**Five Talent Operations Runbook Template**

**DO NOT ALTER THIS DOC – use SAVE AS to create a new doc!!**

This sample Runbook Template and descriptions are written from the point of view of an Operations Center. This could be an internal helpdesk or an outsourced ops center. Procedures and references should be modified to make the most logical sense for a given client situation. Although the goal is to develop "self healing" application stacks, that may not always be 100% possible. This runbook will outline tasks and procedures that require human decisions and/or interactions. Future development cycles should refer back to this "living document" runbook for items to be automated in future releases.

# Content Key

**Orange Text** = “instructional text”. When completing the template please review all instructional text to ensure complete understanding of the purpose of each section.

*Italicized Text* = “examples”*.* These examples are provided as guidance on to how to complete a section and supplement the information provided in the instructional text. In some cases example text (e.g. bullet lists) may be used as-is, added to, or deleted, while in other cases (e.g. sample table entries) it should be replaced with accurate, organization-specific information.

**Required Text** = it is probable that all organizations will need to retain and complete this section.

**Optional Text** = it is likely that only some organizations will need to include this section.

# Important Information Start Here

*All of Section 1 should be held to 1-2 pages of critical information, necessary for Operations Personnel to access during an unplanned downtime as well as daily operations*

## Five Talent Contact Info

**Required**

This section should contain the primary contact information of the Key Personnel at Five Talent software. It should only be 1-2 names, or perhaps the Five Talent 1-800 helpdesk line. If appropriate, perhaps the FTS Technical Operations Manager for this client and/or the Lead Developer. The names should be linked to Section 13 Key Contact Information, where other developers, other personnel should be listed.

*2.1.1 Technical Operations Manager – Wendi Smith*

*NOTE: Wendi should be CC'd on all Important and Critical alert*

*Office: 208-422-1234*

*Mobile: 208-901-1234*

*2.1.2 FTS ServiceDesk (non-dedicate phone number)*

*Request an AWS Engineer*

*877-111-1111*

*2.1.3 E-mail communication*

[*ABC\_Ticket@fivetalent.com*](mailto:ABC_Ticket@fivetalent.com) *- used to create an alert directly in the FTS ticketing system. Place 2-4 key words highlighting the alert in Subject line and details in the body of the e-mai*

[*ABC\_Ops@fivetalent.com*](mailto:ABC_Ops@fivetalent.com) *- Direct e-mail to Five Talent Operations focused on ABC Corp*

[*ABC\_MS@fivetalent.com*](mailto:ABC_MS@fivetalent.com) *- Direct e-mail to Five Talent Management overseeing ABC Corp*

## Client Contact Info

**Required**

This section should contain the primary contact information of 1-2 key members on the Client Side. Whom should be contacted in case of Emergencies and/or for highly important decisions. Key stakeholders may be the Business Owner, the CTO, or the Project Manager. The names should be linked to Section 13 Key Contact Information, where other developers, other personnel should be listed.

*2.2.1 ABC Corp IT Manager – Bill Jones*

*NOTE: Bill should be CC'd on all Important and Critical alert activity*

*Office: 208-422-1234*

*Mobile: 208-901-1234*

*2.2.2 ABC Corp CTO – Sally Newman*

*NOTE: Sally endi should be CC'd on all Critical alert activity.*

*Office: 208-422-1234*

*Mobile: 208-901-1234*

## Alert Acknowledgement Process

**Optional**

Are there any specific acknowledgement procedures for alerts? For example, all critical alerts received via pager must be acknowledged within 15 minutes and the CTO should be CC'd so he/she knows that someone has seen the alert and is working the issue.

*Be sure and CC Wendi & Bill on all important alerts*

*Be sure and CC Wendi, Sally, & Bill on all critical alerts*

## Connecting to AWS and other Environments

**Required**

A quick reference section for Connection Note to key environments. Examples may include the AWS Account number, and perhaps the personnel responsible for setting up new accounts.

