

IT 610 Systems Administration

Instructor Ryan Tolboom

Email Ryan.Tolboom@njit.edu

Office 3506 Guttenberg Information Technologies Center (GITC)

Office Hours 2:00PM to 4:00PM on Tuesdays or via Zoom (email to schedule)



Academic Integrity

Academic Integrity is the cornerstone of higher education and is central to the ideals of this course and the university. Cheating is strictly prohibited and devalues the degree that you are working on. As a member of the NJIT community, it is your responsibility to protect your educational investment by knowing and following the academic code of integrity policy that is found at: [NJIT Academic Integrity Code](#).

Please note that it is my professional obligation and responsibility to report any academic misconduct to the Dean of Students Office. Any student found in violation of the code by cheating, plagiarizing or using any online software inappropriately will result in disciplinary action. This may include a failing grade of F, and/or suspension or dismissal from the university. If you have any questions about the code of Academic Integrity, please contact the Dean of Students Office at dos@njit.edu.

Generative AI

Student use of artificial intelligence (AI) is permitted in this course for non-exam assignments and activities. Additionally, if and when students use AI in this course, the AI must be cited as is shown within the [NJIT Library AI citation page](#) for AI. If you have any questions or concerns about AI technology use in this class, please reach out to your instructor prior to submitting any assignments.

Student Absences for Religious Observations

NJIT is committed to supporting students observing religious holidays. Students must notify their instructors in writing of any conflicts between course requirements and religious observances, ideally by the end of the second week of classes and no later than two weeks before the anticipated absence. All instructors must include a reminder on the course syllabus about this notification process. All instructors are required to provide academically reasonable accommodations, allowing students to complete missed assignments, exams, quizzes, or other coursework within the term. Instructors are encouraged to consider the [NJIT religious holiday calendar](#) and exercise cultural sensitivity when scheduling assessments or major assignments. All instructors must ensure that students are not penalized for properly documented absences and maintain confidentiality regarding religious observances. For questions or additional guidance, please [review the policy](#) or contact the Office of Inclusive Excellence at inclusiveexcellence@njit.edu.

Objective

This course is a hands-on project intensive exploration of the advanced topics in systems administration. In addition to learning core competencies administering a Linux environment, students will also learn about the best practices for supporting a system or multi-server system in a containerized environment. Students will work on a single container image for the midterm project and a multi-container system for the final project.

Grading

- 20% Exercises
- 20% Midterm Exam
- 20% Midterm Project
- 20% Final Exam
- 20% Final Project

Course Materials

- A laptop meeting the [YWCC minimum specs](#)
- [Docker Desktop](#) installed and working
- [git](#) installed and working
- A [GitHub](#) account

Additional Resources

The following web pages will be very helpful while working on projects:

- [Canvas](#) - be sure your credentials are up to date
- [Docker Hub](#) - most images have excellent documentation
- [Linux Administration](#) - this hands-on, OER course in Linux Administration by Adrianna Holden-Gouveia is a great resource for additional help

Project Guidelines

Each project will be given a set of common deliverables that all student projects must meet for credit. Individual project deliverables will be settled upon after submission of the project proposal. The midterm project will be a basic, single-container deployment. The final project will utilize multiple containers and an orchestration framework.

Learning Outcomes

1. Linux

- 1.1 Access a shell prompt and issue commands with correct syntax
- 1.2 Access remote systems using SSH
- 1.3 Archive, compress, unpack, and uncompress files using tar, gzip, and bzip2
- 1.4 Create and edit text files
- 1.5 Create, delete, copy, and move files and directories
- 1.6 Add users, reset passwords, modify user groups, and delete users
- 1.7 Basic git operations
- 1.8 List, set, and change file permissions
- 1.9 Utilize a package management system
- 1.10 Create a package

2. Containers

- 2.1 Configure container engines, create, and manage containers
- 2.2 Create a container image
- 2.3 Build a container image
- 2.4 Create and backup container volumes
- 2.5 Deploy a database in a container

