测试项目 - 样题

***山东省职业技能大赛***

***网络系统管理项目模块D-Automation环境***

# 目录

[目录 2](#_Toc20402)

[《测试项目》简介 3](#_Toc6092)

[介绍 3](#_Toc21088)

[项目和任务的描述 3](#_Toc5803)

[选手须知 5](#_Toc10152)

[第1部分Linux 5](#_Toc27974)

[第2部分Windows 7](#_Toc15392)

[第3部分API 10](#_Toc5153)

[所需的设备、机械、装置和材料 12](#_Toc4957)

[网络表 12](#_Toc16669)

[网络（逻辑）图 15](#_Toc12785)

[网络（物理）图 16](#_Toc15341)

# 《测试项目》简介

以下是提交给 WorldSkills 的所有《测试项目》提案中必须包含的章节或信息清单。

* 目录，包括组成《测试项目》的所有文件、图纸和照片的清单
* 介绍/概述
* 项目和任务的简短描述
* 选手须知
* 完成《测试项目》所需的设备、机械、装置和材料
* 《评分方案》（包括评估标准）
* 其他

# 介绍

此《测试项目》由以下文档/文件组成：

* 样题.docx
* users.csv
* customers.json

准确和最新的文档一直是信息技术的挑战。由于在同一系统有很多工程师一起工作，这就使追踪是谁改变了什么内容变得很困难。Applix 公司决定处理这个问题，并雇佣你来对他们的基础设施进行现代化改造，强化和拓展。

# 项目和任务的描述

您将虚拟机迁移为基础架构即代码(IaC)，并简化创建新服务的过程。你可以访问开发虚拟机(DEV-LIN & DEV-WIN)。你可以使用这些虚拟机来开发及测试你的工作，你应该完全能够使用Ansible和Python来完成所有的任务。

### 登录所有虚拟机和设备：

Linux 用户名 : root / appadmin

Windows用户名 : Administrator / appadmin

密码: Incheon-2022

所有的虚拟机和设备都连接到管理网络(10.22.0.0/24)，都有静态配置IP地址。网络表中的所有IP地址都不会因为评分而更改。管理网络可以被用来配置不同的主机。你可以在SSH, HTTPS or WinRM上使用用户名和密码登录。

你可以在虚拟机上安装额外所需的软件包和功能。

# 选手须知

## 第1部分Linux

从HOST VM使用Ansible和Python脚本配置Linux主机LIN[1-5]。在评分前，所有的Linux VMs都会重置到初始状态。

您可以将Debian 11 ISO连接到VMs。

### Hostname

* 配置主机名
  + 所有主机都自动配置网络表中的主机名

### IPtables

* 配置iptables来过滤所有LIN hosts上的传入流量
  + 默认情况下应丢弃传入数据包
  + 允许服务所需的最小流量
  + 允许从HOST和DEV VMs到LIN主机的SSH和ICMP流量
  + 确保iptables在重启后仍然存在

### DNS

* 配置两台或更多DNS服务器
* 在“dns”组的所有主机上安装DNS服务
  + “dns”组第一台主机应该对于“applix.com”域是权威的主服务器
  + 该组中的所有其他主机都应该是“applix.com”域的从属DNS服务器
  + 确保所有的主机都有<hostname>.applix.com的A记录
  + intranet.applix.com应解析为10.22.0.51
* 自动配置客户端
  + 确保所有LIN服务器和HOST使用第一台DNS主机作为主DNS服务器，所有的从属DNS服务器作为辅助名称服务器
  + 为DNS配置后缀“applix.com”

### Web

* 配置两台及更多web服务器
* 在 “web”组所有主机安装web服务
  + 本地网站应该监听在80端口
  + 主页显示以下内容 “Hello from <hostname> !”
  + 创建一个虚拟机监听在8081端口上，名为“intranet.applix.com”，显示以下内容

“Welcome to the intranet of Applix”

“This site was served by <hostname>”