*Section 2.4.1 AWS Console Login:* <https://console.aws.amazon.com/console/home>

*Section 2.4.2 Connecting to the ABC Corp on premise network*

*Everyone is require to have a personal account from ABC Corp. To receive this username account, contact the TOM*

*Section 2.4.3 Jumpbox access*

*Open a remote Desktop Connection to the jumpbox at 172.23.11.201*

*Section 2.4.4 Nagios access*

*From the Jumpbox, access the nagios web interface at* [*https://herkmonitor/*](https://herkmonitor/) *, log in with ABC standard account credentials*

## Notes

**Optional**

*High level summary of the Client's environment and infrastructure, so the person troubleshooting can quickly see the big picture and begin troubleshooting.*

*Example: Client ABC has an application running in the N. California region with 5-6 EC2 instances in a Custom VPC with multiple public and private subnets, 2 load balancers, an RDS instance, and several Lambda functions. Content is distributed via CloudFront. DNS records are externally managed via GoDaddy.*

# Alert Matrices

**Required**

Make sure that before you follow any of the Alert links, you are sure the host name, machine or device type (Windows Servers, Unix/Linux Servers, EC2 Instances, Network Devices, Storage Devices) by searching for the Hostname in this run book and verifying. Each Section of Alerts and processes are based on which type or device. All items in this section should ONLY be a hyperlink to the resolution steps in Section 4.

## Windows Server Alerts

*Section 3.1.1 PROBLEM: <HOSTNAME> is DOWN – Go Here*

*Section 3.1.2 PROBLEM: HTTP is <WARNING or CRITICAL> on host <HOSTNAME> - Go Here*

*Section 3.1.3 PROBLEM: Windows Disk\_<drive letter> is <WARNING or CRITICAL> on host <HOSTNAME> - Go Here*

## Linux alerts

### <Hostname> is currently Down

### <Hostname> network utilization is down

### <Hostname> CPU usage is down

### <Hostname> Memory usage is down

## Network Alerts

## Storage Alerts

## Database Alerts

## Service Alerts

## Application alerts

## Virtual Infrastructure Alerts

# Troubleshooting / Resolution Steps

**Required**

This section is the target of all the hyperlinks in Section 3.x. Often multiple alerts will all point to a single resolution process. Give details to either resolve the alert, OR gather information and escalate to the next level.

## Windows Server Alerts - Resolution

*Section 4.1.1 <HOSTNAME> is down*

1. *Connect to the ABC Corp network and Jumpbox*
2. *Start >> Run >> enter "cmd" and press enter*
3. *Type "ping <<hostname>> and check for a response*
4. *If it pings, RDP to the server and verify you can connect. If you can the alert should clear wihtin 5 minutes*
5. *If it is not responding to Ping, or the alert does not clear, verify if this is a physical server or if it is a VM server, or if it is an EC2 instance*
6. *If virtual, troubleshoot the issue using vSphere Client*
7. *If physical, check if iLO (HP Servers) or DRAC (Dell Servers) is available by going to https://<ServerName-ILO in a browser from the ABC Corp jumpbox*
8. *IF EC2, go to the AWS Console and drill down to the EC2 services*
9. *Attempt to reboot/restart the server*
10. *If unable to resolve, ensure all possibilities in the ABC Pre-Escalation Checklist and General Troubleshooting document have been exhausted, verify the priority of hte server/device, and escalate as defined*

*Section 4.1.3 Disk Drive Alerts*

1. *Connect to the ABC Corp network and Jumpbox*
2. *RDP into the Host*
3. *Verify which drive is having the space issues*
4. *Try to determine where the space is being taken up and escalate as defined*
5. *NOTE: If you notice within 5 minutes that the C: drive is constantly losing space – this becomes a Priority 1 – Critical for ANY SERVER. Escalate IMMEDIATELY*

## Linux alerts

## Network Alerts

## Storage Alerts

## Database Alerts

## Service Alerts

## Application alerts

## Virtual Infrastructure Alerts

# Escalation Processes

**Required**

The purpose of using a standardized prioritization model is to allow common definitions and understandings of business impact guide prioritization of cases and incidents based on real need. A standard prioritization mode removes emotion and perception from incident creation and allows support agents to respond with confidence based on impact and urgency to the business

## Impact

A measure of the effect of an Incident, Problem, or Change on Business Processes. Impact is often based on how Service Levels will be affected. Impact and Urgency are used to assign Priority.