3. Container Orchestration

- 3.1 Use a container orchestration system to run a multi-container environment
- 3.2 Automate a deployment using popular automation tools
- 3.3 Design a custom deployment for a development environment

Course Outline

Week	Presentations	Exercises	Learning Outcomes
1	<ul style="list-style-type: none">• UNIX Systems• System Administration with Containers	<ul style="list-style-type: none">• Getting Started	<ul style="list-style-type: none">• 1.1 Access a shell prompt and issue commands with correct syntax• 1.7 Basic git operations• 2.1 Configure container engines, create, and manage containers• 2.3 Build a container image
2	<ul style="list-style-type: none">• Best Practices• Terminal Tips and Tricks• Docker Best Practices• Project	<ul style="list-style-type: none">• Project Proposal	
3	<ul style="list-style-type: none">• Users and Permissions• Package Management in Linux	<ul style="list-style-type: none">• Creating a Container Image	<ul style="list-style-type: none">• 1.1 Access a shell prompt and issue commands with correct syntax• 1.3 Archive, compress, unpack, and uncompress files using tar, gzip, and bzip2• 1.4 Create and edit text files• 1.5 Create, delete, copy, and move files and directories• 1.6 Add users, reset passwords, modify user groups, and delete users• 1.8 List, set, and change file permissions• 1.9 Utilize a package management system

Week	Presentations	Exercises	Learning Outcomes
			<ul style="list-style-type: none"> • 2.1 Configure container engines, create, and manage containers • 2.2 Create a container image • 2.3 Build a container image
4	<ul style="list-style-type: none"> • Storage • Backup 	<ul style="list-style-type: none"> • Creating a Debian Package 	<ul style="list-style-type: none"> • 1.1 Access a shell prompt and issue commands with correct syntax • 1.3 Archive, compress, unpack, and uncompress files using tar, gzip, and bzip2 • 1.4 Create and edit text files • 1.5 Create, delete, copy, and move files and directories • 1.8 List, set, and change file permissions • 1.9 Utilize a package management system • 1.10 Create a package • 2.1 Configure container engines, create, and manage containers • 2.2 Create a container image • 2.3 Build a container image • 2.4 Create and backup container volumes
5	<ul style="list-style-type: none"> • Automate the Boring Stuff 	<ul style="list-style-type: none"> • Volumes 	<ul style="list-style-type: none"> • 1.1 Access a shell prompt and issue commands with correct syntax • 2.1 Configure container engines, create, and manage containers • 2.2 Create a container image • 2.3 Build a container image • 2.4 Create and backup container volumes • 2.5 Deploy a database in a container • 3.2 Automate a deployment using popular automation tools
6	<ul style="list-style-type: none"> • Midterm Review 	<ul style="list-style-type: none"> • Practice Midterm 	
7	<ul style="list-style-type: none"> • Midterm • Midterm Deliverables Due 		<ul style="list-style-type: none"> • 1.1 Access a shell prompt and issue commands with correct syntax • 1.2 Access remote systems using SSH • 1.3 Archive, compress, unpack, and uncompress files using tar, gzip, and bzip2 • 1.4 Create and edit text files • 1.5 Create, delete, copy, and move files and directories • 1.6 Add users, reset passwords, modify user groups, and delete users • 1.7 Basic git operations • 1.8 List, set, and change file permissions • 1.9 Utilize a package management system • 1.10 Create a package • 2.1 Configure container engines, create, and manage containers • 2.2 Create a container image • 2.3 Build a container image • 2.4 Create and backup container volumes • 2.5 Deploy a database in a container

