704 設定管理	12
5. 705 サービスオペレーション	14
Linux Professional Instituteについて	16

はじめに

Welcome this Exam Preparation Checklist!

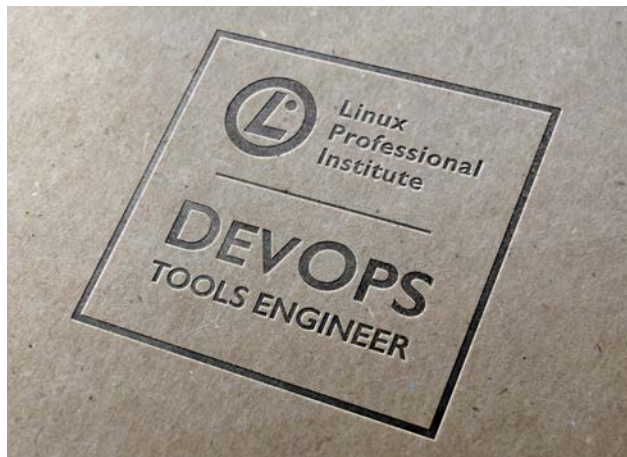
このチェックリストは、Linux Professional Institute DevOps Tools Engineer認定資格の、公式の試験範囲や勉強方法を紹介しています。これを目安に、何が試験に関係ある事柄なのか確かめながら、試験勉強に取り組んでください。

実際の試験では、あらゆるトピックからの問題が出題されます。各トピックは複数の項目に分割されています。これらには重要度が設定されているので、それに合わせて勉強時間を調整してください。

このリストの各トピックの最初で、求められるスキルについて説明しています。また、主要な知識分野と重要なファイル、コマンド、用語の詳細なリストも掲載しています。説明に記載されているタスクを実際に行うことができること、そして記載されているファイルとコマンドを、それぞれのコンテキストで適用する方法を理解できるようになります。

試験、認定資格、サポート等に関してご質問がございましたら、info-ja@lpi.orgにお問い合わせください。

ご健闘をお祈りいたします!



The Author



Fabian Thorns is the Director of Certification Development at Linux Professional Institute, LPI. He is M.Sc. Business Information Systems, a regular speaker at open source events and the author of numerous articles and books. Fabian has been part of the exam development team since 2010. Connect with him via

email: fthorns@lpi.org.

チェックリストの使い方

このガイドでは、3段階の学習フェーズで試験勉強を進めることを前提としています：

- **学習フェーズ (Study)** では、試験に関連するすべてのものについて、積極的にノートを取りましょう。試験が近くなったら、これらのメモを見直すと効果的です。そのため、最初からメモを取り、十分に包括的であることを確認します。後で詳細を思い出してください。
- **復習フェーズ (Preparation)** では、学習フェーズで取ったノートを繰り返し見直しましょう。試験範囲と注意点の両方を確認し、分からない部分を詳しく調べ、ノートに書き足しましょう。これらのメモに加えて、まだ自信がないと感じるトピックにも印を付けます。
- **完了フェーズ (Completion)** では、今まで書いてきたノートをすべて見直し、全てのトピックについて自信があることを確認しましょう。まだ試験まで時間がある場合は、復習フェーズでマークしたメモをもう一度見直して、苦手な部分をなくしましょう。

次の段階に進む前に、各段階を完了するようにしましょう。一歩ずつ進むことで時間をかけて知識を蓄積し、それを磨くのに十分な時間を確保することができます。

このチェックリストは、LPIの公式の試験範囲を示しています。各トピックの右側に、それぞれのフェーズを完了した後にチェックする、3つのチェックボックスがあるので活用してください。試験勉強の際は、メモをとるための専用のノートなどを用意しましょう。

注意: LPIは試験準備の具体的な方法を推奨していません。ですが、試験勉強は公式の試験範囲に沿って行いましょう。 www.lpi.org をチェックして、最新バージョンの試験目的に取り組んでいることを確認してください。