|  |  |
| --- | --- |
| Impact Assessment Only needed IF Priority is not found | |
| ***I1*** | Severe stoppage or unrecoverable impact of IT services that are directly related to revenue generating processes. |
| ***I2*** | Significant degradation or partial outage to IT services that are directly related to revenue generating processes. |
| ***I3*** | Impact to non-critical or non-revenue generating IT services that support or will eventually affect critical business or revenue generating IT Services. |
| ***I4*** | Impact to systems that are not tied in any way to revenue generating IT Services. |

## Urgency

A measure of how long it will be until an Incident, Problem, or Change has a significant impact on the business. For example, a high impact incident may have low urgency if the impact will not affect the business until the end of the financial year. Impact and Urgency are used to assign Priority.

|  |
| --- |
| Urgency Assessment Only needed IF Priority is not found |

|  |  |
| --- | --- |
| U1 | Impact within 0-4 hours. |
| U2 | Impact within 4-12 hours. |
| U3 | Impact within 13 hours to 6 days. |
| U4 | Impact after 7 days or some unknown time. |

## Priority

A category used to identify the relative importance of an Incident, Problem or Change. Priority is based on Impact and Urgency, and is used to identify required times for actions to be taken.

|  |
| --- |
| Priority Calculation |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **I1** | **I2** | **I3** | **I4** |
| **U1** | **1 - Critical** | **1 - Critical** | **2 - Major** | **3 - Normal** |
| **U2** | **1 - Critical** | **2 - Major** | **3 - Normal** | **3 - Normal** |
| **U3** | **2 - Major** | **3 - Normal** | **4 - Low** | **4 - Low** |
| **U4** | **3 - Normal** | **3 - Normal** | **4 - Low** | **4 - Low** |
|  |  |  |  |  |

|  |
| --- |
| Client Notification Priority table |

|  |  |
| --- | --- |
| 1 – Critical  (P1) | - Level 1 Technician, Escalate to OC Admin if Alert cannot be cleared  - OC Admin, Escalation to Taos Oncall if Escalation criteria has been meet  - Taos Oncall, Notify Onyx Management every 2 hours until Alert has been cleared |
| 2 – Major  (P2) | - Level 1 Tech, Escalate to OC Admin if Escalation criteria has been meet  - OC Admin, Escalate to Taos Oncall if Escalation criteria has been meet  - Taos Oncall, Notify Onyx Management every 4 hours until Alert has been cleared |
| 3 – Normal  (P3) | - Level 1 Tech, Escalate to OC Admin if Escalation criteria has been meet  - OC Admin, Escalate to Taos Oncall if Escalation criteria has been meet  - Taos Oncall, Notify Onyx Management each day until Alert has been cleared |
| 4 – Low  (P4) | - Level 1 Tech, Escalate to OC Admin if Escalation criteria has been meet  - OC Admin, Escalate to Taos Oncall if Escalation criteria has been meet  - Taos Oncall, Notify Onyx Management each day until Alert has been cleared |

Escalation Criteria

* Has the documentation been reviewed?
* Has the ticket been worked on thoroughly?
* Are you unable to find a workaround?

If all three are checked then Escalate to next Level.

## Contact order

1st Network Operations Manager

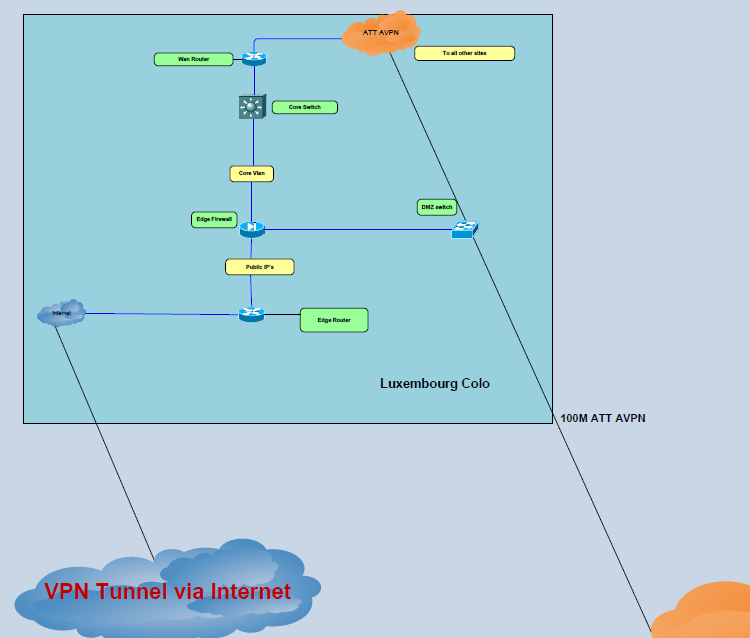
2nd Client IT manager

3rd Head of IT

# Network Diagrams

**Required**

Insert diagrams of the OnPremise and Cloud network diagrams to give engineers an overview of the environment. Often a single image is used here, with hyperlinks to the original source Visio or LucidChart diagrams. Where appropriate, show security rules and access points, including VPN connections, Direct Connect, etc.



