# ****Topic 701: Software Engineering****

### ****701.1 Modern Software Development (weight: 6)****

**Weight:**6

**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.

**Key Knowledge Areas:**

* 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

### ****701.2 Standard Components and Platforms for Software (weight: 2)****

**Weight:** 2

**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.

**Key Knowledge Areas:**

* 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

### ****701.3 Source Code Management (weight: 5)****

**Weight:** 5

**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.

**Key Knowledge Areas:**

* 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

### ****701.4 Continuous Integration and Continuous Delivery (weight: 5)****

**Weight:** 5

**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.

**Key Knowledge Areas:**

* Understand the concepts of Continuous Integration and Continuous Delivery
* 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

# ****Topic 702: Container Management****

### ****702.1 Container Usage (weight: 7)****

**Weight:**7

**Description:**Candidates should be able to build, share and operate Docker containers. This includes creating 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.

**Key Knowledge Areas:**

* 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

### ****702.2 Container Deployment and Orchestration (weight: 5)****

**Weight:** 5

**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.

**Key Knowledge Areas:**

* Understand the application model of Docker Compose
* 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

### ****702.3 Container Infrastructure (weight: 4)****

**Weight:** 4

**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.

**Key Knowledge Areas:**

* Use Docker Machine to setup a Docker host
* Understand Docker networking concepts, including overlay networks
* Create and manage Docker networks
* Understand Docker storage concepts
* 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

# ****Topic 703: Machine Deployment****

### ****703.1 Virtual Machine Deployment (weight: 4)****

**Weight:** 4

**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.

**Key Knowledge Areas:**

* 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

### ****703.2 Cloud Deployment (weight: 2)****

**Weight:** 2

**Description:** Candidates should be able to configure IaaS cloud instances and adjust them to match their available hardware resources, specifically, disk space and volumes. Additinally, 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.

**Key Knowledge Areas:**

* 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.

### ****703.3 System Image Creation (weight: 2)****

**Weight:** 2

**Description:** Candidates should be able to create images for containers, virtual machines and IaaS cloud instances.

**Key Knowledge Areas:**

* Understand the functionality and features of Packer
* Create and maintain template files
* Build images from template files using different builders

# ****Topic 704: Configuration Management****

### ****704.1 Ansible (weight: 8)****

**Weight:**8

**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.

**Key Knowledge Areas:**

* 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

# ****704.2 Other Configuration Management Tools (weight: 2)****

**Weight:**2

**Description:** Candidates should understand the main features and principles of important configuration management tools other than Ansible.

**Key Knowledge Areas:**

* Basic feature and architecture knowledge of Puppet.
* Basic feature and architecture knowledge of Chef.

# ****Topic 705: Service Operations****

### ****705.1 IT Operations and Monitoring (weight: 4)****

**Weight:**4

**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 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.

**Key Knowledge Areas:**

* 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 the role in deployment