I. ソフトウェアエンジニアリング

701 ソフトウェアエンジニアリング

701.1 モダンなソフトウェア開発	Study	Preparation	Completion
総重量: 6			
説明: データの永続性・セッション・ステータス情報・トランザクション・並行性・セキュリティ・パフォーマンス・可用性・スケーリング・ロードバランス・メッセージング・モニタリング・APIを、どう操作するか理解している。さらに、アジャイルの影響とソフトウェア開発についてのDevOpsについて理解している。			
主な知識分野:			
• アプリケーションベースのサービスについての、理解と設計	✓	✓	✓
• 共通APIのコンセプトと基準の理解	✓	✓	✓
• データストレージ・サービスステータス・セッション操作の側面の理解	✓	✓	✓
• コンテナで実行することを想定したソフトウェアの設計	✓	✓	✓
• クラウドサービスへデプロイすることを想定したソフトウェアの設定	✓	✓	✓
• マイグレーションのリスクの知識と、モノリシックなレガシーソフトウェアのインテグレーション	✓	✓	✓
• 共通のセキュリティリスクとそれらの軽減の理解	✓	✓	✓
• アジャイルソフトウェア開発のコンセプトの理解	✓	✓	✓
• DevOpsのコンセプトと、ソフトウェア開発者とオペレータへの影響についての理解	✓	✓	✓
利用されるファイル・用語・ユーティリティ:			
• REST, JSON	✓	✓	✓
• Service Orientated Architectures (SOA)	✓	✓	✓
• マイクロサービス	✓	✓	✓
• 普遍的なサーバ構成	✓	✓	✓
• 疎結合	✓	✓	✓
• クロスサイトスクリプティング・SQLインジェクション・冗長なエラー報告・API認証・より強力な伝送路暗号化の継続性	✓	✓	✓
• CORSヘッダとCSRFトークン	✓	✓	✓
• ACIDプロパティとCAP定理	✓	✓	✓

701 ソフトウェアエンジニアリング

701.2 ソフトウェアのコンポーネントとプラットフォームの標準	Study	Preparation	Completion
総重量: 2			
説明: 共通のクラウドプラットフォームにより提供される、サービスを理解している。必要なサービスの設定の理解があり、アプリケーションアーキテクチャーとデプロイのプログラムの連携のサービスを利用することができる。参照実装として、OpenStackのサービスコンポーネントを利用することができる。			
主な知識分野:			
• オブジェクトストレージの機能と概念	✓	✓	✓
• リレーショナルデータベースとNoSQLデータベースの機能と概念	✓	✓	✓
• メッセージブローカーとメッセージキューの機能と概念	✓	✓	✓
• ビッグデータサービスの機能と概念	✓	✓	✓
• アプリケーションランタイム/PaaSの機能と概念	✓	✓	✓
• コンテンツデリバリーネットワークの機能と概念	✓	✓	✓
利用されるファイル・用語・ユーティリティ:			
• OpenStack Swift	✓	✓	✓
• OpenStack Trove	✓	✓	✓
• OpenStack Zaqar	✓	✓	✓
• CloudFoundry	✓	✓	✓
• OpenShift	✓	✓	✓
701.3 ソースコード管理	Study	Preparation	Completion
総重量: 5			
説明: ソースコードの管理と共有に、Gitを利用することができる。タグ・ブランチ・リポジトリの利用だけでなく、リポジトリの作成と提供が含まれる。さらに、ファイルのマージとマージコンフリクトを解決することができる。			
主な知識分野:			
• Gitのコンセプトとリポジトリ構成の理解	✓	✓	✓
• Gitリポジトリ内でのファイルの管理	✓	✓	✓
• ブランチとタグの管理	✓	✓	✓
• サブモジュールだけではなく、リポジトリとブランチの作業	✓	✓	✓
• ファイルとブランチのマージ	✓	✓	✓
• 中央集中型と分散型のSCM(Source Code Management)ソリューションを含んだ、SVNとCVSの知識	✓	✓	✓