# Monitoring System

**Optional**

Monitoring systems can be implemented to watch for various metrics on systems and processes. Include information here to log in and utilize the tool. Often a tool will send an email alert; this section should be a reference to understanding and responding to the alerts.

*Example*

*7.1 Connect to the OpsView Monitoring System*

*1. Connect to the ABC Corp network using the Cisco AnyConnect VPN*

*2. Open a browser and connect to the OpsView web interface –* [*http://irv-opsview101.ra.local:3000*](http://irv-opsview101.ra.local:3000)

*3. Login using credentials found in Password Manager Pro*

# Resource List

**Required**

The resource list should include all the Company IT assets regardless of location – both OnPremise and in the Cloud. Where possible, an automate tool should be used to capture information. Use judgement on how much detail to list per system. Critical Production systems will receive more attention then simple Development systems. One note to keep in mind is the “connected” nature of various components – What actions trigger a lambda function? What S3 buckets are used with a particular website?

## Windows Servers

## Linux Servers

## Networks

## Storage

## Database Instances / RDS

## Lambda Functions

## EC2 Servers

# Environment Overview

**Optional**

*Has more detail then Section 2.5. High level summary of the Client's environment and infrastructure, if could list out summary information about departments, the typical development cycle in place, preferences for certain OS levels, or perhaps target environments – Dev, Test, Prod. Also include references to other sources of information and/or people resources.*

# Supporting SOPs

**Optional**

*This sections has various procedures to setup and/or fix things within a specific application. For example, under VMware, could include a few sections on the ESXi environment.*

## Change Management

## Release Management

## Patching and Updates

## Amazon Web Services

## Citrix

## VMware

## Windows

## Storage

## Linux

## Network Devices

## Application

**Optional**

The following is a very comprehensive section for listing out details and procedures for an “Application”. In this context, an application should be thought of as an entire stack – EC2, Database, messaging, queues, web servers, load balancers. This section could be a complete document on it’s own. By asking and filing out the sub-sections, the client AND Five Talent will gain a deep understanding of a given application.

### Service or system overview

**Service or system name:**

### Business overview

What business need is met by this service or system? What expectations do we have about availability and performance?

*(e.g. Provides reliable automated reconciliation of logistics transactions from the previous 24 hours)*

### Technical overview

What kind of system is this? Web-connected order processing? Back-end batch system? Internal HTTP-based API? ETL control system?

*(e.g. Internal API for order reconciliation based on Ruby and RabbitMQ, deployed in Docker containers on Kubernetes)*

### Service Level Agreements (SLAs)

What explicit or implicit expectations are there from users or clients about the availability of the service or system?

*(e.g. Contractual 99.9% service availability outside of the 03:00-05:00 maintenance window)*

### Service owner

Which team owns and runs this service or system?

*(e.g. The Sneaky Sharks team (Bangalore) develops and runs this service:* [*sneaky.sharks@company.com*](https://mail.google.com/mail/?view=cm&fs=1&tf=1&to=sneaky.sharks@company.com) */ #sneaky-sharkson Slack / Extension 9265)*

### Contributing applications, daemons, services, middleware

Which distinct software applications, daemons, services, etc. make up the service or system? What external dependencies does it have?

*(e.g. Ruby app + RabbitMQ for source messages + PostgreSQL for reconciled transactions)*

### System characteristics

### Hours of operation

During what hours does the service or system actually need to operate? Can portions or features of the system be unavailable at times if needed?

#### Hours of operation - core features

*(e.g. 03:00-01:00 GMT+0)*

#### Hours of operation - secondary features

*(e.g. 07:00-23:00 GMT+0)*

### Data and processing flows

How and where does data flow through the system? What controls or triggers data flows?