## 第2部分Windows

从HOST VM使用Ansible和python配置Windows主机WIN[1-5]。

### Hostname

* + 使用自动化配置主机名；

### Root certificate

* 配置CA，发布所有的证书；

### Security and logging

* 在所有Windows hosts停止和禁用远程桌面服务
* 创建名为LogUptime的计划任务，以每30秒将当前正常运行时间附加到 C:\uptime.txt中，格式如下所示

XX days XX hours XX minutes XX seconds XX days XX hours XX minutes XX seconds XX days XX hours XX minutes XX seconds

### Customer Deployments

* 将“dc”组所有服务器配置为所有WIN hosts的DNS服务器
* 将“dc”组所有服务器配置为域控制器
* 使用customers.com作为域名
* 使用“Incheon-2022”作为安全密码
* 将所有非DC的Windows hosts加入域
* 在“iis”组的所有服务器安装IIS功能
* 部署客户web环境
  + 信息是customers.json文件的所有客户
  + 基于“name”属性创建OU
  + 基于“username”和“password”属性，在此OU上创建AD用户
  + 为domain\_prefix.customers.com创建一个指向随机IIS服务器的DNS条目，并使用domain\_prefix作为randomization seed
  + 在所选的IIS服务器上创建一个虚拟主机监听80端口，显示“message”属性

## 第3部分API

创建一个Python API来查询有关基础设施的数据。API应当被托管在**HOST** VM上，所有API文件应位于/data/api下。你可以自由地使用Debian ISO上任何可用的模块和框架。

* API应该在IP 10.22.0.50的443端口监听HTTPS
  + 确保您的API通过https://api.applix.com/可从HOST和DEV VMs访问
  + 在DEV VMs上Edge & Firefox浏览器应该没有证书警报
* 创建名为“applix-api”的systemd服务
  + 该服务应该开机自启
  + API应当可以通过“start, restart, stop”systemd命令进行控制
* Endpoints一般信息
  + 确保API可以返回关于内容格式的信息
  + Token身份验证是使用HTTP请求中的“token”头部完成的

### Endpoint /stats (GET)

* + 此端点应当可以以JSON格式从HOST VM返回使用情况的统计数据

{

"ram\_free": "X",

"uptime": "XX:XX:XX:XX,

"api\_pid": "X",

"api\_prio": "X",

"total\_proc": "X”

}

* + ram\_free = free amount of memory in MB
  + uptime = uptime in format dd:hh:mm:ss
  + api\_pid = pid of current API process
  + api\_prio = current CPU priority of API process
  + total\_proc = total amount of processes running on system