701 ソフトウェアエンジニアリング

	Study	Preparation	Completion
利用されるファイル・用語・ユーティリティ:			
• git	✓	✓	✓
• .gitignore	✓	✓	✓
701.4 継続的インテグレーションと継続的デリバリー	Study	Preparation	Completion
総重量: 5			
説明: 継続的インテグレーションと継続的インテグレーションの連鎖の原理とコンポーネントの理解。CI/CDの連鎖のトリガ・ユニットテスト・テストの導入と受諾・ソフトウェアのパッケージング・テストが行われたソフトウェア成果物のデプロイの操作などを含む、Jenkinsを用いたCI/CDの連鎖の実装。試験範囲は、Jenkins version 2.0以降の機能をカバーしている。			
主な知識分野:			
• 継続的インテグレーションと継続的デリバリーの概念の理解	✓	✓	✓
• ビルド・ユニットテスト・テストの導入と受諾・成果物管理・デリバリー・デプロイを含むCI/CDの連鎖のコンポーネントの概念の理解	✓	✓	✓
• デプロイのベストプラクティスの理解	✓	✓	✓
• Jenkinsの、Jenkinsプラグイン・Jenkins API・通知・分散ビルドを含んだ、アーキテクチャーと機能の理解	✓	✓	✓
• パラメータ操作を含む、Jenkinsのジョブジョブの実行と定義	✓	✓	✓
• 指紋取得(Fingerprinting)、成果物、成果物リポジトリ	✓	✓	✓
• どのようにJenkinsモデルの継続的デリバリーが連鎖するか、そしてJenkinsで宣言的継続的デリバリーの連鎖を実装するか	✓	✓	✓
• 有効な認証と承認モデルの知識	✓	✓	✓
• パイプラインプラグインの理解	✓	✓	✓
• Copy Artifactプラグイン・Fingerprintプラグイン・Dockerパイプライン・Docker Build and Publishプラグイン・Gitプラグイン・Credentialsプラグインなどの、重要なJenkinsモジュールの機能の理解	✓	✓	✓
• ArtifactoryとNexusの知識	✓	✓	✓
利用されるファイル・用語・ユーティリティ:			
• Step, Node, Stage	✓	✓	✓
• Jenkins SDL	✓	✓	✓
• Jenkinsfile	✓	✓	✓
• Declarative Pipeline	✓	✓	✓
• Blue-green and canary deployment	✓	✓	✓

2. コンテナ管理

Topic 702 コンテナ管理			
702.1 コンテナの利用方法	Study	Preparation	Completion
総重量: 7			
説明: Dockerコンテナのビルド・シェア・操作することが可能である。ネットワークとストレージボリュームをコンテナに接続するだけでなく、Dockerレジストリを用いて、コンテナの作成・操作などを実現するDockerfileを作成できることを含まれる。試験範囲は、Dockerのversion 17.06以降の機能をカバーしている。			
主な知識分野:			
• Dockerの構造を理解する	✓	✓	✓
• Dockerレジストリから存在しているDockerイメージを利用する	✓	✓	✓
• Dockerfileを作成し、Dockerfileからイメージをビルドする	✓	✓	✓
• Dockerリポジトリにイメージをアップロードする	✓	✓	✓
• Dockerコンテナを操作しアクセスする	✓	✓	✓
• コンテナをDockerネットワークに接続する	✓	✓	✓
• コンテナストレージの共有あるいは永続化のために、Dockerボリュームを利用する	✓	✓	✓
利用されるファイル・用語・ユーティリティ:	✓	✓	✓
• docker	✓	✓	✓
• Dockerfile	✓	✓	✓
• .dockerignore	✓	✓	✓
702.2 コンテナのデプロイとオーケストレーション	Study	Preparation	Completion
総重量: 5			
説明: サービスを提供するために、複数のコンテナを同時に動かし管理することができる。これには、既存のKubernetesクラスタを利用するだけでなく、存在しているDocker Swarmによる連結をDocker Composeで実現して、Dockerコンテナ群でのオーケストレーションの実現含んでいる。これには、Docker Compose version 1.14以降、Docker 17.06以降に含まれたDockerSwarm、Kubernetes 1.6以降がカバーされている。			
主な知識分野:			
• Docker Composeのアプリケーションモデルを理解	✓	✓	✓
• Docker Composeファイル(version 3以降)の作成と実行	✓	✓	✓
• Docker Swarmモードの構造と機能の理解	✓	✓	✓
• サービス・スタックの定義とシークレットの利用方法を含んだ、Docker Swarmでコンテナを実行する	✓	✓	✓
• Kubernetesの構造とアプリケーションモデルの理解	✓	✓	✓

