ANSIBLE CHEAT SHEET

**Ansible**
- It is an open source engine that automates deployment, orchestration, cloud provisioning and other tools.
- It uses a playbook to describe jobs and uses YAML, which is human readable.
- It is designed for multi-tier deployment. It is agentless and works by connecting nodes through ssh.

**Environment Setup**
- Types of machines:
  - Control machine: manages other machines
  - Remote machine: controlled by other machines
  - Multiple remote systems can be handled by one machine.
  - Remote machine managing done by ansible by default.
- Ansible doesn’t leave any software running on them.
- Therefore there is no need of an upgrade when moving to a newer version.
- Install it through apt, yum, pkg, OpenCSW
- Installing it through apt:
  - $ sudo apt-get update
  - $ sudo apt-get install -y ansible
- Run ansible version to make sure it was installed properly.

**YAML**
- YAML syntax is used to express ansible playbooks
  - **Key-value pair**
    - Dictionary is represented in key-value pair
      - Ex: james:
        - name: jame john
        - rollo: 34
        - ehd: b
        - sex: male
      - Representing lists:
        - Each element has to be written in a new line with “-” as the prefix
        - countries:
          - America
          - Iceland
      - Lists inside the dictionary:
        - name: james john
        - rollo: 34
        - ehd: b

**Ad hoc Commands**
- General syntax of ad-hoc command:
  - Command hostgroup module/options/arguments

**FUNCTION**
- Check connectivity of hosts
- Check host system’s info
- Transferring files
- Create new user
- Delete user
- Check if package is installed

**COMMANDS**
- Ansible group - m ping
- Ansible group - m ls
- Ansible group - m copy a
- Ansible group - m user a name ansible pass: pass
- Ansible group - m user a name ansible state: absent
- Ansible group - m yum a name ansible state: latest
- Ansible group - m yum a name ansible state: present
- Ansible group - m service a name ansible state: started
- Ansible group - m service a name ansible state: stopped
- Ansible group - m service a name ansible state: restarted

**Variables**
- Same as using variables in programming languages
  - Ex: hosts = [”james john”]
  - tomatoc_port = 9080
  - Here tomatoc_port is assigned to 9080
- Keywords used:
  - Block: ansible syntax to execute a block
  - Name: name of the block
  - Action: the code that is to be executed
  - Register: renders the output
  - Always: states that below word will be run
  - Debug: displays the message

**Terms**
- Exception handling:
  - Similar to any other programming language
  - Keywords: rescue and always
  - The code is written in block.
  - It goes to the rescue phase and gets executed if the command in the block fails.
  - Thereby block is the same as “try block”, catch block is like “rescue” and always performs the same function as we know.

**Playbooks**
- It is the place where all YAML files are stored and executed.
  - Acts like a to-do list
  - YAML – another markup language
  - A playbook can have more than one play.
  - Plays map the instructions defined against a particular host.
  - Typically written in a text editor like notepad or notepad++
  - Sample playbook/Ansible file:
    - Name: install and configure DB
  - hosts: testServer
  - become: yes
  - vars:
    - oracle_db_port_value: 1521
  - tasks:
    - name: Install the Oracle DB
    - yum: id to Install the DB:
    - name: Install the Oracle DB:
    - name: Ensure the installed service is enabled service:
      - name: <your service name>
    - Tags of YAML:
      - Name: name of the playbook
      - Hosts: specifies the list of hosts. Tasks can be on the same machine or a different one.
      - Vars: defines the variables which you can use
      - Tasks: It is the list of action that needs to be performed. A task is always linked to a module.

**Troubleshooting**
- Common strategies to debug playbooks are:
  - Debug and register
  - Use verbosity
  - Playbook issues:
    - Quoting
    - Indentation
    - Some drawbacks are:
      - OS restrictions: is OS dependent so code on one OS will not work for another.
      - Once playbook is running, adding of hosts is not possible.
      - Error reporting is mediocre.