*(e.g. mobile requests / scheduled batch jobs / inbound IoT sensor data )*

### Infrastructure and network design

What servers, containers, schedulers, devices, vLANs, firewalls, etc. are needed?

*(e.g. '10+ Ubuntu 14 VMs on AWS IaaS + 2 AWS Regions + 2 VPCs per Region + Route53')*

### Resilience, Fault Tolerance (FT) and High Availability (HA)

How is the system resilient to failure? What mechanisms for tolerating faults are implemented? How is the system/service made highly available?

*(e.g. 2 Active-Active data centres across two cities + two or more nodes at each layer)*

### Throttling and partial shutdown

How can the system be throttled or partially shut down e.g. to avoid flooding other dependent systems? Can the throughput be limited to (say) 100 requests per second? etc. What kind of connection back-off schemes are in place?

#### Throttling and partial shutdown - external requests

*(e.g. Commercial API gateway allows throttling control)*

#### Throttling and partial shutdown - internal components

*(e.g. Exponential backoff on all HTTP-based services + /health healthcheck endpoints on all services)*

### Expected traffic and load

Details of the expected throughput/traffic: call volumes, peak periods, quiet periods. What factors drive the load: bookings, page views, number of items in Basket, etc.)

*(e.g. Max: 1000 requests per second with 400 concurrent users - Friday @ 16:00 to Sunday @ 18:00, driven by likelihood of barbecue activity in the neighborhood)*

#### Hot or peak periods

\_

#### Warm periods

\_

#### Cool or quiet periods

\_

### Environmental differences

What are the main differences between Production/Live and other environments? What kinds of things might therefore not be tested in upstream environments?

*(e.g. Self-signed HTTPS certificates in Pre-Production - certificate expiry may not be detected properly in Production)*

### Tools

What tools are available to help operate the system?

*(e.g. Use the queue-cleardown.sh script to safely cleardown the processing queue nightly)*

### Required resources

What compute, storage, database, metrics, logging, and scaling resources are needed? What are the minimum and expected maximum sizes (in CPU cores, RAM, GB disk space, GBit/sec, etc.)?

### Required resources - compute

*(e.g. Min: 4 VMs with 2 vCPU each. Max: around 40 VMs)*

### Required resources - storage

*(e.g. Min: 10GB Azure blob storage. Max: around 500GB Azure blob storage)*

### Required resources - database

*(e.g. Min: 500GB Standard Tier RDS. Max: around 2TB Standard Tier RDS)*

### Required resources - metrics

*(e.g. Min: 100 metrics per node per minute. Max: around 6000 metrics per node per minute)*

### Required resources - logging

*(e.g. Min: 60 log lines per node per minute (100KB). Max: around 6000 log lines per node per minute (1MB))*

### Required resources - other

*(e.g. Min: 10 encryption requests per node per minute. Max: around 100 encryption requests per node per minute)*

### Security and access control

### Password and PII security

What kind of security is in place for passwords and Personally Identifiable Information (PII)? Are the passwords hashed with a strong hash function and salted?

*(e.g. Passwords are hashed with a 10-character salt and SHA265)*

### Ongoing security checks

How will the system be monitored for security issues?

*(e.g. The ABC tool scans for reported CVE issues and reports via the ABC dashboard)*

### System configuration

### Configuration management

How is configuration managed for the system?

*(e.g. CloudInit bootstraps the installation of Puppet - Puppet then drives all system and application level configuration except for the XYZ service which is configured via App.config files in Subversion)*

### Secrets management

How are configuration secrets managed?

*(e.g. Secrets are managed with Hashicorp Vault with 3 shards for the master key)*

### System backup and restore

### Backup requirements

Which parts of the system need to be backed up?

*(e.g. Only the CoreTransactions database in PostgreSQL and the Puppet master database need to be backed up)*

### Backup procedures

How does backup happen? Is service affected? Should the system be [partially] shut down first?

*(e.g. Backup happens from the read replica - live service is not affected)*

### Restore procedures

How does restore happen? Is service affected? Should the system be [partially] shut down first?

*(e.g. The Booking service must be switched off before Restore happens otherwise transactions will be lost)*

### Monitoring and alerting

### Log aggregation solution

What log aggregation & search solution will be used?