Topic 702 コンテナ管理

	Study	Preparation	Completion
<ul style="list-style-type: none"> Deployment, Services, ReplicaSet, Podsの定義を含む、Kubernetesのコンテナベースアプリケーションの定義と管理 	✓	✓	✓
利用されるファイル・用語・ユーティリティ:			
<ul style="list-style-type: none"> docker-compose 	✓	✓	✓
<ul style="list-style-type: none"> docker 	✓	✓	✓
<ul style="list-style-type: none"> kubectl 	✓	✓	✓
702.3 コンテナインフラストラクチャー	Study	Preparation	Completion
総重量: 4			
<p>説明: コンテナの実行環境をセットアップすることができる。特定のコンテナホストだけではなく、ローカルのワークステーションでコンテナを動作させることも含まれている。さらに、コンテナインフラストラクチャー・ストレージ・ネットワーキング・特定のセキュリティの側面からのコンテナの知識がある。この試験範囲は、Docker version 17.06以降とDocker Machine 0.12以降の機能をカバーしている。</p>			
主な知識分野:			
<ul style="list-style-type: none"> Dockerホストをセットアップするために、Docker Machineを利用する 	✓	✓	✓
<ul style="list-style-type: none"> オーバーレイネットワークを含む、Dockerネットワーキングの概念の理解 	✓	✓	✓
<ul style="list-style-type: none"> Dockerネットワークの理解と管理 	✓	✓	✓
<ul style="list-style-type: none"> Dockerストレージの概念の理解 	✓	✓	✓
<ul style="list-style-type: none"> Dockerボリュームの作成と管理 	✓	✓	✓
<ul style="list-style-type: none"> Flockerとflannelの知識 	✓	✓	✓
<ul style="list-style-type: none"> サービスディスカバリーの概念の理解 	✓	✓	✓
<ul style="list-style-type: none"> CoreOS Container Linux, rkt, etcdの基本的な機能の知識 	✓	✓	✓
<ul style="list-style-type: none"> コンテナ仮想化とコンテナイメージのセキュリティリスクと、それらの軽減の方法の理解 	✓	✓	✓
利用されるファイル・用語・ユーティリティ:			
<ul style="list-style-type: none"> docker-machine 	✓	✓	✓

3. マシンデプロイメント

703 マシンデプロイメント			
703.1 仮想マシンのデプロイメント	Study	Preparation	Completion
総重量: 4			
説明: オペレーティングシステムと特定の設定ファイルのセットとソフトウェアが入った仮想マシンのデプロイの自動化ができる。			
主な知識分野:			
• Vagrantのストレージとネットワークの機能を含んだ構造と概念の理解	✓	✓	✓
• Atlasからのボックスの検索と利用	✓	✓	✓
• Vagrantfileの作成と実行	✓	✓	✓
• Vagrant仮想マシンへのアクセス	✓	✓	✓
• Vagrant仮想マシンとホスシステム間のフォルダの共有と同期	✓	✓	✓
• File, Shell, Ansible, Dockerを含んだVagrantのプロビジョニングの理解	✓	✓	✓
• 複数マシンのセットアップの理解	✓	✓	✓
利用されるファイル・用語・ユーティリティ:			
• vagrant	✓	✓	✓
• Vagrantfile	✓	✓	✓
703.2 クラウドへのデプロイ	Study	Preparation	Completion
総重量: 2			
説明: IaaSクラウドインスタンスが設定可能で、特にディスクの容量や空き容量の、利用可能なハードウェアリソースにマッチした調整をすることができる。さらに、インスタンスにセキュアなSSHログインを許可し、Ansibleのような設定管理ツールを利用可能に準備することができる。			
主な知識分野:			
• ユーザデータと初期化とcloud-initの設定を含む、cloud-initの機能と概念の理解	✓	✓	✓
• SSHキーのようなログイン証明書と配布リポジトリからのソフトウェアパッケージのインストールを含む、ファイルシステムの作成・リサイズ・マウント、ユーザアカウントの設定、cloud-initの利用	✓	✓	✓
• スナップショット・一時停止・クローン・リソース制限などの、コンピューティングインスタンスの仮想化とIaaSクラウドの、機能と影響の理解	✓	✓	✓