### Endpoint /customers/ (PUT)

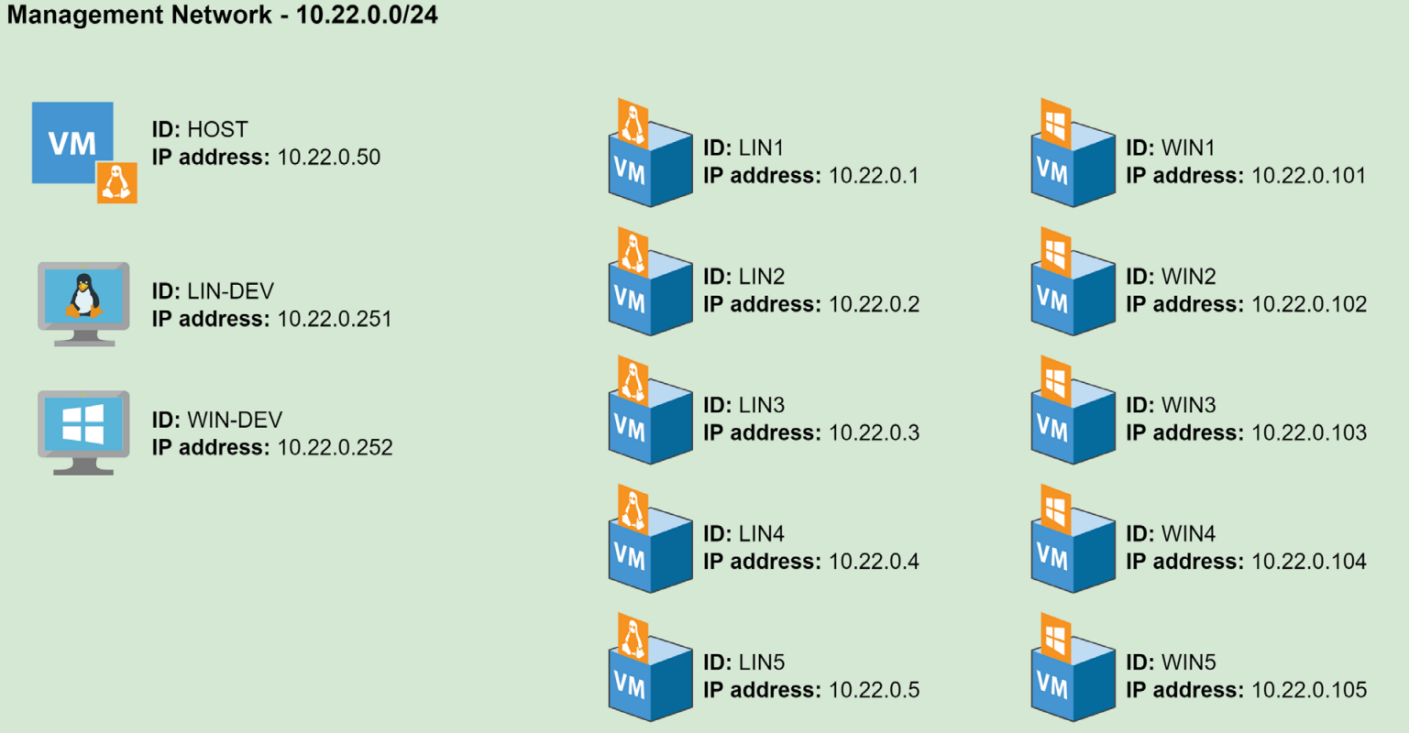
* + 此端点用于部署新客户
  + 此端点需要以下参数以JSON格式编写，在HTTP body中
    - name = customer name
    - domain\_prefix = prefix for domain
    - username = Active Directory username
    - password = Active Directory password
    - message = text displayed on website
  + 此端点应当可以将新用户添加在customers.json文件
    - 确保name, domain\_prefix和username都是唯一的
    - 如果不唯一，则返回一个HTTP Forbidden错误

# 所需的设备、机械、装置和材料

## 网络表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **IP** | **OS** | **HOSTNAME** | **GROUP** | **DESCRIPTION** |
| DEV-LIN | 10.22.0.251 | Debian 11.3 (GNOME) | * dev-lin | * dev | Development VM with the following software installed:   * Postman * Python3 |
| DEV-WIN | 10.22.0.252 | Windows 10 Pro | * dev-win | * dev | Development VM with the following software installed:   * Postman * Python3 |
| HOST | 10.22.0.50 | Debian 11.3 | host | host | Host Server |
| LIN1 | 10.22.0.1 | Debian 11.3 | APPLIX-LINUXHOST-001 | web | Dynamic Config |
| LIN2 | 10.22.0.2 | Debian 11.3 | APPLIX-LINUXHOST-002 | dns | Dynamic Config |
| LIN3 | 10.22.0.3 | Debian 11.3 | APPLIX-LINUXHOST-003 | dns | Dynamic Config |
| LIN4 | 10.22.0.4 | Debian 11.3 | APPLIX-LINUXHOST-004 | web/dns | Dynamic Config |
| LIN5 | 10.22.0.5 | Debian 11.3 | APPLIX-LINUXHOST-005 | dns | Dynamic Config |
| WIN1 | 10.22.0.101 | Windows Server 2019 | PETER | dc | Dynamic Config |
| WIN2 | 10.22.0.102 | Windows Server 2019 (Core) | MAX | dc | Dynamic Config |
| WIN3 | 10.22.0.103 | Windows Server 2019 (Core) | BEN | iis | Dynamic Config |
| WIN4 | 10.22.0.104 | Windows Server 2019 | ANNA | iis | Dynamic Config |
| WIN5 | 10.22.0.105 | Windows Server 2019 (Core) | LISA | iis | Dynamic Config |

## 网络（逻辑）图



## 网络（物理）图