*(e.g. The system will use the existng in-house ELK cluster. 2000-6000 messages per minute expected at normal load levels)*

### Log message format

What kind of log message format will be used? Structured logging with JSON? log4j style single-line output?

*(e.g. Log messages will use log4j compatible single-line format with wrapped stack traces)*

### Events and error messages

What significant events, state transitions and error events may be logged?

*(e.g. IDs 1000-1999: Database events; IDs 2000-2999: message bus events; IDs 3000-3999: user-initiated action events; ...)*

### Metrics

What significant metrics will be generated?

*(e.g. Usual VM stats (CPU, disk, threads, etc.) + around 200 application technical metrics + around 400 user-level metrics)*

### Health checks

How is the health of dependencies (components and systems) assessed? How does the system report its own health?

#### Health of dependencies

*(e.g. Use /health HTTP endpoint for internal components that expose it. Other systems and external endpoints: typically HTTP 200 but some synthetic checks for some services)*

#### Health of service

*(e.g. Provide /health HTTP endpoint: 200 --> basic health, 500 --> bad configuration + /health/deps for checking dependencies)*

### Operational tasks

### Deployment

How is the software deployed? How does roll-back happen?

*(e.g. We use GoCD to coordinate deployments, triggering a Chef run pulling RPMs from the internal yum repo)*

### Batch processing

What kind of batch processing takes place?

*(e.g. Files are pushed via SFTP to the media server. The system processes up to 100 of these per hour on a cron schedule)*

### Power procedures

What needs to happen when machines are power-cycled?

*(e.g. \*\*\* WARNING: we have not investigated this scenario yet! \*\*\*)*

### Routine and sanity checks

What kind of checks need to happen on a regular basis?

*(e.g. All /health endpoints should be checked every 60secs plus the synthetic transaction checks run every 5 mins via Pingdom)*

### Troubleshooting

How should troubleshooting happen? What tools are available?

*(e.g. Use a combination of the /health endpoint checks and the abc-\*.sh scripts for diagnosing typical problems)*

### Maintenance tasks

### Patching

How should patches be deployed and tested?

#### Normal patch cycle

*(e.g. Use the standard OS patch test cycle together with deployment via Jenkins and Capistrano)*

#### Zero-day vulnerabilities

*(e.g. Use the early-warning notifications from UpGuard plus deployment via Jenkins and Capistrano)*

### Daylight-saving time changes

Is the software affected by daylight-saving time changes (both client and server)?

*(e.g. Server clocks all set to UTC+0. All date/time data converted to UTC with offset before processing)*

### Data cleardown

Which data needs to be cleared down? How often? Which tools or scripts control cleardown?

*(e.g. Use abc-cleardown.ps1 run nightly to clear down the document cache)*

### Log rotation

Is log rotation needed? How is it controlled?

*(e.g. The Windows Event Log ABC Service is set to a maximum size of 512MB)*

### Failover and Recovery procedures

What needs to happen when parts of the system are failed over to standby systems? What needs to during recovery?

### Failover

\_

### Recovery

\_

### Troubleshooting Failover and Recovery

What tools or scripts are available to troubleshoot failover and recovery operations?

*(e.g. Start with running SELECT state\_\_desc FROM sys.database\_\_mirroring\_\_endpoints on the PRIMARY node and then use the scripts in the db-failover Git repo)*

# Backup / Restore

**Optional**

*This sections contains information about*

# Antivirus and Related Security

**Optional**

*This sections contains information about*

# Systems Provisioning

**Optional**

*This sections contains information about*

# Key Contact Information

**Optional**

*This sections contains information about*

# AWS IAM Users, Groups, Roles

**Optional**

*This sections contains information about*

# Appendix

**Optional**

*This sections contains information about*

## Customer Processes to be known

## Other environment-specific information for reference

## Index of Terms

# Document Revision History

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Revision Number* | *Revision Date* | *Section Reference* | *Summary of Changes* | *Author* | *Reviewer* |
| *1.0* | *July 10, 2018* | *Entire Doc* | *Setup structure and headings, TOC* | *Brian Keith* |  |
| *1.2* | *July 25, 2018* | *Section 8.1 Tagging Categories* | *Listing of Basic Tag categories* | *Brian Keith* |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

# Headings Check

### Heading 3

#### Heading 4

##### Heading 5

###### Heading 6

Heading 7

Heading 8

Heading 9