Week	Presentations	Exercises	Learning Outcomes
8	<ul style="list-style-type: none"> • Orchestration • Kubernetes • High-Availability Postgres in Kubernetes 	<ul style="list-style-type: none"> • Hello Kubernetes 	<ul style="list-style-type: none"> • 1.1 Access a shell prompt and issue commands with correct syntax • 1.4 Create and edit text files • 1.5 Create, delete, copy, and move files and directories • 2.1 Configure container engines, create, and manage containers • 2.2 Create a container image • 3.1 Use a container orchestration system to run a multi-container environment • 3.2 Automate a deployment using popular automation tools • 3.3 Design a custom deployment for a development environment
9	<ul style="list-style-type: none"> • Container Runtimes • Deploying an Application on Kubernetes 	<ul style="list-style-type: none"> • Exploring a Kubernetes Database Deployment 	<ul style="list-style-type: none"> • 1.1 Access a shell prompt and issue commands with correct syntax • 2.1 Configure container engines, create, and manage containers • 2.2 Create a container image • 2.3 Build a container image • 3.1 Use a container orchestration system to run a multi-container environment • 3.2 Automate a deployment using popular automation tools • 3.3 Design a custom deployment for a development environment
10	<ul style="list-style-type: none"> • Docker Compose 	<ul style="list-style-type: none"> • Docker Compose 	<ul style="list-style-type: none"> • 1.1 Access a shell prompt and issue commands with correct syntax • 1.4 Create and edit text files • 1.7 Basic git operations • 2.1 Configure container engines, create, and manage containers • 2.2 Create a container image • 2.3 Build a container image • 2.4 Create and backup container volumes • 3.1 Use a container orchestration system to run a multi-container environment • 3.3 Design a custom deployment for a development environment
11	<ul style="list-style-type: none"> • Cloud Computing • DevOps, SRE, and other Buzzwords 	<ul style="list-style-type: none"> • Creating a Developer Environment 	<ul style="list-style-type: none"> • 1.1 Access a shell prompt and issue commands with correct syntax • 1.4 Create and edit text files • 1.5 Create, delete, copy, and move files and directories • 1.9 Utilize a package management system • 2.1 Configure container engines, create, and manage containers • 2.2 Create a container image • 2.3 Build a container image • 2.4 Create and backup container volumes

Week	Presentations	Exercises	Learning Outcomes
			<ul style="list-style-type: none"> • 2.5 Deploy a database in a container • 3.1 Use a container orchestration system to run a multi-container environment • 3.2 Automate a deployment using popular automation tools • 3.3 Design a custom deployment for a development environment
12	<ul style="list-style-type: none"> • git 	<ul style="list-style-type: none"> • GitHub Exploration 	<ul style="list-style-type: none"> • 1.1 Access a shell prompt and issue commands with correct syntax • 1.7 Basic git operations
13	<ul style="list-style-type: none"> • What's Missing? 	<ul style="list-style-type: none"> • Updating a Linux Kernel 	<ul style="list-style-type: none"> • 1.1 Access a shell prompt and issue commands with correct syntax • 1.5 Create, delete, copy, and move files and directories
14	<ul style="list-style-type: none"> • Final Review 	<ul style="list-style-type: none"> • Practice Final 	
15	<ul style="list-style-type: none"> • Final • Final Deliverables Due 		<ul style="list-style-type: none"> • 1.1 Access a shell prompt and issue commands with correct syntax • 1.2 Access remote systems using SSH • 1.3 Archive, compress, unpack, and uncompress files using tar, gzip, and bzip2 • 1.4 Create and edit text files • 1.5 Create, delete, copy, and move files and directories • 1.6 Add users, reset passwords, modify user groups, and delete users • 1.7 Basic git operations • 1.8 List, set, and change file permissions • 1.9 Utilize a package management system • 1.10 Create a package • 2.1 Configure container engines, create, and manage containers • 2.2 Create a container image • 2.3 Build a container image • 2.4 Create and backup container volumes • 2.5 Deploy a database in a container • 3.1 Use a container orchestration system to run a multi-container environment • 3.2 Automate a deployment using popular automation tools • 3.3 Design a custom deployment for a development environment