703 マシンデプロイメント

703.3 システムイメージの作成

Study

Preparation

Completion

総重量: 2

説明: コンテナ・仮想マシン・IaaSクラウドインスタンスのイメージを作成することができる。

主な知識分野:

- Packerの機能と仕様の理解
- テンプレートファイルの作成と管理
- 異なるビルダを利用したテンプレートファイルからイメージを構築

利用されるファイル・用語・ユーティリティ:

- packer

4. 設定管理

Topic 704 設定管理			
704.1 Ansible	Study	Preparation	Completion
総重量: 8			
説明: Ansibleを利用して、対象のサーバが設定ファイルにより特定の状態であり、ソフトウェアがインストールされている状態にすることができる。この試験範囲は、Ansible version 2.2以降をカバーしている。			
主な知識分野:			
• システムの設定とソフトウェアのインストールの自動化の、概念の理解	✓	✓	✓
• インベントリファイルの作成と管理	✓	✓	✓
• リモートシステムのAnsibleの動作の理解	✓	✓	✓
• 非特権アカウントの利用を含む、AnsibleのSSHログイン証明書の管理	✓	✓	✓
• task, handlers, loops, registersを含むAnsible playbookの作成・管理・実行	✓	✓	✓
• 変数の定義と利用	✓	✓	✓
• Ansible vaultを利用した、パスワードの管理	✓	✓	✓
• 共通のfilter, loop, conditionalを用いた、Jinja2テンプレートの記述	✓	✓	✓
• Ansible roleの利用と理解、Ansible GalaxyからのAnsible roleのインストール	✓	✓	✓
• ファイルのinclude, copy, template, ini_file, lineinfile, patch, replace, user, group command, shell, service, systemd, cron, apt, debconf, yum, git, debugの、重要なAnsible taskの理解と利用	✓	✓	✓
• AnsibleのDynamic Inventoryの知識	✓	✓	✓
• 非LinuxシステムでのAnsibleの仕様の知識	✓	✓	✓
• Ansibleコンテナの知識	✓	✓	✓
利用されるファイル・用語・ユーティリティ:	✓	✓	✓
• ansible.cfg	✓	✓	✓
• ansible-playbook	✓	✓	✓
• ansible-vault	✓	✓	✓
• ansible-galaxy	✓	✓	✓
• ansible-doc	✓	✓	✓

Topic 704 設定管理

704.2 他の設定管理ツール	Study	Preparation	Completion
総重量: 2			
説明: Ansible以外の重要な設定管理ツールの主な仕様と概念の理解している。			
主な知識分野:			
• Puppetの基本的な仕様と構造の知識	✓	✓	✓
• Chefの基本的な仕様と構造の知識	✓	✓	✓
利用されるファイル・用語・ユーティリティ:			
• Manifest, Class, Recipe, Cookbook	✓	✓	✓
• puppet	✓	✓	✓
• chef	✓	✓	✓
• chef-solo	✓	✓	✓
• chef-client	✓	✓	✓
• chef-server-ctl	✓	✓	✓
• knife	✓	✓	✓

5. サービスオペレーション

Topic 705 サービスオペレーション			
705.1 IT オペレーションと監視	Study	Preparation	Completion
総重量: 4			
説明: ITインフラがサービスのデリバリーを実行する方法を理解している。これには、ITサービスの機能・非機能の理解と、Prometheusを利用したそれらの監視と測定の方法を理解、ITオペレーションの主要な目標についての理解が含まれている。さらに、ITインフラの主要なセキュリティリスクを理解している。この試験範囲は、Prometheus バージョン1.7以降の機能セットをカバーしている。			
主な知識分野:			
• 可用性・反応速度・黄桃などの非機能プロパティを含む、ITオペレーションとサービスのプロビジョニングの目標を理解している	✓	✓	✓
• サービスの技術的な機能を監視・測定するための、指標や基準を理解し特定する	✓	✓	✓
• サービスの論理的な機能を監視・測定するための、指標や基準を理解し特定する	✓	✓	✓
• Exporters, Pushgateway, Altermanager, Grafanaを含む、Prometheusの構造の理解	✓	✓	✓
• Prometheusを利用して、コンテナとマイクロサービスを監視	✓	✓	✓
• ITインフラに対するIT攻撃の原理の理解	✓	✓	✓
• ITインフラを守る重要な方法の原理の理解	✓	✓	✓
• ITインフラコンポーネントの主要部分と、デプロイでの役割の理解	✓	✓	✓
利用されるファイル・用語・ユーティリティ:			
• Prometheus, Node exporter, Pushgateway, Altermanager, Grafana	✓	✓	✓
• Service exploits, ブルートフォースアタック, DoS	✓	✓	✓
• セキュリティアップデート、パケットフィルタリング、アプリケーションゲートウェイ	✓	✓	✓
• 仮想化ホスト、DNSとロードバランサー	✓	✓	✓

Topic 705 サービスオペレーション

705.2 ログの管理と分析	Study	Preparation	Completion
総重量: 4			
説明: オペレーションとトラブルシューティングにおいて、ログファイルの役割を理解している。ログデータの取得と標準化を、Logstashに基づいた中央集中化したログインフラの設定をすることができる。さらに、ログデータの保存とアクセスのために、ElasticSearchとKibanaの利用方法を理解している。			
主な知識分野:			
• アプリケーションとシステムログの働きの理解	✓	✓	✓
• ログメッセージのライフサイクルとLogstashのプラグインを含む、Logstashの機能と構造の理解	✓	✓	✓
• ログデータ管理(Elastic Stack)の背景とするElasticsearchとKibanaの構造と機能の理解	✓	✓	✓
• ログデータの収集・標準化・変換・出荷のためのLogstashの設定	✓	✓	✓
• Logstashにログデータを送るため、syslogとFilebeatの設定	✓	✓	✓
• emailによる警告を送るための、Logstashの設定	✓	✓	✓
• ログ管理のためのアプリケーションサポートの理解	✓	✓	✓
利用されるファイル・用語・ユーティリティ:			
• logstash	✓	✓	✓
• input, filter, output	✓	✓	✓
• grok filter	✓	✓	✓
• Log files, metrics	✓	✓	✓
• syslog.conf	✓	✓	✓
• /etc/logstash/logstash.yml	✓	✓	✓
• /etc/filebeat/filebeat.yml	✓	✓	✓

About LPI

LPIは、オープンソースのナレッジおよびスキル認定を広く利用できるようにすることによって、すべての人にとって経済的で創造的な機会を可能にすることを使命とする非営利団体です。私たちはLinuxの世界的な認定基準であり、世界中のオープンソース専門家のためのキャリア支援組織です。

60万以上の試験が提供され、LPIは世界初で最大のベンダ中立のLinuxおよびオープンソーススキル認定機関です。私たちは180カ国以上の国々で認定されたITプロフェッショナルを擁し、9カ国語で試験を実施し、何百ものトレーニングパートナーを擁しています。

